

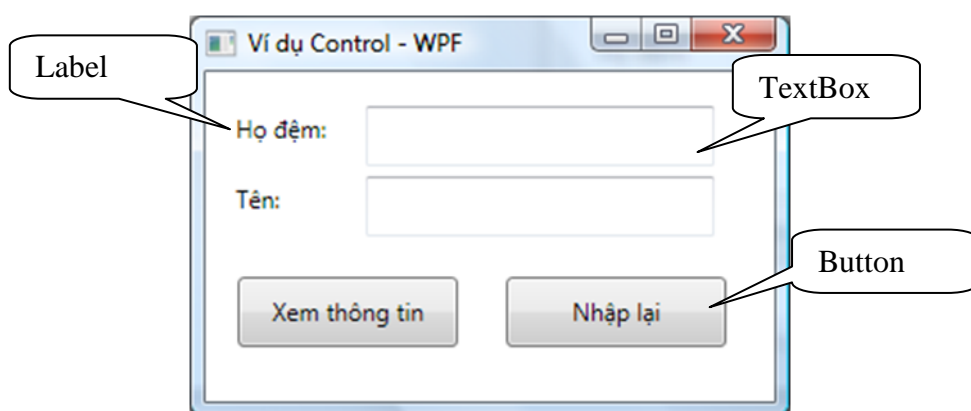
## Bài 2

# CÁC ĐIỀU KHIỂN (CONTROL) CƠ BẢN CỦA WPF

Trong lập trình giao diện người dùng, điều khiển (Control) là các nhân tố quan trọng cấu thành nên giao diện người dùng, cho phép họ giao tiếp với ứng dụng. Control có thể được hiểu một cách đơn giản là các phần tử trên một cửa sổ như các nhãn (Label), hộp soạn thảo (TextBox), nút bấm (Button), hộp danh sách (ListBox, ComboBox),... để hiển thị các thông tin tới người dùng và cho phép người dùng nhập thông tin cần thiết cho chương trình. Phần này giới thiệu cách tạo lập và sử dụng các Control cơ bản nhất của cửa sổ xây dựng bằng công nghệ WPF.

### 1. Tổng quan về tạo lập các điều khiển với WPF

Điểm khác biệt cơ bản giữa mã lệnh tạo giao diện dựa trên WPF so với phương pháp cũ là ứng dụng WPF sử dụng các đặc tả XAML (ngoài việc sử dụng mã lệnh C# hay VB.Net) để định nghĩa giao diện, trong khi phương pháp cũ phải sử dụng trực tiếp mã lệnh của C# hay VB.Net để định nghĩa giao diện. Ví dụ, để xây dựng giao diện cửa sổ đơn giản như Hình 2.1 dưới đây.



Hình 2.1 Một ví dụ về cửa sổ với các control đơn giản

Đoạn mã trình bằng XAML:

```
<Grid>
  <Label Height="30" HorizontalAlignment="Left" Margin="10,15,0,0" Name="label1"
  VerticalAlignment="Top" Width="60">Họ đệm:</Label>

  <Label Height="30" HorizontalAlignment="Left" Margin="10,50,0,0" Name="label2"
  VerticalAlignment="Top" Width="60">Tên:</Label>
```

```

<TextBox Height="30" Margin="80,17,30,0" Name="textBox1" VerticalAlignment="Top" />

<TextBox Height="30" Margin="80,52,30,0" Name="textBox2" VerticalAlignment="Top" />

<Button Height="35" HorizontalAlignment="Left" Margin="16,0,0,27" Name="button1"
VerticalAlignment="Bottom" Width="110">Xem thông tin</Button>

<Button Height="35" HorizontalAlignment="Right" Margin="0,0,24,27" Name="button2"
VerticalAlignment="Bottom" Width="110">Nhập lại</Button>

</Grid>

```

Đoạn mã trình bằng C#:

```

// Tạo nhãn Họ đệm
this.label1 = new System.Windows.Forms.Label();
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(17, 16);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(48, 13);
this.label1.TabIndex = 0;
this.label1.Text = "Họ đệm:";

//Tạo nhãn Tên
this.label2 = new System.Windows.Forms.Label();

this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(17, 50);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(29, 13);
this.label2.TabIndex = 0;
this.label2.Text = "Tên:";

//Tạo TextBox nhập Họ đệm
this.textBox1 = new System.Windows.Forms.TextBox();
this.textBox1.Location = new System.Drawing.Point(100, 16);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(160, 20);
this.textBox1.TabIndex = 1;

//Tạo TextBox nhập Tên

```

```

this.textBox2 = new System.Windows.Forms.TextBox();
this.textBox2.Location = new System.Drawing.Point(100, 50);
this.textBox2.Name = "textBox2";
this.textBox2.Size = new System.Drawing.Size(160, 20);
this.textBox2.TabIndex = 1;

// Tạo nút bấm Xem thông tin
this.button1 = new System.Windows.Forms.Button();
this.button1.Location = new System.Drawing.Point(20, 114);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(99, 38);
this.button1.TabIndex = 2;
this.button1.Text = "Xem thông tin";
this.button1.UseVisualStyleBackColor = true;

//Tạo nút bấm nhập lại
this.button2 = new System.Windows.Forms.Button();
this.button2.Location = new System.Drawing.Point(161, 114);
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(99, 38);
this.button2.TabIndex = 2;
this.button2.Text = "Nhập lại";
this.button2.UseVisualStyleBackColor = true;

```

Như vậy, điều chúng ta cần là tìm hiểu các thẻ XAML để mô tả các Control cần thiết. Tuy nhiên bạn không cần phải lo lắng nếu như chưa quen với các mã lệnh XAML (dựa trên XML) này vì bộ công cụ Visual Studio.Net 2008 đã hỗ trợ thiết kế giao diện trực quan và tự động sinh mã XAML tương ứng.

