МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ)

Рабочая программа дисциплины (модуля),
как компонент образовательной программы
базового высшего образования
по направлению подготовки
09.03.01 Информатика и вычислительная техника,
утвержденной первым проректором ФГАОУ ВО
РУТ (МИИТ) Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Разработка и интеграция программных интерфейсов (API)

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Цифровая инженерия транспортных процессов

Форма обучения: Очная

Рабочая программа дисциплины (модуля) в виде электронного документа выгружена из единой корпоративной информационной системы управления университетом и соответствует оригиналу

Простая электронная подпись, выданная ФГАОУ ВО РУТ (МИИТ)
ID подписи: 937226
Подписал: руководитель образовательной программы Проневич Ольга Борисовна
Дата: 08.09.2026

1. Общие сведения о дисциплине (модуле).

Целью освоения учебной дисциплины является формирование у обучающихся системных знаний и практических навыков разработки, документирования, тестирования, обеспечения безопасности и интеграции программных интерфейсов приложений (API) для эффективного обмена данными между компонентами информационных систем транспортного предприятия.

Дисциплина предназначена для формирования и закрепления у обучающихся компетенций, необходимых для решения следующих профессиональных задач:

Проектирование и разработка RESTful API с использованием современных веб-фреймворков (FastAPI).

Создание интерактивной документации (Swagger/OpenAPI) и спецификаций для программных интерфейсов.

Интеграция программных модулей и внешних систем через API.

Обеспечение безопасности API (аутентификация, авторизация, валидация данных).

Тестирование API (модульное, интеграционное, функциональное).

Управление версиями API и их мониторинг.

2. Планируемые результаты обучения по дисциплине (модулю).

Перечень формируемых результатов освоения образовательной программы (компетенций) в результате обучения по дисциплине (модулю):

ППК-Р1 - Способен разрабатывать и отлаживать программный код;

ППК-Р3 - Способен выполнять интеграцию программных модулей и компонентов и проверять работоспособность выпусков программного продукта.

Обучение по дисциплине (модулю) предполагает, что по его результатам обучающийся будет:

Знать:

Интерфейсы взаимодействия с внешней средой: протоколы HTTP/HTTPS, форматы данных JSON/XML/Protobuf, архитектурные стили REST и GraphQL, принципы работы OpenAPI/Swagger, основы веб-безопасности (CORS, OAuth2, JWT).

Интерфейсы взаимодействия внутренних модулей системы: паттерны проектирования API (Router-Controller-Service-Repository), Dependency

Injection, схемы данных и валидацию на основе Pydantic, концепции Middleware, обработчики исключений и слои приложения.

Уметь:

Писать программный код процедур интеграции программных модулей: создавать эндпоинты для обмена данными, обрабатывать HTTP-запросы и формировать структурированные ответы, реализовывать клиентские библиотеки для взаимодействия с внешними API, обрабатывать ошибки и исключения с возвратом понятных клиенту статус-кодов и сообщений.

Использовать выбранную среду программирования для разработки процедур интеграции: применять веб-фреймворк FastAPI для построения серверной части, использовать инструменты для тестирования (pytest, httpx), отладки и профилирования, а также средства автоматической генерации документации (Swagger UI, ReDoc).

Владеть:

Навыками разработки, документирования и версионирования программных интерфейсов, включая описание контракта API в формате OpenAPI и ведение интерактивной документации.

Навыками интеграции внешних и внутренних сервисов с использованием API, включая реализацию аутентификации и авторизации, обработку ошибок, организацию асинхронного взаимодействия и настройку мониторинга работоспособности.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 3 з.е. (108 академических часа(ов).

3.2. Объем дисциплины (модуля) в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Тип учебных занятий	Количество часов	
	Всего	Семестр №3
Контактная работа при проведении учебных занятий (всего):	48	48
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	16	16

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 60 академических часа (ов).

3.4. При обучении по индивидуальному учебному плану, в том числе при ускоренном обучении, объем дисциплины (модуля) может быть реализован полностью в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации.

4. Содержание дисциплины (модуля).

4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	Тема 1. Введение в программные интерфейсы (API). Рассматриваемые вопросы: - Роль API в современной разработке и интеграции систем на транспорте. - Эволюция подходов к интеграции: от файлового обмена к сервис-ориентированной архитектуре (SOA) и микросервисам. - Виды API: открытые, партнерские, внутренние. - Сетевые протоколы: обзор и сравнение (HTTP/HTTPS, WebSocket, gRPC). - Принципы клиент-серверного взаимодействия.
2	Тема 2. Архитектурные стили API: REST, GraphQL, gRPC. Рассматриваемые вопросы: - Сравнительный анализ архитектурных подходов. - Принципы RESTful архитектуры: клиент-сервер, отсутствие состояния, кэшируемость, единообразие интерфейса. - Ресурсы и их идентификация через URI. Использование HTTP-методов (GET, POST, PUT, PATCH, DELETE). - Коды состояния HTTP и их семантика. Идемпотентность запросов. - Введение в GraphQL и gRPC: преимущества и области применения.
3	Тема 3. Форматы обмена данными: JSON, XML, Protobuf. Рассматриваемые вопросы: - Сравнение форматов: структура, синтаксис, производительность. - Сериализация и десериализация данных в Python. - Работа с вложенными структурами и сложными объектами. - Обработка дат, времени и специальных типов. - Схемы данных и валидация (JSON Schema, Pydantic).
4	Тема 4. Стандарты описания API: OpenAPI и Swagger. Рассматриваемые вопросы: - Введение в OpenAPI Specification (OAS). Структура спецификации.

