**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет**

**имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

 **(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Отчёт по лабораторной работе №2:

«Объектно-ориентированные возможности языка Python»

Выполнилa:

Студентка группы ИУ5-55Б

Е. И. Мащенко

***2020***

***Постановка задачи***

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.](https://docs.python.org/3/library/abc.html)
6. 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)
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.](https://docs.python.org/3/library/math.html)
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
10. Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
11. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
12. 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

def main():

    r = Rectangle( 11, 11, "синего")

    c = Circle("зеленого", 11)

    s = Square("красного", 11)

    print(r)

    print(c)

    print(s)

if \_\_name\_\_ == "\_\_main\_\_":

    # execute only if run as a script

    main()

figure.py

from abc import ABC, abstractmethod

#абстрактный класс

class Figure(ABC):

    # абстрактный метод, который будет необходимо переопределять для каждого подкласса

    @abstractmethod

    def square(self):

        pass

color.py

class FigureColor:

    def \_\_init\_\_(self):

        self.\_\_color = None

    @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.color import FigureColor

class Rectangle(Figure):

    figure\_type = "Прямоугольник"

    @classmethod

    def get\_figure\_type(cls):

        return cls.figure\_type

    def \_\_init\_\_(self, width, height, color):

        self.width = width

        self.height = height

        self.figurecolor = FigureColor()

        self.figurecolor.color = color

    def square(self):

        return self.width\*self.height

    def \_\_repr\_\_(self):

        return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(

            Rectangle.get\_figure\_type(),

            self.figurecolor.color,

            self.width,

            self.height,

            self.square()

        )

square.py

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.color import FigureColor

class Square(Rectangle):

    figure\_type = "Квадрат"

    @classmethod

    def get\_figure\_type(cls):

        return cls.figure\_type

    def \_\_init\_\_(self, color, length):

        self.length = length

        super().\_\_init\_\_( self.length, self.length, color)

    def \_\_repr\_\_(self):

        return '{} {} цвета со стороной {} площадью {}.'.format(

            Square.get\_figure\_type(),

            self.figurecolor.color,

            self.length,

            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, radius):

        self.radius = radius

        self.figurecolor = FigureColor()

        self.figurecolor.color = color

    def square(self):

        return math.pi\*(self.radius\*\*2)

    def \_\_repr\_\_(self):

        return '{} {} цвета радиусом {} площадью {}.'.format(

            Circle.get\_figure\_type(),

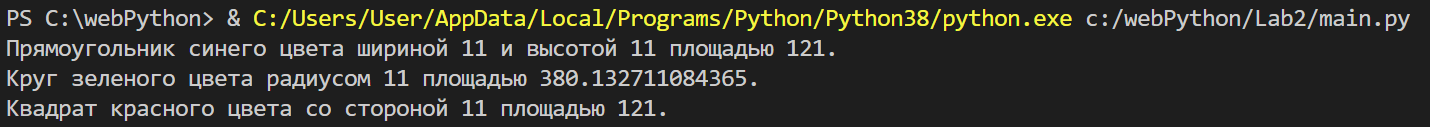
            self.figurecolor.color,

            self.radius,

            self.square()

        )

***Экранные формы с примерами выполнения программы***

******