

NGÔN NGỮ LẬP TRÌNH JAVA

Giảng viên: TS. Vũ Hữu Tiến

Email: tienvh@ptit.edu.vn

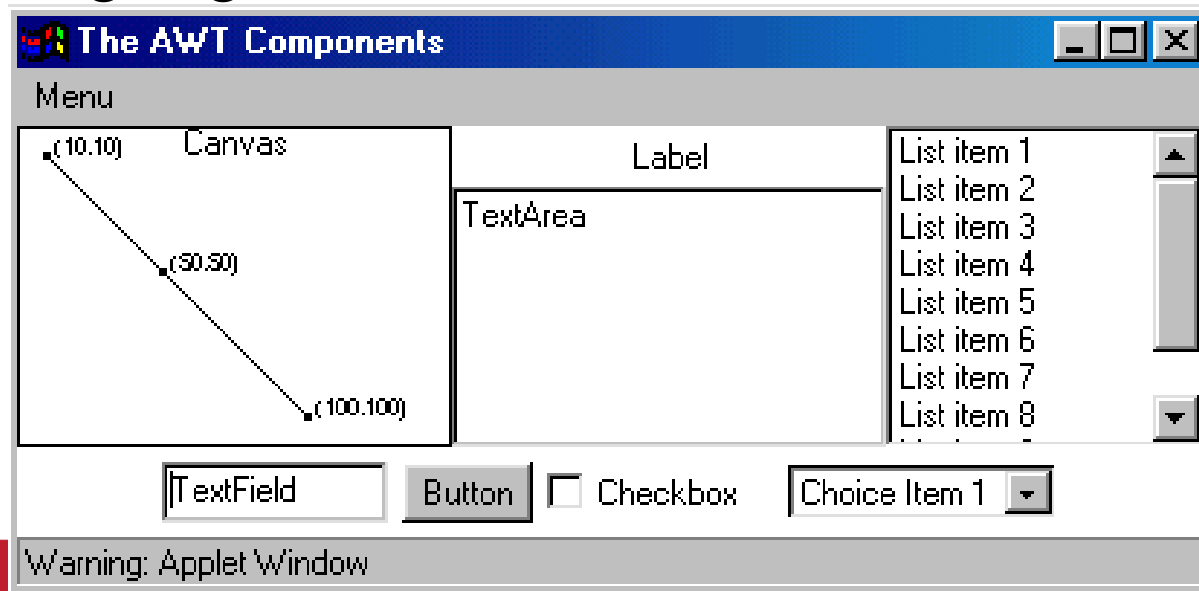
Mobile: 0939919396

GIỚI THIỆU MÔN HỌC

- Nội dung môn học:
 - **Chương 1:** Tổng quan về lập trình hướng đối tượng và ngôn ngữ lập trình Java
 - **Chương 2:** Lớp và đối tượng trong Java
 - **Chương 3:** Tính kế thừa và đa hình trong Java
 - **Chương 4: Lập trình giao diện trong Java**
 - **Chương 5:** Xử lý nhập/ xuất trong Java
 - **Chương 6:** Xử lý ngoại lệ trong Java
 - **Chương 7:** Lập trình đa luồng trong Java

LẬP TRÌNH GIAO DIỆN

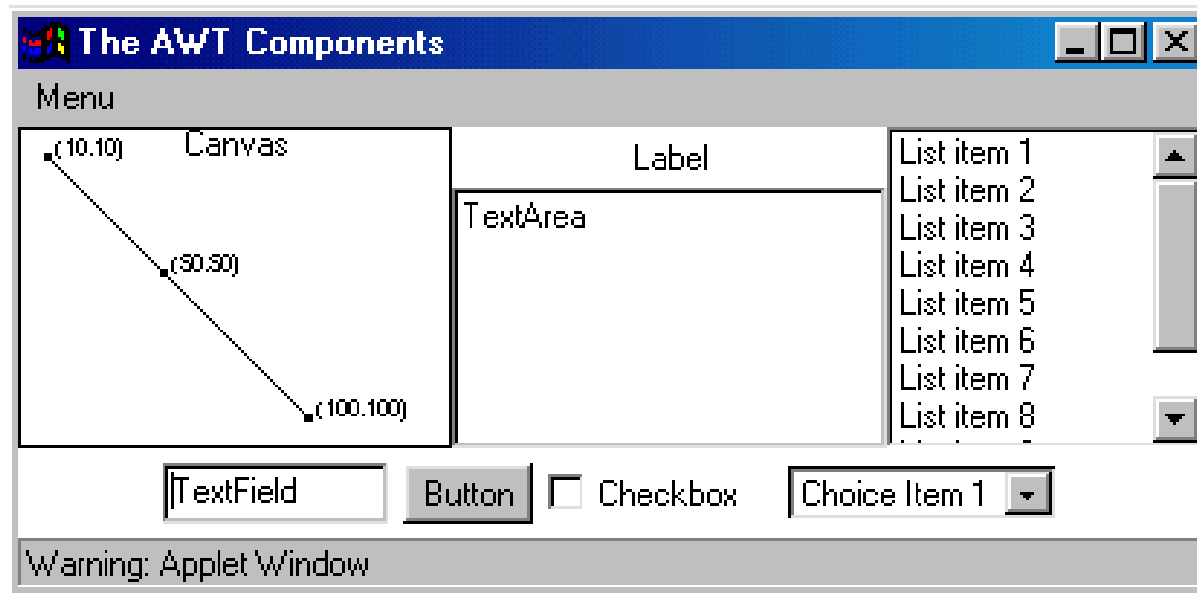
- **AWT: Abstract Window Toolkit.** AW là một API của Java được sử dụng để tạo ra Giao diện đồ họa người sử dụng (Graphical User Interface (GUI)) trong các chương trình Java.
- AWT là một Platform phụ thuộc vào hệ điều hành vì tùy thuộc vào mỗi hđh, các components của AWT sẽ có cách thể hiện tương ứng.



