Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гриб Владислав Валерьевич
Должность: Ректор
Дата подписания: 06.11.2026 16:51:04
Уникальный программный ключ:
637517d24e103c3db032acf37e879d98ec1c5bb2f5eb89c29abfcd7f43985447

Образовательное частное учреждение высшего образования

«МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ А.С. ГРИБОЕДОВА»

(ИМПЭ им. А.С. Грибоедова)

ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
(МЕЖДУНАРОДНЫЙ ДИПЛОМАТИЧЕСКИЙ КОЛЛЕДЖ)

УТВЕРЖДАЮ
Директор Международного
дипломатического колледжа
_____ Н.И. Платонова
«16» февраля 2026г.

РАБОЧАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
Разработка и интеграция модулей программного обеспечения

Специальность
09.02.11 Разработка и управление программным обеспечением

Квалификация выпускника:
Программист

на базе среднего общего образования

Форма обучения: очная

Москва

Рабочая программа профессионального модуля «Разработка и интеграция модулей программного обеспечения». Специальность 09.02.11 Разработка и управление программным обеспечением /– М.: ИМПЭ им. А. С. Грибоедова.

Рабочая программа профессионального модуля составлена на основании Федерального государственного образовательного стандарта 09.02.11 Разработка и управление программным обеспечением, утвержденного приказом №138 от 24.02.2025 г. «Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением».

Организация-разработчик:	Образовательное частное учреждение высшего образования «Московский университет имени А. С. Грибоедова»
Ответственный рецензент:	Е.В. Михалёва, к. ф.-м. н. исполнительный директор института информационных систем и инженерно-компьютерных технологий *(Ф.И.О., уч. степень, уч. звание, должность)*

Рабочая программа профессионального модуля рассмотрена и одобрена на заседании Международного дипломатического колледжа «16» февраля 2026 г., протокол №6

Директор Международного
дипломатического колледжа_____/Н.И. Платонова

Согласовано от Библиотеки_____/О. Е. Стёпкина

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

1.1. Цель и планируемые результаты освоения профессионального модуля

В результате изучения профессионального модуля обучающийся должен освоить основной вид деятельности «Разработка и интеграция модулей программного обеспечения» и соответствующие ему общие компетенции и профессиональные компетенции:

1.1.1. Перечень общих компетенций

Код	Наименование общих компетенций
ОК 01.	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02.	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
ОК 03.	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях
ОК 04.	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05.	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09.	Пользоваться профессиональной документацией на государственном и иностранном языках

1.1.2. Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций
ВД 2	Разработка и интеграция модулей программного обеспечения
ПК 2.1.	Проектировать модули программного обеспечения
ПК 2.2.	Разрабатывать модули программного обеспечения
ПК 2.3.	Выполнять интеграцию модулей и компонентов программного обеспечения
ПК 2.4.	Выполнять тестирование и отладку программного обеспечения
ПК 2.5.	Осуществлять документирование программных модулей программного обеспечения

1.1.3. В результате освоения профессионального модуля обучающийся должен:

Иметь практический опыт в	- проектирование модулей ПО с учетом требований заказчика; - создания архитектурных диаграмм и спецификаций модулей; - определение интерфейсов и взаимодействия модулей в системе; - создания модулей программного обеспечения на различных языках программирования; - отладки и тестирования разработанных модулей; - применения структурного и объектно-ориентированного программирования;

	- оптимизации кода и алгоритмов программных модулей для увеличения производительности; - мониторинга и анализа производительности приложений; - интеграции программных модулей и компонентов в единое программное решение; - работы с API и веб-сервисами для взаимодействия между модулями; - работы с интеграционными платформами и инструментами; обеспечения совместимости и стабильности системы; - отладки программного обеспечения на уровне программных модулей; - тестирования программного обеспечения; - формирования тестовых сценариев; - подготовки тестовых платформ (установка операционной системы, дополнительного ПО и другого по необходимости); - оценки объема тестирования ПО с целью определения необходимых ресурсов для его выполнения; - настройки тестовой среды и аппаратных средств для выполнения тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенции; - формирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с установленными регламентами; выполнения тестовых процедур на тестовых данных; - создания технической документации для модулей; - документирования кода, API и интерфейсов; - работы со специализированным ПО по документированию программного кода.

уметь	- проектировать модули, соответствующие бизнес-задачам; - создавать архитектурные диаграммы и документацию; - определять структуру и интерфейсы модулей; - анализировать требования к модулю и определять его функциональность; - проектировать архитектуру модуля, включая выбор подходящих паттернов проектирования и структуры данных; - создавать диаграммы классов, последовательностей и прочих диаграмм для визуализации проектируемого модуля; - выбирать подходящие языки программирования и технологии для реализации модуля; - проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами; - учитывать требования к масштабируемости, производительности и безопасности при проектировании модуля; - проводить анализ и оптимизацию проектируемого модуля для повышения его эффективности и качества; - разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий; - применять паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей; - анализировать требования и определять функциональность модуля; - создавать интерфейсы для взаимодействия с другими модулями и системами; - обеспечивать безопасность, производительность и масштабируемость при разработке модулей;

	- оптимизировать проектируемые модули для повышения их эффективности и качества; - работать с системой контроля версий; - улучшать производительность модулей, выявляя и устраняя узкие места; - проводить анализ и мониторинг производительности приложений; применять инструменты для рефакторинга и оптимизации программного кода; - интегрировать модули и компоненты, обеспечивая их взаимодействие; - работать с API и устанавливать соединения между компонентами; - отслеживать и устранять конфликты и ошибки интеграции; - анализировать и определять зависимости между модулями и компонентами; работать с различными форматами данных и протоколами передачи данных; - анализировать требования к программному обеспечению и составлять планы тестирования; - создавать тестовые сценарии и тест-кейсы для проверки функциональности и соответствия требованиям; - выполнять тестирование программного обеспечения вручную и автоматизировать процесс тестирования; - анализировать результаты тестирования и документировать найденные ошибки; - разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении; - выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования; - использовать системы контроля дефектов ПО; - составлять отчет о выполнении тестирования ПО; - описывать функциональность модулей в документации; - создавать диаграммы для иллюстрации работы модулей; - программировать с использованием комментариев для документирования кода; - использовать специальные метки/теги для отметки важных частей кода в документации; - вести журнал изменений и фиксировать обновления программных модулей; - разбивать модули на логические блоки и описывать каждый блок отдельно; - включать в документацию особенности модулей, такие как ограничения, уязвимости или оптимальные настройки; - проводить регулярное обновление документации при изменении модулей или добавлении нового функционала.

знать	- основные принципы проектирования модулей программного обеспечения; - языки программирования и технологии для реализации модулей; - паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей; - методы анализа требований и способов определения функциональности модуля; - принципы создания интерфейсов для взаимодействия с другими модулями и системами; - принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей;

	- методы анализа и оптимизации проектируемых модулей для повышения их эффективности и качества; - язык программирования, основные конструкции, синтаксис; - паттерны проектирования; - структуры данных; - принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP; - работу с инструментальным программным обеспечением; - методы оптимизации кода и алгоритмов; - эффективные алгоритмы и структуры данных для повышения производительности; - многопоточность в программных модулях; - методы оптимизации сетевых протоколов для ускорения обмена данными; - кэширование данных; - управление памятью; - техники повышения производительности программного обеспечения; - общие принципы функционирования аппаратных, программных и программно-аппаратных средств администрируемой информационнокоммуникационной системы; - международные стандарты локальных вычислительных сетей; - методы и подходы к интеграции модулей и компонентов; - принципы версионирования и управления изменениями при интеграции; - принципы безопасности при интеграции модулей и компонентов; - принципы и методы тестирования программного обеспечения; - основы программирования и архитектуры программного обеспечения; - основы баз данных и SQL-запросов; - инструменты для автоматизации тестирования; - основы разработки и отладки программного обеспечения на разных языках программирования; - понятие дефекта программного обеспечения; - критерии качества ПО; - виды и типы тестирования ПО; - техники ручного тестирования; - техники автоматизированного тестирования; - жизненный цикл дефекта ПО; - принципы работы в системе контроля дефектов; - основные понятия о качестве ПО; - стандарты технической документации; - принципы документирования программного обеспечения; - инструменты для создания технической документации и комментирования кода.

2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

2.1. Структура профессионального модуля очной формы

Структура профессионального модуля очной формы

Коды профессиональных общих компетенций	Наименования разделов профессионального модуля	Суммарный объем нагрузки, час.	Объем профессионального модуля, ак. час.							
			Работа обучающихся во взаимодействии с преподавателем					Самостоятельная работа	Консультации	Промежуточная аттестация
			Обучение по МДК			Практики				
			Всего	В том числе		Учебная	Производственная			
				Лабораторных и практических занятий	Курсовых работ (проектов)					
1	2	3	4	5	6	7	8	9	10	11
ОК 01-05, 09, ПК 2.1. - 2.5.	МДК.02.01. Разработка программных модулей	248	90	78	20			48	2	10
	МДК.02.02. Поддержка и тестирование программных модулей	62	22	24				8	2	6
	МДК.02.03. Технология разработки программного обеспечения	42	16	18					2	6
	Учебная практика	144				144				
	Производственная практика	144					144			

	Экзамен по модулю	6								6
	Всего:	**646**	**128**	**120**	**20**	**144**	**144**	**58**	**6**	**28**

2.2. Тематический план и содержание профессионального модуля (ПМ)

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	**Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная учебная работа обучающихся, курсовая работа (проект)**	**Объем в часах очной формы**
1	**2**	**3**
МДК.02.01. Разработка программных модулей		**248**
Модуль 1. Основы проектирования и архитектуры модулей		**36**
Тема 1.1. Принципы проектирования и анализ требований	**Содержание учебного материала** Основные принципы проектирования модулей программного обеспечения. Методы анализа требований и способов определения функциональности модуля. Принципы высококачественных модулей: сильная связность, слабое зацепление, информационное скрытие. Методы декомпозиции сложной задачи на подзадачи и модули. Определение ответственности и границ модуля.	**8**
	В том числе, практических занятий и лабораторных работ Анализ требований и декомпозиция на модули Создание спецификации программного модуля. Информационное скрытие и проектирование интерфейсов (API).	**6**

	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**4**
Тема 1.2. Спецификации и интерфейсы модулей	**Содержание учебного материала** Разработка формальных и неформальных спецификаций модуля: интерфейсы (API), контракты, пред-/постусловия. Оптимизация взаимодействия между модулями для повышения сопровождаемости, тестируемости и переиспользования. Применение	**8**

	диаграмм компонентов и диаграмм пакетов (UML) для визуализации модульной структуры системы и зависимостей.	
	В том числе, практических занятий и лабораторных работ Визуализация архитектуры с помощью UML (Диаграммы компонентов и пакетов). Модульная архитектура простого приложения. Проектирование интерфейсы программного обеспечения для взаимодействия с другими модулями и системами.	**8**
	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**2**
Модуль 2. Объектно-ориентированное программирование и паттерны		**36**

Тема 2.1. Базовые принципы ООП	**Содержание учебного материала** Принципы ООП. Инкапсуляция, Наследование, Полиморфизм, Абстракция. Их практическое применение. Базовые отношения между классами: Ассоциация, Агрегация, Композиция. Выбор между «has-a» и «is-a».	**6**
	В том числе, практических занятий и лабораторных работ Реализация принципов ООП. Анализ и реализация связей в системе. Проектирование и реализация консольного приложения с применением принципов ООП.	**6**
	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**2**
Тема 2.2. SOLID и	**Содержание учебного материала**	**12**

абстрактные конструкции	Абстрактные классы и интерфейсы. Контрактоориентированное программирование. Принципы проектирования SOLID. Применение диаграмм классов UML для визуализации и проектирования. Модульная архитектура: Принципы, преимущества, примеры (монолиты, микросервисы). Системы контроля версий (Git), системы сборки и менеджеры зависимостей (Maven/Gradle/NuGet). Использование сторонних и создание собственных библиотек (сборка, версионирование, публикация). Введение в архитектурные шаблоны высокого уровня: Сравнение MVC, MVP, MVVM (на уровне концепции, без глубокой реализации). Паттерны проектирования: отношения между классами и объектами (наследование, реализация, ассоциация, композиция, агрегация), интерфейсы, абстрактные классы, порождающие паттерны, паттерны поведения, структурные паттерны, поведенческие паттерны, паттерны объектов. Модели асинхронности: Callback, Promises/Futures, async/await. Пул потоков, задачи (Task), отмена операций. Синхронизация: блокировки, семафоры, проблемы race condition и deadlock. Паттерны параллельного программирования (Producer-Consumer и др.). Применение лямбда-выражений в асинхронных операциях.	
	В том числе, практических занятий и лабораторных работ Сравнение абстрактных классов и интерфейсов. Применение SOLID. Диаграмма классов UML. Модульная архитектура: Рефакторинг монолита в модули. Разработка программных модулей для работы с массивами. Работа через систему контроля версий. Создание и использование библиотеки (NuGet/NPM/Maven). Работа с большими объемами текста. Кодирование и декодирование строк. Построение регулярных выражений. Чтение и запись файлов в разных кодировках. Организация асинхронного вызова методов. Создание программного модуля, который будет выполнять методы в рамках параллельных задач.	**8**

	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к	**2**

	их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	
Модуль 3. Алгоритмы и структуры данных		**36**
Тема 3.1. Основы анализа алгоритмов и базовые концепции	**Содержание учебного материала** Алгоритмы и структуры данных. Оценка сложности алгоритмов. Понятие асимптотической оценки. Большие О-нотации. Анализ временной и пространственной сложности. Анализ худшего, лучшего и среднего случаев. Амортизационный анализ. Базовые структуры данных и их анализ. Линейный поиск, бинарный поиск. Поиск в неотсортированных данных. Простые: пузырьковая, выбором, вставками. Эффективные: быстрая, слиянием, пирамидальная. Timsort, Counting Sort. Сравнительный анализ алгоритмов сортировки. Хеш-таблица и хеш-функция. Коллизии и разрешение коллизий. Методы хеширования и сжатия данных. Эффективность и применение хеш-структур. Деревья и графы. Представление графов и деревьев.	**6**
	В том числе, практических занятий и лабораторных работ Сравнение сложности алгоритмов поиска. Применение рекурсивных алгоритмов. Реализация и сравнение структур данных последовательного доступа. Стек и очередь: приложения и реализации. Сравнительный анализ алгоритмов сортировки. Создание хеш-таблиц и их использование для ускорения поиска данных.	**6**

	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**4**
Тема 3.2. Продвинутые структуры данных, алгоритмы и методы проектирования	**Содержание учебного материала** Бинарные деревья: обходы (pre-order, in-order, post-order). Бинарные деревья поиска. Сбалансированные деревья: AVL, Красно-черные. B-деревья и B+-деревья. Кучи (пирамиды) и приоритетные очереди. Представления графов: матрица смежности, список смежности. Обходы: поиск в глубину (DFS), поиск в ширину (BFS).	**10**

	Кратчайшие пути: Дейкстра, Флойд-Уоршелл. Алгоритм A*. Минимальное остовное дерево: Крускала, Прима. Топологическая сортировка. Методы разработки алгоритмов. Разделяй и властвуй. Жадные алгоритмы и доказательства оптимальности. Динамическое программирование: нисходящее (с мемоизацией) и восходящее, задача о рюкзаке, редакционное расстояние, оптимизация рекуррентных соотношений. Алгоритмы работы с текстовыми данными. Операции над строками. Поиск подстроки (наивный алгоритм поиска, алгоритм Кнута-Морриса-Пратта, алгоритм Бойера-Мура). Проблемы на строках (Задача о рюкзаке, редакционное расстояние). Алгоритмы с использованием хеширования (хеш-функции для строк, алгоритм Рабина-Карпа). Строки и структуры данных (операции с динамическими строками, триальные деревья). Кучи и очереди. Очереди с приоритетом и кучи. Куча и ее применение.	

