



NATIONAL ECONOMICS UNIVERSITY

SCHOOL OF INFORMATION TECHNOLOGY AND DIGITAL ECONOMICS

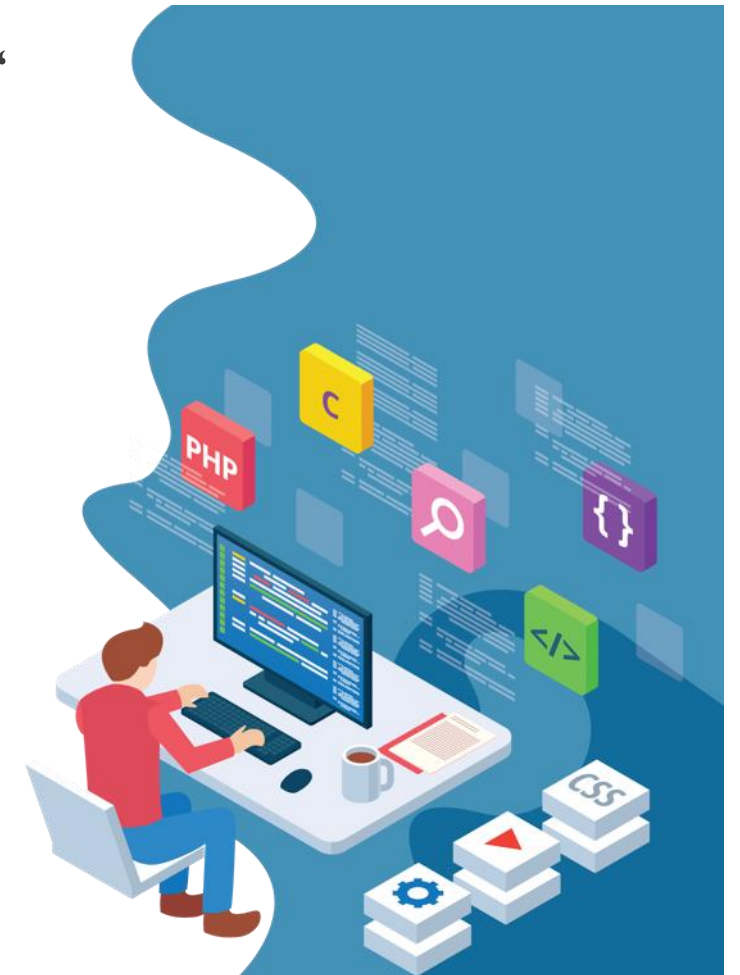
CHAPTER

WINDOWS FORMS APP (.NET FRAMEWORK)

PHAM THAO

WINDOWS FORMS APP (.NET FRAMEWORK)

- Windows Forms App (.NET Framework)" and "WinForms App“
- Working with basic controls
 - Label
 - Text Box
 - Button
 - List Box
 - Combo box
 - Check Box List
- Dialog box control
 - Message Box
 - Open File Dialog Box



WIN FORMS APP (.NET FRAMEWORK)

- "Windows Forms App (.NET Framework)" and "WinForms App" both refer to applications built using Windows Forms, but they target different versions of the .NET framework.
- Here are the key differences between the two:

Template Name	Target Framework	Cross-Platform Compatibility	Modernization	Development Ecosystem	Support and Lifecycle
Windows Forms App (.NET Framework)	.NET Framework	Windows-only	Limited	Visual Studio with traditional Windows Forms designer	Mature with limited new features
WinForms App	.NET Core, .NET 5, .NET 6, etc.	Windows, macOS, Linux	Modern development tools and cross-platform support	Visual Studio 2019 or later, Visual Studio Code, etc.	Active development with regular updates

WIN FORMS APP (.NET FRAMEWORK)

■ Target Framework:

- Windows Forms App (.NET Framework): This template is used to create Windows Forms applications that target the traditional .NET Framework, which has been in use for many years. It's typically used for Windows desktop applications on older versions of Windows.
- WinForms App: This template is used to create Windows Forms applications, but it targets the newer .NET platform, such as .NET Core (now known as .NET 5 and later versions) or .NET 6 and beyond. It is part of the more modern and cross-platform .NET ecosystem.

■ Cross-Platform Compatibility:

- Windows Forms App (.NET Framework): Applications built using this template are typically limited to running on Windows and rely on the .NET Framework, which is Windows-specific. They may not be easily portable to other platforms.
- WinForms App: Applications built with this template can target multiple platforms, including Windows, macOS, and Linux, depending on the selected target framework (.NET Core, .NET 5, .NET 6, etc.). This enables cross-platform development and deployment.

■ Modernization:

- Windows Forms App (.NET Framework): These applications use the traditional Windows Forms technology and may not benefit from the latest performance enhancements, features, and security improvements introduced in more recent .NET platforms.
- WinForms App: Applications built on newer .NET platforms can leverage the advancements made in those platforms, such as better performance, modern controls, and support for the latest C# language features.

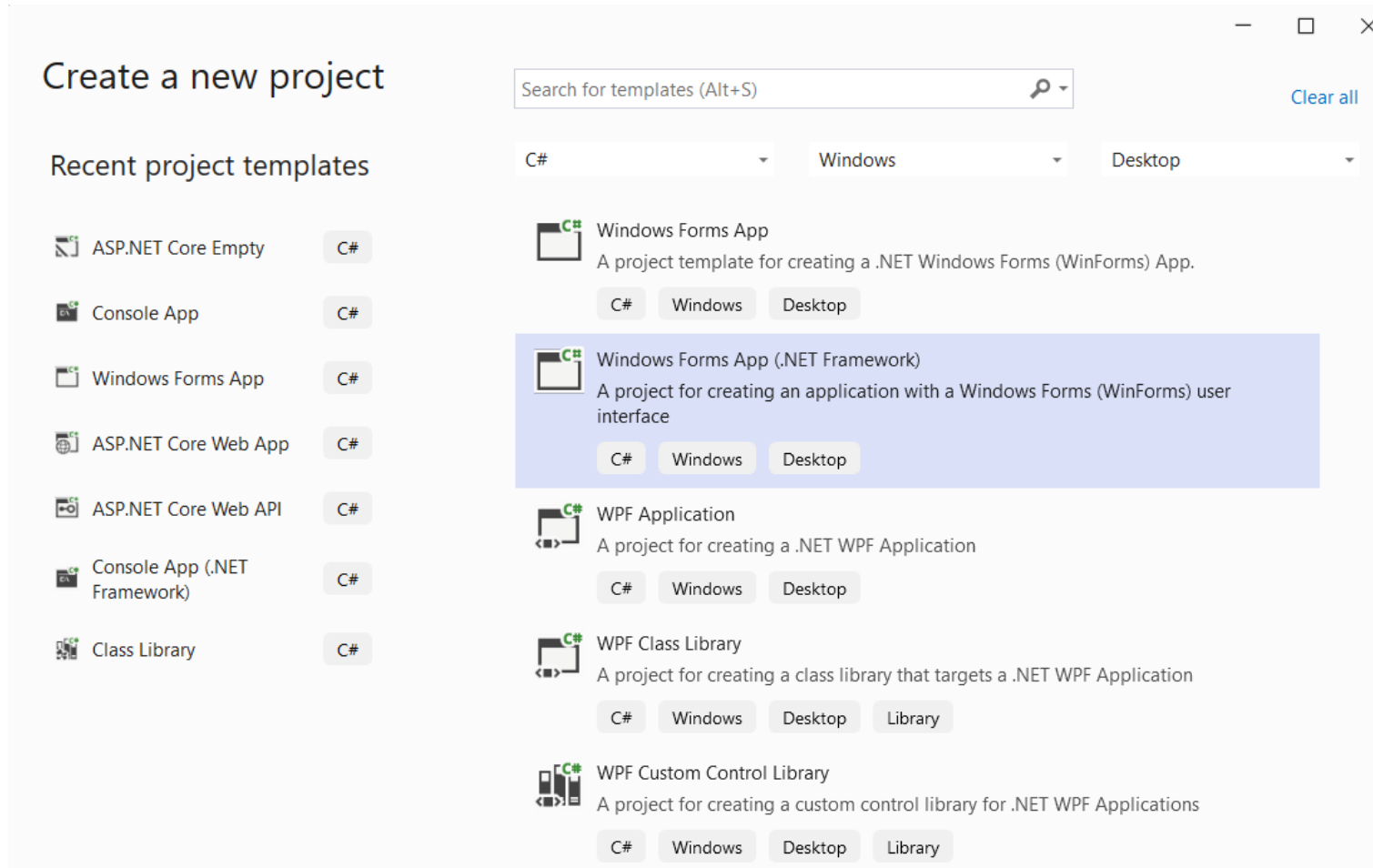
■ Development Ecosystem:

- Windows Forms App (.NET Framework): Development for this template typically involves using Visual Studio with the Windows Forms designer, following the traditional .NET development model.
- WinForms App: Development for this template involves using modern development tools like Visual Studio 2019 or later, Visual Studio Code, and newer libraries and packages.

■ Support and Lifecycle:

- Windows Forms App (.NET Framework): The .NET Framework, including Windows Forms for .NET Framework, is a mature technology with a well-defined lifecycle. It has limited support for new features and improvements.
- WinForms App: Applications targeting newer .NET platforms benefit from active development and support, with regular updates, improvements, and access to the latest features in the .NET ecosystem.

