

**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:
Пичурин В. Е.

Подпись и дата:

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю. Е.

Подпись и дата:

Москва, 2023

Условие работы:

Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием `pip`.

Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.

Все файлы проекта (кроме основного файла `main.py`) должны располагаться в пакете `lab_python_oop`.

Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета `lab_python_oop`.

Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать [здесь](#).

Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь](#).

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.

Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа `math.pi` из модуля `math`.

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. Используйте метод `format` - <https://pyformat.info/>

Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.

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`.

Код программы:

`geometric_figure.py`

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

```
class GeometricFigure(ABC):
    @abstractmethod
    def calculate_area(self):
        pass
```

color.py

```
class Color:
    def __init__(self, color):
        self.color = color
```

rectangle.py

```
from lab_python_oop.color import Color
from lab_python_oop.geometric_figure import GeometricFigure

class Rectangle(GeometricFigure, Color):
    figure_type = "Прямоугольник"

    def __init__(self, width, height, color):
        GeometricFigure.__init__(self)
        Color.__init__(self, color)
        self.width = width
        self.height = height

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

    def __repr__(self):
        return "{} , {} цвет, шириной {} и высотой {}. Площадь: {}".format(
            self.figure_type,
            self.color,
            self.width,
            self.height,
            self.calculate_area()
        )
```

circle.py

```
import math

from lab_python_oop.color import Color
from lab_python_oop.geometric_figure import GeometricFigure

class Circle(GeometricFigure, Color):
    figure_type = "Круг"

    def __init__(self, radius, color):
        GeometricFigure.__init__(self)
        Color.__init__(self, color)
```

```

        self.radius = radius

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

    def __repr__(self):
        return "{} {}, {} цвет, радиусом {}. Площадь: {}".format(
            self.figure_type,
            self.color,
            self.radius,
            self.calculate_area()
        )

```

square.py

```

from lab_python_oop.rectangle import Rectangle

class Square(Rectangle):
    figure_type = "Квадрат"

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

    def __repr__(self):
        return "{} {}, {} цвет, со стороной {}. Площадь: {}".format(
            self.figure_type,
            self.color,
            self.width,
            self.calculate_area()
        )

```

main.py

```

from art import tprint

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

def main():
    width_height = 5
    radius_side_length = 5

    rectangle = Rectangle(width_height, width_height, "синий")
    circle = Circle(radius_side_length, "зеленый")
    square = Square(radius_side_length, "красный")

    print(rectangle)
    print(circle)
    print(square)

    tprint("Vadim")

```

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

Результат выполнения

Прямоугольник, синий цвет, шириной 5 и высотой 5. Площадь: 25

Круг, зеленый цвет, радиусом 5. Площадь: 78.53981633974483

Квадрат, красный цвет, со стороной 5. Площадь: 25

```
— — — — —
\\ // _ _ _ | | ( ) _ _ _ _
\\ // / _ ' | / _ ' ||| ' _ _ \
 \V/ | ( | | ( | | | | | | |
  \_ / \_ , | \_ , ||| | | | |
```