

GIT Department of Computer Engineering

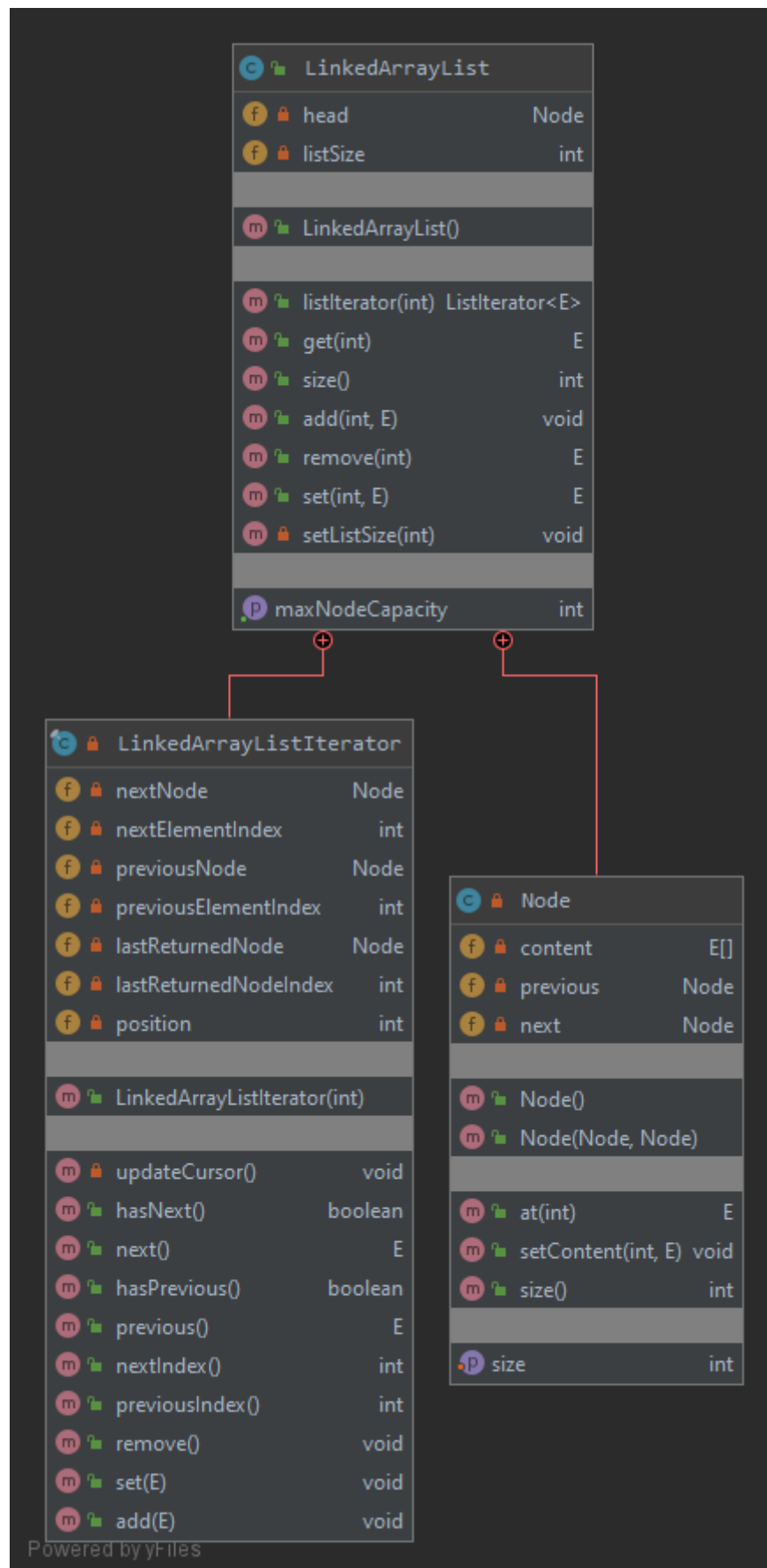
CSE 222/505 – Spring 2020

Homework #3 Part 1 Report

Abdullah ÇELİK

171044002

Class Diagram



Problem Solution Approach

LinkedList class implements List interface and extends AbstractList abstract class. In the API of AbstractList, we see the phrases

" This class provides a skeletal implementation of the [List](#) interface to minimize the effort required to implement this interface backed by a "random access" data store (such as an array) ".

