

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

**Факультет ИУ  
Кафедра ИУ5**

**Курс «Основы информатики»  
Отчет по лабораторной работе  
№4(2)**

Выполнил студент группы ИУ5-33Б:  
Хасанова К.М.  
Подпись и дата:

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

## Описание задания

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.

## Текст программы

### figure.py

```
from abc import ABC, abstractmethod

class Figure(ABC):
    @abstractmethod
    def area(self):
        pass

    @classmethod
    @abstractmethod
    def name(cls):
        pass
```

### 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
```

### rectangle.py

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

class Rectangle(Figure):
    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

@classmethod
def name(cls):
    return cls.FIGURE_TYPE

def __repr__(self):
    return "Фигура: {}, цвет: {}, ширина: {}, высота: {}, площадь: {:.2f}".format(
        self.name(), self.color.color, self.width, self.height, self.area()
    )

```

### square.py

```

from lab_python_oop.rectangle import Rectangle

class Square(Rectangle):
    FIGURE_TYPE = "Квадрат"

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

    @classmethod
    def name(cls):
        return cls.FIGURE_TYPE

```

### circle.py

```

from lab_python_oop.figure import Figure
from lab_python_oop.color import Color
import math

class Circle(Figure):
    FIGURE_TYPE = "Круг"

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

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

    @classmethod
    def name(cls):
        return cls.FIGURE_TYPE

    def __repr__(self):
        return "Фигура: {}, цвет: {}, радиус: {}, площадь: {:.2f}".format(
            self.name(), self.color.color, self.radius, self.area()
        )

```

### 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

def main():
    n = 22

    # Создание объектов
    rect = Rectangle(n, n, "синий")
    circle = Circle(n, "зеленый")
    square = Square(n, "красный")

    # Вывод информации о фигурах
    print(rect)
    print(circle)
    print(square)

    # Использование внешнего пакета numpy
    arr = np.array([1, 2, 3, 4, 5])
    print("Использование numpy: сумма элементов массива {} равна {}".format(arr,
np.sum(arr)))

if __name__ == "__main__":
    main()

```

## Выполнение программы

```

Фигура: Прямоугольник, цвет: синий, ширина: 22, высота: 22, площадь: 484.00
Фигура: Круг, цвет: зеленый, радиус: 22, площадь: 1520.53
Фигура: Квадрат, цвет: красный, ширина: 22, высота: 22, площадь: 484.00
Использование numpy: сумма элементов массива [1 2 3 4 5] равна 15

Process finished with exit code 0

```