

原 jdk8 sorted 排序值comparing 和thenComparing 偶遇问题解析

2018年03月02日 14:32:50

阅读数: 1632

标签:

java8排序

sorted

comparing

thenComparing

[更多](#)

在java8之前, 排序有两种方法:

1.定义比较器类 继承Comparator接口, 然后重写compare (a, b) 方法, 返回的结果是int类型

0 代表 a==b

1 代表 a>b

-1 代表 a<b

2.定义比较器对象

```
1 Comparator<Person> c1 = new Comparator<Person>() {  
2     @Override  
3     public int compare(Person o1, Person o2) {  
4         return o1.getAge() - o2.getAge();  
5     }  
6 };  
7  
8 Collections.sort(personList, c1);
```

好了, 前面的排序就讲到这了, 不是我要讲解的重点。

dk8中, 排序可以使用stream中的sorted。

```
ersonList.stream().sorted(Comparator.comparing((Person::getAge)).collect(Collectors.toList())); //按照年龄有小到大排序
```

```
personList.stream().sorted(Comparator.comparing(Person::getAge).thenComparing(Person::getId)).collect(Collectors.toList()) //先按年龄从小到大排序，年龄相同再按id从小到大排序
```

再看看comparing和thenComparing的源码

```
public static <T, U extends Comparable<? super U>> Comparator<T> comparing(
    Function<? super T, ? extends U> keyExtractor)
{
    Objects.requireNonNull(keyExtractor);
    return (Comparator<T> & Serializable)
        (c1, c2) -> keyExtractor.apply(c1).compareTo(keyExtractor.apply(c2));
}
```

```
default <U extends Comparable<? super U>> Comparator<T> thenComparing(
    Function<? super T, ? extends U> keyExtractor)
{
    return thenComparing(comparing(keyExtractor));
}
```

default 就不说了，是一个jdk8的新特性，不明白的请自行google。

初次看这个两个源码，发现他们传入的参数类型是一模一样的，但是请注意返回类型可不一样，为什么这里要突显出来。还是请看代码：

```
personList.stream().sorted(Comparator.comparing(( p) -> p.getCars().size()).thenComparing(Person::getId)).collect(Collectors.toList());
```

代码在p.getCars.size() 处报错，cannot resolve method "getCars()"。这里一定有人会纠结为什么我写的也是函数式编程，按道理没问题呀！但是编辑器就是不认识p.getCars()。

解：在使用comparing的时候p没有指定类型，比较的时候就不知道是哪一种类型，自然也找不到这个类的方法。代码在编译的时候不通过，idea语法检查也不通过。...下代码就行，指定了p的类型Person。

```
personList.stream().sorted(Comparator.comparing((Person p) -> p.getCars().size()).thenComparing(Person::getId)).collect(Collectors.toList());
```

当然如果将comparing里面的代码分开来写，看的会更加清楚，特别指定了类型是Person。

```
Function<Person, Integer> function = p -> -p.getCars().size();  
personList = personList.stream().sorted(Comparator.comparing(function).thenComparing(Person::getId)).collect(Collectors.toList());
```

完整代码如下：

```
public class Car {  
    private String carName;  
  
    public String getCarName() {  
        return carName;  
    }  
  
    public void setCarName(String carName) {  
        this.carName = carName;  
    }  
  
    @Override  
    public String toString() {  
        return "Car{" +  
            "carName='" + carName + '\'' +  
            '}';  
    }  
}
```

```
public class Person {  
    private int id;  
    private String name;  
  
    private List<Car> cars;  
  
    public List<Car> getCars() {  
        return cars;  
    }  
  
    public void setCars(List<Car> cars) {  
        this.cars = cars;  
    }  
}
```

```
}

public Person(int id, String name) {
    this.id = id;
    this.name = name;
}

Person() {
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public static void main(String[] args) {
    Person p1 = new Person(9, "aa");
    p1.setCars(new ArrayList<>());
    Car car10 = new Car();
    car10.setCarName("car10");
    Car car11 = new Car();
    car11.setCarName("car11");
    p1.getCars().add(car10);
    p1.getCars().add(car11);
}
```

```
Person p2 = new Person(3, "aa");
p2.setCars(new ArrayList<>());
Car car20 = new Car();
car20.setCarName("car20");
Car car21 = new Car();
car21.setCarName("car21");
Car car23 = new Car();
car23.setCarName("car23");
p2.getCars().add(car20);
p2.getCars().add(car21);
p2.getCars().add(car23);

List<Person> personList = new ArrayList<>();
personList.add(p1);
personList.add(p2);
Function<Person, Integer> function = p -> -p.getCars().size();
personList = personList.stream().sorted(Comparator.comparing(function).thenComparing(Person::getId)).collect(Collectors.toList());
System.out.println(personList);
}

@Override
public String toString() {
    return "Person{" +
        "id=" + id +
        ", name='" + name + '\'' +
        ", cars=" + cars +
        '}';
}
}
```

了，今天就分享到这里，不知大家有没有注意到，-p.getCars.size() 可以按照Person里面Car的数量逆序排序。里面的东西还有很多，待你发现哦~



《C语言/C++学习指南》语法篇（从入门到精通）

学习指南

一门零基础、从入门到精通的C语言C++语法教程，由毕业于清华大学的业内人士执课。从最简单的HelloWorld入门程序，到最深入的C语言C++核心概念，均为您娓娓道来，言之必详、听之必懂。让C语言C++编程变得简单，让C语言C++编程变得有趣，让喜欢C语言C++的人学会C语言C++！



想对作者说点什么

Java Comparator的双重比较问题

538

Java Comparator的双重比较问题类型推导失败

JAVA8-用lamda表达式和增强版Comparator进行排序

8088

以前的排序一般对象实现Comparable或者Comparator接口，经常是通过匿名类实现。可以参见以前的博文 J...

