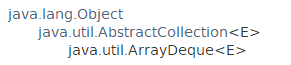
ArrayDeque数组双端队列

# ArrayDeque数组双端队列

## 简单介绍



public class ArrayDeque<E> extends AbstractCollection<E>

implements **Deque**<E>, Cloneable, Serializable

All Implemented Interfaces: Serializable, Cloneable, Iterable<E>, Collection<E>, **Deque**<E>, Queue<E>

This class is a member of the Java Collections Framework. Since:1.6

## 功能介绍

### 数组双端队列：容量无限制，随着使用扩增

**Resizable-array implementation of the Deque interface**. Array deques have no capacity restrictions; they grow as necessary to support usage.

### 线程不安全，不支持并发访问

They are not thread-safe; in the absence of external synchronization, they do not support concurrent access by multiple threads.

### 不支持Null元素

Null elements are prohibited. This class is likely to be faster than Stack when used as a stack, and faster than **LinkedList** when used as a queue.

ArrayDeque用作stack时，比Stack快，用作queue时，比LinkedList快。

### 操作时间固定

Most ArrayDeque operations run in amortized constant time. Exceptions include remove, removeFirstOccurrence, removeLastOccurrence, contains, iterator.remove(), and the bulk operations, all of which run in linear time.

### iterators

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. 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 and Iterator interfaces.

## 构造方法

默认数组大小为16。

ArrayDeque()

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

ArrayDeque(Collection<? extends E> c)

Constructs a deque containing the elements of the specified collection, in the order they are returned by the collection's iterator.

ArrayDeque(int numElements)

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

## 一般方法

### 添加元素

add、addFirst、addLast

offer、offerFirst、offerLast(源码中调用addXxx方法)

ArrayDeque只是数组双端队列，并没有阻塞，因此没有put和take等阻塞式方法。

### 获取元素

element、getFirst、getLast

poll、pollFirst、pollLast

peek、peekFirst、peekLast

### remove

remove、removeFirst、removeLast

### toArray

**Object[] toArray()**

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

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

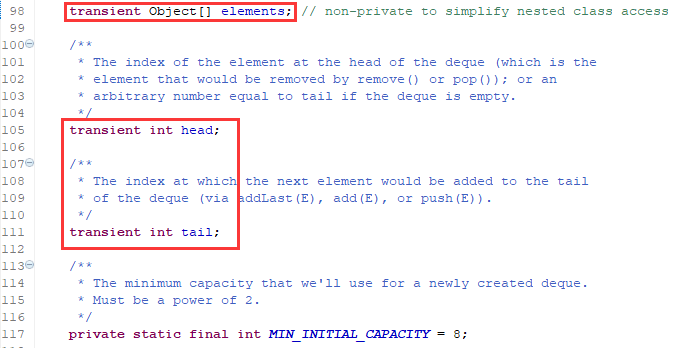
Returns an array containing all of the elements in this deque in proper sequence (from first to last element); the runtime type of the returned array is that of the specified array.

