JavaScript is disabled on your browser.

[Skip navigation links](#1fob9te)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/LoopingListIterator.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingIterator.html)
* [Next Class](http://docs.google.com/org/apache/commons/collections4/iterators/NodeListIterator.html)
* [Frames](http://docs.google.com/index.html?org/apache/commons/collections4/iterators/LoopingListIterator.html)
* [No Frames](http://docs.google.com/LoopingListIterator.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#1t3h5sf) |
* [Method](#2s8eyo1)

org.apache.commons.collections4.iterators

## Class LoopingListIterator<E>

* [java.lang.Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)
  + org.apache.commons.collections4.iterators.LoopingListIterator<E>
* All Implemented Interfaces: [Iterator](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true)<E>, [ListIterator](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true)<E>, [OrderedIterator](http://docs.google.com/org/apache/commons/collections4/OrderedIterator.html)<E>, [ResettableIterator](http://docs.google.com/org/apache/commons/collections4/ResettableIterator.html)<E>, [ResettableListIterator](http://docs.google.com/org/apache/commons/collections4/ResettableListIterator.html)<E>  
    
  public class LoopingListIterator<E>  
  extends [Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)  
  implements [ResettableListIterator](http://docs.google.com/org/apache/commons/collections4/ResettableListIterator.html)<E>  
  A ListIterator that restarts when it reaches the end or when it reaches the beginning.  
  The iterator will loop continuously around the provided list, unless there are no elements in the collection to begin with, or all of the elements have been [removed](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#remove--).  
  Concurrent modifications are not directly supported, and for most collection implementations will throw a ConcurrentModificationException.Since: 3.2

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [LoopingListIterator](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#LoopingListIterator-java.util.List-)([List](https://docs.oracle.com/javase/7/docs/api/java/util/List.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> list) Constructor that wraps a list. |

### Method SummaryAll Methods Instance Methods Concrete Methods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + void | * + [add](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#add-E-)([E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html) obj) Inserts the specified element into the underlying list. |
| * + boolean | * + [hasNext](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#hasNext--)() Returns whether this iterator has any more elements. |
| * + boolean | * + [hasPrevious](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#hasPrevious--)() Returns whether this iterator has any more previous elements. |
| * + [E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html) | * + [next](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#next--)() Returns the next object in the list. |
| * + int | * + [nextIndex](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#nextIndex--)() Returns the index of the element that would be returned by a subsequent call to [next()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#next--). |
| * + [E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html) | * + [previous](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#previous--)() Returns the previous object in the list. |
| * + int | * + [previousIndex](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#previousIndex--)() Returns the index of the element that would be returned by a subsequent call to [previous()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#previous--). |
| * + void | * + [remove](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#remove--)() Removes the previously retrieved item from the underlying list. |
| * + void | * + [reset](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#reset--)() Resets the iterator back to the start of the list. |
| * + void | * + [set](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#set-E-)([E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html) obj) Replaces the last element that was returned by [next()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#next--) or [previous()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#previous--). |
| * + int | * + [size](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#size--)() Gets the size of the list underlying the iterator. |

### Methods inherited from class java.lang.[**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)[clone](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#clone--), [equals](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#equals-java.lang.Object-), [finalize](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#finalize--), [getClass](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#getClass--), [hashCode](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#hashCode--), [notify](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#notify--), [notifyAll](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#notifyAll--), [toString](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#toString--), [wait](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#wait--), [wait](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#wait-long-), [wait](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#wait-long-int-)

### Methods inherited from interface java.util.[**Iterator**](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true)[forEachRemaining](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true#forEachRemaining-java.util.function.Consumer-)

### Constructor Detail

#### LoopingListIterator public LoopingListIterator([List](https://docs.oracle.com/javase/7/docs/api/java/util/List.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> list) Constructor that wraps a list. There is no way to reset a ListIterator instance without recreating it from the original source, so the List must be passed in and a reference to it held.Parameters: list - the list to wrap Throws: [NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html?is-external=true) - if the list it null

### Method Detail

#### hasNext public boolean hasNext() Returns whether this iterator has any more elements. Returns false only if the list originally had zero elements, or all elements have been [removed](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#remove--).Specified by: [hasNext](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true#hasNext--) in interface [Iterator](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Specified by: [hasNext](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true#hasNext--) in interface [ListIterator](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Returns: true if there are more elements

#### next public [E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html) next() Returns the next object in the list. If at the end of the list, returns the first element.Specified by: [next](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true#next--) in interface [Iterator](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Specified by: [next](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true#next--) in interface [ListIterator](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Returns: the object after the last element returned Throws: [NoSuchElementException](https://docs.oracle.com/javase/7/docs/api/java/util/NoSuchElementException.html?is-external=true) - if there are no elements in the list

#### nextIndex public int nextIndex() Returns the index of the element that would be returned by a subsequent call to [next()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#next--). As would be expected, if the iterator is at the physical end of the underlying list, 0 is returned, signifying the beginning of the list.Specified by: [nextIndex](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true#nextIndex--) in interface [ListIterator](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Returns: the index of the element that would be returned if next() were called Throws: [NoSuchElementException](https://docs.oracle.com/javase/7/docs/api/java/util/NoSuchElementException.html?is-external=true) - if there are no elements in the list

