



УТВЕРЖДАЮ
Заместитель директора Ур СЭИ (филиал)
ОУП ВО «АТиСО» по УВР

_____ О.В. Зубкова

« 17» февраля 2026 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)
ПМ.02 РАЗРАБОТКА И ИНТЕГРАЦИЯ МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
СПЕЦИАЛЬНОСТЬ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО
ОБРАЗОВАНИЯ
09.02.11 Разработка и управление программным обеспечением
(код профессии, специальности СПО)

Программист
(наименование квалификации)

Кафедра: Информатики и математики

Разработчик программы:
Зав. кафедрой

Маскайкин Е.П.

17 февраля 2026 г.

Оглавление

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....	3
2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....	9
3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	25
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....	26

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

1.1. ЦЕЛЬ И МЕСТО ПРОФЕССИОНАЛЬНОГО МОДУЛЯ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Цель модуля: освоение вида деятельности «Разработка и интеграция модулей программного обеспечения», формирование у обучающихся практических навыков проектирования, создания, интеграции, тестирования, документирования и обеспечения безопасности программных модулей и компонентов в корпоративных информационных системах.

Профессиональный модуль включен в обязательную часть образовательной программы и является базовым для формирования профессиональных компетенций специалиста в области разработки и управления программным обеспечением.

1.2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Результаты освоения профессионального модуля соотносятся с планируемыми результатами освоения образовательной программы, представленными в матрице компетенций выпускника.

В результате освоения профессионального модуля обучающийся должен:

Код ОК, ПК	Уметь	Знать	Иметь практический опыт
ОК 01	– распознавать задачу и/или проблему в профессиональном и/или социальном контексте; – анализировать задачу и/или проблему и выделять её составные части; – определять этапы решения задачи; – выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; – составлять план действия; – определять необходимые ресурсы; – реализовывать	– актуальный профессиональный и социальный контекст, в котором приходится работать и жить; – основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; – алгоритмы выполнения работ в профессиональной и смежных областях; – методы работы в профессиональной и смежных сферах; – структуру плана для решения задач; – порядок оценки результатов решения задач	– применения методов анализа и синтеза для решения профессиональных задач

Код ОК, ПК	Уметь	Знать	Иметь практический опыт
	составленный план; – оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)	профессиональной деятельности	
ОК 02	– определять задачи для поиска информации; – определять необходимые источники информации; – планировать процесс поиска; – структурировать получаемую информацию; – выделять наиболее значимое в перечне информации; – оценивать практическую значимость результатов поиска; – оформлять результаты поиска; – применять средства информационных технологий для решения профессиональных задач; – использовать современное программное обеспечение; – использовать различные цифровые средства для решения профессиональных задач	– номенклатуру информационных источников, применяемых в профессиональной деятельности; – приемы структурирования информации; – формат оформления результатов поиска информации; – современные средства и устройства информатизации, порядок их применения; – программное обеспечение в профессиональной деятельности, в том числе цифровые средства	– работы с информационными ресурсами и современными ИТ- инструментами

Код ОК, ПК	Уметь	Знать	Иметь практический опыт
ПК 1.1	– проектировать базы данных (в рамках интеграции модулей)	– основы реляционной модели данных, SQL, принципы нормализации	– проектирования структур данных для взаимодействия модулей
ПК 2.1	– 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 паттернов и структур данных; – создавать диаграммы классов, последовательностей и др.; – выбирать языки программирования и технологии; – проектировать интерфейсы для взаимодействия с другими модулями; – учитывать требования к масштабируемости, производительности и безопасности; – проводить анализ и оптимизацию проектируемого модуля	– основные принципы проектирования модулей ПО; – языки программирования и технологии для реализации модулей; – паттерны проектирования и структуры данных; – методы анализа требований и определения функциональности; – принципы создания интерфейсов (REST, SOAP и др.); – принципы обеспечения безопасности, производительности, масштабируемости; – методы анализа и оптимизации модулей	– проектирования модулей ПО с учётом требований заказчика; – создания архитектурных диаграмм и спецификаций модулей; – определения интерфейсов и взаимодействия модулей в системе

Код ОК, ПК	Уметь	Знать	Иметь практический опыт
ПК 2.2	– разрабатывать модули ПО с использованием различных языков и технологий; – применять паттерны проектирования и структуры данных; – анализировать требования и определять функциональность модуля; – создавать интерфейсы для взаимодействия с другими модулями; – обеспечивать безопасность, производительность и масштабируемость; – оптимизировать модули; – работать с системой контроля версий; – выявлять и устранять узкие места; – проводить анализ и мониторинг производительности; – применять инструменты для рефакторинга и оптимизации кода	– язык программирования, основные конструкции, синтаксис; – паттерны проектирования; – структуры данных; – принципы создания интерфейсов (REST, SOAP); – инструментальное ПО; – методы оптимизации кода и алгоритмов; – эффективные алгоритмы и структуры данных; – многопоточность; – методы оптимизации сетевых протоколов; – кэширование данных; – управление памятью; – техники повышения производительности	– создания модулей ПО на различных языках программирования; – отладки и тестирования разработанных модулей; – применения структурного и объектно-ориентированного программирования; – оптимизации кода и алгоритмов; – мониторинга и анализа производительности
ПК 2.3	– интегрировать модули и компоненты, обеспечивая их взаимодействие; – работать с API и устанавливать соединения между	– общие принципы функционирования аппаратных, программных и программно-аппаратных средств; – международные стандарты локальных	– интеграции программных модулей в единое решение; – работы с API и веб-сервисами; – работы с интеграционными платформами и

