



**ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ
АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
КИЗИЛЮРТОВСКИЙ МНОГОПРОФИЛЬНЫЙ КОЛЛЕДЖ**

Российская Федерация
Республика Дагестан,
368118, г. Кизилюрт,
ул. Вишневого, 170.

Тел.: +7(989) 476-00-15
E- mail: omar.g4san@yandex.ru

ОДОБРЕНО
на педагогическом совете № 1
от «29» августа 2024г.

УТВЕРЖДЕНО
директор ПОАНО «КМК» г.Кизилюрт
О.М.Гасанов 
Приказ №2 -О
от «29» августа 2024г.



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения текущего контроля успеваемости, промежуточной атте-
стации обучающихся по учебной дисциплине
(профессиональному модулю)

**ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ**

по специальности 09.02.07 «Информационные системы и программирование»
по программе базовой подготовки
на базе основного общего образования;
форма обучения – очная
Квалификация выпускника – программист

г. Кизилюрт 2024г.



**ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ
АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
КИЗИЛ'ЮРТОВСКИЙ МНОГОПРОФИЛЬНЫЙ КОЛЛЕДЖ**

Российская Федерация
Республика Дагестан,
368118, г. Кизил'юрт,
ул. Вишневского, 170.

Тел.: +7(989) 476-00-15
E- mail: omar.g4san@yandex.ru

ОДОБРЕНО
на педагогическом совете № 1
от «29» августа 2024г.

УТВЕРЖДЕНО
директор ПОАНО «КМК» г.Кизил'юрт
О.М.Гасанов _____
Приказ №2 -О
от «29» августа 2024г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения текущего контроля успеваемости, промежуточной
аттестации обучающихся по учебной дисциплине
(профессиональному модулю)

ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

по специальности 09.02.07 «Информационные системы и программирование»
по программе базовой подготовки
на базе основного общего образования;
форма обучения – очная
Квалификация выпускника – программист

г. Кизил'юрт 2024г.

Содержание

1. Пояснительная записка.....
2. Перечень компетенций с указанием этапов их формирования в процессе освоения основной образовательной программы.....
3. Описание перечня оценочных средств и критериев оценивания компетенций на различных этапах их формирования.....
4. Оценочные средства характеризующих этапы формирования компетенций при изучении профессионального модуля.....
5. Критерии и шкала оценивания компетенций на различных этапах их формирования.....
6. Описание процедуры оценивания знаний и умений, характеризующих этапы формирования компетенций.....

1. Пояснительная записка

Оценочные материалы разработаны в форме фонда оценочных средств в соответствии с пунктом 9 статьи 2 Федерального закона от 29.12.2012 г. №273-ФЗ «Об образовании в Российской Федерации» и раздела II Методика расчета и применения аккредитационных показателей по образовательным программам среднего профессионального образования Приказа Минпросвещения России от 14.04.2023 №272 «Об утверждении аккредитационных показателей, методики расчета и применении аккредитационных показателей по образовательным программам среднего профессионального образования» и пункта 7 Приказа Министерства Просвещения Российской Федерации от 24.08.2022 №762 «Об утверждении порядка организации и осуществления образовательной деятельности по образовательным программам среднего профессионального образования».

Фонд оценочных средств (далее – ФОС) предназначены для оценки уровня освоения компетенций на различных этапах их формирования при изучении профессионального модуля ПМ.01 Разработка модулей программного обеспечения для компьютерных систем.

Общие компетенции:

Код компетенции	Формулировка компетенции	Знания, умения
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	<p>Умения: распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составить план действия; определить необходимые ресурсы; владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)</p> <p>Знания: актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности</p>

ОК 02	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	Умения: определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска
		Знания: номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;	Умения: определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования; рассчитывать размеры выплат по процентным ставкам кредитования; определять инвестиционную привлекательность коммерческих идей в рамках профессиональной деятельности
		Знания: содержание актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования; основы предпринимательской деятельности; основы финансовой грамотности; кредитные банковские продукты
ОК 04	Эффективно взаимодействовать и работать в коллективе и команде	Умения: организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности
		Знания: психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	Умения: грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе
		Знания: особенности социального и культурного контекста; правила оформления документов и построения устных сообщений.
ОК 06	Проявлять гражданско-	Умения: описывать значимость своей специальности

	патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	Знания: сущность гражданско-патриотической позиции, российских духовно-нравственных ценностей; значимость профессиональной деятельности по специальности
ОК 07	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	<p>Умения: соблюдать нормы экологической безопасности; определять направления ресурсосбережения в рамках профессиональной деятельности по специальности</p> <p>Знания: правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения ресурсосбережения</p>
ОК 08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	<p>Умения: использовать физкультурно-оздоровительную деятельность для укрепления здоровья, достижения жизненных и профессиональных целей; применять рациональные приемы двигательных функций в профессиональной деятельности; пользоваться средствами профилактики перенапряжения характерными для данной специальности</p> <p>Знания: роль физической культуры в общекультурном, профессиональном и социальном развитии человека; основы здорового образа жизни; условия профессиональной деятельности и зоны риска физического здоровья для специальности; средства профилактики перенапряжения</p>
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках.	Умения: понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы

		Знания: правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности
--	--	---

Профессиональные компетенции:

Основные виды деятельности	Код и формулировка компетенции	Показатели освоения компетенции
Разработка модулей программного обеспечения для компьютерных систем.	ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.	Практический опыт: Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.
		Умения: Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. Оформлять документацию на программные средства. Оценка сложности алгоритма.
		Знания: Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования. Актуальная нормативно-правовая база в области документирования алгоритмов.
	ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.	Практический опыт: Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля. Разрабатывать мобильные приложения.
		Умения: Создавать программу по разработанному алгоритму как отдельный модуль. Оформлять документацию на программные средства. Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.
		Знания: Основные этапы разработки

		<p>программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования. Знание API современных мобильных операционных систем.</p>
	ПК.1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.	<p>Практический опыт: Использовать инструментальные средства на этапе отладки программного продукта. Проводить тестирование программного модуля по определенному сценарию.</p> <p>Умения: Выполнять отладку и тестирование программы на уровне модуля. Оформлять документацию на программные средства. Применять инструментальные средства отладки программного обеспечения.</p> <p>Знания: Основные принципы отладки и тестирования программных продуктов. Инструментарий отладки программных продуктов.</p>
	ПК 1.4. Выполнять тестирование программных модулей.	<p>Практический опыт: Проводить тестирование программного модуля по определенному сценарию. Использовать инструментальные средства на этапе тестирования программного продукта.</p> <p>Умения: Выполнять отладку и тестирование программы на уровне модуля. Оформлять документацию на программные средства.</p> <p>Знания: Основные виды и принципы тестирования программных продуктов.</p>
	ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.	<p>Практический опыт: Анализировать алгоритмы, в том числе с применением инструментальных средств. Осуществлять рефакторинг и оптимизацию программного кода.</p> <p>Умения: Выполнять оптимизацию и рефакторинг программного кода. Работать с системой контроля версий.</p>

		Знания: Способы оптимизации и приемы рефакторинга. Инструментальные средства анализа алгоритма. Методы организации рефакторинга и оптимизации кода. Принципы работы с системой контроля версий.
	ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ.	Практический опыт: Разрабатывать мобильные приложения.
		Умения: Осуществлять разработку кода программного модуля на современных языках программирования. Оформлять документацию на программные средства.
		Знания: Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования.

2. Перечень компетенций с указанием этапов их формирования в процессе освоения основной образовательной программы

№ п/п	Контролируемые разделы, темы, модули	Код контролируемой компетенции	Наименование оценочного средства
МДК. 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ			
1.	Тема 1.1.1 Жизненный цикл ПО	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
2.	Тема 1.1.2 Структурное программирование	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
3.	Тема 1.1.3 Объектно-ориентированное программирование	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
4	Тема 1.1.4 Паттерны проектирования	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
5	Тема 1.1.5. Событийно-управляемое программирование	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
6	Тема 1.1.6 Оптимизация и рефакторинг кода	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
7	Тема 1.1.7 Разработка пользовательского интерфейса	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
8	Тема 1.1.8 Основы ADO.Net	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания

МДК.01.02. ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ			
1.	Тема 1.2.1 Отладка и тестирование программного обеспечения	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
2.	Тема 1.2.2 Командное тестирование	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
3.	Тема 1.2.3 Документирование	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
МДК.01.03. РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ			
1.	Тема 1.3.1 Основные платформы и языки разработки мобильных приложений	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
2.	Тема 1.3.2 Создание и тестирование модулей для мобильных приложений	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
МДК.01.04. СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ			
1.	Тема 1.4.1 Программирование на языке низкого уровня	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания
2.	Тема 1.4.2 Разработка драйверов устройств в Linux	ОК 01; ОК 02; ОК 03; ОК 04; ОК 05; ОК 06; ОК 07; ОК 08; ОК 09; ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	коллоквиум; тестирование; самостоятельная работа; практические задания

3. Описание перечня оценочных средств и критериев оценивания компетенций на различных этапах их формирования

№ п/п	Наименование оценочного средства	Характеристика оценочного средства	Представление оценочного средства в фонде
1	Реферат	Продукт самостоятельной работы студента, представляющий собой краткое изложение в письменном виде полученных результатов теоретического анализа определенной научной (учебно-исследовательской) темы, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее.	Темы рефератов
2	Доклад, сообщение	Продукт самостоятельной работы студента, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы	Темы докладов, сообщений
3	Собеседование	Средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.	Вопросы по темам/разделам дисциплины
4	Тестирование	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
5	Разноуровневые задачи и задания	Различают задачи: а) ознакомительного, позволяющие оценивать и диагностировать знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, узнавание объектов изучения в рамках определенного раздела дисциплины; б) репродуктивного уровня, позволяющие оценивать и диагностировать умения синтезировать, анализировать, обобщать фактический и теоретический материал с формулированием конкретных выводов, установлением	Комплект разноуровневых задач и заданий

		причинно-следственных связей; в) продуктивного уровня, позволяющие оценивать и диагностировать умения, интегрировать знания различных областей, аргументировать собственную точку зрения, выполнять проблемные задания.	
--	--	--	--

4.Оценочные средства, характеризующие этапы освоения компетенций при изучении профессионального модуля

Вопросы устного опроса

МДК.01.01. РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

1. Опишите основные этапы жизненного цикла программного обеспечения. Какие модели жизненного цикла (каскадная, итеративная, спиральная, Agile) вы знаете и в чём их особенности?
2. В чём суть структурного программирования? Назовите его основные принципы (модульность, использование базовых управляющих конструкций, отказ от goto) и приведите примеры управляющих конструкций.
3. Что такое объектно-ориентированное программирование? Перечислите основные принципы ООП (инкапсуляция, наследование, полиморфизм, абстракция) и дайте их краткую характеристику.
4. Что такое паттерны проектирования? Приведите примеры трёх паттернов (например, Singleton, Factory, Observer, Strategy) и опишите, для решения каких задач они применяются.
5. В чём заключается событийно-управляемое программирование? Как оно реализуется при разработке графических пользовательских интерфейсов (например, обработка нажатий кнопок, событий мыши)?
6. Что такое рефакторинг кода? Назовите основные приёмы рефакторинга (выделение метода, переименование, замена условного оператора полиморфизмом и др.) и объясните, когда его следует выполнять.
7. Какие подходы к оптимизации программного кода вы знаете? Какие инструменты (профилировщики, статические анализаторы) помогают выявить узкие места?
8. Назовите основные элементы пользовательского интерфейса. Какие принципы разработки удобного и интуитивно понятного интерфейса (юзабилити) следует соблюдать?
9. Что такое [ADO.NET](#) и для каких задач он используется? Опишите назначение основных объектов: Connection, Command, DataReader, DataAdapter, DataSet.
10. Как организовать подключение к базе данных и выполнение простого SQL-запроса с использованием [ADO.NET](#) на языке C#? Опишите

последовательность действий и приведите пример кода.

МДК.01.02. ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ

1. Что такое отладка программы? Какие методы и средства отладки (пошаговое выполнение, точки останова, просмотр переменных, логирование) вы знаете?
2. Какие виды тестирования программного обеспечения существуют? Охарактеризуйте модульное, интеграционное, системное и приёмочное тестирование.
3. Что такое тест-кейс (тестовый сценарий)? Из каких элементов он состоит (идентификатор, описание, предусловия, шаги, ожидаемый результат)? Приведите пример тест-кейса для функции расчёта скидки.
4. В чём заключается командное тестирование? Каковы его преимущества и отличия от индивидуального тестирования?
5. Какие инструменты автоматизированного тестирования вам известны? Приведите примеры для модульного тестирования (например, NUnit, JUnit, PyTest) и опишите их основные возможности.
6. Какие техники тестирования (методы «чёрного ящика», «белого ящика», основанные на потоках данных) вы знаете? В чём их различия и когда какая применяется?
7. Что такое отладочная информация? Как её использовать для эффективного поиска ошибок в программе?
8. Как документировать результаты тестирования? Что должно включать в себя итоговое заключение (отчёт) о тестировании программного модуля?
9. Опишите типовой процесс отладки при возникновении ошибки в работе программы: от воспроизведения ошибки до её устранения и проверки исправления.
10. Какие виды документации на программное обеспечение существуют? Какие требования предъявляются к руководству пользователя и руководству программиста?

МДК.01.03. РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

1. Какие основные платформы для разработки мобильных приложений вы знаете (Android, iOS, кроссплатформенные) и каковы их ключевые особенности?
2. Какие языки программирования используются для разработки приложений под Android (Java, Kotlin) и под iOS (Swift, Objective-C)? В чём их преимущества?
3. Что такое Android SDK? Какие основные компоненты входят в состав SDK и для чего они нужны?
4. Как создать простейший модуль (Activity) для Android-приложения?

Опишите основные шаги: создание класса, разметку, регистрацию в манифесте.

5. Какие принципы тестирования мобильных приложений следует учитывать? Какие виды тестов (модульные, интеграционные, UI-тесты) можно проводить?
6. Перечислите основные компоненты Android-приложения: Activity, Service, BroadcastReceiver, ContentProvider. Опишите назначение каждого.
7. Какие инструменты и методы используются для отладки мобильных приложений (Android Studio Debugger, логи, эмуляторы)? Как выявить утечки памяти?
8. В чём основные отличия разработки под iOS по сравнению с Android (экосистема, языки, IDE, способы распространения)?
9. Какие способы локального хранения данных в мобильных приложениях существуют (SharedPreferences, SQLite, файлы)? Опишите, когда какой метод целесообразно применять.
10. Что такое эмулятор и симулятор мобильных устройств? Для чего они используются и каковы их ограничения по сравнению с реальными устройствами?

