Java Swing Example

Let's see a simple swing example where we are creating one **button** and adding it on the JFrame object inside the main() method.

```
import javax.swing.*;

public class FirstSwingExample {

public static void main(String[] args) {

JFrame f=new JFrame();//creating instance of JFrame


JButton b=new JButton("click");//creating instance of JButton

b.setBounds(130,100,100, 40);//x axis, y axis, width, height


f.add(b);//adding button in JFrame


f.setSize(400,500);//400 width and 500 height

f.setLayout(null);//using no layout managers

f.setVisible(true);//making the frame visible

}

}
```

inside constructor

```
import javax.swing.*;

public class Simple {

JFrame f;

Simple(){

f=new JFrame();//creating instance of JFrame
JButton b=new JButton("click");//creating instance of JButton
```

```
b.setBounds(130,100,100, 40);


f.add(b);//adding button in JFrame


f.setSize(400,500);//400 width and 500 height
f.setLayout(null);//using no layout managers
f.setVisible(true);//making the frame visible
}


public static void main(String[] args) {
new Simple();
}
}
```

JButton

The JButton class is used to create a labeled button that has platform independent implementation. The application result in some action when the button is pushed. It inherits AbstractButton class.

JButton class declaration

Let's see the declaration for javax.swing.JButton class.

```
public class JButton extends AbstractButton implements Accessible
```

Commonly used Constructors:

Constructor	Description
JButton()	It creates a button with no text and icon.
JButton(String s)	It creates a button with the specified text.

JButton(Icon i) It creates a button with the specified icon object.

Commonly used Methods of AbstractButton class:

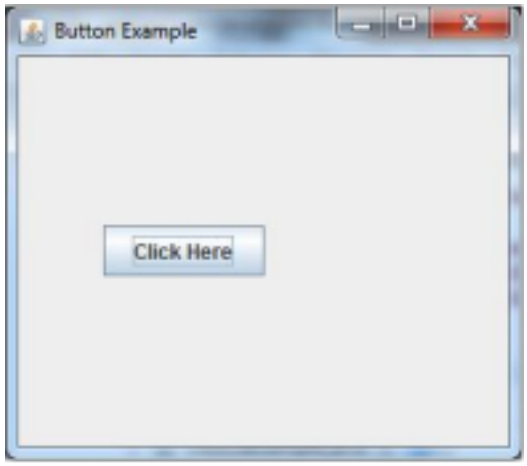
Methods	Description
void setText(String s)	It is used to set specified text on button
String getText()	It is used to return the text of the button.

void setEnabled(boolean b) It is used to enable or disable the button.

void setIcon(Icon b) It is used to set the specified Icon on the button.

Icon getIcon() It is used to get the Icon of the button.
void setMnemonic(int a) It is used to set the mnemonic on the button.
void addActionListener(ActionListener a) It is used to add the action listener to this object.

```
import javax.swing.*;
public class ButtonExample {
public static void main(String[] args) {
    JFrame f=new JFrame("Button Example");
    JButton b=new JButton("Click Here");
    b.setBounds(50,100,95,30);
    f.add(b);
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true); } }
```



ActionListener

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

import javax.swing.*;

public class ButtonExample {

public static void main(String[] args) {

    JFrame f=new JFrame("Button Example");

    final JTextField tf=new JTextField();

    tf.setBounds(50,50, 150,20);

    JButton b=new JButton("Click Here");

    b.setBounds(50,100,95,30);

    b.addActionListener(new ActionListener(){ public

void actionPerformed(ActionEvent e){
```

```
tf.setText("Welcome to Java Programming."); }

   });

   f.add(b);f.add(tf);

   f.setSize(400,400);

   f.setLayout(null);

   f.setVisible(true); } }
```

displaying image on the button:

```
import javax.swing.*;

public class ButtonExample{

ButtonExample(){

JFrame f=new JFrame("Button Example");

JButton b=new JButton(new ImageIcon("D:\\icon.png"));

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

f.add(b);

f.setSize(300,400);

f.setLayout(null);

f.setVisible(true);

f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   }

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

```
    new ButtonExample();
}
}
```

