



国外经典教材·计算机科学与技术

PEARSON
Prentice
Hall

Java for Engineers and Scientists, Second Edition

Java程序设计

——面向工程和科学人员(第2版)

(美) Stephen J. Chapman 著

敖富江 译



清华大学出版社

国外经典教材·计算机科学与技术

Java 程序设计

——面向工程和科学人员

(第2版)

(美) Stephen J. Chapman 著

敖富江 译

清华大学出版社

北 京

Simplified Chinese edition copyright © 2005 by **PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.**

Original English language title from Proprietor's edition of the Work.

Original English language title: **Java for Engineers and Scientists, Second Edition** by Stephen J. Chapman, Copyright © 2004

EISBN: 0-13-033520-7

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由培生教育出版集团授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2004-2819

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有 **Pearson Education** (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

Java 程序设计——面向工程和科学人员(第2版)/(美) 查普曼(Chapman, S.J.)著; 敖富江译.

—北京: 清华大学出版社, 2005.5

书名原文: **Java for Engineers and Scientists, Second Edition**

(国外经典教材·计算机科学与技术)

ISBN 7-302-09941-3

I. J… II. ①查…②敖… III. JAVA 语言—程序设计—教材—英文 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 120382 号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

组稿编辑: 曹 康

封面设计: 久久度文化

印 装 者: 北京鑫霸印务有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 40.25 字数: 1030 千字

版 次: 2005 年 5 月第 1 版 2005 年 5 月第 1 次印刷

书 号: ISBN 7-302-09941-3/TP·6831

印 数: 1~4000

定 价: 78.00 元

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

文稿编辑: 李 阳

版式设计: 康 博

出版说明

近年来,我国的高等教育特别是计算机学科教育,进行了一系列大的调整 and 改革,急需一批门类齐全、具有国际先进水平的计算机经典教材,以适应当前我国计算机科学的教學需要。通过使用国外先进的经典教材,可以了解并吸收国际先进的教学思想和教学方法,使我国的计算机科学教育能够跟上国际计算机教育发展的步伐,从而培育出更多具有国际水准的计算机专业人才,增强我国计算机产业的核心竞争力。为此,我们从国外知名的出版集团 Pearson 引进这套“国外经典教材·计算机科学与技术”教材。

作为全球最大的图书出版机构,Pearson 在高等教育领域有着不凡的表现,其下属的 Prentice Hall 和 Addison Wesley 出版社是全球计算机高等教育的龙头出版机构。清华大学出版社与 Pearson 出版集团长期保持着紧密友好的合作关系,这次引进的“国外经典教材·计算机科学与技术”教材大部分出自 Prentice Hall 和 Addison Wesley 两家出版社。为了组织该套教材的出版,我们在国内聘请了一批知名的专家和教授,成立了一个专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动,各位委员根据国内外高等院校计算机科学及相关专业的现有课程体系,并结合各个专业的培养方向,从 Pearson 出版的计算机系列教材中精心挑选针对性强的题材,以保证该套教材的优秀性和领先性,避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量,我们为这套教材配备了一批经验丰富的编辑、排版、校对人员,制定了更加严格的出版流程。本套教材的译者,全部来自于对应专业的高校教师或拥有相关经验的 IT 专家。每本教材的责编在翻译伊始,就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华,在经过翻译、排版和传统的三审三校之后,我们还请编审委员或相关的专家教授对文稿进行审读,以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限,该套教材在出版过程中很可能还存在一些遗憾,欢迎广大师生来电来信批评指正。同时,也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材,共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

国外经典教材·计算机科学与技术

编审委员会

主任委员:

孙家广 清华大学教授

副主任委员:

周立柱 清华大学教授

委员(按姓氏笔画排序):

