UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA
DIPARTIMENTO DI ECCELLENZA

Università di degli Studi di Salerno
Dipartimento di Informatica

Programmazione ad Oggetti

a.a. 2023-2024

Interfacciamento Grafico in JAVA

Docente: Prof. Massimo Ficco

E-mail: *mficco@unisa.it*

Un po' di storia

All'introduzione di Java 1.0, Sun mise a disposizione una libreria di classi definita **Abstract Window Toolkit** o AWT, per la programmazione fondamentale della GUI

Il modo in cui AWT gestisce gli elementi delle interfacce utente consiste nel delegare la loro creazione e comportamento al toolkit nativo della piattaforma di destinazione

L'idea è che il programmatore specifichi solo la posizione e il comportamento degli elementi della UI e Java crei i peer

Il programma quindi, "in teoria", potrebbe girare su qualsiasi piattaforma adeguando l'aspetto alla piattaforma di destinazione

Limiti di AWT

Presto ci si è resi conto dei limiti che un tale approccio comportava:

Impossibilità di scrivere una libreria grafica portabile di alta qualità che dipendesse solo dagli elementi dell'interfaccia utente nativa poiché possono presentarsi differenze di comportamento sulle diverse piattaforme;

Difficoltà di dare agli utenti un'esperienza coerente e prevedibile

Alcuni ambienti grafici (X11, Motif) non prevedono una molteplicità di componenti: di conseguenza si è giunti ad una libreria portabile ma limitata (minimo comune denominatore)

Le GUI non si presentano bene come le applicazioni native di Windows e Macintosh, né hanno il tipo di funzionalità che gli utenti di queste piattaforme si aspettano

La libreria UI AWT presenta bug diversi su piattaforme diverse

Da AWT a Swing

Nel 1996 Netscape crea una libreria grafica detta Internet Foundation Classes (IFC) che adotta un metodo completamente differente:

Gli elementi dell'interfaccia (pulsanti, menù, ecc.) vengono dipinti in finestre vuote

L'unica funzionalità peer richiesta è il modo in cui costruire le finestre e dipingere in esse

Sun, lavorando con Netscape per perfezionare questo approccio, crea una libreria di componenti grafiche denominata **SWING**

SWING consente ai "widget" di presentarsi e comportarsi allo stesso modo su qualsiasi piattaforma sulla quale il programma viene eseguito

Da AWT a SWING

SWING non è un completo sostituto di AWT

Molte delle classi SWING forniscono componenti alternative, più utili e più capaci delle loro controparti AWT

- La classe JButtondelle SWING offre più funzionalità della classe Button di AWT
- L'architettura essenziale, in particolare la gestione degli eventi, rimane la stessa

Gli elementi di GUI basati su SWING saranno visualizzati più lentamente rispetto a quelli basati sui peer (AWT)

- Su un computer ragionevolmente moderno la differenza di velocità non dovrebbe costituire un problema

Di solito le SWING usano un nome di classe che semplicemente aggiunge una **J** davanti al nome originale del componente AWT

- Ad es.: JTextArea è la controparte della classe AWT TextArea

Vantaggi di SWING

Ha una serie di elementi di interfaccia utente molto più ricca e dipende molto meno dalla piattaforma di base

