

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Федорова Марина Владимировна
Должность: Директор филиала
Дата подписания: 06.08.2026 09:11:33
Уникальный программный ключ:
e766def0e2eb455f02135d659e45051ac23041da

Приложение
к ОПОП по специальности
09.02.11 Разработка и управление
программным обеспечением

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ
ПМ.03 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ
СИСТЕМ
для специальности
09.02.11 РАЗРАБОТКА И УПРАВЛЕНИЕ ПРОГРАММНЫМ
ОБЕСПЕЧЕНИЕМ
Базовая подготовка
среднего профессионального образования
(год начала подготовки 2026)

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ	3
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ	4
3. ОЦЕНКА УРОВНЕЙ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	5
4. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ	255
5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ (ВИДА ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ)	667

1. ОБЩИЕ ПОЛОЖЕНИЯ

Фонд оценочных средств (ФОС) разработан с целью установления соответствия образовательных достижений студентов требованиям программы подготовки специалистов среднего звена по профессиональному модулю ПМ.03 Проектирование и разработка информационных систем для компьютерных систем.

ФОС включают контрольные материалы для проведения текущего контроля и промежуточной аттестации.

ФОС текущего контроля используется для оперативного и регулярного управления учебной деятельностью студентов.

ФОС промежуточной аттестации студентов по профессиональному модулю предназначен для оценки степени достижения запланированных результатов обучения по завершению изучения междисциплинарных курсов профессионального модуля, экзамена (квалификационного) по завершению изучения профессионального модуля в целом.

ФОС разработан на основании:

- программы подготовки специалистов среднего звена по специальности СПО 09.02.11 Разработка и управление программным обеспечением;
- рабочей программы профессионального модуля ПМ.03 Проектирование и разработка информационных систем;
- учебного плана по специальности СПО 09.02.11 Разработка и управление программным обеспечением.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

Результатом в рамках освоения профессионального модуля *ПМ.03 Проектирование и разработка информационных систем* является овладение студентами вида профессиональной деятельности *Проектирование и разработка информационных систем*, в том числе профессиональными (ПК) и общими (ОК) компетенциями:

Код	Наименование результата обучения
ОК 1	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 2	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях
ОК 4	Эффективно взаимодействовать и работать в коллективе и команде
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 6	Проявлять гражданско-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
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 9	Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1	Собирать исходные данные для разработки проектной документации на информационную систему.
ПК 3.2	Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика.
ПК 3.3	Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием.
ПК 3.4	Производить разработку модулей информационной системы в соответствии с техническим заданием.
ПК 3.5	Интегрировать информационную систему с существующими информационными системами заказчика.
ПК 3.6	Осуществлять модульное и интеграционное тестирование информационной системы.
ПК 3.7	Разрабатывать техническую документацию на эксплуатацию информационной системы.

3. ОЦЕНКА УРОВНЕЙ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Оценивание уровней сформированности профессиональных и общих компетенций проводится в рамках текущего и промежуточного контроля.

В результате освоения профессионального модуля ПМ.03 Проектирование и разработка информационных систем студенты демонстрируют три уровня сформированности профессиональных компетенций: пороговый, базовый и повышенный.

Для каждого конкретного этапа формирования компетенции определены категории «знать», «уметь», «практический опыт», в которые вкладывается следующий смысл:

«приобрести практический опыт» – решать усложненные задачи на основе приобретенных умений и навыков, с их применением в профессиональных деятельности;

«уметь» – решать типичные задачи на основе воспроизведения стандартных алгоритмов решения;

«знать» - воспроизводить и объяснять учебный материал с требуемой степенью научной точности и полноты.

Код ОК, ПК	Уметь	Знать	Владеть навыками
ОК.01Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	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	-
ОК.02Использовать современные средства поиска, анализа и интерпретации информации, и	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной	номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации,	-

информационные технологии для выполнения задач профессиональной деятельности	деятельности	современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности в том числе с использованием цифровых средств	
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	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях	содержание актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования; основы предпринимательской деятельности; основы финансовой грамотности; правила разработки бизнес-планов; порядок выстраивания презентации; кредитные банковские продукты	-
ОК.04 Эффективно взаимодействовать и работать в коллективе и команде	Эффективно взаимодействовать и работать в коллективе и команде	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	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	особенности социального и культурного контекста; правила оформления документов и построения устных сообщений	-

особенностей социального и культурного контекста			
ОК.06 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации и межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	Проявлять гражданско-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	сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности; стандарты антикоррупционного поведения и последствия его нарушения	-
ОК.07 Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения ресурсосбережения; принципы бережливого производства; основные направления изменения климатических условий региона	-

ОК.08 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	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	-
ОК.09 Пользоваться профессиональной документацией на государственном и иностранном языках	Пользоваться профессиональной документацией на государственном и иностранном языках	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	-
ПК 3.1. Собирать исходные данные для разработки проектной документации и на информационную систему	– проводить сбор и анализ исходных данных для разработки проектной документации на информационную систему; – определять требования и функциональность информационной системы на основе собранных данных;	– основные принципы и методы сбора и анализа исходных данных для разработки проектной документации на информационную систему; – возможности типовой ИС; – предметную область автоматизации; – инструменты и методы выявления требований;	– сбора в соответствии с трудовым заданием документации заказчика касательно его запросов и потребностей применительно к типовой ИС; – анкетирования представителей заказчика в соответствии с трудовым заданием;

	– организовывать и управлять процессом сбора исходных данных для разработки проектной документации; – проводить анкетирование; - проводить интервьюирование	– технологии межличностной и групповой коммуникации в деловом взаимодействии, основы конфликтологии; – архитектуру, устройство и функционирование вычислительных систем; – коммуникационное оборудование; – сетевые протоколы; – основы современных операционных систем; – основы современных систем управления базами данных; – устройство и функционирование современных ИС; – современные стандарты информационного взаимодействия систем; – программные средства и платформы инфраструктуры информационных технологий организаций; – системы классификации и кодирования информации, в том числе присвоение кодов документам и элементам справочников; – отраслевую нормативную техническую документацию; – источники информации, необходимой для профессиональной деятельности; – современный	– интервьюирования представителей заказчика в соответствии с трудовым заданием; документирования собранных данных в соответствии с регламентами организации
--	---	--	---

		отечественный и зарубежный опыт в профессиональной деятельности; – основы бухгалтерского учета и отчетности организаций; – основы налогового законодательства российской федерации; – культуру речи; правила деловой переписки	
ПК 3.2. Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями и заказчика	– выбирать оптимальные технологии для реализации проекта; – разрабатывать планы проекта и управлять процессом разработки; – документировать проектную документацию в соответствии со стандартами и нормативными документами; - оценивать риски и принимать меры по их управлению	– методологию разработки информационных систем; – принципы и методы анализа требований заказчика; – методы проектирования информационных систем и их компонентов; – принципы и методы выбора технологий для реализации проекта; – методы оценки рисков и управления проектом; – методы документирования проектной документации; – стандарты и нормативные документов в области разработки информационных систем; – принципы и методы обеспечения безопасности информационных систем; - принципы и методы управления изменениями в информационных системах	разработки проектной документации для информационных систем
ПК 3.3.	– анализировать	– принципы	– разработки

Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием	требований безопасности информационных систем; – разрабатывать и реализовывать подсистемы безопасности информационных систем; тестировать и проводить отладку подсистем безопасности информационных систем	безопасности информационных систем; – современные методы и технологии в области безопасности информационных систем; законодательных и нормативных актов в области безопасности информационных систем	подсистем безопасности информационных систем; – применения современных методов и технологий в области безопасности информационных систем; оптимизации подсистем безопасности информационных систем
ПК 3.4. Производить разработку модулей информационной системы в соответствии с техническим заданием	– разрабатывать модули информационной системы с использованием выбранного языка программирования; – разрабатывать модули информационной системы в соответствии с требованиями, описанными в техническом задании; – разрабатывать API; организовывать взаимодействие модулей информационной системы	– языки программирования и работы с базами данных; – инструменты и методы модульного тестирования; – основы современных операционных систем; – основы современных систем управления базами данных; – устройство и функционирование современных ИС; – теорию баз данных; – системы хранения и анализа баз данных; – основы программирования; – современные объектно-ориентированные языки программирования; – современные структурные языки программирования; – языки современных бизнес-приложений; – современные методики тестирования разрабатываемых ИС; – современные стандарты информационного взаимодействия систем;	– разработки кода ИС и баз данных ИС в соответствии с техническим заданием; – верификации кода ИС и баз данных ИС относительно дизайна ИС и структуры баз данных ИС в соответствии с трудовым заданием; устранения обнаруженных несоответствий в соответствии с трудовым заданием

		– программные средства и платформы инфраструктуры информационных технологий организаций; – системы классификации и кодирования информации, в том числе присвоение кодов документам и элементам справочников; – отраслевую нормативную техническую документацию; – источники информации, необходимой для профессиональной деятельности; – основные языки программирования, такие как понимание принципов работы и особенностей выбранного языка программирования; – методологии разработки модулей информационной системы; – основные инструменты разработки, такие как среды разработки, системы контроля версий; - структуру и содержание технического задания	
ПК 3.5. Интегрировать информационную систему с существующими информационными	– работать в команде над интеграцией модулей в информационную систему; – выполнять интеграцию программный модулей в программный	– принципы интеграции информационной системы с другими системами; – современные технологии и инструменты для разработки интеграции информационной системы;	– интеграции информационной системы с существующими системами заказчика; – разработки API для интеграции информационной системы; – тестирования и

системами заказчика	продукт; – кодировать на языках программирования; находить и анализировать ключевые понятия и термины в сторонней документации для интеграции, а также разбираться в их контексте и использовании в рамках проекта.	– принципы тестирования и отладки интеграции информационной системы; – форматы обмена данными; интерфейсы обмена данными	отладки интеграции информационной системы; – проектирования интерфейсов обмена данными в соответствии с трудовым заданием; разработки интерфейсов обмена данными в соответствии с трудовым заданием
ПК 3.6. Осуществляют модульное и интеграционное тестирование информационной системы	– документировать тесты в соответствии с требованиями организации; – разрабатывать скрипты и/или программные модули для автоматизации; тестирования по, в том числе для проверки информационной безопасности разрабатываемого ПО; – оформлять тестовые случаи; – применять различные техники проектирования тестов (тест-дизайна); – применять универсальные языки моделирования (сценариев); – применять языки программирования для написания программного кода; – применять специализированное ПО для создания автотестов; – применять стандарты оформления кода; анализировать	– нормативно-технические материалов по вопросам испытания и тестирования ПО; – основные понятия о качестве ПО; – виды технической документации; – российские и международные стандарты тестирования информационных систем; – требования по обеспечению безопасности аппаратных и программных средств автоматизированных систем, используемых при выполнении тестовых процедур, включая вопросы антивирусной защиты; – основы работы в операционной системе, в которой производится тестирование, на уровне, необходимом для тестирования разработанного ПО; – классификация видов и типов тестирования ПО; – техники проектирования и комбинаторики тестов; – основы работы	– выделения классов эквивалентности значений каждого типа входных данных; – составления списка комбинаций значений из различных классов эквивалентности; – построения тестовых случаев, в которых сочетаются одна перестановка значений с необходимыми внешними ограничениями; – написания/настроек и программ для автоматизированного тестирования ПО; – разработки рабочих заданий по подготовке тестовых данных и выполнению тестовых процедур ПО; – описания тестовых случаев; разработки автоматизированных тестов, в том числе для проверки информационной безопасности разрабатываемого

	тестовые случаи на предмет полноты учета покрытия	необходимых приложений; – системы автоматизированного тестирования ПО; – языки программирования; тестовые данные, обеспечивающие проверку безопасности ПО	ПО
ПК 3.7. Разрабатывать техническую документацию на эксплуатацию информационной системы	– собирать и анализировать информацию о системе; – описывать процедуры установки и настройки системы; – описывать основные функции и возможности системы; – описывать процедуры обслуживания и регулярного обновления системы разрабатывать руководство пользователя	– принципы работы информационных систем; – процедуры установки и настройки системы; типы, виды и содержание документации на информационные системы в соответствии с ISO и ГОСТ на каждом этапе жизненного цикла информационных систем	– разработки технической документации на эксплуатацию информационной системы для компании; – участия в проекте по внедрению новой информационной системы в компанию, включая разработку соответствующей документации; проведения обучения пользователей по использованию информационной системы на основе разработанной документации

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ
ПМ.03 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ
(С ПРАВИЛЬНЫМИ ОТВЕТАМИ)**

№ пп	Содержание вопроса	Правильный ответ	Проверяемые компетенции
1.	... – это способ разобраться в сложной задаче перед тем, как начать программировать.	Системный анализ	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
2.	... – это список того, что система должна уметь делать и как должна работать.	Требования к информационной системе	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
3.	... – это любой человек или организация, которых затрагивает создание или работа системы.	Заинтересованное лицо (стейкхолдер)	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
4.	... – это беседа аналитика с пользователем или заказчиком для выяснения потребностей.	Интервью как метод сбора требований	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
5.	... – это сбор мнений и потребностей множества пользователей через письменные вопросы.	Анкетирование (опрос) как метод сбора требований	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
6.	... – это метод, когда аналитик смотрит, как пользователь выполняет задачу в реальной среде	Наблюдение за рабочим процессом	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7

			ОК 1-ОК 9
7.	... – это описание того, как пользователь взаимодействует с системой для достижения конкретной цели.	Вариант использования (Use Case)	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
8.	... – это короткая запись требования от лица пользователя в формате: «Как [роль], я хочу [действие], чтобы [ценность]».	Пользовательская история (User Story)	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
9.	... – это официальный документ, в котором подробно описаны все требования к будущей системе.	Спецификация требований к ПО (SRS)	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
10.	... – это таблица, которая связывает требования с другими артефактами проекта: задачами, кодом, тестами.	Матрица трассируемости требований	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
11.	... – это самостоятельная часть программы, решающая одну конкретную задачу.	Программный модуль	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
12.	... – это «договор» о том, как другие части программы могут обращаться к модулю.	Интерфейс модуля (контракт)	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
13.	... – это автоматическая проверка того, что отдельная функция или метод работает корректно.	Модульное тестирование (Unit Testing)	ПК 3.1 ПК 3.2 ПК 3.3

			ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
14.	... – это способ организации программы, где задачи разделены между «клиентом» (интерфейс) и «сервером» (логика и данные).	Клиент-серверная архитектура	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
15.	... – это трёхзначные коды, которые сервер возвращает клиенту, сообщая о результате запроса.	HTTP-статусы ответов сервера	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
16.	... – это механизм, который ловит сбои сети или задержки сервера и не даёт приложению «зависнуть» или упасть.	Обработка сетевых ошибок и таймаутов	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
17.	... – это набор правил и точек доступа, через которые одна программа может запрашивать данные или функции у другой.	API (Application Programming Interface)	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
18.	... – это самый распространённый стиль построения веб-API, основанный на стандартных HTTP-методах и работе с ресурсами.	REST API	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
19.	... – это программа, которая устанавливается и работает непосредственно на компьютере пользователя.	Настольное приложение (Desktop App)	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7

			ОК 1-ОК 9
20.	... – это процесс превращения исходного кода и всех зависимостей в готовый установочный файл, который пользователь может запустить.	Сборка и упаковка настольного приложения	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
21.	... – это повседневная работа по поддержанию системы в рабочем состоянии.	Техническая эксплуатация информационной системы	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
22.	... – это команда или инструмент, через который пользователи сообщают о проблемах с ИС.	Служба технической поддержки (Help Desk)	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
23.	Инцидент и запрос на обслуживание – в чём разница?	Инцидент — это «что-то сломалось», запрос — «хочу что-то новое или изменить».	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
24.	... – это инструмент, который постоянно «следит» за здоровьем системы и предупреждает о проблемах.	Система мониторинга ИС	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
25.	... – это документ, в котором прописано: что, когда и как делать для поддержания системы в порядке.	Регламент технического обслуживания	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
26.	... – это процесс проверки, что программа работает так, как задумано, и не ломается в неожиданных ситуациях.	Тестирование программного обеспечения	ПК 3.1 ПК 3.2 ПК 3.3

			ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
27.	... – это повторная проверка уже работавшего функционала после внесения изменений в код.	Регрессионное тестирование	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
28.	... – это процесс установки новой версии ПО для исправления ошибок, добавления функций или повышения безопасности.	Обновление (апдейт) информационной системы	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
29.	... – это экстренное возвращение системы к предыдущей, стабильной версии после неудачного обновления.	Откат версии (rollback)	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
30.	... – это копия продакшена, на которой проверяют обновления и новые функции перед выпуском.	Тестовая среда (стейджинг)	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9

ТЕСТ
(с правильными ответами)

№ пп	Содержание вопроса	Правильный ответ	Проверяемые компетенции
1.	Системный анализ – это ... а) процесс написания кода программы б) изучение предметной области для определения требований к будущей информационной системе в) тестирование готового программного продукта г) установка программного обеспечения на сервер	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
2.	Функциональные требования к системе описывают ...	в	ПК 3.1 ПК 3.2

