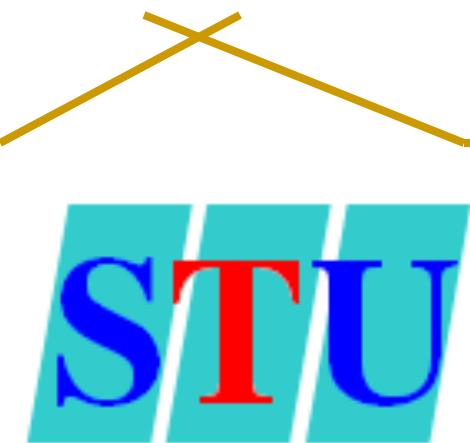
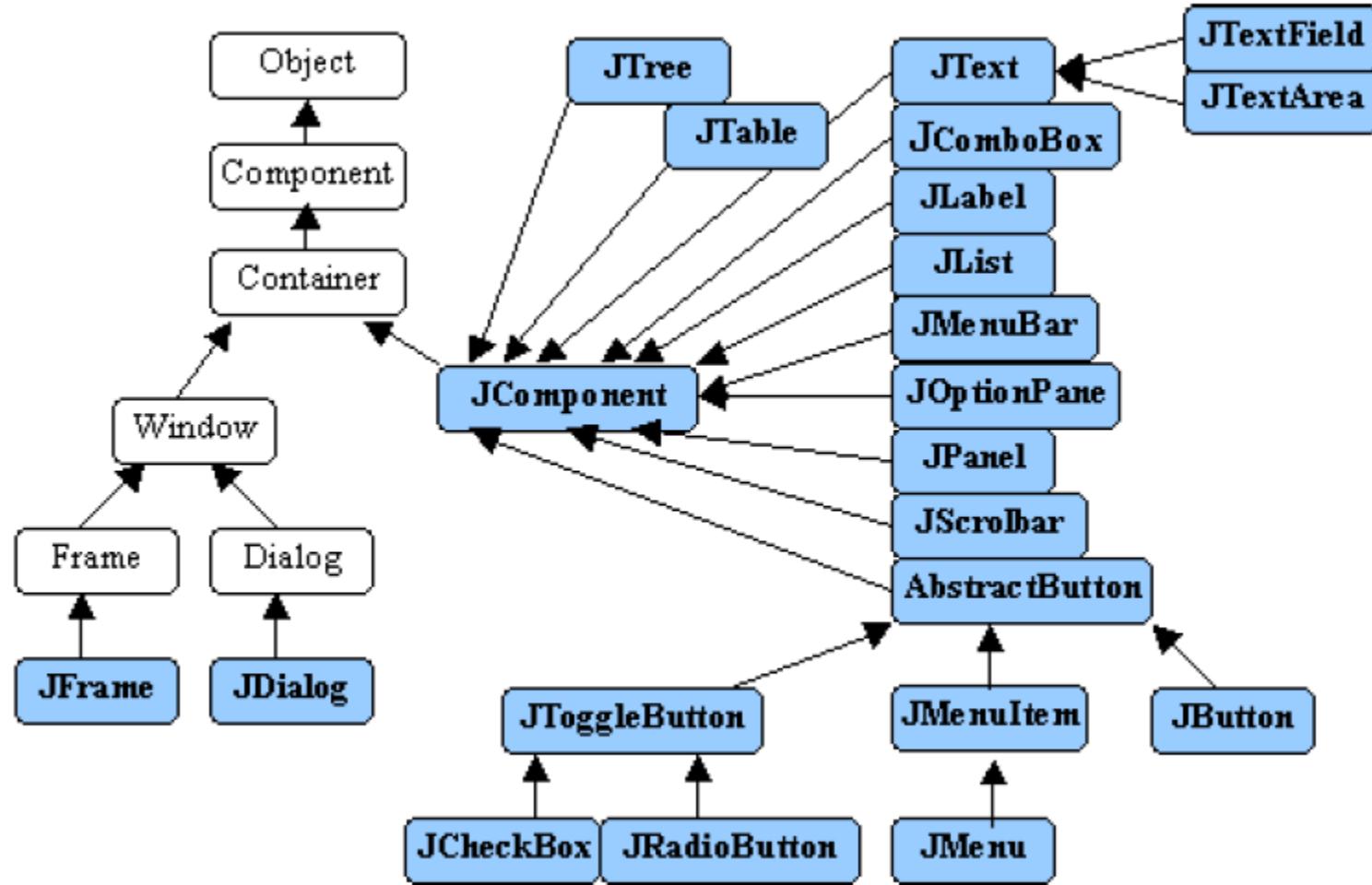