МДК.01.04. СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

1. Что такое языки программирования низкого уровня? В чём их отличие от языков высокого уровня и для каких задач они применяются?
2. Какие основные конструкции языка ассемблера вы знаете? Приведите примеры команд пересылки данных, арифметических операций и переходов.
3. Дайте определение системного программирования. Какие задачи решаются с его помощью (разработка ОС, драйверов, компиляторов, утилит)?
4. Опишите основные этапы загрузки операционной системы. Какую роль в этом процессе играет загрузчик (bootloader)?
5. Что такое драйвер устройства? Какие функции он выполняет в операционной системе?
6. Как разработать простейший драйвер для Linux (модуль ядра)? Опишите основные шаги: создание модуля, функции init и exit, регистрация, сборка и загрузка модуля.
7. Какие функции ядра Linux используются для взаимодействия с аппаратными устройствами (работа с портами ввода-вывода, прерываниями, DMA)?
8. Какие средства и методы применяются для отладки драйверов в Linux (использование printk, отладчик ядра, трассировка)?
9. Как организовано взаимодействие между драйвером и пользовательским приложением в Linux (через файловые операции: открытие, чтение, запись, управление)?
10. Какие основные риски и сложности возникают при разработке драйверов (нестабильность системы, сложность отладки, проблемы с производительностью) и как их можно минимизировать?

Темы рефератов, докладов, сообщений

МДК.01.01. РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

1. Сравнительный анализ моделей жизненного цикла программного обеспечения: каскадная, итеративная, спиральная, Agile
2. Принципы структурного программирования и их реализация в современных языках
3. Основные парадигмы программирования: структурная, объектно-ориентированная, функциональная, логическая
4. Принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм, абстракция
5. Паттерны проектирования: классификация и примеры использования в разработке ПО
6. Паттерн Singleton: реализация, преимущества и недостатки
7. Паттерн Observer и его применение в событийно-управляемых системах
8. Паттерн Strategy: гибкая замена алгоритмов в программе
9. Событийно-управляемое программирование: принципы и реализация в GUI-приложениях
10. Рефакторинг программного кода: приёмы, инструменты, практика применения
11. Оптимизация программного кода: методы, профилирование, анализ производительности
12. Разработка пользовательского интерфейса: принципы юзабилити и современные тренды
13. [ADO.NET](#): архитектура, основные объекты и применение для доступа к данным
14. Подключение к базе данных и выполнение запросов с использованием [ADO.NET](#) на C#
15. Управление транзакциями в [ADO.NET](#): обеспечение целостности данных
16. Разработка многоуровневых приложений с использованием [ADO.NET](#)
17. Применение паттерна Repository для работы с данными в приложениях на C#

МДК.01.02. ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ

1. Методы и средства отладки программного обеспечения: инструменты и практики
2. Виды тестирования программного обеспечения: модульное, интеграционное, системное, приёмочное
3. Техники тестирования: метод «чёрного ящика», метод «белого ящика», тестирование на основе потоков данных

4. Разработка тестовых сценариев (тест-кейсов): структура, требования, примеры
5. Автоматизация тестирования: инструменты и фреймворки (JUnit, JUnit, PyTest)
6. Командное тестирование: преимущества, организация, роли участников
7. Документирование результатов тестирования: отчёты, заключения, рекомендации
8. Инструменты логирования и мониторинга для поиска и устранения ошибок
9. Процесс отладки: воспроизведение ошибки, локализация, устранение, проверка
10. Написание руководства пользователя и руководства программиста: структура и требования
11. Тестирование производительности: методы, инструменты, анализ результатов
12. Непрерывная интеграция и тестирование в процессе разработки ПО
13. Управление дефектами: классификация, жизненный цикл, инструменты отслеживания

МДК.01.03. РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

1. Сравнительный анализ мобильных платформ: Android, iOS, кроссплатформенные решения
2. Языки программирования для мобильных приложений: Java, Kotlin, Swift, Objective-C
3. Android SDK: структура, основные компоненты, инструменты разработки
4. Создание и жизненный цикл Activity в Android-приложениях
5. Сервисы (Services) в Android: виды, назначение, реализация
6. BroadcastReceiver и ContentProvider в Android: использование и примеры
7. Тестирование мобильных приложений: модульное, интеграционное, UI-тестирование
8. Отладка мобильных приложений: инструменты Android Studio, логирование, эмуляция
9. Управление памятью в мобильных приложениях: утечки, оптимизация, профилирование
10. Способы локального хранения данных в мобильных приложениях: SharedPreferences, SQLite, файлы
11. Разработка под iOS: особенности экосистемы, Swift, Xcode
12. Работа с сетью в мобильных приложениях: библиотеки, асинхронность, обработка ошибок
13. Разработка кроссплатформенных мобильных приложений с использованием Flutter или React Native
14. UI/UX-дизайн мобильных приложений: принципы, подходы, инструменты

МДК.01.04. СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

1. Языки программирования низкого уровня: ассемблер, машинный код, область применения
2. Основные команды и конструкции языка ассемблера: пересылка данных,

- арифметика, переходы
3. Системное программирование: задачи, инструменты, области применения
 4. Архитектура операционной системы Linux и её взаимодействие с аппаратным обеспечением
 5. Этапы загрузки операционной системы: BIOS, bootloader, ядро, init
 6. Разработка модулей ядра Linux: структура, функции init и exit, сборка, загрузка
 7. Создание простейшего драйвера для Linux: пошаговое руководство
 8. Взаимодействие драйвера с пользовательским приложением через файловые операции
 9. Отладка драйверов Linux: использование printk, отладчик ядра, трассировка
 10. Обработка прерываний и работа с портами ввода-вывода в драйверах Linux
 11. Прямой доступ к памяти (DMA) и его использование в драйверах устройств
 12. Управление памятью в ядре Linux: механизмы и функции для разработки драйверов
 13. Риски и сложности разработки драйверов: нестабильность, безопасность, производительность
 14. Архитектура драйверов Windows: WDM, KMDF, UMDF
 15. Сравнительный анализ разработки драйверов в Linux и Windows
 16. Разработка системных утилит и обслуживающих программ на языке Си
 17. Взаимодействие с оборудованием через системные вызовы в Linux

Указания по выполнению тестовых заданий

Типы заданий	Последовательность действий при выполнении заданий
Тестовые задания на установление соответствия	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответов ожидаются пары элементов. 2. Внимательно прочитать оба списка: иск 1 - вопросы, утверждения, факты, понятия и т.д. иск 2 - утверждение, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов. 4. Записать попарно буквы/цифры (в зависимости от задания) вариантов ответов (например, А1 или Б4)
Тестовые задания на установление последовательности	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответов ожидаются последовательность элементов. 2. Внимательно прочитать предложенные варианты ответов. 3. Построить верную последовательность из предложенных элементов. 4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности без пробелов и знаков препинания (например, БВА или 135)
Тестовые задания с выбором двух (и более) правильных ответов из перечня	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответов ожидаются два и более вариантов ответов, наиболее верных. 2. Внимательно прочитать предложенные варианты ответов. 3. Выбрать два и более вариантов ответов, наиболее верных. 4. Записать только номера/буквы выбранных вариантов ответов.

--	--

Тестовые задания на установление соответствия

МДК.01.01. РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

№	Содержание задания	Правильный ответ	Компетенция
1	<p>Установите соответствие между этапом жизненного цикла ПО и его описанием.</p> <p>Этап: А) Анализ требований; Б) Проектирование; В) Реализация (кодирование); Г) Тестирование.</p> <p>Описание: 1) Создание архитектуры и детальных спецификаций; 2) Написание исходного кода; 3) Проверка соответствия продукта требованиям; 4) Определение потребностей пользователей и функций системы.</p>	А–4, Б–1, В–2, Г–3	ПК 1.1, ОК 02
2	<p>Установите соответствие между принципом структурного программирования и его определением.</p> <p>Принцип: А) Модульность; Б) Базовые управляющие конструкции; В) Отказ от оператора goto; Г) Структурное проектирование.</p> <p>Определение: 1) Программа состоит из последовательности шагов, ветвлений и циклов; 2) Использование только трёх конструкций: следование, ветвление, цикл; 3) Разбиение программы на логически завершённые блоки; 4) Передача управления только через управляющие конструкции.</p>	А–3, Б–2, В–4, Г–1	ПК 1.1, ОК 01
3	<p>Установите соответствие между принципом ООП и его характеристикой.</p> <p>Принцип: А) Инкапсуляция; Б) Наследование; В) Полиморфизм; Г) Абстракция.</p> <p>Характеристика: 1) Возможность одного объекта выступать в разных ролях; 2) Соккрытие внутреннего состояния объекта; 3) Создание нового класса на основе существующего; 4) Выделение существенных характеристик объекта.</p>	А–2, Б–3, В–1, Г–4	ПК 1.2, ОК 01
4	<p>Установите соответствие между паттерном проектирования и его назначением.</p> <p>Паттерн: А) Singleton; Б) Factory; В) Observer;</p>	А–2, Б–3, В–1, Г–4	ПК 1.2, ОК 02

№	Содержание задания	Правильный ответ	Компетенция
	Г) Strategy. Назначение: 1) Уведомление зависимых объектов об изменении состояния; 2) Гарантирует единственный экземпляр класса; 3) Создание объектов без указания конкретных классов; 4) Выбор алгоритма во время выполнения.		
5	Установите соответствие между событием в GUI и его обработчиком. Событие: А) Нажатие кнопки; Б) Перемещение мыши; В) Нажатие клавиши; Г) Закрытие окна. Обработчик: 1) MouseMove; 2) KeyDown; 3) FormClosing; 4) Click.	А–4, Б–1, В–2, Г–3	ПК 1.2, ОК 01
6	Установите соответствие между приёмом рефакторинга и его описанием. Приём: А) Выделение метода; Б) Переименование; В) Замена условного оператора полиморфизмом; Г) Удаление дублирования. Описание: 1) Перенос функциональности в специализированные классы; 2) Замена условного оператора полиморфизмом; 3) Извлечение повторяющегося кода в отдельную функцию; 4) Присвоение осмысленных имён.	А–3, Б–4, В–2, Г–1	ПК 1.5, ОК 02
7	Установите соответствие между методом оптимизации кода и его действием. Метод: А) Профилирование; Б) Кэширование; В) Развёртывание цикла; Г) Оптимизация запросов к БД. Действие: 1) Сокращение времени выполнения цикла; 2) Выявление узких мест; 3) Сохранение результатов вычислений; 4) Ускорение доступа к данным.	А–2, Б–3, В–1, Г–4	ПК 1.5, ОК 01
8	Установите соответствие между элементом пользовательского интерфейса и его назначением. Элемент: А) Кнопка; Б) Текстовое поле; В) Флажок (CheckBox); Г) Радиокнопка (RadioButton). Назначение: 1) Выбор одного значения из списка; 2) Запуск действия; 3) Ввод и редактирование текста; 4) Выбор нескольких опций из нескольких.	А–2, Б–3, В–4, Г–1	ПК 1.2, ОК 05
9	Установите соответствие между	А–3, Б–1, В–4, Г–2	ПК 1.2, ОК 02

№	Содержание задания	Правильный ответ	Компетенция
	<p>объектом ADO.NET и его назначением.</p> <p>Объект: А) Connection; Б) Command; В) DataReader; Г) DataSet.</p> <p>Назначение: 1) Выполнение команд SQL; 2) Хранение данных в памяти с возможностью изменения; 3) Установка соединения с БД; 4) Последовательное чтение данных только вперёд.</p>		
10	<p>Установите соответствие между термином на русском языке и его английским эквивалентом.</p> <p>Русский: А) Класс; Б) Объект; В) Интерфейс; Г) Наследование.</p> <p>Английский: 1) Interface; 2) Class; 3) Object; 4) Inheritance.</p>	А–2, Б–3, В–1, Г–4	ОК 09
11	<p>Установите соответствие между инструментом отладки и его назначением.</p> <p>Инструмент: А) Отладчик (Debugger); Б) Трассировка; В) Логирование; Г) Профилировщик.</p> <p>Назначение: 1) Запись событий программы в файл; 2) Пошаговое выполнение и просмотр переменных; 3) Измерение времени выполнения и поиск узких мест; 4) Вывод последовательности вызовов функций.</p>	А–2, Б–4, В–1, Г–3	ПК 1.3
12	<p>Установите соответствие между видом тестирования и его объектом.</p> <p>Вид: А) Модульное; Б) Интеграционное; В) Системное.</p> <p>Объект: 1) Взаимодействие нескольких модулей; 2) Отдельная функция или класс; 3) Вся система в целом.</p>	А–2, Б–1, В–3	ПК 1.4
13	<p>Установите соответствие между мобильной платформой и средой разработки.</p> <p>Платформа: А) Android; Б) iOS; В) Кроссплатформенная (Xamarin).</p> <p>Среда разработки: 1) Xcode; 2) Android Studio; 3) Visual Studio с Xamarin.</p>	А–2, Б–1, В–3	ПК 1.6

МДК.01.02. ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ

№	Содержание задания	Правильный ответ	Компетенция
1	Установите соответствие между видом	А–2, Б–4, В–1, Г–3	ПК 1.4, ОК 02

