**Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Московский государственный технический университет имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)**

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **«Информатики и систем управления»** |
| **КАФЕДРА** | **«Систем обработки информации и управления»** |

**Лабораторная работа 2**

Объектно-ориентированные возможности языка Python.

ПО ДИСЦИПЛИНЕ:«Разработка интернет приложений»

Студент ИУ5-52Б Чепкин Д. А.

(Группа) (И.О. Фамилия)

Москва 2020

**Задание:**

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»:
   * 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.

Main:

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("синего", 23, 23)

    c = Circle("зеленого", 23)

    s = Square("красного", 23)

    print(r)

    print(c)

    print(s)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

circle:

from lab\_python\_oop.figure import Figure

from lab\_python\_oop.color import FigureColor

import math

class Circle(Figure):

    Type = "Круг"

    def getType(self):

        return self.Type

    def \_\_init\_\_(self, color\_param, r\_param):

        self.r = r\_param

        self.fcolor = FigureColor()

        self.fcolor.colorprop = color\_param

    def square(self):

        return math.pi\*(self.r\*\*2)

    def \_\_repr\_\_(self):

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

            self.getType(),

            self.fcolor.colorprop,

            self.r,

            self.square()

            )

Color:

class FigureColor:

    def \_\_init\_\_(self):

        self.color = None

    def getColor(self):

        return self.color

    def setColor(self, value):

        self.color = value

Figure:

from abc import ABC, abstractmethod

class Figure(ABC):

    @abstractmethod

    def square(self):

        pass

rectangle:

from lab\_python\_oop.figure import Figure

from lab\_python\_oop.color import FigureColor

class Rectangle(Figure):

    Type = "Прямоугольник"

    def getType(self):

        return self.Type

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

        self.width = width\_param

        self.height = height\_param

        self.fcolor = FigureColor()

        self.fcolor.colorprop = color\_param

    def square(self):

        return self.width\*self.height

    def \_\_repr\_\_(self):

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

            self.getType(),

            self.fcolor.colorprop,

            self.width,

            self.height,

            self.square()

            )

Square:

from lab\_python\_oop.rectangle import Rectangle

class Square(Rectangle):

    Type = "Квадрат"

    @classmethod

    def getType(self):

        return self.Type

    def \_\_init\_\_(self, color\_param, side\_param):

        self.side = side\_param

        super().\_\_init\_\_(color\_param, self.side, self.side)

    def \_\_repr\_\_(self):

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

            self.getType(),

            self.fcolor.colorprop,

            self.side,

            self.square()

            )

Результат:

