**8.4 Panel :-**

A panel is both a component and a container as it is a subclass of both Component and Container. As a component it can be added to another container and as a container it can be added with components. Panels work both ways.

The default layout manager for panel is FlowLayout which can be changed as per the requirement of the layout.

It is known as a child window. As a child window, it does not have a border.

Panels can be nested (known as nested panels) for better layout presentation.

Following program illustrates the panels' usage in layout management. In the following program, three buttons are added to the north (top) of the frame and three buttons to the south (bottom) of the frame. Without panels, this arrangement is not possible with mere layout managers.

The applet itself is a panel. In fact, it extends the panel class so that the panel will run in a browser. The applet or panel is then divided into two sections. It's layout is set for 1 row and 2 columns with 10 pixels between each row and column. The left side of this applet is a panel that has a layout of its own. The right side is a canvas. A canvas is a component that is used for creating custom graphic components. In this applet it is used to paint a background color based on the user's input values for red, green, and blue. (These values may range from 0 to 255). Let's return to the left panel again. It's layout is set for 3 rows and two columns with 10 pixels between each row and column. The following components are added: label, textbox, label, textbox, label, and textbox. They are added from left to right, top to bottom.

Panels are used to organize the components on the main panel (or applet if it will be run in a browser).

Additional panels can be nested in the main panel or applet in the layout of your choice.

Components are then added to these panels. In this example, 3 labels and 3 textboxes have been added to a panel. Then this panel plus a canvas have been added to the applet (panel).

Layout managers determine where the components will be placed in relation to one another. If the platform is changed for the applet or application, the layout manager takes care of the necessary adjustments.

In this program, you will see that three buttons have been added to the panel. The panel and a text area added to the frame. The position for the panel on the frame has been specified south of the frame by using **BorderLayout.**SOUTH since the position of the text area has been specified the center of the frame using **BorderLayout.**CENTER. Here, the add() method has been used for both operations (add buttons to the panel and add panel and text area to the frame).

**BorderLayout :**  
**BorderLayout** is the class of the *java.awt.\*;* package which is used to arranging and resizing it's components to fit in five rigions : north, south, east, west and center. Each region may contain only one component. All regions are represented by the NORTH, SOUTH, EAST, WEST and CENTER constants of the **BorderLayout** class.

|  |
| --- |
| **import**java.awt.\*; **import**java.awt.event.\*; **public class**CreateContainer{   **public static void**main(String[] args){   Panel panel = **new**Panel();   panel.add(**new**Button("Button 1"));   panel.add(**new**Button("Button 2"));   panel.add(**new**Button("Button 3"));   Frame frame = **new**Frame("Container Frame");   TextArea txtArea = **new**TextArea();   frame.add(txtArea, BorderLayout.CENTER);   frame.add(panel, BorderLayout.SOUTH);   frame.setSize(400,400);   frame.setVisible(**true**);   frame.addWindowListener(**new**WindowAdapter(){   **public void**windowClosing(WindowEvent we){   System.exit(0);   }   });   }} |