# 源码分析

## ArrayDeque

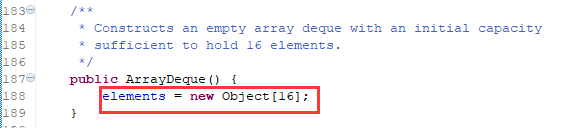


## 数据结构：数组

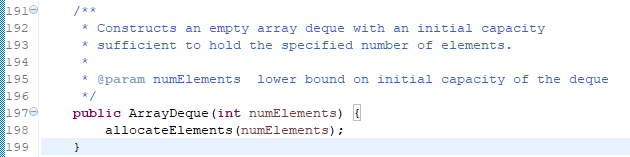


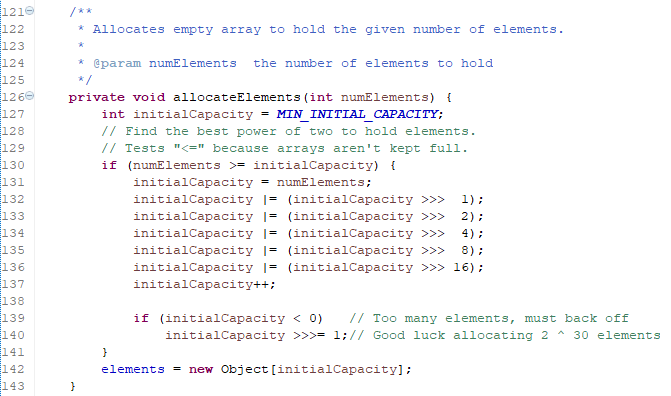
## 构造方法

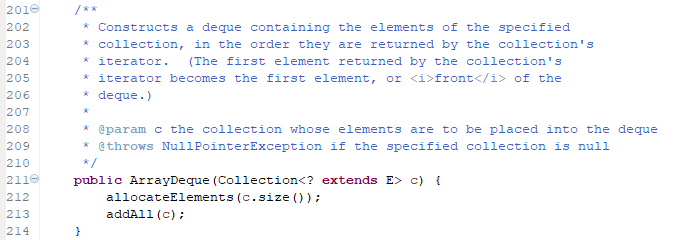
默认为数组大小为16.



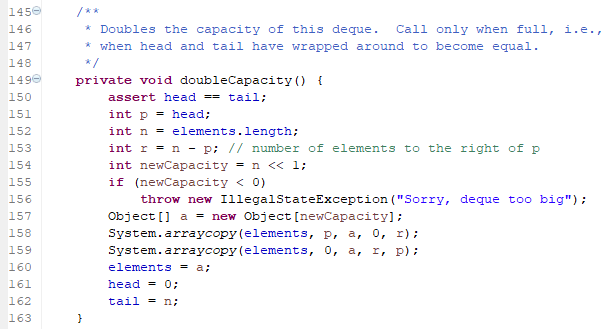
指定大小：







## 容量二倍扩充doubleCapacity

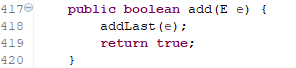


## 核心方法介绍

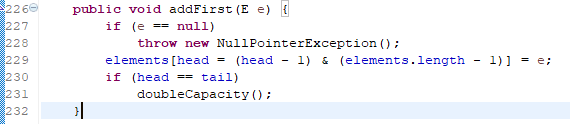
ArrayDeque只是数组双端队列，并没有阻塞，因此没有put和take等阻塞式方法。

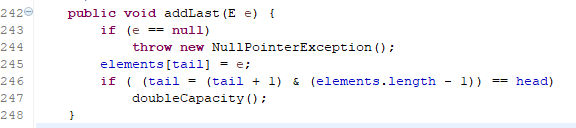
### 添加元素

add方法和addLast一样

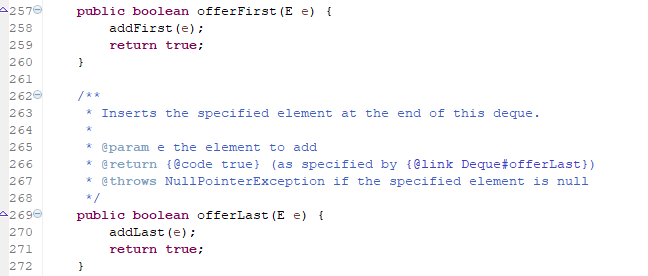


若数组满了，则自动二倍扩容。

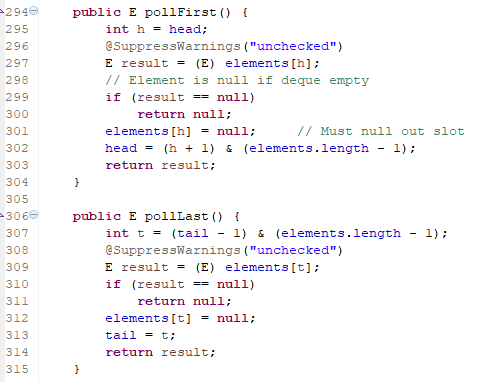




offerFirst、offerLast与addFirst、addLast相同



### 获取元素



### 其他的