| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ArrayDeque.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/AbstractSet.html)   [**NEXT CLASS**](http://docs.google.com/java/util/ArrayList.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/ArrayDeque.html)    [**NO FRAMES**](http://docs.google.com/ArrayDeque.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#26in1rg) |

## **java.util**

Class ArrayDeque<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.ArrayDeque<E>**

**Type Parameters:**E - the type of elements held in this collection **All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Iterable](http://docs.google.com/java/lang/Iterable.html)<E>, [Collection](http://docs.google.com/java/util/Collection.html)<E>, [Deque](http://docs.google.com/java/util/Deque.html)<E>, [Queue](http://docs.google.com/java/util/Queue.html)<E>

public class **ArrayDeque<E>**extends [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<E>implements [Deque](http://docs.google.com/java/util/Deque.html)<E>, [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

Resizable-array implementation of the [Deque](http://docs.google.com/java/util/Deque.html) interface. Array deques have no capacity restrictions; they grow as necessary to support usage. They are not thread-safe; in the absence of external synchronization, they do not support concurrent access by multiple threads. Null elements are prohibited. This class is likely to be faster than [Stack](http://docs.google.com/java/util/Stack.html) when used as a stack, and faster than [LinkedList](http://docs.google.com/java/util/LinkedList.html) when used as a queue.

Most ArrayDeque operations run in amortized constant time. Exceptions include [remove](http://docs.google.com/java/util/ArrayDeque.html#remove(java.lang.Object)), [removeFirstOccurrence](http://docs.google.com/java/util/ArrayDeque.html#removeFirstOccurrence(java.lang.Object)), [removeLastOccurrence](http://docs.google.com/java/util/ArrayDeque.html#removeLastOccurrence(java.lang.Object)), [contains](http://docs.google.com/java/util/ArrayDeque.html#contains(java.lang.Object)), [iterator.remove()](http://docs.google.com/java/util/ArrayDeque.html#iterator()), and the bulk operations, all of which run in linear time.

The iterators returned by this class's iterator method are *fail-fast*: If the deque is modified at any time after the iterator is created, in any way except through the iterator's own remove method, the iterator will generally throw a [ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html). Thus, in the face of concurrent modification, the iterator fails quickly and cleanly, rather than risking arbitrary, non-deterministic behavior at an undetermined time in the future.

Note that the fail-fast behavior of an iterator cannot be guaranteed as it is, generally speaking, impossible to make any hard guarantees in the presence of unsynchronized concurrent modification. Fail-fast iterators throw ConcurrentModificationException on a best-effort basis. Therefore, it would be wrong to write a program that depended on this exception for its correctness: *the fail-fast behavior of iterators should be used only to detect bugs.*

This class and its iterator implement all of the *optional* methods of the [Collection](http://docs.google.com/java/util/Collection.html) and [Iterator](http://docs.google.com/java/util/Iterator.html) interfaces.

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

**Since:** 1.6 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.util.ArrayDeque)

| **Constructor Summary** | |
| --- | --- |
| [**ArrayDeque**](http://docs.google.com/java/util/ArrayDeque.html#ArrayDeque())()            Constructs an empty array deque with an initial capacity sufficient to hold 16 elements. |
| [**ArrayDeque**](http://docs.google.com/java/util/ArrayDeque.html#ArrayDeque(java.util.Collection))([Collection](http://docs.google.com/java/util/Collection.html)<? extends [E](http://docs.google.com/java/util/ArrayDeque.html)> c)            Constructs a deque containing the elements of the specified collection, in the order they are returned by the collection's iterator. |
| [**ArrayDeque**](http://docs.google.com/java/util/ArrayDeque.html#ArrayDeque(int))(int numElements)            Constructs an empty array deque with an initial capacity sufficient to hold the specified number of elements. |

| **Method Summary** | |
| --- | --- |
| boolean | [**add**](http://docs.google.com/java/util/ArrayDeque.html#add(E))([E](http://docs.google.com/java/util/ArrayDeque.html) e)            Inserts the specified element at the end of this deque. |
| void | [**addFirst**](http://docs.google.com/java/util/ArrayDeque.html#addFirst(E))([E](http://docs.google.com/java/util/ArrayDeque.html) e)            Inserts the specified element at the front of this deque. |
| void | [**addLast**](http://docs.google.com/java/util/ArrayDeque.html#addLast(E))([E](http://docs.google.com/java/util/ArrayDeque.html) e)            Inserts the specified element at the end of this deque. |
| void | [**clear**](http://docs.google.com/java/util/ArrayDeque.html#clear())()            Removes all of the elements from this deque. |
| [ArrayDeque](http://docs.google.com/java/util/ArrayDeque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> | [**clone**](http://docs.google.com/java/util/ArrayDeque.html#clone())()            Returns a copy of this deque. |
| boolean | [**contains**](http://docs.google.com/java/util/ArrayDeque.html#contains(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Returns true if this deque contains the specified element. |
| [Iterator](http://docs.google.com/java/util/Iterator.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> | [**descendingIterator**](http://docs.google.com/java/util/ArrayDeque.html#descendingIterator())()            Returns an iterator over the elements in this deque in reverse sequential order. |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**element**](http://docs.google.com/java/util/ArrayDeque.html#element())()            Retrieves, but does not remove, the head of the queue represented by this deque. |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**getFirst**](http://docs.google.com/java/util/ArrayDeque.html#getFirst())()            Retrieves, but does not remove, the first element of this deque. |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**getLast**](http://docs.google.com/java/util/ArrayDeque.html#getLast())()            Retrieves, but does not remove, the last element of this deque. |
| boolean | [**isEmpty**](http://docs.google.com/java/util/ArrayDeque.html#isEmpty())()            Returns true if this deque contains no elements. |
| [Iterator](http://docs.google.com/java/util/Iterator.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> | [**iterator**](http://docs.google.com/java/util/ArrayDeque.html#iterator())()            Returns an iterator over the elements in this deque. |
| boolean | [**offer**](http://docs.google.com/java/util/ArrayDeque.html#offer(E))([E](http://docs.google.com/java/util/ArrayDeque.html) e)            Inserts the specified element at the end of this deque. |
| boolean | [**offerFirst**](http://docs.google.com/java/util/ArrayDeque.html#offerFirst(E))([E](http://docs.google.com/java/util/ArrayDeque.html) e)            Inserts the specified element at the front of this deque. |
| boolean | [**offerLast**](http://docs.google.com/java/util/ArrayDeque.html#offerLast(E))([E](http://docs.google.com/java/util/ArrayDeque.html) e)            Inserts the specified element at the end of this deque. |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**peek**](http://docs.google.com/java/util/ArrayDeque.html#peek())()            Retrieves, but does not remove, the head of the queue represented by this deque, or returns null if this deque is empty. |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**peekFirst**](http://docs.google.com/java/util/ArrayDeque.html#peekFirst())()            Retrieves, but does not remove, the first element of this deque, or returns null if this deque is empty. |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**peekLast**](http://docs.google.com/java/util/ArrayDeque.html#peekLast())()            Retrieves, but does not remove, the last element of this deque, or returns null if this deque is empty. |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**poll**](http://docs.google.com/java/util/ArrayDeque.html#poll())()            Retrieves and removes the head of the queue represented by this deque (in other words, the first element of this deque), or returns null if this deque is empty. |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**pollFirst**](http://docs.google.com/java/util/ArrayDeque.html#pollFirst())()            Retrieves and removes the first element of this deque, or returns null if this deque is empty. |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**pollLast**](http://docs.google.com/java/util/ArrayDeque.html#pollLast())()            Retrieves and removes the last element of this deque, or returns null if this deque is empty. |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**pop**](http://docs.google.com/java/util/ArrayDeque.html#pop())()            Pops an element from the stack represented by this deque. |
| void | [**push**](http://docs.google.com/java/util/ArrayDeque.html#push(E))([E](http://docs.google.com/java/util/ArrayDeque.html) e)            Pushes an element onto the stack represented by this deque. |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**remove**](http://docs.google.com/java/util/ArrayDeque.html#remove())()            Retrieves and removes the head of the queue represented by this deque. |
| boolean | [**remove**](http://docs.google.com/java/util/ArrayDeque.html#remove(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Removes a single instance of the specified element from this deque. |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**removeFirst**](http://docs.google.com/java/util/ArrayDeque.html#removeFirst())()            Retrieves and removes the first element of this deque. |
| boolean | [**removeFirstOccurrence**](http://docs.google.com/java/util/ArrayDeque.html#removeFirstOccurrence(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Removes the first occurrence of the specified element in this deque (when traversing the deque from head to tail). |
| [E](http://docs.google.com/java/util/ArrayDeque.html) | [**removeLast**](http://docs.google.com/java/util/ArrayDeque.html#removeLast())()            Retrieves and removes the last element of this deque. |
| boolean | [**removeLastOccurrence**](http://docs.google.com/java/util/ArrayDeque.html#removeLastOccurrence(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Removes the last occurrence of the specified element in this deque (when traversing the deque from head to tail). |
| int | [**size**](http://docs.google.com/java/util/ArrayDeque.html#size())()            Returns the number of elements in this deque. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**toArray**](http://docs.google.com/java/util/ArrayDeque.html#toArray())()            Returns an array containing all of the elements in this deque in proper sequence (from first to last element). |
| | <T> T[] | | --- | | [**toArray**](http://docs.google.com/java/util/ArrayDeque.html#toArray(T%5B%5D))(T[] a)            Returns an array containing all of the elements in this deque in proper sequence (from first to last element); the runtime type of the returned array is that of the specified array. |