WINDOWS FORMS APP (.NET FRAMEWORK)



- Windows Forms App (.NET Framework)

WINDOWS FORMS APP (.NET FRAMEWORK)

Configure your new project

Windows Forms App (.NET Framework)

C#

Windows

Desktop

Project name

WindowsFormsAppLaiSuat

Location

D:\BMCNTT\NET LT .NET\51 WINDOWS FORM WEB APP NET FRAMEWORK

Solution name ⓘ

WindowsFormsAppLaiSuat

☐

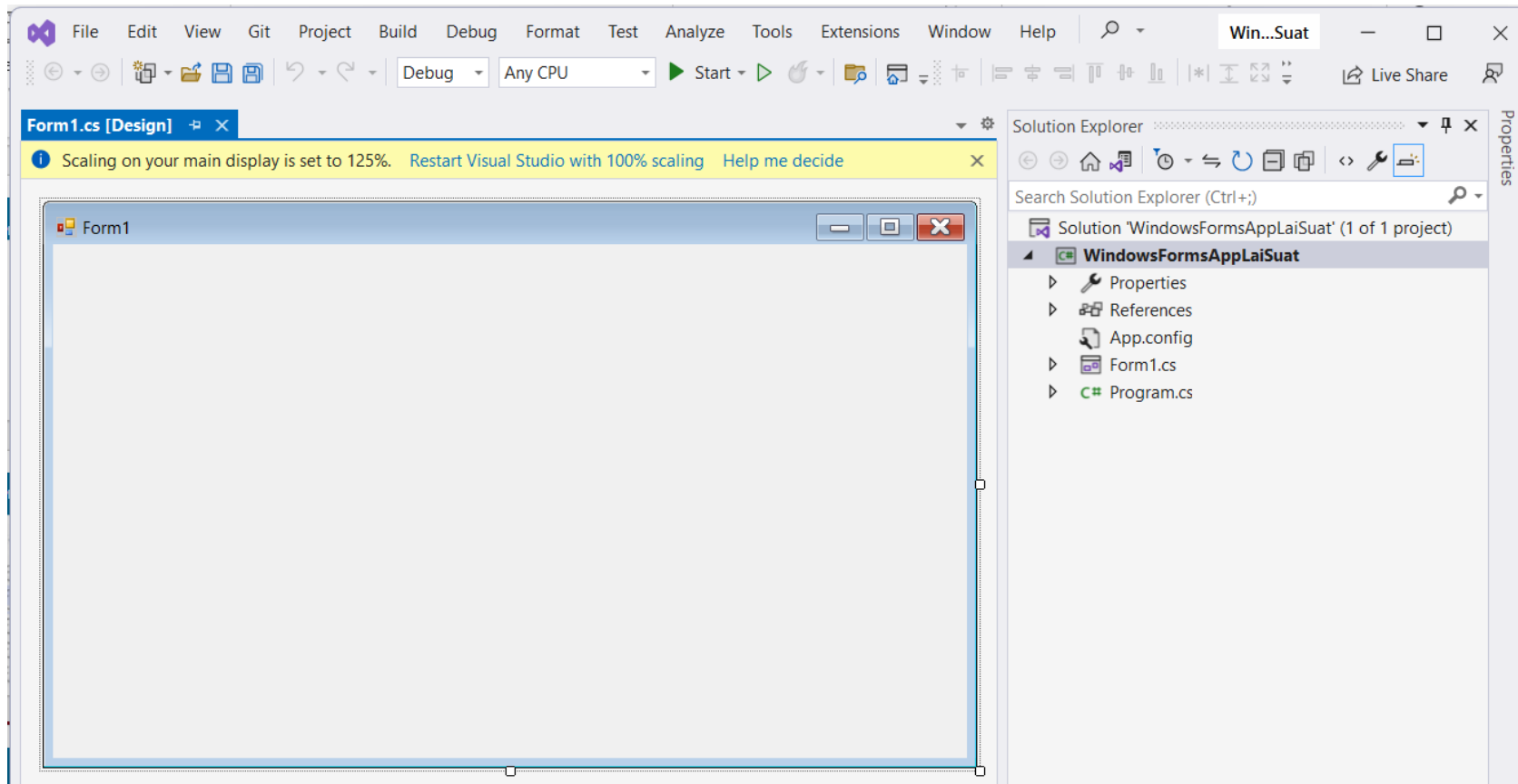
Place solution and project in the same directory

Framework

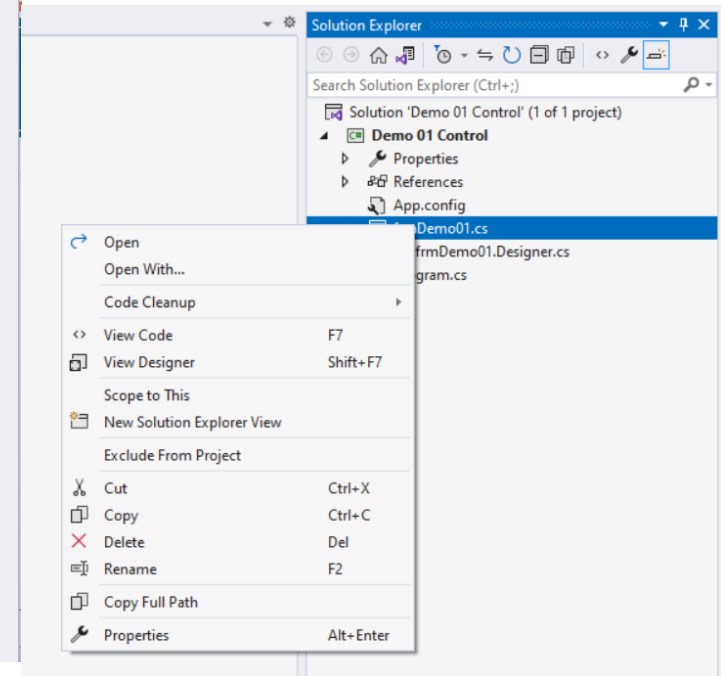
.NET Framework 4.7.2

Project will be created in "D:\BMCNTT\NET LT .NET\51 WINDOWS FORM WEB APP NET FRAMEWORK
\WindowsFormsAppLaiSuat\WindowsFormsAppLaiSuat\"

WINDOWS FORMS APP (.NET FRAMEWORK)



■ View code (F7)



WINDOWS FORMS APP (.NET FRAMEWORK)

```
namespace Demo_01_Control
{
    3 references
    public partial class frmDemo01 : Form
    {
        /// <summary>
        /// Đây là hàm tạo form.
        /// Khi đối tượng form này được khởi tạo,
        /// Một số control trên form sẽ được
        /// khởi tạo theo phần design đã đưa vào trước đó
        /// </summary>
        1 reference
        public frmDemo01()
        {
            InitializeComponent();
        }
    }
}
```

- View code (F7)

WINDOWS FORMS APP (.NET FRAMEWORK)

```
11 namespace Demo_01_Control
12 {
13     3 references
14     public partial class frmDemo01 : Form
15     {
16         // tạo form.
17         // form này được khởi tạo,
18         // control trên form sẽ được
19         // theo phần design đã đưa vào
20
21         InitializeComponent()
22     }
23
24     Run To Cursor      Ctrl+F10
25     Force Run To Cursor
26     Execute in Interactive  Ctrl+E, Ctrl+E
27     Snippet
28     Cut                Ctrl+X
29     Copy               Ctrl+C
30     Paste              Ctrl+V
31     Annotation
32     Outlining
```

tạo form.
form này được khởi tạo,
control trên form sẽ được
theo phần design đã đưa vào

1()
Component();

