



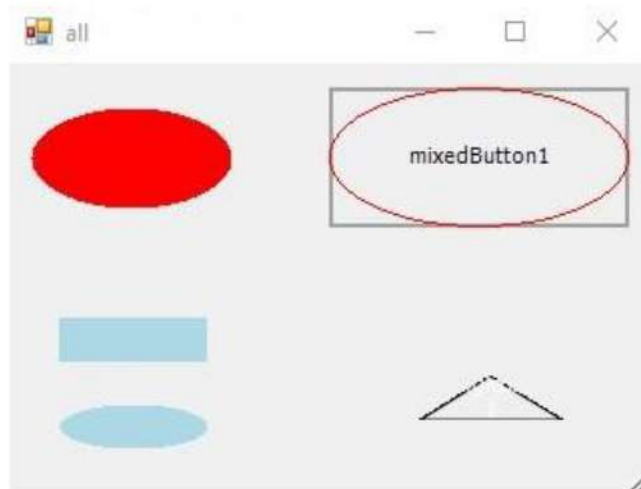
# برمجة متقدمة C# ADVANCE PROGRAMMING IN C#

المحاضرة الحادي عشر

علوم حاسوب وتقنية المعلومات - مستوى ثالث - ترم ثاني

أ/ نجم الدين الدغار [najmuddin.developer@gmail.com](mailto:najmuddin.developer@gmail.com)

## ❖ انشاء ادوات.



### 1- انشاء [PictureBox] دائرية

```
class CircalPictureBox : PictureBox
{
    public CircalPictureBox()
    {
        // Image = Image.FromFile(@"");
        BackColor = Color.Red; // لان لونها غير مرئي
    }
    protected override void OnPaint(PaintEventArgs e)
    {
        base.OnPaint(e);
        GraphicsPath gpthe = new GraphicsPath();
        gpthe.AddEllipse(0, 0, Width, Height);
        this.Region = new Region(gpthe);
        // e.Graphics.DrawEllipse(Pens.Yellow, 0, 0, Width - 2, Height - 2); // بهذه الحالة يتم الرسم على المساحة الاصلية فقط لم يتم التعديل على الشكل الاصيلي
    }
}
```

### 2- توضيح الفرق بين الرسم باستخدام [CreateGraphics] و [GraphicsPath& Region]

أيضا توضيح الفرق بين الرسم داخل احداث مختلفة.

```
class MixedButton : Button//Control//UserControl
{
    public enum Dircetion { Top, Bottm, Lfet, Right }
    public Dircetion setDircetion ;
    public Dircetion choosegraphics
    {
        set
        {
            setDircetion = value;
            switch (value)
            {
                case Dircetion.Top:
                    CreateGraphics().DrawLine(Pens.Blue, 0, 0, Width - 1, 0); break;
                case Dircetion.Bottm:
                    CreateGraphics().DrawLine(Pens.Blue, 0, Height - 1, Width - 1, Height - 1); break;
                case Dircetion.Lfet:
                    CreateGraphics().DrawLine(Pens.Blue, 0, 0, 0, Height - 1); break;
            }
        }
    }
}
```

```

        case Direction.Right:
CreateGraphics().DrawLine(Pens.Blue, Width - 1, 0, Width - 1, Height - 1);
break;
    }

    }
    get { return setDirection; }
}
public MixedButton()
{
    Width = 150; Height = 70;
    FlatStyle = FlatStyle.Popup; //border
    CreateGraphics().DrawLine(new Pen(Color.Black, 3), 2, 2, 0,
Height - 2); //يظهر ويختفي من دون ان نلاحظه
    MouseMove += delegate {
        // Text = "MouseMove";

        CreateGraphics().DrawLine(new Pen(Color.Black, 3), 2, 2,
Width - 2, Height - 2);

    };

    Click += (s, e) =>
    {
        Text = "click";
        GraphicsPath p = new GraphicsPath();
        p.AddEllipse(0, 0, Width - 1, Height - 1); // لا يختفي لانه رسم
        مساحة الاداة الاصلية وكأنه نحت
        Region = new Region(p);

    };
}
protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e);
    //رسم فقط حيث ان المساحة الاصلية تظل كما هي
    // e.Graphics.DrawRectangle(Pens.Red, 0, 0, Width - 1, Height -
1);
    e.Graphics.DrawEllipse(Pens.Red, 0, 0, Width - 1, Height -
1);
    //نحت على المساحة الاصلية اي انه ما قبل الرسم لن يكون من مساحة الاداة
    //GraphicsPath gpthe = new GraphicsPath();
    //gpthe.AddEllipse(0, 0, Width, Height);
    //this.Region = new Region(gpthe);
}
}

