

```
1 package components.WaitingLine;
2
3
4 import components.standard.Standard;
5 import components.queue.Queue;
6 import components.queue.Queue1L;
7
8
9     private Queue1L rep;
10
11     /**
12      * Creator of initial representation.
13      */
14     private void createNewRep() {
15         this.rep = new Queue1L<>();
16     }
17
18     /**
19      * No-argument constructor.
20      */
21
22     public WaitingLineKernal() {
23         this.createNewRep();
24     }
25
26     public void add(T x) {
27         if (this.rep.length() > 0) {
28             T y = this.rep.dequeue();
29             if (y.equals(x)) {
30                 System.out.println("You are already in the Waiting
31 Line");
32             }
33         } else {
34             this.rep.enqueue(x);
35         }
36     }
37
38     public T remove(int pos) {
39         Queue<T> temp = new Queue1L<>();
40         for(int i = 0; i > pos - 1; i++) {
41             T y = this.rep.dequeue();
42             temp.enqueue(y);
43         }
44         T x = this.rep.dequeue();
```

```
45         if(temp.length() > 0) {
46             T y = temp.dequeue();
47             this.rep.enqueue(y);
48         }
49
50         return x;
51     }
52
53     public T seat() {
54         T x = this.rep.dequeue();
55         return x;
56     }
57
58     public int locate(T x) {
59         int place = -1;
60         Queue<T> temp = new Queue1L<>();
61         for (int i = 0; i < this.rep.length(); i++) {
62             T y = this.rep.dequeue();
63             if (y.equals(x)) {
64                 place = i;
65             }
66             temp.enqueue(y);
67         }
68         if(temp.length() > 0) {
69             T y = temp.dequeue();
70             this.rep.enqueue(y);
71         }
72         return place;
73     }
74
75     public static <T> void main(String[] args) {
76
77     }
78
79 }
80
```