

Ali Adhaghar

برمجة متقدمة

ثالث علوم حاسب وتقنية معلومات

المستوى	تألت علوم حاسب وتكنولوجيا معلومات	الموضوع
المادة	برمجة (متقدمة)	أمثلة متنوعة

هذا الصف تعديل لصف الـ [Form] لحساب جمع رقمين

```

class addForm:Form
{
    private Label Label1;
    private Label Label2;
    private Label Label3;
    private TextBox textBox1;
    private TextBox textBox2;
    private TextBox textBox3;
    private Button Button1;
    private Button Button2;

    public addForm()
    {
        InitializeComponent();//Start();
    }

    private void InitializeComponent();//Start()
    {
        this.Width = 290;
        this.Height = 140;
        this.Text = "جمع عددين";

        Label1=new Label();
        Label1.Size = new Size(70, 20);
        Label1.Location = new Point(10,10);
        Label1.Text = "العدد الأول";
        this.Controls.Add(Label1);

        textBox1 = new TextBox();
        textBox1.Size = new Size(70, 20);
        textBox1.Location = new Point(10, 30);
        textBox1.Text = "";
        this.Controls.Add(textBox1);

        Label2 = new Label();
        Label2.Size = new Size(70, 20);
        Label2.Location = new Point(85, 10);
        Label2.Text = "العدد الثاني";
        this.Controls.Add(Label2);

        textBox2 = new TextBox();
        textBox2.Size = new Size(70, 20);
        textBox2.Location = new Point(85, 30);
        textBox2.Text = "";
        this.Controls.Add(textBox2);

        Label3 = new Label();
        Label3.Size = new Size(70, 20);
        Label3.Location = new Point(160, 10);
        Label3.Text = "النتيجة";
        this.Controls.Add(Label3);
    }
}

```

```

textBox3 = new TextBox();
textBox3.Size = new Size(70, 20);
textBox3.Location = new Point(160, 30);
textBox3.Text = "";
this.Controls.Add(textBox3);

```

```

Button1 = new Button();
Button1.Size = new Size(60, 30);
Button1.Location = new Point(200, 60);
Button1.Text = "جمع";
Button1.Click += new EventHandler(sumButton); // Button1.Click += sumButton
this.Controls.Add(Button1);

```

```

Button2 = new Button();
Button2.Size = new Size(60, 30);
Button2.Location = new Point(10, 60);
Button2.Text = "إغلاق";
Button2.Click += new EventHandler(closeButton);
this.Controls.Add(Button2);

```

```

}
private void sumButton(object sender, EventArgs e)
{
    int x, y, z;
    int.TryParse(textBox1.Text, out x);
    int.TryParse(textBox2.Text, out y);
    z = x + y;
    textBox3.Text = z.ToString();
}

```

```

private void closeButton(object sender, EventArgs e)
{
    this.Close();
}
}

```

مثال للتعامل مع الـ [delegate]

```

namespace ConsoleApplication1

```

```

{
    delegate void show(); // صورة كثرية على شاشة مع نصي التوضيح
    delegate long getResult(int x);
    delegate void showMessage(string word);
    class Program
    {
        static void Main(string[] args)
        {
            printWelcome();
            show psh = printWelcome;
            psh(); // psh.Invoke();
            psh += printWelcome;
            psh(); // psh.Invoke();
            psh -= printWelcome;
            psh.Invoke();
        }
    }
}

```

```

show pw1 = new show s(printWelcome);
pw1();
beta(pw1);
// =====

show pw2 = ()=>{ Console.WriteLine("Welcome ....");};
pw2();
// =====

showMessage pw2 = print;
pw2 += Console.WriteLine;
pw2("AAAAAAAAAAAAAAAAAAAA");
pw2.Invoke("AAAAA");
// =====

// =====
showMessage pe;
pe = delegate(string word)
{
    Console.WriteLine(word);
};
pe("ppppppp");

showMessage pw3 = (x) => { Console.WriteLine(x); };

pw3("BBBBBB");
// =====

// =====
showMessage pw33 = (x) =>
{
    Console.WriteLine(x);
};

pw33("BBBBBB");
// =====

// =====
getResult gR1 = new getResult(fact);
long temp = gR1(4);
Console.WriteLine(temp.ToString());
// =====

// =====
getResult gR2 = delegate(int x) {
    long f = 1;
    for (int i = 1; i <= x; i++)
        f *= i;
    return f;
};

temp = gR2(4);
Console.WriteLine(temp.ToString());
// =====

// =====
Func<int, long, long> f1 = new Func<long, long>(fact);
Func<long, long> f2 = (x) => { long y = 1; for (int i = 1; i < x; ++i, y *= i); return y; };

```

```
Func<long, long> ff = new Func<long, long>(fact);

String outputStr = string.Format("Factorial : {0}", ff(5));
Console.WriteLine(outputStr);
Action<string> Aprint = Console.WriteLine;
Aprint("wwwwwww");
Aprint(outputStr);
```

```
Predicate<int> p = new Predicate<int>(checkodd);
if (p(9)) Console.WriteLine("odd");
Console.ReadKey();
```

```
}
// =====
```

```
static bool checkodd(int x)
{
    return x % 2 != 0;
}
// =====
```

```
static void printWelcome()
{
    Console.WriteLine("Welcome ....");
}
// =====
```

```
static void print(string word)
{
    Console.WriteLine(word);
}
// =====
```

```
static long fact(int x)
{
    return (x <= 0) ? 1 : x * fact(x-1);
}
// =====
```

```
static void beta(show h)
{
    h( );
}
}
```

```
}
```

