
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«25» мая 2026 г.
Протокол № 2

**Рабочая программа дисциплины (модуля)
«Разработка на Python»**

Направление подготовки: 02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Прикладной искусственный интеллект

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2026

**Москва
2026**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения	5
3. Тематический план	11
4. Содержание дисциплины (модуля)	11
5. Учебно-методическое обеспечение	12
6. Материально-техническое обеспечение	12
7. Методические и оценочные материалы	14

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Разработка на Python» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению 02.03.01 «Математика и компьютерные науки», профиль «Прикладной искусственный интеллект», утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Разработка на Python» формирует у обучающегося практические навыки разработки масштабируемых и производительных веб-сервисов, востребованных в современной IT-индустрии. Освоение инструментов FastAPI, GitLab, Redis и loguru обеспечивает готовность к участию в реальных проектах в составе команды под руководством опытного специалиста.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 «Математика и компьютерные науки», профиль «Прикладной искусственный интеллект» и входит в вариативную часть Блока 1 формируемую участниками образовательных отношений.

Дисциплина (модуль) изучается на 3 курсе в 6 семестре. Доступна к изучению после успешного освоения дисциплины (модуля): «Язык программирования Python. Основной».

Цель изучения дисциплины (модуля): подготовить обучающегося к разработке и сопровождению современных веб-приложений на Python с использованием асинхронных технологий, инструментов версионирования и интеграции с внешними сервисами.

Задачи изучения дисциплины (модуля):

- освоить принципы клиент-серверной архитектуры и организацию взаимодействия между фронтендом и бэкендом;
- научиться использовать Git и GitLab для командной разработки, включая ветвление, коммиты, merge request и разрешение конфликтов;
- освоить основы HTTP и разработку API с помощью FastAPI, включая маршрутизацию, обработку запросов и возврат ответов;
- научиться применять асинхронное программирование, кеширование с Redis и структурированное логирование с loguru для повышения производительности и надежности приложений;
- развить навыки автоматизации процессов через реализацию периодических задач с использованием apscheduler и BackgroundTasks FastAPI.

В результате освоения дисциплины (модуля), обучающийся должен:

знать:

- принципы работы клиент-серверной архитектуры: роль фронтенда и бэкенда;
- основы организации проекта: структура каталогов, работа с;
- основные команды Git: создание репозитория, ветвление, коммиты, слияние и разрешение конфликтов;
- основы работы с GitLab: создание проектов, управление ветками, MR;
- принципы работы HTTP: методы запросов (GET, POST, PUT, DELETE), структура запроса и ответа;
- основные возможности FastAPI: асинхронные маршруты, работа с запросами, возвращение ответов;
- основы асинхронного программирования: что такое async/await, задачи асинхронности;
- что такое middleware, их назначение и основные примеры использования в FastAPI;
- основы кеширования: зачем оно нужно, типы кеширования (в памяти, распределенное), как кеширование влияет на производительность;

- основные принципы логирования: уровни логов, структурированное логирование, назначение логов в разработке;

- что такое периодические задачи и как их использовать для автоматизации процессов;

- основы работы с планировщиком и FastAPI BackgroundTasks для реализации периодических задач.

уметь:

- работать с Git для управления версионностью проекта: выполнять коммиты, переключаться между ветками, работать с удаленными репозиториями;

- настраивать проект на FastAPI: создавать основные маршруты, разделять файлы по назначению;

- взаимодействовать с внешними API через HTTP-запросы с использованием библиотеки httpx;

- обрабатывать данные из сторонних API, передавать их фронтенду через маршруты FastAPI;

- использовать асинхронные функции для работы с длительными операциями;

- реализовывать middleware в FastAPI;

- настраивать кеширование с использованием Redis для повышения производительности приложений;

- использовать loguru для добавления логирования в приложение;

- настраивать формат вывода логов;

- управлять уровнями логирования;

- логировать исключения и критические события;

- создавать периодические задачи с использованием apscheduler и BackgroundTasks FastAPI;

- автоматизировать выполнение фоновых операций.

