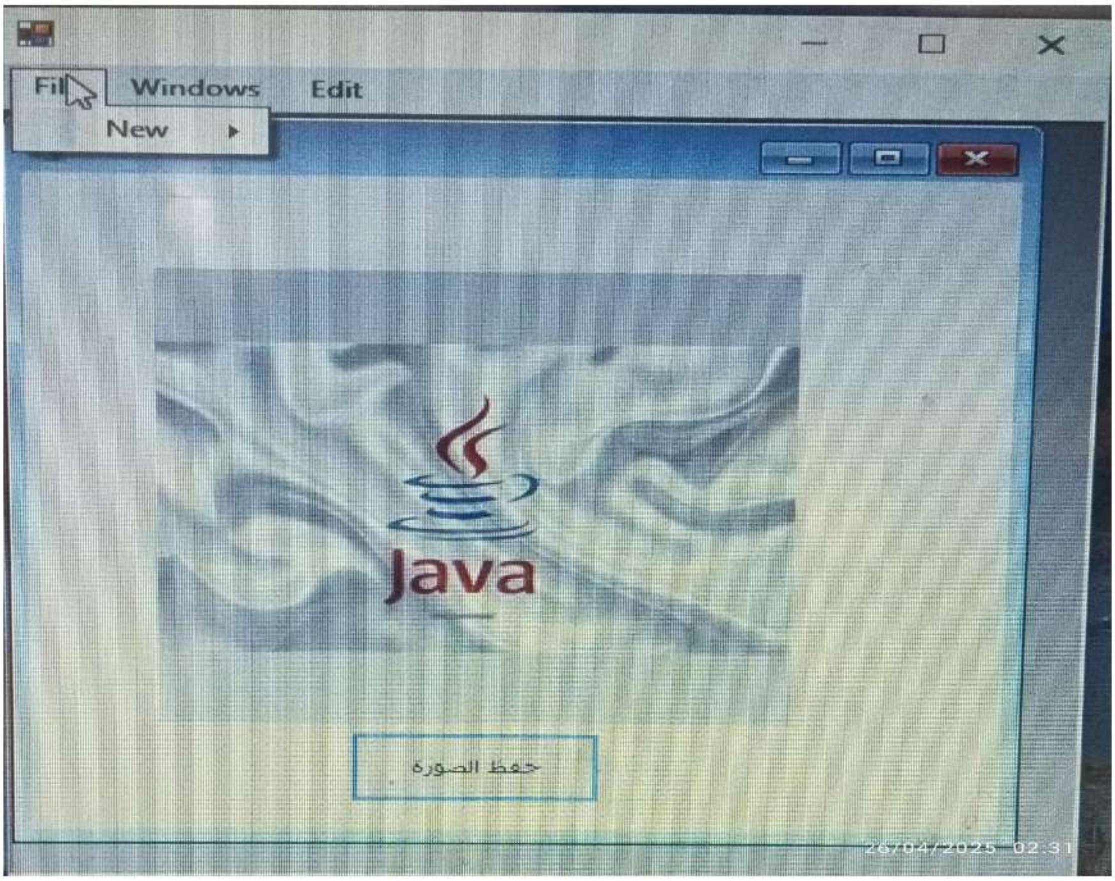
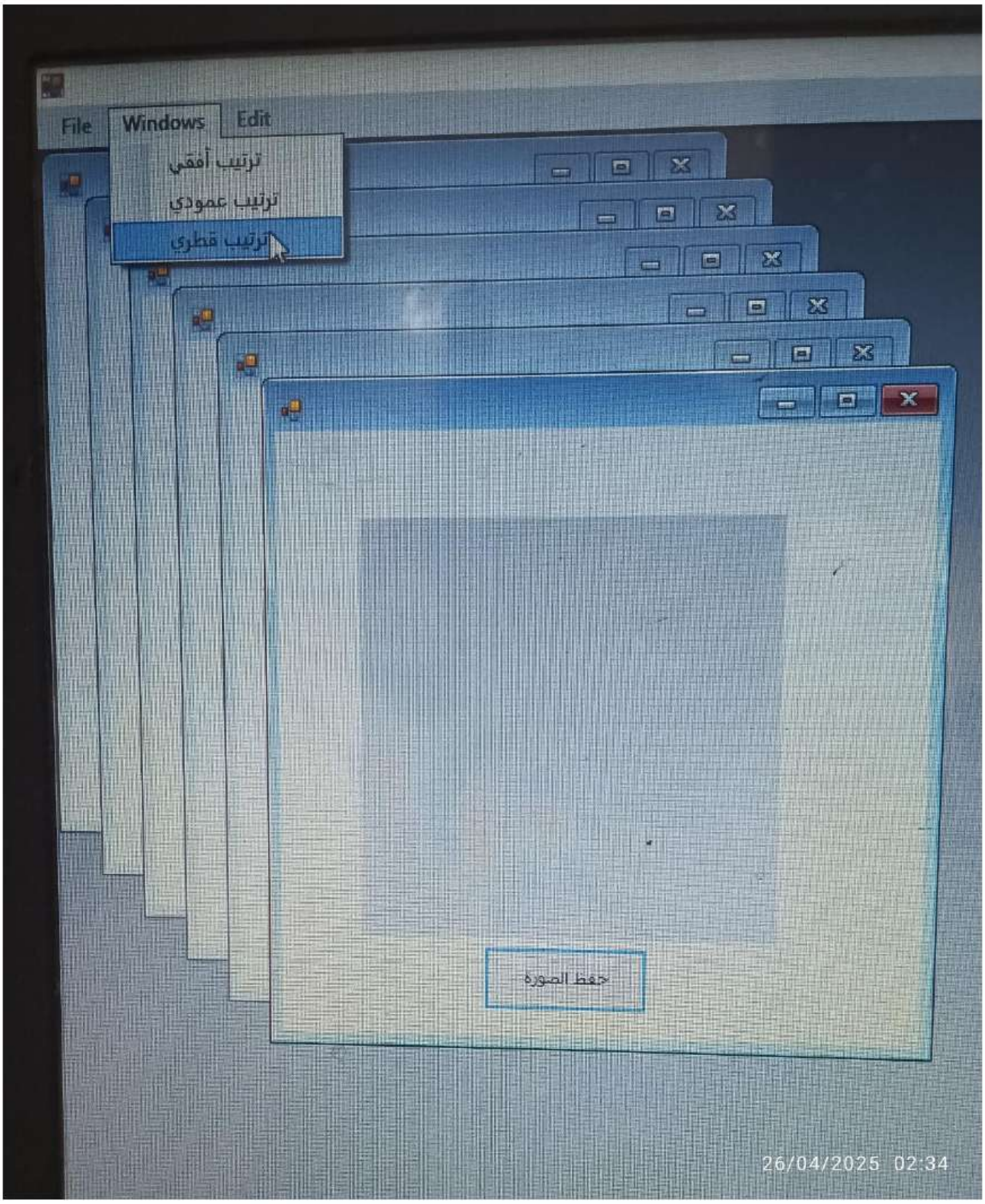
برمجة متقدمة ADVANCE PROGRAMMING IN C#

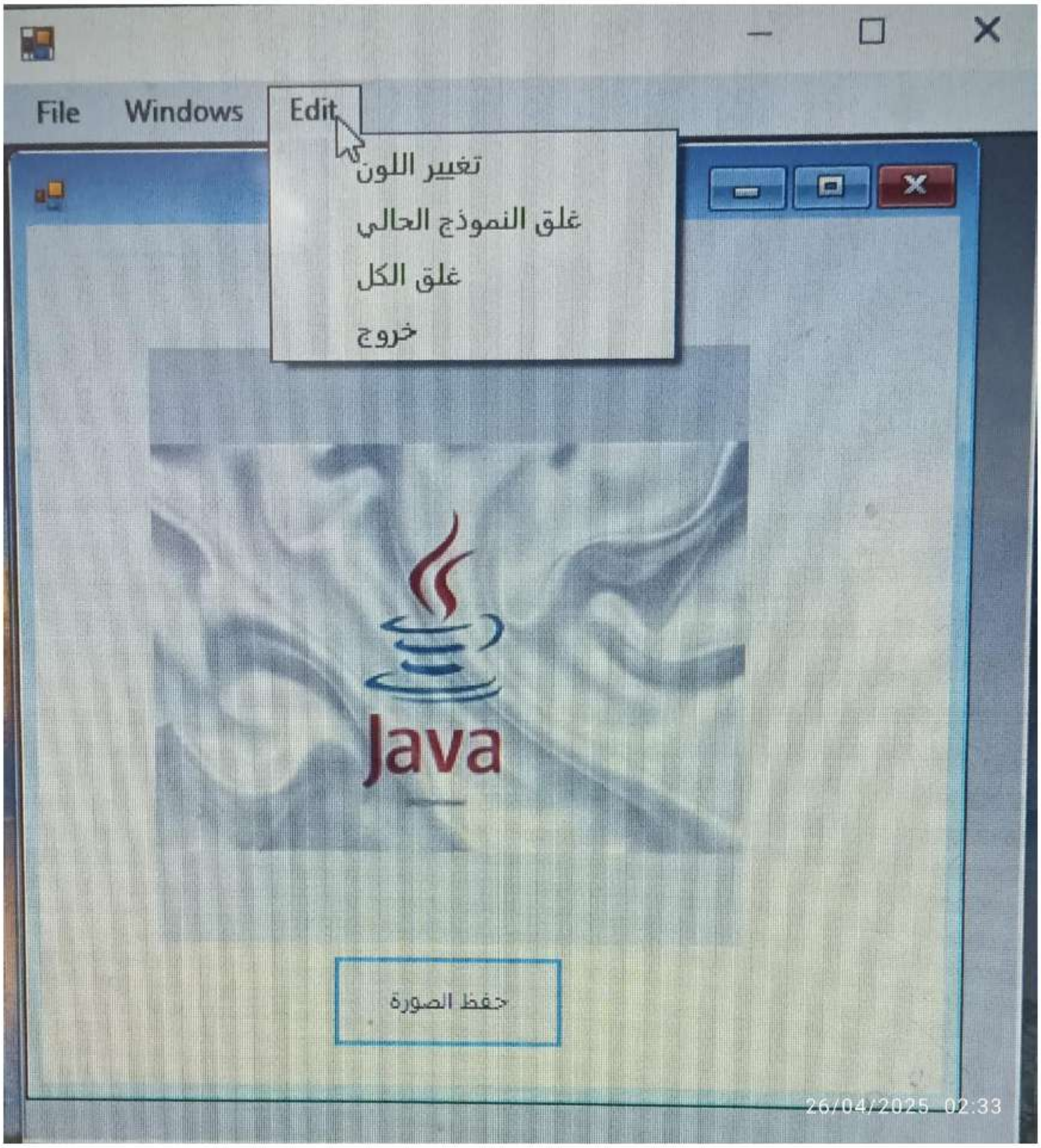
المحاضرة الثالث عشر

علوم حاسوب وتقنية المعلومات - مستوى ثالث – ترم ثاني

أ/ نجم الدين الدغار najmuddin.developer@gmail.com


File
Windows
Edit
New
Java
حفظ الصورة
28/04/2025 02:31


File
Windows
Edit
ترتيب أفقي
ترتيب عمودي
ترتيب قطري
حفظ الصورة
26/04/2025 02:34



الكود

```
using System;
using System.Drawing;
using System.Windows.Forms;
using System.IO;
using System.ComponentModel;
namespace Advance_Programming_Labs.lab13
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            this.IsMdiContainer = true;
            this.WindowState = FormWindowState.Maximized;
            CreateMenu();
            LoadSavedColor();
        }
        private void CreateMenu()
        {
            // إنشاء القائمة الرئيسية
            MenuStrip mainMenu = new MenuStrip();

            // قائمة File
            ToolStripMenuItem fileMenu = new ToolStripMenuItem("&File");
            ToolStripMenuItem newItem = new ToolStripMenuItem("&New",
null/*الايقونة*//*,*//*NewItem_Click الحدث*/);
            newItem.DropDownItems.Add("nothing");
            newItem.Click += NewItem_Click;//يمكن ارسالة كبارمتر عبر المشيد
            fileMenu.DropDownItems.Add(newItem);

            // قائمة Windows
            ToolStripMenuItem windowsMenu = new
ToolStripMenuItem("&Windows");
            windowsMenu.DropDownItems.Add("ترتيب أفقي", null,
ArrangeHorizontal_Click);
            windowsMenu.DropDownItems.Add("ترتيب عمودي", null,
ArrangeVertical_Click);
            windowsMenu.DropDownItems.Add("ترتيب قطري", null,
ArrangeCascade_Click);

            // قائمة Edit
            ToolStripMenuItem editMenu = new ToolStripMenuItem("&Edit");

            editMenu.DropDownItems.Add("&تغيير اللون", null, SetColor);
            editMenu.DropDownItems.Add("&غلق النموذج الحالي", null,
CloseCurrent_Click);
            editMenu.DropDownItems.Add("&غلق الكل", null, CloseAll_Click);
            editMenu.DropDownItems.Add("&خروج", null, Exit_Click);

            mainMenu.Items.AddRange(new ToolStripItem[] { fileMenu,
windowsMenu, editMenu });
            this.Controls.Add(mainMenu);
        }

        private void SetColor(object sender, EventArgs e)
        {
            ColorDialog colorDialog = new ColorDialog();

            if (colorDialog.ShowDialog() == DialogResult.OK)
            {
