

CHƯƠNG 3

XÂY DỰNG GIAO DIỆN TRỰC QUAN VÀ ỨNG DỤNG TRONG ĐIỀU KHIỂN

GVHD: TS. GVC LÊ THỊ THÚY NGÀ

BỘ MÔN: ĐIỀU KHIỂN HỌC

NỘI DUNG

3.1 Thiết kế giao diện

3.2 Các điều khiển cơ bản

3.3 Các điều khiển phân nhóm

3.4 Giao tiếp ngoại vi

3.5 Một số ứng dụng giao diện trực quan trong kỹ thuật điều khiển

3.2 CÁC ĐIỀU KHIỂN CƠ BẢN

❖ Mỗi một đối tượng trong C# đều có 3 đặc tính là: Thuộc tính - Properties, Phương thức - Methods và Sự kiện – Events.

- Thuộc tính - **Properties**: là tập hợp các đặc tính để mô tả một đối tượng như: tên, chiều cao, chiều rộng, màu chữ, màu nền... Các thuộc tính có thể xác định trong khi thiết kế (Design time) hoặc trong lúc thi hành (Run time).
- Phương thức - **Methodes**: là những đoạn chương trình chứa trong điều khiển, cho điều khiển biết cách thức để thực hiện một công việc nào đó, chẳng hạn làm ẩn sự xuất hiện của một điều khiển (phương thức Hide).
- Sự kiện - **Events**: nếu như thuộc tính mô tả đối tượng, phương thức chỉ ra cách thức đối tượng hành động thì sự kiện là những phản ứng của đối tượng. Khi tạo một chương trình trong C#, lập trình chủ yếu theo sự kiện, lập trình theo cách này có nghĩa là phải biết khi nào sự kiện xảy ra và làm gì khi sự kiện đó xảy ra? Điều này có nghĩa là chương trình chỉ thi hành khi người dùng thực hiện một thao tác nào đó trên giao diện.

3.2 CÁC ĐIỀU KHIỂN CƠ BẢN

❖ Các điều khiển cơ bản trong C#:

- Form
- Label
- TextBox
- Button.
- Combobox
- ListBox
- CheckBox
- RadioButton

■ **Form:**

- Thuộc tính thông dụng của **Form**:

| | |
|-----------------------|---|
| Name | Tên form, thường ký hiệu bắt đầu bởi frm |
| BackColor | Thiết lập màu nền cho Form. |
| BackgroundImage | Thiết lập ảnh nền cho Form. |
| BackgroundImageLayout | Thiết lập chế độ hiển thị ảnh nền trên Form. Tile: hiển thị ảnh từ trên xuống, Center: hiển thị ảnh từ giữa ra, Stretch: dẫn đều ảnh trên Form. |
| Cursor | Thiết lập chế độ hiển thị con trỏ trên Form. |
| Enabled | Nếu nhận giá trị True thì cho phép người dùng tác động lên Form, ngược lại thì nhận giá trị False. |
| Font | Thiết lập kiểu chữ, cỡ chữ cho các điều khiển trên Form. |
| ForeColor | Thiết lập màu chữ cho các điều khiển trên Form. |

| | |
|-----------------|--|
| FormBorderStyle | Thiết lập kiểu đường viền cho Form. Fixed Single: không thể thay đổi kích thước của Form, Sizable: có thể phóng to thu nhỏ và thay đổi kích thước của Form, Sizable ToolWindow: có thể thay đổi kích thước của Form... |
| Icon | Thiết lập biểu tượng cho Form (các tệp ảnh có đuôi .ico). |
| MainMenuStrip | Gắn kết Form với Menu. |
| Opacity | Thiết lập độ trong suốt cho nền của Form, nếu độ trong suốt < 100% thì Form sẽ dần trở nên trong suốt có thể nhìn xuyên qua thấy những gì nằm bên dưới Form. |
| ShowIcon | Nếu nhận giá trị True thì cho phép hiển thị biểu tượng đã được thiết lập ở thuộc tính Icon, ngược lại thì nhận giá trị False. |

| | |
|---------------|--|
| StartPosition | Thiết lập vị trí xuất hiện của Form trên màn hình. Manual: xuất hiện ở góc trên bên trái màn hình, CenterScreen: xuất hiện ở giữa màn hình... |
| Text | Thiết lập dòng tiêu đề của Form. |
| Window State | Thiết lập trạng thái của Form khi chạy chương trình. Normal: hiển thị Form đúng theo kích cỡ thiết kế, Maximized: phóng to Form bằng màn hình, Minimized: thu nhỏ Form trên thanh Taskbar của hệ điều hành |

Sự kiện thông dụng của Form:

| | |
|-------------|---|
| Load | Sự kiện Load được kích hoạt khi Form được nạp vào bộ nhớ, nó thường được dùng để khởi tạo các giá trị và trạng thái cho các biến, các điều khiển... trên Form. |
| Click | Được kích hoạt khi người dùng kích chuột trên Form. |
| FormClosed | Được kích hoạt khi người dùng kích chuột vào nút Close x ở góc trên bên phải để đóng Form. |
| FormClosing | Cũng được kích hoạt khi người dùng kích chuột vào nút Close x, nhưng xảy ra trước sự kiện FormClosed tức là được phát sinh trước khi cửa sổ Form chuẩn bị đóng lại. |

