

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

Факультет «Радиотехнический»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчёт по лабораторной работе №2  
«Объектно-ориентированные возможности языка Python»

Выполнил:

студент группы РТ5-31Б  
Петров Егор

Подпись и дата:

Проверил:

преподаватель каф. ИУ5  
Гапанюк Ю.Е.

Подпись и дата:

Москва, 2021 г

## Описание задания

### 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.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.
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 `getr`, который возвращает в виде строки основные параметры фигуры, ее цвет и площадь.
  - 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. 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`.

# Текст программы

## main.py:

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
```

```
def main():
    r = Rectangle("синего", 2, 2)
    c = Circle("зеленого", 2)
    s = Square("красного", 2)
    print(r)
    print(c)
    print(s)
    print(Circle.get_figure_type())
```

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

## figure.py

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

```
class Figure(ABC):
    """
    Абстрактный класс «Геометрическая фигура»
    """
    @abstractmethod
    def square(self):
        """
        содержит виртуальный метод для вычисления площади фигуры.
        """
        pass
```

## color.py

```
class FigureColor:
    """
    Класс «Цвет фигуры»
    """

    def __init__(self): # конструктор класса
        self._color = None

    @property
    def colorproperty(self):
        """
        Get-аксеccop
        """
        return self._color

    @colorproperty.setter
    def colorproperty(self, value):
        """
        Set-аксеccop
        """
        self._color = value
```

## rectangle.py

```
from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor

class Rectangle(Figure):
    """
    Класс «Прямоугольник» наследуется от класса «Геометрическая фигура».
    """
    FIGURE_TYPE = "Прямоугольник"

    @classmethod
    def get_figure_type(cls): # метод для возврата типа фигуры
        return cls.FIGURE_TYPE

    def __init__(self, color_param, width_param, height_param): # конструктор класса
        """
        Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В
        конструкторе создается объект класса «Цвет фигуры» для хранения цвета.
        """
        self.width = width_param
        self.height = height_param
        self.fc = FigureColor()
        self.fc.colorproperty = color_param

    def square(self):
        """
        Класс должен переопределять метод, вычисляющий площадь фигуры.
        """
        return self.width*self.height

    def __repr__(self): # метод отвечает за строковое представление объекта
        return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(
            Rectangle.get_figure_type(),
            self.fc.colorproperty,
            self.width,
            self.height,
            self.square()
        )
```

## circle.py

```
from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor
import math

class Circle(Figure):
    """
    Класс «Круг» наследуется от класса «Геометрическая фигура».
    """
    FIGURE_TYPE = "Круг"

    @classmethod
    def get_figure_type(cls): # метод для возврата типа фигуры
        return cls.FIGURE_TYPE

    def __init__(self, color_param, r_param): # конструктор класса
        """
        Класс должен содержать конструктор по параметрам «радиус» и «цвет». В конструкторе
        создается объект класса «Цвет фигуры» для хранения цвета.
        """
        self.r = r_param
        self.fc = FigureColor()
        self.fc.colorproperty = color_param
```

```

def square(self):
    """
    Класс должен переопределять метод, вычисляющий площадь фигуры.
    """
    return math.pi*(self.r**2)

def __repr__(self): # метод отвечает за строковое представление объекта
    return '{ } {} цвета радиусом {} площадью {}'.format(
        Circle.get_figure_type(),
        self.fc.colorproperty,
        self.r,
        self.square()
    )

```

## square.py

```

from lab_python_oop.rectangle import Rectangle

```

```

class Square(Rectangle):
    """
    Класс «Квадрат» наследуется от класса «Прямоугольник».
    """
    FIGURE_TYPE = "Квадрат"

    @classmethod
    def get_figure_type(cls): # метод для возврата типа фигуры
        return cls.FIGURE_TYPE

    def __init__(self, color_param, side_param): # конструктор класса
        """
        Класс должен содержать конструктор по параметрам «сторона» и «цвет».
        super() организует доступ к унаследованным элементам
        """
        self.side = side_param
        super().__init__(color_param, self.side, self.side)

    def __repr__(self): # метод отвечает за строковое представление объекта
        return '{ } {} цвета со стороной {} площадью {}'.format(
            Square.get_figure_type(),
            self.fc.colorproperty,
            self.side,
            self.square()
        )

```

## Примеры работы программы

```
Прямоугольник синего цвета шириной 24 и высотой 2 площадью 48.  
Круг зеленого цвета радиусом 24 площадью 1809.5573684677208.  
Квадрат красного цвета со стороной 24 площадью 576.  
Квадрат  
  
Process finished with exit code 0
```