

LAB 10 - LẬP TRÌNH GIAO DIỆN NÂNG CAO

A. Mục tiêu:

Tạo giao diện người dùng với AWT và SWING

- ✓ Tiếp tục phát triển tổ chức giao diện người dùng đồ họa với các control khác
- ✓ Lập trình tùy biến các đối tượng GUI dạng danh sách như: JCheckBox, JRadio, JComboBox, JList.
- ✓ Các control nâng cao khác: JTable, JColorChoose, JTabbedPane, JSlider
- ✓ Thiết kế Menu: MenuBar, Menu, MenuItem, PopupMenu, Toolbar
- ✓ Thiết kế giao diện người dùng với các control nâng cao trên công cụ trực quan NetBeans IDE.

B. Nội dung

Bài tập 1:

Xây dựng màn hình ứng dụng quản lý thông tin nhân viên (có thể sử dụng thao tác kéo thả trong NetBeans)

Yêu cầu:

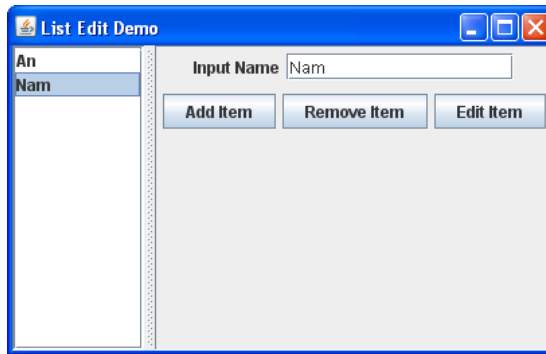
1. Khi chọn nút “ĐĂNG KÝ”, yêu cầu người dùng nhập Họ tên. Nếu người dùng chưa nhập, ứng dụng sẽ hiển thị thông báo. Ngược lại ghi thông tin nhân viên

đăng ký ra tập tin văn bản HOSONV.TXT, theo quy ước mỗi dòng trong tập tin là thông tin của mỗi nhân viên, các thông tin được phân biệt bằng dấu ; (Lưu ý: nếu tập tin HOSONV.TXT đã tồn tại thì cho phép ghi thêm).

2. Khi người dùng chọn nút “BỎ QUA”, thiết lập nội dung hiển thị của các thành phần trên màn hình giống như lúc ban đầu.

Bài tập 2:

Thiết kế giao diện như hình sau:



Yêu cầu xử lý:

- + Khi nhấn nút “**Add Item**” thì thêm nội dung ô nhập vào JList, cảnh báo người dùng trường hợp để trống ô nhập hoặc nhập trùng dữ liệu đã có.
- + Khi nhấn nút “**Remove Item**” thì cho phép xóa các dòng đang chọn trong JList, trước khi xóa phải hỏi xác nhận lại, cảnh báo người dùng trường hợp không chọn mà xóa.
- + Khi nhấn nút “**Edit Item**” thì cho phép sửa nội dung dòng đang chọn thành nội dung mới trong ô nhập liệu.
- + Khi nhấn chọn một dòng trên JList thì hiện nội dung dòng đó lên ô nhập liệu.

Bài tập 3:

Xây dựng ứng dụng tra cứu từ điển đơn giản.

Thiết kế giao diện Swing (có thể dùng kéo thả) và xử lý ứng dụng từ điển đơn giản như sau:



Yêu cầu xử lý:

- Danh sách các từ đọc từ tập tin **dict.dat** có cấu trúc mỗi dòng là cặp giá trị tiếng anh và nghĩa tiếng việt như sau:

```
benefit:lợi, lợi ích  
big:to,lớn  
boy:con trai  
buy:mua hàng  
cat:con mèo
```

và lưu sẵn vào đối tượng Word (từ) của ArrayList, mỗi Word gồm có các thuộc tính gồm : từ tiếng anh, nghĩa tiếng việt.

- Khi nhập vào JTextField 1 từ tiếng anh cần tra thì chương trình sẽ dò tìm đến chữ nào khớp với ký tự gần nhất trong JList.