王成山	天津大学教授
王 珊	中国人民大学教授
冯少荣	厦门大学教授
冯全源	西南交通大学教授
刘乐善	华中科技大学教授
刘腾红	中南财经政法大学教授
吉根林	南京师范大学教授
孙吉贵	吉林大学教授
阮秋琦	北京交通大学教授
何 晨	上海交通大学教授
吴百锋	复旦大学教授
李 彤	云南大学教授
杨宗源	华东师范大学教授
沈钧毅	西安交通大学教授
邵志清	华东理工大学教授
陈 纯	浙江大学教授
陈 钟	北京大学教授
陈道蓄	南京大学教授
周伯生	北京航空航天大学教授
孟祥旭	山东大学教授
姚淑珍	北京航空航天大学教授
徐佩霞	中国科学技术大学教授
徐晓飞	哈尔滨工业大学教授
秦小麟	南京航空航天大学教授
钱培德	苏州大学教授
曹元大	北京理工大学教授
龚声蓉	苏州大学教授
谢希仁	中国人民解放军理工大学教授

前 言

本书的目标是向工程和科学人员提供 Java 编程语言、结构化编程技术等方面的知识，并提供一些良好的编程习惯。

Java 是一门相对较新的编程语言，自出现后就迅速地占领了编程界。它具有多个优点。其中，一个主要的优点是它提供了几乎完全的平台无关性。这样，为某一台计算机编写的应用程序很可能不需要作任何改动，就可以在另一台计算机上正确运行。人们能够编写在公司的所有计算机上都能运行的简单应用程序，无论这些计算机是 PC、Mac 计算机，还是 Unix 工作站。第二个主要的优点是 Java 的语法规则与 C 语言的相类似，但它摒弃了 C 语言的很多非常模糊和凌乱的特征。类似于 C 语言的语法规则使得它对于众多程序员来说，似曾相识，因此更容易接受。Java 相对于 C 来说是一种提高，如 Java 将字符串作为对象来处理，并使用一组标准的方法集操作该对象。而在 C 语言中，则使用指针来操作字符串，这种方式非常容易出错。

Java 的第三个主要优点是它的面向对象性，这使得使用 Java 编写的代码可以在多个应用程序间更具可重用性。为一个应用程序编写的类和方法可以不加修改地应用于另一个应用程序中，因为对象中数据和方法的封装方式阻止了所不期望的应用程序间的交互。

Java 的第四个优点是它具有设备无关性的图形应用。而 C 和 Fortran 等编程语言不提供设备无关性的图形应用，因此程序员必须关心用于显示图形的硬件的特定细节。C 等编程语言的定义不包含“在较高层次处理图形的标准 API 函数”。相反，Java 的 Swing 图形类则提供了一种较高层次的抽象，在所有的 Java 应用中，该抽象都是相似的，这实现了设备无关性的图形应用。

Java 的最后一个优点在于它是免费的。Sun 公司提供了一个免费的 Java 软件开发包，可以通过它的 Web 站点(<http://java.sun.com>)下载。这个开发包包含免费的 Java 编译器、开发工具和类库。厂家(如 Borland、IBM 和 Sun 等)都免费提供了他们的 Java 集成开发环境的个人版本，包括一些图形编排工具和一些优秀的调试器。对于降低开发成本来说，Java 是最好的选择。在 20 世纪 70 年代和 80 年代，Unix 操作系统和 C 语言之所以能够快速达到其顶峰，是因为它们对众多大学是完全免费的，那一

代的学生们都使用这两种语言，继而在工作后广泛地传播、使用它们。Java 的发展过程有点类似，但它的传播速度更快。

另外，利用 Java 很容易编写小应用程序，这些小应用程序可以自由地通过 Web 进行交换和运行。在当今的互联世界里，这具有极大的吸引力。

由于 Java 是一门较新的编程语言，因此它的一个不足在于没有足够的用于解决科学和工程问题的可重用类库。当然，随着越来越多的大学采用 Java，这样的类库将在未来的几年中出现。Java 的第二个不足是它的 I/O 系统非常复杂、混乱。本书的前几章将仅使用 Java I/O 系统的一个小子集来避免这个问题，在第 16 章中再详细介绍 Java 的 I/O 系统。

