

Java gui tutorial

BASIC SWING

SWING contains a rich set of controls that enable Java developers to build GUI programs. In the following sections you will learn how to work with those controls. You also learn how to generate actions on them.

Working with J controls

JFrame: **JFrame** is a container that enables you to put controls like `JTextField`, `JButton`... in it.

Example:

```
import javax.swing.*; //don't forget to import this package

class MyJFrame extends JFrame{

    public MyJFrame(){

        super("TestFrame");//Display the word TestFrame on the title bar of the frame

        setSize(600,600); //set frame size—width=600, height=600

        setVisible(true);//make it appear on the screen

    }

}

public class TestFrame{

    public static void main(String[] args){

        new MyJFrame();

    }

}
```

JButton , JTextField, and JLabel

Example:

```
import javax.swing.*;

class ButtonTextFieldLabel extends JFrame{

    JButton bnt1,bnt2,bnt3;

    JTextField tf;

    JLabel lbl;

    public ButtonTextFieldLabel(){

        super("Hello");

        //create a text box, a label and 3 buttons objects

        tf=new JTextField("TextField");

        lbl=new JLabel("Label");

        bnt1=new JButton("Button1");

        bnt2=new JButton("Button2");

        bnt3=new JButton("Button3");

        setLayout(null);/*setting the layout of displaying controls to null
        will enable you to specify the locations of your controls to display
        */

        //set specific locations of the controls by using setBounds() method

        tf.setBounds(150,300,60,40);//x=90,y=100,width=60,height=40

        lbl.setBounds(150,150,80,40);

        bnt1.setBounds(50,200,80,40);

        bnt2.setBounds(150,200,80,40);

        bnt3.setBounds(250,200,80,40);
```

```

//add controls on the JFrame

add(tf);

add(lbl);

add(bnt1);

add(bnt2);

add(bnt3);

add(tf);

setSize(500,500);

setVisible(true);}

}

public class TestBTL{

    public static void main(String[] args){

        new ButtonTextFieldLabel();

    }

}

```

JCheckBox, JRadioButton, and JPanel

Example:

```

import javax.swing.*;

import java.awt.*;

class ChkRadio extends JFrame{

    JCheckBox chk1,chk2;

    JRadioButton rad1,rad2;

    public ChkRadio(){

```

```
super("CheckRadio");

//create two check boxes, two radio buttons objects

chk1=new JCheckBox("Female");

chk2=new JCheckBox("Male");

rad1=new JRadioButton("Single");

rad2=new JRadioButton("Married");

rad1.setSelected(true);//set default selection to true

chk1.setSelected(true);

//add radion buttons to group

ButtonGroup gr=new ButtonGroup();//create a group object

gr.add(rad1);

gr.add(rad2);

JPanel panel=new JPanel();//create a panel object to put radio buttons

panel.setLayout(new GridLayout(0,1));

panel.add(rad1);

panel.add(rad2);

setLayout(null);/*setting the layout of displaying controls to null

will enable you to specify the locations of your controls to display

*/

//set specific locations of the controls by using setBounds() method

chk1.setBounds(100,50,70,40);

chk2.setBounds(100,100,70,40);

//rad1.setBounds(100,175,70,40);

//rad2.setBounds(100,275,70,40);

panel.setBounds(100,175,70,40);
```

```
//add controls on the JFrame
```

```
add(chk1);
```

```
add(chk2);
```

```
add(panel);
```

```
setSize(500,500);
```

```
setVisible(true);}

}
```

```
public class TestChkRadio{
```

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

```
        new ChkRadio();
```

```
    }
```

```
}
```

JComboBox, JList, and JScrollPane

Example:

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
class ComList extends JFrame{
```

```
    public ComList(){
```

```
        super("Combox&List");
```

```
        JComboBox Ccountry;
```

```
        JList lanlist;
```

```
        Container contentpane=getContentPane();
```

```
contentpane.setLayout(new FlowLayout());

String[] counlist={"Cambodia","China","India","South Korea","UsA"};

String[] languages={"Khmer","Chinese","Indian","Korean"};

//Create Combobox,List, and ScrollPane objects

Ccountry=new JComboBox(counlist);

lanlist=new JList(languages);

JScrollPane scr=new JScrollPane(lanlist);

//set specific locations of the controls by using setBounds() method

Ccountry.setBounds(10,40,100,100);

lanlist.setBounds(100,40,100,40);

contentpane.add(Ccountry);

contentpane.add(scr);

setSize(500,500);

setVisible(true);}

}

public class TestComList{

    public static void main(String[] args){

        new ComList();

    }

}
```

JMenuBar, JMenu, and JMenuItem

Example:

```
import javax.swing.*;

import java.awt.*;

class MyMenu extends JFrame{

public MyMenu(){

    super("TestFrame");//Display the word TestFrame on the title of the frame

    JMenuBar mbar;

    JMenu mfile;

