

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)

Тестирование и рефакторинг кода на Python

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): IT-сервисы и технологии обработки данных на транспорте

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: директор центра Проневич Ольга Борисовна

Дата: 21.10.2026

1. Общие сведения о дисциплине (модуле).

Целями освоения учебной дисциплины являются формирование у обучающихся системных знаний и практических навыков проверки работоспособности, тестирования, отладки, инспекции и рефакторинга программного кода на Python, а также автоматизации контроля качества программных и ML-компонентов в современных процессах разработки и MLOps.

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

- выявление, воспроизведение, диагностика и устранение дефектов программного кода;
- разработка автоматических тестов и применение средств статического и динамического контроля качества;
- рефакторинг и оптимизация Python-кода с сохранением его внешнего поведения и документированием изменений;
- интеграция тестирования, мониторинга моделей, переобучения и A/B-тестирования развертываний в MLOps-процессы.

2. Планируемые результаты обучения по дисциплине (модулю).

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

МО-2 - Способен проектировать и внедрять системы автоматизации жизненного цикла машинного обучения (MLOps);

ППК-Р1 - Способен разрабатывать и отлаживать программный код;

ППК-Р2 - Способен проверять работоспособность и проводить рефакторинг кода программного обеспечения.

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

Знать:

методы и приемы отладки программного кода на уровне отдельных функций, модулей и программных компонентов;

методы и средства проверки работоспособности компьютерного программного обеспечения;

типичные ошибки, возникающие при разработке компьютерного программного обеспечения, методы их диагностики и исправления;

методы и средства рефакторинга и инспекции программного кода;
руководящие и нормативно-технические документы по стандартизации требований к программному коду и документам автоматизированных систем;
порядок отражения изменений в системе управления версиями, а также результатов рефакторинга, оптимизации и инспекции в коллективной базе знаний;

назначение мониторинга моделей машинного обучения, автоматизированного переобучения и А/В-тестирования развертываний.

Уметь:

выявлять ошибки и дефекты в программном коде Python;
отлаживать программный код на уровне программных модулей;
применять методы и средства проверки работоспособности компьютерного программного обеспечения;

интерпретировать диагностические данные проверки работоспособности программного обеспечения и анализировать значения полученных характеристик;

воспроизводить дефекты программного кода, зафиксированные в базе данных дефектов, и выяснять причины их возникновения;

вносить изменения в программный код для устранения выявленных дефектов;

анализировать программный код на соответствие требованиям по читаемости, сопровождаемости и производительности;

применять методы и средства рефакторинга и инспекции программного кода;

использовать инструменты MLOps при тестировании и эксплуатации компонентов машинного обучения;

развертывать системы мониторинга моделей машинного обучения;

автоматизировать переобучение моделей и А/В-тестирование развертываний;

публиковать результаты рефакторинга и инспекции в коллективной базе знаний.

Владеть:

навыками разработки и запуска модульных, интеграционных и параметризованных тестов для программного кода Python;

навыками диагностики, воспроизведения и документирования дефектов программного обеспечения;

навыками применения статического анализа, проверки типов, профилирования и оценки покрытия кода тестами;

навыками безопасного рефакторинга программных модулей с сохранением их внешнего поведения;

навыками организации контроля качества кода в системе управления версиями и CI/CD-пайплайне;

навыками мониторинга ML-компонентов, автоматизации переобучения и проверки развертываний.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 2 з.е. (72 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 №2
Контактная работа при проведении учебных занятий (всего):	16	16
В том числе:		
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 56 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.

Не предусмотрено учебным планом

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Тема 1. Организация контроля качества Python-кода и работа с дефектами Рассматриваемые вопросы: <ul style="list-style-type: none">- виды ошибок и дефектов программного обеспечения;- жизненный цикл дефекта: обнаружение, воспроизведение, исправление и повторная проверка;- оформление диагностической информации и карточки дефекта;- структура учебного репозитория, ветвление и фиксация изменений в Git.
2	Тема 2. Модульное тестирование Python-кода с использованием pytest Рассматриваемые вопросы: <ul style="list-style-type: none">- структура теста и соглашения именования;- проверки assert и организация набора тестов;- тестирование функций, классов и исключительных ситуаций;- запуск тестов и интерпретация отчета pytest.
3	Тема 3. Параметризация, fixtures, mocks и интеграционное тестирование Рассматриваемые вопросы: <ul style="list-style-type: none">- параметризованные тесты и граничные случаи;- fixtures и управление тестовыми данными;- подмена внешних зависимостей с помощью mock;- интеграционные тесты модулей, API и операций с данными.
4	Тема 4. Отладка и интерпретация диагностических данных Рассматриваемые вопросы: <ul style="list-style-type: none">- воспроизведение дефектов и минимизация проблемного сценария;- отладка в IDE, breakpoint, пошаговое выполнение и просмотр состояния;- анализ traceback, исключений, stderr/stdout и журналов;- логирование, уровни сообщений и диагностический контекст.
5	Тема 5. Инспекция, статический анализ и оценка качества кода Рассматриваемые вопросы: <ul style="list-style-type: none">- проверка соответствия PEP 8 и настройка Ruff/Flake8;- форматирование кода и единые правила проекта;- статическая проверка типов с mypy;- покрытие тестами, профилирование времени и памяти, анализ производительности.
6	Тема 6. Рефакторинг Python-кода и устранение технического долга Рассматриваемые вопросы: <ul style="list-style-type: none">- признаки некачественного кода и типовые code smells;- декомпозиция функций и классов, устранение дублирования;- упрощение условной логики и улучшение именования;- безопасный рефакторинг под защитой автоматических тестов.
7	Тема 7. Тестирование и мониторинг компонентов машинного обучения Рассматриваемые вопросы: <ul style="list-style-type: none">- проверки входных данных, схемы, качества и воспроизводимости;- тестирование feature engineering и предсказаний модели;- мониторинг качества модели, data drift и concept drift;- формирование метрик, логов и сигналов для переобучения.

№ п/п	Тематика практических занятий/краткое содержание
8	Тема 8. Автоматизация контроля качества в CI/CD и итоговый проект Рассматриваемые вопросы: - автоматический запуск тестов, линтеров и проверки покрытия; - версионирование кода и публикация результатов инспекции в базе знаний; - автоматизация переобучения и А/В-тестирование развертываний; - итоговый pipeline: тестирование, рефакторинг, мониторинг и документирование.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение рекомендуемой литературы и нормативно-технических материалов по тестированию, отладке и рефакторингу программного кода.
2	Работа с официальной документацией Python, pytest, coverage.py, Ruff, mypy и Git.
3	Самостоятельная разработка тестов для функций, модулей и операций обработки данных.
4	Воспроизведение, диагностика и документирование программных дефектов.
5	Подготовка CI/CD-сценария автоматической проверки качества проекта.
6	Выполнение итогового практического проекта и подготовка к промежуточной аттестации.
7	Подготовка к промежуточной аттестации.

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

№ п/п	Библиографическое описание	Место доступа
1	Моргунов, Е. П. PostgreSQL. Профессиональный SQL : учебное пособие / Е. П. Моргунов ; под редакцией Е. В. Рогова. — Москва : ДМК Пресс, 2025. — 444 с. — ISBN 978-5-93700-373-7.	https://e.lanbook.com/book/514901

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 во 2 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

Директор ЦС IT - программ ВИШ

О.Б. Проневич

Председатель учебно-методической
комиссии

Д.В. Паринов