

**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
Хардиков Владислав Дмитриевич
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:

```
PS F:\Projects> & C:/Users/gmuse/AppData/Local/Microsoft/Windows/
enter title of this book:скокото там градусов по кельвину
enter author of this book:я
enter year of publishing of this book:2020
Title: скокото там градусов по кельвину, Author: я, Year: 2020
PS F:\Projects>
```

Задание 1. Результат работы

```
PS F:\Projects> & C:/Users/gmuse/AppData/Local
Enter the initial radius of the circle: 5
Initial radius: 5.0
Enter the new radius of the circle: 6
Updated radius: 6.0
PS F:\Projects>
```

Задание 2. Результат работы

Исходный код программы:

```
# Задание 1: Базовый класс и методы

class Book:

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

    def display_info(self):
        return ("Title: " + self.title + ", Author: " + self.author + ", Year: " + str(self.year))
```

```
title = input('enter title of this book:')

author = input('enter author of this book:')

while True:

    year = input('enter year of publishing of this book:')

    try:

        year_int = int(year)

        if 1000 <= year_int <= 2025:

            break

        else:

            print('Please enter a valid year between 1000 and 2025.')

    except ValueError:

        print('Please enter a valid integer for the year.')

    year = str(year_int)

    book1 = Book(title, author, year)

    print(book1.display_info())

# Задание 2: Работа с конструктором

class Circle:

    def __init__(self, radius):

        self.radius = radius

    def get_radius(self):

        return self.radius

    def set_radius(self, radius):

        self.radius = radius
```

```
while True:

    try:

        initial_radius = float(input('Enter the initial radius of the circle: '))

        if initial_radius > 0:

            break

        else:
```

```
    print('Radius must be a positive number.')
except ValueError:
    print('Please enter a valid number for the radius.')
circle1 = Circle(initial_radius)

print(f'Initial radius: {circle1.get_radius()}')

while True:
    try:
        new_radius = float(input('Enter the new radius of the circle: '))
        if new_radius > 0:
            break
        else:
            print('Radius must be a positive number.')
    except ValueError:
        print('Please enter a valid number for the radius.')
circle1.set_radius(new_radius)
print(f'Updated radius: {circle1.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.