## 2. Các điều khiển cơ bản trong WPF

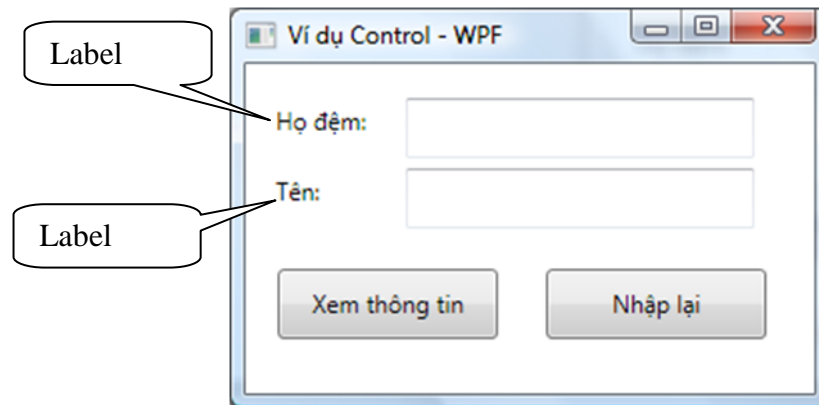
Chúng ta sẽ tìm hiểu chi tiết một số điều khiển cơ bản của cửa sổ:

- **Label:** Nhãn.
- **TextBox:** Hộp soạn thảo.
- **Button:** Nút bấm.
- **CheckBox:** Hộp chọn.
- **RadioButton:** Hộp chọn radio (chỉ được phép chọn 1 mục trong mỗi nhóm).
- **ListBox:** Hộp danh sách

- **ComboBox:** Hộp danh sách thả xuống..

## 1.1 LABEL - Nhãn

Nhãn (**Label**) là các điều kiện để hiển thị các văn bản tĩnh, thường được sử dụng để làm nhãn cho các control khác như Textbox, ListBox, ComboBox,....



Hình 2.2 Minh họa về label

Các Label được mô tả bằng đoạn mã XAML sau:

```
<Grid>
  <Label Height="30" HorizontalAlignment="Left" Margin="10,15,0,0" Name="label1"
  VerticalAlignment="Top" Width="60">Họ đệm:</Label>
</Grid>
```

Nhãn được bắt đầu `<Label>` và kết thúc là `</Label>`, nội dung của nhãn là đoạn văn bản đặt giữa cặp thẻ này. Trong ví dụ này “Họ đệm:” là nội dung của nhãn.

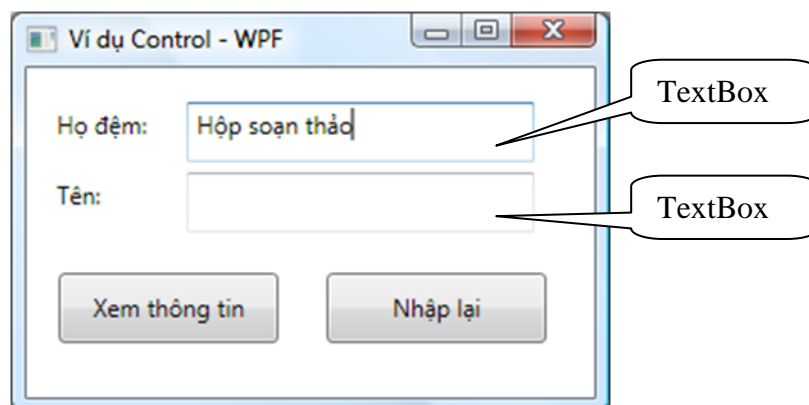
Bên trong thẻ `<Label>` có rất nhiều đặc tính để mô tả về thẻ, trong đó:

- **Height="30"** : Độ cao của khung nhãn là 30px
- **HorizontalAlignment="Left"** : Nhãn được căn trái trong cửa sổ
- **Margin="10,15,0,0"** : có 4 giá trị là Left,Top,Right,Bottom
- **Name="label1"** : Tên của nhãn là label1
- **VerticalAlignment="Top"** :Nhãn được căn theo đỉnh của cửa sổ.
- **Width="60"**: Chiều rộng của nhãn là 60px

Trên đây là một số đặc tính cơ bản của nhãn, ngoài ra còn có nhiều đặc tính khác áp dụng cho nhãn như màu nền, màu chữ,....

## 1.2 TextBox – Hộp soạn thảo

Hộp soạn thảo (**TextBox**) là control cho phép người dùng nhập dữ liệu dạng văn bản.



Hình 2.3 Minh họa về TextBox

Dưới đây là đoạn mã XAML của hộp soạn thảo

```
<Grid>
<TextBox Height="30" Margin="80,17,30,0" Name="textBox1" VerticalAlignment="Top" /> Hộp soạn
thảo
</TextBox>

<TextBox Height="30" Margin="80,52,30,0" Name="textBox2" VerticalAlignment="Top" />

</Grid>
```

Hộp soạn thảo được tạo nên bởi thẻ `<TextBox/>`. Nếu muốn thiết lập sẵn nội dung mặc định cho hộp soạn thảo, ta đặt nội dung này vào giữa cặp thẻ `<TextBox/>` Nội dung `</TextBox>`. Nếu không muốn đặt giá trị mặc định thì không cần thẻ đóng `</TextBox>`.

