

CHƯƠNG IV

LẬP TRÌNH WINDOWS FORMS

1

Nội dung

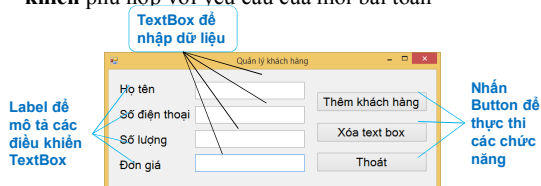
1. Giới thiệu
2. Windows Form và các điều khiển cơ bản
3. Hộp thoại
4. Menu và Toolbar
5. Ứng dụng Multiple Document Interface (MDI)

2

1. Giới thiệu

Ứng dụng có giao diện đồ họa

- GUI (Graphical User Interface): Giao diện người dùng đồ họa
- GUI được thiết kế bằng việc thêm vào **Form** các **điều khiển** phù hợp với yêu cầu của mỗi bài toán

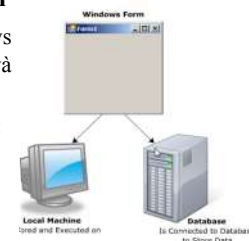


4

1. Giới thiệu

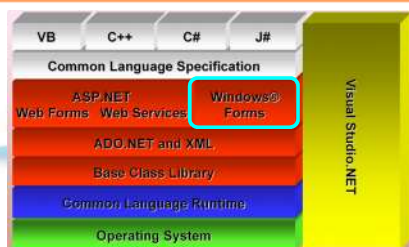
Ứng dụng Windows Form

- Các ứng dụng Windows Form là ứng dụng GUI và được thực thi trên máy local.
- Vai trò của Windows Forms:
 - Chứa các điều khiển
 - Xử lý dữ liệu user nhập
 - Hiển thị thông tin
 - Kết nối CSDL



4

1. Giới thiệu



- Windows Form cung cấp một mô hình lập trình chuẩn cho phép bạn xây dựng các ứng dụng với ít lỗi nhất

5

2. Windows Form và các điều khiển cơ bản

- 2.1. Form
- 2.2. Label
- 2.3. TextBox
- 2.4. CheckBox
- 2.5. RadioButton
- 2.6. CheckBox
- 2.7. Các điều khiển Calendar
- 2.8. List Box và ComboBox

