

## Bài 6: Lập trình Windows Form

### 4.1. Làm việc với dự án

#### 4.1.1. Tạo dự án mới

Chọn Start\All Programs\Microsoft Visual Studio. Sau đó chọn File\New\Project.

Trong màn hình hiện ra, thực hiện các công việc sau:

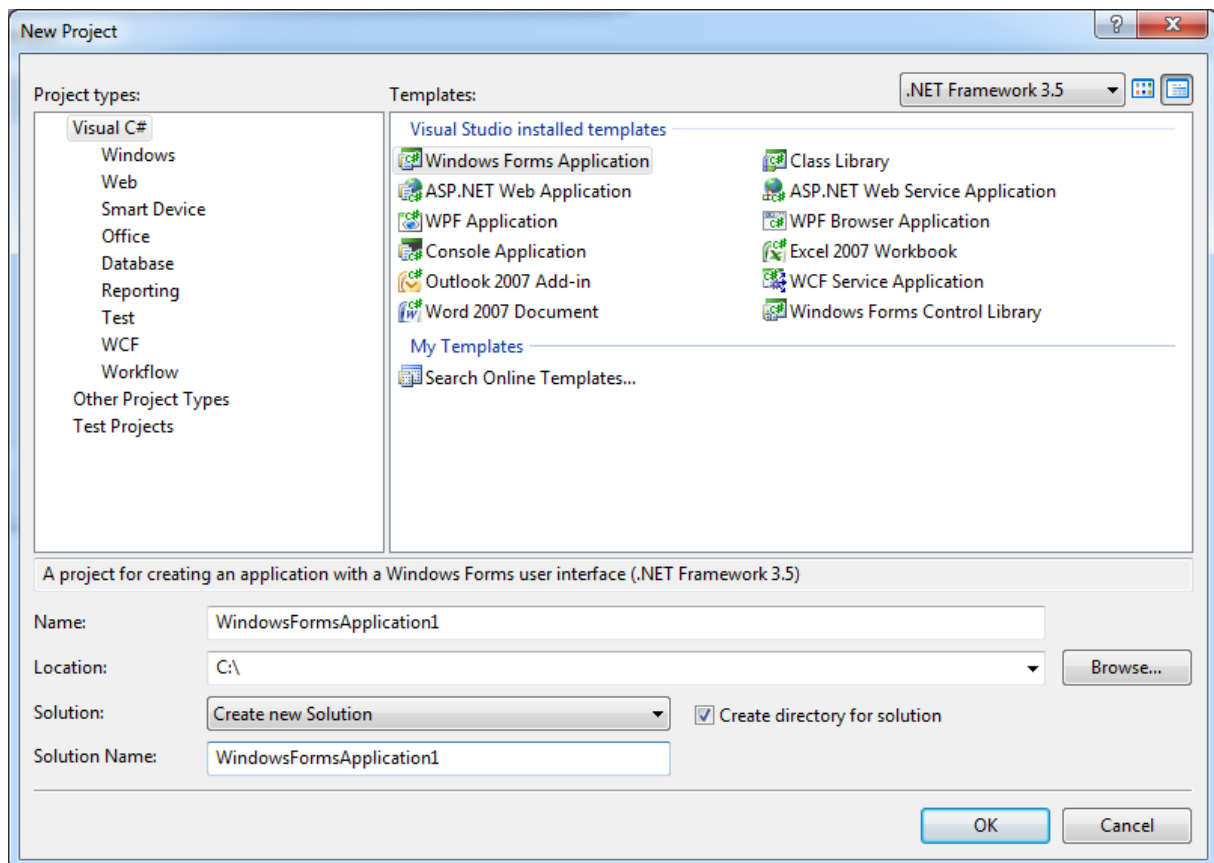
Trong khung Project types, chọn Visual C#.

Trong khung Templates, chọn Windows Application.

Nhập tên dự án vào hộp Name

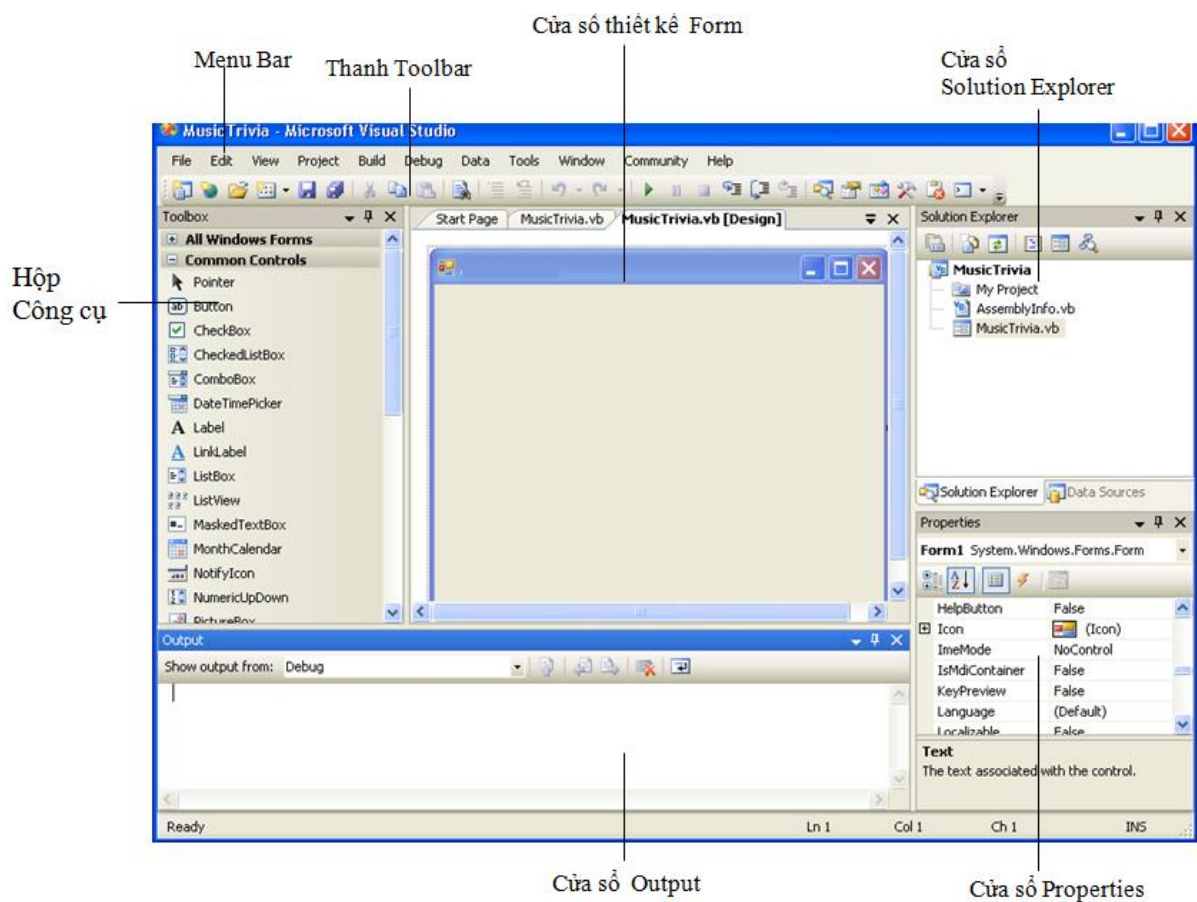
Chọn vị trí chứa dự án trong hộp Location.

Nhấn nút OK để tạo dự án mới.



Hình 4.1 Giao diện tạo đề án mới

Khi đó sẽ xuất hiện giao diện như sau:



Hình 4.2 Giao diện làm việc của Visual Studio với Windows Application project

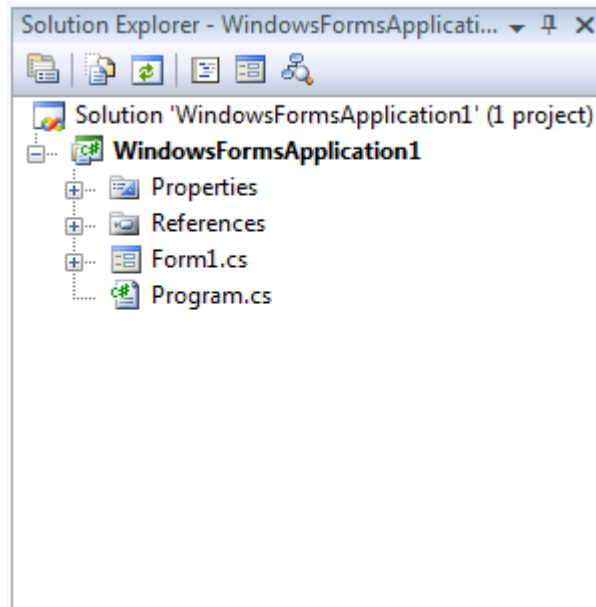
#### 4.1.2. Cửa sổ giải pháp (Solution Explorer) và dự án (Project)

Giải pháp (Solution) là một đối tượng dùng để chứa các dự án (Project).

Một chương trình bao gồm một giải pháp và ít nhất một dự án. Tập tin dự án chứa thông tin đặc biệt liên quan đến một tác vụ lập trình đơn lẻ. Tập tin giải pháp chứa thông tin về một hay nhiều tập tin dự án.

Để mở cửa sổ Solution Explorer, chọn View/Solution Explorer.

