



## Giới thiệu ứng dụng giao diện đồ họa - GUI

**Java Swing**

**Thiết kế giao diện đồ họa**

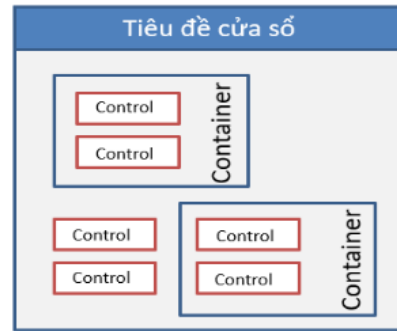
**Lập trình giao diện đồ họa**

code



**GUI ?**

Ứng dụng dùng giao diện đồ họa để giao tiếp với người dùng



**Cửa sổ**

**Các thành phần của giao diện**

**Khung chứa chứa các điều khiển**

**Điều khiển : button, text box, combo box, v.v.**

**GUI - Graphical User Interface**

**Chú ý**

**1 ứng dụng có thể có nhiều cửa sổ**

**1 khung chứa có thể chứa n điều khiển | khung chứa #**

**Ví dụ**





# Thiết kế giao diện đồ họa

JFrame

Cửa sổ

Control

Các phần tử giao diện được sử dụng để nhập | trình bày dữ liệu

Container

Khung chứa

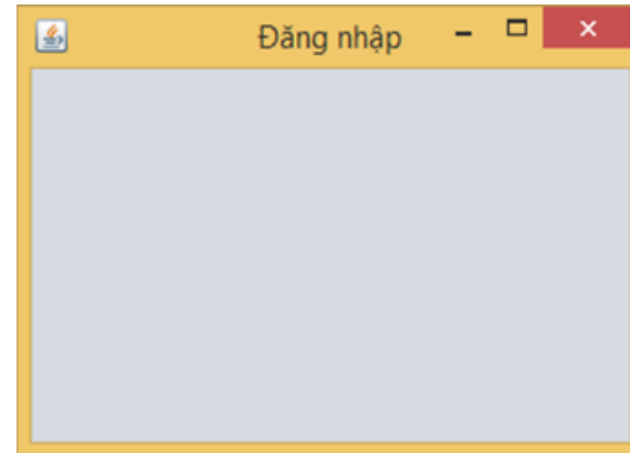
JPanel





## JFrame - Cửa sổ

Dùng tạo cửa sổ



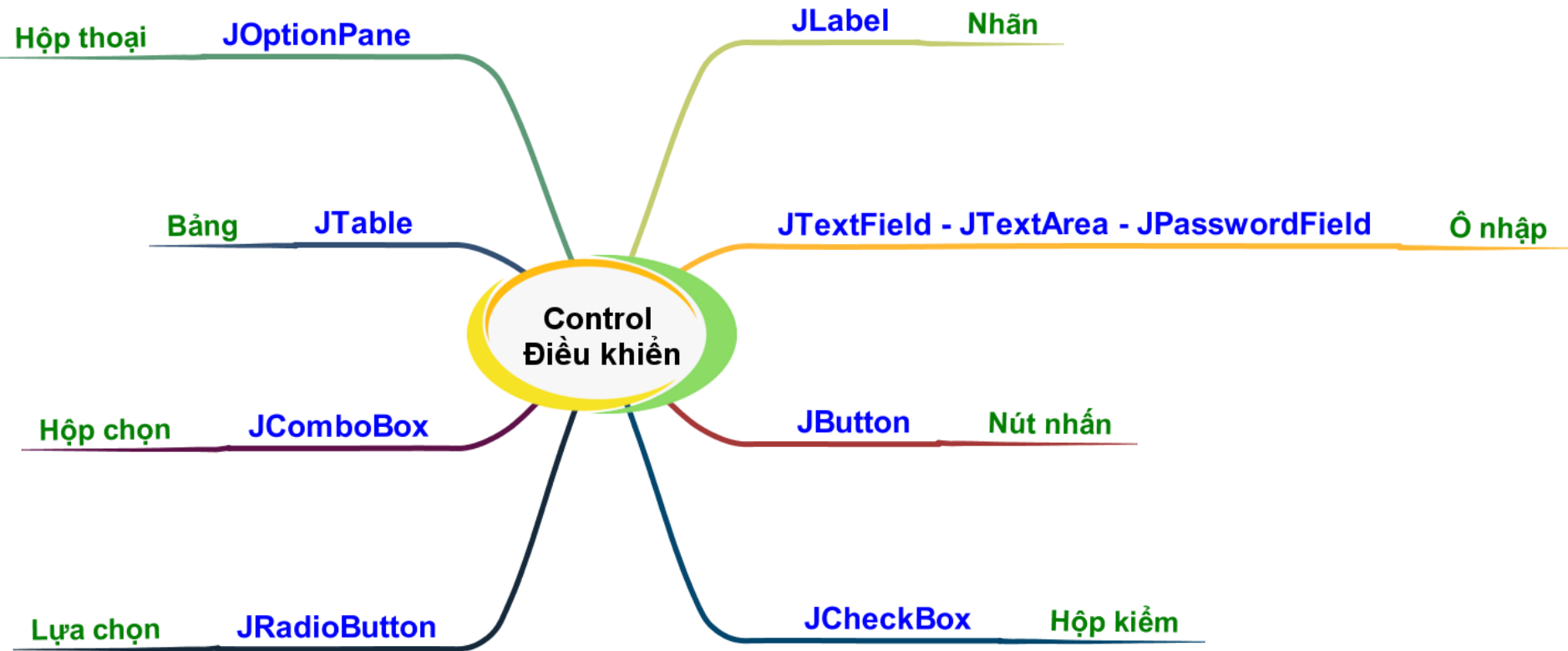
Các thành phần của cửa sổ

Biểu tượng - Icon

Tiêu đề - Title

Thu, phóng, tắt cửa sổ







# JLabel - Nhãn

Dùng tạo nhãn

Ví dụ

- lblTitle
- lblUsername
- lblPassword

