

PHÁT TRIỂN ỨNG DỤNG CSDL 1

Tháng 9/2014

# LẬP TRÌNH ỨNG DỤNG VỚI NGÔN NGỮ C#

## Bài tập lập trình giao diện với C#

Tóm tắt nội dung bài thực hành:

Hướng dẫn về môi trường lập trình, cấu trúc chương trình, sử dụng các kiểu dữ liệu cơ bản, các cấu trúc điều khiển của ngôn ngữ lập trình c#.

Bộ môn **Hệ thống thông tin**

Khoa Công nghệ thông tin

ĐH Khoa học tự nhiên TP HCM



## MỤC LỤC

<b>1</b>	<b>Đăng nhập.....</b>	<b>1</b>
1.1	Thiết kế giao diện đăng nhập .....	1
1.2	Thiết kế xử lý giao diện .....	1
<b>2</b>	<b>Quản lý thu chi .....</b>	<b>2</b>
2.1	Thiết kế giao diện quản lý thu chi.....	2
2.2	Thiết kế xử lý giao diện .....	3
<b>3</b>	<b>Kết nối giao diện Đăng nhập và Quản lý thu chi .....</b>	<b>4</b>

## 1 Yêu cầu chung

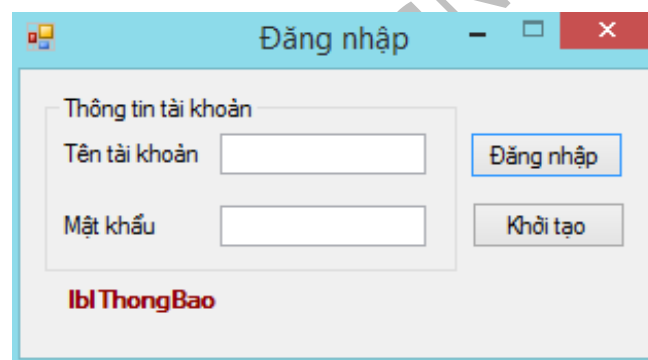
- Sinh viên thực hiện 3 bài tập bên dưới trong cùng một project.
- Tất cả các bài tập bên dưới đều không được sử dụng designer để thực hiện giao diện mà phải tự thực hiện bằng dòng lệnh.
- Các thành phần giao diện được khởi tạo trong hàm MyInit() đặt trong tập tin giao diện .cs (không đặt trong tập tin designer.cs).

Ví dụ: với form có tên là frmDangNhap, sinh viên cần đặt hàm MyInit trong tập tin frmDangNhap.cs

## 2 Đăng nhập

### 2.1 Thiết kế giao diện đăng nhập

- Thiết kế giao diện như hình sau:



### 2.2 Thiết kế xử lý giao diện

- txtMatKhaui: sử dụng thuộc tính PasswordChar để textbox luôn hiển thị dấu ‘-’ khi người dùng nhập mật khẩu.
- lblThongBao:
  - Sử dụng thuộc tính Font và ForeColor để in đậm và tô màu đỏ dòng thông báo.
  - Mặc định hoặc khi không có lỗi, người dùng sẽ không thấy label này.
- btnKhoiTao: đưa các textbox, label về lại trạng thái ban đầu khi giao diện được mở lên.
- btnDangNhap:

- Kiểm tra tên tài khoản có phải là MSSV hợp lệ không, nếu không thì báo lỗi “Tên đăng nhập phải là MSSV hợp lệ”. MSSV hợp lệ là MSSV của sinh viên đang thực hiện bài hoặc tên viết tắt của giáo viên thực hành (ví dụ: giáo viên Hoàng Anh Tú sẽ có tên viết tắt là hatu).
- Kiểm tra mật khẩu có dài hơn 8 ký tự không, nếu không thì báo lỗi “Mật khẩu phải dài hơn 8 ký tự”.
- Nếu không có lỗi xảy ra, đổi lblThongBao sang màu xanh và hiện thông báo “Đăng nhập thành công”.

