Programmazione Orientata agli Oggetti in Linguaggio Java

Programmazione Grafica: Organizzazione del Codice Parte a

versione 1.0

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

Sommario

- Lo “Stile Swing”
- Linee Guida
  - ⇨Requisiti Tipici della GUI
  - ⇨Azioni
  - ⇨Architettura di Base
- Un Altro Esempio

Lo "Stile Swing"

- Il convertitore
  - ⇨un'applicazione scritta nel tipico "stile swing"
- Lo "stile swing" o ("big blob")
  - ⇨lo stile dello Swing tutorial
  - ⇨di molti libri sull'argomento
  - ⇨e lo stile del codice sviluppato dalla maggior parte degli IDE
  - ⇨in sintesi: tutto in una classe

Lo "Stile Swing"

- Nel nostro esempio
  - ⇨la classe Principale contiene tutto il codice per la creazione dell'interfaccia grafica (GUI)
  - ⇨contiene tutti i gestori di eventi sotto forma di classi interne (normalmente anonime)
  - ⇨contiene nei metodi di gestione degli eventi la logica applicativa (conversione del voto)

Lo "Stile Swing"

- Questo stile
  - ⇨è giustificato prevalentemente dal tentativo di produrre esempi in poche linee di codice
  - ⇨funziona in applicazioni di piccole dimensioni
  - ⇨ma rende praticamente impossibile lo sviluppo di applicazioni medio-grandi
- Nel seguito
  - ⇨alcune linee guida per l'organizzazione del codice

Linee Guida

- Una caratteristica dello stile Swing
  - ⇨tutti i componenti sono proprietà della classe
- In effetti
  - ⇨ci sono controlli che vanno manipolati in metodi diversi della classe
  - ⇨JTextField: per prelevare i valori dell'utente
  - ⇨JLabel: per aggiornare i messaggi
  - ⇨il fatto che siano proprietà è giustificato

Linee Guida

- Ma...
  - ⇨tutti gli altri controlli potrebbero essere variabili locali al metodo inizializza()
  - ⇨es: pannelli
- Linea guida
  - ⇨ridurre al minimo il numero di proprietà delle classi che costruiscono l'interfaccia grafica
  - ⇨per rendere meglio gestibile il codice e ridurre l'accoppiamento tra i metodi

Linee Guida

- Un'altra pratica diffusa dello "stile swing"
  - ⇨utilizzare il frame come actionListener per tutti gli eventi sollevati dai componenti
- Pratica del tutto discutibile
  - ⇨in questo modo il frame deve in generale gestire eventi diversi
  - ⇨questo costringe ad usare un if nel metodo actionPerformed() per analizzare il tipo di evento e decidere cosa fare

Linee Guida

>> varie\convertitore\
PrincipaleGestore.java

- Esempio
  - ⇨il convertitore sviluppato con questo stile
  - ⇨è stato aggiunto un bottone “Ripulisci” che ripulisce il campo di testo
  - ⇨il frame funge da gestore per entrambi i bottoni
  - ⇨ogni volta che il metodo actionPerformed() viene invocato, deve per prima cosa distinguere il bottone sorgente dell’evento

Linee Guida

- Linea guida
  - ⇨è opportuno evitare questo approccio
  - ⇨i gestori devono essere il più specializzati possibili
  - ⇨è preferibile avere molti piccoli gestori specializzati che un singolo gestore “generico” il cui codice di gestione tende a diventare difficilmente manutenibile

Linee Guida

- Ulteriori linee guida
  - ⇨riguardano alcuni tipici requisiti delle interfacce utente
  - ⇨requisiti irrilevanti per la console
  - ⇨ma particolarmente significativi nelle applicazioni professionali con GUI
- Nel seguito
  - ⇨elenchiamo questi requisiti

Requisiti Tipici della GUI

- Requisito n. 1
  - ⇨esecuzione delle azioni
- Nelle interfacce grafiche
  - ⇨frequentemente la stessa azione applicativa deve essere eseguibile con gesti diversi
  - ⇨es: conversione: invio nel campo di testo o click sul bottone converti
  - ⇨spesso: selezione di voce da un menu oppure click su un bottone

