

Bài 4

List và Pane

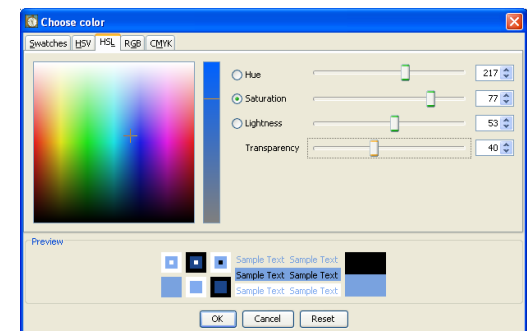
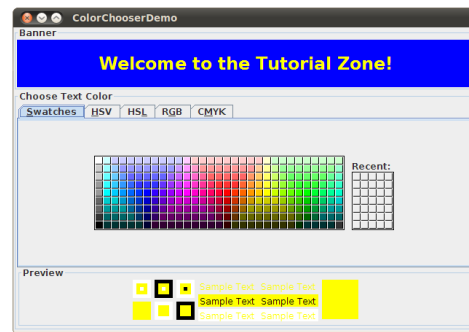
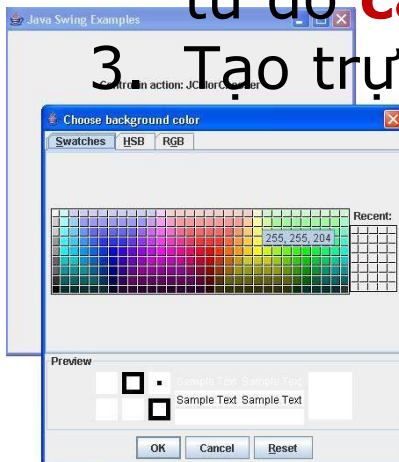
Mục tiêu

- Giới thiệu JColorChooser
- Giới thiệu Jlist
- Giới thiệu JComboBox
- Giới thiệu JTabbedPane
- Giới thiệu JSplitPane



