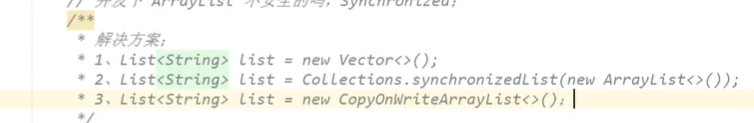


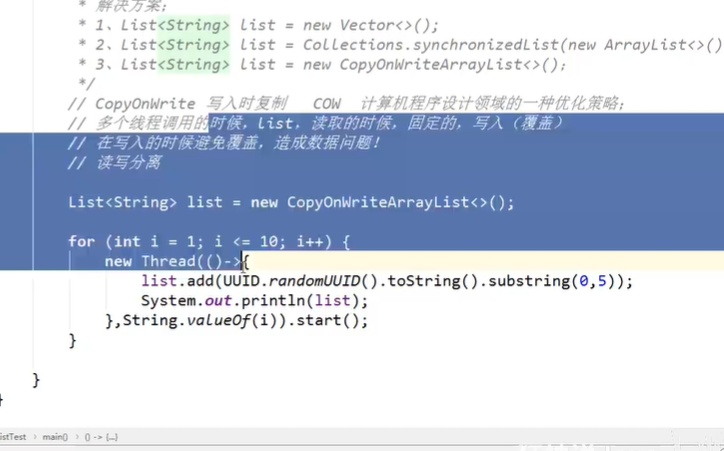




并发修改异常



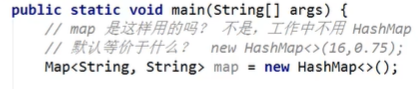




Synchronic会降低运行效率

set同理

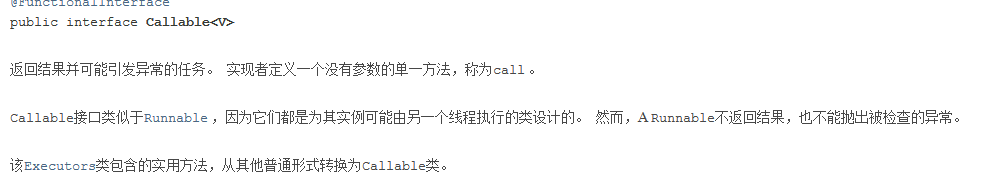




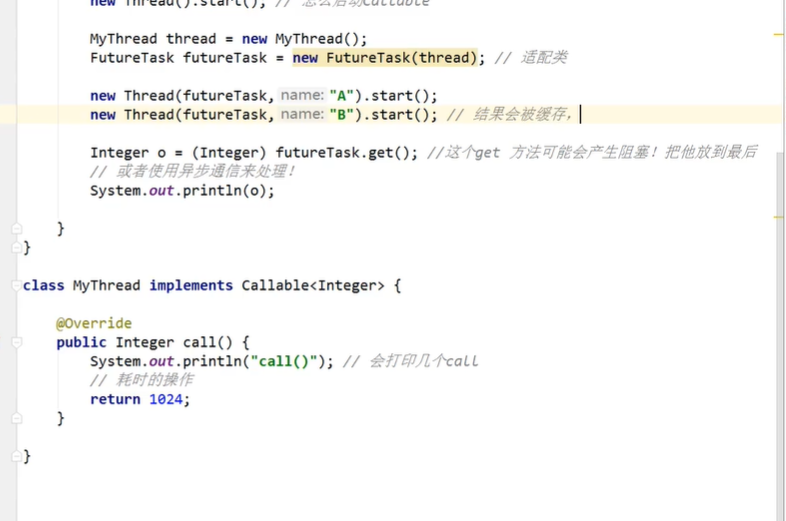
并发map