владеть:

- навыками создания и ведения проекта в GitLab;

- навыками разработки проектов в команде с использованием Git: создавать MR, проводить код-ревью, разрешать конфликты при слиянии;

- навыками написания бэкенда для готового фронтенда, обеспечив корректное взаимодействие клиентской и серверной частей;

- навыками организации проектной структуры для комплексного FastAPI-приложения, разделяя логику API, бизнес-логику и обработку данных;

- навыками реализации взаимодействия с несколькими сторонними API, комбинируя их данные в одном сервисе;

- навыками интеграции асинхронной обработки данных в проект, обеспечивая стабильность и производительность приложения;

- навыками разработки middleware;

- навыками настройки кеширования с продвинутыми функциями: инвалидация кеша, настройка TTL;

- навыками организации централизованного структурированного логирования с использованием loguru;

- навыками реализации комплексных периодических задач, используя комбинацию планировщика и FastAPI BackgroundTasks, обеспечивая стабильную и предсказуемую работу сервиса;

- навыками связи результатов логирования, кеширования и периодических задач для мониторинга и улучшения производительности приложения.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Код и содержание компетенции	Индикатор компетенции и перечень планируемых результатов обучения по дисциплине (модулю)
УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1. Знает методы поиска и анализа информации в области разработки, основные принципы критической оценки источников информации и их релевантности.
	УК-1.2. Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем.
	УК-1.3. Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации.
УК-3. Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде (соответствует УНК-02)	УК-3.1. (УНК-02.01.) Принимает участие в групповом взаимодействии в ходе профессиональной деятельности (базовый) УНК-02.01.01. У-1. Проявляет понимание своей роли в команде УНК-02.01.01. У-2. Умеет ясно выражать свои мысли в группе УНК-02.01.01. У-3. Способен слушать и учитывать мнения других участников команды. УНК-02.01.01. У-4. Умеет давать конструктивную обратную связь УНК-02.01.01. У-5. Умеет работать в команде, проявлять инициативу и поддерживать коллег УНК-02.01.01. У-6. Обладает навыками разрешения конфликтных ситуаций и поиска компромиссов УНК-02.01.01. З-1. Знает основы командной работы, роли и ответственности каждого участника.
	УК-3.2. (УНК-02.02.) Проявляет лидерство и осуществляет наставничество (базовый) УНК-02.02.01. У-1. Умеет мотивировать и вдохновлять команду для достижения общих целей УНК-02.02.01. У-2. Способен организовывать работу группы, распределять обязанности и контролировать выполнение задач УНК-02.02.01. У-3. Умеет делиться знаниями и опытом, оказывать поддержку и осуществлять наставничество коллегам УНК-02.02.01. У-4. Способен принимать ответственные решения и демонстрировать пример профессионализма УНК-02.02.01. З-1. Знает основы лидерства, мотивации и командообразования УНК-02.02.01. З-2. Знает принципы эффективного наставничества и развития персонала УНК-02.02.01. З-3. Знает методы оценки эффективности работы команды и индивидуальных участников
ОПК-6. Способен разрабатывать алгоритмы и компьютерные программы,	ОПК-6.1. Знает принципы алгоритмизации, основные структуры данных, парадигмы программирования, методы разработки, тестирования и отладки компьютерных программ.

пригодные для практического применения	ОПК-6.2. Умеет разрабатывать, документировать, тестировать и отлаживать алгоритмы и компьютерные программы, пригодные для практического применения, с использованием современных языков и сред разработки.
	ОПК-6.3. Имеет практический опыт разработки и сопровождения компьютерных программ и программных комплексов для решения прикладных задач.
ППК-ДА3. Способен проводить аналитические исследования с применением технологий больших данных	ППК-ДА3.2. – Обеспечивает соответствие результатов анализа бизнес-задачам заказчика (средний) ППК-ДА3.2. 3-1. Знает методы перевода бизнес-требований в аналитические задачи ППК-ДА3.2. 3-2. Знает ключевые бизнес-метрики в предметной области
ППК-Р5. Способен разрабатывать требования и проектировать программное обеспечение	ППК-Р5.1. Анализирует возможности реализации требований к компьютерному программному обеспечению (средний) ППК-Р5.1. 3-1. Знает возможности существующей программно-технической архитектуры. ППК-Р5.1. 3-2. Знает возможности современных и перспективных средств разработки программных продуктов, технических средств. ППК-Р5.1. 3-3. Знает методологии разработки компьютерного программного обеспечения и технологии программирования. ППК-Р5.1. У-1. Умеет проводить сбор и систематизацию требований к компьютерному программному обеспечению. ППК-Р5.1. У-2. Умеет выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению ППК-Р5.1. У-3. Умеет вырабатывать варианты реализации требований к компьютерному программному обеспечению.
	ППК-Р5.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие (средний) ППК-Р5.2. 3-1. Знает методы и средства проектирования компьютерного программного обеспечения. ППК-Р5.2. 3-2. Знает методы и средства проектирования программных интерфейсов. ППК-Р5.2. У-1. Умеет выбирать средства реализации требований к компьютерному программному обеспечению. ППК-Р5.2. У-2. Умеет вырабатывать варианты реализации компьютерного программного обеспечения. ППК-Р5.2. У-3. Умеет проводить оценку и обоснование рекомендуемых решений.
	ППК-Р5.3. Проектирует компьютерное программное обеспечение (продвинутый) ППК-Р5.3. 3-1. Знает принципы построения и виды архитектуры компьютерного программного обеспечения. ППК-Р5.3. 3-2. Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке компьютерного программного обеспечения. ППК-Р5.3. 3-3. Знает нормативно-технические документы (стандарты), определяющие требования к технической документации на компьютерное программное обеспечение. ППК-Р5.3. 3-4. Знает методы и средства проектирования компьютерного программного обеспечения.