    JMenuItem itemnew,itemopen,itemsave,itemclose;

    mbar=new JMenuBar();//create a menu bar object

    mfile=new JMenu("File");//create menu object to be added to the menu bar

    mbar.add(mfile);

    itemnew= new JMenuItem("New...");//create menu items to be added to the menu

    itemopen=new JMenuItem("Open...");

    mfile.add(itemnew);

    mfile.add(itemopen);

    mfile.addSeparator();//set separator

    itemsave= new JMenuItem("Save");

    itemclose=new JMenuItem("Close");

    mfile.add(itemsave);

    mfile.add(itemclose);

    setLayout(new GridLayout(0,1));

    setJMenuBar(mbar);//set the menu bar to the frame window
```

```
setSize(600,600); //set frame size—width=600, height=600
```

```
setVisible(true);//make it appear on the screen
```

```
}
```

```
}
```

```
public class TestMenu{
```

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

```
        new MyMenu();
```

```
    }
```

```
}
```

Event Handling

So far, you have learnt some basic concepts of Java Swing components. However, once you perform actions such as clicking, selecting...on those components, they will do nothing. Therefore, in this section, you will learn how to generate actions for those components when you interact with them. This is called event handling.

In the summary table below, you will see interfaces, adapter classes, methods that will be used to generate actions for components.

AWT Event Listeners with Swing components

Action Event

Interface	ActionListener
Adapter class	
Registration method	addActionListener
Target components	JButton, JList, ...
Method	actionPerformed(ActionEvent e)

Adjustment Event

	Interface	AdjustmentListener
	Adapter class	
	Registration method	addAdjustmentListener
	Target components	JScrollPane
	Method	adjustmentValueChanged(AdjustmentEvent e)
Container Event		
	Interface	ContainerListener
	Adapter class	ContainerAdapter
	Registration method	AddContainerListener
	Target components	JFrame, JApplet, ...
	Methods	componentAdded(ContainerEvent e) componentRemoved(ContainerEvent e)
Focus Event		
	Interface	FocusListener
	Adapter class	FocusAdapter
	Registration method	addFocusListener
	Target components	JTextField, JTextArea, JButton, ...
	Methods	focusGained(FocusEvent e) focusLost(FocusEvent e)
Item Event		
	Interface	ItemListener
	Adapter class	
	Registration method	addItemListener
	Target components	JComboBox, JList, ...
	Method	itemStateChanged(ItemEvent e)
Key Event		
	Interface	KeyListener
	Adapter class	KeyAdapter
	Registration method	addKeyListener
	Target components	JTextField, JTextArea, ...
	Methods	keyPressed(KeyEvent e) keyReleased(KeyEvent e) keyTyped(KeyEvent e)
Mouse Event		
	Interface	MouseListener
	Adapter class	MouseAdapter

	Registration method	addMouseListener
	Target components	JButton, JComboBox,...
	Methods	mouseClicked(MouseEvent e) mouseEntered(MouseEvent e) mousePressed(MouseEvent e) mouseReleased(MouseEvent e)
Mouse Motion Event		
	Interface	MouseMotionListener
	Adapter class	MouseMotionAdapter
	Registration method	addMouseMotionListener
	Target components	JFrame, JApplet, ...
	Methods	mouseMoved(MouseMotionEvent e) mouseDragged(MouseMotionEvent e)
Text Event		
	Interface	TextListener
	Adapter class	
	Registration method	addTextListener
	Target components	JTextField, JTextArea,...
	Methods	textValueChanged(TextEvent e)
Window Event		
	Interface	WindowListener
	Adapter class	WindowAdapter
	Registration method	addWindowListener
	Target components	Jframe,...
	Methods	windowClosed(WindowEvent e) windowClosing(WindowEvent e) windowOpened(WindowEvent e) windowActivated(WindowEvent e) windowDeactivated(WindowEvent e)

Java GUI Action Event example:

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

import javax.swing.*;
```

```
class ButtonEvent extends JFrame implements ActionListener{ //implementing interface
ActionListener

    JButton bnt1,bnt2,bnt3;

    public ButtonEvent(){

    super("Hello");

    bnt1=new JButton("Button1");

    bnt2=new JButton("Button2");

    bnt3=new JButton("Button3");

    setLayout(null);

    bnt1.setBounds(10,40,70,30);

    bnt2.setBounds(90,40,70,30);

    bnt3.setBounds(170,40,70,30);

    bnt1.addActionListener(this);//Registering event

    bnt2.addActionListener(this);

    bnt3.addActionListener(this);

    add(bnt1);

    add(bnt2);

    add(bnt3);

    addWindowListener(new Listener());

    setSize(600,600);

    setVisible(true);}

//Do something when an action is performed

    public void actionPerformed(ActionEvent event){

        if(event.getActionCommand()=="Button1") bnt1.setLabel("Click me");

        if(event.getActionCommand()=="Button2") bnt2.setLabel("Click me");
```

```
if(event.getActionCommand()=="Button3") bnt3.setLabel("Click me");

//use setLabel() method to display text on the button

//use getLabel() method if you want to get the text displayed on the button

}

//extends WindowAdapter class and use windowClosing() method to close window

private class Listener extends WindowAdapter{

    public void windowClosing(WindowEvent event){

        System.exit(0);

    }

}

}

public class ButtonTest{

    public static void main(String[] args){

        new ButtonEvent();

    }

}

}
```

By: <http://www.worldbestlearningcenter.com>