SWING - 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

S.N.	Constructor & Description
1	**protected Component** This creates a new Component.

Class methods

S.N.	Method & Description
1	**boolean action***Eventevt, Objectwhat* Deprecated. As of JDK version 1.1, should register this component as ActionListener on component which fires action events.
2	**void add***PopupMenupopup* Adds the specified popup menu to the component.
3	**void addComponentListener***ComponentListenerl* Adds the specified component listener to receive component events from this component.

4 **void addFocusListener*FocusListenerl***

Adds the specified focus listener to receive focus events from this component when this component gains input focus.

5 **void addHierarchyBoundsListener*HierarchyBoundsListenerl***

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*HierarchyListenerl***

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*InputMethodListenerl***

Adds the specified input method listener to receive input method events from this component.

8 **void addKeyListener*KeyListenerl***

Adds the specified key listener to receive key events from this component.

9 **void addMouseListener*MouseListenerl***

Adds the specified mouse listener to receive mouse events from this component.

10 **void addMouseMotionListener*MouseMotionListenerl***

Adds the specified mouse motion listener to receive mouse motion events from this component.

11 **void addMouseWheelListener*MouseWheelListenerl***

Adds the specified mouse wheel listener to receive mouse wheel events from this component.

12 **void addNotify**

Makes this Component displayable by connecting it to a native screen resource.

13 **void addPropertyChangeListener*PropertyChangeListenerlistener***

Adds a PropertyChangeListener to the listener list.

14 **void addPropertyChangeListener*StringpropertyName, PropertyChangeListenerlistener***

Adds a PropertyChangeListener to the listener list for a specific property.

15 **void applyComponentOrientation*ComponentOrientationorientation***

Sets the ComponentOrientation property of this component and all components contained within it.

16 **boolean areFocusTraversalKeysSet*intid***

Returns whether the Set of focus traversal keys for the given focus traversal operation has been explicitly defined for this Component.

17 **int checkImage***Imageimage, ImageObserverobserver*

Returns the status of the construction of a screen representation of the specified image.

18 **int checkImage***Imageimage, intwidth, intheight, ImageObserverobserver*

Returns the status of the construction of a screen representation of the specified image.

19 **boolean contains***intx, inty*

Checks whether this component "contains" the specified point, where x and y are defined to be relative to the coordinate system of this component.

20 **boolean contains***Pointp*

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***ImageProducerproducer*

Creates an image from the specified image producer.

22 **Image createImage***intwidth, intheight*

Creates an off-screen drawable image to be used for double buffering.

23 **VolatileImage createVolatileImage***intwidth, intheight*

Creates a volatile off-screen drawable image to be used for double buffering.

24 **VolatileImage createVolatileImage***intwidth, intheight, ImageCapabilitiescaps*

Creates a volatile off-screen drawable image, with the given capabilities.

25 **void deliverEvent***Evente*

Deprecated. As of JDK version 1.1, replaced by dispatchEvent*AWTEvente*.

26 **void disable**

Deprecated. As of JDK version 1.1, replaced by setEnabled*boolean*.

27 **protected void disableEvents***longeventsToDisable*

Disables the events defined by the specified event mask parameter from being delivered to this component.

28 **void dispatchEvent***AWTEvente*

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 setEnabled*boolean*.

31 **void enable*booleanb***

Deprecated. As of JDK version 1.1, replaced by setEnabled*boolean*.

32 **protected void enableEvents*longeventsToEnable***

Enables the events defined by the specified event mask parameter to be delivered to this component.

33 **void enableInputMethods*booleanenable***

Enables or disables input method support for this component.

34 **protected void firePropertyChange*StringpropertyName, booleanoldValue, booleannewValue***

Support for reporting bound property changes for boolean properties.

35 **void firePropertyChange*StringpropertyName, byteoldValue, bytenewValue***

Reports a bound property change.

36 **void firePropertyChange*StringpropertyName, charoldValue, charnewValue***

Reports a bound property change.