	ППК-Р5.3. У-1. Умеет разрабатывать и изменять архитектуру компьютерного программного обеспечения и согласовывать ее с системным аналитиком и архитектором программного обеспечения. ППК-Р5.3. У-2. Умеет проектировать структуры данных. ППК-Р5.3. У-3. Умеет проектировать программные интерфейсы.
	ППК-Р5.4. Выполняет логическое проектирование системы (продвинутый) ППК-Р5.4. З-1. Знает устройство и функционирование ИТ-систем/продуктов ППК-Р5.4. З-2. Знает методы моделирования и описания устройства и функционирования ИТ-систем/продуктов, их частей, обеспечения и окружения ППК-Р5.4. З-3. Знает базовые технологии взаимодействия и интеграции систем и компонентов ППК-Р5.4. З-4. Знает методы функциональной декомпозиции ИТ-систем ППК-Р5.4. У-1. Умеет декомпозировать ИТ-системы и ИТ-продукты на подсистемы и элементы поставки ППК-Р5.4. У-2. Умеет описывать интерфейсы пользователя на логическом уровне ППК-Р5.4. У-3. Умеет описывать интеграции со смежными системами на логическом уровне
ППК-Р6. Способен участвовать в промышленной разработке программного обеспечения	ППК-Р6.1. Работает в соответствии с промышленными методологиями разработки (средний) ППК-Р6.1. З-1. Знает принципы Agile и их применение в промышленных проектах. ППК-Р6.1. З-2. Знает процессы code review, принципы коллективного владения кодом (collective code ownership). ППК-Р6.1. У-1. Умеет оценивать объем задачи и срок ее выполнения, участвовать в планировании спринтов. ППК-Р6.1. У-2. Умеет работать в команде с использованием инструментов управления проектами.
	ППК-Р6.2. Использует инструменты промышленной разработки (средний) ППК-Р6.2. З-1. Знает принципы Continuous Integration and Continuous Delivery (CI/CD). ППК-Р6.2. З-2. Знает системы мониторинга и логирования в продуктивной среде. ППК-Р6.2. У-1. Умеет настраивать потоки работ CI/CD. ППК-Р6.2. У-2. Умеет работать с контейнеризацией и оркестрацией. ППК-Р6.2. У-3. Умеет настраивать мониторинг в продуктивной среде.
	ППК-Р6.3. Разрабатывает масштабируемый и поддерживаемый код (продвинутый) ППК-Р6.3. З-1. Знает принципы чистого кода, SOLID, DRY, KISS и др. ППК-Р6.3. З-2. Знает принципы предметно-ориентированного проектирования (ПОП) программного обеспечения. ППК-Р6.3. З-3. Знает паттерны проектирования и антипаттерны. ППК-Р6.3. У-1. Умеет разрабатывать модульный и тестируемый программный код. ППК-Р6.3. У-2. Умеет выполнять модульное, интеграционное и нагрузочное тестирование.

