

AWT SCROLLBAR CLASS

http://www.tutorialspoint.com/awt/awt_scrollbar.htm

Copyright © tutorialspoint.com

Introduction

Scrollbar control represents a scroll bar component in order to enable user to select from range of values.

Class declaration

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

```
public class Scrollbar
    extends Component
    implements Adjustable, Accessible
```

Field

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

- **static int HORIZONTAL** --A constant that indicates a horizontal scroll bar.
- **static int VERTICAL** --A constant that indicates a vertical scroll bar.

Class constructors

S.N.	Constructor & Description
1	Scrollbar Constructs a new vertical scroll bar.
2	Scrollbar(int orientation) Constructs a new scroll bar with the specified orientation.
3	Scrollbar(int orientation, int value, int visible, int minimum, int maximum) Constructs a new scroll bar with the specified orientation, initial value, visible amount, and minimum and maximum values.

Class methods

S.N.	Method & Description
1	void addAdjustmentListener(AdjustmentListener l) Adds the specified adjustment listener to receive instances of AdjustmentEvent from this scroll bar.
2	void addNotify

Creates the Scrollbar's peer.

3

int getBlockIncrement

Gets the block increment of this scroll bar.

4

int getLineIncrement

Deprecated. As of JDK version 1.1, replaced by getUnitIncrement.

5

int getMaximum

Gets the maximum value of this scroll bar.

6

int getMinimum

Gets the minimum value of this scroll bar.

7

int getOrientation

Returns the orientation of this scroll bar.

8

int getPageIncrement

Deprecated. As of JDK version 1.1, replaced by getBlockIncrement.

9

int getUnitIncrement

Gets the unit increment for this scrollbar.

10

int getValue

Gets the current value of this scroll bar.

11

boolean

getValueIsAdjusting

Returns true if the value is in the process of changing as a result of actions being taken by the user.

12

int getVisible

Deprecated. As of JDK version 1.1, replaced by getVisibleAmount.

13

int getVisibleAmount

Gets the visible amount of this scroll bar.

14

protected String paramString

Returns a string representing the state of this Scrollbar.

15

protected void processAdjustmentEvent*AdjustmentEvent*

Processes adjustment events occurring on this scrollbar by dispatching them to any registered AdjustmentListener objects.

16

protected void processEvent*AWTEvent*

Processes events on this scroll bar.

17

void removeAdjustmentListener*AdjustmentListener*

Removes the specified adjustment listener so that it no longer receives instances of AdjustmentEvent from this scroll bar.

18

void setBlockIncrement*int*

Sets the block increment for this scroll bar.

19

void setLineIncrement*int*

Deprecated. As of JDK version 1.1, replaced by *setUnitIncrement*.

20

void setMaximum*int newMaximum*

Sets the maximum value of this scroll bar.

21

void setMinimum*int newMinimum*

Sets the minimum value of this scroll bar.

22

void setOrientation*int orientation*

Sets the orientation for this scroll bar.

23

void setPageIncrement*int*

Deprecated. As of JDK version 1.1, replaced by *setBlockIncrement*.

24

void setUnitIncrement*int*

Sets the unit increment for this scroll bar.

25

void setValue*int newValue*

Sets the value of this scroll bar to the specified value.

26

void setValuesAdjusting*booleanb*

Sets the valuesAdjusting property.

27

void setValues*intvalue, intvisible, intminimum, intmaximum*

Sets the values of four properties for this scroll bar: value, visibleAmount, minimum, and maximum.

28

void setVisibleAmount*intnewAmount*

Sets the visible amount of this scroll bar.

29

AccessibleContext getAccessibleContext

Gets the AccessibleContext associated with this Scrollbar.

30

AdjustmentListener[] getAdjustmentListeners

Returns an array of all the adjustment listeners registered on this scrollbar.

31

<T extends EventListener>T[] getListeners*Class < T > listenerType*

Returns an array of all the objects currently registered as FooListeners upon this Scrollbar.

