

Programmazione Orientata agli Oggetti in Linguaggio Java

Programmazione Grafica: Introduzione

versione 1.0

Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons
(vedi ultima pagina)

G. Mecca – Università della Basilicata – mecca@unibas.it



Programmazione Grafica: Introduzione >> Sommario



Sommario

- Panoramica
- Un Pò di Storia
- Caratteristiche di Swing
- Progetti di Riferimento

G. Mecca - Programmazione Orientata agli Oggetti

2



Panoramica

- Finora

- ⇒ programmazione con la console
- ⇒ ragionevole solo in un contesto didattico

- Nelle applicazioni industriali

- ⇒ due tecnologie principali per l'interfaccia utente
- ⇒ interfaccia Web (HTML) (>>)
- ⇒ interfaccia grafica



Panoramica

- Graphical User Interface (GUI)

- ⇒ interfaccia utente grafica disegnata sullo schermo ("desktop")
- ⇒ all'interno di finestre
- ⇒ con cui l'utente può interagire attraverso il mouse
- ⇒ utilizzando tipici componenti (o "controlli") grafici come menu, bottoni ecc.
- ⇒ oppure disegnando o visualizzando immagini



Panoramica

- Funzionalità rilevanti per il programmatore
 - ⇒ disegno di linee e punti sullo schermo (funzionalità di base) utilizzata per il disegno delle finestre e dei componenti oppure per le animazioni
 - ⇒ utilizzo di componenti predefiniti per costruire l'interfaccia utente delle applicazioni (menu, bottoni, icone)
 - ⇒ “catturare” e gestire le azioni dell'utente



Panoramica

- Nei linguaggi tradizionali (es: C/C+)
 - ⇒ il linguaggio non fornisce funzionalità per la programmazione grafica
 - ⇒ è possibile utilizzare librerie esterne appositamente sviluppate
- Una possibile soluzione
 - ⇒ utilizzare la libreria grafica fornita dal sistema operativo



Panoramica

○ Infatti

- ⇒ ogni sistema operativo ha una sua libreria per la programmazione grafica
- ⇒ necessaria per lo sviluppo dell'interfaccia utente del sistema (desktop)

○ Esempio

- ⇒ in Windows: Win32 API (per il C)
- ⇒ in Linux: X11 (XLib) (per il C)



Panoramica

○ Caratteristiche tipiche di queste librerie

- ⇒ sono relativamente di basso livello (quasi sempre basate sul C)
- ⇒ non sono portabili
- ⇒ sono pensate per essere efficienti
- ⇒ sono centrate attorno ad un componente fondamentale
- ⇒ il Window Manager



Panoramica

○ Window Manager

- ⇒ componente responsabile del disegno delle finestre sullo schermo
- ⇒ consente le operazioni tipiche sulle finestre
- ⇒ creazione, distruzione, riduzione a icona, ridimensionamento, trascinamento, sovrapposizione, “rinfresco”
- ⇒ gestisce i movimenti del mouse



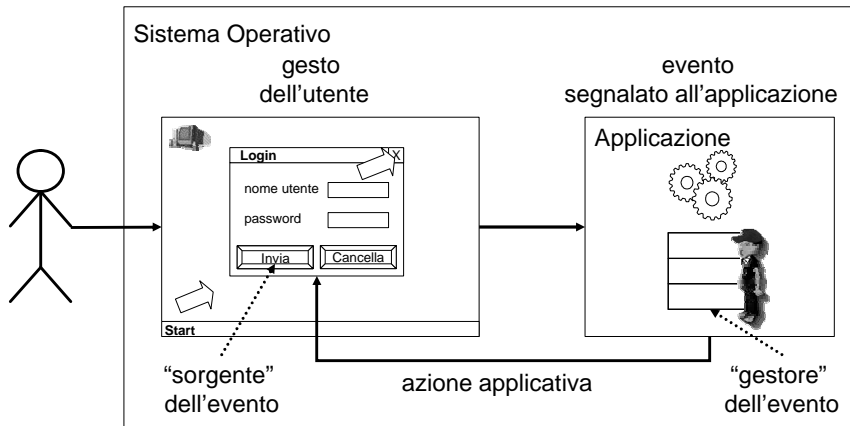
Panoramica

○ Schiacciare un bottone: processo tipico

- ⇒ la libreria di sistema disegna attraverso linee e punti sullo schermo il bottone
- ⇒ l'utente seleziona il bottone con il mouse e schiaccia il tasto sinistro
- ⇒ il Window manager intercetta il gesto dell'utente e ridisegna il bottone
- ⇒ il Window manager segnala all'applicazione l'evento che si è verificato



