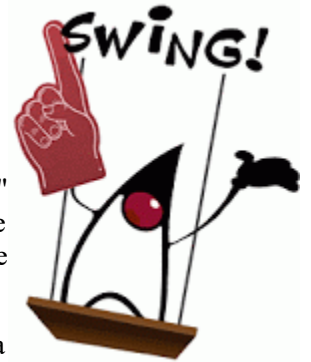
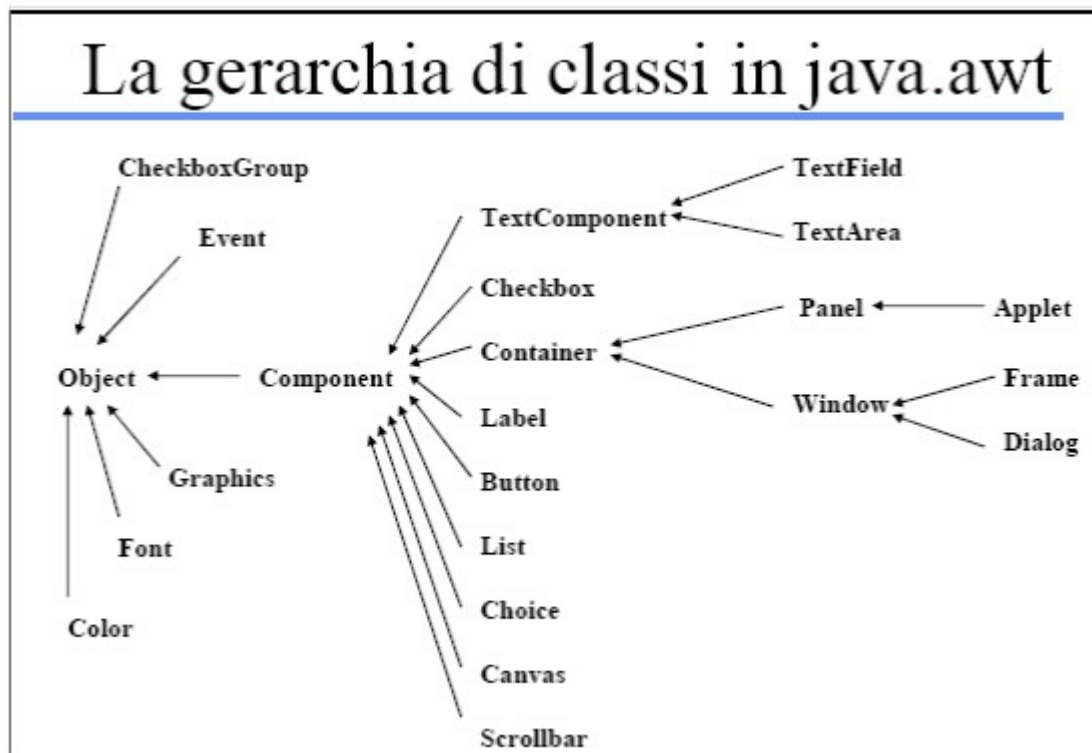


AWT vs SWING



La più grande differenza tra i componenti AWT ed i componenti Swing ("*lightweight*" o leggeri) è che i componenti Swing sono implementati senza una riga di codice nativo: scritti in codice Java senza ricorso alle primitive di sistema, direttamente *disegnati dalla JVM*.

Grazie all'assenza di questa restrizione possono avere molte più funzionalità rispetto a quelli contenuti nell'AWT. Anche l'oggetto swing più semplice () ha maggiori funzionalità del suo analogo presente nell'AWT (detto "*heavyweight*" o pesante cioè disegnato dall'hardware e non dal software proprio perché crea un "*peer*" - un aggancio al sistema sottostante - per ciascun componente. È il sistema operativo ospite che fornisce, per esempio, il bottone o la listbox (tecnicamente *widget*).



JButton e JLabel Swing possono mostrare immagini al posto, o in combinazione, del testo.

Si può facilmente cambiare il comportamento o la forma di un componente Swing sia invocando dei metodi su di essi che creando delle sottoclassi.

I componenti Swing possono essere in parte trasparenti e non devono per forza essere rettangolari. I bottoni, per esempio, possono essere tondi.

I componenti Swing permettono di specificare quale aspetto della GUI usare. Invece, i *component* di AWT hanno sempre il [look and feel](#) della piattaforma nativa.

Un'altra interessante caratteristica è quella che i componenti Swing con uno stato utilizzano un modello per mantenerlo.



Un JSlider, per esempio, utilizza un oggetto BoundedRangeModel per mantenere il suo valore corrente ed una forbice di valori accettabili. I modelli sono avviati automaticamente, così che non si ha a che fare con essi finché non si troverà vantaggioso sfruttare le possibilità che essi possono offrire.

