**深入理解Java集合之---Deque**

[我是吸血鬼](https://www.jianshu.com/u/06abc6ce0160)关注

0.1152018.05.26 19:51:31字数 566阅读 13,750

**Deque**

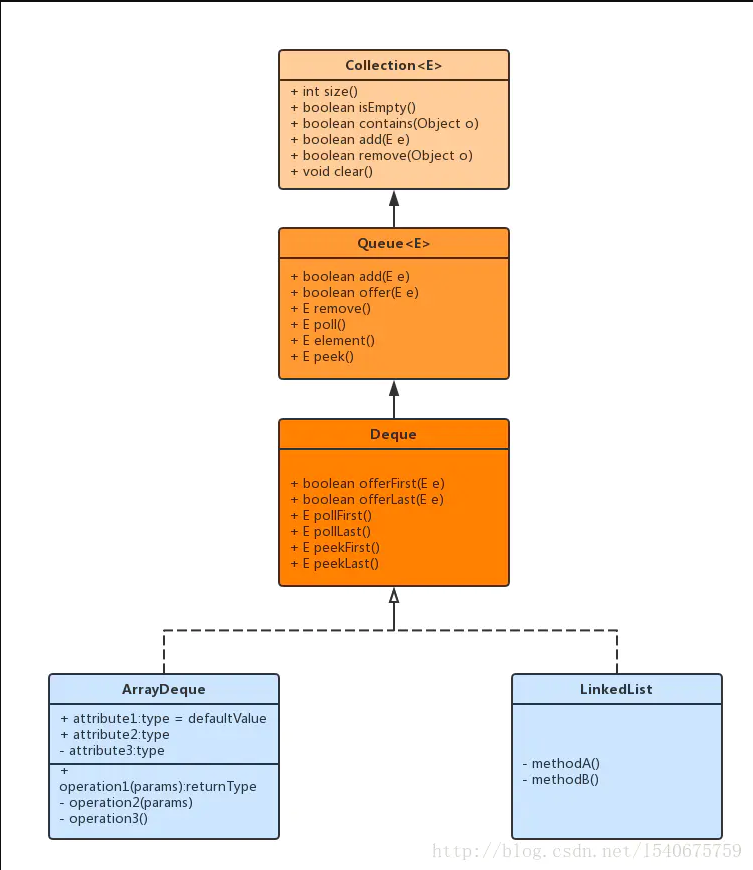
interface Deque <E>

子接口：

BlockingDeque <E>

实现类：

* [ArrayDeque](https://docs.oracle.com/javase/8/docs/api/java/util/ArrayDeque.html)
* [ConcurrentLinkedDeque](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ConcurrentLinkedDeque.html)
* [LinkedBlockingDeque](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/LinkedBlockingDeque.html)
* [LinkedList](https://docs.oracle.com/javase/8/docs/api/java/util/LinkedList.html)



Deque

线性集合，支持两端的元素插入和移除。Deque是double ended queue的简称，习惯上称之为双端队列。大多数Deque 实现对它们可能包含的元素的数量没有固定的限制，但是该接口支持容量限制的deques以及没有固定大小限制的deque。

在大多数实现中，插入操作不会失败。

发音为deck,千万因为发音问题出丑哦。

Deque同时扩展了Queue接口，当Deque作为队列的时候，会产生FIFO(先进先出)行为。元素添加在双端队列的末尾并从头开始删除。

| **Queue 方法** | **等效Deque方法** |
| --- | --- |
| add(e) | addLast(e) |
| offer(e) | offerLast(e) |
| remove() | removeFirst() |
| poll() | pollFirst() |
| element() | getFirst() |
| peek() | peekFirst() |

同时，Deque也可以作为LIFO(后进先出)堆栈,此接口优于传统的Stack类使用。

Stack和Deque方法的比较

| **栈方法** | **等效Deque方法** |
| --- | --- |
| push(e) | addFirst(e) |
| pop() | removeFirst() |
| peek() | peekFirst() |

与List接口不同，此接口不支持索引访问元素。

这个接口提供了两种方法去除内部元素，removeFirstOccurrence和removeLastOccurrence。

| **removeFirstOccurrence(Object o)** | **removeLastOccurrence(Object o)** |
| --- | --- |
| 从此双端队列中移除指定元素的第一个匹配项。 | 从此双端队列中移除指定元素的最后一次出现。 |

虽然 Deque 实现没有严格要求禁止插入 null 元素，但建议最好这样做。建议任何事实上允许 null 元素的 Deque 实现用户最好不 要利用插入 null 的功能。这是因为各种方法会将null 用作特殊的返回值来指示双端队列为空。

Deque 实现通常不定义基于元素的 equals 和 hashCode 方法，而是从 Object 类继承基于身份的equals 和 hashCode 方法。

**Deque的使用场景**  
在一般情况，不涉及到并发的情况下，有两个实现类，可根据其自身的特性进行选择，分别是：

* LinkedList 大小可变的链表双端队列，允许元素为插入null。
* ArrayDeque 大下可变的数组双端队列，不允许插入null。
* ConcurrentLinkedDeque 大小可变且线程安全的链表双端队列，非阻塞，不允许插入null。
* LinkedBlockingDeque 为线程安全的双端队列，在队列为空的情况下，获取操作将会阻塞，直到有元素添加。

注意：LinkedList 和 ArrayDeque 是线程不安全的容器。