№	Содержание задания	Правильный ответ	Компетенция
	<p>тестирования и его характеристикой.</p> <p>Вид: А) Модульное; Б) Интеграционное; В) Системное; Г) Приёмочное.</p> <p>Характеристика: 1) Проверка всей системы; 2) Проверка отдельных функций; 3) Проверка соответствия требованиям заказчика; 4) Проверка взаимодействия между модулями.</p>		
2	<p>Установите соответствие между методом отладки и его описанием.</p> <p>Метод: А) Пошаговое выполнение; Б) Точки останова; В) Просмотр переменных; Г) Логирование.</p> <p>Описание: 1) Вставка операторов вывода; 2) Исполнение оператор за оператором; 3) Остановка в заданном месте; 4) Наблюдение за значениями переменных.</p>	А–2, Б–3, В–4, Г–1	ПК 1.3, ОК 01
3	<p>Установите соответствие между элементом тест-кейса и его содержанием.</p> <p>Элемент: А) Идентификатор; Б) Описание; В) Шаги; Г) Ожидаемый результат.</p> <p>Содержание: 1) Шаги для выполнения; 2) Что должно получиться; 3) Краткое пояснение; 4) Уникальный номер.</p>	А–4, Б–3, В–1, Г–2	ПК 1.4, ОК 05
4	<p>Установите соответствие между техникой тестирования и её описанием.</p> <p>Техника: А) Чёрный ящик; Б) Белый ящик; В) Тестирование потоков данных; Г) Эквивалентное разбиение.</p> <p>Описание: 1) Основано на внутренней структуре; 2) Основано на внешних спецификациях; 3) Проверка путей данных; 4) Разделение входных данных на классы.</p>	А–2, Б–1, В–3, Г–4	ПК 1.4, ОК 02
5	<p>Установите соответствие между инструментом автоматизированного тестирования и его назначением.</p> <p>Инструмент: А) NUnit; Б) JUnit; В) PyTest; Г) Selenium.</p> <p>Назначение: 1) Тестирование JavaScript; 2) Модульное тестирование .NET; 3) Модульное тестирование Java; 4) Модульное тестирование Python.</p>	А–2, Б–3, В–4, Г–1	ПК 1.4, ОК 02
6	<p>Установите соответствие между ролью в командном тестировании и её задачей.</p> <p>Роль: А) Тестировщик; Б) Разработчик; В) Аналитик; Г) Тест-менеджер.</p> <p>Задача: 1) Разработка и выполнение тестов; 2)</p>	А–1, Б–2, В–4, Г–3	ОК 04, ОК 05

№	Содержание задания	Правильный ответ	Компетенция
	Написание модульных тестов; 3) Координация процесса; 4) Определение требований.		
7	<p>Установите соответствие между видом документации и её назначением.</p> <p>Вид: А) Руководство пользователя; Б) Руководство программиста; В) Техническое задание; Г) Инструкция по установке.</p> <p>Назначение: 1) Описание установки; 2) Инструкция по работе; 3) Описание архитектуры; 4) Документ с требованиями.</p>	А–2, Б–3, В–4, Г–1	ПК 1.2, ОК 05
8	<p>Установите соответствие между этапом процесса отладки и его действием.</p> <p>Этап: А) Воспроизведение ошибки; Б) Локализация ошибки; В) Устранение ошибки; Г) Проверка исправления.</p> <p>Действие: 1) Исправление кода; 2) Повторение действий; 3) Определение места ошибки; 4) Повторное тестирование.</p>	А–2, Б–3, В–1, Г–4	ПК 1.3, ОК 01
9	<p>Установите соответствие между термином и его определением.</p> <p>Термин: А) Дефект; Б) Ошибка; В) Сбой; Г) Отказ.</p> <p>Определение: 1) Тестирование пользователями; 2) Несоответствие требованиям; 3) Действие программиста; 4) Полная потеря работоспособности.</p>	А–2, Б–3, В–4, Г–1	ПК 1.4, ОК 02
10	<p>Установите соответствие между термином на русском языке и его английским эквивалентом.</p> <p>Русский: А) Тестирование; Б) Отладка; В) Тест-кейс; Г) Дефект.</p> <p>Английский: 1) Debugging; 2) Testing; 3) Defect; 4) Test case.</p>	А–2, Б–1, В–4, Г–3	ОК 09
11	<p>Установите соответствие между алгоритмической конструкцией и её описанием.</p> <p>Конструкция: А) Последовательность; Б) Ветвление; В) Цикл.</p> <p>Описание: 1) Выполнение блока несколько раз; 2) Выбор одного из путей по условию; 3) Выполнение команд одна за другой.</p>	А–3, Б–2, В–1	ПК 1.1
12	<p>Установите соответствие между приёмом рефакторинга и его целью.</p> <p>Приём: А) Выделение метода; Б) Переименование; В) Удаление дублирования.</p> <p>Цель: 1) Сделать код более читаемым; 2)</p>	А–3, Б–1, В–2	ПК 1.5

№	Содержание задания	Правильный ответ	Компетенция
	Устранить повторяющийся код; 3) Извлечь повторяющуюся логику в отдельную функцию.		
13	<p>Установите соответствие между особенностью тестирования мобильных приложений и её описанием.</p> <p>Особенность: А) Разные разрешения экрана; Б) Разные версии ОС; В) Сенсорное управление.</p> <p>Описание: 1) Необходимость тестирования на разных устройствах с разными диагоналями; 2) Проверка на разных версиях Android/iOS; 3) Тестирование жестов, касаний.</p>	А–1, Б–2, В–3	ПК 1.6

МДК.01.03. РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

№	Содержание задания	Правильный ответ	Компетенция
1	<p>Установите соответствие между мобильной платформой и её характеристикой.</p> <p>Платформа: А) Android; Б) iOS; В) Windows Mobile; Г) Xamarin.</p> <p>Характеристика: 1) Разработка на Swift; 2) Открытая платформа на Linux; 3) ОС от Microsoft; 4) Кроссплатформенная разработка на C#.</p>	А–2, Б–1, В–3, Г–4	ПК 1.6, ОК 02
2	<p>Установите соответствие между языком программирования и его областью применения в мобильной разработке.</p> <p>Язык: А) Java; Б) Kotlin; В) Swift; Г) JavaScript (React Native).</p> <p>Применение: 1) iOS; 2) Кроссплатформенная; 3) Android (основной); 4) Android (современный).</p>	А–3, Б–4, В–1, Г–2	ПК 1.6, ОК 01
3	<p>Установите соответствие между компонентом Android-приложения и его описанием.</p> <p>Компонент: А) Activity; Б) Service; В) BroadcastReceiver; Г) ContentProvider.</p> <p>Описание: 1) Фоновый процесс; 2) Получение широковеб-сообщений; 3) Предоставление данных другим приложениям; 4) Экран с</p>	А–4, Б–1, В–2, Г–3	ПК 1.6, ОК 02

№	Содержание задания	Правильный ответ	Компетенция
	пользовательским интерфейсом.		
4	<p>Установите соответствие между способом локального хранения данных и его описанием.</p> <p>Способ: А) SharedPreferences; Б) SQLite; В) Внутреннее хранилище; Г) Внешнее хранилище.</p> <p>Описание: 1) Хранение в таблицах; 2) Хранение в виде файлов; 3) Хранение настроек "ключ-значение"; 4) Хранение на съёмном носителе.</p>	А–3, Б–1, В–2, Г–4	ПК 1.2, ОК 02
5	<p>Установите соответствие между видом тестирования мобильных приложений и его описанием.</p> <p>Вид: А) Модульное; Б) Интеграционное; В) UI-тестирование; Г) Тестирование на устройстве.</p> <p>Описание: 1) Проверка взаимодействия компонентов; 2) Проверка отдельных классов; 3) Проверка интерфейса; 4) Проверка на реальном устройстве.</p>	А–2, Б–1, В–3, Г–4	ПК 1.4, ОК 01
6	<p>Установите соответствие между инструментом разработки и его назначением.</p> <p>Инструмент: А) Android Studio; Б) Xcode; В) ADB; Г) Logcat.</p> <p>Назначение: 1) Утилита для связи с устройством Android; 2) Среда разработки для Android; 3) Среда разработки для iOS; 4) Просмотр логов в Android.</p>	А–2, Б–3, В–1, Г–4	ПК 1.3, ОК 02
7	<p>Установите соответствие между способом отладки мобильного приложения и его описанием.</p> <p>Способ: А) Отладка на эмуляторе; Б) Отладка на реальном устройстве; В) Логирование через Logcat; Г) Удалённая отладка.</p> <p>Описание: 1) Использование реального телефона; 2) Имитация разных моделей; 3) Отслеживание выполнения через логи; 4) Отладка через Wi-Fi.</p>	А–2, Б–1, В–3, Г–4	ПК 1.3, ОК 01

№	Содержание задания	Правильный ответ	Компетенция
8	<p>Установите соответствие между состоянием Activity и его описанием.</p> <p>Состояние: А) Created; Б) Started; В) Resumed; Г) Paused.</p> <p>Описание: 1) Activity видна и активна; 2) Activity создана, но не отображена; 3) Activity видна, но не в фокусе; 4) Activity частично перекрыта.</p>	А–2, Б–3, В–1, Г–4	ПК 1.6, ОК 02
9	<p>Установите соответствие между термином на русском языке и его английским эквивалентом.</p> <p>Русский: А) Эмулятор; Б) Устройство; В) Симулятор; Г) Реальное устройство.</p> <p>Английский: 1) Device; 2) Simulator; 3) Emulator; 4) Real device.</p>	А–3, Б–1, В–2, Г–4	ОК 09
10	<p>Установите соответствие между этапом разработки мобильного приложения и его содержанием.</p> <p>Этап: А) Проектирование интерфейса; Б) Реализация; В) Тестирование; Г) Публикация.</p> <p>Содержание: 1) Написание кода; 2) Тестирование на устройствах; 3) Создание макетов; 4) Размещение в магазине.</p>	А–3, Б–1, В–2, Г–4	ПК 1.2, ОК 01
11	<p>Установите соответствие между алгоритмом и его применением в мобильном приложении.</p> <p>Алгоритм: А) Сортировка; Б) Поиск; В) Шифрование.</p> <p>Применение: 1) Поиск контактов в списке; 2) Упорядочивание списка товаров; 3) Защита данных пользователя.</p>	А–2, Б–1, В–3	ПК 1.1
12	<p>Установите соответствие между способом оптимизации мобильного приложения и его действием.</p> <p>Способ: А) Сжатие ресурсов; Б) Кэширование данных; В) Асинхронная загрузка.</p> <p>Действие: 1) Хранение часто используемых данных в памяти; 2) Уменьшение размера изображений и видео; 3) Загрузка данных в фоне без</p>	А–2, Б–1, В–3	ПК 1.5

№	Содержание задания	Правильный ответ	Компетенция
	блокировки интерфейса.		
13	<p>Установите соответствие между инструментом и его использованием в разработке мобильных приложений (добавлено для выравнивания).</p> <p>Инструмент: А) Git; Б) Jira; В) Firebase.</p> <p>Использование: 1) Управление версиями; 2) Отслеживание задач и багов; 3) Аналитика и облачные сервисы.</p>	А–1, Б–2, В–3	ОК 04, ОК 02

МДК.01.04. СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

№	Содержание задания	Правильный ответ	Компетенция
1	<p>Установите соответствие между командой ассемблера и её действием.</p> <p>Команда: А) MOV; Б) ADD; В) SUB; Г) JMP.</p> <p>Действие: 1) Сложение; 2) Пересылка данных; 3) Вычитание; 4) Безусловный переход.</p>	А–2, Б–1, В–3, Г–4	ПК 1.1, ОК 02
2	<p>Установите соответствие между этапом загрузки ОС и его описанием.</p> <p>Этап: А) POST; Б) BIOS; В) Bootloader; Г) Загрузка ядра.</p> <p>Описание: 1) Загрузка ядра; 2) Проверка аппаратных средств; 3) Прошивка начального запуска; 4) Программа, запускающая ядро.</p>	А–2, Б–3, В–4, Г–1	ПК 1.1, ОК 02
3	<p>Установите соответствие между функцией ядра Linux и её назначением.</p> <p>Функция: А) printk; Б) kmalloc; В) request_irq; Г) ioremap.</p> <p>Назначение: 1) Выделение памяти; 2) Вывод сообщений в лог; 3) Регистрация обработчика прерываний; 4) Отображение памяти.</p>	А–2, Б–1, В–3, Г–4	ПК 1.2, ОК 02
4	<p>Установите соответствие между компонентом драйвера и его описанием.</p> <p>Компонент: А) module_init; Б)</p>	А–3, Б–1, В–2, Г–4	ПК 1.2, ОК 05

№	Содержание задания	Правильный ответ	Компетенция
	<p>module_exit; В) file_operations; Г) MODULE_LICENSE.</p> <p>Описание: 1) Функция выгрузки; 2) Структура файловых операций; 3) Функция инициализации; 4) Указание лицензии.</p>		
5	<p>Установите соответствие между способом взаимодействия драйвера и приложения и его описанием.</p> <p>Способ: А) Через файловую систему (/dev); Б) Через системные вызовы; В) Через procfs; Г) Через интерфейс netlink.</p> <p>Описание: 1) Доступ как к файлу; 2) Использование ioctl; 3) Передача данных через специальные файлы; 4) Обмен между ядром и пользовательским пространством.</p>	А–1, Б–2, В–3, Г–4	ПК 1.2, ОК 01
6	<p>Установите соответствие между методом отладки драйверов и его описанием.</p> <p>Метод: А) printk; Б) kgdb; В) Ftrace; Г) Kprobes.</p> <p>Описание: 1) Пошаговое выполнение ядра; 2) Вывод сообщений в лог; 3) Трассировка вызовов; 4) Динамические точки останова.</p>	А–2, Б–1, В–3, Г–4	ПК 1.3, ОК 02
7	<p>Установите соответствие между риском при разработке драйверов и способом его минимизации.</p> <p>Риск: А) Сбой системы; Б) Утечка памяти; В) Неправильная обработка прерываний; Г) Ошибки синхронизации.</p> <p>Способ: 1) Статический анализ; 2) Проверка возвращаемых значений; 3) Использование kmemleak; 4) Применение механизмов блокировки.</p>	А–2, Б–3, В–1, Г–4	ПК 1.5, ОК 01
8	<p>Установите соответствие между термином и его определением (системное программирование).</p> <p>Термин: А) Драйвер; Б) Прерывание; В) Bootloader; Г) Ядро ОС.</p> <p>Определение: 1) Загрузка ядра; 2) Программа управления устройством; 3)</p>	А–2, Б–3, В–1, Г–4	ПК 1.1, ОК 02

№	Содержание задания	Правильный ответ	Компетенция
	Сигнал процессору; 4) Центральная часть ОС.		
9	<p>Установите соответствие между системным вызовом Linux и его действием.</p> <p>Вызов: А) read; Б) write; В) open; Г) close.</p> <p>Действие: 1) Открытие файла; 2) Чтение данных; 3) Запись данных; 4) Закрытие дескриптора.</p>	А–2, Б–3, В–1, Г–4	ПК 1.2, ОК 02
10	<p>Установите соответствие между термином на русском языке и его английским эквивалентом.</p> <p>Русский: А) Ядро; Б) Драйвер; В) Прерывание; Г) Загрузчик.</p> <p>Английский: 1) Driver; 2) Interrupt; 3) Bootloader; 4) Kernel.</p>	А–4, Б–1, В–2, Г–3	ОК 09
11	<p>Установите соответствие между видом тестирования драйвера и его описанием.</p> <p>Вид: А) Модульное тестирование драйвера; Б) Интеграционное тестирование драйвера; В) Тестирование производительности.</p> <p>Описание: 1) Проверка работы драйвера с оборудованием; 2) Проверка отдельных функций драйвера изолированно; 3) Измерение скорости передачи данных.</p>	А–2, Б–1, В–3	ПК 1.4
12	<p>Установите соответствие между мобильной ОС и её архитектурной особенностью.</p> <p>ОС: А) Android; Б) iOS; В) Windows Mobile.</p> <p>Особенность: 1) Ядро Linux; 2) Ядро XNU (Unix-подобное); 3) Ядро Windows NT.</p>	А–1, Б–2, В–3	ПК 1.6
13	<p>Установите соответствие между способом оптимизации системного кода и его описанием (добавлено для выравнивания).</p> <p>Способ: А) Использование кэша; Б) Оптимизация циклов; В) Уменьшение числа системных вызовов.</p>	А–2, Б–3, В–1	ПК 1.5

№	Содержание задания	Правильный ответ	Компетенция
	Описание: 1) Снижение накладных расходов на переключение контекста; 2) Ускорение доступа к часто используемым данным; 3) Уменьшение времени выполнения за счёт правильной организации циклов.		