JTextField Example with ActionListener

```
import javax.swing.*;
import java.awt.event.*;
public class TextFieldExample implements ActionListener{
    JTextField tf1,tf2,tf3;
    JButton b1,b2;
    TextFieldExample(){
        JFrame f= new JFrame();
        tf1=new JTextField();
        tf1.setBounds(50,50,150,20);
        tf2=new JTextField();
        tf2.setBounds(50,100,150,20);
        tf3=new JTextField();
        tf3.setBounds(50,150,150,20);
        tf3.setEditable(false);
        b1=new JButton("+");
```

```
    b1.setBounds(50,200,50,50);

    b2=new JButton("-");

    b2.setBounds(120,200,50,50);
    b1.addActionListener(this);

    b2.addActionListener(this);

    f.add(tf1);f.add(tf2);f.add(tf3);f.add(b1);f.add(b2)

    ; f.setSize(300,300);

    f.setLayout(null);

    f.setVisible(true);

}

public void actionPerformed(ActionEvent e) {

    String s1=tf1.getText();

    String s2=tf2.getText();

    int a=Integer.parseInt(s1);

    int b=Integer.parseInt(s2);

    int c=0;

    if(e.getSource()==b1){

        c=a+b;

    }else if(e.getSource()==b2){
```

```
            c=a-b;

        }

        String result=String.valueOf(c);

        tf3.setText(result);
    }

public static void main(String[] args) {

    new TextFieldExample();

} }
```

JTextArea Example

```
import javax.swing.*;

public class TextAreaExample

{

    TextAreaExample(){

        JFrame f= new JFrame();

        JTextArea area=new JTextArea("Welcome to

        javatpoint"); area.setBounds(10,30, 200,200);

        f.add(area);

        f.setSize(300,300);
```

```
    f.setLayout(null);

    f.setVisible(true);

  }

public static void main(String args[])

  {
  new TextAreaExample();

  }}
```

JTextArea Example with ActionListener

```
import javax.swing.*;

import java.awt.event.*;

public class TextAreaExample implements ActionListener{

JLabel l1,l2;

JTextArea area;

JButton b;

TextAreaExample() {

  JFrame f= new JFrame();

  l1=new JLabel();

  l1.setBounds(50,25,100,30);
```

```
    l2=new JLabel();

    l2.setBounds(160,25,100,30);

    area=new JTextArea();

    area.setBounds(20,75,250,200);

    b=new JButton("Count Words");
    b.setBounds(100,300,120,30);

    b.addActionListener(this);

    f.add(l1);f.add(l2);f.add(area);f.add(b)

    ; f.setSize(450,450);

    f.setLayout(null);

    f.setVisible(true);

}

 public void actionPerformed(ActionEvent

    e){ String text=area.getText();

    String words[]=text.split("\\s");

    l1.setText("Words: "+words.length);

l2.setText("Characters: "+text.length()); }

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

```
    new TextAreaExample();

}

}
```

JPasswordField Example

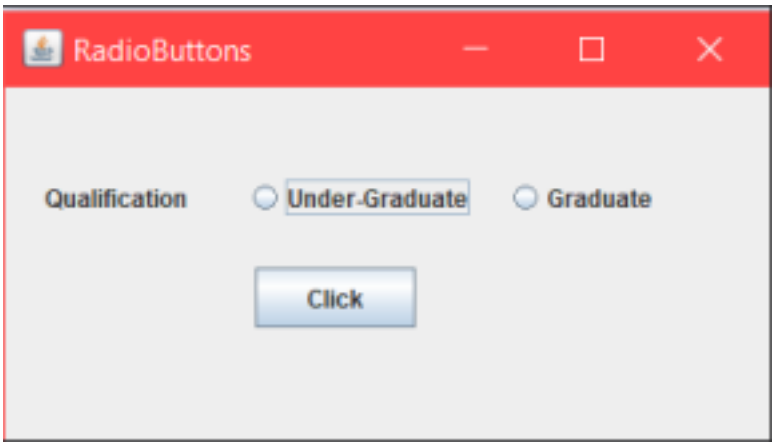
```
import javax.swing.*;

public class PasswordFieldExample {

    public static void main(String[] args) {

    JFrame f=new JFrame("Password Field Example");