Requisiti Tipici della GUI

- Infatti
  - ⇨all'interno di Swing ci sono varie sorgenti di ActionEvent
  - ⇨i bottoni
  - ⇨i campi di testo in corrispondenza dell'invio
  - ⇨le voci dei menu
  - ⇨il Timer allo scadere dell'intervallo prefissato

Requisiti Tipici della GUI

- Si tratta di un aspetto importante
  - ⇨per migliorare l'usabilità dell'interfaccia
- Usabilità
  - ⇨si può definire come "*La efficacia, efficienza e soddisfazione con cui specificati utenti raggiungono specificati obbiettivi in particolari ambienti (software)*" (ISO 9241)
  - ⇨l'interfaccia grafica gioca un ruolo essenziale

Requisiti Tipici della GUI

- Linea guida
  - ⇨per migliorare l'usabilità dell'applicazione, consentire di eseguire le stesse azioni in modi diversi
  - ⇨ciascun utente deve poter scegliere il modo in cui eseguire le azioni, a seconda delle proprie preferenze e delle proprie abitudini

Requisiti Tipici della GUI

- Un requisito collegato all'usabilità
  - ⇨quello dell'accessibilità
- Accessibilità
  - ⇨"mettere (le applicazioni) a disposizione di **tutti** gli individui, indipendentemente dai loro requisiti hardware e software, dall'infrastruttura di rete, dal loro linguaggio di nascita, dalla loro cultura, posizione geografica e attitudini fisiche e mentali" (Tim Berners Lee, Direttore del W3C)

Requisiti Tipici della GUI

- In particolare
  - ⇨questo vuol dire rendere l'applicazione usabile anche da utenti "diversamente abili"
- Esempio
  - ⇨non vedenti (utilizzano tecnologie di supporto come i "lettori di schermo")
  - ⇨utenti con handicap motori (utilizzano tecnologie di supporto per l'uso del mouse e dei tasti)

Requisiti Tipici della GUI

- Strumenti per migliorare l'accessibilità
  - ⇨ce ne sono vari
  - ⇨in particolare: "tooltip" e utilizzo della tastiera
- Tooltip
  - ⇨fornire suggerimenti a comparsa ("tooltip")
  - ⇨chiariscono la funzione dei componenti e rendono possibile l'uso ai non vedenti

Requisiti Tipici della GUI

- Tasti “mnemonici”
  - ⇨tasto per eseguire l’azione quando il componente corrispondente è visibile; tipicamente alt + un tasto
  - ⇨normalmente usati per navigare i menu
- Tasti “acceleratori”
  - ⇨combinazioni di tasti per eseguire direttamente il comando senza dover nemmeno visualizzare il componente

Requisiti Tipici della GUI

- Linea guida
  - ⇨prevedere gli strumenti di supporto all’accessibilità
  - ⇨suggerimenti a comparsa per i componenti che possono generare eventi che corrispondono ad azioni applicative
  - ⇨consentire l’utilizzo dell’applicazione attraverso tasti mnemonici ed acceleratori

Requisiti Tipici della GUI

- Un ulteriore requisito tipico
  - ⇨non tutte le azioni applicative devono essere sempre abilitate
- In generale
  - ⇨ci sono condizioni in cui un'azione (operazione) deve essere disponibile
  - ⇨e condizioni in cui non deve essere così (es: salvare un documento prima di averlo creato)

Requisiti Tipici della GUI

- Abilitare e disabilitare l'azione
  - ⇨l'azione deve poter essere abilitata o disabilitata a seconda delle condizioni di funzionamento
  - ⇨questo corrisponde ad abilitare o disabilitare tutti i componenti che possono innescare l'azione in questione
  - ⇨chiamando il metodo setEnabled(boolean) di java.awt.Component

Requisiti Tipici della GUI

- Linea guida
  - ⇨analizzare i possibili stati dell'interfaccia
  - ⇨abilitare esclusivamente le azioni possibili in un certo stato disabilitando le altre
  - ⇨importante per non disorientare l'utente consentendogli di eseguire azioni che sollevano eccezioni
  - ⇨o di avere effettuato un'operazione che in realtà non è stato possibile eseguire