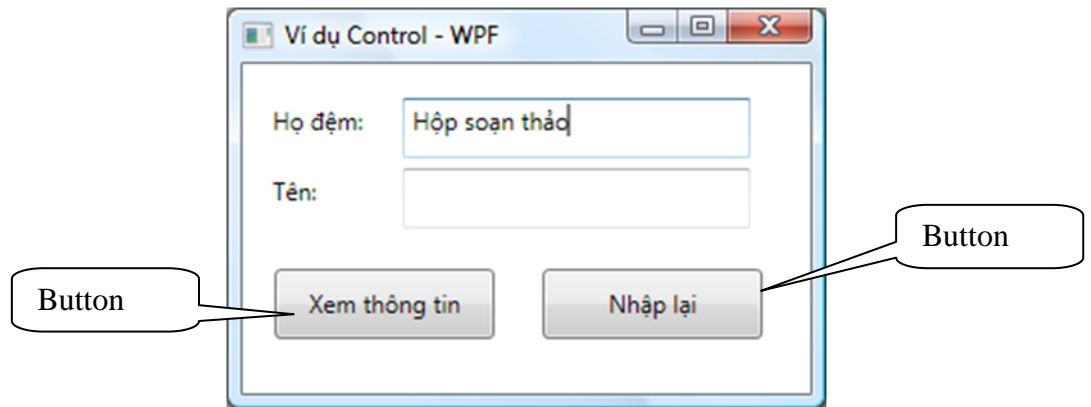
Thẻ `<TextBox/>` cũng có nhiều đặc tính, trong đó:

- `Margin="80,17,30,0"`: Cách lề trái 80, đỉnh cửa sổ 17, cạnh phải 30
- `Name="textBox1"`: Tên của hộp soạn thảo là textBox1
- `VerticalAlignment="Top"`: Căn theo đỉnh cửa sổ

Đặc điểm của hộp soạn thảo với các đặc tính trên là khi người dùng co dãn, thay đổi kích thước cửa sổ, chiều rộng của hộp soạn thảo tự động co dãn theo.

### 1.3 Button – Nút bấm

Nút bấm (**Button**) là loại điều khiển cho phép người dùng nhấn chuột để chọn lệnh, khi nhấn vào nút bấm, nó sẽ sinh ra sự kiện **Click** và sẽ chạy các lệnh gắn với sự kiện này.



Hình 2.4 Minh họa về nút bấm

Dưới đây là đoạn mã sinh ra các nút bấm trên

```
<Grid>
  <Button Height="35" HorizontalAlignment="Left" Margin="16,0,0,27" Name="button1"
  VerticalAlignment="Bottom" Width="110" Click="button1_Click">Xem thông tin</Button>
  <Button Height="35" HorizontalAlignment="Right" Margin="0,0,24,27" Name="button2"
  VerticalAlignment="Bottom" Width="110">Nhập lại</Button>
</Grid>
```

Nút bấm được bắt đầu bằng thẻ `<Button>` và kết thúc bằng thẻ `</Button>`. Nhãn của nút bấm được đặt trong cặp thẻ `<Button> Nhãn nút bấm </Button>`.

Nút bấm có nhiều đặc tính, trong đó:

- `Height="35"`: Chiều cao nút bấm là 35
- `Width="110"`: Chiều rộng là 110
- `HorizontalAlignment="Left"`: Căn theo lề trái
- `VerticalAlignment="Bottom"`: Căn theo đáy cửa sổ
- `Margin="16,0,0,27"`: Cách lề trái 16, cách đáy 27
- `Name="button1"`: Tên nút bấm là button1

- **Click="button1\_Click"**: Khi nhấn chuột vào nút sẽ kích hoạt phương thức button1\_Click()

Vì dụ về đoạn mã lệnh gắn với cửa sổ chứa các điều khiển trên:

```
namespace WpfControlSample1
{
    public partial class Window1 : Window
    {
        public Window1()
        {
            InitializeComponent();
        }

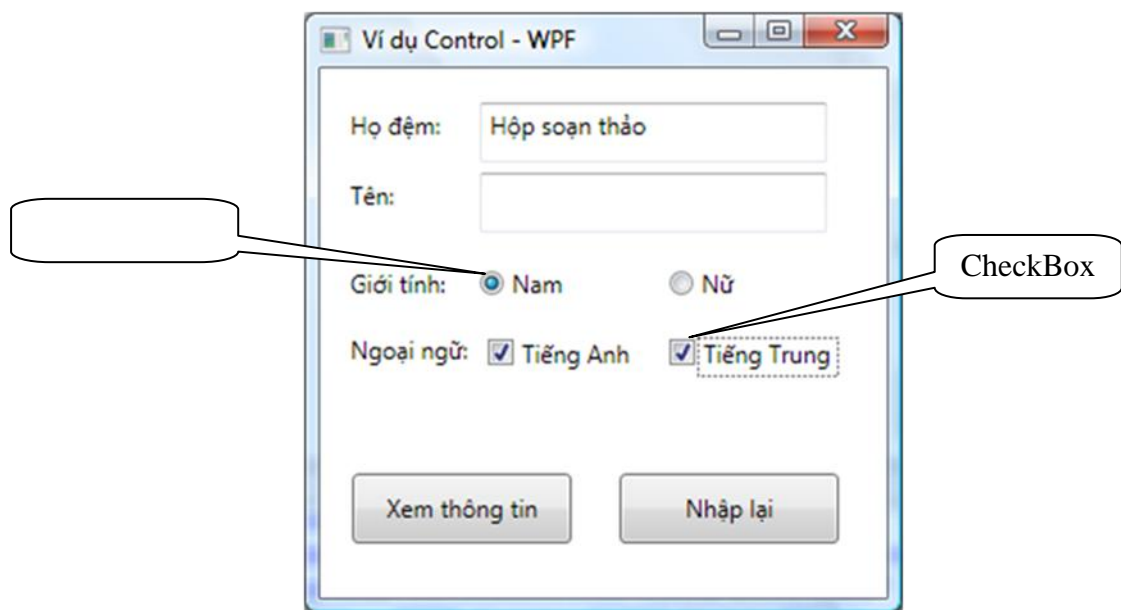
        private void button1_Click(object sender, RoutedEventArgs e)
        {
            String strHoTen;
            strHoTen = textBox1.Text + " " + textBox2.Text;
            MessageBox.Show("Xin chào: " + strHoTen);
        }
    }
}
```

## 1.4 Radio Button và CheckBox

Radio Button và CheckBox đều là điều khiển dạng hộp chọn. Tuy nhiên, điểm khác biệt cơ bản giữa hai loại điều khiển này là:

**Radio Button:** là hộp chọn theo nhóm, nghĩa là các hộp trong cùng một nhóm sẽ loại trừ nhau, tại một thời điểm người dùng chỉ được chọn một trong các mục. Ví dụ như hộp chọn giới tính, ta phải sử dụng radio vì tại một thời điểm chỉ cho phép chọn Nam hoặc Nữ

**CheckBox:** là hộp chọn mà người dùng có thể chọn một hoặc nhiều mục cùng một lúc. Ví dụ như mục chọn Ngoại ngữ, cho phép người dùng chọn đồng thời nhiều mục.