Код ОК, ПК	Уметь	Знать	Иметь практический опыт
	компонентами; – отслеживать и устранять конфликты и ошибки интеграции; – анализировать и определять зависимости между модулями; – работать с различными форматами данных и протоколами	сетей; – методы и подходы к интеграции модулей; – принципы версионирования и управления изменениями; – принципы безопасности при интеграции	инструментами; – обеспечения совместимости и стабильности системы
ПК 2.4	– анализировать требования и составлять планы тестирования; – создавать тестовые сценарии и тест-кейсы; – выполнять тестирование вручную и автоматизированно; – анализировать результаты и документировать ошибки; – разрабатывать стратегии отладки и исправлять ошибки; – выполнять модульные тесты; – использовать системы контроля дефектов; – составлять отчёты о тестировании	– принципы и методы тестирования ПО; – основы программирования и архитектуры ПО; – основы БД и SQL; – инструменты автоматизации тестирования; – основы разработки и отладки; – понятие дефекта; – критерии качества ПО; – виды и типы тестирования; – техники ручного и автоматизированного тестирования; – жизненный цикл дефекта; – принципы работы в системе контроля дефектов; – основные понятия о качестве ПО	– отладки ПО на уровне модулей; – тестирования ПО; – формирования тестовых сценариев; – подготовки тестовых платформ; – оценки объёма тестирования; – настройки тестовой среды; – формирования отчётности; – выполнения тестовых процедур
ПК 2.5	– описывать функциональность модулей в документации; – создавать	– стандарты технической документации; – принципы документирования ПО;	– создания технической документации для модулей; – документирования

Код ОК, ПК	Уметь	Знать	Иметь практический опыт
	диаграммы для иллюстрации работы модулей; – программировать с использованием комментариев; – использовать специальные метки/теги для документации; – вести журнал изменений; – разбивать модули на логические блоки; – включать в документацию особенности модулей; – регулярно обновлять документацию	– инструменты для создания технической документации и комментирования кода	кода, API и интерфейсов; – работы со специализированным ПО по документированию кода
ПК 3.1	– выполнять техническое проектирование бизнес-приложений и сопровождение проектных решений (в части интеграции)	– методы проектирования бизнес-приложений, архитектурные подходы	– проектирования бизнес-приложений с учётом интеграционных требований
ПК 3.2	– разрабатывать бизнес-приложения (в части модулей и интеграции)	– технологии разработки бизнес-приложений, паттерны	– разработки бизнес-приложений с модульной архитектурой
ПК 3.6	– осуществлять поддержку и обслуживание бизнес-приложений (в части интеграционных аспектов)	– методы поддержки и обслуживания интегрированных систем	– сопровождения интегрированных решений

2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

2.1. ТРУДОЕМКОСТЬ ОСВОЕНИЯ МОДУЛЯ

Наименование составных частей модуля	Объем в часах	В т.ч. работа с преподавателем	Сам. работа
Учебные занятия по МДК.02.01 (5 семестр)	138	90	48
Учебные занятия по МДК.02.02 (6 семестр)	132	104	28
Учебные занятия по МДК.02.03 (7 семестр)	72	58	14
Учебные занятия по МДК.02.04 (5 семестр)	64	30	34
Учебные занятия по МДК.02.05 (6 семестр)	64	30	34
Учебные занятия по МДК.02.06 (7 семестр)	88	68	20
Итого учебных занятий по МДК	558	380	178
Учебная практика (7 семестр)	144	144	—
Производственная практика (7 семестр)	144	144	—
Итого практика	288	288	—
Всего по модулю	846	668	178

2.2. СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Раздел 1. Разработка программных модулей (МДК.02.01) – 138 часов (5 семестр)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
Тема 1.1. Использование принципов объектно-ориентированного	Содержание: – Модульная архитектура, принципы, преимущества.	60 (лекции – 24, практика – 36, СР – 0)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
программирования при разработке программных модулей	– Архитектурные шаблоны (MVC, MVVM, MVP). – Инструменты разработки, системы контроля версий. – Работа с библиотеками, массивами, коллекциями, строками, датой и временем. – Паттерны проектирования: наследование, интерфейсы, абстрактные классы, порождающие, структурные, поведенческие паттерны. – Система ввода-вывода, работа с файлами и папками, сжатие потоков. – Регулярные выражения, кодирование/декодирование текста. – Асинхронная модель, пул потоков, синхронизация. – Параллельное программирование (создание задач, ожидание, продолжение, отмена). Практические занятия (36 ч): 1. Разработка модулей для работы с массивами (Git). 2. Разработка модулей для работы с коллекциями (Git). 3. Разработка модулей для работы с датой и временем (Git). 4. Разработка модулей с использованием паттернов (Git). 5. Навигация по файловой системе, чтение/запись файлов, потоки. 6. Работа с большими текстами, кодирование, регулярные выражения. 7. Организация асинхронного вызова методов. 8. Создание модуля с параллельными задачами. Самостоятельная работа (24 ч): изучение документации, выполнение заданий.	
Тема 1.2. Ключевые алгоритмы и структуры данных	Содержание: – Алгоритмы и структуры данных, оценка сложности (О-нотации). – Основные структуры: массив,	48 (лекции – 24, практика – 24, СР – 0)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
	связный список, стек, очередь. – Алгоритмы сортировки и поиска, рекурсия. – Хеш-таблицы, разрешение коллизий. – Деревья и графы, поиск в глубину/ширину, алгоритм Дейкстры. – Жадные алгоритмы и динамическое программирование. – Алгоритмы работы с текстом (поиск подстроки, редакционное расстояние, алгоритм Рабина-Карпа). – Кучи и очереди с приоритетом. Практические занятия (24 ч): 9. Оценка сложности алгоритмов. 10. Применение рекурсивных алгоритмов. 11. Алгоритмы сортировки и поиска. 12. Создание хеш-таблиц. 13. Нахождение кратчайших путей (Дейкстра). 14. Решение задачи о рюкзаке (динамическое программирование). 15. Реализация строковых алгоритмов. 16. Реализация приоритетных очередей. Самостоятельная работа (24 ч): решение задач.	
Тема 1.3. Проектирование модулей	Содержание: – Принципы проектирования, методы анализа требований. – Декомпозиция, создание спецификаций. – Принципы безопасности, производительности, масштабируемости. – Проектирование классов, инкапсуляция, наследование, полиморфизм. – Диаграммы классов и компонентов. Практические занятия (18 ч): 17. Анализ требований и определение функциональности. 18. Создание спецификации модуля. 19. Проектирование структуры (диаграммы классов).	30 (лекции – 12, практика – 18, СР – 0)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
	20. Проектирование компонентов (диаграммы компонентов). 21. Проектирование интерфейсов взаимодействия. 22. Анализ и оптимизация модуля. Самостоятельная работа (0 ч):	
Тема 1.4. Создание модулей для взаимодействия с пользователем	Содержание: – Виды интерфейсов (командная строка, графический, речевой). – Принципы разработки GUI, компоненты, компоновка, события. – Работа с окнами, открытие/закрытие, передача данных. – Валидация, сообщения об ошибках. – Многопоточность и асинхронность в GUI. – Стил, цвет, шрифты, стилизация. – Работа с текстом, изображениями, графиками (библиотеки). Практические занятия (12 ч): 23. Проектирование главного окна с панелями. 24. Разработка многооконного приложения. 25. Разработка стилей. 26. Модули для текстовой информации. 27. Модули для работы с изображениями. 28. Модули для графиков и диаграмм. 29. Модули для аудио/видео. 30. Загрузка данных из интернета в фоновом режиме. 31. Форма регистрации с валидацией. Самостоятельная работа (0 ч):	– (часы уже учтены в практических занятиях)
Тема 1.5. Создание модулей для взаимодействия с базами данных	Содержание: – Технологии доступа к данным, безопасность. – ORM: понятие, преимущества, использование. – CRUD-операции, выполнение запросов. Практические занятия (12 ч):	—