	В том числе, практических занятий и лабораторных работ Бинарные деревья поиска и балансировка. Нахождение кратчайших путей в графах с использованием алгоритма Дейкстры. Решение задачи о рюкзаке с использованием метода динамического программирования. Жадные алгоритмы и доказательства оптимальности. Хеширование строк и алгоритм Рабина-Карпа. Реализация приоритетных очередей для планирования задач.	**6**
	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**4**
Модуль 4. Реализация модулей для взаимодействия с базами данных		**36**
Тема 4.1. Архитектура доступа к данным и проектирование слоя	**Содержание учебного материала** Архитектурные подходы и паттерны. Data Access Layer (DAL). Паттерн Repository: абстракция над хранилищем данных. Паттерн Unit of Work. Active Record и Data Mapper - сравнение подходов. CQRS (Command Query Responsibility Segregation): разделение операций. Управление транзакциями и изоляцией. Транзакции: явные и	**6**

1
2

	неявные, распределённые транзакции. Уровни изоляции и их влияние на производительность. Оптимистическая блокировка. Пессимистическая блокировка и устойчивость к сбоям. Миграции и версионирование схемы БД. Database-First подходы. Инструменты миграций: Flyway, DbUp. Стратегии деплоя изменений БД. Blue-Green миграции и zero-downtime deployments.	

	В том числе, практических занятий и лабораторных работ Сравнение архитектурных паттернов DAL. Реализация CQRS на примере системы заказов. Транзакции и уровни изоляции. Миграции БД: стратегии и инструменты.	**8**
	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**6**
Тема 4.2. Реализация и технологии ORM, обеспечение безопасности в БД	**Содержание учебного материала** Технологии и подключение к БД. ADO.NET, Entity Framework Core, Dapper, другие ORM. Абстракция провайдеров данных. Connection Strings и управление подключениями. Пулы соединений и настройка. Мониторинг подключений. Конфигурация маппинга (Fluent API, атрибуты). Отношения: один-ко-многим, многиеко-многим, наследование. Механизм отслеживания изменений. Микро-ORM для высокопроизводительных сценариев. Реализация CRUD и выполнение запросов. Создание. Чтение. Хранимые процедуры. Обновление. Оптимистическая блокировка. Удаление. Пагинация. Динамические запросы. Оптимизация работы с БД. Проблема N+1 и способы её решения. Индексы: создание, анализ, покрывающие индексы. Анализ планов запросов. Кэширование запросов: первого и второго уровня. Пакетные операции. Автоматические повторные попытки. Безопасность в модулях данных. Принцип наименьших привилегий для пользователей БД. Шифрование данных: на уровне приложения и БД. Маскирование данных в логах и отладке. Аудит изменений данных (кто, когда, что изменил).	**8**
	В том числе, практических занятий и лабораторных работ	**6**

	Настройка подключений и пулов соединений. Проблема N+1 и стратегии загрузки данных. Асинхронное программирование в DAL. Кэширование данных: стратегии и реализация. Оптимизация массовых операций.	
	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**2**
Модуль 5. Реализация модулей для взаимодействия с пользователем		**36**
Тема 5.1. Основы разработки пользовательских интерфейсов	**Содержание учебного материала** Виды пользовательского интерфейса (командная строка, графический, речевой). Основные этапы и принципы разработки графического пользовательского интерфейса. Технологии и инструменты разработки графического пользовательского интерфейса. Компоненты графического пользовательского интерфейса. Типы элементов управления. Компоновка элементов управления. События. Обработчики событий. Работа с окнами. Основные методы работы с окнами. Создание окна: функции и классы. Открытие и закрытие окон. Взаимодействие с окнами (например, передача данных).	**8**
	В том числе, практических занятий и лабораторных работ Проектирование главного окна приложения с несколькими панелями и элементами управления. Разработка модулей многооконного приложения. Разработка стилей для приложения для улучшения взаимодействия с пользователем.	**6**

	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к	**6**
	их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	
Тема 5.2. Обработка событий, валидация, многопоточность	**Содержание учебного материала** Примеры валидации (проверка формата ввода). Сообщения об ошибках и уведомления пользователя. Использование регулярных выражений для валидации. Многопоточность и асинхронная работа окон. Многопоточность в GUI-приложениях. Проблемы синхронизации потоков. Использование асинхронных вызовов для долго выполнимых операций. Значение стиля в UX/UI дизайне. Основы теории цвета. Работа с цветом и шрифтами. Стилизация. Работа с текстом, изображениями. Построение графиков и диаграмм. Библиотеки для построения графиков и диаграмм.	**8**
	В том числе, практических занятий и лабораторных работ Разработка модулей для представления текстовой информации. Разработка модулей для представления информации в виде графиков и диаграмм. Реализация загрузки данных из интернета в фоновом режиме без блокировки основного потока приложения. Разработка формы регистрации с элементами ввода и проверкой корректности введенных данных.	**6**
	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**2**

Модуль 6. Качество, надежность и инженерия программных модулей		**38**
Тема 6.1. Принципы безопасности, производительности и масштабируемости программных модулей	**Содержание учебного материала** Основные понятия: безопасность программного обеспечения, производительность модулей, масштабируемость архитектуры. Принципы безопасного программирования. Уязвимости на уровне модулей. Криптография в модулях. Безопасная конфигурация и зависимости. Методология оптимизации производительности. Оптимизация на уровне кода. Алгоритмическая оптимизация: сложность, кэш-локальность. Мемоизация и кэширование результатов вычислений. Ленивые вычисления. Оптимизация ввода-вывода и сети. Принципы проектирования масштабируемых	**10**
		15
	систем. Паттерны масштабируемости. Микросервисная архитектура и масштабируемость. Принципы отказоустойчивых систем. Работа с ошибками в модулях. Метрики: Prometheus, Grafana, мониторинг бизнес-метрик. Логи: ELK Stack, Loki, централизованное логирование. Трейсинг: OpenTelemetry, Jaeger, распределенная трассировка. Проактивный мониторинг. Энергоэффективность и экологичность. Управление затратами в облаке. Современные облачные практики.	
	В том числе, практических занятий и лабораторных работ Оптимизация проектируемых модулей для повышения их эффективности и качества. Решение задач на оптимизацию алгоритмов. Анализ и мониторинг производительности приложений. Обеспечение производительности и масштабируемости при разработке модулей программного обеспечения. Улучшение производительности модулей посредством выявления и устранения узких мест. Обеспечение безопасности при разработке модулей программного обеспечения.	**12**

	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**16**
Курсовой проект	**Содержание учебного материала** Выбор темы курсовой работы. Подбор литературы к написанию курсовой работы. Написание содержания курсовой работы. Написание введения курсовой работы. Написание теоретической части курсовой работы. Написание практической части курсовой работы. Написание заключения курсовой работы. Оформление списка литературы курсовой работы. Подготовка речи к защите курсовой работы. Подготовка компьютерной презентации к защите курсовой работы. **Тематика курсовых работ** Создание и внедрение модуля сквозного аудита изменений данных в распределённой корпоративной системе. Разработка и интеграция модуля динамического ценообразования в систему	**20**

	управления онлайн-продажами. Создание универсального модуля импорта/экспорта данных для платформы электронной отчётности. Разработка и внедрение модуля управления жизненным циклом заявок в корпоративный портал. Создание и интеграция модуля генерации и верификации электронных подписей в систему документооборота. Разработка модуля планирования ресурсов для интеграции в систему управления проектами. Создание модуля бронирования и управления расписанием для системы онлайнзаписи на услуги. Разработка и интеграция модуля рекомендаций на основе истории покупок в интернетмагазин. Создание высоконагруженного модуля управления сессиями пользователей для вебпортала. Разработка и внедрение модуля расчёта сложных налогов и сборов в бухгалтерский комплекс. Создание и интеграция модуля управления промо-акциями и скидками в CRM-систему розничной сети. Разработка модуля конфигуратора сложных продуктов (например, автомобиля или компьютера) для сайта. Создание модуля автоматического составления отчётности по KPI для системы управления эффективностью. Разработка и интеграция модуля уведомлений (Email, SMS, Push) в корпоративную информационную систему. Разработка и интеграция модуля репликации данных между разнородными БД для системы аналитики. Создание модуля витрины данных для отчётного модуля финансового департамента. Разработка и внедрение модуля архивного хранения редко используемых данных с прозрачным доступом.	

	Создание модуля работы с пространственными данными для системы кадастра. Разработка и интеграция модуля синхронизации справочников между центральной и периферийными системами. Создание модуля миграции данных между версиями системы или при смене платформы Разработка и интеграция модуля интерактивной панели управления с	

	графиками для BI-системы. Создание модуля конструктора отчётов с drag-and-drop интерфейсом для бизнеспользователей. Разработка мобильного модуля для проведения инвентаризации с интеграцией в складскую систему. Создание и интеграция модуля видеоконференций в корпоративный портал. Разработка модуля чат-бота для службы технической поддержки с интеграцией в тикет-систему. Создание модуля для проведения онлайн-опросов и сбора анкет в системе управления персоналом. Разработка и внедрение модуля двухфакторной аутентификации (2FA) для корпоративных веб-приложений. Создание модуля мониторинга состояния системы и бизнес-метрик в реальном времени. Разработка и интеграция модуля распределённого кэширования для ускорения работы высоконагруженного веб-сервиса. Создание модуля управления конфигурацией и секретами для облачных микросервисов. Разработка модуля балансировки нагрузки и управления сессиями для кластера приложений. Создание и внедрение модуля защиты от DDoS-атак и сканирования уязвимостей на уровне приложения. Разработка модуля автоматического масштабирования вычислительных ресурсов для фоновых задач. Создание модуля сбора и анализа логов из распределённых компонентов системы. Разработка и интеграция модуля тестирования производительности API шлюза. Разработка и интеграция модуля электронного документооборота с внешними операторами. Создание модуля управления взаимоотношениями с клиентами для малого бизнеса.	

	Разработка и интеграция модуля управления цепочкой поставок в ERP-систему. Создание модуля системы бронирования и продажи билетов для транспортной компании. Разработка и внедрение модуля управления обучением в корпоративный портал. Создание модуля биллинга и начисления платежей для телеком-оператора.	

	Разработка модуля трекинга посылок и уведомлений для службы доставки. Создание и интеграция модуля управления арендой и бронированием для гостиничного бизнеса. Разработка модуля технической поддержки и управления инцидентами. Создание модуля управления маркетинговыми кампаниями и рассылками. Разработка и интеграция модуля управления активами и техническим обслуживанием. Создание модуля системы резервирования столиков для сети ресторанов. Разработка модуля управления библиотечным фондом и арендой книг. Создание и интеграция модуля управления медицинскими назначениями и историей болезни в больнице. Разработка модуля системы голосования и сбора мнений для внутреннего портала. Создание модуля управления расписанием и нагрузкой для учебного заведения. Разработка и интеграция модуля управления флотом и логистикой для транспортной компании. Создание модуля системы управления контентом для корпоративного сайта.	
Консультации		**2**
Промежуточная аттестация в форме экзамена		**8**
МДК.02.02 Поддержка и тестирование программных модулей		**62**
Модуль 1. Отладка и тестирование программных модулей		**34**
Тема 1.2. Отладка программного модуля и обработка исключений	**Содержание учебного материала** Понятие отладки. Понятия ошибки, дефекта, сбоя, отказа. Типы ошибок. Инструменты для отладки. Процесс пошаговой отладки (установка точек останова, шаг за шагом выполнение кода, просмотр состояния переменных, выполнение отдельных частей кода). Стратегии поиска ошибок. Процесс управления инцидентами и проблемами. Системы отслеживания ошибок (Bug Tracking Systems). Документирование процесса отладки. Суть и роль исключений в программе. Классификация исключительных ситуаций. Синтаксическая конструкция для управления исключениями. Логика (стратегия) работы с исключениями. Методы отладки кода с использованием исключений и логирования.	**6**

	В том числе, практических занятий и лабораторных работ Понятие отладки. Ошибки, дефекты, сбои, отказы. Типы ошибок (Синтаксические, логические, ошибки времени выполнения). Процесс пошаговой отладки. Работа с системой отслеживания ошибок (Bug Tracking System): создание и	**8**

	обработка заявок. Документирование процесса отладки и управления инцидентами. Сущность и назначение исключений. Исключения времени выполнения. Механизм обработки. Конструкция try-catch-finally. Логика работы. Создание иерархии собственных исключений. Обработка исключений в реальном сценарии. Обработка ошибок и исключение в RESTful API.	
	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**4**

Тема 1.2. Основы тестирования ПО	**Содержание учебного материала** Понятие процесса тестирования программного обеспечения. Этапы процесса тестирования программного обеспечения. техники ручного тестирования и автоматизированного тестирования. Виды функционального тестирования: модульное тестирование, интеграционное тестирование, системное тестирование, приемочное тестирование. Методы функционального тестирования: тестирование черного ящика, тестирование белого ящика, серое тестирование. Методы и уровни функционального тестирования. иды нефункционального тестирования: тестирование производительности, тестирование безопасности, тестирование удобства использования (Usability Testing), тестирование совместимости. Методы и инструменты для проведения нефункционального тестирования. Нагрузочное тестирование и стресс-тестирование.	**6**
	В том числе, практических занятий и лабораторных работ Понятие и этапы процесса тестирования. Анализ требований к программному обеспечению и составление планов тестирования. Техники ручного тестирования. Методы "черного", "белого" и "серого" ящика. Тестирование удобства использования.	**8**
	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**2**
Модуль 2. Автоматизация и поддержка программных модулей		**18**

Тема 2.1. Автоматизация тестирования	**Содержание учебного материала** Автоматизация функционального тестирования. Инструменты для автоматизации функционального тестирования (Selenium, TestComplete). Преимущества и недостатки автоматизации тестирования. Инструменты для автоматизации тестирования. Разработка автоматизированных тестовых сценариев. Интеграция автоматизированных тестов в процесс разработки. CI/CD и автоматизация тестирования. Модульные тесты. Тестирование интеграции. Методы и инструменты для тестирования интегрированных решений. Методы разработки тестовых случаев: анализ граничных значений, эквивалентное разделение, таблица принятия решений. Критерии покрытия тестами. Выбор инструментов для разработки и выполнения тестов.	**6**
	В том числе, практических занятий и лабораторных работ Разработка модульных тестов. Интеграция тестов в CI/CD пайплайн. Тестирование интеграции. Написание и выполнение тестов для проверки взаимодействия между модулями. Разработка модульных тестов с проверкой результатов тестирования с учетом погрешности. Тестирование RESTful API. Тестирование производительности.	**4**
	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций	**-**

	преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	

Тема 2.2. Поддержка программных модулей	**Содержание учебного материала** Использование искусственного интеллекта и машинного обучения в тестировании. Работы, выполняемые при поддержке программного обеспечения. Исправление дефектов. Ревьюирование кода. Рефакторинг кода. Оптимизация кода. Стандарты разработки и оформления документации на программное обеспечение. Принципы документирования программного обеспечения. Виды тестовой документации. Тестовая документация подготовительного этапа. Тестовая документация на этапе завершения работ по тестированию. Тестовые случаи и сценарии. Написание тестовых случаев. Структура тестового сценария. Отчет о дефектах. Инструменты для создания технической документации и комментирования кода.	**4**
	В том числе, практических занятий и лабораторных работ Разработка документации на программное обеспечение в соответствии со стандартами. Ведение журнала изменений и фиксация обновления программных модулей. Ревьюирование, рефакторинг и оптимизация кода. Разработка Программы и методики испытаний.	**4**
	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**-**
Консультации		**2**
Промежуточная аттестация в форме экзамена		**6**
МДК.02.03 Технология разработки программного обеспечения		**42**
Модуль 1. Среда и инструменты разработки программного обеспечения		**34**
Тема 1.1. Технологии и инструменты	**Содержание учебного материала** Системы статического анализа кода, проверяющие соблюдение принципов SOLID и паттернов. Инструменты UML для проектирования. Среды разработки (IDE) с	**6**

реализации модулей	поддержкой рефакторинга и визуализации зависимостей. Интерактивные среды для анализа алгоритмов. Библиотеки для работы со сложными структурами данных. Профайлеры для оценки сложности алгоритмов. ORM-фреймворки и микробиблиотеки. Инструменты для построения и анализа запросов. Системы управления миграциями БД.	
	В том числе, практических занятий и лабораторных работ Проектирование диаграммы классов модуля. Написание кода модуля с применением паттернов. Реализация модуля с использованием специфичной структуры данных. Написание юнит-тестов, проверяющих корректность работы алгоритма на граничных условиях. Профилирование модуля для выявления узких мест. Разработка модуля доступа к данным с использованием ORM. Написание миграции для изменения схемы таблицы.	**8**
	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем). Подготовка к практическим работам с использованием методических рекомендаций преподавателя, оформление лабораторно-практических работ, отчетов и подготовка к их защите. Подготовка рефератов, докладов, презентаций на заданные темы. Выполнение индивидуальных заданий.	**-**

Тема 1.2. Технологии создания пользовательских интерфейсов и интеграции	**Содержание учебного материала** Обзор GUI-фреймворков для создания оконных приложений. Концепция привязки данных и шаблон MVVM. Средства стилизации. Технологии создания API для взаимодействия модулей: REST, gRPC, GraphQL. Инструменты для документирования API. Клиенты для тестирования API. Фреймворки для модульного тестирования GUI. Инструменты для автоматизации E2E-сценариев. Методы тестирования асинхронных и событийно-ориентированных интерфейсов. Профайлеры и APM-системы для мониторинга производительности модулей в runtime. Статические анализаторы безопасности. Принципы безопасного кодирования для модулей. Системы управления зависимостями и сборкой для разных языков. Стратегии управления версиями модулей. Инструменты контейнеризации для изоляции среды модуля (Docker). Принципы CI/CD применительно к модульной	**10**
	разработке. Обзор платформ (GitHub Actions, GitLab CI). Этапы пайплайна для модуля: сборка, статический анализ, модульное и интеграционное тестирование, упаковка, публикация в артефакт-репозиторий.	

	В том числе, практических занятий и лабораторных работ Создание модуля GUI для отображения и редактирования данных из модуля DAL. Реализация формы с валидацией ввода и привязкой к модели данных. Разработка модуля, предоставляющего REST API для своих функций. Документирование API с помощью Swagger. Создание клиентского модуля, который потребляет этот API. Написание модульных тестов для GUI-компонентов (проверка реакции на ввод). Создание интеграционного теста, который проверяет полный сценарий. Профилирование модуля для поиска утечек памяти или узких мест. Запуск сканирования модуля на уязвимости в зависимостях. Анализ отчета и применение рекомендаций. Создание конфигурационного файла сборки для модуля, включающего зависимости, этапы компиляции, упаковки и запуска тестов. Сборка модуля в Docker-контейнер. Настройка CI-пайплайна на GitHub Actions для автоматической сборки и тестирования модуля при каждом коммите. Конфигурация этапа публикации успешно собранного артефакта (JAR-файла) в GitHub Packages.	**10**
	Самостоятельная работа Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем).	**-**
Консультации		**2**
Промежуточная аттестация в форме дифференцированного зачета		**10**
Всего:		**42**

Учебная практика **1. Содержание работ** Анализ требований и проектирование модуля: 1. Изучение предметной области: - выбрать вариант темы из списка; - определить ключевые роли пользователей (не менее 3); - сформулировать функциональные требования к модулю (не менее 5). 2. Разработка технического задания: - описать назначения и цели модуля; - создать диаграмму вариантов использования (UML Use Case) для ключевых функций (не менее 5); - спроектировать диаграмму	**144**

последовательности (UML Sequence Diagram) для сложного сценария. 3. Проектирование пользовательского интерфейса: - создать эскизы основных экранов (не менее 4: главная страница, каталог, корзина, личный кабинет); - разработать кликабельный прототип интерфейса с навигацией; - провести юзабилити-тестирования прототипа и фиксация рекомендаций. Разработка и реализация модуля: 1. Настройка среды разработки: - установить и настроить IDE, СУБД, систему контроля версий; - инициализировать репозиторий проекта, создать структуры папок; - настроить подключения к БД через ORM. 2. Реализация базы данных и CRUD-операций: - спроектировать схемы БД (не менее 4–5 связанных таблиц); - написать DDL-скрипты для создания таблиц; - наполнить БД тестовыми данными (не менее 20 записей); - реализовать backend-логики для CRUD одной из ключевых сущностей. 3. Разработка пользовательского интерфейса и бизнес-логики: - создать веб-форм или оконных интерфейсов для отображения и редактирования данных; - реализовать валидации данных на стороне клиента и сервера; - добавление функции поиска/фильтрации данных по одному критерию. Тестирование, отладка и документирование: 1. Модульное и интеграционное тестирование: - написать и выполнить не менее 5 модульных тестов для ключевых функций; - провести интеграционного тестирования взаимодействия 2–3 компонентов. 2. Отладка и исправление дефектов: - фиксировать в системе учёта дефектов не менее 3 типовых ошибок; - локализовать и устранить дефекты. 3. Техническое документирование: - составить «Руководства пользователя» для модуля; - написать «Инструкции по запуску проекта» (установка зависимостей, настройка окружения); - сформировать отчёта о тестировании (тест-кейсы). Интеграция модуля и защита работы: 1. Интеграция модуля в учебный проект: - обеспечить взаимодействие модуля с другими компонентами системы (например, модулем авторизации); - настроить API для обмена данными. **2. Требования к реализации бизнес-логики** При выполнении учебной практики студент должен реализовать базовую бизнес-логику для выбранного модуля информационной системы в соответствии с предметной областью. Бизнес-логика должна включать следующие элементы: Основная операция (Транзакция) Хранимая процедура или метод, выполняющий ключевое действие системы. Примеры: Оформление продажи: списание товара, создание чека. Запись на приём: проверка доступности времени, создание записи.	

Выдача книги: обновление статуса книги, фиксация даты возврата. Изменение статуса (Жизненный цикл сущности) Процедура или триггер, сопровождающие изменение состояния объекта. Примеры: При отмене заказа: возврат товаров на остаток, изменение статуса заказа.	

При увольнении сотрудника: блокировка учётной записи, передача дел. При завершении обслуживания: закрытие заявки, формирование акта. Контроль целостности и бизнес-правил (Защита от ошибок) Триггер или проверка, запрещающая некорректные действия. Примеры: Нельзя выдать книгу читателю с просроченной задолженностью. Нельзя оформить продажу при отсутствии товара на складе. Нельзя назначить встречу на недоступное время. Общие требования: Данные должны быть реалистичными: использовать синтетические данные, сгенерированные скриптами, или обезличенные реальные данные (при работе в организации). Объём данных: для тестирования операций — не менее 20 записей по каждой ключевой сущности. Документирование: каждый сценарий должен быть описан в технической документации (руководство пользователя, инструкция по запуску). Тестирование: для каждого сценария должны быть написаны модульные тесты (не менее 5 тестов на ключевые функции). **3. Варианты для выполнения задания** Порядок выбора задания: 1. Выбор темы: Номер варианта определяется по первой буквой фамилии студента. 2. Адаптация под место практики: На базе колледжа: Студент выполняет задание строго по описанию варианта, используя указанные ключевые сущности. В профильной организации: Студент выполняет задание на примере деятельности организации, согласуя список сущностей с руководителем практики. ВАРИАНТ 1 (Фамилии на «А») Тема: Туристическое агентство Задание: Проектирование и разработка модуля для учета клиентов, туров, бронирований и отелей. Ключевые сущности: Клиенты, Туры, Отели, Заказы (Бронирования), Страны/Курорты, Статусы оплаты. ВАРИАНТ 2 (Фамилии на «Б»)	

Тема: Библиотечный фонд Задание: Создание модуля для учета книг, выдачи и возврата литературы. Ключевые сущности: Книги, Читатели, Авторы, Жанры, Выдача книг (Учет), Штрафы. ВАРИАНТ 3 (Фамилии на «В») Тема: Ветеринарная клиника	

Задание: Проектирование и разработка модуля для учета пациентов, владельцев и истории болезней. Ключевые сущности: Пациенты (Животные), Владельцы, Врачи, Приемы (Визиты), Услуги, Диагнозы. ВАРИАНТ 4 (Фамилии на «Г») Тема: Гостиничный комплекс Задание: Разработка модуля для управления номерным фондом и заселением. Ключевые сущности: Номера, Категории номеров, Гости, Бронирования, Дополнительные услуги, Счета. ВАРИАНТ 5 (Фамилии на «Д», «Е», «Ё») Тема: Служба доставки еды Задание: Создание модуля для учета заказов, курьеров и маршрутов. Ключевые сущности: Клиенты, Блюда, Категории блюд, Заказы, Состав заказа, Курьеры. ВАРИАНТ 6 (Фамилии на «Ж») Тема: Жилищно-коммунальное хозяйство (ЖКХ) Задание: Разработка модуля для учета жильцов, квартир и начислений. Ключевые сущности: Жильцы (Собственники), Квартиры, Виды услуг (Тарифы), Приборы учета (Счетчики), Показания счетчиков, Квитанции. ВАРИАНТ 7 (Фамилии на «З») Тема: Завод (складской учет) Задание: Проектирование и разработка модуля для учета материалов, поставщиков и накладных. Ключевые сущности: Материалы/Товары, Категории материалов, Склады, Поставщики, Приходные накладные, Расходные накладные. ВАРИАНТ 8 (Фамилии на «И», «Й») Тема: Интернет-провайдер Задание: Создание модуля для учета абонентов, тарифов и баланса. Ключевые сущности: Абоненты, Договоры, Тарифные планы, Услуги, Платежи, Оборудование (Роутеры и пр.). ВАРИАНТ 9 (Фамилии на «К») Тема: Кадровое агентство Задание: Разработка модуля для хранения резюме и вакансий. Ключевые сущности: Соискатели, Резюме, Навыки, Работодатели, Вакансии, Собеседования (Отклики). ВАРИАНТ 10 (Фамилии на «Л»)	

Тема: Логистическая компания Задание: Проектирование и разработка модуля для учета автопарка и грузоперевозок. Ключевые сущности: Автомобили, Водители, Заказчики, Рейсы (Путевые листы), Грузы, Расход топлива. ВАРИАНТ 11 (Фамилии на «М», «Н») Тема: Медицинский центр (Поликлиника)	

Задание: Создание модуля «Регистратура» для учета пациентов, врачей и приемов. Ключевые сущности: Пациенты, Врачи, Специализации врачей, Кабинеты, Запись на прием (Талоны), Медицинские карты. ВАРИАНТ 12 (Фамилии на «О») Тема: Онлайн-школа (Курсы) Задание: Разработка модуля для учета студентов, курсов и успеваемости. Ключевые сущности: Студенты, Преподаватели, Курсы, Группы, Уроки, Оценки/Результаты тестов. ВАРИАНТ 13 (Фамилии на «П») Тема: Прокат автомобилей (Каршеринг) Задание: Проектирование и разработка модуля для учета аренды и состояния авто. Ключевые сущности: Автомобили, Тарифы аренды, Клиенты, Договоры аренды, Штрафы/Повреждения, Техническое обслуживание. ВАРИАНТ 14 (Фамилии на «Р») Тема: Ресторан Задание: Создание модуля для учета ингредиентов, технологических карт и продаж. Ключевые сущности: Ингредиенты (Склад), Блюда, Технологические карты (Состав блюда), Столики, Заказы, Сотрудники. ВАРИАНТ 15 (Фамилии на «С») Тема: Страховая компания Задание: Разработка модуля для учета полисов и страховых случаев. Ключевые сущности: Клиенты, Виды страхования, Полисы, Агенты, Страховые случаи (Убытки), Выплаты. ВАРИАНТ 16 (Фамилии на «Т») Тема: Таксопарк Задание: Проектирование и разработка модуля для диспетчерской: водители, заказы, смены. Ключевые сущности: Водители, Автомобили, Диспетчеры, Клиенты, Поездки (Заказы), Тарифы. ВАРИАНТ 17 (Фамилии на «У») Тема: Университет (Деканат) Задание: Создание модуля для учета факультетов, студентов и сессий. Ключевые сущности: Факультеты/Кафедры, Специальности, Группы, Студенты, Дисциплины, Ведомости (Оценки).	

ВАРИАНТ 18 (Фамилии на «Ф») Тема: Фитнес-клуб Задание: Разработка модуля для учета абонементов и посещений. Ключевые сущности: Клиенты, Виды абонементов, Проданные абонементы, Тренеры, Групповые занятия,	

Журнал посещений. ВАРИАНТ 19 (Фамилии на «Х») Тема: Химчистка Задание: Проектирование и разработка модуля для учета услуг, квитанций и выдачи. Ключевые сущности: Клиенты, Виды услуг (Прайс-лист), Приемные пункты, Заказы (Квитанции), Вещи в заказе, Статусы готовности. ВАРИАНТ 20 (Фамилии на «Ц», «Ч») Тема: Частное охранное предприятие Задание: Создание модуля для учета объектов, сотрудников и инцидентов. Ключевые сущности: Объекты охраны, Охранники, Экипажи ГБР, Смены/Графики, Тревожные вызовы (Инциденты), Оружие/Спецсредства. ВАРИАНТ 21 (Фамилии на «Ш», «Щ») Тема: Школа искусств Задание: Разработка модуля для учета учеников, инструментов и концертов. Ключевые сущности: Ученики, Преподаватели, Направления (Музыка/Рисование), Расписание занятий, Инструменты (Инвентарь), Отчетные концерты. ВАРИАНТ 22 (Фамилии на «Ы», «Э») Тема: Электроника (Интернет-магазин) Задание: Проектирование и разработка модуля магазина: каталог, корзина, заказы. Ключевые сущности: Товары, Категории товаров, Характеристики товаров, Клиенты, Заказы, Позиции заказа. ВАРИАНТ 23 (Фамилии на «Ю») Тема: Ювелирный салон Задание: Создание модуля для учета изделий, проб и продаж. Ключевые сущности: Изделия, Типы изделий (Кольца/Серьги), Металлы/Пробы, Драгоценные камни, Поставщики, Продажи (Чеки). ВАРИАНТ 24 (Фамилии на «Я») Тема: Ярмарка (Выставка) Задание: Разработка модуля для учета участников, павильонов и посетителей.	

Ключевые сущности: Павильоны/Стенды, Участники (Компании), Организаторы, Договоры аренды, Оборудование, Билеты посетителей.	

