برمجة متقدمة ADVANCE PROGRAMMING IN C#

المحاضرة الثامنة

علوم حاسوب وتقنية المعلومات - مستوى ثالث – ترم ثاني

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

❖ إضافة خصائص بسيطة ومركبة للأدوات.

- في هذا المحاضرة، نوضح كيفية **إضافة خصائص جديدة إلى الأدوات (Controls)** وتحديدًا في فئة مخصصة ButtonWithProperteالمشتقة منButton .
- **أهمية إضافة خصائص إضافية للأدوات:**

عند إنشاء أدوات مخصصة مثل ButtonWithProperte، يمكننا توسيع وظائف الأداة دون الحاجة إلى إعادة كتابة الكود لكل زر بعض الفوائد:

- ✅ **يمكن استخدام الزر بنفس الخصائص في أي مشروع.**
- ✅ **بدلاً من تعديل إعدادات كل زر يدويًا، يمكن التحكم فيها من نافذة Properties.**
- ✅ **مثل تغيير اللون عند الضغط، أو تغيير النص تلقائيًا بناءً على العملية المحددة.**

١. أنواع الخصائص في#C

الخصائص في #C نوعان رئيسيان:

◆ الخصائص البسيطة(Simple Properties)

وهي الخصائص التي تخزن قيمة مباشرة، مثل النص (Text) أو العرض (Width) يمكن تغليفها باستخدام set وget.

مثال في الكود:

```
public string Ttext
{
    set { Text = value; }
    get => Text;
}
```

- **هذه خاصية تتحكم في نص الزر، حيث يمكن تعيين (set) نص جديد أو استرجاع (get)النص الحالي.**

◆ الخصائص المركبة(Complex Properties)

هي الخصائص التي تستند إلى أنواع بيانات مركبة مثل enum.

مثال في الكود:

```
enum myenumOP { sub, mul, div, add }
public myenumOP operation
{
    set { op = value; Invalidate(); Text = op.ToString(); }
    get { return op; }
}
```

- **هذه خاصية تعتمد على التعداد enumوتتحكم في العملية (operation) عند تغيير قيمتها، يتم تحديث نص الزر تلقائيًا ليعكس العملية المحددة.**

٢. طرق تغليف الخصائص(Encapsulation)

◆ التغليف باستخدام الدوال العادية:

في هذه الطريقة، نستخدم دوال set وgetللتحكم في القيم يدويًا

مثال في الكود:

```
enum myenumOP { sub, mul, div, add }
```

```
myenumOP op;
public void setop(myenumOP k)
{
    op = k;
}

public myenumOP getop()
{
    return op;
}
```

✅ميزة هذه الطريقة :

- تعطي تحكمًا أكبر عند تنفيذ منطق معقد عند تعيين القيم.

❌عيبها:

- لا تظهر كخاصية في نافذة Properties عند التصميم في بيئة Visual Studio.

◆التغليف باستخدام set وget الخاصة بالنظام

في هذه الطريقة، نستخدم الخصائص (Properties) بدلًا من الدوال:

مثال في الكود:

```
public myenumOP operation
{
    set { op = value; Invalidate(); Text = op.ToString(); }
    get { return op; }
}
```

✅ميزة هذه الطريقة:

- تظهر الخاصية داخل نافذة Properties في Visual Studio أثناء التصميم (عند السحب والإفلات).
- يمكن تعديلها بسهولة في وقت التصميم بدلاً من وقت التشغيل فقط.

◆مثال شامل:

```
namespace Advance_Programming_Labs.lab8
{
     enum myenumOP { sub, mul, div, add }
     enum Myenumbool { True, False, }
    //ملاحظة اذا عرفت انيومريشن داخل الكلاس وقمت بتغليفة باستخدتم ست وجت يجب ان يكون عام
    class ButtonWithProperte : Button
    {
        string x = "private attribute";//خاص يحتاج تغليف
        public int ttext;//عام

        public void set(string y)
        {
            Text = y;
          //  x= y;
        }
        public string get() => Text/*x*/;
        //set,getتغليف عن طريق
        public string Ttext
        {         //خاصية بسيطة
            set
```

```
            {
                Text = value;
            }
            get => Text;// "ali";
        }
        //----------------
        myenumOP op;//خاصية مركبة
        //تغليف عادي
        public void setop(myenumOP k)
        {
            op = k;
        }
        public myenumOP getop()
        {
            return op;
        }
        //set,getتغليف عن طريق
        public myenumOP operation
        {
            set { op = value; Invalidate();/*تعمل على التحديث*/
                Text = op.ToString();
            }
            get { return op; }

        }
        Myenumbool isred ;
        public Myenumbool RedBackColor
        {
            set
            {
                if (value == Myenumbool.True)
                {
                    BackColor = Color.Red;
                }
                else
                {
                    UseVisualStyleBackColor = true;//اللون الاصلي
                }
                isred = value;
            }
            get { return isred; }
        }
      public ButtonWithProperte()
        {
            RedBackColor = Myenumbool.True;
            Width += 50;
        }
    }

}
```

