

LẬP TRÌNH JAVA (TT)

Đỗ Ngọc Như Loan - Nguyễn Thị Hồng Anh

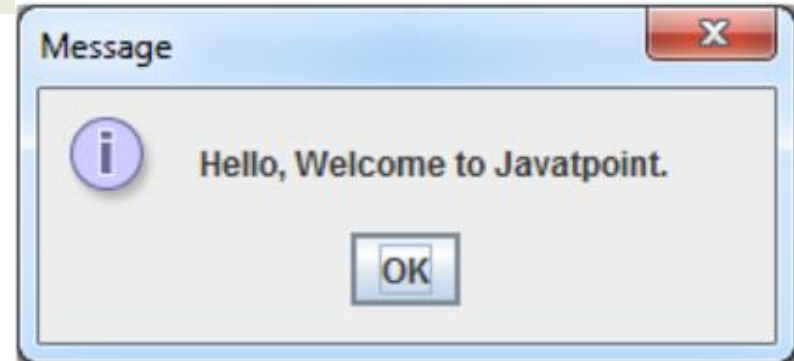
Chương 3. Lập trình giao diện đồ họa

Hiển thị Dialog (JOptionPane)



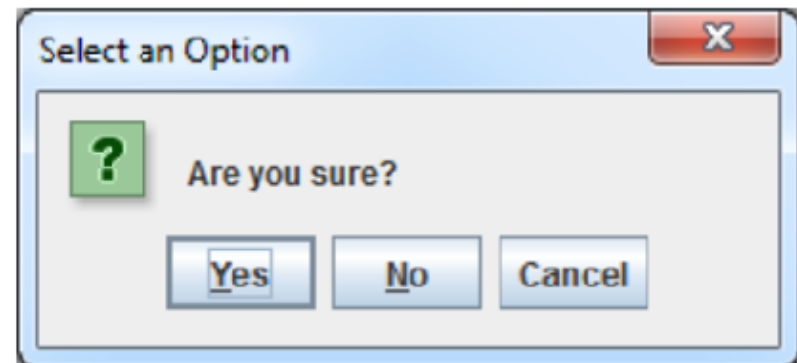
- Message Dialog

`showMessageDialog`



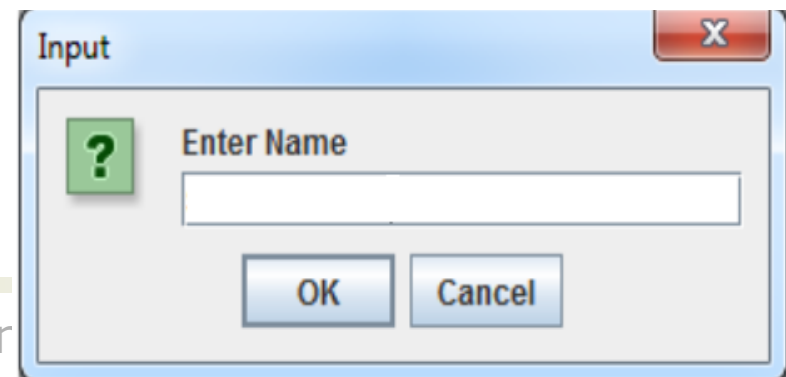
- Confirm Dialog

`showConfirmDialog`



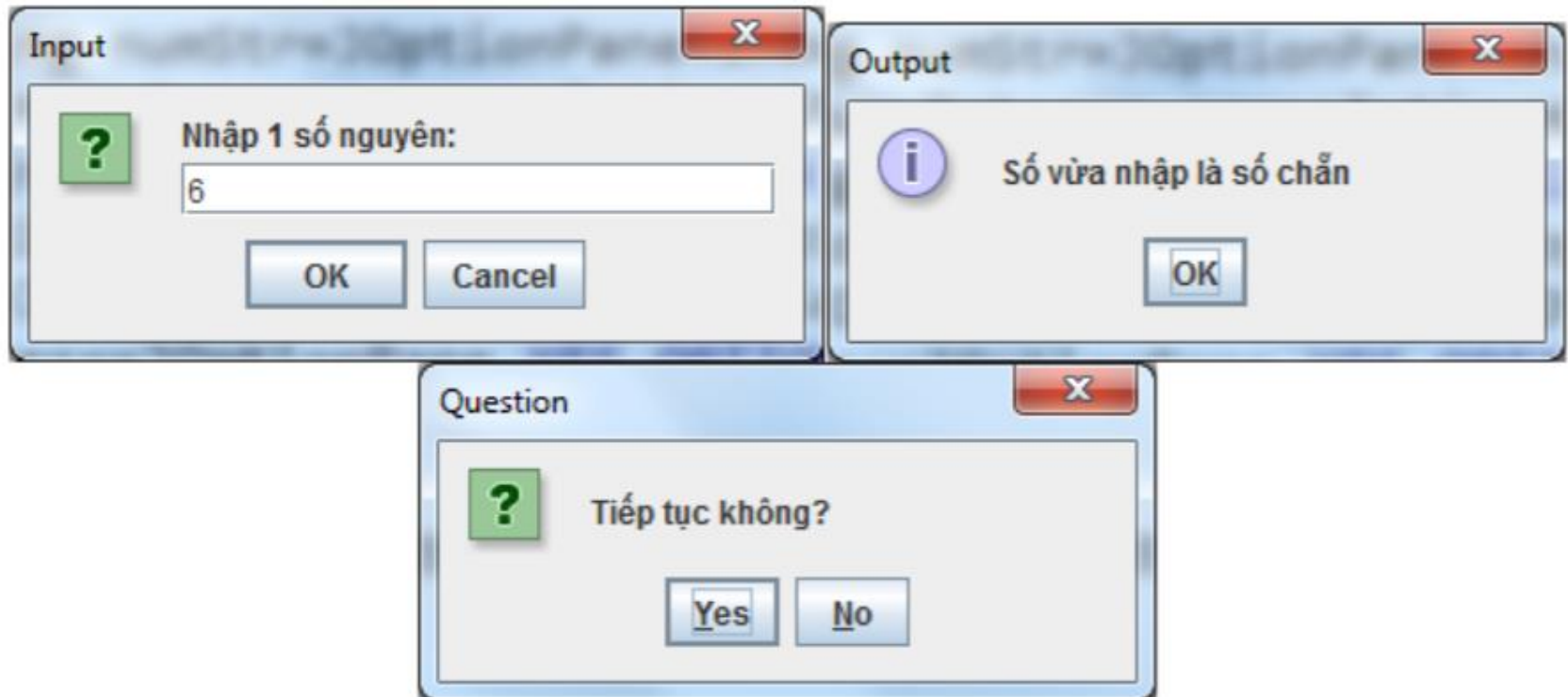
- Input Dialog

`showInputDialog`



Chương 3. Lập trình giao diện đồ họa

Ví dụ



Chương 3. Lập trình giao diện đồ họa

Ví dụ



```
import javax.swing.JOptionPane;

public class EvenOdd {
    public static void main(String[] args) {
        int qes;
        do {
            String numStr=JOptionPane.showInputDialog("Nhập 1 số
nguyên: ");
            if(numStr!=null && !numStr.trim().equals("")) {
                try {
                    int num=Integer.parseInt(numStr);
                    String result="Số vừa nhập là số "+ ((num%2==0)
? "chẵn" : "lẻ");
                    JOptionPane.showMessageDialog(null, result,
"Output",JOptionPane.INFORMATION_MESSAGE); }

```