- **Label - Nhãn:** được dùng để hiển thị những thông tin có tính chất cố định người sử dụng không có khả năng thay đổi ví dụ như dòng thông báo, hướng dẫn ...
- Thuộc tính thông dụng của **Label**:

| | |
|-------------|--|
| Name | Tên nhãn, thường ký hiệu bắt đầu bởi lbl |
| BackColor | Thiết lập màu nền cho nhãn, nếu thiết lập BackColor = Transparent (mục lựa chọn đầu tiên trong tab Web) thì nhãn sẽ có nền giống với nền của Form. |
| BorderStyle | Thiết lập kiểu đường viền cho nhãn. |
| Font | Thiết lập kiểu chữ và cỡ chữ cho nhãn. |
| ForeColor | Thiết lập màu chữ cho nhãn. |
| Image | Thiết lập ảnh hiển thị trên nhãn. |
| Locked | Locked = True: khóa không cho phép dịch chuyển vị trí của nhãn trên Form, ngược lại thì nhận giá trị False. |
| Text | Tiêu đề của nhãn. |
| TextAlign | Thiết lập chế độ căn chỉnh: trái, phải hoặc giữa của tiêu đề nhãn. |
| Visible | Hiện hoặc ẩn nhãn. |

- **TextBox – Hộp văn bản:** dùng để nhập dữ liệu đầu vào từ phía người sử dụng và hiển thị các kết quả đã tính toán được.
- Thuộc tính thông dụng của **TextBox**:

| | |
|-----------|--|
| Name | Tên hộp văn bản, thường ký hiệu bắt đầu bởi txt |
| BackColor | Thiết lập màu nền cho TextBox. |
| Enabled | Enabled=False: không cho phép người dùng truy cập vào TextBox (Hộp Textbox bị mờ đi), ngược lại thì bằng True. |
| Font | Thiết lập kiểu chữ và cỡ chữ cho hộp văn bản. |
| ForeColor | Thiết lập màu chữ cho hộp văn bản. |
| Locked | Locked = True: khóa không cho phép dịch chuyển vị trí của hộp văn bản trên Form, ngược lại thì nhận giá trị False |
| MaxLength | Quy định chiều dài tối đa được chấp nhận của hộp văn bản, giá trị mặc định là 32767 hoặc 0, tức là có thể chứa 32767 ký tự. Mọi xác lập khác 0, ví dụ 5 thì chỉ cho phép người dùng nhập tối đa 5 ký tự vào hộp văn bản. |

- Thuộc tính thông dụng của **TextBox**:

| | |
|--------------|---|
| Multiline | Multiline = False: chỉ cho phép hiển thị văn bản trên một dòng, và khi thiết kế ta chỉ thay đổi được độ dài của hộp văn bản. Multiline = True: cho phép văn bản được hiển thị trên nhiều dòng, và có thể thay đổi cả độ dài lẫn độ rộng của hộp văn bản khi thiết kế. |
| PasswordChar | Thuộc tính này cho phép người sử dụng bảo mật được thông tin nhập vào Textbox. Ví dụ đặt thuộc tính này bằng ký tự “*” khi đó toàn bộ dữ liệu nhập vào sẽ được hiển thị dưới dạng dấu hoa thị. Thuộc tính này chỉ được hỗ trợ khi thuộc tính Multiline = False. |
| ReadOnly | ReadOnly = True: hộp văn bản vẫn được truy cập nhưng người dùng không thể thay đổi được nội dung bên trong. |
| ScrollBars | Thiết lập thanh cuộn ngang và dọc cho hộp văn bản, có hiệu lực khi thuộc tính Multiline = True. Thanh cuộn ngang chỉ có hiệu lực khi thuộc tính WordWrap = False. |
| TabIndex | Thứ tự truy cập của hộp văn bản khi người dùng bấm phím Tab, thứ tự đầu tiên là 0. |
| Text | Chứa nội dung của hộp văn bản. |
| TextAlign | Thiết lập chế độ căn chỉnh: trái, phải hoặc giữa của dữ liệu trong hộp TextBox. |
| Visible | Visible = True: hiển thị hộp văn bản, Visible = False: ẩn hộp văn bản |
| WordWrap | Chỉ có hiệu lực khi thuộc tính Multiline = True. WordWrap = True: dòng văn bản được tự động cuộn xuống dòng khi gặp lề bên phải của hộp TextBox, ngược lại thì nhận giá trị False. |

- Sự kiện thông dụng của **TextBox**:

| | |
|-------------|---|
| TextChanged | Được kích hoạt khi người dùng thực hiện sự thay đổi bất kỳ trong hộp văn bản như: thêm, xoá, sửa, dán văn bản |
| Click | Được kích hoạt khi người dùng kích chuột vào hộp văn bản. |
| DoubleClick | Được kích hoạt khi người dùng kích đúp chuột vào hộp văn bản. |
| Enter | Được kích hoạt khi người dùng chuyển tiêu điểm tới hộp văn bản. |
| KeyPress | Trả về ký tự (trừ các ký tự đặc biệt như phím Delete, Home, Ctrl, F1...) mà người sử dụng gõ vào hộp văn bản thông qua thuộc tính KeyChar |
| KeyDown | Trả về mã Ascii của tất cả các ký tự mà người sử dụng gõ vào hộp văn bản thông qua thuộc tính KeyValue. |
| Leave | Được kích hoạt khi hộp văn bản mất tiêu điểm. |
| MouseMove | Được kích hoạt khi người dùng di chuyển chuột qua hộp văn bản. |
| MouseLeave | Được kích hoạt khi người dùng dời chuột ra khỏi hộp văn bản. |

- **Button – Nút nhấn, Nút lệnh:** cho phép người dùng thực hiện một hành động nào đó.

