



JAVA SWING

java gui library

tutorialspoint
SIMPLY EASY LEARNING

www.tutorialspoint.com

 <https://www.facebook.com/tutorialspointindia>

 <https://twitter.com/tutorialspoint>

About the Tutorial

JAVA provides a rich set of libraries to create Graphical User Interface in a platform independent way. In this tutorial, we'll look at SWING GUI controls.

Audience

This tutorial is designed for software professionals who are willing to learn JAVA GUI Programming in simple and easy steps. This tutorial provides great understanding on JAVA GUI Programming concepts and after completing this tutorial you will be at an intermediate level of expertise, from where you can take yourself to higher levels of expertise.

Prerequisites

Before proceeding with this tutorial, you should have a basic understanding of Java programming language, text editor, execution of programs, etc.

Disclaimer & Copyright

© Copyright 2016 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com.

Table of Contents

About the Tutorial.....	i
Audience	i
Prerequisites	i
Disclaimer & Copyright.....	i
Table of Contents	ii
1. SWING – OVERVIEW	1
MVC Architecture.....	1
Swing Features.....	1
2. SWING – ENVIRONMENT SETUP	2
Setting Up the Path for Windows 2000/XP	2
Setting Up the Path for Windows 95/98/ME	2
Setting Up the Path for Linux, UNIX, Solaris, FreeBSD	2
Popular Java Editors	3
3. SWING – CONTROLS.....	4
Component Class.....	5
Container Class.....	27
JComponent Class	34
SWING UI Elements.....	48
JLabel Class	50
JButton Class	57
JColorChooser Class.....	62
JCheckBox Class.....	68
JRadioButton Class	74
JList Class	79

JComboBox Class.....	91
JTextField Class	102
JPasswordField Class	109
JTextArea Class.....	114
ImageIcon Class.....	120
JScrollBar Class.....	126
JOptionPane Class	133
JFileChooser Class	146
JProgressBar Class	158
JSlider Class	166
JSpinner Class.....	174
4. SWING – EVENT HANDLING.....	181
What is an Event?	181
Types of Event.....	181
What is Event Handling?	181
Steps Involved in Event Handling	182
Callback Methods.....	182
5. SWING – EVENT CLASSES.....	186
EventObject Class.....	186
SWING Event Classes.....	187
AWTEvent Class.....	188
ActionEvent Class	191
InputEvent Class.....	193
KeyEvent Class	195
MouseEvent Class	204
WindowEvent Class.....	207

AdjustmentEvent Class.....	210
ComponentEvent Class.....	211
ContainerEvent Class.....	213
MouseEvent Class.....	214
PaintEvent Class	215
6. SWING – EVENT LISTENERS	217
SWING Event Listener Interfaces.....	217
ActionListener Interface	218
ComponentListener Interface.....	221
ItemListener Interface.....	225
KeyListener Interface	229
MouseListener Interface	233
WindowListener Interface	237
AdjustmentListener Interface.....	241
ContainerListener Interface.....	244
MouseEventListener Interface.....	248
FocusListener Interface	252
7. SWING – EVENT ADAPTERS	256
SWING Adapters	256
FocusAdapter Class	256
KeyAdapter Class	261
MouseAdapter Class	264
MouseEventAdapter Class.....	269
WindowAdapter Class	273

8. SWING – LAYOUTS.....	278
Layout Manager	278
LayoutManager Interface	279
LayoutManager2 Interface	280
AWT Layout Manager Classes.....	281
BorderLayout Class.....	282
CardLayout Class	287
FlowLayout Class	293
GridLayout Class.....	298
GridBagLayout Class	303
GroupLayout Class.....	310
SpringLayout Class	317
9. SWING – MENU CLASSES.....	323
JMenuBar Class	324
JMenuItem Class	332
JMenu Class	342
JCheckboxMenuItem Class	353
JRadioButtonMenuItem Class	360
JPopupMenu Class	367
10. SWING – CONTAINERS.....	377
SWING Containers.....	377
JPanel Class	378
JFrame Class	382
JWindow Class	388

1. Swing – Overview

Swing API is a set of extensible GUI Components to ease the developer's life to create JAVA based Front End/GUI Applications. It is build on top of AWT API and acts as a replacement of AWT API, since it has almost every control corresponding to AWT controls. Swing component follows a Model-View-Controller architecture to fulfill the following criterias.

