C:\Users\Rich\Documents\NetBeansProjects\Lab08\src\Tree.java

```
1
2 import java.util.Iterator;
3
4 /**
5  *
6  * @author Goodrich, Tamassia, Goldwasser
7  */
8 public interface Tree<E> extends Iterable<E> {
9   /**
10   *
11   * @return the position of the root of the tree or null if empty.
12   */
13   Position<E> root();
14   /**
15   *
16   * @param p position
17   * @return the position of the parent of position p or null if p is the root
18   */
19   Position <E> parent(Position<E> p) throws IllegalArgumentException;
20
21   /**
22   *
23   * @param p
24   * @return an itreable collection containing the children of position p (if any)
25   * @throws IllegalArgumentException
26   */
27   Iterable<Position<E>>children(Position<E> p) throws IllegalArgumentException;
28
29   /**
30   *
31   * @param p
32   * @return the number of children of position p
33   *
34   */
35   int numChildren(Position<E> p) throws IllegalArgumentException;
36
37   /**
38   *
39   * @return true if positio p has at least one child
40   */
41   boolean isInternal(Position<E> p) throws IllegalArgumentException;
42
43   /**
44   *
45   * @param p
46   * @return treue if posion p does not have any children
47   */
48   boolean isExternal(Position<E> p) throws IllegalArgumentException;
49
50   /**
51   *
52   * @param p
53   * @return true if position p is the root of the tree
54   */
55   boolean isRoot(Position<E> p) throws IllegalArgumentException;
56
```

```
57    /**
58     *
59     * @return the number of positions(and hence elements)
60     * that are contained in the tree.
61     */
62    int size();
63
64    /**
65     *
66     * @return true if the tree does not contain any positions.
67     */
68    boolean isEmpty();
69
70    /**
71     *
72     * @return an iterator for all elements in the tree
73     * (so that the tree itself is Iterable).
74     */
75    Iterator<E> iterator();
76
77    /**
78     *
79     * @return an iterable collection of all position of the tree.
80     */
81    Iterable<Position<E>> positions();
82 }
83
```