**如何使用迭代器**

通常情况下，你会希望遍历一个集合中的元素。例如，显示集合中的每个元素。

一般遍历数组都是采用for循环或者增强for，这两个方法也可以用在集合框架，但是还有一种方法是采用迭代器遍历集合框架，它是一个对象，实现了Iterator 接口或 ListIterator接口。

迭代器，使你能够通过循环来得到或删除集合的元素。ListIterator 继承了 Iterator，以允许双向遍历列表和修改元素。

序号 迭代器方法描述

1 使用 Java Iterator

这里通过实例列出 Iterator 和 ListIterator 接口提供的所有方法。

遍历 ArrayList

实例

import java.util.\*;

public class Test{

public static void main(String[] args) {

List<String> list=new ArrayList<String>();

list.add("Hello");

list.add("World");

list.add("HAHAHAHA");

//第一种遍历方法使用 For-Each 遍历 List

for (String str : list) { //也可以改写 for(int i=0;i<list.size();i++) 这种形式

System.out.println(str);

}

//第二种遍历，把链表变为数组相关的内容进行遍历

String[] strArray=new String[list.size()];

list.toArray(strArray);

for(int i=0;i<strArray.length;i++) //这里也可以改写为 for(String str:strArray) 这种形式

{

System.out.println(strArray[i]);

}

//第三种遍历 使用迭代器进行相关遍历

Iterator<String> ite=list.iterator();

while(ite.hasNext())//判断下一个元素之后有值

{

System.out.println(ite.next());

}

}

}

解析：

三种方法都是用来遍历ArrayList集合，第三种方法是采用迭代器的方法，该方法可以不用担心在遍历的过程中会超出集合的长度。

遍历 Map

实例

import java.util.\*;

public class Test{

public static void main(String[] args) {

Map<String, String> map = new HashMap<String, String>();

map.put("1", "value1");

map.put("2", "value2");

map.put("3", "value3");

//第一种：普遍使用，二次取值

System.out.println("通过Map.keySet遍历key和value：");

for (String key : map.keySet()) {

System.out.println("key= "+ key + " and value= " + map.get(key));

}

//第二种

System.out.println("通过Map.entrySet使用iterator遍历key和value：");

Iterator<Map.Entry<String, String>> it = map.entrySet().iterator();

while (it.hasNext()) {

Map.Entry<String, String> entry = it.next();

System.out.println("key= " + entry.getKey() + " and value= " + entry.getValue());

}

//第三种：推荐，尤其是容量大时

System.out.println("通过Map.entrySet遍历key和value");

for (Map.Entry<String, String> entry : map.entrySet()) {

System.out.println("key= " + entry.getKey() + " and value= " + entry.getValue());

}

//第四种

System.out.println("通过Map.values()遍历所有的value，但不能遍历key");

for (String v : map.values()) {

System.out.println("value= " + v);

}

}

}