- A single API is to be sufficient to support multiple look and feel.
- API is to be model driven so that the highest level API is not required to have data.
- API is to use the Java Bean model so that Builder Tools and IDE can provide better services to the developers for use.

MVC Architecture

Swing API architecture follows loosely based MVC architecture in the following manner.

- Model represents component's data.
- View represents visual representation of the component's data.
- Controller takes the input from the user on the view and reflects the changes in Component's data.
- Swing component has Model as a seperate element, while the View and Controller part are clubbed in the User Interface elements. Because of which, Swing has a pluggable look-and-feel architecture.

Swing Features

- **Light Weight** - Swing components are independent of native Operating System's API as Swing API controls are rendered mostly using pure JAVA code instead of underlying operating system calls.
- **Rich Controls** - Swing provides a rich set of advanced controls like Tree, TabbedPane, slider, colorpicker, and table controls.
- **Highly Customizable** - Swing controls can be customized in a very easy way as visual apperance is independent of internal representation.
- **Pluggable look-and-feel** - SWING based GUI Application look and feel can be changed at run-time, based on available values.

2. Swing – Environment Setup

This section guides you on how to download and set up Java on your machine. Please use the following steps to set up the environment.

Java SE is freely available from the link [Download Java](#). Hence, you can download a version based on your operating system.

Follow the instructions to download Java and run the **.exe** to install Java on your machine. Once you have installed Java on your machine, you would need to set the environment variables to point to the correct installation directories.

Setting Up the Path for Windows 2000/XP

Assuming you have installed Java in **c:\Program Files\java\jdk** directory:

Step 1: Right-click on 'My Computer' and select 'Properties'.

Step 2: Click the 'Environment variables' button under the 'Advanced' tab.

Step 3: Alter the 'Path' variable so that it also contains the path to the Java executable. Example, if the path is currently set to '**C:\WINDOWS\SYSTEM32**', then change your path to read '**C:\WINDOWS\SYSTEM32;c:\Program Files\java\jdk\bin**'.

Setting Up the Path for Windows 95/98/ME

Assuming you have installed Java in **c:\Program Files\java\jdk** directory:

Step 1: Edit the '**C:\autoexec.bat**' file and add the following line at the end:
'SET PATH=%PATH%;C:\Program Files\java\jdk\bin'

Setting Up the Path for Linux, UNIX, Solaris, FreeBSD

Environment variable PATH should be set to point to where the Java binaries have been installed. Refer to your Shell documentation if you have trouble doing this.

Example, if you use **bash** as your shell, then you would add the following line to the end
'.bashrc: export PATH=/path/to/java:\$PATH'

Popular Java Editors

To write your Java programs, you will need a text editor. There are even more sophisticated IDE available in the market. But for now, you can consider one of the following:

- **Notepad:** On Windows machine, you can use any simple text editor like Notepad (Recommended for this tutorial), TextPad.
- **Netbeans:** Netbeans is a Java IDE that is open source and free, which can be downloaded from <http://www.netbeans.org/index.html>.
- **Eclipse:** Eclipse is also a Java IDE developed by the Eclipse open source community and can be downloaded from <http://www.eclipse.org/>

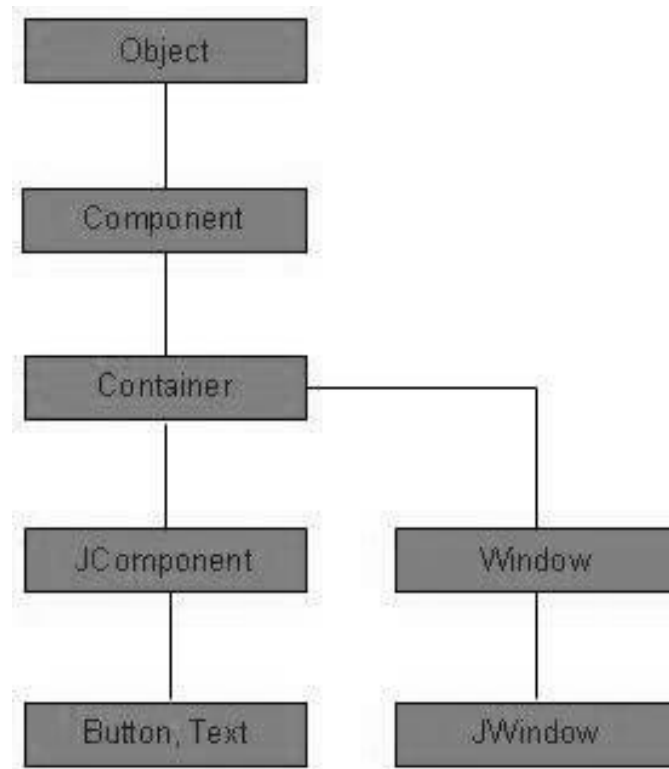
3. Swing – Controls

Every user interface considers the following three main aspects:

UI Elements: These are the core visual elements the user eventually sees and interacts with. GWT provides a huge list of widely used and common elements varying from basic to complex, which we will cover in this tutorial.