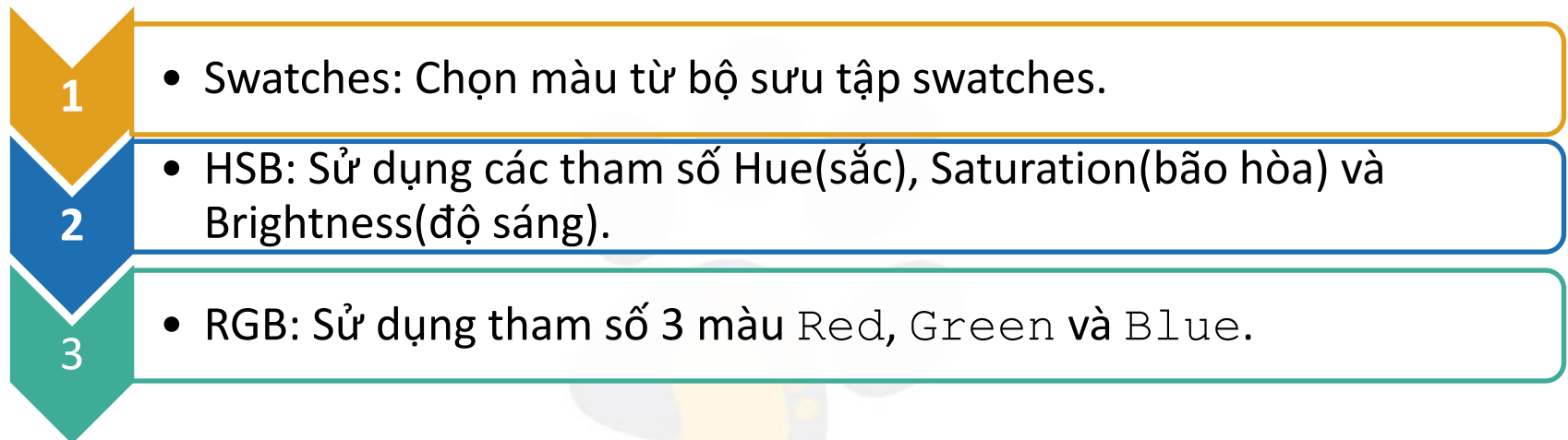
JColorChooser

- Đây là control cho phép lựa chọn **màu sắc** được xây dựng sẵn trong **Swing**.
- JColorChooser cung cấp 3 phương pháp để lấy màu sắc:
 1. Sử dụng phương thức tĩnh để gọi một cửa sổ dialog và **trả về màu đã chọn**.
 2. Sử dụng phương thức tĩnh để tạo cửa sổ dialog và từ đó **cài đặt màu**.
 3. Tạo trực tiếp và **đặt vào** trong một container.



JColorChooser

- Control này cung cấp 3 tab chọn màu:



- JColorChooser có 2 constructor:

- `public JColorChooser()`
- `public JColorChooser(Color initialColor)`

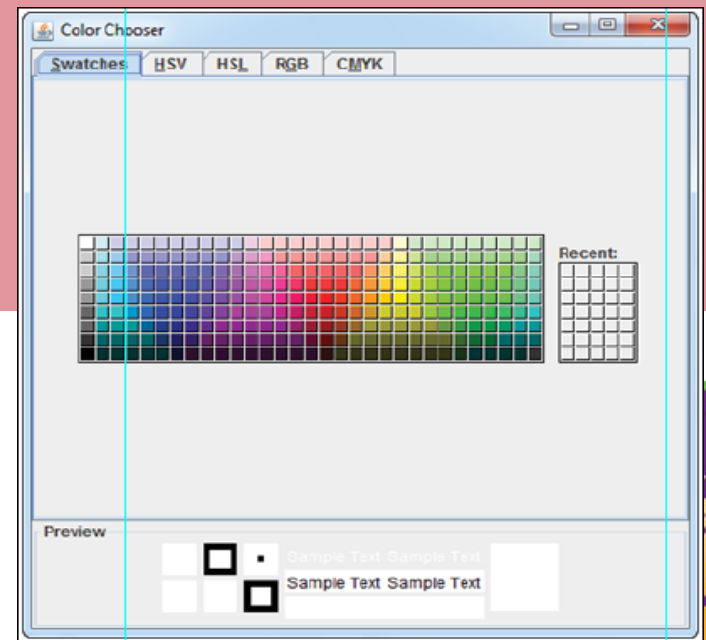


JColorChooser

Demo:

Code Snippet

```
JFrame frmColorChooser;  
JColorChooser ccrChoser;  
// Creates a frame with the title "Color Chooser"  
frmColorChooser = new JFrame("Color Chooser");  
// Creates a color chooser  
ccrChoser = new JColorChooser();  
// Adds the color chooser to the frame  
frmColorChooser.getContentPane().add(ccrChoser);
```



JColorChooser

Quản lý sự kiện lựa chọn màu sắc:

1. Nhận đối tượng **ColorSelectionModel** từ phương thức **getSelectionModel()** của JColorChooser.
2. Sử dụng đối tượng trên đăng ký **addChangeListener()**, phương thức này nhận vào một đối tượng **ChangeListener**.

Code Snippet

```
// Declare the chooser and model
JColorChooser ccrChoser;
ColorSelectionModel colorSelectionModel;
// Retrieve the color selection model
colorSelectionModel = ccrChoser.getSelectionModel();
// Register a change listener with the selection model
colorSelectionModel.addChangeListener(new ChangeListener() { . . . });
```

JColorChooser

Demo:

Code Snippet

```
// Declare a chooser and a label
JLabel lblMessage;
// Create and add the ccrChoser and lblMessage
...
// Retrieve the color selection model
colorSelectionModel = ccrChoser.getSelectionModel();
// Add a change listener to the color selection model
colorSelectionModel.addChangeListener(new ChangeListener() {
    public void stateChanged(ChangeEvent e) {
        Color clr = ccrChoser.getColor();
        lblMessage.setBackground(clr);
    }
});
```