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
	32. Разработка модулей для работы с БД. 33. Разработка модулей для запросов к БД. Самостоятельная работа (0 ч):	
Тема 1.6. Принципы безопасности, производительности и масштабируемости	Содержание: – Безопасность, производительность, масштабируемость (горизонтальная/вертикальная). – Метрики, инструменты мониторинга, нагрузочное тестирование. – Оптимизация кода: цели, методы, профилирование. Практические занятия (12 ч): 34. Оптимизация модулей. 35. Решение задач на оптимизацию алгоритмов. 36. Анализ производительности, обеспечение масштабируемости. 37. Выявление и устранение узких мест. 38. Обеспечение безопасности. Самостоятельная работа (0 ч):	–
Итого по МДК.02.01		138

Раздел 2. Осуществление интеграции программных модулей (МДК.02.02) – 132 часа (6 семестр)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
Тема 2.1. Основы интеграции программных модулей	Содержание: – Разработка REST API: клиент-сервер, HTTP структура, методы (GET, POST, DELETE, PUT, PATCH), заголовки, тело. – Маршрутизация, группировка, статические ресурсы. – Обработка запросов (path, query, body), валидация.	72 (лекции – 24, практика – 48, СР – 0)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
	– Формирование ответов, сериализация/десериализация. – Фоновые задачи. – Аутентификация и авторизация (oAuth, JWT, сессии, роли). – WebSocket API: взаимодействие, настройки соединения. – Микросервисная архитектура, синхронное (REST, gRPC) и асинхронное (брокеры сообщений) взаимодействие. Практические занятия (48 ч): 1. Клиентское приложение для работы с публичным API. 2. REST API приложение с CRUD для 3–4 сущностей. 3. Работа с удалённым источником данных. 4. Работа со статическими ресурсами (загрузка/удаление изображений). 5. Обработка path и query параметров. 6. Обработка ошибок и передача сообщений. 7. Валидация данных. 8. Добавление фоновых задач. 9. Добавление аутентификации, авторизации, ролевой системы. 10. Клиент для работы с WebSocket. 11. Серверное приложение по WebSocket. 12. Микросервисное приложение с REST. 13. Микросервисное приложение с gRPC. 14. Микросервисное приложение с брокером сообщений (consumer/producer). Самостоятельная работа (12 ч): изучение документации, выполнение заданий.	
Тема 2.2. Управление и мониторинг интегрированной системы	Содержание: – Настройка конфигурации и сборки. – Логирование: конфигурация, уровни, форматы. – Мониторинг: метрики, сбор статистики. – Контейнеризация, средства доставки и развёртывания. Практические занятия (16 ч): 15. Настройка конфигурации REST API (порт, хост, ключи). 16. Внедрение логирования.	24 (лекции – 8, практика – 16, СР – 0)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
	17. Упаковка приложения в контейнер и доставка. Самостоятельная работа (8 ч): изучение инструментов контейнеризации.	
Тема 2.3. Безопасность при интеграции	Содержание: – HTTPS, WSS. – Защита от SQL-инъекций, CSRF, XSS. – Хеширование паролей с солью. – Анализ уязвимостей, аудиты безопасности. Практические занятия (12 ч): 18. Добавление SSL сертификата. 19. Настройка конфигурации безопасности. Самостоятельная работа (4 ч): изучение стандартов безопасности.	18 (лекции – 6, практика – 12, СР – 0)
Тема 2.4. Оптимизация и масштабируемость интегрированных решений	Содержание: – Горизонтальное и вертикальное масштабирование. – Кэширование данных, оптимизация запросов к БД. – Профилирование, уменьшение времени отклика. Практические занятия (12 ч): 20. Реализация кэширования. 21. Оптимизация производительности через профилирование. Самостоятельная работа (4 ч): анализ производительности.	18 (лекции – 6, практика – 12, СР – 0)
Итого по МДК.02.02		132

