

## Esame di Programmazione II, 24 giugno 2019

Si crei un progetto Eclipse e, nella directory dei sorgenti, si crei il package `it.univr.tictactoe`. Si copi al suo interno le classi del compito. Se si realizzano nuove classi, le si crei dentro il package `it.univr.tictactoe`. Non si modifichi le dichiarazioni dei metodi. Si possono definire altri campi, metodi, costruttori e classi, ma devono essere **private**. La consegna fornita compila. Anche la soluzione che verrà consegnata dovrà compilare, altrimenti non verrà corretta.

Il gioco del tris consiste in una tabella 3x3 in cui due giocatori, alternativamente, piazzano i simboli croce e cerchio. Vince il giocatore che per primo piazza tre simboli uguali nella stessa riga, colonna o diagonale, come ad esempio la croce in questa immagine:

```
  |  |  |
--|  |  |
  |  |  |
--|  |  |
  |  |  |
  |  |  |
```

Il gioco può anche finire in parità se tutte le caselle sono piene ma nessun giocatore ha vinto, come nell'immagine seguente:

```
  |  |  |
--|  |  |
  |  |  |
--|  |  |
  |  |  |
  |  |  |
```

**Esercizio 1 [14 punti]** Si completi la classe `TicTacToe.java`, che implementa un gioco del tris, dove indicato con `TODO`. Tale codice definisce già le variabili d'istanza `crossPlayer` e `circlePlayer`, che sono i due giocatori che si alternano al gioco. La loro classe `Player.java` è già scritta e completa. Si faccia attenzione a lanciare le eccezioni come indicato nei commenti dei metodi. Si noti che la tabella del tris è stata implementata con un array monodimensionale di 9 caselle, che vanno interpretate come se fossero distribuite secondo la seguente figura:

0	1	2
3	4	5
6	7	8

Il metodo `toString()` dovrà descrivere il gioco con una stringa del tipo:

```
X|  |  |
----
X|O|O
----
 |  |X
```

Si noti che la classe `TicTacToe` è astratta perché ha il metodo `isGameOver()` astratto. Tale metodo verrà implementato nelle sottoclassi.

**Esercizio 2 [8 punti]** Si scriva la sottoclasse concreta `SimpleTicTacToe.java` di `TicTacToe.java` il cui metodo `isGameOver()` specifica che un gioco è finito quando non ci sono più caselle vuote. Si scriva la sottoclasse `RowsTicTacToe.java` di `SimpleTicTacToe.java` il cui metodo `isGameOver()` specifica che un gioco è finito quando non ci sono più caselle vuote oppure quando uno dei giocatori ha piazzato tre simboli uguali su una stessa riga o su una stessa colonna. Si scriva la sottoclasse `FullTicTacToe.java` di `RowsTicTacToe.java` il cui metodo `isGameOver()` specifica che un gioco è finito quando non ci sono più caselle vuote oppure quando uno dei giocatori ha piazzato tre simboli uguali su una stessa riga o su una stessa colonna o su una stessa diagonale.

**Esercizio 3 [10 punti]** Si completi la classe `Main.java` in modo da creare un `FullTicTacToe` e fare giocare alternativamente Alessandra e poi Giovanni, leggendo ogni volta le coordinate  $x$  e  $y$  da tastiera, stampando il gioco dopo ogni mossa, finché il gioco non risulta finito. Se una mossa genera un'eccezione, si stampi sul video il messaggio dell'eccezione e si torni a richiedere una mossa corretta.

*Se tutto è corretto, un'esecuzione del `Main` dovrebbe rassomigliare a quanto riportato nel file di testo `risultato_main.txt`.*