Java 的另外一个不足在于它的非常复杂的应用编程接口(API)。但这是 Java 的灵活性和功能强大性的另一个侧面反映。标准的 API 能够使得程序员以平台无关性的方式完成很多工作，但学习一个复杂的 API 会花费较长的时间。有些教科书通常向初学者隐藏 Java 的复杂性，只是在开始的章节中使用一些用户自定义的简单类，直到学习者具有足够的编程经验，才能够彻底理解 Java 的含义。很多导师和学生不认同这种方式，本书则满足了他们的需求。除了第 5 章中介绍的用于显示工程数据的绘图类外，本书中使用的所有类都是标准 Java SDK 1.2 版或更新版的一部分。但使用这些类只是临时性的，因为在第 12 章中介绍了使用标准 Java SDK 编写绘图类的方法。

本书概述

本书集中了作者在国防和地理学领域编写、维护大型程序的经验。编写大型、可维护程序的策略和技术与初级工程人员在学校中学习的编程方式完全不同。维护和修改大型程序令人难以置信的巨额开销，要求这些程序的可读性强、可维护性高。Java 编程语言满足了这个需求。它的平台无关性使得在计算机间移植程序非常容易，其简单的(相对于 C 语言)语法规则和严格的面向对象方式满足了编写整洁程序的要求。本书将讲授 Java 语言的基础知识和一种非常良好的编程风格，采用该风格可以编写出可维护性更高的程序。

让学生们认识到在程序设计过程的早期阶段就投入大量的精力，以使得程序具有更好的可维护性，是非常困难的。类的划分必须非常简单、自然，这样就能够在

较短的时间完成类的编写，并且在未来很长一段时间内不需要对其进行维护。由于多数项目非常简单，通常，学生们能够即时地编写出程序代码。学生们通过一个课时的学习就能够演示所有的程序语句，通过所有的测试，但这样无法养成开发大型项目所需的良好编程习惯。

本书从一开始就以良好的编程风格讲授 Java 编程，该风格适用于大型项目。本书强调在编写程序代码前履行一个详细设计过程的重要性，并强调使用自顶向下的设计技术，将程序划分为多个逻辑部分(类和方法)，可以独立地实现这些部分^①。本书通过在后面介绍的示例中嵌入前面示例中的类和方法来演示对象的可重用性。最后，本书强调了在程序发布之前，使用多个不同的输入数据集完整地测试整个程序的重要性。

本书尝试以交互的方式提供学习内容，在本书的 Web 站点上提供了书中所有示例的源代码，并鼓励您下载、运行和修改这些源代码。某些章节后面的练习是以这些示例的内容为基础的。该站点中还包含了在某些练习中用到的绘图包。

本书也面向大多数工程人员，提供入门性的编程课程结构。通常这种课程是“Introduction to Engineering Problem Solving”课程范围内的一个模板，而学习具体编程语言的时间是非常有限的。这种课程通常讲授简单的过程性编程，采用的编程语言是 Fortran、C、Basic 或 Pascal 等，而较高级的内容则在另外一门独立的课程中介绍。本书的第 2-6 章介绍了过程化编程，可以作为这种入门性课程的基础。您可以通过一门编程语言学习结构化的编程技术，例如 Pascal 语言，并且这种技术相对于语言本身来说更具有实用价值。另外，在第 5 章中初学者可以学会如何编程实现绘图，而在标准的 Fortran、C、Basic 或 Pascal 等语言中这是无法实现的。

本书内容安排

翻一翻其他的 Java 书籍，就可以发现大多数都超过 1000 页，其中大部分篇幅都在介绍如何使用 Java API 类。本书则采取一种不同的方式，仅仅考虑 Java API 的一个用于科学计算和显示结果的较小子集。例如，主要介绍了具有现代特征的 Java2D 图形和 Swing 图形用户界面，完全忽略了过去显示数据的方式。这样做能够

① 本书在强调将程序划分为类和方法的重要性时，未引入任何面向对象的设计方法。

将更多的篇幅集中于编程技术和技术问题求解上，而书的篇幅却只有其他同类书籍的一半。

第 1 章介绍了 Java 的一些应用，以演示本书中介绍的所有基本规则。这些应用非常适合于介绍基本规则，因为它们非常简单，并且不掩盖示例所演示的知识点。本书逻辑性较强，介绍了编程语言基础(第 2-4 章)、数组(第 5 章)、方法(第 6 章)、类(第 7 章)、字符串(第 8 章)，以及面向对象编程的特征。第 12-14 章中介绍了数据的图形显示和设备无关性显示，第 15 章简短地解释了小应用程序，本书在第 16 章总结了 Java I/O 系统方面的内容。

