

深入探讨Iterator模式 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/267/2021_2022__E6_B7_B1_E5_85_A5_E6_8E_A2_E8_c104_267894.htm java.util包中包含了一系列重要的集合类。本文将从分析源码入手，深入研究一个集合类的内部结构，以及遍历集合的迭代模式的源码实现内幕。下面我们先简单讨论一个根接口Collection，然后分析一个抽象类AbstractList和它的对应Iterator接口，并仔细研究迭代子模式的实现原理。本文讨论的源代码版本是JDK 1.4.2，因为JDK 1.5在java.util中使用了很多泛型代码，为了简化问题，所以我们还是讨论1.4版本的代码。集合类的根接口CollectionCollection接口是所有集合类的根类型。它的一个主要的接口方法是：`boolean add(Object c)`add()方法将添加一个新元素。注意这个方法会返回一个boolean，但是返回值不是表示添加成功与否。仔细阅读doc可以看到，Collection规定：如果一个集合拒绝添加这个元素，无论任何原因，都必须抛出异常。这个返回值表示的意义是add()方法执行后，集合的内容是否改变了（就是元素有无数量，位置等变化），这是由具体类实现的。即：如果方法出错，总会抛出异常；返回值仅仅表示该方法执行后这个Collection的内容有无变化。类似的还有：`boolean addAll(Collection c)`.boolean `remove(Object o)`.boolean `removeAll(Collection c)`.boolean `retainAll(Collection c)`.`Object[] toArray()`方法很简单，把集合转换成数组返回。`Object[] toArray(Object[] a)`方法就有点复杂了，首先，返回的Object[]仍然是把集合的所有元素变成的数组，但是类型和参数a的类型是相同的，比如执行：`String[] o`

= (String[])c.toArray(new String[0]).得到的o实际类型是String[]。其次，如果参数a的大小装不下集合的所有元素，返回的将是一个新的数组。如果参数a的大小能装下集合的所有元素，则返回的还是a，但a的内容用集合的元素来填充。尤其要注意的是，如果a的大小比集合元素的个数还多，a后面的部分全部被置为null。最后一个最重要的方法是iterator()，返回一个Iterator（迭代子），用于遍历集合的所有元素。用Iterator模式实现遍历集合Iterator模式是用于遍历集合类的标准访问方法。它可以把访问逻辑从不同类型的集合类中抽象出来，从而避免向客户端暴露集合的内部结构。例如，如果没有使用Iterator，遍历一个数组的方法是使用索引：for(int i=0. i而访问一个链表（LinkedList）又必须使用while循环

：while((e=e.next())!=null) { ... e.data() ... }以上两种方法客户端都必须事先知道集合的内部结构，访问代码和集合本身是紧耦合，无法将访问逻辑从集合类和客户端代码中分离出来，每一种集合对应一种遍历方法，客户端代码无法复用。更恐怖的是，如果以后需要把ArrayList更换为LinkedList，则原来的客户端代码必须全部重写。为解决以上问题，Iterator模式总是用同一种逻辑来遍历集合：for(Iterator it = c.iterator().

it.hasNext().) { ... }奥秘在于客户端自身不维护遍历集合的"指针"，所有的内部状态（如当前元素位置，是否有下一个元素）都由Iterator来维护，而这个Iterator由集合类通过工厂方法生成，因此，它知道如何遍历整个集合。客户端从不直接和集合类打交道，它总是控制Iterator，向它发送"向前"，"向后"，"取当前元素"的命令，就可以间接遍历整个集合。首先看看java.util.Iterator接口的定义：public interface Iterator { boolean

hasNext(). Object next(). void remove().}依赖前两个方法就能完成遍历，典型的代码如下：

```
for(Iterator it = c.iterator().  
it.hasNext(). ) { Object o = it.next(). // 对o的操作... }
```

在JDK1.5中，还对上面的代码在语法上作了简化：

```
// Type是具体的类型，如String。  
for(Type t: c) { // 对t的操作... }
```

每一种集合类返回的Iterator具体类型可能不同，Array可能返回ArrayIterator，Set可能返回SetIterator，Tree可能返回TreeIterator，但是它们都实现了Iterator接口，因此，客户端不关心到底是哪种Iterator，它只需要获得这个Iterator接口即可，这就是面向对象的威力。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com