| **Methods inherited from class java.util.**[**AbstractCollection**](http://docs.google.com/java/util/AbstractCollection.html) |
| --- |
| [addAll](http://docs.google.com/java/util/AbstractCollection.html#addAll(java.util.Collection)), [containsAll](http://docs.google.com/java/util/AbstractCollection.html#containsAll(java.util.Collection)), [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)), [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) |
| --- |
| [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.**[**Collection**](http://docs.google.com/java/util/Collection.html) |
| --- |
| [addAll](http://docs.google.com/java/util/Collection.html#addAll(java.util.Collection)), [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()), [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)) |

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

### ArrayDeque

public **ArrayDeque**()

Constructs an empty array deque with an initial capacity sufficient to hold 16 elements.

### ArrayDeque

public **ArrayDeque**(int numElements)

Constructs an empty array deque with an initial capacity sufficient to hold the specified number of elements.

**Parameters:**numElements - lower bound on initial capacity of the deque

### ArrayDeque

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

Constructs a deque containing the elements of the specified collection, in the order they are returned by the collection's iterator. (The first element returned by the collection's iterator becomes the first element, or *front* of the deque.)

**Parameters:**c - the collection whose elements are to be placed into the deque **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified collection is null

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

### addFirst

public void **addFirst**([E](http://docs.google.com/java/util/ArrayDeque.html) e)

Inserts the specified element at the front of this deque.

**Specified by:**[addFirst](http://docs.google.com/java/util/Deque.html#addFirst(E)) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Parameters:**e - the element to add **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified element is null

### addLast

public void **addLast**([E](http://docs.google.com/java/util/ArrayDeque.html) e)

Inserts the specified element at the end of this deque.

This method is equivalent to [add(E)](http://docs.google.com/java/util/ArrayDeque.html#add(E)).

**Specified by:**[addLast](http://docs.google.com/java/util/Deque.html#addLast(E)) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Parameters:**e - the element to add **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified element is null

### offerFirst

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

Inserts the specified element at the front of this deque.

**Specified by:**[offerFirst](http://docs.google.com/java/util/Deque.html#offerFirst(E)) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Parameters:**e - the element to add **Returns:**true (as specified by [Deque.offerFirst(E)](http://docs.google.com/java/util/Deque.html#offerFirst(E))) **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified element is null

### offerLast

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

Inserts the specified element at the end of this deque.

**Specified by:**[offerLast](http://docs.google.com/java/util/Deque.html#offerLast(E)) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Parameters:**e - the element to add **Returns:**true (as specified by [Deque.offerLast(E)](http://docs.google.com/java/util/Deque.html#offerLast(E))) **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified element is null

### removeFirst

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

**Description copied from interface:** [**Deque**](http://docs.google.com/java/util/Deque.html#removeFirst()) Retrieves and removes the first element of this deque. This method differs from [pollFirst](http://docs.google.com/java/util/Deque.html#pollFirst()) only in that it throws an exception if this deque is empty.

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

### removeLast

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

**Description copied from interface:** [**Deque**](http://docs.google.com/java/util/Deque.html#removeLast()) Retrieves and removes the last element of this deque. This method differs from [pollLast](http://docs.google.com/java/util/Deque.html#pollLast()) only in that it throws an exception if this deque is empty.

**Specified by:**[removeLast](http://docs.google.com/java/util/Deque.html#removeLast()) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**the tail of this deque **Throws:** [NoSuchElementException](http://docs.google.com/java/util/NoSuchElementException.html) - if this deque is empty

### pollFirst

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

**Description copied from interface:** [**Deque**](http://docs.google.com/java/util/Deque.html#pollFirst()) Retrieves and removes the first element of this deque, or returns null if this deque is empty.