Chương 3. Lập trình giao diện đồ họa

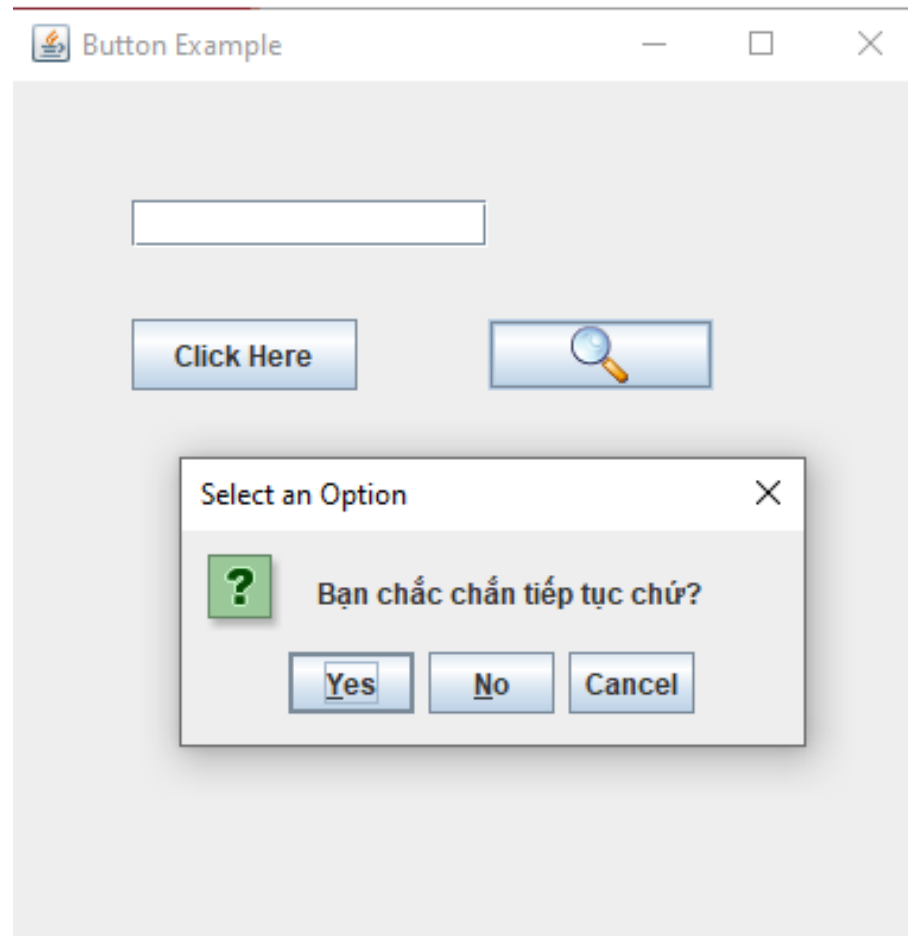
Ví dụ



```
        catch (NumberFormatException ex){
            JOptionPane.showMessageDialog(null, "Nhập sai
dữ liệu");
        }
    }
    qes=JOptionPane.showConfirmDialog(null, "Tiếp tục
không?", "Question",JOptionPane.YES_NO_OPTION);
    } while(qes==JOptionPane.YES_OPTION);
}
}
```

Chương 3. Lập trình giao diện đồ họa

JButton – Xử lý sự kiện



Chương 3. Lập trình giao diện đồ họa

JButton – Xử lý sự kiện (1)



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

public class JButtonAction {
    public static void main(String[] args) {
        JFrame f=new JFrame("Button Example");
        JTextField tf=new JTextField();
        tf.setBounds(50,50, 150,20);
        JButton b=new JButton("Click Here");
        b.setBounds(50,100,95,30);
        b.addActionListener(new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent e){
                tf.setText("Welcome to Javapoint.");
            }
        });
    }
}
```

Chương 3. Lập trình giao diện đồ họa

JButton – Xử lý sự kiện (1)



