**Объектно-ориентированные возможности языка Python.**

Студент: Абдуллаев Гасан  
Группа: ИУ5-36Б  
Дата: 15.11.24г.

Описание 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»:
   * 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. 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:

**main.py:**

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.circle import Circle

from lab\_python\_oop.square import Square

import matplotlib.pyplot as plt

def main():

N = 5

rectangle = Rectangle(N, N, "синий")

circle = Circle(N, "зеленый")

square = Square(N, "красный")

print(rectangle)

print(circle)

print(square)

plt.plot([1, 2, 3], [1, 4, 9])

plt.title("Пример графика")

plt.show()

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

main()

**circle.py:**

from math import pi

from lab\_python\_oop.figure import Figure

from lab\_python\_oop.color import Color

class Circle(Figure):

FIGURE\_TYPE = "Круг"

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

self.radius = radius

self.color = Color(color)

def area(self):

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

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

return "{} {} цвета радиусом {} имеет площадь {:.2f}".format(

self.FIGURE\_TYPE, self.color.color, self.radius, self.area()

)

**Square.py:**

from lab\_python\_oop.rectangle import Rectangle

class Square(Rectangle):

FIGURE\_TYPE = "Квадрат"

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

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

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

return "{} {} цвета со стороной {} имеет площадь {:.2f}".format(

self.FIGURE\_TYPE, self.color.color, self.width, self.area()

)

**Rectangle.py:**

from lab\_python\_oop.figure import Figure

from lab\_python\_oop.color import Color

class Rectangle(Figure):

FIGURE\_TYPE = "Прямоугольник"

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

self.width = width

self.height = height

self.color = Color(color)

def area(self):

return self.width \* self.height

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

return "{} {} цвета шириной {} и высотой {} имеет площадь {:.2f}".format(

self.FIGURE\_TYPE, self.color.color, self.width, self.height, self.area()

)

**Figure.py:**

from abc import ABC, abstractmethod

class Figure(ABC):

@abstractmethod

def area(self):

pass

@abstractmethod

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

pass

**color.py:**

class Color:

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

self.\_color = color

@property

def color(self):

return self.\_color

@color.setter

def color(self, value):

self.\_color = value

**test.py:**

import unittest

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.circle import Circle

from lab\_python\_oop.square import Square

class TestFigures(unittest.TestCase):

def test\_rectangle\_area(self):

rect = Rectangle(3, 4, "синий")

self.assertEqual(rect.area(), 12)

def test\_circle\_area(self):

circle = Circle(3, "зеленый")

self.assertAlmostEqual(circle.area(), 28.27, places=2)

def test\_square\_area(self):

square = Square(5, "красный")

self.assertEqual(square.area(), 25)

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

Снимки экрана:

