

特辑

[美] Joseph L. Weber 著

卜照斌 段来盛 赖 伶 等译



Special Edition
Using Java 2

Java 2

编程详解

que®



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

附带光盘包括JBuilder2、
正式版等!

Java 2 编程详解

Special Edition Using Java 2

[美] Joseph L. Weber 著

卜照斌 段来盛 赖 伶 等译



964781

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

有光

内 容 简 介

Java 是面向对象的、支持多线程的、体系结构独立的解释型动态语言,具有高度的安全性、可移植性和代码可重用性,具有很高的运行效率,是目前软件中极为健壮的编程语言。Java 2 是 SUN 公司 1998 年 12 月发行的 Java 版本,它对前面的版本进行了完善和提高,增加了基础类、二维图形接口类、媒体框架类和商业及 Java Wallet 等内容。本书较全面地介绍了 Java 2 的内容,并就部分内容与 Java 1.0 和 1.1 版本作了对照讲解。本书是综合性的工具书,包含了 Java 编程语言的各方面的信息。如果读者不熟悉 Java,使用这本书就很合适;如果读者已是 Java 专家,那么本书也将成为您的编程宝典。

Authorized translation from the English language edition published by QUE-CONSUMER, an imprint of Macmillan Computer Publishing U.S.A.

Copyright © 12/08/97

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

SIMPLIFIED CHINESE language edition published by Publishing House of Electronics Industry, China.

Copyright © 1999

本书中文简体专有翻译出版权由美国 Macmillan Computer Publishing 下属的 QUE-CONSUMER 授予电子工业出版社。该专有出版权受法律保护。

图书在版编目(CIP)数据

Java 2 编程详解/(美)韦伯(Weber, J.L.)著;卜熙斌译. —北京:电子工业出版社,1999.6
书名原文:Special Edition Using Java 2
ISBN 7-5053-5323-3

I. J… II. ①韦… ②卜… III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 25275 号

书 名: Java 2 编程详解
著 者:[美]Joseph L. Weber
译 者:卜熙斌 段来盛 赖 伶 等
责任编辑:董 娅
排版制作:华燕实业公司
印 刷 者:北京天竺新华印刷厂
装 订 者:三河市金马印装公司
出版发行:电子工业出版社 URL: <http://www.phei.com.cn>
北京市海淀区万寿路 173 信箱 邮编 100036
经 销:各地新华书店
开 本:787×1092 1/16 印张:78.5 字数:2009.6 千字 附光盘:1 张
版 次:1999 年 8 月第 1 版 1999 年 8 月第 1 次印刷
书 号: ISBN 7-5053-5323-3
TP·2650
印 数:2500 册 定价:150.00 元(含光盘)
版权贸易合同登记号 图字:01-98-0148

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。若书店售缺,请与本社发行部联系调换。电话 68279077

译 者 序

Internet 是本世纪末人类科技史上的重要里程碑,它在人类生活中正在起着越来越重要的作用,它对软件开发的的要求是业界人士无法忽略的重要因素,Java 正是针对 Internet 应用开发的编程语言。在不久的将来 Java 将作为计算机操作系统的组成部分。

Java 是面向对象的、支持多线程的、体系结构独立的解释型动态语言,具有高度的安全性、可移植性和代码可重用性,具有很高的运行效率,是目前软件中极为健壮的编程语言。

Java 2 是 SUN 公司 1998 年 12 月发行的 Java 版本,它是对前面版本的完善和提高,它新增加了基础类、二维图形接口类、媒体框架类和商业及 Java Wallet 类等内容。本书较全面地介绍了 Java 2 的内容,并就部分内容与 Java 1.0 和 1.1 版本中的相应部分作了对照讲解。本书是综合性的工具书,是计算机 Internet 应用开发人员的备用书,也可作为计算机应用人员的参考书。

本书的前言由李璟翻译,第 1 章到第 25 章由卜照斌翻译,第 26 章到第 27 章由赖伶翻译,第 28 章到第 30 章由陶文生翻译,第 31 章到第 32 章由杜建斌翻译,第 33 章到第 35 章及附录由寇国华翻译,第 36 章到第 56 章由段来盛翻译,蔡剑枫、李璟和张红对本书中的部分例子作了测试工作。另外,张宾、刘理、余国齐和刘汪洋完成了文字输入和编排整理工作。

