

**Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

**Лабораторна робота №2**

з дисципліни  
«ООП»

Виконала:

студентка групи ІМ-21  
Рабійчук Дар'я  
номер у списку групи: 19

Перевірів:

Порєв В. М.

Київ 2023

**Мета:** отримати вміння та навички використовувати інкапсуляцію, абстракцію типів, успадкування та поліморфізм на основі класів C++, запрограмувавши простий графічний редактор в об'єктно-орієнтованому стилі.

### **Завдання:**

1. Створити у середовищі MS Visual Studio C++ проект типу Windows Desktop Application з ім'ям Lab2.
2. Скомпілювати проект і отримати виконуваний файл програми.
3. Перевірити роботу програми. Налогодити програму.
4. Проаналізувати та прокоментувати результати та вихідний текст програми.
5. Оформити звіт.

### **Варіант 19:**

- статичний масив ( $19 \bmod 3 = 1$ )
- "гумовий" слід ( $19 \bmod 4 = 3$ ) – пунктирна лінія чорного кольору
- прямокутник:
- ввід від центру до одного з кутів ( $19 \bmod 2 = 1$ )
- чорний контур прямокутника без заповнення ( $19 \bmod 5 = 4$ )
- еліпс:
- по двом протилежним кутам охопл. прямокутника ( $19 \bmod 2 = 1$ )
- чорний контур прямокутника без заповнення ( $19 \bmod 5 = 4$ )
- колір заповнення: жовтий ( $19 \bmod 6 = 1$ )
- позначка поточного типу об'єкту: в заголовку вікна ( $19 \bmod 2 = 1$ )

### **Текст програми:**

#### **Form1.cs**

```
using System.Drawing;
using System.Drawing.Drawing2D;

namespace lab2
{
    public partial class Form1 : Form
    {
        ShapeObjectsEditor shapeObjectsEditor = new ShapeObjectsEditor();
        Graphics gfx;
        Bitmap bmp;
        public Form1()
        {
            InitializeComponent();
        }

        private void rectangleItem_Click(object sender, EventArgs e)
        {
            Text = "Прямокутник";
            shapeObjectsEditor.RectangleEditor();
        }

        private void lineItem_Click(object sender, EventArgs e)
```

```

{
    Text = "Лінія";
    shapeObjectsEditor.LineEditor();
}

private void ellipseItem_Click(object sender, EventArgs e)
{
    Text = "Еліпс";
    shapeObjectsEditor.EllipseEditor();
}

private void pictureBox_MouseUp(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        shapeObjectsEditor.InitPen(Color.Black, Color.Yellow, DashStyle.Solid,
(float)brushSizePicker.Value);
        bmp = new Bitmap(pictureBox.BackgroundImage);
        gfx = Graphics.FromImage(bmp);
        gfx.SmoothingMode = SmoothingMode.AntiAlias;
        shapeObjectsEditor.OnMouseUp(e, gfx);
        pictureBox.Image = bmp;
        shapeObjectsEditor.DisposePen();
    }
}

private void pictureBox_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        pictureBox.BackgroundImage = pictureBox.Image != null ? pictureBox.Image :
new Bitmap(Width, Height);
        shapeObjectsEditor.InitPen(Color.Black, Color.Yellow, DashStyle.Dash,
(float)brushSizePicker.Value);
        shapeObjectsEditor.OnMouseDown(e);
    }
}

private void pictureBox_MouseMove(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        bmp = new Bitmap(Width, Height);
        gfx = Graphics.FromImage(bmp);
        pictureBox.Image = bmp;
        shapeObjectsEditor.OnMouseMove(e, gfx);
    }
}

private void dotItem_Click(object sender, EventArgs e)
{
    Text = "Крапка";
    shapeObjectsEditor.DotEditor();
}

private void clearItem_Click(object sender, EventArgs e)
{
    Text = "Очищено";
    pictureBox.Image = null;
    pictureBox.BackgroundImage = null;
}

private void Form1_Load(object sender, EventArgs e)
{

