

# Java - 08

## 数据结构的简单了解

**排序对Java来说是小菜一碟。**你可以使用各种工具来收集和管理你的数据，而不用写你自己的排序算法。Java集合框架（Java Collections Framework）中有一个数据结构应该能适用于你需要做的几乎任何事情。无论你是想要维护一个可以轻松增加元素的列表？想按名字查找列表，还是想要自动去除所有重复元素，都可以用它....

## 简单挑战——排序问题

假设你现在是一名负责管理酒吧点唱机的程序员，你的任务是负责将收到的歌曲数据进行排序，其他人会负责将数据封装到一个List中。而你现在有一个模拟类来测试你的代码：

```
public class MockSongs {
    public static List<String> getSongStrings(){
        List<String> songs = new ArrayList<>();
        //模拟将要处理的列表
        songs.add("sunrise");
        songs.add("noprice");
        songs.add("thanks");
        songs.add("$100");
        songs.add("havana");
        songs.add("114514");
        return songs;
    }
}
```

## 你的任务——按照字母顺序对歌曲排序

嗯哼，看起来有些难，不是么。所以对于初学的同学，这里还有一些tips。

**tip:**正如前面所说，你其实并不需要实现一个排序算法，我们检查**java.util.List**和**java.util.Collections**的API

**java.util.List:**

`sort(Comparator)`:根据**Comparator**指定的顺序对这个列表排序

**java.util.Collections:**

`sort(List)`:根据元素的自然顺序

`sort(List,Comparator)`:根据**Comparator**指定的顺序对指定列表排序

如果对**Comparator**感兴趣，你可以考虑自己重构一个**Comparator**

## 进阶挑战——加入对象

在前面的task中，我们在List中加入了String作为存储的歌曲名字，现在我们需要加入更多的信息，于是我们引入Song对象来存储数据

```
@Data
@AllArgsConstructor //这里的注解代表类的构造器（全参）以及所有数据的get和set函数,建议自己
写代码实现构造器和get
public class Song{
    private String title;
    private String artist;
    private int bpm;
}
```

你可能对@这类的注解感兴趣，如果你有多余的精力，可以去了解一下关于@这类注解的用法和效果，这对你以后做项目会很有帮助

你可以仿造上面的模拟类来构建一个新的模拟类，用于模拟歌曲的数据，我们的要求是列表中包含真正的Song实例，而不是简单的String。

**tips:** 注意泛型的使用，为了完成这个任务，你需要了解什么是泛型。这是一个比较复杂的问题，所以推荐你有选择地学习一下以下的博客：[java 泛型详解](#)，[Java 中的泛型](#)。

在学习的过程中你可能会了解泛型的很多特性和用法，但是对于大多数程序员来说，真正重要的只有其中三点：

1. 创建一个泛型类（如ArrayList）的实例：创建一个ArrayList的时候，你需要告诉它你希望在这个列表中存放什么样的对象
2. 泛型类型变量的声明和赋值：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？
3. 声明和调用接受泛型类型的方法：面对一个泛型方法，你需要了解的它使用泛型的目的，以及使用泛型的意义。这涉及到一些复杂的多态问题，但很不幸的是，这些都是你们以后需要了解的。

这里只是以List为例了解一下数据结构，实际上我们还经常使用map和set这些抽象数据结构。

要求：将实现的代码和运行截图推送到GitHub仓库上，此处提交链接：

出题人QQ：[1727448271@qq.com](mailto:1727448271@qq.com)