

## 关键术语

hash map (散列映射表)	set (集合)
hash set (散列集)	read-only view(只读视图)
linked hash map (链式散列映射表)	tree map (树形映射表)
linked hash set (键式散列集)	tree set (树形集)
map (映射表)	

## 本章小结

1. 集合存储的是不重复的元素。若要在合集中存储重复的元素,需要使用线性表。
2. 映射表中存储的是键/值对。它提供使用键快速查询一个值。
3. Java 合集框架支持三种类型的集合:散列集 `HashSet`、链式散列集 `LinkedHashSet` 和树形集 `TreeSet`。`HashSet` 以一个不可预知的顺序存储元素;`LinkedHashSet` 以元素被插入的顺序存储元素;`TreeSet` 存储已排好序的元素。`HashSet`、`LinkedHashSet` 和 `TreeSet` 中的所有方法都继承自 `Collection` 接口。
4. `Map` 接口将键映射到元素上。键类似于索引。`List` 中,索引为整数。`Map` 中,键可以为任何对象。映射表不能包含相同的键。每个键可以映射最多一个值。`Map` 接口提供了查询、更新以及获取值的合集以及键的合集的方法。
5. Java 合集框架支持三种类型的映射表:散列映射表 `HashMap`、链式散列映射表 `LinkedHashMap` 和树形映射表 `TreeMap`。对于定位一个值、插入一个条目和删除一个条目而言,`HashMap` 是很高效的。`LinkedHashMap` 支持映射表中的条目排序。`HashMap` 类中的条目是没有顺序的,但 `LinkedHashMap` 中的条目可以按某种顺序来获取,该顺序既可以是它们被插入映射表中的顺序(称为插入顺序),也可以是它们最后一次被访问的时间的顺序,从最早到最晚(称为访问顺序)。对于遍历排好序的键,`TreeMap` 是高效的。键可以使用 `Comparable` 接口来排序,也可以使用 `Comparator` 接口来排序。

## 测试题

回答位于网址 [www.cs.armstrong.edu/liang/intro10e/quiz.html](http://www.cs.armstrong.edu/liang/intro10e/quiz.html) 的本章测试题。

## 编程练习题

### 21.2 ~ 21.4 节

- 21.1 (在散列集上进行集合操作) 创建两个链接散列集合 `{"George","Jim","John","Blake","Kevin","Michael"}` 和 `{"George","Katie","Kevin","Michelle","Ryan"}`, 然后求它们的并集、差集和交集。(可以先备份一份这些集合,以防随后进行的集合操作改变原来的集合。)
- 21.2 (按升序显示不重复的单词) 编写一个程序,从文本文件中读取单词,并将所有不重复的单词按升序显示。文本文件被作为命令行参数传递。
- \*\*21.3 (统计 Java 源代码中的关键字) 修改程序清单 21-7 中的程序。如果关键字在注释或者字符串中,则不进行统计。将 Java 文件名从命令行传递。假设 Java 源代码是正确的,行注释和段落注释不会交叉。
- \*21.4 (统计元音和辅音) 编写一个程序,提示用户输入一个文本文件名,然后显示文件中的元音和辅音的数目。使用一个集合存储元音 A、E、I、O 和 U。
- \*\*\*21.5 (突出显示语法) 编写一个程序,将一个 Java 文件转换为一个 HTML 文件。在 HTML 文件中,关键字、注释和字面量分别用粗体的深蓝色、绿色和蓝色显示。使用命令行传递 Java 文件和 HTML 文件。例如,下面的命令

```
java Exercise21_05 Welcome.java Welcome.html
```