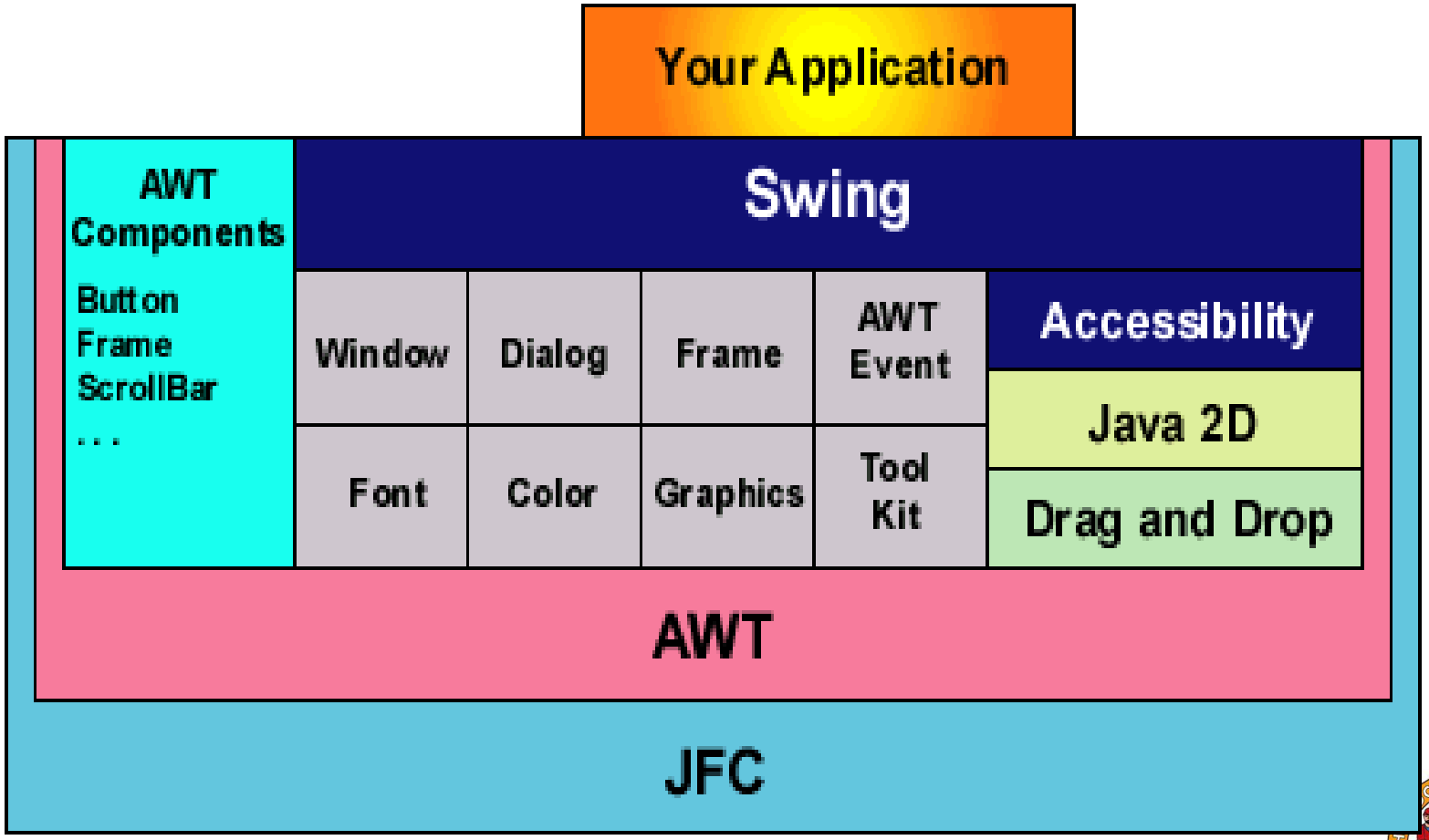
È molto meno incline ai bug specifici della piattaforma

