



链滴

DelayQueue

作者: [614756773](#)

原文链接: <https://ld246.com/article/1595746780421>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



功能

- 延迟队列中的所有元素都有一个时间 **倒计时的作用**,在调用take时会阻塞去获取数据,直到队列头数据倒计时结束,才返回。

源码分析

- 父级
 - BlockingQueue **所以延迟队列也是阻塞的**
- 成员
 - PriorityQueue q **该优先级队列用于保存元素,并且能够将元素按照倒计时排序,倒计时小的在列前面**
 - Thread leader **Leader/Follower模式,减少线程的开销**
- Leader/Follower模式
 - 先从优先级队列中取出时间最小的元素
 - 如果时间已经为0了,那么可以直接返回给外部
 - 否则
 - 如果有leader,自己又不是leader那就休眠等待, **leader的时间结束后返回给外部,然后会唤其他follower线程,这就明显的能够减少一些轮询开销了**
 - 如果没有leader,那么将当前线程设置成leader,然后计算出元素剩下的倒计时,并且按照计算

的时间休眠

- 关键代码如下

```
public E take() throws InterruptedException {
    final ReentrantLock lock = this.lock;
    lock.lockInterruptibly();
    try {
        for (;;) {
            E first = q.peek();
            if (first == null)
                available.await();
            else {
                long delay = first.getDelay(NANOSECONDS);
                if (delay <= 0)
                    return q.poll();
                first = null;
                if (leader != null)
                    available.await();
                else {
                    Thread thisThread = Thread.currentThread();
                    leader = thisThread;
                    try {
                        available.awaitNanos(delay);
                    } finally {
                        if (leader == thisThread)
                            leader = null;
                    }
                }
            }
        }
    } finally {
        if (leader == null && q.peek() != null)
            available.signal();
        lock.unlock();
    }
}
```