

Защищено:  
Гапанюк Ю.Е.

"\_\_" 2025 г.

Демонстрация:  
Гапанюк Ю.Е.

"\_\_" 2025 г.

**Отчет по лабораторной работе № 2 по курсу  
Парадигмы и конструкции языков программирования  
ГУИМЦ**

**Тема работы: "Объектно-ориентированные возможности языка  
Python."**

5  
(количество листов)

ИСПОЛНИТЕЛЬ:

студент группы ИУ5Ц-54Б

Щетинин Д.С.

\_\_\_\_\_

(подпись)

"\_\_" 2025 г.

# 1. Тема и задание для выполнения лабораторной работы.

Тема работы: "Изучение объектно-ориентированных возможностей языка Python."

## Задание:

1. Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.
2. Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.
3. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.
4. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.
5. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать [здесь](#).
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь](#).
7. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета. Класс должен переопределять метод, вычисляющий площадь фигуры.
8. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля [math](#).
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
  - Определите метод "get", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
  - Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
10. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов (используйте следующую конструкцию - [https://docs.python.org/3/library/\\_main\\_.html](https://docs.python.org/3/library/_main_.html)). Создайте следующие объекты и выведите о них информацию в консоль (N - номер Вашего варианта по списку группы):
  - Прямоугольник синего цвета шириной N и высотой N.
  - Круг зеленого цвета радиусом N.
  - Квадрат красного цвета со стороной N.
  - Также вызовите один из методов внешнего пакета, установленного с использованием pip.

# 2. Листинг программы

## Файл: main.py

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
import numpy as np # пример внешнего пакета

def main():
    N = 5 # номер варианта

    # Создаем объекты
    rectangle = Rectangle(N, N, "синего")
    circle = Circle(N, "зеленого")
    square = Square(N, "красного")

    # Выводим информацию о фигурах
```

```
print(rectangle)
print(circle)
print(square)

# Пример использования внешнего пакета
arr = np.array([1, 2, 3])
print("\nПример работы с внешним пакетом (numpy):")
print("Сумма элементов массива:", np.sum(arr))

if __name__ == "__main__":
    main()
```

**Файл: lab\_python\_oop/circle.py**

```
import math
from geometric_figure import GeometricFigure
from color import Color

class Circle(GeometricFigure):
    FIGURE_TYPE = "Круг"

    def __init__(self, radius, color):
        self.radius = radius
        self.color = Color(color)

    def area(self):
        return math.pi * self.radius ** 2

    def __repr__(self):
        return "{} {} цвета радиусом {}, площадью {:.2f}.".format(
            self.FIGURE_TYPE,
            self.color.color,
            self.radius,
            self.area()
        )
```

@classmethod

```
def get_name(cls):
    return cls.FIGURE_TYPE
```

**Файл: lab\_python\_oop/color.py**

```
class Color:
```

```
    def __init__(self, color):
        self._color = color
```

@property

```
def color(self):
    return self._color
```

@color.setter

```
def color(self, value):
    self._color = value
```

**Файл: lab\_python\_oop/geometric\_figure.py**

```
from abc import ABC, abstractmethod
```

```
class GeometricFigure(ABC):
```

```
    @abstractmethod
```

```
    def area(self):
```

```
        pass
```

```
@classmethod
def get_name(cls):
    return cls.__name__
```

**Файл: lab\_python\_oop/rectangle.py**

```
import sys
import os
sys.path.append(os.path.dirname(os.path.abspath(__file__)))
from geometric_figure import GeometricFigure
from color import Color

class Rectangle(GeometricFigure):
    FIGURE_TYPE = "Прямоугольник"

    def __init__(self, width, height, color):
        self.width = width
        self.height = height
        self.color = Color(color)

    def area(self):
        return self.width * self.height

    def __repr__(self):
        return "{} {} цвета шириной {} и высотой {}, площадью {}".format(
            self.FIGURE_TYPE,
            self.color.color,
            self.width,
            self.height,
            self.area()
        )

@classmethod
def get_name(cls):
    return cls.FIGURE_TYPE
```

**Файл: lab\_python\_oop/square.py**

```
from .rectangle import Rectangle

class Square(Rectangle):
    FIGURE_TYPE = "Квадрат"

    def __init__(self, side, color):
        super().__init__(side, side, color)

    def __repr__(self):
        return "{} {} цвета со стороной {}, площадью {}".format(
            self.FIGURE_TYPE,
            self.color.color,
            self.width,
            self.area()
        )

@classmethod
def get_name(cls):
    return cls.FIGURE_TYPE
```

### **3. Результаты работы программы**

C:\Users\Денис\Desktop\Python-2lab\venv\Scripts\python.exe C:\Users\Денис\Desktop\Python-2lab\main.py  
Прямоугольник синего цвета шириной 5 и высотой 5, площадью 25.

Круг зеленого цвета радиусом 5, площадью 78.54.

Квадрат красного цвета со стороной 5, площадью 25.

Пример работы с внешним пакетом (numpy):

Сумма элементов массива: 6