

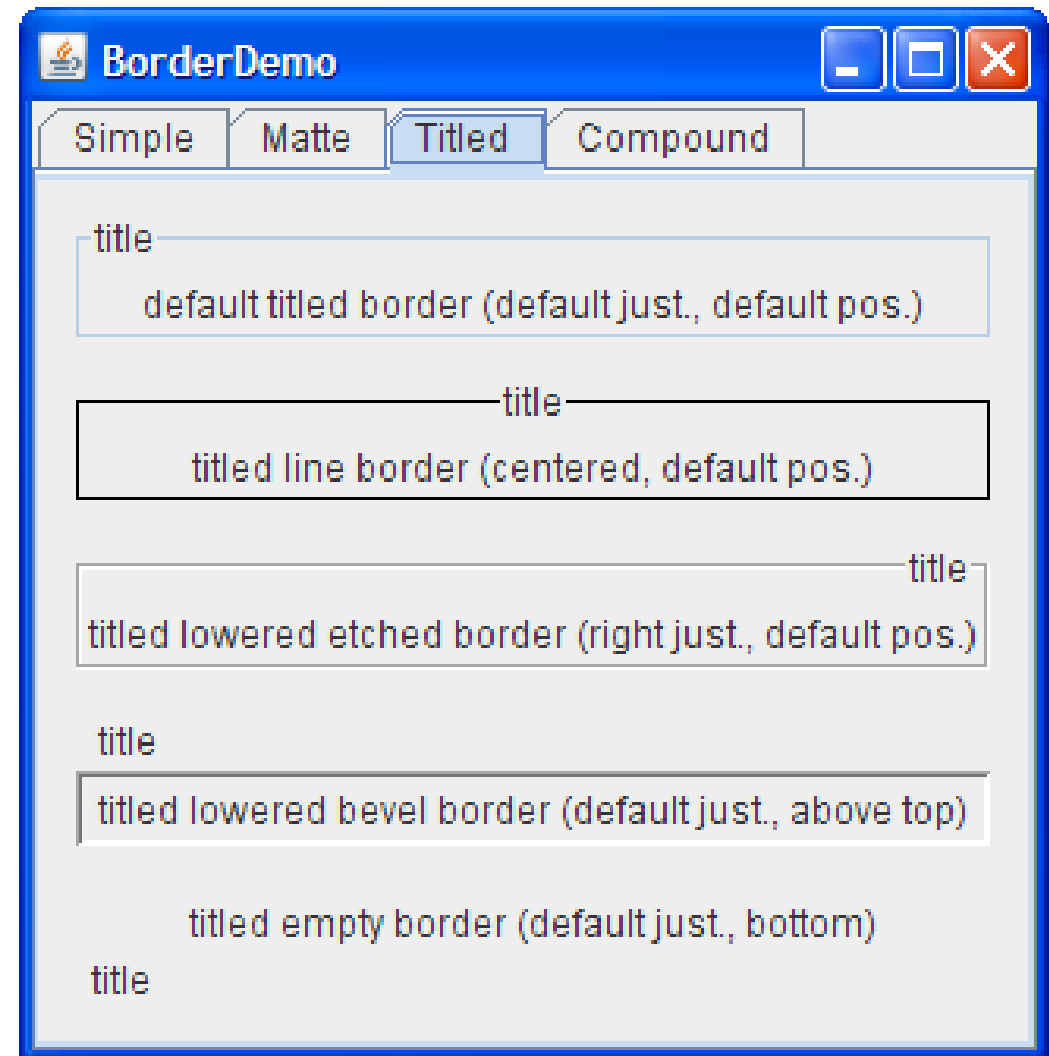
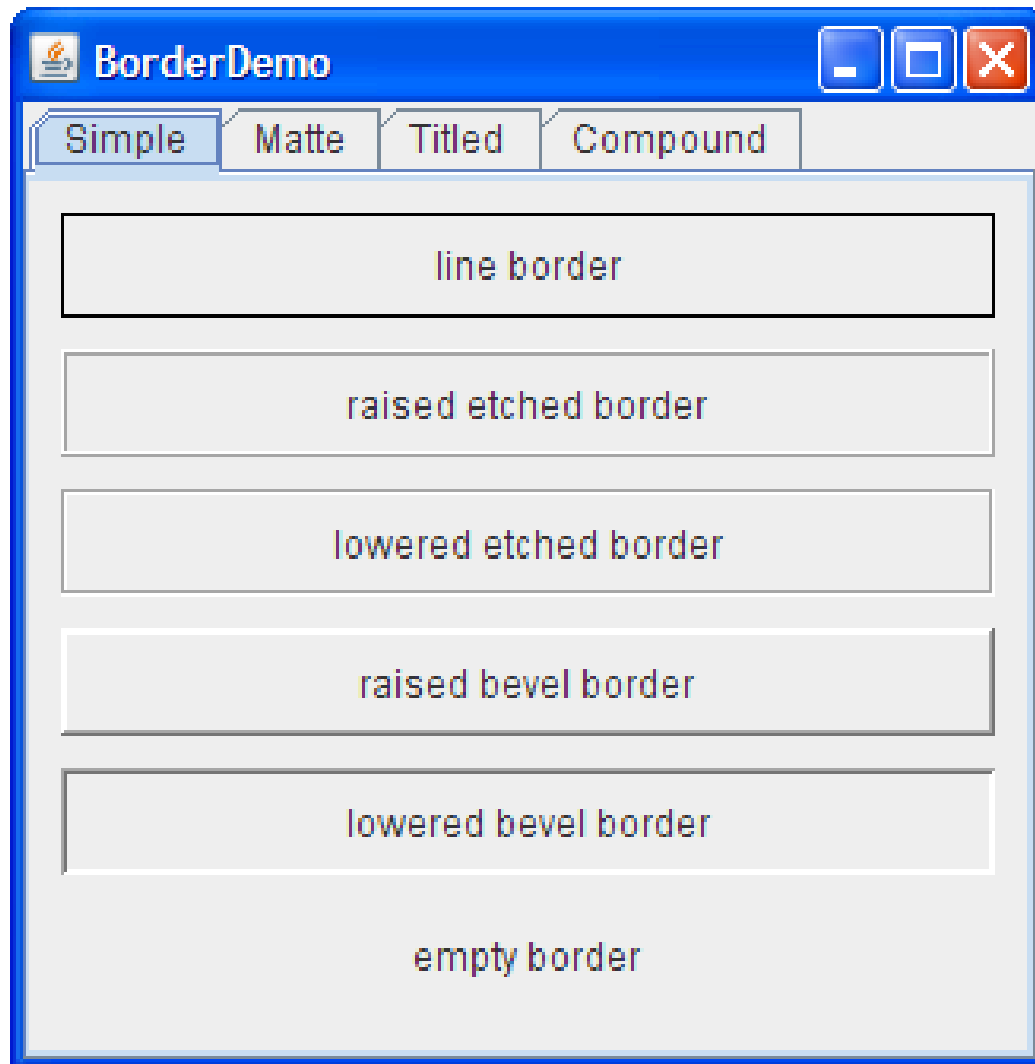
Bài 03:

CÁC LOẠI LAYOUT MANAGER

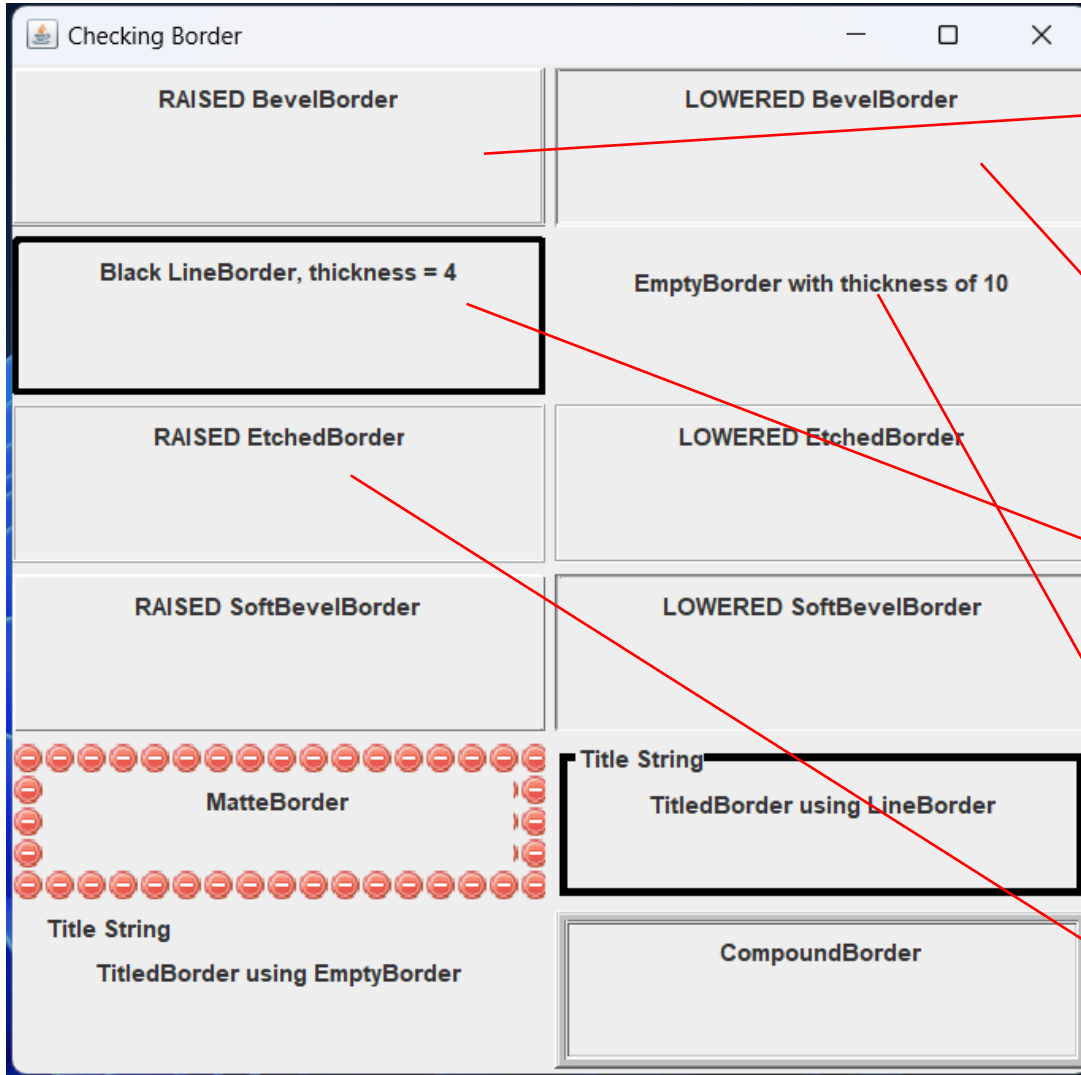
THƯỜNG DÙNG

Giảng Viên: ThS. Giang Hào Côn

Border và Border Title



Sử dụng Simple Border



```
JPanel p = new JPanel();
p.setBorder(new BevelBorder(BevelBorder.RAISED));
p.add(new JLabel("RAISED BevelBorder"));
content.add(p);
```

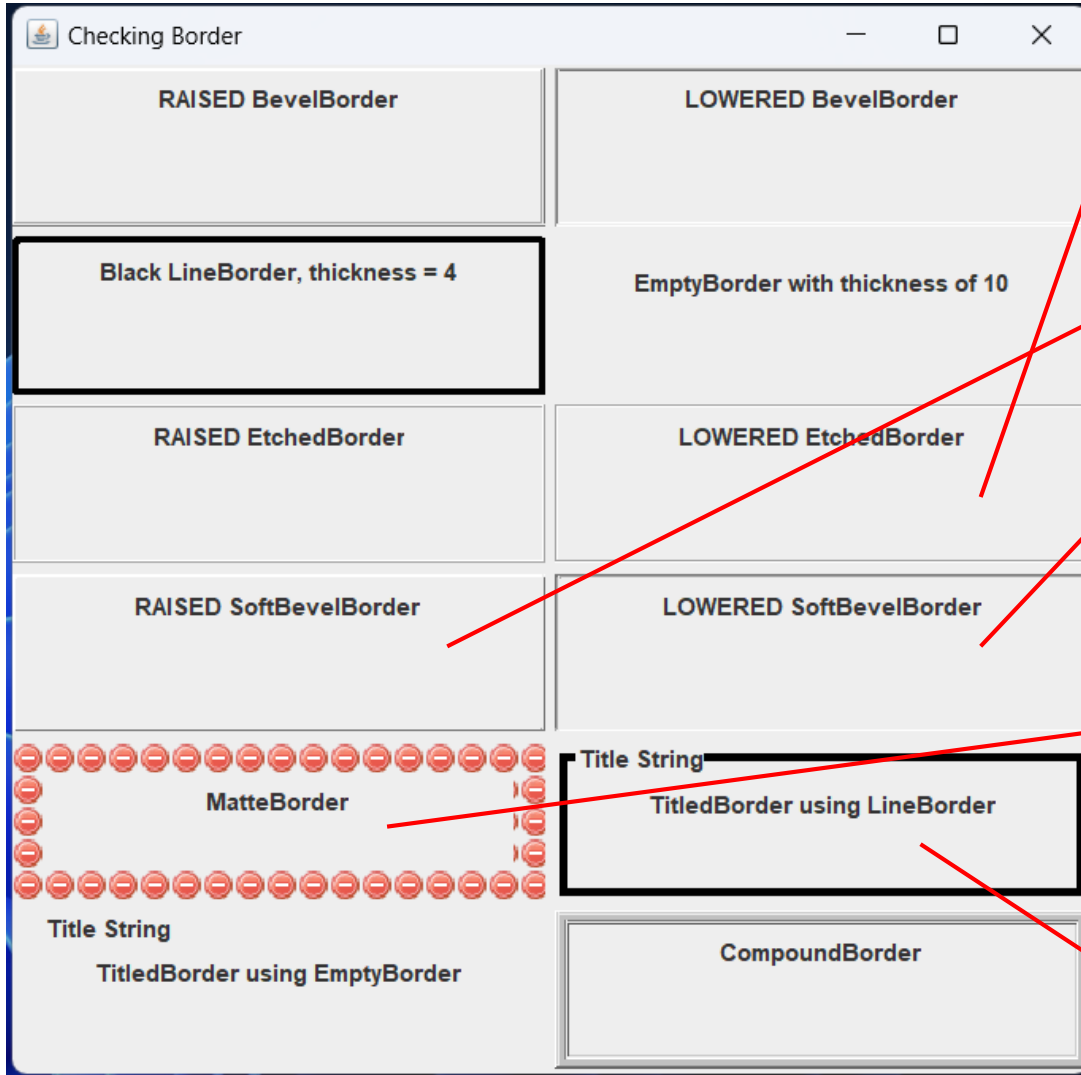
```
p = new JPanel();
p.setBorder(new BevelBorder(BevelBorder.LOWERED));
p.add(new JLabel("LOWERED BevelBorder"));
content.add(p);
```

```
p = new JPanel();
p.setBorder(new LineBorder(Color.black, 4, true));
p.add(new JLabel("Black LineBorder, thickness = 4"));
content.add(p);
```

```
p = new JPanel();
p.setBorder(new EmptyBorder(10, 10, 10, 10));
p.add(new JLabel("EmptyBorder with thickness of 10"));
content.add(p);
```

```
p = new JPanel();
p.setBorder(new EtchedBorder(EtchedBorder.RAISED));
p.add(new JLabel("RAISED EtchedBorder"));
content.add(p);
```