- Thuộc tính thông dụng của **Button**:

| | |
|-----------------|--|
| Name | Tên nút nhấn, thường ký hiệu bắt đầu bởi btn |
| BackColor | Thiết lập màu nền cho Button. |
| BackgroundImage | Thiết lập ảnh nền cho Button. |
| Enabled | Enabled=False: người dùng không thể tác động lên nút lệnh, ngược lại thì bằng True. |
| Font | Thiết lập kiểu chữ và cỡ chữ cho nút lệnh. |
| ForeColor | Thiết lập màu chữ cho nút lệnh. |
| Image | Thiết lập ảnh hiển thị trên nút lệnh. |
| Locked | Locked = True: khóa không cho phép dịch chuyển vị trí của nút lệnh trên Form, ngược lại thì nhận giá trị False |
| TabIndex | Thứ tự truy cập của nút lệnh khi người dùng bấm phím Tab. |
| Text | Tiêu đề của nút lệnh. Ta có thể quy định phím nóng cho nút lệnh bằng cách đặt dấu "&" trước một ký tự của Text, ví dụ &Quit sẽ được hiển thị là Quit, khi người sử dụng bấm Alt+Q chương trình sẽ kích hoạt nút lệnh Quit. |
| Visible | Visible = True: hiển thị nút lệnh, Visible = False: ẩn nút lệnh. |

-
- Sự kiện thông dụng của **Button**:

| | |
|------------|--|
| Click | Được kích hoạt khi người dùng kích chuột vào Button . |
| Enter | Được kích hoạt khi người dùng chuyển tiêu điểm tới Button . |
| Leave | Được kích hoạt khi Button mất tiêu điểm. |
| MouseDown | Được kích hoạt khi người dùng đặt chuột vào Button . |
| MouseUp | Được kích hoạt khi người dùng đưa chuột ra khỏi Button . |
| MouseMove | Được kích hoạt khi người dùng di chuyển chuột trên Button . |
| MouseLeave | Được kích hoạt khi người dùng dời chuột ra khỏi Button . |

- **Combobox:** cho phép lưu trữ và lựa chọn một mục dữ liệu trong một hộp danh sách thả xuống.

- Thuộc tính thông dụng của **Combobox**:

| | |
|---------------|---|
| Name | Tên hộp ComboBox, thường bắt đầu bởi cbo |
| BackColor | Thiết lập màu nền cho hộp Combo. |
| DataSource | Thiết lập nguồn dữ liệu cho Combo. |
| DropDownStyle | <p>DropDown gồm một hộp văn bản cho phép người sử dụng có thể nhập dữ liệu, kế bên có một mũi tên , nhấn vào đó sẽ xổ ra một danh sách các mục dữ liệu cho phép người dùng chọn lựa. Simple luôn hiển thị sẵn danh sách các mục dữ liệu bên dưới hộp văn bản và cho phép người sử dụng có thể nhập dữ liệu vào hộp văn bản.</p> <p>DropDownList tương tự như DropDown nhưng người sử dụng chỉ có thể chọn các phần tử từ danh sách, khi gõ một ký tự vào hộp văn bản thì danh sách sẽ cuộn đến các phần tử được bắt đầu bởi ký tự đó.</p> |
| Enabled | Nếu Enabled = False hộp Combo sẽ không hoạt động. |
| Font | Xác lập kiểu chữ và cỡ chữ cho hộp Combo. |
| ForeColor | Xác lập màu chữ cho hộp Combo. |
| Items | Khởi tạo giá trị các phần tử của hộp Combo trong thời gian thiết kế. |

| | |
|---------------------------|---|
| Items.Count | Trả về tổng số phần tử của hộp Combo trong thời gian thi hành. |
| Items[n] | Trả về nội dung phần tử thứ n của hộp Combo trong thời gian thi hành |
| SelectedItem Hoặc Text | Trả về nội dung của phần tử hiện hành đang được chọn. |
| SelectedIndex | Trả về số thứ tự của phần tử đang được chọn trong hộp Combo, phần tử đầu tiên có SelectedIndex = 0, nếu không có phần tử nào được chọn thì SelectedIndex = -1 |
| Locked | True: không cho phép dịch chuyển vị trí của hộp Combo trên Form |
| Sorted | True: các phần tử trong danh sách được sắp xếp theo thứ tự ABC. |
| TabIndex | Thứ tự truy cập khi người dùng bấm phím Tab. |
| Visible | True: hiển thị hộp Combo, False: ẩn hộp Combo. |

- Sự kiện thông dụng của **Combobox**:

| | |
|------------------------|---|
| Click | Được kích hoạt khi người dùng kích chuột vào hộp Combo. |
| DoubleClick | Được kích hoạt khi người dùng kích đúp chuột vào hộp Combo. |
| Enter | Được kích hoạt khi người dùng chuyển tiêu điểm tới hộp Combo. |
| Leave | Được kích hoạt khi hộp Combo mất tiêu điểm. |
| SelectedIndex_ Changed | Được kích hoạt khi người dùng thay đổi trạng thái lựa chọn các dòng dữ liệu trong hộp văn bản. |
| TextChanged | Được kích hoạt khi người dùng nhập, sửa, xóa dữ liệu tại vùng văn bản của hộp Combo hoặc khi ta thay đổi thuộc tính Text của hộp Combo từ mã lệnh. |
| DropDown | Chỉ xảy ra đối với hộp Combo DropDown và DropDownList, sự kiện này được gọi ngay sau khi người dùng nhấp mũi tên để thả hộp danh sách xuống (phím tắt Alt+). Vì thế sự kiện này chủ yếu được sử dụng để nhập dữ liệu cho các phần tử của hộp Combo. |

- Phương thức thông dụng của **Combobox**:

| | |
|---------------------------------|--|
| ComboName.Items.Add(Item) | dùng để bổ sung một phần tử Item cho hộp Combo trong thời gian thi hành và thường được viết trong thủ tục Form_Load. Ví dụ: thêm phần tử ABC vào hộp Combo cboabc: <i>cboabc.Items.Add("ABC");</i> |
| ComboName.Items.Remove(Item) | dùng để loại bỏ một phần tử Item của danh sách theo nội dung trong thời gian thi hành. Ví dụ: xóa phần tử ABC trong hộp Combo cboabc: <i>cboabc.Items.Remove("ABC");</i> |
| ComboName.Items.RemoveAt(Index) | dùng để loại bỏ một phần tử của hộp Combo theo chỉ số Index trong thời gian thi hành. Ví dụ: xóa phần tử thứ 2 trong hộp Combo cboabc: <i>cboabc.Items.RemoveAt(2);</i> |
| ComboName.Items.Clear() | dùng để loại bỏ tất cả các phần tử của hộp Combo trong thời gian thi hành. Ví dụ: xóa tất cả các phần tử trong hộp Combo cboabc: <i>cboabc.Items.Clear();</i> |

