

**КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ПО ПРОГРАММЕ
ПОДГОТОВКИ СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА ПО
СПЕЦИАЛЬНОСТИ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО
ОБРАЗОВАНИЯ**

09.02.07

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ

- ПМ.01 Разработка модулей программного обеспечения для компьютерных систем
МДК.01.01 Разработка программных модулей
МДК.01.02 Поддержка и тестирование программных модулей
МДК.01.03 Разработка мобильных приложений
МДК.01.04 Системное программирование
ПМ.01.ЭК Экзамен по модулю
- ПМ.02 Осуществление интеграции программных модулей
МДК.02.01 Технология разработки программного обеспечения
МДК.02.02 Инструментальные средства разработки программного обеспечения
МДК.02.03 Математическое моделирование
ПМ.02.ЭК Экзамен по модулю
- ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем
МДК.04.01 Внедрение и поддержка компьютерных систем
МДК.04.02 Обеспечение качества функционирования компьютерных систем
ПМ.04.ЭК Экзамен по модулю
- ПМ.11 Разработка, администрирование и защита баз данных
МДК.11.01 Технология разработки и защиты баз данных
ПМ.11.ЭК экзамен по модулю

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

МДК 01.01 Разработка программных модулей

Подготовки специалистов среднего звена по специальности

код специальности 09.02.07 Информационные системы и программирование

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности (профессии) СПО 09.02.07 Информационные системы и программирование базовой подготовки программы МДК 01.01 Разработка программных модулей.

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчик: Бахаев Д. В., преподаватель,
Голубенко М. И., преподаватель.

ОДОБРЕН
цикловой методической комиссией
Протокол № 1
от «3» сентября 2022 г.
Председатель Гаврикова Е. Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения МДК, подлежащие проверке
3. Оценка освоения МДК
 - 3.1. Контроль и оценка освоения МДК
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет материалов
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения МДК Разработка программных модулей обучающийся должен обладать предусмотренными ФГОС по специальности/профессии СПО 09.02.07 Информационные системы и программирование базового уровня следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями

Обучающийся должен знать:

З 1	- основные этапы разработки программного обеспечения;
З 2	- основные принципы технологии структурного и объектно-ориентированного программирования;
З 3	- способы оптимизации и приемы рефакторинга;
З 4	- основные принципы отладки и тестирования программных продуктов.

Обучающийся должен уметь:

У 1	- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
У 2	- создавать программу по разработанному алгоритму как отдельный модуль;
У 3	- выполнять отладку и тестирование программы на уровне модуля;
У 4	- осуществлять разработку кода программного модуля на современных языках программирования;
У 5	- уметь выполнять оптимизацию и рефакторинг программного кода;
У 6	- оформлять документацию на программные средства.

Обучающийся должен иметь практический опыт:

- в разработке кода программного продукта на основе готовой спецификации на уровне модуля;
- в использовании инструментальных средств на этапе отладки программного продукта;
- в проведении тестирования программного модуля по определенному сценарию;
- в использовании инструментальных средств на этапе отладки программного продукта;
- в разработке мобильных приложений.

Формируемые ОК:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

Формируемые ПК:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ.

Личностные результаты учебной дисциплины

Код ЛР	Личностные результаты реализации программы (дескрипторы)
ЛР 1	-осознающий себя гражданином и защитником великой страны
ЛР 2	-проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	-демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих
ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личностного и профессионального, конструктивного «цифрового следа»

ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	- проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	- уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 10	- принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР 17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
ЛР 19	- профессиональная идентичность и ответственность
ЛР 20	- самооценка и рефлексия результатов своей деятельности и развития

Формами промежуточной аттестации по МДК являются другие формы контроля – 5 семестр, дифференцированный зачёт – 6 семестр, курсовой проект – 6 семестр.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по МДК осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих и профессиональных компетенций:

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней; ОК 1, 2, 3, 5, 9, 10; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля.
У 2. Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 3, 5, 9, 10; ПК 1.1, 1.2.	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма.
У 3. Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 3, 5, 9, 10; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля.
У 4. Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 3, 5, 9, 10; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.
У 5. Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 3, 5, 9, 10; ПК 1.4, 1.5.	Производит оптимизацию программного кода и его рефакторинг.
У 6. Оформлять документацию на программные средства. ОК 1, 2, 3, 4, 5, 6, 7, 9, 10; 11 ПК 1.2.	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства.
Знать:	
З 1. Основные этапы разработки программного обеспечения ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.1, 1.2.	Знает основные процессы жизненного цикла программного обеспечения; на основе процессов понимает последовательность разработки программного обеспечения.
З 2. Основные принципы технологии структурного и объектно-ориентированного программирования; ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.1, 1.2.	Понимает принципиальные различия в технологиях структурного и объектно-ориентированного программирования; определяет на основе какой технологии построен код программы;
З 3. Способы оптимизации и приемы рефакторинга; ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.3, 1.4, 1.5.	Знает основные способы оптимизации и рефакторинга кода; применяет методы к написанному коду; составляет код программы на основе методов.
З 4. Основные принципы отладки и тестирования программных продуктов. ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.3, 1.4.	Понимает последовательность действий при отладке программного продукта; осознает принципы и методы тестирования; создает тестовые кейсы на основе различных методов тестирования.

3.ОЦЕНКА ОСВОЕНИЯ МДК

3.1. Контроль и оценка освоения МДК по темам (разделам)

Элемент МДК	Текущий контроль		Промежуточная аттестация	
	Осваиваемые результаты	Метод контроля	Проверяемые результаты	Форма контроля
Раздел 1. Разработка программных модулей				
Тема 1.1. Жизненный цикл ПО	ПК 1.1 ПК 1.2 ОК 1-ОК 9 З 1 У 6 ЛР7, ЛР13, ЛР15	Устный опрос	ОК 1 -10 ПК 1.1 – 1.6 У 1 -6 З 1 - 4	5-й семестр – контрольная работа работа, 6 семестр – курсовой проект, дифференцированный зачет.
Тема 1.2 Структурное программирование	ПК 1.1 ПК 1.2 ОК 1-ОК 9 З 1, 2 У 1, 2, 6 ЛР7, ЛР13, ЛР15	Устный опрос Самостоятельная работа Практическая работа Тестовое задание		
Тема 1.3 Объектно-ориентированное программирование	ПК 1.1 ПК 1.2 ОК 1-ОК 9 З 1, 2 У 1-У 4, У 6 ЛР7, ЛР13, ЛР15	Устный опрос Самостоятельная работа Практическая работа Тестовое задание		
Тема 1.4 Паттерны программирования	ПК 1.1 ПК 1.2 ОК 1-ОК 9 З 1, 2 У 1-У 4, У 6 ЛР7, ЛР13, ЛР15	Устный опрос Самостоятельная работа Практическая работа		
Тема 1.5 Событийно-управляемое программирование	ПК 1.1 ПК 1.2 ОК 1-ОК 9 З 1, 2, 4 У 1 – У 4 ЛР7, ЛР13, ЛР15	Устный опрос Самостоятельная работа Практическая работа Тестовое задание		

Тема 1.6 Оптимизация и рефакторинг кода	ПК 1.4 ПК 1.5 ОК 1-ОК 9 З 1- З 4 У 1- У 6 ЛР7, ЛР13, ЛР15	Устный опрос Самостоятельная работа Практическая работа		
Тема 1.7 Разработка пользовательского интерфейса	ПК 1.2 ОК 1-ОК 9 З 1 – З 4 У 1 – У 6 ЛР7, ЛР13, ЛР15	Устный опрос Самостоятельная работа Практическая работа Тестовое задание		
Тема 1.8 Основы ADO.Net	ПК 1.1 ПК 1.2 ОК 1-ОК 9 З 1 – З 4 У 1 – У 6 ЛР7, ЛР13, ЛР15	Устный опрос Самостоятельная работа Практическая работа Тестовое задание		

3.1.1. Методы и критерии оценивания

1. Устный опрос. Критерии оценивания.

Оценка 5 «отлично» - ответил на вопросы в объеме лекционного и дополнительного материала, дал полные грамотные ответы на все дополнительные вопросы.

Оценка 4 «хорошо» - грамотно изложил ответы на вопросы, но содержание и формулировки имеют отдельные неточности (допускается нечеткая формулировка определений), в полной мере ответил на заданные дополнительные вопросы.

Оценка 3 «удовлетворительно» - ответил на часть вопросов в объеме лекционного материала и ответил на часть дополнительных вопросов.

Оценка 2 «неудовлетворительно» - допустил ошибки в определении базовых понятий, искажил их смысл, не ответил на дополнительные вопросы.

2. Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

3. Самостоятельная работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме; учтены все требования к данной работе; самостоятельно поставлены цели и задачи работы, соответствующие заданной теме/проблеме; получены результаты в соответствии с поставленной целью; работа оформлена аккуратно и грамотно.

Оценка 4 «хорошо» - выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью, и объем выполненной части работы позволяет получить недостаточно результатов в соответствии с поставленной целью.

4. Практическая работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме, с соблюдением алгоритма выполнения: последовательности проведения измерений, заполнения таблиц, графиков и др.; правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; получены результаты в соответствии с поставленной целью.

Оценка 4 «хорошо»- выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью и объем выполненной части работы не позволяет получить никаких результатов в соответствии с поставленной целью.

4. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Контрольная работа

1. Форма проведения: письменная – тестовое задание

2. Условия выполнения:

1. Инструкция для обучающихся:

2. Время выполнения: 45 минут.

3. Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.

4. Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда виртуализации VMWare Work station версии 12 и выше; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

5. Информационные источники, допустимые к использованию на экзамене: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК»

6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения промежуточной аттестации

1. Перечень тем (разделов), выносимых на экзамен:

Тема 1.1. Жизненный цикл ПО;

Тема 1.2 Структурное программирование;

Тема 1.3 Объектно-ориентированное программирование.

Тема 1.4 Паттерны программирования.

Тема 1.5 Событийно-управляемое программирование

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У 1. Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 03, ОК 09, ОК 10, ПК 4.2	- настройка предложенного программного обеспечения, обоснование варианта конфигурации.
У 2. Использовать методы защиты программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 03, ОК 09, ОК 10, ПК 4.4	- подбор и настройка оборудования и программного обеспечения, направленного на защиту компьютерных систем; - использование корректных систем защиты для различных видов аппаратного и программного обеспечения.
У 3. Проводить инсталляцию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 03, ОК 09, ОК 10, ПК 4.1	- организация доступа к программному обеспечению различным категориям пользователей; - обеспечивать совместимость компонент с ранее установленными программными продуктами.
У 4. Производить настройку отдельных компонентов программного обеспечения компьютерных систем ОК 01, ОК 02, ОК 03, ОК 09, ОК 10, ПК 4.3	- настройка программного обеспечения отраслевой направленности для обеспечения его бесперебойной работы; - предложение вариантов модификации программного обеспечения.
У 5. Анализировать риски и характеристики качества программного обеспечения ОК 01, ОК 02, ОК 03, ОК 09, ОК 10, ПК 4.4	- анализ условий эксплуатации программного обеспечения; проверка настроек конфигурации; - анализ функционирования с помощью инструментальных средств; определение причин несоответствия выполняемых функций требованиям заказчика.
Знать:	
З 1: Основные методы и средства эффективного анализа; ОК 01, ОК 02, ОК 05, ОК 09, ОК 10, ПК 4.2	- перечисление основных этапов жизненного цикла программного обеспечения; - определение основных методов и средств эффективного анализа программного и аппаратного обеспечения.
З 2: Основные виды работ на этапе сопровождения программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ОК 10, ПК 4.3	- демонстрирует знание основных видов работ на этапах внедрения и сопровождения программного обеспечения.
З 3: Основные принципы контроля конфигурации и поддержки целостности	- формулирует основные принципы контроля конфигурации программного

конфигурации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ОК 10, ПК 4.3	обеспечение и принципы поддержки целостности конфигурации программного обеспечения.
34: Средства защиты программного обеспечения в компьютерных системах; ОК 01, ОК 02, ОК 05, ОК 09, ОК 10, ПК 4.4	- определяет основные средства защиты программного обеспечения в компьютерных системах и способы их настройки и применения.

Образец тестового задания

Вариант 1

1. Алгоритм - это:
 - a) Указание на выполнение действий,
 - b) Система правил, описывающая последовательность действий, которые необходимо выполнить для решения задачи,
 - c) Процесс выполнения вычислений, приводящих к решению задачи
2. Свойство алгоритма – дискретность, выражает, что:
 - a) Команды должны следовать последовательно друг за другом,
 - b) Каждая команда должна быть описана в расчете на конкретного исполнителя,
 - c) Разбиение алгоритма на конечное число команд
3. Формальное исполнение алгоритма – это:
 - a) Исполнение алгоритма конкретным исполнителем с полной записью его рассуждений,
 - b) Разбиение алгоритма на конкретное число команд и пошаговое их исполнение,
 - c) Исполнение алгоритма не требует рассуждений, а осуществляется исполнителем автоматически
 - d) Исполнение алгоритма осуществляется исполнителем на уровне его знаний
4. Самое важное свойство алгоритма:
 - a) Визуальность,
 - b) Массовость,
 - c) Дискретность,
 - d) Аудиальность,
5. Какой алгоритм называется линейным:
 - a) Выполнение операций зависит от условия,
 - b) Операции выполняются друг за другом,
 - c) Одни и те же операции выполняются многократно
 - d) Присутствие всех возможных операций в одном алгоритме
6. Графическое задание алгоритма – это:
 - a) Способ представления алгоритма с помощью геометрических фигур,
 - b) Представление алгоритма в форме таблиц и расчетных формул,
 - c) Система обозначений и правил для единообразной и точной записи алгоритмов и их исполнения
7. В расчете на кого должен строиться алгоритм:
 - a) В расчете на ЭВМ,
 - b) В расчете на умственные способности товарища,

- c) В расчете на конкретного исполнителя
- 8. Какое из перечисленных свойств относится к свойствам алгоритма:
 - a) Визуальность,
 - b) Совокупность,
 - c) Аудиальность,
 - d) Понятность
- 9. Система программирования – это:
 - a) комплекс любимых программ программиста
 - b) комплекс программ, облегчающий работу программиста
 - c) комплекс программ, обучающих начальным шагам программиста
- 10. Использование одного имени для задания общих для класса действий, что означает способность объектов выбирать внутренний метод, исходя из типа данных, определяет свойство ООП
 - a) Полиморфизм
 - b) Управление событиями
 - c) Инкапсуляция
 - d) Наследование
- 11. Свойство ООП, которое может быть смоделировано с помощью таксономической классификационной схемы (иерархии) называется
 - a) Инкапсуляция
 - b) Управление событиями
 - c) Полиморфизм
 - d) Наследование
- 12. Понятие класса в ООП включает в себя
 - a) Поля и методы класса
 - b) Процедуры и функции обработки
 - c) Поля и функции обработки
 - d) Поля и процедуры обработки
- 13. Назначение конструктора объекта
 - a) Только выделяет память под объект
 - b) Выделяет память и задает начальное значение полям
 - c) Задает начальное значение полям
 - d) Выделяет память, задает начальное значение полям, выполняет любые проверки, заданные программистом
- 14. Моделями типа «черный ящик» являются
 - a) Модели мышления
 - b) Модели, описывающие зависимость параметров состояния объекта от входных параметров
 - c) Модели, описывающие входные и выходные параметры объекта без учета внутренней структуры объекта
 - d) Модели «аварийного» ящика на самолетах
- 15. Моделями типа «белый ящик» являются
 - a) Модели мышления
 - b) Модели, описывающие зависимость параметров состояния объекта от входных параметров
 - c) Модели, описывающие входные и выходные параметры объекта с учетом

внутренней структуры объекта

- d) Модели, описывающие выходные данные в программе
16. Программа «заглушка» служит для
- a) запуска программы на выполнение
 - b) имитации запуска другой программы на выполнение
 - c) проверки правильности работы программы
 - d) имитации передачи параметров в другой модуль
17. Какие методы сборки программы существуют
- a) монолитная
 - b) пошаговая
 - c) одновременная
 - d) постепенная
18. Какой метод тестирования программы учитывает закон распределения входных данных
- a) детерминированное тестирование
 - b) функциональное тестирование
 - c) стохастическое тестирование
 - d) логическое тестирование
19. Программирование сверху вниз – это
- a) Процесс, при котором от начального предположения осуществляется движение по направлению к лучшим решениям
 - b) Процесс пошагового разбиения алгоритма на все более мелкие части с целью получения таких элементов, для которых можно написать конкретные команды
 - c) Метод сведения трудной задачи к последовательности более простых
 - d) Исследование древовидной модели пространства решений и ориентация на поиск оптимального решения
20. Загрузочный модуль программы – результат работы
- a) Грамматики
 - b) Транслятора
 - c) Интерпретатора
 - d) Редактора связей (компоновщика)
21. Интегрированная система программирования включает компонент для перевода исходного текста программы в машинный код, который называется
- a) строителем кода
 - b) компилятором
 - c) переводчиком
 - d) преобразователем
22. Результатом компиляции программы на языке высокого уровня является
- a) Командный файл
 - b) Объектный файл
 - c) Исходный текст программы на языке высокого уровня
 - d) Дисплейный файл
23. Порядковый номер элемента в массиве, который всегда начинается с нуля
- a) Индекс
 - b) Команда
 - c) Псевдокоманда

- d) Макрокаманда
- 24. Объектно-ориентированная программа - ...
 - a) Программирование, основанное на объектах
 - b) Совокупность объектов и способов их взаимодействия
 - c) Структура среды программирования
- 25. Модуль – это...
 - a) Средство для создания, изменения исходных файлов, которые содержат написанную на языке программирования программу
 - b) Автономно компилируемая программная единица, включающая в себя различные компоненты раздела описаний (типы, константы, переменные, процедуры и функции) и, возможно некоторые исполняемые операторы иницилирующей части
 - c) Набор инструментов, которые используются для преобразования символов
- 26. Атрибуты (основные характеристики), которые описывают особенности объекта (цвет, ширина, положение и т.д.)
 - a) Объект
 - b) Событие
 - c) Свойство
- 27. Совокупность данных (компонентов) и методов работы с ними
 - a) Объект
 - b) Событие
 - c) Свойство
- 28. Отклик на внешнее воздействие
 - a) Объект
 - b) Событие
 - c) Свойство

4.Эталоны ответов

1) В; 2) С; 3) С; 4) В; 5) В; 6) А; 7) С; 8) D; 9) В; 10) С; 11) D; 12) А; 13) А; 14) С; 15) В; 16) В; 17) В, С; 18) D; 19) В; 20) В; 21) В; 22) В; 23) А; 24) В; 25) В; 26) С; 27) А; 28) В.

Критерии оценки ответов

Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

2. Дифференцированный зачет

1.Форма проведения: письменная – тестовое задание, практико-ориентированное задание

2.Условия выполнения:

1.Инструкция для обучающихся:

2.Время выполнения: 90 минут.

3.Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.

4.Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда программирования PyCharm с установленным интерпретатором Python версии 3.8; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

5. Информационные источники, допустимые к использованию на экзамене:
<http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК»

6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения промежуточной аттестации

1. Перечень тем (разделов), выносимых на экзамен:

Тема 1.1. Жизненный цикл ПО;

Тема 1.2 Структурное программирование;

Тема 1.3 Объектно-ориентированное программирование.

Тема 1.4 Паттерны программирования.

Тема 1.5 Событийно-управляемое программирование.

Тема 1.6 Оптимизация и рефакторинг кода.

Тема 1.7 Разработка пользовательского интерфейса.

Тема 1.8 Основы ADO.Net.

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У 1. Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 03, ОК 09, ОК 10, ПК 4.2	- настройка предложенного программного обеспечения, обоснование варианта конфигурации.
У 2. Использовать методы защиты программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 03, ОК 09, ОК 10, ПК 4.4	- подбор и настройка оборудования и программного обеспечения, направленного на защиту компьютерных систем; - использование корректных систем защиты для различных видов аппаратного и программного обеспечения.
У 3. Проводить инсталляцию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 03, ОК 09, ОК 10, ПК 4.1	- организация доступа к программному обеспечению различным категориям пользователей; - обеспечивать совместимость компонент с ранее установленными программными продуктами.
У 4. Производить настройку отдельных компонентов программного обеспечения компьютерных систем ОК 01, ОК 02, ОК 03, ОК 09, ОК 10, ПК 4.3	- настройка программного обеспечения отраслевой направленности для обеспечения его бесперебойной работы; - предложение вариантов модификации программного обеспечения.
У 5. Анализировать риски и характеристики качества программного обеспечения ОК 01, ОК 02, ОК 03, ОК 09, ОК 10, ПК 4.4	- анализ условий эксплуатации программного обеспечения; проверка настроек конфигурации; - анализ функционирования с помощью инструментальных средств; определение

	причин несоответствия выполняемых функций требованиям заказчика.
Знать:	
З 1: Основные методы и средства эффективного анализа; ОК 01, ОК 02, ОК 05, ОК 09, ОК 10, ПК 4.2	- перечисление основных этапов жизненного цикла программного обеспечения; - определение основных методов и средств эффективного анализа программного и аппаратного обеспечения.
З 2: Основные виды работ на этапе сопровождения программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ОК 10, ПК 4.3	- демонстрирует знание основных видов работ на этапах внедрения и сопровождения программного обеспечения.
З 3: Основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ОК 10, ПК 4.3	- формулирует основные принципы контроля конфигурации программного обеспечение и принципы поддержки целостности конфигурации программного обеспечения.
З4: Средства защиты программного обеспечения в компьютерных системах; ОК 01, ОК 02, ОК 05, ОК 09, ОК 10, ПК 4.4	- определяет основные средства защиты программного обеспечения в компьютерных системах и способы их настройки и применения.

Образец тестового задания

Вариант 1

1. Независимость программных продуктов от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и т.п., означает их ...

- a. модифицируемость
- b. надежность
- c. мобильность
- d. эффективность

2. Бессбойность и устойчивость в работе программных продуктов, точность выполнения предписанных функций обработки, возможность диагностики возникающих в процессе работы программ ошибок, означает их ...

- a. модифицируемость
- b. надежность
- c. мобильность
- d. эффективность

3. Возможность применения алгоритма к целому классу однотипных задач, различающихся конкретным значением, является одним из основных свойств алгоритма ...

- a. детерминированностью
- b. дискретностью
- c. определенностью

- d. массовостью
- 4. Максимально возможная интеграция программных продуктов с другими программами, обеспечение обмена данными в общих форматах представления (экспорт/импорт баз данных, внедрение или связывание объектов обработки и др.) означает их...
 - a. модифицируемость
 - b. эффективность
 - c. мобильность
 - d. коммуникативность
- 5. Способность программных продуктов к внесению изменений, например расширение функций обработки, переход на другую техническую базу обработки и т.п., означает их...
 - a. модифицируемость
 - b. эффективность
 - c. мобильность
 - d. учет человеческого фактора
- 6. Наличие дружественного интерфейса, контекстно-зависимой подсказки, хорошей документации является таким показателем качества программного продукта, как ...
 - a. коммуникативность
 - b. эффективность
 - c. мобильность
 - d. учет человеческого фактора
- 7. В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:
 - постановки задачи,
 - выбора критериев эффективности,
 - проведения предварительных научно-исследовательских работ,
 - разработки технического задания, определяет содержание этапа модели ЖЦ
 - a. Эскизный проект
 - b. Технический проект
 - c. Техническое задание
 - d. Рабочий проект
- 8. В соответствии с ГОСТ 19.102-77 легальное описание действий в виде:
 - структуры входных и выходных данных,
 - уточнения методов решения,
 - общего алгоритма,
 - разработки документации эскизного проекта, определяет содержание этапа модели ЖЦ ...
 - a. Эскизный проект
 - b. Технический проект
 - c. Техническое задание
 - d. Рабочий проект
- 9. В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:
 - уточнения структуры входных и выходных данных,
 - разработки алгоритмов,
 - форм данных,
 - семантики и синтаксиса языка,

- структуры программы,
 - конфигурации технических средств,
 - плана работ, определяет содержание этапа модели ЖЦ ...
- a. Эскизный проект
 - b. Технический проект
 - c. Техническое задание
 - d. Рабочий проект
10. В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:
 - программирования и отладки;
 - разработки документов;
 - подготовки и проведения испытаний;
 - корректировки программы и документов по итогам испытаний, определяет содержание этапа модели ЖЦ...
 - a. Эскизный проект
 - b. Технический проект
 - c. Техническое задание
 - d. Рабочий проект
 11. Наличие обратных связей между этапами жизненного цикла, определяющих возможность проведения проверок и корректировок проектируемого ПС на каждой стадии разработки, является основной особенностью ...
 - a. каскадной модели жизненного цикла
 - b. итерационной модели жизненного цикла
 - c. спиральной модели жизненного цикла ПО
 - d. модели жизненного цикла UML
 12. Основными особенностями:
 - последовательным выполнением входящих в состав ЖЦ этапов,
 - окончанием каждого предыдущего этапа до начала последующего,
 - отсутствием временного перекрытия этапов,
 - отсутствием возврата к предыдущим этапам, - наличием результата только в конце разработки, характеризуется...
 - a. каскадной модели жизненного цикла
 - b. итерационной модели жизненного цикла
 - c. спиральной модели жизненного цикла ПО
 - d. модели жизненного цикла UML
 13. Выявлением и устранением ошибок только на стадии тестирования, которая может растянуться во времени или вообще никогда не завершиться, характеризуется ...
 - a. каскадной модели жизненного цикла
 - b. итерационной модели жизненного цикла
 - c. спиральной модели жизненного цикла ПО
 - d. модели жизненного цикла UML
 14. Функционирование программного продукта в соответствии с техническим заданием соответствует характеристике эксплуатационных требований ...
 - a. правильность
 - b. универсальность
 - c. надежность
 - d. адаптируемость

15. Обеспечение правильной работы при любых допустимых данных и защиты от неправильных данных соответствует характеристике эксплуатационных требований ...
- правильность
 - универсальность
 - надежность
 - защищенность
16. Обеспечение правильности результатов при наличии различного рода сбоев соответствует характеристике эксплуатационных требований ...
- правильность
 - универсальность
 - надежность
 - защищенность
17. Обеспечение погрешности результатов не выше заданной соответствует характеристике эксплуатационных требований...
- правильность
 - проверяемость
 - точность результатов
 - защищенность
18. Использование минимально возможного количества ресурсов технических средств (например, времени микропроцессора, объема оперативной памяти, объема внешней памяти, количества внешних устройств и др.) соответствует характеристике эксплуатационных требований...
- универсальность
 - надежность
 - аппаратная совместимость
 - эффективность
19. Возможность быстрой модификации с целью приспособления к изменяющимся условиям функционирования соответствует характеристике эксплуатационных требований ...
- универсальность
 - эффективность
 - адаптируемость
 - ренерабельность
20. Возможность «параллельного» использования несколькими процессами соответствует характеристике эксплуатационных требований ...
- универсальность
 - программная совместимость
 - повторная входимость
 - ренерабельность
21. Функциональная диаграмма имеет аббревиатуру ...
- ERD
 - SADT
 - DFD
 - STD
22. Диаграмма потоков данных имеет аббревиатуру ...
- ERD

- b. SADT
 - c. DFD
 - d. STD
23. Диаграмма переходов состояний имеет аббревиатуру ...
- a. ERD
 - b. SADT
 - c. DFD
 - d. STD
24. Диаграмма «сущность–связь» имеет аббревиатуру ...
- a. ERD
 - b. SADT
 - c. DFD
 - d. STD
25. Вершиной древовидной структуры диаграмм, представляющей собой самое общее описание системы и ее взаимодействия с внешней средой, является следующий вид диаграммы IDEF0 ...
- a. диаграмма декомпозиции
 - b. диаграмма дерева узлов
 - c. контекстная диаграмма
 - d. диаграмма для экспозиции
26. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Содержит описание функций программного обеспечения с точки зрения пользователя ...
- a. модель использования
 - b. логическая модель
 - c. модель реализации
 - d. модель развертывания
27. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Описывает ключевые понятия моделируемого программного обеспечения (классы, интерфейсы и т. п.), т. е. средства, обеспечивающие его функциональность ...
- a. логическая модель
 - b. модель реализации
 - c. модель процессов
 - d. модель развертывания
28. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Определяет реальную организацию программных модулей в среде разработки ...
- a. модель использования
 - b. модель реализации
 - c. модель процессов
 - d. модель развертывания
29. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Отображает организацию вычислений и позволяет оценить производительность, масштабируемость и надежность программного обеспечения ...

- a. модель использования
 - b. модель реализации
 - c. модель процессов
 - d. модель развертывания
30. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Показывает, каким образом программные компоненты размещаются на конкретном оборудовании ...
- a. модель использования
 - b. модель реализации
 - c. модель процессов
 - d. модель развертывания
31. Служит связующим звеном между процессами, т. е. позволяет описать порядок взаимодействия процессов ...
- a. нотация
 - b. прецедент
 - c. поток событий
 - d. сценарий
32. Подробное процедурное описание вариантов использования системы всеми заинтересованными лицами, а также внешними системами, реализует ...
- a. нотация
 - b. прецедент
 - c. поток событий
 - d. сценарий
33. При обращении к функции в соответствии с требованиями хорошего стиля программирования рекомендуется (но не обязательно), чтобы ...
- a. имена формальных и фактических параметров были различными
 - b. количество формальных параметров соответствовало количеству фактических параметров
 - c. типы формальных параметров соответствовали типам фактических параметров
 - d. порядок перечисления фактических параметров соответствовал порядку перечисления формальных параметров
34. В объектно-ориентированной среде программирования обработчики любых событий для компонентов размещаются в окне ...
- a. Формы
 - b. Инспектора объектов
 - c. Кода
 - d. Деревя объектов
35. В объектно-ориентированной среде программирования к любому компоненту в программе можно обратиться по его свойству ...
- a. Caption
 - b. Text
 - c. Align
 - d. Name
36. Абстрактным называется класс, который ...
- a. не содержит функций
 - b. не содержит переменных

- c. содержит хотя бы одну виртуальную функцию
 - d. содержит хотя бы одну чисто виртуальную функцию
37. Модульное тестирование предполагает ...
- a. тестирование минимально возможного для тестирования компонента, например отдельного класса или функции
 - b. поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
 - c. тестирование интегрированной системы на ее соответствие исходным требованиям
 - d. имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика
38. Интеграционное тестирование предполагает ...
- a. тестирование минимально возможного для тестирования компонента, например отдельного класса или функции
 - b. поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
 - c. тестирование интегрированной системы на ее соответствие исходным требованиям
 - d. распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок
39. Системное тестирование предполагает ...
- a. поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
 - b. тестирование интегрированной системы на ее соответствие исходным требованиям
 - c. имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика
 - d. распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок
40. Альфа-тестирование предполагает ...
- a. поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
 - b. тестирование интегрированной системы на ее соответствие исходным требованиям
 - c. имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика
 - d. распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок
41. Бета-тестирование предполагает ...
- a. поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
 - b. тестирование интегрированной системы на ее соответствие исходным требованиям
 - c. имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика
 - d. распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

42. Каждая команда программы должна быть выполнена хотя бы один раз в соответствии с методом тестирования ...
- a. покрытие операторов;
 - b. покрытие решений;
 - c. покрытие условий;
 - d. покрытие решений/условий;
43. Каждое направление перехода в программе должно быть реализовано, по крайней мере, один раз в соответствии с методом тестирования ...
- a. покрытие операторов;
 - b. покрытие решений;
 - c. покрытие условий;
 - d. покрытие решений/условий;
44. Все возможные результаты каждого условия в решении должны быть выполнены, по крайней мере, один раз в соответствии с методом тестирования ...
- a. покрытие операторов;
 - b. покрытие решений;
 - c. покрытие условий;
 - d. покрытие решений/условий;
45. Видимость переменных и методов класса только в пределах данного модуля определяет спецификатора доступа ...
- a. private
 - b. protected
 - c. public
 - d. published
46. Видимость переменных и методов класса в классах потомков определяет спецификатора доступа ...
- a. private
 - b. protected
 - c. public
 - d. published
47. Видимость переменных и методов класса в любом месте программы определяет спецификатора доступа ...
- a. private
 - b. protected
 - c. public
 - d. published
48. Процесс, в ходе которого один объект может приобретать свойства другого, называется...
- a. наследование
 - b. инкапсуляция
 - c. полиморфизм
 - d. динамический полиморфизм
49. Процесс перегрузки функций и операторов реализует принцип ...
- a. наследования
 - b. инкапсуляции
 - c. статического полиморфизма

- d. динамического полиморфизма
- 50. На основе наследования и виртуальных функций реализуется принцип ...
 - a. наследования
 - b. инкапсуляции
 - c. статического полиморфизма
 - d. динамического полиморфизма

Практико-ориентированное задание

Вариант 1

Пользователь вводит набор целых чисел. Найти произведение нечетных элементов в указанном интервале и вывести его. Если таких элементов нет, то вывести сообщение об этом.

Требования к заданию:

1. Обязателен графический интерфейс.
2. 3 тестовых набора, описывающих разные ситуации.
3. Переменные и функции должны быть поименованы в соответствии с выполняемыми задачами.
4. Наличие комментариев к каждому блоку кода.
5. Презентация работы программы.

4. Эталоны ответов

Эталоны ответов на вопросы тестового задания

1) A; 2) B; 3) D; 4) D; 5) C; 6) D; 7) A; 8) B; 9) D; 10) D; 11) B; 12) A; 13) A; 14) C; 15) C; 16) D; 17) C; 18) D; 19) C; 20) C; 21) B; 22) C; 23) D; 24) A; 25) C; 26) A; 27) C; 28) B; 29) D; 30) B; 31) C; 32) D; 33) D; 34) B; 35) D; 36) A; 37) B; 38) C; 39) C; 40) C; 41) C; 42) D; 43) C; 44) B; 45) A; 46) B; 47) C; 48) A; 49) C; 50) D.

Эталон ответа на практико-ориентированное задание

```
from tkinter import *
from tkinter import ttk
# Импортируем модуль для создания интерфейса

# модуль расчета произведения нечетных чисел
def show_message():
    numbers = entry.get()
    list_number = [int(_) for _ in numbers.split()]
    array = []
    for number in list_number:
        if number % 2 != 0:
            array.append(number)
    result = 1
    for number in array:
        result *= number
    if result == 1:
        result = 'Нечетных элементов нет!'
    label["text"] = result # получаем введенный текст

# модуль создания и обработки интерфейса
```

```
root = Tk()
root.title('Произведение')
root.geometry("250x200")
entry = ttk.Entry()
entry.pack(anchor=NW, padx=9, pady=9)
btn = ttk.Button(text="Расчет", command=show_message)
btn.pack(anchor=NW, padx=9, pady=9)
label = ttk.Label()
label.pack(anchor=NW, padx=9, pady=9)
root.mainloop()
```

5. Зачётная ведомость.

3. Курсовой проект

1. Форма проведения: (устная)

2. Условия выполнения

Время выполнения задания: 180 мин.

Оборудование учебного кабинета: посадочные места по количеству обучающихся.

Технические средства обучения: телевизор, компьютер.

Информационные источники:

1. Программирование на языке высокого уровня Python: учеб. пособие для СПО / Д. Ю. Федоров. – М.: Издательство Юрайт, 2019. – 126 с. – (Серия: Профессиональное образование).
2. Основы программирования на Python: учебное пособие для вузов / С. А. Чернышев. – Москва: Издательство Юрайт, 2022. – 286с. – (Высшее образование). – Текст: непосредственный.
3. Программирование. Основы Python: учебное пособие для СПО / Т. П. Никитина, Л. В. Королев. – Санкт-Петербург: Лань, 2023. – 156 с.: ил. – Текст: непосредственный.
4. Языки программирования. Python: решение сложных задач: учебное пособие для СПО / С. В. Борзунов, С.Д. Кургалин. – Санкт-Петербург: Лань, 2023. – 192 с.: ил. – Текст: непосредственный.
5. Основы программирования: учебник для СПО / Т. Ю. Шеина. – 2-е изд., стер. – Санкт-Петербург: Лань, 2023. – 292 с.: ил. – Текст: непосредственный.
6. Программирование в Python 3: учебное пособие / Д. В. Полупанов, С. Р. Абдюшева, А. М. Ефимов. – Уфа: РИЦ БашГУ, 2020. – 164 с.

Требования охраны труда: Обучающимся запрещается: находиться в аудитории в верхней одежде; иметь при себе средства связи; разговаривать между собой; обмениваться любыми материалами и предметами с другими участниками защиты; произвольно выходить из аудитории.

3. Пакет материалов для проведения защиты курсового проекта:

3.1 Примерный перечень курсовых работ:

1. Разработка программного комплекса для обработки сведений по учету программно-аппаратного обеспечения предприятия.
2. Разработка программного комплекса для обработки сведений по учету работы печатного издательства.

3. Разработка программного комплекса для обработки сведений по работе медицинского учреждения.
4. Разработка программного комплекса для обработки сведений по учету работы отдела кадров предприятия.
5. Разработка программного комплекса для обработки сведений по учету работы спортивного клуба.
6. Разработка информационно-справочной системы для обработки сведений по отпуску льготных лекарств в аптеке.
7. Разработка программного комплекса для обработки сведений по учету аудиторного фонда колледжа.
8. Разработка программного комплекса для обработки сведений по восхождениям в альпинистском клубе.
9. Разработка программного комплекса для обработки сведений по учету самолетных рейсов.
10. Разработка программного комплекса для обработки сведений по учету работы оптового склада.
11. Разработка приложения для покупки, аренды и обслуживания автомобиля.
12. Разработка программного комплекса для обработки сведений по учету работы библиотеки.
13. Разработка программного комплекса для обработки сведений по учету работы страховой компании.
14. Разработка приложения учета виниловых пластинок в частной коллекции.
15. Разработка системы тестирования знаний студентов по дисциплине технологии программирования.
16. Разработка системы анализа спортивных достижений в клубных матчах по футболу.
17. Разработка программного комплекса электронного автомобильного музея.
18. Разработка информационно-справочной системы расписания поездов.
19. Разработка информационной системы станции технического обслуживания автомобилей.
20. Разработка справочной информационной системы книжного магазина.
21. Разработка информационно-справочной системы "Контроль локальной сети предприятия".
22. Разработка информационно-справочной системы "Железнодорожные билеты".
23. Разработка программы автоматизации учета книг в домашней библиотеке.
24. Разработка приложения учета продаж железнодорожных билетов.
25. Разработка программного комплекса для турагентства с использованием базы данных.
26. Разработка приложения ведения счетов по пластиковым картам.
27. Разработка программного комплекса электронного вызова сотрудников технической поддержки.
28. Разработка программного комплекса учёта физических лиц в Налоговой инспекции.
29. Разработка приложения автоматизации рабочего места администратора гостиницы.

30. Разработка программного комплекса подбора и расчета комплектующих компьютера.
31. Разработка программного комплекса для общения по локальной сети с выделенными сервером.
32. Разработка электронного словаря с возможностью подключения к базе данных.
33. Разработка программы-тренажера работы на клавиатуре.
34. Разработка текстового редактора.
35. Разработка графического редактора (возможности: рисование линий, окружностей, квадратов)
36. Разработка программы-переводчика с английского языка на русский с обращением к базе данных слов.
37. Разработка программы для сдачи теоретической части экзамена в ГИБДД
38. Разработка программы для регистрации нового пользователя и хранения его персональных данных в отдельном файле таблиц.
39. Разработка программы для обучения переводу чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную и обратно.
40. Сравнение производительности различных методов сортировки и поиска в Python.
41. Создание игры-головоломки на Python.
42. Исследование и разработка методов оптимизации кода на Python.
43. Разработка программы для автоматического обновления программного обеспечения на Python с использованием GitHub и библиотеки requests.
44. Разработка программного комплекса учёта компьютерного и сетевого оборудования организации с использованием фреймворка Django.
45. Разработка приложения с графическим интерфейсом для управления мультимедиа.
46. Разработка программы мониторинга и анализа сетевой активности на Python с использованием библиотеки Scapy.
47. Разработка программы мониторинга системных ресурсов на Python с использованием библиотеки psutil.
48. Подбор, анализ и сравнение методов эффективного использования памяти на Python.
49. Создание веб-приложения с использованием NodeJS для управления библиотекой.
50. Анализ и сравнительное исследование популярных паттернов проектирования в разработке модулей ПО.
51. Разработка программного обеспечения с использованием контейнеризации (например, Docker) на Python.
52. Создание чат-бота для поддержки клиентов на Python с использованием машинного обучения.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

МДК 01.02 Поддержка и тестирование программных модулей

Подготовки специалистов среднего звена по специальности

код специальности 09.02.07 Информационные системы и программирование

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности (профессии) СПО 09.02.07 Информационные системы и программирование базовой подготовки программы МДК 01.02 Поддержка и тестирование программных модулей.

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчик: Бахаев Д. В., преподаватель,
Голубенко М. И., преподаватель.

ОДОБРЕН
цикловой методической комиссией
Протокол № 1
от «3» сентября 2022 г.
Председатель Гаврикова Е. Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения МДК, подлежащие проверке
3. Оценка освоения МДК
 - 3.1. Контроль и оценка освоения МДК
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет материалов
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения МДК 01.02 Поддержка и тестирование программных модулей обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 Информационные системы и программирование базового уровня следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями

Обучающийся должен знать:

З 1	- основные этапы разработки программного обеспечения;
З 2	- основные принципы технологии структурного и объектно-ориентированного программирования;
З 3	- способы оптимизации и приемы рефакторинга;
З 4	- основные принципы отладки и тестирования программных продуктов.

Обучающийся должен уметь:

У 1	- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
У 2	- создавать программу по разработанному алгоритму как отдельный модуль;
У 3	- выполнять отладку и тестирование программы на уровне модуля;
У 4	- осуществлять разработку кода программного модуля на современных языках программирования;
У 5	- уметь выполнять оптимизацию и рефакторинг программного кода;
У 6	- оформлять документацию на программные средства.

Обучающийся должен иметь практический опыт:

- в разработке кода программного продукта на основе готовой спецификации на уровне модуля;
- в использовании инструментальных средств на этапе отладки программного продукта;
- в проведении тестирования программного модуля по определенному сценарию;
- в использовании инструментальных средств на этапе отладки программного продукта;
- в разработке мобильных приложений.

Формируемые ОК:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

Формируемые ПК:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ.

Личностные результаты учебной дисциплины

Код ЛР	Личностные результаты реализации программы (дескрипторы)
ЛР 1	-осознающий себя гражданином и защитником великой страны
ЛР 2	-проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	-демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих
ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личностного и профессионального, конструктивного «цифрового следа»

ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	- проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	- уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 10	- принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР 17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
ЛР 19	- профессиональная идентичность и ответственность
ЛР 20	- самооценка и рефлексия результатов своей деятельности и развития

Формами промежуточной аттестации по МДК являются контрольная работа – 5 семестр, экзамен – 6 семестр.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по МДК осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих и профессиональных компетенций:

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней; ОК 1, 2, 3, 5, 9, 10; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля.
У 2. Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 3, 5, 9, 10; ПК 1.1, 1.2.	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма.
У 3. Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 3, 5, 9, 10; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля.
У 4. Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 3, 5, 9, 10; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.
У 5. Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 3, 5, 9, 10; ПК 1.4, 1.5.	Производит оптимизацию программного кода и его рефакторинг.
У 6. Оформлять документацию на программные средства. ОК 1, 2, 3, 4, 5, 6, 7, 9, 10; 11 ПК 1.2.	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства.
Знать:	
З 1. Основные этапы разработки программного обеспечения ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.1, 1.2.	Знает основные процессы жизненного цикла программного обеспечения; на основе процессов понимает последовательность разработки программного обеспечения.
З 2. Основные принципы технологии структурного и объектно-ориентированного программирования; ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.1, 1.2.	Понимает принципиальные различия в технологиях структурного и объектно-ориентированного программирования; определяет на основе какой технологии построен код программы;
З 3. Способы оптимизации и приемы рефакторинга; ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.3, 1.4, 1.5.	Знает основные способы оптимизации и рефакторинга кода; применяет методы к написанному коду; составляет код программы на основе методов.
З 4. Основные принципы отладки и тестирования программных продуктов. ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.3, 1.4.	Понимает последовательность действий при отладке программного продукта; осознает принципы и методы тестирования; создает тестовые кейсы на основе различных методов тестирования.

3.ОЦЕНКА ОСВОЕНИЯ МДК

3.1. Контроль и оценка освоения МДК по темам (разделам)

Элемент МДК				
	Текущий контроль		Промежуточная аттестация	
	Осваиваемые результаты	Метод контроля	Проверяемые результаты	Форма контроля
Раздел 1. Разработка программных модулей				
Тема 1.1. Жизненный цикл ПО	У 1-6 З 1-4 ПК 1.1 ПК 1.2 ОК 1-ОК 10	Устный опрос	ОК 1, 2, 7, 9. ПК 1.1 – 1.6 У 1 -6 З 1 - 4	5-й семестр – контрольная работа, 6 семестр – экзамен.
Тема 1.2 Структурное программирование	У 1-6 З 1-4 ПК 1.1 ПК 1.2 ОК 1-ОК 9	Устный опрос Самостоятельная работа Практическая работа Тестовое задание		
Тема 1.3 Объектно- ориентированное программирование	У 1-6 З 1-4 ПК 1.1 ПК 1.2 ОК 1-ОК 9	Устный опрос Самостоятельная работа Практическая работа Тестовое задание		
Тема 1.4 Паттерны программирования	У 1-6 З 1-4 ПК 1.1 ПК 1.2 ОК 1-ОК 9	Устный опрос Самостоятельная работа Практическая работа		
Тема 1.5 Событийно- управляемое программирование	У 1-6 З 1-4 ПК 1.1 ПК 1.2 ОК 1-ОК 9	Устный опрос Самостоятельная работа Практическая работа Тестовое задание		
Тема 1.6 Оптимизация и рефакторинг кода	У 1-6 З 1-4 ПК 1.4 ПК 1.5 ОК 1-ОК 9	Устный опрос Самостоятельная работа Практическая работа		
Тема 1.7 Разработка пользовательского интерфейса	У 1-6 З 1-4 ПК 1.2 ОК 1-ОК 9	Устный опрос Самостоятельная работа		

		Практическая работа Тестовое задание		
Тема 1.8 Основы ADO.Net	У 1-6 З 1-4 ПК 1.1 ПК 1.2 ОК 1-ОК 9	Устный опрос Самостоятельная работа Практическая работа Тестовое задание		

3.1.1. Методы и критерии оценивания

1. Устный опрос. Критерии оценивания.

Оценка 5 «отлично» - ответил на вопросы в объеме лекционного и дополнительного материала, дал полные грамотные ответы на все дополнительные вопросы.

Оценка 4 «хорошо» - грамотно изложил ответы на вопросы, но содержание и формулировки имеют отдельные неточности (допускается нечеткая формулировка определений), в полной мере ответил на заданные дополнительные вопросы.

Оценка 3 «удовлетворительно» - ответил на часть вопросов в объеме лекционного материала и ответил на часть дополнительных вопросов.

Оценка 2 «неудовлетворительно» - допустил ошибки в определении базовых понятий, искажил их смысл, не ответил на дополнительные вопросы.

2. Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

3. Самостоятельная работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме; учтены все требования к данной работе; самостоятельно поставлены цели и задачи работы, соответствующие заданной теме/проблеме; получены результаты в соответствии с поставленной целью; работа оформлена аккуратно и грамотно.

Оценка 4 «хорошо» - выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью, и объем выполненной части работы позволяет получить недостаточно результатов в соответствии с поставленной целью.

4. Лабораторная работа. Критерии оценивания.

Выполнение работы в полном объеме с соблюдением необходимой последовательности проведения опытов, измерений – 2 балла;

Рациональный и самостоятельный выбор и подготовка необходимого оборудования для выполнения работ, обеспечивающих получение точных результатов – 2 балла;

Описание хода лабораторной работы в логической последовательности – 1 балл;

Корректная формулировка выводов по результатам лабораторной работы – 2 балла;
Выполнения всех записей, таблиц, рисунков, чертежей, графиков, вычислений в соответствии с заданием, технически грамотно и аккуратно – 2 балла;
Соблюдение правил техники безопасности при выполнении лабораторной работы – 1 балл
Перевод баллов в отметку:

Оценка 5 «отлично» - от 9 до 10 баллов

Оценка 4 «хорошо» - от 6 до 8 баллов.

Оценка 3 «удовлетворительно» - от 3 до 5 баллов.

Оценка 2 «неудовлетворительно» - от 1 до 2 баллов.

5. Практическая работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме, с соблюдением алгоритма выполнения: последовательности проведения измерений, заполнения таблиц, графиков и др.; правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; получены результаты в соответствии с поставленной целью.

Оценка 4 «хорошо»- выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью и объем выполненной части работы не позволяет получить никаких результатов в соответствии с поставленной целью.

4. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Контрольная работа

1.Форма проведения: письменная (тестирование).

2.Условия выполнения:

1.Инструкция для обучающихся:

2.Время выполнения: 90 минут на тестовое задание.

3.Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.

4.Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда программирования PyCharm с установленным интерпретатором Python не ниже версии 3.8; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

5.Информационные источники, допустимые к использованию на экзамене: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».

6.Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения промежуточной аттестации

3.1. Перечень тем, выносимых на контрольную работу

1. Отладка и тестирование программного обеспечения:

- тестирование как часть процесса верификации программного обеспечения;
- виды ошибок. методы отладки;
- тестирование «белым ящиком»;

- тестирование «черным ящиком»;
- методы тестирования;
- классификация тестирования по уровням;
- модульное тестирование;
- интеграционное тестирование;
- тестирование производительности;
- регрессионное тестирование;
- составление собственных сценариев тестирования для программных модулей;
- составление документации для разных сценариев тестирования.

2. Документирование:

- средства разработки технической документации;
- технологии разработки документов;
- разработка технической документации для программных модулей;
- оформление документации на программные средства с использованием инструментальных средств;
- документирование программного обеспечения в соответствии с единой системой программной документации;
- разработка технической документации в соответствии с еспд для программных модулей;
- автоматизация разработки технической документации;
- автоматизированные средства оформления документации;
- автоматизация разработки технической документации для различных программных модулей средствами языка программирования;
- встроенные возможности документирования у языков программирования.

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней; ОК 1, 2, 7, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля.
У 2. Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 7, 9; ПК 1.1, 1.2.	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма.
У 3. Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 7, 9; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля.
У 4. Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 7, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.

У 5. Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 7, 9; ПК 1.4, 1.5.	Производит оптимизацию программного кода и его рефакторинг.
У 6. Оформлять документацию на программные средства. ОК 1, 2, 7, 9; ПК 1.2.	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства.
Знать:	
З 1. Основные этапы разработки программного обеспечения ОК 1, 2, 7, 9; ПК 1.1, 1.2.	Знает основные процессы жизненного цикла программного обеспечения; на основе процессов понимает последовательность разработки программного обеспечения.
З 2. Основные принципы технологии структурного и объектно-ориентированного программирования; ОК 1, 2, 7, 9; ПК 1.1, 1.2.	Понимает принципиальные различия в технологиях структурного и объектно-ориентированного программирования; определяет на основе какой технологии построен код программы;
З 3. Способы оптимизации и приемы рефакторинга; ОК 1, 2, 7, 9; ПК 1.3, 1.4, 1.5.	Знает основные способы оптимизации и рефакторинга кода; применяет методы к написанному коду; составляет код программы на основе методов.
З 4. Основные принципы отладки и тестирования программных продуктов. ОК 1, 2, 7, 9; ПК 1.3, 1.4.	Понимает последовательность действий при отладке программного продукта; осознает принципы и методы тестирования; создает тестовые кейсы на основе различных методов тестирования.

Образец тестового задания

1. Что такое отладка в Python?
2. Какие инструменты Python можно использовать для отладки кода?
3. Что такое модульное тестирование?
4. Как написать модульный тест в Python?
5. Что такое интеграционное тестирование?
6. Что такое функциональное тестирование?
7. Что такое регрессионное тестирование?
8. Какие виды тестирования не покрывают все возможные сценарии использования программы?
9. Что такое система контроля версий и зачем она нужна при разработке ПО?
10. Что такое документирование кода и зачем оно нужно?
11. Какие инструменты существуют для документирования кода на Python?
12. Что такое стиль PEP8 и почему его важно соблюдать?
13. Как правильно оформить комментарии в коде на Python?
14. Что такое docstring и как его использовать?
15. Зачем нужно документировать функции и методы?
16. В чем отличие документации к функциям и методам?

17. Какие форматы файлов используются для документирования в Python?
18. Что такое метаданные модуля и как их использовать?
19. Как организовать систему документирования для большого проекта?
20. Что такое PEP и зачем он нужен?

4.Эталоны ответов

1. Отладка – это процесс поиска и устранения ошибок в программе.
2. Для отладки кода в Python можно использовать такие инструменты, как pdb, ipdb, traceback, logging.
3. Модульное тестирование – это тестирование отдельных модулей или функций программы.
4. Для написания модульного теста в Python можно использовать библиотеку unittest.
5. Интеграционное тестирование – это тестирование взаимодействия между различными модулями программы.
6. Функциональное тестирование – это тестирование соответствия программы ее функциональному назначению.
7. Регрессионное тестирование – это тестирование, которое проводится после внесения изменений в программу, чтобы убедиться, что эти изменения не привели к новым ошибкам.
8. Ни один из видов тестирования не может покрыть все возможные сценарии использования программы.
9. Система контроля версий – это инструмент для хранения разных версий кода и управления ими. Она нужна для того, чтобы разработчики могли работать над проектом параллельно и не мешать друг другу.
10. Документирование кода – это написание комментариев к коду, которые объясняют его работу. Оно нужно для того, чтобы другие разработчики могли понять, как работает код.
11. Для документирования кода на Python можно использовать инструменты reStructuredText, Sphinx, Epydoc.
12. PEP8 – это стандарт оформления кода на языке Python. Он важен для того, чтобы код был читаемым и понятным для других разработчиков.
13. Комментарии в коде на Python оформляются с помощью символов #.
14. Docstring – это строка документации, которая идет вместе с определением функции или класса. Она используется для описания того, что делает функция или класс.
15. Документация функций и методов нужна для того, чтобы другие разработчики знали, что они делают и как их использовать.
16. Документация к функциям обычно описывает, что функция делает и какие аргументы принимает, а документация к методам описывает, что метод делает и какие параметры принимает.
17. Файлы документации в Python обычно имеют расширение .py или .rst.
18. Метаданные модуля – это информация о модуле, такая как автор, версия, описание. Они используются для того, чтобы другие разработчики могли узнать больше о модуле.
19. Для организации системы документирования большого проекта можно использовать системы управления документами, такие как Confluence или Jira.
20. PEP – это Python Enhancement Proposal, предложение по улучшению языка Python. Он нужен для того, чтобы стандартизировать язык и сделать его более понятным для разработчиков.

Критерии оценки ответов

Оценка 5 «отлично» - на 85% и более вопросов дан верный ответ.

Оценка 4 «хорошо» - на 75% и более вопросов дан верный ответ.

Оценка 3 «удовлетворительно» - на 55% и более вопросов дан верный ответ.

Оценка 2 «неудовлетворительно» - правильные ответы даны ниже, чем на 55% вопросов.

2. Экзамен

1. Форма проведения: в форме билетов.

2. Условия выполнения:

1. Инструкция для обучающихся:

2. Время выполнения: 90 минут на тестовое задание, 30 – на практическую работу.

3. Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.

4. Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда программирования PyCharm с установленным интерпретатором Python не ниже версии 3.8; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

5. Информационные источники, допустимые к использованию на экзамене: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».

6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения экзамена

3.1. Перечень тем, выносимых на экзамен:

1. Отладка и тестирование программного обеспечения:

- тестирование как часть процесса верификации программного обеспечения;
- виды ошибок. методы отладки;
- тестирование «белым ящиком»;
- тестирование «черным ящиком»;
- методы тестирования;
- классификация тестирования по уровням;
- модульное тестирование;
- интеграционное тестирование;
- тестирование производительности;
- регрессионное тестирование;
- составление собственных сценариев тестирования для программных модулей;
- составление документации для разных сценариев тестирования.

2. Документирование:

- средства разработки технической документации;
- технологии разработки документов;
- разработка технической документации для программных модулей;
- оформление документации на программные средства с использованием инструментальных средств;
- документирование программного обеспечения в соответствии с единой системой программной документации;
- разработка технической документации в соответствии с еспд для программных

модулей;

- автоматизация разработки технической документации;
- автоматизированные средства оформления документации;
- автоматизация разработки технической документации для различных программных модулей средствами языка программирования;
- встроенные возможности документирования у языков программирования.

3.2. Перечень вопросов, выносимых на экзамен

1. Что такое интерпретатор Python?
2. Какой тип данных в Python является базовым?
3. Как создать переменную в Python?
4. Какая операция не определена для строки в Python?
5. Какие типы данных могут быть преобразованы в числа в Python?
6. В чем разница между списками и кортежами в Python?
7. Какие виды циклов существуют в Python?
8. Что возвращает функция `max` в Python?
9. Что означает ключевое слово `“global”` в Python?
10. В каком порядке выполняются операции в арифметических выражениях в Python?
11. Какие операторы сравнения существуют в Python?
12. Что делает функция `print` в Python?
13. Для чего используется функция `input` в Python?
14. Как определить, является ли число четным или нечетным в Python?
15. Какую роль играют отступы в Python?
16. Что произойдет, если в операторе `if` не поставить двоеточие после условия?
17. Можно ли использовать вложенные циклы в Python?
18. Как работает оператор `break` в Python?
19. Как выполнить несколько команд в одной строке в Python?
20. Что будет, если не закрыть круглые скобки в операторе `print` в Python?
21. В каких случаях используется оператор `%` в Python?
22. Как вставить символ в строку в Python?
23. Что происходит, если функция не возвращает значение в Python?
24. Что делает оператор `in` в Python?
25. Зачем нужны исключения в Python?
26. Как использовать `try...except` в Python?
27. Что делает метод `sorted` в Python?
28. Зачем нужен метод `len` в Python?
29. Для чего нужен метод `split` в Python?
30. Почему Python называют объектно-ориентированным языком программирования?

Эталоны ответов на вопросы

1. Интерпретатор Python - это программа, которая выполняет код на языке Python. Он преобразует исходный код в машинный код и выполняет его.
2. В Python базовым типом данных является целое число (`int`).
3. Чтобы создать переменную в Python, нужно присвоить ей значение. Например: `variable = 10`.
4. Операция деления на строку не определена в Python.
5. В Python числами могут быть преобразованы следующие типы данных: целые числа (`int`), вещественные числа (`float`), комплексные числа (`complex`) и кортежи (`tuple`), содержащие числа.

6. Списки и кортежи в Python - это структуры данных, которые используются для хранения групп элементов. Основное отличие между ними заключается в том, что списки могут изменяться (добавлять, удалять или изменять элементы), в то время как кортежи являются неизменяемыми структурами.
7. В Python существуют следующие виды циклов: цикл for, цикл while и цикл nested (вложенный цикл).
8. Функция max в Python возвращает наибольшее значение из списка, кортежа или строки.
9. Ключевое слово "global" в Python используется для объявления глобальных переменных. Оно позволяет изменять значение переменной, объявленной в глобальной области видимости, внутри функции.
10. В Python арифметические операции выполняются в следующем порядке: сначала выполняются операции умножения и деления, затем - сложения и вычитания.
11. В Python существуют следующие операторы сравнения: == (равно), != (не равно), < (меньше), > (больше), <= (меньше или равно), >= (больше или равно).
12. Функция print в Python используется для вывода информации на экран.
13. Функция input в Python используется для получения ввода от пользователя.
14. Чтобы определить, является ли число четным или нечетным в Python, можно использовать оператор modulo (%). Если остаток от деления числа на 2 равен 0, то число является четным, в противном случае - нечетным.
15. Отступы в Python используются для обозначения блоков кода. Они помогают улучшить читаемость и структурированность кода.
16. Если в операторе if не поставить двоеточие после условия, то Python интерпретирует следующий код как часть условия, что может привести к ошибкам.
17. В Python можно использовать вложенные циклы. Например, цикл for внутри цикла while.
18. Оператор break в Python используется для прерывания цикла. Он останавливает выполнение цикла и переходит к следующей строке кода.
19. Чтобы выполнить несколько команд в одной строке в Python, можно использовать символ ";".
20. Если не закрыть круглые скобки в операторе print в Python, то Python интерпретирует все последующие символы как часть выражения, что может привести к ошибкам.
21. Оператор % в Python используется для операции деления по модулю (modulo).
22. Чтобы вставить символ в строку в Python, можно использовать оператор "+". Например: "Привет" + " " + "мир".
23. Если функция не возвращает значение в Python, то она возвращает None.
24. Оператор in в Python используется для проверки наличия элемента в списке, кортеже или строке.
25. Исключения в Python используются для обработки ошибок и нестандартных ситуаций в программе.
26. Для использования try...except в Python нужно поместить код, который может вызвать исключение, в блок try, а затем обработать исключение в блоке except.
27. Метод sorted в Python используется для сортировки элементов списка, кортежа или строки.
28. Метод len в Python используется для получения длины списка, кортежа или строки.
29. Метод split в Python используется для разделения строки на подстроки по определенному символу или подстроке.
30. Python называют объектно-ориентированным языком программирования, потому что он предоставляет возможности для создания объектов с определенными свойствами и методами,

а также для наследования этих свойств и методов от других объектов.

3.3. Практические задания к экзамену:

Напишите программу на языке Python, которая будет имитировать работу системы управления базой данных. Программа должна содержать следующие функции:

1. Создание новой базы данных: программа должна предоставлять пользователю возможность создать новую базу данных с заданным именем.
2. Добавление новой записи в базу данных: программа должна предоставлять пользователю возможность добавить новую запись в базу данных, содержащую псевдоним, полное имя и пароль.
3. Поиск записи в базе данных: программа должна предоставлять пользователю возможность найти запись в базе данных по псевдониму, полному имени и идентификатору.
4. Обновление записи в базе данных: программа должна предоставлять пользователю возможность обновить существующую запись в базе данных, изменяя псевдоним, полное имя и пароль.
5. Удаление записи из базы данных: программа должна предоставлять пользователю возможность удалить существующую запись из базы данных.
6. Вывод всех записей из базы данных: программа должна предоставлять пользователю возможность вывести все записи из базы данных.
7. Создание отчетов: программа должна предоставлять пользователю возможность создавать отчеты на основе данных в базе данных.
8. Обработка ошибок: программа должна предоставлять пользователю возможность обрабатывать ошибки, возникающие при работе с базой данных.
9. Закрытие базы данных: программа должна предоставлять пользователю возможность закрыть базу данных после окончания работы с ней.

Эталоны ответов на практические задания

```
AbstartUser.py
from abc import ABC, abstractmethod

class AbstractUser(ABC):

    @property
    def id(self) -> float:
        return self.id

    @property
    def username(self) -> str:
        return self.username

    @username.setter
    @abstractmethod
    def username(self, new_username: str) -> None:
        pass
```

```
@property
def password(self) -> str:
    return self.password

@password.setter
@abstractmethod
def password(self, new_password: str) -> None:
    pass

@property
def fullname(self) -> str:
    return self.fullname

@fullname.setter
@abstractmethod
def fullname(self, new_fullname: str) -> None:
    pass
```

User.py

```
from random import random
from ..abstracts.AbstractUser import AbstractUser
from ..utils import encoder, decoder
```

```
class User(AbstractUser):
```

```
    def __init__(self, username: str, password: str) -> None:
        self._id = self._get_next_id()
        self._username = username
        self._password = encoder(password)
        self._fullname = None
```

```
    @staticmethod
    def _get_next_id() -> float:
        return random()
```

```
    @property
    def id(self) -> float:
        return self._id
```

```
    @property
    def username(self) -> str:
        return self._username
```

```
    @username.setter
    def username(self, new_username: str) -> None:
```

```

self._username = new_username

@property
def password(self) -> str:
    return ".join(self._password)

@password.setter
def password(self, new_password: str) -> None:
    repeat_password = input("[*] Введите пароль, что выполнить изменения: ")
    decode_password = ".join(decoder(self._password))
    if repeat_password == decode_password:
        self._password = encoder(new_password)
    else:
        print("[!] Пароли не совпадают.")

@property
def fullname(self) -> str:
    return self._fullname

@fullname.setter
def fullname(self, new_fullname) -> None:
    if self._fullname is None:
        self._fullname = new_fullname
    else:
        print("[!] Имя уже задано.")

def convert(self) -> dict:
    properties: dict = self.__dict__
    correct_properties: dict = {}

    for key, value in properties.items():
        correct_properties[key[1:]] = value

    return correct_properties

def __repr__(self) -> str:
    properties = self.convert()
    template: str = r'User({0})'
    string_properties: list[str] = []

    for key, value in properties.items():
        string_properties.append(f'{key}={value}')

    string_properties: str = ', '.join(string_properties)

```

```
return template.format(string_properties)
```

Controller.py

```
from pymongo import MongoClient
from pymongo.cursor import Cursor
```

```
class Controller:
```

```
    def __init__(self,
                 name: str,
                 collection: str) -> None:
        self._client = MongoClient('localhost', 8001)
        self._db = self._client[name]
        self._collection = self._db[collection]

    def insert(self, document: dict) -> None:
        self._collection.insert_one(document)

    def insert_all(self, documents: list[dict]) -> None:
        self._collection.insert_many(documents)

    def find(self, query: dict = None) -> list:
        response: Cursor = self._collection.find(query or {})

        return list(response)

    def update(self,
              document: dict,
              new_document: dict,
              many: bool = False,
              upsert: bool = False) -> None:
        if many:
            self._collection.update_many(document,
                                         {'$set': new_document},
                                         upsert=upsert)
        else:
            self._collection.update_one(document,
                                         {'$set': new_document},
                                         upsert=upsert)

    def delete(self, document: dict, many: bool = False) -> None:
        if many:
            self._collection.delete_many(document)
        else:
            self._collection.delete_one(document)
```

utils.py

```
from env import KEY
```

```
def encoder(password: str) -> list[str]:
```

```
    password_length: int = len(password)
```

```
    key_length: int = len(KEY)
```

```
    max_value: int = max(password_length, key_length)
```

```
    values: list[str] = []
```

```
    for i in range(max_value):
```

```
        if len(values) >= password_length:
```

```
            break
```

```
        xor_value = ord(password[i % password_length]) ^ ord(KEY[i % key_length])
```

```
        values.append(hex(xor_value)[2:])
```

```
    return values
```

```
def decoder(password: list[str]) -> list[str]:
```

```
    password_length: int = len(password)
```

```
    key_length: int = len(KEY)
```

```
    max_value: int = max(password_length, key_length)
```

```
    values: list[str] = []
```

```
    for i in range(max_value):
```

```
        if len(values) >= password_length:
```

```
            break
```

```
        xor_value = int(password[i % password_length], base=16) ^ ord(KEY[i % key_length])
```

```
        values.append(hex(xor_value)[2:])
```

```
    values = [chr(int(i, base=16)) for i in values]
```

```
    return values
```

```
def document_to_string(document: dict) -> str:
```

```
    string: str = ""
```

```
    for key, value in document.items():
```

```
        string += f'{key}: {value}\n'
```

```
    return string
```

```

env.py
KEY = 'lorem ipsum dolor sit amet, consectetur adipiscing'

main.py
from bson import ObjectId
from RtD.controllers.Controller import Controller
from RtD.utils import document_to_string
from RtD.objects.User import User
from pprint import pprint

if __name__ == '__main__':
    user_controller: Controller = Controller(
        name='db',
        collection='users',
    )

    pprint(user_controller)

    for user in user_controller.find():
        pprint(user)

```

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней; ОК 1, 2, 7, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля.
У 2. Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 7, 9; ПК 1.1, 1.2.	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма.
У 3. Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 7, 9; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля.
У 4. Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 7, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.
У 5. Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 7, 9; ПК 1.4, 1.5.	Производит оптимизацию программного кода и его рефакторинг.

У 6. Оформлять документацию на программные средства. ОК 1, 2, 7, 9; ПК 1.2.	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства.
Знать:	
З 1. Основные этапы разработки программного обеспечения ОК 1, 2, 7, 9; ПК 1.1, 1.2.	Знает основные процессы жизненного цикла программного обеспечения; на основе процессов понимает последовательность разработки программного обеспечения.
З 2. Основные принципы технологии структурного и объектно-ориентированного программирования; ОК 1, 2, 7, 9; ПК 1.1, 1.2.	Понимает принципиальные различия в технологиях структурного и объектно-ориентированного программирования; определяет на основе какой технологии построен код программы;
З 3. Способы оптимизации и приемы рефакторинга; ОК 1, 2, 7, 9; ПК 1.3, 1.4, 1.5.	Знает основные способы оптимизации и рефакторинга кода; применяет методы к написанному коду; составляет код программы на основе методов.
З 4. Основные принципы отладки и тестирования программных продуктов. ОК 1, 2, 7, 9; ПК 1.3, 1.4.	Понимает последовательность действий при отладке программного продукта; осознает принципы и методы тестирования; создает тестовые кейсы на основе различных методов тестирования.

Образец экзаменационного билета

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ краевое государственное автономное профессиональное образовательное учреждение «Дальнегорский индустриально-технологический колледж»		
Утверждаю Заместитель директора _____ (Ф.И.О.) _____ (подпись) « ____ » _____ 20__ г.	Экзаменационный билет №1 по ПМ (индекс, название) Группа (ы) _____ Специальность: код, название	Рассмотрено на заседании цикловой методической комиссии Председатель ____ (Ф.И.О.) _____ (подпись) « ____ » _____ 20__ г.
1. Что такое интерпретатор Python? 2. Какой тип данных в Python является базовым? 3. Напишите программу на языке Python, которая будет имитировать работу системы управления базой данных. Программа должна содержать следующие функции: 1. Создание новой базы данных: программа должна предоставлять пользователю возможность создать новую базу данных с заданным именем.		

2. Добавление новой записи в базу данных: программа должна предоставлять пользователю возможность добавить новую запись в базу данных, содержащую псевдоним, полное имя и пароль.

3. Поиск записи в базе данных: программа должна предоставлять пользователю возможность найти запись в базе данных по псевдониму, полному имени и идентификатору.

4. Обновление записи в базе данных: программа должна предоставлять пользователю возможность обновить существующую запись в базе данных, изменяя псевдоним, полное имя и пароль.

5. Удаление записи из базы данных: программа должна предоставлять пользователю возможность удалить существующую запись из базы данных.

6. Вывод всех записей из базы данных: программа должна предоставлять пользователю возможность вывести все записи из базы данных.

7. Создание отчетов: программа должна предоставлять пользователю возможность создавать отчеты на основе данных в базе данных.

8. Обработка ошибок: программа должна предоставлять пользователю возможность обрабатывать ошибки, возникающие при работе с базой данных.

9. Закрытие базы данных: программа должна предоставлять пользователю возможность закрыть базу данных после окончания работы с ней.

Критерии оценивания

Оценка 5 «отлично» - на 85% и более вопросов дан верный ответ и в практической части реализованы все основные требования.

Оценка 4 «хорошо» - на 75% и более вопросов дан верный ответ и в практической части реализована часть основных требований.

Оценка 3 «удовлетворительно» - на 55% и более вопросов дан верный ответ и в практической части требования реализованы меньше, чем наполовину.

Оценка 2 «неудовлетворительно» - правильные ответы даны ниже, чем на 55% вопросов, и требования в практической части не реализованы.

4. Экзаменационная ведомость.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

МДК 01.03 Разработка мобильных приложений

Подготовки специалистов среднего звена по специальности

код специальности 09.02.07 Информационные системы и программирование

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности (профессии) СПО 09.02.07 Информационные системы и программирование базовой подготовки программы МДК 01.03 Разработка мобильных приложений.

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчик: Бахаев Д. В., преподаватель,
Голубенко М. И., преподаватель.

ОДОБРЕН
цикловой методической комиссией
Протокол № 1
от «2» сентября 2022 г.
Председатель Гаврикова Е. Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения МДК, подлежащие проверке
3. Оценка освоения МДК
 - 3.1. Контроль и оценка освоения МДК
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет материалов
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения МДК 01.03 Разработка мобильных приложений обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 Информационные системы и программирование базового уровня следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями

Обучающийся должен знать:

З 1	- основные этапы разработки программного обеспечения;
З 2	- основные принципы технологии структурного и объектно-ориентированного программирования;
З 3	- способы оптимизации и приемы рефакторинга;
З 4	- основные принципы отладки и тестирования программных продуктов.

Обучающийся должен уметь:

У 1	- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
У 2	- создавать программу по разработанному алгоритму как отдельный модуль;
У 3	- выполнять отладку и тестирование программы на уровне модуля;
У 4	- осуществлять разработку кода программного модуля на современных языках программирования;
У 5	- уметь выполнять оптимизацию и рефакторинг программного кода;
У 6	- оформлять документацию на программные средства.

Личностные результаты с учетом особенностей учебной дисциплины:

ЛР1	- осознающий себя гражданином и защитником великой страны
ЛР2	- проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	- демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих
ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие

	насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личностного и профессионального, конструктивного «цифрового следа».
ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	ЛР 8. Проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	ЛР 9. Уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности.
ЛР 10	ЛР 10. Принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности

Л 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия; выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
Л 19	- профессиональная идентичность и ответственность.
Л 20	- самооценка и рефлексия результатов своей деятельности и развития

Обучающийся должен иметь практический опыт:

- в разработке кода программного продукта на основе готовой спецификации на уровне модуля;
- в использовании инструментальных средств на этапе отладки программного продукта;
- в проведении тестирования программного модуля по определенному сценарию;
- в использовании инструментальных средств на этапе отладки программного продукта;
- в разработке мобильных приложений.

Формируемые ОК:

- ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;
- ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;
- ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;
- ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;
- ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;
- ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;
- ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;
- ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня

физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

Формируемые ПК:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ.

Личностные результаты учебной дисциплины

Код ЛР	Личностные результаты реализации программы (дескрипторы)
ЛР 1	-осознающий себя гражданином и защитником великой страны
ЛР 2	-проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	-демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих
ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личностного и профессионального, конструктивного «цифрового следа»
ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий,

	нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	- проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	- уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 10	- принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР 17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
ЛР 19	- профессиональная идентичность и ответственность
ЛР 20	- самооценка и рефлексия результатов своей деятельности и развития

Формами промежуточной аттестации по МДК являются другие формы контроля - 5 семестр, экзамен – 6 семестр.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по МДК осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих и профессиональных компетенций:

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней;	Определяет необходимый язык программирования; проводит разработку кода программного модуля.

ОК 1, 2, 3, 5, 9; ПК 1.2.	
У 2. Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 3, 5, 9; ПК 1.1, 1.2.	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма.
У 3. Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 3, 5, 9; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля.
У 4. Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 3, 5, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.
У 5. Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 3, 5, 9; ПК 1.4, 1.5.	Производит оптимизацию программного кода и его рефакторинг.
У 6. Оформлять документацию на программные средства. ОК 1, 2, 3, 4, 5, 6, 7, 9; 11 ПК 1.2.	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства.
Знать:	
З 1. Основные этапы разработки программного обеспечения ОК 1, 2, 3, 4, 5, 9; ПК 1.1, 1.2.	Знает основные процессы жизненного цикла программного обеспечения; на основе процессов понимает последовательность разработки программного обеспечения.
З 2. Основные принципы технологии структурного и объектно-ориентированного программирования; ОК 1, 2, 3, 4, 5, 9; ПК 1.1, 1.2.	Понимает принципиальные различия в технологиях структурного и объектно-ориентированного программирования; определяет на основе какой технологии построен код программы;
З 3. Способы оптимизации и приемы рефакторинга; ОК 1, 2, 3, 4, 5, 9; ПК 1.3, 1.4, 1.5.	Знает основные способы оптимизации и рефакторинга кода; применяет методы к написанному коду; составляет код программы на основе методов.
З 4. Основные принципы отладки и тестирования программных продуктов. ОК 1, 2, 3, 4, 5, 9; ПК 1.3, 1.4.	Понимает последовательность действий при отладке программного продукта; осознает принципы и методы тестирования; создает тестовые кейсы на основе различных методов тестирования.

3.ОЦЕНКА ОСВОЕНИЯ МДК

3.1. Контроль и оценка освоения МДК по темам (разделам)

Элемент МДК	Текущий контроль		Промежуточная аттестация	
	Осваиваемые результаты	Метод контроля	Проверяемые результаты	Форма контроля
Раздел. 1 Разработка мобильных приложений				

Тема 1.1 Основные платформы и языки разработки мобильных приложений	У 1-6 З 1-4 ОК 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11; ПК 1.1, 1.2, 1.3, 1.4., 1.5, 1.6	Устный опрос, практические и лабораторные работы, тестирование	У 1-6 З 1-4 ОК 1, 2, 7, 9, ПК 1.1, 1.2, 1.3, 1.4., 1.5, 1.6	Тестирование – 5 семестр
Тема 1.2 Создание и тестирование модулей для мобильных приложений	У 1-6 З 1-4 ОК 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11; ПК 1.1, 1.2, 1.3, 1.4., 1.5, 1.6	Устный опрос, практические и лабораторные работы, тестирование и экзамен		Экзамен – 6 семестр

3.1.1. Методы и критерии оценивания

1. Устный опрос. Критерии оценивания.

Оценка 5 «отлично» - ответил на вопросы в объеме лекционного и дополнительного материала, дал полные грамотные ответы на все дополнительные вопросы.

Оценка 4 «хорошо» - грамотно изложил ответы на вопросы, но содержание и формулировки имеют отдельные неточности (допускается нечеткая формулировка определений), в полной мере ответил на заданные дополнительные вопросы.

Оценка 3 «удовлетворительно» - ответил на часть вопросов в объеме лекционного материала и ответил на часть дополнительных вопросов.

Оценка 2 «неудовлетворительно» - допустил ошибки в определении базовых понятий, искажил их смысл, не ответил на дополнительные вопросы.

2. Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

3. Самостоятельная работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме; учтены все требования к данной работе; самостоятельно поставлены цели и задачи работы, соответствующие заданной теме/проблеме; получены результаты в соответствии с поставленной целью; работа оформлена аккуратно и грамотно.

Оценка 4 «хорошо» - выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью, и объем выполненной части работы позволяет получить недостаточно результатов в соответствии с поставленной целью.

4. Лабораторная работа. Критерии оценивания.

Выполнение работы в полном объеме с соблюдением необходимой последовательности

проведения опытов, измерений – 2 балла;

Рациональный и самостоятельный выбор и подготовка необходимого оборудования для выполнения работ, обеспечивающих получение точных результатов – 2 балл;

Описание хода лабораторной работы в логической последовательности – 1 балл;

Корректная формулировка выводов по результатам лабораторной работы – 2 балла;

Выполнения всех записей, таблиц, рисунков, чертежей, графиков, вычислений в соответствии с заданием, технически грамотно и аккуратно – 2 балла;

Соблюдение правил техники безопасности при выполнении лабораторной работы – 1 балл

Перевод баллов в отметку:

Оценка 5 «отлично» - от 9 до 10 баллов

Оценка 4 «хорошо» - от 6 до 8 баллов.

Оценка 3 «удовлетворительно» - от 3 до 5 баллов.

Оценка 2 «неудовлетворительно» - от 1 до 2 баллов.

5. Практическая работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме, с соблюдением алгоритма выполнения: последовательности проведения измерений, заполнения таблиц, графиков и др.; правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; получены результаты в соответствии с поставленной целью.

Оценка 4 «хорошо»- выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью и объем выполненной части работы не позволяет получить никаких результатов в соответствии с поставленной целью.

4. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Контрольная работа

1.Форма проведения: контрольная работа (тестирование).

2.Условия выполнения:

1.Инструкция для обучающихся:

2.Время выполнения: 90 минут на тестовое задание.

3.Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.

4.Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда программирования PyCharm с установленным интерпретатором Python не ниже версии 3.8; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

5.Информационные источники, допустимые к использованию на экзамене: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».

6.Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения тестирования

1. Перечень тем (разделов), выносимых на тестирование:

1. Основные платформы мобильных приложений, сравнительная характеристика

2. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения
3. Основные языки для разработки мобильных приложений (Java, ObjectiveC и др.)
4. Инструменты разработки мобильных приложений (JDK/AndroidStudio/WebView/Phonegap и др.)
5. Инструментарий среды разработки мобильных приложений. Структура типичного мобильного приложения.
6. Архитектура платформы Android. Dalvik Virtual Machine. Среда разработки для Android. Eclipse IDE. П
7. Плагин ADT. Android Virtual Device.
8. Android SDK. Версии SDK и Android API Level.
9. Инструментарий среды разработки мобильных приложений
10. Структура типичного мобильного приложения
11. Элементы управления и контейнеры
12. Работа со списками
13. Способы хранения данных

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней; ОК 1, 2, 3, 5, 9, 10; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля.
У 2. Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 3, 5, 9, 10; ПК 1.1, 1.2.	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма.
У 3. Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 3, 5, 9, 10; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля.
У 4. Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 3, 5, 9, 10; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.
У 5. Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 3, 5, 9, 10; ПК 1.4, 1.5.	Производит оптимизацию программного кода и его рефакторинг.
У 6. Оформлять документацию на программные средства. ОК 1, 2, 3, 4, 5, 6, 7, 9, 10; 11 ПК 1.2.	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства.
Знать:	
З 1. Основные этапы разработки программного обеспечения	Знает основные процессы жизненного цикла программного обеспечения; на

ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.1, 1.2.	основе процессов понимает последовательность разработки программного обеспечения.
3 2. Основные принципы технологии структурного и объектно-ориентированного программирования; ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.1, 1.2.	Понимает принципиальные различия в технологиях структурного и объектно-ориентированного программирования; определяет на основе какой технологии построен код программы;
3 3. Способы оптимизации и приемы рефакторинга; ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.3, 1.4, 1.5.	Знает основные способы оптимизации и рефакторинга кода; применяет методы к написанному коду; составляет код программы на основе методов.
3 4. Основные принципы отладки и тестирования программных продуктов. ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.3, 1.4.	Понимает последовательность действий при отладке программного продукта; осознает принципы и методы тестирования; создает тестовые кейсы на основе различных методов тестирования.

