

C# Programming

Các điều khiển nâng cao của C#

By Hoàng Hữu Việt

Email: viethh@vinhuni.edu.vn

Viện Kỹ thuật và Công nghệ, Đại học Vinh

Cao Thanh Sơn

Email: sonct@vinhuni.edu.vn

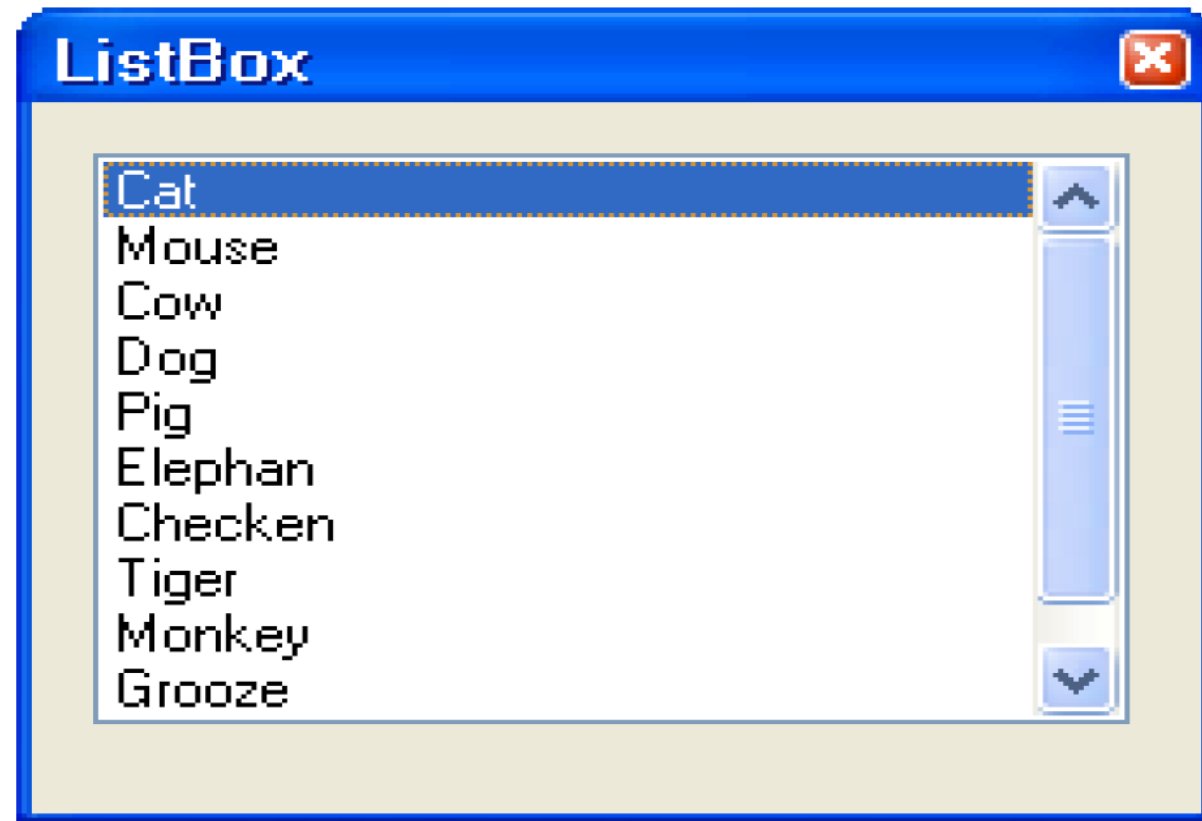
Vinh, 9/2020

Mục đích, chuẩn đầu ra và nội dung

- Mục đích
 - ❑ Giới thiệu về các điều khiển nâng cao để xây dựng các giao diện của ứng dụng
- Chuẩn đầu ra
 - ❑ Lập trình điều khiển nâng cao để xây dựng các giao diện của các ứng dụng
 - ❑ Thiết kế và lập trình các giao diện người dùng của ứng dụng
 - ❑ Phát triển kỹ năng lập trình, kỹ năng tìm kiếm và đọc hiểu tài liệu
- Nội dung
 - ❑ Điều khiển `ListBox`, `CheckedListBox`
 - ❑ Điều khiển `ComboBox`
 - ❑ Điều khiển `TabControl`
 - ❑ Điều khiển `Menu`, `ToolBar`

Điều khiển ListBox

- Dùng để tạo các đối tượng cho phép xem và chọn các dòng dữ liệu (items) từ một danh sách (list)



Điều khiển ListBox

■ Các thuộc tính thường dùng

- ❑ Items: Danh sách các dòng dữ liệu (Items[0] = "Cat", Items[1] = "Mouse", v.v).
- ❑ MultiColumn: Có/không chia ListBox thành nhiều cột.
- ❑ SelectedIndex: Số thứ tự của dòng được chọn. Nếu không có dòng nào được chọn, giá trị là -1.
- ❑ SelectedIndices: Trả về một mảng số thứ tự của các dòng được chọn.
- ❑ SelectedItem: Trả về dữ liệu của dòng được chọn.
- ❑ SelectedItems: Trả về một mảng dữ liệu của các dòng được chọn.
- ❑ SelectionMode: Thiết lập chế độ chọn số dòng.
- ❑ Sorted: Có/không sắp xếp tăng dần các dòng dữ liệu.

Điều khiển ListBox

■ Các phương thức thường dùng

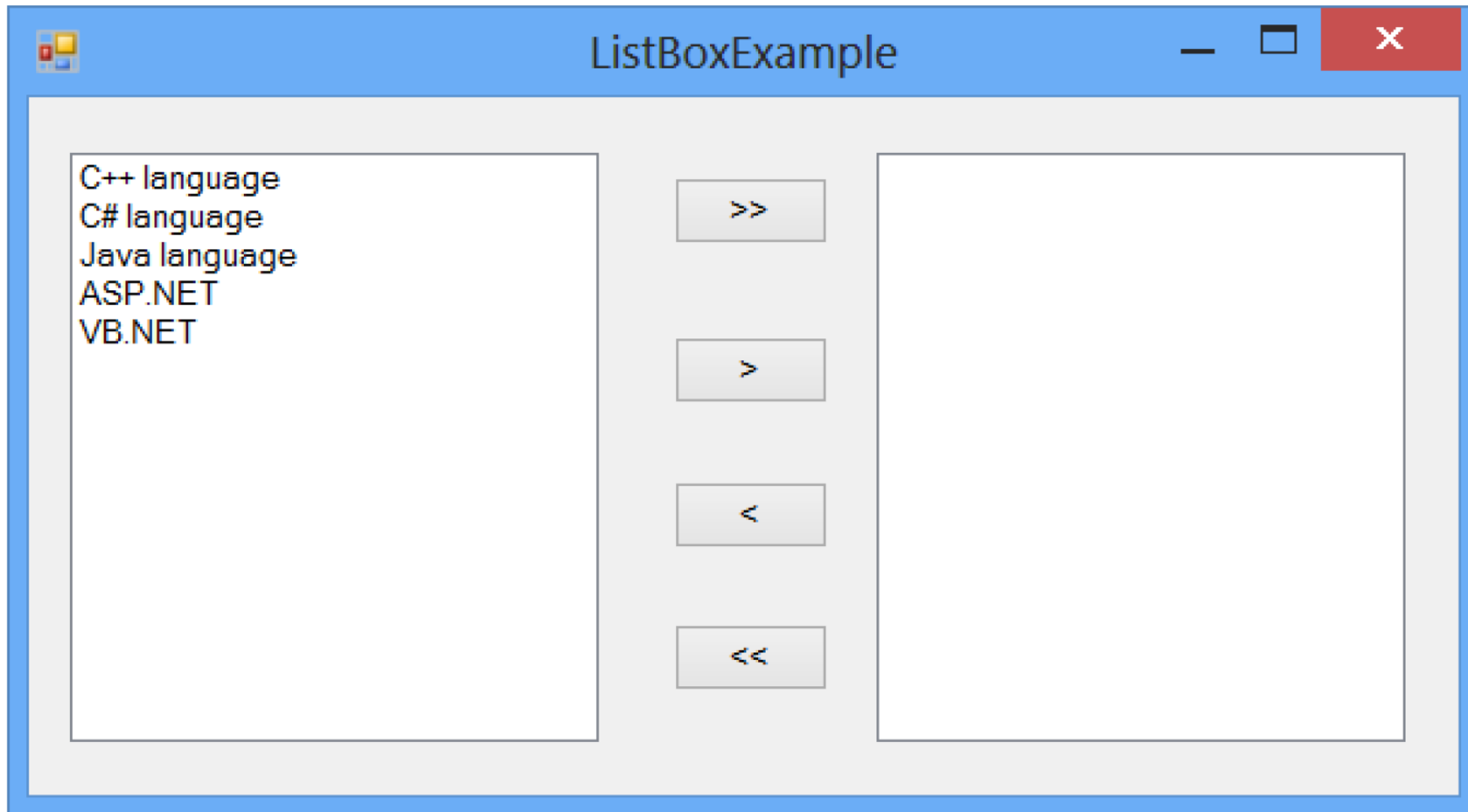
- ❑ `Add()`: Dùng để thêm một dòng dữ liệu vào đối tượng.
- ❑ `Clear()`: Dùng để xóa tất cả các dòng dữ liệu trong đối tượng.
- ❑ `GetSelected()`: Trả về `True` nếu số thứ tự (`index`) của dòng trong tham số của phương thức được chọn.
- ❑ `RemoveAt()`: Dùng để xóa một dòng dữ liệu trong đối tượng ở vị trí chỉ ra.

