

[Overview](#) [Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

Java™ 2 Platform

Std. Ed. v1.4.2

[PREV CLASS](#) [NEXT CLASS](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

java.util

Interface Iterator

All Known Subinterfaces:[ListIterator](#)**All Known Implementing Classes:**[BeanContextSupport.BCSIterator](#)public interface **Iterator**

An iterator over a collection. Iterator takes the place of Enumeration in the Java collections framework. Iterators differ from enumerations in two ways:

- Iterators allow the caller to remove elements from the underlying collection during the iteration with well-defined semantics.
- Method names have been improved.

This interface is a member of the [Java Collections Framework](#).

Since:

1.2

See Also:[Collection](#), [ListIterator](#), [Enumeration](#)

Method Summary

boolean	hasNext () Returns true if the iteration has more elements.
Object	next () Returns the next element in the iteration.
void	remove () Removes from the underlying collection the last element returned by the iterator (optional operation).

Method Detail

hasNextpublic boolean **hasNext**()

Returns true if the iteration has more elements. (In other words, returns true if next would return an

element rather than throwing an exception.)

Returns:

`true` if the iterator has more elements.

next

```
public Object next()
```

Returns the next element in the iteration.

Returns:

the next element in the iteration.

Throws:

[NoSuchElementException](#) - iteration has no more elements.

remove

```
public void remove()
```

Removes from the underlying collection the last element returned by the iterator (optional operation). This method can be called only once per call to `next`. The behavior of an iterator is unspecified if the underlying collection is modified while the iteration is in progress in any way other than by calling this method.

Throws:

[UnsupportedOperationException](#) - if the `remove` operation is not supported by this Iterator.

[IllegalStateException](#) - if the `next` method has not yet been called, or the `remove` method has already been called after the last call to the `next` method.

[Overview](#) [Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

*Java™ 2 Platform
Std. Ed. v1.4.2*

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Submit a bug or feature](#)

For further API reference and developer documentation, see [Java 2 SDK SE Developer Documentation](#). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2003 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#). Also see the [documentation redistribution policy](#).