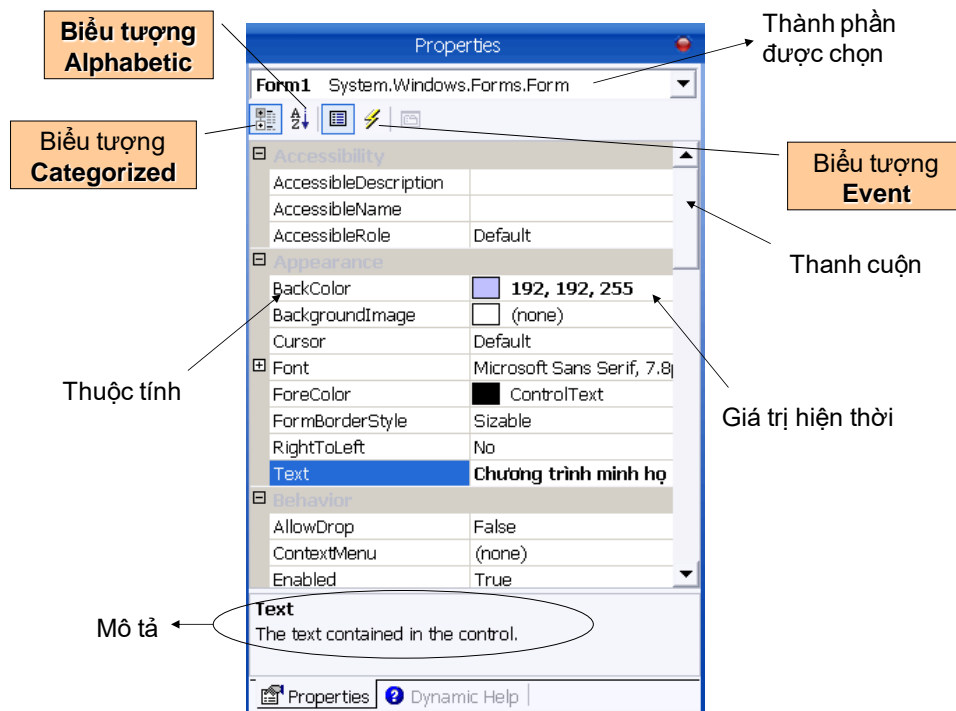
Cửa sổ Solution Explorer chứa tên Project, các lớp tham chiếu đến, các tập tin.



Hình 4.3 Cửa sổ Solution Explorer

#### 4.1.3. Cửa sổ thuộc tính (Properties)

Cửa sổ Properties trình bày danh sách các thuộc tính của đối tượng đang được chọn. Để đóng/mở cửa sổ properties chọn View/Properties Windows hoặc nhấn F4.



Hình 4.4 Cửa sổ thuộc tính (Properties)

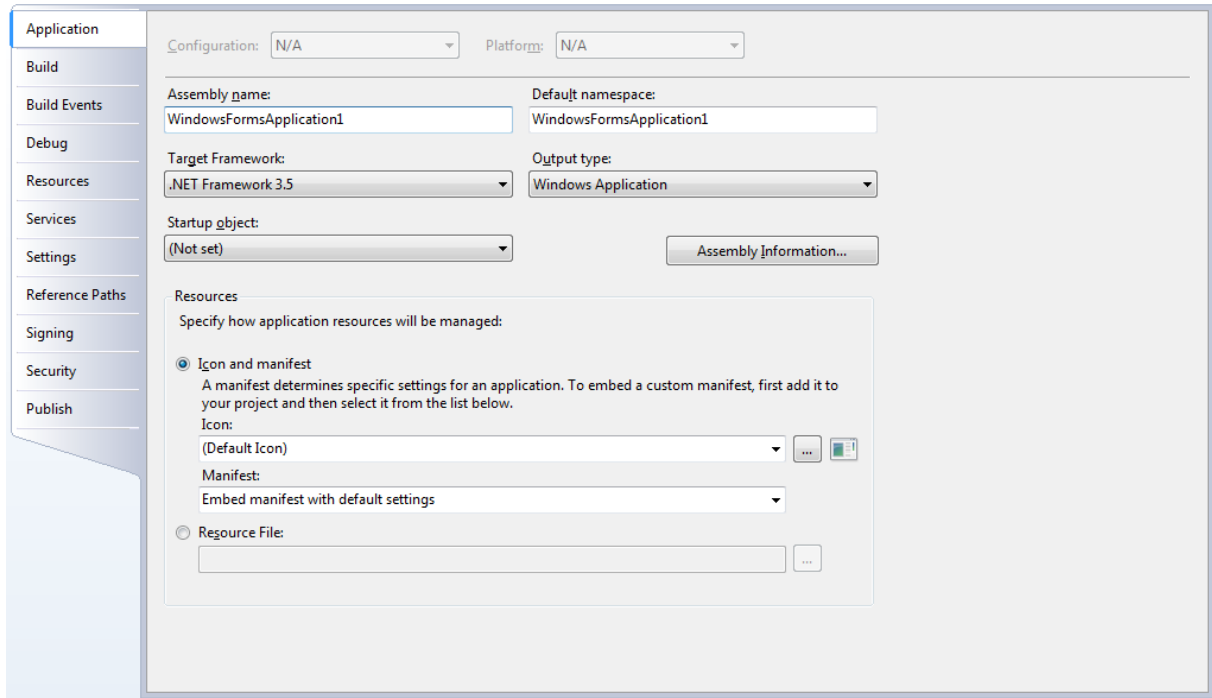
Trong đó có các kiểu sắp xếp:

- Categorized: Sắp xếp thuộc tính theo từng loại.
- Alphabetical: Sắp xếp thuộc tính theo bảng chữ cái.
- Events: Để riêng sự kiện đối với người sử dụng.

#### 4.1.4. Cửa sổ thuộc tính của dự án

Mỗi dự án đều có các thuộc tính khác nhau, tùy thuộc vào cấu hình mà ta chọn.

Để mở cửa sổ thuộc tính của dự án, chọn từ thực đơn Project\Tên dự án hoặc nhấn phải chuột vào tên của dự án rồi chọn Properties.



Hình 4.5 Cửa sổ thuộc tính của dự án.

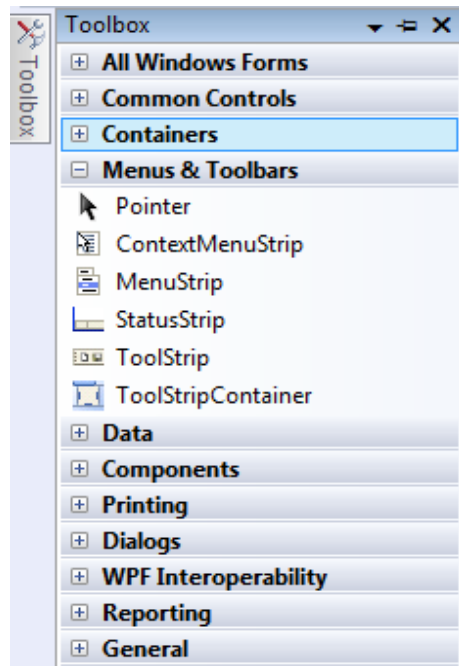
Trong cửa sổ này ta có thể:

- Chọn phương thức khởi động cho ứng dụng.
- Chỉ định tệp tin biểu tượng.
- Đặt tên cho file Assembly (Exe, Dll).
- Thông tin diễn giải cho tệp tin Assembly: Tựa đề, chú giải, tên sản phẩm, bản quyền, phiên bản ...

#### 4.1.5. Hộp công cụ

Hộp công cụ chứa các đối tượng phục vụ cho việc thiết kế giao diện.

Để hiển thị hộp công cụ chọn thực đơn View/ToolBox.

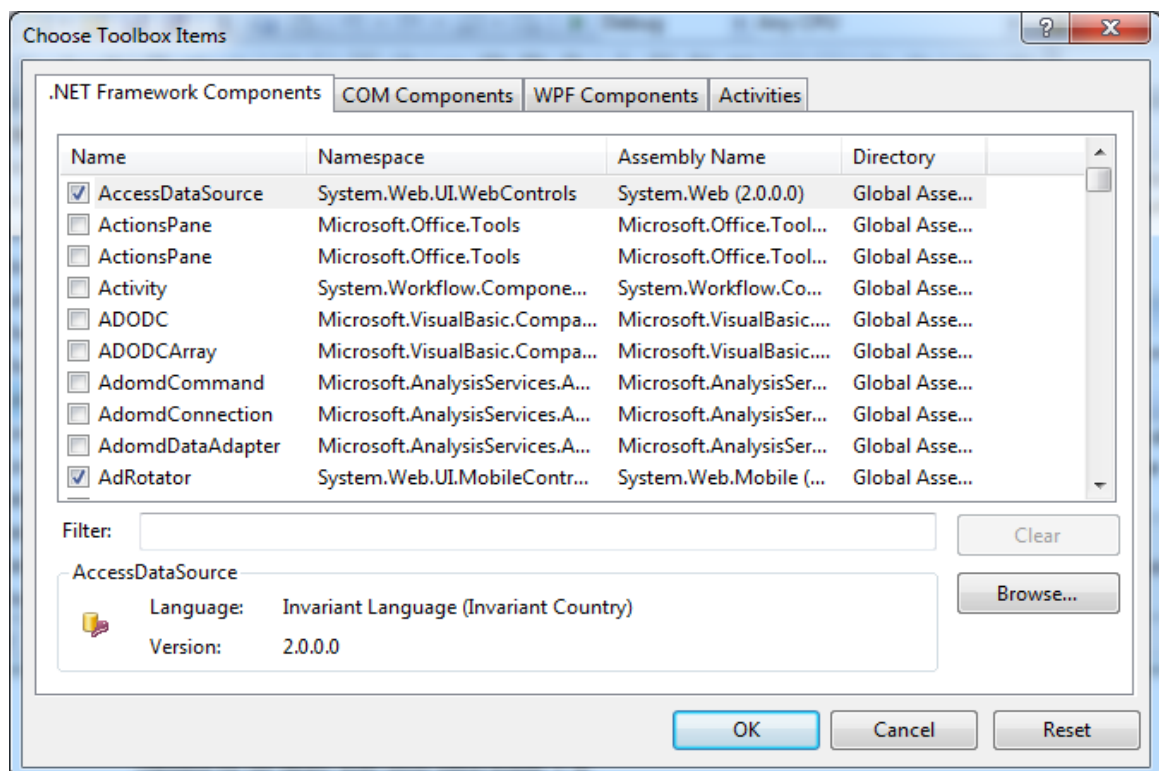


Hình 4.6 Hộp công cụ

Để đưa một điều khiển vào trong Form thiết kế:

- Kích chọn vào biểu tượng bên trái của điều khiển trong hộp công cụ.
- Di chuyển vào Form, kéo chuột để vẽ.

