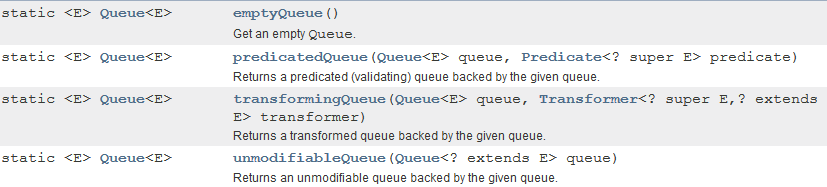
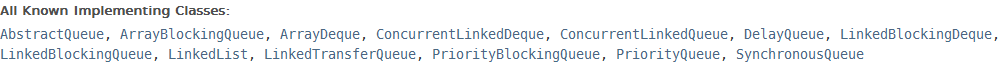
Queue队列及QueueUtils工具类

1. commons-collections中有个单独的子包叫.queue，里面存放的就是关于队列的一些类，如CircularFifoQueue<E>。
2. 在**主包org.apache.commons.collections4**中，有queue队列的工具类QueueUtils，提供了**几个静态方法**获取对象。



1. **接口Queue**存在于java.util包中，直接继承于Collection接口，间接继承于Iterable接口。属于集合中的一种。

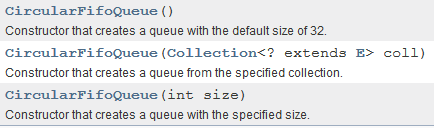
具有很多实现类，如LinkedList类。



1. CircularFifoQueue<E>类：**循环先入先出队列**。

继承框架：

1. **构造方法**：三个。**空参数、指定大小、对集合包装**。

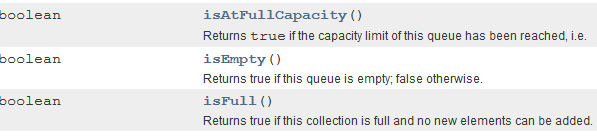


1. 添加方法：add





1. 判断方法：



1. 查询大小：

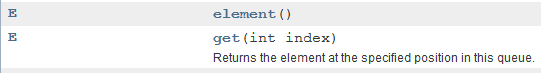




1. 获取迭代器：



1. 获取元素：





1. 其他继承方法略。
2. **只读队列：UnmodifiableQueue<E>类**：不可修改的队列，即**只读队列**。





继承体系：



1. 没有构造方法。通过静态方法对队列进行包装从而创建此类的对象。



也可以通过QueueUtils工具类获取对象。



1. 添加方法：add、addAll、offer方法。

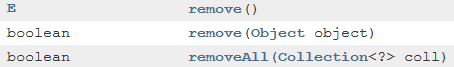




1. 迭代器：



1. 删除方法：





1. retainAll方法：



1. 示例：

Queue<Integer> queue = **new** CircularFifoQueue<Integer>(2);

queue.add(999);

queue.add(444);

queue.add(555);

UnmodifiableQueue<Integer> onlyRead = (UnmodifiableQueue<Integer>) UnmodifiableQueue.*unmodifiableQueue*(queue);

// onlyRead .add(666);抛出了java.lang.UnsupportedOperationException

Iterator<Integer> iterator = onlyRead.iterator();

**while**(iterator .hasNext()) {

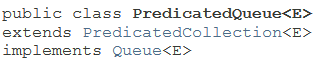
System.***out***.println(iterator.next());

}

1. **断言队列：PredicatedQueue<E>**







1. 构造方法是protected修饰的，其静态方法：



1. 其他大部分方法都是继承Queue规定的方法。
2. 示例：

Queue<String> queue = **new** CircularFifoQueue<String>(2);

Predicate<String> predicate = **new** Predicate<String>() {

@Override

**public** **boolean** evaluate(String object) {

**return** object.length()<10;

}

};

Queue<String> preQueue = PredicatedQueue.*predicatedQueue*(queue, predicate);

preQueue.add("zhaohong");

preQueue.add("zhaohong2");

// preQueue.add("zhaohongzxy");抛出异常java.lang.IllegalArgumentException