LẬP TRÌNH GIAO DIỆN

- **Component**

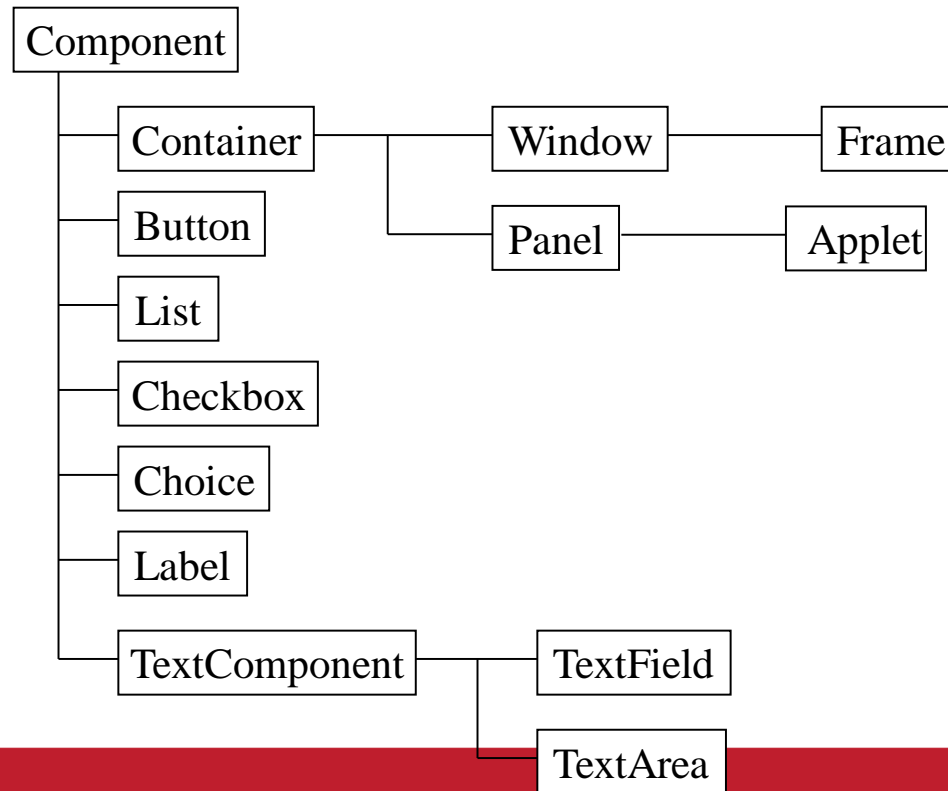
- Component: là một đối tượng biểu diễn một thành phần đồ họa trong giao diện cho phép hiển thị thông tin hoặc cho người dùng tương tác với chương trình.
- Thành phần đồ họa bao gồm label, button, text field, scroll bar và menu.



LẬP TRÌNH GIAO DIỆN

- **Container**

- Container là một loại Component đặc biệt dùng để chứa và tổ chức các component khác.
- Frame và Panel là hai container điển hình của Java



LẬP TRÌNH GIAO DIỆN

- **Frame**

- Constructor:

- Frame(): Tạo frame không có tiêu đề
 - Frame(String title): Tạo một frame có tiêu đề

```
import java.awt.*;
import java.awt.event.*;
public class SimpleExample extends Frame {
    SimpleExample(String title){
        super(title);
    }
    public static void main(String args[]){
        // Creating the instance of Frame
        SimpleExample fr=new SimpleExample("This is the first Frame");
        fr.setSize(300,200);
        fr.setVisible(true);
    }
}
```

