

**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 №4

з дисципліни
«ООП»

Виконала:

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

Перевірів:

Порєв В. М.

Київ 2023

Мета: : Мета роботи – отримати вміння та навички проектування класів, виконавши модернізацію коду графічного редактора в об'єктно-орієнтованому стилі для забезпечення зручного додавання нових типів об'єктів.

Завдання:

1. Створити у середовищі MS Visual Studio C++ проект Win32 з ім'ям Lab4.
2. Написати вихідний текст програми згідно варіанту завдання.
3. Скомпілювати вихідний текст і отримати виконуваний файл програми.
4. Перевірити роботу програми. Налогодити програму.
5. Проаналізувати та прокоментувати результати та вихідний текст програми.
6. 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;
using lab4.Shapes;
namespace lab4
{
    public partial class Form1 : Form
    {
        MyEditor editor = new();
        MyToolStrip toolStrip = new();
        Graphics gfx;
        Bitmap bmp;
        public Form1()
        {
            InitializeComponent();
        }

        private void pictureBox_MouseUp(object sender, MouseEventArgs e)
        {
            if (e.Button == MouseButton.Left)
            {
                editor.InitPen(Color.Black, Color.Yellow, DashStyle.Solid,
(float)brushSizePicker.Value);
                bmp = new(pictureBox.BackgroundImage);
                gfx = Graphics.FromImage(bmp);
                gfx.SmoothingMode = SmoothingMode.AntiAlias;
```

```

        editor.OnMouseUp(e, gfx);
        pictureBox.Image = bmp;
        editor.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);
        editor.InitPen(Color.Black, Color.Yellow, DashStyle.Dash,
(float)brushSizePicker.Value);
        editor.OnMouseDown(e);
    }
}

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

private void dotButton_Click(object sender, EventArgs e)
{
    toolStrip.Button((sender as ToolStripButton).ToolTipText);
    editor.Start(new Dot());
}

private void lineButton_Click(object sender, EventArgs e)
{
    toolStrip.Button((sender as ToolStripButton).ToolTipText);
    editor.Start(new Line());
}

private void rectangleButton_Click(object sender, EventArgs e)
{
    toolStrip.Button((sender as ToolStripButton).ToolTipText);
    editor.Start(new RectangleShape());
}

private void ellipseButton_Click(object sender, EventArgs e)
{
    toolStrip.Button((sender as ToolStripButton).ToolTipText);
    editor.Start(new Elipse());
}

private void cubeButton_Click(object sender, EventArgs e)
{
    toolStrip.Button((sender as ToolStripButton).ToolTipText);
    editor.Start(new Cube());
}

private void barButton_Click(object sender, EventArgs e)
{
    toolStrip.Button((sender as ToolStripButton).ToolTipText);
    editor.Start(new Bar());
}

private void clearButton_Click(object sender, EventArgs e)

```

```

        {
            Text = "Очищено";
            pictureBox.Image = null;
            pictureBox.BackgroundImage = null;
        }
    }
}

```

Dot.cs

```

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

namespace lab4.Shapes
{
    class Dot : Shape
    {
        public override void Show(Graphics g, Pen pen)
        {
            g.DrawRectangle(pen, this.x2, this.y2, pen.Width, pen.Width);
        }
    }
}

```

Shape.cs

```

using System.Drawing;

namespace lab4
{
    abstract class Shape
    {
        public int x1, y1, x2, y2;

        public void Set(int _x1, int _y1, int _x2, int _y2)
        {
            this.x1 = _x1;
            this.y1 = _y1;
            this.x2 = _x2;
            this.y2 = _y2;
        }

        public abstract void Show(Graphics g, Pen pen);
        public virtual void Fill(Graphics g, SolidBrush brush) { }
    }
}

```

MyEditor.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;
using System.Drawing.Drawing2D;
using lab4.Shapes;
namespace lab4
{
    class MyEditor
    {
        protected int x1, y1, x2, y2;
        protected Shape currShape;
        protected Pen pen;
        protected SolidBrush brush;
    }
}

