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)

Объектно ориентированное программирование на Python

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: 07.09.2026

1. Общие сведения о дисциплине (модуле).

Целями освоения учебной дисциплины являются формирование у обучающихся системных знаний и практических навыков в области объектно-ориентированного программирования для разработки и отладки программного кода при решении прикладных задач в транспортной отрасли. В рамках курса рассматриваются принципы ООП (инкапсуляция, наследование, полиморфизм, абстракция), синтаксис и стандартные библиотеки языка программирования Python, паттерны проектирования, принципы чистого кода (SOLID, DRY, KISS), инструменты коллективной разработки (Git), методы отладки и тестирования программного кода. Особое внимание уделяется практическим аспектам разработки программных модулей, оформлению кода в соответствии со стандартами (PEP8), адаптации алгоритмов под специфику задач транспортной отрасли и применению паттернов проектирования для создания поддерживаемого и масштабируемого кода.

Задачами освоения дисциплины являются:

Формирование теоретических знаний о парадигме объектно-ориентированного программирования, её основных принципах и областях применения.

Освоение синтаксиса и стандартных библиотек языка программирования Python для разработки программных модулей и компонентов.

Формирование практических навыков разработки и отладки программного кода с использованием современных сред разработки (PyCharm, VS Code).

Освоение принципов чистого кода (SOLID, DRY, KISS) и паттернов проектирования для создания поддерживаемого и масштабируемого кода.

Приобретение навыков описания задач предметной области в виде формальных математических моделей, пригодных для последующей алгоритмизации и программной реализации.

Развитие компетенций по работе с системами управления версиями (Git) и оформлению программного кода в соответствии с нормативно-техническими документами (стандарты и регламенты).

Формирование навыков выбора и обоснования алгоритмов для программной реализации моделей, оценки их эффективности для решения задач транспортной отрасли.

Дисциплина предназначена для формирования компетенций, необходимых для разработки, отладки и сопровождения программного

обеспечения в транспортных организациях, и предназначена для студентов всех ролевых траекторий: «Инженер данных», «Аналитик данных», «MLOps / Data Scientist».

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.

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

Знать:

Синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования, методологии разработки компьютерного программного обеспечения.

Методы и приемы отладки программного кода, нормативно-технические документы, определяющие требования к оформлению программного кода.

Уметь:

Применять выбранный язык программирования для написания программного кода, использовать среду программирования и возможности имеющейся технической и программной архитектуры.

Выявлять ошибки в программном коде, отлаживать программный код на уровне программных модулей, применять нормативно-технические документы к оформлению программного кода, регистрировать изменения в системе управления версиями.

Владеть:

Навыками описания задач предметной области в виде формальных математических моделей, пригодных для последующей алгоритмизации и программной реализации.

Методами адаптации методов и алгоритмов под специфику задачи, навыками оценки эффективности выбранных алгоритмов.

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 №1
Контактная работа при проведении учебных занятий (всего):	32	32
В том числе:		
Занятия лекционного типа	16	16
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 40 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 объектно-ориентированное программирование. Рассматриваемые вопросы: - Парадигмы программирования: структурное, процедурное, объектно-ориентированное, функциональное. - Основные принципы ООП: инкапсуляция, наследование, полиморфизм, абстракция. - Классы и объекты: понятие, атрибуты, методы. - Отношения между классами: ассоциация, агрегация, композиция. - Нотации UML для графического отображения структуры классов и их взаимодействий. - Программное обеспечение для моделирования.

№ п/п	Тематика лекционных занятий / краткое содержание
2	Тема 2. Классы и объекты в Python. Рассматриваемые вопросы: - Синтаксис объявления классов в Python. - Атрибуты класса и экземпляра. - Методы: экземпляра, класса, статические. - Конструктор __init__ и деструктор __del__. - Магические методы (__str__, __repr__, __eq__ и др.). - Модификаторы доступа: публичные, защищённые (_), приватные (__). - Свойства (property) и инкапсуляция.
3	Тема 3. Наследование и полиморфизм. Рассматриваемые вопросы: - Базовые и производные классы. - Наследование: синтаксис и особенности в Python. - Переопределение методов. - Функция super() и вызов методов родительского класса. - Множественное наследование и порядок разрешения методов (MRO). - Абстрактные классы и абстрактные методы (модуль abc). - Полиморфизм: понятие и реализация в Python. - Принцип подстановки Лисков.
4	Тема 4. Инкапсуляция и абстракция. Рассматриваемые вопросы: - Сокрытие данных и реализация. - Интерфейсы и абстрактные классы. - Геттеры, сеттеры и свойства (@property). - Принципы проектирования классов. - Инкапсуляция на уровне модулей и пакетов. - Абстракция как способ управления сложностью. - Примеры проектирования классов для транспортных задач.
5	Тема 5. Паттерны проектирования. Часть 1. Рассматриваемые вопросы: - Понятие паттерна проектирования. - Классификация паттернов: порождающие, структурные, поведенческие. - Порождающие паттерны: Одиночка (Singleton), Фабричный метод (Factory Method), Строитель (Builder). - Реализация паттернов на Python. - Применение паттернов для решения типовых задач. - Выбор адекватного паттерна для конкретной бизнес-задачи.
6	Тема 6. Паттерны проектирования. Часть 2. Рассматриваемые вопросы: - Структурные паттерны: Декоратор (Decorator), Адаптер (Adapter), Прокси (Proxy). - Поведенческие паттерны: Наблюдатель (Observer), Стратегия (Strategy), Команда (Command). - Реализация паттернов на Python. - Сравнительный анализ паттернов. - Обоснование выбора паттерна для конкретной задачи. - Применение паттернов в транспортных проектах.
7	Тема 7. Принципы чистого кода и качество программного обеспечения. Рассматриваемые вопросы: - Принципы SOLID: Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion. - Принципы DRY, KISS, YAGNI. - Code Review и его роль в обеспечении качества кода.