مثال للتعامل مع الـ [List]

```
namespace ConsoleApplication1
{
    delegate bool check(int x);
    class Program
    {
        static void Main(string[] args)
        {
            int[] p = new int[] {10,30,40,1,7,200};
            List<int> L1 = new List<int>( );
            L1.Add(10);
            L1.Add(200);
            L1.Add(300);
            L1.Add(40);
```



```

L1.Add(500);
L1.AddRange(p);
check lp = new check((x) => { return x % 2 == 0; });
int pp = 0;
foreach (int x in L1)
{
    Console.WriteLine(x.ToString()); طابع العنصر
    if (lp(x))
        pp++;
}
Console.WriteLine("===== " + pp.ToString() + " =====");

```

```

20/6/20 نوع L1.Remove(200);
printList(L1);
int ctt = L1.Count((x) => x % 2 == 0);
Console.WriteLine("===== " + ctt.ToString() + " =====");
Console.WriteLine(count(L1, (x) => x > 200).ToString());

Console.WriteLine("=====");
int ct = L1.Count((x) => x > 100);
Console.WriteLine(ct.ToString());

L1.ForEach(Console.WriteLine); طابع العنصر
Console.WriteLine(L1.IndexOf(300).ToString()); موقع العنصر

Console.WriteLine("=====");
L1.Where((a, i) => (a % 20 == 0) && (i < 3)).ToList<int>().ForEach(Console.WriteLine);
Console.WriteLine(L1.Aggregate((s, x) => s = s + x * x).ToString());
List<int> L2 = L1.OfType<int>().ToList<int>(); // Interface
}
static void printList(List<int> L)
{
    foreach (int x in L)
        Console.WriteLine(x.ToString());
}
static int count(int[] L, check ch)
{
    int c = 0;
    foreach (int x in L)
        if (ch(x)) c++;
    return c;
}
}

```

مثال (1) للتعامل مع مصفوفة أهداف

```
public class Form1 : Form
```

```
{
```

```
    TextBox[] x;
```

```
    int count = 0;
```

```
    public Form1()
```

```
    {
```

```
        InitializeComponent();
```

```
    }
```

```
// ===== creat 1st Ad2020
```

```
// creat button click event
```

```
private void button1_Click(object sender, EventArgs e)
```

```
{
```

```
    //if (x != null)
```

```
    //{
```

```
    //    for (int i = 0; i < count; i++)
```

```
    //    {
```

```
    //        if(x[i]!=null)
```

```
    //            { x[i].Dispose();
```

```
    //                x[i]=null; }
```

```
    //}
```

```
    count = Convert.ToInt32(textBox1.Text);
```

```
    x=new TextBox[Convert.ToInt32(textBox1.Text)];
```

```
    int newTop = 10,
```

```
        StepTop = 30;
```

```
    Random rd = new Random();
```

```
    for(int i=0;i<x.Length;i++)
```

```
    {
```

```
        x[i] = new TextBox();
```

```
        x[i].Top = newTop;
```

```
        x[i].Left = 20;
```

```
        x[i].Text = rd.Next(0, 100).ToString();
```

```
        this.Controls.Add(x[i]);
```

```
        newTop += StepTop;
```

```
    }
```

```
}
```

```
private void button2_Click(object sender, EventArgs e)
```

```
{
```

```
    for (int i = 0; i < x.Length; i++)
```

```
        if (Convert.ToInt32(x[i].Text) % 2 == 1)
```

```
            x[i].BackColor = Color.LightGreen;
```

```
        // else
```

```
        // x[i].BackColor = Color.White;
```

```
}
```

```
private void button3_Click(object sender, EventArgs e)
```

```
{
```

```
    int sum = 0;
```

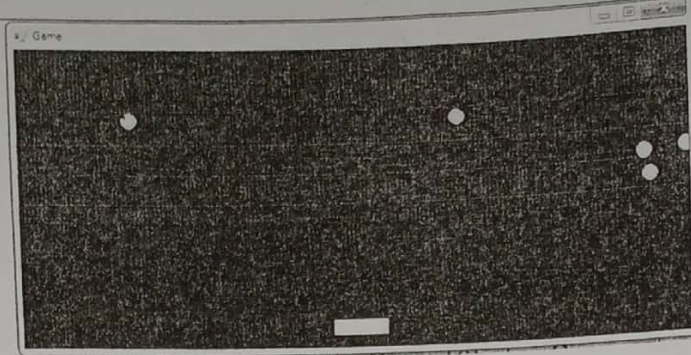
```
    for (int i = 0; i < x.Length; i++)
```

```
        sum+=Convert.ToInt32(x[i].Text);
```

```
    textBox2.Text = sum.ToString();
```

```
}
```

```
}
```



```

public class Form1 : Form
{
    private PictureBox[] Shape2 = new PictureBox[5];
    private Timer timer1 = new Timer();
    public Form1( )
    {
        InitializeComponent();
        this.KeyPreview = true;
        this.KeyDown += new KeyEventHandler(Form5_KeyDown);
        Random rnd = new Random();
        for (int i = 0; i < 5; ++i)
        {
            Shape2[i] = new PictureBox();
            Shape2[i].Image = new Bitmap("e:\\game.png");
            Shape2[i].SizeMode = PictureBoxSizeMode.AutoSize;
            Shape2[i].Left = (int) (rnd.NextDouble() * this.Width);
            Shape2[i].Top = (int) (rnd.NextDouble() * this.Height / 4);
        }
        this.Controls.AddRange(Shape2);
        timer1.Interval = 200;
        timer1.Enabled = true;
        timer1.Tick += new EventHandler(timer1_Tick);
    }
    private void Form1_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.A)
        {
            button1.Left -= 20;
        }
        else if (e.KeyCode == Keys.S)
        {
            button1.Left += 20;
        }
        else if (e.KeyCode == Keys.D)
        {
            Random rnd = new Random();
            for (int i = 0; i < 5; ++i)
            {
                Shape2[i].Left = (int) (rnd.NextDouble() * this.Width);
                Shape2[i].Top = (int) (rnd.NextDouble() * this.Height / 4);
            }
        }
    }
}

