

简介

星期四,2019年9月19日

14:34

一、概述

1. Concurrent包是JDK1.5提供的一套并发包
2. Concurrent包中主要是包含了5块内容：BlockingQueue、ConcurrentMap、ExecutorService、Lock、原子性操作

BlockingQueue - 阻塞式队列

星期四, 2019年9月19日 14:37

一、概述

1. 本身是一个队列，意味着满足队列FIFO的特点
2. 阻塞式队列都是**有界**的，意味着所有的阻塞式队列的大小是固定的
3. 如果队列已满，则添加元素的线程会被阻塞，直到队列中有元素被取出；如果队列为空，则获取元素的线程也会被阻塞，直到队列中被添加元素
4. 阻塞式队列适合于生产消费模型
5. 阻塞式队列中要求元素非空
6. 重要方法：

	抛出异常	返回值	长期阻塞	定时阻塞
放	add	offer - false	put	offer
拿	remove	poll - null	take	poll

二、实现类

1. ArrayBlockingQueue - 阻塞式顺序队列
 - a. 底层是基于数组来进行存储
 - b. 在使用的时候需要指定容量，并且容量指定之后是不可变的
2. LinkedBlockingQueue - 阻塞式链式队列
 - a. 底层是基于链表实现
 - b. 在使用的时候可以指定容量也可以不指定。如果指定了容量，则容量固定；如果不指定容量，则容量默认为Integer.MAX_VALUE，即 $2^{31}-1$ 。到了实际开发中，一般不会向一个队列中来添加21亿个值，所以一般人为的认为此时这个队列是无界的
3. PriorityBlockingQueue - 具有优先级的阻塞式队列
 - a. 在使用的时候如果不指定容量，则默认容量为11
 - b. 在遍历元素的时候是有序的，这也就要求添加的元素对应的类必须实现Comparable - 自然排序 - 如果是通过实现Comparable接口来实现的排序称之为是自然排序
 - c. 迭代遍历不保证排序

4. SynchronousQueue - 同步队列

- a. 在使用的时候不需要指定容量。默认容量是1并且只能为1

扩展：BlockingDeque - 阻塞式双向队列 - 允许从两端放或者拿元素

ConcurrentMap - 并发映射

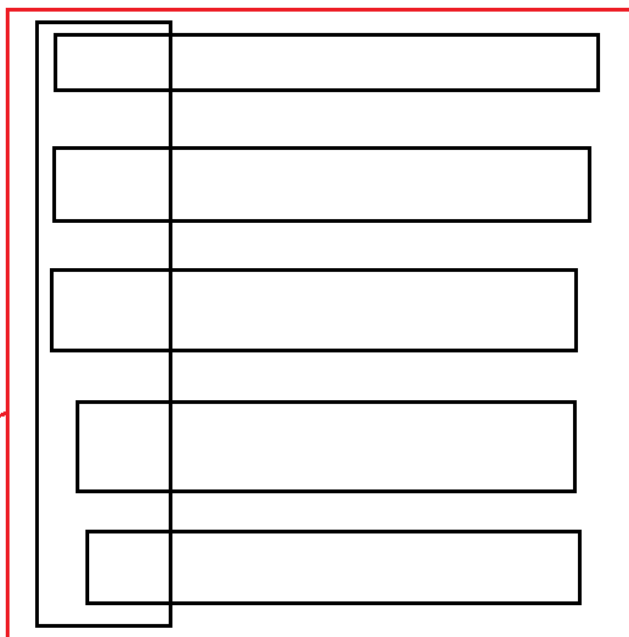
星期四, 2019年9月19日 15:52

一、概述

1. 本质上是一个Map，依然以键值对结构来存储数据
2. ConcurrentMap提供了并发并且安全的方式来读写数据

二、ConcurrentHashMap - 并发哈希映射

Hashtable对外提供的方法都是同步非静态方法 --- 同步非静态方法的锁对象是this - this表示当前对象 - Hashtable是以当前对象对象作为锁对象



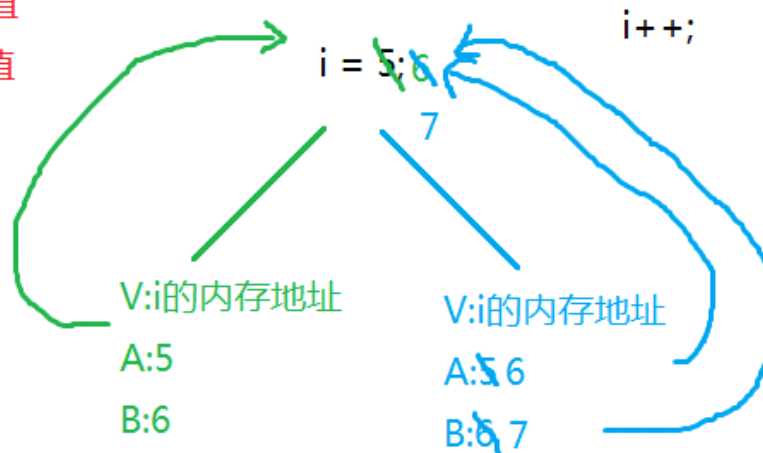
1. 底层是基于数组+链表存储，默认初始容量为16，默认加载因子是0.75，每次扩容默认增加一倍
2. ConcurrentHashMap是一个异步式线程安全的映射：在ConcurrentHashMap引入了**分段(桶)锁**机制来解决Hashtable所带来的效率降低的问题。在后续JDK版本中，ConcurrentHashMap在分段锁的基础上引入了**读写锁**来提高效率：
 - a. 读锁：允许多个线程读，不允许线程写
 - b. 写锁：只能允许一个线程写，不允许线程读
3. 在JDK1.8中，ConcurrentHashMap引入了CAS (Compare And Swap，比较和交换) 无锁算法保证异步线程安全。CAS算法需要与具体的内核结合，目前几乎所有的内核架构都是支持CAS

CAS的语义：我认为V的值应该是A，如果是，那么将V的值更新为B，否则不修改并告诉V的值实际为多少

V：内存值

A：旧的预期值

B：新的预期值



CAS过程中只要被打断，那么所有过程从头另来

4. 在JDK1.8中，为了提高增删效率，ConcurrentHashMap引入了红黑树机制。当桶中的元素个数超过8个的时候，桶中的链表会扭转成一棵红黑树；如果红黑树的节点个数不足7个，那么红黑树会扭转回链表

三、ConcurrentNavigableMap