- **Listbox – Hộp danh sách:** là một tập hợp các chuỗi ký tự được trình bày dưới dạng liệt kê thành từng dòng trong một khung hình chữ nhật. Ta có thể chọn, bổ sung hoặc xóa một giá trị trong hộp danh sách.
-

- Thuộc tính thông dụng của **Listbox**:

| | |
|-------------|--|
| Name | Tên hộp ListBox, thường bắt đầu bởi lst |
| BackColor | Thiết lập màu nền cho hộp danh sách. |
| DataSource | Thiết lập nguồn dữ liệu cho ListBox |
| Enabled | Nếu Enabled = False hộp danh sách sẽ không hoạt động. |
| Font | Xác lập kiểu chữ và cỡ chữ cho hộp danh sách. |
| ForeColor | Xác lập màu chữ cho hộp danh sách |
| MultiColumn | MultiColumn = True: cho phép hiển thị dữ liệu theo nhiều cột. MultiColumn = False: chỉ cho phép hiển thị dữ liệu theo 1 cột. |
| ColumnWidth | Thiết lập độ rộng cho mỗi cột trong ListBox. |

| | |
|------------------------|--|
| Items | Khởi tạo giá trị cho các phần tử của hộp danh sách trong thời gian thiết kế. Khi chọn thuộc tính Items trong cửa sổ Properties, C# mở ra một hộp soạn thảo cho phép người lập trình gõ vào giá trị các phần tử. Mỗi phần tử được đặt trên một dòng riêng biệt, để xuống dòng nhấn Enter. |
| Items.Count | Trả về tổng số phần tử của danh sách trong thời gian thi hành. |
| Items(n) | Trả về nội dung phần tử thứ n của danh sách trong thời gian thi hành. |
| SelectedItem hoặc Text | Tương tự như thuộc tính Items(n), nhưng chỉ có thể trả về nội dung của phần tử hiện hành đang được chọn. |
| Locked | Locked = True: khóa không cho phép dịch chuyển vị trí của hộp danh sách trên Form, ngược lại thì nhận giá trị False |
| SelectedIndex | Trả về số thứ tự của phần tử đang được chọn trong danh sách, phần tử đầu tiên có SelectedIndex = 0, nếu không có phần tử nào được chọn thì SelectedIndex = -1 |

| | |
|---------------|--|
| SelectionMode | Quy định chế độ lựa chọn các phần tử trong hộp danh sách khi thực thi chương trình. SelectionMode có 4 giá trị: None - không cho phép lựa chọn các phần tử, One - cho phép chọn một phần tử, MultiSimple - cho phép lựa chọn nhiều phần tử riêng biệt, MultiExtended - cho phép chọn một khối các phần tử liên nhau. |
| SelectedItems | Trả về tập các phần tử đang được chọn. |
| Sorted | Nếu Sorted = True thì các phần tử trong danh sách được sắp xếp theo |
| TabIndex | Thứ tự truy cập khi người dùng bấm phím Tab. |
| Visible | True: hiển thị hộp danh sách, False: ẩn hộp danh sách. |

- Sự kiện thông dụng của **Listbox**:

| | |
|------------------------|--|
| Click | Được kích hoạt khi người dùng kích chuột vào hộp danh sách. |
| DoubleClick | Được kích hoạt khi người dùng kích đúp chuột vào hộp danh sách. |
| Enter | Được kích hoạt khi người dùng chuyển tiêu điểm tới hộp danh sách. |
| Leave | Được kích hoạt khi hộp danh sách mất tiêu điểm. |
| SelectedIndex_ Changed | Được kích hoạt khi người dùng thay đổi trạng thái lựa chọn các dòng dữ liệu trong hộp danh sách. |

- Phương thức thông dụng của **Listbox**:

| | |
|--------------------------------|--|
| ListName.Items.Add(Item) | dùng để bổ sung một phần tử Item cho hộp danh sách trong thời gian thi hành và thường được viết trong thủ tục Form_Load. Ví dụ: thêm phần tử ABC vào hộp danh sách lstabc: <i>lstabc.Items.Add("ABC");</i> |
| ListName.Items.Remove(Item) | dùng để loại bỏ một phần tử Item của danh sách theo nội dung trong thời gian thi hành. Ví dụ: xóa phần tử ABC trong hộp danh sách lstabc: <i>lstabc.Items.Remove("ABC");</i> |
| ListName.Items.RemoveAt(Index) | dùng để loại bỏ một phần tử của hộp danh sách theo chỉ số Index trong thời gian thi hành. Ví dụ: xóa phần tử thứ 2 trong hộp Combo lstabc: <i>lstabc.Items.RemoveAt(2);</i> |
| ListName.Items.Clear() | dùng để loại bỏ tất cả các phần tử của hộp danh sách trong thời gian thi hành. Ví dụ: xóa tất cả các phần tử trong hộp danh sách lstabc: <i>lstabc.Items.Clear();</i> |

