



[cep.edu.vn](http://cep.edu.vn)

# TRƯỜNG CAO ĐẲNG KINH TẾ - KẾ HOẠCH ĐÀ NẴNG

## KHOA TIN HỌC & NGOẠI NGỮ

# IFF

## SLIDE BÀI GIẢNG

## LẬP TRÌNH JAVA 2

THS. NGUYỄN ĐÌNH THÀ

*Đà Nẵng – 2022*

# CHƯƠNG 2. SWING

# MỤC TIÊU

Kết thúc bài học này, sinh viên có khả năng:

- Thiết kế được giao diện đồ họa
- Xử lý được sự kiện click chuột
- Đọc dữ liệu từ giao diện người dùng
- Hiển thị được dữ liệu lên giao diện người dùng

# Giới thiệu giao diện đồ họa - GUI

Ứng dụng giao diện đồ họa là ứng dụng sử dụng giao diện đồ họa để giao tiếp với người dùng:

QUẢN LÝ NHÂN VIÊN

10:22 AM

MÃ NHÂN VIÊN

HỌ VÀ TÊN

TUỔI

EMAIL

LƯƠNG

New

Save

Delete

Find

Open

Exit

<| << >> >| Record: 1 of 10

MÃ	HỌ VÀ TÊN	TUỔI	EMAIL	LƯƠNG
----	-----------	------	-------	-------

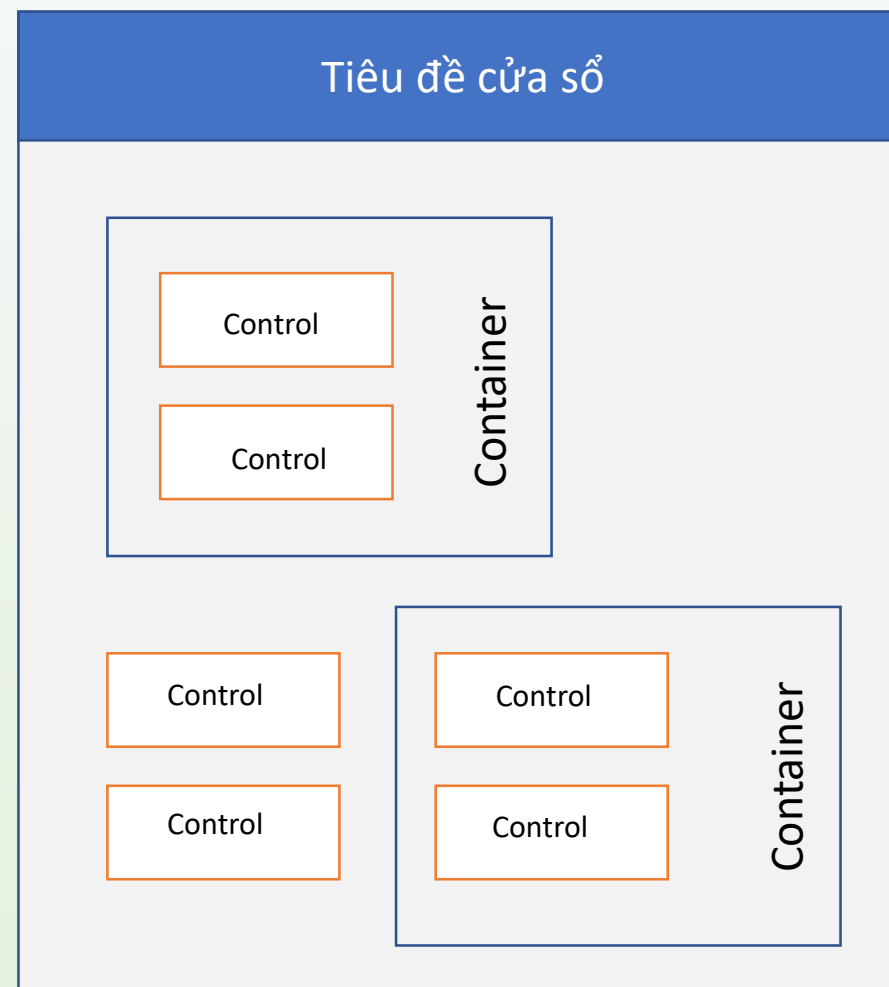
# Các thành phần cơ bản của GUI

- Cửa sổ - Window: Cửa sổ ứng dụng.
- Khung chứa – Container: Chứa một hoặc nhiều điều khiển
- Điều khiển – Control: TextField, Radio, CheckBox, Button, Table...

