

Московский государственный технический  
университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №2

Выполнил:

Студент группы ИУ5-53Б

Балабас Анна

Руководители: Гапанюк Ю.Е.

Дата: 24.10.21

Москва, 2021 г.

Задание:

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 "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.

В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов (используйте следующую конструкцию - [https://docs.python.org/3/library/\\_\\_main\\_\\_.html](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.

Figure.py

```
from abc import ABC, abstractmethod

class Figure(ABC):
    """
    Абстрактный класс «Геометрическая фигура»
    """
    @abstractmethod
    def square(self):
        """
        содержит виртуальный метод для вычисления площади фигуры.
        """
    pass
```

circle.py

```
from figure import Figure
from color import FigureColor
import math

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 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.colorproperty,
            self.r,
            self.square()
        )
```

color.py

```

class FigureColor:
    """
    Класс «Цвет фигуры»
    """

    def __init__(self):
        self._color = None

    @property
    def colorproperty(self):
        """
        Get-аксесор
        """
        return self._color

    @colorproperty.setter
    def colorproperty(self, value):
        """
        Set-аксесор
        """
        self._color = value

```

rectangle.py

```

from figure import Figure
from 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 square(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.square()
        )

```

square.py

```

from 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.square()
        )

```

main.py

```

from rectangle import Rectangle
from circle import Circle
from square import Square
import sys
from PyQt5.QtWidgets import QApplication, QWidget

def main():
    r = Rectangle("синего", 2, 2)
    c = Circle("зеленого", 2)
    s = Square("красного", 2)
    print(r)
    print(c)
    print(s)

    app = QApplication(sys.argv)
    w = QWidget()
    w.resize(350,250)
    w.move(300,300)
    w.setWindowTitle('Window')
    w.show()

    sys.exit(app.exec_())
if __name__ == "__main__":
    main()

```

```
Terminal: Local x + v
Collecting PyQt5-sip<13,>=12.8
Collecting PyQt5-sip<13,>=12.8
  Downloading PyQt5_sip-12.9.0-cp37m-cp37m-win_amd64.whl (62 kB)
    |████████████████████| 62 kB 4.5 MB/s
Collecting PyQt5-Qt5>=5.15.2
  Downloading PyQt5_Qt5-5.15.2-py3-none-win_amd64.whl (50.1 MB)
    |████████████████████| 50.1 MB 2.2 MB/s
Installing collected packages: PyQt5-sip, PyQt5-Qt5, pyqt5
Successfully installed PyQt5-Qt5-5.15.2 PyQt5-sip-12.9.0 pyqt5-5.15.6
WARNING: You are using pip version 21.1.2; however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\annab\Downloads\lab2\venv\Scripts\python.exe -m pip install --upgrade pip' command.
PS C:\Users\annab\Downloads\lab2> |
```

```
main x
C:\Users\annab\Downloads\lab2\venv\Scripts\python.exe C:/Users/annab/Downloads/lab2/main.py
Прямоугольник синего цвета шириной 2 и высотой 2 площадью 4.
Круг зеленого цвета радиусом 2 площадью 12.566370614359172.
Квадрат красного цвета со стороной 2 площадью 4.
```

