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)

Управление сборкой и конфигурацией Python-приложений в Linux

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: 09.09.2026

1. Общие сведения о дисциплине (модуле).

Целью освоения учебной дисциплины является формирование у обучающихся системных знаний и практических навыков сборки, конфигурирования, упаковки и развертывания Python-приложений в среде Linux, обеспечивающих воспроизводимость, управляемость и масштабируемость компонентов data-платформ транспортного предприятия.

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

Настройка окружения для разработки и сборки Python-приложений в Linux.

Управление зависимостями и версионирование компонентов.

Сборка, упаковка и публикация Python-пакетов.

Конфигурирование приложений через переменные окружения и файлы конфигурации.

Проверка работоспособности собранного программного продукта.

Контейнеризация Python-приложений с использованием Docker.

Интеграция процессов сборки и конфигурации в CI/CD-пайплайны.

2. Планируемые результаты обучения по дисциплине (модулю).

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

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:

Методы и средства сборки и интеграции программных модулей и компонентов: системы управления зависимостями (pip, poetry, uv), инструменты сборки (setuptools, wheel, hatch), форматы пакетов (sdist, wheel), версионирование и семантическое управление версиями, артефакты сборки и их хранение, механизмы интеграции модулей в единый продукт.

Методы и средства проверки работоспособности выпусков программных продуктов: виды тестирования (модульное, интеграционное, системное), инструменты автоматизации тестирования (pytest, unittest),

использование тестовых покрытий и статического анализа, валидация конфигураций, smoke-тестирование и приемочные испытания.

Уметь:

Выполнять процедуры сборки программных модулей и компонентов в программный продукт: настраивать pyproject.toml и setup.py, управлять зависимостями, выполнять сборку пакетов, публиковать артефакты в репозиториях, использовать системы сборки для создания дистрибутивов.

Производить настройки параметров программного продукта и осуществлять запуск процедур сборки: работать с конфигурационными файлами, управлять переменными окружения, использовать профили и конфигурации для разных сред (dev, staging, production), проводить проверку работоспособности собранного продукта через тестирование и валидацию.

Владеть:

Навыками организации процесса сборки и конфигурации Python-приложений в Linux-среде, включая настройку виртуальных окружений, управление зависимостями, создание и публикацию пакетов, а также контейнеризацию приложений.

Навыками проверки работоспособности выпусков программного продукта, включая запуск тестовых наборов, анализ результатов тестирования, проверку соответствия требованиям и подготовку релизных артефактов.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 4 з.е. (144 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 №5
Контактная работа при проведении учебных занятий (всего):	64	64
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	32	32

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 80 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 управление сборкой Python-приложений. Рассматриваемые вопросы: - Жизненный цикл разработки Python-приложения. - Роль сборки и конфигурации в жизненном цикле. - Экосистема инструментов для сборки Python-проектов. - Принципы воспроизводимости сборок. - Введение в Linux-среду для разработки.
2	Тема 2. Виртуальные окружения и управление версиями Python. Рассматриваемые вопросы: - Необходимость изоляции окружений. - Инструменты: venv, virtualenv, pyenv, conda. - Управление версиями Python через pyenv. - Сравнение подходов к управлению окружениями. - Практика создания и активации окружений.
3	Тема 3. Управление зависимостями: pip, requirements.txt. Рассматриваемые вопросы: - Формат requirements.txt. - Фиксация зависимостей (pip freeze). - Разделение зависимостей: prod, dev, test. - Управление версиями пакетов. - Разрешение конфликтов зависимостей.
4	Тема 4. Современные инструменты управления зависимостями: poetry и uv. Рассматриваемые вопросы: - Введение в Poetry: pyproject.toml, poetry.lock. - Команды poetry: add, install, update, build, publish. - Введение в uv: быстрый менеджер зависимостей.