Requisiti Tipici della GUI

- Quindi, riassumendo
  - ⇨tipicamente una stessa azione può essere innescata da componenti diversi
  - ⇨l'azione deve poter essere abilitata e disabilitata all'occorrenza abilitando o disabilitando i componenti
  - ⇨è necessario fornire il supporto all'accessibilità dell'azione, prevedendo tooltip e tasti per tutti i componenti relativi

Requisiti Tipici della GUI

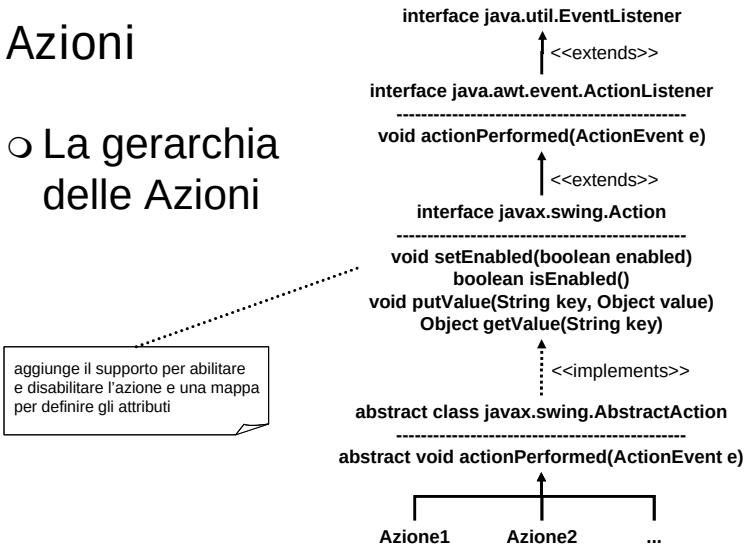
- Utilizzando ActionListener ordinari
  - ⇨dovrei registrare lo stesso listener su tutti i componenti che scatenano l'azione applicat.
  - ⇨dovrei disabilitare e riabilitare ogni volta ciascun componente separatamente per impedire/consentire l'esecuzione dell'azione
  - ⇨dovrei specificare per ciascun componente il tooltip, lo mnemonico e l'acceleratore
  - ⇨la probabilità di errori è molto alta

Azioni

- Una soluzione a questi problemi
  - ⇨le "azioni swing"
- Azione swing
  - ⇨oggetto che implementa l'interfaccia javax.swing.Action
  - ⇨consente di creare in maniera centralizzata l'azione specificandone tutte le caratteristiche e poi di usarla come listener evoluto per diversi componenti

Azioni

- La gerarchia delle Azioni



Azioni

- Gli attributi dell'azione
  - ⇨javax.swing.Action.NAME: nome
  - ⇨javax.swing.Action.SHORT_DESCRIPTION: usato per visualizzare i "tooltip"
  - ⇨javax.swing.Action.MNEMONIC: tasto mnemonico
  - ⇨javax.swing.Action.ACCELERATOR_KEY: sequenza di tasti acceleratore
  - ⇨javax.swing.Action.SMALL_ICON: icona da visualizzare nei menu e nelle barre degli strumenti

Azioni

- Utilizzo tipico dell'azione
  - ⇨nel costruttore vengono inizializzate le proprietà aggiungendo valori alla mappa
  - ⇨poi l'oggetto viene utilizzato per costruire i componenti corrispondenti
  - ⇨i componenti acquisiscono le proprietà
  - ⇨abilitando o disabilitando l'azione tutti i componenti corrispondenti vengono abilitati/disabilitati contemporaneamente

Azioni

>> varie\convertitore\finale\ AzioneConverti.java

- Un esempio
  - ⇨il convertitore
  - ⇨definisco un'azione chiamata AzioneConverti
  - ⇨specificandone le varie proprietà
  - ⇨la utilizzo per costruire il bottone di conversione
  - ⇨e la aggiungo ai gestori del campo di testo che contiene la temperatura da convertire