Offrirà un risultato coerente su tutte le piattaforme

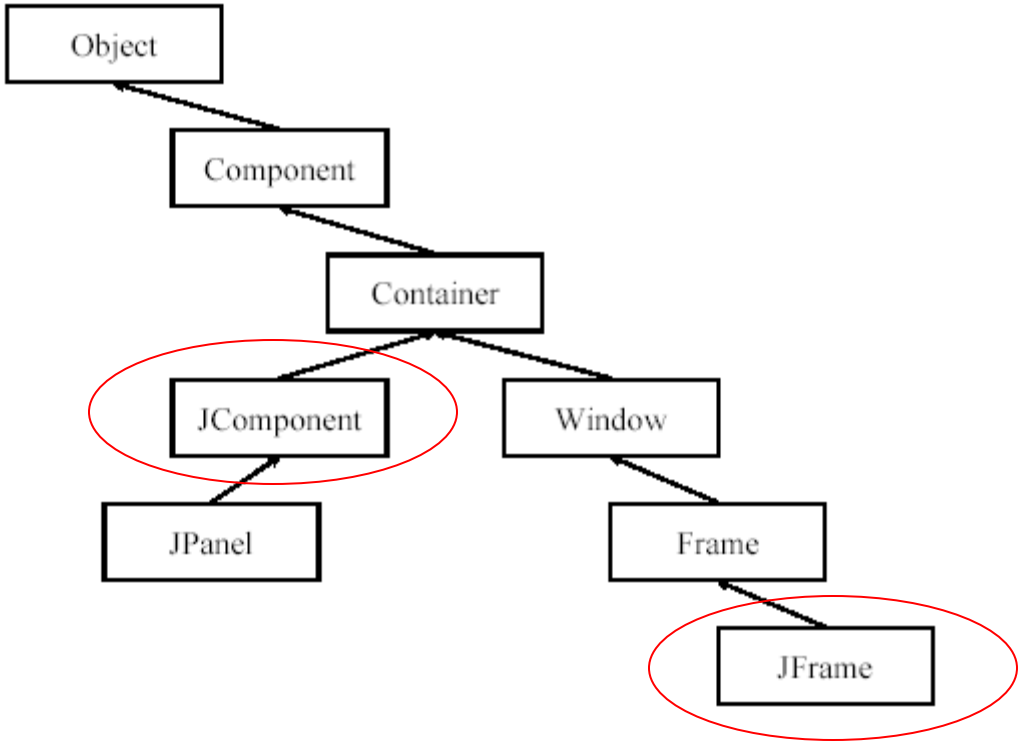
L'interfaccia grafica sarà diversa dai controlli nativi su alcune piattaforme, quindi l'utente avrà meno familiarità con essa

Per questo Sun ha sviluppato un'ambientazione indipendente dalla piattaforma, denominata Metal

OTHER APIs


Your Application
AWT Components
Button
Frame
ScrollBar
. . .
Swing
Window
Dialog
Frame
AWT Event
Accessibility
Java 2D
Font
Color
Graphics
Tool Kit
Drag and Drop
AWT
JFC

Le classi Principali



Creare una Finestra

Per utilizzare una classe Swing occorre importarla singolarmente (istruzione import) oppure importare tutte le classi come segue :

import javax.swing.*; ***// rende accessibili tutte le classi Swing***

Poiché le classi Swing ereditano delle proprietà dalle stesse superclassi di AWT, è possibile utilizzare contemporaneamente componenti di AWT e Swing.

È comunque consigliato limitarsi all'uso di Swing.

Frames

Una finestra top-level (non contenuta all'interno di un'altra) in Java prende il nome di **frame**

La libreria AWT ha una classe corrispondente basata sui peer, denominata Frame.

La versione SWING si chiama JFrame

JFrame estende Frame ed è uno dei pochissimi componenti SWING non disegnati su un'area di lavoro (canvas)

I frames sono esempi di contenitori

Un contenitore è un componente speciale che può contenere altri componenti

I **contenitori** e i loro **layout manager** determinano in che modo i **componenti** devono essere organizzati e visualizzati