Panoramica



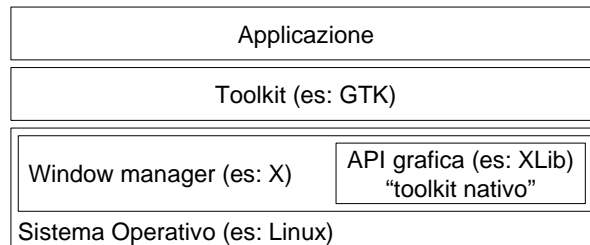
Panoramica

- Programmare con la libreria del S.O.
 - ⇒ normalmente è molto complesso
- Per semplificare la programmazione
 - ⇒ vengono sviluppati "toolkit" aggiuntivi
- Toolkit grafico
 - ⇒ libreria che consente il disegno dei componenti (menu, icone, bottoni, testo ecc.)
 - ⇒ pensata per essere più ad alto livello rispetto alla libreria di sistema (detta "toolkit nativo")

Panoramica

○ Esempi di toolkit grafici

- ⇒ Windows: Microsoft Foundation Cl. (MFC)
- ⇒ Linux: Motif, GTK, QT...



Panoramica

○ Le piattaforme moderne (Java/.NET)

- ⇒ c'è l'esigenza di svincolare lo sviluppo delle applicazioni grafiche dalle librerie di sistema
- ⇒ dal momento che il codice viene eseguito da una macchina virtuale e non dal processore
- ⇒ di conseguenza queste piattaforme hanno un proprio toolkit grafico

○ Il toolkit grafico di Java

- ⇒ Swing/JFC



Un Pò di Storia

- Programmazione grafica in Java
 - ⇒ si è evoluta attraversando varie fasi
- Il problema centrale
 - ⇒ rispettare il motto di Java: “write one, run anywhere”
 - ⇒ ovvero l’indipendenza dalla piattaforma e dalle diverse librerie di sistema



Un Pò di Storia

- Java 1.0: Abstract Windowing Toolkit
 - ⇒ il package `java.awt`: primo toolkit di Java
 - ⇒ tipico approccio Java: il package “incapsula” l’API del sistema operativo sottostante
 - ⇒ per ogni elemento grafico tipico (es: bottone) c’è una classe awt (es: `java.awt.Button`)
 - ⇒ le operazioni di disegno del bottone sono rimandate al corrispondente elemento bottone della libreria nativa detto “peer” (es: bottone Windows o bottone XLib)



Un Pò di Storia

- Vantaggio di questo approccio
 - ⇒ molto veloce perchè sfrutta su ogni sistema la libreria nativa
- Svantaggio
 - ⇒ le librerie di sistema sono troppo diverse
 - ⇒ problema di “minimo comune denominatore”
 - ⇒ la resa grafica era inaffidabile (banchi diversi)
 - ⇒ “write once, debug anywhere”



Un Pò di Storia

- Java 1.1
 - ⇒ riorganizza completamente la gestione degli eventi introducendo il package `java.awt.event`
- Nel 1996
 - ⇒ Netscape rilascia la libreria IFC (“Internet Foundation Classes”)
 - ⇒ pensata per lo sviluppo di applicazioni ricche in ambiente Web
 - ⇒ indipendente dalla piattaforma



Un Pò di Storia

- Java 2: Swing (javax.swing)
 - ⇒ una nuova libreria grafica ispirata a IFC
- Idea
 - ⇒ le finestre sono gli unici componenti affidati alla piattaforma nativa (Window manager)
 - ⇒ per ottenere una reale indipendenza dalla libreria nativa, tutti gli altri componenti sono disegnati in finestre vuote utilizzando linee e punti



Un Pò di Storia

- Vantaggi di questo approccio
 - ⇒ consente una reale portabilità delle applicazioni
 - ⇒ tutte le API native supportano il disegno di linee e punti e lo fanno correttamente
- Svantaggio di questo approccio
 - ⇒ più lento
 - ⇒ Swing stenta a decollare perchè inizialmente troppo lento



Un Pò di Storia

- Versioni successive di Swing
 - ⇒ significativi miglioramenti nella versione 1.4
 - ⇒ e successivamente nella 1.5
 - ⇒ ci riferiremo a queste versioni
- Attualmente
 - ⇒ una libreria molto completa e di ottime prestazioni (se adeguatamente utilizzata...)
 - ⇒ parte di un gruppo di package più ampio