Chương 4: Graphics Programming



Trường Đại học Công nghệ Sài Gòn
Khoa Công nghệ Thông tin

Kiến trúc Swing



Container trong Java Swing

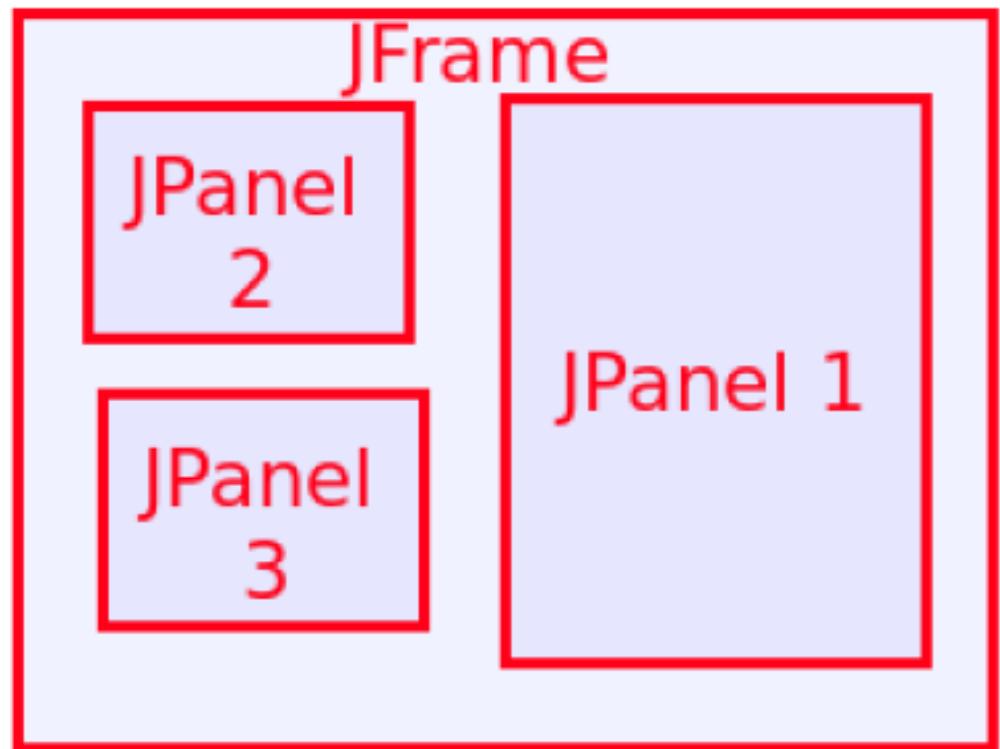
- Có 2 kiểu trình chứa: Top-level-container và Multi-purpose container
- Top Level Container
 - JFrame
 - JDialog
 - JApplet: được sử dụng cho web
 - JWindow: Là màn hình chờ khi mới bắt đầu ứng dụng (Splash screen)

Container trong Java Swing

- Có 2 kiểu trình chứa: Top-level-container và Multi-purpose container

- Multi-purpose container

- JPanel
- JLayered
- JDesktopPane
- JInternalFrame



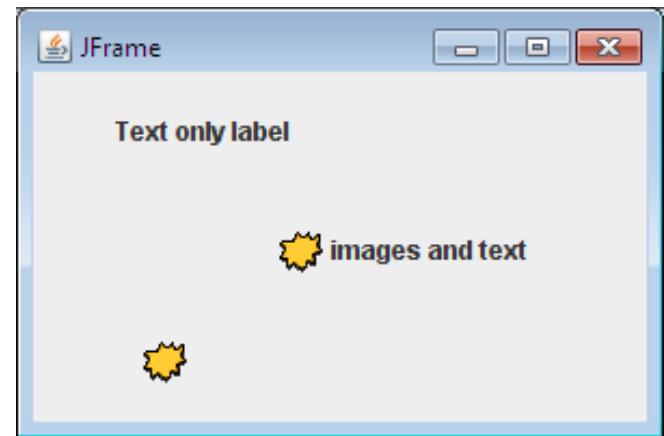
Các thành phần GUI căn bản



- **Label**
- **Text Input**
- **Choice Components**
- **List**
- **Menus**
- **Mnemonics**
- **Toolbars**
- **Tooltips**
- **Dialog Boxes**

