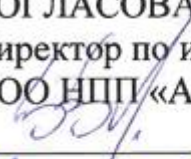


Федеральное государственное образовательное бюджетное
учреждение высшего образования
**«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)**

Красноярский филиал Финуниверситета

СОГЛАСОВАНО
Директор по инновациям
ООО НПП «АВАКС-ГеоСервис»

В.В. Кретинин
«24» июня 2026 г.



УТВЕРЖДАЮ
Заместитель директора
по учебно-методической работе
Красноярского филиала
Финуниверситета

О.С. Вергейчик
«24» июня 2026 г.

РАБОЧАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
ПМ. 02 РАЗРАБОТКА И ИНТЕГРАЦИЯ МОДУЛЕЙ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ
по специальности 09.02.11 Разработка и управление программным
обеспечением

г. Красноярск – 2026

Рабочая программа профессионального модуля разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением

Разработчик:

Цирулькевич А.В., преподаватель 1КК

(фамилия, имя, отчество, должность, квалификационная категория)

Рабочая программа профессионального модуля рассмотрена и рекомендована к утверждению на заседании предметной (цикловой) комиссии общепрофессиональных дисциплин.

Протокол от «27» июня 2026 г. № 11

Председатель предметной (цикловой)
комиссии


(подпись)

О.А. Полтавец
(инициалы, фамилия)

1. Общая характеристика рабочей программы профессионального модуля

1.1. Цель и планируемые результаты освоения профессионального модуля

В результате изучения профессионального модуля студент 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	Общие компетенции
ОК 1	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 2	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях
ОК 4	Эффективно взаимодействовать и работать в коллективе и команде
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 6	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 9	Пользоваться профессиональной документацией на государственном и иностранном языках

1.1.2. Перечень профессиональных компетенций

Код	Профессиональные компетенции
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 студент должен:

иметь практический опыт	— в разработке программных модулей; — в интеграции программных модулей и компонентов; — в тестировании и отладке программных модулей; — в документировании программных модулей; — в обеспечении безопасности программного обеспечения; — в применении математического моделирования и численных методов при решении практических задач.
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; — 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. — строить простейшие математические модели; — решать задачи линейного, нелинейного и динамического программирования; — применять сетевые методы планирования и управления; — рассчитывать характеристики систем массового обслуживания; — решать игровые задачи; — разрабатывать имитационные модели; — вычислять погрешности приближённых значений;
--	---

	— решать алгебраические и трансцендентные уравнения численными методами; — решать системы линейных алгебраических уравнений; — выполнять интерполяцию и экстраполяцию функций; — выполнять численное интегрирование; — решать обыкновенные дифференциальные уравнения; — решать задачи оптимизации численными методами. — анализировать код на наличие уязвимостей; — эксплуатировать и защищать уязвимые приложения; — создавать и предотвращать межсайтовый скриптинг; — реализовывать токены и проверки Origin/Referer для защиты от CSRF; — составлять модели угроз для типового веб-приложения; — настраивать безопасную аутентификацию с JWT и refresh токенами; — реализовывать RBAC-систему с разделением привилегий; — шифровать данные с использованием AES и RSA; — хешировать пароли с salt и adaptive functions (bcrypt, Argon2); — анализировать сетевой трафик; — сканировать уязвимости; — настраивать HTTPS и создавать самоподписанные сертификаты; — защищать от brute-force атак; — безопасно работать с файлами; — реализовывать безопасную десериализацию данных; — проводить аудит логов безопасности; — настраивать CORS-политики; — защищать от DDoS-атак; — безопасно работать с памятью; — создавать безопасное API с валидацией входных данных; — реализовывать end-to-end шифрование; — настраивать TLS 1.3; — создавать secure OAuth 2.0 провайдера с PKCE; — имплементировать JWE для защищённых токенов; — разрабатывать secure password manager; — настраивать HSM-эмулятор; — реализовывать secure multi-party computation; — настраивать quantum-resistant cryptography; — разрабатывать secure API gateway; — создавать hardware-backed key storage; — имплементировать digital signature system; — настраивать certificate transparency logs;
--	---

	— разрабатывать secure session management.
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 информационно-коммуникационной 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. – понятие модели, классификация моделей; – понятие математической модели, типы математических моделей; – принципы построения математических моделей; – основные этапы математического моделирования; – каноническую задачу линейного программирования; – графический и симплексный методы решения задач линейного программирования; – транспортную задачу и задачу о назначениях; – целочисленное программирование; – основные понятия и определения нелинейного программирования; – методы решения задач нелинейного программирования; – основные понятия и определения динамического программирования; – задачи, решаемые методами динамического программирования; – основные понятия и определения теории графов; – методы нахождения кратчайшего пути; – сетевые графики и расчёт временных параметров; – марковский случайный процесс; – системы массового обслуживания: основные понятия, классификация; – схему гибели и размножения; – предмет и задачи теории игр; – основные понятия теории игр; – матричные игры, биматричные игры, игры в развёрнутой форме; – основные понятия имитационного моделирования; – примеры имитационных моделей; – методы имитационного моделирования; – инструментальные средства имитационного моделирования;
--	--

	– способы хранения чисел в памяти компьютера; – абсолютную и относительную погрешность; – верные, сомнительные, значащие цифры; – погрешности арифметических действий; – оценку погрешностей значений функции; – отделение корней; – метод половинного деления, метод простой итерации; – метод Ньютона: метод хорд, касательных; – метод Ньютона: комбинированный метод хорд и касательных; – сравнение методов вычислений по скорости сходимости итерационного процесса; – решение систем линейных алгебраических уравнений методом Гаусса; – применение метода Гаусса для вычисления определителей и нахождения обратной матрицы; – метод простой итерации, метод Зейделя; – сравнение методов вычислений по скорости сходимости итерационного процесса; – понятие интерполяции; – интерполяционный многочлен Лагранжа; – интерполяционные формулы Ньютона; – интерполяция сплайнами; – экстраполяция функций; – квадратурные формулы Ньютона-Котеса; – квадратурную формулу Гаусса; – сравнение методов численного интегрирования; – метод Эйлера, уточнённую схему Эйлера; – метод Рунге-Кутты; – сравнение методов; – методы минимизации функции одной переменной: метод дихотомии, метод золотого сечения; – методы минимизации функции двух переменных: покоординатный спуск, наискорейший спуск. – основы кибербезопасности и уязвимости ПО; – модели угроз и анализ рисков; – уязвимости веб-приложений: OWASP Top 10; – принципы безопасной аутентификации и авторизации; – криптографию для разработчиков; – принципы безопасного проектирования архитектуры; – криптографические протоколы и их реализацию; – криптографию в мобильных приложениях; – криптографию в веб-приложениях; – криптографию в облачных средах.
--	--

1.2 Количество часов, отводимое на освоение профессионального модуля

Всего часов 1044 часа, в том числе в форме практической подготовки 296 час.

Из них на освоение:

МДК 02.01 Разработка программных модулей 202 час. в том числе самостоятельная работа 40 час.

МДК 02.02 Осуществление интеграции программных модулей 184 час. в том числе самостоятельная работа 40 час.

МДК 02.03 Поддержка и тестирование программных модулей 100 час. в том числе самостоятельная работа 20 час.

МДК 02.04 Математическое моделирование 90 час. в том числе самостоятельная работа 20 час.

МДК 02.05 Численные методы 88 час. в том числе самостоятельная работа 20 час.

МДК 02.06 Безопасность программного обеспечения 110 час. в том числе самостоятельная работа 20 час.

Практики, в том числе учебная 108 час.

производственная 144 час.

Экзамен по модулю 18 час.