JColorChooser

- Trả về màu hiện tại và cài đặt màu:

Code Snippet

```
JColorChooser ccrChooser;  
Color clrColor;  
// Creates a color chooser  
ccrChooser = new JColorChooser();  
// Retrieves the current color selected  
clrColor = ccrChooser.getColor();
```

Code Snippet

```
JColorChooser ccrChooser;  
// Creates a color chooser  
ccrChooser = new JColorChooser();  
// Sets blue as the current color  
ccrChooser.setColor(Color.blue);
```

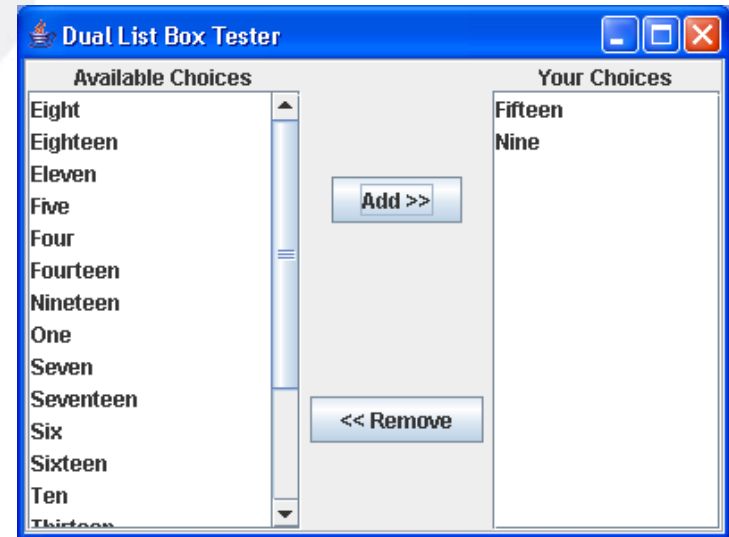

JList

- Jlist là lớp cho phép hiển thị một **nhóm** các item.
- Cho phép chọn **một hoặc nhiều** item.
- Có thể hiển thị các item thành một hoặc nhiều **cột**.
- Jlist **không hỗ trợ cuộn**, chức năng này được cung cấp thông qua JScrollPane.
- Jlist có 3 constructor:

1. `public JList()`

2. `public JList(Object[] items)`

3. `public JList(Vector vecItems)`



JList

■ Tạo Jlist:

1

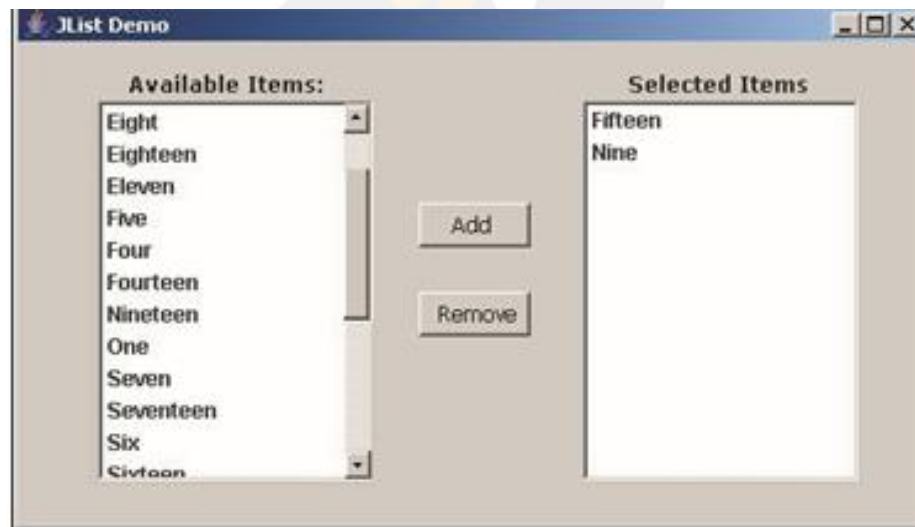
- Khởi tạo JList.

