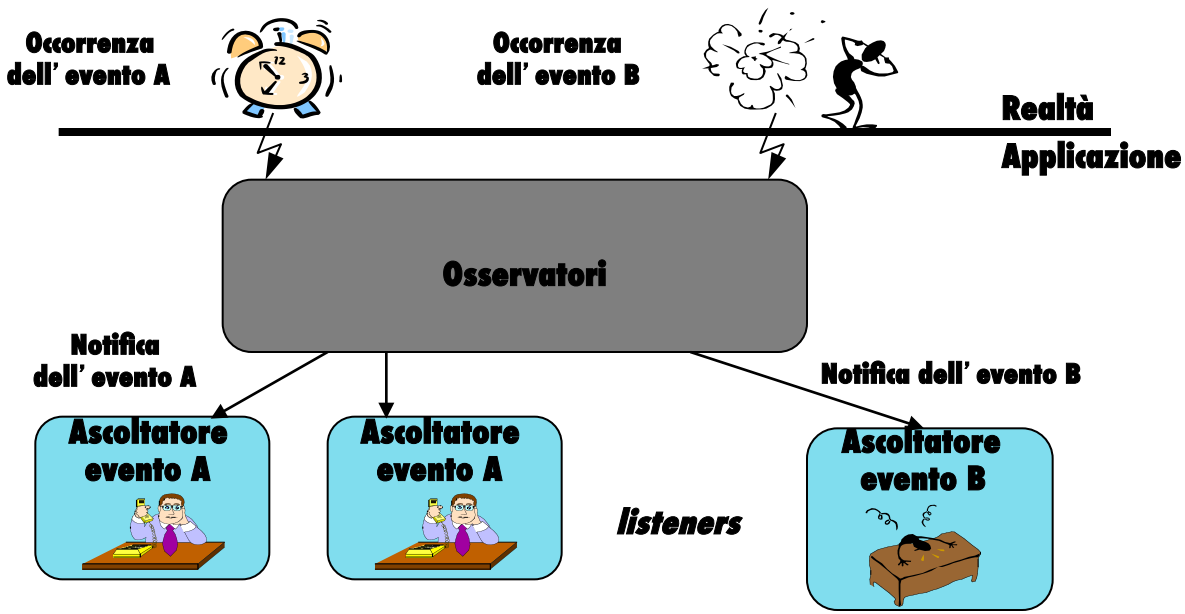
Esercitazioni Prova Finale
Introduzione a Swing

Lo stile basato su eventi

- I componenti interagiscono attraverso l'invio di messaggi broadcast o multicast
  - "Attenzione! C'è un incendio"
  - "Si avvisano i signori passeggeri che …"
- Ogni componente notifica un cambiamento nel proprio stato o in quello dell'ambiente inviando un messaggio
  - In tal caso agisce da "generatore dell'evento"
- Tutti i componenti interessati all'evento descritto da tale messaggio ne ricevono copia
  - In tal caso agiscono da "ascoltatori dell'evento"
- In generale, il generatore dell'evento non conosce né il numero né l'identità degli ascoltatori

Caratteristiche generali

- Con il paradigma a eventi
  - L'applicazione è puramente "reattiva"
  - Non è possibile identificare staticamente un flusso di controllo unitario
    - Il programma principale si limita a inizializzare l'applicazione, istanziando gli osservatori e associandovi gli opportuni handler

