

Comparable和Comparator接口

Comparable接口（内比较器）

一、基本介绍

Comparable可以认为是一个内比较器，实现了Comparable接口的类有一个特点，就是这些类是可以和自己比较的。

Comparable是排序接口。若一个类实现了Comparable接口，就意味着该类支持排序。实现了Comparable接口的类的对象的列表或数组可以通过Collections.sort或Arrays.sort进行自动排序。此外，实现此接口的对象可以用作有序映射中的键或有序集合中的集合，无需指定比较器。该接口定义如下：

```
public interface Comparable<T>
{
    public int compareTo(T o);
}
```

compareTo方法的返回值是int，有三种情况：

- 1、比较者大于被比较者（也就是compareTo方法里面的对象），那么返回正整数
- 2、比较者等于被比较者，那么返回0
- 3、比较者小于被比较者，那么返回负整数

Comparator接口（外比较器）

一、基本介绍

Comparator是比较接口，我们如果需要控制某个类的次序，而该类本身不支持排序(即没有实现Comparable接口)，那么我们就可以建立一个“该类的比较器”来进行排序，这个“比较器”只需要实现Comparator接口即可。也就是说，我们可以通过实现Comparator来新建一个比较器，然后通过这个比较器对类进行排序。

//进行类之间的排序

```
public interface Comparator<T>
{
    int compare(T o1, T o2);
    boolean equals(Object obj);
}
```

Comparator可以认为是是一个外比较器，个人认为有两种情况可以使用实现Comparator接口的方式：

- 1、一个对象不支持自己和自己比较（没有实现Comparable接口），但是又想对两个对象进行比较
- 2、一个对象实现了Comparable接口，但是开发者认为compareTo方法中的比较方式并不是自己想要的那种比较方式

二、可以通过实现Comparator接口重写compare () 来做个自定义比较器

Comparator接口里面有一个compare方法，方法有两个参数T o1和T o2，是泛型的表示方式，分别表示待比较的两个对象，方法返回值和Comparable接口一样是int，有三种情况：

- 1、o1大于o2，返回正整数，降序排序
- 2、o1等于o2，返回0
- 3、o1小于o2，返回负整数，升序排序