Раздел 3. Поддержка и тестирование программных модулей (МДК.02.03) – 72 часа (7 семестр)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
Тема 3.1. Качество программного обеспечения	Содержание: – Определение качества, метрики (статичные и динамические). – Стандарты и модели качества,	12 (лекции – 4, практика – 8, СР – 0)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
	инструменты оценки. Практические занятия (8 ч): 1. Анализ качества модуля по метрикам. 2. Статический анализ кода для выявления дефектов. 3. Разработка процессов обеспечения качества. Самостоятельная работа (2 ч): изучение стандартов.	
Тема 3.2. Отладка программного модуля	Содержание: – Понятие отладки, ошибки, дефекты, сбои, отказы. – Инструменты отладки, пошаговая отладка, точки останова. – Стратегии поиска ошибок. – Документирование отладки. Практические занятия (8 ч): 4. Разработка стратегии отладки и исправление ошибок. 5. Код-ревью и парное программирование. Самостоятельная работа (2 ч): практика отладки.	12 (лекции – 4, практика – 8, СР – 0)
Тема 3.3. Обработка исключений	Содержание: – Понятие исключения, типы, механизм обработки. – Логирование ошибок. Практические занятия (8 ч): 6. Основные конструкции обработки исключений. 7. Практическое использование исключений. 8. Обработка ошибок в RESTful API. Самостоятельная работа (2 ч): разработка собственного обработчика.	12 (лекции – 4, практика – 8, СР – 0)
Тема 3.4. Тестирование программных модулей	Содержание: – Процесс тестирования, этапы. – Техники ручного и автоматизированного тестирования. – Модель работы с дефектами. – Виды и типы тестирования (функциональное, нефункциональное,	36 (лекции – 12, практика – 24, СР – 0)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
	модульное, интеграционное, системное, нагрузочное и др.). – Тестирование по белому и чёрному ящику: методы покрытия, классы эквивалентности, граничные значения. – Модульные тесты, тестирование интеграции, RESTful API, производительности. – Разработка через тестирование. Практические занятия (24 ч): 9. Анализ требований, составление плана тестирования, работа с системой контроля дефектов. 10–11. Тестирование методами белого ящика (покрытие операторов, условий, комбинаторное). 12–14. Тестирование по чёрному ящику (классы эквивалентности, граничные значения, причинно-следственные связи). 15–18. Разработка модульных тестов для различных ситуаций. 19. Тестирование интеграции. 20. Тестирование RESTful API. 21. Тестирование производительности. 22. Разработка через тестирование. Самостоятельная работа (4 ч): изучение инструментов тестирования.	
Тема 3.5. Поддержка программных модулей	Содержание: – Работы по поддержке: исправление дефектов, ревью, рефакторинг, оптимизация. – Стандарты оформления документации, принципы документирования, инструменты. – Виды тестовой документации, тестовые случаи, сценарии, отчёты о дефектах. Практические занятия (8 ч): 23. Разработка документации, ведение журнала изменений. 24. Ревью, рефакторинг, оптимизация кода. 25. Разработка программы и методики испытаний.	– (часы уже учтены в практических занятиях)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
	26. Создание спецификаций API. Самостоятельная работа (4 ч): оформление документации.	
Итого по МДК.02.03		72

Раздел 4. Математическое моделирование (МДК.02.04) – 64 часа (5 семестр)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
Тема 4.1. Математическое моделирование как методология	Содержание: – Понятие модели, классификация, математическая модель, типы, принципы построения, этапы моделирования. Практические занятия (4 ч): 1. Построение простейших математических моделей. Самостоятельная работа (4 ч): изучение подходов.	8 (лекции – 4, практика – 4, СР – 0)
Тема 4.2. Линейное программирование	Содержание: – Каноническая задача, основные определения. Графический и симплекс-метод. Транспортная задача, задача о назначениях, целочисленное программирование. Практические занятия (8 ч): 2. Решение задач симплекс-методом. 3. Решение транспортной задачи. 4. Решение задачи о назначениях. 5. Применение инструментальных средств. Самостоятельная работа (6 ч): решение задач.	14 (лекции – 6, практика – 8, СР – 0)
Тема 4.3. Нелинейное программирование	Содержание: – Основные понятия, методы решения. Практические занятия (4 ч): 6. Решение задач нелинейного программирования. Самостоятельная работа (4 ч): изучение методов.	8 (лекции – 4, практика – 4, СР – 0)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
Тема 4.4. Динамическое программирование	Содержание: – Основные понятия, задачи (распределение ресурсов, замена оборудования, оптимальный путь, резервирование). Практические занятия (4 ч): 7. Решение задач распределения ресурсов и замены оборудования. 8. Решение задач оптимального пути и резервирования. Самостоятельная работа (6 ч): решение задач.	10 (лекции – 4, практика – 4, СР – 0)
Тема 4.5. Сетевые методы планирования и управления	Содержание: – Основы теории графов, кратчайший путь, дерево решений, сетевые графики, временные параметры. Практические занятия (4 ч): 9. Решение задач сетевого планирования. Самостоятельная работа (4 ч): изучение методов.	8 (лекции – 4, практика – 4, СР – 0)
Тема 4.6. Системы массового обслуживания	Содержание: – Марковские процессы, классификация СМО, схема гибели и размножения. Практические занятия (4 ч): 10. Расчёт характеристик простейших СМО. Самостоятельная работа (4 ч): изучение теории.	8 (лекции – 4, практика – 4, СР – 0)
Тема 4.7. Теория игр	Содержание: – Основные понятия, матричные, биматричные игры, игры в развёрнутой форме. Практические занятия (4 ч): 11. Решение игр с нулевой суммой. 12. Решение задач в развёрнутой форме. Самостоятельная работа (4 ч): изучение теории.	8 (лекции – 4, практика – 4, СР – 0)
Итого по МДК.02.04		64

Раздел 5. Численные методы (МДК.02.05) – 64 часа (6 семестр)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
Тема 5.1. Приближенные числа и действия над ними	Содержание: – Способы хранения чисел, абсолютная и относительная погрешность, верные/сомнительные цифры, погрешности арифметических действий. Практические занятия (4 ч): 1. Вычисление погрешностей. Самостоятельная работа (4 ч): расчёты.	8 (лекции – 4, практика – 4, СР – 0)
Тема 5.2. Численные методы решения алгебраических и трансцендентных уравнений	Содержание: – Отделение корней, метод половинного деления, простой итерации, методы Ньютона (хорд, касательных, комбинированный). Практические занятия (8 ч): 2. Решение уравнений методом половинного деления и простых итераций. 3. Решение уравнений методами Ньютона. 4. Мониторинг производительности разработанных приложений. Самостоятельная работа (6 ч): реализация методов.	14 (лекции – 6, практика – 8, СР – 0)
Тема 5.3. Численные методы решения СЛАУ	Содержание: – Метод Гаусса (решение, определитель, обратная матрица), методы простой итерации и Зейделя. Практические занятия (6 ч): 5. Решение СЛАУ методом Гаусса. 6. Решение СЛАУ методами простой итерации и Зейделя. 7. Мониторинг производительности. Самостоятельная работа (6 ч): реализация методов.	12 (лекции – 6, практика – 6, СР – 0)
Тема 5.4. Интерполяция и экстраполяция функций	Содержание: – Интерполяционные многочлены Лагранжа, Ньютона, сплайн-интерполяция, экстраполяция. Практические занятия (4 ч): 8. Составление интерполяционных формул. 9. Экстраполяция функций.	10 (лекции – 4, практика – 4, СР – 0)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
	Самостоятельная работа (6 ч): решение задач.	
Тема 5.5. Численное интегрирование	Содержание: – Квадратурные формулы Ньютона-Котеса, Гаусса. Практические занятия (4 ч): 10. Вычисление интегралов по формулам Ньютона-Котеса. 11. Вычисление интегралов по формулам Гаусса. Самостоятельная работа (6 ч): решение задач.	10 (лекции – 4, практика – 4, СР – 0)
Тема 5.6. Численные методы решения ОДУ	Содержание: – Метод Эйлера (уточнённая схема), метод Рунге-Кутты. Практические занятия (4 ч): 12. Решение ОДУ методом Эйлера. 13. Решение ОДУ методом Рунге-Кутты. Самостоятельная работа (6 ч): реализация методов.	10 (лекции – 4, практика – 4, СР – 0)
Итого по МДК.02.05		64