■ Sự kiện thường dùng

- ❑ `SelectedIndexChanged`: Sự kiện xảy ra khi chọn một dòng trong đối tượng.

Điều khiển ListBox

- Ví dụ xây dựng một form để chuyển dữ liệu giữa 2 đối tượng lớp ListBox:

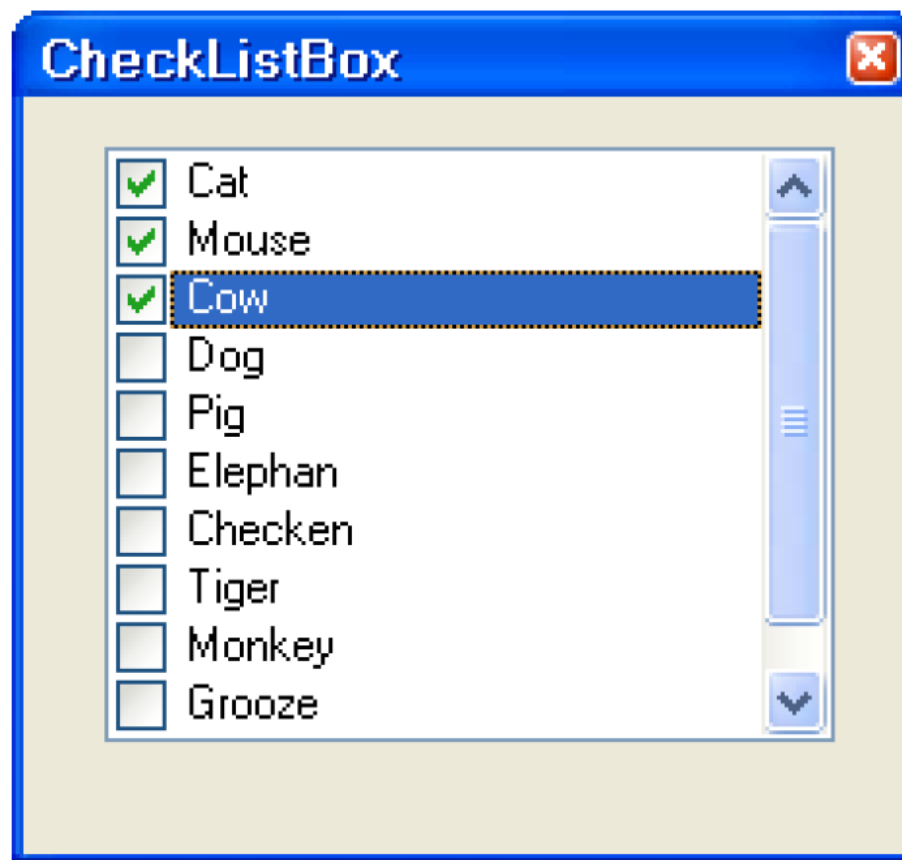


Điều khiển ListBox

```
private void Form1_Load(object sender, EventArgs e)
{
    listBox1.Items.Add("C++ language");
    ...
}
private void btnAddAll_Click(object sender, EventArgs e)
{
    for (int i = 0; i < listBox1.Items.Count; i++)
        listBox2.Items.Add(listBox1.Items[i]);
    listBox1.Items.Clear();
}
private void btnAdd_Click(object sender, EventArgs e)
{
    if (listBox1.SelectedIndex == -1) return;
    listBox2.Items.Add(listBox1.SelectedItem);
    listBox1.Items.RemoveAt(listBox1.SelectedIndex);
}
```

Điều khiển CheckListBox

- Mở rộng của điều khiển ListBox bằng cách bổ sung thêm một điều khiển CheckBox ở mỗi dòng dữ liệu.
- Cho phép chọn đồng thời nhiều dòng dữ liệu.



Điều khiển `CheckedListBox`

■ Các thuộc tính thường dùng

- ❑ `CheckedItems`: Mảng giá trị của các dòng dữ liệu được chọn trong đối tượng.
- ❑ `CheckedIndices`: Mảng chỉ số của các dòng dữ liệu được chọn trong đối tượng.
- ❑ `SelectionMode`: Thiết lập chế độ chọn số dòng.

■ Phương thức thường dùng

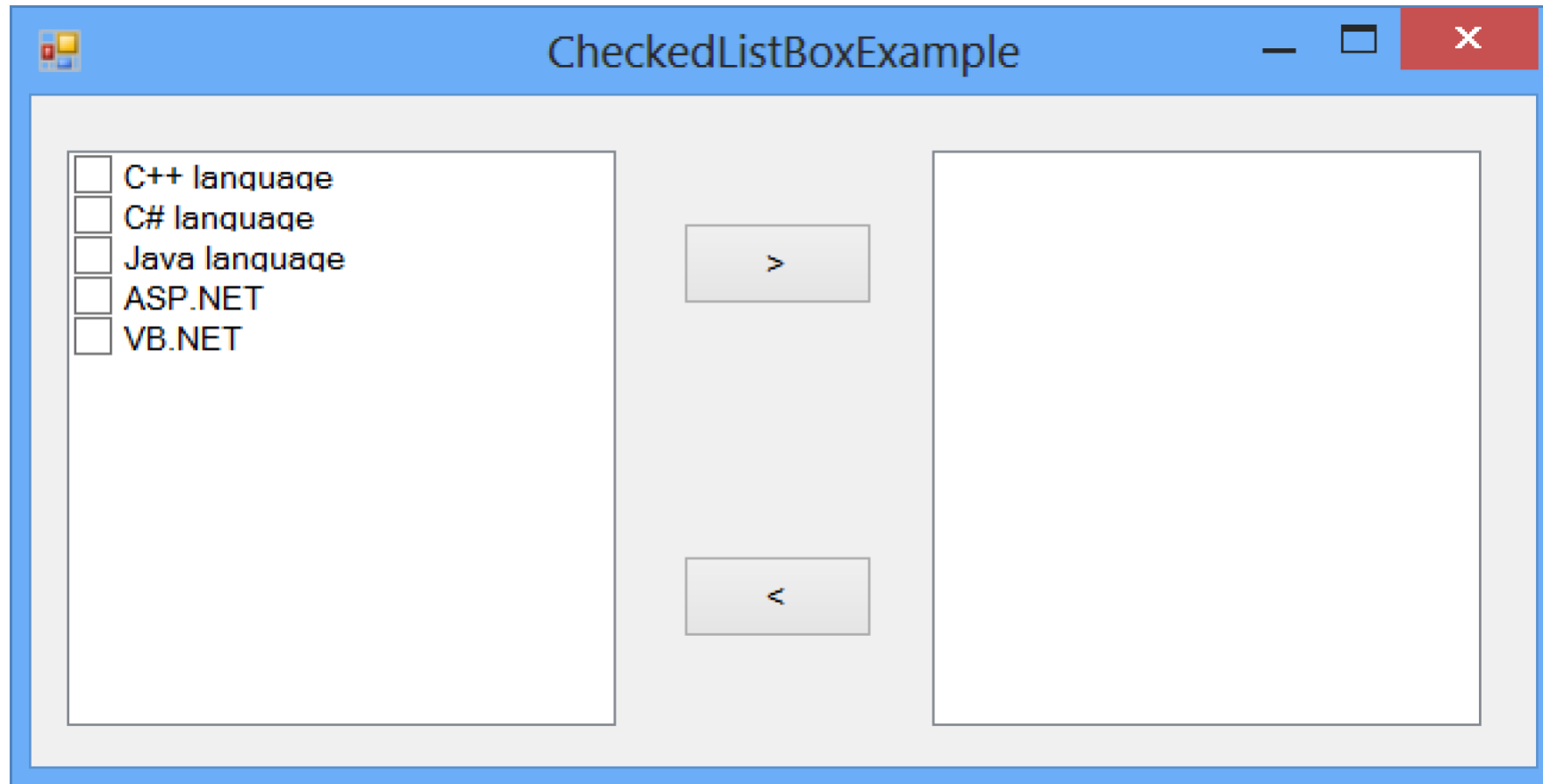
- ❑ `GetItemChecked()`: Trả về `True` nếu số thứ tự (`index`) dòng chỉ ra trong tham số của phương thức được chọn.