将 Welcome.java 转换为 Welcome.html。图 21-8a 显示了一个 Java 文件，它对应的 HTML 文件如图 21-8b 所示。

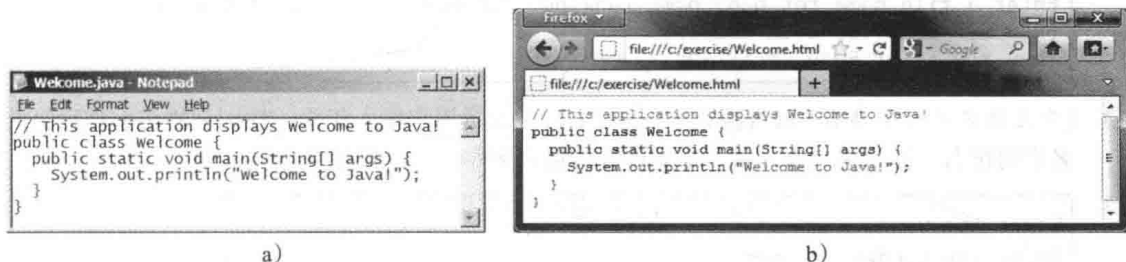


图 21-8 a 中纯文本形式的 Java 代码被显示在 b 中的 HTML 中，其中突出了它的语法

## 21.5 ~ 21.7 节

- \*21.6 (统计输入数字的个数) 编写一个程序，读取个数不定的整数，然后查找其中出现频率最高的数字。当输入为 0 时，表示结束输入。例如，如果输入的数据是 2 3 40 3 5 4 -3 3 3 2 0，那么数字 3 的出现频率是最高的。如果出现频率最高的数字不是一个而是多个，则应该将它们全部报告。例如，在线性表 9 30 3 9 3 2 4 中，3 和 9 都出现了两次，所以 3 和 9 都应该被报告。
- \*\*21.7 (改写程序清单 21-9) 改写程序清单 21-9，将单词按出现频率的升序显示。  
(提示：创建一个名为 WordOccurrence 的类实现 Comparable 接口。这个类包含两个域：word 和 count。使用 compareTo 方法比较单词的出现次数。对程序清单 21-9 散列集中的每个对，创建 WordOccurrence 的一个实例，并把它储存到一个数组线性表中。使用 Collections.sort 方法对该数组线性表进行排序。如果将 WordOccurrence 的实例存入树形集，会发生什么错误？)
- \*\*21.8 (统计文本文件中单词的出现频率) 改写程序清单 21-9，从文本文件中读取文本，文本文件名被作为命令行参数传递。单词由空格、标点符号(,;.:?)、引号('')以及括号分隔。统计单词不区分大小写(例如，认为 Good 和 good 是一样的单词)。单词必须以字母开头。以单词的字母顺序显示输出，每个单词前面显示它的出现次数。
- \*\*21.9 (使用映射表猜首府) 改写编程练习题 8.37，在映射表中存储州和它的首府的条目。你的程序应该提示用户输入一个州，然后显示这个州的首府。
- \*21.10 (统计每个关键字的出现次数) 重写程序清单 21-7，读入一个 Java 源代码文件并且统计文件中每个关键字的出现次数。如果关键字是在注释中或者字符串面值中，则不要进行统计。
- \*\*21.11 (婴儿姓名流行度排名) 使用编程练习题 12.31 中的数据文件编写一个程序，使得用户可以选择一个年份、性别，输入一个姓名，然后显示在选择的年份和性别条件下，该姓名的排名，如图 21-9 所示。为了获得最好的效率，为男孩名字和女孩名字分别创建两个数组。每个数组针对 10 个年份具有 10 个元素。每个元素是一个映射表，以值对的方式存储了姓名和相应的排名，并将姓名作为键。假设数据文件保存在 [www.cs.armstrong.edu/liang/data/babynamesranking2001.txt](http://www.cs.armstrong.edu/liang/data/babynamesranking2001.txt), ..., [www.cs.armstrong.edu/liang/data/babynamesranking2010.txt](http://www.cs.armstrong.edu/liang/data/babynamesranking2010.txt) 中。

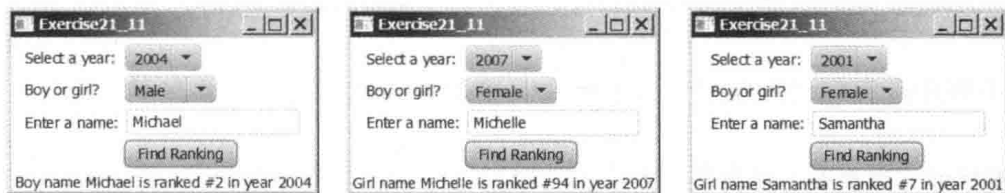


图 21-9 用户选择一个年份和性别，输入年份，单击 Find Ranking 按钮显示排名

- \*\*21.12** (可以同时用于两个性别的姓名) 编写一个程序, 提示用户输入编程练习题 12.31 中描述的文件名, 然后显示文件中可以同时用于两种性别的姓名。使用集合存储姓名并找到两个集合中的共同姓名。下面是一个运行示例:

```
Enter a file name for baby name ranking: babynamesranking2001.txt 
69 names used for both genders
They are Tyler Ryan Christian ...
```

- \*\*21.13** (婴儿姓名流行度排名) 修改编程练习题 21.11, 提示用户输入年份、性别和姓名, 然后显示该名字的排名。提示用户输入另一个查询或者退出程序。下面是一个运行示例:

```
Enter the year: 2010 
Enter the gender: M 
Enter the name: Javier 
Boy name Javier is ranked #190 in year 2010
Enter another inquiry? Y 
Enter the year: 2001 
Enter the gender: F 
Enter the name: Emily 
Girl name Emily is ranked #1 in year 2001
Enter another inquiry? N 
```

- \*\*21.14** (Web 爬虫) 重写编程练习题 12.18, 为 `ListOfPendingURLs` 和 `listofTraversedURLs` 采用合适的新的数据结构以提高性能。
- \*\*21.15** (加法测试题) 重写编程练习题 11.16, 将答案保存在一个集合中, 而不是线性表中。