Hình 2.5 Minh họa về hộp chọn Radio và CheckBox

Dưới đây là đoạn mã tạo Radio Button và CheckBox.

```
<Grid>

<Label Height="30" HorizontalAlignment="Left" Margin="10,94,0,0" Name="label3"
VerticalAlignment="Top" Width="60">Giới tính:</Label>

<RadioButton Height="22" Margin="80,99,0,0" Name="radioButton1" VerticalAlignment="Top"
HorizontalAlignment="Left" Width="79" GroupName="GioiTinh" IsChecked="True">Nam</RadioButton>

<RadioButton Height="22" HorizontalAlignment="Right" Margin="0,99,30,0" Name="radioButton2"
VerticalAlignment="Top" Width="79" GroupName="GioiTinh">Nữ</RadioButton>

<Label HorizontalAlignment="Left" Margin="10,127,0,105" Name="label4" Width="69">Ngoại
ngữ:</Label>

<CheckBox Margin="84,0,119,110" Name="checkBox1" Height="20" VerticalAlignment="Bottom"
IsChecked="True">Tiếng Anh</CheckBox>

<CheckBox HorizontalAlignment="Right" Margin="0,0,24,110" Name="checkBox2" Width="85"
Height="20" VerticalAlignment="Bottom">Tiếng Trung</CheckBox>

</Grid>
```

Radio Button được tạo bởi thẻ `<RadioButton>` và kết thúc bởi `</RadioButton>`, giữa cặp thẻ này là nhãn của Radio Button `<RadioButton> Nhãn </RadioButton>`.



CheckBox được tạo bởi thẻ `<CheckBox>` và kết thúc bởi `</CheckBox>`, giữa cặp thẻ này là nhãn của CheckBox `<CheckBox> Nhãn </CheckBox>`.

Cả hai thẻ này đều có đặc tính `IsChecked="True"` hoặc `IsChecked="False"`. Mục nào có thuộc tính này sẽ được tự động chọn khi cửa sổ bắt đầu hiển thị.

Đối với Radio Button, vì là hộp chọn loại trừ, nếu trong một cửa sổ có nhiều nhóm Radio Button khác nhau thì các Radio Button của mỗi nhóm được phân biệt bởi đặc tính `GroupName="TenNhom"`. Ví dụ, trên cùng một cửa sổ có hai Radio Button chọn Giới tính (Nam; Nữ) và ba Radio Button khác chọn nghề nghiệp (Kinh doanh; Kỹ Thuật; Marketing) thì các Radio Button Nam, Nữ phải có cùng GroupName với nhau, ba Radio Button Kinh doanh, Kỹ Thuật, Marketing phải có cùng GroupName và khác với GroupName của nhóm giới tính.

Đoạn mã lệnh minh họa kiểm tra mục chọn Radio

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    String strMessage, strHoTen, strTitle, strNgoaiNgu="";
    strHoTen = textBox1.Text + " " + textBox2.Text;

    if (radioButton1.IsChecked==true)
        strTitle = "Mr.";
    else
        strTitle = "Miss/Mrs.";
    strMessage = "Xin chào: " + strTitle + " " + strHoTen;
    MessageBox.Show(strMessage);
}
```

Đoạn mã lệnh minh họa kiểm tra mục chọn CheckBox

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    String strMessage, strNgoaiNgu="";

    if (checkBox1.IsChecked == true)
    {
        strNgoaiNgu = "Tiếng Anh";
    }
    if (checkBox2.IsChecked == true)
    {
        strNgoaiNgu = (strNgoaiNgu.Length==0) ? "Tiếng Trung": (strNgoaiNgu+ " và Tiếng Trung");
    }
}
```

```

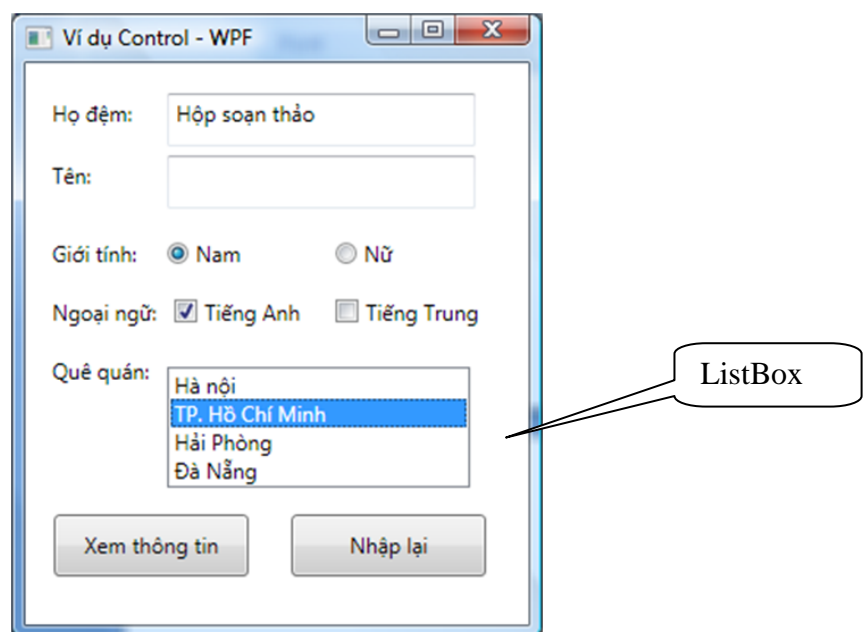
    }
    strMessage = "Ngoại ngữ: " + strNgoaiNgu;
    MessageBox.Show(strMessage);
}

```

## 1.5 ListBox - Hộp danh sách

Hộp danh sách (**ListBox**) và là điều khiển hiển thị một danh sách các mục theo từng dòng và cho phép người dùng chọn một hay nhiều phần tử của danh sách.

