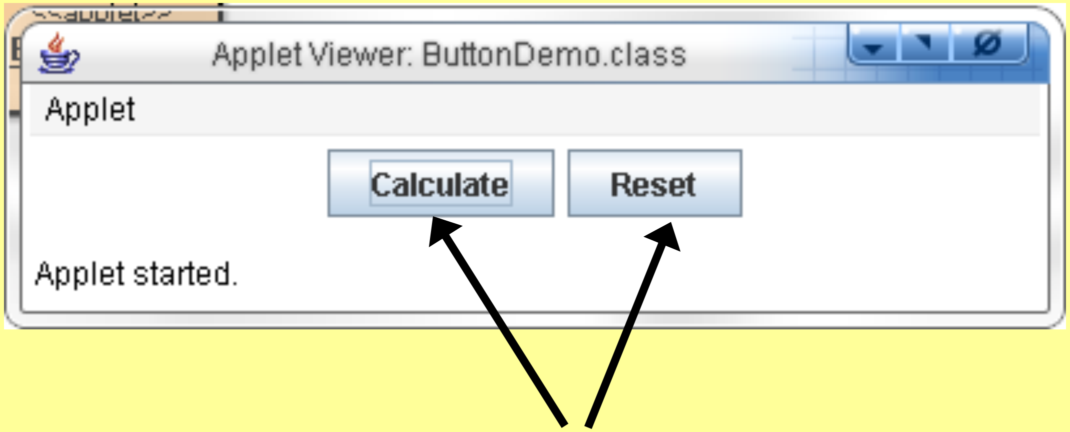
GUI Programming 102
Swing

INTRODUCTION

- The following Swing components will be described:
  - buttons (`JButton`)
  - labels (`JLabel`)
  - text fields (`JTextField`)
  - password fields (`JPasswordField`)
  - checkboxes (`JCheckBox`)
  - radio buttons (`JRadioButton`)
  - lists (`JList`)
  - combo boxes (`JCombobox`)
  - text areas (`JTextArea`)
  - sliders (`JSlider`)

ButtonDemo DEMO APPLET

- *ButtonDemo*‘s user interface:



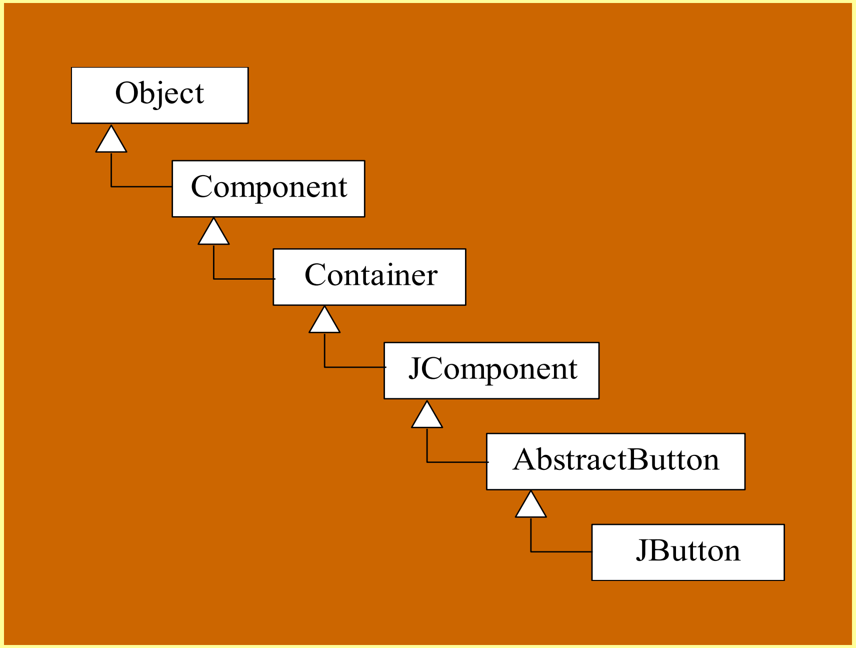
Applet Viewer: ButtonDemo.class

Applet started.

Buttons

BUTTONS

- Buttons can be represented as `JButton` objects.