```
JButton c= new JButton(new ImageIcon("E:\\Imagelcon\\Find.png"));
c.setBounds(200,100 , 95, 30);
c.addActionListener(new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e){
        JOptionPane.showConfirmDialog(f, "Bạn chắc chắn tiếp tục
chứ?");
            }    });
f.add(b);f.add(tf); f.add(c);
f.setSize(400,400);
f.setLayout(null);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setVisible(true);
    }    }
```


Chương 3. Lập trình giao diện đồ họa

JButton – Xử lý sự kiện (2)



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

public class JButtonAction implements ActionListener {
    JFrame f;
    JButton b,c;
    JTextField tf;
    public JButtonAction() {
        f=new JFrame("Button Example");
        tf=new JTextField();
        tf.setBounds(50,50, 150,20);
        b=new JButton("Click Here");
        b.setBounds(50,100,95,30);
        b.addActionListener(this);
    }
}
```

Chương 3. Lập trình giao diện đồ họa

JButton – Xử lý sự kiện (2)



```
c= new JButton(new ImageIcon("E:\\Imagelcon\\Find.png"));
c.setBounds(200,100 , 95, 30);
c.addActionListener(this);
f.add(b);f.add(tf); f.add(c);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
```

Chương 3. Lập trình giao diện đồ họa

JButton – Xử lý sự kiện (2)



@Override

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

```
    if (e.getSource() == b) {
```

```
        tf.setText("Welcome to Javapoint.");
```

```
    } else if (e.getSource() == c) {
```

```
        JOptionPane.showConfirmDialog(f, "Bạn chắc chắn tiếp tục  
chứ?");
```

```
    }
```

```
}
```

```
public static void main(String[] args) {
```

```
    new JButtonAction();
```

```
}
```

```
}
```

Chương 3. Lập trình giao diện đồ họa

JCheckBox



- Đối tượng cho phép chọn/ không chọn giá trị
- Cho phép lựa chọn đồng thời nhiều giá trị
- Một số constructor thường dùng:
 - JCheckBox(): tạo ra 1 nút check box chưa được lựa chọn không có text và icon.
 - JCheckBox(String s): tạo ra 1 checkbox chưa có lựa chọn có text.
 - JCheckBox(String text, boolean selected): tạo 1 checkbox có text, lựa chọn hoặc không được lựa chọn ngay từ đầu.

Chương 3. Lập trình giao diện đồ họa

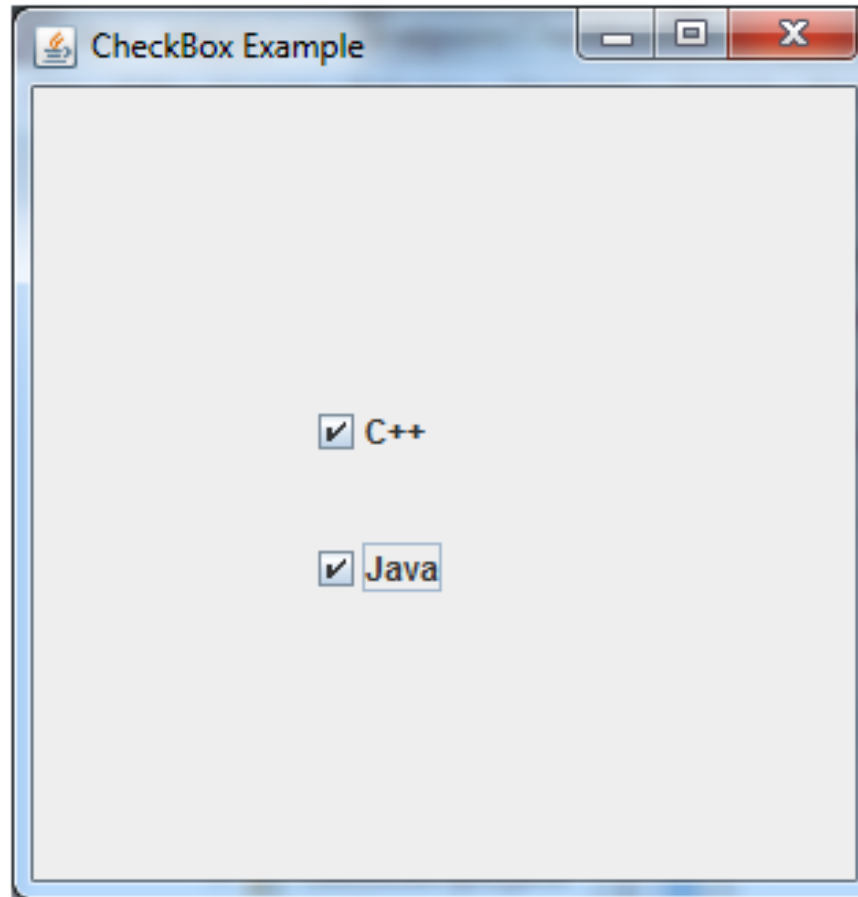
JCheckBox



- Các thuộc tính thường dùng:
 - Text: văn bản hiển thị trên đối tượng
 - Selected: đối tượng được lựa chọn hay không
- Một số phương thức thường dùng:
 - `protected String paramString():` trả về xâu đại diện của JCheckBox.
 - `boolean isSelected():` trả về true khi JCheckBox đang được select, false là ngược lại.
 - `setSelected(Boolean value):` dùng để set trạng thái cho JCheckBox là đang được chọn hoặc đang được bỏ chọn.