No issue

```
frmDemo01.Designer.cs*  frmDemo01.cs [Design]*
- Demo_01_Control.frmDemo01
- InitializeComponent()

base.Dispose(disposing),

}

#region Windows Form Designer generated code

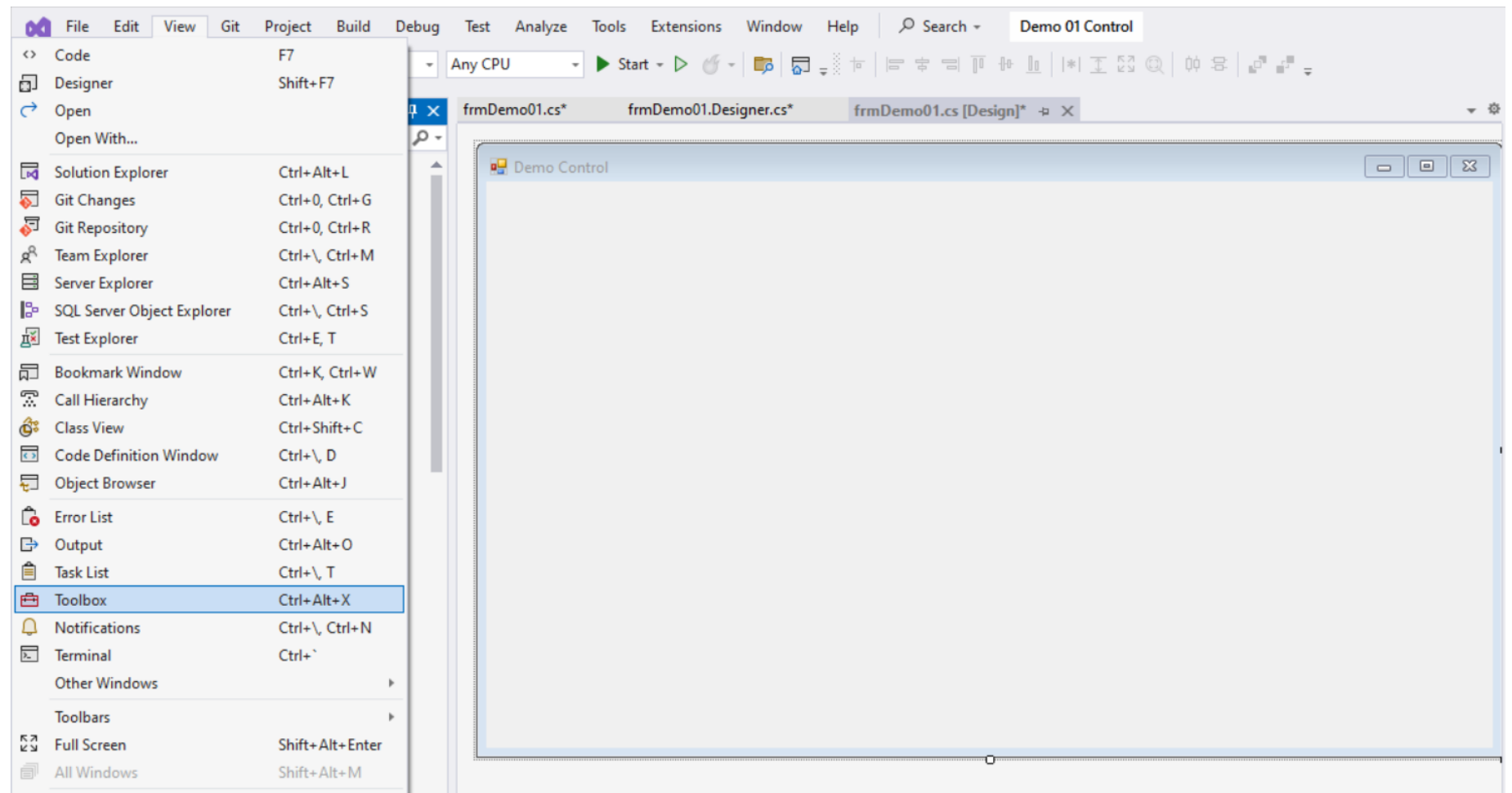
/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
1 reference
private void InitializeComponent()
{
    this.components = new System.ComponentModel.Container();
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.ClientSize = new System.Drawing.Size(800, 450);
    this.Text = "Demo Control";
}

#endregion

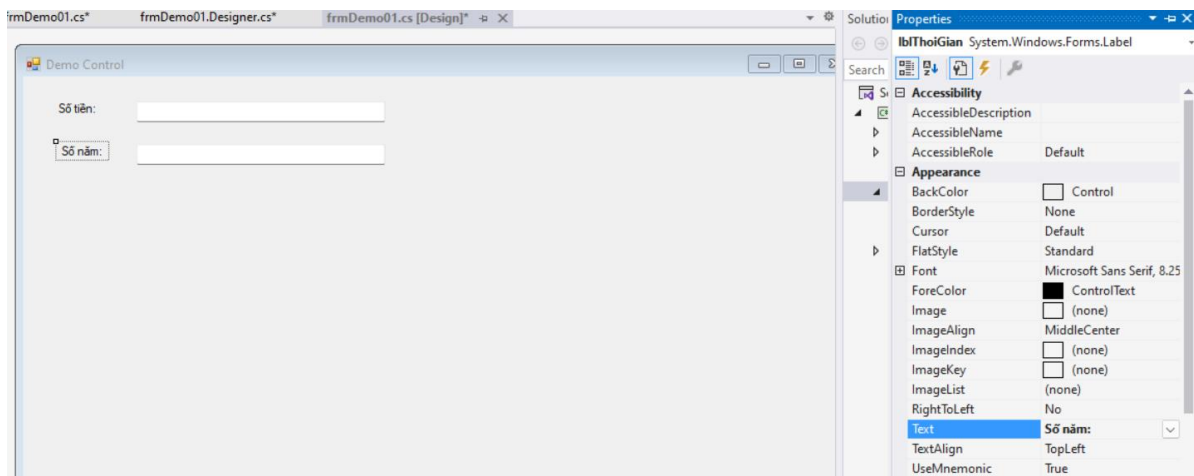
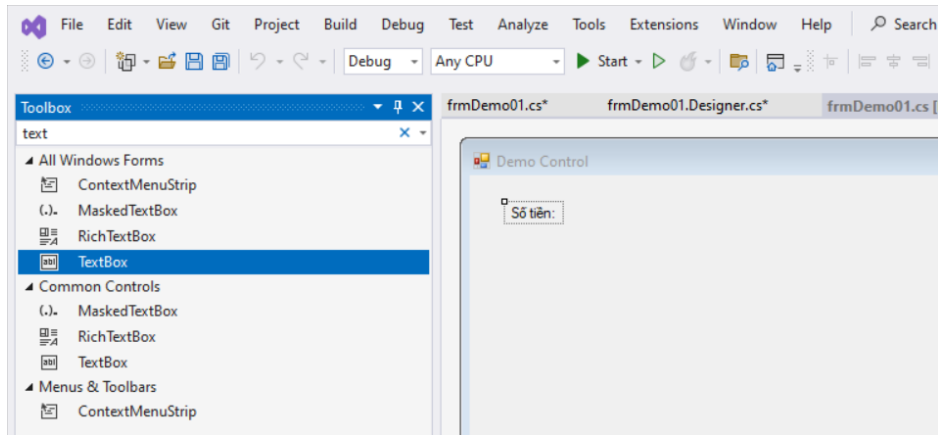
}
```

ADD CONTROL TO FORMS

- View →
Toolbox



ADD CONTROL

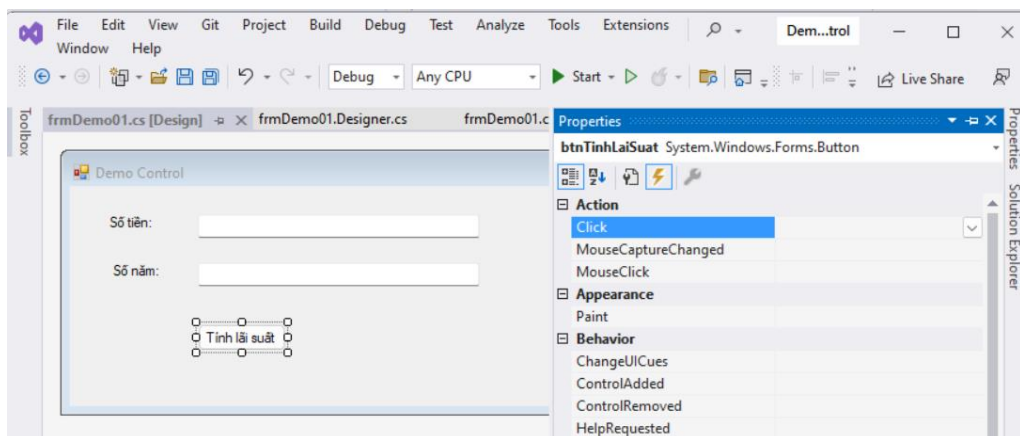
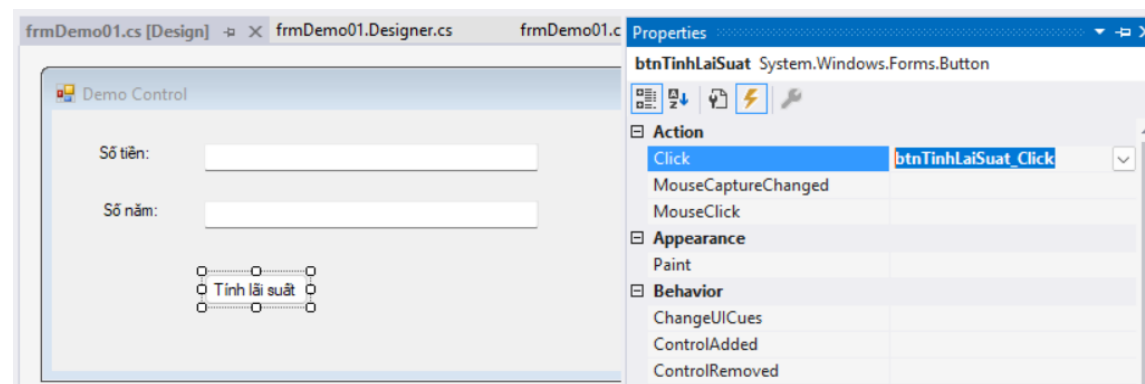
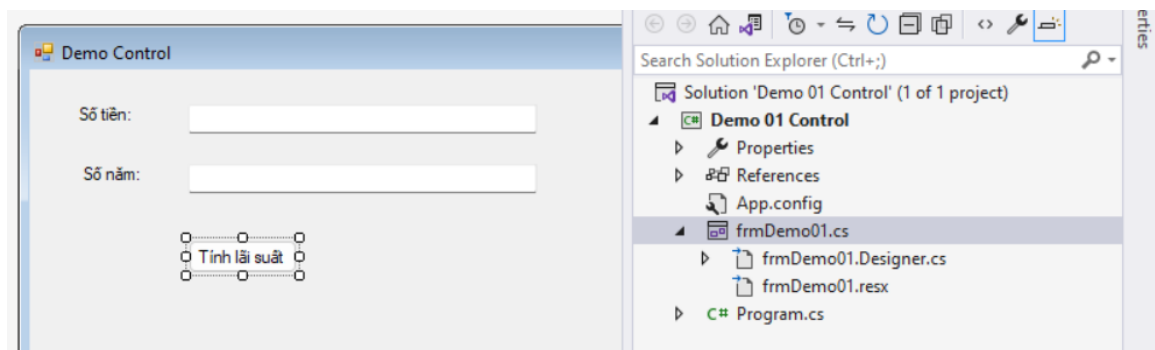


1 reference

```
private void InitializeComponent()
{
    this.label1 = new System.Windows.Forms.Label();
    this.txtSoTien = new System.Windows.Forms.TextBox();
    this.txtSoNam = new System.Windows.Forms.TextBox();
    this.lblThoiGian = new System.Windows.Forms.Label();
    this.lblMucLaisuat = new System.Windows.Forms.Label();
    this.btnTinhLaiSuat = new System.Windows.Forms.Button();
    this.SuspendLayout();
    //
    // label1
    //
    this.label1.AutoSize = true;
    this.label1.Location = new System.Drawing.Point(31, 24);
    this.label1.Name = "label1";
    this.label1.Size = new System.Drawing.Size(43, 13);
    this.label1.TabIndex = 0;
    this.label1.Text = "Số tiền:";
    //
    // txtSoTien
    //
    this.txtSoTien.Location = new System.Drawing.Point(109, 24);
    this.txtSoTien.Name = "txtSoTien";
    this.txtSoTien.Size = new System.Drawing.Size(240, 20);
    this.txtSoTien.TabIndex = 1;
    //

```

ADD BUTTON



```
//  
// btnTinhLaiSuat  
//  
this.btnTinhLaiSuat.Location = new System.Drawing.Point(109, 117);  
this.btnTinhLaiSuat.Name = "btnTinhLaiSuat";  
this.btnTinhLaiSuat.Size = new System.Drawing.Size(75, 23);  
this.btnTinhLaiSuat.TabIndex = 5;  
this.btnTinhLaiSuat.Text = "Tính lãi suất";  
this.btnTinhLaiSuat.UseVisualStyleBackColor = true;  
this.btnTinhLaiSuat.Click += new System.EventHandler(this.btnTinhLaiSuat_Click);  
//Có thể gắn thêm các hàm khác
```