- A way to create a `JButton` object:

```
public JButton(String s)
```

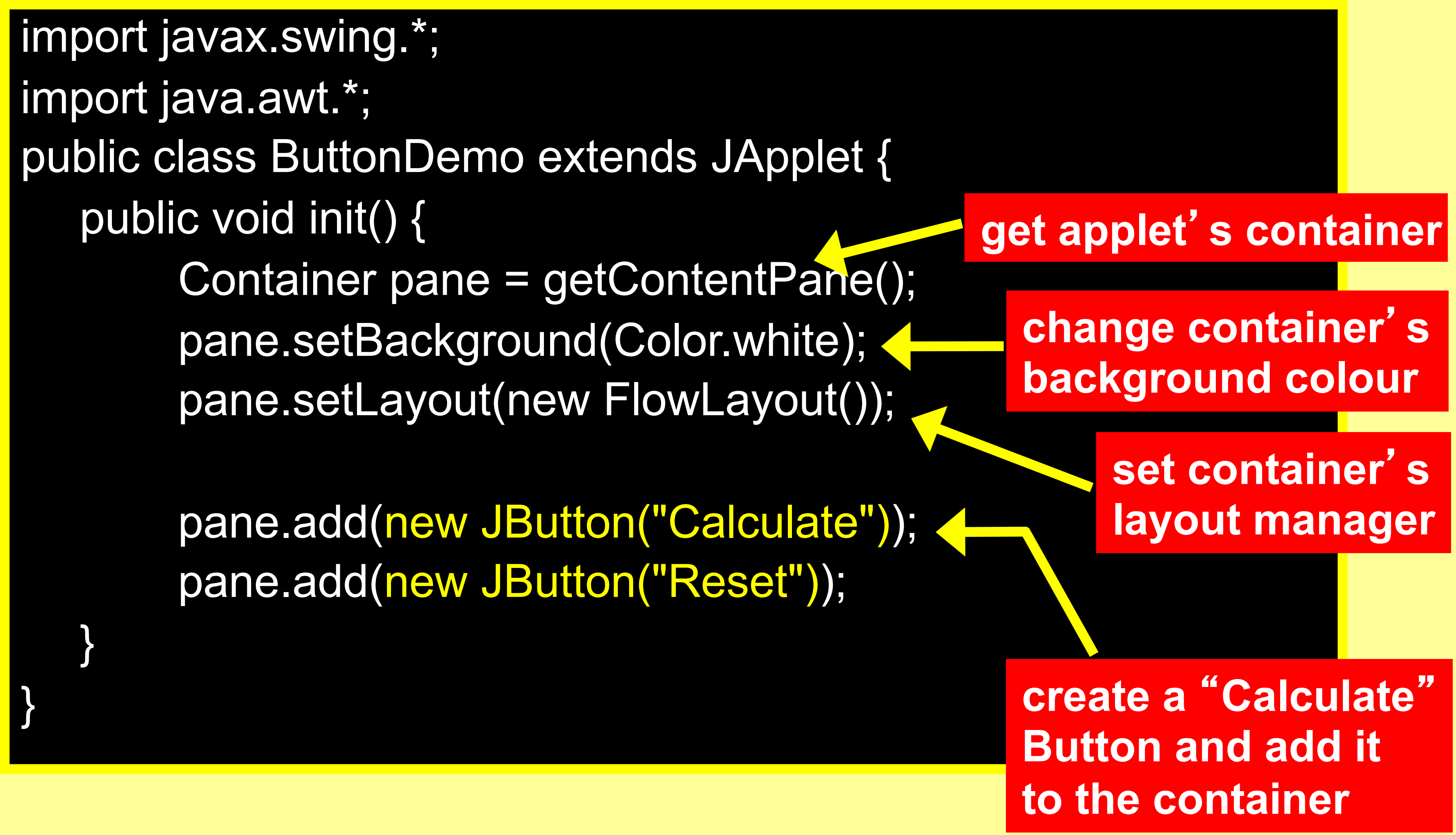
where

`s:` button label

- Example:

```
new JButton("OK")
```

ButtonDemo CODE



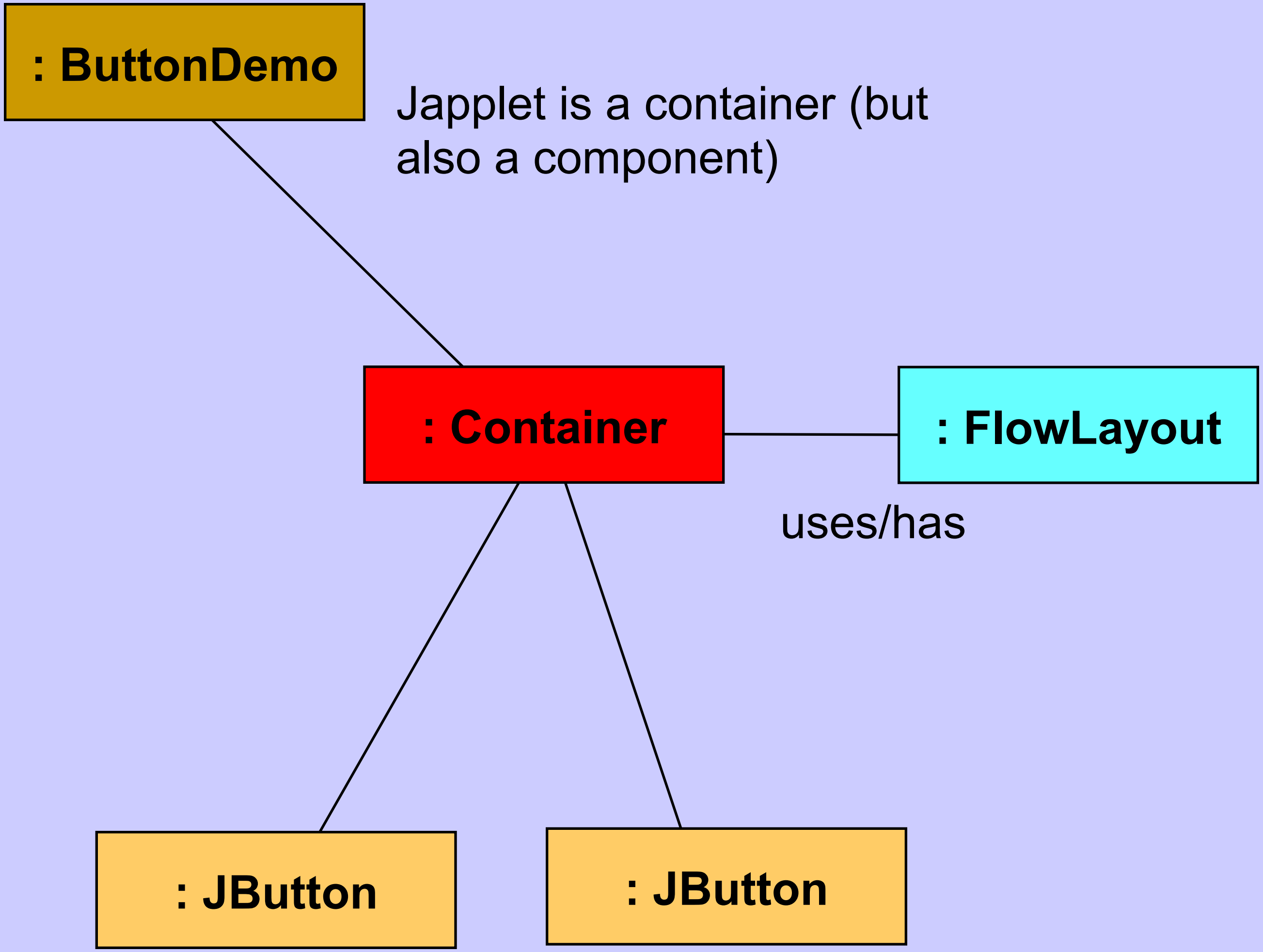
```
import javax.swing.*;
import java.awt.*;
public class ButtonDemo extends JApplet {
    public void init() {
        Container pane = getContentPane();
        pane.setBackground(Color.white);
        pane.setLayout(new FlowLayout());

        pane.add(new JButton("Calculate"));
        pane.add(new JButton("Reset"));
    }
}
```

get applet’ s container

change container’ s background colour

set container’ s layout manager

create a “Calculate” Button and add it to the container



: ButtonDemo

Japplet is a container (but also a component)

: Container

: FlowLayout

uses/has

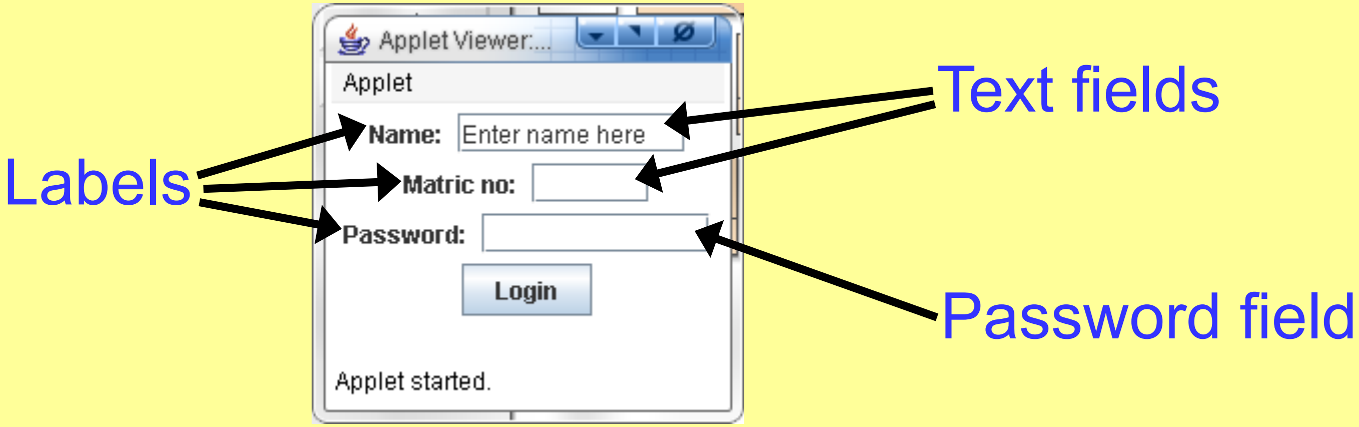
: JButton

: JButton

component

LoginUI DEMO APPLET

- *LoginUI*'s user interface:



Name: Enter name here

Matric no:

Password:

Applet started.