### 3 Quản lý thu chi

#### 3.1 Thiết kế giao diện quản lý thu chi

- Thiết kế giao diện thu chi như hình sau:

**Quản lý thu chi**

**Danh mục**

Loại: ☒ Thu ☐ Chi

Tài khoản: ☒ Ví ☐ Thẻ ngân hàng

Danh mục:

**Thông tin chung**

Ngày: 10/ 6/2014

Số tiền: 0.000 VND

**Thu/chi**

Chon	Ngày	Thu/Chi	Số tiền
* <input type="checkbox"/>			

- Các control sử dụng trong giao diện thiết kế thu chi: label, textbox, radiobutton, combobox, button, numericupdown, datagridview, groupbox, panel, datetimepicker.

### 3.2 Thiết kế xử lý giao diện

- Giao diện chính frmQuanLyThuChi:
  - Sử dụng thuộc tính Opacity để làm giao diện trong suốt 90%.
  - Sử dụng thuộc tính ControlBox để làm mất nút Minimize, Maximize và Close.
  - Sử dụng thuộc tính Text để đổi tiêu đề form thành Quản lý thu chi.
- DateTimePicker dtpNgay: xử lý sự kiện Load của form để nạp thông tin ngày hiện tại.
- RadioButton rboLoai và rboTaiKhoan:
  - Người dùng có thể chọn cùng lúc thông tin Loại và Tài khoản bằng Panel.
  - Bắt sự kiện CheckedChanged của radio button tài khoản ví, tài khoản ngân hàng, thu và chi để hiển thị các thành phần trong combo box danh mục tương ứng. Các danh mục của tương ứng của từng phần như sau:

Tài khoản/Loại	Thu	Chi
Ví	Rút tiền, Học bổng	Ăn uống, Giải trí, Học tập
Thẻ ngân hàng	Học bổng, Nạp tiền	Rút tiền, Đóng tiền

- nudSoTien:
  - Sử dụng thuộc tính Increment để khi nhấn mũi tên sẽ tăng/giảm 5000 VND.
  - Sử dụng thuộc tính Maximum và Minimum để thiết lập giá trị lớn nhất là 50000000 và giá trị nhỏ nhất là 500.
  - Sử dụng thuộc tính ThousandsSeperator để ngăn cách các số thuộc hàng ngàn.
- grdThuChi:

- Tạo các cột: Chọn loại DataGridViewCheckBoxColumn; Ngày, Thu/Chi, Số tiền, Danh mục, Tài khoản loại DataGridViewTextBoxColumn.
- Không tự động phát sinh các cột bằng cách gán thuộc tính AutoGenerateColumns trong sự kiện Load của form.
- Xử lý sự kiện CellClick:
  - ✧ Khi người dùng chọn 1 ô trên grdThuChi, lấy các dữ liệu trong trên grdThuChi thể hiện lại lên các ô trong giao diện.
  - ✧ Highlight toàn bộ dòng của ô được chọn.
- btnDong: sử dụng hàm Close của lớp Form để đóng giao diện khi người dùng Click vào nút Đóng.
- btnThem:
  - Xử lý sự kiện Click:
    - ✧ Kiểm tra xem Danh mục có trống không, nếu trống thì sử dụng hàm Show của lớp MessageBox để hiển thị thông báo lỗi.
    - ✧ Kiểm tra ngày có nhỏ hơn ngày hiện tại không, nếu nhỏ hơn thì sử dụng MessageBox để báo lỗi.
    - ✧ Thêm thông tin vào grdThuChi.
- btnXoa: Lặp qua từng phần tử trong thuộc tính SelectedRows của grdThuChi, gán mỗi phần tử bằng biến item. Sau đó, sử dụng hàm RemoveAt của thuộc tính Rows trong grdThuChi và truyền vào item.Index để xóa phần tử tại vị trí được chọn.

#### 4 Kết nối giao diện Đăng nhập và Quản lý thu chi

Mục tiêu của phần này, sau khi đã đăng nhập thành công, mở giao diện Quản lý thu chi cùng với hiện tên tài khoản của người đã đăng nhập vào vị trí góc dưới bên trái của giao diện Quản lý thu chi.