	ППК-Р6.3. У-3. Умеет проводить рефакторинг для повышения качества кода. ППК-Р6.3. У-4. Умеет применять принципы ПОП при разработке программного обеспечения на языках программирования высокого уровня абстракций и в LowCode и NoCode системах.
	ППК-Р6.4. Участвует в развертывании и поддержке программного обеспечения (продвинутой) ППК-Р6.4. З-1. Знает стратегии развертывания промышленного программного обеспечения. ППК-Р6.4. З-2. Знает основы работы с облачными платформами. ППК-Р6.4. У-1. Умеет развертывать приложения в облаке. ППК-Р6.4. У-2. Умеет обнаруживать и устранять инциденты с работой продуктивного программного обеспечения.
ППК-Р7. Способен применять искусственный интеллект (ИИ) для генерации и отладки программного кода	ППК-Р7.1. Применяет ИИ-инструменты для генерации программного кода (базовый) ППК-Р7.1. З-1. Знает принципы работы современных генеративных ИИ-моделей для генерации кода. ППК-Р7.1. З-2. Знает ограничения и риски использования ИИ-генерации (безопасность, качество кода, лицензирование). ППК-Р7.1. У-1. Умеет формулировать корректные текстовые запросы (промты) для генерации кода. ППК-Р7.1. У-2. Умеет интегрировать ИИ-инструменты в среду разработки.
	ППК-Р7.2. Использует ИИ для анализа и отладки кода (средний) ППК-Р7.2. З-1. Знает методы ИИ-анализа кода. ППК-Р7.2. З-2. Знает форматы и инструменты для автоматизированного тестирования с ИИ. ППК-Р7.2. У-1. Умеет настраивать ИИ-инструменты для поиска уязвимостей. ППК-Р7.2. У-2. Умеет интерпретировать рекомендации ИИ по исправлению кода.
	ППК-Р7.3. Оптимизирует код с помощью ИИ (продвинутой) ППК-Р7.3. З-1. Знает методы ИИ-оптимизации. ППК-Р7.3. З-2. Знает критерии качества кода, применяемые ИИ-системами. ППК-Р7.3. У-1. Умеет использовать ИИ для рефакторинга. ППК-Р7.3. У-2. Умеет проверять корректность оптимизаций, предложенных ИИ.
	ППК-Р7.4. Оценивает этические и профессиональные аспекты применения ИИ в разработке (продвинутой) ППК-Р7.4. З-1. Знает этические нормы использования ИИ (конфиденциальность, плагиат кода и т.п.). ППК-Р7.4. З-2. Знает лицензионные ограничения сгенерированного кода. ППК-Р7.4. У-1. Умеет проверять код на соответствие стандартам после ИИ-генерации. ППК-Р7.4. У-2. Умеет документировать использование ИИ в разработке.

ППК-ДА4. Способен выявлять бизнес-проблемы или бизнес-возможности	ППК-ДА4.2. Выявление истинных бизнес-проблем или бизнес-возможностей (продвинутый) ППК-ДА4.2. 3-1. Знает предметную область и специфику деятельности организации в объеме, достаточном для решения задач бизнес-анализа ППК-ДА4.2. 3-2. Знает инструменты, техники анализа бизнес-ситуации и предметной области, включая анализ данных. ППК-ДА4.2. 3-3. Знает перспективные и существующие цифровые технологии и цифровые возможности для бизнеса в контексте предметной области и специфики деятельности организации.
РГ-1. Способен руководить процессами разработки компьютерного программного обеспечения	РГ-1.1. Руководит разработкой программного кода (средний) РГ-1.1. 3-1. Знает методы принятия управленческих решений. РГ-1.1. 3-2. Знает основные принципы и методы управления персоналом РГ-1.1. 3-1 Знает технологии межличностной и групповой коммуникации в деловом взаимодействии, основы конфликтологии РГ-1.1. У-1. Способен распределять задачи на разработку программного кода между исполнителями. РГ-1.1. У-2. Применяет лучшие мировые практики оформления программного кода. РГ-1.1. У-3. Оценивает качество и эффективность программного кода.
	РГ-1.2. Руководит проверкой работоспособности компьютерного программного обеспечения (средний) РГ-1.2. 3-1. Знает основные методы измерения и оценки характеристик компьютерного программного обеспечения РГ-1.2. 3-2. Знает методы и средства проверки работоспособности компьютерного программного обеспечения РГ-1.2. 3-3. Знает методы и средства рефакторинга и оптимизации программного кода РГ-1.2. 3-4. Знает о возможности отказа частей системы в результате воздействий внутреннего и внешнего нарушителя (хакер, неосторожный пользователь, программист, поставщик компонентов) РГ-1.2. У-1. Умеет распределять задачи на проверку работоспособности компьютерного программного обеспечения между исполнителями. РГ-1.2. У-2. Умеет оценивать результаты проверки работоспособности компьютерного программного обеспечения РГ-1.2. У-3. Умеет принимать управленческие решения по результатам проверки работоспособности компьютерного программного обеспечения об исправлении ошибок, рефакторинге, оптимизации и инспекции кода РГ-1.2. У-4. Умеет включать поиск возможных отказов из воздействий внутреннего или внешнего нарушителя (хакер, неосторожный пользователь, программист, поставщик компонентов) в процедуры инспекции программного кода.
	РГ-1.3. Руководит интеграцией программных модулей и компонентов компьютерного программного обеспечения (продвинутый) РГ-1.3. 3-1. Знает методы и средства сборки модулей и компонентов компьютерного программного обеспечения