Раздел 6. Безопасность программного обеспечения (МДК.02.06) – 88 часов (7 семестр)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
Тема 6.1. Основы безопасности ПО	Содержание: – Введение в кибербезопасность, модели угроз, анализ рисков. – Уязвимости веб-приложений (OWASP Top 10). – Безопасная аутентификация и авторизация. – Криптография для разработчиков. Практические занятия (40 ч): 1. Анализ кода на уязвимости (ручной review). 2. SQL-инъекции: эксплуатация и защита. 3. XSS-атаки: создание и	60 (лекции – 20, практика – 40, СР – 0)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
	предотвращение. 4. CSRF-защита. 5. Составление модели угроз для веб-приложения. 6. Настройка JWT аутентификации с refresh-токенами. 7. Реализация RBAC. 8. Шифрование AES/RSA. 9. Хэширование паролей с солью (bcrypt, Argon2). 10. Анализ трафика Wireshark. 11. Сканирование уязвимостей OWASP ZAP, Burp Suite. 12. Настройка HTTPS, самоподписанные сертификаты. 13. Защита от brute-force (ограничение попыток). 14. Безопасная работа с файлами. 15. Безопасная десериализация. 16. Аудит логов безопасности. 17. Настройка CORS. 18. Защита от DDoS (rate limiting). 19. Безопасная работа с памятью. 20. Создание безопасного API с валидацией. Самостоятельная работа (10 ч): изучение OWASP, практика.	
Тема 6.2. Разработка безопасного ПО и прикладная криптография	Содержание: – Принципы безопасного проектирования архитектуры. – Криптографические протоколы, реализация в мобильных, веб- и облачных приложениях. Практические занятия (28 ч): 21. Реализация end-to-end шифрования (Signal Protocol). 22. Настройка TLS 1.3. 23. Создание OAuth 2.0 провайдера с PKCE. 24. Имплементация JWE. 25. Разработка безопасной системы голосования (гомоморфное шифрование). 26. Создание криптовалютного кошелька (ECDSA).	28 (лекции – 0, практика – 28, СР – 0)

Наименование разделов и тем	Примерное содержание учебного материала, практических и лабораторных занятий	Объем часов
	27. Реализация менеджера паролей с клиентским шифрованием. 28. Настройка HSM эмулятора. 29. Разработка защищённого файлового хранилища. 30. Имплементация zero-knowledge proof. 31. Создание смарт-контракта с защитой от reentrancy. 32. Реализация secure multi-party computation. 33. Настройка пост-квантовой криптографии. 34. Разработка secure API gateway. 35. Создание аппаратно-защищённого хранилища ключей. 36. Имплементация системы цифровых подписей с timestamping. 37. Настройка certificate transparency. 38. Разработка защищённого управления сессиями. 39. Создание криптографически безопасного RNG. Самостоятельная работа (10 ч): реализация проектов.	
Итого по МДК.02.06		88

2.3.КУРСОВАЯ РАБОТА (ПРОЕКТ) (в рамках МДК.02.02)

Выполняется комплексный курсовой проект по всему профессиональному модулю.

Примерная тематика курсовых проектов (работ):

1. Развитие и интеграция HR-модуля в кадровую систему предприятия для автоматизации процессов найма и управления персоналом.
2. Создание системы управления проектами с использованием модульной архитектуры.
3. Разработка и интеграция модуля управления проектами в CRM-систему.
4. Создание и интеграция платежного модуля для электронной коммерции.
5. Создание платформы для обмена сообщениями.
6. Интеграция разных баз данных в единую систему.
7. Модульная архитектура и интеграция модулей в распределенной системе управления складскими запасами.
8. Создание модуля аутентификации.
9. Интеграция базы данных с модулем обработки данных.

10. Создание и интеграция аналитического модуля для обработки данных в медицинской информационной системе.
11. Разработка и интеграция образовательного модуля в LMS-систему.
12. Разработка и интеграция геолокационного модуля в систему управления транспортом.
13. Разработка и интеграция нового модуля для системы управления складом.

Структура курсового проекта:

1. Введение (актуальность, цели, задачи).
2. Анализ предметной области, техническое задание.
3. Проектирование модулей (архитектура, диаграммы классов, компонентов, интерфейсы).
4. Реализация модулей (код, документация).
5. Интеграция модулей (настройка взаимодействия, тестирование).
6. Обеспечение безопасности, производительности.
7. Тестирование и отладка (модульные тесты, интеграционное тестирование).
8. Документирование (техническая документация, API-спецификации, отчёты о тестировании).
9. Заключение, выводы.
1. Список использованных источников.

2.4.УЧЕБНАЯ И ПРОИЗВОДСТВЕННАЯ ПРАКТИКИ

Учебная практика – 144 часа (7 семестр)

Виды работ:

1. Проектирование модулей ПО с учётом технического задания.
2. Визуализация и описание архитектурных решений.
3. Определение интерфейсов и взаимодействия модулей.
4. Создание модулей ПО.
5. Работа с API и веб-сервисами.
6. Работа с интеграционными платформами и инструментами.
7. Отладка ПО на уровне модулей.
8. Тестирование ПО (формирование сценариев, подготовка платформ, оценка объёма, выполнение тестов).
9. Оформление отчётности по тестированию.
10. Создание технической документации, документирование кода, API, интерфейсов.
11. Работа со специализированным ПО по документированию.