     JPasswordField value = new JPasswordField();

     JLabel l1=new JLabel("Password:");

        l1.setBounds(20,100, 80,30);

         value.setBounds(100,100,100,30);

            f.add(value); f.add(l1);

            f.setSize(300,300);

            f.setLayout(null);

            f.setVisible(true);

}

}
```

JRadioButton

We use the JRadioButton class to create a radio button. Radio button is use to select one option from multiple options. It is used in filling forms, online objective papers and quiz.

We add radio buttons in a ButtonGroup so that we can select only one radio button at a time.
We use “ButtonGroup” class to create a ButtonGroup and add radio button in a group.
Methods Used :

JRadioButton() : Creates a unselected RadioButton with no text




```
// Java program to show JRadioButton Example.
// in java. Importing different Package.
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

class Demo extends JFrame {

        // Declaration of object of JRadioButton class.
        JRadioButton jRadioButton1;
```

```
// Declaration of object of JRadioButton class.
JRadioButton jRadioButton2;

// Declaration of object of JButton class.
JButton jButton;

// Declaration of object of ButtonGroup class.
ButtonGroup G1;

// Declaration of object of JLabel class.
JLabel L1;

// Constructor of Demo class.
public Demo()
{

        // Setting layout as null of JFrame.
        this.setLayout(null);

        // Initialization of object of "JRadioButton" class.
        jRadioButton1 = new JRadioButton();

        // Initialization of object of "JRadioButton" class.
        jRadioButton2 = new JRadioButton();

        // Initialization of object of "JButton" class.
        jButton = new JButton("Click");

        // Initialization of object of "ButtonGroup" class.
        G1 = new ButtonGroup();

        // Initialization of object of " JLabel" class.
        L1 = new JLabel("Qualification");

        // setText(...) function is used to set text of radio button.
        // Setting text of "jRadioButton2".
        jRadioButton1.setText("Under-Graduate");

        // Setting text of "jRadioButton4".
        jRadioButton2.setText("Graduate");

        // Setting Bounds of "jRadioButton2".
        jRadioButton1.setBounds(120, 30, 120, 50);
```

```
                // Setting Bounds of "jRadioButton4".
                jRadioButton2.setBounds(250, 30, 80, 50);

                // Setting Bounds of "jButton".
                jButton.setBounds(125, 90, 80, 30);

                // Setting Bounds of JLabel "L2".
                L1.setBounds(20, 30, 150, 50);

                // "this" keyword in java refers to current object.
                // Adding "jRadioButton2" on JFrame.
                this.add(jRadioButton1);

                // Adding "jRadioButton4" on JFrame.
                this.add(jRadioButton2);
                // Adding "jButton" on JFrame.
                this.add(jButton);

                // Adding JLabel "L2" on JFrame.
                this.add(L1);

                // Adding "jRadioButton1" and "jRadioButton3" in a Button Group "G2".
                G1.add(jRadioButton1);
                G1.add(jRadioButton2);
        }
}

class RadioButton {
        // Driver code.
        public static void main(String args[])
        { // Creating object of demo class.
                Demo f = new Demo();

                // Setting Bounds of JFrame.
                f.setBounds(100, 100, 400, 200);

                // Setting Title of frame.
                f.setTitle("RadioButtons");