Тестовые задания на установление последовательности

МДК.01.01. РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

Тестовые задания на установление последовательности

№	Содержание задания	Правильный ответ (порядок номеров)	Компетенция
1	Установите последовательность этапов жизненного цикла программного обеспечения (каскадная модель): 1) Тестирование и отладка; 2) Анализ и определение требований; 3) Эксплуатация и сопровождение; 4) Кодирование (реализация); 5) Проектирование архитектуры.	2, 5, 4, 1, 3	ПК 1.1, ОК 02
2	Установите последовательность действий при создании консольного приложения на C# (без использования ООП): 1) Написать код с использованием управляющих конструкций; 2) Запустить компиляцию и проверить результат; 3) Создать проект в среде разработки; 4) Добавить необходимые пространства имён и определить точку входа (Main); 5) Протестировать программу на разных входных данных.	3, 4, 1, 2, 5	ПК 1.2, ОК 01
3	Установите последовательность шагов при реализации наследования в ООП: 1) Создать объект производного класса; 2) Создать конструктор производного класса; 3) Создать базовый класс с нужными полями и методами; 4) Вызвать методы базового класса через объект производного;	3, 5, 2, 1, 4	ПК 1.2, ОК 01

№	Содержание задания	Правильный ответ (порядок номеров)	Компетенция
	5) Объявить производный класс, указав базовый.		
4	<p>Установите последовательность действий при применении паттерна Singleton (одиночка) на C#:</p> <p>1) Создать приватный статический экземпляр класса;</p> <p>2) Добавить публичный статический метод (или свойство) для доступа к экземпляру;</p> <p>3) Сделать конструктор класса приватным;</p> <p>4) Проверить, что создаётся только один объект при многократном обращении;</p> <p>5) В методе доступа проверять, создан ли экземпляр, и при необходимости создавать его.</p>	3, 1, 5, 2, 4	ПК 1.2, ОК 02
5	<p>Установите последовательность этапов разработки пользовательского интерфейса (Windows Forms):</p> <p>1) Разместить элементы управления на форме и настроить их свойства;</p> <p>2) Создать проект Windows Forms;</p> <p>3) Протестировать работу интерфейса;</p> <p>4) Написать обработчики событий для элементов;</p> <p>5) Спроектировать макет формы (расположение кнопок, полей, меток).</p>	2, 5, 1, 4, 3	ПК 1.2, ОК 05
6	<p>Установите последовательность действий при рефакторинге метода (выделение метода):</p> <p>1) Создать новый метод с понятным именем;</p> <p>2) Заменить выделенный фрагмент вызовом нового метода;</p> <p>3) Найти повторяющийся или слишком длинный участок кода;</p> <p>4) Перенести фрагмент кода в новый метод;</p> <p>5) Проверить работоспособность программы после замены.</p>	3, 1, 4, 2, 5	ПК 1.5, ОК 01
7	Установите последовательность шагов при	5, 1, 3, 2, 4	ПК 1.5, ОК 02

№	Содержание задания	Правильный ответ (порядок номеров)	Компетенция
	<p>оптимизации кода с использованием профилирования:</p> <ol style="list-style-type: none"> 1) Запустить профилировщик и выполнить программу; 2) Изменить код, чтобы устранить узкое место; 3) Выявить наиболее затратные функции или операции; 4) Снова запустить профилировку, чтобы убедиться в улучшении; 5) Определить цель оптимизации (например, время выполнения или память). 		
8	<p>Установите последовательность действий при подключении к базе данных через ADO.NET (SqlConnection):</p> <ol style="list-style-type: none"> 1) Открыть соединение с помощью метода Open(); 2) Создать объект SqlConnection с указанием строки подключения; 3) Выполнить команду (например, SqlCommand); 4) Закрыть соединение методом Close(); 5) Обработать полученные данные (например, через SqlDataReader). 	2, 1, 3, 5, 4	ПК 1.2, ОК 02
9	<p>Установите последовательность этапов отладки программы с использованием точек останова:</p> <ol style="list-style-type: none"> 1) Запустить программу в режиме отладки; 2) Проанализировать значения переменных в окне наблюдения; 3) Установить точки останова в подозрительных местах; 4) Пошагово выполнить код до проблемного участка; 5) Исправить ошибку и продолжить выполнение. 	3, 1, 4, 2, 5	ПК 1.3, ОК 01
10	<p>Установите последовательность действий при работе с системой контроля версий Git (создание коммита):</p> <ol style="list-style-type: none"> 1) Выполнить команду git commit -m "message"; 2) Выполнить команду git add . 	3, 4, 2, 1, 5	ОК 04, ОК 09

№	Содержание задания	Правильный ответ (порядок номеров)	Компетенция
	(добавить изменения); 3) Создать репозиторий (git init) или клонировать существующий; 4) Внести изменения в файлы; 5) Выполнить команду git push для отправки на удалённый репозиторий (по желанию).		

МДК.01.02. ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ

Тестовые задания на установление последовательности

№	Содержание задания	Правильный ответ (порядок номеров)	Компетенция
1	Установите последовательность этапов тестирования по классическому V-образному подходу (сверху вниз): 1) Модульное тестирование; 2) Приёмочное тестирование; 3) Интеграционное тестирование; 4) Системное тестирование.	1, 3, 4, 2	ПК 1.4, ОК 02
2	Установите последовательность действий при создании тест-кейса для функции "расчёт скидки": 1) Написать шаги тест-кейса; 2) Определить ожидаемый результат; 3) Сформулировать цель тест-кейса; 4) Выполнить тест-кейс и записать фактический результат; 5) Задать входные данные (например, сумму покупки и категорию клиента).	3, 5, 1, 2, 4	ПК 1.4, ОК 05
3	Установите последовательность шагов при использовании отладчика (пошаговое выполнение): 1) Установить точку останова в коде; 2) Выполнять код шаг за шагом, отслеживая значения переменных; 3) Запустить программу в режиме отладки; 4) При достижении точки останова проанализировать состояние программы; 5) Исправить обнаруженную ошибку и продолжить отладку.	1, 3, 4, 2, 5	ПК 1.3, ОК 01

№	Содержание задания	Правильный ответ (порядок номеров)	Компетенция
4	<p>Установите последовательность этапов написания модульных тестов (например, с использованием NUnit):</p> <ol style="list-style-type: none"> 1) Запустить тесты и проанализировать результаты; 2) Написать тестовые методы с атрибутами [Test]; 3) Подключить библиотеку для тестирования; 4) Создать класс с тестами; 5) Определить тестовые данные и ожидаемые результаты. 	3, 4, 5, 2, 1	ПК 1.4, ОК 02
5	<p>Установите последовательность действий при документировании результатов тестирования:</p> <ol style="list-style-type: none"> 1) Записать фактический результат выполнения тест-кейса; 2) Сравнить фактический результат с ожидаемым; 3) Выполнить тест-кейс; 4) Указать статус (PASS/FAIL); 5) Составить итоговый отчёт о тестировании. 	3, 1, 2, 4, 5	ПК 1.4, ОК 05
6	<p>Установите последовательность этапов отладки ошибки, обнаруженной в логах:</p> <ol style="list-style-type: none"> 1) Воспроизвести ошибку в тестовой среде; 2) Прочитать сообщение об ошибке и стек вызовов; 3) Исправить код; 4) Установить точки останова в подозрительных местах; 5) Проверить исправление повторным запуском. 	2, 1, 4, 3, 5	ПК 1.3, ОК 01
7	<p>Установите последовательность действий при командном тестировании (роли и взаимодействие):</p> <ol style="list-style-type: none"> 1) Обсудить план тестирования и распределить обязанности; 2) Провести совместный обзор результатов; 3) Каждый участник выполняет свои тесты; 4) Назначить руководителя 	4, 1, 3, 2, 5	ОК 04, ОК 05

№	Содержание задания	Правильный ответ (порядок номеров)	Компетенция
	тестирования; 5) Подготовить сводный отчёт.		
8	Установите последовательность действий при использовании инструментов автоматизации тестирования (например, Selenium): 1) Написать скрипт автоматизации; 2) Установить необходимые драйверы и библиотеки; 3) Запустить автоматические тесты; 4) Создать проект и настроить окружение; 5) Проанализировать результаты тестирования.	2, 4, 1, 3, 5	ПК 1.4, ОК 02
9	Установите последовательность этапов приёмочного тестирования: 1) Подготовить тестовые сценарии, основанные на требованиях заказчика; 2) Выполнить тесты; 3) Согласовать критерии приёмки; 4) Составить акт приёмки; 5) Передать результат заказчику.	3, 1, 2, 4, 5	ПК 1.4, ОК 05
10	Установите последовательность шагов при использовании Git для командной работы (Pull Request): 1) Создать ветку для новой функциональности; 2) Выполнить коммиты в свою ветку; 3) Создать Pull Request в основную ветку; 4) Получить ревью и внести правки; 5) Влить изменения после одобрения.	1, 2, 3, 4, 5	ОК 04, ОК 09

МДК.01.03. РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Тестовые задания на установление последовательности

№	Содержание задания	Правильный ответ	Компетенция
1	Установите соответствие между мобильной платформой и её характеристикой. Платформа: А) Android; Б) iOS; В) Windows Mobile; Г) Xamarin. Характеристика: 1) Разработка на Swift; 2)	А–2, Б–1, В–3, Г–4	ПК 1.6, ОК 02

№	Содержание задания	Правильный ответ	Компетенция
	Открытая платформа на Linux; 3) ОС от Microsoft; 4) Кроссплатформенная разработка на C#.		
2	<p>Установите соответствие между языком программирования и его областью применения в мобильной разработке.</p> <p>Язык: А) Java; Б) Kotlin; В) Swift; Г) JavaScript (React Native).</p> <p>Применение: 1) iOS; 2) Кроссплатформенная; 3) Android (основной); 4) Android (современный).</p>	А–3, Б–4, В–1, Г–2	ПК 1.6, ОК 01
3	<p>Установите соответствие между компонентом Android-приложения и его описанием.</p> <p>Компонент: А) Activity; Б) Service; В) BroadcastReceiver; Г) ContentProvider.</p> <p>Описание: 1) Фоновый процесс; 2) Получение широковеб-сообщений; 3) Предоставление данных другим приложениям; 4) Экран с пользовательским интерфейсом.</p>	А–4, Б–1, В–2, Г–3	ПК 1.6, ОК 02
4	<p>Установите соответствие между способом локального хранения данных и его описанием.</p> <p>Способ: А) SharedPreferences; Б) SQLite; В) Внутреннее хранилище; Г) Внешнее хранилище.</p> <p>Описание: 1) Хранение в таблицах; 2) Хранение в виде файлов; 3) Хранение настроек "ключ-значение"; 4) Хранение на съёмном носителе.</p>	А–3, Б–1, В–2, Г–4	ПК 1.2, ОК 02
5	<p>Установите соответствие между видом тестирования мобильных приложений и его описанием.</p> <p>Вид: А) Модульное; Б) Интеграционное; В) UI-тестирование; Г) Тестирование на устройстве.</p> <p>Описание: 1) Проверка взаимодействия компонентов; 2) Проверка отдельных классов; 3) Проверка интерфейса; 4) Проверка на реальном устройстве.</p>	А–2, Б–1, В–3, Г–4	ПК 1.4, ОК 01
6	Установите правильную последовательность действий при отладке мобильного	1, 2, 3, 4, 5, 6	ПК 1.3, ОК 01

№	Содержание задания	Правильный ответ	Компетенция
	<p>приложения на реальном устройстве с использованием Android Studio:</p> <p>1) Подключить устройство к компьютеру через USB-кабель и включить режим отладки (USB Debugging) в настройках разработчика.</p> <p>2) Запустить приложение в режиме отладки (нажать кнопку Debug) и выбрать подключённое устройство в качестве целевого.</p> <p>3) Установить точки останова в нужных местах кода.</p> <p>4) Выполнить действия в приложении, которые приводят к воспроизведению ошибки.</p> <p>5) Когда выполнение остановится на точке останова, проанализировать значения переменных и состояние стека вызовов.</p> <p>6) Исправить ошибку в коде, перезапустить отладку и проверить, устранена ли проблема.</p>		
7	<p>Установите соответствие между способом отладки мобильного приложения и его описанием.</p> <p>Способ: А) Отладка на эмуляторе; Б) Отладка на реальном устройстве; В) Логирование через Logcat; Г) Удалённая отладка.</p> <p>Описание: 1) Использование реального телефона; 2) Имитация разных моделей; 3) Отслеживание выполнения через логи; 4) Отладка через Wi-Fi.</p>	А–2, Б–1, В–3, Г–4	ПК 1.3, ОК 01
8	<p>Установите соответствие между состоянием Activity и его описанием.</p> <p>Состояние: А) Created; Б) Started; В) Resumed; Г) Paused.</p> <p>Описание: 1) Activity видна и активна; 2) Activity создана, но не отображена; 3) Activity видна, но не в фокусе; 4) Activity частично перекрыта.</p>	А–2, Б–3, В–1, Г–4	ПК 1.6, ОК 02
9	<p>Установите соответствие между термином на русском языке и его английским эквивалентом.</p> <p>Русский: А) Эмулятор; Б) Устройство; В) Симулятор; Г) Реальное устройство.</p>	А–3, Б–1, В–2, Г–4	ОК 09

№	Содержание задания	Правильный ответ	Компетенция
	Английский: 1) Device; 2) Simulator; 3) Emulator; 4) Real device.		
10	<p>Установите правильную последовательность этапов публикации мобильного приложения в Google Play (Android):</p> <p>1) Подготовить иконку приложения, скриншоты, описание и другие маркетинговые материалы.</p> <p>2) Зарегистрировать учётную запись разработчика в Google Play Console и оплатить регистрационный взнос.</p> <p>3) Собрать финальную подписанную версию APK или AAB (Android App Bundle) с помощью ключа подписи.</p> <p>4) Создать карточку приложения в консоли, заполнить все поля (название, категория, возрастные ограничения).</p> <p>5) Загрузить подготовленный APK/AAB в раздел «Релизы» и заполнить информацию о версии.</p> <p>6) Отправить приложение на проверку (рецензирование) в Google Play.</p> <p>7) После одобрения опубликовать приложение в магазине.</p>	2, 4, 1, 3, 5, 6, 7	ПК 1.6, ОК 02, ОК 05
11	<p>Установите соответствие между алгоритмом и его применением в мобильном приложении.</p> <p>Алгоритм: А) Сортировка; Б) Поиск; В) Шифрование.</p> <p>Применение: 1) Поиск контактов в списке; 2) Упорядочивание списка товаров; 3) Защита данных пользователя.</p>	А–2, Б–1, В–3	ПК 1.1, ОК 01
12	<p>Установите соответствие между способом оптимизации мобильного приложения и его действием.</p> <p>Способ: А) Сжатие ресурсов; Б) Кэширование данных; В) Асинхронная загрузка.</p> <p>Действие: 1) Хранение часто используемых данных в памяти; 2) Уменьшение размера изображений и видео; 3) Загрузка данных в фоне без блокировки интерфейса.</p>	А–2, Б–1, В–3	ПК 1.5, ОК 02
13	Установите соответствие между инструментом и его использованием в разработке мобильных приложений.	А–1, Б–2, В–3	ОК 04, ОК 02

№	Содержание задания	Правильный ответ	Компетенция
	Инструмент: А) Git; Б) Jira; В) Firebase. Использование: 1) Управление версиями; 2) Отслеживание задач и багов; 3) Аналитика и облачные сервисы.		

