



# برمجة متقدمة #C ADVANCE PROGRAMMING IN C#

## المحاضرة التاسعة

علوم حاسوب وتقنية المعلومات - مستوى ثالث - ترم ثاني

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

- 1- توضيح تغير الابعاد
- 2- إدارة الاحداث والخصائص
- 3- الفرق بين Panel&Control&UserControl
- 4- كيفية التعامل مع UserControl من التصميم
- 5- الرسم على الأدوات

## 1- توضيح تغير الابعاد & 2- إدارة الاحداث والخصائص الكود

```

public class CustomPanel : Panel
{
    private Button innerButton; // زر داخل البنل

    public CustomPanel()
    {
        innerButton = new Button();
        innerButton.Dock = DockStyle.Fill; // جعله بحجم البنل
        innerButton.Click += InnerButton_Click;
        this.Controls.Add(innerButton);

        // حدث تغيير الحجم
        this.Resize += Panel_Resize;
    }

    // ضبط حجم الزر عند تغيير حجم البنل
    private void Panel_Resize(object sender, EventArgs e)
    {
        innerButton.Size = this.Size; // جعل الزر بنفس حجم البنل
        //innerButton.Width = Width;
        //innerButton.Height = Height;
    }

    // الخاص بالبنل عند الضغط على الزر Click استدعاء حدث
    private void InnerButton_Click(object sender, EventArgs e)
    {
        this.OnClick(e); // الخاص بالبنل OnClick استدعاء
        // this.OnClick(EventArgs.Empty/*null*/);
    }

    // تغليف خاصية النص لزر
    public string ButtonText
    {
        get { return innerButton.Text; }
        set { innerButton.Text = value; }
    }

    // إعادة تعريف OnClick
    protected override void OnClick(EventArgs e)
    {
        base.OnClick(e); // استدعاء الحدث الأصلي لكي يتم تشغيل حدث الكلك اذا وجد
        MessageBox.Show("custompanel تم الضغط على");
    }
}

```

## الهدف

- إدارة الأحداث (Resize, Click, OnClick) ومعرفة كيفية توجيه الأحداث من عنصر مخفي (Button) إلى العنصر الأساسي (Panel).
- تغليف الخصائص (Encapsulation) باستخدام get و set لإنشاء خاصية نص (Text).
- الفرق بين Click و OnClick ومعرفة ترتيب استدعائهما.
- متى نستخدم override مع OnClick وماذا يحدث إذا لم نستدعِ .base.OnClick(e);
- شرح العلاقة بين الدالة والحدث التابع لها مثلًا الدالة OnClick والأحداث Click.

## توضيح الأهداف

- عند إضافة Click للـ Panel، لا يمكن استدعاؤه مباشرة عند النقر لأن الزر يغطيه بالكامل.
- عند التنفيذ الـ Button يغطي الـ Panel بالكامل، مما يجعل أي حدث Click موجهاً للزر فقط.
- لحل المشكلة، جعلنا الـ Button يستدعي الـ OnClick الخاص بـ Panel عند الضغط عليه لكي يتم استدعاء حدث الـ Click الخاص بـ Panel.

## لماذا استخدمنا OnClick()؟

- لأن Click لا يتم استدعاؤه إلا إذا كان هناك مشترك (=+) ، ولكن OnClick يتم استدعاؤه دائماً عند الضغط، (الحدث)، أيضاً عند استدعاء OnClick، سيتم استدعاء أي مشترك (=+) في Click.
- عند النقر على Panel أو Button، يتم تنفيذ OnClick أولاً، ثم Click إذا كان هناك مشتركون (=+) فيه حيث أن الـ Click عبارة عن حدث (Event) والـ OnClick دالة (Method) قابلة للوراثة [راجع الفقرة 8 من المحاضرة الرابعة](#).

## متى نستخدم override ؟

- عندما نريد تغيير سلوك الحدث بدون الحاجة إلى += خارجي.
  - عند إنشاء عنصر مخصص (Custom Control) بحيث يكون سلوكه مختلفاً عن سلوك الأداة الأصلية.
- ملاحظة/إذا لم نستدعِ base.On...(e)، فلن يتم تنفيذ الحدث المشترك به(=+) .