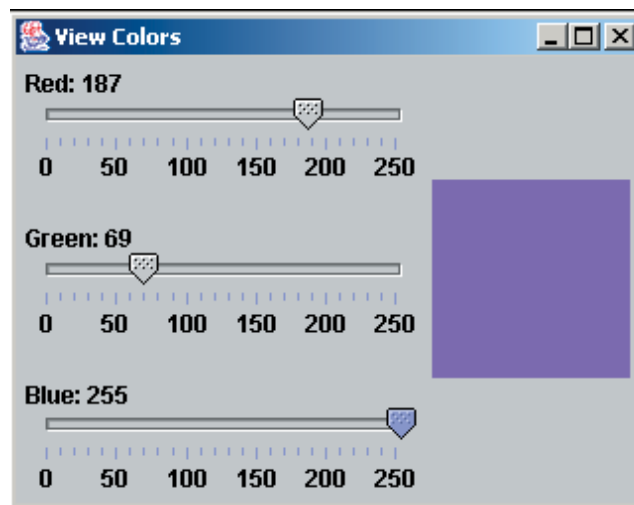
- Khi nhấn nút Enter trên JTextField hoặc nhấn nút **Tra từ** thì nghĩa tương ứng của từ sẽ hiển thị vào JTextArea bên phải tương ứng.

- ❖ **Mở rộng (Về nhà làm thêm):** Bổ sung chức năng thêm từ, cập nhật từ, xóa từ vào ứng dụng trên.



Bài tập 04:

Viết chương trình đổi màu panel dùng JSlider như hình:



Bài tập 05:

Thiết kế giao diện như hình bên dưới:

First Name	Last Name	Years
Plato		428-347 B.C.
Thomas	Aquinas	1225-1274
Soren	Kierkegaard	1813-1855

+ Khi người dùng nhấn nút “**Add Philosopher**” thì chương trình lần lượt hiện ra các cửa sổ cho nhập vào First Name, Last Name và Years, sau đó đưa các thông tin này lên JTable

+ Khi người dùng nhấn nút “**Remove Selected Philosopher**” thì chương trình sẽ xóa mục đang chọn trong JTable, nếu không chọn mục nào để xóa thì phải thông báo, trước khi xóa phải hỏi xác

Bài tập 06:

Viết chương trình quản lý Account

Acc Number	Acc Name	Acc Money
12345	Nguyen Van Teo	50000
12346	Tran Van Ty	6000

Bài tập 07:

(i) Viết lớp NhanVien với các yêu cầu sau:

Thuộc tính: **mã nhân viên** (String), **phân xưởng** (String), **số sản phẩm** (int).

Các hàm khởi tạo.

Các hàm get/set.

Hàm **getChuan()**: trả về giá trị 300 nếu là phân xưởng A, còn lại trả về 500. (chỉ có các phân xưởng là A, B, C, D).

Hàm **VuotChuan()**: trả về true khi số sản phẩm vượt chuẩn (chuẩn tùy theo phân xưởng), ngược lại trả về false.

Hàm **TinhLuong()**: trả về lương của nhân viên, lương = số sản phẩm * đơn giá, nếu số sản phẩm vượt chuẩn thì phần vượt được tính đơn giá là 30000, còn lại tính đơn giá là 20000.

Hàm **toString()**: trả về mã nhân viên.

(ii) Thiết kế giao diện như hình bên dưới

Ma nhan vien	Phan xuong	So sp	Vuot chuan
M001	A	500	x
M002	B	500	
M003	A	200	

Giao diện cho nhập mã nhân viên, số sản phẩm, chọn phân xưởng. Yêu cầu ô Tiền lương được định dạng như hình.

Yêu cầu xử lý:

- + Khi người dùng chọn phân xưởng nào trên combobox thì hiện Số sản phẩm chuẩn tương ứng.
- + Nút “**Tính lương**” : xuất ra lương của nhân viên đang nhập (xem hình). Yêu cầu kiểm tra dữ liệu nhập.
- + Nút “**Thêm**” : thêm nhân viên đang nhập vào JTable với các cột như hình. Chú ý không được thêm khi:
 - Không nhập đủ dữ liệu.
 - Số sản phẩm không phải là số.
 - Trùng mã nhân viên.
- + Nút “**Xóa**” : xóa một nhân viên đang chọn trên JTable. Chú ý phải hỏi trước khi xóa và không được xóa khi chưa chọn dòng nào.
- + Khi chọn 1 dòng trên JTable thì hiện thông tin nhân viên đó lên các ô nhập liệu (xem hình).
- + Khi người dùng nhấn nút “**Sửa**” thì chương trình sửa thông tin của nhân viên đang chọn vào JTable. Chú ý không được sửa khi:
 - Không có đủ dữ liệu.
 - Số SP không phải là số.
 - Trùng mã nhân viên.

Bài tập 08:

(i).Viết lớp SinhVien với yêu cầu sau:

Thuộc tính: mã sinh viên (String), họ tên (String), mã lớp (String), điểm môn 1 (double), điểm môn 2 (double).

Các hàm khởi tạo.

Các hàm get/set.

Hàm DiemTrungBinh(): trả về điểm trung bình của 2 môn học.