ADD BUTTON - CREATE AN EVENT HANDLER AT RUN TIME

1. Open the form that you want to add an event handler to.
2. Add a method to your form with the method signature for the event that you want to handle.
3. For example, if you were handling the Click event of a Button control, you would create a method such as the following:

C#

```
private void button1_Click(object sender, System.EventArgs e)
{
    // Add event handler code here.
}
```

1. Add code to the event handler as appropriate to your application.
2. Determine which form or control you want to create an event handler for.
3. In a method within your form's class, add code that specifies the event handler to handle the event. For example, the following code specifies the event handler button1_Click handles the Click event of a Button control:

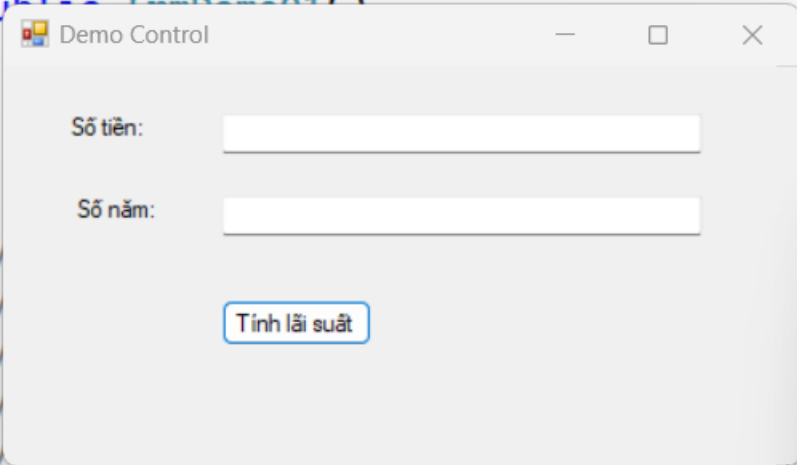
```
button1.Click += new EventHandler(button1_Click);
```

<https://learn.microsoft.com/en-us/dotnet/desktop/winforms/how-to-create-event-handlers-at-run-time-for-windows-forms?view=netframeworkdesktop-4.8>

MESSAGEBOX

1 reference

```
public FormDemo01()
```

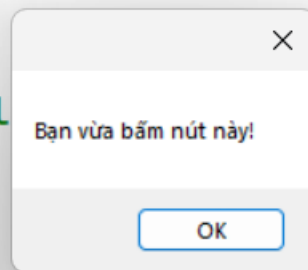


Demo Control

Số tiền:

Số năm:

Tính lãi suất



×

Bạn vừa bấm nút này!

OK

1 reference

```
private void btnTinhLaiSuat_Click(object sender, EventArgs e)
{
    MessageBox.Show("Bạn vừa bấm nút này!");
}
```


MESSAGEBOX

1 reference

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

```
{
```

```
// MessageBox.Show("Bạn vừa bấm nút này!");
```

```
MessageBox.Show("Bạn vừa bấm nút Tính lãi suất", "Thông báo", MessageBoxButtons.YesNoCancel, Mes
```

▲ 5 of 21 ▼ DialogResult MessageBox.Show(string text, string caption, MessageBoxButtons buttons)
Displays a message box with specified text, caption, and buttons.
buttons: One of the MessageBoxButtons values that specifies which buttons to display in the message box.

★ OK
★ OKCancel
★ RetryCancel
★ YesNoCancel
★ YesNo
AbortRetryIgnore
OK
OKCancel
RetryCancel

```
EventArgs e)
```

```
it", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.)
```

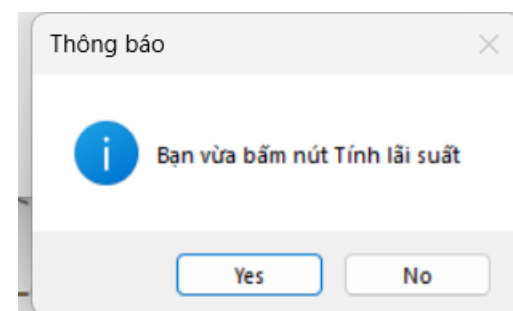
s buttons, MessageBoxIcon icon)

the message box.

MessageBoxIcon.Information = 64
The message box contains a symbol consisting of a lowercase letter i in a circle.
★ IntelliCode suggestion based on this context

★ Warning
★ Question
★ Information
★ Exclamation
★ Error
Asterisk
Error
Exclamation
Hand

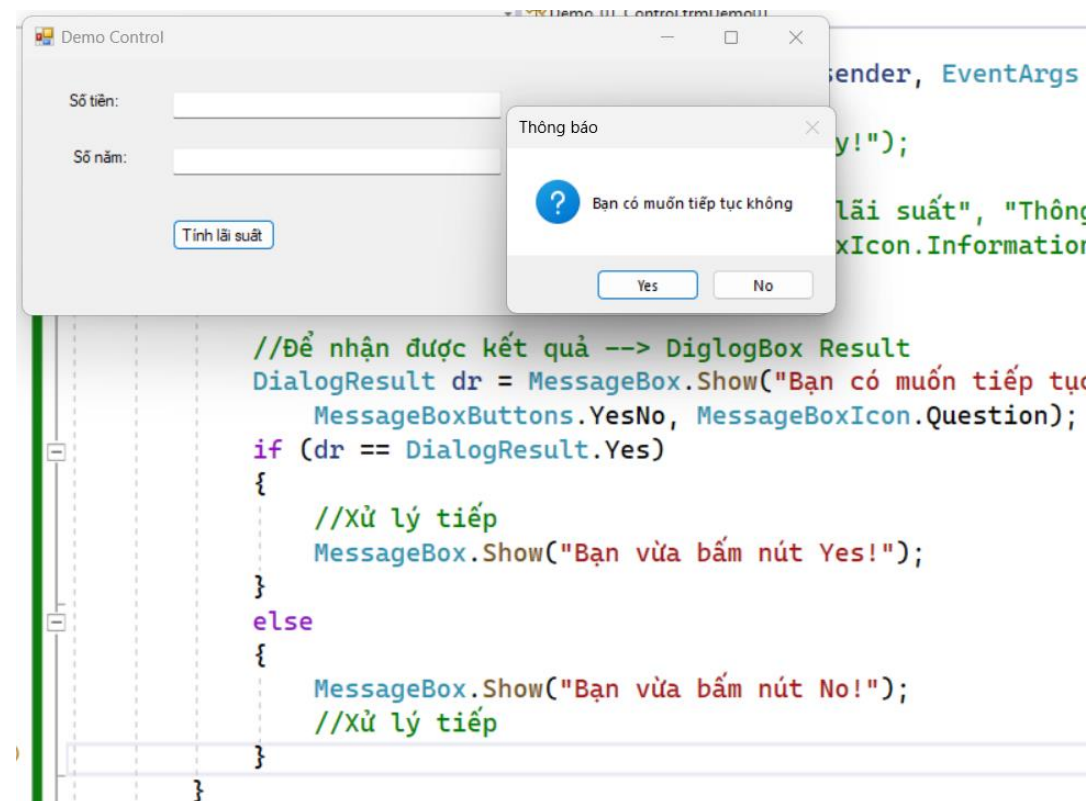
MessageBoxButtons.YesNoCancel = 3
The message box contains Yes, No, and Cancel buttons.
★ IntelliCode suggestion based on this context



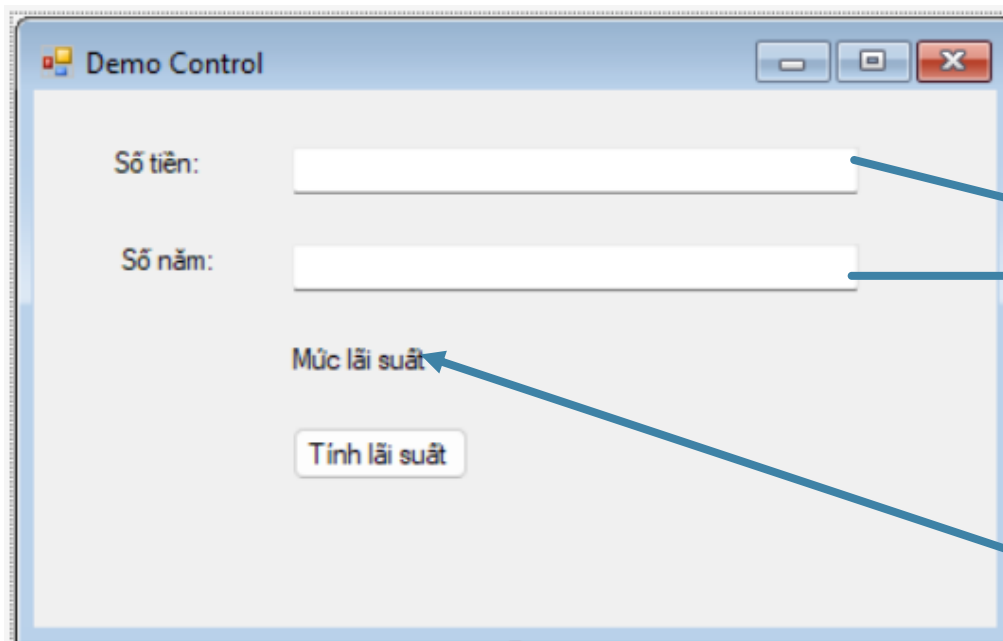
MESSAGE BOX – DIALOG RESULT

```
1 reference
private void btnTinhLaiSuat_Click(object sender, EventArgs e)
{
    // MessageBox.Show("Bạn vừa bấm nút này!");
    /*
    MessageBox.Show("Bạn vừa bấm nút Tính lãi suất", "Thông báo",
        MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    */

    //Để nhận được kết quả --> DialogResult Result
    DialogResult dr = MessageBox.Show("Bạn có muốn tiếp tục không", "Thông báo",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (dr == DialogResult.Yes)
    {
        //Xử lý tiếp
        MessageBox.Show("Bạn vừa bấm nút Yes!");
    }
    else
    {
        MessageBox.Show("Bạn vừa bấm nút No!");
        //Xử lý tiếp
    }
}
```



GET DATA FROM TEXTBOX AND LABEL



Demo Control

Số tiền:

Số năm:

Mức lãi suất

Tính lãi suất

```
//Xử lý tiếp  
MessageBox.Show("Bạn vừa bấm nút Yes!");  
//Lấy giá trị số năm và thời gian từ 2 ô text  
int sonam = int.Parse(this.txtSoNam.Text);  
int sotien = int.Parse(this.txtSoTien.Text);  
//Gọi hàm tính lãi suất  
float muclaisuat = 0.0f;  
//muclaisuat = TinhLaiSuat(sotien, sonam);  
//Nhận về giá trị mức lãi suất  
this.lblMucLaisuat.Text = muclaisuat.ToString();  
//Hiện lên trên label
```

CALL METHOD

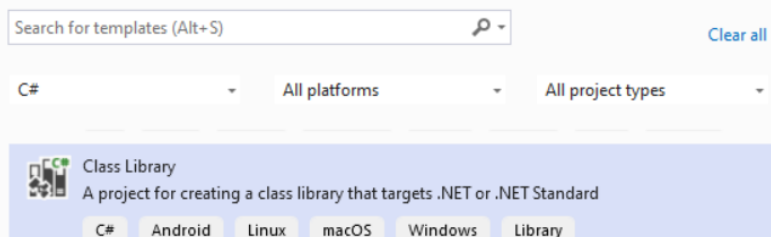
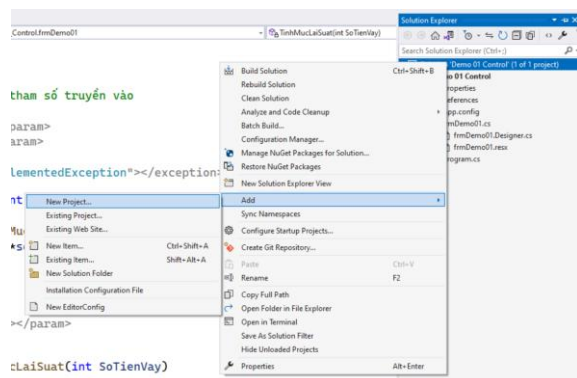
```
14 //Xử lý tiếp
15 MessageBox.Show("Bạn vừa bấm nút Yes!");
16 //Lấy giá trị số năm và thời gian từ 2 ô text
17 int sonam = int.Parse(this.txtSoNam.Text);
18 int sotien = int.Parse(this.txtSoTien.Text);
19 //Gọi hàm tính lãi suất
20 float muclaisuat = 0.0f;
21 muclaisuat = TinhLaiSuat(sotien, sonam);
22
23 // Generate method 'TinhLaiSuat'
24 Change 'TinhLaiSuat' to 'btnTinhLaiSuat'.
25 Change 'TinhLaiSuat' to 'btnTinhLaiSuat_Click'.
26
27 else {
    // CS0103 The name 'TinhLaiSuat' does not exist in the current context
    Lines 62 to 63
    {
        private float TinhLaiSuat(int sotien, int sonam)
        {
            throw new NotImplementedException();
        }
    }
}

/// <summary>
/// Tính lãi suất theo các tham số truyền vào
/// </summary>
/// <param name="sotien"></param>
/// <param name="sonam"></param>
/// <returns></returns>
/// <exception cref="NotImplementedException"></exception>
1 reference
private float TinhLaiSuat(int sotien, int sonam)
{
    float muclaisuat = TinhMucLaiSuat(sotien);
    return muclaisuat*sonam*sotien/100;
}
```

```
1 reference
private static float TinhMucLaiSuat(int SoTienVay)
{
    float MucLaiSuat = 0;
    if (SoTienVay < 5000)
    {
        // Console.WriteLine("Không cho vay");

        MucLaiSuat = 0;
    }
    else if (SoTienVay < 100000)
    {
        MucLaiSuat = 10.5F;
    }
    else if (SoTienVay < 250000)
    {
        MucLaiSuat = 10;
    }
    else if (SoTienVay < 500000)
    {
        MucLaiSuat = 9.5F;
    }
    else
    {
        MucLaiSuat = 9;
    }
    return MucLaiSuat;
}
```

