

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

Факультет Радиотехнический
Кафедра РТ5

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

Выполнил:

студент группы РТ5-31Б:

Бабасанова Н. С.

Руководитель:

преподаватель каф. ИУ5

Гапанюк Ю. Е.

Москва, 2023г.

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 `"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). 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.

11. Дополнительное задание. Протестируйте корректность работы Вашей программы с помощью модульного теста.

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(): # var 2
    r = Rectangle("синего", 2, 2)
    c = Circle("зеленого", 2)
    s = Square("красного", 2)
    print(r)
    print(c)
    print(s)
```

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

- circle.py

```
from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor
from math import pi # math многовато, поэтому ! пи
```

```
class Circle(Figure):
    """
```

Класс «Круг» наследуется от класса «Геометрическая фигура».

"""

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 = FigureColor()

self.fc.colorproperty = color_param

def area(self):

"""

Класс должен переопределять метод, вычисляющий площадь фигуры.

"""

return pi * (self.r ** 2)

def __repr__(self):

return '{} {} цвета радиусом {} площадью {}'.format(

Circle.get_figure_type(),

self.fc.colorproperty,

self.r,

round(self.area(), 3) # округляем до 3-х цифр после запятой

)

- color.py

class FigureColor: # Класс «Цвет фигуры»

def __init__(self):

self._color = None

@property

def colorproperty(self): # Getter

return self._color

@colorproperty.setter

def colorproperty(self, value): # Setter

self._color = value

- figure.py

from abc import ABC, abstractmethod # abc = abstract

```
class Figure(ABC): # Абстрактный класс «Геометрическая фигура»
    @abstractmethod
    def area(self):
        pass
```

- 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):
        return cls.FIGURE_TYPE

    def __init__(self, color_param, width_param, height_param):
        """
        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 area(self):
        """
        Класс должен переопределять метод, вычисляющий площадь фигуры.
        """
        return self.width * self.height

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

- 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):
        return cls.FIGURE_TYPE

    def __init__(self, color_param, side_param):
        """
        Класс должен содержать конструктор по параметрам «сторона» и «цвет».
        """
        self.side = side_param
        super().__init__(color_param, self.side, self.side)

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

```

- test_main.py
- # Тест класса Rectangle
- from lab_python_oop.rectangle import *
-
- def test_main():
 - string1 = "Прямоугольник синего цвета шириной 2 и высотой 2 площадью 4."
 - string2 = Rectangle("синего", 2, 2).__str__()
 - assert string2 == string1

Результаты вывода:

- Вывод программы:

Прямоугольник синего цвета шириной 2 и высотой 2 площадью 4.
 Круг зеленого цвета радиусом 2 площадью 12.566.
 Квадрат красного цвета со стороной 2 площадью 4.

- Проверка корректности работы программы:

```
===== test session starts =====
platform win32 -- Python 3.11.4, pytest-7.4.3, pluggy-1.3.0 -- C:\Users\nanny\Desktop\git\Lab2\venv\Scripts\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\nanny\Desktop\git\Lab2
collected 1 item

test_main.py::test_main PASSED [100%]

===== 1 passed in 0.02s =====
```