	а) скорость работы и надёжность системы б) внешний вид пользовательского интерфейса в) что именно должна делать система (её функции и возможности) г) стоимость разработки и сроки внедрения		ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
3.	Нефункциональные требования к системе – это ... а) список функций, которые система не должна выполнять б) требования к качеству работы системы: производительность, безопасность, удобство в) пожелания пользователей, которые можно не учитывать г) требования, которые проверяются только после выпуска системы	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
4.	Заинтересованное лицо (стейкхолдер) в проекте разработки ИС – это ... а) только заказчик, который оплачивает разработку б) только программисты, которые пишут код в) любой человек или организация, которых затрагивает создание или использование системы г) только конечные пользователи системы	в	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
5.	Интервью как метод сбора требований – это ... а) рассылка письменных вопросов большому количеству пользователей б) беседа аналитика с пользователем или заказчиком для выяснения потребностей в) наблюдение за работой пользователя без его участия в диалоге г) изучение документации по аналогичным системам	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
6.	Анкетирование (опрос) целесообразно использовать, когда ... а) нужно глубоко изучить рабочий процесс одного специалиста б) пользователей много и необходимо собрать статистику по их потребностям в) требования уже собраны и нужно только задокументировать г) система очень сложная и требует прототипирования	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
7.	Вариант использования (Use Case) описывает ... а) техническую архитектуру системы и схему базы данных б) взаимодействие пользователя (актора) с системой для достижения конкретной цели в) список всех ошибок, которые могут возникнуть при работе системы г) план тестирования программного продукта	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
8.	Пользовательская история (User Story) обычно формулируется в формате ... а) «Если [условие], то [действие], иначе	в	ПК 3.1 ПК 3.2 ПК 3.3

	[альтернатива]» б) «Система должна [функция] со скоростью [значение]» в) «Как [роль], я хочу [действие], чтобы [ценность]» г) «Требование №[номер]: [описание]»		ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
9.	Спецификация требований к ПО (SRS) – это ... а) устная договорённость между заказчиком и разработчиком б) официальный документ с подробным описанием всех требований к системе в) черновик интерфейса, нарисованный от руки г) список исправленных ошибок в предыдущей версии системы	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
10.	Матрица трассируемости требований нужна для ... а) связи требований с задачами, кодом и тестами для контроля их реализации б) определения стоимости каждой функции системы в) распределения нагрузки между программистами команды г) визуализации пользовательского интерфейса в виде макетов	а	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
11.	Программный модуль – это ... а) самостоятельная часть программы, решающая одну конкретную задачу б) полный исходный код всего проекта в) графический интерфейс пользователя г) серверная часть веб-приложения	а	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
12.	Зачем модулю предоставлять публичный интерфейс (контракт)? а) чтобы код выглядел красивее в редакторе б) чтобы автоматически создавать резервные копии в) чтобы ускорить компиляцию проекта г) чтобы скрыть внутреннюю реализацию и менять её без поломок в зависимом коде	г	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
13.	Модульное (unit) тестирование применяется для ... а) проверки работы всей информационной системы в целом б) автоматической проверки корректности отдельной функции или метода в) оценки нагрузки на сервер при 1000 одновременных пользователей г) поиска уязвимостей безопасности в операционной системе	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
14.	Основной принцип клиент-серверной архитектуры заключается в ... а) выполнении всех вычислений на устройстве пользователя б) разделении задач: клиент отвечает за интерфейс,	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5

	сервер – за логику и хранение данных в) хранении всех данных исключительно в облаке без локального кэширования г) работе программы только при полном отсутствии интернета		ПК 3.6 ПК 3.7 ОК 1-ОК 9
15.	HTTP-статус 404 Not Found означает, что ... а) запрос выполнен успешно и данные возвращены б) на сервере произошла внутренняя ошибка в) запрошенный ресурс или страница не существует г) для доступа к ресурсу требуется авторизация	в	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
16.	Зачем в клиент-серверном приложении обрабатывать сетевые таймауты и ошибки? а) чтобы уменьшить размер установочного файла приложения б) чтобы приложение не «зависало» и корректно информировало пользователя о проблемах связи в) чтобы автоматически выбирать более быстрого интернет-провайдера г) чтобы увеличить скорость работы серверной базы данных	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
17.	API (Application Programming Interface) – это ... а) визуальный редактор для создания форм б) система управления версиями кода (например, Git) в) язык программирования для написания веб-сайтов г) набор правил и точек доступа, позволяющий разным программам обмениваться данными	г	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
18.	В REST API для обновления существующего ресурса обычно используется HTTP-метод ... а) GET б) POST в) PUT или PATCH г) DELETE	в	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
19.	Какое преимущество характерно для настольных (desktop) приложений по сравнению с веб-приложениями? а) они всегда работают быстрее любых других типов программ б) имеют прямой доступ к ресурсам ОС и могут функционировать без интернета в) автоматически обновляются без участия пользователя и администратора г) не требуют установки и запускаются прямо из браузера	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
20.	Сборка и упаковка настольного приложения нужна для ...	б	ПК 3.1 ПК 3.2

	а) сокращения количества строк в исходном коде б) превращения кода и всех зависимостей в готовый установочный файл для конечного пользователя в) шифрования программы от авторов-разработчиков г) замены локальной базы данных на облачное хранилище		ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
21.	Техническая эксплуатация информационной системы – это ... а) повседневная работа по поддержанию системы в рабочем состоянии б) процесс написания нового кода для расширения функционала в) однократная установка программы на компьютер пользователя г) обучение персонала работе с новым программным обеспечением	а	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
22.	Инцидент в процессе сопровождения ИС – это ... а) плановое обновление версии программного обеспечения б) незапланированное прерывание или ухудшение качества работы системы в) запрос пользователя на создание новой учётной записи г) документ с описанием архитектуры системы	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
23.	Система мониторинга ИС предназначена для ... а) автоматического написания кода программы б) создания резервных копий данных по расписанию в) постоянного наблюдения за параметрами системы и предупреждения о проблемах г) обучения новых сотрудников работе с системой	в	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
24.	Регламент технического обслуживания – это ... а) список всех известных ошибок в системе б) документ, определяющий что, когда и как делать для поддержания системы в порядке в) график отпусков сотрудников службы поддержки г) инструкция по установке операционной системы на сервер	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
25.	Тестирование программного обеспечения – это ... а) проверка того, что программа работает согласно требованиям и не содержит критических ошибок б) процесс продажи лицензий на использование программы в) написание технической документации для пользователей г) настройка сетевого оборудования для доступа к системе	а	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
26.	Регрессионное тестирование проводится для того, чтобы ... а) проверить работу системы на устаревшем	б	ПК 3.1 ПК 3.2 ПК 3.3

	оборудовании б) убедиться, что новые изменения не сломали ранее работавший функционал в) протестировать систему только на одном браузере г) оценить скорость работы системы при минимальной нагрузке		ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
27.	Тестовая среда (стейджинг) нужна для ... а) хранения архивных копий удалённых данных б) проверки обновлений и новых функций перед выпуском в продакшен в) проведения обучающих вебинаров для пользователей г) резервного копирования базы данных в облако	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
28.	Откат версии (rollback) применяется, когда ... а) необходимо удалить устаревшие логи для освобождения места на диске б) требуется изменить дизайн пользовательского интерфейса в) пользователь забыл свой пароль и нужно восстановить доступ г) после обновления система работает с критическими ошибками и нужно вернуться к стабильной версии	г	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
29.	Служба технической поддержки (Help Desk) работает преимущественно с ... а) исходным кодом программы для исправления багов б) разработкой маркетинговой стратегии продукта в) закупкой нового серверного оборудования для компании г) обращениями пользователей: инцидентами и запросами на обслуживание	г	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9
30.	Обновление (апдейт) информационной системы включает обязательный этап ... а) удаление всех пользовательских данных для ускорения работы б) создание резервной копии перед установкой новой версии в) отключение системы мониторинга на время работ г) смена паролей всех пользователей системы	б	ПК 3.1 ПК 3.2 ПК 3.3 ПК 3.4 ПК 3.5 ПК 3.6 ПК 3.7 ОК 1-ОК 9

4. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ

4.1. Типовые задания для оценки освоения МДК.03.01 Проектирование информационных систем

Вопросы для подготовки к экзамену/зачёту

1. Введение в системный анализ.
2. Цикл жизни информационной системы.
3. Классификация требований к ИС.
4. Методы выявления требований.
5. Интервьюирование как метод сбора информации.
6. Анализ существующих бизнес-процессов.
7. Структура функциональных требований.
8. Нефункциональные требования и их специфика.
9. Приоритизация требований.
10. Визуализация требований: нотации и схемы.
11. Диаграммы вариантов использования.
12. Диаграммы активностей.
13. Диаграммы состояний.
14. Диаграммы последовательностей.
15. Диаграммы классов.
16. Использование глоссариев в системном анализе.
17. Проверка полноты и непротиворечивости требований.
18. Методика построения спецификации требований.
19. Введение в модели прецедентов.
20. Поведение системы: событийные модели.
21. Моделирование объектов и атрибутов.
22. Диаграммы связей и отношений.
23. Интерпретация бизнес-логики через диаграммы.
24. Связь между требованиями и модулями.
25. Конфликт требований и методы их устранения.
26. Методы анализа потребностей заинтересованных сторон.
27. Документирование ограничений.
28. Переход от требований к архитектуре.
29. Учет требований безопасности.
30. Использование шаблонов требований.
31. Разработка модели данных.
32. Определение сущностей и связей.
33. Разработка ER-диаграмм.
34. Использование нормализации данных.
35. Проектирование интерфейсов с учетом требований.
36. Прототипирование интерфейсов пользователя.
37. Модели взаимодействия с системой.
38. Определение объемов информации.
39. Интеграционные требования.
40. Учет миграции данных в проекте.
41. Жизненный цикл требований.
42. Ревизия требований.
43. Версионирование требований.
44. Аудит требований.
45. Трассировка требований.

46. Формирование матрицы соответствия требований.
47. Документирование сценариев использования.
48. Обоснование необходимости требований.
49. Анализ рисков на этапе сбора требований.
50. Участие команды в согласовании требований

Примеры практических заданий

1. Проведение интервью с «заказчиком». Составление списка требований.
2. Формализация требований с использованием таблиц. Разработка диаграммы прецедентов.
3. Построение диаграммы активностей. Создание диаграммы состояний для объекта. Создание диаграммы классов.
4. Определение функциональных требований. Уточнение нефункциональных требований.
5. Приоритизация требований методом MoSCoW. Разработка модели данных.
6. Проектирование ER-диаграммы
7. Нормализация таблиц до 3НФ
8. Определение сущностей и связей
9. Разработка глоссария проекта. Моделирование бизнес-процесса
10. Построение карты заинтересованных сторон. Анализ сценариев взаимодействия.
11. Проектирование интерфейса пользователя. Проработка шаблона спецификации требований.
12. Разработка прототипа интерфейса. Подготовка таблицы ограничений.
13. Идентификация рисков на этапе анализа. Оценка полноты требований по чек-листу.
14. Формирование структуры ТЗ. Построение модели прецедентов.
15. Анализ конфликта требований. Составление матрицы соответствия.
16. Сопоставление требований и модулей. Создание модели поведения системы.
17. Определение объемов данных на входе/выходе. Работа с гипотетическим заказчиком.
18. Сценарное моделирование. Формализация альтернативных потоков данных.
19. Выявление точек интеграции. Описание функциональных блоков.
20. Разработка и ревизия требований. Работа с фреймами требований. Подготовка презентации требований.
21. Составление и согласование технического задания. Разработка примеров пользовательских историй. Описание ограничений и допущений
22. Визуализация модели данных. Описание возможных изменений в требованиях.
23. Моделирование переходов между состояниями. Разработка примеров диаграмм взаимодействия.
24. Сбор требований по сценарному описанию. Ведение журнала требований.
25. Инспекция требований в группе. Подготовка отчета о завершении анализа требований

4.2. Типовые задания для оценки освоения МДК.03.02 Разработка кода информационных систем

Вопросы для подготовки к экзамену/зачёту

1. Принципы модульного программирования
2. Организация и структура программного проекта
3. Основы процедур и функций
4. Объектно-ориентированное проектирование модулей
5. Принципы инкапсуляции и абстракции
6. Типы данных и структуры хранения
7. Работа с файлами и файловыми потоками
8. Ввод/вывод в консольных приложениях
9. Работа со строками и регулярные выражения
10. Обработка исключений и управление ошибками
11. Использование модульной библиотеки
12. Принципы повторного использования кода

13. Взаимодействие модулей в пределах одной системы
14. Разработка вспомогательных утилит и сервисов
15. Работа с датой и временем
16. Принципы инициализации и завершения программ
17. Разработка кода для работы с сетью (TCP, UDP)
18. Введение в асинхронное программирование
19. Протокол HTTP: структура и использование
20. Обработка входящих и исходящих запросов
21. Работа с сериализацией и десериализацией данных
22. Обработка JSON и XML
23. Основы многопоточности
24. Состояния и события в пользовательской логике
25. Работа с конфигурационными файлами
26. Тестируемость и структурированность кода
27. Использование шаблонов проектирования (Factory, Singleton)
28. Устойчивость к сбоям и восстановление
29. Принципы локализации и интернационализации
30. Разработка CLI-интерфейса
31. Введение в клиент-серверную архитектуру.
32. Сетевые протоколы: TCP/IP и UDP.
33. Архитектура REST.
34. Структура HTTP-запроса и ответа.
35. Принципы проектирования API.
36. Методы GET, POST, PUT, DELETE
37. Коды состояний HTTP.
38. Документирование API
39. Аутентификация в API.
40. Обработка ошибок в API
41. Подходы к построению GUI.
42. Событийно-ориентированное программирование
43. Архитектура настольных приложений.
44. Связывание пользовательского интерфейса с логикой
45. Работа с таблицами и формами.
46. Обработка событий нажатия
47. Динамическое обновление интерфейса.
48. Построение графиков и визуализация данных
49. Состояния компонентов интерфейса.
50. Сокеты и двусторонняя передача данных
51. Связь клиента и сервера через сокеты.
52. Обмен файлами через сеть
53. Подключение к базе данных.
54. Работа с запросами SELECT, INSERT, UPDATE, DELETE
55. ORM и абстракции уровня данных.
56. Связь клиентского интерфейса и базы данных
57. Механизмы кеширования и обновления.
58. Модель жизненного цикла клиент-серверного приложения.
59. Регистрация, авторизация и сессии.
60. Отладка и профилирование сетевого взаимодействия
61. Дайте определение беспилотной транспортной системы (БТС). В чём её принципиальные отличия от традиционных транспортных средств?
62. Перечислите и кратко охарактеризуйте основные типы БТС (наземные, воздушные, водные, гибридные).

63. Опишите классификацию БТС по степени автономности (уровни SAE от 0 до 5). Приведите примеры систем для каждого уровня.
64. Каковы основные сферы применения БТС в современном мире? Приведите 3–4 конкретных примера с указанием решаемых задач.
65. Назовите ключевые исторические вехи в развитии беспилотного транспорта (от первых прототипов до современных решений).

