|  |  |
| --- | --- |
|  | **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 «Системы обработки информации и управления»**

Вопияшин Никита ИУ5-35Б

Парадигмы и конструкции языков программирования

**ОТЧЁТ ПО**

**Лабораторной работе №2**

Москва

2023

**Задание.**

Создать пакет "lab\_python\_oop" и внутри него создать следующие файлы и классы:

* figure.py: создать абстрактный класс "Геометрическая фигура" с абстрактным методом для вычисления площади.
* color.py: создать класс "Цвет фигуры" со свойством для описания цвета геометрической фигуры.
* rectangle.py: создать класс "Прямоугольник", который наследуется от "Геометрической фигуры". Включить конструктор, переопределить метод вычисления площади и определить метод repr для возврата информации о прямоугольнике.
* circle.py: создать класс "Круг" аналогично "Прямоугольнику", задав параметр "радиус" и использовать math.pi для вычисления площади.
* square.py: создать класс "Квадрат", который наследуется от "Прямоугольника". Включить конструктор по длине стороны.

В каждом из классов "Прямоугольник", "Круг" и "Квадрат" определить метод repr, который будет возвращать информацию о фигуре, ее цвете и площади в виде строки.

В главном файле "main.py" создать объекты этих классов, установить им цвет и вывести информацию о них с помощью метода repr.

Запустить консольное приложение, чтобы убедиться, что оно работает правильно.

Дополнительное задание. Протестировать корректность работы 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  
  
def main():  
 rec = Rectangle(2, 4, "красного")  
 cir = Circle(4, "жёлтого")  
 squ = Square(4, "зелёного")  
 print(rec)  
 print(cir)  
 print(squ)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

figure.py

from abc import ABC, abstractmethod  
class Figure(ABC):  
 @abstractmethod  
 def area(self):  
 pass

color.py

class Color:  
 def \_\_init\_\_(self, color=None):  
 self.\_color = color

rectangle.py

from lab\_python\_oop.figure import Figure  
from lab\_python\_oop.color import Color  
  
  
class Rectangle(Figure):  
 type = "Прямоугольник"  
  
 def \_\_init\_\_(self, width, length, color):  
 self.width = width  
 self.length = length  
 self.r\_color = Color(color)  
  
 def area(self):  
 return self.width \* self.length  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета с длиной {}, шириной {}, площадью {}.'.format(  
 self.type,  
 self.r\_color.\_color,  
 self.length,  
 self.width,  
 self.area()  
 )

circle.py

from lab\_python\_oop.figure import Figure  
from lab\_python\_oop.color import Color  
import math  
  
  
class Circle(Figure):  
 type = "Круг"  
  
 def \_\_init\_\_(self, radius, color):  
 self.radius = radius  
 self.c\_color = Color(color)  
  
 def area(self):  
 return math.pi \* (self.radius \* self.radius)  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета радиусом {}, площадью {}.'.format(  
 self.type,  
 self.c\_color.\_color,  
 self.radius,  
 self.area()  
 )

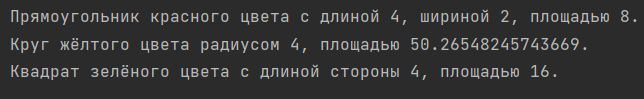
square.py

from lab\_python\_oop.rectangle import Rectangle  
  
  
class Square(Rectangle):  
 type = "Квадрат"  
  
 def \_\_init\_\_(self, side, color):  
 self.side = side  
 super().\_\_init\_\_(self.side, self.side, color)  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета с длиной стороны {}, площадью {}.'.format(  
 self.type,  
 self.r\_color.\_color,  
 self.side,  
 self.area()  
 )

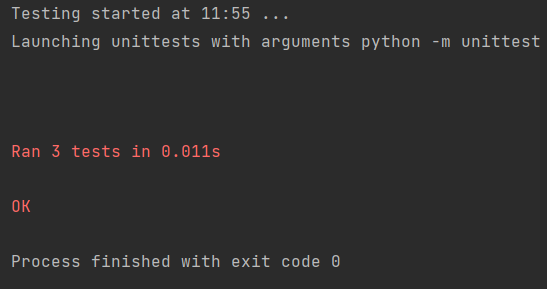
test.py

import unittest  
  
from lab\_python\_oop.square import Square  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.rectangle import Rectangle  
  
  
green = "зелёного"  
red = "красного"  
yellow = "жёлтого"  
  
  
class TestFigures(unittest.TestCase):  
 def setUp(self):  
 self.a = Rectangle(2, 4, red)  
 self.b = Circle(4, yellow)  
 self.c = Square(4, green)  
  
 def test\_area(self):  
 from math import pi  
 self.assertEqual(self.a.area(), 8)  
 self.assertEqual(self.b.area(), pi \* 4 \* 4)  
 self.assertEqual(self.c.area(), 16)  
  
 def test\_color(self):  
 self.assertEqual(self.a.r\_color.\_color, red)  
 self.assertEqual(self.b.c\_color.\_color, yellow)  
 self.assertEqual(self.c.r\_color.\_color, green)  
  
 def test\_repr(self):  
 self.assertEqual(str(self.a), 'Прямоугольник красного цвета с длиной 4, шириной 2, площадью 8.')  
 self.assertEqual(str(self.b), 'Круг жёлтого цвета радиусом 4, площадью 50.26548245743669.')  
 self.assertEqual(str(self.c), 'Квадрат зелёного цвета с длиной стороны 4, площадью 16.')  
  
  
def main():  
 print(Rectangle(2, 4, red))  
 print(Circle(4, yellow))  
 print(Square(4, green))  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Экранные формы:

****

Результат работы модульного тестирования:

****