Образец тестового задания

1. Что такое мобильное приложение?
 - а) Программа, которая устанавливается на мобильное устройство и выполняет определенные функции.
 - б) Веб-приложение, которое работает только в браузере.
 - в) Приложение, которое работает только на настольных компьютерах.
2. Какие языки программирования используются для разработки мобильных приложений?
 - а) JavaScript, Python, Java, Swift.
 - б) JavaScript, Python, C++, Swift.
 - в) JavaScript, Python, Java, C++.
3. Что такое React Native?
 - а) Фреймворк для разработки мобильных приложений на JavaScript.
 - б) Библиотека для создания веб-приложений.
 - в) Язык программирования для создания игр.
4. Какие преимущества имеет разработка мобильных приложений на JavaScript?
 - а) Высокая скорость разработки, возможность быстрого прототипирования.
 - б) Легкость интеграции с веб-сервисами и базами данных.
 - в) Кроссплатформенность, возможность использования одного кода для разных платформ.
5. Какие инструменты используются для разработки мобильных приложений на JavaScript?
 - а) React Native, Ionic, Cordova.
 - б) React Native, Angular, Vue.js.
 - в) React Native, Node.js, Express.js.
6. Какие функции можно реализовать в мобильном приложении на JavaScript?
 - а) Работа с камерой, GPS, уведомлениями.
 - б) Работа с базами данных, веб-сервисами.
 - в) Работа с графикой, анимацией.

7. Какие преимущества имеет React Native перед нативными мобильными приложениями?
- Более быстрая разработка, возможность использования одного кода для разных платформ.
 - Более низкая стоимость разработки, возможность использования одного кода для разных платформ.
 - Более легкая интеграция с веб-сервисами и базами данных.
8. Какие платформы поддерживает React Native?
- iOS и Android.
 - iOS, Android, Windows Phone.
 - iOS, Android, BlackBerry.
9. Какие преимущества имеет разработка мобильных приложений на JavaScript перед разработкой нативных приложений?
- Более быстрая разработка, возможность использования одного кода для разных платформ.
 - Более низкая стоимость разработки, возможность использования одного кода для разных платформ.
 - Более легкая интеграция с веб-сервисами и базами данных.
10. Какие инструменты используются для тестирования мобильных приложений на JavaScript?
- Jest, Enzyme, Detox.
 - Mocha, Chai, Sinon.
 - Karma, Jasmine, Protractor.
11. Какие инструменты используются для сборки и развертывания мобильных приложений на JavaScript?
- npm, yarn, webpack.
 - npm, yarn, gulp.
 - npm, yarn, grunt.
12. Какие преимущества имеет разработка мобильных приложений на JavaScript перед разработкой нативных приложений?
- Более быстрая разработка, возможность использования одного кода для разных платформ.
 - Более низкая стоимость разработки, возможность использования одного кода для разных платформ.
 - Более легкая интеграция с веб-сервисами и базами данных.
13. Какие инструменты используются для создания пользовательского интерфейса в мобильных приложениях на JavaScript?
- React Native, Ionic, Cordova.
 - React Native, Angular, Vue.js.
 - React Native, Node.js, Express.js.
14. Что такое JavaScript?
- Язык программирования, используемый для создания веб-страниц.
 - Язык программирования, используемый для создания мобильных приложений.
 - Язык программирования, используемый для создания игр.
15. Какие основные принципы лежат в основе JavaScript?
- Объектно-ориентированное программирование.
 - Функциональное программирование.

- с) Прототипное программирование.
16. Какие основные конструкции языка JavaScript?
- а) Переменные, функции, операторы, циклы.
б) Классы, методы, свойства, наследование.
с) Объекты, массивы, прототипы, замыкания.
17. Какие возможности предоставляет JavaScript для работы с веб-страницами?
- а) Изменение стилей элементов на странице.
б) Обработка событий на странице.
с) Создание и манипуляция элементами на странице.
18. Какие библиотеки и фреймворки могут использоваться вместе с JavaScript для разработки веб-приложений?
- а) React, Angular, Vue.js.
б) jQuery, MooTools, Prototype.
с) Node.js, Express.js, Socket.io.
19. Каковы основные этапы разработки мобильного приложения?
20. Какие основные компоненты включает в себя мобильное приложение?
21. Какие технологии обычно используются при разработке мобильных приложений?
22. Что такое кроссплатформенная разработка мобильных приложений и каковы ее преимущества?
23. Какие виды монетизации мобильных приложений существуют?

4. Эталонные ответы:

1. а, 2. а, 3. а, 4. а, 5. а, 6. а, 7. а, 8. а, 9. а, 10. а, 11. а, 12. а, 13. а, 14. а, 15. с, 16. а, 17. с, 18.а
19. создание концепции и проектирование, разработка и тестирование, публикация и поддержка, 20. пользовательский интерфейс, базу данных, серверные компоненты, 21. iOS, Android, React Native, Flutter, Cordova, Ionic, Unity, 22. возможность разработки на различных платформах с использованием одной кодовой базы; снижение затрат на разработку; удобство использования для пользователей, 23. реклама, подписка, внутренняя монетизация, интеграция со сторонними сервисами.

Критерии оценивания

- Оценка 5 «отлично» - на 85% и более вопросов дан верный ответ.
Оценка 4 «хорошо» - на 75% и более вопросов дан верный ответ.
Оценка 3 «удовлетворительно» - на 55% и более вопросов дан верный ответ.
Оценка 2 «неудовлетворительно» - правильные ответы даны ниже, чем на 55% вопросов.

КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ИТОГОВОЙ АТТЕСТАЦИИ

2. Экзамен

1.Форма проведения: тестовое и практическое задания.

2.Условия выполнения:

- 1.Инструкция для обучающихся:
2.Время выполнения: 90 минут на тестовое задание, 30 – на практическую работу.
3.Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.
4.Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда программирования PyCharm с установленным интерпретатором Python не ниже версии 3.8; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде

КГА ПОУ «ДИТК» и сети Интернет.

5. Информационные источники, допустимые к использованию на экзамене: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».

6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения промежуточной аттестации

3.1. Перечень тем, выносимых на экзамен

1. Основные платформы и языки разработки мобильных приложений:

- основные платформы мобильных приложений, сравнительная характеристика;
- нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения;
- установка инструментария и настройка среды для разработки мобильных приложений;
- основные языки для разработки мобильных приложений (Java, Objective-C и др.);
- установка инструментария и настройка среды для разработки мобильных приложений;
- инструменты разработки мобильных приложений (JDK/AndroidStudio/WebView/Phonegap и др.);
- установка среды разработки мобильных приложений с применением виртуальной машины.

2. Создание и тестирование модулей для мобильных приложений:

- инструментарий среды разработки мобильных приложений;
- создание эмуляторов и подключение устройств;
- структура типичного мобильного приложения;
- создание локального сервера;
- структура типичного веб-приложения;
- создание и настройка шаблона проекта;
- элементы управления и контейнеры;
- изучение и комментирование кода;
- обработчики событий;
- изменение элементов дизайна;
- обработка событий: подсказки;
- обработка событий: цветовая индикация;
- работа со списками;
- подготовка стандартных модулей;
- обработка событий: переключение между экранами;
- способы хранения данных;
- передача данных между модулями;
- способы взаимодействий с базами данных с помощью API;
- рефакторинг кода;
- тестирование и оптимизация мобильного приложения;
- тестирование и оптимизация веб-приложения.

3.2. Перечень вопросов, выносимых на экзамен

1. Основные платформы мобильных приложений, сравнительная характеристика

2. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения
3. Основные языки для разработки мобильных приложений (Java, ObjectiveC и др.)
4. Инструменты разработки мобильных приложений (JDK/AndroidStudio/WebView/Phonegap и др.)
5. Инструментарий среды разработки мобильных приложений. Структура типичного мобильного приложения.
6. Архитектура платформы Android. Dalvik Virtual Machine. Среда разработки для Android. Eclipse IDE. П
7. Плагин ADT. Android Virtual Device.
8. Android SDK. Версии SDK и Android API Level.
9. Инструментарий среды разработки мобильных приложений
10. Структура типичного мобильного приложения
11. Элементы управления и контейнеры
12. Работа со списками
13. Способы хранения данных
14. Структура проекта Android-приложения в Eclipse. Каталоги ресурсов. Файл R.java
15. Графический интерфейс пользователя в Android-приложениях. XML-разметка интерфейса.
16. XML-разметка интерфейса пользователя
17. Ресурсы в Android-приложениях
18. Ресурсы в Windows Phone-приложениях.
19. Многопоточные приложения в Android и Windows Phone. Использование системных таймеров и системного времени.
20. Использование объектов Intent. Intent-фильтры
21. Использование ресурсов. Ссылки на ресурсы. Загрузка простых типов из ресурсов. Загрузка файлов произвольного типа.
22. Пользовательские настройки. Использование SharedPreferences. Виды настроек. Службы в Android. Компонент Service.
23. Датчики мобильных устройств. Управление датчиками в приложении. Виды датчиков и особенности их использования
24. Программный доступ к дисплею устройства. Менеджер окон. Параметры дисплея.

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней; ОК 1, 2, 3, 5, 9, 10; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля.
У 2. Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 3, 5, 9, 10; ПК 1.1, 1.2.	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма.

У 3. Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 3, 5, 9, 10; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля.
У 4. Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 3, 5, 9, 10; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.
У 5. Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 3, 5, 9, 10; ПК 1.4, 1.5.	Производит оптимизацию программного кода и его рефакторинг.
У 6. Оформлять документацию на программные средства. ОК 1, 2, 3, 4, 5, 6, 7, 9, 10; 11 ПК 1.2.	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства.
Знать:	
З 1. Основные этапы разработки программного обеспечения ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.1, 1.2.	Знает основные процессы жизненного цикла программного обеспечения; на основе процессов понимает последовательность разработки программного обеспечения.
З 2. Основные принципы технологии структурного и объектно-ориентированного программирования; ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.1, 1.2.	Понимает принципиальные различия в технологиях структурного и объектно-ориентированного программирования; определяет на основе какой технологии построен код программы;
З 3. Способы оптимизации и приемы рефакторинга; ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.3, 1.4, 1.5.	Знает основные способы оптимизации и рефакторинга кода; применяет методы к написанному коду; составляет код программы на основе методов.
З 4. Основные принципы отладки и тестирования программных продуктов. ОК 1, 2, 3, 4, 5, 9, 10, 11; ПК 1.3, 1.4.	Понимает последовательность действий при отладке программного продукта; осознает принципы и методы тестирования; создает тестовые кейсы на основе различных методов тестирования.

Образец тестового задания

1. Набор средств программирования, который содержит инструменты, необходимые для создания, компиляции и сборки мобильного приложения называется:

а) Android SDK

б) JDK

в) плагин ADT

г) Android NDK

2. С какой целью был создан Open Handset Alliance?

а) писать историю развития ОС Android

б) продавать смартфоны под управлением Android

- в) рекламировать смартфоны под управлением Android
г) разрабатывать открытые стандарты для мобильных устройств
3. С какой целью инструмент Intel* Graphics Performance Analyzers (Intel* GPA) System Analyzer используется в среде разработки Intel* Beacon Mountain?
а) позволить разработчикам оптимизировать загрузенность системы при использовании процедур OpenGL
б) для ускорения работы эмулятора в среде разработки
в) для оптимизированной обработки данных и изображений
г) позволить разработчикам эффективно распараллелить C++ мобильные приложения
4. Библиотеки, реализованные на базе PacketVideo OpenCORE:
а) Media Framework
б) SQLite
в) FreeType
г) 3D библиотеки
5. Какой движок баз данных используется в ОС Android?
а) InnoDB
б) DBM
в) MyISAM
г) SQLite
6. С какой целью инструмент Intel* Integrated Performance Primitives (Intel* IPP) используется в среде разработки Intel* Beacon Mountain?
а) для оптимизированной обработки данных и изображений
б) позволить разработчикам оптимизировать загрузенность системы при использовании процедур OpenGL
в) для ускорения работы эмулятора в среде разработки
г) позволить разработчикам эффективно распараллелить C++ мобильные приложения
7. Intel XDK поддерживает разработку под:
а) JavaFX Mobile
б) Apple iOS, BlackBerry OS
в) MtkOS, Symbian OS, Microsoft Windows 8
г) Android, Apple iOS, Microsoft Windows 8, Tizen
8. Каждый приемник широковещательных сообщений является наследником класса...
а) ViewReceiver
б) IntentReceiver
в) ContentProvider
г) BroadcastReceiver
9. Какой класс является основным строительным блоком для компонентов пользовательского интерфейса (UI), определяет прямоугольную область экрана и отвечает за прорисовку и обработку событий?
а) GUI
б) View
в) UIComponent
г) Widget
10. Какой слушатель используется для отслеживания события касания экрана устройства?
а) OnPressListener

б) onTouchListener

в) OnClickListener

г) OnInputListener

11. В какой папке необходимо разместить XML файлы, которые определяют все меню приложения?

а) res/value

б) res/items

в) res/layout

г) res/menu

12. Фоновые приложения ...

а) после настройки не предполагают взаимодействия с пользователем,

большую часть времени находятся и работают в скрытом состоянии

б) выполняют свои функции и когда видимы на экране, и когда скрыты другими приложениями

в) небольшие приложения, отображаемые в виде графического объекта на рабочем столе

г) большую часть времени работают в фоновом режиме, однако допускают взаимодействие с пользователем и после настройки

13. Полный иерархический список обязательных файлов и папок проекта можно увидеть на вкладке ...

а) Package Explorer

б) Internet Explorer

в) Navigator

г) Project Explorer

14. Какой компонент управляет распределенным множеством данных приложения?

а) сервис (Service)

б) активность (Activity)

в) приемник широковещательных сообщений (Broadcast Receiver)

г) контент-провайдер (Content Provider)

15. Какой язык разметки используется для описания иерархии компонентов графического пользовательского интерфейса Android-приложения?

а) html

б) xml

в) gml

г) xhtml

16. Выберите верную последовательность действий, необходимых для создания в приложении контент-провайдера.

а) Создание класса наследника от класса ContentProvider; Определение способа организации данных; Заполнение контент-провайдера данными

б) Проектирование способа хранения данных; Определение способа организации данных;

в) Создание класса наследника от класса ContentProvider; Заполнение контент-провайдера данными; Определение способа работы с данными

г) Проектирование способа хранения данных; Создание класса наследника от класса ContentProvider; Определение строки авторизации провайдера, URI для его строк и имен

столбцов

17. Выберите верные утверждения относительно объекта-намерения (Intent).

- а) представляет собой структуру данных, содержащую описание операции, которая должна быть выполнена, и обычно используется для запуска активности или сервиса
- б) используется для передачи сообщений пользователю
- в) используется для получения инструкций от пользователя
- г) используются для передачи сообщений между основными компонентами приложений

18. Расположение элементов мобильного приложения:

- а) полезно для передачи иерархии
- б) влияет на удобство использования
- в) полезно для создания пространственных отношений между объектами на экране и объектами реального мира
- г) все варианты ответа верны

19. Какие элементы управления применяются для действий по настройке?

- а) командные элементы управления
- б) элементы выбора
- в) элементы ввода
- г) элементы отображения

20. Примерами комбо-элементов не являются:

- а) комбо-список
- б) все вышеперечисленное
- в) комбо-кнопка
- г) комбо-поле

21. Дизайн или проектирование интерфейса для графических дизайнеров:

- а) все варианты ответа верны
- б) прозрачность и понятность информации
- в) тон, стиль, композиция, которые являются атрибутами бренда
- г) передача информации о поведении посредством ожидаемого назначения

22. Более крупные элементы:

- а) привлекают больше внимания
- б) все варианты ответа верны
- в) размер не влияет на уровень внимания
- г) привлекают меньше внимания

23. К традиционным типографическим инструментам не относят

- а) масштаб
- б) цвет
- в) разреженность
- г) выравнивание по сетке

24. К элементам ввода относят:

- а) ограничивающие элементы ввода
- б) ползунки
- в) счетчики
- г) все вышеперечисленное

25. Выделяют следующие категории плотности экрана для Android-устройств:

- а) HDPI, XHDPI, XXHDPI, и XXXHDPI

- б) правильный вариант ответа отсутствует
- в) LDPI, MDPI, HDPI, XHDPI, XXHDPI, и XXXHDPI
- г) LDPI, MDPI, HDPI
26. Следующие утверждения не верны:
- а) не используйте интерфейсные элемент
- б) картинки работают быстрее, чем слова
- в) на любом шаге должна быть возможность вернуться назад
- г) если объекты похожи, они должны выполнять сходные действия
27. Следующие утверждения верны:
- а) текстура бесполезна для передачи различий или привлечения внимания
- б) восприятие направления затруднено при больших размерах объектов
- в) все варианты ответа верны
- г) люди легко воспринимают контрастность
28. Основные вкладки (FixedTabs) удобны при отображении
- а) от четырех вкладок
- б) двух вкладок
- в) трех и более вкладок
- г) трех и менее вкладок
29. Диалоговое окно, содержащее линейку процесса выполнения какого-то действия – это
- а) DatePickerDialog
- б) AlertDialog
- в) ProgressDialog
- г) DialogFragment
30. Уведомления стоит использовать, когда
- а) сообщение не требует ответа пользователя, но важно для продолжения его работы
- б) сообщение является важным и требует немедленного прочтения и ответа
- в) сообщение является важным, однако требует немедленного прочтения, но не ответа
- г) сообщение является важным, однако не требует немедленного прочтения и ответа
31. Какой метод запускает новую активность?
- а) startActivity()
- б) beginActivity()
- в) intentActivity()
- г) newActivity()

Образец практического задания

Необходимо разработать инженерный калькулятор.

Требования к заданию:

1. Обязательное наличие графического интерфейса.
2. Классическое расположение числовых и операционных кнопок.
3. Наличие дисплея (поля вывода результата вычисления).
4. Наличие следующих операций: очистка поля, корень, сложение, вычитание, умножение, деление, рассчитать, плюс/минус и точка.
5. Практичных и удобный дизайн (минимальная стилизация компонентов).

Эталоны ответов

Тестовая часть 1. а, 2. а, 3. а, 4. а, 5. а, 6. а, 7. а, 8. а, 9. а, 10. а, 11. а, 12. а, 13. а, 14. а, 15. с, 16. а, 17. с, 18. а, 19. б, 20. г, 21. а, 22. а, 23. б, 24. г, 25. в, 26. а, 27. в, 28. г, 29. в, 30. г, 31. а.

Практическая часть

```
from tkinter import *
import math
import tkinter.messagebox

root = Tk()
root.title("Scientific Calculator")
root.configure(background='white')
root.resizable(width=False, height=False)
root.geometry("480x568+450+90")
calc = Frame(root)
calc.grid()

class Calc():
    def __init__(self):
        self.total = 0
        self.current = ""
        self.input_value = True
        self.check_sum = False
        self.op = ""
        self.result = False

    def numberEnter(self, num):
        self.result = False
        firstnum = txtDisplay.get()
        secondnum = str(num)
        if self.input_value:
            self.current = secondnum
            self.input_value = False
        else:
            if secondnum == '.':
                if secondnum in firstnum:
                    return
            self.current = firstnum + secondnum
        self.display(self.current)

    def sum_of_total(self):
        self.result = True
        self.current = float(self.current)
        if self.check_sum == True:
            self.valid_function()
        else:
            self.total = float(txtDisplay.get())

    def display(self, value):
        txtDisplay.delete(0, END)
        txtDisplay.insert(0, value)

    def valid_function(self):
        if self.op == "add":
            self.total += self.current
```



```

if self.op == "sub":
    self.total -= self.current
if self.op == "multi":
    self.total *= self.current
if self.op == "divide":
    self.total /= self.current
if self.op == "mod":
    self.total %= self.current
self.input_value = True
self.check_sum = False
self.display(self.total)

def operation(self, op):
    self.current = float(self.current)
    if self.check_sum:
        self.valid_function()
    elif not self.result:
        self.total = self.current
        self.input_value = True
    self.check_sum = True
    self.op = op
    self.result = False

def Clear_Entry(self):
    self.result = False
    self.current = "0"
    self.display(0)
    self.input_value = True

def All_Clear_Entry(self):
    self.Clear_Entry()
    self.total = 0

def pi(self):
    self.result = False
    self.current = math.pi
    self.display(self.current)

def tau(self):
    self.result = False
    self.current = math.tau
    self.display(self.current)

def e(self):
    self.result = False
    self.current = math.e
    self.display(self.current)

def mathPM(self):
    self.result = False
    self.current = -(float(txtDisplay.get()))
    self.display(self.current)

```

```
def squared(self):
    self.result = False
    self.current = math.sqrt(float(txtDisplay.get()))
    self.display(self.current)

def cos(self):
    self.result = False
    self.current = math.cos(math.radians(float(txtDisplay.get())))
    self.display(self.current)

def cosh(self):
    self.result = False
    self.current = math.cosh(math.radians(float(txtDisplay.get())))
    self.display(self.current)

def tan(self):
    self.result = False
    self.current = math.tan(math.radians(float(txtDisplay.get())))
    self.display(self.current)

def tanh(self):
    self.result = False
    self.current = math.tanh(math.radians(float(txtDisplay.get())))
    self.display(self.current)

def sin(self):
    self.result = False
    self.current = math.sin(math.radians(float(txtDisplay.get())))
    self.display(self.current)

def sinh(self):
    self.result = False
    self.current = math.sinh(math.radians(float(txtDisplay.get())))
    self.display(self.current)

def log(self):
    self.result = False
    self.current = math.log(float(txtDisplay.get()))
    self.display(self.current)

def exp(self):
    self.result = False
    self.current = math.exp(float(txtDisplay.get()))
    self.display(self.current)

def acosh(self):
    self.result = False
    self.current = math.acosh(float(txtDisplay.get()))
    self.display(self.current)

def asinh(self):
```

```

self.result = False
self.current = math.asinh(float(txtDisplay.get()))
self.display(self.current)

def expm1(self):
    self.result = False
    self.current = math.expm1(float(txtDisplay.get()))
    self.display(self.current)

def lgamma(self):
    self.result = False
    self.current = math.lgamma(float(txtDisplay.get()))
    self.display(self.current)

def degrees(self):
    self.result = False
    self.current = math.degrees(float(txtDisplay.get()))
    self.display(self.current)

def log2(self):
    self.result = False
    self.current = math.log2(float(txtDisplay.get()))
    self.display(self.current)

def log10(self):
    self.result = False
    self.current = math.log10(float(txtDisplay.get()))
    self.display(self.current)

def log1p(self):
    self.result = False
    self.current = math.log1p(float(txtDisplay.get()))
    self.display(self.current)

added_value = Calc()

txtDisplay = Entry(calc, font=('Helvetica', 20, 'bold'),
                  bg='black', fg='white',
                  bd=30, width=28, justify=RIGHT)
txtDisplay.grid(row=0, column=0, columnspan=4, pady=1)
txtDisplay.insert(0, "0")

numberpad = "789456123"
i = 0
btn = []
for j in range(2, 5):
    for k in range(3):
        btn.append(Button(calc, width=6, height=2,
                          bg='black', fg='white',
                          font=('Helvetica', 20, 'bold'),
                          bd=4, text=numberpad[i]))

```

```
btn[i].grid(row=j, column=k, pady=1)
btn[i]["command"] = lambda x=numberpad[i]: added_value.numberEnter(x)
i += 1
```

```
btnClear = Button(calc, text=chr(67), width=6,
                  height=2, bg='powder blue',
                  font=('Helvetica', 20, 'bold'),
                  bd=4, command=added_value.Clear_Entry
                  ).grid(row=1, column=0, pady=1)
```

```
btnAllClear = Button(calc, text=chr(67) + chr(69),
                    width=6, height=2,
                    bg='powder blue',
                    font=('Helvetica', 20, 'bold'),
                    bd=4,
                    command=added_value.All_Clear_Entry
                    ).grid(row=1, column=1, pady=1)
```

```
btnsq = Button(calc, text="\u221A", width=6, height=2,
              bg='powder blue', font=('Helvetica',
              20, 'bold'),
              bd=4, command=added_value.squared
              ).grid(row=1, column=2, pady=1)
```

```
btnAdd = Button(calc, text="+", width=6, height=2,
                bg='powder blue',
                font=('Helvetica', 20, 'bold'),
                bd=4, command=lambda: added_value.operation("add")
                ).grid(row=1, column=3, pady=1)
```

```
btnSub = Button(calc, text="-", width=6,
                height=2, bg='powder blue',
                font=('Helvetica', 20, 'bold'),
                bd=4, command=lambda: added_value.operation("sub")
                ).grid(row=2, column=3, pady=1)
```

```
btnMul = Button(calc, text="x", width=6,
                height=2, bg='powder blue',
                font=('Helvetica', 20, 'bold'),
                bd=4, command=lambda: added_value.operation("multi")
                ).grid(row=3, column=3, pady=1)
```

```
btnDiv = Button(calc, text="/", width=6,
                height=2, bg='powder blue',
                font=('Helvetica', 20, 'bold'),
                bd=4, command=lambda: added_value.operation("divide")
                ).grid(row=4, column=3, pady=1)
```

```
btnZero = Button(calc, text="0", width=6,
                 height=2, bg='black', fg='white',
                 font=('Helvetica', 20, 'bold'),
                 bd=4, command=lambda: added_value.numberEnter(0)
```

```

        ).grid(row=5, column=0, pady=1)

btnDot = Button(calc, text=".", width=6,
                height=2, bg='powder blue',
                font=('Helvetica', 20, 'bold'),
                bd=4, command=lambda: added_value.numberEnter(".")
                ).grid(row=5, column=1, pady=1)
btnPM = Button(calc, text=chr(177), width=6,
                height=2, bg='powder blue', font=('Helvetica', 20, 'bold'),
                bd=4, command=added_value.mathPM
                ).grid(row=5, column=2, pady=1)

btnEquals = Button(calc, text="=", width=6,
                    height=2, bg='powder blue',
                    font=('Helvetica', 20, 'bold'),
                    bd=4, command=added_value.sum_of_total
                    ).grid(row=5, column=3, pady=1)
# ROW 1 :
btnPi = Button(calc, text="pi", width=6,
                height=2, bg='black', fg='white',
                font=('Helvetica', 20, 'bold'),
                bd=4, command=added_value.pi
                ).grid(row=1, column=4, pady=1)

btnCos = Button(calc, text="Cos", width=6,
                 height=2, bg='black', fg='white',
                 font=('Helvetica', 20, 'bold'),
                 bd=4, command=added_value.cos
                 ).grid(row=1, column=5, pady=1)

btntan = Button(calc, text="tan", width=6,
                 height=2, bg='black', fg='white',
                 font=('Helvetica', 20, 'bold'),
                 bd=4, command=added_value.tan
                 ).grid(row=1, column=6, pady=1)

btnsin = Button(calc, text="sin", width=6,
                 height=2, bg='black', fg='white',
                 font=('Helvetica', 20, 'bold'),
                 bd=4, command=added_value.sin
                 ).grid(row=1, column=7, pady=1)

# ROW 2 :
btn2Pi = Button(calc, text="2pi", width=6,
                 height=2, bg='black', fg='white',
                 font=('Helvetica', 20, 'bold'),
                 bd=4, command=added_value.tau
                 ).grid(row=2, column=4, pady=1)

btnCosh = Button(calc, text="Cosh", width=6,
                  height=2, bg='black', fg='white',
                  font=('Helvetica', 20, 'bold'),

```

```

        bd=4, command=added_value.cosh
    ).grid(row=2, column=5, pady=1)

    btntanh = Button(calc, text="tanh", width=6,
        height=2, bg='black', fg='white',
        font=('Helvetica', 20, 'bold'),
        bd=4, command=added_value.tanh
    ).grid(row=2, column=6, pady=1)

    btnsinh = Button(calc, text="sinh", width=6,
        height=2, bg='black', fg='white',
        font=('Helvetica', 20, 'bold'),
        bd=4, command=added_value.sinh
    ).grid(row=2, column=7, pady=1)

# ROW 3 :
    btnlog = Button(calc, text="log", width=6,
        height=2, bg='black', fg='white',
        font=('Helvetica', 20, 'bold'),
        bd=4, command=added_value.log
    ).grid(row=3, column=4, pady=1)

    btnExp = Button(calc, text="exp", width=6, height=2,
        bg='black', fg='white',
        font=('Helvetica', 20, 'bold'),
        bd=4, command=added_value.exp
    ).grid(row=3, column=5, pady=1)

    btnMod = Button(calc, text="Mod", width=6,
        height=2, bg='black', fg='white',
        font=('Helvetica', 20, 'bold'),
        bd=4, command=lambda: added_value.operation("mod")
    ).grid(row=3, column=6, pady=1)

    btnE = Button(calc, text="e", width=6,
        height=2, bg='black', fg='white',
        font=('Helvetica', 20, 'bold'),
        bd=4, command=added_value.e
    ).grid(row=3, column=7, pady=1)

# ROW 4 :
    btnlog10 = Button(calc, text="log10", width=6,
        height=2, bg='black', fg='white',
        font=('Helvetica', 20, 'bold'),
        bd=4, command=added_value.log10
    ).grid(row=4, column=4, pady=1)

    btncos = Button(calc, text="log1p", width=6,
        height=2, bg='black', fg='white',
        font=('Helvetica', 20, 'bold'),
        bd=4, command=added_value.log1p
    ).grid(row=4, column=5, pady=1)

```

```

btnexpm1 = Button(calc, text="expm1", width=6,
                  height=2, bg='black', fg='white',
                  font=('Helvetica', 20, 'bold'),
                  bd=4, command=added_value.expm1
                  ).grid(row=4, column=6, pady=1)

btngamma = Button(calc, text="gamma", width=6,
                  height=2, bg='black', fg='white',
                  font=('Helvetica', 20, 'bold'),
                  bd=4, command=added_value.lgamma
                  ).grid(row=4, column=7, pady=1)
# ROW 5 :
btnlog2 = Button(calc, text="log2", width=6,
                 height=2, bg='black', fg='white',
                 font=('Helvetica', 20, 'bold'),
                 bd=4, command=added_value.log2
                 ).grid(row=5, column=4, pady=1)

btndeg = Button(calc, text="deg", width=6,
                height=2, bg='black', fg='white',
                font=('Helvetica', 20, 'bold'),
                bd=4, command=added_value.degrees
                ).grid(row=5, column=5, pady=1)

btnacosh = Button(calc, text="acosh", width=6,
                  height=2, bg='black', fg='white',
                  font=('Helvetica', 20, 'bold'),
                  bd=4, command=added_value.acosh
                  ).grid(row=5, column=6, pady=1)

btnasinh = Button(calc, text="asinh", width=6,
                  height=2, bg='black', fg='white',
                  font=('Helvetica', 20, 'bold'),
                  bd=4, command=added_value.asinh
                  ).grid(row=5, column=7, pady=1)

lblDisplay = Label(calc, text="Scientific Calculator",
                  font=('Helvetica', 30, 'bold'),
                  bg='black', fg='white', justify=CENTER)

lblDisplay.grid(row=0, column=4, columnspan=4)

def iExit():
    iExit = tkinter.messagebox.askyesno("Scientific Calculator",
                                         "Do you want to exit ?")
    if iExit > 0:
        root.destroy()
        return

```

```

def Scientific():
    root.resizable(width=False, height=False)
    root.geometry("944x568+0+0")

def Standard():
    root.resizable(width=False, height=False)
    root.geometry("480x568+0+0")

menubar = Menu(calc)

# ManuBar 1 :
filemenu = Menu(menubar, tearoff=0)
menubar.add_cascade(label='File', menu=filemenu)
filemenu.add_command(label="Standard", command=Standard)
filemenu.add_command(label="Scientific", command=Scientific)
filemenu.add_separator()
filemenu.add_command(label="Exit", command=iExit)

# ManuBar 2 :
editmenu = Menu(menubar, tearoff=0)
menubar.add_cascade(label='Edit', menu=editmenu)
editmenu.add_command(label="Cut")
editmenu.add_command(label="Copy")
editmenu.add_separator()
editmenu.add_command(label="Paste")

root.config(menu=menubar)

root.mainloop()

```

Критерии оценивания

Оценка 5 «отлично» - на 85% и более вопросов дан верный ответ и в практической части реализованы все основные требования.

Оценка 4 «хорошо» - на 75% и более вопросов дан верный ответ и в практической части реализована часть основных требований.

Оценка 3 «удовлетворительно» - на 55% и более вопросов дан верный ответ и в практической части требования реализованы меньше, чем наполовину.

Оценка 2 «неудовлетворительно» - правильные ответы даны ниже, чем на 55% вопросов, и требования в практической части не реализованы.

4.Экзаменационная ведомость.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

МДК.01.04 Системное программирование

подготовки специалистов среднего звена по специальности

код специальности 09.02.07 Информационные системы и программирование

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности СПО 09.02.07 Информационные системы и программирование базовой подготовки программы Системное программирование.

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчик: Бахаев Д. В., преподаватель,
Голубенко М. И., преподаватель.

ОДОБРЕН
цикловой методической комиссией
Протокол № 1
от «3» сентября 2022 г.
Председатель Гаврикова Е. Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения МДК, подлежащие проверке
3. Оценка освоения МДК
 - 3.1. Контроль и оценка освоения МДК
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет материалов
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения МДК.01.04 Системное программирование обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 Информационные системы и программирование базового уровня следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями

Обучающийся должен знать:

З 1	- основные этапы разработки программного обеспечения;
З 2	- основные принципы технологии структурного и объектно-ориентированного программирования;
З 3	- способы оптимизации и приемы рефакторинга;
З 4	- основные принципы отладки и тестирования программных продуктов.

Обучающийся должен уметь:

У 1	- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
У 2	- создавать программу по разработанному алгоритму как отдельный модуль;
У 3	- выполнять отладку и тестирование программы на уровне модуля;
У 4	- осуществлять разработку кода программного модуля на современных языках программирования;
У 5	- уметь выполнять оптимизацию и рефакторинг программного кода;
У 6	- оформлять документацию на программные средства.

Личностные результаты с учетом особенностей учебной дисциплины:

ЛР1	- осознающий себя гражданином и защитником великой страны
ЛР2	- проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	- демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих
ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение.

	Стремящийся к формированию в сетевой среде личностного и профессионального, конструктивного «цифрового следа».
ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	ЛР 8. Проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	ЛР 9. Уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности.
ЛР 10	ЛР 10. Принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
Л 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия; выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
Л 19	- профессиональная идентичность и ответственность.
Л 20	- самооценка и рефлексия результатов своей деятельности и развития

Обучающийся должен иметь практический опыт:

- в разработке кода программного продукта на основе готовой спецификации на уровне модуля;
- в использовании инструментальных средств на этапе отладки программного продукта;
- в проведении тестирования программного модуля по определенному сценарию;
- в использовании инструментальных средств на этапе отладки программного продукта;
- в разработке мобильных приложений.

Формируемые ОК:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

Формируемые ПК:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ.

Личностные результаты учебной дисциплины

Код ЛР	Личностные результаты реализации программы (дескрипторы)
ЛР 1	-осознающий себя гражданином и защитником великой страны
ЛР 2	-проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	-демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине,

	принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих
ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личностного и профессионального, конструктивного «цифрового следа»
ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	- проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	- уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 10	- принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность

ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР 17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
ЛР 19	- профессиональная идентичность и ответственность
ЛР 20	- самооценка и рефлексия результатов своей деятельности и развития

Формами промежуточной аттестации по МДК являются проверочная работа 7 семестр, экзамен – 8 семестр.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по МДК осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих и профессиональных компетенций:

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней; ОК 1, 2, 3, 5, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля.
У 2. Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 3, 5, 9; ПК 1.1, 1.2.	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма.
У 3. Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 3, 5, 9; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля.
У 4. Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 3, 5, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.
У 5. Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 3, 5, 9; ПК 1.4, 1.5.	Производит оптимизацию программного кода и его рефакторинг.
У 6. Оформлять документацию на программные средства. ОК 1, 2, 3, 4, 5, 6, 7, 9; 11 ПК 1.2.	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства.
Знать:	
З 1. Основные этапы разработки программного обеспечения ОК 1, 2, 3, 4, 5, 9; ПК 1.1, 1.2.	Знает основные процессы жизненного цикла программного обеспечения; на основе процессов понимает последовательность разработки программного обеспечения.

3 2. Основные принципы технологии структурного и объектно-ориентированного программирования; ОК 1, 2, 3, 4, 5, 9; ПК 1.1, 1.2.	Понимает принципиальные различия в технологиях структурного и объектно-ориентированного программирования; определяет на основе какой технологии построен код программы;
3 3. Способы оптимизации и приемы рефакторинга; ОК 1, 2, 3, 4, 5, 9; ПК 1.3, 1.4, 1.5.	Знает основные способы оптимизации и рефакторинга кода; применяет методы к написанному коду; составляет код программы на основе методов.
3 4. Основные принципы отладки и тестирования программных продуктов. ОК 1, 2, 3, 4, 5, 9; ПК 1.3, 1.4.	Понимает последовательность действий при отладке программного продукта; осознает принципы и методы тестирования; создает тестовые кейсы на основе различных методов тестирования.

3.ОЦЕНКА ОСВОЕНИЯ МДК

3.1. Контроль и оценка освоения МДК по темам (разделам)

Элемент МДК	Текущий контроль		Промежуточная аттестация	
	Осваиваемые результаты	Метод контроля	Проверяемые результаты	Форма контроля
	Раздел. 1 Системное программирование			
Тема 1.1 Программирование на языке низкого уровня	З 1-4 У 1-6 ОК 1-9 ПК 1.1-1.6	Устный опрос, практические и лабораторные работы, тестирование	З 1-4 У 1-6 ОК 1-9 ПК 1.1-1.6	Тестирование – 7 семестр Экзамен – 8 семестр

3.1.1. Методы и критерии оценивания

1. Устный опрос. Критерии оценивания.

Оценка 5 «отлично» - ответил на вопросы в объеме лекционного и дополнительного материала, дал полные грамотные ответы на все дополнительные вопросы.

Оценка 4 «хорошо» - грамотно изложил ответы на вопросы, но содержание и формулировки имеют отдельные неточности (допускается нечеткая формулировка определений), в полной мере ответил на заданные дополнительные вопросы.

Оценка 3 «удовлетворительно» - ответил на часть вопросов в объеме лекционного материала и ответил на часть дополнительных вопросов.

Оценка 2 «неудовлетворительно» - допустил ошибки в определении базовых понятий, искажил их смысл, не ответил на дополнительные вопросы.

2. Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

3. Самостоятельная работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме; учтены все требования к данной

работе; самостоятельно поставлены цели и задачи работы, соответствующие заданной теме/проблеме; получены результаты в соответствии с поставленной целью; работа оформлена аккуратно и грамотно.

Оценка 4 «хорошо» - выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью, и объем выполненной части работы позволяет получить недостаточно результатов в соответствии с поставленной целью.

4.Лабораторная работа. Критерии оценивания.

Выполнение работы в полном объеме с соблюдением необходимой последовательности проведения опытов, измерений – 2 балла;

Рациональный и самостоятельный выбор и подготовка необходимого оборудования для выполнения работ, обеспечивающих получение точных результатов – 2 балл;

Описание хода лабораторной работы в логической последовательности – 1 балл;

Корректная формулировка выводов по результатам лабораторной работы – 2 балла;

Выполнения всех записей, таблиц, рисунков, чертежей, графиков, вычислений в соответствии с заданием, технически грамотно и аккуратно – 2 балла;

Соблюдение правил техники безопасности при выполнении лабораторной работы – 1 балл

Перевод баллов в отметку:

Оценка 5 «отлично» - от 9 до 10 баллов

Оценка 4 «хорошо» - от 6 до 8 баллов.

Оценка 3 «удовлетворительно» - от 3 до 5 баллов.

Оценка 2 «неудовлетворительно» - от 1 до 2 баллов.

5. Практическая работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме, с соблюдением алгоритма выполнения: последовательности проведения измерений, заполнения таблиц, графиков и др.; правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; получены результаты в соответствии с поставленной целью.

Оценка 4 «хорошо»- выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью и объем выполненной части работы не позволяет получить никаких результатов в соответствии с поставленной целью.

КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Контрольная работа

1.Форма проведения: контрольная работа (письменное тестирование).

2.Условия выполнения:

1.Инструкция для обучающихся:

2.Время выполнения: 90 минут на контрольную работу.

3.Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.
 4.Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда программирования PyCharm с установленным интерпретатором Python не ниже версии 3.8; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

5.Информационные источники, допустимые к использованию на экзамене: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».

6.Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения тестирования

3.1. Перечень тем (разделов), выносимых на тестирование:

1. Программирование на языке низкого уровня:

- подсистемы управления ресурсами;
- управление процессами;
- управление потоками;
- параллельная обработка потоков;
- создание процессов и потоков;
- обмен данными между процессами. Передача сообщений;
- анонимные и именованные каналы;
- сетевое программирование сокетов;
- динамически подключаемые библиотеки DLL;
- сервисы;
- виртуальная память. Выделение памяти процессам;
- работа с буфером экрана.

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней; ОК 1, 2, 3, 5, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля.
У 2. Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 3, 5, 9; ПК 1.1, 1.2.	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма.
У 3. Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 3, 5, 9; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля.
У 4. Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 3, 5, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.

У 5. Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 3, 5, 9; ПК 1.4, 1.5.	Производит оптимизацию программного кода и его рефакторинг.
У 6. Оформлять документацию на программные средства. ОК 1, 2, 3, 4, 5, 6, 7, 9; 11 ПК 1.2.	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства.
Знать:	
З 1. Основные этапы разработки программного обеспечения ОК 1, 2, 3, 4, 5, 9; ПК 1.1, 1.2.	Знает основные процессы жизненного цикла программного обеспечения; на основе процессов понимает последовательность разработки программного обеспечения.
З 2. Основные принципы технологии структурного и объектно-ориентированного программирования; ОК 1, 2, 3, 4, 5, 9; ПК 1.1, 1.2.	Понимает принципиальные различия в технологиях структурного и объектно-ориентированного программирования; определяет на основе какой технологии построен код программы;
З 3. Способы оптимизации и приемы рефакторинга; ОК 1, 2, 3, 4, 5, 9; ПК 1.3, 1.4, 1.5.	Знает основные способы оптимизации и рефакторинга кода; применяет методы к написанному коду; составляет код программы на основе методов.
З 4. Основные принципы отладки и тестирования программных продуктов. ОК 1, 2, 3, 4, 5, 9; ПК 1.3, 1.4.	Понимает последовательность действий при отладке программного продукта; осознает принципы и методы тестирования; создает тестовые кейсы на основе различных методов тестирования.

Образец письменного тестирования

- 1) Что представляет собой прикладное ПО? Приведите примеры прикладного ПО.
- 2) Что представляет собой промежуточное ПО? Приведите примеры промежуточного ПО.
- 3) Что представляет собой системное ПО?
- 4) Какие функции выполняет системное ПО?
- 5) Какие группы системного ПО существуют?
- 6) Дайте определение системного программирования.
- 7) На какие классы разделяют системное ПО в рамках традиционной классификации?
- 8) На какие классы разделяют системное ПО в рамках альтернативной классификации?
- 9) Приведите примеры базового ПО.
- 10) Дайте определение операционной системы.
- 11) Для чего предназначены интерфейсные оболочки ОС?
- 12) Что представляют собой системы управления файлами? Для чего они предназначены?
- 13) Приведите примеры сервисного ПО.
- 14) Что представляет собой современная система программирования?
- 15) Какие средства включает в себя система программирования?
- 16) Дайте определение транслятора.
- 17) Что представляет собой ассемблер?
- 18) Дайте определение компилятора.
- 19) Дайте определение интерпретатора.

- 20) В чем отличие интерпретатора от компилятора?
- 21) На какие этапы разделяют процесс компиляции?
- 22) Какие фазы анализа выделяют?
- 23) Что представляет собой лексический анализ?
- 24) Что представляет собой синтаксический анализ?
- 25) Что представляет собой семантический анализ?
- 26) Какие фазы синтеза выделяют?

4. Эталонные ответы:

1) Прикладное ПО представляет собой программы, разработанные для выполнения конкретных задач, которые пользователи могут использовать для решения своих повседневных задач или профессиональных целей. Примеры прикладного ПО включают текстовые редакторы, электронные таблицы, бухгалтерские программы, графические редакторы и игры, 2) Промежуточное ПО – это слой программного обеспечения, который находится между системным ПО и прикладным ПО. Оно обеспечивает абстракцию от аппаратного обеспечения и предоставляет интерфейсы для разработки приложений. Примеры промежуточного ПО включают веб-серверы, серверы баз данных, интеграционные платформы и микросервисы, 3) Системное ПО – это программное обеспечение, необходимое для управления аппаратными ресурсами компьютера и обеспечения функционирования других программ, 4) Системное ПО выполняет функции, связанные с управлением устройствами компьютера, планированием задач, управлением памятью, обслуживанием файловой системы и предоставлением интерфейсов для прикладного программирования, 5) Системное ПО можно разделить на два основных типа: базовое ПО и операционные системы. Базовое ПО включает в себя компоненты, необходимые для функционирования компьютера, такие как драйверы устройств и утилиты. Операционные системы обеспечивают интерфейс между прикладным программным обеспечением и аппаратными ресурсами, 6) Системное программирование – это процесс разработки программного обеспечения, направленного на управление аппаратными и программными ресурсами компьютера, 7) В рамках традиционной классификации системное ПО делится на три класса: базовое, операционные системы и служебное. Базовое ПО обеспечивает основные функции, такие как управление памятью и устройствами. Операционные системы предоставляют интерфейс для работы с прикладными программами и другими системными компонентами. Служебное ПО включает программы и утилиты, которые обеспечивают дополнительные функции, такие как архиваторы, антивирусы и средства диагностики, 8) В альтернативной классификации системное ПО делится на четыре класса: базовые системы, оболочки операционных систем, системы управления базами данных и прикладные системы. Базовые системы обеспечивают основные функции, необходимые для работы компьютера. Оболочки операционных систем предоставляют графический интерфейс и другие дополнительные возможности. Системы управления базами данных обеспечивают хранение, извлечение и обработку данных. Прикладные системы включают программы для решения специфических задач, таких как обработка текстов, электронные таблицы и графические редакторы, 9) Базовое ПО включает драйверы устройств, утилиты и другие компоненты, необходимые для нормального функционирования компьютера. Примеры базового ПО включают операционные системы, драйверы устройств и файловые менеджеры, 10) Операционная система – это программное обеспечение, которое управляет ресурсами компьютера, такими как процессор, память, устройства и файлы, и предоставляет

интерфейс для прикладных программ и пользователей, 12) Интерфейсные оболочки ОС предназначены для упрощения взаимодействия пользователей с операционной системой путем предоставления графических интерфейсов, которые могут быть более интуитивно понятными и удобными для использования, чем командная строка, 13) Системы управления файлами представляют собой программное обеспечение, предназначенное для организации, хранения и поиска файлов на компьютере. Они обеспечивают выполнение основных операций с файлами, таких как создание, удаление, перемещение, копирование и изменение атрибутов файлов, 14) Сервисное ПО включает различные программы и утилиты, предназначенные для выполнения дополнительных задач, таких как дефрагментация диска, резервное копирование данных, восстановление системы после сбоев и удаление нежелательных файлов. Примеры сервисного ПО включают антивирусы, дефрагментаторы, архиваторы и средства резервного копирования, 15) Современная система программирования – это комплекс инструментов и технологий, предназначенных для разработки, отладки и развертывания программного обеспечения. Она включает в себя языки программирования, среды разработки, библиотеки и фреймворки, а также инструменты для автоматизации сборки, тестирования и развертывания приложений, 16) Транслятор – это программа или часть программы, которая переводит исходный код с одного языка программирования на другой. Трансляторы могут быть компиляторами, которые переводят исходный код в машинный код, или интерпретаторами, которые выполняют исходный код строка за строкой, 17) Ассемблер - это программа, которая переводит мнемонические коды ассемблера в двоичные инструкции процессора, 18) Компилятор - это программа, которая переводит текст программы на языке высокого уровня в машинные коды, 19) Интерпретатор - это программа, которая выполняет программу строка за строкой, не генерируя при этом исполняемый код, 20) Основное отличие интерпретатора от компилятора заключается в том, что интерпретатор выполняет программу сразу после перевода строки, в то время как компилятор сначала создает исполняемый файл, который затем выполняется, 21) Процесс компиляции разделяют на несколько этапов: лексический анализ, синтаксический анализ и семантический анализ, 22) Фазы анализа включают в себя: идентификацию переменных, определение типов данных, проверку синтаксиса и т.д., 23) Лексический анализ - это процесс разбиения исходного кода на отдельные лексемы (слова), 24) Синтаксический анализ - это процесс проверки правильности написания программы с точки зрения грамматики языка программирования, 25) Семантический анализ - это процесс определения смысла программы, проверка совместимости типов данных и т.п., 26) Фазы синтеза включают генерацию объектного кода и создание исполняемого файла.

5. Критерии оценивания

Оценка 5 «отлично» - на 85% и более вопросов дан верный ответ.

Оценка 4 «хорошо» - на 75% и более вопросов дан верный ответ.

Оценка 3 «удовлетворительно» - на 55% и более вопросов дан верный ответ.

Оценка 2 «неудовлетворительно» - правильные ответы даны ниже, чем на 55% вопросов.

КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ИТОГОВОЙ АТТЕСТАЦИИ

2. Экзамен

1. Форма проведения: письменное тестовое и практическое задания.

2. Условия выполнения:

1. Инструкция для обучающихся:
2. Время выполнения: 45 минут на письменное тестовое задание, 45 – на практическую работу.
3. Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.
4. Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.
5. Информационные источники, допустимые к использованию на экзамене: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».
6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения промежуточной аттестации

3.1. Перечень тем (разделов), выносимых на тестирование:

1. Программирование на языке низкого уровня:

- подсистемы управления ресурсами;
- управление процессами;
- управление потоками;
- параллельная обработка потоков;
- создание процессов и потоков;
- обмен данными между процессами. Передача сообщений;
- анонимные и именованные каналы;
- сетевое программирование сокетов;
- динамически подключаемые библиотеки DLL;
- сервисы;
- виртуальная память. Выделение памяти процессам;
- работа с буфером экрана.

3.2. Перечень практических заданий к экзамену:

1. Запросите реестр и выведите информацию о версии программного обеспечения браузера Internet Explorer или MS Edge, установленного на компьютере.
2. Проверьте список программ, которые запускаются при старте системы.
3. Перечислите значения реестра ветки «Software\Microsoft\Windows\CurrentVersion\Explorer».
4. Используйте метод модуля OS для вызова внешней программы Notepad.
5. Запустите программу Notepad путём создания процесса.
6. Перечислите список процессов, запущенных на системе.
7. Отправитель отправляет строку, предполагая, что получатель прослушивает порт 5000 машины, и отображает полученный контент, если он получает строку до свидания (без учета регистра), он заканчивает прослушивание.
8. Напишите программу, которая имитирует чат-бота: сервер заранее создает словарь (базу данных), а затем автоматически отвечает в соответствии с полученным контентом.

Эталонные ответы на практические задания

1.

```
import win32api  
import win32con
```

```

key          =          win32api.RegOpenKey(win32con.HKEY_LOCAL_MACHINE,
'SOFTWARE\\Microsoft\\Internet Explorer', 0,
          win32con.KEY_ALL_ACCESS)
print(win32api.RegQueryValue(key, ""))
print(win32api.RegQueryValueEx(key, 'Version'))
print(win32api.RegQueryInfoKey(key))
win32api.RegCloseKey(key)

```

Результат:

```

('9.11.17134.0', 1)
(34, 10, 132302030397159009)

```

2.

```

from win32api import *
from win32con import *

def get_values(fullname):
    name = str.split(fullname, '\\', 1)
    try:
        if name[0] == 'HKEY_LOCAL_MACHINE':
            key = RegOpenKey(HKEY_LOCAL_MACHINE, name[1], 0, KEY_READ)
        elif name[0] == 'HKEY_CURRENT_USER':
            key = RegOpenKey(HKEY_CURRENT_USER, name[1], 0, KEY_READ)
        elif name[0] == 'HKEY_CLASSES_ROOT':
            key = RegOpenKey(HKEY_CLASSES_ROOT, name[1], 0, KEY_READ)
        elif name[0] == 'HKEY_CURRENT_CONFIG':
            key = RegOpenKey(HKEY_CURRENT_CONFIG, name[1], 0, KEY_READ)
        elif name[0] == 'HKEY_USERS':
            key = RegOpenKey(HKEY_USERS, name[1], 0, KEY_READ)
        else:
            raise ValueError('Error,no key named', name[0])
        info = RegQueryInfoKey(key)
        for i in range(0, info[1]):
            ValueName = RegEnumValue(key, i)
            print(str.ljust(ValueName[0], 20), ValueName[1])
        RegCloseKey(key)
    except Exception as e:
        print('Sth is wrong')
        print(e)

if __name__ == '__main__':
    KeyNames =
    ['HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run',

```



```

r'HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce',
    r'HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run',

r'HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce']
for KeyName in KeyNames:
    print('*'*30)
    print(KeyName)
    get_values(KeyName)

```

Результат:

```

*****
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
SecurityHealth      %ProgramFiles%\Windows Defender\MSASCuiL.exe
Sysdiag             "C:\Program Files (x86)\Huorong\Sysdiag\bin\HipsTray.exe"
*****
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
*****
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
OneDrive            "C:\Users\ASUS\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background
Wechat              C:\Program Files (x86)\Tencent\WeChat\WeChat.exe
QQ2009              "E:\QQbrower\Bin\QQ.exe" /background
BaiduYunGuanjia     "E:\云盘\BaiduNetdisk\baidunetdisk.exe" AutoRun
BaiduYunDetect      "E:\云盘\BaiduNetdisk\YunDetectService.exe"
qbclipboard         "C:\Program Files (x86)\Tencent\QQBrowser\QQBrowser.exe" --type=assistant --clipboard
AlibabaMusic        C:\Program Files (x86)\Xiami\XiamiPC.exe --tray
Thunder             E:\Thunder\Program\Thunder.exe -silent -StartType:AutoRun
*****
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce

```

3.

```

import winreg

key = winreg.OpenKey(winreg.HKEY_CURRENT_USER,
r'Software\Microsoft\Windows\CurrentVersion\Explorer')
try:
    i = 0
    while 1:
        Name, Value, Type = winreg.EnumValue(key, i)
        print(repr(Name), ':', repr(Value), ':', Type)
        i += 1
except WindowsError as e:
    # print(e)
    pass

"""
_winreg.QueryValueEx(key,value_name)
Эта функция используется для получения данных и типа указанного значения в ключе, а
тип возвращаемого значения - кортеж.
"""

print('=' * 20)
Name = 'link'

```



```

    ]

def enumProcess():
    hSnapshot = kernel32.CreateToolhelp32Snapshot(15, 0)
    fProcessEntry32 = tagPROCESSENTRY32()
    processClass = collections.namedtuple("processInfo", "processName processID")
    processSet = []
    if hSnapshot:
        fProcessEntry32.dwSize = sizeof(fProcessEntry32)
        listloop = kernel32.Process32First(hSnapshot, byref(fProcessEntry32))
        while listloop:
            processName = (fProcessEntry32.szExeFile)
            processID = fProcessEntry32.th32ProcessID
            processSet.append(processClass(processName, processID))
            listloop = kernel32.Process32Next(hSnapshot, byref(fProcessEntry32))
        return processSet

for i in enumProcess():
    print("Name:", i.processName, "\tID:", i.processID)

```

Результат:

```

Name: b'[System Process]' ID: 0
Name: b'System' ID: 4
Name: b'Registry' ID: 120
Name: b'smss.exe' ID: 444
Name: b'csrss.exe' ID: 712
Name: b'wininit.exe' ID: 820
Name: b'services.exe' ID: 892
Name: b'lsass.exe' ID: 912
Name: b'svchost.exe' ID: 676
Name: b'svchost.exe' ID: 704
Name: b'fontdrvhost.exe' ID: 816
Name: b'WUDFHost.exe' ID: 1052
Name: b'svchost.exe' ID: 1124
Name: b'svchost.exe' ID: 1172
Name: b'svchost.exe' ID: 1368
Name: b'svchost.exe' ID: 1384
Name: b'svchost.exe' ID: 1504
Name: b'svchost.exe' ID: 1524
Name: b'svchost.exe' ID: 1600
Name: b'svchost.exe' ID: 1664

```

7.

Код получателя

```
import socket
```

```
# Использовать протокол IPV4, использовать протокол UDP для передачи данных
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
# Привязать порт и номер порта, пустая строка означает любой доступный IP-адрес этой машины
```

```
s.bind(('', 5000))
while True:
    data, addr = s.recvfrom(1024) # 1024 - указанный размер кеша
    # Отобразить полученный контент
    print(f'receive message:{data.decode()} from PORT {addr[1]} on {addr[0]}')
    if data.decode().lower() == 'bye':
        break
s.close()
```

Код отправителя

```
import socket
import sys
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
# Предполагается, что 192.168.0.104 - это IP-адрес принимающей машины
s.sendto("hello world".encode(),("192.168.0.104",5000))
s.sendto("bye".encode(),("192.168.0.104",5000))
s.close()
```

Результат:

```
receive message:hello world from PORT 61999 on 192.168.0.104
receive message:bye from PORT 61999 on 192.168.0.104
```

8.

Код сервера

```
import socket

words = {'how are you?': 'Fine,thank you', 'how old are you': '3', 'what is your name?':
'TianMaoSpirit',
        'where do you work': 'CQUPT', 'bye': 'Bye'} # Когда вы узнаете о разработке базы
данных позже, вы можете использовать поисковые роботы для получения и сохранения
данных в базе данных, а затем получить большое количество словарей из базы данных
HOST = "
PORT = 50007
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Привязать сокет
s.bind((HOST, PORT))
# Начать слушать
s.listen(1)
print('Listening at port:', PORT)
client_socket, addr = s.accept() # Пассивно принимать TCP-клиентские соединения,
(блокируя) ожидая прибытия соединения, если клиентский запрос получен, сокет и адрес
клиентского сокета будут возвращены
print('Connected by', addr)
while True:
    data = client_socket.recv(1024)
    data = data.decode()
    if not data:
```

```
        break
    print('Received message:', data)
    client_socket.sendall(words.get(data, 'nothing').encode())
client_socket.close()
s.close()
```

Код клиента

```
import socket
import sys
```

```
HOST = '127.0.0.1' # IP-адрес хоста сервера
```

```
PORT=50007 # Номер порта хоста сервера
```

```
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
```

```
try:
```

```
    s.connect((HOST,PORT))
```

```
except Exception as e:
```

```
    print('Server not found or not open')
```

```
    sys.exit()
```

```
while True:
```

```
    c = input('Input the content you want to send:')
```

```
    s.sendall(c.encode()) # отправить данные
```

```
    data = s.recv(1024) # Получать данные с сервера, установить размер кеша на 1 м = 1024
```

```
бит
```

```
    data = data.decode()
```

```
    print('Received:',data)
```

```
    if c.lower()=='bye':
```

```
        break
```

```
s.close() # Закрыть соединение
```

Результат:

```
Received message: how are you?
```

```
Received message: where do you work?
```

```
Received message: how old are you
```

```
Received message: what is your name?
```

```
Received message: bye
```

```
Input the content you want to send:how are you?
```

```
Received: Fine,thank you
```

```
Input the content you want to send:where do you work?
```

```
Received: nothing
```

```
Input the content you want to send:how old are you
```

```
Received: 3
```

```
Input the content you want to send:what is your name?
```

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней; ОК 1, 2, 3, 5, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля.
У 2. Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 3, 5, 9; ПК 1.1, 1.2.	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма.
У 3. Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 3, 5, 9; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля.
У 4. Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 3, 5, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.
У 5. Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 3, 5, 9; ПК 1.4, 1.5.	Производит оптимизацию программного кода и его рефакторинг.
У 6. Оформлять документацию на программные средства. ОК 1, 2, 3, 4, 5, 6, 7, 9; 11 ПК 1.2.	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства.
Знать:	
З 1. Основные этапы разработки программного обеспечения ОК 1, 2, 3, 4, 5, 9; ПК 1.1, 1.2.	Знает основные процессы жизненного цикла программного обеспечения; на основе процессов понимает последовательность разработки программного обеспечения.
З 2. Основные принципы технологии структурного и объектно-ориентированного программирования; ОК 1, 2, 3, 4, 5, 9; ПК 1.1, 1.2.	Понимает принципиальные различия в технологиях структурного и объектно-ориентированного программирования; определяет на основе какой технологии построен код программы;
З 3. Способы оптимизации и приемы рефакторинга; ОК 1, 2, 3, 4, 5, 9; ПК 1.3, 1.4, 1.5.	Знает основные способы оптимизации и рефакторинга кода; применяет методы к написанному коду; составляет код программы на основе методов.
З 4. Основные принципы отладки и тестирования программных продуктов. ОК 1, 2, 3, 4, 5, 9; ПК 1.3, 1.4.	Понимает последовательность действий при отладке программного продукта; осознает принципы и методы тестирования; создает

Часть А

1 Установите порядок, в котором осуществляется перевод произвольного двоичного числа в системе счисления с основанием $q=2n$. Укажите порядок следования всех 3 вариантов ответа: 1 рассмотреть каждую группу как n -разрядное двоичное число и записать ее соответствующей цифрой в системе счисления с основанием $q = 2n$; 2 двоичное число разбить слева и справа (целую и дробную части) на группы по цифр в каждой; 3 если в последних правой и левой группах окажется меньше празрядов, то их надо дополнить справа и слева нулями до нужного числа разрядов

- a. 3 1 2
- b. 2 1 3
- c. 1 2 3
- d. 3 2 1

2 По сколько цифр в группе нужно разбивать двоичное число при переводе его в четверичную систему счисления?

- a. 6
- b. 2
- c. 3
- d. 4

3 Переведите двоичное число 10001111011 в шестандцатеричную систему счисления:

- a. 5fa
- b. 8fb
- c. 8fc
- d. 8fa

4 Чему будет равен результат сложение двоичных чисел 1001 и 1000

- a. 10001
- b. 11001
- c. 10011
- d. 10101

5 Выполните деление в двоичной системе счисления $1110:10=$

- a. 101
- b. 011
- c. 110
- d. 111

6 Переведите двоичное число 1010101 в восьмеричную систему счисления

- a. 125
- b. 124
- c. 126
- d. 123

7 Установите порядок, в котором осуществляется перевод целого числа в системе счисления с основанием $q=2n$. Укажите порядок следования всех 3 вариантов ответа: 1 данное двоичное число разбить справа налево на группы по пцифр в каждой; 2 рассмотреть каждую группу как n -разрядное двоичное число и записать ее соответствующей цифрой в системе счисления с основанием $q = 2n$; 3 если в последней левой группе окажется меньше

раз рядов, то ее надо дополнить слева нулями до нужного числа разрядов

- a. 1 3 2
- b. 1 2 3
- c. 3 2 1
- d. 2 1 3

8 Переведите шестнадцатеричное число FACC в двоичную систему счисления

- a. 1111001011011001
- b. 1111101011001100
- c. 1011010111111100
- d. 1101100111011010

9 Выполните вычисление в двоичной системе счисления $110101-101=$

- a. 110100
- b. 110010
- c. 110000
- d. 110001

10 В саду 100 фруктовых деревьев - 14 яблонь и 42 груши. В какой системе счисления посчитаны деревья?

- a. В шестеричной
- b. В шестнадцатеричной
- c. В двоичной
- d. В восьмеричной

11 Умножьте в двоичной системе счисления $1111*11=$

- a. 1111111
- b. 101111
- c. 1010101
- d. 101101

12 Установите соответствие. Укажите соответствие для всех 4 вариантов ответа:

- 1) 4
- 2) 5
- 3) 6
- 4) 7

- a. 4=101 5=110 6=100 7=111
- b. 5=101 6=110 4=100 7=111
- c. 6=101 5=110 4=100 7=111
- d. 4=101 7=110 6=100 5=111

13 Выполните умножение в системе счисления $10101*101=$

- a. 110101011
- b. 1101001
- c. 111001001
- d. 1010101

14 Установите соответствие. Укажите соответствие для всех 4 вариантов ответа:

- a. A=1010 B=1111 F=1110 E=1011
- b. A=1010 F=1111 E=1110 B=1011
- c. A=1010 C=1111 F=1110 E=1011
- d. A=1010 B=1111 F=1110 C=1011

15 В какой системе счисления будет верным равенство $7+8=16$?

- a. 12
- b. 6
- c. 9
- d. 16

16 $FFFF+1=$

- a. 1FFF
- b. 10000
- c. FFFE
- d. FFF1

17 Выполните вычитание в двоичной системе счисления $1111-101$

- a. 1010
- b. 1110
- c. 1100
- d. 1001

18 Результат деления в двоичной системе счисления числа 110 на 11 равен

- a. 11
- b. 10
- c. 01
- d. 101

19 Перевести смешанное двоичное $111100000,101$ число в восьмеричную систему счисления

- a. 610,4
- b. 740,5
- c. 170,5
- d. 1E0,A

20 Переведите восьмеричное число 0,25 в двоичную систему счисления

- a. 0,010101
- b. 0,101010
- c. 0,110111
- d. 0,101011

21 Алгоритм-это:

- a. Указание на выполнение действий,
- b. Система правил, описывающая последовательность действий, которые необходимо выполнить для решения задачи,
- c. Процесс выполнения вычислений, приводящих к решению задачи

22 Свойство алгоритма – дискретность, выражает, что:

- a. Команды должны следовать последовательно друг за другом,
- b. Каждая команда должна быть описана в расчете на конкретного исполнителя,
- c. Разбиение алгоритма на конечное число команд

23 Формальное исполнение алгоритма – это:

- a. Исполнение алгоритма конкретным исполнителем с полной записью его рассуждений,
- b. Разбиение алгоритма на конкретное число команд и пошаговое их исполнение,
- c. Исполнение алгоритма не требует рассуждений, а осуществляется исполнителем автоматически
- d. Исполнение алгоритма осуществляется исполнителем на уровне его знаний

24 Самое важное свойство алгоритма:

- a. Визуальность,
- b. Массовость,
- c. Дискретность,
- d. Аудиальность,

25 Какой алгоритм называется линейным:

- a. Выполнение операций зависит от условия,
- b. Операции выполняются друг за другом,
- c. Одни и те же операции выполняются многократно
- d. Присутствие всех возможных операций в одном алгоритме

26 Графическое задание алгоритма – это:

- a. Способ представления алгоритма с помощью геометрических фигур,
- b. Представление алгоритма в форме таблиц и расчетных формул,
- c. Система обозначений и правил для единообразной и точной записи алгоритмов и их исполнения

27 В расчете на кого должен строиться алгоритм:

- a. В расчете на ЭВМ,
- b. В расчете на умственные способности товарища,
- c. В расчете на конкретного исполнителя

28 Какое из перечисленных свойств относится к свойствам алгоритма:

- a. Визуальность,
- b. Совокупность,
- c. Аудиальность,
- d. Понятность

29 Псевдокоманда DB означает

- a. определить байт;
- b. определить слово (2 байта);
- c. определить двойное слово (4 байта);
- d. определить 6 байт
- e. определить учетверенное слово (8 байт);

30 Псевдокоманда DW означает

- a. определить байт;
- b. определить слово (2 байта);
- c. определить двойное слово (4 байта);
- d. определить 6 байт
- e. определить учетверенное слово (8 байт);

31 Псевдокоманда DD означает

- a. определить байт;
- b. определить слово (2 байта);
- c. определить двойное слово (4 байта);
- d. определить 6 байт
- e. определить учетверенное слово (8 байт);

32 К сегментным регистрам относят

- a. CS, DS, SS и ES.
- b. AX, BX, CX и DX
- c. SP и BP.
- d. SI и DI

33 К индексным регистрам относят

- a. CS, DS, SS и ES.
- b. AX, BX, CX и DX
- c. SP и BP.
- d. SI и DI

34 Устанавливает бит результата в 1, если оба бита, бит источника и бит приемника установлены в 1

- a. AND
- b. OR
- c. XOR (НЕ ИЛИ)
- d. NOT

35 устанавливает бит результата в 1, если бит источника отличается от бита приемника.

- a. AND
- b. OR
- c. XOR
- d. NOT

36 Запись регистра в память осуществляется при помощи команды

- a. MOV
- b. XCHG
- c. XLAT

37 Порядковый номер элемента в массиве, который всегда начинается с нуля

- a. Индекс
- b. Команда
- c. Псевдокоманда
- d. Макрокоманда

38 Позволяет сохранять длину исходного текста программы

- a. Индекс
- b. Команда
- c. Псевдокоманда
- d. Макрокоманда

Часть Б

1. Запросите реестр и выведите информацию о версии программного обеспечения браузера Internet Explorer или MS Edge, установленного на компьютере.

Эталоны ответов к тестовой части

1. c; 2. b; 3. a; 4. d; 5. d; 6. b; 7. d; 8. d; 9. a; 10. b; 11. d; 12. d; 13. c; 14. a; 15. b; 16. a; 17. d; 18. d; 19. d; 20. b; 21. b; 22. a; 23. c; 24. b; 25. b; 26. a; 27. c; 28. d; 29. a; 30. b; 31. c; 32. b; 33. d; 34. a; 35. c; 36. a; 37. a; 38. d.

Эталоны ответов к практической части

1.

```
import win32api
import win32con

key = win32api.RegOpenKey(win32con.HKEY_LOCAL_MACHINE,
'SOFTWARE\\Microsoft\\Internet Explorer', 0,
win32con.KEY_ALL_ACCESS)
```

```
print(win32api.RegQueryValue(key,""))
print(win32api.RegQueryValueEx(key,'Version'))
print(win32api.RegQueryInfoKey(key))
win32api.RegCloseKey(key)
```

Критерии оценивания

Оценка 5 «отлично» - на 85% и более вопросов дан верный ответ и в практической части реализованы все основные требования.

Оценка 4 «хорошо» - на 75% и более вопросов дан верный ответ и в практической части реализована часть основных требований (больше половины).

Оценка 3 «удовлетворительно» - на 55% и более вопросов дан верный ответ и в практической части требования реализованы меньше, чем наполовину.

Оценка 2 «неудовлетворительно» - правильные ответы даны ниже, чем на 55% вопросов, и требования в практической части не реализованы.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

для специальности

09.02.07 Информационные системы и программирование

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности СПО 09.02.07 Информационные системы и программирование профессионального модуля ПМ.01 Разработка модулей программного обеспечения для компьютерных систем.

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчики:

Голубенко Максим Игоревич, преподаватель

Бахаев Денис Викторович преподаватель

ОДОБРЕН

цикловой методической комиссией

Протокол № 1

от «3» сентября 2022 г.

Председатель Гаврикова Е.Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения ПМ, подлежащие проверке
3. Оценка освоения ПМ
 - 3.1. Контроль и оценка освоения ПМ
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет материалов
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения ПМ.01 Разработка модулей программного обеспечения для компьютерных систем обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 Информационные системы и программирование следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями.

Обучающийся должен знать:

З 1	основные этапы разработки программного обеспечения;
З 2	основные принципы технологии структурного и объектно-ориентированного программирования;
З 3	способы оптимизации и приемы рефакторинга;
З 4	основные принципы отладки и тестирования программных продуктов,

Обучающийся должен уметь:

У 1	осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
У 2	создавать программу по разработанному алгоритму как отдельный модуль;
У 3	выполнять отладку и тестирование программы на уровне модуля;
У 4	осуществлять разработку кода программного модуля на современных языках программирования;
У 5	выполнять оптимизацию и рефакторинг программного кода;
У 6	оформлять документацию на программные средства.

Обучающийся должен иметь практический опыт: разработке кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений.

Формируемые ОК:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

ВД 2. Осуществление интеграции программных модулей.

Формируемые ПК:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ.

Личностные результаты с учетом особенностей учебной дисциплины:

Код ЛР	Личностные результаты реализации программы (дескрипторы)
ЛР 1	- осознающий себя гражданином и защитником великой страны
ЛР 2	- проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	- демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих
ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личностного и профессионального, конструктивного «цифрового следа»

ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	- проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	- уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 10	- принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР 17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
ЛР 19	- профессиональная идентичность и ответственность
ЛР 20	- самооценка и рефлексия результатов своей деятельности и развития

Промежуточной аттестации по ПМ является экзамен по билетам.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПМ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

Результатом освоения профессионального модуля является готовность обучающегося к выполнению вида профессиональной деятельности Сопровождение и обслуживание программного обеспечения компьютерных систем. Для подтверждения такой готовности обязательна констатация сформированности у обучающегося всех профессиональных компетенций, входящих в состав профессионального модуля. Общие компетенции формируются в процессе освоения ОПОП в целом, поэтому по результатам освоения профессионального модуля возможно оценивание положительной динамики их формирования.

Оценка запланированных результатов по ПМ

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У 1 Осуществлять разработку кода программного модуля на языках низкого и высокого уровней ОК 1, ОК 02, ОК 4, ОК 05 ОК 9, ПК 1.2, ПК 1.6.	определяет необходимый язык программирования; проводит разработку кода программного модуля.
У 2 Создавать программу по разработанному алгоритму как отдельный модуль ОК 1, ОК 2, ОК 03, ОК 4, ОК 9, ПК 1.1, ПК 1.2, ПК 1.6.	разрабатывает или выбирает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма с возможностью параллельного использования модуля в других программах.
У 3 Выполнять отладку и тестирование программы на уровне модуля ОК 1, ОК 2, ОК 05, ОК 9, ПК 1.3, ПК 1.4	определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы как на уровне модуля, так и на уровне всего программного комплекса.
У 4 Осуществлять разработку кода программного модуля на современных языках программирования ОК 1, ОК 02, ОК 4, ОК 05 ОК 9, ПК 1.2, ПК 1.6.	определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.
У 5 Выполнять оптимизацию и рефакторинг программного кода ОК 1, ОК 02, ОК 4, ОК 05 ОК 9, ПК 1.5.	определяет способы и производит оптимизацию программного кода и его рефакторинг.
У 6 Оформлять документацию на программные средства ОК 1, ОК 2, ОК 08, ОК 9, ПК 1.1	применяет стандарты оформления документации; грамотно оформляет документацию на программные средства..
Знать:	
З 1 Основные этапы разработки программного обеспечения ОК 1, ОК 2, ОК 4, ПК 2.1	знает основные процессы жизненного цикла программного обеспечения; на основе процессов понимает последовательность разработки программного обеспечения.
З 2 Основные принципы технологии структурного и объектно-ориентированного программирования ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.3	понимает принципиальные различия в технологиях структурного и объектно-ориентированного программирования; определяет на основе какой технологии построен код программы;
З 3 Способы оптимизации и приемы рефакторинга	знает основные способы оптимизации и рефакторинга кода; применяет методы к

ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2	написанному коду; составляет код программы на основе методов.
3 4 Основные принципы отладки и тестирования программных продуктов ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.4, ПК 2.5	понимает последовательность действий при отладке программного продукта; осознает принципы и методы тестирования; создает тестовые кейсы на основе различных методов тестирования

3. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ЭКЗАМЕНА ПО ПМ

1. Экзамен по билетам

1. Форма проведения: экзамен.

2. Условия выполнения

Время выполнения задания: 180 мин.

Оборудование учебного кабинета: автоматизированные рабочие места в количестве 15 шт.

Технические средства обучения: автоматизированные рабочие места в количестве 15 шт.

Информационные источники: отсутствуют.

Требования охраны труда: в соответствии с требованиями СНиП.

3. Пакет материалов для проведения экзамена:

1.3.1 Перечень тем, выносимых на экзамен:

1.1 Жизненный цикл ПО

1.2 Структурное программирование

1.3 Объектно-ориентированное программирование

1.4 Паттерны проектирования

1.5 Событийно-управляемое программирование

1.6 Оптимизация и рефакторинг кода

1.7 Разработка пользовательского интерфейса

1.8 Основы ADO.Net

2.1 Отладка и тестирование программного обеспечения

2.2 Документирование

3.1 Основные платформы и языки разработки мобильных приложений

3.2 Создание и тестирование модулей для мобильных приложений

4.1 Программирование на языке низкого уровня

3.2. Перечень вопросов, выносимых на экзамен

1. Какие основные принципы структурного программирования и как они реализуются в языке программирования Python?
2. Какие основные принципы объектно-ориентированного программирования и как они реализуются в языке программирования Python?
3. Какие основные паттерны проектирования и как они реализуются в языке программирования Python?
4. Какие основные принципы событийно-управляемого программирования и как они реализуются в языке программирования Python?
5. Какие основные методы оптимизации и рефакторинга кода существуют в языке программирования Python и как они могут быть применены?

6. Какие основные принципы и инструменты используются при разработке пользовательского интерфейса на Python?
7. Какие основные принципы и инструменты используются при работе с базами данных на Python?
8. Что такое отладка программного обеспечения и почему она важна?
9. Какие инструменты Python используются для отладки?
10. Что такое модульное тестирование и как оно применяется в Python?
11. Каких основных принципов следует придерживаться при тестировании программного обеспечения на Python?
12. Что такое документирование программного обеспечения и почему оно важно для проектов на Python?
13. Как документировать классы и методы в Python, и в чем отличие документации для классов от документации для функций?
14. Каким образом использование хорошей документации может улучшить совместную работу в команде разработчиков?
15. Как организовать документацию проекта на Python, чтобы она была легко доступной и понятной для других разработчиков и пользователей?
16. Какие основные платформы используются для разработки мобильных приложений, и как они отличаются друг от друга?
17. Какие языки программирования используются для разработки мобильных приложений под различные платформы, и какие преимущества и недостатки у каждого из них?
18. Что такое кросс-платформенная разработка мобильных приложений, и какие инструменты и фреймворки используются для этой цели?
19. Какие факторы следует учитывать при выборе платформы и языка разработки для мобильного приложения, и как эти факторы могут повлиять на процесс разработки и конечный результат?
20. Какие основные принципы следует соблюдать при проектировании модулей для мобильных приложений на Python, чтобы обеспечить их эффективность и масштабируемость?
21. Каким образом использование абстракций и интерфейсов может упростить процесс создания и тестирования модулей на Python? Приведите примеры.
22. Какие методы тестирования модулей являются наиболее эффективными для обнаружения ошибок и обеспечения надежности мобильного приложения на Python?
23. Каким образом интеграционное тестирование модулей влияет на общее качество мобильного приложения на Python, и как его проводить эффективно?
24. Что такое язык низкого уровня и какие характеристики определяют его? Как Python связан с программированием на языке низкого уровня?
25. Каким образом Python обеспечивает возможность работы с низкоуровневыми конструкциями и функциями, такими как указатели и работа с памятью?
26. Какие инструменты и библиотеки Python позволяют взаимодействовать с низкоуровневыми аспектами операционной системы, такими как файловая система, сетевые интерфейсы и управление процессами?
27. Каким образом Python обеспечивает возможность написания расширений на языке низкого уровня, таких как C или C++, и интеграции их с кодом на Python?

28. Какие преимущества и недостатки существуют при использовании Python для работы с низкоуровневыми задачами по сравнению с языками низкого уровня, такими как C или C++?
29. Какие сценарии использования могут потребовать написания низкоуровневого кода на Python? Приведите примеры таких сценариев.
30. Какие основные предосторожности и рекомендации следует учитывать при использовании Python для выполнения задач, требующих низкоуровневого программирования?

Эталоны ответов на вопросы:

1. Принципы структурного программирования включают в себя использование блоков, циклов и условных операторов для организации кода. В Python эти принципы реализуются с помощью следующих конструкций:

Блоки кода: В Python блоки кода обозначаются отступами. Каждый уровень отступа соответствует вложенности блока.

Циклы: Python поддерживает циклы for и while. Цикл for используется для итерации по элементам последовательности, а цикл while используется для выполнения блока кода, пока условие истинно.

Условные операторы: Python поддерживает условные операторы if, elif и else для проверки условий и выполнения соответствующего блока кода.

Пример кода, иллюстрирующего использование этих принципов:

```
# Блок кода
if condition:
    # Вложенный блок кода
    if condition2:
        # Еще более вложенный блок кода
        print('Условие истинно')
    else:
        print('Условие ложно')
else:
    print('Первое условие ложно')
```

В этом примере мы используем условные операторы if и else для проверки условий и выполнения соответствующего блока кода. Мы также используем отступы для обозначения вложенности блоков кода.

2. Принципы объектно-ориентированного программирования включают в себя инкапсуляцию, наследование и полиморфизм. В Python эти принципы реализуются с помощью следующих конструкций:

Инкапсуляция: В Python инкапсуляция достигается путем определения классов и методов внутри них. Классы определяют структуру объекта, а методы определяют его поведение.

Наследование: Python поддерживает множественное наследование, что позволяет классам наследовать свойства и методы от нескольких родительских классов.

Полиморфизм: Python поддерживает полиморфизм через перегрузку функций и методов. Это позволяет одному и тому же имени функции или метода иметь разные реализации в зависимости от типов аргументов.

Пример кода, иллюстрирующего использование этих принципов:

```
class Animal:
```

```

def speak(self):
    print('Животные говорят')
class Dog(Animal):
    def speak(self):
        print('Собака лает')
class Cat(Animal):
    def speak(self):
        print('Кот мяукает')
dog = Dog()
cat = Cat()
dog.speak() # Выводит: Собака лает
cat.speak() # Выводит: Кот мяукает

```

В этом примере мы создаем классы `Animal`, `Dog` и `Cat`, которые наследуют поведение от класса `Animal`. Метод `speak` переопределяется в каждом классе, что демонстрирует полиморфизм.

3.

Паттерны проектирования - это шаблоны, которые помогают разработчикам создавать более гибкие и масштабируемые системы. В Python есть несколько популярных паттернов проектирования, включая:

Фабричный метод: Этот паттерн позволяет создавать объекты без явного указания конструктора. В Python это можно реализовать с помощью функции, которая возвращает новый объект.

Стратегия: Этот паттерн позволяет отделить алгоритм от его реализации. В Python это можно реализовать с помощью абстрактных базовых классов и наследования.

Декоратор: Этот паттерн позволяет добавлять поведение к существующим классам и методам без изменения их исходного кода. В Python это можно реализовать с помощью декораторов.

Прототип: Этот паттерн позволяет создавать новые объекты на основе существующих прототипов. В Python это можно реализовать с помощью словарей, которые содержат свойства и методы объекта.

Пример кода, иллюстрирующего использование паттерна проектирования "Фабричный метод":

```

class AnimalFactory:
    def create_animal(self, animal_type):
        if animal_type == 'dog':
            return Dog()
        elif animal_type == 'cat':
            return Cat()
        else:
            raise ValueError('Неизвестный тип животного')

```

В этом примере мы создаем класс `AnimalFactory`, который содержит метод `create_animal`. Этот метод принимает тип животного в качестве аргумента и возвращает новый объект соответствующего класса (`Dog` или `Cat`). Это демонстрирует использование паттерна проектирования "Фабричный метод".

4.

Принципы событийно-управляемого программирования включают в себя обработку событий и реакцию на них. В Python эти принципы реализуются с помощью следующих конструкций:

Обработка событий: В Python обработка событий осуществляется с помощью обработчиков событий. Обработчик событий - это функция или метод, который вызывается при возникновении определенного события.

Многопоточность: Python поддерживает многопоточность, что позволяет выполнять несколько задач одновременно. Это особенно полезно при обработке событий, когда необходимо реагировать на несколько событий одновременно.

Асинхронность: Python также поддерживает асинхронное программирование, что позволяет выполнять задачи параллельно, не блокируя основной поток выполнения. Это особенно полезно при обработке событий, когда необходимо реагировать на события, не ожидая завершения других задач.

Пример кода, иллюстрирующего использование этих принципов:

```
import threading
import asyncio
def event_handler(event):
    print(f'Обработано событие: {event}')
def event_generator():
    for i in range(10):
        yield i
def event_producer():
    for event in event_generator():
        threading.Thread(target=event_handler, args=(event,)).start()
async def event_consumer():
    for event in event_generator():
        await asyncio.sleep(1)
        print(f'Получено событие: {event}')
event_producer()
asyncio.run(event_consumer())
```

В этом примере мы создаем два потока: один для генерации событий (`event_producer`) и один для обработки событий (`event_consumer`). Обработчик событий (`event_handler`) вызывается в отдельном потоке для каждого события. Это демонстрирует использование многопоточности и асинхронности при обработке событий.

5. .

Оптимизация и рефакторинг кода - это процессы, направленные на улучшение производительности и читаемости кода. В Python существуют различные методы оптимизации и рефакторинга, включая:

Оптимизация скорости выполнения: Python имеет несколько инструментов для оптимизации скорости выполнения кода, таких как использование функций с замыканиями, использование функций-генераторов, использование модуля Cython для компиляции Python-кода в C-код и использование многопоточности.

Оптимизация использования памяти: Python имеет несколько инструментов для оптимизации использования памяти, таких как использование словарей вместо списков для хранения ключей и значений, использование функций-генераторов вместо списков для

создания больших последовательностей и использование модуля `pumba` для компиляции Python-кода в машинный код.

Рефакторинг кода: Рефакторинг кода - это процесс изменения структуры кода без изменения его функциональности. В Python рефакторинг кода может быть выполнен с помощью таких инструментов, как модуль `unittest` для тестирования кода, модуль `doctest` для проверки документации кода и модуль `pylint` для анализа кода и выявления потенциальных проблем.

Пример кода, иллюстрирующего оптимизацию скорости выполнения:

```
def slow_function(n):
    result = 0
    for i in range(n):
        result += i
    return result
def fast_function(n):
    return sum(range(n))
```

В этом примере функция `slow_function` использует цикл `for` для суммирования чисел от 0 до `n`, что может быть медленным для больших значений `n`. Функция `fast_function` использует встроенную функцию `sum`, которая более эффективна для больших значений `n`. Это демонстрирует оптимизацию скорости выполнения кода.

б. .

При разработке пользовательского интерфейса на Python используются следующие принципы и инструменты:

Использование фреймворков: В Python существует множество фреймворков для разработки пользовательского интерфейса, таких как `Tkinter`, `wxPython`, `PyQt`, `PyGTK` и другие. Фреймворки предоставляют готовые компоненты интерфейса, такие как кнопки, поля ввода, списки и т.д., а также инструменты для их настройки и взаимодействия с пользователем.

Использование шаблонов: Для создания сложных и гибких интерфейсов часто используются шаблоны, такие как `MVC` (Model-View-Controller), `MVVM` (Model-View-ViewModel) и другие. Шаблоны помогают разделить код на логические слои и упрощают поддержку и расширение интерфейса.

Использование библиотек: Для создания более сложных интерфейсов, таких как графические приложения или веб-приложения, используются специализированные библиотеки, такие как `Matplotlib` для создания графиков, `Pygame` для создания игр, `Django` для создания веб-приложений и другие.

Тестирование: Для обеспечения качества и надежности интерфейса используются инструменты тестирования, такие как `Selenium`, `PyTest` и другие. Тестирование помогает выявить ошибки и проблемы в интерфейсе до его выпуска.

Пример кода, иллюстрирующего использование фреймворка `Tkinter` для создания простого интерфейса:

```
from tkinter import Tk, Button
def hello_world():
    print('Привет, мир!')
root = Tk()
button = Button(root, text='Привет, мир!', command=hello_world)
button.pack()
```

```
root.mainloop()
```

В этом примере мы создаем простой интерфейс с кнопкой, которая вызывает функцию `hello_world` при нажатии. Это демонстрирует использование фреймворка Tkinter для создания пользовательского интерфейса на Python.

7. .

При работе с базами данных на Python используются следующие принципы и инструменты:

Использование SQL: Python предоставляет возможность работы с базами данных через SQL. Для этого используются библиотеки, такие как `sqlite3`, `psycopg2`, `MySQLdb` и другие. С их помощью можно выполнять SQL-запросы, изменять данные, создавать и удалять таблицы и т.д.

Использование ORM (Object-Relational Mapping): ORM позволяет работать с базами данных через объекты, а не через SQL-запросы. Это упрощает работу с данными и делает код более читаемым. В Python существуют различные ORM-библиотеки, такие как `SQLAlchemy`, `Django ORM`, `Peewee` и другие.

Использование NoSQL: NoSQL - это альтернативный подход к хранению данных, который не использует SQL. В Python существуют различные NoSQL-библиотеки, такие как `MongoDB`, `CouchDB`, `Redis` и другие. Они предоставляют гибкие возможности для хранения и доступа к данным.

Тестирование: Для обеспечения качества и надежности работы с базами данных используются инструменты тестирования, такие как `SQLAlchemy Testing`, `pytest-sqlalchemy` и другие. Тестирование помогает выявить ошибки и проблемы в работе с базами данных до их выпуска.

Пример кода, иллюстрирующего использование библиотеки `sqlite3` для работы с базой данных:

```
import sqlite3
# Создание подключения к базе данных
conn = sqlite3.connect('my_database.db')
# Создание курсора
c = conn.cursor()
# Создание таблицы
c.execute("""
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    age INTEGER NOT NULL
)
""")
# Вставка данных в таблицу
c.execute("""
INSERT INTO users (name, age)
VALUES (?, ?)
""", ('Иван', 25))
# Закрытие подключения
conn.commit()
conn.close()
```

В этом примере мы создаем подключение к базе данных, создаем таблицу и вставляем данные в нее. Это демонстрирует использование библиотеки `sqlite3` для работы с базами данных на Python.

8. .

Отладка программного обеспечения - это процесс обнаружения, анализа и устранения ошибок, которые могут возникнуть в программном коде во время его выполнения или в процессе разработки. Этот процесс играет ключевую роль в разработке программного обеспечения, поскольку ошибки, называемые багами, могут привести к неправильной работе программы, нежелательным сбоям или даже критическим ошибкам, которые могут повлиять на работу всей системы.

Отладка важна по нескольким причинам:

Обеспечение качества продукта: Отладка позволяет разработчикам обнаруживать и исправлять ошибки в программном коде до его выпуска, что помогает обеспечить высокое качество программного продукта.

Экономия времени и ресурсов: Исправление ошибок на ранних стадиях разработки программы стоит гораздо меньше, чем после выпуска продукта. Отладка помогает предотвратить потерю времени и ресурсов на поиск и исправление ошибок в более поздние этапы разработки или даже после выпуска программы.

Повышение производительности разработчика: Как только ошибка обнаружена и исправлена, разработчик может продолжить работу над проектом без прерываний, что способствует повышению эффективности и производительности.

Улучшение пользовательского опыта: Неправильная работа программы из-за ошибок может негативно сказаться на пользовательском опыте. Отладка помогает обеспечить корректную и плавную работу программы, что в свою очередь повышает удовлетворенность пользователей.

Таким образом, отладка программного обеспечения является важным этапом в процессе разработки, который помогает создать надежные, качественные и безопасные программные продукты.

9. .

В Python существует несколько инструментов, которые помогают разработчикам в процессе отладки программного кода. Вот некоторые из них:

pdb (Python Debugger):

`pdb` - это встроенный отладчик Python, который предоставляет широкий набор инструментов для отслеживания и исправления ошибок в коде. Он позволяет устанавливать точки останова, отслеживать выполнение программы по шагам, анализировать значения переменных и многое другое. `pdb` запускается из командной строки или встроен в интерактивную оболочку Python, что делает его удобным инструментом для отладки на всех этапах разработки.

IDE (интегрированные среды разработки):

Многие популярные IDE для Python, такие как PyCharm, Visual Studio Code, Sublime Text, и другие, предоставляют встроенные инструменты отладки. Они обычно предоставляют графический интерфейс для управления отладкой, а также расширенные функции, такие как просмотр значений переменных, отслеживание стека вызовов и т. д.

pdb++:

`pdb++` - это расширение для `pdb`, которое предоставляет дополнительные функции и улучшенный пользовательский интерфейс для отладки. Он добавляет цветовую подсветку,

автоматическое завершение команд, улучшенный вывод данных и другие удобные функции.

ipdb:

ipdb - это оболочка отладчика, основанная на pdb, но с поддержкой интерактивной среды IPython. Это делает его более мощным инструментом для отладки, так как он предоставляет больше возможностей для работы с интерактивным кодом и объектами.

Эти инструменты обеспечивают разработчикам Python широкий выбор средств для отладки и помогают им эффективно обнаруживать и устранять ошибки в программном коде.

10. .

Модульное тестирование - это методика тестирования программного обеспечения, при которой отдельные компоненты программы (модули) тестируются независимо от остальных частей системы. Цель модульного тестирования - проверить правильность работы отдельных функций, методов или классов в изоляции от других компонентов. Это позволяет выявить и исправить ошибки в коде на ранних этапах разработки, а также облегчает отладку и поддержку программного обеспечения в дальнейшем.

В Python модульное тестирование обычно осуществляется с использованием стандартной библиотеки unittest или более простых и удобных в использовании библиотек, таких как pytest. Вот как применяется модульное тестирование в Python:

Написание тестовых случаев:

Сначала разработчик создает набор тестовых случаев для проверки различных аспектов функциональности модуля. Тестовые случаи определяются как методы в классе тестов, используя функции-ассерты для проверки ожидаемых результатов.

Запуск тестов:

После написания тестов разработчик запускает их с помощью специальных инструментов, таких как unittest или pytest. Эти инструменты автоматически обнаруживают тестовые случаи в коде и выполняют их, предоставляя отчет о прохождении или неудачах каждого теста.

Анализ результатов:

Разработчик анализирует результаты выполнения тестов, чтобы выяснить, были ли найдены какие-либо ошибки или несоответствия в поведении модуля. Если тесты не прошли успешно, это может указывать на наличие ошибок в коде, которые нужно исправить.

Итеративный процесс:

Модульное тестирование является итеративным процессом, который повторяется при каждом изменении кода. После внесения изменений в программу разработчик запускает тесты, чтобы убедиться, что изменения не привели к появлению новых ошибок или не нарушили существующую функциональность.

Модульное тестирование в Python позволяет разработчикам создавать надежное и стабильное программное обеспечение, повышая уверенность в правильной работе кода и ускоряя процесс разработки благодаря раннему обнаружению и исправлению ошибок.

11. .

При тестировании программного обеспечения на Python следует придерживаться нескольких основных принципов, которые помогают обеспечить эффективное и качественное тестирование:

Полное покрытие кода тестами:

Важно убедиться, что каждая строка кода в программе была протестирована хотя бы одним тестовым случаем. Это гарантирует, что все возможные ветви выполнения программы были испытаны и не содержат ошибок.

Тестирование граничных значений:

Необходимо убедиться, что тесты охватывают граничные значения входных данных, такие как минимальные и максимальные значения, пустые строки, нулевые значения и т. д. Это позволяет проверить, как программа обрабатывает экстремальные случаи.

Использование "черного ящика" и "белого ящика":

"Черный ящик" - это тестирование, основанное на функциональных требованиях программы, без знания внутренней реализации. "Белый ящик" - это тестирование, которое учитывает внутреннюю структуру и логику программы. Оба подхода важны для полного покрытия тестами и выявления различных категорий ошибок.

Автоматизация тестирования:

Автоматизация тестирования позволяет повысить эффективность и скорость процесса тестирования, а также уменьшить вероятность человеческих ошибок. Для этого в Python часто используются библиотеки, такие как unittest, pytest, nose и другие.

Постоянное обновление тестов:

Тесты должны регулярно обновляться вместе с изменениями в коде программы. При каждом изменении кода необходимо проверить, не нарушило ли оно существующую функциональность, и внести соответствующие изменения в тесты.

Изоляция тестов:

Тесты должны быть максимально изолированы друг от друга, чтобы ошибка в одном тесте не привела к провалу других тестов. Это достигается путем создания независимых тестовых случаев и использованием моков (mocks) или заглушек (stubs) для изоляции зависимостей.

12. .

Документирование программного обеспечения - это процесс создания и поддержания документации, которая описывает функциональность, структуру, использование и другие аспекты программного продукта. Эта документация может включать в себя различные элементы, такие как комментарии в исходном коде, документационные строки (docstrings), руководства пользователя, спецификации интерфейсов API и т. д.

Документирование программного обеспечения важно по нескольким причинам, особенно для проектов на Python:

Улучшение понимания кода: Четкая и подробная документация позволяет другим разработчикам лучше понимать структуру и логику кода, что упрощает совместную работу и поддержку проекта.

Сокращение времени на обучение: Хорошо документированный код и документация позволяют новым участникам команды быстрее ориентироваться в проекте и начать вносить свой вклад без длительного периода обучения.

Повышение качества кода: Документация может содержать описания алгоритмов, методов реализации и решений, принятых в проекте, что помогает разработчикам принимать информированные решения и избегать ошибок.

Улучшение сопровождаемости и поддержки: Хорошо документированный код проще поддерживать и обновлять в будущем, поскольку новые разработчики или текущие участники команды могут быстро понять его логику и назначение.

Повышение привлекательности проекта: Проекты с хорошей документацией обычно воспринимаются как более профессиональные и надежные, что может привлечь больше пользователей или участников.

13. .

В Python классы и методы документируются с использованием специальных строк, называемых docstrings, которые являются строковыми литералами, расположенными в самом начале определения класса или метода. Документация для классов и методов обычно описывает их назначение, интерфейс, параметры (для методов) и возвращаемые значения.

Пример документации для класса и метода в Python:

```
class MyClass:
    """Описание класса."""
    def my_method(self, param1, param2):
        """Описание метода.
        :param param1: Описание параметра param1.
        :type param1: Тип параметра param1.
        :param param2: Описание параметра param2.
        :type param2: Тип параметра param2.
        :return: Описание возвращаемого значения.
        :rtype: Тип возвращаемого значения.
        """
        # Реализация метода
        pass
```

Отличие документации для классов от документации для функций заключается в том, что документация для класса обычно описывает весь класс в целом, включая его атрибуты, методы, свойства и общее назначение. В то время как документация для методов описывает каждый метод класса, его назначение, параметры и возвращаемые значения.

Основные элементы документации включают:

Описание: Общее описание класса или метода, описывающее его назначение и основные функции.

Параметры: Описание параметров метода (если это метод) с указанием их названия, типа и описания.

Возвращаемые значения: Описание возвращаемых значений метода (если есть) с указанием их типа и описания.

Документирование классов и методов помогает другим разработчикам лучше понимать и использовать ваш код, а также улучшает документацию проекта в целом, что способствует его поддержке и развитию.

14. .

Хорошая документация играет ключевую роль в улучшении совместной работы в команде разработчиков по нескольким причинам:

Понимание кода: Четкая и подробная документация помогает разработчикам быстрее понимать функциональность, логику и структуру кода. Это ускоряет процесс внесения изменений или исправлений, поскольку разработчики могут легко ориентироваться в коде проекта.

Уменьшение зависимости от конкретных разработчиков: Хорошо документированный код позволяет членам команды легко заменять друг друга при необходимости. Наличие подробной документации позволяет новым участникам команды

быстро включиться в работу над проектом, даже если они не имеют опыта работы с ним ранее.

Снижение вероятности ошибок: Документация помогает избежать недопонимания и неправильного использования функций или методов. Это снижает вероятность ошибок и багов, так как разработчики имеют четкое представление о том, как должен использоваться код.

Сокращение времени на обучение: Подробная документация уменьшает время, необходимое для обучения новых членов команды или стажеров. Вместо того, чтобы тратить время на объяснение каждой части кода, разработчики могут использовать документацию для самостоятельного изучения проекта.

Улучшение коммуникации: Документация служит важным инструментом коммуникации между членами команды. Она помогает установить общий язык и согласовать понимание терминов, используемых в проекте, что способствует эффективной командной работе.

В целом, хорошая документация существенно улучшает эффективность и качество совместной работы в команде разработчиков, способствуя более быстрой разработке, лучшему пониманию кода и повышению уровня взаимодействия внутри команды.

15. .

Организация документации проекта на Python таким образом, чтобы она была легко доступной и понятной для других разработчиков и пользователей, играет важную роль в успешной разработке и использовании программного обеспечения. Вот несколько рекомендаций по организации документации проекта:

Используйте стандартные средства документирования Python:

Используйте стандартные инструменты документирования Python, такие как docstrings для функций, классов и методов, а также модульные docstrings для модулей. Это позволит автоматически генерировать документацию с помощью инструментов типа Sphinx.

Используйте форматирование docstrings в стиле reStructuredText или Markdown:

Хотя docstrings могут содержать обычные строки, структурирование и форматирование их с помощью языков разметки, таких как reStructuredText или Markdown, делает документацию более читабельной и удобной для использования. Это особенно полезно при генерации документации с помощью инструментов, таких как Sphinx.

Используйте инструменты для генерации документации:

Используйте инструменты автоматической генерации документации, такие как Sphinx, чтобы создать структурированную и профессионально оформленную документацию для вашего проекта. Sphinx позволяет создавать различные форматы документации, включая HTML, PDF и другие, что делает ее доступной для различных типов пользователей.

Организируйте документацию по разделам:

Структурируйте документацию вашего проекта по разделам, включая общее описание проекта, руководство пользователя, API-спецификацию, примеры использования и т. д. Это поможет пользователям и разработчикам быстрее найти нужную информацию.

Обеспечьте легкий доступ к документации:

Убедитесь, что документация проекта легко доступна для всех участников команды разработчиков и пользователей. Разместите ее в репозитории проекта на GitHub или другой

платформе контроля версий, создайте ссылки на документацию на веб-сайте проекта и обновляйте ее регулярно.

Поддерживайте актуальность документации:

Регулярно обновляйте документацию проекта, чтобы отражать текущее состояние проекта и любые изменения, внесенные в код или функциональность. Это поможет избежать недопонимания и недочетов при использовании вашего программного обеспечения.

16. .

Основные платформы, используемые для разработки мобильных приложений, включают:

iOS (Apple):

iOS - мобильная операционная система, разработанная Apple Inc. для своих устройств, таких как iPhone и iPad. Для разработки приложений под iOS используется интегрированная среда разработки (IDE) Xcode, а язык программирования - Swift или Objective-C. Преимущества iOS включают высокий уровень безопасности, качество аппаратных компонентов устройств Apple и консистентность пользовательского интерфейса.

Android (Google):

Android - мобильная операционная система, разработанная Google, используемая на широком спектре устройств, от смартфонов и планшетов до умных часов и ТВ-приставок. Для разработки приложений под Android используется Android Studio как основная IDE, а языки программирования включают Java и Kotlin. Преимущества Android включают широкое распространение устройств, гибкость настройки и доступ к множеству API.

Cross-platform (Кросс-платформенные решения):

Кросс-платформенные инструменты позволяют разработчикам создавать приложения, которые могут работать на нескольких платформах с помощью общего кода. Некоторые из популярных кросс-платформенных инструментов включают Xamarin (C#), React Native (JavaScript), Flutter (Dart) и PhoneGap (HTML, CSS, JavaScript). Преимущества кросс-платформенной разработки включают повышение производительности, экономию времени и ресурсов, а также упрощение сопровождения.

Основные различия между этими платформами включают в себя язык программирования, используемый для разработки, доступные инструменты и среды разработки, экосистему и сторонние библиотеки, а также специфические требования и ограничения каждой платформы. Выбор платформы зависит от целей проекта, аудитории, бюджета и других факторов.

17. .

Для разработки мобильных приложений под различные платформы используются различные языки программирования. Вот основные языки программирования и их преимущества и недостатки:

Swift (iOS):

Преимущества:

Разработан Apple специально для iOS и macOS.

Отличная интеграция с платформой и фреймворками.

Безопасный и высокопроизводительный язык.

Недостатки:

Ограниченность в выборе платформы (только для iOS и macOS).

Отсутствие обратной совместимости с Objective-C.

Objective-C (iOS):

Преимущества:

Долгое время использовался для разработки iOS-приложений.

Широкая база знаний и опыт разработчиков.

Обратная совместимость с Swift.

Недостатки:

Сложный и менее читаемый синтаксис по сравнению с современными языками.

Более низкая производительность по сравнению с Swift.

Java (Android):

Преимущества:

Широкое распространение и большое сообщество разработчиков.

Кроссплатформенность: Java может использоваться не только для Android, но и для разработки серверных приложений.

Множество инструментов и библиотек.

Недостатки:

Производительность: приложения на Java могут быть менее производительными по сравнению с некоторыми другими языками.

Долгий процесс разработки из-за многослойной архитектуры и длинного цикла разработки.

Kotlin (Android):

Преимущества:

Официальный язык разработки под Android, разработанный JetBrains.

Совместим с Java, что облегчает миграцию существующих проектов.

Более безопасный и удобный синтаксис по сравнению с Java.

Недостатки:

Меньшее количество материалов и опыта разработчиков по сравнению с Java.

Некоторые библиотеки или инструменты могут быть менее совместимы с Kotlin.

JavaScript (React Native):

Преимущества:

Кроссплатформенность: приложения, разработанные с использованием React Native, могут работать как на iOS, так и на Android.

Большое сообщество разработчиков и множество доступных библиотек и компонентов.

Быстрая разработка благодаря использованию JavaScript.

Недостатки:

Некоторые функции и API могут быть недоступны на нативном уровне.

Производительность приложений может быть несколько ниже, чем у нативных приложений.

Dart (Flutter):

Преимущества:

Высокая производительность благодаря компиляции в нативный код.

Кроссплатформенность: приложения, разработанные на Flutter, могут работать как на iOS, так и на Android.

Горячая перезагрузка: быстрая итерация разработки.

Недостатки:

Меньшее количество сторонних библиотек и компонентов по сравнению с некоторыми другими фреймворками.

Не такая широкая поддержка от сторонних разработчиков и компаний, как у некоторых других фреймворков.

Каждый из этих языков имеет свои уникальные особенности, преимущества и недостатки, и выбор зависит от требований проекта, целевой аудитории, опыта разработчиков и других факторов.

18. .

Кросс-платформенная разработка мобильных приложений - это подход, при котором разработчики используют общий код для создания приложений, которые могут работать на различных мобильных платформах, таких как iOS и Android. Это позволяет ускорить процесс разработки, снизить затраты и обеспечить единообразие внешнего вида и поведения приложений на разных платформах.

Для кросс-платформенной разработки мобильных приложений существует несколько инструментов и фреймворков. Вот некоторые из них:

React Native:

React Native - это фреймворк, разработанный Facebook, который позволяет создавать мобильные приложения с использованием JavaScript и React. Он позволяет разработчикам использовать общий код для создания приложений для iOS и Android, сохраняя при этом нативный внешний вид и производительность.

Flutter:

Flutter - это фреймворк, разработанный Google, который позволяет создавать красивые и высокопроизводительные мобильные приложения с использованием одного и того же кода на языке программирования Dart. Flutter использует свою собственную визуальную библиотеку и движок рендеринга, что обеспечивает высокую производительность и кроссплатформенную совместимость.

Xamarin:

Xamarin - это фреймворк, при помощи которого разработчики могут создавать мобильные приложения на C# и .NET, используя общий код для различных платформ, включая iOS, Android и Windows. Он обеспечивает нативный доступ к API и библиотекам платформы, что позволяет создавать приложения с высокой производительностью и нативным внешним видом.

Ionic:

Ionic - это фреймворк, основанный на веб-технологиях, таких как HTML, CSS и JavaScript, который позволяет создавать гибридные мобильные приложения, работающие на различных платформах с помощью общего кода. Он использует веб-технологии для создания пользовательского интерфейса и доступа к устройственным функциям через Apache Cordova или Capacitor.

PhoneGap / Apache Cordova:

PhoneGap (также известный как Apache Cordova) - это инструмент, который позволяет разработчикам создавать мобильные приложения с использованием веб-технологий, таких как HTML, CSS и JavaScript, и упаковывать их в нативные приложения для различных платформ.

19. .

При выборе платформы и языка разработки для мобильного приложения следует учитывать ряд факторов, которые могут существенно повлиять на процесс разработки и конечный результат. Вот основные факторы, которые стоит учитывать:

Целевая аудитория:

Распространенность платформы среди целевой аудитории.

Предпочтения пользователей относительно операционных систем.

Функциональные требования:

Необходимость использования специфических функций и возможностей, доступных только на определенной платформе.

Возможность интеграции с аппаратными компонентами устройства (камера, сенсоры, NFC и т. д.).

Технические ограничения:

Требования к производительности и эффективности приложения.

Ограничения API и доступные возможности разработки на каждой платформе.

Опыт разработчиков:

Знание и опыт работы с определенными языками программирования и фреймворками.

Готовность и возможности команды разработчиков работать с выбранной платформой и технологиями.

Сроки и бюджет проекта:

Время, доступное для разработки приложения.

Затраты на обучение, поддержку и разработку на разных платформах.

Поддержка и сообщество:

Наличие активного сообщества разработчиков, доступных ресурсов и инструментов для поддержки разработки на выбранной платформе.

Выбор правильной платформы и языка разработки должен основываться на анализе этих факторов с учетом требований и целей проекта. Неправильный выбор может привести к затянутому процессу разработки, увеличенным затратам и недовольству пользователей. Следует также учитывать возможность расширения приложения на другие платформы в будущем и обеспечить его устойчивость к изменениям в рыночной среде.

20. .

При проектировании модулей для мобильных приложений на Python следует соблюдать ряд основных принципов, чтобы обеспечить их эффективность и масштабируемость:

Принцип разделения ответственности (Single Responsibility Principle):

Каждый модуль должен отвечать за выполнение одной конкретной задачи или функции. Это позволяет уменьшить сложность кода, облегчить его понимание и обеспечить легкость тестирования и поддержки.

Принцип модульности (Modularity):

Разделение приложения на небольшие модули с четко определенными интерфейсами позволяет легко изменять и заменять отдельные компоненты без необходимости переписывания всего приложения. Это также способствует повторному использованию кода и упрощает параллельную разработку.

Принцип абстракции (Abstraction):

Использование абстракций позволяет скрыть детали реализации модулей за интерфейсами или высокоуровневыми API, обеспечивая простоту использования и возможность замены компонентов без изменения кода, который их использует.

Принцип независимости (Independence):

Модули должны быть независимыми друг от друга, чтобы изменения в одном модуле не приводили к неожиданным последствиям в других частях приложения. Это обеспечивает гибкость и устойчивость к изменениям.

Принцип простоты (Simplicity):

Стремитесь к простоте и ясности в проектировании модулей, избегая излишней сложности и избыточной функциональности. Простые модули легче понимать, тестировать и поддерживать.

Принцип тестируемости (Testability):

Модули должны быть легко тестируемыми, что обеспечивает быструю обратную связь и повышает надежность приложения. Используйте мокирование и инъекцию зависимостей для создания изолированных тестов.

Принцип расширяемости (Extensibility):

Сделайте модули расширяемыми, чтобы их можно было легко адаптировать и расширять для удовлетворения новых требований. Используйте интерфейсы, плагины и шаблоны проектирования для обеспечения гибкости и возможности добавления новой функциональности.

21. .

Использование абстракций и интерфейсов является важным аспектом проектирования модулей на Python, поскольку это позволяет упростить процесс создания, тестирования и поддержки кода. Вот как использование абстракций и интерфейсов может упростить процесс:

Упрощение сложности:

Абстракции и интерфейсы позволяют скрыть сложность внутренней реализации модуля, предоставляя простой и понятный интерфейс для взаимодействия с ним. Это делает код более читаемым, понятным и легко поддерживаемым.

Разделение ответственности:

Использование абстракций и интерфейсов помогает разделить функциональность модуля на независимые компоненты, каждый из которых выполняет свою определенную задачу. Это способствует уменьшению связанности и повышению модульности, что делает код более гибким и масштабируемым.

Облегчение тестирования:

Абстракции и интерфейсы облегчают тестирование, поскольку они позволяют создавать макеты и модули для тестирования отдельных компонентов модуля независимо от его реальной реализации. Это помогает проводить модульное тестирование более эффективно и обеспечивает большую уверенность в корректности работы модуля.

Пример использования абстракций и интерфейсов в Python:

```
from abc import ABC, abstractmethod
# Абстрактный класс для модуля работы с базой данных
class DatabaseModule(ABC):
    @abstractmethod
    def connect(self):
        pass
```

```

    @abstractmethod
    def query(self, sql):
        pass
# Конкретная реализация модуля работы с базой данных MySQL
class MySQLDatabase(DatabaseModule):
    def connect(self):
        # Логика подключения к базе данных MySQL
        pass
    def query(self, sql):
        # Логика выполнения SQL-запроса в MySQL
        pass
# Конкретная реализация модуля работы с базой данных PostgreSQL
class PostgreSQLDatabase(DatabaseModule):
    def connect(self):
        # Логика подключения к базе данных PostgreSQL
        pass
    def query(self, sql):
        # Логика выполнения SQL-запроса в PostgreSQL
        Pass

```

В этом примере DatabaseModule является абстрактным классом, определяющим интерфейс для работы с базой данных. Классы MySQLDatabase и PostgreSQLDatabase представляют конкретные реализации этого интерфейса для работы с MySQL и PostgreSQL соответственно. При этом все они обеспечивают одинаковый интерфейс для взаимодействия с базой данных, что позволяет легко подменять одну реализацию другой и проводить тестирование кода с использованием моков и макетов.

22. .

Для обнаружения ошибок и обеспечения надежности мобильного приложения на Python рекомендуется использовать следующие методы тестирования модулей:

Модульное тестирование (Unit Testing):

Модульное тестирование проверяет отдельные компоненты (модули) приложения на корректность и соответствие требованиям.

Тестирование проводится независимо от других компонентов, что упрощает обнаружение и изоляцию ошибок.

Для Python существует богатый инструментарий для модульного тестирования, включая встроенный модуль unittest и сторонние библиотеки, такие как pytest.

Интеграционное тестирование (Integration Testing):

Интеграционное тестирование проверяет взаимодействие между различными компонентами приложения, чтобы убедиться, что они работают вместе корректно.

Этот вид тестирования помогает обнаруживать ошибки, которые могут возникнуть из-за неправильной интеграции компонентов.

Для мобильных приложений на Python интеграционные тесты могут включать тестирование API, взаимодействие с базой данных, обработку пользовательского ввода и т. д.

Приемочное тестирование (Acceptance Testing):

Приемочное тестирование проверяет, соответствует ли приложение функциональным требованиям и ожиданиям пользователей.

Тесты позволяют убедиться, что приложение выполняет свои основные функции и обеспечивает требуемый пользовательский опыт.

Для Python мобильных приложений можно использовать инструменты автоматизации тестирования пользовательского интерфейса, такие как Appium или PyAutoGUI.

Стресс-тестирование (Stress Testing):

Стресс-тестирование оценивает производительность и стабильность приложения при высокой нагрузке или экстремальных условиях использования.

Этот вид тестирования помогает обнаруживать узкие места в приложении, такие как утечки памяти, зависания или сбои.

Для Python мобильных приложений можно использовать инструменты для нагрузочного тестирования, такие как Locust или Apache JMeter.

Эти методы тестирования позволяют обеспечить высокую надежность и качество мобильного приложения на Python, обнаруживая и исправляя ошибки на разных этапах разработки. Комбинация этих методов позволяет достичь максимального покрытия тестами и уверенности в работоспособности приложения перед выпуском в продакшн.

23. .

Интеграционное тестирование модулей играет важную роль в обеспечении общего качества мобильного приложения на Python. Вот несколько способов, как интеграционное тестирование влияет на качество приложения и как его можно проводить эффективно:

Обнаружение интеграционных проблем: Интеграционные тесты позволяют проверить взаимодействие между различными компонентами приложения и выявить проблемы, такие как неправильная передача данных, некорректная обработка событий или неправильное взаимодействие с внешними сервисами.

Проверка работы системы в целом: Интеграционные тесты оценивают, как все компоненты приложения работают вместе в реальной среде. Это позволяет убедиться, что приложение в целом соответствует функциональным требованиям и ожиданиям пользователей.

Улучшение стабильности и надежности: Проведение интеграционного тестирования помогает выявить и устранить проблемы, которые могут привести к сбоям или неправильной работе приложения. Это способствует повышению стабильности и надежности приложения.

Предотвращение негативных последствий: Обнаружение и исправление проблем на ранних этапах разработки с помощью интеграционного тестирования позволяет избежать негативных последствий, таких как ошибки в продакшн, потерю данных или негативный опыт пользователей.

Для эффективного проведения интеграционного тестирования модулей мобильного приложения на Python следует учитывать следующие рекомендации:

Планирование тестов: Определите, какие компоненты и взаимодействия нужно протестировать. Разработайте тестовые сценарии и наборы данных для проверки различных аспектов интеграции.

Автоматизация: Автоматизируйте интеграционные тесты, чтобы повысить эффективность и повторяемость процесса. Используйте фреймворки для автоматизации тестирования, такие как pytest или unittest.

Изоляция тестов: убедитесь, что каждый интеграционный тест проверяет конкретное взаимодействие или компонент, и что тесты не зависят друг от друга. Изолируйте тесты, чтобы обеспечить надежность и устойчивость.

Мониторинг результатов: внимательно отслеживайте результаты интеграционного тестирования и реагируйте на обнаруженные проблемы. Фиксируйте и анализируйте любые ошибки, чтобы предотвратить их повторение.

Эффективное проведение интеграционного тестирования помогает повысить качество мобильного приложения на Python и обеспечить безопасную и стабильную работу в реальной среде.

24. .

Язык низкого уровня - это язык программирования, который предоставляет непосредственное или близкое к непосредственному управление аппаратным обеспечением компьютера. Вот основные характеристики, определяющие язык низкого уровня:

Прямое управление ресурсами: Язык низкого уровня позволяет программисту манипулировать ресурсами компьютера, такими как память, процессор и периферийные устройства, на более прямом уровне, чем языки высокого уровня.

Близкое к машинному коду: Программы, написанные на языках низкого уровня, обычно ближе к машинному коду, который понимает процессор, чем программы, написанные на языках высокого уровня. Это обеспечивает большую гибкость и контроль, но часто требует более тщательного управления памятью и ресурсами.

Минимальный уровень абстракции: Языки низкого уровня предоставляют минимальное количество абстракций, что позволяет программистам иметь прямой доступ к аппаратным ресурсам и реализовывать операции на более низком уровне, чем в языках высокого уровня.

Python, хотя и является языком программирования высокого уровня, имеет ряд особенностей, связанных с программированием на языке низкого уровня:

Возможность написания расширений: Python предоставляет возможность написания расширений на низкоуровневых языках, таких как C или C++, и интеграции их с кодом на Python. Это позволяет использовать низкоуровневые библиотеки и функции в Python-программах для выполнения специфических задач.

Библиотеки для работы с низкоуровневыми аспектами: Python обладает широким набором библиотек и модулей, которые обеспечивают возможность работы с низкоуровневыми аспектами операционной системы, такими как файловая система, сетевые интерфейсы и управление процессами.

Использование Cython и ctypes: Cython позволяет писать C-подобный код и компилировать его в расширения Python, а ctypes позволяет вызывать функции из динамических библиотек напрямую из Python. Эти возможности позволяют программистам использовать функциональность низкоуровневых библиотек и операционной системы в Python-программах.

Хотя Python не является языком низкого уровня в привычном понимании, он обладает средствами для работы с низкоуровневыми аспектами и интеграции с низкоуровневыми языками, что позволяет разработчикам использовать его в широком спектре приложений, включая те, которые требуют более низкоуровневого доступа к аппаратным ресурсам.

25. .

Python, будучи высокоуровневым языком программирования, обычно скрывает от программиста многие низкоуровневые детали, такие как работа с указателями и управление памятью, которые часто характерны для языков низкого уровня, таких как C или C++. Тем не менее, Python предоставляет некоторые механизмы, которые позволяют работать с низкоуровневыми аспектами:

Работа с указателями:

Python не предоставляет прямой доступ к памяти и указателям, как это делают языки низкого уровня. Однако, модуль `ctypes` позволяет взаимодействовать с библиотеками, написанными на языке C, и использовать указатели для работы с памятью через функции и структуры данных, определенные в этих библиотеках.

Работа с памятью:

Python автоматически управляет памятью через механизм сборки мусора, что освобождает программиста от необходимости вручную выделять и освобождать память, как это делается в низкоуровневых языках. Однако, для некоторых задач, требующих явного управления памятью, таких как работа с большими объемами данных или интеграция с библиотеками на языке C, Python предоставляет модуль `ctypes` или возможность написания расширений на C с использованием API Python C API.

Модуль `ctypes`:

Модуль `ctypes` в Python предоставляет возможность вызывать функции и работать с данными, определенными в библиотеках на языке C. Он позволяет создавать обертки для функций и структур данных, определенных в C, и обеспечивает способ обмена данными между Python и C.

Хотя Python не является языком низкого уровня и не предоставляет прямого доступа к низкоуровневым конструкциям и функциям, как это делают языки низкого уровня, его возможности модуля `ctypes` и возможность написания расширений на C позволяют программистам взаимодействовать с низкоуровневыми аспектами, когда это необходимо.

26. .

Python предоставляет широкий спектр инструментов и библиотек для взаимодействия с низкоуровневыми аспектами операционной системы. Вот некоторые из них:

Модуль `os`: предоставляет функции для взаимодействия с операционной системой, такие как создание и удаление файлов, выполнение команд, управление процессами и многое другое.

Модуль `sys`: предоставляет доступ к низкоуровневым системным функциям и переменным, таким как аргументы командной строки, пути поиска модулей и другие системные параметры.

Модуль `subprocess`: позволяет запускать внешние процессы и взаимодействовать с ними через стандартные потоки ввода, вывода и ошибок.

Модуль `socket`: предоставляет возможности для работы с сетевыми сокетами, что позволяет создавать клиент-серверные приложения, обмениваться данными по сети и т. д.

Модуль `threading` и `multiprocessing`: позволяют создавать и управлять потоками и процессами в Python, что полезно для параллельного выполнения задач, взаимодействия между компонентами приложения и др.

Модуль `asyncio`: предоставляет инструменты для асинхронного программирования, включая возможность создания сетевых серверов и клиентов с использованием асинхронных сокетов.

Модули для ввода-вывода данных: например, модуль `io` для работы с потоками данных, модуль `pathlib` для работы с путями к файлам и каталогам, а также модули `pickle` и `json` для сериализации и десериализации данных.

Эти инструменты и библиотеки позволяют программистам Python взаимодействовать с различными низкоуровневыми аспектами операционной системы, такими как файловая система, сетевые интерфейсы и управление процессами, обеспечивая при этом удобный и высокоуровневый интерфейс.

27. .

Python предоставляет различные инструменты и библиотеки для взаимодействия с низкоуровневыми аспектами операционной системы. Вот некоторые из них:

`os` и `shutil`: Модуль `os` предоставляет функции для взаимодействия с операционной системой, такие как создание и удаление файлов, доступ к каталогам, выполнение команд в командной строке и другие. Модуль `shutil` предоставляет более высокоуровневый интерфейс для работы с файлами и каталогами.

`subprocess`: Модуль `subprocess` предоставляет возможность запуска внешних процессов из Python и взаимодействия с ними через их стандартные потоки ввода/вывода.

`socket`: Модуль `socket` позволяет создавать сетевые соединения и взаимодействовать с сетевыми интерфейсами операционной системы, включая создание сокетов, установление соединений TCP/UDP и т. д.

`multiprocessing`: Модуль `multiprocessing` предоставляет высокоуровневый интерфейс для создания и управления параллельными процессами, что позволяет использовать несколько ядер процессора и улучшить производительность приложения.

`asyncio`: Модуль `asyncio` предоставляет асинхронный ввод/вывод и возможность создания сетевых серверов и клиентов с использованием событийных циклов, что позволяет эффективно обрабатывать множество соединений без создания отдельного потока для каждого из них.

`ctypes`: Модуль `ctypes` позволяет вызывать функции из библиотек на языке C из Python и работать с данными, определенными в этих библиотеках. Это позволяет взаимодействовать с низкоуровневыми аспектами операционной системы, доступными через функции C API.

Эти инструменты и библиотеки позволяют программистам Python взаимодействовать с низкоуровневыми аспектами операционной системы, такими как файловая система, сетевые интерфейсы и управление процессами, и создавать мощные и гибкие приложения.

28. .

Использование Python для работы с низкоуровневыми задачами имеет свои преимущества и недостатки по сравнению с языками низкого уровня, такими как C или C++. Вот некоторые из них:

Преимущества использования Python:

Простота и удобство: Python имеет простой и понятный синтаксис, что делает его легко читаемым и понятным даже для начинающих программистов. Это позволяет разработчикам быстрее создавать и поддерживать код.

Большое количество библиотек и инструментов: Python имеет обширную экосистему библиотек и инструментов для решения различных задач, включая работу с сетями, базами данных, веб-разработку, машинное обучение и многое другое. Это позволяет быстрее разрабатывать и развертывать приложения.

Управление памятью: Python обеспечивает автоматическое управление памятью через механизм сборки мусора, что избавляет программиста от необходимости вручную выделять и освобождать память. Это делает разработку более безопасной и удобной.

Быстрая разработка: благодаря своей простоте и множеству готовых инструментов, разработка приложений на Python может быть быстрой и эффективной, что особенно полезно для прототипирования и итеративной разработки.

Недостатки использования Python:

Низкая производительность: Python обычно медленнее по сравнению с языками низкого уровня, такими как C или C++. Это связано с интерпретируемой природой языка и дополнительными слоями абстракции, которые могут замедлять выполнение кода.

Ограниченный доступ к системным вызовам и ресурсам: Python, будучи высокоуровневым языком, скрывает от программиста некоторые низкоуровневые детали, такие как доступ к системным вызовам или управление аппаратным обеспечением, что может быть недостатком в некоторых сценариях использования.

Зависимость от внешних библиотек и модулей: в некоторых случаях для работы с низкоуровневыми задачами может потребоваться использование сторонних библиотек или расширений, написанных на языке C или C++, что может усложнить развертывание и управление зависимостями.

Ограниченные возможности для оптимизации: Python, будучи интерпретируемым языком, обычно имеет ограниченные возможности для оптимизации кода по сравнению с языками, компилируемыми в машинный код, такими как C или C++. Это может привести к увеличению времени выполнения приложения в некоторых случаях.

Выбор между использованием Python и языков низкого уровня зависит от конкретных требований проекта, а также от уровня производительности, безопасности, скорости разработки и других факторов, которые нужно учитывать при принятии решения.

29. .

Написание низкоуровневого кода на Python может потребоваться в следующих сценариях использования:

Взаимодействие с системными библиотеками или API:

Например, если требуется написать программу, которая взаимодействует с операционной системой, управляет процессами или использует специфические функции операционной системы, низкоуровневый код на Python может использоваться для вызова соответствующих системных вызовов или функций API.

Работа с сетевыми протоколами:

Например, написание сетевых клиентов или серверов с использованием сокетов, особенно при работе с протоколами низкого уровня, такими как TCP/IP или UDP.

Интеграция с библиотеками на языке C/C++:

Например, если требуется использовать функциональность, доступную только через библиотеки на языке C/C++, низкоуровневый код на Python может использоваться для создания оберток (bindings), позволяющих использовать эти библиотеки из Python.

Выполнение задач, требующих высокой производительности или оптимизации:

Например, обработка больших объемов данных, реализация алгоритмов с высокой вычислительной сложностью, работа с мультимедийными данными (обработка изображений, звука, видео) и т. д.

Работа с аппаратным обеспечением:

Например, управление периферийными устройствами через системные вызовы или библиотеки на языке C/C++, написание драйверов устройств или программирование микроконтроллеров.

Рассмотрим конкретные примеры сценариев использования, которые могут потребовать написания низкоуровневого кода на Python:

Разработка системного утилитарного приложения:

Например, создание утилиты для мониторинга ресурсов операционной системы, которая может взаимодействовать с системными вызовами для получения информации о процессах, использовании CPU и памяти, а также для управления службами.

Работа с сетевыми протоколами:

Например, написание сетевого сервера для обработки данных в реальном времени, работающего с протоколами низкого уровня, такими как TCP/IP или UDP, чтобы обеспечить эффективное общение между клиентами и сервером.

Обработка изображений и видео:

Например, реализация алгоритмов компьютерного зрения для обнаружения объектов на изображениях или для анализа видеопотока. Это может потребовать использования специализированных библиотек или API для работы с изображениями и видео.

Работа с базами данных и хранилищами данных:

Например, написание библиотеки для работы с распределенной базой данных или файловой системой, которая требует взаимодействия с низкоуровневыми механизмами хранения данных и управления файлами.

Интеграция с аппаратным обеспечением:

Например, написание драйверов для работы с периферийными устройствами, такими как принтеры, сканеры, считыватели карт или датчики, для обеспечения их взаимодействия с приложением на Python.

Разработка высокопроизводительных вычислительных приложений:

Например, реализация алгоритмов машинного обучения или обработки больших данных, которые требуют использования оптимизированных структур данных и алгоритмов для эффективной обработки больших объемов данных.

30. .

При использовании Python для выполнения задач, требующих низкоуровневого программирования, следует учитывать несколько основных предосторожностей и рекомендаций:

Производительность:

Python является интерпретируемым языком, что может привести к недостаточной производительности при выполнении вычислительно интенсивных задач. Важно учитывать производительность кода и применять оптимизации, такие как использование более эффективных алгоритмов, использование встроенных функций и структур данных из стандартной библиотеки Python, а также использование специализированных библиотек или расширений на языке C/C++.

Управление памятью:

Python автоматически управляет памятью через механизм сборки мусора, что упрощает разработку, но может привести к неэффективному использованию памяти в некоторых случаях. При работе с большими объемами данных или при интенсивном

использовании памяти важно следить за утечками памяти и избегать создания большого количества ненужных объектов.

Ограничения интерпретатора:

Интерпретатор Python имеет ограничения на использование ресурсов, таких как память и процессорное время. Необходимо учитывать эти ограничения при проектировании и написании кода, чтобы избежать ситуаций, когда приложение работает медленно или может потребовать неожиданных прерываний из-за превышения лимитов.

Использование сторонних библиотек и расширений:

Для выполнения низкоуровневых задач на Python часто требуется использование сторонних библиотек или расширений на языке C/C++. Важно учитывать совместимость, производительность и безопасность этих библиотек, а также следить за обновлениями и поддержкой.

Тестирование и отладка:

Низкоуровневый код обычно требует более тщательного тестирования и отладки из-за его более прямого воздействия на операционную систему и аппаратное обеспечение. Важно проводить обширное тестирование различных аспектов приложения и использовать инструменты для отладки, чтобы обнаружить и исправить ошибки.

Безопасность:

Использование низкоуровневого кода может представлять риски для безопасности, такие как уязвимости в коде из-за ошибок в управлении памятью или несанкционированного доступа к ресурсам системы. Важно применять лучшие практики безопасного программирования и проводить аудит безопасности для обнаружения и устранения потенциальных уязвимостей.

3.3. Практико-ориентированные задания к экзамену:

1. Создайте модуль, который принимает изображение в качестве входных данных, применяет к нему фильтр (например, ч/б, сепия, негатив) и возвращает измененное изображение.
2. Напишите модуль, который обеспечивает соединение с базой данных (например, SQLite), добавление, удаление и запрос данных из таблицы.
3. Создайте модуль, который анализирует текстовый файл, подсчитывает количество слов, находит самые часто встречающиеся слова и удаляет стоп-слова (например, артикли, предлоги).
4. Напишите модуль для взаимодействия с открытым API (например, API Twitter, GitHub или OpenWeatherMap). Модуль должен позволять выполнять запросы к API, получать данные и обрабатывать ответы.
5. Разработайте модуль, который осуществляет поиск файлов с определенным расширением в указанной директории, копирует их в другую директорию и создает архив.
6. Напишите модуль, который обеспечивает аутентификацию пользователя (например, через ввод логина и пароля) и доступ к защищенным ресурсам (например, страницам веб-приложения) с использованием базы данных.
7. Создайте модуль Python для обработки изображений, который предоставляет функции для изменения размера, поворота, наложения фильтров и других манипуляций с изображениями. Модуль должен поддерживать форматы изображений JPEG, PNG и GIF.
8. Создайте модуль Python для автоматической генерации отчетов на основе данных из базы данных или других источников. Модуль должен поддерживать настройку формата отчета, включая заголовки, таблицы, графики и диаграммы.

9. Напишите модуль для манипуляции файлами и каталогами в файловой системе. Модуль должен включать функции для чтения и записи файлов, создания и удаления каталогов, а также переименования и перемещения файлов.
10. Напишите модуль, который принимает список элементов и сортирует его по заданному критерию (например, числовому значению, длине строки) с использованием встроенной функции сортировки Python или алгоритма сортировки, написанного вручную.
11. Напишите модульные тесты для основных функций вашего программного модуля, чтобы убедиться в их корректной работе. Используйте библиотеки, такие как unittest или pytest.
12. Протестируйте программный модуль на крайних случаях, таких как пустые входные данные, максимальные или минимальные значения, чтобы убедиться, что программа корректно обрабатывает такие ситуации.
13. Убедитесь, что ваш программный модуль работает корректно на разных версиях Python (например, Python 2 и Python 3). Протестируйте его на разных версиях и внесите необходимые изменения для совместимости.
14. Создайте тесты для проверки обработки ошибок в вашем программном модуле. Убедитесь, что программа корректно реагирует на ошибки и исключения, возникающие в процессе работы.
15. Проведите тестирование производительности программного модуля для определения времени выполнения основных операций. Оптимизируйте код при необходимости для улучшения производительности.
16. Ваш программный модуль имеет пользовательский интерфейс, протестируйте его на наличие ошибок и проверьте удобство использования. Убедитесь, что все элементы интерфейса работают корректно.
17. Проверьте взаимодействие вашего программного модуля с внешними системами, такими как базы данных, веб-сервисы или API. Убедитесь, что данные правильно передаются и обрабатываются.
18. Создайте мобильное приложение на Python, которое функционирует как простой калькулятор, выполняющий основные арифметические операции (сложение, вычитание, умножение, деление).
19. Напишите мобильное приложение, которое позволяет пользователям создавать, редактировать и удалять задачи, а также отмечать их как выполненные.
20. Разработайте мобильное приложение, которое отображает галерею изображений, позволяя пользователям просматривать и управлять изображениями на своем устройстве.
21. Создайте мобильное приложение, которое позволяет пользователям вести список покупок, добавлять новые элементы, отмечать их как купленные и удалять.
22. Напишите мобильное приложение для создания и редактирования заметок, обеспечивая возможность сохранения, редактирования и удаления заметок пользователями.
23. Разработайте мобильное приложение для обмена сообщениями между пользователями, обеспечивая отправку текстовых сообщений, возможность прикрепления файлов и управление контактами.
24. Создайте приложение на Python для выполнения системных задач, таких как управление процессами, мониторинг ресурсов или администрирование файловой системы.
25. Напишите модуль, который анализирует и фильтрует записи из системных журналов (например, syslog) для выявления определенных событий или ошибок.
26. Разработайте модуль для мониторинга использования ресурсов системы, таких как

процессорное время, память и сетевой трафик, с возможностью предоставления отчетов или уведомлений о превышении пороговых значений.

27. Напишите модуль, который автоматически создает резервные копии файлов и директорий на заданном расписании и сохраняет их на внешнем носителе или в облаке.

28. Разработайте модуль, который непрерывно мониторит работоспособность системы и при обнаружении проблемных ситуаций отправляет уведомления администратору по электронной почте или SMS.

29. Создайте приложение, которое реализует сетевой протокол для взаимодействия между несколькими устройствами, например, для передачи данных между сервером и клиентом.

30. Разработайте приложение, которое автоматически проверяет наличие обновлений для установленного программного обеспечения и устанавливает их по мере необходимости. Эталоны ответов на практические задания:

1.

```
from PIL import Image, ImageOps
def apply_filter(image_path, filter_name):
    Аргументы:
    image_path (str): Путь к исходному изображению.
    filter_name (str): Название фильтра ('ч/б', 'сепия', 'негатив').
    # Открываем изображение
    image = Image.open(image_path)
    # Применяем фильтр к изображению
    if filter_name == 'ч/б':
        image = ImageOps.grayscale(image)
    elif filter_name == 'сепия':
        image = ImageOps.colorize(image.convert('L'), '#704214', '#C0C090')
    elif filter_name == 'негатив':
        image = ImageOps.invert(image)
    else:
        raise ValueError("Неподдерживаемый фильтр. Доступные фильтры:
'ч/б', 'сепия', 'негатив'")
    return image
# Пример использования
if __name__ == "__main__":
    input_image_path = "test.jpeg"
    output_image_path = "test2.jpeg"
    filter_name = "ч/б" # Можно изменить на 'сепия' или 'негатив'
    filtered_image = apply_filter(input_image_path, filter_name)
    filtered_image.show() # Показываем измененное изображение
    filtered_image.save(output_image_path) # Сохраняем измененное
изображение
```

2.

```
import sqlite3
class DatabaseManager:
```

```

def __init__(self, db_name):

    self.db_name = db_name
    self.conn = None
def connect(self):

    self.conn = sqlite3.connect(self.db_name)
def disconnect(self):
    if self.conn:
        self.conn.close()
def execute_query(self, query):

    if not self.conn:
        raise Exception("Соединение с базой данных не установлено.")
    cursor = self.conn.cursor()
    cursor.execute(query)
    result = cursor.fetchall()
    cursor.close()
    return result
def execute_update(self, query):

    if not self.conn:
        raise Exception("Соединение с базой данных не установлено.")
    cursor = self.conn.cursor()
    cursor.execute(query)
    self.conn.commit()
    cursor.close()
def create_table(self, table_name, columns):
    query = f"CREATE TABLE IF NOT EXISTS {table_name}
({columns})"
    self.execute_update(query)
def insert_data(self, table_name, values):
    query = f"INSERT INTO {table_name} VALUES {values}"
    self.execute_update(query)
def delete_data(self, table_name, condition):

    query = f"DELETE FROM {table_name} WHERE {condition}"
    self.execute_update(query)
def select_data(self, table_name, columns="*", condition=None):
    query = f"SELECT {columns} FROM {table_name}"
    if condition:
        query += f" WHERE {condition}"
    return self.execute_query(query)
# Пример использования
if __name__ == "__main__":

```

```

db_manager = DatabaseManager("example.db")
db_manager.connect()
# Создаем таблицу
db_manager.create_table("users", "id INTEGER PRIMARY KEY, name
TEXT, age INTEGER")
# Вставляем данные
db_manager.insert_data("users", "(1, 'Alice', 30)")
db_manager.insert_data("users", "(2, 'Bob', 25)")
# Удаляем данные
db_manager.delete_data("users", "name='Alice'")
# Запрашиваем данные
results = db_manager.select_data("users")
print("Результаты запроса к таблице users:")
for row in results:
    print(row)
db_manager.disconnect()

```

3.

```

import re
from collections import Counter
from nltk.corpus import stopwords
def analyze_text(file_path):
    # Загружаем стоп-слова из библиотеки NLTK
    stop_words = set(stopwords.words('english'))
    # Считываем текст из файла
    with open(file_path, 'r', encoding='utf-8') as file:
        text = file.read()
    # Приводим текст к нижнему регистру и разбиваем на слова
    words = re.findall(r'\b\w+\b', text.lower())
    # Подсчитываем общее количество слов
    total_words = len(words)
    # Удаляем стоп-слова
    words_without_stopwords = [word for word in words if word not in
stop_words]
    # Находим самые часто встречающиеся слова
    word_counts = Counter(words_without_stopwords)
    most_common_words = word_counts.most_common(5) # Топ-5
наиболее часто встречающихся слов
    return {
        'total_words': total_words,
        'most_common_words': most_common_words,
        'words_without_stopwords': words_without_stopwords
    }
# Пример использования
if __name__ == "__main__":
    file_path = "sample_text.txt"

```



```

analysis_results = analyze_text(file_path)
print(f'Общее количество слов: {analysis_results['total_words']}')
print("Наиболее часто встречающиеся слова:")
for word, count in analysis_results['most_common_words']:
    print(f'{word}: {count} раз")
print("Слова без стоп-слов:")
print(analysis_results['words_without_stopwords'])

```

4.

```

class APIClient:
    def __init__(self, api_key=None):

        self.api_key = api_key
    def request(self, method, url, params=None, headers=None):
        response = requests.request(method, url, params=params,
headers=headers)
        response.raise_for_status() # Проверяем статус ответа
        return response.json()
# Пример использования
if __name__ == "__main__":
    # Пример взаимодействия с API OpenWeatherMap
    openweathermap_api_key = "YOUR_API_KEY" # Введите свой API
ключ OpenWeatherMap
    openweathermap_api_client =
APIClient(api_key=openweathermap_api_key)
    city = "Moscow"
    url =
f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={open
weathermap_api_key}"
    response = openweathermap_api_client.request("GET", url)
    print(f"Погода в городе {city}:")
    print(response)

```

5.

```

import os
import shutil
import zipfile
def search_and_copy_files(source_dir, target_dir, extension):
    # Проверяем, существует ли целевая директория, и создаем ее, если
она не существует
    if not os.path.exists(target_dir):
        os.makedirs(target_dir)
    # Создаем список файлов с указанным расширением в исходной
директории
    files_to_copy = [file for file in os.listdir(source_dir) if
file.endswith(extension)]
    # Копируем файлы в целевую директорию

```

```

for file in files_to_copy:
    source_path = os.path.join(source_dir, file)
    target_path = os.path.join(target_dir, file)
    shutil.copy(source_path, target_path)
# Создаем ZIP-архив из скопированных файлов
zip_file_path = os.path.join(target_dir, f"archive_{extension}.zip")
with zipfile.ZipFile(zip_file_path, 'w') as zipf:
    for file in files_to_copy:
        file_path = os.path.join(target_dir, file)
        zipf.write(file_path, os.path.basename(file_path))
return zip_file_path
# Пример использования
if __name__ == "__main__":
    source_directory = "source"
    target_directory = "target"
    file_extension = ".txt"
    archive_path = search_and_copy_files(source_directory,
target_directory, file_extension)
print(f'Архив успешно создан: {archive_path}')

```

6.

```

class Authenticator:
    def __init__(self, db_name):
        self.db_name = db_name
    def create_table(self):
        with sqlite3.connect(self.db_name) as conn:
            cursor = conn.cursor()
            cursor.execute("""
                CREATE TABLE IF NOT EXISTS users (
                    id INTEGER PRIMARY KEY,
                    username TEXT UNIQUE,
                    password TEXT
                )
            """)
            conn.commit()
    def register_user(self, username, password):
        with sqlite3.connect(self.db_name) as conn:
            cursor = conn.cursor()
            try:
                cursor.execute("INSERT INTO users (username, password)
VALUES (?, ?)", (username, password))
                conn.commit()
                return True
            except sqlite3.IntegrityError:
                # Пользователь с таким логином уже существует
                return False

```

```

def authenticate(self, username, password):
    with sqlite3.connect(self.db_name) as conn:
        cursor = conn.cursor()
        cursor.execute("SELECT * FROM users WHERE username = ?
AND password = ?", (username, password))
        return cursor.fetchone() is not None
# Пример использования
if __name__ == "__main__":
    db_name = "users.db"
    authenticator = Authenticator(db_name)
    authenticator.create_table()
    # Регистрация нового пользователя
    username = input("Введите логин: ")
    password = input("Введите пароль: ")
    if authenticator.register_user(username, password):
        print("Регистрация прошла успешно.")
    else:
        print("Пользователь с таким логином уже существует.")
    # Аутентификация пользователя
    username = input("Введите логин: ")
    password = input("Введите пароль: ")
    if authenticator.authenticate(username, password):
        print("Аутентификация прошла успешно. Доступ разрешен.")
        # Здесь можно предоставить доступ к защищенным ресурсам
    else:
        print("Неверный логин или пароль. Доступ запрещен.")

```

7.

```

from PIL import Image, ImageFilter
class ImageProcessor:
    @staticmethod
    def resize_image(image_path, output_path, size):

        with Image.open(image_path) as img:
            resized_img = img.resize(size)
            resized_img.save(output_path)
    @staticmethod
    def rotate_image(image_path, output_path, angle):

        with Image.open(image_path) as img:
            rotated_img = img.rotate(angle)
            rotated_img.save(output_path)
    @staticmethod
    def apply_filter(image_path, output_path, filter_name):

        with Image.open(image_path) as img:

```

```

        filtered_img = img.filter(getattr(ImageFilter, filter_name))
        filtered_img.save(output_path)
# Пример использования
if __name__ == "__main__":
    input_image_path = "input_image.jpg"
    output_image_path = "output_image.jpg"
    # Изменяем размер изображения
    ImageProcessor.resize_image(input_image_path, output_image_path,
(300, 200))
    # Поворачиваем изображение
    ImageProcessor.rotate_image(output_image_path, output_image_path,
90)
    # Накладываем фильтр на изображение
ImageProcessor.apply_filter(output_image_path, output_image_path, "BLUR")

```

8.

```

import pandas as pd
import matplotlib.pyplot as plt
class ReportGenerator:
    @staticmethod
    def generate_report(data, report_format='text'):

        if report_format == 'text':
            return data.to_string()
        elif report_format == 'html':
            return data.to_html()
        elif report_format == 'pdf':
            # Для генерации PDF-отчета можно использовать другие
библиотеки, такие как pdfkit или reportlab
            raise NotImplementedError("PDF report generation is not
implemented yet.")
        else:
            raise ValueError("Invalid report format specified.")
    @staticmethod
    def generate_chart(data, chart_type='bar', title=None, xlabel=None,
ylabel=None):

        if chart_type not in ['bar', 'line', 'pie']:
            raise ValueError("Invalid chart type specified.")
        if chart_type == 'bar':
            data.plot(kind='bar')
        elif chart_type == 'line':
            data.plot(kind='line')
        elif chart_type == 'pie':
            data.plot(kind='pie', subplots=True, figsize=(8, 4))
        plt.title(title)

```

```

plt.xlabel(xlabel)
plt.ylabel(ylabel)
plt.show()
# Пример использования
if __name__ == "__main__":
    # Создаем DataFrame с данными
    data = pd.DataFrame({
        'City': ['Moscow', 'Novosibirsk', 'Khabarovsk', 'Vladivostok',
'Dalnegorsk'],
        'Population': [8398748, 3990456, 2705994, 2325502, 1680992]
    })
    # Генерируем текстовый отчет
    text_report = ReportGenerator.generate_report(data, report_format='text')
    print("Текстовый отчет:")
    print(text_report)
    # Генерируем HTML-отчет
    html_report = ReportGenerator.generate_report(data,
report_format='html')
    with open("report.html", "w") as f:
        f.write(html_report)
    print("HTML-отчет сгенерирован в файле 'report.html'")
    # Строим столбчатую диаграмму
    ReportGenerator.generate_chart(data, chart_type='bar', title='Population
by City', xlabel='City', ylabel='Population')

```

9.

```

import os
import shutil
class FileSystemManager:
    @staticmethod
    def read_file(file_path):
        with open(file_path, 'r') as file:
            return file.read()
    @staticmethod
    def write_file(file_path, content):
        with open(file_path, 'w') as file:
            file.write(content)
    @staticmethod
    def create_directory(directory_path):
        os.makedirs(directory_path, exist_ok=True)
    @staticmethod
    def remove_directory(directory_path):
        shutil.rmtree(directory_path)

```

```

    @staticmethod
    def rename_file(old_file_path, new_file_path):

        os.rename(old_file_path, new_file_path)
    @staticmethod
    def move_file(source_file_path, destination_file_path):

        shutil.move(source_file_path, destination_file_path)
# Пример использования
if __name__ == "__main__":
    # Чтение содержимого файла
    file_content = FileSystemManager.read_file("example.txt")
    print("Содержимое файла example.txt:")
    print(file_content)
    # Запись в файл
    new_content = "Новое содержимое для файла"
    FileSystemManager.write_file("example.txt", new_content)
    print("Содержимое файла example.txt после записи:")
    print(FileSystemManager.read_file("example.txt"))
    # Создание каталога
    FileSystemManager.create_directory("new_directory")
    # Переименование файла
    FileSystemManager.rename_file("example.txt",
    "new_directory/renamed_example.txt")
    print("Содержимое файла после переименования:")

    print(FileSystemManager.read_file("new_directory/renamed_example.txt"))
    # Перемещение файла
    FileSystemManager.move_file("new_directory/renamed_example.txt",
    "renamed_example.txt")
    print("Содержимое файла после перемещения:")
    print(FileSystemManager.read_file("renamed_example.txt"))
    # Удаление каталога
    FileSystemManager.remove_directory("new_directory")

```

10.

```

class Sorter:
    @staticmethod
    def sort_list(data, key=None, reverse=False):
        return sorted(data, key=key, reverse=reverse)
# Пример использования
if __name__ == "__main__":
    # Список элементов для сортировки
    data = ["apple", "banana", "orange", "grape", "kiwi"]
    # Сортировка в алфавитном порядке
    sorted_data = Sorter.sort_list(data)

```

```
print("Сортировка в алфавитном порядке:", sorted_data)
# Сортировка по длине строки
sorted_data = Sorter.sort_list(data, key=len)
print("Сортировка по длине строки:", sorted_data)
```

11.

```
import unittest
from sorter import Sorter
class TestSorter(unittest.TestCase):
    def test_sort_list_alphabetical(self):
        # Подготовка данных
        data = ["banana", "apple", "grape", "kiwi", "orange"]
        expected_result = ["apple", "banana", "grape", "kiwi", "orange"]
        # Выполнение операции
        sorted_data = Sorter.sort_list(data)
        # Проверка результата
        self.assertEqual(sorted_data, expected_result)
    def test_sort_list_by_length(self):
        # Подготовка данных
        data = ["banana", "apple", "grape", "kiwi", "orange"]
        expected_result = ["kiwi", "apple", "grape", "banana", "orange"]
        # Выполнение операции
        sorted_data = Sorter.sort_list(data, key=len)
        # Проверка результата
        self.assertEqual(sorted_data, expected_result)
    def test_sort_list_reverse(self):
        # Подготовка данных
        data = ["banana", "apple", "grape", "kiwi", "orange"]
        expected_result = ["orange", "kiwi", "grape", "banana", "apple"]
        # Выполнение операции
        sorted_data = Sorter.sort_list(data, reverse=True)
        # Проверка результата
        self.assertEqual(sorted_data, expected_result)
if __name__ == "__main__":
    unittest.main()
```

12.

```
class TestSorterEdgeCases(unittest.TestCase):
    def test_sort_list_empty(self):
        # Подготовка данных
        data = []
        expected_result = []
        # Выполнение операции
        sorted_data = Sorter.sort_list(data)
        # Проверка результата
        self.assertEqual(sorted_data, expected_result)
    def test_sort_list_single_element(self):
```

```

# Подготовка данных
data = ["apple"]
expected_result = ["apple"]
# Выполнение операции
sorted_data = Sorter.sort_list(data)
# Проверка результата
self.assertEqual(sorted_data, expected_result)
def test_sort_list_identical_elements(self):
# Подготовка данных
data = [3, 3, 3, 3, 3]
expected_result = [3, 3, 3, 3, 3]
# Выполнение операции
sorted_data = Sorter.sort_list(data)
# Проверка результата
self.assertEqual(sorted_data, expected_result)
if __name__ == "__main__":
    unittest.main()

```

13.

```

import sys
if sys.version_info[0] < 3:
    # Python 2
    from __future__ import print_function
class Sorter:
    @staticmethod
    def sort_list(data, key=None, reverse=False):
        if sys.version_info[0] < 3:
            # Python 2
            return sorted(data, cmp=key, reverse=reverse)
        else:
            # Python 3
            return sorted(data, key=key, reverse=reverse)
# Пример использования
if __name__ == "__main__":
    # Список элементов для сортировки
    data = ["banana", "apple", "grape", "kiwi", "orange"]
    # Сортировка в алфавитном порядке
    sorted_data = Sorter.sort_list(data)
    print("Сортировка в алфавитном порядке:", sorted_data)

```

14.

```

class TestSorterErrorHandling(unittest.TestCase):
    def test_sort_list_invalid_key(self):
        # Подготовка данных
        data = ["banana", "apple", "grape", "kiwi", "orange"]
        # Выполнение операции с недопустимой функцией ключа
        сортировки

```



```

        with self.assertRaises(TypeError):
            Sorter.sort_list(data, key="invalid_key")
def test_sort_list_invalid_data_type(self):
    # Попытка сортировки данных неподходящего типа
    with self.assertRaises(TypeError):
        Sorter.sort_list(123)
if __name__ == "__main__":
    unittest.main()

```

15.

```

import timeit
setup_code = ""
from sorter import Sorter
data = ["banana", "apple", "grape", "kiwi", "orange"] * 1000
# Измерение времени выполнения сортировки в алфавитном порядке
alphabetical_time = timeit.timeit(stmt='Sorter.sort_list(data)',
setup=setup_code, number=1000)
print("Время выполнения сортировки в алфавитном порядке:",
alphabetical_time)
# Измерение времени выполнения сортировки по длине строки
length_time = timeit.timeit(stmt='Sorter.sort_list(data, key=len)',
setup=setup_code, number=1000)
print("Время выполнения сортировки по длине строки:", length_time)
# Измерение времени выполнения сортировки в обратном порядке
reverse_time = timeit.timeit(stmt='Sorter.sort_list(data, reverse=True)',
setup=setup_code, number=1000)
print("Время выполнения сортировки в обратном порядке:",
reverse_time)

```

16.

```

import time
from sorter import Sorter
def measure_performance(func, data):
    start_time = time.time()
    func(data)
    end_time = time.time()
    return end_time - start_time
data = ["banana", "apple", "grape", "kiwi", "orange"] * 1000
# Измерение времени выполнения сортировки в алфавитном порядке
alphabetical_time = measure_performance(Sorter.sort_list, data)
print("Время выполнения сортировки в алфавитном порядке:",
alphabetical_time)
# Измерение времени выполнения сортировки по длине строки
length_time = measure_performance(lambda d: Sorter.sort_list(d, key=len),
data)
print("Время выполнения сортировки по длине строки:", length_time)
# Измерение времени выполнения сортировки в обратном порядке

```

```
reverse_time = measure_performance(lambda d: Sorter.sort_list(d,
reverse=True), data)
print("Время выполнения сортировки в обратном порядке:",
reverse_time)
```

17.

```
import sqlite3
from sorter import Sorter
# Создаем соединение с базой данных SQLite
conn = sqlite3.connect('test.db')
cursor = conn.cursor()
# Создаем таблицу
cursor.execute("CREATE TABLE IF NOT EXISTS items (name TEXT,
value INTEGER)")
# Вставляем некоторые данные
cursor.execute("INSERT INTO items (name, value) VALUES (?, ?)",
('apple', 10))
cursor.execute("INSERT INTO items (name, value) VALUES (?, ?)",
('orange', 5))
cursor.execute("INSERT INTO items (name, value) VALUES (?, ?)",
('banana', 20))
# Сохраняем изменения
conn.commit()
# Получаем данные из базы данных
cursor.execute("SELECT name FROM items")
data = [row[0] for row in cursor.fetchall()]
# Закрываем соединение
conn.close()
# Сортируем данные
sorted_data = Sorter.sort_list(data)
# Выводим отсортированные данные
print("Отсортированные данные:", sorted_data)
```

18.

```
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
from kivy.uix.button import Button
from kivy.uix.textinput import TextInput
class CalculatorApp(App):
    def build(self):
        self.expression = ""
        self.result = TextInput(multiline=False, readonly=True, font_size=40,
halign="right")
        buttons = [
            ['7', '8', '9', '/'],
            ['4', '5', '6', '*'],
            ['1', '2', '3', '-'],
```

```

        ['.', '0', '=', '+']
    ]
    layout = BoxLayout(orientation='vertical')
    for row in buttons:
        h_layout = BoxLayout()
        for button in row:
            h_layout.add_widget(Button(text=button,
on_press=self.on_button_press))
        layout.add_widget(h_layout)
    layout.add_widget(self.result)
    return layout
def on_button_press(self, instance):
    if instance.text == '=':
        try:
            self.result.text = str(eval(self.expression))
        except:
            self.result.text = "Error"
    else:
        self.expression += instance.text
        self.result.text = self.expression
if __name__ == '__main__':
    CalculatorApp().run()

```

19.

```

import sqlite3
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
from kivy.uix.button import Button
from kivy.uix.label import Label
from kivy.uix.textinput import TextInput
class TaskManagerApp(App):
    def build(self):
        self.conn = sqlite3.connect('tasks.db')
        self.cursor = self.conn.cursor()
        self.create_table()
        self.task_input = TextInput(hint_text='Enter task', multiline=False)
        self.add_button = Button(text='Add Task', on_press=self.add_task)
        self.task_list_layout = BoxLayout(orientation='vertical')
        self.refresh_task_list()
        layout = BoxLayout(orientation='vertical')
        layout.add_widget(self.task_input)
        layout.add_widget(self.add_button)
        layout.add_widget(self.task_list_layout)
        return layout
    def create_table(self):

```

```

        self.cursor.execute("CREATE TABLE IF NOT EXISTS tasks (id
INTEGER PRIMARY KEY, task TEXT, done INTEGER)")
        self.conn.commit()
    def refresh_task_list(self):
        self.task_list_layout.clear_widgets()
        self.cursor.execute("SELECT * FROM tasks")
        tasks = self.cursor.fetchall()
        for task in tasks:
            task_label = Label(text=task[1], size_hint_y=None, height=40)
            delete_button = Button(text='Delete', size_hint_x=None, width=100,
on_press=self.delete_task)
            delete_button.task_id = task[0]
            done_button = Button(text='Done' if task[2] == 0 else 'Undone',
size_hint_x=None, width=100, on_press=self.toggle_done)
            done_button.task_id = task[0]
            task_layout = BoxLayout(orientation='horizontal')
            task_layout.add_widget(task_label)
            task_layout.add_widget(delete_button)
            task_layout.add_widget(done_button)
            self.task_list_layout.add_widget(task_layout)
    def add_task(self, instance):
        task = self.task_input.text.strip()
        if task:
            self.cursor.execute("INSERT INTO tasks (task, done) VALUES (?,
?)", (task, 0))
            self.conn.commit()
            self.task_input.text = ""
            self.refresh_task_list()
    def delete_task(self, instance):
        task_id = instance.task_id
        self.cursor.execute("DELETE FROM tasks WHERE id=?", (task_id,))
        self.conn.commit()
        self.refresh_task_list()
    def toggle_done(self, instance):
        task_id = instance.task_id
        self.cursor.execute("SELECT done FROM tasks WHERE id=?",
(task_id,))
        done = self.cursor.fetchone()[0]
        new_done = 1 if done == 0 else 0
        self.cursor.execute("UPDATE tasks SET done=? WHERE id=?",
(new_done, task_id))
        self.conn.commit()
        self.refresh_task_list()
    def on_stop(self):
        self.conn.close()

```

```
if __name__ == '__main__':  
    TaskManagerApp().run()
```

20.

```
import os  
from kivy.app import App  
from kivy.uix.boxlayout import BoxLayout  
from kivy.uix.button import Button  
from kivy.uix.image import Image  
from kivy.uix.scrollview import ScrollView  
class GalleryApp(App):  
    def build(self):  
        self.image_folder = "/path/to/your/image/folder" # Путь к папке с  
изображениями  
        self.image_layout = BoxLayout(orientation='vertical')  
        self.load_images()  
        scroll_view = ScrollView()  
        scroll_view.add_widget(self.image_layout)  
        return scroll_view  
    def load_images(self):  
        for root, dirs, files in os.walk(self.image_folder):  
            for file in files:  
                if file.endswith((''.jpg', '.png', '.jpeg')):  
                    image_path = os.path.join(root, file)  
                    image = Image(source=image_path, size_hint_y=None,  
height=200)  
                    delete_button = Button(text='Delete', size_hint=(None, None))  
                    delete_button.bind(on_press=lambda instance,  
path=image_path: self.delete_image(path))  
                    image_layout = BoxLayout(orientation='horizontal')  
                    image_layout.add_widget(image)  
                    image_layout.add_widget(delete_button)  
                    self.image_layout.add_widget(image_layout)  
    def delete_image(self, image_path):  
        os.remove(image_path)  
        self.image_layout.clear_widgets()  
        self.load_images()  
if __name__ == '__main__':  
    GalleryApp().run()
```

21.

```
from kivy.app import App  
from kivy.uix.boxlayout import BoxLayout  
from kivy.uix.button import Button  
from kivy.uix.label import Label  
from kivy.uix.textinput import TextInput  
class ShoppingListApp(App):
```

```

def build(self):
    self.shopping_list = []
    self.item_input = TextInput(hint_text='Enter item', multiline=False)
    self.add_button = Button(text='Add Item', on_press=self.add_item)
    self.shopping_list_layout = BoxLayout(orientation='vertical')
    self.refresh_shopping_list()
    layout = BoxLayout(orientation='vertical')
    layout.add_widget(self.item_input)
    layout.add_widget(self.add_button)
    layout.add_widget(self.shopping_list_layout)
    return layout

def refresh_shopping_list(self):
    self.shopping_list_layout.clear_widgets()
    for item in self.shopping_list:
        item_label = Label(text=item, size_hint_y=None, height=40)
        delete_button = Button(text='Delete', size_hint_x=None, width=100,
on_press=self.delete_item)
        delete_button.item = item
        buy_button = Button(text='Bought', size_hint_x=None, width=100,
on_press=self.mark_as_bought)
        buy_button.item = item
        item_layout = BoxLayout(orientation='horizontal')
        item_layout.add_widget(item_label)
        item_layout.add_widget(delete_button)
        item_layout.add_widget(buy_button)
        self.shopping_list_layout.add_widget(item_layout)

def add_item(self, instance):
    item = self.item_input.text.strip()
    if item:
        self.shopping_list.append(item)
        self.item_input.text = ""
        self.refresh_shopping_list()

def delete_item(self, instance):
    item = instance.item
    self.shopping_list.remove(item)
    self.refresh_shopping_list()

def mark_as_bought(self, instance):
    item = instance.item
    self.shopping_list[self.shopping_list.index(item)] = f"{item} (Bought)"
    self.refresh_shopping_list()

if __name__ == '__main__':
    ShoppingListApp().run()