                // Setting Visible status of frame as true.
                f.setVisible(true);
        }
}
```

Java JTabbedPane

The JTabbedPane class is used to switch between a group of components by clicking on a tab with a given title or icon. It inherits JComponent class.

JTabbedPane class declaration

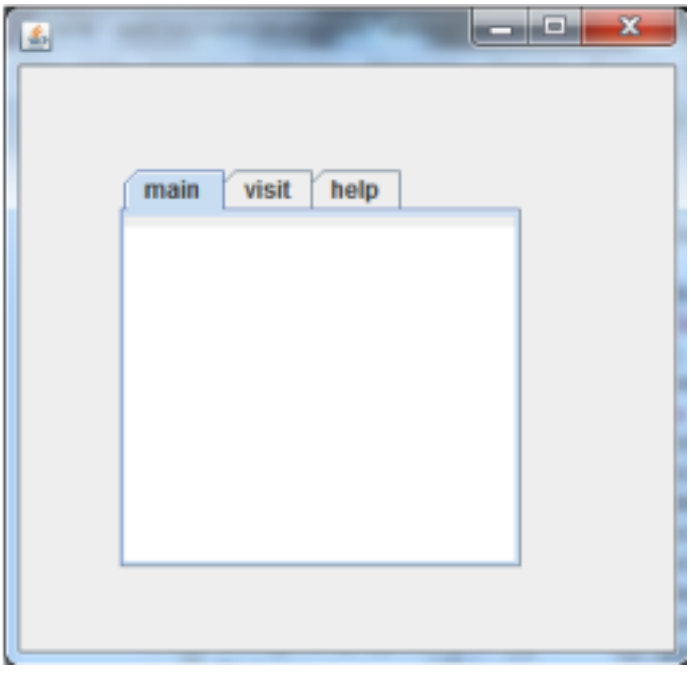
Let's see the declaration for javax.swing.JTabbedPane class.

public class JTabbedPane extends JComponent implements Serializable, Accessible, SwingConstants

Commonly used Constructors:

Constructor	Description
JTabbedPane()	Creates an empty TabbedPane with a default tab placement of JTabbedPane.Top.
JTabbedPane(int tabPlacement)	Creates an empty TabbedPane with a specified tab placement.
JTabbedPane(int tabPlacement, int tabLayoutPolicy)	Creates an empty TabbedPane with a specified tab placement and tab layout policy.

Java JTabbedPane Example