37 **void firePropertyChange*StringpropertyName, doubleoldValue, doublenewValue***

Reports a bound property change.

38 **void firePropertyChange*StringpropertyName, floatoldValue, floatnewValue***

Reports a bound property change.

39 **void firePropertyChange*StringpropertyName, longoldValue, longnewValue***

Reports a bound property change.

40 **protected void firePropertyChange*StringpropertyName, ObjectoldValue, ObjectnewValue***

Support for reporting bound property changes for Object properties.

41 **void firePropertyChange*StringpropertyName, shortoldValue, shortnewValue***

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***intwidth, intheight*

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***Rectanglerv*

Stores the bounds of this component into **return value** rv and return rv.

50 **ColorModel getColorModel**

Gets the instance of ColorModel used to display the component on the output device.

51 **Component getComponentAt***intx, inty*

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***Pointp*

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 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***intid*

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***Fontfont*

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 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***Pointrv*

Stores the x,y origin of this component into **return value** rv and return 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 mininimum 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***StringpropertyName*

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***Dimensionrv***Stores the width/height of this component into return value rv and return rv.**

93 **Toolkit getToolkit**

Gets the toolkit of this component.

94 **Object getTreeLock**

Gets this component's locking object *theobjectthatownsthethreadsychronizationmonitor*

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***Eventevt, Objectwhat*

Deprecated. As of JDK version 1.1, replaced by processFocusEvent*FocusEvent*

99 **boolean handleEvent***Eventevt*

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***Imageimg, intinfoflags, intx, inty, intw, inth*

Repaints the component when the image has changed.

103 **boolean inside***intx, inty*

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 **boolean isDoubleBuffered**

Returns true if this component is painted to an offscreen image (**buffer**)

that's copied to the screen later.

109 **boolean isEnabled**

Determines whether this component is enabled.

110 **boolean isFocusable**

Returns whether this Component can be focused.

111 **boolean isFocusCycleRoot***Containercontainer*

Returns whether the specified Container is the focus cycle root of this Component's focus traversal cycle.

112 **boolean isFocusOwner**

Returns true if this Component is the focus owner.

113 **boolean isFocusTraversable**

Deprecated. As of 1.4, replaced by isFocusable.

114 **boolean isFontSet**

Returns whether the font has been explicitly set for this Component.

115 **boolean isForegroundSet**

Returns whether the foreground color has been explicitly set for this Component.

116 **boolean isLightweight**

A lightweight component doesn't have a native toolkit peer.

117 **boolean isMaximumSizeSet**

Returns true if the maximum size has been set to a non-null value otherwise returns false.

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***Eventevt, intkey*

Deprecated. As of JDK version 1.1, replaced by processKeyEvent*KeyEvent*.

125 **boolean keyUp***Eventevt, intkey*

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*PrintStreamout***

Prints a listing of this component to the specified output stream.

129 **void list*PrintStreamout, intindent***

Prints out a list, starting at the specified indentation, to the specified print stream.

130 **void list*PrintWriterout***

Prints a listing to the specified print writer.

131 **void list*PrintWriterout, intindent***

Prints out a list, starting at the specified indentation, to the specified print writer.

132 **Component locate*intx, inty***

Deprecated. As of JDK version 1.1, replaced by getComponentAt*int, int*.

133 **Point location**

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

134 **boolean lostFocus*Eventevt, Objectwhat***

Deprecated. As of JDK version 1.1, replaced by processFocusEvent*FocusEvent*.

135 **boolean mouseDown*Eventevt, intx, inty***

Deprecated. As of JDK version 1.1, replaced by processMouseEvent*MouseEvent*.

136 **boolean mouseDrag*Eventevt, intx, inty***

Deprecated. As of JDK version 1.1, replaced by processMouseMotionEvent*MouseEvent*.

137 **boolean mouseEnter*Eventevt, intx, inty***

Deprecated. As of JDK version 1.1, replaced by processMouseEvent*MouseEvent*.

138 **boolean mouseExit*Eventevt, intx, inty***

Deprecated. As of JDK version 1.1, replaced by processMouseEvent*MouseEvent*..