Производственная практика **1. Содержание работ** Анализ и планирование модернизации системы: 1. Создать аудит существующего проекта: - провести полный анализ архитектуры, кода и производительности модуля; - выявить 3–5 ключевых архитектурных ограничений и	**144**

проблем производительности; - составить отчёт о техническом состоянии проекта. 2. Разработать техническое задание на модернизацию системы: - создать описание 2 новых функциональных блоков для расширения системы; - разработать UML-диаграммы (активности, последовательности, состояний) для новых сценариев; - сформировать схему изменений в базе данных и требования к интерфейсам. 3. Составить план устранения технического долга: - определить 3–4 критических пункта технического долга в существующем коде; - создать план рефакторинга с оценкой трудозатрат и приоритетами выполнения; - разработать стратегию миграции данных при изменении схемы БД. Реализация модернизации и оптимизации: 1. Реализовать систему кэширования для повышения производительности: - создать механизм кэширования часто запрашиваемых данных; - реализовать стратегию инвалидации кэша при изменениях данных; - интегрировать кэширование в существующие API-эндпоинты. 2. Разработать модуль аналитики и отчётности: - создать интерфейс для построения пользовательских отчётов с фильтрами; - реализовать визуализацию данных с использованием графических библиотек; - сделать систему экспорта отчётов в форматы PDF и Excel. 3. Создать расширенную систему управления доступом (RBAC): - реализовать модель ролей, разрешений и политик доступа; - создать административный интерфейс для управления правами пользователей; - интегрировать проверку прав доступа во все критические операции системы. 4. Внедрить механизмы обеспечения безопасности: - реализовать шифрование чувствительных данных в базе данных; - создать систему аудита всех действий пользователей; - внедрить защиту от распространённых уязвимостей (XSS, CSRF, SQL-инъекции). Реализация интеграции и автоматизации: 1. Создать интеграцию с внешними сервисами: 1. - разработать модуль взаимодействия с платёжным шлюзом (эмуляция API); - реализовать систему уведомлений через Email, SMS и мессенджеры; - создать интерфейс мониторинга статуса интеграций. 2. Реализовать автоматизированное развёртывание: - создать Docker-контейнеры для всех компонентов системы; - разработать скрипты автоматического деплоя в различные среды; - сделать конфигурационные файлы для dev, staging и production окружений. 3. Создать систему мониторинга и алертинга: - внедрить сбор метрик производительности приложения; - разработать дашборды для мониторинга состояния системы; - реализовать систему оповещения о критических событиях. Проведение тестирование и оптимизации системы: 1. Провести нагрузочное тестирование системы: - создать сценарии нагрузочного тестирования для 1000+ пользователей; - реализовать автоматизированный запуск тестов производительности; - проанализировать результаты и выявить узкие места системы. 2. Оптимизировать производительность базы данных; - создать и проанализировать индексы для часто используемых запросов. -	

реализовать оптимизацию сложных SQL-запросов; - сделать механизм шардирования или партиционирования данных при необходимости. 3. Провести комплексное тестирование системы: - создать набор автоматических тестов (юнит, интеграционные, E2E); - реализовать тестирование отказоустойчивости и восстановления после сбоев; - провести security-тестирование на выявление уязвимостей.	

Создание документации: 1. Создать полный пакет эксплуатационной документации: - разработать руководство администратора по установке и настройке; - создать инструкцию по обновлению системы и миграции данных; - подготовить регламенты обслуживания и аварийного восстановления. 2. Разработать пользовательскую документацию: - создать руководство пользователя с иллюстрациями и примерами; - подготовить видеоматериалы или скринкасты по использованию системы; - сделать FAQ и справочные материалы. **2. Требования к условиям выполнения** При выполнении задач интеграции, импорта и тестирования обязательно использовать реалистичные данные: На предприятии: обезличенные реальные данные организации. В колледже: студент обязан сгенерировать синтетические данные с помощью специализированных инструментов или написанных скриптов. Объем данных должен соответствовать задачам: Для импорта и обработки — не менее 1 000 записей. Для нагрузочного тестирования — не менее 100 000 записей. Данные должны быть релевантными предметной области и включать корректные форматы (даты, имена, номера, суммы и т.д.). Все операции по обслуживанию системы (резервное копирование, импорт/экспорт данных, мониторинг) должны быть оформлены в виде исполняемых скриптов, а не выполняться вручную через интерфейс. Скрипты должны сопровождаться документацией по их использованию и настройке. **3. Варианты для выполнения задания** Порядок выбора и выполнения: Выбор темы: Номер варианта определяется по первой буквой фамилии студента (тема совпадает с учебной практикой). Адаптация: На базе колледжа: студент выполняет роль администратора, самостоятельно генерируя данные и сценарии нагрузки. В профильной организации: задание на практику выполняется на примере профильной организации. ВАРИАНТ 1 (Фамилии на «А») Тема: Туристическое агентство Задание: Разработать и внедрить модуль автоматического обновления каталога туров и отелей через интеграцию с внешними API поставщиков (эмуляция). Реализовать систему кэширования данных для ускорения поиска туров	

по сложным критериям (страна, дата, бюджет). Настроить автоматическое резервное копирование базы данных заказов и отправку уведомлений клиентам по email/SMS. ВАРИАНТ 2 (Фамилии на «Б»)	

Тема: Библиотечный фонд Задание: Создать модуль массового импорта библиографических данных из CSV/Excel и API электронных каталогов. Реализовать систему формирования отчётов о должниках с экспортом в PDF/XML. Провести нагрузочное тестирование модуля поиска книг в каталоге свыше 50 000 записей, оптимизировать индексы БД. ВАРИАНТ 3 (Фамилии на «В») Тема: Ветеринарная клиника Задание: Интегрировать систему с внешним сервисом онлайн-записи через REST API с аутентификацией. Реализовать систему разграничения прав доступа (RBAC) для врачей, ассистентов, администраторов. Внедрить шифрование персональных данных владельцев животных и истории болезней в БД. ВАРИАНТ 4 (Фамилии на «Г») Тема: Гостиничный комплекс Задание: Настроить интеграцию с системой онлайн-бронирования через JSON-файлы для синхронизации номерного фонда. Реализовать кэширование данных о доступности номеров. Разработать процедуры аварийного восстановления базы данных бронирований. ВАРИАНТ 5 (Фамилии на «Д», «Е», «Ё») Тема: Служба доставки еды Задание: Автоматизировать обновление меню и цен через импорт файлов от поставщиков. Реализовать REST API для мобильного приложения курьеров для обновления статусов заказов. Оптимизировать алгоритм распределения заказов между курьерами, провести нагрузочное тестирование в часы пик. ВАРИАНТ 6 (Фамилии на «Ж») Тема: Жилищно-коммунальное хозяйство (ЖКХ) Задание: Реализовать импорт показаний счетчиков из CSV-файлов (10 000+ записей). Автоматизировать формирование и печать квитанций в PDF. Настроить систему мониторинга и оповещения о критических показаниях. ВАРИАНТ 7 (Фамилии на «З») Тема: Завод (складской учет) Задание: Настроить автоматический импорт электронных накладных от поставщиков в XML-формате. Реализовать систему резервного копирования БД по расписанию с возможностью отката. Провести аудит безопасности модуля учета, устранить риски SQL-инъекций, настроить журналирование операций.	

ВАРИАНТ 8 (Фамилии на «И», «Й») Тема: Интернет-провайдер Задание: Интегрировать систему с оборудованием биллинга для загрузки данных об абонентской задолженности. Реализовать скрипт автоматического отключения услуг при достижении лимита. Разработать панель администратора для массовых операций (смена тарифа, блокировка). ВАРИАНТ 9 (Фамилии на «К») Тема: Кадровое агентство Задание: Реализовать парсинг и импорт резюме с сайтов поиска работы (CSV/JSON). Разработать модуль аналитических отчётов для работодателей (статистика по вакансиям, ТОП-навыки). Обеспечить безопасное хранение персональных данных соискателей. ВАРИАНТ 10 (Фамилии на «Л») Тема: Логистическая компания Задание: Интегрировать систему с GPS-трекерами для импорта координат и обновления статусов рейсов. Оптимизировать запросы для отчётов по истории маршрутов и расходу топлива. Настроить систему оповещения диспетчеров о критических отклонениях от графика. ВАРИАНТ 11 (Фамилии на «М», «Н») Тема: Медицинский центр (Поликлиника) Задание: Настроить безопасный обмен данными со страховыми компаниями (импорт реестров, экспорт услуг). Реализовать строгое разграничение доступа к медицинским картам по ролям. Внедрить систему аудита действий с конфиденциальными данными. ВАРИАНТ 12 (Фамилии на «О») Тема: Онлайн-школа (Курсы) Задание: Интегрировать платформу с внешними системами тестирования для импорта результатов. Реализовать автоматическую генерацию и рассылку сертификатов в PDF. Оптимизировать систему для работы с видеоконтентом и нагрузкой при 100+ одновременных подключений. ВАРИАНТ 13 (Фамилии на «П») Тема: Прокат автомобилей (Каршеринг) Задание: Настроить интеграцию с телематическим оборудованием автомобилей для обновления статусов. Реализовать	

систему автоматического начисления штрафов за нарушения условий аренды. Разработать скрипты для ежедневного резервного копирования БД договоров. ВАРИАНТ 14 (Фамилии на «Р») Тема: Ресторан Задание: Автоматизировать обновление меню и технологических карт через загрузку файлов от шеф-повара. Реализовать ежедневный экспорт данных о продажах в Excel для анализа. Интегрировать систему с фискальным регистратором (эмуляция). ВАРИАНТ 15 (Фамилии на «С») Тема: Страховая компания Задание: Реализовать безопасный импорт базы клиентов из CRM. Внедрить шифрование конфиденциальных полей (паспортные данные, суммы выплат). Разработать модуль формирования регламентной отчётности для контролирующих органов. ВАРИАНТ 16 (Фамилии на «Т») Тема: Таксопарк Задание: Настроить автоматический импорт данных о выполненных заказах и сменах водителей. Провести оптимизацию запросов для поиска ближайшего автомобиля и построения истории поездок. Реализовать систему мониторинга работоспособности диспетчерского ядра. ВАРИАНТ 17 (Фамилии на «У») Тема: Университет (Деканат) Задание: Реализовать импорт списков абитуриентов и результатов ЕГЭ из файлов. Автоматизировать формирование и рассылку ведомостей успеваемости преподавателям. Настроить систему резервного копирования учебных планов и данных студентов. ВАРИАНТ 18 (Фамилии на «Ф») Тема: Фитнес-клуб Задание: Интегрировать систему с турникетами для автоматической фиксации посещений. Реализовать автоматическую проверку и блокировку просроченных абонементов с уведомлением клиентов. Разработать модуль управления расписанием групповых занятий. ВАРИАНТ 19 (Фамилии на «Х») Тема: Химчистка	

Задание: Реализовать автоматический импорт обновлений прайс-листа услуг из файлов поставщика. Настроить систему SMS-информирования клиентов о готовности заказа. Оптимизировать процесс формирования отчётов о ежедневной выручке. ВАРИАНТ 20 (Фамилии на «Ц», «Ч») Тема: Частное охранное предприятие Задание: Настроить автоматический импорт графиков дежурств из Excel. Реализовать защищённое журналирование всех событий (тревоги, смены, выезды) без возможности удаления. Разработать систему формирования отчётов по инцидентам для заказчика. ВАРИАНТ 21 (Фамилии на «Ш», «Щ») Тема: Школа искусств Задание: Реализовать синхронизацию расписания занятий с календарём преподавателей. Автоматизировать учёт оплат и формирование списка должников с экспортом для бухгалтерии. Внедрить систему управления инвентарём (инструменты, материалы) с отслеживанием состояния. ВАРИАНТ 22 (Фамилии на «Ы», «Э») Тема: Электроника (Интернет-магазин) Задание: Настроить автоматическое обновление остатков и цен через импорт CSV-файлов от поставщиков. Интегрировать модуль корзины с платежным шлюзом (эмуляция API). Провести нагрузочное тестирование каталога товаров, оптимизировать кэширование. ВАРИАНТ 23 (Фамилии на «Ю») Тема: Ювелирный салон Задание: Реализовать загрузку каталога новых изделий с характеристиками из файлов производителя. Внедрить систему сквозного аудита всех операций с ТМЦ (приёмка, продажа, инвентаризация). Настроить разграничение прав доступа к финансовым отчётам. ВАРИАНТ 24 (Фамилии на «Я») Тема: Ярмарка (Выставка) Задание: Реализовать массовую регистрацию участников через загрузку списка компаний из Excel. Автоматизировать генерацию персональных QR-кодов для пропусков. Разработать модуль управления арендой оборудования и формирования итоговых счетов.	

Экзамен по модулю	**6**
Всего:	**646**

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

3.1. Для реализации программы профессионального модуля должны быть предусмотрены следующие специальные помещения

Реализация профессионального модуля происходит в кабинете, который предназначен для проведения теоретических и практических занятий по междисциплинарному курсу.

Оборудование учебного кабинета: рабочие места обучающихся; рабочее место преподавателя; технические средства обучения: мультимедийный комплекс (проектор, экран), компьютер, колонки; оборудование: камера видеонаблюдения; доска меловая.

Помещение для самостоятельной и воспитательной работы:

столы, стулья, ПК обучающихся, программное обеспечение: 1С Предприятие, OnlyOffice (AGPLv3), GIMP (GNU GPL 3), Krita (GNU GPL 3), Inkscape (GNU GPL 3), SciLab (CeCILL), Kaspersky Endpoint Security, Агент администрирования Kaspersky Security Center, Mozilla Firefox (MPL 2), Apache NetBeans IDE (Apache License 2), Notepadqq (GNU GPL 3), VSCodium (MIT), Mono Development IDE (MIT), PyCharm Community Edition (Apache License 2), Android Studio (Apache License 2), Python (Python Software Foundation License), Java (GNU GPL), Node.js (MIT), Git (GNU GPL 2), GitHub Desktop (No Copyright), UnityHub (Unity Companion License), FreeCAD (LGPL-2.0-or-later), Sweet Home 3D (GNU GPL 2+), draw.io (Apache License 2), DBeaver (Apache License 2), PostgreSQL (License (free and open-source)), MariaDB (GNU GPL), PSPP (GNU GPL 3), Wireshark (GNU GPL 2+), Imunes (Creative Commons Attribution 4.0 International Public License), VirtualBox (GNU GPL 2), Apache (Apache License 2), Консультант+ (Коммерческая лицензия), OBS (GNU GPL).

3.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации должен иметь электронные образовательные и информационные ресурсы, рекомендуемых для использования в образовательном процессе.