Sử dụng Simple Border



```
p = new JPanel();
p.setBorder(new EtchedBorder(EtchedBorder.LOWERED));
p.add(new JLabel("LOWERED EtchedBorder"));
content.add(p);

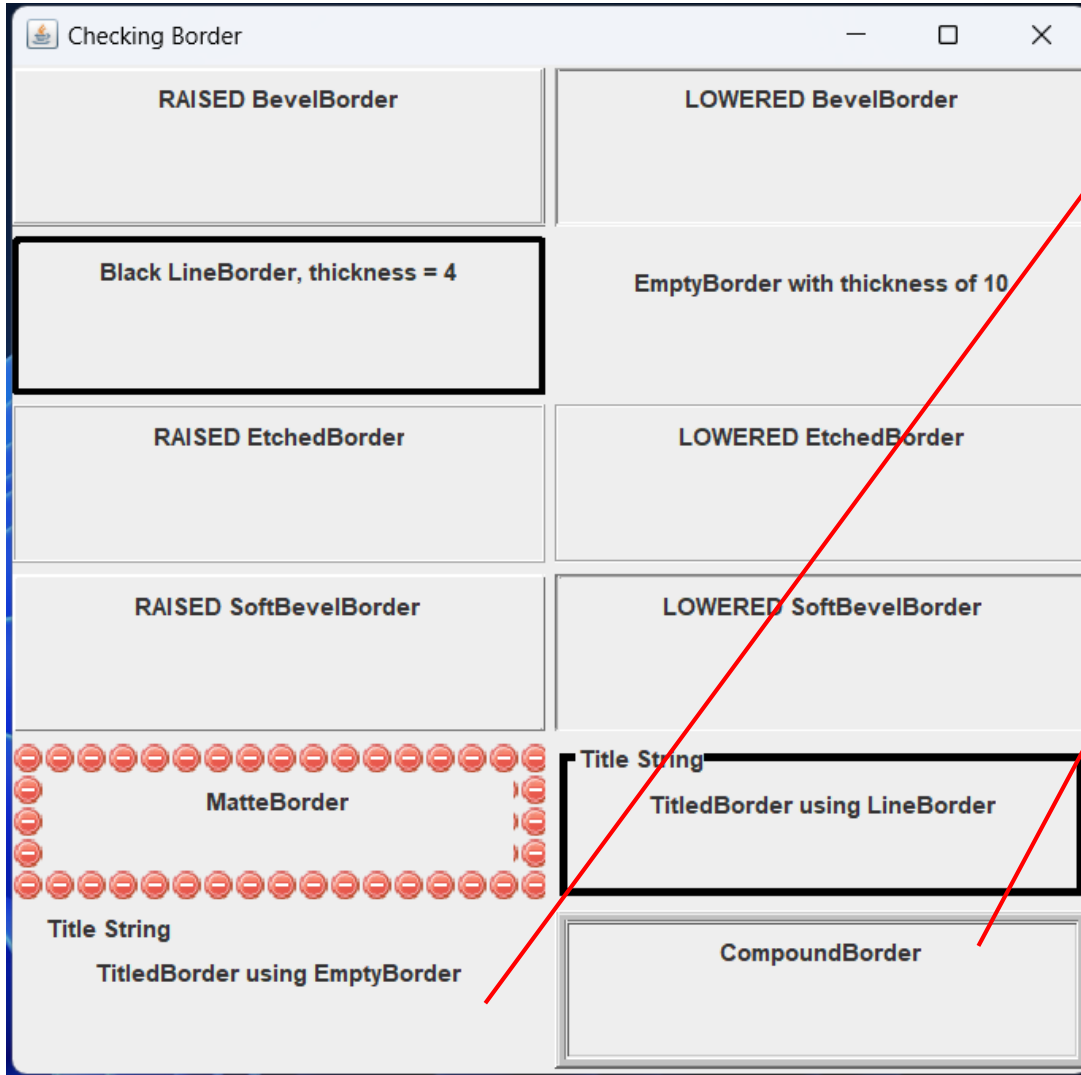
p = new JPanel();
p.setBorder(new SoftBevelBorder(SoftBevelBorder.RAISED));
p.add(new JLabel("RAISED SoftBevelBorder"));
content.add(p);

p = new JPanel();
p.setBorder(new SoftBevelBorder(SoftBevelBorder.LOWERED));
p.add(new JLabel("LOWERED SoftBevelBorder"));
content.add(p);

p = new JPanel();
p.setBorder(new MatteBorder(new ImageIcon("Hinh/btnXoa.png")));
p.add(new JLabel("MatteBorder"));
content.add(p);

p = new JPanel();
p.setBorder(new TitledBorder(new EmptyBorder(10, 10, 10, 10), "Title String"));
p.add(new JLabel("TitledBorder using EmptyBorder"));
content.add(p);
```

Sử dụng Simple Border



```
p = new JPanel();
p.setBorder(new TitledBorder(new EmptyBorder(10, 10, 10, 10), "Title String"));
p.add(new JLabel("TitledBorder using EmptyBorder"));
content.add(p);
```

//Declare Border Color

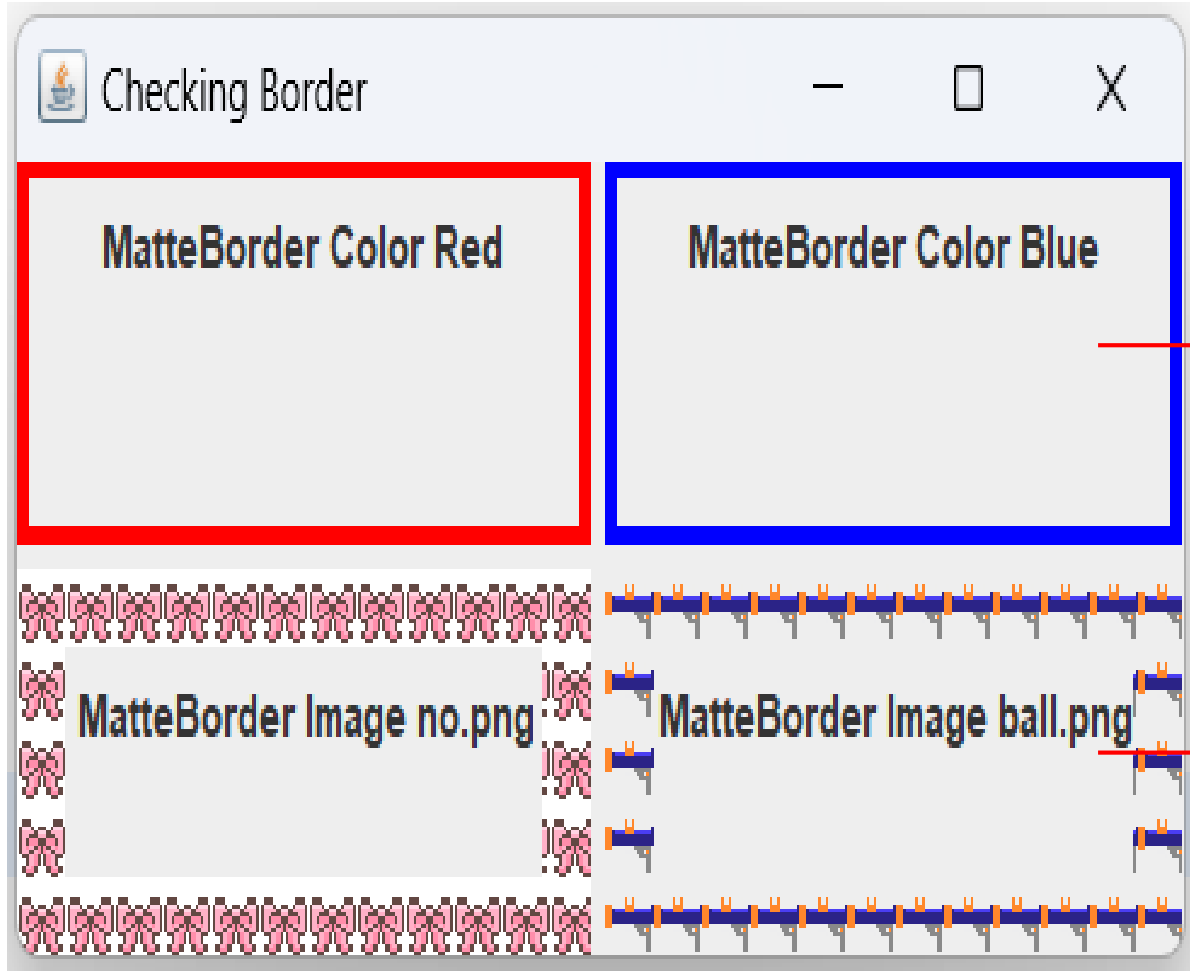
```
Color c1 = new Color(86, 86, 86);
Color c2 = new Color(192, 192, 192);
Color c3 = new Color(204, 204, 204);
```

```
Border b1 = new BevelBorder(EtchedBorder.RAISED, c3, c1);
Border b2 = new MatteBorder(3, 3, 3, 3, c2);
Border b3 = new BevelBorder(EtchedBorder.LOWERED, c3, c1);
```