Производственная практика – 144 часа (7 семестр)

Виды работ:

1. Проектирование модулей ПО с учётом технического задания.
2. Визуализация и описание архитектурных решений.
3. Определение интерфейсов и взаимодействия модулей.
4. Создание модулей ПО, оптимизация кода и алгоритмов, мониторинг производительности.
5. Интеграция модулей в единое решение, работа с API, веб-сервисами, интеграционными платформами.
6. Обеспечение совместимости и стабильности системы.
7. Отладка ПО на уровне модулей.

8. Тестирование ПО (сценарии, подготовка платформ, настройка среды, выполнение тестов, отчётность).

9. Создание технической документации, документирование кода, API, интерфейсов.

1. Работа со специализированным ПО по документированию.

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

3.1. Материально-техническое обеспечение

Реализация программы требует наличия лаборатории «Разработки и интеграции программных решений», оснащённой в соответствии с требованиями ФГОС.

Оборудование лаборатории:

- рабочие места по количеству обучающихся (компьютеры с доступом в Интернет);
- рабочее место преподавателя;
- мультимедийное оборудование (проектор, экран).

Программное обеспечение:

- ОС Windows / Linux;
- среды разработки: Visual Studio, VS Code, PyCharm, IntelliJ IDEA (или аналоги);
- системы контроля версий (Git);
- СУБД (PostgreSQL, MySQL, MongoDB);
- инструменты для тестирования (JUnit, pytest, Postman, Selenium);
- инструменты для контейнеризации (Docker);
- средства для моделирования и численных методов (Matlab, Octave, Python с библиотеками);
- браузеры и инструменты разработчика.

3.2. Учебно-методическое обеспечение

Основная литература:

1. 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>
2. 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>.
3. Лаврищева, Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. — Москва : Издательство Юрайт, 2026. — 280 с. — (Высшее образование). — ISBN 978-5-534-01056-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/584533>.

Дополнительная литература:

4. Маркин, А. В. Базы данных. PostgreSQL : учебник для среднего профессионального образования / А. В. Маркин. — Москва : Издательство Юрайт, 2026. — 828 с. —

- (Профессиональное образование). — ISBN 978-5-534-21780-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/590497>.
5. Чернышев, С. А. Основы программирования на Python : учебник для среднего профессионального образования / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 349 с. — (Профессиональное образование). — ISBN 978-5-534-17056-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588671>
 6. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская. — 4-е изд. — Москва : Издательство Юрайт, 2025. — 108 с. — (Профессиональное образование). — ISBN 978-5-534-20429-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563861>.
 7. Подбельский, В. В. Программирование. Базовый курс C# : учебник для среднего профессионального образования / В. В. Подбельский. — Москва : Издательство Юрайт, 2026. — 369 с. — (Профессиональное образование). — ISBN 978-5-534-11467-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/598935>.

Интернет-ресурсы:

<https://metanit.com> – материалы по программированию;
<https://www.sql-ex.ru> – упражнения по SQL;
<https://www.postgresql.org/docs> – документация PostgreSQL;
<https://docs.microsoft.com/ru-ru/dotnet> – документация .NET;
<https://docs.python.org/3> – документация Python;
<https://owasp.org> – материалы по безопасности веб-приложений.

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

4.1. Критерии, формы и методы контроля (оценки)

Код ПК, ОК	Критерии оценки результата (показатели освоённости компетенций)	Формы контроля и методы оценки
ОК 01, ОК 02	– демонстрирует способность анализировать задачи, планировать поиск информации, структурировать данные, оценивать результаты	Экспертное наблюдение; защита курсового проекта; отчёты по практикам; тестирование
ПК 2.1	– проектирует модули, создаёт архитектурные диаграммы, определяет интерфейсы, анализирует требования, выбирает технологии, учитывает масштабируемость и безопасность	Практические работы; защита курсового проекта; экзамен по МДК.02.01
ПК 2.2	– разрабатывает модули на различных языках, применяет паттерны, оптимизирует код, работает с Git, проводит профилирование	Практические работы; контрольные работы; экзамен по МДК.02.01

Код ПК, ОК	Критерии оценки результата (показатели освоённости компетенций)	Формы контроля и методы оценки
ПК 2.3	– интегрирует модули, работает с API, устраняет конфликты, анализирует зависимости, использует различные форматы и протоколы	Практические работы; экзамен по МДК.02.02; защита курсового проекта
ПК 2.4	– составляет планы тестирования, разрабатывает тест-кейсы, выполняет автоматизированное тестирование, документирует ошибки, использует системы контроля дефектов	Практические работы; дифзачёт по МДК.02.03; отчёт по учебной практике
ПК 2.5	– создаёт техническую документацию, документирует код и API, ведёт журнал изменений, обновляет документацию	Практические работы; дифзачёт по МДК.02.03; отчёт по производственной практике
ПК 3.1, ПК 3.2, ПК 3.6	– применяет знания при проектировании и разработке бизнес-приложений, интеграции и сопровождении	Экспертное наблюдение; защита курсового проекта; экзамен по МДК.02.02

4.2. Комплект оценочных заданий по модулю

Тип задания	Инструкция	Пример задания	Компетенции
Задание закрытого типа на установление соответствия (Знать)	Прочитайте текст и установите соответствие между терминами и определениями.	Установите соответствие между паттерном проектирования и его описанием: А) Singleton; Б) Factory; В) Observer; Г) Decorator. 1) Обеспечивает создание объектов без указания конкретных классов; 2)	ПК 2.1, ПК 2.2

Тип задания	Инструкция	Пример задания	Компетенции
		Гарантирует единственный экземпляр класса; 3) Определяет механизм уведомления об изменении состояния; 4) Динамически добавляет поведение объекту.	
Задание закрытого типа на установление соответствия (Уметь)	Прочитайте текст и установите соответствие между HTTP-методами и их назначением.	Установите соответствие между HTTP-методом и его функцией: А) GET; Б) POST; В) PUT; Г) DELETE. 1) Создание нового ресурса; 2) Получение данных; 3) Удаление ресурса; 4) Полное обновление ресурса.	ПК 2.3
Задание закрытого типа на установление последовательности (Знать)	Прочитайте текст и установите правильную последовательность этапов разработки модуля.	Расположите этапы разработки модуля в правильном порядке: 1) Тестирование; 2) Проектирование; 3) Реализация; 4) Анализ требований.	ПК 2.1, ПК 2.2, ПК 2.4
Задание закрытого типа на установление	Прочитайте текст и установите правильную последовательность	Расположите действия при отладке в правильном порядке: 1) Воспроизведение	ПК 2.4

