



LẬP TRÌNH JAVA 2

BÀI 1: SWING INTRODUCTION

PHẦN 1

- Kết thúc bài học này bạn có khả năng
 - ❖ Thiết kế được giao diện đồ họa đơn giản
 - ❖ 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

Chú ý: Bài học này chỉ có mục đích giúp sinh viên sử dụng giao diện đồ họa để thay thế Console bằng Graphics.

- ❑ Ứng dụng giao diện đồ họa là ứng dụng sử dụng giao diện đồ họa để giao tiếp với người dùng
 - ❖ Nhập và hiển thị dữ liệu từ các thành phần đồ họa

QUẢN LÝ NHÂN VIÊN

10:22 AM

MÃ NHÂN VIÊN

HỌ VÀ TÊN

TUỔI

EMAIL

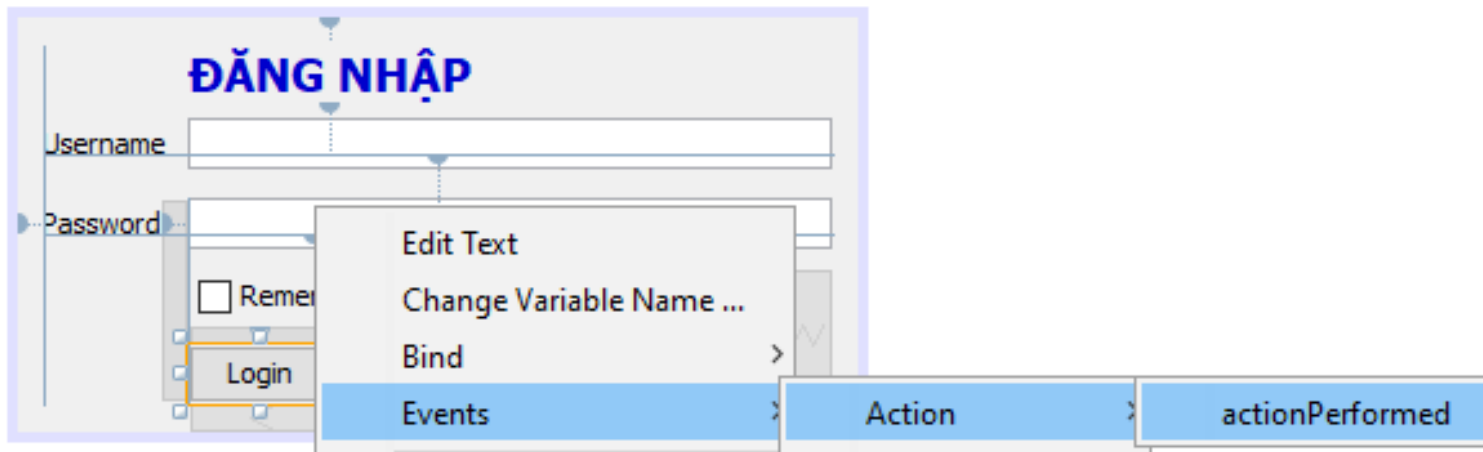
LƯƠNG

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

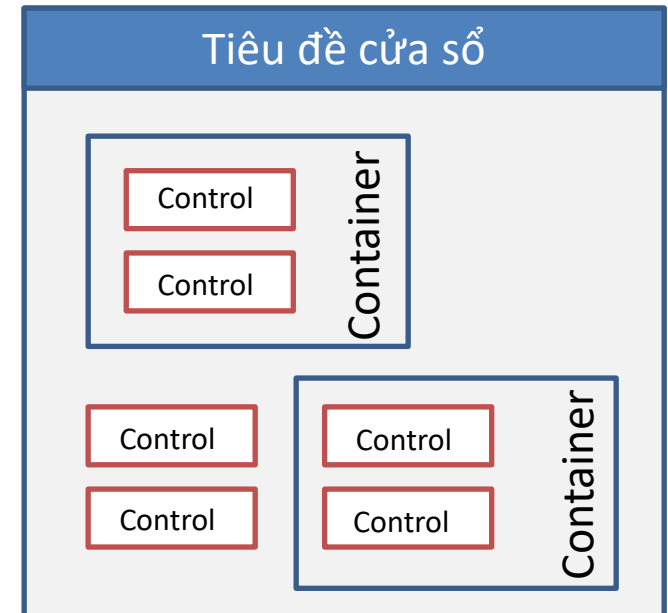
New
Save
Delete
Find
Open
Exit

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

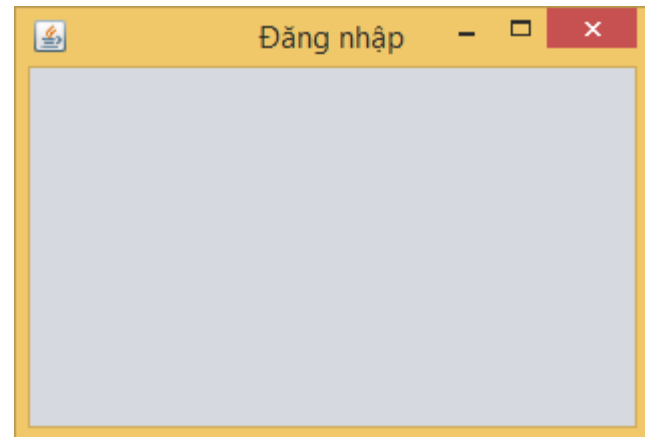
- ❑ Lập trình giao diện thường gắn kết với các sự kiện xảy ra trong quá trình sử dụng.
- ❑ Khi bạn click chuột vào nút thì sự kiện Action xảy ra.
- ❑ Nhấp đúp vào nút hoặc thực hiện theo hình sau



