Comparator,Comparable 接口区别

收集整理：by Volunteer　　2008 年 12 月 1 日 星期一

需求：

做一个对象排序的功能,需要按不同规则排序.

分析：

查看 Arrays,Collections 的 sort 方法,了解 Comparator,Comparable 两个接口,找到适用的排序办法.

整理：

- comparable 是通用的接口，用户可以实现它来完成自己特定的比较；
- 而 comparator 可以看成一种算法的实现，在需要容器集合 collection 需要比较功能的时候，来指定这个比较器，这可以看出一种设计模式，将算法和数据分离，就像 C++ STL 中的函数对象一样。
  - ☺ 前者应该比较固定，和一个具体类相绑定；而后者比较灵活，它可以被用于各个需要比较功能的类使用。
  - ☺ 可以说前者属于“静态绑定”，而后者可以“动态绑定”。
- 一个类实现了 Camparable 接口表明这个类的对象之间是可以相互比较的。如果用数学语言描述的话就是这个类的对象组成的集合中存在一个全序。这样，这个类对象组成的集合就可以使用 Sort 方法排序了。
- 而 Comparator 的作用有两个：
  1. 如果类的设计师没有考虑到 Compare 的问题而没有实现 Comparable 接口，可以通过 Comparator 来实现比较算法进行排序
  2. 为了使用不同的排序标准做准备，比如：升序、降序或其他什么序。

代码：

请参考老师上课讲解的代码，自己敲一敲，好好体会体会。

程序演示见详情：

详情：

Comparable & Comparator 都是用来实现集合中的排序的，只是 Comparable 是在集合内部定义的方法实现的排序，Comparator 是在集合外部实现的排序，所以，如想实现排序，就需要在集合外定义 Comparator 接口的方法或 在集合内实现 Comparable 接口的方法。

具体请看 〈Thinking in java〉

- Comparable 是一个对象本身就已经支持自比较所需要实现的接口（如 String Integer 自己就可以完成比较大小操作）
- 而 Comparator 是一个专用的比较器，当这个对象不支持自比较或者自比较函数不能满足你的要求时，你可以写一个比较器来完成两个对象之间大小的比较。

可以说一个是自己完成比较，一个是外部程序实现比较的差别而已。

- 用 Comparator 是策略模式（strategy design pattern），就是不改变对象自身，而用一个策略对象（strategy object）来改变它的行为。

比如：你想对整数采用绝对值大小来排序，Integer 是不符合要求的，你不需要去修改 Integer 类（实际上你也不能这么做）去改变它的排序行为，只要使用一个实现了 Comparator 接口的对象来实现控制它的排序就行了。

java 代码

```
//AbsComparator.java
import java.util.*;


public class AbsComparator implements Comparator {
    public int compare(Object o1, Object o2) {
        int v1 = Math.abs(((Integer)o1).intValue());
        int v2 = Math.abs(((Integer)o2).intValue());
        return v1 > v2 ? 1 : (v1 == v2 ? 0 : -1);
    }
}
```

可以用下面这个类测试 AbsComparator：

```
//Test.java
import java.util.*;

public class Test {
    public static void main(String[] args) {

        //产生一个 20 个随机整数的数组（有正有负）
         Random rnd = new Random();
        Integer[] integers = new Integer[20];
         for(int i = 0; i < integers.length; i++) {
            integers[i] = new Integer(rnd.nextInt(100) *
(rnd.nextBoolean() ? 1 : -1));
        }

        System.out.println("用 Integer 内置方法排序：");
        Arrays.sort(integers);
        System.out.println(Arrays.asList(integers));

        System.out.println("用 AbsComparator 排序：");
        Arrays.sort(integers,    new    AbsComparator());
        System.out.println(Arrays.asList(integers));
    }
}
```

现在能看出 Comparator 和 Comparable 的区别了吧？

什么？？还没头绪!!

那你得继续琢磨琢磨：）