Примеры практических заданий

1. Разработка функции обработки строки с регулярными выражениями
2. Создание модуля для работы с файлами. Написание обработчика исключений.
3. Создание структуры данных для хранения пользовательской информации.
4. Реализация модуля сериализации/десериализации. Разработка текстового меню управления модулем
5. Проектирование и реализация TCP-сервера
6. Написание клиента, отправляющего JSON-запрос
7. Обработка запроса по HTTP. Разработка сервиса фильтрации логов.
8. Сборка консольного приложения из нескольких модулей. Пример использования асинхронных вызовов.
9. Реализация многопоточного обработчика задач. Работа с датами и их форматами
10. Разработка утилиты для чтения конфигурации.
11. Валидация пользовательского ввода.
12. Пример реализации шаблона «Одиночка»
13. Разработка класса-обёртки для API вызовов.
14. Построение интерфейса взаимодействия между модулями
15. Реализация CLI-утилиты для работы с файлами. Пример использования событийного механизма.
16. Создание консольной игры с многомодульной архитектурой. Подключение внешней библиотеки через модуль.
17. Имитация падения и восстановление процесса. Реализация функции локализации сообщений.
18. Тестирование отдельных модулей вручную. Создание модульной библиотеки с открытым API.
19. Интеграция пользовательского ввода с логикой обработки. Обработка XML-файла через модуль конфигурации.
20. Рефакторинг кода для повышения читаемости и стабильности
21. Реализация простого REST-сервиса
22. Разработка API для списка задач. Тестирование API через запросы POST и GET.
23. Создание документации для API. Пример клиента с использованием API
24. Разработка интерфейса с таблицей. Интеграция событий интерфейса с логикой.
25. Построение формы ввода с валидацией. Реализация обработки кнопок и меню.
26. Динамическое изменение элементов интерфейса.
27. Подключение к СУБД.
28. Выполнение базовых SQL-запросов
29. Работа с транзакциями
30. Отображение данных из БД в интерфейсе. Сохранение введенных данных в БД.
31. Реализация авторизации пользователя
32. Реализация REST API для CRUD-операций
33. Создание логики отображения пользовательских данных
34. Передача файлов через сокет
35. Разработка клиента с сокетами
36. Настройка кеширования в клиенте
37. Пример использования ORM

38. Уведомления об обновлении данных
39. Подключение API к настольному приложению
40. Настройка логирования сетевых вызовов
41. Обработка ошибок соединения
42. Пример загрузки/выгрузки данных
43. Упрощённый чат на сокетах
44. Работа с конфигурацией соединений
45. Разработка клиент-серверного учебного проекта

4.3. Типовые задания для оценки освоения МДК.03.03 Сопровождение информационных систем

Вопросы для подготовки к экзамену/зачёту

1. Цели и задачи сопровождения ИС.
2. Этапы ввода системы в промышленную эксплуатацию
3. Мониторинг работоспособности компонентов.
4. Уровни поддержки: 1-я, 2-я, 3-я линии
5. Ведение документации по сопровождению
6. Управление инцидентами и заявками
7. Модели управления ИТ-услугами (ITIL, COBIT)
8. Планирование обслуживания и обновлений
9. Резервное копирование: стратегии и частота
10. Восстановление после сбоев и катастроф
11. Журналирование и логирование в ИС
12. Работа с SLA и контроль выполнения
13. Учет программных и аппаратных ресурсов
14. Управление конфигурациями компонентов
15. Ведение технического паспорта системы
16. Контроль лицензий и версий ПО
17. Регламенты обработки инцидентов
18. Миграция данных и платформ
19. Подготовка систем к масштабированию
20. Взаимодействие с пользователями и обучение
21. Виды тестирования: модульное, интеграционное, системное
22. Автоматизация тестирования: цели и инструменты
23. Тест-кейсы и тест-планы
24. Отчеты о тестировании и анализ дефектов
25. CI/CD как часть поддержки ИС
26. Роль тестов в процессе выпуска обновлений
27. Тестирование безопасности и уязвимостей
28. Тестирование производительности
29. Стресс-тестирование и нагрузочное тестирование
30. Инструменты управления тестированием
31. Обратная совместимость при обновлениях
32. Контроль версий и миграции
33. Введение в управление изменениями
34. Построение стратегии релизов
35. Тестирование интерфейсов и UX
36. Приемочное тестирование
37. Документация по тестированию
38. Поддержка модульных и функциональных тестов
39. Ручное и автоматическое регрессионное тестирование
40. Практики DevOps в процессе сопровождения

Примеры практических заданий

1. Настройка логирования и журналирования событий
2. Разработка схемы резервного копирования
3. Моделирование инцидента и формирование заявки
4. Оформление отчёта по уровню SLA
5. Подготовка регламента ввода ИС в эксплуатацию
6. Настройка мониторинга ресурсов приложения
7. Проведение процедуры восстановления после сбоя
8. Создание базы знаний для технической поддержки
9. Разработка чек-листа для технической диагностики
10. Анализ и интерпретация логов системы
11. Разработка тест-кейсов для проверки бизнес-функции
12. Проведение модульного тестирования с логированием
13. Написание скрипта автоматизированного теста
14. Сборка пайплайна CI с шагом тестирования
15. Имитация инцидента и проверка фикса в новой версии
16. Разработка регрессионного набора тестов
17. Проведение нагрузочного тестирования
18. Сравнительный анализ версии «до» и «после» обновления
19. Проверка совместимости компонентов при миграции
20. Генерация и оформление отчёта о тестировании

Вопросы для устного опроса по разделам по МДК.03.01 Проектирование информационных систем

1. Что такое системный анализ и какова его роль в жизненном цикле информационной системы?
2. Объясните разницу между функциональными и нефункциональными требованиями. Приведите по два примера каждого типа для системы «Электронный журнал».
3. Какие основные методы сбора требований вы знаете? В каких ситуациях целесообразно использовать интервью, а в каких — анкетирование?
4. Что такое стейкхолдер? Назовите основные группы заинтересованных лиц в проекте разработки учебного портала и их типичные потребности.
5. Как проводится анализ существующих бизнес-процессов? Зачем это нужно перед автоматизацией?
6. Что такое приоритизация требований? Опишите метод MoSCoW и приведите пример распределения требований по категориям для мобильного приложения колледжа.
7. Для чего используются диаграммы вариантов использования (Use Case)? Опишите структуру одного варианта использования на примере «Записаться на консультацию к преподавателю».
8. В чём разница между диаграммой активностей и диаграммой состояний? Приведите пример бизнес-процесса, который удобнее моделировать каждым из этих инструментов.
9. Когда применяется диаграмма последовательностей? Изобразите устно последовательность сообщений при авторизации пользователя в системе.
10. Что показывает диаграмма классов? Назовите основные элементы (класс, атрибут, метод, связь) и приведите пример для предметной области «Библиотека».
11. Как интерпретировать бизнес-логику через диаграммы? Почему визуальное моделирование помогает избежать недопонимания между аналитиком и разработчиком?
12. Что такое глоссарий в системном анализе? Зачем он нужен и какие термины обязательно включать в глоссарий проекта «Система учёта практики студентов»?
13. Из каких разделов состоит спецификация требований к ПО (SRS)? Почему этот документ считается «договором» между заказчиком и разработчиками?

14. Как проверить требования на полноту и непротиворечивость? Назовите 3–4 типичных признака проблемных требований.
15. Что такое трассировка требований? Как матрица соответствия помогает контролировать реализацию и тестирование?
16. Объясните жизненный цикл требования: от выявления до вывода из эксплуатации. На каких этапах требование может быть изменено или отклонено?
17. Что такое версионирование требований? Зачем фиксировать изменения в требованиях и как это влияет на команду разработки?
18. Как проводится аудит требований? Кто участвует в этом процессе и какие артефакты проверяются?
19. Что такое сущность, атрибут и связь в модели данных? Приведите пример для предметной области «Расписание занятий».
20. Для чего используются ER-диаграммы? Опишите основные типы связей (1:1, 1:M, M:M) и способы их отображения.
21. Что такое нормализация данных? Объясните цель первых трёх нормальных форм и приведите пример устранения избыточности.
22. Как требования к системе влияют на структуру модели данных? Приведите пример, когда изменение функционального требования требует пересмотра схемы БД.
23. Что такое интеграционные требования? Почему их важно учитывать на этапе проектирования модели данных?
24. Как учитывать миграцию данных при проектировании новой системы? Какие риски возникают при переносе данных из устаревшей системы?
25. Как требования к пользовательскому интерфейсу формулируются на этапе системного анализа? Что такое прототип и зачем его создавать до начала разработки?
26. Какие модели взаимодействия с системой вы знаете (синхронное/асинхронное, запрос-ответ, событийная)? Приведите пример использования каждой в учебном проекте.
27. Что такое ограничения (constraints) в требованиях? Назовите типы ограничений (технические, правовые, ресурсные) и их влияние на архитектуру системы.
28. Как учитывать требования безопасности на этапе сбора требований? Приведите примеры требований к аутентификации, авторизации и аудиту для системы с персональными данными.
29. Что делать, если требования от разных стейкхолдеров конфликтуют? Опишите алгоритм выявления и разрешения конфликтов требований.
30. Почему важно вовлекать команду разработки в согласование требований? Как раннее участие программистов и тестировщиков снижает риски проекта?

по МДК.03.02 Разработка кода информационных систем

1. Что такое модульное программирование и как правильная организация проекта влияет на повторное использование кода и командную разработку?
2. В чём разница между процедурой и функцией? Как принципы инкапсуляции и абстракции помогают проектировать независимые и понятные модули?
3. Какие шаблоны проектирования (например, Factory, Singleton) вы знаете? В каких реальных сценариях их применение оправдано, а когда оно избыточно?
4. Как организуется взаимодействие модулей в пределах одной системы? Какие зависимости допустимы, а какие нарушают архитектуру?
5. Что делает код тестируемым и структурированным? Как модульность упрощает написание unit-тестов и изоляцию ошибок?
6. Как происходит инициализация и корректное завершение программы? Зачем используются конфигурационные файлы и как их безопасно читать/изменять?
7. Какие типы данных и структуры хранения вы чаще всего используете в проектах? Как выбор структуры влияет на скорость доступа и потребление памяти?
8. Опишите работу с файлами и файловыми потоками. В чём особенности консольного

ввода/вывода по сравнению с файловым и сетевым?

9. Как в вашем языке программирования реализована работа со строками и регулярными выражениями? Приведите пример задачи, где регулярки незаменимы.

10. Как правильно обрабатывать исключения? Приведите пример, когда «глушение» ошибки (try...catch pass) приводит к скрытому сбою, и как этого избежать.

11. Что такое сериализация и десериализация? Как происходит преобразование объектов в JSON/XML и обратно, и где это применяется на практике?

12. Как спроектировать модуль, устойчивый к сбоям? Опишите механизм повторных попыток (retry), логирования ошибок и graceful shutdown.

13. Что такое клиент-серверная архитектура? Как работают сокеты и чем на практике отличаются протоколы TCP и UDP?

14. Опишите структуру HTTP-запроса и ответа. Как сервер принимает входящий запрос, маршрутизирует его и формирует ответ?

15. Что такое REST и какие принципы лежат в основе проектирования API? Когда и для чего используются методы GET, POST, PUT, DELETE?

16. Что означают группы кодов состояния HTTP (2xx, 3xx, 4xx, 5xx)? Как правильно обрабатывать и возвращать ошибки в API, чтобы клиент мог на них отреагировать?

17. Зачем документировать API и какие инструменты для этого вы знаете? Какие методы аутентификации (API-ключ, JWT, OAuth, Basic) вы применяли или изучали?

18. Как организовать двустороннюю передачу данных через сокет? В каких сценариях сокет предпочтительнее HTTP, и как реализовать обмен файлами по сети?

19. Какие подходы к построению графического интерфейса (GUI) вы знаете? Что такое событийно-ориентированное программирование и как оно работает в десктопных приложениях?

20. Как связывается пользовательский интерфейс с бизнес-логикой? Опишите типичный путь данных от нажатия кнопки до обновления экрана.

21. Как реализовать динамическое обновление интерфейса без перезагрузки окна? Для чего нужны состояния компонентов и как управлять их жизненным циклом?

22. Как работать с таблицами и формами в настольном приложении? Как обрабатываются события ввода, валидации и отправки данных?

23. Как встроить построение графиков или визуализацию данных в десктопное приложение? Какие библиотеки или компоненты вы использовали?

24. Сравните архитектуру настольного приложения с прямым доступом к БД и архитектуру, где UI обращается к локальному API/сервису. Какие плюсы и минусы у каждого подхода?

25. Как устанавливается подключение к базе данных из приложения? Опишите назначение и базовый синтаксис запросов: SELECT, INSERT, UPDATE, DELETE.

26. Что такое ORM и зачем нужна абстракция уровня данных? Как клиентский интерфейс связывается с БД через серверную часть, и какие риски есть при прямом доступе?

27. Зачем нужны механизмы кеширования и как они влияют на производительность клиент-серверного приложения? Опишите типичный жизненный цикл такого приложения.

28. Как реализуется регистрация, авторизация и поддержка сессий? Где и как безопасно хранить токены или идентификаторы сессии?

29. В чём разница между синхронным, асинхронным и многопоточным выполнением кода? В каких задачах многопоточность даёт выигрыш, а когда создаёт гонки данных или deadlock?

30. Что такое отладка и профилирование сетевого взаимодействия? Какими инструментами (Postman, Wireshark, логи, профайлеры) вы пользовались для поиска проблем в API или сетевых модулях?

31. Изобразите в виде схемы основные подсистемы БТС и укажите их взаимосвязи.

32. Какие датчики используются в БТС для восприятия окружающей среды? Перечислите 5 основных типов и кратко опишите их назначение.

33. Как работают лидары? В чём их преимущества и недостатки по сравнению с радарами и камерами?

34. Какова роль бортовых вычислительных систем в БТС? Какие требования к ним предъявляются (производительность, надёжность, энергопотребление)?

35. Перечислите основные исполнительные механизмы в наземных БТС. Как они интегрированы в систему управления?

36. Сравните варианты энергоснабжения БТС: ДВС, гибридные и электрические системы. Укажите плюсы и минусы каждого варианта.

по МДК.03.03 Сопровождение информационных систем

1. Каковы основные цели и задачи сопровождения информационной системы после её ввода в эксплуатацию?

2. Опишите ключевые этапы ввода системы в промышленную эксплуатацию. Что может пойти не так на каждом этапе и как это предотвратить?

3. Что такое мониторинг работоспособности компонентов ИС? Какие метрики (ЦП, память, диск, сеть, ошибки) важно отслеживать в первую очередь и почему?

4. В чём разница между уровнями поддержки (1-я, 2-я, 3-я линии)? Как инцидент эскалируется между ними и какие компетенции требуются на каждом уровне?

5. Какие стратегии резервного копирования вы знаете (полная, инкрементальная, дифференциальная)? Как выбрать частоту бэкапов для разных типов данных?

6. Что такое план восстановления после сбоев (DRP)? Чем отличается восстановление данных из бэкапа от аварийного восстановления всей инфраструктуры?

7. Зачем нужно журналирование и логирование в ИС? Какие события обязательно фиксировать в логах для последующего анализа инцидентов и аудита?

8. Что такое SLA (Service Level Agreement) и как контролируется его выполнение? Какие последствия могут быть при систематическом нарушении соглашений об уровне обслуживания?

9. Как организуется управление инцидентами и заявками в службе поддержки? В чём ключевая разница между инцидентом (сбой) и запросом на обслуживание (изменение)?

10. Что такое модели ITIL и COBIT? Как они помогают стандартизировать процессы сопровождения ИС и повысить предсказуемость результатов?

11. Как планируются работы по обслуживанию и обновлению системы? Что такое «окно обслуживания» и как минимизировать простой для пользователей?

12. Зачем ведётся документация по сопровождению? Какие документы (регламенты, инструкции, схемы) обязательно обновлять при каждом изменении системы?

13. Что такое технический паспорт системы и какие разделы он должен содержать (архитектура, зависимости, контакты, процедуры восстановления)?

14. Как осуществляется учёт программных и аппаратных ресурсов? Зачем нужен реестр конфигураций (CMDB) и как он помогает при диагностике проблем?

15. Почему важен контроль лицензий и версий ПО? Какие юридические и технические риски возникают при использовании нелегального или устаревшего ПО?

16. Как организуется взаимодействие с пользователями в процессе сопровождения? Зачем проводить обучение и как оценивать его эффективность (опросы, метрики, обратная связь)?

17. Какие виды тестирования применяются в процессе сопровождения ИС? В чём разница между модульным, интеграционным и системным тестированием?

18. Что такое тест-кейс и тест-план? Как они связаны между собой и с требованиями к системе? Приведите пример тест-кейса для функции авторизации.

19. Как автоматизация тестирования ускоряет процесс сопровождения? Какие задачи (регресс, smoke, API-тесты) целесообразно автоматизировать в первую очередь?

20. Что должно входить в отчёт о тестировании? Как анализ дефектов (классификация, корневые причины) помогает предотвращать повторные сбои?

21. Какую роль играют тесты в процессе выпуска обновлений? Почему нельзя выпускать релиз без прохождения регрессионного тестирования?

22. Что такое тестирование безопасности и как выявляются уязвимости в работающей

системе (сканирование, пентест, анализ зависимостей)?

23. В чём разница между нагрузочным и стресс-тестированием? Когда и зачем проводится каждый из этих видов тестов и какие метрики анализируются?

24. Какие инструменты управления тестированием вы знаете (Jira, TestRail, Zephyr)? Как они помогают координировать работу команды качества и разработчиков?

25. Что такое обратная совместимость при обновлениях? Почему её нарушение может привести к проблемам у пользователей и как её проверять перед релизом?

26. Как организован контроль версий и миграции данных при обновлении системы? Что делать, если миграция прошла с ошибками (откат, hotfix, ручное исправление)?

27. Что такое управление изменениями (Change Management)? Зачем нужна формальная процедура согласования изменений в продакшене и кто участвует в CAB (Change Advisory Board)?

28. Как строится стратегия релизов? В чём разница между частыми минорными обновлениями (патчи, фичи) и редкими мажорными релизами (архитектурные изменения)?

29. Как проводится тестирование интерфейсов и пользовательского опыта (UX)? Почему важно тестировать не только функционал, но и удобство, доступность, локализацию?

30. Что такое приемочное тестирование (UAT) и кто его проводит? Как результаты приемки влияют на решение о выпуске обновления в продакшен и как фиксировать замечания?

Критерии оценки:

Отметка «5»: ответ полный и правильный на основании изученных теорий; материал изложен в определенной логической последовательности, литературным языком. Ответ самостоятельный.

