

**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 №5

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 програмувати багатовіконний інтерфейс програми на C++ в об'єктно-орієнтованому стилі.

Завдання:

1. Створити у середовищі MS Visual Studio C++ проект Desktop Application з ім'ям Lab5.
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.

Текст програми:

Form1.cs

```
using System.Drawing;
using System.Drawing.Drawing2D;
using lab5.Shapes;
namespace lab5
{
    public partial class Form1 : Form
    {
        MyEditor editor = MyEditor.Instance;
        MyTable ShapeTableForm = new MyTable();
        DataGridView Table;
        Button deleteBtn;
        MyToolStrip toolStrip = new();
        Graphics gfx;
        Bitmap bmp;
        Dictionary<string, Shape> shapeDict = new()
        {
            { "Крапка", new Dot() },
            { "Лінія", new Line() },
            { "Прямокутник", new RectangleShape() },
            { "Еліпс", new Ellipse() },
            { "Куб", new Cube() },
            { "Гантеля", new Bar() },
        };
        public Form1()
        {
            InitializeComponent();
            Table = ShapeTableForm.dataGridView1;
            Table.CellClick += this.ShapeTable_Cell_Click;
            deleteBtn = ShapeTableForm.deleteButton;
            deleteBtn.Click += this.DeleteBtn_Click;
        }

        private void pictureBox_MouseUp(object sender, MouseEventArgs e)
        {
            if (e.Button == MouseButton.Left)
            {
                if (shapeDict.ContainsKey(Text))
                {
                    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);
                }
            }
        }
    }
}
```

```

        ShapeTableForm.AddData(this.Text, editor.GetPoints(),
(float)brushSizePicker.Value);
        ShapeTableForm.ShowData();
        pictureBox.Image = bmp;
        editor.DisposePen();
    }
}

private void pictureBox_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        if (shapeDict.ContainsKey(Text))
        {
            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)
    {
        if (shapeDict.ContainsKey(Text))
        {
            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 tableButton_Click(object sender, EventArgs e)
    {
        if (!ShapeTableForm.Visible)
        {
            tableButton.Text = "Закрити таблицю";
            ShapeTableForm.Show();
        }
        else
        {
            tableButton.Text = "Відкрити таблицю";
            ShapeTableForm.Hide();
        }
    }

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

e) private void ShapeTable_Cell_Click(object sender, DataGridViewCellEventArgs
    {
        SelectOrErase(false);
    }

    private void DeleteBtn_Click(object sender, EventArgs e)
    {
        SelectOrErase(true);
    }

    bool SelectOrErase(bool toDelete)
    {
        if (Table.RowCount != 0)
        {
            int currRow = Table.CurrentCell.RowIndex;
            pictureBox.Image = new Bitmap(Width, Height);
            pictureBox.BackgroundImage = pictureBox.Image;
            for (int i = 0; i < Table.RowCount; i++)
            {
                int x1 = Convert.ToInt32(Table.Rows[i].Cells[1].Value);
                int y1 = Convert.ToInt32(Table.Rows[i].Cells[2].Value);
                int x2 = Convert.ToInt32(Table.Rows[i].Cells[3].Value);
                int y2 = Convert.ToInt32(Table.Rows[i].Cells[4].Value);
                string shape = Table.Rows[i].Cells[0].Value.ToString();
                float penSize =
                (float)Convert.ToInt32(Table.Rows[i].Cells[5].Value);
                string? content = String.Empty;
                if (i == currRow)
                {
                    editor.InitPen(Color.Red, Color.Blue, DashStyle.Solid,
                    penSize);
                    if (toDelete)
                    {
                        foreach (var line in File.ReadLines("Data.txt"))
                        {

