

# SWING - JOPTIONPANE CLASS

[http://www.tutorialspoint.com/swing/swing\\_joptionpane.htm](http://www.tutorialspoint.com/swing/swing_joptionpane.htm)

Copyright © tutorialspoint.com

## Introduction

The class **JOptionPane** is a component which provides standard methods to pop up a standard dialog box for a value or informs user of something.

## Class declaration

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

```
public class JOptionPane
    extends JComponent
    implements Accessible
```

## Field

Following are the fields for **javax.swing.JOptionPane** class:

- **static int CANCEL\_OPTION** -- Return value from class method if CANCEL is chosen.
- **static int CLOSED\_OPTION** -- Return value from class method if user closes window without selecting anything, more than likely this should be treated as either a CANCEL\_OPTION or NO\_OPTION.
- **static int DEFAULT\_OPTION** -- Type meaning Look and Feel should not supply any options -- only use the options from the JOptionPane.
- **static int ERROR\_MESSAGE** -- Used for error messages.
- **protected Icon icon** -- Icon used in pane.
- **static string ICON\_PROPERTY** -- Bound property name for icon.
- **static int INFORMATION\_MESSAGE** -- Used for information messages.
- **static string INITIAL\_SELECTION\_VALUE\_PROPERTY** -- Bound property name for initialSelectionValue.
- **static string INITIAL\_VALUE\_PROPERTY** -- Bound property name for initialValue.
- **protected Object initialSelectionValue** -- Initial value to select in selectionValues.
- **protected Object initialValue** -- Value that should be initially selected in options.
- **static string INPUT\_VALUE\_PROPERTY** -- Bound property name for inputValue.
- **protected Object inputValue** -- Value the user has input.
- **protected Object message** -- Message to display.
- **static string MESSAGE\_PROPERTY** -- Bound property name for message.
- **static string MESSAGE\_TYPE\_PROPERTY** -- Bound property name for type.
- **static int OK\_CANCEL\_OPTION** -- Type used for showConfirmDialog.
- **protected int messageType** -- Message type.
- **static int NO\_OPTION** -- Return value from class method if NO is chosen.
- **static int OK\_OPTION** -- Return value from class method if OK is chosen.
- **static string OPTION\_TYPE\_PROPERTY** -- Bound property name for optionType.
- **protected Object[] options** -- Options to display to the user.
- **static string OPTIONS\_PROPERTY** -- Bound property name for option.
- **protected int optionType** -- Option type, one of DEFAULT\_OPTION, YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION or OK\_CANCEL\_OPTION.
- **static int PLAIN\_MESSAGE** -- No icon is used.
- **static int QUESTION\_MESSAGE** -- Used for questions.
- **static string SELECTION\_VALUES\_PROPERTY** -- Bound property name for selectionValues.

- **protected Object[] selectionValues** -- Array of values the user can choose from.
- **static Object UNINITIALIZED\_VALUE** -- Indicates that the user has not yet selected a value.
- **protected Object value** -- Currently selected value, will be a valid option, or UNINITIALIZED\_VALUE or null.
- **static string VALUE\_PROPERTY** -- Bound property name for value.
- **static string WANTS\_INPUT\_PROPERTY** -- Bound property name for wantsInput.
- **protected boolean wantsInput** -- If true, a UI widget will be provided to the user to get input.
- **static int WARNING\_MESSAGE** -- Used for warning messages.
- **static int YES\_NO\_CANCEL\_OPTION** -- Type used for showConfirmDialog.
- **static int YES\_NO\_OPTION** -- Type used for showConfirmDialog.
- **static int YES\_OPTION** -- Return value from class method if YES is chosen.

## Class constructors

S.N.	Constructor & Description
1	<b>JOptionPane</b> Creates a JOptionPane with a test message.
2	<b>JOptionPaneObjectmessage</b> Creates a instance of JOptionPane to display a message using the plain-message message type and the default options delivered by the UI.
3	<b>JOptionPaneObjectmessage, intmessageType</b> Creates an instance of JOptionPane to display a message with the specified message type and the default options
4	<b>JOptionPaneObjectmessage, intmessageType, intoptionType</b> Creates an instance of JOptionPane to display a message with the specified message type and options.
5	<b>JOptionPaneObjectmessage, intmessageType, intoptionType, Iconicon</b> Creates an instance of JOptionPane to display a message with the specified message type, options, and icon.
6	<b>JOptionPaneObjectmessage, intmessageType, intoptionType, Iconicon, Object[]options</b> Creates an instance of JOptionPane to display a message with the specified message type, icon, and options.
7	<b>JOptionPaneObjectmessage, intmessageType, intoptionType, Iconicon, Object[]options, ObjectinitialValue</b> Creates an instance of JOptionPane to display a message with the specified message type, icon, and options, with the initially-selected option specified.