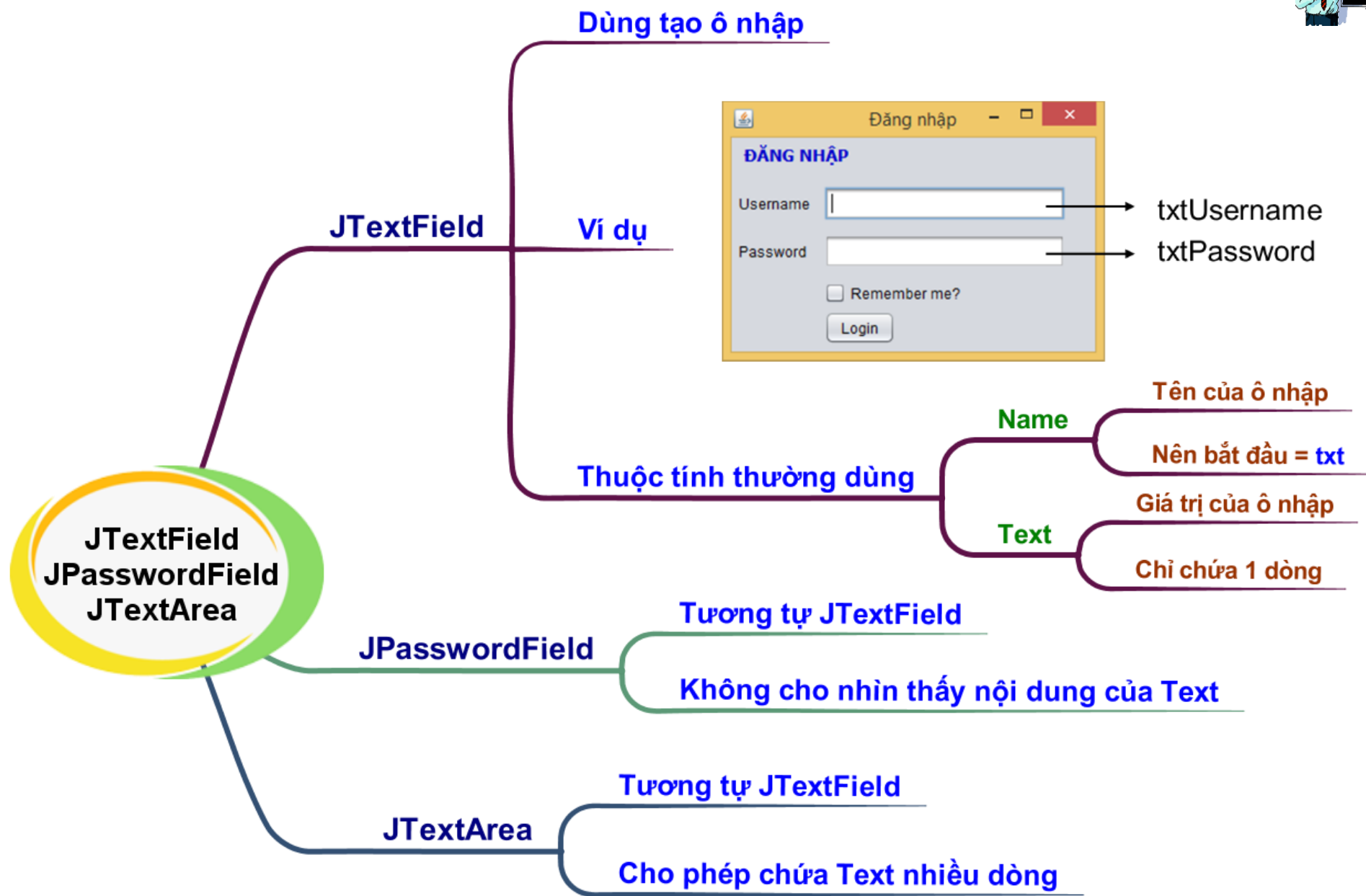
Các thuộc tính thường dùng

Name

Nên bắt đầu Lbl

Text | Icon







Dùng tạo nút nhấn

Ví dụ

The screenshot shows a Java Swing window titled "Đăng nhập" (Login). It contains a title bar with standard window controls. The main area has a label "ĐĂNG NHẬP" in blue. Below it are two text input fields: "Username" and "Password". There is a checkbox labeled "Remember me?". At the bottom right is a button labeled "Login". An arrow points from the text "btnLogin" to this button.

btnLogin

Các thuộc tính thường dùng

**Text**

Tên nhãn của nút

**Name**

Tên của nút

Nên bắt đầu = **btn**







## JCheckBox - Hộp kiểm

Dùng để tạo hộp kiểm

Ví dụ

chkRemember

Đăng nhập

**ĐĂNG NHẬP**

Username

Password

☐ Remember me?

Login

Các thuộc tính thường dùng

Name

Tên của hộp kiểm

Nên bắt đầu = **chk**

Text

Nhãn đính kèm

Selected

Trạng thái





## JCheckBox - Hộp kiểm

Dùng để tạo hộp kiểm

Ví dụ

chkRemember

Đăng nhập

**ĐĂNG NHẬP**

Username

Password

☐ Remember me?

