



**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

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

**Якимова Татьяна ИУ5-35Б  
Парадигмы и конструкции языков программирования**

**ОТЧЁТ ПО  
Лабораторной работе №2**

Москва  
2023

## Задание:

Создать пакет "lab\_python\_oop" и внутри него создать следующие файлы и классы:

- figure.py: создать абстрактный класс "Геометрическая фигура" с абстрактным методом для вычисления площади.
- color.py: создать класс "Цвет фигуры" со свойством для описания цвета геометрической фигуры.
- rectangle.py: создать класс "Прямоугольник", который наследуется от "Геометрической фигуры". Включить конструктор, переопределить метод вычисления площади и определить метод герг для возврата информации о прямоугольнике.
- circle.py: создать класс "Круг" аналогично "Прямоугольнику", задав параметр "радиус" и использовать math.pi для вычисления площади.
- square.py: создать класс "Квадрат", который наследуется от "Прямоугольника". Включить конструктор по длине стороны.

В каждом из классов "Прямоугольник", "Круг" и "Квадрат" определить метод герг, который будет возвращать информацию о фигуре, ее цвете и площади в виде строки.

В главном файле "main.py" создать объекты этих классов, установить им цвет и вывести информацию о них с помощью метода герг.

Запустить консольное приложение, чтобы убедиться, что оно работает правильно.

Дополнительное задание. Протестировать корректность работы программы с помощью модульного теста.

## Текст программы.

### Файл circle.py

```
from figure import Figure
from color import Figure_color
import math

class Circle(Figure):

    FIGURE_TYPE = "Круг"

    @classmethod
    def get_figure_type(cls):
```

```

        return cls.FIGURE_TYPE

    def __init__(self, color_param, r_param):
        """
        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 = Figure_color()
        self.fc.color_property = color_param

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

    def __repr__(self):
        return '{} цвета {} радиусом {} площадью {}'.format(
            Circle.get_figure_type(),
            self.fc.color_property,
            self.r,
            self.square()
        )

```

## Файл color.py

```

class Figure_color:

    def __init__(self):
        self._color = None

    @property
    def color_property(self):
        return self._color

    @color_property.setter
    def color_property(self, value):
        self._color = value

```

## Файл figure.py

```

from abc import ABC, abstractmethod

class Figure(ABC):

    @abstractmethod
    def square(self):
        pass

```

## Файл rectangle.py

```

from lab2.figure import Figure
from lab2.color import Figure_color

```

```

class Rectangle(Figure):
    figure_type = "Прямоугольник"

    @classmethod
    def get_figure_type(self):
        return self.figure_type

    def __init__(self, color_param, width_param, height_param):
        self.width = width_param
        self.height = height_param
        self.fc = Figure_color()
        self.fc.color_property = color_param

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

    def __repr__(self):
        return '{} цвета {} шириной {} и высотой {} площадью {}'.format(
            Rectangle.get_figure_type(),
            self.fc.color_property,
            self.width,
            self.height,
            self.square()
        )

```

## Файл square.py

```

from lab2.rectangle import Rectangle

class Square(Rectangle):
    figure_type = "Квадрат"

    @classmethod
    def get_figure_type(self):
        return self.figure_type

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

    def __repr__(self):
        return '{} цвета {} со стороной {} площадью {}'.format(
            Square.get_figure_type(),
            self.fc.color_property,
            self.side,
            self.square()
        )

```

## Файл main.py

```

from rectangle import Rectangle
from circle import Circle
from square import Square

def main():
    r = Rectangle("red", 2, 2)
    c = Circle("cyan", 2)
    s = Square("green", 2)

```

```
print(r)
print(c)
print(s)

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

## Результат выполнения

```
Прямоугольник цвета red шириной 2 и высотой 2 площадью 4.
Круг цвета cyan радиусом 2 площадью 12.566370614359172.
Квадрат, цвета green со стороной 2 площадью 4.
```