

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: Кудрявцев Р. В.
Студент группы: РТ5-31Б

Проверяющий: Гапанюк Ю.Е.
Доцент кафедры ИУ5

Москва, 2023 г.

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.

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

Текст программы

figure.py

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

```
class Figure(ABC):
    @abstractmethod
    def square(self):
        pass
```

color.py

```
class FigureColor:

    def __init__(self):
        self._color = None

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

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

rectangle.py

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

```
class Rectangle(Figure):
    FIGURE_TYPE = "Прямоугольник"

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE
```

```

def __init__(self, color_param, width_param, height_param):
    self.width = width_param
    self.height = height_param
    self.fc = FigureColor()
    self.fc.colorproperty = 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.colorproperty,
        self.width,
        self.height,
        self.square()
    )

```

square.py

```

from lab_python_oop.rectangle import Rectangle

```

```

class Square(Rectangle):
    FIGURE_TYPE = "Квадрат"

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

    def __init__(self, color_param, side_param):
        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.square()
        )

```

circle.py

```

from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor
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):
    self.r = r_param
    self.fc = FigureColor()
    self.fc.colorproperty = color_param

def square(self):
    return math.pi*(self.r**2)

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

```

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", 3, 2)
    c = Circle("зеленого", 5)
    s = Square("красного", 5)
    print(r)
    print(c)
    print(s)

if __name__ == "__main__":
    main()

```

tests.py

```

import unittest

from main import *

class TestSolutions(unittest.TestCase):
    def setUp(self):
        self.rectangle = Rectangle("синего", 3, 2)
        self.circle = Circle("зеленого", 5)
        self.square = Square("красного", 5)
    def test_task1_solution(self):
        result = self.rectangle.__repr__()

```

```

        self.assertEqual(
            result, 'Прямоугольник синего цвета шириной 3 и высотой 2 площадью
6.'
        )

    def test_task2_solution(self):
        result = self.circle.__repr__()
        self.assertEqual(
            result, 'Круг зеленого цвета радиусом 5 площадью 78.53981633974483.'
        )

    def test_task3_solution(self):
        result = self.square.__repr__()
        self.assertEqual(
            result, 'Квадрат красного цвета со стороной 5 площадью 25.'
        )

if __name__ == '__main__':
    unittest.main()

```

Экранные формы с примерами выполнения программы

main.py

Прямоугольник синего цвета шириной 3 и высотой 2 площадью 6.
 Круг зеленого цвета радиусом 5 площадью 78.53981633974483.
 Квадрат красного цвета со стороной 5 площадью 25.

tests.py

Ran 3 tests in 0.002s

OK