Swing Components and Containers

A component is an independent visual control. Swing Framework contains a large set of components which provide rich functionalities and allow high level of customization. They all are derived from JComponent class. All these components are lightweight components. This class provides some common functionality like pluggable look and feel, support for accessibility, drag and drop, layout, etc.

A container holds a group of components. It provides a space where a component can be managed and displayed. Containers are of two types:

1. **Top level Containers**
    - It inherits Component and Container of AWT.
    - It cannot be contained within other containers.
    - Heavyweight.
    - Example: JFrame, JDialog, JApplet
2. **Lightweight Containers**
    - It inherits JComponent class.
    - It is a general purpose container.
    - It can be used to organize related components together.
    - Example: JPanel

JButton

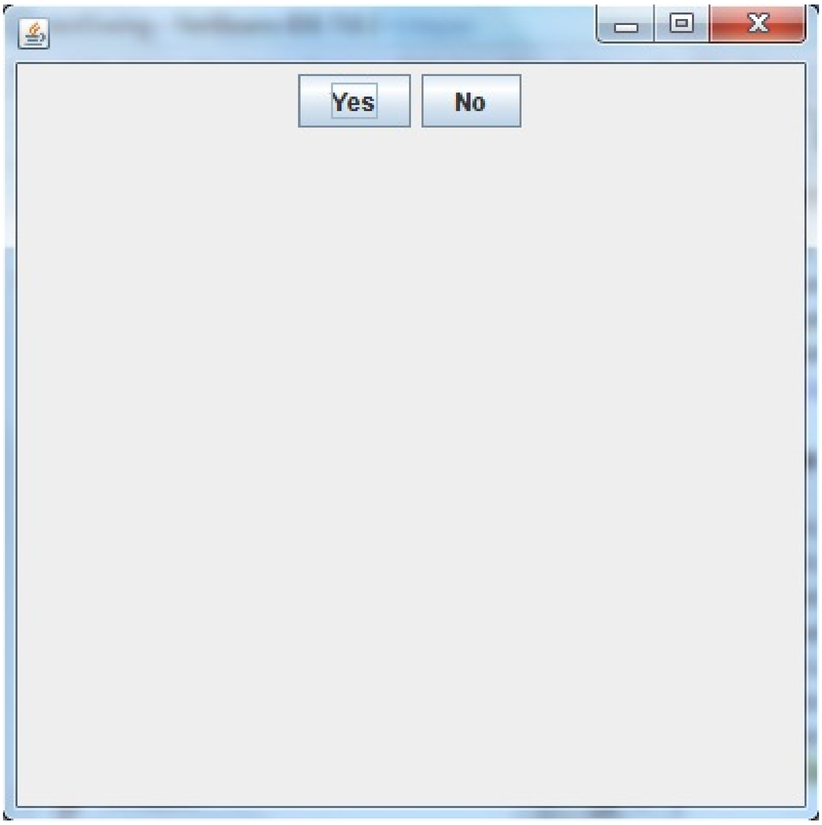
JButton class provides functionality of a button. JButton class has three constuctors,

```
JButton(Icon ic)

JButton(String str)

JButton(String str, Icon ic)
```

It allows a button to be created using icon, a string or both. JButton supports **ActionEvent**. When a button is pressed an **ActionEvent** is generated.

Example using JButton

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class testswing extends JFrame
{

 testswing()
 {
  JButton bt1 = new JButton("Yes");
  JButton bt2 = new JButton("No");
  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)
  setLayout(new FlowLayout());
  setSize(400, 400);
  add(bt1);
  add(bt2);