//Create Panel Use Border Color

```
p = new JPanel();
p.setBorder(new CompoundBorder(new CompoundBorder(b1, b2), b3));
p.add(new JLabel("CompoundBorder"));
content.add(p);
```

Sử dụng Matte Border



```
p = new JPanel();  
p.setBorder(new MatteBorder(4, 4, 4, 4, Color.red));  
p.add(new JLabel("MatteBorder Color Red"));  
content.add(p);
```

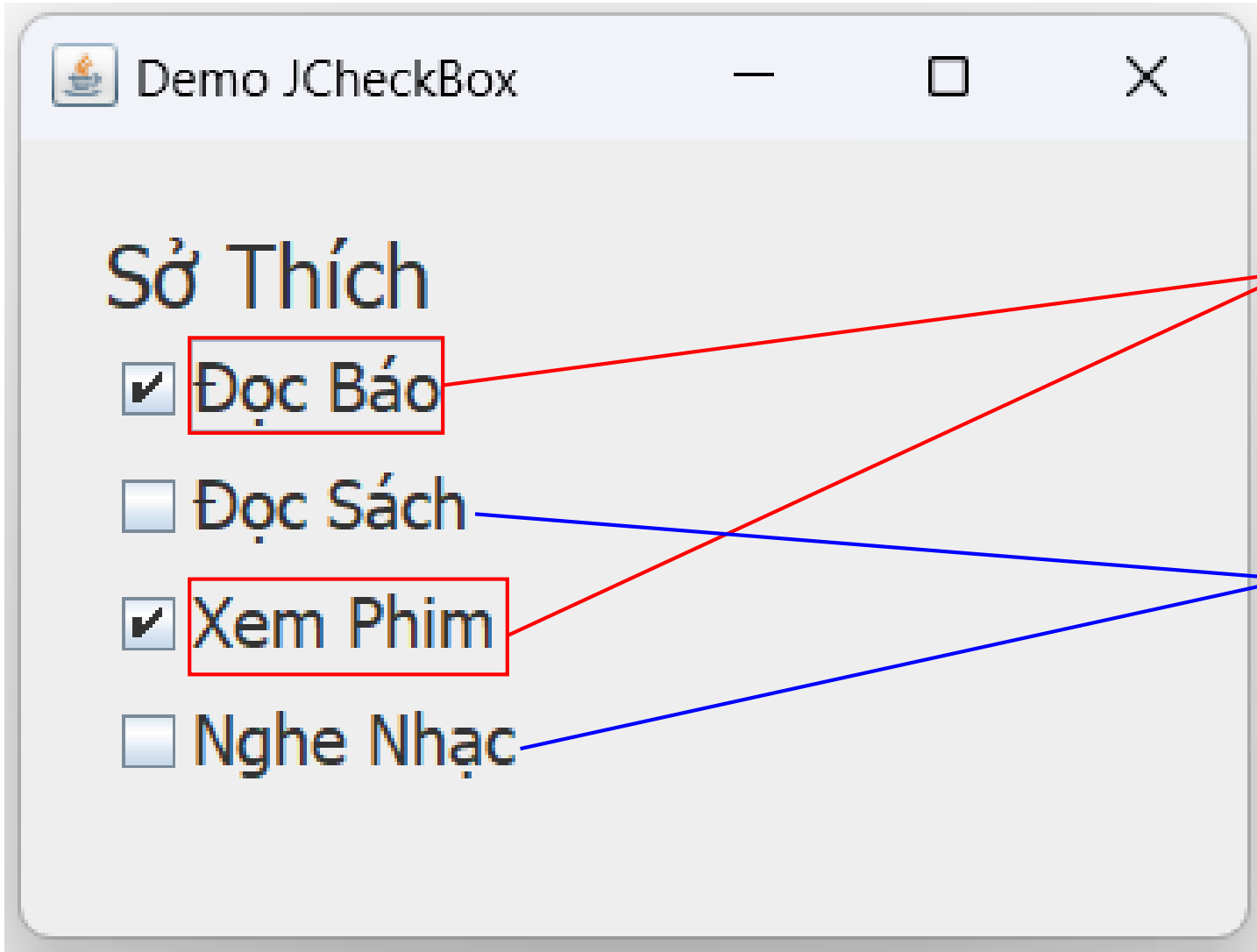
```
p = new JPanel();  
p.setBorder(new MatteBorder(new ImageIcon("images/no.png")));  
p.add(new JLabel("MatteBorder Image no.png"));  
content.add(p);
```

Sử dụng Title Border



```
JPanel p = new JPanel();
Border lineBorder = BorderFactory.createTitledBorder("title");
p.setBorder(lineBorder);
p.add(new JLabel("Default Title Border"));
panel.add(p);
p = new JPanel();
lineBorder = BorderFactory.createTitledBorder(new LineBorder(Color.CYAN), "title",
TitledBorder.RIGHT, TitledBorder.BELOW_TOP);
p.setBorder(lineBorder);
p.add(new JLabel("title border right just and below top"));
panel.add(p);
p = new JPanel();
lineBorder = BorderFactory.createTitledBorder(null, "title",
TitledBorder.DEFAULT_JUSTIFICATION, TitledBorder.BELOW_BOTTOM);
p.setBorder(lineBorder);
p.add(new JLabel("title border default just and below bottom"));
panel.add(p);
p = new JPanel();
lineBorder = BorderFactory.createTitledBorder(null, "title", TitledBorder.CENTER,
TitledBorder.ABOVE_TOP);
p.setBorder(lineBorder);
p.add(new JLabel("Title Border center Just and default position"));
panel.add(p);
```


JCheckBox



Trạng thái chọn
(checked)

Trạng thái không chọn
(unchecked)

JCheckBox

Tạo một đối tượng JCheckBox:

```
JCheckBox checkBox = new JCheckBox("Tùy chọn");
```

Tên biến do
người dùng tự đặt

Dòng chữ hiện ở
kế bên nút chọn (ô vuông)

