Semaphore信号量

资源有限共享

停车场为例

并发集合

List set map

并发环境下

遍历的过程中不容许更新操作（增删改）

ConcurrentCollection

**非阻塞式集合（Non-Blocking Collection） ConcurrentLinkedDeque**

这类集合也包括添加和移除数据的方法。如果方法不能立即被执行，则返回null或抛出异常，但是调用这个方法的线程不会被阻塞。

实例

添加大量的数据到一个列表中；

从同一个列表中移除大量的数据。

**阻塞式集合（Blocking Collection） LinkedBlockingDeque**

阻塞式集合（Blocking Collection）：这类集合包括添加和移除数据的方法。当集合已满或为空时，被调用的添加或者移除方法就不能立即被执行，那么调用这个方法的线程将被阻塞，一直到该方法可以被成功执行。

分析ConcurrentHashMap

HsahMap

DataStructure

**ConcurrentLinkedDeque**

**[add](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/ConcurrentLinkedQueue.html" \l "add(E))**([E](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/ConcurrentLinkedQueue.html" \o "ConcurrentLinkedQueue 中的类型参数) e)           将指定元素插入此队列的尾部。

**[offer](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/ConcurrentLinkedQueue.html" \l "offer(E))**([E](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/ConcurrentLinkedQueue.html" \o "ConcurrentLinkedQueue 中的类型参数) e)           将指定元素插入此队列的尾部。

**[poll](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/ConcurrentLinkedQueue.html" \l "poll())**()        获取并移除此队列的头，如果此队列为空，则返回 null。

Node

**LinkedBlockingDeque**

|  |
| --- |
| **[LinkedBlockingDeque](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/LinkedBlockingDeque.html" \l "LinkedBlockingDeque())**()            创建一个容量为 [Integer.MAX\_VALUE](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Integer.html" \l "MAX_VALUE) 的 LinkedBlockingDeque。 |
| **[LinkedBlockingDeque](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/LinkedBlockingDeque.html" \l "LinkedBlockingDeque(java.util.Collection))**([Collection](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/Collection.html" \o "java.util 中的接口)<? extends [E](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/LinkedBlockingDeque.html" \o "LinkedBlockingDeque 中的类型参数)> c)            创建一个容量为 [Integer.MAX\_VALUE](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Integer.html" \l "MAX_VALUE) 的 LinkedBlockingDeque，最初包含给定 collection 的元素，以该 collection 迭代器的遍历顺序添加。 |
| **[LinkedBlockingDeque](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/LinkedBlockingDeque.html" \l "LinkedBlockingDeque(int))**(int capacity)            创建一个具有给定（固定）容量的 LinkedBlockingDeque。 |

**[put](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/LinkedBlockingDeque.html" \l "put(E))**([E](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/LinkedBlockingDeque.html" \o "LinkedBlockingDeque 中的类型参数) e)   
          将指定的元素插入此双端队列表示的队列中（即此双端队列的尾部），必要时将一直等待可用空间。

**[take](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/LinkedBlockingDeque.html" \l "take())**()   
          获取并移除此双端队列表示的队列的头部（即此双端队列的第一个元素），必要时将一直等待可用元素。