№ п/п	Тематика лекционных занятий / краткое содержание
	- Стандарты оформления кода (PEP8 для Python). - Нормативно-технические документы по оформлению программного кода. - Типичные ошибки при разработке и методы их предотвращения.
8	Тема 8. Системы управления версиями и коллективная разработка. Рассматриваемые вопросы: - Понятие системы управления версиями. - Основы работы с Git: инициализация репозитория, коммиты, ветки, слияние. - Регламент использования системы управления версиями. - Стратегии ветвления: Git Flow, GitHub Flow. - Регистрация изменений и работа с удалёнными репозиториями. - Совместная разработка: Pull Request, Code Review, разрешение конфликтов.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Тема 1. Установка и настройка среды разработки. Рассматриваемые вопросы: - Установка Python и менеджера пакетов pip. - Установка и настройка IDE (PyCharm / VS Code). - Настройка виртуального окружения (venv). - Первый запуск и настройка интерпретатора. - Создание первого проекта и написание простейшей программы. - Основы работы с отладчиком (Debugger).
2	Тема 2. Разработка классов и объектов. Рассматриваемые вопросы: - Создание классов для решения прикладных задач. - Реализация конструкторов и методов. - Создание экземпляров классов и работа с ними. - Использование магических методов. - Реализация инкапсуляции с использованием свойств (property). - Практика на примере транспортной задачи (класс Vehicle, Transport, Cargo).
3	Тема 3. Наследование и полиморфизм на практике. Рассматриваемые вопросы: - Создание иерархии классов с использованием наследования. - Переопределение методов родительского класса. - Использование super() для вызова методов родителя. - Реализация абстрактных классов и методов. - Демонстрация полиморфизма на примере. - Разрешение конфликтов при множественном наследовании.
4	Тема 4. Паттерны проектирования. Часть 1. Рассматриваемые вопросы: - Реализация паттерна Одиночка (Singleton). - Реализация паттерна Фабричный метод (Factory Method). - Реализация паттерна Строитель (Builder). - Тестирование и отладка реализованных паттернов. - Обоснование выбора конкретного паттерна для задачи. - Практическое применение на транспортных задачах.

№ п/п	Тематика практических занятий/краткое содержание
5	Тема 5. Паттерны проектирования. Часть 2. Рассматриваемые вопросы: - Реализация паттерна Декоратор (Decorator). - Реализация паттерна Наблюдатель (Observer). - Реализация паттерна Стратегия (Strategy). - Реализация паттерна Адаптер (Adapter). - Сравнительный анализ реализованных паттернов. - Выбор оптимального паттерна для конкретной транспортной задачи.
6	Тема 6. Применение принципов чистого кода. Рассматриваемые вопросы: - Рефакторинг существующего кода с применением SOLID. - Применение принципов DRY, KISS при разработке. - Оценка качества и эффективности программного кода. - Выявление типичных ошибок и их исправление. - Code Review: анализ чужого кода и поиск проблем. - Применение стандартов оформления кода PEP8.
7	Тема 7. Работа с Git. Рассматриваемые вопросы: - Инициализация репозитория и настройка Git. - Создание коммитов и работа с историей. - Создание и переключение веток. - Слияние веток и разрешение конфликтов. - Работа с удалённым репозиторием (GitHub). - Регистрация изменений и оформление Pull Request. - Отработка навыков на учебном проекте.
8	Тема 8. Комплексный проект. Рассматриваемые вопросы: - Разработка программного модуля для транспортной задачи. - Применение всех изученных принципов ООП и паттернов. - Реализация кода с соблюдением стандартов оформления. - Отладка и тестирование разработанного модуля. - Оформление технической документации. - Защита проекта перед преподавателем.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение документации по выбранному языку программирования (Python).
2	Выполнение домашних заданий по разработке классов и объектов.
3	Выполнение комплексного проекта: разработка программного модуля с применением ООП.
4	Подготовка к промежуточной аттестации.
5	Подготовка к промежуточной аттестации.

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

№ п/п	Библиографическое описание	Место доступа
1	Объектно-ориентированное программирование : учебное пособие / В. В. Лозовский, Е. Н. Штрекер, Е. С. Данилович [и др.]. — Москва : РТУ МИРЭА, 2025. — 484 с. — ISBN 978-5-7339-2498-4.	https://e.lanbook.com/book/493547
2	Лозовский, В. В. Объектно-ориентированное программирование : учебно-методическое пособие / В. В. Лозовский, В. А. Морозов, А. А. Гололобов. — Москва : РТУ МИРЭА, 2024. — 107 с. — ISBN 978-5-7339-2120-4.	https://e.lanbook.com/book/405227

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 1 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