Creare una Finestra

Rendere la nostra interfaccia una sottoclasse di JFrame

Inizializzarla invocando un opportuno metodo costruttore

Impostarne le dimensioni esprimendole in pixel

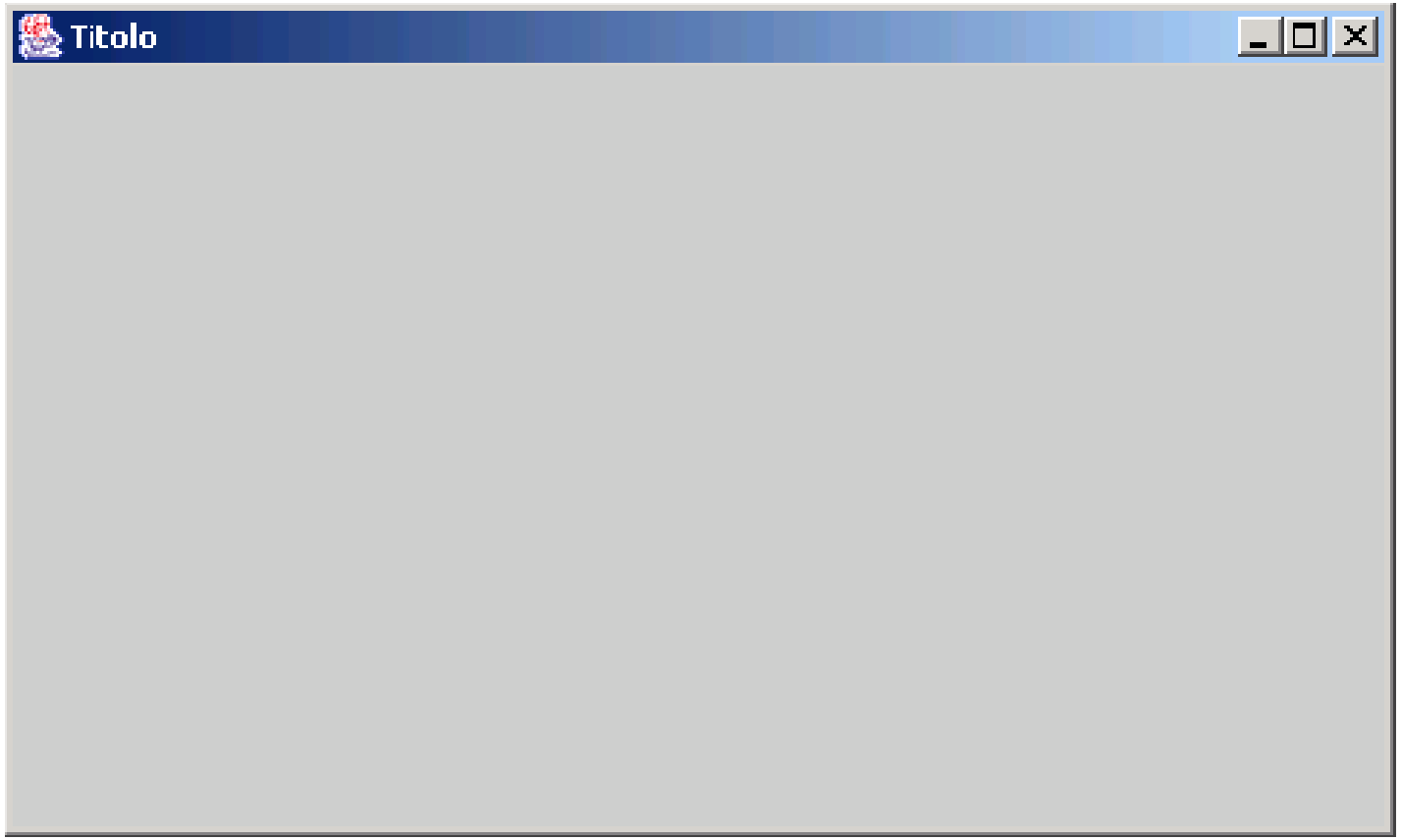
Associare un evento alla chiusura della finestra

Visualizzare il frame

Codice

```
public class Semplice extends JFrame{


        public Semplice(String titolo){
                super(titolo);
                setSize(500, 300);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }

        public static void main(String[] args) {

          Semplice frame = new Semplice(“Titolo");

          frame.setVisible(true);
         } }
```