Основная учебная литература:

1. Шестернинов, Е. Е. Индивидуальный проект. Шаг в профессию. Базовый уровень. Практикум: учебное пособие, разработанное в комплекте с учебником для образовательных организаций, реализующих образовательные программы среднего профессионального образования / Е. Е. Шестернинов. – 2-е изд., стер. - Москва: Просвещение, 2025. - 82 с. – (Учебник СПО). - ISBN 978-5-09-124912-5. - Текст: электронный. - URL: https://znanium.ru/catalog/product/2202038

2. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие / Г.Н. Федорова. — Москва: КУРС ИНФРА-М, 2026. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст: электронный. - URL: https://znanium.ru/catalog/product/2213137

3. Гагарина, Л. Г. Разработка и эксплуатация автоматизированных информационных систем учебное пособие / Л.Г. Гагарина, Ю.С. Шевнина. — 2-е изд., перераб. и доп. — Москва ИНФРА-М, 2025. — 358 с. — (Среднее профессиональное образование). — DOI 10.12737/1985727. - ISBN 978-5-16-018360-2. - Текст: электронный. - URL: https://znanium.ru/catalog/product/1985727

4. Гагарина, Л. Г. Проектирование и архитектура программных систем: учебное пособие / Л.Г. Гагарина, А.Р. Федоров, П.А. Федоров. — 2-е изд., испр. и доп. — Москва: ИНФРА-М, 2025. — 334 с. — (Среднее профессиональное образование). - ISBN 978-5-16-020565-6. - Текст: электронный. - URL: https://znanium.ru/catalog/product/2181823

5. Щербак, А. В. Поддержка и тестирование программных модулей : учебник для среднего профессионального образования / А. В. Щербак. — Москва : Издательство Юрайт, 2026. — 145 с. — (Профессиональное образование). — ISBN 978-5-534-19290-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/590253

Дополнительная учебная литература:

1. Трофимов, В. В. Алгоритмизация и программирование: учебник для вузов / В. В. Трофимов, Т. А. Павловская. - 4-е изд. - Москва: Издательство Юрайт, 2025. - 108 с. - (Высшее образование). - ISBN 978-5-534-20430-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. - URL: https://urait.ru/bcode/562040

2. Черпаков, И. В. Основы программирования: учебник и практикум для среднего профессионального образования / И. В. Черпаков. - 2-е изд., перераб. и доп. - Москва: Издательство Юрайт, 2025. - 196 с. - (Профессиональное образование). - ISBN 978-5-53418760-1. - Текст: электронный // Образовательная платформа Юрайт [сайт]. - URL: https://urait.ru/bcode/561922

3. Гниденко, И. Г. Технология разработки программного обеспечения : учебник для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 248 с. — (Профессиональное образование). — ISBN 978-5-534-18131-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/585518

4. Казарин, О. В. Основы информационной безопасности: надежность и безопасность программного обеспечения : учебник для среднего профессионального образования / О. В. Казарин, И. Б. Шубинский. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 352 с. — (Профессиональное образование). — ISBN 978-5-534-19384-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/587457

5. Зараменских, Е. П. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Е. П. Зараменских. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 119 с. — (Профессиональное образование). — ISBN 978-5-534-21417-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/571330

6. Грекул, В. И. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 404 с. — (Профессиональное образование). — ISBN 978-5-534-19506-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/566739

7. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 273 с. —

(Профессиональное образование). — ISBN 978-5-534-20362-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. —
URL: https://urait.ru/bcode/584914

8. Толстобров, А. П. Архитектура ЭВМ: учебник для среднего профессионального образования / А. П. Толстобров. — 3-е изд., испр. и доп. — Москва: Издательство Юрайт, 2025. — 162 с. — (Профессиональное образование). — ISBN 978-5-534-16832-7. — Текст: электронный // Образовательная платформа Юрайт [сайт]. —
URL: https://urait.ru/bcode/566762

9. Соколова, В. В. Вычислительная техника и информационные технологии. Разработка мобильных приложений: учебник для вузов / В. В. Соколова. — Москва: Издательство Юрайт, 2025. — 160 с. — (Высшее образование). — ISBN 978-5-534-16302-5. — URL: https://urait.ru/bcode/561336

Базы данных и/или Интернет-ресурсы:

1. https://app.diagrams.net/ – Инструмент для диаграмм (IDEF0, DFD, UML)
2. https://www.lucidchart.com/pages – Онлайн-сервис для бизнес-диаграмм и прототипирования
3. https://plantuml.com/ru/ – Текстовый язык для создания UML-диаграмм
4. https://miro.com/ru/ – Онлайн-доска для командной работы и моделирования процессов
5. https://github.com/ – Платформа для хостинга кода и совместной работы
6. https://replit.com/ – Онлайн-среда разработки для многих языков
7. https://www.postman.com/ – Инструмент для тестирования API
8. https://junit.org/ – Фреймворк для модульного тестирования
9. https://kubernetes.io/docs/home/ - Документация по Kubernetes
10. https://docs.docker.com/ - Официальная документация по Docker
11. https://www.wolframalpha.com/ - Вычислительный движок и база знаний
12. https://www.scilab.org/ - Свободная среда для численных вычислений
13. https://letsencrypt.org/ - SSL-сертификаты
14. https://www.ssllabs.com/ - Тестирование SSL-конфигураций

Официальные, справочно-библиографические и периодические издания:

1. Информационные технологии и вычислительные системы: научный журнал / гл. ред. Попков Ю.С.; учред. ФГУ «Федеральный исследовательский центр «Информатика и управление» РАН». - М., 2026
2. Экономика и математические методы: журнал / гл. ред. В.Л. Макаров / Российская
Академия наук, Центральный экономико-математический институт, Институт проблем рынка - Российская академия наук, Москва, 2026
3. Экономический журнал Высшей школы экономики / Higher School of Economics Economic Journal: журнал / гл. ред. Е.Е. Гавриленков / Национальный исследовательский университет «Высшая школа экономики» - Издательский дом Высшей школы экономики, Москва, 2026

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Код ПК, ОК	Критерии оценки результата (показатели освоенности компетенций)	Формы контроля и методы оценки
ОК 01.	распознает задачу и/или проблему в профессиональном и/или социальном контексте; анализирует задачу и/или проблему; определяет этапы решения задачи; выявляет и эффективно находит информацию, необходимую для решения задачи и/или проблемы; составляет план действия; определяет необходимые ресурсы; оценивает результат и последствия своих действий (самостоятельно или с помощью наставника)	Контрольные работы, зачеты, квалификационные испытания, защита курсовых и дипломных проектов (работ), учебная и производственная практики, экзамены. Интерпретация результатов и выполнения практических лабораторных заданий, оценка решения го ситуационных задач оценка тестово контроля, за результатов наблюдений в деятельностью и обучающегося процессе учебной производственной практики.
ОК 02.	определяет задачи для поиска информации; определяет необходимые источники информации; планирует процесс поиска; структурирует полученную информацию; выделяет наиболее значимое в перечне информации; оценивает практическую значимость результатов поиска; оформляет результаты поиска	
ОК 03.	определяет актуальность нормативно-правовой документации в профессиональной деятельности; применяет современную научную профессиональную терминологию; определяет и выстраивает траектории профессионального развития и самообразования	
ОК 04.	организовывает работу коллектива и команды; взаимодействует с коллегами, руководством, клиентами в ходе профессиональной деятельности	
ОК 05.	излагает свои мысли и оформляет документы по профессиональной тематике на государственном языке, проявляет толерантность в рабочем коллективе	
ОК 09.	понимает общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимает тексты на базовые профессиональные темы; участвует в диалогах на знакомые общие и профессиональные темы; пишет простые связные сообщения на знакомые или интересующие профессиональные темы	

ПК 2.1.	проектирует модули, соответствующие бизнесзадачам; создает архитектурные диаграммы и документацию; определяет структуру и интерфейсы модулей; анализирует требования к модулю и определять его функциональность; проектирует архитектуру модуля, включая выбор подходящих паттернов проектирования и структуры данных; создает диаграммы классов, последовательностей и прочих диаграмм для визуализации проектируемого модуля; выбирает подходящие языки программирования и технологии для реализации модуля; проектирует интерфейсы программного обеспечения для взаимодействия с другими модулями и системами; учитывает требования к масштабируемости, производительности и безопасности при проектировании модуля; проводит анализ и оптимизацию проектируемого модуля для повышения его эффективности и качества
ПК 2.2.	разрабатывает модули программного обеспечения с использованием различных языков программирования и технологий; применяет паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей; анализирует требования и определять функциональность модуля; создает интерфейсы для взаимодействия с другими модулями и системами; обеспечивает безопасность, производительность и масштабируемость при разработке модулей; оптимизирует проектируемые модули для повышения их эффективности и качества; работает с системой контроля версий; улучшает производительность модулей, выявляя и устраняя узкие места; проводит анализ и мониторинг производительности приложений; применяет инструменты для рефакторинга и оптимизации программного кода.
ПК 2.3.	интегрирует модули и компоненты, обеспечивая их взаимодействие; работает с API и устанавливает соединения между компонентами; отслеживает и устраняет конфликты и ошибки интеграции; анализирует и определяет зависимости между модулями и компонентами; работает с различными форматами данных и протоколами передачи данных

ПК 2.4.	анализирует требования к программному обеспечению и составляет планы тестирования; создает тестовые сценарии и тест-кейсы для проверки функциональности и соответствия требованиям; выполняет тестирование программного обеспечения вручную и автоматизирует процесс тестирования; анализирует результаты тестирования и документирует найденные ошибки; разрабатывает стратегии отладки и исправляет ошибки в программном обеспечении; выполняет модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования; использует системы контроля дефектов ПО; составляет отчет о выполнении тестирования ПО	
ПК 2.5.	описывает функциональность модулей в документации; создает диаграммы для иллюстрации работы модулей; программирует с использованием комментариев для документирования кода; использует специальные метки/теги для отметки важных частей кода в документации; ведет журнал изменений и фиксировать обновления программных модулей; разбивает модули на логические блоки и описывать каждый блок отдельно; включает в документацию особенности модулей, такие как ограничения, уязвимости или оптимальные настройки; проводит регулярное обновление документации при изменении модулей или добавлении нового функционала.	

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения по учебной дисциплине

Темы для подготовки к промежуточной аттестации

МДК 02.01. Разработка программных модулей

1. Принципы проектирования программных модулей.
2. Методы анализа требований для определения функциональности модуля.
3. Принципы высококачественных модулей: сильная связность, слабое зацепление, информационное скрытие.
4. Методы декомпозиции сложной задачи на подзадачи и модули.
5. Определение ответственности и границ модуля.
6. Разработка спецификаций модуля: формальные и неформальные подходы.
7. Проектирование интерфейсов модуля (API) и контрактов.
8. Использование предусловий и постусловий в спецификациях.
9. Оптимизация взаимодействия между модулями для сопровождаемости, тестируемости и переиспользования.
10. Применение диаграмм компонентов (UML) для визуализации модульной структуры.
11. Основные принципы ООП: инкапсуляция, наследование, полиморфизм, абстракция.
12. Практическое применение принципов ООП в проектировании модулей.
13. Базовые отношения между классами: ассоциация, агрегация, композиция.
14. Критерии выбора между отношениями «has-a» (агрегация/композиция) и «is-a» (наследование).
15. Абстрактные классы и интерфейсы: сравнительный анализ и применение.
16. Контрактно-ориентированное программирование.
17. Принципы проектирования SOLID.
18. Применение диаграмм классов UML для визуализации и проектирования.
19. Модульная архитектура: принципы, преимущества и ограничения.
20. Примеры архитектурных стилей: монолитные и микросервисные приложения.
21. Системы контроля версий: принципы работы Git и branching-стратегии.
22. Системы сборки и менеджеры зависимостей: Maven, Gradle, NuGet.
23. Использование сторонних библиотек: оценка, внедрение, управление версиями.
24. Создание и публикация собственных библиотек: сборка, версионирование, распространение.
25. Архитектурные шаблоны высокого уровня: MVC, MVP, MVVM (концептуальное сравнение).
26. Классификация паттернов проектирования: порождающие, структурные, поведенческие.
27. Взаимосвязь паттернов проектирования с отношениями между классами (наследование, композиция и др.).
28. Модели асинхронного программирования: Callback, Promises/Futures, async/await.
29. Управление потоками: пулы потоков (Thread Pool) и задачи (Task).

30. Механизмы отмены асинхронных и длительных операций.
31. Синхронизация доступа к общим ресурсам: блокировки, семафоры, мьютексы.
32. Проблемы параллельного программирования: состояние гонки (race condition) и взаимная блокировка (deadlock).
33. Паттерны параллельного программирования: Producer-Consumer, Fork-Join и другие.
34. Применение лямбда-выражений и функциональных интерфейсов в асинхронных операциях.
35. Оценка сложности алгоритмов: понятие асимптотической оценки.
36. Большие О-нотации: анализ временной и пространственной сложности.
37. Анализ алгоритмов: худший, лучший и средний случаи.
38. Амортизационный анализ сложности операций.
39. Базовые структуры данных: массивы (статические и динамические).
40. Линейные структуры данных: связные списки, стек, очередь, дек.
41. Алгоритмы поиска: линейный и бинарный поиск, их применение.
42. Алгоритмы сортировки: от простых (пузырьковая, вставками) до эффективных (быстрая, слиянием).
43. Хеш-таблицы: принцип работы, хеш-функции, разрешение коллизий.
44. Древовидные структуры: бинарные деревья, деревья поиска, сбалансированные деревья (AVL, красно-черные).
45. Графы: представление (матрица, список смежности), алгоритмы обхода (DFS, BFS). 46. Расширенные алгоритмы на графах: поиск кратчайшего пути (Дейкстра, Флойд 47. Алгоритмы построения минимального остовного дерева: Крускала и Прима.
48. Методы разработки алгоритмов: "разделяй и властвуй", жадные алгоритмы, динамическое программирование.
49. Алгоритмы работы со строками: поиск подстроки (Кнута-Морриса-Пратта, БойераМура).
50. Специализированные структуры данных: B-деревья, B+-деревья, бинарные кучи и приоритетные очереди.

МДК.02.02 Поддержка и тестирование программных модулей

1. Внутреннее и внешнее качество ПО: сравнительный анализ.
2. Стандарт ISO/IEC 25010: Модель качества программного обеспечения.
3. Практическое применение и интерпретация метрик цикломатической сложности.
4. Оценка уровня связности модуля: от функциональной до случайной.
5. Оценка уровня зацепления модулей: от зацепления по данным до зацепления по содержимому.
6. Динамические метрики и их связь с надежностью и производительностью системы.
7. Принцип единственной ответственности как основа высокой связности.
8. Принцип инверсии зависимостей как основа слабого зацепления.
9. Инструменты оценки качества: от линтеров до платформ типа SonarQube.
10. Паттерны рефакторинга как средство устранения «запахов кода».
11. Профилирование как метод поиска «узких мест» в модуле.
12. Классификация дефектов по степени критичности и приоритету исправления.
13. Отладка как научный метод: гипотеза, эксперимент, анализ.

14. Современные возможности IDE для отладки (условные точки, трассировка).
15. Поиск и локализация «плавающих» дефектов.
16. Стратегии локализации ошибок: бинарный поиск, анализ симптомов.
17. Работа с системами отслеживания ошибок: жизненный цикл дефекта.
18. Процесс управления инцидентами и проблемами.
19. Архитектурные подходы к обработке ошибок и исключений.
20. Построение иерархии пользовательских исключений.
21. Шаблоны обработки исключений: специфичный перехват, проброс, трансформация.
22. Логирование в отладке: выбор уровня детализации и точек вставки логов.
23. Корреляция логов и трассировок для анализа распределенных сбоев.
24. Влияние механизма исключений на производительность.
25. Best practices документирования и воспроизведения дефектов.
26. Планирование тестирования: оценка усилий, определение стратегии.
27. Модульное тестирование: использование mock-объектов и заглушек.
28. Подходы к интеграционному тестированию: Big Bang, incremental, sandwich.
29. Системное тестирование: проверка нефункциональных требований.
30. Приемочное тестирование пользователем (UAT) и операционное (OAT).
31. Техники тестирования черного ящика: таблицы решений, анализ причинноследственных связей.
32. Критерии покрытия кода в тестировании белого ящика.
33. Проектирование тестов для нефункциональных требований.
34. Инструменты для тестирования безопасности (OWASP ZAP, Burp Suite).
35. Методологии нагрузочного тестирования: планирование, выполнение, анализ.
36. Тестирование совместимости на различных платформах.
37. Экономическое обоснование автоматизации тестирования (ROI).
38. Паттерны проектирования автоматизированных тестов (Page Object, Screenplay).
39. Интеграция тестов в CI/CD: пайплайны, gate, отчетность.
40. Пирамида тестирования как оптимальная стратегия автоматизации.
41. Методы разработки тестовых случаев: таблицы решений, диаграммы переходов состояний.
42. Анализ рисков как основа для определения приоритетов тестирования.
43. Инструменты для тестирования API (Postman, REST Assured, SoapUI).
44. Стратегии тестирования контейнеризованных и serverless-приложений.
45. Особенности тестирования в Agile и DevOps-методологиях.
46. Виды работ по сопровождению ПО: корректирующее, адаптивное, совершенствующее, профилактическое.
47. Ревью кода: формальные и неформальные подходы, инструменты (GitHub PR, Gerrit).
48. Стандарты внутренней документации: соглашения об именовании, комментарии, архитектурные решения (ADR).
49. Создание и поддержание тестовой документации: тест-план, тест-кейс, чек-лист, отчет.
50. Современные подходы к документации: living documentation, doc-as-code, автоматическая генерация.