№ п/п	Тематика лекционных занятий / краткое содержание
	- Описание эндпоинтов, параметров, тел запросов и ответов. - Компоненты и схемы данных. - Генерация интерактивной документации (Swagger UI, ReDoc). - Использование Swagger Editor и Swagger Codegen.
5	Тема 5. Введение в FastAPI и асинхронность. Рассматриваемые вопросы: - Почему FastAPI? Ключевые преимущества. - Первое приложение: маршруты и обработчики. - Запуск сервера (Uvicorn). Жизненный цикл запроса. - Концепция асинхронного программирования: Event Loop, async/await. - Сравнение синхронного и асинхронного подходов.
6	Тема 6. Схемы данных и валидация с Pydantic. Рассматриваемые вопросы: - Основы Pydantic: BaseModel, поля, типы данных. - Обязательные и необязательные поля. Значения по умолчанию. - Валидация данных (валидаторы, field_validator, model_validator). - Вложенные модели и списки. - Использование Pydantic для парсинга и валидации запросов в FastAPI.
7	Тема 7. Обработка запросов и формирование ответов в FastAPI. Рассматриваемые вопросы: - Path Parameters, Query Parameters, Request Body. - Обработка Form Data и File Upload. - Модели ответов (response_model). Фильтрация данных. - Управление статус-кодами ответов (status_code). - Заголовки и Cookies.
8	Тема 8. Обработка ошибок и исключений в API. Рассматриваемые вопросы: - Исключения в FastAPI. Встроенные HTTPException. - Создание пользовательских обработчиков исключений. - Валидация ошибок и возврат понятных сообщений клиенту. - Логирование ошибок. Инструменты мониторинга (Sentry).
9	Тема 9. Dependency Injection (DI) в FastAPI. Рассматриваемые вопросы: - Концепция внедрения зависимостей для разделения логики. - Создание зависимостей (глобальные, для маршрута, для эндпоинта). - Использование зависимостей для аутентификации, доступа к БД, валидации. - Внедрение сервисов через DI.
10	Тема 10. Безопасность и аутентификация в API. Рассматриваемые вопросы: - Основы безопасности API: CORS, OAuth2, JWT (JSON Web Tokens). - Реализация аутентификации в FastAPI. - Защита эндпоинтов. Роли и права доступа (авторизация). - Работа с паролями (хэширование). - Использование OAuth2-провайдеров (Google, Yandex).
11	Тема 11. Взаимодействие с базой данных через API. Рассматриваемые вопросы: - Интеграция с СУБД (PostgreSQL). - Работа с ORM (SQLAlchemy). Паттерн "Репозиторий". - Асинхронная работа с БД. Транзакции.

№ п/п	Тематика лекционных занятий / краткое содержание
	- Связь между моделями данных и схемами Pydantic. - Миграции базы данных (Alembic).
12	Тема 12. Тестирование API: модульное и интеграционное. Рассматриваемые вопросы: - Виды тестирования API: модульное, интеграционное, функциональное, E2E. - Инструменты: pytest, httpx. - Создание клиента для тестов. Fixtures для подготовки данных. - Изоляция тестов. Mocking внешних зависимостей. - Проверка статус-кодов, тел ответов и схем данных.
13	Тема 13. Управление версиями API. Рассматриваемые вопросы: - Стратегии версионирования: через URL, заголовки, параметры запроса. - Совместимость версий: мажорные, минорные, патч-версии. - Поддержка устаревших версий (deprecation policy). - Плавный переход пользователей на новые версии. - Документирование изменений (Changelog).
14	Тема 14. Асинхронное программирование и работа с внешними сервисами. Рассматриваемые вопросы: - Глубокое погружение в асинхронность: Event Loop, Coroutines, Tasks. - Работа с внешними API через aiohttp. - Таймауты и повторные попытки (retries). - Параллельные и конкурентные запросы (asyncio.gather). - Асинхронные очереди и фоновые задачи (Celery, RQ).

