

Laboratorio di Algoritmi e Strutture dati

29 aprile 2015

★ Esercizio searchtree

Si crei un package *dictionary*, all'interno del quale si realizzino, insieme alle corrispondenti classi di test, le interfacce e classi seguenti.

Un'interfaccia **Dictionary** che rappresenti il tipo di dato astratto *dizionario*. Il tipo di dato astratto deve permettere di associare a chiavi di un tipo generico *K* valori di un tipo generico *V*:

```
public interface Dictionary<K,V> {  
    /*  
     * restituisce true sse il dizionario è vuoto  
     */  
    boolean isEmpty();  
  
    /*  
     * restituisce il valore associato alla  
     * chiave key (o null se la chiave non è  
     * presente)  
     */  
    V find(K key);  
  
    /*  
     * associa il valore value alla chiave  
     * key e restituisce il valore precedentemente  
     * associato a key (o null se la chiave non  
     * era associata a nessun valore)  
     */  
    V add(K key, V value);  
  
    /*  
     * rimuove la chiave key dal dizionario,  
     * se presente; altrimenti non fa nulla  
     */  
    void remove(K key);  
}
```

Un'interfaccia derivata **SortedDictionary** che realizzi il tipo astratto *dizionario ordinato*, cioè un dizionario in cui vi è un ordine totale sulle chiavi:

```

public interface SortedDictionary<K extends Comparable<? super
    K>, V> extends Dictionary<K, V> {
    K minKey();
    K maxKey();
    V elementOfMinKey();
    V elementOfMaxKey();
}

```

Implementare l'interfaccia **SortedDictionary** tramite una e una sola delle seguenti classi:

SearchTree (4pt) : implementa l'interfaccia tramite un albero di ricerca semplice;

AVLTree (8pt) : implementa l'interfaccia tramite un albero di ricerca bilanciato AVL.