LẬP TRÌNH GIAO DIỆN

- **Panel**

- Panel được sử dụng để nhóm các component.
- Panel không hiển thị một cách trực quan. Do đó, Panel phải được add vào Frame để hiển thị.

```
import java.awt.*;
import java.awt.event.*;
public class SimplePanel extends Panel {
    SimplePanel(){
    }
    public static void main(String args[]){
        SimplePanel pn=new SimplePanel();
        Frame fr = new Frame("This is Panel");
        fr.add(pn);
        fr.setSize(300,200);
        fr.setVisible(true);
    }
}
```

LẬP TRÌNH GIAO DIỆN

- **Component: Label**

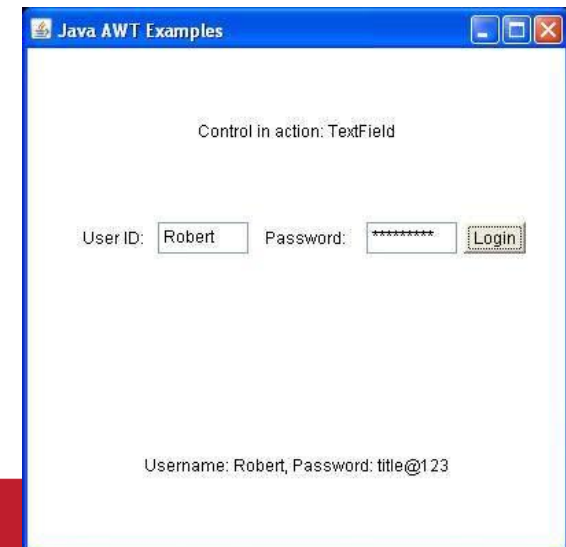
- Nhãn là một Component dùng để hiển thị một dòng text đơn.
- Nhãn có thuộc tính read-only. Tức là người dùng không thể click vào nhãn để thay đổi nội dung text.
- Nội dung text có thể được căn chỉnh trong phạm vi của nhãn.

```
Label aLabel = new Label("Enter password:");  
aLabel.setAlignment(Label.RIGHT);  
aPanel.add(aLabel);
```


LẬP TRÌNH GIAO DIỆN

- **Component: TextField**

- TextField là một vùng cho phép người dùng nhập 1 dòng text trong đó.
- Constructor:
 - TextField()
 - TextField(int columns): tạo ra textfield với số cột cho trước.
 - TextField(String s): tạo ra textfield với dòng text cho trước.
 - TextField(String s, int columns)



LẬP TRÌNH GIAO DIỆN

- **Component: TextArea**

- TextField là một vùng cho phép người dùng nhập nhiều dòng text trong đó.
- Constructor:
 - TextArea()
 - TextArea(int row, int col):
tạo ra text area với số dòng, cột cho trước.
 - TextArea(String s):
tạo ra text area với dòng text cho trước.
 - TextField(String s, int row, int col)



LẬP TRÌNH GIAO DIỆN

- **Component: Button**

- Lớp Button biểu diễn một nút bấm và text kèm theo nút bấm đó.
- Khi một nút được click, nút bấm sẽ gọi đến đối tượng Listeners. Đây là một đối tượng có nhiệm vụ triển khai (implement) giao diện ActionListener để bắt các sự kiện khi người dùng click chuột.

```
Button okButton = new Button("Ok");  
Button cancelButton = new Button("Cancel");
```

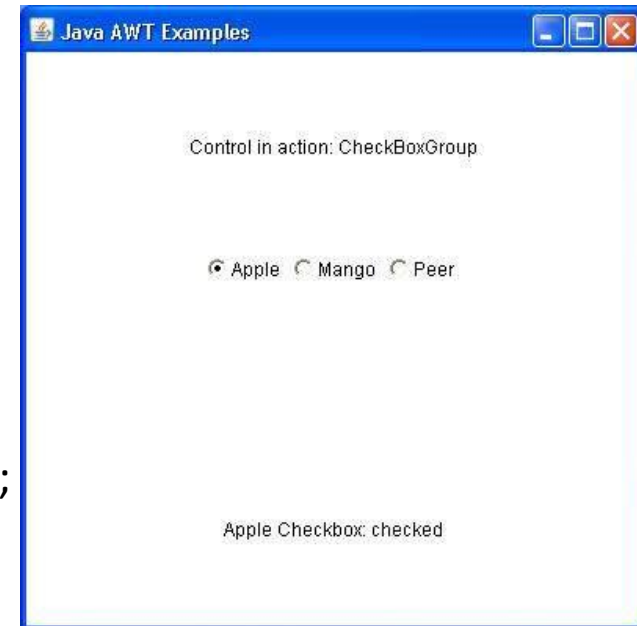
```
pn.add(okButton);  
pn.add(cancelButton);
```

