

Дьяконова Алісія

ПЗПІ-23-7

Лабораторна 2

Номер 6

Мета роботи: Навчитися використовувати механізм спадкування класів. Навчитися використовувати колекції.

На оцінку "задовільно"

```
Circle: Center=(0, 0), Radius=5
Area: 78,53981633974483
Perimeter: 31,41592653589793
After moving:
Circle: Center=(10, 10), Radius=5
After scaling:
Circle: Center=(10, 10), Radius=10
Area: 314,1592653589793
Perimeter: 62,83185307179586
Rectangle: TopLeft=(0, 0), Width=10, Height=5
Area: 50
Perimeter: 30
After moving:
Rectangle: TopLeft=(20, 20), Width=10, Height=5
After scaling:
Rectangle: TopLeft=(20, 20), Width=5, Height=2,5
Area: 12,5
Perimeter: 15
```

Class Figure.cs

using System;

abstract class Figure

```
{
    public double X { get; set; }
    public double Y { get; set; }

    public Figure(double x, double y)
    {
        X = x;
        Y = y;
    }
}
```

```
}
```

```
public abstract void Move(double deltaX, double deltaY);
```

```
public abstract void Scale(double factor);
```

```
public abstract double GetArea();
```

```
public abstract double GetPerimeter();
```

```
public abstract string GetInfo();
```

```
}
```

Class Circle.cs

```
using System;
```

```
class Circle : Figure
```

```
{
```

```
    public double Radius { get; set; }
```

```
    public Circle(double x, double y, double radius) : base(x, y)
```

```
    {
```

```
        Radius = radius;
```

```
    }
```

```
    public override void Move(double deltaX, double deltaY)
```

```
    {
```

```
        X += deltaX;
```

```
        Y += deltaY;
```

```
    }
```

```
    public override void Scale(double factor)
```

```

{
    Radius *= factor;
}

public override double GetArea()
{
    return Math.PI * Radius * Radius;
}

public override double GetPerimeter()
{
    return 2 * Math.PI * Radius;
}

public override string GetInfo()
{
    return $"Circle: Center=({X}, {Y}), Radius={Radius}";
}
}

```

Class Rectangle.cs

using System;

class Rectangle : Figure

```

{

    public double Width { get; set; }
    public double Height { get; set; }

    public Rectangle(double x, double y, double width, double height) : base(x, y)
    {
        Width = width;
    }
}

```

```
        Height = height;
    }

    public override void Move(double deltaX, double deltaY)
    {
        X += deltaX;
        Y += deltaY;
    }

    public override void Scale(double factor)
    {
        Width *= factor;
        Height *= factor;
    }

    public override double GetArea()
    {
        return Width * Height;
    }

    public override double GetPerimeter()
    {
        return 2 * (Width + Height);
    }

    public override string GetInfo()
    {
        return $"Rectangle: TopLeft=({X}, {Y}), Width={Width}, Height={Height}";
    }
}
```

Program.cs

using System;

class Program

{

static void Main(string[] args)

{

Circle circle = new Circle(0, 0, 5);

Console.WriteLine(circle.GetInfo());

Console.WriteLine(\$"Area: {circle.GetArea()}");

Console.WriteLine(\$"Perimeter: {circle.GetPerimeter()}");

circle.Move(10, 10);

Console.WriteLine("After moving:");

Console.WriteLine(circle.GetInfo());

circle.Scale(2);

Console.WriteLine("After scaling:");

Console.WriteLine(circle.GetInfo());

Console.WriteLine(\$"Area: {circle.GetArea()}");

Console.WriteLine(\$"Perimeter: {circle.GetPerimeter()}");

Rectangle rectangle = new Rectangle(0, 0, 10, 5);

Console.WriteLine(rectangle.GetInfo());

Console.WriteLine(\$"Area: {rectangle.GetArea()}");

Console.WriteLine(\$"Perimeter: {rectangle.GetPerimeter()}");

```
rectangle.Move(20, 20);  
  
Console.WriteLine("After moving:");  
Console.WriteLine(rectangle.GetInfo());  
  
rectangle.Scale(0.5);  
Console.WriteLine("After scaling:");  
Console.WriteLine(rectangle.GetInfo());  
Console.WriteLine($"Area: {rectangle.GetArea()}");  
Console.WriteLine($"Perimeter: {rectangle.GetPerimeter()}");  
}  
}
```

Висновок: В результаті виконання роботи навчилася використовувати механізм спадкування класів, навчилася використовувати колекції.