Hàm KetQua(): trả về “Đậu” khi điểm trung bình ≥ 5 , ngược lại là “Rớt”. Hàm toString(): trả về mã sinh viên.

(ii) Thiết kế giao diện như hình bên dưới

Giao diện cho nhập mã sinh viên, họ tên, chọn mã lớp, nhập điểm môn 1, môn 2.

Ma sinh vien	Ten sinh vien	Ma Lop	Diem mon 1	Diem mon 2	Diem trung binh	Ket qua
0712345	Nguyen Thi An	TCTH30A	5.0	6.0	5.5	DAU
0712346	Nguyen An	TCTH30A	4.0	3.0	3.5	ROT

Yêu cầu xử lý:

+ Nút “**Kết quả**”: xuất ra điểm trung bình và kết quả của sinh viên đang nhập (xem hình). Chú ý ô nhập Điểm môn 1 và môn 2 phải là số.

+ Nút “**Thêm**”: thêm một sinh viên vào table với các cột như hình. Chú ý không được thêm khi:

- Không nhập đủ dữ liệu.
- Điểm môn 1 và môn 2 không phải là số.
- Trùng mã sinh viên.

+ Nút “**Xóa**”: xóa một sinh viên đang chọn trên table. Chú ý phải hỏi trước khi xóa và không được xóa khi chưa chọn dòng nào.

+ Khi chọn 1 dòng trên table thì hiện thông tin sinh viên đó lên các ô nhập liệu (xem hình).

Bài tập 09:

(i) Viết lớp SinhVien với yêu cầu sau:

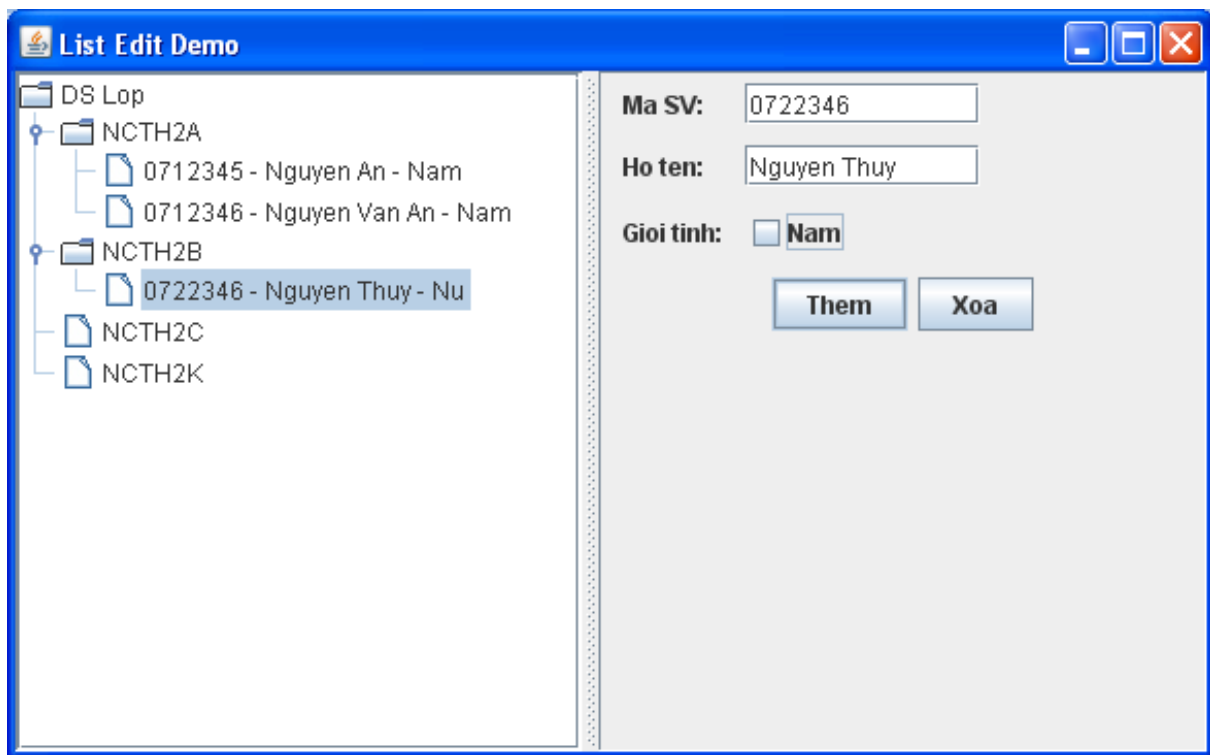
Thuộc tính: mã sinh viên (String), họ tên (String), giới tính (boolean), mã lớp (String).

