

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**  
**(РУТ (МИИТ))**



Рабочая программа дисциплины (модуля),  
как компонент образовательной программы  
высшего образования - программа бакалавриата  
по направлению подготовки  
09.03.01 Информатика и вычислительная техника,  
утвержденной первым проректором 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): 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 РУТ (МИИТ)  
ID подписи: 937226  
Подписал: руководитель образовательной программы  
Проневич Ольга Борисовна  
Дата: 19.05.2025

## 1. Общие сведения о дисциплине (модуле).

Целями освоения учебной дисциплины являются формирование у студентов базы знаний и навыков в области объектно-ориентированного программирования на Python, а также формирование и закрепление у студентов компетенций в области прикладной информатики.

Задачами освоения дисциплины (модуля):

- Обеспечение качества в проектах в области информационных технологий в соответствии с установленными регламентами;
- Распространение информации в проектах в области информационных технологий в соответствии с трудовым заданием.

## 2. Планируемые результаты обучения по дисциплине (модулю).

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

**ОПК-5** - Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем;

**ОПК-7** - Способен участвовать в настройке и наладке программно-аппаратных комплексов;

**ОПК-8** - Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

**ПК-2** - Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности .

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

### **Знать:**

- представление о функциональных возможностях языка;
- методы разработки алгоритмов в парадигме объектно-ориентированного программирования;
- принципы объектно-ориентированного анализа и проектирования;
- основы объектно-ориентированного подхода к программированию;
- форматы и интерфейсы обмена данными между информационными системами;
- основы конфигурационного управления;
- виды архитектур информационных систем.

### **Уметь:**

- эффективно использовать инструментарий высокоуровневых языков программирования для анализа больших данных;
- разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения;
- описывать задачу с точки зрения объектно-ориентированного подхода и подбирать соответствующие структуры для разработки алгоритма и программного кода;
- проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение;
- писать в современных средах разработки объектно-ориентированные программы.

**Владеть:**

- владеет навыками программирования, отладки и тестирования прототипов;
- основными приемами объектно-ориентированного программирования на языке Python;
- навыками использования библиотек Python, работой с классами и шаблонизаторами;
- системами контроля версий и поддержки конфигурационного управления, отслеживания ошибок;
- навыками работы в современных средах разработки и тестирования приложений, системах контроля версий и поддержки конфигурационного управления, отслеживания ошибок.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 6 з.е. (216 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	№4
Контактная работа при проведении учебных занятий (всего):	96	48	48
В том числе:			
Занятия лекционного типа	32	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 120 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. Парадигмы программирования, понятия классов и объектов. Рассматриваемые вопросы: - Элементы объектно-ориентированного программирования (далее ООП). - История, принципы и преимущества ООП. - Классы и экземпляры классов, объекты. - Атрибуты и методы классов. Поиск в иерархии классов. - Деревья классов. Создание классов в программе.
2	Тема 2. Основные операторы создания класса ООП. Рассматриваемые вопросы: - Оператор class, связи с деревьями классов. - Экземпляры, атрибуты, поля классов. - Параметр self в создании классов, принципы работы параметра self. Инициализатор <code>__init__</code> в описании классов. Конструктор объекта <code>__new__</code> . Деструктор.
3	Тема 3. Методы классов в ООП. Рассматриваемые вопросы: - Оператор функции def. Аргументы функции. - Определение методов класса Правила создания методов класса. - Вызовы методов класса в объектах. - Уровни методов класса. Параметр cls. Понятие «магического метода».
4	Тема 4. Уровни доступа к атрибутам класса. Рассматриваемые вопросы: - Реализация режимов доступа к классам Python. - Публичный доступ к классу. Наследованный доступ к классу из дочерних классов. - Закрытие класса от доступа извне. - Расширенные функции доступа к классу. Метод <code>__slots__</code> .

