



## Problema 15

23 Maggio 2016

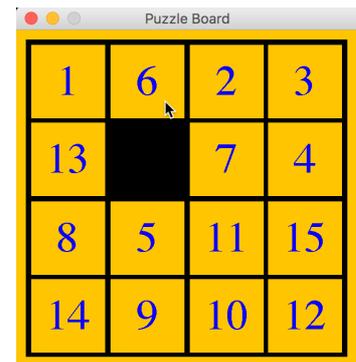
### Descrizione

Il “rompicapo del 15” è un gioco molto conosciuto. Si tratta di una tavoletta in cui sono collocati 15 tasselli numerati, secondo una matrice  $4 \times 4$ , in modo che resti un quadratino vuoto (o *lacuna*: vedi figura). A partire da una disposizione disordinata, l’obiettivo è di riordinare i tasselli attraverso una sequenza di mosse in ciascuna delle quali uno dei tasselli adiacenti alla lacuna viene spostato su di essa.

In questo esercizio di programmazione orientata agli oggetti il compito è di modellare (in Java) la tavoletta del rompicapo.

Innanzitutto, definisci un protocollo appropriato, che comprenda:

- Un costruttore per istanziare un modello della tavoletta;
- Un metodo per verificare se i tasselli sono ordinati;
- Un metodo per verificare se un dato tassello può essere spostato;
- Un metodo per visualizzare (stampare) la configurazione del rompicapo;
- Un metodo per spostare un dato tassello.



Ai fini delle operazioni indicate sopra, un tassello viene identificato dal numero corrispondente, riportato su di esso.

Quindi, caratterizza la rappresentazione interna della (generica) configurazione di una tavoletta, introducendo le variabili di istanza che ti sembrano adatte a modellare la situazione.

Completa quindi la definizione della classe, in particolare codificando il costruttore e i metodi del protocollo.

Una volta realizzato, il modello può essere integrato con uno strumento che consente un’interazione più immediata. Il package *puzzleboard* (jar-file *puzzleboard.jar*) mette a disposizione una classe `PuzzleBoard`, una piccola GUI (Graphical User Interface) come quella raffigurata sopra, utilizzabile attraverso il seguente protocollo:

```
PuzzleBoard gui = new PuzzleBoard( n )      crea la rappresentazione grafica di un tavoletta  $n \times n$ 
gui.setNumber( i, j, k )                   colloca in posizione  $\langle i, j \rangle$  un tassello con numero  $k$ 
gui.clear( i, j )                          cancella il tassello in posizione  $\langle i, j \rangle$ 
                                           (in corrispondenza alla quale si crea una lacuna)
gui.display()                              aggiorna la visualizzazione
                                           (i due metodi precedenti non lo fanno automaticamente)
int k = gui.get()                           restituisce il numero associato al tassello su cui si è
                                           agito attraverso un click del mouse
                                           (l’esecuzione di get resta sospesa fino a quando l’utente
                                           attua una selezione valida con il mouse)
```

Gli indici che rappresentano le coordinate dei tasselli sono definiti nell’intervallo  $[1, n]$ .

Una classe che utilizza questo package deve premettere l’istruzione: `import puzzleboard.PuzzleBoard;`

I comandi per la compilazione e per l’esecuzione sono quindi:

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

In *DrJava* è invece necessario indicare il jar-file nell’ambito dei *classpath*, attraverso le preferenze dell’applicazione.

L’integrazione delle funzionalità del package *puzzleboard* ti consente di sperimentare il modello della tavoletta giocando con il rompicapo. In ogni caso, non ti viene richiesto di scrivere un programma che riesca a risolvere il puzzle, ma semplicemente di scrivere un programma che consenta all’utente di giocare agendo con il mouse.