

```
1
2 import components.queue.QueueSecondary;
5
6 /**
7  * {@code Queue} represented as a {@code Sequence} of entries, with
8  * implementations of primary methods.
9  *
10 * @param <T>
11 *         type of {@code Queue} entries
12 * @correspondence this = $this.entries
13 */
14 public class Queue3<T> extends QueueSecondary<T> {
15
16     /*
17      * Private members
18
19      */
20     /**
21      * Entries included in {@code this}.
22      */
23     private Sequence<T> entries;
24
25     /**
26      * Creator of initial representation.
27      */
28     private void createNewRep() {
29         this.entries = new Sequence1L<T>();
30     }
31
32     /*
33      * Constructors
34
35      */
36     /**
37      * No-argument constructor.
38      */
39     public Queue3() {
40         this.createNewRep();
41     }
42
43     /*
44      * Standard methods removed to reduce clutter...
```

```
45     */
46
47     /*
48     * Kernel methods
49     */
50
51     @Override
52     public final void enqueue(T x) {
53         assert x != null : "Violation of: x is not null";
54
55         this.entries.add(this.entries.length(), x);
56     }
57
58     @Override
59     public final T dequeue() {
60         assert this.length() > 0 : "Violation of: this != <>";
61
62         T x = this.entries.remove(0);
63         return x;
64     }
65
66     @Override
67     public final int length() {
68
69         return this.entries.length();
70     }
71
72     /**
73      * Reports the front of {@code this}.
74      *
75      * @return the front entry of {@code this}
76      * @aliases reference returned by {@code front}
77      * @requires this != <>
78      * @ensures <front> is prefix of this
79      */
80     //Did it in the same file
81     @Override
82     public T front() {
83         assert this.length() > 0 : "Violation of: this != <>";
84
85         //Created a temporary variable to avoid using the instance
86         variable Queue3<T> temp = this;
```

```
87         T x = temp.dequeue();
88
89         return x;
90
91     }
92
93     /*
94     * Iterator removed to reduce clutter...
95     */
96
97 }
```