Il **difetto principale**¹ della libreria AWT è quello di essere **piuttosto povera**: le interfacce grafiche costruite infatti risultano scarse e brutte. La ragione di questa povertà è da ricercare nelle scelte di progetto della AWT. Tale libreria infatti si appoggia pesantemente al sistema operativo ospite, utilizzando i componenti di interfaccia grafica già presenti nel sistema. Per questo motivo la AWT segue la logica del "minimo comun denominatore", e fornisce al programmatore solo quei componenti di interfaccia grafica che sono comuni ai vari sistemi operativi a finestre esistenti che la AWT supporta, ovvero Windows, MacOS e Unix/Motif. Così abbiamo le MenuBar, ma *non* abbiamo, per esempio, le ToolBar che non sono "di sistema" sotto Unix o Mac. Quindi mantenendo questa impostazione, per garantire la **portabilità delle applicazioni** tra i vari sistemi, di meglio non si può fare.

In effetti il fatto che i programmi in Java non fossero né particolarmente gradevoli all'occhio né efficienti nell'interazione e ricchi di funzionalità dal punto di vista dell'utente, è stato un chiaro freno alla diffusione di Java. Un programma Java sfigurava ampiamente confrontato ad un programma nativo di Windows o Mac.

Liberi dal vincolo del "minimo comun denominatore", i progettisti di **Swing** hanno scelto di percorrere la via opposta, creando un package ricco di funzionalità spesso non presenti nella piattaforma ospite.

Per risolvere definitivamente il problema della differenza di visualizzazione, viene offerta al programmatore la possibilità di scegliere il **Look & Feel** con cui visualizzare le proprie GUI, al contrario di quanto avveniva su AWT, che era vincolato alla rappresentazione dei componenti della macchina ospite.

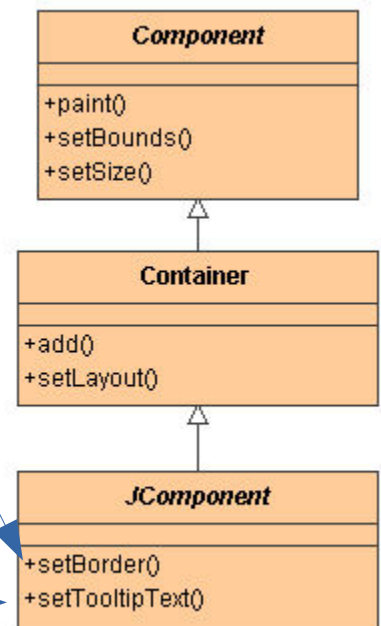
Questa caratteristica, denominata "**Pluggable Look & Feel**", è forse l'aspetto più sorprendente di Swing, in quanto è totalmente trasparente al programmatore. Qualsiasi GUI realizzata in Swing può essere visualizzata in uno qualunque dei L&F disponibili sul sistema, senza che venga richiesta alcuna modifica a livello di codice sorgente. Non esistono problemi di compatibilità nel passaggio tra un L&F e l'altro, dal momento che i componenti Swing sono stati progettati per mantenere una netta separazione tra la rappresentazione grafica di un componente, le informazioni che contiene e il controllo che implementa. Questo design, definito "**Model Control View**" (MCV), risulta, nella maggior parte dei casi, del tutto trasparente al programmatore, che può utilizzare i componenti Swing nello stesso modo in cui era abituato con quelli AWT. Nel momento in cui ci fosse la necessità di progettare componenti personalizzati, si potranno sfruttare pienamente le funzionalità offerte da questo design.

La presenza di queste *features* non va ad incidere sulla caratteristica più richiesta da un programmatore: la **facilità d'uso**.

Il padre di tutti i componenti Swing: JComponent

La classe JComponent fornisce in esclusiva alcune funzionalità, che vengono ereditate da qualunque componente Swing. Tra queste possiamo segnalare:

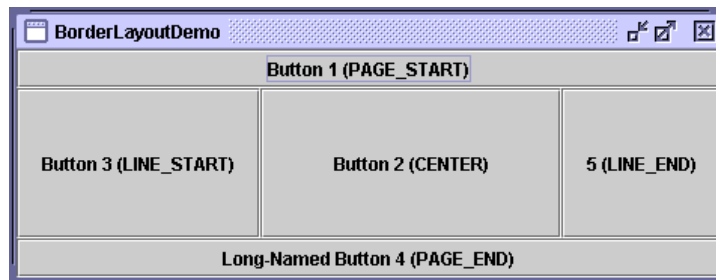
- **Bordi**: ogni componente Swing può essere dotato di un bordo grazie al metodo `setBorder(Border)`. La classe `BorderFactory` fornisce strumenti per creare diversi tipi di bordo, da semplici contorni a sofisticate decorazioni
- **Pluggable look and feel**: è la possibilità offerta da ogni JComponent di essere visualizzato in maniera differente a seconda del Look & Feel impostato dall'utente.
- **Tool tips**: il metodo `setToolTipText(String)` permette di aggiungere un `ToolTip`, un messaggio testuale che compare sul componente dopo che l'utente vi ha lasciato fermo il mouse per qualche istante.



