|  |  |
| --- | --- |
|  | **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»**

Абалуев Антон ИУ5-35Б

Парадигмы и конструкции языков программирования

**ОТЧЁТ ПО**

**Лабораторной работе №2 на С#**

Москва

2023

**Задание.**

Создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.

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

* figure.cs: Создать абстрактный класс "Геометрическая фигура" с абстрактным методом для вычисления площади.
* color.cs: Создать класс "Цвет фигуры" со свойством для описания цвета геометрической фигуры.
* rectangle.cs: Создать класс "Прямоугольник", который наследуется от "Геометрической фигуры". Включить конструктор, переопределить метод вычисления площади и определить метод repr для возврата информации о прямоугольнике.
* circle.cs: 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 "Program.cs" 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.

**Текст программы.**

Program.cs

// See https://aka.ms/new-console-template for more information

using System;

class Program

{

static void Main()

{

Rectangle r = new Rectangle(1, 2, "красного");

Circle c = new Circle(1, "зеленого");

Square s = new Square(2, "синего");

Console.WriteLine(r);

Console.WriteLine(c);

Console.WriteLine(s);

}

}

figure.cs

using System;

public abstract class Figure

{

public abstract double Area();

}

color.py

public class Color

{

private string \_color;

public Color(string color)

{

\_color = color;

}

public string GetColor()

{

return \_color;

}

}

public class Rectangle : Figure

{

public static string FigureType = "Прямоугольник";

private double \_a;

private double \_b;

private Color \_rColor;

public Rectangle(double a, double b, string color)

{

\_a = a;

\_b = b;

\_rColor = new Color(color);

}

public override double Area()

{

return \_a \* \_b;

}

protected Color RColor

{

get { return \_rColor; }

}

protected double A

{

get { return \_a; }

}

protected double B

{

get { return \_b; }

}

public override string ToString()

{

return $"{FigureType} {RColor.GetColor()} цвета с длиной стороны {A}, шириной стороны {B}, площадью {Area()}.";

}

}

circle.py

using System;

public class Circle : Figure

{

public static string FigureType = "Круг";

private double radius;

private Color Circle\_Color;

public Circle(double r, string color)

{

radius = r;

Circle\_Color = new Color(color);

}

public override double Area()

{

return Math.PI \* (radius \* radius);

}

public override string ToString()

{

return $"{FigureType} {Circle\_Color.GetColor()} цвета радиусом {radius} площадью {Area()}.";

}

}

square.py

public class Square : Rectangle

{

public new static string FigureType = "Квадрат";

public Square(double a, string color) : base(a, a, color)

{}

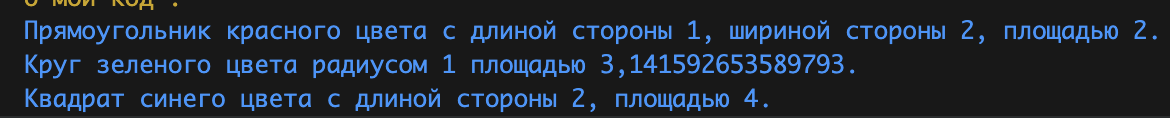
public override string ToString()

{

return $"{FigureType} {RColor.GetColor()} цвета с длиной стороны {A}, площадью {Area()}.";

}

}

****