

Cenni su interfaccia GUI:Swing

Prof. Francesco Viglietti
Www.in4matika.altervista.org



Programmazione ad eventi

Ricordiamo brevemente il funzionamento della programmazione ad eventi. In questa programmazione è necessario specificare gli oggetti (sorgenti) con cui si può avere un'interazione (evento), e, cioè un insieme di classi che, in modo concorrente, controllano se le sorgenti hanno riscontrato l'evento (ascoltatori).

Ogni componente grafico è una potenziale sorgente, mentre gli ascoltatori sono oggetti che implementano specifiche interfacce, dette appunto listener.

A partire da Java 2 il package grafico Swing (`javax.swing`) scritto in Java e realmente indipendente dalla piattaforma, esso definisce una gerarchia di classi che forniscono ogni tipo di componente grafico (finestre, pannelli, frame, bottoni, aree di testo, checkbox, liste a discesa, etc). Grazie alla presenza di Swing è possibile operare con la programmazione event-driven. Grazie a questo tipo di programmazione non è necessario utilizzare algoritmi stile input/elaborazione/output... ma è sufficiente operare sulle reazione agli eventi che l'utente, in modo interattivo, genera sui componenti grafici. Questa programmazione è basata sui concetti di evento e di ascoltatore degli eventi

eventi

Quando si crea l'interfaccia di un software si definisce un insieme di oggetti con i quali l'utente potrà interagire. Sarà opportuno:

- predisporre la rappresentazione grafica inserendo uno o più componenti grafiche (sorgenti) nell'interfaccia;
- decidere quali eventi (azioni compiute dall'utente) gestire e predisporre le conseguenti reazioni del sistema;
- definire un ascoltatore di uno o più eventi fra quelli individuati;
- associare l'ascoltatore alla sorgente.

E' importante notare che ogni componente può essere sorgente di diversi eventi: per esempio, un pulsante prevede non solo la pressione, ma anche altre azioni legate al mouse (singolo o doppio clic, passaggio del cursore, pulsante destro ecc.), alla modifica di stato delle sue proprietà ecc.

eventi

In qualunque momento, l'utente può cliccare su una componente o premere un tasto della tastiera, e la GUI

delle applicazioni deve intercettare tali eventi e gestirli. Oltre agli eventi generati dall'utente esistono anche eventi time-driven, generati da thread eseguiti in parallelo (o...pseudo-parallelo), che non dipendono dall'utente ma che implicano un aggiornamento della GUI. Tutti queste considerazioni stanno alla base della programmazione orientata agli eventi.

Java prevede particolari oggetti, i listener (ascoltatori); questi vengono 'avvisati' dal sistema quando

si verifica un evento che li riguarda e che hanno il compito di gestire tali eventi: l'implementazione del metodo o dei metodi di gestione spetta al programmatore. Ad esempio: il listener degli eventi da tastiera verrà invocato quando l'utente premerà, appunto, un tasto della tastiera.

I listener sono interfacce da implementare, il che significa che spesso siamo 'obbligati' a riscriverne alcuni metodi.

Eventi in netbeans

The image displays the NetBeans IDE interface with two main panels: Source and Design. The Source panel on the left shows the Java code for `ProvaUI` class, which extends `JFrame`. The code includes a constructor `ProvaUI()` that calls `initComponents()` and sets the visibility of `lblMondo` to false. A private method `btnCliccaActionPerformed` is also present, which sets the visibility of `lblMondo` to true when the button is clicked. The `main` method is also visible. The Design panel on the right shows a visual representation of the GUI with a window titled "Ciao mondo!" containing a button labeled "Clicca qui". The Palette on the right shows the "Swing Controls" section with "1: classe JButton" highlighted. The "Events" table for the `btnClicca [JButton]` component shows "3: evento" for the `actionPerformed` event, which is linked to the `btnCliccaActionPerformed` method. Red arrows and boxes highlight these key elements and their relationships.

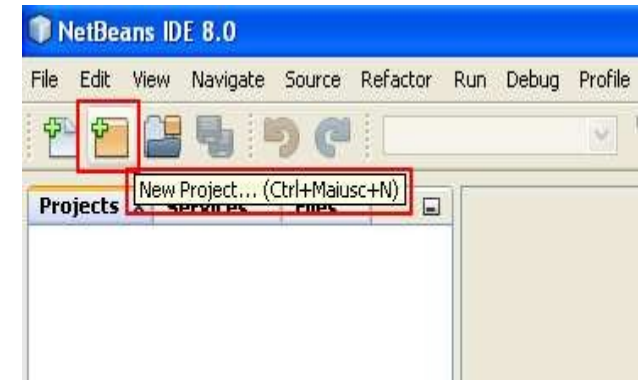
Nelle **figure** vediamo il costruttore che richiama il **metodo `initComponents()`**. Questo metodo viene generato automaticamente da **NetBeans** e contiene tutto il codice dell'interfaccia grafica, ascoltatori compresi, costruita nell'area disegno. Il **codice non può essere modificato** (vedi *Tips and tricks - Guarded Blocks*) direttamente dal programmatore ma solo attraverso l'area delle proprietà dei singoli oggetti.

