

Проверил:
Гапанюк Ю.Е.

"__" _____ 2019 г.

**Отчет по лабораторной работе № 3 по курсу
“Разработка интернет-приложений”
«Python. Объектно-ориентированные возможности»**

6
(количество листов)

ИСПОЛНИТЕЛЬ:

студент группы **ИУ5-54**

Меркулова Н. А.

(подпись)

"__" _____ 2019 г.

1. Задание

Необходимо разработать программу, реализующую работу с классами.

1. Программа должна быть разработана в виде консольного приложения на языке Python 3.
2. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab_python_oop.
3. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab_python_oop.
 - 3.1. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать здесь - <https://docs.python.org/3/library/abc.html>
 - 3.2. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь - <https://docs.python.org/3/library/functions.html#property>
 - 3.3. 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.
 - 3.4. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля math - <https://docs.python.org/3/library/math.html>
 - 3.5. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны.
 - 3.6. Для классов «Прямоугольник», «Квадрат», «Круг»:
 - 3.6.1. Определите метод "get", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
 - 3.6.2. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
4. 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:
 - 4.1. Прямоугольник синего цвета шириной 3 и высотой 2.
 - 4.2. Круг зеленого цвета радиусом 5.
 - 4.3. Квадрат красного цвета со стороной 5.

2. Исходный код

2.1. circle.py

```
from lab_python_oop.figure import Figure
```

```

from lab_python_oop.color import Color
import math

class Circle(Figure):

    FIGURE_TYPE = "Circle"

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

    def __init__(self, radius, color):
        self.radius = radius
        self.color = Color()
        self.color.colorproperty = color

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

    def __repr__(self):
        return "{} color: {}, radius: {}, area: {}".format(
            Circle.get_figure_type(),
            self.color.colorproperty,
            self.radius,
            self.area()
        )

```

2.2. color.py

```

class Color:
    def __init__(self):
        self.__color = None

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

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

```

2.3. figure.py

```

from abc import ABC, abstractmethod

```

```
class Figure():
    @abstractmethod
    def area(self):
        pass
```

2.4. rectangle.py

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

```
class Rectangle(Figure):

    FIGURE_TYPE = "Rectangle"

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

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

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

    def __repr__(self):
        return "{} color: {}, width: {}, height: {}, area: {}".format(
            Rectangle.get_figure_type(),
            self.color.colorproperty,
            self.width,
            self.height,
            self.area()
        )
```

2.5. square.py

```
from lab_python_oop.rectangle import Rectangle
```

```
class Square(Rectangle):

    FIGURE_TYPE = "Square"

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

    def __init__(self, side_size, color):
        self.side = side_size
```

```

        super().__init__(self.side, self.side, color)

    def __repr__(self):
        return "{} color: {}, side: {}, area: {}".format(
            Square.get_figure_type(),
            self.color.colorproperty,
            self.side,
            self.area()
        )

```

2.6. 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():
    rectangle = Rectangle(3, 2, "black")
    circle = Circle(5, "green")
    square = Square(5, "red")
    print(rectangle)
    print(circle)
    print(square)

if __name__ == "__main__":
    main()

```

3. Скриншот работы программы

```

Rectangle color: black, width: 3, height: 2, area: 6
Circle color: green, radius: 5, area: 78.53981633974483
Square color: red, side: 5, area: 25

Process finished with exit code 0

```

4. Диаграммы классов

