

```
1
2 import java.util.Iterator;
3
4 import components.queue.Queue;
5 import components.queue.QueueSecondary;
6 import components.sequence.Sequence;
7 import components.sequence.Sequence1L;
8
9 /**
10  * {@code Queue} represented as a {@code Sequence} of
11  *   entries, with
12  *   implementations of primary methods.
13  *   @param <T>
14  *       type of {@code Queue} entries
15  *   @correspondence this = $this.entries
16  */
17 public class Queue3<T> extends QueueSecondary<T> {
18
19     /*
20      * Private members
21      */
22
23     /**
24      * Entries included in {@code this}.
25      */
26     private Sequence<T> entries;
27
28     /**
29      * Creator of initial representation.
30      */
31     private void createNewRep() {
32         this.entries = new Sequence1L<T>();
```

```
33     }
34
35     /*
36     * Constructors
37     */
38
39     /**
40     * No-argument constructor.
41     */
42     public Queue3() {
43         this.createNewRep();
44     }
45
46     /*
47     * Standard methods removed to reduce clutter...
48     */
49
50     /*
51     * Kernel methods
52     */
53
54     @Override
55     public final void enqueue(T x) {
56         assert x != null : "Violation of: x is not null";
57
58         this.entries.add(0, x);
59
60     }
61
62     @Override
63     public final T dequeue() {
```

```
64         assert this.length() > 0 : "Violation of: this /=  
    <>";  
65  
66         return this.entries.remove(this.entries.length() -  
    1);  
67     }  
68  
69     @Override  
70     public final int length() {  
71  
72         return this.entries.length();  
73     }  
74  
75     @Override  
76     public void clear() {  
77         // TODO Auto-generated method stub  
78  
79     }  
80  
81     @Override  
82     public Queue<T> newInstance() {  
83         // TODO Auto-generated method stub  
84         return null;  
85     }  
86  
87     @Override  
88     public void transferFrom(Queue<T> arg0) {  
89         // TODO Auto-generated method stub  
90  
91     }  
92  
93     @Override  
94     public Iterator<T> iterator() {  
95         // TODO Auto-generated method stub
```

```

96         return null;
97     }
98
99     /*
100      * Iterator removed to reduce clutter...
101      */
102
103     /**
104      * Reports the front of {@code this}.
105      *
106      * @return
107      *
108      * @aliases reference returned by {@code front}
109      * @requires this != <>
110      * @ensures <front> is prefix of this
111      */
112     @Override
113     public T front() {
114         assert this.length() > 0 : "Violation of: this !=
115         <>";
116         T temp = this.dequeue();
117         //copying the temporary value into a new variable
118         // so it does not copy
119         // its reference value
120         T temp2 = new T(temp); //Did not know how to
121         properly do this part
122         this.enqueue(temp2);
123         for (int i = 0; i < this.length() - 2; i++) {
124             temp2 = this.dequeue();
125             this.enqueue(temp2);
126         }
127         return temp;
128     }
129 }
```

Queue3.java

Wednesday, August 31, 2022, 6:30 PM

127 }

128