[**Java提高篇（三八）-----Java集合细节（四）：保持compareTo和equals同步**](http://www.cnblogs.com/chenssy/p/3895471.html)

在Java中我们常使用Comparable接口来实现排序，其中compareTo是实现该接口方法。我们知道compareTo返回0表示两个对象相等，返回正数表示大于，返回负数表示小于。同时我们也知道equals也可以判断两个对象是否相等，那么他们两者之间是否存在关联关系呢？

[复制代码](javascript:void(0);)

public class Student implements Comparable<Student>{ private String id; private String name; private int age; public Student(String id,String name,int age){ this.id = id; this.name = name; this.age = age; } public boolean equals(Object obj){ if(obj == null){ return false; } if(this == obj){ return true; } if(obj.getClass() != this.getClass()){ return false; } Student student = (Student)obj; if(!student.getName().equals(getName())){ return false; } return true; } public int compareTo(Student student) { return this.age - student.age; } /\*\* 省略getter、setter方法 \*/}

[复制代码](javascript:void(0);)

Student类实现Comparable接口和实现equals方法，其中compareTo是根据age来比对的，equals是根据name来比对的。

[复制代码](javascript:void(0);)

public static void main(String[] args){ List<Student> list = new ArrayList<>(); list.add(new Student("1", "chenssy1", 24)); list.add(new Student("2", "chenssy1", 26)); Collections.sort(list); //排序 Student student = new Student("2", "chenssy1", 26); //检索student在list中的位置 int index1 = list.indexOf(student); int index2 = Collections.binarySearch(list, student); System.out.println("index1 = " + index1); System.out.println("index2 = " + index2); }

[复制代码](javascript:void(0);)

按照常规思路来说应该两者index是一致的，因为他们检索的是同一个对象，但是非常遗憾，其运行结果：

index1 = 0index2 = 1

为什么会产生这样不同的结果呢？这是因为indexOf和binarySearch的实现机制不同，indexOf是基于equals来实现的只要equals返回TRUE就认为已经找到了相同的元素。而binarySearch是基于compareTo方法的，当compareTo返回0 时就认为已经找到了该元素。在我们实现的Student类中我们覆写了compareTo和equals方法，但是我们的compareTo、equals的比较依据不同，一个是基于age、一个是基于name。比较依据不同那么得到的结果很有可能会不同。所以知道了原因，我们就好修改了：将两者之间的比较依据保持一致即可。

对于compareTo和equals两个方法我们可以总结为：**compareTo是判断元素在排序中的位置是否相等，equals是判断元素是否相等，既然一个决定排序位置，一个决定相等，所以我们非常有必要确保当排序位置相同时，其equals也应该相等。**

**细节（4.1）：实现了compareTo方法，就有必要实现equals方法，同时还需要确保两个方法同步。**

**参考文献：编写高质量代码：改善Java程序的151个建议**

**-----原文出自:**[**http://cmsblogs.com/?p=1242**](http://cmsblogs.com/?p=1242)**,请尊重作者辛勤劳动成果,转载说明出处.**

**-----个人站点:**[**http://cmsblogs.com**](http://cmsblogs.com/)