

SWING - BORDERLAYOUT CLASS

http://www.tutorialspoint.com/swing/swing_borderlayout.htm

Copyright © tutorialspoint.com

Introduction

The class **BorderLayout** arranges the components to fit in the five regions: east, west, north, south and center. Each region is can contain only one component and each component in each region is identified by the corresponding constant NORTH, SOUTH, EAST, WEST, and CENTER.

Class declaration

Following is the declaration for **java.awt.BorderLayout** class:

```
public class BorderLayout
    extends Object
    implements LayoutManager2, Serializable
```

Field

Following are the fields for **java.awt.BorderLayout** class:

- **static String AFTER_LAST_LINE** -- Synonym for PAGE_END.
- **static String AFTER_LINE_ENDS** -- Synonym for LINE_END.
- **static String BEFORE_FIRST_LINE** -- Synonym for PAGE_START.
- **static String BEFORE_LINE_BEGINS** -- Synonym for LINE_START.
- **static String CENTER** -- The center layout constraint *middleofcontainer*.
- **static String EAST** -- The east layout constraint *rightsideofcontainer*.
- **static String LINE_END** -- The component goes at the end of the line direction for the layout.
- **static String LINE_START** -- The component goes at the beginning of the line direction for the layout.
- **static String NORTH** -- The north layout constraint *topofcontainer*.
- **static String PAGE_END** -- The component comes after the last line of the layout's content.
- **static String PAGE_START** -- The component comes before the first line of the layout's content.
- **static String SOUTH** -- The south layout constraint *bottomofcontainer*.
- **static String WEST** -- The west layout constraint *leftsideofcontainer*.

Class constructors

S.N.	Constructor & Description
1	BorderLayout Constructs a new border layout with no gaps between components.
2	BorderLayout(int hgap, int vgap) Constructs a border layout with the specified gaps between components.

Class methods

S.N.	Method & Description
1	void addLayoutComponent <i>Componentcomp, Objectconstraints</i> Adds the specified component to the layout, using the specified constraint object.
2	void addLayoutComponent <i>Stringname, Componentcomp</i> If the layout manager uses a per-component string, adds the component comp to the layout, associating it with the string specified by name.
3	int getHgap Returns the horizontal gap between components.
4	float getLayoutAlignmentX <i>Containerparent</i> Returns the alignment along the x axis.
5	float getLayoutAlignmentY <i>Containerparent</i> Returns the alignment along the y axis.
6	int getVgap Returns the vertical gap between components.
7	void invalidateLayout <i>Containertarget</i> Invalidates the layout, indicating that if the layout manager has cached information it should be discarded.
8	void layoutContainer <i>Containertarget</i> Lays out the container argument using this border layout.
9	Dimension maximumLayoutSize <i>Containertarget</i> Returns the maximum dimensions for this layout given the components in the specified target container.
10	Dimension minimumLayoutSize <i>Containertarget</i> Determines the minimum size of the target container using this layout manager.
11	Dimension preferredLayoutSize <i>Containertarget</i> Determines the preferred size of the target container using this layout manager, based on the components in the container.
12	void removeLayoutComponent <i>Componentcomp</i> Removes the specified component from this border layout.

13 **void setHgap***inthgap*

Sets the horizontal gap between components.

14 **void setVgap***intvgap*

Sets the vertical gap between components.

15 **String toString**

Returns a string representation of the state of this border layout.

Methods inherited

This class inherits methods from the following classes:

- java.lang.Object

BorderLayout Example

Create the following java program using any editor of your choice in say **D:/ > SWING > com > tutorialspoint > gui >**