أي حدث يبدأ بـ On يقوم بتشغيل الحدث المقابل له  
مثل OnClick(EventArgs e) يشغل Click و  
OnResize(EventArgs e) يشغل Resize وهكذا

إعلم أن

أي دالة On... تشغل الحدث المقابل لها (OnResize) يشغل (Resize).  
إذا كان هناك عنصر يغطي الاداة، فإن Click لن يعمل إلا إذا تم استدعاء OnClick يدوياً  
وهكذا مع أي حدث.  
يجب استدعاء base.On...(e); في override حتى لا يتم تعطيل الحدث المرتبطة.

### 3- الفرق بين UserControl&Control&Panel

#### تفضيلات

- إذا كان الهدف إنشاء عنصر تحكم جديد من الصفر (بدون عناصر داخلية) ورث من Control.
- إذا كان الهدف تصميم واجهة تجمع عدة عناصر داخلها ورث من UserControl.
- إذا كنت تحتاج فقط إلى تجميع عناصر مع دعم AutoScroll استخدم Panel مباشرةً.
- UserControl يدعم التصميم اليدوي (سحب وإفلات).

### 4- كيفية التعامل مع UserControl من التصميم

#### خطوات إنشاء UserControl

- a. من Project اختر (Add User Control (Windows Forms).
- b. سمّه واضغط Add.
- c. سيتم فتح مصمم الواجهة، أضف أدوات مثل Button, Label وقم بتصميمها وبرمجتها حسب الحاجة ثم قم بإعادة بناء المشروع، بعدها يمكن سحبها من قائمة الأدوات Toolbox.

**بناء زر بالاعتماد على خطوط مستقيمة مع إضافة خاصية العنوان الكود**

```
using System;
using System.Drawing;
using System.Windows.Forms;
namespace Advance_Programming_Labs.lab9
{
    class MyButton : UserControl
    {
        private Label label1;
        public string TText
        {
            set
            {
                label1.Text = value;
            }
            get {
                return label1.Text; }
        }
        public MyButton()
        {
            Start();
            label1.Left = 2;
            label1.Top = 2;
            label1.Width = this.Width - 5;
            label1.Height = this.Height - 5;

            this.Resize += delegate (object sender, EventArgs e)
            {
                label1.Left = 2;
                label1.Top = 2;
                label1.Width = this.Width - 5;
                label1.Height = this.Height - 5;
            };

            label1.Click += delegate (object sender, EventArgs e)
            {
                this.OnClick(e);
            };
        }

        private void Start()
        {
            //label
            this.label1 = new Label();
            label1.Text = "label1";
            this.label1.TextAlign = ContentAlignment.MiddleCenter; // توسط
            label1.MouseDown += new
            MouseEventHandler(this.label1_MouseDown);
            label1.MouseUp += new MouseEventHandler(this.label1_MouseUp);

            Controls.Add(this.label1);

            Size = new Size(100, 30);
            this.Paint += new PaintEventHandler(this.myPaint);
            this.MouseDown += new MouseEventHandler(this.myMouseDown);
        }
    }
}
```



```

        this.MouseUp += new MouseEventHandler(this.myMouseUp);
    }

    private void myPaint(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 myMouseDown(object sender, MouseEventArgs e)
    {
        DrawLine(new Pen(Color.Black, 2), new Pen(Color.White, 2));
    }

    private void myMouseUp(object sender, MouseEventArgs e)
    {
        DrawLine(new Pen(Color.White, 2), new Pen(Color.Black, 2));
    }

    void DrawLine(Pen pen1, Pen pen2) {
        Pen p1 = pen1;
        Pen p2 = pen2;
        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);
    }
}

```

الواجب:

- 1- أنشئ Panel عادي في Form، وأضف بداخله زر (Button).
- 2- أنشئ UserControl بنفس التصميم، بحيث يحتوي على Panel وزر (Button).
- 3- اجعل Button في UserControl يستدعي onClick الخاص بـ UserControl.
- 4- أضف Click لكل العنصرين (Panel العادي و UserControl).
- 5- جرب تنفيذ البرنامج ولاحظ الفرق بين Panel العادي و UserControl.
- 6- ما الفرق بينهما؟ متى تختار Panel ومتى تختار UserControl؟