

**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
«Объектно-ориентированные возможности языка Python»

Выполнил:
студент группы ИУ5-52Б
Бабин Артём

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2021 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`.

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
import numpy as np

def main():
    print(Rectangle(2, 2, 'blue').repr(), '\n')
    print(Circle(2, 'green').repr(), '\n')
    print(Square(2, 'red').repr(), '\n')
    print('np.ones(10) result is', np.ones(10))

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

Figure.py:

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

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

FigureColor.py:

```
class FigureColor:
    def __init__(self, color=None):
        self._color = color

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

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

Rectangle.py:

```
from lab_python_oop.Figure import Figure
from lab_python_oop.FigureColor import FigureColor
```

```
class Rectangle(Figure):

    figure_name = 'Rectangle'

    def __init__(self, width, height, color):
        self.width = width
        self.height = height
        self.color_object = FigureColor(color)

    def square(self):
        return self.width * self.height
```

```

    @staticmethod
    def get_name():
        return Rectangle.figure_name

    def repr(self):
        return "{}\nWidth: {}\nHeight: {}\nColor: {}\nSquare:
{}".format(Rectangle.get_name(), self.width, self.height,
self.color_object.color, self.square())

```

Circle.py:

```

from lab_python_oop.Figure import Figure
from lab_python_oop.FigureColor import FigureColor
import math

```

```

class Circle(Figure):

    figure_name = 'Circle'

    def __init__(self, radius, color):
        self.radius = radius
        self.color_object = FigureColor(color)

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

    @staticmethod
    def get_name():
        return Circle.figure_name

    def repr(self):
        return "{}\nRadius: {}\nColor: {}\nSquare:
{}".format(Circle.get_name(), self.radius, self.color_object.color,
self.square())

```

Square.py:

```

from lab_python_oop.Rectangle import Rectangle

class Square(Rectangle):

```

```

figure_name = 'Square'

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

    @staticmethod
    def get_name():
        return Square.figure_name

    def repr(self):
        return "{}\nWidth: {}\nColor: {}\nSquare:
{}".format(Square.get_name(), self.width, self.color_object.color,
self.square())

```

Экранные формы с примерами выполнения программы:

Пример	Вывод программы
1	<pre> Rectangle Width: 2 Height: 2 Color: blue Square: 4 Circle Radius: 2 Color: green Square: 12.566370614359172 Square Width: 2 Color: red Square: 4 np.ones(10) result is [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.] </pre>