МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Криворізький національний університет

Кафедра моделювання та програмного забезпечення

Звіт

з лабораторної роботи №1

УПРАВЛІННЯ ВЛАСТИВОСТЯМИ ГРАФІЧНИХ ОБ`ЄКТІВ ДОДАТКІВ

Студент групи ІПЗ-21-2
Губарєв Р.В.
+380980190289
Викладачі
Козиков А. В.
Гриценко А. М.

Кривий Ріг

2022

1. Короткі теоретичні відомості про середовище розробки проекту

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які містять інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів. Ці продукти дають змогу розробляти як консольні програми, так і програми з графічним інтерфейсом, включно з підтримкою технології Windows Forms.

2. Короткі теоретичні відомості про візуальні об`єкти (компонентів), що використовуються.

MenuStrip – створює меню, що настроюється

GroupBox – групує набір елементів керування

PictureBox – відображає графічні файли, такі як растрові малюнки та піктограми

TrackBar – дозволяє користувачу задавати значення на шкалі масштабу, переміщуючи курсор

Label – відображає недоступний для зміни користувачем текст

Button – запускає, зупиняє чи перериває процес

3. Вихідний текст програми

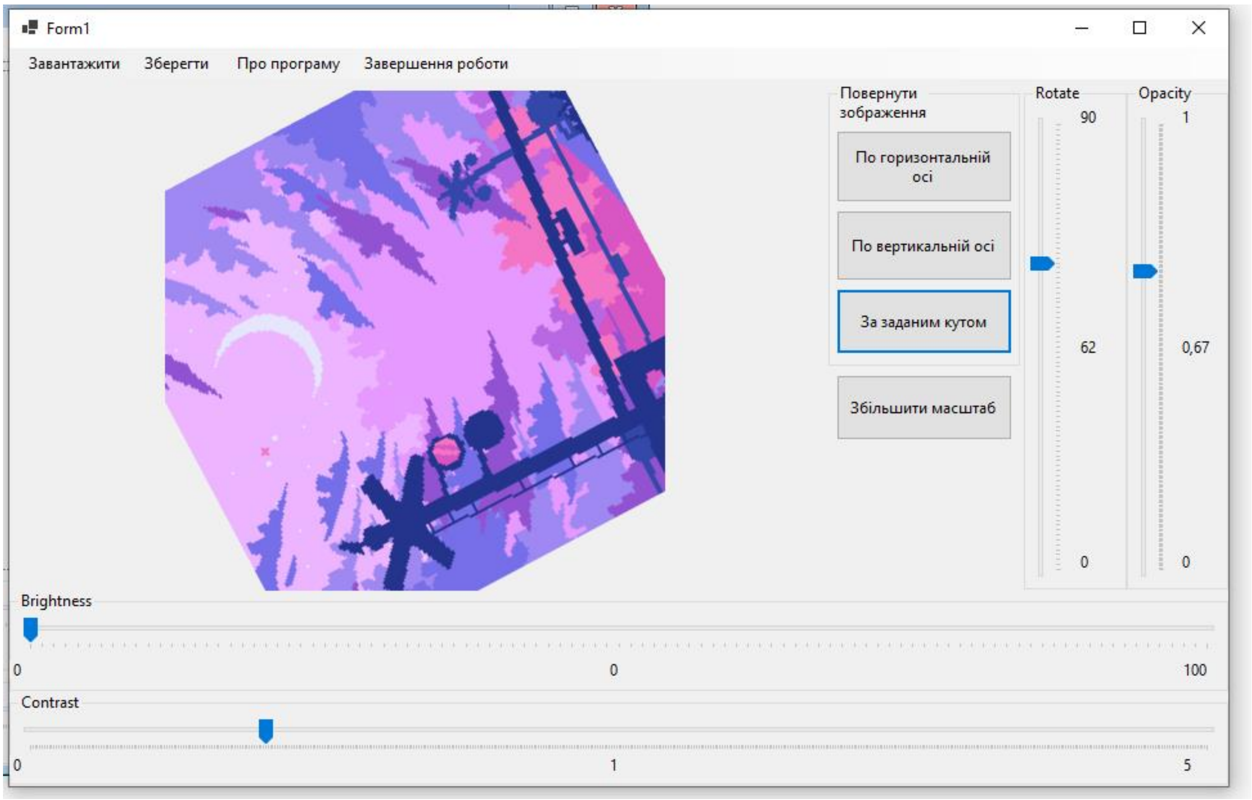
Form1

```
using Emgu.CV;
using Emgu.CV.Structure;
using System.Drawing.Drawing2D;
using System.Drawing.Imaging;
using System.Windows.Forms;

namespace Lab1
{
    public partial class Form1 : Form
    {
        public Image<Bgr, byte> ImgInput { get; set; }
        Image<Bgr, byte> imgOutput;
        Image original = null;
        public Form1()
        {
            InitializeComponent();
        }

        private void panel1_Paint(object sender, PaintEventArgs e)
        {

        }

        private void Form1_Load(object sender, EventArgs e)
        {
            try
            {
                if (ImgInput != null)
                {
                    pictureBox1.Image = ImgInput.AsBitmap();
                }

                lblCurrentBrightness.Text = trackBar2.Value.ToString();
                lblMinBrightness.Text = trackBar2.Minimum.ToString();
                lblMaxBrightness.Text = trackBar2.Maximum.ToString();

                lblMinContrast.Text = ((float)trackBar1.Minimum /
100).ToString();
                lblMaxContrast.Text = ((float)trackBar1.Maximum /
100).ToString();
                lblCurrentContrast.Text = ((float)trackBar1.Value /
100).ToString();

                lblCurrentOpacity.Text = ((float)trackBar3.Value /
255).ToString();
                lblMinOpacity.Text = ((float)trackBar3.Minimum /
255).ToString();
                lblMaxOpacity.Text = ((float)trackBar3.Maximum /
255).ToString();

                lblCurrentRotate.Text = trackBar4.Value.ToString();
                lblMinRotate.Text = trackBar4.Minimum.ToString();
                lblMaxRotate.Text = trackBar4.Maximum.ToString();
            }
            catch (Exception)
            {

                throw;
            }
        }

        private void ContrastBrightnessAdjust()
        {
            try
            {
                lblCurrentContrast.Text = ((float)trackBar1.Value /
100).ToString();
                lblCurrentBrightness.Text = trackBar2.Value.ToString();
                imgOutput =
ImgInput.Mul(double.Parse(lblCurrentContrast.Text)) + trackBar2.Value;
                pictureBox1.Image = imgOutput.AsBitmap();
            }
            catch (Exception)
            {

                throw;
            }
        }

        private void OpacityAdjust()
        {
            lblCurrentOpacity.Text = Math.Round(((float)trackBar3.Value /
255), 2).ToString();
        }
        private void RotateAdjust()
        {
            lblCurrentRotate.Text = trackBar4.Value.ToString();
        }

        private void trackBar1_Scroll(object sender, EventArgs e)
        {
            try
            {
                ContrastBrightnessAdjust();
            }
            catch (Exception)
            {

                throw;
            }
        }

        private void trackBar2_Scroll(object sender, EventArgs e)
        {
            try
            {
                ContrastBrightnessAdjust();
            }
            catch (Exception)
            {

                throw;
            }
        }

        private void завантажитиРисунокToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            OpenFileDialog dialog = new OpenFileDialog();
