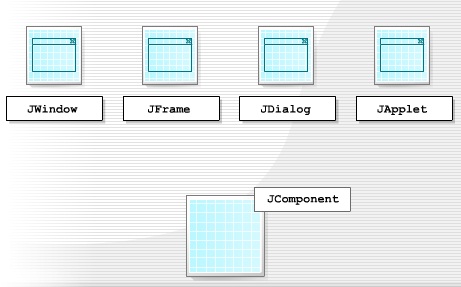
## Pane

In Java, a component is the basic user interface object and is found in all Java applications. Components include lists, buttons, panels, and windows.

A container is a component that holds and manages other components. Containers display components using a layout manager.

Swing components inherit from the javax.Swing.JComponent class, which is the root of the Swing component hierarchy. JComponent, in turn, inherits from the Container class in the Abstract Windowing Toolkit (AWT). So Swing is based on classes inherited from AWT.

Swing provides the following useful top-level containers, all of which inherit from JComponent:

[](http://1.bp.blogspot.com/--zXC96ryIOA/TzTOKFVFytI/AAAAAAAAAGI/qisB4_FqkRE/s1600/Swing+components1.6.jpg)

**JWindow**

JWindow is a top-level window that doesn't have any trimmings and can be displayed anywhere on a desktop. JWindow is a heavyweight component. You usually use JWindow to create pop-up windows and "splash" screens. JWindow extends AWT's Window class.

**JFrame**

JFrame is a top-level window that can contain borders and menu bars. JFrame is a subclass of JWindow and is thus a heavyweight component. You place a JFrame on a JWindow. JFrame extends AWT's Frame class.

**JDialog**

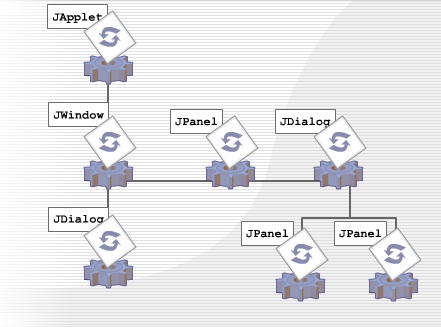
JDialog is a lightweight component that you use to create dialog windows. You can place dialog windows on a JFrame or JApplet. JDialog extends AWT's Dialog class.

**JApplet**

JApplet is a container that provides the basis for applets that run within web browsers. JApplet is a lightweight component that can contain other graphical user interface (GUI) components. JApplet extends AWT's Applet class.

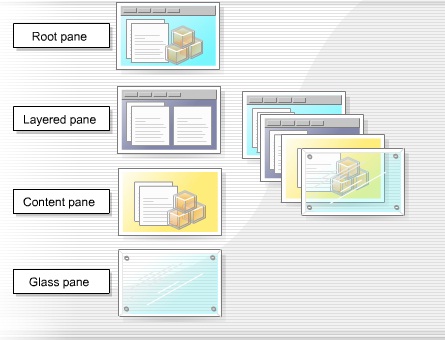
All Swing components - including the JApplet and JDialog containers - need to be contained at some level inside a JWindow or JFrame.

Each top-level container depends on another intermediate container called the root, which provides a number of components to each.

[](http://4.bp.blogspot.com/-dRDplceQMw8/TzTO9vTtF9I/AAAAAAAAAGQ/aY1g5wwzj04/s1600/root+components1.7.jpg)

JApplet is the root container for Swing applets and JFrame is the root container for a standalone GUI application.

Each top-level container consists of the following panes:

[](http://1.bp.blogspot.com/-6cpCqUZjLMM/TzTPflZKc8I/AAAAAAAAAGY/Dly8bpsEL7o/s1600/top+level+container1.8.jpg)

**Root pane**

The root pane is an intermediate container that manages the layered pane, content pane, and glass pane. It can also manage an optional menu bar. You use a root pane to paint over multiple components or to catch input events.

**Layered pane**

The layered pane contains the content pane and the optional menu bar. It can also contain other components, which it arranges so that they overlap each other. This enables you to add pop-up menus to applications. The layered pane provides six functional layers in which you place the components you add to it. You use each of these functional layers for a specific function.

**Content pane**

The content pane holds all the visible components of the root pane, except the menu bar. It covers the visible section of the JFrame or JWindow and you use it to add components to the display area. Java automatically creates a content pane when you create a JFrame or JWindow but you can create your own content pane, which has to be opaque.

**Glass pane**

The glass pane is invisible by default but you can make it visible. When it is visible, it covers the components of the content pane, blocks all input events from reading these components, and can paint over an existing area containing one or more components.

One of the enhancements to JTabbedPane is the use of a component to represent the tab in a JTabbedPane. This new feature offers a convenient way to show several items in a small amount of space. It does this by dividing the information across separate tabs so that a user can select

* one tab to list a particular set of components
* a different tab to list a different set of components

By adding a **Close** button, you can enable the removal of the current tab from JTabbedPane.