计算机二级辅导:Java线程新特征(原子量)计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/645/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E4_c97_645363.htm 所谓的原子量即操作变量的操作是“ 原子的 ”，该操作不可再分，因此是线程安全的。 为何要使用原子变量呢，原因是多个线程对单个变量操作也会引起一些问题。在Java5之前，可以通过volatile、synchronized关键字来解决并发访问的安全问题，但这样太麻烦。 Java5之后，专门提供了用来进行单变量多线程并发安全访问的工具包java.util.concurrent.atomic，其中的类也很简单。 下面给出一个反面例子（切勿模仿）： import java.util.concurrent.ExecutorService. import java.util.concurrent.Executors. import java.util.concurrent.atomic.AtomicLong. /** * Java线程：新特征-原子量 **/ public class Test { public static void main(String[] args) { ExecutorService pool = Executors.newFixedThreadPool(2). Runnable t1 = new MyRunnable("张三", 2000). Runnable t2 = new MyRunnable("李四", 3600). Runnable t3 = new MyRunnable("王五", 2700). Runnable t4 = new MyRunnable("老张", 600). Runnable t5 = new MyRunnable("老牛", 1300). Runnable t6 = new MyRunnable("胖子", 800). //执行各个线程 pool.execute(t1). pool.execute(t2). pool.execute(t3). pool.execute(t4). pool.execute(t5). pool.execute(t6). //关闭线程池 pool.shutdown(). } } class MyRunnable implements Runnable { private static AtomicLong aLong = new AtomicLong(10000). //原子量，每个线程都可以自由操作 private String name. //操作人 private int x. //

操作数额 MyRunnable(String name, int x) { this.name = name. this.x = x. } public void run() { System.out.println(name "执行了" x "，当前余额：" aLong.addAndGet(x)). } } 运行结果： 李四执行了3600，当前余额：13600 王五执行了2700，当前余额：16300 老张执行了600，当前余额：16900 老牛执行了1300，当前余额：18200 胖子执行了800，当前余额：19000 张三执行了2000，当前余额：21000 Process finished with exit code 0 张三执行了2000，当前余额：12000 王五执行了2700，当前余额：18300 老张执行了600，当前余额：18900 老牛执行了1300，当前余额：20200 胖子执行了800，当前余额：21000 李四执行了3600，当前余额：15600 Process finished with exit code 0 张三执行了2000，当前余额：12000 李四执行了3600，当前余额：15600 老张执行了600，当前余额：18900 老牛执行了1300，当前余额：20200 胖子执行了800，当前余额：21000 王五执行了2700，当前余额：18300 Process finished with exit code 0 从运行结果可以看出，虽然使用了原子量，但是程序并发访问还是有问题，那究竟问题出在哪里了？ 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com