```

```

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

    public void OnMouseUp(MouseEventArgs e, Graphics g)
    {
        this.x2 = e.X;
        this.y2 = e.Y;
        currShape.Set(x1, y1, x2, y2);
        currShape.Show(g, pen);
        currShape.Fill(g, brush);
    }

    public virtual void OnMouseMove(MouseEventArgs e, Graphics g)
    {
        this.x2 = e.X;
        this.y2 = e.Y;
        currShape.Set(x1, y1, x2, y2);
        currShape.Show(g, pen);
    }

    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();
    }
}
}

```

Bar.cs

```

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

namespace lab4.Shapes
{
    class Bar : Shape
    {
        Line line = new();
        Ellipse el1 = new();
        Ellipse el2 = new();
        int radius = 10;
        public override void Show(Graphics g, Pen pen)
        {
            line.Set(x1, y1, x2, y2);
            el1.Set(line.x1 - radius, line.y1 - radius, line.x1 + radius, line.y1 +
radius);
            el2.Set(line.x2 - radius, line.y2 - radius, line.x2 + radius, line.y2 +
radius);
        }
    }
}

```

```

        line.Show(g, pen);
        el1.Show(g, pen);
        el2.Show(g, pen);
    }
    public override void Fill(Graphics g, SolidBrush brush)
    {
        el1.Fill(g, brush);
        el2.Fill(g, brush);
    }
}

```

Cube.cs

```

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

namespace lab4.Shapes
{
    class Cube : Shape
    {
        public override void Show(Graphics g, Pen pen)
        {
            RectangleShape rect1 = new();
            RectangleShape rect2 = new();
            rect1.Set(x1, y1, x2, y2);
            rect2.Set(x1 + Math.Abs(x1 - x2), y1 + Math.Abs(x1 - x2), x2 + Math.Abs(x1 -
x2), y2 + Math.Abs(x1 - x2));
            rect1.Show(g, pen);
            rect2.Show(g, pen);
            g.DrawLine(pen, rect1.x1 + (rect1.x1 - rect1.x2), rect1.y1 + (rect1.y1 -
rect1.y2), rect2.x1 + (rect2.x1 - rect2.x2), rect2.y1 + (rect2.y1 - rect2.y2));
            g.DrawLine(pen, rect1.x1 + (rect1.x1 - rect1.x2), rect1.y1 - (rect1.y1 -
rect1.y2), rect2.x1 + (rect2.x1 - rect2.x2), rect2.y1 - (rect2.y1 - rect2.y2));
            g.DrawLine(pen, rect1.x1 - (rect1.x1 - rect1.x2), rect1.y1 + (rect1.y1 -
rect1.y2), rect2.x1 - (rect2.x1 - rect2.x2), rect2.y1 + (rect2.y1 - rect2.y2));
            g.DrawLine(pen, rect1.x1 - (rect1.x1 - rect1.x2), rect1.y1 - (rect1.y1 -
rect1.y2), rect2.x1 - (rect2.x1 - rect2.x2), rect2.y1 - (rect2.y1 - rect2.y2));
        }
    }
}

```

Ellipse.cs

```

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

namespace lab4.Shapes
{
    class Ellipse : Shape
    {
        public override void Show(Graphics g, Pen pen)
        {
            g.DrawEllipse(pen,
                (this.x1 > this.x2 ? this.x2 : this.x1),
                (this.y1 > this.y2 ? this.y2 : this.y1),
                Math.Abs(this.x1 - this.x2),
                Math.Abs(this.y1 - this.y2));
        }

        public override void Fill(Graphics g, SolidBrush brush)
        {
            g.FillEllipse(brush,
                (this.x1 > this.x2 ? this.x2 : this.x1),

```

```

        (this.y1 > this.y2 ? this.y2 : this.y1),
        Math.Abs(this.x1 - this.x2),
        Math.Abs(this.y1 - this.y2));
    }
}

```

Line.cs

```

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

namespace lab4.Shapes
{
    class Line : Shape
    {
        public override void Show(Graphics g, Pen pen)
        {
            g.DrawLine(pen, this.x1, this.y1, this.x2, this.y2);
        }
    }
}