Chương 3. Lập trình giao diện đồ họa

Ví dụ



Lập trình Java

Chương 3. Lập trình giao diện đồ họa

Ví dụ



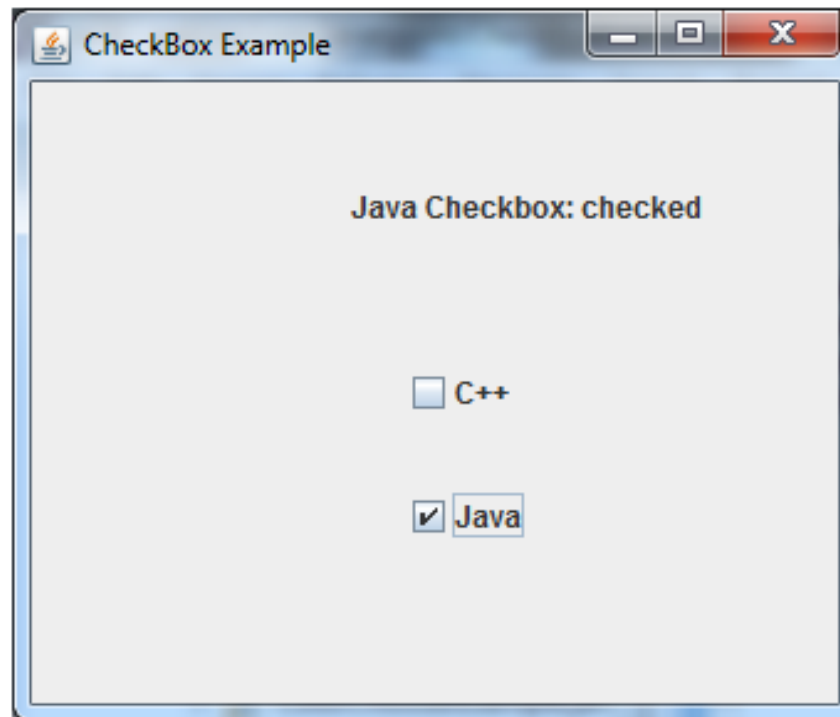
```
import javax.swing.*;

public class CheckBoxExample {
    CheckBoxExample(){
        JFrame f= new JFrame("CheckBox Example");
        JCheckBox checkBox1 = new JCheckBox("C++");
        checkBox1.setBounds(100,100, 50,50);
        JCheckBox checkBox2 = new JCheckBox("Java", true);
        checkBox2.setBounds(100,150, 50,50);
        f.add(checkBox1);    f.add(checkBox2);
        f.setSize(400,400); f.setLayout(null); f.setVisible(true);
    }

    public static void main(String args[])    {
        new CheckBoxExample();    }}
```

Chương 3. Lập trình giao diện đồ họa

Ví dụ JCheckBox



Chương 3. Lập trình giao diện đồ họa

Java JCheckBox Example with ItemListener



```
import javax.swing.*;
import java.awt.event.*;
public class CheckBoxExample {
    CheckBoxExample(){
        JFrame f= new JFrame("CheckBox Example");
        JLabel label = new JLabel();
        label.setHorizontalAlignment(JLabel.CENTER);
        label.setSize(400,100);
        JCheckBox checkbox1 = new JCheckBox("C++");
        checkbox1.setBounds(150,100, 50,50);
        JCheckBox checkbox2 = new JCheckBox("Java");
        checkbox2.setBounds(150,150, 50,50);
        f.add(checkbox1); f.add(checkbox2); f.add(label);
    }
}
```

Chương 3. Lập trình giao diện đồ họa



```
checkbox1.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent e) {
        label.setText("C++ Checkbox: "
            + (e.getStateChange() == 1 ? "checked" : "unchecked"));
    }
});

checkbox2.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent e) {
        label.setText("Java Checkbox: "
            + (e.getStateChange() == 1 ? "checked" : "unchecked"));
    }
});
```

Chương 3. Lập trình giao diện đồ họa



```
f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
}
public static void main(String args[])
{
    new CheckBoxExample();
}
}
```

Chương 3. Lập trình giao diện đồ họa

JRadioButton



- Đối tượng cho phép chọn/không chọn giá trị.
- Cho phép chọn một giá trị tại 1 thời điểm
- Một số constructor thường dùng:
 - `JRadioButton()`: tạo ra 1 radio button không có text và không được lựa chọn.
 - `JRadioButton(String s)`: tạo 1 radio button có text và không được lựa chọn.
 - `JRadioButton(String s, boolean selected)`: tạo radio button có text và trạng thái được chọn.

Chương 3. Lập trình giao diện đồ họa

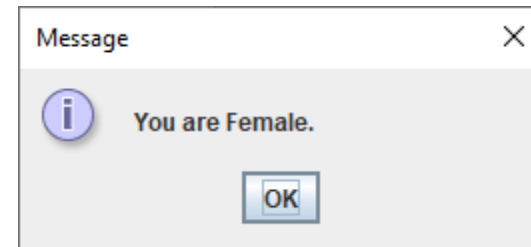
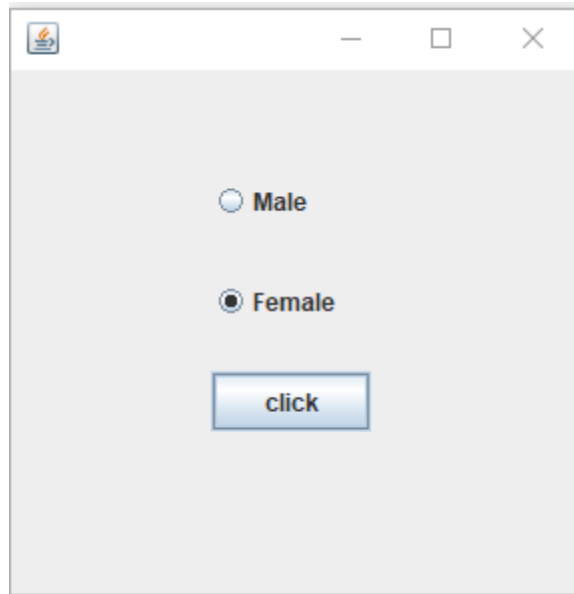
JRadioButton