```
import javax.swing.*;
public class TabbedPaneExample {
JFrame f;
TabbedPaneExample(){
    f=new JFrame();
    JTextArea ta=new JTextArea(200,200);
    JPanel p1=new JPanel();
    p1.add(ta);
    JPanel p2=new JPanel();
    JPanel p3=new JPanel();
    JTabbedPane tp=new JTabbedPane();
    tp.setBounds(50,50,200,200);
    tp.add("main",p1);
    tp.add("visit",p2);
    tp.add("help",p3);
    f.add(tp);
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
}
public static void main(String[] args) {
    new TabbedPaneExample();
}}
```

JMenuBar, JMenu and JMenuItem

The JMenuBar class is used to display menubar on the window or frame. It may have several menus.

The object of JMenu class is a pull down menu component which is displayed from the menu bar. It inherits the JMenuItem class.

The object of JMenuItem class adds a simple labeled menu item. The items used in a menu must belong to the JMenuItem or any of its subclass.

JMenuBar class declaration

1. **public class** JMenuBar **extends** JComponent **implements** MenuElement,

Accessible

JMenu class declaration

1. **public class** JMenu **extends** JMenuItem **implements** MenuElement, Accessible

JMenuItem class declaration

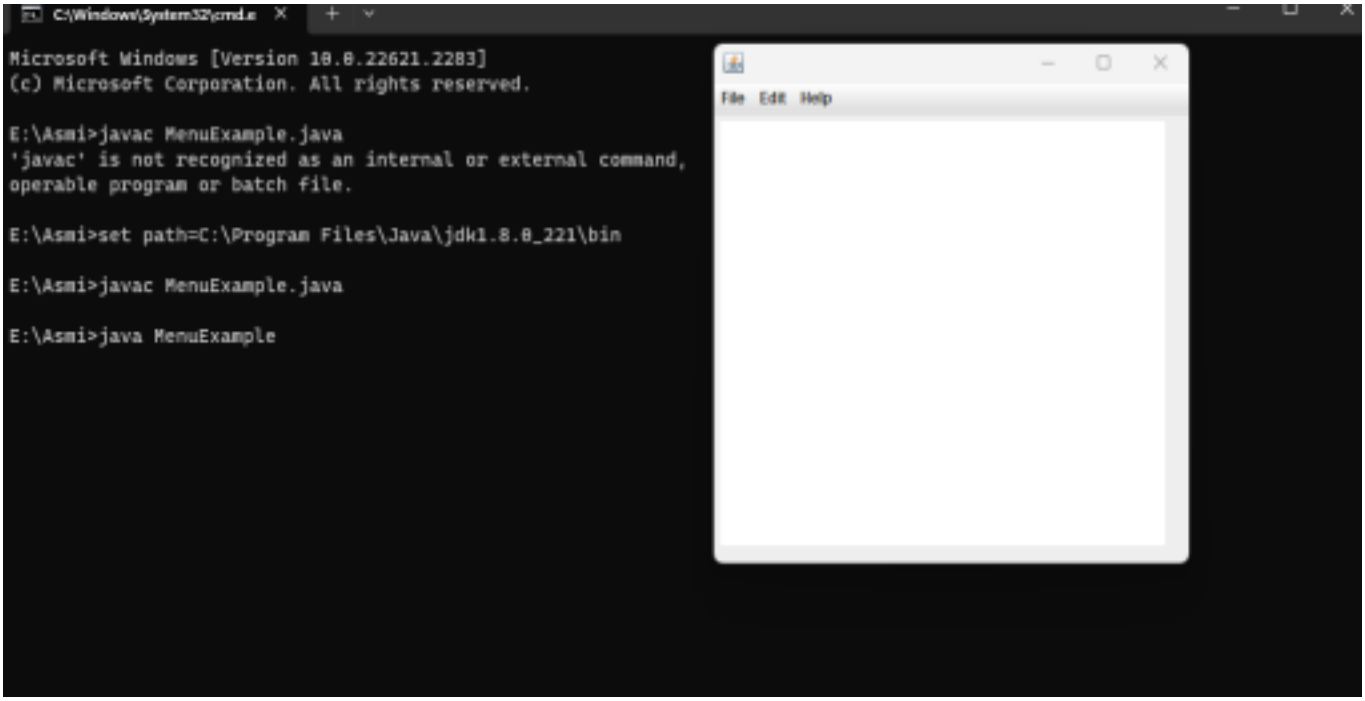
1. **public class** JMenuItem **extends** AbstractButton **implements** Accessible, MenuElement

Example of creating Edit menu for Notepad:

```
import javax.swing.*;

import java.awt.event.*;

public class MenuExample implements ActionListener{

JFrame f;

JMenuBar mb;

JMenu file,edit,help;

JMenuItem cut,copy,paste,selectAll;

JTextArea ta;

MenuExample(){

f=new JFrame();

cut=new JMenuItem("cut");

copy=new JMenuItem("copy");

paste=new JMenuItem("paste");
```

```
selectAll=new JMenuItem("selectAll");

cut.addActionListener(this);

copy.addActionListener(this);

paste.addActionListener(this);

selectAll.addActionListener(this);
mb=new JMenuBar();

file=new JMenu("File");

edit=new JMenu("Edit");

help=new JMenu("Help");

edit.add(cut);edit.add(copy);edit.add(paste);edit.add(selectAll);

mb.add(file);mb.add(edit);mb.add(help);

ta=new JTextArea();

ta.setBounds(5,5,360,320);

f.add(mb);f.add(ta);

f.setJMenuBar(mb);

f.setLayout(null);

f.setSize(400,400);

f.setVisible(true);

}
```

```
public void actionPerformed(ActionEvent e) {

if(e.getSource()==cut)

ta.cut();

if(e.getSource()==paste)

ta.paste();

if(e.getSource()==copy)
ta.copy();

if(e.getSource()==selectAll)

ta.selectAll();

}

public static void main(String[] args) {

   new MenuExample();