```

```
    }  
  }  
}
```

## ShapeObjectsEditor.cs

```
using System.Drawing;  
using System.Drawing.Drawing2D;  
using System.Windows.Forms;  
using lab2.ShapeEditors;  
  
namespace lab2  
{  
    class ShapeObjectsEditor  
    {  
        ShapeEditor shapeEditor;  
  
        public void DotEditor()  
        {  
            this.shapeEditor = new DotEditor();  
        }  
  
        public void LineEditor()  
        {  
            this.shapeEditor = new LineEditor();  
        }  
  
        public void RectangleEditor()  
        {  
            this.shapeEditor = new RectangleEditor();  
        }  
  
        public void EllipseEditor()  
        {  
            this.shapeEditor = new EllipseEditor();  
        }  
  
        public void OnMouseDown(MouseEventArgs e)  
        {  
            if (this.shapeEditor != null)  
            {  
                this.shapeEditor.OnMouseDown(e);  
            }  
        }  
  
        public void OnMouseUp(MouseEventArgs e, Graphics g)  
        {  
            if (this.shapeEditor != null)  
            {  
                this.shapeEditor.OnMouseUp(e, g);  
            }  
        }  
  
        public void OnMouseMove(MouseEventArgs e, Graphics g)  
        {  
            if (this.shapeEditor != null)  
            {  
                this.shapeEditor.OnMouseMove(e, g);  
            }  
        }  
  
        public void InitPen(Color penColor, Color brushColor, DashStyle style, float  
size)  
        {  
            if (this.shapeEditor != null)  
            {  
                this.shapeEditor.InitPen(penColor, brushColor, style, size);  
            }  
        }  
    }  
}
```

```

    }
}

public void DisposePen()
{
    if (this.shapeEditor != null)
    {
        this.shapeEditor.DisposePen();
    }
}
}
}

```

## ShapeEditor.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;
using System.Drawing.Drawing2D;

namespace lab2
{
    abstract class ShapeEditor
    {
        protected int x1, y1, x2, y2;
        protected Pen pen;
        protected SolidBrush brush;

        public void OnMouseDown(MouseEventArgs e)
        {
            this.x1 = e.X;
            this.y1 = e.Y;
        }

        public abstract void OnMouseUp(MouseEventArgs e, Graphics g);

        public virtual void OnMouseMove(MouseEventArgs e, Graphics g)
        {
            this.OnMouseUp(e, g);
        }

        public void InitPen(Color penColor, Color brushColor, DashStyle style, float
size)
        {
            this.pen = new Pen(penColor);
            this.pen.DashStyle = style;
            this.pen.Width = size;
            this.brush = new SolidBrush(brushColor);
        }

        public void DisposePen()
        {
            this.pen.Dispose();
            this.brush.Dispose();
        }
    }
}

```

## DotEditor.cs

```

using System.Drawing;
using System.Windows.Forms;
using lab2.Shapes;

```

```

namespace lab2.ShapeEditors
{
    class DotEditor : ShapeEditor
    {
        public override void OnMouseUp(MouseEventArgs e, Graphics g)
        {
            this.x2 = e.X;
            this.y2 = e.Y;

            Pen newPen = new Pen(Color.Black, 2f);

            Dot DotShape = new Dot();
            DotShape.Set(this.x1, this.y1, this.x2, this.y2);
            DotShape.Show(g, newPen);

            newPen.Dispose();
        }
    }
}

```

## EllipseEditor.cs

```

using System.Drawing;
using System.Windows.Forms;
using lab2.Shapes;

namespace lab2.ShapeEditors
{
    class EllipseEditor : ShapeEditor
    {
        public override void OnMouseUp(MouseEventArgs e, Graphics g)
        {
            this.x2 = e.X;
            this.y2 = e.Y;

            Ellipse ellipseShape = new Ellipse();
            ellipseShape.Set(this.x1, this.y1, this.x2, this.y2);
            ellipseShape.Fill(g, brush);
            ellipseShape.Show(g, pen);
        }

        public override void OnMouseMove(MouseEventArgs e, Graphics g)
        {
            this.x2 = e.X;
            this.y2 = e.Y;