```



```

private void timer1_Tick(object sender, EventArgs e)
{
    Random rnd = new Random();
    for (int i = 0; i < 5; i++)
    {
        Shape2[i].Top = Shape2[i].Top + 20;
        if ((Shape2[i].Top + Shape2[i].Height + 2 >= button1.Top) &&
            (Shape2[i].Left >= button1.Left) &&
            (Shape2[i].Left + Shape2[i].Width <= button1.Left + button1.Width))
        {
            Shape2[i].Left = (int)(rnd.NextDouble() * this.Width);
            Shape2[i].Top = (int)(rnd.NextDouble() * this.Height / 4);
        }
    }
}

```

هذا الصف تعديل لصف مربع النص لأدخال الأعداد فقط

```

class NumText : TextBox
{
    public NumText()
    {
        start( ); // InitializeComponent()
    }
    void start( ) // InitializeComponent()
    {
        this.BackColor = Color.Blue;
        this.ForeColor = Color.Yellow;
        this.KeyPress += new KeyPressEventHandler(keyp);
    }
    private void keyp(object sender, KeyPressEventArgs e)
    {
        if (!(((e.KeyChar >= 48) && (e.KeyChar <= 57)) || (e.KeyChar == 8)))
        {
            e.KeyChar = Convert.ToChar(0); // e.Handled = true;
        }
    }
}

```

هذا الصف تعديل للصف [Panel] ويحتوي على زر الغرض منه توضيح تغيير الأبعاد وكذلك إدارة الأحداث والخصائص

```

class PanButton : Panel
{
    private Button button1;
    public String TText
    {
        get
        { return button1.Text; }
        set
        { button1.Text = value; }
    }
    // =====
    public PanButton()
    {
        Start( ); // InitializeComponent();
        this.Resize += delegate(object sender, EventArgs e)

```

```

{
    button1.Left = 0;
    button1.Top = 0;
    button1.Width = this.Width;
    button1.Height = this.Height;
};
button1.Click += delegate(object sender, EventArgs e)
{
    this.OnClick(e);
};
button1.Left = 0;
button1.Top = 0;
button1.Width = this.Width;
button1.Height = this.Height;
button1.Text = "Ok";
}
// =====
private void Start( ) //InitializeComponent( )
{
    button1 = new Button( );
    // button1
    button1.Location = new Point(0, 0);
    button1.Name = "button1";
    button1.Size = new Size(200, 60);
    button1.TabIndex = 0;
    button1.Text = "Ok";
    // PanButton
    this.Controls.Add(this.button1);
    this.Size = new Size(200, 60);
}
}
}

```

هذا الصف تعديل للصف [Panel] ويحتوي على أزرار ومربعات نص لحساب عملية الجمع

```

Class SumClass : Panel
{
    private Label label1;
    private Label label2;
    private Label label3;
    private TextBox textBox1;
    private TextBox textBox2;
    private TextBox textBox3;
    private Button button1;
    private Button button2;

    public SumClass( )
    {
        InitializeComponent();
    }

    private void InitializeComponent()
    {
        label1 = new Label();
        label2 = new Label();
        label3 = new Label();
        textBox1 = new TextBox();

```

```

textBox2 = new TextBox();
textBox3 = new TextBox();
button1 = new Button();
button2 = new Button();
//
// label1
//
label1.Location = new Point(312, 32);
label1.Name = "label1";
label1.RightToLeft = RightToLeft.Yes;
label1.Size = new Size(62, 13);
label1.TabIndex = 13;
label1.Text = "العدد الأول:";

// label2
//
label2.AutoSize = true;
label2.Location = new Point(306, 64);
label2.Name = "label2";
label2.RightToLeft = RightToLeft.Yes;
label2.Size = new Size(67, 13);
label2.TabIndex = 14;
label2.Text = "العدد الثاني:";
//
// label3
//
label3.AutoSize = true;
label3.Location = new Point(328, 101);
label3.Name = "label3";
label3.RightToLeft = RightToLeft.Yes;
label3.Size = new Size(47, 13);
label3.TabIndex = 15;
label3.Text = "النتيجة:";
//
// textBox1
//
textBox1.Location = new Point(201, 29);
textBox1.Name = "textBox1";
textBox1.Size = new Size(86, 20);
textBox1.TabIndex = 10;
//
// textBox2
//
textBox2.Location = new Point(201, 62);
textBox2.Name = "textBox2";
textBox2.Size = new Size(86, 20);
textBox2.TabIndex = 11;
//
// textBox3
//
textBox3.Location = new Point(201, 101);
textBox3.Name = "textBox3";
textBox3.Size = new Size(86, 20);
textBox3.TabIndex = 12;