            dialog.Filter = "Images (*.jpg;*.png;*.bmp)|*.jpg;*.png;*.bmp;";

            if (dialog.ShowDialog() == DialogResult.OK)
            {
                ImgInput = new Image<Bgr, byte>(dialog.FileName);
                pictureBox1.Image = ImgInput.AsBitmap();
            }
        }

        private void зберегтиToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            try
            {
                SaveFileDialog dialog = new SaveFileDialog();
                dialog.Filter = "Images (*.jpg;)|*.jpg;| Images
(*.png;)|*.png;| Images(*.bmp)|*.bmp";

                if (dialog.ShowDialog() == DialogResult.OK)
                {
                    imgOutput.Save(dialog.FileName);
                    MessageBox.Show("Image saved successfully.");
                }
            }
            catch (Exception)
            {

                throw;
            }
        }

        private void завершенняРоботиToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            Close();
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
        }

        private void trackBar3_Scroll(object sender, EventArgs e)
        {
            if (original == null) original =
(Bitmap)pictureBox1.Image.Clone();
            pictureBox1.BackColor = Color.Transparent;
            pictureBox1.Image = SetAlpha((Bitmap)original, trackBar3.Value);

            OpacityAdjust();
        }

        static Bitmap SetAlpha(Bitmap bmpIn, int alpha)
        {
            Bitmap bmpOut = new Bitmap(bmpIn.Width, bmpIn.Height);
            float a = alpha / 255f;
            Rectangle r = new Rectangle(0, 0, bmpIn.Width, bmpIn.Height);

            float[][] matrixItems = {
            new float[] {1, 0, 0, 0, 0},
            new float[] {0, 1, 0, 0, 0},
            new float[] {0, 0, 1, 0, 0},
            new float[] {0, 0, 0, a, 0},
            new float[] {0, 0, 0, 0, 1}};

            ColorMatrix colorMatrix = new ColorMatrix(matrixItems);

            ImageAttributes imageAtt = new ImageAttributes();
            imageAtt.SetColorMatrix(colorMatrix, ColorMatrixFlag.Default,
ColorAdjustType.Bitmap);

            using (Graphics g = Graphics.FromImage(bmpOut))
                g.DrawImage(bmpIn, r, r.X, r.Y, r.Width, r.Height,
GraphicsUnit.Pixel, imageAtt);

            return bmpOut;
        }

        private void trackBar4_Scroll(object sender, EventArgs e)
        {
            RotateAdjust();
        }

        private Bitmap RotateImage(Bitmap bmp, float angle)
        {
            Bitmap rotatedImage = new Bitmap(bmp.Width, bmp.Height);
            rotatedImage.SetResolution(bmp.HorizontalResolution,
bmp.VerticalResolution);

            using (Graphics g = Graphics.FromImage(rotatedImage))
            {
                g.TranslateTransform(bmp.Width / 2, bmp.Height / 2);
                g.RotateTransform(angle);
                g.TranslateTransform(-bmp.Width / 2, -bmp.Height / 2);
                g.DrawImage(bmp, new Point(0, 0));
            }

            return rotatedImage;
        }

        private void проПрограмуToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            AboutBox1 aboutBox = new AboutBox1();
            aboutBox.Show();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Image img = pictureBox1.Image;
            img.RotateFlip(RotateFlipType.RotateNoneFlipX);
            pictureBox1.Image = img;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Image img = pictureBox1.Image;
            img.RotateFlip(RotateFlipType.RotateNoneFlipY);
            pictureBox1.Image = img;
        }

        private void button3_Click(object sender, EventArgs e)
        {
            if (original == null) original =
(Bitmap)pictureBox1.Image.Clone();
            pictureBox1.Image = RotateImage((Bitmap)original,
trackBar4.Value);
        }

        public Image ResizeImg(Image b, int nWidth, int nHeight)
        {
            Image result = new Bitmap(nWidth, nHeight);
            using (Graphics g = Graphics.FromImage((Image)result))
            {
                g.InterpolationMode = InterpolationMode.HighQualityBicubic;
                g.DrawImage(b, 0, 0, nWidth, nHeight);
                g.Dispose();
            }
            return result;
        }

        private void button4_Click(object sender, EventArgs e)
        {
            Image img = pictureBox1.Image;
            pictureBox1.Image = ResizeImg(img, 100, 70);
        }
    }
}
```

