

奋斗的程序猿

梦想、拼搏、求真、务实。www.songaw.com

博客园

新随笔

联系

订阅

管理

随笔 - 108 文章 - 0 评论 - 13

昵称： 不断努力的青春  
园龄： 5年1个月  
粉丝： 51  
关注： 5  
+加关注

< 2020年3月 >						
日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

搜索

找找看

谷歌搜索

积分与排名

积分 - 117343

排名 - 5140

随笔分类

android相关

dubbo相关

Git相关(2)

hibernate相关

HttpClient相关

ibatis相关

Idea相关(2)

java8相关(1)

java基础(12)

Java进阶多线程(7)

Java虚拟机相关(1)

Jenkins相关

jsp相关

kafka相关(3)

面试题思考：java中快速失败(fail-fast)和安全失败(fail-safe)的区别是什么？

一：快速失败（fail—fast）

在用迭代器遍历一个集合对象时，如果遍历过程中对集合对象的内容进行了修改（增加、删除、修改），则会抛出Concurrent Modification Exception。

原理：迭代器在遍历时直接访问集合中的内容，并且在遍历过程中使用一个 modCount 变量。集合在被遍历期间如果内容发生变化，就会改变modCount的值。每当迭代器使用hashNext()/next()遍历下一个元素之前，都会检测modCount变量是否为expectedmodCount值，是的话就返回遍历；否则抛出异常，终止遍历。

注意：这里异常的抛出条件是检测到 modCount! =expectedmodCount 这个条件。如果集合发生变化时修改modCount值刚好又设置为了expectedmodCount值，则异常不会抛出。因此，不能依赖于这个异常是否抛出而进行并发操作的编程，这个异常只建议用于检测并发修改的bug。

场景：java.util包下的集合类都是快速失败的，不能在多线程下发生并发修改（迭代过程中被修改）。

二：安全失败（fail—safe）

采用安全失败机制的集合容器，在遍历时不是直接在集合内容上访问的，而是先复制原有集合内容，在拷贝的集合上进行遍历。

原理：由于迭代时是对原集合的拷贝进行遍历，所以在遍历过程中对原集合所作的修改并不能被迭代器检测到，所以不会触发Concurrent Modification Exception。

缺点：基于拷贝内容的优点是避免了Concurrent Modification Exception，但同样地，迭代器并不能访问到修改后的内容，即：迭代器遍历的是开始遍历那一刻拿到的集合拷贝，在遍历期间原集合发生的修改迭代器是不知道的。

场景：java.util.concurrent包下的容器都是安全失败，可以在多线程下并发使用，并发修改。

另一篇博客：

在我们详细讨论这两种机制的区别之前，首先得先了解并发修改。

1.什么是同步修改？

当一个或多个线程正在遍历一个集合Collection，此时另一个线程修改了这个集合的内容（添加，删除或者修改）。这就是并发修改

2.什么是 fail-fast 机制？

fail-fast机制在遍历一个集合时，当集合结构被修改，会抛出Concurrent Modification Exception。

fail-fast会在以下两种情况下抛出ConcurrentModificationException

（1）单线程环境

集合被创建后，在遍历它的过程中修改了结构。

注意 remove()方法会让expectModcount和modcount 相等，所以是不会抛出这个异常。

（2）多线程环境

当一个线程在遍历这个集合，而另一个线程对这个集合的结构进行了修改。

注意，迭代器的快速失败行为无法得到保证，因为一般来说，不可能对是否出现不同步并发修改做出任何硬性保证。快速失败迭代器会尽最大努力抛出 ConcurrentModificationException。因此，为提高这类迭代器的正确性而编写一个依赖于此异常的程序是错误的做法：迭代器的快速失败行为应该仅用于检测 bug。

3. fail-fast机制是如何检测的？

linux相关(12)
Maven相关(2)
mybatis(1)
mysql(4)
nginx相关
orcale相关
PHP(2)
POI解析相关
redis相关(2)
Springboot相关(5)
springcloud相关(2)
springmvc
Spring基础相关(2)
Struts2相关
Tomcat相关(4)
windows相关
zookeeper相关
分布式事务相关(3)
面试题思考(22)
面试相关(14)
爬虫相关(2)
权限框架相关
算法相关
文件操作相关(1)
项目问题相关(3)