```

```

//
// button1
//
button1.Location = new Point(309, 144);
button1.Name = "button1";
button1.Size = new Size(74, 28);
button1.TabIndex = 8;
button1.Text = "جمع";
button1.Click += new EventHandler(this.SumClick);
//
// button2
//
button2.Location = new Point(12, 144);
button2.Name = "button2";
button2.Size = new Size(66, 28);
button2.TabIndex = 9;
button2.Text = "خروج";
//
// SumClass
//
this.ClientSize = new Size(397, 184); // Width and Height
this.Controls.Add(label3);
this.Controls.Add(label2);
this.Controls.Add(label1);
this.Controls.Add(textBox3);
this.Controls.Add(textBox2);
this.Controls.Add(textBox1);
this.Controls.Add(button2);
this.Controls.Add(button1);
this.Name = "SumClass";
}

private void SumClick(object sender, EventArgs e)
{
    int x = Convert.ToInt32(textBox1.Text);
    int y = Convert.ToInt32(textBox2.Text);
    int z = x + y;
    textBox3.Text = z.ToString();
}
}
}

```

مثال رقم ١ لتوضيح رسم زر يعتمد على خطوط مستقيمة

```

class mButon00 : UserControl
{
    public mButon00()
    {
        InitializeComponent(); // Start();
    }

    private void InitializeComponent() //Start()
    {
        this.Size = new Size(117, 35);
        this.Paint += new PaintEventHandler(this.mb_Paint);
    }
}

```



```

    }

private void mb_Paint(object sender, PaintEventArgs e)
{
    Pen p1 = new Pen(Color.White, 2);
    Pen p2 = new Pen(Color.Black, 2);
    e.Graphics.DrawLine(p1, 0, 0, this.Width, 0);
    e.Graphics.DrawLine(p1, 0, 0, 0, this.Height);
    e.Graphics.DrawLine(p2, 0, this.Height, this.Width, this.Height);
    e.Graphics.DrawLine(p2, this.Width, 0, this.Width, this.Height);
}
}
}

```

مثال رقم ٢ لتوضيح بناء زر يالاعتماد على خطوط مستقيمة

```

class mButon01 : UserControl
{
    public mButon01()
    {
        InitializeComponent();
    }

    private void InitializeComponent()
    {
        this.Size = new Size(117, 35);
        this.Paint += new PaintEventHandler(this.mb_Paint);
        this.MouseDown += new MouseEventHandler(this.mb_MouseDown);
        this.MouseUp += new MouseEventHandler(this.mb_MouseUp);
    }

    private void mb_Paint(object sender, PaintEventArgs e)
    {
        Pen p1 = new Pen(Color.White, 2);
        Pen p2 = new Pen(Color.Black, 2);
        e.Graphics.DrawLine(p1, 0, 0, this.Width, 0);
        e.Graphics.DrawLine(p1, 0, 0, 0, this.Height);
        e.Graphics.DrawLine(p2, 0, this.Height, this.Width, this.Height);
        e.Graphics.DrawLine(p2, this.Width, 0, this.Width, this.Height);
    }

    private void mb_MouseDown(object sender, MouseEventArgs e)
    {
        Pen p1 = new Pen(Color.Black, 2);
        Pen p2 = new Pen(Color.White, 2);
        this.CreateGraphics().DrawLine(p1, 0, 0, this.Width, 0);
        this.CreateGraphics().DrawLine(p1, 0, 0, 0, this.Height);
        this.CreateGraphics().DrawLine(p2, 0, this.Height, this.Width, this.Height);
        this.CreateGraphics().DrawLine(p2, this.Width, 0, this.Width, this.Height);
    }

    private void mb_MouseUp(object sender, MouseEventArgs e)
    {
        Pen p1 = new Pen(Color.White, 2);
        Pen p2 = new Pen(Color.Black, 2);
        this.CreateGraphics().DrawLine(p1, 0, 0, this.Width, 0);
    }
}

```

```

this.CreateGraphics().DrawLine(p1, 0, 0, 0, this.Height);
this.CreateGraphics().DrawLine(p2, 0, this.Height, this.Width, this.Height);
this.CreateGraphics().DrawLine(p2, this.Width, 0, this.Width, this.Height);
}
}
}

```

مثال رقم ٣ لتوضيح بناء زر يالاعتماد على خطوط مستقيمة مع إضافة خاصية العنوان

```

class mButon02 : UserControl
{
    private Label label1;
    public String text
    {
        get
        { return label1.Text; }
        set
        { label1.Text = value; }
    }
    // =====
    public mButon02()
    {
        InitializeComponent();
        label1.Left = 5;
        label1.Top = 5;
        label1.Width = this.Width - 10;
        label1.Height = this.Height - 10;
        // =====
        this.Resize += delegate(object sender, EventArgs e)
        {
            label1.Left = 5;
            label1.Top = 5;
            label1.Width = this.Width-10;
            label1.Height = this.Height-10;
        };
        label1.Click += delegate(object sender, EventArgs e)
        {
            this.OnClick(e);
        };
    }
    // =====
    private void InitializeComponent()
    {
        label1 = new Label();
        //
        // label1
        //
        label1.Location = new Point(3, 9);
        label1.Name = "label1";
        label1.Size = new Size(42, 17);
        label1.TabIndex = 0;
        label1.Text = "label1";
        label1.TextAlign = ContentAlignment.MiddleCenter;
        label1.MouseDown += new MouseEventHandler(this.label1_MouseDown);
        label1.MouseUp += new MouseEventHandler(this.label1_MouseUp);
    }
}