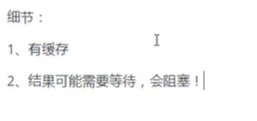
Callable创建线程



有返回值，可以抛出异常，方法call，类似run



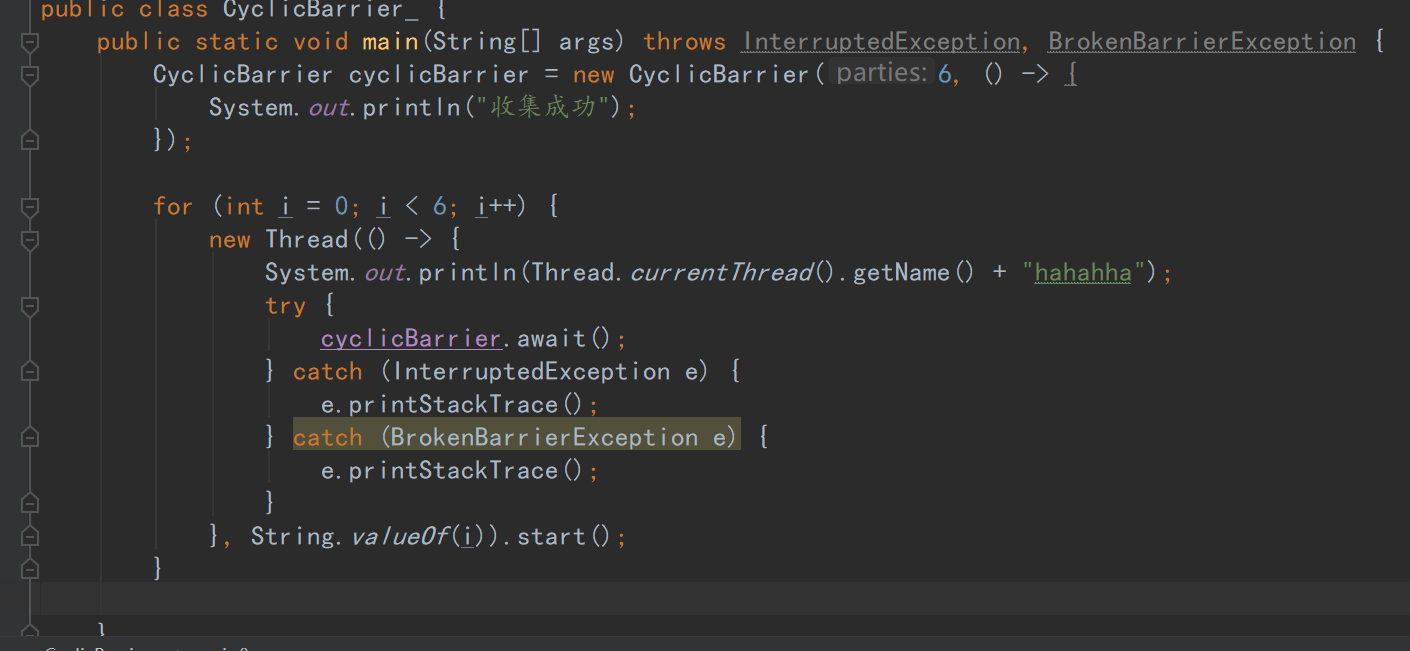
两个结果，但只会显示一个，被缓存了，提高效率



CountDownLatch

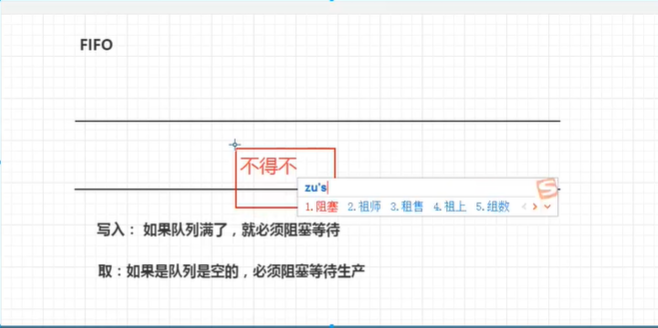


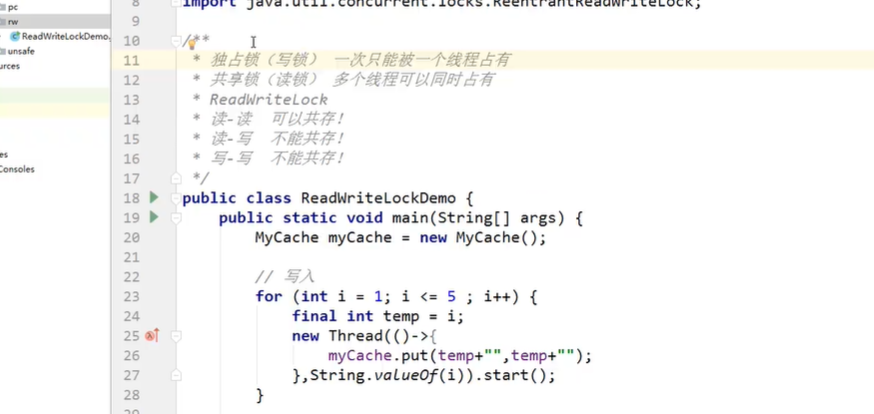
Cyclicbarrier