Тип задания	Инструкция	Пример задания	Компетенции
последовательности (Уметь)	ь действий при отладке.	ошибки; 2) Установка точек останова; 3) Анализ состояния переменных; 4) Исправление кода.	
Задание с выбором одного верного ответа (Знать)	Прочитайте текст, выберите правильный ответ и запишите аргументы.	Какой протокол используется для асинхронного взаимодействия микросервисов через брокер сообщений? 1) HTTP; 2) gRPC; 3) AMQP; 4) WebSocket.	ПК 2.3
Задание с выбором одного верного ответа (Уметь)	Прочитайте текст, выберите правильный ответ и запишите аргументы.	Какой инструмент позволяет упаковать приложение в контейнер для последующего развёртывания? 1) Git; 2) Docker; 3) Jenkins; 4) Maven.	ПК 2.3
Задание с выбором нескольких вариантов (Знать)	Прочитайте текст, выберите все правильные ответы и запишите обоснование.	Какие из перечисленных методов относятся к тестированию по чёрному ящику? 1) Метод классов эквивалентности; 2) Метод покрытия операторов; 3) Метод граничных значений;	ПК 2.4

Тип задания	Инструкция	Пример задания	Компетенции
		4) Комбинаторное покрытие условий.	
Задание с выбором нескольких вариантов (Уметь)	Прочитайте текст, выберите все правильные ответы и запишите обоснование.	Какие форматы данных могут использоваться при обмене информацией в REST API? 1) JSON; 2) XML; 3) HTML; 4) YAML.	ПК 2.3
Задание комбинированное с выбором одного верного ответа и обоснованием (Знать)	Прочитайте текст, выберите правильный ответ и запишите развёрнутое обоснование.	Что такое CI/CD? 1) Непрерывная интеграция и доставка; 2) Система контроля версий; 3) Методология тестирования; 4) Архитектурный паттерн.	ПК 2.3
Задание комбинированное с выбором одного верного ответа и обоснованием (Уметь)	Прочитайте текст, выберите правильный ответ и запишите развёрнутое обоснование.	Какой метод оптимизации позволяет уменьшить время выполнения запросов к БД путём хранения часто используемых данных в памяти? 1) Кэширование; 2) Индексирование; 3) Нормализация; 4) Репликация.	ПК 2.3, ПК 2.2

Тип задания	Инструкция	Пример задания	Компетенции
Задание комбинированное с выбором нескольких вариантов и обоснованием (Знать)	Прочитайте текст, выберите все правильные ответы и запишите развёрнутое обоснование.	Какие из перечисленных уязвимостей входят в OWASP Top 10? 1) SQL-инъекции; 2) XSS; 3) CSRF; 4) Небезопасная десериализация.	ПК 2.2, ПК 2.3
Задание комбинированное с выбором нескольких вариантов и обоснованием (Уметь)	Прочитайте текст, выберите все правильные ответы и запишите развёрнутое обоснование.	Какие меры помогут защитить REST API от несанкционированного доступа? 1) Использование JWT; 2) Настройка CORS; 3) Ограничение скорости запросов; 4) Шифрование данных в транзите.	ПК 2.3, ПК 2.5
Задание открытого типа с развёрнутым ответом (Знать)	Прочитайте текст и запишите развёрнутый обоснованный ответ.	Опишите принципы построения RESTful API и основные требования к нему.	ПК 2.3
Задание открытого типа с развёрнутым ответом (Уметь)	Прочитайте текст и запишите развёрнутый обоснованный ответ.	Напишите пример кода на Python с использованием Flask для создания простого REST API с одним GET-эндпоинтом.	ПК 2.3

Тип задания	Инструкция	Пример задания	Компетенции
Задание открытого типа с развёрнутым ответом (Знать)	Прочитайте текст и запишите развёрнутый обоснованный ответ.	Объясните разницу между модульным тестированием и интеграционным тестированием.	ПК 2.4
Задание открытого типа с развёрнутым ответом (Уметь)	Прочитайте текст и запишите развёрнутый обоснованный ответ.	Опишите процесс документирования API с использованием OpenAPI (Swagger).	ПК 2.5
Практическое задание (Уметь)	Выполните задание и предоставьте код.	Создайте модуль на выбранном языке для вычисления факториала числа с использованием рекурсии и напишите модульный тест для него.	ПК 2.2, ПК 2.4
Практическое задание (Уметь)	Выполните задание и предоставьте код.	Разработайте REST API для управления списком задач (CRUD) с использованием Spring Boot (или аналогичного фреймворка).	ПК 2.3
Практическое задание (Уметь)	Выполните задание и предоставьте отчёт.	Проведите анализ кода предоставленного модуля на предмет	ПК 2.2, ПК 2.5

Тип задания	Инструкция	Пример задания	Компетенции
		уязвимостей (SQL-инъекции, XSS) и предложите меры защиты.	
Практическое задание (Уметь)	Выполните задание и предоставьте документацию.	Разработайте техническую документацию для модуля интеграции с внешним API (включая описание эндпоинтов, форматы запросов/ответов, примеры).	ПК 2.5

4.3. Перечень вопросов к дифференцированному зачёту, зачету и экзамену

МДК.02.01 – экзамен (5 семестр)

1. Модульная архитектура, принципы, преимущества.
2. Архитектурные шаблоны MVC, MVVM, MVP.
3. Паттерны проектирования: классификация, примеры.
4. Принципы ООП (инкапсуляция, наследование, полиморфизм, абстракция).
5. Структуры данных: массив, список, стек, очередь, хеш-таблица.
6. Алгоритмы сортировки и поиска, оценка сложности.
7. Рекурсия, примеры, преимущества и недостатки.
8. Динамическое программирование (задача о рюкзаке, редакционное расстояние).
9. Алгоритмы на графах (поиск в глубину/ширину, Дейкстра).
10. Работа с файлами и потоками ввода-вывода.
11. Асинхронное программирование, пул потоков.
12. Параллельное программирование, задачи, продолжения, отмена.
13. Проектирование классов и интерфейсов.
14. Диаграммы классов и компонентов (UML).
15. Разработка GUI: компоненты, события, валидация.
16. Работа с изображениями, графиками, мультимедиа.
17. ORM, CRUD-операции, работа с БД.
18. Принципы безопасности, производительности и масштабируемости.
19. Методы оптимизации кода, профилирование.
20. Системы контроля версий (Git).