Отметка «4»: ответ полный и правильный на основании изученных теорий; материал изложен в определенной логической последовательности, при этом допущены две-три несущественные ошибки, исправленные по требованию преподавателя.

Отметка «3»: ответ полный, но при этом допущена существенная ошибка, или неполный, несвязный.

Фонд тестовых заданий

по МДК.03.01 Проектирование информационных систем

Тип 1: Один правильный ответ

1. Системный анализ в жизненном цикле ИС – это ...

- а) этап написания программного кода
- б) процесс изучения предметной области и формирования требований к системе
- в) тестирование готового продукта перед сдачей заказчику
- г) обучение пользователей работе с интерфейсом

2. Функциональные требования описывают ...

- а) что именно должна делать система (функции, действия, реакции)
- б) скорость загрузки страниц и время отклика сервера в) требования к шифрованию и аудиту доступа
- г) бюджет и сроки разработки

3. Нефункциональные требования – это ...

- а) список функций, которые система выполнять не будет
- б) пожелания пользователей без бизнес-обоснования
- в) ограничения и характеристики качества работы системы (производительность, безопасность, удобство)
- г) требования, проверяемые только после вывода системы из эксплуатации

4. Интервью как метод сбора требований применяется для ...

- а) массового сбора мнений от тысяч пользователей
- б) глубокого изучения потребностей конкретного стейкхолдера или эксперта

- в) автоматического формирования отчётов по логам системы
- г) визуализации диаграмм классов

5. Метод MoSCoW используется для ...

- а) приоритизации требований по категориям: Must, Should, Could, Won't
- б) шифрования конфиденциальных данных в требованиях
- в) моделирования бизнес-процессов в нотации BPMN
- г) трассировки требований к тест-кейсам

6. Диаграмма вариантов использования (Use Case) показывает ...

- а) внутреннюю архитектуру сервера и схемы БД
- б) состояния объекта и переходы между ними
- в) последовательность вызовов методов во времени
- г) взаимодействие акторов с системой для достижения конкретных целей

7. Диаграмма последовательностей отображает ...

- а) статическую структуру классов в системе
- б) временной порядок обмена сообщениями между объектами или компонентами
- в) распределение прав доступа пользователей
- г) этапы тестирования программного модуля

8. На диаграмме классов связи между объектами называются ...

- а) события
- б) транзакции
- в) сценарии
- г) ассоциации

9. Глоссарий в системном анализе необходим для ...

- а) ускорения компиляции исходного кода
- б) единого понимания терминов всеми участниками проекта
- в) автоматической генерации тест-кейсов
- г) шифрования спецификации требований

10. Спецификация требований (SRS) – это ...

- а) устная договорённость с заказчиком
- б) формализованный документ, описывающий все требования, ограничения и интерфейсы системы
- в) черновик пользовательского интерфейса
- г) план маркетингового продвижения продукта

11. ER-диаграмма используется для ...

- а) моделирования сетевых протоколов
- б) планирования спринтов в Agile
- в) описания пользовательских сценариев
- г) визуализации сущностей, их атрибутов и связей в базе данных

12. Нормализация данных проводится с целью ...

- а) увеличения дублирования информации для ускорения чтения
- б) устранения избыточности и аномалий при вставке/обновлении/удалении
- в) преобразования реляционной модели в графовую
- г) автоматического создания резервных копий

13. Прототипирование интерфейсов позволяет ...

- а) заменить разработку бэкенда
- б) быстро согласовать структуру и логику UI до начала кодирования
- в) провести нагрузочное тестирование сервера
- г) автоматически сгенерировать ER-диаграмму

14. Жизненный цикл требования начинается с ...

- а) выявления и документирования
- б) удаления из системы после релиза

в) автоматического тестирования

г) шифрования в хранилище

15. Трассировка требований – это ...

а) поиск ошибок в исходном коде

б) отслеживание связи требования от источника через реализацию к тестам

в) логирование действий пользователей в системе

г) резервное копирование документации

16. Требование «Все пароли должны храниться в захешированном виде» относится к ...

а) функциональным требованиям

б) интеграционным требованиям

в) требованиям безопасности

г) требованиям к миграции

17. При конфликте требований от разных стейкхолдеров аналитик должен ...

а) выбрать требование того, кто громче настаивает

б) автоматически отклонить все требования

в) игнорировать конфликт до этапа тестирования

г) выявить бизнес-ценность, приоритет и предложить компромиссное решение

18. Анализ рисков на этапе сбора требований помогает ...

а) заранее выявить угрозы срыву сроков, бюджета или безопасности системы

б) сократить количество программистов в команде

в) отменить этап проектирования

г) увеличить стоимость лицензий

19. Учет миграции данных важен для ...

а) обеспечения переноса исторических данных из старой системы в новую без потерь

б) замены аппаратных серверов на виртуальные

в) ускорения работы компилятора

г) шифрования сетевого трафика

20. Участие команды разработки в согласовании требований нужно для ...

а) снижения зарплаты аналитиков

б) оценки технической реализуемости, сроков и выявления скрытых зависимостей

в) замены документации устными инструкциями

г) автоматического написания кода

№	Ответ	№	Ответ
1	б	11	г
2	а	12	б
3	в	13	б
4	б	14	а
5	а	15	б
6	г	16	в
7	б	17	г
8	г	18	а
9	б	19	а
10	б	20	б

Тип 2: Несколько правильных ответов

21. Какие из перечисленных являются НЕФУНКЦИОНАЛЬНЫМИ требованиями? (выберите 3)

а) Система должна формировать отчёт за месяц

б) Время отклика интерфейса не более 2 сек

в) Поддержка 1000 одновременных сессий

- г) Возможность экспорта данных в PDF
 - д) Хранение логов 90 дней с шифрованием
- 22. Какие диаграммы UML относятся к моделированию ПОВЕДЕНИЯ системы? (выберите 3)**
- а) Диаграмма классов
 - б) Диаграмма активностей
 - в) Диаграмма состояний
 - г) Диаграмма последовательностей
 - д) ER-диаграмма
- 23. Какие методы сбора требований относятся к глубинным (качественным)? (выберите 2–3)**
- а) Анкетирование 500 пользователей
 - б) Интервью с экспертами предметной области
 - в) Наблюдение за рабочим процессом
 - г) Анализ аналогичных систем
 - д) Массовая рассылка SMS
- 24. Какие элементы входят в концептуальную модель данных? (выберите 3)**
- а) Сущности
 - б) Атрибуты
 - в) Связи (1:1, 1:M, M:M)
 - г) HTTP-заголовки
 - д) CSS-стили
- 25. Какие ограничения обычно документируются в требованиях? (выберите 3)**
- а) Технические (ОС, СУБД, языки)
 - б) Правовые (152-ФЗ, лицензирование)
 - в) Ресурсные (бюджет, сроки, команда)
 - г) Цветовая палитра логотипа
 - д) Любимые браузеры заказчика
- 26. Какие артефакты используются для управления изменениями требований? (выберите 3)**
- а) Матрица трассировки
 - б) Журнал версий требований
 - в) Протокол ревизии (review)
 - г) Диаграмма Ганта проекта
 - д) Файл стилей UI
- 27. При определении объёмов информации учитываются ... (выберите 3)**
- а) Ожидаемое количество записей в год
 - б) Средний размер файла/документа
 - в) Рост данных за 3–5 лет
 - г) Количество кнопок в интерфейсе
 - д) Версия браузера пользователя
- 28. При анализе бизнес-процессов аналитик выявляет ... (выберите 3)**
- а) Входные и выходные данные процесса
 - б) Ответственных исполнителей и роли
 - в) Узкие места и ручной труд
 - г) Исходный код модулей
 - д) Пароли от серверов
- 29. Связь между требованиями и модулями системы отражается в ... (выберите 2)**
- а) Матрице соответствия требований
 - б) Архитектурной схеме компонентов
 - в) Файле robots.txt
 - г) README.md репозитория
- 30. Матрица трассировки требований связывает ... (выберите 3)**
- а) Бизнес-цели с требованиями

- б) Требования с модулями/функциями
- в) Требования с тест-кейсами
- г) Тесты с зарплатой разработчиков
- д) Серверы с IP-адресами

№	Правильные варианты
21	б, в, д
22	б, в, г
23	б, в, г
24	а, б, в
25	а, б, в
26	а, б, в
27	а, б, в
28	а, б, в
29	а, б
30	а, б, в

Тип 3: Верно / Неверно

31. Визуализация требований с помощью нотаций полностью заменяет текстовое описание спецификации.

- ☐ Верно
- ☐ Неверно

32. Первая нормальная форма требует устранения повторяющихся групп атрибутов и атомарности значений.

- ☐ Верно
- ☐ Неверно

33. Диаграмма связей и отношений показывает временную последовательность вызовов методов.

- ☐ Верно
- ☐ Неверно

34. Ревизия требований проводится только один раз – на этапе проектирования.

- ☐ Верно
- ☐ Неверно

35. Модели прецедентов описывают, как система реагирует на внешние события для достижения целей пользователей.

- ☐ Верно
- ☐ Неверно

№	Ответ
31	Неверно (Нотации дополняют, но не заменяют текст: нужны ограничения, бизнес-правила, исключения)
32	Верно
33	Неверно (Время показывает диаграмма последовательностей; связи – статические зависимости)
34	Неверно (Ревизии проводятся регулярно на протяжении всего жизненного цикла)
35	Неверно

Тип 4: Установите соответствие

36. Соотнесите диаграмму с её назначением:

- | | |
|--------------------------|---|
| 1. Диаграмма активностей | А. Статическая структура данных и методов |
| 2. Диаграмма состояний | Б. Взаимодействие акторов и целей системы |

- | | |
|--------------------------------------|-------------------------------------|
| 3. Диаграмма классов | В. Этапы выполнения бизнес-процесса |
| 4. Диаграмма вариантов использования | Г. Переходы объекта между статусами |

37. Соотнесите тип требования с примером:

- | | |
|--|---|
| 1. Функциональное | А. Страница загружается ≤ 1.5 сек при 3G |
| 2. Нефункциональное (производительность) | Б. Система должна позволять отменять заказ до оплаты |
| 3. Требование безопасности | В. Использование только PostgreSQL 14+ и Linux Ubuntu |
| 4. Ограничение | Г. Все запросы к API шифруются по TLS 1.3 |

38. Соотнесите тип связи в ER-диаграмме с описанием:

- | | |
|---------------------------|---|
| 1. 1:1 (один к одному) | А. Студент $\leftrightarrow$ Зачётная книжка |
| 2. 1:M (один ко многим) | Б. Автор $\leftrightarrow$ Книги |
| 3. M:M (многие ко многим) | В. Студенты $\leftrightarrow$ Преподаватели (на курсах) |

39. Соотнесите метод сбора требований с характеристикой:

- | | |
|------------------------|--|
| 1. Интервью | А. Массовый сбор количественных данных |
| 2. Анкетирование | Б. Выявление скрытых проблем в реальном процессе |
| 3. Наблюдение | В. Глубинный диалог с экспертом |
| 4. Анализ документации | Г. Изучение регламентов, форм, отчётов |

41. Соотнесите процесс управления требованиями с его сутью:

- | | |
|--------------------|---|
| 1. Версионирование | А. Проверка соответствия требованиям стандартам и целям |
| 2. Аудит | Б. Фиксация изменений и хранение истории правок |
| 3. Трассировка | В. Оценка влияния и связей с другими артефактами |
| 4. Приоритизация | Г. Распределение по важности и срочности |

№	Ответ
36	1-В, 2-Г, 3-А, 4-Б
37	1-Б, 2-А, 3-Г, 4-В
38	1-А, 2-Б, 3-В
39	1-В, 2-А, 3-Б, 4-Г
40	1-Б, 2-А, 3-В, 4-Г

Тип 5: Установите правильную последовательность

41. Этапы жизненного цикла ИС:

1. Проектирование архитектуры
2. Сопровождение и поддержка
3. Сбор и анализ требований
4. Тестирование и внедрение
5. Разработка и кодирование

42. Построение спецификации требований (SRS):

1. Согласование с заказчиком
2. Сбор требований
3. Верификация и рецензирование
4. Структурирование и документирование
5. Внесение в репозиторий требований

43. Жизненный цикл требования:

1. Реализация в коде
2. Тестирование и валидация
3. Выявление и документирование
4. Приоритизация и согласование
5. Вывод из эксплуатации или архивирование

44. Проверка требований на качество:

1. Проверка на непротиворечивость
2. Проверка на полноту
3. Проверка на тестируемость
4. Проверка на однозначность

45. Переход от требований к архитектуре:

1. Выделение ключевых сценариев
2. Определение компонентов и их связей
3. Формирование требований к интеграции
4. Выбор технологий и платформ
5. Создание архитектурной схемы

№	Правильный порядок
41	3 → 1 → 5 → 4 → 2
42	2 → 4 → 3 → 1 → 5
43	3 → 4 → 1 → 2 → 5
44	4 → 1 → 2 → 3 (любой логичный порядок допустим, но обычно: однозначность → непротиворечивость → полнота → тестируемость)
45	1 → 3 → 2 → 4 → 5

Тип 6: Завершение утверждения / Выбор термина

46. Событийные модели поведения системы описывают ...

- а) статическую структуру таблиц в БД
- б) как система реагирует на внешние или внутренние события
- в) порядок установки серверного ПО
- г) цветовую схему интерфейса

47. Интеграционные требования определяют ...

- а) шрифт и отступы в формах
- б) расписание отпусков команды
- в) правила обмена данными между модулями, внешними системами и API
- г) объём оперативной памяти разработчика

48. Сценарий использования (Use Case Scenario) обязательно включает ...

- а) только успешный путь (main flow)
- б) основной поток, альтернативные пути и обработку исключений
- в) исходный код функции
- г) IP-адрес сервера

49. Метод анализа потребностей стейкхолдеров «Карта эмпатии» помогает ...

- а) скомпилировать проект
- б) создать резервную копию БД в) настроить брандмауэр
- г) понять боли, ожидания и контекст пользователей до проектирования

50. Диаграмма активностей применяется для ...

- а) моделирования параллельных и последовательных шагов процесса
- б) хранения паролей в открытом виде
- в) визуализации зависимостей между библиотеками
- г) расчёта бюджета проекта

№	Правильный порядок
46	б
47	в
48	б
49	г

по МДК.03.02 Разработка кода информационных систем

Часть 1: Выберите ОДИН правильный ответ

1. Основной принцип модульного программирования заключается в ...

- а) написании всего кода в одном файле для простоты компиляции
- б) разделении программы на независимые части, решающие конкретные задачи
- в) использовании только глобальных переменных
- г) отказе от использования функций и процедур

2. Процедура отличается от функции тем, что ...

- а) процедура всегда возвращает значение, а функция – нет
- б) функция предназначена для вычислений и возврата результата, а процедура – для выполнения действий без обязательного возврата значения
- в) процедуры нельзя вызывать из других модулей
- г) функции работают только с числами

3. Принцип инкапсуляции в ООП означает ...

- а) объединение данных и методов работы с ними в одном классе с ограничением прямого доступа
- б) создание множества наследников от одного базового класса
- в) автоматическое удаление неиспользуемых переменных
- г) шифрование исходного кода перед компиляцией

4. Какой тип файлового потока используется для записи текстовых данных?

- а) BinaryStream
- б) NetworkStream
- в) TextWriter / StreamWriter
- г) MemoryStream

5. Регулярные выражения (RegEx) в программировании применяются для ...

- а) сжатия графических файлов
- б) поиска, валидации и замены текста по заданному шаблону
- в) управления многопоточностью
- г) настройки сетевого протокола TCP

6. Блок try...catch...finally служит для ...

- а) ускорения выполнения кода в 2 раза
- б) перехвата и безопасной обработки исключительных ситуаций
- в) автоматического резервного копирования файлов
- г) изменения прав доступа к операционной системе

7. Повторное использование кода (code reuse) достигается за счёт ...

- а) копирования одинаковых фрагментов во все файлы проекта
- б) отказа от комментирования кода
- в) использования только встроенных функций языка
- г) вынесения общей логики в библиотеки, модули и классы

8. Протокол TCP отличается от UDP тем, что ...

- а) TCP гарантирует доставку и порядок пакетов, а UDP – нет
- б) TCP работает только в локальной сети, а UDP – в интернете
- в) UDP поддерживает шифрование, а TCP – нет
- г) TCP не требует установки соединения

9. В REST-архитектуре метод HTTP PUT используется для ...

- а) получения данных
- б) создания нового ресурса
- в) полного обновления существующего ресурса
- г) удаления ресурса

10. HTTP-код 500 Internal Server Error означает ...

- а) ресурс не найден
- б) доступ запрещён

в) на сервере произошла непредвиденная ошибка

г) запрос успешно выполнен

11. Аутентификация в современных API чаще всего реализуется через ...

а) передачу логина и пароля в открытом виде в каждом запросе

б) JWT-токены или API-ключи в заголовках запросов

в) отправку SMS-кодов на каждый запрос

г) использование только IP-адреса клиента

12. Событийно-ориентированное программирование в GUI подразумевает, что ...