AboutBox1

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

namespace Lab1
{
    partial class AboutBox1 : Form
    {
        public AboutBox1()
        {
            InitializeComponent();
            this.Text = "Про програму";
            this.labelProductName.Text = "Лабораторна робота 1";
            this.labelVersion.Text = String.Format("Version {0}",
AssemblyVersion);
            this.labelCopyright.Text = AssemblyCopyright;
            this.labelCompanyName.Text = "Rostik Hubariev";
        }

        #region Assembly Attribute Accessors

        public string AssemblyTitle
        {
            get
            {
                object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyTitleAttribute)
, false);
                if (attributes.Length > 0)
                {
                    AssemblyTitleAttribute titleAttribute =
(AssemblyTitleAttribute)attributes[0];
                    if (titleAttribute.Title != "")
                    {
                        return titleAttribute.Title;
                    }
                }
                return
System.IO.Path.GetFileNameWithoutExtension(Assembly.GetExecutingAssembly().CodeBas
e);
            }
        }

        public string AssemblyVersion
        {
            get
            {
                return
Assembly.GetExecutingAssembly().GetName().Version.ToString();
            }
        }

        public string AssemblyDescription
        {
            get
            {
                object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyDescriptionAttr
ibute), false);
                if (attributes.Length == 0)
                {
                    return "";
                }
                return ((AssemblyDescriptionAttribute)attributes[0]).Description;
            }
        }

        public string AssemblyProduct
        {
            get
            {
                object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyProductAttribut
e), false);
                if (attributes.Length == 0)
                {
                    return "";
                }
                return ((AssemblyProductAttribute)attributes[0]).Product;
            }
        }

        public string AssemblyCopyright
        {
            get
            {
                object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyCopyrightAttrib
ute), false);
                if (attributes.Length == 0)
                {
                    return "";
                }
                return ((AssemblyCopyrightAttribute)attributes[0]).Copyright;
            }
        }

        public string AssemblyCompany
        {
            get
            {
                object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyCompanyAttribut
e), false);
                if (attributes.Length == 0)
                {
                    return "";
                }
                return ((AssemblyCompanyAttribute)attributes[0]).Company;
            }
        }
        #endregion

        private void AboutBox1_Load(object sender, EventArgs e)
        {

        }

        private void okButton_Click(object sender, EventArgs e)
        {
            Close();
        }
    }
}
```

4. Знімок інтерфейсу програми з результатами роботи



5. Висновки

В цій лабораторній роботі я ознайомився з Microsoft Forms. Також я ознайомився з інтерфейсом програми і навчився створювати просту форму з «редактором» зображення.