## 排序算法类模板

“排序算法类模板”中的Example类展示了习惯约定：我们会将排序代码放在类的sort方法中，该类还将包含辅助函数less()和exch()（可能还有其它辅助函数）以及一个示例用例main()。Example类还包含了一些早期调试使用的代码：测试用例main()将标准输入得到的字符串排序，并用私有方法，show()打印字符数组的内容。

大多数情况下，排序代码只会通过两个方法发操作数据：less()方法对元素进行比较，exch()方法将元素交换位置。Exch()方法的实现很简单，通过Comparable接口实现less()方法也不困难。

排序算法类模板如下：

1. **package** com.lagoon.template;
2. /\*\*
3. \* @Author WinkiLee
4. \* @Date 2019/4/10 8:52
5. \* @Description
6. \*/
7. **public** **class** Template {
9. **public** **static** **void** sort(Comparable[] a){
10. /\*\*
11. \* 算法编写区域
12. \*/
13. }
15. /\*\*
16. \* 比较大小
17. \* @param v
18. \* @param w
19. \* @return false or true
20. \*/
21. **private** **static** **boolean** less(Comparable v,Comparable w){
22. **return** v.compareTo(w)<0;
23. }
25. /\*\*
26. \* 交换数
27. \* @param a
28. \* @param i
29. \* @param j
30. \*/
31. **private** **static**  **void** exch(Comparable[] a,**int** i,**int** j){
32. Comparable t=a[i];
33. a[i]=a[j];
34. a[j]=t;
35. }
37. /\*\*
38. \* 打印结果
39. \* @param a
40. \*/
41. **private**  **static** **void** show(Comparable[] a){
42. //在单行中打印数组
43. **for** (**int** i=0;i<a.length;i++){
44. System.out.println(a[i]+"");
45. System.out.println();
46. }
47. }
49. /\*\*
50. \* 测试元素数组是否有序
51. \* @param a
52. \* @return true or false
53. \*/
54. **public** **static** **boolean** isSorted(Comparable[] a){
55. //测试元素数组是否有序
56. **for** (**int** i=1;i<a.length;i++){
57. **if** (less(a[i],a[i-1])){
58. **return** **false**;
59. }
60. }
61. **return** **true**;
62. }
64. /\*\*
65. \* main方法及测试
66. \* @param args
67. \*/
68. **public** **static** **void** main(String[] args) {
69. //测试比较大小
70. **if** (less(3,5)==**true**){
71. System.out.println("排序正确，3比5小");
72. }
74. //测试交换
75. Comparable[] a ={1,2,3,5,4,6,7};
76. exch(a,3,4);
77. **for** (**int** v=0;v<a.length;v++){
78. System.out.print(a[v]);
79. }
81. //测试有序
82. **if** (isSorted(a)){
83. System.out.println("该数组有序");
84. }
85. **else** {
86. System.out.println("该数组无序");
87. }
88. }
89. }