由于时间和精力有限,难以做到尽善尽美,不当之处在所难免。恳请读者批评指正。

译 者

一九九九年一月

鸣 谢

感谢所有为本书做出贡献的人。像这样一部书不是靠一个人的努力能完成的,他是所有撰稿人、编辑和开发者共同努力的结果。有许多人为本书做出了自己的贡献,遗憾的是无法知道所有这些人的姓名,但我仍要衷心地感谢他们所做出的努力。

特别感谢 **Ben Milstead**,他开发了这项工作,他与 **Joe Carpenter** 和 **Jodi Cornelius** 一起帮助我们保证本书内容的正确性。具有高度的精确性,这对于本书来说是最重要的。

感谢 **Jeff Taylor**,他是本书最后一位责任编辑,通过他的通盘调校使得本书更加流畅易读。

不能不提到 **Mark Wutka** 为本书做出的努力。他帮助撰写了本书的一些部分,从第一版开始他就用卓越的才能和勤奋的工作帮助我们增删本书。**Mark Wutka** 应该为他对本书的贡献而当之无愧地得到我们无尽的感激和衷心的感谢。还要感谢:

Stephanie Gould 我的第一位责任编辑,是他说服我编著了我的处女作。

Mark Arend 和 **Gene deGroot** 当我已不再年轻时,给了我毅力和创造力。

Shawn Gubine 我交往最早和最亲密的朋友,在这里没有理由不介绍他。

Scott Morris 他告诉了我为什么大多数软件按照自己的方式运行,是“因为它们不问我”,并让我知道信心是成功的百分之九十九。

David 和 **Dorothy** 我的兄弟,他们给了我很大支持。

最后,我要感谢我的妻子,**Kim** 应该得到和我制作本书所得到的一样的声誉。

告诉我们您的想法

作为本书的读者,您是最重要的批评家和评论员。我们尊重您的意见,我们希望知道我们什么做得对、什么可以做得更好、您在什么地方看到的本书以及您想要我们知道的。消息。

作为 **Macmillan Computer Publishing** 出版社的执行编辑,欢迎您对我们的作品发表评论,您可以发电传、电子邮件或直接写信给我,让我知道在本书中您喜欢什么和不喜欢什么以及我们该如何做才能使本书更加优秀。

请注意:我无能力帮助您解答与本书有关的技术问题,另外由于我收到的邮件很多,所以我无法回复所有的信件。如果写信给我们,请注明书名和作者以及您的姓名和电话或者传真号码。我将仔细考虑您的评论并参与与本书工作的作者和编辑讨论您的建议。

电传:317-817-7070

电子邮件地址:java@mcp.com

邮件地址:Tim Ryan, Executive Editor

Java Team

Macmillan Computer Publishing

201 West 103rd Street

Indianapolis, IN 46290 USA

前 言

欢迎您来到令人惊奇和充满动感的 Java 世界！如果您从未接触过 Java,那么在这里可以享受到难得的乐趣。Java 是一种非常丰富且简单易学的语言,Java 为程序员提供了前所未有的编程方法以解决人们看来十分复杂的任务。

什么是 Java? Java 是 1995 年 6 月由 Sun Microsystems 引进我们这个世界具有革命性的编程语言。从那时起成千上万的程序员捧起 Java 书并逐渐认识到 Java 是功能如此强大的语言。

Java 是面向对象的编程语言,这就意味着人们可以用 Java 十分容易地开发复杂的程序。另外 Java 还内置有线程支持、网络支持和大量的其它工具。

本书的使用对象

如果读者不熟悉 Java,那么使用这本书就很合适。不要被本书的厚度所吓倒,它包含了 Java 编程语言的各方面信息,本书的设计风格可以使读者很容易起步和继续学下去。

如果读者已经是 Java 专家,那么本书将成为您的编程宝典。实际上它可以始终不离案头。本书作为信息提供者包含了最完整的参考和 Java 编程语言的各方面的示例集合。目前可用的 API 在本书中都有所探究,所有的编程方法都有文档。在本书的封面与封底之间,读者可以找到大量的例子和解释,它们可以使您作为程序员的生活变得格外舒适。