```

### 3-إنشاء زر مع إضافة خاصية مركبة تمثل الشكل اما دائري او مربع

```

class CircularRectangularButton:Control
{
    public enum ShapeType { Rectangular, Circular }
    //-
    private ShapeType _Shape= ShapeType.Rectangular;
    public ShapeType Shape
    {
        set { if (value != _Shape)
            {
                _Shape = value;
                Invalidate(); // OnPaint تجري عملية التحديث يعني تستدعي حدث

```

```

    }
    }
    get { return _Shape; }
}
// -
// public ShapeType Shape { get; set; } =
ShapeType.Rectangular; // يغني عن السابق
protected override void OnPaint(PaintEventArgs e)
{
    base.OnPaint(e);
    GraphicsPath path = new GraphicsPath ();
    Rectangle rect = new Rectangle(0, 0, Width-1, Height-1);
    // switch (Shape) // get هنا تستغل
    switch (_Shape)
    {
        case ShapeType.Rectangular:
            path.AddRectangle(rect);
            // path.AddRectangle(new Rectangle(0, 0, Width - 1,
Height - 1));
            break;
        case ShapeType.Circular:
            path.AddEllipse(rect);
            break;
        default:
            break;
    }
    this.Region = new Region(path);
}
public CircularRectangularButton()
{ BackColor = Color.LightBlue; // مرئي عندما نوريث من كنترول يكون لونها غير مرئي
    Click += (s, ss) => {
        MessageBox.Show("Test");
    };
}
public Color MyColor
{
    set
    {
        if (value != Color.Red)
        {
            BackColor = value;
        }
        else
        {
            MessageBox.Show("غير مسموع بالخلفية الحمراء");
        }
    }
    get { return base.BackColor; }
}
public Font MyFont
{
    set
    {
        MessageBox.Show("لا يوجد نص على الاداة");
        if (false)
        {
            Font = value;
        }
    }
    get { return base.Font; }}
}

```



## إنشاء زر مثلث مع التنعيم وعمل [override] للخاصية [Text]

```
public class TriangleButton : Button
{
    public override string Text
    {
        get => base.Text;
        set
        {
            if (value!="")
            {
                base.Text = value;
            }
            else
                MessageBox.Show("لا يمكن وضع قيمة فارغة");
        }
    }
    protected override void OnPaint(PaintEventArgs e)
    {
        base.OnPaint(e);
        Graphics g = e.Graphics;
        g.SmoothingMode = SmoothingMode.AntiAlias;

        GraphicsPath path = new GraphicsPath();
        Point[] points = {
            new Point(Width / 2, 0), // رأس المثلث
            new Point(0, Height), // الزاوية اليسرى
            new Point(Width, Height) // الزاوية اليمنى
        };
        path.AddPolygon(points);
        this.Region = new Region(path);
        //using تعمل على تدمير الكائن الذي ينشئ داخلها تلقائياً لتحرير الموارد
        using (SolidBrush brush = new SolidBrush(this.BackColor))
        {
            g.FillPath(brush, path);
        }
        using (Pen pen = new Pen(this.ForeColor, 2))
        {
            g.DrawPath(pen, path);
        }

        StringFormat sf = new StringFormat
        {
            Alignment = StringAlignment.Center,
            LineAlignment = StringAlignment.Center
        };
        g.DrawString(this.Text, this.Font, Brushes.White,
            this.ClientRectangle/*width,height*/, sf);
    }
}
```

### الواجب:

- 1- إنشاء زر خماسي الاضلاع عند تحريك الماوس عليه يتغير لونه وحجمه بحيث إذا كان حجمه سوف يزداد على حجم الفورم يمنع زيادة الحجم للزر او انه يتم زيادة حجم الفورم؟
- 2- إنشاء [Label] على شكل نجمة تأخذ خلفية بلون احمر او ازرق او ابيض فقط ؟
- 3- إنشاء زر متعدد الاشكال بحيث يمكنه التبديل بين دائري او مثلث او خماسي عند تغيير خاصية

[ShapeType]