

SWING - CONTAINER CLASS

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

Copyright © tutorialspoint.com

Introduction

The class **Container** is the super class for the containers of AWT. Container object can contain other AWT components.

Class declaration

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

```
public class Container
    extends Component
```

Class constructors

S.N.	Constructor & Description
1	Container This creates a new Container.

Class methods

S.N.	Method & Description
1	Component addComponentcomp Appends the specified component to the end of this container.
2	Component addComponentcomp, intindex Adds the specified component to this container at the given position.
3	void addComponentcomp, Objectconstraints Adds the specified component to the end of this container.
4	void addComponentcomp, Objectconstraints, intindex Adds the specified component to this container with the specified constraints at the specified index.
5	Component addStringname, Componentcomp Adds the specified component to this container.
6	void addContainerListenerContainerListenerl Adds the specified container listener to receive container events from this container.
7	protected void addImplComponentcomp, Objectconstraints, intindex

Adds the specified component to this container at the specified index.

8 **void addNotify**

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

9 **void addPropertyChangeListener***PropertyChangeListenerlistener*

Adds a PropertyChangeListener to the listener list.

10 **void addPropertyChangeListener***StringpropertyName, PropertyChangeListenerlistener*

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

11 **void applyComponentOrientation***ComponentOrientationo*

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

12 **boolean areFocusTraversalKeysSet***intid*

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

13 **int countComponents**

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

14 **void deliverEvent***Evente*

Deprecated. As of JDK version 1.1, replaced by dispatchEventAWT*Evente*

15 **void doLayout**

Causes this container to lay out its components.

16 **Component findComponentAt***intx, inty*

Locates the visible child component that contains the specified position.

17 **Component findComponentAtPoint***p*

Locates the visible child component that contains the specified point.

18 **float getAlignmentX**

Returns the alignment along the x axis.

19 **float getAlignmentY**

Returns the alignment along the y axis.

20 **Component getComponent***intn*

Gets the nth component in this container.

21	Component getComponentAt <i>intx, inty</i>
	Locates the component that contains the x,y position.
22	Component getComponentAtPoint <i>p</i>
	Gets the component that contains the specified point.
23	int getComponentCount
	Gets the number of components in this panel.
24	Component[] getComponents
	Gets all the components in this container.
25	int getComponentZOrder <i>Componentcomp</i>
	Returns the z-order index of the component inside the container.
26	ContainerListener[] getContainerListeners
	Returns an array of all the container listeners registered on this container.
27	Set<AWTKeyStroke> getFocusTraversalKeys <i>intid</i>
	Returns the Set of focus traversal keys for a given traversal operation for this Container.
28	FocusTraversalPolicy getFocusTraversalPolicy
	Returns the focus traversal policy that will manage keyboard traversal of this Container's children, or null if this Container is not a focus cycle root.
29	Insets getInsets
	Determines the insets of this container, which indicate the size of the container's border.
30	LayoutManager getLayout
	Gets the layout manager for this container.
31	<T extends EventListener> T[] getListenersClass <i>< T > listenerType</i>
	Returns an array of all the objects currently registered as FooListeners upon this Container.
32	Dimension getMaximumSize
	Returns the maximum size of this container.
33	Dimension getMinimumSize
	Returns the minimum size of this container.
34	Point getMousePosition <i>booleanallowChildren</i>
	Returns the position of the mouse pointer in this Container's coordinate space if the

Container is under the mouse pointer, otherwise returns null.

35 **Dimension** **getPreferredSize**

Returns the preferred size of this container.

36 **Insets** **insets**

Deprecated. As of JDK version 1.1, replaced by `getInsets`.

37 **void** **invalidate**

Invalidates the container.

38 **boolean** **isAncestorOf***Componentc*

Checks if the component is contained in the component hierarchy of this container.

39 **boolean** **isFocusCycleRoot**

Returns whether this Container is the root of a focus traversal cycle.

40 **boolean** **isFocusCycleRoot***Containercontainer*

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

41 **boolean** **isFocusTraversalPolicyProvider**

Returns whether this container provides focus traversal policy.

42 **boolean** **isFocusTraversalPolicySet**

Returns whether the focus traversal policy has been explicitly set for this Container.

43 **void** **layout**

Deprecated. As of JDK version 1.1, replaced by `doLayout`.

44 **void** **list***PrintStreamout, intindent*

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

45 **void** **list***PrintWriterout, intindent*

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

46 **Component** **locate***intx, inty*

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

47 **Dimension** **minimumSize**

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

48 **void** **paint***Graphicsg*

Paints the container.

49 **void paintComponents***Graphicsg*

Paints each of the components in this container.

50 **protected String paramString**

Returns a string representing the state of this Container.

51 **Dimension preferredSize**

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

52 **void print***Graphicsg*

Prints the container.

53 **void printComponents***Graphicsg*

Prints each of the components in this container.

54 **protected void processContainerEvent***ContainerEvente*

Processes container events occurring on this container by dispatching them to any registered ContainerListener objects.

55 **protected void processEvent***AWTEvente*

Processes events on this container.

56 **void remove***Componentcomp*

Removes the specified component from this container.

57 **void remove***intindex*

Removes the component, specified by index, from this container.

58 **void removeAll**

Removes all the components from this container.

59 **void removeContainerListener***ContainerListenerl*

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

60 **void removeNotify**

Makes this Container undisplayable by removing its connection to its native screen resource.

61 **void setComponentZOrder***Componentcomp, intindex*

Moves the specified component to the specified z-order index in the container.

- 62 **void setFocusCycleRoot***boolean focusCycleRoot*
Sets whether this Container is the root of a focus traversal cycle.
- 63 **void setFocusTraversalKeys***int id, Set < ? extends AWTKeyStroke > keystrokes*
Sets the focus traversal keys for a given traversal operation for this Container.
- 64 **void setFocusTraversalPolicy***FocusTraversalPolicy policy*
Sets the focus traversal policy that will manage keyboard traversal of this Container's children, if this Container is a focus cycle root.
- 65 **void setFocusTraversalPolicyProvider***boolean provider*
Sets whether this container will be used to provide focus traversal policy.
- 66 **void setFont***Font f*
Sets the font of this container.
- 67 **void setLayout***LayoutManager mgr*
Sets the layout manager for this container.
- 68 **void transferFocusBackward**
Transfers the focus to the previous component, as though this Component were the focus owner.
- 69 **void transferFocusDownCycle**
Transfers the focus down one focus traversal cycle.
- 70 **void update***Graphics g*
Updates the container.
- 71 **void validate**
Validates this container and all of its subcomponents.
- 72 **protected void validateTree**
Recursively descends the container tree and recomputes the layout for any subtrees marked as needing it *those marked as invalid*.

Methods inherited

This class inherits methods from the following classes:

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