◆اذا طلب منك بناء أداة كما بالشكل مزودة بخاصية نوع العملية بحيث عند اختيار العملية يقوم بتغير النص الخاص بالزر حسب العملية المختارة:

الكود

```
public class SimpleCalcPanelWithProperties : Panel
{
    private Label[] labels = new Label[3];
    private TextBox[] textBoxes = new TextBox[3];
    private Button btn;
    public enum myenum { add, sub, div, mul }
    myenum op;
    public myenum Operation
    {
        set
        {
            op = value;
            if (op == myenum.add)
            {
                btn.Text = "add";
            }
            else if (op == myenum.div)
            {
                btn.Text = "div";
            }
            else if (op == myenum.mul)
            {
                btn.Text = "mul";
            }
            else if (op == myenum.sub)
            {
                btn.Text = "sub";
            }
            //  btn.Text = op.ToString();يغني عن جميع الشروط السابقة
        }
        get { return op; }
    }

    public SimpleCalcPanelWithProperties()
    {
        // ضبط خصائص النموذج
        this.Text = "آلة ضرب بسيطة";
        this.Size = new Size(200, 150);
        BackColor = Color.Blue;
        Top = Left = 10;
        // أسماء الحقول
        string[] labelTexts = { "النتيجة" ,"العدد الثاني" ,"العدد الأول" };

        // إنشاء الحقول باستخدام الدوران
        for (int i = 0; i < 3; i++)
```

```
            {
                // إنشاء وتصميم التسمية الليبل
                labels[i] = new Label()
                {//طريقة مباشر لاعطاء قيم للخصائص
                    Text = labelTexts[i],
                    Left = 100,
                    Top = 20 + (i * 30),
                    Width = 80,
  ForeColor = Color.White
                };

                // إنشاء وتصميم مربع النص
                textBoxes[i] = new TextBox()
                {
                    Left = 10,
                    Top = 20 + (i * 30),
                    //  Top = 20 + (i * labels[i].Top),//error   Top لان الليبل
في هذه الحظة عاده لن يتم انشاءه لذلك لن نستطيع الوصول لخصائصةمن ضمنها

                    Width = 80
                };

                // جعل مربع النص الأخير (النتيجة) للقراءة فقط
                if (i == 2) textBoxes[i].ReadOnly = true;

                // إضافة الأدوات إلى النموذج
                this.Controls.Add(labels[i]);
                this.Controls.Add(textBoxes[i]);
            }

            // إنشاء وتصميم الزر
            btn = new Button()
            {
                Text = myenum.add.ToString(),
                Left = 10,
                Top = 110,
                Width = 80,
                ForeColor = Color.White
            };

            this.btn.Click += Add;

            // إضافة الزر إلى النموذج
            this.Controls.Add(btn);
        }
        double x, y, z;
        void Add(object s, EventArgs e)
        {
            bool f = false;
            try
            {
                x = Convert.ToDouble(textBoxes[0].Text);
            }
            catch (Exception)
            {

                MessageBox.Show("تحذير" ,"خطا في صندوق الادخال الاول",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }
            try
            {
                y = Convert.ToDouble(textBoxes[1].Text);
```

```
            }
            catch (Exception)
            {

                MessageBox.Show("تحذير" ,"خطا في صندوق الادخال الثاني",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }
            switch (op)
            {
                case myenum.add: z = x + y; f = true; break;
                case myenum.sub: z = x - y; f = true; break;
                case myenum.mul: z = x * y; f = true; break;
                case myenum.div: z = x / y; f = true; break;// تجنب القسمة على
صفر
            }
            if (f)
            {
                textBoxes[2].Text = z.ToString();
            }
        }
    }
```