Thêm điều khiển vào hộp công cụ: Nhấn phải chuột vào Toolbox/ Chọn “Choose Item”, cửa sổ liệt kê danh sách các .Net Framework Component xuất hiện như hình dưới đây:



Hình 4.7 Thêm điều khiển vào hộp công cụ

Tích chọn các đối tượng cần thêm vào hộp công cụ và nhấn nút OK.

#### 4.1.6. Biên dịch và chạy chương trình

Nhấn phím F5 (hoặc chọn Debug\Start Debugging) để biên dịch và chạy chương trình. Chọn Debug\Stop Debugging để dừng chương trình đang chạy.

#### 4.1.7. Ví dụ

Viết chương trình hiển thị lời chào.

##### **Bước 1:**

Tạo dự án mới.

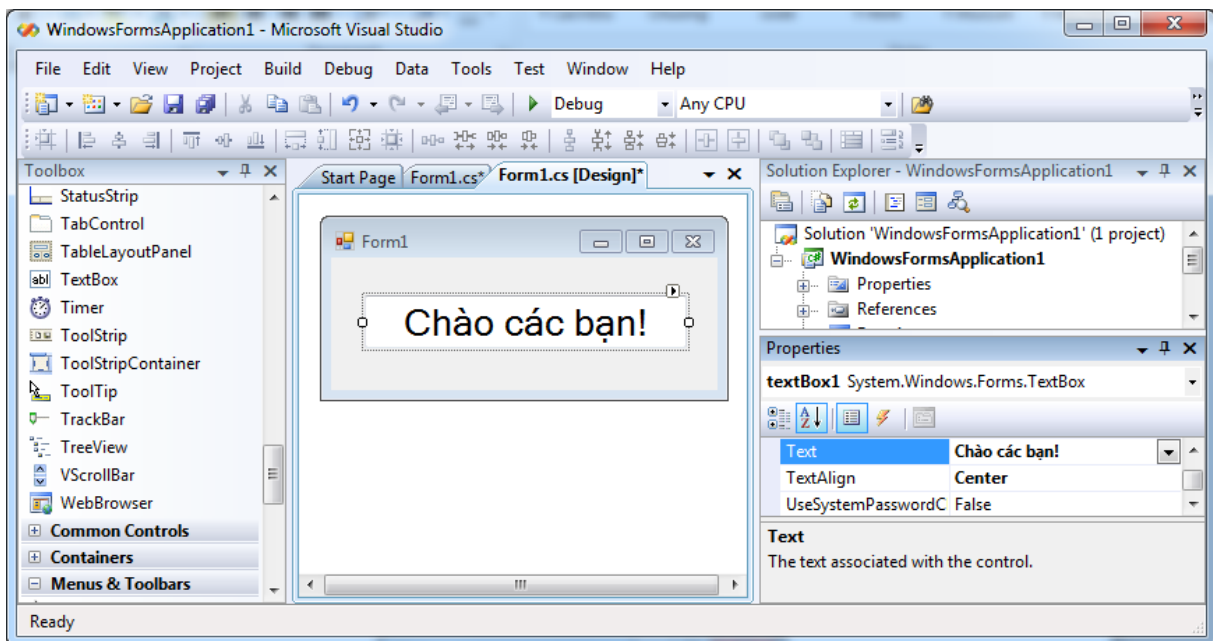
Trong cửa sổ Toolbox, nhấn chọn vào đối tượng TextBox, di chuyển chuột vào Form và vẽ.

##### **Bước 2:**

Trên Form, nhấn chọn TextBox.

Trong cửa sổ Properties, chọn thuộc tính Font, chọn size 20.

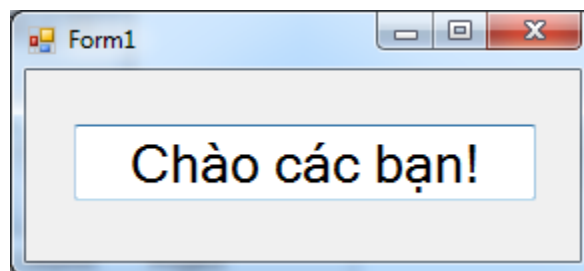
Chọn thuộc tính Text, nhập vào văn bản: “Chào các bạn!”



Hình 4.8 Ví dụ đầu tiên

##### **Bước 3:**

Nhấn phím F5 (hoặc chọn Debug\Start Debugging) để biên dịch và chạy chương trình, kết quả như hình dưới đây:



Hình 4.9 Kết quả chương trình

## 4.2. Form

### 4.2.1. Khái niệm

Form là giao diện giao tiếp giữa người dùng và ứng dụng thông qua các đối tượng trên Form. Một ứng dụng có thể có nhiều Form.

Các dạng:

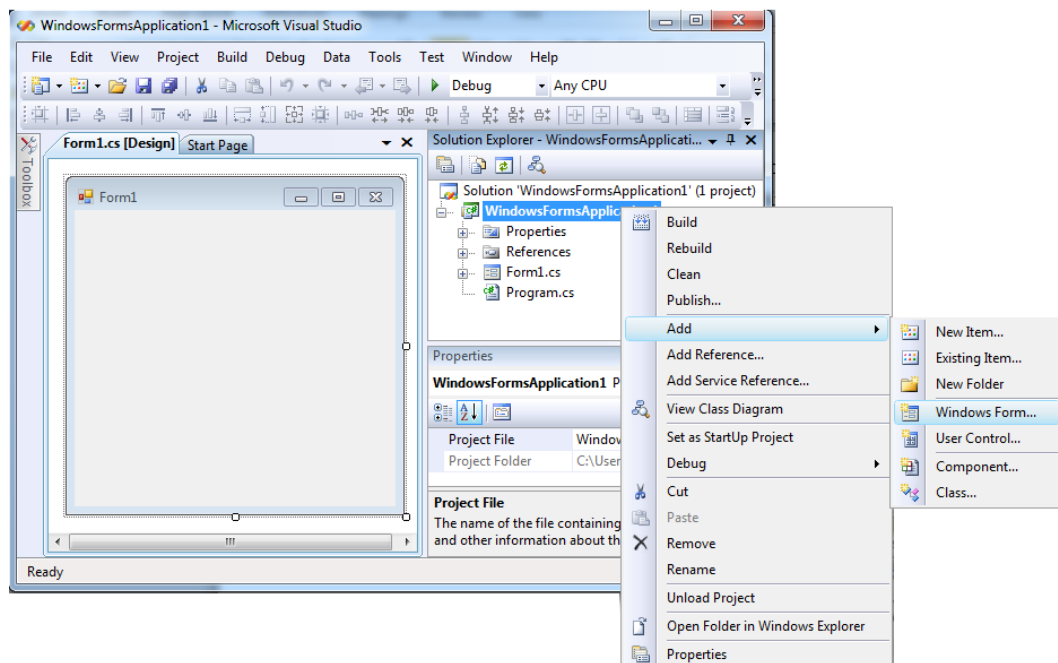
- Normal Form: Là Form bình thường, trên đó ta có thể bố trí các đối tượng phục vụ giao tiếp giữa người dùng và ứng dụng. Normal Form không thể chứa Form khác.
- MDI Form (Multiple Documents Interface Form): Là Form cho phép chứa các Form khác. Thông thường, trên Form này ta bố trí menu, thanh công cụ để gọi Form khác hoặc thi hành công việc đơn lẻ, thanh trạng thái để hiển thị trạng thái của ứng dụng.

Chuyển đổi dạng Form: Thiết lập giá trị thuộc tính IsMdiContainer:

- *True*: Chuyển từ Normal Form sang MDI Form.
- *False*: Chuyển từ MDI Form sang Normal Form.

Thêm một Form mới:

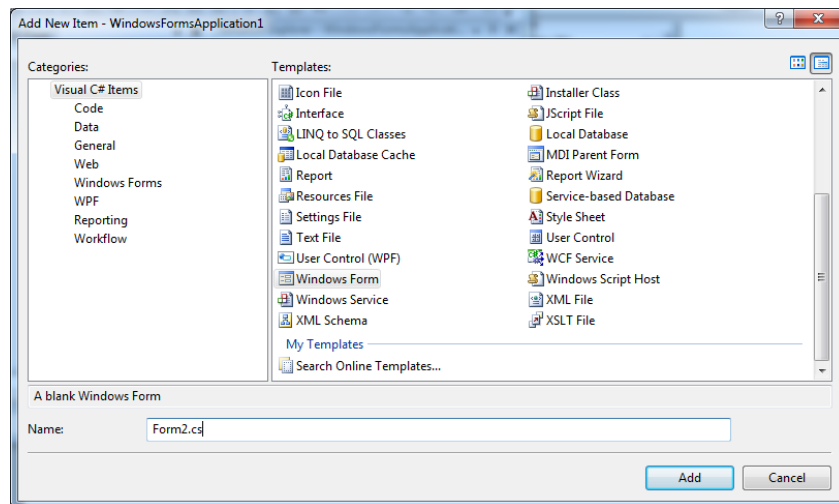
- Nhấn phải chuột vào Project/ Chọn Add/Chọn Windows Form...



