

## 08问题处理思路

### 简单挑战、你的任务

#### 1.api是什么？

查询资料：

api是应用程序编程接口，相当于交流的桥梁，是一些预先定义的函数

list的api即为特定list中的固定功能，可以通过固定指令实现

#### 2.comparator是什么？

[Java Comparator使用指南 ---- 看这一篇就够了 .thencomparing\(new-CSDN博客\)](#)

comparator是一个比较接口，我们可以通过重写这个接口实现我们想要的数值比较并根据比较结果对数值进行排序

*注：song的输出格式参考了这篇文章*

### 编写代码过程中遇到的问题及解决方案

#### 1.根据要求创建Song类，查询网络后希望通过compareTest（list）命令实现对Song中对象的排序

```
public class Main {  
    public static void main(String[] args) {  
        compareTest(list);  
        for(Song s : list){  
            System.out.println(s.toString());  
        }  
    }  
}
```

#### 2.查询资料，要想比较对象中特定字段，需要重构compare方法

```
@Override  
public int compare(Song o1, Song o2) {  
    String a = o1.title;  
    String b = o2.title;  
    return Collator.getInstance(Locale.ENGLISH).compare(o1,o2);  
}
```

模仿网上教程重构的compare方法。

这里按idea给的建议用的Locale.English，后来才发现不对。

此时的完整代码（Song从始至终几乎没变，只标示了Main）

```
import java.text.Collator;  
import java.util.*;  
  
public class Main {
```

```

public static void main(String[] args) {

    Main.compareTest();

}

public static void compareTest() {
    List<Song> list = new ArrayList<>();
    list.add(new Song("qndsehdhj","ncisuhdkhv",237));
    list.add(new Song("djinnccs","jiad",38937));
    list.add(new Song("cilaiif","efjjbh",129));
    list.add(new Song("uisbhb","bwiuhdd",91));
    list.add(new Song("hsjxniiuuuhzxh","iouixb",9772));

    Collections.sort(list,new Comparator<Song>() {

        @Override
        public int compare(Song o1, Song o2) {
            String a = o1.title;
            String b = o2.title;
            return Collator.getInstance(Locale.ENGLISH).compare(o1,o2);
        }
    });
    System.out.println(list);
}
}

```

3.然后又报错了，根据提示导入了包

```
import java.util.Comparator;
```

4.反馈异常ClassCastException

这个我查了好久也没明白为什么，最后在网上看到有人代码里用的是uk，但是改完了依然报异常。

```
Collator.getInstance(Locale.ENGLISH).compare(o1,o2);
```

改成了

```
Collator.getInstance(Locale.UK).compare(o1,o2);
```

然后开始觉得是compareTest的问题，把它改成了非静态方法。

5.异常没了，但是无法从静态上下文引用非静态方法。

创建了Main的实例a，用于引用方法。

6.又报了异常ClassCastException

终于发现了比较的对象错了，应该比较两个对象的title字段而不是两个对象

7.终于没有异常了，但是现在输出的格式不对：

```
[Song@728938a9, Song@21b8d17c, Song@6433a2, Song@5910e440, Song@6267c3bb]
```

在Song里添加了print方法，实例化了一个Song对象来调用

8.输出的是实例化对象中的字段，并不是list

查询资料，最终改为

```
for (Song s:list){  
    System.out.println(s.print());  
}
```

其它

对思考题2的理解：

不可以。泛型arraylist是list的子类型，但是Arraylist<music>和Arraylist<Song>/Arraylist<game>都是并列关系

参考：[泛型，继承和子类型(Java™ 教程-Java Tutorials 中文版>学习 Java 语言>泛型(更新))

(pingfangx.github.io)](<https://pingfangx.github.io/java-tutorials/java/generics/inheritance.html#:~:text=泛型类和子类型>)

你可以通过继承泛型类或实现泛型接口来对其进行子类型化。

一个类或接口的类型形参与另一个的类型形参之间的关系由 extends 和 implements 子句确定。以, Collections 类为例， ArrayList 实现 List， List extends Collection。)

思考题3：

泛型可以减少代码量，提高可读性，并且安全高效

参考：[JAVA泛型是什么?有什么优缺点?](#)

[适合哪些使用场景? 请简要阐述什么是java泛型及其优点?-CSDN博客](#)