2. Структура и содержание профессионального модуля

2.1. Структура профессионального модуля

Коды компетенций	Наименования разделов профессионального модуля	Суммарный объем нагрузки, час.	в т.ч. в форме практической подготовки	Объем профессионального модуля, ак. час.						
				Работа студентов во взаимодействии с преподавателем						Самостоятельная работа
				Обучение по МДК				Практики		
				Всего	в том числе			Учебная	Производственная	
Промежуточная аттестация	лабораторные и практические занятия	Курсовые проекты (работы)								
1	2	3	4	5	6	7	8	9	10	11
ОК.01 – ОК.09, ПК 2.1, ПК 2.2, ПК 2.5	Раздел 1. Разработка программных модулей	202	90	144	18	90				40
ОК.01 – ОК.09, ПК 2.1, ПК 2.3, ПК 2.5	Раздел 2. Осуществление интеграции программных модулей	184	44	144		44	20			40
ОК.01 – ОК.09, ПК 2.4, ПК 2.5	Раздел 3. Поддержка и тестирование программных модулей	100	48	80		48				20
ОК.01 – ОК.09, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5	Раздел 4. Математическое моделирование	90	32	70		32				20
ОК.01 – ОК.09, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5	Раздел 5. Численные методы	88	32	68		32				20
ОК.01 – ОК.09, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5	Раздел 6. Безопасность программного обеспечения	110	50	72	18	50				20
ОК.01 – ОК.09,	Учебная практика	108						108		

ПК 2.1 – ПК 2.5										
ОК.01 – ОК.09, ПК 2.1 – ПК 2.5	Производственная практика	144							144	
	Экзамен по модулю	18								
	Всего:	1044	296	578	36	296	20	108	144	160

2.2. Тематический план и содержание профессионального модуля

Наименование разделов профессионального модуля (ПМ), междисциплинарных курсов (МДК) и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работа (проект)	Объем часов
1	2	3
Раздел 1. Разработка программных модулей		184
МДК.02.01 Разработка программных модулей		184
Тема 1.1. Использование принципов объектно-ориентированного программирования при разработке программных модулей	Содержание 1. Модульная архитектура построения приложений. Принципы. Преимущества. Примеры приложений 2. Архитектурные шаблоны, применяемые при разработке программных модулей (MVC, MVVM, MVP) 3. Инструменты разработки приложений с модульной архитектурой. Системы контроля версий. 4. Работа с библиотеками (применение стандартных библиотек, создание библиотек). Базовые принципы работы с массивами, коллекциями, строками. Работа с датой и временем. 5. 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. 6. Система ввода-вывода, средства доступа к файлам и папкам файловой системы, чтения/записи, сжатия потоков и механизмов изолированного хранения. 7. Работа со строками, регулярными выражениями, кодирование/декодирование текста. 8. Асинхронная модель программирования. Пул потоков. Шаблон асинхронного вызова методов. Синхронизация вызывающего потока. Передача и прием специальных данных состояния.	28

9. Параллельное программирование. Создание задачи. Методы ожидания выполнения задачи. Лямбда-выражения в качестве задачи. Создание продолжения задачи. Возврат значений из задачи. Отмена задачи.	
В том числе практических и лабораторных занятий	18
1. Разработка программных модулей для работы с массивами. Работа через систему контроля версий.	2
2. Разработка программных модулей для работы с коллекциями. Работа через систему контроля версий.	2
3. Разработка программных модулей для работы с датой и временем. Работа через систему контроля версий.	2
4. Разработка программных модулей с использованием паттернов проектирования. Работа через систему контроля версий.	2
5. Навигация по файловой системе. Чтение и запись файлов. Работа с потоками. Работа с изолированным хранилищем.	2
6. Работа с большими объемами текста. Кодирование и декодирование строк. Построение регулярных выражений. Чтение и запись файлов в разных кодировках.	2
7. Организация асинхронного вызова методов	2
8. Создание программного модуля, который будет выполнять методы в рамках параллельных задач	4
Самостоятельная работа	6
1. Подготовить конспект по теме «Модульная архитектура приложений: принципы, преимущества, примеры».	
2. Составить сравнительную таблицу архитектурных шаблонов MVC, MVVM, MVP с указанием области применения.	
3. Изучить систему контроля версий Git: основные команды, ветвление, слияние. Оформить шпаргалку.	
4. Разработать пример использования паттернов проектирования (Singleton, Factory, Observer) на выбранном языке программирования.	
5. Подготовить доклад на тему «Принципы SOLID и их применение при проектировании модулей».	

	6. Изучить работу с коллекциями, строками, датой и временем в выбранном языке. Составить примеры кода. 7. Разработать пример асинхронного вызова методов и обработки результатов. 8. Изучить механизмы работы с файловой системой и потоками ввода-вывода. 9. Подготовить обзор библиотек для работы с регулярными выражениями. 10. Разработать пример многопоточного приложения с синхронизацией потоков.	
Тема 1.2. Ключевые алгоритмы и структуры данных для выполнения задач программных модулей	Содержание 1. Алгоритмы и структуры данных. Оценка сложности алгоритмов. Понятие асимптотической оценки. Большие О-нотации. Временная сложность алгоритма. Пространственная сложность алгоритма. Анализ худшего, лучшего и среднего случаев. 2. Основные структуры данных (массив, связный список, стек, очередь; операции вставки, поиска и удаления; представление данных в памяти). 3. Алгоритмы сортировки и поиска. Основы рекурсии: примеры, преимущества и недостатки. 4. Хеш-таблица и хеш-функция. Коллизии и разрешение коллизий. Методы хеширования и сжатия данных. Эффективность и применение хеш-структур. 5. Деревья и графы. Представление графов и деревьев. Поиск в глубину и ширину. Минимум затратный путь (алгоритм Дейкстры). Деревья поиска и обхода. 6. Жадные алгоритмы и динамическое программирование. Основные идеи динамического программирования. 7. 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) 8. Кучи и очереди. Очереди с приоритетом и кучи. Куча и ее применение.	32

	В том числе практических и лабораторных занятий	18
	1. Оценка сложности алгоритмов	2
	2. Применение рекурсивных алгоритмов	2
	3. Работа с алгоритмами сортировки и поиска	2
	4. Создание хеш-таблиц и их использование для ускорения поиска данных	2
	5. Нахождение кратчайших путей в графах с использованием алгоритма Дейкстры	2
	6. Решение задачи о рюкзаке с использованием метода динамического программирования	4
	7. Реализация строковых алгоритмов	2
	8. Реализация приоритетных очередей для планирования задач	2
	Самостоятельная работа	8
	1. Подготовить конспект «Асимптотическая оценка сложности алгоритмов. Big O нотация».	
	2. Составить таблицу основных структур данных (массив, список, стек, очередь, дерево, граф) с указанием операций и сложности.	
	3. Разработать примеры алгоритмов сортировки (пузырьковая, быстрая, слиянием) с анализом сложности.	
	4. Изучить рекурсию: примеры, преимущества и недостатки. Оформить конспект.	
	5. Разработать пример хеш-таблицы и обработки коллизий.	
	6. Изучить алгоритмы поиска в глубину и ширину на графах. Составить примеры.	
	7. Подготовить доклад на тему «Динамическое программирование: задача о рюкзаке, редакционное расстояние».	
	8. Разработать пример реализации алгоритма Дейкстры для поиска кратчайшего пути.	
	9. Изучить строковые алгоритмы (Кнута-Морриса-Пратта, Бойера-Мура, Рабина-Карпа).	
	10. Разработать пример использования приоритетной очереди и кучи.	
Тема 1.3. Проектирование	Содержание	28