МДК.02.03 Технология разработки программных модулей

1. Системы статического анализа кода: принцип работы, цели и примеры инструментов.
2. Интеграция статических анализаторов в процесс разработки: проверка стиля, качества, уязвимостей и соблюдения принципов.
3. Инструменты UML для проектирования: классификация (визуальные редакторы, генераторы кода) и примеры.
4. Современные среды разработки как платформы для анализа: встроенные рефакторинги, визуализаторы зависимостей, подсказки по сложности.
5. Интерактивные среды для анализа алгоритмов: назначение, возможности для обучения и отладки.
6. Библиотеки для работы со сложными структурами данных: обзор и выбор для конкретных задач.
7. Профайлеры для оценки алгоритмической сложности и производительности в runtime: принцип профилирования, типы профайлеров (инструментирующие, семплирующие).
8. ORM-фреймворки и микро-ORM: сравнительный анализ подходов, примеры.
9. Инструменты для построения, анализа и оптимизации SQL-запросов: встроенные в IDE, независимые клиенты, анализаторы планов выполнения.
10. Системы управления миграциями базы данных: принцип версионирования схемы, обзор инструментов.
11. Профайлеры запросов к БД и APM-системы: мониторинг производительности на уровне запросов и транзакций.
12. Обзор GUI-фреймворков для создания декстопных и веб-приложений: критерии выбора, семейства.
13. Концепция привязки данных и шаблон MVVM: архитектурные преимущества, реализация в различных фреймворках.
14. Средства стилизации UI: от CSS-препроцессоров и CSS-in-JS для веба до систем ресурсов и стилей в нативных фреймворках.
15. Технологии создания API для межмодульного взаимодействия: сравнительный анализ REST, gRPC, GraphQL по критериям производительности, типизации и гибкости.
16. Инструменты для автоматического документирования API: подходы на основе кода и аннотаций.
17. Клиенты для тестирования API на этапе разработки и интеграции: функционал и использование.
18. Фреймворки для модульного тестирования GUI: подходы к изоляции и симуляции интерфейса.
19. Инструменты для автоматизации End-to-End (E2E) сценариев для GUI: классификация.
20. Методы тестирования асинхронных и событийно-ориентированных интерфейсов: моки таймеров, симуляторы событий, тестирование с ожиданиями.
21. Статические анализаторы безопасности кода: интеграция в CI/CD для поиска уязвимостей.
22. Принципы безопасного кодирования для модулей: OWASP рекомендации, защита от инъекций, работа с чувствительными данными.

23. Системы управления зависимостями и сборкой для разных языков: общая логика работы.
24. Стратегии управления версиями модулей и библиотек: семантическое версионирование (SemVer), совместимость, dependency hell.
25. Инструменты контейнеризации для изоляции среды выполнения модуля: Docker как стандарт, создание и использование Docker-образов.
26. Принципы CI/CD применительно к модульной разработке: цели непрерывной интеграции и доставки для независимых модулей.
27. Обзор платформ для реализации CI/CD пайплайнов: облачные и self-hosted решения.
28. Этапы типового CI/CD пайплайна для модуля: от коммита к артефакту.
29. Публикация артефактов: использование артефакт-репозиториев.
30. Мониторинг и observability модулей в production: APM-системы, сбор метрик, логирование и распределенная трассировка.

Темы для самостоятельной работы обучающихся

МДК 02.01. Разработка программных модулей

1. Архитектурные подходы к доступу к данным: Data Access Layer (DAL).
2. Паттерн Repository: абстракция над хранилищем данных.
3. Паттерн Unit of Work: управление транзакциями и целостностью.
4. Сравнение подходов Active Record и Data Mapper.
5. Архитектурный паттерн CQRS (Command Query Responsibility Segregation).
6. Технологии доступа к данным: ADO.NET, Entity Framework Core, Dapper, другие ORM.
7. Управление подключениями к БД: connection strings, пулы соединений, мониторинг.
8. Конфигурация ORM: маппинг сущностей (Fluent API, атрибуты).
9. Реализация CRUD-операций и выполнение запросов.
10. Управление транзакциями: уровни изоляции, распределенные транзакции.
11. Стратегии управления конкурентным доступом: оптимистическая и пессимистическая блокировка.
12. Оптимизация работы с БД: проблема N+1 и методы ее решения.
13. Индексы в базах данных: создание, анализ, покрывающие индексы.
14. Анализ планов выполнения запросов (Query Execution Plans).
15. Кэширование данных: кэширование запросов первого и второго уровня.
16. Принципы безопасного программирования для модулей данных.
17. Безопасность на уровне БД: принцип наименьших привилегий, шифрование данных.
18. Обеспечение конфиденциальности: маскирование данных в логах и отладочной информации.
19. Аудит изменений данных: механизмы отслеживания «кто, когда, что изменил».
20. Управление миграциями базы данных: инструменты и стратегии деплоя.
21. Виды пользовательских интерфейсов: командная строка, графический, речевой.
22. Принципы и этапы разработки графического пользовательского интерфейса (GUI).
23. Технологии и инструменты для разработки GUI.
24. Компоненты GUI: типы элементов управления, компоновка (layout).
25. Событийно-ориентированное программирование в GUI: события и обработчики.

26. Работа с окнами: создание, управление жизненным циклом, взаимодействие между окнами.
27. Валидация пользовательского ввода: методы, регулярные выражения, обратная связь.
28. Многопоточность в GUI-приложениях: проблемы и паттерны (например, BackgroundWorker).
29. Основы UX/UI дизайна: теория цвета, работа со шрифтами, стилизация.
30. Визуализация данных в GUI: построение графиков и диаграмм с использованием специализированных библиотек.
31. Основные нефункциональные требования: безопасность, производительность, масштабируемость.
32. Принципы безопасного программирования на уровне модулей.
33. Криптографические примитивы и их применение в модулях.
34. Методология оптимизации производительности: профилирование, анализ узких мест.
35. Алгоритмическая оптимизация: выбор структур данных, учет кэш-локальности.
36. Оптимизационные техники: мемоизация, кэширование, ленивые вычисления.
37. Оптимизация операций ввода-вывода и сетевого взаимодействия.
38. Принципы проектирования масштабируемых систем: горизонтальное и вертикальное масштабирование.
39. Паттерны масштабируемости: шардирование, репликация, кэширование.
40. Принципы построения отказоустойчивых систем: idempotency, retry, circuit breaker.
41. Системы мониторинга и метрики: Prometheus, Grafana, бизнес-метрики.
42. Централизованное логирование: стеки ELK, Loki, анализ логов.
43. Распределенная трассировка (Distributed Tracing): OpenTelemetry, Jaeger.
44. Проактивный мониторинг и алертинг.
45. Принципы энергоэффективного («зеленого») программирования.
46. Управление затратами в облачных средах: оптимизация использования ресурсов.
47. Современные облачные практики: serverless, containers, infrastructure as code.
48. Стратегии развертывания: blue-green deployments, canary releases, zero-downtime миграции.
49. Наблюдаемость как ключевая характеристика современных распределенных систем.
50. Управление конфигурацией и секретами в облачных и гибридных средах.

МДК.02.02 Поддержка и тестирование программных модулей

1. Определение и сущность качества программного модуля.
2. Международные стандарты качества программного обеспечения (ISO 25010 и др.).
3. Статические метрики объема кода: количество строк кода и его ограничения.
4. Метрика цикломатической сложности Маккейба.
5. Метрики связности и их классификация.
6. Метрики зацепления и их классификация.
7. Динамические метрики качества: покрытие кода тестами, частота отказов, время отклика.
8. Принципы проектирования, влияющие на качество модулей (SOLID, DRY, KISS).
9. Инструменты статического анализа кода (линтеры) и их роль.
10. Понятие «запахов кода» и их связь с качеством.

11. Роль рефакторинга в поддержании высокого качества кодовой базы.
12. Инструменты профилирования для анализа производительности модуля.
13. Разграничение понятий: ошибка, дефект, сбой.
14. Основные типы программных ошибок: синтаксические, логические, ошибки времени выполнения.
15. Процесс и философия отладки программного обеспечения.
16. Функциональные возможности современных инструментов отладки (дебаггеров).
17. Техники пошаговой отладки: точки останова, шаг за шагом, просмотр состояния.
18. Стратегии поиска ошибок: «разделяй и властвуй», обратное прослеживание.
19. Системы отслеживания ошибок (Bug Tracking Systems): назначение и базовые функции.
20. Документирование процесса отладки и отчетности об ошибках.
21. Роль и классификация исключительных ситуаций в программировании.
22. Различия между проверяемыми (checked) и непроверяемыми (unchecked) исключениями.
23. Синтаксические конструкции для управления исключениями (try-catch-finally, throw/throws).
24. Стратегии работы с исключениями: перехват, проброс, трансляция, логирование.
25. Использование логирования (logging) как инструмента отладки и анализа.
26. Цели и основные этапы процесса тестирования программного обеспечения.
27. Сравнение ручного и автоматизированного тестирования.
28. Модульное тестирование: цели и объекты тестирования.
29. Интеграционное тестирование: цели и подходы.
30. Системное тестирование и его место в жизненном цикле.
31. Приемочное тестирование и его формы.
32. Метод тестирования черного ящика (Black-box Testing): принципы и техники.
33. Метод тестирования белого ящика (White-box Testing): принципы и критерии покрытия.
34. Виды нефункционального тестирования: производительность, безопасность, удобство использования.
35. Тестирование производительности: основные виды.
36. Тестирование удобства использования и его критерии.
37. Нагрузочное и стресс-тестирование: сравнительная характеристика.
38. Преимущества и недостатки автоматизации тестирования.
39. Инструменты для автоматизации функционального тестирования (Selenium, TestComplete).
40. Интеграция автоматизированных тестов в процесс разработки (CI/CD).
41. Методы разработки тестовых случаев: анализ граничных значений, эквивалентное разделение.
42. Критерии покрытия тестами: операторы, ветви, условия, пути.
43. Выбор инструментов для разработки и выполнения тестов.
44. Особенности тестирования мобильных и облачных приложений.
45. Тестирование микросервисной архитектуры: специфические задачи и подходы.
46. Работы, выполняемые на этапе сопровождения программного обеспечения.
47. Процесс ревьюирования кода как практика повышения качества.
48. Стандарты разработки и оформления документации на ПО.
49. Виды тестовой документации: планы, спецификации, сценарии, отчеты.

50. Инструменты для создания технической документации и комментирования кода.

МДК.02.03 Технология разработки программных модулей

1. Настройка и кастомизация правил статического анализатора кода для проверки принципов SOLID и code smells.
2. Автоматическая генерация UML-диаграмм (классов, последовательности) из существующего кода и реверс-инжиниринг.
3. Сравнение возможностей рефакторинга в различных IDE и их влияние на производительность.
4. Анализ алгоритмов с помощью Big O на практике: использование профайлера для верификации теоретической оценки.
5. Реализация нестандартных структур данных с использованием базовых библиотек языка.
6. Сравнение производительности микро-ORM и нативного SQL в высоконагруженных сценариях: написание и профилирование бенчмарков.
7. Продвинутые техники оптимизации SQL-запросов с использованием инструментов анализа планов выполнении.
8. Разработка и отладка сложных миграций БД с поддержкой отката в распределенной системе.
9. Выбор GUI-фреймворка для кроссплатформенного десктопного приложения: сравнительный анализ Qt, .NET MAUI, Electron.
10. Реализация паттерна MVVM «с нуля» и сравнение со встроенной поддержкой во фреймворках.
11. Создание системы дизайн-токенов и тем оформления для переиспользуемых UIкомпонентов.
12. Проектирование API-шлюза для агрегации данных из нескольких микросервисов с использованием GraphQL.
13. Автоматизация поддержки актуальности Swagger-документации на основе аннотаций в коде.
14. Разработка коллекции автотестов для REST API и их интеграция в CI.
15. Тестирование компонентов с использованием библиотек: подходы к мокированию и проверке состояния.
16. Написание устойчивых E2E-тестов с использованием паттерна Page Object.
17. Стратегии тестирования WebSocket-соединений и real-time обновлений интерфейса.
18. Интеграция визуального регрессионного тестирования в пайплайн для UI-модулей.
19. Проведение аудита безопасности зависимостей проекта помощью OWASP DependencyCheck или Snyk.
20. Реализация модуля с соблюдением принципов безопасности: хеширование паролей, безопасные заголовки HTTP, валидация входных данных.
21. Разрешение конфликтов версий зависимостей в крупном монорепозитории с множеством модулей.
22. Создание и публикация пакета (библиотеки) в публичный реестр с соблюдением SemVer.

23. Оптимизация Docker-образа для production-модуля: многоэтапная сборка, уменьшение размера, безопасность.
24. Проектирование отказоустойчивого CI/CD пайплайна для модуля с этапами кэширования зависимостей и параллельным выполнением тестов.
25. Автоматизация развертывания (CD) модуля в облачную среду из артефактрепозитория.
26. Сбор и анализ пользовательских метрик производительности модуля в Prometheus.
27. Настройка алертинга на основе метрик и логов для критически важных модулей системы.
28. Практика «Infrastructure as Code» для модуля: описание среды развертывания.
29. Реализация стратегии Blue-Green Deployment или Canary Release для модуля с минимальным временем простоя.
30. Организация мониторинга технического долга модуля через метрики качества кода и регулярный аудит.