Layouts: They define how UI elements should be organized on the screen and provide a final look and feel to the GUI (Graphical User Interface). This part will be covered in the Layout chapter.

Behavior: These are the events which occur when the user interacts with UI elements. This part will be covered in the Event Handling chapter.



Every SWING controls inherits properties from the following Component class hierarchy.

Sr. No.	Class & Description
1	<u>Component</u> A Component is the abstract base class for the non menu user-interface controls of SWING. Component represents an object with graphical representation
2	<u>Container</u> A Container is a component that can contain other SWING components

3	<p>JComponent A JComponent is a base class for all SWING UI components. In order to use a SWING component that inherits from JComponent, the component must be in a containment hierarchy whose root is a top-level SWING container</p>
---	---

Component Class

Introduction

The class **Component** is the abstract base class for the non menu user-interface controls of AWT. Component represents an object with graphical representation.

Class Declaration

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

```
public abstract class Component
    extends Object
        implements ImageObserver, MenuContainer, Serializable
```

Field

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

- **static float BOTTOM_ALIGNMENT** - Ease-of-use constant for getAlignmentY.
- **static float CENTER_ALIGNMENT** - Ease-of-use constant for getAlignmentY and getAlignmentX.
- **static float LEFT_ALIGNMENT** - Ease-of-use constant for getAlignmentX.
- **static float RIGHT_ALIGNMENT** - Ease-of-use constant for getAlignmentX.
- **static float TOP_ALIGNMENT** - Ease-of-use constant for getAlignmentY().

Class Constructors

Sr.No.	Constructor & Description
1	<p>protected Component() This creates a new Component</p>

Class Methods

Sr.No.	Method & Description
1	boolean action(Event evt, Object what) Deprecated. As of JDK version 1.1, should register this component as ActionListener on the component which fires action events.
2	void add(PopupMenu popup) Adds the specified popup menu to the component.
3	void addComponentListener(ComponentListener l) Adds the specified component listener to receive the component events from this component.
4	void addFocusListener(FocusListener l) Adds the specified focus listener to receive focus events from this component, when this component gains input focus.
5	void addHierarchyBoundsListener(HierarchyBoundsListener l) Adds the specified hierarchy bounds listener to receive hierarchy bounds events from this component, when the hierarchy to which this container belongs changes.
6	void addHierarchyListener(HierarchyListener l) Adds the specified hierarchy listener to receive hierarchy changed events from this component, when the hierarchy to which this container belongs changes.
7	void addInputMethodListener(InputMethodListener l) Adds the specified input method listener to receive input method events from this component.
8	void addKeyListener(KeyListener l) Adds the specified key listener to receive key events from this component.
9	void addMouseListener(MouseListener l) Adds the specified mouse listener to receive mouse events from this component.

10	<p>void addMouseMotionListener(MouseMotionListener l)</p> <p>Adds the specified mouse motion listener to receive mouse motion events from this component.</p>
11	<p>void addMouseWheelListener(MouseWheelListener l)</p> <p>Adds the specified mouse wheel listener to receive mouse wheel events from this component.</p>
12	<p>void addNotify()</p> <p>Makes this Component displayable by connecting it to a native screen resource.</p>
13	<p>void addPropertyChangeListener (PropertyChangeListener listener)</p> <p>Adds a PropertyChangeListener to the listener list.</p>
14	<p>void addPropertyChangeListener (String propertyName, PropertyChangeListener listener)</p> <p>Adds a PropertyChangeListener to the listener list for a specific property.</p>
15	<p>void applyComponentOrientation(ComponentOrientation orientation)</p> <p>Sets the ComponentOrientation property of this component and all components contained within it.</p>
16	<p>boolean areFocusTraversalKeysSet(int id)</p> <p>Returns whether the set of focus traversal keys for the given focus traversal operation has been explicitly defined for this Component.</p>
17	<p>int checkImage(Image image, ImageObserver observer)</p> <p>Returns the status of the construction of a screen representation of the specified image.</p>
18	<p>int checkImage(Image image,int width,int height, ImageObserver observer)</p> <p>Returns the status of the construction of a screen representation of the specified image.</p>
19	<p>boolean contains(int x,int y)</p> <p>Checks whether this component "contains" the specified point, where x and y are defined to be relative to the coordinate system of this component.</p>