- Một số phương thức thường dùng:
 - `isSelected()`: trả về `true` nếu đối tượng được check
- Lưu ý:
 - Các radio button nên được đặt trong 1 button group

Chương 3. Lập trình giao diện đồ họa

JRadioButton – Ví dụ 1



Chương 3. Lập trình giao diện đồ họa

JRadioButton – Ví dụ 1



```
public class JRadioEx1 extends JFrame implements
ActionListener{
    JRadioButton rb1,rb2;
    JButton b;
    JRadioEx1 (){
        rb1=new JRadioButton("Male");
        rb1.setBounds(100,50,100,30);
        rb2=new JRadioButton("Female");
        rb2.setBounds(100,100,100,30);
        ButtonGroup bg=new ButtonGroup();
        bg.add(rb1);bg.add(rb2);
```

Chương 3. Lập trình giao diện đồ họa

JRadioButton – Ví dụ 1

```
b=new JButton("click");  
b.setBounds(100,150,80,30);  
b.addActionListener(this);  
add(rb1);add(rb2);add(b);  
setSize(300,300);  
setLayout(null);  
setVisible(true);  
}
```


Chương 3. Lập trình giao diện đồ họa

JRadioButton – Ví dụ 1

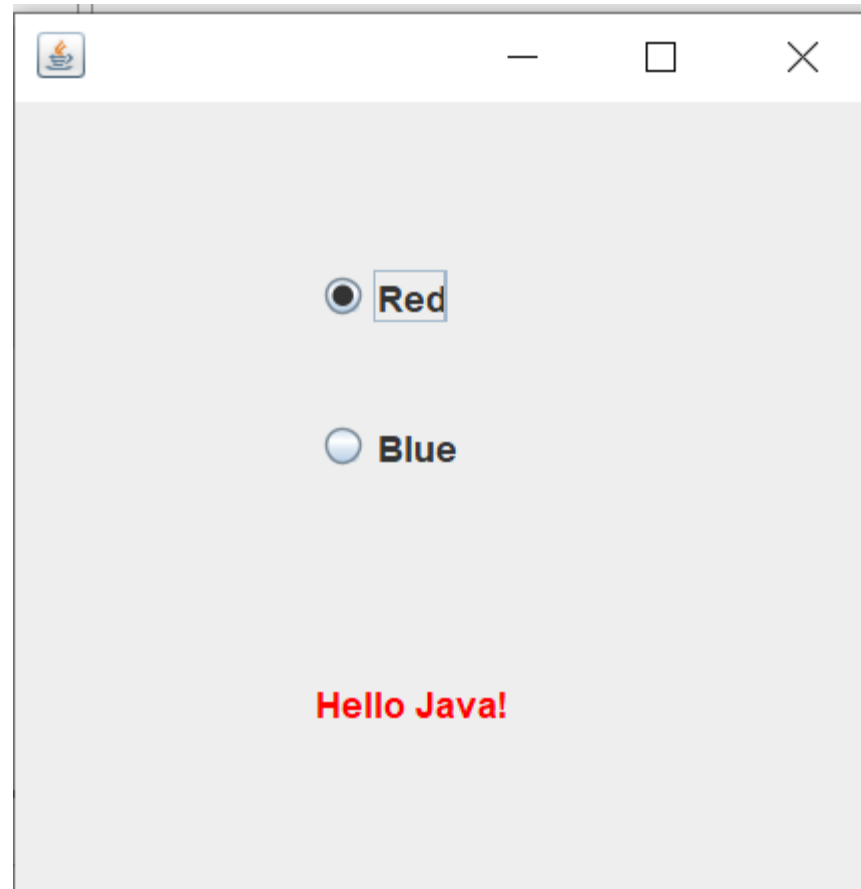


```
public void actionPerformed(ActionEvent e){
    if(rb1.isSelected()){
        JOptionPane.showMessageDialog(this,"You are Male.");
    }
    if(rb2.isSelected()){
        JOptionPane.showMessageDialog(this,"You are Female.");
    }
}

public static void main(String args[]){
    new JRadioEx1(); }
}
```

Chương 3. Lập trình giao diện đồ họa

JRadioButton – Ví dụ 2



Chương 3. Lập trình giao diện đồ họa

JRadioButton – Ví dụ 2



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

public class JRadioEx2 extends JFrame implements ActionListener{
    JRadioButton rb1,rb2;
    JButton b; JLabel label;
    JRadioEx2 (){
        label = new JLabel("Hello Java!");
        rb1=new JRadioButton("Red");
        rb1.setBounds(100,50,100,30);
        rb2=new JRadioButton("Blue");
        rb2.setBounds(100,100,100,30);
```

Chương 3. Lập trình giao diện đồ họa

JRadioButton – Ví dụ 2



```
label.setBounds(100,150,100,100);
ButtonGroup bg=new ButtonGroup();
bg.add(rb1);bg.add(rb2);
rb1.addActionListener(this);
rb2.addActionListener(this);
add(rb1);add(rb2);add(label);
setSize(300,300);
setLayout(null);
setVisible(true);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
```

Chương 3. Lập trình giao diện đồ họa

JRadioButton – Ví dụ 2