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Тема 1. Настройка инструментов и основы HTTP. Рассматриваемые вопросы: - Установка Python, VS Code/PyCharm, Git. - Работа с HTTP-клиентами: cURL, Postman, HTTPie. - Отправка GET и POST запросов. - Анализ HTTP-заголовков и кодов ответа.
2	Тема 2. Первое приложение на FastAPI. Рассматриваемые вопросы: - Создание проекта. Установка зависимостей (fastapi, uvicorn). - Создание простых эндпоинтов (GET, POST). - Запуск сервера и тестирование через Swagger UI. - Обработка Path и Query параметров.
3	Тема 3. Валидация данных с Pydantic. Рассматриваемые вопросы: - Создание Pydantic-моделей для запросов и ответов. - Валидация входных данных (обязательные поля, типы, ограничения). - Обработка ошибок валидации. - Вложенные модели.
4	Тема 4. CRUD-операции: работа с данными в памяти. Рассматриваемые вопросы:

№ п/п	Тематика практических занятий/краткое содержание
	- Реализация API для управления сущностью (например, "Транспортное средство"). - Эндпоинты: GET /items, GET /items/{id}, POST /items, PUT /items/{id}, DELETE /items/{id}. - Хранение данных в списке (в памяти). - Обработка ошибок (404 Not Found).
5	Тема 5. Интеграция с PostgreSQL через SQLAlchemy. Рассматриваемые вопросы: - Настройка подключения к PostgreSQL. - Создание моделей SQLAlchemy. - Реализация CRUD-операций с БД. - Использование Dependency Injection для сессий БД.
6	Тема 6. Реализация аутентификации и авторизации (JWT). Рассматриваемые вопросы: - Регистрация пользователей (хэширование паролей). - Вход в систему: выдача JWT-токена. - Защита эндпоинтов через зависимость. - Проверка прав доступа (роли).
7	Тема 7. Тестирование API с pytest. Рассматриваемые вопросы: - Написание модульных и интеграционных тестов для API. - Использование TestClient. - Создание fixtures для тестовых данных. - Mocking внешних сервисов.
8	Тема 8. Документирование и версионирование API. Итоговый проект. Рассматриваемые вопросы: - Создание и настройка OpenAPI-спецификации. - Добавление примеров и описаний в Swagger UI. - Реализация версионирования (v1/v2). - Защита итогового проекта.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение рекомендуемой литературы и официальной документации FastAPI, Pydantic, SQLAlchemy, pytest.
2	Работа с официальной документацией OpenAPI/Swagger, Docker, Git.
3	Написание тестов для разработанного API.
4	Подготовка к защите итогового практического проекта.
5	Подготовка к промежуточной аттестации.
6	Подготовка к текущему контролю.

5. Перечень изданий, которые рекомендуется использовать при освоении дисциплины (модуля).

№ п/п	Библиографическое описание	Место доступа

1	Рудалев, В. Г. Интерфейс api win32 на примерах : учебно-методическое пособие / В. Г. Рудалев, Д. А. Щеглаков. — Воронеж : ВГУ, 2010. — 32 с.	https://e.lanbook.com/book/358451
2	Нурматова, Е. В. Организация доступа к базам данных на основе DB API-, ORM- технологий : учебное пособие / Е. В. Нурматова. — Москва : РТУ МИРЭА, 2024. — 37 с. — ISBN 978-5-7339-2325-3.	https://e.lanbook.com/book/448958

6. Перечень современных профессиональных баз данных и информационных справочных систем, которые могут использоваться при освоении дисциплины (модуля).

https://habr.com/ru - база знаний в виде статей, обзоров

http://library.miit.ru/ - электронно-библиотечная система научно-технической библиотеки МИИТ

https://superset.apache.org/user-docs/intro/#documentation – документация Apache Superset

https://www.postgresql.org/ - официальная страница PostgreSQL

https://yandex.cloud/ru/docs/datalens/ - официальная документация Yandex Datalens

https://docs.docker.com/ - официальная документация Docker

https://www.metabase.com/docs/latest/ - официальная документация Metabase

https://tableau.pro/ - официальный сайт Tableau с лучшими практиками проектирования дашбордов

https://revealthedata.com/blog/ - Блог о визуализации данных и BI-системах

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

MS Office (Excel, Word, PowerPoint)
Apache Superset
Yandex Datalens
Metabase
PostgreSQL
Docker
Apache Airflow
Браузер Chrome / Яндекс браузер

Draw.io

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).

Компьютер преподавателя с доступом к сети Интернет

Компьютеры студентов с установленным Python-окружением и табличным процессором

Доступ к учебной базе событийных данных и экспериментальным выборкам

Доступ к PostgreSQL и BI-системе для визуализации результатов

Доступ к облачной виртуальной машине или Docker-контейнерам для воспроизводимого анализа

Проектор и экран для проектора

Маркерная доска

9. Форма промежуточной аттестации:

Зачет в 3 семестре.

10. Оценочные материалы.

Оценочные материалы, применяемые при проведении промежуточной аттестации, разрабатываются в соответствии с локальным нормативным актом РУТ (МИИТ).

Авторы:

руководитель образовательной программы О.Б. Проневич

Согласовано:

Директор Д.В. Паринов

Руководитель образовательной программы О.Б. Проневич

Председатель учебно-методической комиссии Д.В. Паринов