**Specified by:**[pollFirst](http://docs.google.com/java/util/Deque.html#pollFirst()) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**the head of this deque, or null if this deque is empty

### pollLast

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

**Description copied from interface:** [**Deque**](http://docs.google.com/java/util/Deque.html#pollLast()) Retrieves and removes the last element of this deque, or returns null if this deque is empty.

**Specified by:**[pollLast](http://docs.google.com/java/util/Deque.html#pollLast()) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**the tail of this deque, or null if this deque is empty

### getFirst

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

**Description copied from interface:** [**Deque**](http://docs.google.com/java/util/Deque.html#getFirst()) Retrieves, but does not remove, the first element of this deque. This method differs from [peekFirst](http://docs.google.com/java/util/Deque.html#peekFirst()) only in that it throws an exception if this deque is empty.

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

### getLast

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

**Description copied from interface:** [**Deque**](http://docs.google.com/java/util/Deque.html#getLast()) Retrieves, but does not remove, the last element of this deque. This method differs from [peekLast](http://docs.google.com/java/util/Deque.html#peekLast()) only in that it throws an exception if this deque is empty.

**Specified by:**[getLast](http://docs.google.com/java/util/Deque.html#getLast()) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**the tail of this deque **Throws:** [NoSuchElementException](http://docs.google.com/java/util/NoSuchElementException.html) - if this deque is empty

### peekFirst

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

**Description copied from interface:** [**Deque**](http://docs.google.com/java/util/Deque.html#peekFirst()) Retrieves, but does not remove, the first element of this deque, or returns null if this deque is empty.

**Specified by:**[peekFirst](http://docs.google.com/java/util/Deque.html#peekFirst()) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**the head of this deque, or null if this deque is empty

### peekLast

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

**Description copied from interface:** [**Deque**](http://docs.google.com/java/util/Deque.html#peekLast()) Retrieves, but does not remove, the last element of this deque, or returns null if this deque is empty.

**Specified by:**[peekLast](http://docs.google.com/java/util/Deque.html#peekLast()) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**the tail of this deque, or null if this deque is empty

### removeFirstOccurrence

public boolean **removeFirstOccurrence**([Object](http://docs.google.com/java/lang/Object.html) o)

Removes the first occurrence of the specified element in this deque (when traversing the deque from head to tail). If the deque does not contain the element, it is unchanged. More formally, removes the first element e such that o.equals(e) (if such an element exists). Returns true if this deque contained the specified element (or equivalently, if this deque changed as a result of the call).

**Specified by:**[removeFirstOccurrence](http://docs.google.com/java/util/Deque.html#removeFirstOccurrence(java.lang.Object)) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Parameters:**o - element to be removed from this deque, if present **Returns:**true if the deque contained the specified element

### removeLastOccurrence

public boolean **removeLastOccurrence**([Object](http://docs.google.com/java/lang/Object.html) o)

Removes the last occurrence of the specified element in this deque (when traversing the deque from head to tail). If the deque does not contain the element, it is unchanged. More formally, removes the last element e such that o.equals(e) (if such an element exists). Returns true if this deque contained the specified element (or equivalently, if this deque changed as a result of the call).

**Specified by:**[removeLastOccurrence](http://docs.google.com/java/util/Deque.html#removeLastOccurrence(java.lang.Object)) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Parameters:**o - element to be removed from this deque, if present **Returns:**true if the deque contained the specified element

### add

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

Inserts the specified element at the end of this deque.

This method is equivalent to [addLast(E)](http://docs.google.com/java/util/ArrayDeque.html#addLast(E)).

**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/ArrayDeque.html)>**Specified by:**[add](http://docs.google.com/java/util/Deque.html#add(E)) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.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/ArrayDeque.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/ArrayDeque.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:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified element is null

### offer

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

Inserts the specified element at the end of this deque.

This method is equivalent to [offerLast(E)](http://docs.google.com/java/util/ArrayDeque.html#offerLast(E)).

**Specified by:**[offer](http://docs.google.com/java/util/Deque.html#offer(E)) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Specified by:**[offer](http://docs.google.com/java/util/Queue.html#offer(E)) in interface [Queue](http://docs.google.com/java/util/Queue.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Parameters:**e - the element to add **Returns:**true (as specified by [Queue.offer(E)](http://docs.google.com/java/util/Queue.html#offer(E))) **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified element is null

### remove

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

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

This method is equivalent to [removeFirst()](http://docs.google.com/java/util/ArrayDeque.html#removeFirst()).