6

2.1. Form

- Là thành phần cơ bản của ứng dụng GUI, là các cửa sổ mà người dùng nhìn thấy khi chạy chương trình
- Tạo form bằng cách tạo một đối tượng của lớp Form (trong namespace **System.Windows.Forms**)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace LoginApplication
{
    public partial class frmLogin : Form
    {
    }
}
```

Code View of Visual Studio IDE

7

2.1. Form (*frm*)

Thuộc tính

- **Name** - tên được sử dụng để xác định Form
- **Text** - chuỗi hiển thị trên thanh tiêu đề Form.
- **BackColor** - màu nền của Form.
- **Font** - tên font, kiểu và kích thước của văn bản hiển thị trên Form.
 - Mặc định, các điều khiển của Form sử dụng font được thiết lập cho Form.

8

2.1. Form

Thuộc tính . . .

- **Size** - Chiều rộng và chiều cao của Form (pixel).
- **AcceptButton**: Nút lệnh được nhấn khi user ấn phím Enter
- **CancelButton**: Nút lệnh được nhấn khi user ấn phím ESC
- **WindowState**: Xác định trạng thái hiển thị ban đầu của form

9

2.1. Form

Phương thức

- Show - Hiển thị form
- Hide - Ẩn form
- Close - Đóng form

10

2.1. Form

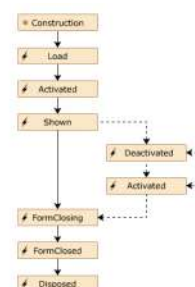
Event (sự kiện)

- Là một thông điệp được gửi bởi một đối tượng để báo hiệu sự xuất hiện của một hành động. Hành động này có thể được gây ra bởi sự tương tác của người sử dụng chẳng hạn như kích chuột hoặc nó có thể được kích hoạt bởi các logic khác của chương trình
- **Trình xử lý sự kiện (event handler)**: Là phương thức được gọi đến khi một sự kiện xảy ra

Mỗi điều khiển có một nhóm sự kiện khác nhau

2.1. Form

Vòng đời của một form



Event

- Activated
- Click
- Deactivated
- FormClosed
- FormClosing
- GotFocus
- Load
- Resize

Các điều khiển

- Điều khiển (control) là các đối tượng nằm trong Form
- **Control** class là lớp cơ sở của tất cả các điều khiển
- Mỗi điều khiển có một tập các thuộc tính, phương thức, và sự kiện
- Sử dụng điều khiển:
 - `[this].Tên_điều_khiển.Tên_thuộc_tính`
 - `[this].Tên_điều_khiển.Tên_phương_thức([danh_sách_đối_số])`

2.2. Label – nhãn (*lbl*) A Label

- **Label** dùng để hiển thị tiêu đề, kết quả hay văn bản mang tính mô tả mà người dùng không thể thay đổi.
- **Các thuộc tính**
 - **Name** - Tên sử dụng để xác định Label
 - **Text** - Văn bản được hiển thị trên Label.
 - **AutoSize** - Cho phép tự động thay đổi kích thước của Label để vừa với nội dung.

14

2.2. Label

Các thuộc tính . . .

- **BorderStyle** - Xác định hình dáng đường viền của Label.
- **Font** - Tên font, kiểu và kích thước của văn bản được hiển thị trên Label.
- **TextAlign** - Chỉ ra cách văn bản được căn chỉnh trong phạm vi Label.

15

2.3. TextBox - Hộp văn bản (*txt*) abl TextBox

- **TextBox** cho phép người dùng **nhập** dữ liệu từ bàn phím và có thể **hiển thị** dữ liệu cho người dùng
- **Các thuộc tính**
 - **Name** - Tên sử dụng để xác định TextBox.
 - **Text** - Văn bản được hiển thị trong TextBox.
 - **TextAlign** - Xác định cách văn bản được căn chỉnh trong phạm vi TextBox.

16

TextBox (Hộp văn bản) abl TextBox

Các thuộc tính . . .

- **Font**: Tên font, kiểu và kích thước của văn bản hiển thị trong TextBox
- **Maxlength**: Số ký tự cho phép nhập
- **MultiLine**: Cho phép nhập nhiều dòng
- **ReadOnly**: Chỉ cho hiển thị, không cho nhập
- **PasswordChar**: Thay thế ký tự nhập bằng ký tự này

17

TextBox (Hộp văn bản) abl TextBox

Các sự kiện

- **TextChanged**: Xảy ra khi thuộc tính text thay đổi
- **KeyPress**: Xảy ra khi text box nhận focus và người dùng nhấn rồi thả một phím

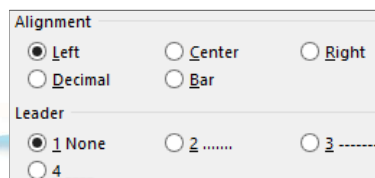
18

2.4. Button - nút lệnh (*btn*) Button

- Button ra lệnh cho ứng dụng thực hiện một chức năng khi được nhấn
- **Các thuộc tính**
 - **Name**: tên được sử dụng để xác định Button.