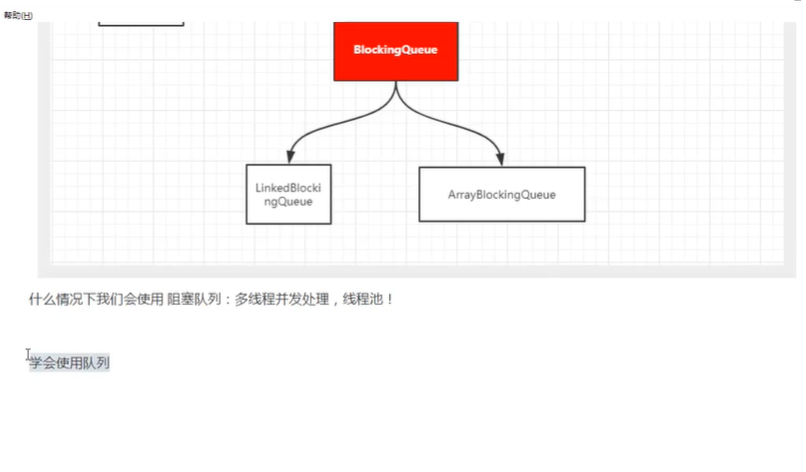
读写锁

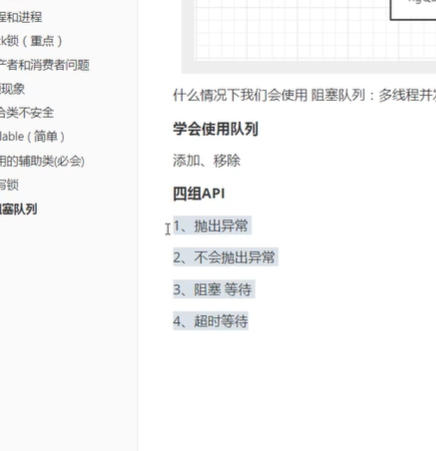
阻塞队列





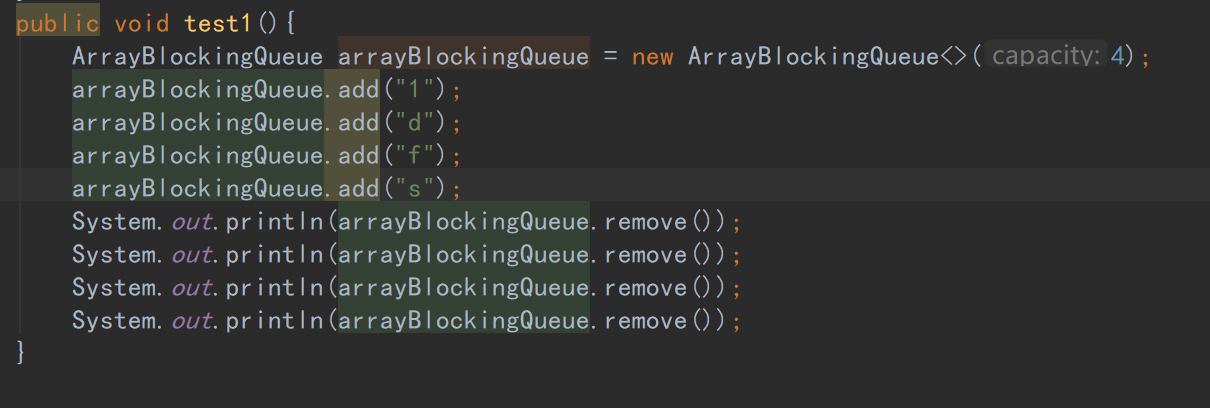






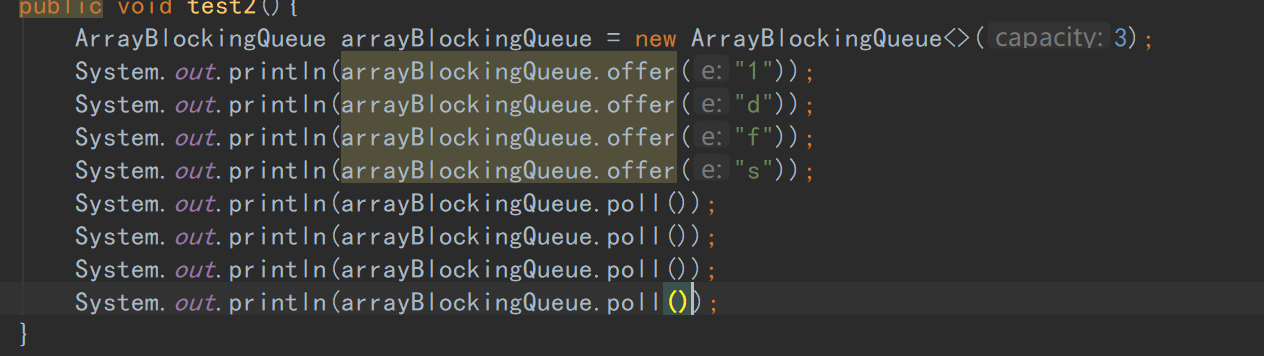
ArrayBlockingQueue

先进先出，会抛出异常，队列满了和空会抛出异常



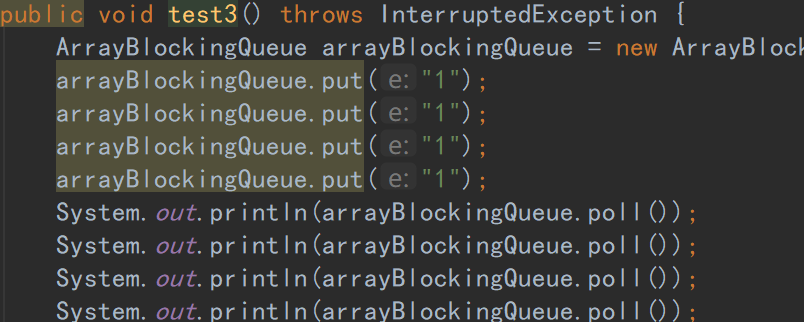
