

直页 新随笔 联系 聚合XML 管理 长袖连衣裙 包邮 常用链接 我的随笔 我的评论 我的参与 最新评论 留言簿(3) <u>给我留言</u> 查看公开留言 查看私人留言 随笔分类 CSS(3) XML DB(3) XML EJB XML Hibernate XML HTML(3) XML ibatis XML Java基础(11) XML JS(3) XML JSP(1) XML Spring(1) XML Struts(2) XML Struts2(1) XML Swing XML 设计模式(7) XML

```
java.util.ConcurrentModificationException
工作中碰到个ConcurrentModificationException。代码如下:/
List list = ...:
for(Iterator iter = list.iterator(); iter.hasNext();) {
  Object obj =/iter.next();
  if(***) {
     list.remove(obj);
在执行了remove方法之后,再去执行循环,iter.next()的时候,报java.util.ConcurrentModificationException(当然,如果remove的是最后一条,就不会
再去执行next()操作了)
下面来看一下源码
public interface Iterator<E> {
  boolean hasNext();
  E next();
  void remove();
public interface Collection<E> extends Iterable<E> {
  Iterator<E> iterator();
  boolean add(E o);
  boolean remove(Object o);
这里有两个remove方法
接下来来看看AbstractList
public abstract class AbstractList<E> extends AbstractCollection<E> implements List<E> {
//AbstractCollection和List都继承了Collection
  protected transient int modCount = 0;
  private class Itr implements Iterator<E> { //内部类Itr
     int cursor = 0;
     int lastRet = -1;
     int expectedModCount = modCount;
     public boolean hasNext() {
       return cursor != size();
     public E next() {
       checkForComodification(); //特别注意这个方法
       try {
```

随笔 - 37 文章 - 16 trackbacks - 0

```
E next = get(cursor)
   随笔档案
                                           lastRet = cursor++;
                                           return next;
                                        } catch(IndexOutOfBoundsException e) {
2009年3月 (1)
                                           checkForComodification();
2009年2月 (1)
                                           throw new NoSuchElementException();
2008年8月 (5)
2008年6月 (2)
2008年5月 (1)
2008年4月 (1)
                                     public void remove() {
2008年3月 (1)
                                        if (lastRet == -1)
2008年2月 (2)
                                           throw new IllegalStateException();
2008年1月 (7)
                                        checkForComodification();
2007年12月 (1)
2007年11月 (4)
                                        try {
                                           AbstractList.this.remove(lastRet); //执行remove对象的操作
                                           if (lastRet < cursor)
   最新评论 XML
                                             cursor--;
                                           lastRet = -1;
                                           expectedModCount = modCount; //重新设置了expectedModCount的值, 避免了ConcurrentModificationException的产生
1. re: JAVA集合小结
                                        } catch(IndexOutOfBoundsException e) {
很好,谢谢LZ
                                           throw new ConcurrentModificationException()
                    --解
2. re: java.util.ConcurrentModifi
cationException[未登录]
@子墨+1
                                     final void checkForComodification() {
               --飞天奔月
                                        if (modCount!= expectedModCount) //当expectedModCount和modCount不相等时,就抛出ConcurrentModificationException
3. re: JAVA集合小结
                                           throw new ConcurrentModificationException();
AbstractrMap 怎么在Collection
中实现了?
                    --11
4. re: java.util.ConcurrentModifi
cationException
                                remove(Object o)在ArrayList中实现如下:
用remove (index) 方法就不会
                                public boolean remove(Object o) {
有这种情况
                                   if (o == null) {
                  --子墨
                                        for (int index = 0; index < size; index + +)
5. re: Oracle分页SQL
                                     if (elementData[index] == null) {
评论内容较长,点击标题查看
                                        fastRemove(index);
               --软件秀秀
                                        return true;
                                   } else {
   阅读排行榜
                                     for (int index = 0; index < size; index + +)
                                        if (o.equals(elementData[index])) {
1. java.util.ConcurrentModificati
                                           fastRemove(index);
onException(39468)
                                           return true;
2. JAVA集合小结(18945)
3. Oracle分页SQL(7258)
4. WebSphere TLinkageError
                                   return false;
解决办法(1002)
<u>5. struts validate之ValidatorActi</u>
                                private void fastRemove(int index) {
                                   modCount++; //只增加了modCount
onForm(974)
   评论排行榜
                                所以,产生ConcurrentModificationException的原因就是:
1. Oracle分页SQL(7)
                                执行remove(Object o)方法之后,modCount和expectedModCount不相等了。然后当代码执行到next()方法时,判断了checkForComodificati
2. JAVA集合小结(3)
                                on(), 发现两个数值不等, 就抛出了该Exception。
```

