

JPanel in Java Swing Example

A panel in swing is similar to Panel in AWT, is a lightweight container that is designed to group a set of components, including other panels. It is visually represented as window that does not contain a title bar, menu bar or border. It is simplest of all the containers. Its default layout manager is FlowLayout.

Panels are represented by objects created from JPanel class by calling the constructor.

public JPanel ()

After the panel has been created, other components can be added to **JPanel** object by calling its **add (component)** method inherited from the **Container** class. The following code fragment demonstrates creating a panel and adding two buttons to it.

```
JPanel p = new JPanel();  
p.add (new JButton ("OK")) ;  
p.add (new JButton ("Cancel") ) ;
```

Using panels, you can design complex GUIs.

```
import javax.swing.*;  
import java.awt.event.*;  
import java.awt.*;  
class JPanelExample extends JFrame  
{  
    JPanel panel1,panel2 ;  
    JTextField txtData;  
    JButton[] btnData = new JButton[12];  
    JPanelExample()  
    {  
        panel1=new JPanel();  
        txtData = new JTextField(20);  
        panel1.add(txtData);  
        add(panel1,"North");  
        panel2=new JPanel(new GridLayout(3,4));  
        for(int i=0;i<=9;i++)  
        {  
            btnData[i]=new JButton(""+i);  
            panel2.add(btnData[i]);  
        }  
        add(panel2,"Center");  
    }  
}  
class JPanelJavaExample  
{  
    public static void main(String[] args)  
    {  
        JPanelExample frame = new JPanelExample();  
        frame.setTitle("JPanel Java Example");  
        frame.setBounds(100,200,220,200);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.pack();  
        frame.setVisible(true);  
    }  
}
```