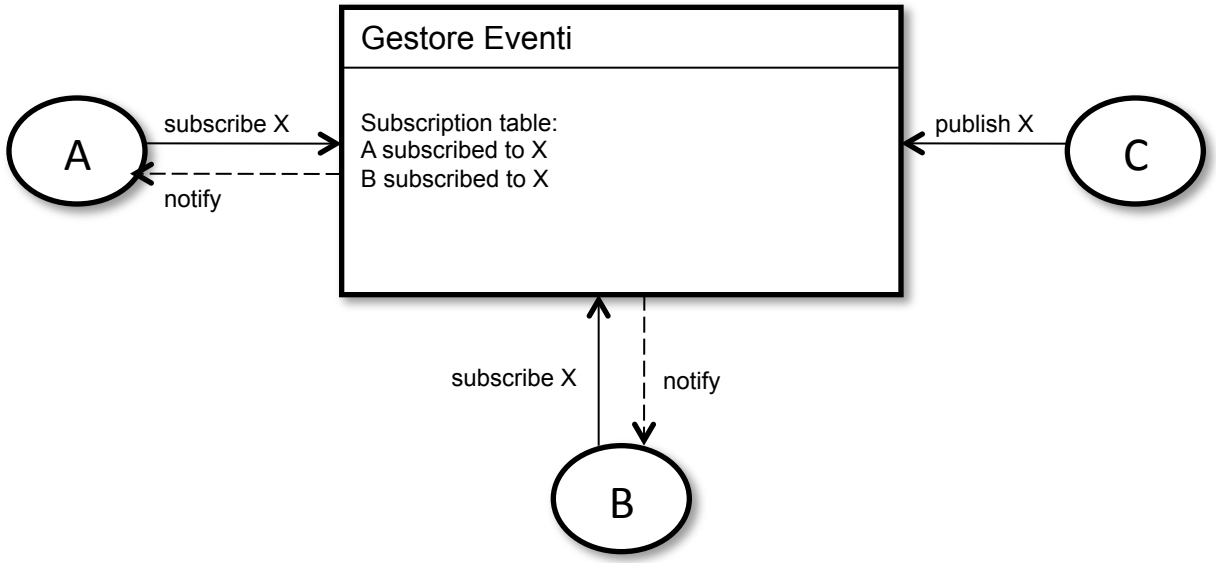


Vantaggi

- Utili per sistemi altamente dinamici ed evolutivi, in cui nuovi moduli si possono aggiungere o possono essere eliminati
- Per comunicare tra di loro, i moduli non devono conoscere le rispettive identità
  - come invece accade in Java
    - nomeModulo.nomeServizio (parametri)
- Si parla più in generale di paradigma di progettazione "publish and subscribe"
  - moduli che generano eventi ("publish")
  - moduli che sottoscrivono l'interesse ad essere notificati dell'occorrenza di certi eventi ("subscribe")

Observer Pattern / Publish-Subscribe Pattern

- *"Don't call us. Give us your telephone number, and we will call you if and when we have a job for you."* *[Hollywood Principle]*
- *"Instead of your program running the system, the system runs your program"*



Modello a eventi in Java

- Un'applicazione può essere costruita come componenti che interagiscono attraverso eventi
- Tutti i componenti di una determinata interfaccia possono
  - generare determinati eventi.
  - o mettersi in ascolto di determinati eventi.
- Qui noi vediamo il package javax.swing
  - libreria che fornisce le classi che consentono la progettazione delle interfacce utente secondo lo stile ad eventi
  - Altre librerie interessanti: Java2D, Java3D…

GUI

- GUI: Graphical User Interface
- L'interfaccia utente costituisce il mezzo con il quale l'utente interagisce con l'applicazione
  - costituisce il "look&feel"
- Fondamentale per l'usabilità del software
- Come si progetta?

“GUI programming is hard; everybody admits it.“

Stephen Ferg

Progettazione

- Separare GUI e parte funzionale
- La GUI si preoccupa di interagire con l'utente
  - visualizzare
  - accettare input
- La parte funzionale contiene la logica applicativa

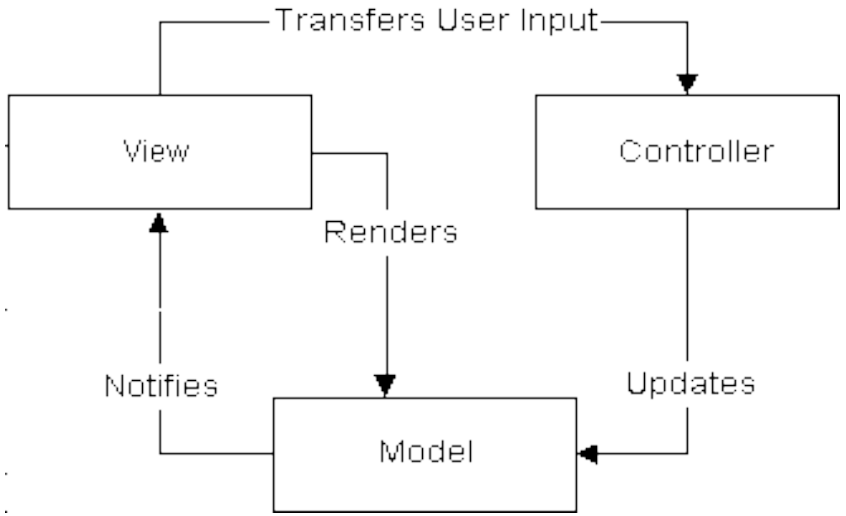
- Vantaggi
  - GUI modificabile senza toccare la parte funzionale e viceversa
  - Diverse strategie "look&feel" per la stessa applicazione
    - Spesso le modifiche si incentrano sulla GUI al fine di migliorare l'usabilità