модулей	1. Основные принципы проектирования модулей программного обеспечения. Методы анализа требований и способов определения функциональности модуля. Методы анализа и оптимизации проектируемых модулей для повышения их эффективности и качества. Декомпозиция задачи на подзадачи. Создание спецификаций модуля.	
	2. Принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей	
	3. Принципы проектирования классов. Проектирование классов с учётом инкапсуляции. Использование наследования: создание иерархий классов. Полиморфизм: перегрузка методов и интерфейсов.	
	4. Применение диаграмм классов при проектировании требований к внутренней структуре программного модуля.	
	5. Применение диаграмм компонентов для визуализации организации компонентов проектируемого модуля	
	В том числе практических и лабораторных занятий	18
	1. Анализ требований к модулю и определение его функциональности	4
	2. Создание спецификации программного модуля	2
	3. Проектирование требований к внутренней структуре программного модуля средствами диаграмм классов. Применение паттернов проектирования	4
	4. Проектирование требований к организации компонентов модуля средствами диаграммы компонентов	2
	5. Проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами	4
	6. Анализ и оптимизация проектируемого модуля для повышения его эффективности и качества	2
	Самостоятельная работа	
	1. Подготовить конспект «Основные принципы проектирования модулей ПО».	8
	2. Изучить методы анализа требований и определения функциональности модуля. Оформить пример.	
	3. Разработать спецификацию программного модуля для учебного проекта.	

	4. Изучить принципы декомпозиции задачи на подзадачи. Составить пример. 5. Подготовить доклад на тему «Принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей». 6. Разработать диаграмму классов для учебного модуля. 7. Изучить применение диаграмм компонентов для визуализации структуры модуля. 8. Разработать пример проектирования интерфейсов модуля для взаимодействия с другими модулями. 9. Изучить принципы инкапсуляции, наследования и полиморфизма на практических примерах. 10. Подготовить анализ и оптимизацию проектируемого модуля для повышения эффективности и качества.	
Тема 1.4. Создание программных модулей для взаимодействия с пользователем	Содержание 1. Виды пользовательского интерфейса (командная строка, графический, речевой). Основные этапы и принципы разработки графического пользовательского интерфейса. 2. Технологии и инструменты разработки графического пользовательского интерфейса. 3. Компоненты графического пользовательского интерфейса. Типы элементов управления. компоновка элементов управления. События. Обработчики событий. 4. Работа с окнами. Основные методы работы с окнами. Создание окна: функции и классы. Открытие и закрытие окон. Взаимодействие с окнами (например, передача данных). Примеры валидации (проверка формата ввода). Сообщения об ошибках и уведомления пользователя. Использование регулярных выражений для валидации. 5. Многопоточность и асинхронная работа окон. Многопоточность в GUI-приложениях. Проблемы синхронизации потоков. Использование асинхронных вызовов для долго выполняемых операций. 6. Значение стиля в UX/UI дизайне. Основы теории цвета. Работа с цветом и шрифтами. Стилизация.	34

	7. Работа с текстом, изображениями. Построение графиков и диаграмм. Библиотеки для построения графиков и диаграмм. Работа с мультимедиа	
	В том числе практических и лабораторных занятий	20
	1. Проектирование главного окна приложения с несколькими панелями и элементами управления.	2
	2. Разработка модулей многооконного приложения	2
	3. Разработка стилей для приложения для улучшения взаимодействия с пользователем	2
	4. Разработка модулей для представления текстовой информации	2
	5. Разработка модулей для работы с изображениями	2
	6. Разработка модулей для представления информации в виде графиков и диаграмм	2
	7. Разработка модулей для работы аудио и видео	2
	8. Реализация загрузки данных из интернета в фоновом режиме без блокировки основного потока приложения.	2
	9. Разработка формы регистрации с элементами ввода и проверкой корректности введенных данных.	4
	Самостоятельная работа	6
	1. Подготовить конспект «Виды пользовательского интерфейса: командная строка, графический, речевой».	
	2. Изучить основные этапы и принципы разработки графического интерфейса. Оформить обзор.	
	3. Разработать пример простого GUI-приложения с несколькими элементами управления.	
	4. Изучить компоненты графического интерфейса и обработчики событий.	
	5. Подготовить доклад на тему «UX/UI дизайн: теория цвета, типографика, композиция».	
	6. Разработать пример многооконного приложения с передачей данных между окнами.	
	7. Изучить методы валидации пользовательского ввода с использованием регулярных выражений.	

	8. Разработать пример работы с изображениями и построения графиков в приложении. 9. Изучить многопоточность и асинхронную работу в GUI-приложениях. 10. Разработать пример стилизации приложения для улучшения взаимодействия с пользователем.	
Тема 1.5. Создание модулей для взаимодействия с базами данных	Содержание	
	1. Взаимодействие приложения с базой данных. Технологии доступа к данным. Безопасность при работе с базами данных. 2. Понятие и преимущества ORM. Концепцией объектно-реляционного отображения и использование ORM-библиотек. Применение ORM для работы с базами данных. 3. Реализация CRUD-операций в приложении. Выполнение запросов к базе данных.	12
	В том числе практических и лабораторных занятий	6
	1. Разработка программных модулей для работы с базами данных	4
	2. Разработка программных модулей для работы с запросами к базе данных	2
	Самостоятельная работа	
	1. Подготовить конспект «Технологии доступа к данным: ODBC, JDBC, ADO.NET, PDO». 2. Изучить понятие ORM и преимущества объектно-реляционного отображения. Оформить обзор. 3. Разработать пример CRUD-операций для учебного модуля. 4. Изучить применение ORM-библиотек (Hibernate, Entity Framework, SQLAlchemy, Eloquent). 5. Подготовить доклад на тему «Безопасность при работе с базами данных: SQL-инъекции и защита». 6. Разработать пример выполнения параметризованных запросов. 7. Изучить механизмы транзакций при работе с БД из программного модуля. 8. Разработать пример работы с хранимыми процедурами из кода приложения. 9. Изучить методы кэширования данных при взаимодействии с БД.	6

	10. Подготовить обзор инструментов миграции баз данных (Flyway, Liquibase).	
Тема 1.6 Принципы безопасности, производительности и масштабируемости программных модулей	Содержание	14
	1. Основные понятия: безопасность программного обеспечения, производительность модулей, масштабируемость архитектуры. Методы обеспечения безопасности. Факторы, влияющие на производительность. Техники повышения производительности программного обеспечения Масштабируемость: горизонтальная и вертикальная масштабируемость; принципы проектирования для масштабируемости; использование облачных технологий для масштабирования. Метрики безопасности (например, количество уязвимостей). Инструменты для мониторинга производительности. Подходы к нагрузочному тестированию. 2. Понятие оптимизации кода. Основные цели оптимизации: повышение скорости выполнения, снижение потребления памяти, улучшение читаемости и поддержки кода. Методы улучшения алгоритмов. Профилирование и отладка производительности. Специфичные методы оптимизации для разных языков программирования.	
	В том числе практических и лабораторных занятий	10
	1. Оптимизация проектируемых модулей для повышения их эффективности и качества	2
	2. Решение задач на оптимизацию алгоритмов	2
	3. Анализ и мониторинг производительности приложений. Обеспечение производительности и масштабируемости при разработке модулей программного обеспечения	2
	4. Улучшение производительности модулей посредством выявления и устранения узких мест	2
	5. Обеспечение безопасности при разработке модулей программного обеспечения	2
	Самостоятельная работа	6
	1. Подготовить конспект «Основные понятия: безопасность ПО, производительность, масштабируемость».	