## Chú ý:

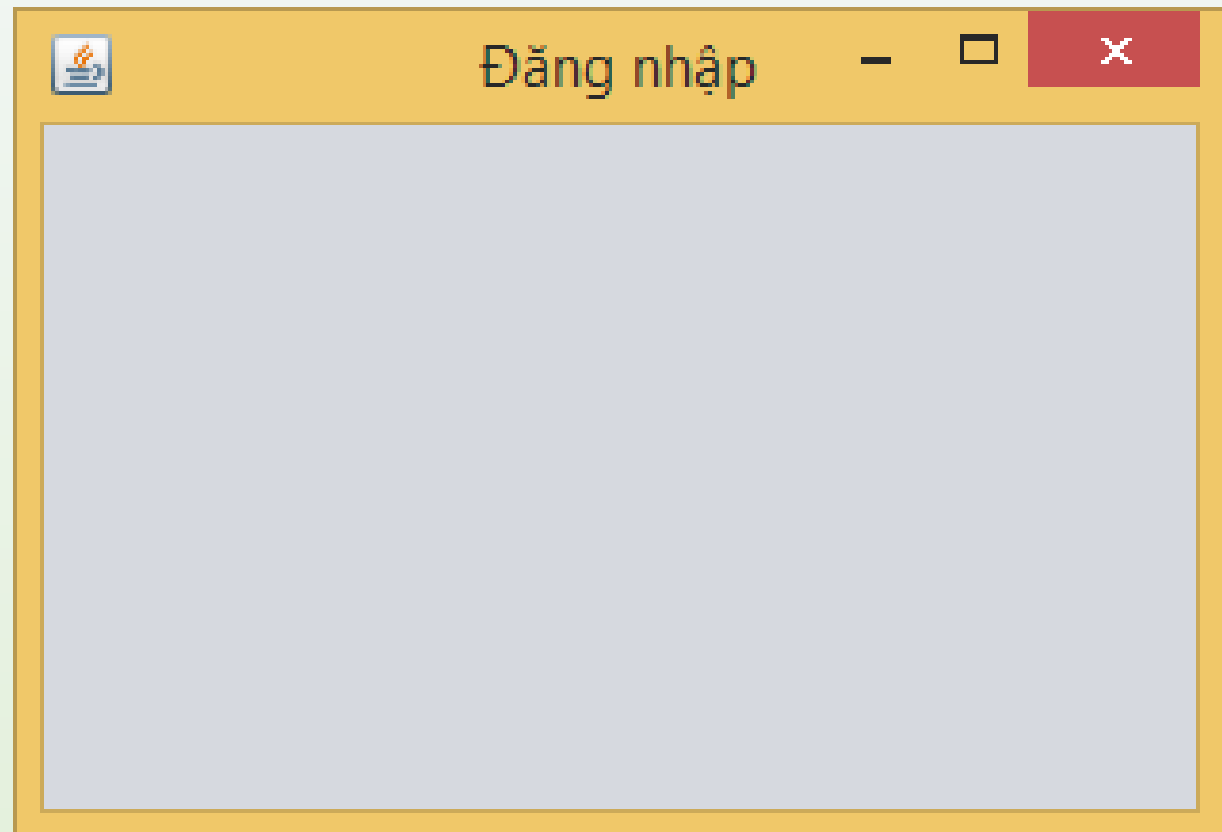
Một ứng dụng có thể có nhiều cửa sổ.

Một khung chứa có thể chứa nhiều điều khiển hoặc khung chứa khác



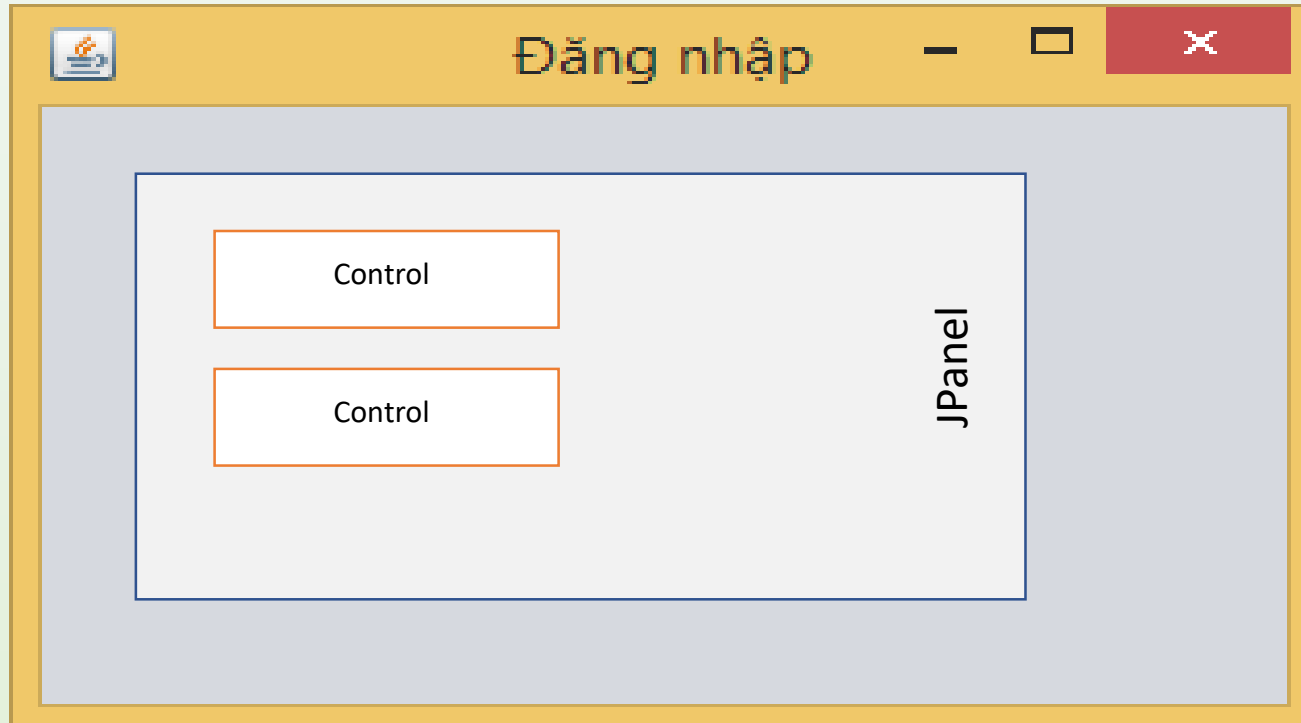
# Window

- Trong Swing, JFrame được sử dụng để tạo cửa sổ
- Các thành phần liên quan:
  - ✓ Biểu tượng - Icon
  - ✓ Tiêu đề - Title
  - ✓ Thu, phóng, tắt cửa sổ