Label

- Tạo Nhãn dạng chuỗi : **JLabel("Chuỗi")**
- Tạo Nhãn dạng biểu tượng : **JLabel(icon);**
 - Tạo đối tượng ảnh:



`ImageIcon icon=new ImageIcon(imgURL,"");`

- Tạo Nhãn dạng biểu tượng và chuỗi
 - **JLabel(icon,"chuỗi", JLabel.CENTER)**

JTextField



■ **JTextField()**

- Tạo textfield rỗng với 1 columns

■ **JTextField(String s)**

- Tạo textfield có nội dung s

■ **JTextField(int cols)**

- Tạo textfield rỗng với cols columns

JTextField



■ **JTextField(String text, int cols)**

ví dụ

```
JTextField mmText = new JTextField (10);
```

```
JTextField txtName = new JTextField ("Hello", 10);
```

Phương thức:

getText():String

setText(String s)

JPasswordField



- Sử dụng nhập mật khẩu
- Giống textfield, ngoại trừ kí tự nhập bởi người dùng chỉ xuất hiện các kí tự *
- Có thể thay đổi kí tự xuất hiện:

public void setEchoChar(char c)

JPasswordField



- **String getText()**

- **char[] getPassword()**

- Trả về dãy kí tự trong password field

- Dùng Arrays.equals(char[],char[]) so sánh 2 dãy kí tự

- **char getEchoChar()**

- **void setEchoChar(char ch)**

JFormattedTextField

- Thành phần sử dụng input và output thông tin có định dạng. Thừa kế từ JTextField và thêm 2 đối tượng formatter và value
 - Formatter chuyển value của field vào text
- Sử dụng nhập số và ngày

JFormattedTextField

- Sử dụng mặt nạ định dạng nhập: `format=new MaskFormatter("chuỗi định dạng");`
- Kí tự sử dụng thành lập chuỗi mặt nạ:
 - # : kí tự số
 - U: thường thành hoa
 - L: Hoa thành thường
 - A: kí tự hoặc số
 - ?: kí tự bất kì
 - *: tùy ý
 - H: kí tự số Hex
- VD: "###-####"

JFormattedTextField

○ Sử dụng thêm phương thức

- format.setValidCharacters : cài đặt kí tự hợp lệ
- format.setInvalidCharacters : cài đặt kí tự không hợp lệ
- JFormattedTextField(format);

