

Java 数据结构-引入实例版

前后对比：

相较于单纯的 String 类型 List，这个 List 为一个真正的 Song 实例 list。

在 List<> 中间的类型由 String 替换成了 Song1（我的代码中用到 song1）

在 Song 类中添加了构造方法，getSongName 方法，通过该方法返回歌曲名。

在 Compare 方法中参数类型换成了 Song1，o1.getSongName().substring(0,1) 会提取 getSongName() 中字符串中第一个字符来进行排序比较。

因为 list 中引入了 song 实例，和先前 System.out.println(song) 相比，后来 System.out.println(song) 则会打出路径名称，体现了对象和 String 的区别。

```
public class MockSongs { 6个用法 新*
    public static List<Song1> getSongStrings() { 2个用法 新*
        List<Song1> songs = new ArrayList<>();
        songs.add(new Song1( songName: "This Night", artistName: "***
        songs.add(new Song1( songName: "Alice", artistName: "Dreamlan
        songs.add(new Song1( songName: "11111", artistName: "***"));
        songs.add(new Song1( songName: "TTT", artistName: "***"));
        songs.add(new Song1( songName: "R2222", artistName: "***"));
        songs.add(new Song1( songName: "S3333", artistName: "***"));
        Collections.sort(songs, new Comparator<Song1>() { 新*
            @Override 新*
            public int compare(Song1 o1, Song1 o2) {
                return o1.getSongName().substring(0,1).toUpperCase();
            }
        });
        for (Song1 song : songs) {
            System.out.println(song.getSongName());
        }

        return songs;
    }
}
```

```
package com.java.data;

public class Demo2 { 新*
    public static void main(String[] args) { 新*
        MockSongs m = new MockSongs();
        MockSongs.getSongs();
    }
}
```

```
.encoding=UTF-8 -classpath /Users/duqiu/Desktop/java/javapro/out/produ
11111
Alice
R2222
S3333
This Night
TTT

进程已结束 退出代码为 0
```