Ví dụ về hộp danh sách chọn Quê quán:



Hình 2.6 Minh họa về hộp chọn ListBox

Dưới đây là đoạn mã tạo ListBox

```

<Grid>

    <Label Height="30" HorizontalAlignment="Left" Margin="10,0,0,126" Name="label5"
    VerticalAlignment="Bottom" Width="69">Quê quán:</Label>

    <ListBox Height="68" Margin="80,0,33,77" Name="listBox1" VerticalAlignment="Bottom"
    SelectedIndex="0">
        <ListBoxItem>Hà nội</ListBoxItem>
        <ListBoxItem>TP. Hồ Chí Minh</ListBoxItem>
        <ListBoxItem>Hải Phòng</ListBoxItem>
    </ListBox>

```

```
<ListBoxItem>Đà Nẵng</ListBoxItem>
</ListBox>

</Grid>
```

ListBox được tạo bởi thẻ `<ListBox>` và kết thúc bằng thẻ đóng `</ListBox>`.

Mỗi phần tử của danh sách nằm trong cặp thẻ `<ListBoxItem>` **Nhãn** `</ListBoxItem>` lồng bên trong cặp thẻ trên.

Đặc tính `SelectedIndex="k"` để yêu cầu tự động chọn phần tử thứ n trong danh sách khi mở cửa sổ. Phần tử đầu tiên của danh sách có giá trị là 0, phần tử cuối cùng là n-1. Nếu muốn khi mở cửa sổ không chọn phần tử nào thì đặt giá trị k bằng -1.

Đoạn mã lệnh lấy mục chọn từ ListBox

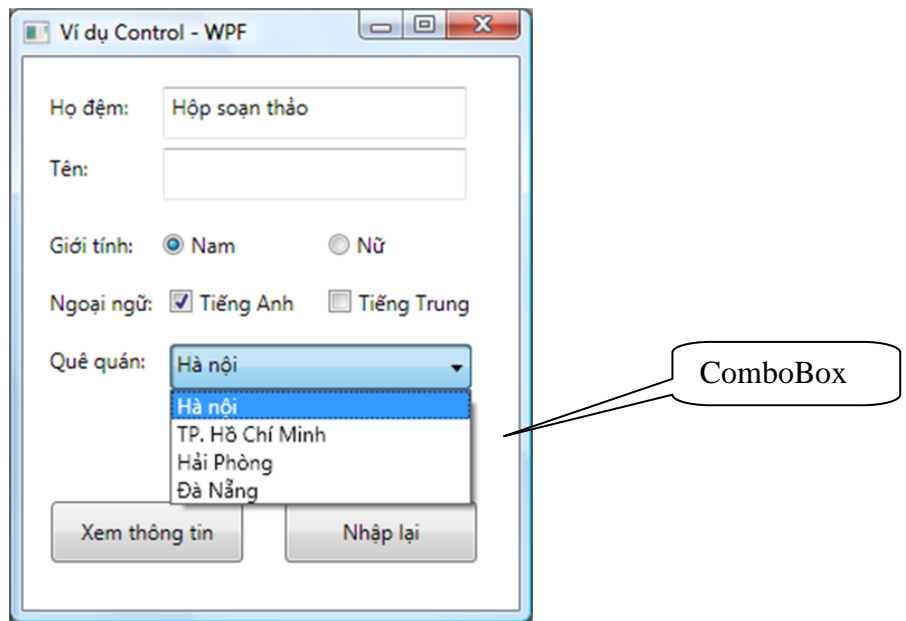
```
private void button1_Click(object sender, RoutedEventArgs e)
{
    String strMessage;

    if (listBox1.SelectedIndex >= 0)//Nếu đã có một mục trong danh sách được chọn
    {
        strMessage = "Quê Quán: " + listBox1.Text;
    }
    MessageBox.Show(strMessage);
}
```

## 1.6 ComboBox - hộp danh sách thả xuống

Hộp danh sách thả xuống (**ComboBox**) là các điều khiển hiển thị một danh sách theo từng dòng cho người dùng chọn. Tuy nhiên, khác với ListBox, ComboBox gọn gàng hơn bởi vì nó chỉ hiển thị 1 dòng và khi nhấn vào biểu tượng tam giác bên cạnh thì danh sách mới được mở ra. Combox chỉ cho phép chọn 1 dòng tại 1 thời điểm.

Ví dụ về hộp danh sách thả xuống chọn Quê quán:



Hình 2.7 Minh họa về hộp chọn ComboBox

Dưới đây là đoạn mã tạo ComboBox

```
<Grid>

<Label Height="30" HorizontalAlignment="Left" Margin="10,0,0,126" Name="label5"
VerticalAlignment="Bottom" Width="69">Quê quán:</Label>

<ComboBox Height="26" Margin="84,0,27,126" Name="comboBox1" VerticalAlignment="Bottom"
SelectedIndex="0">
  <ComboBoxItem>Hà nội</ComboBoxItem>
  <ComboBoxItem>TP. Hồ Chí Minh</ComboBoxItem>
  <ComboBoxItem>Hải Phòng</ComboBoxItem>
  <ComboBoxItem>Đà Nẵng</ComboBoxItem>
</ComboBox>

</Grid>
```

ComboBox được tạo bởi thẻ `<ComboBox>` và kết thúc bằng thẻ đóng `</ComboBox>`.

Mỗi phần tử của danh sách nằm trong cặp thẻ `<ComboBoxItem>` Nhãn `</ComboBoxItem>` Lồng bên trong cặp thẻ trên.

Đặc tính **SelectedIndex="k"** để yêu cầu tự động chọn phần tử thứ n trong danh sách khi mở cửa sổ. Phần tử đầu tiên của danh sách có giá trị là 0, phần tử cuối cùng là n-1. Nếu muốn khi mở cửa sổ không chọn phần tử nào thì đặt giá trị k bằng -1.