■ Sự kiện thường dùng

- ❑ `ItemCheck()`: Được gọi khi chọn/hủy chọn một dòng của đối tượng.

Điều khiển CheckedListBox

- Ví dụ xây dựng một form để chuyển dữ liệu giữa 2 đối tượng lớp CheckedListBox:

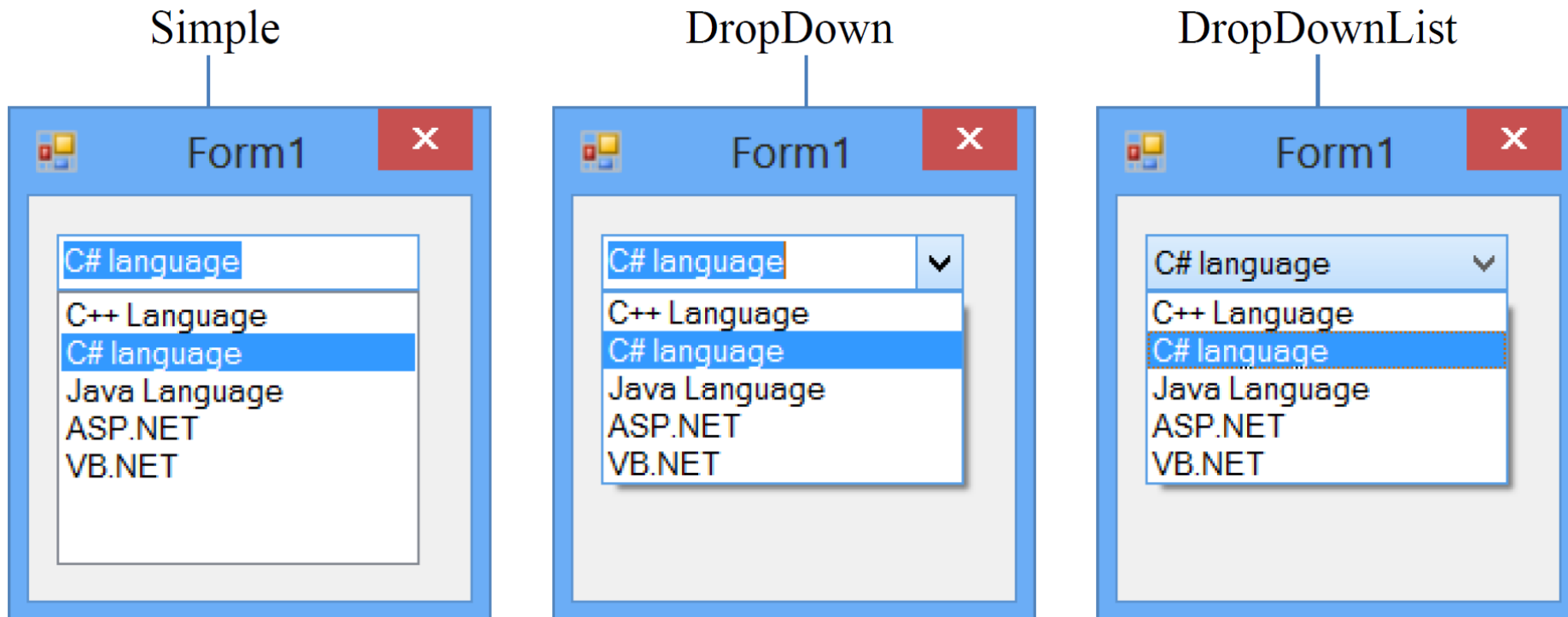


Điều khiển CheckedListBox

```
private void Form1_Load(object sender, EventArgs e)
{
    listBox1.Items.Add("C++ language");
    ...
}
private void btnAdd_Click(object sender, EventArgs e)
{
    for (int i = 0; i < checkedListBox1.Items.Count; i++)
    {
        if (checkedListBox1.GetItemChecked(i) == true)
            checkedListBox2.Items.Add(checkedListBox1.Items[i]);
    }
    for (int i = checkedListBox1.Items.Count - 1; i >= 0 ; i--)
    {
        if (checkedListBox1.GetItemChecked(i) == true)
            checkedListBox1.Items.RemoveAt(i);
    }
}
```

Điều khiển ComboBox

- Cho phép chọn các dòng dữ liệu từ một danh sách.
- Hiển thị như một đối tượng lớp TextBox với một nút chọn ở bên phải.
- Có thể gõ dữ liệu ở vùng TextBox hoặc dùng con trỏ chuột để chọn dòng dữ liệu.



Điều khiển ComboBox

■ Các thuộc tính thường dùng

- ❑ `DropDownStyle`: Kiểu hiển thị của đối tượng.
- ❑ `Items`: Danh sách các dòng dữ liệu.
- ❑ `SelectedIndex`: Trả về số thứ tự của dòng được chọn.
- ❑ `SelectedItem`: Trả về dữ liệu của dòng được chọn.

■ Các phương thức thường dùng

- ❑ `Add()`: Thêm một dòng dữ liệu.
- ❑ `RemoveAt()`: Xóa một dòng dữ liệu ở vị trí chỉ ra.
- ❑ `Clear()`: Xóa tất cả các dòng dữ liệu.

■ Sự kiện thường dùng

- ❑ `SelectedIndexChanged()`: Được gọi khi chọn một dòng trong đối tượng.

Điều khiển ComboBox

- Ví dụ xây dựng một form để thay đổi phong chữ, kiểu chữ và cỡ chữ của một đối tượng lớp Label.

The screenshot shows a Windows application window titled "ComboBoxExample". Inside the window, there are three ComboBox controls arranged horizontally. The first is labeled "Font:" and contains a list of fonts including Arial, Aharoni, Aldhabi, Algerian, Andalus, Angsana New, AngsanaUPC, Aparajita, Arabic Typesetting, Arial (selected), Arial Black, Arial Narrow, Arial Unicode MS, and Baskerville Old Face. The second is labeled "Font Style:" and contains a list of styles including Bold (selected), Regular, Italic, and Bold Italic. The third is labeled "Size:" and contains a list of font sizes from 8 to 20, with 10 selected. Below these three controls is a "Sample" label that displays the text "AaBbCcDdYyZz" in the selected font (Arial), style (Bold), and size (10).