To implement an unmodifiable list, the programmer needs only to extend this class and provide implementations for the [get\(int\)](#) and [size\(\)](#) methods.

To implement a modifiable list, the programmer must additionally override the [set\(int, E\)](#) method (which otherwise throws an UnsupportedOperationException). If the list is variable-size the programmer must additionally override the [add\(int, E\)](#) and [remove\(int\)](#) methods.

Before I override these methods to make a modifiable list, I wrote the nested iterator class and this class implements the ListIterator interface. After carefully overriding the methods required by the ListIterator interface in the class, for the methods I override the LinkedList class, I created the iterator to the required position and use the methods of the listiterator. As a result of these processes, LinkedList class started working correctly.

Test Cases

LinkedList class implements List interface and extends AbstractList abstract class. For this reason, the some methods required by the List interface have been overridden AbstractList abstract class. I overrode some AbstractList class methods and these methods were tested to work right or not. It was not necessary to override again the methods which were not overridden. Because these methods calls methods which were overridden and works right. So it was not tested methods which were overridden.

TEST ID	SCENERIO	TEST DATA	EXPECTED RESULTS	ACTUAL RESULTS	PASS /FAIL
T01	Creating character LinkedList	LinkedList Class	Succesfully created	As expected	Pass

T02	int size() method called when list is empty and has some elements	Size : 0 Size : 10	Successfully returned right size	As expected	Pass
T03	ListIterator listIterator(index) method called when list is empty and list has some elements	List size : 0 Index : 0 List size : 10 Index : 0 Index : 5 Index : 10	Successfully returned right iterator at desired index	As expected	Pass
T04	ListIterator listIterator(index) method called when list is empty and list has some elements	List size : 0 Index : -1 Index : 1 List size : 10 Index : -1 Index : 11	IndexOutOfBoundsException Exception thrown	As expected	Pass
T05	get(index) method called when list has some elements	List size : 10 Index : 0 Index : 5 Index : 9	Successfully get element	As expected	Pass
T06	get(index) method called when list is empty and has some elements	List Size : 0 Index : -1 Index : 0 Index : 1 List Size : 10 Index : -1 Index : 10	IndexOutOfBoundsException Exception thrown	As expected	Pass
T07	add(index, element) method called when list is empty and has some elements	List size : 0 Index : 0 Element : 'X' List size : 10 Index : 0 Element : 'X' List size : 11 Index : 5 Element : 'Y' List size : 12 Index : 12 Element : 'Z'	Successfully added	As expected	Pass

T08	add(index, element) method called when list is empty and has some elements	List size : 0 Index : -1 Element : 'X' Index : 1 Element : 'Y' List size : 10 Index : -1 Element : 'X' Index : 11 Element : 'Y'	IndexOutOfBoundsException Exception thrown	As expected	Pass
T09	remove(index) method called when list has some elements	List size : 10 Index : 0 List size : 9 Index : 4 List size : 8 Index : 7	Successfully removed	As expected	Pass
T10	remove(index) method called when list is empty and has elements	List size : 0 Index : 0 List size : 10 Index : -1 Index : 10	IndexOutOfBoundsException Exception thrown	As expected	Pass
T11	set(index,element) method called when list has some elements	List size : 10 Index : 0 Element : 'X' Index : 5 Element : 'Y' Index : 9 Element : 'Z'	Successfully setted	As expected	Pass
T12	set(index,element) method called when list is empty and has some elements	List size : 0 Index : 0 Element : 'X' List size : 10 Index : -1 Element : 'X' Index : 10 Element : 'Y'	IndexOutOfBoundsException Exception thrown	As expected	Pass

