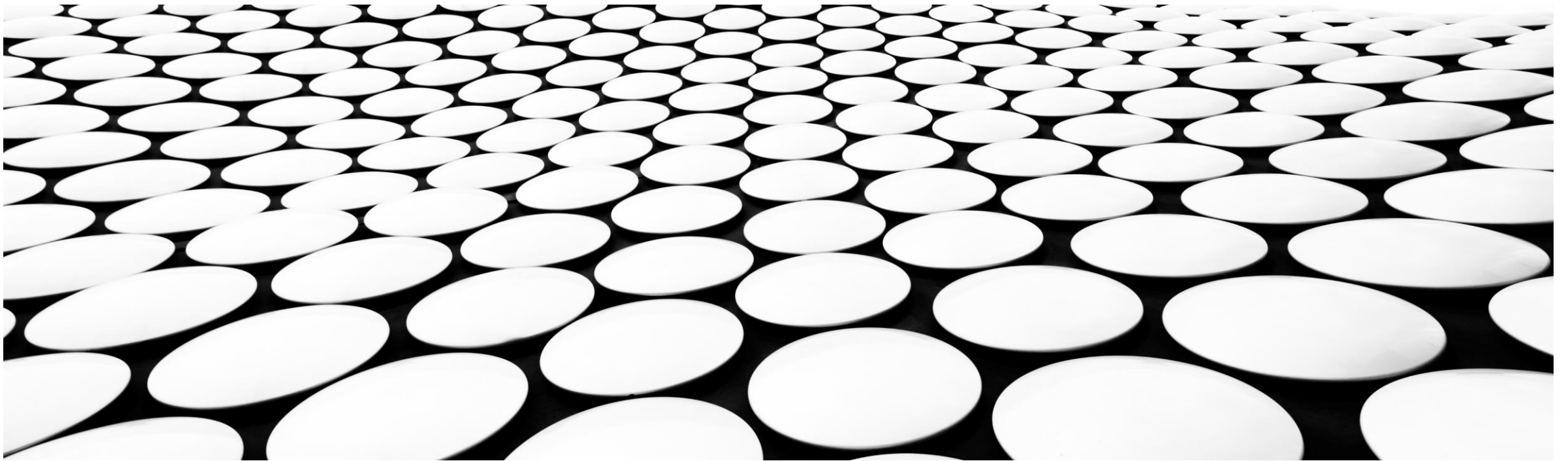

XÂY DỰNG WINDOWS FORMS APPLICATION



5. ListBox (ListBox)

a. Công dụng:

- Dùng để hiển thị một danh sách các lựa chọn.

b. Tạo ListBox:

- Chọn công cụ  ListBox
- rê chuột và vẽ ListBox trên form.

c. Thuộc tính:

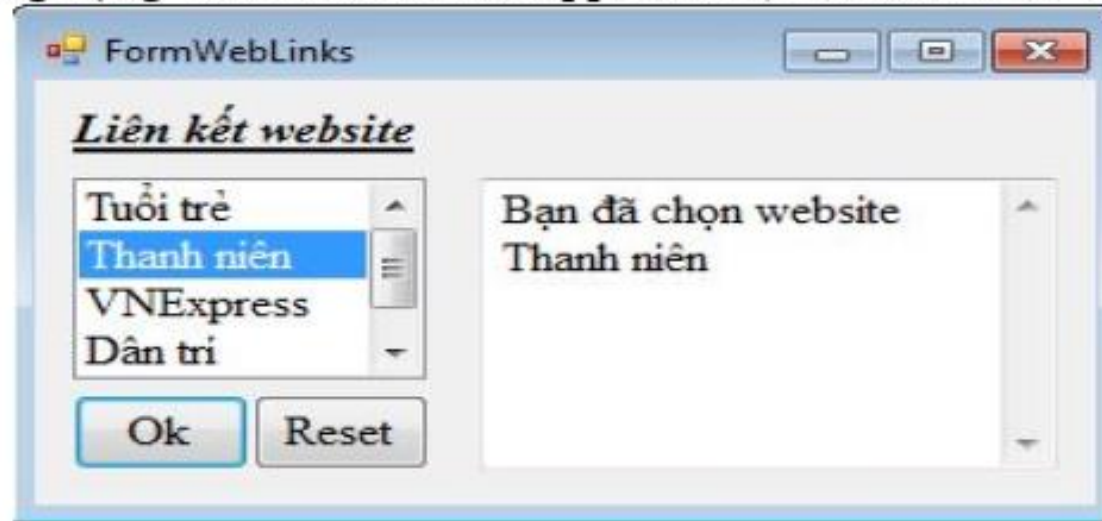
Thuộc tính	Mô tả
Items	Các mục giá trị trong ListBox
SelectedItem	Item được chọn
Phương thức	Mô tả
Add("chuỗi")	Thêm một mục giá trị là "chuỗi"
ToString()	Trả về chuỗi ký tự được chọn

