A.A. 2016 - 17

Metodologie di Programmazione (canale A-L)
Homework 2 - consegna 29 marzo 2017

Definire una classe Java `MyIntTree` che implementi il tipo dato astratto degli *alberi* di interi. Ciascun nodo di un albero deve avere un valore intero associato (accessibile tramite il metodo `getValue`), e può avere un numero arbitrario (ma finito), anche zero, di figli, ciascuno dei quali è a sua volta un albero. `MyIntTree` deve implementare la seguente interfaccia; i metodi sono spiegati nel commento.

```
public interface IntTree { // contratto degli alberi di interi

   public int getValue(); // restituisce il valore associato alla radice

   public int childrenNumber ();
      // restituisce il numero di figli di this (0 se e' una foglia)

   public int nodes (); // restituisce il numero di nodi di this; 1 se this  una foglia

   public int height ();
      // restituisce la lunghezza del cammino piu' lungo dalla radice a una foglia

   public boolean equals (IntTree t);
      /* t1.equals(t2) e' true se t1 e t2 sono isomorfi,
      * ovvero indistunguibili ad un osservatore esterno */

   public void addChild (IntTree child); // aggiunge child come primo figlio di this

   public IntTree followPath (int[] path) throws NoSuchTreeException;
      /* restituisce il sottoalbero individuato da path. Ad esempio, se path = [2,3,1],
      * restituisce il primo figlio del terzo figlio del secondo figlio di this;
      * se un tale sottoalbero non esiste lancia NoSuchTreeException */

   public void visit ();
      /* stampa la sequenza di valori associati ai nodi dell'albero corrispondente
      *  ad una visita in profondita' (depth-first) pre-order */
```

```
   public void printIntTree ();
      /* pritty prints this, ad esempio usando le parentesi, oppure l'indentazione */
}
```

Esempio. In questo esempio rappresentiamo un albero come una lista con parentesi: il primo elemento è la radice, i seguenti sono i figli. Una visita in profondità pre-order dell'albero (2 (3 5 9) 4) produce la sequenza: 2 3 5 9 4.

L'implementazione deve funzionare correttamente in combinazione con il seguente programma main:

```
public class TestIntTree {

   public static void main(String[] args) throws NoSuchTreeException {
      // test MyIntTree

      MyIntTree t = new MyIntTree(7); // crea un albero con un solo nodo etichettato 7
      System.out.println(t.nodes()); // stampa 1
      System.out.println(t.height()); // stampa 0

      MyIntTree t1 = new MyIntTree(9);
      MyIntTree t2 = new MyIntTree(9);
      t1.addChild(new MyIntTree(5));
      t2.addChild(new MyIntTree(5));
      System.out.println(t1.equals(t2)); // stampa true

      t1.addChild(t2);
      t1.addChild(new MyIntTree(3));
      t1.visit(); // stampa 9 3 9 5 5

      int[] path = {2,1};
      t.addChild(t1.followPath(path));
      t.visit(); // stampa 7 5
      t.followPath(path); // lancia NoSuchTreeException
   }
}
```