```

```

//
// mButon02 ( UserControl : this class )
//
this.Controls.Add(this.label1);
this.Name = "mButon02";
this.Size = new Size(117, 35);
this.Paint += new PaintEventHandler(this.mb_Paint);
this.MouseDown += new MouseEventHandler(this.mb_MouseDown);
this.MouseUp += new MouseEventHandler(this.mb_MouseUp);
}
// =====
private void mb_Paint(object sender, PaintEventArgs e)
{
    Pen p1 = new Pen(Color.White, 2);
    Pen p2 = new Pen(Color.Black, 2);
    e.Graphics.DrawLine(p1, 0, 0, this.Width, 0);
    e.Graphics.DrawLine(p1, 0, 0, 0, this.Height);
    e.Graphics.DrawLine(p2, 0, this.Height, this.Width, this.Height);
    e.Graphics.DrawLine(p2, this.Width, 0, this.Width, this.Height);
}
// =====
private void mb_MouseDown(object sender, MouseEventArgs e)
{
    Pen p1 = new Pen(Color.Black, 2);
    Pen p2 = new Pen(Color.White, 2);
    this.CreateGraphics().DrawLine(p1, 0, 0, this.Width, 0);
    this.CreateGraphics().DrawLine(p1, 0, 0, 0, this.Height);
    this.CreateGraphics().DrawLine(p2, 0, this.Height, this.Width, this.Height);
    this.CreateGraphics().DrawLine(p2, this.Width, 0, this.Width, this.Height);
}
// =====
private void mb_MouseUp(object sender, MouseEventArgs e)
{
    Pen p1 = new Pen(Color.White, 2);
    Pen p2 = new Pen(Color.Black, 2);
    this.CreateGraphics().DrawLine(p1, 0, 0, this.Width, 0);
    this.CreateGraphics().DrawLine(p1, 0, 0, 0, this.Height);
    this.CreateGraphics().DrawLine(p2, 0, this.Height, this.Width, this.Height);
    this.CreateGraphics().DrawLine(p2, this.Width, 0, this.Width, this.Height);
}
// =====
private void label1_MouseDown(object sender, MouseEventArgs e)
{
    this.OnMouseDown(e);
}
// =====
// =====
private void label1_MouseUp(object sender, MouseEventArgs e)
{
    this.OnMouseUp(e);
}
// =====
}
}

```

```

public class Shape : Control // UserControl Or Panel
{
    public Shape( )
    {
        this.Paint +=
            new PaintEventHandler(Shape_Paint );
    }

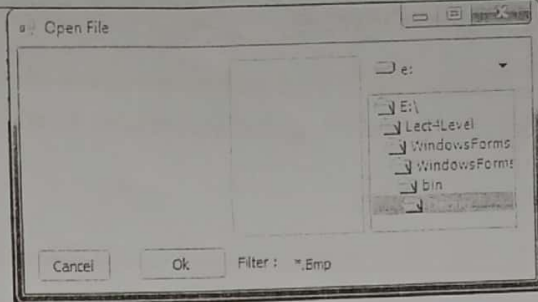
    public enum ShapeType
    {
        Circular = 0,
        Rectangular = 1,
    }

    private ShapeType _shape = ShapeType.Rectangular;

    [Description("Determines the type of the shape : rectangle or circle"),
    Category("Appearance"),
    DefaultValue(typeof(ShapeType), "Rectangular"),
   Browsable(true)]
    public ShapeType shape
    {
        get { return _shape; }
        set
        {
            if (_shape == value)
                return;
            _shape = value;
            Invalidate( );
        }
    }

    private void Shape_Paint(object sender, PaintEventArgs e)
    {
        Graphics gfx = e.Graphics;
        Rectangle rec = this.ClientRectangle;
        rec.Width -= 1;
        rec.Height -= 1;
        switch (_shape)
        {
            case ShapeType.Rectangular:
                gfx.DrawRectangle(new Pen(Color.Black, 1), rec);
                break;
            case ShapeType.Circular:
                gfx.DrawEllipse(new Pen(Color.Black, 1), rec);
                break;
        }
    }
}

```

```

public partial class Form18 : Form
{
    public string fileName;
    // =====
    public Form18()
    {
        InitializeComponent();
        fileListBox1.MouseDown += new MouseEventHandler(FmouseDown);
        panel1.DragDrop += new DragEventHandler(PDragDrop);
        panel1.DragEnter += new DragEventHandler(PDragEnter);
    }
    // =====
    private void Form18_Load(object sender, EventArgs e)
    {
        fileListBox1.Pattern = textBox1.Text.Trim();
        //Environment.CurrentDirectory = "c:\\";
        //dirListBox1.Path = Environment.CurrentDirectory;
    }
    // =====
    private void FmouseDown(object sender, MouseEventArgs e)
    {
        if (e.Button == MouseButtons.Left)
        {
            string str = null;
            str += fileListBox1.Path.ToString().Trim();
            str += "\\";
            str += fileListBox1.FileName.ToString().Trim();
            this.DoDragDrop(str, DragDropEffects.All);
        }
    }
    // =====
    private void PDragDrop(object sender, DragEventArgs e)
    {
        string str = e.Data.GetData(DataFormats.Text).ToString();
        Bitmap b = new Bitmap(str);
        pictureBox1.Image = b;
    }
    // =====
    private void PDragEnter(object sender, DragEventArgs e)
    {
        e.Effect = DragDropEffects.All;
    }
    // =====
}