本书是如何组织的

本书由 11 个部分组成。每个部分都包含有关于 Java 语言是如何组成的信息。

第一部分“Java 概述”介绍了 Java 语言设计和虚拟机,介绍了 Java 能够做什么和在当今某些程序中如何实现。明确指示了如何通过下载和安装 Java 开发工具(JDK)启动 Java。最后还介绍了包括 Java 开发工具(JDK)在内的一些免费工具的使用。

第二部分“Java 语言”介绍了 Java 的语法。Java 的语言规则是建立在语言句法规则上的,所有程序都是按照语言规则建立的,这一部分遍历了方法的各个部分。本部分的后半部分介绍了如何开发具体的 Java 程序,包括小应用程序和应用程序。作为初学者可以从每一章中多方面地熟悉 Java 编程。对于 Java 专家来说,这一部分对 Java 语言的各个方面都做了详尽的探究,读者可以将第二部分作为重要的参考资料和学习工具来使用。

第三部分“用户界面”详细介绍了如何使用 Java 建立图形用户界面。介绍了如何使用传统的 AWT 界面,并演示了拥有 Java 基础类的 Java 2 的新特性。另外还探究了如何建立和操作图像以及 Java 2 的二维图形系统。

第四部分“输入输出”介绍了在 Java 应用程序中读取和写入数据。该部分由基本组件开始,顺序地介绍了流技术和文件的读取,然后介绍了如何开发网络应用程序。在这里读者可以得到无价的有关国际互联网的 TCP/IP 协议的信息。在本部分结束时还介绍了一些高级特性,比如,确保数据传输的安全性。该部分还讲述了使用 Java 序列化发送和重新获得 Java 所

有对象和通过远程方法调用来运行远程机上的 Java 程序。最后还讨论了 Java 2 新的管理 API,它可以与更高级的设备对话,比如路由器。

第五部分“数据库”详细介绍了建立当前商业应用程序的最重要部分。数据库几乎是所有商业应用程序的核心,Java 的 JDBC(Java 数据库连接)可以减轻 Java 应用程序和数据库之间通信的负担。在这部分里读者可以了解到数据库是如何工作的,可以了解一点历史和学习一些必要的术语,然后探究一下 Java 的 JDBC 界面。该界面可以连接、发送和存储数据到任何与 JDBC 兼容的数据库中。欢迎读者来到平台无关和 DBMS 无关系统的世界!

第六部分“基于组件的开发”对于想了解如何使开发周期快速轻松的人来说这部分是十分迷人的。基于组件的开发已经有几年的历史了,但是还没有像 Java 做得这么容易。该部分介绍了如何使用三种不同的组件模式:Java 特有的 JavaBean 说明、CORBA(很快将成为工业标准并映射在 JavaBean 中)和 COM(微软公司的与窗口对话的界面)。

第七部分“高级Java”介绍了在准备进入下一步时的一些非常复杂的技术。本部分向读者展示了高级的 Java 技术。在这部分里读者可以学到 Java 服务器端的开发能力和如何使用 Java Wallet 建立高级应用程序。还可以学习到 Java 内置的数据结构和应用程序,以及如何建立本地应用程序。该部分最后介绍了 Java 和 C++ 的比较。

第八部分“调试Java”介绍了所有的程序调试技术。这部分讲述的内容在学习开发应用程序时显得非常重要和有价值。在这里可以找到大量的关于 Sun. tools. debug 软件包的各个方面以及 Java 虚拟机的操作码。

第九部分“JavaScript”讲述的是Java的远亲JavaScript。使用JavaScript可以很容易地完成某些任务,因为它可以控制浏览器,所以可以做一些 Java 无法做的事情。该部分讲述了如何编写 JavaScript 程序,通过对本章内容的学习可以使读者多懂一种语言。