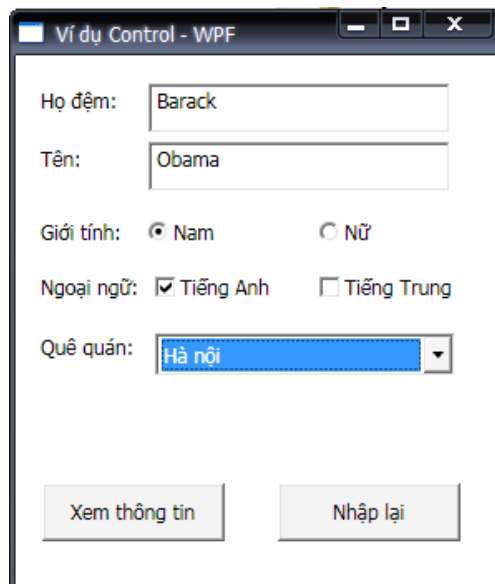
Đoạn mã lệnh lấy mục chọn từ ComboBox

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    String strMessage;

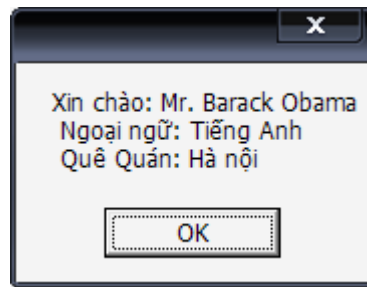
    if (comboBox1.SelectedIndex >= 0)//Nếu đã có một mục trong danh sách được chọn
    {
        strMessage = "Quê Quán: " + comboBox1.Text;
    }
    MessageBox.Show(strMessage);
}
```

### 3. Ví dụ xây dựng các control trong WPF

Yêu cầu: Viết chương trình hiển thị cửa sổ như sau



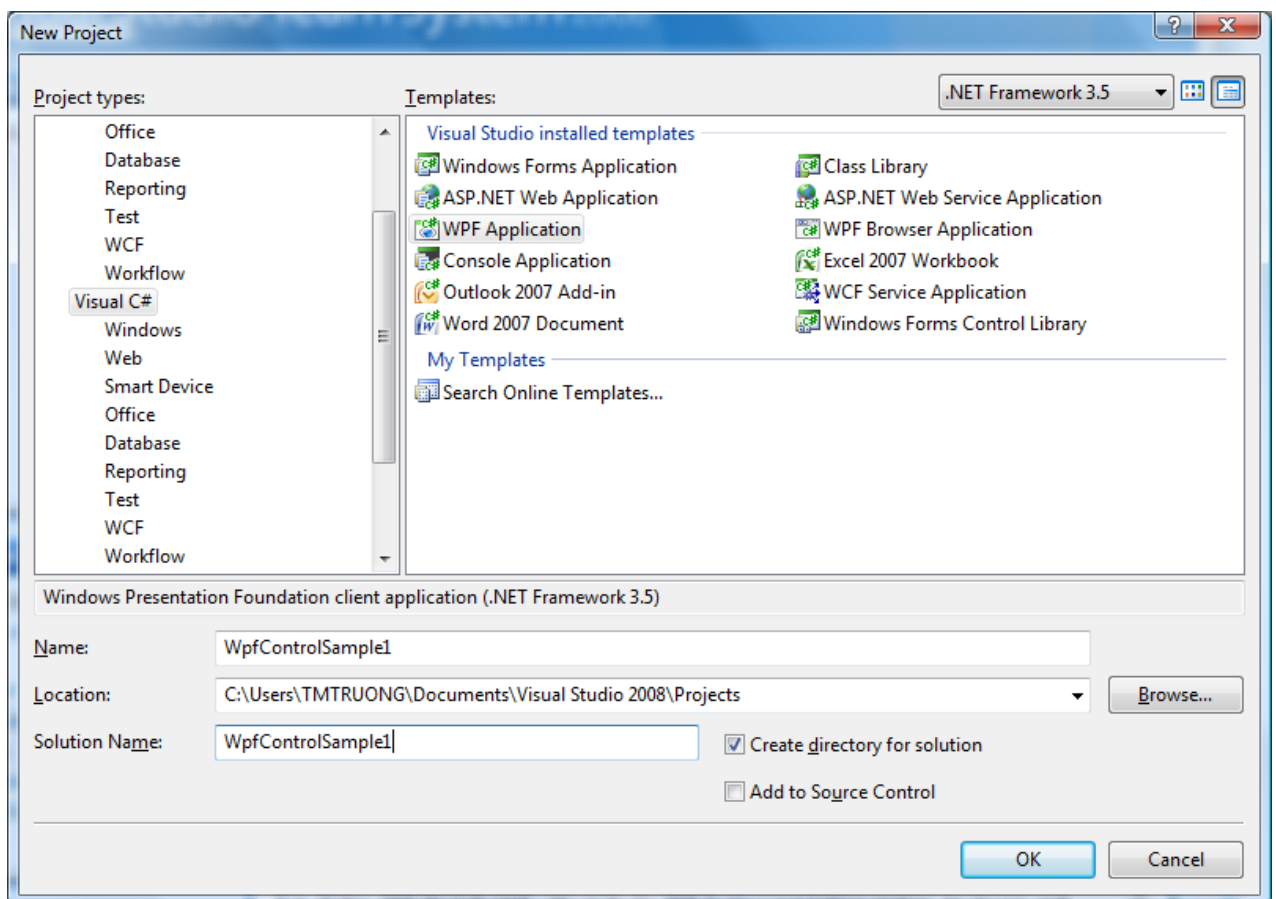