# Container

- Trong Swing có nhiều loại Container. Trong bài này chúng ta sẽ tìm hiểu JPanel
- JPanel là container được sử dụng để nhóm các control hoặc container khác.



# Control

---

- Control là các phần tử giao diện được sử dụng để tiếp nhận dữ liệu từ người dùng hoặc trình bày dữ liệu đến người dùng.
- Các Control thường dùng:
  - ✓ JLabel - Nhãn
  - ✓ JTextField – Ô nhập
  - ✓ JButton – Nút nhấn
  - ✓ JCheckBox – Hộp kiểm
  - ✓ JRadioButton – Lựa chọn
  - ✓ JComboBox – Hộp chọn
  - ✓ JTable – Bảng
  - ✓ JOptionPane – Hộp thoại
  - ✓ ...

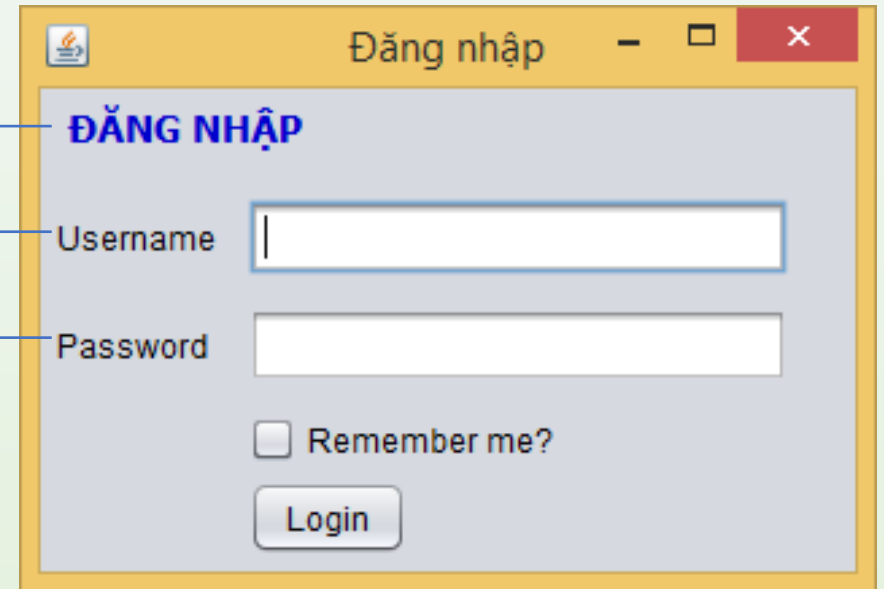


# JLabel

- JLabel là điều khiển được sử dụng để tạo nhãn.
- Thuộc tính quan trọng nhất của JLabel là Text
- Khi đặt tên bạn cần bắt đầu bởi *lbl*

➤ Ví dụ:

lblTitle  
lblUsername  
lblPassword

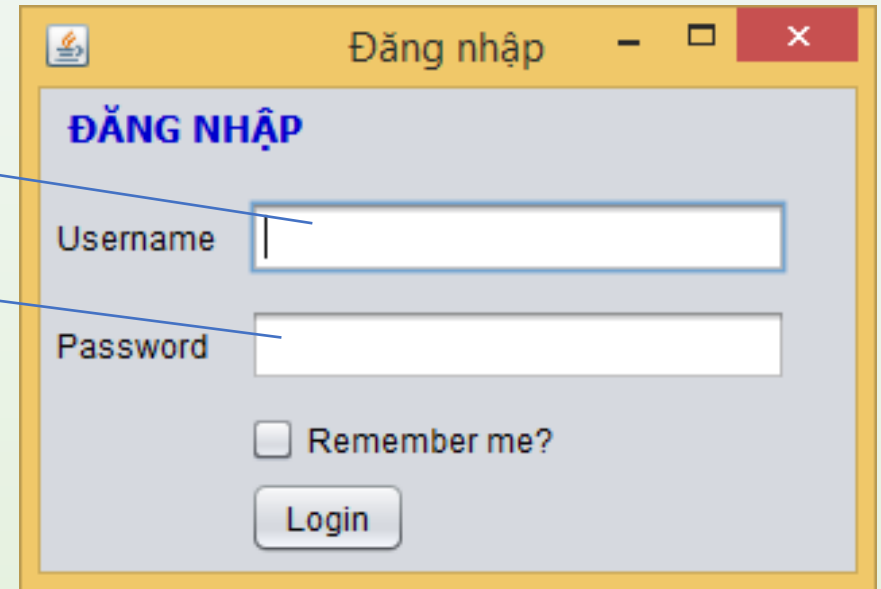


# JTextField, JTextArea và JPasswordField

- JTextField được sử dụng để tạo ô nhập. 2 thuộc tính thường dùng là text và name:
  - ✓ Text: giá trị của ô nhập
  - ✓ Name: tên của ô nhập, nên bắt đầu bởi *txt*