**HashMap entry<k,v>**

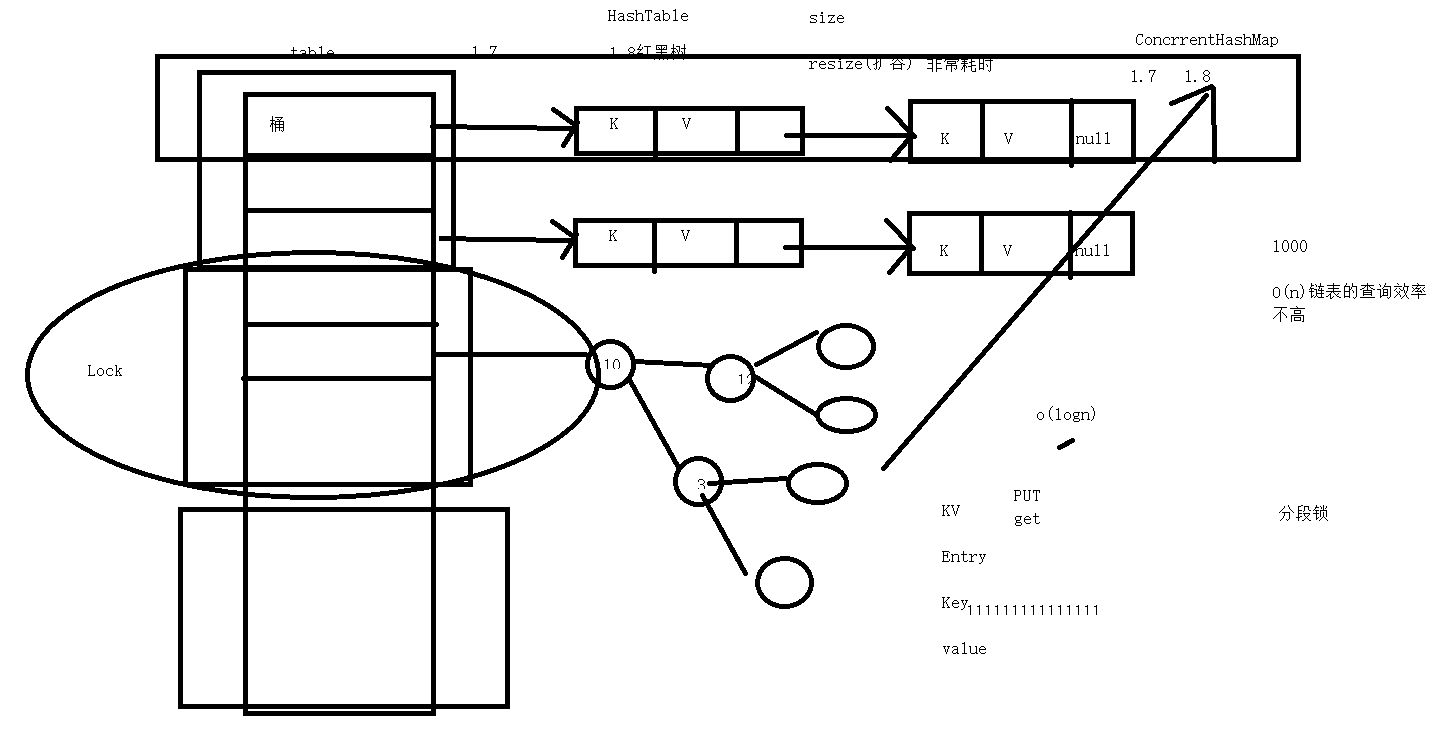
**LinkedTransferQueue 生产-消费**

**PriorityBlockingQueue 优先级**

[ArrayBlockingQueue](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/ArrayBlockingQueue.html" \o "java.util.concurrent 中的类" \t "http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/classFrame)   
[ConcurrentHashMap](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/ConcurrentHashMap.html" \o "java.util.concurrent 中的类" \t "http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/classFrame)   
[ConcurrentLinkedQueue](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/ConcurrentLinkedQueue.html" \o "java.util.concurrent 中的类" \t "http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/classFrame)   
[ConcurrentSkipListMap](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/ConcurrentSkipListMap.html" \o "java.util.concurrent 中的类" \t "http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/classFrame)   
[ConcurrentSkipListSet](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/ConcurrentSkipListSet.html" \o "java.util.concurrent 中的类" \t "http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/classFrame)   
[CopyOnWriteArrayList](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/CopyOnWriteArrayList.html" \o "java.util.concurrent 中的类" \t "http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/classFrame)   
[CopyOnWriteArraySet](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/CopyOnWriteArraySet.html" \o "java.util.concurrent 中的类" \t "http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/classFrame)

**HashMap**

**数组+链表**



原子操作

CAS

AtomicInteger

[tomicBoolean](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/atomic/AtomicBoolean.html" \o "java.util.concurrent.atomic 中的类" \t "http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/atomic/classFrame)   
[AtomicInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/atomic/AtomicInteger.html" \o "java.util.concurrent.atomic 中的类" \t "http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/atomic/classFrame)

[AtomicLong](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/atomic/AtomicLong.html" \o "java.util.concurrent.atomic 中的类" \t "http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/atomic/classFrame)

Long

64

32 32

CAS

CompareAndSet

 此引用所引用的对象类型。

ABA问题

1. -》120--》100
2. 》120

[AtomicStampedReference](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/atomic/AtomicStampedReference.html" \o "java.util.concurrent.atomic 中的类" \t "http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/concurrent/atomic/classFrame)

100 1---->120 2----->100----3

100 1------120