LẬP TRÌNH GIAO DIỆN

- **Component: Checkbox và RadioButton**

- Checkbox: được sử dụng cho việc chọn nhiều phương án. Người dùng click vào checkbox để chọn hoặc hủy lựa chọn.
- Radiobutton: tương tự checkbox nhưng radiobutton chỉ cho phép chọn một trong số các phương án đã đưa ra.
- Khởi tạo:
 - Checkbox()
 - Checkbox(String text)
 - CheckboxGroup();

```
CheckboxGroup fruitGroup = new CheckboxGroup();  
Checkbox chkApple = new Checkbox("Apple",fruitGroup,true);  
Checkbox chkMango = new Checkbox("Mango",fruitGroup,false);  
Checkbox chkPeer = new Checkbox("Peer",fruitGroup,false);
```

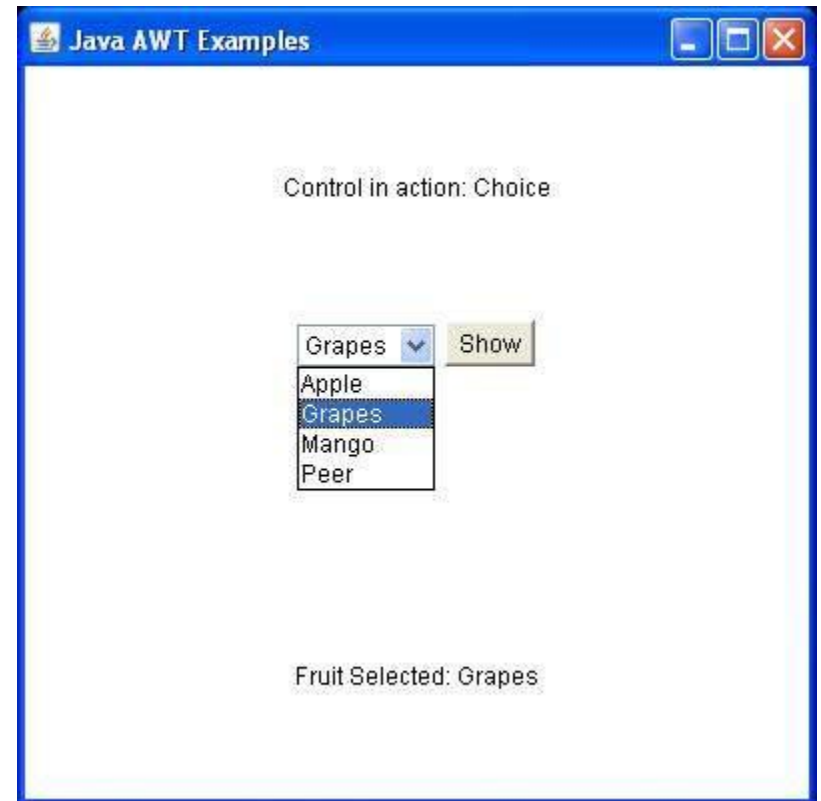


LẬP TRÌNH GIAO DIỆN

- **Component: Choice List**

- Hiển thị một danh sách các lựa chọn trên giao diện người dùng. Người dùng có thể chọn một hoặc nhiều item trên danh sách.
- Constructor:

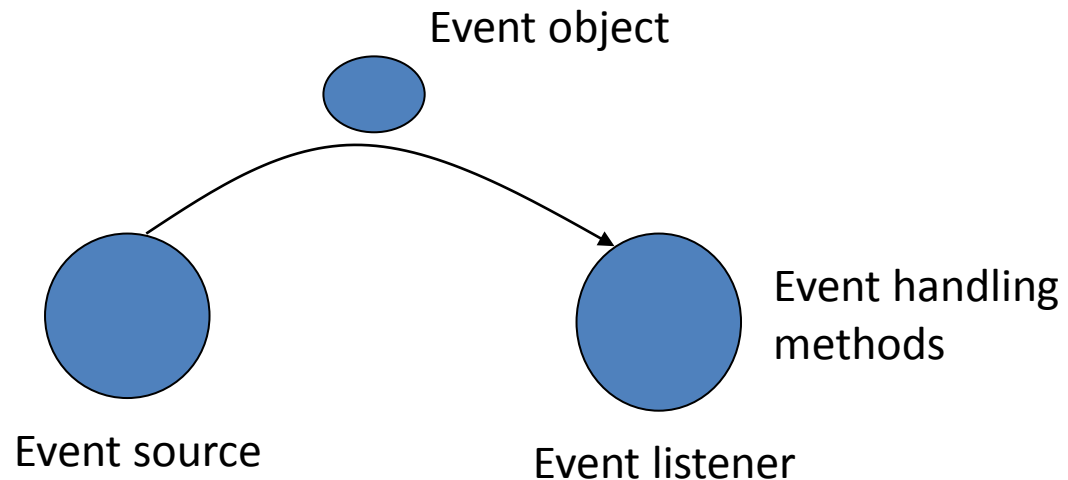
```
Choice colors = new Choice();  
colors.addItem("Red");  
color.addItem("Blue");
```



