Отчет №4

Circle.py

import math  
  
class Circle(GeometricFigure):  
 FIGURE\_TYPE = "Круг"  
  
 def \_\_init\_\_(self, radius, color):  
 self.radius = radius  
 self.color = FigureColor(color)  
  
 def area(self):  
 return math.pi \* self.radius \*\* 2  
  
 def \_\_repr\_\_(self):  
 return "Фигура: {}, Цвет: {}, Радиус: {}, Площадь: {:.2f}".format(  
 self.FIGURE\_TYPE, self.color.color, self.radius, self.area()  
 )

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

Geometric\_figure.py

from abc import ABC, abstractmethod  
  
class GeometricFigure(ABC):  
 @abstractmethod  
 def area(self):  
 pass  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.\_\_name\_\_

Rectangle.py

from .geometric\_figure import GeometricFigure  
from .figure\_color import FigureColor  
  
class Rectangle(GeometricFigure):  
 FIGURE\_TYPE = "Прямоугольник"  
  
 def \_\_init\_\_(self, width, height, color):  
 self.width = width  
 self.height = height  
 self.color = FigureColor(color)  
  
 def area(self):  
 return self.width \* self.height  
  
 def \_\_repr\_\_(self):  
 return "Фигура: {}, Цвет: {}, Ширина: {}, Высота: {}, Площадь: {:.2f}".format(  
 self.FIGURE\_TYPE, self.color.color, self.width, self.height, self.area()  
 )

Square.py

from .rectangle import Rectangle  
  
class Square(Rectangle):  
 FIGURE\_TYPE = "Квадрат"  
  
 def \_\_init\_\_(self, side, color):  
 super().\_\_init\_\_(side, side, color)  
  
 def \_\_repr\_\_(self):  
 return "Фигура: {}, Цвет: {}, Сторона: {}, Площадь: {:.2f}".format(  
 self.FIGURE\_TYPE, self.color.color, self.width, self.area()  
 )

Test\_figures.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(2, "зелёный")  
 self.assertAlmostEqual(circle.area(), 12.5664, places=4)  
  
 def test\_square\_area(self):  
 square = Square(5, "красный")  
 self.assertEqual(square.area(), 25)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 unittest.main()

Main.py

from lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.square import Square  
import requests  
  
  
def main():  
 N = 13  
  
 rect = Rectangle(N, N, "синий")  
 circle = Circle(N, "зелёный")  
 square = Square(N, "красный")  
  
 print(rect)  
 print(circle)  
 print(square)  
  
 response = requests.get("https://jsonplaceholder.typicode.com/todos/1")  
 print("Ответ от внешнего пакета 'requests':", response.json())  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

