

Linguaggi di Programmazione con Laboratorio
Corso di Laurea in Matematica
Appello del 19 Giugno 2017

Nota. Per superare la prova occorre svolgere tutti gli esercizi, riportando su ciascuno il punteggio minimo di 15/30

Esercizio1 (Tempo Atteso per lo Svolgimento = 25')

Si consideri l'API `AlberoAPI<A>`, nell'allegato, che introduce valori per esprimere alberi etichettati con oggetti di generico tipo `A`.

Si discuta in Java, un ADT, `AlberoADT<A>`, per tale API, limitandosi a rispondere ai soli seguenti punti:

- a. Si dica se gli alberi di tipo `AlberoAPI<A>` sono Modificabili o Non-Modificabili, motivando la risposta (in 2 righe max.);
- b. Si mostri l'intestazione della classe e lo stato concreto utilizzato per l'implementazione scelta;
- c. Si definiscano i costruttori della classe;
- d. Si definiscano `AF` ed `I`.
- e. Si mostri la definizione di `getSize`

Esercizio2 (TAS = 25')

Si considerino i valori albero definiti in Java in esercizio 1.

- a. Definire, in OCaml, un modulo API, `TREEAPI`, per tali valori.
- b. Si assuma definita, in OCaml, un ADT, `TreeADT`, per `TREEAPI`. Vogliamo definire un modulo `TreeADTE` che specializzi `TreeADT` in alberi contenenti al più `k` nodi, dove $k \geq 0$ sia un parametro fornito al momento della costruzione del valore. A questo scopo si risponda ai soli punti seguenti:
 1. Si mostrino intestazione e stato concreto del modulo `TreeADTE`;
 2. Si mostri la definizione dei costruttori;
 3. Si mostri la definizione delle operazioni `isRoot` e `getSize`.

Esercizio3 (TAS = 20')

Sia `MuQueueAPI` l'interfaccia, allegata, per code Modificabili. Sia `MuQSingleADTXE` un ADT già definito per `MuQueueAPI` con l'ulteriore vincolo di singolo accodamento degli oggetti. `MuQSingleADTXE` è equipaggiato con gli additional (`equals`, `clone`, `toString`, `elements`).

Si chiede di specializzare `MuQSingleADTXE` in un ADT `MuQSingleADTXEE` per code aventi numero massimo $K \geq 0$ di oggetti in coda, dove `K` sia fissato al momento della costruzione della coda.

Esercizio1 (Allegato: AlberoAPI<A>)

/*

AlberoAPI<A>, introduce valori della forma [V - T1,..., Tn], per alberi etichettati con oggetti di generico tipo A. V deve essere un oggetto di tipo A e T1,...,Tn devono essere n>=0 alberi di tipo AlberoAPI<A>. Quando n=0, il valore [V] esprime l'abero senza figli.

Infine, [] esprime l'albero vuoto che vogliamo anche per distinguere [V - []] da [V].

*/

```
public interface AlberoAPI<A>{
    public boolean isEmpty();
    public boolean isRoot();
    public boolean hasSons();
    public int getSize();
    public A getRoot();
    public LinkedList<AlberoAPI<A>> getSons();
}
```

Esercizio3 (Allegato: MuQueueAPI<A>)

```
interface MuQueueAPI<A> extends Cloneable{
    public void add(A x);
    public A get();
    public void remove();
    public int size();
    public boolean empty();
}
```