МДК.01.04. СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Тестовые задания на установление последовательности

№	Содержание задания	Правильный ответ (порядок номеров)	Компетенция
1	Установите последовательность этапов загрузки операционной системы Linux: 1) Загрузка ядра; 2) POST (Power-On Self Test); 3) Запуск init процесса; 4) Работа BIOS/UEFI; 5) Загрузка Bootloader (GRUB).	2, 4, 5, 1, 3	ПК 1.1, ОК 02
2	Установите последовательность шагов при создании простейшего модуля ядра Linux: 1) Написать функции init и exit; 2) Написать Makefile для сборки; 3) Скомпилировать модуль; 4) Загрузить модуль с помощью insmod; 5) Определить лицензию (MODULE_LICENSE).	1, 5, 2, 3, 4	ПК 1.2, ОК 02
3	Установите последовательность действий при отладке драйвера с помощью printk: 1) Вставить printk в нужные места кода; 2) Скомпилировать и загрузить драйвер; 3) Проверить вывод сообщений в системном логге (dmesg); 4) Проанализировать сообщения и найти ошибку; 5) Исправить код и повторить.	1, 2, 3, 4, 5	ПК 1.3, ОК 01
4	Установите последовательность этапов обработки прерывания в драйвере:	2, 3, 1, 5, 4	ПК 1.2, ОК 02

№	Содержание задания	Правильный ответ (порядок номеров)	Компетенция
	1) Выполнить обработчик прерывания; 2) Зарегистрировать обработчик (request_irq); 3) При возникновении прерывания аппаратное устройство сигнализирует; 4) Освободить обработчик при выгрузке драйвера (free_irq); 5) Сообщить об окончании обработки.		
5	Установите последовательность шагов при взаимодействии с устройством через файловую операцию read: 1) Пользовательское приложение вызывает read; 2) Драйвер обрабатывает вызов в своей функции read; 3) Ядро передаёт управление драйверу; 4) Драйвер читает данные из устройства; 5) Данные возвращаются в пользовательское приложение.	1, 3, 2, 4, 5	ПК 1.2, ОК 01
6	Установите последовательность действий при создании системного вызова в ядре Linux: 1) Добавить номер системного вызова в таблицу; 2) Написать реализацию системного вызова; 3) Перекомпилировать ядро; 4) Создать прототип в заголовочном файле; 5) Проверить работу системного вызова из пользовательского приложения.	2, 4, 1, 3, 5	ПК 1.1, ОК 02
7	Установите последовательность шагов при настройке компиляции модуля ядра с помощью Makefile: 1) Указать путь к исходникам ядра; 2) Определить имя модуля (obj-m); 3) Запустить make в каталоге модуля; 4) Проверить наличие	2, 1, 3, 4, 5	ПК 1.2, ОК 02

№	Содержание задания	Правильный ответ (порядок номеров)	Компетенция
	скомпилированного .ko файла; 5) Загрузить модуль.		
8	Установите последовательность этапов тестирования производительности драйвера (измерение скорости): 1) Запустить тестовую программу, использующую драйвер; 2) Измерить время выполнения операций (чтение/запись); 3) Подключить профилировщик; 4) Проанализировать результаты; 5) Оптимизировать код драйвера при необходимости.	3, 1, 2, 4, 5	ПК 1.5, ОК 02
9	Установите последовательность действий при возникновении ошибки в драйвере (kernel panic): 1) Записать стек вызовов из сообщения об ошибке; 2) Проанализировать код в месте ошибки; 3) Исправить ошибку; 4) Перекомпилировать драйвер; 5) Загрузить исправленный драйвер.	1, 2, 3, 4, 5	ПК 1.3, ОК 01
10	Установите последовательность шагов при использовании утилиты strace для трассировки системных вызовов: 1) Запустить программу с strace; 2) Проанализировать список системных вызовов и их аргументы; 3) Выполнить команду strace -e trace=open программу; 4) Отфильтровать нужные вызовы; 5) Сделать вывод о взаимодействии с ядром.	3, 1, 2, 4, 5 (обычно сначала команда, затем анализ)	ПК 1.3, ОК 02

Тестовые задания с выбором двух (и более) правильных ответов

МДК.01.01. РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

№	Содержание задания	Правильные ответы	Компетенция
----------	---------------------------	------------------------------	--------------------

№	Содержание задания	Правильные ответы	Компетенция
1	Какие из перечисленных этапов входят в жизненный цикл программного обеспечения? 1) анализ требований; 2) проектирование; 3) кодирование; 4) продажа компьютеров; 5) тестирование.	1, 2, 3, 5	ОК 02, ПК 1.1
2	Какие принципы относятся к структурному программированию? 1) использование только трёх базовых конструкций (следование, ветвление, цикл); 2) отказ от оператора goto; 3) разбиение программы на функции; 4) использование глобальных переменных везде; 5) использование наследования классов.	1, 2, 3	ОК 01, ПК 1.1
3	Какие из перечисленных принципов относятся к ООП? 1) инкапсуляция; 2) наследование; 3) полиморфизм; 4) рекурсия; 5) абстракция.	1, 2, 3, 5	ОК 02, ПК 1.2
4	Какие из перечисленных паттернов проектирования существуют? 1) Singleton; 2) Factory; 3) Observer; 4) Loop; 5) Strategy.	1, 2, 3, 5	ОК 01, ПК 1.2
5	Какие события в Windows Forms можно обработать для кнопки? 1) Click; 2) MouseEnter; 3) KeyDown; 4) DatabaseConnect; 5) MouseMove.	1, 2, 3, 5	ОК 02, ПК 1.2
6	Какие приёмы рефакторинга кода существуют? 1) выделение метода; 2) переименование переменных; 3) удаление дублирующегося кода; 4) добавление комментариев ко всем строкам; 5) замена условного оператора полиморфизмом.	1, 2, 3, 5	ПК 1.5
7	Какие методы оптимизации кода вы знаете? 1) профилирование; 2) кэширование;	1, 2, 3, 5	ПК 1.5

№	Содержание задания	Правильные ответы	Компетенция
	3) развёртывание цикла; 4) удаление всех комментариев; 5) использование быстрых библиотек.		
8	Какие элементы пользовательского интерфейса используются для ввода текста? 1) TextBox; 2) RichTextBox; 3) Button; 4) Label; 5) MaskedTextBox.	1, 2, 5	ОК 02, ПК 1.2
9	Какие объекты ADO.NET используются для работы с базой данных? 1) Connection; 2) Command; 3) DataReader; 4) DataSet; 5) PaintBox.	1, 2, 3, 4	ПК 1.2
10	Какие из перечисленных понятий относятся к событийно-управляемому программированию? 1) обработчик события; 2) источник события; 3) подписка на событие; 4) наследование; 5) делегат.	1, 2, 3, 5	ОК 01, ПК 1.2

МДК.01.02. ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ

№	Содержание задания	Правильные ответы	Компетенция
1	Какие виды тестирования существуют? 1) модульное; 2) интеграционное; 3) системное; 4) приёмочное; 5) кулинарное.	1, 2, 3, 4	ОК 02, ПК 1.4
2	Какие методы отладки используются в программировании? 1) пошаговое выполнение; 2) точки останова; 3) просмотр переменных; 4) логирование; 5) удаление кода.	1, 2, 3, 4	ПК 1.3

№	Содержание задания	Правильные ответы	Компетенция
3	Какие элементы включает тест-кейс? 1) идентификатор; 2) описание; 3) шаги выполнения; 4) ожидаемый результат; 5) список покупок.	1, 2, 3, 4	ПК 1.4, ПК 1.5
4	Какие техники тестирования вы знаете? 1) чёрный ящик; 2) белый ящик; 3) тестирование потоков данных; 4) эквивалентное разбиение; 5) тестирование на скорость.	1, 2, 3, 4	ОК 02, ПК 1.4
5	Какие инструменты используются для модульного тестирования? 1) NUnit; 2) JUnit; 3) PyTest; 4) Selenium; 5) MS Word.	1, 2, 3	ПК 1.4
6	Какие роли существуют в командном тестировании? 1) тестировщик; 2) разработчик; 3) аналитик; 4) тест-менеджер; 5) бухгалтер.	1, 2, 3, 4	ОК 04, ОК 05
7	Какие виды документации относятся к программному обеспечению? 1) руководство пользователя; 2) руководство программиста; 3) техническое задание; 4) инструкция по установке; 5) меню столовой.	1, 2, 3, 4	ПК 1.2, ОК 05
8	Какие этапы включает процесс отладки? 1) воспроизведение ошибки; 2) локализация ошибки; 3) устранение ошибки; 4) проверка исправления; 5) удаление всех тестов.	1, 2, 3, 4	ПК 1.3, ОК 01
9	Какие термины относятся к тестированию? 1) дефект; 2) ошибка; 3) сбой; 4) отказ; 5) проект.	1, 2, 3, 4	ПК 1.4, ОК 02

№	Содержание задания	Правильные ответы	Компетенция
10	Какие утверждения о тестировании верны? 1) тестирование помогает найти ошибки; 2) тестирование гарантирует полное отсутствие ошибок; 3) тестирование бывает ручным и автоматизированным; 4) тестирование проводится только после выпуска программы; 5) тестирование — часть жизненного цикла ПО.	1, 3, 5	ОК 01, ПК 1.4

МДК.01.03. РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

№	Содержание задания	Правильные ответы	Компетенция
1	Какие мобильные платформы существуют? 1) Android; 2) iOS; 3) Windows Phone (Mobile); 4) Linux Desktop; 5) HarmonyOS.	1, 2, 3, 5	ОК 02, ПК 1.6
2	Какие языки программирования используются для Android? 1) Java; 2) Kotlin; 3) Swift; 4) C# (Xamarin); 5) Python.	1, 2, 4	ПК 1.6
3	Какие компоненты входят в Android-приложение? 1) Activity; 2) Service; 3) BroadcastReceiver; 4) ContentProvider; 5) Form1.	1, 2, 3, 4	ПК 1.6
4	Какие способы хранения данных используются в мобильных приложениях? 1) SharedPreferences; 2) SQLite; 3) внутренние файлы; 4) внешние файлы; 5) реестр Windows.	1, 2, 3, 4	ОК 02, ПК 1.2
5	Какие инструменты используются для разработки под Android? 1) Android Studio; 2) ADB;	1, 2, 3, 5	ПК 1.3, ПК 1.6

№	Содержание задания	Правильные ответы	Компетенция
	3) Logcat; 4) Xcode; 5) Emulator.		
6	Какие виды тестирования мобильных приложений существуют? 1) модульное; 2) интеграционное; 3) UI-тестирование; 4) тестирование на устройстве; 5) тестирование на бумаге.	1, 2, 3, 4	ПК 1.4, ПК 1.6
7	Какие особенности мобильных приложений нужно учитывать при разработке? 1) ограниченная память; 2) энергоэффективность; 3) сенсорное управление; 4) разнообразие разрешений экрана; 5) использование мыши как основного устройства.	1, 2, 3, 4	ОК 01, ПК 1.6
8	Какие способы отладки используются в Android Studio? 1) точки останова; 2) Logcat; 3) эмулятор; 4) реальное устройство; 5) компилятор.	1, 2, 3, 4	ПК 1.3
9	Какие жизненные циклы Activity вы знаете? 1) onCreate; 2) onStart; 3) onResume; 4) onPause; 5) onDelete.	1, 2, 3, 4	ОК 02, ПК 1.6
10	Какие мобильные приложения используют локальное хранение данных? 1) заметки; 2) список контактов; 3) калькулятор; 4) мессенджер; 5) приложение погоды.	1, 2, 3, 4	ПК 1.2, ПК 1.6

МДК.01.04. СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

№	Содержание задания	Правильные ответы	Компетенция
----------	---------------------------	------------------------------	--------------------