本书特色

本书主要强调如何以适当的方式编写出可靠的 Java 程序。这些特色对于学习 Java 的人员来说很有用。主要包括：

1. 强调问题求解

本书从开始就开发并运行了一个具有实际意义的示例，这个示例解决了特定工程环境中的问题。本书强调解决语言中遇到的问题，仅仅引入了能够使程序正常执行所需要的最少的 Java 类。本书首先介绍了一个能够解决特殊问题的 Java 程序，而不是设计一个在 Web 浏览器中执行的小应用程序。在后续的章节中陆续地引入了数组、字符串、绘图，以及详细的类库等知识。

2. 强调交互式学习

本书中所有的示例都可以通过本书的 Web 站点下载。我们鼓励您下载并运行这些示例。另外，每章后面的练习需要您修改并增强代码的功能。我们鼓励您在程序中重用这些组件。

3. 强调强类型化和数据字典

Java 类和方法中的每一个变量和引用都必须经过精确定义，因此强类型化是该语言的一个内在特征。本书强调在每个方法中精确地声明每一个变量和引用，并且强调制作数据字典的重要性。数据字典描述了每个变量的特定目的，利用它可以使

得代码清晰、易理解。

4. 强调自顶向下的设计方法

第3章中引入了自顶向下的设计方法，本书自始至终都使用了该设计方法。这种方法要求您在编写代码前首先应当思考适当的程序设计方法。该方法强调在开始工作之前，要清晰地定义所需要解决的问题，以及所需要的输入输出。一旦问题被适当地定义后，就可以将它划分为一些独立的类，以表示程序中具体的对象，并使用方法来表示这些对象的行为。最后，本书强调测试整个过程的所有阶段的重要性，包括分别测试所有的类和方法，以及测试最终产品。其中，给出了一些示例，这些示例在某些数据集中能够正确地工作，而在其他数据集中却无法正常工作。

在逐步细化的过程中，引入了伪代码和流程图，并且几乎在所有的示例中都使用了伪代码。

本书中介绍的设计过程可以概述如下：

- (1) 确定用户需求
- (2) 分析并分解问题
- (3) 详细设计
- (4) 实现：将算法转换为具体的 Java 语句
- (5) 测试

5. 强调 Java 类库

Java 的一个巨大优点是它包含了大量的类库(程序包)。在编写 Java 程序时，可以使用这些类库实现很多功能。当编译器厂商已经提供了实现某个功能的类和方法后，您再去重新编写代码实现该功能是没有任何意义的。Java 中对象的可重用属性使得编程更加快捷、简单、可靠。本书强调 Java 语言所固有的可重用性优点。

6. “良好的编程习惯”框

为了便利您阅读本书，书中以突出显示的方式给出了一些良好的编程习惯。另外，在每一章的结尾处重新概括了该章中介绍的所有编程习惯。一个“良好的编程习惯”框的样式如下所示：

良好的编程习惯

对于每一个结构体，都缩进两到三个空格，以提高程序代码的可读性。

7. “编程时易犯的错误”框

这种框强调了一些易出现的错误。一个“编程时易犯的错误”框的示例如下：

编程时易犯的错误

在 `while` 语句后面添加一个分号，将产生逻辑错误。Java 将能够编译并执行这个程序，但该程序可能会出现死循环。

教学特色

本书包含一些非常有用的教学特色，这些特色能够增强您的理解能力。每一章提供了一到两个小测试，并且在附录 C 中提供了这些测试的答案。这些小测试可以用来测试您的理解能力。另外本书还提供了大约 250 个章节练习题，本书的 Web 站点中提供了这些练习题的答案。在所有章节中，使用突出显示的“良好的编程习惯”框给出了一些良好的编程习惯，并使用“编程时易犯的错误”框给出了易出现的错误。每一章的结尾处提供了该章的小结和良好的编程习惯的概述。最后，通过本书的 Web 站点还可以下载书中所使用的所有特定类。