Login

Các thuộc tính thường dùng

Name

Tên của hộp kiểm

Nên bắt đầu = **chk**

Text

Nhãn đính kèm

Selected

Trạng thái





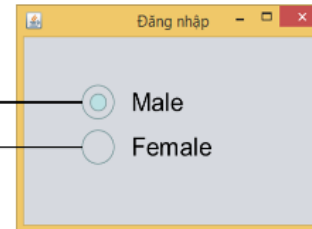
# JRadioButton ButtonGroup

## JRadioButton

Được sử dụng để tạo các mục chọn

Ví dụ

rdoMale  
rdoFemale



Các thuộc tính thường dùng

Name

Tên của mục chọn

Nên bắt đầu = rdo

Text

## ButtonGroup

Dùng để tạo nhóm cho các radio

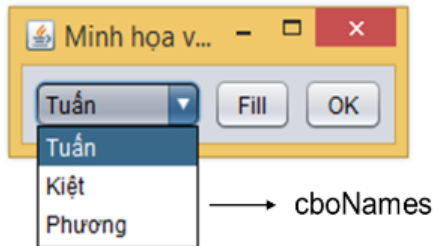
Trong mỗi nhóm chỉ 1 radio được chọn





Được sử dụng để tạo danh sách  
xổ xuống và chỉ cho phép chọn 1

Ví dụ



## JComboBox - Hộp chọn

Các thuộc tính thường dùng

**Name**

Tên của hộp chọn

Nên bắt đầu = cbo

**Model**

Chứa danh sách dữ liệu

**SelectedIndex**

Vị trí mục được chọn

**SelectedItem**

Dữ liệu mục chọn

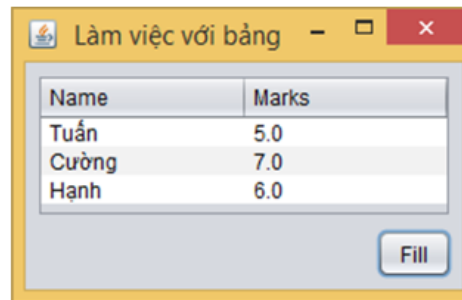




Dùng để hiển thị dữ liệu dưới dạng bảng

## JTable Bảng

Ví dụ



Name	Marks
Tuấn	5.0
Cường	7.0
Hạnh	6.0

Các thuộc tính thường dùng

Name

Tên của bảng

Nên bắt đầu = **tbl**

Model

Dữ liệu bảng

SelectionMode

Chế độ chọn các hàng

RowHeight

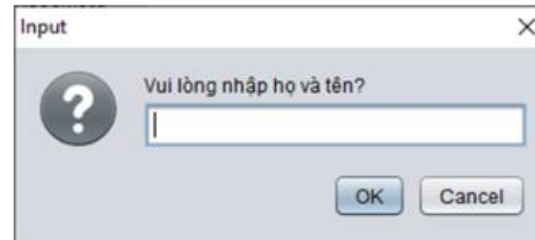
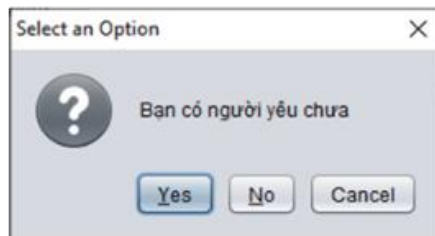
Chiều cao mỗi hàng





Là một dạng cửa sổ được sử dụng để tạo các hộp thoại thông báo

## JOptionPane Hộp thoại



Các dạng thông báo thường dùng

Xác nhận – Confirm

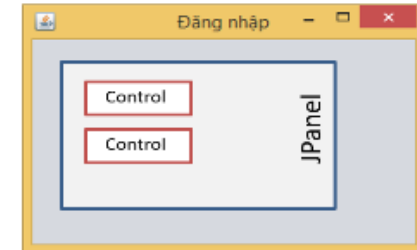
Nhập – Input

Thông báo - Message