## Class methods

S.N.	Method & Description
1	<b>JDialog createDialogComponentparentComponent, Stringtitle</b> Creates and returns a new JDialog wrapping this centered on the parentComponent in the parentComponent's frame.
2	<b>JDialog createDialogStringtitle</b>

Creates and returns a new parentless JDialog with the specified title.

3 **JInternalFrame createInternalFrame***ComponentparentComponent, Stringtitle*

Creates and returns an instance of JInternalFrame.

4 **AccessibleContext getAccessibleContext**

Returns the AccessibleContext associated with this JOptionPane.

5 **static JDesktopPane getDesktopPaneForComponent***ComponentparentComponent*

Returns the specified component's desktop pane.

6 **static Frame getFrameForComponent***ComponentparentComponent*

Returns the specified component's Frame.

7 **Icon getIcon**

Returns the icon this pane displays.

8 **Object getInitialSelectionValue**

Returns the input value that is displayed as initially selected to the user.

9 **Object getInitialValue**

Returns the initial value.

10 **Object getInputValue**

Returns the value the user has input, if wantsInput is true.

11 **int getMaxCharactersPerLineCount**

Returns the maximum number of characters to place on a line in a message.

12 **Object getMessage**

Returns the message-object this pane displays.

13 **int getMessageType**

Returns the message type.

14 **Object[] getOptions**

Returns the choices the user can make.

15 **int getOptionType**

Returns the type of options that are displayed.

16 **static Frame getRootFrame**

Returns the Frame to use for the class methods in which a frame is not provided.

17 **Object[] getSelectionValues**

Returns the input selection values.

18 **OptionPaneUI getUI**

Returns the UI object which implements the L&F for this component.

19 **String getUIClassID**

Returns the name of the UI class that implements the L&F for this component.

20 **Object getValue**

Returns the value the user has selected.

21 **boolean getWantsInput**

Returns the value of the wantsInput property.

22 **protected String paramString**

Returns a string representation of this JOptionPane.

23 **void selectInitialValue**

Requests that the initial value be selected, which will set focus to the initial value.

24 **void setIconIconnewIcon**

Sets the icon to display.

25 **void setInitialSelectionValueObjectnewValue**

Sets the input value that is initially displayed as selected to the user.

26 **void setInitialValueObjectnewInitialValue**

Sets the initial value that is to be enabled -- the Component that has the focus when the pane is initially displayed.

27 **void setInputValueObjectnewValue**

Sets the input value that was selected or input by the user.

28 **void setMessageObjectnewMessage**

Sets the option pane's message-object.

29 **void setMessageTypeintnewType**

Sets the option pane's message type.

30 **void setOptionsObject[]newOptions**

Sets the options this pane displays.

31 **void setOptionTypeintnewType**

Sets the options to display.

32 **static void setRootFrameFramenewRootFrame**

Sets the frame to use for class methods in which a frame is not provided.

33 **void setSelectionValuesObject[]newValues**

Sets the input selection values for a pane that provides the user with a list of items to choose from.

34 **void setUIOptionPaneUIui**

Sets the UI object which implements the L&F for this component.

35 **void setValueObjectnewValue**

Sets the value the user has chosen.

36	<b>void setWantsInput</b> <i>booleannewValue</i>	Sets the wantsInput property.
37	<b>static int showConfirmDialog</b> <i>ComponentparentComponent, Objectmessage</i>	Brings up a dialog with the options Yes, No and Cancel; with the title, Select an Option.
38	<b>static int showConfirmDialog</b> <i>ComponentparentComponent, Objectmessage, Stringtitle, intoptionType</i>	Brings up a dialog where the number of choices is determined by the optionType parameter.
39	<b>static int showConfirmDialog</b> <i>ComponentparentComponent, Objectmessage, Stringtitle, intoptionType, intmessageType</i>	Brings up a dialog where the number of choices is determined by the optionType parameter, where the messageType parameter determines the icon to display.
40	<b>static int showConfirmDialog</b> <i>ComponentparentComponent, Objectmessage, Stringtitle, intoptionType, intmessageType, Iconicon</i>	Brings up a dialog with a specified icon, where the number of choices is determined by the optionType parameter.
41	<b>static String showInputDialog</b> <i>ComponentparentComponent, Objectmessage</i>	Shows a question-message dialog requesting input from the user parented to parentComponent.
42	<b>static String showInputDialog</b> <i>ComponentparentComponent, Objectmessage, ObjectinitialSelectionValue</i>	Shows a question-message dialog requesting input from the user and parented to parentComponent.
43	<b>static String showInputDialog</b> <i>ComponentparentComponent, Objectmessage, Stringtitle, intessageType</i>	Shows a dialog requesting input from the user parented to parentComponent with the dialog having the title title and message type messageType.
44	<b>static Object showInputDialog</b> <i>ComponentparentComponent, Objectmessage, Stringtitle, intmessageType, Iconicon, Object[]selectionValues, ObjectinitialSelectionValue</i>	Prompts the user for input in a blocking dialog where the initial selection, possible selections, and all other options can be specified.
45	<b>static String showInputDialog</b> <i>Objectmessage</i>	Shows a question-message dialog requesting input from the user.
46	<b>static String showInputDialog</b> <i>Objectmessage, ObjectinitialSelectionValue</i>	Shows a question-message dialog requesting input from the user, with the input value initialized to initialSelectionValue.
47	<b>static int showInternalConfirmDialog</b> <i>ComponentparentComponent, Objectmessage</i>	Brings up an internal dialog panel with the options Yes, No and Cancel; with the title, Select an Option.
48	<b>static int showInternalConfirmDialog</b> <i>ComponentparentComponent, Objectmessage, Stringtitle, intoptionType</i>	Brings up a internal dialog panel where the number of choices is determined by the optionType parameter.
49	<b>static int showInternalConfirmDialog</b> <i>ComponentparentComponent, Objectmessage, Stringtitle, intoptionType, intmessageType</i>	Brings up an internal dialog panel where the number of choices is determined by the optionType parameter, where the messageType parameter determines the icon to display.
50	<b>static int showInternalConfirmDialog</b> <i>ComponentparentComponent, Objectmessage, Stringtitle, intoptionType, intmessageType, Iconicon</i>	Brings up an internal dialog panel with a specified icon, where the number of choices is determined by the optionType parameter.