Labels

Text fields

Password field

LABELS

- Labels are represented as `JLabel` objects.
- Several ways to create a `JLabel` object:

```
    public JLabel()
    public JLabel(String s)
    public JLabel(String s, int j)
```

where

`s`: label text

`j`: alignment

Possible values :

```
        SwingConstants.LEFT
        SwingConstants.RIGHT
        SwingConstants.CENTER
```

- Examples:

```
new JLabel("Name")
new JLabel("RM5.00",SwingConstants.RIGHT)
```

JLabel.setText() METHOD

- The text of a label can be changed by sending a setText() message to the JLabel object.
- Example:

```
JLabel label = new JLabel("Country");
…
label.setText("Negara");
```

TEXT FIELDS

- Text fields are represented as `JTextField` objects.
- To create a `JTextField` object:

```
public JTextField(int length)
public JTextField(String s, int length)
```

where

`length`:	length of field (in columns)
`s`:	initial value of field

- Examples:

```
new JTextField(10)
new JTextField("Enter name here",10)
```

JTextField.setText() METHOD

- The value of a textfield can be set by sending a setText() message to the JTextField object.
- Example:

```
JTextField status = new JTextField(10);
…
status.setText("Login failed!");
```

JTextField.getText() METHOD

- The value of a textfield can be retrieved by sending a getText() message to the JTextField object.
- Example:

```
JTextField tfName = new JTextField(10);
…
String name = tfName.getText();
```

JTextField.setColumns() METHOD

- The length of a textfield can be changed by sending a `setColumns`() message to the `JTextField` object.
- Example:

```
JTextField tf = new JTextField(10);
…
tf.setColumns(20);
```

JTextField.setEditable() METHOD

- Sometimes we would like to make a text field non-editable; in other words, the user is not allowed to modify its value.
- This is achieved by sending a `setEditable()` message to the `JTextField` object.
- Example:

```
JTextField status = new JTextField(10);
status.setEditable(false);
```

PASSWORD FIELDS

- A password field is a special kind of text field. Password fields are represented as `JPasswordField` objects.
- To create a `JPasswordField` object:

```
public JPasswordField(int length)
```

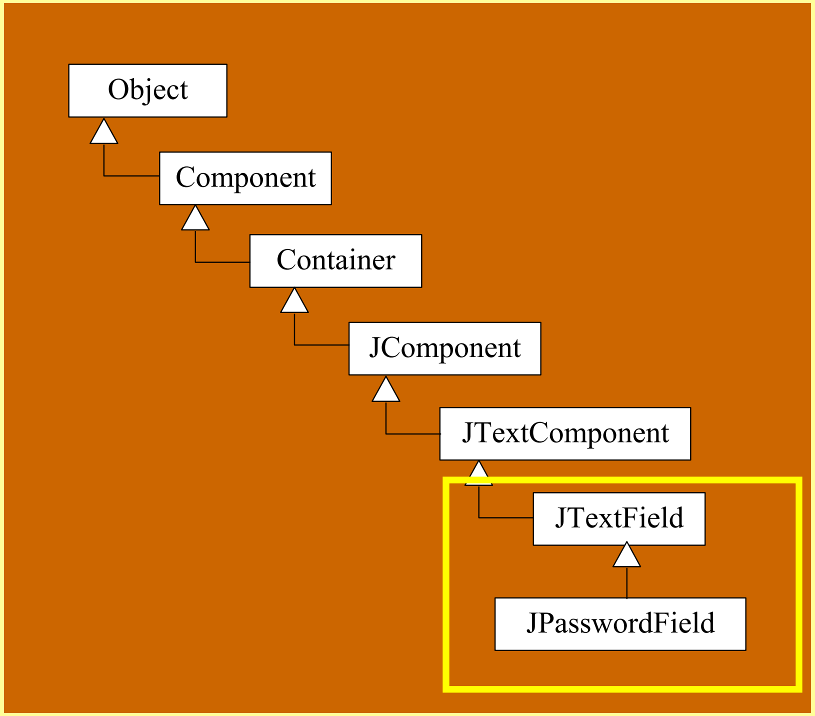
where

`length`: length of field (in columns)

Example:

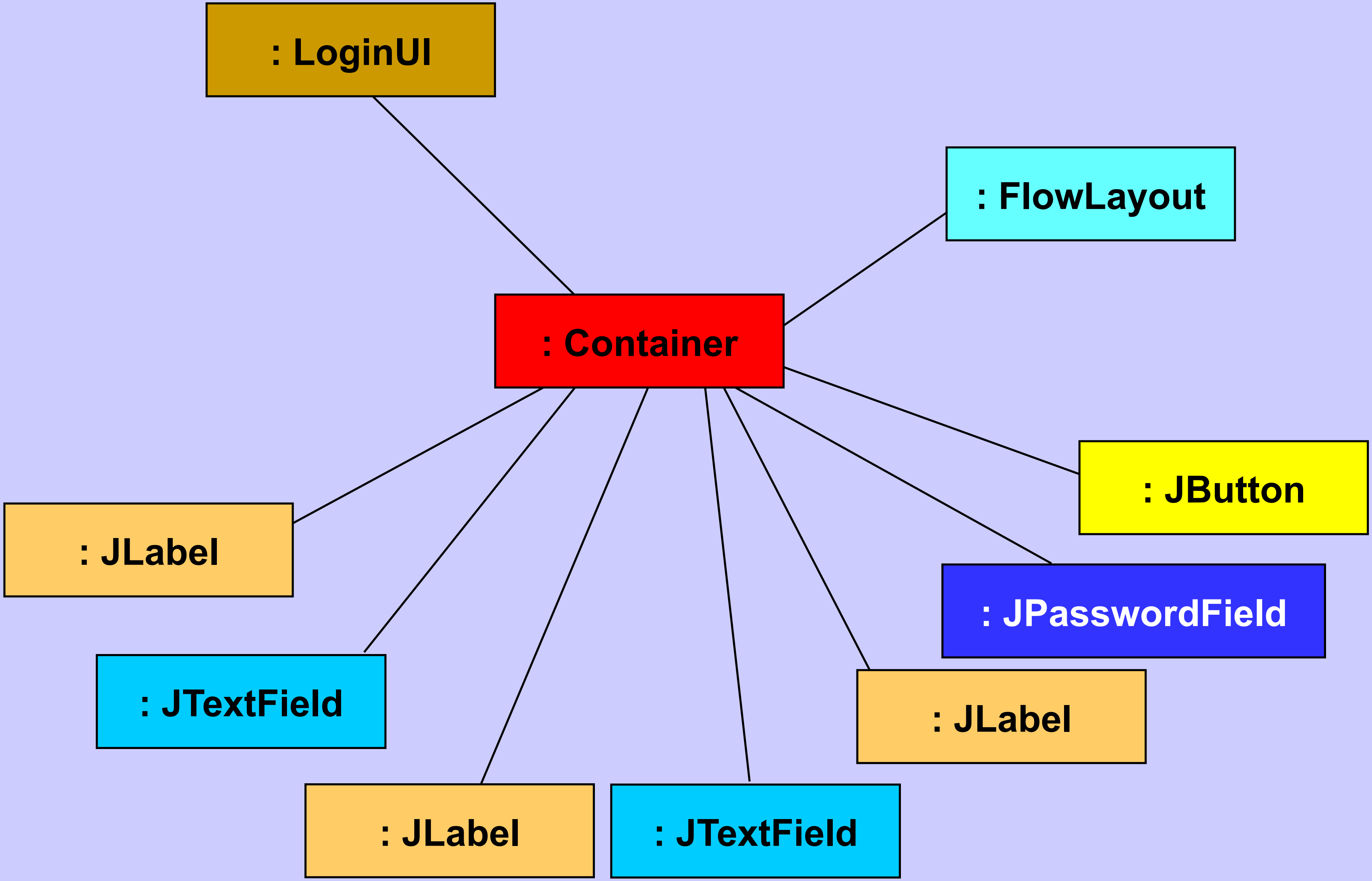
```
new JPasswordField(10)
```

- It is not surprising that `JPasswordField` is, in fact, a subclass of `JTextField`.



LoginUI CODE

```
public class LoginUI extends JApplet {
    public void init() {
        Container pane = getContentPane();
        pane.setBackground(Color.white);
        pane.setLayout(new FlowLayout());
        pane.add(new JLabel("Name: "));
        pane.add(new JTextField("Enter name here", 10));
        pane.add(new JLabel("Matric no: "));
        pane.add(new JTextField(5));
        pane.add(new JLabel("Password: "));
        pane.add(new JPasswordField(10));
        pane.add(new JButton("Login"));
    }
}
```



: LoginUI

: FlowLayout

: Container

: JButton

: JLabel

: JPasswordField

: JTextField

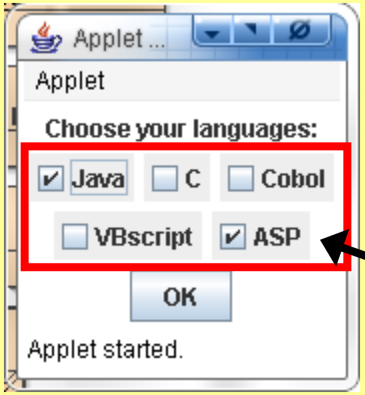
: JLabel

: JLabel

: JTextField

CheckBoxDemo DEMO APPLET

- *CheckBoxDemo*'s user interface:



Choose your languages:

☑ Java ☐ C ☐ Cobol

☐ VBscript ☑ ASP

Applet started.

Checkboxes

CHECK BOXES

- Check boxes are represented as `JCheckBox` objects.
- To create a `JCheckBox` object:

```
public JCheckBox(String label)
public JCheckBox(String label, boolean status)
```

where

`label`:	check box label
`status`:	initial status of check box (default is false)

