### ArrayList:

### 程序1、

线程138357780

线程11864934404

Exception in thread "Thread-0" java.util.ConcurrentModificationException

at java.util.ArrayList$Itr.checkForComodification(ArrayList.java:909)

at java.util.ArrayList$Itr.remove(ArrayList.java:873)

at com.captain.demo.threadNoSafe.ThreadSafeDemo1.run(ThreadSafeDemo1.java:29)

at java.lang.Thread.run(Thread.java:748)

线程1 999

Arraylist最终提示最终由999个size，说明在另一个线程中也对ArrayList中进行了next（）或者remove（）操作。而Vector不让其进入

**在程序1中，因为用的iterator，而且还是多个线程操作一个list。**

**而ArrayList中没有线程安全保护，即一个个进的保护，导致Arraylist被混乱操作，不知道怎么删除**

### 程序2、

线程2 199200

线程2 199201

线程2 199202

线程2 199203

线程2 199204

线程2 199205

线程2 199206

线程2 199207

线程2 199208

线程2 199209

线程2 199210

线程2 199211

线程3-----------------------------------197915

123

线程2 199212

线程2 199213

线程2 199214

如果多线程add非常快的话，会弹出删除的代码。但是如果多线程add添加了睡眠的话，就会没运行删除的代码，而且会有下面这个运行错误的提示，数组越

Arraylist的ADD底层代码如下：

**public** **boolean** add(E e) {

ensureCapacityInternal(size + 1); // Increments modCount!!

elementData[size++] = e;

**return** **true**;

}

**所以推测是线程加了睡眠，然后使各个线程的间隔时间逐渐拉短，然后出现了线程同时运行add这个函数导致出现数组越界错误！**

Exception in thread "Thread-2" java.lang.ArrayIndexOutOfBoundsException: 22

at java.util.ArrayList.add(ArrayList.java:463)

at com.captain.demo.threadNoSafe.ThreadSafeDemo2.run(ThreadSafeDemo2.java:25)

at java.lang.Thread.run(Thread.java:748)

### Vector:

### 程序1、

线程11475046501

线程1-610945701

Exception in thread "Thread-0" java.util.ConcurrentModificationException

at java.util.Vector$Itr.checkForComodification(Vector.java:1210)

at java.util.Vector$Itr.remove(Vector.java:1176)

at com.captain.demo.threadNoSafe.ThreadSafeDemo1.run(ThreadSafeDemo1.java:29)

at java.lang.Thread.run(Thread.java:748)

线程1 1000

Vector是安全性数组，不允许多线程同时操作，最后输出1000也刚好。

### 程序2、

线程2 2

线程2 2

线程2 4

线程2 4

线程3--------------------------------------3

线程2 4

线程2 4

线程3--------------------------------------4

线程3--------------------------------------4

线程2 5

线程2 4

线程3--------------------------------------6

线程2 6

线程2 6

线程3--------------------------------------7

线程2 7

线程2 7

线程2 7

线程3--------------------------------------8

线程2 8

线程3--------------------------------------9

线程2 9

线程2 9

线程2 9

线程2 10

线程3--------------------------------------10

线程3--------------------------------------10

线程2 11

线程2 11

线程2 12

线程3--------------------------------------12

没问题！