- 51 **static String showInternalInputDialog***Component parentComponent, Object message*  
Shows an internal question-message dialog requesting input from the user parented to parentComponent.
- 52 **static String showInternalInputDialog***Component parentComponent, Object message, String title, int messageType*  
Shows an internal dialog requesting input from the user parented to parentComponent with the dialog having the title title and message type messageType.
- 53 **static Object showInternalInputDialog**  
*Component parentComponent, Object message, String title, int messageType, ImageIcon, Object[] selectionValues, Object initialValue*  
Prompts the user for input in a blocking internal dialog where the initial selection, possible selections, and all other options can be specified.
- 54 **static void showInternalMessageDialog***Component parentComponent, Object message*  
Brings up an internal confirmation dialog panel.
- 55 **static void showInternalMessageDialog***Component parentComponent, Object message, String title, int messageType*  
Brings up an internal dialog panel that displays a message using a default icon determined by the messageType parameter.
- 56 **static void showInternalMessageDialog***Component parentComponent, Object message, String title, int messageType, ImageIcon*  
Brings up an internal dialog panel displaying a message, specifying all parameters.
- 57 **static void showMessageDialog***Component parentComponent, Object message*  
Brings up an information-message dialog titled "Message".
- 58 **static void showMessageDialog***Component parentComponent, Object message, String title, int messageType*  
Brings up a dialog that displays a message using a default icon determined by the messageType parameter.
- 59 **static void showMessageDialog***Component parentComponent, Object message, String title, int messageType, ImageIcon*  
Brings up a dialog displaying a message, specifying all parameters.
- 60 **static int showOptionDialog**  
*Component parentComponent, Object message, String title, int optionType, int messageType, ImageIcon, Object[] options, Object initialValue*  
Brings up a dialog with a specified icon, where the initial choice is determined by the initialValue parameter and the number of choices is determined by the optionType parameter.
- 61 **void updateUI**  
Notification from the UIManager that the L&F has changed.
- 62 **static int showInternalOptionDialog**  
*Component parentComponent, Object message, String title, int optionType, int messageType, ImageIcon, Object[] options, Object initialValue*  
Brings up an internal dialog panel with a specified icon, where the initial choice is determined by the initialValue parameter and the number of choices is determined by the optionType parameter.

## Methods inherited

This class inherits methods from the following classes:

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

## JOptionPane 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.*;

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.showDialogDemo();
    }

    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 showDialogDemo(){
        headerLabel.setText("Control in action: JOptionPane");

        JButton okButton = new JButton("OK");
        JButton javaButton = new JButton("Yes/No");
        JButton cancelButton = new JButton("Yes/No/Cancel");

        okButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(
                    mainFrame, "Welcome to Tutorialspoint.com");
            }
        });

        javaButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int output = JOptionPane.showConfirmDialog(mainFrame
                    , "Click any button"
                    , "Tutorialspoint.com"
                    , JOptionPane.YES_NO_OPTION);

                if(output == JOptionPane.YES_OPTION){
                    statusLabel.setText("Yes selected.");
                }else if(output == JOptionPane.NO_OPTION){
                    statusLabel.setText("No selected.");
                }
            }
        });

        cancelButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int output = JOptionPane.showConfirmDialog(mainFrame
                    , "Click any button"
                    , "Tutorialspoint.com"
                    , JOptionPane.YES_NO_OPTION);
            }
        });
    }
}
```