	РГ-1.3. 3-2. Знает методы и программные интерфейсы взаимодействия компьютерного программного обеспечения с внешними программными компонентами РГ-1.3. 3-3. Знает методы проектирования и разработки программных интерфейсов взаимодействия внутренних модулей компьютерного программного обеспечения РГ-1.3. 3-4. Знает методы и средства разработки процедур для развертывания компьютерного программного обеспечения РГ-1.3. У-1. Умеет назначать задания на разработку процедур интеграции, сборку, подключение к внешней среде, проверку работоспособности выпусков программного продукта РГ-1.3. У-2. Умеет оценивать результаты выполнения назначенных заданий на разработку процедур интеграции, сборку, подключение к внешней среде, проверку работоспособности выпусков программного продукта РГ-1.3. У-3. Умеет принимать управленческие решения по результатам проверки работоспособности выпусков программного продукта (решение о выпуске/невыпуске версии, отправка задач на доработку, добавление новых задач, передача на тестирование)
РГ-4. Способен организовать промышленную разработку программного обеспечения	РГ-4.1. Внедряет промышленные методологии разработки (средний) РГ-4.1. 3-1. Знает принципы Agile и их применение в промышленных проектах. РГ-4.1. 3-2. Знает процессы code review, принципы коллективного владения кодом (collective code ownership). РГ-4.1. У-1. Умеет оценивать объем задач и срок их выполнения. РГ-4.1. У-2. Умеет распределять задачи между участниками команды. РГ-4.1. У-3. Умеет организовать работу в команде с использованием инструментов управления проектами.
	РГ-4.3. Организует разработку масштабируемого и поддерживаемого кода (продвинутый) РГ-4.3. 3-1. Знает принципы чистого кода, SOLID, DRY, KISS и др. РГ-4.3. 3-2. Знает принципы предметно-ориентированного проектирования (ПОП) программного обеспечения. РГ-4.3. 3-3. Знает паттерны проектирования и антипаттерны. РГ-4.3. У-1. Умеет управлять разработкой модульного и тестируемого программного кода. РГ-4.3. У-2. Умеет организовать выполнение модульного, интеграционного и нагрузочного тестирования. РГ-4.3. У-3. Умеет управлять техническим долгом, в том числе организовать рефакторинг для повышения качества кода. РГ-4.3. У-4. Умеет применять принципы ПОП при разработке программного обеспечения.
	РГ-4.4. Организует развертывание и поддержку программного обеспечения (продвинутый) РГ-4.4. 3-1. Знает стратегии развертывания промышленного программного обеспечения. РГ-4.4. У-2. Умеет управлять обнаружением и устранением инцидентов с работой продуктивного программного обеспечения.

3. Тематический план

№ п/ п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятель ная работа	
		Лекции	Практические занятия			
1	Базовые инструменты и взаимодействие с сервисами	4	6		20	Домашнее задание
2	Проектирование веб-приложений на FastAPI	10	16	2	50	Домашнее задание Контрольная работа
3	Инженерные практики и промышленная разработка	10	16	4	50	Домашнее задание Контрольная работа Проект
	Зачет с оценкой			2		
Итого:		24	38	8	120	
Объем дисциплины (модуля) (в ак. ч.)		190				
Объем дисциплины (модуля) (в зач. ед.)		5				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Базовые инструменты и взаимодействие с сервисами	Git и GitLab Основы HTTP и работа с API
2	Проектирование веб-приложений на FastAPI	FastAPI Асинхронные маршруты и обработка запросов Dependency Injection Взаимодействие с внешними API через асинхронные библиотеки MongoDB
3	Инженерные практики и промышленная разработка	Основы логирования и использование logging Основы кеширования и использование Redis Продвинутое кеширование и инвалидация кеша Периодические задачи в FastAPI

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Чернышев, С. А. Основы программирования на Python : учебник для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588667>.
2. Виафоре, П. Надежный Python : практическое руководство / П. Виафоре, М. Райтман. - Санкт-Петербург : БХВ-Петербург, 2023. - 352 с. - ISBN 978-5-9775-1174-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2123392>.

Дополнительная литература:

1. Бейдер, Д. Чистый Python. Тонкости программирования для профи : практическое руководство / Д. Бейдер. - Санкт-Петербург : Питер, 2021. - 288 с. - (Серия «Библиотека программиста»). - ISBN 978-5-4461-0803-9. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1766370>.
2. Хиллард, Д. Секреты Python Pro / Д. Хиллард. - Санкт-Петербург : Питер, 2021. - 320 с. - ISBN 978-5-4461-1684-3. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2142837>.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и

обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное

Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robot Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Разработка на Python» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, консультации, домашние задания, контрольные работы, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар – это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Консультации – структурированные встречи, на которых преподаватели предоставляют индивидуальную или групповую помощь в освоении учебного материала, обсуждении вопросов и решении проблем, возникающих в процессе обучения.