Un Pò di Storia

- Java Foundation Classes (JFC)
 - ⇒ collezione di package Java per la programmazione grafica
 - ⇒ componenti grafici Swing
 - ⇒ gestione degli eventi AWT
 - ⇒ Java 2D (API per il disegno in 2 dimensioni)
 - ⇒ Accessibility API
 - ⇒ Drag & Drop
 - ⇒ Internazionalizzazione



Caratteristiche di Swing

- Swing
 - ⇒ una tecnologia molto complessa
- Alcune caratteristiche
 - ⇒ numero molto alto di componenti
 - ⇒ funzionalità avanzate di disegno
 - ⇒ “look & feel” variabile
 - ⇒ gestione sofisticata della disposizione (“layout”)
 - ⇒ gestione complessa degli eventi



Caratteristiche di Swing

- Un esempio
 - ⇒ SwingSet2
 - ⇒ un'applicazione dimostrativa di Swing
 - ⇒ distribuita con l'SDK di Java
 - ⇒ in demo/jfc/swingset2
 - ⇒ `java -jar swingset2.jar`

>> SwingSet2



Caratteristiche di Swing

- I package di swing
(17 in J2SE 1.5)

- ⇒ javax.swing
- ⇒ javax.swing.border
- ⇒ javax.swing.colorchooser
- ⇒ javax.swing.event
- ⇒ javax.swing.filechooser
- ⇒ javax.swing.plaf
- ⇒ javax.swing.plaf.basic
- ⇒ javax.swing.plaf.metal
- ⇒ javax.swing.plaf.multi
- ⇒ javax.swing.plaf.synth
- ⇒ javax.swing.table
- ⇒ javax.swing.text
- ⇒ javax.swing.text.html
- ⇒ javax.swing.text.html.parser
- ⇒ javax.swing.text.rtf
- ⇒ javax.swing.tree
- ⇒ javax.swing.undo



Caratteristiche di Swing

- I Package di AWT (12)
- Altri package rilevanti

- ⇒ java.awt
- ⇒ java.awt.color
- ⇒ java.awt.datatransfer
- ⇒ java.awt.dnd
- ⇒ java.awt.event
- ⇒ java.awt.font
- ⇒ java.awt.geom
- ⇒ java.awt.im
- ⇒ java.awt.im.spi
- ⇒ java.awt.image
- ⇒ java.awt.image.renderable
- ⇒ java.awt.print
- ⇒ javax.accessibility
- ⇒ javax.print
- ⇒ javax.print.attribute
- ⇒ javax.print.attribute.standard
- ⇒ javax.print.event
- ⇒ java.applet
- ⇒ javax.sound.midi
- ⇒ javax.sound.midi.spi
- ⇒ javax.sound.sampled
- ⇒ javax.sound.sampled.spi



Caratteristiche di Swing

ATTENZIONE
alla difficoltà di
Swing

- Il problema didattico connesso con Swing
 - ⇒ notevole complessità tecnologica (libreria complessa e sofisticata tecnicamente)
 - ⇒ presentare tutti i componenti è impossibile
 - ⇒ notevole complessità concettuale (eventi, thread, classi interne...)
 - ⇒ notevole complessità metodologica (gestione delle prestazioni, organizzazione del codice ...)



Caratteristiche di Swing

- Swing tutorial
 - ⇒ parte del tutorial di Java disponibile sul sito della Sun
 - ⇒ è possibile scaricarlo in formato .zip dall'URI <http://java.sun.com/docs/books/tutorial/information/download.html>
 - ⇒ un riferimento fondamentale per il programmatore
 - ⇒ in particolare la sezione "How to..."



Progetti di Riferimento

- Alcune applicazioni già viste
 - ⇒ La Morra Cinese
 - ⇒ Indovina il Numero
- La nuova versione con Swing
 - ⇒ logica applicativa sostanzialmente inalterata
 - ⇒ cambia significativamente lo strato di interfaccia e controllo

>> morraCineseSwing
>> indovinallNumeroSwing



Riassumendo

- Panoramica
- Un Pò di Storia
- Caratteristiche di Swing
- Progetti di Riferimento



Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- Questo lavoro viene concesso in uso secondo i termini della licenza "Attribution-ShareAlike" di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all'indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.