```

```

private void driveListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    dirListBox1.Path = driveListBox1.Drive;
}
// =====
private void dirListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    fileListBox1.Path = dirListBox1.Path;
}
// =====

// =====
private void fileListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    //Bitmap b=new Bitmap(fileListBox1.Path.ToString()+"\\"+fileListBox1.FileName.ToString());
    //pictureBox1.Image = b;

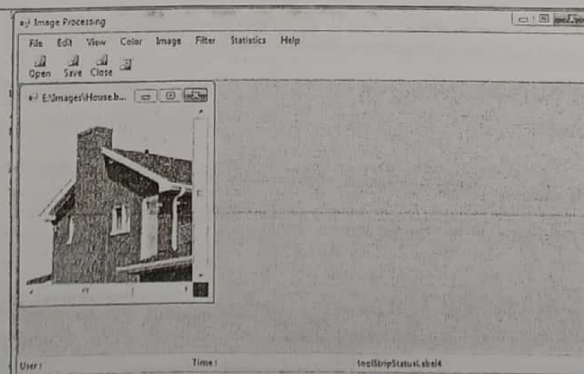
    //pictureBox1.Width = b.Width;
    //pictureBox1.Height = b.Height;
    //Graphics g = pictureBox1.CreateGraphics();
    //g.DrawImage(b,new Point(0,0));
}
// =====

// =====
private void button1_Click(object sender, EventArgs e)
{
    string str = fileListBox1.Path.ToString().Trim()+"\\";
    str += fileListBox1.FileName.ToString().Trim();
    fileName = str;
    this.Close();
}
// =====

// =====
private void button2_Click(object sender, EventArgs e)
{
    fileName = null;
    this.Close();
}
}

```

بداية مشروع بسيط لمعالجة الصور



```
namespace WindowsFormsApplication4
```

```
{  
    public partial class Form1 : Form  
    {  
        // =====  
        public Form1()  
        {  
            InitializeComponent();  
        }  
        // =====  
  
        // =====  
        private void openToolStripMenuItem_Click(object sender, EventArgs e)  
        {  
            OpenFileDialog fileOpen = new OpenFileDialog();  
            if (fileOpen.ShowDialog(this) == DialogResult.OK)  
            {  
                Form6 newfile = new Form6();  
                try  
                {  
                    newfile.pictureBox1.Image = new Bitmap(fileOpen.FileName);  
                    newfile.MdiParent = this;  
                    newfile.Location = new Point(0, 0);  
                    newfile.Text = fileOpen.FileName;  
                    newfile.Show();  
                }  
                catch (Exception ex)  
                {  
                    newfile.Dispose();  
                    MessageBox.Show(ex.Message);  
                }  
            }  
        }  
        // =====  
  
        // =====  
        private void toolStripButton1_Click(object sender, EventArgs e)  
        {  
            openToolStripMenuItem_Click(sender, e);  
        }  
        // =====  
  
        // =====  
        // Begin save menu item  
        private void saveToolStripMenuItem_Click(object sender, EventArgs e)  
        {  
            ((Form6)this.ActiveMdiChild).pictureBox1.Image.Save(this.ActiveMdiChild.Text);  
        }  
        // End save menu item  
        // =====  
  
        // =====  
        // Begin save As menu item  
        private void saveAsToolStripMenuItem_Click(object sender, EventArgs e)
```

```

{
    SaveFileDialog saveFileDialog = new SaveFileDialog();
    saveFileDialog.InitialDirectory = Environment.GetFolderPath(
        Environment.SpecialFolder.MyPictures);
    saveFileDialog.Filter = "Picture Files (*.Bmp)|*.Bmp|All Files (*.*)|*.*";
    if (saveFileDialog.ShowDialog(this) == DialogResult.OK)
    {
        string FileName = saveFileDialog.FileName;
        ((Form6)this.ActiveMdiChild).pictureBox1.Image.Save(FileName);
    }
}
// End save As menu item
// =====

// =====
// Begin Print Dialog
private void printToolStripMenuItem_Click(object sender, EventArgs e)
{
    printDialog1.ShowDialog(this);
}
// End Print Dialog
// =====

// =====
// Begin Print Page Setup
private void pageToolStripMenuItem_Click(object sender, EventArgs e)
{
    //pageSetupDialog1.ShowDialog(this);
}
// =====

// =====
private void closeToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.ActiveMdiChild.Close();
}
// =====

// =====
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    foreach (Form f in this.MdiChildren)
        f.Close();
}
// =====

// =====
private void exitToolStripMenuItem1_Click(object sender, EventArgs e)
{
    Application.Exit();
}
// =====

// =====

```


// Lighting menu Item

private void lightingToolStripMenuItem_Click(object sender, EventArgs e)

```
{
    float InRatio = 0.10F;
    Bitmap Temp=new Bitmap(((Form6)this.ActiveMdiChild).pictureBox1.Image);
    Prograss1.Visible = true;
    Prograss1.Maximum = Temp.Height;
    Prograss1.Minimum = 0;
    Prograss1.Value = 0;
    for (int y = 0; y < Temp.Height ; y++)
    {
        for (int x = 0; x < Temp.Width; x++)
        {
            Color oldColor = Temp.GetPixel(x, y);
            int cR = oldColor.R + (int)((float)(255 - oldColor.R) * InRatio);
            int cG = oldColor.G + (int)((float)(255 - oldColor.G) * InRatio);
            int cB = oldColor.B + (int)((float)(255 - oldColor.B) * InRatio);
            Color newColor = Color.FromArgb(cR, cG, cB);
            Temp.SetPixel(x, y, newColor);
        }
        Prograss1.Value += 1;
    }
    Prograss1.Visible = false;
    ((Form6)this.ActiveMdiChild).pictureBox1.Image = Temp;
}
```