  setVisible(true);
 }
 public static void main(String[] args)
 {
  new testswing();
 }
}
```



JTextField

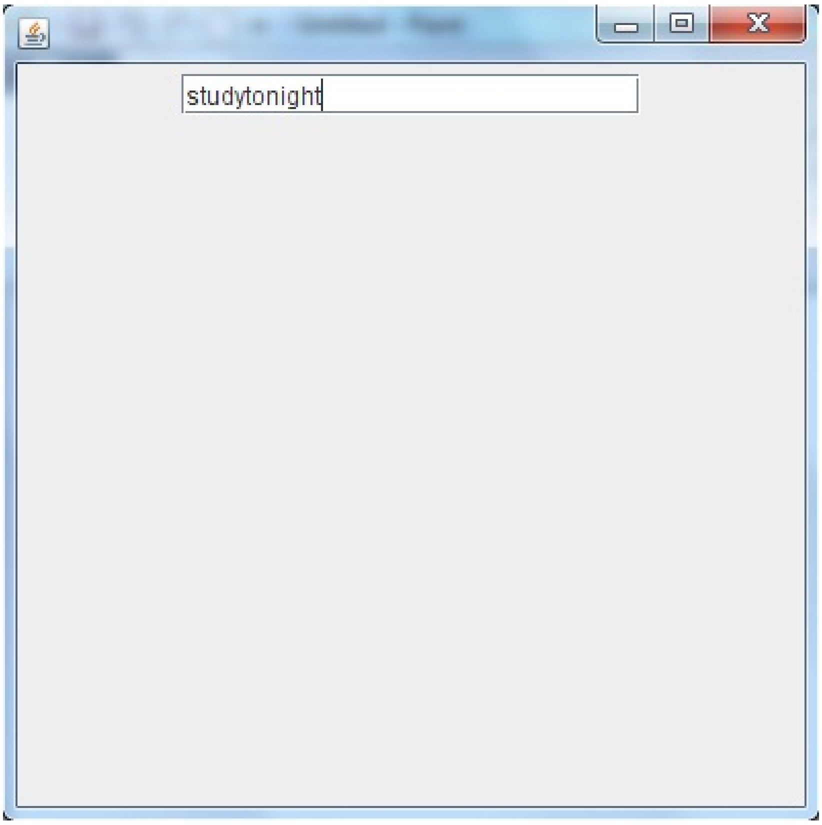
JTextField is used for taking input of single line of text. It is most widely used text component. It has three constructors,

```
JTextField(int cols)
JTextField(String str, int cols)
JTextField(String str)
```

cols represent the number of columns in text field.

Example using JTextField

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class MyTextField extends JFrame
{
 public MyTextField()
 {
  JTextField jtf = new JTextField(20);
  add(jtf);
  setLayout(new FlowLayout());
  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  setSize(400, 400);
  setVisible(true);
 }
 public static void main(String[] args)
 {
  new MyTextField();
 }
}
```



studytonight

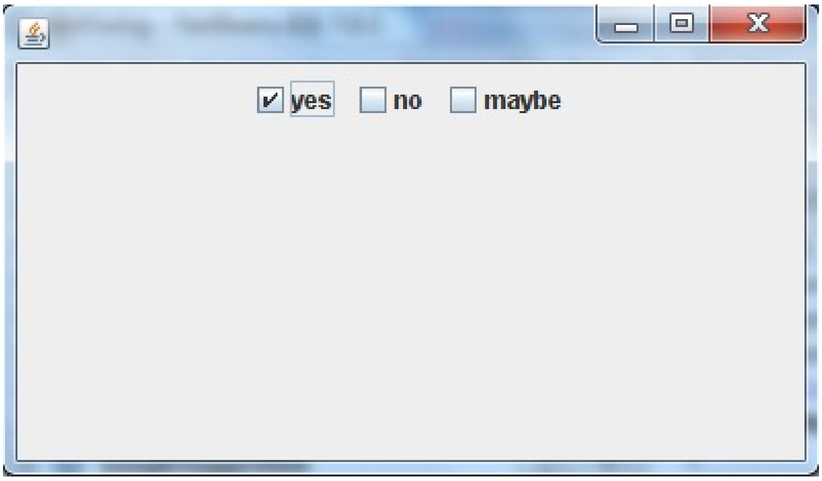
JCheckBox

JCheckBox class is used to create checkboxes in frame. Following is constructor for JCheckBox,

```
JCheckBox(String str)
```