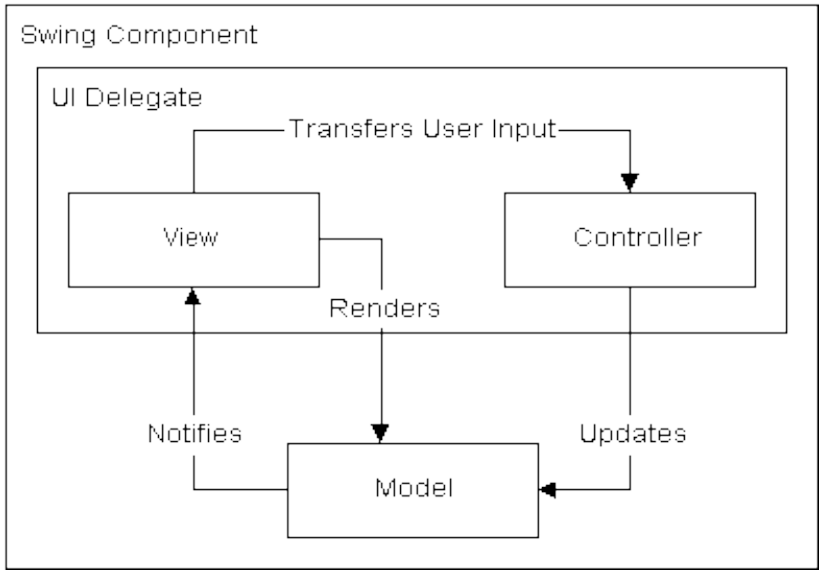
Model-View-Controller

- Swing si basa su un "paradigma" di progettazione che si incontra anche in altri campi:
  - Un componente ha un suo "modello logico"
  - Un componente ha un suo "aspetto visuale"
  - Un componente ha "comportamenti" che consentono la sua interazione con il resto dell'applicazione
- Esempio -> JButton
  - Model: Premuto/Rilasciato (variabile booleana)
  - View: Dipende dal look&feel
  - Controller:
    - Sulla base dell'input utente (bottone premuto) aggiorna il Model o implementa la logica applicativa.
    - In Java: ActionPerformed della classe ActionListener collegata al JButton (ci torneremo)

Model-View-Controller

- L'obiettivo del paradigma MVC è quello di disaccoppiare il *dato* dal suo *comportamento* e dalla sua *view.*
- Non è sempre possibile farlo.





AWT e Swing

- Abstract Windowing Toolkit (AWT)
  - E' la precedente versione delle librerie grafiche di JAVA. Fornisce ancora alcuni importanti componenti per il funzionamento e la creazione dell' interfaccia
- Swing
  - Consentono di realizzare applicazioni con un "look and feel" più aggiornato e elegante
  - Si basano su un modello di interazione introdotto da JAVA2
  - Hanno superato alcuni dei difetti di AWT
    - Chiedono servizi al sistema operativo, ma (normalmente) ad un livello più basso e riescono così ad avere un maggiore controllo del "look&feel"
    - Sono una "estensione" del core di JAVA e possono risultare più complesse da programmare
    - Consentono un maggior controllo del look&feel di un' applicazione e garantiscono il medesimo look&feel su tutte le piattaforme