// End Lighting Button

//

//

// Darking menu Item

private void darkingToolStripMenuItem_Click(object sender, EventArgs e)

```
{
    float InRatio = 0.10F;
    Bitmap Temp = new Bitmap(((Form6)this.ActiveMdiChild).pictureBox1.Image);
    Prograss1.Maximum = Temp.Height;
    Prograss1.Minimum = 0;
    Prograss1.Value = 0;
    Prograss1.Visible = true;
    for (int y = 0; y < Temp.Height; y++)
    {
        for (int x = 0; x < Temp.Width; x++)
        {
            Color oldColor = Temp.GetPixel(x, y);
            int cR = oldColor.R - (int)((float)(255 - oldColor.R) * InRatio);
            int cG = oldColor.G - (int)((float)(255 - oldColor.G) * InRatio);
            int cB = oldColor.B - (int)((float)(255 - oldColor.B) * InRatio);
            Color newColor = Color.FromArgb(cR, cG, cB);
            Temp.SetPixel(x, y, newColor);
        }
        Prograss1.Value += 1;
    }
    Prograss1.Visible = false;
    ((Form6)this.ActiveMdiChild).pictureBox1.Image = Temp;
}
```

// End Darking Button

```
//
// Begin Gray menu Item
private void grayToolStripMenuItem_Click(object sender, EventArgs e)
{
    Bitmap Temp = new Bitmap(((Form6)this.ActiveMdiChild).pictureBox1.Image);
    Prograss1.Maximum = Temp.Height;
    Prograss1.Minimum = 0;
    Prograss1.Value = 0;
    Prograss1.Visible = true;
    for (int y = 0; y < Temp.Height; y++)
    {
        for (int x = 0; x < Temp.Width; x++)
        {
            Color oldColor = Temp.GetPixel(x, y);
            int gray = (oldColor.R + oldColor.G + oldColor.B) / 3;
            Color newColor = Color.FromArgb(gray, gray, gray);
            Temp.SetPixel(x, y, newColor);
        }
        Prograss1.Value += 1;
    }
    Prograss1.Visible = false;
    ((Form6)this.ActiveMdiChild).pictureBox1.Image = Temp;
}
// End Gray Button
// =====
```

```
// =====
// Begin Negative menu item
private void negaiveToolStripMenuItem_Click(object sender, EventArgs e)
{
    Bitmap Temp = new Bitmap(((Form6)this.ActiveMdiChild).pictureBox1.Image);
    Prograss1.Maximum = Temp.Height;
    Prograss1.Minimum = 0;
    Prograss1.Value = 0;
    Prograss1.Visible = true;
    for (int y = 0; y < Temp.Height; y++)
    {
        for (int x = 0; x < Temp.Width; x++)
        {
            Color oldColor = Temp.GetPixel(x, y);
            int gray = (oldColor.R + oldColor.G + oldColor.B) / 3;
            int negative = 255 - gray;
            Color newColor = Color.FromArgb(negative, negative, negative);
            Temp.SetPixel(x, y, newColor);
        }
        Prograss1.Value += 1;
    }
    Prograss1.Visible = false;
    ((Form6)this.ActiveMdiChild).pictureBox1.Image = Temp;
}
// End Negative Button
// =====
// =====
```

```
// Begin Drag Drop
private void Form1_DragDrop(object sender, DragEventArgs e)
{
    string[] FileList = (string[] ) e.Data.GetData(DataFormats.FileDrop);// as string[];
    foreach (string File in FileList)
    {
        Form6 newfile = new Form6();
        newfile.pictureBox1.Image =
        new Bitmap(File);
        newfile.MdiParent = this;
        newfile.Location = new Point(0, 0);
        newfile.Text = File;
        newfile.Show();
    }
}
```

```
//End Drag Drop
```

```
//=====
```

```
// Begin Darg Enter Event
```

```
private void Form1_DragEnter(object sender, DragEventArgs e)
{
    if (e.Data.GetDataPresent(DataFormats.FileDrop))
        e.Effect = DragDropEffects.All;
}
```

```
private void Form1_Load(object sender, EventArgs e)
```

```
{
    if ((Program.Filelist!=null)&&(Program.Filelist.Length > 0))
    {
        foreach (string File in Program.Filelist)
        {
            Form6 newfile = new Form6();
            // فتح ملف
            newfile.pictureBox1.Image = new Bitmap(File);
            newfile.MdiParent = this;
            newfile.Location = new Point(0, 0);
            newfile.Text = File;
            newfile.Show();
        }
    }
}
```

```
Form editForm;
```

```
private void copyToolStripMenuItem_Click(object sender, EventArgs e)
```

```
{
    editForm = this.ActiveMdiChild;
    //Clipboard.SetData( );
}
```

```
private void pastToolStripMenuItem_Click(object sender, EventArgs e)
```

```
{
    ((Form6)this.ActiveMdiChild).getCopy(editForm);
}
```

```
//=====
```

```
}
```

```
}
```

```

public partial class Form6 : Form
{
    // =====
    public Rectangle clip=new Rectangle(1,1,1,1);
    public Point startPoint=new Point(1,1);
    public bool downFlag = false;
    public bool visibleFlag = false;
    // =====
    public Form6()
    {
        InitializeComponent();
        this.Load += new EventHandler(Form6_Load);
        this.Resize += new EventHandler(Form6_Resize);
        pictureBox1.MouseDown+=new MouseEventHandler(pictureBox1_MouseDown);
        pictureBox1.MouseMove+=new MouseEventHandler(pictureBox1_MouseMove);
        pictureBox1.MouseUp+=new MouseEventHandler(pictureBox1_MouseUp);
        label1.BackColor = Color.Transparent;
        label1.Parent = pictureBox1;
    }
    // =====

    // =====
    private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
    {
        downFlag = true;
        clip = new Rectangle(e.X, e.Y, 1, 1);
        startPoint = new Point(e.X, e.Y);
        label1.Visible = true;
        label1.Location = new Point(clip.Left, clip.Top);
        label1.Size = new Size(clip.Width, clip.Height);
    }
    // =====

    // =====
    private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
    {
        if (downFlag)
        {
            clip = new Rectangle((e.X < clip.Left) ? e.X : clip.Left,
                                (e.Y < clip.Top) ? e.Y : clip.Top,
                                Math.Abs(startPoint.X - e.X),
                                Math.Abs(startPoint.Y - e.Y));
            label1.Location = new Point(clip.Left, clip.Top);
            label1.Size = new Size(clip.Width, clip.Height);
            visibleFlag = true;
        }
    }
    // =====
    private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
    {
        downFlag = false;
    }
}

