<https://mp.weixin.qq.com/s?__biz=MzIyNDU2ODA4OQ==&mid=2247484013&idx=1&sn=57910d41e513a5ae9e6f089725e05d7c&chksm=e80db41bdf7a3d0d13080dd12114d7567c44e9e81df07ceec3b6010fbad33a85ff29cce12f6c&scene=21#wechat_redirect>

# 你能说说Java中Comparable和Comparator的区别吗

Java 中为我们提供了两种比较机制：Comparable 和 Comparator，二者都是用来实现对象的比较、排序。

下面分别对Comparable 和 Comparator做具体介绍并总结。

## Comparable

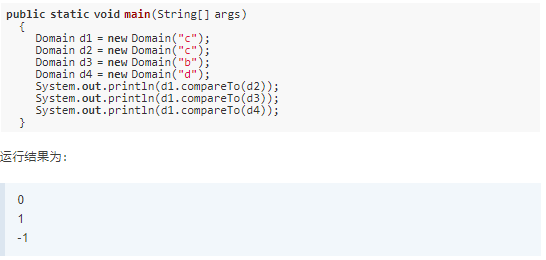
Comparable可以认为是一个内比较器，实现了Comparable接口的类有一个特点，就是这些类是可以和自己比较的，至于具体和另一个实现了Comparable接口的类如何比较，则依赖compareTo方法的实现。

如果add进入一个Collection的对象想要Collections的sort方法帮你自动进行排序的话，那么这个对象必须实现Comparable接口。compareTo方法的返回值是int，有三种情况：

* 比较者大于被比较者，返回正整数
* 比较者等于被比较者，返回0
* 比较者小于被比较者，返回负整数

写个很简单的例子：

**public** **class** **Domain** **implements** **Comparable**<**Domain**>  
{  
   **private** String str;  
  
   **public** Domain(String str)  
   {  
       **this**.str = str;  
   }  
  
   **public** int compareTo(Domain domain)  
   {  
       **if** (**this**.str.compareTo(domain.str) > 0)  
           **return** 1;  
       **else** **if** (**this**.str.compareTo(domain.str) == 0)  
           **return** 0;  
       **else**   
           **return** -1;  
   }  
  
   **public** String getStr()  
   {  
       **return** str;  
   }  
}



注意一下，前面说实现Comparable接口的类是可以支持和自己比较的，但是其实代码里面Comparable的泛型未必就一定要是Domain，将泛型指定为String或者指定为其他任何任何类型都可以，只要开发者指定了具体的比较算法就行。

Comparator

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：

* o1大于o2，返回正整数
* o1等于o2，返回0
* o1小于o3，返回负整数

写个很简单的例子：



因为泛型指定死了，所以实现Comparator接口的实现类只能是两个相同的对象（不能一个Domain、一个String）进行比较了，实现Comparator接口的实现类一般都会以"待比较的实体类+Comparator"来命名

## 总结

如果实现类没有实现Comparable接口，又想对两个类进行比较（或者实现类实现了Comparable接口，但是对compareTo方法内的比较算法不满意），那么可以实现Comparator接口，自定义一个比较器，写比较算法。

实现Comparable接口的方式比实现Comparator接口的耦合性要强一些，如果要修改比较算法，要修改Comparable接口的实现类，而实现Comparator的类是在外部进行比较的，不需要对实现类有任何修改。因此：

* 对于一些普通的数据类型（比如 String, Integer, Double…），它们默认实现了Comparable 接口，实现了 compareTo 方法，我们可以直接使用。
* 而对于一些自定义类，它们可能在不同情况下需要实现不同的比较策略，我们可以新创建 Comparator 接口，然后使用特定的 Comparator 实现进行比较。

**不同之处：**

个人感觉说出上文观点，这个提问就算回答完了，如果非要说不同之处，那就是：

* Comparator位于java.util包下，而Comparable位于java.lang包下
* 实现Comparable接口的方式比实现Comparator接口的耦合性要强
* 等等………..