LẬP TRÌNH GIAO DIỆN

- **Xử lý sự kiện (Event Handling Model)**

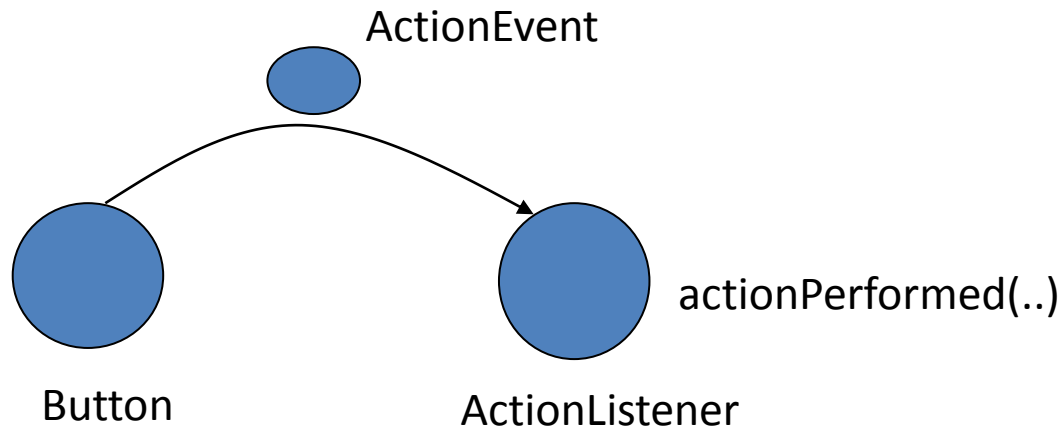
- Java hỗ trợ một số lớp sự kiện hỗ trợ người dùng tương tác với các thành phần GUI.
- Mỗi thành phần, ví dụ button, có thể sinh ra một sự kiện tương ứng. Các thành phần này là nguồn sự kiện.
- Lớp `EventListener` có nhiệm vụ đợi các sự kiện xảy ra. Khi sự kiện xảy ra, lớp `EventListener` nhiệm vụ thực hiện các phương thức xử lý sự kiện của lớp đó.



LẬP TRÌNH GIAO DIỆN

- **Ví dụ một số sự kiện**

- Khi người dùng click vào button “Cancel”, đối tượng `ActionEvent` được tạo ra, `ActionListener` sẽ bắt sự kiện này và thực hiện một phương thức `actionPerformed()`.
- Khi người dùng đóng cửa sổ, đối tượng `WindowEvent` được tạo ra, `WindowListener` sẽ lắng nghe và bắt sự kiện này.



LẬP TRÌNH GIAO DIỆN

- **Xử lý sự kiện (Event Handling Model)**
 - **Các lớp sự kiện của AWT**

| Sự kiện | Listener Interface | Mô tả |
|-----------------|--------------------------------------|--|
| ActionEvent | ActionListener | Sự kiện xảy ra khi một nút được click, một danh sách các item được double click hoặc một menu được lựa chọn. |
| AdjustmentEvent | AdjustmentListener | Sự kiện xảy ra khi scrollbar được sử dụng. |
| ComponentEvent | ComponentListener | Sự kiện xảy ra khi component được thay đổi kích thước, di chuyển, ẩn hoặc hiện. |
| FocusEvent | FocusListener | Sự kiện xảy ra khi component |
| ItemEvent | ItemListener | Xảy ra khi menu item được chọn, giải phóng hoặc khi checkbox hoặc list item được click |
| WindowEvent | WindowListener | Xảy ra khi cửa sổ được activated, closed, opened hoặc quit. |
| TextEvent | TextListener | Xảy ra khi giá trị của text field hoặc text area thay đổi |
| MouseEvent | MouseListener MouseMotionListener | Xảy ra khi chuột được di chuyển, click, giữ hoặc giải phóng |
| KeyEvent | KeyListener | Xảy ra khi input được nhận từ keyboard |

LẬP TRÌNH GIAO DIỆN

- **Các bước xử lý sự kiện trong Java**

- Bước 1: Khai báo lớp xử lý sự kiện. Lớp này có nhiệm vụ implement lớp giao diện Listener

```
public class DemoClass implements ActionListener {  
}
```

- Bước 2: Gán một đối tượng xử lý sự kiện của lớp xử lý sự kiện được khai báo ở trên cho các components liên quan.

```
anyComponent.addActionListener(instanceOf DemoClass);
```

- Bước 3: Triển khai các phương thức xử lý sự kiện của lớp giao diện Listener

```
public void actionPerformed(ActionEvent e) {  
    ...//code that reacts to the action...  
}
```

LẬP TRÌNH GIAO DIỆN

- **Lớp giao diện ActionListener**

- Lớp này bao gồm 1 phương thức xử lý sự kiện duy nhất:

- `public void actionPerformed(ActionEvent e)`

- Ví dụ: Tạo một nút bấm để đóng frame

```
import java.awt.*;
import java.awt.event.*;

public class ButtonDemo extends Frame
implements ActionListener{
    Button bn=new Button("Exit");
    ButtonDemo(String str){
        super(str);
        setLayout(new FlowLayout());
        this.add(bn);
        bn.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e)
    {
        System.exit(0);
    }

    public static void main(String args[]){
        ButtonDemo fr = new ButtonDemo("Frame test");
        fr.setSize(300,300);
        fr.setVisible(true);
    }
}
```

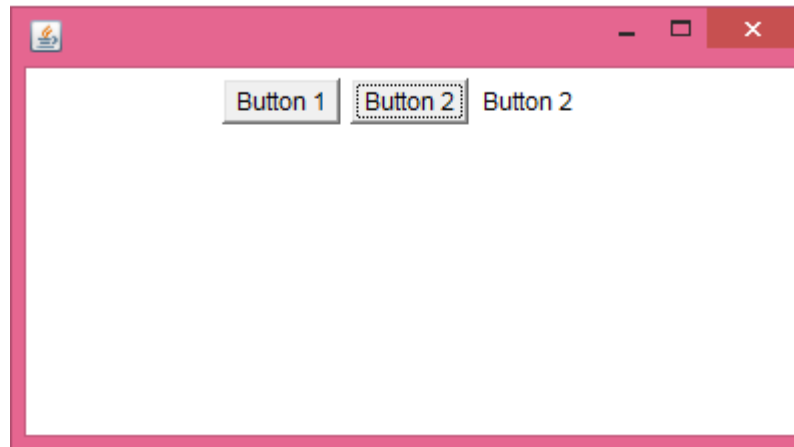
LẬP TRÌNH GIAO DIỆN

- **Ví dụ: Xử lý sự kiện ActionListener**

```
public class SimpleExample extends Frame implements ActionListener {  
}
```

```
Button bt1 = new Button("Button 1");  
this.add(bt1);  
bt1.addActionListener(this);
```

```
public void actionPerformed(ActionEvent e){  
}
```



LẬP TRÌNH GIAO DIỆN

- **Ví dụ: Xử lý sự kiện WindowEvent**

S.N. Method & Description

1 **void windowActivated(WindowEvent e)**

Invoked when the Window is set to be the active Window.

void windowClosed(WindowEvent e)

2 Invoked when a window has been closed as the result of calling dispose on the window.

void windowClosing(WindowEvent e)

3 Invoked when the user attempts to close the window from the window's system menu.

void windowDeactivated(WindowEvent e)

4 Invoked when a Window is no longer the active Window.

void windowDeiconified(WindowEvent e)

5 Invoked when a window is changed from a minimized to a normal state.

void windowIconified(WindowEvent e)

6 Invoked when a window is changed from a normal to a minimized state.

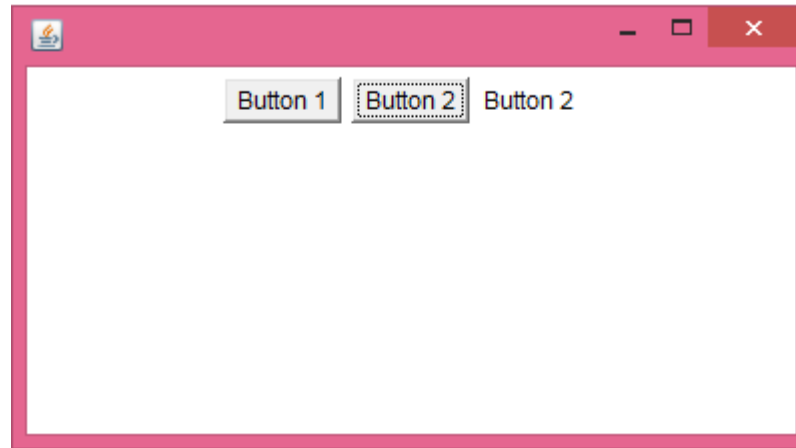
void windowOpened(WindowEvent e)

7

LẬP TRÌNH GIAO DIỆN

- **Ví dụ: Xử lý sự kiện WindowEvent**

```
public class SimpleExample extends Frame implements ActionListener, WindowListener{  
}  
@Override  
public void windowClosing(WindowEvent e) {  
    System.exit(0);  
}
```



LẬP TRÌNH GIAO DIỆN

- **Lớp giao diện MouseListener**

- Phương thức xử lý mouse click:

- ```
public void mouseClicked(MouseEvent e)
```

- Phương thức xử lý mouse enter:

- ```
public void mouseEntered(MouseEvent e)
```

- Phương thức xử lý khi mouse exit a component:

- ```
public void mouseExited(MouseEvent e)
```

# LẬP TRÌNH GIAO DIỆN

---

- **Lớp giao diện MouseListener**

- Phương thức xử lý mouse press:

- ```
public void mousePressed(MouseEvent e)
```