20	boolean contains(Point p) Checks whether this component "contains" the specified point, where the point's x and y coordinates are defined to be relative to the coordinate system of this component.
21	Image createImage(ImageProducer producer) Creates an image from the specified image producer.
22	Image createImage(int width,int height) Creates an off-screen drawable image to be used for double buffering.
23	VolatileImage createVolatileImage(int width,int height) Creates a volatile off-screen drawable image to be used for double buffering.
24	VolatileImage createVolatileImage(int width,int height, ImageCapabilities caps) Creates a volatile off-screen drawable image, with the given capabilities.
25	void deliverEvent(Event e) Deprecated. As of JDK version 1.1, replaced by <code>dispatchEvent(AWTEvent e)</code> .
26	void disable() Deprecated. As of JDK version 1.1, replaced by <code>setEnabled(boolean)</code> .
27	protected void disableEvents(long eventsToDisable) Disables the events defined by the specified event mask parameter from being delivered to this component.
28	void dispatchEvent(AWTEvent e) Dispatches an event to this component or one of its sub components.
29	void doLayout() Prompts the layout manager to lay out this component.
30	void enable() Deprecated. As of JDK version 1.1, replaced by <code>setEnabled(boolean)</code> .
31	void enable(boolean b)

	Deprecated. As of JDK version 1.1, replaced by <code>setEnabled(boolean)</code> .
32	protected void enableEvents(long eventsToEnable) Enables the events defined by the specified event mask parameter to be delivered to this component.
33	void enableInputMethods(boolean enable) Enables or disables input method support for this component.
34	protected void firePropertyChange(String propertyName, boolean oldValue, boolean newValue) Supports reporting bound property changes for boolean properties.
35	void firePropertyChange(String propertyName, byte oldValue, byte newValue) Reports a bound property change.
36	void firePropertyChange(String propertyName, char oldValue, char newValue) Reports a bound property change.
37	void firePropertyChange(String propertyName, double oldValue, double newValue) Reports a bound property change.
38	void firePropertyChange(String propertyName, float oldValue, float newValue) Reports a bound property change.
39	void firePropertyChange(String propertyName, long oldValue, long newValue) Reports a bound property change.
40	protected void firePropertyChange(String propertyName, Object oldValue, Object newValue) Supports reporting bound property changes for Object properties.
41	void firePropertyChange(String propertyName, short oldValue, short newValue)

	Reports a bound property change.
42	AccessibleContext getAccessibleContext() Gets the AccessibleContext associated with this Component.
43	float getAlignmentX() Returns the alignment along the x axis.
44	float getAlignmentY() Returns the alignment along the y axis.
45	Color getBackground() Gets the background color of this component.
46	int getBaseline(int width,int height) Returns the baseline.
47	Component.BaselineResizeBehavior getBaselineResizeBehavior() Returns an enum indicating how the baseline of the component changes as the size changes.
48	Rectangle getBounds() Gets the bounds of this component in the form of a Rectangle object.
49	Rectangle getBounds(Rectangle rv) Stores the bounds of this component into "return value" rv and returns rv.
50	ColorModel getColorModel() Gets the instance of ColorModel used to display the component on the output device.
51	Component getComponentAt(int x,int y) Determines if this component or one of its immediate subcomponents contains the (x , y) location, and if so, returns the containing component.
52	Component getComponentAt(Point p) Returns the component or subcomponent that contains the specified point.

53	ComponentListener[] getComponentListeners() Returns an array of all the component listeners registered on this component.
54	ComponentOrientation getComponentOrientation() Retrieves the language-sensitive orientation that is to be used to order the elements or the text within this component.
55	Cursor getCursor() Gets the cursor set in the component.
56	DropTarget getDropTarget() Gets the DropTarget associated with this Component.
57	Container getFocusCycleRootAncestor() Returns the Container which is the focus cycle root of this Component's focus traversal cycle.
58	FocusListener[] getFocusListeners() Returns an array of all the focus listeners registered on this component.
59	Set<AWTKeyStroke> getFocusTraversalKeys(int id) Returns the Set of focus traversal keys for a given traversal operation for this Component.
60	boolean getFocusTraversalKeysEnabled() Returns whether focus traversal keys are enabled for this Component.
61	Font getFont() Gets the font of this component.
62	FontMetrics getFontMetrics(Font font) Gets the font metrics for the specified font.
63	Color getForeground() Gets the foreground color of this component.
64	Graphics getGraphics()

