

Η Διασύνδεση του Σωρού

```
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  
  
}
```

Ενδεικτική χρήση της κλάσης Heap (1)

```
public static void main(String[] args) {  
  
    Student matrix1[ ] = {  
        new Student(17040, "Kostas"),  
        new Student(17015, "Nikos"),  
        new Student(17025, "Anastasia"),  
        new Student(17050, "Vasiliki"),  
        new Student(17020, "Maria"),  
        new Student(17010, "Vasilis"),  
        new Student(17015, "Dimitra"),  
        new Student(17055, "Petros"),  
        new Student(17045, "Sofia"),  
        new Student(17005, "Alekos"),  
    };  
  
    Heap myHeap = new Heap();  
}
```

Ενδεικτική χρήση της κλάσης Heap (1-συνέχεια)

```
try {  
  
    for (int i=0; i<matrix1.length; i++)  
        myHeap.insert(matrix1[i]);  
  
    int size=myHeap.size();  
  
    for (int i=0; i<size; i++) {  
        Student S= (Student)myHeap.remove();  
        System.out.println(S);  
    }  
}  
catch (HeapFullException Error) {  
    System.out.println(Error);  
}  
catch (HeapEmptyException Error) {  
    System.out.println(Error);  
}  
catch (Exception Error) {  
    System.out.println(Error);  
}  
}
```

Ταξινόμηση μέσω σωρού (heapsort)

```
public static boolean heapSort(Object[] array) {  
  
    Heap TransferHeap = new Heap(20);  
  
    try{  
        for (int i=0; i<array.length; i++)  
            TransferHeap.insert(array[i]);  
        int size = TransferHeap.size();  
        for (int i=size-1; i>=0; i--)  
            array[i]=TransferHeap.remove();  
        return true;  
    }  
    catch (HeapFullException Error){  
        System.out.println("ARRAY TOO BIG! UNABLE TO SORT");  
        return false;  
    }  
}
```

Ενδεικτική χρήση της κλάσης Heap (2)

```
public static void main(String[] args) {  
  
    String matrix2[ ] = {"Kostas", "Nikos", "Yannis", "Anna",  
                          "Ioulia", "Stelios", "Andreas", "Dimitris",  
                          "Panos", "Eleni", "Katerina"};  
  
    System.out.println("##### INITIAL ARRAY #####");  
    for (int i=0; i<matrix2.length; i++)  
        System.out.println(matrix2[i]);  
  
    if (heapSort(matrix2)) {  
        System.out.println("##### SORTED ARRAY #####");  
        for (int i=0; i<matrix2.length; i++)  
            System.out.println(matrix2[i]);  
    }  
}
```