Hình 4.10 Thêm form vào dự án

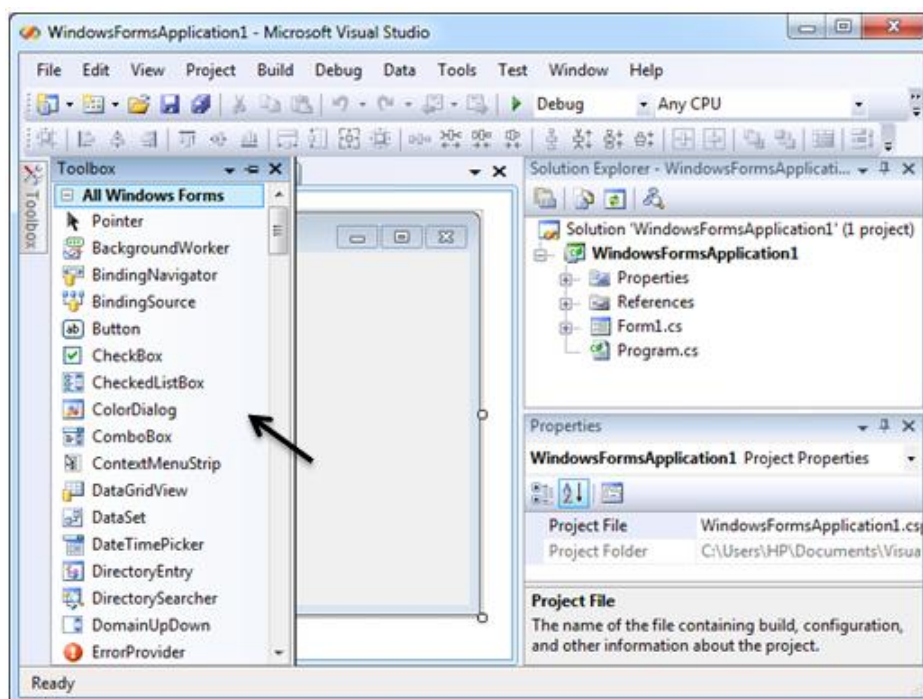
- Nhập tên Form vào hộp Name/ Nhấn Add.





Hình 4.11 Thiết lập tên form

Các đối tượng có thể bố trí trên Form: Các đối tượng chứa trong cửa sổ Toolbox.



Hình 4.12 Thêm các đối tượng vào form

#### 4.2.2. Một số thuộc tính

Thuộc tính	Mô tả
Name	Tên của form. Tên không được trùng lặp.
AcceptButton	Nút được chọn nhấn khi ấn phím <i>Enter</i> .
CancelButton	Nút được chọn nhấn khi ấn phím <i>Esc</i> .
BackgroundImage	Ảnh nền của form.
Font	Font hiển thị của form và font chữ ngầm định của các đối tượng của form.
FormBorderStyle	Kiểu đường viền của form.
ForeColor	Màu chữ của Form và màu chữ của các đối tượng trên form.



Text	Chuỗi tiêu đề form.
MaximizeBox	Có/không nút phóng to.
MinimizeBox	Có/không nút thu nhỏ.
StartPosition	Vị trí bắt đầu khi hiện form
WindowState	Trạng thái ban đầu của form khi form được nạp.
Icon	Biểu tượng của form.
ShowIcon	Cho phép/không cho phép hiển thị biểu tượng đã thiết lập trong thuộc tính Icon.
ShowInTaskbar	Cho phép/không cho phép form nằm trên TaskBar khi nhấn nút thu nhỏ.
Opacity	Độ trong suốt của form theo %.
Width	Chiều rộng của form.
Height	Chiều cao của form.

Thiết lập giá trị cho thuộc tính:

Trong khi thiết kế: Sử dụng cửa sổ Properties để thiết lập giá trị của thuộc tính.

Trong khi thực thi ứng dụng: Đối tượng.Tên thuộc tính = giá trị;

Ví dụ:

```
Form1 frm = new Form1();
frm.Text = "Form1";
```

#### 4.2.3. Một số phương thức

Phương thức	Mô tả
Close	Đóng form và giải phóng các tài nguyên.
Hide	Ẩn form và không giải phóng tài nguyên.
Activate	Kích hoạt form.
Show	Hiển thị form lên màn hình.
ShowDialog	Hiện form lên màn hình dạng Modal.

Gọi phương thức: Đối tượng.Tên phương thức();

Ví dụ:

```
Form1 frm = new Form1();
frm.Show();
```

#### 4.2.4. Một số sự kiện

Sự kiện	Mô tả
Load	Xảy ra khi nạp form. Thông thường, trong sự kiện này thực thi các lệnh khai báo, khởi tạo, trình bày giao diện,...
FormClosing	Xảy ra khi đang đóng form. Thông thường, trong sự kiện này thực thi các công việc cần thiết trước khi form đóng.

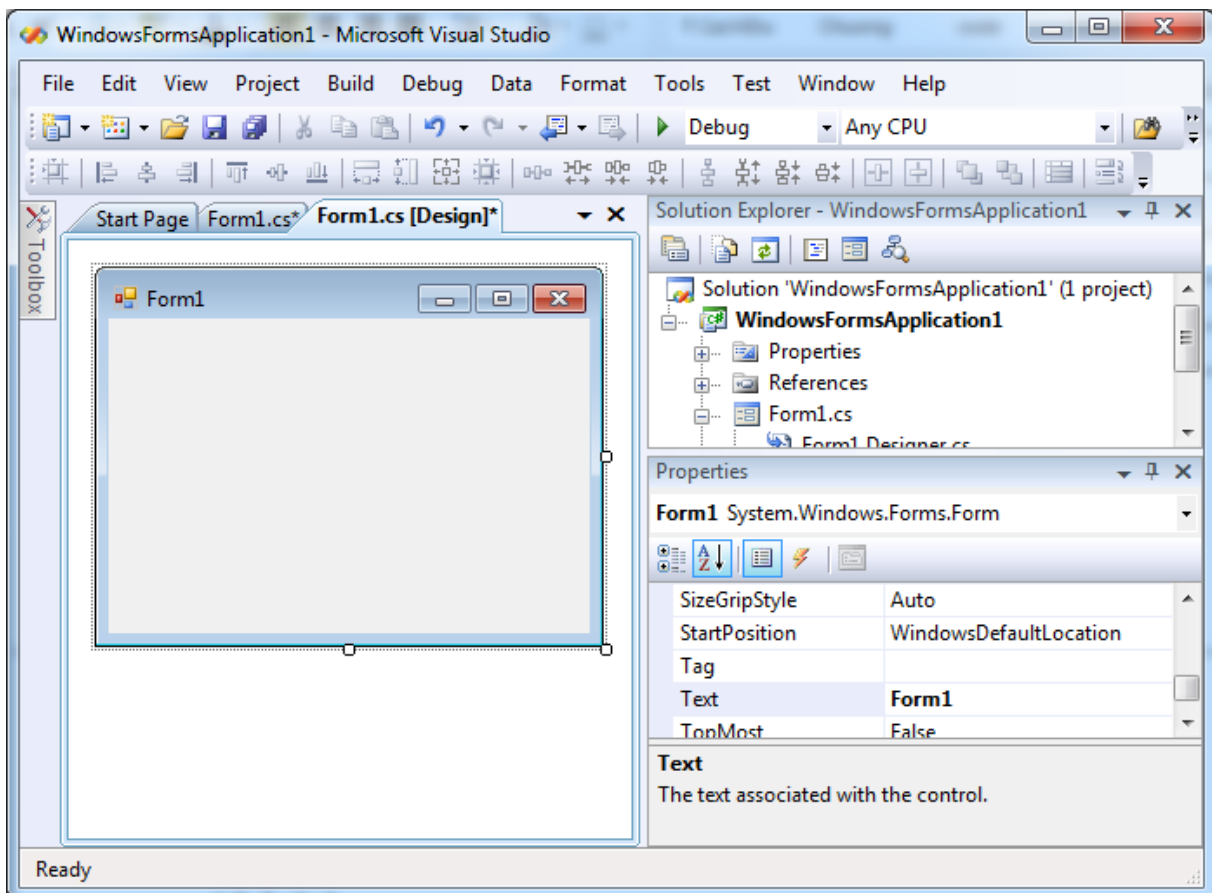
FormClosed	Xảy ra khi form đã đóng.
Click	Xảy ra khi nhấn chuột trên vùng làm việc của form.
Activated	Xảy ra khi form được kích hoạt.
KeyPress	Xảy ra khi một phím được nhấn.
Resize	Xảy ra khi kích thước của form thay đổi.
Deactivate	Xảy ra khi form khác form hiện thời được kích hoạt.

#### 4.2.5. Ví dụ

Viết chương trình thực hiện các công việc:

- Khi form được nạp thì thay đổi tiêu đề của form thành “Xin chào”.
- Khi nhấn chuột lên form thì thông báo xảy ra sự kiện click.
- Khi thay đổi kích thước thì thông báo kích thước của form.