Thiết lập văn bản và biểu tượng:

```
1 checkBox.setText("Tùy chọn"); // Thiết lập văn bản hiển thị trên JCheckBox
2 checkBox.setIcon(icon); // Thiết lập biểu tượng cho JCheckBox c
3 checkBox.setSelectedIcon(selectedIcon); // Thiết lập biểu tượng khi JCheckBox được chọn
4 checkBox.setDisabledIcon(disabledIcon); // Thiết lập biểu tượng khi JCheckBox bị vô hiệu hóa
```

JCheckBox

Thiết lập màu sắc:

```
1 checkBox.setForeground(Color.RED); // Thiết lập màu chữ  
2 checkBox.setBackground(Color.YELLOW); // Thiết lập màu nền
```

Thiết lập Font chữ:

```
checkBox.setFont(new Font("Arial", Font.BOLD, 12)); // Thiết lập font chữ
```

Thiết lập Viền:

```
checkBox.setBorder(BorderFactory.createLineBorder(Color.BLACK)); // Thiết lập viền đơn giản  
checkBox.setBorder(BorderFactory.createEtchedBorder()); // Thiết lập viền etsched  
checkBox.setBorder(BorderFactory.createRaisedBevelBorder()); // Thiết lập viền raised bevel  
checkBox.setBorder(BorderFactory.createLoweredBevelBorder()); // Thiết lập viền lowered bevel
```

JCheckBox

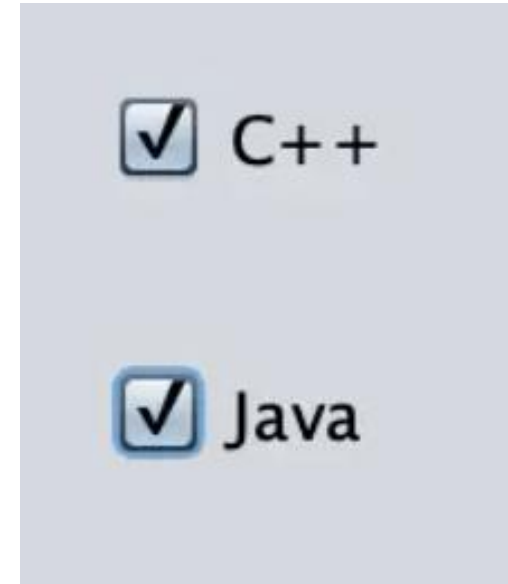
Xử lý sự kiện với JCheckBox:

```
1 JCheckBox checkBox = new JCheckBox("Tùy chọn");
2
3 checkBox.addItemListener(new ItemListener() {
4     @Override
5     public void itemStateChanged(ItemEvent e) {
6         if (e.getStateChange() == ItemEvent.SELECTED) {
7             // Đã chọn JCheckBox
8             System.out.println("JCheckBox đã được chọn");
9         } else {
10            // Đã bỏ chọn JCheckBox
11            System.out.println("JCheckBox đã được bỏ chọn");
12        }
13    }
14 });
```

JCheckBox

Ví dụ 01:

```
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();
    }
}
```



JCheckBox

Ví dụ 02:

```
import javax.swing.*;
import java.awt.event.*;
public class CheckBoxExample
{
    CheckBoxExample(){
        JFrame f= new JFrame("CheckBox Example");
        final 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);
        checkbox1.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {
                label.setText("C++ Checkbox: "
                    + (e.getStateChange()==1?"checked":"unchecked"));
            }
        });
        checkbox2.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {
                label.setText("Java Checkbox: "
                    + (e.getStateChange()==1?"checked":"unchecked"));
            }
        });
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
    }
    public static void main(String args[])
    {
        new CheckBoxExample();
    }
}
```

Java Checkbox: checked

☐ C++

☒ Java

JCheckBox

Ví dụ 03:

```
import javax.swing.*;
import java.awt.event.*;
public class CheckBoxExample extends JFrame implements ActionListener{
    JLabel l;
    JCheckBox cb1,cb2,cb3;
    JButton b;
    CheckBoxExample(){
        l=new JLabel("Food Ordering System");
        l.setBounds(50,50,300,20);
        cb1=new JCheckBox("Pizza @ 100");
        cb1.setBounds(100,100,150,20);
        cb2=new JCheckBox("Burger @ 30");
        cb2.setBounds(100,150,150,20);
        cb3=new JCheckBox("Tea @ 10");
        cb3.setBounds(100,200,150,20);
        b=new JButton("Order");
        b.setBounds(100,250,80,30);
        b.addActionListener(this);
        add(l);add(cb1);add(cb2);add(cb3);add(b);
        setSize(400,400);
        setLayout(null);
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

```
public void actionPerformed(ActionEvent e){
    float amount=0;
    String msg="";
    if(cb1.isSelected()){
        amount+=100;
        msg="Pizza: 100\n";
    }
    if(cb2.isSelected()){
        amount+=30;
        msg+="Burger: 30\n";
    }
    if(cb3.isSelected()){
        amount+=10;
        msg+="Tea: 10\n";
    }
    msg+="-----\n";
    JOptionPane.showMessageDialog(this,msg+"Total: "+amount);
}
public static void main(String[] args) {
    new CheckBoxExample();
}
}
```


Ví dụ 03:

Food Ordering System

☒ Pizza @ 100

☒ Burger @ 30

☒ Tea @ 10

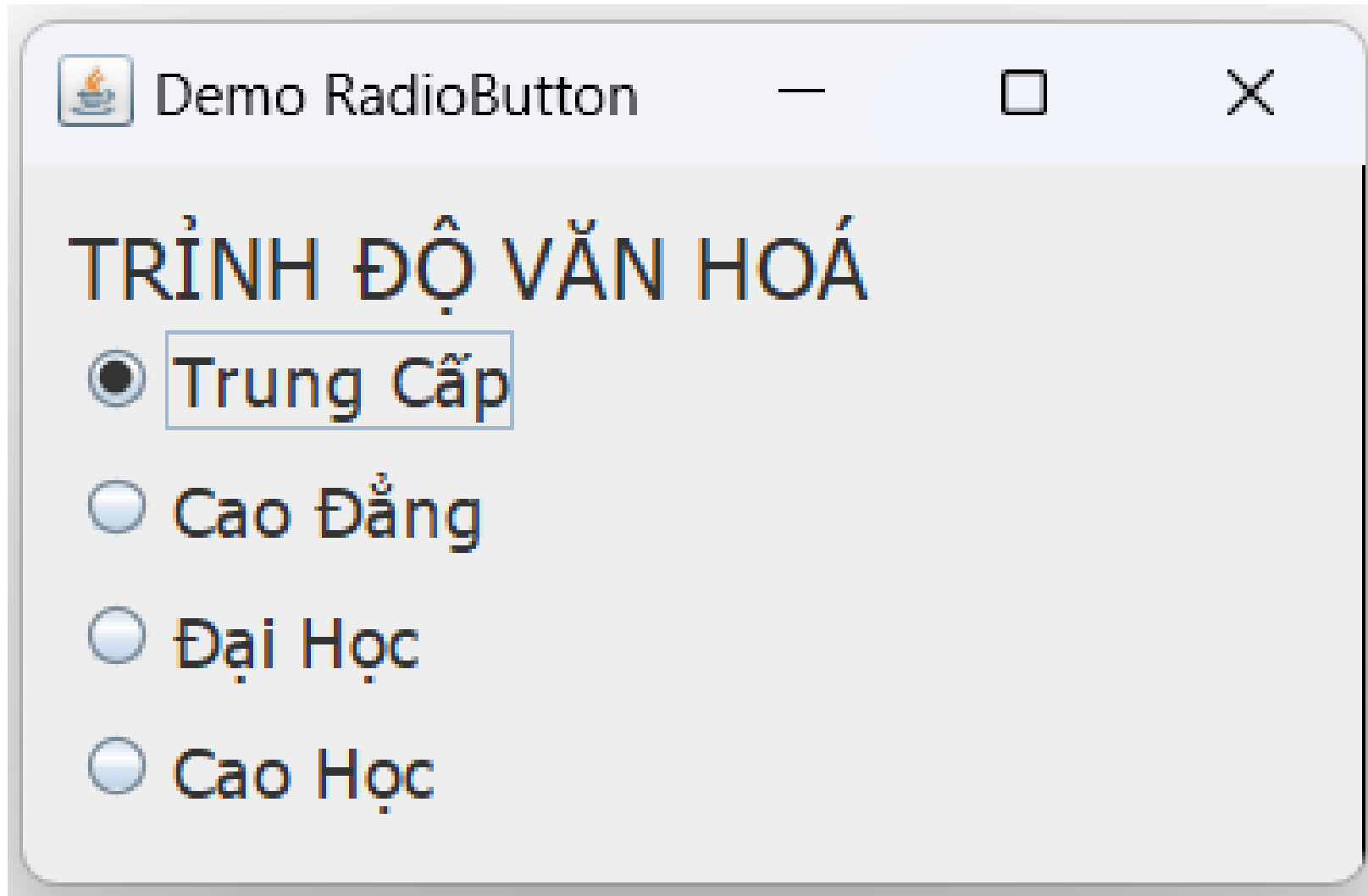
Message

 Pizza: 100
Burger: 30
Tea: 10

Total: 140.0

Ứng dụng của JCheckBox trong giao diện người dùng

JRadioButton



JRadioButton

Tạo các JRadioButton và thêm chúng vào ButtonGroup để quản lý:

```
JRadioButton radioButton1 = new JRadioButton("Option 1");  
JRadioButton radioButton2 = new JRadioButton("Option 2");  
JRadioButton radioButton3 = new JRadioButton("Option 3");  
ButtonGroup buttonGroup = new ButtonGroup();  
buttonGroup.add(radioButton1);  
buttonGroup.add(radioButton2);  
buttonGroup.add(radioButton3);
```

Tính năng của JRadioButton

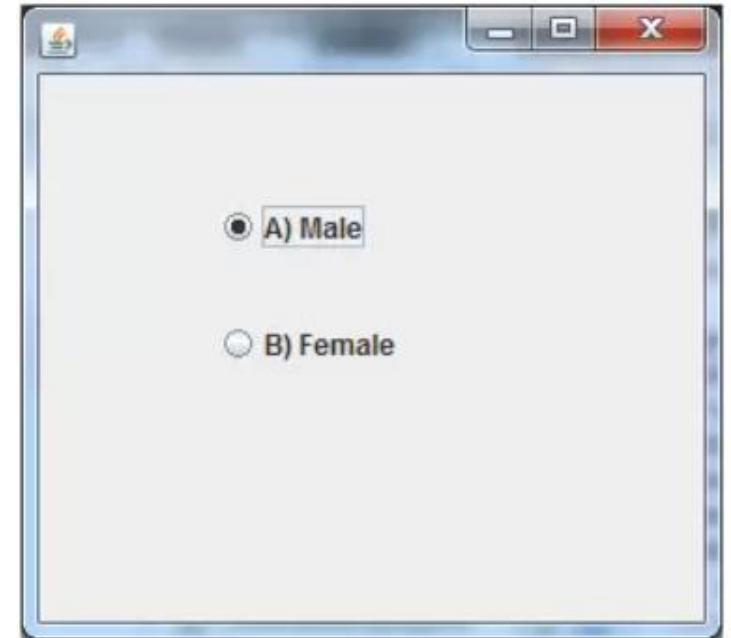
- **Lựa chọn đơn:** JRadioButton cho phép người dùng chọn một trong số các tùy chọn đơn.
- **Quản lý nhóm:** Các JRadioButton có thể được nhóm lại bằng cách sử dụng ButtonGroup để đảm bảo rằng chỉ một JRadioButton trong nhóm được chọn vào một thời điểm.
- **Trạng thái chọn:** JRadioButton có hai trạng thái chọn – chọn và không chọn. Chỉ một JRadioButton có thể được chọn trong một nhóm.
- **Sự kiện ItemListener:** JRadioButton cung cấp các sự kiện ItemListener để bắt các sự kiện thay đổi trạng thái của JRadioButton.
- **Tùy chỉnh giao diện:** JRadioButton cho phép bạn tùy chỉnh giao diện bằng cách thay đổi các thuộc tính như vị trí, kích thước, màu sắc, font chữ và hình ảnh.

Ứng dụng của JRadioButton trong giao diện người dùng

JRadioButton

Ví dụ 01