- **Checkbox - Hộp đánh dấu:** cho phép đồng thời không chọn, chọn một, hoặc chọn nhiều khả năng trong một nhóm các lựa chọn.
- Thuộc tính thông dụng của **Checkbox**:

| | |
|-----------------|--|
| Name | Tên hộp CheckBox, thường bắt đầu bằng chk |
| BackColor | Thiết lập màu nền cho hộp CheckBox. |
| BackgroundImage | Thiết lập ảnh nền cho hộp CheckBox. |
| Checked | Trả về giá trị của hộp CheckBox ứng với trạng thái của nó khi tương tác với người sử dụng. Checked = True: hộp CheckBox đang được chọn, Checked = False: hộp CheckBox không được chọn. |
| CheckState | Thiết lập trạng thái cho hộp CheckBox, CheckState = Checked: hộp CheckBox được chọn, CheckState = Unchecked: hộp CheckBox không được chọn. |
| Enabled | Nếu Enabled = False hộp CheckBox sẽ không hoạt động. |
| Font | Xác lập kiểu chữ và cỡ chữ của nội dung hộp CheckBox. |

| | |
|-----------|--|
| ForeColor | Xác lập màu chữ của nội dung hộp CheckBox. |
| Image | Thiết lập ảnh hiển thị trên hộp CheckBox. |
| Locked | Locked = True: khóa không cho phép dịch chuyển vị trí của hộp |
| TabIndex | Thứ tự truy cập khi người dùng bấm phím Tab |
| Text | Thiết lập nội dung của hộp CheckBox. |
| Visible | Visible = True: hiển thị hộp CheckBox, Visible = False: ẩn hộp |
| | |

-
- Sự kiện thông dụng của **Checkbox**:

| | |
|----------------|--|
| Click | Được kích hoạt khi người dùng kích chuột vào hộp CheckBox. |
| Enter | Được kích hoạt khi người dùng chuyển tiêu điểm tới hộp CheckBox. |
| Leave | Được kích hoạt khi hộp CheckBox mất tiêu điểm |
| CheckedChanged | Được kích hoạt khi hộp CheckBox thay đổi trạng thái. |

- **Radiobutton – Nút tùy chọn:** chỉ cho phép người dùng chọn một khả năng trong một nhóm các lựa chọn
- Thuộc tính thông dụng của **Radiobutton:**

| | |
|-----------------|--|
| Name | Tên nút tùy chọn, thường bắt đầu bằng rdo |
| BackColor | Thiết lập màu nền cho nút tùy chọn. |
| BackgroundImage | Thiết lập ảnh nền cho nút tùy chọn. |
| Checked | Trả về giá trị của nút tùy chọn khi tương tác với người sử dụng. Checked = True: nút tùy chọn đang được chọn, Checked = False: nút tùy chọn không được chọn. |
| Enabled | Nếu Enabled = False nút tùy chọn sẽ không hoạt động. |
| Font | Xác lập kiểu chữ và cỡ chữ của nội dung nút tùy chọn. |
| ForeColor | Xác lập màu chữ của nội dung nút tùy chọn. |

| | |
|----------|---|
| Image | Thiết lập ảnh hiển thị trên nút tùy chọn. |
| Locked | Locked = True: khóa không cho phép dịch chuyển vị trí của nút tùy chọn trên Form, ngược lại thì nhận giá trị False. |
| TabIndex | Thứ tự truy cập khi người dùng bấm phím Tab. |
| Text | Thiết lập nội dung của nút tùy chọn. |
| Visible | True: hiển thị nút tùy chọn, False: ẩn nút tùy chọn. |

-
- Sự kiện thông dụng của **Radiobutton**:

| | |
|----------------|--|
| Click | Được kích hoạt khi người dùng kích chuột vào nút tùy chọn. |
| Enter | Được kích hoạt khi người dùng chuyển tiêu điểm tới nút tùy chọn. |
| Leave | Được kích hoạt khi nút tùy chọn mất tiêu điểm. |
| CheckedChanged | Được kích hoạt khi nút tùy chọn thay đổi trạng thái. |

VÍ DỤ ÁP DỤNG

Ví dụ 1: Ứng dụng Windows Forms để thiết kế giao diện HÌNH CHỮ NHẬT, với yêu cầu:

- Nhập **Chiều dài** và **Chiều rộng** từ bàn phím..
- Click **Tính diện tích** thì kết quả hiển thị trong các ô chứa diện tích.
- Click **Tính chu vi** thì kết quả hiển thị trong các ô chứa chu vi.

The screenshot shows a Windows Forms application window titled "HÌNH CHỮ NHẬT". Inside the window, there are two labels: "Chiều dài" (Length) and "Chiều rộng" (Width), each followed by a text input box. Below these, there are two buttons: "Tính diện tích" (Calculate Area) and "Tính chu vi" (Calculate Perimeter). At the bottom of the window, there are two more empty text input boxes, presumably for displaying the calculated area and perimeter.

VÍ DỤ ÁP DỤNG

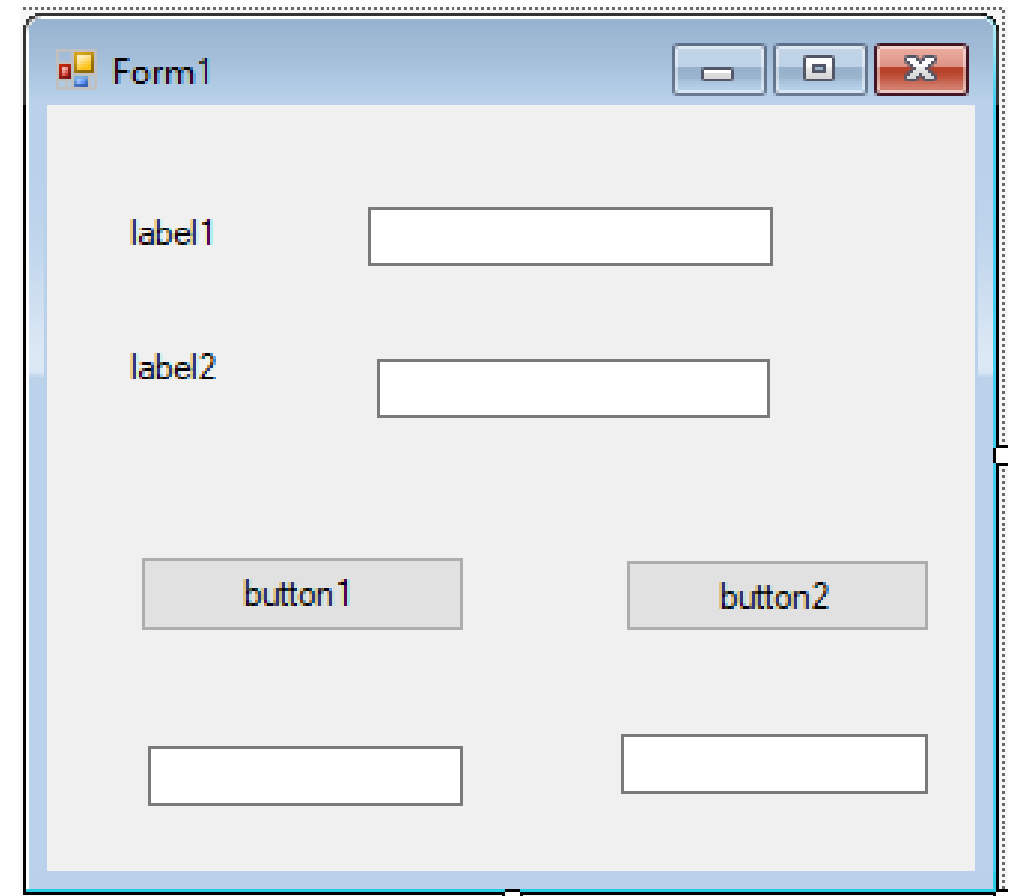
Các bước thực hiện:

- Khởi động phần mềm Microsoft Visual Studio
- Chọn File > New Project > Windows Form Application
- Tại ô Name đặt tên bài và vị trí lưu (Browse)
- Bấm OK.
- Xây dựng giao diện.

VÍ DỤ ÁP DỤNG

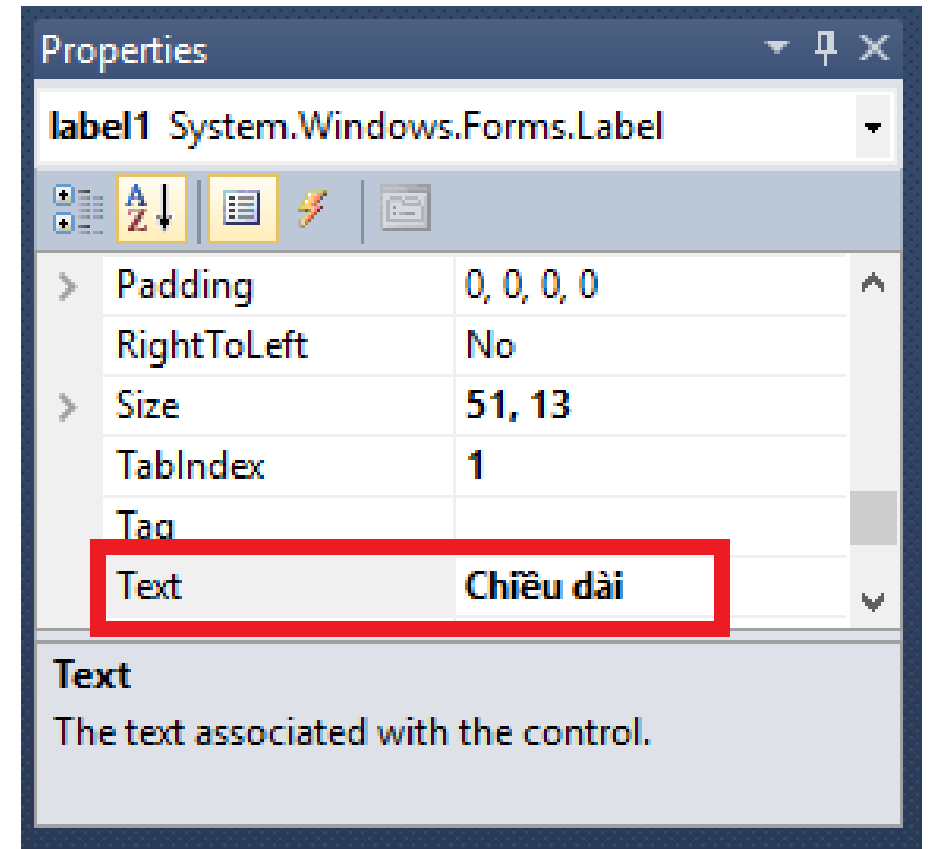
Các bước xây dựng giao diện:

Bước 1: Tạo giao diện cho chương trình. Chọn View > Toolbox > Common Control. Sau đó nhấp chọn Label, Textbox, Button kéo thả vào Form.