            Ellipse ellipseShape = new Ellipse();
            ellipseShape.Set(this.x1, this.y1, this.x2, this.y2);
            ellipseShape.Show(g, pen);
        }
    }
}

```

## LineEditor.cs

```

using System.Drawing;
using System.Windows.Forms;
using lab2.Shapes;

namespace lab2.ShapeEditors
{
    class LineEditor : ShapeEditor
    {
        public override void OnMouseUp(MouseEventArgs e, Graphics g)
        {

```

```

        this.x2 = e.X;
        this.y2 = e.Y;

        Line LineShape = new Line();
        LineShape.Set(this.x1, this.y1, this.x2, this.y2);
        LineShape.Show(g, pen);
    }
}

```

## RectangleEditor.cs

```

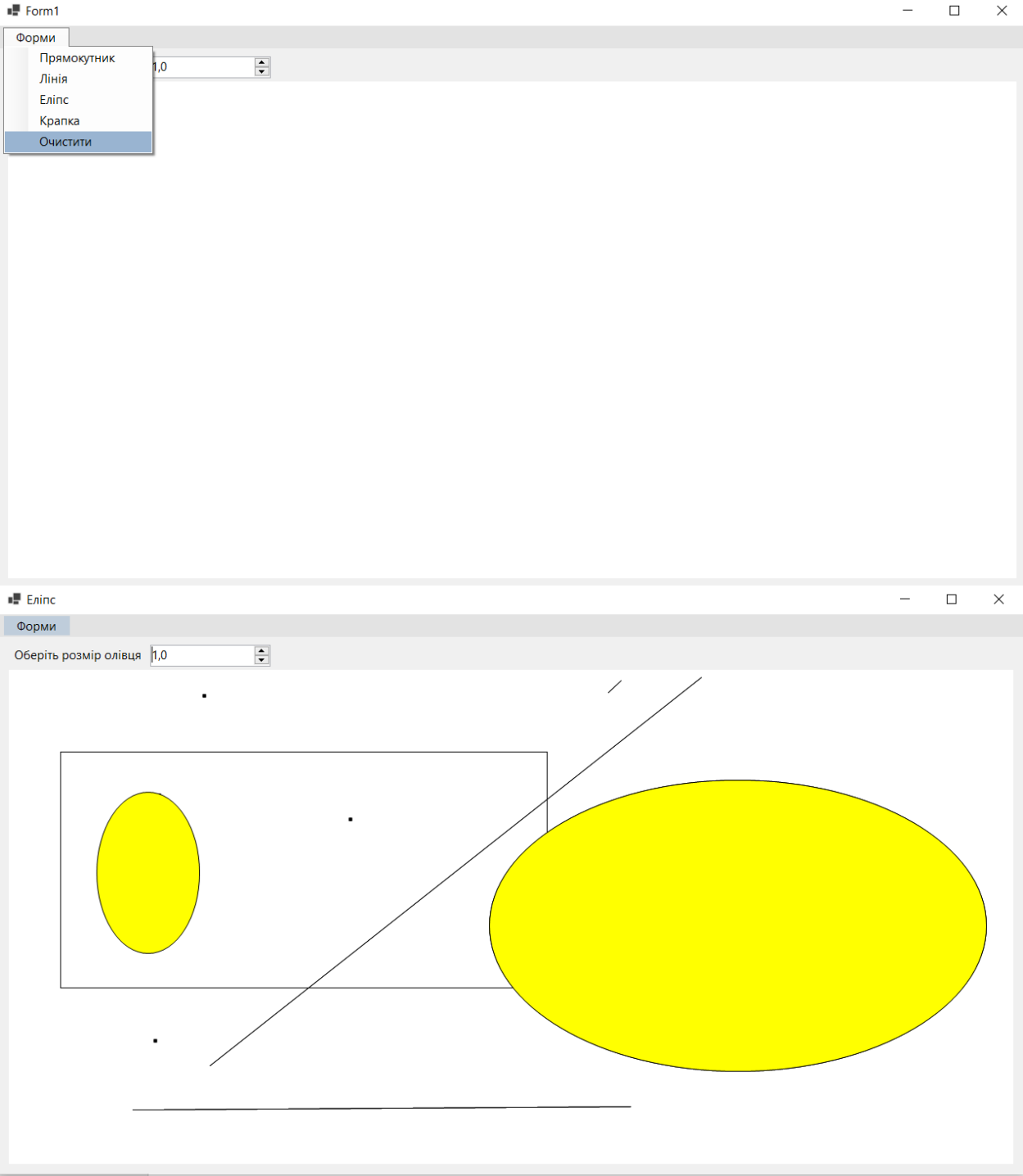
using System.Drawing;
using System.Windows.Forms;
using lab2.Shapes;