	2. Изучить методы обеспечения безопасности программных модулей. Оформить таблицу угроз и мер защиты. 3. Разработать пример оптимизации кода для повышения производительности. 4. Изучить методы профилирования и отладки производительности. Оформить обзор инструментов. 5. Подготовить доклад на тему «Горизонтальная и вертикальная масштабируемость: принципы и примеры». 6. Разработать пример использования кэширования для повышения производительности. 7. Изучить методы оптимизации алгоритмов и структур данных. 8. Разработать пример мониторинга производительности приложения. 9. Изучить принципы обеспечения безопасности при разработке модулей (OWASP Top 10). 10. Подготовить анализ узких мест приложения и предложения по их устранению.	
Промежуточная аттестация в форме экзамена		18
Всего		202
Раздел 2. Осуществление интеграции программных модулей		164
МДК.02.02 Осуществление интеграции программных модулей		162
Тема 2.1. Основы интеграции программных модулей	Содержание 1. Разработка REST API. Клиент-серверное взаимодействие. Особенности передачи информации по HTTP протоколу. Структура HTTP запроса. HTTP методы: GET, POST, DELETE, PUT, PATCH. HTTP заголовки. Тело запроса. 2. Маршрутизация запросов. Группировка маршрутов. Статические ресурсы. 3. Обработка запросов пользователя. Path, Query параметры. Обработка содержимого body: raw, objects, forms, multipart. Валидация данных. 4. Формирование и отправка ответов: object, file. Параметры ответов: статус код, тип содержимого, заголовки, cookies. Перенаправления. Сериализация/десериализация объектов. 5. Создание и управление фоновыми задачами.	70

6. Аутентификация и авторизация. OAuth, JWT, forms. Сессии. Ролевое разграничение доступа к ресурсам.	
7. Разработка WebSocket API. Взаимодействие клиента и сервера по WebSocket протоколу. Настройки соединения. Открытие и закрытие соединения. Передача сообщения серверу.	
8. Разработка микросервисов. Микросервисная и монолитная архитектура.	
9. Синхронное (REST, gRPC) и асинхронное (брокеры сообщений) взаимодействие между микросервисами.	
В том числе практических и лабораторных занятий	30
1. Создание клиентского приложения для работы с публичным API	4
2. Создание REST API приложения с реализацией: добавления, удаления, изменения и создания данных (от 3 - 4 сущностей)	2
3. Расширение функционала REST API приложения: работа с удаленным источником данных	2
4. Расширение функционала REST API приложения: работа со статическими изображениями (ресурсами) - загрузка, передача, удаление.	2
5. Расширение функционала REST API приложения: обработка path и query параметров	2
6. Расширение функционала REST API приложения: обработка ошибок, передача сообщений об ошибке пользователю	2
7. Расширение функционала REST API приложения: валидация полученных данных	2
8. Расширение функционала REST API приложения: добавление фоновых задач	2
9. Расширение функционала REST API приложения: добавление аутентификации и авторизации, создание ролевой системы	2
10. Создание клиентского приложения для работы с публичным WebSocket.	2
11. Создание серверного приложения для работы по websocket протоколу.	2
12. Создание микросервисного приложения с взаимодействием по REST	2
13. Создание микросервисного приложения с взаимодействием по gRPC	2
14. Создание микросервисного приложения с взаимодействием через брокера приложений (consumer, producer)	2
Самостоятельная работа	10

	1. Подготовить конспект «Клиент-серверное взаимодействие: основные принципы». 2. Изучить структуру HTTP-запроса и ответа. Оформить примеры. 3. Разработать пример REST API с реализацией CRUD-операций. 4. Изучить методы аутентификации и авторизации (OAuth, JWT, сессии). 5. Подготовить доклад на тему «Микросервисная и монолитная архитектура: сравнение». 6. Разработать пример взаимодействия микросервисов по REST. 7. Изучить протокол WebSocket и его применение для real-time приложений. 8. Разработать пример работы с брокером сообщений (RabbitMQ, Kafka). 9. Изучить методы сериализации и десериализации данных (JSON, XML, Protobuf). 10. Подготовить обзор инструментов документирования API (Swagger, OpenAPI).	
Тема 2.2. Управление и мониторинг интегрированной системы	Содержание	
	1. Настройка конфигурации и сборки приложения. 2. Логирование событий. Конфигурация логирования. Уровни логирования. Логирование в файлы различного формат. 3. Мониторинг приложения: нагрузка, ошибки, сбор статистики. Внедрение сборщика метрик. 4. Инструменты контейнеризации. Контейнеризация приложения. Средства доставки и средства развертывания решения.	22
	В том числе практических и лабораторных занятий	6
	1. Настроить конфигурацию rest api приложения (порт, хост, данные для подключения к источнику данных, приватные ключи).	2
	2. Внедрить логирование в rest api приложение.	2
	3. Упаковка rest api приложения в контейнер и доставка на другое устройство	2
	Самостоятельная работа	
	1. Подготовить конспект «Настройка конфигурации и сборки приложения». 2. Изучить уровни логирования и форматы логов. Оформить примеры.	10

	3. Разработать пример настройки логирования в приложении. 4. Изучить инструменты мониторинга приложений (Prometheus, Grafana, Zabbix). 5. Подготовить доклад на тему «Контейнеризация приложений: Docker и Kubernetes». 6. Разработать пример Dockerfile для учебного приложения. 7. Изучить методы сбора метрик и настройки оповещений. 8. Разработать пример контейнеризации и доставки приложения на другое устройство. 9. Изучить инструменты сбора статистики и анализа нагрузки. 10. Подготовить обзор средств доставки и развертывания решений (CI/CD).	
Тема 2.3. Безопасность при интеграции	Содержание	
	1. Протоколы с использованием безопасного соединения: HTTPS, WSS (WebSocket Secure). 2. Предотвращение угроз безопасности: SQL инъекции, CSRF, XSS. Хеширование чувствительных данных, применение алгоритмов хеширования паролей с солью. 3. Анализ уязвимостей. Регулярные аудиты безопасности. Применение лучших практик защиты информации.	16
	В том числе практических и лабораторных занятий	4
	1. Добавление SSL сертификата в приложение	2
	2. Настройка конфигурации безопасности приложения	2
	Самостоятельная работа	
	1. Подготовить конспект «Протоколы с использованием безопасного соединения: HTTPS, WSS». 2. Изучить угрозы безопасности при интеграции: SQL-инъекции, CSRF, XSS. Оформить таблицу. 3. Разработать пример хеширования паролей с солью. 4. Изучить методы анализа уязвимостей. Оформить обзор инструментов. 5. Подготовить доклад на тему «Регулярные аудиты безопасности: цели и методы». 6. Разработать пример настройки SSL-сертификата для приложения.	10

	7. Изучить принципы безопасной передачи данных между модулями. 8. Разработать пример защиты API с использованием токенов. 9. Изучить методы предотвращения атак типа «человек посередине». 10. Подготовить чек-лист безопасности для интегрированной системы.	
Тема 2.4. Оптимизация и масштабируемость интегрированных решений	Содержание	16
	1. Масштабирование интегрированных решений. Горизонтальное и вертикальное масштабирование. 2. Оптимизации производительности. Кэширование данных. Оптимизация запросов к базам данных. 3. Профилирование кода. Уменьшение времени отклика.	
	В том числе практических и лабораторных занятий	
	1. Реализация кэширования данных в rest api приложение	2
	2. Оптимизация производительности rest api через профилирование	2
	Самостоятельная работа	10
	1. Подготовить конспект «Масштабирование интегрированных решений: горизонтальное и вертикальное». 2. Изучить методы оптимизации производительности интегрированных систем. Оформить обзор. 3. Разработать пример кэширования данных в REST API. 4. Изучить профилирование кода и методы уменьшения времени отклика. 5. Подготовить доклад на тему «Оптимизация запросов к базам данных в интегрированных системах». 6. Разработать пример оптимизации производительности API через профилирование. 7. Изучить методы балансировки нагрузки в распределённых системах. 8. Разработать пример масштабирования приложения с использованием контейнеров. 9. Изучить принципы отказоустойчивости и восстановления после сбоев. 10. Подготовить анализ узких мест интегрированной системы и предложения по оптимизации.	
Курсовой проект		20
Примерная тематика курсовых работ (проектов)		