```

22.

```

from kivy.app import App
from kivy.uix.boxlayout import BoxLayout

```

```

from kivy.uix.button import Button
from kivy.uix.label import Label
from kivy.uix.textinput import TextInput
class NoteApp(App):
    def build(self):
        self.notes = []
        self.note_input = TextInput(hint_text='Enter note', multiline=True)
        self.save_button = Button(text='Save Note', on_press=self.save_note)
        self.note_list_layout = BoxLayout(orientation='vertical')
        self.refresh_note_list()
        layout = BoxLayout(orientation='vertical')
        layout.add_widget(self.note_input)
        layout.add_widget(self.save_button)
        layout.add_widget(self.note_list_layout)
        return layout
    def refresh_note_list(self):
        self.note_list_layout.clear_widgets()
        for index, note in enumerate(self.notes):
            note_label = Label(text=note, size_hint_y=None, height=100,
halign='left', valign='top')
            delete_button = Button(text='Delete', size_hint_x=None, width=100,
on_press=self.delete_note)
            delete_button.index = index
            edit_button = Button(text='Edit', size_hint_x=None, width=100,
on_press=self.edit_note)
            edit_button.index = index
            note_layout = BoxLayout(orientation='horizontal')
            note_layout.add_widget(note_label)
            note_layout.add_widget(edit_button)
            note_layout.add_widget(delete_button)
            self.note_list_layout.add_widget(note_layout)
    def save_note(self, instance):
        note = self.note_input.text.strip()
        if note:
            self.notes.append(note)
            self.note_input.text = ""
            self.refresh_note_list()
    def delete_note(self, instance):
        index = instance.index
        del self.notes[index]
        self.refresh_note_list()
    def edit_note(self, instance):
        index = instance.index
        self.note_input.text = self.notes[index]
        del self.notes[index]

```

```
        self.refresh_note_list()
if __name__ == '__main__':
    NoteApp().run()
```

23.

```
import os
import shutil
def list_processes():
    return [(pid, name) for pid, name in zip(os.listdir('/proc'),
os.listdir('/proc')) if pid.isdigit()]
def kill_process(pid):
    try:
        os.kill(int(pid), signal.SIGKILL)
        return True
    except OSError:
        return False
def list_files(directory):
    return os.listdir(directory)
def delete_file(file_path):
    if os.path.exists(file_path):
        os.remove(file_path)
        return True
    return False
def copy_file(source, destination):
    if os.path.exists(source):
        shutil.copy2(source, destination)
        return True
    return False
def move_file(source, destination):
    if os.path.exists(source):
        shutil.move(source, destination)
        return True
    return False
if __name__ == '__main__':
    while True:
        print("\nSelect an option:")
        print("1. List running processes")
        print("2. Kill a process")
        print("3. List files in a directory")
        print("4. Delete a file")
        print("5. Copy a file")
        print("6. Move a file")
        print("7. Exit")
        choice = input("Enter your choice: ")
        if choice == '1':
            processes = list_processes()
```



```

if processes:
    print("Running processes:")
    for pid, name in processes:
        print(f"PID: {pid}, Name: {name}")
else:
    print("No running processes found")
elif choice == '2':
    pid = input("Enter the PID of the process to kill: ")
    if kill_process(pid):
        print(f"Process with PID {pid} killed successfully")
    else:
        print(f"Failed to kill process with PID {pid}")
elif choice == '3':
    directory = input("Enter the directory path to list files: ")
    files = list_files(directory)
    if files:
        print(f"Files in directory {directory}:")
        for file in files:
            print(file)
    else:
        print("No files found in the directory")
elif choice == '4':
    file_path = input("Enter the path of the file to delete: ")
    if delete_file(file_path):
        print(f"File {file_path} deleted successfully")
    else:
        print(f"Failed to delete file {file_path}")
elif choice == '5':
    source = input("Enter the source file path: ")
    destination = input("Enter the destination directory path: ")
    if copy_file(source, destination):
        print(f"File {source} copied to {destination} successfully")
    else:
        print(f"Failed to copy file {source} to {destination}")
elif choice == '6':
    source = input("Enter the source file path: ")
    destination = input("Enter the destination directory path: ")
    if move_file(source, destination):
        print(f"File {source} moved to {destination} successfully")
    else:
        print(f"Failed to move file {source} to {destination}")
elif choice == '7':
    print("Exiting...")
    break
else:

```

```
print("Invalid choice. Please try again.")
```

24.

```
import os
import psutil
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
from kivy.uix.button import Button
from kivy.uix.label import Label
class SystemApp(App):
    def build(self):
        layout = BoxLayout(orientation='vertical')
        # Панель с информацией о процессах
        processes_label = Label(text="Processes:")
        layout.add_widget(processes_label)
        self.processes_layout = BoxLayout(orientation='vertical')
        layout.add_widget(self.processes_layout)
        # Панель с информацией о текущей директории
        current_dir_label = Label(text="Current Directory:")
        layout.add_widget(current_dir_label)
        self.current_dir_label = Label(text=os.getcwd())
        layout.add_widget(self.current_dir_label)
        # Кнопки для управления процессами
        kill_button = Button(text="Kill Process", on_press=self.kill_process)
        layout.add_widget(kill_button)
        refresh_button = Button(text="Refresh", on_press=self.refresh)
        layout.add_widget(refresh_button)
        self.refresh()
        return layout
    def refresh(self, *args):
        # Обновление информации о процессах
        self.processes_layout.clear_widgets()
        for process in psutil.process_iter(['pid', 'name']):
            self.processes_layout.add_widget(Label(text=str(process.info)))
    def kill_process(self, *args):
        # Завершение выбранного процесса
        pass # В этом месте необходимо добавить логику для завершения
выбранного процесса
if __name__ == '__main__':
    SystemApp().run()
```

25.

```
import re
import syslog
def filter_log(file_path, pattern):
    filtered_logs = []
    try:
```

```

with open(file_path, 'r') as file:
    for line in file:
        if re.search(pattern, line):
            filtered_logs.append(line.strip())
except FileNotFoundError:
    print("File not found.")
return filtered_logs
if __name__ == "__main__":
    log_file_path = "/var/log/syslog" # Путь к журнальному файлу
    search_pattern = r'ERROR|FAILED' # Паттерн для поиска
    filtered_logs = filter_log(log_file_path, search_pattern)
    for log in filtered_logs:
        print(log)

```

26.

```

import psutil
import smtplib
from email.mime.text import MIMEText
class ResourceMonitor:
    def __init__(self, threshold_cpu=80, threshold_memory=80,
threshold_network=1024):
        self.threshold_cpu = threshold_cpu # Пороговое значение
использования CPU (%)
        self.threshold_memory = threshold_memory # Пороговое значение
использования памяти (%)
        self.threshold_network = threshold_network # Пороговое значение
сетевого трафика (байт)
    def check_usage(self):
        cpu_usage = psutil.cpu_percent()
        memory_usage = psutil.virtual_memory().percent
        network_traffic = psutil.net_io_counters().bytes_sent +
psutil.net_io_counters().bytes_recv
        if cpu_usage > self.threshold_cpu:
            self.send_notification(f"High CPU usage detected: {cpu_usage}%")
        if memory_usage > self.threshold_memory:
            self.send_notification(f"High memory usage detected:
{memory_usage}%")
        if network_traffic > self.threshold_network:
            self.send_notification(f"High network traffic detected:
{network_traffic} bytes")
    def send_notification(self, message):
        sender_email = "your_email@gmail.com"
        recipient_email = "recipient_email@gmail.com"
        password = "your_password"
        msg = MIMEText(message)
        msg['Subject'] = 'Resource Monitor Alert'

```

```

msg['From'] = sender_email
msg['To'] = recipient_email
try:
    server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
    server.login(sender_email, password)
    server.sendmail(sender_email, recipient_email, msg.as_string())
    server.quit()
    print("Notification sent successfully!")
except Exception as e:
    print(f"Failed to send notification: {str(e)}")
if __name__ == "__main__":
    monitor = ResourceMonitor()
    monitor.check_usage()

```

27.

```

import shutil
import os
import schedule
import time
from datetime import datetime
from dropbox import Dropbox
from dotenv import load_dotenv
load_dotenv()
class BackupManager:
    def __init__(self, source_path, destination_path):
        self.source_path = source_path
        self.destination_path = destination_path
    def backup_files(self):
        timestamp = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
        backup_folder = os.path.join(self.destination_path,
f"backup_{timestamp}")
        try:
            shutil.copytree(self.source_path, backup_folder)
            print(f"Backup created successfully at {backup_folder}")
            self.upload_to_cloud(backup_folder)
        except Exception as e:
            print(f"Failed to create backup: {str(e)}")
    def upload_to_cloud(self, backup_folder):
        try:
            dbx = Dropbox(os.getenv("DROPBOX_ACCESS_TOKEN"))
            with open(backup_folder, "rb") as f:
                dbx.files_upload(f.read(), f"/{os.path.basename(backup_folder)}")
            print("Backup uploaded to Dropbox successfully")
        except Exception as e:
            print(f"Failed to upload backup to Dropbox: {str(e)}")
if __name__ == "__main__":

```

```

source_path = "/path/to/source"
destination_path = "/path/to/destination"
backup_manager = BackupManager(source_path, destination_path)
# Установка расписания для создания резервной копии каждый день
в 00:00
schedule.every().day.at("00:00").do(backup_manager.backup_files)
while True:
    schedule.run_pending()
    time.sleep(60) # Проверка расписания каждую минуту

```

28.

```

import psutil
import smtplib
from email.mime.text import MIMEText
class SystemMonitor:
    def __init__(self, threshold_cpu=80):
        self.threshold_cpu = threshold_cpu # Порог загрузки CPU (%)
    def check_system(self):
        cpu_usage = psutil.cpu_percent(interval=1)
        if cpu_usage > self.threshold_cpu:
            self.send_notification(f"High CPU usage detected: {cpu_usage}%")
    def send_notification(self, message):
        sender_email = "your_email@gmail.com"
        recipient_email = "recipient_email@gmail.com"
        password = "your_password"
        msg = MIMEText(message)
        msg['Subject'] = 'System Monitor Alert'
        msg['From'] = sender_email
        msg['To'] = recipient_email
        try:
            server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
            server.login(sender_email, password)
            server.sendmail(sender_email, recipient_email, msg.as_string())
            server.quit()
            print("Notification sent successfully!")
        except Exception as e:
            print(f"Failed to send notification: {str(e)}")
if __name__ == "__main__":
    monitor = SystemMonitor()
    while True:
        monitor.check_system()

```

29.

```

Сервер:
import socket
HOST = '127.0.0.1' # Локальный адрес сервера
PORT = 65432 # Порт, на котором сервер прослушивает подключения

```

```

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as
server_socket:
    server_socket.bind((HOST, PORT))
    server_socket.listen()
    print("Сервер запущен и ожидает подключения...")
    conn, addr = server_socket.accept()
    with conn:
        print(f"Подключен клиент {addr}")
        while True:
            data = conn.recv(1024)
            if not data:
                break
            print(f"Получено от клиента: {data.decode()}")
            response = input("Ответ сервера: ")
            conn.sendall(response.encode())

```

Клиент:

```

import socket
HOST = '127.0.0.1' # Локальный адрес сервера
PORT = 65432 # Порт, на котором сервер прослушивает подключения
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as
client_socket:
    client_socket.connect((HOST, PORT))
    while True:
        message = input("Введите сообщение для сервера: ")
        client_socket.sendall(message.encode())
        data = client_socket.recv(1024)
        print(f"Ответ от сервера: {data.decode()}")

```

30.

```

import subprocess
def check_for_updates():
    try:
        # Проверяем наличие обновлений
        subprocess.run(["apt", "update"])
        # Проверяем доступные обновления
        result = subprocess.run(["apt", "list", "--upgradable"],
capture_output=True, text=True)
        updates = result.stdout.splitlines()[1:] # Пропускаем первую строку с
заголовком
        if updates:
            print("Доступны следующие обновления:")
            for update in updates:
                print(update)
            install_updates = input("Хотите установить обновления? (y/n): ")
            if install_updates.lower() == 'y':
                subprocess.run(["apt", "upgrade", "-y"])

```

```

        print("Обновления успешно установлены.")
    else:
        print("Обновления не установлены.")
    else:
        print("Нет доступных обновлений.")
except Exception as e:
    print(f"Ошибка при проверке обновлений: {e}")
if __name__ == "__main__":
    check_for_updates()

```

Оценка запланированных результатов по профессиональному модулю

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У 1 Осуществлять разработку кода программного модуля на языках низкого и высокого уровней ОК 1, ОК 02, ОК 4, ОК 05 ОК 9, ПК 1.2, ПК 1.6.	определяет необходимый язык программирования; проводит разработку кода программного модуля.
У 2 Создавать программу по разработанному алгоритму как отдельный модуль ОК 1, ОК 2, ОК 03, ОК 4, ОК 9, ПК 1.1, ПК 1.2, ПК 1.6.	разрабатывает или выбирает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма с возможностью параллельного использования модуля в других программах.
У 3 Выполнять отладку и тестирование программы на уровне модуля ОК 1, ОК 2, ОК 05, ОК 9, ПК 1.3, ПК 1.4	определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы как на уровне модуля, так и на уровне всего программного комплекса.
У 4 Осуществлять разработку кода программного модуля на современных языках программирования ОК 1, ОК 02, ОК 4, ОК 05 ОК 9, ПК 1.2, ПК 1.6.	определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.
У 5 Выполнять оптимизацию и рефакторинг программного кода ОК 1, ОК 02, ОК 4, ОК 05 ОК 9, ПК 1.5.	определяет способы и производит оптимизацию программного кода и его рефакторинг.
У 6 Оформлять документацию на программные средства ОК 1, ОК 2, ОК 08, ОК 9, ПК 1.1	применяет стандарты оформления документации; грамотно оформляет документацию на программные средства..
Знать:	
З 1 Основные этапы разработки программного обеспечения ОК 1, ОК 2, ОК 4, ПК 2.1	знает основные процессы жизненного цикла программного обеспечения; на основе процессов понимает

	последовательность разработки программного обеспечения.
3 2 Основные принципы технологии структурного и объектно-ориентированного программирования ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.3	понимает принципиальные различия в технологиях структурного и объектно-ориентированного программирования; определяет на основе какой технологии построен код программы;
3 3 Способы оптимизации и приемы рефакторинга ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2	знает основные способы оптимизации и рефакторинга кода; применяет методы к написанному коду; составляет код программы на основе методов.
3 4 Основные принципы отладки и тестирования программных продуктов ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.4, ПК 2.5	понимает последовательность действий при отладке программного продукта; осознает принципы и методы тестирования; создает тестовые кейсы на основе различных методов тестирования

Образец экзаменационного билета
МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

Утверждаю
Заместитель директора
Борзенкова Е.К.
(Ф.И.О.)

(подпись)

**Экзаменационный билет №1
по ПМ.01 Разработка модулей
программного обеспечения для
компьютерных систем**

Рассмотрено на заседании
цикловой методической
комиссии
Председатель Гаврикова Е.
Ю.
(Ф.И.О.)

«___» _____ 2024 г.

Группа (ы) _____

Специальность:

09.02.07 Информационные
системы и программирование

«___» _____ 2024 г.
(подпись)

1. Какие основные принципы структурного программирования и как они реализуются в языке программирования Python?
2. Создайте модуль, который принимает изображение в качестве входных данных, применяет к нему фильтр (например, ч/б, сепия, негатив) и возвращает измененное изображение.

Критерии оценки ответов обучающихся

Отметка 5 «отлично» - продемонстрирован высокий уровень знаний по теоретическому вопросу, тематика вопроса полностью раскрыта. Практико-ориентированное задание выполнено верно.

Отметка 4 «хорошо» - продемонстрировано понимание и знание основного содержания теоретического вопроса билета, однако допущены недочеты в определениях терминов и понятий. Практико-ориентированное задание выполнено с замечаниями.

Отметка 3 «удовлетворительно» - продемонстрировано слабое владение основным содержанием по теоретическому вопросу билета, допущены неточности в определениях терминов и понятий. Практико-ориентированное задание выполнено с ошибками.

Отметка 2 «неудовлетворительно» - не продемонстрировано владение знаниями и умениями, тема теоретического вопроса билета не раскрыта. Практико-ориентированное задание не выполнено.

4.Комплект билетов – 30 шт.

5.Оценочная ведомость по профессиональному модулю.

6.Сводный экзаменационный протокол на группу студентов по экзамену по профессиональному модулю.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

МДК 02.01 Технология разработки программного обеспечения

Подготовки специалистов среднего звена по специальности

код специальности 09.02.07 Информационные системы и программирование

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности (профессии) СПО 09.02.07 Информационные системы и программирование базовой подготовки программы МДК 02.01 Технология разработки программного обеспечения.

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчик: Бахаев Д. В., преподаватель,
Голубенко М. И., преподаватель.

ОДОБРЕН
цикловой методической комиссией
Протокол № 1
от «3» сентября 2022 г.
Председатель Гаврикова Е. Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения МДК, подлежащие проверке
3. Оценка освоения МДК
 - 3.1. Контроль и оценка освоения МДК
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет материалов
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения МДК 02.01 Технология разработки программного обеспечения обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 Информационные системы и программирование базового уровня следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями

Обучающийся должен знать:

З 1	модели процесса разработки программного обеспечения;
З 2	основные принципы процесса разработки программного обеспечения;
З 3	основные подходы к интегрированию программных модулей;
З 4	основы верификации и аттестации программного обеспечения;
З 5	основные методы отладки;
З 6	методы и схемы обработки исключительных ситуаций;
З 7	приемы работы с инструментальными средствами тестирования и отладки;
З 8	стандарты качества программной документации;
З 9	основы организации инспектирования и верификации;
З 10	встроенные и основные специализированные инструменты анализа качества программных продуктов.

Обучающийся должен уметь:

У 1	использовать выбранную систему контроля версий;
У 2	использовать методы для получения кода с заданной функциональностью и степенью качества;
У 3	выполнять тестирование интеграции;
У 4	организовывать постобработку данных;
У 5	использовать приемы работы в системах контроля версий;
У 6	выполнять отладку, используя методы и инструменты условной компиляции;
У 7	выявлять ошибки в системных компонентах на основе спецификаций.

Личностные результаты с учетом особенностей учебной дисциплины:

ЛР1	- осознающий себя гражданином и защитником великой страны
ЛР2	- проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	- демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих

ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личностного и профессионального, конструктивного «цифрового следа».
ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	ЛР 8. Проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	ЛР 9. Уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности.
ЛР 10	ЛР 10. Принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
Л 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия; выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
Л 19	- профессиональная идентичность и ответственность.

Обучающийся должен иметь практический опыт:

- участия в выработке требований к программному обеспечению;
- участия в проектировании программного обеспечения с использованием специализированных программных пакетов;
- разработки тестовых наборов (пакетов) для программного модуля;
- разработки тестовых сценариев программного средства;
- инспектирования разработанных программных модулей на предмет соответствия стандартам кодирования.

Формируемые ОК:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

ВД 2. Осуществление интеграции программных модулей.

Формируемые ПК:

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2 Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

Формами промежуточной аттестации по МДК являются контрольная работа - 7 семестр, экзамен – 8 семестр.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по МДК осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих и профессиональных компетенций:

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1 Использовать выбранную систему контроля версий ОК 1, ОК 4, ОК 5, ОК 9, ПК 2.3	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; эффективно взаимодействовать и работать в коллективе и команде; пользоваться

	<p>профессиональной документацией на государственном и иностранном языках. выполнять отладку программного модуля с использованием специализированных программных средств.</p>
<p>У 2 Использовать методы для получения кода с заданной функциональностью и степенью качества ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.3</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств.</p>
<p>У 3 Выполнять тестирование интеграции ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>

<p>У 4 Организовывать постобработку данных ОК 1, ОК 2, ОК 9, ПК 2.1</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>
<p>У 5 Использовать приемы работы в системах контроля версий ОК 1, ОК 4, ОК 9, ПК 2.3</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. выполнять отладку программного модуля с использованием специализированных программных средств.</p>
<p>У 6 Выполнять отладку, используя методы и инструменты условной компиляции ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>У 7 Выявлять ошибки в системных компонентах на основе спецификаций</p>	<p>использовать современные средства поиска, анализа и интерпретации информации, и</p>

<p>ОК 2, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4</p>	<p>информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>Знать:</p>	
<p>3 1 Модели процесса разработки программного обеспечения ОК 1, ОК 2, ОК 4, ПК 2.1</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>
<p>3 2 Основные принципы процесса разработки программного обеспечения ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.3</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия</p>

	компонент. выполнять интеграцию модулей в программное обеспечение.
3 3 Основные подходы к интегрированию программных модулей ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение.
3 4 Основы верификации и аттестации программного обеспечения ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.4, ПК 2.5	использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
3 5 Основные методы отладки ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на

	<p>государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>3 6 Методы и схемы обработки исключительных ситуаций ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>3 7 Приемы работы с инструментальными средствами тестирования и отладки ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов</p>

	и тестовых сценариев для программного обеспечения.
3 8 Стандарты качества программной документации ОК 2, ОК 9, ПК 2.1	использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.
3 9 Основы организации инспектирования и верификации ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.5	использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
3 10 Встроенные и основные специализированные инструменты анализа качества программных продуктов ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4, ПК 2.5	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку

	программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
--	---

3.ОЦЕНКА ОСВОЕНИЯ МДК

3.1. Контроль и оценка освоения МДК по темам (разделам)

Элемент МДК	Текущий контроль		Промежуточная аттестация	
	Осваиваемые результаты	Метод контроля	Проверяемые результаты	Форма контроля
	Раздел. 1 Разработка мобильных приложений			
Тема 1.1 Основные понятия и стандартизация требований к программному обеспечению	У 1-7 З 1-10 ОК 1, ОК 2, ОК 4-5, ОК 9, ПК 2.1, ПК 2.5	Устный опрос, практические и лабораторные работы, тестирование	У 1-7 З 1-10 ОК 1, 2, 5, 9; ПК 1.1, 1.2, 1.3, 1.4., 1.5, 1.6	Тестирование – 7 семестр
Тема 1.2 Описание и анализ требований. Диаграммы IDEF	У 1-7 З 1-10 ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.5	Устный опрос, практические и лабораторные работы, тестирование		
Тема 1.3 Оценка качества программных средств	У 1-7 З 1-10 ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4	Устный опрос, практические и лабораторные работы, тестирование и экзамен		Экзамен – 8 семестр

3.1.1. Методы и критерии оценивания

1. Устный опрос. Критерии оценивания.

Оценка 5 «отлично» - ответил на вопросы в объеме лекционного и дополнительного материала, дал полные грамотные ответы на все дополнительные вопросы.

Оценка 4 «хорошо» - грамотно изложил ответы на вопросы, но содержание и формулировки имеют отдельные неточности (допускается нечеткая формулировка определений), в полной мере ответил на заданные дополнительные вопросы.

Оценка 3 «удовлетворительно» - ответил на часть вопросов в объеме лекционного материала и ответил на часть дополнительных вопросов.

Оценка 2 «неудовлетворительно» - допустил ошибки в определении базовых понятий,

исказил их смысл, не ответил на дополнительные вопросы.

2. Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

3.Лабораторная работа. Критерии оценивания.

Выполнение работы в полном объеме с соблюдением необходимой последовательности проведения опытов, измерений – 2 балла;

Рациональный и самостоятельный выбор и подготовка необходимого оборудования для выполнения работ, обеспечивающих получение точных результатов – 2 балл;

Описание хода лабораторной работы в логической последовательности – 1 балл;

Корректная формулировка выводов по результатам лабораторной работы – 2 балла;

Выполнения всех записей, таблиц, рисунков, чертежей, графиков, вычислений в соответствии с заданием, технически грамотно и аккуратно – 2 балла;

Соблюдение правил техники безопасности при выполнении лабораторной работы – 1 балл

Перевод баллов в отметку:

Оценка 5 «отлично» - от 9 до 10 баллов

Оценка 4 «хорошо» - от 6 до 8 баллов.

Оценка 3 «удовлетворительно» - от 3 до 5 баллов.

Оценка 2 «неудовлетворительно» - от 1 до 2 баллов.

4. Практическая работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме, с соблюдением алгоритма выполнения: последовательности проведения измерений, заполнения таблиц, графиков и др.; правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; получены результаты в соответствии с поставленной целью.

Оценка 4 «хорошо»- выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью и объем выполненной части работы не позволяет получить никаких результатов в соответствии с поставленной целью.

4. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Контрольная работа

1.Форма проведения: контрольная работа (тестирование).

2.Условия выполнения:

1.Инструкция для обучающихся:

2.Время выполнения: 90 минут на тестовое задание.

3.Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.

4.Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда программирования PyCharm с установленным интерпретатором Python не ниже версии 3.8; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде

КГА ПОУ «ДИТК» и сети Интернет.

5. Информационные источники, допустимые к использованию на экзамене: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».

6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения тестирования

3.1. Перечень тем (разделов), выносимых на тестирование:

1. Основные понятия и стандартизация требований к программному обеспечению:

- понятия требований, классификация, уровни требований;
- методологии и стандарты, регламентирующие работу с требованиями;
- современные принципы и методы разработки программного контроля версий;
- основные подходы к интегрированию программных модулей;
- стандарты приложений;
- методы организации работы в команде разработчиков;
- системы кодирования.

2. Описание и анализ требований. Диаграммы IDEF:

- описание требований: унифицированный язык моделирования – краткий словарь. Диаграммы UML;
- описание и оформление требований (спецификация);
- анализ требований и стратегии выбора решения;

3. Оценка качества программных средств:

- цели и задачи и виды тестирования;
- стандарты качества программной документации;
- меры и метрики;
- тестовое покрытие;
- тестовый сценарий, тестовый пакет;
- анализ спецификаций;
- верификация и аттестация программного обеспечения.

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1 Использовать выбранную систему контроля версий ОК 1, 2, 5, 9; ПК 2.3	выполнять отладку программного модуля с использованием специализированных программных средств.
У 2 Использовать методы для получения кода с заданной функциональностью и степенью качества ОК 1, 2, 5, 9; ПК 2.1, ПК 2.3	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств.
У 3 Выполнять тестирование интеграции	разрабатывать требования к программным модулям на основе анализа проектной и

ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4	технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
У 4 Организовывать постобработку данных ОК 1, ОК 2, ОК 9, ПК 2.1	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.
У 5 Использовать приемы работы в системах контроля версий ОК 1, ОК 9, ПК 2.3	выполнять отладку программного модуля с использованием специализированных программных средств.
У 6 Выполнять отладку, используя методы и инструменты условной компиляции ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
У 7 Выявлять ошибки в системных компонентах на основе спецификаций ОК 2, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
Знать:	
З 1 Модели процесса разработки программного обеспечения ОК 1, ОК 2, ПК 2.1	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.
З 2 Основные принципы процесса разработки программного обеспечения ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять

	отладку программного модуля с использованием специализированных программных средств.
3 3 Основные подходы к интегрированию программных модулей ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение.
3 4 Основы верификации и аттестации программного обеспечения ОК 2, ОК 9, ПК 2.1, ПК 2.4, ПК 2.5	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
3 5 Основные методы отладки ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
3 6 Методы и схемы обработки исключительных ситуаций ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
3 7 Приемы работы с инструментальными средствами тестирования и отладки ОК 1, ОК 2, ОК 9, ПК 2.3, ПК 2.4	выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
3 8 Стандарты качества программной документации ОК 2, ОК 9, ПК 2.1	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

<p>3 9 Основы организации инспектирования и верификации ОК 2, ОК 9, ПК 2.1, ПК 2.5</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>
<p>3 10 Встроенные и основные специализированные инструменты анализа качества программных продуктов ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4, ПК 2.5</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>

Образец тестового задания

1. Буква «U» в аббревиатуре «UML» означает:
 - a. United
 - b. Unified
 - c. Universal
2. Модель UML состоит из (укажите лишнее):
 - a. Сущностей
 - b. Множеств
 - c. Отношений
3. Отношения UML подразделяются на (укажите лишнее):
 - a. Зависимости
 - b. Ассоциации
 - c. Уточнения
 - d. Обобщения
 - e. Реализации
4. Множество канонических диаграмм UML:
 - a. Определяется стандартом языка
 - b. Является соглашением пользователей языка
 - c. Определяется производителями инструментов, поддерживающих UML
5. Множество канонических структурных диаграмм UML включает в себя (укажите лишнее):
 - a. Диаграммы классов
 - b. Диаграммы вариантов использования
 - c. Диаграммы компонентов
 - d. Диаграммы объектов

6. Множество канонических поведенческих диаграмм UML включает в себя (укажите лишнее):
- Диаграммы состояний
 - Диаграммы деятельности
 - Диаграммы потоков данных
 - Диаграммы последовательности
7. Множество канонических поведенческих диаграмм UML включает в себя:
- Диаграммы классов
 - Диаграммы деятельности
 - Диаграммы последовательности
 - Диаграммы компонентов
 - Диаграммы развертывания
8. Актёр – это:
- внешняя сущность по отношению к компьютерной системе, которая может только снабжать информацией систему;
 - внешняя сущность по отношению к компьютерной системе, которая может только получать информацию из системы;
 - внутренняя сущность компьютерной системы, которая может только снабжать информацией систему;
 - внешняя сущность по отношению к компьютерной системе, которая взаимодействует с этой системой;
 - внутренняя сущность компьютерной системы, которая может только получать информацию из системы
9. Между двумя вариантами использования на диаграмме вариантов использования могут существовать следующие связи:
- агрегирование;
 - зависимость;
 - обобщения;
 - расширение
 - включение
10. Кто из указанных лиц принимал непосредственное участие в разработке первых версий языка UML?
- Джон фон Нейман
 - Джеймс Рубах
 - Гради Буч
 - Никлаус Вирт
 - Деннис Ритчи
11. Какой графический символ служит для изображения варианта использования (use case) на диаграмме вариантов использования?
- Эллипс
 - Ромб
 - Прямоугольник
12. Какое определение диаграммы вариантов использования правильно:
- диаграмма вариантов использования - это диаграмма, на которой изображаются функции моделируемой системы
 - диаграмма вариантов использования - это диаграмма, на которой изображаются

отношения между сотрудниками компании

с. диаграмма вариантов использования - это диаграмма, на которой изображаются отношения между актерами и вариантами использования

13. Какие из перечисленных диаграмм относятся к каноническим в языке UML:

- a. Диаграмма кооперации
- b. Диаграмма ERD
- c. Диаграмма функций системы
- d. Диаграмма развертывания
- e. Диаграмма последовательности

14. Какие элементы относятся к механизму расширения языка UML

- a. модели
- b. помеченные значения
- c. атрибуты
- d. стереотипы

15. Может ли разработчик уточнять семантику графических изображений языка UML:

- a. нет
- b. да, в произвольной форме
- c. да, но только в рамках механизма расширения

16. Выберите правильное окончание фразы: " Ассоциация на диаграмме вариантов использования связывает..."

- a. отдельных актеров между собой
- b. отдельного актера с вариантом использования
- c. отдельные варианты использования между собой

17. Какое высказывание справедливо применительно к операции включения?

- a. Отношение включения связывает актера с отдельным вариантом использования
- b. Отношение включения связывает только два варианта использования
- c. Отношение включения используется для изображения вложенности диаграмм вариантов использования друг в друга

18. Какое из перечисленных ниже определений объекта является правильным в контексте ООП

- a. объект - это логическая абстракция элемента физической системы
- b. объект - это физическая система, подлежащая представлению в форме некоторой модели
- c. объект - это экземпляр некоторого класса

19. Какое из приведенных ниже определений принципа полиморфизма является правильным в контексте ООП?

- a. полиморфизм характеризует свойство некоторых объектов принимать одинаковые формы в зависимости от обстоятельств
- b. полиморфизм характеризует сокрытие отдельных деталей внутреннего устройства классов от внешних по отношению к нему объектов или пользователей
- c. полиморфизм характеризует общий принцип независимости интерфейса операций от особенностей их реализации в фактическом классе, в соответствии с которым знание о более общей категории разрешается применять для более частной категории

20. Какие из перечисленных диаграмм относятся к каноническим в языке UML?

- a. диаграмма вариантов использования
- b. диаграмма моделей
- c. диаграмма IDEF0

d. диаграмма состояний

21. Что такое технология разработки программного обеспечения (ПО)?

22. Каковы основные этапы процесса разработки ПО?

23. Что такое Agile-методология в разработке ПО?

24. Какие инструменты используются для разработки ПО?

25. Что такое DevOps в контексте разработки ПО?

4. Эталонные ответы:

1. b, 2. c, 3. c, 4. a, 5. d, 6. a, 7. b, 8. d, 9. a, 10. b и c, 11. a, 12. c, 13. d, 14. b, 15. a, 16. b, 17. c, 18. c, 19. a, 20. A, 21. это совокупность методов, инструментов и практик, используемых для создания ПО, 22. анализ требований, проектирование, кодирование, тестирование, отладка и внедрение, 23. это итеративный подход к разработке, который предполагает короткие циклы разработки с частыми демонстрациями результатов заказчику, 24. интегрированные среды разработки (IDE), системы контроля версий, системы непрерывной интеграции и доставки, средства тестирования и анализа качества кода, 25. это методология, объединяющая процессы разработки (Dev) и эксплуатации (Ops), направленная на автоматизацию и упрощение процессов развертывания и обновления ПО.

Критерии оценивания

Оценка 5 «отлично» - на 85% и более вопросов дан верный ответ.

Оценка 4 «хорошо» - на 75% и более вопросов дан верный ответ.

Оценка 3 «удовлетворительно» - на 55% и более вопросов дан верный ответ.

Оценка 2 «неудовлетворительно» - правильные ответы даны ниже, чем на 55% вопросов.

2. Экзамен

1. Форма проведения: тестовое и практическое задания.

2. Условия выполнения:

1. Инструкция для обучающихся:

2. Время выполнения: 90 минут на тестовое задание, 30 – на практическую работу.

3. Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.

4. Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

5. Информационные источники, допустимые к использованию на экзамене: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».

6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения экзамена

3.1. Перечень тем, выносимых на экзамен

1. Основные понятия и стандартизация требований к программному обеспечению:

- понятия требований, классификация, уровни требований;
- методологии и стандарты, регламентирующие работу с требованиями;
- современные принципы и методы разработки программного контроля версий;
- основные подходы к интегрированию программных модулей;
- стандарты приложений;
- методы организации работы в команде разработчиков;

- системы кодирования.
2. Описание и анализ требований. Диаграммы IDEF:
- описание требований: унифицированный язык моделирования – краткий словарь. Диаграммы UML;
 - описание и оформление требований (спецификация);
 - анализ требований и стратегии выбора решения;
3. Оценка качества программных средств:
- цели и задачи и виды тестирования;
 - стандарты качества программной документации;
 - меры и метрики;
 - тестовое покрытие;
 - тестовый сценарий, тестовый пакет;
 - анализ спецификаций;
 - верификация и аттестация программного обеспечения.

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1 Использовать выбранную систему контроля версий ОК 1, ОК 5, ОК 9, ПК 2.3	выполнять отладку программного модуля с использованием специализированных программных средств.
У 2 Использовать методы для получения кода с заданной функциональностью и степенью качества ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств.
У 3 Выполнять тестирование интеграции ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
У 4 Организовывать постобработку данных ОК 1, ОК 2, ОК 9, ПК 2.1	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

У 5 Использовать приемы работы в системах контроля версий ОК 1, ОК 4, ОК 9, ПК 2.3	выполнять отладку программного модуля с использованием специализированных программных средств.
У 6 Выполнять отладку, используя методы и инструменты условной компиляции ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
У 7 Выявлять ошибки в системных компонентах на основе спецификаций ОК 2, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
Знать:	
З 1 Модели процесса разработки программного обеспечения ОК 1, ОК 2, ПК 2.1	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.
З 2 Основные принципы процесса разработки программного обеспечения ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств.
З 3 Основные подходы к интегрированию программных модулей ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение.
З 4 Основы верификации и аттестации программного обеспечения ОК 2, ОК 9, ПК 2.1, ПК 2.4, ПК 2.5	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. осуществлять

	<p>разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>
<p>3 5 Основные методы отладки ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>3 6 Методы и схемы обработки исключительных ситуаций ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>3 7 Приемы работы с инструментальными средствами тестирования и отладки ОК 1, ОК 2, ОК 9, ПК 2.3, ПК 2.4</p>	<p>выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>3 8 Стандарты качества программной документации ОК 2, ОК 9, ПК 2.1</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>
<p>3 9 Основы организации инспектирования и верификации ОК 2, ОК 9, ПК 2.1, ПК 2.5</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>
<p>3 10 Встроенные и основные специализированные инструменты анализа качества программных продуктов ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4, ПК 2.5</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных</p>

	<p>программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>
--	--

Образец практического задания

Для создания диаграммы классов нужны соответствующие наборы элементов – фигуры, при помощи которых будет выстраиваться схема. Чтобы фигуры были доступны, во вкладке «Поиск», в левой части экрана надо ввести «Класс UML».

Заполнение диаграммы стоит начать с определения классов-сущностей.

Сценарий состоит из:

- заказа;
- заказчика (клиента);
- комплектующих изделий, входящих в заказ.

Исходя из этого формируются следующие классы-сущности: Заказ, Клиент, Комплектующее изделие, а также в заказ может входить множество комплектующих, поэтому необходимо создать еще один класс Состав заказа. Каждый класс имеет атрибуты и операции. Атрибуты содержат информацию базы, а операции – варианты использования этой информации. Первым создаваемым классом будет класс «Клиент», в котором будут содержаться следующие атрибуты:

- ИМЯ (ФИО клиента);
- АДРЕС (адрес клиента);
- ТЕЛЕФОН (номер телефона клиента); и операции:
- ДОБАВИТЬ КЛИЕНТА;
- УДАЛИТЬ КЛИЕНТА;
- ПОЛУЧИТЬ ИНФОРМАЦИЮ.

Далее по аналогии создаются другие классы. Класс «Заказ».

Атрибуты:

- НОМЕР ЗАКАЗА;
- ДАТА ОФОРМЛЕНИЯ;
- ДАТА ВЫПОЛНЕНИЯ.

Операции:

- СОЗДАНИЕ НОВОГО ЗАКАЗА;
- ЗАНЕСТИ ИНФОРМАЦИЮ;
- ПОЛУЧИТЬ ИНФОРМАЦИЮ.

Класс «Комплектующее изделие». Атрибуты:

- НАИМЕНОВАНИЕ;
- ПРОИЗВОДИТЕЛЬ;
- ЦЕНА;
- ОПИСАНИЕ.

Операции:

- ДОБАВИТЬ НОВОЕ КОМПЛЕКТУЮЩЕЕ;
- УДАЛИТЬ КОМПЛЕКТУЮЩЕЕ;

- ПОЛУЧИТЬ ИНФОРМАЦИЮ (о комплектующем).

Класс «Состав заказа». Атрибуты:

- НОМЕР ПУНКТА ЗАКАЗА;
- КОЛИЧЕСТВО КОМПЛЕКТУЮЩИХ;
- ЦЕНА.

Операции:

- СОЗДАТЬ СТРОКУ ЗАКАЗА;
- ДОБАВИТЬ ИНФОРМАЦИЮ (о строке);
- ПОЛУЧИТЬ ИНФОРМАЦИЮ (о строке).

Чтобы занести все данные в диаграмму в наборе элементов нужно выбрать фигуру «Занятие» и поместить на рабочее поле. Далее ввести имя класса и имена участников. Чтобы добавить больше полей для участников (атрибутов и операций) нужно нажать правой кнопкой на существующего участника и в появившемся окне выбрать пункт «Вставить “Член перед...”».

ПРИМЕЧАНИЕ: обратить внимание на разделительную линию, отделяющую атрибуты от операций, имена атрибутов пишутся над линией, а имена операций – под линией.

4. Эталонный вариант ответа

Практическая часть.

После заполнения всех классов сущностей должно получиться как на рисунке:

Клиент
Имя
Адрес
Телефон
Добавить клиента
Удалить клиента
Получить информацию

Заказ
Номер заказа
Дата оформления
Дата выполнения
Создание нового заказа
Занести информацию
Получить информацию

Состав заказа
Номер пункта заказа
Количество комплектующих
Цена
Создать строку заказа
Добавить информацию
Получить информацию

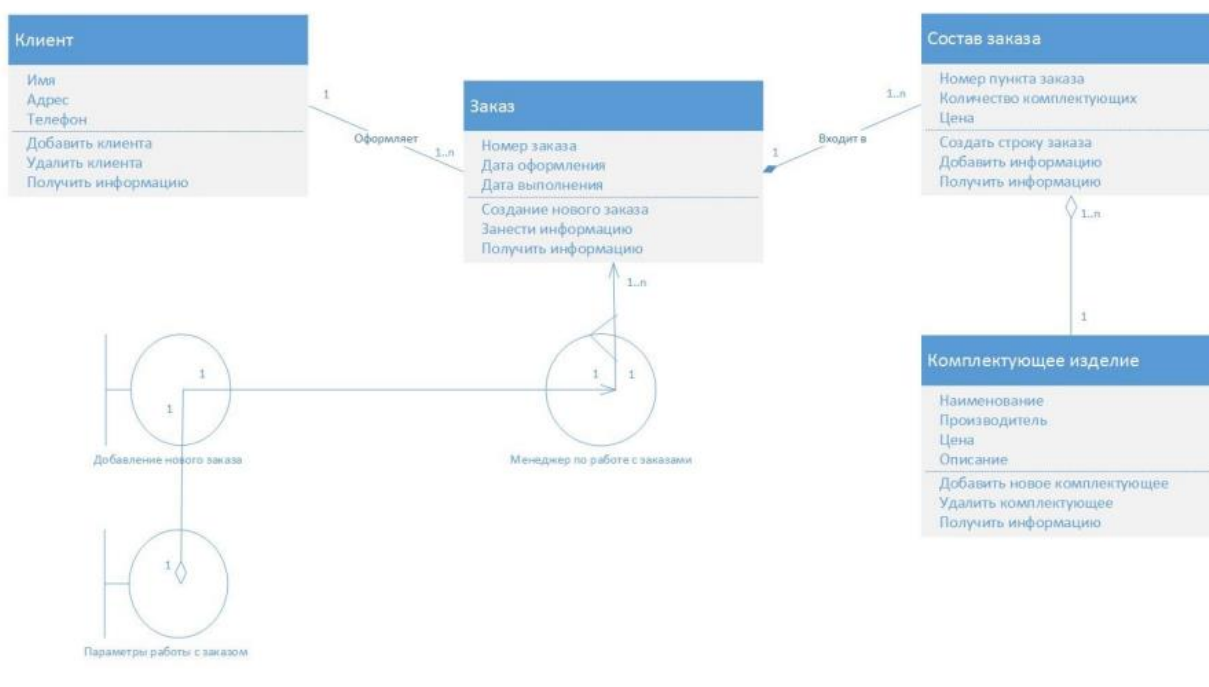
Комплектующее изделие
Наименование
Производитель
Цена
Описание
Добавить новое комплектующее
Удалить комплектующее
Получить информацию

Теперь необходимо добавить отношения между классами. Первая связь между классами «Клиент» и «Заказ». В данном случае подойдет только отношение «Ассоциация», которое представляет общую связь между экземплярами классов, так как классы просто связаны друг с другом и никакие другие связи здесь не подойдут. Чтобы задать связь в наборе фигур следует выбрать фигуру «Ассоциация» и после соединения классов подписать словом «Оформляет», так как клиент оформляет заказ. Также, задается кратность связи. Чтобы ее показать, нужно нажать правой кнопкой мыши на линию связи и выбрать пункт «Показать

кратность связи». Здесь будет связь со стороны клиента – 1, а со стороны заказа – 1..n. Поставлена именно такая кратность связи из-за того, что один клиент может сделать несколько заказов, а каждый заказ может поступать только от одного клиента.

Классы «Заказ» и «Состав заказа» имеют отношение композиции, поскольку строка заказа – это часть заказа, без которого существовать не может. Для построения связи необходимо в наборе элементов выбрать фигуру «Композиция» и разместить на схеме так, чтобы начало соединительной линии исходило из класса «Состав заказа». Аналогично подписывается связь и отображается кратность. В одном заказе может находиться несколько строк заказа, а строка заказа может относиться только к одному заказу, поэтому кратность выставляется как 1 – у заказа и 1..n – у состава заказа.

Следующее отношение между классами «Состав заказа» и «Комплектующее изделие» – это отношение агрегации, так как комплектующие изделия являются частью строки заказа, но каждый из этих элементов самостоятельный класс. Чтобы реализовать связь на схеме, необходимо выбрать фигуру «Агрегат» и разместить так, чтобы начало связи исходило из класса «Комплектующее изделие». Также стоит показать кратность, «Комплектующее изделие» - 1, «Состав заказа» - 1..n, потому что одно комплектующее изделие может входить во много строк заказа, а в одну строку заказа может входить лишь одно комплектующее изделие.



Критерии оценивания

Оценка 5 «отлично» - на 85% и более вопросов дан верный ответ и в практической части реализованы все основные требования.

Оценка 4 «хорошо» - на 75% и более вопросов дан верный ответ и в практической части реализована часть основных требований (больше половины).

Оценка 3 «удовлетворительно» - на 55% и более вопросов дан верный ответ и в практической части требования реализованы меньше, чем наполовину.

Оценка 2 «неудовлетворительно» - правильные ответы даны ниже, чем на 55% вопросов, и требования в практической части не реализованы.

5. Экзаменационная ведомость.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

МДК 02.02 Инструментальные средства разработки программного обеспечения

Подготовки специалистов среднего звена по специальности

код специальности 09.02.07 Информационные системы и программирование

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности (профессии) СПО 09.02.07 Информационные системы и программирование базовой подготовки программы МДК 02.02 Инструментальные средства разработки программного обеспечения.

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчик: Бахаев Д. В., преподаватель,
Голубенко М. И., преподаватель.

ОДОБРЕН
цикловой методической комиссией
Протокол № 1
от «2» сентября 2022 г.
Председатель Гаврикова Е. Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения МДК, подлежащие проверке
3. Оценка освоения МДК
 - 3.1. Контроль и оценка освоения МДК
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет материалов
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения МДК 02.02 Инструментальные средства разработки программного обеспечения обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 Информационные системы и программирование базового уровня следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями

Обучающийся должен знать:

З 1	модели процесса разработки программного обеспечения;
З 2	основные принципы процесса разработки программного обеспечения;
З 3	основные подходы к интегрированию программных модулей;
З 4	основы верификации и аттестации программного обеспечения;
З 5	основные методы отладки;
З 6	методы и схемы обработки исключительных ситуаций;
З 7	приемы работы с инструментальными средствами тестирования и отладки;
З 8	стандарты качества программной документации;
З 9	основы организации инспектирования и верификации;
З 10	встроенные и основные специализированные инструменты анализа качества программных продуктов.

Обучающийся должен уметь:

У 1	использовать выбранную систему контроля версий;
У 2	использовать методы для получения кода с заданной функциональностью и степенью качества;
У 3	выполнять тестирование интеграции;
У 4	организовывать постобработку данных;
У 5	использовать приемы работы в системах контроля версий;
У 6	выполнять отладку, используя методы и инструменты условной компиляции;
У 7	выявлять ошибки в системных компонентах на основе спецификаций.

Личностные результаты с учетом особенностей учебной дисциплины:

ЛР1	- осознающий себя гражданином и защитником великой страны
ЛР2	- проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	- демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих

ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личностного и профессионального, конструктивного «цифрового следа».
ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	ЛР 8. Проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	ЛР 9. Уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности.
ЛР 10	ЛР 10. Принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
Л 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия; выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
Л 19	- профессиональная идентичность и ответственность.

Обучающийся должен иметь практический опыт:

- участия в выработке требований к программному обеспечению;
- участия в проектировании программного обеспечения с использованием специализированных программных пакетов;
- разработки тестовых наборов (пакетов) для программного модуля;
- разработки тестовых сценариев программного средства;
- инспектирования разработанных программных модулей на предмет соответствия стандартам кодирования.

Формируемые ОК:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

ВД 2. Осуществление интеграции программных модулей.

Формируемые ПК:

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2 Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

Формами промежуточной аттестации по МДК являются другие формы контроля 7 семестр, экзамен – 8 семестр.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по МДК осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих и профессиональных компетенций:

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1 Использовать выбранную систему контроля версий ОК 1, ОК 02, ОК 4, ОК 9, ПК 2.3	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. выполнять отладку программного модуля с использованием специализированных программных средств.
У 2 Использовать методы для получения кода с заданной функциональностью и степенью качества ОК 1, ОК 2, ОК 03, ОК 4, ОК 9, ПК 2.1, ПК 2.3	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств.
У 3 Выполнять тестирование интеграции ОК 1, ОК 2, ОК 05, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках.

	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>У 4 Организовывать постобработку данных ОК 1, ОК 2, ОК 06, ОК 9, ПК 2.1</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>
<p>У 5 Использовать приемы работы в системах контроля версий ОК 1, ОК 4, ОК 07, ОК 9, ПК 2.3</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. выполнять отладку программного модуля с использованием специализированных программных средств.</p>
<p>У 6 Выполнять отладку, используя методы и инструменты условной компиляции ОК 1, ОК 2, ОК 08, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках.</p>

	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>У 7 Выявлять ошибки в системных компонентах на основе спецификаций ОК 2, ОК 05, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4</p>	<p>использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>Знать:</p>	
<p>З 1 Модели процесса разработки программного обеспечения ОК 1, ОК 2, ОК 4, ПК 2.1</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>
<p>З 2 Основные принципы процесса разработки программного обеспечения ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.3</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска,</p>

	<p>анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение.</p>
<p>3 3 Основные подходы к интегрированию программных модулей ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение.</p>
<p>3 4 Основы верификации и аттестации программного обеспечения ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.4, ПК 2.5</p>	<p>использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного</p>

	обеспечения на предмет соответствия стандартам кодирования.
3 5 Основные методы отладки ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
3 6 Методы и схемы обработки исключительных ситуаций ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
3 7 Приемы работы с инструментальными средствами тестирования и отладки ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.3, ПК 2.4	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и

	<p>информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>3 8 Стандарты качества программной документации ОК 2, ОК 9, ПК 2.1</p>	<p>использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>
<p>3 9 Основы организации инспектирования и верификации ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.5</p>	<p>использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>
<p>3 10 Встроенные и основные специализированные инструменты анализа качества программных продуктов ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4, ПК 2.5</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для</p>

	<p>выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>
--	--

3.ОЦЕНКА ОСВОЕНИЯ МДК

3.1. Контроль и оценка освоения МДК по темам (разделам)

Элемент МДК	Текущий контроль		Промежуточная аттестация	
	Осваиваемые результаты	Метод контроля	Проверяемые результаты	Форма контроля
	Раздел. 2 Средства разработки программного обеспечения			
Тема 2.1 Современные технологии и инструменты интеграции	У 1-7 З 1-10 ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.5	Устный опрос, практические и лабораторные работы, тестирование	У 1-7 З 1-10 ОК 1, 2, 5, 9; ПК 1.1, 1.2, 1.3, 1.4., 1.5, 1.6	Тестирование – 7 семестр
Тема 2.2 Инструментарий тестирования и анализа качества программных средств	У 1-7 З 1-10 ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.5	Устный опрос, практические и лабораторные работы, тестирование Экзамен Курсовой проект	У 1-7 З 1-10 ОК 1, 2, 5, 9; ПК 1.1, 1.2, 1.3, 1.4., 1.5, 1.6	Экзамен Курсовой проект – 8 семестр

3.1.1. Методы и критерии оценивания

1.Устный опрос. Критерии оценивания.

Оценка 5 «отлично» - ответил на вопросы в объеме лекционного и дополнительного материала, дал полные грамотные ответы на все дополнительные вопросы.

Оценка 4 «хорошо» - грамотно изложил ответы на вопросы, но содержание и формулировки имеют отдельные неточности (допускается нечеткая формулировка определений), в полной мере ответил на заданные дополнительные вопросы.

Оценка 3 «удовлетворительно» - ответил на часть вопросов в объеме лекционного материала и ответил на часть дополнительных вопросов.

Оценка 2 «неудовлетворительно» - допустил ошибки в определении базовых понятий, искажил их смысл, не ответил на дополнительные вопросы.

2. Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

3.Лабораторная работа. Критерии оценивания.

Выполнение работы в полном объеме с соблюдением необходимой последовательности проведения опытов, измерений – 2 балла;

Рациональный и самостоятельный выбор и подготовка необходимого оборудования для выполнения работ, обеспечивающих получение точных результатов – 2 балл;

Описание хода лабораторной работы в логической последовательности – 1 балл;

Корректная формулировка выводов по результатам лабораторной работы – 2 балла;

Выполнения всех записей, таблиц, рисунков, чертежей, графиков, вычислений в соответствии с заданием, технически грамотно и аккуратно – 2 балла;

Соблюдение правил техники безопасности при выполнении лабораторной работы – 1 балл

Перевод баллов в отметку:

Оценка 5 «отлично» - от 9 до 10 баллов

Оценка 4 «хорошо» - от 6 до 8 баллов.

Оценка 3 «удовлетворительно» - от 3 до 5 баллов.

Оценка 2 «неудовлетворительно» - от 1 до 2 баллов.

4. Практическая работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме, с соблюдением алгоритма выполнения: последовательности проведения измерений, заполнения таблиц, графиков и др.; правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; получены результаты в соответствии с поставленной целью.

Оценка 4 «хорошо»- выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью и объем выполненной части работы не позволяет получить никаких результатов в соответствии с поставленной целью.

4. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Контрольная работа

1.Форма проведения: контрольная работа (тестирование).

2.Условия выполнения:

1.Инструкция для обучающихся:

2.Время выполнения: 45 минут на тестовое задание.

3.Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.

4.Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента

– 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда программирования PyCharm с установленным интерпретатором Python не ниже версии 3.8; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

5. Информационные источники, допустимые к использованию на экзамене: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».

6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения тестирования

3.1. Перечень тем выносимых на тестирование:

1. Современные технологии и инструменты интеграции:

- понятие репозитория проекта, структура проекта.
- виды, цели и уровни интеграции программных модулей.
- автоматизация бизнес-процессов.
- выбор источников и приемников данных, сопоставление объектов данных.
- транспортные протоколы.
- стандарты форматирования сообщений.
- организация работы команды в системе контроля версий.

2. Инструментарий тестирования и анализа качества программных средств:

- отладка программных продуктов.
- инструменты отладки.
- отладочные классы.
- ручное и автоматизированное тестирование.
- методы и средства организации тестирования.
- инструментарии анализа качества программных продуктов в среде разработки.
- обработка исключительных ситуаций.
- методы и способы идентификации сбоев и ошибок.
- выявление ошибок системных компонентов.

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1 Использовать выбранную систему контроля версий ОК 1, ОК 9, ПК 2.3	выполнять отладку программного модуля с использованием специализированных программных средств.
У 2 Использовать методы для получения кода с заданной функциональностью и степенью качества ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств.
У 3 Выполнять тестирование интеграции	разрабатывать требования к программным модулям на основе анализа проектной и

<p>ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4</p>	<p>технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>У 4 Организовывать постобработку данных ОК 1, ОК 2, ОК 9, ПК 2.1</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>
<p>У 5 Использовать приемы работы в системах контроля версий ОК 1, ОК 9, ПК 2.3</p>	<p>выполнять отладку программного модуля с использованием специализированных программных средств.</p>
<p>У 6 Выполнять отладку, используя методы и инструменты условной компиляции ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>У 7 Выявлять ошибки в системных компонентах на основе спецификаций ОК 2, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>Знать:</p>	
<p>З 1 Модели процесса разработки программного обеспечения ОК 1, ОК 2, ПК 2.1</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>
<p>З 2 Основные принципы процесса разработки программного обеспечения ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять</p>

	отладку программного модуля с использованием специализированных программных средств.
3 3 Основные подходы к интегрированию программных модулей ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение.
3 4 Основы верификации и аттестации программного обеспечения ОК 2, ОК 9, ПК 2.1, ПК 2.4, ПК 2.5	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
3 5 Основные методы отладки ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
3 6 Методы и схемы обработки исключительных ситуаций ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
3 7 Приемы работы с инструментальными средствами тестирования и отладки ОК 1, ОК 2, ОК 9, ПК 2.3, ПК 2.4	выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
3 8 Стандарты качества программной документации ОК 2, ОК 9, ПК 2.1	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

<p>3 9 Основы организации инспектирования и верификации ОК 2, ОК 9, ПК 2.1, ПК 2.5</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>
<p>3 10 Встроенные и основные специализированные инструменты анализа качества программных продуктов ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4, ПК 2.5</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>

Образец тестового задания

1. Программная инженерия:

- 1) software engineering
- 2) Инструменты создания программного обеспечения
- 3) Коллектив инженеров-программистов, разрабатывающих программное обеспечение для компьютеров
- 4) Дисциплина, изучающая применение строгого систематического количественного подхода к разработке, эксплуатации и сопровождению программного обеспечения
- 5) Комплекс программ, предназначенный для решения инженерных задач, связанных с большим количеством расчетов
- 6) Инженерная индустрия применения прикладного программного обеспечения
- 7) Совокупность инженерных методов и средств создания программного обеспечения
- 8) Прикладное программное обеспечение для решения офисных задач

2. Построение SADT-модели включает в себя выполнение следующих действий:

- 1) Написание программного обеспечения для разрабатываемой системы по требованиям заказчика
- 2) Сбор информации об объекте, определение его границ
- 3) Определение цели и точки зрения модели, построение, обобщение и декомпозиция диаграмм
- 4) Представление исследуемой системы в графическом виде
- 5) Представление исследуемого объекта средствами системного моделирования
- 6) Критическая оценка, рецензирование и комментирование
- 7) Разработка, отладка и тестирование программного обеспечения
- 8) Использование графических пакетов для представления системы в виде модели

3. Моделирование основывается на принципах:

- 1) Выбор модели оказывает определяющее влияние на подход к решению проблемы и на то, как будет выглядеть это решение
 - 2) Декомпозиции системы на отдельные подзадачи
 - 3) Инкапсуляции и полиморфизма
 - 4) Децентрализации управления системой
 - 5) Каждая модель может быть представлена с различной степенью точности; лучшие модели – те, что ближе к реальности
 - 6) Открытой трансформируемой системы
 - 7) Нельзя ограничиваться созданием только одной модели. Наилучший подход при разработке любой нетривиальной системы – использовать совокупность нескольких моделей, почти независимых друг от друга
 - 8) Анализа и синтеза проектирования систем
4. В бизнес-процессах выделяют классы процессов:
- 1) Решающие бизнес-процессы
 - 2) Регламентирующие бизнес-процессы
 - 3) Основные бизнес-процессы
 - 4) Бизнес-процессы поведения системы
 - 5) Программируемые бизнес-процессы
 - 6) Экономические бизнес-процессы
 - 7) Обеспечивающие бизнес-процессы
 - 8) Бизнес-процессы управления
5. CASE-средства классифицируются по следующим признакам:
- 1) По применяемым методологиям и моделям систем и БД
 - 2) По используемому программному обеспечению
 - 3) По этапам жизненного цикла программного обеспечения
 - 4) По степени интегрированности с СУБД
 - 5) По уровням детализации и декомпозиции проектируемой системы
 - 6) По доступным платформам
 - 7) По используемым языкам программирования
 - 8) По степени сложности моделируемой системы
6. К малым интегрированным средствам моделирования относятся:
- 1) ARIS Toolset
 - 2) Design/IDEF
 - 3) ERwin
 - 4) BPwin
 - 5) Designer/2000
 - 6) Paradigm Plus
 - 7) Model Mart
 - 8) Rational Rose
7. К средним интегрированным средствам моделирования относятся:
- 1) Rational Rose
 - 2) Design/IDEF
 - 3) BPwin
 - 4) Designer/2000
 - 5) ARIS Toolset
 - 6) Model Mart

- 7) Paradigm Plus
- 8) ERwin
8. Объектно-ориентированная методология (ООМ) включает в себя составные части:
 - 1) Объектно-ориентированный анализ
 - 2) Объектно-ориентированный подкласс
 - 3) Объектно-ориентированное проектирование
 - 4) Объектно-ориентированная парадигма
 - 5) Объектно-ориентированная экспозиция
 - 6) Объектно-ориентированное моделирование
 - 7) Объектно-ориентированное программирование
 - 8) Объектно-ориентированная декомпозиция
9. К основным понятиям объектно-ориентированного подхода относятся:
 - 1) Обобщение
 - 2) Полиморфизм
 - 3) Инкапсуляция
 - 4) Реализация
 - 5) Агрегирование
 - 6) Наследование
 - 7) Ассоциация
 - 8) Композиция
10. Главные принципы объектного подхода:
 - 1) Абстрагирование
 - 2) Наследование
 - 3) Ограничение доступа или инкапсуляция
 - 4) Безграничный доступ или инкапсуляция
 - 5) Модульность и иерархия
 - 6) Агрегирование
 - 7) Композиция
 - 8) Обобщение и специализация
11. Дополнительные принципы объектного подхода:
 - 1) Реализация
 - 2) Типизация
 - 3) Параллелизм
 - 4) Внедрение
 - 5) Перпендикулярность
 - 6) Сохраняемость или устойчивость
 - 7) Несохранимость или неустойчивость
 - 8) Динамичность
12. К инструментальным средствам объектно-ориентированного анализа и проектирования относятся:
 - 1) Rational Rose
 - 2) Model Mart
 - 3) MS Visio
 - 4) ARIS
 - 5) IDEF1X
 - 6) Erwin

7) BPwin

8) JAM

13. К инструментальным средствам представления функциональных моделей относятся:

1) JAM

2) Model Mart

3) MS Visio

4) ARIS

5) IDEF0

6) Erwin

7) BPwin

8) Rational Rose

14. Методологии, поддерживаемые в BPwin:

1) IDEF1X

2) IDEF0

3) IDEF1

4) IDEF3

5) IDEFX

6) IDEF5

7) DFD

8) DFD1X

15. Диаграмма IDEF0 может содержать следующие типы диаграмм:

1) Диаграмму классов

2) Контекстную диаграмму, диаграмму декомпозиции

3) Диаграмму компонентов

4) Диаграмму дерева узлов

5) Диаграмму взаимодействий

6) Диаграмму только для экспозиции (FEO)

7) Диаграмму последовательности, диаграмму кооперации

8) Диаграмму узлов

16. Уровни логической модели:

1) Диаграмма сущность

2) Диаграмма связь

3) Диаграмма пакетов

4) Диаграмма сущность-связь

5) Модель данных, основанная на классах

6) Модель данных, основанная на ключах

7) Полная операционная модель

8) Полная атрибутивная модель

17. Внутренние стрелки не входящие в состав диаграммы IDEF0:

1) mechanism- output

2) output-input

3) mechanism- input

4) output-control

5) output-input feedback

6) output-control feedback

7) output-mechanism

8) control feedback- mechanism

18. Типы стрелок не входящие в состав диаграммы IDEF0:

- 1) Input
- 2) Editor
- 3) Control
- 4) Properties
- 5) Output
- 6) Mechanism
- 7) Call
- 8) Dictionary

19. Quick Reports – создание простейших отчетов – позволяет создавать отчеты:

- 1) Group/Totals. Табличный отчет с автоматической группировкой и сортировкой данных
- 2) Report Header. Печатается единожды в начале отчета
- 3) Columnar. Простой табличный отчет
- 4) Page Header. Печатается в верхней части каждой страницы
- 5) Vertical. Простой вертикальный отчет
- 6) Group Header. Печатается в начале каждой группы
- 7) Blank Report. Бланк. Создается пустой бланк отчета, в который не включаются данные
- 8) Detail. Печатается для каждой строчки набора данных

20. BPwin допускает следующие переходы с одной нотации на другую:

- 1) IDEF3 → DFD
- 2) DFD → IDEF0
- 3) IDEF0 → DFD
- 4) DFD → DFD
- 5) IDEF3 → IDEF0
- 6) IDEF0 → IDEF3
- 7) IDEF3 → IDEF3
- 8) DFD → IDEF3

21. Какие основные виды инструментальных средств используются при разработке программного обеспечения?

22. Что такое интегрированная среда разработки и какие функции она обычно включает?

23. Что такое компилятор и какие задачи он выполняет в процессе разработки ПО?

24. В чем заключается роль отладчика в процессе создания ПО и какие основные возможности он предоставляет разработчикам?

25. Что такое система контроля версий и зачем она нужна в процессе разработки программного обеспечения?

4. Эталонные ответы:

1) 1 и 4, 2) 2, 3 и 6, 3) 1, 5 и 7, 4) 3, 7 и 8, 5) 1, 4 и 6, 6) 3, 4 и 7, 7) 2, 4 и 5, 8) 1, 3 и 7, 9) 2, 3 и 6, 10) 1, 2, 3 и 5, 11) 2, 3 и 6, 12) 1, 3 и 4, 13) 2, 6 и 7, 14) 2, 4 и 7, 15) 2, 4 и 6, 16) 4, 6 и 8, 17) 1, 3 и 8, 18) 2, 4 и 8, 19) 3, 5 и 7, 20) 3, 6 и 8, 21) инструментальные средства подразделяются на встроенные и независимые, 22) интегрированная среда разработки (IDE) – это программная среда, которая включает в себя множество инструментов для разработки программного обеспечения, 23) компилятор – это инструмент, который преобразует исходный код программы на языке программирования в двоичный код, понятный компьютеру, 24) отладчик – это средство, которое помогает разработчикам находить и исправлять ошибки в программе. Он предоставляет возможности для установки точек

останова, отслеживания значений переменных, выполнения кода построчно и просмотра стека вызовов, 25) система контроля версий – это программное обеспечение, которое позволяет разработчикам сохранять историю изменений в коде проекта, возвращаться к предыдущим версиям и объединять изменения между несколькими разработчиками.

Критерии оценивания

Оценка 5 «отлично» - на 85% и более вопросов дан верный ответ.

Оценка 4 «хорошо» - на 75% и более вопросов дан верный ответ.

Оценка 3 «удовлетворительно» - на 55% и более вопросов дан верный ответ.

Оценка 2 «неудовлетворительно» - правильные ответы даны ниже, чем на 55% вопросов.

2. Экзамен

1. Форма проведения: теоретическая и практическое задания.

2. Условия выполнения:

1. Инструкция для обучающихся:

2. Время выполнения: 30 минут на теоретическое задание, 30 – на практическую работу.

3. Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.

4. Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

5. Информационные источники, допустимые к использованию на экзамене: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».

6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения экзамена

3.1. Перечень тем, выносимых на экзамен: темы из рабочей программы

1. Современные технологии и инструменты интеграции:

- понятие репозитория проекта, структура проекта.
- виды, цели и уровни интеграции программных модулей.
- автоматизация бизнес-процессов.
- выбор источников и приемников данных, сопоставление объектов данных.
- транспортные протоколы.
- стандарты форматирования сообщений.
- организация работы команды в системе контроля версий.

2. Инструментарий тестирования и анализа качества программных средств:

- отладка программных продуктов.
- инструменты отладки.
- отладочные классы.
- ручное и автоматизированное тестирование.
- методы и средства организации тестирования.
- инструментарии анализа качества программных продуктов в среде разработки.
- обработка исключительных ситуаций.
- методы и способы идентификации сбоев и ошибок.
- выявление ошибок системных компонентов.

3.2. Перечень практических заданий к экзамену:

1. Составить программу для вычисления площади треугольника по формуле Герона.
2. Дана последовательность действительных чисел. Выяснить, будет ли она возрастающей.
3. Составить программу для вычисления площади треугольника по через синус угла и двум прилежащим к нему сторонам.
4. Найти произведение положительных элементов одномерного массива A размера N.
5. Определить время года по номеру месяца. Номер месяца вводить с клавиатуры.
6. Составить программу для анализа, введенного пользователем числа (целое или нет; положительное, отрицательное или нуль; четное или нечетное).
7. В переменную последовательно вводятся N вещественных чисел. Вычислить максимальное значение.
8. В массив A[N] занесены натуральные числа. Найти сумму тех элементов, которые кратны данному K.
9. Составить программу вычисления факториала введенного с клавиатуры числа. // результат вывести в таком виде: fact=1*2*3=6 (при n=3).
10. Определить максимальный элемент массива A[10] и его порядковый номер.
11. Составить программу для вычисления суммы всех натуральных чисел, кратных числу b и меньших 100.
12. В переменную последовательно вводят числа, отличные от нуля. Окончание ввода - ноль. Определить среднее арифметическое отрицательных чисел.
13. С клавиатуры вводятся числа. Суммировать числа до тех пор, пока сумма не станет больше 100. Вывести сумму и количество просуммированных чисел.
14. Составить программу, которая запрашивает дату (число, месяц, год) и проверяет корректность введенным пользователем данных.
15. Найти сумму положительных элементов одномерного массива A размера N.
16. В массиве целых чисел есть нулевые элементы. Создать массив из номеров этих элементов.
17. Написать программу вывода на экран четных чисел из интервала от 0 до 100.
18. Составить программу, которая по номеру дня недели выводит на экран расписание уроков в вашей группе в соответствующий день.
19. Написать программу для подсчета суммы чисел, кратных 3 в диапазоне от 30 до 60.
20. Составить программу для нахождения минимального значения среди элементов, стоящих до первого четного элемента.
21. Составить программу для поиска произведения положительных элементов массива.
22. Написать программу для нахождения в массиве из N элементов количества нулевых элементов.
23. Найти количество положительных элементов одномерного массива A размера N.
24. Написать программу для определения максимального элемента массива A[20].
25. Определить минимальный элемент массива A[15] и его порядковый номер.

Эталоны ответов на практические задания

1.

```
from math import sqrt

def area_of_triangle(a, b, c):
    s = (a + b + c) / 2
    return sqrt(s * (s - a) * (s - b) * (s - c))
```

```
a = float(input("Введите сторону a: "))
b = float(input("Введите сторону b: "))
c = float(input("Введите сторону c: "))
area = area_of_triangle(a, b, c)
print("Площадь треугольника:", area)
```

2.

```
def is_increasing(sequence):
    for i in range(1, len(sequence)):
        if sequence[i] < sequence[i - 1]:
            return False
    return True

sequence = [float(x) for x in input("Enter the sequence: ").split()]

if is_increasing(sequence):
    print("The sequence is increasing.")
else:
    print("The sequence is not increasing.")
```

3.

```
import math

def area_triangle(a, b, gamma):
    assert gamma <= 180 and gamma >= 0
    half_angle = gamma / 2
    sin_gamma = math.sin(math.radians(half_angle))
    area = (a * b * sin_gamma) / 2
    return area

a = float(input("Введите первую сторону треугольника: "))
b = float(input("Введите вторую сторону треугольника: "))
gamma = float(input("Введите градусную меру угла между этими сторонами: "))
area = area_triangle(a, b, gamma)
print("Площадь треугольника:", area)
```

4.

```
result = 1
for number in array:
    if number > 0:
        result *= number

print(result)
```

5.

```
seasons = {1: "Winter", 2: "Winter", 3: "Spring", 4: "Spring", 5: "Spring",
           6: "Summer", 7: "Summer", 8: "Summer", 9: "Autumn", 10: "Autumn",
           11: "Autumn", 12: "Winter"}
```

```
month = int(input("Enter month number (1-12): "))
season = seasons[month]
print(f"Season: {season}")
```

6.

```
number = int(input('Enter a number: '))

is_integer = False if number == int(number) else True
sign = " if number == 0 else 'positive' if number > 0 else 'negative'

is_even = False if (number % 2) else True
is_odd = True if (number % 2) else False

print(f'The analyzed number: {number}')
print(f'\nInteger?: {is_integer}')
print(f'Sign: {sign}')
print(f'Even?: {is_even}')
print(f'Odd?: {is,odd}')
```

7.

```
n = int(input())
maximum = float('-inf')

for _ in range(n):
    number = float(input())
    maximum = max(maximum, number)

print(maximum)
```

8.

```
n = int(input())
k = int(input())
a = []
s = 0

for i in range(n):
    num = int(input())
    a.append(num)

for elem in a:
    if elem % k == 0:
        s += elem

print(s)
```

9.

```
def factorial(n):
    if n == 0:
```

```

    return 1
else:
    return n * factorial(n-1)

n = int(input())
fact = factorial(n)

numbers = str(n)
numbers = numbers.replace("0", "")
numbers_list = numbers.split(" ")
result = ""
count = 2

for number in numbers_list:
    result += f"{number}*"

first_number = int(numbers_list[0])
last_number = n

result_reverse = ""

while last_number > 1:
    last_number -= 1
    result_reverse += str(last_number) + "*"

if fact == 0:
    print("1")
else:
    result = result[:-1]
    result_reverse = result_reverse[:-1]

```

10.

```

A = [1, 5, 2, -5, 0, 3, 4, 2, 1, -10]

max_elem = A[0]
max_index = 0
for i in range(len(A)):
    if A[i] > max_elem:
        max_elem = A[i]
        max_index = i
print("Максимальный элемент:", max_elem)
print("Порядковый номер максимального элемента:", max_index)

```

11.

```

b = int(input())

```

```
sum = 0
for i in range(b, 101):
    if i % b == 0:
        sum += i
print(sum)
```

12.

```
total_sum = 0
negatives_count = 0

number = int(input())

while number != 0:
    total_sum += number
    if number < 0:
        negatives_count += 1
    number = int(input())

average = total_sum / negatives_count
print(average)
```

13.

```
total = 0
count = 0
while total <= 100:
    number = float(input())
    total += number
    count += 1

print("Сумма:", total)
print("Количество чисел:", count)
```

14.

```
def main():
    year = int(input("Enter year: "))
    month = int(input("Enter month (1-12): "))
    day = int(input("Enter day (1-31): "))
    check_date(year, month, day)

def check_date(year, month, day):
    current_year = 2021
    if year < 100 or year > current_year:
        print("Invalid year")
    elif month < 1 or month > 12:
        print("Invalid month")
    elif day < 1 or day > days_in_month(year, month):
        print("Invalid day")
    else:
```



```
print("Date valid")

def days_in_month(year, month):
    if month == 2:
        if (year % 4 == 0 and year % 100 != 0) or year % 400 == 0:
            return 29
        else:
            return 28
    else:
        return [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31][month - 1]

main()
```

15.

```
arr = [1, -2, 3, -4, 5]

total = 0
for num in arr:
    if num > 0:
        total += num

print(total)
```

16.

```
def index_of_zero(arr):
    for i, v in enumerate(arr):
        if v == 0:
            yield i

arr = map(int, raw_input().split())
result = list(index_of_zero(arr))
print(result)
```

17.

```
for i in range(101):
    if i%2 == 0:
        print(i)
```

18.

```
def print_schedule(day_number):
    lessons = [
        {'day_number': 0, 'lesson_name': 'Математика', 'start_time': '8:30', 'end_time': '10:00'},
        {'day_number': 0, 'lesson_name': 'Русский язык', 'start_time': '10:30', 'end_time': '12:00'},
        {'day_number': 0, 'lesson_name': 'Литература', 'start_time': '12:30', 'end_time': '14:00'},
        {'day_number': 1, 'lesson_name': 'Математика', 'start_time': '8:30', 'end_time': '10:00'},
        {'day_number': 1, 'lesson_name': 'Русский язык', 'start_time': '10:30', 'end_time': '12:00'},
        {'day_number': 1, 'lesson_name': 'Литература', 'start_time': '12:30', 'end_time': '14:00'},
        {'day_number': 2, 'lesson_name': 'Математика', 'start_time': '8:30', 'end_time': '10:00'},
        {'day_number': 2, 'lesson_name': 'Русский язык', 'start_time': '10:30', 'end_time': '12:00'},
```

```

    {'day_number': 2, 'lesson_name': 'Литература', 'start_time': '12:30','end_time':'14:00'},
    {'day_number': 3, 'lesson_name': 'Математика', 'start_time': '8:30', 'end_time':'10:00'},
    {'day_number': 3, 'lesson_name': 'Русский язык', 'start_time': '10:30','end_time':'12:00'},
    {'day_number': 3, 'lesson_name': 'Литература', 'start_time': '12:30','end_time':'14:00'},
    {'day_number': 4, 'lesson_name': 'Математика', 'start_time': '8:30', 'end_time':'10:00'},
    {'day_number': 4, 'lesson_name': 'Русский язык', 'start_time': '10:30','end_time':'12:00'},
    {'day_number': 4, 'lesson_name': 'Литература', 'start_time': '12:30','end_time':'14:00'},
    {'day_number': 5, 'lesson_name': 'Математика', 'start_time': '8:30', 'end_time':'10:00'},
    {'day_number': 5, 'lesson_name': 'Русский язык', 'start_time': '10:30','end_time':'12:00'},
    {'day_number': 5, 'lesson_name': 'Литература', 'start_time': '12:30','end_time':'14:00'},
]

lessons_by_day_number = {}
for day_number, lessons in enumerate(lessons):
    lessons_by_day_number[day_number] = list(filter(lambda x: x['day_number'] ==
day_number, lessons))

return lessons_by_day_number[day_number]

day_number = int(input('Введите номер дня недели (начиная с 0 - понедельник): '))
print_schedule(day_number)

```

19.

```

sum = 0

for i in range(30, 61):
    if i % 3 == 0:
        sum += i

print(sum)

```

20.

```

arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
min_value = float("inf")
for i, val in enumerate(arr):
    if val % 2 == 0 and val != 0:
        index = i
        break
    if arr[i - 1] % 2 != 0 and arr[i - 1]
        min_value = min(min_value, val)
print("Минимальное значение:", min_value)

```

21.

```

arr = [1,-2,3,4,-5,6]
prod = 1
for num in arr:
    if num > 0:
        prod *= num

```

```
print(prod)
```

22.

```
N = int(input())
zero_count = 0
for _ in range(N):
    num = int(input())
    if num == 0:
        zero_count += 1
print(zero_count)
```

23.

```
N = int(input())
pos_count = 0
for _ in range(N):
    num = int(input())
    if num > 0:
        pos_count += 1
print(pos_count)
```

24.

```
A = [int(x) for x in input().split()]
max_elem = A[0]
for elem in A:
    if elem > max_elem:
        max_elem = elem
print(max_elem)
```

25.

```
A = [_ for _ in range(15)]
min_elem = min(A)
pos = A.index(min_elem)
print(f"Минимальный элемент: {min_elem}, его позиция: {pos}")
```

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
У 1 Использовать выбранную систему контроля версий ОК 1, ОК 9, ПК 2.3	выполнять отладку программного модуля с использованием специализированных программных средств.
У 2 Использовать методы для получения кода с заданной функциональностью и степенью качества ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств.

<p>У 3 Выполнять тестирование интеграции ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>У 4 Организовывать постобработку данных ОК 1, ОК 2, ОК 9, ПК 2.1</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>
<p>У 5 Использовать приемы работы в системах контроля версий ОК 1, ОК 9, ПК 2.3</p>	<p>выполнять отладку программного модуля с использованием специализированных программных средств.</p>
<p>У 6 Выполнять отладку, используя методы и инструменты условной компиляции ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>У 7 Выявлять ошибки в системных компонентах на основе спецификаций ОК 2, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>Знать:</p>	
<p>З 1 Модели процесса разработки программного обеспечения ОК 1, ОК 2, ПК 2.1</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>
<p>З 2 Основные принципы процесса разработки программного обеспечения</p>	<p>разрабатывать требования к программным модулям на основе анализа проектной и</p>

ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3	технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств.
3 3 Основные подходы к интегрированию программных модулей ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение.
3 4 Основы верификации и аттестации программного обеспечения ОК 2, ОК 9, ПК 2.1, ПК 2.4, ПК 2.5	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
3 5 Основные методы отладки ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
3 6 Методы и схемы обработки исключительных ситуаций ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
3 7 Приемы работы с инструментальными средствами тестирования и отладки ОК 1, ОК 2, ОК 9, ПК 2.3, ПК 2.4	выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
3 8 Стандарты качества программной документации	разрабатывать требования к программным модулям на основе анализа проектной и

ОК 2, ОК 9, ПК 2.1	технической документации на предмет взаимодействия компонент.
3 9 Основы организации инспектирования и верификации ОК 2, ОК 9, ПК 2.1, ПК 2.5	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
3 10 Встроенные и основные специализированные инструменты анализа качества программных продуктов ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4, ПК 2.5	разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

Образец КИМ

Часть А

1. Что представляет собой репозиторий проекта?
 - а. Структура базы данных проекта.
 - б. Хранилище версий кода проекта.
 - в. Набор тестовых данных для проекта.
 - г. Инструмент для отладки программы.
2. На каких уровнях может осуществляться интеграция программных модулей?
 - а. Только на уровне исходного кода.
 - б. Уровень интерфейсов и уровень данных.
 - в. Только на уровне исполняемого кода.
 - г. Уровень проектирования и уровень тестирования.
3. Что включает в себя автоматизация бизнес-процессов?
 - а. Только создание электронной почты.
 - б. Автоматизацию только финансовых процессов.
 - в. Упрощение и автоматизацию повседневных бизнес-операций.
 - г. Только управление проектами.
4. Какие протоколы используются для транспортировки данных при интеграции программных модулей?
 - а. HTTP, FTP, SMTP.
 - б. XML, JSON, HTML.
 - в. SQL, NoSQL, GraphQL.
 - г. TCP, UDP, IP.