T13	listIterator(index) method called then hasNext() method called when list is empty and has some elements.	List size : 0 Index : 0 List size : 10 Index : 0 Index : 5 Index : 10	Return false List size : 0 Index : 0 List size : 10 Index : 10 Return true List size : 10 Index : 0 Index : 5	As expected	Pass
T14	listIterator(index) method called then hasPrevious() method called when list is empty and some elements	List size : 0 Index : 0 List size : 10 Index : 0 Index : 5 Index : 10	Return false List size : 0 Index : 0 List size : 10 Index : 0 Return true List size : 10 Index : 5 Index : 10	As expected	Pass
T15	listIterator(index) method called then nextIndex() method called when list has some elements	List size : 10 Index : 5	Return 5	As expected	Pass
T16	listIterator(index) method called then previousIndex() method called when list has some elements	List size : 10 Index : 5	Return 4	As expected	Pass
T17	listIterator(index) method called then next() method called when list has some elements	List size : 10 Index : 0 Index : 5 Index : 9	Return right next values	As expected	Pass
T18	listIterator(index) method called then next() method called when list is empty	List size : 0 Index : 0 List size : 10 Index : 10	NoSuchElementException thrown	As expected	Pass
T19	listIterator(index) method called then previous() method called when list has some elements	List size : 10 Index : 1 Index : 5 Index: 10	Return right previous values	As expected	Pass

T20	listIterator(index) method called then previous() method called when list is empty and has some elements	List size : 0 Index : 0 List size : 10 Index : 0	NoSuchElementException thrown	As expected	Pass
T21	listIterator(index) method called then add(element) method called when list is empty	List size : 0 Element : 'A' Element : 'B' Element : 'C'	Successfully added	As expected	Pass
T22	listIterator(iterIndex) method called then add(index,element) method called when list has some elements	List size : 10 IterIndex : 0 Element : 'X' List size : 11 IterIndex : 5 Element : 'X' List size : 12 IterIndex : 12 Element : 'X'	Successfully added	As expected	Pass
T23	Respectively called listIterator(iterIndex), next() and remove() methods when list has some elements	List size : 10 IterIndex : 0 List size : 9 IterIndex : 4 List size : 8 IterIndex : 7	Successfully removed	As expected	Pass
T24	Respectively called listIterator(iterIndex), previous() and remove() methods when list has some elements	List size : 10 IterIndex : 10 List size : 9 IterIndex : 5 List size : 8 IterIndex : 1	Successfully removed	As expected	Pass
T25	listIterator(iterIndex) method called then remove (index,element) method called before next,previous and add methods when list has some elements	List size : 10 IterIndex : 0 IterIndex : 5 IterIndex : 10	IllegalStateException thrown	As expected	Pass

T26	Respectively called listIterator(iterIndex), next() and set(element) methods when list has some elements	List Size : 10 IterIndex : 0 Element : 'X' IterIndex : 5 Element : 'X' IterIndex : 9 Element : 'X'	Successfully setted	As expected	Pass
T27	listIterator(iterIndex) method called then set (element) method called before next,previous and add methods when list has some elements	List Size : 10 IterIndex : 0 Element : 'X' IterIndex : 5 Element : 'X' IterIndex : 9 Element : 'X'	IlleagleStateException thrown	As expected	Pass
T28	While hasNext method is true, call next method	List size : 10 IterIndex : 0	Successfully printed list from start to end	As expected	Pass
T29	While hasPrevious method is true, call previous method	List size : 10 IterIndex : 10	Successfully printed list from end to start	As expected	Pass

Running and Results

T01 - Creating a character LinkedList

Process is successful.

Size of list : 0 Elements : []

T02 - 'int size()' method

When list is empty

method will be called as list.size()

Size of list : 0 Elements : []

When list has some elements, method will be called as list.size()

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

T03 - 'ListIterator<E> listIterator(int index)' method

When LinkedList is empty, method will be called as list.listIterator(0)

Size of list : 0 Elements : []

List iterator was in 0. index

When LinkedList has some elements

method will be called respectively as list.listIterator(0), list.listIterator(5), list.listIterator(10)

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

List iterator was in 0., 5. and 10. index

T04

When LinkedList is empty
method will be called respectively as list.listIterator(-1), list.listIterator(1).
Expected for IndexOutOfBoundsException to be thrown.
Size of list : 0 Elements : []
Method was called as list.listIterator(-1).
IndexOutOfBoundsException was caught.
Method was called as list.listIterator(1).
IndexOutOfBoundsException was caught.