- Ví dụ:

txtUsername  
txtPassword

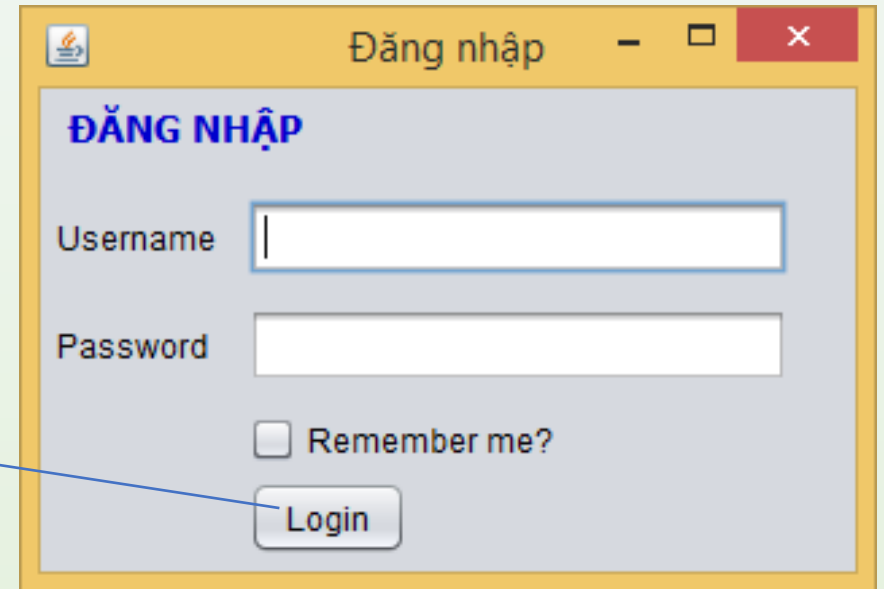


- JTextArea chỉ khác JTextField là cho phép chứa Text nhiều dòng
- JPasswordField thì không cho nhìn thấy nội dung của Text

# JButton

- JButton được sử dụng để tạo nút nhấn. Các thuộc tính thường sử dụng là text và name:
  - Text: nhãn của nút
  - Name: tên của nút, nên bắt đầu bởi *btn*
- Ví dụ:

btnLogin



# JCheckBox

- JCheckBox được sử dụng để tạo hộp kiểm. Các thuộc tính thường dùng là text, name và selected:
  - ✓ Text: Nhãn đính kèm
  - ✓ Selected: Trạng thái
  - ✓ Name: Tên, nên bắt đầu bởi *chk*
- Ví dụ:

chkRemember

The screenshot shows a Java Swing window titled "Đăng nhập" (Login). The window has a yellow title bar with standard OS controls. The main content area has a light blue background. At the top, there's a header with the text "ĐĂNG NHẬP" in blue. Below the header, there are three input fields: "Username", "Password", and a checkbox labeled "Remember me?". The "Remember me?" checkbox is currently unchecked. A blue arrow points from the text "chkRemember" to the checkbox, indicating that the checkbox's name attribute is set to "chkRemember". At the bottom of the window, there is a "Login" button.

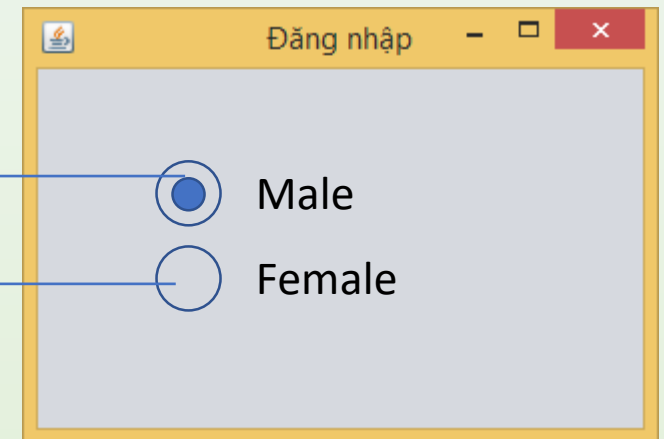
# JRadioButton, ButtonGroup

- JRadioButton được sử dụng để tạo các mục chọn. Các thuộc tính thường dùng là text, name và selected:
  - ✓ Text: Nhãn đính kèm
  - ✓ Selected: Trạng thái
  - ✓ Name: Tên, nên bắt đầu bởi *rdo*
- ButtonGroup: nhóm mà radio thuộc vào. Trong mỗi nhóm bạn chỉ có thể được chọn một radio.

