

```
1 import components.map.Map;
2 import components.map.Map.Pair;
3 import components.map.Map1L;
4 import components.queue.Queue;
5 import components.queue.Queue1L;
6 import components.standard.Standard;
7
8 /**
9  * Interface for a to-do list kernel component with primary
10  * methods.
11  *
12  * This interface defines methods for managing to-do list events,
13  * including
14  * adding events, retrieving their importance and date, and
15  * converting
16  * importance and date values to strings.
17  *
18  * This interface also includes fields for storing event
19  * information and sorted
20  * queues for efficient retrieval and sorting of events by
21  * importance and date.
22  *
23  * @initially <pre>
24  *   ensures
25  *     event = "" and
26  *     allEventsImp = {} and
27  *     allEventsDate = {} and
28  *     sortedByImp = <>, sortedByDate = <>
29  * </pre>
30  *
31  * @author Jahnvi Acharya
32  */
33 public interface ToDoListKernel extends Standard<ToDoList> {
34
35     /**
36      * Constant representing an empty string for the event.
37      */
38     String EVENT = "";
39
40     /**
41      * Map containing all events and their importance.
42      */
43     Map<String, Integer> ALL_EVENTS_IMP = new Map1L<String,
44 Integer>();
45 }
```

```
40     /**
41      * Map containing all events and their date.
42      */
43     Map<String, Integer> ALL_EVENTS_DATE = new Map1L<String,
Integer>();
44
45     /**
46      * Queue containing pairs of events and their importance,
sorted by
47      * importance.
48      */
49     Queue<Pair<String, Integer>> SORTED_BY_IMP = new Queue1L<>();
50
51     /**
52      * Queue containing pairs of events and their date, sorted by
date.
53      */
54     Queue<Pair<String, Integer>> SORTED_BY_DATE = new Queue1L<>();
55
56     /**
57      * Retrieves the importance of the specified event.
58      *
59      * @param event
60      *           the event to retrieve importance for
61      * @requires The specified event exists in {@code
ALL_EVENTS_IMP}.
62      * @ensures Returns the importance of the specified event.
63      * @return the importance number of the event
64      */
65     int getImportance(String event);
66
67     /**
68      * Retrieves the date of the specified event.
69      *
70      * @param event
71      *           the event to retrieve date for
72      * @requires The specified event exists in {@code
ALL_EVENTS_IMP}.
73      * @ensures Returns the date of the specified event in the
right format of
74      *           YearMonthDay.
75      * @return the date number of the event
76      */
77     int getDate(String event);
78
```

```
79     /**
80      * Adds importance for the specified event.
81      *
82      * @param event
83      *         the event to add importance for
84      * @param imp
85      *         the importance to add
86      * @requires event to be a key in {@code ALL_EVENTS_IMP} and
imp is an
87      *         Integer between -1 and 3 inclusive.
88      * @ensures updates the importance for the specified event.
89      */
90     void addImportance(String event, int imp);
91
92     /**
93      * Adds date for the specified event.
94      *
95      * @param event
96      *         the event to add date for
97      * @param date
98      *         the date to add
99      * @requires event to be a key in {@code ALL_EVENTS_IMP} and
date is a 6
100     *         digit integer in the format YearMonthDay.
101     * @ensures updates the date for the specified event.
102     */
103     void addDate(String event, int date);
104
105     /**
106      * Converts the given importance value to a string
representation.
107      *
108      * @param i
109      *         the importance value to convert
110      * @requires i to be an integer between -1 and 3 inclusive.
111      * @ensures Returns the string representation of the
importance.
112      * @return the string representation of the importance
113      */
114     String getStringImportance(int i);
115
116     /**
117      * Converts the given date value to a string representation.
118      *
119      * @param i
```

```
120      *           the date value to convert
121      * @requires i to be a 6 digit integer in the format
      YearMonthDay.
122      * @ensures Returns the string representation of the date.
123      * @return the string representation of the date
124      */
125      String getStringDate(int i);
126
127 }
128
```