

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное
образовательное учреждение высшего образования**

«Московский технический университет связи и информатики»

Кафедра “Математическая кибернетика и информационные технологии”

ОТЧЕТ

по лабораторной работе №5

по дисциплине «Введение в информационные технологии»

Тема: «Работа с классами»

Выполнил: студент группы БВТ2505

Вакалюк А.А.

Проверил: Павликов. А.Е.

Москва, 2025

Цель работы:

Получить практический опыт работы с ООП в Python

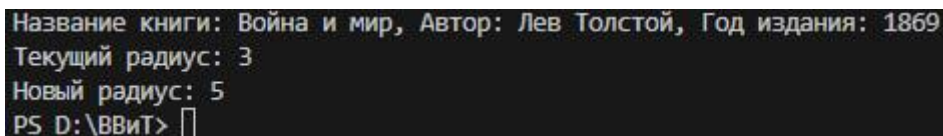
Задание 1:

1. Определить класс Book, который имеет три атрибута: title (название), author (автор), и year (год издания).
2. Добавить метод get_info(), который возвращает информацию о книге в формате: "Название книги: [title], Автор: [author], Год издания: [year]".

Задание 2:

1. Определить класс Circle для представления круга.
2. Использовать конструктор __init__ для инициализации радиуса круга (radius).
3. Добавить метод get_radius(), который возвращает значение радиуса.
4. Добавить метод set_radius(new_radius), который позволяет изменить радиус круга.
5. Создать объект класса Circle, измените его радиус и выведите новый радиус на экран

Скриншоты выполнения:



```
Название книги: Война и мир, Автор: Лев Толстой, Год издания: 1869
Текущий радиус: 3
Новый радиус: 5
PS D:\ВВит>
```

Исходный код программы:

```
class Book: def __init__(self, title, author, year): self.title = title self.author = author self.year = year

def get_info(self):
    return f'Название книги: {self.title}, Автор: {self.author}, Год
издания: {self.year}'

book = Book("Война и мир", "Лев Толстой", 1869) print(book.get_info())

class Circle: def __init__(self, radius): self.radius = radius

def get_radius(self):
    return self.radius

def set_radius(self, new_radius):
    self.radius = new_radius

circle = Circle(3) print(f"Текущий радиус:", circle.get_radius()) circle.set_radius(5)
print(f"Новый радиус:", circle.get_radius())
```

Заключение

В ходе лабораторной работы были изучены основные принципы объектно-ориентированного программирования в Python. Были созданы классы с атрибутами и методами, освоены механизмы работы с конструкторами, а также реализованы методы для получения и изменения значений атрибутов.

На примере классов Book и Circle была закреплена структура создания собственных объектов, а также применение методов для работы с данными внутри класса.

В результате выполнения работы были получены практические навыки определения классов, использования конструктора `init`, создания объектов и взаимодействия с их атрибутами и методами.