Примерные варианты тестовых оценочных заданий (ТОЗ) для контрольного рубежа в рамках текущего контроля

МДК 02.01. Разработка программных модулей

Тестовое задание	Оцениваемые компетенции
ВОПРОСЫ ОТКРЫТОГО ТИПА	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Степень взаимозависимости и связанности элементов внутри одного модуля называется _____. +: связность	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Мера зависимости одного модуля от внутреннего устройства других модулей называется _____. +: зацепление	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Принцип, при котором детали реализации модуля скрыты от внешнего мира и доступен только четко определенный интерфейс, называется _____. +: инкапсуляция	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Набор из пяти принципов объектно-ориентированного проектирования, способствующих созданию понятного и гибкого кода, известен как принципы _____. +: SOLID	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Принцип, гласящий, что класс должен иметь только одну причину для изменения, называется принципом _____. +: единственной ответственности	ПК 2.1.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Паттерн проектирования, который определяет семейство алгоритмов, инкапсулирует каждый из них и делает их взаимозаменяемыми, называется _____. +: Стратегия	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Структура данных, в которой добавление и удаление элементов происходит по принципу "последним пришел – первым вышел" (LIFO), называется _____. +: стек	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Эффективная структура данных для поиска, вставки и удаления, основанная на хеш-функции, называется _____. +: хеш-таблица	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Рекурсивный метод проектирования алгоритмов, разбивающий задачу на подзадачи того же типа, называется _____. +: "разделяй и властвуй"	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Метод проектирования алгоритмов, при котором задача разбивается на перекрывающиеся подзадачи, результаты которых запоминаются и повторно используются, называется _____. +: динамическое программирование	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Паттерн слоя доступа к данным, который абстрагирует набор объектов и операции над ними, имитируя коллекцию в памяти, называется _____. +: Репозиторий	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Паттерн для группировки набора операций с базой данных в единую транзакцию, чтобы все они либо завершились успешно, либо были отменены, называется _____. +: Единица работы	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Архитектурный подход, разделяющий операции чтения данных (Query) и операции изменения данных (Command), называется _____. +: CQRS	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Технология, которая автоматически связывает объекты приложения с записями в реляционной базе данных, называется _____. +: ORM (Object-Relational Mapping)	ПК 2.2.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Механизм, при котором изменения в графическом интерфейсе автоматически отражают изменения в объекте данных, и наоборот, называется _____. +: привязка данных	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Состояние, при котором два или более потока одновременно обращаются к общим данным и как минимум один из них изменяет их, называется _____. +: состояние гонки	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Ситуация, в которой два или более потока взаимно блокируют друг друга, ожидая освобождения ресурсов, называется _____. +: взаимная блокировка	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Объект, используемый для синхронизации потоков и ограничивающий количество потоков, которые могут одновременно получить доступ к ресурсу, называется _____. +: семафор	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Принцип безопасного программирования, согласно которому пользователю или процессу должны предоставляться минимально	ПК 2.2.

необходимые привилегии, называется принципом _____. +: наименьших привилегий	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Стратегия развертывания, при которой работают два идентичных производственных окружения ("синее" и "зеленое"), а трафик переключается между ними, называется _____ развертывание. +: сине-зеленое	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Архитектурный шаблон, производный от MVC, где роль контроллера играет Посредник (Presenter), тесно связанный с Представлением, называется _____. +: MVP	ПК 2.2.
ВОПРОСЫ ЗАКРЫТОГО ТИПА	
Какой принцип SOLID гласит: "Программные сущности должны быть открыты для расширения, но закрыты для модификации"? +: Принцип открытости/закрытости -: Принцип единственной ответственности -: Принцип подстановки Барбары Лисков -: Принцип инверсии зависимостей	ПК 2.1.

Какой принцип объектно-ориентированного программирования позволяет трактовать объекты разных классов как объекты общего родительского класса? +: Полиморфизм -: Инкапсуляция -: Абстракция -: Наследование	ПК 2.2.
Какой тип связи между классами предполагает, что один объект является частью другого, но может существовать самостоятельно? +: Агрегация -: Композиция -: Наследование -: Реализация	ПК 2.2.
Какая структура данных работает по принципу "первым пришел, первым вышел" (FIFO)? -: Стек (Stack) -: Куча (Heap) +: Очередь (Queue) -: Связный список (Linked List)	ПК 2.2.
Какой архитектурный шаблон четко разделяет ответственность за данные, отображение и логику управления? -: MVP (Model-View-Presenter) -: MVVM (Model-View-ViewModel) +: MVC (Model-View-Controller) -: Сервис-ориентированная архитектура (SOA)	ПК 2.2.
Какая уязвимость возникает при выполнении непроверенных пользовательских данных в качестве части команды СУБД? -: XSS (Межсайтовый скриптинг)	ПК 2.1.
+: SQL-инъекция -: Race condition (Состояние гонки) -: Переполнение буфера	
Какой инструмент используется для описания и визуализации статической структуры классов в системе? -: Диаграмма последовательности UML -: Диаграмма развертывания UML +: Диаграмма классов UML -: Диаграмма деятельности UML	ПК 2.2.

Какое отношение между классами в UML является наиболее сильным, где часть не может существовать без целого? +: Композиция -: Агрегация -: Ассоциация -: Зависимость	ПК 2.2.
Какой паттерн слоя доступа к данным используется для группировки набора операций в одну бизнес-транзакцию, чтобы все они либо завершились успешно, либо были отменены? -: Репозиторий (Repository) -: Data Mapper +: Единица работы (Unit of Work) -: Активная запись (Active Record)	ПК 2.2.

МДК.02.02 Поддержка и тестирование программных модулей

Тестовое задание	Оцениваемые компетенции
ВОПРОСЫ ОТКРЫТОГО ТИПА	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Нормативный стандарт, определяющий модель качества программного обеспечения, известен как _____. +: ISO 25010	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Метрика, измеряющая количество линейно независимых путей в графе потока управления программы, называется _____. +: цикломатическая сложность	ПК 2.4
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Мера зависимости одного модуля от деталей реализации других модулей называется _____. +: зацепление	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Динамическая метрика, показывающая, какая часть исходного кода была выполнена во время тестов, называется _____. +: покрытие кода	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс поиска, анализа и устранения причин дефектов в программном	ПК 2.4.

обеспечении называется _____. +: отладка	

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Механизм обработки аномальных или исключительных ситуаций во время выполнения программы называется _____. +: исключение	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Результат ошибки программиста, приводящий к некорректному состоянию программы, называется _____. +: дефект	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Программная система для регистрации, отслеживания статуса и управления жизненным циклом дефектов называется _____. +: система отслеживания ошибок	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Вид тестирования, при котором тестировщик не имеет доступа к исходному коду и внутренней структуре системы, называется тестированием _____. +: черного ящика	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Вид тестирования, оценивающий удобство и простоту использования интерфейса системы для конечного пользователя, называется _____. +: тестирование удобства использования	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс проверки кода другими разработчиками на предмет соответствия стандартам и выявления дефектов называется _____. +: ревью кода	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс изменения внутренней структуры кода без изменения его внешнего поведения с целью улучшения читаемости и сопровождаемости называется _____. +: рефакторинг	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Метод разработки тестовых случаев, при котором входные данные разбиваются на классы эквивалентности, называется _____. +: эквивалентное разделение	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Техника тестирования, при которой тестовые данные выбираются на границах допустимых диапазонов, называется _____. +: анализ граничных значений	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Вид тестирования, имитирующий ожидаемую или пиковую нагрузку на систему для оценки ее поведения, называется _____. +: нагрузочное тестирование	ПК 2.4.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Критерий покрытия тестами, требующий выполнения каждой ветви (как true, так и false) каждого условия в коде, называется покрытием _____. +: условий	ПК 2.4.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Формальное описание шагов, данных и ожидаемых результатов для проверки конкретной функциональности, называется _____. +: тестовый случай	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Документ, фиксирующий обнаруженный дефект с описанием шагов для его воспроизведения, называется _____. +: отчет об ошибке	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Стратегия поиска ошибки, при которой программа исследуется от точки сбоя назад к его причине, называется _____. +: обратное прослеживание	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Принцип, согласно которому модуль должен быть открыт для расширения, но закрыт для модификации, называется принципом _____. +: открытости/закрытости	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Метод оптимизации, при котором результаты дорогостоящих вызовов функций сохраняются и переиспользуются, называется _____. +: мемоизация	ПК 2.4.
ВОПРОСЫ ЗАКРЫТОГО ТИПА	
Какой нормативный стандарт определяет модель качества программного обеспечения, включающую характеристики функциональности, надежности и сопровождаемости? -: ISO 9001 -: IEEE 829 +: ISO 25010 -: SWEBOK	ПК 2.4.
Какая метрика измеряет количество линейно независимых путей через исходный код модуля? -: Количество строк кода (SLOC) +: Цикломатическая сложность (Cyclomatic Complexity) -: Глубина наследования (Depth of Inheritance) -: Коэффициент связности (Cohesion)	ПК 2.4.

Как называется мера зависимости одного модуля от внутреннего устройства других модулей? -: Связность -: Инкапсуляция +: Зацепление -: Сложность	ПК 2.4.
Как называется процесс локализации, анализа и устранения причин дефектов в программе? -: Тестирование -: Верификация -: Валидация +: Отладка	ПК 2.4.
Как называется результат ошибки программиста, приводящий к некорректному состоянию программы? +: Дефект -: Отказ -: Инцидент -: Сбой	ПК 2.4.
Как называется механизм обработки аномальных ситуаций во время выполнения программы? -: Ассерция +: Исключение -: Код возврата -: Логирование	ПК 2.4.
Как называется вид тестирования, при котором тестировщик не имеет доступа к исходному коду системы? -: Тестирование белого ящика (White-box) +: Тестирование черного ящика (Black-box) -: Модульное тестирование (Unit Testing) -: Статическое тестирование (Static Testing)	ПК 2.4.
Как называется процесс проверки кода другими разработчиками на предмет соответствия стандартам? -: Рефакторинг -: Инспекция +: Ревью кода -: Валидация кода	ПК 2.4.

Как называется тестирование, имитирующее ожидаемую или пиковую нагрузку на систему? -: Стресс-тестирование (Stress Testing) -: Объемное тестирование (Volume Testing) +: Нагрузочное тестирование (Load Testing) -: Тестирование стабильности (Soak Testing)	ПК 2.4.

МДК.02.03 Технология разработки программных модулей

Тестовое задание	Оцениваемые компетенции
ВОПРОСЫ ОТКРЫТОГО ТИПА	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Инструмент, автоматически проверяющий стиль кода, выявляющий потенциальные ошибки и уязвимости, называется _____ анализатор кода. +: статический	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Фреймворк, который автоматически отображает объекты приложения на записи в реляционной базе данных, называется _____. +: ORM	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Архитектурный шаблон, в котором Модель Представления	ОК 01

(ViewModel) связывает Модель данных с Представлением через привязку данных, называется _____. +: MVVM	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Программный интерфейс, использующий протокол HTTP и стандартные методы (GET, POST) для взаимодействия между модулями, называется _____ API. +: REST	ОК 01
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Практика автоматической сборки и тестирования кода при каждом изменении в репозитории называется Непрерывная _____. +: интеграция	ОК 02
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Инструмент для визуального проектирования, позволяющий создавать диаграммы классов и последовательностей, называется _____ редактор. +: UML	ОК 02

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Практика описания серверной инфраструктуры (серверы, сети) с помощью конфигурационных файлов называется «Инфраструктура как _____». +: код	ОК 02
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Инструмент для отслеживания производительности приложения в реальном времени, показывающий медленные запросы и узкие места, называется _____ система. +: APM	ОК 02
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Технология, которая автоматически связывает свойства данных в коде с элементами пользовательского интерфейса, называется _____ данных. +: привязка	ОК 02
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Специализированные анализаторы, ищущие в коде известные уязвимости (например, из списка OWASP), называются _____ анализаторы безопасности. +: статические	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Принцип, при котором код, данные и конфигурация модуля упаковываются вместе в единый развертываемый образ, называется _____. +: контейнеризация	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Подход к разработке, объединяющий практики разработки и эксплуатации для повышения скорости выпуска релизов, называется _____. +: DevOps	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ)	ПК 2.2.

Инструмент для автоматического форматирования кода в соответствии с заданным стилем называется _____. +: линтер	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Протокол и система для отслеживания изменений в исходном коде называется система контроля _____. +: версий	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Программа, которая транслирует исходный код в машинный код или промежуточное представление, называется _____. +: компилятор	ПК 2.2.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс улучшения внутренней структуры кода без изменения его внешнего поведения называется _____. +: рефакторинг	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Специальная программа для пошагового выполнения кода, установки точек останова и просмотра переменных, называется _____. +: отладчик	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Язык запросов для работы с данными в реляционных базах данных называется _____. +: SQL	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Фундаментальный протокол глобальной сети, отвечающий за адресацию и маршрутизацию пакетов данных между компьютерами, называется _____. +: IP	ПК 2.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Система для хранения собранных артефактов (библиотек, Dockerобразов) называется _____ репозиторий. +: артефакт	ПК 2.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Инструмент, который выполняет написанные тестовые сценарии, имитируя действия пользователя в браузере, называется _____ для E2E-тестирования. +: Selenium	ПК 2.3.
ВОПРОСЫ ЗАКРЫТОГО ТИПА	
Какой инструмент является стандартом де-факто для автоматизации тестирования веб-интерфейсов? -: TestComplete -: JMeter -: Postman +: Selenium	ПК 2.4.
Какой архитектурный шаблон предполагает наличие Модели Представления (ViewModel) для связывания данных и интерфейса? -: Model-View-Controller (MVC) -: Model-View-Presenter (MVP) +: Model-View-ViewModel (MVVM) -: Service-Oriented Architecture (SOA)	ПК 2.4.

Какой популярный клиент используется разработчиками для ручного тестирования и документирования API? -: Swagger UI -: cURL +: Postman -: Jenkins	ПК 2.4.
Какой инструмент для контейнеризации позволяет упаковать приложение со всеми зависимостями в переносимый образ? -: Kubernetes +: Docker -: Vagrant -: Terraform	ПК 2.4.
Какой инструмент используется для визуального проектирования диаграмм классов, компонентов и последовательностей? +: UML-редактор -: Система контроля версий -: Интегрированная среда разработки -: Профайлер	ПК 2.4.
Какой тип анализа кода выполняется без запуска программы для поиска потенциальных ошибок и проблем с качеством? -: Динамический анализ (Dynamic Analysis) +: Статический анализ (Static Analysis) -: Профилирование (Profiling) -: Модульное тестирование (Unit Testing)	ПК 2.4.
Какой подход к разработке объединяет практики разработки (Dev) и эксплуатации (Ops) для ускорения выпуска релизов? -: Agile -: Scrum +: DevOps -: Waterfall	ПК 2.4.
Какой принцип ООП позволяет классу-наследнику использовать или переопределять методы и свойства родительского класса? -: Инкапсуляция -: Полиморфизм +: Наследование -: Абстракция	ПК 2.4.

Какой формат используется gRPC по умолчанию для сериализации структурированных данных? -: JSON -: XML +: Protocol Buffers (protobuf) -: YAML	ПК 2.4.

Шкала оценивания тестовых заданий

Данный вид контроля, рассчитан на выявление уровня усвоения теоретического и практического материала в рамках изучения дисциплин. Критерии оценивания теста: 1 вопрос = 1 балл.

Оценка «5»	30-27 Баллов
Оценка «4»	26-23 Баллов
Оценка «3»	22-18 Баллов
Оценка «2»	0-17 Баллов

ЛИСТ АКТУАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Методические рекомендации при подготовке к теоретическим занятиям

Основными видами аудиторной работы обучающихся являются лекции.

В ходе теоретических занятий преподаватель излагает и разъясняет основные, наиболее сложные понятия темы, а также связанные с ней теоретические и практические проблемы, дает рекомендации и указания на самостоятельную работу.

В ходе теоретических занятий необходимо вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. Задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. **Контроль ведения конспекта лекций**

1. Конспект лекций обучающегося должен вестись в отдельной тетради с количеством страниц, рекомендованном ведущим лектором. На обороте титульного листа тетрадь с конспектом лекций должна содержать подписи:

- фамилия и инициалы обучающегося;
- номер группы;
- ФИО ведущих учебный предмет преподавателей.

2. В начале каждой лекции обучающийся должен записать тему и перечень рассматриваемых вопросов, а также перечень рекомендуемой литературы (при условии, что они оговариваются лектором).

3. При записи содержания лекции следует применять минимум сокращений, желательно общеупотребимых. Обучающимся может быть разработана своя система сокращений, с выносом ее в конце конспекта для пояснения.

4. Табличный, графический, цифровой материал должен быть точно перенесен в конспект с доски (проектора) без искажений динамики зависимостей и обозначений формул, а также значений результатов расчетов.

5. Тетрадь с конспектом должна содержать полный объем прочитанного материала лекций по их количеству и полноте изложения, включая материал для самостоятельного изучения и конспектирования.