Example using JCheckBox

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class Test extends JFrame
{
 public Test()
 {
  JCheckBox jcb = new JCheckBox("yes");
  add(jcb);
  jcb = new JCheckBox("no");
  add(jcb);
  jcb = new JCheckBox("maybe");
  add(jcb);
  setLayout(new FlowLayout());
  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  setSize(400, 400);
  setVisible(true);
 }
 public static void main(String[] args)
 {
  new Test();
 }
}
```



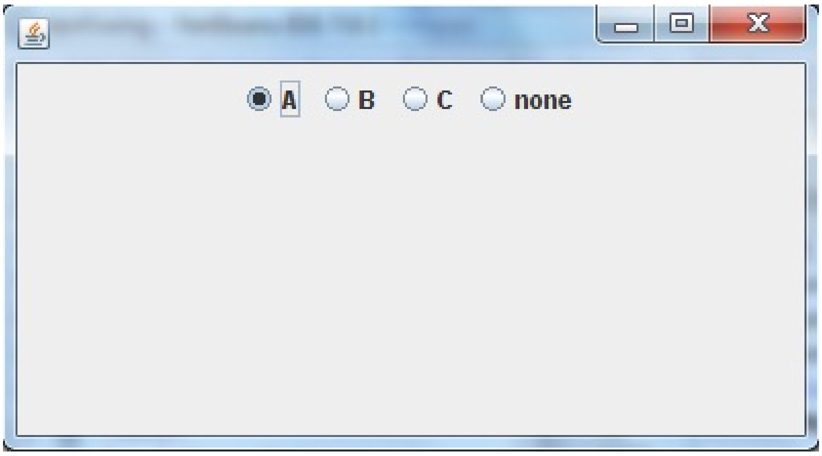
JRadioButton

Radio button is a group of related button in which only one can be selected. JRadioButton class is used to create a radio button in Frames. Following is the constructor for JRadioButton,

```
JRadioButton(String str)
```

Example using JRadioButton

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class Test extends JFrame
{
 public Test()
 {
  JRadioButton jcb = new JRadioButton("A");
  add(jcb);
  jcb = new JRadioButton("B");
  add(jcb);
  jcb = new JRadioButton("C");
  add(jcb);
  jcb = new JRadioButton("none");
  add(jcb);
  setLayout(new FlowLayout());
  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  setSize(400, 400);
  setVisible(true);
 }
 public static void main(String[] args)
 {
  new Test();
 }
}
```



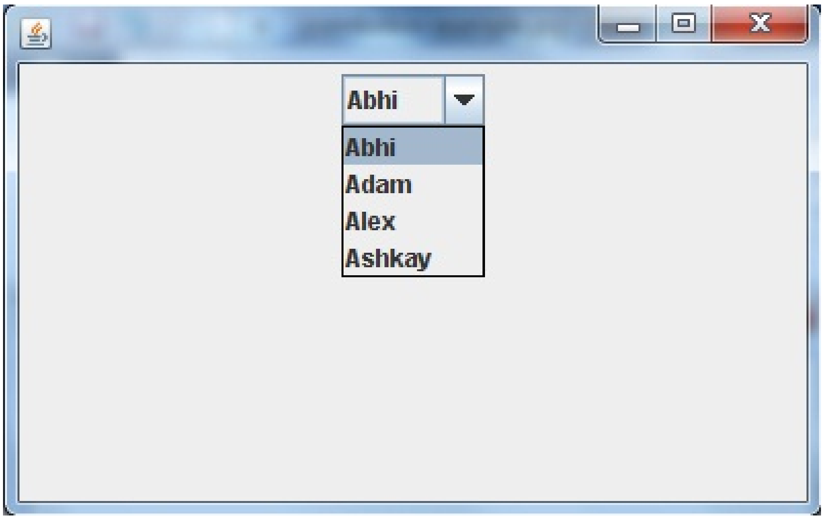
JComboBox

Combo box is a combination of text fields and drop-down list**JComboBox** component is used to create a combo box in Swing. Following is the constructor for JComboBox,

```
JComboBox(String arr[])
```