2

- Tạo JScrollPane và đặt nó vào như tham số.

3

- Thêm JScrollPane ở trên vào trong container.



JList

■ Demo (với mảng):

Code Snippet

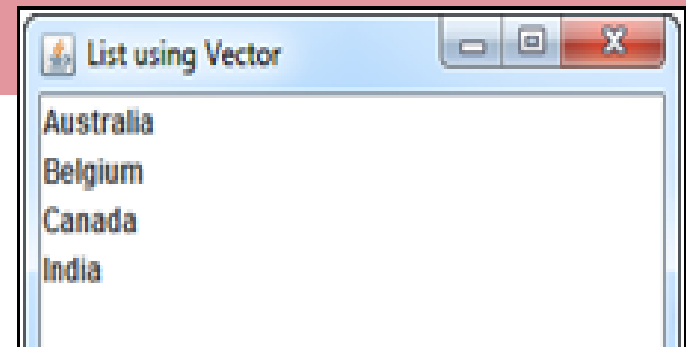
```
JPanel pnlCountries;  
JList lstCountries;  
JScrollPane scpScroller;  
String[] strCountries;  
...  
// Create a string array and initialize it the names of the countries  
String[] strCountries = {"Australia", "Belgium", "Canada", .. "India"};  
// Create a list with the string array  
lstCountries = new JList(strCountries);  
// Provide the list to the scrollpane to facilitate scrolling  
scpScroller = new JScrollPane(lstCountries);  
// Add the JScrollPane to the panel  
pnlCountries.add(scpScroller);
```

JList

■ Demo (với vector):

Code Snippet

```
// Creates a vector to store the names of the countries Vector
vecCountries = new vector();
// Adds the names of the countries to the vector
vecCountries.addElement("Australia");
vecCountries.addElement("Belgium");
vecCountries.addElement("Canada");
...
vecCountries.addElement("India");
// Creates a list with a Vector
JList lstCountries = new JList(vecCountries);
```



JList

■ Demo (chèn vào JScrollPane):

Code Snippet

```
JPanel pnlCountries;  
JList lstCountries;  
JScrollPane scpScroller;  
String[] strCountries;  
...  
// Creates string array and initialize it with countries name  
String[] strCountries = {"Australia", "Belgium", "Canada", ... "India"};  
// Creates a list with the string array  
lstCountries = new JList(strCountries);  
// Provides the list to the scrollpane to facilitate scrolling scpScroller = new  
ScrollPane(lstCountries);  
// Adds the JScrollPane to the panel pnlCountries.add(scpScroller);
```

JList

- Xử lý sự kiện chọn item:

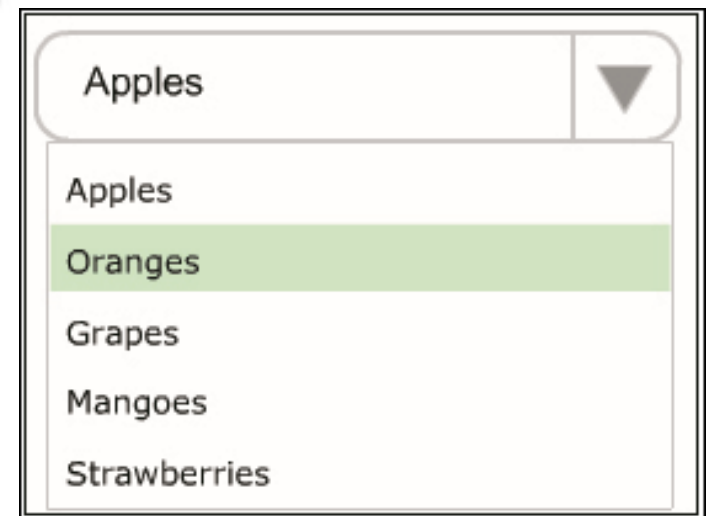
Code Snippet

```
JList lstCountries;  
...  
lstCountries.addListSelectionListener(new ListSelectionListener() {  
    public void valueChanged(ListSelectionEvent e) {  
        // Add the action code  
        ...  
        ...  
    }  
});
```