Записи должны быть сделаны аккуратным почерком, разборчиво.

Методические рекомендации при проведении практических занятий

Практические занятия необходимы для закрепления изученного теоретического материала, контроля преподавателем степени подготовленности обучающихся по изучаемым учебным предметом, дисциплине или профессиональному модулю.

Перед практическим занятием следует изучить конспект лекции и рекомендованную преподавателем литературу, обращая внимание на практическое применение теории и на методику решения типовых задач. Логическая связь теоретических и практических занятий заключается в том, что информация, полученная на теоретических занятиях, в процессе самостоятельной работы на практическом занятии осмысливается и перерабатывается, при помощи преподавателя анализируется, после чего усваивается.

Методические рекомендации обучающимся для самостоятельной работы

Самостоятельная работа - это планируемая работа обучающихся, выполняемая по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Самостоятельная работа выполняет ряд функций, среди которых особенно выделяются:

развивающая (повышение культуры умственного труда, приобщение к творческим видам деятельности, обогащение интеллектуальных способностей, обучающихся);

- ✓ ориентирующая и стимулирующая (процессу обучения придается ускорение и мотивация);
- ✓ воспитательная (формируются и развиваются профессиональные качества специалиста);
- ✓ исследовательская (новый уровень профессионально-творческого мышления);
- ✓ информационно-обучающая (учебная деятельность обучающихся на аудиторных занятиях).

Целью самостоятельных занятий является самостоятельное более глубокое изучение обучающимися отдельных вопросов курса с использованием рекомендуемой дополнительной литературы и других информационных источников.

Задачами самостоятельной работы обучающихся являются:

- ✓ систематизация и закрепление полученных теоретических знаний и практических умений, обучающихся;
- ✓ углубление и расширение теоретических знаний;
- ✓ формирование умения использовать справочную литературу;
- ✓ развитие познавательных способностей и активности обучающихся: творческой инициативы, самостоятельности, ответственности и организованности;

Внеаудиторная самостоятельная работа включает такие формы работы, как:

Индивидуальные занятия (домашние занятия):

- ✓ изучение программного материала дисциплины (работа с учебником и конспектом лекции);
- ✓ изучение рекомендуемых литературных источников;
- ✓ конспектирование источников;
- ✓ выполнение контрольных работ;
- ✓ работа со словарями и справочниками;
- ✓ использование аудио- и видеозаписи;
- ✓ работа с электронными информационными ресурсами и ресурсами Internet;
- ✓ составление плана и тезисов ответа на лекции;
- ✓ составление схем, таблиц, для систематизации учебного материала;
- ✓ выполнение тестовых заданий; ✓ решение задач;
- ✓ подготовка презентаций;
- ✓ ответы на контрольные вопросы;
- ✓ аннотирование, реферирование, рецензирование текста;
- ✓ написание эссе, тезисов, докладов, рефератов;
- ✓ составление глоссария, кроссворда по темам дисциплины; ✓ работа с компьютерными программами;
- ✓ подготовка к дифференцированному зачету.

Групповая самостоятельная работа обучающихся:

- ✓ подготовка к занятиям, проводимым с использованием активных форм обучения

✓
(круглые столы, деловые игры);

✓ анализ деловых ситуаций (мини-кейсов) и др; ✓ участие в Интернет - конференциях.

Получение консультаций для разъяснений по вопросам изучаемого учебного предмета.

Но наиболее часто при изучении учебного предмета используются следующие виды самостоятельной работы обучающихся:

✓ поиск (подбор) литературы (в том числе электронных источников информации) по заданной теме;

✓ подготовка рефератов;

подготовка докладов.

Реферат - форма письменной работы. Как правило, реферат является самостоятельным библиографическим исследованием обучающегося, носящим описательно-аналитический характер. Объем реферата может достигать 10-15 страниц. Подготовка реферата подразумевает самостоятельное изучение обучающимся нескольких литературных источников (монографий, научных статей и т.д.) по теме, не рассматриваемой подробно на лекции, систематизацию материала и краткое его изложение. Цель написания реферата - привитие навыков краткого и лаконичного представления собранных материалов и фактов в соответствии с требованиями, предъявляемыми к научным отчетам, обзорам и статьям.

Работа обучающегося над рефератом включает следующие операции:

✓ уяснение сути темы (по её названию);

✓ выявление литературных источников по данной теме;

✓ ознакомление с содержанием источников, направленным на осмысление его внешней и внутренней структуры;

✓ работа над выделением главных смысловых компонентов текста (ключевых слов и предложений);

✓ отбор наиболее важных сведений из выделенных фрагментов;

✓ составление «связок» из отобранного материала в соответствии с логикой изложения фактов;

✓ написание реферата.

✓ Реферат в структурном отношении должен включать:

✓ заголовочную часть (введение);

✓ собственно, реферативную часть (изложение основных положений по плану с соответствующими названиями и нумерацией);

✓ заключительную часть (выводы);

✓ справочную часть (список использованной литературы); ✓ оглавление (содержание).

Текст реферата должен иметь связанное, цельное построение. Содержание реферата должна пронизывать главная идея, которая объединяет все информационные элементы источников, а также приводимые факты. При этом допускается сохранение отдельных структурных элементов оригинала. Однако не допустимо простое переписывание положение литературных источников. Язык должен быть кратким, ясным, доступным.

Оформление реферата должно быть выполнено машинописным способом в соответствии со следующими требованиями: **печатный шрифт** – Times New Roman, кегль (размер) 14; **листы бумаги** – формата А4, все страницы должны быть пронумерованы, с размером полей

по 2 см; **междустрочный интервал** – полуторный; **абзац –** с отступом первой строки 1,25 см; **текст –** должен быть **выровнен по ширине** и **структурирован** по главам (разделам, параграфам), сопровождаться ссылками на источники при использовании прямых (в кавычках) цитат и фрагментов опубликованных текстов, т.е. на такое использование источников должно быть явным образом указано в сносках или за текстовом списке литературы в соответствии с требованиями ГОСТа.

Реферат оценивается исходя из следующих критериев:

- ✓ поставлена ли цель в работе;
- ✓ сумел ли обучающийся самостоятельно составить логический план к теме и реализовать его;
- ✓ каков научный уровень реферата;
- ✓ собран ли достаточный фактический материал;
- ✓ удалось ли раскрыть тему;
- ✓ показана ли связь рассматриваемой темы с современными проблемами науки и общества, со специальностью обучающегося;

каков авторский вклад в систематизацию, структурирование материала, в составлении заключения;

- ✓ достигнута ли цель работы.

Тематика рефератов приводится в рамках тем лекционных занятий. Возможно написание реферата по теме, предложенной самим обучающимся (по согласованию с преподавателем). Реферат на занятиях (научно-практических конференциях) может быть представлен в виде доклада.

Реферат должен быть выполнен за один месяц до начала экзаменационной сессии. Обучающиеся, не представившие в установленный срок реферат, либо получившие оценку «неудовлетворительно», к сдаче экзамена не допускаются.

Углубленному изучению материала по отдельным разделам учебного предмета способствует подготовка обучающимися докладов. Доклад - это самостоятельная работа, анализирующая и обобщающая публикации по заданной тематике, предполагающая выработку и обоснование собственной позиции автора в отношении рассматриваемых вопросов. Подготовка доклада - достаточно кропотливый труд. Его написанию предшествует изучение широкого круга экономических первоисточников, монографий, статей, обобщение личных наблюдений. Работа над докладом способствует развитию самостоятельного, творческого мышления, учит применять экономические знания на практике при анализе актуальных социальных и правовых проблем. Рекомендуемое время доклада - 10-12 минут.

Непременным условием закрепления теоретических и практических знаний, обучающихся является выполнение ими домашних заданий. Одним из основных видов деятельности обучающегося является самостоятельная работа, которая включает в себя изучение лекционного материала, учебников и учебных пособий, первоисточников, подготовку сообщений, выступления на групповых занятиях, выполнение заданий преподавателя.

✓

Методические рекомендации обучающимся по изучению рекомендованной литературы

Самостоятельная работа обучающихся включает в себя выполнение различного рода заданий, которые ориентированы на более глубокое усвоение материала изучаемого учебного предмета. По каждой теме учебного предмета обучающимся предлагается перечень заданий для самостоятельной работы.

К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны исполняться самостоятельно и представляться в установленный срок, а также соответствовать установленным требованиям по оформлению.

Обучающимся следует:

- выполнять все плановые задания, выдаваемые преподавателем для самостоятельного выполнения, и разбирать на консультациях неясные вопросы;

- при подготовке к экзамену параллельно прорабатывать соответствующие теоретические и практические разделы учебного предмета, фиксируя неясные моменты для их обсуждения на плановой консультации.

- Для подготовки к занятиям, текущему контролю и промежуточной аттестации обучающиеся могут воспользоваться электронной библиотекой ВУЗа, расположенной по электронному адресу https://www.iprbookshop.ru/, где они имеют возможность получить доступ к учебно-методическим материалам как библиотеки ВУЗа, так и иных электронных библиотечных систем. В свою очередь, обучающиеся могут взять на дом необходимую литературу на абонементе в библиотеке, а также воспользоваться читальным залом.

Методические рекомендации по балльно-рейтинговой системе учета и оценки достижений обучающихся

Общие положения

- Балльно-рейтинговая система учета и оценки достижений, обучающихся (далее – БРС) в Частном образовательном учреждении высшего образования «Московский университет имени А. С. Грибоедова» (далее - Университет) предназначена для решения следующих задач:

- стимулирования обучающихся к регулярной учебной и внеучебной деятельности;

- организации непрерывного контроля освоения обучающимися элементов основных профессиональных образовательных программ среднего профессионального образования;

- комплексной оценки результативности учебной и внеучебной деятельности обучающихся в течение семестра;

- создания для преподавателей, руководства факультетов и кафедр надежного инструментария для осуществления систематического и регулярного контроля за учебной и внеучебной деятельностью обучающихся.

БРС является обязательной для всех участников образовательного процесса и реализуется в электронной информационно-образовательной среде «Электронный университет» (далее – ЭИОС) для всех форм и технологий обучения.

БРС обеспечивает рейтинговый учет и оценку достижений обучающегося за семестр и включает в себя учебный (далее – Rуд) и внеучебный (далее – Rвр) рейтинги.

Учебный рейтинг обучающихся

Перечень дисциплин к изучению в семестре Вы можете увидеть в Личном кабинете в ЭИОС «Электронный Университет» (далее Личный кабинет). Чтобы попасть в Личный кабинет, Вам нужно перейти по ссылке https://impe.lms.mirapolis.ru/mira/ и ввести свой логин и пароль, выданный при поступлении в Университет.

Ваш учебный рейтинг по дисциплине будет формироваться на основе результатов выполнения следующих мероприятий:

- прохождение рубежного контроля (тестирования) по результатам изучения каждого модуля (раздела) дисциплины;
- выполнение рейтинговой работы (домашней творческой работы, расчетно-аналитического задания, реферата, контрольной работы, группового проекта) при ее наличии.

Доступ к тестовым заданиям по всем модулям изучаемых дисциплин (рубежному контролю) обеспечивается с начала семестра, однако проходить тестирование необходимо только по окончании изучения каждого модуля (раздела) дисциплины.

Рейтинговая работа должна быть выполнена и передана на проверку преподавателю не позднее, чем за 2 недели до начала экзаменационной сессии через Личный кабинет. Выполнение обучающимся рейтинговой работы допускается не более одного раза в течение установленного календарным учебным графиком периода. Представленная на проверку работа исправлению не подлежит.

При возникновении вопросов по процедуре тестирования, некорректного отображения контента, иных вопросов, связанных с технической поддержкой системы тестирования, обращайтесь в Центр поддержки.

Для прохождения тестирования Вам предоставляются две попытки с ограничением по времени (45 минут). Выполнение тестирования допускается не более 2-ух раз в течение

установленного календарным учебным графиком периода. Как итоговый учитывается Ваш лучший результат из двух попыток.

Допуск обучающегося к промежуточной аттестации осуществляется с любым количеством баллов, набранных в ходе выполнения Рейтинговых мероприятий по учебной дисциплине, в том числе с количеством баллов – 0.

Если по результатам выполнения всех рейтинговых мероприятий (тестирования по всем модулям дисциплины и выполнения рейтинговой работы) Вы набрали средний балл - 75 баллов и более, Вы можете претендовать на оценку «хорошо», «отлично» или «зачтено» по итогам работы в семестре без прохождения промежуточной аттестации (т.е. «автоматом»).

Средний балл рассчитывается как сумма баллов за все элементы рейтингового контроля (тестирование по каждому модулю, выполнение рейтинговой работы) разделенная на количество элементов контроля. При этом оценка определяется в соответствии со шкалой:

75-84 балла – хорошо, 85-100

баллов – отлично, 75-100 –

зачтено.

Если средний балл менее 75, или Вы претендуете на более высокую оценку, то Вы сдаете зачет/экзамен по дисциплине на общих основаниях.

Промежуточная аттестация **для обучающихся по очной, очно-заочной формам обучения** осуществляется в форме итогового тестирования по дисциплине.

для Промежуточная аттестация **обучающихся с применением дистанционных образовательных технологий** в форме проведения письменных и/или устных зачетов и экзаменов. выполнения

Если студент, обучающийся с применением ЭИОС, претендует на более высокую оценку, чем «автомат», то для открытия итогового тестирования необходимо обратиться в сервис «Центр поддержки».

Для выполнения итогового тестирования предоставляются также две попытки.

Обращаем Ваше внимание на то, что если в течение семестра Вы оформляли отсрочку оплаты обучения, то за две недели до начала зачетно-экзаменационной сессии финансовая задолженность должна быть погашена. В противном случае у Вас не будет возможности получить оценки по дисциплинам «автоматом» или пройти промежуточную аттестацию.

Внеучебный рейтинг обучающихся

Внеучебный рейтинг применяется для активизации обучающихся к участию в творческой, спортивной, общественной и научно-исследовательской деятельности Университета и является составной частью портфолио обучающегося, складывается из баллов, полученных обучающимися за участие в различных внеучебных мероприятиях. Подтверждение суммы баллов за внеучебный рейтинг осуществляется деканами на основе анализа представленных Вами документов (сертификатов, грамот, свидетельств, благодарностей и иных материалов) и заполненного Вами в Личном кабинете портфолио.

Ликвидация академической задолженности

Если в ходе промежуточной аттестации по учебной дисциплине Вы получили неудовлетворительную оценку или не явились на зачет/экзамен (не прошли итоговое тестирование), то у Вас образуется академическая задолженность.

Все набранные ранее в течение семестра по результатам выполнения рейтинговых мероприятий баллы обнуляются.

Ликвидировать академическую задолженность Вы должны в течение года, следующего за семестром образования академической задолженности (но не более двух раз).

Ликвидация академической задолженности проходит в форме выполнения итогового теста. Шкала оценивания при ликвидации академической задолженности:

при форме контроля по дисциплине – **экзамен/ зачет с оценкой: зачет:**

- «неудовлетворительно» – 49 баллов и менее; «не зачтено» – 49 баллов и менее; «зачтено» – от 50 до 100 баллов
- «удовлетворительно» – от 50 до 65 баллов;
- «хорошо» – от 66 до 84 баллов;
- «отлично» – от 85 до 100 баллов

при форме контроля по дисциплине –

Пройти итоговое тестирование Вы можете не более двух раз в течение установленного календарным графиком периода. Учитывается лучший результат, полученный Вами в ходе выполнения итогового тестирования. Для выполнения одного сеанса тестирования устанавливается время – один академический час (45 минут).