



Problema 12

21 Aprile 2021

Parte I

In relazione al *rompicapo delle N regine*, modifica la realizzazione della classe `Board` sviluppata nella precedente esercitazione in laboratorio (problema 11) utilizzando istanze di `SList<Integer>`, in sostituzione di `IntSList`, per rappresentare le liste delle righe, delle colonne, delle diagonali (orientate secondo ciascuna delle due direzioni) che sono minacciate da una regina collocata sulla scacchiera.

Sperimenta la nuova versione della classe `Board` attraverso il programma `Queens.java`, discusso a lezione, che codifica gli algoritmi per determinare il numero di soluzioni e la lista delle soluzioni — di tipo `SList<Board>`. Verifica, in particolare, il numero delle soluzioni che si ottengono al variare della dimensione della scacchiera.

Parte II

Il programma può essere integrato con uno strumento per visualizzare le configurazioni della scacchiera attraverso una GUI (Graphical User Interface) specifica. Questo strumento è confezionato nel package *queens* (jar-file `queens.jar` — associato a questo problema), che rende disponibile la classe `ChessboardView` con il seguente semplice protocollo:

<code>ChessboardView gui = new ChessboardView(n)</code>	crea la rappresentazione grafica di una scacchiera $n \times n$
<code>gui.setQueens(s)</code>	visualizza la configurazione (disposizione di regine) codificata dalla stringa <i>s</i> attraverso coppie lettera-cifra separate da spazi bianchi

Quando viene invocato il metodo `setQueens`, una successiva invocazione dello stesso metodo resta in sospeso finché l'utente effettua un click con il mouse sull'immagine della scacchiera. Questa funzionalità ha lo scopo di consentire all'utente di vedere una configurazione per un tempo sufficiente prima di passare ad una diversa visualizzazione attraverso la stessa GUI.

Il parametro del costruttore di `ChessboardView` deve essere un numero intero compreso fra 1 e 15, che rappresenta il numero di righe e di colonne della scacchiera da visualizzare.

L'argomento del metodo `setQueens` è una stringa composta da coppie lettera-cifra, dove una lettera minuscola è seguita immediatamente da una cifra (se necessario estendendo le cifre decimali con le prime lettere maiuscole, adottate nella notazione esadecimale). Le coppie devono essere separate da uno o più spazi bianchi (spazi che possono eventualmente anche precedere e/o seguire le coppie). Non devono in ogni caso comparire altri caratteri o combinazioni di caratteri diverse da quelle descritte. Le coppie lettera-cifra rappresentano le coordinate di colonna e riga secondo le convenzioni in uso da parte dei giocatori di scacchi.

Una classe che utilizza questo package deve premettere l'istruzione: `import queens.*;`

I comandi per la compilazione e per l'esecuzione da terminale devono fare riferimento esplicito al jar-file:

```
javac -classpath "queens.jar:." MainClass.java
java -classpath "queens.jar:." MainClass
```

In *DrJava* o *BlueJ* è invece necessario indicare il jar-file nell'ambito dei *classpath* (delle librerie) attraverso le preferenze dell'applicazione (sezione "Resource Locations" per *DrJava*, "Libraries" per *BlueJ*).

Applicando il package *queens*, scrivi un metodo statico che, data una lista di tipo `SList<Board>` relativa a scacchiere tutte della stessa dimensione, visualizzi in successione l'immagine di ciascuna scacchiera che fa parte della lista sulla stessa GUI. Utilizza infine questo programma per visualizzare tutte le soluzioni del rompicapo delle N regine per una data dimensione della scacchiera.

Nella pagina seguente è illustrato un esempio di visualizzazione relativa a una scacchiera 8×8 , visualizzazione che si può ottenere eseguendo le istruzioni riportate in testa all'immagine.

```
queens.ChessboardView view = new queens.ChessboardView( 8 );  
view.setQueens( " b1 e2 g3 d4 a5 h6 f7 c8 " );
```