JComboBox

- Là tổ hợp giữa danh sách thả và văn bản. Item được chọn sẽ hiển thị mặc định, các item khác chỉ hiển thị khi mở danh sách thả.
- Control này chỉ cho phép lựa chọn một item không như Jlist (có thể chọn nhiều).
- Các constructor:
 - `public JComboBox()`
 - `public JComboBox(Object[] items)`
 - `public JComboBox(Vector items)`



JComboBox

■ Các bước tạo:

1

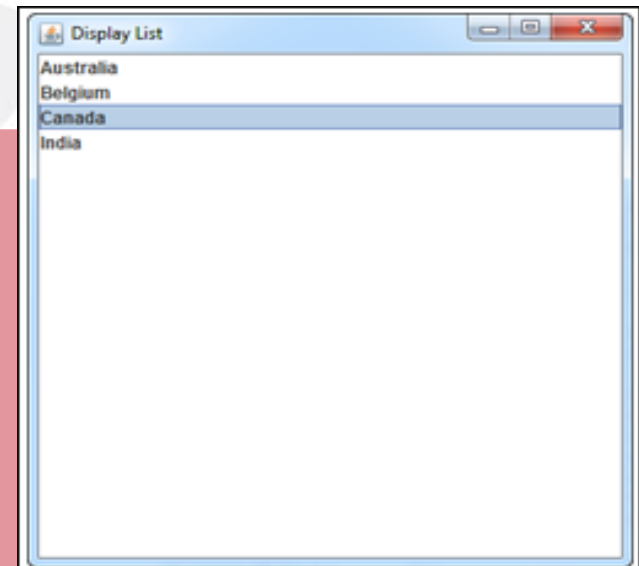
- Tạo mới JComboBox.

2

- Thêm vào container bằng cách sử dụng phương thức add().

Code Snippet

```
// Creates a panel
JPanel pnlFonts = new JPanel();
// Creates a string array with 10 elements
String[] strFonts = new String[10];
// Initialize the array with font names
...
// Creates a combo box with the items from string array
JComboBox cboFonts = new JComboBox(strFonts);
// Enables the inputting of text
cboFonts.setEditable(true);
// Adds the combo box to the panel
pnlFonts.add(cboFonts);
```



JComboBox

■ Xử lý sự kiện:

Code Snippet

```
JComboBox cboFonts;  
...  
// Register the ActionListener  
cboFonts.addActionListener(new ActionListener()  
{  
    // Handle the action performed event  
    public void actionPerformed(ActionEvent e)  
    {  
        // Add the code to handle the event  
    }  
});
```

JComboBox

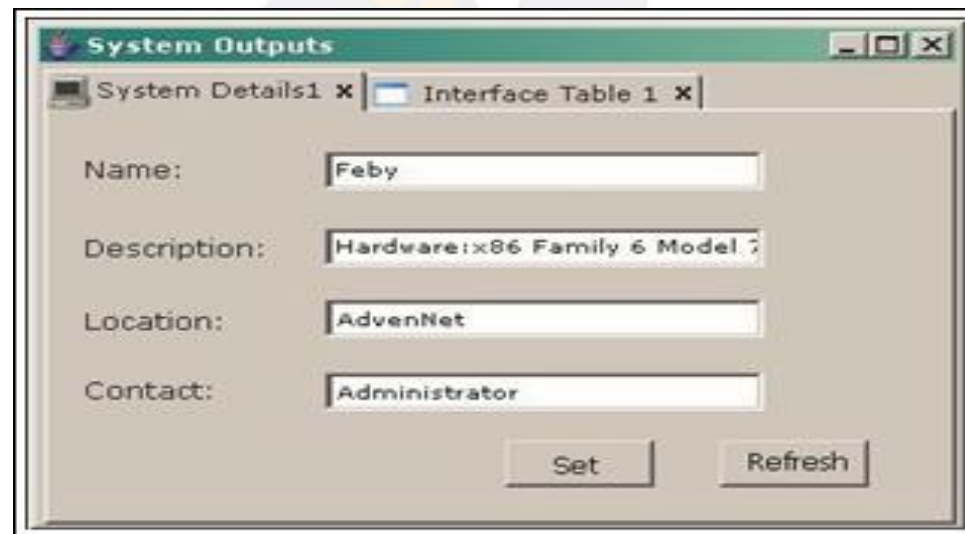
■ Xử lý sự kiện:

Code Snippet

```
JComboBox cboFonts = new JComboBox(strFonts);  
// Register the ItemListener  
cboFonts.addItemListener(new ItemListener() { // Handle the item state changed  
    event  
    public void itemStateChanged(ItemEvent e) {  
        // Check if it a Select or Deselect event  
        if (e.getStateChange() == ItemEvent.SELECTED) {  
            // SELECT event  
            ...  
        } else {  
            // DESELECT event  
            ... } } });
```

JTabbedPane

- Là control cho phép đặt các component tương tự nhau (như panels) trên vùng không gian chia sẻ.
- Mỗi panel được thêm vào JTabbedPane như là một tab. Tab này có thể thêm văn bản và icon.
- Khi tab nào được click thì panel tương ứng sẽ được chọn và hiển thị.



JTabbedPane

■ Các bước tạo:

1

- Khởi tạo đối tượng `JTabbedPane`.

2

- Tạo các `JPanel` cho mỗi tab.

3

- Thêm các component vào panels.

4

- Sử dụng phương thức `addTab()` của `JTabbedPane` để thêm panel.



JTabbedPane

■ Demo:

Code Snippet

```
Jframe frmSubjects;  
JPanel pnlEnglish;  
JTabbedPane tpnSubjects;  
...  
// Creates a frame with the title "Subjects"  
    frmSubjects = new JFrame("Subjects");  
// Create an image icon  
    ImageIcon iconEnglish = new ImageIcon("English.gif");  
// Adds panel having the title "English", icon and tooltip to // tabbedpane  
    tpnSubjects.addTab("English",iconEnglish,pnlEnglish, "English Subject");  
  
// Adds the tabbedpane to the frame  
    frmSubjects.getContentPane().add(tpnSubjects);
```

JTabbedPane

■ Xử lý sự kiện:

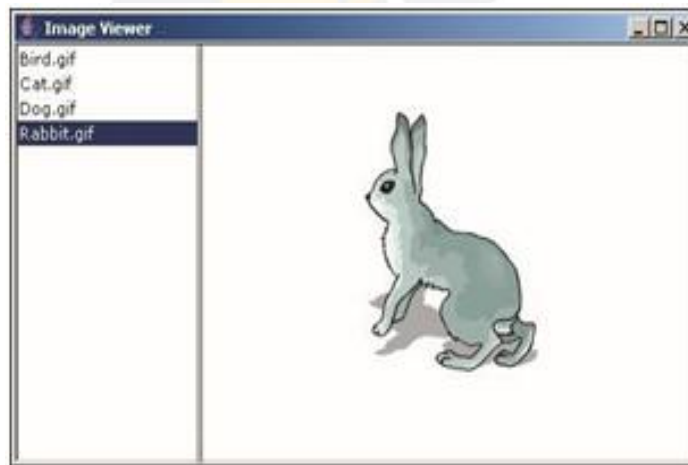
Code Snippet

```
JTabbedPane tpnSubjects;  
...  
// Register the ChangeListener  
tpnSubjects.addChangeListener(new ChangeListener() {  
    // Handle the state changed event  
    public void stateChanged(ChangeEvent e) {  
        // Add the action code  
        ...  
    }  
});
```



JSplitPane

- Là control cho phép hiển thị 2 phân vùng được phân chia.
- 2 phân vùng có thể chia theo chiều dọc hoặc ngang.
- Phần chia cách có thể kéo để phân chia lại diện tích giữa 2 vùng.
- Ưu điểm của control này là khả năng cho phép thay đổi yêu cầu về kích thước mỗi vùng.



