

# 后端

## Java08

### task1: 你的任务——按照字母顺序对歌曲排序

首先是读题目，任务其实很简单。

刚读完题觉得只需要在Comparator里的compare方法里实现读取第一个字母用他们的ascii码升序排列就可以。

但是歌单里的歌名不是全部是字母欸，（想来歌名本就可以不是字母），所以题目要求照字母顺序排序是否有一些歧义，我就先假设“1”和“\$”也是“字母”来排序了，如果我理解错了改题的学长Q我一下！

首先是自己重构一个Comparator接口，查阅了一些资料，在这里总结一下基本流程（主要感觉东西越学越多，有点理不清楚了）：

1. 接口声明（据传：任何类默认实现了equals（Object obj），所以可以不用写出）

```
public interface Compatator<T>{  
    int compare(T a,T b);  
}
```

2. 用Comparator实现方法：compareTest:

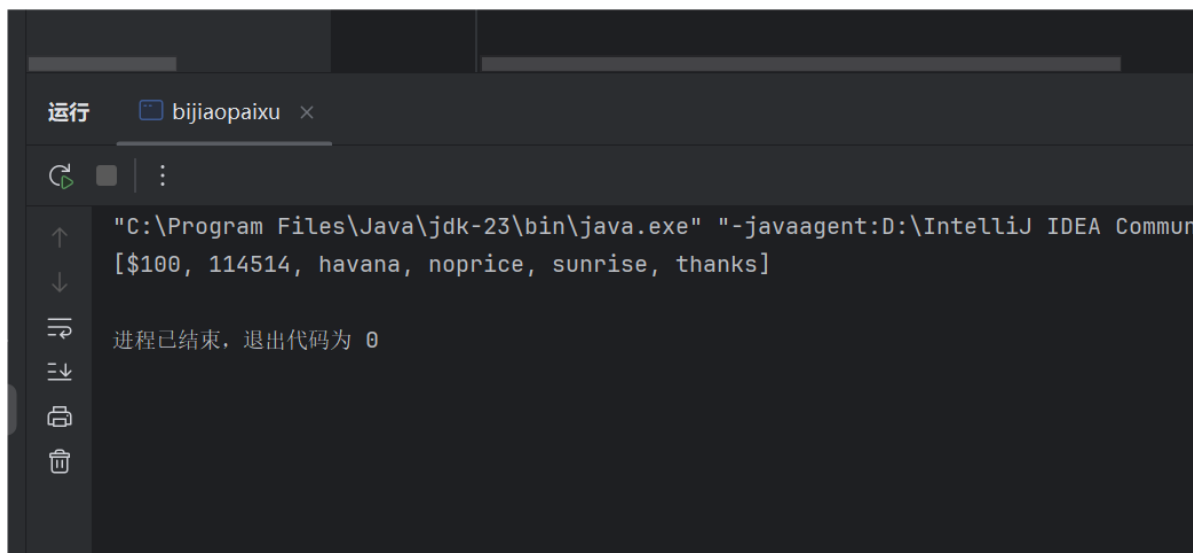
```
public static List<String> compareTest(List<String> songs){  
    songs.sort(new Comparator<String>() {  
        @Override  
        public int compare(String o1, String o2) {  
            char a = o1.charAt(0);  
            char b = o2.charAt(0);  
            return (int) a < (int) b ? -1 : a == b ? 0 : 1;//a<b返回-1, a==b返回0, a>b返回1  
        }  
    });  
    return songs;  
}
```

其实我之前不知道接口里的方法还能在new完之后再定义，那如果只用作这一个题排序的话感觉写个Comparator类完全也是可以的。不过感觉每次做新的招新题都会有很多语法不太熟悉，要看着别人代码才能跟上，菜还是要多练啊。感觉只是跟着招新题做基础还是很不牢，等有时间了把各种基本的方法整理一下！

3. 主函数

```
public static void main(String[] args) {  
    MockSongs songs=new MockSongs();  
    List<String> a=songs.getSongStrings();  
    compareTest(a);  
    System.out.println(a);  
}
```

好了到这里这个简单，但我敲得艰辛的代码就完成了 ✨,°.☆(¯▽¯)/\$.° ✨。完整代码会一同上传 github，文件名为Java08题task1答题代码



```
运行  bijiaopaixu x
C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:D:\IntelliJ IDEA Commun
[$100, 114514, havana, noprice, sunrise, thanks]
进程已结束，退出代码为 0
```

## task2：进阶挑战——加入对象

看到这个倒是松了一口气，因为在task1的资料查阅里大部分都是以加入了对象的例子来讲解的，在后端的前几题中，我有了解泛型，这里算是继续学习了。

首先是@注解，据资料显示，注解应该是用于对代码进行一些有功能的注释（区别于我们“//”无功能的注释）可以告诉电脑一些对类，方法的信息。这里罗列三个便于理解：

@Data：注解于实类上，直接为类提供了一些方法，比如：get，set，equals，hashCode，canEqual，toString（顺便也查阅了这几个方法的作用）

