| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AbstractQueue.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/AbstractMap.SimpleImmutableEntry.html)   [**NEXT CLASS**](http://docs.google.com/java/util/AbstractSequentialList.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/AbstractQueue.html)    [**NO FRAMES**](http://docs.google.com/AbstractQueue.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#3rdcrjn) |

## **java.util**

Class AbstractQueue<E>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.util.AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<E>  
 **java.util.AbstractQueue<E>**

**Type Parameters:**E - the type of elements held in this collection **All Implemented Interfaces:** [Iterable](http://docs.google.com/java/lang/Iterable.html)<E>, [Collection](http://docs.google.com/java/util/Collection.html)<E>, [Queue](http://docs.google.com/java/util/Queue.html)<E> **Direct Known Subclasses:** [ArrayBlockingQueue](http://docs.google.com/java/util/concurrent/ArrayBlockingQueue.html), [ConcurrentLinkedQueue](http://docs.google.com/java/util/concurrent/ConcurrentLinkedQueue.html), [DelayQueue](http://docs.google.com/java/util/concurrent/DelayQueue.html), [LinkedBlockingDeque](http://docs.google.com/java/util/concurrent/LinkedBlockingDeque.html), [LinkedBlockingQueue](http://docs.google.com/java/util/concurrent/LinkedBlockingQueue.html), [PriorityBlockingQueue](http://docs.google.com/java/util/concurrent/PriorityBlockingQueue.html), [PriorityQueue](http://docs.google.com/java/util/PriorityQueue.html), [SynchronousQueue](http://docs.google.com/java/util/concurrent/SynchronousQueue.html)

public abstract class **AbstractQueue<E>**extends [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<E>implements [Queue](http://docs.google.com/java/util/Queue.html)<E>

This class provides skeletal implementations of some [Queue](http://docs.google.com/java/util/Queue.html) operations. The implementations in this class are appropriate when the base implementation does *not* allow null elements. Methods [add](http://docs.google.com/java/util/AbstractQueue.html#add(E)), [remove](http://docs.google.com/java/util/AbstractQueue.html#remove()), and [element](http://docs.google.com/java/util/AbstractQueue.html#element()) are based on [offer](http://docs.google.com/java/util/Queue.html#offer(E)), [poll](http://docs.google.com/java/util/Queue.html#poll()), and [peek](http://docs.google.com/java/util/Queue.html#peek()), respectively but throw exceptions instead of indicating failure via false or null returns.

A Queue implementation that extends this class must minimally define a method [Queue.offer(E)](http://docs.google.com/java/util/Queue.html#offer(E)) which does not permit insertion of null elements, along with methods [Queue.peek()](http://docs.google.com/java/util/Queue.html#peek()), [Queue.poll()](http://docs.google.com/java/util/Queue.html#poll()), [Collection.size()](http://docs.google.com/java/util/Collection.html#size()), and a [Collection.iterator()](http://docs.google.com/java/util/Collection.html#iterator()) supporting [Iterator.remove()](http://docs.google.com/java/util/Iterator.html#remove()). Typically, additional methods will be overridden as well. If these requirements cannot be met, consider instead subclassing [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html).

This class is a member of the  [Java Collections Framework](http://docs.google.com/technotes/guides/collections/index.html).

**Since:** 1.5

| **Constructor Summary** | |
| --- | --- |
| protected | [**AbstractQueue**](http://docs.google.com/java/util/AbstractQueue.html#AbstractQueue())()            Constructor for use by subclasses. |

