Лабораторна робота №6
Делегати

Мета: Одержати практичні навички створення та роботи з делегатами.
Хід роботи

Завдання 1
Створити 3-4 делегати для проєкту WinForm
Код програми
MainForm.cs

```csharp
using LW_Daryev_WinForm_NewEdition.Nomalisation;
using LW_Daryev_WinForm_NewEdition.Draw;
using LW_Daryev_WinForm_NewEdition.Shape;
using LW_Daryev_WinForm_NewEdition.Brush;
using LW_Daryev_WinForm_NewEdition.HisotyManagment;


namespace LW_Daryev_WinForm_NewEdition
{
    public delegate void AddShapeToListDelegate(IShapeDraw shape);
    public delegate void ShowDrModeOnStatusStripDelegate(string mode);
    public delegate void ShowDrawingStatusOnStatusStrip(bool status);

    public partial class MainForm : Form
    {
        HistoryManager HistoryOnMainForm = new HistoryManager();
        public UserCanvas CanvasOnMainForm;
        AddShapeToListDelegate AddShapeToList;
        public ShowDrModeOnStatusStripDelegate ShowDrModeOnStatusStrip;
        public ShowDrawingStatusOnStatusStrip showDrawingStatusOnStatusStrip;

        public MainForm()
        {
            InitializeComponent();
            InitializeControlsLS();
            mainPicture.Image = new Bitmap(mainPicture.Width, mainPicture.Height);
            CanvasOnMainForm = new UserCanvas((mainPicture.Image as Bitmap), new
List<IBrushDraw>());
            SizeToolStripComboBox.Items.AddRange(new object[] { "1", "2", "3", "4",
"5", "6", "7", "8", "9", "10", "12", "14", "16", "18", "20", "24", "28", "32" });
            SizeToolStripComboBox.SelectedIndex = 4;
            AddShapeToList += (IShapeDraw obj) => {
CanvasOnMainForm.ShapesList.Add(obj); };
            ShowDrModeOnStatusStrip += (string mode) => { StatusModeStripValue.Text
= $"{mode}"; };
            showDrawingStatusOnStatusStrip += (bool status) => {
DrawingStatusShow.Text = status ? "drawing" : " "; };
        }
        public void InitializeControlsLS()
        {
            menuBarStrip.Location = NormaliseMashine.DenormalisePoint(this, 0.0f,
0.0f);
            menuBarStrip.Size = NormaliseMashine.DenormaliseSize(this, 1.0f, 0.05f);

            toolsToolStrip.Location = NormaliseMashine.DenormalisePoint(this, 0.0f,
0.1f);
```

```csharp
            toolsToolStrip.Size = NormaliseMashine.DenormaliseSize(this, 0.11f,
1.0f);

            propToolStrip.Location = NormaliseMashine.DenormalisePoint(this, 0.11f,
0.05f);
            propToolStrip.Size = NormaliseMashine.DenormaliseSize(this, 0.89f,
0.05f);

            mainPicture.Location = NormaliseMashine.DenormalisePoint(this, 0.13f,
0.13f);
            mainPicture.Size = NormaliseMashine.DenormaliseSize(this, 0.85f, 0.85f);
        }
        public void SetControlsLS()
        {
            mainPicture.Location = NormaliseMashine.DenormalisePoint(this, 0.13f,
0.08f);
            mainPicture.Size = NormaliseMashine.DenormaliseSize(this, 0.85f, 0.85f);
        }
        private void mainPicture_MouseDown(object sender, MouseEventArgs e)
        {
            if (CanvasOnMainForm.IsDrawing) return;

            showDrawingStatusOnStatusStrip(true);
            CanvasOnMainForm.IsDrawing = true;
            float SizeValue;
            Color ColorValue = ColorInfoAndChangeToolStripButton.BackColor;
            try
            {
                SizeValue =
float.Parse(SizeToolStripComboBox.SelectedItem.ToString());
            }
            catch
            {
                SizeToolStripComboBox.SelectedIndex = 4;
                SizeValue =
float.Parse(SizeToolStripComboBox.SelectedItem.ToString());
            }
            switch (CanvasOnMainForm.DrMode)
            {
                case MyDrawMode.NONE: return;
                case MyDrawMode.LINE:
                    AddShapeToList(new MyLine(e.Location, e.Location, ColorValue,
SizeValue));
                    break;
                case MyDrawMode.RECTANGLE:
                    AddShapeToList(new MyRectangle(e.Location, e.Location,
ColorValue, SizeValue));
                    break;
                case MyDrawMode.TRIANGLE:
                    AddShapeToList(new MyTriangle(e.Location, e.Location,
ColorValue, SizeValue));
                    break;
                case MyDrawMode.ELLIPSE:
                    AddShapeToList(new MyEllipse(e.Location, e.Location, ColorValue,
SizeValue));
                    break;
                case MyDrawMode.CIRCLE:
                    AddShapeToList(new MyCircle(e.Location, e.Location, ColorValue,
SizeValue));
```

```csharp
                        break;
                    case MyDrawMode.BRUSH:
                        CanvasOnMainForm.BrushList.Add(new MyBrush(e.Location,
ColorValue, SizeValue));
                        break;
                    case MyDrawMode.ERRASER:
                        CanvasOnMainForm.BrushList.Add(new MyBrush(e.Location,
Color.White, SizeValue));
                        break;
                    default: return;

                }

        }
        private void mainPicture_MouseMove(object sender, MouseEventArgs e)
        {
            if (CanvasOnMainForm.DrMode == MyDrawMode.NONE) return;
            if (CanvasOnMainForm.IsDrawing)
            {
                if (CanvasOnMainForm.DrMode == MyDrawMode.BRUSH ||
CanvasOnMainForm.DrMode == MyDrawMode.ERRASER)
                {
                    if (CanvasOnMainForm.BrushList.Any())
                        CanvasOnMainForm.BrushList.Last().AddPosition(e.Location);
                }
                else
                {
                    CanvasOnMainForm.ShapesList.Last().EndPoint = e.Location;
                }
                mainPicture.Invalidate();

            }
        }
        private void mainPicture_MouseUp(object sender, MouseEventArgs e)
        {
            if (CanvasOnMainForm.DrMode == MyDrawMode.NONE) return;
            showDrawingStatusOnStatusStrip(false);

            if (CanvasOnMainForm.IsDrawing)
            {
                if (CanvasOnMainForm.DrMode == MyDrawMode.BRUSH)
                {
                    if (CanvasOnMainForm.BrushList.Any())
                        CanvasOnMainForm.BrushList.Last().AddPosition(e.Location);
                }
                else { CanvasOnMainForm.ShapesList.Last().EndPoint = e.Location; }

                CanvasOnMainForm.DrawAll();
                mainPicture.Invalidate();
                CanvasOnMainForm.IsDrawing = false;
                HistoryOnMainForm.SaveState(CanvasOnMainForm.ShapesList,
CanvasOnMainForm.BrushList);
            }
        }
        private void mainPicture_Paint(object sender, PaintEventArgs e)
        {
            CanvasOnMainForm.DrawPreview(e.Graphics);
        }
```

```csharp
        private void ColorInfoAndChangeToolStripButton_Click(object sender,
EventArgs e)
        {
            ColorDialog colorDialog = new ColorDialog();
            if (colorDialog.ShowDialog() == DialogResult.OK)
            {
                ColorInfoAndChangeToolStripButton.BackColor = colorDialog.Color;
            }
        }

        private void toolStripLabel3_Click(object sender, EventArgs e) //Undo
        {
            var state = HistoryOnMainForm.Undo();
            if (state == null) return;

            CanvasOnMainForm.ShapesList = new
List<IShapeDraw>(state.ShapesSnapshot);
            CanvasOnMainForm.BrushList = new List<IBrushDraw>(state.BrushSnapshot);

            CanvasOnMainForm.ClearCanvas();
            CanvasOnMainForm.DrawAll();
            mainPicture.Invalidate();
        }
        private void toolStripLabel5_Click(object sender, EventArgs e) //Redo
        {
            var state = HistoryOnMainForm.Redo();
            if (state == null) return;

            CanvasOnMainForm.ShapesList = new
List<IShapeDraw>(state.ShapesSnapshot);
            CanvasOnMainForm.BrushList = new List<IBrushDraw>(state.BrushSnapshot);

            CanvasOnMainForm.ClearCanvas();
            CanvasOnMainForm.DrawAll();
            mainPicture.Invalidate();
        }
    }
}
```

Висновок:
На лабораторній роботі було одержано практичні навички створення та
роботи з делегатами.

ЛР.ОК.19.ПІ231.04.06