Консультации могут включать разъяснение сложных тем, подготовку к экзаменам и помощь в выполнении проектных работ, что способствует более глубокому пониманию предмета и улучшению академической успеваемости.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал, использовать различные источники

информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Контрольная работа – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы – получить специальные знания по одной или нескольким темам дисциплины и продемонстрировать навыки их практического применения.

Проект – исследовательская работа по дисциплине (модулю) и презентация результатов.

Для успешной подготовки к проекту рекомендуется: четко определить цели и задачи проекта; составить план работы, разбив проект на этапы с указанием сроков выполнения каждого из них; использовать разнообразные источники информации и инструменты для исследования темы; регулярно проверять прогресс и вносить коррективы в план, если это необходимо.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине «Разработка на Python»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *зачета с оценкой*, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом, если иное не указано в силлабусе дисциплины:

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и
9	Отлично	
8	Отлично	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине, но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	
1	Не сдан	

Дисциплина (модуль) «Разработка на Python» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашнее задание	15%	11	Набор задач по темам недели
Проект	25%	1	Исследовательская работа по дисциплине (модулю) и презентация результатов
Контрольная работа	30%	3	Письменная работа с набором задач, которые нужно решить за ограниченное время
Зачет с оценкой	30%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Разработка на Python»: $\langle 0,25 \times \text{проект} + 0,15 \times \text{среднее за домашние задания} + 0,3 \times \text{среднее за контрольные работы} + 0,3 \times \text{зачет с оценкой} \rangle$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1: Работа с Git и основы взаимодействия с API

1. Установите Git и настройте учётную запись на GitLab;
2. Создайте новый проект в GitLab, инициализируйте локальный репозиторий и свяжите его с удалённым;
3. Создайте ветку feature/api-request, перейдите в неё;
4. Напишите Python-скрипт, который отправляет GET-запрос к публичному API с использованием библиотеки `httpx`;
5. Выведите статус ответа и тело запроса в консоль;
6. Добавьте комментарий в код: `# Запрос к внешнему API`;
7. Закоммитьте изменения, отправьте ветку в GitLab и создайте Merge Request в main с описанием проделанной работы.

Домашнее задание 2: Разработка асинхронного API на FastAPI

1. Создайте FastAPI-приложение с асинхронным маршрутом `/ping`, возвращающим `{"status": "ok"}`;
2. Добавьте маршрут `/posts`, который асинхронно получает данные с `https://jsonplaceholder.typicode.com/posts` и возвращает первые 10 постов;
3. Реализуйте параметризованный маршрут `/posts/{post_id}`, который возвращает данные конкретного поста по ID;
4. Используйте Dependency Injection для выделения логики запроса к внешнему API в отдельную функцию;
5. Добавьте обработку ошибок: если пост не найден, возвращайте HTTP 404;
6. Сохраните полученные данные в MongoDB (или в JSON-файл, если MongoDB недоступна);
7. Оформите код с комментариями и выложите в GitLab в новой ветке feature/fastapi-app.

Домашнее задание 3: Логирование, кеширование и фоновые задачи

1. Настройте структурированное логирование в приложении с помощью `loguru`: логируйте входящие запросы, ошибки и ключевые события;
2. Подключите Redis и реализуйте кеширование для маршрута `/posts/{post_id}`: при первом запросе данные берутся из внешнего API и сохраняются в Redis с TTL 60 секунд, при повторном — из кеша;
3. Добавьте механизм инвалидации кеша: при ручном запросе на `/cache/clear` кеш очищается;
4. Реализуйте периодическую задачу с помощью `apscheduler`, которая каждые 2 минуты асинхронно получает один случайный пост и сохраняет его в MongoDB;
5. Добавьте логирование запуска и результата фоновой задачи;
6. Выложите реализацию в ветку feature/caching-logging-scheduler и создайте Merge Request.

Примерное описание задания и критерии оценивания к проекту

Цель проекта: разработать полнофункциональное FastAPI-приложение, демонстрирующее навыки работы с асинхронностью, внешними API, системами хранения данных, кешированием, логированием и периодическими задачами, а также применение современных практик командной разработки.

