

OrderedListPriorityQueue.java

```
1  /* Vincent Chan
2   * masc0264
3   */
4  package data_structures;
5
6  import java.util.Iterator;
7
8  public class OrderedListPriorityQueue<E> implements PriorityQueue<E>{
9      /* Functions Included
10       * ====PUBLIC====
11       * Constructor           //Ln. 26
12       * insert(object)        //Ln. 31
13       * remove()              //Ln. 38
14       * peek()                //Ln. 44
15       * contains(object)      //Ln. 51
16       * size()                 //Ln. 57
17       * clear()                //Ln. 62
18       * isEmpty()              //Ln. 67
19       * isFull()               //Ln. 72
20       * iterator()            //Ln. 78
21       */
22
23     //Variable Declarations
24     OrderedList<E> list;
25
26     //Constructor
27     public OrderedListPriorityQueue() {
28         list = new OrderedList();
29     } //End Constructor
30
31     //Inserts a new object into the queue.
32     //Returns true if insertion successful
33     public boolean insert(E object) {
34         list.insert(object);
35         return true;
36     } //End insert()
37
38     //Removes the highest priority object
39     //in the queue the longest.
40     public E remove() {
41         return list.pop();
42     } //End remove()
43
44     //Returns the object of the highest priority
45     //that has been in the queue the longest
46     //but does not remove it.
47     public E peek() {
48         return list.peek();
49     } //End peek()
50
51     //Returns true of the priority queue contains
52     //specified element
53     public boolean contains(E obj) {
54         return list.contains(obj);
55     } //End contains()
56
57     //Returns the size of the list
58     public int size() {
59         return list.size();
60     } //End size()
61
62     //Returns the PQ to an empty state
63     public void clear() {
64         list = new OrderedList();
```

```
65     } //End clear()
66
67     //Returns true if the queue is empty
68     public boolean isEmpty() {
69         return list.isEmpty();
70     } //End isEmpty()
71
72     //Returns true if the queue is full.
73     //Lists always return false.
74     public boolean isFull() {
75         return false;
76     } //End isFull()
77
78     //Returns an Iterator of the queue.
79     public Iterator<E> iterator() {
80         return list.iterator();
81     } //End iterator()
82 } //End OrderedListPriorityQueue
```