CALL METHOD FROM OTHER PROJECT



Configure your new project

Class Library C# Android Linux macOS Windows Library

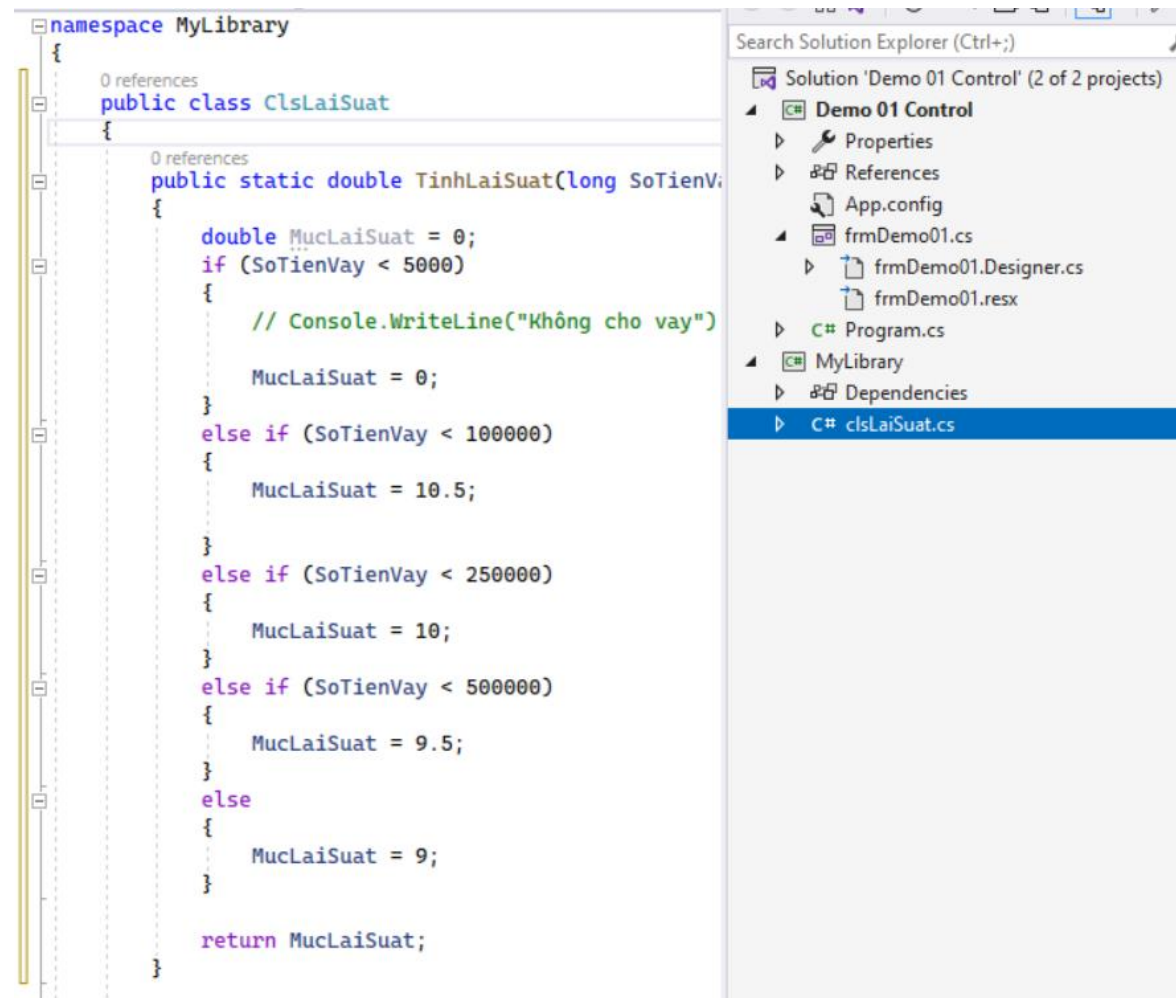
Project name

MyLibrary

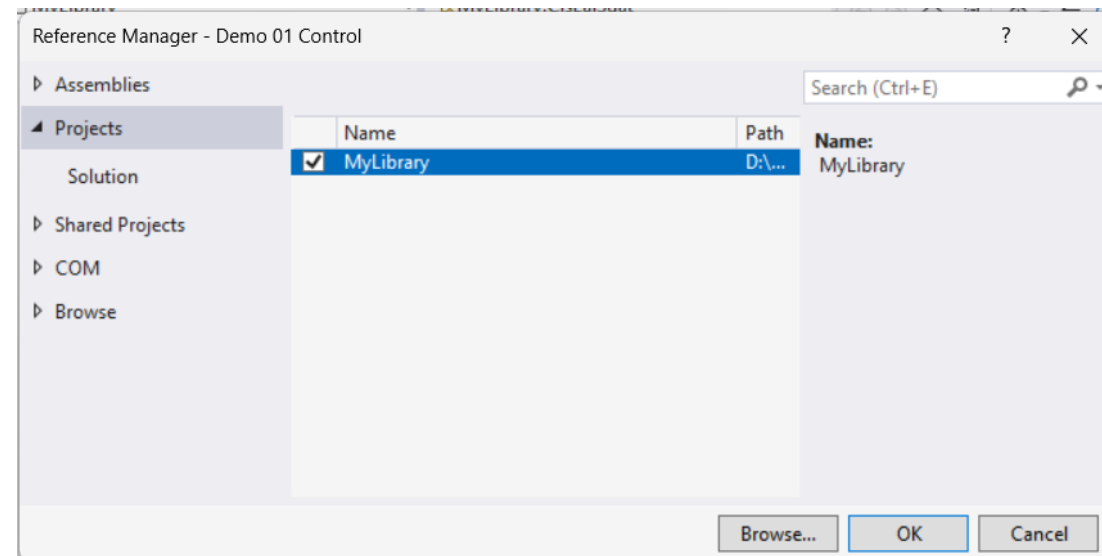
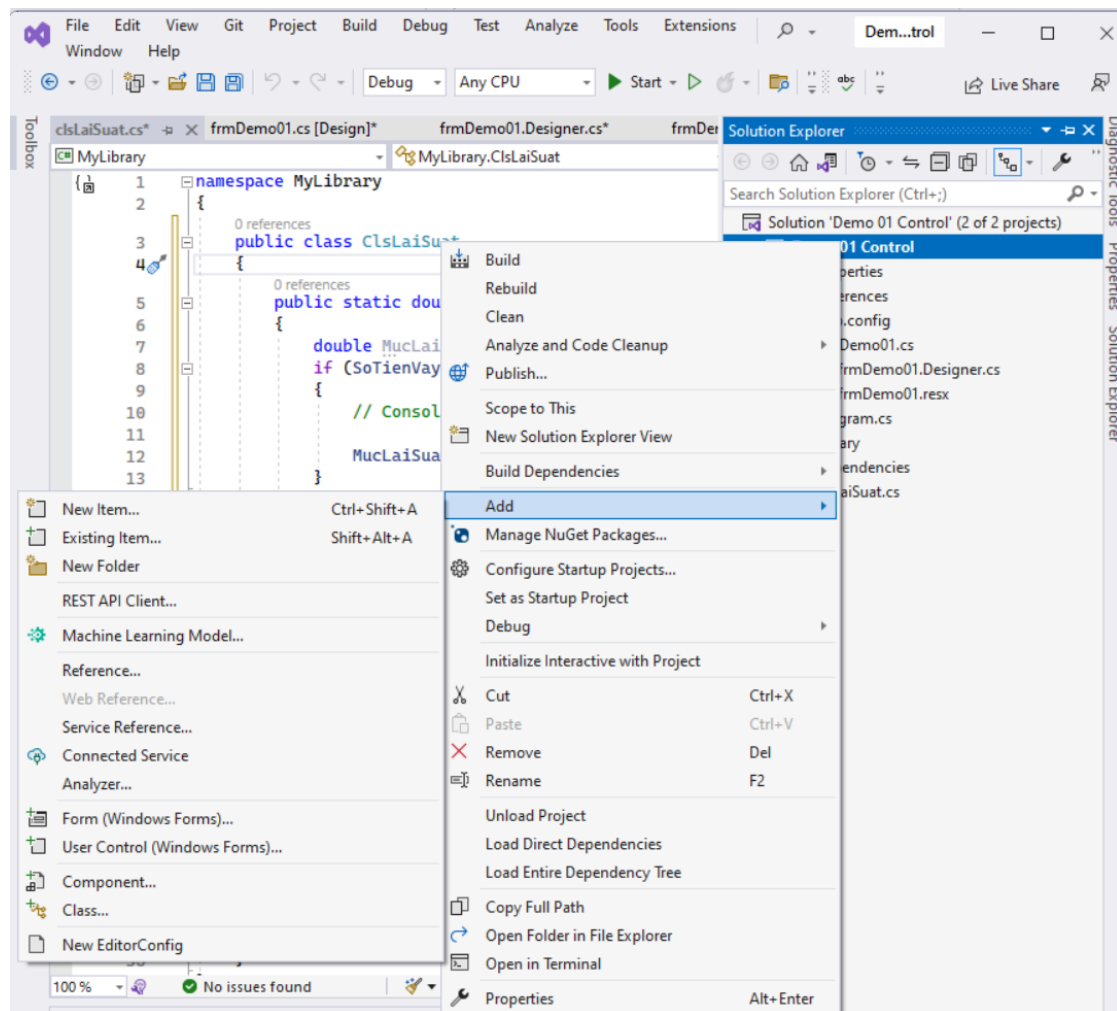
Location

D:\BMCNT\NET.LT.NET\50 WF BasicControl Demo\Demo 01 Control

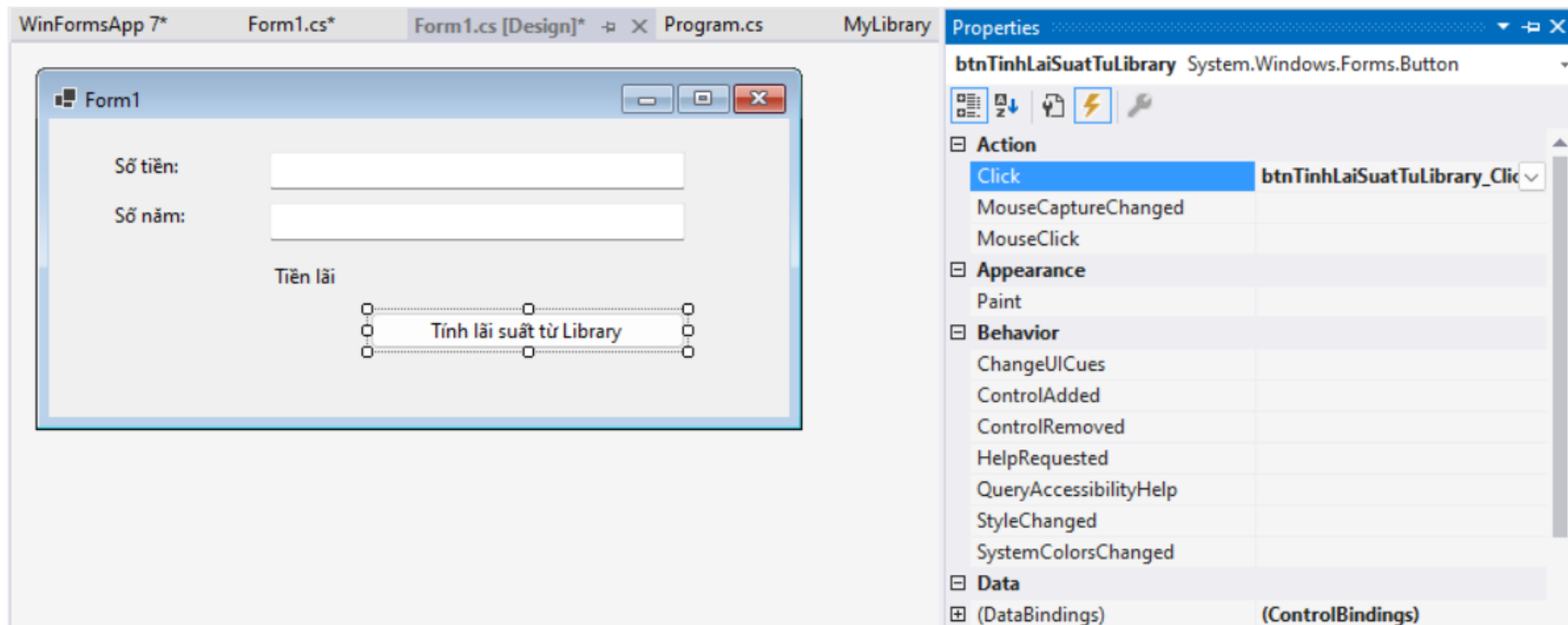
Project will be created in "D:\BMCNT\NET.LT.NET\50 WF BasicControl Demo\Demo 01 Control\MyLibrary"



CALL METHOD FROM OTHER PROJECT



CALL METHOD FROM OTHER PROJECT



1 reference
`private void btnTinhLaiSuatTuLibrary_Click(object sender, EventArgs e)`
{

MyLibrary.

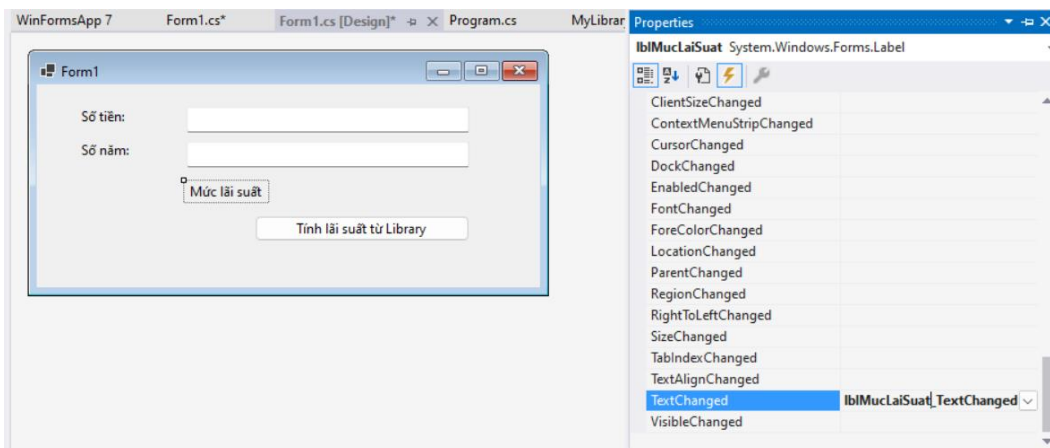
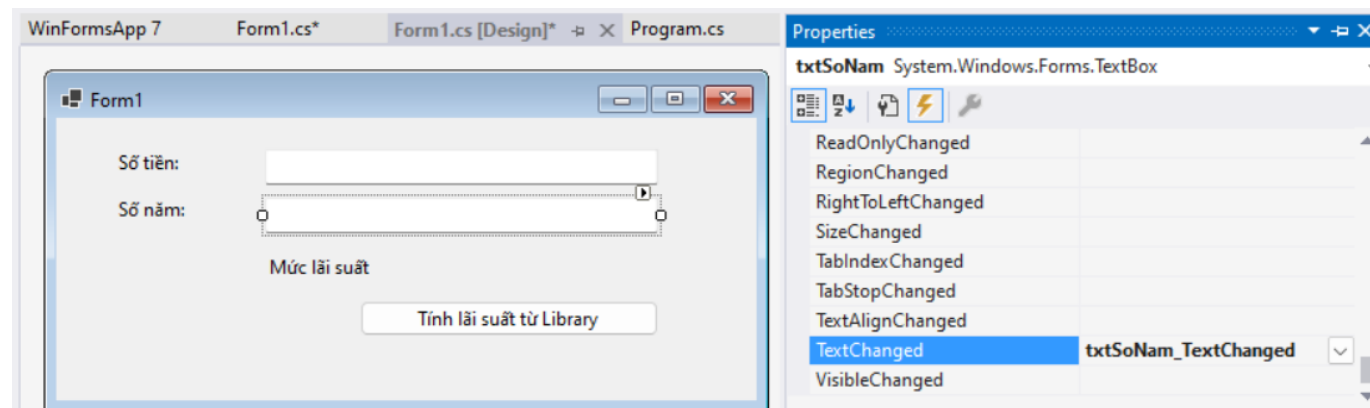
`MyLibrary.ClsLaiSuat` class MyLibrary.ClsLaiSuat

