~\OneDrive - VietNam National University - HCM INTERNATIONAL UNIVERSITY\Desktop\DSA\DSA LAB NEW\Lab 3 Stacks & Queues\ITITSB22029_DoMinhDuy_Lab3\PriorityQApp\PriorityQApp.java

```
1  // priorityQ.java
2  // demonstrates priority queue
3  // to run this program: C>java PriorityQApp
4  ////////////////////////////////////////////////////////////////
5  class PriorityQ {
6     private int maxSize;
7     private long[] queArray;
8     private int nItems;
9
10    public PriorityQ(int s) { // constructor
11       maxSize = s;
12       queArray = new long[maxSize];
13       nItems = 0;
14    }
15
16    public void insert(long item) { // insert item in sorted order
17       int j;
18       if (nItems == 0) { // if no items,
19          queArray[nItems++] = item; // insert at 0
20       } else { // if items,
21          for (j = nItems - 1; j >= 0; j--) { // start at end,
22             if (item < queArray[j]) // if new item smaller,
23                queArray[j + 1] = queArray[j]; // shift upward
24             else
25                break; // done shifting
26          }
27          queArray[j + 1] = item; // insert it
28          nItems++;
29       }
30    }
31
32    public long remove() { // remove minimum item
33       return queArray[--nItems];
34    }
35
36    public long peekMin() { // peek at minimum item
37       return queArray[nItems - 1];
38    }
39
40    public boolean isEmpty() { // true if queue is empty
41       return (nItems == 0);
42    }
43
44    public boolean isFull() { // true if queue is full
45       return (nItems == maxSize);
46    }
47
```

```
48    public void displayQueue() { // display the current state of the queue
49       System.out.print("Queue: ");
50       for (int i = 0; i < nItems; i++) {
51          System.out.print(queArray[i] + " ");
52       }
53       System.out.println();
54    }
55 }
56 
57 class CustomerQueueApp {
58    public static void main(String[] args) {
59       PriorityQ customerQueue = new PriorityQ(10); // A priority queue to hold customers
60 
61       // Simulating customer arrivals with different priority levels
62       customerQueue.insert(30); // Customer 1 (low priority)
63       customerQueue.displayQueue();
64       customerQueue.insert(10); // Customer 2 (high priority)
65       customerQueue.displayQueue();
66       customerQueue.insert(20); // Customer 3 (medium priority)
67       customerQueue.displayQueue();
68 
69       // Simulate serving customers based on priority
70       System.out.println("Serving customers based on priority:");
71       while (!customerQueue.isEmpty()) {
72          long customer = customerQueue.remove(); // Serve highest-priority customer first
73          System.out.println("Serving customer with priority: " + customer);
74          customerQueue.displayQueue();
75       }
76    }
77 }
78 
79 ////////////////////////////////////////////////////////////////
80 
```