                this.ForeColor = colorDialog.Color;
                SaveColor();
            }
        }
        private void SaveColor()
        {
            File.WriteAllText("color.txt",
                this.ForeColor.ToArgb().ToString());
        }

        private void LoadSavedColor()
        {
            if (File.Exists("color.txt"))
            {
                int argb = Convert.ToInt32(File.ReadAllText("color.txt"));
                this.ForeColor = Color.FromArgb(argb);
            }
        }
        private void NewItem_Click(object sender, EventArgs e)
        {
            ImageViewerForm imageViewerForm1 = new ImageViewerForm();
            imageViewerForm1.MdiParent = this;
            imageViewerForm1.Show();
        }
        // ترتيب النماذج
        private void ArrangeHorizontal_Click(object sender, EventArgs e)
        {
            //1
            //LayoutMdi(MdiLayout.TileHorizontal);
            //2
            if (Application.OpenForms.Count > 1)
            {
                int width = this.ClientRectangle.Width;
                int height = this.ClientRectangle.Height /
(Application.OpenForms.Count - 1);

                for (int i = 1; i < Application.OpenForms.Count; i++)
                {
                    Application.OpenForms[i].Width = width;
                    Application.OpenForms[i].Height = height;
                    Application.OpenForms[i].Left =
this.ClientRectangle.Left;
                    Application.OpenForms[i].Top = (i - 1) * height +
this.ClientRectangle.Top;
                }
            }
        }
        private void ArrangeVertical_Click(object sender, EventArgs e)
        {//1
         //LayoutMdi(MdiLayout.TileVertical);
         //2
            if (Application.OpenForms.Count > 1)
            {
                int width = this.ClientRectangle.Width /
(Application.OpenForms.Count - 1);
                int height = this.ClientRectangle.Height;

                for (int i = 1; i < Application.OpenForms.Count; i++)
                {
                    Application.OpenForms[i].Width = width;
                    Application.OpenForms[i].Height = height;
                    Application.OpenForms[i].Left = (i - 1) * width +
this.ClientRectangle.Left;
                    Application.OpenForms[i].Top =
this.ClientRectangle.Top;
                }
            }
        }
        private void ArrangeCascade_Click(object sender, EventArgs e)
        {
            //1
            //LayoutMdi(MdiLayout.Cascade);