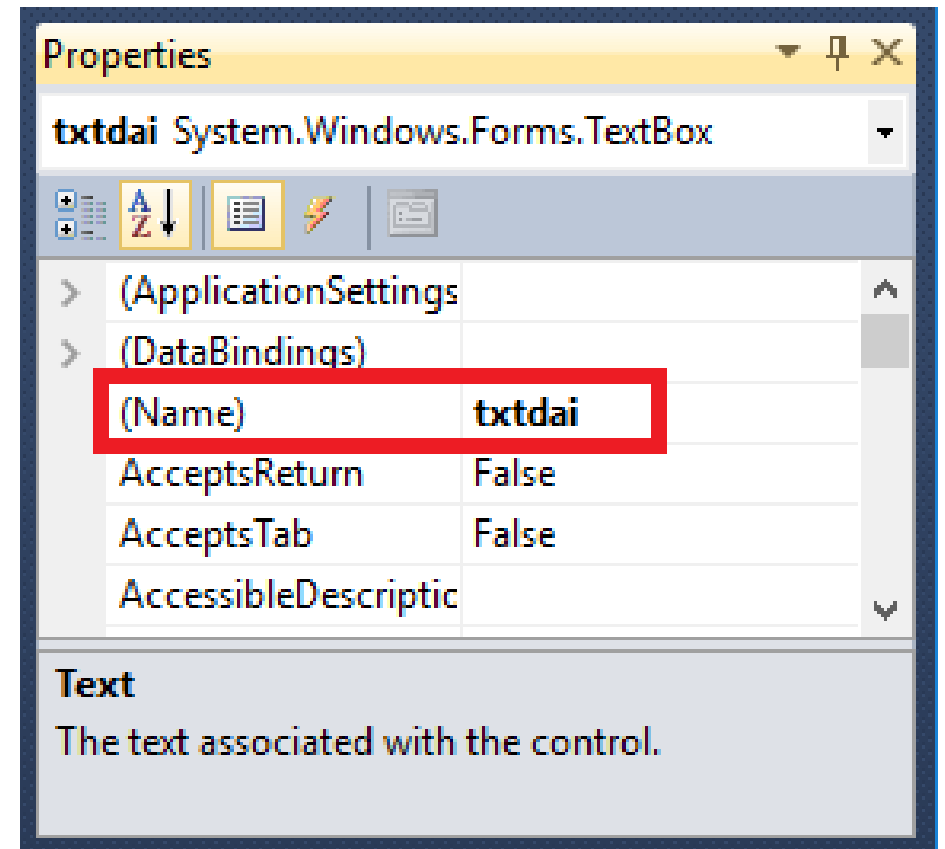
VÍ DỤ ÁP DỤNG

Bước 2: Muốn đổi tên Label, Button thì đánh dấu bằng cách bấm chuột trái vào Label, Button sau đó chọn Properties > Text



VÍ DỤ ÁP DỤNG

Bước 3: Đặt tên biến cho Textbox thì đánh dấu bằng cách bấm chuột trái vào Textbox sau đó chọn Properties > Name. Thao tác lần lượt cho các Textbox txt dai, txt trong, txt dt, txt cv.



VÍ DỤ ÁP DỤNG

Bước 4: Xử lý sự kiện Tính diện tích, nhấp đúp Button **Tính diện tích**, giao diện lập trình Form.cs*

```
private void button1_Click(object sender, EventArgs e)
{
    float a, b;
    double dt;
    a = float.Parse(txttai.Text);
    b = float.Parse(txtrong.Text);
    dt = a * b;
    txttdt.Text = dt.ToString();
}
```

Hàm tính diện tích

VÍ DỤ ÁP DỤNG

Bước 5: Xử lý sự kiện Tính chu vi, nhấp đúp Button **Tính chu vi**, giao diện lập trình Form.cs*

```
private void button2_Click(object sender, EventArgs e)
{
    float a, b;
    double cv;
    a = float.Parse(txttai.Text);
    b = float.Parse(txtrong.Text);
    cv = 2 * (a + b);
    txtcv.Text = cv.ToString();
}
```

Hàm tính chu vi

VÍ DỤ ÁP DỤNG

Kết quả thực hiện:

HÌNH CHỮ NHẬT

Chiều dài: 5.5

Chiều rộng: 8.5

Tính diện tích Tính chu vi

46.75 28

BÀI TẬP 3.1+3.2

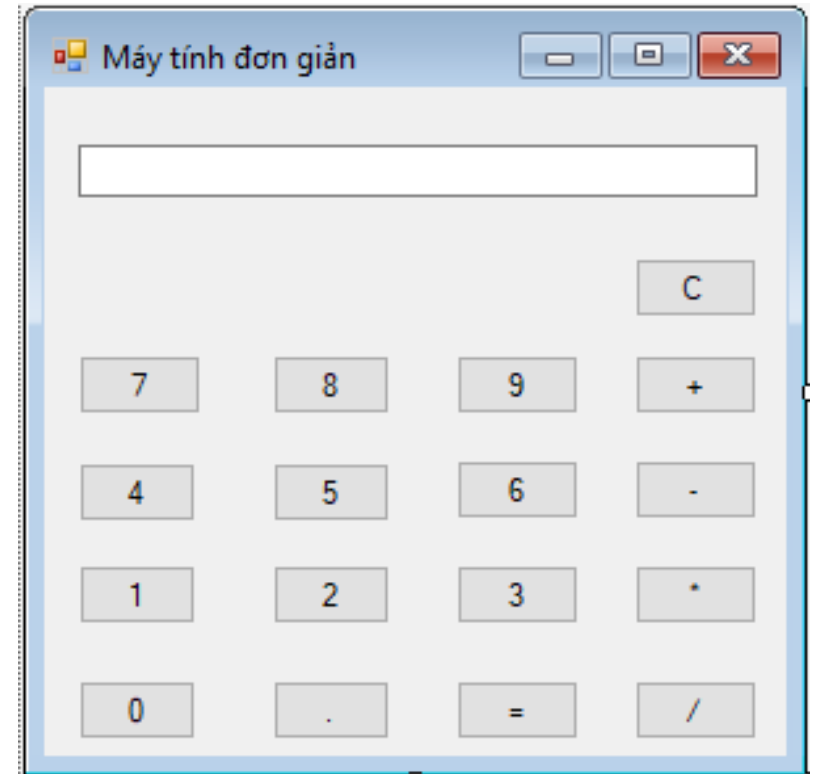
- **Bài 1:** Ứng dụng Windows Forms để thiết kế giao diện GIẢI PHƯƠNG TRÌNH như hình bên, với yêu cầu:
 - Có thể lựa chọn bài toán: **Phương trình bậc 1** hoặc **Phương trình bậc 2**.
 - Khi lựa chọn **Phương trình bậc 1** thì hệ số c không hiển thị trên giao diện.
 - Click **Giải** thì kết quả hiển thị trong ô **Kết quả**.
 - Click **Xóa** thì tất cả các ô đều trống rỗng.
 - Click **Thoát** thì thoát khỏi Form (đóng Form).