Offer，poll不会抛出异常,会返回true和false

element返回对手元素

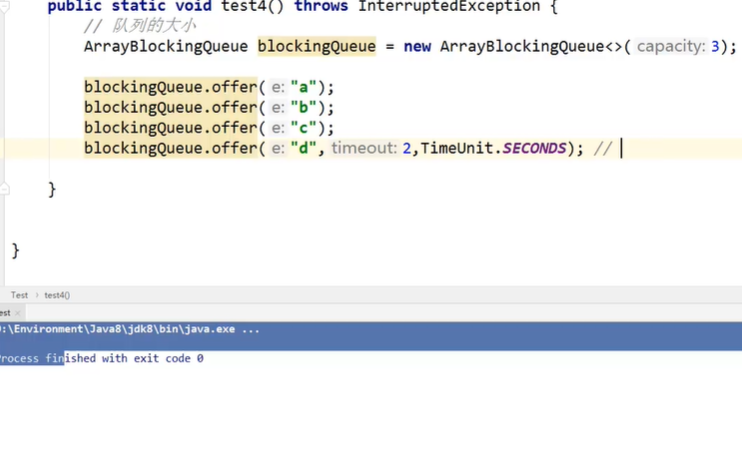


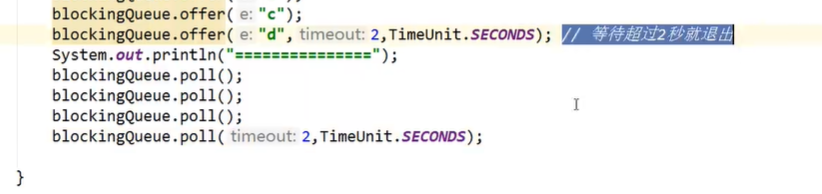
Peek检测队首元素

Put一直阻塞直到添加成功，take同理



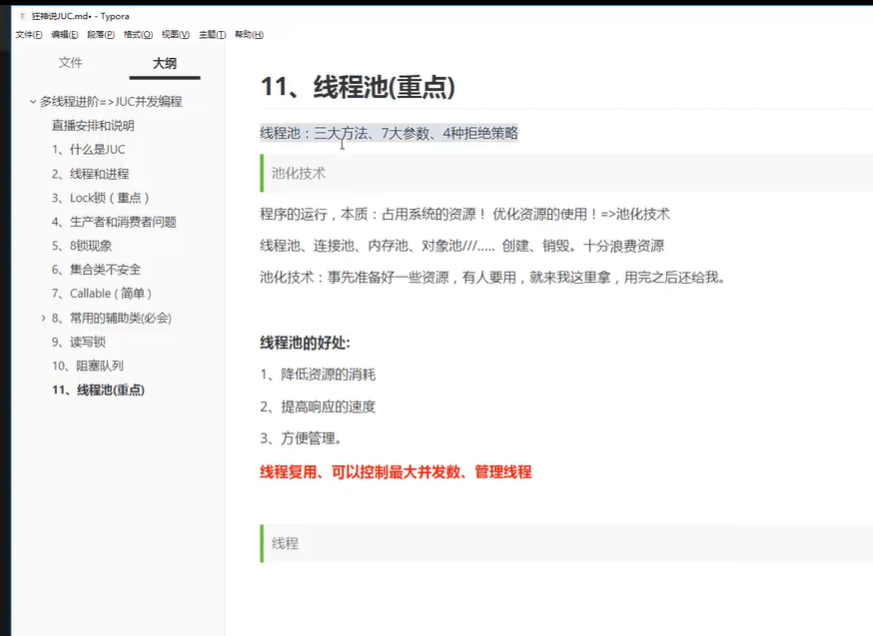
超市阻塞，等待超时就退出。









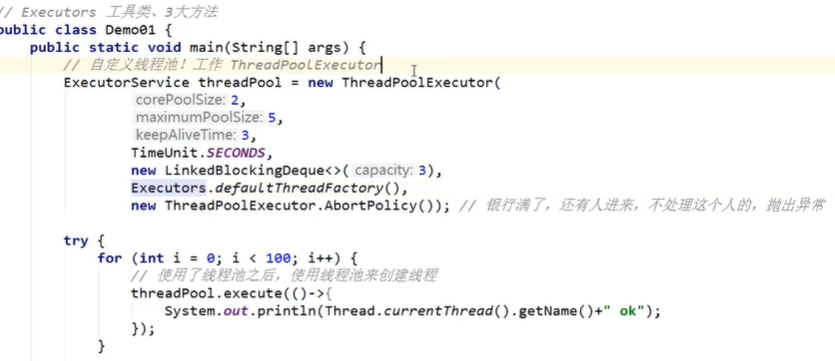