第十部分“Java资源”是一些超出本书介绍范围的附加资源。在这里得到一些超级的国际互联网站点和一些其他资料,他们可以使读者跟上 Java 前进的步伐。

第十一部分“附录”概述了附带的 CD-ROM 中的资源。

本书的使用约定

本书使用了多种文体和印刷惯例以期使本书更易阅读。

注意:表示附加信息用于帮助读者避免发生问题和用于描述特性。

提示:该部分给出了更容易或作为替换的执行过程的方法。在该部分可以看到一些额外的要领和使编程更加容易的方法。

警告:该部分对危险提出警告(例如删除文件)。

在本书中使用交叉参考来帮助读者查看在书中其它部分出现的有关信息。例如:参见第 47 章的“Object 类”。

目 录

第一部分 Java 概述	(1)	3.3.1 从光盘安装在 Windows 95 和 Windows NT 上的 JDK	(33)
第 1 章 Java 能做什么	(3)	3.3.2 从光盘安装在 x86 和 SPARC Solaris 操作系统上的 JDK	(33)
1.1 Java 应用程序的几种类型	(4)	3.3.3 下载 JDK	(34)
1.2 了解 Java 语言	(5)	3.4 安装下载的 JDK	(35)
1.3 Java 开发工具	(5)	3.4.1 在 Solaris x86 平台和 SPARC 平台 上安装	(35)
1.4 Java 小应用程序	(5)	3.4.2 在 Windows 操作系统上安装	(36)
1.5 Java 图形用户界面应用程序	(9)	3.4.3 在 Macintosh 上安装	(37)
1.6 Java 命令行应用程序	(11)	3.5 测试 Java 编译器和 JVM(Java 虚 拟机)	(38)
1.7 Java 是客户/服务器方式的	(11)	3.5.1 创建新的 Java 工程	(38)
1.8 如何跟上形势	(12)	3.5.2 运行 Unix 或 Windows 的 Java 应用程序	(38)
第 2 章 Java 设计	(13)	3.5.3 在 Macintosh 上运行 Java 应 用程序	(39)
2.1 解释 Java	(14)	3.6 在 Windows 3.1 平台上安装 IBM 的小 应用程序开发工具	(39)
2.2 Java 是面向对象的	(15)	第 4 章 JDK 工具	(42)
2.3 Java 虚拟机	(15)	4.1 JDK 工具参考	(43)
2.3.1 Java 源代码	(16)	4.2 AppletViewer(小应用程序浏览器)	(43)
2.3.2 Java 堆栈	(16)	4.3 java,Java 解释器	(44)
2.3.3 Java 寄存器	(16)	4.4 javac,Java 编译器	(45)
2.3.4 无用单元收集堆	(17)	4.5 javap,Java 反编译器	(46)
2.3.5 Java 的方法区	(17)	4.6 Javah C 头文件和承接模块文件的创 建器	(46)
2.4 安全性和 Java 虚拟机	(17)	4.7 javadoc 工具(文档生成器)	(47)
2.4.1 可执行内容与安全性	(17)	4.8 jdb,Java 调试器	(48)
2.4.2 Java 实现安全性的办法	(19)	4.9 CLASSPATH 环境变量	(48)
2.4.3 Java 语言级的安全性	(20)	4.10 Macintosh 问题	(49)
2.4.4 编译过的 Java 代码的安全性	(21)	4.10.1 Macintosh 平台上的小应用程序 浏览器	(49)
2.4.5 Java 系统在运行时刻的安全性	(21)	4.10.2 Java Runner, Macintosh 平台上 的 Java 解释器	(50)
2.4.6 可执行代码的安全性	(22)	4.11 Java 编译器	(51)
2.4.7 开放问题	(24)	4.12 JavaH: C 头文件产生器	(51)
2.5 Java API	(25)	第二部分 Java 语言	(53)
2.5.1 Java 核心 API	(25)		
2.5.2 JavaBean API	(27)		
2.5.3 JDK 1.2 新增加的内容	(28)		
2.5.4 Java 嵌入式 API	(29)		
第 3 章 安装并启动 JDK	(30)		
3.1 为什么需要使用 Sun 公司的 Java 开发 工具来编写 Java 程序	(31)		
3.2 进一步了解 Java 既是编译的又是解 释的	(31)		
3.3 获取和安装 Sun 公司的 JDK	(32)		