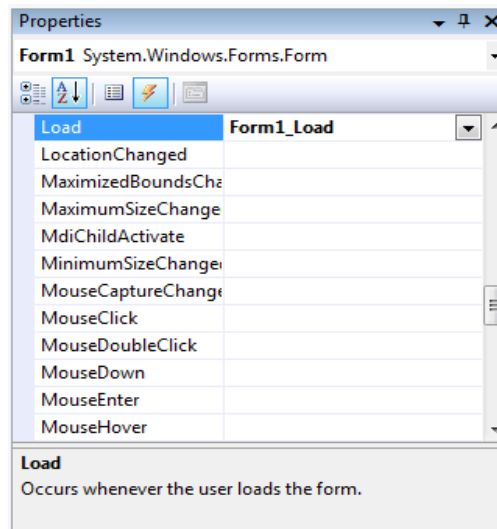
**Bước 1:** Thiết kế giao diện như hình sau:



Hình 4.13 Thiết kế giao diện

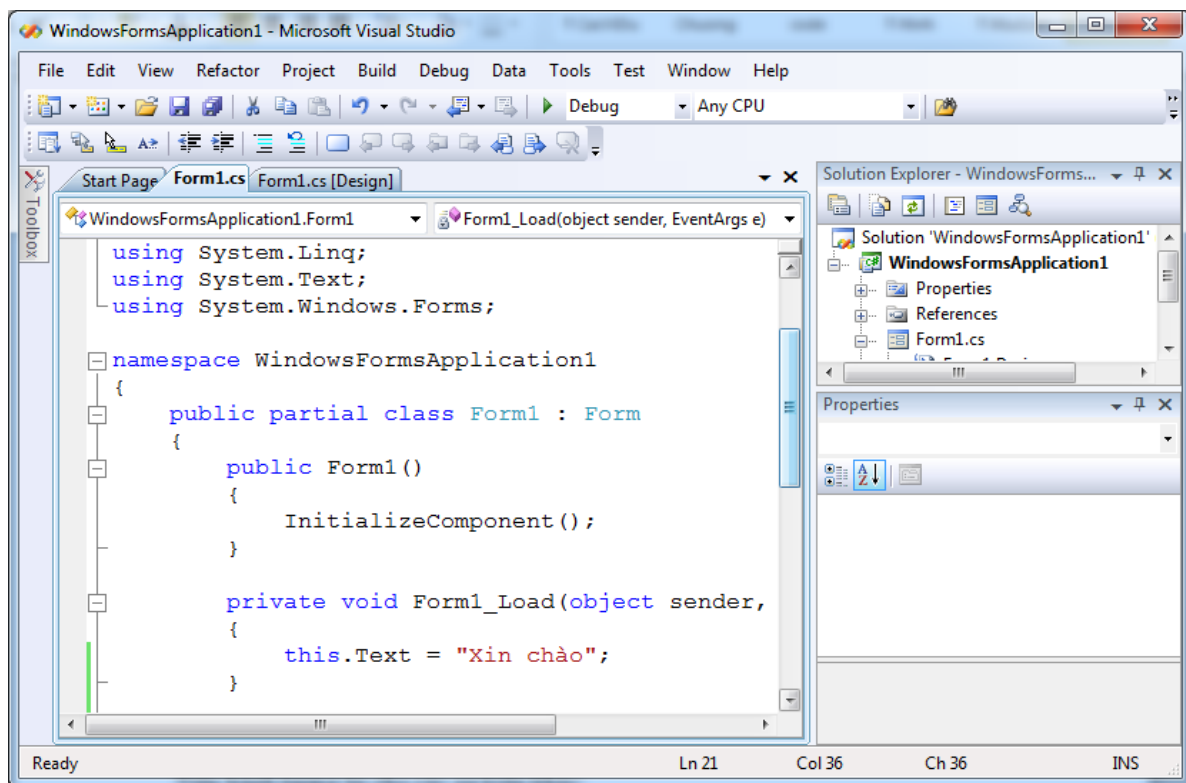
**Bước 2:** Viết mã

Viết mã cho sự kiện Form\_Load: Trong cửa sổ Properties, chọn biểu tượng Event. Duyệt tìm sự kiện Load và nhấn đúp chuột, như hình dưới đây:



Hình 4.14 Lựa chọn sự kiện

Khi đó sẽ mở ra cửa sổ code, tiến hành viết mã lệnh như hình dưới đây:



Hình 4.15 Viết mã lệnh

Tiến hành tương tự cho các sự kiện khác:

```
Private void Form1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Xảy ra sự kiện Click");
}
Private void Form1_Resize(object sender, EventArgs e)
{
    string height = "Chiều cao của form là: " + this.Height.ToString();
    string width = "Chiều rộng của form là: " + this.Width.ToString();
    MessageBox.Show(height+width);
}
```

**Bước 3:** Nhấn F5 để chạy chương trình

### 4.3. Button

#### 4.3.1. Khái niệm

Button là nút lệnh được bố trí trên form để thực hiện một công việc nào đó.

#### 4.3.2. Một số thuộc tính

Thuộc tính	Mô tả
Name	Tên của button.
FlatStyle	Kiểu của button.
Image	Ảnh hiển thị trên Button.
Text	Văn bản xuất hiện trên Button.
Enable	Có/ không cho phép đáp ứng các sự kiện.
BackColor	Màu nền của Button.
ForeColor	Màu chữ của văn bản trên Button.
Visible	Có/ không cho phép Button lộ diện.

#### 4.3.3. Một số sự kiện

Sự kiện	Mô tả
MouseClick	Xảy ra khi người sử dụng nhấn chuột vào Button.
EnabledChanged	Xảy ra khi thuộc tính Enable của Button thay đổi.

### 4.4. Label

#### 4.4.1. Khái niệm Label

Label là đối tượng hiển thị văn bản kết hợp với hình ảnh. Thông thường, label dùng để chú giải cho đối tượng nhập liệu khác trên Form.

#### 4.4.2. Một số thuộc tính của Label

Thuộc tính	Mô tả
Name	Tên của Label.
AutoSize	Tự động thay đổi kích thước của Label để phù hợp với văn bản.
Fonts	Font chữ.
ForeColor	Màu chữ.
Image	Ảnh xuất hiện trên Label.
Text	Văn bản xuất hiện trên Label.
TextAlign	Căn lề văn bản.
BorderStyle	Kiểu đường viền của Label.

## 4.5. TextBox

### 4.5.1. Khái niệm

TextBox là đối tượng này cho phép người sử dụng nhập dữ liệu đầu vào từ bàn phím.

### 4.5.2. Một số thuộc tính

Thuộc tính	Mô tả
Name	Tên của TextBox.
Enable	Có/Không cho phép thao tác với TextBox.
Multiline	Có/Không cho phép nhập dữ liệu nhiều dòng.
ReadOnly	Có/Không cho phép sửa nội dung của TextBox.
Text	Văn bản nhập/ hiển thị.
BorderStyle	Kiểu đường viền.
CharacterCasing	Định dạng chuỗi nhập vào chuyển sang chữ hoa, chữ thường hay theo mặc định.
Maxlength	Số lượng ký tự lớn nhất cho phép nhập.
ScrollBars	Có/không cho phép xuất hiện thanh cuộn khi thuộc tính Multiline thiết lập là True.
TextAlign	Căn lề văn bản.
WordWrap	Tự động xuống dòng nếu chuỗi giá trị dài hơn kích thước của đối tượng.
PasswordChar	Sử dụng trong trường hợp nhập mật khẩu. Thay bởi hiển thị văn bản thật thì sẽ hiển thị ký tự thay thế được thiết lập trong thuộc tính này.

### 4.5.3. Một số phương thức

Phương thức	Mô tả
Clear	Xóa văn bản trong TextBox.
Focus	Đưa tiêu điểm đến TextBox.

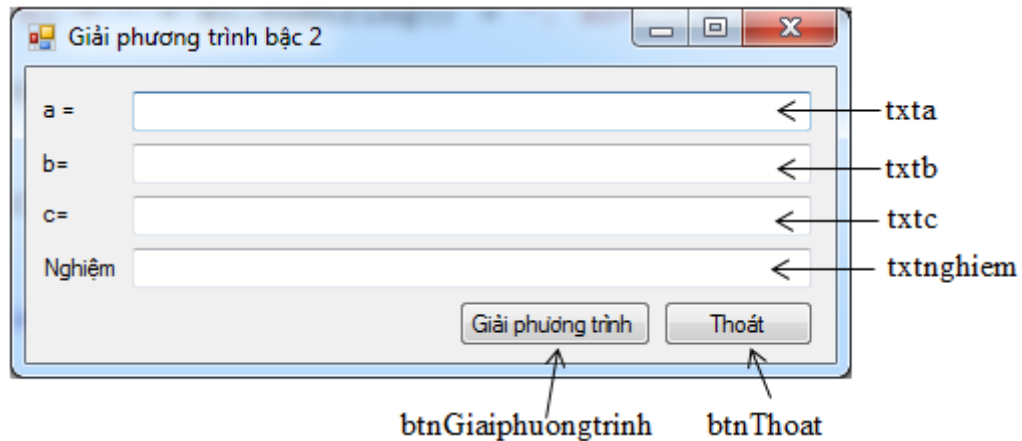
### 4.5.4. Một số sự kiện

Sự kiện	Mô tả
TextChanged	Xảy ra khi nội dung trong TextBox thay đổi.
KeyDown	Xảy ra một phím bất kỳ được nhấn giữ.
KeyUp	Xảy ra khi một phím đang nhấn được nhả ra.
KeyPress	Xảy ra khi một phím bất kỳ được nhấn.