* Nhập giá trị vào ListBox: <Ten_ListBox>.Items.Add ("Chuỗi") ;

* Lấy giá trị trong ListBox: <Ten_ListBox>.SelectedItem.ToString() ;

Ví dụ 4.1:

* Khởi tạo một ứng dụng Windows Forms Application, lưu với tên là Vi Du 4.1 như sau:



* Yêu cầu:

- ListBox **lstWeb** (Liên kết website) chứa các giá trị:
 - + Tuổi trẻ
 - + Thanh niên
 - + VNExpress
 - + Dân trí
 - + Công an
- TextBox **txtKQ** (chứa kết quả) để trống.
- Nhấp button **btnOk** (Ok) sẽ hiện trong txtKQ tên website được chọn ở lstWeb.
- Nhấp button **btnReset** (Reset) sẽ xóa trống txtKQ.

* Hướng dẫn:

- Thiết kế Form như yêu cầu, trong đó form có các thuộc tính sau:

- + AutoSize: True
- + Font: Times New Roman
- + Size: 12
- + Text: FormWebLinks

- Nhấp đúp vào button **Ok** rồi thêm đoạn code sau:

```
this.txtKQ.Text = "Bạn đã chọn website ";  
this.txtKQ.Text += this.lstWeb.SelectedItem.ToString();
```

- Nhấp đúp vào button **Reset** rồi thêm đoạn code sau:

```
this.txtKQ.ResetText();
```

* Cải tiến: Ta có thể đưa các giá trị của lstWeb trong **Form1_Load**.

+ Thiết kế lại form như sau



+ Nhấp đúp chuột vào nền form, rồi gõ đoạn code


```
this.lstWeb.Items.Add("Tuổi trẻ");  
this.lstWeb.Items.Add("Thanh niên");  
this.lstWeb.Items.Add("VNExpress");  
this.lstWeb.Items.Add("Dân trí");  
this.lstWeb.Items.Add("Công an");  
this.lstWeb.SelectedItem = "Tuổi trẻ";
```

6. ComboBox (ComboBox)

a. Công dụng:

- Dùng để hiển thị một danh sách các lựa chọn / hoặc nhập vào một giá trị.

b. Tạo ComboBox:

- Chọn công cụ  ComboBox

- rê chuột và vẽ ComboBox trên form.

c. Thuộc tính:

Thuộc tính	Mô tả
DisplayMember	Gán nội dung thể hiện trên ComboBox
Items	Liệt kê các mục giá trị trong ComboBox
SelectedItem	Lấy Item được chọn
SelectedText	Lấy nội dung thể hiện trên ComboBox từ DisplayMember
SelectedValue	Lấy giá trị từ ValueMember
ValueMember	Gán giá trị cho ComboBox

Ví dụ 4.2:

* Khởi tạo một ứng dụng Windows Forms Application, lưu với tên là Vi Du 4.2 như sau:



* Yêu cầu:

Thực hiện giống như Ví dụ 4.1, nhưng thay ListBox bằng ComboBox *cbWeb*.

7. RadioButton (RadioButton)

a. Công dụng:

- Dùng để chọn một trong các lựa chọn trong danh sách.

b. Tạo RadioButton:

- Chọn công cụ  RadioButton
- rê chuột và vẽ RadioButton trên form.

c. Thuộc tính:

Thuộc tính	Mô tả
Checked	Không có dấu chọn (False) / Có dấu chọn (True)

8. GroupBox (GroupBox)

a. Công dụng:

- Tạo ra một nhóm.

b. GroupBox:

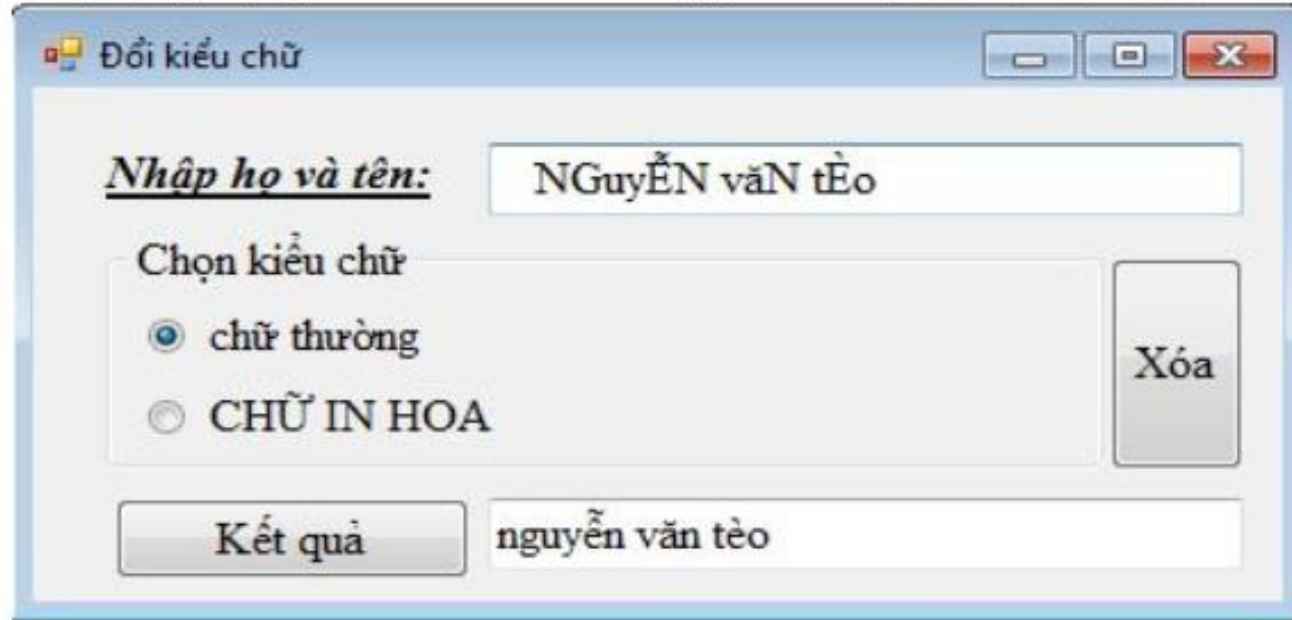
- Chọn công cụ  GroupBox
- rê chuột và vẽ GroupBox trên form.

c. Thuộc tính:

Thuộc tính	Mô tả
BackgroundImage	Hình nền
BackgroundImageLayout	None / Tile / Center / Stretch / Zoom

Ví dụ 4.3:

* Khởi tạo một ứng dụng Windows Forms Application, lưu với tên là Vi Du 4.3 như sau:



* Yêu cầu:

- Nhập họ và tên vào TextBox txtHoTen
- Chọn kiểu chữ
 - + Radio Button (rad1): chữ thường
 - + Radio Button (rad2): chữ HOA
- Nhấp vào Button Kết quả (btnKQ) sẽ in họ và tên bằng kiểu chữ được chọn trong TextBox txtKQ.
- Nhấp vào Button Xóa (btnXoa) sẽ xóa trống txtHoTen, txtKQ, rad1 được chọn và đưa con trỏ vào ô TextBox txtHoTen.

- Nhấp đúp vào nút **Kết quả** rồi thêm đoạn code sau:

```
string hoten=this.txtHoTen.Text.Trim();  
if (this.rad1.Checked == true)  
    txtKQ.Text = hoten.ToLower();  
if (this.rad2.Checked == true)  
    txtKQ.Text = hoten.ToUpper();
```

- Nhấp đúp vào nút **Xóa** rồi thêm đoạn code sau:

```
this.txtHoTen.Clear();  
this.txtKQ.Clear();  
this.rad1.Checked = true;  
this.txtHoTen.Focus();
```


9. Timer (🕒 Timer)

a. Công dụng:

- Quy định khoảng thời gian định kỳ để thực hiện một công việc.

b. Tạo Timer:

- Chọn công cụ 🕒 Timer

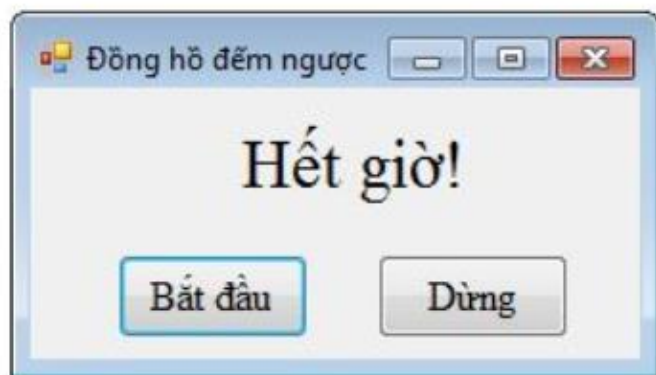
- rê chuột và vẽ Timer → là control dạng invisible (ẩn).

c. Thuộc tính:

Thuộc tính	Mô tả
Enabled	Bật / tắt chế độ hẹn thời gian
Interval	Khoảng thời gian định kỳ

Ví dụ 4.4:

* Khởi tạo một ứng dụng Windows Forms Application, lưu với tên là Vi Du 4.4 như sau:



* Yêu cầu:

- Tạo Timer1 có Enabled = false; Interval = 1000

- Khởi tạo biến đếm i = 20

- Button Bắt đầu (btnBatDau): dùng để bật chế độ hẹn thời gian.

- Button Dừng (btnDung): dùng để dừng chương trình.

- Timer1: Đếm ngược từ 20 đến 1 và xuất ra dòng chữ “Hết giờ”.

+ Xuất giá trị biến đếm i ra Label lblDongHo.

+ Giảm biến i xuống 1 đơn vị.

+ Khi biến đếm i < 0 thì tắt chế độ hẹn giờ và xuất “Hết giờ!” ra Label lblDongHo.

- Khai báo biến đếm i: qua code, thêm đoạn code để được kết quả như sau:

```
public Form1 ()  
{  
    InitializeComponent();  
}  
int i = 10;
```

- Nhấp đúp vào nút **Bắt đầu** rồi thêm đoạn code sau:

```
this.timer1.Enabled = true;
```

- Nhấp đúp vào nút **Dừng** rồi thêm đoạn code sau:

```
Application.Exit();
```

- Nhấp đúp vào nút **Timer1** (Timer1_Tick) rồi thêm đoạn code sau:

```
this.lblDongHo.Text = i.ToString();  
i--;  
if (i < 0)  
    this.timer1.Enabled = false;
```

10. RichTextBox (RichTextBox)

a. Công dụng:

- Dùng để nhập văn bản với định dạng văn bản đa dạng.

b. Tạo RichTextBox:

- Chọn công cụ  RichTextBox
- rê chuột và vẽ RichTextBox trên form.

c. Thuộc tính:


Thuộc tính	Mô tả
ReadOnly	Không cho soạn thảo, chỉ đọc.

11. Panel (Panel)

a. Công dụng:

- Bảng chứa các control hay một nhóm các control.

b. Tạo Panel:

- Chọn công cụ  Panel
- rê chuột và vẽ Panel.

c. Thuộc tính:


Thuộc tính	Mô tả
AutoScroll	Tự động cuộn nếu số control nằm ngoài vùng
BorderStyle	None / FixedSingle / Fixed3D

12. PictureBox (PictureBox)

a. Công dụng:

- Khung chứa hình ảnh.

b. Tạo PictureBox:

- Chọn công cụ  PictureBox
- rê chuột và vẽ PictureBox.

c. Thuộc tính:

Thuộc tính	Mô tả	Image	Hình chứa trong PictureBox
------------	-------	-------	----------------------------


III. Menu và ToolBar

1. MenuStrip (MenuStrip)

a. Công dụng:

- Tạo menu.

b. Tạo MenuStrip:


- Nhấp đúp vào control  MenuStrip.
- Nhập menu.