```

RectangleShape.cs

```

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

namespace lab4.Shapes
{
    class RectangleShape : Shape
    {
        public override void Show(Graphics g, Pen pen)
        {
            g.DrawRectangle(pen,
                (this.x1 < this.x2 ? (this.x1 - this.x2 + this.x1) : this.x2),
                (this.y1 < this.y2 ? (this.y1 - this.y2 + this.y1) : this.y2),
                2 * Math.Abs(this.x1 - this.x2),
                2 * Math.Abs(this.y1 - this.y2));
        }
    }
}

```

MyToolStrip.cs

```

namespace lab4
{
    class MyToolStrip
    {
        public Form currentForm;
        public void Button(string text)
        {
            currentForm = Form.ActiveForm;
            this.currentForm.Text = text;
        }
    }
}

```

Результат роботи програми:

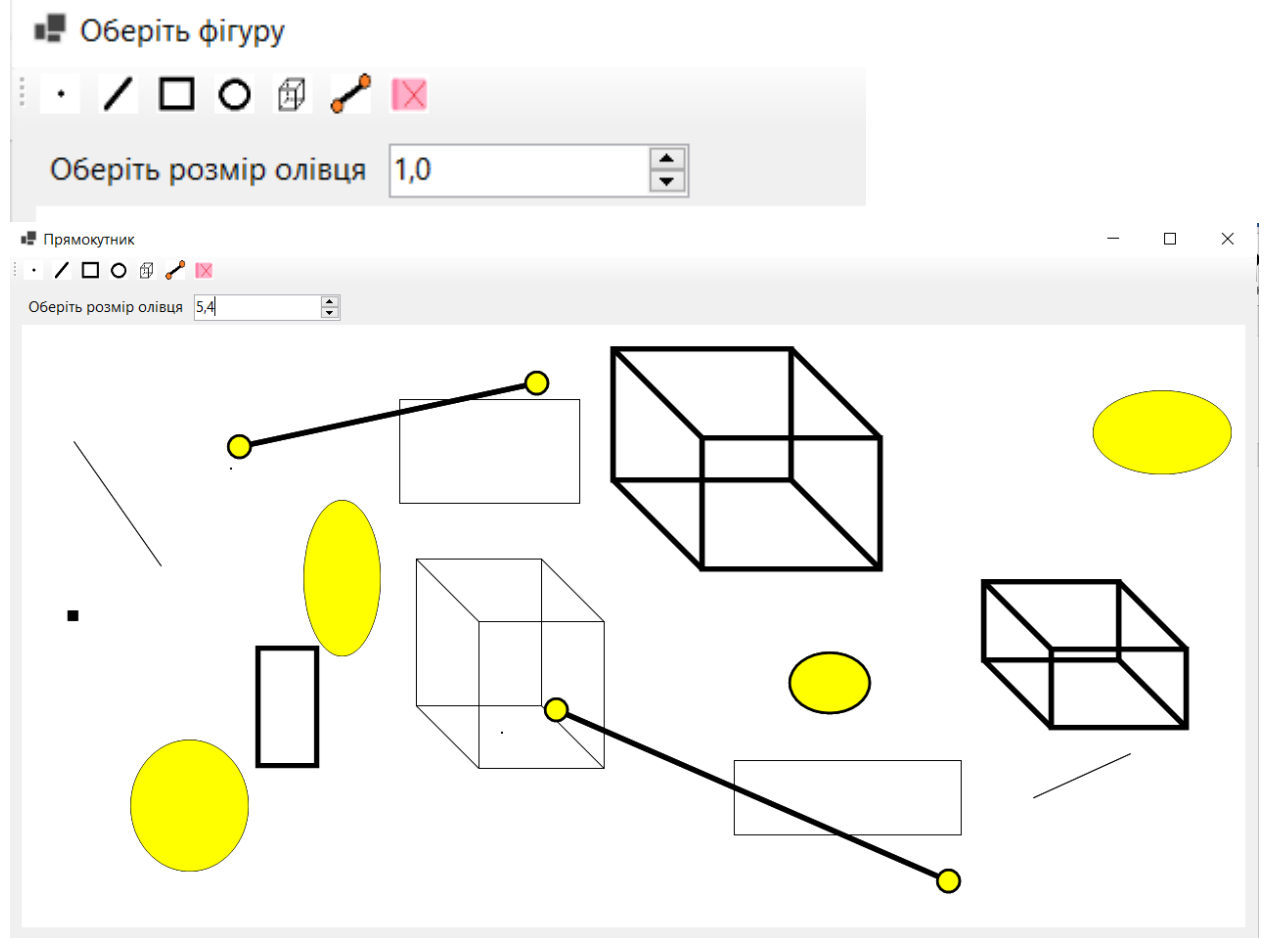
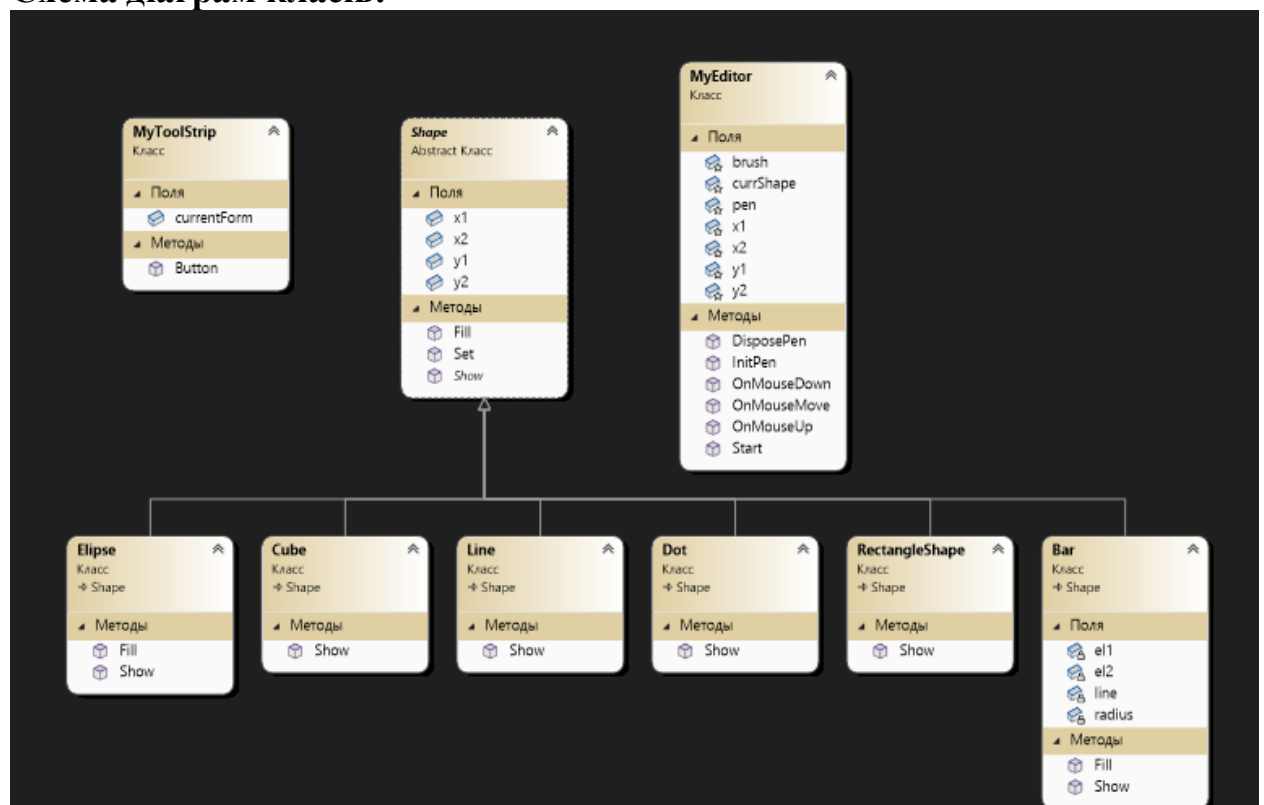


Схема діаграм класів:



Висновок: Ми отримали вміння та навички проектування класів, виконавши модернізацію коду графічного редактора в об'єктно-орієнтованому стилі для забезпечення зручного додавання нових типів об'єктів.