

C:\Users\Rich\Documents\NetBeansProjects\Lab07\src\PositionalList.java

```

1 /**
2  * Data Structures & Algorithms 6th Edition
3  * Goodrick, Tamassia, Goldwasser
4  * Code Fragement 7.8
5  */
6
7 /**
8  * An interface for positional lists.
9  */
10 public interface PositionalList<E> {
11
12     /**
13      * @return the number of elements in the list.
14      */
15     int size();
16
17     /**
18      * @return true if the list is empty.
19      */
20     boolean isEmpty();
21
22     /**
23      * @return the first Position in the list ( or null, if empty ).
24      */
25     Position<E> first();
26
27     /**
28      * @return the last Position in the list ( or null, if empty ).
29      */
30     Position<E> last();
31
32     /**
33      * @param p a position in the list,
34      * @return position immediately before p ( or null if p is first ).
35      * @throws IllegalArgumentException if p is not in list.
36      */
37     Position<E> before( Position<E> p ) throws IllegalArgumentException;
38
39     /**
40      * @param p a position in the list,
41      * @return position immediately after p ( or null if p is last ).
42      * @throws IllegalArgumentException if p is not in list.
43      */
44     Position<E> after( Position<E> p ) throws IllegalArgumentException;
45
46     /**
47      * @param e element to be inserted at front of list
48      * @return position of inserted element
49      */
50     Position<E> addFirst( E e );
51
52     /**
53      * @param e element to be inserted at back of list
54      * @return position of inserted element
55      */
56     Position<E> addLast( E e );

```

```
57
58 /**
59  * @param p position to be inserted before
60  * @param e element to be inserted before position p
61  * @return position of e
62  * @throws IllegalArgumentException if p not in list
63  */
64 Position<E> addBefore( Position<E> p, E e ) throws IllegalArgumentException;
65
66 /**
67  * @param p position to be inserted after
68  * @param e element to be inserted after position p
69  * @return position of e
70  * @throws IllegalArgumentException if p not in list
71  */
72 Position<E> addAfter( Position<E> p, E e ) throws IllegalArgumentException;
73
74 /**
75  * @param p position to store element at
76  * @param e element to be stored at p
77  * @return the element that is replaced
78  * @throws IllegalArgumentException if p is not in list
79  */
80 E set( Position<E> p, E e ) throws IllegalArgumentException;
81
82 /**
83  * @param p position of element to be removed
84  * @return removed element
85  * @throws IllegalArgumentException if p not in list
86  */
87 E remove( Position<E> p ) throws IllegalArgumentException;
88
89 }
90
```