Điều khiển ComboBox

```
private void Form1_Load(object sender, EventArgs e)
{
    FontFamily[] ff = FontFamily.Families;
    for (int i = 0; i < ff.Length; i++)
    {
        cboFont.Items.Add(ff[i].Name);
    }
    cboStyle.Items.Add("Regular");
    cboStyle.Items.Add("Italic");
    cboStyle.Items.Add("Bold");
    cboStyle.Items.Add("Bold Italic");
    for (int i = 8; i <= 72; i++)
    {
        cboSize.Items.Add(i);
    }
}
```

Điều khiển ComboBox

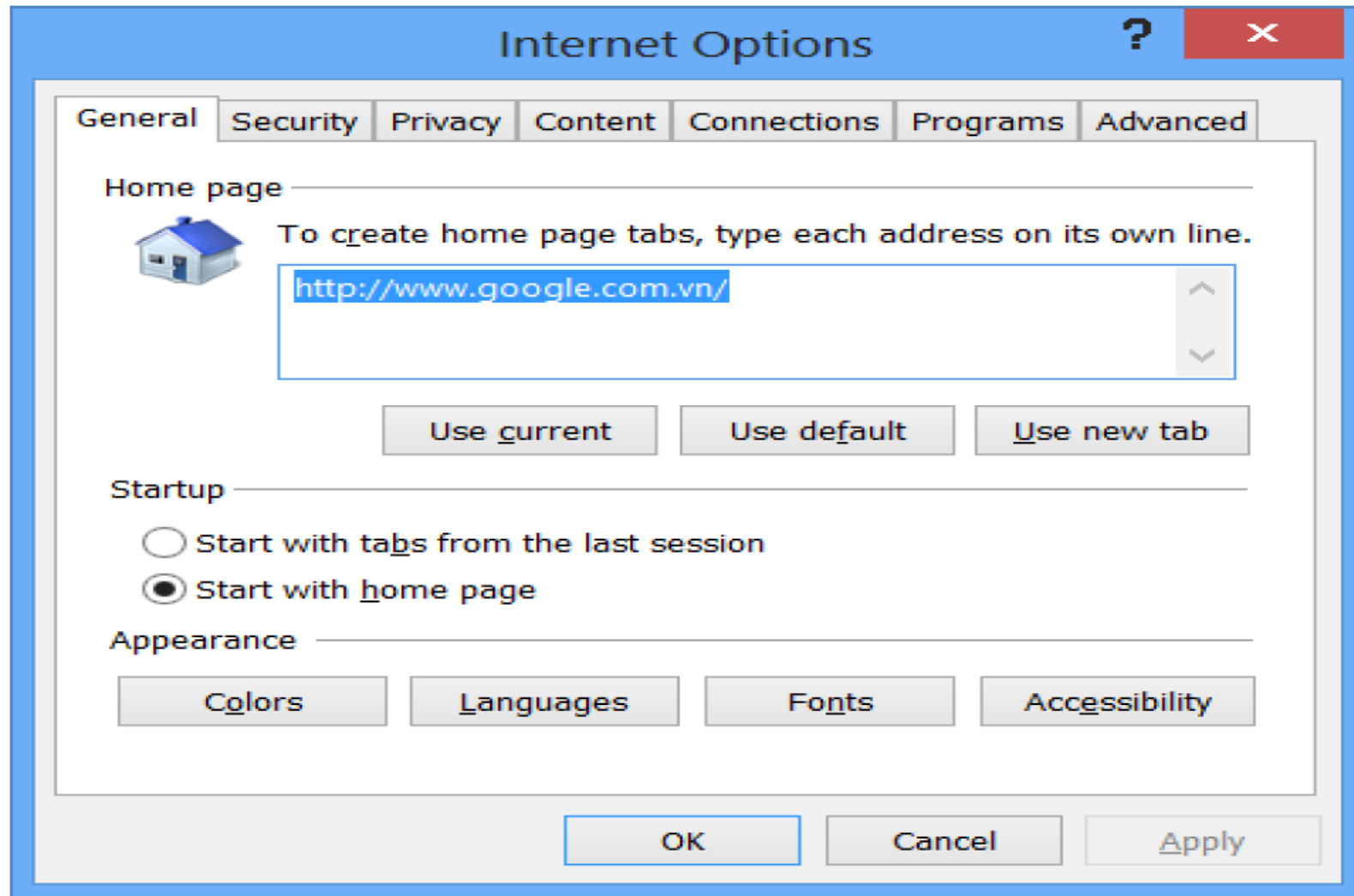
```
private void cboFont_SelectedIndexChanged(object sender,..)
{
    if (cboFont.SelectedIndex == -1) return;
    lbSample.Font = new Font(cboFont.Text,lbSample.Font.Size,lbSample.Font.Style);
}

private void cboSize_SelectedIndexChanged(object sender,..)
{
    if (cboSize.SelectedIndex == -1) return;
    lbSample.Font = new Font(lbSample.Font.Name,
    (float)Convert.ToDouble(cboSize.Text),lbSample.Font.Style);
}

private void cboStyle_SelectedIndexChanged(object sender,..)
{
    if (cboStyle.SelectedIndex == 0)
    lbSample.Font = new Font(cboFont.Text, lbSample.Font.Size,FontStyle.Regular);
}
```

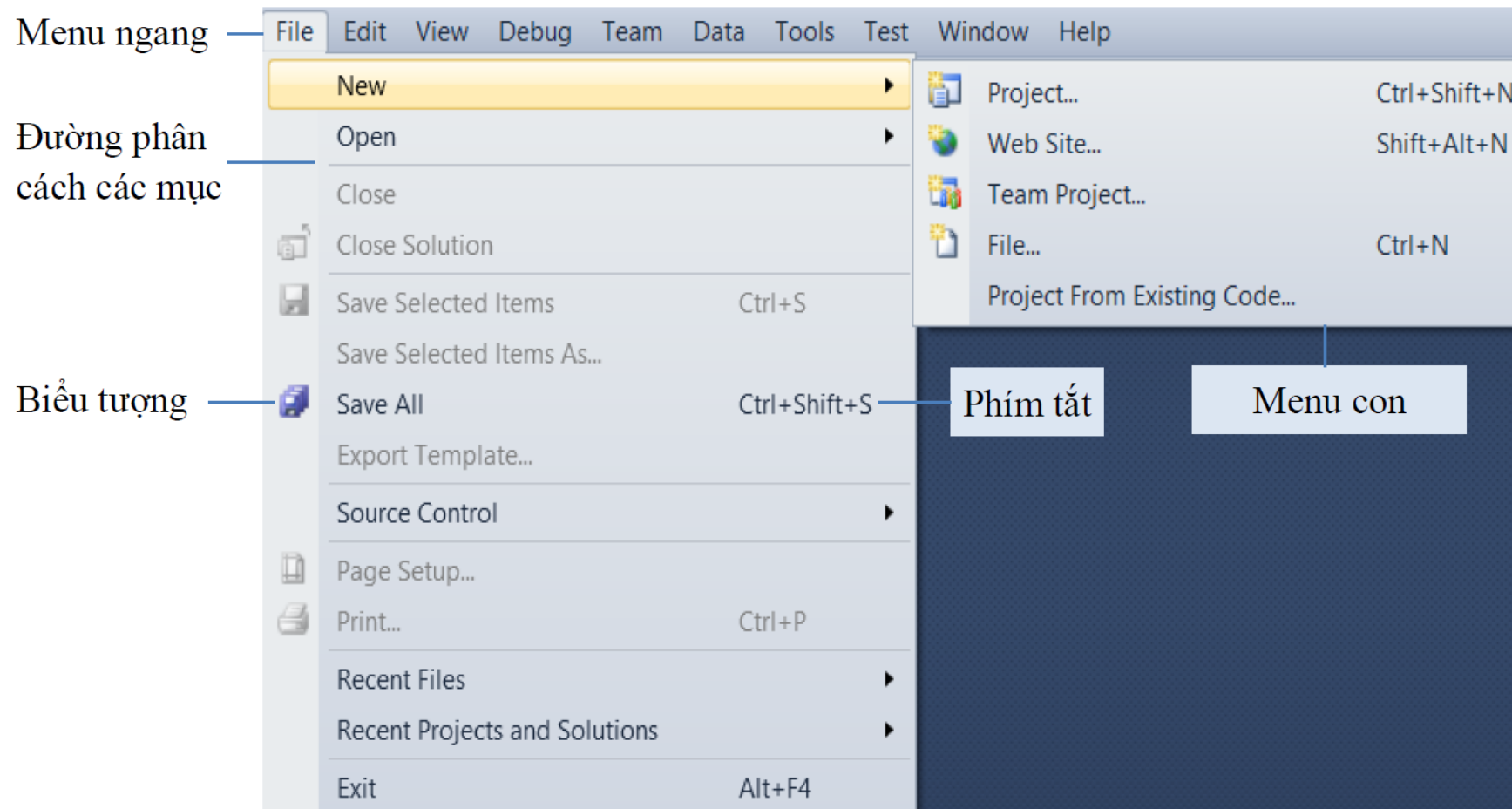

Điều khiển TabControl

- Cho phép tạo các trang Tab trong ứng dụng



Điều khiển Menu

- Menu dùng để nhóm các lệnh có liên quan với nhau
- Menu thường gồm menu ngang và menu dọc