Dùng để nhóm các control | container



JPanel  
Khung chứa

Thuộc tính thường dùng

border

Đường kẻ viền

Tabbed panel

Human Resource Management

Staff Management

EDITION: LIST

Staff Id:

Fullname:

Country:

Gender: ☒ Male ☐ Female

Status: ☒ Single?

Notes:

Add Remove Update Reset





Dùng để tổ chức giao diện

Tabbed Panel

Các thuộc tính thường dùng

Name

Tên của Tab

Thường bắt đầu = tab

Tab title

Tiêu đề

border

Đường kẻ viền

tabPlacement

Vị trí đặt tab

selectedIndex

Chọn tab







# Lập trình giao diện

Là công việc lập trình với mục đích

Thu nhận dữ liệu từ người dùng

Hiển thị dữ liệu để người dùng xem

Thay đổi giao diện tùy thuộc vào dữ liệu

event (sự kiện)

Là làm thay đổi trạng thái của một đối tượng khi bị tác động (nhấn chuột, kéo chuột...)

Các sự kiện

ActionPerformed

MouseClicked

WindowOpen





Đăng nhập

**ĐĂNG NHẬP**

Username

Password

☐ Remember me?

Login

Ví dụ

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

Khi click chuột vào **Login** thì sự kiện **ActionPerformed** xảy ra

ActionPerformed

Làm việc (viết code) với các điều khiển

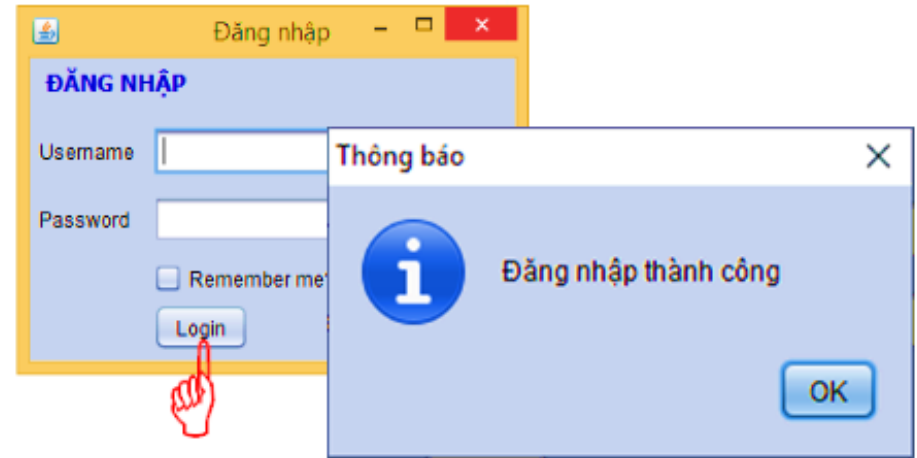
JOptionPane

Các thuộc tính **XYZ**

JComboBox

JTable





**Yêu cầu**

**JOptionPane**

**Viết code**

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





Đọc ( lấy DL từ control )

TT có kiểu Boolean

Boolean value = control.isXyz()

