Iterator学习

Iterator 接口主要用来对Collection集合进行遍历迭代.

主要方法:

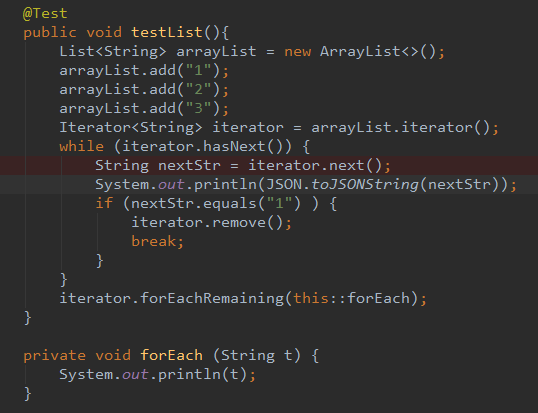
hasNext ()

Next()

Remove()

forEachRemaining()

迭代原理:



首先通过arrayList.iterator() 新建一个迭代器.

内部维护了: cursor 相当于集合的下标 默认为0

lastRet 最后一个元素返回的下标 (-1开始)

expectedModCount = modCount 相当于集合内部元素变化的计数器,迭代过程中如果两者不等,则立马报错

迭代过程:

从0开始只要cursor != size(集合的元素) 进入循环

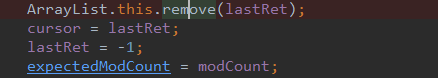
调用next方法获取集合元素,因为cursor从0开始,也就是可以从第一个元素开始获取.返回下标元素,将cursor和lastRet加一.

Remove方法:

说到底调用的还是集合内部的remove方法,但是为什么不会像调用集合的remove方法一样报错呢?

List里面删除元素后,如果不是最后一个元素会调用System.arraycopy()方法拷贝.也就是说下标会减少,后面元素会往前移动.导致循环到最后下标越界.

但是iterator方法的remove调用完之后会做如下处理:



首先更新cursor(相当于指针)因为lastRet是比cursor少1,也就是当集合的元素前移的时候,迭代器的指针也跟着前移.不会引起下标越界,并且将lastRet置为-1,个人理解是为了不能再下次迭代之前再次调用remove方法,因为remove方法有个限制不能为-1.

forEachRemaining方法:

1.8新增方法,用的不多.主要用来返回剩下的集合元素,见上面使用方法.函数式编程.