- Ví dụ:

rdoMale

rdoFemale



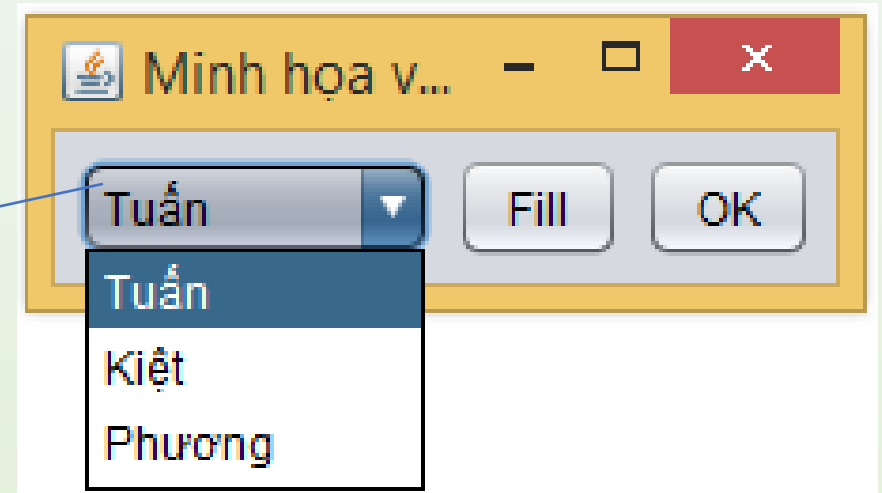
# JComboBox

- JComboBox được sử dụng để tạo danh sách xổ xuống và chỉ cho phép chọn 1 giá trị trong danh sách. Các thuộc tính thường sử dụng là `model`, `selectedIndex`, `selectedItem` và `name`:
  - ✓ `Model`: chứa danh sách dữ liệu
  - ✓ `SelectedIndex`: vị trí mục được chọn
  - ✓ `SelectedItem`: dữ liệu mục chọn
  - ✓ `Name`: Tên, nên bắt đầu bởi *cbo*

- Ví dụ:

```
void initPhongBan() {  
    String [] dsSV = {"Tuấn", "Kiệt", "Phương"};  
    cboNames.setModel(new DefaultComboBoxModel(dsSV));  
}
```

cboNames

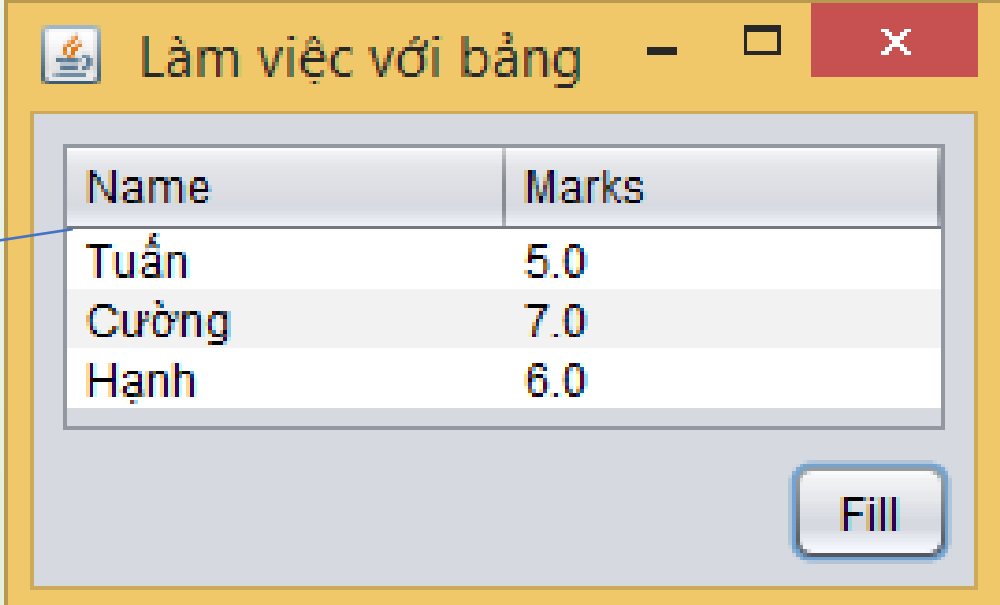


*Có thể gán giá trị cho cbo bởi code và gọi hàm này ngay sau hàm `initComponents()`;*

# Jtable, DefaultTableModel

- JTable được sử dụng để biểu diễn dữ liệu dạng bảng. Các thuộc tính thường dùng là model, selectionMode, rowHeight và name:
  - ✓ Model: dữ liệu bảng
  - ✓ SelectionMode: chế độ chọn các hàng
  - ✓ RowHeight: chiều cao mỗi hàng
  - ✓ Name: Tên, nên bắt đầu bởi *tbl*
- Ví dụ:

tblStudents

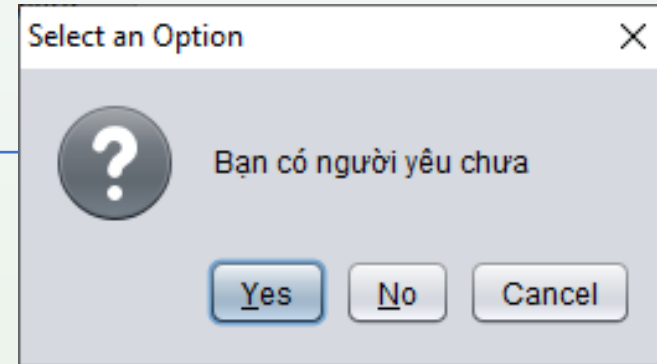


Name	Marks
Tuấn	5.0
Cường	7.0
Hạnh	6.0

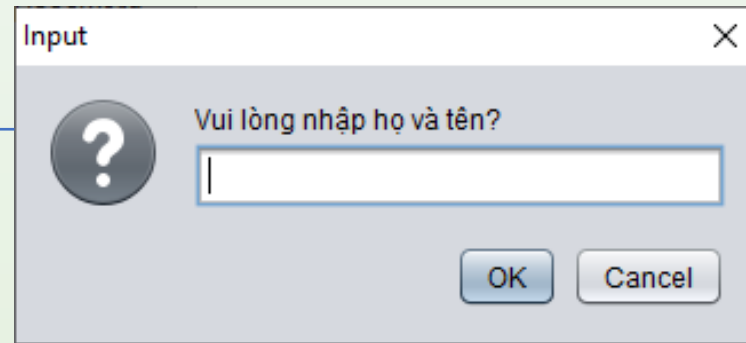
# JOptionPane

- JOptionPane là một dạng cửa sổ được sử dụng để tạo các hộp thoại thông báo. Có 3 dạng thông báo thường dùng là:

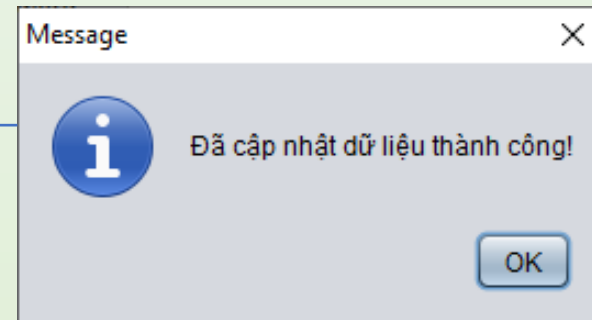
✓ Xác nhận – Confirm



✓ Nhập – Input



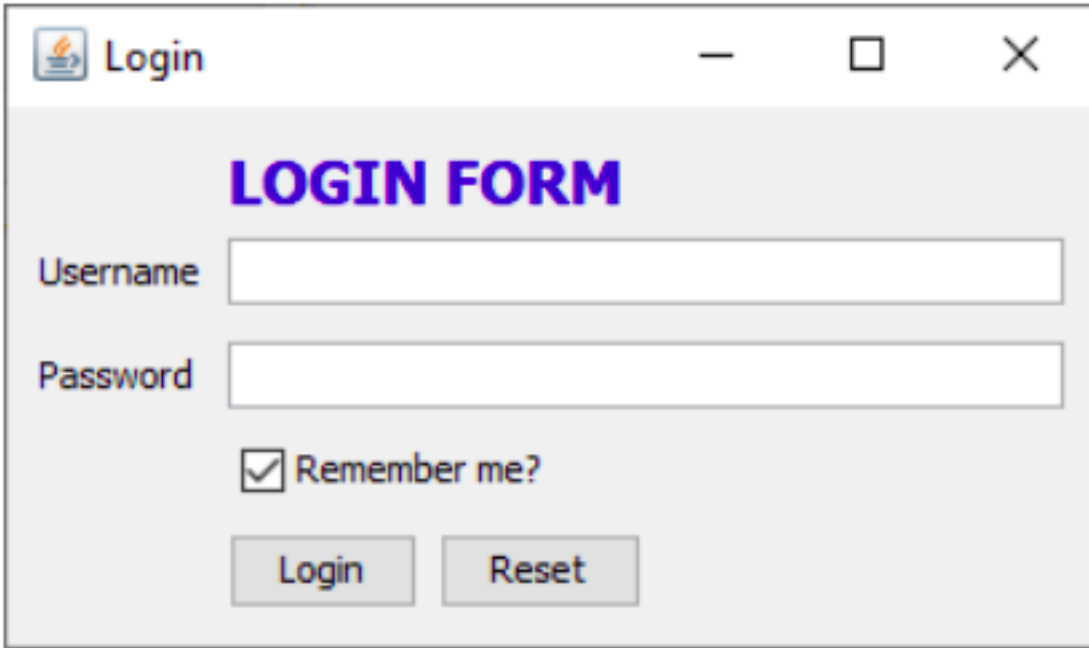
✓ Thông báo - Message





# Demo1

Thiết kế giao diện như hình sau:




The image shows a web browser window with the title "Login". Inside the window, there is a form titled "LOGIN FORM" in purple text. The form contains two input fields: "Username" and "Password". Below the "Password" field, there is a checkbox labeled "Remember me?" which is checked. At the bottom of the form, there are two buttons: "Login" and "Reset".

# Demo2

Thiết kế giao diện như hình sau:

Control	Name	Description
TextField	txtId	Staff Id
TextField	txtFullname	Fullname
ComboBox	cboCountry	Country
RadioButton	rdoMale	Gender
RadioButton	rdoFemale	
ButtonGroup	grpGender	
CheckBox	chkStatus	Single?
TextArea	txtNotes	Notes
Table	tblStaffList	Staff List

 Staff Edition

## Staff Edition

Staff Id

Fullname

Country

Việt Nam

Gender

☒ Male

☐ Female

Status

☒ Single?