随笔档案
2019年5月(1)
2019年2月(1)
2018年12月(1)
2018年11月(2)
2018年8月(11)
2018年7月(52)

迭代器在遍历过程中是直接访问内部数据的，因此内部的数据在遍历的过程中无法被修改。为了保证不被修改，迭代器内部维护了一个标记“mode”，当集合结构改变（添加删除或者修改），标记“mode”会被修改，而迭代器每次的hasNext()和next()方法都会检查该“mode”是否被改变，当检测到被修改时，抛出Concurrent Modification Exception

下面看看ArrayList迭代器部分的源码

```
private class Itr implements Iterator<E> {
    int cursor;
    int lastRet = -1;
    int expectedModCount = ArrayList.this.modCount;

    public boolean hasNext() {
        return (this.cursor != ArrayList.this.size);
    }

    public E next() {
        checkForComodification();
        /** 省略此处代码 */
    }

    public void remove() {
        if (this.lastRet < 0)
            throw new IllegalStateException();
        checkForComodification();
        /** 省略此处代码 */
    }

    final void checkForComodification() {
        if (ArrayList.this.modCount == this.expectedModCount)
            return;
        throw new ConcurrentModificationException();
    }
}
```

可以看到它的标记“mode”为 expectedModeCount

4. fail-safe机制

fail-safe任何对集合结构的修改都会在一个复制的集合上进行修改，因此不会抛出ConcurrentModificationException

fail-safe机制有两个问题

- (1) 需要复制集合，产生大量的无效对象，开销大
- (2) 无法保证读取的数据是目前原始数据结构中的数据。

5 fail-fast 和 fail-safe的例子

```
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;

public class FailFastExample
{

    public static void main(String[] args)
    {
        Map<String,String> premiumPhone = new HashMap<String,String>();
        premiumPhone.put("Apple", "iPhone");
        premiumPhone.put("HTC", "HTC one");
        premiumPhone.put("Samsung", "S5");

        Iterator iterator = premiumPhone.keySet().iterator();

        while (iterator.hasNext())
        {
            System.out.println(premiumPhone.get(iterator.next()));
            premiumPhone.put("Sony", "Xperia Z");
        }
    }
}
```

2018年6月(26)

2018年5月(14)

最新评论

1. Re:Linux下的tar压缩解压命令详解

不错

--包子铺1234

2. Re:面试题思考: 什么是事务 (AC ID) ?

学习了

--多来哈米

3. Re:面试题思考: java中快速失败(fail-fast)和安全失败(fail-safe)的区别是什么?

@ HNZcodeHashMap是不保证有序的呀...

--weiqiang

4. Re:面试题思考: GET和POST两种基本请求方法的区别

GET和POST本质上没有区别,脱离具体运行环境谈这个区别确实比较坑。浏览器和http服务器的实现上做了具体的限制,而且还略有不同。如果是使用网络编程,GET和POST只是http协议头中的一个me t...

--夜清晨

5. Re:Java 面试题 自己写的答案

nice

--清藻清藻

阅读排行榜

1. linux连接mysql命令(82322)

2. Linux下的tar压缩解压命令详解(52379)

3. Spring中使用RedisTemplate操作Redis (spring-data-redis) (48277)

4. 面试题思考: 解释一下什么叫AOP (面向切面编程) (20161)

5. mysql 查看当前使用的配置文件my.cnf的方法(推荐)(16977)

```
}

}
```

输出

```
iPhone
Exception in thread "main" java.util.ConcurrentModificationException
    at java.util.HashMap$HashIterator.nextEntry(Unknown Source)
    at java.util.HashMap$KeyIterator.next(Unknown Source)
    at FailFastExample.main(FailFastExample.java:20)
```

```
import java.util.concurrent.ConcurrentHashMap;
import java.util.Iterator;

public class FailSafeExample
{

    public static void main(String[] args)
    {
        ConcurrentHashMap<String,String> premiumPhone =
            new ConcurrentHashMap<String,String>();
        premiumPhone.put("Apple", "iPhone");
        premiumPhone.put("HTC", "HTC one");
        premiumPhone.put("Samsung","S5");

        Iterator iterator = premiumPhone.keySet().iterator();

        while (iterator.hasNext())
        {
            System.out.println(premiumPhone.get(iterator.next()));
            premiumPhone.put("Sony", "Xperia Z");
        }

    }

}
```