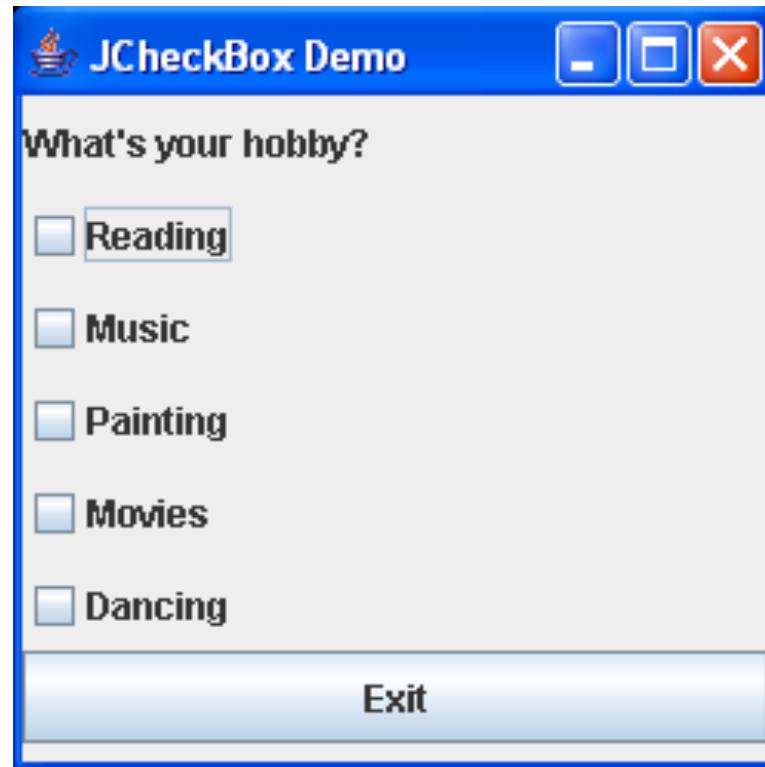
JSpinner

- Thành phần sử dụng input thông tin định trước
- new JSpinner(danh sách chuỗi)
 - VD: String[] list={"chuoi1","chuoi2","chuoi3"}
 - new JSpinner(list)
- getValue(): truy vấn dữ liệu

Choice Components



- JCheckBox
- JRadioButton





- Thành phần được dùng cho phép người dùng nhiều lựa chọn.
- check box gây ra một Item event khi trạng thái thay đổi (ON hoặc OFF)
- Hiện thực interface ItemListener xử lí sự kiện
- Checkbox gọi phương thức itemStateChanged :

```
public void itemStateChanged (ItemEvent e)
```

JCheckBox



■ **boolean isSelected()**

- Trả về trạng thái của checkbox

■ **void setSelected(boolean state)**

- Thay đổi trạng thái

■ **String getText()**

■ **void setText(String text)**

- Tại một thời điểm chỉ một button trong group của radio buttons được chọn.
- Mỗi JRadioButtons được thêm tới một đối tượng **ButtonGroup**
- ButtonGroup chỉ điều khiển hành vi của button
- Khi người dùng chọn ,JRadioButton gây ra một Action event

- **boolean isSelected()**
- **void setSelected(boolean state)**
- **String getText()**
- **void setText(String text)**

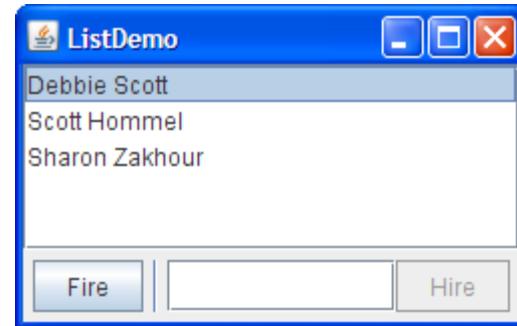
JList

```
listModel = new DefaultListModel();
listModel.addElement("Debbie Scott");
listModel.addElement("Scott Hommel");
listModel.addElement("Alan Sommerer");

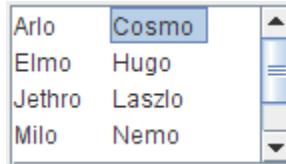
list = new JList(listModel);

listScroller = new JScrollPane(list);

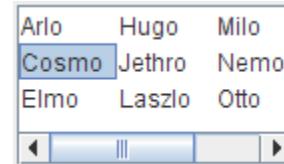
listScroller.setPreferredSize(new Dimension(250, 80));
```



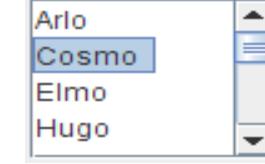
```
list.setLayoutOrientation(JList.HORIZONTAL_WRAP);
```



HORIZONTAL_WRAP



VERTICAL_WRAP



VERTICAL

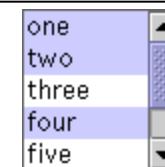
```
list.setSelectionMode(ListSelectionModel.SINGLE_INTERVAL_SELECTION);
```



SINGLE_SELECTION



SINGLE_INTERVAL_SELECTION



MULTIPLE_INTERVAL_SELECTION

Jlist (tt)

■ Khởi dụng

- JList(ListModel)
- JList(Object[])
- JList(Vector)
- JList()

Jlist (tt)

- void addListSelectionListener(ListSelectionListener)
- void setSelectedIndex(int)
- int getSelectedIndex()
- int getMinSelectionIndex()
- int getMaxSelectionIndex()
- void setEditable(boolean)