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. Создания 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		
Промежуточная аттестация в форме дифференцированного зачета		2
Всего		184
Раздел 3. Поддержка и тестирование программных модулей		100
МДК.02.03 Поддержка и тестирование программных модулей		98
Тема 3.1. Качество программного обеспечения	Содержание	14
	Определение качества программного модуля. Метрики качества программных модулей (статические метрики: количество строк кода, цикломатическая сложность, коэффициент связности и сцепленной: динамические метрики: покрытие кода тестами, частота отказов, время отклика). Принципы проектирования качественных модулей. Стандарты и модели качества программных модулей. Применение моделей качества. Инструменты для оценки качества. Практические аспекты повышения качества.	
	В том числе практических и лабораторных занятий	6
	1. Анализ и оценка качества программного модуля с использованием метрик качества программных модулей	2

	2. Использование статического анализа кода для выявления дефектов	2
	3. Разработка и применение процессов обеспечения качества в жизненном цикле разработки программных модулей	2
	Самостоятельная работа	4
	1. Подготовить конспект «Определение качества программного модуля». 2. Изучить метрики качества ПО: статические и динамические. Оформить таблицу. 3. Разработать пример анализа качества модуля с использованием метрик.	
Тема 3.2. Отладка программного модуля	Содержание	10
	1. 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). 2. Стратегии поиска ошибок (метод половинного деления, метод исключения, проверка граничных условий, поиск паттернов повторяющихся ошибок). 3. Документирование процесса отладки.	
	В том числе практических и лабораторных занятий	
	1. Разработка стратегии отладки и исправление ошибок в программном обеспечении	
	2. Код-ревью и парное программирование	2
	Самостоятельная работа	4
	1. Подготовить конспект «Понятие отладки. Ошибки, дефекты, сбои, отказы». 2. Изучить типы ошибок в программных модулях. Оформить таблицу. 3. Разработать стратегию отладки для учебного модуля. 4. Изучить инструменты отладки (точки останова, пошаговое выполнение, просмотр переменных).	
Тема 3.3. Обработка исключений	Содержание	12
	1. Понятие исключения. Типы исключений. Механизм обработки исключений. Логика работы с исключениями. Методы отладки кода с использованием исключений и логирования.	

	В том числе практических и лабораторных занятий	6
	1. Основные конструкции для обработки исключительных ситуаций	2
	2. Практическое использование исключений в реальной задаче	2
	3. Обработка ошибок и исключение в RESTful API	2
	Самостоятельная работа	6
	1. Подготовить конспект «Понятие исключения. Типы исключений».	
	2. Изучить механизм обработки исключений в выбранном языке программирования.	
	3. Разработать пример использования конструкций try-catch-finally.	
	4. Изучить логику работы с исключениями и логированием. Оформить пример.	
	5. Подготовить доклад на тему «Обработка ошибок в RESTful API».	
	6. Разработать пример практического использования исключений в реальной задаче.	
	7. Изучить методы отладки кода с использованием исключений.	
	8. Разработать пример генерации пользовательских исключений.	
	9. Изучить принципы обработки ошибок на разных уровнях приложения.	
	10. Подготовить обзор антипаттернов обработки исключений.	
Тема 3.4. Тестирование программных модулей	Содержание	30
	1. Понятие процесса тестирования программного обеспечения. Этапы процесса тестирования программного обеспечения. техники ручного тестирования и автоматизированного тестирования	
	2. Модель работы с дефектами. Принципы работы в системе контроля дефектов.	
	3. Виды тестирования (функциональное тестирование, нефункциональное тестирование, статическое и динамическое тестирование).	
	4. Типы тестирования (модульное тестирование, интеграционное тестирование, системное тестирование, приемочное тестирование, нагрузочное тестирование, стресс-тестирование)	
	5. Тестирование по белому ящику. Метод покрытия операторов. Метод покрытия условий.	

	6. Тестирование по белому ящику. Метод комбинаторного покрытия условий.	
	7. Тестирование по черному ящику. Метод классов эквивалентности.	
	8. Тестирование по черному ящику. Метод граничных значений.	
	9. Модульные тесты. Тестирование интеграции. Методы и инструменты для тестирования интегрированных решений.	
	В том числе практических и лабораторных занятий	24
	1. Анализ требований к программному обеспечению и составление планов тестирования. Использование систем контроля дефектов программного обеспечения	2
	2. Тестирование методами белого ящика. Метод покрытия операторов. Метод покрытия условий.	2
	3. Тестирование методами белого ящика. Метод комбинаторного покрытия условий.	2
	4. Тестирование по черному ящику. Метод классов эквивалентности. Тестирование по черному ящику. Метод граничных значений.	2
	5. Тестирование по черному ящику. Анализ причинно-следственных связей.	2
	6. Разработка модульных тестов.	2
	7. Разработка модульных тестов с проверкой результатов тестирования с учетом погрешности.	2
	8. Разработка модульных тестов для отдельно компилируемых модулей.	2
	9. Разработка модульных тестов для проверки коллекций.	2
	10. Тестирование интеграции. Написание и выполнение тестов для проверки взаимодействия между модулями	2
	11. Тестирование RESTful API. Тестирование производительности.	2
	12. Разработка через тестирование.	2
	Самостоятельная работа	6
	1. Подготовить конспект «Понятие процесса тестирования ПО. Этапы тестирования».	
	2. Изучить виды и типы тестирования. Оформить сравнительную таблицу.	
	3. Разработать тест-кейсы для учебного модуля.	

	4. Изучить техники тестирования по белому и чёрному ящику. Оформить примеры. 5. Подготовить доклад на тему «Модульное тестирование: цели и инструменты». 6. Разработать пример модульных тестов с использованием фреймворка. 7. Изучить методы тестирования интеграции модулей. 8. Разработать пример тестирования RESTful API. 9. Изучить принципы разработки через тестирование (TDD). 10. Подготовить обзор инструментов автоматизированного тестирования.	
Тема 3.5. Поддержка программных модулей	Содержание	14
	1. Работы, выполняемые при поддержке программного обеспечения. Исправление дефектов. Ревьюирование кода. Рефакторинг кода. Оптимизация кода. 2. Стандарты разработки и оформления документации на программное обеспечение. Принципы документирования программного обеспечения. Инструменты для создания технической документации и комментирования кода 3. 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	
	В том числе практических и лабораторных занятий	8
	1. Разработка документации на программное обеспечение в соответствии со стандартами. Ведение журнала изменений и фиксация обновления программных модулей.	2
	2. Ревьюирование, рефакторинг и оптимизация кода.	2
	3. Разработка Программы и методики испытаний.	2
	4. Создание спецификаций API	2
Промежуточная аттестация в форме дифференцированного зачета		2
Всего		100
Раздел 4. Математическое моделирование		90
МДК.02.04 Математическое моделирование		88
	Содержание	6