When LinkedList has some elements
method will be called respectively as list.listIterator(-1), list.listIterator(11).
Expected for IndexOutOfBoundsException to be thrown.
Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]
Method was called as list.listIterator(-1).
IndexOutOfBoundsException was caught.
Method was called as list.listIterator(11).
IndexOutOfBoundsException was caught.

T05 - Testing 'E get(int index)' method

When list have some elements
method will be called respectively as list.get(0), list.get(5), list.get(9)
Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]
0. index : A
5. index : F
9. index : J

T06

When list is empty
method will be called respectively as list.get(-1), list.get(0), list.get(1).
Expected for IndexOutOfBoundsException to be thrown.
Size of list : 0 Elements : []
Method was called as list.get(-1).
IndexOutOfBoundsException was caught.
Method was called as list.get(0).
IndexOutOfBoundsException was caught.
Method was called as list.get(1).
IndexOutOfBoundsException was caught.

When list has some elements
method will be called respectively as list.get(-1), list.get(10).
Expected for IndexOutOfBoundsException to be thrown.
Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]
Method was called as list.get(-1).
IndexOutOfBoundsException was caught.
Method was called as list.get(10).
IndexOutOfBoundsException was caught.

T07 - Testing 'void add(int index, E e)' method

When list is empty

method will be called as list.add(0,'X')

Before adding

Size of list : 0 Elements : []

After adding

Size of list : 1 Elements : [X]

When list has some elements

method will be called respectively as list.add(0,'X'), list.add(5,'Y'), list.add(12,'Z')

Before adding

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

After adding

Size of list : 13 Elements : [X, A, B, C, D, Y, E, F, G, H, I, J, Z]

T08

When list is empty

method will be called respectively as list.add(-1,'X'), list.add(1,'Y')

Expected for IndexOutOfBoundsException to be thrown.

Size of list : 0 Elements : []

Method was called as list.add(-1,'X').

IndexOutOfBoundsException was caught.

Method was called as list.add(1,'Y').

IndexOutOfBoundsException was caught.

When list has some elements

method will be called respectively as list.add(-1,'X'), list.add(11,'Y').

Expected for IndexOutOfBoundsException to be thrown.

Method was called as list.add(-1,'X')

IndexOutOfBoundsException was caught.

Method was called as list.add(11,'Y')

IndexOutOfBoundsException was caught.

T09 - Testing 'E remove(int index)' method

When list has some elements

method will be called respectively as list.remove(0), list.remove(4), list.remove(7)

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

Removed 0. index : A

Size of list : 9 Elements : [B, C, D, E, F, G, H, I, J]

Removed 4. index : F

Size of list : 8 Elements : [B, C, D, E, G, H, I, J]

Removed 7. index : J

Size of list : 7 Elements : [B, C, D, E, G, H, I]

T10

When list is empty

method will be called as `list.remove(0)`.

Expected for `IndexOutOfBoundsException` to be thrown.

Size of list : 0 Elements : []

Method was called as `list.remove(0)`.

`IndexOutOfBoundsException` was caught.

When list has some elements

method will be called respectively as `list.remove(-1)`, `list.remove(10)`.

Expected for `IndexOutOfBoundsException` to be thrown.

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

Method was called as `list.remove(-1)`.

`IndexOutOfBoundsException` was caught.

Method was called as `list.remove(10)`.

`IndexOutOfBoundsException` was caught.

T11 - Testing 'E set(int index, E e)' method

When list has some elements

method was called respectively as `list.set(0,'X')`, `list.set(5,'Y')`, `list.set(9,'Z')`

Before set method :

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

After set method :

Previous 0. index : A

Size of list : 10 Elements : [X, B, C, D, E, F, G, H, I, J]

Before set method :

Size of list : 10 Elements : [X, B, C, D, E, F, G, H, I, J]

After set method :

Previous 5. index : F

Size of list : 10 Elements : [X, B, C, D, E, Y, G, H, I, J]

Before set method :

Size of list : 10 Elements : [X, B, C, D, E, Y, G, H, I, J]

After set method :

Previous 9. index : J

Size of list : 10 Elements : [X, B, C, D, E, Y, G, H, I, Z]

T12

When list is empty

method will be called as `list.set(0,'X')`.

