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:

```java
public class Container extends Component
```

Class constructors

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Constructor &amp; Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Container</strong></td>
</tr>
<tr>
<td></td>
<td>This creates a new Container.</td>
</tr>
</tbody>
</table>

Class methods

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Method &amp; Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Component add</strong> Componentcomp</td>
</tr>
<tr>
<td></td>
<td>Appends the specified component to the end of this container.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Component add</strong> Componentcomp, int index</td>
</tr>
<tr>
<td></td>
<td>Adds the specified component to this container at the given position.</td>
</tr>
<tr>
<td>3</td>
<td><strong>void add</strong> Componentcomp, Object constraints</td>
</tr>
<tr>
<td></td>
<td>Adds the specified component to the end of this container.</td>
</tr>
<tr>
<td>4</td>
<td><strong>void add</strong> Componentcomp, Object constraints, int index</td>
</tr>
<tr>
<td></td>
<td>Adds the specified component to this container with the specified constraints at the specified index.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Component add</strong> String name, Component comp</td>
</tr>
<tr>
<td></td>
<td>Adds the specified component to this container.</td>
</tr>
</tbody>
</table>
void addContainerListener(ContainerListener

Adds the specified container listener to receive container events from this container.

protected void addImpl(Component comp, Object constraints, int index

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

void addNotify

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

void addPropertyChangeListener(PropertyChangeListener listener

Adds a PropertyChangeListener to the listener list.

void add Property Change Listener(String propertyName, PropertyChangeListener listener

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

void applyComponentOrientation(ComponentOrientation

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

boolean areFocusTraversalKeysSet(int id

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

int countComponents

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

void deliverEvent(Event

Deprecated. As of JDK version 1.1, replaced by dispatchEvent

void doLayout

Causes this container to lay out its components.

Component findComponentAt(int x, int y

Locates the visible child component that contains the specified position.

Component findComponentAt(Point p

Locates the visible child component that contains the specified point.
float getAlignmentX
Returns the alignment along the x axis.

float getAlignmentY
Returns the alignment along the y axis.

Component getComponent
Gets the nth component in this container.

Component getComponentAt
Locates the component that contains the x,y position.

Component getComponentAt
Gets the component that contains the specified point.

int getComponentCount
Gets the number of components in this panel.

Component[] getComponents
Gets all the components in this container.

int getComponentZOrder
Returns the z-order index of the component inside the container.

ContainerListener[] getContainerListeners
Returns an array of all the container listeners registered on this container.

Set<AWTKeyStroke> getFocusTraversalKeys
Returns the Set of focus traversal keys for a given traversal operation for this Container.

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.

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[] getListeners`  

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**

Retruns 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**

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**

Returns whether the specified Container is the focus cycle root of this Container's focus traversal cycle.
boolean isFocusTraversalPolicyProvider
Returns whether this container provides focus traversal policy.

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

void layout
Deprecated. As of JDK version 1.1, replaced by doLayout.

void listPrintStreamout, indent
Prints a listing of this container to the specified output stream.

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

Component locateintx, inty
Deprecated. As of JDK version 1.1, replaced by getComponentAtint, int.

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

void paintGraphicsg
Paints the container.

void paintComponentsGraphicsg
Paints each of the components in this container.

protected String paramString
Returns a string representing the state of this Container.

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

void printGraphicsg
Prints the container.

53  

```java
void printComponents(Graphics g)
```

Prints each of the components in this container.

54  

```java
protected void processContainerEvent(ContainerEvent e)
```

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

55  

```java
protected void processEvent(AWTEvent e)
```

Processes events on this container.

56  

```java
void removeComponent(Component comp)
```

Removes the specified component from this container.

57  

```java
void remove(int index)
```

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

58  

```java
void removeAll()
```

Removes all the components from this container.

59  

```java
void removeContainerListener(ContainerListener l)
```

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

60  

```java
void removeNotify()
```

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

61  

```java
void setComponentZOrder(Component comp, int index)
```

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

62  

```java
void setFocusCycleRoot(boolean focusCycleRoot)
```

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

63  

```java
void setFocusTraversalKeys(int id, Set< ? extends AWTKeyStroke > keystrokes)
```

Sets the focus traversal keys for a given traversal operation for this Container.
void setFocusTraversalPolicy

Sets the focus traversal policy that will manage keyboard traversal of this Container's children, if this Container is a focus cycle root.

void setFocusTraversalPolicyProvider

Sets whether this container will be used to provide focus traversal policy.

void setFont

Sets the font of this container.

void setLayout

Sets the layout manager for this container.

void transferFocusBackward

Transfers the focus to the previous component, as though this Component were the focus owner.

void transferFocusDownCycle

Transfers the focus down one focus traversal cycle.

void updateGraphics

Updates the container.

void validate

Validates this container and all of its subcomponents.

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