Тема 4.1. Математическое моделирование как методология решения практических задач	1. Понятие модели. Классификация моделей. Понятие математической модели. Типы математических моделей. Принципы построения математических моделей. Основные этапы математического моделирования.	
	В том числе практических и лабораторных занятий	2
	1. Построение простейших математических моделей	2
	Самостоятельная работа	2
Тема 4.2. Линейное программирование	Содержание	14
	1. Каноническая задача линейного программирования. Основные определения. Графический метод решения задач линейного программирования. Симплексный метод решения задач линейного программирования. Транспортная задача. Задача о назначениях. Целочисленное программирование.	
	В том числе практических и лабораторных занятий	
	1. Решение задач линейного программирования симплексным методом	
	2. Решение транспортной задачи	2
	3. Решение задачи о назначениях	2
	4. Применение инструментальных средств для решения задач линейного программирования	2
	Самостоятельная работа	2
Тема 4.3. Нелинейное программирование	Содержание	10
	1. Основные понятия и определения нелинейного программирования. 2. Методы решения задач нелинейного программирования.	
	В том числе практических и лабораторных занятий	4
	1. Решение задач нелинейного программирования	4
	Самостоятельная работа	2
Тема 4.4. Динамическое программирование	Содержание	10
	1. Основные понятия и определения динамического программирования. Задачи, решаемые методами динамического программирования:	
	В том числе практических и лабораторных занятий	4
	1. Решение задач оптимального распределения ресурсов, о замене оборудования	2
	2. Решение задач определения оптимального пути, оптимального резервирования	2

	Самостоятельная работа	2
Тема 4.5. Сетевые методы планирования и управления	Содержание	6
	1. Основные понятия и определения теории графов. Нахождение кратчайшего пути. Дерево решений. Сетевые графики. Расчет временных параметров.	
	В том числе практических и лабораторных занятий	2
	1. Решение задач на применение методов сетевого планирования	2
	Самостоятельная работа	2
Тема 4.8. Системы массового обслуживания	Содержание	6
	1. Марковский случайный процесс. Системы массового обслуживания: основные понятия, классификация. Схема гибели и размножения	
	В том числе практических и лабораторных занятий	2
	1. Расчет характеристик простейших систем массового обслуживания	2
	Самостоятельная работа	2
Тема 4.9. Теория игр	Содержание	8
	1. Предмет и задачи теории игр. Основные понятия теории игр. Матричные игры. Биматричные игры. Игры в развернутой форме	
	В том числе практических и лабораторных занятий	4
	1. Решение игровых задач с нулевой суммой.	2
	2. Решение задач в развернутой форме	2
	Самостоятельная работа	2
Тема 4.10. Имитационное моделирование	Содержание	10
	1. Основные понятия имитационного моделирования. Примеры имитационных моделей. Методы имитационного моделирования. Инструментальные средства имитационного моделирования	
	В том числе практических и лабораторных занятий	6
	1. Разработка простейшей имитационной модели	4
	2. Решение задач массового обслуживания методами имитационного моделирования	2
	Самостоятельная работа	2
Промежуточная аттестация в форме дифференцированного зачета		2
Всего		90

Раздел 5. Численные методы		88
МДК.02.05 Численные методы		86
Тема 5.1. Приближенные числа и действия над ними	Содержание	10
	1. Способы хранения чисел в памяти компьютера. Абсолютная погрешность, относительная погрешность.	
	2. Верные, сомнительные, значащие цифры.	
	3. Погрешности арифметических действий. Оценка погрешностей значений функции	
	В том числе практических и лабораторных занятий	2
	1. Вычисление погрешностей приближенных значений. Вычисление погрешностей результатов арифметических действий.	2
Тема 5.2. Численные методы решения алгебраических и трансцендентных уравнений	Самостоятельная работа	4
	Содержание	12
	1. Отделение корней. Метод половинного деления. Метод простой итерации.	
	2. Методы Ньютона: метод хорд, касательных.	
	3. Методы Ньютона: комбинированный метод хорд и касательных. Сравнение методов вычислений по скорости сходимости итерационного процесса	
	В том числе практических и лабораторных занятий	6
	1. Решение алгебраических и трансцендентных уравнений приближенными методами (метод половинного деления, метод простых итераций)	2
	2. Решение алгебраических и трансцендентных уравнений приближенными методами (методы Ньютона)	2
	3. Мониторинг и анализ производительности разработанных приложений для численного решения уравнений.	2
Тема 5.3. Численные методы решения систем линейных алгебраических уравнений	Самостоятельная работа	4
	Содержание	10
	1. Решение систем линейных алгебраических уравнений методом Гаусса. Применение метода Гаусса для вычисления определителей и нахождения обратной матрицы.	

	2. Метод простой итераций. Метод Зейделя. Сравнение методов вычислений по скорости сходимости итерационного процесса.	
	В том числе практических и лабораторных занятий	6
	1. Решение систем линейных алгебраических уравнений методом Гаусса. Вычисление определителя. Нахождение обратной матрицы	2
	2. Решение систем линейных алгебраических уравнений методом простой итерации, методом Зейделя	2
	3. Мониторинг и анализ производительности разработанных приложений для численного решения систем линейных алгебраических уравнений.	2
	Самостоятельная работа	2
Тема 5.4. Интерполяция и экстраполяция функций	Содержание	10
	1. Понятие интерполяции. Интерполяционный многочлен Лагранжа. Интерполяционные формулы Ньютона.	
	2. Интерполяция сплайнами. 3. Экстраполяция функций.	
	В том числе практических и лабораторных занятий	4
	1. Составление интерполяционных формул Лагранжа и Ньютона. Интерполяция сплайнами.	2
	2. Экстраполирование функций	2
	Самостоятельная работа	2
Тема 5.5. Численное интегрирование	Содержание	8
	1. Квадратурные формулы Ньютона-Котеса. 2. Квадратурная формула Гаусса. Сравнение методов численного интегрирования	
	В том числе практических и лабораторных занятий	
	1. Вычисление интегралов при помощи формул Ньютона – Котеса	2
	2. Вычисление интегралов при помощи формул Гаусса.	2
	Самостоятельная работа	4
Тема 5.6. Численные методы решения обыкновенных дифференциальных уравнений	Содержание	8
	1. Метод Эйлера. Уточненная схема Эйлера. 2. Метод Рунге – Кутты. Сравнение методов.	
	В том числе практических и лабораторных занятий	4

	1. Нахождение решений обыкновенных дифференциальных уравнений при помощи формул Эйлера.	2
	2. Нахождение решений обыкновенных дифференциальных уравнений методом Рунге – Кутта.	2
	Самостоятельная работа	2
Тема 5.7 Численное решение задач оптимизации	Содержание	8
	1. Методы минимизации функции одной переменной: метод дихотомии, метод золотого сечения. Методы минимизации функции двух переменных: покоординатный спуск, наискорейший спуск	
	В том числе практических и лабораторных занятий	6
	1. Нахождение экстремумов функций одной переменной приближенными методами	2
	2. Нахождение экстремумов функций двух переменных приближенными методами	4
	Самостоятельная работа	2
Промежуточная аттестация в форме дифференцированного зачета		2
Всего		88
Раздел 6. Безопасность программного обеспечения		92
МДК.02.06 Безопасность программного обеспечения		92
Тема 6.1. Основы безопасности программного обеспечения	Содержание	40
	1. Введение в кибербезопасность и уязвимости ПО.	
	2. Модели угроз и анализ рисков.	
	3. Уязвимости веб-приложений: OWASP Top 10.	
	4. Безопасная аутентификация и авторизация.	
	5. Криптография для разработчиков.	
	В том числе практических и лабораторных занятий	28
	1. Анализ кода на наличие уязвимостей - ручной review 1000 строк кода	2
	2. SQL инъекции - эксплуатация и защита уязвимого приложения	2
	3. XSS атаки - создание и предотвращение межсайтового скриптинга. CSRF защита - реализация токенов и проверки Origin/Referer	2
	4. Составление модели угроз для типового веб-приложения	2
	5. Настройка безопасной аутентификации с JWT и refresh токенами	2

