

Java汉字排序(3)按笔划排序

2015-11-02 原文

对于包含汉字的字符串来说，排序的方式主要有两种：一种是拼音，一种是笔画。

本文就讲述如何实现按笔划排序的比较器(Comparator)。

作者：[Jeff](#) 发表于：2007年12月21日 11:27 最后更新于：2007年12月21日 12:38

版权声明：可以任意转载，转载时请务必以超链接形式标明文章[原始出处](#)和作者信息及本
版权声明。

<http://www.blogjava.net/jeff-lau/archive/2007/12/21/169257.html>

笔画排序

要按笔画排序，就要实现笔画比较器。

```
class StokeComparator implements Comparator<String>
```

如果有个方法可以求得汉字的笔画数，上面的功能就很容易实现。如何求一个汉字的笔画数？最容易想到的就是查表法。建一个汉字笔画数表，如：

汉字Unicode编码笔画数

一	U4E00	1
二	U4E8C	2
龍	U9F8D	16
...

表二

如果是连续的、按unicode编码排好顺序的表，实际存储在笔画数表中的只需最后一列就够了。

那如何建这个表呢？这个表存储在哪里？

建汉字笔画数表

现在大多数系统还只能支持Unicode中的基本汉字那部分汉字，编码从U9FA6-U9FBF。所以我们只建这部分汉字的笔画表。汉字笔画数表，我们可以按照下面的方法生成：

1. 用java程序生成一个文本文件(Chinese.csv)。包括所有的从U9FA6-U9FBF的字符的编码和文字。利用excel的按笔画排序功能，对Chinese.csv文件中的内容排序。
2. 编写Java程序分析Chinese.csv文件，求得笔画数, 生成ChineseStroke.csv。矫正笔画数，重新按汉字的Unicode编码对ChineseStroke.csv文件排序。
3. 只保留ChineseStroke.csv文件的最后一列，生成Stroke.csv。

在这里 [下载上面3个步骤生成的3个文件](#)。

生成Chinese.csv的Java程序

```
1.  /**
2.   * @author Jeff
3.   *
4.   * Copyright (c) 复制或转载本文，请保留该注释。
5.   */
6.  package chinese.utility.preface;
7.
8.  import java.io.IOException;
9.  import java.io.PrintWriter;
10.
11.  public class ChineseCoder {
12.
13.      public static void main(String[] args) throws
IOException {
14.          PrintWriter out = new PrintWriter("Chinese.csv");
15.          // 基本汉字
16.          for (char c = 0x4E00; c <= 0x9FA5; c++) {
17.              out.println((int) c + "," + c);
18.          }
19.          out.flush();
20.          out.close();
21.
22.      }
23.
24.  }
```

初始化笔画数

从Excel排序过后的Chinese.csv文件来看，排好序的文件还是有一定规律的。在文件的第9行-12行可以看出：逐行扫描的时候，当unicode会变小了，笔画数也就加1。

20059,冫

20101, J

19969,丁

19970,丂

用下面的Java程序分析吧。

```
1.    /**
2.     * @author Jeff
3.     *
4.     * Copyright (c) 复制或转载本文， 请保留该注释。
5.     */
6.    package chinese.utility.preface;
7.
8.    import java.io.File;
9.    import java.io.IOException;
10.   import java.io.PrintWriter;
11.   import java.util.Scanner;
12.
13.   public class Stroke {
14.
15.       /**
16.        * @param args
17.        * @throws IOException
18.        */
19.       public static void main(String[] args) throws
IOException {
20.           Scanner in = new Scanner(new
File("Chinese.csv"));
21.           PrintWriter out = new
PrintWriter("ChineseStroke.csv");
22.           String oldLine = "";
23.           ;
24.           while (in.hasNextLine()) {
25.               String line = in.nextLine();
26.               ) {
27.                   stroke++;
28.               }
29.               oldLine = line;
30.               out.println(line + "," + stroke);
31.           }
32.           out.flush();
```

```
33.         out.close();
34.         in.close();
35.     }
36.
37. }
```