            //2
            if (Application.OpenForms.Count > 1)
            {

                int width = this.ClientRectangle.Width / 4;
                int height = this.ClientRectangle.Height / 4;
                width *= 3;
                height *= 3;
                for (int i = 1; i < Application.OpenForms.Count; i++)
                {

                    Application.OpenForms[i].Width = width;
                    Application.OpenForms[i].Height = height;
                    Application.OpenForms[i].Left = (i - 1) * 25 +
this.ClientRectangle.Left;
                    Application.OpenForms[i].Top = (i - 1) * 25 +
this.ClientRectangle.Top;
                }
            }
        }
        private void CloseCurrent_Click(object sender, EventArgs e) =>
ActiveMdiChild?.Close();
        private void CloseAll_Click(object sender, EventArgs e)
        {    //------1----------
            foreach (Form form in MdiChildren)
            {
                form.Close();
            }
            //--------2---------

            //int c = Application.OpenForms.Count;
            //for (int i = 1; i < c;i++)
            //{
            //    Application.OpenForms[1].Close();
            //}

            //--------3----------
            //c = MdiChildren.Length;
            //for (int i = 0; i <c; i++)
            //{
            //    MdiChildren[0].Close();
            //}
        }
        private void Exit_Click(object sender, EventArgs e) =>
Application.Exit();
        [STAThread]
        static void Main(string[] args)
        {
            Application.EnableVisualStyles();
            MainForm form = new MainForm();
            form.ShowDialog();
        }
    }
    //---------------------- public partial class ImageViewerForm : Form
    {
        private PictureBox pictureBox;
        private Button btnSave;
        public ImageViewerForm()
        {
            Createui();
            SetupDragDrop();
        }
        private void Createui()
        {
            // إنشاء عناصر الواجهة
            this.Size = new Size(400, 400);

            pictureBox = new PictureBox
            {
                Size = new Size(250, 250),
                Location = new Point(50, 50),
                BackColor = Color.LightGray,
                SizeMode = PictureBoxSizeMode.Zoom,
                AllowDrop = true
            };
            btnSave = new Button
            {
                Text = "حفظ الصورة",