Expected for `IndexOutOfBoundsException` to be thrown.

Size of list : 0 Elements : []

Method was called as `list.set(0,'X')`.

`IndexOutOfBoundsException` was caught.

When list has some elements

method will be called respectively as `list.set(-1,'X')`, `list.set(10,'Y')`.

Expected for `IndexOutOfBoundsException` to be thrown.

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

Method was called as `list.set(-1,'X')`.

`IndexOutOfBoundsException` was caught.

Method was called as `list.set(10,'Y')`.

`IndexOutOfBoundsException` was caught.

T13 - Testing 'boolean hasNext()' method

When list is empty and iterator is at index 0
method will be called as iter.hasNext()

Size of list : 0 Elements : []

has next? false

When list has some elements and iterator respectively is at index 0, 5, 10
method will be called as iter.hasNext().

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

0. index has next? true

5. index has next? true

10. index has next? false

T14 - Testing 'boolean hasPrevious()' method

When list is empty and iterator is at index 0
method will be called as iter.hasPrevious()

Size of list : 0 Elements : []

0. has previous ? false

When list has some elements and iterator is at index 0., 5., 10.
method will be called as iter.hasPrevious()

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

0. has previous ? false

5. has previous ? true

10. has previous ? true

T15 - Testing 'int nextIndex()' method

When list has some elements and iterator is at index 5
method will be called as iter.nextIndex()

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

When iterator is at 5. index, next element's index : 5

T16 - Testing 'int previousIndex()' method

When list has some elements and iterator is at index 5
method will be called as iter.previousIndex()

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

When iterator is at 5. index, previous element's index : 4

T17 - Testing 'E next()' method

When list has some elements, and iterator is at indexes 0., 5. and 9.
method will be called as iter.next()

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

Iterator is at 0. index and value : A

Iterator is at 5. index and value : F

Iterator is at 9. index and value : J

T18

When list is empty and iterator is at index 0
method will be called as iter.next().
Expected NoSuchElementException to be thrown.

Size of list : 0 Elements : []

When iterator is at 0. index and list is empty
method was called as iter.next().
NoSuchElementException was caught.

When list has some elements and iterator is at index 10
method will be called as iter.next().
Expected NoSuchElementException to be thrown.

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

When iterator is at 10. index and list has some elements
method was called as iter.next().
NoSuchElementException was caught.

T19 - Testing 'E previous()' method

When list has some elements and iterator is at respectively index 1., 5. and 10.
method will be as called iter.previous().

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

Iterator was at 1. index and value : A

Iterator was at 5. index and value : E

Iterator was at 10. index and value : J

T20

When list is empty and iterator is at index 0
method will be called as iter.previous().
Expected NoSuchElementException to be thrown.

Size of list : 0 Elements : []

When iterator is at 0. index and list is empty
method was called as iter.previous().
NoSuchElementException was caught.

When list has some elements and iterator is at index 0
method will be called as iter.previous().
Expected for NoSuchElementException to be thrown.

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

When iterator is at 0. index and list has some elements
method was called as iter.previous().
NoSuchElementException was caught.

T21 - 'void add(E e)' method

When list is empty and iterator is at index 0
method will be called consecutive.

Before adding method :

Size of list : 0 Elements : []

After adding method :

Size of list : 3 Elements : [A, B, C]

T22

When list has some elements and iterator is respectively at indexes 0., 5, 12.
method will be called as iter.add().