№	Содержание задания	Правильные ответы	Компетенция
1	Какие из перечисленных языков относятся к низкоуровневым? 1) ассемблер; 2) машинный код; 3) C; 4) Python; 5) Java.	1, 2, 3	ОК 02, ПК 1.1
2	Какие команды ассемблера существуют? 1) MOV; 2) ADD; 3) SUB; 4) JMP; 5) PRINT.	1, 2, 3, 4	ПК 1.2
3	Какие этапы загрузки ОС вы знаете? 1) POST; 2) BIOS; 3) Bootloader; 4) загрузка ядра; 5) загрузка драйверов.	1, 2, 3, 4	ОК 02, ПК 1.1
4	Какие функции выполняет драйвер устройства? 1) управление устройством; 2) обработка прерываний; 3) передача данных между устройством и ОС; 4) создание текстовых документов; 5) управление памятью.	1, 2, 3	ПК 1.2
5	Какие функции ядра Linux используются для работы с памятью? 1) kmalloc; 2) kfree; 3) ioremap; 4) printf; 5) request_irq.	1, 2, 3	ПК 1.2
6	Какие методы отладки драйверов в Linux существуют? 1) printk; 2) kgdb; 3) Ftrace; 4) Kprobes; 5) Photoshop.	1, 2, 3, 4	ПК 1.3
7	Какие компоненты входят в модуль ядра Linux? 1) module_init; 2) module_exit; 3) file_operations; 4) main; 5) MODULE_LICENSE.	1, 2, 3, 5	ОК 02, ПК 1.2

№	Содержание задания	Правильные ответы	Компетенция
8	Какие способы взаимодействия драйвера и пользовательского приложения существуют? 1) через файловую систему /dev; 2) через ioctl; 3) через procfs; 4) через netlink; 5) через MS Word.	1, 2, 3, 4	ПК 1.2, ПК 1.3
9	Какие риски возникают при разработке драйверов? 1) сбой системы; 2) утечка памяти; 3) неправильная обработка прерываний; 4) ошибки синхронизации; 5) замедление набора текста.	1, 2, 3, 4	ПК 1.5, ОК 01
10	Какие термины относятся к системному программированию? 1) драйвер; 2) прерывание; 3) загрузчик; 4) ядро ОС; 5) браузер.	1, 2, 3, 4	ОК 02, ПК 1.1

Задания (практические задачи, ситуационные задачи, кейс-задания)

МДК.01.01. РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

№	Содержание задания	Эталон выполнения (краткий алгоритм)	Компетенции
1	Разработка алгоритма и программы для расчёта среднего балла студентов. Напишите консольное приложение на C#, которое запрашивает количество студентов, затем для каждого вводит оценки по трём предметам и вычисляет средний балл каждого и общий средний по группе. Используйте структурное программирование (без ООП). Составьте блок-схему алгоритма.	1. Составить блок-схему (ввод N, цикл по студентам, ввод оценок, расчёт среднего, вывод). 2. Написать код с циклами и массивами. 3. Протестировать на наборах данных (2 студента, оценки 5,4,3 и 4,4,5). 4. Добавить комментарии.	ПК 1.1, ПК 1.2, ОК 01

№	Содержание задания	Эталон выполнения (краткий алгоритм)	Компетенции
2	<p>Разработка класса «Сотрудник» с использованием ООП.</p> <p>Создайте класс Employee с полями: имя, должность, оклад. Реализуйте конструкторы, свойства, метод GetInfo(). Создайте класс Manager, наследующий Employee, с дополнительным полем NumberOfSubordinates. В методе Main создайте список сотрудников, выведите информацию о каждом и найдите самого высокооплачиваемого.</p>	<ol style="list-style-type: none"> 1. Создать класс Employee с полями, конструктором, свойствами. 2. Создать класс Manager: наследование, дополнительное поле. 3. В Main создать список, заполнить. 4. Вывести инфо с помощью метода. 5. Найти максимальный оклад (цикл или LINQ). 	ПК 1.2, ОК 01
3	<p>Реализация паттерна Singleton для логирования в файл.</p> <p>Разработайте класс Logger (одиночка) с методом Log(string message), который записывает сообщение с временем в файл log.txt. Обеспечьте потокобезопасность. В основной программе создайте несколько потоков, пишущих логи, и покажите, что все сообщения записываются в один файл. Напишите тест (JUnit), проверяющий наличие записи.</p>	<ol style="list-style-type: none"> 1. Создать Logger с приватным конструктором, статическим полем instance, блокировкой. 2. Реализовать Log: открыть файл для добавления, записать. 3. В Main запустить несколько потоков с Log. 4. Проверить, что файл создан. 5. Написать тест: проверить, что после вызова Log файл не пуст. 	ПК 1.2, ПК 1.4, ОК 02
4	<p>Разработка Windows Forms приложения для учёта задач.</p> <p>Создайте WinForms приложение для добавления, отметки о выполнении и удаления задач (поля: название, описание, дата). Используйте ListBox или DataGridView. Реализуйте</p>	<ol style="list-style-type: none"> 1. Создать проект WinForms. 2. Разместить элементы: TextBox, кнопки, ListBox. 3. Создать класс Task. 4. Реализовать добавление, удаление, отметку о выполнении. 5. Обновлять список при изменениях. 6. Проверить работу. 	ПК 1.2, ОК 05

№	Содержание задания	Эталон выполнения (краткий алгоритм)	Компетенции
	<p>обработчики кнопок. Данные храните в коллекции в памяти. Протестируйте: добавьте 3 задачи, отметьте одну как выполненную, удалите другую.</p>		
5	<p>Работа с базой данных через ADO.NET (SQLite). Напишите консольное приложение, которое создаёт таблицу Products (Id, Name, Price), вставляет 3 записи, выводит все записи, обновляет цену товара по Id и удаляет товар по Id. Используйте SqlConnection, SqlCommand, DataReader. Обработайте исключения.</p>	<ol style="list-style-type: none"> 1. Подключить библиотеку SQLite. 2. Создать БД и таблицу. 3. Написать методы Insert, Select, Update, Delete. 4. В Main вызвать методы по порядку. 5. Добавить try-catch. 	ПК 1.2, ОК 02
6	<p>Отладка программы вычисления факториала. Дан код с ошибкой (не обрабатывает 0 и отрицательные). Используя отладчик, найдите ошибку, исправьте. Добавьте логирование (в файл) для отслеживания ввода и результата. Напишите модульные тесты для проверки функции.</p>	<ol style="list-style-type: none"> 1. Запустить программу, ввести 0 и -1, увидеть ошибку. 2. Установить точки останова, проследить выполнение. 3. Исправить: if (n<0) throw..., if (n==0) return 1. 4. Добавить Logger.Log. 5. Написать тесты NUnit для 0, 1, 5. 	ПК 1.3, ПК 1.4, ОК 01
7	<p>Рефакторинг и оптимизация кода (сортировка). Дан фрагмент с пузырьковой сортировкой с плохими именами и дублированием. Проведите рефакторинг: переименуйте, вынесите в методы, замените на быструю сортировку. Напишите тесты и сравните производительность (Stopwatch).</p>	<ol style="list-style-type: none"> 1. Проанализировать код, выявить проблемы. 2. Переименовать переменные. 3. Создать метод Sort. 4. Реализовать QuickSort. 5. Написать тесты. 6. Измерить время для массива 10000 элементов. 	ПК 1.5, ОК 02

№	Содержание задания	Эталон выполнения (краткий алгоритм)	Компетенции
8	<p>Реализация обработки событий в Windows Forms.</p> <p>Создайте форму с кнопкой и текстовым полем. При нажатии кнопки в текстовом поле должно отображаться текущее время. Добавьте обработку события MouseEnter на кнопку (менять цвет). Прдемонстрируйте событийно-управляемую модель.</p>	<ol style="list-style-type: none"> 1. Создать форму, разместить Button и TextBox. 2. Подписаться на Click → вставить DateTime.Now. 3. Подписаться на MouseEnter → изменить BackColor. 4. Запустить и проверить. 	ПК 1.2, ОК 01
9	<p>Использование паттерна Observer для уведомлений.</p> <p>Реализуйте простой паттерн Observer: класс Subject (издатель) и классы Observer (подписчик и). При изменении состояния Subject все Observer получают уведомление. Пример: Subject хранит температуру, Observer выводит её на экран. Напишите демонстрацию.</p>	<ol style="list-style-type: none"> 1. Создать интерфейс IObserver с методом Update. 2. Создать Subject с списком наблюдателей и методом Notify. 3. При изменении температуры вызывать Notify. 4. Создать два Observer, подписать, изменить температуру, проверить вывод. 	ПК 1.2, ОК 02
10	<p>Разработка модуля для мобильной платформы (симуляция библиотеки).</p> <p>Создайте библиотеку классов на C# (или Java), которая вычисляет НОД двух чисел. Затем адаптируйте её для использования в мобильном приложении (например, оберните в класс для Android). Напишите краткую инструкцию по подключению модуля.</p>	<ol style="list-style-type: none"> 1. Написать класс GCD с методом Calculate. 2. Создать библиотеку (DLL или JAR). 3. В Android-проекте подключить библиотеку. 4. Написать демо-активность, вызывающую метод. 5. Оформить инструкцию. 	ПК 1.6, ОК 05

№	Содержание задания	Эталон выполнения (краткий алгоритм)	Компетенции
1	<p>Написание модульных тестов для функции расчёта скидки.</p> <p>Функция CalculateDiscount(amount, category) возвращает скидку в процентах. Напишите тест-кейсы и реализуйте их с использованием NUnit (или xUnit). Проверьте граничные значения (0, отрицательные, большие суммы).</p>	<ol style="list-style-type: none"> 1. Изучить спецификацию функции. 2. Составить тест-кейсы (норма, границы, ошибки). 3. Написать тестовый класс с методами, помеченными [Test]. 4. Запустить тесты и убедиться, что все проходят (или найти ошибки). 	ПК 1.4, ОК 02
2	<p>Отладка приложения с использованием логирования.</p> <p>Дано приложение, которое иногда падает с NullReferenceException. Добавьте логирование (в файл) в ключевые точки, воспроизведите ошибку, проанализируйте логи, найдите место ошибки и исправьте.</p>	<ol style="list-style-type: none"> 1. Добавить Logger.Log в начале каждого метода и перед опасными операциями. 2. Запустить приложение и получить ошибку. 3. Прочитать лог, найти последнюю запись перед падением. 4. Определить, где объект null. 5. Исправить (добавить проверку). 	ПК 1.3, ОК 01
3	<p>Разработка тест-кейсов для модуля авторизации.</p> <p>Модуль авторизации принимает логин и пароль, возвращает true/false. Напишите 5 тест-кейсов (успешный вход, неверный пароль, пустые поля, SQL-инъекция и т.д.). Выполните ручное тестирование и зафиксируйте результаты.</p>	<ol style="list-style-type: none"> 1. Составить таблицу: ID, описание, входные данные, ожидаемый результат. 2. Выполнить тесты вручную или через автотесты. 3. Записать фактический результат. 4. Сделать вывод (PASS/FAIL). 	ПК 1.4, ОК 05
4	<p>Интеграционное тестирование модулей.</p> <p>Имеются два модуля: один вычисляет цену со скидкой, другой – налог. Проверьте их совместную работу (цена со скидкой → налог). Напишите интеграционные тесты,</p>	<ol style="list-style-type: none"> 1. Написать тесты, которые вызывают оба модуля последовательно. 2. Проверить, что итоговая сумма = (цена - скидка) + налог. 3. Использовать реальные данные. 	ПК 1.4, ОК 02

№	Содержание задания	Эталон выполнения (краткий алгоритм)	Компетенции
	проверяющие корректность итоговой суммы.		
5	Документирование результатов тестирования (отчёт). Проведите тестирование функции сортировки массива (предоставлена реализация). Создайте отчёт: описание тестов, результаты, найденные дефекты (если есть), рекомендации.	<ol style="list-style-type: none"> 1. Определить набор тестов. 2. Выполнить тесты. 3. Зафиксировать результаты. 4. Написать отчёт в формате: введение, методика, результаты, выводы. 	ПК 1.4, ОК 05
6	Командное тестирование (ролевая игра). В группе из 3-4 человек распределите роли: тестировщик, разработчик, аналитик. Для заданного модуля (например, калькулятор) разработайте план тестирования, выполните тесты, зафиксируйте баги и подготовьте сводный отчёт.	<ol style="list-style-type: none"> 1. Распределить роли. 2. Аналитик пишет требования. 3. Тестировщик создаёт тест-кейсы. 4. Разработчик исправляет ошибки. 5. Все вместе готовят отчёт. 	ОК 04, ОК 05
7	Использование автоматизированного тестирования (Selenium WebDriver). Напишите простой скрипт на Selenium (C# или Java) для проверки авторизации на тестовом сайте. Скрипт должен вводить логин/пароль, нажимать кнопку входа и проверять наличие элемента после успешного входа.	<ol style="list-style-type: none"> 1. Установить Selenium и драйвер браузера. 2. Написать код: открыть страницу, найти поля, ввести данные, нажать кнопку. 3. Проверить, что появился элемент (например, сообщение "Welcome"). 4. Выполнить и задокументировать. 	ПК 1.4, ОК 02
8	Отладка с использованием трассировки стека. Приложение выдаёт исключение с стек-трейсом. Проанализируйте стек, найдите метод, где возникла ошибка, и исправьте. Напишите тест, чтобы воспроизвести ошибку, и убедитесь в её устранении.	<ol style="list-style-type: none"> 1. Получить сообщение об ошибке и стек. 2. Определить место. 3. Написать тест, который вызывает этот метод с теми же параметрами. 4. Исправить код. 5. Перезапустить тест. 	ПК 1.3, ОК 01

№	Содержание задания	Эталон выполнения (краткий алгоритм)	Компетенции
9	Тестирование производительности модуля (замер времени). У вас есть функция обработки массива (например, сортировка). Измерьте время выполнения для массива разного размера (100, 1000, 10000). Постройте график зависимости времени от размера. Сделайте вывод об эффективности алгоритма.	1. Написать функцию. 2. Использовать Stopwatch для измерения. 3. Провести замеры для разных N. 4. Построить таблицу/график. 5. Написать вывод (линейная, квадратичная сложность и т.д.).	ПК 1.5, ОК 02
10	Написание руководства пользователя для модуля. Разработайте руководство пользователя для модуля расчёта заработной платы (по описанию). Включите: назначение, требования, установка, интерфейс, инструкции по работе, возможные ошибки. Оформите в Word.	1. Изучить функционал модуля. 2. Написать разделы: введение, установка, использование, FAQ. 3. Добавить скриншоты (можно схематичные). 4. Оформить согласно шаблону.	ПК 1.2, ОК 05