[Java 8] (3) 字符串遍历和Comparator接口

1.1万

遍历一个字符串 在Java 8中，CharSequence接口新添加了一个方法叫做chars()，方法的签名是这个样子的： p...

java8与并行

540

申明式的编程方式 不需要提供明确的指令操作，所有的细节指令被程序库封装，只要提出要求，申明用意[java] ...

Java8 Comparator的深入学习

1540

```
package com.lgx.jdk8.part02; import java.util.*; import java.util.stream.Collectors; /** * Compar...
```

Java 8 语言变化

732

Java 8 的最大变化在于添加了对 lambda 表达式 的支持。Lambda 表达式是可按引用传递的代码块。它们类似...



征婚-找个程序猿当男朋友

想找个合适的男朋友，谈以结婚为目的的恋爱

va源码-字符串大小比较(Comparing Strings)

1623

这个例子，发现了JavaDoc中的一个问题： compareTo方法返回的结果按理和两个字符串在字典中的先后位...

Comparable 与 Comparator 比较

2714

1.两个接口的原型Java中，Comparable与Comparator接口都是用来做比较的。那么这两个接口在实际使用中到...

java8 Comparator

 1035

集合最简排序

Java8：Lambda表达式增强版Comparator和排序

 743

1、概述 在这篇教程里，我们将要去了解下已经发布的JDK8中的Lambda表达式——特别是怎样使用它来编写C...

相关热词

jdk8和 jdk8万年历 jdk8堆栈 jdk8函数式 jdk8时间

JDK8中的排序

 120

```
import java.util.ArrayList; import java.util.Collections; import java.util.Comparator; import java.u...
```

再谈Comparator-使用lambda表达式

 1673

关于Comparable和Comparator 的区别以及作用: 1. Java 中 Comparable 和 Comparator 比较 - 如果天空不死 - ...

Comparable接口中compareTo用法

 4471

compareTo就是比较两个值，如果前者大于后者，返回1，等于返回0，小于返回-1，我下面给出了例子，由于比...

Java 自定义比较器

 465

Comparator 和 Comparable

lambda实体属性去重，对实体的某个属性进行去重

 3256

下面我将介绍一下如何为一个List<People> 这样的List里面的某个属性进行去重。顺便介绍一下...

t 集合排序,java8新特性,学习一下

 2241

5做一个项目,遇到了list<Entity>;需要按时间倒叙排列发现Java8新特性封装的比较好:例子:List<...

java stream sorted排序 考虑null值

 552

项目里使用到排序, java里没有像C# 里的linq, 只有stream,查找stream.sorted源码看到有个Comparator.nullsLas...

Java8 sort map by key or value

 1652

1. Sort by Key SortByKeyExample.java package com.mkyong.test; import java.util.HashMap; import jav...

Java List 多维度排序 jdk8

 72

在工作中遇到了需要按某一实体类多维度排序的场景。（Hbase备份Oracle字段 检索查询后order by操作） 获取...

Python sorted 函数，对字典按值进行排序

 1243

字典实际上并不能排序，我们排的是items，即dict.items()，将字典转换成了一个列表sorted函数的原型为：sort...

个人资料



归海一刀之渔...

关注

原创	粉丝	喜欢	评论
52	8	2	4

等级：  3 访问： 1万+

积分： 745 排名： 7万+

勋章： 

最新文章

- 基于redis实现分布式限流器-pdlr
- 手写springmvc--PMVC
- 一个字段首字母大小写引发空值
- 如何理解阻塞，非阻塞，同步，异步
- 实际开发中两个有意思的优化

个人分类

hadoop2	1篇
java	47篇
java, 多线程	8篇
git	1篇
ivm	7篇

展开

归档

2018年8月	1篇
2018年7月	1篇
2018年6月	6篇
2018年5月	3篇
全年4月	31篇

展开

热门文章

- JVM参数MetaspaceSize的误解

阅读量：2651

jdk8 sorted 排序值comparing 和thenComparing 偶遇问题解析

阅读量：1622

教你代码实现抢红包功能

阅读量：971

什么是java中的CAS

阅读量：911

如果有一个500G的超大文件，里面都是数值，如何对这些数值排序？

阅读量：740

最新评论

记一次生产问题--Completa...
percy0601：写的真心不错，CompletableFuture确实在开发过程中方便了很多，但是实际生产难免会出问...

java concurrent包下...
u011381576：[reply]wzgandwj2010[/reply] 这里不加锁就是来凸显CyclicBarri...

java concurrent包下...
wzgandwj2010：尽量严谨点，多线程对sum操作，可否加个锁

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 💬 客服论坛

关于 招聘 广告服务 网站地图

©2018 CSDN版权所有 京ICP证09002463号

🐾 百度提供搜索支持

0

- 经营性网站备案信息
- 网络110报警服务
- 中国互联网举报中心
- 北京互联网违法和不良信息举报中心

0