Điều khiển Menu

■ Các thuộc tính thường dùng

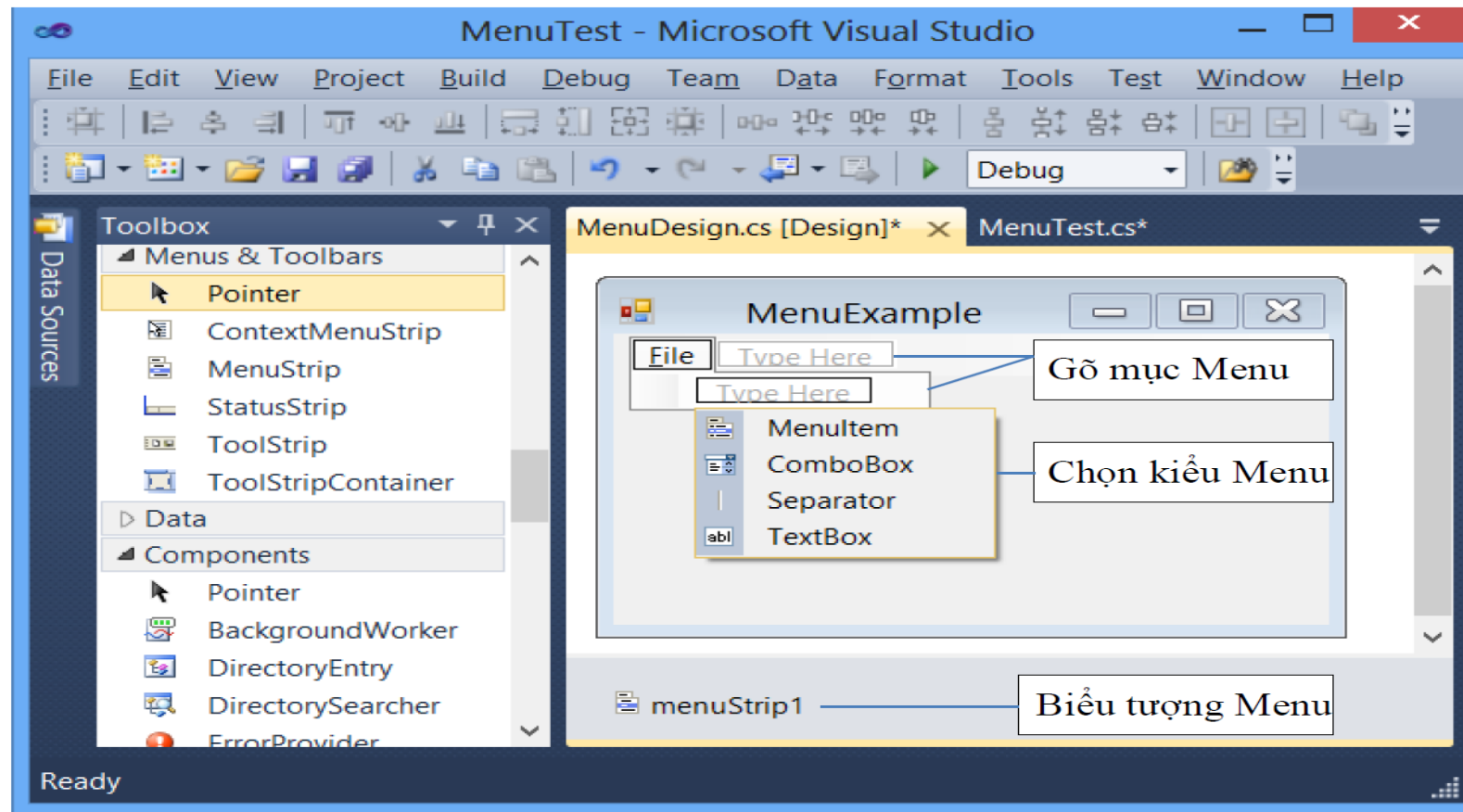
- ❑ Name: Tên thực đơn được dùng trong mã lệnh.
- ❑ Checked: Có/không mục của thực đơn xuất hiện đối tượng CheckBox khi chọn mục thực đơn.
- ❑ ShortCutKey: Đặt phím tắt cho mục của thực đơn
- ❑ ShowShortcut: Có/không phím tắt hiển thị trên các mục của thực đơn.
- ❑ Text: Dòng văn bản xuất hiện trên mục của thực đơn.

■ Các sự kiện thường dùng

- ❑ Click(): Xảy ra khi một mục của thực đơn được chọn hoặc ấn phím nóng.

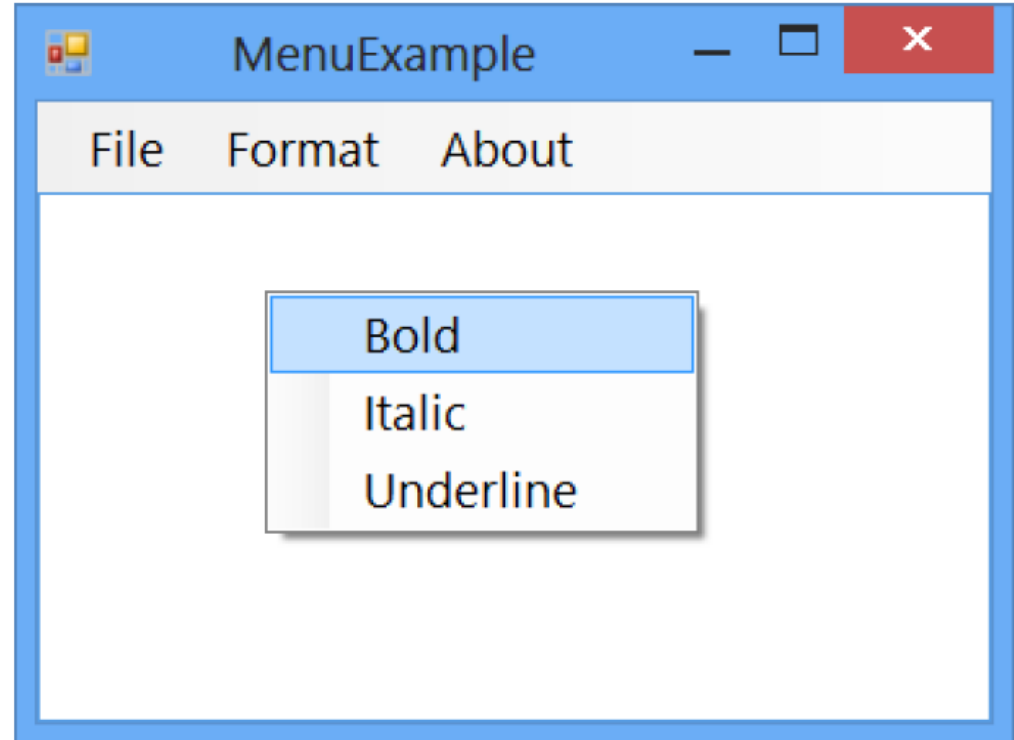
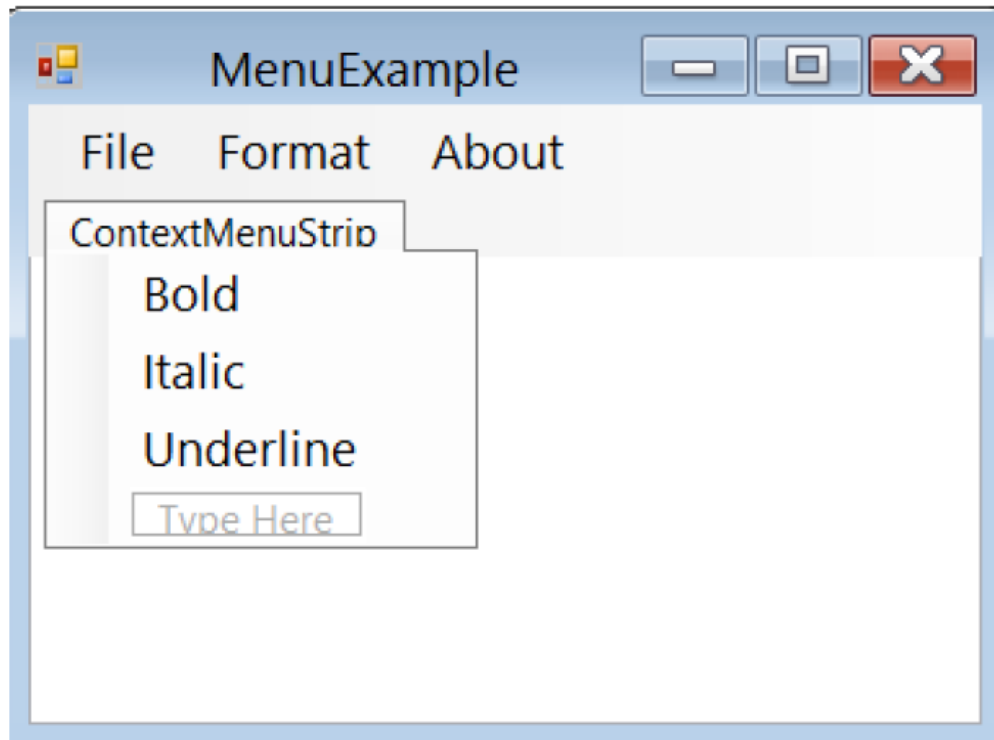
Điều khiển Menu