### 4.5.5. Ví dụ

Viết chương trình giải phương trình bậc 2

### Bước 1: Thiết kế giao diện



Hình 4.16 Kết quả chương trình

### Bước 2: Viết mã

#### Mã lệnh nút Giải phương trình:

```
private void btnGiaiphuongtrinh_Click(object sender, EventArgs e)
{
    double a = double.Parse(txta.Text);
    double b = double.Parse(txtb.Text);
    double c = double.Parse(txtc.Text);
    double delta = b * b - 4 * a * c;
    string nghiem = "";
    if (a != 0)
    {
        if (delta < 0) nghiem = "Phương trình vô nghiệm";
        else if (delta == 0)
            nghiem = "x1 = x2 = " + (-b / (2 * a)).ToString();
        else
        {
            double x1 = (-b + Math.Sqrt(delta)) / (2 * a);
            double x2 = (-b - Math.Sqrt(delta)) / (2 * a);
            nghiem = "x1 = " + x1.ToString() + ". x2 = " + x2.ToString();
        }
    }
    if (a == 0)
    {
        if (b != 0)
            nghiem = "x = " + (-c/b).ToString();
        else if (c == 0)
            nghiem = "Phương trình vô số nghiệm";
        else
            nghiem = "Phương trình vô nghiệm";
    }
    txtngkiem.Text = nghiem;
}
```

#### Mã lệnh nút Thoát:

```
private void btnThoat_Click(object sender, EventArgs e)
{
}
```

```

        DialogResult dlr = MessageBox.Show("Bạn có thực sự muốn thoát không", "Thông báo", MessageBoxButtons.YesNo);
        if (dlr == DialogResult.Yes) this.Close();
    }

```

**Bước 3:** Nhấn F5 để chạy chương trình

## 4.6. ListBox

### 4.6.1. Khái niệm

ListBox là đối tượng cho phép xem và chọn các dòng dữ liệu.

### 4.6.2. Một số thuộc tính

Thuộc tính	Mô tả
Name	Tên của ListBox.
Datasource	Tập dữ liệu điền vào ListBox.
DisplayMember	Tên của trường tương ứng với chuỗi trình bày trên ListBox.
MultiColumn	Có/Không chia ListBox thành nhiều cột.
ColumnWidth	Chiều rộng của mỗi cột.
Items	Mảng các dòng trong ListBox.
SelectionMode	Cho phép chọn một hay nhiều phần tử cùng một lúc.
Sorted	Có/Không sắp xếp dữ liệu trong ListBox. Ngầm định là False.
ValueMember	Giá trị ứng với khóa.
SelectedItem	Trả về phần tử được chọn.
SelectedItems	Trả về tập phần tử được chọn.
SelectedValue	Lấy giá trị phần tử kiểu đối tượng đang chọn.
SelectedIndex	Gán hay lấy giá trị chỉ số của phần tử đang chọn.
SelectedIndices	Trả về một mảng các chỉ số của các dòng được chọn.

### 4.6.3. Một số phương thức

Phương thức	Mô tả
GetSelected(index)	Trả về True nếu dòng thứ index được chọn, ngược lại trả về False.
Add	Thêm một phần tử vào ListBox.
RemoveAt(index):	Xoá phần tử tại vị trí index.
Clear	Xoá tất cả các phần tử.

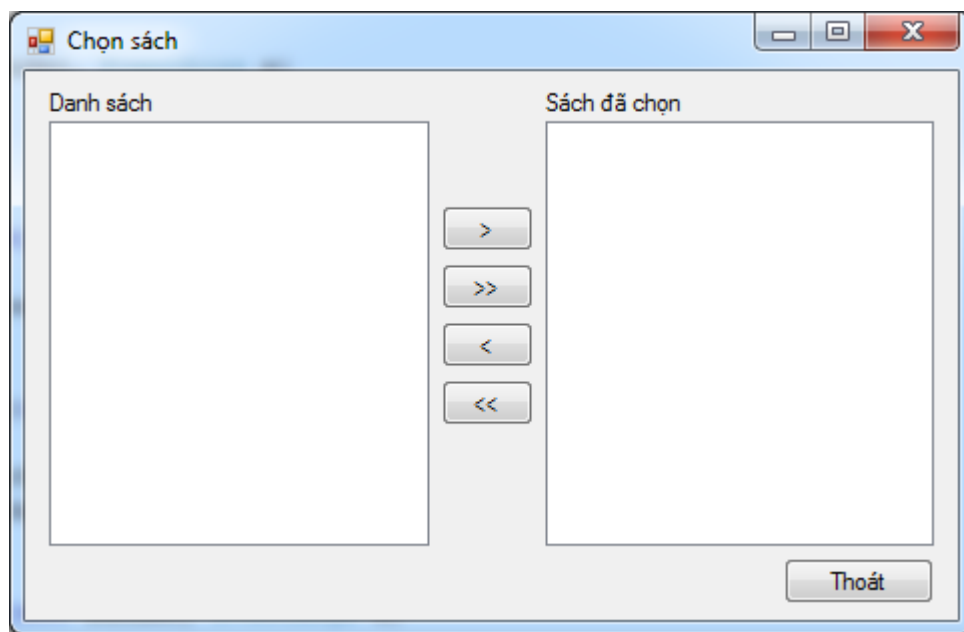
### 4.6.4. Một số sự kiện



Sự kiện	Mô tả
MouseClicked	Xảy ra khi người sử dụng nhấn chuột.
MouseDoubleClick	Xảy ra khi người sử dụng nhấn đúp chuột.
SelectedIndexChanged	Xảy ra khi chỉ mục của phần tử bị thay đổi.
SelectedValueChanged	Xảy ra khi giá trị của phần tử thay đổi.
EnabledChange	Xảy ra khi thuộc tính Enable thay đổi giá trị.

#### 4.6.5. Ví dụ

**Bước 1:** Thiết kế giao diện như hình dưới đây:



Hình 4.17 Thiết kế giao diện

**Bước 2:** Thiết lập thuộc tính Name cho các điều khiển:

- Nút >: btnthemdonle
- Nút >>: btnthemtoanbo
- Nút <: btnxoadonle
- Nút <<: btnxoatoanbo
- Hộp Danh sách: lstdanhsach
- Hộp Sách đã chọn: lstsachchon
- Hộp danh sách và sách đã chọn thiết lập thuộc tính Sorted là True.

**Bước 3:** Viết mã

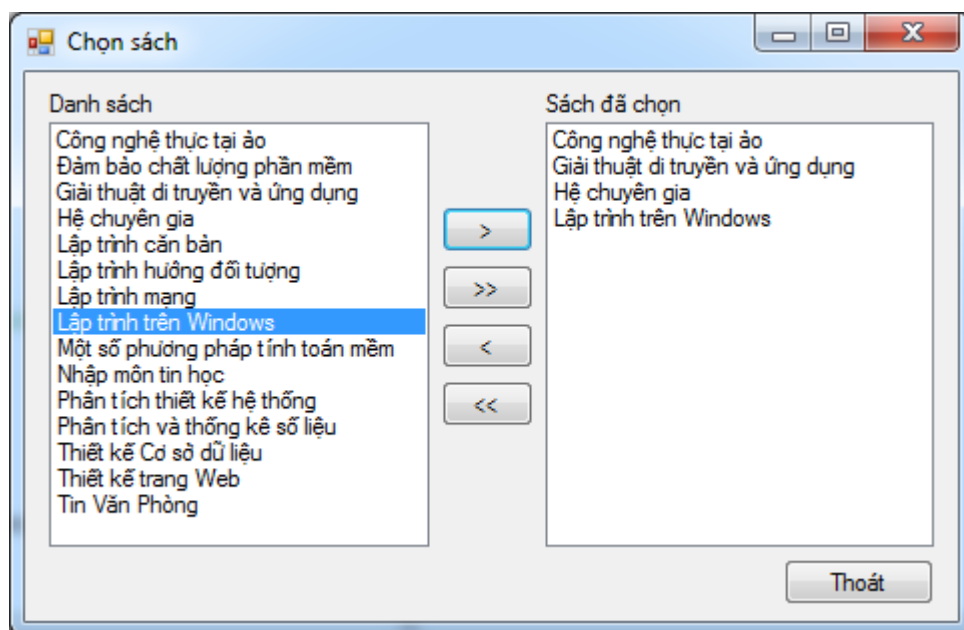
```
Private void Form1_Load(object sender, EventArgs e)
{
    lstdanhsach.Items.Add("Nhập môn tin học");
    lstdanhsach.Items.Add("Tin Văn Phòng");
    lstdanhsach.Items.Add("Lập trình mạng");
    lstdanhsach.Items.Add("Lập trình trên Windows");
    lstdanhsach.Items.Add("Công nghệ thực tại ảo");
}
```

```

        lstdanhsach.Items.Add("Phân tích thiết kế hệ thống");
        lstdanhsach.Items.Add("Lập trình hướng đối tượng");
        lstdanhsach.Items.Add("Thiết kế trang Web");
        lstdanhsach.Items.Add("Đảm bảo chất lượng phần mềm");
        lstdanhsach.Items.Add("Giải thuật di truyền và ứng dụng");
        lstdanhsach.Items.Add("Hệ chuyên gia");
        lstdanhsach.Items.Add("Lập trình căn bản");
        lstdanhsach.Items.Add("Một số phương pháp tính toán mềm");
        lstdanhsach.Items.Add("Phân tích và thống kê số liệu");
        lstdanhsach.Items.Add("Thiết kế Cơ sở dữ liệu");
    }
    Private void btnchondonle_Click(object sender, EventArgs e)
    {
        lstsachchon.Items.Add(lstdanhsach.SelectedItem);
    }
    Private void btntchontoanbo_Click(object sender, EventArgs e)
    {
        for (int i = 0; i < lstdanhsach.Items.Count; i++)
            lstsachchon.Items.Add(lstdanhsach.Items[i]);
    }
    Private void btnxoadonle_Click(object sender, EventArgs e)
    {
        lstsachchon.Items.Remove(lstsachchon.SelectedItem);
    }
    Private void btnxoatoanbo_Click(object sender, EventArgs e)
    {
        lstsachchon.Items.Clear();
    }
    Private void btnthoat_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}

```

**Bước 4:** Chạy chương trình



Hình 4.18 Kết quả chương trình

## 4.7. ComboBox

### 4.7.1. Khái niệm

Là sự kết hợp của TextBox và ListBox. Điều khiển ComboBox được dùng để trình bày danh sách phần tử, và cho phép người dùng chọn một mục trong danh sách các mục đổ xuống hoặc nhập trực tiếp từ hộp TextBox.