	6. Реализация RBAC системы с разделением привилегий. Шифрование данных с использованием AES и RSA	2
	7. Хэширование паролей с salt и adaptive functions (bcrypt, Argon2). Анализ сетевого трафика с помощью Wireshark	2
	8. Сканирование уязвимостей OWASP ZAP и Burp Suite	2
	9. Настройка HTTPS и создание самоподписанных сертификатов	2
	10. Защита от brute-force атак с ограничением попыток входа	2
	11. Безопасная работа с файлами. Реализация безопасной десериализации данных	2
	12. Аудит логов безопасности и выявление подозрительной активности. Настройка CORS политик для веб-приложений. Защита от DDOS атак с помощью rate limiting	2
	13. Безопасная работа с памятью в приложениях	2
	14. Создание безопасного API с валидацией всех входных данных	2
	Самостоятельная работа	2
Тема 6.2. Разработка безопасного ПО и прикладная криптография	Содержание	36
	1. Принципы безопасного проектирования архитектуры.	
	2. Криптографические протоколы и их реализация.	
	3. Криптография в мобильных приложениях.	
	4. Криптография в веб-приложениях.	
	5. Криптография в облачных средах.	
	В том числе практических и лабораторных занятий	26
	1. Реализация end-to-end шифрования для мессенджера на Signal Protocol	2
	2. Настройка TLS 1.3 с perfect forward secrecy и современными cipher suites	2
	3. Создание secure OAuth 2.0 провайдера с PKCE и защитой от атак. Имплементация JWE (JSON Web Encryption) для защищённых токенов	2
	4. Разработка безопасного voting system с homomorphic encryption. Создание cryptocurrency wallet с ECDSA и hierarchical deterministic keys	2
	5. Создание cryptocurrency wallet с ECDSA и hierarchical deterministic keys	2
	6. Реализация secure password manager с client-side encryption. Настройка HSM эмулятора для аппаратной защиты ключей. Разработка secure file storage с encryption at rest и in transit	2

	7. Имплементация zero-knowledge proof для аутентификации без пароля. Создание blockchain smart contract с защитой от reentrancy attacks	2
	8. Реализация secure multi-party computation для совместных вычислений	2
	9. Настройка quantum-resistant cryptography с lattice-based алгоритмами. Разработка secure API gateway с JWT verification и rate limiting	2
	10. Создание hardware-backed key storage для мобильного приложения. Имплементация digital signature system с timestamping	2
	11. Имплементация digital signature system с timestamping	2
	12. Настройка certificate transparency logs для мониторинга SSL сертификатов. Разработка secure session management с защитой от hijacking	2
	13. Создание cryptographically secure RNG (random number generator)	2
	Самостоятельная работа	10
Промежуточная аттестация в форме экзамена		18
Всего		110
Учебная практика		108
Виды работ:		
1. Проектирование модулей программного обеспечения с учетом технического задания		
2. Визуализации и описания архитектурных решений		
3. Определение интерфейсов и взаимодействия модулей в системе		
4. Создание модулей программного обеспечения		
5. Работа с API и веб-сервисами для взаимодействия между модулями		
6. Работа с интеграционными платформами и инструментами		
7. Отладка программного обеспечения на уровне программных модулей		
8. Тестирование программного обеспечения		
9. Формирование тестовых сценариев		
10. Подготовка тестовых платформ (установка операционной системы, дополнительного программного обеспечения и другого по необходимости)		
11. Оценка объема тестирования программного обеспечения с целью определения необходимых ресурсов для его выполнения		
12. Формирование и представление отчетности о подготовке к выполнению задания на тестирование программного обеспечения в соответствии с установленными регламентами		
13. Выполнение тестовых процедур на тестовых данных		

14. Создание технической документации для модулей	
15. Документирование кода, API и интерфейсов	
16. Работа со специализированным программным обеспечением по документированию программного кода	
Производственная практика Виды работ: 1. Проектирование модулей программного обеспечения с учетом технического задания 2. Визуализации и описания архитектурных решений 3. Определение интерфейсов и взаимодействия модулей в системе 4. Создание модулей программного обеспечения 5. Оптимизация кода и алгоритмов программных модулей для увеличения производительности 6. Мониторинг и анализ производительности приложений 7. Интеграция программных модулей и компонентов в единое программное решение 8. Работа с API и веб-сервисами для взаимодействия между модулями 9. Работа с интеграционными платформами и инструментами 10. Обеспечение совместимости и стабильности системы 11. Отладка программного обеспечения на уровне программных модулей 12. Тестирование программного обеспечения 13. Формирование тестовых сценариев 14. Подготовка тестовых платформ (установка операционной системы, дополнительного программного обеспечения и другого по необходимости) 15. Оценка объема тестирования программного обеспечения с целью определения необходимых ресурсов для его выполнения 16. Настройки тестовой среды и аппаратных средств для выполнения тестирования программного обеспечения в соответствии с заданием на тестирование в пределах своей компетенции 17. Формирование и представление отчетности о подготовке к выполнению задания на тестирование программного обеспечения в соответствии с установленными регламентами 18. Выполнение тестовых процедур на тестовых данных 19. Создание технической документации для модулей 20. Документирование кода, API и интерфейсов 21. Работа со специализированным программным обеспечением по документированию программного кода	144
Промежуточная аттестация в форме экзамена по модулю	18
Всего	1044

3. Условия реализации рабочей программы профессионального модуля

3.1 Для реализации программы дисциплины должны быть предусмотрены следующие специальные помещения:

Лаборатория «Разработки и интеграции программных решений», оснащенная:

– Мебель:

- 1) столы компьютерные (регулируемые по высоте);
- 2) стулья регулируемые;
- 3) стол учителя с ящиками для хранения или тумбой;
- 4) кресло учителя;
- 5) шкаф для хранения учебных пособий.

– Оборудование и технические средства:

- 1) доска магнитно-маркерная;
- 2) демонстрационный экран;
- 3) компьютер учителя с периферией/ноутбук с дополнительным монитором, подключением к сети Интернет и локальной сети (лицензионное программное обеспечение, образовательный контент, система защиты от вредоносной информации);

4) компьютер ученика с периферией/ноутбук с дополнительным монитором, подключением к сети Интернет и локальной сети (лицензионное программное обеспечение, образовательный контент, система защиты от вредоносной информации).