| **Method Summary** | |
| --- | --- |
| boolean | [**add**](http://docs.google.com/java/util/AbstractQueue.html#add(E))([E](http://docs.google.com/java/util/AbstractQueue.html) e)            Inserts the specified element into this queue if it is possible to do so immediately without violating capacity restrictions, returning true upon success and throwing an IllegalStateException if no space is currently available. |
| boolean | [**addAll**](http://docs.google.com/java/util/AbstractQueue.html#addAll(java.util.Collection))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [E](http://docs.google.com/java/util/AbstractQueue.html)> c)            Adds all of the elements in the specified collection to this queue. |
| void | [**clear**](http://docs.google.com/java/util/AbstractQueue.html#clear())()            Removes all of the elements from this queue. |
| [E](http://docs.google.com/java/util/AbstractQueue.html) | [**element**](http://docs.google.com/java/util/AbstractQueue.html#element())()            Retrieves, but does not remove, the head of this queue. |
| [E](http://docs.google.com/java/util/AbstractQueue.html) | [**remove**](http://docs.google.com/java/util/AbstractQueue.html#remove())()            Retrieves and removes the head of this queue. |

| **Methods inherited from class java.util.**[**AbstractCollection**](http://docs.google.com/java/util/AbstractCollection.html) |
| --- |
| [contains](http://docs.google.com/java/util/AbstractCollection.html#contains(java.lang.Object)), [containsAll](http://docs.google.com/java/util/AbstractCollection.html#containsAll(java.util.Collection)), [isEmpty](http://docs.google.com/java/util/AbstractCollection.html#isEmpty()), [iterator](http://docs.google.com/java/util/AbstractCollection.html#iterator()), [remove](http://docs.google.com/java/util/AbstractCollection.html#remove(java.lang.Object)), [removeAll](http://docs.google.com/java/util/AbstractCollection.html#removeAll(java.util.Collection)), [retainAll](http://docs.google.com/java/util/AbstractCollection.html#retainAll(java.util.Collection)), [size](http://docs.google.com/java/util/AbstractCollection.html#size()), [toArray](http://docs.google.com/java/util/AbstractCollection.html#toArray()), [toArray](http://docs.google.com/java/util/AbstractCollection.html#toArray(T%5B%5D)), [toString](http://docs.google.com/java/util/AbstractCollection.html#toString()) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Methods inherited from interface java.util.**[**Queue**](http://docs.google.com/java/util/Queue.html) |
| --- |
| [offer](http://docs.google.com/java/util/Queue.html#offer(E)), [peek](http://docs.google.com/java/util/Queue.html#peek()), [poll](http://docs.google.com/java/util/Queue.html#poll()) |

| **Methods inherited from interface java.util.**[**Collection**](http://docs.google.com/java/util/Collection.html) |
| --- |
| [contains](http://docs.google.com/java/util/Collection.html#contains(java.lang.Object)), [containsAll](http://docs.google.com/java/util/Collection.html#containsAll(java.util.Collection)), [equals](http://docs.google.com/java/util/Collection.html#equals(java.lang.Object)), [hashCode](http://docs.google.com/java/util/Collection.html#hashCode()), [isEmpty](http://docs.google.com/java/util/Collection.html#isEmpty()), [iterator](http://docs.google.com/java/util/Collection.html#iterator()), [remove](http://docs.google.com/java/util/Collection.html#remove(java.lang.Object)), [removeAll](http://docs.google.com/java/util/Collection.html#removeAll(java.util.Collection)), [retainAll](http://docs.google.com/java/util/Collection.html#retainAll(java.util.Collection)), [size](http://docs.google.com/java/util/Collection.html#size()), [toArray](http://docs.google.com/java/util/Collection.html#toArray()), [toArray](http://docs.google.com/java/util/Collection.html#toArray(T%5B%5D)) |

| **Constructor Detail** |
| --- |

### AbstractQueue

protected **AbstractQueue**()

Constructor for use by subclasses.

| **Method Detail** |
| --- |

### add

public boolean **add**([E](http://docs.google.com/java/util/AbstractQueue.html) e)

Inserts the specified element into this queue if it is possible to do so immediately without violating capacity restrictions, returning true upon success and throwing an IllegalStateException if no space is currently available.

This implementation returns true if offer succeeds, else throws an IllegalStateException.