                Size = new Size(100, 40),
                Location = new Point(pictureBox.Left + pictureBox.Width /
2 - 50/*نصف عرض الزر 50*/, pictureBox.Bottom + 5)
            };
            btnSave.Click += BtnSave_Click;

            this.Controls.Add(pictureBox);
            this.Controls.Add(btnSave);
        }
        private void SetupDragDrop()
        {

            pictureBox.DragEnter += Form_DragEnter;
            pictureBox.DragDrop += Form_DragDrop;
        }
        private void Form_DragEnter(object sender, DragEventArgs e)
        {
            // التحقق من نوع الملف المسحوب
            if (e.Data.GetDataPresent(DataFormats.FileDrop))
                e.Effect = DragDropEffects.Copy;
            else
                e.Effect = DragDropEffects.None;
        }
        private void Form_DragDrop(object sender, DragEventArgs e)
        {
            try
            {
                // اخذ الملفات المسحوبة
                string[] files =
(string[])e.Data.GetData(DataFormats.FileDrop);

                if (files.Length > 0)
                {
                    string imagePath = files[0];
                    // نشاط /يجب التحقق من مسار المف هل صورة...png,jpg,
                    if (File.Exists(imagePath))
                    {
                        pictureBox.Image = null; // تحرير الصورة السابقة
                        pictureBox.Image = Image.FromFile(imagePath);
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"خطأ في تحميل الصورة: {ex.Message}");
            }
        }
        private void BtnSave_Click(object sender, EventArgs e)
        {
            if (pictureBox.Image == null)
            {
                MessageBox.Show("لا توجد صورة للحفظ!");
                return;
            }

            using (SaveFileDialog sfd = new SaveFileDialog())
            {
                sfd.Filter = "ملفات الصور|*.jpg;*.png;*.bmp";
                if (sfd.ShowDialog() == DialogResult.OK)
                {
                    // حفظ الصورة باستخدام FileStream
                    using (FileStream fs = new FileStream(sfd.FileName,
FileMode.Create))
                    {
                        pictureBox.Image.Save(fs,
System.Drawing.Imaging.ImageFormat.Png);
                    }
                    MessageBox.Show("تم الحفظ بنجاح!");
                }
            }
        }

    }
    //-----------------
    [Description("صندوق نص مع تغيير لون عند تجاوز الحد")]
    [DefaultProperty("MaxLengthWarning")]
    [ToolboxItem(true)]