- Examples:

```
new JCheckBox("C")
new JCheckBox("Java",true)
```

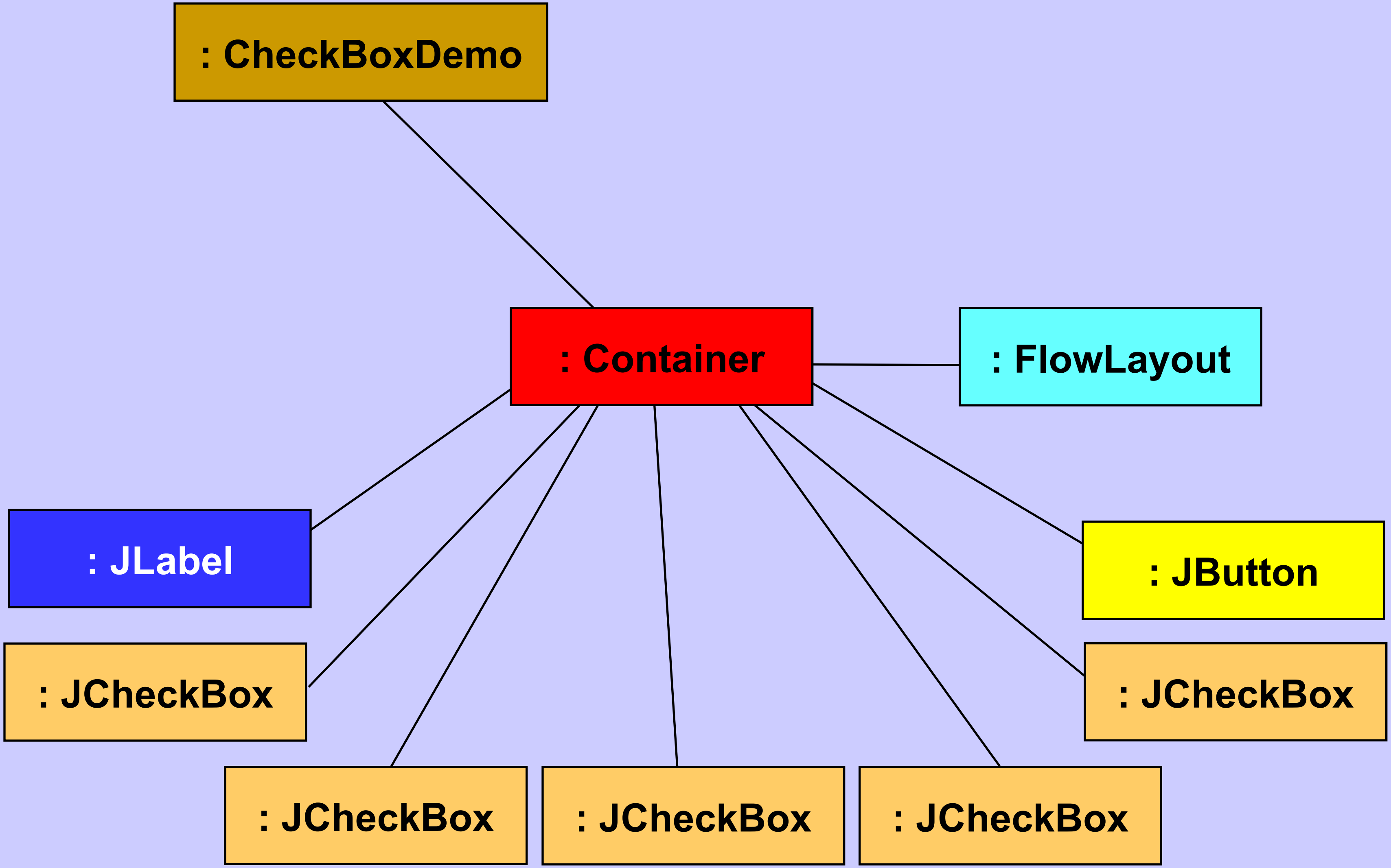
JCheckBox.isSelected() METHOD

- The current status of a check box can be acquired by sending an isSelected() message to the JCheckBox object.
- Example:

```
JCheckBox cbox = new JCheckBox("C");
…
if (cbox.isSelected()) {
      /* The check box is selected */
      …
}
else {
      /* The check box is not selected*/
      …
}
```

CheckBoxDemo CODE

```
public class CheckBoxDemo extends JApplet {
    public void init() {
        Container pane = getContentPane();
        pane.setBackground(Color.white);
        pane.setLayout(new FlowLayout());
        pane.add(new JLabel("Choose your languages:"));
        pane.add(new JCheckBox("Java", true));
        pane.add(new JCheckBox("C"));
        pane.add(new JCheckBox("Cobol"));
        pane.add(new JCheckBox("VBscript"));
        pane.add(new JCheckBox("ASP", true));
        pane.add(new JButton("OK"));
    }
}
```



: CheckBoxDemo

: Container

: FlowLayout

: JLabel

: JButton

: JCheckBox

: JCheckBox

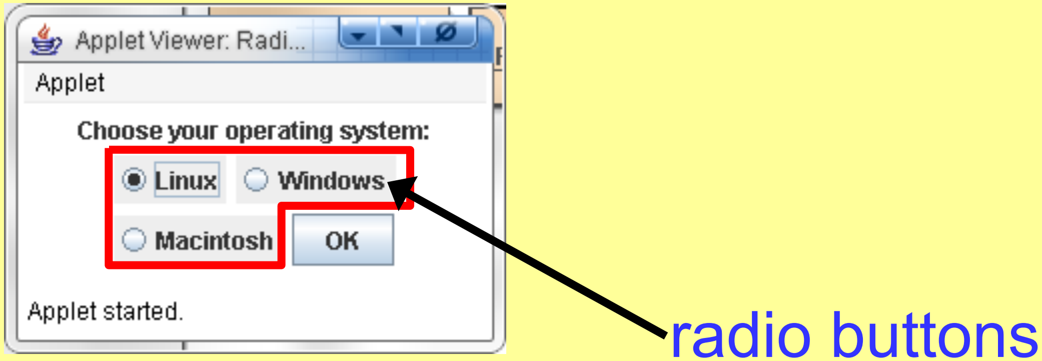
: JCheckBox

: JCheckBox

: JCheckBox

RadioButtonDemo1 DEMO APPLET

- *RadioButtonDemo1* 's user interface:



RADIO BUTTONS

- Radio buttons are represented as `JRadioButton` objects.
- To create a `JRadioButton` object:

```
public JRadioButton(String label)
public JRadioButton(String label, boolean status)
```

where

`label`:	radio button label
`status`:	initial status of radio button (default is false)

- Examples:

```
new JRadioButton("Windows")
new JRadioButton("Macintosh",true)
```

