

Linguaggi di Programmazione per la Sicurezza

Esame del 15 settembre 2011 - (Prova di laboratorio)

1 Forza quattro

Forza quattro è un gioco da tavolo in cui due giocatori (*rosso* e *nero*) devono riuscire a creare una sequenza di 4 delle loro pedine (in verticale o in orizzontale, non in diagonale) su una scacchiera posta in verticale.

Poiché la scacchiera è posta in verticale, una pedina inserita in una certa colonna va sempre a occupare la posizione libera situata più in basso nella colonna stessa.

X			
X			
X		O	
X	O	O	

Table 1: Esempio di scacchiera in cui ha vinto il rosso (X).

1.1 Classe Java [pt. 2]

Scrivere una classe *ForzaQuattro* che rappresenti una scacchiera 4×4 di interi; un cella vale:

- 0 se non contiene alcuna pedina;
- 1 se contiene una pedina rossa;
- 2 se contiene una pedina nera.

All'inizio la scacchiera è vuota.

Metodo addPedina La classe deve disporre di un metodo *addPedina(int colore, int colonna)* la cui implementazione viene mostrata in Figura 1.

```
public void addPedina(int color, int column) {
    if (color > 0 && color < 3 && column >= 0 && column < 4) {
        if (freeRow[column] < 4) {
            board[freeRow[column]][column] = color;
            freeRow[column]++;
        }
    }
}
```

Figure 1: Metodo *addPedina*

Il metodo *addPedina* controlla che il parametro *color* indichi uno dei due colori (1 o 2) e che *column* indichi una colonna della scacchiera: se i parametri sono corretti, viene eseguito l'inserimento solo se c'è spazio nella colonna selezionata.

Metodo won La classe deve disporre di un metodo booleano *won(int color)* che dica se il colore *color* ha vinto.

2 Coverage

In JUnit, per il metodo *addPedina*, scrivere:

- un test set che soddisfi la copertura delle decisioni, ma che non soddisfi anche la copertura delle condizioni; [pt. 2]
- un test set che soddisfi la copertura delle condizioni. [pt. 2]

Non è necessario eseguire alcun controllo (nessuna *assert*) all'interno dei test case.

3 JML

Scrivere in JML la seguente postcondizione al costruttore:

- tutte le celle della scacchiera valgono 0. [pt. 1]

Scrivere in JML le seguenti precondizioni al metodo *addPedina(int color, int column)*:

- il parametro *color* indica un colore valido; [pt. 0.25]
- il parametro *column* indica una colonna valida; [pt. 0.25]

Scrivere in JML le seguenti postcondizioni al metodo *addPedina(int color, int column)*:

- il contenuto delle colonne diverse da *column* dopo l'esecuzione del metodo è uguale al contenuto prima dell'esecuzione del metodo; [pt. 1.25]
- la differenza in valore assoluto tra il numero di celle rosse e il numero di celle nere è al massimo uno. [pt. 1.25]

Totale punti = 2 + 4 + 4 = 10

package esami;

```
public class ForzaQuattro_20110915 {
    /*@ spec_public @*/ private int[][] board;
    /*@ spec_public @*/ private int[] freeRow;

    //tutte le celle della scacchiera valgono 0
    //@ ensures (\forall int i, j; i >= 0 && i < 4 && j >= 0 && j < 4; board[i][j] == 0);
    public ForzaQuattro_20110915() {
        board = new int[4][4];
        freeRow = new int[4];
    }

    //il parametro color indica un colore valido
    //@ requires color > 0 && color < 3;
    //il parametro column indica una colonna valida
    //@ requires column >= 0 && column < 4;
    //il contenuto delle colonne diverse da column dopo l'esecuzione del metodo e'
    //uguale al contenuto prima dell'esecuzione del metodo
    //@ ensures (\forall int i, j; i >= 0 && i < 4 && j >= 0 && j < 4 && j != column; \old(board[i][j]) == board[i]
[j]);
    //la differenza in valore assoluto tra il numero di celle rosse e il numero di celle nere e' al massimo uno
    //@ ensures Math.abs((\num_of int h, k; h >= 0 && h < 4 && k >= 0 && k < 4; board[h][k] == 1) -
    //@ (\num_of int h, k; h >= 0 && h < 4 && k >= 0 && k < 4; board[h][k] == 2)) <= 1;
    public void addPedina(int color, int column) {
        if(color > 0 && color < 3 && column >= 0 && column < 4) {
            if(freeRow[column] < 4) {
                board[freeRow[column]][column] = color;
                freeRow[column]++;
            }
        }
    }

    public boolean won(int color) {
        int counter;
        for(int i = 0; i < 4; i++) {
            counter = 0;
            for(int j = 0; j < 4; j++) {
                if(board[i][j] == color) {
                    counter++;
                }
                else {
                    break;
                }
            }
            if(counter == 4) {
                return true;
            }
        }
        for(int j = 0; j < 4; j++) {
            counter = 0;
            for(int i = 0; i < 4; i++) {
                if(board[i][j] == color) {
                    counter++;
                }
                else {
                    break;
                }
            }
        }
    }
}
```

```
        }  
    }  
    if(counter == 4) {  
        return true;  
    }  
}  
return false;  
}
```

```
public static void main(String[] args) {  
    ForzaQuattro_20110915 f = new ForzaQuattro_20110915();  
    //viola la preconditione "il parametro color indica un colore valido"  
    //f.addPedina(3, 0);  
  
    //viola la preconditione "il parametro column indica una colonna valida"  
    //f.addPedina(2, 5);  
}  
}
```

```

package esami;

import org.junit.Test;

public class ForzaQuattro_20110915_Condizioni {

    //"color > 0" a true
    //"color < 3" a true
    //"column >= 0" a true
    //"column < 4" a true
    //"freeRow[column] < 4" a true
    @Test
    public void testCondizioni1() {
        ForzaQuattro_20110915 f = new ForzaQuattro_20110915();
        f.addPedina(1, 0);
    }

    //"freeRow[column] < 4" a false
    @Test
    public void testCondizioni2() {
        ForzaQuattro_20110915 f = new ForzaQuattro_20110915();
        f.addPedina(1, 0);
        f.addPedina(2, 0);
        f.addPedina(1, 0);
        f.addPedina(2, 0);
        f.addPedina(1, 0);
    }

    //"color > 0" a false
    @Test
    public void testCondizioni3() {
        ForzaQuattro_20110915 f = new ForzaQuattro_20110915();
        f.addPedina(-1, 0);
    }

    //"color < 3" a false
    @Test
    public void testCondizioni4() {
        ForzaQuattro_20110915 f = new ForzaQuattro_20110915();
        f.addPedina(4, 0);
    }

    //"column >= 0" a false
    @Test
    public void testCondizioni5() {
        ForzaQuattro_20110915 f = new ForzaQuattro_20110915();
        f.addPedina(1, -1);
    }

    //"column < 4" a false
    @Test
    public void testCondizioni6() {
        ForzaQuattro_20110915 f = new ForzaQuattro_20110915();
        f.addPedina(1, 5);
    }
}

```

```
package esami;

import org.junit.Test;

public class ForzaQuattro_20110915_Decisioni {

    //"color > 0 && color < 3 && column >= 0 && column < 4" a true
    //"freeRow[column] < 4" a true
    @Test
    public void testDecisioni1() {
        ForzaQuattro_20110915 f = new ForzaQuattro_20110915();
        f.addPedina(1, 0);
    }

    //"freeRow[column] < 4" a false
    @Test
    public void testDecisioni2() {
        ForzaQuattro_20110915 f = new ForzaQuattro_20110915();
        f.addPedina(1, 0);
        f.addPedina(2, 0);
        f.addPedina(1, 0);
        f.addPedina(2, 0);
        f.addPedina(1, 0);
    }

    //"color > 0 && color < 3 && column >= 0 && column < 4" a false
    @Test
    public void testDecisioni3() {
        ForzaQuattro_20110915 f = new ForzaQuattro_20110915();
        f.addPedina(-1, 0);
    }
}
```