- Các bước thiết kế
 - Kéo điều khiển MenuStrip từ cửa sổ Toolbox vào Form
 - Set Image: Chèn hình ảnh; ShortCutKey: Đặt phím tắt

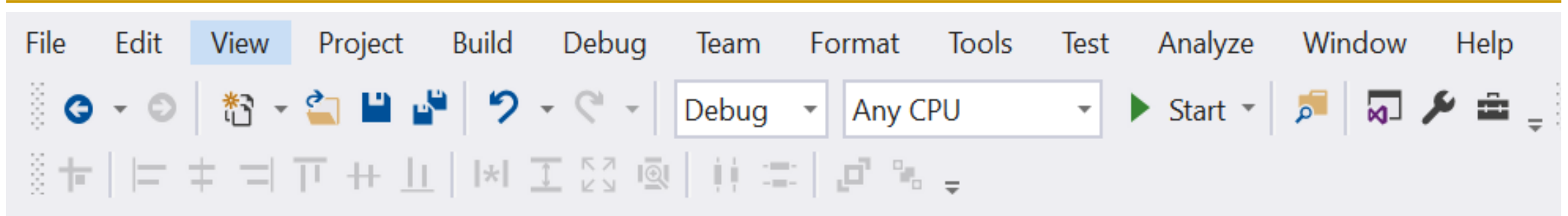


Menu ngữ cảnh

- Kéo điều khiển ContextMenuStrip vào Form và thiết kế như các bước thiết kế thực đơn chính.
- Chọn thuộc tính ContextMenuStrip của đối tượng là tên của thực đơn ngữ cảnh.



Điều khiển Toolbar

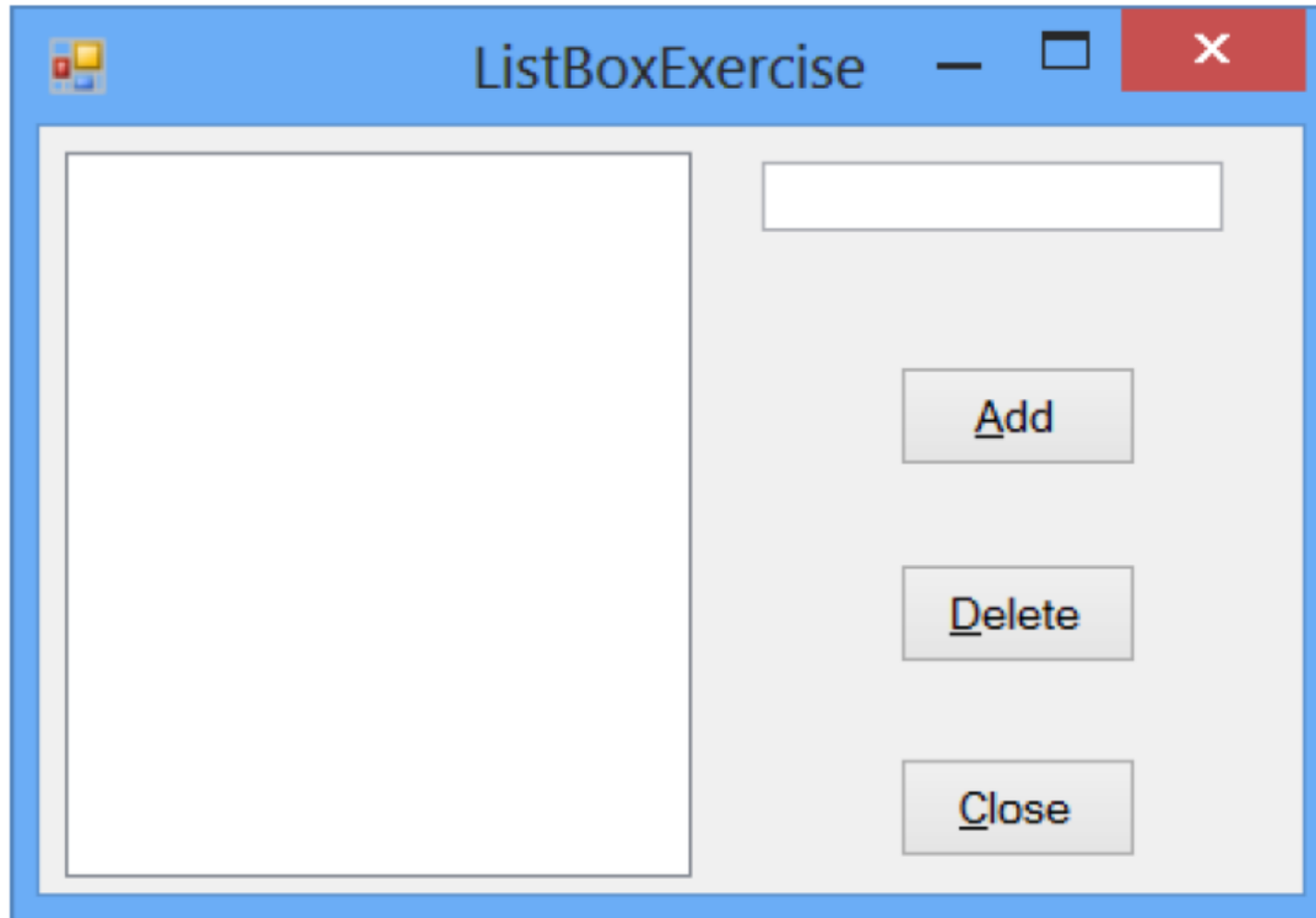


■ Các bước thiết kế

- ❑ Kéo điều khiển ToolStrip từ cửa sổ Toolbox vào Form.
- ❑ Đưa con trỏ chuột vào đối tượng lớp ToolStrip, nhấn chuột phải và chọn các đối tượng tạo thanh công cụ.
- ❑ Chọn cửa sổ Properties và đặt tên ở thuộc tính Name.
- ❑ Chèn hình ảnh cho đối tượng bằng cách nháy chuột phải, chọn Set Image, chọn Local Resource và chọn Import.