上面用的这个规律有问题吗？有问题，从ChineseStroke.csv文件抽取最后几个汉字就发现，笔画数不对。为什么呢？

- 笔画数可能不是连续的。
- n+1笔画数的最小Unicode码可能比n笔画数的最大Unicode码要大

我们要人工核对ChineseStroke文件，但只要核对在笔画变化的那几个汉字的笔画数。最后，我发现，只有笔画数多于30的少数几个汉字的笔画数不对。核对并矫正笔画数后，用Excel按Unicode重新排序，去掉汉字和Unicode两列，只保留笔画数那列，得到Stroke.csv文件。

求得笔画数的方法和笔画比较器方法

求得笔画数的方法测试代码：

```
1.  /**
2.   * @author Jeff
3.   *
4.   * Copyright (c) 复制或转载本文，请保留该注释。
5.   */
6.  package chinese.utility.test;
7.
8.  import static org.junit.Assert.assertEquals;
9.
10. import org.junit.Before;
11. import org.junit.Test;
12. import chinese.utility.Chinese;
13.
14. public class StrokeTest {
15.
16.     Chinese chinese;
17.
18.     @Before
19.     public void setUp() {
20.         chinese = new Chinese();
21.     }
22.
23.     @Test
24.     public void testStroke() {
```

```
25.         assertEquals(, chinese.stroke('一'));
26.     }
27.
28.     @Test
29.     public void testStroke2() {
30.         assertEquals(, chinese.stroke('二'));
31.     }
32.
33.     @Test
34.     public void testStroke16() {
35.         assertEquals(, chinese.stroke('龍'));
36.     }
37.
38.     @Test
39.     public void testStrokeABC() {
40.         assertEquals(-, chinese.stroke('a'));
41.     }
42.
43. }
```

求得笔画数的方法代码

```
1.  /**
2.   * @author Jeff
3.   *
4.   * Copyright (c) 复制或转载本文, 请保留该注释。
5.   */
6.  package chinese.utility;
7.
8.  import java.util.Comparator;
9.
10. public class StrokeComparator implements
11. Comparator<String> {
12.
13.     public int compare(String o1, String o2) {
14.
15.         Chinese chinese = new Chinese();
16.
17.         ; i < o1.length() && i < o2.length(); i++) {
18.             int codePoint1 = o1.codePointAt(i);
19.             int codePoint2 = o2.codePointAt(i);
20.             if (codePoint1 == codePoint2)
21.                 continue;
22.
23.             int stroke1 = chinese.stroke(codePoint1);
24.             int stroke2 = chinese.stroke(codePoint2);
25.
26.             || stroke2 < ) {
27.                 return codePoint1 - codePoint2;
28.             }
29.         }
30.     }
31. }
```

```
28.
29.         if (stroke1 != stroke2) {
30.             return stroke1 - stroke2;
31.         }
32.     }
33.
34.     return o1.length() - o2.length();
35. }
36. }
```

笔画比较器测试

```
1.  /**
2.   * @author Jeff
3.   *
4.   * Copyright (c) 复制或转载本文，请保留该注释。
5.   */
6.  package chinese.utility.test;
7.
8.  import java.util.Comparator;
9.
10. import org.junit.Assert;
11. import org.junit.Before;
12. import org.junit.Test;
13.
14. import chinese.utility.StrokeComparator;
15.
16. public class StrokeComparatorTest {
17.
18.     private Comparator<String> comparator;
19.
20.     @Before
21.     public void setUp() {
22.         comparator = new StrokeComparator();
23.     }
24.
25.     /**
26.     * 相同笔画数
27.     */
28.     @Test
29.     public void testCompareEquals() {
30.         Assert.assertTrue(comparator.compare());
31.     }
32.
33.     /**
34.     * 不同笔画数
35.     */
36.     @Test
37.     public void testCompare() {
38.         Assert.assertTrue(comparator.compare());
```

```

39.         Assert.assertTrue(comparator.compare());
40.     }
41.
42.     /**
43.      * 长度不同
44.      */
45.     @Test
46.     public void testCompareDefficultLength() {
47.         Assert.assertTrue(comparator.compare());
48.     }
49.
50.     /**
51.      * 非汉字的比较
52.      */
53.     @Test
54.     public void testABC() {
55.         Assert.assertTrue(comparator.compare());
56.         Assert.assertTrue(comparator.compare());
57.     }
58. }