а) код выполняется строго сверху вниз без отклонений

б) программа реагирует на действия пользователя или системные события (клик, ввод, таймер)

в) интерфейс обновляется только при перезагрузке окна

г) все события обрабатываются на сервере

13. ORM (Object-Relational Mapping) позволяет ...

а) писать запросы на SQL вручную в каждой функции

б) работать с реляционной БД через объекты языка программирования без прямого SQL

в) автоматически создавать графический интерфейс

г) шифровать таблицы в базе данных

14. Механизм кеширования в клиент-серверных приложениях нужен для ...

а) увеличения нагрузки на сервер

б) ускорения доступа к часто запрашиваемым данным и снижения числа запросов к БД

в) замены резервного копирования

г) отключения сетевых запросов

15. Сессия при авторизации используется для ...

а) хранения временного состояния аутентифицированного пользователя на стороне сервера

б) ускорения компиляции серверного кода

в) архивирования старых логов

г) синхронизации времени между клиентом и сервером

16. Профилирование сетевого взаимодействия выполняется для ...

а) изменения MAC-адреса сетевой карты

б) отключения брандмауэра

в) автоматической генерации API-документации

г) выявления узких мест, задержек и избыточной передачи данных

17. Конфигурационные файлы (JSON, YAML, INI) применяются для ...

а) хранения бизнес-логики приложения

б) вынесения настроек, путей, ключей и параметров среды из кода

в) сжатия изображений

г) управления потоками исполнения

18. Шаблоны проектирования Factory и Singleton относятся к ...

а) структурным шаблонам для работы с БД

б) шаблонам безопасности данных

в) поведенческим шаблонам для маршрутизации запросов

г) порождающим шаблонам для управления созданием объектов

19. Прямая связь клиентского интерфейса с базой данных в современных системах ...

а) рекомендуется для максимальной скорости

б) не рекомендуется из-за рисков безопасности, сложности масштабирования и блокировок

в) является единственным способом работы с таблицами

г) автоматически шифрует все запросы

20. Локализация и интернационализация (i18n/l10n) позволяют ...

а) переводить интерфейс и данные приложения на разные языки и региональные стандарты

б) ускорять загрузку страниц в 10 раз

в) отключать сетевые соединения при работе офлайн

г) автоматически обновлять версию ОС

21. Какой уровень автономности по классификации SAE предполагает полное отсутствие необходимости вмешательства человека в управление?

- а) Уровень 2
- б) Уровень 3
- в) Уровень 4
- г) Уровень 5

22. Какой датчик использует лазерное излучение для построения 3D-карты окружающей среды?

- а) Радар
- б) Лидар
- в) Ультразвуковой датчик
- г) Камера

23. Что означает аббревиатура V2X?

- а) Vehicle-to-eXchange
- б) Vehicle-to-Everything
- в) Vehicle-to-External
- г) Vehicle-to-Experience

24. Какая система компенсирует недостатки GNSS в условиях городской застройки?

- а) INS (инерциальная навигационная система)
- б) DSRC
- в) CAN-шина
- г) OBD-II

№	Ответ	№	Ответ
1	б	13	б
2	б	14	б
3	а	15	а
4	в	16	г
5	б	17	б
6	б	18	г
7	г	19	б
8	а	20	б
9	в	21	а
10	в	22	б
11	б	23	а
12	б	24	г

Часть 2: Выберите НЕСКОЛЬКО правильных ответов

25. Какие преимущества даёт модульная структура проекта? (выберите 3)

- а) Упрощение тестирования отдельных частей
- б) Возможность параллельной разработки разными программистами
- в) Увеличение размера исполняемого файла в 5 раз
- г) Повторное использование кода в других проектах
- д) Автоматическое устранение всех багов

26. Какие форматы чаще всего используются для сериализации данных в API а) JSON

- б) XML
- в) MP4
- г) BMP
- д) Protocol Buffers / MessagePack

27. Какие компоненты характерны для архитектуры настольного приложения? (выберите 3)

- а) Графический интерфейс (окна, кнопки, таблицы)
- б) Слой бизнес-логики

в) Слой работы с данными/БД

г) Веб-сервер Nginx

д) Docker-контейнер оркестратор

28. При обработке сетевых запросов сервер должен ... (выберите 3)

а) Валидировать входящие данные

б) Возвращать понятные коды состояния и сообщения об ошибках

в) Игнорировать заголовки авторизации

г) Логировать запросы для анализа

д) Отправлять все данные в открытом виде

29. Какие инструменты используются для отладки и профилирования сети? (выберите 3)

а) Postman / Insomnia

б) Wireshark

в) Microsoft Word

г) Браузерные DevTools (Network tab)

д) Калькулятор Windows

30. Устойчивость к сбоям (fault tolerance) обеспечивается за счёт ... (выберите 3)

а) Повторных попыток (retry) при временных ошибках сети

б) Graceful shutdown (корректного завершения процессов)

в) Игнорирования всех исключений в коде

г) Резервирования данных и механизмов восстановления

д) Отключения логирования

31. Связывание UI с логикой (Data Binding / MVC / MVVM) решает задачи ... (выберите 2–3)

а) Разделения ответственности между слоями

б) Автоматического обновления интерфейса при изменении данных

в) Упрощения поддержки и тестирования кода

г) Замены базы данных на текстовые файлы

д) Шифрования трафика

32. Жизненный цикл клиент-серверного приложения включает ... (выберите 3)

а) Инициализацию соединения и аутентификацию

б) Обмен запросами/ответами

в) Обработку ошибок и закрытие соединения

г) Перекомпиляцию ядра ОС

д) Физическую замену серверов

33. При разработке CLI-интерфейса важно предусмотреть ... (выберите 3)

а) Чёткий синтаксис команд и справку (–help)

б) Валидацию аргументов и понятные сообщения об ошибках

в) Коды выхода (exit codes) для автоматизации

г) Графические иконки на рабочем столе

д) Автоматическую отправку email

34. Какие меры повышают тестируемость кода? (выберите 3)

а) Чёткое разделение модулей и отсутствие глобального состояния

б) Использование инъекции зависимостей

в) Написание unit-тестов для критичных функций

г) Объединение всех классов в один файл

д) Отключение проверки типов

35. Какие уровни автономности по классификации SAE предполагают частичное участие человека в управлении?

а) Уровень 0

б) Уровень 1

в) Уровень 2

г) Уровень 3

д) Уровень 4

е) Уровень 5

36. Какие датчики используются в беспилотных транспортных системах для восприятия окружающей среды?

а) Лидар

б) Радар

в) Гироскоп

г) Ультразвуковой датчик

д) Акселерометр

е) Камера

37. Какие технологии связи применяются в БТС для обмена данными?

а) Wi-Fi

б) 4G/5G

в) V2X

г) DSRC

д) Bluetooth

е) NFC

№	Правильные варианты
25	а, б, г
26	а, б, д
27	а, б, в
28	а, б, г
29	а, б, г
30	а, б, г
31	а, б, в
32	а, б, в
33	а, б, в
34	а, б, в
35	а, б, г
36	а, в, д
37	а, д, е

Часть 3: Верно / Неверно

38. Абстракция позволяет скрыть сложную реализацию, оставив только необходимый интерфейс для использования.

○ Верно

○ Неверно

39. Протокол UDP гарантирует, что пакеты дойдут до получателя в том порядке, в котором были отправлены.

○ Верно

○ Неверно

40. Нормализация базы данных всегда замедляет чтение данных, поэтому в проектах её никогда не применяют.

○ Верно

○ Неверно

41. Сокеты обеспечивают двустороннюю передачу данных между клиентом и сервером в реальном времени.

○ Верно

○ Неверно

42. Документирование API (например, через Swagger/OpenAPI) необходимо только для внутренних проектов, но не для публичных.

- Верно
- Неверно

43. Уровень автономности 5 по классификации SAE предполагает, что водитель должен постоянно контролировать движение транспортного средства.

- Верно
- Неверно

44. Лидар использует лазерное излучение для построения 3D-карты окружающей среды.

- Верно
- Неверно

45. Технология V2X позволяет транспортным средствам обмениваться данными только с другими ТС.

- Верно
- Неверно

№	Ответ
38	Верно
39	Неверно (<i>UDP не гарантирует доставку и порядок; это делает TCP</i>)
40	Неверно (<i>Нормализация устраняет избыточность и аномалии; денормализация применяется осознанно для ускорения чтения</i>)
41	Верно
42	Неверно (<i>Документация критична именно для публичных и кросс-командных API, чтобы разработчики понимали, как с ним работать</i>)
43	Неверно. (<i>Уровень 5 предполагает полное отсутствие необходимости вмешательства человека.</i>)
44	Верно.
45	Неверно. (<i>V2X включает взаимодействие с инфраструктурой, пешеходами и т.д. (V2V, V2I, V2P и др.).</i>)

Часть 4: Установите соответствие

46. Соотнесите HTTP-метод с его назначением в REST API:

- | | |
|-----------|---------------------------------|
| 1. GET | А. Частичное обновление ресурса |
| 2. POST | Б. Получение данных |
| 3. PATCH | В. Удаление ресурса |
| 4. DELETE | Г. Создание нового ресурса |

47. Соотнесите HTTP-код с его значением:

- | | |
|--------|---|
| 1. 200 | А. Неавторизован (требуется аутентификация) |
| 2. 401 | Б. Сервис временно недоступен |
| 3. 404 | В. Успешный ответ |
| 4. 503 | Г. Ресурс не найден |

48. Соотнесите компонент приложения с его функцией:

- | | |
|----------------------|--|
| 1. Файл конфигурации | А. Промежуточная обработка запросов (логирование, авторизация) |
| 2. ORM-модель | Б. Хранение настроек вне исходного кода |
| 3. Middleware | В. Быстрый доступ к частым данным без запроса к БД |
| 4. Кэш | Г. Представление таблицы БД в виде объекта кода |

49. Соотнесите тип ошибки с примером её обработки:

- | | |
|----------------------------|---|
| 1. Ошибка ввода-вывода | А. Возврат 401 Unauthorized, предложение повторить вход |
| 2. Ошибка авторизации | Б. Проверка формата email/телефона до отправки запроса |
| 3. Ошибка валидации данных | В. try...catch при чтении несуществующего файла |
| 4. Таймаут сети | Г. retry с экспоненциальной задержкой или возврат 504 |

50. Соотнесите паттерн с задачей:

- | | |
|--------------|---|
| 1. Singleton | А. Обеспечение единственного экземпляра класса (напр., соединение с |
|--------------|---|

- | | | |
|----|------------|--|
| 2. | Factory | БД) |
| 3. | Observer | Б. Абстракция доступа к данным, замена SQL-запросов методами |
| 4. | Repository | В. Создание объектов без указания конкретного класса |
| | | Г. Уведомление подписчиков об изменении состояния |

51. Типы карт и их применение в БТС

Тип карты	Применение в архитектуре БТС
1. 2D-векторная карта (Open StreetMap)	А) Точное позиционирование с точностью до 10 см, содержит данные о разметке, знаках, светофорах, геометрии дорог
2. 3D-карта окружения	Б) Навигация по улицам, поиск точек интереса (POI), построение общего маршрута
3. HD-карта (High-Definition)	В) Визуальная локализация по признакам местности (здания, деревья, столбы)

№	Ответ
46	1-Б, 2-Г, 3-А, 4-В
47	1-В, 2-А, 3-Г, 4-Б
48	1-А, 2-В, 3-Г, 4-Б
49	1-В, 2-А, 3-Б, 4-Г
50	1-Б, 2-Г, 3-А, 4-В
51	1-В, 2-Б, 3-А

Часть 5: Установите правильную последовательность

52. Этапы обработки HTTP-запроса сервером:

1. Выполнение бизнес-логики
2. Приём и парсинг запроса
3. Валидация и аутентификация
4. Формирование и отправка ответа
5. Обращение к БД / внешним API

53. Жизненный цикл клиент-серверного соединения:

1. Установка соединения (TCP handshake)
2. Передача данных
3. Закрытие соединения
4. Аутентификация клиента
5. Обработка ошибок и восстановление

54. Обработка события нажатия кнопки в GUI:

1. Пользователь кликает мышью
2. Система генерирует событие Click
3. Обработчик (callback) принимает событие
4. Выполняется бизнес-логика
5. Обновляется состояние интерфейса

55. Этапы подключения к базе данных через ORM:

1. Настройка строки подключения в конфиге
2. Инициализация контекста/подключения
3. Формирование запроса через методы модели
4. ORM транслирует запрос в SQL и выполняет его
5. Получение и маппинг результатов в объекты

56. Шаги безопасной обработки исключения:

1. Попытка выполнения рискованной операции (try)
2. Перехват конкретного типа ошибки (catch)
3. Логирование деталей для разработчиков
4. Возврат безопасного fallback или понятного сообщения клиенту
5. Очистка ресурсов (finally)

57. Расположите этапы обработки данных сенсорами в правильной последовательности:

1. Классификация и идентификация объектов (пешеходы, машины, знаки)
2. Слияние данных с разных датчиков (sensor fusion)
3. Фильтрация и предварительная обработка сырых данных
4. Сбор сырых данных с датчиков (лидар, радар, камеры)
5. Передача обработанной информации в модуль планирования

58. Расположите этапы в порядке их выполнения:

1. Корректировка позиции на основе данных INS и одометрии
2. Сопоставление данных с HD-картой для уточнения позиции
3. Получение данных GNSS (GPS/ГЛОНАСС)
4. Построение локальной карты окружения с помощью SLAM (при отсутствии HD-карт)
5. Определение точного местоположения с сантиметровой точностью

№	Правильный порядок
52	2 → 3 → 5 → 1 → 4
53	1 → 4 → 2 → 5 → 3
54	1 → 2 → 3 → 4 → 5
55	1 → 2 → 3 → 4 → 5
56	1 → 2 → 4 → 5 → 3 (логирование может быть в catch, порядок 1-2-3-4-5 допустим, но ключевое: перехват → лог → ответ клиенту → очистка)
57	5 → 3 → 2 → 1 → 4
58	2 → 3 → 5 → 1 → 4

Часть 6: Завершение утверждения / Терминология

59. Архитектурный стиль REST основан на использовании стандартных HTTP-методов, stateless-соединений и работе с ...

- а) ресурсами, доступными через URI
- б) двоичными потоками
- в) закрытыми сокетами
- г) проприетарными протоколами

60. async/await в программировании применяется для ...

- а) блокировки всех потоков до завершения операции
- б) выполнения долгих операций (сеть, диск) без блокировки основного потока UI
- в) ускорения математических вычислений на CPU
- г) шифрования данных в памяти

61. Инкапсуляция в коде реализуется через модификаторы доступа (private, protected, public) и позволяет ...

- а) открыть все поля для прямого изменения извне
- б) автоматически генерировать документацию

- в) запретить создание объектов класса
- г) скрыть внутреннее состояние объекта и предоставить контролируемый доступ через методы

62. Токен авторизации в API передаётся в заголовке ...

- а) Content-Type
- б) Authorization: Bearer <token>
- в) User-Agent
- г) Accept-Language

63. Сокет (socket) в сетевом программировании представляет собой ...

- а) графический элемент интерфейса
- б) файл конфигурации сервера
- в) конечную точку двунаправленного сетевого соединения, идентифицируемую IP и портом
- г) зашифрованный архив данных

64. Уровень автономности 5 по классификации SAE означает:

- а) частичную автоматизацию с постоянным контролем водителя;
- б) автоматизацию отдельных функций (круиз-контроль);
- в) полную автоматизацию без участия человека в любых условиях;
- г) возможность автоматического торможения при угрозе столкновения.

65. HD-карты для БТС — это:

- а) карты низкого разрешения для общего ориентирования;
- б) высокодетализированные карты с точностью до сантиметров, содержащие данные о разметке, знаках и геометрии дорог;
- в) карты, обновляемые только раз в год;
- г) карты, предназначенные исключительно для пешеходов.

№	Правильный порядок
59	а
60	б
61	г
62	б
63	в
64	а
65	г

по МДК.03.03 Сопровождение информационных систем

Часть 1: Выберите ОДИН правильный ответ

1. Основная цель сопровождения информационной системы – это ...

- а) написание нового кода для расширения функционала
- б) поддержание работоспособности, актуальности и безопасности системы в течение всего жизненного цикла
- в) однократная установка ПО на сервер
- г) обучение пользователей перед запуском проекта

2. На каком этапе ввода системы в эксплуатацию проводится финальное тестирование и подготовка пользователей?

- а) Опытная эксплуатация
- б) Предпроектное обследование
- в) Промышленная эксплуатация
- г) Вывод из эксплуатации

3. Мониторинг работоспособности компонентов ИС позволяет ...

- а) автоматически писать код для исправления багов
- б) своевременно выявлять отклонения в работе и предотвращать сбои до их влияния на пользователей

в) заменять аппаратное обеспечение без остановки системы

г) шифровать все данные в реальном времени

4. Задача 1-й линии поддержки (Help Desk) – это ...

