

LABORATORIO DI PROGRAMMAZIONE 2008–2009
TURNO 3 (SERALE)
PROVA IN ITINERE
TEMA DEL PROGETTO
12.I.2009

VINCENZO MARRA

AVVERTENZA. *In tutto il progetto, usate solo le classi delle librerie standard di Java. Non è ammesso l'uso del package `prog` allegato al libro di testo del corso.*

1. PROGETTO SUDOKU

Da *Wikipedia*, alla voce *Sudoku*:

Sudoku è un gioco di logica nel quale al giocatore o solutore viene proposta una *griglia* di 9×9 celle, ciascuna delle quali può contenere un numero da 1 a 9, oppure essere vuota; la griglia è suddivisa in 9 righe orizzontali, nove colonne verticali e, da bordi in neretto, in 9 “sottogriglie”, chiamate *regioni*, di 3×3 celle contigue. Le griglie proposte al giocatore hanno da 20 a 35 celle contenenti un numero. Scopo del gioco è quello di riempire le caselle bianche con numeri da 1 a 9, in modo tale che in ogni riga, colonna e regione siano presenti tutte le cifre da 1 a 9 e, pertanto, senza ripetizioni.

Si veda la Figura 1 per un esempio di griglia di gioco.

Scopo di questo progetto è lo sviluppo di un programma scritto in Java che permetta all'utente di giocare a Sudoku, secondo le modalità descritte qui di seguito.

1.1. **Il menu principale.** All'avvio, il programma presenta all'utente il *menu principale* dell'applicazione:

- 1) Inizia partita con griglia corrente.
- 2) Scegli griglia.
- 3) Estrai griglia a caso.

5	3		7					
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

FIGURA 1. Una griglia del Sudoku.

4) Esci.

Nel caso 1, se non è ancora stata impostata alcuna griglia corrente, il programma visualizza un messaggio d'errore appropriato, e torna al menu principale. (Ciò avviene, in particolare, immediatamente dopo ciascun avvio del programma). Se invece è già stata impostata una griglia di gioco corrente (tramite le scelte 2 o 3, cfr. sotto), il programma avvia il gioco passando al menu di gioco, per il quale cfr. §1.2.

Nel caso 2, il programma permette all'utente di scegliere la griglia con cui giocare scegliendola da un file, denominato `db_sudoku`, che memorizza un certo numero di griglie di gioco. (Per altri dettagli circa questo file, si veda il §1.3.) Più in dettaglio, se il file non esiste ancora, il programma ne crea uno vuoto, informa l'utente del fatto che non vi sono griglie memorizzate nel file, e torna al menu principale. Se il file esiste ma non contiene alcuna griglia, il programma informa l'utente del fatto che non vi sono griglie memorizzate nel file, e torna al menu principale. Altrimenti, il programma chiede all'utente il numero¹ della griglia da scegliere. Se il numero inserito non corrisponde a una griglia esistente, il programma chiede all'utente di reinserire il numero della griglia desiderata; altrimenti, carica la griglia dal file, la imposta come griglia corrente, e torna al menu principale.

Nel caso 3, se il file non esiste ancora, il programma ne crea uno vuoto, informa l'utente del fatto che non vi sono griglie memorizzate nel file, e torna al menu principale. Se il file esiste ma non contiene alcuna griglia, il programma informa l'utente del fatto che non vi sono griglie memorizzate nel file, e torna al menu principale. Altrimenti, il programma carica e imposta come griglia corrente una griglia di gioco estratta a caso (tramite generazione di un appropriato numero pseudo-casuale) dal file `db_sudoku`, e poi torna al menu principale.

Nel caso 4, il programma termina.

1.2. Il menu di gioco. Se dal menu principale l'utente sceglie 1, e vi è una griglia corrente impostata, il programma presenta all'utente una visualizzazione (in formato testo, usando la console) della griglia di gioco corrente, seguita dal *menu di gioco* dell'applicazione:

- 1) Mossa.
- 2) Suggerimento.
- 3) Termina partita.
- 4) Sospendi partita.

Nel caso 1, il programma chiede all'utente di inserire le coordinate (riga e colonna) della casella su cui l'utente intende scrivere un numero. Se la riga e la colonna non sono comprese fra 1 e 9, il programma chiede all'utente il reinserimento dei dati. Se la riga e la colonna soddisfano queste specifiche, il programma chiede all'utente di digitare un numero fra 1 e 9 da inserire nella casella specificata. Se l'utente non inserisce il dato nel formato richiesto, il programma ne chiede il reinserimento. Acquisite le coordinate della casella da impostare, e il numero da scrivere nella casella, il programma invoca un metodo appropriato di una istanza della classe

¹Cioè, la posizione della griglia nel file: 3 vuol dire la terza griglia memorizzata, 4 la quarta, eccetera.