@Override - 注释于方法，检查该方法是否是重写方法。如果发现其父类，或者是引用的接口中并没有该方法时，会报编译错误。

@AllArgsConstructo：注解于类，自动生成一个包含所有参数的构造函数（这个看上去就能用得我很爽）

好到这里代码应该时没问题了，我们把代码写出来！（每次再IDEA上敲代码都有一种油然而生的快感，为什么它能这么好用！）

代码已上传github

遇到的问题：

1. 在过程中我在尝试是否能用泛型去定义一个compareTest方法使得我可以在compareTest<>里通过输入Integer或String来做到用bpm或歌名排序，但是这样之后compare方法的实现一直报错，我查阅资料没找到解决办法，于是作罢，选择写三个方法分别表示三种排列顺序。
2. 再来，我在comparator泛型使用代替了task1中的依然有报错，最后放弃自己重写comparator中的compare了，转而使用现有的comparing方法来排序。（总的来说，在试错上花费了大量的时间，罪过罪过）

然后是对题中问题的回答：

多态和泛型类型的结合是怎么实现的，如果你有一个ArrayList< music >引用变量，能把一个ArrayList< Song >赋给它吗？一个ArrayList< game >呢？

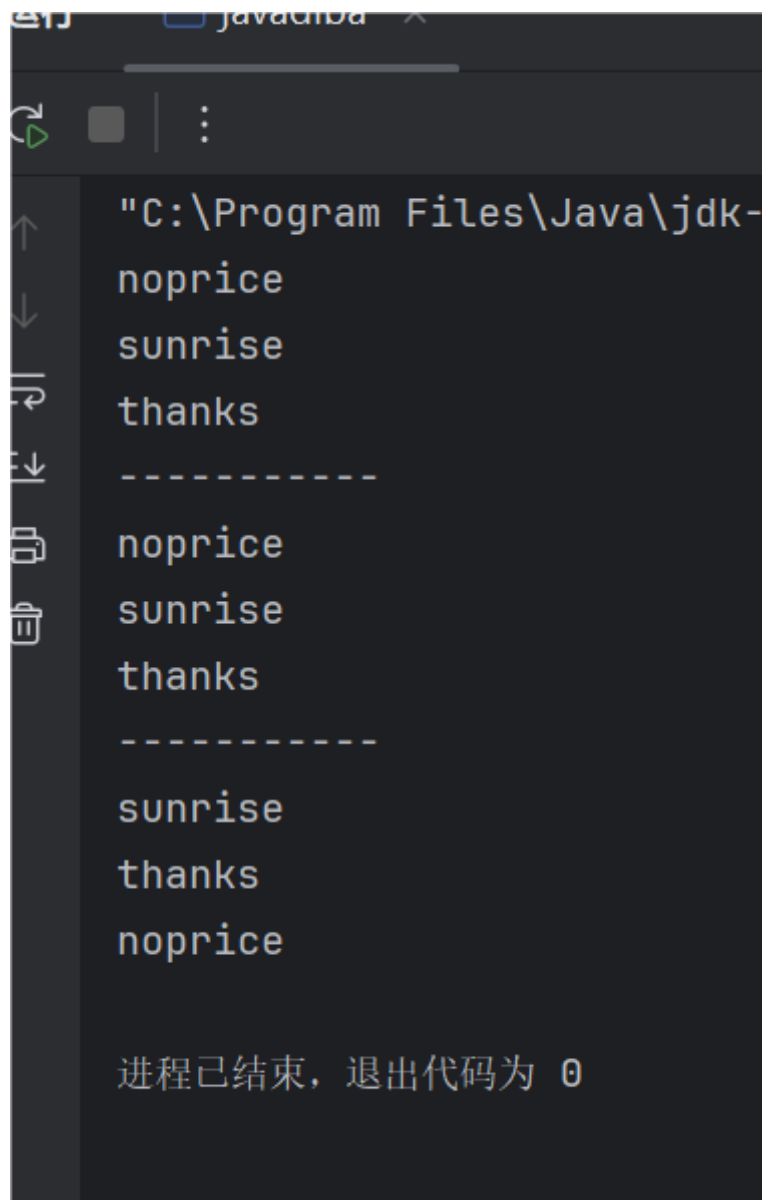
不加其他条件的话，对于ArrayList，不能把一

个ArrayList赋给它，除

非Song是Music的子类（而且还需要进行类型转换）。所以也不能将ArrayList赋给ArrayList理由同上。

大概根据文字体现了一个代码出来，实际意义不大，就只贴在此处了。

```
class Music {}
class Song extends Music {}
ArrayList<Music> musicList;
ArrayList<Song> songList = new ArrayList<>();
// 如果 Song 是 Music 的子类，可以通过类型转换赋值
// musicList = (ArrayList<Music>) songList;
class Game {}
ArrayList<Game> gameList = new ArrayList<>();
// 以下赋值是错误的，因为 Game 和 Music 没有继承关系
// musicList = gameList;
```

A screenshot of a Java IDE window titled 'javaciba'. The main editor area displays a list of strings: "C:\Program Files\Java\jdk-", noprice, sunrise, thanks, followed by a dashed line separator, then noprice, sunrise, thanks, another dashed line separator, then sunrise, thanks, noprice. At the bottom, a message states '进程已结束，退出代码为 0'. The left sidebar shows standard IDE icons for navigation and development.

```
"C:\Program Files\Java\jdk-
noprice
sunrise
thanks
-----
noprice
sunrise
thanks
-----
sunrise
thanks
noprice

进程已结束，退出代码为 0
```

由于才疏学浅，这些回答偏向个人理解，如有错误，请学长雅正。

XY 致上

