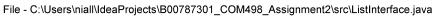
```
1 public interface ListInterface<T>
2 {
3
       public void add(T newEntry);
4
       /*Adds a new entry to end of the list.
5
       Entries currently in the list are unaffected.
6
       The list's size is increase by 1.
7
8
       Oparam newEntry The object to be added as a new list entry
9
10
11
       public void add(int newPosition, T newEntry);
12
       /*Adds a new entry to a specific position within the list.
13
       Entries currently at position are moved up by one.
14
       List size increased by 1.
15
16
       Oparam newPosition An int that specifies the index position of
17
       new entry.
18
19
       Oparam newEntry The object to be added as a new list entry
20
21
       @throws IndexOutOfBoundsException if either new position < 1
22
       or newPosition > getLength() + 1
23
        */
24
25
       public T remove(int position);
26
       /* Removes entry at given position from the list.
27
       Entries originally at higher positions all moved down by one
28
       and list size shrinks by 1.
29
30
       Oparam newPosition An int that specifies the index position of
31
       entry to be removed.
32
33
       @return Removed entry at given position
34
35
       @throws IndexOutOfBoundsException if either position to be removed
36
        < 1 or position to be removed > getLength()
37
38
39
       public void clear();
40
       /* Removes all entries from the list
41
42
43
44
       public T replace(int position, T newEntry);
45
       /* Replaces the entry at the given position in this list
46
47
       Oparam newPosition An int that specifies the index position of
48
       entry to be removed.
49
50
       @return Removed entry at given position
51
52
        @throws IndexOutOfBoundsException if either position to be replaced
53
        < 1 or position to be replaced > getLength()
54
55
56
       public T getEntry(int position);
57
       /* Retrieved the entry at the given position in this list
58
59
       Oparam newPosition An int that specifies the index position of
60
       entry to be retrieved.
61
62
       @return Removed entry at given position
63
64
        @throws IndexOutOfBoundsException if either position to be retrieved
65
        < 1 or position to be replaced > getLength()
66
67
68
       public T[] toArray();
69
       /st Retrieves all entries in the list in the order which they appear.
70
71
       Oreturn A newly allocated array of all the entries in the list.
72
       If the list is empty, returned array is empty.
73
74
75
       public int getLength();
76
```



```
@return the integer length of listArray
78
79
       public boolean isEmpty();
80
81
       /* Checks whether list is empty.
82
83
       @return true if getLength == 0, otherwise false
84
85 }
86
```