Griglia (che andrà opportunamente sviluppata), passandogli i dati acquisiti. Il metodo dovrà:

- apportare alla griglia di gioco corrente le modifiche richieste, nel caso in cui la casella non sia già occupata da un numero, e il numero inserito dall'utente dia luogo a una nuova configurazione ammessa dal gioco; oppure
- sollevare un'eccezione controllata di tipo `GrigliaException`, altrimenti.

Nel caso in cui il metodo sollevi l'eccezione, il programma intercetta l'eccezione, fornisce la spiegazione appropriata all'utente, visualizza la griglia corrente (che non è stata modificata, a causa dell'eccezione sollevata) e torna al menu di gioco.

Nel caso 2, il programma sceglie prima a caso una casella bianca dalla griglia di gioco corrente. Per far ciò, il programma genera numeri pseudo casuali compresi fra 1 e 9 per la riga e la colonna della casella da scegliere, fino a quando la casella selezionata non risulti bianca. A quel punto, il programma aggiorna la griglia corrente, riempiendo la casella selezionata con l'unico numero ammesso per quella casella. Per poter reperire facilmente questo numero, sarà opportuno memorizzare nel file `db_sudoku` (cfr. §1.3), oltre alle configurazioni iniziali di gioco, anche le corrispondenti soluzioni finali.² Per finire, il programma visualizza la nuova griglia di gioco corrente, e torna al menu di gioco. Questo a meno che la casella riempita come suggerimento non fosse l'ultima bianca, nel qual caso, il programma mostra la casella corrente (completa), informa l'utente che il gioco è terminato, e torna al menu principale. (Si noti che dopo il rientro al menu principale la griglia corrente è nuovamente quella precedente all'inizio della partita appena conclusa, come nel caso 3 descritto qui di seguito).

Nel caso 3, il programma torna al menu principale, ripristinando la situazione precedente all'inizio della partita corrente.

Nel caso 4, il programma termina, ma imposta come griglia corrente lo stato attuale della partita. Per esempio, se la configurazione iniziale con cui era iniziata questa partita prevedeva 27 numeri già inseriti, e se l'utente ha inserito correttamente altri 3 numeri, allora la nuova griglia corrente avrà 30 numeri già specificati. (Si noti invece che il contenuto del file `db_sudoku` rimane invariato.)

1.3. Il file `db_sudoku`. Il file memorizza alcune griglie di gioco. La scelta del formato esatto con cui memorizzare i dati è parte dell'esame. Ecco qualche suggerimento.

- È opportuno che il file tenga traccia del numero complessivo di griglie di gioco memorizzate nel file. Infatti, questa informazione è usata per implementare, ad esempio, il punto 3 del menu principale.
- Per ogni griglia di gioco memorizzata, il file dovrà in effetti memorizzare due informazioni distinte: la prima è la configurazione iniziale della griglia di gioco; la seconda è la soluzione completa della griglia. Infatti, la prima informazione è ovviamente necessaria. La seconda serve per implementare, ad esempio, il punto 2 del menu di gioco, quando il computer deve fornire un suggerimento all'utente.

²Si noti che di solito nel gioco del Sudoku la configurazione iniziale è scelta in modo tale da determinare una e una sola soluzione possibile; assumete sempre questo fatto.

- Per costruire un file `db_sudoku` con cui fare dei test (e da consegnare, cfr. §2), potete copiare alcune delle griglie e delle soluzioni pubblicate da quasi tutti i quotidiani italiani, anche in formato elettronico. Si veda, per esempio, <http://sudoku.repubblica.it/>.

2. MATERIALE DA CONSEGNARE

- (1) Il sorgente della classe `Griglia`, che implementa una griglia di gioco.
- (2) Il sorgente della classe `GrigliaException`, che implementa l'eccezione omonima.
- (3) Il sorgente della classe `Sudoku`, che contiene il metodo `main` per avviare l'applicazione.
- (4) Il sorgente di tutte le altre classi ausiliarie eventualmente implementate.
- (5) Un file `db_sudoku` contenente qualche griglia d'esempio.

Nota. Il codice sorgente deve essere opportunamente commentato.

3. MODALITÀ DI CONSEGNA

Create un **unico** archivio (per esempio un file `.zip`) contenente il materiale da consegnare, il cui nome segua il formato

`cognome_matricola .`

Poi eseguite l'upload di questo file tramite il sito:

<http://upload.dico.unimi.it/> ,

autenticandovi tramite login e password della vostra casella di posta d'ateneo, o tramite login e password per accedere al SiLab. Scegliete poi il corso di Laboratorio di Programmazione, Turno 3. Infine, scegliete la sessione

Progetto in Itinere 2008-2009 (Sudoku)

ed eseguite l'upload del file.

La sessione di consegna aprirà il 23 gennaio 2009, e chiuderà il 26 gennaio 2009 alla mezzanotte.

(V. Marra) DIPARTIMENTO DI INFORMATICA E COMUNICAZIONE, UNIVERSITÀ DEGLI STUDI DI MILANO, VIA COMELICO, 39-41, I-20135 MILAN, ITALY

E-mail address: `marra@dico.unimi.it`