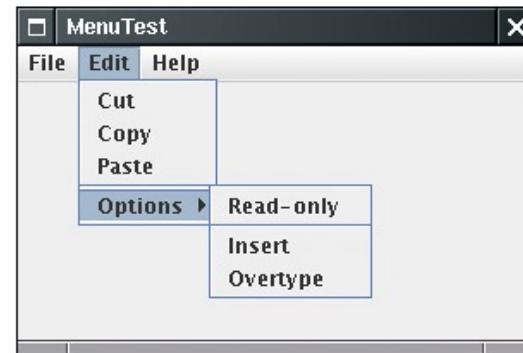
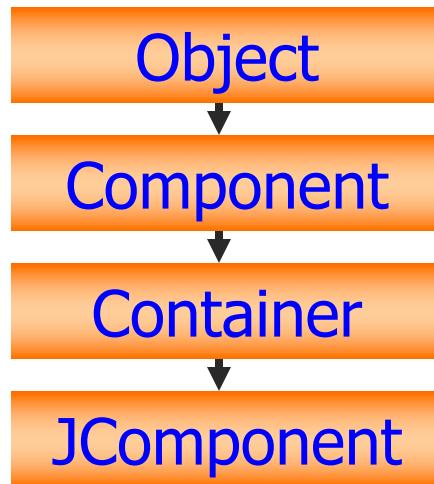
DefaultListModel

- **void add(int index, Object element)**
- **void addElement(Object obj)**
- **void clear()**
- **boolean contains(Object elem)**
- **void copyInto(Object[] anArray) , object[] toArray()**

DefaultListModel

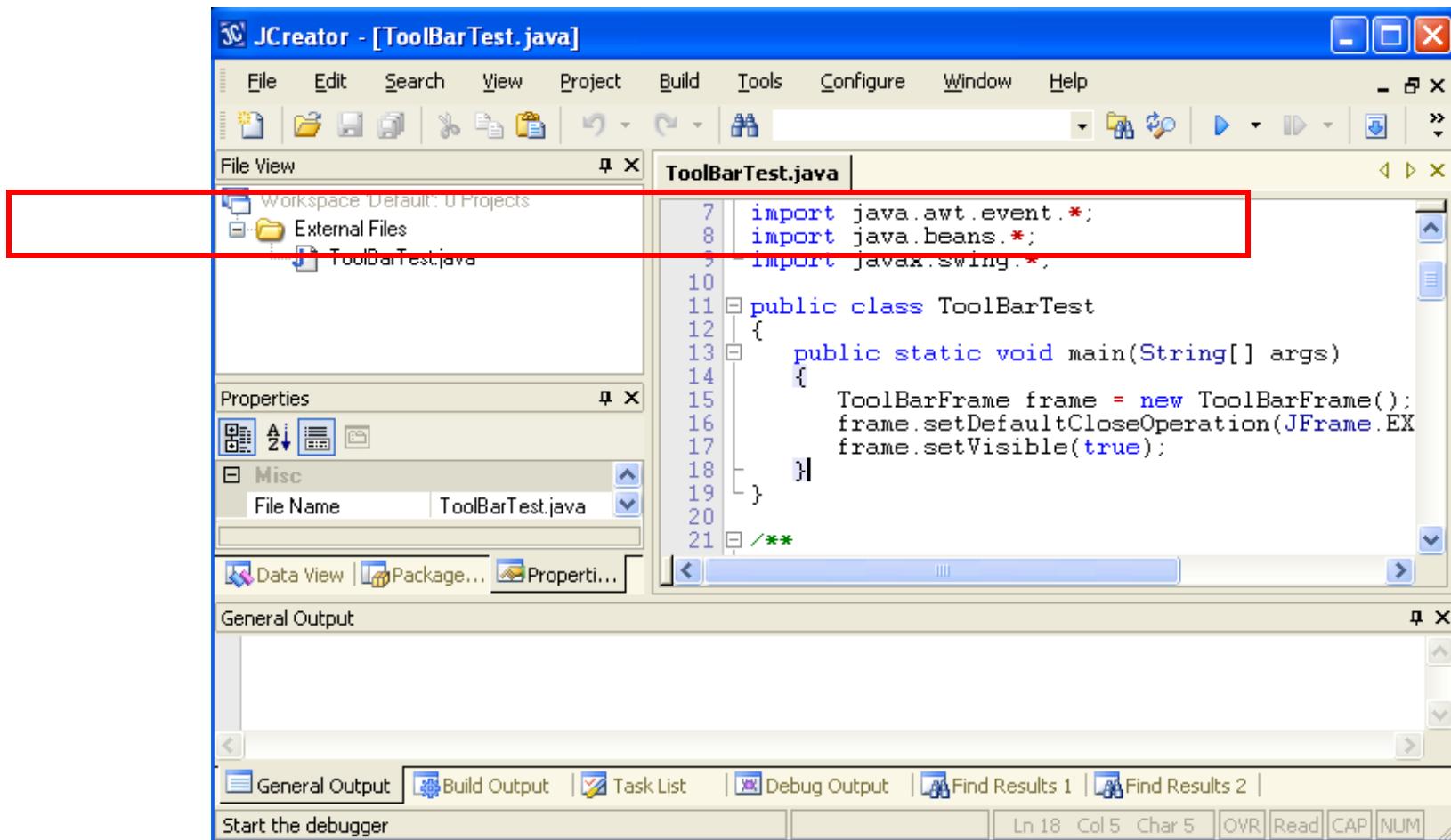
- object firstElement() , lastElement()
- int getSize(), size()
- Int indexOf(Object elem) Tìm kiếm
- boolean isEmpty()
- object remove(int index)
- void removeRange(int fromIndex, int toIndex)
- setElementAt(Object obj, int index)