Bài tập

1. Xây dựng một form cho phép thêm và xóa các dòng trong ListBox.

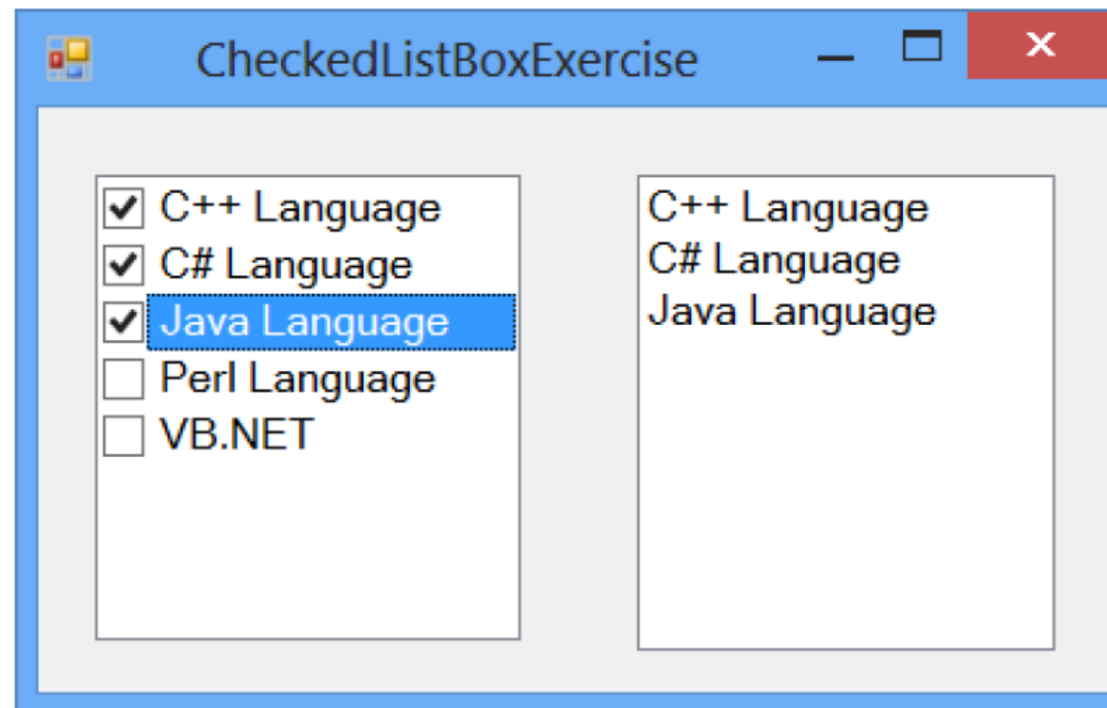


The image shows a screenshot of a Windows application window titled "ListBoxExercise". The window has a standard Windows XP-style title bar with a blue background, a minimize button, a maximize button, and a red close button. The main content area of the window is light gray and contains a large, empty white rectangular box on the left, which is a ListBox. To the right of the ListBox, there is a text input field at the top. Below the text input field, there are three buttons stacked vertically: "Add", "Delete", and "Close". Each button has a light gray background and a thin black border. The text on the buttons is black, and the "Add" and "Delete" buttons have a small underline under the first letter of the text.

Bài tập

2. Xây dựng một form cho phép:

- ❑ Nếu chọn một dòng trong CheckedListBox bên trái, dòng được chọn sẽ chuyển sang ListBox bên phải.
- ❑ Nếu bỏ chọn một dòng trong CheckedListBox bên trái, dòng được chọn sẽ xóa khỏi ListBox bên phải.



Bài tập

3. Xây dựng một form chứa TabControl với các trang Tab có giao diện sau.

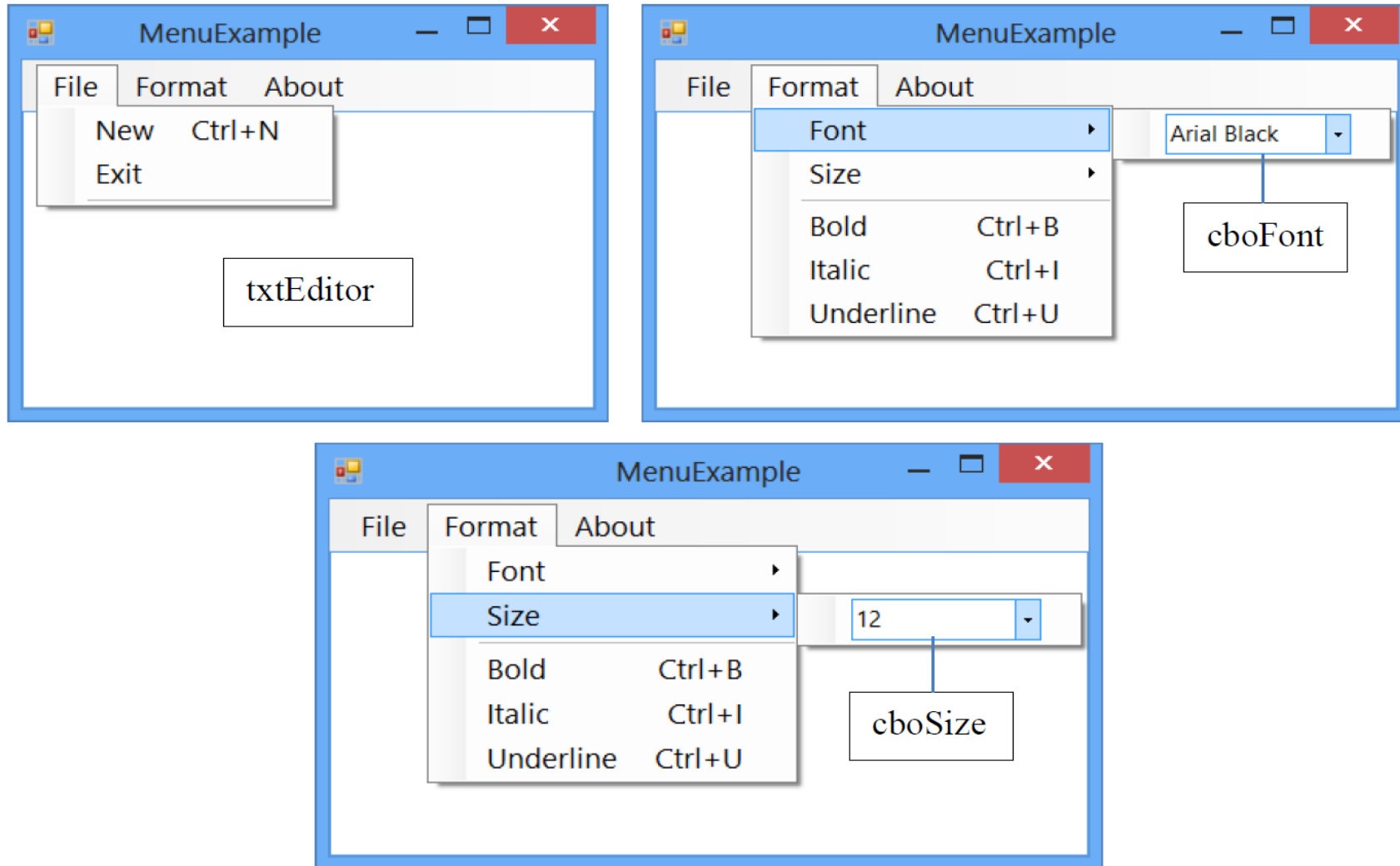
- Khi chọn Font, Style và Size trong Tab1, dòng chữ sẽ thay đổi tương ứng.
- Khi chọn Color trong Tab2, màu chữ sẽ thay đổi tương ứng.

The screenshot shows a window titled 'TabControlExample' with two tabs: 'Font' (selected) and 'Color'. The 'Font' tab contains three input fields labeled 'Font:', 'Font Style:', and 'Size:'. Below these fields is a 'Sample' label and a text area displaying 'AaBbCcDcYyZz'.