### 4.7.2. Một số thuộc tính

Thuộc tính	Mô tả
Name	Tên của ComboBox.
Datasource	Tập dữ liệu điền vào điều khiển.
DisplayMember	Tên của trường tương ứng với chuỗi trình bày trên điều khiển.
DropDownStyle	Kiểu danh sách đổ xuống của combobox: + DropDown: Cho phép thêm mới chuỗi (mặc định). + DropDownList: Chỉ cho phép chọn trong danh sách. + Simple: Dạng danh sách.
Items	Mảng các phần tử trong ComboBox.
MaxDropDownItems	Số phần tử lớn nhất được liệt kê. Mặc định là 8.
ValueMember	Giá trị ứng với khóa.
FlatStyle	Kiểu đường viền của ComboBox
MaxLength	Số ký tự tối đa có thể nhập vào hộp Text của ComboBox
Text	Giá trị ứng với phần tử đang chọn hoặc là giá trị đang nhập vào.
SelectedText	Giá trị dạng chuỗi của phần tử được chọn.
SelectedItem	Giá trị phần tử đang được chọn.
SelectedValue	Gán hay lấy giá trị phần tử kiểu object đang chọn.
SelectedIndex	Chỉ số dòng được chọn. Nếu không chọn có giá trị -1.
Sorted	Có/Không sắp xếp dữ liệu trong ComboBox. Ngầm định là False.

### 4.7.3. Một số phương thức

Phương thức	Mô tả
Add	Thêm một phần tử vào ComboBox.
RemoveAt(index)	Xóa phần tử tại vị trí index.
Clear	Xóa tất cả phần tử trong ComboBox.

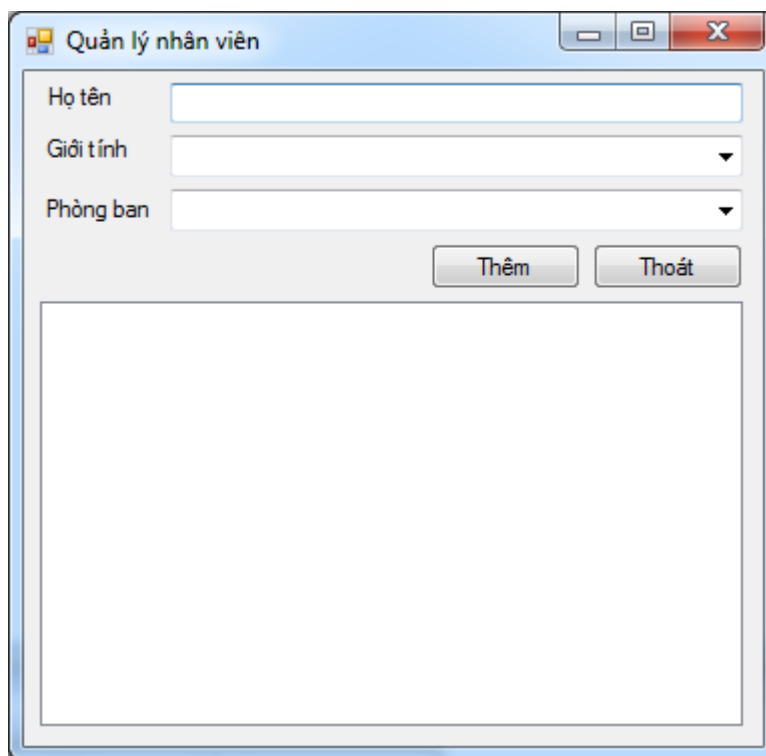
### 4.7.4. Một số sự kiện

Sự kiện	Mô tả
---------	-------

MouseClicked	Xảy ra khi người sử dụng Click chuột trên ComboBox.
MouseDoubleClick	Xảy ra khi người sử dụng Click chuột hai lần liên tiếp
SelectedIndexChanged	Xảy ra khi chỉ mục của phần tử bị thay đổi
SelectedValueChanged	Xảy ra khi giá trị của phần tử được thay đổi
SelectionChangeCommitted	Xảy ra khi người dùng kết thúc quá trình chọn phần tử trên ComboBox.
TextChanged	Xảy ra khi chuỗi trên ComboBox thay đổi
EnabledChange	Xảy ra khi thuộc tính Enable thay đổi giá trị

#### 4.7.5. Ví dụ

**Bước 1:** Thiết kế giao diện



Hình 4.19 Thiết kế giao diện

**Bước 2:** Thiết lập thuộc tính Name cho các điều khiển

- Hộp nhập họ tên: txthoten
- Hộp nhập giới tính: cbgioitinh
- Hộp nhập phòng ban: cbphongban
- Nút thêm: btnthem
- Nút thoát: btnthoat
- Hộp danh sách: lstdanhsach
- Thiết lập thuộc tính Sorted của ComboBox giới tính và phòng ban là True.

**Bước 3:** Viết mã

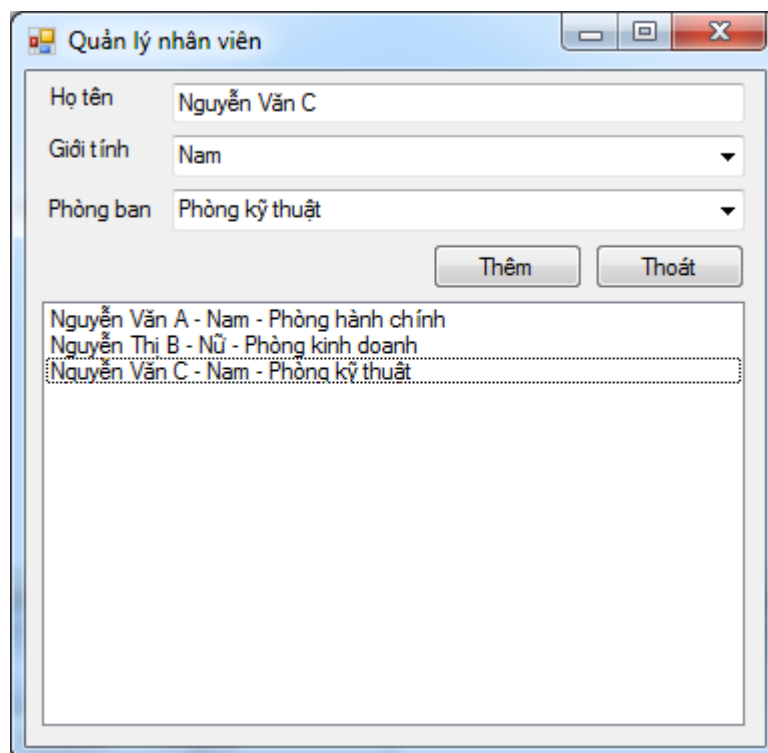
```

Private void Form1_Load(object sender, EventArgs e)
{
    cbogioitinh.Items.Add("Nam");
    cbogioitinh.Items.Add("Nữ");
    cbophongban.Items.Add("Phòng tài vụ");
    cbophongban.Items.Add("Phòng hành chính");
    cbophongban.Items.Add("Phòng kinh doanh");
    cbophongban.Items.Add("Phòng kỹ thuật");
}
Private void btnthem_Click(object sender, EventArgs e)
{
    string st = txthoten.Text + " - " + cbogioitinh.Text + " - " +
    cbophongban.Text;
    lstdanhsach.Items.Add(st);
}

Private void btnthoat_Click(object sender, EventArgs e)
{
    this.Close();
}

```

**Bước 4:** Chạy chương trình



*Hình 4.20 Kết quả chương trình*

## 4.8. CheckBox

### 4.8.1. Khái niệm

CheckBox là đối tượng cho phép chọn/không chọn giá trị và cho phép chọn đồng thời nhiều đối tượng

#### 4.8.2. Một số thuộc tính

Thuộc tính	Mô tả
FlatStyle	Đường viền cho CheckBox.
Appearance	Hình dạng của CheckBox.
Checked	Có giá trị True nếu CheckBox được chọn và ngược lại có giá trị False.
Text	Chuỗi văn bản trên CheckBox.
TextAlign	Căn lề cho văn bản trên CheckBox.

#### 4.8.3. Một số sự kiện

Sự kiện	Mô tả
MouseClick	Xảy ra khi người sử dụng kích chuột vào CheckBox.
CheckedChanged	Xảy ra khi người sử dụng thực hiện việc chọn hoặc bỏ chọn trên CheckBox.
CheckedStateChanged	Xảy ra khi trạng thái Checked của CheckBox thay đổi..