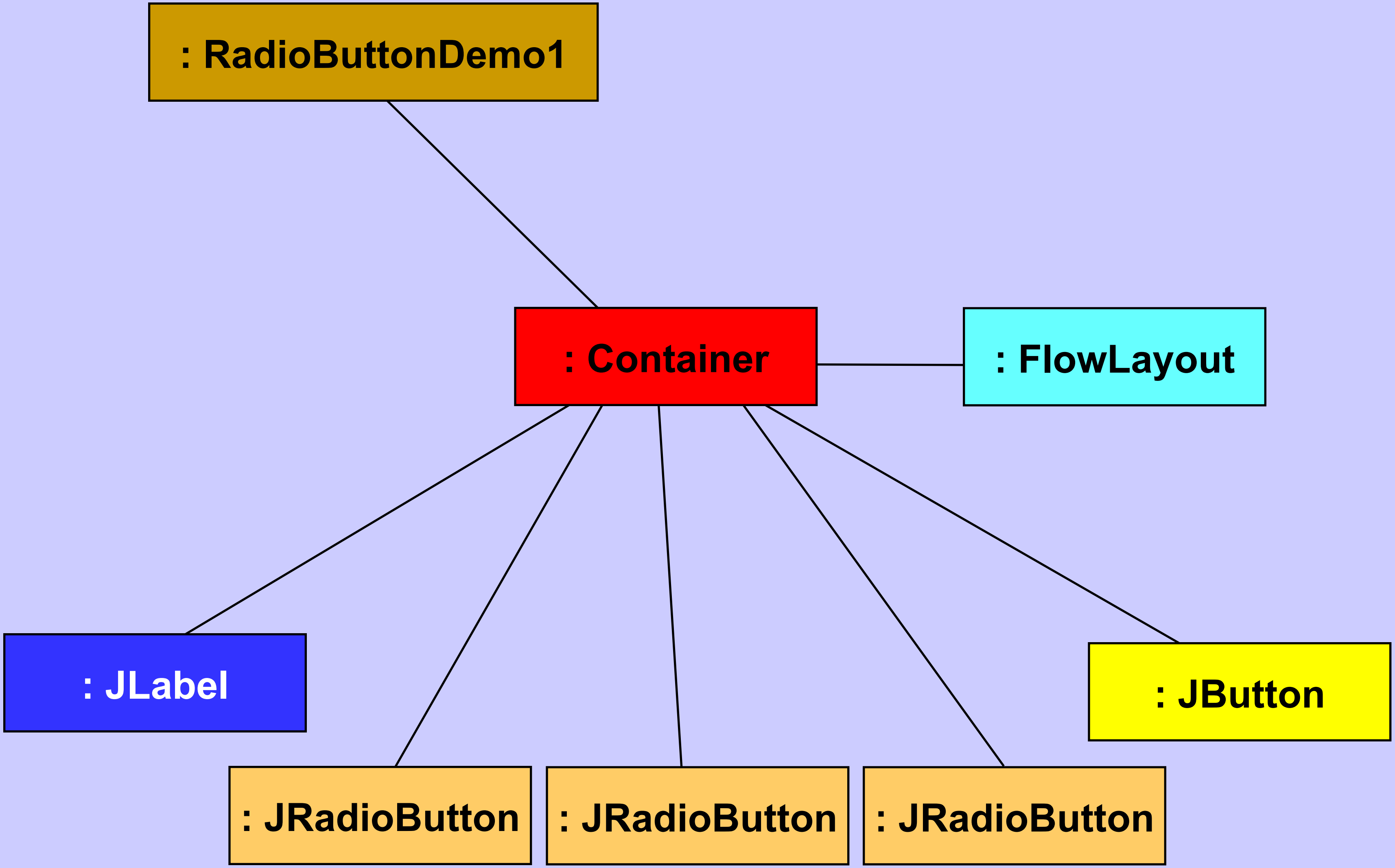
JRadioButton.isSelected() METHOD

- The current status of a radio button can be acquired by sending an isSelected() message to the JRadioButton object.
- Example:

```
JRadioButton rbtn = new JRadioButton("DOS");
…
if (rbtn.isSelected()) {
      /* The radio button is selected */
      …
}
else {
      /* The radio button is not selected*/
      …
}
```

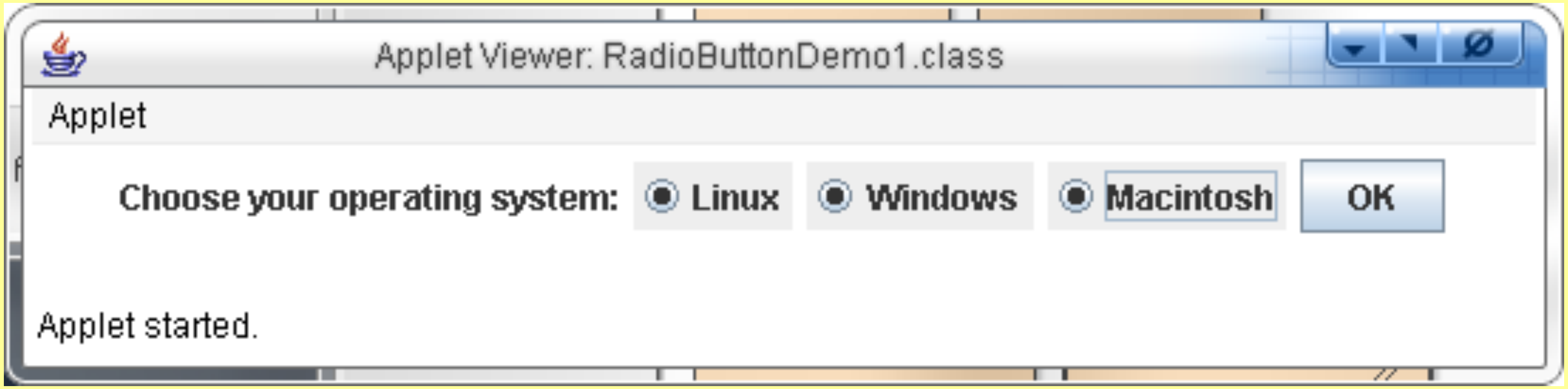
RadioButtonDemo1 CODE

```
public class RadioButtonDemo1 extends JApplet {
    public void init() {
        Container pane = getContentPane();
        pane.setBackground(Color.white);
        pane.setLayout(new FlowLayout());
        pane.add(new JLabel("Choose your operating system:"));
        pane.add(new JRadioButton("Linux", true));
        pane.add(new JRadioButton("Windows"));
        pane.add(new JRadioButton("Macintosh"));
        pane.add(new JButton("OK"));
    }
}
```


: RadioButtonDemo1
: Container
: FlowLayout
: JLabel
: JButton
: JRadioButton
: JRadioButton
: JRadioButton

ButtonGroup OBJECTS

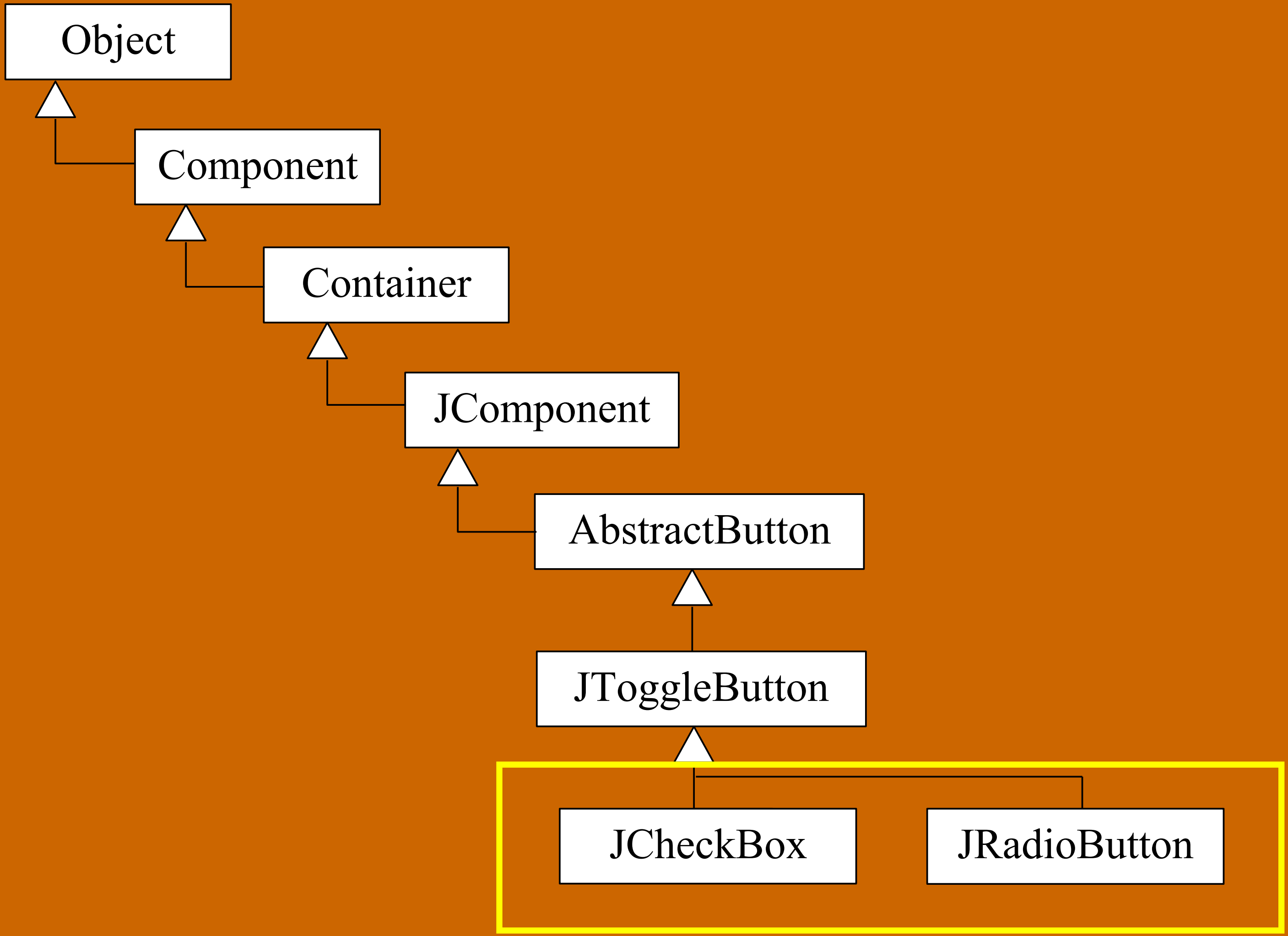
- Observe what happens when another radio button is selected in the *RadioButtonDemo1* applet.