	Creates a graphics context for this component.
65	GraphicsConfiguration getGraphicsConfiguration() Gets the GraphicsConfiguration associated with this Component.
66	int getHeight() Returns the current height of this component.
67	HierarchyBoundsListener[] getHierarchyBoundsListeners() Returns an array of all the hierarchy bounds listeners registered on this component.
68	HierarchyListener[] getHierarchyListeners() Returns an array of all the hierarchy listeners registered on this component.
69	boolean getIgnoreRepaint()
70	InputContext getInputContext() Gets the input context used by this component for handling the communication with input methods, when the text is entered in this component.
71	InputMethodListener[] getInputMethodListeners() Returns an array of all the input method listeners registered on this component.
72	InputMethodRequests getInputMethodRequests() Gets the input method request handler which supports requests from input methods for this component.
73	KeyListener[] getKeyListeners() Returns an array of all the key listeners registered on this component.
74	<T extends EventListener> T[] getListeners(Class<T> listenerType) Returns an array of all the objects currently registered as FooListeners upon this Component.
75	Locale getLocale()

	Gets the locale of this component.
76	Point getLocation() Gets the location of this component in the form of a point specifying the component's top-left corner.
77	Point getLocation(Point rv) Stores the x,y origin of this component into "return value" rv and returns rv.
78	Point getLocationOnScreen() Gets the location of this component in the form of a point specifying the component's top-left corner in the screen's coordinate space.
79	Dimension getMaximumSize() Gets the maximum size of this component.
80	Dimension getMinimumSize() Gets the minimum size of this component.
81	MouseListener[] getMouseListeners() Returns an array of all the mouse listeners registered on this component.
82	MouseMotionListener[] getMouseMotionListeners() Returns an array of all the mouse motion listeners registered on this component.
83	Point getMousePosition() Returns the position of the mouse pointer in this Component's coordinate space, if the Component is directly under the mouse pointer, otherwise returns null.
84	MouseWheelListener[] getMouseWheelListeners() Returns an array of all the mouse wheel listeners registered on this component.
85	String getName() Gets the name of the component.
86	Container getParent()

	Gets the parent of this component.
87	java.awt.peer.ComponentPeer getPeer() Deprecated. As of JDK version 1.1, programs should not directly manipulate peers; replaced by boolean isDisplayable().
88	Dimension getPreferredSize() Gets the preferred size of this component.
89	PropertyChangeListener[] getPropertyChangeListeners() Returns an array of all the property change listeners registered on this component.
90	PropertyChangeListener[] getPropertyChangeListeners(String propertyName) Returns an array of all the listeners which have been associated with the named property.
91	Dimension getSize() Returns the size of this component in the form of a Dimension object.
92	Dimension getSize(Dimension rv) Stores the width/height of this component into "return value" rv and returns rv.
93	Toolkit getToolkit() Gets the toolkit of this component.
94	Object getTreeLock() Gets this component's locking object (the object that owns the thread synchronization monitor) for AWT component-tree and layout operations.
95	int getWidth() Returns the current width of this component.
96	int getX() Returns the current x coordinate of the components origin.

97	int getY() Returns the current y coordinate of the components origin.
98	boolean gotFocus(Event evt, Object what) Deprecated. As of JDK version 1.1, replaced by processFocusEvent(FocusEvent).
99	boolean handleEvent(Event evt) Deprecated. As of JDK version 1.1 replaced by processEvent(AWTEvent).
100	boolean hasFocus() Returns true if this Component is the focus owner.
101	void hide() Deprecated. As of JDK version 1.1, replaced by setVisible(boolean).
102	boolean imageUpdate(Image img,int infoflags,int x,int y,int w,int h) Repaints the component when the image has changed.
103	boolean inside(int x,int y) Deprecated. As of JDK version 1.1, replaced by contains(int, int).
104	void invalidate() Invalidates this component.
105	boolean isBackgroundSet() Returns whether the background color has been explicitly set for this Component.
106	boolean isCursorSet() Returns whether the cursor has been explicitly set for this Component.
107	boolean isDisplayable() Determines whether this component is displayable.