设置 CLASSPATH 环境变量

为了使用特定的绘图类 `JPlot2D`，您必须首先设置 `CLASSPATH` 环境变量，这样，Java 编译器才能够找到该程序包。这个变量必须包含类路径结构的父目录。例如，如果程序包出现在目录“`C:\packages`”下的子目录中，则 `CLASSPATH` 变量必须包含目录“`C:\packages`”。

在不同的操作系统中，环境变量的设置方式是不同的。在 Windows NT 4.0/2000/XP 中，环境变量的设置是通过“控制面板”中的“系统”选项来完成的。详情请参阅 Windows 的帮助系统。

在 Windows 95/98 中，可以在 `autoexec.bat` 文件中设置 `CLASSPATH` 环境变量，

设置该变量的命令如下所示：

```
set CLASSPATH=.;c:\packages
```

如果该文件中已经存在了一个 CLASSPATH，则可以在该路径后面添加一个分号(;), 并在分号后面添上 “c:\packages”。

对于运行 C 命令解释程序的 Linux 和 Unix 操作系统，可以使用文本编辑器打开 .login 文件，并添加如下命令行来设置类路径：

```
setenv CLASSPATH.:$HOME/ packages
```

如果 .login 文件中已经存在 CLASSPATH，则在其后添加一个冒号(:)，并紧跟着添加 “\$HOME/ packages”。

对于运行 Bourne 或 Korn 命令解释程序的 Linux 和 Unix 操作系统，类路径的设置是通过使用文本编辑器打开 .profile 文件，并添加如下命令行来实现的：

```
CLASSPATH=.:$HOME/ packages  
export CLASSPATH
```

如果 CLASSPATH 已经存在于该文件中，则在存在的路径后添加一个冒号(:)，并紧跟着添加 “\$HOME/ packages”。

敬告读者

无论我是多么仔细地校对本书中的每一个文档，都会出现少数印刷错误。如果您发现了此类错误，请您通过出版商通知我，我将在后续的印刷和版本中更正这些错误。非常感谢您提出此类问题。

根据软件许可协议，您可以出于非商业目的免费地使用并修改本书中的类。但对于本书我有一个特别的要求，如果您发现并修改了 bug，或是增强了某些类的功能，请您复制一份给我。我的 Email 地址可以通过本书的 Web 站点获得。

我将在本书的 http://www.prenhall.com/author_chapman/ 站点中维护一个勘误表。请查看这个站点，以便及时更新、纠正错误。

Stephen J. Chapman
Melbourne, Australia

目 录

第 1 章 计算机和 Java 编程语言	1
1.1 计算机概述	1
1.1.1 CPU	2
1.1.2 主存储器和辅助存储器	2
1.1.3 输入输出设备	2
1.2 计算机中的数据表示	3
1.2.1 二进制数值系统	3
1.2.2 二进制数值的八进制和十六进制表示	4
1.2.3 存储器中的数据类型	5
1.3 计算机语言	8
1.4 Java 编程语言	9
1.5 Java 的基本元素	10
1.6 对象、方法和类	11
1.7 小应用程序和应用程序	11
1.8 第一个 Java 应用程序	12
1.9 编译和执行 Java 程序	14
1.10 小结	15
1.11 练习	16
第 2 章 Java 的基本元素	17
2.1 Java 的命名方法	17
2.2 常量和变量	18
2.2.1 整型常量和变量	20
2.2.2 浮点常量和变量	21
2.2.3 布尔常量和变量	22
2.2.4 字符常量和变量	22
2.2.5 保持常量的一致性	23