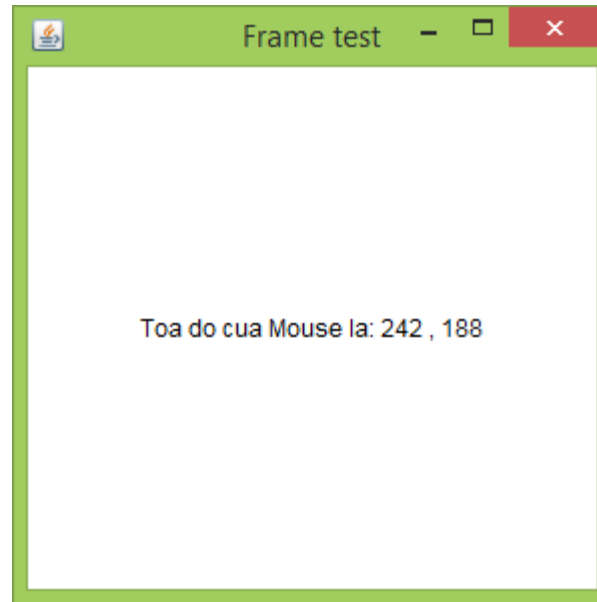
- Phương thức xử lý mouse release:

- ```
public void mouseReleased(MouseEvent e)
```

# LẬP TRÌNH GIAO DIỆN

---

- Ví dụ:





# LẬP TRÌNH GIAO DIỆN

---

- **Lớp giao diện ItemListener**

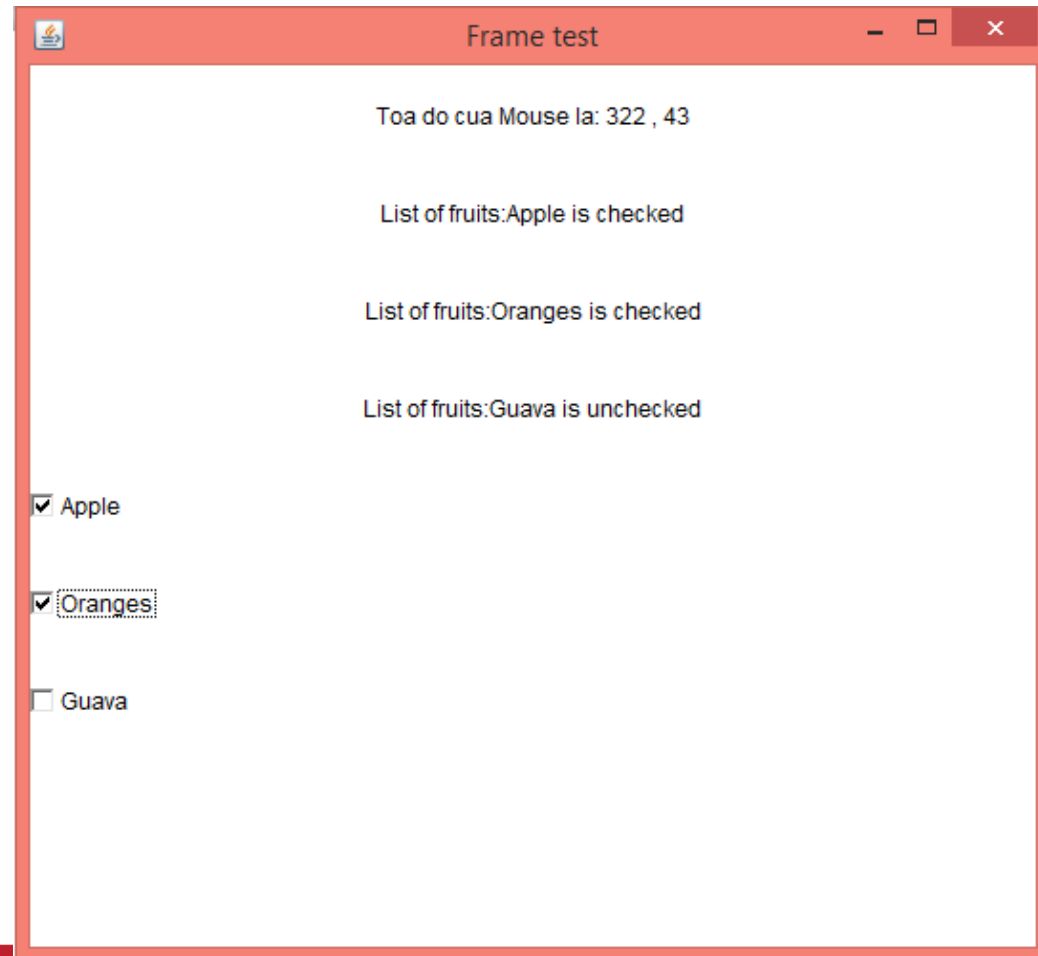
| S.N. | Method & Description |
|------|----------------------|
|------|----------------------|

|   |                                                                                                                |
|---|----------------------------------------------------------------------------------------------------------------|
| 1 | <b>void itemStateChanged(ItemEvent e)</b><br>Invoked when an item has been selected or deselected by the user. |
|---|----------------------------------------------------------------------------------------------------------------|