5. Что является стандартами форматирования сообщений при интеграции?
- a. HTML, CSS.
 - b. XML, JSON.
 - c. SQL, NoSQL.
 - d. HTTP, HTTPS.
6. Как организуется работа команды в системе контроля версий?
- a. Все члены команды работают независимо от системы контроля версий.
 - b. Каждый член команды имеет свою ветку кода.
 - c. Все изменения вносятся в одну общую ветку (branch).
 - d. Все ответы верны.
7. Что такое отладка программных продуктов?
- a. Процесс нахождения и устранения ошибок в программном коде.
 - b. Процесс написания программного кода.
 - c. Процесс тестирования пользовательского интерфейса.
 - d. Процесс создания документации к программе.
8. Какие инструменты могут использоваться для отладки программ?
- a. Только текстовые редакторы.
 - b. Интегрированные среды разработки (IDE), отладчики.
 - c. Только электронные таблицы.
 - d. Браузеры интернета.
9. Что включает в себя ручное тестирование программного продукта?
- a. Использование автоматических скриптов тестирования.
 - b. Тестирование без участия человека.
 - c. Тестирование, проводимое тестировщиками вручную без использования автоматизированных средств.
 - d. Только тестирование производительности.
10. Какие методы и средства организации тестирования могут применяться в разработке программных продуктов?
- a. Только тестирование в рамках одного модуля.
 - b. Модульное тестирование, интеграционное тестирование, системное тестирование.
 - c. Только функциональное тестирование.
 - d. Только тестирование безопасности.
11. Задание:
1. В современном _____ тестирования особое внимание уделяется автоматизации процессов для увеличения эффективности и скорости проверки программного продукта.
 2. Один из основных инструментов анализа качества кода — это статический _____, который позволяет выявлять потенциальные ошибки и недочеты на ранних этапах разработки.
 3. Для интеграции тестирования в процесс разработки широко используются инструменты для автоматизации _____ тестирования, такие как Selenium и Appium.
12. Вопросы:
1. Какие основные преимущества автоматизации процессов в современном тестировании?
 2. Какие задачи может решить статический анализ кода в процессе разработки?
 3. Почему важна интеграция тестирования в процесс разработки, и какие инструменты могут быть использованы для автоматизации функционального тестирования?

Часть Б

Задание Части Б является практическим с применением персонального компьютера. И индивидуально для каждого студента.

Образец практического задания

1. Составить программу для вычисления площади треугольника по формуле Герона

4. Эталонны ответов

Часть А.

1. b; 2. b; 3. c; 4. a; 5. b; 6. c; 7. a; 8. b; 9. c; 10. b

11. мире, анализ, функционального.

12.

1. Основные преимущества автоматизации процессов в современном тестировании включают повышение эффективности тестирования, ускорение процесса проверки программного продукта, увеличение покрытия тестами, снижение вероятности человеческих ошибок, а также возможность запуска тестов в автоматическом режиме при каждом изменении кода.

2. Статический анализ кода в процессе разработки может решать следующие задачи: выявление потенциальных ошибок и недочетов на ранних этапах разработки, обеспечение соответствия кода стандартам и правилам оформления, анализ использования ресурсов и оптимизация кода.

3. Интеграция тестирования в процесс разработки важна для обеспечения непрерывной проверки качества кода. Инструменты для автоматизации функционального тестирования, такие как Selenium и Appium, позволяют выполнять тесты на различных уровнях приложения (веб, мобильные устройства) и интегрировать их в процессы непрерывной интеграции и развёртывания (CI/CD), обеспечивая раннее выявление и устранение дефектов.

Практическая часть.

1. Репозиторий проекта – это централизованное хранилище файлов и данных, связанных с проектом. Он содержит всю информацию, необходимую для разработки, тестирования и развёртывания программного обеспечения. Репозитории могут быть разных классов уровней, каждый из которых имеет свои особенности и назначение.

Классы уровней репозитория:

- Локальный репозиторий: это самый базовый уровень репозитория, который обычно используется разработчиками на своих компьютерах. Он позволяет им хранить и управлять своими файлами и кодами, а также синхронизировать их с удалённым репозиторием.
- Удалённый репозиторий (например, GitHub, GitLab, Bitbucket): это централизованное хранилище, которое может быть доступно для большого числа разработчиков. Он используется для хранения и управления версиями файлов проекта, позволяя команде работать вместе и отслеживать изменения.
- Главный репозиторий или репозиторий с открытым исходным кодом: это высокоуровневый репозиторий, который используется для хранения больших проектов с открытым исходным кодом. Он обеспечивает доступ к коду для широкой аудитории, а также позволяет пользователям просматривать, коммитить и обсуждать изменения в проекте.

2. Инструментарий анализа качества программных продуктов включает в себя различные

инструменты и методы, которые позволяют оценить качество и эффективность программного обеспечения. В среде разработки такие инструменты могут быть использованы для проверки кода на наличие ошибок, оптимизации производительности, анализа покрытия тестами и других целей. Некоторые из наиболее распространенных инструментов анализа качества ПО включают:

- Статические анализаторы кода - инструменты, которые проверяют код на наличие синтаксических ошибок, потенциальных проблем с производительностью и совместимостью. Примеры: SonarQube, CheckStyle, PMD.
 - Динамические анализаторы кода (или профилировщики) - инструменты для анализа производительности и оптимизации кода во время выполнения программы. Примеры: JProfiler, YourKit, VisualVM.
 - Системы контроля версий - инструменты для управления изменениями в коде, отслеживания ошибок и предоставления информации о ходе разработки проекта. Примеры: Git, Mercurial, Subversion.
 - Автоматические тесты - инструменты для создания и выполнения тестов, которые проверяют правильность работы кода. Примеры: Junit, TestNG, PyTest.
 - Средства контроля качества кода - инструменты для оценки качества кода, такие как метрики сложности, читаемости и дублирования кода. Примеры: Мсcabe, Cyclomatic Complexity.
 - Средства анализа покрытия тестами - инструменты для определения того, какие части кода покрыты тестами, и какие части не покрыты. Примеры: JaCoCo, Cobertura.
 - Системы непрерывной интеграции - инструменты для автоматизации процесса сборки проекта и тестирования после каждого коммита в систему контроля версий. Примеры: Jenkins, TeamCity, Bamboo.
 - Инструменты анализа зависимостей - инструменты для визуализации и анализа взаимосвязей между различными компонентами и зависимостями в проекте. Примеры: Graphviz, Maven Dependency Plugin.
 - Средства статического анализа - инструменты для поиска ошибок и потенциальных уязвимостей в коде без его исполнения. Примеры: SpotBugs, Error Prone, Infer.
 - Средства анализа производительности - инструменты для измерения и оптимизации производительности приложений. Примеры: New Relic, Яндекс.Метрика, Google Analytics.
- Выбор инструментов зависит от требований и целей проекта, а также от предпочтений разработчика.

```
from math import sqrt

def area_of_triangle(a, b, c):
    s = (a + b + c) / 2
    return sqrt(s * (s - a) * (s - b) * (s - c))

a = float(input("Введите сторону a: "))
b = float(input("Введите сторону b: "))
c = float(input("Введите сторону c: "))
area = area_of_triangle(a, b, c)
print("Площадь треугольника:", area)
```

Критерии оценивания

Оценка 5 «отлично» - теоретическая часть: полные и правильные ответы на все вопросы, умение объяснить и аргументировать свою точку зрения, знание терминологии и основных понятий. Практическая часть: программа работает корректно и соответствует заданию; хорошо структурированный и понятный код; соблюдение основных правил и принципов программирования.

Оценка 4 «хорошо» - теоретическая часть: большинство ответов на вопросы правильные, но могут быть незначительные ошибки; частичное понимание и объяснение своей точки зрения. Практическая часть: программа в основном работает корректно, но есть незначительные ошибки; структура кода понятная, но может содержать незначительные ошибки или несоблюдение стандартов.

Оценка 3 «удовлетворительно» - теоретическая часть: ответы на вопросы содержат много ошибок или неполные; затруднения при объяснении своей точки зрения или непонимание основных понятий. Практическая часть: в программе есть существенные ошибки, которые делают ее неработоспособной; код может быть запутанным или содержать значительные ошибки в структуре и логике.

Оценка 2 «неудовлетворительно» - теоретическая часть: Неправильные ответы на большинство вопросов; отсутствие понимания и неспособность объяснить свою точку зрения или основные понятия. Практическая часть: программа не работает или содержит критические ошибки, которые не позволяют ее выполнить; непонятный или неструктурированный код, содержащий множество ошибок.

5. Экзаменационная ведомость.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

МДК 02.03 Математическое моделирование

подготовки специалистов среднего звена по специальности

код специальности 09.02.07 Информационные системы и программирование

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности СПО 09.02.07 «Информационные системы и программирование» базового уровня программы «Обеспечение качества функционирования компьютерных систем».

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчик:

Бахаев Денис Викторович, преподаватель

Голубенко Максим Игоревич, преподаватель

ОДОБРЕН

цикловой методической комиссией

Протокол № 1

от «3» сентября 2022 г.

Председатель Гаврикова Е.Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения МДК, подлежащие проверке
3. Оценка освоения МДК
 - 3.1. Контроль и оценка освоения МДК
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет материалов
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения МДК «Математическое моделирование» обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 «Информационные системы и программирование», базовый уровень следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями.

Обучающийся должен уметь:

У 1.	Использовать выбранную систему контроля версий
У 2.	Использовать методы для получения кода с заданной функциональностью и степенью качества

Обучающийся должен знать:

З 1	Модели процесса разработки программного обеспечения
З 2	Основные принципы процесса разработки программного обеспечения
З 3	Основные подходы к интегрированию программных модулей
З 4	Основы верификации и аттестации программного обеспечения

Обучающийся должен иметь практический опыт в: интеграции модулей в программное обеспечение, отладке программных модулей.

Формируемые ОК:

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 4. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языках.

Формируемые ПК:

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

Личностные результаты МДК

Код ЛР	Личностные результаты реализации программы (дескрипторы)
ЛР 1	- осознающий себя гражданином и защитником великой страны
ЛР 2	- проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно

	взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	- демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих
ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личного и профессионального, конструктивного «цифрового следа»
ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	- проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	- уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 10	- принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 13	- умение реализовывать личностные качества в производственном процессе

ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР 17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
ЛР 19	- профессиональная идентичность и ответственность
ЛР 20	- самооценка и рефлексия результатов своей деятельности и развития

Формами промежуточной аттестации по МДК являются: контрольная работа (3 семестр), экзамен (4 семестр).

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по МДК осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих и профессиональных компетенций:

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У 1. Использовать выбранную систему контроля версий; ОК 01, ОК 02, ОК 5, ОК 9, ПК 2.1, ПК 2.2, ПК 2.5.	- использует языки структурного программирования для создания независимых программ; - использует языки объектно-ориентированного программирования для создания независимых программ; -использует языки сценариев для создания независимых программ; - разрабатывает графический интерфейс приложения.
У 2. Использовать методы для получения кода с заданной функциональностью и степенью качества; ОК 01, ОК 02, ОК 04, ОК 5, ОК 09, ПК 2.1, ПК 2.3, ПК 2.4	- создает проект по разработке приложения и формулировать его задачи; - выполняет управление проектом с использованием инструментальных средств
Знать:	
З 1: Модели процесса разработки программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.5	- объясняет суть объектно-ориентированного программирования; - перечисляет спецификации языка; - объясняет алгоритм создания графического пользовательского интерфейса (GUI); - поясняет суть файловый ввод-вывод; -объясняет порядок создания сетевого сервера и сетевого клиента.
З 2: Основные принципы процесса разработки программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.1, ПК 2.4	- соблюдает регламенты по обновлению и техническому сопровождению обслуживаемой информационной системы;

	<ul style="list-style-type: none"> - объясняет состав и содержание работ на стадии ввода в действие информационной системы; - перечисляет основные задачи сопровождения информационной системы.
<p>3 3: Основные подходы к интегрированию программных модулей; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.2, ПК 2.4</p>	<ul style="list-style-type: none"> - объясняет основные процессы управления проектом разработки; - точно называет критерии оценки качества функционирования информационной системы; - объясняет виды испытания информационной системы на этапах ее сопровождения.
<p>34: Основы верификации и аттестации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.5</p>	<ul style="list-style-type: none"> - объясняет суть верификации программного обеспечения; - объясняет суть аттестации программного обеспечения; - поясняет разницу между верификацией и аттестацией

3.ОЦЕНКА ОСВОЕНИЯ МДК

3.1. Контроль и оценка освоения МДК по темам (разделам)

Элемент МДК	Формы и методы контроля			
	Текущий контроль		Промежуточная аттестация	
	Осваиваемые результаты	Метод контроля	Проверяемые результаты	Форма контроля
Тема 1. Основы моделирования. Детерминированные задачи	ОК 1, ОК 2, ОК 04 ОК 5, ОК 9, У 1, У 2, 3 1, 3 2, ПК 2.1. – 2.5	Устный опрос Самостоятельная работа Практическая работа Тестовое задание	ОК 1, ОК 2, ОК 04 ОК 5, ОК 9, У 1, У 2, 3 1, 3 2, ПК 2.1. – 2.5	7-й семестр – контрольная работа,
Тема 2. Задачи в условиях неопределенности	ОК 1, ОК 2, ОК 04, ОК 5, ОК 9, У 1, У 2, 3 3, 3 4, ПК 2.1. – 2.5	Устный опрос Практическая работа Тестовое задание	ОК 1, ОК 2, ОК 04, ОК 5, ОК 9, У 1, У 2, 3 3, 3 4, ПК 2.1. – 2.5	8 семестр – дифференцированный зачет

3.1.1. Методы и критерии оценки

1.Устный опрос. Критерии оценивания.

Оценка 5 «отлично» - ответил на вопросы в объеме лекционного и дополнительного материала, дал полные грамотные ответы на все дополнительные вопросы.

Оценка 4 «хорошо» - грамотно изложил ответы на вопросы, но содержание и формулировки имеют отдельные неточности (допускается нечеткая формулировка определений), в полной мере ответил на заданные дополнительные вопросы.

Оценка 3 «удовлетворительно» - ответил на часть вопросов в объеме лекционного материала и ответил на часть дополнительных вопросов.

Оценка 2 «неудовлетворительно» - допустил ошибки в определении базовых понятий, исказил их смысл, не ответил на дополнительные вопросы.

2. Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

3. Практическая работа. Критерии оценивания.

Оценка «5» - работа выполнена в полном объеме, с соблюдением алгоритма выполнения: последовательности проведения измерений, заполнения таблиц, графиков и др.; правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; получены результаты в соответствии с поставленной целью, отчет составлен по всем требованиям.

Оценка «4» - выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета, незначительные ошибки в составлении отчета.

Оценка «3» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью, значительные ошибки в оформлении отчета.

Оценка «2» - работа выполнена не полностью и объем выполненной части работы не позволяет получить никаких результатов в соответствии с поставленной целью, отчет не сформирован или сформирован неполно.

4. Самостоятельная работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме; учтены все требования к данной работе; самостоятельно поставлены цели и задачи работы, соответствующие заданной теме/проблеме; получены результаты в соответствии с поставленной целью; работа оформлена аккуратно и грамотно.

Оценка 4 «хорошо» - выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью, и объем выполненной части работы позволяет получить недостаточно результатов в соответствии с поставленной целью.

4. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Контрольная работа (3 семестр)

1. Форма проведения: тест

2. Условия выполнения

1. Инструкция для обучающихся.

2. Время выполнения задания: 45 минут.

3. Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.

4. Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда виртуализации VMWare Work station версии 12 и выше; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

5. Информационные источники: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».

6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения проверочной работы

3.1. Перечень тем, контролируемых в ходе промежуточной аттестации.

Тема 1. Основы моделирования. Детерминированные задачи.

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У 1. Использовать выбранную систему контроля версий; ОК 01, ОК 02, ОК 5, ОК 9, ПК 2.1, ПК 2.2, ПК 2.5.	- использует языки структурного программирования для создания независимых программ; - использует языки объектно-ориентированного программирования для создания независимых программ; -использует языки сценариев для создания независимых программ; - разрабатывает графический интерфейс приложения.
У 2. Использовать методы для получения кода с заданной функциональностью и степенью качества; ОК 01, ОК 02, ОК 04, ОК 5, ОК 09, ПК 2.1, ПК 2.3, ПК 2.4	- создает проект по разработке приложения и формулировать его задачи; - выполняет управление проектом с использованием инструментальных средств
Знать:	
З 1: Модели процесса разработки программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.5	- объясняет суть объектно-ориентированного программирования; - перечисляет спецификации языка; - объясняет алгоритм создания графического пользовательского интерфейса (GUI); - поясняет суть файловый ввод-вывод; -объясняет порядок создания сетевого сервера и сетевого клиента.
З 2: Основные принципы процесса разработки программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.1, ПК 2.4	- соблюдает регламенты по обновлению и техническому сопровождению обслуживаемой информационной системы; - объясняет состав и содержание работ на стадии ввода в действие информационной системы;

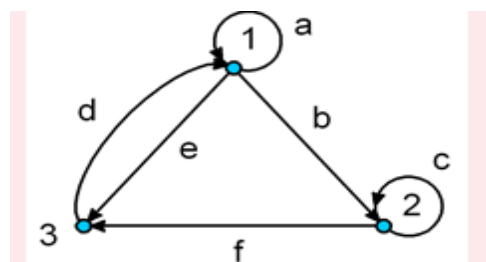
	- перечисляет основные задачи сопровождения информационной системы.
3 3: Основные подходы к интегрированию программных модулей; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.2, ПК 2.4	- объясняет основные процессы управления проектом разработки; - точно называет критерии оценки качества функционирования информационной системы; - объясняет виды испытания информационной системы на этапах ее сопровождения.
34: Основы верификации и аттестации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.5	- объясняет суть верификации программного обеспечения; - объясняет суть аттестации программного обеспечения; - поясняет разницу между верификацией и аттестацией

Образец тестового задания к контрольной работе

1. ... – это образец, служащий эталоном (стандартом) для серийного или массового воспроизведения, а также тип, марка какого-либо изделия, конструкции.
2. ... модель – модель, основанная на подобии явлений, имеющих различную физическую природу, но описываемых одинаковыми математическими уравнениями.
3. ... – это совокупность всех значений переборных переменных, при которых система уравнений или неравенств имеет хотя бы одно решение.
4. ... решение — решение, которое по тем или иным признакам предпочтительнее других.
5. ... моделью называется совокупность уравнений или других математических соотношений, отражающих основные свойства изучаемого объекта или явления в рамках принятой умозрительной физической модели и особенности его взаимодействия с окружающей средой на пространственно-временных границах области его локализации.
6. Соотнесите ситуацию и используемые математические модели:

1. Ситуация определенности	А) Линейное программирование
2. Ситуация рискованности	Б) Теория массового обслуживания
3. Ситуация неопределенности	В) Теория игр
7. Наиболее известным и широко применяемым на практике для решения общей задачи линейного программирования является
8. Множество всех допустимых решений системы задачи линейного программирования является
9. Задача линейного программирования может достигать максимального значения:
 - а) только в одной точке
 - б) в двух точках
 - в) во множестве точек
 - г) в одной или двух точках
 - д) в одной или во множестве точек
10. Транспортная задача является задачей программирования.

11. Задача, процесс нахождения решения которой является многоэтапным, относится к задачам:
 - а) линейного программирования
 - б) теории игр
 - в) динамического программирования
 - г) нелинейного программирования
 - д) параметрического программирования
12. Суть метода ... заключается в том, что нужно двигаться к минимуму в направлении наиболее быстрого убывания функции, определяемого антиградиентом.
13. В отличие от прямого симплекс – метода, двойственный симплекс – метод не требует нахождения начального базисного решения. Укажите, верно ли утверждение.
14. Задача о загрузке рюкзака является задачей программирования.
15. ... - метод оптимизации многошаговых задач в условиях отсутствия обратной связи (последствия) и аддитивности целевой функции.
16. Как называется линия без стрелки, соединяющая вершины графа?
17. Пусть граф отражает отношения «позвонила по телефону» между девочками Аня, Вера, Галя, Даша. Какой путь называется циклом:
 - а) Аня-Вера-Галя
 - б) Аня-Вера-Галя-Даша
 - в) Аня-Вера-Галя-Аня
18. Как называется направленная линия, соединяющая вершины графа?
19. Путь по рёбрам и вершинам графа, в который ребро графа входит один раз, называется ...
20. Какого типа граф изображен на рисунке?



4.Эталоны ответов

- 1) Модель; 2) аналоговая; 3) Множество решений; 4) Оптимальное; 5) Математической; 6) 1А, 2Б, 3В; 7) симплекс – метод; 8) выпуклым; 9) в; 10) линейного; 11) в; 12) градиентного спуска; 13) верно; 14) динамического; 15) Динамическое программирование; 16) ребро; 17) в; 18) дуга; 19) циклом; 20) оргграф.

Критерии оценки ответов

Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

2.Дифференцированный зачет

1.Форма проведения: письменная – тест и практико-ориентированное задание

2.Условия выполнения

1.Инструкция для обучающихся.

2. Время выполнения задания: 45 минут.
3. Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.
4. Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда виртуализации VMWare Work station версии 12 и выше; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.
5. Информационные источники: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».
6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения проверочной работы

3.1. Перечень тем, контролируемых в ходе промежуточной аттестации.

Тема 1. Основы моделирования. Детерминированные задачи.

Тема 2. Задачи в условиях неопределенности.

3.1. Практико-ориентированное задания к дифференцированному зачету:

Произведите анализ предметной области «Фирмы по оказанию бухгалтерских услуг». Опишите бизнес-процессы предметной области. Постройте концептуальную схему информационной системы.

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У 1. Использовать выбранную систему контроля версий; ОК 01, ОК 02, ОК 5, ОК 9, ПК 2.1, ПК 2.2, ПК 2.5.	- использует языки структурного программирования для создания независимых программ; - использует языки объектно-ориентированного программирования для создания независимых программ; -использует языки сценариев для создания независимых программ; - разрабатывает графический интерфейс приложения.
У 2. Использовать методы для получения кода с заданной функциональностью и степенью качества; ОК 01, ОК 02, ОК 04, ОК 5, ОК 09, ПК 2.1, ПК 2.3, ПК 2.4	- создает проект по разработке приложения и формулировать его задачи; - выполняет управление проектом с использованием инструментальных средств
Знать:	
З 1: Модели процесса разработки программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.5	- объясняет суть объектно-ориентированного программирования; - перечисляет спецификации языка; - объясняет алгоритм создания графического пользовательского интерфейса (GUI); - поясняет суть файловый ввод-вывод;

	-объясняет порядок создания сетевого сервера и сетевого клиента.
3 2: Основные принципы процесса разработки программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.1, ПК 2.4	- соблюдает регламенты по обновлению и техническому сопровождению обслуживаемой информационной системы; - объясняет состав и содержание работ на стадии ввода в действие информационной системы; - перечисляет основные задачи сопровождения информационной системы.
3 3: Основные подходы к интегрированию программных модулей; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.2, ПК 2.4	- объясняет основные процессы управления проектом разработки; - точно называет критерии оценки качества функционирования информационной системы; - объясняет виды испытания информационной системы на этапах ее сопровождения.
34: Основы верификации и аттестации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.5	- объясняет суть верификации программного обеспечения; - объясняет суть аттестации программного обеспечения; - поясняет разницу между верификацией и аттестацией

4. Эталоны ответов

Предметная область фирмы по оказанию бухгалтерских услуг включает в себя ряд ключевых бизнес-процессов, которые могут быть следующими:

1. Прием и анализ документов: Этот процесс включает в себя прием и обработку документов от клиентов, а также их анализ для определения необходимых бухгалтерских действий.

2. Ведение бухгалтерского учета: включает в себя осуществление бухгалтерских операций, ведение журналов учета, составление отчетности и обеспечение соответствия законодательству.

3. Консультации клиентов: Бухгалтерская фирма может предоставлять консультации своим клиентам по вопросам налогообложения, бухгалтерскому учету и другим финансовым вопросам.

4. Подготовка налоговой отчетности: включает в себя составление и сдачу налоговых деклараций, а также расчет и уплату налогов.

5. Аудит и проверка финансовой отчетности: В случае необходимости бухгалтерская фирма может проводить аудит финансовой отчетности клиентов для подтверждения ее достоверности и соответствия законодательству.

6. Управление клиентскими данными: включает в себя создание и поддержание базы данных клиентов, а также управление информацией о контрактах, услугах и т. д.

Концептуальная схема информационной системы для такой фирмы может включать следующие компоненты:

1. Модуль управления клиентской базой данных: содержит информацию о клиентах, контактных данных, истории взаимодействия и услугах, предоставляемых клиентам.

2. Модуль бухгалтерского учета: включает журналы учета, модули для ведения бухгалтерского учета и генерации отчетности.

3. Модуль налогового учета и отчетности: включает в себя функционал для составления и сдачи налоговых отчетов, расчета налогов и контроля за их уплатой.

4. Модуль консультаций и обратной связи с клиентами: предоставляет возможность консультировать клиентов, отвечать на их вопросы и предоставлять необходимую информацию.

5. Модуль аудита и контроля качества: позволяет проводить аудит финансовой отчетности, а также контролировать качество оказываемых услуг.

6. Модуль отчетности и аналитики: содержит инструменты для генерации различных отчетов, анализа финансовых данных и мониторинга ключевых показателей.

7. Модуль администрирования и безопасности: обеспечивает управление доступом к данным, резервное копирование, защиту от угроз информационной безопасности и т. д.

Эти модули взаимодействуют между собой, обеспечивая эффективное функционирование информационной системы и поддержку основных бизнес-процессов бухгалтерской фирмы.

Критерии оценки ответов, обучающихся

Оценка 5 «отлично» - дан развернутый полный ответ, присутствует четкое понимание методологии решения поставленной задачи.

Оценка 4 «хорошо» - дан развернутый ответ, присутствуют пробелы в понимании алгоритма решения задачи, не влияющие на основную суть.

Оценка 3 «удовлетворительно» - дан неполный ответ, присутствуют проблемы в понимании основ дисциплины.

Оценка 2 «неудовлетворительно» - дан неправильный ответ.

4.Комплект заданий.

5.Экзаменационная ведомость.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

ПМ.02 Осуществление интеграции программных модулей

подготовки специалистов среднего звена

код специальности 09.02.07 Информационные системы и программирование

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности СПО 09.02.07 Информационные системы и программирование профессионального модуля ПМ.02 Осуществление интеграции программных.

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчики:

Голубенко Максим Игоревич, преподаватель

Бахаев Денис Викторович преподаватель

ОДОБРЕН

цикловой методической комиссией

Протокол № 1

от «3» сентября 2022 г.

Председатель Гаврикова Е.Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения ПМ, подлежащие проверке
3. Оценка освоения ПМ
 - 3.1. Контроль и оценка освоения ПМ
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет материалов
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения ПМ.02 Осуществление интеграции программных компьютерных систем обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 Информационные системы и программирование следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями.

Обучающийся должен знать:

З 1	модели процесса разработки программного обеспечения;
З 2	основные принципы процесса разработки программного обеспечения;
З 3	основные подходы к интегрированию программных модулей;
З 4	основы верификации и аттестации программного обеспечения;
З 5	основные методы отладки;
З 6	методы и схемы обработки исключительных ситуаций;
З 7	приемы работы с инструментальными средствами тестирования и отладки;
З 8	стандарты качества программной документации;
З 9	основы организации инспектирования и верификации;
З 10	встроенные и основные специализированные инструменты анализа качества программных продуктов.

Обучающийся должен уметь:

У 1	использовать выбранную систему контроля версий;
У 2	использовать методы для получения кода с заданной функциональностью и степенью качества;
У 3	выполнять тестирование интеграции;
У 4	организовывать постобработку данных;
У 5	использовать приемы работы в системах контроля версий;
У 6	выполнять отладку, используя методы и инструменты условной компиляции;
У 7	выявлять ошибки в системных компонентах на основе спецификаций.

Обучающийся должен иметь практический опыт: в настройке отдельных компонентов программного обеспечения компьютерных систем; в выполнении отдельных видов работ на этапе поддержки программного обеспечения компьютерной системы.

Формируемые ОК:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

ВД 2. Осуществление интеграции программных модулей.

Формируемые ПК:

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2 Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

Личностные результаты с учетом особенностей учебной дисциплины:

Код ЛР	Личностные результаты реализации программы (дескрипторы)
ЛР 1	- осознающий себя гражданином и защитником великой страны
ЛР 2	- проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	- демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих
ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля

ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личностного и профессионального, конструктивного «цифрового следа»
ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	- проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	- уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 10	- принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР 17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
ЛР 19	- профессиональная идентичность и ответственность
ЛР 20	- самооценка и рефлексия результатов своей деятельности и развития

Промежуточной аттестации по ПМ является экзамен по билетам.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПМ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

Результатом освоения профессионального модуля является готовность обучающегося к выполнению вида профессиональной деятельности Сопровождение и обслуживание программного обеспечения компьютерных систем. Для подтверждения такой готовности обязательна констатация сформированности у обучающегося всех

профессиональных компетенций, входящих в состав профессионального модуля. Общие компетенции формируются в процессе освоения ОПОП в целом, поэтому по результатам освоения профессионального модуля возможно оценивание положительной динамики их формирования.

Оценка запланированных результатов по ПМ

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
МДК 02.01 Технология разработки программного обеспечения	
Уметь:	
<p>У 1 Использовать выбранную систему контроля версий ОК 1, ОК 02, ОК 4, ОК 9, ПК 2.3</p>	<p>Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. выполнять отладку программного модуля с использованием специализированных программных средств.</p>
<p>У 2 Использовать методы для получения кода с заданной функциональностью и степенью качества ОК 1, ОК 2, ОК 03, ОК 4, ОК 9, ПК 2.1, ПК 2.3</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств.</p>
<p>У 3 Выполнять тестирование интеграции ОК 1, ОК 2, ОК 05, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с</p>

	использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
У 4 Организовывать постобработку данных ОК 1, ОК 2, ОК 06, ОК 9, ПК 2.1	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.
У 5 Использовать приемы работы в системах контроля версий ОК 1, ОК 4, ОК 07, ОК 9, ПК 2.3	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. выполнять отладку программного модуля с использованием специализированных программных средств.
У 6 Выполнять отладку, используя методы и инструменты условной компиляции ОК 1, ОК 2, ОК 08, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
У 7 Выявлять ошибки в системных компонентах на основе спецификаций ОК 2, ОК 05, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4	использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных

	средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
Знать:	
3 1 Модели процесса разработки программного обеспечения ОК 1, ОК 2, ОК 4, ПК 2.1	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.
3 2 Основные принципы процесса разработки программного обеспечения ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.3	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение.
3 3 Основные подходы к интегрированию программных модулей ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение.
3 4 Основы верификации и аттестации программного обеспечения ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.4, ПК 2.5	использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным

	<p>модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>
<p>3 5 Основные методы отладки ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>3 6 Методы и схемы обработки исключительных ситуаций ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>3 7 Приемы работы с инструментальными средствами тестирования и отладки ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. выполнять отладку программного модуля с использованием специализированных программных</p>

	средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
3 8 Стандарты качества программной документации ОК 2, ОК 9, ПК 2.1	использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.
3 9 Основы организации инспектирования и верификации ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.5	использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
3 10 Встроенные и основные специализированные инструменты анализа качества программных продуктов ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4, ПК 2.5	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
МДК 02.02 Инструментальные средства разработки программного обеспечения	
Уметь:	
У 1 Использовать выбранную систему контроля версий	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; эффективно взаимодействовать и работать в коллективе и

<p>ОК 1, ОК 02, ОК 4, ОК 9, ПК 2.3</p>	<p>команде; пользоваться профессиональной документацией на государственном и иностранном языках. выполнять отладку программного модуля с использованием специализированных программных средств.</p>
<p>У 2 Использовать методы для получения кода с заданной функциональностью и степенью качества ОК 1, ОК 2, ОК 03, ОК 4, ОК 9, ПК 2.1, ПК 2.3</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств.</p>
<p>У 3 Выполнять тестирование интеграции ОК 1, ОК 2, ОК 05, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>У 4 Организовывать постобработку данных ОК 1, ОК 2, ОК 06, ОК 9, ПК 2.1</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>

<p>У 5 Использовать приемы работы в системах контроля версий ОК 1, ОК 4, ОК 07, ОК 9, ПК 2.3</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. выполнять отладку программного модуля с использованием специализированных программных средств.</p>
<p>У 6 Выполнять отладку, используя методы и инструменты условной компиляции ОК 1, ОК 2, ОК 08, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>У 7 Выявлять ошибки в системных компонентах на основе спецификаций ОК 2, ОК 05, ОК 9, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4</p>	<p>использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>Знать:</p>	
<p>З 1 Модели процесса разработки программного обеспечения ОК 1, ОК 2, ОК 4, ПК 2.1</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>

<p>3 2 Основные принципы процесса разработки программного обеспечения ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.3</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение.</p>
<p>3 3 Основные подходы к интегрированию программных модулей ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.2</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять интеграцию модулей в программное обеспечение.</p>
<p>3 4 Основы верификации и аттестации программного обеспечения ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.4, ПК 2.5</p>	<p>использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>
<p>3 5 Основные методы отладки ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном</p>

	<p>языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>3 6 Методы и схемы обработки исключительных ситуаций ОК 1, ОК 2, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>3 7 Приемы работы с инструментальными средствами тестирования и отладки ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.3, ПК 2.4</p>	<p>выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>
<p>3 8 Стандарты качества программной документации ОК 2, ОК 9, ПК 2.1</p>	<p>использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>
<p>3 9 Основы организации инспектирования и верификации</p>	<p>использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать</p>

ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.5	в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
3 10 Встроенные и основные специализированные инструменты анализа качества программных продуктов ОК 1, ОК 2, ОК 4, ОК 9, ПК 2.1, ПК 2.3, ПК 2.4, ПК 2.5	выбирать способы решения задач профессиональной деятельности применительно к различным контекстам; использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; эффективно взаимодействовать и работать в коллективе и команде; пользоваться профессиональной документацией на государственном и иностранном языках. разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент. выполнять отладку программного модуля с использованием специализированных программных средств. осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
МДК 02.03 Математическое моделирование	
Уметь:	
У 1. Использовать выбранную систему контроля версий; ОК 01, ОК 02, ОК 5, ОК 9, ПК 2.1, ПК 2.2, ПК 2.5.	<ul style="list-style-type: none"> - использует языки структурного программирования для создания независимых программ; - использует языки объектно-ориентированного программирования для создания независимых программ; -использует языки сценариев для создания независимых программ; - разрабатывает графический интерфейс приложения.
У 2. Использовать методы для получения кода с заданной функциональностью и степенью качества; ОК 01, ОК 02, ОК 04, ОК 5, ОК 09, ПК 2.1, ПК 2.3, ПК 2.4	<ul style="list-style-type: none"> - создает проект по разработке приложения и формулировать его задачи; - выполняет управление проектом с использованием инструментальных средств
Знать:	

<p>З 1: Модели процесса разработки программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.5</p>	<ul style="list-style-type: none"> - объясняет суть объектно-ориентированного программирования; - перечисляет спецификации языка; - объясняет алгоритм создания графического пользовательского интерфейса (GUI); - поясняет суть файловый ввод-вывод; -объясняет порядок создания сетевого сервера и сетевого клиента.
<p>З 2: Основные принципы процесса разработки программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.1, ПК 2.4</p>	<ul style="list-style-type: none"> - соблюдает регламенты по обновлению и техническому сопровождению обслуживаемой информационной системы; - объясняет состав и содержание работ на стадии ввода в действие информационной системы; - перечисляет основные задачи сопровождения информационной системы.
<p>З 3: Основные подходы к интегрированию программных модулей; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.2, ПК 2.4</p>	<ul style="list-style-type: none"> - объясняет основные процессы управления проектом разработки; - точно называет критерии оценки качества функционирования информационной системы; - объясняет виды испытания информационной системы на этапах ее сопровождения.
<p>З4: Основы верификации и аттестации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 2.5</p>	<ul style="list-style-type: none"> - объясняет суть верификации программного обеспечения; - объясняет суть аттестации программного обеспечения; - поясняет разницу между верификацией и аттестацией

3.КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ЭКЗАМЕНА ПО ПМ

1.Экзамен по билетам

1.Форма проведения: экзамен.

2.Условия выполнения

Время выполнения задания: 90 мин.

Оборудование учебного кабинета: автоматизированные рабочие места в количестве 15 шт.

Технические средства обучения: автоматизированные рабочие места в количестве 15 шт.

Информационные источники: отсутствуют.

Требования охраны труда: в соответствии с требованиями СНиП.

3.Пакет материалов для проведения экзамена:

1.3.1 Перечень тем, выносимых на экзамен:

1. Основные методы внедрения и анализа функционирования программного обеспечения
2. Загрузка и установка программного обеспечения
3. Основные методы обеспечения качества функционирования.
4. Методы и средства защиты компьютерных систем.

3.2. Перечень вопросов, выносимых на экзамен

1. Что такое стандартизация требований к программному обеспечению?

2. Какие основные понятия используются при стандартизации требований к программному обеспечению?
3. Какие преимущества дает стандартизация требований к программному обеспечению?
4. Какие организации занимаются стандартизацией требований к программному обеспечению?
5. Что такое описание и анализ требований?
6. Какие методы используются для описания и анализа требований?
7. Что такое диаграммы IDEF?
8. Что такое оценка качества программных средств?
9. Какие методы используются для оценки качества программных средств?
10. Какие характеристики программного продукта могут быть оценены при оценке качества?
11. Какие технологии и инструменты используются для интеграции приложений?
12. Какие преимущества дает использование API для интеграции приложений?
13. Что такое ESB и как он используется для интеграции приложений?
14. Какие проблемы могут возникнуть при интеграции приложений и как их можно решить?
15. Какие методы используются для синхронизации данных между приложениями?
16. Какие инструменты используются для тестирования программных средств?
17. Какие методы используются для анализа качества программных средств?
18. Какие метрики используются для оценки производительности программных средств?
19. Какие инструменты используются для автоматизации тестирования программных средств?
20. Какие проблемы могут возникнуть при тестировании программных средств и как их можно решить?
21. Что такое моделирование?
22. Какие виды моделей существуют?
23. Какие методы используются для моделирования?
24. Какие задачи относятся к детерминированным?
25. Какие преимущества дает моделирование?
26. Что такое задачи в условиях неопределенности?
27. Какие методы используются для решения задач в условиях неопределенности?
28. Какие виды неопределенности могут возникнуть при решении задач?
29. Какие задачи относятся к задачам в условиях неопределенности?
30. Какие преимущества дает использование методов решения задач в условиях неопределенности?

Эталоны ответов на вопросы:

1. Стандартизация требований к программному обеспечению – это процесс установления общих правил и норм, которым должно соответствовать программное обеспечение. Это позволяет обеспечить совместимость, безопасность и эффективность программных продуктов. В контексте темы "Современные технологии и инструменты интеграции" стандартизация требований к программному обеспечению играет важную роль, так как она позволяет обеспечить совместимость различных приложений и систем, что является ключевым фактором для успешной интеграции. Стандартизация требований к программному обеспечению также позволяет упростить процесс внедрения новых технологий и снизить затраты на разработку и поддержку программного обеспечения.
2. При стандартизации требований к программному обеспечению используются такие

понятия, как функциональные требования, нефункциональные требования, пользовательские требования, требования к безопасности, требования к производительности, требования к надежности и т.д. В контексте темы "Современные технологии и инструменты интеграции" эти понятия играют важную роль, так как они определяют основные характеристики и функциональность программного обеспечения, которые должны быть учтены при интеграции различных приложений и систем. Например, требования к безопасности могут включать в себя защиту от несанкционированного доступа, шифрование данных и другие меры, которые должны быть реализованы при интеграции приложений. Требования к производительности могут включать в себя определение максимальной скорости обработки данных, время отклика и другие параметры, которые должны быть учтены при интеграции приложений.

3. Стандартизация требований к программному обеспечению имеет ряд преимуществ, которые особенно важны в контексте темы "Современные технологии и инструменты интеграции". Во-первых, стандартизация требований к программному обеспечению позволяет обеспечить совместимость различных приложений и систем, что является ключевым фактором для успешной интеграции. Во-вторых, стандартизация требований к программному обеспечению упрощает процесс внедрения новых технологий и снижает затраты на разработку и поддержку программного обеспечения. В-третьих, стандартизация требований к программному обеспечению повышает безопасность и надежность программных продуктов, что особенно важно при интеграции приложений, которые обрабатывают конфиденциальную информацию или критически важные данные. В-четвертых, стандартизация требований к программному обеспечению позволяет упростить процесс коммуникации между заказчиком и разработчиком, так как все требования и ожидания от программного продукта четко определены и документированы.

4. Стандартизацией требований к программному обеспечению занимаются различные организации, такие как Международная организация по стандартизации (ISO), Международная электротехническая комиссия (IEC), Американский национальный институт стандартов (ANSI), Европейский комитет по стандартизации (CEN) и другие. В контексте темы "Современные технологии и инструменты интеграции" эти организации играют важную роль, так как они разрабатывают стандарты и рекомендации, которые определяют общие правила и нормы для разработки и интеграции программного обеспечения. Например, стандарты ISO/IEC 25010 и ISO/IEC 25012 определяют требования к качеству и безопасности программного обеспечения, которые должны быть учтены при интеграции приложений. Стандарты ISO/IEC 27001 и ISO/IEC 27002 определяют требования к информационной безопасности, которые должны быть учтены при интеграции приложений, обрабатывающих конфиденциальную информацию.

5. Описание и анализ требований – это процесс определения и документирования потребностей и ожиданий заказчика от разрабатываемого программного обеспечения. Он включает в себя сбор информации о функциональных и нефункциональных требованиях, а также анализ этих требований для определения их приоритета и взаимосвязи. В контексте темы "Инструментарий тестирования и анализа качества программных средств" описание и анализ требований играют важную роль, так как они определяют основные характеристики и функциональность программного обеспечения, которые должны быть проверены и оценены при тестировании и анализе качества. Например, функциональные требования определяют, какие функции и возможности должно предоставлять программное обеспечение, а нефункциональные требования определяют, как эти функции

и возможности должны быть реализованы, например, с точки зрения производительности, надежности или безопасности. Описание и анализ требований позволяют разработчикам и тестировщикам понять, что должно быть сделано, чтобы программное обеспечение соответствовало ожиданиям заказчика и удовлетворяло его потребности.

6. Для описания и анализа требований используются различные методы, такие как интервьюирование заказчика, анкетирование, наблюдение, анализ документов, моделирование и другие. Каждый метод имеет свои преимущества и недостатки, и выбор метода зависит от конкретной ситуации и требований проекта. В контексте темы "Инструментарий тестирования и анализа качества программных средств" методы описания и анализа требований играют важную роль, так как они позволяют разработчикам и тестировщикам получить полное представление о функциональности и характеристиках программного обеспечения, которые должны быть проверены и оценены при тестировании и анализе качества. Например, интервьюирование заказчика позволяет получить информацию о его потребностях и ожиданиях от программного обеспечения, а моделирование позволяет визуализировать и анализировать функциональность и взаимосвязь требований. Каждый метод имеет свои преимущества и недостатки, и выбор метода зависит от конкретной ситуации и требований проекта.

7. Диаграммы IDEF – это набор графических методов для моделирования бизнес-процессов и систем. Они используются для описания и анализа требований к программному обеспечению, а также для проектирования и документирования систем. Диаграммы IDEF включают в себя диаграммы IDEF0, IDEF1X, IDEF2 и другие. В контексте темы "Инструментарий тестирования и анализа качества программных средств" диаграммы IDEF играют важную роль, так как они позволяют визуализировать и анализировать функциональность и взаимосвязь требований. Например, диаграмма IDEF0 позволяет представить структуру и взаимодействие различных компонентов программного обеспечения, а диаграмма IDEF1X позволяет представить структуру базы данных и взаимосвязь между таблицами. Использование диаграмм IDEF позволяет разработчикам и тестировщикам получить полное представление о функциональности и характеристиках программного обеспечения, которые должны быть проверены и оценены при тестировании и анализе качества.

8. Оценка качества программных средств – это процесс определения соответствия программного продукта установленным требованиям и стандартам. Она включает в себя анализ функциональности, производительности, надежности, безопасности, удобства использования и других характеристик программного продукта. В контексте темы "Оценка качества программных средств" оценка качества играет важную роль, так как она позволяет определить, насколько хорошо программное обеспечение соответствует ожиданиям заказчика и удовлетворяет его потребности. Оценка качества также позволяет выявить возможные проблемы и противоречия в требованиях, а также определить, какие характеристики программного продукта требуют улучшения или доработки. В результате оценки качества разработчики и тестировщики получают информацию о том, какие аспекты программного обеспечения требуют дополнительной работы или изменений, чтобы достичь желаемого уровня качества.

9. Для оценки качества программных средств используются различные методы, такие как тестирование, анализ кода, экспертная оценка, метрики качества, бенчмаркинг и другие. Каждый метод имеет свои преимущества и недостатки, и выбор метода зависит от конкретной ситуации и требований проекта. В контексте темы "Оценка качества

программных средств" методы оценки качества играют важную роль, так как они позволяют разработчикам и тестировщикам получить информацию о том, насколько хорошо программное обеспечение соответствует ожиданиям заказчика и удовлетворяет его потребности. Например, тестирование позволяет проверить функциональность и надежность программного обеспечения, а анализ кода позволяет выявить возможные ошибки и проблемы в коде. Каждый метод имеет свои преимущества и недостатки, и выбор метода зависит от конкретной ситуации и требований проекта.

10. При оценке качества программного продукта могут быть оценены такие характеристики, как функциональность, производительность, надежность, безопасность, удобство использования, совместимость, масштабируемость, документированность и другие. Каждая характеристика имеет свои критерии оценки, которые могут быть определены в соответствии с требованиями проекта. В контексте темы "Оценка качества программных средств" оценка характеристик играет важную роль, так как она позволяет определить, насколько хорошо программное обеспечение соответствует ожиданиям заказчика и удовлетворяет его потребности. Например, функциональность программного обеспечения может быть оценена по количеству и качеству предоставляемых функций, а производительность может быть оценена по скорости обработки данных и времени отклика. Оценка характеристик позволяет разработчикам и тестировщикам получить информацию о том, какие аспекты программного обеспечения требуют дополнительной работы или изменений, чтобы достичь желаемого уровня качества.

11. Для интеграции приложений используются различные технологии и инструменты, такие как API, ESB, синхронизация данных и другие. API (Application Programming Interface) – это набор функций и процедур, которые позволяют приложениям взаимодействовать друг с другом. ESB (Enterprise Service Bus) – это программное обеспечение, которое позволяет объединять различные приложения и системы в единую инфраструктуру. Синхронизация данных – это процесс обновления и согласования данных между различными приложениями и системами. В контексте темы "Современные технологии и инструменты интеграции" использование этих технологий и инструментов играет важную роль, так как они позволяют обеспечить совместимость и взаимодействие между различными приложениями и системами. Например, API позволяет приложениям обмениваться данными и выполнять определенные функции, а ESB позволяет объединять различные приложения и системы в единую инфраструктуру, что упрощает процесс интеграции. Использование этих технологий и инструментов позволяет разработчикам и интеграторам создавать сложные системы, которые могут взаимодействовать друг с другом и обмениваться данными.

12. Использование API для интеграции приложений имеет ряд преимуществ. Во-первых, API позволяет приложениям обмениваться данными и выполнять определенные функции, что упрощает процесс интеграции. Во-вторых, API позволяет разработчикам создавать модульные и повторно используемые компоненты, которые могут быть использованы в различных приложениях. В-третьих, API обеспечивает стандартизацию и согласованность между различными приложениями и системами, что упрощает процесс интеграции. В-четвертых, API позволяет разработчикам создавать масштабируемые и гибкие системы, которые могут легко адаптироваться к изменяющимся требованиям бизнеса. В-пятых, API обеспечивает безопасность и защиту данных при обмене информацией между различными приложениями и системами. В контексте темы "Современные технологии и инструменты интеграции" использование API является одним из ключевых инструментов для

обеспечения совместимости и взаимодействия между различными приложениями и системами.

13. ESB (Enterprise Service Bus) – это программное обеспечение, которое позволяет объединять различные приложения и системы в единую инфраструктуру. ESB выполняет роль посредника между различными приложениями и системами, обеспечивая их взаимодействие и обмен данными. В контексте темы "Современные технологии и инструменты интеграции" ESB играет важную роль, так как он позволяет разработчикам и интеграторам создавать сложные системы, которые могут взаимодействовать друг с другом и обмениваться данными. ESB предоставляет набор сервисов и инструментов для управления потоками данных, маршрутизации сообщений, преобразования форматов данных и других операций, необходимых для интеграции приложений. Использование ESB позволяет упростить процесс интеграции, обеспечить совместимость и взаимодействие между различными приложениями и системами, а также создать масштабируемые и гибкие системы, которые могут легко адаптироваться к изменяющимся требованиям бизнеса.

14. При интеграции приложений могут возникнуть различные проблемы, такие как несовместимость форматов данных, проблемы безопасности, проблемы производительности и другие. В контексте темы "Современные технологии и инструменты интеграции" эти проблемы могут быть решены с помощью использования соответствующих технологий и инструментов. Например, для решения проблемы несовместимости форматов данных можно использовать преобразование форматов данных или создание промежуточных слоев, которые обеспечивают совместимость между различными приложениями и системами. Для решения проблем безопасности можно использовать шифрование данных, аутентификацию и авторизацию пользователей, а также другие меры безопасности. Для решения проблем производительности можно использовать оптимизацию запросов к базам данных, кэширование данных и другие методы. В целом, использование современных технологий и инструментов интеграции позволяет разработчикам и интеграторам создавать сложные системы, которые могут взаимодействовать друг с другом и обмениваться данными, а также решать возникающие проблемы и обеспечивать совместимость и безопасность при интеграции приложений.

15. Для синхронизации данных между приложениями используются различные методы, такие как репликация данных, кэширование данных, асинхронная синхронизация и другие. Репликация данных – это процесс создания копий данных в различных приложениях или системах, что позволяет обеспечить доступность данных в случае сбоев или отказов. Кэширование данных – это процесс хранения данных в промежуточном слое, который обеспечивает быстрый доступ к данным и уменьшает нагрузку на основную систему. Асинхронная синхронизация – это процесс обновления и согласования данных между различными приложениями и системами в фоновом режиме, что позволяет обеспечить актуальность данных и уменьшить время ожидания пользователей. В контексте темы "Современные технологии и инструменты интеграции" использование этих методов позволяет разработчикам и интеграторам создавать сложные системы, которые могут взаимодействовать друг с другом и обмениваться данными, а также обеспечивать актуальность и доступность данных в различных приложениях и системах.

16. Для тестирования программных средств используются различные инструменты, такие как автоматические тестовые фреймворки, инструменты для функционального и нефункционального тестирования, инструменты для тестирования производительности, инструменты для тестирования безопасности и другие. Автоматические тестовые

фреймворки позволяют автоматизировать процесс тестирования, что позволяет ускорить процесс тестирования и повысить его эффективность. Инструменты для функционального и нефункционального тестирования позволяют проверить соответствие программного обеспечения функциональным и нефункциональным требованиям. Инструменты для тестирования производительности позволяют оценить производительность программного обеспечения и выявить узкие места в его работе. Инструменты для тестирования безопасности позволяют проверить соответствие программного обеспечения требованиям безопасности и выявить возможные уязвимости. В контексте темы "Инструментарий тестирования и анализа качества программных средств" использование этих инструментов позволяет разработчикам и тестировщикам получить полную информацию о качестве программного обеспечения и выявить возможные проблемы и противоречия в требованиях.

17. Для анализа качества программных средств используются различные методы, такие как тестирование, анализ кода, экспертная оценка, метрики качества и другие. Тестирование позволяет проверить функциональность и надежность программного обеспечения, а также выявить возможные ошибки и проблемы в его работе. Анализ кода позволяет выявить возможные ошибки и проблемы в коде программного обеспечения, а также оценить его качество и читаемость. Экспертная оценка позволяет получить независимую оценку качества программного обеспечения от опытных специалистов. Метрики качества позволяют измерить различные характеристики программного обеспечения, такие как производительность, надежность, безопасность и другие. В контексте темы "Инструментарий тестирования и анализа качества программных средств" использование этих методов позволяет разработчикам и тестировщикам получить полную информацию о качестве программного обеспечения и выявить возможные проблемы и противоречия в требованиях.

18. Для оценки производительности программных средств используются различные метрики, такие как время отклика, скорость обработки данных, пропускная способность, использование ресурсов и другие. Время отклика – это время, которое требуется программному обеспечению для выполнения определенной операции. Скорость обработки данных - это количество данных, которое программное обеспечение может обработать за определенный промежуток времени. Пропускная способность – это количество запросов или транзакций, которые программное обеспечение может обработать за определенный промежуток времени. Использование ресурсов – это количество ресурсов, которое программное обеспечение использует для своей работы, таких как память, процессорное время и другие. В контексте темы "Инструментарий тестирования и анализа качества программных средств" использование этих метрик позволяет разработчикам и тестировщикам получить полную информацию о производительности программного обеспечения и выявить возможные проблемы и противоречия в требованиях.

19. Для автоматизации тестирования программных средств используются различные инструменты, такие как Selenium, JUnit, TestNG, Cucumber, Robot Framework и другие. Selenium – это инструмент для автоматизации тестирования веб-приложений, который позволяет создавать тесты на различных языках программирования. JUnit и TestNG – это инструменты для автоматизации тестирования на языке Java. Cucumber – это инструмент для автоматизации тестирования на языке Ruby. Robot Framework – это инструмент для автоматизации тестирования на различных языках программирования. В контексте темы "Инструментарий тестирования и анализа качества программных средств" использование этих инструментов позволяет разработчикам и тестировщикам автоматизировать процесс

тестирования, что позволяет ускорить процесс тестирования и повысить его эффективность.

20. При тестировании программных средств могут возникнуть различные проблемы, такие как неполное покрытие тестами, некорректные тестовые данные, неполное понимание требований, нехватка ресурсов и другие. Для решения этих проблем можно использовать следующие подходы: неполное покрытие тестами: для решения этой проблемы можно использовать методы тестирования, такие как модульное тестирование, интеграционное тестирование, системное тестирование и другие, чтобы обеспечить полное покрытие тестами всех функциональных возможностей программного обеспечения. Некорректные тестовые данные: для решения этой проблемы можно использовать методы генерации тестовых данных, такие как случайные значения, значения из диапазона, значения из списка и другие, чтобы обеспечить корректность тестовых данных. Неполное понимание требований: для решения этой проблемы можно использовать методы анализа требований, такие как интервьюирование заказчика, анализ документов, моделирование и другие, чтобы обеспечить полное понимание требований к программному обеспечению. Нехватка ресурсов: для решения этой проблемы можно использовать методы оптимизации тестирования, такие как параллельное тестирование, тестирование в облаке, использование виртуальных машин и другие, чтобы обеспечить достаточное количество ресурсов для тестирования программного обеспечения. В целом, использование различных методов и инструментов тестирования позволяет разработчикам и тестировщикам получить полную информацию о качестве программного обеспечения и выявить возможные проблемы и противоречия в требованиях.

21. Моделирование – это процесс создания моделей, которые представляют реальные объекты, процессы или системы. Моделирование позволяет упростить и абстрагировать сложные системы, чтобы лучше понять их структуру и поведение. В контексте темы "Основы моделирования. Детерминированные задачи" моделирование играет важную роль, так как оно позволяет анализировать и оптимизировать детерминированные задачи, которые имеют четко определенные входные данные и результаты. Моделирование может быть использовано для различных целей, таких как планирование, прогнозирование, оптимизация, анализ и другие. В результате моделирования можно получить информацию о поведении системы, ее характеристиках и возможностях, что позволяет принимать обоснованные решения и разрабатывать эффективные стратегии.

22. Существует множество видов моделей, которые могут быть использованы для моделирования различных объектов, процессов или систем. Некоторые из них включают: Модели данных: Модели данных представляют структуру и организацию данных в системе. Они могут быть представлены в виде таблиц, графов, деревьев и других структур. Модели процессов: Модели процессов представляют последовательность действий или операций, которые происходят в системе. Они могут быть представлены в виде диаграмм потоков, блок-схем, графов и других структур. Модели систем: Модели систем представляют структуру и взаимодействие компонентов системы. Они могут быть представлены в виде диаграмм классов, диаграмм последовательностей, диаграмм состояний и других структур. Модели объектов: Модели объектов представляют структуру и поведение объектов в системе. Они могут быть представлены в виде классов, объектов, методов и других структур. Модели событий: Модели событий представляют последовательность событий, которые происходят в системе. Они могут быть представлены в виде диаграмм состояний, графов и других структур. Модели временных рядов: Модели временных рядов

представляют изменение значений во времени. Они могут быть представлены в виде графиков, диаграмм и других структур. Модели вероятностные: Модели вероятностные представляют случайные процессы и события. Они могут быть представлены в виде графов, диаграмм и других структур.

23. Существует множество методов, которые могут быть использованы для моделирования различных объектов, процессов или систем. Некоторые из них включают: Методы анализа: Методы анализа позволяют анализировать и оптимизировать детерминированные задачи, которые имеют четко определенные входные данные и результаты. Они могут быть использованы для планирования, прогнозирования, оптимизации и других целей. Методы имитационного моделирования: Методы имитационного моделирования позволяют создавать модели, которые имитируют поведение реальных объектов, процессов или систем. Они могут быть использованы для анализа и оптимизации систем, которые имеют сложные взаимодействия и зависимости. Методы статистического моделирования: Методы статистического моделирования позволяют создавать модели, которые описывают статистические свойства объектов, процессов или систем. Они могут быть использованы для анализа и прогнозирования данных, которые имеют случайные компоненты. Методы экспертных оценок: Методы экспертных оценок позволяют использовать знания и опыт экспертов для создания моделей, которые описывают поведение объектов, процессов или систем. Они могут быть использованы для анализа и оптимизации систем, которые имеют сложные взаимодействия и зависимости. Методы машинного обучения: Методы машинного обучения позволяют создавать модели, которые автоматически обучаются на основе данных. Они могут быть использованы для анализа и прогнозирования данных, которые имеют случайные компоненты.

24. Задачи, которые относятся к детерминированным, имеют четко определенные входные данные и результаты. Они могут быть решены с помощью методов моделирования, таких как анализ, имитационное моделирование, статистическое моделирование, экспертные оценки и другие. Примеры детерминированных задач включают: Планирование производства: Задача планирования производства имеет четко определенные входные данные, такие как количество заказов, доступные ресурсы и ограничения, и результаты, такие как план производства. Прогнозирование спроса: Задача прогнозирования спроса имеет четко определенные входные данные, такие как исторические данные о продажах, сезонные факторы и другие переменные, и результаты, такие как прогноз спроса. Оптимизация маршрутов: Задача оптимизации маршрутов имеет четко определенные входные данные, такие как начальная и конечная точки, ограничения по времени и ресурсам, и результаты, такие как оптимальный маршрут. Анализ данных: Задача анализа данных имеет четко определенные входные данные, такие как набор данных, переменные и метрики, и результаты, такие как выводы и рекомендации на основе анализа данных. Оптимизация процессов: Задача оптимизации процессов имеет четко определенные входные данные, такие как текущие процессы, ограничения и цели, и результаты, такие как оптимизированный процесс.

25. Моделирование имеет ряд преимуществ, которые могут быть полезны при анализе и оптимизации детерминированных задач. Некоторые из них включают: Упрощение сложных систем: Моделирование позволяет упростить и абстрагировать сложные системы, чтобы лучше понять их структуру и поведение. Анализ и оптимизация: Моделирование позволяет анализировать и оптимизировать детерминированные задачи, которые имеют четко определенные входные данные и результаты. Прогнозирование: Моделирование

позволяет прогнозировать поведение системы на основе входных данных и результатов. Принятие обоснованных решений: Моделирование позволяет принимать обоснованные решения на основе анализа и оптимизации детерминированных задач. Разработка эффективных стратегий: Моделирование позволяет разрабатывать эффективные стратегии на основе анализа и оптимизации детерминированных задач.

26. Задачи в условиях неопределенности – это задачи, в которых входные данные или результаты не являются четко определенными или предсказуемыми. Они могут быть решены с помощью методов моделирования, таких как имитационное моделирование, статистическое моделирование, экспертные оценки и другие. Примеры задач в условиях неопределенности включают: Управление рисками: Задача управления рисками имеет неопределенные входные данные, такие как вероятность возникновения рисков, их влияние на бизнес и другие переменные, и результаты, такие как стратегия управления рисками. Оптимизация портфеля инвестиций: Задача оптимизации портфеля инвестиций имеет неопределенные входные данные, такие как доходность различных активов, их корреляция и другие переменные, и результаты, такие как оптимальный портфель инвестиций. Оптимизация цепочки поставок: Задача оптимизации цепочки поставок имеет неопределенные входные данные, такие как спрос на товары, доступность ресурсов и другие переменные, и результаты, такие как оптимальная структура цепочки поставок. Управление проектами: Задача управления проектами имеет неопределенные входные данные, такие как сроки выполнения задач, доступность ресурсов и другие переменные, и результаты, такие как план управления проектом.

27. Для решения задач в условиях неопределенности могут быть использованы различные методы, такие как имитационное моделирование, статистическое моделирование, экспертные оценки и другие. Имитационное моделирование: Имитационное моделирование позволяет создавать модели, которые имитируют поведение реальных объектов, процессов или систем в условиях неопределенности. Оно может быть использовано для анализа и оптимизации систем, которые имеют сложные взаимодействия и зависимости. Статистическое моделирование: Статистическое моделирование позволяет создавать модели, которые описывают статистические свойства объектов, процессов или систем в условиях неопределенности. Оно может быть использовано для анализа и прогнозирования данных, которые имеют случайные компоненты. Экспертные оценки: Экспертные оценки позволяют использовать знания и опыт экспертов для создания моделей, которые описывают поведение объектов, процессов или систем в условиях неопределенности. Они могут быть использованы для анализа и оптимизации систем, которые имеют сложные взаимодействия и зависимости. Методы машинного обучения: Методы машинного обучения позволяют создавать модели, которые автоматически обучаются на основе данных в условиях неопределенности. Они могут быть использованы для анализа и прогнозирования данных, которые имеют случайные компоненты.

28. При решении задач в условиях неопределенности могут возникнуть различные виды неопределенности, такие как: Неопределенность входных данных: Входные данные могут быть неполными, неточными или непредсказуемыми. Например, спрос на товары может быть неопределенным из-за изменения предпочтений потребителей или экономических условий. Неопределенность результатов: Результаты могут быть непредсказуемыми или неоднозначными. Например, доходность инвестиций может быть неопределенной из-за изменения рыночных условий или экономической ситуации. Неопределенность взаимодействий: Взаимодействия между различными элементами системы могут быть

неопределенными или непредсказуемыми. Например, взаимодействие между различными компонентами цепочки поставок может быть неопределенным из-за изменения спроса на товары или доступности ресурсов. Неопределенность ограничений: Ограничения, такие как доступность ресурсов или сроки выполнения задач, могут быть неопределенными или непредсказуемыми. Например, доступность рабочей силы может быть неопределенной из-за изменения экономической ситуации или изменения законодательства.

29. Задачи, которые относятся к задачам в условиях неопределенности, имеют неопределенные входные данные или результаты. Они могут быть решены с помощью методов моделирования, таких как имитационное моделирование, статистическое моделирование, экспертные оценки и другие. Примеры задач в условиях неопределенности включают: Управление рисками: Задача управления рисками имеет неопределенные входные данные, такие как вероятность возникновения рисков, их влияние на бизнес и другие переменные, и результаты, такие как стратегия управления рисками. Оптимизация портфеля инвестиций: Задача оптимизации портфеля инвестиций имеет неопределенные входные данные, такие как доходность различных активов, их корреляция и другие переменные, и результаты, такие как оптимальный портфель инвестиций. Оптимизация цепочки поставок: Задача оптимизации цепочки поставок имеет неопределенные входные данные, такие как спрос на товары, доступность ресурсов и другие переменные, и результаты, такие как оптимальная структура цепочки поставок. Управление проектами: Задача управления проектами имеет неопределенные входные данные, такие как сроки выполнения задач, доступность ресурсов и другие переменные, и результаты, такие как план управления проектом.

30. Использование методов решения задач в условиях неопределенности имеет ряд преимуществ, которые могут быть полезны при анализе и оптимизации задач. Некоторые из них включают: Улучшение принятия решений: Использование методов решения задач в условиях неопределенности позволяет принимать обоснованные решения на основе анализа и оптимизации задач. Уменьшение рисков: Использование методов решения задач в условиях неопределенности позволяет уменьшить риски, связанные с неопределенностью входных данных или результатов. Повышение эффективности: Использование методов решения задач в условиях неопределенности позволяет повысить эффективность задач, связанных с управлением рисками, оптимизацией портфеля инвестиций, оптимизацией цепочки поставок и управлением проектами. Улучшение прогнозирования: Использование методов решения задач в условиях неопределенности позволяет улучшить прогнозирование поведения системы на основе входных данных и результатов. Разработка эффективных стратегий: Использование методов решения задач в условиях неопределенности позволяет разрабатывать эффективные стратегии на основе анализа и оптимизации задач.

3.3. Практико-ориентированные задания к экзамену:

1. Напишите программу, которая проверяет, соответствует ли имя файла определенному шаблону. Например, имя файла должно начинаться с буквы и заканчиваться цифрой.
2. Напишите программу, которая проверяет, соответствует ли содержимое файла определенному формату. Например, все строки должны быть разделены символом новой строки.
3. Напишите программу, которая проверяет, соответствует ли размер файла определенному диапазону. Например, размер файла должен быть между 100 КБ и 1 МБ.
4. Напишите программу, которая проверяет, соответствует ли дата создания файла

определенному диапазону. Например, файл должен быть создан в течение последних 30 дней.

5. Напишите программу, которая будет выполнять юнит-тестирование некоторой функции. Функция должна принимать на вход список чисел и возвращать сумму этих чисел. Напишите несколько тестов для проверки работы этой функции (например, с пустым списком, с положительными и отрицательными числами).

6. Создайте скрипт, который будет анализировать код на Python с помощью статического анализатора. Например, используйте библиотеку `pylint` для поиска потенциальных проблем в коде. Запустите анализатор на некотором файле с кодом и выведите полученные результаты.

7. Напишите программу, которая будет выполнять функциональное тестирование веб-приложения. Используйте библиотеку `Selenium WebDriver` для автоматизации тестов. Например, напишите тест, который будет проверять корректность заполнения формы на веб-странице (например, ввод имени и отправка формы).

8. Напишите скрипт на Python, который будет использовать библиотеку `pytest` для автоматизации тестирования некоторой функциональности программы. Например, функция должна производить математические вычисления. Напишите несколько тестов, чтобы проверить правильность работы этой функции.

9. Создайте программу для анализа производительности кода на Python. Используйте библиотеку `timeit` для измерения времени выполнения определенной операции или функции. Напишите функцию, которая выполняет какое-то сложное вычисление, и замерьте время выполнения этой функции с разными наборами данных. Выведите результаты анализа.

10. Напишите программу на Python, которая будет считывать текстовый файл с требованиями к определенному программному продукту. Программа должна анализировать текст и выделять ключевые требования, такие как функциональные и нефункциональные характеристики. Выведите на экран полученные требования.

11. Разработайте скрипт на Python, который будет создавать диаграммы IDEF0 для моделирования бизнес-процессов. Используйте библиотеку `matplotlib` для визуализации диаграмм. Создайте несколько блоков с функциями и процессами, связанных стрелками, чтобы продемонстрировать процесс выполнения определенной задачи.

12. Напишите программу, которая будет анализировать требования к пользовательскому интерфейсу программного продукта. Программа должна искать ключевые слова и предложения, указывающие на требования к интерфейсу (например, "кнопка", "вкладка", "меню"). Выведите количество упоминаний каждого элемента интерфейса и сформируйте отчет.

13. Напишите программу, которая будет анализировать код другой программы на Python и выдавать метрики качества кода, такие как количество строк, сложность модуля, уровень вложенности и длина функций.

14. Разработайте скрипт, который будет сравнивать две версии одной и той же программы на Python и выдавать отчет о различиях в качестве кода, таких как изменения в структуре программы, удаление или добавление функций, а также изменения в комментариях.

15. Напишите программу, которая будет проводить автоматическое тестирование других программ на Python по заданным критериям качества кода, таким как правильность использования переменных, соответствие стандартам кодирования, наличие документации и комментариев.

16. Напишите программу на Python, которая использует библиотеку requests для выполнения HTTP-запроса к выбранному веб-сервису. Результат запроса должен быть обработан и выведен в удобочитаемом формате.
17. Создайте скрипт на Python, который использует библиотеку json для работы с данными в формате JSON. Программа должна считать данные из JSON-файла, обработать их и сохранить в новый файл в другом формате.
18. Разработайте программу на Python, использующую библиотеку Flask для создания простого веб-сервиса API. Сервис должен предоставлять возможность получать и отправлять данные по HTTP-запросам.
19. Напишите скрипт на Python, который будет использовать библиотеку SQLite3 для работы с базой данных SQLite. Программа должна выполнять CRUD-операции (Create, Read, Update, Delete) с данными в базе.
20. Создайте программу на Python, использующую библиотеку pySerial для взаимодействия с устройством через последовательный порт. Программа должна отправлять команды устройству и получать данные от него.
21. Напишите программу на Python для моделирования детерминированного процесса с использованием уравнения вида $y = ax + b$, где a и b - заданные коэффициенты.
22. Реализуйте программу, которая моделирует детерминированную картину движения точки по прямой с использованием уравнения $x = vt$, где v - постоянная скорость, t - время. Программа должна выводить координаты точки для разных значений времени.
23. Создайте программу для моделирования детерминированной математической функции с использованием библиотеки matplotlib. Программа должна строить график функции $y = f(x)$ для заданного интервала значений x .
24. Напишите программу, которая использует детерминированную модель для прогнозирования будущих значений временного ряда. Программа должна использовать метод наименьших квадратов для подгонки функции к заданным данным и предсказания будущих значений.
25. Разработайте программу для моделирования детерминированных динамических систем с использованием библиотеки scipy. Программа должна решать систему дифференциальных уравнений и выводить графики изменения состояний системы со временем.
26. Напишите программу, которая будет случайным образом выбирать число от 1 до 10, а затем запрашивать у пользователя угадать это число. При этом программа должна давать подсказки в случае неверного ответа (например, "Ваше число слишком большое" или "Ваше число слишком маленькое").
27. Создайте программу, которая будет генерировать случайный пароль заданной длины (например, 8 символов) из букв и цифр. При этом программа должна предоставлять пользователю возможность выбрать, должен ли пароль содержать только цифры, только буквы или и то, и другое.
28. Напишите программу, которая будет симулировать бросок игральной кости. При этом пользователь должен иметь возможность указать количество бросков и выбрать, хочет ли он увидеть результат каждого броска или только общий итог.
29. Разработайте игру "Камень, ножницы, бумага" против компьютера. Пользователь должен иметь возможность вводить свой выбор, а затем программа должна случайным образом выбирать выбор компьютера и сообщать результат раунда.
30. Создайте программу, которая будет генерировать случайные целые числа в заданном

диапазоне и проверять, кратны ли они заданному числу. Пользователь должен иметь возможность указать диапазон чисел и число, на которое они должны быть кратны. программа должна выводить результат для каждого сгенерированного числа.

Эталоны ответов на практические задания:

1.

```
import re

def check_filename_pattern(filename):
    pattern = r'^[a-zA-Z].*\d$' # шаблон: начинается с буквы и заканчивается цифрой
    if re.match(pattern, filename):
        print(f'Имя файла '{filename}' соответствует заданному шаблону.")
    else:
        print(f'Имя файла '{filename}' не соответствует заданному шаблону.")

filename = "example1.txt"
check_filename_pattern(filename) # Вывод: "Имя файла 'example1.txt' соответствует
заданному шаблону."

filename = "2example.txt"
check_filename_pattern(filename) # Вывод: "Имя файла '2example.txt' не соответствует
заданному шаблону."
```

2.

```
def check_file_format(file_path):
    try:
        with open(file_path, 'r') as file:
            lines = file.readlines()
            for line in lines:
                if not line.endswith('\n'):
                    return False
            return True
    except FileNotFoundError:
        print("Файл не найден")
        return False

file_path = 'file_to_check.txt' # замените 'file_to_check.txt' на путь к файлу, который вы
хотите проверить
if check_file_format(file_path):
    print("Формат файла соответствует требуемому")
else:
    print("Формат файла не соответствует требуемому")
```

3.

```
def check_file_size(file_size):
    if 100 <= file_size <= 1000:
        return True
    else:
```

```
    return False

file_size = 500 # Размер файла в КБ
if check_file_size(file_size):
    print("Размер файла соответствует диапазону")
else:
    print("Размер файла не соответствует диапазону")
```

4.

```
import os
import datetime

def check_file_creation_date(file_path):
    file_stat = os.stat(file_path)
    creation_time = file_stat.st_ctime
    file_creation_date = datetime.datetime.fromtimestamp(creation_time)
    thirty_days_ago = datetime.datetime.now() - datetime.timedelta(days=30)
    if file_creation_date > thirty_days_ago:
        return True
    else:
        return False

file_path = "path_to_your_file" # Укажите путь к вашему файлу
if check_file_creation_date(file_path):
    print("Файл был создан в течение последних 30 дней")
else:
    print("Файл был создан более 30 дней назад")
```

5.

```
import unittest

def sum_of_numbers(numbers):
    return sum(numbers)

class TestSumOfNumbers(unittest.TestCase):
    def test_empty_list(self):
        self.assertEqual(sum_of_numbers([]), 0)

    def test_positive_numbers(self):
        self.assertEqual(sum_of_numbers([1, 2, 3, 4, 5]), 15)

    def test_negative_numbers(self):
        self.assertEqual(sum_of_numbers([-1, -2, -3, -4, -5]), -15)

if __name__ == '__main__':
    unittest.main()
```

6.

```
import pylint.lint

def analyze_code(file_path):
    print(f"Анализируем файл {file_path}...")
    (pylint_score, _) = pylint.lint.Run([file_path], do_exit=False)
    print(f"Результат анализа: {pylint_score}")

if __name__ == "__main__":
    file_path = "path/to/your/file.py" # замените на путь к вашему файлу с кодом
    analyze_code(file_path)
```

7.

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time

# Запуск веб-браузера
driver = webdriver.Chrome()

# Переход на веб-страницу для тестирования
driver.get("https://www.example.com")

# Нахождение поля ввода для имени
name_input = driver.find_element_by_id("name") # Предположим, у нас есть поле ввода с id="name"

# Ввод имени
name_input.send_keys("John Doe")

# Ожидание
time.sleep(2)

# Отправка формы
submit_button = driver.find_element_by_id("submit") # Предположим, у нас есть кнопка отправки с id="submit"
submit_button.click()

# Проверка результата
result_message = driver.find_element_by_id("result") # Предположим, у нас есть элемент с id="result" для отображения результата
assert "Form submitted successfully" in driver.page_source

# Закрытие браузера
driver.quit()
```

8.

```
# Файл test_math_functions.py
```



```
import pytest
from math_functions import *

# Тест для функции сложения
def test_addition():
    assert addition(3, 5) == 8
    assert addition(-2, 2) == 0
    assert addition(0, 0) == 0

# Тест для функции вычитания
def test_subtraction():
    assert subtraction(10, 3) == 7
    assert subtraction(5, 5) == 0
    assert subtraction(-5, 7) == -12

# Тест для функции умножения
def test_multiplication():
    assert multiplication(4, 5) == 20
    assert multiplication(-3, 4) == -12
    assert multiplication(0, 10) == 0

# Тест для функции деления
def test_division():
    assert division(20, 4) == 5
    assert division(7, 2) == 3.5
    assert division(10, 0) == "Error: division by zero"
```

python

Copy code

```
# Файл math_functions.py
def addition(a, b):
    return a + b

def subtraction(a, b):
    return a - b

def multiplication(a, b):
    return a * b

def division(a, b):
    if b == 0:
        return "Error: division by zero"
    else:
        return a / b
```

Теперь можно запустить тесты с помощью pytest:

Ввести в командной строке:
pytest test_math_functions.py

9.

```
import timeit

# Функция для сложного вычисления
def complex_calculation(n):
    result = 0
    for i in range(n):
        result += i
    return result

# Замер времени выполнения функции
def measure_time(func, *args):
    start_time = timeit.default_timer()
    func(*args)
    end_time = timeit.default_timer()
    return end_time - start_time

# Разные наборы данных
inputs = [100, 1000, 10000]

# Замер времени выполнения функции для разных наборов данных
for n in inputs:
    time_taken = measure_time(complex_calculation, n)
    print(f"Время выполнения для n={n}: {time_taken:.6f} секунд")
```

10.

```
def extract_requirements(file_path):
    with open(file_path, 'r') as file:
        text = file.read()

    # Выполните анализ текста и извлеките ключевые требования здесь
    # Например, используйте регулярные выражения для поиска функциональных и
    нефункциональных требований

    functional_requirements = []
    non_functional_requirements = []

    # Пример регулярных выражений для поиска функциональных требований
    # functional_requirements = re.findall(r'(?i)functional requirement: (.+)', text)
    # Пример регулярных выражений для поиска нефункциональных требований
    # non_functional_requirements = re.findall(r'(?i)non-functional requirement: (.+)', text)

    # Вывод найденных требований
```

```

print("Functional Requirements:")
for req in functional_requirements:
    print(req)
print("Non-Functional Requirements:")
for req in non_functional_requirements:
    print(req)

# Пример использования функции
file_path = 'requirements.txt' # Замените 'requirements.txt' на путь к вашему файлу с
# требованиями
extract_requirements(file_path)

```

11.

```

import matplotlib.pyplot as plt

# Функция для создания блока с функцией
def create_function_block(x, y, text):
    plt.text(x, y, text, ha='center', va='center', bbox=dict(facecolor='white', edgecolor='black',
    boxstyle='round,pad=0.5'))

# Функция для создания блока с процессом
def create_process_block(x, y, text):
    plt.text(x, y, text, ha='center', va='center', bbox=dict(facecolor='white', edgecolor='black',
    boxstyle='round,pad=0.5'))

# Функция для создания связи между блоками
def create_arrow(start, end):
    plt.arrow(start[0], start[1], end[0]-start[0], end[1]-start[1], head_width=0.1, head_length=0.1,
    fc='black', ec='black' )

# Создание блоков с функциями
create_function_block(0, 0, "User Input")
create_function_block(0, 2, "Data Processing")
create_function_block(0, 4, "Data Output")

# Создание блоков с процессами
create_process_block(4, 1, "Data Collection")
create_process_block(4, 3, "Data Analysis")
create_process_block(4, 5, "Report Generation")

# Создание связей между блоками
create_arrow([0, 0.5], [3, 1])
create_arrow([0, 1.5], [3, 1])
create_arrow([0, 2.5], [3, 3])
create_arrow([0, 3.5], [3, 3])
create_arrow([0, 4.5], [3, 5])

```

```
# Отображение диаграммы
plt.axis('off')
plt.show()
```

12.

```
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords

# текст требований к интерфейсу
requirements = "Требуется добавить кнопку 'Отправить' на форму ввода. Меню должно
содержать вкладки 'Главная' и 'Настройки'."

# токенизация текста
tokens = word_tokenize(requirements)

# удаление стоп-слов
stop_words = set(stopwords.words('russian'))
filtered_tokens = [word for word in tokens if word.lower() not in stop_words]

# ключевые слова для анализа
interface_elements = ["кнопка", "вкладка", "меню"]

# счетчики для упоминаний каждого элемента
counts = {element: 0 for element in interface_elements}

# поиск упоминаний ключевых слов
for token in filtered_tokens:
    for element in interface_elements:
        if element in token.lower():
            counts[element] += 1

# вывод отчета
for element, count in counts.items():
    print(f"Упоминаний {element}: {count}")
```

13.

```
import radon.metrics

def analyze_code(file_path):
    with open(file_path, 'r') as file:
        code = file.read()
    # Получаем метрики модуля
    module_metrics = radon.metrics.analyze(code)
    print(f'Количество строк: {module_metrics.loc}')
    print(f'Сложность модуля: {module_metrics.complexity}')
```

```

# Получаем метрики функций
functions = radon.metrics.Functions.from_code(code)
for function in functions:
    print(f'Функция {function.name}:')
    print(f'  Длина функции: {function.length}')
    print(f'  Уровень вложенности: {function.volume}')

# Пример использования программы
file_path = 'path_to_your_file.py'
analyze_code(file_path)

```

14.

```

import difflib

def compare_versions(file1, file2):
    with open(file1, 'r') as f1:
        version1 = f1.readlines()
    with open(file2, 'r') as f2:
        version2 = f2.readlines()

    differ = difflib.Differ()

    diff = list(differ.compare(version1, version2))

    added = [line[2:] for line in diff if line.startswith('+ ')]
    removed = [line[2:] for line in diff if line.startswith('- ')]
    changed = [line[2:] for line in diff if line.startswith('? ')]

    report = {
        'added': added,
        'removed': removed,
        'changed': changed
    }

    return report

report = compare_versions('version1.py', 'version2.py')
print(report)

```

15.

```

import unittest
from pylint.lint import Run
import pycodestyle
import pydocstyle

class CodeQualityTest(unittest.TestCase):

```

```

def test_variables(self):
    # Проверяем правильность использования переменных
    pass # Здесь можно добавить проверки на правильное использование переменных

def test_coding_standards(self):
    # Проверяем соответствие стандартам кодирования
    style = pycodestyle.StyleGuide()
    result = style.check_files(['your_code.py'])
    self.assertEqual(result.total_errors, 0)

def test_documentation(self):
    # Проверяем наличие документации и комментариев
    result = pydocstyle.check(['your_code.py'])
    self.assertEqual(result.total_errors, 0)

if __name__ == '__main__':
    unittest.main()

```

16.