SWING <http://java.sun.com/docs/books/tutorial/uiswing/index.html>

Layout <http://java.sun.com/docs/books/tutorial/uiswing/layout/visual.html>

BorderLayout

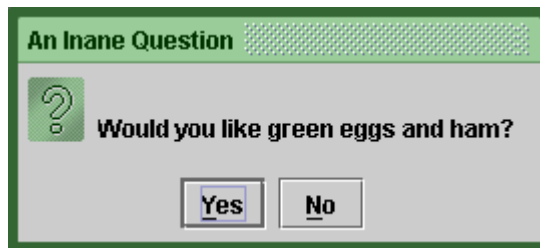


Containers <http://java.sun.com/docs/books/tutorial/uiswing/components/components.html>

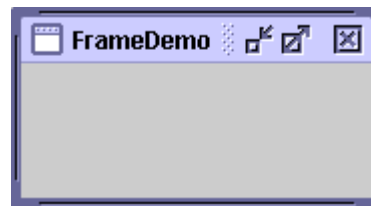
Top-Level Containers



[Applet](#)

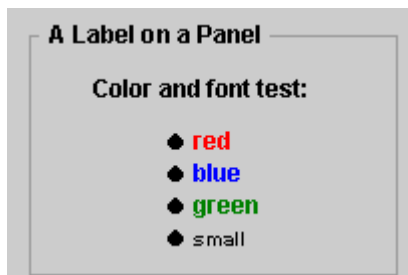


[Dialog](#)

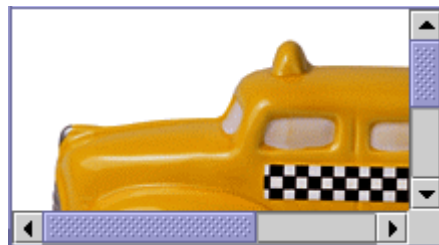


[Frame](#)

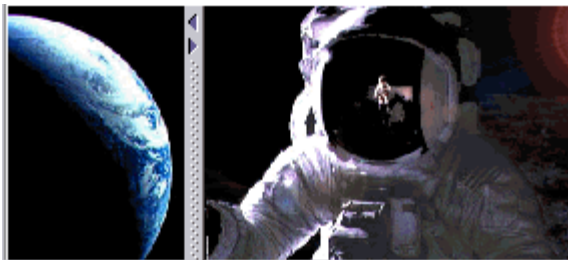
General-Purpose Containers



[Panel](#)



[Scroll pane](#)



[Split pane](#)

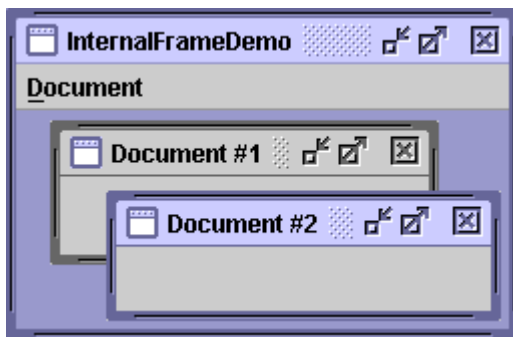


[Tabbed pane](#)

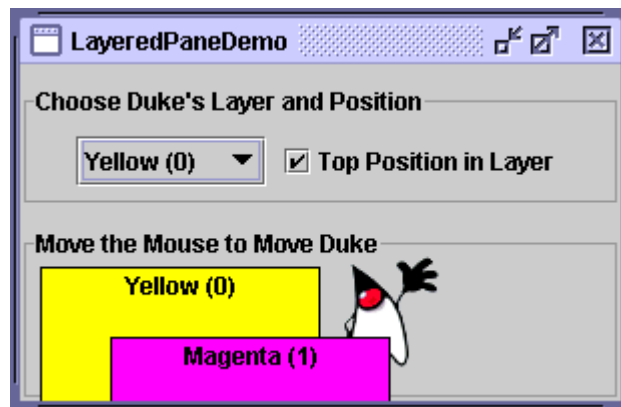


[Tool bar](#)

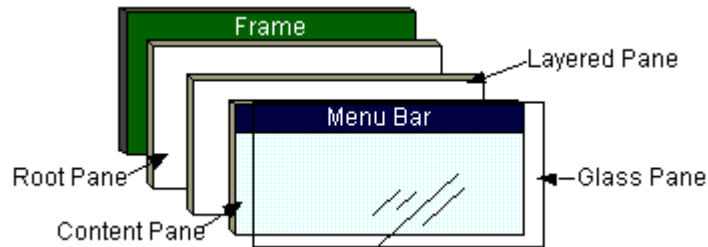
Special-Purpose Containers



[Internal frame](#)