```

笔画比较器

```

1.     /**
2.      * @author Jeff
3.      *
4.      * Copyright (c) 复制或转载本文，请保留该注释。
5.      */
6.     package chinese.utility.test;
7.
8.     import java.util.Comparator;
9.
10.    import org.junit.Assert;
11.    import org.junit.Before;
12.    import org.junit.Test;
13.
14.    import chinese.utility.StrokeComparator;
15.
16.    public class StrokeComparatorTest {
17.
18.        private Comparator<String> comparator;
19.
20.        @Before
21.        public void setUp() {
22.            comparator = new StrokeComparator();
23.        }
24.
25.        /**
26.         * 相同笔画数
27.         */

```

```

28.         @Test
29.         public void testCompareEquals() {
30.             Assert.assertTrue(comparator.compare());
31.         }
32.
33.         /**
34.          * 不同笔画数
35.          */
36.         @Test
37.         public void testCompare() {
38.             Assert.assertTrue(comparator.compare());
39.             Assert.assertTrue(comparator.compare());
40.         }
41.
42.         /**
43.          * 长度不同
44.          */
45.         @Test
46.         public void testCompareDefficultLength() {
47.             Assert.assertTrue(comparator.compare());
48.         }
49.
50.         /**
51.          * 非汉字的比较
52.          */
53.         @Test
54.         public void testABC() {
55.             Assert.assertTrue(comparator.compare());
56.             Assert.assertTrue(comparator.compare());
57.         }
58.     }

```

其他程序的汉字排序

Microsoft在这方面做得比较好。如Sql server 2000，Word和Excel都能按拼音和笔画排序。而Oracle只能是采取宽松拼音排序法。

Java汉字排序(3)按笔划排序的更多相关文章

1. Java汉字排序(2)按拼音排序

对于包含汉字的字符串来说,排序的方式主要有两种:一种是拼音,一种是笔画. 本文就讲述如何实现按拼音排序的比较器(Comparator). 作者:Jeff 发表于:2007年12月21日 11:27 最 ...

2. Java汉字排序(1)排序前要了解的知识 (数组和list的排序接口)

对于包含汉字的字符串来说,排序的方式主要有两种:一种是拼音,一种是笔画. 本文就讲述如何实现按拼音排序的比较器(Comparator). 作者:Jeff 发表于:2007年12月21日 11:27 最 ...

3. Java面试宝典系列之基础排序算法

本文就是介绍一些常见的排序算法.排序是一个非常常见的应用场景,很多时候,我们需要根据自己需要排序的数据类型,来自定义排序算法,但是,在这里,我们只介绍这些基础排序算法,包括:插入排序.选择排序.冒泡排 ...

4. Java常见排序算法之Shell排序

在学习算法的过程中,我们难免会接触很多和排序相关的算法.总而言之,对于任何编程人员来说,基本的排序算法是必须要掌握的. 从今天开始,我们将要进行基本的排序算法的讲解.Are you ready?Let ...

5. Java比较器对数组, 集合排序一

数组排序非常简单,有前辈们的各种排序算法,再加上Java中强大的数组辅助类Arrays与集合辅助类Collections,使得排序变得非常简单,如果说结合比较器Comparator接口和Collato ...