```

import requests

url = "http://api.example.com"
response = requests.get(url)

if response.status_code == 200:
    data = response.json()
    print("Результат запроса:")
    for key, value in data.items():
        print(f"{key}: {value}")
else:
    print(f"Ошибка запроса: {response.status_code}")

```

17.

```

import json

# Чтение данных из JSON-файла
with open('input.json', 'r') as file:
    data = json.load(file)

# Обработка данных
processed_data = {} # Здесь выполняется обработка данных (можно добавить любую
необходимую логику)

# Сохранение обработанных данных в новом формате
with open('output.txt', 'w') as file:
    file.write(str(processed_data)) # Здесь происходит сохранение в новый файл в другом
формате (например, текстовом)

```

```
print("Данные успешно обработаны и сохранены в новом файле.")
```

18.

```
from flask import Flask, request, jsonify
```

```
app = Flask(__name__)
```

```
# Пример данных
```

```
data = {  
    "id": 1,  
    "name": "John",  
    "age": 30  
}
```

```
# Роут для получения данных
```

```
@app.route('/api/data', methods=['GET'])
```

```
def get_data():
```

```
    return jsonify(data)
```

```
# Роут для отправки данных
```

```
@app.route('/api/data', methods=['POST'])
```

```
def update_data():
```

```
    new_data = request.get_json()
```

```
    data.update(new_data)
```

```
    return jsonify(data)
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

19.

```
import sqlite3
```

```
# Подключение к базе данных
```

```
conn = sqlite3.connect('example.db')
```

```
cursor = conn.cursor()
```

```
# Создание таблицы
```

```
cursor.execute("CREATE TABLE IF NOT EXISTS users  
                (id INTEGER PRIMARY KEY, name TEXT, age INTEGER)")
```

```
# Создание записи в базе данных (Create)
```

```
def create_user(name, age):
```

```
    cursor.execute("INSERT INTO users (name, age) VALUES (?, ?)", (name, age))
```

```
    conn.commit()
```

```
# Получение всех записей из базы данных (Read)
```

```

def get_all_users():
    cursor.execute('SELECT * FROM users')
    return cursor.fetchall()

# Обновление записи в базе данных (Update)
def update_user(id, name, age):
    cursor.execute('UPDATE users SET name = ?, age = ? WHERE id = ?', (name, age, id))
    conn.commit()

# Удаление записи из базы данных (Delete)
def delete_user(id):
    cursor.execute('DELETE FROM users WHERE id = ?', (id,))
    conn.commit()

# Пример использования функций для создания, чтения, обновления и удаления данных
create_user('Alice', 25)
create_user('Bob', 30)
print(get_all_users())

update_user(1, 'Alice Smith', 26)
print(get_all_users())

delete_user(2)
print(get_all_users())

# Закрытие соединения
conn.close()

```

20.

```

import serial

def send_command(serial_port, command):
    serial_port.write(command.encode())

def read_data(serial_port):
    data = serial_port.readline().decode()
    return data

def main():
    port_name = '/dev/ttyUSB0' # замените на нужное значение
    baudrate = 9600 # замените на нужное значение
    timeout = 1 # замените на нужное значение

    serial_port = serial.Serial(port_name, baudrate, timeout=timeout)

    # Отправляем команду устройству

```



```
send_command(serial_port, 'AT\r\n')

# Получаем данные от устройства
data = read_data(serial_port)
print('Received data:', data)

serial_port.close()

if __name__ == '__main__':
    main()
```

21.

```
a = 2
b = 5

import matplotlib.pyplot as plt

x_values = range(10)
y_values = [a*x + b for x in x_values]

plt.plot(x_values, y_values)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Deterministic Process: y = ax + b')
plt.show()
```

22.

```
v = 3 # постоянная скорость
t_values = [0, 1, 2, 3, 4] # значения времени

for t in t_values:
    x = v * t # вычисляем координату точки
    print(f'При t = {t}, x = {x}')
```

23.

```
import matplotlib.pyplot as plt

# Определение функции y = f(x)
def f(x):
    return x**2 - 2*x + 2

# Задание интервала значений x
x_values = range(-10, 11)

# Вычисление значений функции y = f(x)
y_values = [f(x) for x in x_values]

# Создание графика
```

```
plt.plot(x_values, y_values)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Deterministic Function:  $y = f(x) = x^2 - 2x + 2$ ')
plt.grid(True)
plt.show()
```

24.

```
import numpy as np
from scipy.optimize import curve_fit

# Заданные данные временного ряда
x = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
y = np.array([3, 5, 7, 9, 11, 13, 15, 17, 19, 21])

# Функция для подгонки данных
def func(x, a, b):
    return a * x + b

# Подгонка функции к данным
popt, _ = curve_fit(func, x, y)

a = popt[0]
b = popt[1]

# Прогнозирование будущих значений
future_values = []
for i in range(11, 16):
    future_values.append(func(i, a, b))

print("Прогноз будущих значений:", future_values)
```

25.

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# Определение функции, описывающей систему
def model(y, t):
    k = 0.3
    dydt = -k * y
    return dydt

# Начальное условие
y0 = 5

# Временные точки для решения дифференциального уравнения
```

```
t = np.linspace(0, 20, 100)

# Решение дифференциального уравнения
y = odeint(model, y0, t)

# Построение графика изменения состояния системы
plt.plot(t, y)
plt.xlabel('Time')
plt.ylabel('y(t)')
plt.title('Dynamic System Simulation')
plt.grid()
plt.show()
```

26.

```
import random

number = random.randint(1, 10)

while True:
    guess = int(input("Угадайте число от 1 до 10: "))

    if guess == number:
        print("Поздравляю, вы угадали число!")
        break
    elif guess < number:
        print("Ваше число слишком маленькое.")
    else:
        print("Ваше число слишком большое.")
```

27.

```
import random
import string

def generate_password(length, characters_type):
    if characters_type == 'digits':
        characters = string.digits
    elif characters_type == 'letters':
        characters = string.ascii_letters
    else:
        characters = string.ascii_letters + string.digits

    password = ''.join(random.choice(characters) for i in range(length))
    return password

length = int(input("Введите длину пароля: "))
characters_type = input("Выберите тип символов (digits - только цифры, letters - только буквы, any - буквы и цифры): ")
```

```
password = generate_password(length, characters_type)
print("Сгенерированный пароль: ", password)
```

28.

```
import random

def roll_dice():
    return random.randint(1, 6)

num_rolls = int(input("Введите количество бросков: "))
show_results = input("Хотите видеть результат каждого броска? (да/нет): ")

total_sum = 0

if show_results.lower() == 'да':
    for i in range(num_rolls):
        result = roll_dice()
        print(f"Бросок {i+1}: {result}")
        total_sum += result
else:
    for i in range(num_rolls):
        total_sum += roll_dice()

print(f"Общий результат: {total_sum}")
```

29.

```
import random

def get_user_choice():
    user_choice = input("Выберите камень (r), ножницы (s) или бумагу (p): ")
    return user_choice

def get_computer_choice():
    choices = ['r', 's', 'p']
    computer_choice = random.choice(choices)
    return computer_choice

def determine_winner(user_choice, computer_choice):
    if user_choice == computer_choice:
        return "Ничья"
    elif (user_choice == 'r' and computer_choice == 's') or (user_choice == 's' and
computer_choice == 'p') or (user_choice == 'p' and computer_choice == 'r'):
        return "Вы победили!"
    else:
        return "Компьютер победил!"
```

```

def main():
    print("Добро пожаловать в игру 'Камень, ножницы, бумага!!!")
    user_choice = get_user_choice()
    computer_choice = get_computer_choice()
    print("Выбор компьютера:", computer_choice)
    print(determine_winner(user_choice, computer_choice))

if __name__ == "__main__":
    main()

```

30.

```

import random

def check_multiples(start, end, multiple_of):
    for _ in range(10): # Генерируем 10 случайных чисел
        number = random.randint(start, end) # Генерируем случайное число в заданном диапазоне
        if number % multiple_of == 0:
            print(f"{number} is a multiple of {multiple_of}")
        else:
            print(f"{number} is not a multiple of {multiple_of}")

# Пример использования
start_range = int(input("Enter the start of the range: "))
end_range = int(input("Enter the end of the range: "))
multiple = int(input("Enter the number to check for multiples: "))

check_multiples(start_range, end_range, multiple)

```

Оценка запланированных результатов по учебной дисциплине (предмету)

Результаты обучения (элементы)	Показатели оценки результата
<p>ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;</p> <p>ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;</p> <p>ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;</p>	<p>участия в выработке требований к программному обеспечению;</p> <p>участия в проектировании программного обеспечения с использованием специализированных программных пакетов;</p> <p>разработки тестовых наборов (пакетов) для программного модуля;</p> <p>разработки тестовых сценариев программного средства;</p> <p>инспектирования разработанных программных модулей на предмет соответствия стандартам кодирования.</p> <p>использовать выбранную систему контроля версий;</p>

<p>ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;</p> <p>ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;</p> <p>ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;</p> <p>ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;</p> <p>ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;</p> <p>ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.</p>	<p>использовать методы для получения кода с заданной функциональностью и степенью качества;</p> <p>выполнять тестирование интеграции;</p> <p>организовывать постобработку данных;</p> <p>использовать приемы работы в системах контроля версий;</p> <p>выполнять отладку, используя методы и инструменты условной компиляции;</p> <p>выявлять ошибки в системных компонентах на основе спецификаций модели процесса разработки программного обеспечения;</p> <p>основные принципы процесса разработки программного обеспечения;</p> <p>основные подходы к интегрированию программных модулей;</p> <p>основы верификации и аттестации программного обеспечения;</p> <p>основные методы отладки;</p> <p>методы и схемы обработки исключительных ситуаций;</p> <p>приемы работы с инструментальными средствами тестирования и отладки;</p> <p>стандарты качества программной документации;</p> <p>основы организации инспектирования и верификации;</p> <p>встроенные и основные</p>
<p>ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p> <p>ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.</p> <p>ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.</p> <p>ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p> <p>ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>	<p>специализированные инструменты анализа качества программных продуктов</p>

Образец экзаменационного билета
МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

Утверждаю
Заместитель директора
Борзенкова Е.К.
(Ф.И.О.)

(подпись)

**Экзаменационный билет №1
по ПМ.02 Осуществление
интеграции программных
модулей**

Рассмотрено на заседании
цикловой методической
комиссии
Председатель Гаврикова Е.
Ю.
(Ф.И.О.)

«___» _____ 2024 г.

Группа (ы) _____

Специальность:

09.02.07 Информационные
системы и программирование

(подпись)

«___» _____ 2024 г.

1. Что такое стандартизация требований к программному обеспечению?
2. Напишите программу, которая проверяет, соответствует ли имя файла определенному шаблону. Например, имя файла должно начинаться с буквы и заканчиваться цифрой.

Критерии оценки ответов обучающихся

Отметка 5 «отлично» - продемонстрирован высокий уровень знаний по теоретическому вопросу, тематика вопроса полностью раскрыта. Практико-ориентированное задание выполнено верно.

Отметка 4 «хорошо» - продемонстрировано понимание и знание основного содержания теоретического вопроса билета, однако допущены недочеты в определениях терминов и понятий. Практико-ориентированное задание выполнено с замечаниями.

Отметка 3 «удовлетворительно» - продемонстрировано слабое владение основным содержанием по теоретическому вопросу билета, допущены неточности в определениях терминов и понятий. Практико-ориентированное задание выполнено с ошибками.

Отметка 2 «неудовлетворительно» - не продемонстрировано владение знаниями и умениями, тема теоретического вопроса билета не раскрыта. Практико-ориентированное задание не выполнено.

4.Комплект билетов – 30 шт.

5. Экзаменационная ведомость.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

МДК 04.01 Внедрение и поддержка компьютерных систем

Подготовки специалистов среднего звена по специальности

код специальности 09.02.07 Информационные системы и программирование

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности СПО 09.02.07 «Информационные системы и программирование» базового уровня программы МДК 04.01 «Внедрение и поддержка компьютерных систем».

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчик: Бахаев Денис Викторович, преподаватель

ОДОБРЕН

цикловой методической комиссией

Протокол № 1

от «06» сентября 2022 г.

Председатель Гаврикова Е.Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения МДК, подлежащие проверке
3. Оценка освоения МДК
 - 3.1. Контроль и оценка освоения МДК
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет материалов
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения МДК «Внедрение и поддержка компьютерных систем» обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 «Информационные системы и программирование», базовый уровень следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями.

Обучающийся должен уметь:

У 1.	Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем.
У 2.	Использовать методы защиты программного обеспечения компьютерных систем.
У 3.	Проводить инсталляцию программного обеспечения компьютерных систем.
У 4.	Производить настройку отдельных компонентов программного обеспечения компьютерных систем.
У 5.	Анализировать риски и характеристики качества программного обеспечения.

Обучающийся должен знать:

З 1	Основные методы и средства эффективного анализа функционирования программного обеспечения.
З 2	Основные виды работ на этапе сопровождения программного обеспечения.
З 3	Основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения.
З 4	Средства защиты программного обеспечения в компьютерных системах.

Обучающийся должен иметь практический опыт: в настройке отдельных компонентов программного обеспечения компьютерных систем; в выполнении отдельных видов работ на этапе поддержки программного обеспечения компьютерной системы.

Формируемые ОК:

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 4. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языках.

Формируемые ПК:

Код	Наименование видов деятельности и профессиональных компетенций
ПК 4.1.	- осуществлять инсталляцию, настройку и обслуживание программного обеспечения компьютерных систем
ПК 4.2	- осуществлять измерения эксплуатационных характеристик программного обеспечения компьютерных систем
ПК 4.3	- выполнять работы по модификации отдельных компонент программного обеспечения в соответствии с потребностями заказчика
ПК 4.4	- обеспечивать защиту программного обеспечения компьютерных систем программными средствами

Личностные результаты с учетом особенностей учебной дисциплины:

Код ЛР	Личностные результаты реализации программы (дескрипторы)
ЛР 1	- осознающий себя гражданином и защитником великой страны
ЛР 2	- проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	- демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих
ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личного и профессионального, конструктивного «цифрового следа»
ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	- проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	- уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 10	- принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий

	либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР 17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
ЛР 19	- профессиональная идентичность и ответственность
ЛР 20	- самооценка и рефлексия результатов своей деятельности и развития

Формами промежуточной аттестации по МДК являются: проверочная работа (3 семестр), экзамен (4 семестр).

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по МДК осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих и профессиональных компетенций:

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У 1. Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 9, ПК 4.2	- настройка предложенного программного обеспечения, обоснование варианта конфигурации.
У 2. Использовать методы защиты программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- подбор и настройка оборудования и программного обеспечения, направленного на защиту компьютерных систем; - использование корректных систем защиты для различных видов аппаратного и программного обеспечения.
У 3. Проводить инсталляцию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 4.1	- организация доступа к программному обеспечению различным категориям пользователей; - обеспечивать совместимость компонент с ранее установленными программными продуктами.

У 4. Производить настройку отдельных компонентов программного обеспечения компьютерных систем ОК 01, ОК 02, ОК 09, ПК 4.3	- настройка программного обеспечения отраслевой направленности для обеспечения его бесперебойной работы; - предложение вариантов модификации программного обеспечения.
У 5. Анализировать риски и характеристики качества программного обеспечения ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- анализ условий эксплуатации программного обеспечения; проверка настроек конфигурации; - анализ функционирования с помощью инструментальных средств; определение причин несоответствия выполняемых функций требованиям заказчика.
Знать:	
З 1: Основные методы и средства эффективного анализа; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.2	- перечисление основных этапов жизненного цикла программного обеспечения; - определение основных методов и средств эффективного анализа программного и аппаратного обеспечения.
З 2: Основные виды работ на этапе сопровождения программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- демонстрирует знание основных видов работ на этапах внедрения и сопровождения программного обеспечения.
З 3: Основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- формулирует основные принципы контроля конфигурации программного обеспечения и принципы поддержки целостности конфигурации программного обеспечения.
З4: Средства защиты программного обеспечения в компьютерных системах; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.4	- определяет основные средства защиты программного обеспечения в компьютерных системах и способы их настройки и применения.

3.ОЦЕНКА ОСВОЕНИЯ МДК

3.1. Контроль и оценка освоения МДК по темам (разделам)

Элемент МДК	Формы и методы контроля			
	Текущий контроль		Промежуточная аттестация	
	Осваиваемые результаты	Метод контроля	Проверяемые результаты	Форма контроля
Тема 1. Основные методы внедрения и анализа функционирования программного обеспечения	ОК 1, ОК 2, ОК 5, ОК 9, У 1, У 2, У 3, У 5, З 1, З 2	Устный опрос Самостоятельная работа Практическая работа Тестовое задание	ОК 1, ОК 2, ОК 04, ОК 5, ОК 9, У 1, У 2, З 1, З 2 ПК 4.1 - 4.2 ОК 1, ОК 2, ОК 5, ОК 9,	3-й семестр – контрольная работа, 4 семестр – экзамен

Тема 2. Загрузка и установка программного обеспечения	ОК 1, ОК 2, ОК 5, ОК 9, У 1, У 2, 3 3, 3 4	Устный опрос Практическая работа Тестовое задание	У 1 – 5, 31 - 34, ПК 4.1 - 4.4	
---	--	---	--------------------------------	--

3.1.1. Методы и критерии оценки

1. Устный опрос. Критерии оценивания.

Оценка 5 «отлично» - ответил на вопросы в объеме лекционного и дополнительного материала, дал полные грамотные ответы на все дополнительные вопросы.

Оценка 4 «хорошо» - грамотно изложил ответы на вопросы, но содержание и формулировки имеют отдельные неточности (допускается нечеткая формулировка определений), в полной мере ответил на заданные дополнительные вопросы.

Оценка 3 «удовлетворительно» - ответил на часть вопросов в объеме лекционного материала и ответил на часть дополнительных вопросов.

Оценка 2 «неудовлетворительно» - допустил ошибки в определении базовых понятий, исказил их смысл, не ответил на дополнительные вопросы.

2. Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

3. Практическая работа. Критерии оценивания.

Оценка «5» - работа выполнена в полном объеме, с соблюдением алгоритма выполнения: последовательности проведения измерений, заполнения таблиц, графиков и др.; правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; получены результаты в соответствии с поставленной целью, отчет составлен по всем требованиям.

Оценка «4» - выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета, незначительные ошибки в составлении отчета.

Оценка «3» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью, значительные ошибки в оформлении отчета.

Оценка «2» - работа выполнена не полностью и объем выполненной части работы не позволяет получить никаких результатов в соответствии с поставленной целью, отчет не сформирован или сформирован неполно.

4. Самостоятельная работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме; учтены все требования к данной работе; самостоятельно поставлены цели и задачи работы, соответствующие заданной теме/проблеме; получены результаты в соответствии с поставленной целью; работа оформлена аккуратно и грамотно.

Оценка 4 «хорошо» - выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью, и объем выполненной части работы позволяет получить недостаточно результатов в соответствии с поставленной целью.

4. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Контрольная работа (3 семестр)

1. Форма проведения: тест

2. Условия выполнения

1. Инструкция для обучающихся.

2. Время выполнения задания: 45 минут.

3. Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.

4. Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда виртуализации VMWare Work station версии 12 и выше; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

5. Информационные источники: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».

6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения проверочной работы

3.1. Перечень тем, контролируемых в ходе промежуточной аттестации.

Тема 1. Основные методы внедрения и анализа функционирования программного обеспечения

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У 1. Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 9, ПК 4.2	- настройка предложенного программного обеспечения, обоснование варианта конфигурации.
У 2. Использовать методы защиты программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- подбор и настройка оборудования и программного обеспечения, направленного на защиту компьютерных систем; - использование корректных систем защиты для различных видов аппаратного и программного обеспечения.
У 3. Проводить инсталляцию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 4.1	- организация доступа к программному обеспечению различным категориям пользователей;

	- обеспечивать совместимость компонент с ранее установленными программными продуктами.
У 4. Производить настройку отдельных компонентов программного обеспечения компьютерных систем ОК 01, ОК 02, ОК 09, ПК 4.3	- настройка программного обеспечения отраслевой направленности для обеспечения его бесперебойной работы; - предложение вариантов модификации программного обеспечения.
У 5. Анализировать риски и характеристики качества программного обеспечения ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- анализ условий эксплуатации программного обеспечения; проверка настроек конфигурации; - анализ функционирования с помощью инструментальных средств; определение причин несоответствия выполняемых функций требованиям заказчика.
Знать:	
З 1: Основные методы и средства эффективного анализа; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.2	- перечисление основных этапов жизненного цикла программного обеспечения; - определение основных методов и средств эффективного анализа программного и аппаратного обеспечения.
З 2: Основные виды работ на этапе сопровождения программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- демонстрирует знание основных видов работ на этапах внедрения и сопровождения программного обеспечения.
З 3: Основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- формулирует основные принципы контроля конфигурации программного обеспечения и принципы поддержки целостности конфигурации программного обеспечения.
З 4: Средства защиты программного обеспечения в компьютерных системах; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.4	- определяет основные средства защиты программного обеспечения в компьютерных системах и способы их настройки и применения.

Образец тестового задания

- Какие программы можно отнести к системному ПО:
 - утилиты
 - экономические программы
 - статистические программы
 - мультимедийные программы
- Этап, занимающий наибольшее время, в жизненном цикле программы:
 - сопровождение
 - проектирование
 - тестирование
 - программирование

- E) формулировка требований
3. Этап, занимающий наибольшее время, при разработке программы:
- A) тестирование
 - B) сопровождение
 - C) проектирование
 - D) программирование
 - E) формулировка требований
4. Первый этап в жизненном цикле программы:
- A) формулировка требований
 - B) анализ требований
 - C) проектирование
 - D) автономное тестирование
 - E) комплексное тестирование
5. Один из необязательных этапов жизненного цикла программы:
- A) оптимизация
 - B) проектирование
 - C) тестирование
 - D) программирование
 - E) анализ требований
6. Самый большой этап в жизненном цикле программы:
- A) эксплуатация
 - B) изучение предметной области
 - C) программирование
 - D) тестирование
 - E) корректировка ошибок
7. Какой этап выполняется раньше:
- A) отладка
 - B) оптимизация
 - C) программирование
 - D) тестирование
8. Что выполняется раньше:
- A) проектирование
 - B) программирование
 - C) отладка
 - D) тестирование
9. В стадии разработки программы не входит:
- A) автоматизация программирования
 - B) постановка задачи
 - C) составление спецификаций
 - D) эскизный проект
 - E) тестирование
10. Самый важный критерий качества программы:
- A) работоспособность
 - B) надежность
 - C) эффективность

D) быстродействие

E) простота эксплуатации

11. Способы оценки качества:

A) сравнение с аналогами

B) наличие документации

C) оптимизация программы

D) структурирование алгоритма

12. Наиболее важный критерий качества:

A) надежность

B) быстродействие

C) удобство в эксплуатации

D) удобный интерфейс

E) эффективность

13. Способы оценки надежности:

A) тестирование

B) сравнение с аналогами

C) трассировка

D) оптимизация

14. В каких единицах можно измерить надежность:

A) отказов/час

B) км/час

C) Кбайт/сек

D) операций/сек

15. В каких единицах можно измерить быстродействие:

A) отказов/час

B) км/час

C) Кбайт/сек

D) операций/сек

16. Что относится к этапу программирования:

A) написание кода программы

B) разработка интерфейса

C) работоспособность

D) анализ требований

17. На каком этапе производится выбор языка программирования:

A) проектирование

B) программирование

C) отладка

D) тестирование

18. Процедура поиска ошибки, когда известно, что она есть это:

A) отладка

B) тестирование

C) компоновка

D) транзакция

E) трансляция

19. На каком этапе создания программы могут появиться синтаксические ошибки:

- A) программирование
- B) проектирование
- C) анализ требований
- D) тестирование

20. Когда приступают к тестированию программы:

- A) когда программа уже закончена
- B) после постановки задачи
- C) на этапе программирования
- D) на этапе проектирования
- E) после составления спецификаций

21. Какие методы используются для внедрения программного обеспечения?

22. Что такое анализ требований и как он помогает в процессе внедрения программного обеспечения?

23. Какие методы используются для анализа функционирования программного обеспечения?

24. Какие методы используются для анализа производительности программного обеспечения?

25. Какие преимущества дает анализ кода программного обеспечения?

4.Эталоны ответов

1) A; 2) A; 3) A; 4) A; 5) A; 6) A; 7) C; 8) A; 9) A; 10) A; 11) A; 12) A; 13) A; 14) A; 15) D; 16) A; 17) A; 18) A; 19) A; 20) A.

21) Для внедрения программного обеспечения используются такие методы, как анализ требований, проектирование, разработка, тестирование, развертывание и поддержка.

22) Анализ требований - это процесс определения потребностей и ожиданий пользователей от программного обеспечения. Он помогает понять, какие функции и возможности должны быть реализованы в программе, чтобы удовлетворить потребности пользователей.

23) Для анализа функционирования программного обеспечения используются такие методы, как тестирование, мониторинг, аудит безопасности, анализ производительности и анализ кода.

24) Для анализа производительности программного обеспечения используются такие методы, как мониторинг, профилирование, бенчмаркинг и анализ времени выполнения.

25) Анализ кода программного обеспечения позволяет выявить ошибки и проблемы в его структуре и реализации. Он помогает улучшить качество кода, повысить его надежность и безопасность, а также упростить его поддержку и дальнейшее развитие.

Критерии оценки ответов

Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

5.Ведомость

2.Экзамен

1.Форма проведения: письменная – тест и практико-ориентированное задание

2.Условия выполнения

1.Инструкция для обучающихся.

2. Время выполнения задания: 45 минут.

3. Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.
4. Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда виртуализации VMWare Work station версии 12 и выше; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.
5. Информационные источники: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».
6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3. Пакет материалов для проведения проверочной работы

3.1. Перечень тем, контролируемых в ходе промежуточной аттестации.

Тема 1. Основные методы внедрения и анализа функционирования программного обеспечения

Тема 2. Загрузка и установка программного обеспечения

3.2. Образец тестового задания к экзамену:

1. Чему нужно уделять больше времени, чтобы получить хорошую программу:
 - А) тестированию
 - В) программированию
 - С) отладке
 - Д) проектированию
2. Основное назначение тестирования:
 - А) повышение надежности программы
 - В) обнаружение ошибок
 - С) повышение эффективности программы
 - Д) улучшение эксплуатационных характеристик
 - Е) приведение программы к структурированному виду
3. Назначение отладки:
 - А) поиск причин существующих ошибок
 - В) поиск возможных ошибок
 - С) составление спецификаций
 - Д) разработка алгоритма
4. Отладка программ это:
 - А) локализация и исправление ошибок
 - В) алгоритмизация программирования
 - С) компиляция и компоновка
5. Что выполняется раньше, отладка или тестирование:
 - А) отладка
 - В) тестирование
6. Нахождение наилучшего варианта из множества возможных:
 - А) оптимизация
 - В) тестирование
 - С) автоматизация
 - Д) отладка
 - Е) сопровождение
7. Что такое оптимизация программ:
 - А) улучшение работы существующей программы
 - В) создание удобного интерфейса пользователя

- C) разработка модульной конструкции программы
 - D) применение методов объектно-ориентированного программирования
8. Критерии оптимизации:
- A) время выполнения или размер требуемой памяти
 - B) размер программы и ее эффективность
 - C) независимость модулей
 - D) качество программы, ее надежность
9. Критерии оптимизации:
- A) эффективность использования ресурсов
 - B) структурирование алгоритма
 - C) структурирование программы
10. Оптимизация программы это:
- A) модификация
 - B) отладка
 - C) повышение сложности программы
 - D) уменьшение сложности программы
11. Результат оптимизации программы:
- A) эффективность
 - B) надежность
 - C) машино-независимость
 - D) мобильность
12. Какой этап проектирования может быть исключен:
- A) эскизный проект
 - B) технический проект
 - C) рабочий проект
13. Какие этапы проектирования можно объединять:
- A) технический и рабочий
 - B) эскизный и рабочий
 - C) технический и эскизный
14. Процесс преобразования постановки задачи в план алгоритмического или вычислительного решения это:
- A) проектирование
 - B) анализ требований
 - C) программирование
 - D) тестирование
15. Составление спецификаций это:
- A) формализация задачи
 - B) эскизный проект
 - C) поиск алгоритма
 - D) отладка
16. Этап разработки программы, на котором дается характеристика области применения программы:
- A) техническое задание
 - B) эскизный проект
 - C) технический проект

- D) внедрение
- E) рабочий проект

17. Укажите правильную последовательность создания программы:

- A) формулирование задачи, анализ требований, проектирование, программирование
- B) анализ требований, проектирование, программирование, тестирование, отладка
- C) анализ требований, программирование, проектирование, тестирование
- D) анализ требований, проектирование, программирование, модификация, трассировка
- E) формулирование задачи, анализ требований, программирование, проектирование,

отладка

18. Уточнение структуры входных и выходных данных, разработка алгоритмов, определение элементов интерфейса входят в:

- A) технический проект
- B) рабочий проект
- C) эскизный проект

19. Несуществующий метод проектирования:

- A) алгоритмическое
- B) нисходящее
- C) восходящее

20. Повышает ли качество программ оптимизация:

- A) Верно
- B) Неверно

3.4. Практико-ориентированные задания к экзамену

Вариант 1. Windows 7 HomeBasic 32 bit

Условие для выполнения практического задания

Установить операционную систему Windows 7, Windows 8 или Windows 8.1 на виртуальную машину VMWare Workstation. Установить обновления, при необходимости.

1. Настроить параметры языка и региона установки, настройки часового пояса.
2. Создать учетную запись UserX и назначить его Администратором. (где X– номер варианта)
3. Создать учетную запись UserXX и назначить ее стандартными правами (где XX– номер варианта, введенный дважды)
4. Выполните резервное копирование установленной операционной системы и создать установочный образ системы.
5. Выполнить настройку файла подкачки, 3078 или 4096 МБ, в зависимости от разрядности операционной системы.
6. Создать точки восстановления системы.
7. Настроить автоматическое упорядочивание значков Рабочего стола.
8. При помощи групповых политик запретить пользователю с обычными правами подключать внешние накопители, запретить доступ к Панели управления и параметрам компьютера, запретить доступ к свойствам подключений Локальной сети.
9. Подключить сетевую папку как сетевой диск, расположенную по адресу 192.168.2.9, назначить ему букву S.

4. Эталоны ответов

1) A; 2) A; 3) A; 4) A; 5) A; 6) A; 7) A; 8) A; 9) A; 10) A; 11) A; 12) A; 13) A; 14) A; 15) A; 16) A; 17) A; 18) A; 19) A; 20) A.

Эталоны ответов на практико-ориентированные задания

Эталоном ответа на практическое задание будет являться полностью работоспособная версия операционной системы, соответствующая заданной в варианте, с выполненными заданиями. Проверка каждой выполненной работы будет проводиться преподавателем индивидуально.

5.Комплект заданий.

6.Экзаменационная ведомость.

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У 1. Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 9, ПК 4.2	- настройка предложенного программного обеспечения, обоснование варианта конфигурации.
У 2. Использовать методы защиты программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- подбор и настройка оборудования и программного обеспечения, направленного на защиту компьютерных систем; - использование корректных систем защиты для различных видов аппаратного и программного обеспечения.
У 3. Проводить инсталляцию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 4.1	- организация доступа к программному обеспечению различным категориям пользователей; - обеспечивать совместимость компонент с ранее установленными программными продуктами.
У 4. Производить настройку отдельных компонентов программного обеспечения компьютерных систем ОК 01, ОК 02, ОК 09, ПК 4.3	- настройка программного обеспечения отраслевой направленности для обеспечения его бесперебойной работы; - предложение вариантов модификации программного обеспечения.
У 5. Анализировать риски и характеристики качества программного обеспечения ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- анализ условий эксплуатации программного обеспечения; проверка настроек конфигурации; - анализ функционирования с помощью инструментальных средств; определение причин несоответствия выполняемых функций требованиям заказчика.
Знать:	
З 1: Основные методы и средства эффективного анализа; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.2	- перечисление основных этапов жизненного цикла программного обеспечения; - определение основных методов и средств эффективного анализа программного и аппаратного обеспечения.

З 2: Основные виды работ на этапе сопровождения программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- демонстрирует знание основных видов работ на этапах внедрения и сопровождения программного обеспечения.
З 3: Основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- формулирует основные принципы контроля конфигурации программного обеспечения и принципы поддержки целостности конфигурации программного обеспечения.
З4: Средства защиты программного обеспечения в компьютерных системах; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.4	- определяет основные средства защиты программного обеспечения в компьютерных системах и способы их настройки и применения.

Критерии оценки ответов, обучающихся

Оценка 5 «отлично» - верно выполнено 70% теста и 50% практико-ориентированного задания без ошибок.

Оценка 4 «хорошо» - верно выполнено 50% теста и 40% практико-ориентированного задания, либо допущены ошибки при выполнении.

Оценка 3 «удовлетворительно» - верно выполнено 50% теста и не выполнено практико-ориентированное задание, либо выполнено с существенными ошибками.

Оценка 2 «неудовлетворительно» - верно выполнено менее 50% теста и не выполнено практико-ориентированное задание.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

МДК 04.02 Обеспечение качества функционирования компьютерных систем

Подготовки специалистов среднего звена по специальности

код специальности 09.02.07 Информационные системы и программирование

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности СПО 09.02.07 «Информационные системы и программирование» базового уровня программы «Обеспечение качества функционирования компьютерных систем».

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчик: Бахаев Денис Викторович, преподаватель

ОДОБРЕН

цикловой методической комиссией

Протокол № 1

от «3» сентября 2022 г.

Председатель Гаврикова Е.Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения МДК, подлежащие проверке
3. Оценка освоения МДК
 - 3.1. Контроль и оценка освоения МДК
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет материалов
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения МДК «Обеспечение качества функционирования компьютерных систем» обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 «Информационные системы и программирование», базовый уровень следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями.

Обучающийся должен уметь:

У 1.	Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем
У 2.	Использовать методы защиты программного обеспечения компьютерных систем
У 3.	Проводить инсталляцию программного обеспечения компьютерных систем
У 4.	Производить настройку отдельных компонентов программного обеспечения компьютерных систем
У 5.	Анализировать риски и характеристики качества программного обеспечения

Обучающийся должен знать:

З 1	Основные методы и средства эффективного анализа функционирования программного обеспечения
З 2	Основные виды работ на этапе сопровождения программного обеспечения
З 3	Основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения
З 4	Средства защиты программного обеспечения в компьютерных системах

Обучающийся должен иметь практический опыт: в настройке отдельных компонентов программного обеспечения компьютерных систем; в выполнении отдельных видов работ на этапе поддержки программного обеспечения компьютерной системы.

Формируемые ОК:

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 4. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языках.

Формируемые ПК:

ПК 4.1. Осуществлять инсталляцию, настройку и обслуживание программного обеспечения компьютерных систем

ПК 4.2 Осуществлять измерения эксплуатационных характеристик программного обеспечения компьютерных систем

ПК 4.3 Выполнять работы по модификации отдельных компонент программного обеспечения в соответствии с потребностями заказчика

ПК 4.4 Обеспечивать защиту программного обеспечения компьютерных систем программными средствами.

Личностные результаты МДК

Код ЛР	Личностные результаты реализации программы (дескрипторы)
ЛР 1	- осознающий себя гражданином и защитником великой страны
ЛР 2	- проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	- демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих
ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личного и профессионального, конструктивного «цифрового следа»
ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	- проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства
ЛР 9	- уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 10	- принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий

	либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР 17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
ЛР 19	- профессиональная идентичность и ответственность
ЛР 20	- самооценка и рефлексия результатов своей деятельности и развития

Формами промежуточной аттестации по МДК являются: контрольная работа (3 семестр), экзамен (4 семестр).

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по МДК осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих и профессиональных компетенций:

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У 1. Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 9, ПК 4.2	- настройка предложенного программного обеспечения, обоснование варианта конфигурации.
У 2. Использовать методы защиты программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- подбор и настройка оборудования и программного обеспечения, направленного на защиту компьютерных систем; - использование корректных систем защиты для различных видов аппаратного и программного обеспечения.
У 3. Проводить инсталляцию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 4.1	- организация доступа к программному обеспечению различным категориям пользователей; - обеспечивать совместимость компонент с ранее установленными программными продуктами.

У 4. Производить настройку отдельных компонентов программного обеспечения компьютерных систем ОК 01, ОК 02, ОК 09, ПК 4.3	- настройка программного обеспечения отраслевой направленности для обеспечения его бесперебойной работы; - предложение вариантов модификации программного обеспечения.
У 5. Анализировать риски и характеристики качества программного обеспечения ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- анализ условий эксплуатации программного обеспечения; проверка настроек конфигурации; - анализ функционирования с помощью инструментальных средств; определение причин несоответствия выполняемых функций требованиям заказчика.
Знать:	
З 1: Основные методы и средства эффективного анализа; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.2	- перечисление основных этапов жизненного цикла программного обеспечения; - определение основных методов и средств эффективного анализа программного и аппаратного обеспечения.
З 2: Основные виды работ на этапе сопровождения программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- демонстрирует знание основных видов работ на этапах внедрения и сопровождения программного обеспечения.
З 3: Основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- формулирует основные принципы контроля конфигурации программного обеспечения и принципы поддержки целостности конфигурации программного обеспечения.
З4: Средства защиты программного обеспечения в компьютерных системах; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.4	- определяет основные средства защиты программного обеспечения в компьютерных системах и способы их настройки и применения.

3.ОЦЕНКА ОСВОЕНИЯ МДК

3.1. Контроль и оценка освоения МДК по темам (разделам)

Элемент МДК	Формы и методы контроля			
	Текущий контроль		Промежуточная аттестация	
	Осваиваемые результаты	Метод контроля	Проверяемые результаты	Форма контроля
Тема 1. Основные методы обеспечения качества функционирования	ОК 1, ОК 2, ОК 04 ОК 5, ОК 9, У 1, У 2, У 3, У 5,3 1, 3 2	Устный опрос Самостоятельная работа Практическая работа Тестовое задание	ОК 1, ОК 2, ОК 04, ОК 5, ОК 9, У 1, У 2, 3 1, 3 2 ПК 4.1 - 4.2	3-й семестр – контрольная работа, 4 семестр – экзамен

Тема 2. Методы и средства защиты компьютерных систем	ОК 1, ОК 2, ОК 04, ОК 5, ОК 9, У 1, У 2, 3 3, 3 4	Устный опрос Практическая работа Тестовое задание	ОК 1, ОК 2, ОК 5, ОК 9, У 1 – 5, 31 – 34, ПК 4.1 – 4.4	
--	---	---	--	--

3.1.1. Методы и критерии оценки

1. Устный опрос. Критерии оценивания.

Оценка 5 «отлично» - ответил на вопросы в объеме лекционного и дополнительного материала, дал полные грамотные ответы на все дополнительные вопросы.

Оценка 4 «хорошо» - грамотно изложил ответы на вопросы, но содержание и формулировки имеют отдельные неточности (допускается нечеткая формулировка определений), в полной мере ответил на заданные дополнительные вопросы.

Оценка 3 «удовлетворительно» - ответил на часть вопросов в объеме лекционного материала и ответил на часть дополнительных вопросов.

Оценка 2 «неудовлетворительно» - допустил ошибки в определении базовых понятий, исказил их смысл, не ответил на дополнительные вопросы.

2. Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

3. Практическая работа. Критерии оценивания.

Оценка «5» - работа выполнена в полном объеме, с соблюдением алгоритма выполнения: последовательности проведения измерений, заполнения таблиц, графиков и др.; правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики, вычисления; получены результаты в соответствии с поставленной целью, отчет составлен по всем требованиям.

Оценка «4» - выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета, незначительные ошибки в составлении отчета.

Оценка «3» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью, значительные ошибки в оформлении отчета.

Оценка «2» - работа выполнена не полностью и объем выполненной части работы не позволяет получить никаких результатов в соответствии с поставленной целью, отчет не сформирован или сформирован неполно.

4. Самостоятельная работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме; учтены все требования к данной работе; самостоятельно поставлены цели и задачи работы, соответствующие заданной теме/проблеме; получены результаты в соответствии с поставленной целью; работа оформлена аккуратно и грамотно.

Оценка 4 «хорошо» - выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью, и объем выполненной части работы позволяет получить недостаточно результатов в соответствии с поставленной целью.

4.КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Контрольная работа (3 семестр)

1.Форма проведения: тест

2.Условия выполнения

1.Инструкция для обучающихся.

2. Время выполнения задания: 45 минут.

3. Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.

4. Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда виртуализации VMWare Work station версии 12 и выше; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

5. Информационные источники: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».

6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3.Пакет материалов для проведения проверочной работы

3.1. Перечень тем, контролируемых в ходе промежуточной аттестации.

Тема 1. Основные методы обеспечения качества функционирования.

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У 1. Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 9, ПК 4.2	- настройка предложенного программного обеспечения, обоснование варианта конфигурации.
У 2. Использовать методы защиты программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- подбор и настройка оборудования и программного обеспечения, направленного на защиту компьютерных систем; - использование корректных систем защиты для различных видов аппаратного и программного обеспечения.
У 3. Проводить инсталляцию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 4.1	- организация доступа к программному обеспечению различным категориям пользователей; - обеспечивать совместимость компонент с ранее установленными программными продуктами.
У 4. Производить настройку отдельных компонентов программного обеспечения компьютерных систем ОК 01, ОК 02, ОК 09, ПК 4.3	- настройка программного обеспечения отраслевой направленности для обеспечения его бесперебойной работы;

	- предложение вариантов модификации программного обеспечения.
У 5. Анализировать риски и характеристики качества программного обеспечения ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- анализ условий эксплуатации программного обеспечения; проверка настроек конфигурации; - анализ функционирования с помощью инструментальных средств; определение причин несоответствия выполняемых функций требованиям заказчика.
Знать:	
З 1: Основные методы и средства эффективного анализа; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.2	- перечисление основных этапов жизненного цикла программного обеспечения; - определение основных методов и средств эффективного анализа программного и аппаратного обеспечения.
З 2: Основные виды работ на этапе сопровождения программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- демонстрирует знание основных видов работ на этапах внедрения и сопровождения программного обеспечения.
З 3: Основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- формулирует основные принципы контроля конфигурации программного обеспечения и принципы поддержки целостности конфигурации программного обеспечения.
З4: Средства защиты программного обеспечения в компьютерных системах; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.4	- определяет основные средства защиты программного обеспечения в компьютерных системах и способы их настройки и применения.

Образец тестового задания

1. Под совокупностью программ, выполняемых вычислительной системой, понимается:

- А. программное обеспечение;
- Б. драйвер;
- В. операционная система;
- Г. утилиты.

2. Программы, непосредственно обеспечивающие выполнение необходимых пользователям работ называются:

- А. Системными;
- Б. Прикладными;
- В. Необходимыми;
- Г. Стандартными.

3. Классификация программного обеспечения делится на сколько видов?

- А. 2
- Б. 4
- В. 3

Г. 5

4. В системные программы вспомогательного назначения относятся:

- А. БИОС;
- Б. Утилиты;
- В. Драйвера;
- Г. Прикладное ПО.

5. Возможность выполнения одних и тех же программ на разных компьютерах с получением одинаковых результатов называется:

- А. Аппаратной совместимостью;
- Б. Разрядной совместимостью;
- В. Программной совместимостью;
- Г. Виртуальной совместимостью.

6. Безопасный режим, в котором компьютер запускается с минимальным количеством работающих программ и служб:

- А. Чистая загрузка;
- Б. Начальная загрузка;
- В. Полная загрузка;
- Г. Безопасная загрузка.

7. Какого вида тестирования не существует?

- А. Тестирование совместимости;
- Б. Конфигурационное тестирование;
- В. Регрессионное тестирование;
- Г. Виртуальное тестирование.

8. Не существующая среда обитания вирусов:

- А. Загрузочная;
- Б. Файловая;
- В. Игровая;
- Г. Нет верного ответа.

9. Вредоносная программа, которая собирает и отправляет информацию о пользователе:

- А. Червь;
- Б. Троянский конь;
- В. Программа-шпион;
- Г. Боты.

10. Какое расширение имеет исполняемый файл?

- А. dll
- Б. bin
- В. doc
- Г. exe

11. Процесс установки запускается при помощи файла:

- А. Setup.exe
- Б. Turbo.exe
- В. Startup.exe
- Г. Autorun.inf

12. Нарушение нормального функционирования отдельной программы, устройства или компьютера в целом:
- А. Сбой;
 - Б. Отказ;
 - В. Поломка;
 - Г. Ошибка.
13. Глобальные аппаратные и программные настройки системы хранятся в разделе реестра:
- А. HKEY_CURRENT_CONFIG;
 - Б. HKEY_LOCAL_MACHINE;
 - В. HKEY_USERS;
 - Г. HKEY_CLASSES_ROOT.
14. Программное обеспечение, являющееся частной собственностью авторов и правообладателей:
- А. Свободное;
 - Б. Проприетарное;
 - В. Стандартное;
 - Г. Авторское.
15. С помощью какого теста проверяется совместимость продукта с программным и аппаратным обеспечением?
- А. Регрессионное тестирование;
 - Б. Тестирование совместимости;
 - В. Инсталляционное тестирование;
 - Г. Конфигурационное тестирование.
16. Программное обеспечение, пользователи которого имеют права («свободы») на его неограниченную установку, запуск, свободное использование, изучение, распространение и изменение:
- А. Свободное;
 - Б. Проприетарное;
 - В. Стандартное;
 - Г. Авторское.
17. Мероприятия по приведению ПО в соответствие с законом и лицензионными соглашениями:
- А. Оптимизация ПО;
 - Б. Актуализация ПО;
 - В. Легализация ПО;
 - Г. Разработка ПО.
18. Иерархическая база данных, содержащая записи, определяющие параметры и настройки операционных систем Microsoft Windows:
- А. Системный реестр;
 - Б. СУБД;
 - В. Каталог;
 - Г. Корневой каталог.

19. Программы, облегчающие процесс создания новых программ для компьютера, называются:
- А. Стандартными;
 - Б. Системными;
 - В. Инструментальными;
 - Г. Прикладными.
20. Какое расширение имеют файлы библиотек?
- А. dll
 - Б. bin
 - В. docx
 - Г. exe
21. Какие методы обеспечения качества функционирования компьютерных систем вы знаете?
22. Что такое тестирование и как оно помогает обеспечить качество функционирования компьютерных систем?
23. Какие преимущества дает аудит безопасности компьютерной системы?
24. Как мониторинг производительности помогает обеспечить качество функционирования компьютерной системы?
25. Какие преимущества дает резервное копирование данных для обеспечения качества функционирования компьютерной системы?

4.Эталоны ответов

1) А; 2) Б; 3) В; 4) Б; 5) В; 6) Б; 7) Г; 8) В; 9) В; 10) Г; 11) А; 12) А; 13) Б; 14) Б; 15) Г; 16) А; 17) В; 18) А; 19) В; 20) А.

21) Существует несколько методов обеспечения качества функционирования компьютерных систем, включая тестирование, аудит безопасности, мониторинг производительности, резервное копирование данных и обновление программного обеспечения.

22) Тестирование - это процесс проверки работоспособности и соответствия требованиям компьютерной системы. Оно помогает выявить ошибки и проблемы в работе системы, а также оценить ее производительность и надежность.

23) Аудит безопасности компьютерной системы позволяет выявить уязвимости и потенциальные угрозы безопасности, а также оценить эффективность мер защиты. Это помогает предотвратить возможные атаки на систему и сохранить конфиденциальность данных.

24) Мониторинг производительности позволяет отслеживать работу компьютерной системы в режиме реального времени. Он помогает выявить проблемы с производительностью, такие как низкая скорость обработки данных или нехватка ресурсов, и принять меры для их устранения.

25) Резервное копирование данных позволяет сохранить информацию в случае сбоя или потери данных. Это помогает предотвратить потерю важных данных и обеспечивает возможность быстрого восстановления системы после сбоя.

Критерии оценки ответов

Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

2.Экзамен

1.Форма проведения: письменная – тест и практико-ориентированное задание

2.Условия выполнения

- 1.Инструкция для обучающихся.
2. Время выполнения задания: 45 минут.
3. Оборудование учебного кабинета: посадочные места для обучающихся и преподавателя.
4. Технические средства обучения: компьютер преподавателя – 1 шт.; компьютер студента – 15 шт.; на каждом компьютере ПО: Windows, Microsoft Office, среда виртуализации VMWare Work station версии 12 и выше; компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.
5. Информационные источники: <http://ditk.dgo4u.ru/> интернет-ресурс СДО MOODLE КГА ПОУ «ДИТК».
6. Требования охраны труда: Инструкция по охране труда при работе в компьютерном классе.

3.Пакет материалов для проведения проверочной работы

3.1. Перечень тем, контролируемых в ходе промежуточной аттестации.

Тема 1. Основные методы обеспечения качества функционирования.

Тема 2. Методы и средства защиты компьютерных систем.

3.2. Тестовые задания к экзамену по вариантам:

1. Назовите категории моделей качества:
 - a. теоретические модели
 - b. практические модели
 - c. модели управления данными
 - d. модели управления потоками
 - e. комбинированная модель
2. ... - свойство ПО сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных условиях применения
 - a. Уязвимость ПО
 - b. Надёжность ПО
 - c. Качество ПО
3. ... — это событие, при котором вся система или ее часть выходят из работоспособного состояния.
 - a. Сбой
 - b. Отказ
 - c. Прерывание
4. Назовите виды сбоев:
 - a. Сбой в системном программном обеспечении
 - b. Ресурсный сбой
 - c. Сбой в прикладном программном обеспечении
 - d. Сбой в результате действий пользователя
 - e. Аппаратный сбой
5. Основными методами предотвращения угроз надежности:
 - a. предотвращение ошибок проектирования за счет использования CASE-технологий, а также необходимо проводить систематическое тестирование и обязательную сертификацию ПО.

b. устранение негативных воздействий и дефектов, отражающихся на надежности функционирования программных средств

c. предотвращение ошибок в работе аппаратных и программных средствах

6. Назовите оперативные методы повышения надежности ПО:

a. Временная избыточность

b. Информационная избыточность

c. Программная избыточность

d. Аппаратная избыточность

e. Ресурсная избыточность

7. ... — это последствия внутренних дефектов ПО, которые следует квалифицировать как первичные причины обнаруженных аномалий результатов

a. Первичные ошибки

b. Вторичные ошибки

c. Критические ошибки

8. Умеренные ошибки — это

a. ошибки, на которые пользователь не обратит внимания при применении

программного продукта вследствие отсутствия их проявления и последствия которых обычно так и не обнаруживаются.

b. ошибки, которые влияют на конечного пользователя, но имеются слабые последствия или обходные пути, позволяющие сохранить достаточную функциональность программного продукта.

c. ошибки с высоким влиянием, останавливают выпуск версии программного продукта.

9. Риск — это

a. неопределенность, существующая в каждом проекте

b. процесс, связанный с идентификацией, анализом рисков и принятием решений,

которые включают максимизацию положительных и минимизацию отрицательных последствий наступления рисков событий

c. возможность понести потери во время выполнения проекта

10. Из какого количества этапов состоит Методика анализа рисков

a. 3

b. 4

c. 5

11. ... - небольшая программа, написанная в машинных кодах, способная внедряться в другие программы, хранящиеся на дисках запоминающих устройств компьютера и умеющая самовоспроизводиться

a. компьютерный вирус

b. вредоносная программа

c. зараженная программа

12. Назовите стадии вирусов:

a. активная и пассивная

b. временная и постоянная

c. проникновение и заражение

13. Какие вирусы классифицируются По среде обитания:

a. файловые, сетевые, загрузочные

b. резидентные и нерезидентные

- c. безвредные, опасные и очень опасные
 - d. вирусы-черви, троянские программы
14. Какие виды вирусов классифицируются по Особенностям алгоритма
- a. файловые вирусы и макровирусы
 - b. резидентные и нерезидентные
 - c. троянские программы и вирусы - черви
15. Назовите показатели качества антивирусной программы
- a. Надежность и удобство работы
 - b. Скорость работы
 - c. Производительность
 - d. Качество обнаружения вирусов
 - e. Актуальность обновления базы вирусов
16. ... — это программа находит зараженные вирусами файлы, но и «лечат» их, т. е. удаляют из файла тело программы-вируса, возвращая файлы в исходное состояние.
- a. Доктора
 - b. Детекторы
 - c. Ревизоры
 - d. Фаги
 - e. Фильтры
17. Принцип работы программ - фильтров
- a. запоминают исходное состояние программ, каталогов и системных областей диска тогда, когда компьютер не заражен вирусом, а затем периодически или по желанию пользователя сравнивают текущее состояние с исходным.
 - b. осуществляют последовательный просмотр файлов на предмет нахождения в них вирусных программ. Если сигнатура вируса была обнаружена, то производится процедура удаления вирусного кода из тела программы или документа.
 - c. осуществляют поиск характерного для конкретного вируса кода (сигнатуры) в оперативной памяти и в файлах, и при обнаружении выдают соответствующее сообщение.
 - d. находят зараженные вирусами файлы, но и «лечат» их, т. е. удаляют из файла тело программы-вируса, возвращая файлы в исходное состояние.
 - e. небольшие резидентные программы, предназначенные для обнаружения подозрительных действий при работе компьютера, характерных для вирусов.
18. ... - комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами.
- a. Монитор безопасности
 - b. Межсетевой экран
 - c. Антивирусная программа
19. Традиционный сетевой (или межсетевой) экран -
- a. программа (или неотъемлемая часть операционной системы) на шлюзе (сервере, передающем трафик между сетями) или аппаратное решение, контролирующее входящие и исходящие потоки данных между подключенными сетями.
 - b. программа, установленная на пользовательском компьютере и предназначенная для защиты от несанкционированного доступа только этого компьютера.

с. использование сетевого экрана сервером, для ограничения доступа к собственным ресурсам.

20. ... - понимают совокупность документированных решений, принимаемых руководством организации и направленных на защиту информации и ассоциированных с ней ресурсов.
- Политика безопасности
 - Механизм идентификации и аутентификации
 - Мониторинг безопасности
21. ... — это проверка принадлежности субъекту доступа предъявленного им идентификатора и подтверждение его подлинности
- Идентификация
 - Аутентификация
 - Протоколирование
 - Аудит
22. Протоколирование — это...
- сбор и накопление информации о событиях, происходящих в информационной системе. У каждого сервиса свой набор возможных событий
 - анализ накопленной информации, проводимый оперативно, в реальном времени или периодически
 - присвоение субъектам и объектам доступа личного идентификатора и сравнение его с заданным
23. Конечное множество используемых для кодирования информации знаков называется: ...
24. Наиболее надежным механизмом для защиты содержания сообщений является: ...
25. Совокупность свойств, обуславливающих пригодность информации удовлетворять определенные потребности в соответствии с ее назначением, называется: ...
26. Соотнесите основные понятия в области информационной безопасности:
- Атака
 - Уязвимость АС
 - Угроза безопасности АС
 - Защищенная система
- некоторое неудачное свойство системы, которое делает возможным возникновение и реализацию угрозы
 - система со средствами защиты, которые успешно и эффективно противостоят угрозам безопасности
 - возможные воздействия на АС, которые прямо или косвенно могут нанести ущерб ее безопасности
 - действие, предпринимаемое злоумышленником, которое заключается в поиске и использовании той или иной уязвимости системы

3.3. Практико-ориентированные задания к экзамену

Программное обеспечение	Вариант 1	Вариант 2	Вариант 3	Вариант 4	Вариант 5	Вариант 6
Операционная система	Windows 7 (32 bit)	Windows 7 (64 bit)	Windows 8 (32bit)	Windows 8 (64bit)	Windows 8.1 (32 bit)	Windows 8.1 (64 bit)

Офисные приложения	MS Word	MS Excel	MS PowerPoint	MS Publisher	MS Access	MS Word
Браузер	Mozilla Firefox	Google Chrome	Yandex Browser	Mozilla Firefox	Google Chrome	Yandex Browser
Архиваторы	Winrar	7zip	Pkzip	Winrar	7zip	Pkzip
Графические редакторы	Adobe Photoshop CS2	Corel DRAW X7	Adobe Photoshop CS2	Corel DRAW X7	Adobe Photoshop CS2	Corel DRAW X7
Программное обеспечение воспроизведения для звука	LouderIt	AIMP	Sonique	Windows Media Player	UltraPlayer	AIMP
Программное обеспечение для просмотра видеоизображений	VLC Media Player	Media Player Classic	MKV Player	VLC Media Player	Quick Time	MPlayer
Программное обеспечение для открытия файлов PDF	Adobe Acrobat Reader	Foxit PDF Reader	Wondershare PDF Reader	Adobe Acrobat Reader	Foxit PDF Reader	Wondershare PDF Reader
Антивирусное программное обеспечение	Eset Nod32	Kaspersky Anti-Virus	Dr. Web	Kaspersky Internet Security	Norton Antivirus	Avast! Internet Security
СУБД	PostgreSQL	PostgreSQL	PostgreSQL	PostgreSQL	PostgreSQL	PostgreSQL

Условие для выполнения практического задания

1. Установить и настроить программное обеспечение на предложенную версию Windows в виртуальной машине VMWare Workstation, установку производить не на системный диск. (Программное обеспечение необходимо найти в сети Интернет; Adobe Photoshop CS2, Corel DRAW X7 и пакет Microsoft Office находятся на диске D: хостовой системы в папке «Программные пакеты».) Все ПО должно быть настроено на работу «По умолчанию».
2. Решить проблемы совместимости, если таковые возникнут.
3. Определить объем дискового накопителя и оперативной памяти, необходимой для одновременного запуска всего предложенного ПО. Составить таблицу в MS Excel с указанием занимаемого дискового пространства и количества оперативной памяти на каждое ПО.
4. Продемонстрировать запуск системных приложений очистки диска и дефрагментации.
5. Настроить операционную систему таким образом, чтобы при старте запускался браузер на странице <https://itk-dg.ru/>.
6. Произвести быструю проверку ПК при помощи антивирусного пакета.

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
---	------------------------------

Уметь:	
У 1. Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 9, ПК 4.2	- настройка предложенного программного обеспечения, обоснование варианта конфигурации.
У 2. Использовать методы защиты программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- подбор и настройка оборудования и программного обеспечения, направленного на защиту компьютерных систем; - использование корректных систем защиты для различных видов аппаратного и программного обеспечения.
У 3. Проводить инсталляцию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 4.1	- организация доступа к программному обеспечению различным категориям пользователей; - обеспечивать совместимость компонент с ранее установленными программными продуктами.
У 4. Производить настройку отдельных компонентов программного обеспечения компьютерных систем ОК 01, ОК 02, ОК 09, ПК 4.3	- настройка программного обеспечения отраслевой направленности для обеспечения его бесперебойной работы; - предложение вариантов модификации программного обеспечения.
У 5. Анализировать риски и характеристики качества программного обеспечения ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- анализ условий эксплуатации программного обеспечения; проверка настроек конфигурации; - анализ функционирования с помощью инструментальных средств; определение причин несоответствия выполняемых функций требованиям заказчика.
Знать:	
З 1: Основные методы и средства эффективного анализа; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.2	- перечисление основных этапов жизненного цикла программного обеспечения; - определение основных методов и средств эффективного анализа программного и аппаратного обеспечения.
З 2: Основные виды работ на этапе сопровождения программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- демонстрирует знание основных видов работ на этапах внедрения и сопровождения программного обеспечения.
З 3: Основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- формулирует основные принципы контроля конфигурации программного обеспечения и принципы поддержки целостности конфигурации программного обеспечения.
З4: Средства защиты программного обеспечения в компьютерных системах; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.4	- определяет основные средства защиты программного обеспечения в компьютерных системах и способы их настройки и

	применения.
--	-------------

4. Эталоны ответов

Тестовая часть 1) А, С, Е; 2) В; 3) В; 4) А, С, D, Е; 5) А; 6) А, В, С; 7) В; 8) В; 9) С; 10) А; 11) А; 12) А; 13) А; 14) А; 15) А, В, D; 16) А, D; 17) Е; 18) В; 19) А; 20) А; 21) В; 22) А; 23) алфавитом (алфавит); 24) криптография; 25) качество (качество информации); 26) 1-d; 2-a; 3-c; 4-b.

Эталоны ответов на практико-ориентированные задания. Эталоном ответа на практическое задание будет являться полностью работоспособная версия операционной системы, соответствующая заданной в варианте, с установленным и настроенным программным обеспечением и выполненными заданиями. Проверка каждой выполненной работы будет проводиться преподавателем индивидуально.

Критерии оценки ответов, обучающихся

Оценка 5 «отлично» - верно выполнено 70% теста и 50% практико-ориентированного задания без ошибок.

Оценка 4 «хорошо» - верно выполнено 50% теста и 40% практико-ориентированного задания, либо допущены ошибки при выполнении.

Оценка 3 «удовлетворительно» - верно выполнено 50% теста и не выполнено практико-ориентированное задание, либо выполнено с существенными ошибками.

Оценка 2 «неудовлетворительно» - верно выполнено менее 50% теста и не выполнено практико-ориентированное задание.

4.Комплект заданий.

5.Экзаменационная ведомость.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем

Подготовки специалистов среднего звена по специальности

код специальности 09.02.07 Информационные системы и программирование

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности СПО 09.02.07 Информационные системы и программирование профессионального модуля ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем.

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчики:

Голубенко Максим Игоревич, преподаватель

Бахаев Денис Викторович преподаватель

ОДОБРЕН

цикловой методической комиссией

Протокол № 1

от «3» сентября 2022 г.

Председатель Гаврикова Е.Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения ПМ, подлежащие проверке
3. Оценка освоения ПМ
 - 3.1. Контроль и оценка освоения ПМ
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет материалов
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 Информационные системы и программирование следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями.

Обучающийся должен уметь:

У 1.	Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем.
У 2.	Использовать методы защиты программного обеспечения компьютерных систем.
У 3.	Проводить инсталляцию программного обеспечения компьютерных систем.
У 4.	Производить настройку отдельных компонентов программного обеспечения компьютерных систем.
У 5.	Анализировать риски и характеристики качества программного обеспечения.

Обучающийся должен знать:

З 1	Основные методы и средства эффективного анализа функционирования программного обеспечения.
З 2	Основные виды работ на этапе сопровождения программного обеспечения.
З 3	Основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения.
З 4	Средства защиты программного обеспечения в компьютерных системах.

Обучающийся должен иметь практический опыт: в настройке отдельных компонентов программного обеспечения компьютерных систем; в выполнении отдельных видов работ на этапе поддержки программного обеспечения компьютерной системы.

Формируемые ОК:

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 4. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языках.

Формируемые ПК:

Код	Наименование видов деятельности и профессиональных компетенций
ПК 4.1.	- осуществлять инсталляцию, настройку и обслуживание программного обеспечения компьютерных систем
ПК 4.2	- осуществлять измерения эксплуатационных характеристик программного обеспечения компьютерных систем
ПК 4.3	- выполнять работы по модификации отдельных компонент программного обеспечения в соответствии с потребностями заказчика
ПК 4.4	- обеспечивать защиту программного обеспечения компьютерных систем программными средствами

Личностные результаты с учетом особенностей учебной дисциплины:

Код ЛР	Личностные результаты реализации программы (дескрипторы)
ЛР 1	- осознающий себя гражданином и защитником великой страны
ЛР 2	- проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости. Экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующие и участвующие в деятельности общественных организаций. Готовый использовать свой личный и профессиональный потенциал для защиты национальных интересов России
ЛР 3	- демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих
ЛР 4	- принимающий семейные ценности своего народа, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания
ЛР 5	- занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 6	- принимающий цели и задачи научно-технологического, экономического, информационного развития России, готовый работать на их достижение. Стремящийся к формированию в сетевой среде личного и профессионального, конструктивного «цифрового следа»
ЛР 7	- готовый соответствовать ожиданиям работодателей: проектно мыслящий, эффективно взаимодействующий с членами команды и сотрудничающий с другими людьми, осознанно выполняющий профессиональные требования, ответственный, пунктуальный, дисциплинированный, трудолюбивый, критически мыслящий, нацеленный на достижение поставленных целей; демонстрирующий профессиональную жизнестойкость
ЛР 8	- проявляющий и демонстрирующий уважение к представителям различных этнокультурных, социальных, конфессиональных и иных групп. Сопричастный к сохранению, преумножению и трансляции культурных традиций и ценностей многонационального российского государства

ЛР 9	- уважающий этнокультурные, религиозные права человека, в том числе с особенностями развития; ценящий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 10	- принимающий активное участие в социально значимых мероприятиях, соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России; готовый оказать поддержку нуждающимся. Соблюдающий и пропагандирующий правила здорового и безопасного образа жизни, спорта; предупреждающий либо преодолевающий зависимости от алкоголя, табака, психоактивных веществ, азартных игр и т.д.
ЛР 11	- лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением
ЛР 12	- осознающий приоритетную ценность личности человека; уважающий собственную и чужую уникальность в различных ситуациях, во всех формах и видах деятельности
ЛР 13	- умение реализовывать личностные качества в производственном процессе
ЛР 14	- стрессоустойчивость, коммуникабельность
ЛР 15	- опыт научно-исследовательской деятельности
ЛР 16	- открытый к текущим и перспективным изменениям в мире труда, демонстрирующий навыки самообразования и саморазвития
ЛР 17	- инновационность мышления в реализации производственных задач
ЛР 18	- выполнение социальных норм и правил, внутреннего распорядка колледжа и предприятия
ЛР 19	- профессиональная идентичность и ответственность
ЛР 20	- самооценка и рефлексия результатов своей деятельности и развития

Промежуточной аттестации по ПМ является экзамен по билетам.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПМ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

Результатом освоения профессионального модуля является готовность обучающегося к выполнению вида профессиональной деятельности Сопровождение и обслуживание программного обеспечения компьютерных систем. Для подтверждения такой готовности обязательна констатация сформированности у обучающегося всех профессиональных компетенций, входящих в состав профессионального модуля. Общие компетенции формируются в процессе освоения ОПОП в целом, поэтому по результатам освоения профессионального модуля возможно оценивание положительной динамики их формирования.

Оценка запланированных результатов по ПМ

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
МДК 04.01 Внедрение и поддержка компьютерных систем	
Уметь:	
У 1. Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем;	- настройка предложенного программного обеспечения, обоснование варианта конфигурации.

ОК 01, ОК 02, ОК 9, ПК 4.2	
У 2. Использовать методы защиты программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- подбор и настройка оборудования и программного обеспечения, направленного на защиту компьютерных систем; - использование корректных систем защиты для различных видов аппаратного и программного обеспечения.
У 3. Проводить инсталляцию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 4.1	- организация доступа к программному обеспечению различным категориям пользователей; - обеспечивать совместимость компонент с ранее установленными программными продуктами.
У 4. Производить настройку отдельных компонентов программного обеспечения компьютерных систем ОК 01, ОК 02, ОК 09, ПК 4.3	- настройка программного обеспечения отраслевой направленности для обеспечения его бесперебойной работы; - предложение вариантов модификации программного обеспечения.
У 5. Анализировать риски и характеристики качества программного обеспечения ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- анализ условий эксплуатации программного обеспечения; проверка настроек конфигурации; - анализ функционирования с помощью инструментальных средств; определение причин несоответствия выполняемых функций требованиям заказчика.
Знать:	
З 1: Основные методы и средства эффективного анализа; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.2	- перечисление основных этапов жизненного цикла программного обеспечения; - определение основных методов и средств эффективного анализа программного и аппаратного обеспечения.
З 2: Основные виды работ на этапе сопровождения программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- демонстрирует знание основных видов работ на этапах внедрения и сопровождения программного обеспечения.
З 3: Основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- формулирует основные принципы контроля конфигурации программного обеспечения и принципы поддержки целостности конфигурации программного обеспечения.
З4: Средства защиты программного обеспечения в компьютерных системах; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.4	- определяет основные средства защиты программного обеспечения в компьютерных системах и способы их настройки и применения.

МДК 04.02 Обеспечение качества функционирования компьютерных систем	
Уметь:	
У 1. Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 9, ПК 4.2	- настройка предложенного программного обеспечения, обоснование варианта конфигурации.
У 2. Использовать методы защиты программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- подбор и настройка оборудования и программного обеспечения, направленного на защиту компьютерных систем; - использование корректных систем защиты для различных видов аппаратного и программного обеспечения.
У 3. Проводить инсталляцию программного обеспечения компьютерных систем; ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 4.1	- организация доступа к программному обеспечению различным категориям пользователей; - обеспечивать совместимость компонент с ранее установленными программными продуктами.
У 4. Производить настройку отдельных компонентов программного обеспечения компьютерных систем ОК 01, ОК 02, ОК 09, ПК 4.3	- настройка программного обеспечения отраслевой направленности для обеспечения его бесперебойной работы; - предложение вариантов модификации программного обеспечения.
У 5. Анализировать риски и характеристики качества программного обеспечения ОК 01, ОК 02, ОК 04, ОК 09, ПК 4.4	- анализ условий эксплуатации программного обеспечения; проверка настроек конфигурации; - анализ функционирования с помощью инструментальных средств; определение причин несоответствия выполняемых функций требованиям заказчика.
Знать:	
З 1: Основные методы и средства эффективного анализа; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.2	- перечисление основных этапов жизненного цикла программного обеспечения; - определение основных методов и средств эффективного анализа программного и аппаратного обеспечения.
З 2: Основные виды работ на этапе сопровождения программного обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	- демонстрирует знание основных видов работ на этапах внедрения и сопровождения программного обеспечения.
З 3: Основные принципы контроля конфигурации и поддержки целостности конфигурации программного	- формулирует основные принципы контроля конфигурации программного обеспечения и принципы поддержки

обеспечения; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.3	целостности конфигурации программного обеспечения.
34: Средства защиты программного обеспечения в компьютерных системах; ОК 01, ОК 02, ОК 05, ОК 09, ПК 4.4	- определяет основные средства защиты программного обеспечения в компьютерных системах и способы их настройки и применения.

3. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ЭКЗАМЕНА ПО ПМ

1. Экзамен по билетам

1. Форма проведения: экзамен.

2. Условия выполнения

Время выполнения задания: 90 мин.

Оборудование учебного кабинета: автоматизированные рабочие места в количестве 15 шт.

Технические средства обучения: автоматизированные рабочие места в количестве 15 шт.

Информационные источники: отсутствуют.

Требования охраны труда: в соответствии с требованиями СНиП.

3. Пакет материалов для проведения экзамена:

1.3.1 Перечень тем, выносимых на экзамен:

1. Основные методы внедрения и анализа функционирования программного обеспечения
2. Загрузка и установка программного обеспечения
3. Основные методы обеспечения качества функционирования.
4. Методы и средства защиты компьютерных систем.

3.2. Перечень вопросов, выносимых на экзамен

1. ГОСТ Р ИСО/МЭК 12207. О чем этот стандарт? Назовите основные группы процессов, согласно стандарту.
2. Жизненный цикл программного обеспечения. Описать основные этапы.
3. Понятие совместимости программного обеспечения. Аппаратная и программная совместимость.
4. Что такое «самотестирование системы»? Зачем нужны программы тестирования компьютера?
5. Что такое «межсетевой экран», в чем суть его работы? Какие межсетевые экраны бывают?
6. Виды тестирования программных продуктов в зависимости от целей. Кратко опишите каждый вид тестирования.
7. Что такое «виртуализация»? Чем платформы виртуализации помогают предприятиям или организациям?
8. Что такое «антивирусные программы»? Какие антивирусные программы бывают? Какие виды анализа используются в современных антивирусных программах?
9. Что такое BIOS? Зачем нужен и какие функции выполняет?
10. Что такое «резервирование»? Зачем необходимо выполнять резервное копирование данных?
11. Какие виды внедрения программного обеспечения вы знаете? Опишите кратко принцип каждого из них.

12. Чем отличаются аппаратно-программные платформы серверов и рабочих станций?
13. Какие бывают средства защиты от копирования?
14. Какие существуют методы и средства защиты компьютерных систем?
15. Какие виды вредоносных программ вы знаете? Назовите характерные признаки каждого вида.
16. Какие виды алгоритмических ошибок программного обеспечения вы знаете? Дайте краткую характеристику каждого вида.
17. Какие существуют оперативные методы повышения надежности программного обеспечения? Дайте краткую характеристику каждого метода.
18. На какие категории можно разделить риски при создании программного обеспечения?
19. Опишите последовательность действий при разработке и внедрении программного продукта.
20. Дестабилизирующие факторы и угрозы надежности программных средств. Дайте краткое описание.
21. Назовите основные критерии качества программного обеспечения и дайте краткую характеристику каждому.
22. Какие виды ошибок возникают на этапе разработки программного обеспечения информационных систем? Опишите кратко каждый вид.
23. Что такое информационная безопасность? Назовите основные составляющие информационной безопасности и дайте им определение.
24. Какие существуют типы алгоритмов шифрования? Дайте краткую характеристику каждому.
25. Дайте определение понятий «идентификация», «аутентификация», «авторизация». Составьте последовательность выполнения этих понятий при входе в защищенную информационную систему.
26. Какие виды программного обеспечения бывают? Дайте краткое определение каждого вида.
27. От чего зависит производительность ПК? Как можно улучшить производительность?
28. Что такое «безопасный режим» в ОС Windows? Чем он может помочь при диагностике системы?
29. Какие есть эксплуатационные характеристики качества программного обеспечения? Приведите примеры.
30. Чем сервер отличается от рабочей станции? Назовите виды серверов, дайте краткое описание каждому.

Эталоны ответов на вопросы:

1. Настоящий стандарт идентичен международному стандарту ИСО/МЭК 12207-2008* "Системная и программная инженерия. Процессы жизненного цикла программных средств". Это один из самых известных и распространенных процессно-ориентированных стандартов в области управления ИТ.

Методологическая основа ГОСТ Р ИСО/МЭК 12207 - разбиение процессов на группы, которых в стандарте вводится три:

А. Основные. Это процессы, непосредственно относящиеся к жизненному циклу информационной системы. Можно считать, что это производственные процессы организации. (Заказ, Поставка, Разработка, эксплуатация, сопровождение ПО)

В. Вспомогательные. Это процессы, предназначенные для поддержки основных процессов. Сами по себе эти процессы организации не нужны - только в связи с основными процессами, которые они обслуживают. Несколько процессов из этой группы связано с управлением качеством. (Документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместный анализ, аудит, решение проблем)

С. Организационные. Это общекорпоративные процессы, такие как "Обучение" или "Управление". Эти процессы существуют в организации независимо от того, как организовано производство и как устроены вспомогательные процессы. (Управление, создание инфраструктуры, усовершенствование, обучение)

2. У программного обеспечения, как у живого существа есть свой жизненный цикл. Жизненный цикл ПО – это стадии, которые проходит программный продукт от появления идеи до ее реализации в коде, имплементации в бизнес и последующей поддержки. Модели жизненного цикла во многом определяют и методологии разработки ПО.

Обычно к этапам жизненного цикла относят:

Анализ требований

Проектирование

Программирование

Тестирование и отладку

Эксплуатацию, сопровождение и поддержку

Существует некая вариативность в прохождении этапов ЖЦ во время разработки и внедрения продукта на рынок. Для каждого продукта это происходит по-своему, но чтобы процессом как-то управлять были сформулированы модели жизненного цикла ПО – упрощенное и обобщенное представление о том, как развивается продукт. В реальности жизнь продукта не соответствует модели.

Среди моделей жизненного цикла программного обеспечения наиболее известны следующие:

Каскадная модель (она же “водопадная” - waterfall)

Итерационные модели

Инкрементная модель

Спиральная модель

3. Совместимость — способность различных объектов — аппаратных или программных компонентов взаимодействовать друг с другом. По отношению к компьютерам можно выделить аппаратную и программную совместимость.

Аппаратная совместимость — способность одного устройства работать с узлами другого устройства. Включает в себя электромагнитную совместимость.

Программная совместимость — способность выполнения одинаковых программ с получением одних и тех же результатов. В случае представления программ в виде двоичного кода, говорят о двоичной совместимости.

Совместимостью программ называется способность программ к взаимодействию друг с другом, возможно, в рамках более крупного программного комплекса.

Для обеспечения совместимости аппаратного и программного обеспечения используется специальный вид ПО, который носит название «драйвер».

Драйвер – это компьютерное программное обеспечение, с помощью которого другое программное обеспечение (операционная система) получает доступ к аппаратному обеспечению некоторого устройства.

4. При запуске компьютера происходит тестирование основных его компонентов:

процессора, памяти, вспомогательных микросхем, приводов дисков, клавиатуры и видеоподсистемы. Самотестирование компьютерной системы происходит всегда при включении питания. Эта процедура находится в модуле ROM BIOS – ПЗУ системы организации ввода – вывода и называется POST (Power On Self-Test). Самотестирование проходит достаточно быстро, кроме проверки памяти и проверки работоспособности привода гибких дисков. Проверка важнейших устройств при включении питания позволяет локализовать неисправность уже на этапе начальной загрузки операционной системы.

Тестовые программы используются для идентификации конфигурации компьютера, его системных ресурсов, для его диагностики и оценки относительной производительности. Тестовые программы можно разделить на две группы: специализированные продукты, ориентированные на анализ какой-либо одной подсистемы компьютера. Например, широко используются программы для оценки производительности процессоров, жестких дисков; существуют программы, тестирующие работу оперативной памяти, видеоконтроллеров, звуковых карт и т. д.; универсальные программы, позволяющие выполнять комплексный анализ всех подсистем компьютера.

5. Межсетевой экран — это устройство обеспечения безопасности сети, которое осуществляет мониторинг входящего и исходящего сетевого трафика и на основании установленного набора правил безопасности принимает решение: пропустить или блокировать конкретный трафик.

Межсетевые экраны бывают двух видов: аппаратные и программные. Они выполняют одинаковые функции, но работают немного по-разному:

Программно-аппаратные комплексы (ПАК), или аппаратные МЭ — специальные устройства или компоненты роутеров, на которых установлено фильтрующее программное обеспечение. Все железо и ПО внутри этого устройства специализировано на фильтрации трафика и нескольких смежных задачах, дополнительные программы на аппаратный МЭ поставить нельзя. Это снижает уязвимость такого устройства к атакам и позволяет обеспечивать защищенность более высокого класса.

Программные МЭ — программное обеспечение на сервере, которое занимается фильтрацией трафика. По сути, это то же ПО, что установлено в аппаратном МЭ, но оно устанавливается на сам сервер.

Аппаратные МЭ обычно стоят на границе сети, например, там, где внутренняя сеть подключается к интернету. Программные стоят на узлах самой внутренней сети, то есть защищают непосредственно компьютеры и серверы.

6. Все виды тестирования программного обеспечения, в зависимости от преследуемых целей, можно условно разделить на следующие группы:

- Функциональные
- Нефункциональные
- Связанные с изменениями

Функциональные тесты базируются на функциях и особенностях, а также взаимодействии с другими системами, и могут быть представлены на всех уровнях тестирования: компонентном или модульном (Component/Unit testing), интеграционном (Integration testing), системном (System testing) и приемочном (Acceptance testing). Функциональные виды тестирования рассматривают внешнее поведение системы.

Нефункциональное тестирование описывает тесты, необходимые для определения характеристик программного обеспечения, которые могут быть измерены различными величинами. В целом, это тестирование того, "Как" система работает.

После проведения необходимых изменений, таких как исправление бага/дефекта, программное обеспечение должно быть пере тестировано для подтверждения того факта, что проблема была действительно решена.

7. Виртуализация - это процесс создания виртуальных машин на хостовом ПК. Каждая виртуальная машина представляет собой изолированную среду, в которой могут выполняться приложения и операционные системы. Это позволяет экономить ресурсы сервера и увеличивать его производительность.

Платформы виртуализации позволяют предприятиям и организациям экономить ресурсы серверов, увеличивать производительность и гибкость своих ИТ-инфраструктур. Они позволяют создавать виртуальные машины на одном физическом сервере, что позволяет использовать его ресурсы более эффективно. Это особенно полезно для компаний, у которых есть несколько приложений, работающих на разных операционных системах и языках программирования.

Также платформы виртуализации предоставляют возможность быстрого развертывания новых приложений и сервисов, что позволяет быстро реагировать на изменения в бизнес-среде. Кроме того, они обеспечивают безопасность данных, так как виртуальные машины изолированы друг от друга и от физической инфраструктуры.

8. Антивирусные программы - программное обеспечение, которое предназначено для обнаружения и удаления компьютерных вирусов и других вредоносных программ. Они также могут предоставлять защиту от других угроз, таких как шпионское ПО, рекламное ПО и трояны. Антивирусное программное обеспечение может быть установлено на компьютерах, мобильных устройствах или серверах. Некоторые антивирусные программы также предоставляют функции защиты от фишинга, спама и других онлайн-угроз.

Различают следующие виды антивирусных программ:

- детекторы (сканеры);
- доктора (или фаги, дезинфекторы);
- ревизоры;
- фильтры (сторожа, мониторы);
- иммунизаторы.

Существует несколько основополагающих методов поиска вирусов, которые применяются антивирусными программами:

- Сканирование - заключается в поиске сигнатур, выделенных из ранее обнаруженных вирусов.
- Эвристический анализ – заключается в предположении о том, что действия каких-либо программ являются вирусоподобными.
- Обнаружение изменений. Заражая компьютер, вирус делает изменения на жестком диске: дописывает свой код в заражаемый файл, изменяет системные области диска и т. д. На обнаружении таких изменений основываются работа антивирусных программ-ревизоров.
- Резидентные мониторы - антивирусные программы, постоянно находящиеся в оперативной памяти компьютера и отслеживающие все подозрительные действия, выполняемые другими программами.

9. BIOS (Basic Input/Output System) — это базовая система ввода-вывода, которая является программным обеспечением, которое находится на материнской плате компьютера. BIOS отвечает за загрузку операционной системы и настройку параметров работы компьютера. Он также содержит информацию о конфигурации компьютера, такую

как тип процессора, объем оперативной памяти и другие параметры.

BIOS выполняет несколько важных функций. Во-первых, он отвечает за загрузку компьютера и настройку его параметров. Без BIOS компьютер не сможет загрузиться и работать. Во-вторых, BIOS позволяет настраивать параметры работы компьютера, такие как частота процессора, скорость работы жесткого диска и другие. В-третьих, BIOS хранит информацию о конфигурации компьютера и позволяет обновлять его программное обеспечение. Наконец, BIOS обеспечивает безопасность компьютера, проверяя правильность загрузки операционной системы и защищая компьютер от вирусов и других угроз.

10. Резервирование — это процесс создания копии данных или системы, чтобы предотвратить потерю информации в случае сбоя или повреждения. Оно может быть выполнено на физическом уровне (например, копирование данных на внешний жесткий диск) или на уровне программного обеспечения (например, создание резервной копии файлов на облачном хранилище).

Резервное копирование данных необходимо для того, чтобы защитить данные от потери в случае сбоя, аварии или повреждения оборудования. Это позволяет быстро восстановить данные в случае необходимости, что особенно важно для бизнеса и организаций, где данные являются критическими. Резервное копирование также помогает снизить риски потери данных и защитить репутацию компании.

11. • Инкрементное внедрение - это метод, при котором новое программное обеспечение внедряется постепенно, по частям. Этот метод подходит для компаний, которые не могут позволить себе полностью изменить свою инфраструктуру или бизнес-процессы.

- Параллельное внедрение - это метод, при котором старое и новое программное обеспечение работают одновременно. Этот метод позволяет быстро перейти на новое программное обеспечение и избежать проблем с переходом на новое ПО.

- Пилотное внедрение - это метод, при котором новая система тестируется на небольшой группе пользователей или на отдельном отделе компании. Этот метод помогает выявить проблемы до того, как они станут серьезными, и позволяет быстро внести изменения в систему.

- Полное внедрение - это метод, при котором старая система полностью заменяется новой системой. Этот метод обычно используется при серьезных изменениях в бизнес-процессах или при переходе на новую версию программного обеспечения.

12. Аппаратно-программная платформа серверов и рабочих станций отличается по нескольким параметрам:

Архитектура: серверы обычно имеют более мощную архитектуру с большим количеством процессоров, оперативной памяти и жестких дисков, чем рабочие станции.

Назначение: серверы предназначены для обработки больших объемов данных и выполнения сложных задач, в то время как рабочие станции используются для работы с документами, графикой и мультимедиа.

Производительность: серверы могут обрабатывать большие объемы данных быстрее, чем рабочие станции, что делает их более эффективными для обработки данных.

Надежность: серверы имеют более высокую надежность и отказоустойчивость, чем рабочие станции, благодаря использованию более надежных компонентов и резервированию важных компонентов.

13. 1. DRM (Digital Rights Management) - технология защиты от копирования,

которая используется в различных форматах контента, таких как музыка, фильмы, игры и т.д.

2. Шифрование - метод защиты информации, при котором данные шифруются с помощью ключа, что делает их недоступными для чтения без соответствующего ключа.

3. Лицензирование - система контроля за использованием программного обеспечения, при которой пользователь должен купить лицензию на использование программы.

4. Ограничение доступа - метод защиты, при котором доступ к данным ограничен определенным кругом лиц.

5. Аутентификация - процесс проверки подлинности пользователя перед предоставлением ему доступа к системе или данным.

Эти средства защиты от копирования могут использоваться как отдельно, так и в комбинации друг с другом для обеспечения безопасности данных и предотвращения несанкционированного доступа к ним.

14. Методы и средства защиты компьютерной системы включают в себя:

Антивирусное программное обеспечение: это программное обеспечение, предназначенное для обнаружения и удаления вирусов и других вредоносных программ с компьютера.

Файрволы: это программное обеспечение, которое контролирует доступ к компьютеру через сеть и блокирует нежелательные соединения и трафик.

Криптография: это методы шифрования данных, позволяющие защитить их от несанкционированного доступа.

Аутентификация пользователей: это процесс проверки личности пользователя перед предоставлением доступа к компьютерной системе.

Двухфакторная аутентификация: это метод аутентификации, который требует дополнительного подтверждения личности пользователя, например, с помощью SMS-сообщения или специального кода.

Резервное копирование данных: это процесс создания копии важных данных на случай их потери или повреждения.

Сетевые экраны: это программное обеспечение для защиты сети от внешних угроз, таких как хакерские атаки или вирусы.

Пароли: это секретные коды, которые используются для аутентификации пользователей и защиты данных.

15. Вредоносные программы – это программы, намеренно разработанные и внедряемые для нанесения ущерба компьютерам и компьютерным системам. Существует несколько видов вредоносных программ:

- Шпионские программы – это разновидность вредоносных программ, скрывающихся на устройстве, отслеживающих активность и осуществляющих кражу конфиденциальной информации: финансовых данных, учетных записей, данных для входа и прочих данных.

- Программы-вымогатели – это вредоносные программы, осуществляющие блокировку или отказ доступа пользователей к системе или данным до момента выплаты выкупа. Программы-шифровальщики – это тип программ-вымогателей, выполняющих шифрование пользовательских файлов и требующих оплаты в определенный срок и часто в цифровой валюте, например, в биткойнах.

- Троянские программы маскируются под легальное программное

обеспечение, чтобы обманом заставить пользователей запустить вредоносные программы на компьютере.

- Черви – это один из наиболее часто встречающихся типов вредоносных программ, распространяющихся по компьютерным сетям, используя уязвимости операционной системы. Черви представляют собой отдельные программы, распространяющиеся путем самокопирования с целью заражения других компьютеров, при этом никаких действий со стороны пользователей или злоумышленников не требуется.

- Вирус – это фрагмент кода, который вставляется в приложение и запускается при его запуске. Попав в сеть, вирус может использоваться для кражи конфиденциальных данных, запуска DDoS-атак или атак программ-вымогателей. Обычно вирус распространяется через зараженные веб-сайты, при совместном доступе к файлам, при загрузке зараженных вложений электронной почты.

- Бот – это компьютер, зараженный вредоносной программой, которым злоумышленники могут управлять удаленно. Боты, иногда называемые зомби-компьютерами, могут использоваться для запуска атак, а также стать частью ботнета – набора ботов, объединенных в сеть.

16. Существует множество видов алгоритмических ошибок в программном обеспечении, вот некоторые из них:

1. Ошибки в логике программы: это ошибки, которые происходят из-за неправильной логики в программе. Например, если программа не выполняет ожидаемые действия, если ввод пользователя не обрабатывается должным образом или если код не соответствует требованиям задачи.

2. Ошибки времени выполнения: это ошибки, возникающие в процессе выполнения программы. Они могут быть связаны с неверными вычислениями, неверными данными или ошибками доступа к памяти.

3. Ошибки синхронизации: это проблемы с синхронизацией между различными частями программы, которые могут привести к неправильному поведению или сбоям.

4. Ошибки ввода-вывода: это ошибки в работе с файлами и другими источниками данных, которые могут возникнуть при неправильном чтении, записи или обработке данных.

5. Ошибки безопасности: это ошибки, связанные с безопасностью программы, такие как уязвимости в системе безопасности или отсутствие защиты от несанкционированного доступа.

6. Ошибки производительности: это проблемы, связанные со скоростью работы программы или ее потреблением ресурсов, такие как перегрузка процессора или недостаток памяти.

17. Оперативные методы повышения надежности:

1. Временная избыточность;

2. Информационная избыточность;

3. Программная избыточность.

- Временная избыточность состоит в использовании некоторой части производительности компьютера для контроля исполнения программ и восстановления (рестарта) вычислительного процесса. Для этого при проектировании систем должен предусматриваться запас производительности, который будет использоваться на контроль и оперативное повышение надежности функционирования. Величина временной избыточности зависит от требований к надежности и находится в пределах от 5-10%

производительности процессора до 3-4-кратного дублирования производительности отдельной машины в многопроцессорных вычислительных комплексах.

- Информационная избыточность состоит в дублировании исходных и промежуточных данных, обрабатываемых программами. Избыточность используется для сохранения достоверности данных, которые в наибольшей степени влияют на нормальное функционирование ПС и требуют значительного времени на восстановление. Их защищают 2-3 кратным дублированием с периодическим обновлением.

- Программная избыточность используется для контроля и обеспечения достоверности наиболее важных решений ПО обработке информации. Она заключается в сопоставлении результатов обработки одинаковых исходных данных Программами, различающимися используемыми алгоритмами, и в исключении искажений при несовпадении результатов. Программная избыточность необходима также для реализации программ автоматического контроля и восстановления данных с использованием информационной избыточности и для функционирования всех средств обеспечения надежности, использующих временную избыточность.

18. Риски при создании программного обеспечения можно разделить на следующие категории:

1. Технические риски: связаны с ошибками в коде, неправильным выбором технологий и инструментов, а также с проблемами совместимости с аппаратным обеспечением.

2. Бизнес-риски: связаны с изменением требований заказчика, изменением бизнес-процессов или с изменениями в законодательстве.

3. Регуляторные риски: связаны с требованиями регуляторов к безопасности и защите данных.

4. Юридические риски: связаны с нарушением авторских прав, нарушением конфиденциальности данных и другими юридическими вопросами.

5. Экологические риски: связаны с воздействием программного обеспечения на окружающую среду и наоборот, а также необходимостью соблюдения экологических стандартов.

19. Разработка и внедрение программного продукта - это длительный и сложный процесс, который включает в себя несколько этапов. Вот общая последовательность действий:

Анализ требований: на этом этапе определяются требования к программному продукту, его функциональность, возможности, ограничения и т.д.

Проектирование: на этом этапе разрабатывается дизайн программного продукта, определяются его структура, интерфейсы, алгоритмы и т.д.

Программирование: на этом этапе осуществляется написание кода программного продукта на выбранном языке программирования.

Тестирование: на этом этапе производится тестирование программного продукта на соответствие требованиям и исправление выявленных ошибок.

Внедрение: на этом этапе программное обеспечение устанавливается на компьютеры пользователей и настраивается для работы.

Поддержка и обновление: на этом этапе обеспечивается поддержка и обновление программного продукта, исправление ошибок и добавление новых функций.

20. Дестабилизирующие факторы — это факторы, которые могут повлиять на надежность программного средства. К ним относятся:

- Ошибки в коде: ошибки в коде могут привести к неправильному поведению программы или к ее остановке.
- Неправильная настройка: неправильная настройка параметров может привести к некорректной работе программы.
- Отсутствие обновлений: если программное средство не обновляется, то оно может стать уязвимым для атак хакеров.
- Недостаточная защита: если программное средство не защищено должным образом, то оно может быть уязвимым для злоумышленников.
- Несовместимость с операционной системой: если программное средство несовместимо с операционной системой, то оно может работать некорректно.
- Перегрузка процессора: если процессор перегружен, то программное средство может работать медленно или вообще перестать работать.
- Низкая производительность: если программное средство имеет низкую производительность, то пользователи могут испытывать задержки при работе с ним.
- Сбои в работе: сбои в работе программного средства могут привести к потере данных или к остановке работы.

Угрозы надежности программных средств — это угрозы, которые могут снизить надежность программного средства. К ним относятся:

- Хакерские атаки: хакеры могут атаковать программное средство, чтобы получить доступ к конфиденциальной информации или нарушить его работу.
- Вирусы: вирусы могут заражать программное средство и нарушать его работу.
- Отказ оборудования: отказ оборудования может привести к сбоям в работе программного средства.
- Нестабильность сети: нестабильность сети может привести к ошибкам в передаче данных между программным средством и другими системами.
- Человеческий фактор: человеческий фактор может привести к ошибкам при работе с программным средством.
- Изменение требований: изменение требований к программному средству может привести к необходимости его доработки или замены.

21. Основные критерии качества программного обеспечения - это:

- функциональность – способность ПО выполнять набор функций (действий), удовлетворяющих заданным или подразумеваемым потребностям пользователей. Набор указанных функций определяется во внешнем описании ПО;
- надежность – это его способность с достаточно большой вероятностью безотказно выполнять определенные функции при заданных условиях и в течение заданного периода времени;
- эффективность – соотношение уровня услуг, предоставляемых ПО пользователю при заданных условиях, и объема используемых для этого ресурсов. К числу таких ресурсов могут относиться требуемые аппаратные средства, время выполнения программ, затраты на подготовку данных и интерпретацию результатов;
- эргономичность – характеристики ПО, которые позволяют минимизировать усилия пользователя по подготовке исходных данных, применению ПО и оценке полученных результатов, а также вызывать положительные эмоции определенного или подразумеваемого пользователя;
- модифицируемость – характеристики ПО, которые позволяют

минимизировать усилия по внесению изменений для устранения ошибок и по его модификации в соответствии с изменяющимися потребностями пользователей. Модифицируемость ПО существенно зависит от степени и качества его документированности;

- мобильность – способность ПО быть перенесенным из одной среды (окружения) в другую, в частности, с одной аппаратной платформы на другую.

22. Применительно к информационным системам ошибки, возникающие на различных этапах процесса разработки, группируются следующим образом:

- системные ошибки. К ним относятся ошибки в формулировании требований, в описаниях целей и спецификаций, включая спецификации оборудования и общего ПО. Эти ошибки проявляются в результате отклонения характеристик функционирования ПО в системе и характеристик взаимодействующих объектов от предполагаемых при проектировании. При автономной и начальной отладке программ доля системных ошибок невелика (примерно 10% [5]), но она существенно возрастает (до 35—40%) на завершающих этапах комплексной отладки. В процессе эксплуатации системные ошибки являются преобладающими;

- алгоритмические ошибки. К ним относятся ошибки детального проектирования, спецификации оборудования и общего ПО, а также ошибки в спецификации языка программирования. К алгоритмическим ошибкам в значительной мере относятся и ошибки, выявленные при модификации программы и связанные с изменениями в алгоритме при корректировке существующих или добавлении новых функций. Алгоритмические ошибки – это в первую очередь ошибки, обусловленные некорректной постановкой функциональных задач: когда не полностью оговорены условия, необходимые для получения правильного результата. Именно такие ошибки являются наиболее частыми и составляют около 70% всех алгоритмических ошибок. Около 20% алгоритмических ошибок составляют ошибки сопряжения функциональных подпрограмм. Оставшаяся часть алгоритмических ошибок приходится на просчеты в использовании ресурсов компьютерных сетей и мейнфреймов в информационных системах;

- программные ошибки – ошибки кодирования программ, ошибки в спецификации оборудования, общего ПО и языка программирования, а также ошибки, возникшие при модификации ПО.

23. Информационная безопасность — это комплекс мер, направленных на защиту информации от несанкционированного доступа, изменения, уничтожения, раскрытия или доступа к ней. Она включает в себя защиту данных, сетей, приложений, систем и устройств от различных угроз, таких как вирусы, хакеры, взломщики, злоумышленники и т. д. Информационная безопасность является важным аспектом для любой организации, поскольку она защищает ее ресурсы и данные от потери или кражи.

Есть три основных составляющих информационной безопасности: доступность, целостность и конфиденциальность.

Доступность – это возможность за приемлемое время получить требуемую информационную услугу. Под целостностью подразумевается актуальность и непротиворечивость информации, ее защищенность от разрушения и несанкционированного изменения. Конфиденциальность – это защита от несанкционированного доступа к информации.

24. Классифицируют два основных типа алгоритмов шифрования – симметричные и асимметричные.

Симметричные алгоритмы шифрования используют один и тот же ключ для шифрования и дешифрования данных, что обеспечивает высокую скорость шифрования и дешифрования. Однако они имеют ограничения на длину ключа и могут быть взломаны с помощью криптоанализа.

Асимметричные алгоритмы, напротив, используют два разных ключа: открытый ключ для шифрования данных и закрытый ключ для их дешифрования. Они обеспечивают большую безопасность, поскольку взлом одного ключа не дает доступа к другому. Однако они медленнее, чем симметричные алгоритмы, и требуют больше ресурсов для работы.

25. Идентификация - это процесс определения личности пользователя или объекта, который пытается получить доступ к системе. Идентификация может происходить с помощью различных методов, включая ввод логина и пароля, использование биометрических данных (например, отпечатков пальцев или распознавания лица) или использование токенов (физических устройств, которые содержат уникальный идентификатор пользователя).

Авторизация - это процесс предоставления пользователю или объекту разрешения на доступ к определенной информации или ресурсам в системе. Авторизация может происходить автоматически или по запросу пользователя, и обычно основывается на уровне доступа пользователя или объекта.

Аутентификация - это процесс проверки подлинности пользователя или объекта при попытке получить доступ к защищенной информационной системе. Аутентификация может осуществляться с помощью различных механизмов, включая проверку логина и пароля пользователя, использование токенов или биометрических данных, а также использование сертификатов безопасности.

Последовательность выполнения этих понятий может выглядеть следующим образом:

Идентификация: пользователь вводит логин и пароль или использует другой механизм идентификации.

Авторизация: система проверяет уровень доступа пользователя и предоставляет ему разрешение на доступ к определенным ресурсам.

Аутентификация: система проверяет подлинность пользователя и его право на доступ к системе. Если аутентификация проходит успешно, пользователю предоставляется доступ к ресурсам, иначе доступ будет заблокирован.

26. Существует несколько видов программного обеспечения:

Системное программное обеспечение - это программы, которые управляют работой компьютера и обеспечивают его взаимодействие с пользователем. К системному программному обеспечению относятся операционные системы, драйверы устройств, утилиты и другие программы.

Прикладное программное обеспечение - это программы, предназначенные для решения определенных задач или выполнения конкретных функций. Прикладное программное обеспечение может быть создано для различных целей, таких как создание веб-сайтов, работа с электронными таблицами, обработка изображений и многое другое.

Системы программирования - это инструменты, используемые для разработки программного обеспечения. Они предоставляют набор инструментов и языков программирования для написания кода и создания приложений.

Средства разработки - это программы и инструменты, которые помогают программистам создавать и поддерживать приложения. Они могут включать в себя

интегрированные среды разработки (IDE), компиляторы, отладчики и другие инструменты.

27. Производительность компьютера зависит от многих факторов, таких как процессор, оперативная память, жесткий диск, видеокарта и операционная система. Чтобы улучшить производительность, можно выполнить следующие действия:

Обновить процессор: если ваш процессор устарел, то это может замедлять работу компьютера. Вы можете обновить процессор, купив более мощный процессор или заменив старый на новый.

Увеличить объем оперативной памяти: увеличение объема оперативной памяти может улучшить производительность компьютера, особенно если вы работаете с большими файлами или используете многозадачность.

Обновить жесткий диск: если жесткий диск медленный или имеет недостаточную емкость, это может привести к замедлению работы компьютера. Вы можете заменить жесткий диск на более быстрый и большой.

Установить более быструю видеокарту: если у вас установлена старая или медленная видеокарта, это может замедлить работу компьютера при запуске игр или графических приложений. Вы можете установить более мощную видеокарту.

Обновить операционную систему: некоторые операционные системы могут работать медленнее других. Вы можете попробовать обновить операционную систему до последней версии.

Очистить жесткий диск от ненужных файлов: хранение большого количества ненужных файлов на жестком диске может замедлять его работу. Вы можете использовать программы для очистки жесткого диска от ненужных файлов.

Отключить ненужные программы: запуск множества программ одновременно может замедлять работу компьютера, так как они используют ресурсы процессора и оперативной памяти. Вы можете отключить ненужные программы или закрыть их, когда они не используются.

Использовать SSD-накопитель: использование твердотельного накопителя (SSD) вместо жесткого диска может значительно ускорить работу компьютера, особенно при запуске программ и работе с файлами.

28. Безопасный режим (Safe Mode) – это режим загрузки операционной системы Windows, при котором загружаются только необходимые для работы системы драйверы и службы. Этот режим используется для диагностики и устранения проблем в работе компьютера.

В безопасном режиме можно запустить диагностику системы, чтобы выявить проблемы, которые могут препятствовать нормальной работе компьютера. Например, если компьютер не загружается, можно попробовать запустить его в безопасном режиме и проверить, запускается ли система в этом режиме.

Также в безопасном режиме можно проверить наличие вирусов или других вредоносных программ, которые могут мешать нормальной работе компьютера. Для этого нужно запустить антивирусную программу и проверить компьютер на наличие угроз в безопасном режиме.

29. Эксплуатационные характеристики качества программного обеспечения определяют, как хорошо ПО функционирует в реальной эксплуатационной среде. Они включают в себя такие характеристики, как устойчивость к нагрузкам, быстродействие, отказоустойчивость, масштабируемость, совместимость и другие. Измерение эксплуатационных характеристик производится путем тестирования и мониторинга работы

ПО в реальных условиях. Например, при тестировании устойчивости к нагрузкам производятся нагрузочные тесты, которые позволяют проверить, как ПО ведет себя при большом количестве запросов или пользователей. Примером эксплуатационной характеристики является отказоустойчивость, которая определяет, насколько быстро и безопасно система восстанавливается после сбоев или отказов.

30. Сервер и рабочая станция - это два разных типа компьютеров, которые отличаются по своим функциям и назначению.

Сервер - это компьютер, который используется для хранения и обработки данных, а также для предоставления доступа к этим данным другим компьютерам в сети. Серверы обычно имеют более мощные процессоры, больше оперативной памяти и хранилища, а также специальное программное обеспечение для управления данными и обеспечения безопасности.

Рабочая станция - это персональный компьютер, который предназначен для выполнения задач пользователя. Рабочие станции обычно имеют меньше ресурсов, чем серверы, и используются для выполнения различных задач, таких как работа с офисными приложениями, просмотр веб-страниц, игры и т.д.

Таким образом, основное отличие между сервером и рабочей станцией заключается в их назначении и ресурсах, которые они используют для выполнения своих задач.

3.3. Практико-ориентированные задания к экзамену:

1. У вас на работе не запускается ПК. Какие действия вы будете предпринимать?
2. Прокомментировать программу построчно и определить, что она делает. Задача 1.1.