第 5 章 面向对象编程	(55)	6.5 API	(75)
5.1 面向对象编程:新的思维方式	(56)	第 7 章 数据类型及其它标记	(76)
5.2 简短的编程历史	(56)	7.1 Java 的两种数据类型	(77)
5.2.1 过程语言	(56)	7.2 了解布尔变量	(78)
5.2.2 结构化程序设计	(57)	7.2.1 声明变量	(78)
5.2.3 面向对象编程	(57)	7.2.2 标识符:变量命名	(78)
5.3 对象	(57)	7.2.3 修改布尔变量	(79)
5.3.1 传统的程序设计	(58)	7.3 各种整数类型	(80)
5.3.2 OOP 方法	(58)	7.3.1 整数类型值的限制	(80)
5.3.3 通过继承扩充对象	(59)	7.3.2 创建整数变量	(81)
5.4 多实体对象	(61)	7.3.3 整数运算	(81)
5.5 组织代码	(61)	7.4 运算符	(81)
5.6 对象和对象如何与 Java 类联系	(61)	7.4.1 算术运算符	(82)
5.7 建立层次:一种面向对象设计的方法	(61)	7.4.2 赋值运算符	(82)
5.7.1 把代码分成最小的实体	(62)	7.4.3 增量/减量运算符	(83)
5.7.2 在实体间寻找共同点	(62)	7.5 字符变量	(83)
5.7.3 在实体之间寻找差异	(62)	7.6 浮点变量	(84)
5.7.4 在实体间寻找最大的共同点	(62)	7.7 数组	(85)
5.7.5 把剩下的共同对象放在一起并 重复	(62)	7.8 空白	(86)
5.7.6 利用对象扩充程序	(63)	7.9 注释	(87)
5.8 Java 并不是完全面向对象的语言	(63)	7.9.1 传统的注释	(87)
第 6 章 Hello World:用户的第一个 Java 程序	(64)	7.9.2 C++ 风格的注释	(87)
6.1 HelloWorld 应用程序	(65)	7.9.3 javadoc 注释	(88)
6.1.1 创建文件	(65)	7.10 源程序中的符号:赋值	(88)
6.1.2 编译代码	(65)	7.10.1 代表整数格式的字符	(89)
6.1.3 运行程序	(65)	7.10.2 代表字符格式的字符	(90)
6.2 理解 HelloWorld	(66)	7.10.3 代表浮点格式的字符	(91)
6.2.1 声明类	(66)	7.10.4 代表字符串格式的字符	(91)
6.2.2 Main 方法	(66)	7.11 创建和撤销对象	(92)
6.2.3 写屏	(67)	7.11.1 使用 new 操作符创建对象	(92)
6.2.4 System.out 和 system.in	(67)	第 8 章 方法	(94)
6.3 作为小应用程序的 HelloWorld 运行 在 Netscape 中	(70)	8.1 方法的两个部分	(95)
6.3.1 新的源代码和编译新的源代码	(70)	8.1.1 声明	(95)
6.3.2 创建 HTML 文件	(71)	8.2 块和语句	(103)
6.3.3 在小应用程序浏览器中运行 程序	(71)	8.2.1 语句加标志	(104)
6.3.4 在 Netscape Navigator 中运行 HelloWorld	(72)	8.2.2 范围	(104)
6.3.5 理解源代码	(72)	8.3 分隔符	(105)
6.3.6 小应用程序简短的生命周期	(73)	第 9 章 使用表达式	(106)
6.4 关键字	(74)	9.1 什么是表达式	(107)
		9.2 如何计算表达式	(107)
		9.2.1 运算符关联	(107)
		9.2.2 Java 运算符的优先顺序	(108)
		9.2.3 小结——运算符表	(109)
		9.2.4 计算顺序	(109)