Menu



Toolbars

- Toolbar là thành phần chứa các nút lệnh cho phép truy xuất nhanh một số lệnh phổ biến trong chương trình.



Tooltips

- Tooltip là thành phần cho phép xuất hiện một chuỗi mô tả khi mouse lướt qua một thành phần.
- Tooltip sẽ mất khi di chuyển mouse sang vị trí khác.
- Tạo tooltip:
 - `public JToolTip()`
- Có thể thêm tooltip tới bất kì JComponent, sử dụng phương thức `setToolTipText`:
 - `void setToolTipText(String text)`

Ví dụ:

```
exitButton.setToolTipText("Exit");
```

Dialog Boxes

- Dialog box là cửa sổ xuất hiện trên đỉnh của cửa sổ hoạt động.
- Sử dụng để:
 - Truyền một thông điệp
 - Xác nhận một hành động
 - Cho phép người dùng nhập dữ liệu
 - Chọn màu
 - Chọn file. . . .

JOptionPane

- JOptionPane có bốn *phương thức static* :

- **showMessageDialog(...)**

- Hiển thị thông báo, chờ click OK

- **showConfirmDialog(...)**

- Hiển thị thông báo, chờ một xác nhận (VD OK/Cancel)

JOptionPane

- JOptionPane có bốn *phương thức static* :

- **showOptionDialog(...)**

- **Hiển thị thông báo, chờ người dùng chọn một tập chọn lựa**

- **showInputDialog(...)**

- **Hiển thị thông báo và chờ nhận dữ liệu**

- Chuỗi thông tin có thể chứa newline ('\n')

JOptionPane – Các tham số

- Component parent
 - parent component (có thể null)
- Object message
 - Chuỗi thông báo trên dialog (string, icon, component)
- String title
 - Nội dung title bar của dialog

JOptionPane – Các tham số

- int messageType

- Một trong các dạng: ERROR_MESSAGE, INFORMATION_MESSAGE, WARNING_MESSAGE, QUESTION_MESSAGE, PLAIN_MESSAGE

- int optionType

- Một trong các dạng: DEFAULT_OPTION, YES_NO_OPTION, YES_NO_CANCEL_OPTION, OK_CANCEL_OPTION