Methods inherited

This class inherits methods from the following classes:

- java.awt.Component
- java.lang.Object

Choice Example

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

AwtControlDemo

```
package com.tutorialspoint.gui;

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

public class AwtControlDemo {

    private Frame mainFrame;
    private Label headerLabel;
    private Label statusLabel;
    private Panel controlPanel;
```

```

public AwtControlDemo(){
    prepareGUI();
}

public static void main(String[] args){
    AwtControlDemo awtControlDemo = new AwtControlDemo();
    awtControlDemo.showScrollbarDemo();
}

private void prepareGUI(){
    mainFrame = new Frame("Java AWT Examples");
    mainFrame.setSize(400,400);
    mainFrame.setLayout(new GridLayout(3, 1));
    mainFrame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent windowEvent){
            System.exit(0);
        }
    });
    headerLabel = new Label();
    headerLabel.setAlignment(Label.CENTER);
    statusLabel = new Label();
    statusLabel.setAlignment(Label.CENTER);
    statusLabel.setSize(350,100);

    controlPanel = new Panel();
    controlPanel.setLayout(new FlowLayout());

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

private void showScrollbarDemo(){
    headerLabel.setText("Control in action: Scrollbar");

    final Scrollbar horizontalScroller = new Scrollbar(Scrollbar.HORIZONTAL);
    final Scrollbar verticalScroller = new Scrollbar();
    verticalScroller.setOrientation(Scrollbar.VERTICAL);
    horizontalScroller.setMaximum(100);
    horizontalScroller.setMinimum(1);
    verticalScroller.setMaximum(100);
    verticalScroller.setMinimum(1);

    horizontalScroller.addAdjustmentListener(new AdjustmentListener() {

        @Override
        public void adjustmentValueChanged(AdjustmentEvent e) {
            statusLabel.setText("Horizontal: "
                +horizontalScroller.getValue()
                +" ,Vertical: "
                + verticalScroller.getValue());
        }
    });

    verticalScroller.addAdjustmentListener(new AdjustmentListener() {

        @Override
        public void adjustmentValueChanged(AdjustmentEvent e) {
            statusLabel.setText("Horizontal: "
                +horizontalScroller.getValue()
                +" ,Vertical: "+ verticalScroller.getValue());
        }
    });

    controlPanel.add(horizontalScroller);
    controlPanel.add(verticalScroller);

    mainFrame.setVisible(true);
}

```

```
}  
}
```

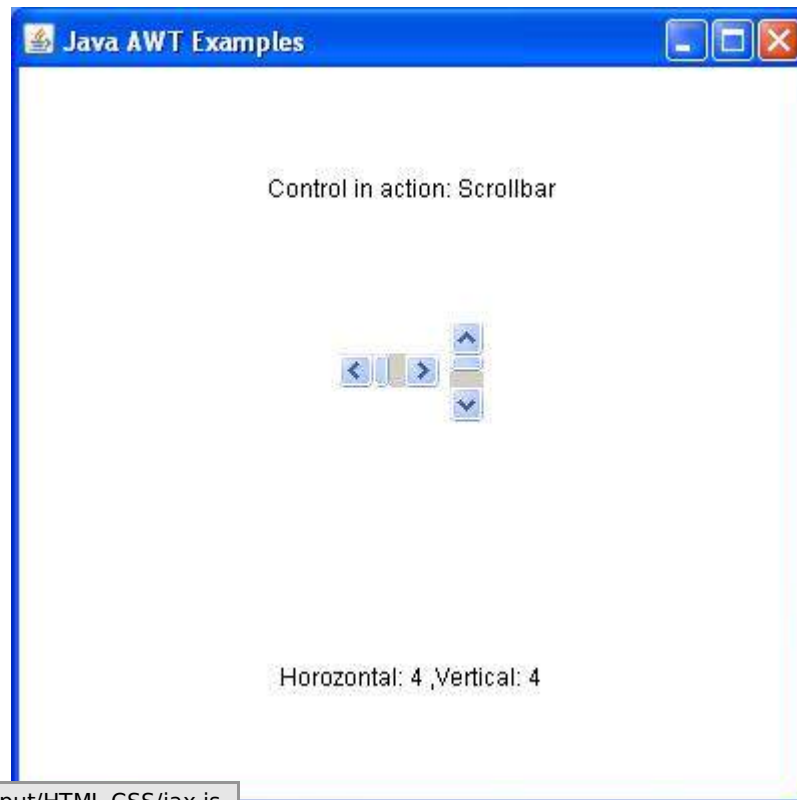
Compile the program using command prompt. Go to **D:/ > AWT** and type the following command.

```
D:\AWT>javac com\tutorialspoint\gui\AwtControlDemo.java
```

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

```
D:\AWT>java com.tutorialspoint.gui.AwtControlDemo
```

Verify the following output



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