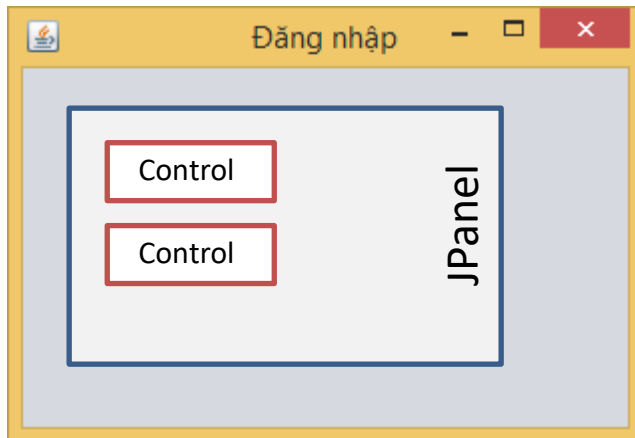
- ❑ Cửa sổ - Window
 - ❖ Cửa sổ ứng dụng.
- ❑ Khung chứa – Container
 - ❖ Chứa một hoặc nhiều điều khiển
- ❑ Điều khiển – Control
 - ❖ TextField, Radio, CheckBox, Button, Table...
- ❑ Chú ý:
 - ❖ Một ứng dụng có thể có nhiều cửa sổ.
 - ❖ Một khung chứa có thể chứa nhiều điều khiển hoặc khung chứa khác



- ❑ 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



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



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

❑ 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

Đăng nhập

ĐĂNG NHẬP

Username

Password

☐ Remember me?

Login

□ Jlabel

➤ Methods:

- ✓ void setFont (Font f)
- ✓ void setText(String S) quy định chuỗi văn bản.
- ✓ String getText()
- ✓ void setIcon(Icon) quy định Icon
- ✓ getLength(): đưa ra chiều dài của chuỗi text.

➤ Events:

- ✓ mouseClicked

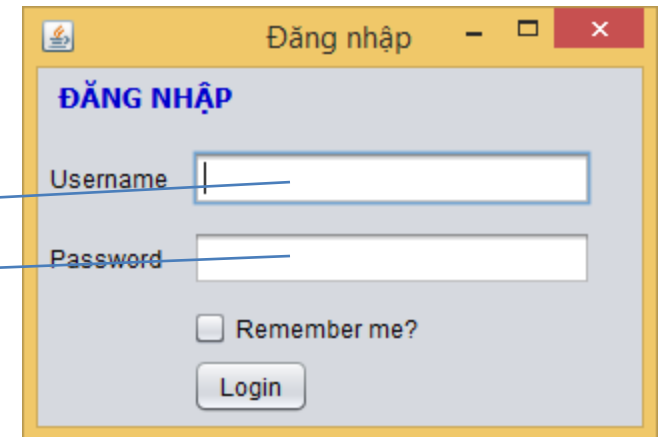
❑ JTextField được sử dụng để tạo ô nhập. 2 thuộc tính thường dùng là Text và Name

- ❖ Text: giá trị của ô nhập
- ❖ Name: tên của ô nhập, nên bắt đầu bởi txt
- ❖ Ví dụ:
 - txtUsername
 - txtPassword

JLabel

❑ JTextArea chỉ khác JTextField là cho phép chứa Text nhiều dòng

❑ JPasswordField thì không cho nhìn thấy nội dung của Text



□ JTextField

➤ Methods:

- ✓ void setText(String text): gán giá trị text cho JTextField
Vd: txtHoten.setText("Nguyen Van Teo")
- ✓ String getText(): lấy giá trị text của JTextField
Vd: String s = txtHoten.getText();
- ✓ void setEditable(boolean b): đặt/tắt chế độ nhập
- ✓ trim(): bỏ đi khoảng trắng đầu và cuối chuỗi
- ✓ setFont: thiết lập font cho JTextField
 - ✓ textField.setFont(new Font("Arial", Font.BOLD, 12));
- ✓ requestFocus(): chuyển con trỏ về JTextField

➤ Events:

- ✓ *ActionListener*
- ✓ *TextListener*
 - Cài đặt phương thức *keyPressed()*, *keyReleased()*, *keyTyped()* ;