Задачи, которые необходимо выполнить студенту:

- создать репозиторий в GitLab, настроить структуру проекта и организовать работу с ветками;
- реализовать асинхронные маршруты FastAPI для получения и обработки данных из публичного REST API (например, <https://jsonplaceholder.typicode.com>);
- использовать Dependency Injection для разделения логики маршрутов и бизнес-логики;
- интегрировать MongoDB (или другой источник данных) для хранения результатов;
- подключить Redis и реализовать кеширование ответов с возможностью инвалидации по запросу;
- настроить структурированное логирование с помощью loguru: логировать запросы, ошибки и фоновые операции;
- реализовать периодическую задачу с использованием apscheduler, которая автоматически получает данные с внешнего API и сохраняет их в базу каждые 2–5 минут;
- обеспечить обработку ошибок и возврат корректных HTTP-статусов;
- оформить код в соответствии с PEP 8, добавить комментарии и документацию (описание endpoints в Swagger);
- выложить проект в GitLab, создать Merge Request и представить результаты.

Критерии оценивания проекта:

- корректная реализация асинхронных маршрутов и взаимодействие с внешним API с использованием асинхронных библиотек;
- применение принципов инжиниринга: логирование, кеширование с инвалидацией, фоновые задачи и работа с Redis;
- использование современных практик разработки: структура проекта, Dependency Injection, обработка ошибок, работа с MongoDB;
- соблюдение стандартов командной разработки: работа с Git и GitLab, ветвление, коммиты, MR, читаемость кода;
- функциональность и стабильность приложения: корректная работа всех компонентов, отсутствие критических ошибок при запуске и выполнении задач.

Примерные задания для контрольной работы

1. Напишите асинхронный маршрут FastAPI `@app.get("/health")`, который возвращает JSON-ответ `{"status": "ok"}` при успешном запуске приложения.
2. Объясните, зачем используется `async` и `await` в FastAPI. Приведите пример асинхронной функции, выполняющей HTTP-запрос к внешнему API с помощью `httpx`.
3. Реализуйте зависимость (dependency), которая проверяет наличие заголовка `X-API-Key` в запросе. Если ключ отсутствует или не равен `secret123`, возвращается ошибка 401.
4. Напишите код, который с помощью `httpx.AsyncClient` получает данные с <https://jsonplaceholder.typicode.com/posts/1> и возвращает только поле `title`.
5. Опишите, как подключить MongoDB в FastAPI-приложении. Приведите пример асинхронного сохранения документа (например, `{"post_id": 1, "title": "Test"}`) в коллекцию `posts`.
6. Как настроить логирование в FastAPI с помощью `loguru`? Напишите код, который логирует каждый входящий запрос (метод и путь) с уровнем `INFO`.
7. Объясните, зачем используется Redis в веб-приложениях. Напишите функцию, которая сохраняет данные в Redis с TTL 30 секунд и проверяет их наличие.

8. Реализуйте кеширование для маршрута `/posts/{post_id}`: при первом запросе данные берутся из внешнего API и кешируются, при повторном — возвращаются из Redis. Укажите, как можно инвалидировать кеш по запросу `/cache/clear`.

9. Напишите пример периодической задачи с использованием `BackgroundTasks` и `apscheduler`, которая каждые 5 минут логирует сообщение "Выполнение фоновой задачи...".

10. Опишите, как реализовать graceful shutdown FastAPI-приложения, чтобы завершить все фоновые задачи и корректно отключиться от Redis и MongoDB.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция	Индикатор
1	Опишите роли участников команды при разработке Python-приложения на FastAPI и распределении задач по реализации функционала.	Разработчик пишет код на Python, архитектор проектирует API, тестировщик проверяет эндпоинты, все несут ответственность за этап	УК-3, УК-1	УК-3.1, УК-1.2
2	Как мотивировать команду и распределить ресурсы при релизе Python-сервиса с ограниченным временем?	Четкое распределение задач, контроль сроков, пример профессионализма, оценка трудоемкости этапов проекта	УК-3, РГ-1	УК-3.2, РГ-1.1
3	Подготовьте презентацию результатов тестирования FastAPI-приложения для стейкхолдеров и руководства компании.	Структура: цель, методы, результаты, выводы, рекомендации с учетом целевых установок заинтересованных сторон	УК-1, ППК-ДА3	УК-1.3, ППК-ДА3.2
4	Оформите документацию по проекту Python-приложения согласно профессиональным стандартам и требованиям к техническим спецификациям.	Титульный лист, содержание, введение, основная часть, выводы, список источников, приложения по ГОСТ	УК-1, ППК-Р5	УК-1.3, ППК-Р5.3
5	Дайте определение архитектуре FastAPI и запишите её смысл в контексте Python-разработки.	FastAPI это асинхронный веб-фреймворк для построения API с автоматической документацией и валидацией данных	УК-1, ОПК-6	УК-1.2, ОПК-6.1