Các hàm khởi tạo.

Các hàm get/set.

Hàm toString(): trả về chuỗi thông tin gồm: mã sinh viên – họ tên – giới tính.

(ii) Thiết kế giao diện như hình bên dưới:



Yêu cầu giao diện: Trên JTree có sẵn nút gốc “DS lop” và các nút con là các mã lớp: NCTH2A, NCTH2B, NCTH2C, NCTH2K.

Yêu cầu xử lý:

+ Nút “**Thêm**”: thêm một sinh viên vào lớp đang chọn trên JTree (xem hình). Chú ý không được thêm khi:

- Không nhập đủ dữ liệu.
- Không chọn mã lớp trên JTree.
- Trùng mã sinh viên.

+ Nút “**Xóa**”: xóa sinh viên đang chọn trên JTree. Chú ý phải hỏi trước khi xóa và không được xóa khi:

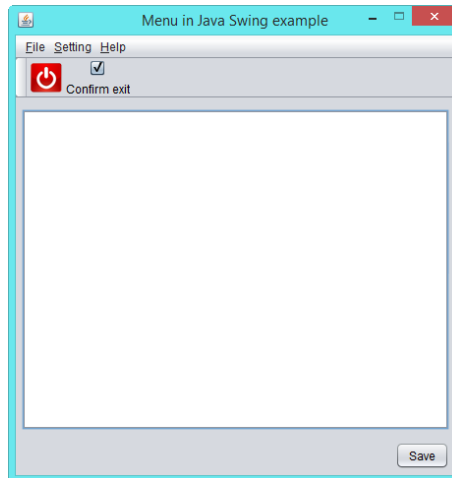
- Không chọn nút muốn xóa.
- Chọn vào nút gốc hoặc nút mã lớp.

+ Khi chọn nút sinh viên thì hiện thông tin sinh viên đó lên các ô nhập liệu (xem hình).

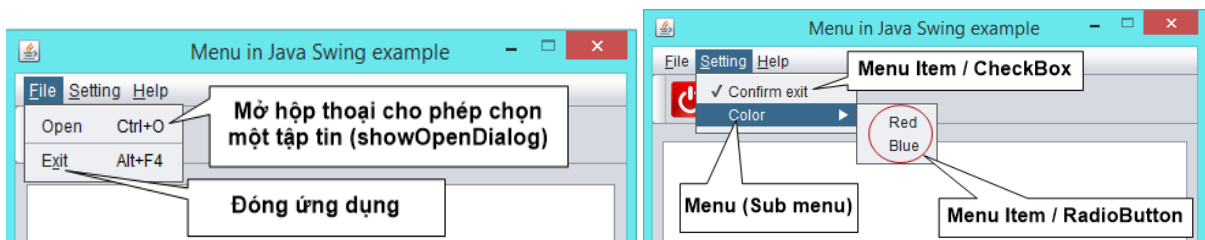
Bài tập 10:

Thiết kế và xử lý giao diện Menu

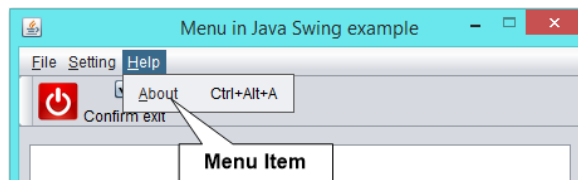
1. Thiết kế giao diện ứng dụng



- File menu và Setting menu

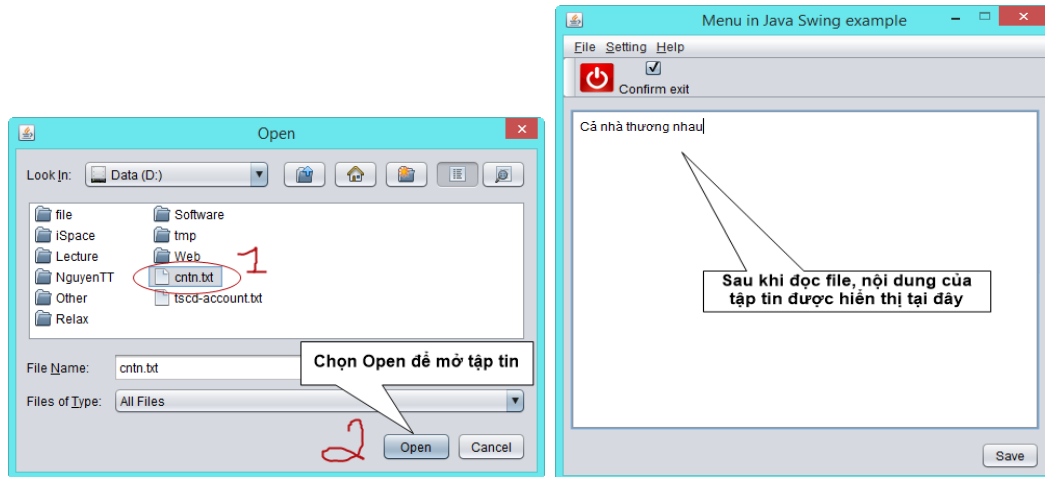


- Help menu

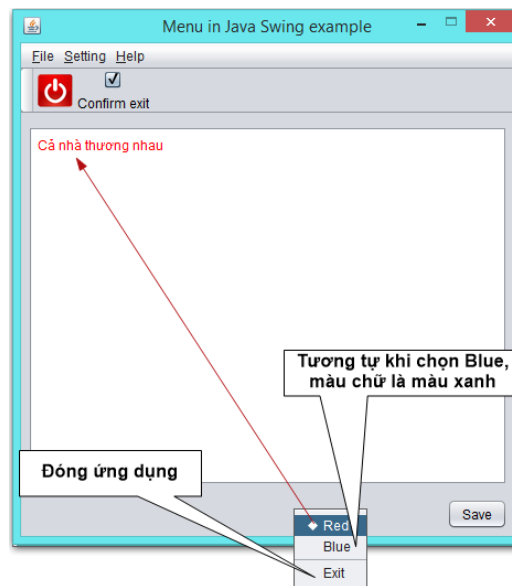


2. Xử lý sự kiện trên giao diện

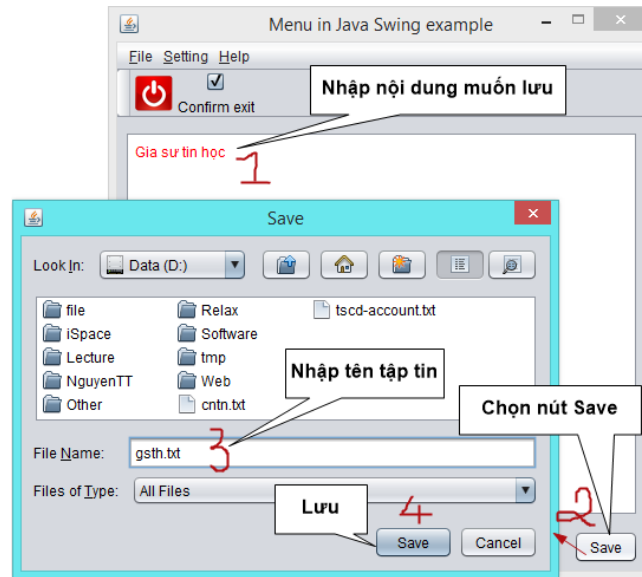
- Khi người dùng chọn Open menu, hiển thị hộp thoại cho phép người dùng chọn một tập tin txt và đọc nội dung tập tin đó hiển thị ra màn hình



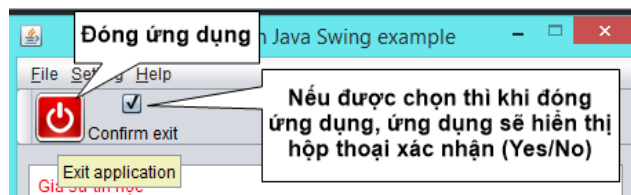
- Khi người dùng chọn **Exit** menu, ứng dụng sẽ bị đóng (Ứng dụng sẽ hiển thị hộp thoại xác nhận hay không còn tùy thuộc vào **Confirm exit** menu có được chọn?)
- **Popup menu** cho phép người dùng thiết lập màu chữ và đóng ứng dụng



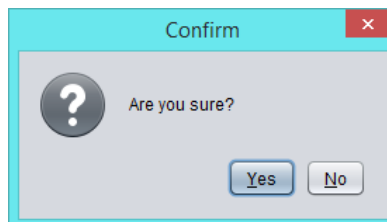
1. Khi người dùng chọn nút “**Save**”, ứng dụng sẽ hiển thị hộp thoại cho phép người dùng lựa chọn nơi lưu tập tin.



2. Xử lý tool bar:



Hộp thoại xác nhận khi đóng ứng dụng. Ứng dụng sẽ bị đóng nếu chọn Yes.



3. Xử lý cho Setting menu

Confirm exit, Red và Blue xử lý tương tự như **Confirm exit** của **tool bar** và **Red/Blue** của **popup menu**