2. ToolStrip (ToolStrip)

a. Công dụng:

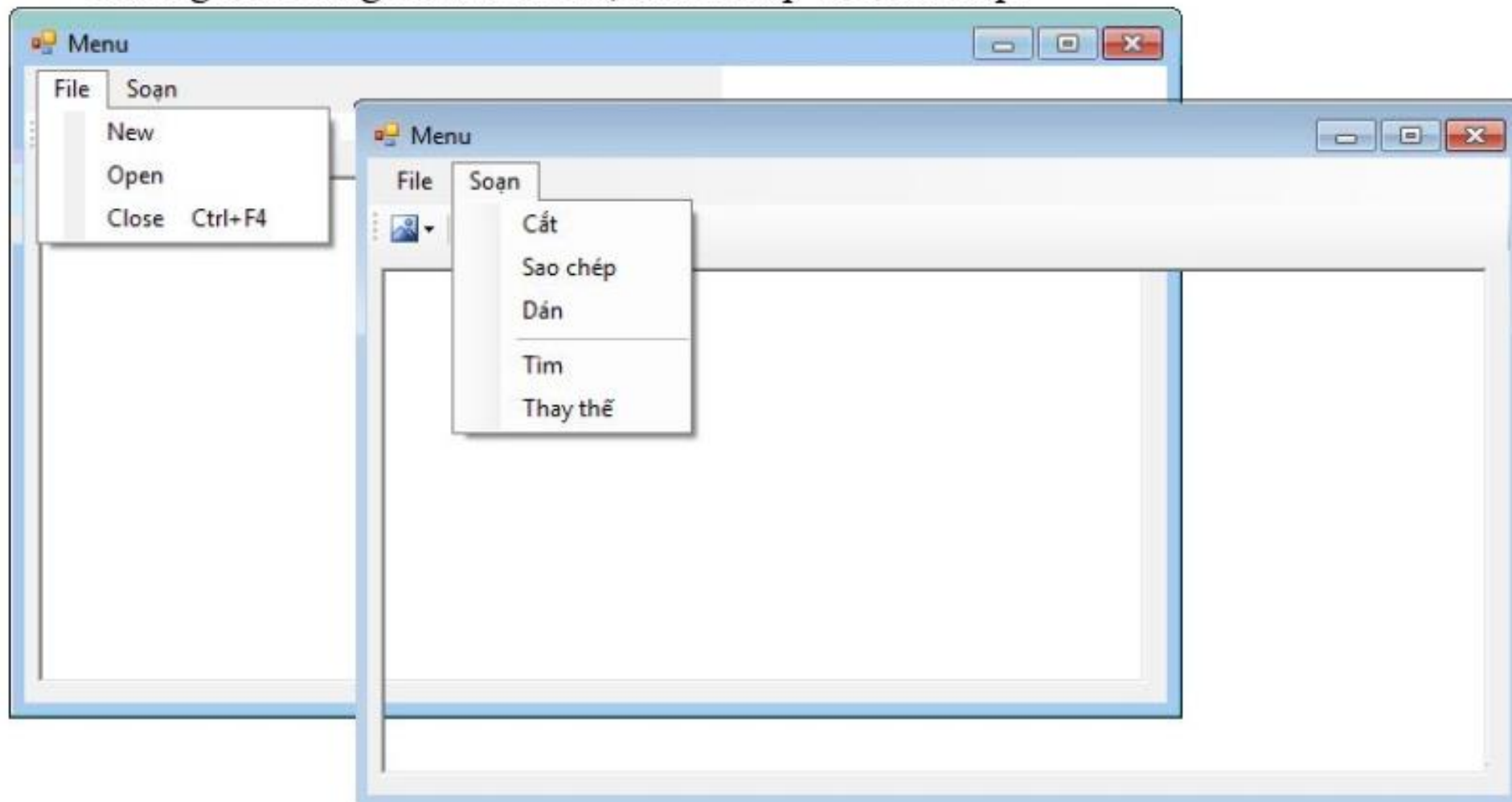
- Tạo ra toolbar.

b. Tạo ToolStrip:

- Chọn công cụ  ToolStrip
- Kéo control ToolStrip vào trong form.
- Bấm vào ToolStrip vừa tạo, ta có thể tạo ra các tool (công cụ) như: Label, TextBox, Button, ComboBox, DropDownButton....

Ví dụ 4.5:

- * Tạo ứng dụng Windows Forms Application như hình.
- * Hướng dẫn: dùng RichTextBox, MenuStrip và ToolStrip.



