
Programmazione II, 6 febbraio 2012 – Laurea in INFORMATICA**Esercizio 1**[16 punti]

Realizzare una classe `AlberoBinario` che implementa un albero binario i cui nodi sono oggetti della classe `Nodo` definita come segue:

```
public class Nodo {  
  
    private int value;  
    private Nodo left; // figlio sinistro  
    private Nodo right; // figlio destro  
    public Nodo(int valore) {  
value= valore;  
left= null;  
right= null;  
    }  
    /**  
     * Permette di leggere il valore associato al nodo  
     * @return un intero corrispondente al valore del nodo  
     */  
    public int getValue() { return value; }  
  
    public void setLeftChild(Nodo child) { left= child; }  
    public void setRightChild(Nodo child) { right= child; }  
    public Nodo getLeftChild() { return left; }  
    public Nodo getRightChild() { return right; }  
}
```

La classe `AlberoBinario` dovrà contenere

- una variabile di istanza `private`, `root`, di tipo `Nodo`, che rappresenta la radice dell'albero;
- un costruttore senza parametri che crea un albero vuoto,

e implementare la seguente interfaccia:

```
public interface IntAlbero {  
  
    /**  
     * Permette di inserire l'intero passato come parametro  
     * nella prima posizione libera piu' a sinistra  
     */  
    public void InserisciValore(int valore);  
  
    /**  
     * Permette di cercare un valore nel sotto-albero sinistro  
     * @param valore il valore da cercare  
     * @return true se il valore presente, false altrimenti  
     */  
    public boolean RicercaValore(int valore);  
  
    /**  
     * Cancella un valore dal sotto-albero sinistro
```

```

    * @param valore il valore da cancellare
    * @throws TreeException se il valore non è presente nell'albero
    */
    public void EliminaValore(int valore) throws TreeException;
}

```

La classe `TreeException` è definita come segue:

```

public class TreeException extends RuntimeException {
    public TreeException(String err) {
        super(err);
    }
}

```

Esercizio 2[16 punti]

- (a) Definire una classe `ErreTre` i cui oggetti sono vettori nello spazio reale a tre dimensioni. La classe deve contenere i seguenti campi e metodi:
- tre variabili `protected` di tipo `double` (le coordinate del vettore),
 - un costruttore che inizializza le coordinate con i valori passati come parametro,
 - un metodo `String toString()` che restituisce una descrizione del vettore.
 - i metodi `void piu(ErreTre v)` e `void moltS(double s)` che effettuano rispettivamente la somma tra vettori e la moltiplicazione per uno scalare
 - il metodo `boolean equals(Object o)` che restituisce `true` se l'oggetto `o` è un vettore con coordinate uguali a quelle dell'oggetto su cui il metodo è invocato.
- (b) Definire quindi la classe `ErreQuattro` dei vettori dello spazio a quattro dimensioni come sottoclasse di `ErreTre`.
- (c) Scrivere infine un programma `Vettori` che legge e stampa oggetti costruiti con le classi definite sopra.