№ п/п	Тематика лекционных занятий / краткое содержание
5	Тема 5. Наследование в ООП Python. Основные правила наследования. Композиция классов. Рассматриваемые вопросы: - Роль наследования в ООП. - Родительские и дочерние классы. Суперклассы. - Пространство имен классов. Виды наследования. - Отличие композиции класса от наследования.
6	Тема 6. Наследование в ООП Python. Делегирование и доступ к классам при наследовании. - Функция <code>super()</code> для применения при наследовании классов. - Делегирование атрибутов классов. - Наследование от встроенных типов <code>type</code> и <code>object</code> .
7	Тема 7. Свойства классов в ООП Python. Рассматриваемые вопросы: - Доступ к привычным методам наследованных классов. - Реализация свойств ( <code>property</code> ) классов для чтения и записи данных (геттеры и сеттеры). - Оптимизация программирования свойств. - Функция-декоратор <code>@property</code> .
8	Тема 8. Полиморфизм в ООП Python. - Изменение вычислительного результата простых арифметических операций. - Виды полиморфизма. Полиморфизм методов класса. - Переопределение методов суперклассов при наследовании. - Абстрактные классы и полиморфизм. - Создание абстрактных классов и применение их полиморфизма в объектах.

## 4.2. Занятия семинарского типа.

### Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Тема 1. Правила формирования класса для программирования в IDE PyCharm. Отработка навыков создания простых классов и объектов класса. Рассматриваемые вопросы: - Инициализация объекта с помощью метода <code>__init__</code> : назначение, синтаксис, передача параметров, установка начальных значений атрибутов. - Правила оформления классов в PyCharm: именование, структура файлов, автодополнение, подсказки типов ( <code>type hints</code> ), рефакторинг. - Практическая отработка: создание простых классов (например, <code>Book</code> , <code>Student</code> , <code>Car</code> ) с реализацией <code>__init__</code> , <code>__new__</code> и <code>__del__</code> , создание и удаление объектов, наблюдение за порядком вызова методов.
2	Тема 2. Правила использования стандартного первого аргумента <code>self</code> для методов объекта, методов инициализации и конструирования Рассматриваемые вопросы: - Решение задач для понимания инициализации объекта с помощью метода <code>__init__</code> , - конструирования методом <code>new</code> , применение деструкторов класса.
3	Тема 3. Программирование композитных классов. Рассматриваемые вопросы:

№ п/п	Тематика практических занятий/краткое содержание
	<ul style="list-style-type: none"> <li>- Создание экземпляров классов</li> <li>- объединение их поведения с помощью методов (композиция) в различных вычислительных задачах.</li> </ul>
4	<p>Тема 4. Программирование классов с различными уровнями доступа к полям и методам классов. Статические методы класса.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- Правила использования синтаксиса программирования Python для указания уровня видимости атрибутов класса.</li> <li>- Решение задач с закрытыми полями и методами в классе, статическими атрибутами <code>@staticmethod</code> и <code>@classmethod</code>.</li> </ul>
5	<p>Тема 5. Программирование классов с одиночным наследованием.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- Разбор различных задач наследования в классах от одного класса-родителя.</li> <li>- Создание новых классов на основе существующих классов, наследуя их атрибуты и методы.</li> <li>- Дочерний класс (подкласс или производный класс).</li> <li>- Наследование атрибутов и методов родительского класса (суперкласса или базового класса).</li> </ul>
6	<p>Тема 6. Программирование делегирования и доступ к классам при наследовании.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- Использование функции <code>super()</code> для доступа к атрибутам и методам родительского класса из дочернего класса.</li> <li>- Определение атрибутов в родительском классе и их использование в дочернем классе.</li> <li>- Базовое поведение класса при делегировании атрибутов родительского класса.</li> <li>- Расширение поведения программы дочерними классами.</li> <li>- Наследование от встроенных типов, таких как <code>int</code>, <code>str</code> и <code>list</code>. Наследование от <code>type</code> и <code>object</code>.</li> </ul>
7	<p>Тема 7. Реализация свойств классов (<code>property</code>), чтения и записи данных (геттеры и сеттеры).</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- Программирование свойств классов с использованием декоратора <code>property</code> и геттеров/сеттеров, а также применение операторов Python, таких как <code>is</code>, <code>not</code>, <code>and</code>, <code>or</code>, <code>not in</code>, <code>in</code>, <code>==</code>, <code>!=</code>, <code>&gt;</code>, <code>&lt;</code>, <code>&gt;=</code>, <code>&lt;=</code>, контроль доступ к атрибутам класса, обработка операции чтения и записи данных в/из свойств.</li> <li>- Программирование дескрипторов свойств класса.</li> </ul>
8	<p>Тема 8. Реализация полиморфизма в классах.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- Техника использования одного и того-же кода для разных типов данных, изменение вычислительного результата наследуемых методов.</li> <li>- Программирование структурного, функционального и параметрического полиморфизма.</li> <li>- Абстрактные классы - базовый класс с абстрактными методами, их реализация в производных классах.</li> </ul>
9	<p>Тема 9. Программирование множественного наследования в классах. Техника создания кода типа «<code>mixin</code>».</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- Наследование от нескольких суперклассов для создания более сложных и гибких программных систем.</li> <li>- Понимание работы алгоритма C3 MRO и его применение для эффективного использования множественного наследования.</li> <li>- Программирование расширения функциональности класса без переопределения его атрибутов при множественном наследовании.</li> <li>- Создание новой функциональности класса не изменяя его код и не влияя на поведение других классов, которые наследуются от него.</li> </ul>

