GroupLayout

javatpoint.com/java-grouplayout

GroupLayout *groups its components and places them in a Container hierarchically.* The grouping is done by instances of the Group class.

Group is an abstract class and two concrete classes which implement this Group class are SequentialGroup and ParallelGroup.

SequentialGroup positions its child sequentially one after another where as ParallelGroup aligns its child on top of each other.

The GroupLayout class provides methods such as createParallelGroup() and createSequentialGroup() to create groups.

GroupLayout treats each axis independently. That is, there is a group representing the horizontal axis, and a group representing the vertical axis. Each component must exists in both a horizontal and vertical group, otherwise an IllegalStateException is thrown during layout, or when the minimum, preferred or maximum size is requested.

Nested Classes

Modifier and Type	Class	Description
static class	GroupLayout.Alignment	Enumeration of the possible ways ParallelGroup can align its children.
class	GroupLayout.Group	Group provides the basis for the two types of operations supported by GroupLayout: laying out components one after another (SequentialGroup) or aligned (ParallelGroup).
class	GroupLayout.ParallelGroup	It is a Group that aligns and sizes it's children.
class	GroupLayout.SequentialGroup	It is a Group that positions and sizes its elements sequentially, one after another.

Fields

Modifier and Type	Field	Description
static int	DEFAULT_SIZE	It indicates the size from the component or gap should be used for a particular range value.
static int	PREFERRED_SIZE	It indicates the preferred size from the component or gap should be used for a particular range value.

Constructors

GroupLayout(Container host)	It creates a GroupLayout for the specified Container.

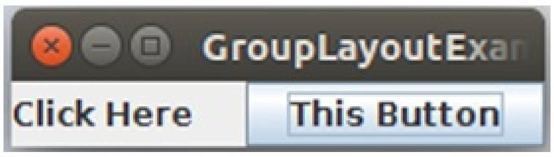
Useful Methods

Modifier and Type	Field	Description
void	addLayoutComponent(Component component, Object constraints)	It notify that a Component has been added to the parent container.
void	addLayoutComponent(String name, Component component)	It notify that a Component has been added to the parent container.
GroupLayout.ParallelGroup	createBaselineGroup(boolean resizable, boolean anchorBaselineToTop)	It creates and returns a ParallelGroup that aligns it's elements along the baseline.
GroupLayout.ParallelGroup	createParallelGroup()	It creates and returns a ParallelGroup with an alignment of Alignment.LEADING

GroupLayout.ParallelGroup	createParallelGroup(GroupLayout.Alignment alignment)	It creates and returns a ParallelGroup with the specified alignment.
GroupLayout.ParallelGroup	createParallelGroup(GroupLayout.Alignment alignment, boolean resizable)	It creates and returns a ParallelGroup with the specified alignment and resize behavior.
GroupLayout.SequentialGroup	createSequentialGroup()	It creates and returns a SequentialGroup.
boolean	getAutoCreateContainerGaps()	It returns true if gaps between the container and components that border the container are automatically created.
boolean	getAutoCreateGaps()	It returns true if gaps between components are automatically created.
boolean	getHonorsVisibility()	It returns whether component visiblity is considered when sizing and positioning components.
float	getLayoutAlignmentX(Container parent)	It returns the alignment along the x axis.
float	getLayoutAlignmentY(Container parent)	It returns the alignment along the y axis.
Dimension	maximumLayoutSize(Container parent)	It returns the maximum size for the specified container.

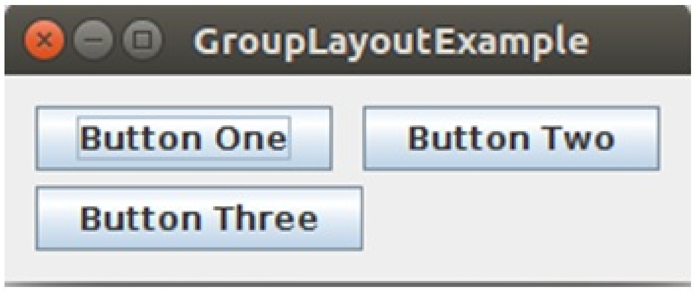
Example

```
publicclass GroupExample {
    publicstaticvoid main(String[] args) {
        JFrame frame = new JFrame("GroupLayoutExample");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container contentPanel = frame.getContentPane();
        GroupLayout groupLayout = new GroupLayout(contentPanel);
        contentPanel.setLayout(groupLayout);
        JLabel clickMe = new JLabel("Click Here");
        JButton button = new JButton("This Button");
        groupLayout.setHorizontalGroup(
                    groupLayout.createSequentialGroup()
                                .addComponent(clickMe)
                                .addGap(10, 20, 100)
                                .addComponent(button));
        groupLayout.setVerticalGroup(
                     groupLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)
                                .addComponent(clickMe)
                                .addComponent(button));
        frame.pack();
        frame.setVisible(true);
    }
}
```



Example 2

```
import java.awt.Container;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
importstatic javax.swing.GroupLayout.Alignment.*;
publicclass GroupExample2 {
    publicstaticvoid main(String[] args) {
        JFrame frame = new JFrame("GroupLayoutExample");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container myPanel = frame.getContentPane();
        GroupLayout groupLayout = new GroupLayout(myPanel);
        groupLayout.setAutoCreateGaps(true);
        groupLayout.setAutoCreateContainerGaps(true);
        myPanel.setLayout(groupLayout);
        JButton b1 = new JButton("Button One");
        JButton b2 = new JButton("Button Two");
        JButton b3 = new JButton("Button Three");
        groupLayout.setHorizontalGroup(groupLayout.createSequentialGroup()
                .addGroup(groupLayout.createParallelGroup(LEADING).addComponent(b1).addComponent(b3))
                .addGroup(groupLayout.createParallelGroup(TRAILING).addComponent(b2)));
        groupLayout.setVerticalGroup(groupLayout.createSequentialGroup()
                .addGroup(groupLayout.createParallelGroup(BASELINE).addComponent(b1).addComponent(b2))
                .addGroup(groupLayout.createParallelGroup(BASELINE).addComponent(b3)));
        frame.pack();
        frame.setVisible(true);
    }
}
```


GroupLayoutExample
Button One
Button Two
Button Three