2.3	字符串	24
2.4	赋值语句和算术运算	26
2.4.1	整数运算	26
2.4.2	浮点运算	27
2.4.3	运算层次	27
2.4.4	操作数的数值提升	29
2.4.5	赋值转换和强制转换	31
2.5	赋值运算符	33
2.6	递增和递减运算符	34
2.7	数学方法	35
2.8	标准输入和输出	37
2.8.1	使用标准输出流	38
2.8.2	使用标准输入流读入字符串	39
2.8.3	使用标准输入流读入数值和布尔值	41
2.9	问题示例	45
2.10	调试 Java 程序	52
2.11	小结	53
2.12	良好的编程习惯概述	54
2.13	练习	54
第 3 章	分支结构和程序设计	59
3.1	程序设计技术概述	59
3.2	使用伪代码和流程图	64
3.3	关系运算符与逻辑运算符	66
3.3.1	关系运算符	66
3.3.2	逻辑运算符	68
3.4	选择结构	73
3.4.1	if 结构	73
3.4.2	if/else 结构	75
3.4.3	在一个 if 结构中执行多条语句	77
3.4.4	使用 if 结构的示例	78

3.4.5	if 结构中的相等测试	84
3.4.6	嵌套 if 结构	86
3.4.7	条件运算符	88
3.4.8	switch 结构	88
3.5	GUI 输入和输出	92
3.6	小结	97
3.7	良好的编程习惯概述	97
3.8	练习	98
第 4 章	循环结构	101
4.1	while 循环	101
4.1.1	控制 while 循环	102
4.1.2	使用 while 循环的示例	104
4.2	do/while 循环	107
4.3	for 循环	109
4.3.1	for 循环的要求	117
4.3.2	continue 语句和 break 语句	120
4.3.3	嵌套循环	121
4.3.4	带标号的 break 语句和 continue 语句	122
4.4	格式化输出数据	126
4.5	问题示例	131
4.6	Java 程序调试知识	136
4.7	小结	139
4.8	良好的编程习惯概述	139
4.9	练习	139
第 5 章	一维数组、文件访问和绘图	146
5.1	数组概述	146
5.2	声明数组	147
5.3	在 Java 语句中使用数组元素	149
5.3.1	初始化数组	150
5.3.2	数组下标的越界	151

5.3.3 数组声明中命名常量的使用	152
5.4 数组引用赋值和数据泄漏	157
5.5 数组中数据的读入和写出	161
5.5.1 读文件	162
5.5.2 写文件	164
5.6 问题示例	165
5.7 绘图	175
5.8 小结	179
5.9 良好的编程习惯概述	179
5.10 练习	179
第 6 章 方法	184
6.1 使用方法的原因	185
6.2 方法的定义	186
6.3 Java 中的变量传递: 按值传递策略	188
6.4 问题示例	192
6.5 变量的生命期和作用域	195
6.6 递归方法	196
6.7 参数的强制转换	199
6.8 方法的重载	200
6.9 使用 <code>java.util.Arrays</code> 类进行数据排序	208
6.10 小结	209
6.11 良好的编程习惯概述	210
6.12 练习	210
第 7 章 类和面向对象编程	217
7.1 面向对象编程概述	217
7.1.1 对象	217
7.1.2 消息	218
7.1.3 类	219
7.1.4 类变量和类方法	220
7.1.5 类的层次结构和类的继承	221

7.1.6 面向对象编程	222
7.2 类的结构	222
7.3 第一个示例: Timer 类	223
7.3.1 Timer 类的实现	224
7.3.2 Timer 类的使用方法	225
7.3.3 Timer 类的注释	227
7.4 作用域	227
7.5 this 引用	229
7.6 方法的分类	230
7.7 成员访问修饰符	236
7.8 标准 Java 包	238
7.9 创建自己的包	240
7.9.1 设置类路径	241
7.9.2 用户定义包的使用方法	241
7.10 终结函数和垃圾回收	242
7.11 使用静态类成员	243
7.11.1 静态变量	243
7.11.2 静态方法	244
7.12 小结	251
7.13 良好的编程习惯概述	252
7.14 练习	252
第 8 章 字符串	258
8.1 创建并实例化字符串	258
8.2 String 类的方法	259
8.2.1 子字符串	260
8.2.2 字符串的连接	262
8.2.3 字符串的比较	263
8.2.4 在字符串中定位字符和子字符串	268
8.2.5 String 类的其他方法	269
8.2.6 valueOf()方法	270