Example using JComboBox

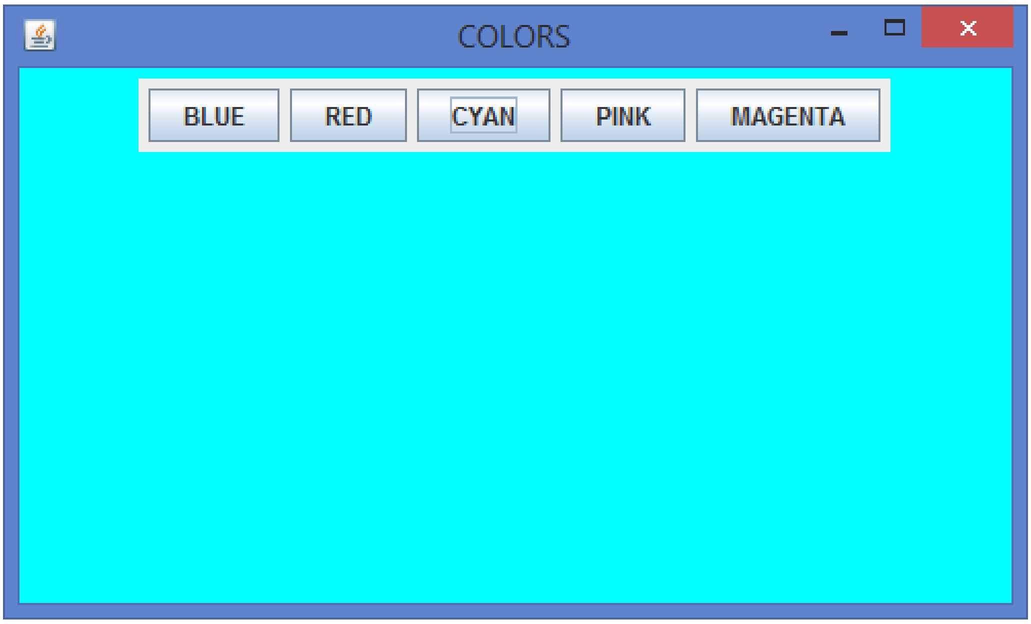
```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class Test extends JFrame
{
 String name[] = {"Abhi","Adam","Alex","Ashkay"};
 public Test()
 {
  JComboBox jc = new JComboBox(name);
  add(jc);
  setLayout(new FlowLayout());
  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  setSize(400, 400);
  setVisible(true);
 }
 public static void main(String[] args)
 {
  new Test();
 }
}
```



A program to change background color of a frame (Using Action Event)

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


class StColor extends JFrame implements ActionListener{

JFrame frame;
JPanel panel;
JButton b1,b2,b3,b4,b5;

StColor(){

   frame = new JFrame("COLORS");
   frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   panel = new JPanel();
   panel.setSize(100, 50);
   b1 = new JButton("BLUE");
   b1.addActionListener(this);

   b2 = new JButton("RED");
   b2.addActionListener(this);

   b3 = new JButton("CYAN");
   b3.addActionListener(this);

   b4 = new JButton("PINK");
   b4.addActionListener(this);

   b5 = new JButton("MAGENTA");
   b5.addActionListener(this);


   panel.add(b1);
   panel.add(b2);
   panel.add(b3);
   panel.add(b4);
   panel.add(b5);

   frame.getContentPane().add(panel);
   frame.setSize(500,300);
   frame.setVisible(true);
   frame.setLayout(new FlowLayout());

}

    @Override
    public void actionPerformed(ActionEvent e) {


        Pressing of button initially occurred
        Object see = e.getSource();

        if(see ==(b1)){
        frame.getContentPane().setBackground(java.awt.Color.blue);
        }
        if(see == b2){
            frame.getContentPane().setBackground(java.awt.Color.red);
        }
        if(see == b3){
        frame.getContentPane().setBackground(java.awt.Color.cyan);
        }
        if(see == b4){
            frame.getContentPane().setBackground(java.awt.Color.pink);
        }
        if(see == b5){
        frame.getContentPane().setBackground(java.awt.Color.magenta);
        }
    }
}

class Test {
    public static void main(String[] args) {
        StColor o = new StColor();
    }
}
```

Ouput:


COLORS
BLUE
RED
CYAN
PINK
MAGENTA