```
import javax.swing.*;
public class RadioButtonExample {
    JFrame f;
    RadioButtonExample(){
        f=new JFrame();
        JRadioButton r1=new JRadioButton("A) Male");
        JRadioButton r2=new JRadioButton("B) Female");
        r1.setBounds(75,50,100,30);
        r2.setBounds(75,100,100,30);
        ButtonGroup bg=new ButtonGroup();
        bg.add(r1);bg.add(r2);
        f.add(r1);f.add(r2);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
    public static void main(String[] args) {
        new RadioButtonExample();
    }
}
```

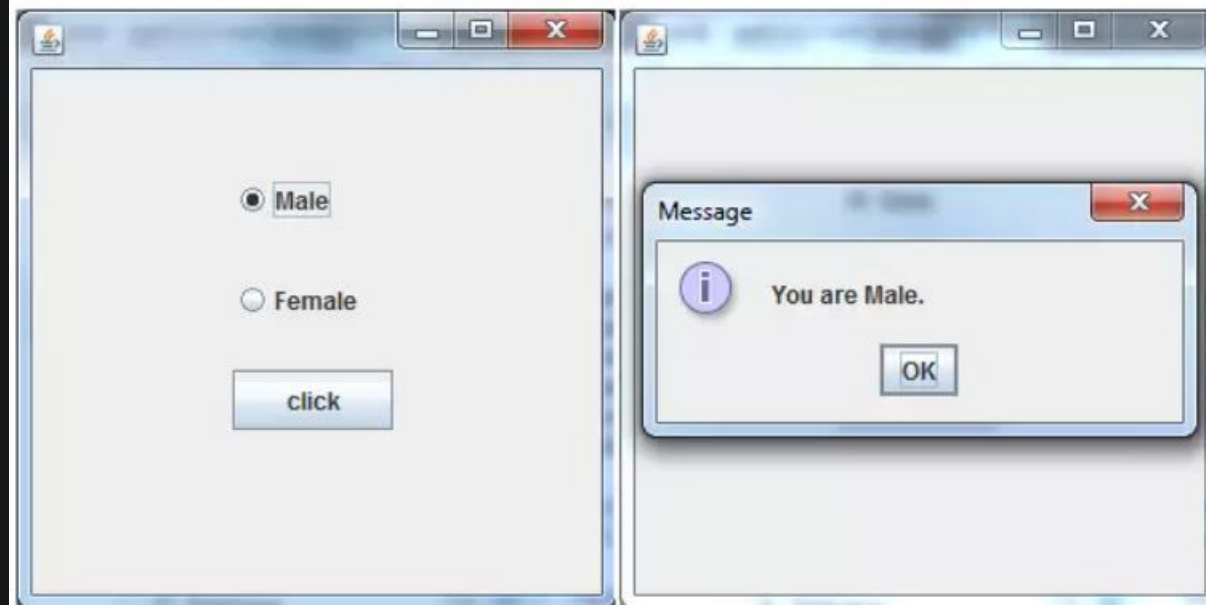


JRadioButton

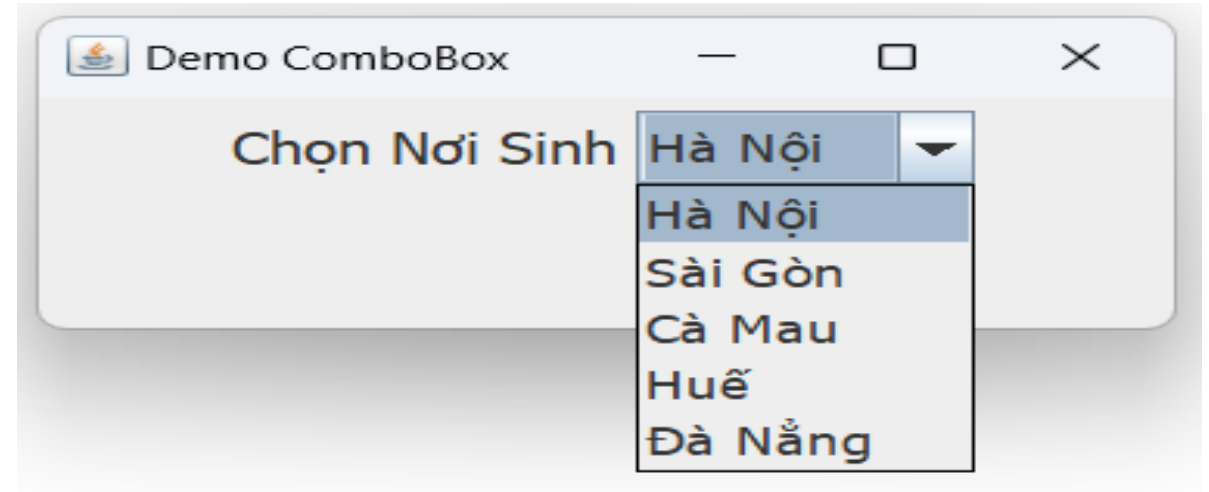
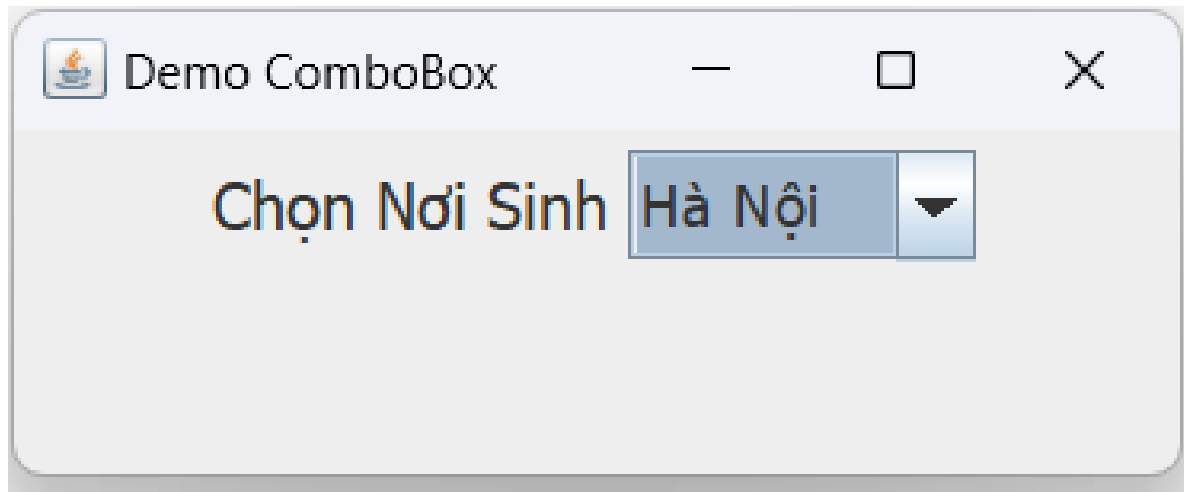
Ví dụ 02

```
import javax.swing.*;
import java.awt.event.*;
class RadioButtonExample extends JFrame implements ActionListener{
    JRadioButton rb1,rb2;
    JButton b;
    RadioButtonExample(){
        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);
        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);
    }
}
```

```
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 RadioButtonExample();
}}
```



JComboBox



Cách tạo JComboBox

```
String[] items = {"Item 1", "Item 2", "Item 3"};  
JComboBox<String> comboBox = new JComboBox<>(items);
```

Cấu hình JComboBox

```
comboBox.addItem("Item 4"); // Thêm mục lựa chọn  
comboBox.setSelectedIndex(2); // Thiết lập mục được chọn mặc định là "Item 3"  
comboBox.setBounds(50, 50, 100, 20); // Thiết lập vị trí và kích thước
```

Thêm 1 Mục vào JComboBox

```
JComboBox<String> comboBox = new JComboBox<>();  
comboBox.addItem("Mục 1");
```

JComboBox

Thêm nhiều Mục từ mảng hoặc Collection vào JComboBox

```
String[] items = {"Mục 1", "Mục 2", "Mục 3"};
```

```
JComboBox<String> comboBox = new JComboBox<>(items);
```

```
// Hoặc sử dụng Collection
```

```
List<String> itemList = Arrays.asList("Mục 1", "Mục 2", "Mục 3");
```

```
comboBox.addAll(itemList);
```

Thiết Lập Mục Lựa Chọn Mặc Định

```
comboBox.setSelectedItem("Mục 2"); // Chọn mục theo giá trị  
comboBox.setSelectedIndex(1); // Chọn mục theo chỉ số
```

Thay Đổi Font và Màu Sắc:

```
comboBox.setFont(new Font("Arial", Font.BOLD, 14));  
comboBox.setForeground(Color.BLUE);
```

Tạo JComboBox trống

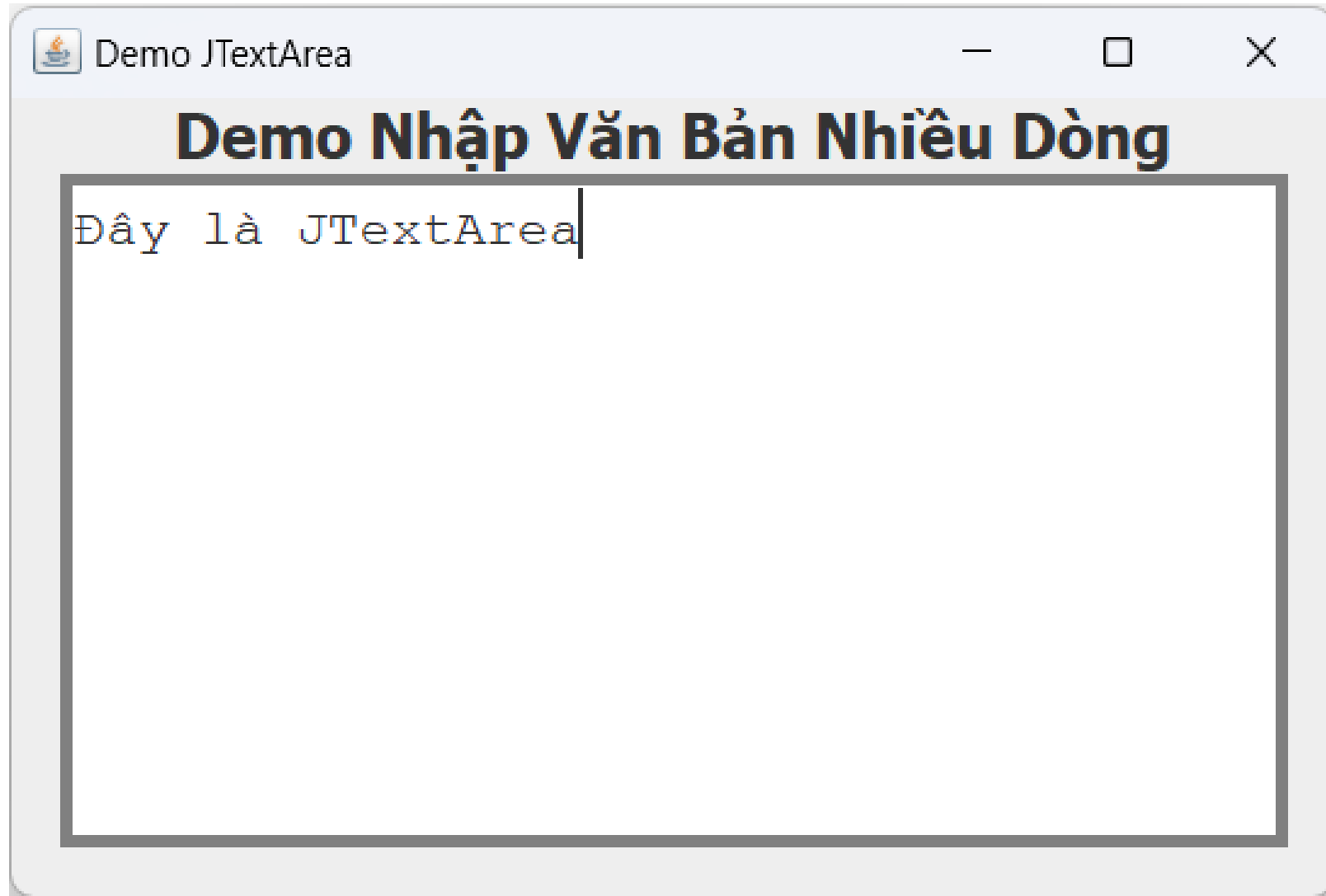
```
JComboBox<String> comboBox = new JComboBox<>();
```

Sử dụng ActionListener để xử lý sự kiện trong JComboBox

