

JAVA.UTIL.LINKEDLIST CLASS

http://www.tutorialspoint.com/java/util/java_util_linkedlist.htm

Copyright © tutorialspoint.com

Introduction

The **java.util.LinkedList** class operations perform we can expect for a doubly-linked list. Operations that index into the list will traverse the list from the beginning or the end, whichever is closer to the specified index.

Class declaration

Following is the declaration for **java.util.LinkedList** class:

```
public class LinkedList<E>
    extends AbstractSequentialList<E>
    implements List<E>, Deque<E>, Cloneable, Serializable
```

Parameters

Following is the parameter for **java.util.LinkedList** class:

- **E** -- This is the type of elements held in this collection.

Field

Fields inherited from class `java.util.AbstractList`.

Class constructors

S.N.	Constructor & Description
------	---------------------------

- | | |
|---|---|
| 1 | LinkedList

This constructs constructs an empty list. |
| 2 | LinkedListCollection < ? extends E > c

This constructs a list containing the elements of the specified collection, in the order they are returned by the collection's iterator. |

Class methods

S.N.	Method & Description
------	----------------------

- | | |
|---|---|
| 1 | <u>boolean addEe</u>

This method appends the specified element to the end of this list. |
| 2 | <u>void addintindex, Element</u>

This method inserts the specified element at the specified position in this list. |

3

[boolean addAllCollection < ?extendsE > c](#)

This method appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's iterator.

4

[boolean addAllintindex, Collection < ?extendsE > c](#)

This method inserts all of the elements in the specified collection into this list, starting at the specified position.

5

[void addFirstEe](#)

This method returns inserts the specified element at the beginning of this list..

6

[void addLastEe](#)

This method returns appends the specified element to the end of this list.

7

[void clear](#)

This method removes all of the elements from this list.

8

[Object clone](#)

This method returns returns a shallow copy of this LinkedList.

9

[boolean containsObjecto](#)

This method returns true if this list contains the specified element.

10

[Iterator<E> descendingIterator](#)

This method returns an iterator over the elements in this deque in reverse sequential order.

11

[E element](#)

This method retrieves, but does not remove, the head *firstelement* of this list.

12

[E getintindex](#)

This method returns the element at the specified position in this list.

13

[E getFirst](#)

This method returns the first element in this list.

14

[E getLast](#)

This method returns the last element in this list.

15

[int indexOfObject](#)

This method returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.

16

[int lastIndexOfObject](#)

This method returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element.

17

[ListIterator<E> listIterator\(int index\)](#)

This method returns a list-iterator of the elements in this list *in proper sequence*, starting at the specified position in the list.

18

[boolean offer\(E e\)](#)

This method adds the specified element as the tail *last element* of this list.

19

[boolean offerFirst\(E e\)](#)

This method inserts the specified element at the front of this list.

20

[boolean offerLast\(E e\)](#)

This method inserts the specified element at the end of this list.

21

[E peek](#)

This method retrieves, but does not remove, the head *first element* of this list.

22

[E peekFirst](#)

This method retrieves, but does not remove, the first element of this list, or returns null if this list is empty.

23

[E peekLast](#)

This method retrieves, but does not remove, the last element of this list, or returns null if this list is empty.

24

[E poll](#)

This method retrieves and removes the head *first element* of this list.

26

[E pollFirst](#)

This method retrieves and removes the first element of this list, or returns null if this list is empty.

27

[E pollLast](#)

This method retrieves and removes the last element of this list, or returns null if this list is empty.

28

[E pop](#)

This method pops an element from the stack represented by this list.

29

[void pushEe](#)

This method pushes an element onto the stack represented by this list.

30

[E remove](#)

This method retrieves and removes the head *firstelement* of this list.

31

[E removeintindex](#)

This method removes the element at the specified position in this list.

32

[boolean removeObjecto](#)

This method removes the first occurrence of the specified element from this list, if it is present.

33

[E removeFirst](#)

This method removes and returns the first element from this list.

34

[boolean removeFirstOccurrenceObjecto](#)

This method removes the first occurrence of the specified element in this list *whentraversingthelistfromheadtotail*.

35

[E removeLast](#)

This method removes and returns the last element from this list.

36

[boolean removeLastOccurrenceObjecto](#)

This method removes the last occurrence of the specified element in this list *whentraversingthelistfromheadtotail*.

37

[E setindex, Element](#)

This method replaces the element at the specified position in this list with the specified element.

38

[int size](#)

This method returns the number of elements in this list.

39

[Object\[\] toArray](#)

This method returns an array containing all of the elements in this list in proper sequence *from first to last element*.

40

[<T> T\[\] toArray\(T\[\] a\)](#)

This method returns an array containing all of the elements in this list in proper sequence *from first to last element*, the runtime type of the returned array is that of the specified array.

Methods inherited

This class inherits methods from the following classes:

- java.util.AbstractSequentialList
- java.util.AbstractList
- java.util.AbstractCollection
- java.util.Object
- java.util.List
- java.util.Deque

Processing math: 100%