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

### poll

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

Retrieves and removes the head of the queue represented by this deque (in other words, the first element of this deque), or returns null if this deque is empty.

This method is equivalent to [pollFirst()](http://docs.google.com/java/util/ArrayDeque.html#pollFirst()).

**Specified by:**[poll](http://docs.google.com/java/util/Deque.html#poll()) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Specified by:**[poll](http://docs.google.com/java/util/Queue.html#poll()) in interface [Queue](http://docs.google.com/java/util/Queue.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**the head of the queue represented by this deque, or null if this deque is empty

### element

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

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

This method is equivalent to [getFirst()](http://docs.google.com/java/util/ArrayDeque.html#getFirst()).

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

### peek

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

Retrieves, but does not remove, the head of the queue represented by this deque, or returns null if this deque is empty.

This method is equivalent to [peekFirst()](http://docs.google.com/java/util/ArrayDeque.html#peekFirst()).

**Specified by:**[peek](http://docs.google.com/java/util/Deque.html#peek()) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Specified by:**[peek](http://docs.google.com/java/util/Queue.html#peek()) in interface [Queue](http://docs.google.com/java/util/Queue.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**the head of the queue represented by this deque, or null if this deque is empty

### push

public void **push**([E](http://docs.google.com/java/util/ArrayDeque.html) e)

Pushes an element onto the stack represented by this deque. In other words, inserts the element at the front of this deque.

This method is equivalent to [addFirst(E)](http://docs.google.com/java/util/ArrayDeque.html#addFirst(E)).

**Specified by:**[push](http://docs.google.com/java/util/Deque.html#push(E)) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Parameters:**e - the element to push **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified element is null

### pop

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

Pops an element from the stack represented by this deque. In other words, removes and returns the first element of this deque.

This method is equivalent to [removeFirst()](http://docs.google.com/java/util/ArrayDeque.html#removeFirst()).

**Specified by:**[pop](http://docs.google.com/java/util/Deque.html#pop()) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**the element at the front of this deque (which is the top of the stack represented by this deque) **Throws:** [NoSuchElementException](http://docs.google.com/java/util/NoSuchElementException.html) - if this deque is empty

### size

public int **size**()

Returns the number of elements in this deque.

**Specified by:**[size](http://docs.google.com/java/util/Collection.html#size()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Specified by:**[size](http://docs.google.com/java/util/Deque.html#size()) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Specified by:**[size](http://docs.google.com/java/util/AbstractCollection.html#size()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**the number of elements in this deque

### isEmpty

public boolean **isEmpty**()

Returns true if this deque contains no elements.

**Specified by:**[isEmpty](http://docs.google.com/java/util/Collection.html#isEmpty()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Overrides:**[isEmpty](http://docs.google.com/java/util/AbstractCollection.html#isEmpty()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**true if this deque contains no elements

### iterator

public [Iterator](http://docs.google.com/java/util/Iterator.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **iterator**()

Returns an iterator over the elements in this deque. The elements will be ordered from first (head) to last (tail). This is the same order that elements would be dequeued (via successive calls to [remove()](http://docs.google.com/java/util/ArrayDeque.html#remove()) or popped (via successive calls to [pop()](http://docs.google.com/java/util/ArrayDeque.html#pop())).

**Specified by:**[iterator](http://docs.google.com/java/lang/Iterable.html#iterator()) in interface [Iterable](http://docs.google.com/java/lang/Iterable.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Specified by:**[iterator](http://docs.google.com/java/util/Collection.html#iterator()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Specified by:**[iterator](http://docs.google.com/java/util/Deque.html#iterator()) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Specified by:**[iterator](http://docs.google.com/java/util/AbstractCollection.html#iterator()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**an iterator over the elements in this deque

### descendingIterator

public [Iterator](http://docs.google.com/java/util/Iterator.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **descendingIterator**()

**Description copied from interface:** [**Deque**](http://docs.google.com/java/util/Deque.html#descendingIterator()) Returns an iterator over the elements in this deque in reverse sequential order. The elements will be returned in order from last (tail) to first (head).