Esecuzione



*Esempio: Look and Feel

```
import javax.swing.JFrame;
import javax.swing.UIManager;
public class LookFeel extends JFrame {
        public LookFeel(String titolo)
        {
                super(titolo);
                setSize(500, 300);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }
        public static void main(String[] args) {
        //UIManager.getSystemLookAndFeelClassName()
        //"com.sun.java.swing.plaf.motif.MotifLookAndFeel“,
        //UIManager.getCrossPlatformLookAndFeelClassName()
        //"com.sun.java.swing.plaf.windows.WindowsLookAndFeel"
        //"javax.swing.plaf.mac.MacLookAndFeel"
        try{
          UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
                }catch(Exception e){e.printStackTrace();}
          LookFeel frame = new LookFeel("SwingApplication");
          frame.setVisible(true);
        }}
```

*Look and Feel

È possibile cambiare il Look and Feel dell'interfaccia

Occorre utilizzare la classe UIManager della package java.swing.

Occorre specificare la classe che implementa il Look and Feel desiderato

- La classe deve essere disponibile sul sistema utilizzato

I Thread e gli Eventi

Ad ogni finestra è associato un thread di esecuzione

- Quando termina il main non termina il programma

Interagire con l'interfaccia grafica scatena eventi

- Occorre gestire i diversi utenti

Evento di chiusura

setDefaultCloseOperation(JFrame.FLAG);

Questo metodo permette di modificare l'azione di default di Java di continuare ad eseguire l'applicazione che ha generato il frame anche dopo aver cliccato sull'apposito bottone di chiusura del frame

FLAG può assumere i seguenti valori:

- **EXIT_ON_CLOSE**: terminazione del programma quando il frame viene chiuso
- **DISPOSE_ON_CLOSE**: chiude e libera il solo oggetto frame, il processo continua
- **DO_NOTHING_ON_CLOSE**: mantiene aperto il frame e continua il processo (ignoriamo l'evento)
- **HIDE_ON_CLOSE**: chiude il frame (lasciandolo in memoria) e continua l'esecuzione (es: Window Messenger)

Alcuni metodi di JFrame

Vari metodi per visualizzare o nascondere i frames :

- **show()** oppure **setVisible(true)** per rendere visibile il frame (che non lo è per default)
- **hide()** oppure **setVisible(false)** per nasconderlo
- **setBounds(int, int, int, int):** per scegliere la posizione iniziale
  - Argomenti (nell'ordine da sinistra a destra): x ed y dell'angolo superiore sinistro. larghezza ed altezza

Visualizzare informazioni in un frame

In Java i frame sono studiati per fungere esclusivamente da contenitori di componenti: Barre di menu, e altri elementi di GUI

In AWT era possibile inserire componenti direttamente in un frame, in SWING questo non è possibile

Componenti e immagini vengono inglobati in un altro componente, chiamato **pannello**, che viene aggiunto al frame

La struttura di un JFrame è molto complessa e composta da un numero di pannelli

Il pannello di interesse per le GUI è il pannello del contenuto
È in questo pannello che vengono inseriti tutti i componenti GUI

JPanel

I Pannelli sono realizzati dalla **classe JPanel**

Vengono aggiunti al pannello del contenuto

Sono elementi di GUI che godono di due utili proprietà:

- Hanno una superficie su cui è possibile disegnare
- Sono a loro volta dei contenitori; possono quindi a loro volta contenere componenti GUI quali pulsanti, barre di scorrimento, ecc.

