МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Лабораторная работа №5

**Работа с классами**

**по дисциплине**

**«Введение в информационные технологии»**

Выполнил: студент гр. БВТ2403

Косарев С.А.

Проверил:

Москва, 2024 г.

**Цель работы**: Получить практический опыт работы с ООП в Python.

**Оборудование:** компьютер с установленным программным обеспечением Python.

**Методы и технологии:**

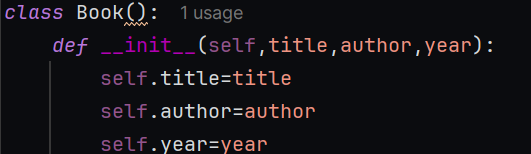
Основные функции, использованные в проекте, включают:

1. Для создания класса был использован метод class. Для инициализации переменных в классе была использована функция \_\_init\_\_.
2. Для забивания переменных в класс был использован self. Для создания функции get\_info() используется def get\_info(). Также была использована f-строка для вывода выражения с использованием переменной. Для вывода значений используется print. Для изменения заданной переменной, сохраненной в класс, использованы @property и @...setter

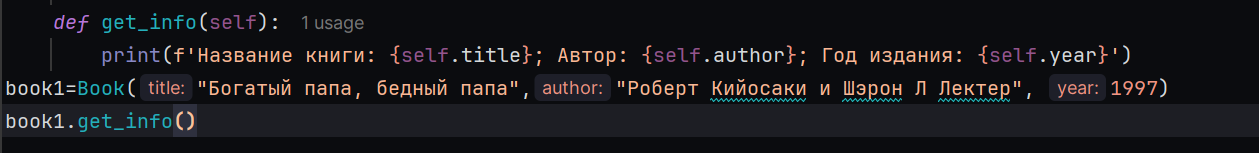
Реализация:

**Задание 1: Базовый класс и методы**

Для создания класса используем class Book, затем для инициации характеристик класса пишем функцию \_\_init\_\_ и записываем характеристики. После функции нужно описать переменные, которые принимает функция, обращаясь к self.

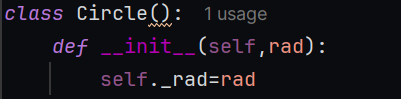


Создаем функцию через def и прописываем вывод переменной, используя f-строку, для удобства. Затем добавляем элемент класса Book и вписываем в него нужные характеристики, в соответствии с заданием. Выводим нужную нам функцию.

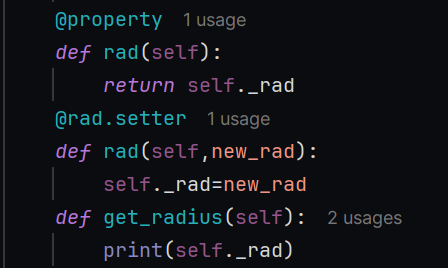


**Задание 2: Работа с конструктором**

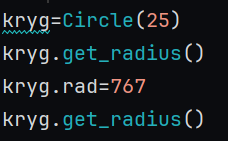
Для определения класса Circle прописываем class Circle(). После, заносим в класс характеристику радиуса с помощью \_\_init\_\_.



Нам всё ещё нужно получать доступ к атрибутам, поэтому мы предоставляем его через @property и объявляем для каждого атрибута метод изменения @...setter. Прописываем функцию вывода радиуса через def и print.



Создаем элемент и вносим длину радиуса. Выводим радиус через функцию get\_radius. Изменяем длину радиуса и выводим новую длину.



**Вывод:** Мы получили практический опыт работы с ООП в Python.