139 **boolean mouseMove*Eventevt, intx, inty***

Deprecated. As of JDK version 1.1, replaced by processMouseMotionEvent*MouseEvent*..

140 **boolean mouseUp***Eventevt, intx, inty*

Deprecated. As of JDK version 1.1, replaced by processMouseEvent*MouseEvent*.

141 **void move***intx, inty*

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***Graphicsg*

Paints this component.

144 **void paintAll***Graphicsg*

Paints this component and all of its subcomponents.

145 **boolean postEvent***Evente*

Deprecated. As of JDK version 1.1, replaced by dispatchEvent*AWTEvent*.

146 **boolean prepareImage***Imageimage, intwidth, intheight, ImageObserverobserver*

Prepares an image for rendering on this component at the specified width and height.

147 **void print***Graphicsg*

Prints this component.

148 **void printAll***Graphicsg*

Prints this component and all of its subcomponents.

149 **protectedvoid processComponentEvent***ComponentEvente*

Processes component events occurring on this component by dispatching them to any registered ComponentListener objects.

150 **protected void processEvent***AWTEvente*

Processes events occurring on this component.

151 **protected void processFocusEvent***FocusEvente*

Processes focus events occurring on this component by dispatching them to any registered FocusListener objects.

152 **protected void processHierarchyBoundsEvent***HierarchyEvente*

Processes hierarchy bounds events occurring on this component by dispatching them to any registered HierarchyBoundsListener objects.

153 **protected void processHierarchyEvent***HierarchyEvente*

Processes hierarchy events occurring on this component by dispatching them to any registered HierarchyListener objects.

154 **protectedvoid processInputMethodEvent***InputMethodEvente*

Processes input method events occurring on this component by dispatching them to any registered InputMethodListener objects.

155 **protected void processKeyEvent***KeyEvente*

Processes key events occurring on this component by dispatching them to any registered KeyListener objects.

156 **protected void processMouseEvent***MouseEvente*

Processes mouse events occurring on this component by dispatching them to any registered MouseListener objects.

157 **protected void processMouseMotionEvent***MouseEvente*

Processes mouse motion events occurring on this component by dispatching them to any registered MouseMotionListener objects.

158 **protected void processMouseWheelEvent***MouseWheelEvente*

Processes mouse wheel events occurring on this component by dispatching them to any registered MouseWheelListener objects.

159 **void remove***MenuComponentpopup*

Removes the specified popup menu from the component.

160 **void removeComponentListener***ComponentListenerl*

Removes the specified component listener so that it no longer receives component events from this component.

161 **void removeFocusListener***FocusListenerl*

Removes the specified focus listener so that it no longer receives focus events from this component.

162 **void removeHierarchyBoundsListener***HierarchyBoundsListenerl*

Removes the specified hierarchy bounds listener so that it no longer receives hierarchy bounds events from this component.

163 **void removeHierarchyListener***HierarchyListenerl*

Removes the specified hierarchy listener so that it no longer receives hierarchy changed events from this component.

164 **void removeInputMethodListener***InputMethodListenerl*

Removes the specified input method listener so that it no longer receives input method events from this component.

165 **void removeKeyListener***KeyListenerl*

Removes the specified key listener so that it no longer receives key events from this component.

166 **void removeMouseListener***MouseListenerl*

Removes the specified mouse listener so that it no longer receives mouse events from this component.

167 **void removeMouseMotionListener***MouseMotionListenerl*

Removes the specified mouse motion listener so that it no longer receives mouse motion events from this component.

168 **void removeMouseWheelListener***MouseWheelListenerl*

Removes the specified mouse wheel listener so that it no longer receives mouse wheel events from this component.

169 **void removeNotify**

Makes this Component undisplayable by destroying it native screen resource.

170 **void removePropertyChangeListener***PropertyChangeListenerlistener*

Removes a PropertyChangeListener from the listener list.

