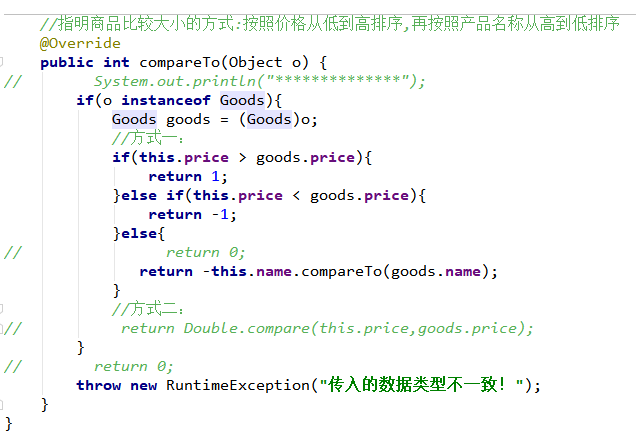
## 说明：

Java中的对象，正常情况下，只能进行比较：== 或 != 。不能使用 > 或 < 的 但是在开发场景中，我们需要对多个对象进行排序，言外之意，就需要比较对象的大小。  
如何实现？使用两个接口中的任何一个：Comparable 或 Comparator

## 自然排序

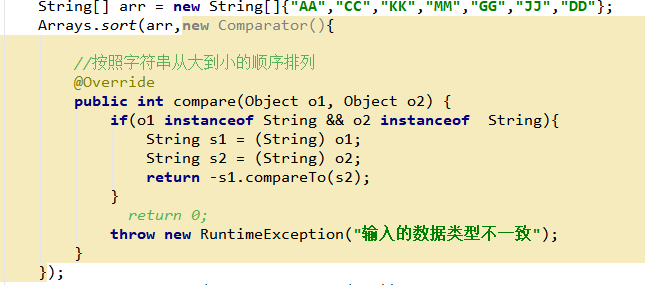
Comparable接口的使用举例： 自然排序  
1.像String、包装类等实现了Comparable接口，重写了compareTo(obj)方法，给出了比较两个对象大小的方式。  
2.像String、包装类重写compareTo()方法以后，进行了从小到大的排列  
3. 重写compareTo(obj)的规则：  
 如果当前对象this大于形参对象obj，则返回正整数，  
 如果当前对象this小于形参对象obj，则返回负整数，  
 如果当前对象this等于形参对象obj，则返回零。  
4. 对于自定义类来说，如果需要排序，我们可以让自定义类实现Comparable接口，重写compareTo(obj)方法。在compareTo(obj)方法中指明如何排序





## 定制排序

Comparator接口的使用：定制排序  
1.背景：  
当元素的类型没有实现java.lang.Comparable接口而又不方便修改代码，  
或者实现了java.lang.Comparable接口的排序规则不适合当前的操作，  
那么可以考虑使用 Comparator 的对象来排序  
2.重写compare(Object o1,Object o2)方法，比较o1和o2的大小：  
如果方法返回正整数，则表示o1大于o2；  
如果返回0，表示相等；  
返回负整数，表示o1小于o2。

创建Comparator的匿名实现类对象，重写compare方法