Per aggiungere un pannello in un frame:

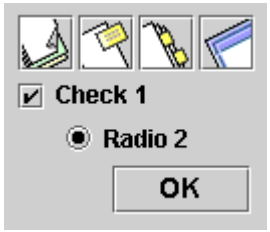
```
Container contentPane = frame.getContentPane();
```

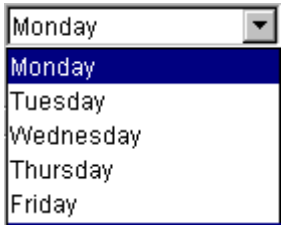
Composizione dei Pannelli

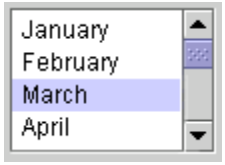
- I pannelli possono essere composti da altri sottopannelli
- Tutti devono essere inclusi nel pannello principale, detto “content pane”.
- È necessario impostare un pannello come “content pane” di ogni JFrame.
- I componenti (ad esempio i bottoni) vanno inseriti in un pannello dopo essere stati creati ed opportunamente inizializzati.

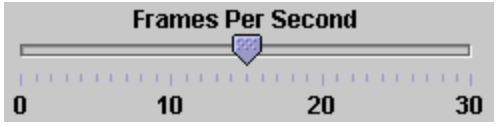
Alcuni Componenti

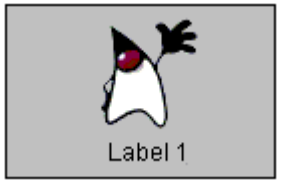
- Buttons
- Combo boxes
- Lists
- Menus
- Text Fields
- Labels





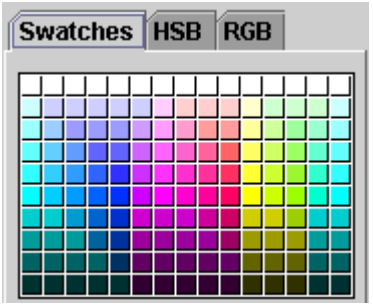


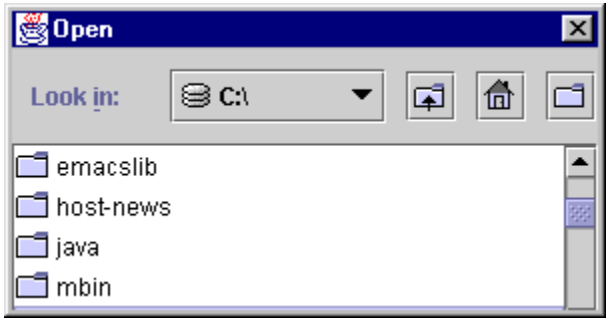


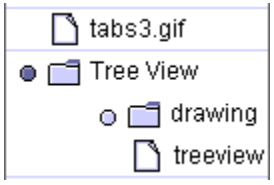


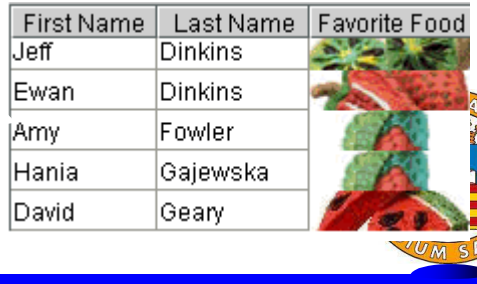
Alcuni Componenti

- Tool tips
- Progress bars
- Colour choosers
- File choosers
- Tables
- Text
- Trees