输出

```
S5
HTC one
iPhone
```

6. fail-fast和 fail-safe 的区别

	Fail Fast Iterator	Fail Safe Iterator
Throw ConcurrentModification Exception	Yes	No
Clone object	No	Yes
Memory Overhead	No	Yes
Examples	HashMap,Vector,ArrayList, HashSet	CopyOnWriteArra

评论排行榜

1. Spring中使用RedisTemplate操作Redis (spring-data-redis) (2)

2. Java 面试题 自己写的答案(2)

3. 面试题思考：java中快速失败(fail-fast)和安全失败(fail-safe)的区别是什么？ (2)

4. 面试题思考：GET和POST两种基本请求方法的区别(2)

5. kafka 并发数配置过程中踩到的坑 InstanceAlreadyExistsException(1)

推荐排行榜

1. 面试题思考：java中快速失败(fail-fast)和安全失败(fail-safe)的区别是什么？ (5)

2. 面试题思考：GET和POST两种基本请求方法的区别(3)

3. 面试题思考：解释一下什么叫AOP（面向切面编程） (2)

4. 面试题思考：什么是 Java 的反射机制(2)

5. 面试题思考：Cookie 和 Session 的区别(1)

yList, ConcurrentHashMap

我的个人首页<http://www.songaw.com>

分类: [面试题思考](#)

好文要顶

关注我

收藏该文

[不断努力的青春](#)  
关注 - 5  
粉丝 - 51  
[+加关注](#)

5

0

« 上一篇: [面试题思考：Cookie 和 Session的区别](#)

» 下一篇: [面试题思考：GET和POST两种基本请求方法的区别](#)

posted @ 2018-07-29 21:27 不断努力的青春 阅读(6022) 评论(2) 编辑 收藏

评论列表

#1楼 2019-08-10 19:35 HNZcode

麻烦问下，为什么安全失败测试用例输出是倒叙？谢谢

支持(0) 反对(0)

#2楼 2019-10-29 23:57 weiqiang

@ HNZcode  
HashMap是不保证有序的呀

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) [网站首页](#)。

- 【推荐】超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】96秒100亿！ 哪些“黑科技”支撑全球最大流量洪峰？
- 【推荐】独家下载电子书 | 前端必看！阿里这样实现前端代码智能生成

相关博文：

- Java的快速失败和安全失败
  - java中fail-fast 和 fail-safe的区别
  - 快速失败(fail-fast)和安全失败(fail-safe)的区别
  - Java集合(16)--快速失败机制(Fail-Fast)
  - Fail-Fast分析
- » 更多推荐...

大咖问答：D2 前端技术专场问答

**最新 IT 新闻:**

- Facebook通知全球4.5万员工停止非重要出差 两名员工感染
  - 重磅! 北京中小微企业2至6月社保单位缴费全部免征
  - 一年卖上亿台, 比手机还赚钱! 无线耳机芯片混战已打响
  - 136亿美元的比特币丢失, 谁是丢币事件里最倒霉的人?
  - 被谷歌出卖定位信息成“嫌疑犯”, 花钱才避免冤屈
- » 更多新闻...

**历史上的今天:**

- 2018-07-29 面试题思考: Cookie 和 Session的区别
- 2018-07-29 面试题思考: BS与CS的区别与联系
- 2018-07-29 面试题思考: 什么是事务 (ACID) ?
- 2018-07-29 面试题思考: 什么是 Java 的反射机制
- 2018-07-29 面试题思考: 什么是 对象/关系 映射集成模块
- 2018-07-29 面试题思考: 什么是基于注解的切面实现? (AOP是Aspect Oriented Program...
- 2018-07-29 面试题思考: Stack和Heap的区别