

**Московский государственный технический  
университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»**

**Курс «Парадигмы и конструкции языков программирования»**

**Отчет по лабораторной работе №2**

Выполнил:  
студент группы ИУ5-35Б:

Аннакулиева Д.Д.  
Подпись и дата:

Проверил:  
преподаватель каф.  
ИУ5  
Гапанюк Ю.  
Подпись и дата:

Москва, 2023 г.

## 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 `repr`, 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>). 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.

## Код:

### square.py

```
from lab_python_oop.rectangle import Rectangle

class Square(Rectangle):
    FIGURE_TYPE = "Square"

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

    def __repr__(self):
        return 'type: { } color: { } side: { } area: {}'.format(
            self.FIGURE_TYPE,
            self.color.color,
            self.width,
            self.area()
        )
```

### rectnngle.py

```
from lab_python_oop.geometry import Geometry
from lab_python_oop.color import Color

class Rectangle(Geometry):
    FIGURE_TYPE = "Rectangle"

    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 'type: { } color: { } width: { } height: { } area: {}'.format(
            self.FIGURE_TYPE,
            self.color.color,
            self.width,
            self.height,
            self.area()
        )
```

### main.py

```
import sys
sys.path.append('/Users/annakulievadiana/Desktop/lab2/')

import numpy as np
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
```

```

def main():
    rectangle = Rectangle(3, 9, "blue")
    circle = Circle(3, "green")
    square = Square(3, "red")

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

    print(np.version.version)

if __name__ == "__main__":
    main()

```

```

from abc import ABC, abstractmethod

```

```

class Geometry(ABC):
    @abstractmethod
    def area(self):
        pass...

```

### **color.py**

```

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

```

### **circle.py**

```

import math
from lab_python_oop.geometry import Geometry
from lab_python_oop.color import Color

```

```

class Circle(Geometry):
    FIGURE_TYPE = "Circle"

    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 'type: {} color: {} radius: {} area: {}'.format(
            self.FIGURE_TYPE,
            self.color.color,
            self.radius,
            self.area()
        )

```

