

# 2024090905014-林仪-Java-08

## 1. 比较器

```
public class Main {  
    public static void main(String[] args) {  
        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("hava");  
        songs.add("114514");  
        songs.sort(new Comparator<String>() { //自定义比较器使得字母顺序前的在前面  
            @Override  
            public int compare(String o1, String o2) {  
                for (int i = 0; i < Math.min(o1.length(), o2.length()); i++) {  
                    if (o1.charAt(i) - (o2.charAt(i)) != 0) {  
                        return o1.charAt(i) - o2.charAt(i);  
                    } //循环比较每一个字符  
                    else  
                        i++;  
                    if (i == Math.min(o1.length(), o2.length())-1) {  
                        return o1.length() - o2.length();  
                        //在前面字符一致的情况下使长的排在后面  
                    }  
                }  
                return 0;  
            }  
        });  
        songs.forEach(System.out::println);  
  
    }  
}
```

```
C:\Users\Administra
$100
114514
hava
havana
noprice
sunrise
thanks
```

比较器返回两个字符串的差值会根据这个值对元素排序

## 2. 泛型

泛型可以确定集合里面放什么元素，用ArrayList<T.>的方式，T可为任意类或接口

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        ArrayList<music> musics = new ArrayList<>();
        musics.add(new song("111","wo"));
        musics.add(new song("222","wo"));
        musics.forEach(System.out::println);
        ArrayList<game> games = new ArrayList<>();
        games.add(new game(){
            @Override
            public void play() {}
        });
        games.forEach(System.out::println);
    }
}

class music{
    String name;
```

```

String artist;

@Override
public String toString() {
    return "music{" +
        "name='" + name + '\'' +
        ", artist='" + artist + '\'' +
        '}';
}

public music(String name, String artist) {
    this.name = name;
    this.artist = artist;
}
}

class song extends music{
    public song(String name, String artist) {
        super(name, artist);
    }
}

interface game{
    void play();
}

```

```

C:\Users\Administrator\Documents\jdk8\openj
music{name='111', artist='wo'}
music{name='222', artist='wo'}
Main$1@3d494fbf

Process finished with exit code 0

```

<T.>中的T.指定后可以放入其子类或者实现类

<.? extend A>可以限制类型为A及其子类

<.? super A>可以限制类型为A及其父类