Azioni

- Attenzione alla terminologia
  - ⇨non bisogna fare confusione tra le due accezioni del termine “azione”
- Accezione generale
  - ⇨azione come “azione applicativa”: operazione eseguita in risposta ad un evento
- Accezione specifica
  - ⇨azione come oggetto di tipo javax.swing.Action; realizza una azione applicativa

Azioni

- Linea guida
  - ⇨in un’applicazione bisognerebbe evitare l’utilizzo di ActionListener generici
  - ⇨e viceversa utilizzare sempre azioni
  - ⇨definendone sempre le proprietà significative (nome, descrizione e almeno uno dei tasti)
  - ⇨e condividendole tra i diversi componenti che possono innescare l’azione

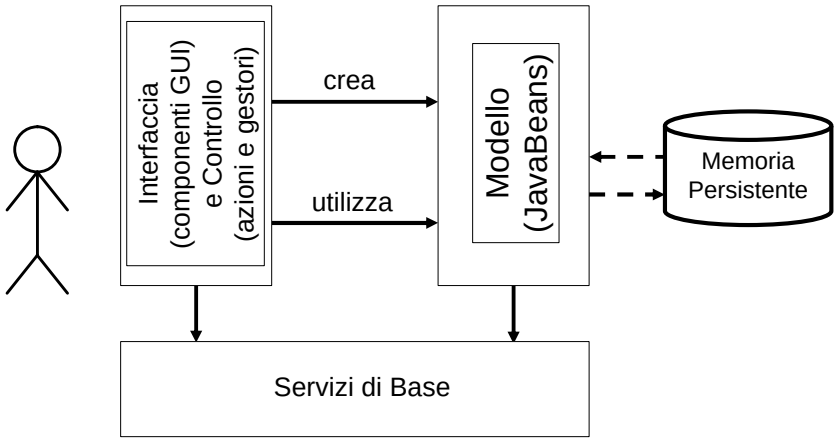
Architettura di Base

- Un ulteriore miglioramento necessario
  - ⇨nello sviluppo dell'applicazione è indispensabile rispettare perlomeno l'architettura di base presentata in precedenza
  - ⇨il codice di interfaccia e controllo in uno strato
  - ⇨il codice del modello e della persistenza in strati separati

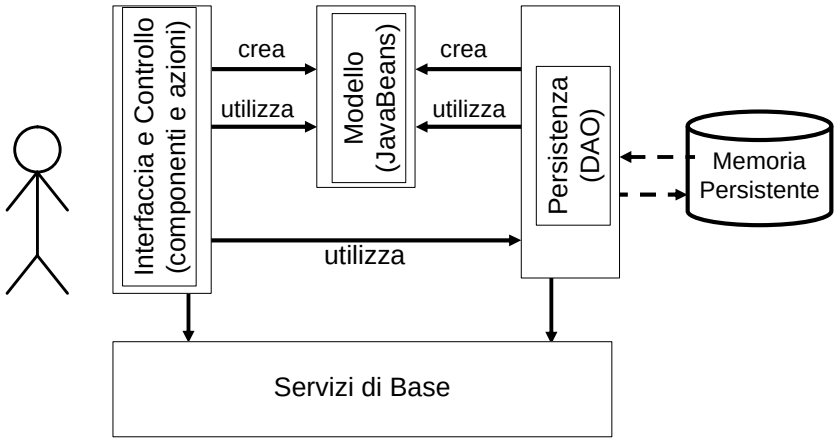
Architettura di Base

- Nel caso di Swing
  - ⇨il codice di interfaccia è il codice che gestisce i frame e i componenti
  - ⇨il codice di controllo è rappresentato dalle azioni e dagli altri gestori
  - ⇨il codice del modello resta lo stesso: JavaBeans che implementano i concetti del dominio applicativo e la relativa logica applicativa

Architettura di Base



Architettura di Base



Architettura di Base

>> it.unibas.convertitore.finale

- Nel caso del convertitore
  - ⇨la logica applicativa (molto semplice) è rappresentata dall'operazione di conversione
  - ⇨per la quale sarebbe necessario utilizzare un bean Convertitore
- it.unibas.convertitore.finale
  - ⇨un'applicazione Swing costruita secondo l'architettura di base e le linee guida discusse