 - **Text**: văn bản được hiển thị trên Button.
 - **Enable**: cho phép sử dụng nút lệnh hay không
- **Sự kiện**:
 - **Click**: xảy ra người sử dụng nhấn chuột vào button

19

2.5. Radio Button - nút chọn (*rad*) RadioButton



The screenshot shows the 'RadioButton' properties window. Under the 'Alignment' section, 'Left' is selected. Under the 'Leader' section, '1 None' is selected.

- Trong một nhóm các Radio Button, người dùng chỉ có thể chọn 1 lựa chọn

20

2.5. Radio Button (nút chọn) RadioButton

Các thuộc tính:

- **Name**: tên được sử dụng để xác định nút chọn
- **Text**: văn bản được hiển thị cạnh nút chọn
- **Enable**: cho phép sử dụng nút chọn hay không
- **Checked**: chỉ ra nút chọn có được lựa chọn hay không (*true* –được chọn; *false*: không được chọn)

21

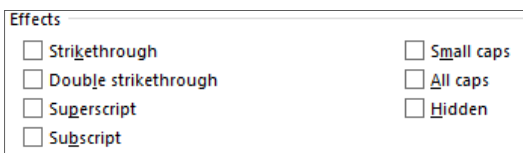
2.5. Radio Button (nút chọn) RadioButton

Các sự kiện:

- **CheckedChange**: sự kiện xảy ra mỗi khi thuộc tính checked thay đổi giá trị

22

2.6. CheckBox - hộp kiểm (*chk*) CheckBox



The screenshot shows the 'CheckBox' properties window. Under the 'Effects' section, several options are listed: Strikethrough, Double strikethrough, Superscript, Subscript, Small caps, All caps, and Hidden. None are currently checked.

- Trong một nhóm các CheckBox, người dùng có thể chọn 0, 1 . . . hay tất cả các lựa chọn

23

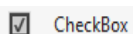
2.6. CheckBox(hộp kiểm) CheckBox

Các thuộc tính:

- **Name**: tên được sử dụng để xác định CheckBox
- **Text**: văn bản được hiển thị cạnh CheckBox
- **Enable**: cho phép sử dụng CheckBox hay không
- **Checked**: chỉ ra CheckBox có được lựa chọn hay không (*true* –được chọn; *false*: không được chọn)

24

2.6. CheckBox(hộp kiểm)



Các sự kiện:

- **CheckedChange**: sự kiện xảy ra mỗi khi thuộc tính checked thay đổi giá trị

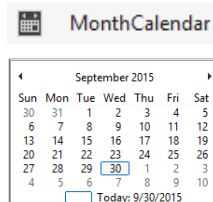
25

2.7. Các điều khiển Calendar

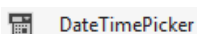
- Hiện lịch trên form, cho phép người dùng hiển thị | chọn dữ liệu ngày tháng thời gian

- Thuộc tính **SelectionStart**: trả về ngày được chọn đầu tiên (hoặc duy nhất)

- Sự kiện **DateChanged**: xảy ra khi một ngày (hoặc khoảng ngày tháng mới) được chọn



2.7. Các điều khiển Calendar

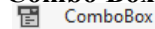


Wednesday, September 30, 2015

- Thuộc tính **value**: trả về ngày hiện tại hiển thị trong điều khiển
- Sự kiện **ValueChanged**: xảy ra khi giá trị thuộc tính value thay đổi

27

2.8. List Box (lst) và Combo Box (cbo)



- Cho phép người sử dụng lựa chọn các phần tử từ một danh sách
- Có hầu hết các thuộc tính và phương thức giống nhau (trừ DropDownStyle của Combo Box và SelectionMode của List Box)
 - **SelectionMode**: **One** – Chọn một; **MultiSimple**: để chọn nhiều → nhấn vào phần tử muốn chọn; **MultiExtended**: dùng phím Ctrl | Shift và nhấn chuột để chọn nhiều phần tử

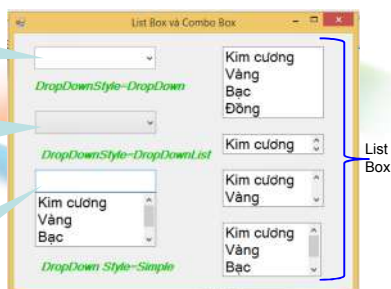
28

List Box và Combo Box

Cho phép nhập dữ liệu vào textbox

Không cho phép nhập dữ liệu vào textbox

Cho phép nhập dữ liệu vào textbox



29

2.8. List Box và Combo Box

Items Collection:

- Là danh sách các phần tử hiển thị trong điều khiển

Các phương thức

- **Tên_đối_tượng.Items.Add(giá_trị_phần_tử)**
Thêm phần tử vào danh sách

- **Tên_đối_tượng.Items.Insert(chỉ_số, giá_trị_phần_tử)**

Thêm phần tử vào vị trí xác định trong danh sách

30

2.8. List Box và Combo Box

Items Collection - Các phương thức

- **Tên_đối_tượng.Items.Remove(giá_trị_phần_tử)**
Xóa phần tử trong danh sách
- **Tên_đối_tượng.Items.RemoveAt(chỉ_số)**
Xóa phần tử ở vị trí xác định trong danh sách
- **Tên_đối_tượng.Items.Clear()**
Xóa toàn bộ các phần tử

31

2.8. List Box và Combo Box

