# **Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования «Московский технический университет связи и информатики»**.

# Лабораторная работа №5:

**Работа с классами**

Выполнил: Безматерных Иван Алексеевич БВТ2402

Проверил:

Москва 2024

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

**Задание 1:**

1. Определите класс Book, который имеет три атрибута: **title** (название), **author** (автор), и **year** (год издания).

**Ход работы:**

1. Определение класса: Класс Book определяется с использованием ключевого слова class.  
2. Инициализатор (\_\_init\_\_ метод): Метод \_\_init\_\_ инициализирует атрибуты экземпляра класса. Мы принимаем три аргумента — title, author, и year — и присваиваем их атрибутам объекта.  
3. Метод \_\_str\_\_: Этот метод позволяет удобно выводить информацию о книге в читаемом формате. Он будет вызван при использовании функции print() для экземпляра класса Book.

**Код программы:**

class Book:

def\_init\_(self, title, author, year):

self.title = title #Название книги

self.author = author #Автор книги

self.year = year #Год издания

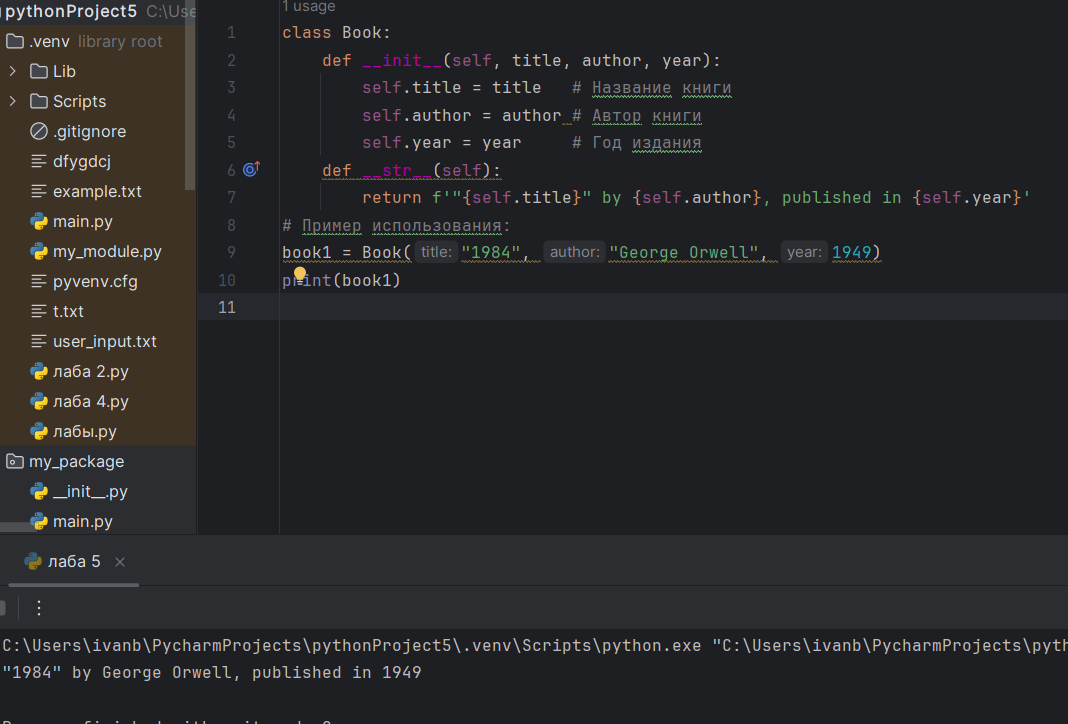
def\_str\_(self):

return f’’{self.title}’’ by {self.author}, published in {self.year}’

#Пример использования

book1 = Book(‘’1984’’, ‘’George Orwell’’, 1949)

print(book1)



1. Добавьте метод get\_info(), который возвращает информацию о книге в формате: "Название книги: [**title**], Автор: [**author**], Год издания: [**year**]".

**Ход работы:**

1. Определение класса: Класс Book определяется с использованием ключевого слова class.  
2. Инициализатор (\_\_init\_\_ метод): Метод \_\_init\_\_ инициализирует атрибуты экземпляра класса. Мы принимаем три аргумента — title, author, и year — и присваиваем их атрибутам объекта.  
3. Метод \_\_str\_\_: Этот метод позволяет удобно выводить информацию о книге в читаемом формате. Он будет вызван при использовании функции print() для экземпляра класса Book.

**Код программы:**

class Book:

def\_init\_(self, title, author, year):