Khi nhấn **Xem thông tin**, nếu chọn Nam thì hiện lời chào “Xin chào Mr. Barack Obama”, nếu chọn Nữ thì hiện lời chào “Xin chào Miss/Mrs. Barack Obama”, như sau:



Nếu chọn Nhập lại, đưa trạng thái các control trên cửa sổ về trạng thái ban đầu.

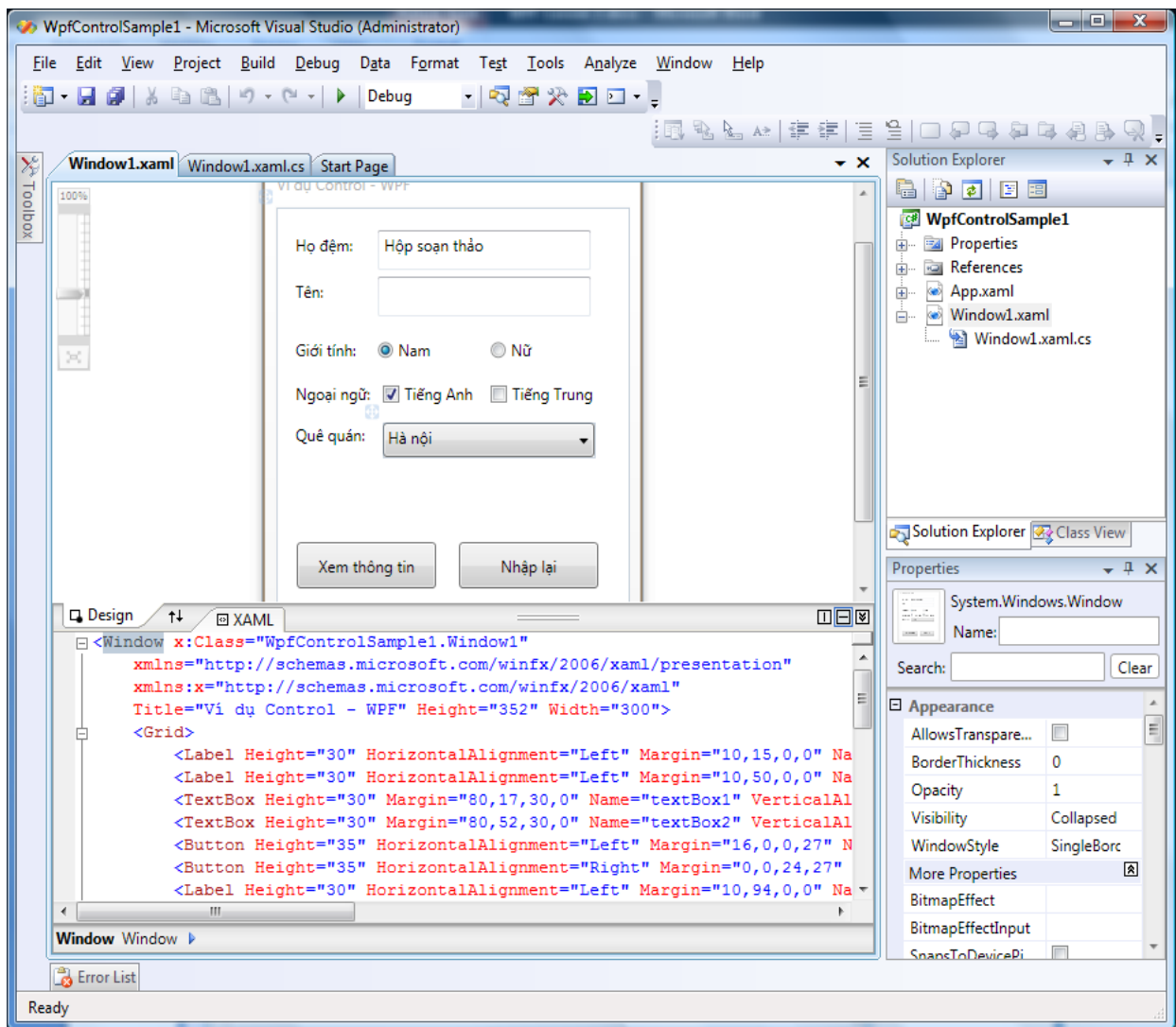
Các bước thực hiện như sau:

### 3.1 Tạo ứng dụng WPF



Ở đây, ta chú ý chọn .Net Framework 3.5.

Giao diện thiết kế ứng dụng:



### 3.2 Mã XAML của giao diện

Mở file Window1.xaml tương ứng với file code Window1.xaml.cs.