TT có kiểu # boolean

Type value = control.getXyz()

Ghi (thay đổi DL trên control )

control.setXyz(value)

Các control có thuộc tính Xyz

Ví dụ

Đọc

txtUsername.getText()

txtPassowd.getText()

chkRemember.isSelected()

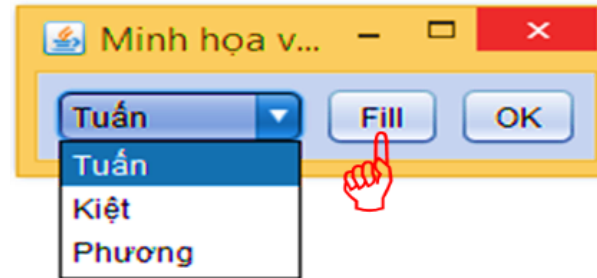
Ghi

txtUsername.setText("")

txtPassword.setText("")

chkRemember.setSelected( false)





# JComboBox

## Dữ liệu

Mảng chuỗi 1 chiều

`String[] biến = { giá_trị1, giá_trị_2,...giá_trị_n}`

Ví dụ

```
String[] data = { "Việt Nam", "Hàn Quốc", "Nhật Bản", "Trung Quốc" }
```

## Đổ dữ liệu vào hộp

```
DefaultComboBoxModel model = new DefaultComboBoxModel();
```

```
for ( String phanTu : data ) model.addElement(phanTu);
```

```
cboXYZ.setModel(model);
```

## Làm việc với 1 mục

Chọn mục có index

`cboXYZ.setSelectedIndex(index)`

Lấy index của mục được chọn

`cboXYZ.getSelectedIndex()`

Đọc Item của mục được chọn

`cboXYZ.getSelectedItem()`

Ví dụ

```
cboCountry.setSelectedIndex(1)
```

```
int index = cboCountry.getSelectedIndex()
```

```
String country = (String)cboCountry.getSelectedItem()
```

## Các công việc khác





## Các công việc khác

Tính số lượng các phần tử

`cboXYZ.getItemCount()`

`int n = cboXYZ.getItemCount()`

Lấy dữ liệu tại mục có index

`cboXYZ.getItemAt(index)`

`String Item = cboXYZ.getItemAt(index)`

Thêm phần tử vào hộp

`cboXYZ.addItem(item)`

Xóa phần tử có Item

`cboXYZ.removeItem(item)`

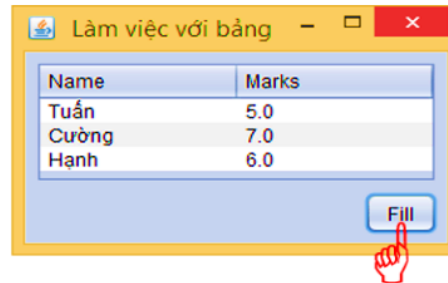
Xóa phần tử có index

`cboXYZ.removeItemAt(index)`

Xóa tất cả các phần tử

`cboXYZ.removeAllItems()`





```
Object[][] data = { { giá_trị_11, giá_trị_12, ..., giá_trị_1n },  
                    { giá_trị_21, giá_trị_22, ..., giá_trị_2n },  
                    ...  
                    { giá_trị_n1, giá_trị_n2, ..., giá_trị_nn } };
```

```
DefaultTableModel model = ( DefaultTableModel ) tblXYZ.getModel();  
model.setRowCount(0);  
for (Object[] row : data) model.addRow(row);
```

Vị trí hàng được chọn đầu tiên      `int index = tblXYZ.getSelectedRow()`

Số lượng hàng được chọn      `int count = tblXYZ.getSelectedRowCount()`

Vị trí các hàng được chọn      `int[] indices = tblXYZ.getSelectedRows()`





## Các công việc khác

Lấy Model

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

Thêm hàng dữ liệu

```
model.addRow(rowData);
```

Chèn hàng dữ liệu

```
model.insertRow(rowIndex, rowData);
```

Xóa hàng có rowIndex

```
model.removeRow(rowIndex);
```

Xóa các hàng chỉ để lại một số

```
model.setRowCount(rowCount);
```

Lấy dữ liệu trong ô

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

Ghi dữ liệu vào ô

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





Human Resource Management

Staff Management

EDITION

LIST

Staff Id

Fullname

Country

Việt Nam

Gender

☒ Male

☐ Female

Status

☒ Single?

Notes

Add

Remove

Update

Reset

Đăng nhập

**ĐĂNG NHẬP**

Username

Password

☐ Remember me?

Login

