

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»



ЗВІТ
про виконання лабораторної роботи № 4

Виконав:
студент гр. 122-19-3
Файнштейн Д. В.

Перевірили:
доц. Приходченко С.Д.
ас. Шевцова О.С.

Дніпро
2020

Лабораторна робота №4-ООР

Вивчення поліморфізму в класах C#

Варіант 24 (У роботі – 4)

Цілі роботи

- навчитися створювати віртуальні методи;
- навчитися перевантажувати арифметичні операції і операції порівняння

Умова задачі

Порядок виконання роботи

1. Використовуючи навички, отримані в результаті виконання лабораторної роботи #3-ООР, виконати моделювання і кодування класів, що представляють собою найпростішу класову ієрархію.

Класи повинні містити поля даних для опису відповідних графічних об'єктів, конструктори, деструктори, відповідні set- і get-методи, віртуальний метод Show, що виводить на екран інформацію про графічний об'єкт і його предків (якщо такі є), і віртуальний метод GetClassName, який повертає ім'я класу об'єкта.

2. Розширити можливості класів, що входять в класову ієрархію, за рахунок перевантаження операцій відповідно до варіанта завдання.

Завдання на лабораторну роботу

Проект створювати як проект Windows Forms. Модифікувати лабораторну №2 таким чином, щоб на формі відображалися об'єкти обох класів (в найпростішому випадку - на pictureBox).

Створити класову модель, яка включає в себе абстрактний клас SGraphicsObject, і його два спадкоємця. Всі класи, за винятком абстрактного, повинні містити конструктори, деструктори, відповідні set- і get-методи, а також метод Show, який може мати різну сигнатуру для різних класів і повинен виводити на екран в текстовому вигляді інформацію про об'єкт і його предків. Абстрактний клас повинен містити декларацію полів і методів.

Обов'язково перевантажувати зазначену в варіанті операцію всередині класу.

На формі Windows

Примітка: абстрактний клас не потребує реалізації методів.

| Номер варианта | <class1> | <class2> | Операция |
|-------------------|-------------|---------------|--|
| 4 | Треугольник | Прямоугольник | Операция " <u>>=</u> ", которая сравнивает площади графических объектов |

Лістинг програми

CGraphicsObject.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;
using System.Windows.Forms;

namespace Lab4_Fainshtein
{
    public abstract class CGraphicsObject
    {
        protected double radius1; // Первый Радиус
        protected double radius2; // Второй Радиус
        protected double angleDeg; // Угол
        protected double width; // Ширина
        protected double length; // Длина
        protected List<Point> points; // Координаты
        protected double arcLength; // Длина дуги

        public abstract double Area(); // Площадь
        public virtual void Show(Label label) { }
        public virtual void Draw(PaintEventArgs e) { }
        public virtual string GetClassName()
        {
            return "CGraphicsObject";
        }

        public static bool operator >=(CGraphicsObject tr1, CGraphicsObject r1)
        {
            return tr1.Area() >= r1.Area();
        }
        public static bool operator <=(CGraphicsObject tr1, CGraphicsObject r1)
        {
            return tr1.Area() <= r1.Area();
        }
    }
}
```

Triangle.cs

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Math;

namespace Lab4_Fainshtein
{
    public class Triangle : CGraphicsObject
    {
        public Triangle(int x1, int y1, int x2, int y2, int x3, int y3)
```

```

    {
        points = new List<Point>();
        points.Add(new Point(x1, y1));
        points.Add(new Point(x2, y2));
        points.Add(new Point(x3, y3));
    }
    ~Triangle()
    {

    }

    public List<Point> Points
    {
        get { return points; }
        set { points = value; }
    }

    public override void Draw(PaintEventArgs e)
    {
        Point[] PointsArr = new Point[3];
        for (int i = 0; i < 3; i++)
            PointsArr[i] = points[i];
        e.Graphics.DrawPolygon(new Pen(Color.Black), PointsArr);
    }

    public override void Show(Label label)
    {
        string text = string.Format("Треугольник\nКоординаты: \n{0}\n{1}\n{2}",
points[0], points[1], points[2]);
        label.Text = text;
    }
    public override string GetClassName()
    {
        return "Triangle";
    }
    public override double Area()
    {
        double side1 = Sqrt(Pow(points[1].X - points[0].X, 2) + Pow(points[1].Y -
points[0].Y, 2));
        double side2 = Sqrt(Pow(points[1].X - points[2].X, 2) + Pow(points[1].Y -
points[2].Y, 2));
        double side3 = Sqrt(Pow(points[2].X - points[0].X, 2) + Pow(points[2].Y -
points[0].Y, 2));
        double p = (side1 + side2 + side3) / 2;
        double S = Sqrt(p * (p - side1) * (p - side2) * (p - side3));
        return S;
    }
}
}