```
<Window x:Class="WpfControlSample1.Window1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Ví dụ Control - WPF" Height="352" Width="300">
    <Grid>
        <Label Height="30" HorizontalAlignment="Left" Margin="10,15,0,0" Name="label1"
VerticalAlignment="Top" Width="60" Focusable="False">Họ đệm:</Label>
        <Label Height="30" HorizontalAlignment="Left" Margin="10,50,0,0" Name="label2"
VerticalAlignment="Top" Width="60">Tên:</Label>
        <TextBox Height="30" Margin="80,17,30,0" Name="textBox1" VerticalAlignment="Top">Họ soạn
thảo</TextBox>
        <TextBox Height="30" Margin="80,52,30,0" Name="textBox2" VerticalAlignment="Top" />
```

```

<Button Height="35" HorizontalAlignment="Left" Margin="16,0,0,27" Name="button1"
VerticalAlignment="Bottom" Width="110" Click="button1_Click">Xem thông tin</Button>
<Button Height="35" HorizontalAlignment="Right" Margin="0,0,24,27" Name="button2"
VerticalAlignment="Bottom" Width="110" Click="button2_Click">Nhập lại</Button>
<Label Height="30" HorizontalAlignment="Left" Margin="10,94,0,0" Name="label3"
VerticalAlignment="Top" Width="60">Giới tính:</Label>
<RadioButton Height="22" Margin="80,99,0,0" Name="radioButton1" VerticalAlignment="Top"
HorizontalAlignment="Left" Width="79" GroupName="GioiTinh" IsChecked="True">Nam</RadioButton>
<RadioButton Height="22" HorizontalAlignment="Right" Margin="0,99,30,0" Name="radioButton2"
VerticalAlignment="Top" Width="79" GroupName="GioiTinh">Nữ</RadioButton>
<Label HorizontalAlignment="Left" Margin="10,127,0,154" Name="label4" Width="69">Ngoại
ngữ:</Label>
<CheckBox Margin="84,132,119,0" Name="checkBox1" Height="20" VerticalAlignment="Top"
IsChecked="True">Tiếng Anh</CheckBox>
<CheckBox HorizontalAlignment="Right" Margin="0,132,24,0" Name="checkBox2" Width="85"
Height="20" VerticalAlignment="Top">Tiếng Trung</CheckBox>
<Label Height="30" HorizontalAlignment="Left" Margin="10,0,0,126" Name="label5"
VerticalAlignment="Bottom" Width="69">Quê quán:</Label>
<ComboBox Height="26" Margin="84,0,27,126" Name="comboBox1" VerticalAlignment="Bottom"
SelectedIndex="0">
    <ComboBoxItem>Hà nội</ComboBoxItem>
    <ComboBoxItem>TP. Hồ Chí Minh</ComboBoxItem>
    <ComboBoxItem>Hải Phòng</ComboBoxItem>
    <ComboBoxItem>Đà Nẵng</ComboBoxItem>
</ComboBox>
</Grid>
</Window>

```

Gán tên phương thức `button1_Click` xử lý sự kiện *Click* cho nút bấm `button1` này, khi nhấn chuột vào nút “Xem thông tin” thì phương thức `button1_Click` sẽ được thực thi.

```

<Button Height="35" HorizontalAlignment="Left" Margin="16,0,0,27" Name="button1"
VerticalAlignment="Bottom" Width="110" Click="button1_Click">Xem thông tin</Button>

```

Gán tên phương thức `button2_Click` xử lý sự kiện *Click* cho nút bấm `button2` này, khi nhấn chuột vào nút “Nhập lại” thì phương thức `button2_Click` sẽ được thực thi.

```

<Button Height="35" HorizontalAlignment="Right" Margin="0,0,24,27" Name="button2"
VerticalAlignment="Bottom" Width="110" Click="button2_Click">Nhập lại</Button>

```



### 3.3 Mã lệnh C# xử lý các sự kiện nhấn nút

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfControlSample1
{
    /// <summary>
    /// Interaction logic for Window1.xaml
    /// </summary>
    public partial class Window1 : Window
    {
        public Window1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, RoutedEventArgs e)
        {
            String strMessage, strHoTen, strTitle, strNgoaiNgu="";
            strHoTen = textBox1.Text + " " + textBox2.Text;
            if (radioButton1.IsChecked==true)
                strTitle = "Mr.";
            else
                strTitle = "Miss/Mrs.";
            strMessage = "Xin chào: " + strTitle + " " + strHoTen;

            if (checkBox1.IsChecked == true)
            {
```

```

        strNgoaiNgu = "Tiếng Anh";
    }
    if (checkBox2.IsChecked == true)
    {
        strNgoaiNgu = (strNgoaiNgu.Length==0) ? "Tiếng Trung": (strNgoaiNgu+" và Tiếng Trung");
    }
    strMessage += "\n Ngoại ngữ: " + strNgoaiNgu;
    if (comboBox1.SelectedIndex >= 0)//Nếu đã có một mục trong danh sách được chọn
    {
        strMessage += "\n Quê Quán: " + comboBox1.Text;
    }
    MessageBox.Show(strMessage);
}

private void button2_Click(object sender, RoutedEventArgs e)
{
    textBox1.Text = "";
    textBox2.Text = "";
    radioButton1.IsChecked = true;
    radioButton2.IsChecked = false;
    checkBox1.IsChecked = false;
    checkBox2.IsChecked = false;
    comboBox1.SelectedIndex = 0;
}

}

}

```

F5 để chạy ứng dụng. Ta sẽ thu được kết quả tương tự như yêu cầu.

## Câu hỏi ôn tập

### 1. Control là gì?

Control phân tử trên một cửa sổ như các nhãn (Label), hộp soạn thảo (TextBox), nút bấm (Button), hộp danh sách (ListBox, ComboBox),... để hiển thị các thông tin tới người dùng và cho phép người dùng nhập thông tin cần thiết cho chương trình.

### 2. Các control trong WPF được định bằng?

- A. Đặc tả XAML
- B. Mã trình (C# hoặc Visual Basic)

- C. Cả A và B

*Câu trả lời: C*

**3. Các control cơ bản trong WPF bao gồm? (chọn nhiều)**

- A. Label, TextBox, Button
- B. Radio Button, CheckBox, ListBox và ComboBox
- C. Grid, WrapPanel
- D. Cả A, B và C

*Câu trả lời: A và B*

**4. Được phép chọn đồng thời nhiều Radio Button trong cùng nhóm:**

- A. Đúng
- B. Sai

*Câu trả lời: B*

**5. Được phép chọn đồng thời nhiều CheckBox:**

- A. Đúng
- B. Sai

*Câu trả lời: A*

**6. Khi dùng thẻ <TextBox> để tạo hộp soạn thảo, bắt buộc phải kết thúc bằng thẻ đóng </TextBox>:**

- A. Đúng
- B. Sai

*Câu trả lời: B*

## Tài liệu tham khảo

1. Windows Presentation Foundation, URL: <http://msdn.microsoft.com/en-us/library/ms754130.aspx>.
2. Introducing Windows Presentation Foundation, URL: <http://msdn.microsoft.com/en-us/library/aa663364.aspx>.
3. WPF Architecture, URL: <http://msdn.microsoft.com/en-us/library/ms750441.aspx>.
4. WPF Tutorial, URL: <http://dotnetslackers.com/articles/silverlight/WPFTutorial.aspx>.

5. WPF Tutorials, URL: <http://www.wpftutorial.net/>.
6. Windows Presentation Foundation Control, <http://msdn.microsoft.com/en-us/library/ms752069.aspx>