```

```

        if (line != $"{shape} {x1} {y1} {x2} {y2}
{penSize}") content += $"{line}\n";
    }
    File.WriteAllText("Data.txt", content);
    ShapeTableForm.ShowData();
    SelectOrErase(false);
    return false;
}
}
else editor.InitPen(Color.Black, Color.Yellow, DashStyle.Solid,
penSize);

bmp = new(pictureBox.Image);
gfx = Graphics.FromImage(bmp);
editor.Start(shapeDict[shape], x1, y1, x2, y2, gfx);
pictureBox.Image = bmp;
editor.DisposePen();
if (shapeDict.ContainsKey(Text)) editor.Start(shapeDict[Text]);
}
}
return true;
}
}
}
}

```

Shape.cs

```

using System.Drawing;

namespace lab5
{
    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 lab5.Shapes;
namespace lab5
{
    class MyEditor
    {
        private static readonly MyEditor editorInstance = new MyEditor();
        protected int x1, y1, x2, y2;
        public Shape currShape;
        protected Pen pen;
    }
}

```

```

protected SolidBrush brush;
private MyEditor() {
}
public static MyEditor Instance
{
    get
    {
        return editorInstance;
    }
}
public void Start(Shape shape)
{
    currShape = shape;
}

public void Start(Shape shape, int _x1, int _y1, int _x2, int _y2, Graphics g)
{
    currShape = shape;
    this.x1 = _x1;
    this.x2 = _x2;
    this.y1 = _y1;
    this.y2 = _y2;
    currShape.Set(x1, y1, x2, y2);
    g.SmoothingMode = SmoothingMode.AntiAlias;
    currShape.Show(g, pen);
    currShape.Fill(g, brush);
}

public int[] GetPoints()
{
    return new int[] { x1, y1, x2, y2 };
}

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

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

public virtual void OnMouseMove(MouseEventArgs e, Graphics g)
{
    if (currShape != null)
    {
        this.x2 = e.X;
        this.y2 = e.Y;
        currShape.Set(x1, y1, x2, y2);
        g.SmoothingMode = SmoothingMode.AntiAlias;
        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();
    }
}
}

```

MyToolStrip.cs

```

namespace lab5
{
    class MyToolStrip
    {
        public Form currentForm;
        public void Button(string text)
        {
            currentForm = Form.ActiveForm;
            this.currentForm.Text = text;
        }
    }
}

```

MyTable.cs

```

namespace lab5
{
    public partial class MyTable : Form
    {
        public MyTable()
        {
            InitializeComponent();
            File.WriteAllText("Data.txt", String.Empty);
        }

        public void AddData(string name, int[] arr, float penSize)
        {
            if (arr[0] != 0)
            {
                File.AppendAllText("Data.txt", $"{name} {arr[0]} {arr[1]} {arr[2]}
{arr[3]} {penSize}\n");
            }
        }

        public void ShowData()
        {
            dataGridView1.Rows.Clear();
            foreach (var line in File.ReadLines("Data.txt"))
            {
                var array = line.Split();
                dataGridView1.Rows.Add(array);
            }
        }

        public void Clear()

```

```

    {
        File.WriteAllText("Data.txt", String.Empty);
        ShowData();
    }

    private void MyTable_FormClosing(object sender, FormClosingEventArgs e)
    {
        e.Cancel = true;
        Hide();
    }

    private void dataGridView1_CellContentClick(object sender,
    DataGridViewCellEventArgs e)
    {
    }
}
}
}

```

Результат роботи програми:

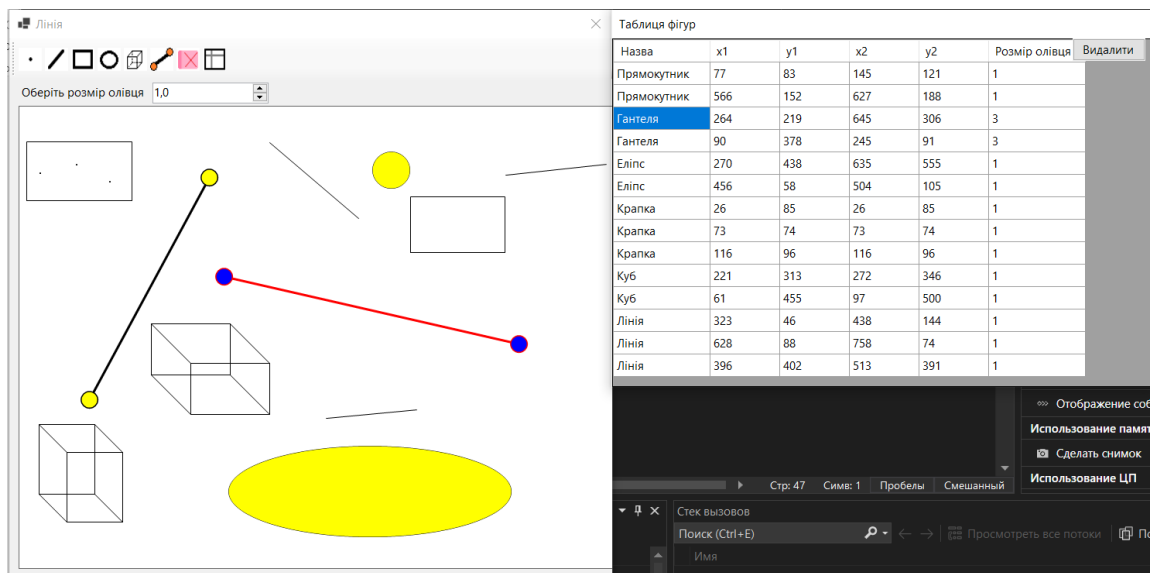
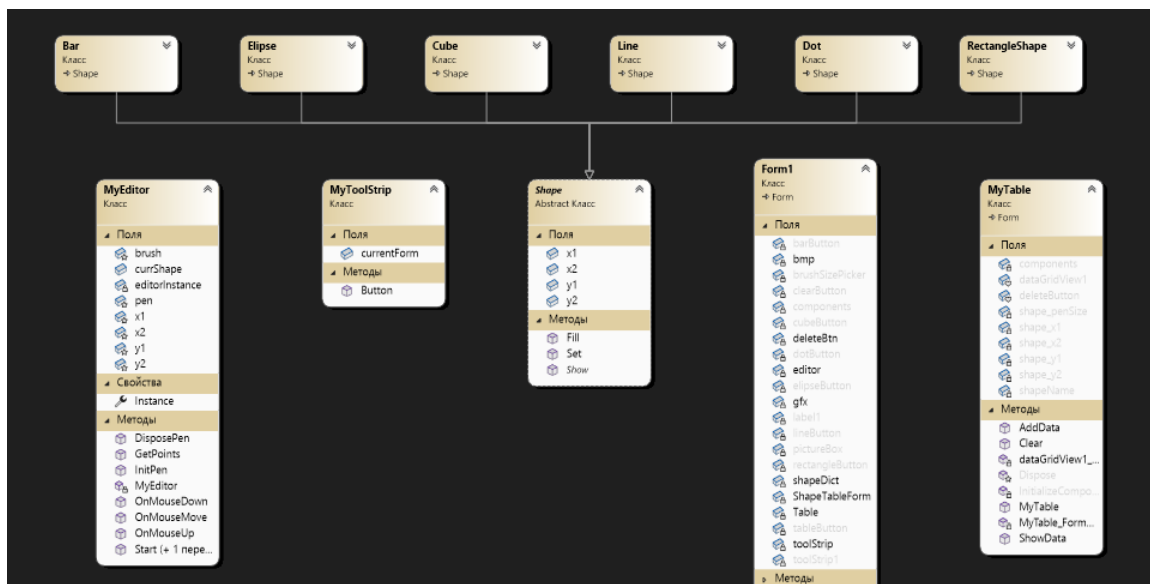


Схема діаграм класів:



Висновок: Ми отримали вміння та навички програмувати багатовіконний інтерфейс програми на C# в об'єктно-орієнтованому стилі.