```
JComboBox<String> comboBox = new JComboBox<>(new String[]{"Mục 1", "Mục 2", "Mục 3"});  
comboBox.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        String selected = (String) comboBox.getSelectedItem();  
        System.out.println("Mục được chọn: " + selected);  
    }  
});
```

Sử dụng ItemListener để xử lý sự kiện trong JComboBox

```
comboBox.addItemListener(new ItemListener() {  
    @Override  
    public void itemStateChanged(ItemEvent e) {  
        if (e.getStateChange() == ItemEvent.SELECTED) {  
            String selected = (String) e.getItem();  
            System.out.println("Mục được chọn: " + selected);  
        }  
    }  
});
```



JTextArea

Tạo một đối tượng và Đặt kích thước và vị trí của JTextArea

```
JTextArea textArea = new JTextArea();  
textArea.setBounds(x, y, width, height);
```

Đặt và lấy nội dung của JTextArea

```
textArea.setText("Nội dung văn bản");  
String content = textArea.getText();
```

JTextArea

Thiết lập thuộc tính font, màu sắc, kích thước, và căn chỉnh

```
textArea.setFont(new Font("Arial", Font.PLAIN, 12));  
textArea.setForeground(Color.BLUE);  
textArea.setBackground(Color.WHITE);  
textArea.setAlignmentX(Component.CENTER_ALIGNMENT);
```

Thiết lập thanh cuộn cho textarea

```
JScrollPane scrollPane = new JScrollPane(textArea);  
scrollPane.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);  
scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);  
container.add(scrollPane);
```

Ứng dụng thực tế của JTextArea trong Java Swing

JScrollPane

Để sử dụng JScrollPane, ta code như sau:

```
JTextArea textArea = new JTextArea(20, 20);
JScrollPane scrollableTextArea = new JScrollPane(textArea);

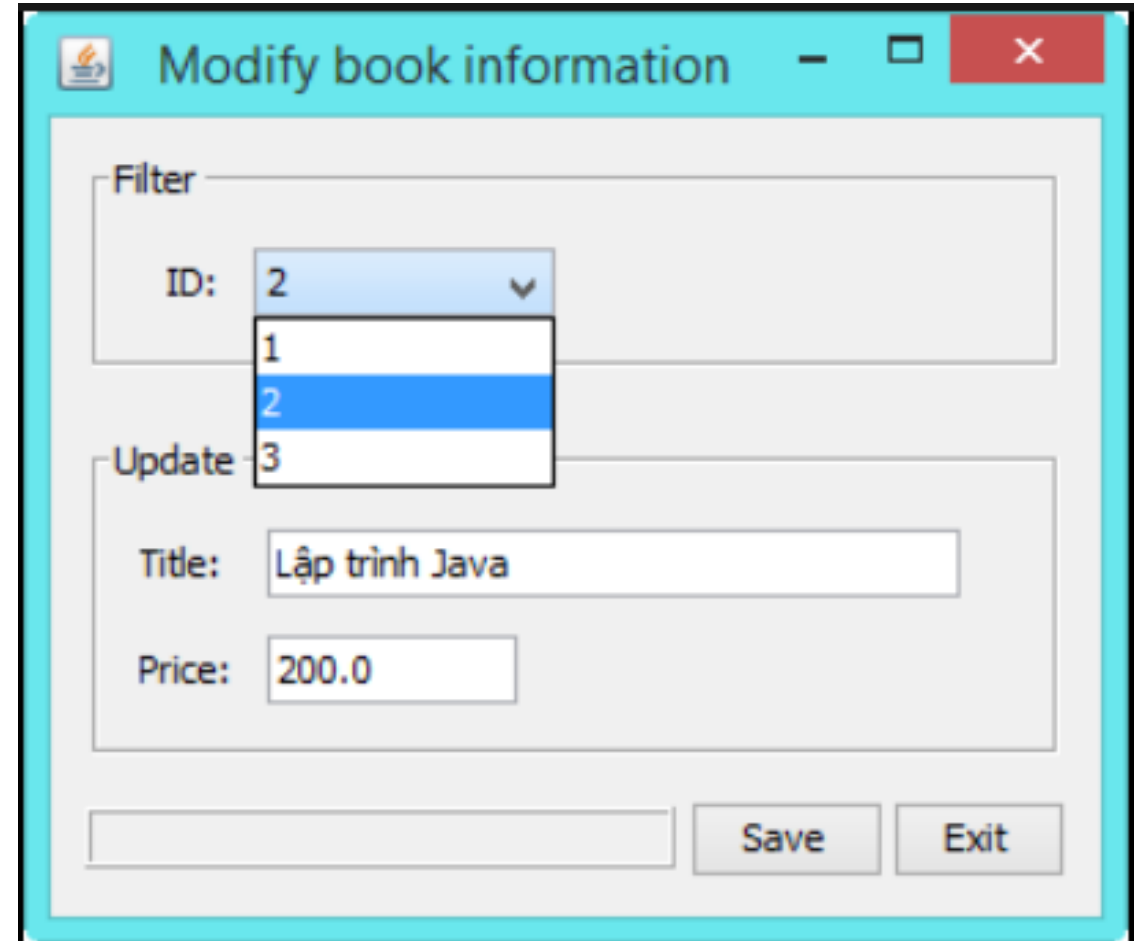
scrollableTextArea.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
scrollableTextArea.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);

frame.getContentPane().add(scrollableTextArea);
```

**Ứng dụng của JScrollPane
trong Java Swing**

Câu hỏi ôn tập

- 1) Trình bày điểm khác nhau của JCheckBox và JRadioButton ?
- 2) Trình bày ứng dụng thực tế của JCheckBox, JRadioButton và JTextArea trong lập trình giao diện Java Swing.
- 3) Viết code Java cho giao diện sau đây:



Modify book information

Filter

ID: 2

1
2
3

Update

Title: Lập trình Java

Price: 200.0

Save Exit