№ п/п	Тематика лекционных занятий / краткое содержание
	- Сравнение poetry, uv и pip/requirements. - Выбор инструмента для проекта.
5	Тема 5. Системы сборки Python: setuptools и wheel. Рассматриваемые вопросы: - Структура setup.py и setup.cfg. - Создание sdist и wheel дистрибутивов. - Настройка метаданных пакета. - Управление entry_points и консольными скриптами. - Публикация пакетов в PyPI и частных репозиториях.
6	Тема 6. Современные системы сборки: hatch и flit. Рассматриваемые вопросы: - Введение в Hatch: универсальный инструмент. - Введение в Flit: минималистичная сборка. - Сравнение hatch, flit и setuptools. - Переход с setuptools на современные системы. - Выбор системы сборки для проекта.
7	Тема 7. Конфигурирование Python-приложений. Рассматриваемые вопросы: - Источники конфигурации: переменные окружения, файлы, аргументы CLI. - Форматы конфигурации: .env, JSON, YAML, TOML. - Библиотеки для управления конфигурацией: python-dotenv, pydantic-settings. - Иерархия конфигураций и переопределение параметров. - Безопасное хранение секретов.
8	Тема 8. Управление окружениями для разных сред (dev/staging/prod). Рассматриваемые вопросы: - Подходы к разделению конфигураций по средам. - Использование профилей и переменных окружения. - Настройка приложения для работы в разных средах. - Примеры: настройка подключения к БД, внешним API. - Автоматизация переключения между средами.
9	Тема 9. Логирование и мониторинг в Python-приложениях. Рассматриваемые вопросы: - Библиотека logging: настройка и использование. - Форматы логов и обработчики (handlers). - Интеграция с системами логирования (ELK, Loki). - Настройка уровней логирования для разных сред. - Структурированное логирование с python-json-logger.
10	Тема 10. Структура проекта и best practices для Python. Рассматриваемые вопросы: - Рекомендуемая структура Python-проекта. - Организация исходного кода, тестов, документации. - Использование make или task-раннеров для автоматизации. - Поддержка нескольких версий Python. - Соглашения о кодировании и оформлении.
11	Тема 11. Контейнеризация Python-приложений с Docker. Рассматриваемые вопросы: - Dockerfile для Python-приложения. - Многоступенчатая сборка (multi-stage builds). - Оптимизация размера образа.

№ п/п	Тематика лекционных занятий / краткое содержание
	- Управление зависимостями в контейнере. - Настройка контейнера через переменные окружения.
12	Тема 12. Docker Compose для окружений разработки и тестирования. Рассматриваемые вопросы: - Использование Docker Compose для локальной разработки. - Настройка окружения с зависимостями (БД, кэш, очередь). - Переопределение конфигураций для разных сред. - Совместная работа с приложением и сервисами. - Интеграция с CI/CD.
13	Тема 13. Интеграция сборки и конфигурации с CI/CD. Рассматриваемые вопросы: - Автоматизация сборки в CI/CD-пайплайнах. - Управление версиями и артефактами. - Автоматическая публикация пакетов. - Проверка работоспособности в CI. - Безопасное управление секретами в CI/CD.
14	Тема 14. Тестирование собранного программного продукта. Рассматриваемые вопросы: - Виды тестирования после сборки. - Smoke-тестирование и sanity-проверки. - Интеграционные и системные тесты. - Проверка конфигураций и переменных окружения. - Автоматизация проверки работоспособности.
15	Тема 15. Управление версиями и релизами Python-проектов. Рассматриваемые вопросы: - Семантическое версионирование (SemVer). - Автоматизация версионирования в CI/CD. - Создание релизных артефактов (changelog, release notes). - Стратегии управления релизными ветками. - Интеграция с системой управления версиями.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Тема 1. Настройка Linux-окружения для разработки. Рассматриваемые вопросы: - Установка Python через pyenv. - Настройка редактора (VS Code/PyCharm). - Основные команды Linux для разработчика. - Настройка Git и SSH-ключей.
2	Тема 2. Создание виртуальных окружений. Рассматриваемые вопросы: - Использование venv и virtualenv. - Использование poetry для создания окружения. - Работа с pyenv для переключения версий Python. - Активация и деактивация окружений

