

ESERCITAZIONE TIGA: componenti Swing, applet

Il settore finanze di un ente pubblico territoriale decide di integrare il proprio portale web con un servizio di “calcolatrice” che, visualizzata in un frame laterale accanto ai moduli per il calcolo delle imposte, permetta ai contribuenti di eseguire semplici operazioni di calcolo.

In particolare, il committente sottolinea la possibilità che in un futuro prossimo vengano aggiunte ulteriori operazioni.

1) Sviluppare¹ una applet dotata di interfaccia grafica come in figura, che realizza il servizio di calcolatrice a notazione postfissa.

In alto a destra viene visualizzato – come etichetta – il valore attuale dell’ **accumulatore**; in alto a sinistra – in un campo di testo modificabile – il valore dell’ **operando**.

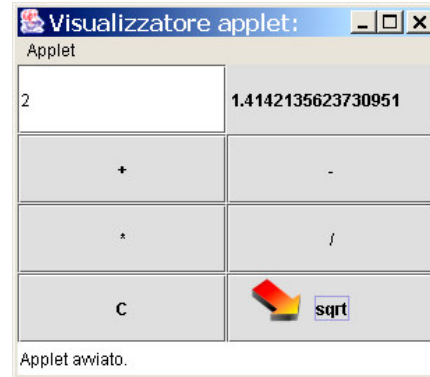


Fig.1 – Applet calcolatrice

Le operazioni binarie disponibili sono la somma (+), la sottrazione (-), la moltiplicazione (*) e la divisione (/); quelle unarie, la radice quadrata (**sqrt**) e l’azzeramento (**C**). Tutte le operazioni lasciano inalterato l’operando e collocano il risultato nell’accumulatore. Nel caso di operazione unaria $U(\bullet)$, il nuovo valore dell’accumulatore diventa:

$$\text{accumulatore} = U(\text{accumulatore}).$$

Nel caso di operazione binaria $(\bullet)B(\bullet)$, il nuovo valore dell’ accumulatore diventa:

$$\text{accumulatore} = (\text{accumulatore})B(\text{operando}).$$

2) Eseguire un intervento di *manutenzione perfetta*, consistente nell’aggiunta di una nuova operazione, ad esempio l’operazione unaria di negazione (**neg**). Quante istruzioni sono state modificate o aggiunte in totale ?

3) Inserire dei valori inconsistenti (stringhe alfanumeriche) come valore dell’operando, ed eseguire una operazione binaria. Quali eccezioni vengono generate? Eseguire delle operazioni matematicamente indeterminate, quali la divisione per zero. Cosa accade? Elencare le principali anomalie. L’operazione di azzeramento si potrebbe matematicamente ottenere sommando all’accumulatore il suo opposto. Esistono degli stati dai quali non è possibile uscire senza l’azzeramento?

SUGGERIMENTI:

Si consiglia di gestire in modo unitario gli eventi originati dai sei tasti, definendo un singolo ascoltatore per gli eventi di azione. All’interno del metodo `actionPerformed` vengono invocati due servizi, relativi all’applicazione. Il primo servizio serve ad impostare un opportuno *stato* a seconda del bottone pigiato (riferibile come argomento attraverso `getSource()`), mentre il secondo servizio esegue una corrispondente operazione della calcolatrice, aggiornando contestualmente gli oggetti grafici dell’interfaccia, a seconda dello *stato* attuale.

Si osservi che l’editazione di un campo di testo *non* necessita di alcuna gestione di eventi da parte del programmatore, in quanto inclusa nel comportamento predefinito dell’oggetto.

La stringa contenuta nel campo di testo può essere prelevata con il metodo `String getText()`.

La stringa contenuta nella etichetta può essere modificata con il metodo `setText(String s)`.

¹ D’ora in poi i adoperi l’ambiente *netBeans IDE*® per l’editazione e la compilazione del codice. Per l’esecuzione delle applet si usi l’applicativo *appletviewer*® oppure un web browser. Nel caso di *applet viewer*®, eventuali eccezioni vengono stampate sulla finestra dei comandi da cui l’applicazione è stata avviata. Nel caso del web browser le eccezioni possono essere visualizzate nella *Java console*. Tale finestra appare al caricamento di una applet se nel *Java*® *plug-in Control Panel* (i) è stato impostato *Basic* → *Java Console* → *Show Console* ed (ii) il pannello *Browser* → *Settings* contiene il proprio web browser opportunamente selezionato.