DEMO

Chạy và giải thích





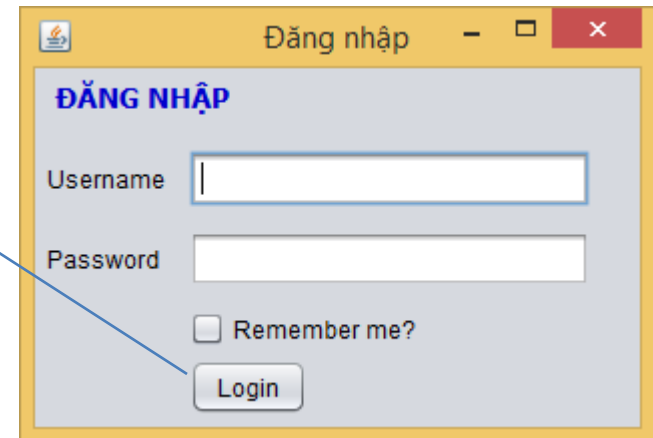
LẬP TRÌNH JAVA 2

BÀI 1: SWING INTRODUCTION

PHẦN 2

❑ JButton được sử dụng để tạo nút nhấn. Các thuộc tính thường sử dụng

- ❖ Text: nhãn của nút
- ❖ Name: tên của nút, nên bắt đầu bởi btn
- ❖ Ví dụ:
 - btnLogin



❏ JButton

➤ Methods:

- ✓ setText (String text)
- ✓ getText ()
- ✓ setForeground (Color fg)
- ✓ setFocusCycleRoot (boolean b)

➤ Events

- ✓ actionPerformed

➤ Ví dụ

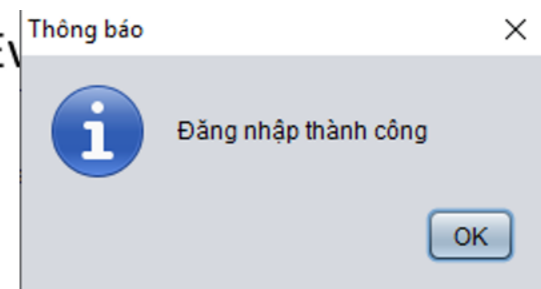
```
private void btnLoginActionPerformed(ActionEvent e)
```

```
// TODO add your handling code here:
```

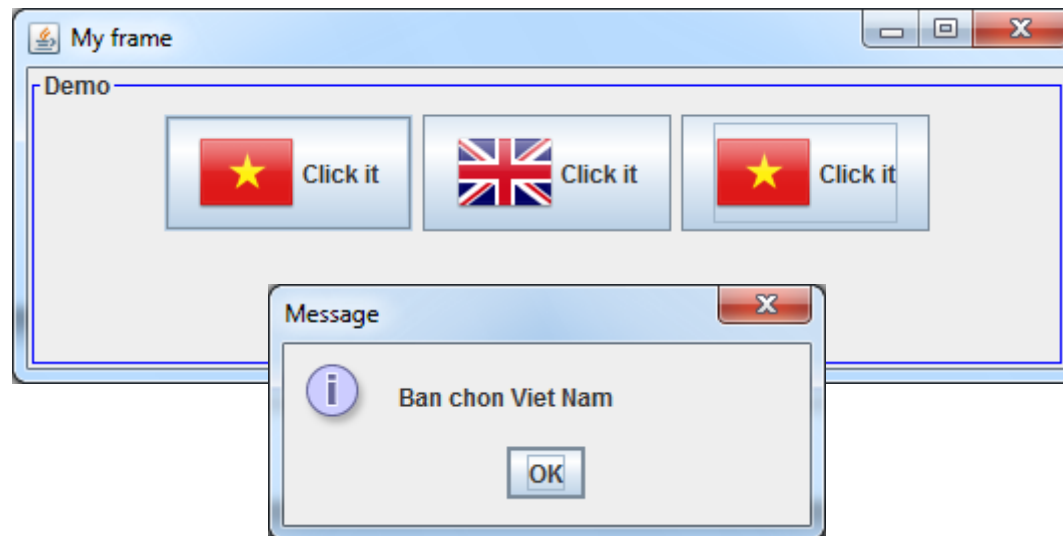
```
JOptionPane.showMessageDialog(this,
```

```
"Đăng nhập thành công", "Thông báo",  
JOptionPane.INFORMATION_MESSAGE);
```

```
}
```

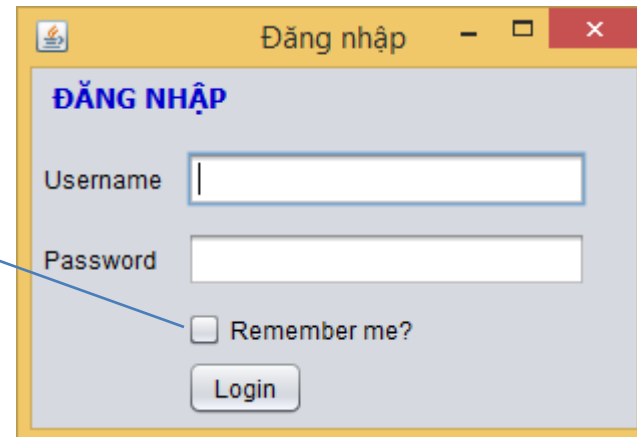


❑ Demo 1:



❑ JCheckBox được sử dụng để tạo hộp kiểm. Các thuộc tính thường dùng

- ❖ Text: Nhãn đính kèm
- ❖ Selected: Trạng thái
- ❖ Name: Tên, nên bắt đầu bởi chk
- ❖ Ví dụ:
 - chkRemember



JCheckBox

Methods:

- ❖ setSelected (boolean b)
- ❖ isSelected ()
- ❖ setText (String text)
- ❖ getText ()

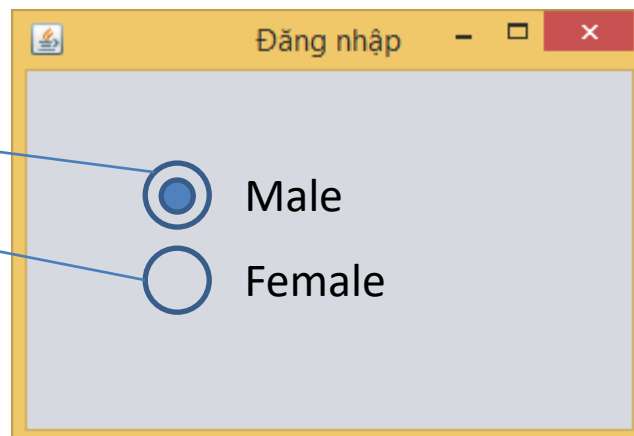
Events:

- ❖ actionPerformed

- ❑ JRadioButton được sử dụng để tạo các mục chọn. Các thuộc tính thường dùng
 - ❖ 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 chọn một radio.
 - ❖ Ví dụ:

➤ rdoMale

➤ rdoFemale



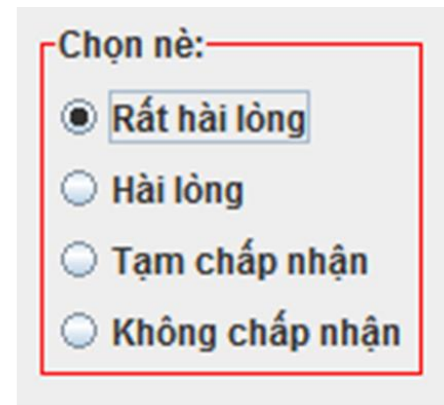
□ JRadioButton

□ Methods:

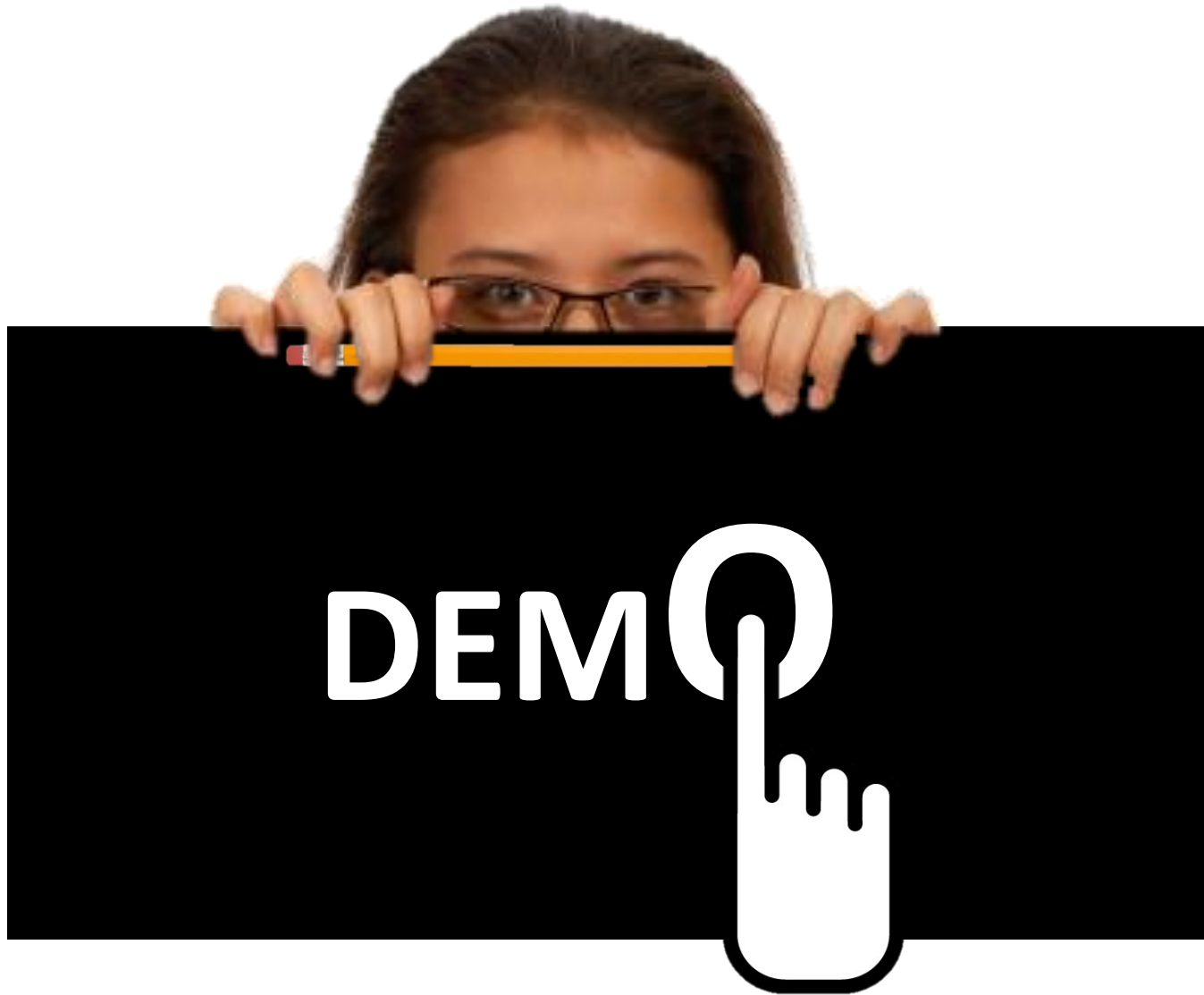
- ❖ setSelected (boolean b)
- ❖ isSelected()
- ❖ setText (String text)
- ❖ getText ()