Creare GUI con swing

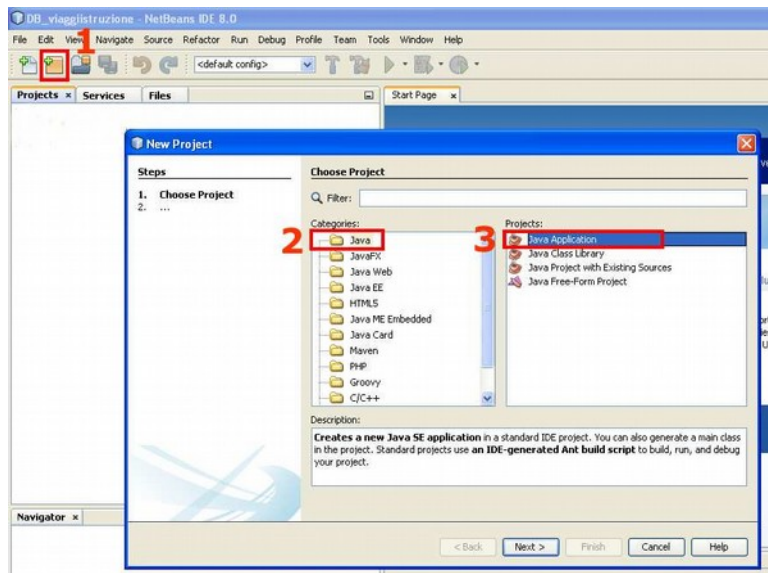
Perché tutto lo sviluppo Java nell'IDE si svolge nell'ambito di progetti, abbiamo prima bisogno di creare un New Project in cui memorizzare le fonti e file di progetto.

In questo esempio costruiremo una semplice applicazione memorizzata interamente in un unico progetto.

Per creare l'applicazione ContactEditor con New Project:
Fare clic sull'icona New Project nella barra degli strumenti IDE.



Nel riquadro Categories, selezionare Java nel pannello Choose Projects, scegliere Java Application. Fare clic su Next.



Inserisci ContactEditor nel campo Project Name e specificare il percorso del progetto.

Lasciare deselezionata la casella della cartella per l'uso dell'archiviazione di librerie.

Assicurarsi che la casella di controllo MAIN project sia selezionata e cancellare il campo Create Main Class.

Fare clic su Finish

Creare GUI con swing

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

1 Project Name: ContactEditor

2 Project Location: E:\Documents and Settings\All Users\Documents\Dropbox\JAVA_appunti_in_costruzione\ContactEditor Browse...

Project Folder: s and Settings\All Users\Documents\Dropbox\JAVA_appunti_in_costruzione\ContactEditor\ContactEditor

3 Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

4 Create Main Class

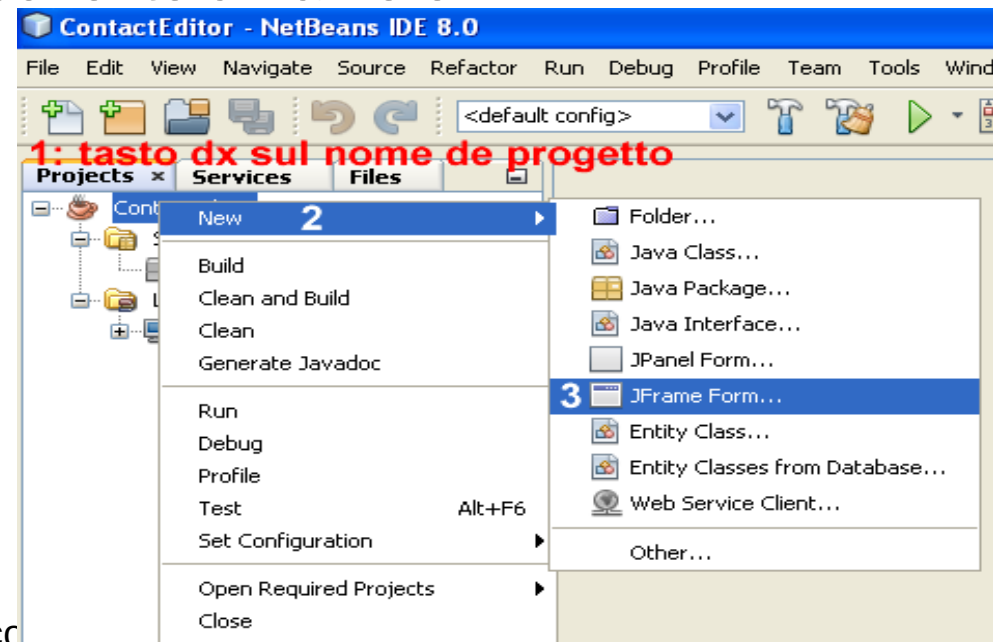
< Back Next > Finish Cancel Help

Creare un Container JFrame

Dopo aver creato la nuova applicazione, la cartella di origine dei pacchetti nella Finestra Project conterrà un nodo vuoto <default package>. Per procedere con la costruzione dell'interfaccia, è necessario creare un contenitore di Java all'interno del quale metteremo gli altri componenti GUI richiesti.

Per creare il contenitore si utilizza il componente JFrame e lo si posiziona in un nuovo package.

- Per aggiungere un contenitore JFrame: nella Finestra Projects, fare clic su ContactEditor e scegliere il modulo New> JFrame Form.



Le finestre del GUI builder