а) исправление критических ошибок в исходном коде

б) проектирование архитектуры новой системы

в) первичный приём обращений, регистрация инцидентов и решение типовых проблем

г) проведение нагрузочного тестирования

5. Документация по сопровождению должна обновляться ...

а) только при выводе системы из эксплуатации

б) только по запросу аудитора

в) раз в 5 лет по графику

г) при каждом изменении конфигурации, процедур или состава компонентов

6. Инцидент в процессе сопровождения – это ...

а) плановое обновление версии ПО

б) незапланированное прерывание или ухудшение качества услуги ИС

в) запрос на создание новой учётной записи

г) документ с описанием архитектуры

7. Модель ITIL в управлении ИТ-услугами фокусируется на ...

а) стандартизации процессов предоставления услуг и непрерывном улучшении

б) написании кода на конкретных языках программирования

в) замене всех серверов на облачные решения

г) автоматическом удалении старых логов

8. «Окно обслуживания» при планировании обновлений – это ...

а) период, когда система работает в тестовом режиме

б) заранее согласованное время для проведения работ с минимальным влиянием на пользователей

в) время, отведённое на обучение новых сотрудников

г) интервал для резервного копирования без остановки БД

9. Стратегия полного резервного копирования (full backup) подразумевает ...

а) копирование только изменённых данных с момента последнего бэкапа

б) сохранение только конфигурационных файлов

в) копирование всей базы данных или системы целиком на момент создания копии

г) архивирование логов за последний час

10. План восстановления после катастроф (DRP) отличается от обычного бэкапа тем, что ...

а) описывает пошаговые действия по восстановлению всей инфраструктуры в аварийной ситуации

б) содержит только список паролей администраторов

в) используется только для тестирования производительности

г) заменяет необходимость в мониторинге

11. Журналирование (логирование) в ИС необходимо для ...

а) ускорения выполнения запросов к базе данных

б) анализа инцидентов, аудита безопасности и отслеживания действий пользователей

в) автоматического создания резервных копий

г) шифрования сетевого трафика

12. Соглашение об уровне обслуживания (SLA) определяет ...

а) зарплату сотрудников службы поддержки

б) список запрещённых к использованию программ

в) целевые метрики доступности, времени реакции и устранения инцидентов

г) график отпусков технических специалистов

13. Учёт программных и аппаратных ресурсов ведётся для ...

а) увеличения нагрузки на серверы

б) контроля лицензий, планирования закупок и диагностики проблем

в) автоматического удаления неиспользуемых файлов

г) замены операционной системы на всех узлах

14. Управление конфигурациями (CMDB) позволяет ...

- а) хранить только исходный код приложений
- б) шифровать конфигурационные файлы
- в) автоматически тестировать пользовательский интерфейс
- г) отслеживать взаимосвязи компонентов, их версии и историю изменений

15. Технический паспорт системы должен содержать ...

- а) только контактные данные разработчиков
- б) архитектуру, зависимости, процедуры восстановления и регламенты обслуживания
- в) список личных предпочтений администраторов
- г) исходный код всех модулей

16. Контроль лицензий ПО важен для ...

- а) увеличения скорости компиляции
- б) соблюдения законодательства и избежания юридических рисков
- в) автоматического обновления всех программ
- г) отключения антивирусной защиты

17. Регламент обработки инцидентов описывает ...

- а) этапы регистрации, классификации, эскалации и закрытия инцидентов
- б) порядок написания нового функционала
- в) процедуру увольнения сотрудников поддержки
- г) правила выбора цветовой схемы интерфейса

18. Миграция данных при обновлении системы требует ...

- а) полного удаления старых данных перед переносом
- б) предварительного бэкапа, валидации целостности и плана отката на случай ошибок
- в) отключения всех пользователей на неопределённый срок
- г) изменения языка программирования серверной части

19. Подготовка системы к масштабированию включает ...

- а) удаление всех индексов в базе данных для ускорения записи
- б) проектирование горизонтального/вертикального масштабирования и балансировки нагрузки
- в) отказ от резервного копирования для экономии места
- г) использование только одного сервера для всех компонентов

20. Обучение пользователей в процессе сопровождения нужно для ...

- а) снижения нагрузки на службу поддержки и повышения эффективности использования системы
- б) увеличения количества баг-репортов
- в) замены документации устными инструкциями
- г) отключения функции самопомощи в интерфейсе

№	Ответ	№	Ответ
1	б	11	б
2	а	12	в
3	б	13	б
4	в	14	г
5	г	15	б
6	б	16	б
7	а	17	а
8	б	18	б
9	в	19	б
10	а	20	а

Часть 2: Выберите НЕСКОЛЬКО правильных ответов

21. Какие виды тестирования применяются в процессе сопровождения ИС? (выберите 3)

- а) Модульное (проверка отдельных функций)
- б) Интеграционное (взаимодействие компонентов)

- в) Системное (проверка всей системы целиком)
- г) Маркетинговое (анализ конкурентов)
- д) Финансовое (расчёт бюджета)

22. Автоматизация тестирования целесообразна для ... (выберите 3)

- а) Регрессионного тестирования после каждого обновления
- б) Проверки критичных бизнес-сценариев (smoke-тесты)
- в) API-тестирования и валидации ответов
- г) Тестирования удобства интерфейса (UX) глазами пользователя
- д) Оценки креативности дизайна

23. Тест-план должен содержать ... (выберите 3)

- а) Цели и область тестирования
- б) Ресурсы, сроки и ответственных
- в) Критерии начала и завершения тестирования
- г) Исходный код всех модулей
- д) Личные данные тестирующих

24. Отчёт о тестировании включает ... (выберите 3)

- а) Количество выполненных/проваленных тест-кейсов
- б) Классификацию найденных дефектов по критичности
- в) Рекомендации по выпуску или доработке
- г) Пароли от тестовых сред
- д) График отпусков команды

25. CI/CD (Continuous Integration/Delivery) в поддержке ИС позволяет ... (выберите 3)

- а) Автоматически запускать тесты при каждом коммите
- б) Быстро и безопасно доставлять обновления в продакшен
- в) Снижать риск человеческой ошибки при развёртывании
- г) Полностью отказаться от ручного тестирования
- д) Увеличивать время простоя при обновлениях

26. Тестирование безопасности (Security Testing) направлено на выявление ... (выберите 3)

- а) Уязвимостей к инъекциям (SQL, XSS)
- б) Небезопасных конфигураций и прав доступа
- в) Утечек чувствительных данных в логах или ответах
- г) Неудобного расположения кнопок в интерфейсе
- д) Медленной загрузки изображений

27. Нагрузочное тестирование отличается от стресс-тестирования тем, что ... (выберите 2)

- а) Нагрузочное проверяет работу при ожидаемом пике пользователей
- б) Стресс-тестирование нагружает систему сверх предела для определения точки отказа
- в) Нагрузочное проводится только на тестовых данных
- г) Стресс-тестирование не требует мониторинга метрик
- д) Оба вида идентичны и проводятся одновременно

28. Инструменты управления тестированием (Jira, TestRail, Zephyr) помогают ... (выберите 3)

- а) Вести реестр тест-кейсов и их статусов
- б) Связывать дефекты с требованиями и задачами
- в) Формировать отчёты и метрики качества
- г) Автоматически писать код для исправления багов
- д) Заменять живых тестирующих

29. Обратная совместимость при обновлениях означает, что ... (выберите 2)

- а) Новая версия системы корректно работает со старыми клиентами/интерфейсами
- б) Старые данные и форматы поддерживаются после миграции
- в) Пользователи обязаны обновлять свои устройства одновременно с сервером
- г) Все предыдущие версии удаляются сразу после релиза
- д) Тестирование проводится только на новой функциональности

30. Управление изменениями (Change Management) включает ... (выберите 3)

- а) Формальную заявку на изменение с обоснованием и оценкой рисков
- б) Согласование в CAB (Change Advisory Board) для критичных изменений
- в) План внедрения, тестирования и отката
- г) Автоматическое применение всех изменений без проверки
- д) Игнорирование влияния на пользователей

№	Правильные варианты
21	а, б, в
22	а, б, в
23	а, б, в
24	а, б, в
25	а, б, в
26	а, б, в
27	а, б, в
28	а, б
29	а, б, в
30	а, б

Часть 3: Верно / Неверно

31. Приёмочное тестирование (UAT) проводится заказчиком или конечными пользователями перед вводом системы в эксплуатацию.

- ☐ Верно
- ☐ Неверно

32. Ручное регрессионное тестирование всегда эффективнее автоматизированного, потому что человек находит больше ошибок.

- ☐ Верно
- ☐ Неверно *(Автоматизация быстрее, повторяемее и надёжнее для регресса; ручное тестирование лучше для исследовательских и UX-проверок)*

33. Логирование на уровне DEBUG рекомендуется для рабочего (продакшен) сервера, так как помогает быстрее находить ошибки.

- ☐ Верно
- ☐ Неверно *(Уровень DEBUG создаёт огромный объём логов, снижает производительность и может содержать чувствительные данные; в продакшене используют INFO/WARNING/ERROR)*

34. DevOps-практики в сопровождении объединяют разработку и эксплуатацию для ускорения доставки обновлений и повышения стабильности.

- ☐ Верно
- ☐ Неверно

35. SLA может включать штрафные санкции для поставщика услуг при систематическом нарушении согласованных метрик.

- ☐ Верно
- ☐ Неверно

№	Ответ
31	Верно
32	Неверно <i>(Автоматизация быстрее, повторяемее и надёжнее для регресса; ручное тестирование лучше для исследовательских и UX-проверок)</i>
33	Неверно <i>(Уровень DEBUG создаёт огромный объём логов, снижает производительность и может содержать чувствительные данные; в продакшене используют INFO/WARNING/ERROR)</i>
34	Верно
35	Верно

Часть 4: Установите соответствие

36. Соотнесите уровень поддержки с его задачами:

- | | |
|-----------------------------------|---|
| 1. 1-я линия (Help Desk) | А. Глубокий анализ, исправление кода, работа с вендорами |
| 2. 2-я линия (Technical Support) | Б. Первичный приём, регистрация, решение типовых вопросов |
| 3. 3-я линия (Expert/Development) | В. Диагностика, воспроизведение, решение сложных инцидентов |

37. Соотнесите тип бэкапа с характеристикой:

- | | |
|---------------------|---|
| 1. Полный (Full) | А. Копирует изменения с момента последнего полного бэкапа |
| 2. Инкрементальный | Б. Копирует всю систему/БД целиком |
| 3. Дифференциальный | В. Копирует изменения с момента последнего бэкапа любого типа |

38. Соотнесите вид тестирования с целью:

- | | |
|-------------------|--|
| 1. Модульное | А. Проверка взаимодействия компонентов между собой |
| 2. Интеграционное | Б. Проверка соответствия системы требованиям заказчика |
| 3. Системное | В. Тестирование изолированной функции или класса |
| 4. Приёмочное | Г. Проверка работы всей системы в условиях, близких к продакшену |

39. Соотнесите метрику SLA с её значением:

- | | |
|-------------------------------------|---|
| 1. Доступность 99.9% | А. Максимально допустимое время от регистрации инцидента до начала работы |
| 2. Время реакции ≤ 15 мин | Б. Допустимый объём данных, которые можно потерять при сбое |
| 3. Время устранения ≤ 4 часа | В. Допустимое время простоя в месяц (~43 минуты) |
| 4. Потеря данных (RPO) ≤ 5 мин | Г. Максимальное время на восстановление работоспособности |

40. Соотнесите процесс с его артефактом:

- | | |
|------------------------------|---|
| 1. Управление инцидентами | А. Заявка на изменение (RFC), протокол CAB |
| 2. Управление изменениями | Б. Тикет инцидента, журнал эскалаций |
| 3. Управление конфигурациями | В. CMDB, реестр конфигурационных единиц |
| 4. Тестирование | Г. Тест-план, отчёт о дефектах, матрица трассировки |

№	Ответ
36	1-Б, 2-А, 3-В, 4-Г
37	1-В, 2-А, 3-Г, 4-Б
38	1-В, 2-А, 3-Г, 4-Б
39	1-Б, 2-В, 3-А
40	1-Б, 2-В, 3-А

Часть 5: Установите правильную последовательность

41. Этапы обработки инцидента:

1. Регистрация и классификация
2. Диагностика и решение
3. Эскалация (при необходимости)
4. Закрытие и документирование
5. Уведомление пользователя

42. Жизненный цикл изменения (Change Management):

1. подача заявки на изменение (RFC)
2. Оценка рисков и планирование
3. Согласование в CAB (для критичных)

4. Внедрение в тестовой/продакшен-среде
5. Пост-реализационный обзор и закрытие

43. Процесс восстановления после сбоя:

1. Обнаружение инцидента и оповещение
2. Оценка ущерба и выбор точки восстановления
3. Применение бэкапов и логов в правильном порядке
4. Проверка целостности данных и функциональности
5. Возврат системы в продакшен и мониторинг

44. Подготовка к выпуску обновления:

1. Разработка и модульное тестирование
2. Интеграция и регрессионное тестирование
3. Развёртывание в стейджинг-среде
4. Приёмочное тестирование (UAT)
5. Развёртывание в продакшен с планом отката

45. Настройка мониторинга компонента ИС:

1. Определение ключевых метрик (ЦП, память, ошибки)
2. Выбор инструмента (Zabbix, Prometheus и др.)
3. Настройка сбора, хранения и визуализации метрик
4. Настройка алертов при превышении порогов
5. Регулярный анализ и оптимизация правил

№	Правильный порядок
41	1 → 3 → 2 → 5 → 4
42	1 → 2 → 3 → 4 → 5
43	1 → 2 → 3 → 4 → 5
44	1 → 2 → 3 → 4 → 5
45	1 → 2 → 3 → 4 → 5

Часть 6: Завершение утверждения / Терминология

46. Матрица трассировки требований в сопровождении связывает ...

- а) зарплату сотрудников с количеством закрытых тикетов
- б) требования с тест-кейсами, модулями и инцидентами для контроля покрытия
- в) цвета интерфейса с предпочтениями пользователей
- г) версии ОС с моделями ноутбуков

47. Стратегия «синего-зелёного» развёртывания (blue-green deployment) позволяет ...

- а) увеличить время простоя при обновлении
- б) переключать трафик между двумя идентичными средами для минимизации риска и быстрого отката
- в) использовать только один сервер для всех задач
- г) автоматически удалять старые версии ПО

48. Конфигурационная единица (CI) в CMDB – это ...

- а) любой компонент ИС (сервер, ПО, документ, лицензия), подлежащий учёту и управлению
- б) код ошибки в логах приложения
- в) идентификатор пользователя в системе
- г) версия браузера клиента

49. «Graceful shutdown» (корректное завершение) процесса означает ...

- а) аварийное завершение с потерей данных в памяти
- б) завершение с сохранением состояния, освобождением ресурсов и записью в логи
- в) принудительную остановку через диспетчер задач
- г) перезагрузку сервера без предупреждения

50. Ретроспектива инцидента (Post-Incident Review) проводится для ...

- а) поиска виноватого и наложения штрафа
- б) автоматического закрытия всех тикетов

- в) анализа корневых причин, извлечения уроков и улучшения процессов
г) увеличения количества алертов в мониторинге

№	Правильный порядок
46	б
47	б
48	а
49	б
50	в

Критерии оценки:

90-100 баллов «отлично» заслуживает студент, показавший всестороннее систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной программой; как правило, оценка «отлично» выставляется студентам, усвоившим взаимосвязь основных понятий междисциплинарного курса и их значение для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала.

80-90 баллов «хорошо» заслуживает студент, обнаруживший полное знание учебно-программного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе; как правило, оценка «хорошо» выставляется студентам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.

60-80 баллов «удовлетворительно» заслуживает студент, обнаруживший знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой; как правило, оценка «удовлетворительно» выставляется студентам, допустившим погрешности в ответе на зачете, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.

Менее 60 баллов «неудовлетворительно» выставляется студенту, обнаружившему проблемы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий; как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжать обучение или приступить к профессиональной деятельности по окончании учебного заведения без дополнительных занятий по соответствующему междисциплинарному курсу.

Кейс-задачи

по МДК.03.01 Проектирование информационных систем

Кейс 1. Выявление требований и анализ стейкхолдеров

Контекст: Колледж запускает систему «Учёт производственной практики». В процесс вовлечены: студенты, руководители практики от колледжа, наставники от компаний-партнёров, деканат, IT-отдел. Сейчас отчёты сдаются на бумаге, теряются, обрабатываются вручную.

Задание:

1. Составьте карту заинтересованных сторон и опишите их ключевые потребности.
2. Выберите по одному методу сбора требований для каждой группы (интервью, анкетирование, наблюдение, анализ документов). Обоснуйте выбор.
3. Сформулируйте 3 функциональных и 2 нефункциональных требования.