□ Events:

- ❖ actionPerformed

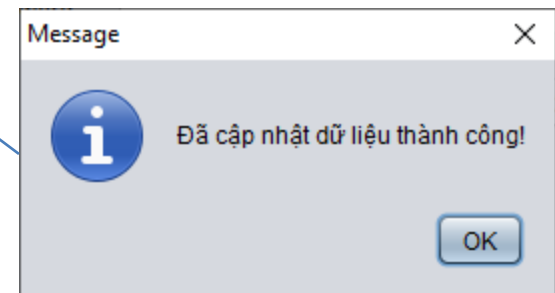
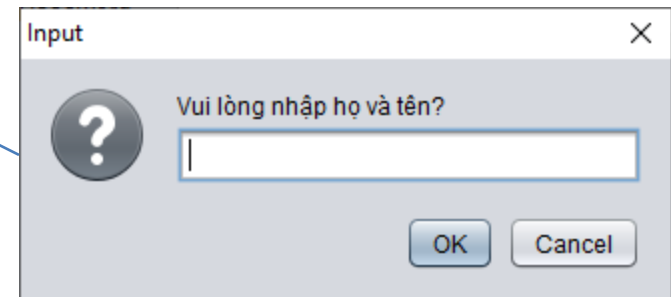
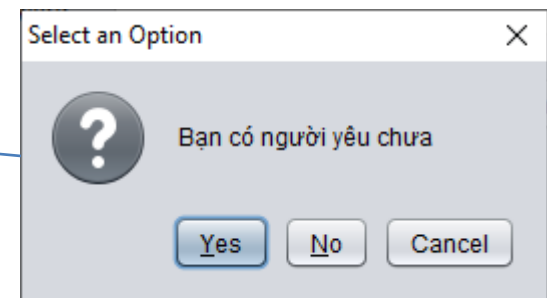


```
if(rad1.isSelected())
{
    //...
}
```



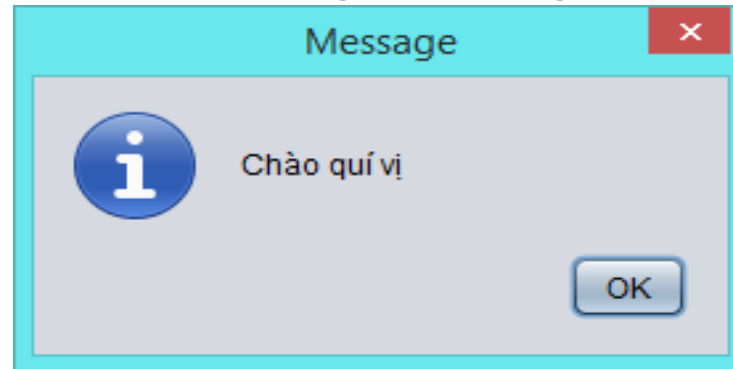
❑ JOptionPane là một dạng cửa sổ được sử dụng để tạo các hộp thoại thông báo. Có 3 dạng thông báo thường dùng

- ❖ Xác nhận – Confirm
- ❖ Nhập – Input
- ❖ Thông báo - Message

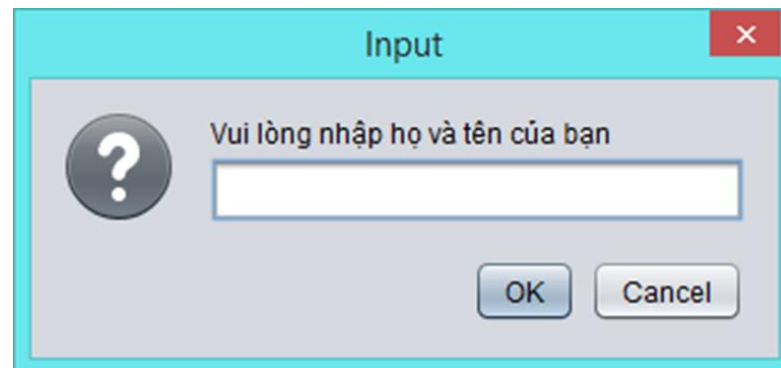


❑ Ví dụ:

✓ `JOptionPane.showMessageDialog(this,"Chào quý vị");`

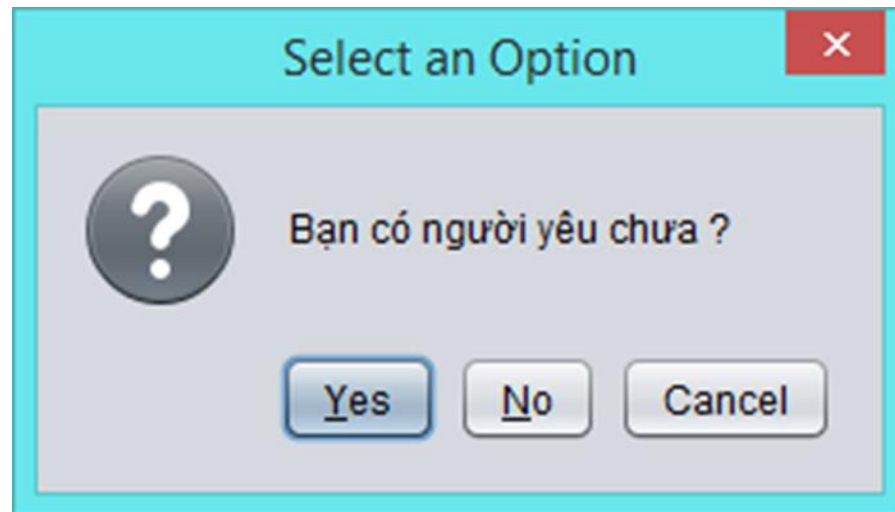


✓ `String input = JOptionPane.showInputDialog(this,"Vui lòng nhập họ và tên của bạn");`



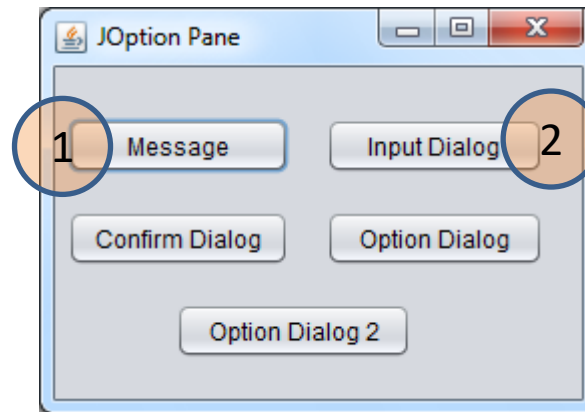
❑ Ví dụ:

```
int choice=JOptionPane.showConfirmDialog(this,"Bạn có  
người yêu chưa?");  
If(JOptionPane.YES_OPTION==choice) {  
//do something  
}  
else if(JOptionPane.NO_OPTION==choice){  
//do something  
}  
else{  
  
}
```



❑ JDialog

❑ Ví dụ:



1

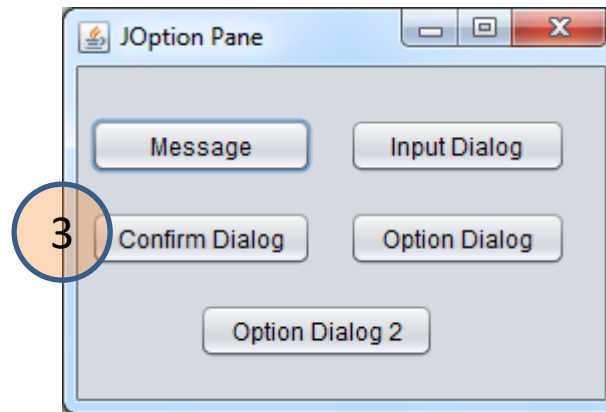
```
private void btnMessageActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JOptionPane.showMessageDialog(this, "Hi! chao cac ban");
}
```

2

```
private void btnInputDialogActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String name=JOptionPane.showInputDialog("Nhap ho ten:");
    if (name!=null) {
        JOptionPane.showMessageDialog(this, "Chao ban "+name);
    }
}
```

❑ JDialog

❑ Ví dụ:



3

```
private void btnConfirmActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int ret = JOptionPane.showConfirmDialog(this, "Are you sure?"
        , "Confirm", JOptionPane.YES_NO_OPTION
        , JOptionPane.QUESTION_MESSAGE);
    if(ret == JOptionPane.YES_OPTION) {
        //Xử lý trường hợp người dùng chọn Yes
        JOptionPane.showMessageDialog(this, "Ban chon: YES ");
    } else {
        //Xử lý trường hợp người dùng chọn No
        JOptionPane.showMessageDialog(this, "Ban chon: NO");
    }
}
```