Applet Viewer: RadioButtonDemo1.class

Choose your operating system: ● Linux ● Windows ● Macintosh

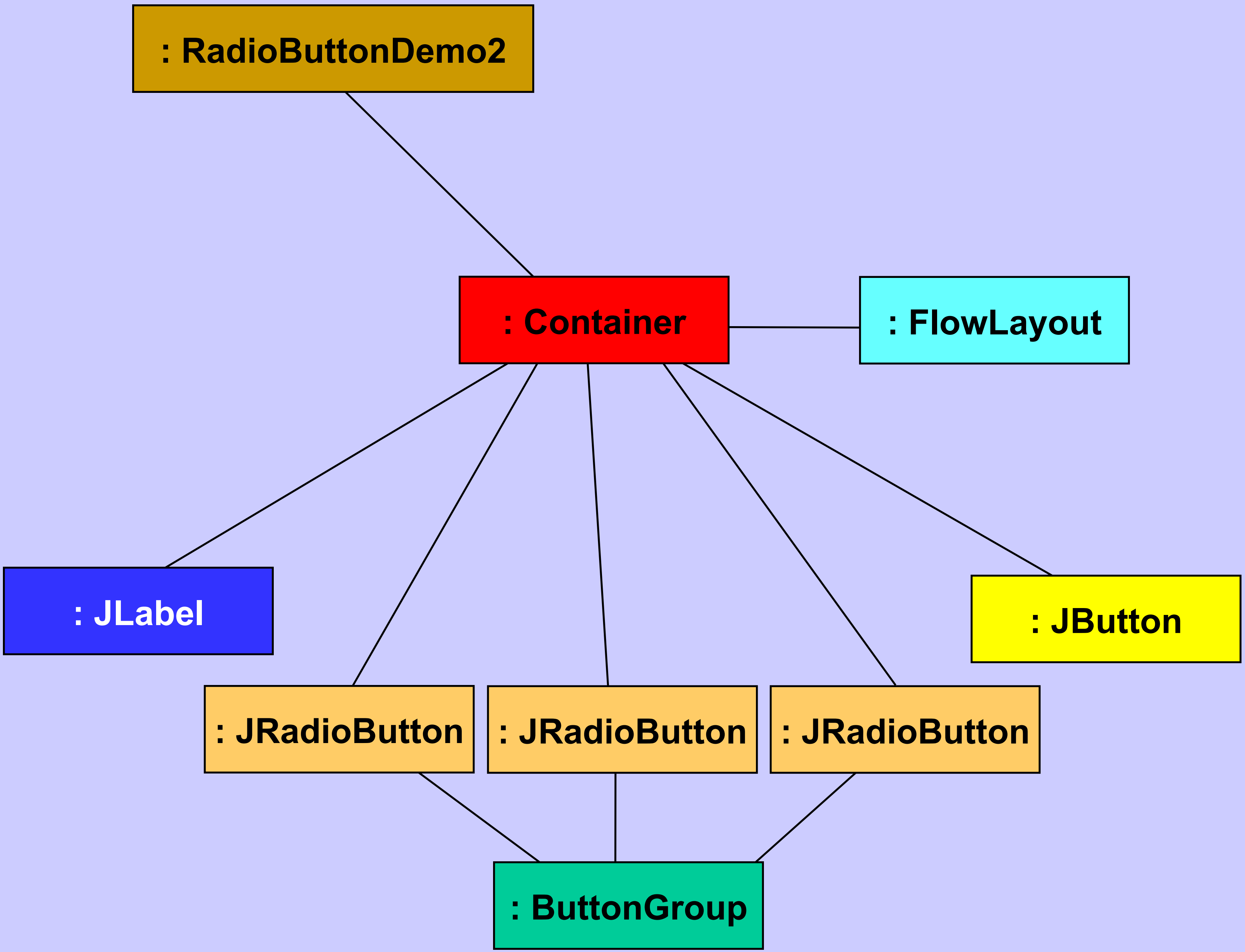
Applet started.


Object
Component
Container
JComponent
AbstractButton
JToggleButton
JCheckBox
JRadioButton

- We need to use a `ButtonGroup` object to group radio buttons so that only one of them can be selected at a time.

RadioButtonDemo2 CODE

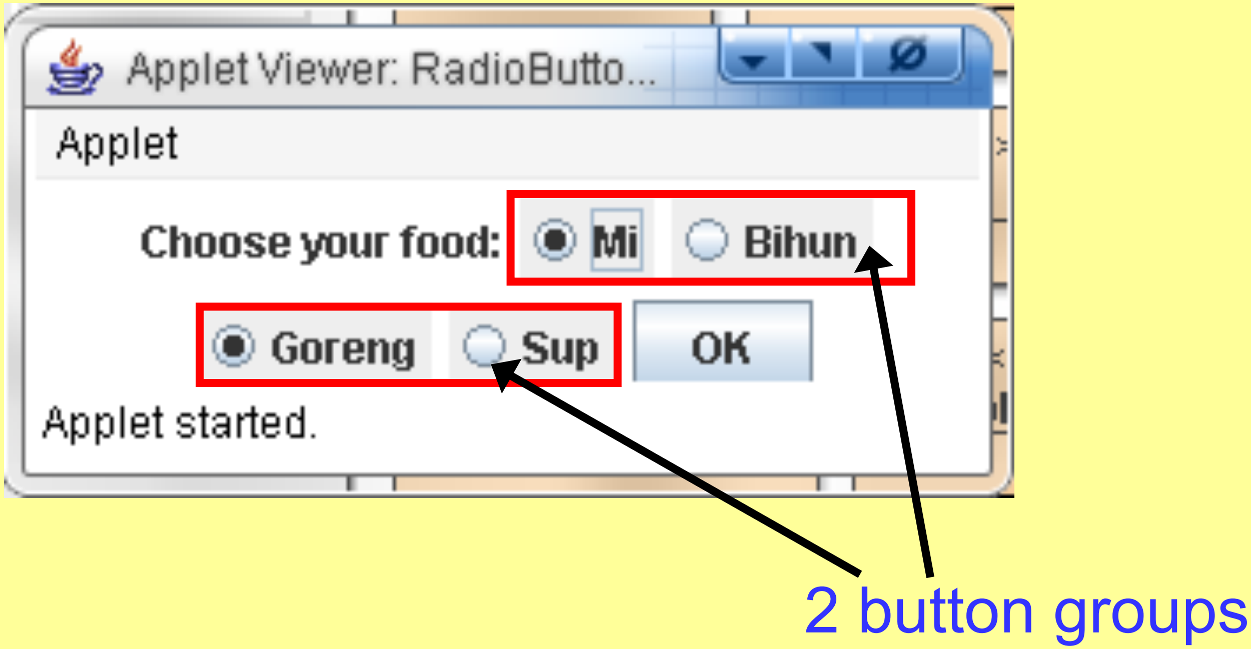
```
public class RadioButtonDemo2 extends JApplet {
    public void init() {
            Container pane = getContentPane();
            pane.setBackground(Color.white);
            pane.setLayout(new FlowLayout());
            pane.add(new JLabel("Choose your operating system:"));
            ButtonGroup grp = new ButtonGroup();
            JRadioButton rbtn = new JRadioButton("Linux", true);
            pane.add(rbtn);
            grp.add(rbtn);
            rbtn = new JRadioButton("Windows");
            pane.add(rbtn);
            grp.add(rbtn);
            rbtn = new JRadioButton("Macintosh");
            pane.add(rbtn);
            grp.add(rbtn);
            pane.add(new JButton("OK"));
      }
}
```