6	Проанализируйте экономическую информацию для выбора между FastAPI и Django для проекта веб-сервиса.	Сравнение затрат, производительности, рисков, учет специфики задач, влияние на эффективность бизнес-решений	УК-1, ОПК-6	УК-1.2, ОПК-6.2
7	Обоснуйте экономическое решение о выборе фреймворка для Python-разработки на основе анализа затрат и выгод.	Решение выбирается если учитывает интересы всех сторон, повышает защиту инвестора от неблагоприятных изменений	УК-1, ОПК-6	УК-1.3, ОПК-6.3
8	Запишите формулу для оценки времени обработки запроса и выберите метод анализа данных для оптимизации Python-приложения.	Формула зависит от операций ввода-вывода и асинхронных вызовов, метод статистического анализа для оценки значимости оптимизаций	ОПК-6, ППК-ДА3	ОПК-6.2, ППК-ДА3.2
9	Оптимизируйте выбор архитектуры Python-приложения и обоснуйте влияние на бизнес показатели компании.	Выбор архитектуры с максимальной эффективностью, оценка воздействия на ключевые показатели компании, скорость разработки	ОПК-6, ППК-Р5	ОПК-6.3, ППК-Р5.3
10	Обоснуйте экономическое решение в учебном проекте с учетом системного мышления при проектировании архитектуры Python-сервиса.	Использование модели с учетом взаимосвязей между модулями, сервисами и бизнес-целями компании	ОПК-6, РГ-4	ОПК-6.3, РГ-4.1
11	Назовите правовые нормы регулирующие обработку персональных данных в Python-приложениях и их смысл для проекта.	ФЗ-152 О персональных данных, требования к безопасности API, ответственность за нарушение требований при сборе данных	ППК-Р5, УК-1	ППК-Р5.3, УК-1.1
12	Примените правовые нормы при оценке рисков нарушения требований при работе с данными в Python-приложении.	Анализ рисков несоблюдения законодательных требований, оценка вероятности и пороговых значений, меры минимизации штрафов	ППК-Р5, ППК-Р6	ППК-Р5.3, ППК-Р6.1

13	Дайте правовую оценку ситуации с нарушением требований по защите данных в проекте разработки Python-приложения.	Определение нарушений законодательных актов, применение санкций, влияние на репутацию организации	ППК-Р5, УК-1	ППК-Р5.3, УК-1.3
14	Выберите метод статистического анализа для проверки гипотез о влиянии изменений на производительность Python-приложения.	А/В тест, проверка гипотез о значимости различий, расчет доверительных интервалов для оценок времени обработки	ППК-ДА3, ППК-ДА4	ППК-ДА3.2, ППК-ДА4.2
15	Адаптируйте аналитическую модель оценки метрик приложения под бизнес потребности и оцените влияние решений.	Перевод технических метрик в бизнес показатели, оценка воздействия на ключевые показатели компании, оптимизация процессов	ППК-ДА3, ППК-Р6	ППК-ДА3.2, ППК-Р6.3
16	Выберите тип визуализации для представления данных о результатах тестирования Python-приложения и коммуникации с руководством.	График времени отклика и доверительных интервалов, интерактивный дашборд для публичной дискуссии о результатах анализа	ППК-ДА3, УК-1	ППК-ДА3.2, УК-1.3
17	Выявите бизнес проблему на основе анализа отклонений в метриках Python-приложения и примите решение.	Проблема низкой производительности и/или высоких ошибок, выбор стратегии реагирования, решение о корректировке функционала или архитектуры	ППК-ДА4, ППК-Р7	ППК-ДА4.2, ППК-Р7.2
18	Классифицируйте бизнес возможности при масштабировании Python-сервиса с учетом стратегического планирования.	Оптимизация вычислительных ресурсов, разработка плана действий с учетом целей компании, выявление резервов роста	ППК-ДА4, РГ-4	ППК-ДА4.2, РГ-4.3
19	Сформулируйте рекомендации на основе статистических выводов о результатах тестирования кода для управления продуктом.	При значимом улучшении внедрение изменения, при отсутствии эффекта	ППК-Р6, ППК-Р7	ППК-Р6.3, ППК-Р7.3

		отказ от изменения для эффективности системы		
20	Оцените риски реализации проекта разработки Python-приложения и разработайте меры управления.	Анализ вероятности срывов сроков, мониторинг индикаторов успешности, пересмотр плана при выявлении проблем	РГ-1, УК-1	РГ-1.3, УК- 1.3