Кейс 2. Моделирование бизнес-процессов и визуализация

Контекст: Текущий процесс согласования отчёта занимает 5–7 дней из-за ручного переноса данных и потери документов.

Задание:

1. Нарисуйте диаграмму активностей текущего процесса.
2. Выявите узкие места и предложите оптимизированный процесс.
3. Постройте диаграмму активностей «To-Be» и выделите 2–3 Use Case.

Кейс 3. Классификация требований и разрешение конфликтов

Контекст: IT-отдел требует строгой двухфакторной аутентификации, студенты просят вход по одному клику через VK ID, деканат хочет видеть все действия в логах.

Задание:

1. Распределите требования по типам (функциональные, нефункциональные, ограничения).
2. Примените метод MoSCoW для приоритизации.
3. Предложите компромиссное решение и зафиксируйте его в шаблоне требования.

Кейс 4. Моделирование данных и нормализация

Контекст: В системе нужно хранить: студентов, группы, компании, наставников, отчёты, оценки, даты практики. Сейчас данные дублируются, возникают ошибки при обновлении.

Задание:

1. Выделите сущности и их атрибуты.
2. Определите связи (1:1, 1:M, M:M) и запишите правила целостности.
3. Постройте ER-диаграмму и доведите модель до 3НФ.

Кейс 5. Прототипирование и модели взаимодействия

Контекст: Пользователи жалуются: «Много кликов, непонятно, куда нажимать, нет подтверждения сохранения».

Задание:

1. Создайте low-fidelity прототип экрана «Подача отчёта».
2. Опишите модель взаимодействия: что происходит от нажатия «Отправить» до появления статуса «Отправлено».
3. Свяжите каждый элемент интерфейса с функциональным требованием.

Кейс 6. Спецификация требований и трассировка

Контекст: Требования разбросаны по письмам, чатам и заметкам. Команда путается, что уже сделано, а что нет.

Задание:

1. Структурируйте требования по шаблону SRS (введение, общие положения, функциональные, нефункциональные, ограничения).
2. Проверьте требования на однозначность, полноту, тестируемость.
3. Постройте матрицу: Требование → Модуль → Тест-кейс → Статус.

Кейс 7. Требования безопасности и анализ рисков

Контекст: Система будет хранить паспортные данные, оценки, контакты компаний. По закону требуется защита ПДн.

Задание:

1. Сформулируйте 4 требования безопасности (аутентификация, авторизация, шифрование, аудит).
2. Проведите анализ рисков: вероятность × влияние.
3. Предложите меры снижения для 2 критических рисков.

Кейс 8. Интеграция и миграция данных

Контекст: Система должна получать данные о студентах из существующей ERP-системы колледжа и импортировать 3 года архивных отчётов из Excel.

Задание:

1. Сформулируйте интеграционные требования (формат, протокол, частота синхронизации).
2. Составьте план миграции: очистка, маппинг, тестовый прогон, валидация.
3. Оцените объём данных и требования к хранилищу на 3 года.

Кейс 9. Управление изменениями и версионирование

Контекст: В ходе разработки деканат просит добавить «автоматический расчёт стипендии по практике». Разработчики возражают: это усложнит архитектуру и сорвёт сроки.

Задание:

1. Опишите процесс внесения изменения: заявка → оценка → согласование → внедрение.
2. Покажите, как изменится версия SRS и матрица трассировки.
3. Подготовьте краткий отчёт для комитета по изменениям (CAB).

Кейс 10. Переход от требований к архитектуре и согласование команды

Контекст: Требования утверждены. Аналитик предлагает микросервисы, разработчики настаивают на монолите из-за сроков и компетенций.

Задание:

1. Сопоставьте требования с будущими модулями/слоями системы.
2. Постройте диаграмму последовательности для сценария «Подача отчёта» и диаграмму состояний для сущности «Отчёт».
3. Подготовьте аргументы для выбора архитектуры (монолит vs модульный монолит vs микросервисы).

по МДК.03.02 Разработка кода информационных систем

Кейс 1. Модульная архитектура утилиты обработки данных

Контекст: Требуется разработать утилиту для парсинга серверных логов, фильтрации по шаблону и экспорта отчётов. Код быстро разрастается, если писать всё в одном файле.

Задание:

1. Спроектируйте структуру проекта: выделите модули (parser, filter, exporter, config, main).
2. Реализуйте базовые классы с инкапсуляцией: LogEntry, RuleFilter, CSVExporter.
3. Обеспечьте корректную инициализацию ресурсов (открытие файлов, пулы) и их освобождение при завершении.

Кейс 2. Консольная утилита с конфигурацией и обработкой ошибок

Контекст: CLI-инструмент для массовой переименовки файлов по регулярным выражениям должен читать настройки из внешнего файла и корректно реагировать на сбои.

Задание:

1. Напишите парсер CLI-аргументов и загрузку настроек из YAML/JSON.
2. Реализуйте применение regex к именам файлов с валидацией входных данных.
3. Оборните файловые операции в try...catch, логируйте ошибки, не прерывая обработку остальных файлов.

Кейс 3. Асинхронный сетевой модуль проверки доступности

Контекст: Нужно проверить доступность 500+ хостов и портов. Синхронный код работает >10 минут, требуется оптимизация.

Задание:

1. Реализуйте проверку портов через TCP connect и UDP send/recv.
2. Перепишите код с использованием asyncio или пула потоков.
3. Добавьте таймауты, повторные попытки (retry) и graceful shutdown по Ctrl+C.

Кейс 4. REST API для системы управления заказами

Контекст: Требуется разработать бэкенд для создания, чтения, обновления и удаления заказов с документацией и безопасностью.

Задание:

1. Определите эндпоинты: /api/orders (GET/POST), /api/orders/{id} (GET/PUT/DELETE).
2. Реализуйте валидацию входящих данных, возврат корректных HTTP-статусов и структурированных ошибок.
3. Добавьте JWT-аутентификацию и документацию через OpenAPI/Swagger.

Кейс 5. Desktopное приложение с таблицами и графиками

Контекст: Нужно создать GUI для визуализации продаж: таблица с данными, фильтры, динамический график, обновление без перезагрузки окна.

Задание:

1. Выберите фреймворк (Qt, WPF, Tkinter, Electron). Реализуйте форму с таблицей и графиком.
2. Свяжите данные с UI через паттерн MVVM/MVC или наблюдатель.
3. Обработайте клики, валидируйте ввод фильтров, обновляйте график асинхронно.

Кейс 6. Работа с БД: ORM vs прямые запросы

Контекст: Приложение хранит пользователей и заказы. Требуется реализовать CRUD через ORM, а сложные отчёты – через SQL.

Задание:

1. Настройте подключение к PostgreSQL/SQLite. Создайте модели User, Order.
2. Реализуйте CRUD через ORM. Для отчёта «Сумма заказов по месяцам» напишите raw SELECT.
3. Свяжите результаты запросов с таблицей в GUI/CLI.

Кейс 7. Система аутентификации и управления сессиями

Контекст: Требуется реализовать регистрацию, вход, хранение сессий и разграничение доступа для ролей user и admin.

Задание:

1. Создайте AuthManager (Singleton) для управления токенами/сессиями.
2. Реализуйте хэширование паролей, генерацию JWT, проверку прав.
3. Обеспечьте отзыв токенов, автоматическое обновление сессий и безопасное завершение.

Кейс 8. Сериализация, кеширование и обмен файлами

Контекст: Клиентское приложение должно получать от сервера большие объёмы данных в JSON/XML, кэшировать их и загружать файлы по запросу.

Задание:

1. Напишите сериализатор/десериализатор для JSON и XML. Обработайте несовместимые версии схем.
2. Реализуйте in-memory или дисковый кэш с TTL и инвалидацией при обновлении.
3. Добавьте эндпоинт /api/files/download с поддержкой chunked-передачи.

Кейс 9. Тестирование, отладка и профилирование сетевого модуля

Контекст: Модуль отправки уведомлений работает нестабильно: иногда «виснет», потребляет много памяти, падает при сетевых сбоях.

Задание:

1. Напишите unit- и integration-тесты с моками внешних API.
2. Профилируйте сеть и память (cProfile, Py-Spy, Chrome DevTools, Wireshark).
3. Найдите узкие места, исправьте утечки сокетов/памяти, добавьте retry и circuit breaker.

Кейс 10. Локализация, модульные библиотеки и полный цикл

Контекст: Приложение нужно адаптировать под RU/EN/ES, упаковать как переиспользуемую библиотеку + standalone CLI, корректно работать с датами/часовыми поясами.

Задание:

1. Вынесите все строки в .json/.po файлы. Реализуйте подгрузку по локали.
2. Отформатируйте даты/время с учётом часовых поясов (UTC → локальное).
3. Упакуйте код в пакет (pip install, npm publish, или DLL/so), добавьте CLI-обёртку.

Кейс 11. Потеря сигнала GNSS в тоннеле

Контекст: Беспилотный автобус движется по маршруту в мегаполисе. При въезде в длинный тоннель сигнал GNSS (GPS/ГЛОНАСС) пропадает. Система начинает работать нестабильно: наблюдаются колебания траектории, замедление реакции на препятствия.

Задание:

1. Какие компоненты архитектуры БТС должны компенсировать потерю GNSS? Опишите принцип их работы.
2. Какие данные и алгоритмы помогут сохранить точное позиционирование в тоннеле?
3. Предложите алгоритм действий системы при потере GNSS-сигнала (3–4 шага).
4. Какие меры предосторожности должен предпринять БТС в такой ситуации?

Кейс 12. Плохие погодные условия

Контекст: БТС движется по шоссе в условиях сильного снегопада. Камеры практически не видят дорожную разметку и знаки, лидар зашумлён из-за снежинок, радар работает, но не даёт детализированной картины окружения.

Задание:

1. Какие сенсоры останутся работоспособными в этих условиях? Обоснуйте ответ.
2. Как система должна перераспределить приоритеты между данными с разных датчиков?
3. Какие изменения в стратегии движения (скорость, дистанция до впереди идущего ТС) необходимо внести? Почему?

по МДК.03.03 Сопровождение информационных систем

Кейс 1. Ввод системы в промышленную эксплуатацию

Контекст: Колледж внедряет новую систему «Электронный журнал». Тестирование пройдено, но перед запуском нужно подготовить инфраструктуру, обучить пользователей и обеспечить поддержку.

Задание:

1. Составьте чек-лист предзапусковых проверок (бэкапы, мониторинг, права доступа, документация).
2. Разработайте план обучения для трёх групп пользователей: преподаватели, студенты, администраторы.
3. Подготовьте регламент «окна обслуживания» для будущих обновлений.

Кейс 2. Обработка инцидента: пользователи не могут войти в систему

Контекст: В 9:00 утра поступило 15 обращений: «Не могу войти в личный кабинет». Система работает, но авторизация не проходит.

Задание:

1. Классифицируйте инцидент (критичность, влияние, категория).
2. Опишите действия 1-й, 2-й и 3-й линии поддержки.
3. Составьте шаблон ответа пользователю и внутренний отчёт об инциденте.

Кейс 3. Настройка мониторинга и алертинга

Контекст: Система периодически «подвисает» в часы пик, но пользователи не всегда сообщают об этом. Нужно внедрить проактивный мониторинг.

Задание:

1. Определите 5–7 ключевых метрик для мониторинга (ЦП, память, диск, время отклика, ошибки БД).
2. Настройте пороги срабатывания алертов и каналы уведомлений (Telegram, email).
3. Составьте инструкцию для дежурного администратора: «Что делать, если сработал алерт».

Кейс 4. Планирование и откат обновления

Контекст: Вышло обновление системы с исправлением уязвимости. Нужно установить его с минимальным риском для пользователей.

Задание:

1. Составьте план обновления: бэкап → тестирование на стейджинге → окно обслуживания → установка → проверка.
2. Напишите сценарий отката (rollback) на случай неудачи.
3. Подготовьте коммуникацию для пользователей: что, когда и как изменится.

Кейс 5. Стратегия резервного копирования и восстановление

Контекст: В системе хранятся оценки и персональные данные. По регламенту $RPO \leq 1$ час, $RTO \leq 4$ часа. Нужно настроить бэкапы и проверить восстановление.

Задание:

1. Выберите типы бэкапов (full, incremental, differential) и частоту для БД, файлов, конфигураций.
2. Напишите скрипт бэкапа с проверкой целостности и уведомлением.
3. Проведите учебное восстановление: удалите тестовые данные → восстановите из бэкапа → сверьте результат.

Кейс 6. Управление конфигурациями и технический паспорт

Контекст: В системе 15 серверов, 30+ компонентов, частые изменения. Теряется актуальность информации о конфигурациях.

Задание:

1. Определите перечень конфигурационных единиц (ЦИ): серверы, ПО, БД, сетевые правила, учётные записи.
2. Создайте шаблон технического паспорта системы (архитектура, зависимости, контакты, процедуры).
3. Предложите процесс обновления CMDB при каждом изменении.

Кейс 7. Тестирование перед выпуском обновления

Контекст: Разработчики передали новую версию модуля отчётности. Нужно убедиться, что обновление не сломает существующий функционал.

Задание:

1. Составьте тест-план: какие виды тестирования, в какой последовательности, кто отвечает.
2. Напишите 3–5 тест-кейсов для критичных сценариев (формирование отчёта, экспорт, права доступа).
3. Оформите отчёт о тестировании: что проверено, какие дефекты найдены, рекомендация к выпуску.

Кейс 8. Нагрузочное тестирование и подготовка к масштабированию

Контекст: Ожидается рост пользователей в 3 раза к началу учебного года. Нужно оценить, выдержит ли система нагрузку, и подготовить план масштабирования.

Задание:

1. Определите сценарии нагрузки: типичные действия пользователей, пиковое время, целевые метрики (RPS, latency).

2. Запустите нагрузочный тест (JMeter, k6, Locust), соберите метрики.
3. Предложите план масштабирования: вертикальное/горизонтальное, кеширование, оптимизация запросов.

Кейс 9. Миграция данных в новую версию системы

Контекст: В новой версии изменена структура таблицы «Оценки»: добавлены поля, изменены типы данных. Нужно перенести данные без потерь.

Задание:

1. Проанализируйте изменения схемы БД, составьте маппинг полей «старое → новое».
2. Напишите скрипт миграции с проверками: целостность, дубликаты, значение по умолчанию.
3. Проведите тестовую миграцию на копии БД, сверьте результаты, подготовьте план отката.

Кейс 10. Внедрение DevOps-практик в процесс сопровождения

Контекст: Команда тратит много времени на ручное развёртывание, тестирование и мониторинг. Руководство хочет внедрить CI/CD и автоматизацию.

Задание:

1. Опишите этапы пайплайна: сборка → модульные тесты → интеграционные тесты → развёртывание на staging → ручное подтверждение → продакшен.
2. Выберите инструменты: GitLab CI / GitHub Actions / Jenkins, Docker, Ansible, Prometheus.
3. Предложите метрики успеха: время от коммита до продакшена, % автоматических деплоев, время восстановления.

Критерии оценки:

отметка «5»: Задание выполнено в полном объёме с соблюдением необходимой последовательности. Студент работал полностью самостоятельно.

отметка «4»: Практическое задание выполнено студентом в полном объёме и самостоятельно. Допускается отклонение от необходимой последовательности выполнения, не влияющее на правильность конечного результата. Допускаются неточности и небрежность в оформлении результатов задания.

отметка «3»: Практическое задание выполнено и оформлено студентом с помощью преподавателя или хорошо подготовленных и уже выполнивших на «отлично» данную работу студентов. На выполнение задания затрачено много времени.

отметка «2»: Выставляется в том случае, когда студент оказался неподготовленным к выполнению задания. Полученные результаты не позволяют сделать правильных выводов и полностью расходятся с поставленной целью. Обнаружено плохое знание теоретического материала и отсутствие необходимых умений. Руководство и помощь со стороны преподавателя неэффективны из-за плохой подготовки студента.

Ситуационные задачи для учебной практики

по ПМ.03 «Проектирование и разработка информационных систем»

Задание 1. Сбор и структурирование требований для ТЗ

Контекст: Колледж запускает веб-портал «Портфолио студента». Текущий процесс сбора работ разрознён, нет единого формата.

Задание:

1. Выделите 3 группы стейкхолдеров и опишите их потребности.
2. Сформулируйте 5 функциональных и 3 нефункциональных требования.
3. Составьте оглавление ТЗ по ГОСТ 34.602-89 или IEEE 830.

Задание 2. Архитектура и выбор стека технологий

Контекст: На основе ТЗ необходимо определить технологический стек и спроектировать архитектуру системы.

Задание:

1. Выберите стек (Frontend, Backend, БД, хостинг) и аргументируйте выбор.
2. Нарисуйте схему архитектуры (клиент → API-шлюз/сервер → БД).
3. Опишите контракт взаимодействия: формат данных, протокол, аутентификация.

Задание 3. Адаптивная вёрстка и семантика

Контекст: Требуется реализовать главную страницу портала с учётом мобильных устройств и стандартов доступности.