IV. Common Dialog (Hộp thoại dùng chung)

1. **FontDialog** (FontDialog)

a. **Công dụng:**

- Tạo ra hộp thoại Font.

b. **Tạo FontDialog:**

- Nhấp đúp vào control  FontDialog

2. **OpenFileDialog** (OpenFileDialog)

a. **Công dụng:**

- Tạo ra hộp thoại Open File.

b. **Tạo OpenFileDialog:**


- Nhấp đúp vào control  OpenFileDialog

3. **SaveFileDialog** (SaveFileDialog)

a. **Công dụng:**

- Tạo ra hộp thoại Save File.

b. **Tạo SaveFileDialog:**

- Nhấp đúp vào control  SaveFileDialog

4. **PrintPreviewDialog** (PrintPreviewDialog)

a. **Công dụng:**

- Xem trước khi in.

b. **Tạo SaveFileDialog:**

- Nhấp đúp vào control  PrintPreviewDialog

5. **PrintDialog** (PrintDialog)

a. **Công dụng:**

- Tạo ra hộp thoại Print File.

b. **Tạo PrintDialog:**


- Nhấp đúp vào control  PrintDialog

6. **FolderBrowserDialog** (FolderBrowserDialog)

a. **Công dụng:**

- Tạo ra hộp thoại Browser.

b. **Tạo FolderBrowserDialog:**

- Nhấp đúp vào control  FolderBrowserDialog


```
//Xóa
private void toolStripButton2_Click(object sender, EventArgs e)
{
    richTextBox1.Clear();
}
//Thoát
private void toolStripButton3_Click(object sender, EventArgs e)
{
    DialogResult ok;
    ok = MessageBox.Show("Bạn có muốn thoát không?", "Question", MessageBoxButtons.OKCancel, MessageBoxIcon.Stop);
    if (ok == DialogResult.OK)
        Application.Exit();
}
//Mở file đọc ra richTextBox
private void toolStripButton1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        richTextBox1.LoadFile(openFileDialog1.FileName, RichTextBoxStreamType.PlainText);
    }
}
//menu đóng
private void closeToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult ok;
    ok = MessageBox.Show("Bạn có muốn thoát không?", "Question", MessageBoxButtons.OKCancel, MessageBoxIcon.Stop);
    if (ok == DialogResult.OK)
        Application.Exit();
}
```

```
//Mở file có filter
```

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

```
{
```

```
    openFileDialog1.Filter = "Text Files|.txt|Doc Files|.docx|All Files|*.*";
```

```
    openFileDialog1.Title = "Chọn các file";
```

```
    //Cho phép chọn nhiều file cùng lúc - Mặc định là false
```

```
    openFileDialog1.Multiselect = false;
```

```
    //Mở hộp thoại
```

```
    openFileDialog1.ShowDialog();
```

```
    //Lấy giá trị
```

```
    richTextBox1.LoadFile(openFileDialog1.FileName, RichTextBoxStreamType.PlainText);
```

```
}
```

```
//Lưu file
```

```
private void toolStripButton6_Click(object sender, EventArgs e)
```

```
{
```

```
    saveFileDialog1.DefaultExt = "txt";
```

```
    saveFileDialog1.Filter = "RichTextFile |*.rtf|Text file (*.txt)|*.txt|XML file (*.xml)|*.xml|All files (*.*)|*.*";
```

```
    saveFileDialog1.AddExtension = true;
```

```
    saveFileDialog1.RestoreDirectory = true;
```

```
    saveFileDialog1.Title = "Where do you want to save the file?";
```

```
    saveFileDialog1.InitialDirectory = @"C:/";
```

```
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
```

```
    {
```

```
        MessageBox.Show("You selected the file: " + saveFileDialog1.FileName);
```

```
    }
```

```
    saveFileDialog1.Dispose();
```

```
    saveFileDialog1 = null;
```

```
}
```

```
}
```

```
{
```

5. ỨNG DỤNG MDI

- Ứng dụng SDI (Single Document Interface): Paint, NotePad . . .
- Ứng dụng MDI (Multiple Document Interface)
 - Ứng dụng có các **cửa sổ chính** (parent form), các **cửa sổ con** (child form) và có thể có các **form bình thường** khác
 - Các cửa sổ con luôn nằm trong giới hạn của cửa sổ cha. Khi đóng cửa sổ cha, tất cả các cửa sổ con đều đóng

5. ỨNG DỤNG MDI

- Tạo parent form: tạo form → thiết lập thuộc tính IsMdiContainer = true
- Tạo form con
- Code Node trong Form cha gọi Form con:

```
Form fr=new Form7();  
fr.MdiParent = this;  
fr.Show();
```