The screenshot shows a Windows Forms application window titled "Form1". The main title of the application is "GIẢI PHƯƠNG TRÌNH". Below the title, there is a section labeled "Mời bạn nhập" (Please enter). This section contains two radio buttons: "Phương trình bậc 1" (Linear equation) and "Phương trình bậc 2" (Quadratic equation). Below the radio buttons, there are three input fields labeled "Nhập a", "Nhập b", and "Nhập c". To the right of these input fields, there are three buttons: "Giải" (Solve), "Xóa" (Clear), and "Thoát" (Exit). At the bottom, there is a larger input field labeled "Kết quả" (Result).

BÀI TẬP 3.1+3.2

Bài 2: Ứng dụng Windows Forms để thiết kế giao diện nhập vào dãy số gồm 10 số nguyên. Hiện thị ra giao diện dãy các số nguyên tố và tổng các số chẵn trong dãy số nguyên ban đầu dãy số.

Bài 3: Ứng dụng Windows Forms để thiết kế giao diện máy tính đơn giản như hình bên.



BÀI TẬP 3.1+3.2

- **Bài 4**: Ứng dụng Windows Forms để thiết kế giao diện QUẢN LÝ KHÁCH SẠN

The screenshot shows a Windows Forms application window titled "Quản lý khách sạn". The main area is titled "QUẢN LÝ KHÁCH LƯU TRÚ". The form is divided into several sections:

- Họ và tên**: A text input field.
- Địa chỉ**: A text input field.
- Thời gian lưu trú**: A section containing two date pickers: **Ngày đến** (Arrival date) and **Ngày đi** (Departure date).
- Loại phòng**: A section with two radio buttons: **Phòng đơn** (Single room) and **Phòng đôi** (Double room).
- Dịch vụ sử dụng**: A section with two checkboxes: **Internet** and **Giặt là** (Laundry).
- Hóa đơn**: A large empty rectangular box for displaying the bill.
- Buttons**: There are three buttons: **TÍNH TIỀN** (Calculate bill) located in the center, **XÓA** (Delete) at the bottom left, and **THOÁT** (Exit) at the bottom right.

BÀI TẬP 3.1+3.2

- Yêu cầu:
 - Có thể nhập **Họ tên, địa chỉ, ngày đến** và **ngày đi** của khách lưu trú.
 - Có thể lựa chọn loại phòng: **Phòng đơn** hoặc **Phòng đôi**.
 - Có thể chọn các dịch vụ: **Internet, Giặt là**.
 - Click **Tính tiền** thì Hóa đơn hiện ra các thông tin: Họ tên, địa chỉ, ngày đến, ngày đi, loại phòng đã ở, các loại dịch vụ đã sử dụng và Tổng tiền cần thanh toán.
 - Click **Xóa** thì tất cả các ô đều trống rỗng.
 - Click **Thoát** thì thoát khỏi Form (đóng Form).

BÀI TẬP 3.1+3.2

- **Bài 5**: Ứng dụng Windows Forms để thiết kế giao diện QUẢN LÝ SINH VIÊN
- Yêu cầu:
 - Có thể nhập **Họ tên, mã số sinh viên, điểm, giới tính (Nam hoặc Nữ)** của sinh viên.
 - Nếu là **Cán bộ lớp** thì có thể là **Lớp trưởng, Lớp phó, Bí thư**.
 - Click **Hiển thị** thì hiện ra tất cả các thông tin của sinh viên, mỗi sinh viên hiển thị trên 1 dòng trong danh sách.
 - Click **Thêm** thì cho phép nhập thêm sinh viên mới với đầy đủ thông tin trên.
 - Click **Xóa** thì tất cả các ô đều trống rỗng.
 - Click **Thoát** thì thoát khỏi Form (đóng Form).

BÀI TẬP 3.1+3.2

- Giao diện QUẢN LÝ SINH VIÊN tham khảo

The image displays two screenshots of a software application titled "Phần mềm quản lý sinh viên" (Student Management Software). Both screenshots show the "Chương trình quản lý sinh viên" (Student Management Program) window.

Left Screenshot (Empty Form):

- Thông tin Sinh Viên (Student Information):**
 - Họ Tên (Last Name):
 - Giới (Gender):
 - Mã SV (Student ID):
 - Năm sinh (Year of Birth):
 - Ngành (Major):
 - Cán bộ lớp (Class Staff): ☐ Cán bộ
 - Điểm (Score):
 - Xếp loại (Rank):
- Buttons:** Thêm (Add), 0/0, Lùi (Back)

Right Screenshot (Filled Form):

- Thông tin Sinh Viên (Student Information):**
 - Họ Tên (Last Name):
 - Giới (Gender):
 - Mã SV (Student ID):
 - Năm sinh (Year of Birth):
 - Ngành (Major):
 - Cán bộ lớp (Class Staff): ☐ Cán bộ
 - Điểm (Score):
 - Xếp loại (Rank):
- Buttons:** Thêm (Add), 0/0, Lùi (Back)

BÀI TẬP 3.1+3.2

- **Bài 5:** Ứng dụng Windows Forms để thiết kế giao diện QUẢN LÝ SINH VIÊN

Phần mềm quản lý sinh viên

Chương trình quản lý sinh viên

Thông tin Sinh Viên

Họ Tên: Nguyễn Thùy Linh Giới: Nữ

Mã SV: 112 Năm sinh: 2000

Ngành: Tự Động Hoá Cán bộ lớp: ☒ Cán bộ

Điểm: 10

Xếp loại: Xuất Sắc

Thêm 1 Lùi

Phần mềm quản lý sinh viên

Chương trình quản lý sinh viên

Thông tin Sinh Viên

Họ Tên: Nguyễn Thùy Linh Giới: Nữ

Mã SV: 112 Năm sinh: 1998

Ngành: Tự Động Hoá Cán bộ lớp: ☒ Cán bộ

Điểm: 10

Xếp loại: Xuất Sắc

Thêm 2/2 Lùi