- 3. String StringBuffer(2)
- 4. java.util.ConcurrentModificationException(2)
- 5. ThreadLocal学习(1)

# 要避免这个Exception,就应该使用remove()方法。

这里我们就不看add(Object o)方法了,也是同样的原因,但没有对应的add()方法。一般嘛,就另建一个List了

下面是网上的其他解释, 更能从本质上解释原因:

Iterator 是工作在一个独立的线程中,并且拥有一个 mutex 锁。 Iterator 被创建之后会建立一个指向原来对象的单链索引表,当原来的对象数量发生变化时,这个索引表的内容不会同步改变,所以当索引指针往后移动的时候就找不到要迭代的对象,所以按照 fail-fast 原则 Iterator 会马上抛出 java.util.Concurr entModificationException 异常。

所以 Iterator 在工作的时候是不允许被迭代的对象被改变的。但你可以使用 Iterator 本身的方法 remove() 来删除对象, Iterator.remove() 方法会在删除 当前迭代对象的同时维护索引的一致性。

http://gceclub.sun.com.cn/yuanchuang/week-14/iterator.html



posted on 2008-08-31 19:44 EvanLiu 阅读(39469) 评论(2) 编辑 收藏 所属分类: Java基础

# FeedBack:

# re: java.util.ConcurrentModificationException 2011-11-30 03:28 子墨

用remove (index) 方法就不会有这种情况 回复 更多评论

# re: java.util.ConcurrentModificationException[未登录] 2012-03-06 13:58 <u>飞天奔月</u>

@子墨 +1

回复 更多评论

新用户注册 刷新评论列表

### 博问 - 解决您的IT难题

博客园 博问 IT新闻 Java程序员招聘

标题

姓名

主页

内容(请不要发表任何与政治相关的内容)

#### Remember Me?

登录

## [使用Ctrl+Enter键可以直接提交]



### IT新闻:

- · <u>柯达获准出售数字成像专利</u>
- 微软正整合PC、平板和智能机操作系统
- · 图灵之谜-《艾伦·图灵传》序
- Facebook联合创始人休斯与同性男友完婚
- · 诺基亚WP8手机长啥样? 双核概念机出炉

## 博客园首页随笔:

- 利用FlexCell实现的一些报表统计应用
- · [原]JavaScript随手练
- · localStorage兼容方案实现
- 关于javascript中的作用域的一点小记录
- · JavaScript 学习笔记之二:编写自己的jQuery扩展分页插件(分享yQuery)

#### 知识库:

- · 程序员不是一般的人
- 做iOS开发的这2年: 30而立投身iOS开发嫌晚?
- · 图解Git
- 微信架构的启示
- · <u>技术工程师的能力与目标</u>

# 网站导航:

博客园 <u>IT新闻</u> 知识库 <u>C++博客</u> 程序员招聘 管理

#### 相关文章:

WebSphere下LinkageError解决办法

ThreadLocal学习

JMS简单不完全实现

JMS基本概念(转)

JAAS学习

String StringBuffer Oracle分页SOL JAVA集合小结

Copyright ©2012 EvanLiu Powered by: <u>博客园</u> 模板提供: <u>沪江博客</u>