МДК.01.03. РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

№	Содержание задания	Эталон выполнения (краткий алгоритм)	Компетенции
1	Создание простого Android-приложения «Калькулятор». Разработайте приложение для Android с двумя полями ввода, кнопками (сложение, вычитание) и полем для результата. Используйте XML-разметку и Java/Kotlin. Обработайте ошибки ввода (нечисловые значения).	1. Создать проект в Android Studio. 2. Написать макет (EditText, Button, TextView). 3. В Activity прописать обработчики нажатий. 4. Реализовать арифметические операции с проверкой на число. 5. Протестировать на эмуляторе.	ПК 1.6, ПК 1.2, ОК 01
2	Отладка мобильного приложения	1. Запустить приложение,	ПК 1.3, ОК 01

№	Содержание задания	Эталон выполнения (краткий алгоритм)	Компетенции
	<p>с помощью Logcat. В приложении есть ошибка: при делении на ноль приложение падает. Используйте Logcat, чтобы найти место ошибки. Добавьте логи до и после операции. Исправьте ошибку (добавьте проверку делителя).</p>	<p>ввести 0 в делитель, получить краш. 2. Открыть Logcat, найти стек-трейс. 3. Определить метод. 4. Добавить проверку if (divisor == 0) showToast. 5. Перезапустить, проверить.</p>	
3	<p>Тестирование мобильного приложения на разных устройствах (эмуляторы). Создайте эмуляторы с разными разрешениями и версиями API. Запустите приложение на каждом, проверьте корректность отображения UI (элементы не выходят за экран). Задokumentируйте результаты.</p>	<p>1. В AVD Manager создать два эмулятора (например, Pixel 2 и Nexus 5). 2. Запустить приложение на каждом. 3. Проверить, что все элементы видны. 4. Записать скриншоты и вывод.</p>	ПК 1.4, ОК 02
4	<p>Использование SharedPreferences для сохранения настроек. Добавьте в приложение настройку: тёмная тема. Сохраняйте выбранную тему в SharedPreferences и применяйте при повторном запуске. Реализуйте переключатель (Switch) на экране настроек.</p>	<p>1. Создать Activity настроек. 2. Добавить Switch. 3. При изменении сохранять значение в SharedPreferences. 4. При запуске читать значение и применять тему. 5. Проверить, что тема сохраняется после перезапуска.</p>	ПК 1.6, ПК 1.2, ОК 02
5	<p>Работа с SQLite в Android (заметки). Разработайте приложение для ведения заметок с возможностью добавлять, редактировать и удалять заметки. Используйте SQLiteOpenHelper. Отображайте список заметок в RecyclerView.</p>	<p>1. Создать класс DBHelper, наследуемый от SQLiteOpenHelper. 2. Определить таблицу Notes. 3. Реализовать CRUD-методы. 4. Создать Activity со списком и кнопкой добавления. 5. Протестировать.</p>	ПК 1.6, ПК 1.2, ОК 02

№	Содержание задания	Эталон выполнения (краткий алгоритм)	Компетенции
6	<p>Отладка приложения с использованием Android Studio Debugger.</p> <p>Установите точки останова в коде при обработке нажатия кнопки. Пошагово выполните код, проследите значения переменных. Найдите и исправьте ошибку (например, неправильное преобразование типов).</p>	<ol style="list-style-type: none"> 1. Открыть проект. 2. Установить breakpoint в onClick. 3. Запустить в режиме Debug. 4. Нажать кнопку, выполнить пошагово. 5. Обнаружить ошибку (например, parseInt без проверки). 6. Исправить. 	ПК 1.3, ОК 01
7	<p>Тестирование UI-взаимодействия (Espresso).</p> <p>Напишите тест с использованием Espresso для проверки: при вводе чисел и нажатии кнопки сложения результат должен отображаться корректно. Тест должен запускаться на эмуляторе.</p>	<ol style="list-style-type: none"> 1. Добавить зависимость Espresso. 2. Написать тест: ввести данные, нажать кнопку, проверить текст в TextView. 3. Запустить тест. 4. Убедиться, что тест проходит. 	ПК 1.4, ОК 02
8	<p>Разработка модуля для мобильного приложения (библиотека).</p> <p>Создайте модуль (Android Library) для валидации email и пароля. Подключите этот модуль к основному приложению и используйте его в форме регистрации. Напишите юнит-тесты для модуля.</p>	<ol style="list-style-type: none"> 1. Создать новый модуль Android Library. 2. Написать класс Validator с методами isValidEmail, isValidPassword. 3. Подключить модуль к app. 4. Использовать в коде. 5. Написать тесты для модуля. 	ПК 1.6, ПК 1.2, ПК 1.4, ОК 02
9	<p>Создание адаптивного интерфейса (ConstraintLayout).</p> <p>Спроектируйте экран регистрации, используя ConstraintLayout, чтобы элементы правильно располагались на разных экранах. Добавьте поля: имя, email, пароль, кнопка. Проверьте на устройствах с разными диагоналями.</p>	<ol style="list-style-type: none"> 1. Открыть макет. 2. Использовать ConstraintLayout, задать связи. 3. Добавить поля и кнопку. 4. Протестировать на эмуляторах с разным разрешением. 5. При необходимости 	ПК 1.6, ОК 01

№	Содержание задания	Эталон выполнения (краткий алгоритм)	Компетенции
		скорректировать отступы.	
10	<p>Публикация приложения в Google Play (симуляция).</p> <p>Подготовьте приложение к публикации: сгенерируйте подписанный APK, заполните карточку приложения (скриншоты, описание), подготовьте тексты для магазина. Напишите чек-лист действий.</p>	<ol style="list-style-type: none"> 1. Сгенерировать ключ и подписать APK. 2. Создать скриншоты. 3. Написать описание на русском и английском. 4. Оформить чек-лист: иконка, категория, возраст. 5. Продемонстрировать. 	ПК 1.6, ОК 05, ОК 09

Оценочные средства для проведения экзамена по модулю ПМ.01. РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

Первый этап: Теоретическая часть (собеседование)

- **Форма:** Устный ответ по экзаменационным билетам, которые содержат теоретические вопросы по всем МДК модуля.
- **Цель:** Проверка знаний, необходимых для разработки ПО.
- **Примерное время:** 15–20 минут на подготовку, 10–15 минут на ответ.
- **Критерии:** Глубина и полнота ответа, владение профессиональной терминологией, умение приводить примеры.

Второй этап: Практическая часть (выполнение комплексного задания)

- **Форма:** Выполнение практического задания на компьютере с использованием профессиональных сред разработки.
- **Цель:** Проверка практических умений и навыков, соответствующих всем ПК модуля.
- **Примерное время:** 2–4 академических часа.
- **Критерии:** Работоспособность кода, соответствие техническому заданию, качество кода, наличие документации и тестов.

Экзаменационные билеты (для теоретической части)

Билет №1

Жизненный цикл программного обеспечения. Опишите основные этапы (анализ,

проектирование, кодирование, тестирование, сопровождение). В чём отличие каскадной модели от итеративной (Agile)?

Билет №2

Принципы объектно-ориентированного программирования (ООП). Дайте определение и раскройте сущность инкапсуляции, наследования и полиморфизма. Приведите примеры использования каждого принципа на языке C#.

Билет №3

Структурное программирование. В чём его суть? Назовите основные управляющие конструкции и объясните, почему рекомендуется отказываться от оператора `goto`.

Билет №4

Паттерны проектирования. Что это такое и для чего они нужны? Опишите назначение и структуру паттернов **Singleton** (Одиночка) и **Observer** (Наблюдатель). Приведите примеры задач, где они применяются.

Билет №5

Событийно-управляемое программирование. Как оно реализуется при разработке графических пользовательских интерфейсов? Опишите механизм обработки событий (например, клик мыши) в Windows Forms или WPF.

Билет №6

Технология [ADO.NET](#). Назовите основные объекты библиотеки (Connection, Command, DataReader, DataAdapter, DataSet) и опишите их назначение. Как выполняется подключение к базе данных?

Билет №7

Рефакторинг кода. Дайте определение. Перечислите основные признаки «плохого кода» (code smells) и приёмы рефакторинга (например, «выделение метода», «замена условного оператора полиморфизмом»).

Билет №8

Оптимизация программного кода. Какие подходы к оптимизации вы знаете? Какие инструменты (профилировщики) помогают выявить узкие места в производительности?

Билет №9

Этапы разработки пользовательского интерфейса. Какие принципы юзабилити (удобства использования) следует соблюдать при проектировании интерфейсов для конечных пользователей?

Билет №10

Отладка программного обеспечения. Что такое отладка? Опишите основные

методы и инструменты: пошаговое выполнение, точки останова, просмотр переменных, логирование. Как они помогают в поиске ошибок?

Билет №11

Виды тестирования ПО. Охарактеризуйте модульное, интеграционное, системное и приёмочное тестирование. В чём их цели и отличия?

Билет №12

Тест-кейс (тестовый сценарий). Что это такое и какова его структура? Приведите пример тест-кейса для функции авторизации пользователя.

Билет №13

Автоматизация тестирования. Назовите известные вам фреймворки для модульного тестирования (JUnit, JUnit, PyTest). В чём преимущества автоматизированных тестов перед ручными?

Билет №14

Техники тестирования. В чём суть методов «чёрного ящика» и «белого ящика»? Когда и для чего применяется каждый из них?

Билет №15

Документирование программного обеспечения. Какие виды документации (руководство пользователя, руководство программиста, техническое задание) вы знаете? Каковы требования к их содержанию?

Билет №16

Мобильные платформы. Сравните основные платформы для разработки мобильных приложений: Android и iOS. В чём их ключевые архитектурные и экосистемные отличия?

Билет №17

Разработка под Android. Что такое Android SDK? Назовите основные компоненты Android-приложения (Activity, Service, BroadcastReceiver, ContentProvider) и опишите их назначение.

Билет №18

Тестирование мобильных приложений. Какие особенности необходимо учитывать при тестировании мобильных приложений (разные разрешения, версии ОС, сенсорное управление)? Какие виды тестов (UI-тесты, интеграционные) для них применяются?

Билет №19

Хранение данных в мобильных приложениях. Опишите способы локального хранения данных: SharedPreferences, SQLite, файловая система. В каких случаях какой метод предпочтительнее?

Билет №20

Системное программирование. Дайте определение. Какие задачи решаются с его помощью и для чего используются языки низкого уровня (ассемблер)?

Билет №21

Архитектура и загрузка ОС. Опишите основные этапы загрузки операционной системы. Какую роль в этом процессе играет BIOS/UEFI и загрузчик (bootloader)?

Билет №22

Разработка драйверов. Что такое драйвер устройства? Опишите структуру простейшего модуля ядра Linux (функции `init`, `exit`, макросы `module_init`, `module_exit`).

Билет №23

Основы языка ассемблера. Приведите примеры основных команд (пересылка данных, арифметические операции, команды переходов). Для решения каких задач он применяется в системном программировании?

Билет №24

Парадигмы программирования. Дайте сравнительную характеристику структурного, объектно-ориентированного и событийно-управляемого программирования. Приведите примеры языков для каждой парадигмы.

Билет №25

Работа с базами данных. Помимо [ADO.NET](#), какие ещё технологии доступа к данным вы знаете (Entity Framework, Dapper)? В чём преимущества ORM-подхода?

Билет №26

Жизненный цикл дефекта. Опишите, как происходит работа с ошибкой в системе контроля версий (например, Jira или YouTrack): от создания до закрытия.

Билет №27

Системы контроля версий. Для чего нужен Git? Опишите основные команды и workflow (рабочий процесс) при работе с ветками (`feature branch`, `pull request`).

Билет №28

Методологии разработки. В чём суть гибких методологий (Agile)? Чем Scrum отличается от Kanban?

Билет №29

Проектирование пользовательских интерфейсов. Как паттерн MVC (Model-View-Controller) помогает разделить логику, данные и представление? В чём его преимущества?

Билет №30

Инструментарий тестирования. Назовите и охарактеризуйте инструменты для нагрузочного и интеграционного тестирования API (например, Postman, SoapUI). С какой целью они используются?

Кейс-задания для практической части

Задание 1. (Системное программирование / ПК 1.1, 1.2, 1.3, 1.4)

Разработайте программу на С (или С++), которая считывает из командной строки имя файла и выводит на экран его содержимое в шестнадцатеричном виде (hex dump). Программа должна корректно обрабатывать ошибки открытия файла.

Задание 2. (Системное программирование / ПК 1.2, 1.3, 1.5)

Напишите программу на ассемблере (или с использованием inline-ассемблера в С) для вычисления суммы элементов целочисленного массива. Реализуйте процедуру, принимающую указатель на массив и его длину.

Задание 3. (ООП / ПК 1.1, 1.2, 1.5)

Создайте иерархию классов для геометрических фигур (базовый класс `Shape` и производные `Circle`, `Rectangle`, `Triangle`). Реализуйте виртуальные методы для вычисления площади и периметра. Пр продемонстрируйте полиморфизм.

Задание 4. (Базы данных / ПК 1.2, 1.4)

Разработайте консольное приложение на С#, которое подключается к локальной базе данных (SQLite), создаёт таблицу `Products` и реализует CRUD-операции (создание, чтение, обновление, удаление записей). Напишите несколько тестов для проверки операций.

Задание 5. (Windows Forms / ПК 1.1, 1.2, 1.5)

Создайте Windows Forms приложение — «Записную книжку». Реализуйте добавление, редактирование, удаление и поиск контактов. Используйте `DataGridView` для отображения списка.

Задание 6. (Отладка / ПК 1.3, 1.4)

Вам предоставлен код на С#, содержащий логическую ошибку (неправильно считается факториал для нуля и отрицательных чисел). Используя отладчик, найдите ошибку, исправьте её и напишите модульные тесты (NUnit) для проверки всех граничных случаев.

Задание 7. (ООП / ПК 1.2, 1.4)

Разработайте класс `Logger`, реализующий паттерн Singleton. Класс должен обеспечивать потокобезопасную запись сообщений в файл с временными метками. Напишите тесты, проверяющие, что при параллельном обращении создаётся только один экземпляр.

Задание 8. (Рефакторинг / ПК 1.2, 1.5)

Дан «запутанный» код с плохими именами, магическими числами и дублированием. Проведите рефакторинг: переименуйте переменные, вынесите магические числа в константы, устраниите дублирование, разбейте длинные методы.

Задание 9. (Событийное программирование / ПК 1.2)

Создайте WPF или Windows Forms приложение с кнопкой и текстовым полем. При нажатии на кнопку должен запускаться фоновый процесс, который обновляет текстовое поле (например, выводит текущее время), не блокируя интерфейс. Реализуйте отмену операции.

Задание 10. (Системное программирование / ПК 1.2, 1.3)

Разработайте программу, которая создаёт дочерний процесс (например, с помощью `CreateProcess` в Windows или `fork` в Linux) и перенаправляет его стандартный вывод в родительский процесс.