Architettura di Base

- La convalida dei dati
  - ⇨approccio tipico: utilizzo un metodo che riceve il valore (o i valori) da convalidare
  - ⇨restituisce una stringa contenente un messaggio in cui sono elencati tutti i problemi riscontrati (alternativa: una ArrayList di stringhe)
  - ⇨se il messaggio di errore non è vuoto viene visualizzata una finestra di dialogo per segnalare l'errore e l'azione si interrompe

Un Altro Esempio

- A questo punto
  - ⇨abbiamo fatto un passo avanti rispetto allo “stile Swing”
  - ⇨abbiamo migliorato l’usabilità e l’accessibilità dell’interfaccia
  - ⇨abbiamo migliorato l’organizzazione del codice
  - ⇨siamo pronti per vedere un esempio più complesso

Un Altro Esempio

- it.unibas.morracineseswing
  - ⇨la Morra Cinese con interfaccia utente Swing
  - ⇨architettura di base
  - ⇨complessità medio/bassa
  - ⇨due menu (Gioco, Partita) oltre al menu ?
  - ⇨vari bottoni
  - ⇨la logica applicativa è identica al quella della versione console (3 bean, Gioco, Partita e Mano)

Un Altro Esempio

- Dimensioni delle due versioni
  - ⇨versione console: 436 linee di codice
  - ⇨versione swing: 787 linee di codice (+45% !)
- Il package it.unibas.morracineseswing.gui
  - ⇨la classe Controllo: circa 400 linee di codice
  - ⇨la classe Utilita: 52 linee di codice
  - ⇨di fatto la classe Controllo contiene il 50% del codice dell'applicazione

Un Altro Esempio

- Le azioni
  - ⇨AzioneNuovaPartita
  - ⇨AzioneInterrompiPartita
  - ⇨AzionePunteggio
  - ⇨AzioneGiocata
  - ⇨AzioneEsci
  - ⇨AzioneAbout
  - ⇨vengono abilitate e disabilitate all'occorrenza

Un Altro Esempio

- AzioneGiocata
  - ⇨viene in effetti usata un'unica classe per le tre azioni svolte dai pulsanti (gioca carta, forbici e sasso)
  - ⇨vengono creati tre oggetti diversi della stessa classe, distinti solo per il valore della proprietà corrispondente al valore da giocare
  - ⇨nel metodo actionPerformed() viene effettuata la giocata corrispondente al valore recuperato dalla mappa

Un Altro Esempio

- La strategia di gestione della GUI
  - ⇨il metodo inizializza() dispone tutti i componenti nel pannello principale (inclusi quelli inizialmente invisibili)
  - ⇨la strategia di posizionamento è assoluta
- Schermi
  - ⇨c'è di fatto un unico schermo principale
  - ⇨per il punteggio viene usata una finestra di dialogo

Un Altro Esempio

>> it.unibas.morracineseswing

- Flusso tipico di esecuzione
  - ⇨gesto dell'utente > azione corrispondente
  - ⇨il codice dell'azione interviene sulla logica applicativa (crea i bean e ne chiama i metodi)
  - ⇨abilita o disabilita altre azioni se necessario
  - ⇨al termine chiama un metodo della vista (tipicamente chiamato "schermo" nel senso di metodo di aggiornamento dello schermo) che modifica lo stato dei componenti

Un Altro Esempio

- Le immagini
  - ⇨oggetti di tipo javax.swing.ImageIcon usate per creare una JLabel
  - ⇨vengono costruite a partire dall'URL del file corrispondente
  - ⇨localizzato utilizzando il metodo getResource() di java.lang.Class
  - ⇨i file devono essere visibili sul classpath
  - ⇨es: in una cartella ./build/classes/risorse

Riassumendo

- Lo “Stile Swing”
- Linee Guida
  - ⇨Requisiti Tipici della GUI
  - ⇨Azioni
  - ⇨Architettura di Base
- Un Altro Esempio

Termini della Licenza