

简单挑战——排序问题

先去百度了Collection是啥 Collection接口具有定义所有单例集合共性的方法，下面有两个子接口，一个是List接口，一个是Set接口。

List接口

它有Vector，ArrayList，LinkedList三个形式，实现方式有数组，也有链表。特点：

1. 有序的集合，即存储和取出元素的顺序相同
2. 允许存储重复的元素
3. 有索引，可以使用普通的for循环遍历

Set接口

有TreeSet，HashSet，LinkedHashSet，前两者是无序，最后一个是有序 特点：

1. 不允许存储重复的元素
2. 没有索引，不能使用普通的for循环遍历

Comparator接口

定义如下：

```
public interface Comparator<T> {  
    int compare(T o1, T o2);  
    boolean equals(Object obj);  
}
```

若一个类要实现Comparator接口：它一定要实现compareTo(T o1, T o2) 函数，但可以不实现 equals(Object obj) 函数(对于任何类，已经默认实现了)。int compare(T o1, T o2) 是“比较o1和o2的大小”。当o1<o2时return -1, o1=o2时return 0, o1 > o2时return 1.这样，我们就可以借助其原理来进行排序了，并且我们也能仿造这个来构造其他排序方法。

task1.代码和运行截图都放在github了

泛型

1. 泛型的本质是为了将类型参数化，从而限制类型的传入，使用泛型时，java在编译时就能进行检测并反应出错误，泛型的作用只在编译上，不参与运行
2. 多态和泛型类型的结合是怎么实现的：通过使用泛型，我们可以使接在同一个接口的类在使用方法时可以用不同的数据类型，从而满足我们的需求，并且由于泛型的性质，我们不能把ArrayList< Song >一个ArrayList< game >赋给ArrayList< music >
3. 使用泛型可以提高编程的安全性，避免运行出错，通过泛型可以减少重复繁琐的内容，使代码整洁，具有可读性

task2 代码和截图也放在github了