AWT e Swing

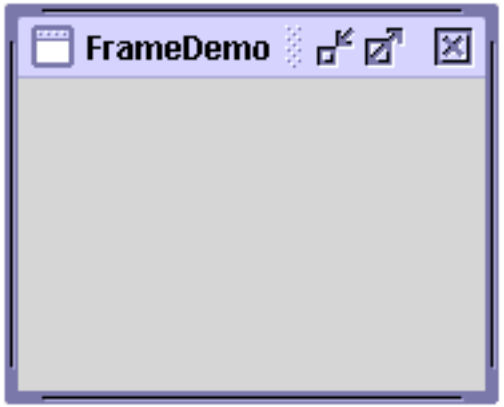
- Il nome di tutti i componenti Swing inizia con J
  - La convenzione è JXxxx
- I componenti Swing sono *lightweight*
  - Un componente lightweight implementa il suo look-and-feel direttamente in Java senza delegarlo al sistema operativo che ospita l'applicazione
    - Maggior uniformità di comportamento
- È bene non includere componenti Swing ed AWT in una stessa interfaccia:
  - I componenti AWT (heavyweight) vengono sempre mostrati "sopra" i componenti Swing (ad esempio con i Menu)
  - Problema dovuto alla mancanza di compatibilità fra i due framework

Un’interfaccia Swing

- Tre elementi fondamentali
  - Contenitori
  - Elementi grafici
  - Eventi

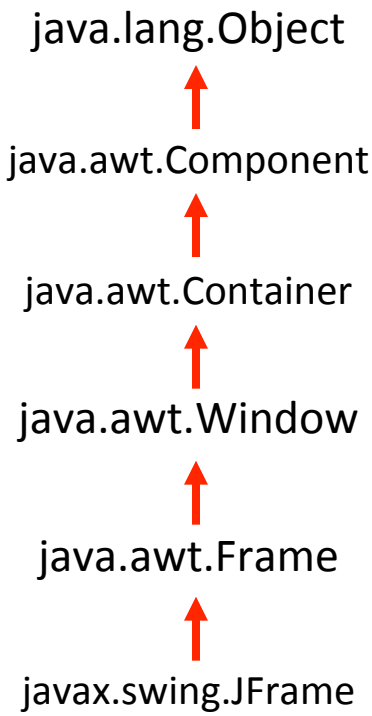
Frame

- Un frame
  - Definisce l'intelaiatura dell'interfaccia
  - Fornisce un rettangolo, "decorato" alla maniera delle finestre cui siamo abituati
  - E' un oggetto COMPLESSO!