Oom是溢出异常



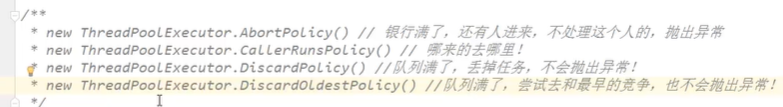
实际开发使用的线程池：



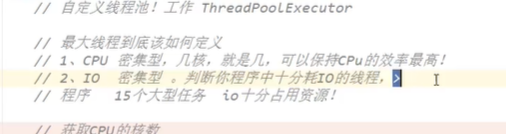
最大线程为最大线程加队列数，超出将会抛出异常。



四种拒绝策略：



获取cpu的核数：



上面是池的最大大小设置

四大函数式接口：

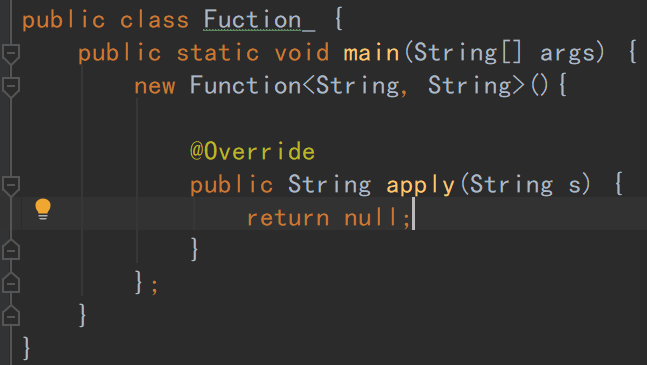


作用是简化编程模型



第一种接口，函数式接口





