

## 分清楚BlockingQueue的几个容易混淆的接口

2015-10-29 16:33 feiying  0  0 阅读 54

Queue是Collection接口的一个分支，它的作用就是模拟先入先出的数据结构，和Stack这种先入后出的数据结构相反，

而对于Queue在Collection中的默认实现，PriorityQueue是优先队列，其并不是线程安全的，举个例子来说，两个线程同时对这个队列进行操作，线程1正在进行出队操作，获得了队头元素Node1，并进行基于场景设置好值了，

正在执行的过程中，时间片轮转，线程2也进行出队的操作，

它也看到队列头是Node1，因此它先出队了，而这个时候线程1再拿队头元素发现已经不是我原来设置好值的Node1节点，

====》因此，这就是线程安全问题，在多线程的环境下，不但你的业务代码需要保证线程无关，而且你使用的数据结构也得线程无关化，毕竟线程中的代码也是你业务逻辑代码中的一部分；

对于多线程的环境下，一种解决方法就是采用阻塞队列来做，也就是BlockingQueue，

阻塞队列能解决两种场景：

a.线程安全性，在操作的过程中，保证所有的内容都是在锁内进行

b.阻塞性，正因为为阻塞队列，所谓的阻塞是，当队列满了，你调用阻塞的添加方法，你的应用程序就因此而阻塞住了，

而当队列没了，你调用阻塞的获取方法的时候，你的应用程序就因此也阻塞住，

所以可以看出，阻塞队列中，当你调用阻塞的操作，会导致你的应用程序也跟着阻塞；

这里列一下这几个方法，

以插入为例：

当调用add方法的时候，成功返回true，失败直接爆出异常；

当调用offer方法的时候，成功返回true，失败返回false；

当调用put方法的时候，成功返回true，失败会一直阻塞，其原因为空间不够，一直阻塞到有可用空间为止；

下载《开发者大全》

下载 (/download/dev.apk)



当调用offer的超时方法的时候，在超时时间内，如果有空间可以增添元素的话，并且会直接返回true，

在超时时间内，如果没有空间的话，会和put一样进行阻塞并且一直等待可用空间，如果空间有了，直接添加返回true，

如果在超时时间之内还没有将元素加入到队列中，那么这里就不等了，直接返回false；

对于删除来将：

remove，poll，take和上述相对；

除此之外，还有peek，element极容易和移除相混淆，这两个方法实际的意思是对于队列头的内容，我就看一眼而已，但是并不对队列进行操作，

这两个方法其实保证的就是并发当中的可见性而已，peek保证看到的都是最新的，如果队列为空，那么peek就返回false；

而element方法当队列为空，直接抛出异常，这就是二者之间的区别；

**总结：**

**阻塞队列中很多方法特别容易混淆，**

**增添元素中add,offer,put,offer超时重载都各有不同的意思，**

**队列头删除是remove,poll,take,poll超时重载，**

**它和peek，element这种队列头查看方法不是一个意思！**

分享：

阅读 54  0

应用服务器技术讨论圈 更多文章

东方通加码大数据业务 拟募资8亿收购微智信业 (/html/308/201504/206211355/1.html)

玩转Netty – 从Netty3升级到Netty4 (/html/308/201504/206233287/1.html)

下载《开发者大全》

下载 (/download/dev.apk)



金蝶中间件2015招聘来吧！Come on！（/html/308/201505/206307460/1.html）

GlassFish 4.1 发布，J2EE 应用服务器（/html/308/201505/206323120/1.html）

Tomcat对keep-alive的实现逻辑（/html/308/201505/206357679/1.html）

猜您喜欢

未来已来，看看腾讯用机器人写的新闻稿（/html/374/201509/209314957/1.html）

2小时搞定移动直播App开发 开始学习（/html/370/201608/2651474783/1.html）

【必看】用数据讲故事的五个步骤。（/html/301/201407/200476942/1.html）

不同优秀程度JavaScript开发者的差别（/html/471/201607/2651552606/1.html）

就是这小子，把学校数据库给黑了（/html/283/201608/2650861987/1.html）