- Icon icon

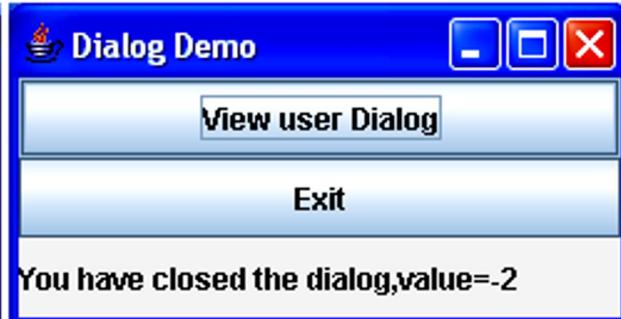
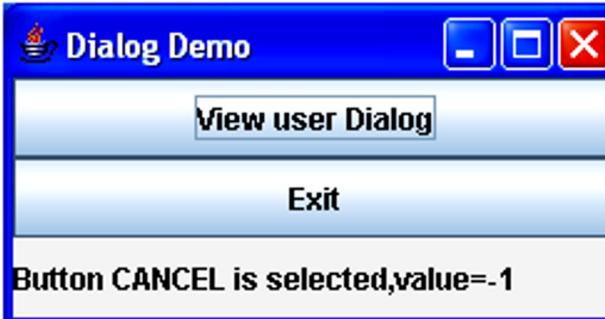
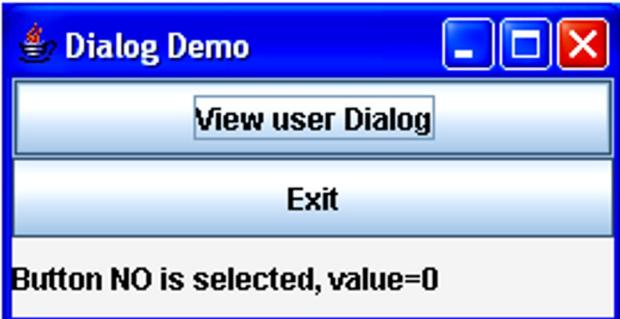
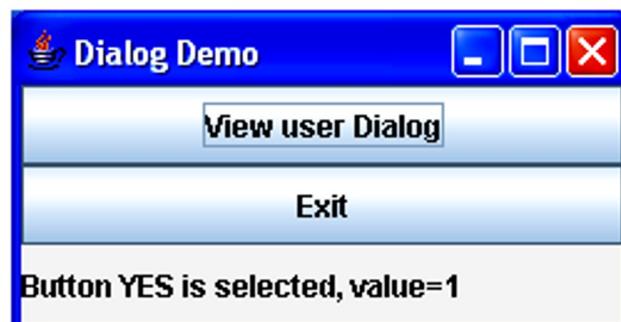
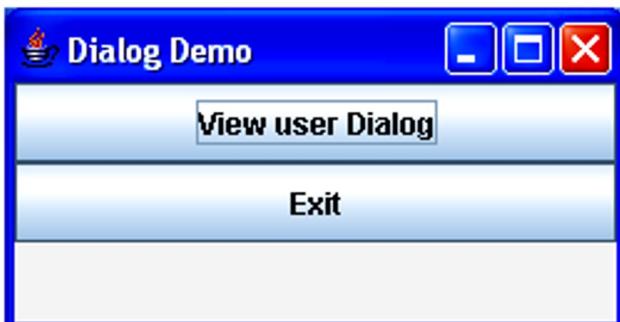
- Thay thế biểu tượng chuẩn