```



```

// =====
private void pictureBox1_MouseClick(object sender, MouseEventArgs e)
{
    label1.Visible = visibleFlag;
    visibleFlag = false;
}
// =====

// =====
private void Form6_Load(object sender, EventArgs e)
{
    panel1.Location = new Point(1, 1);
    pictureBox1.Location =
    new Point(1, 1);
    // panel1.Size =
    // new Size(this.ClientRectangle.Width, this.ClientRectangle.Height);
    this.OnResize(e);
}
// =====

// =====
private void Form6_Resize(object sender, EventArgs e)
{
    panel1.Width = this.ClientRectangle.Width;
    panel1.Height = this.ClientRectangle.Height;
    vScrollBar1.Left = panel1.Width - vScrollBar1.Width;
    vScrollBar1.Top = panel1.Top;
    vScrollBar1.Height = panel1.Height - hScrollBar1.Height;
    hScrollBar1.Top = panel1.Height - hScrollBar1.Height;
    hScrollBar1.Left = panel1.Left;
    hScrollBar1.Width = panel1.Width - vScrollBar1.Width;
    if (pictureBox1.Width > panel1.Width)
    {
        hScrollBar1.Visible = true;
        hScrollBar1.Maximum =
        pictureBox1.Image.Width -
        panel1.Width;
    }
    else
    {
        hScrollBar1.Visible = false;
    }
    // =====
    if (pictureBox1.Height > panel1.Height)
    {
        vScrollBar1.Visible = true;
        vScrollBar1.Maximum =
        pictureBox1.Image.Height - panel1.Height;
    }
    else
    {
        vScrollBar1.Visible = false;
    }
}

```

```

// =====
private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)
{
    pictureBox1.Top = -vScrollBar1.Value;
}
// =====
// =====
private void hScrollBar1_Scroll(object sender, ScrollEventArgs e)
{
    pictureBox1.Left = -hScrollBar1.Value;
}
// =====

// =====
public void getCopy(Form scrForm)
{
    if (scrForm == null) return;
    PictureBox scrPicture
        =(PictureBox)((Panel)scrForm.Controls["panel1"]).Controls["pictureBox1"];
    Label scrLabel = (Label)scrPicture.Controls["label1"];
    if (!scrLabel.Visible) return ;
    Graphics clp = this.pictureBox1.CreateGraphics();
    clp.DrawImage(scrPicture.Image,
        new Rectangle(1, 1, scrLabel.Width, scrLabel.Height),
        new Rectangle(scrLabel.Left, scrLabel.Top, scrLabel.Width, scrLabel.Height),
        GraphicsUnit.Pixel);
}
// =====
}
}
// =====
// =====
// =====
// =====
static class Program
{
    public static bool mdown = false ;
    public static int Mox = 0;
    public static int Moy = 0;
    public static int mtop = 0;
    public static int mleft = 0;
    public static int mwidth = 0;
    public static int mheight = 10;
    public static string [ ] Filelist;
    [STAThread]
    static void Main(string[] args)
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Program.Filelist = args;
        Application.Run(new Form1());
    }
}
}

```

Class FileClass

```
{
    static public void WriteToFile1()
    {
        StreamWriter sw = new StreamWriter("data.txt");
        sw.WriteLine("Welcome ...");
        sw.WriteLine("This is my first file program");
        sw.WriteLine("Good bye...");
        sw.Close();
    }
    static public void ReadFromFile1()
    {
        StreamReader sr = new StreamReader("data.txt");
        string tmp;
        tmp = sr.ReadLine();
        Console.WriteLine(tmp);
        tmp = sr.ReadLine();
        Console.WriteLine(tmp);
        tmp = sr.ReadLine();
        Console.WriteLine(tmp);
        sr.Close();
    }

    static public void WriteToFile2()
    {
        StreamWriter sw = new StreamWriter("data.txt",true);
        string tmp;
        while ((tmp = Console.ReadLine().Trim()) != "")
            sw.WriteLine(tmp);
        sw.Close();
    }

    static public void ReadFromFile2()
    {
        StreamReader sr = new StreamReader("data.txt");
        string tmp;
        while((tmp = sr.ReadLine())!=null)
            Console.WriteLine(tmp);
        sr.Close();
    }
}
```