self.title = title #Название книги

self.author = author #Автор книги

self.year = year #Год издания

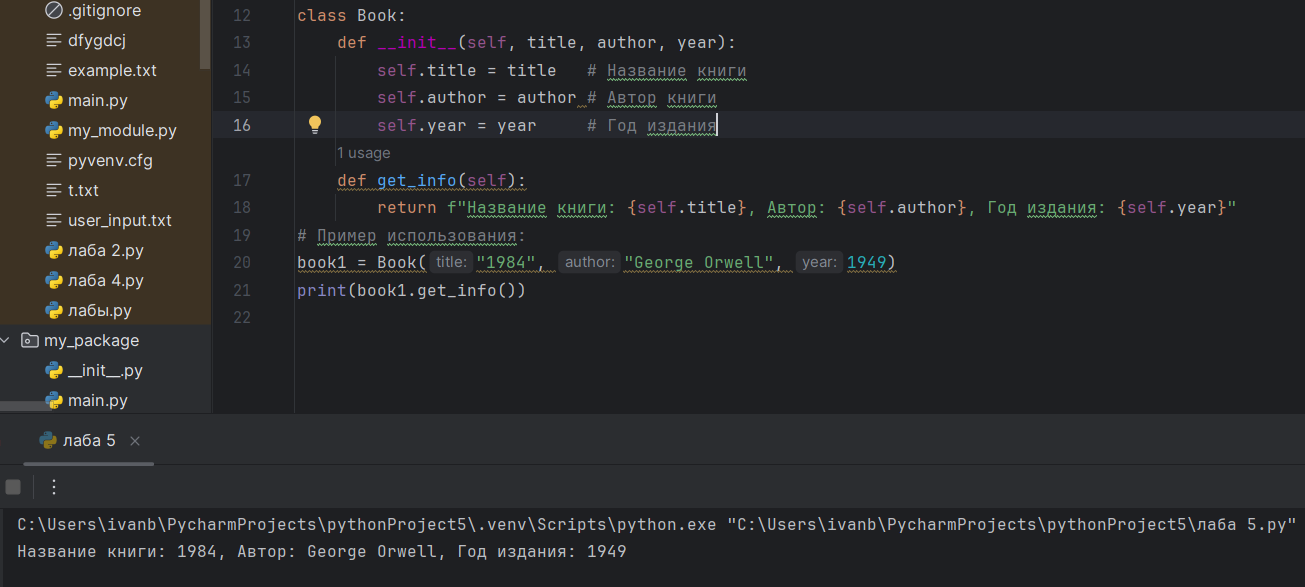
def\_str\_(self):

return f’’{self.title}’’ by {self.author}, published in {self.year}’

#Пример использования

book1 = Book(‘’1984’’, ‘’George Orwell’’, 1949)

print(book.get\_info())

****

**Задание 2:**

1. Определите класс **Circle** для представления круга.
2. Используйте конструктор **\_\_init\_\_** для инициализации радиуса круга (**radius**).
3. Добавьте метод **get\_radius()**, который возвращает значение радиуса.
4. Добавьте метод **set\_radius(new\_radius)**, который позволяет изменить радиус круга.
5. Создайте объект класса **Circle**, измените его радиус и выведите новый радиус на экран.

**Ход работы:**

1. Конструктор \_\_init\_\_: Мы определяем атрибут radius, который представляет радиус круга.  
2. Метод area(): Этот метод вычисляет и возвращает площадь круга по формуле \( \pi r^2 \).  
3. Метод circumference(): Этот метод вычисляет и возвращает длину окружности круга по формуле \( 2\pi r \).  
4. Метод get\_info(): Этот метод возвращает строку с информацией о круге, включая радиус, площадь и длину окружности.  
5. Пример использования: Мы создаем экземпляр класса Circle с радиусом 5 и выводим информацию о круге с помощью метода get\_info().

**Код программы:**

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  
# Создаем объект класса Circle  
my\_circle = Circle(5)  
# Получаем текущий радиус  
print("Текущий радиус круга:", my\_circle.get\_radius())  
# Изменяем радиус  
my\_circle.set\_radius(7)  
# Получаем и выводим новый радиус  
print("Новый радиус круга:", my\_circle.get\_radius())

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

#Создаём объект класса Circle

my\_circle = Circle(5)

#Получаем текущий радиус

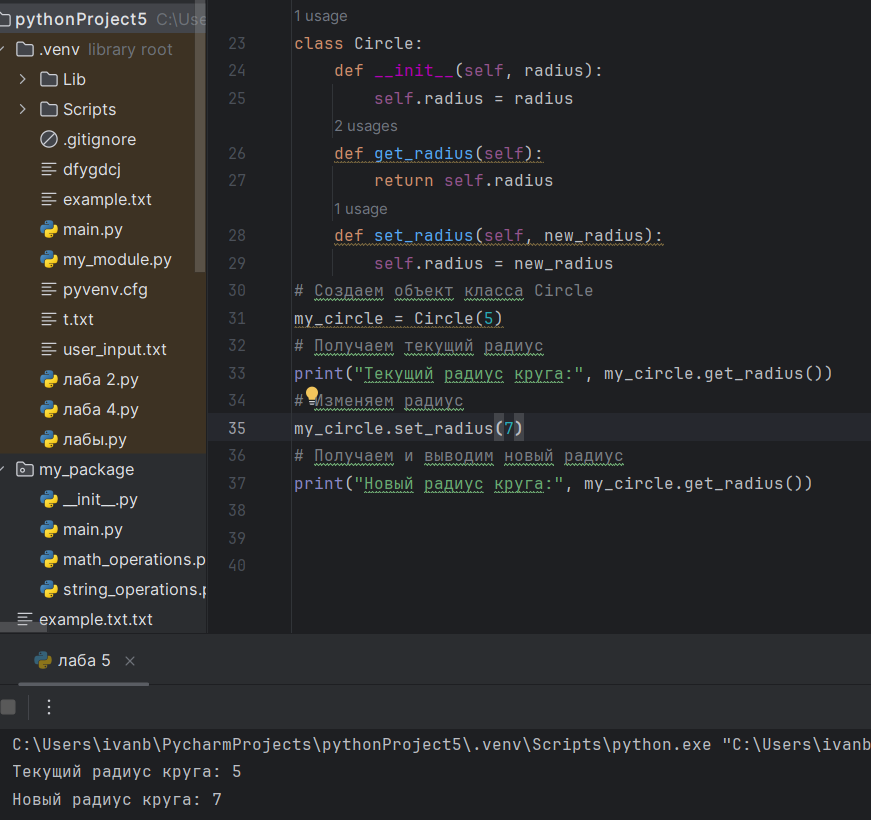
print(‘’Текущий радис круга:’’, my\_circle.get\_radius())

#Изменяем радиус

my\_circle.set\_radius(7)

#Получаем и выводим новый радиус

print(‘’Новый радиус круга:’’, my\_circle.get\_radius())



**Вывод:** я получил практический опыт работы с ООП в Python.