Задание:

1. Сверстайте макет с использованием Flexbox/Grid.
2. Реализуйте 3 брейкпоинта (mobile ≤768px, tablet ≤1024px, desktop).
3. Добавьте семантические теги (<header>, <main>, <nav>, <section>) и базовые ARIA-атрибуты.

Задание 4. Интерактивность и управление состоянием

Контекст: Форма загрузки работы должна валидировать данные, показывать статус загрузки и обрабатывать ошибки без перезагрузки страницы.

Задание:

1. Напишите валидацию полей (тип файла, размер, обязательность).
2. Реализуйте асинхронную отправку с индикаторами: loading, success, error.
3. Организуйте состояние формы в едином объекте или используйте легковесный менеджер состояний.

Задание 5. Проектирование и реализация REST API

Контекст: Серверная часть должна предоставлять endpoints для управления портфолио, проектами и комментариями.

Задание:

1. Определите ресурсы и HTTP-методы: /api/portfolios, /api/projects, /api/comments.
2. Реализуйте роутинг, глобальный обработчик ошибок и middleware для логирования.
3. Стандартизируйте формат ответа: { status, data, message }.

Задание 6. Интеграция с БД и защита данных

Контекст: Данные пользователей и проектов должны храниться надёжно, с защитой от инъекций и валидацией на уровне сервера.

Задание:

1. Спроектируйте схему БД: users, portfolios, projects.
2. Реализуйте подключение, пул соединений, базовые CRUD-операции.
3. Добавьте валидацию входящих данных (Zod, Joi, Pydantic или аналоги).

Задание 7. Интеграция клиент-сервер и обработка сетевых ошибок

Контекст: Фронтенд должен стабильно взаимодействовать с бэкендом, корректно обрабатывая сбои сети и серверные ошибки.

Задание:

1. Создайте обёртку над fetch/axios с базовым URL, заголовками и таймаутом.
2. Реализуйте интерцепторы: добавление токена, маппинг ошибок, логирование.
3. Настройте retry-логику для временных сбоев (5xx, network error).

Задание 8. Аутентификация и защита маршрутов

Контекст: Требуется реализовать регистрацию, вход, разделение прав и защиту приватных маршрутов на клиенте и сервере.

Задание:

1. Реализуйте регистрацию с хэшированием пароля (bcrypt/argon2).
2. Настройте выдачу JWT при успешном входе, проверку токена в middleware.

3. Защитите клиентские маршруты: перенаправление на /login при отсутствии токена.

Задание 9. Тестирование и подготовка к деплою

Контекст: Приложение готово к staging-тестированию. Необходимо проверить работоспособность, настроить окружение и подготовить скрипты развёртывания.

Задание:

1. Напишите 2 unit-теста (фронтенд/бэкенд) и 1 интеграционный тест API.
2. Настройте .env для development, staging, production.
3. Составьте чек-лист деплоя и базовый CI-скрипт (lint → test → build).

Задание 10. Оптимизация, документирование и передача проекта

Контекст: Проект проходит финальную приёмку. Требуется оптимизировать производительность, документировать API и код, подготовить инструкции для передачи.

Задание:

1. Оптимизируйте бандл: tree-shaking, сжатие изображений, lazy loading.
2. Сгенерируйте OpenAPI/Swagger документацию для всех эндпоинтов.
3. Напишите README: установка, запуск, структура проекта, контакты поддержки.

Критерии оценки:

отметка «5»: Задание выполнено в полном объёме с соблюдением необходимой последовательности. Студент работал полностью самостоятельно.

отметка «4»: Практическое задание выполнено студентом в полном объёме и самостоятельно. Допускается отклонение от необходимой последовательности выполнения, не влияющее на правильность конечного результата. Допускаются неточности и небрежность в оформлении результатов задания.

отметка «3»: Практическое задание выполнено и оформлено студентом с помощью преподавателя или хорошо подготовленных и уже выполнивших на «отлично» данную работу студентов. На выполнение задания затрачено много времени.

отметка «2»: Выставляется в том случае, когда студент оказался неподготовленным к выполнению задания. Полученные результаты не позволяют сделать правильных выводов и полностью расходятся с поставленной целью. Обнаружено плохое знание теоретического материала и отсутствие необходимых умений. Руководство и помощь со стороны преподавателя неэффективны из-за плохой подготовки студента.

Ситуационные задачи для производственной практики

по ПМ.03 «Проектирование и разработка информационных систем»

Задание 1. Информационный ресурс учёта продаж

Контекст: Небольшой розничный магазин ведёт учёт продаж в Excel. Требуется веб-приложение для фиксации чеков, контроля остатков и формирования ежедневных отчётов.

Этапы:

1. Спроектировать БД: products, sales, sale_items, users.
2. Реализовать CRUD товаров и оформление продажи (с транзакцией).
3. Добавить автоматический пересчёт остатков и экспорт отчёта в CSV/PDF.
4. Настроить авторизацию: кассир (только продажа), менеджер (отчёты), админ (полный доступ).

Задание 2. Информационный ресурс учёта техники

Контекст: Логистическая компания нуждается в реестре автомобилей и спецтехники с учётом ТО, ремонтов и статусов эксплуатации.

Этапы:

1. Моделировать сущности: equipment, maintenance_log, status_history.
2. Реализовать смену статусов: active → maintenance → retired.

3. Настроить cron-задачу/планировщик для уведомлений о приближающемся ТО.
4. Добавить фильтрацию, поиск и историю изменений.

Задание 3. Информационный ресурс учёта работников

Контекст: HR-отдел хранит данные сотрудников в разрозненных таблицах. Требуется единый реестр с иерархией отделов и защитой персональных данных.

Этапы:

1. Спроектировать БД: employees, departments, contracts, user_roles.
2. Реализовать иерархию отделов (материализованный путь или рекурсивный запрос).
3. Настроить маскирование чувствительных полей (паспорт, телефон) для неавторизованных ролей.
4. Добавить экспорт списка сотрудников и поиск по ФИО/должности.

Задание 4. Информационный ресурс онлайн-библиотеки

Контекст: Учебное заведение хочет цифровую библиотеку с каталогом, поиском, выдачей электронных ресурсов и учётом читателей.

Этапы:

1. БД: books, authors, readers, borrowings, categories.
2. Реализовать поиск (по названию, автору, ISBN) и фильтрацию по категориям.
3. Логика выдачи: проверка лимита, срок возврата, автоматический расчёт просрочки.
4. Добавить панель библиотекаря: управление выдачами, напоминания.

Задание 5. Информационный ресурс сравнения комплектующих ПК

Контекст: IT-порталу нужен инструмент для покомпонентного сравнения процессоров, видеокарт, материнских плат с фильтрами и side-by-side видом.

Этапы:

1. БД: categories, components, specs (гибкая структура: JSONB или EAV).
2. Реализовать админ-панель для добавления компонентов и характеристик.
3. Фронтенд: фильтры, добавление в «список сравнения», таблица сравнения с подсветкой различий.
4. Оптимизировать загрузку при 1000+ позициях (пагинация, кэш).

Задание 6. Информационный ресурс журнала учёта посещаемости

Контекст: Колледжу нужна цифровая замена бумажным журналам: отметка посещаемости преподавателями, просмотр студентами, отчёты для деканата.

Этапы:

1. БД: students, groups, schedule, attendance, teachers.
2. Интерфейс преподавателя: выбор группы/даты, отметка присутствует/отсутствует/опоздал.
3. Автоматический расчёт % посещаемости по месяцу/семестру.
4. Экспорт сводных таблиц, ограничение прав (студент только просмотр).

Задание 7. Информационный ресурс «Умный дом»

Контекст: Разработчику нужна веб-панель для мониторинга и управления IoT-устройствами (температура, освещение, безопасность) через имитацию API.

Этапы:

1. Реестр устройств: devices (тип, статус, last_seen, location).
2. Реализовать WebSocket/SSE для push-обновлений статусов.
3. Очередь команд: ON/OFF, set_temperature, проверка_ack.
4. Мобильно-адаптивный UI, обработка оффлайн-статуса устройств.

Задание 8. Информационный ресурс хода строительных работ

Контекст: Подрядчику нужен инструмент для отслеживания этапов строительства, сроков, ответственных и документации по объектам.

Этапы:

1. БД: projects, tasks, dependencies, files, users.
2. Реализовать цепочку задач с зависимостями (финиш А → старт Б).
3. Расчёт общего прогресса проекта (% завершённых подзадач).
4. Визуализация: timeline/Gantt, загрузка файлов (фотоотчёты, акты).

Задание 9. Информационный ресурс инвентаризации оборудования

Контекст: Организации требуется система учёта IT- и офисного оборудования с QR-кодами, историей перемещений и актами сверки.

Этапы:

1. БД: assets, locations, movement_log, users.
2. Генерация QR-кодов при регистрации оборудования.
3. Веб-сканер QR (через камеру) для быстрой сверки в полях.
4. Отчёты: инвентаризационная ведомость, история перемещений, списание.

Задание 10. Информационный ресурс тестирования сотрудников

Контекст: HR-отделу нужна платформа для онлайн-тестирования при найме и аттестации с автопроверкой, таймером и аналитикой.

Этапы:

1. БД: tests, questions, options, attempts, results, users.
2. Админ-панель: создание тестов, добавление вопросов (MCQ, true/false).
3. Логика теста: таймер, рандомизация порядка вопросов, защита от повторной отправки.
4. Автопроверка, вывод результата, экспорт статистики по отделам.

Критерии оценки:

отметка «5»: Задание выполнено в полном объёме с соблюдением необходимой последовательности. Студент работал полностью самостоятельно.

отметка «4»: Практическое задание выполнено студентом в полном объёме и самостоятельно. Допускается отклонение от необходимой последовательности выполнения, не влияющее на правильность конечного результата. Допускаются неточности и небрежность в оформлении результатов задания.

отметка «3»: Практическое задание выполнено и оформлено студентом с помощью преподавателя или хорошо подготовленных и уже выполнивших на «отлично» данную работу студентов. На выполнение задания затрачено много времени.

отметка «2»: Выставляется в том случае, когда студент оказался неподготовленным к выполнению задания. Полученные результаты не позволяют сделать правильных выводов и полностью расходятся с поставленной целью. Обнаружено плохое знание теоретического материала и отсутствие необходимых умений. Руководство и помощь со стороны преподавателя неэффективны из-за плохой подготовки студента.

5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ (ВИДА ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ)

Код ПК, ОК	Критерии оценки результата (показатели освоённости компетенций)	Формы контроля и методы оценки
ПК 3.1	Оценка «отлично» - изучены требования заказчика по результатам анкет и интервью; изучены типовые решения, обосновано, выбрано и согласовано с заказчиком оптимальное решение; разработано и оформлено техническое задание в полном соответствии с рекомендациями стандартов; разделы технического задания изложены логично и технически грамотно. Оценка «хорошо» - изучены требования заказчика по результатам анкет и интервью; изучены типовые решения, выбрано и согласовано с заказчиком оптимальное решение; разработано и оформлено техническое задание в соответствии с рекомендациями стандартов; разделы технического задания изложены логично и грамотно. Оценка «удовлетворительно» - изучены требования заказчика по результатам анкет и интервью; изучены типовые решения, выбрано и согласовано с заказчиком одно решение; разработано и оформлено техническое задание в соответствии с рекомендациями стандартов; разделы технического задания изложены логично и грамотно.	Контрольные работы, зачеты, квалификационные испытания, защита курсовых и дипломных проектов (работ), экзамены. Интерпретация результатов выполнения практических и лабораторных заданий, оценка решения ситуационных задач, оценка тестового контроля, результатов наблюдений за выполнением различных видов работ во время учебной/ производственной практики
ПК 3.2	Оценка «отлично» - веб приложение разработано и корректно функционирует в полном соответствии с техническим заданием в среде программирования с использованием открытых библиотек; приложение предварительно смоделировано (применены объектные модели); код оформлен в соответствии со стандартами кодирования. Оценка «хорошо» - веб приложение разработано и работоспособно в соответствии с техническим заданием в среде программирования с использованием открытых библиотек; приложение предварительно смоделировано; код оформлен в соответствии со стандартами кодирования. Оценка «удовлетворительно» - веб приложение разработано и работоспособно в соответствии с техническим заданием в среде программирования с использованием открытых библиотек; код оформлен с незначительными отклонениями от стандартов кодирования.	

ПК 3.3	Оценка «отлично» - установлено программное обеспечение для создания резервной копии веб-приложения, создана копия веб приложения, серверные данные зарезервированы, веб-приложение восстановлено из резервной копии (развернуто), веб-сервер настроен; работоспособность проверена, вывод о качестве сделан. Оценка «хорошо» - установлено программное обеспечение для создания резервной копии веб-приложения, создана копия веб приложения, серверные данные зарезервированы, веб-приложение восстановлено из резервной копии (развернуто), веб-сервер настроен без существенных замечаний; работоспособность проверена. Оценка «удовлетворительно» - создана копия веб приложения, серверные данные зарезервированы, веб-приложение восстановлено из резервной копии (развернуто), веб-сервер настроен без существенных замечаний.	
ПК 3.4	Оценка «отлично» - выполнено тестирование веб-приложения в соответствии с тест-планом; результаты тестирования сохранены в системе контроля версий; по результатам тестирования сделаны выводы и внесены предложения по рефакторингу кода; выполнена отладка приложения; результаты отладки сохранены в системе контроля версий; сделаны выводы по результатам отладки. Оценка «хорошо» - выполнено тестирование веб-приложения в соответствии с тест-планом; результаты тестирования сохранены в системе контроля версий; по результатам тестирования сделаны выводы; выполнена отладка приложения; результаты отладки сохранены в системе контроля версий; сделаны выводы по результатам отладки. Оценка «удовлетворительно» - выполнено тестирование веб-приложения в соответствии с тест-планом; результаты тестирования сохранены в системе контроля версий; выполнена отладка приложения; результаты отладки сохранены в системе контроля версий.	
ПК 3.5	Оценка «отлично» - проанализированы источники угроз безопасности; проанализированы методы защиты доступа к данным и защиты кода; предложены и реализованы меры защиты; код сайта и папки проанализированы на предмет наличия вредоносных программ; сделаны выводы о	

	безопасности. Оценка «хорошо» - проанализированы источники угроз безопасности; предложены и реализованы меры защиты; код сайта и папки проанализированы на предмет наличия вредоносных программ; сделаны выводы о безопасности. Оценка «удовлетворительно» - проанализированы источники угроз безопасности; предложены и реализованы меры защиты; код сайта и папки проанализированы на предмет наличия вредоносных программ.	
ПК 3.6	Оценка «отлично» - проанализирован и модифицирован код веб-приложения с помощью системы администрирования; получен работоспособный вариант; проверена работоспособность кода и сделан вывод о результатах оптимизации. Оценка «хорошо» - проанализирован и модифицирован код веб-приложения с помощью системы администрирования; получен практически работоспособный вариант; проверена работоспособность кода и сделан вывод о результатах оптимизации. Оценка «удовлетворительно» - модифицирован код веб-приложения с помощью системы администрирования; получен работоспособный вариант с некоторыми недостатками; проверена работоспособность кода и сделан вывод о результатах оптимизации.	
ПК 3.7	Оценка «отлично» - выбрана с обоснованием выбора система мониторинга работы сайта; система подключена и настроена; настройки обоснованы; выполнен сбор статистики и пояснены его результаты; составлены оригинальные и грамотные тексты для ссылок для размещения на сайтах партнеров и в справочниках. Оценка «хорошо» - выбрана система мониторинга работы сайта; система подключена и настроена; настройки обоснованы; выполнен сбор статистики и пояснены его результаты; применен инструментарий для подбора ключевых словосочетаний; составлены грамотные тексты для ссылок для размещения на сайтах партнеров и в справочниках. Оценка «удовлетворительно» - система мониторинга работы сайта подключена и настроена; выполнен сбор статистики; составлены грамотные тексты для ссылок для размещения на сайтах партнеров и в	

	справочниках.	
OK.01	Обосновывает постановку цели, выбора и применения методов и способов решения профессиональных задач. Оценивает эффективность и качество выполнения профессиональных задач	
OK.02	Использует различные источники, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач	
OK.03	Демонстрирует ответственность за принятые решения. Выполняет самоанализ и коррекцию результатов собственной работы	
OK.04	Взаимодействует с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик. Анализирует работу членов команды (подчиненных)	
OK.05	Демонстрирует грамотность устной и письменной речи. Ясно формулирует и излагает мысли	
OK.06	Соблюдает нормы поведения во время учебных занятий и прохождения учебной и производственной практик. Соблюдает стандарты антикоррупционного поведения	
OK.07	Эффективно выполняет правила ТБ во время учебных занятий, при прохождении учебной и производственной практик. Демонстрирует знания и использование ресурсосберегающих технологий в профессиональной деятельности	
OK.08	Эффективно использует средства физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности	
OK.09	Эффективно использует в профессиональной деятельности необходимую техническую документацию, в том числе на английском языке	