Before adding method :

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

After adding method :

Added 'X' 0. index of iterator

Size of list : 11 Elements : [X, A, B, C, D, E, F, G, H, I, J]

After adding method :

Added 'Y' 5. index of iterator

Size of list : 12 Elements : [X, A, B, C, D, Y, E, F, G, H, I, J]

After adding method :

Added 'Z' 12. index of iterator

Size of list : 13 Elements : [X, A, B, C, D, Y, E, F, G, H, I, J, Z]

T23 - Testing 'void remove()' method

When list has some elements and iterator is at index 0., 4. and 7.
method will be called first iter.next() then iter.remove()

Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]

Before remove, iterator is at index 0 next item : A

After remove

Size of list : 9 Elements : [B, C, D, E, F, G, H, I, J]

Before remove, iterator is at index 4 next item : F

After remove

Size of list : 8 Elements : [B, C, D, E, G, H, I, J]

Before remove, iterator is at index 7 next item : J

After remove

Size of list : 7 Elements : [B, C, D, E, G, H, I]

T24

When list has some elements and iterator is at index 10., 5. and 1.
method will be called first `iter.previous()` then `iter.remove()`
Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]
Before remove, iterator is at index 10 previous item : J
After remove
Size of list : 9 Elements : [A, B, C, D, E, F, G, H, I]
Before remove, iterator is at index 5 previous item : E
After remove
Size of list : 8 Elements : [A, B, C, D, F, G, H, I]
Before remove, iterator is at index 1 previous item : A
After remove
Size of list : 7 Elements : [B, C, D, F, G, H, I]

T25

When list has some elements and iterator is at index 0, 5, 10
method will be called as `iter.remove()` before `iter.next()`, `iter.previous()` and `iter.add()` methods.
Expected for `IllegalStateException` to be thrown
Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]
When iterator is at index 0 and list has some elements
method was called as `iter.remove()` before `iter.next()`, `iter.previous()` and `iter.add()` methods.
`IllegalStateException` was caught.
When iterator is at index 5 and list has some elements
method was called as `iter.remove()` before `iter.next()`, `iter.previous()` and `iter.add()` methods.
`IllegalStateException` was caught.
When iterator is at index 10 and list has some elements
method was called as `iter.remove()` before `iter.next()`, `iter.previous()` and `iter.add()` methods.
`IllegalStateException` was caught.

T26 - Testing 'void set(E e)' method

When list has some elements and iterator is at index 0., 5 and 9.
`iter.next()` method will be called first and then `iter.set('X')`.
Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]
Before setting
Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]
Iterator is at index 0. `iter.next()` : A
After setting
Size of list : 10 Elements : [X, B, C, D, E, F, G, H, I, J]
Before setting
Size of list : 10 Elements : [X, B, C, D, E, F, G, H, I, J]
Iterator is at index 5. `iter.next()` : F
After setting
Size of list : 10 Elements : [X, B, C, D, E, X, G, H, I, J]
Before setting
Size of list : 10 Elements : [X, B, C, D, E, X, G, H, I, J]
Iterator is at index 9. `iter.next()` : J
After setting
Size of list : 10 Elements : [X, B, C, D, E, X, G, H, I, X]

T27

When list has some elements and iterator is at indexes 0., 5. and 9.
method will be called as `iter.set('X')` method before `next`, `previous` and `add` methods.
Expected for `IllegalStateException` to be thrown.
Size of list : 10 Elements : [A, B, C, D, E, F, G, H, I, J]
When iterator is at index 0 and list has some elements
method was called as `iter.set('X')` before `next`, `previous` and `iter.add` methods.
`IllegalStateException` was caught.
When iterator is at index 5 and list has some elements
method was called as `iter.set('X')` before `next`, `previous` and `iter.add` methods.
`IllegalStateException` was caught.
When iterator is at index 9 and list has some elements
method was called as `iter.set('X')` before `next`, `previous` and `iter.add` methods.
`IllegalStateException` was caught.

T28

When list has some elements and iterator is at indexes 0.
While `iter` has next element, `iter.next()` method will be called
Size of list : 10
The list will be printed on the screen from start to end.
A B C D E F G H I J

T29

When list has some elements and iterator is at indexes 10.
While `iter` has previous element, `iter.previous()` method will be called
Size of list : 10
The list will be printed on the screen from end to start.
J I H G F E D C B A