Notes

Create

Update

Delete

Reset

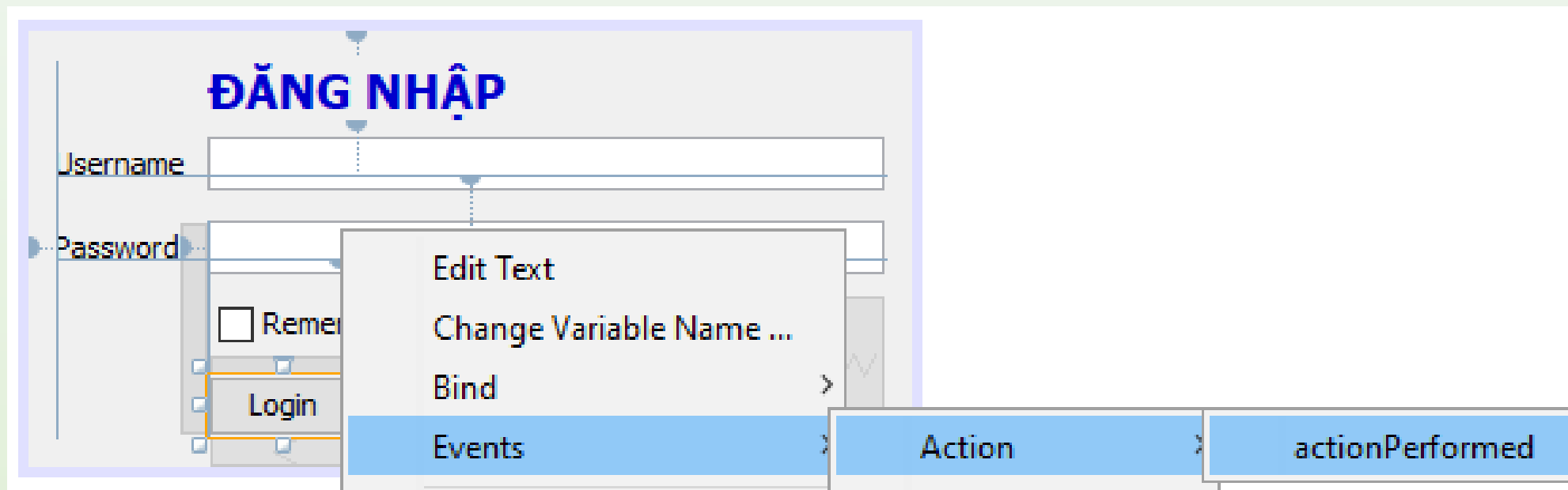
# Lập trình giao diện

---

- Lập trình giao diện là công việc lập trình để:
  - ✓ Thu nhận dữ liệu từ người dùng
  - ✓ Hiển thị dữ liệu để người dùng xem
  - ✓ Thay đổi giao diện tùy thuộc vào dữ liệu
- Lập trình giao diện thường gắn kết với các sự kiện xảy ra trong quá trình sử dụng
  - ✓ Có nhiều loại sự kiện, trong bài học này chúng ta chỉ nghiên cứu sự kiện Action.

# Sự kiện Action

- Khi bạn click chuột vào nút thì sự kiện Action xảy ra.
- Bạn cần viết mã để điều khiển sự kiện này
  - ✓ Đọc dữ liệu từ các điều khiển
  - ✓ Xử lý theo yêu cầu nghiệp vụ
  - ✓ Hiển thị, thay đổi dữ liệu trên giao diện
- Cách tạo sự kiện: Nhấp đúp vào nút hoặc thực hiện theo hình sau:



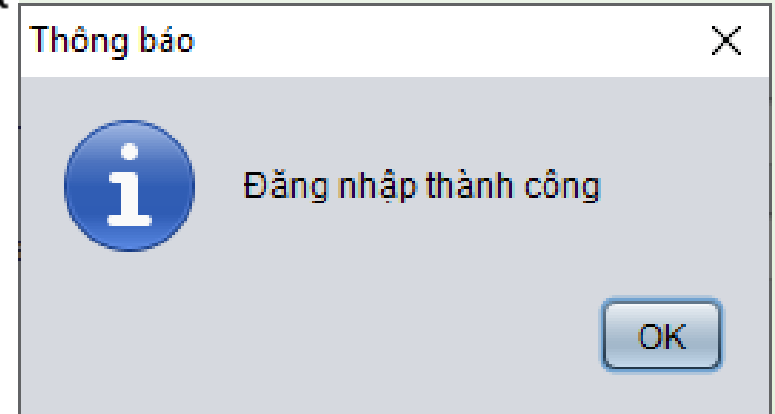
# Xử lý sự kiện

- Đặt code xử lý của bạn ngay sau dòng comment TODO để thực hiện công việc của bạn

```
private void btnLoginActionPerformed(ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

- Ví dụ: Nhập dòng code sau và khi click vào sẽ hiển thị hộp thoại như sau:

```
private void btnLoginActionPerformed(ActionEvent evt) {  
    // TODO add your handling code here:  
    JOptionPane.showMessageDialog(this,  
        "Đăng nhập thành công", "Thông báo",  
        JOptionPane.INFORMATION_MESSAGE);  
}
```



# Làm việc với các thuộc tính

---

- Để đọc/ghi giá trị thuộc tính **Xyz** của control bạn sử dụng:
  - ✓ Đọc (lấy dữ liệu từ control): **Type** value = control.get**Xyz**()
  - ✓ Ghi (thay đổi dữ liệu trên control): control.set**Xyz**(**Type** value)
- Ví dụ: code để đọc và ghi dữ liệu ở control textField có name như sau:
  - ✓ Đọc: String username = txtUsername.getText();
  - ✓ Ghi: txtUsername.setText("admin")

## Làm việc với các thuộc tính

---

- Để đọc giá trị của Username, Password và Remember:
  - ✓ `String username = txtUsername.getText();`
  - ✓ `String password = txtPassword.getText();`
  - ✓ `Boolean remember = chkRemember.isSelected();`
  
- Để ghi dữ liệu lên các điều khiển:
  - ✓ `txtUsername.setText("");`
  - ✓ `chkRemember.setSelected(false);`
  
- Chú ý: Khi kiểu thuộc tính là Boolean, thì theo quy ước của JavaBean chúng ta sử dụng `isXyz()` thay vì `getXyz()`

# Làm việc với JComboBox

---

- Với JComboBox, bạn cần lập trình thực hiện các công việc sau:
  - ✓ Đổ dữ liệu vào JComboBox
  - ✓ Làm việc với các mục
  - ✓ Làm việc với vị trí và giá trị của mục chọn được chọn



# Làm việc với JComboBox

- Đổ dữ liệu vào JComboBox có name là cboCountry, bạn sử dụng đoạn mã sau đây:

```
String[] data = {  
    "Việt Nam", "Lào", "Singapore", "Malaysia"  
};  
cboCountry.setModel(new DefaultComboBoxModel(data));
```

- Làm việc với mục chọn:

```
cboCountry.setSelectedIndex(1) // phần tử thứ 2 trong cboCountry  
int index = cboCountry.getSelectedIndex()  
String country = (String)cboCountry.getSelectedItem()
```

# Làm việc với JComboBox

---

- Các phương thức sau đây giúp bạn làm việc với các mục trong JComboBox:
  - ✓ `cboCountry.getItemCount()`
  - ✓ `cboCountry.getItemAt(index)`
  - ✓ `cboCountry.addItem(item)`
  - ✓ `cboCountry.removeItem(item)`
  - ✓ `cboCountry.removeItemAt(index)`
  - ✓ `cboCountry.removeAllItems()`

# Làm việc với JTable

---

- Với JTable, chúng ta cần lập trình thực hiện được các công việc sau:
  - ✓ Đổ dữ liệu vào bảng
  - ✓ Làm việc với các hàng được chọn
  - ✓ Làm việc với các hàng

# Làm việc với JTable

- Bạn có thể đổ dữ liệu vào bảng name là tblDSCauThu bằng đoạn mã sau:

```
void initTableData() {  
    Object[] columns = {  
        "Mã nhân viên", "Họ tên", "Email", "Giới tính", "Tình trạng"  
    };  
  
    Object[][] rows = {  
        {1, "Văn Toàn", "vantoan@gmail.com", "Nam", "Độc thân"},  
        {2, "Huỳnh Như", "huynhnhu@yahoo.com", "Nữ", "Độc thân"}  
    };  
  
    tblStaffs.setModel(new DefaultTableModel(rows, columns));  
}
```

Tên các cột của bảng

Các dòng bảng ghi

# Làm việc với JTable

---

- Lập trình lấy vị trí các hàng được chọn:

```
// Vị trí hàng được chọn trong bảng
```

```
int index = tblDSCauThu.getSelectedRow();
```

```
// Số hàng được chọn
```

```
int count = tblDSCauThu.getSelectedRowCount();
```

```
// Vị trí các hàng được chọn
```

```
int[] indices = tblDSCauThu.getSelectedRows();
```

# Làm việc với JTable

---

- Lấy Model:

```
DefaultTableModel model = (DefaultTableModel) tblStaffList.getModel();
```

- Làm việc với các hàng:

```
model.addRow(rowData);  
model.insertRow(rowIndex, rowData);  
model.removeRow(rowIndex);  
model.setRowCount(rowCount);
```

- Làm việc với dữ liệu trong ô:

```
model.getValueAt(rowIndex, columnIndex);  
model.setValueAt(value, rowIndex, columnIndex);
```

## Demo3

➤ Tạo giao diện và code cho bảng danh sách nhân viên như sau:

QUẢN LÝ NHÂN VIÊN

Mã nhân viên:

Họ tên:

Email:

Giới tính:

☒ Nam ☐ Nữ

Tình trạng:

☒ Độc thân ☐ Đã kết hôn

Mã nhân viên	Họ tên	Email	Giới tính	Tình trạng
1	Văn Toàn	vantoan@gmail.com	Nam	Độc thân
2	Huỳnh Như	huynhnhu@yahoo.com	Nữ	Độc thân
3	Công Phượng	congphuong@yahoo.com	Nam	Đã kết hôn

Xoá

Cập nhật

Thêm mới

Thoát

- Xoá: xoá dòng thông tin được chọn
- Cập nhật: Cập nhật lại dữ liệu cho dòng thông tin được chọn từ các control ở bên trên
- Thêm mới: Thêm một dòng thông tin (check full các thông tin trước khi thêm)
- Thoát: Exit

# Bài tập

---