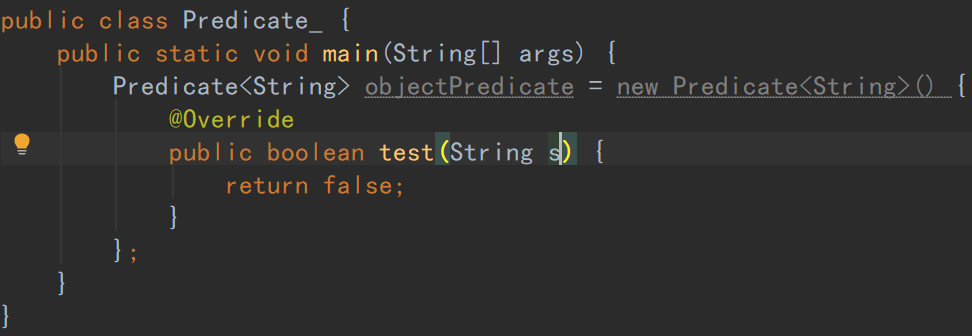
Lamda表达式可以省掉括号：



第二种接口，断定式接口，predicate



演示代码：

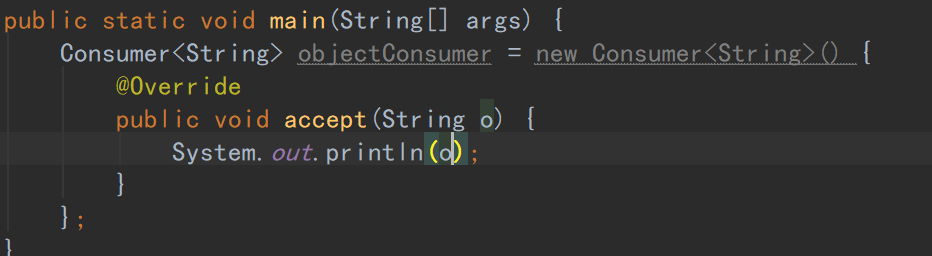


使用lamda表达式简化：



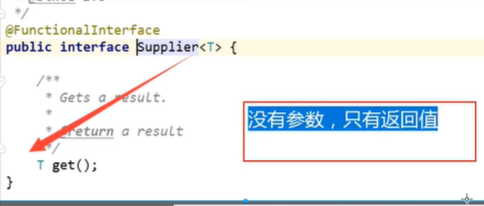
第三种接口：

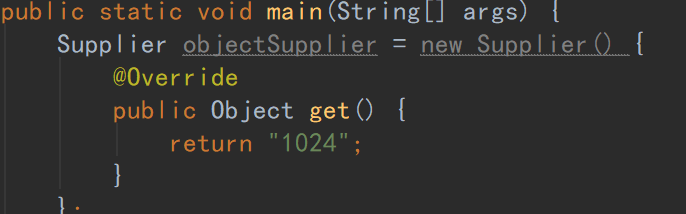




使用lamda表达式：

第四种接口Supplier：







链式编程