1 reference
`private void btnTinhLaiSuatTuLibrary_Click(object sender, EventArgs e)`
{

```
int sotienvay = int.Parse(this.txtSoTien.Text);  
double muclaisuat = MyLibrary.ClsLaiSuat.TinhLaiSuat(sotienvay);  
this.lblMucLaiSuat.Text = muclaisuat.ToString();
```


EVENTS

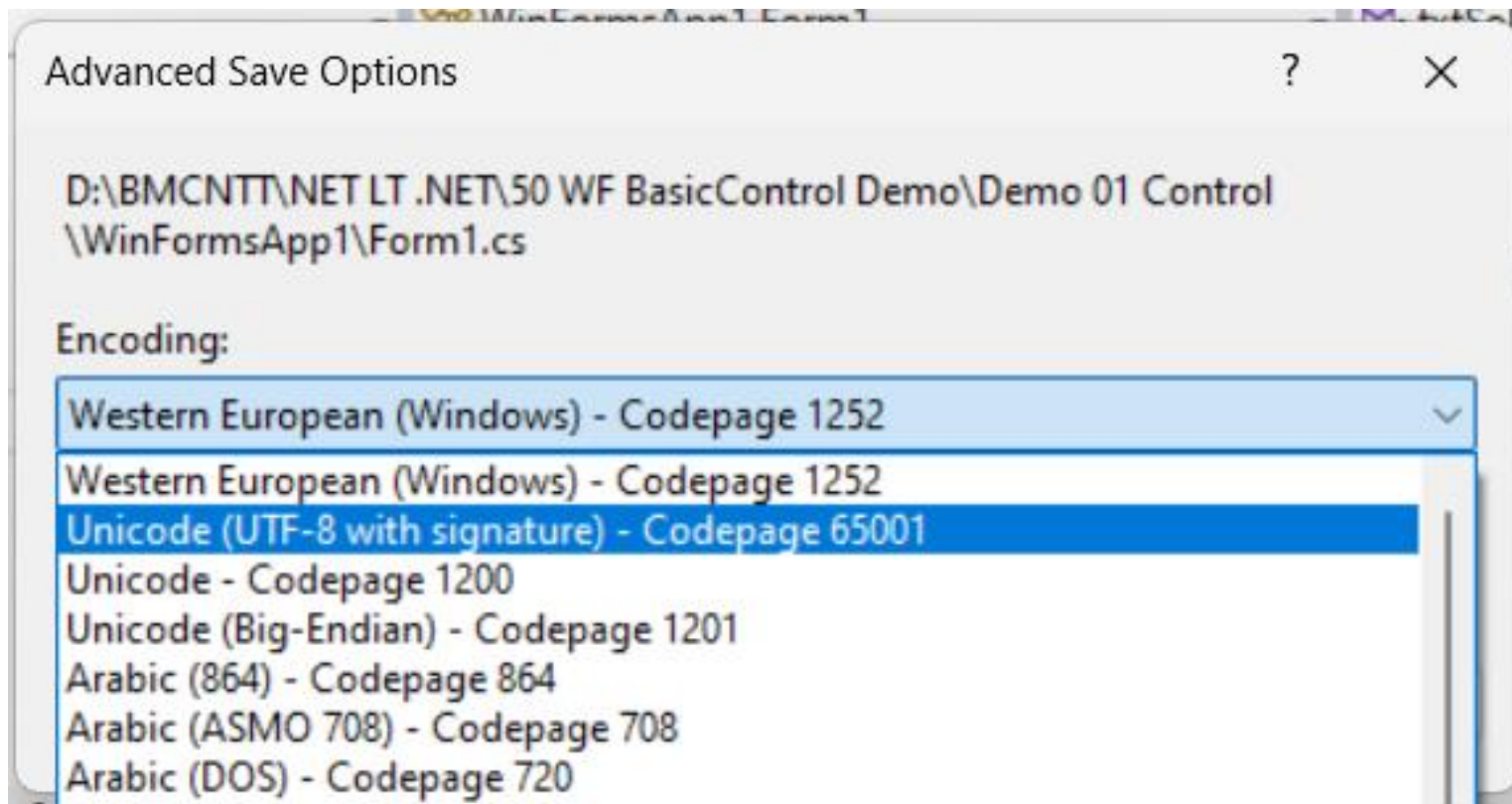
■ Event TextChanged



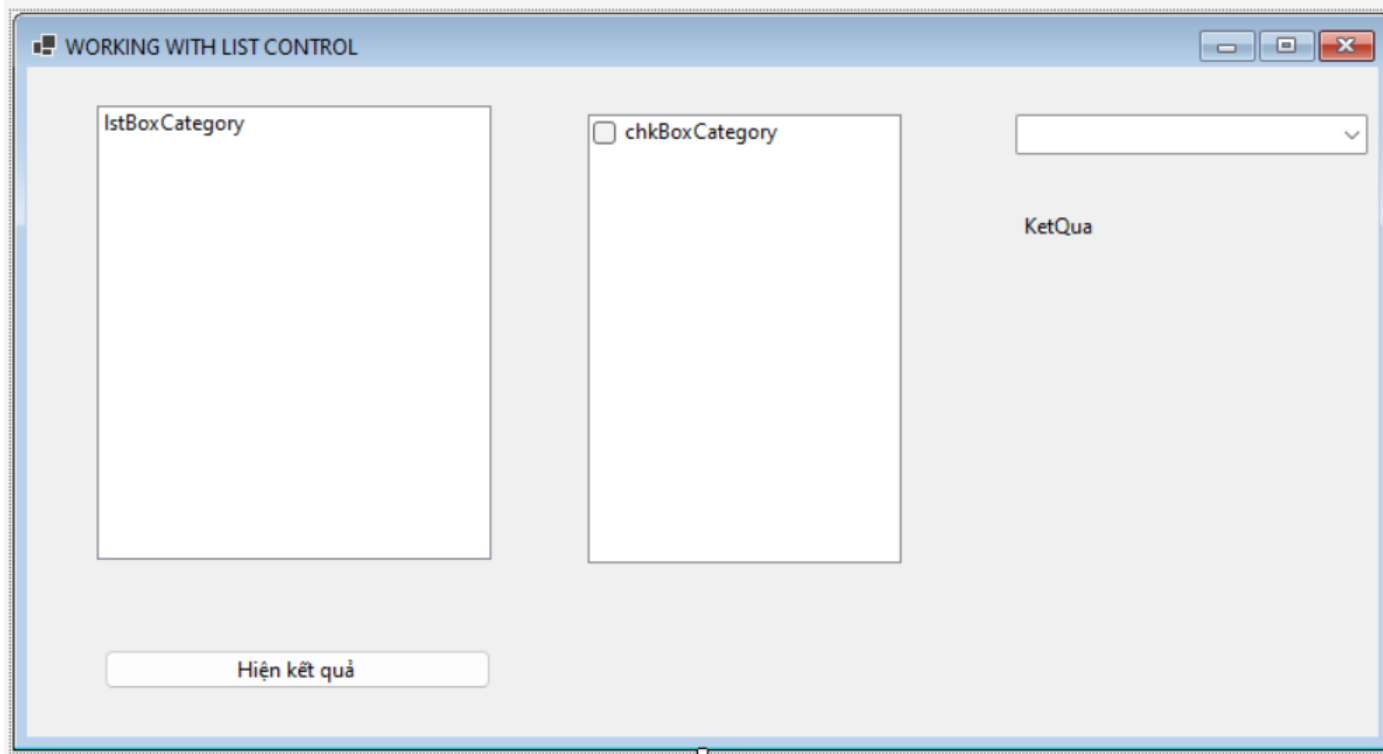
```
1 reference
private void lblMucLaiSuat_TextChanged(object sender, EventArgs e)
{
    MessageBox.Show("Thay đổi");
}

1 reference
private void txtSoNam_TextChanged(object sender, EventArgs e)
{
    //Trường hợp này message box sẽ bật lên mỗi khi người dùng gõ
    //--> Không hợp lý --> Thiết kế khác
    MessageBox.Show("Thay đổi");
    //Có thể thay đổi bằng việc thao tác xử lý rồi hiện vào control khác bên cạnh
}
```

UNICODE



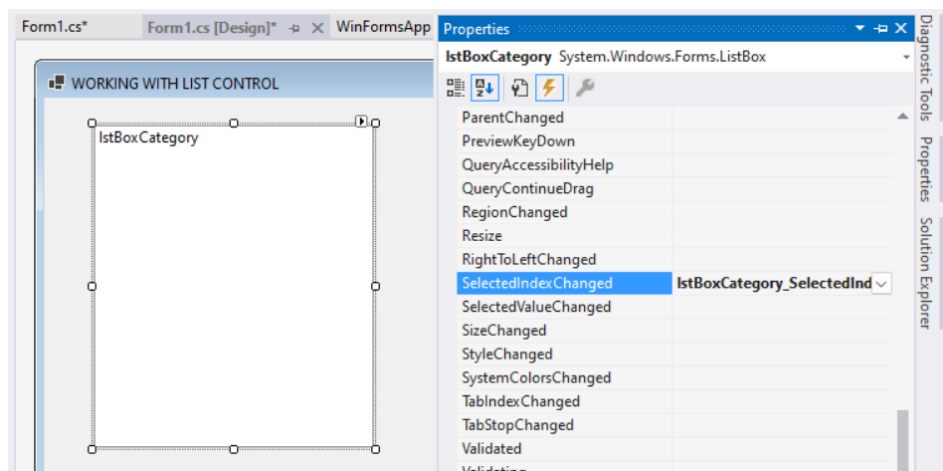
LIST BOX CONTROL



```
1 reference
private void frmListControl_Load(object sender, EventArgs e)
{
    //Load dữ liệu vào Control
    LoadDuLieuListBox();
}
```

```
1 reference
private void LoadDuLieuListBox()
{
    this.lstBoxCategory.Items.Clear();
    this.lstBoxCategory.Items.Add("Phone");
    this.lstBoxCategory.Items.Add("Computer");
    this.lstBoxCategory.Items.Add("IPad");
}
```