```

Rectangle.cs

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Lab4_Fainshtein
{
    public class Rectangle : CGraphicsObject
    {
        public Rectangle(int x, int y, float width, float length)
        {

```

```

        points = new List<Point>();
        points.Add(new Point(x, y));
        this.width = width;
        this.length = length;
    }
    ~Rectangle()
    {

    }

    public List<Point> Points
    {
        get { return points; }
        set { points = value; }
    }
    public double Width
    {
        get { return width; }
        set { width = value; }
    }
    public double Length
    {
        get { return length; }
        set { length = value; }
    }

    public override void Draw(PaintEventArgs e)
    {
        e.Graphics.DrawRectangle(new Pen(Color.Black), points[0].X, points[0].Y,
(float)width, (float)length);
    }

    public override void Show(Label label)
    {
        string text = String.Format("Прямоугольник\nКоординаты: {0}\nШирина: {1},
Длина: {2}", points[0], width, length);
        label.Text = text;
    }
    public override string GetClassName()
    {
        return "Rectangle";
    }
    public override double Area()
    {
        return width * length;
    }
}
}

```

Form1.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Lab4_Fainshtein
{
    public partial class Form1 : Form
    {
        public Form1()
    }
}

```

```

{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    button1.Text = "Обновить";
    label3.Text = "Имена классов фигур";
}

private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
    // Треугольник
    Triangle triangle = new Triangle(20, 20, 225, 66, 80, 185);
    triangle.Draw(e);
    triangle.Show(label1);
    listBox1.Items.Add(triangle.GetClassName());

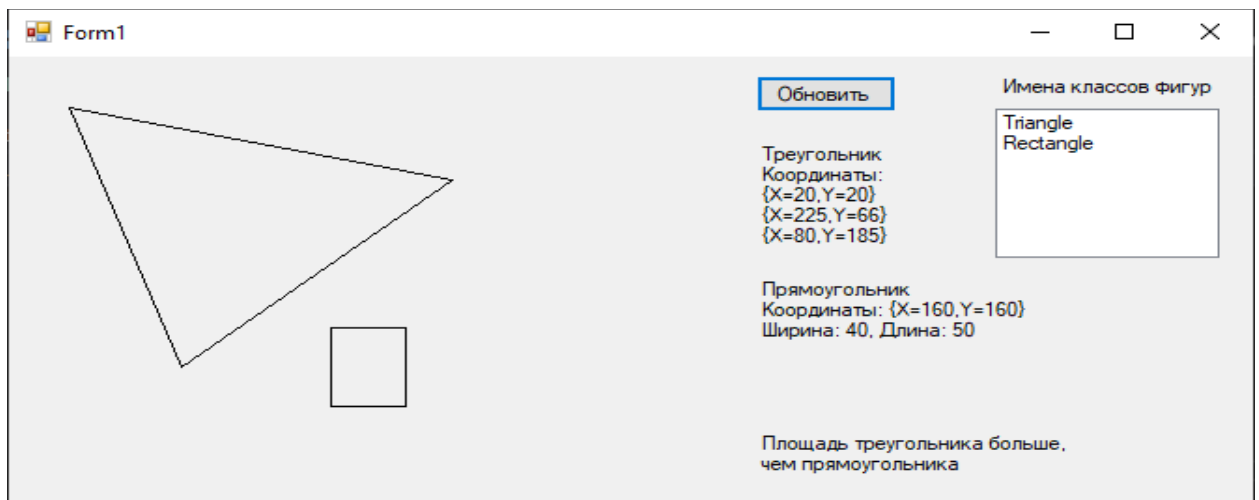
    // Прямоугольник
    Rectangle rectangle = new Rectangle(160, 160, 40, 50);
    rectangle.Draw(e);
    rectangle.Show(label2);
    listBox1.Items.Add(rectangle.GetClassName());

    // Сравнение площадей через оператор
    bool comparison = triangle >= rectangle;
    if (comparison)
    {
        label4.Text = "Площадь треугольника больше,\nчем прямоугольника";
    } else if (!comparison)
    {
        label4.Text = "Площадь прямоугольника больше,\nчем треугольника";
    }
}

private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    pictureBox1.Refresh();
}
}

```

Результат



Висновок: навчився створювати віртуальні методи, навчився перевантажувати арифметичні операції і операції порівняння