#### hasPrevious public boolean hasPrevious() Returns whether this iterator has any more previous elements. Returns false only if the list originally had zero elements, or all elements have been [removed](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#remove--).Specified by: [hasPrevious](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true#hasPrevious--) in interface [ListIterator](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Specified by: [hasPrevious](http://docs.google.com/org/apache/commons/collections4/OrderedIterator.html#hasPrevious--) in interface [OrderedIterator](http://docs.google.com/org/apache/commons/collections4/OrderedIterator.html)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Returns: true if there are more elements

#### previous public [E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html) previous() Returns the previous object in the list. If at the beginning of the list, return the last element. Note that in this case, traversal to find that element takes linear time.Specified by: [previous](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true#previous--) in interface [ListIterator](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Specified by: [previous](http://docs.google.com/org/apache/commons/collections4/OrderedIterator.html#previous--) in interface [OrderedIterator](http://docs.google.com/org/apache/commons/collections4/OrderedIterator.html)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Returns: the object before the last element returned Throws: [NoSuchElementException](https://docs.oracle.com/javase/7/docs/api/java/util/NoSuchElementException.html?is-external=true) - if there are no elements in the list

#### previousIndex public int previousIndex() Returns the index of the element that would be returned by a subsequent call to [previous()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#previous--). As would be expected, if at the iterator is at the physical beginning of the underlying list, the list's size minus one is returned, signifying the end of the list.Specified by: [previousIndex](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true#previousIndex--) in interface [ListIterator](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Returns: the index of the element that would be returned if previous() were called Throws: [NoSuchElementException](https://docs.oracle.com/javase/7/docs/api/java/util/NoSuchElementException.html?is-external=true) - if there are no elements in the list

#### remove public void remove() Removes the previously retrieved item from the underlying list. This feature is only supported if the underlying list's [iterator](https://docs.oracle.com/javase/7/docs/api/java/util/List.html?is-external=true#iterator--) method returns an implementation that supports it. This method can only be called after at least one [next()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#next--) or [previous()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#previous--) method call. After a removal, the remove method may not be called again until another [next()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#next--) or [previous()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#previous--) has been performed. If the [reset()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#reset--) is called, then remove may not be called until [next()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#next--) or [previous()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#previous--) is called again.Specified by: [remove](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true#remove--) in interface [Iterator](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Specified by: [remove](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true#remove--) in interface [ListIterator](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Throws: [UnsupportedOperationException](https://docs.oracle.com/javase/7/docs/api/java/lang/UnsupportedOperationException.html?is-external=true) - if the remove method is not supported by the iterator implementation of the underlying list

#### add public void add([E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html) obj) Inserts the specified element into the underlying list. The element is inserted before the next element that would be returned by [next()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#next--), if any, and after the next element that would be returned by [previous()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#previous--), if any. This feature is only supported if the underlying list's [List.listIterator()](https://docs.oracle.com/javase/7/docs/api/java/util/List.html?is-external=true#listIterator--) method returns an implementation that supports it.Specified by: [add](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true#add-E-) in interface [ListIterator](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Parameters: obj - the element to insert Throws: [UnsupportedOperationException](https://docs.oracle.com/javase/7/docs/api/java/lang/UnsupportedOperationException.html?is-external=true) - if the add method is not supported by the iterator implementation of the underlying list

#### set public void set([E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html) obj) Replaces the last element that was returned by [next()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#next--) or [previous()](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html#previous--). This feature is only supported if the underlying list's [List.listIterator()](https://docs.oracle.com/javase/7/docs/api/java/util/List.html?is-external=true#listIterator--) method returns an implementation that supports it.Specified by: [set](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true#set-E-) in interface [ListIterator](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)> Parameters: obj - the element with which to replace the last element returned Throws: [UnsupportedOperationException](https://docs.oracle.com/javase/7/docs/api/java/lang/UnsupportedOperationException.html?is-external=true) - if the set method is not supported by the iterator implementation of the underlying list

#### reset public void reset() Resets the iterator back to the start of the list.Specified by: [reset](http://docs.google.com/org/apache/commons/collections4/ResettableIterator.html#reset--) in interface [ResettableIterator](http://docs.google.com/org/apache/commons/collections4/ResettableIterator.html)<[E](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingListIterator.html)>

#### size public int size() Gets the size of the list underlying the iterator.Returns: the current list size

[Skip navigation links](#2bn6wsx)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/LoopingListIterator.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/org/apache/commons/collections4/iterators/LoopingIterator.html)
* [Next Class](http://docs.google.com/org/apache/commons/collections4/iterators/NodeListIterator.html)
* [Frames](http://docs.google.com/index.html?org/apache/commons/collections4/iterators/LoopingListIterator.html)
* [No Frames](http://docs.google.com/LoopingListIterator.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#1t3h5sf) |
* [Method](#2s8eyo1)

Copyright © 2001–2019 [The Apache Software Foundation](https://www.apache.org/). All rights reserved.