108	<p>boolean isDoubleBuffered()</p> <p>Returns true if this component is painted to an offscreen image (buffer) that's copied to the screen later.</p>
109	<p>boolean isEnabled()</p> <p>Determines whether this component is enabled.</p>
110	<p>boolean isFocusable()</p> <p>Returns whether this Component can be focused.</p>
111	<p>boolean isFocusCycleRoot(Container container)</p> <p>Returns whether the specified Container is the focus cycle root of this Component's focus traversal cycle.</p>
112	<p>boolean isFocusOwner()</p> <p>Returns true if this Component is the focus owner.</p>
113	<p>boolean isFocusTraversable()</p> <p>Deprecated. As of 1.4, replaced by isFocusable().</p>
114	<p>boolean isFontSet()</p> <p>Returns whether the font has been explicitly set for this Component.</p>
115	<p>boolean isForegroundSet()</p> <p>Returns whether the foreground color has been explicitly set for this Component.</p>
116	<p>boolean isLightweight()</p> <p>A lightweight component doesn't have a native toolkit peer.</p>
117	<p>boolean isMaximumSizeSet()</p> <p>Returns true if the maximum size has been set to a non-null value otherwise returns false.</p>

118	boolean isMinimumSizeSet() Returns whether or not setMinimumSize has been invoked with a non-null value.
119	boolean isOpaque() Returns true if this component is completely opaque, returns false by default.
120	boolean isPreferredSizeSet() Returns true if the preferred size has been set to a non-null value otherwise returns false.
121	boolean isShowing() Determines whether this component is showing on screen.
122	boolean isValid() Determines whether this component is valid.
123	boolean isVisible() Determines whether this component should be visible when its parent is visible.
124	boolean keyDown(Event evt,int key) Deprecated. As of JDK version 1.1, replaced by processKeyEvent(KeyEvent).
125	boolean keyUp(Event evt,int key) Deprecated. As of JDK version 1.1, replaced by processKeyEvent(KeyEvent).
126	void layout() Deprecated. As of JDK version 1.1, replaced by doLayout().
127	void list() Prints a listing of this component to the standard system output stream System.out.
128	void list(PrintStream out) Prints a listing of this component to the specified output stream.
129	void list(PrintStream out,int indent)

	Prints out a list, starting at the specified indentation, to the specified print stream.
130	void list(PrintWriter out) Prints a listing to the specified print writer.
131	void list(PrintWriter out,int indent) Prints out a list, starting at the specified indentation, to the specified print writer.
132	Component locate(int x,int y) Deprecated. As of JDK version 1.1, replaced by <code>getComponentAt(int, int)</code> .
133	Point location() Deprecated. As of JDK version 1.1, replaced by <code>getLocation()</code> .
134	boolean lostFocus(Event evt, Object what) Deprecated. As of JDK version 1.1, replaced by <code>processFocusEvent(FocusEvent)</code> .
135	boolean mouseDown(Event evt,int x,int y) Deprecated. As of JDK version 1.1, replaced by <code>processMouseEvent(MouseEvent)</code> .
136	boolean mouseDrag(Event evt,int x,int y) Deprecated. As of JDK version 1.1, replaced by <code>processMouseMotionEvent(MouseEvent)</code> .
137	boolean mouseEnter(Event evt,int x,int y) Deprecated. As of JDK version 1.1, replaced by <code>processMouseEvent(MouseEvent)</code> .

138	boolean mouseExit(Event evt,int x,int y) Deprecated. As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).
139	boolean mouseMove(Event evt,int x,int y) Deprecated. As of JDK version 1.1, replaced by processMouseMotionEvent(MouseEvent).
140	boolean mouseUp(Event evt,int x,int y) Deprecated. As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).
141	void move(int x,int y) Deprecated. As of JDK version 1.1, replaced by setLocation(int, int).
142	void nextFocus() Deprecated. As of JDK version 1.1, replaced by transferFocus().
143	void paint(Graphics g) Paints this component.
144	void paintAll(Graphics g) Paints this component and all of its subcomponents.
145	boolean postEvent(Event e) Deprecated. As of JDK version 1.1, replaced by dispatchEvent(AWTEvent).
146	boolean prepareImage(Image image,int width,int height, ImageObserver observer) Prepares an image for rendering on this component at the specified width and height.
147	void print(Graphics g) Prints this component.