№ п/п	Тематика практических занятий/краткое содержание
3	Тема 3. Управление зависимостями с pip и requirements.txt. Рассматриваемые вопросы: - Создание requirements.txt. - Установка зависимостей в окружение. - Разделение зависимостей: requirements-dev.txt. - Разрешение конфликтов зависимостей.
4	Тема 4. Управление зависимостями с poetry. Рассматриваемые вопросы: - Инициализация проекта с poetry. - Добавление и обновление зависимостей. - Работа с poetry.lock и воспроизводимость. - Экспорт зависимостей в requirements.txt.
5	Тема 5. Настройка проекта с setuptools. Рассматриваемые вопросы: - Создание setup.py и setup.cfg. - Настройка метаданных проекта. - Сборка sdist и wheel пакетов. - Установка пакета в режиме разработки (pip install -e).
6	Тема 6. Настройка проекта с poetry для сборки. Рассматриваемые вопросы: - Настройка pyproject.toml для сборки. - Сборка пакета с poetry build. - Публикация пакета в PyPI. - Использование частного репозитория.
7	Тема 7. Работа с конфигурациями: .env и переменные окружения. Рассматриваемые вопросы: - Создание .env файлов. - Использование python-dotenv. - Настройка приложения через переменные окружения. - Подходы к хранению секретов.
8	Тема 8. Конфигурирование через файлы (JSON/YAML/TOML). Рассматриваемые вопросы: - Использование JSON/YAML/TOML для конфигурации. - Библиотеки: pyyaml, tomli, orjson. - Создание иерархической конфигурации. - Валидация конфигураций с pydantic.
9	Тема 9. Настройка логирования в Python-приложении. Рассматриваемые вопросы: - Настройка logging в приложении. - Добавление обработчиков (console, file). - Настройка форматов логов. - Интеграция с внешними системами логирования.
10	Тема 10. Создание Dockerfile для Python-приложения. Рассматриваемые вопросы: - Написание Dockerfile. - Использование multi-stage сборки. - Оптимизация размера образа. - Запуск приложения в контейнере.
11	Тема 11. Настройка окружения разработки с Docker Compose. Рассматриваемые вопросы:

№ п/п	Тематика практических занятий/краткое содержание
	- Создание docker-compose.yml. - Настройка сервисов (app, db, redis). - Переопределение конфигураций через .env. - Запуск и управление окружением.
12	Тема 12. Автоматизация сборки в CI/CD (GitHub Actions). Рассматриваемые вопросы: - Настройка workflow для сборки. - Установка зависимостей и сборка пакета. - Запуск тестов в CI. - Сохранение артефактов сборки.
13	Тема 13. Публикация пакетов через CI/CD. Рассматриваемые вопросы: - Автоматическая публикация в PyPI. - Использование токенов и секретов. - Сборка и публикация Docker-образов в CI. - Создание релизов и тегов в Git.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение официальной документации Poetry, setuptools, Docker.
2	амостоятельная настройка окружения для Python-проекта.
3	Разработка и сборка собственного Python-пакета.
4	Интеграция сборки в CI/CD-пайплайн.
5	Подготовка к защите итогового проекта.
6	Подготовка к промежуточной аттестации.
7	Подготовка к текущему контролю.

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

№ п/п	Библиографическое описание	Место доступа
1	PYTHON-программирование : учебное пособие / И. И. Баглаев, Т. Ж. Базаржапова, Н. В. Очирова, Н. В. Юможапова. — Улан-Удэ : БГУ, 2026. — 118 с. — ISBN 978-5-9793-2087-8.	https://e.lanbook.com/book/519059
2	Сергеева, О. А. Программирование на Python : учебно-методическое пособие / О. А. Сергеева. — Кемерово : КемГУ, 2024. — 157 с. — ISBN 978-5-8353-3123-9.	https://e.lanbook.com/book/420758

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:

Экзамен в 5 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