: RadioButtonDemo2
: Container
: FlowLayout
: JLabel
: JButton
: JRadioButton
: JRadioButton
: JRadioButton
: ButtonGroup

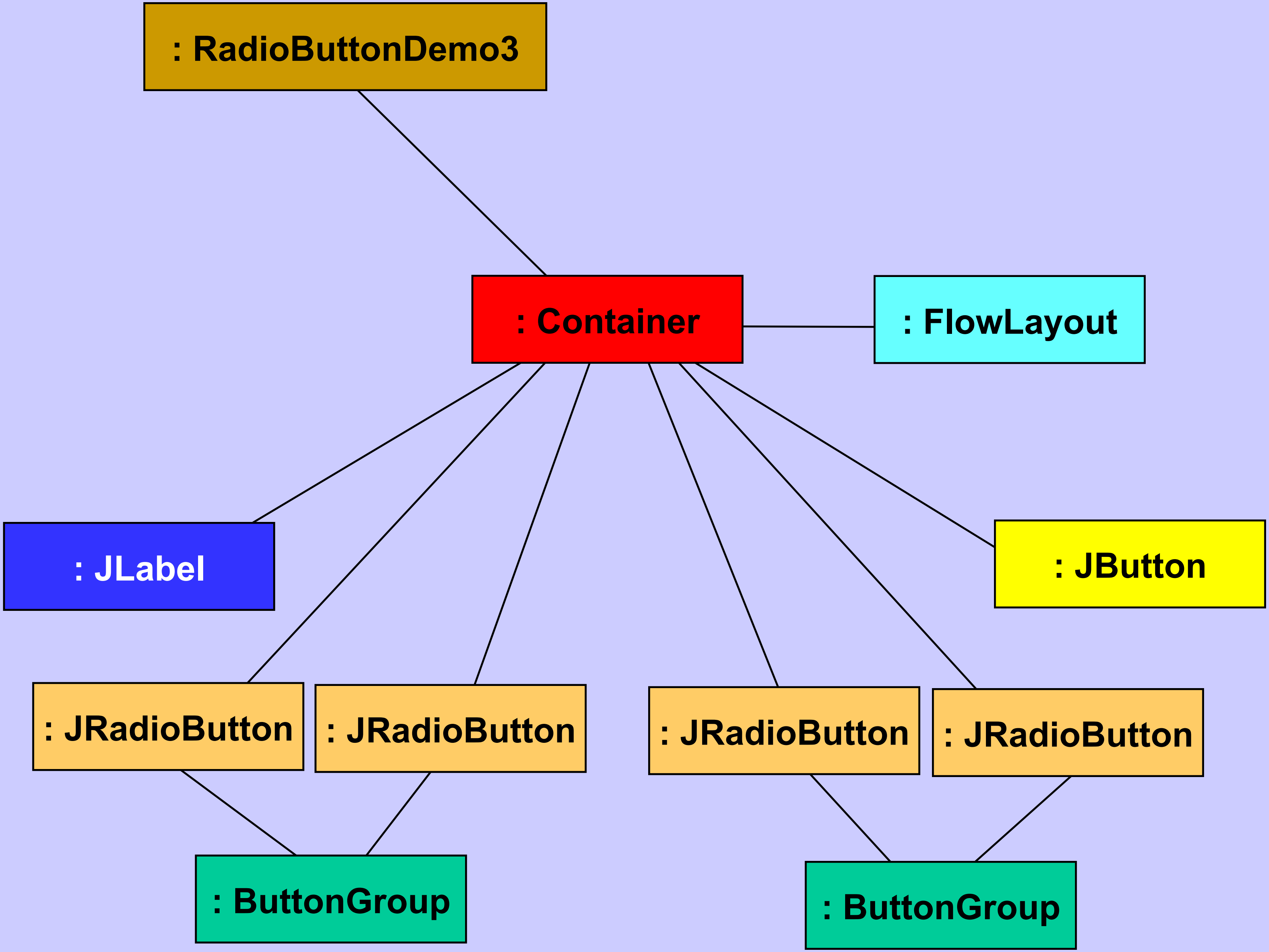
RadioButtonDemo3 DEMO APPLET

- *RadioButtonDemo3*'s user interface:



RadioButtonDemo3 CODE

```
public class RadioButtonDemo3 extends JApplet {
    public void init() {
        Container pane = getContentPane();
        pane.setBackground(Color.white);
        pane.setLayout(new FlowLayout());
        pane.add(new JLabel("Choose your food:"));
        ButtonGroup grp = new ButtonGroup();
        JRadioButton rbtn = new JRadioButton("Mi", true);
        pane.add(rbtn);
        grp.add(rbtn);
        rbtn = new JRadioButton("Bihun");
        pane.add(rbtn);
        grp.add(rbtn);
        grp = new ButtonGroup();
        rbtn = new JRadioButton("Goreng", true);
        pane.add(rbtn);
        grp.add(rbtn);
        rbtn = new JRadioButton("Sup");
        pane.add(rbtn);
        grp.add(rbtn);
        pane.add(new JButton("OK"));
    }
}
```



: RadioButtonDemo3
: Container
: FlowLayout
: JLabel
: JButton
: JRadioButton
: JRadioButton
: JRadioButton
: JRadioButton
: ButtonGroup
: ButtonGroup