```
a = int(input())
b = int(input())
```

```
def c(x, y):
    z = x + y
    return z / 2
```

```
def d(x, y):
    z = x * y
    return z * 2
```

```
def e(x, y):
    z = x / y
    return z ** 2
```

```
c(a, b)
d(a, b)
e(a, b)
```

3. Дана программа, которая ранее работала на Windows XP. Вам необходимо запустить ее на Windows 8.1. Какие способы решения проблем совместимости в данном случае можно использовать?
4. Вам принесли ПК из другого подразделения, вам необходимо протестировать работоспособность аппаратного обеспечения для определения будущего расположения ПК. Как вы будете это делать?
5. Дан список функций с некорректно поименованными переменными. Необходимо

поименовать переменные согласно результату выполнения функции. Задача 2.1

```
a = int(input())
```

```
b = int(input())
```

```
def c(x, y):
```

```
    z = x + y
```

```
    return z
```

```
def d(x, y):
```

```
    z = x * y
```

```
    return z
```

```
def e(x, y):
```

```
    z = x / y
```

```
    return z
```

```
c(a, b)
```

```
d(a, b)
```

```
e(a, b)
```

6. Вы развиваете ваш программный продукт и вам необходимо определить, какие возможности необходимо в него добавить. Как вы сможете это сделать?
7. Вам необходимо составить краткое руководство пользователя по программе VMWare Workstation. Что в него будет входить?
8. Ваш компьютер начал активно использовать ресурсы. Что это значит и как с этим справиться?
9. Компьютер бухгалтера начал сильно «тормозить». Что можно предпринять для улучшения производительности?
10. Прокомментировать программу построчно и определить, что она делает. Задача 1.2.

```
a = input() # <- '1:1:1' # <- '1:1:1'
```

```
def b(c):
```

```
    d = c.split(':')
```

```
    for i, j in enumerate(d):
```

```
        if not len(j) > 1:
```

```
            d[i] = '0' + j
```

```
    return ':'.join(d)
```

```
print(b(a)) # <- '1:1:1'
```

11. Дан список функций с некорректно поименованными переменными. Необходимо поименовать переменные согласно результату выполнения функции. Задача 2.2

```
a = input() # <- '1:1:1'
```

```
def b(c):
```

```
    d = c.split(':')
```

```
    for i, j in enumerate(d):
```

```
        if not len(j) > 1:
```

```
            d[i] = '0' + j
```

```
    return ':'.join(d)
```

```
print(b(a)) # -> '01:01:01'
```

12. Системный администратор дал вам задание создать домен на Windows Server 2008 и подключить к нему пользователей. Как вы будете это делать?
13. Вам необходимо защитить документ от несанкционированного изменения. Какими способами это можно сделать?
14. Вы написали программу, одна из функций которой обращается к файлу в папке по адресу C:\\Windows\\System32. Что это может за собой повлечь?
15. Прокомментировать программу построчно и определить, что она делает. Задача 1.3.
a = [1, 0, True, False]
b = [1, 1, True, True]

```
def c(d):  
    flag = d[0]  
  
    for i in d:  
        if flag != i:  
            return False  
    return True
```

```
print(c(a))  
print(c(b))
```

16. Вы установили новый браузер, но при помощи него ни одна страница в Интернет не открывается. На вашем старом браузере все работает. Как решить эту проблему?
17. Дан список функций с некорректно поименованными переменными. Необходимо поименовать переменные согласно результату выполнения функции. Задача 2.3.

```
a = [1, 0, True, False]  
b = [1, 1, True, True]  
def c(d):
```

```
    flag = d[0]  
    for i in d:  
        if flag != i:  
            return False  
    return True
```

```
print(c(a))  
print(c(b))
```

18. Прокомментировать программу построчно и определить, что она делает. Задача 1.4.
a = int(input()) # 5

```
def b(c, d = True):  
  
    if d:  
        e = range(c + 1)  
    else:  
        e = range(c + 1)[::-1]  
  
    for i in e:  
        print('* ' * i)
```

```
b(a, True)
```

```
b(a, False)
```

19. Вы закупаете новые компьютеры в компанию. Каким образом вы будете настраивать ПК для бухгалтеров и менеджеров? И в чем будет принципиальное отличие настроек для разных типов пользователей?

20. Найти ошибку в коде программы и устранить ее. Задача 3.1.

```
a = input()
```

```
b = input()
```

```
def c(x, y):
```

```
    z = x + y
```

```
    return z
```

```
def d(x, y):
```

```
    z = x * y
```

```
    return x
```

```
def e(x, y):
```

```
    z = x / y
```

```
    return y
```

```
c(a, b)
```

```
d(a, b)
```

```
e(a, b)
```

21. Найти ошибку в коде программы и устранить ее. Задача 3.2.

```
a = input() # <- '1:1:1'
```

```
def b(c):
```

```
    d = c.split(':')
```

```
    for i, j in enumerate(d):
```

```
        if not len(j) > 1:
```

```
            d[j] = '0' + i
```

```
    return ' '.join(d)
```

```
print(b(a)) # <- '1:1:1'
```

22. Вам необходимо создать учетные записи пользователей с различными правами. Какими способами это можно сделать?

23. Дан список функций с некорректно поименованными переменными. Необходимо поименовать переменные согласно результату выполнения функции. Задача 2.4.

```
a = int(input()) # 5
```

```
def b(c, d = True):
```

```
    if d:
```

```
        e = range(c + 1)
```

```
    else:
```

```
        e = range(c + 1)[::-1]
```

```
    for i in e:
```

```
        print('* ' * i)
```

```
b(a, True)
```

```
b(a, False)
```

24. Найти ошибку в коде программы и устранить ее. Задача 3.3.

```
a = [1, 0, True, False]
```

```

b = [1, 1, True, True]
def c(d):
    flag = d[0]
    for i in d:
        if flag != i:
            return False
    return True
print(c(a))
print(c(b))

```

25. Вам необходимо протестировать несколько программ, скачанных из сети Интернет. Каким образом вы можете это сделать без последствий для своего компьютера?

26. Найти ошибку в коде программы и устранить ее. Задача 3.4.

```

a = int (input ()) # 5
def b(c, d = True):
    if d:
        e = range(c)
    else:
        e = range(c)[::-1]
    for i in e:
        print ('* ' * i)
b (a, True)
b (a, False)

```

27. В офисе организации часто отключают электричество. Что можно предпринять, чтобы защитить информацию от потери или повреждения?

28. Вам необходимо собрать несколько ПК под разные задачи. Чем вы будете руководствоваться при сборке и как будете подбирать аппаратную конфигурацию?

29. У вас на ПК появился BSOD, после этого операционная система не загружается, переходя в цикличную перезагрузку. Что могло стать причиной и какими способами можно попробовать решить эту проблему без переустановки операционной системы?

30. Вы входите в команду разработчиков программных систем. У вашей компании появился заказчик. Какие ваши дальнейшие действия?

Эталоны ответов на практические задания:

1. Первоначально необходимо проверить все соединения кабелей, проверить кабель питания, включен ли блок питания на ПК.

Вторым этапом будет визуальная проверка аппаратной части персонального компьютера: проверка сокета процессора, оперативной памяти, видеокарты, жесткого диска. Необходимо проверить качество соединения модулей с материнской платой.

Если, после второго этапа, проблема не была решена, то необходима проверка при помощи специализированного оборудования, которое позволит проверить питание на материнской плате и сопутствующих устройствах ПК и определить устройства, которые вызвали неисправность.

Если специализированного оборудования для проверки нет, то можно воспользоваться методом замены, подключая заведомо рабочее аппаратное обеспечение к работающему ПК.

2. # Принимаем от пользователя значения `a` и `b`,


```

# после - приводим значения к типу int.
a = int (input ())
b = int (input ())
# Функция `c` возвращает сумму аргументов `x` и `y`,
# поделенную надвое.
def c (x, y):
    # Складываем принимаемые аргументы `x` и `y`,
    # и помещаем результат сложения в переменную `z`.
    z = x + y

    # Возвращаем переменную `z`, поделенную пополам.
    return z / 2
# Функция `d` возвращает сумму аргументов `x` и `y`,
# умноженное надвое.
def d(x, y):
    # Умножаем принимаемые аргументы `x` и `y`,
    # и помещаем результат умножения в переменную `z`.
    z = x * y
    # Возвращаем переменную `z`, умноженную надвое.
    return z * 2
# Функция `e` возвращает сумму аргументов `x` и `y`,
# делённую надвое.
def e(x, y):
    # Делим принимаемые аргументы `x` и `y`,
    # и помещаем результат деления в переменную `z`.
    z = x / y
    # Возвращаем переменную `z`, возведённую в степень 2-ки.
    return z ** 2
# Вызов функций `c`, `d` и `e` с аргументами `a` и `b`,
# принимаемых от пользователя.
c(a, b)
d(a, b)
e(a, b)

```

3. В данном случае, первый способ проверки совместимости – использование виртуальной машины с установленной операционной системой Windows 8.1, приложение нужно попробовать установить на виртуальной машине и проверить ее работоспособность. В случае проблем с запуском необходимо, используя встроенные средства устранения проблем совместимости в операционную систему Windows, запустить программу. В случае невозможности запуска стоит проверить наличие обновлений программного обеспечения в сети Интернет или обратиться к разработчику, с целью получения более новой версии.

В случае отсутствия новой версии, на некоторый промежуток времени, пока не найдено другое программное обеспечение, удовлетворяющее требованиям, можно использовать виртуальную машину с установленной Windows XP, на которой будет запускаться только это приложение.

4. При тестировании аппаратного обеспечения необходимо исходить из его состава. Если необходим компьютер для сотрудника, который работает с графикой – нужна мощная

видеокарта и оперативная память. Для сотрудника, занимающегося математическими расчетами – необходим мощный процессор. Для сотрудника, который работает с документами достаточно будет самого простого оборудования. Чтобы определить конфигурацию оборудования проще всего использовать специализированное программное обеспечение, которое позволит определить, насколько тот или иной ПК подходит под заданные параметры. Некоторые виды программного обеспечения позволяют задать оптимальные характеристики, которым должен удовлетворять ПК и проводить тестирование на соответствие этим характеристикам.

5.

```
a: int = int (input ('Введите число a: '))
b: int = int (input ('Введите число b: '))
def addition (num1: int, num2: int) -> int:
    result: int = num1 + num2
    return result
def multiplication (num1: int, num2: int) -> int:
    result: int = num1 * num2
    return result
def division (num1: int, num2: int) -> float:
    result: float = x / y
    return result
addition (a, b)
multiplication (a, b)
division(a, b)
```

6. Чтобы определить, какие функции и возможности необходимо добавить в программный продукт, мы можем использовать различные методы и инструменты.

Во-первых, мы можем провести опрос пользователей, чтобы узнать, что им нужно и что они хотели бы видеть в нашем продукте. Мы можем также изучить рынок и конкурентов, чтобы понять, какие функции уже существуют и какие функции могут быть полезны для наших пользователей.

Кроме того, мы можем проводить фокус-группы и интервью с потенциальными пользователями, чтобы лучше понять их потребности и ожидания. Также мы можем использовать методы анализа данных, такие как анализ пользовательских отзывов, анализ статистики использования и т.д.

Наконец, мы можем привлекать экспертов в области разработки программного обеспечения, чтобы получить их мнение и рекомендации по поводу того, какие функции стоит добавить в наш продукт.

В целом, наша команда разработчиков будет использовать все эти методы и инструменты для определения наиболее важных функций и возможностей, которые необходимо добавить в наш программный продукт.

7. Краткое руководство пользователя по VMWare Workstation должно включать следующие разделы:

- Введение в VMWare Workstation
- Установка и настройка VMWare Workstation на компьютере
- Создание виртуальной машины;
- Настройка виртуальной машины;
- Запуск виртуальной машины;

- Управление виртуальной машиной;
- Подключение к виртуальной машине;
- Резервное копирование и восстановление виртуальной машины;

Краткое руководство пользователя по VMWare Workstation поможет пользователям быстро и легко начать работу с этой программой.

8. Если ваш компьютер начал активно использовать ресурсы, это может быть вызвано несколькими причинами. Некоторые из них могут быть связаны с активностью вредоносных программ или вирусов, которые могут использовать ресурсы компьютера для своей работы. Также это может происходить из-за того, что на вашем компьютере запущено слишком много процессов или приложений, которые используют ресурсы процессора или памяти.

Чтобы справиться с этой проблемой, вы можете попробовать следующие шаги:

1. Проверьте свой компьютер на наличие вирусов и вредоносных программ с помощью антивирусного программного обеспечения.

2. Закройте все ненужные процессы и приложения, которые работают в фоновом режиме.

3. Очистите кэш и временные файлы, чтобы освободить место на жестком диске и ускорить работу компьютера.

4. Обновите драйверы для вашей видеокарты и других устройств, чтобы они работали более эффективно.

5. Если проблема не устраняется, попробуйте перезагрузить компьютер и проверить, не улучшилась ли ситуация.

9 Для улучшения производительности компьютера можно предпринять следующие действия:

- Обновить драйвера для всех устройств, включая видеокарту, звуковую карту, сетевую карту и т.д.

- Очистить жесткий диск от ненужных файлов и приложений, которые занимают место.

- Удалить программы, которые не используются.

- Использовать программы для оптимизации работы компьютера, например, CCleaner или Advanced SystemCare.

- Установить программы для ускорения работы системы, например, RAM Booster или Game Booster.

- Заменить старые комплектующие на новые, если это возможно.

- Проверить систему на вирусы и удалите их, если они есть.

- Регулярно обновлять операционную систему и программы.

- Не запускать много программ одновременно, чтобы избежать перегрузки системы.

10.

```
a = input() # <- '1:1:1' (входные данные)
```

```
# Функция `b` принимает аргумент `c`, который выступает
```

```
# в качестве не нормализованных данных.
```

```
# К примеру, на вход функции `b` поступает '12:0:0',
```

```
# результатом работы будет является '12:00:00'.
```

```
def b(c):
```

```
    # Разделение строки на список по двоеточию.
```

```

d = c.split(':')

# Итерируемся по полученному списку.
for i, j in enumerate(d):
    # Если длина элемента меньше единицы,
    # то к этому элементу в начало добавляем нуль.
    if not len(j) > 1:
        # Замена происходит в самом списке,
        # по которому происходит итерация.
        d[i] = '0' + j

# После всех преобразований, возвращаем "склеенный" массив.
# В качестве разделителя между элементами выступает двоеточие.
return ':'.join(d)

```

```
print(b(a)) # <- '01:01:01' (результат)
```

11.

```
string_date: str = input("Введите дату в соответствии с шаблоном
[hh:mm:ss]:\n")
```

```
def date_normalize(date: str) -> str:
    date_list: list[str] = date.split(':')

    for index, item in enumerate(date_list):
        if not len(item) > 1:
            date_list[index] = '0' + item

    return ':'.join(date_list)

```

```
print(
    date_normalize(string_date)
)
```

12. Для создания домена на Windows Server 2008 и подключения к нему пользователей необходимо выполнить следующие шаги:

- Установить Windows Server 2008 на сервер и настроить его как домен-контроллер.
- Создать учетные записи пользователей и групп в Active Directory.
- Настроить параметры безопасности домена, такие как права доступа и политики безопасности.
 - Настроить подключение пользователей к домену, используя Active Directory Users and Computers или другой инструмент управления пользователями.
 - Проверить работоспособность подключения пользователей к домену путем входа в систему под их учетными записями.

13. Существует несколько способов защиты документа от несанкционированного изменения:

Шифрование: документ можно зашифровать с помощью пароля или криптографического алгоритма, что сделает его недоступным для чтения без знания пароля.

Ограничение прав доступа: можно ограничить права доступа к документу, чтобы

только определенные пользователи могли его просматривать или редактировать.

Резервное копирование: регулярное создание резервных копий документа позволит восстановить его в случае повреждения или удаления.

Использование проверенных источников: загрузка документа из надежного источника, такого как официальный сайт или доверенный партнер, может помочь предотвратить его изменение третьими лицами.

Проверка подписей: если документ содержит цифровую подпись, это может помочь убедиться в его подлинности и защитить от изменения.

Регулярное обновление: обновление программного обеспечения и антивирусных программ может помочь защитить документ от вредоносных программ и атак.

14. Обращение к файлу в этой папке может привести к различным последствиям, в зависимости от того, какие данные хранятся в этом файле. Например, если файл содержит важные системные настройки, то изменение его содержимого может привести к сбоям в работе операционной системы. Также, если этот файл содержит конфиденциальную информацию, то его несанкционированный доступ может привести к утечке данных. Поэтому, перед тем как обращаться к файлу в такой папке, необходимо убедиться в том, что это безопасно и не приведет к негативным последствиям.

Также, если операционная система пользователя установлена на диск, у которого другая буква диска, то программа просто не сможет работать и выключится с ошибкой.

15.

```
a = [1, 0, True, False] # список разных значений
b = [1, 1, True, True] # список одинаковых значений [True == 1]

# Функция `c` принимает некоторый список значений `d`.
# Если значения в этом списке идентичны друг другу (одинаковые),
# то функция возвращает True, иначе - False.
def c(d):
    # Первый элемент списка выбирается в качестве состояния всего списка.
    flag = d[0]

    # Итерация по списку значений.
    for i in d:
        # Если состояние списка (flag) не совпадает с произвольным
        # элементом массива, то возвращается False, иначе - True.
        if flag != i:
            return False
    return True

# Вывод результатов работы функции `c` для списков значений `a` и `b`.
print(c(a))
print(c(b))
```

16. Если вы установили новый браузер и он не работает, возможно, проблема связана с настройками безопасности вашего компьютера. Необходимо выполнить следующие шаги:

- Проверьте настройки безопасности вашего компьютера: убедитесь, что настройки безопасности не блокируют доступ к Интернету. Особое внимание следует уделить настройке брандмауэра.

- Установите последние обновления для вашего браузера: иногда проблемы с браузером могут быть связаны с устаревшими версиями программного обеспечения.

- Очистите кэш и куки вашего браузера: это может помочь исправить проблемы с отображением страниц.

- Попробуйте использовать другой браузер: если проблема не решена, попробуйте использовать другой браузер.

17.

```
array1: list[any] = [1, 0, True, False]
```

```
array2: list[any] = [1, 1, True, True]
```

```
def every(array: list[any]) -> bool:
```

```
    flag: any = array[0]
```

```
    for item in array:
```

```
        if flag == item:
```

```
            return False
```

```
    return True
```

```
print(every(a))
```

```
print(every(b))
```

18.

```
# Принимается данные от пользователя и приводятся к типу int.
```

```
a = int(input()) # <- 5
```

```
# Функция `b` принимает некоторое число, на основе которого рисует
```

```
# треугольник из символов звездочки (*). Дополнительный аргумент `d`
```

```
# определяет, в каком направлении будет смотреть треугольник.
```

```
# По умолчанию `d` находится в состоянии True: треугольник смотрит
```

```
# острым углом вверх, иначе, если False, - вниз.
```

```
def b(c, d = True):
```

```
    # Если `d` является True, то формируется список из цифр
```

```
    # от 0, 1, 2... до n. Иначе: n, ...2, 1, 0. Это необходимо для того,
```

```
    # чтобы выводить длину среза треугольника нужного размера.
```

```
    if d:
```

```
        e = range(c + 1)
```

```
    else:
```

```
        e = range(c + 1)[::-1]
```

```
# Итерация по списку значений с последующим выводом в консоль
```

```
# некоторого количества звездочек (*).
```

```
for i in e:
```

```
    print('* ' * i)
```

```
        # Вывод результата для одного и того же значения,
```

```
        # но с разным направлением отображения.
```

```
        b(a, True)
```

```
        b(a, False)
```

19. Для бухгалтеров и менеджеров ПК стоит настроить следующим образом:

1. Установка операционной системы Windows.

2. Установка необходимых программы для работы с документами, бухгалтерским учетом и CRM.

3. Настройка параметров безопасности для защиты данных от

несанкционированного доступа.

4. Настройка программы для автоматического обновления системы и установки новых программ.

5. Настройка автоматического запуска программ при включении компьютера.

6. Настройка быстрого доступа к часто используемым программам и файлам.

7. В случае, если ПК будет использовать несколько человек, то для каждого из них должна быть создана своя учетная запись с высокой степенью ограничений на изменения документов и файлов.

Настройки для разных типов пользователей могут отличаться в зависимости от их потребностей и предпочтений. Например, для пользователей, которые часто работают с большими объемами данных, можно настроить более мощные параметры производительности и памяти, чтобы ускорить работу приложения. Для пользователей, которые нуждаются в более простом и интуитивно понятном интерфейсе, можно выбрать более простой дизайн и меньше функций. Также можно настроить различные уровни доступа к определенным функциям и данным в зависимости от роли пользователя в компании.

20.

```
a = int(input())
```

```
b = int(input())
```

```
def c(x, y):
```

```
    z = x + y
```

```
    return z
```

```
def d(x, y):
```

```
    z = x * y
```

```
    return z
```

```
def e(x, y):
```

```
    z = x / y
```

```
    return z
```

```
c(a, b)
```

```
d(a, b)
```

```
e(a, b)
```

21.

```
a = input() # <- '1:1:1'
```

```
def b(c):
```

```
    d = c.split(':')
```

```
    for i, j in enumerate(d):
```

```
        if not len(j) > 1:
```

```
            d[i] = '0' + j
```

```
    return ':'.join(d)
```

```
print(b(a)) # <- '1:1:1'
```

22. Для создания учетных записей пользователей с различными правами можно использовать следующие способы:

1. Использование стандартных учетных записей, которые предоставляются при установке операционной системы. Например, в Windows есть учетные записи Administrator и Guest, которые имеют различные уровни доступа.

2. Создание новых учетных записей вручную с помощью панели управления или

консоли управления Active Directory. Для этого необходимо указать имя пользователя, пароль и права доступа.

3. Использование программы для управления пользователями, такой как Active Directory Users & Computers в Windows или FreeIPA в Linux. Эти программы позволяют создавать и настраивать учетные записи пользователей, а также управлять их правами доступа.

4. Создание учетных записей при помощи локальных и групповых политик безопасности, в которых также можно полностью настроить права доступа для каждого пользователя.

23.

```
number: int = int(input('Введите целочисленное число: '))
```

```
def print_triangle(count: int, to_up = True) -> None:
```

```
    if to_up:
```

```
        array_numbers: list[int] = range(count + 1)
```

```
    else:
```

```
        array_numbers: list[int] = range(count + 1)[::-1]
```

```
    for number in array_numbers:
```

```
        print('* ' * number)
```

```
print_triangle(5, to_up=True)
```

```
print_triangle(5, to_up=False)
```

24.

```
a = [1, 0, True, False]
```

```
b = [1, 1, True, True]
```

```
def c(d):
```

```
    flag = d[0]
```

```
    for i in d:
```

```
        if flag != i:
```

```
            return False
```

```
    return True
```

```
print(c(a))
```

```
print(c(b))
```

25. В данном случае можно использовать два варианта: песочницы и виртуальные машины.

Песочница - это программа, которая изолирует приложения и процессы от основной операционной системы и ее ресурсов. Она позволяет запускать приложения в безопасной среде, где они не могут повлиять на работу других программ или системы в целом. Песочницы используются для тестирования приложений, защиты от вредоносных программ и предотвращения заражения компьютера вирусами.

Виртуальная машина (VM) - это программный эмулятор компьютерной системы, который создает виртуальную копию реальной компьютерной системы и позволяет запускать на ней несколько операционных систем одновременно. Виртуальные машины используются для разработки, тестирования и отладки программного обеспечения, а также

для обеспечения безопасности и изоляции приложений от друг друга.

26.

```
a = int(input()) # 5
def b(c, d = True):
    if d:
        e = range(c)
    else:
        e = range(c)[::-1]
    for i in e:
        print('* ' * i)
b(a, True)
b(a, False)
```

27.

1. Использовать резервные источники питания: например, аккумуляторы блоки бесперебойного питания.

2. Создать резервные копии данных: на внешних носителях или в облачных хранилищах.

3. Использовать программное обеспечение для резервного копирования: например, Acronis True Image или Backup & Restore.

4. Обучать сотрудников правилам работы с информацией: например, не сохранять важные данные на рабочем столе или в личных папках.

5. Следить за состоянием оборудования: регулярно проверять работоспособность источников питания и других устройств.

28. Для сборки ПК под разные задачи необходимо учитывать следующие факторы:

1. Процессор: для офисных задач подойдет процессор с низким тепловыделением и невысокой частотой, для игровых - с высокой частотой и хорошим графическим ядром.

2. Оперативная память: для офисных задач хватит 4-8 Гб, для игр - 8-16 Гб и более.

3. Жесткий диск: для хранения данных подойдет HDD, для быстрого доступа к данным - SSD.

4. Видеокарта: для офисной работы можно выбрать встроенную видеокарту, для игр и работы с графикой - дискретную.

5. Блок питания: для ПК с несколькими комплектующими нужен мощный блок питания.

6. Корпус: для компактности и удобства можно выбрать корпус с блоком питания в отдельном отсеке.

7. Операционная система: для офисного ПК можно установить Windows или Linux, для игрового - Windows или игровую версию Linux.

8. Цена: при выборе комплектующих необходимо учитывать бюджет и требования к производительности.

29. BSOD (Blue Screen of Death) — это ошибка, которая возникает в операционной системе Windows и приводит к полной неработоспособности компьютера.

BSOD может возникнуть по разным причинам, таких как:

- Проблемы с драйверами устройств.
- Неисправность оперативной памяти.
- Проблема с жестким диском.
- Перегрев компонентов компьютера.

- Вирусы и вредоносное ПО.
- Проблемы с BIOS.
- Проблемы с программным обеспечением.
- Аппаратные проблемы с материнской платой.
- Проблемы со сторонними устройствами (например, USB-устройствами).
- Проблемы с процессором.
- Проблемы с видеокартой.
- Проблемы с блоком питания.
- Проблемы с сетевой картой.

Чтобы решить эту проблему, необходимо провести диагностику компьютера и определить причину возникновения BSOD.

Для начала необходимо перезагрузить компьютер и попытаться войти в систему снова. Если это не помогает, то можно попытаться загрузить систему в безопасном режиме и попытаться выяснить причину, либо восстановить систему из последней точки восстановления.

30. Необходимо выполнить ряд задач, среди которых:

1. Изучение требований заказчика и составление технического задания.
2. Разработка проекта системы и согласование его с заказчиком.
3. Определение архитектуры проекта.
4. Разработка прототипа системы.
5. Тестирование системы и исправление ошибок.
6. Разработка плана внедрения, подготовка к внедрению и внедрение.
7. Запуск системы в эксплуатацию.
8. Обеспечение технической поддержки и разработка плана сопровождения.
9. Сопровождение системы.

Оценка запланированных результатов по учебной дисциплине (предмету)

Результаты обучения (элементы)	Показатели оценки результата
ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	иметь практический опыт в настройке отдельных компонентов программного обеспечения компьютерных систем; в выполнении отдельных видов работ на этапе поддержки программного обеспечения компьютерной системы; уметь: подбирать и настраивать конфигурацию программного обеспечения компьютерных систем; использовать методы защиты программного обеспечения компьютерных систем; проводить инсталляцию программного обеспечения компьютерных систем; производить настройку отдельных компонентов программного обеспечения компьютерных систем;
ОК 2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;	
ОК 4. Эффективно взаимодействовать и работать в коллективе и команде;	
ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	
ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языках.	
ПК 4.1. Осуществлять инсталляцию,	компьютерных систем;

<p>настройку и обслуживание программного обеспечения компьютерных систем</p> <p>ПК 4.2. Осуществлять измерения эксплуатационных характеристик программного обеспечения компьютерных систем.</p> <p>ПК 4.3. Выполнять работы по модификации отдельных компонент программного обеспечения в соответствии с потребностями заказчика.</p> <p>ПК 4.4. Обеспечивать защиту программного обеспечения компьютерных систем программными средствами.</p>	<p>анализировать риски и характеристики качества программного обеспечения</p> <p>знать:</p> <p>основные методы и средства эффективного анализа функционирования программного обеспечения;</p> <p>основные виды работ на этапе сопровождения программного обеспечения;</p> <p>основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения;</p> <p>средства защиты программного обеспечения в компьютерных системах.</p>
--	--

Образец экзаменационного билета
**МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ**

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

<p>Утверждаю Заместитель директора <u>Борзенкова Е.К.</u> (Ф.И.О.)</p> <p>_____</p> <p style="text-align: center;">(подпись)</p>	<p>Экзаменационный билет №1 по ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем</p>	<p>Рассмотрено на заседании цикловой методической комиссии Председатель <u>Гаврикова</u> <u>Е. Ю.</u> (Ф.И.О.)</p>
<p>«___» _____ 2023 г.</p>	<p>Группа (ы) <u>221</u></p> <p>Специальность: <u>09.02.07 Информационные</u> <u>системы и программирование</u></p>	<p>_____</p> <p style="text-align: center;">(подпись)</p> <p>«___» _____ 2023 г.</p>

1. ГОСТ Р ИСО/МЭК 12207. О чем этот стандарт? Назовите основные группы процессов, согласно стандарту.
2. У вас на работе не запускается ПК. Какие действия вы будете предпринимать?

Критерии оценки ответов обучающихся

Отметка 5 «отлично» - продемонстрирован высокий уровень знаний по теоретическому вопросу, тематика вопроса полностью раскрыта. Практико-ориентированное задание выполнено верно.

Отметка 4 «хорошо» - продемонстрировано понимание и знание основного содержания теоретического вопроса билета, однако допущены недочеты в определениях терминов и понятий. Практико-ориентированное задание выполнено с замечаниями.

Отметка 3 «удовлетворительно» - продемонстрировано слабое владение основным содержанием по теоретическому вопросу билета, допущены неточности в определениях терминов и понятий. Практико-ориентированное задание выполнено с ошибками.

Отметка 2 «неудовлетворительно» - не продемонстрировано владение знаниями и умениями, тема теоретического вопроса билета не раскрыта. Практико-ориентированное задание не выполнено.

4.Комплект билетов – 30 шт.

5.Оценочная ведомость по профессиональному модулю.

6.Сводный экзаменационный протокол на группу студентов по экзамену по профессиональному модулю.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

**краевое государственное автономное
профессиональное образовательное учреждение
«Дальнегорский индустриально-технологический колледж»**

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
МЕЖДИСЦИПЛИНАРНОГО КУРСА**

МДК 11.01 Технология разработки и защиты баз данных

подготовки специалистов среднего звена по специальности

код специальности 09.02.07 «Информационные системы и программирование»

Дальнегорск, 2022 год

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности СПО 09.02.07 Информационные системы и программирование МДК 11.01 «Технология разработки и защиты баз данных».

Разработчики:

Организация-разработчик: КГА ПОУ «ДИТК»

Разработчик: Адаменко Ольга Петровна, преподаватель.

ОДОБРЕН
цикловой методической комиссией
Протокол № 3
от «03» сентября 2022 г.
Председатель Гаврикова Е. Ю.

СОДЕРЖАНИЕ

1. Паспорт комплекта контрольно-оценочных средств
2. Результаты освоения МДК
3. Оценка освоения МДК
 - 3.1. Контроль и оценка освоения
 - 3.1.1 Методы и критерии оценивания
4. Контрольно-оценочные средства для проведения промежуточной аттестации
 - 4.1. Пакет экзаменатора
 - 4.2. Критерии оценки

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения МДК 11.01 Технология разработки и защиты баз данных обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 «Информационные системы и программирование» следующими умениями, знаниями, которые формируют профессиональные компетенции, и общими компетенциями.

Обучающийся должен знать:

31 основные положения теории баз данных, хранилищ данных, баз знаний;

32 основные принципы структуризации и нормализации базы данных;

33 основные принципы построения концептуальной, логической и физической модели данных;

34 методы описания схем баз данных в современных системах управления базами данных;

35 структуры данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;

36 методы организации целостности данных;

37 способы контроля доступа к данным и управления привилегиями;

38 основные методы и средства защиты данных в базах данных.

Обучающийся должен уметь:

У1 работать с современными case-средствами проектирования баз данных;

У2 проектировать логическую и физическую схемы базы данных;

У3 создавать хранимые процедуры и триггеры на базах данных;

У4 применять стандартные методы для защиты объектов базы данных;

У5 выполнять стандартные процедуры резервного копирования и мониторинга выполнения этой процедуры;

У6 выполнять процедуру восстановления базы данных и вести мониторинг выполнения этой процедуры;

У7 обеспечивать информационную безопасность на уровне базы данных.

Обучающийся должен иметь практический опыт:

- работы с объектами базы данных в конкретной системе управления базами данных;
- использования стандартных методов защиты объектов базы данных;
- работы с документами отраслевой направленности

Личностные результаты с учетом особенностей учебной дисциплины:

ЛР1	- осознающий себя частью народа, гражданином России
ЛР2	- принимающий принципы демократического общества и следующий им, готовый защищать Родину, занимающий активную гражданскую позицию избирателя, волонтера, общественного деятеля
ЛР 3	- проявляющий интерес к изучению и освоению культурных традиций России, русского и родного языка; уважающий различные взгляды и вероисповедания, не нарушающие права и свободы других людей, заботящийся о тех, кто нуждается в помощи, в том числе через уплату установленных налогов, уважающий личность другого человека, готовый к рефлексии своих действий, в т.ч. высказываний, и оценке их влияния на других людей, признающий ценность жизни и уважение личности другого человека, его прав и свобод, не ущемляющих права и свободы других людей

ЛР 4	- заботящийся о сохранении исторического и культурного наследия России, принимающий и сохраняющий традиционные семейные ценности своего народа
ЛР 5	- демонстрирующий свободу выбора, самостоятельность и ответственность в принятии решений, стремление к саморазвитию и самосовершенствованию во всех сферах жизни
ЛР 6	- критически мыслящий, интеллектуально самостоятельный; стремящийся к саморазвитию и самосовершенствованию во всех сферах жизни; пользующийся свободой выбора и самостоятельный в принятии решений; готовый к рефлексии своих действий, в т.ч. высказываний, и оценке их влияния на других людей
ЛР 7	- обладающий проектным мышлением, командным духом, способный быть лидером, демонстрирующий готовность к продуктивному взаимодействию и сотрудничеству; демонстрирующий активную гражданскую позицию, в том числе в социальной и трудовой деятельности, мотивированный к познанию и личностному развитию
ЛР 8	- демонстрирующий самоуважение и уважение к другим людям, их правам и свободам; готовый к рефлексии своих действий, в т.ч. высказываний, и оценке их влияния на других людей
ЛР 9	- демонстрирующий развитое правосознание и законопослушность
ЛР 10	- присвоивший ценности, установки, отношения, личностные качества гражданина, необходимые для реализации его собственных прав и свобод, а также прав и свобод других граждан России; участвующий в реализации просветительских программ, поисковых, археологических, исторических, краеведческих отрядах и молодежных объединениях; осознающий важность сохранения и укрепления здоровья, имеющий внутреннюю установку на активное здоровье сбережение и культуры
ЛР 11	- уважающий различные взгляды и вероисповедания, не нарушающие права и свободы других людей
ЛР 12	- самоуважение и уважение к другим людям, их правам и свободам
ЛР 13	- проявляющий ответственность за результат учебной деятельности и профессиональной деятельности, трудолюбивый, упорный и настойчивый в достижении цели, осознающий ценность образования
ЛР 14	- стремящийся к саморазвитию и самосовершенствованию во всех сферах жизни и деятельности, готовый учиться на протяжении жизни; критически мыслящий, интеллектуально самостоятельный, мотивированный к познанию, конструктивно взаимодействующий в коллективе
ЛР 15	- участвующий в исследовательской и научной работе; трудолюбивый, упорный и настойчивый в достижении цели; осознающий ценность образования
ЛР 15	- имеющий положительную динамику в организации собственной учебной деятельности по результатам самооценки, самоанализа и коррекции ее результата, стремящийся к саморазвитию и

	самосовершенствованию во всех сферах жизни и деятельности, готовый учиться на протяжении жизни
ЛР 16	- имеющий положительную динамику в организации собственной учебной деятельности по результатам самооценки, самоанализа и коррекции ее результата, стремящийся к саморазвитию и самосовершенствованию во всех сферах жизни и деятельности, готовый учиться на протяжении жизни
ЛР 17	- участвующий в конкурсах профессионального мастерства и в командных проектах; демонстрирующий свободу и ответственность выбора и принятия решений; критически мыслящий, интеллектуально самостоятельный, мотивированный к познанию
ЛР 18	- демонстрирующий навыки межличностного делового общения, социального имиджа, соблюдающий социальные нормы и правила внутреннего распорядка колледжа и предприятия
ЛР 19	- демонстрирующий интерес к будущей профессии, трудолюбивый, упорный и настойчивый в достижении цели, осознающий ценность образования

Формируемые ОК:

- ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
- ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.
- ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.
- ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.
- ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.
- ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе общечеловеческих ценностей.
- ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.
- ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.
- ОК 9. Использовать информационные технологии в профессиональной деятельности.
- ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.
- ОК 11. Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере.

Формируемые ПК:

- ВД 11 «Разработка, администрирование и защита баз данных»
- ПК 11.1 Осуществлять сбор, обработку и анализ информации для проектирования баз данных
- ПК 11.2 Проектировать базу данных на основе анализа предметной области
- ПК 11.3 Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области

ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных

ПК 11.5 Администрировать базы данных

ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации

Формами промежуточной аттестации по МДК являются: 5 семестр – проверочная работа, 6 семестр – экзамен.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по МДК осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих и профессиональных компетенций:

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
У1 работать с современными case-средствами проектирования баз данных;	– работает с изобразительными средствами, используемыми в ER- моделировании
У2 проектировать логическую и физическую схемы базы данных;	реляционной базы данных;
ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	– демонстрирует способности решения профессиональных задач практического содержания;
ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.	– проектирует логическую и физическую схемы базы данных;
ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.	– осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;
ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.	– планирует и реализовывает собственное профессиональное и личностное развитие;
ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	– умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами;
ОК 9. Использовать информационные технологии в профессиональной деятельности.	– осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста;
ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.	– использует информационные технологии в профессиональной деятельности;
ПК 11.1 Осуществлять сбор, обработку и анализ информации для проектирования баз данных	– пользуется профессиональной документацией на государственном и иностранном языке;
ПК 11.2 Проектировать базу данных на основе анализа предметной области	– выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам;
	– использует современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

<p>ПК 11.3 Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области</p>	<ul style="list-style-type: none"> – осуществляет сбор, обработку и анализ информации для проектирования баз данных. – проектирует базу данных на основе анализа предметной области; – разрабатывает объекты базы данных в соответствии с результатами анализа предметной области; – реализует базу данных в конкретной системе управления базами данных.
<p>У3 создавать хранимые процедуры и триггеры на базах данных; ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие. ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. ОК 9. Использовать информационные технологии в профессиональной деятельности. ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных ПК 11.5 Администрировать базы данных</p>	<ul style="list-style-type: none"> – умеет создавать хранимые процедуры и триггеры на базах данных; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке; – разрабатывает объекты базы данных в соответствии с результатами анализа предметной области; – реализует базу данных в конкретной системе управления базами данных; – выполняет задачи по администрированию базы данных.
<p>У4 применять стандартные методы для защиты объектов базы данных; ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p>	<ul style="list-style-type: none"> – применяет стандартные методы для защиты объектов базы данных; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам;

<p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p> <p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p> <p>ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных</p> <p>ПК 11.5 Администрировать базы данных</p> <p>ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации</p>	<ul style="list-style-type: none"> – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке; – разрабатывает объекты базы данных в соответствии с результатами анализа предметной области; – реализует базу данных в конкретной системе управления базами данных; – выполняет задачи по администрированию базы данных; – обеспечивает защиту информации в базе данных с использованием различных технологий защиты информации.
<p>У5 выполнять стандартные процедуры резервного копирования и мониторинга выполнения этой процедуры;</p> <p>У6 выполнять процедуру восстановления базы данных и вести мониторинг выполнения этой процедуры;</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p>	<ul style="list-style-type: none"> – выполняет стандартные процедуры резервного копирования и восстановления базы данных, ведет мониторинг выполнения этих процедур; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами;

<p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p> <p>ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных</p> <p>ПК 11.5 Администрировать базы данных</p> <p>ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации</p>	<ul style="list-style-type: none"> – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке; – реализует базу данных в конкретной системе управления базами данных; – выполняет задачи по администрированию базы данных; – обеспечивает защиту информации в базе данных с использованием различных технологий защиты информации.
<p>У7 обеспечивать информационную безопасность на уровне базы данных.</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p> <p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p>	<ul style="list-style-type: none"> – обеспечивает информационную безопасность на уровне базы данных; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке;

ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации	– обеспечивает защиту информации в базе данных с использованием различных технологий защиты информации.
Знать:	
31 основные положения теории баз данных, хранилищ данных, баз знаний;	– знает основные положения теории баз данных, хранилищ данных, баз знаний;
32 основные принципы структуризации и нормализации базы данных;	– знает основные принципы структуризации и нормализации базы данных;
33 основные принципы построения концептуальной, логической и физической модели данных;	– понимает основные принципы построения концептуальной, логической и физической модели данных;
34 методы описания схем баз данных в современных системах управления базами данных;	– освоил методы описания схем баз данных в современных системах управления базами данных;
35 структуры данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;	– знает структуры данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
36 методы организации целостности данных;	– знает методы организации целостности данных;
37 способы контроля доступа к данным и управления привилегиями;	– обеспечивает способы контроля доступа к данным и управления привилегиями;
38 основные методы и средства защиты данных в базах данных.	– имеет представление об основных методах и средствах защиты данных в базах данных.

3.ОЦЕНКА ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Контроль и оценка освоения МДК по темам (разделам)

Элемент МДК	Формы и методы контроля			
	Текущий контроль		Промежуточная аттестация	
	Осваиваемые результаты	Метод контроля	Проверяемые результаты	Форма контроля
Раздел 1. Технология проектирования баз данных				
Тема 1.1 Основы хранения и обработки данных. Проектирование БД.	У1, У2, У3 31-36, ОК 1-ОК 5, ОК 9-ОК10 ПК 11.1- ПК 11.4	Устный опрос, тестовое задание, практическая работа, самостоятельная работа	У1, У2, У3 31-34, ОК 1-ОК 5, ОК 9-ОК10 ПК 11.1- ПК 11.3	5 семестр – зачет
Раздел 2. Технология разработки и администрирования баз данных				

Тема 2.1 Разработка и администрирование БД	У1-У6, 31-37, ОК 1-ОК 5, ОК 9-ОК10 ПК 11.1- ПК 11.5	Устный опрос, тестовое задание, практическая работа, самостоятельн ая работа		
Раздел 3. Технология защиты баз данных				
Тема 3.1 Способы и методы защиты БД	У1-У7, 31-38, ОК 1-ОК 5, ОК 9-ОК10 ПК 11.1- ПК 11.6	Устный опрос, тестовое задание, практическая работа, самостоятельн ая работа	У1-У7, 31-38, ОК 1-ОК 5, ОК 9-ОК10 ПК 11.1- ПК 11.6	6 семестр – экзамен

3.1.1. Методы и критерии оценивания

1. Устный опрос. Критерии оценивания.

Оценка 5 «отлично» - ответил на вопросы в объеме лекционного и дополнительного материала, дал полные грамотные ответы на все дополнительные вопросы.

Оценка 4 «хорошо» - грамотно изложил ответы на вопросы, но содержание и формулировки имеют отдельные неточности (допускается нечеткая формулировка определений), в полной мере ответил на заданные дополнительные вопросы.

Оценка 3 «удовлетворительно» - ответил на часть вопросов в объеме лекционного материала и ответил на часть дополнительных вопросов.

Оценка 2 «неудовлетворительно» - допустил ошибки в определении базовых понятий, исказил их смысл, не ответил на дополнительные вопросы.

2. Тестовое задание. Критерии оценивания.

Оценка 5 «отлично» - правильно выполнено 85 – 100 % заданий.

Оценка 4 «хорошо» - правильно выполнено 70 – 84 % заданий.

Оценка 3 «удовлетворительно» правильно выполнено 55 – 69 % заданий.

Оценка 2 «неудовлетворительно» - правильно выполнено 1 – 54 % заданий.

3. Самостоятельная работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме; учтены все требования к данной работе; самостоятельно поставлены цели и задачи работы, соответствующие заданной теме/проблеме; получены результаты в соответствии с поставленной целью; работа оформлена аккуратно и грамотно.

Оценка 4 «хорошо» - выполнены требования к отметке «5», но были допущены два-три недочета; не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью и объем выполненной части работы позволяет получить недостаточно результатов в соответствии с поставленной целью.

4. Практическая работа. Критерии оценивания.

Оценка 5 «отлично» - работа выполнена в полном объеме, с соблюдением алгоритма выполнения: последовательности: 1) правильно и аккуратно выполнены все записи, таблицы, рисунки, чертежи, графики; 2) правильно проведена демонстрация. Получены результаты в соответствии с поставленной целью.

Оценка 4 «хорошо» - выполнены требования к отметке «5», но были допущены два-три недочета или не более одной негрубой ошибки и одного недочета.

Оценка 3 «удовлетворительно» - работа выполнена не в полном объеме, но объем выполненной части работы позволяет получить часть результатов в соответствии с поставленной целью.

Оценка 2 «неудовлетворительно» - работа выполнена не полностью и объем выполненной части работы не позволяет получить никаких результатов в соответствии с поставленной целью.

4. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Проверочная работа (5 семестр)

1. Форма проведения: тестирование, решение практико-ориентированной задачи.

2. Условия выполнения

Время выполнения задания: 90 мин

Оборудование учебного кабинета: рабочие места обучающихся (16 ед.); рабочее место преподавателя; учебная доска; рабочая программа учебной дисциплины «Основы проектирования баз данных», методические указания по выполнению заданий дифференцированного зачета.

Технические средства обучения: компьютеры с лицензионным программным обеспечением; мультимедийный комплекс; электронные учебники. Компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

Программное обеспечение: Windows, Microsoft Office; Антивирусная защита: Kaspersky Free.

Информационные источники: программное обеспечение общего и профессионального назначения; комплект учебно-наглядных пособий; тестовый материал, карточки индивидуальных заданий; методические рекомендации для проведения лабораторных занятий.

Требования охраны труда: ИОТ-11 при работе в кабинете информатики, ИОТ-64 для пользователей персональных компьютеров, ТБ-4 при работе на видеодисплейных терминалах и ЭВМ.

3. Пакет материалов для проведения зачёта

3.1. Перечень тем, контролируемых в ходе промежуточной аттестации.

Тема 1.1 Основы хранения и обработки данных. Проектирование БД.

3.2. Практико-ориентированные задания.

Оценка запланированных результатов по МДК

Результаты обучения (элементы)	Показатели оценки результата
У1 работать с современными case-средствами проектирования баз данных.	– работает с изобразительными средствами, используемыми в ER- моделировании реляционной базы данных;
У2 проектировать логическую и физическую схемы базы данных.	– демонстрирует способности решения профессиональных задач практического содержания;
У3 создавать хранимые процедуры и триггеры на базах данных.	– проектирует логическую и физическую схемы базы данных;

<p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p> <p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p> <p>ПК 11.1 Осуществлять сбор, обработку и анализ информации для проектирования баз данных.</p> <p>ПК 11.2 Проектировать базу данных на основе анализа предметной области.</p> <p>ПК 11.3 Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области.</p>	<ul style="list-style-type: none"> – создает хранимые процедуры и триггеры на базах данных; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – использует современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; – осуществляет сбор, обработку и анализ информации для проектирования баз данных. – проектирует базу данных на основе анализа предметной области. – разрабатывает объекты базы данных в соответствии с результатами анализа предметной области. реализует базу данных в конкретной системе управления базами данных.
<p>31 основные положения теории баз данных, хранилищ данных, баз знаний;</p>	<ul style="list-style-type: none"> – знает основные положения теории баз данных, хранилищ данных, баз знаний;
<p>32 основные принципы структуризации и нормализации базы данных;</p>	<ul style="list-style-type: none"> – знает основные принципы структуризации и нормализации базы данных;
<p>33 основные принципы построения концептуальной, логической и физической</p>	<ul style="list-style-type: none"> – понимает основные принципы построения концептуальной, логической и физической

модели данных;	модели данных;
34 методы описания схем баз данных в современных системах управления базами данных;	– освоил методы описания схем баз данных в современных системах управления базами данных;
35 структуры данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;	– знает структуры данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
36 методы организации целостности данных;	– знает методы организации целостности данных;

Примерный КИМ по другим формам контроля

- I. Ответить на контрольные вопросы.
1. Что такое ER-модель?
 - a) Модель проектирования баз данных, основанная на сущностях и отношениях
 - b) Модель, описывающая таблицы и их структуру
 - c) Модель, описывающая процессы в системе
 - d) Модель, описывающая пользовательский интерфейс
 2. Что такое первичный ключ в таблице базы данных?
 - a) Уникальный идентификатор записи в таблице
 - b) Связь между двумя таблицами
 - c) Сумма всех значений в столбце
 - d) Значение, описывающее тип данных
 3. Что такое внешний ключ в базе данных?
 - a) Ссылка на первичный ключ в другой таблице
 - b) Столбец, содержащий числовые значения
 - c) Описание структуры базы данных
 - d) Объект, хранящий данные таблицы
 4. Какие виды связей между таблицами существуют?
 - a) Одномерные, двумерные, многомерные
 - b) Один к одному, один ко многим, многое ко многим
 - c) Первичные, внешние, альтернативные
 - d) Горизонтальные, вертикальные, диагональные
 5. Что такое нормализация базы данных?
 - a) Процесс устранения избыточности и повышения целостности данных
 - b) Процесс добавления новых данных в таблицу
 - c) Процесс удаления данных из таблицы
 - d) Процесс переименования таблицы
 6. Что такое индекс в базе данных?
 - a) Структура данных, ускоряющая поиск и сортировку данных
 - b) Ключевое поле в таблице
 - c) Столбец с уникальными значениями
 - d) Связь между двумя таблицами
 7. Какие языки запросов используются для работы с базами данных?
 - a) SQL, HTML, CSS

- b) C++, Java, Python
 - c) SQL, PL/SQL, T-SQL
 - d) PHP, JavaScript, Ruby
8. Что такое транзакция в базе данных?
- a) Одиночный запрос к базе данных
 - b) Связь между двумя таблицами
 - c) Логическая единица работы с базой данных, состоящая из одного или нескольких запросов
 - d) Описание структуры таблицы
9. Что такое агрегатные функции в SQL?
- a) Функции для соединения таблиц
 - b) Функции для работы с текстом
 - c) Функции для работы с датами и временем
 - d) Функции для работы с наборами данных (сумма, среднее значение, максимум и др.)
10. Какие виды схем баз данных существуют?
- a) Концептуальная, логическая, физическая
 - b) Плоская, иерархическая, древовидная
 - c) Первичная, внешняя, альтернативная
 - d) Вертикальная, горизонтальная, диагональная

II. Практико-ориентированная задача «Проектирование концептуальной модели базы данных»

Цель: демонстрация практических навыков анализа предметной области, информационных задач и проектирования концептуальной модели базы данных.

Пошаговое проектирование концептуальной модели.

1. Выделить основные абстракции (сущность, атрибут, связь) в предметной области и определить их параметры.
2. Сформировать максимально полный перечень возможных запросов к базе данных на основе анализа предметной области.
3. Построить концептуальную модель.
4. Описать домены (допустимые множества значений, которые могут принимать атрибуты), указывая типы соответствующих данных и их характеристики.

Задание

1. Спроектировать концептуальную модель базы данных страховой компании.
2. Построить ER-диаграмму базы данных страховой компании.
3. Выполнить реализацию поставленной задачи в системе «1С: Предприятие 8.3».
4. Результатом работы является
 - ER-диаграмма базы данных страховой компании;
 - файл выгрузки информационной базы страховой компании.

Описание предметной области

Вы работаете в страховой компании. Вашей задачей является отслеживание ее финансовой деятельности. Компания имеет различные филиалы по всей стране. Каждый филиал характеризуется названием, адресом и телефоном. Деятельность компании организована следующим образом: к вам обращаются различные лица с целью заключения договора о страховании. В зависимости от принимаемых на страхование объектов и страхуемых рисков

договор заключается по определенному виду страхования (например, страхование автотранспорта от угона, страхование домашнего имущества, добровольное медицинское страхование). При заключении договора вы фиксируете дату заключения, страховую сумму, вид страхования, тарифную ставку и филиал, в котором заключался договор.

Контрольные вопросы

1. Дать определение понятию концептуальная модель.
2. Какие действия включает первый шаг построения концептуальной модели?
3. Какие действия включает второй шаг построения концептуальной модели?
4. Какие действия включает третий шаг построения концептуальной модели?
5. Какие действия включает четвёртый шаг построения концептуальной модели?

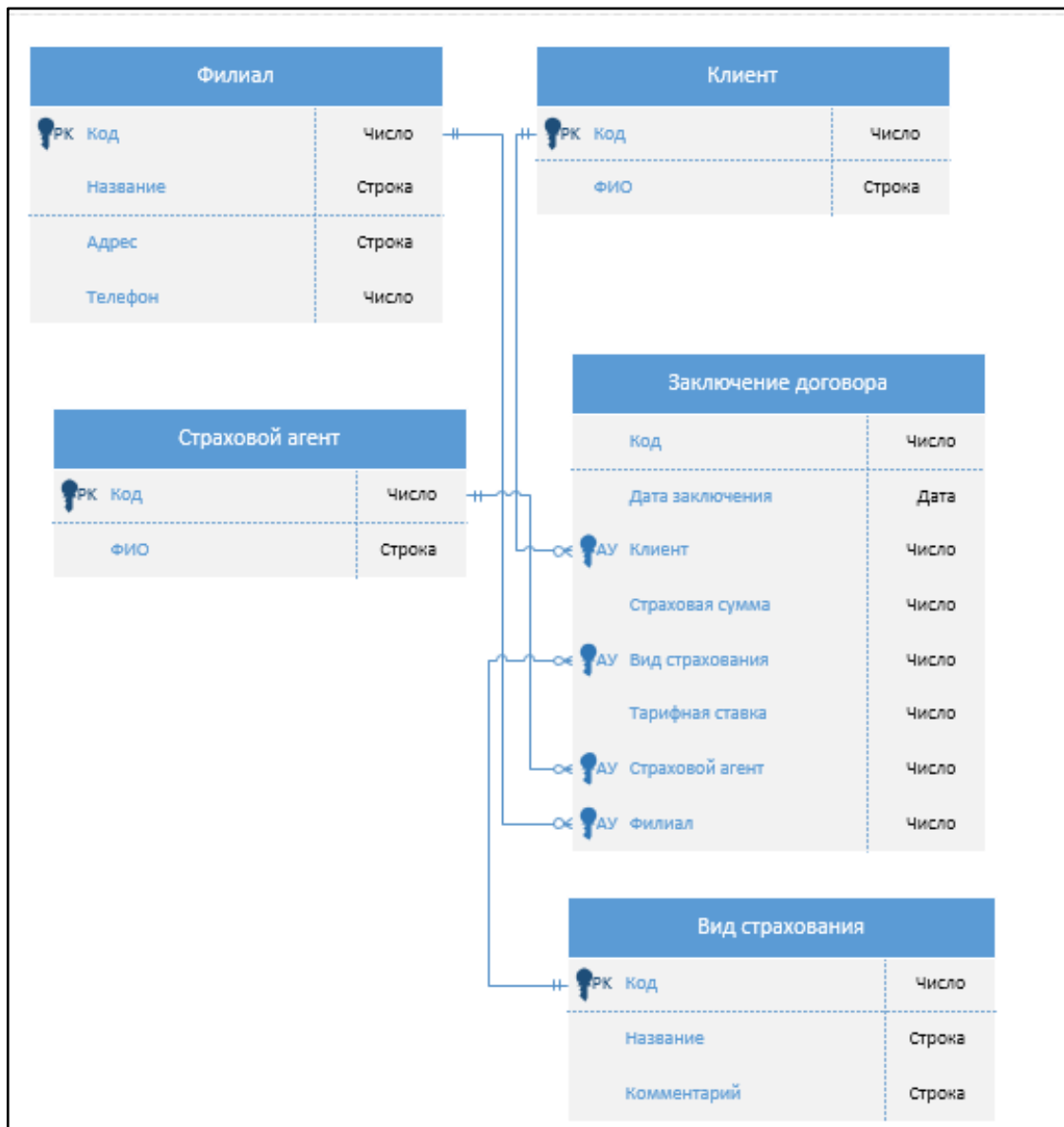
4.Эталоны ответов

- I. Ответить на контрольные вопросы.
1a; 2.a; 3.a; 4.b; 5.a; 6.a; 7.c; 8.c; 9.d; 10.a
- II. Практико-ориентированная задача «Проектирование концептуальной модели базы данных»

Ответы на контрольные вопросы

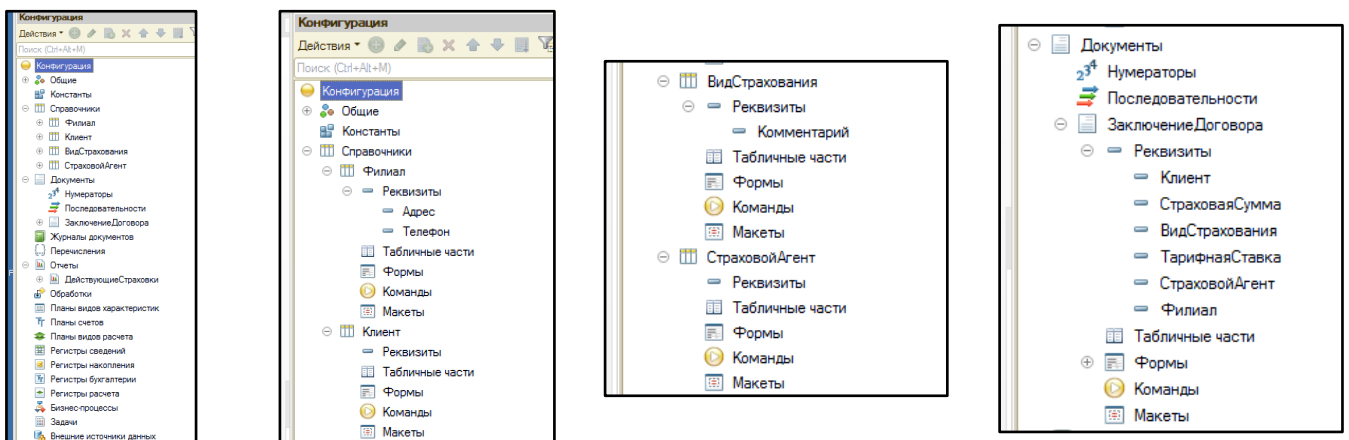
1. Концептуальная (содержательная) модель – это абстрактная модель, определяющая структуру моделируемой системы, свойства её элементов и причинно-следственные связи, присущие системе и существенные для достижения цели моделирования.
2. Выделить основные абстракции (сущность, атрибут, связь) в предметной области и определить их параметры.
3. Сформировать максимально полный перечень возможных запросов к базе данных на основе анализа предметной области.
4. Построить концептуальную модель.
5. Описать домены (допустимые множества значений, которые могут принимать атрибуты), указывая типы соответствующих данных и их характеристики.

ER-диаграмма базы данных страховой компании



Реализация базы данных «Страховая компания» в системе «1С:Предприятие 8.3»

1. Дерево объектов конфигурации базы данных «Страховая компания».



2. Отчет «Действующие страховки», построенный с использованием Схемы компоновки данных (набор данных).

Отчет ДействующиеСтраховки: ОсновнаяСхемаКомпоновкиДанных

Наборы данных | Связи наборов данных | Вычисляемые поля | Ресурсы | Параметры | Макеты | Вложенные схемы | Настройки

Поля:

Наборы данных	Поле	Путь	Ограничение поля				Роль
			Поле	Усло...	Груп...	Упо...	
			Ограничение реквизитов				
		Автозаголовок	Поле	Усло...	Груп...	Упо...	
Клиент	Клиент	Клиент	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/> Клиент		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
ПометкаУдаления	ПометкаУдаления	ПометкаУдаления	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/> Пометка удаления		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Номер	Номер	Номер	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/> Номер		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Проведен	Проведен	Проведен	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/> Проведен		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Дата	Дата	Дата	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/> Дата		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
СтраховаяСумма	СтраховаяСумма	СтраховаяСумма	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/> Страховая сумма		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
ТарифнаяСтавка	ТарифнаяСтавка	ТарифнаяСтавка	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/> Тарифная ставка		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
СтраховойАгент	СтраховойАгент	СтраховойАгент	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/> Страховой агент		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Филиал	Филиал	Филиал	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/> Филиал		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
МоментВремени	МоментВремени	МоментВремени	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	<input type="checkbox"/> Момент времени		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

3. Отчет «Действующие страховки», построенный с использованием *Схемы компоновки данных* (конструктор запроса).

Запрос:

```

ВЫБРАТЬ
    ЗаключениеДоговора.Ссылка КАК Ссылка,
    ЗаключениеДоговора.ВерсияДанных КАК ВерсияДанных,
    ЗаключениеДоговора.ПометкаУдаления КАК ПометкаУдаления,
    ЗаключениеДоговора.Номер КАК Номер,
    ЗаключениеДоговора.Дата КАК Дата,
    ЗаключениеДоговора.Проведен КАК Проведен,
    ЗаключениеДоговора.Клиент КАК Клиент,
    ЗаключениеДоговора.СтраховаяСумма КАК СтраховаяСумма,
    ЗаключениеДоговора.ВидСтрахования КАК ВидСтрахования,
    ЗаключениеДоговора.ТарифнаяСтавка КАК ТарифнаяСтавка,
    ЗаключениеДоговора.СтраховойАгент КАК СтраховойАгент,
    ЗаключениеДоговора.филиал КАК Филиал,
    ЗаключениеДоговора.Представление КАК Представление,
    ЗаключениеДоговора.МоментВремени КАК МоментВремени

ИЗ
    Документ.ЗаключениеДоговора КАК ЗаключениеДоговора
  
```

Автозаполнение Использовать группировки запроса если возможно

4. Отчет «Действующие страховки», построенный с использованием *Схемы компоновки данных* (настройки отчета).

Отчет ДействующиеСтраховки: ОсновнаяСхемаКомпоновкиДанных

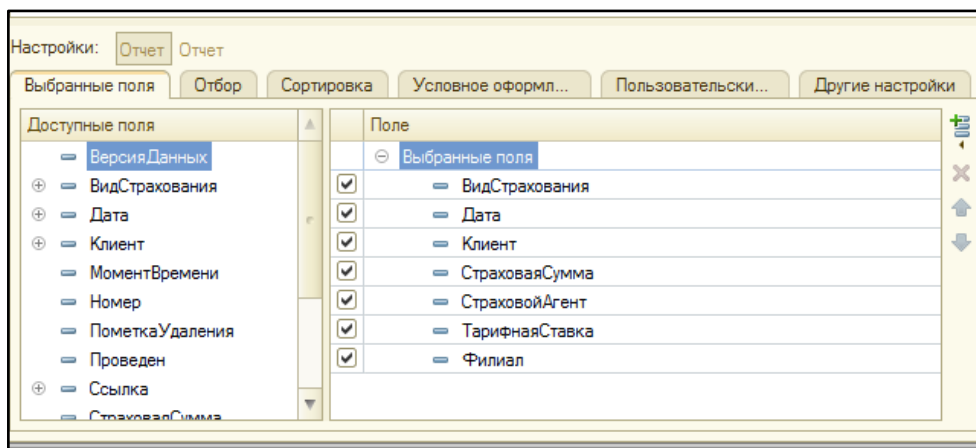
Наборы данных | Связи наборов данных | Вычисляемые поля | Ресурсы | Параметры | Макеты | Вложенные схемы | Настройки

Имя варианта | Представлен...

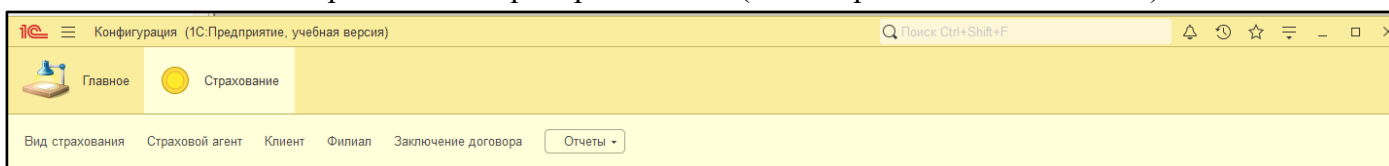
Основной | Основной

Отчет

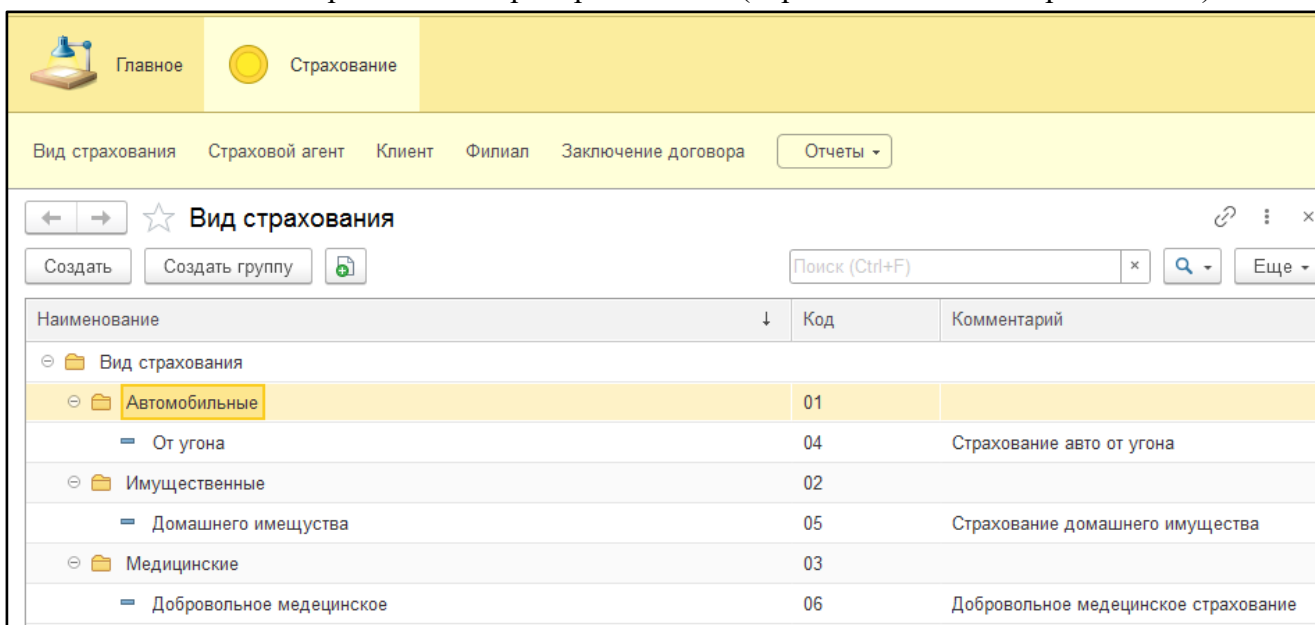
<Детальные записи>



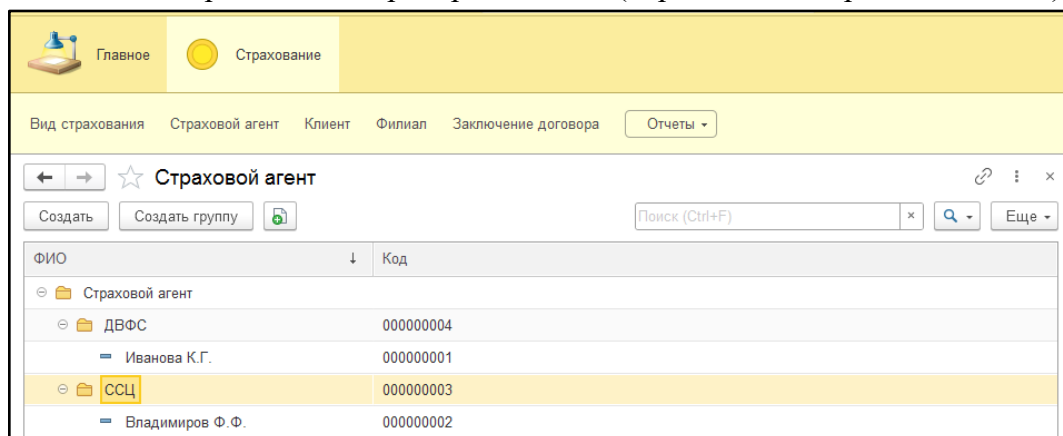
5. Пользовательский режим «1С:Предприятие 8.3» (панели разделов и навигации)



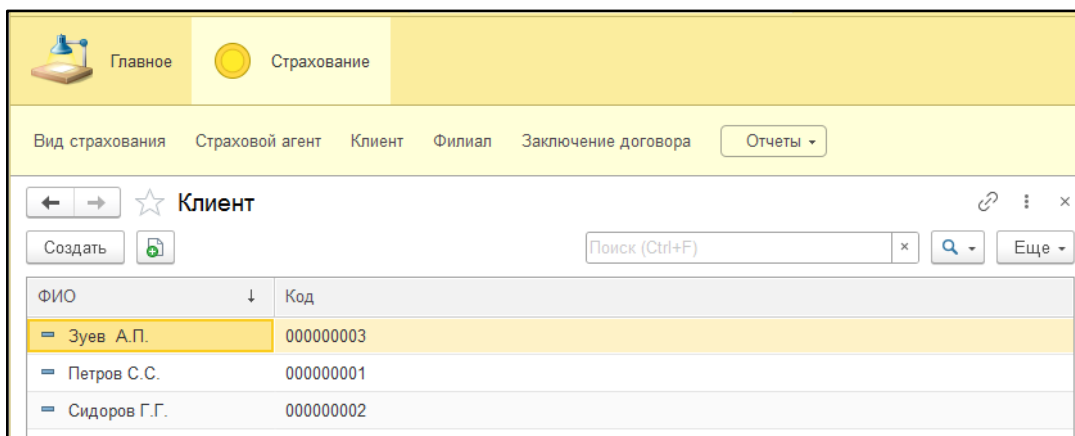
6. Пользовательский режим «1С:Предприятие 8.3» (справочник «Виды страхования»)



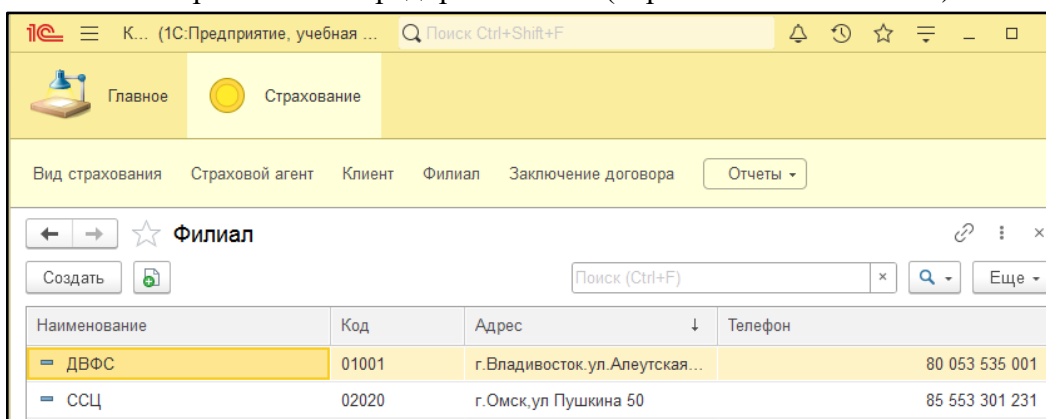
7. Пользовательский режим «1С:Предприятие 8.3» (справочник «Страховой агент»)



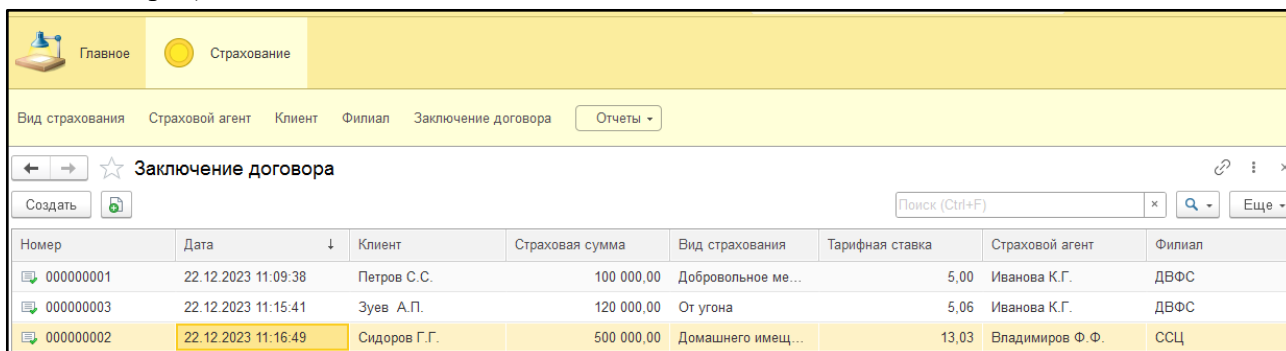
8. Пользовательский режим «1С:Предприятие 8.3» (справочник «Клиенты»)



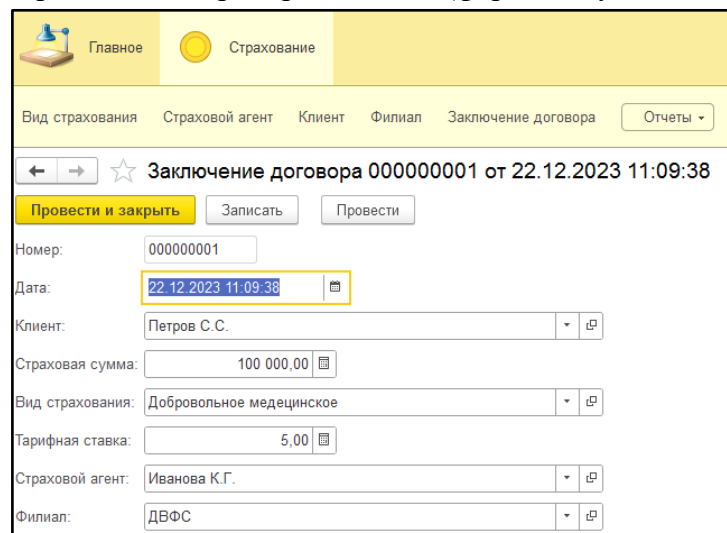
9. Пользовательский режим «1С:Предприятие 8.3» (справочник «Филиалы»)



10. Пользовательский режим «1С:Предприятие 8.3» (список документов «Заключение договора»)



11. Пользовательский режим «1С:Предприятие 8.3» (форма документа «Заключение договора»)



12. Пользовательский режим «1С: Предприятие 8.3» (отчет «Действующие страховки»)

Вид страхования	Дата	Клиент	Страховая сумма	Страховой агент	Тарифная ставка	Филиал
Добровольное медицинское	22.12.2023 11:09:38	Петров С.С.	100 000,00	Иванова К.Г.	5,00	ДВФС
Домашнего имущества	22.12.2023 11:16:49	Сидоров Г.Г.	500 000,00	Владимиров Ф.Ф.	13,03	ССЦ
От угона	22.12.2023 11:15:41	Зуев А.П.	120 000,00	Иванова К.Г.	5,06	ДВФС

Критерии оценивания

Оценка по традиционной системе	Процент	Критерий
Отлично	от 85 %	Продемонстрирован высокий уровень знаний по всему объему изучаемого учебного курса; в полном объеме выполнены практико-ориентированные задания.
Хорошо	от 70% до 85 %	Продемонстрировано понимание основного содержания изучаемого междисциплинарного курса; выполнены практико-ориентированные задания.
Удовлетворительно	от 55 % до 70 %	Продемонстрировано неполное понимание основного содержания изучаемого междисциплинарного курса; частично выполнены практико-ориентированные задания.
Неудовлетворительно	менее 55 %	Не продемонстрировано владение знаниями; не выполнены практико-ориентированные задания.

2. Экзамен (6 семестр)

1.Форма проведения: тестирование, решение практико-ориентированных задач.

2.Условия выполнения

Время выполнения задания: 90 мин

Оборудование учебного кабинета: рабочие места обучающихся (16 ед.); рабочее место преподавателя; учебная доска; рабочая программа МДК.11.01 «Разработка и защита баз данных», методические указания по выполнению заданий дифференцированного зачета.

Технические средства обучения: компьютеры с лицензионным программным обеспечением; мультимедийный комплекс; электронные учебники. Компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет. Программное обеспечение: Windows, Microsoft Office; Антивирусная защита: Kaspersky Free. Информационные источники: программное обеспечение общего и профессионального назначения; комплект учебно-наглядных пособий; тестовый материал, карточки индивидуальных заданий; методические рекомендации для проведения лабораторных занятий.

Требования охраны труда: ИОТ-11 при работе в кабинете информатики, ИОТ-64 для пользователей персональных компьютеров, ТБ-4 при работе на видеодисплейных терминалах и ЭВМ.

3.Пакет экзаменатора

3.1. Перечень тем, выносимых на экзамен:

Тема 1.1 Основы хранения и обработки данных. Проектирование БД.

Тема 2.1 Разработка и администрирование БД

Тема 3.1 Способы и методы защиты БД

3.2. Практико-ориентированные задания.

Примерный КИМ по экзамену

Задание 1. Выполнить тестирование в системе дистанционного обучения Moodle.

Часть А: 26 вопросов с выбором ответа

1. Какова основная цель теории баз данных?
 - a) Обеспечение безопасности данных
 - b) Максимизация производительности компьютера
 - c) Организация эффективного хранения и обработки данных
2. Что такое схема базы данных?
 - a) Документ, содержащий описание базы данных
 - b) Пользовательский интерфейс для доступа к данным
 - c) Специальное оборудование для хранения данных
3. Каким образом база данных может обеспечить целостность данных?
 - a) Автоматической проверкой прав доступа пользователей
 - b) Регулярным обновлением программного обеспечения
 - c) Использованием ограничений целостности и проверок
4. Что такое ER-модель?
 - a) Модель проектирования баз данных, основанная на сущностях и отношениях
 - b) Модель, описывающая таблицы и их структуру
 - c) Модель, описывающая процессы в системе
 - d) Модель, описывающая пользовательский интерфейс
5. Что такое внешний ключ в базе данных?
 - a) Ссылка на первичный ключ в другой таблице
 - b) Столбец, содержащий числовые значения
 - c) Описание структуры базы данных
 - d) Объект, хранящий данные таблицы
6. Что такое индекс в базе данных?
 - a) Структура данных, ускоряющая поиск и сортировку данных
 - b) Ключевое поле в таблице
 - c) Столбец с уникальными значениями
 - d) Связь между двумя таблицами
7. Что такое агрегатные функции в SQL?
 - a) Функции для соединения таблиц
 - b) Функции для работы с текстом
 - c) Функции для работы с датами и временем
 - d) Функции для работы с наборами данных (сумма, среднее значение, максимум и др.)
8. Какие виды схем баз данных существуют?
 - a) Концептуальная, логическая, физическая
 - b) Плоская, иерархическая, древовидная
 - c) Первичная, внешняя, альтернативная
 - d) Вертикальная, горизонтальная, диагональная
9. Какой из вариантов не является компонентом СУБД?

- a) Сервер баз данных
 - b) Клиент баз данных
 - c) SQL-запросы
 - d) Браузер для работы с СУБД
10. Какие операторы используются для задания условий в SQL запросе?
- a) AND, OR, NOT
 - b) +, -, *, /
 - c) =, <>, >, <
 - d) JOIN, UNION, INTERSECT
11. Какое ключевое слово используется для фильтрации данных после группировки?
- a) SORT
 - b) ORDER BY
 - c) GROUP BY
 - d) HAVING
12. Что такое транзакция в базе данных?
- a) Группа операций, считающаяся одним неделимым действием в базе данных
 - b) Систематический процесс копирования и защиты данных в базе данных
 - c) Функция, позволяющая сделать резервную копию базы данных
 - d) Процесс удаления данных, которые больше не нужны
13. Что такое резервное копирование базы данных?
- a) Процесс создания резервной копии данных в базе данных для их защиты от потери
 - b) Удаление ненужных данных из базы данных для освобождения места
 - c) Создание дополнительных копий данных для различных пользователей
 - d) Предоставление доступа к базе данных пользователю
14. Что такое нормализация базы данных?
- a) Процесс организации данных в базе с целью устранения избыточности и повышения эффективности работы
 - b) Создание резервной копии базы данных для защиты от потери данных
 - c) Установка дополнительных ограничений на целостность данных
 - d) Закрытие базы данных для предотвращения несанкционированного доступа
15. Что такое уровни изоляции транзакций в базе данных?
- a) Список доступных операций для каждой транзакции в базе данных
 - b) Права доступа к базе данных для различных пользователей
 - c) Уровни безопасности для защиты данных от взлома
 - d) Уровни, описывающие степень видимости изменений в базе данных для других транзакций
16. Какая команда SQL используется для удаления таблицы из базы данных?
- a) DELETE TABLE
 - b) REMOVE TABLE
 - c) DROP TABLE
 - d) ERASE TABLE
17. Какая технология позволяет ограничить доступ к базе данных только определенным пользователям?
- a) Аутентификация
 - b) Авторизация
 - c) Шифрование

- d) Бэкап
18. Каким образом можно защитить базу данных от несанкционированных изменений?
- a) Шифрование
 - b) Файервол
 - c) Цифровая подпись
 - d) Кластеризация
19. Какую функцию выполняет система контроля целостности базы данных?
- a) Обеспечивает аутентификацию пользователей
 - b) Проверяет, что данные не были изменены или повреждены
 - c) Шифрует данные
 - d) Оптимизирует запросы к базе данных
20. Какие методы аутентификации можно использовать при доступе к базе данных?
- a) Пароль
 - b) Биометрический отпечаток
 - c) Карта доступа
 - d) Все вышеперечисленные
21. Какую функцию выполняет система аудита базы данных?
- a) Осуществляет контроль прав доступа пользователей
 - b) Проверяет целостность данных при их сохранении на диске
 - c) Регистрирует действия пользователей с базой данных
 - d) Анализирует сетевой трафик и обнаруживает атаки
22. Какой тип резервного копирования обеспечивает наименьшее время восстановления базы данных?
- a) Полное резервное копирование
 - b) Инкрементное резервное копирование
 - c) Дифференциальное резервное копирование
 - d) Точечное резервное копирование
23. Каким образом можно защитить базу данных от атак типа отказ в обслуживании (DDoS)?
- a) Использовать систему фильтрации трафика
 - b) Использовать систему аутентификации
 - c) Использовать систему IDS/IPS
 - d) Использовать систему резервного копирования
24. Какая технология позволяет обеспечить конфиденциальность и целостность данных базы данных?
- a) Шифрование данных
 - b) IDS (Intrusion Detection System)
 - c) Аутентификация пользователей
 - d) Файрвол
25. Какая технология позволяет защитить базу данных от неправомерного доступа при утере или компрометации пароля пользователя?
- a) Двухфакторная аутентификация
 - b) IP-фильтрация
 - c) Асимметричное шифрование
 - d) Биометрическая аутентификация
26. Какие из перечисленных утверждений верны относительно резервного копирования базы данных?

- a) Регулярные резервные копии обеспечивают возможность восстановления данных
- b) Резервные копии всегда занимают много места
- c) Резервные копии могут быть созданы только вне рабочего времени
- d) Резервные копии могут быть сохранены на удаленном сервере для дополнительной защиты

Часть В.

Вставьте пропущенные слова.

1. Программное обеспечение для создания и управления базами данных называется _____.
2. Структура данных, ускоряющая поиск записей в таблице – это _____.
3. Существуют следующие методы резервного копирования данных – _____.
4. Использование методов _____ обеспечивает безопасность базы данных.
5. В контексте баз данных набор инструкций, выполняемых по запросу – это _____; набор инструкций, выполняемых автоматически при определенных событиях – это _____.
6. В реляционной базе данных процесс организации данных для устранения избыточности и обеспечения эффективности хранения называется _____.
7. _____ – тип баз данных, предназначенных для хранения и обработки данных, не соответствующих модели реляционных баз данных.
8. Копия данных т.е. _____ используется для восстановления информации в случае потери или повреждения.
9. Такие методы, как _____ обеспечивают конфиденциальность данных в базе данных.

Часть С.

1. Кейс: Проектирование базы данных.

Ситуация: Вам необходимо спроектировать базу данных для управления библиотекой с книгами, авторами и читателями. Какие таблицы, поля и связи между ними будут включены?

2. Кейс: Нормализация данных.

Ситуация: Ваша база данных содержит повторяющиеся данные. Какие шаги вы предпримете для нормализации данных?

Задание 2. Решение практико-ориентированных задач по реализации запросов на языке высокого уровня SQL.

Билет 1. Практическая задача «Реализация запросов с помощью языка SQL»

1. Создайте таблицу "Студенты" с полями "Имя", "Фамилия", "Возраст" и "Группа". Вставьте в нее следующие данные:
 - Иван, Иванов, 20, Группа 1
 - Петр, Петров, 22, Группа 2
 - Алексей, Алексеев, 21, Группа 1
2. Напишите SQL-запрос, который выводит все данные из таблицы "Студенты".
3. Напишите SQL-запрос, который выводит только имена студентов из таблицы "Студенты".
4. Напишите SQL-запрос, который выводит студентов, возраст которых больше или равен 21.
5. Напишите SQL-запрос, который выводит студентов из группы 1.
6. Напишите SQL-запрос, который удаляет студента с именем "Петр".
7. Напишите SQL-запрос, который изменяет группу студента с именем "Алексей" на "Группа 3".

Билет 2. Практическая задача «Реализация запросов с помощью языка SQL»

1. Создайте таблицу "Предметы" с полями "Название" и "Преподаватель". Вставьте в нее следующие данные:
 - Математика, Иванов И.И.
 - Английский язык, Петров П.П.
 - Физика, Сидоров С.С.
2. Напишите SQL-запрос, который выводит список всех предметов и преподавателей из таблицы "Предметы".
3. Напишите SQL-запрос, который находит предметы, преподаваемые Ивановым.
4. Напишите SQL-запрос, который находит предметы, не преподаваемые Сидоровым.

Билет 3. Практическая задача «Реализация запросов с помощью языка SQL»

1. Создайте таблицу "Оценки" с полями "Предмет", "Студент" и "Оценка". Вставьте в нее следующие данные:
 - Математика, Иван, 4
 - Математика, Петр, 5
 - Физика, Иван, 3
 - Физика, Петр, 4
2. Напишите SQL-запрос, который выводит список всех оценок студентов.
3. Напишите SQL-запрос, который выводит среднюю оценку студента с именем "Иван".
4. Напишите SQL-запрос, который выводит среднюю оценку по каждому предмету.
5. Напишите SQL-запрос, который выводит предметы, для которых средняя оценка больше 4.
6. Напишите SQL-запрос, который удаляет все оценки студента с именем "Петр".
7. Напишите SQL-запрос, который изменяет оценку студента с именем "Иван" по предмету "Математика" на 5.
8. Напишите SQL-запрос, который находит студента с максимальной оценкой.
9. Напишите SQL-запрос, который находит предметы, по которым есть оценки ниже 3.

4.Эталоны ответов

Задание 1. Выполнить тестирование в системе дистанционного обучения Moodle (26 тестовых вопроса и задания)

Часть А. 26 вопросов с выбором ответа

1. с; 2. а; 3. с; 4. а; 5. а; 6. а; 7. d; 8. а; 9. d; 10. а, с; 11. d; 12. а; 13. а; 14. а; 15. d; 16. с; 17. b; 18. с; 19. b; 20. d; 21. с; 22. d; 23. а; 24. а; 25. а; 26. а

Часть В. Вставьте пропущенные слова.

1. система управления базами данных (СУБД)
2. индекс
3. полное, дифференциальное, инкрементальное
4. аутентификация, авторизация, шифрование данных
5. Процедура, Триггер.
6. Нормализация
7. NoSQL
8. бэкап базы данных
9. шифрование данных, управление доступом, мониторинг аудита

Часть С.

1. Кейс: Проектирование базы данных.

Ответ: необходимо создать три таблицы: "Книги", "Авторы" и "Читатели". Связи между ними будут представлены внешними ключами, например, автор ID в таблице "Книги" и читатель ID в таблице "Читатели".

2. Кейс: Нормализация данных.

Ответ: Используя процесс нормализации, разбиваем таблицы с повторяющимися данными на отдельные таблицы. Обычно это включает первую, вторую и третью нормальные формы.

Задание 2. Решение практико-ориентированных задач по реализации запросов на языке высокого уровня SQL.

Билет 1. Практическая задача «Реализация запросов с помощью языка SQL»

1. CREATE TABLE Студенты (Имя VARCHAR (255), Фамилия VARCHAR (255), Возраст INT, Группа VARCHAR (255));
INSERT INTO Студенты (Имя, Фамилия, Возраст, Группа)
VALUES ('Иван', 'Иванов', 20, 'Группа 1'),
('Петр', 'Петров', 22, 'Группа 2'),
('Алексей', 'Алексеев', 21, 'Группа 1');
2. SELECT * FROM Студенты;
3. SELECT Имя FROM Студенты;
4. SELECT * FROM Студенты WHERE Возраст >= 21;
5. SELECT * FROM Студенты WHERE Группа = 'Группа 1';
6. DELETE FROM Студенты WHERE Имя = 'Петр';
7. UPDATE Студенты SET Группа = 'Группа 3' WHERE Имя = 'Алексей';

Билет 2. Практическая задача «Реализация запросов с помощью языка SQL»

1. CREATE TABLE Предметы (Название VARCHAR (255), Преподаватель VARCHAR (255));
INSERT INTO Предметы (Название, Преподаватель) VALUES ('Математика', 'Иванов И.И.'), ('Английский язык', 'Петров П.П.'), ('Физика', 'Сидоров С.С.');
2. SELECT * FROM Предметы;
3. SELECT * FROM Предметы WHERE Преподаватель = 'Иванов И.И.';
4. SELECT * FROM Предметы WHERE Преподаватель <> 'Сидоров С.С.';

Билет 3. Практическая задача «Реализация запросов с помощью языка SQL»

1. CREATE TABLE Оценки (Предмет VARCHAR (255), Студент VARCHAR (255), Оценка INT);
INSERT INTO Оценки (Предмет, Студент, Оценка) VALUES ('Математика', 'Иван', 4),
('Математика', 'Петр', 5), ('Физика', 'Иван', 3), ('Физика', 'Петр', 4);
2. SELECT * FROM Оценки;
3. SELECT AVG(Оценка) FROM Оценки WHERE Студент = 'Иван';
4. SELECT Предмет, AVG(Оценка) FROM Оценки GROUP BY Предмет;
5. SELECT Предмет FROM Оценки GROUP BY Предмет HAVING AVG(Оценка) > 4;
6. DELETE FROM Оценки WHERE Студент = 'Петр';
7. UPDATE Оценки SET Оценка = 5 WHERE Студент = 'Иван' AND Предмет = 'Математика';
8. SELECT Студент FROM Оценки GROUP BY Студент HAVING MAX (Оценка);
9. SELECT Предмет FROM Оценки WHERE Оценка <3;

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
<p>У1 работать с современными case-средствами проектирования баз данных;</p> <p>У2 проектировать логическую и физическую схемы базы данных;</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p> <p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p> <p>ПК 11.1 Осуществлять сбор, обработку и анализ информации для проектирования баз данных</p> <p>ПК 11.2 Проектировать базу данных на основе анализа предметной области</p> <p>ПК 11.3 Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области</p>	<ul style="list-style-type: none"> – работает с изобразительными средствами, используемыми в ER- моделировании реляционной базы данных; – демонстрирует способности решения профессиональных задач практического содержания; – проектирует логическую и физическую схемы базы данных; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – использует современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; – осуществляет сбор, обработку и анализ информации для проектирования баз данных. – проектирует базу данных на основе анализа предметной области; – разрабатывает объекты базы данных в соответствии с результатами анализа предметной области;

	<ul style="list-style-type: none"> – реализует базу данных в конкретной системе управления базами данных.
<p>У3 создавать хранимые процедуры и триггеры на базах данных;</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p> <p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке</p> <p>ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных</p> <p>ПК 11.5 Администрировать базы данных</p>	<ul style="list-style-type: none"> – умеет создавать хранимые процедуры и триггеры на базах данных; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке; – разрабатывает объекты базы данных в соответствии с результатами анализа предметной области; – реализует базу данных в конкретной системе управления базами данных; – выполняет задачи по администрированию базы данных.
<p>У4 применять стандартные методы для защиты объектов базы данных;</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p>	<ul style="list-style-type: none"> – применяет стандартные методы для защиты объектов базы данных; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие;

<p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p> <p>ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных</p> <p>ПК 11.5 Администрировать базы данных</p> <p>ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации</p>	<ul style="list-style-type: none"> – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке; – разрабатывает объекты базы данных в соответствии с результатами анализа предметной области; – реализует базу данных в конкретной системе управления базами данных; – выполняет задачи по администрированию базы данных; – обеспечивает защиту информации в базе данных с использованием различных технологий защиты информации.
<p>У5 выполнять стандартные процедуры резервного копирования и мониторинга выполнения этой процедуры;</p> <p>У6 выполнять процедуру восстановления базы данных и вести мониторинг выполнения этой процедуры;</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p> <p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p>	<ul style="list-style-type: none"> – выполняет стандартные процедуры резервного копирования и восстановления базы данных, ведет мониторинг выполнения этих процедур; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности;

<p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p> <p>ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных</p> <p>ПК 11.5 Администрировать базы данных</p> <p>ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации</p>	<ul style="list-style-type: none"> – пользуется профессиональной документацией на государственном и иностранном языке; – реализует базу данных в конкретной системе управления базами данных; – выполняет задачи по администрированию базы данных; – обеспечивает защиту информации в базе данных с использованием различных технологий защиты информации.
<p>У7 обеспечивать информационную безопасность на уровне базы данных.</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p> <p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p> <p>ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации</p>	<ul style="list-style-type: none"> – обеспечивает информационную безопасность на уровне базы данных; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке; – обеспечивает защиту информации в базе данных с использованием различных технологий защиты информации.
<p>Знать:</p>	
<p>З1 основные положения теории баз данных, хранилищ данных, баз знаний;</p>	<ul style="list-style-type: none"> – знает основные положения теории баз данных, хранилищ данных, баз знаний;

32 основные принципы структуризации и нормализации базы данных;	– знает основные принципы структуризации и нормализации базы данных;
33 основные принципы построения концептуальной, логической и физической модели данных;	– понимает основные принципы построения концептуальной, логической и физической модели данных;
34 методы описания схем баз данных в современных системах управления базами данных;	– освоил методы описания схем баз данных в современных системах управления базами данных;
35 структуры данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;	– знает структуры данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
36 методы организации целостности данных;	– знает методы организации целостности данных;
37 способы контроля доступа к данным и управления привилегиями;	– обеспечивает способы контроля доступа к данным и управления привилегиями;
38 основные методы и средства защиты данных в базах данных.	– имеет представление об основных методах и средствах защиты данных в базах данных.

Критерии оценивания

Оценка по традиционной системе	Процент	Критерий
Отлично	от 85 %	Продемонстрирован высокий уровень знаний по всему объему изучаемого учебного курса; в полном объеме выполнены практико-ориентированные задания.
Хорошо	от 70% до 85 %	Продемонстрировано понимание основного содержания изучаемого междисциплинарного курса; выполнены практико-ориентированные задания.
Удовлетворительно	от 55 % до 70 %	Продемонстрировано неполное понимание основного содержания изучаемого междисциплинарного курса; частично выполнены практико-ориентированные задания.
Неудовлетворительно	менее 55 %	Не продемонстрировано владение знаниями; не выполнены практико-ориентированные задания.

5. Экзаменационная ведомость.

5. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ДЛЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

1. Форма проведения: решение практико-ориентированной задачи (составлена на основании оценочных материалов демонстрационного экзамена, утвержденных Протоколом заседания Педагогического совета ФГБОУ ДПО ИРПО от «3» августа 2023 г. № 8; шифр комплекта оценочной документации: КОД 09.02.07-2-2024) и кейс-ситуаций.

2.Условия выполнения

Инструкция для обучающихся:

Уважаемые студенты!

Вы присутствуете на экзамене по ПМ.11 Разработка, администрирование и защита баз данных. Данный экзамен состоит из двух частей: решение практико-ориентированной задачи «Разработка и реализация базы данных для учета заявок на ремонт оборудования» и кейс-ситуаций. Не забывайте делать перерывы в работе. Желаем вам успехов в сдаче экзамена!

Время выполнения задания: 180 мин

Оборудование учебного кабинета: рабочие места обучающихся (16 ед.); рабочее место преподавателя; учебная доска; рабочая программа профессионального модуля «Разработка, администрирование и защита баз данных», методические указания по выполнению заданий экзамена квалификационного.

Технические средства обучения: компьютеры с лицензионным программным обеспечением; мультимедийный комплекс; электронные учебники. Компьютеры подключены к локальной вычислительной сети, информационно-образовательной среде КГА ПОУ «ДИТК» и сети Интернет.

Программное обеспечение: Windows, Microsoft Office; Антивирусная защита: Kaspersky Free. Информационные источники: программное обеспечение общего и профессионального назначения; комплект учебно-наглядных пособий; тестовый материал, карточки индивидуальных заданий; методические рекомендации для проведения лабораторных занятий.

Требования охраны труда: ИОТ-11 при работе в кабинете информатики, ИОТ-64 для пользователей персональных компьютеров, ТБ-4 при работе на видеодисплейных терминалах и ЭВМ.

3.Пакет экзаменатора

3.1. Перечень тем, выносимых на экзамен:

Тема 1.1 Основы хранения и обработки данных. Проектирование БД.

Тема 2.1 Разработка и администрирование БД

Тема 3.1 Способы и методы защиты БД

3.2. Практико-ориентированные задания.

Задание 1. «Разработка и реализация базы данных для учета заявок на ремонт оборудования»

Описание предметной области

Основная цель учёта заявок на ремонт оборудования - эффективное и оперативное осуществление ремонтных работ с минимизацией простоев и удовлетворением запросов клиентов или сотрудников. Эта предметная область широко используется в различных сферах деятельности, таких как сервисные услуги, производство, информационные технологии и другие.

Предметная область учёта заявок на ремонт оборудования касается процесса подачи, обработки и учёта заявок на ремонт различного оборудования.

В данной области включены следующие основные составляющие:

1. Заявка на ремонт: это информация, предоставленная клиентом или сотрудником о неисправности оборудования, которое требует ремонта. Заявка может содержать данные о типе оборудования, его серийном номере, описании проблемы и другой важной информации.

2. Регистрация заявки: этот процесс включает приём и регистрацию заявки в системе учёта. Важными аспектами регистрации являются присвоение уникального идентификатора заявке, сохранение информации о заявке и её приоритете.

3. Обработка заявки: процесс, включающий анализ заявки, определение её

приоритетности и назначение исполнителя (ремонтного специалиста) для задачи. В процессе обработки может потребоваться дополнительная информация или уточнение деталей проблемы у клиента или сотрудника.

4. Исполнение заявки: фактическое выполнение ремонта оборудования. В этом этапе назначенный исполнитель ремонтирует оборудование, вносит необходимые изменения или заменяет неисправные компоненты. Важно отметить, что на этом этапе могут возникать необходимость заказа запчастей или координации работ с другими специалистами.

5. Отчётность и информирование: важной составляющей учёта заявок на ремонт является фиксация и отчёт о выполненной работе. После завершения ремонта, исполнитель должен предоставить отчёт о проделанной работе, включая информацию о затраченных ресурсах (время, материалы, стоимость), причине неисправности и оказанной помощи.

6. Мониторинг и анализ: этот этап предполагает контроль и анализ процесса учёта заявок на ремонт. Важно отслеживать и анализировать время обработки заявок, качество выполненных работ, расходы и прочие параметры, которые могут помочь в оптимизации и улучшении процесса.

Техническое задание для решения задачи по

ПМ.11 «Разработка, администрирование и защита баз данных»

1. Общие сведения

1.1. Наименование проекта: Разработка и реализация базы данных для учета заявок на ремонт оборудования

1.2. Заказчик: ООО "Техносервис".

1.3. Исполнитель: Компания "IT-Решения".

2. Функциональные требования

2.1. Возможность добавления заявок в базу данных с указанием следующих параметров:

- Номер заявки;
- Дата добавления;
- Оборудование, которое требует ремонта;
- Тип неисправности;
- Описание проблемы;
- Клиент, который подал заявку;
- Статус заявки (в ожидании, в работе, выполнено).

2.2. Возможность редактирования заявок:

- Изменение этапа выполнения (выполнено, в работе, не выполнено);
- Изменение описания проблемы;
- Изменение, ответственного за выполнение работ.

2.3. Возможность отслеживания статуса заявки:

- Отображение списка заявок;
- Получение уведомлений о смене статуса заявки;
- Поиск заявки по номеру или по параметрам.

2.4. Возможность назначения ответственных за выполнение работ:

- Добавление исполнителя к заявке;
- Отслеживание состояния работы и получение уведомлений о ее завершении;
- Исполнитель может добавлять комментарии на форме заявки.

2.5. Расчет статистики работы отдела обслуживания:

- Количество выполненных заявок;
- Среднее время выполнения заявки;
- Статистика по типам неисправностей.

3. Нефункциональные требования

3.1. Кроссплатформенность:

- Поддержка работы на ОС семейства Windows.

3.2. Безопасность:

- Логин и пароль для доступа к приложению;
- Доступ к данным должен быть ограничен в зависимости от роли пользователя.

3.3. Удобство использования:

- Простой и интуитивный интерфейс;
- Информативные уведомления и подсказки.

3.4. Производительность:

- Приложение должно иметь быстрый доступ к данным;
- Минимальное время отклика на запросы пользователя.

4. Требования к реализации

4.1. Язык программирования: на усмотрение разработчика

4.2. СУБД: на усмотрение разработчика

5. Требования к документации

5.1. Техническое задание на разработку программного модуля.

5.2. Руководство системному программисту.

Дополнение к техническому заданию

для решения задачи по ПМ.11 «Разработка, администрирование и защита баз данных»

На этапе исполнения заявки: фактическое выполнение ремонта оборудования.

Назначенный исполнитель ремонтирует оборудование, вносит необходимые изменения или заменяет неисправные компоненты. Важно отметить, что на этом этапе могут возникать необходимость заказа запчастей или координации работ с другими специалистами.

Необходимо добавить в функционал программного модуля еще одну роль – менеджера, который будет выполнять роль консультанта при возникающих проблемах с ремонтом оборудования и не выполнении ремонта в срок. При необходимости исполнитель заявки на ремонт может обратиться за помощью при возникновении невозможности выполнения ремонта.

Менеджер имеет право:

1. привлекать других специалистов к выполнению ремонта;
2. продлевать срок выполнения заявки с согласованием клиента.

Также по требованиям заказчика необходимо отслеживать качество работы по отзывам. В функционал программного модуля в этих целях необходимо добавить генерацию QR-кода для оценки работы сервиса (при сканировании кода в телефоне выдается ссылка на гугл-форму с опросом (ссылка в ресурсах)).

Задание 2. Кейс-ситуация с вариантами ответов. Обеспечить выбор верного варианта ответа и обосновать оптимальность выбора (в качестве примера представлены 5 ситуаций).

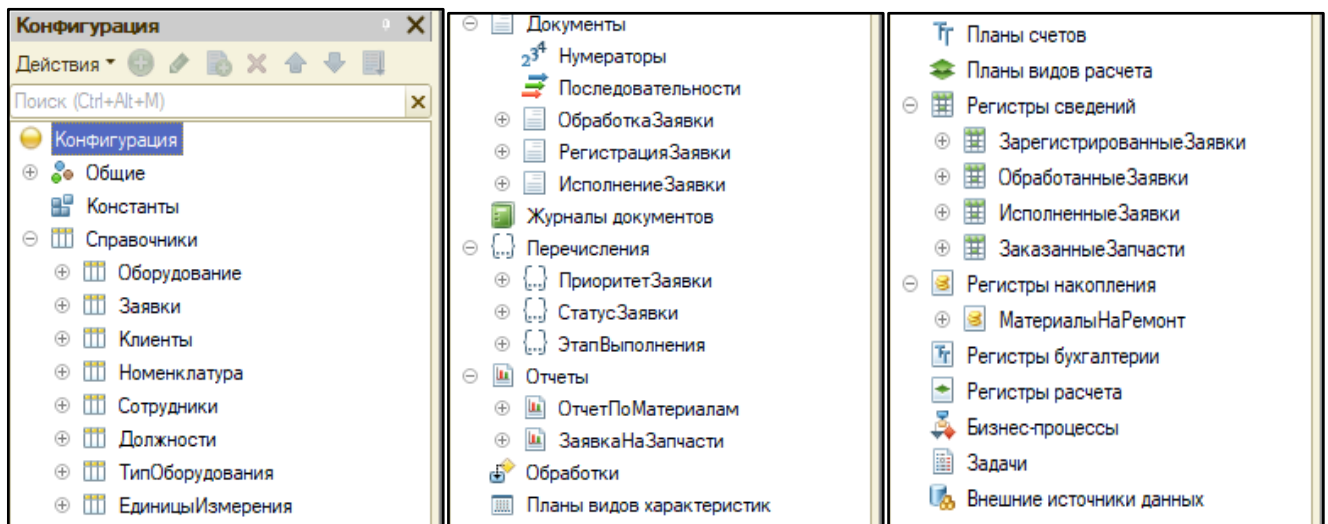
1. Ваш сервер баз данных перестал отвечать, и вы получили уведомление о том, что база данных не доступна. Каков ваш следующий шаг?
 - а) Попробовать перезапустить сервер баз данных.

- б) Проверить журналы ошибок и выявить причину сбоя.
 - в) Игнорировать уведомление и продолжить работу.
2. Ваша база данных содержит большое количество дублирующейся информации. Каким образом вы собираетесь оптимизировать и очистить данные?
 - а) Удалить все данные и создать новую базу данных.
 - б) Использовать SQL-запросы для поиска и удаления дублирующихся записей.
 - в) Ничего не предпринимать, так как дублирование не влияет на работу базы данных.
 3. Пользователи жалуются на медленную работу базы данных. Какие шаги вы предпримете для оптимизации производительности?
 - а) Увеличить объем оперативной памяти на сервере баз данных.
 - б) Использовать индексы для ускорения поисковых запросов.
 - в) Попросить пользователей работать медленнее.
 4. Ваша база данных подверглась атаке хакера, и часть данных была скомпрометирована. Каким образом вы восстановите данные и укрепите безопасность?
 - а) Восстановить данные из резервной копии и усилить механизмы аутентификации.
 - б) Ничего не предпринимать, так как атаки хакеров неизбежны.
 - в) Изменить пароли только для администраторских аккаунтов.
 5. Ваша база данных столкнулась с проблемой совместимости при обновлении программного обеспечения. Как вы решите эту проблему?
 - а) Откатить обновление и вернуться к предыдущей версии программного обеспечения.
 - б) Провести тестирование обновления в тестовой среде перед его внедрением.
 - в) Переустановить базу данных без обновления.

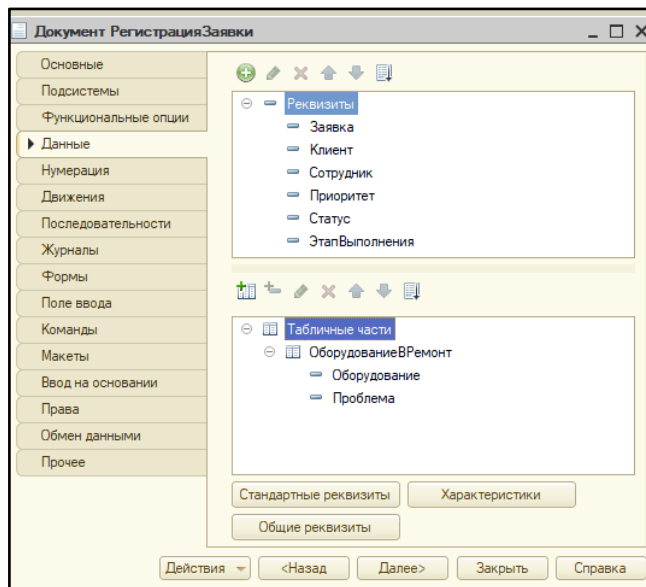
Эталоны ответов

Задание 1. Разработка и реализация базы данных для учета заявок на ремонт оборудования

1. Дерево объектов конфигурации базы данных «Учет заявок».



2. Справочники конфигурации имеют простейшую структуру (стандартные реквизиты).
3. Структура документа «Регистрация заявки».



4. Модуль документа «Регистрация заявки».

```

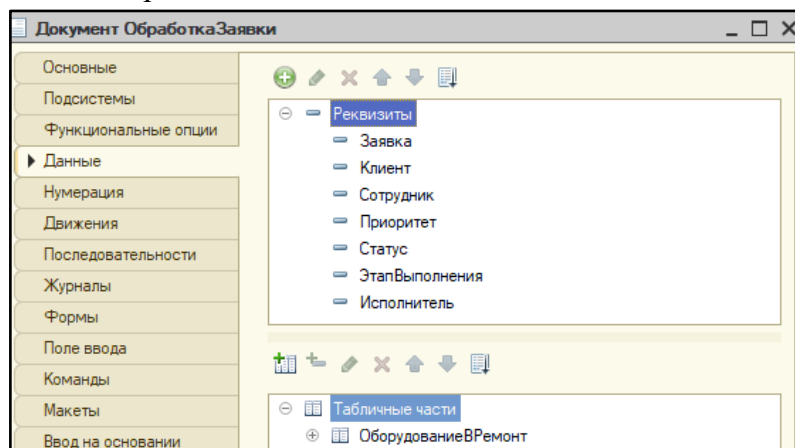
Документ РегистрацияЗаявки: Модуль объекта
[ Процедура ОбработкаПроведения(Отказ, Режим)
  //{{ __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!

  // регистр ЗарегистрированныеЗаявки
  Движения.ЗарегистрированныеЗаявки.Записывать = Истина;
  Для Каждого ТекСтрокаОборудованиеВРемонт Из ОборудованиеВРемонт Цикл
    Движение = Движения.ЗарегистрированныеЗаявки.Добавить();
    Движение.Период = Дата;
    Движение.Заявка = Заявка;
    Движение.Клиент = Клиент;
    Движение.Сотрудник = Сотрудник;
    Движение.Оборудование = ТекСтрокаОборудованиеВРемонт.Оборудование;
    Движение.Приоритет = Приоритет;
  КонечныйЦикл;

  //}} __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

```

5. Структура документа «Обработка заявки».



6. Модуль документа «Обработка заявки».

```

Документ ОбработкаЗаявки: Модуль объекта

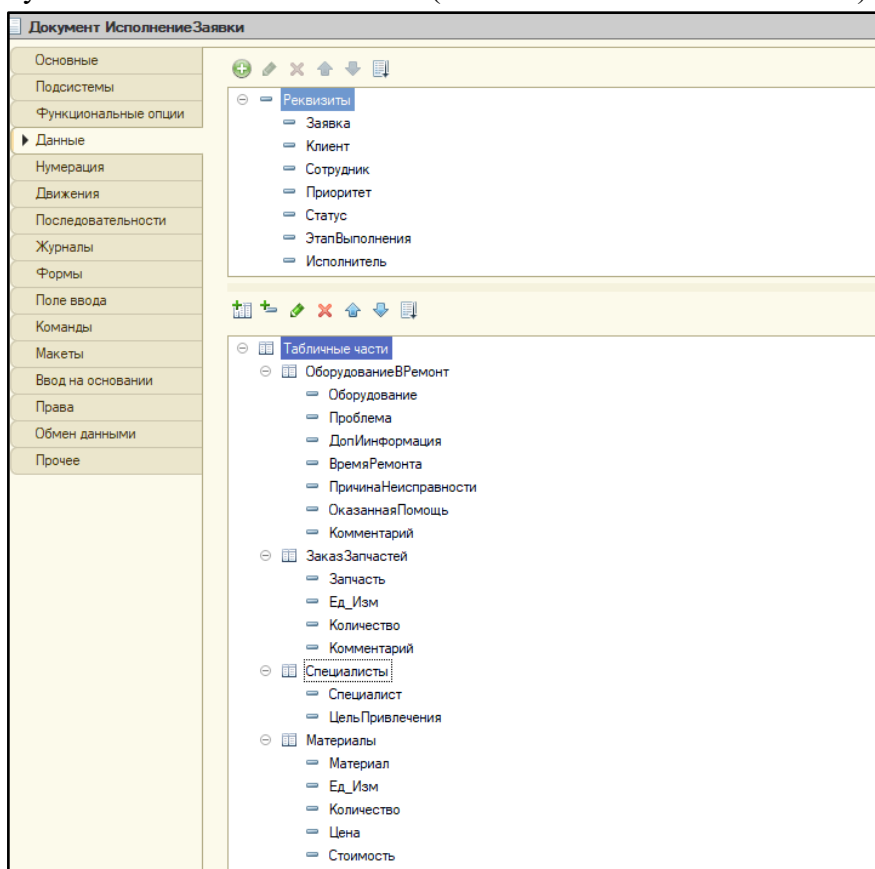
□ Процедура ОбработкаЗаполнения (ДанныеЗаполнения, СтандартнаяОбработка)
  {{{_КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
  Если ТипЗнч (ДанныеЗаполнения) = Тип ("ДокументСсылка.РегистрацияЗаявки") Тогда
    // Заполнение шапки
    Клиент = ДанныеЗаполнения.Клиент;
    Заявка = ДанныеЗаполнения.Заявка;
    Приоритет = ДанныеЗаполнения.Приоритет;
    Сотрудник = ДанныеЗаполнения.Сотрудник;
    Статус = ДанныеЗаполнения.Статус;
    ЭтапВыполнения = ДанныеЗаполнения.ЭтапВыполнения;
    Для Каждого ТекСтрокаОборудованиеВРемонт Из ДанныеЗаполнения.ОборудованиеВРемонт Цикл
      НоваяСтрока.Оборудование = ТекСтрокаОборудованиеВРемонт.Оборудование;
      НоваяСтрока.Проблема = ТекСтрокаОборудованиеВРемонт.Проблема;
    КонечЦикла;
  КонецЕсли;
  }}}_КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
  КонецПроцедуры

□ Процедура ОбработкаПроведения (Отказ, Режим)
  {{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

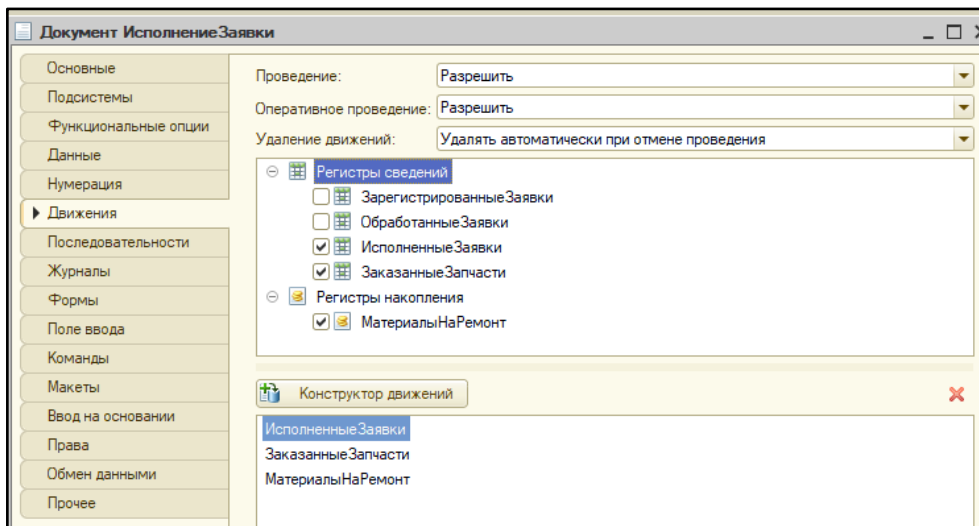
  // регистр ОбработанныеЗаявки
  Движения.ОбработанныеЗаявки.Записывать = Истина;
  Для Каждого ТекСтрокаОборудованиеВРемонт Из ОборудованиеВРемонт Цикл
    Движение = Движения.ОбработанныеЗаявки.Добавить ();
    Движение.Период = Дата;
    Движение.Заявка = Заявка;
    Движение.Клиент = Клиент;
    Движение.Сотрудник = Сотрудник;
    Движение.Оборудование = ТекСтрокаОборудованиеВРемонт.Оборудование;
    Движение.Исполнитель = Исполнитель;
    Движение.Приоритет = Приоритет;
    Движение.Статус = Статус;
    Движение.ЭтапВыполнения = ЭтапВыполнения;
  КонечЦикла;
  }}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  КонецПроцедуры

```

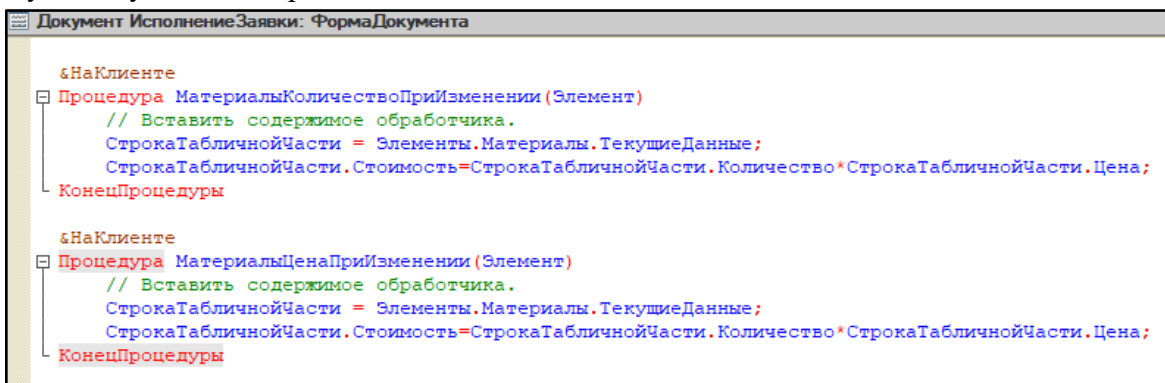
7. Структура документа «Исполнение заявки» (наличие 4-х табличных частей).



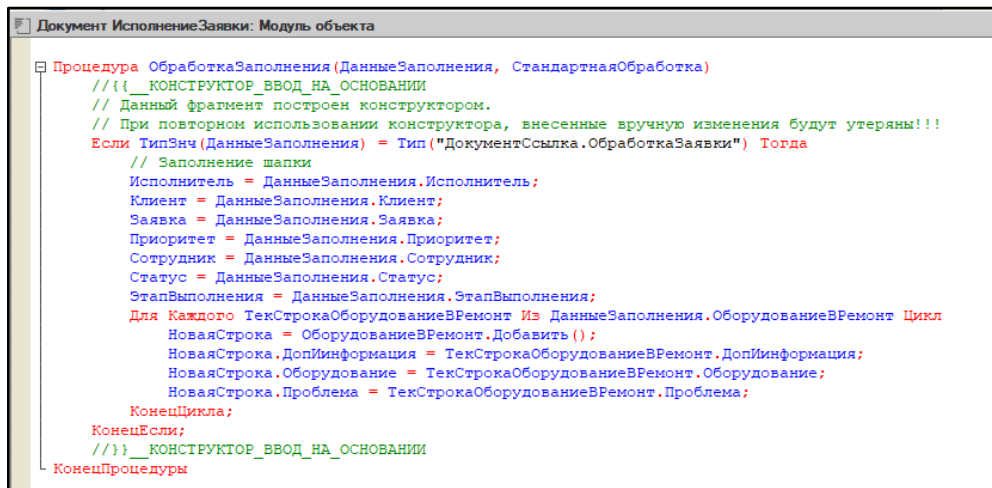
8. Движение документа «Исполнение заявки» по регистрам.



9. Модуль документа «Обработка заявки».



10. Модуль документа «Исполнение заявки», процедура «Обработка заполнения».



11. Модуль документа «Исполнение заявки», процедура «Обработка проведения».

```

Документ ИсполнениеЗаявки: Модуль объекта
[ Процедура ОбработкаПроведения(Отказ, Режим)
  //({ _КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
  // Данный фрагмент построен конструктором.
  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

  // регистр ИсполненныеЗаявки
  Движения.ИсполненныеЗаявки.Записывать = Истина;
  Для Каждого ТекСтрокаОборудованиеВРемонт Из ОборудованиеВРемонт Цикл
    Движение = Движения.ИсполненныеЗаявки.Добавить ();
    Движение.Период = Дата;
    Движение.Заявка = Заявка;
    Движение.Клиент = Клиент;
    Движение.Сотрудник = Сотрудник;
    Движение.Оборудование = ТекСтрокаОборудованиеВРемонт.Оборудование;
    Движение.Исполнитель = Исполнитель;
    Движение.ВремяРемонта = ТекСтрокаОборудованиеВРемонт.ВремяРемонта;
    Движение.Приоритет = Приоритет;
    Движение.Статус = Статус;
    Движение.ЭтапВыполнения = ЭтапВыполнения;
    Движение.ОказаннаяПомощь = ТекСтрокаОборудованиеВРемонт.ОказаннаяПомощь;
  КонечЦикла;

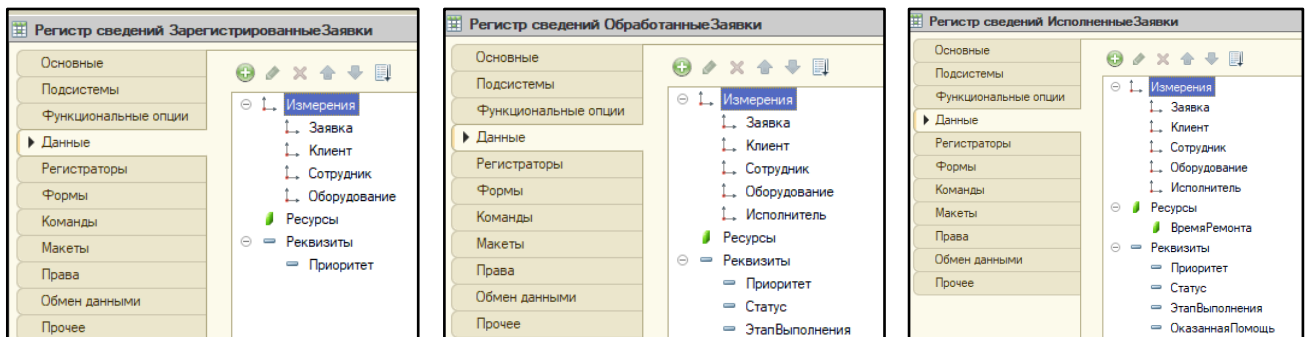
  // регистр ЗаказанныеЗапчасти
  Движения.ЗаказанныеЗапчасти.Записывать = Истина;
  Для Каждого ТекСтрокаЗаказЗапчастей Из ЗаказЗапчастей Цикл
    Движение = Движения.ЗаказанныеЗапчасти.Добавить ();
    Движение.Период = Дата;
    Движение.Заявка = Заявка;
    Движение.Клиент = Клиент;
    Движение.Сотрудник = Сотрудник;
    Движение.Исполнитель = Исполнитель;
    Движение.Запчасть = ТекСтрокаЗаказЗапчастей.Запчасть;
    Движение.Количество = ТекСтрокаЗаказЗапчастей.Количество;
    Движение.ЕдИзм = ТекСтрокаЗаказЗапчастей.Ед_Изм;
  КонечЦикла;

  // регистр МатериалыНаРемонт
  Движения.МатериалыНаРемонт.Записывать = Истина;
  Для Каждого ТекСтрокаМатериалы Из Материалы Цикл
    Движение = Движения.МатериалыНаРемонт.Добавить ();
    Движение.Период = Дата;
    Движение.Заявка = Заявка;
    Движение.Материал = ТекСтрокаМатериалы.Материал;
    Движение.Количество = ТекСтрокаМатериалы.Количество;
    Движение.Цена = ТекСтрокаМатериалы.Цена;
    Движение.Стоимость = ТекСтрокаМатериалы.Стоимость;
    Движение.ЕдИзм = ТекСтрокаМатериалы.Ед_Изм;
  КонечЦикла;

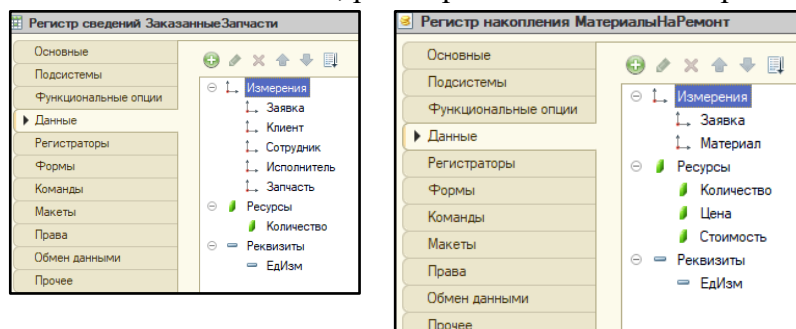
  //}) _КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонечПроцедуры

```

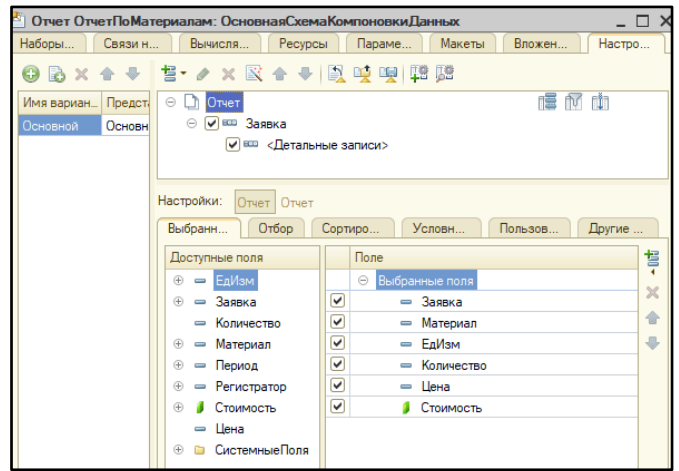
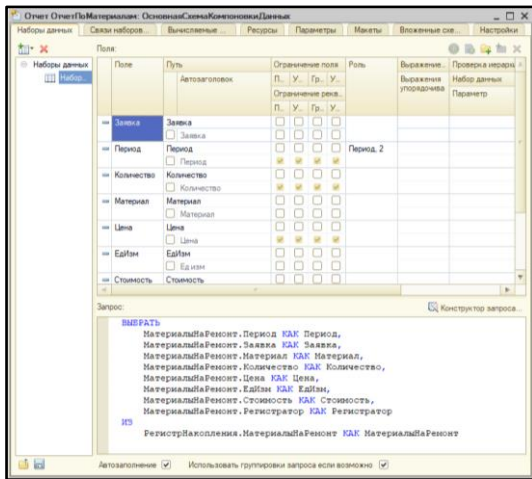
12. Регистры сведений «Зарегистрированные заявки», «Обработанные заявки», «Исполненные заявки» конфигурации.



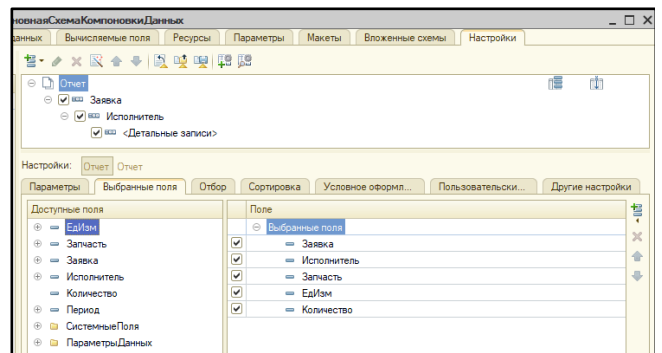
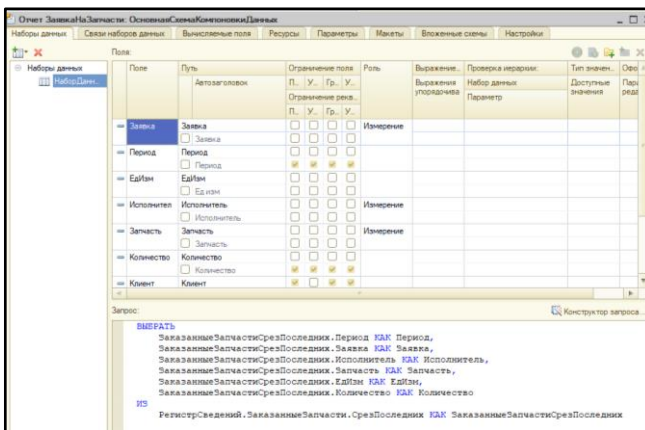
13. Регистр сведений «Заказанные запчасти», регистр накопления «Материалы на ремонт».



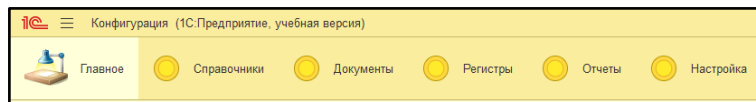
14. Основная схема компоновки данных отчета «Остатки по материалам», вкладки «Наборы данных» и «Настройки».



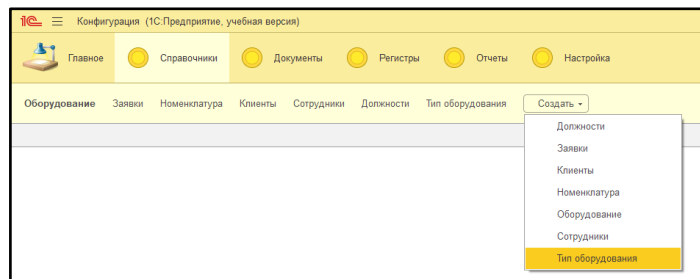
15. Основная схема компоновки данных отчета «Заявка на запчасти», вкладки «Наборы данных» и «Настройки».



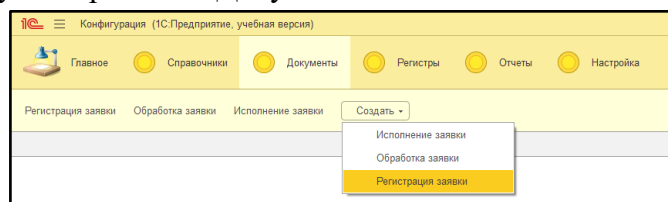
16. Панель разделов конфигурации «Заявки».



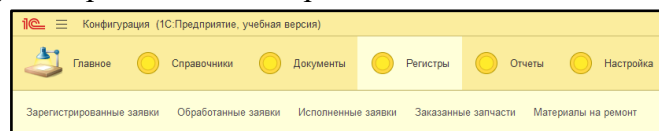
17. Панель команд текущего раздела «Справочники».



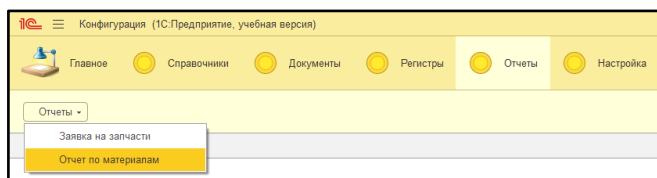
18. Панель команд текущего раздела «Документы».



19. Панель команд текущего раздела «Регистры».



20. Панель команд текущего раздела «Отчеты».



21. Справочники конфигурации «Номенклатура», «Клиенты», «Сотрудники».

Номенклатура

Наименование	Код
Запчасти	00000003
Валик	00000002
Материалы	00000004
Валик	00000006
Картридж	00000005
Тонер	00000001

Клиенты

Наименование	Код
ООО "Альтаир"	00000001
ООО "Меридиан"	00000002

Сотрудники

Наименование	Код	Должность
Иванов И.И.	00000004	Программист
Иванов И.И.	00000001	Наладчик оборудования
Петров П.П.	00000002	Инженер-электронщик
Смирнов С.С.	00000003	Программист

22. Справочники конфигурации «Заявки», «Должности», «Тип оборудования».

Заявки

Наименование	Номер_Заявки
Заправка картриджа (ООО "Альтаир")_10.11.23	00000001
Ремонт МФУ (Смирнов С.С.)_10.11.23	00000002

Должности

Наименование
Инженер-электронщик
Наладчик оборудования
Программист

Тип оборудования

Наименование	Код
Вычислительная техника	00000001
Оргтехника	00000002

23. Список документов «Регистрация заявки».

Регистрация заявки

Дата	Номер	Заявка	Клиент	Сотрудник	Приоритет	Статус	Этап выпол...
10.11.2023 22:00:47	00000001	Заправка картриджа (ООО "Альтаир")_10.11.23	ООО "Альтаир"		Высокий	В ожидании	Не выполнено
10.11.2023 22:01:40	00000002	Ремонт МФУ (Смирнов С.С.)_10.11.23		Смирнов С.С.	Средний	В ожидании	Не выполнено

24. Форма документа «Регистрация заявки».

Регистрация заявки 00000001 от 10.11.2023 22:00:47

Основное | Зарегистрированные заявки

Провести и закрыть | Записать | Провести | Создать на основании

Номер: 00000001

Дата: 10.11.2023 22:00:47

Заявка: Заправка картриджа (ООО "Альтаир")_10.11.23

Клиент: ООО "Альтаир"

Сотрудник:

Приоритет: Высокий

Статус: В ожидании

Этап выполнения: Не выполнено

Добавить

N	Оборудование	Проблема
1	Принтер	Закончился тонер

25. Список документов «Обработка заявки».

Обработка заявки

Дата	Номер	Заявка	Клиент	Сотрудник	Приоритет	Статус	Этап выполнения	Исполнитель
10.11.2023 22:05:44	00000001	Заправка картр...	ООО "Альтаир"		Высокий	В работе	Не выполнено	Иванов И.И.
10.11.2023 22:05:59	00000002	Ремонт МФУ (С...		Смирнов С.С.	Средний	В работе	Не выполнено	Петров П.П.

26. Форма документа «Обработка заявки».

Оценка запланированных результатов по ПМ.11

Результаты обучения: умения, знания и компетенции	Показатели оценки результата
Уметь:	
<p>У1 работать с современными case-средствами проектирования баз данных;</p> <p>У2 проектировать логическую и физическую схемы базы данных;</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p> <p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p> <p>ПК 11.1 Осуществлять сбор, обработку и анализ информации для проектирования баз данных</p> <p>ПК 11.2 Проектировать базу данных на основе анализа предметной области</p> <p>ПК 11.3 Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области</p>	<ul style="list-style-type: none"> – работает с изобразительными средствами, используемыми в ER- моделировании реляционной базы данных; – демонстрирует способности решения профессиональных задач практического содержания; – проектирует логическую и физическую схемы базы данных; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – использует современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности; – осуществляет сбор, обработку и анализ информации для проектирования баз данных. – проектирует базу данных на основе анализа предметной области; – разрабатывает объекты базы данных в соответствии с результатами анализа предметной области;

	<ul style="list-style-type: none"> – реализует базу данных в конкретной системе управления базами данных.
<p>У3 создавать хранимые процедуры и триггеры на базах данных;</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p> <p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке</p> <p>ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных</p> <p>ПК 11.5 Администрировать базы данных</p>	<ul style="list-style-type: none"> – умеет создавать хранимые процедуры и триггеры на базах данных; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке; – разрабатывает объекты базы данных в соответствии с результатами анализа предметной области; – реализует базу данных в конкретной системе управления базами данных; – выполняет задачи по администрированию базы данных.
<p>У4 применять стандартные методы для защиты объектов базы данных;</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p>	<ul style="list-style-type: none"> – применяет стандартные методы для защиты объектов базы данных; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие;

<p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p> <p>ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных</p> <p>ПК 11.5 Администрировать базы данных</p> <p>ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации</p>	<ul style="list-style-type: none"> – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке; – разрабатывает объекты базы данных в соответствии с результатами анализа предметной области; – реализует базу данных в конкретной системе управления базами данных; – выполняет задачи по администрированию базы данных; – обеспечивает защиту информации в базе данных с использованием различных технологий защиты информации.
<p>У5 выполнять стандартные процедуры резервного копирования и мониторинга выполнения этой процедуры;</p> <p>У6 выполнять процедуру восстановления базы данных и вести мониторинг выполнения этой процедуры;</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p> <p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p>	<ul style="list-style-type: none"> – выполняет стандартные процедуры резервного копирования и восстановления базы данных, ведет мониторинг выполнения этих процедур; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности;

<p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p> <p>ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных</p> <p>ПК 11.5 Администрировать базы данных</p> <p>ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации</p>	<ul style="list-style-type: none"> – пользуется профессиональной документацией на государственном и иностранном языке; – реализует базу данных в конкретной системе управления базами данных; – выполняет задачи по администрированию базы данных; – обеспечивает защиту информации в базе данных с использованием различных технологий защиты информации.
<p>У7 обеспечивать информационную безопасность на уровне базы данных.</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p> <p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.</p> <p>ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации</p>	<ul style="list-style-type: none"> – обеспечивает информационную безопасность на уровне базы данных; – выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам; – осуществляет поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности; – планирует и реализовывает собственное профессиональное и личностное развитие; – умеет работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами; – осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста; – использует информационные технологии в профессиональной деятельности; – пользуется профессиональной документацией на государственном и иностранном языке; – обеспечивает защиту информации в базе данных с использованием различных технологий защиты информации.
<p>Знать:</p>	
<p>З1 основные положения теории баз данных, хранилищ данных, баз знаний;</p>	<ul style="list-style-type: none"> – знает основные положения теории баз данных, хранилищ данных, баз знаний;

32 основные принципы структуризации и нормализации базы данных;	– знает основные принципы структуризации и нормализации базы данных;
33 основные принципы построения концептуальной, логической и физической модели данных;	– понимает основные принципы построения концептуальной, логической и физической модели данных;
34 методы описания схем баз данных в современных системах управления базами данных;	– освоил методы описания схем баз данных в современных системах управления базами данных;
35 структуры данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;	– знает структуры данных систем управления базами данных, общий подход к организации представлений, таблиц, индексов и кластеров;
36 методы организации целостности данных;	– знает методы организации целостности данных;
37 способы контроля доступа к данным и управления привилегиями;	– обеспечивает способы контроля доступа к данным и управления привилегиями;
38 основные методы и средства защиты данных в базах данных.	– имеет представление об основных методах и средствах защиты данных в базах данных.

Образец билета.

МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ краевое государственное автономное профессиональное образовательное учреждение «Дальнегорский индустриально-технологический колледж»		
<p style="text-align: center;">Утверждаю Заместитель директора</p> <hr/> <p style="text-align: center;">(Ф.И.О.)</p> <hr/> <p style="text-align: center;">(подпись)</p> <p style="text-align: center;">« ____ » _____ 20__ г.</p>	<p style="text-align: center;">Экзаменационный билет №1 по ПМ (индекс, название)</p> <p>Группа(ы) _____</p> <p>Специальность: код, название</p>	<p>Рассмотрено на заседании цикловой методической комиссии</p> <p>Председатель _____ (Ф.И.О.)</p> <hr/> <p style="text-align: center;">(подпись)</p> <p style="text-align: center;">« ____ » _____ 20__ г.</p>
<p>1. Практико-ориентированная задача «Разработка и реализация базы данных для учета заявок на ремонт оборудования» (составлена на основании оценочных материалов демонстрационного экзамена, утвержденных Протоколом заседания Педагогического совета ФГБОУ ДПО ИРПО от «3» августа 2023 г. № 8; шифр комплекта оценочной документации: КОД 09.02.07-2-2024).</p> <p>2. Кейс-ситуация с вариантами ответов. Обеспечить выбор верного варианта ответа и обосновать оптимальность выбора.</p>		

Критерии оценки ответов обучающихся.

Оценка по традиционной системе	Процент	Критерий
Отлично	от 85 %	Продемонстрирован высокий уровень знаний по всему объему изучаемого учебного курса; в полном объеме выполнены практико-ориентированные задания.
Хорошо	от 70% до 85 %	Продемонстрировано понимание основного содержания изучаемого междисциплинарного курса; выполнены практико-ориентированные задания.
Удовлетворительно	от 55 % до 70 %	Продемонстрировано неполное понимание основного содержания изучаемого междисциплинарного курса; частично выполнены практико-ориентированные задания.
Неудовлетворительно	менее 55 %	Не продемонстрировано владение знаниями; не выполнены практико-ориентированные задания.

3. Экзаменационная ведомость.