[Layered pane](#)

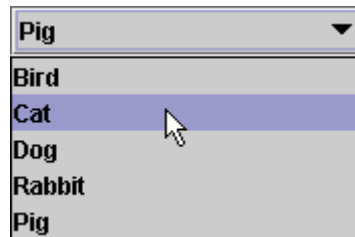


[Root pane](#)

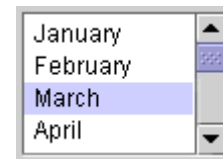
Basic Controls



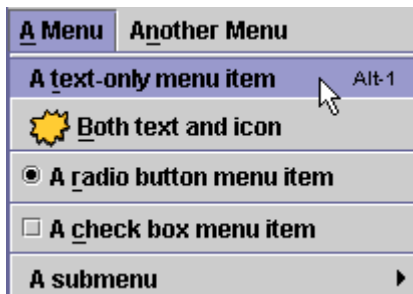
[Buttons](#)



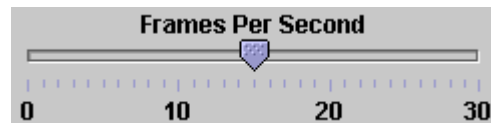
[Combo box](#)



[List](#)



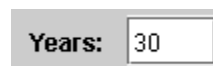
[Menu](#)



[Slider](#)

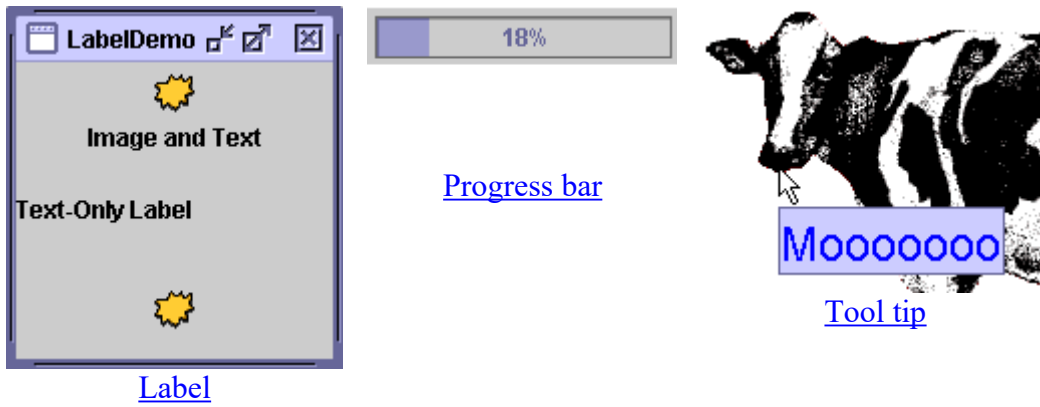


[Spinner](#)



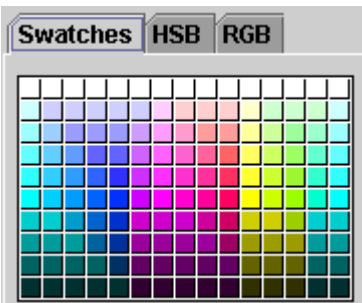
[Text field](#) or [Formatted text field](#)

Uneditable Information Displays

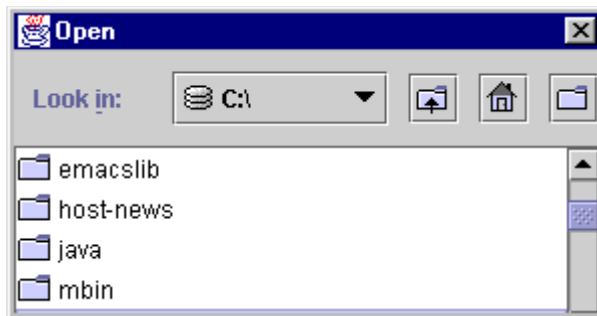


[Label](#)

Interactive Displays of Highly Formatted Information



[Color chooser](#)



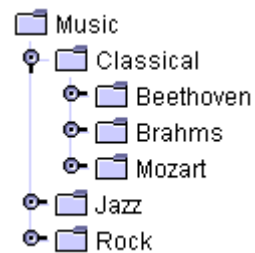
[File chooser](#)

First Name	Last Name	Favorite Food
Jeff	Dinkins	
Ewan	Dinkins	
Amy	Fowler	
Hania	Gajewska	
David	Geary	

[Table](#)



[Text](#)



[Tree](#)