

SWING - JSPINNER CLASS

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

Copyright © tutorialspoint.com

Introduction

The class **JSpinner** is a component which lets the user select a number or an object value from an ordered sequence using an input field.

Class declaration

Following is the declaration for **javax.swing.JSpinner** class:

```
public class JSpinner
    extends JComponent
    implements Accessible
```

Class constructors

S.N.	Constructor & Description
1	JSpinner Constructs a spinner with an Integer SpinnerNumberModel with initial value 0 and no minimum or maximum limits.
2	JSpinnerSpinnerModelmodel Constructs a complete spinner with pair of next/previous buttons and an editor for the SpinnerModel.

Class methods

S.N.	Method & Description
1	void addChangeListenerChangeListenerlistener Adds a listener to the list that is notified each time a change to the model occurs.
2	void commitEdit Commits the currently edited value to the SpinnerModel.
3	protected JComponent createEditorSpinnerModelmodel This method is called by the constructors to create the JComponent that displays the current value of the sequence.
4	protected void fireStateChanged Sends a ChangeEvent, whose source is this JSpinner, to each ChangeListener.
5	AccessibleContext getAccessibleContext Gets the AccessibleContext for the JSpinner.

- 6 **ChangeListener[] getChangeListeners**
Returns an array of all the ChangeListeners added to this JSpinner with addChangeListener.
- 7 **JComponent getEditor**
Returns the component that displays and potentially changes the model's value.
- 8 **SpinnerModel getModel**
Returns the SpinnerModel that defines this spinners sequence of values.
- 9 **Object getNextValue**
Returns the object in the sequence that comes after the object returned by getValue.
- 10 **Object getPreviousValue**
Returns the object in the sequence that comes before the object returned by getValue.
- 11 **SpinnerUI getUI**
Returns the look and feel **L&F** object that renders this component.
- 12 **String getUIClassID**
Returns the suffix used to construct the name of the look and feel **L&F** class used to render this component.
- 13 **Object getValue**
Returns the current value of the model, typically this value is displayed by the editor.
- 14 **void removeChangeListenerChangeListener**
Removes a ChangeListener from this spinner.
- 15 **void setEditorJComponenteditor**
Changes the JComponent that displays the current value of the SpinnerModel.
- 16 **void setModelSpinnerModelmodel**
Changes the model that represents the value of this spinner.
- 17 **void setUISpinnerUIui**
Sets the look and feel **L&F** object that renders this component.
- 18 **void setValueObjectvalue**
Changes current value of the model, typically this value is displayed by the editor.
- 19 **void updateUI**

Resets the UI property with the value from the current look and feel.

Methods inherited

This class inherits methods from the following classes:

- javax.swing.JComponent
- java.awt.Container
- java.awt.Component
- java.lang.Object

JSpinner Example

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

SwingControlDemo.java

```
package com.tutorialspoint.gui;

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

public class SwingControlDemo {

    private JFrame mainFrame;
    private JLabel headerLabel;
    private JLabel statusLabel;
    private JPanel controlPanel;

    public SwingControlDemo(){
        prepareGUI();
    }

    public static void main(String[] args){
        SwingControlDemo swingControlDemo = new SwingControlDemo();
        swingControlDemo.showSpinnerDemo();
    }

    private void prepareGUI(){
        mainFrame = new JFrame("Java Swing 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 JLabel("", JLabel.CENTER);
        statusLabel = new JLabel("", JLabel.CENTER);

        statusLabel.setSize(350,100);

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

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

```

private void showSpinnerDemo(){
    headerLabel.setText("Control in action: JSpinner");
    SpinnerModel spinnerModel =
        new SpinnerNumberModel(10, //initial value
                                0, //min
                                100, //max
                                1); //step
    JSpinner spinner = new JSpinner(spinnerModel);
    spinner.addChangeListener(new ChangeListener() {
        public void stateChanged(ChangeEvent e) {
            statusLabel.setText("Value : "
                                + ((JSpinner)e.getSource()).getValue());
        }
    });
    controlPanel.add(spinner);
    mainFrame.setVisible(true);
}
}

```

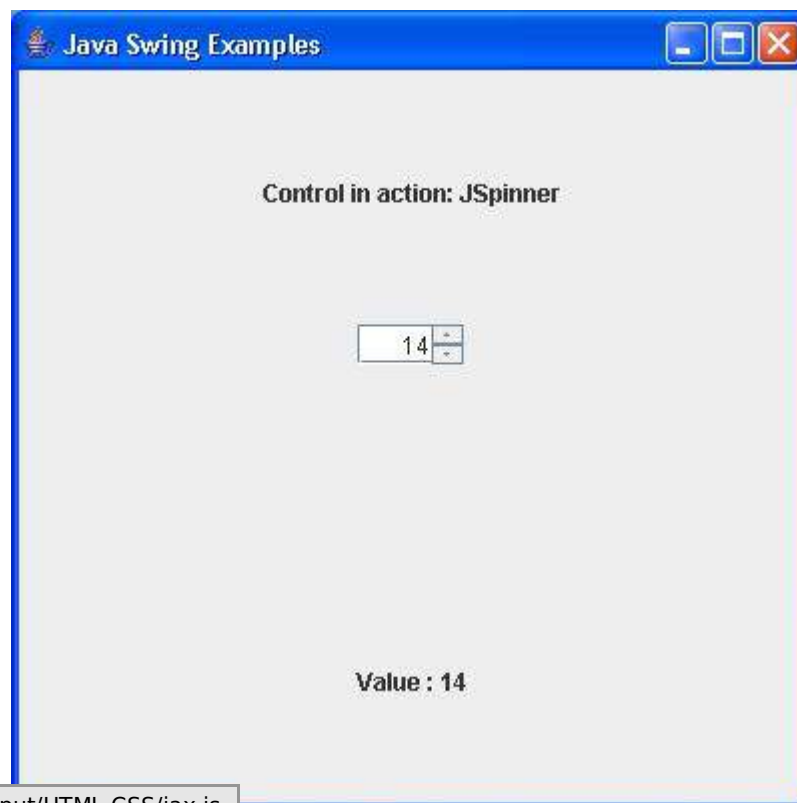
Compile the program using command prompt. Go to **D:/ > SWING** and type the following command.

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

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

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

Verify the following output



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