The screenshot shows the same 'TabControlExample' window, but with the 'Color' tab selected. The 'Color' tab contains three radio buttons labeled 'Black', 'Blue', and 'Red'. Below the radio buttons is a 'Sample' label and a text area displaying 'AaBbCcDcYyZz'.

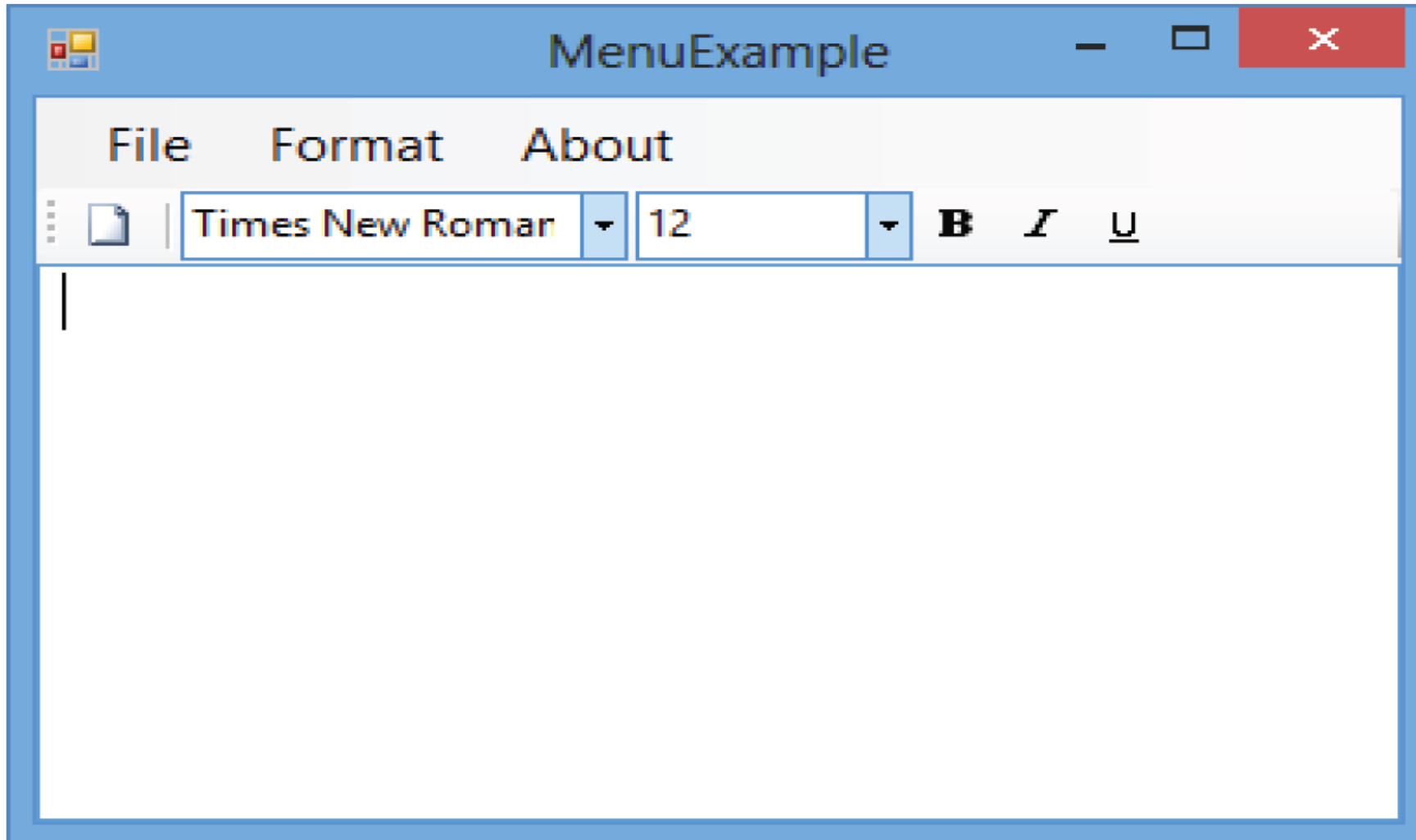
Bài tập

4. Xây dựng một menu với giao diện như hình sau:



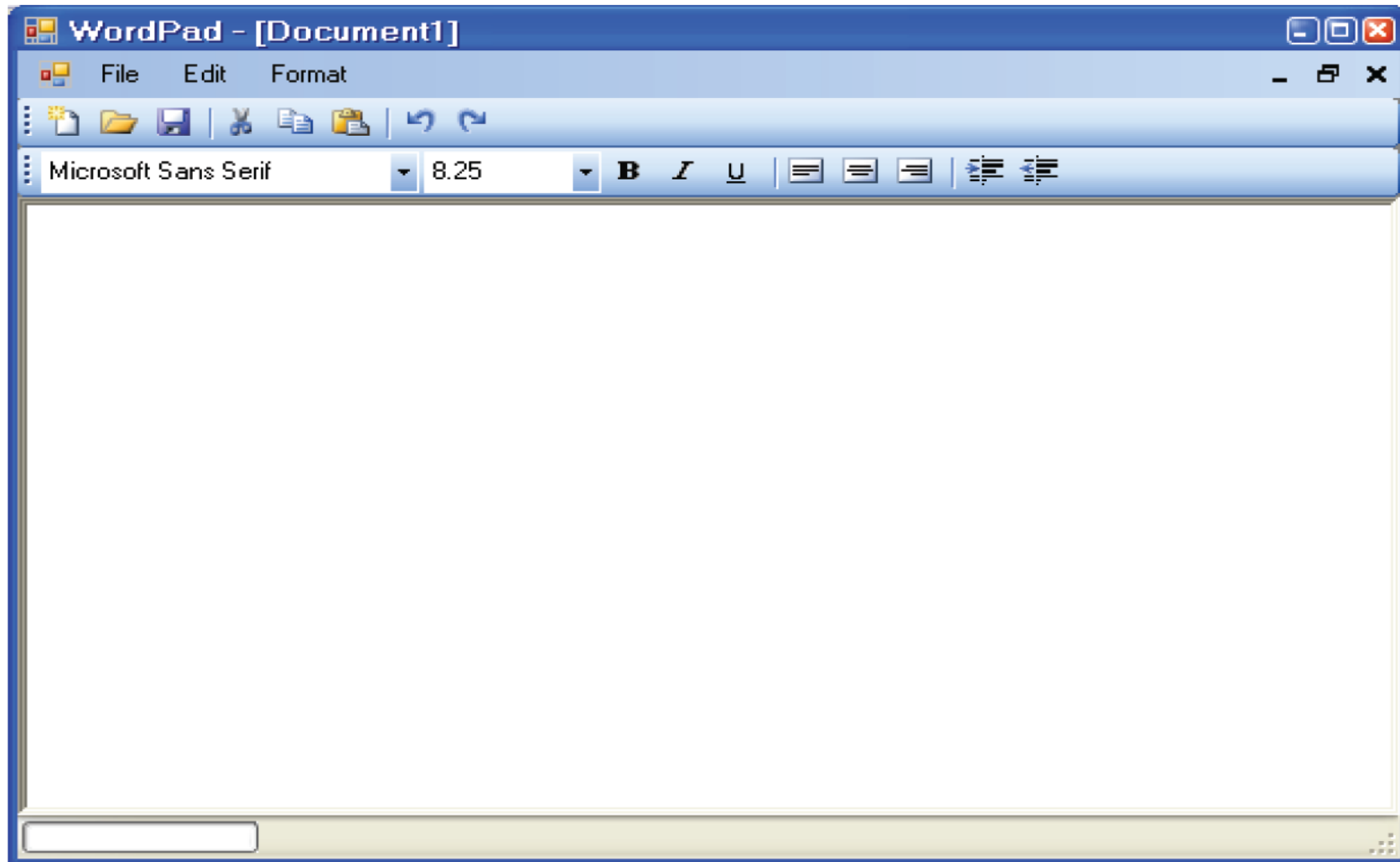
Bài tập

5. Xây dựng ToolBar cho giao diện của bài tập 3 như sau:



Bài tập

6. Xây dựng một hệ soạn thảo văn bản có giao diện sau:



Kết thúc



HỎI & TRẢ LỜI