**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 Р.С. Нищук

(Группа) (И.О. Фамилия)

Email: gidraaa@inbox.ru

# 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.](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.

# Текст программы

main.py:

from lab\_python\_oop.circle import Circle

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.square import Square

from numpy import euler\_gamma

if (\_\_name\_\_ == "\_\_main\_\_"):

    rectangle = Rectangle(3, 3, "синего")

    circle = Circle(3, "зелёного")

    square = Square(3, "красного")

    print (rectangle)

    print (square)

    print (circle)

    print ("Euler gamma constant y = {}".format(euler\_gamma))

circle.py:

from lab\_python\_oop.figure import Figure

from lab\_python\_oop.figure\_color import Color

from math import pi

class Circle(Figure):

    figureType = "Круг"

    @classmethod

    def getFigureType(cls):

        return cls.figureType

    def \_\_init\_\_(self, radius, colorName):

        self.\_radius = radius

        self.\_color = Color(colorName)

    def square(self):

        return pi \* self.\_radius \*\* 2

    def \_\_repr\_\_(self):

        return f"{Circle.getFigureType()} {self.\_color.colorValue} цвета радиусом {self.\_radius}."

figure\_color.py:

class Color():

    def \_\_init\_\_(self, color):

        self.\_color = color

    @property

    def colorValue(self):

        return self.\_color

    @colorValue.setter

    def colorValue(self, value):

        self.\_color = value

figure.py:

from abc import ABC, abstractmethod

class Figure(ABC):

    @abstractmethod

    def square(self):

        pass

rectangle.py:

from lab\_python\_oop.figure import Figure

from lab\_python\_oop.figure\_color import Color

class Rectangle(Figure):

    figureType = "Прямоугольник"

    @classmethod

    def getFigureType(cls):

        return cls.figureType

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

        self.\_width = width

        self.\_height = height

        self.\_color = Color(colorName)

    def square(self):

        return self.\_width \* self.\_height

    def \_\_repr\_\_(self):

        return f"{Rectangle.getFigureType()} {self.\_color.colorValue} цвета шириной {self.\_width} и высотой {self.\_height}."

square.py:

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.figure\_color import Color

class Square(Rectangle):

    figureType = "Квадрат"

    @classmethod

    def getFigureType(cls):

        return cls.figureType

    def \_\_init\_\_ (self, sideLength, colorName):

        self.\_color = Color(colorName)

        super().\_\_init\_\_(sideLength, sideLength, colorName)

    def \_\_repr\_\_(self):

        return f"{Square.getFigureType()} {self.\_color.colorValue} цвета со стороной {self.\_width}."

    pass

# Пример выполнения программы