```
public void actionPerformed(ActionEvent e){
    if(rb1.isSelected()){
        label.setForeground(Color.red);
    }
    if(rb2.isSelected()){
        label.setForeground(Color.blue);
    }
}

public static void main(String args[]){
    new JRadioEx2(); }
}
```

Chương 3. Lập trình giao diện đồ họa

JComboBox



- Được dùng để tạo ra các menu trái xuống các lựa chọn. Mục đã chọn sẽ được hiển thị ngay trên đầu của menu.
- Các constructor thường dùng:
 - `JComboBox()`: tạo 1 combobox với mô hình dữ liệu ngầm định.
 - `JComboBox(Object[] items)`: tạo 1 combobox chứa các phần tử trong mảng chỉ định.
 - `JComboBox(Vector<?> items)`: tạo 1 combobox chứa các phần tử trong vector đã được chỉ định.

Chương 3. Lập trình giao diện đồ họa

JComboBox- Các phương thức thường dùng



- **getItemAt(int index):** trả về item ở vị trí index, phần tử đầu tiên của combo được hiểu ở vị trí 0, nếu index nằm ngoài phạm vi của combo sẽ trả về null.
- **void addItem(Object anObject):** dùng để thêm 1 item vào danh sách.
- **void removeItem(Object anObject):** dùng để xóa 1 item trong danh sách item.
- **void removeAllItems():** dùng để xóa toàn bộ item trong danh sách.

Chương 3. Lập trình giao diện đồ họa

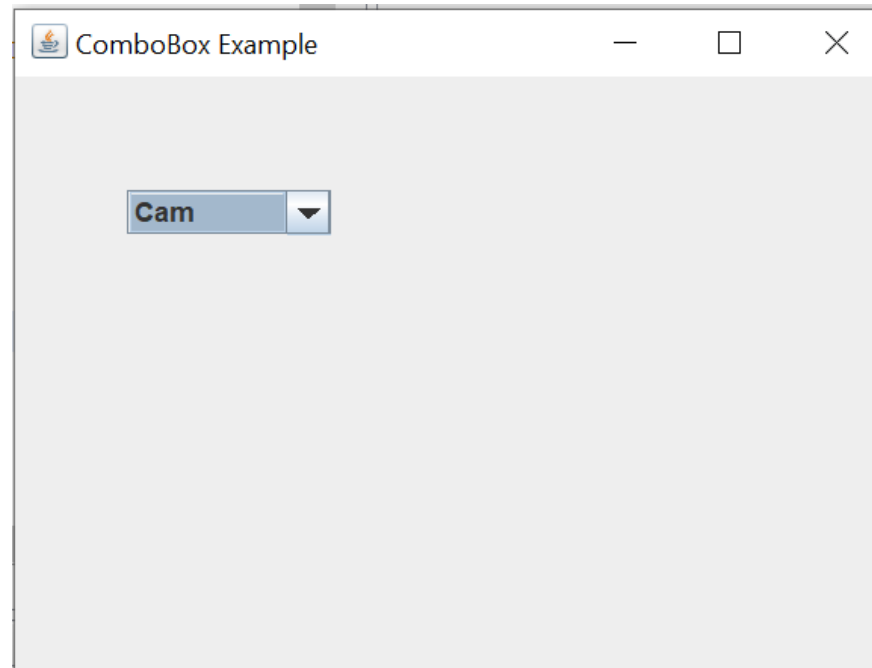
JComboBox- Các phương thức thường dùng



- **getSelectedIndex():** trả về chỉ số của item được chọn
- **getSelectedItem():** trả về nội dung item đang được lựa chọn
- **void setEditable(boolean b):** dùng để thiết lập Combobox có sửa được hay không.

Chương 3. Lập trình giao diện đồ họa

JComboBox – Ví dụ 1



Chương 3. Lập trình giao diện đồ họa

JComboBox – Ví dụ 1

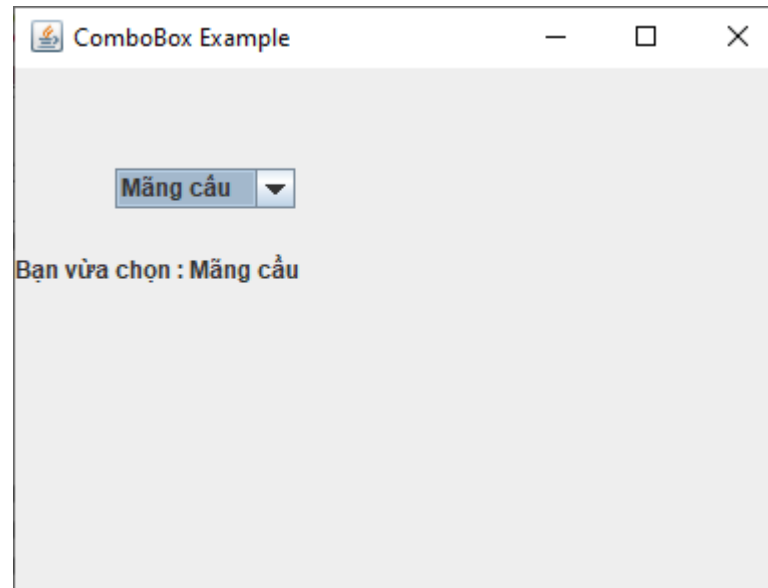


```
import javax.swing.*;

public class JComboExample {
    JFrame f;
    JComboExample(){
        f=new JFrame("ComboBox Example");
        String combo[]{"Cam","Quýt","Ổi","Mãng cầu","Lê"};
        JComboBox cb=new JComboBox(combo);
        cb.setBounds(50, 50,90,20);
        f.add(cb);          f.setLayout(null);
        f.setSize(400,300);    f.setVisible(true);    }
    public static void main(String[] args) {
        new JComboExample();    } }
```

Chương 3. Lập trình giao diện đồ họa

JComboBox – Ví dụ 2



Chương 3. Lập trình giao diện đồ họa

JComboBox – Ví dụ 2



```
public class JComboBoxExample {  
    JFrame f;  
    JComboBoxExample(){  
        f=new JFrame("ComboBox Example");  
        String combo[]={"Cam","Quýt","Ổi","Mãng cầu","Lê"};  
        JComboBox cb=new JComboBox(combo);  
        cb.setBounds(50, 50,90,20);  
        f.add(cb);  
        f.setLayout(null);  
        f.setSize(400,300);  
        JLabel label1 = new JLabel();  
        f.add(label1);  
    }  
}
```

Chương 3. Lập trình giao diện đồ họa

JComboBox – Ví dụ 2



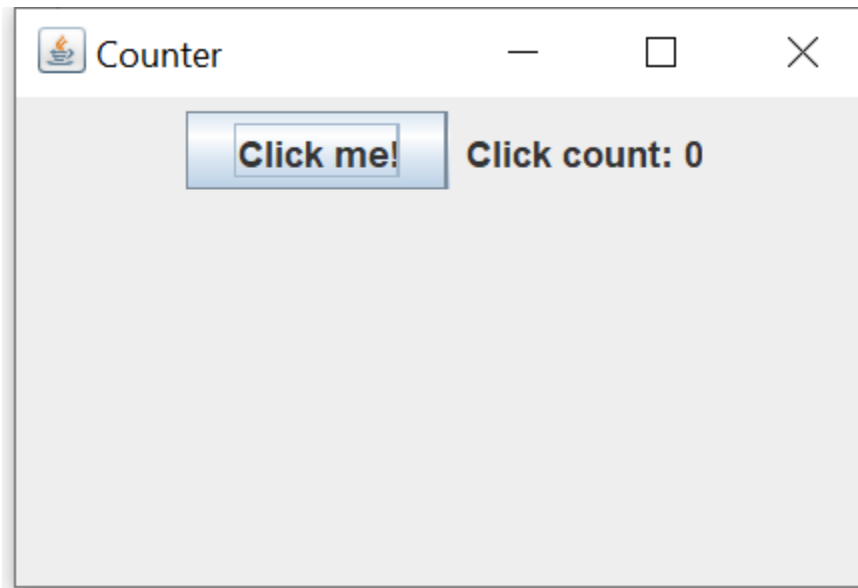
```
label1.setSize(300, 200);
cb.addActionListener(new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e) {
        label1.setText("Bạn vừa chọn : " +
cb.getSelectedItem());    }
});
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setVisible(true);    }
public static void main(String[] args) {
    new JComboExample();
}
}
```

Chương 3. Lập trình giao diện đồ họa

Bài tập



1) Viết chương trình đếm số lần click button.

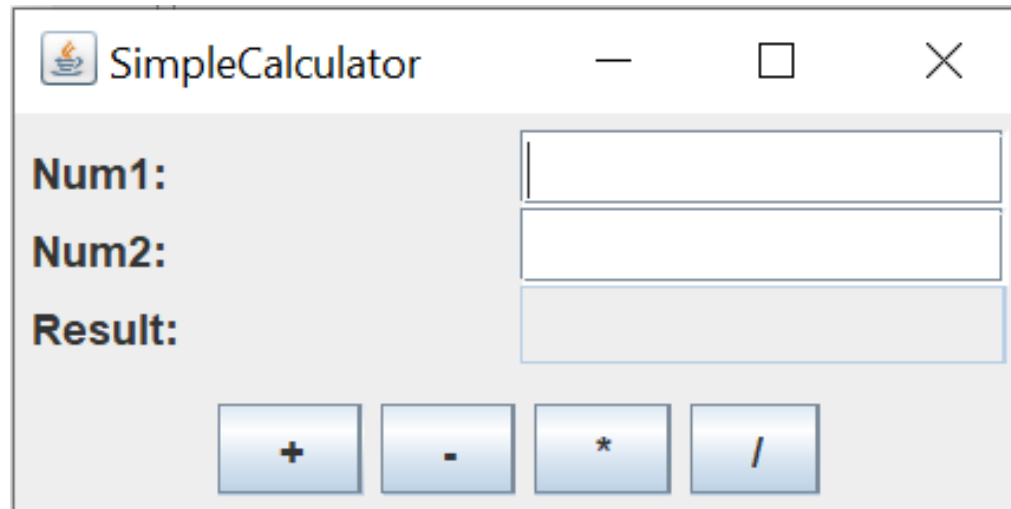


Chương 3. Lập trình giao diện đồ họa

Bài tập



2) Viết chương trình máy tính để thực hiện các phép cộng, trừ, nhân chia. Khi click chọn phép tính, nếu chưa nhập số hoặc số không hợp lệ sẽ hiện thông báo dạng dialog.



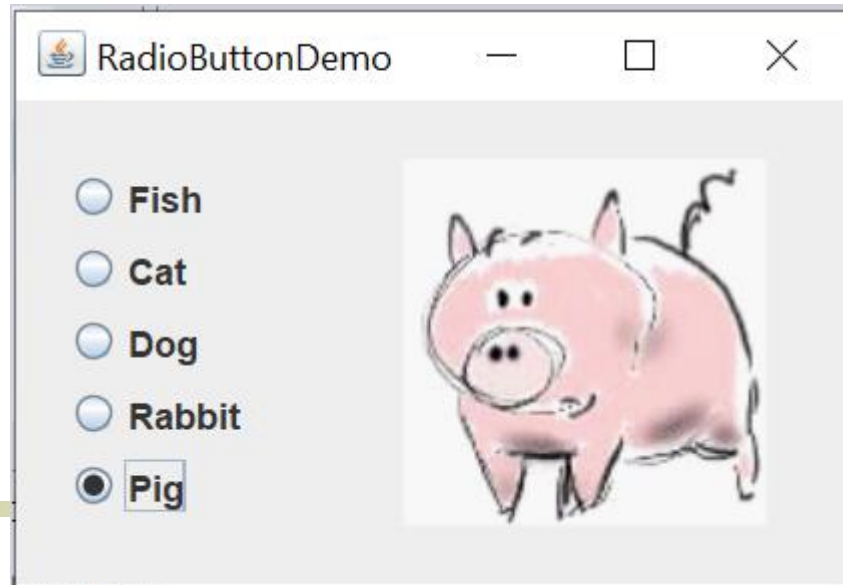
Chương 3. Lập trình giao diện đồ họa

Bài tập



3) Tạo 1 frame, trong đó có 1 combobox có các lựa chọn: White, Green, Blue, Yellow, Red. Khi chọn màu nào thì màu nền của frame sẽ đổi thành màu tương ứng.

4) Viết ứng dụng như sau. Khi chọn con vật nào thì ảnh của con vật đó sẽ hiển thị bên cạnh.

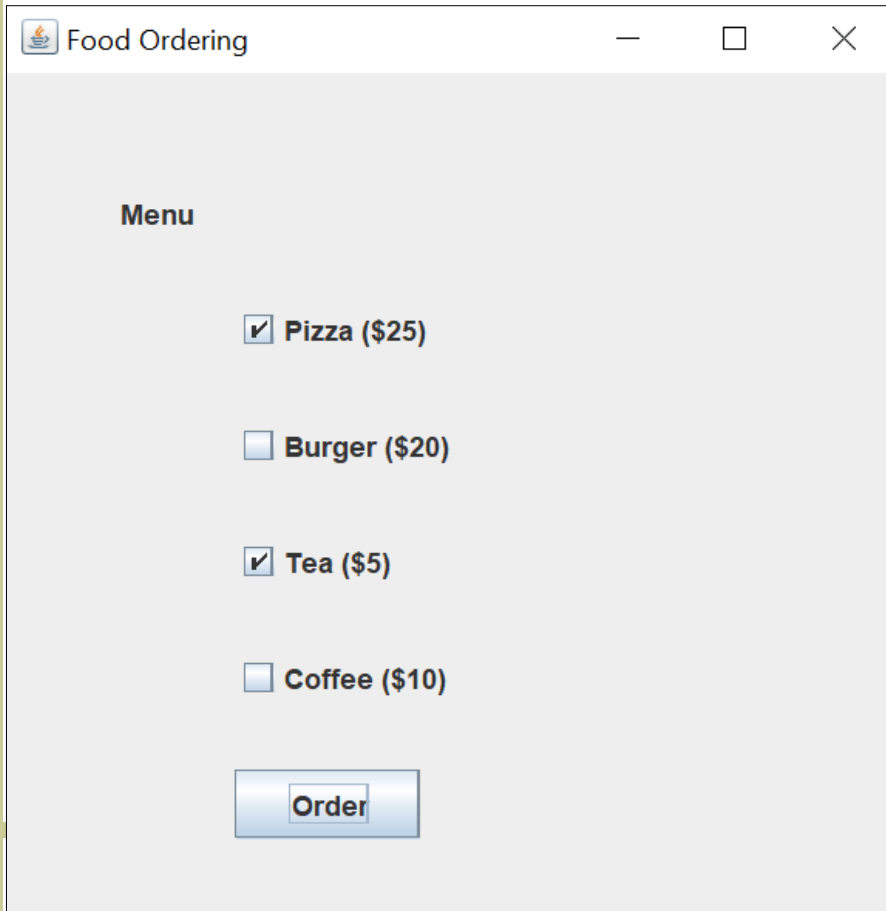


Chương 3. Lập trình giao diện đồ họa

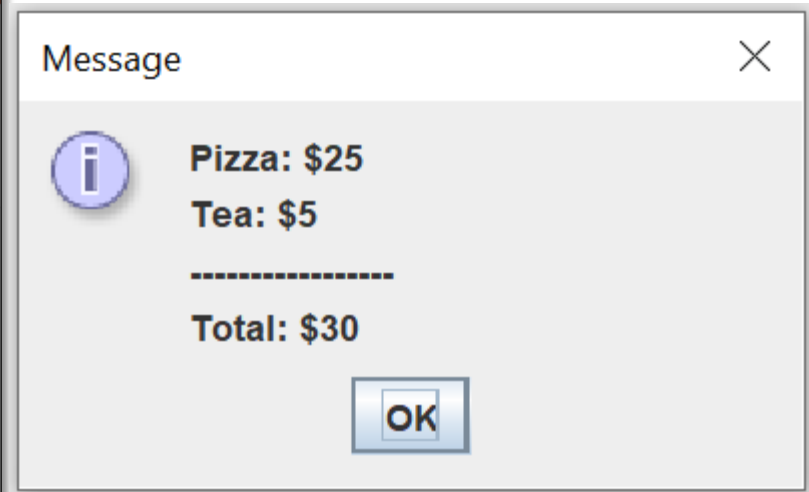
Bài tập



5) Viết ứng dụng đặt thức ăn như sau. Khi click nút Order sẽ hiện dialog chứa thông tin các món đã đặt và số tiền.



The screenshot shows a Java Swing window titled "Food Ordering". Inside the window, there is a section labeled "Menu" containing four items, each with a checkbox and a label: "Pizza (\$25)", "Burger (\$20)", "Tea (\$5)", and "Coffee (\$10)". The checkboxes for "Pizza" and "Tea" are checked. At the bottom of the window, there is a button labeled "Order".



The screenshot shows a Java Swing dialog box titled "Message". It contains an information icon (i) and the following text: "Pizza: \$25", "Tea: \$5", a separator line (-----), and "Total: \$30". At the bottom right of the dialog, there is an "OK" button.