LIST BOX CONTROL

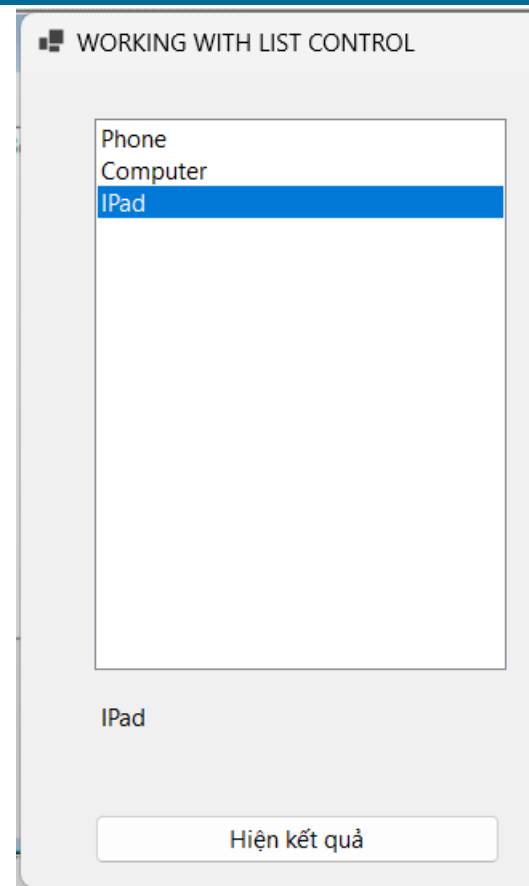


1 reference
`private void LoadDuLieuListBox()`

```
{  
    this.lstBoxCategory.Items.Clear();  
    this.lstBoxCategory.Items.Add("Phone");  
    this.lstBoxCategory.Items.Add("Computer");  
    this.lstBoxCategory.Items.Add("IPad");  
}
```

1 reference
`private void lstBoxCategory_SelectedIndexChanged(object sender, EventArgs e)`

```
{  
    if (lstBoxCategory.SelectedItems.Count > 0)  
    {  
        this.lblKetQua.Text = this.lstBoxCategory.SelectedItem.ToString();  
    }  
}
```



CHECKBOXLIST

1 reference

```
private void frmListControl_Load(object sender, EventArgs e)
{
    //Load dữ liệu vào Control
    LoadDuLieuListBox();
    LoadCheckBoxList();
}
```

1 reference

```
private void LoadCheckBoxList()
{
    this.chkBoxCategory.Items.Clear();
    this.chkBoxCategory.Items.Add("Phone");
    this.chkBoxCategory.Items.Add("Computer");
    this.chkBoxCategory.Items.Add("IPad");
    this.chkBoxCategory.Items.Add("Labtop");
}
```

- ☐ Phone
- ☒ Computer
- ☒ IPad
- ☐ Labtop

Computer,IPad

CHECKBOXLIST

1 reference

```
private void btnKetQua_Click(object sender, EventArgs e)
{
    string results = "";
    if (chkBoxCategory.CheckedItems.Count > 0)
    {
        foreach (var item in this.chkBoxCategory.CheckedItems)
        {
            results += item.ToString() + ",";
        }
        //Bỏ dấu phẩy ở cuối
        results = results.Substring(0, results.Length - 1);
        this.lblKetQuaCheckBox.Text = results;
    }
    else this.lblKetQuaCheckBox.Text = "";
}
```

- ☐ Phone
- ☒ Computer
- ☒ iPad
- ☐ Labtop

Computer,iPad

CHECKBOXLIST

```
private void WhatIsChecked_Click(object sender, System.EventArgs e) {  
    // Display in a message box all the items that are checked.  
  
    // First show the index and check state of all selected items.  
    foreach(int indexChecked in checkedListBox1.CheckedIndices) {  
        // The indexChecked variable contains the index of the item.  
        MessageBox.Show("Index#: " + indexChecked.ToString() + ", is checked. Checked state is:" +  
            checkedListBox1.GetItemCheckState(indexChecked).ToString() + ".");  
    }  
  
    // Next show the object title and check state for each item selected.  
    foreach(object itemChecked in checkedListBox1.CheckedItems) {  
  
        // Use the IndexOf method to get the index of an item.  
        MessageBox.Show("Item with title: \"" + itemChecked.ToString() +  
            "\", is checked. Checked state is: " +  
            checkedListBox1.GetItemCheckState(checkedListBox1.Items.IndexOf(itemChecked)).  
    }  
}
```

<https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.checkedlistbox.items?view=windowsdesktop-7.0>

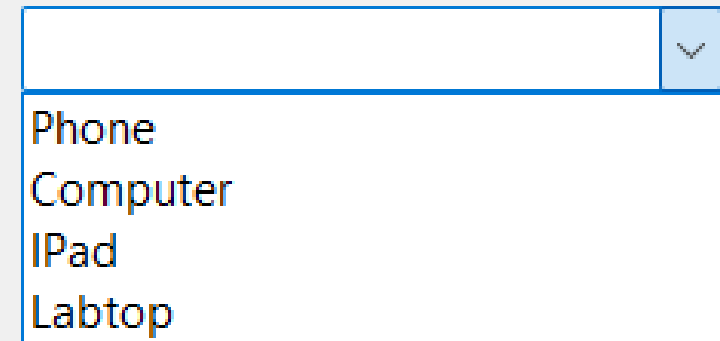
- ☐ Phone
- ☒ Computer
- ☒ iPad
- ☐ Labtop

Computer,iPad

COMBOBOX

1 reference

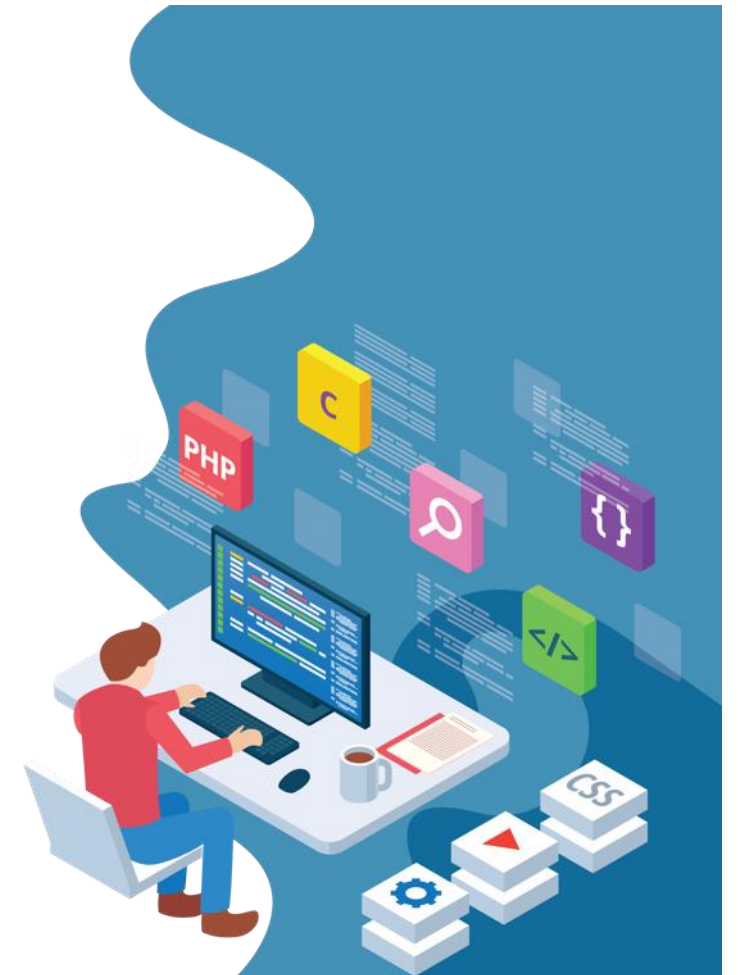
```
private void LoadComboBox()  
{  
    this.cbbBoxCategory.Items.Clear();  
    this.cbbBoxCategory.Items.Add("Phone");  
    this.cbbBoxCategory.Items.Add("Computer");  
    this.cbbBoxCategory.Items.Add("IPad");  
    this.cbbBoxCategory.Items.Add("Labtop");  
}
```



DIALOG BOX CONTROL

Dialog box control

- [ColorDialog](#)
- [FontDialog](#)
- [OpenFileDialog](#)
- [SaveFileDialog](#)
- [PageSetupDialog](#)
- [PrintDialog](#)
- [PrintPreviewDialog](#)



COLOR DIALOG BOX CONTROL

```
private void button1_Click(object sender, System.EventArgs e)
{
    ColorDialog MyDialog = new ColorDialog();
    // Keeps the user from selecting a custom color.
    MyDialog.AllowFullOpen = false ;
    // Allows the user to get help. (The default is false.)
    MyDialog.ShowHelp = true ;
    // Sets the initial color select to the current text color.
    MyDialog.Color = textBox1.ForeColor ;

    // Update the text box color if the user clicks OK
    if (MyDialog.ShowDialog() == DialogResult.OK)
        textBox1.ForeColor = MyDialog.Color;
}
```

AllowFullOpen	Gets or sets a value indicating whether the user can use the dialog box to define custom colors.
AnyColor	Gets or sets a value indicating whether the dialog box displays all available colors in the set of basic colors.
CanRaiseEvents	Gets a value indicating whether the component can raise an event. (Inherited from Component)
Color	Gets or sets the color selected by the user.
Container	Gets the IContainer that contains the Component . (Inherited from Component)
CustomColors	Gets or sets the set of custom colors shown in the dialog box.
DesignMode	Gets a value that indicates whether the Component is currently in design mode. (Inherited from Component)
Events	Gets the list of event handlers that are attached to this Component . (Inherited from Component)
FullOpen	Gets or sets a value indicating whether the controls used to create custom colors are visible when the dialog box is opened.
Instance	Gets the underlying window instance handle (HINSTANCE).

REFERENCES

- <https://learn.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?toc=%2Fvisualstudio%2Fget-started%2Fcsharp%2Ftoc.json&bc=%2Fvisualstudio%2Fget-started%2Fcsharp%2Fbreadcrumb%2Ftoc.json&view=vs-2022>