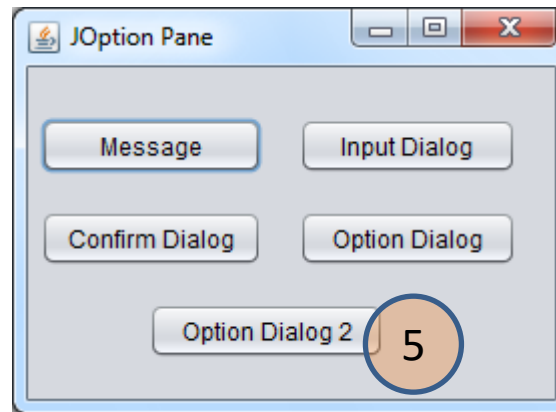
❑ Ví dụ:



4

```
private void btnOptionDialogActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Object[] options = {"Yes, please", "No way!"};
    int n = JOptionPane.showOptionDialog(this, "Would you like green eggs and ham?",
        "A Silly Question", JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE,
        null, //do not use a custom Icon
        options, //the titles of buttons
        options[0]); //default button title
}
```

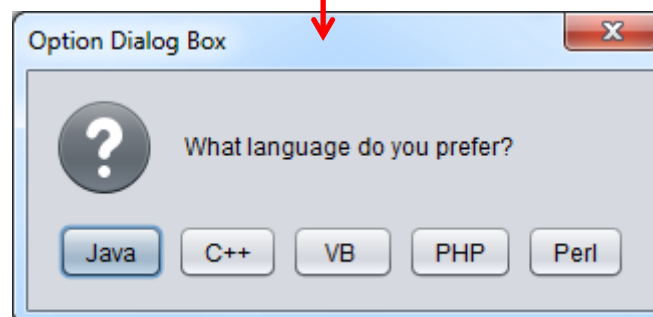
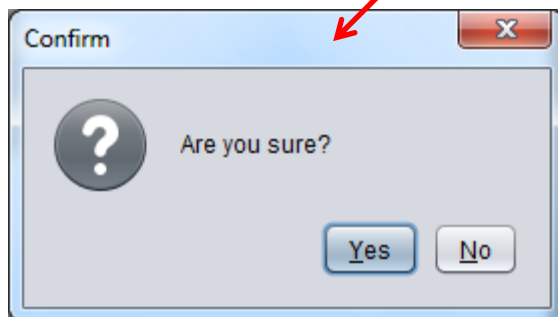
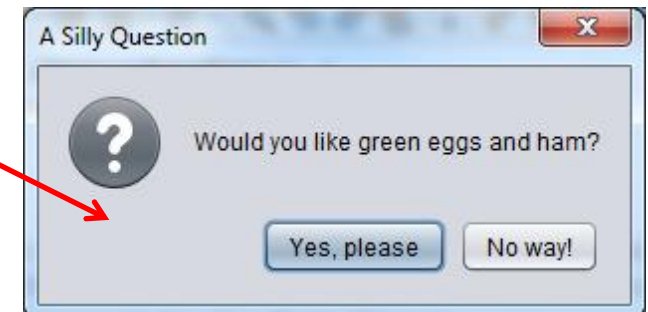
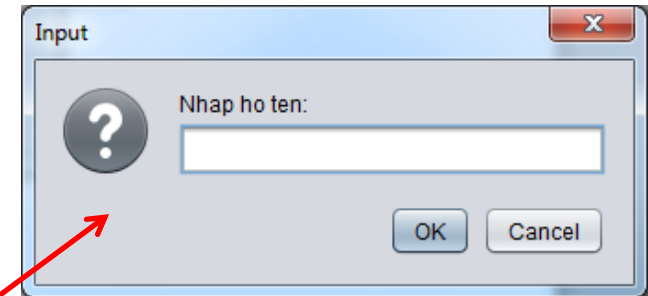
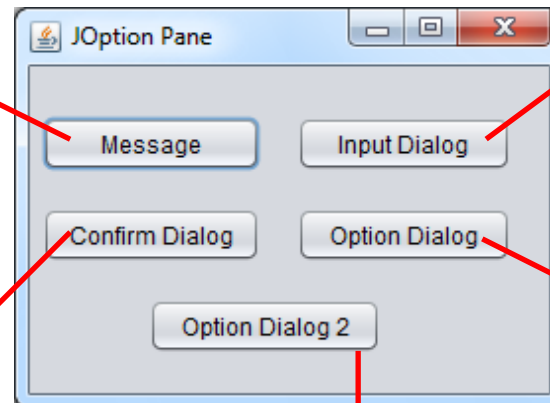
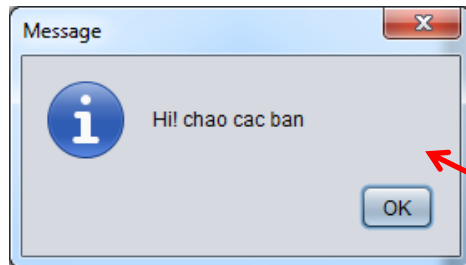
❑ Ví dụ:

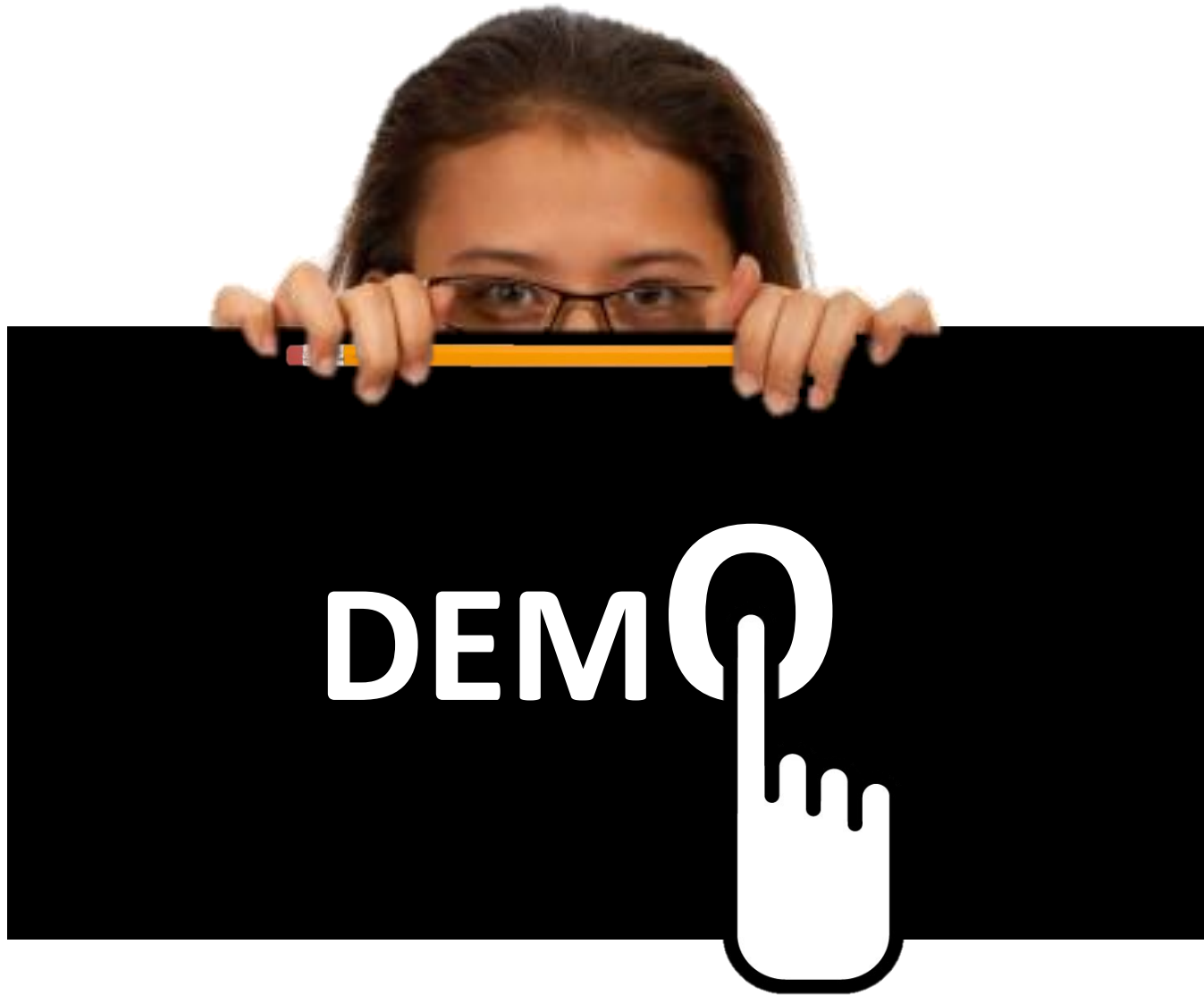


5

```
private void btnOptionDialog2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int messageType = JOptionPane.QUESTION_MESSAGE;
    String[] options = {"Java", "C++", " VB ", "PHP", "Perl"};
    int code = JOptionPane.showOptionDialog(this,
        "What language do you prefer?",
        "Option Dialog Box", 0, messageType,
        null, options, "Java");
    JOptionPane.showMessageDialog(this, "Answer: "+code);
}
```

☐ Ví dụ:





❑ Các thành phần giao diện

- ❖ Container: JFrame, JPanel

- ❖ Control:

- JLabel - Nhãn
- JTextField – Ô nhập
- JButton – Nút nhấn
- JCheckBox – Hộp kiểm
- JRadioButton – Lựa chọn
- JOptionPanel – hộp thoại

❑ Lập trình giao diện

- ❖ Làm việc với thuộc tính của Control
- ❖ Làm việc với sự kiện