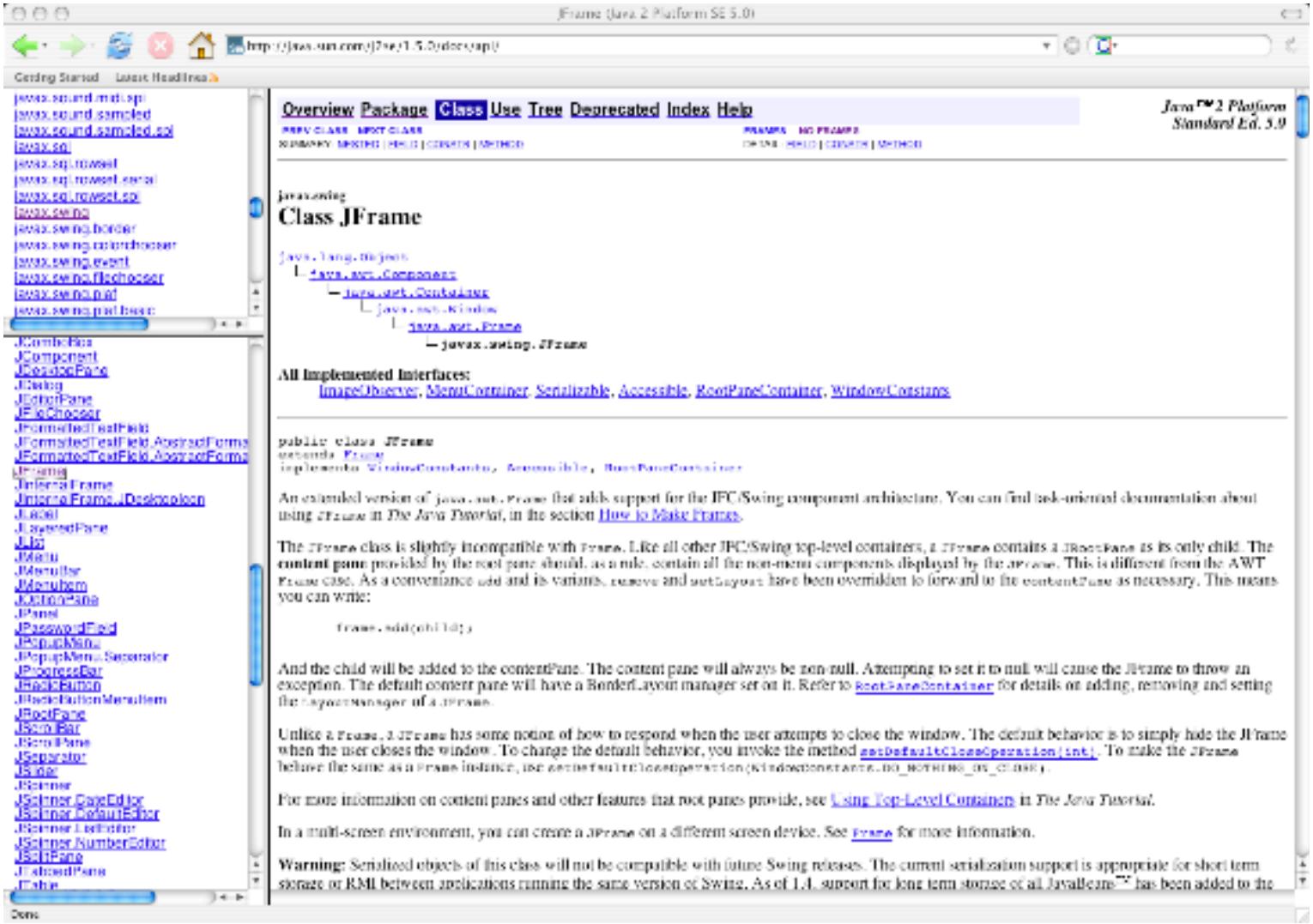


JFrame

- Come in tutta la programmazione ad oggetti occorre conoscere la "gerarchia" entro la quale si colloca JFrame
- Per conoscere la "derivazione" di ogni oggetto delle librerie GUI di Swing si deve fare ricorso alla documentazione on line di Java
  - http://download.oracle.com/javase/6/docs/api/



JFrame


Overview Package Class Use Tree Deprecated Index Help
Java™ 2 Platform Standard Ed. 5.0
javax.swing
Class JFrame
java.lang.Object
java.awt.Component
java.awt.Container
java.awt.Window
java.awt.Frame
javax.swing.JFrame
All Implemented Interfaces:
ImageObserver, MenuContainer, Serializable, Accessible, RootPaneContainer, WindowConstants
public class JFrame
extends Frame
implements WindowConstants, Accessible, RootPaneContainer
An extended version of java.awt.Frame that adds support for the JFC/Swing component architecture. You can find task-oriented documentation about using JFrame in The Java Tutorial, in the section How to Make Frames.
The JFrame class is slightly incompatible with Frame. Like all other JFC/Swing top-level containers, a JFrame contains a JRootPane as its only child. The content pane provided by the root pane should, as a rule, contain all the non-menu components displayed by the JFrame. This is different from the AWT Frame case. As a convenience add and its variants, remove and setLayout have been overridden to forward to the contentPane as necessary. This means you can write:
frame.add(child);
And the child will be added to the contentPane. The content pane will always be non-null. Attempting to set it to null will cause the JFrame to throw an exception. The default content pane will have a BorderLayout manager set on it. Refer to RootPaneContainer for details on adding, removing and setting the LayoutManager of a JFrame.
Unlike a Frame, a JFrame has some notion of how to respond when the user attempts to close the window. The default behavior is to simply hide the JFrame when the user closes the window. To change the default behavior, you invoke the method setDefaultCloseOperation(int). To make the JFrame behave the same as a Frame instance, use setDefaultCloseOperation(WindowConstants.DO_NOTHING_ON_CLOSE).
For more information on content panes and other features that root panes provide, see Using Top-Level Containers in The Java Tutorial.
In a multi-screen environment, you can create a JFrame on a different screen device. See Frame for more information.
Warning: Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeans™ has been added to the