### 4.9. RadioButton

#### 4.9.1. Khái niệm

RadioButton là đối tượng cho phép chọn/không chọn giá trị, tại một thời điểm chỉ có thể chọn một đối tượng. Để chọn nhiều đối tượng phải đặt các đối tượng trong GroupBox hoặc Panel.

#### 4.9.2. Một số thuộc tính

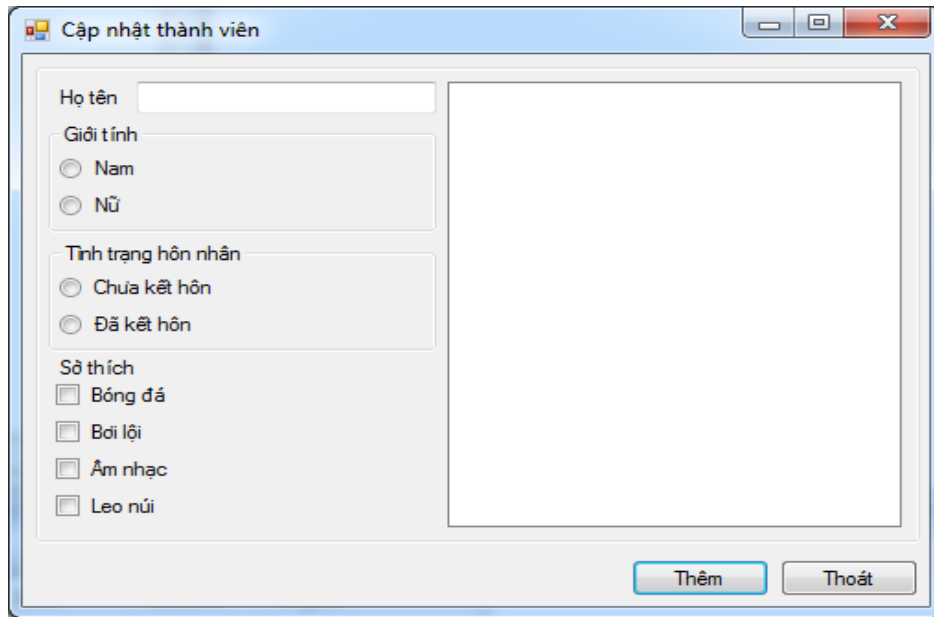
Thuộc tính	Mô tả
FlatStyle	Kiểu đường viền của RadioButton.
Appearance	Hình dạng của RadioButton.
Checked	Trạng thái của RadioButton là được chọn hay chưa được chọn. Nếu được chọn sẽ là true, ngược lại là false.
Text	Văn bản hiển thị trên màn hình.
TextAlign	Căn lề của văn bản của RadioButton.

#### 4.9.3. Một số sự kiện

Sự kiện	Mô tả
MouseClick	Xảy ra khi người sử dụng kích chuột vào CheckBox.
CheckedChanged	Xảy ra khi người sử dụng thực hiện việc chọn hoặc bỏ chọn trên CheckBox.

#### 4.9.4. Ví dụ

##### **Bước 1:** Thiết kế giao diện



*Hình 4.21 Thiết kế giao diện*

##### **Bước 2:** Thiết lập thuộc tính Name cho các điều khiển:

- Hộp họ tên: txthoten.
- Giới tính: rdonam, rdonu.
- Tình trạng hôn nhân: rdochuakethon, rdodakethon.
- Sở thích: chkbongda, chkboiloi, chkamnhac, chkleonui.
- Nút lệnh: btnthem, btnthoat.
- Hộp danh sách: lstdanhsach.

##### **Bước 3:** Viết mã

```
Private void btnthem_Click(object sender, EventArgs e)
{
    string gioitinh = "Giới tính: ";
    if (rdonam.Checked) gioitinh += "Nam";
    if (rdonu.Checked) gioitinh += "Nữ";
    string honnhan = "Tình trạng hôn nhân: ";
    if (rdodakethon.Checked) honnhan += "Đã kết hôn";
    if (rdochuakethon.Checked) honnhan += "Chưa kết hôn";
    string sothich = "Sở thích: ";
    if (chkamnhac.Checked) sothich += "Âm nhạc, ";
    if (chkboiloi.Checked) sothich += "Bơi lội, ";
    if (chkbongda.Checked) sothich += "Bóng đá, ";
    if (chkleonui.Checked) sothich += "Leo núi";
    lstdanhsach.Items.Add("Họ tên: " + txthoten.Text);
    lstdanhsach.Items.Add(gioitinh);
    lstdanhsach.Items.Add(honnhan);
    lstdanhsach.Items.Add(sothich);
}
```



```
Private void btnthoat_Click(object sender, EventArgs e)
{
    this.Close();
}
```

#### **Bước 4:** Chạy chương trình

*Hình 4.22- Kết quả chương trình*

## **4.10. ListView**

### **4.10.1. Khái niệm**

ListView là đối tượng dùng để hiển thị dữ liệu theo các dòng và các cột. Có thể chọn một hoặc nhiều dòng và có thể hiển thị các biểu tượng theo các dòng.

### **4.10.2. Một số thuộc tính**

Tên thuộc tính	Mô tả
CheckBoxes	Có/không xuất hiện các checkbox trên các dòng dữ liệu (ngầm định là False)
Columns	Các cột hiển thị trong chế độ Details.
GridLines	Hiển thị lưới (chỉ hiển thị trong chế độ Details)
Items	Mảng các dòng (ListViewItems) trong ListView.
LargeImageList	Danh sách ảnh (ImageList) hiển thị trên ListView
SmallImageList	Danh sách ảnh (ImageList) hiển thị trên ListView
MultiSelect	Có/Không cho phép chọn nhiều dòng (ngầm định là True)
SelectedItems	Mảng các dòng được chọn
View	Kiểu hiển thị của ListView Icons: Hiển thị danh sách theo các biểu tượng List: Hiển thị danh sách theo một cột

	Details: Hiển thị ListView theo danh sách nhiều cột
--	---

#### 4.10.3. Một số phương thức

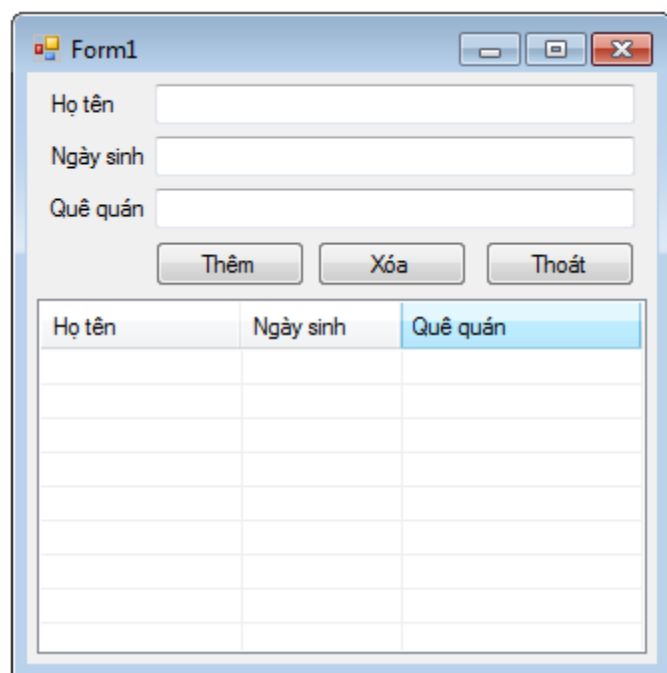
Tên phương thức	Mô tả
Add	Thêm một dòng vào ListView
Clear	Xoá tất cả các dòng của ListView
Remove	Xoá một dòng trong ListView
RemoveAt(index)	Xoá một dòng ở vị trí index

#### 4.10.4. Sự kiện hay dùng

Tên sự kiện	Mô tả
ItemSelectionChanged	Xảy ra khi chọn một dòng

#### 4.10.5. Ví dụ

**Bước 1:** Thiết kế giao diện



Hình 4.27 Thiết kế giao diện

**Bước 2:** Thiết lập thuộc tính

- Thiết lập thuộc tính Name: Hộp Họ tên: txtHoten. Hộp Ngày sinh: txtNgaysinh. Hộp Quê quán: txtQuequan. Nút Thêm: btnThem. Nút Xóa: btnXoa. Nút Thoát: btnThoat. ListView: listView1
- Thiết lập thuộc tính cho listView1:
  - Columns: Thêm 3 cột
  - FullRowSelect: True
  - GridLines: True

- MultiSelect: False
- View: Details

### **Bước 3:** Viết mã

```
int row;
private void btnThem_Click(object sender, EventArgs e)
{
    row = listView1.Items.Count;
    listView1.Items.Add(txtHoten.Text);
    listView1.Items[row].SubItems.Add(txtNgaysinh.Text);
    listView1.Items[row].SubItems.Add(txtQuequan.Text);
}
Private void btnXoa_Click(object sender, EventArgs e)
{
    listView1.Items.RemoveAt(row);
}
Private void btnThoat_Click(object sender, EventArgs e)
{
    this.Close();
}
Private void listView1_ItemSelectionChanged(object sender,
ListViewItemSelectionChangedEventArgs e)
{
    row = e.ItemIndex;
    txtHoten.Text = listView1.Items[row].SubItems[0].Text;
    txtNgaysinh.Text = listView1.Items[row].SubItems[1].Text;
    txtQuequan.Text = listView1.Items[row].SubItems[2].Text;
}
```

### **Bước 4:** Chạy chương trình

Họ tên	Ngày sinh	Quê quán
Nguyễn Văn A	2/3/2000	Hà Nội
Nguyễn Thị B	3/6/2003	Hải Phòng

*Hình 4.28 Kết quả chương trình*