Items Collection - Các thuộc tính

- **Tên_đối_tượng.Items.Count**
Xác định số phần tử trong danh sách
- **Tên_đối_tượng.Items[chỉ_số]**
Tham chiếu đến phần tử ở vị trí xác định
- **Tên_đối_tượng.SelectedIndex**
Xác định chỉ số của phần tử được chọn. Nếu không chọn phần tử nào SelectedIndex = -1

32

2.8. List Box và Combo Box

Các sự kiện

- **SelectedIndexChanged:** Xảy ra khi giá trị thuộc tính SelectedIndex thay đổi

33

Điều khiển ListView

- Dùng để hiển thị dữ liệu theo các dòng và các cột
- Thuộc tính:
 - Items: danh sách các dòng trong list view
 - Columns: các cột hiển thị trong chế độ Details
 - View: Kiểu hiển thị của ListView

34

3. Hộp thoại

3.1. MessageBox

3.2. Các hộp thoại thường dùng

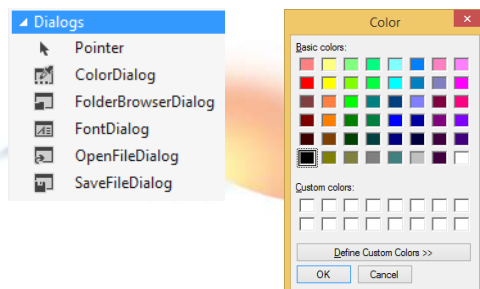
35

3.1. MessageBox (hộp thông điệp)

- Được sử dụng để hiển thị một thông điệp cho người sử dụng
- Để hiển thị thông điệp sử dụng phương thức **Show()**. Phương thức trả lại kiểu đối tượng DialogResult
- Cú pháp
MessageBox.Show(chuỗi_thông_báo , chuỗi_tiêu_đề , nút_lệnh , biểu_tượng)

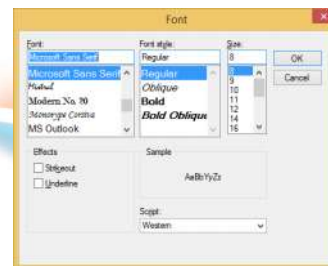
36

3.2. Các hộp thoại thường dùng



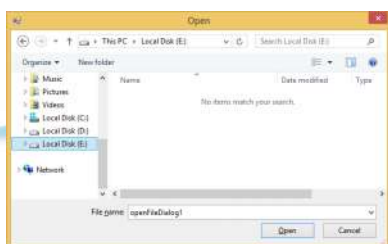
37

3.2. Các hộp thoại thường dùng



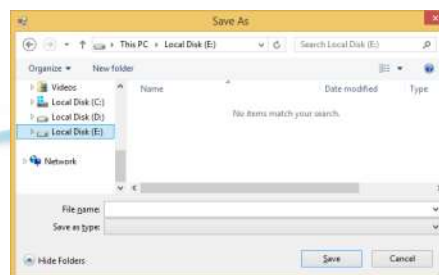
38

3.2. Các hộp thoại thường dùng



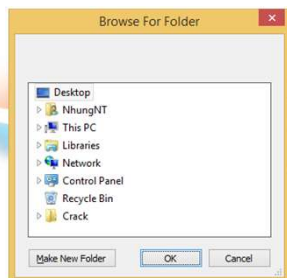
39

3.2. Các hộp thoại thường dùng



40

3.2. Các hộp thoại thường dùng



41

3.2. Các hộp thoại thường dùng

Các bước sử dụng hộp thoại

1. Thêm điều khiển lên form
2. Hiển thị hộp thoại sử dụng phương thức ShowDialog()
3. Sử dụng thông tin từ hộp thoại

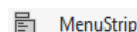
42

4. Menu và Toolbar

- 4.1. Menu
- 4.2. Context Menu
- 4.3. Toolbar

43

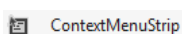
4.1. Menu



- Menu dùng để nhóm các chức năng liên quan
- **Các bước tạo menu**
 1. Thêm điều khiển MenuStrip lên form
 2. Sử dụng Menu Designer để thiết kế menu
 3. Viết code để xử lý sự kiện người dùng nhấn vào các menu item

44

4.2. Context Menu



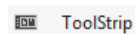
- Là menu hiển thị khi nhấn chuột phải
- **Các bước tạo Context Menu**

Từ 1 – 3 : sử dụng điều khiển ContextMenuStrip thiết kế menu ngữ cảnh tương tự menu chính

 4. Thiết lập thuộc tính ContextMenuStrip của đối tượng là context menu muốn xuất hiện khi nhấn chuột phải

45

4.3. Toolbar



- Toolbar cho phép truy cập nhanh đến các chức năng của ứng dụng
- **Các bước tạo Toolbar**
 1. Thêm điều khiển ToolStrip lên form
 2. Tạo các button trên Toolbar
 3. Viết code xử lý sự kiện người dùng nhấn vào các button

46

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

47

5. Ứng dụng MDI

- **Tạo parent form:** tạo form → thiết lập thuộc tính IsMdiContainer = true
- **Tạo form con:** trong code, khai báo và khởi tạo đối tượng form con → thiết lập thuộc tính MdiParent là form hiện hành (form cha) → hiển thị form con sử dụng phương thức Show

48

6. Kiểm tra hợp lệ dữ liệu

- Sử dụng **try catch**; **if** và **message box**
- Thuộc tính **Maxlength**: Số ký tự cho phép nhập
- Thuộc tính **CharacterCasing**: chuyển tất cả các ký tự sang hoa | thường

49

6. Kiểm tra hợp lệ dữ liệu

ErrorProvider:

- Cho phép hiển thị trực tiếp các thông báo lỗi trên form
- Phương thức **SetError**: Thiết lập chuỗi thông báo lỗi cho điều khiển

```
Đối_tượng_ErrorProvider.SetError(
    tên_điều_khiển, thông_báo_lỗi)
```

50

6. Kiểm tra hợp lệ dữ liệu

Kiểm tra hợp lệ dữ liệu mức field

- Khi người sử dụng nhập xong dữ liệu và chuyển focus đi thì sự kiện **Validating** xảy ra
- Viết code cho sự kiện **Validating**. Nếu dữ liệu không thỏa mãn hãy thiết lập thuộc tính **Cancel** của tham số **e** là **true**

51