№ п/п	Тематика практических занятий/краткое содержание
10	Тема 10. Программирование методов перегрузки операций в классах. Рассматриваемые вопросы: - Создание собственных итерируемых классов методом <code>__next__</code> , - Создание собственных итерируемых классов методом <code>__iter__</code> .
11	Тема 11. Программирование методов перегрузки операций в классах. Рассматриваемые вопросы: - Обработка вызовов свойств методами перегрузки операций <code>__getattr__</code> , <code>__setattr__</code> , <code>__delattr__</code> . - Перегрузка операций вывода результатов, приведение к строке методами <code>__str__</code> и <code>__repr__</code> .
12	Тема 12. Создание декораторов для методов и классов. Рассматриваемые вопросы: - Рассмотрение различных способов создания объектов-декораторов, включая анонимные функции, функции-обертки и классы-обертки. - Практика использования метода <code>_call__</code> для вызова объекта-декоратора как функции.
13	Тема 13. Разработка простых метаклассов. Рассматриваемые вопросы: - Создание метакласса, записывающий в файл данные об его использовании другими классами. - Создание метакласса, который проверяет класс на запрет использования цифр в именах атрибутов и методов. - Создание функции, которая создает класс, на основе переданных ей названия, атрибутов и методов. - Создание классов ORM для работы с базой данных на основе метакласса.
14	Тема 14. Моделирование движения транспортного средства технологией объектно-ориентированного программирования. Рассматриваемые вопросы: - Разработка кода ООП по формулам расчета для моделирования ускорения автомобилей различных видов в наследуемых классах.

#### 4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Работа с учебной литературой
2	Участие в онлайн-конференциях и мастер-классах
3	Поиск алгоритмов обработки данных в открытых источниках
4	Подготовка к практическим занятиям
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	Городняя, Л. В. Парадигма программирования : учебное пособие для вузов / Л. В. Городняя. — 2-е	<a href="https://e.lanbook.com/book/151660">https://e.lanbook.com/book/151660</a>

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

<https://journal.tinkoff.ru/short/ai-for-all/> - база данных нейронных сетей

<https://vc.ru/services/916617-luchshie-neyroseti-bolshaya-podborka-iz-top-200-ii-generatorov-po-kategoriyam> - база данных нейронных сетей

<https://github.com/abalmumcu/bert-rest-api> - профессиональная платформа для командой работы над проектов (нейронная сеть bert)

<http://library.miit.ru/> - электронно-библиотечная система Научно-технической библиотеки МИИТ

<https://proglib.io/p/raspoznavanie-obektov-s-pomoshchyu-yolo-v3-na-tensorflow-2-0-2020-11-08> - профессиональная библиотека программистов

[https://yandex.cloud/ru/blog/posts/2022/12/andrey-berger-and-yandex-cloud?utm\\_referrer=https%3A%2F%2Fyandex.ru%2F](https://yandex.cloud/ru/blog/posts/2022/12/andrey-berger-and-yandex-cloud?utm_referrer=https%3A%2F%2Fyandex.ru%2F) – библиотека профессиональных статей разработчиков Яндекс

<https://yandex.cloud/ru/blog> - библиотека профессиональных статей разработчиков Яндекс

<https://tproger.ru/translations/opencv-python-guide> - библиотека основных команд OpenCV

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

Microsoft Office 2007

VS code

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

Компьютер преподавателя

Компьютеры студентов

экран для проектора, маркерная доска,

Проектор

9. Форма промежуточной аттестации:

Зачет в 3 семестре.

Экзамен в 4 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