6. java结构与算法之选择排序

一 .java结构与算法之选择排序(冒择路兮快归堆) 什么事选择排序:从一组无序数据中选择出中小的的值,将该值与无序区的最左边的的值进行交换. 简单的解释:假设有这样一组数据 12,4,23,5,找到 ...

7. 使用C语言和Java分别实现冒泡排序和选择排序

经典排序算法--冒泡和选择排序法 Java实现冒泡排序 基本思想是,对相邻的元素进行两两比较,顺序相反则进行交换,这样,每一趟会将最小或最大的元素放到顶端,最终达到完全有序,首先看个动图: 我们要清楚 ...

8. 【java多线程系列】java内存模型与指令重排序

在多线程编程中,需要处理两个最核心的问题,线程之间如何通信及线程之间如何同步,线程之间通信指的是线程之间通过何种机制交换信息,同步指的是如何控制不同线程之间操作发生的相对顺序.很多读者可能会说这还不简 ...

9. 【Java学习笔记之十一】Java中常用的8大排序算法详解总结

分类: 1)插入排序(直接插入排序.希尔排序) 2)交换排序(冒泡排序.快速排序) 3)选择排序(直接选择排序.堆排序) 4)归并排序 5)分配排序(基数排序) 所需辅助空间最多:归并排序 所需辅助空 ...

随机推荐

1. 用C++开发Binder服务

用C++来实现Binder服务比较麻烦,原因是没有AIDL的辅助,必须手工来写中间的代码. 首先写一个服务类ExampleServer的代码: `class ExampleServer : public ...`

2. http缓存相关头

<https://mp.weixin.qq.com/s/qOMO0LIdA47j3RjhbCWUEQ> 这里说的一下我对http控制客户端缓存的头的理解. 在请求一个静态文件的时候(图片,css,js)等 ...

3. 在Visual Lisp中处理自动化错误

Handling Automation errors in Visual LISP 翻译自原文Kean's blog:<http://through-the-interface.typepad.com/> ...

4. SQL Server的“错误：9004”

客户的一台现场采集电脑崩溃,无法启动.用PE引导后,将MDF和LDF数据复制出来,往正常的数据库附加时,报如下错误: 处理数据库 'databasename' 的日志时出错.如果可能,请从备份还原.如 ...

5. DEDECMS里面DEDE函数解析

下面来解说下DEDECMS织梦CMS模板里面的函数说明 在文件 `include/inc_function.php`里面 `GetCurUrl()` 获贴切前的脚本的URL `GetAlabNum($str)` 把 ...

6. Microsoft Office Excel 不能访问文件及COM无法访问

Microsoft Office Excel 不能访问文件及COM无法访问 Microsoft Office Excel 不能访问文件“*.xls”. 可能的原因有: 1 文件名称或路径不存在. 2 ...

7. SQL SELECT语句

基本SQL SELECT语句 1. 下面的语句是否可以执行成功 `select ename , job , sal as salary from emp;` 2. 下面的语句 ...

8. 【CUDA学习】全局存储器

全局存储器,即普通的显存,整个网格中的任意线程都能读写全局存储器的任意位置. 存取延时为400-600 clock cycles 非常容易成为性能瓶颈. 访问显存时,读取和存储必须对齐,宽度为4By ...

9. 推荐算法——距离算法

本文内容 用户评分表 曼哈顿(Manhattan)距离 欧式(Euclidean)距离 余弦相似度(cos simliarity) 推荐算法以及数据挖掘算法,计算"距离"是必须的~ ...

10. 安卓开发-问题集-Description Resource Path Location TypeUnparsed aapt error(s)!

Check the console for output.

今天在安卓项目中 `res-drawable-hdpi` 替换图片的时候出现这个问题 问题现象为项目显示一个叉号,但是在项目内容的任何文件都不显示叉号,搞了半天没发现是什么问题,然后就去项目的文件夹下 ...

[Home](#)

Powered By [WordPress](#)