    public class LimitedTextBox : TextBox
    {
        // المتغيرات الداخلية
        private int _maxLengthWarning = 100;
        private Color _warningColor = Color.LightCoral;
        private Color _normalColor = Color.White;

        // السمات للخاصية
        [Category("السلوك")]
        [Description("الحد الأقصى للأحرف قبل تغيير اللون")]
        [DefaultValue(100)]
        [RefreshProperties(RefreshProperties.Repaint)]
        public int MaxLengthWarning
        {
            get => _maxLengthWarning;
            set
            {
                if (value < 0) value = 0;
                _maxLengthWarning = value;
                UpdateColor();
            }
        }

        [Category("المظهر")]
        [Description("لون الخلفية عند تجاوز الحد")]
        [DefaultValue(typeof(Color), "LightCoral")]
        public Color WarningColor
        {
            get => _warningColor;
            set
            {
                _warningColor = value;
                UpdateColor();
            }
        }

        [Category("المظهر")]
        [Description("لون الخلفية الطبيعي")]
        [DefaultValue(typeof(Color), "White")]
        public Color NormalColor
        {
            get => _normalColor;
            set
            {
                _normalColor = value;
                UpdateColor();
            }
        }

        // إخفاء الخاصية الأصلية إذا لزم
        [Browsable(false)]
        [EditorBrowsable(EditorBrowsableState.Never)]
        public new int MaxLength
        {
            get => base.MaxLength;
            set => base.MaxLength = value;
        }

        // Constructor
        public LimitedTextBox()
        {
            this.BackColor = _normalColor;
            this.Multiline = true;
            this.TextChanged += LimitedTextBox_TextChanged;
        }

        private void LimitedTextBox_TextChanged(object sender, EventArgs
e)
        {
            UpdateColor();
        }

        private void UpdateColor()
        {
            if (this.Text.Length > _maxLengthWarning)
            {
                this.BackColor = _warningColor;
            }
            else
            {
                this.BackColor = _normalColor;
            }
        }
        // Override لضمان التحديث عند التغيير البرمجي
        public override string Text
        {
            get => base.Text;
            set
            {
                base.Text = value;
                UpdateColor();
            }
        }
    }
}
```