148	void printAll(Graphics g) Prints this component and all of its subcomponents.
149	protected void processComponentEvent(ComponentEvent e) Processes component events occurring on this component by dispatching them to any registered ComponentListener objects.
150	protected void processEvent(AWTEvent e) Processes events occurring on this component.
151	protected void processFocusEvent(FocusEvent e) Processes focus events occurring on this component by dispatching them to any registered FocusListener objects.
152	protected void processHierarchyBoundsEvent(HierarchyEvent e) Processes hierarchy bounds events occurring on this component by dispatching them to any registered HierarchyBoundsListener objects.
153	protected void processHierarchyEvent(HierarchyEvent e) Processes hierarchy events occurring on this component by dispatching them to any registered HierarchyListener objects.
154	protected void processInputMethodEvent(InputMethodEvent e) Processes input method events occurring on this component by dispatching them to any registered InputMethodListener objects.
155	protected void processKeyEvent(KeyEvent e) Processes key events occurring on this component by dispatching them to any registered KeyListener objects.
156	protected void processMouseEvent(MouseEvent e) Processes mouse events occurring on this component by dispatching them to any registered MouseListener objects.

157	<p>protected void processMouseEvent(MouseEvent e)</p> <p>Processes mouse motion events occurring on this component by dispatching them to any registered MouseMotionListener objects.</p>
158	<p>protected void processMouseWheelEvent(MouseWheelEvent e)</p> <p>Processes mouse wheel events occurring on this component by dispatching them to any registered MouseWheelListener objects.</p>
159	<p>void remove(MenuComponent popup)</p> <p>Removes the specified popup menu from the component.</p>
160	<p>void removeComponentListener(ComponentListener l)</p> <p>Removes the specified component listener so that it no longer receives component events from this component.</p>
161	<p>void removeFocusListener(FocusListener l)</p> <p>Removes the specified focus listener so that it no longer receives focus events from this component.</p>
162	<p>void removeHierarchyBoundsListener(HierarchyBoundsListener l)</p> <p>Removes the specified hierarchy bounds listener so that it no longer receives hierarchy bounds events from this component.</p>
163	<p>void removeHierarchyListener(HierarchyListener l)</p> <p>Removes the specified hierarchy listener so that it no longer receives hierarchy changed events from this component.</p>
164	<p>void removeInputMethodListener(InputMethodListener l)</p> <p>Removes the specified input method listener so that it no longer receives input method events from this component.</p>
165	<p>void removeKeyListener(KeyListener l)</p> <p>Removes the specified key listener so that it no longer receives key events from this component.</p>

166	<p>void removeMouseListener(MouseListener l)</p> <p>Removes the specified mouse listener so that it no longer receives mouse events from this component.</p>
167	<p>void removeMouseMotionListener(MouseMotionListener l)</p> <p>Removes the specified mouse motion listener so that it no longer receives mouse motion events from this component.</p>
168	<p>void removeMouseWheelListener(MouseWheelListener l)</p> <p>Removes the specified mouse wheel listener so that it no longer receives mouse wheel events from this component.</p>
169	<p>void removeNotify()</p> <p>Makes this component undisplayable by destroying its native screen resource.</p>
170	<p>void removePropertyChangeListener(PropertyChangeListener listener)</p> <p>Removes a PropertyChangeListener from the listener list.</p>
171	<p>void removePropertyChangeListener(String propertyName, PropertyChangeListener listener)</p> <p>Removes a PropertyChangeListener from the listener list for a specific property.</p>
172	<p>void repaint()</p> <p>Repaints this component.</p>
173	<p>void repaint(int x,int y,int width,int height)</p> <p>Repaints the specified rectangle of this component.</p>
174	<p>void repaint(long tm)</p> <p>Repaints the component.</p>
175	<p>void repaint(long tm,int x,int y,int width,int height)</p> <p>Repaints the specified rectangle of this component within tm milliseconds.</p>

