МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

Факультет “Информатика и системы управления”  
Кафедра “Системы обработки информации и управления”

A logo with a gold star and blue ribbon

Description automatically generated

Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по лабораторной работе №2  
“Объектно-ориентированные возможности языка Python.”

**Выполнил:**  
Студент группы ИУ5-34Б  
Верзаков Н.В.  
**Преподаватель:**  
Нардид А.Н.

Москва 2025

Задание лабораторной работы[​](https://iu5edu.ru/wiki/cpp2/docs/labs/lab6/#%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5)

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.
11. **Дополнительное задание.** Протестируйте корректность работы Вашей программы с помощью модульного теста.

Текст программы

geometric\_figure.py

from abc import ABC, **abstractmethod**

class GeometricFigure(ABC):

def **\_\_init\_\_**(self):

self.name = self.\_\_class\_\_.\_\_name\_\_

**@abstractmethod**

def **calculate\_area**(self):

pass

def **get\_name**(self):

return self.name

figure\_color.py

class FigureColor:

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

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.geometric\_figure import GeometricFigure

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

class Rectangle(GeometricFigure):

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

super().**\_\_init\_\_**()

self.width = width

self.height = height

self.color\_obj = FigureColor(color)

def **calculate\_area**(self):

return self.width \* self.height

def **\_\_repr\_\_**(self):

return "{} цвета {} шириной {} и высотой {}. Площадь: {:.2f}".**format**(

self.**get\_name**(),

self.color\_obj.color,

self.width,

self.height,

self.**calculate\_area**()

)

square.py

from lab\_python\_oop.rectangle import Rectangle

class Square(Rectangle):

def **\_\_init\_\_**(self, side\_length, color):

super().**\_\_init\_\_**(side\_length, side\_length, color)

def **\_\_repr\_\_**(self):

return "{} цвета {} со стороной {}. Площадь: {:.2f}".**format**(

self.**get\_name**(),

self.color\_obj.color,

self.width,

self.**calculate\_area**()

)

circle.py

import math

from lab\_python\_oop.geometric\_figure import GeometricFigure

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

class Circle(GeometricFigure):

def **\_\_init\_\_**(self, radius, color):

super().**\_\_init\_\_**()

self.radius = radius

self.color\_obj = FigureColor(color)

def **calculate\_area**(self):

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

def **\_\_repr\_\_**(self):

return "{} цвета {} радиусом {}. Площадь: {:.2f}".**format**(

self.**get\_name**(),

self.color\_obj.color,

self.radius,

self.**calculate\_area**()

)

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**():

**print**("Тестирование классов геометрических фигур\n")

rectangle = Rectangle(5, 5, "синий")

circle = Circle(5, "зеленый")

square = Square(5, "красный")

**print**("Созданные фигуры:")

**print**(f"1. {rectangle}")

**print**(f"2. {circle}")

**print**(f"3. {square}")

if \_\_name\_\_ == "\_\_main\_\_":

**main**()

Экранные формы с примерами выполнения программы

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.