JSplitPane

■ Demo:

Code Snippet

```
JFrame frmExplorer;  
JSplitPane spnExplorer;  
// Creates a frame  
frmExplorer = new JFrame("Explorer");  
// Creates a splitpane  
spnExplorer = new JSplitPane();  
// Adds the splitpane to the frame  
frmExplorer.getContentPane().add(spnExplorer);
```

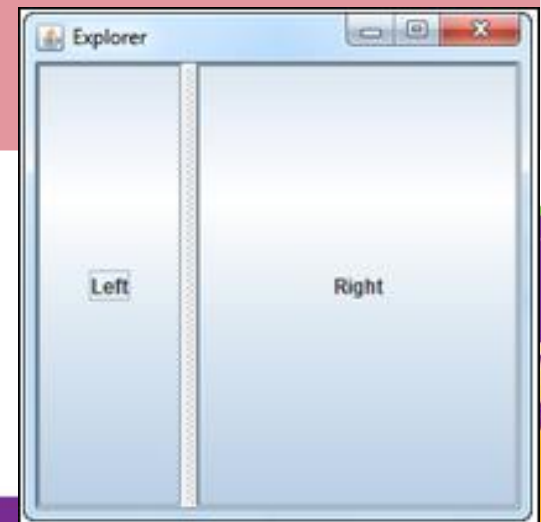
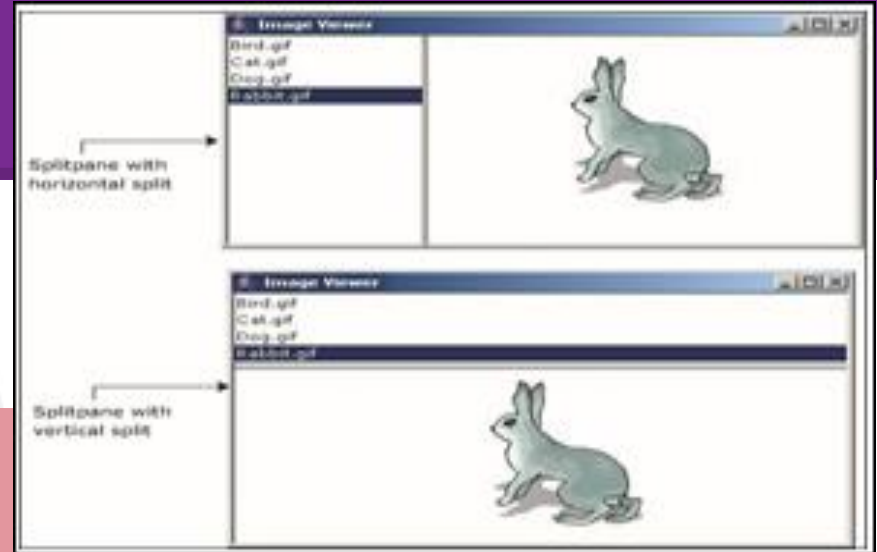


JSplitPane

■ Demo:

Code Snippet

```
JSplitPane spnExplorer;  
JPanel pnlLeft, pnlRight;  
// Create a frame  
frmExplorer = new JFrame("Explorer");  
// Create a splitpane with the given orientation and continuous  
// layout. spnExplorer = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,true,  
pnlLeft,pnlRight);  
// Add the splitpane to the frame  
frmExplorer.getContentPane().add(spnExplorer);
```



Tóm tắt bài học

- ✓ **JColorChooser** là control cho phép lựa chọn màu sắc theo nhiều cách khác nhau.
- ✓ **JList** là component hiển thị tập hợp các item, danh sách các item có thể sắp xếp theo nhiều cột.
- ✓ **JComboBox** là thành phần phức hợp của danh sách thả và văn bản.
- ✓ **JTabbedPane** cho phép đặt nhiều panel trong cùng không gian chia sẻ.
- ✓ **JSplitPane** chia không gian làm 2 vùng riêng biệt, phần phân cách có thể kéo để phân bố lại diện tích kích thước.



HẾT
XIN CẢM ƠN!