176	<p>void requestFocus()</p> <p>Requests that this component get the input focus, and that this component's top-level ancestor become the focused Window.</p>
177	<p>protected boolean requestFocus(boolean temporary)</p> <p>Requests that this component get the input focus, and that this component's top-level ancestor become the focused Window.</p>
178	<p>boolean requestFocusInWindow()</p> <p>Requests that this component get the input focus, if this component's top-level ancestor is already the focused Window.</p>
179	<p>protected boolean requestFocusInWindow(boolean temporary)</p> <p>Requests that this component get the input focus, if this component's top-level ancestor is already the focused Window.</p>
180	<p>void reshape(int x,int y,int width,int height)</p> <p>Deprecated. As of JDK version 1.1, replaced by setBounds(int, int, int, int).</p>
181	<p>void resize(Dimension d)</p> <p>Deprecated. As of JDK version 1.1, replaced by setSize(Dimension).</p>
182	<p>void resize(int width,int height)</p> <p>Deprecated. As of JDK version 1.1, replaced by setSize(int, int).</p>
183	<p>void setBackground(Color c)</p> <p>Sets the background color of this component.</p>
184	<p>void setBounds(int x,int y,int width,int height)</p> <p>Moves and resizes this component.</p>
185	<p>void setBounds(Rectangle r)</p> <p>Moves and resizes this component to conform to the new bounding rectangle r.</p>

186	<p>void setComponentOrientation(ComponentOrientation o)</p> <p>Sets the language-sensitive orientation that is to be used to order the elements or text within this component.</p>
187	<p>void setCursor(Cursor cursor)</p> <p>Sets the cursor image to the specified cursor.</p>
188	<p>void setDropTarget(DropTarget dt)</p> <p>Associates a DropTarget with this component.</p>
189	<p>void setEnabled(boolean b)</p> <p>Enables or disables this component, depending on the value of the parameter b.</p>
190	<p>void setFocusable(boolean focusable)</p> <p>Sets the focusable state of this Component to the specified value.</p>
191	<p>void setFocusTraversalKeys(int id, Set<? extends AWTKeyStroke> keystrokes)</p> <p>Sets the focus traversal keys for a given traversal operation for this Component.</p>
192	<p>void setFocusTraversalKeysEnabled(boolean focusTraversalKeysEnabled)</p> <p>Sets whether focus traversal keys are enabled for this component.</p>
193	<p>void setFont(Font f)</p> <p>Sets the font of this component.</p>
194	<p>void setForeground(Color c)</p> <p>Sets the foreground color of this component.</p>
195	<p>void setIgnoreRepaint(boolean ignoreRepaint)</p> <p>Sets whether or not paint messages received from the operating system should be ignored.</p>

196	void setLocale(Locale l) Sets the locale of this component.
197	void setLocation(int x,int y) Moves this component to a new location.
198	void setLocation(Point p) Moves this component to a new location.
199	void setMaximumSize(Dimension maximumSize) Sets the maximum size of this component to a constant value.
200	void setMinimumSize(Dimension minimumSize) Sets the minimum size of this component to a constant value.
201	void setName(String name) Sets the name of the component to the specified string.
202	void setPreferredSize(Dimension preferredSize) Sets the preferred size of this component to a constant value.
203	void setSize(Dimension d) Resizes this component so that it has width d.width and height d.height .
204	void setSize(int width,int height) Resizes this component so that it has width width and height height .
205	void setVisible(boolean b) Shows or hides this component depending on the value of parameter b .
206	void show() Deprecated. As of JDK version 1.1, replaced by setVisible(boolean).
207	void show(boolean b) Deprecated. As of JDK version 1.1, replaced by setVisible(boolean).

208	Dimension size() Deprecated. As of JDK version 1.1, replaced by getSize().
209	String toString() Returns a string representation of this component and its values.
210	void transferFocus() Transfers the focus to the next component, as though this Component were the focus owner.
211	void transferFocusBackward() Transfers the focus to the previous component, as though this Component were the focus owner.
212	void transferFocusUpCycle() Transfers the focus up one focus traversal cycle.
213	void update(Graphics g) Updates this component.
214	void validate() Ensures that this component has a valid layout.
215	Rectangle bounds() Deprecated. As of JDK version 1.1, replaced by getBounds().
216	protected AWTEvent coalesceEvents(AWTEvent existingEvent, AWTEvent newEvent) Potentially coalesce an event being posted with an existing event.
217	protected String paramString() Returns a string representing the state of this component.
218	protected void firePropertyChange(String propertyName,int oldValue,int newValue) Supports reporting bound property changes for integer properties.
219	Dimension preferredSize()

	Deprecated. As of JDK version 1.1, replaced by <code>getPreferredSize()</code> .
220	<code>boolean prepareImage(Image image, ImageObserver observer)</code> Prepares an image for rendering on this component.
221	<code>Dimension minimumSize()</code> Deprecated. As of JDK version 1.1, replaced by <code>getMinimumSize()</code> .

End of ebook preview

If you liked what you saw...

Buy it from our store @ <https://store.tutorialspoint.com>