}

}
```



JPopupMenu

PopupMenu can be dynamically popped up at specific position within a component. It inherits the JComponent class.

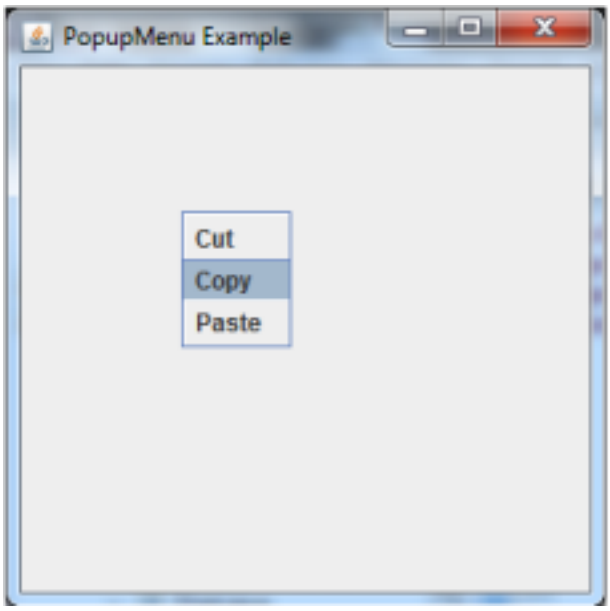
JPopupMenu class declaration

Let's see the declaration for javax.swing.JPopupMenu class.

1. **public class** JPopupMenu **extends** JComponent **implements** Accessible, MenuElement

Commonly used Constructors:

Constructor	Description
JPopupMenu()	Constructs a JPopupMenu without an "invoker".
JPopupMenu(String label)	Constructs a JPopupMenu with the specified title.



JPopupMenu Example

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

class PopupMenuExample

{

    PopupMenuExample(){

        final JFrame f= new JFrame("PopupMenu Example");

        final JPopupMenu popupmenu = new JPopupMenu("Edit");

        JMenuItem cut = new JMenuItem("Cut");

        JMenuItem copy = new JMenuItem("Copy");

        JMenuItem paste = new JMenuItem("Paste");
```

```
        popupmenu.add(cut); popupmenu.add(copy); popupmenu.add(paste);
        f.addMouseListener(new MouseAdapter() {
          public void mouseClicked(MouseEvent e) {
             popupmenu.show(f , e.getX(), e.getY());
          }
        });
        f.add(popupmenu);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
public static void main(String args[])
{
      new
PopupMenuExample(); }}
```

Mouse Events

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.*;



public class

MouseListenerExample

extends JFrame implements

MouseListener {



    private JPanel panel;

    private JLabel label;



    // Constructor to set up

the GUI and event handling

    public
```

```
MouseListenerExample() {

        // Set up the frame


setTitle("MouseListener

Example");

        setSize(400, 300);


setDefaultCloseOperation(JFr

ame.EXIT_ON_CLOSE);


        // Create panel and

label

        panel = new

JPanel();

        label = new
```

```
JLabel("Perform mouse

actions on the panel");



        // Add

MouseListener to the panel



panel.addMouseListener(this)

;



        // Add the label to

the panel

        panel.add(label);



        // Add the panel to

the frame
```

```
        add(panel);


        setVisible(true);
    }


    // Implementing
methods of MouseListener
    @Override
    public void
mouseClicked(MouseEvent e)
{


label.setText("Mouse Clicked
at X: " + e.getX() + " Y: " +
e.getY());
```

```
        }


        @Override
        public void
mousePressed(MouseEvent e)
{


label.setText("Mouse Pressed
at X: " + e.getX() + " Y: " +
e.getY());
        }


        @Override
        public void
mouseReleased(MouseEvent
```

```
e) {


label.setText("Mouse
Released at X: " + e.getX() +
" Y: " + e.getY());
        }


        @Override
        public void
mouseEntered(MouseEvent e)
{


label.setText("Mouse Entered
the Panel");
        }
```

```
        @Override
        public void
mouseExited(MouseEvent e)
{

label.setText("Mouse Exited
the Panel");
        }

        // Main method to run
the program
        public static void
main(String[] args) {
            new
```

```
MouseListenerExample();

        }

    }
```