**Specified by:**[descendingIterator](http://docs.google.com/java/util/Deque.html#descendingIterator()) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**an iterator over the elements in this deque in reverse sequence

### contains

public boolean **contains**([Object](http://docs.google.com/java/lang/Object.html) o)

Returns true if this deque contains the specified element. More formally, returns true if and only if this deque contains at least one element e such that o.equals(e).

**Specified by:**[contains](http://docs.google.com/java/util/Collection.html#contains(java.lang.Object)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Specified by:**[contains](http://docs.google.com/java/util/Deque.html#contains(java.lang.Object)) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Overrides:**[contains](http://docs.google.com/java/util/AbstractCollection.html#contains(java.lang.Object)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Parameters:**o - object to be checked for containment in this deque **Returns:**true if this deque contains the specified element

### remove

public boolean **remove**([Object](http://docs.google.com/java/lang/Object.html) o)

Removes a single instance of the specified element from this deque. If the deque does not contain the element, it is unchanged. More formally, removes the first element e such that o.equals(e) (if such an element exists). Returns true if this deque contained the specified element (or equivalently, if this deque changed as a result of the call).

This method is equivalent to [removeFirstOccurrence(java.lang.Object)](http://docs.google.com/java/util/ArrayDeque.html#removeFirstOccurrence(java.lang.Object)).

**Specified by:**[remove](http://docs.google.com/java/util/Collection.html#remove(java.lang.Object)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Specified by:**[remove](http://docs.google.com/java/util/Deque.html#remove(java.lang.Object)) in interface [Deque](http://docs.google.com/java/util/Deque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Overrides:**[remove](http://docs.google.com/java/util/AbstractCollection.html#remove(java.lang.Object)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Parameters:**o - element to be removed from this deque, if present **Returns:**true if this deque contained the specified element

### clear

public void **clear**()

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

**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/ArrayDeque.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/ArrayDeque.html)>

### toArray

public [Object](http://docs.google.com/java/lang/Object.html)[] **toArray**()

Returns an array containing all of the elements in this deque in proper sequence (from first to last element).

The returned array will be "safe" in that no references to it are maintained by this deque. (In other words, this method must allocate a new array). The caller is thus free to modify the returned array.

This method acts as bridge between array-based and collection-based APIs.

**Specified by:**[toArray](http://docs.google.com/java/util/Collection.html#toArray()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Overrides:**[toArray](http://docs.google.com/java/util/AbstractCollection.html#toArray()) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Returns:**an array containing all of the elements in this deque

### toArray

public <T> T[] **toArray**(T[] a)

Returns an array containing all of the elements in this deque in proper sequence (from first to last element); the runtime type of the returned array is that of the specified array. If the deque fits in the specified array, it is returned therein. Otherwise, a new array is allocated with the runtime type of the specified array and the size of this deque.

If this deque fits in the specified array with room to spare (i.e., the array has more elements than this deque), the element in the array immediately following the end of the deque is set to null.

Like the [toArray()](http://docs.google.com/java/util/ArrayDeque.html#toArray()) method, this method acts as bridge between array-based and collection-based APIs. Further, this method allows precise control over the runtime type of the output array, and may, under certain circumstances, be used to save allocation costs.

Suppose x is a deque known to contain only strings. The following code can be used to dump the deque into a newly allocated array of String:

String[] y = x.toArray(new String[0]);

Note that toArray(new Object[0]) is identical in function to toArray().

**Specified by:**[toArray](http://docs.google.com/java/util/Collection.html#toArray(T%5B%5D)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)>**Overrides:**[toArray](http://docs.google.com/java/util/AbstractCollection.html#toArray(T%5B%5D)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **Parameters:**a - the array into which the elements of the deque are to be stored, if it is big enough; otherwise, a new array of the same runtime type is allocated for this purpose **Returns:**an array containing all of the elements in this deque **Throws:** [ArrayStoreException](http://docs.google.com/java/lang/ArrayStoreException.html) - if the runtime type of the specified array is not a supertype of the runtime type of every element in this deque [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified array is null

### clone

public [ArrayDeque](http://docs.google.com/java/util/ArrayDeque.html)<[E](http://docs.google.com/java/util/ArrayDeque.html)> **clone**()

Returns a copy of this deque.

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a copy of this deque**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ArrayDeque.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/AbstractSet.html)   [**NEXT CLASS**](http://docs.google.com/java/util/ArrayList.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/ArrayDeque.html)    [**NO FRAMES**](http://docs.google.com/ArrayDeque.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#26in1rg) |

[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).