# LẬP TRÌNH GIAO DIỆN

---

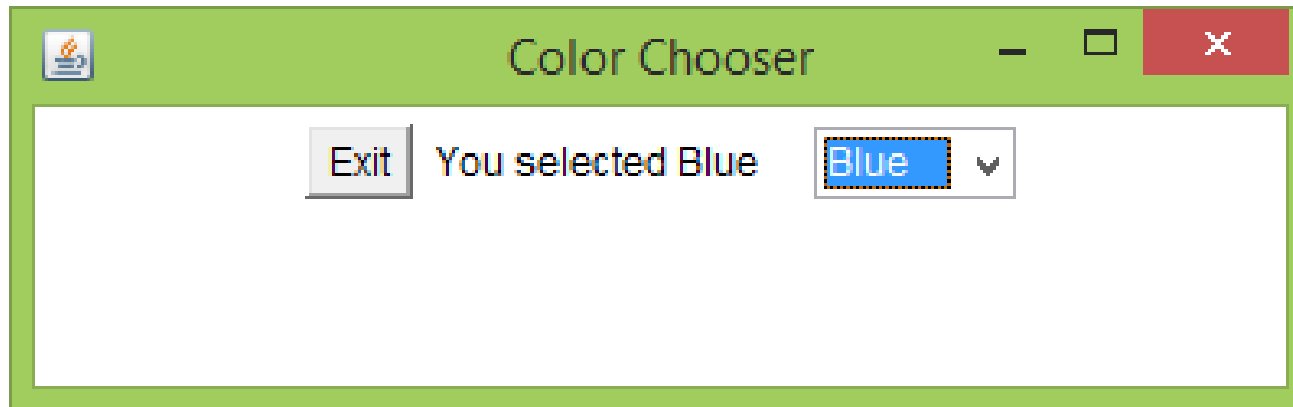
- Lớp giao diện ItemListener: Ví dụ



# LẬP TRÌNH GIAO DIỆN

---

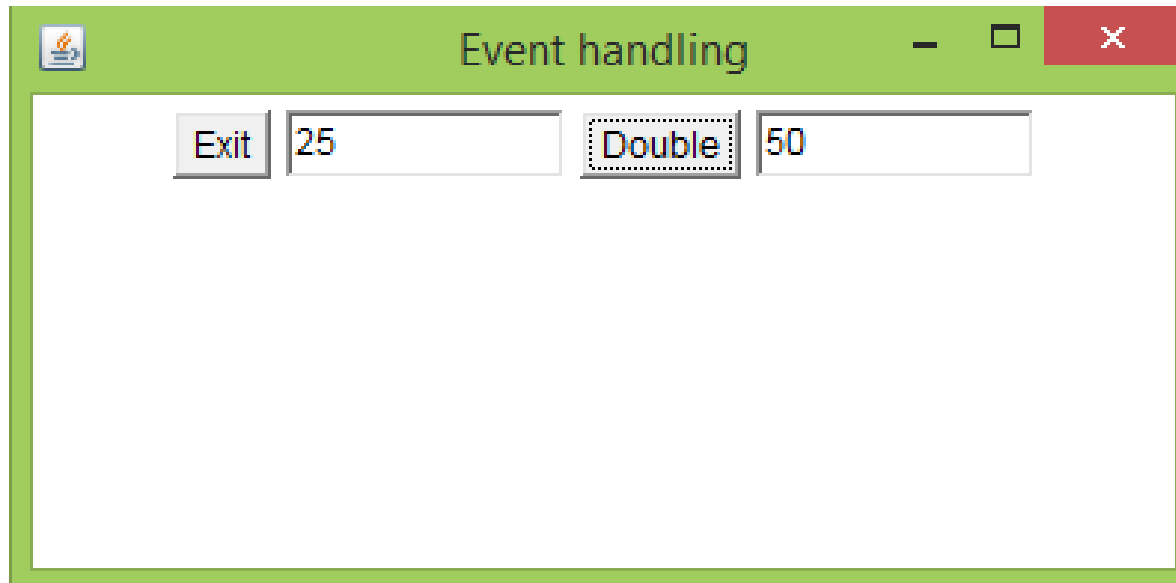
- **Bài tập 2: Tạo giao diện như hình vẽ:**



# LẬP TRÌNH GIAO DIỆN

---

- **Bài tập 1: Tạo giao diện như hình vẽ:**



# LẬP TRÌNH GIAO DIỆN

---

- **Bài tập 2: Tạo giao diện như hình vẽ:**

