

ΑΣΚΗΣΗ 6

Πλήρη και Σχεδόν Πλήρη Δυαδικά Δέντρα Σωρός (Heap) - Ουρές Προτεραιότητας (Priority Queues)

(Βλέπε http://www.iee.ibu.gr/~demos/teaching_GR.html)

Δίνεται το παρακάτω interface, το οποίο περιγράφει τον αφηρημένο τύπο δεδομένων Σωρός (Heap) (ορίζει τις πράξεις της δομής δεδομένων σωρός).

```
public interface HeapInterface {  
  
    public int size();  
    // Επιστρέφει το μέγεθος του σωρού (αριθμό των στοιχείων του)  
  
    public boolean isEmpty();  
    // Επιστρέφει true αν ο σωρός είναι άδειος, false στην  
    // αντίθετη περίπτωση  
  
    public boolean isFull();  
    // Επιστρέφει true αν ο σωρός είναι γεμάτος, false στην  
    // αντίθετη περίπτωση (υπερχείλιση σωρού)  
  
    public void insert(Object item) throws HeapFullException;  
    // Εισάγει ένα νέο στοιχείο στο σωρό  
  
    public Object remove() throws HeapEmptyException;  
    // Διαγράφει και επιστρέφει το στοιχείο που βρίσκεται  
    // στη ρίζα του σωρού  
}
```

Άσκηση 6.1

Να δώσετε μια υλοποίηση του *HeapInterface* με την βοήθεια πίνακα. Στη συνέχεια γράψτε ένα μικρό πρόγραμμα για να ελέγχει τη λειτουργία του σωρού. Το πρόγραμμά σας θα πρέπει να εισάγει διαδοχικά έναν αριθμό δεδομένων στο σωρό με τη χρήση της μεθόδου **insert**. (Θεωρήστε για παράδειγμα ότι τα δεδομένα που εισάγονται είναι τύπου **String** ή τύπου **Student**). Στη συνέχεια διαγράψτε διαδοχικά όλα τα δεδομένα από το σωρό με τη χρήση της μεθόδου **remove**, εμφανίζοντάς τα στην οθόνη. Μία ένδειξη ότι το πρόγραμμά σας δουλεύει σωστά είναι να δείτε τα δεδομένα που εισαγάγατε στο σωρό να εμφανίζονται στην οθόνη κατά φθίνουσα διάταξη.

Άσκηση 6.2

Με βάση την υλοποίηση του σωρού γράψτε μια μέθοδο, η οποία δοθέντος ενός πίνακα με δεδομένα σε τυχαία σειρά να τον ταξινομεί σε αύξουσα διάταξη (ή σε φθίνουσα αντίστοιχα).

Άσκηση 6.3

Υλοποιήστε τα ζητούμενα στις ασκήσεις .6.1 και 6.2 χρησιμοποιώντας την κλάση **ArrayList** της Java στη θέση του πίνακα για την υλοποίηση του σωρού.