اذا طلب منك بناء أداة كما في الشكل اسفل مزودة بخاصية نوع العملية بحيث عند تحديد العملية يقوم بتغير شكل الأداة حسب العملية المختارة .

```
using System;

using System.Collections.Generic;

using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace PasswordTextbox
{
    class operationPanel : Panel
    {
      public enum operation {sum,sub, mult, div}
       private  operation operation_type1;
        TextBox[] txt = new TextBox[3];
        Label[] label = new Label[3];
        Label lab_eq, lab_op;
        Button btn_op;
```

```
double TextN1, TextN2, TextN3;
void get_text()
{
    switch (operation_type1)
    {
        case operation.sum:
            {
                lab_op.Text = "+";
                btn_op.Text = "جمع";
                break;
            }
        case operation.sub:
            {
                lab_op.Text = "-";
                btn_op.Text = "طرح";
                break;
            }
        case operation.mult:
            {
                lab_op.Text = "*";
                btn_op.Text = "ضرب";
                break;
            }
        case operation.div:
            {
                lab_op.Text = "/";
                btn_op.Text = "قسمة";
                break;
            }
    }
}
```

```
public operation operation_type
{
    set
    {
        operation_type1 = value;
        get_text();

    }
    get { return operation_type1; }
}

public operationPanel()
{
    InitializeComponent();
```

```
    }
    public void InitializeComponent()
    {
        Size = new System.Drawing.Size(550, 150);
        BackColor = Color.Green;
        List<string> l = new List<string> { " العدد الاول", "العدد الثاني", "الناتج", };
        for (int i = 0; i < 3; i++)
        {
            label[i] = new Label();
            label[i].Size = new Size(100, 23);
            label[i].Location = new Point((110 + label[i].Width) * i+10, 13);
            label[i].Text = l[i];

            txt[i] = new TextBox();
            txt[i].Size = new Size(120, 20);
            txt[i].Location = new Point((80 + txt[i].Width) * i+10, 50);
            Controls.Add(label[i]);
            Controls.Add(txt[i]);
        }
        lab_op = new Label() { Text = "+", Location = new Point(350, 51) };
        lab_eq = new Label() { Text = "=", Location = new Point(150, 51) };
        btn_op = new Button() { Text = "جمع", Location = new Point(380, 94), Size = new Size(100, 30) };
        btn_op.Click += btn_click;
        Controls.AddRange(new Control[] { lab_op, lab_eq, btn_op });
;
    void btn_click(object sender, EventArgs e)
    {
        if (txt[2].Text.Trim() != "" && txt[1].Text.Trim() != "")
        {
            float n1, n2, n3 = 0f;
            n1 = float.Parse(txt[2].Text.Trim());
            n2 = float.Parse(txt[1].Text.Trim());
            switch (operation_type1)
            {
                case operation.sum: { n3 = n1 + n2; break; }
                case operation.sub: { n3 = n1 - n2; break; }
                case operation.mult: { n3 = n1 * n2; break; }
```

```
                case operation.div: { n3 = ((n2 != 0) ? (n1 / n2) :
0); break; }
            }
            txt[0].Text = n3.ToString();
        }
        else
        {
            MessageBox.Show("ادخل رقم..");
        }
    }

    public double TextN11 { get { return TextN1; } set { TextN1 =
value; txt[2].Text = TextN1.ToString(); } }
    public double TextN21 { get { return TextN2; } set { TextN2 =
value; txt[1].Text = TextN2.ToString(); } }
    public double TextN31 { get { return TextN3; } }


  }
}
```

الواجب:

١- قم بإنشاء أداة مخصصة[ButtonLight] يتغير لونة بشكل عشوائي عند التأشير عليه في حالة تم تحويل الخاصية المضافة له Light الى True من الخصائص؟ تلميح استخدم [Timer]

٢- قم بإنشاء زر [CreateTextBox]بمجرد دخول الماوس عليه يتم انشاء مربعات نص بمواقع عشوائية على النموذج حيث أن عدد مربعات النص التي يتم انشاءها يتم إدخاله عبر الخاصية المضافة له [NumberOfTextBox] وتكون الوان مربعات النص اما احمر او ازرق او اخضر وذلك بعد تحديد اللون من الخاصية المركبة المضافة له [ChooseColor]؟

٣- قم بإنشاء أداة مخصصة مربع نص [NumircCharTextBox] يقبل فقط إدخال الاحرف الإنجليزية والعربية وذلك بعد اختيار اللغة من الخاصية المضافة له [Language]؟

٤- قم بإنشاء ليبل [GoLabel] عند الضغط المزدوج عليه يتحرك للأمام . مقدار الحركة يتم إدخاله عبر الخاصية المضافة لديه [GoTo].

٥- قم بإنشاء أداة مخصصة MyTextBox مشتقة من TextBox تحتوي على:

١- خاصية بسيطة [Length] لتحديد الحد الأقصى للنص المدخل؟

٢- خاصية بسيطة TextLength تعيد عدد الأحرف المدخلة؟

٣- قم بتغيير لون الخلفية إلى أحمر إذا كان النص قد وصل للحد الأقصى، وإلى أبيض إذا لم يصل بعد؟

٦- قم بإنشاء أداة مخصصة MyCheckBox مشتقة من CheckBoxتحتوي على:

خاصية بسيطة IsCheckedText تعيد "نـعم "إذا كان المربع محددًا و "لا "إذا لم يكن محددًا؟