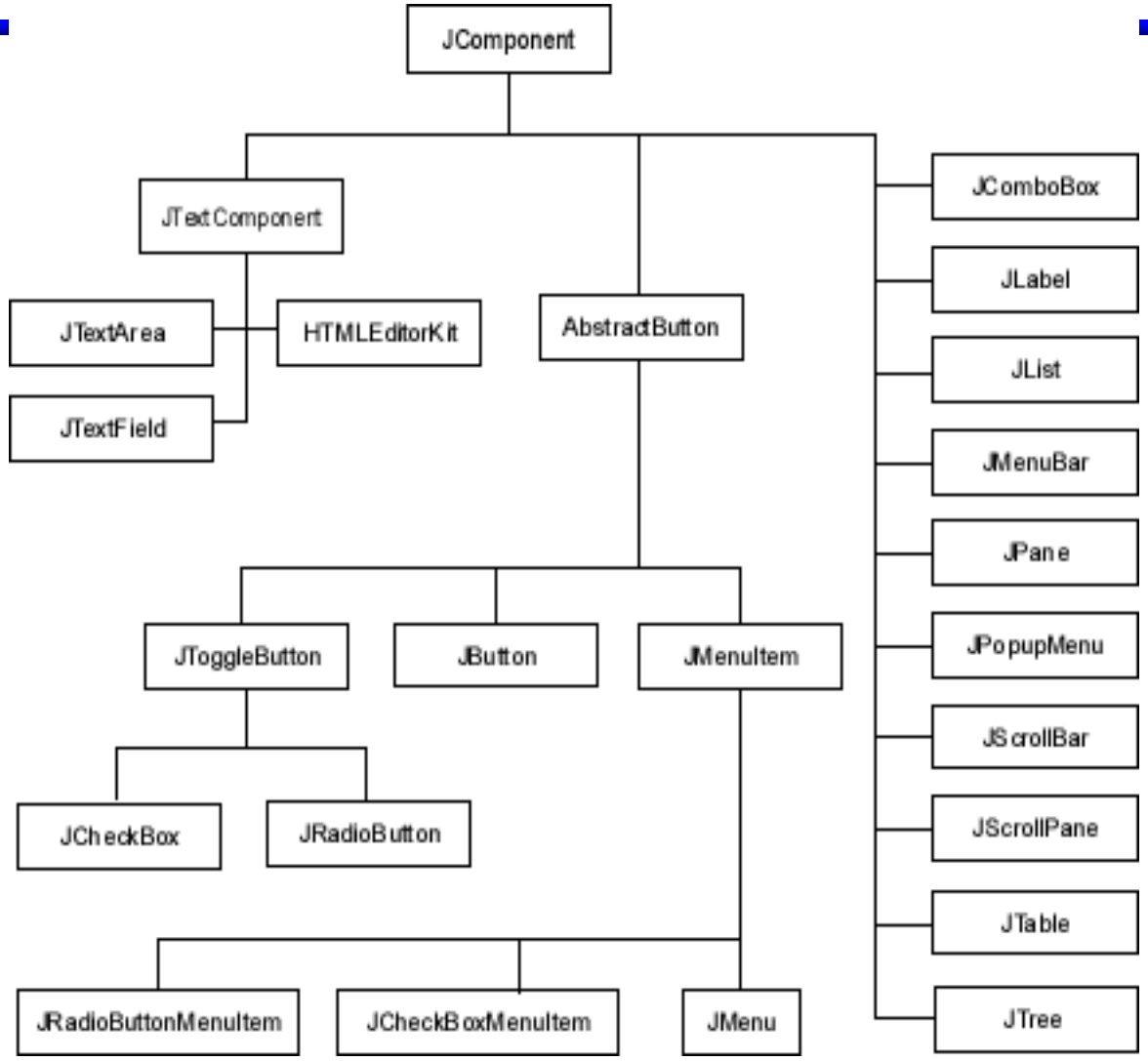






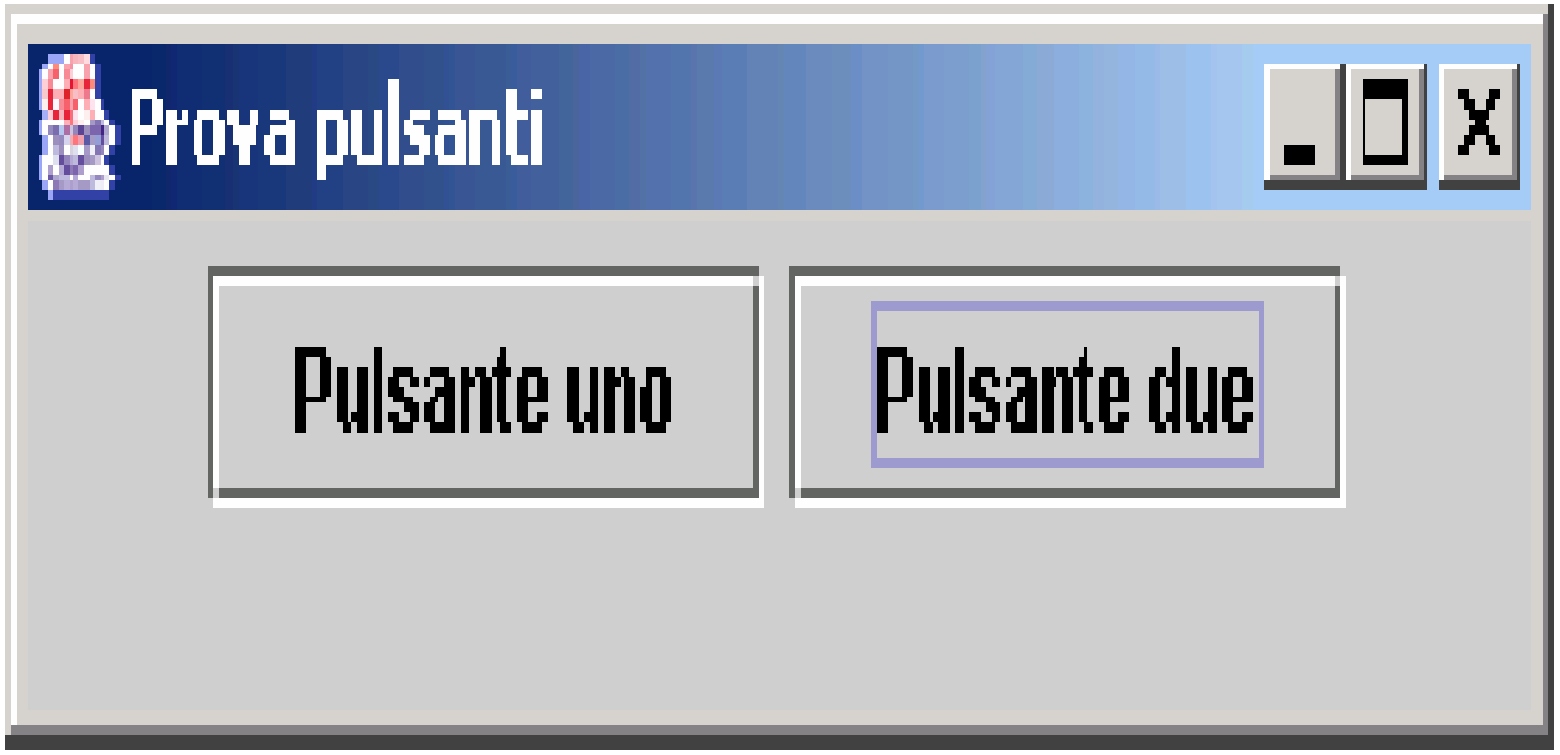
First Name	Last Name	Favorite Food
Jeff	Dinkins	
Ewan	Dinkins	
Amy	Fowler	
Hania	Gajewska	
David	Geary	

UI COMPONENTS



Esempio

```
import javax.swing.*;

public class Pulsanti extends Jframe {
   JButton pulsanteUno = new JButton("Pulsante uno");
   JButton pulsanteDue = new JButton("Pulsante due");

public Pulsanti() {
    super("Prova pulsanti");  //JFrame con titolo
    setSize(300, 80);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    JPanel pannello = new JPanel();
    pannello.add(pulsanteUno);
    pannello.add(pulsanteDue);
    setContentPane(pannello);
    show();
}
…
}
```


Prova pulsanti
Pulsante uno
Pulsante due

Operazioni sui componenti

È possibile attivare o disattivare componenti, renderli visibili o meno:

setEnabled(boolean)

setVisible(boolean)

È possibile ridimensionare pulsanti:

setSize(int, int)

setSize(Dimension)

È possibile verificare lo stato da codice:

isEnabled()

isVisible()

Dimension getSize()