Московский Государственный Технический Университет им. Н.Э. Баумана

Отчет по лабораторной работе № 2 по курсу Базовые компоненты интернет-технологий

"Объектно-ориентированные возможности языка Python"

Исполнитель:

Фруктин А.Е. РТ5-31Б

Проверил:

Гапанюк Юрий Евгеньевич

Москва, 2022

**Задание**

Разработать программу, реализующую работу с классами геометрических фигур. Программа должна быть разработана в виде консольного приложения на языке Python 3.

**Код**

**circle.py:**

from figure import Figure

from color import FigureColor

from PIL import Image, ImageDraw

import math

class Circle(Figure):

    FigureType = "Круг"

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

        self.r = r\_param

        self.fc = FigureColor()

        self.fc.colorproperty = color\_param

    def square(self):

        return math.pi \* (self.r \*\* 2)

    def \_\_repr\_\_(self):

        im1 = Image.new('RGB', (610, 240), (219, 193, 27))

        draw = ImageDraw.Draw(im1)

        draw.ellipse((100, 100, 200, 200), fill='green', outline=(0, 0, 0))

        im1.save('cicle.jpg', quality=95)

        return '{} {} цвета радиусом {} площадью {}.'.format(

            Circle.get\_figure\_type(),

            self.fc.colorproperty,

            self.r,

            self.square()

        )

**color.py:**

class FigureColor:

    def \_\_init\_\_(self):

        self.color = None

    @property

    def colorproperty(self):

        return self.color #Get

    @colorproperty.setter

    def colorproperty(self, value):

        self.color = value

**figure.py:**

from abc import ABC, abstractmethod

class Figure(ABC):

    @classmethod

    def get\_figure\_type(cls):

        return cls.FigureType

    @abstractmethod

    def square(self):

        pass

**main.py:**

from rectangle import Rectangle

from circle import Circle

from square import Square

def main():

    r = Rectangle("синего", 3, 2)

    c = Circle("зеленого", 5)

    s = Square("красного", 5)

    print(r)

    print(c)

    print(s)

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

    main()

**rectangle.py:**

from figure import Figure

from color import FigureColor

from PIL import Image, ImageDraw

class Rectangle(Figure):

    FigureType = "Прямоугольник"

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

        self.width = width\_param

        self.height = height\_param

        self.fc = FigureColor()

        self.fc.colorproperty = color\_param

    def square(self):

        return self.width\*self.height #Площадь

    def \_\_repr\_\_(self):

        im = Image.new('RGB', (610, 240), (219, 193, 27))

        draw = ImageDraw.Draw(im)

        draw.rectangle((200, 100, 300, 200), fill='blue', outline=(255, 255, 255))

        im.save('rectangle.jpg', quality=95)

        print('test')

        return ('{} {} цвета шириной {} и высотой {} площадью {}.'.format(

            Rectangle.get\_figure\_type(),

            self.fc.colorproperty,

            self.width,

            self.height,

            self.square()

        ))

**square.py:**

from rectangle import Rectangle

from PIL import Image, ImageDraw

class Square(Rectangle):

    FigureType = "Квадрат"

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

        self.side = side\_param

        super().\_\_init\_\_(color\_param, self.side, self.side)

    def \_\_repr\_\_(self):

        return '{} {} цвета со стороной {} площадью {}.'.format(

            Square.get\_figure\_type(),

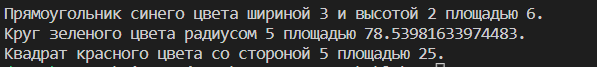
            self.fc.colorproperty,

            self.side,

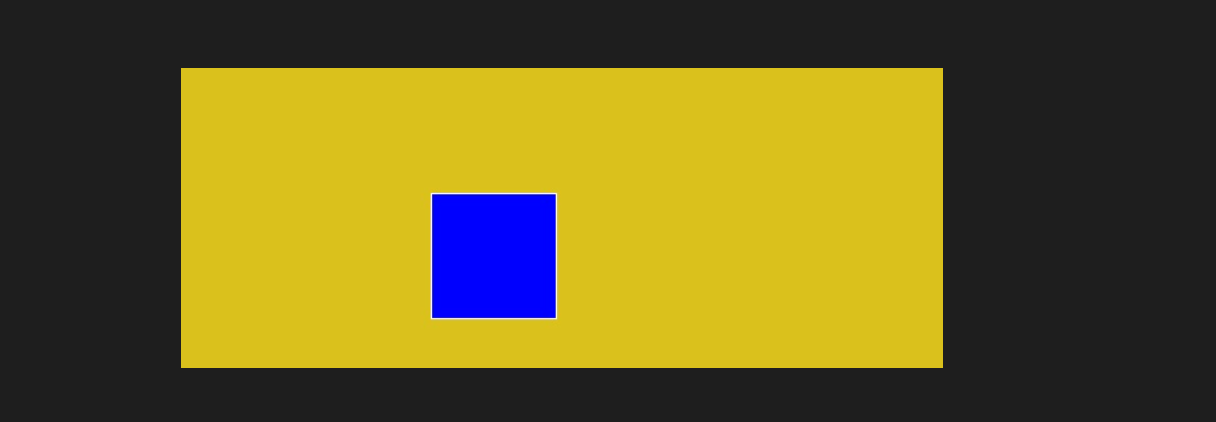
            self.square()

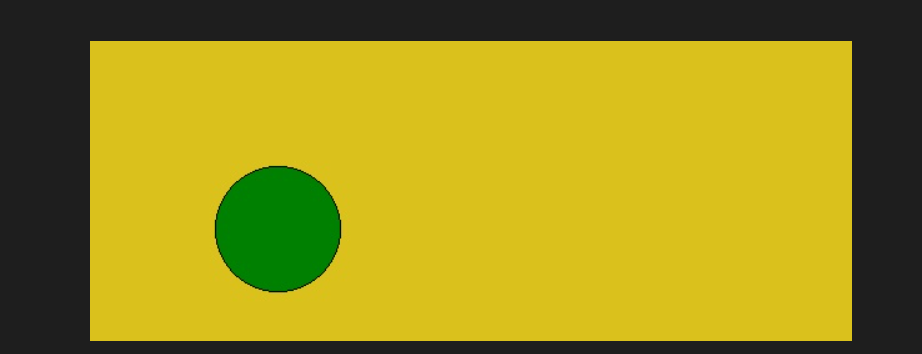
        )

**Результаты работы программы**

****

*Подключенный пакет – pillow:*

****

****