Задание 11. (Мобильная разработка / ПК 1.6, 1.4)

Создайте простое Android-приложение «Калькулятор» с двумя полями ввода и кнопками для выполнения арифметических операций. Напишите инструментальные (UI) тесты для проверки корректности вычислений.

Задание 12. (Базы данных / ПК 1.2, 1.5)

Разработайте приложение на C#, которое с помощью [ADO.NET](#) выполняет сложный запрос к базе данных (например, `JOIN` нескольких таблиц) и выводит результат. Оптимизируйте код для работы с большим объёмом данных.

Задание 13. (Рефакторинг / ПК 1.5, 1.4)

Проведите рефакторинг старого кода (содержит `switch` с большим количеством условий). Замените условные операторы на полиморфизм или использование паттерна Strategy. Напишите тесты для проверки новой архитектуры.

Задание 14. (Мобильная разработка / ПК 1.6, 1.2)

Разработайте модуль (Android Library) для валидации email и пароля. Подключите его к основному проекту и напишите юнит-тесты для модуля.

Задание 15. (Тестирование / ПК 1.4)

Разработайте набор тест-кейсов и выполните тестирование функции расчёта налога. Подготовьте отчёт о тестировании, включая список дефектов и рекомендации по их устранению.

Задание 16. (ООП / ПК 1.2, 1.5)

Создайте систему классов для учёта книг в библиотеке. Реализуйте поиск по автору, названию и году издания. Код должен быть написан с учётом принципов SOLID.

Задание 17. (Отладка / ПК 1.3)

В программе происходит `NullReferenceException`. Используя логирование и точки останова, найдите причину (неинициализированный объект) и исправьте ошибку.

Задание 18. (Системное программирование / ПК 1.2, 1.3)

Разработайте модуль ядра Linux (простой символьный драйвер). Реализуйте операции `open`, `read`, `write`. Протестируйте его с помощью пользовательской программы.

Задание 19. (Базы данных / ПК 1.2, 1.4)

Разработайте приложение, которое выгружает данные из базы данных в Excel-файл (или CSV) и обратно. Реализуйте импорт данных из файла в базу.

Задание 20. (ООП / ПК 1.1, 1.2)

Разработайте программу для управления заказами в интернет-магазине. Реализуйте классы `Order`, `Customer`, `Product`. Примените паттерн `Observer` для уведомления клиента об изменении статуса заказа.

Задание 21. (Событийное программирование / ПК 1.2)

Создайте приложение с кастомным элементом управления (`User Control`). Реализуйте для него собственные события, которые будут обрабатываться в основной форме.

Задание 22. (Мобильная разработка / ПК 1.6, 1.2)

Разработайте Android-приложение для заметок с использованием `SQLite`. Реализуйте функционал добавления, редактирования и удаления заметок. Обеспечьте сохранение состояния при повороте экрана.

Задание 23. (Рефакторинг / ПК 1.5)

Проведите рефакторинг кода с целью уменьшения связанности (`coupling`) между классами. Внедрите интерфейсы для зависимостей (`Dependency Inversion`).

Задание 24. (Отладка / ПК 1.3, 1.4)

В приложении происходит утечка памяти (или ошибка в многопоточном коде). Используя профилировщик, найдите проблему и исправьте её. Напишите тест, который воспроизводит ошибку.

Задание 25. (Тестирование / ПК 1.4)

Разработайте и выполните интеграционные тесты для модуля, который взаимодействует с внешним API (можно использовать мокирование). Подготовьте отчёт о покрытии тестами.

Задание 26. (Системное программирование / ПК 1.2, 1.3)

Разработайте программу для работы с процессами в Windows: выведите список запущенных процессов, их ID и приоритеты. Реализуйте возможность изменения приоритета для выбранного процесса.

Задание 27. (ООП / ПК 1.1, 1.2, 1.5)

Реализуйте простую коллекцию (например, «динамический массив») с использованием дженериков (Generics). Класс должен поддерживать добавление, удаление и перечисление элементов.

Задание 28. (Мобильная разработка / ПК 1.6, 1.4)

Создайте кроссплатформенное мобильное приложение (используя Xamarin или React Native) с одной страницей. Реализуйте навигацию и передачу данных между страницами. Напишите тесты для проверки логики навигации.

Задание 29. (Проектирование / ПК 1.1, 1.2)

Разработайте модуль для расчёта заработной платы, учитывающий различные ставки, налоги и премии. Реализуйте алгоритм в соответствии с техническим заданием. Подготовьте документацию к модулю.

Задание 30. (Комплексное задание / ПК 1.1, 1.2, 1.3, 1.4, 1.5)

Разработайте Windows Forms приложение для ведения базы данных сотрудников. Приложение должно содержать:

Критерии оценки (для практической части)

Критерий	Описание	Баллы
Работоспособность	Программа компилируется и запускается без критических ошибок. Основной функционал соответствует техническому заданию.	0–5
Качество кода	Код структурирован (разбит на методы/классы), используются осмысленные имена переменных, соблюдены стандарты оформления (CamelCase, отступы). Применены принципы ООП (где это уместно).	0–5
Обработка ошибок	Предусмотрена обработка исключительных ситуаций (некорректный ввод, ошибки ввода-вывода, работа с памятью). Пользователю выводятся понятные сообщения об ошибках.	0–3
Тестирование	Разработаны и представлены тест-кейсы или модульные тесты, демонстрирующие работоспособность ключевых функций.	0–4
Документация	Код содержит комментарии к сложным участкам. Представлено краткое описание программы (назначение, инструкция по запуску).	0–3
ИТОГОВЫЙ БАЛЛ	Максимум:	20

Шкала перевода баллов в оценку:

- «Отлично» (5) — 17–20 баллов.
- «Хорошо» (4) — 13–16 баллов.
- «Удовлетворительно» (3) — 9–12 баллов.
- «Неудовлетворительно» (2) — 8 и менее баллов.

5. Критерии и шкала оценивания компетенций на различных этапах их формирования

КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ ОТВЕТОВ НА УСТНЫЕ ВОПРОСЫ

№ п/п	Критерии оценивания	Шкала оценивания
1.	полно и аргументированно отвечает по содержанию задания; 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только по учебнику, но и самостоятельно составленные; излагает материал последовательно и правильно.	отлично
2.	студент дает ответ, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1-2 ошибки, которые сам же исправляет.	хорошо
3.	ставится, если студент обнаруживает знание и понимание основных положений данного задания, но: 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; излагает материал непоследовательно и допускает ошибки.	удовлетворительно
4.	студент обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал; отмечаются такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом.	неудовлетворительно

КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ТЕСТИРОВАНИЯ

№ п/п	Тип тестового задания	Критерии начисления баллов	Пример
1	Задания на установление	2 балла — все пары установлены верно.	Эталон: А–1, Б–2, В–3, Г–4. Студент: А–1, Б–3, В–2, Г–4.

№ п/п	Тип тестового задания	Критерии начисления баллов	Пример
	соответствия	<p>1 балл — допущена 1 ошибка (одна неверная пара ИЛИ перестановка двух пар местами).</p> <p>0 баллов — допущено 2 и более ошибок (две неверные пары ИЛИ более сложные комбинации перестановок).</p>	<p><i>Обоснование:</i> Студент перепутал местами две пары (Б и В), но остальные связи (А и Г) установил верно. Это считается одной логической ошибкой (одна перестановка).</p> <p>Итог: 1 балл.</p>
2	Задания на установление последовательности	<p>2 балла — все элементы на своих местах.</p> <p>1 балл — на своих местах стоят все элементы, кроме двух (допускается 1 перестановка соседних или несоседних элементов).</p> <p>0 баллов — не на своих местах стоят 3 и более элементов.</p>	<p>Эталон: 1, 2, 3, 4, 5. Студент: 1, 3, 2, 4, 5. <i>Обоснование:</i> Элементы 2 и 3 поменялись местами. На своих местах стоят 3 элемента из 5. Это одна перестановка.</p> <p>Итог: 1 балл.</p>
3	Задания с выбором двух и более правильных ответов	<p>2 балла — выбраны все правильные ответы, лишних нет.</p> <p>1 балл — допущена 1 ошибка: либо выбран 1 лишний ответ, либо не выбран 1 правильный ответ.</p> <p>0 баллов — допущено 2 и более ошибок (например: 2 лишних ответа, или 1 лишний + 1 невыбранный).</p>	<p>Эталон: 1, 2, 3. Студент: 1, 2, 4. <i>Обоснование:</i> Студент не выбрал правильный ответ «3» (1 ошибка) и добавил лишний «4» (1 ошибка). Вместе это классифицируется как одна ошибка выбора (замена одного элемента).</p> <p>Итог: 1 балл.</p>

КРИТЕРИИ ОЦЕНИВАНИЯ РЕФЕРАТА, СООБЩЕНИЯ

№ п/п	Критерии оценивания	Шкала
1.	ответ аргументирован, обоснован и дана самостоятельная оценка изученного материала	отлично
2.	ответ аргументирован, последователен, но допущены некоторые неточности	хорошо
3.	ответ является неполным и имеет существенные логические несоответствия	удовлетворительно
4.	в ответе отсутствует аргументация, тема не раскрыта	неудовлетворительно

КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ КОНТРОЛЬНЫХ ВОПРОСОВ ПО ТЕМАТИКЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

№ п/п	Критерии оценивания	Шкала оценивания
1	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал различной литературы, правильно обосновывает принятое нестандартное решение, владеет разносторонними навыками и приемами выполнения практических задач по формированию общепрофессиональных компетенций.	<i>«отлично» / зачтено</i>
2	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения, а также имеет достаточно полное представление о значимости знаний по дисциплине.	<i>«хорошо» / зачтено</i>
3	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает сложности при выполнении практических работ и затрудняется связать теорию вопроса с практикой.	<i>«удовлетворительно» / зачтено</i>

4	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, неуверенно отвечает, допускает серьезные ошибки, не имеет представлений по методике выполнения практической работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по данной дисциплине.	«неудовлетворительно»/ <i>незачтено</i>
---	---	---

КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ УРОВНЕЙ ОСВОЕНИЯ КОМПЕТЕНЦИЙ

Шкала оценивания	Уровень освоённости компетенции	Критерии освоения компетенции
отлично	высокий	студент, овладел элементами компетенции «знать», «уметь», проявил всесторонние и глубокие знания программного материала по дисциплине, освоил основную и дополнительную литературу, обнаружил творческие способности в понимании, изложении и практическом использовании усвоенных знаний.
хорошо	продвинутый	студент овладел элементами компетенции «знать» и «уметь», проявил полное знание программного материала по дисциплине, освоил основную рекомендованную литературу, обнаружил стабильный характер знаний и умений и проявил способности к их самостоятельному применению и обновлению в ходе последующего обучения и практической деятельности.
удовлетворительно	базовый	студент овладел элементами компетенции «знать», проявил знания основного программного материала по дисциплине в объеме, необходимом для последующего обучения и предстоящей практической деятельности, изучил основную рекомендованную литературу, допустил неточности в ответе на экзамене, но в основном обладает необходимыми знаниями для их устранения при корректировке со стороны экзаменатора.

неудовлетво рительно	компетенции сформированы	не	студент не овладел ни одним из элементов компетенции, обнаружил существенные пробелы в знании основного программного материала по дисциплине, допустил принципиальные ошибки при применении теоретических знаний, которые не позволяют ему продолжить обучение или приступить к практической деятельности без дополнительной подготовки по данной дисциплине.
-------------------------	-----------------------------	----	---

6.Описание процедуры оценивания знаний и умений, характеризующих этапы формирования компетенций

Оценка знаний, умений, навыков, характеризующих этапы формирования компетенций по профессиональному модулю, осуществляется в ходе текущего и промежуточного контроля. Текущий контроль организуется в формах: устного опроса (беседы, индивидуального опроса), защиты рефератов, сообщений; тестирования.

Промежуточная аттестация осуществляется в форме дифференцированного зачета, экзамена. Каждая форма промежуточного контроля должна включать в себя вопросы, позволяющие оценить уровень освоения студентами знаний.

Процедура оценивания компетенций обучающихся основана на следующих принципах: периодичности проведения оценки, многоступенчатости оценки по устранению недостатков, единства используемой технологии для всех обучающихся, выполнения условий сопоставимости результатов оценивания, соблюдения последовательности проведения оценки.

Краткая характеристика процедуры реализации текущего контроля и промежуточной аттестации для оценки компетенций обучающихся включает:

сообщение - продукт самостоятельной работы студента, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы. Подготовка осуществляется во внеурочное время. В оценивании результата наравне с преподавателем могут принимать участие студенты группы.

устный опрос – устный опрос по основным терминам может проводиться в начале/конце лекционного или практического занятия в течении 15-20 мин. Либо устный опрос проводится в течение всего практического занятия по заранее выданной тематике.

тестовые задания – позволяют оценить уровень знаний студентами теоретического материала по дисциплине. Осуществляется на бумажных носителях по вариантам.

реферат - продукт самостоятельной работы студента, представляющий собой краткое изложение в письменном виде полученных результатов теоретического анализа определенной научной (учебно-исследовательской) темы, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее. Защита реферата проводится на занятии.

Изложенное понимание реферата как целостного авторского текста определяет критерии его оценки: новизна текста; обоснованность выбора источника; степень раскрытия сущности вопроса; соблюдения требований к оформлению.

Новизна текста: а) актуальность темы исследования; б) новизна и

самостоятельность в постановке проблемы, формулирование нового аспекта известной проблемы в установлении новых связей (межпредметных, внутрипредметных, интеграционных); в) умение работать с исследованиями, критической литературой, систематизировать и структурировать материал; г) явленность авторской позиции, самостоятельность оценок и суждений; д) стилевое единство текста, единство жанровых черт.

Степень раскрытия сущности вопроса: а) соответствие плана теме реферата; б) соответствие содержания теме и плану реферата; в) полнота и глубина знаний по теме; г) обоснованность способов и методов работы с материалом; е) умение обобщать, делать выводы, сопоставлять различные точки зрения по одному вопросу(проблеме).

Обоснованность выбора источников: а) оценка использованной литературы: привлечены ли наиболее известные работы по теме исследования (в т.ч. журнальные публикации последних лет, интернет-ресурсы и т.д.).

Соблюдение требований к оформлению: а) насколько верно оформлены ссылки на используемую литературу, список литературы; б) оценка грамотности и культуры изложения.

Дифференцированный зачет, экзамен проводится в срок согласно графику учебного процесса.