namespace lab2.ShapeEditors
{
    class RectangleEditor : ShapeEditor
    {
        public override void OnMouseUp(MouseEventArgs e, Graphics g)
        {
            this.x2 = e.X;
            this.y2 = e.Y;

            RectangleShape rectangleShape = new RectangleShape();
            rectangleShape.Set(this.x1, this.y1, this.x2, this.y2);
            rectangleShape.Show(g, pen);
        }
    }
}

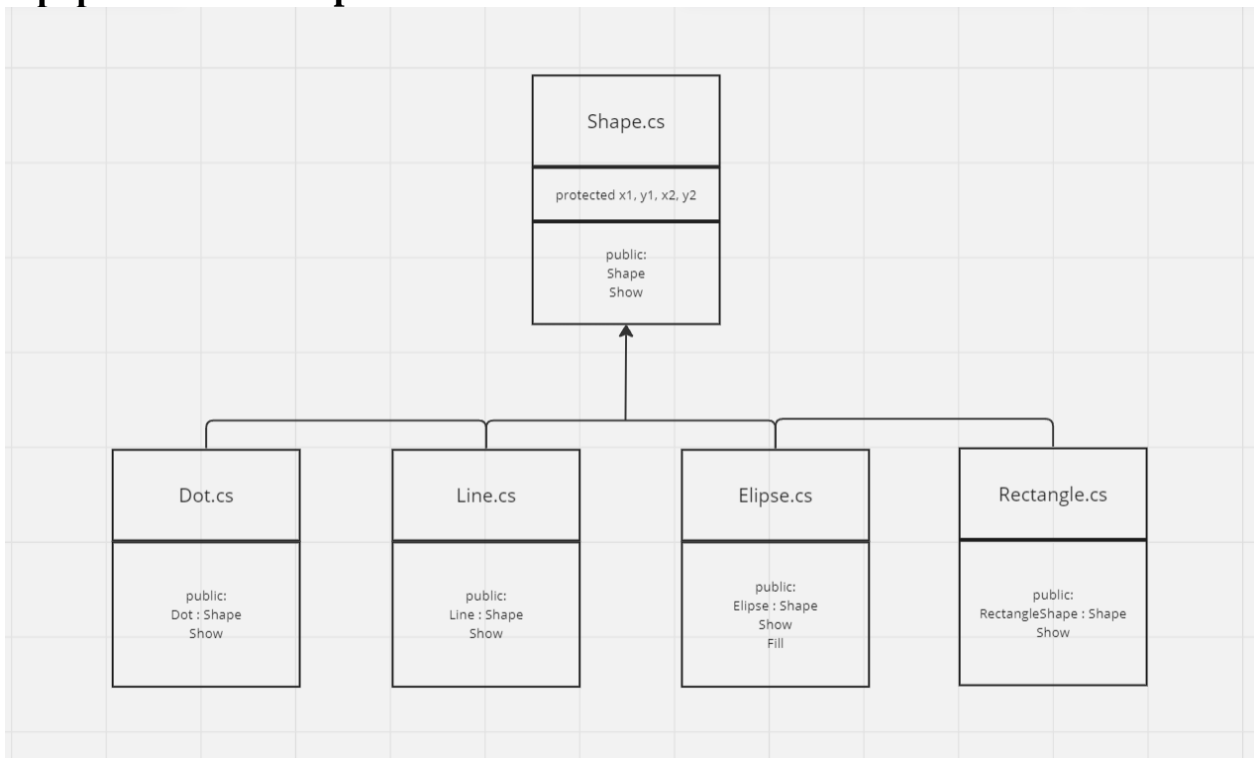
```

Результат роботи програми:

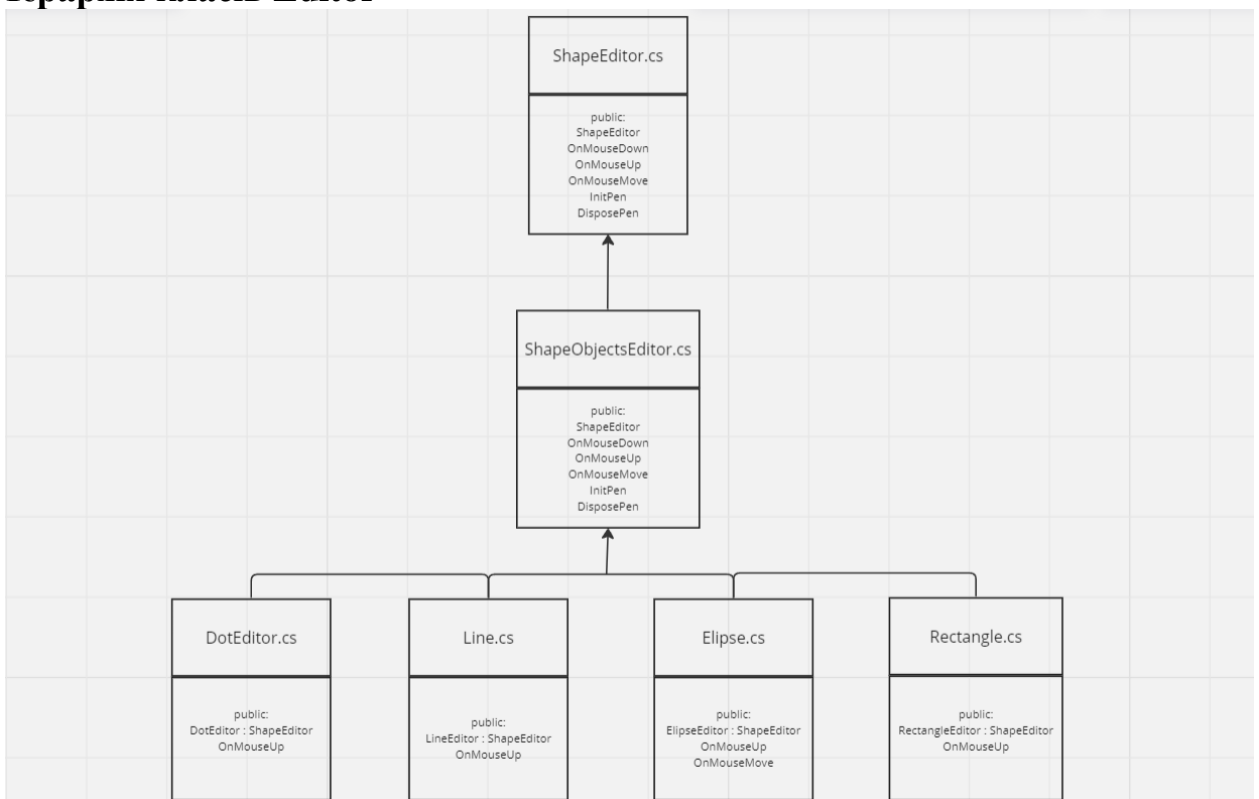




## Схема діаграм класів: Ієрархія класів Shape



## Ієрархія класів Editor



**Висновок:** Ми отримали вміння та навички використовувати інкапсуляцію, абстракцію типів, успадкування та поліморфізм на основі класів C#, запрограмувавши простий графічний редактор в об'єктно-орієнтованому стилі.