171 **void removePropertyChangeListener***StringpropertyName, PropertyChangeListenerlistener*

Removes a PropertyChangeListener from the listener list for a specific property.

172 **void repaint**

Repaints this component.

173 **void repaint***intx, inty, intwidth, intheight*

Repaints the specified rectangle of this component.

174 **void repaint***longtm*

Repaints the component.

175 **void repaint***longtm, intx, inty, intwidth, intheight*

Repaints the specified rectangle of this component within tm milliseconds.

176 **void requestFocus**

Requests that this Component get the input focus, and that this Component's top-level ancestor become the focused Window.

177 **protected boolean requestFocus***booleantemporary*

Requests that this Component get the input focus, and that this Component's top-level

ancestor become the focused Window.

178 **boolean requestFocusInWindow**

Requests that this Component get the input focus, if this Component's top-level ancestor is already the focused Window.

179 **protected boolean requestFocusInWindow***booleantemporary*

Requests that this Component get the input focus, if this Component's top-level ancestor is already the focused Window.

180 **void reshape***intx, inty, intwidth, intheight*

Deprecated. As of JDK version 1.1, replaced by setBounds*int*, *int*, *int*, *int*.

181 **void resize***Dimensiond*

Deprecated. As of JDK version 1.1, replaced by setSize*Dimension*.

182 **void resize***intwidth, intheight*

Deprecated. As of JDK version 1.1, replaced by setSize*int*, *int*.

183 **void setBackground***Colorc*

Sets the background color of this component.

184 **void setBounds***intx, inty, intwidth, intheight*

Moves and resizes this component.

185 **void setBounds***Rectangler*

Moves and resizes this component to conform to the new bounding rectangle r.

186 **void setComponentOrientation***ComponentOrientationo*

Sets the language-sensitive orientation that is to be used to order the elements or text within this component.

187 **void setCursor***Cursorcursor*

Sets the cursor image to the specified cursor.

188 **void setDropTarget***DropTargetdt*

Associate a DropTarget with this component.

189 **void setEnabled***booleanb*

Enables or disables this component, depending on the value of the parameter b.

190 **void setFocusable***booleanfocusable*

Sets the focusable state of this Component to the specified value.

191 **void setFocusTraversalKeys***intid, Set < ?extendsAWTKeyStroke > keystrokes*

Sets the focus traversal keys for a given traversal operation for this Component.

192 **void setFocusTraversalKeysEnabled***booleanfocusTraversalKeysEnabled*

Sets whether focus traversal keys are enabled for this Component.

193 **void setFont***Fontf*

Sets the font of this component.

194 **void setForeground***Colorc*

Sets the foreground color of this component.

195 **void setIgnoreRepaint***booleanignoreRepaint*

Sets whether or not paint messages received from the operating system should be ignored.

196 **void setLocale***Localel*

Sets the locale of this component.

197 **void setLocation***intx, inty*

Moves this component to a new location.

198 **void setLocation***Pointp*

Moves this component to a new location.

199 **void setMaximumSize***DimensionmaximumSize*

Sets the maximum size of this component to a constant value.

200 **void setMinimumSize***DimensionminimumSize*

Sets the minimum size of this component to a constant value.

201 **void setName***Stringname*

Sets the name of the component to the specified string.

202 **void setPreferredSize***DimensionpreferredSize*

Sets the preferred size of this component to a constant value.

203 **void setSize***Dimensiond*

Resizes this component so that it has width d.width and height d.height.

204 **void setSize***intwidth, intheight*

Resizes this component so that it has width width and height height.

205 **void setVisible***booleanb*

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***booleanb*

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***Graphicsg*

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***AWTEventexistingEvent, AWTEventnewEvent*

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***StringpropertyName, intoldValue, intnewValue*

Support for reporting bound property changes for integer properties.

219 **Dimension preferredSize**

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

220 **boolean prepareImage***Imageimage, ImageObserverobserver*

Prepares an image for rendering on this component.

221 **Dimension minimumSize**

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

Methods inherited

This class inherits methods from the following classes:

- java.lang.Object