JDialog

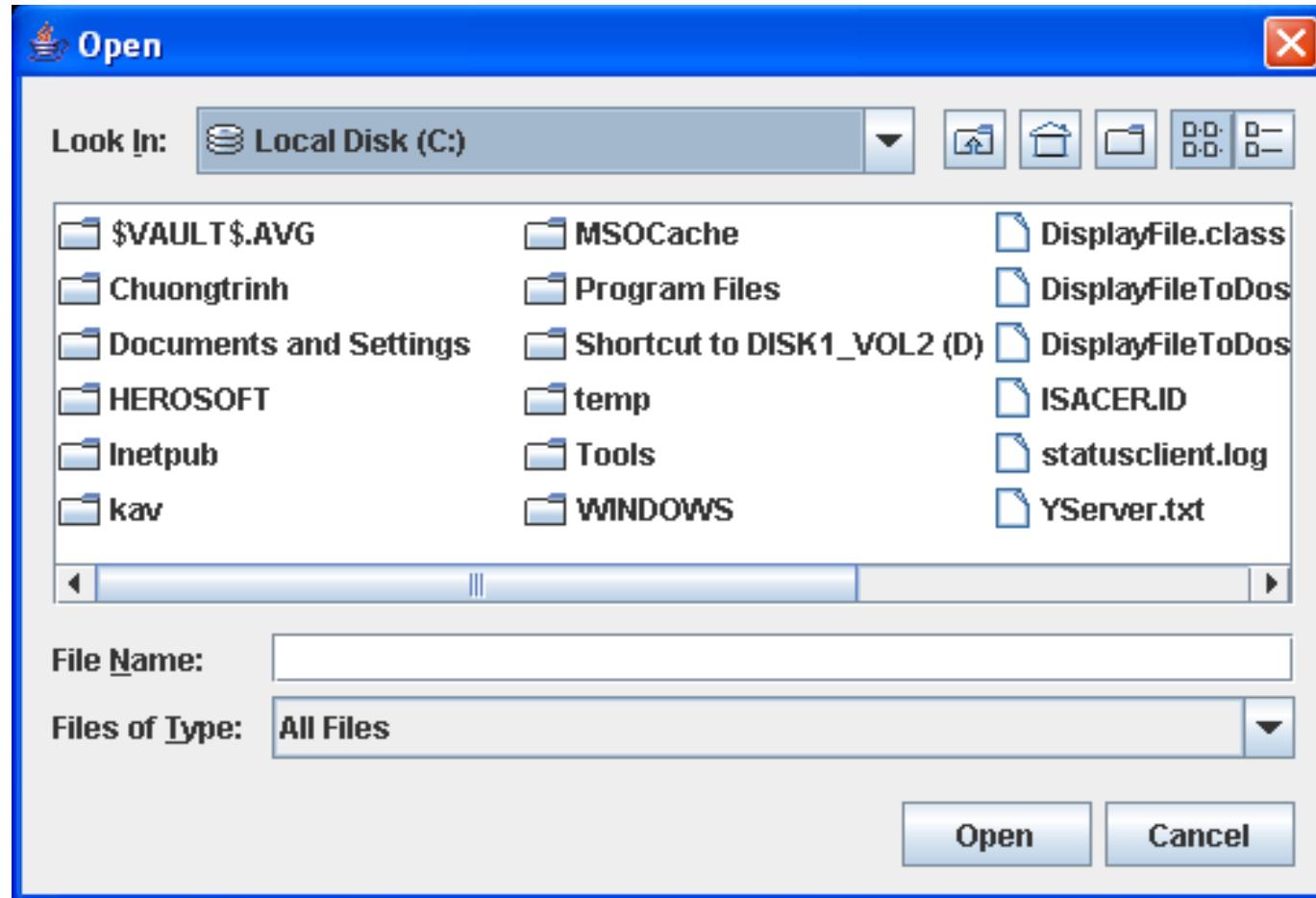
- Xử lý giống main window cho ứng dụng từ JFrame
- Lớp JDialog là phương tiện để tạo hộp thoại của người dùng
- Hiện thực dialog box bằng cách thừa kế JDialog
 - class MyDialog extends JDialog ...{}



JDialogDemo.java



File Dialogs



JFileChooser

- Tạo đối tượng JFileChooser :

- `JFileChooser chooser = new JFileChooser();`

- Hiển thị dialog box, gọi phương thức `showOpenDialog` hay `showSaveDialog`.

- `int result = chooser.showOpenDialog (parent);`

- `int result = chooser.showSaveDialog (parent);`

- **`return JFileChooser.APPROVE_OPTION: chọn (Yes, Ok)`**

- **`return JFileChooser.CANCEL_OPTION: chọn Cancel`**

- **`return JFileChooser.ERROR_OPTION: lỗi`**

JFileChooser

- Nhận đối tượng file chọn:

 - *File f = chooser.getSelectedFile();*

 - *File[] f = chooser.getSelectedFiles();*

- Nhận đường dẫn file chọn:

 - *String filename = chooser.getSelectedFile().getPath();*

JFileChooser

- Chọn chế độ chỉ chọn file, chỉ chọn thư mục hay cả hai:
 - `setFileSelectionMode(JFileChooser.FILES_ONLY)` (mặc định)
 - `setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY)`
 - `setFileSelectionMode(JFileChooser.FILES_AND_DIRECTORIES)`

Color choosers