**Specified by:**[add](http://docs.google.com/java/util/Collection.html#add(E)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/AbstractQueue.html)>**Specified by:**[add](http://docs.google.com/java/util/Queue.html#add(E)) in interface [Queue](http://docs.google.com/java/util/Queue.html)<[E](http://docs.google.com/java/util/AbstractQueue.html)>**Overrides:**[add](http://docs.google.com/java/util/AbstractCollection.html#add(E)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/AbstractQueue.html)> **Parameters:**e - the element to add **Returns:**true (as specified by [Collection.add(E)](http://docs.google.com/java/util/Collection.html#add(E))) **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if the element cannot be added at this time due to capacity restrictions [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the class of the specified element prevents it from being added to this queue [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified element is null and this queue does not permit null elements [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if some property of this element prevents it from being added to this queue

### remove

public [E](http://docs.google.com/java/util/AbstractQueue.html) **remove**()

Retrieves and removes the head of this queue. This method differs from [poll](http://docs.google.com/java/util/Queue.html#poll()) only in that it throws an exception if this queue is empty.

This implementation returns the result of poll unless the queue is empty.

**Specified by:**[remove](http://docs.google.com/java/util/Queue.html#remove()) in interface [Queue](http://docs.google.com/java/util/Queue.html)<[E](http://docs.google.com/java/util/AbstractQueue.html)> **Returns:**the head of this queue **Throws:** [NoSuchElementException](http://docs.google.com/java/util/NoSuchElementException.html) - if this queue is empty

### element

public [E](http://docs.google.com/java/util/AbstractQueue.html) **element**()

Retrieves, but does not remove, the head of this queue. This method differs from [peek](http://docs.google.com/java/util/Queue.html#peek()) only in that it throws an exception if this queue is empty.

This implementation returns the result of peek unless the queue is empty.

**Specified by:**[element](http://docs.google.com/java/util/Queue.html#element()) in interface [Queue](http://docs.google.com/java/util/Queue.html)<[E](http://docs.google.com/java/util/AbstractQueue.html)> **Returns:**the head of this queue **Throws:** [NoSuchElementException](http://docs.google.com/java/util/NoSuchElementException.html) - if this queue is empty

### clear

public void **clear**()

Removes all of the elements from this queue. The queue will be empty after this call returns.

This implementation repeatedly invokes [poll](http://docs.google.com/java/util/Queue.html#poll()) until it returns null.

**Specified by:**[clear](http://docs.google.com/java/util/Collection.html#clear()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/AbstractQueue.html)>**Overrides:**[clear](http://docs.google.com/java/util/AbstractCollection.html#clear()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/AbstractQueue.html)>

### addAll

public boolean **addAll**([Collection](http://docs.google.com/java/util/Collection.html)<? extends [E](http://docs.google.com/java/util/AbstractQueue.html)> c)

Adds all of the elements in the specified collection to this queue. Attempts to addAll of a queue to itself result in IllegalArgumentException. Further, the behavior of this operation is undefined if the specified collection is modified while the operation is in progress.

This implementation iterates over the specified collection, and adds each element returned by the iterator to this queue, in turn. A runtime exception encountered while trying to add an element (including, in particular, a null element) may result in only some of the elements having been successfully added when the associated exception is thrown.

**Specified by:**[addAll](http://docs.google.com/java/util/Collection.html#addAll(java.util.Collection)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/AbstractQueue.html)>**Overrides:**[addAll](http://docs.google.com/java/util/AbstractCollection.html#addAll(java.util.Collection)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/AbstractQueue.html)> **Parameters:**c - collection containing elements to be added to this queue **Returns:**true if this queue changed as a result of the call **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the class of an element of the specified collection prevents it from being added to this queue [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified collection contains a null element and this queue does not permit null elements, or if the specified collection is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if some property of an element of the specified collection prevents it from being added to this queue, or if the specified collection is this queue [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if not all the elements can be added at this time due to insertion restrictions**See Also:**[add(Object)](http://docs.google.com/java/util/AbstractQueue.html#add(E))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AbstractQueue.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/AbstractMap.SimpleImmutableEntry.html)   [**NEXT CLASS**](http://docs.google.com/java/util/AbstractSequentialList.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/AbstractQueue.html)    [**NO FRAMES**](http://docs.google.com/AbstractQueue.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#3rdcrjn) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).