SwingLayoutDemo.java

```
package com.tutorialspoint.gui;

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

public class SwingLayoutDemo {
    private JFrame mainFrame;
    private JLabel headerLabel;
    private JLabel statusLabel;
    private JPanel controlPanel;
    private JLabel msgLabel;

    public SwingLayoutDemo(){
        prepareGUI();
    }

    public static void main(String[] args){
        SwingLayoutDemo swingLayoutDemo = new SwingLayoutDemo();
        swingLayoutDemo.showBorderLayoutDemo();
    }

    private void prepareGUI(){
        mainFrame = new JFrame("Java SWING Examples");
        mainFrame.setSize(400,400);
        mainFrame.setLayout(new GridLayout(3, 1));

        headerLabel = new JLabel("",JLabel.CENTER );
        statusLabel = new JLabel("",JLabel.CENTER);

        statusLabel.setSize(350,100);
        mainFrame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent){
                System.exit(0);
            }
        });
        controlPanel = new JPanel();
        controlPanel.setLayout(new FlowLayout());
```

```

mainFrame.add(headerLabel);
mainFrame.add(controlPanel);
mainFrame.add(statusLabel);
mainFrame.setVisible(true);
}

private void showBorderLayoutDemo(){
    headerLabel.setText("Layout in action: BorderLayout");

    JPanel panel = new JPanel();
    panel.setBackground(Color.darkGray);
    panel.setSize(300,300);
    BorderLayout layout = new BorderLayout();
    layout.setHgap(10);
    layout.setVgap(10);
    panel.setLayout(layout);

    panel.add(new JButton("Center"),BorderLayout.CENTER);
    panel.add(new JButton("Line Start"),BorderLayout.LINE_START);
    panel.add(new JButton("Line End"),BorderLayout.LINE_END);
    panel.add(new JButton("East"),BorderLayout.EAST);
    panel.add(new JButton("West"),BorderLayout.WEST);
    panel.add(new JButton("North"),BorderLayout.NORTH);
    panel.add(new JButton("South"),BorderLayout.SOUTH);

    controlPanel.add(panel);

    mainFrame.setVisible(true);
}
}

```

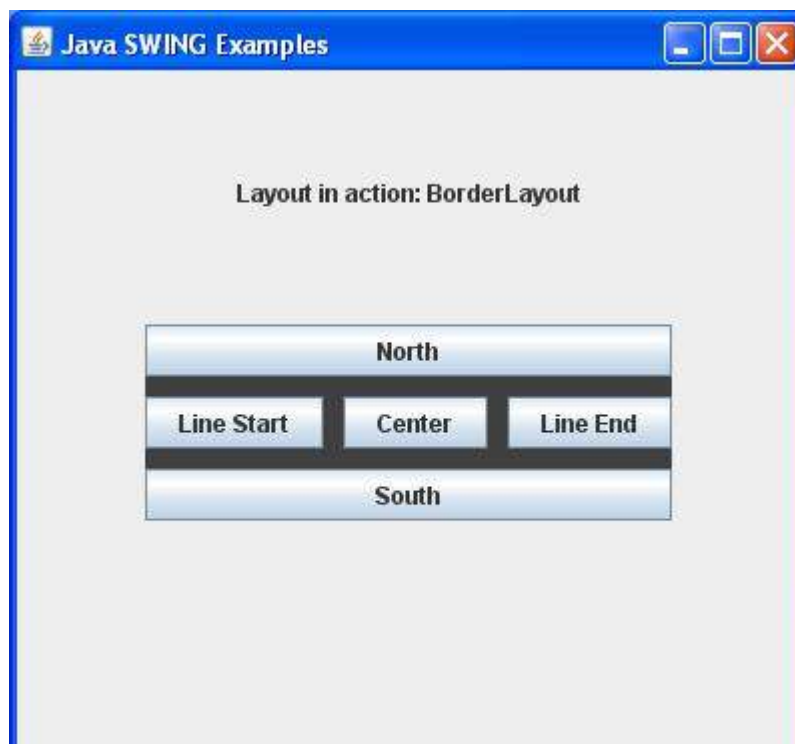
Compile the program using command prompt. Go to **D:/ > SWING** and type the following command.

```
D:\SWING>javac com\tutorialspoint\gui\SwingLayoutDemo.java
```

If no error comes that means compilation is successful. Run the program using following command.

```
D:\SWING>java com.tutorialspoint.gui.SwingLayoutDemo
```

Verify the following output



Loading [MathJax]/jax/output/HTML-CSS/jax.js