МДК.02.02 – экзамен (6 семестр)

1. Архитектура REST API, HTTP методы, заголовки, коды состояния.
2. Маршрутизация, обработка параметров (path, query).
3. Валидация данных, обработка ошибок.
4. Аутентификация и авторизация (JWT, OAuth 2.0, сессии).
5. WebSocket: принципы, взаимодействие клиента и сервера.
6. Микросервисная архитектура: синхронное и асинхронное взаимодействие.
7. gRPC: особенности, использование.
8. Брокеры сообщений (RabbitMQ, Kafka).
9. Логирование, уровни, конфигурация.
10. Мониторинг приложений, метрики.
11. Контейнеризация (Docker), оркестрация (Kubernetes).
12. HTTPS, SSL/TLS, самоподписанные сертификаты.
13. Защита от SQL-инъекций, XSS, CSRF.
14. Хеширование паролей, соль.
15. Безопасность при интеграции, анализ уязвимостей.
16. Кэширование данных, стратегии.
17. Оптимизация запросов к БД.
18. Профилирование и уменьшение времени отклика.
19. Версионирование API.
20. CI/CD: концепции, инструменты.

МДК.02.03 – дифзачет (7 семестр)

1. Качество ПО: метрики, стандарты.
2. Статический анализ кода, инструменты.
3. Отладка: понятие, этапы, стратегии.
4. Обработка исключений, типы исключений.
5. Код-ревью и парное программирование.
6. Модель жизненного цикла дефекта.
7. Виды тестирования (функциональное, нефункциональное).
8. Типы тестирования (модульное, интеграционное, системное, приемочное, нагрузочное).
9. Тестирование по белому ящику: методы покрытия.
10. Тестирование по чёрному ящику: классы эквивалентности, граничные значения.
11. Автоматизация тестирования, инструменты.
12. Разработка через тестирование (TDD).
13. Тестирование REST API (Postman, Newman).
14. Тестирование производительности (JMeter).
15. Системы контроля дефектов (Jira, Bugzilla).
16. Тестовая документация: планы, тест-кейсы, сценарии, отчёты.
17. Документирование кода (комментарии, XML-документация).
18. Документирование API (OpenAPI, Swagger).
19. Рефакторинг кода, цели и методы.
20. Поддержка ПО: версионирование, управление изменениями.

МДК.02.04 – дифзачет (5 семестр)

1. Математическое моделирование: этапы, классификация моделей.
2. Линейное программирование: каноническая форма, симплекс-метод.
3. Транспортная задача, метод потенциалов.
4. Задача о назначениях, венгерский алгоритм.
5. Нелинейное программирование: методы решения.

6. Динамическое программирование: принцип оптимальности Беллмана.
7. Сетевые модели: графы, нахождение кратчайшего пути, дерево решений.
8. Сетевые графики, расчёт временных параметров.
9. Системы массового обслуживания: классификация, показатели эффективности.
10. Марковские процессы, схема гибели и размножения.
11. Теория игр: матричные игры, седловая точка, смешанные стратегии.
12. Биматричные игры.
13. Имитационное моделирование: принципы, инструменты.

МДК.02.05 – зачет (6 семестр)

1. Приближенные числа: погрешности, верные цифры.
2. Погрешности арифметических действий.
3. Отделение корней, метод половинного деления.
4. Метод простой итерации.
5. Методы Ньютона (хорд, касательных).
6. Решение СЛАУ методом Гаусса.
7. Метод простой итерации и Зейделя для СЛАУ.
8. Интерполяция многочленами Лагранжа и Ньютона.
9. Сплайн-интерполяция.
10. Экстраполяция функций.
11. Численное интегрирование: формулы Ньютона-Котеса.
12. Квадратурные формулы Гаусса.
13. Численное решение ОДУ: метод Эйлера.
14. Метод Рунге-Кутты.
15. Численные методы оптимизации (дихотомия, золотое сечение).
16. Методы минимизации функций нескольких переменных (покоординатный спуск, наискорейший спуск).

МДК.02.06 – дифзачет (7 семестр)

1. Основы кибербезопасности, модели угроз.
2. OWASP Top 10.
3. SQL-инъекции: эксплуатация и защита.
4. XSS-атаки, защита.
5. CSRF, защита.
6. Аутентификация с использованием JWT.
7. RBAC, управление ролями.
8. Хеширование паролей (bcrypt, Argon2).
9. Симметричное шифрование (AES).
10. Асимметричное шифрование (RSA).
11. SSL/TLS, настройка HTTPS.
12. Защита от brute-force атак (rate limiting).
13. Безопасная десериализация.
14. Безопасность API (валидация, CORS).
15. Логирование безопасности, аудит.
16. Безопасность контейнеров и облачных сред.
17. Криптографические протоколы (Signal, OAuth).
18. JWE, JWS.
19. Zero-knowledge proof.
20. Пост-квантовая криптография.

4.5. Критерии оценивания

Оценка	Критерии
Отлично	– Теоретический материал освоен полностью, без пробелов; – Практические задания выполнены верно, код работает, соответствует техническому заданию; – Обоснован выбор инструментов и методов; – Соблюдены требования к документации; – На защите курсового проекта/экзамене демонстрирует глубокое понимание темы, отвечает на все вопросы.
Хорошо	– Теоретический материал освоен полностью, но есть незначительные пробелы; – Практические задания выполнены с небольшими ошибками, которые не влияют на общий результат; – Обоснование выбора решений частичное; – Документация оформлена, но есть недочёты; – На защите отвечает на большинство вопросов, но не на все.
Удовлетворительно	– Теоретический материал освоен частично, есть существенные пробелы; – Практические задания выполнены с ошибками, требуют доработки; – Обоснование выбора решений слабое; – Документация оформлена не полностью; – На защите отвечает только на наводящие вопросы.
Неудовлетворительно	– Теоретический материал не освоен; – Практические задания не выполнены или выполнены с грубыми ошибками; – Отсутствует документация; – На защите не может ответить на вопросы.