3.1.1 Перечень лицензионного и свободно распространяемого программного обеспечения

Освоение программы профессионального модуля предполагает использование следующего программного обеспечения:

№ п/п	Наименование программного обеспечения	Назначение
1	Пакеты приложений для работы с текстовыми документами, таблицами и графическими изображениями (или их аналоги): — Microsoft Office (Word, Excel, PowerPoint) / LibreOffice / Р7-Офис	Оформление проектной и технической документации по БД, подготовка отчётов по практикам, курсовому проекту, ведение таблиц с результатами тестирования и оптимизации запросов
2	Интернет-браузеры (не менее трёх различных): — Google Chrome, Mozilla Firefox, Microsoft Edge, Яндекс.Браузер	Тестирование веб-интерфейсов для работы с базой данных (phpMyAdmin, pgAdmin в веб-режиме, собственные админ-панели), проверка корректности отображения данных из БД
3	Текстовый редактор с подсветкой синтаксиса кода: — Visual Studio Code, Notepad++, Sublime Text, PhpStorm	Написание и редактирование SQL-скриптов, хранимых процедур, триггеров, функций, скриптов миграции и резервного копирования

4	Система управления базами данных: — PostgreSQL, MySQL, MariaDB, Microsoft SQL Server, SQLite	Хранение и обработка данных веб-приложений; проектирование, разработка, администрирование и защита баз данных; выполнение лабораторных работ по темам ПМ.01
5	Система виртуализации: — VirtualBox, VMware Workstation Player, Hyper-V, Docker	Развертывание и тестирование серверных конфигураций СУБД, изолированных сред для отработки резервного копирования, репликации, восстановления и защиты БД
6	Веб-сервер с функцией интерпретации и/или исполнения программного кода: — Apache, Nginx, XAMPP, OpenServer	Локальная разработка и запуск серверных скриптов, тестирование взаимодействия веб-приложения с базой данных, отработка защиты от SQL-инъекций
7	Инструменты загрузки файлов на сервер: — FileZilla, WinSCP, PuTTY (SCP/SFTP), rsync	Деплой проектов на удалённый хостинг, передача дампов БД и резервных копий, настройка удалённого администрирования СУБД
8	Инструменты коллективной работы с исходным кодом: — Git, GitHub, GitLab, Bitbucket	Организация командной разработки, контроль версий SQL-скриптов, миграций, хранимых процедур, совместная работа над схемой БД
9	Инструменты тестирования кода и сервисов: — pgTAP, MySQL Test Framework, DbUnit, JMeter, Postman, pytest	Проверка работоспособности и качества SQL-кода, модульное и интеграционное тестирование БД, нагрузочное тестирование запросов, тестирование API взаимодействия с БД
10	Графический растровый и векторный редакторы: — Draw.io , Figma, GIMP, Inkscape, Adobe Photoshop	Создание и обработка графических элементов интерфейсов, построение ER-диаграмм, схем баз данных, визуализация архитектуры БД и схем связей
11	Специализированные CASE-средства и клиенты для работы с БД: — DBeaver, pgAdmin, MySQL Workbench, dbdiagram.io , ERwin	Проектирование концептуальной, логической и физической моделей БД, визуальное построение ER-диаграмм, администрирование СУБД, выполнение запросов, мониторинг
12	Инструменты резервного копирования и восстановления БД: — pg_dump / pg_restore, mysqldump / mysqlbackup, pgBackRest, Barman	Выполнение полного, инкрементального и дифференциального резервного копирования, восстановление данных, настройка автоматического бэкапа в штатном режиме

13	Инструменты мониторинга и журналирования СУБД: — pgBadger, Prometheus + Grafana, Zabbix, ELK Stack (Elasticsearch, Logstash, Kibana)	Мониторинг производительности БД, анализ медленных запросов, сбор и агрегация журналов событий, настройка оповещений о критических событиях сервера БД
14	Инструменты обеспечения безопасности БД: — OpenSSL, Wireshark, OWASP ZAP, Burp Suite, HashiCorp Vault	Настройка защищённых соединений (SSL/TLS, VPN), аудит безопасности БД, тестирование на устойчивость к SQL-инъекциям, управление секретами и ключами шифрования

3.2 Информационное обеспечение обучения

Для реализации программы библиотечный фонд структурного подразделения должен иметь печатные и /или электронные образовательные и информационные ресурсы для использования в образовательном процессе.

Основные печатные и электронные издания:

1. Гордеев А.Н., Широков В.А. Основные этапы жизненного цикла программного продукта // Учебник. Москва : ЮРАЙТ, 2021. Электронный ресурс. Доступ: Университетская библиотека онлайн (UBO).
2. Лаптев А.В., Малыгин А.В. Объектно-ориентированное проектирование и паттерны проектирования // Учебное пособие. Екатеринбург : Уральский федеральный университет, 2022. Электронный ресурс. Доступ: Научная электронная библиотека (eLibrary.ru).
3. Гусев В.А., Андрианов А.В. Основы математического моделирования сложных систем // Учебник. Москва : ЮРАЙТ, 2021. Электронный ресурс. Доступ: Университетская библиотека онлайн (UBO).
4. Барбашин В.А., Смирнова А.С. Применение численных методов в математическом моделировании // Учебное пособие. Новосибирск : Новосибирский государственный университет, 2022. Электронный ресурс. Доступ: Электронно-библиотечная система (ЭБС) Института математики СО РАН.
5. Бабешко В.А., Малютин И.А. Анализ и интерпретация результатов математического моделирования // Практическое руководство. Владивосток : Дальневосточный федеральный университет, 2023. Электронный ресурс. Доступ: Российская электронная библиотека (elibrary.ru).

Дополнительные источники:

1. Кириллов А.В., Зарипов В.Н. Примеры и упражнения по курсу математического моделирования // Учебно-методическое пособие. Санкт-Петербург : Санкт-Петербургский политехнический университет Петра Великого, 2021. Электронный ресурс. Доступ: Центр коллективного пользования научным оборудованием (СПбПУ).
2. Егоров С.А., Мишанин А.В. Контроль качества результатов математического моделирования // Практическое руководство. Казань :

Казанский федеральный университет, 2022. Электронный ресурс. Доступ: Виртуальная лаборатория KFU.

3. Филиппов В.А., Сергеев А.В. Исследование устойчивости математических моделей технических объектов // Методическое пособие. Ульяновск : Ульяновский государственный университет, 2023. Электронный ресурс. Доступ: Учебно-методический портал регионального центра повышения квалификации.

4. Контроль и оценка результатов освоения профессионального модуля

Код ПК, ОК	Критерии оценки результата (показатели освоённости 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 выполнения практических и лабораторных заданий, оценка решения ситуационных задач, оценка тестового контроля, результатов наблюдений за деятельностью обучающегося в процессе учебной и 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.06	описывает значимость своей специальности	
ОК.07	соблюдает нормы экологической безопасности определять направления ресурсосбережения в рамках профессиональной деятельности по специальности	
ОК.08	чередует смену деятельности; выполняет комплекс лечебной гимнастики с учетом профессиональной деятельности	
ОК.09	понимает общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимает тексты на базовые профессиональные темы	

	участвовать в диалогах на знакомые общие и профессиональные темы; пишет простые связные сообщения на знакомые или интересующие профессиональные темы	
ПК 2.1	проектирует модули программного обеспечения с учетом технического задания; визуализирует и описывает архитектурные решения; определяет интерфейсы и взаимодействие модулей в системе	
ПК 2.2	создает модули программного обеспечения; оптимизирует код и алгоритмы программных модулей для увеличения производительности; мониторит и анализирует производительность приложений	
ПК 2.3	проводит интеграцию программных модулей и компонентов в единое программное решение; работает с API и веб-сервисами для взаимодействия между модулями; работает с интеграционными платформами и инструментами; обеспечивает совместимость и стабильность системы	
ПК 2.4	проводит отладку программного обеспечения на уровне программных модулей; тестирует программное обеспечение; формирует тестовые сценарии; готовит тестовые платформы (устанавливает операционную систему, дополнительное программное обеспечение и другое по необходимости); проводит оценку объема тестирования программного обеспечения с целью определения необходимых ресурсов для его выполнения; настраивает тестовые среды и аппаратные средства для выполнения тестирования программного обеспечения в соответствии с заданием на тестирование в пределах своей компетенции; формирует и предоставляет отчетность о подготовке к выполнению задания на тестирование программного обеспечения в соответствии с установленными регламентами; выполняет тестовые процедуры на тестовых данных	
ПК 2.5	создает техническую документацию для модулей; документирует код, API и интерфейсов; работает со специализированным программным обеспечением по документированию программного кода	