9.3 Java 表达式与 C 语言表达式的差异	11.6 创建类的实例	(137)
9.4 位运算符	11.7 引用类的部分	(138)
9.5 移位运算符	11.7.1 特殊变量 this	(140)
9.6 类型转换	11.7.2 特殊变量 super	(141)
9.6.1 隐式类型转换	11.8 变量	(142)
9.6.2 强制运算符	11.8.1 变量修饰符	(146)
9.6.3 强制并转换整数	11.8.2 使用方法来提供受保护的访问	(148)
9.6.4 强制并转换字符	11.8.3 使用 finalize() 方法	(149)
9.6.5 强制并转换布尔值	11.9 嵌入类	(151)
9.7 字符串加法	11.9.1 什么是嵌入类	(151)
第 10 章 控制流	11.9.2 使用嵌入类创建程序	(151)
10.1 控制执行	11.9.3 同步嵌入类	(153)
10.2 布尔运算的 true 运算符和 false 运算符	11.9.4 嵌入类是如何工作的	(154)
10.2.1 关系运算符	11.9.5 为什么使用嵌入类	(154)
10.2.2 等于运算符	11.10 包	(154)
10.3 逻辑表达式	11.11 引入包中的类	(155)
10.3.1 条件与运算符和条件或运算符	11.12 引入整个包	(155)
10.3.2 一元逻辑运算符	11.13 不通过引入使用类	(156)
10.4 条件运算符	11.14 使用包来组织代码	(158)
10.5 控制流中的布尔类型	11.15 所有 java.lang 类的隐式引入	(158)
10.6 控制流语句	第 12 章 接口	(159)
10.6.1 if 语句	12.1 什么是接口	(160)
10.6.2 if-else 语句	12.2 创建接口	(160)
10.7 循环语句	12.2.1 声明	(162)
10.7.1 While 语句	12.2.2 接口体	(163)
10.7.2 do 语句	12.3 使用接口	(166)
10.7.3 for 语句	12.3.1 覆盖方法	(166)
10.7.4 switch 语句	12.3.2 修饰符	(166)
10.8 跳转语句	12.3.3 参数列表	(167)
10.8.1 break 语句	12.3.4 体	(168)
10.8.2 continue 语句	12.4 在其它类中使用接口	(168)
10.8.3 return 语句	12.4.1 使用接口的域	(168)
第 11 章 类	12.4.2 把接口当作类型使用	(168)
11.1 什么是类	12.5 异常	(171)
11.2 为什么使用类	第 13 章 线程	(173)
11.3 Java 中的类	13.1 什么是线程	(174)
11.4 声明类	13.2 为什么使用线程	(174)
11.4.1 访问说明符	13.3 如何使类可线程化	(174)
11.4.2 类名	13.3.1 扩展 Thread 类	(174)
11.4.3 超类——扩展另一个类	13.3.2 使用 Runnable 接口	(175)
11.5 构造器	13.4 巧妙的线程竞赛	(175)
11.5.1 覆盖方法	13.5 理解 Great Race	(178)
	13.6 线程处理	(180)
	13.7 测试巧妙的线程竞赛	(181)

13.8 改变优先级	(182)	15.2 在小应用程序中增加图像	(227)
13.9 关于线程的优先级、Netscape 和 Windows	(183)	15.3 在小应用程序中增加声音	(229)
13.10 同步	(185)	15.4 使用小应用程序控制浏览器	(230)
13.11 哲学家进餐问题	(186)	15.4.1 修改状态信息	(231)
13.12 改变线程的运行状态	(186)	15.4.2 修改浏览器显示的页	(232)
13.13 获得正在运行的线程的数量	(188)	15.5 小结	(233)
13.14 查找所有正在运行的线程	(188)	第 16 章 JAR 档案文件	(235)
13.15 超级属性	(190)	16.1 为什么使用 JAR	(236)
第 14 章 编写小应用程序	(192)	16.1.1 绑定	(236)
14.1 Java 的孩子	(193)	16.1.2 压缩	(236)
14.2 小应用程序和 HTML 语言	(193)	16.1.3 向下兼容性	(236)
14.3 在 HTML 页中包含 Java 小应用 程序	(194)	16.1.4 可移植性	(236)
14.3.1 包括替换信息	(195)	16.1.5 安全性	(237)
14.3.2 附加的 <APPLET> 属性	(197)	16.2 什么时候使用 JAR 档案	(237)
14.4 使用 Java 插件	(198)	16.2.1 Jar 工具	(237)
14.4.1 在 Internet Explorer 上使用 Java 插件	(199)	16.2.2 <APPLET> 标志	(239)
14.4.2 在 Netscape 中使用 Java 插件	(199)	16.2.3 兼容的浏览器	(241)
14.4.3 为所有的浏览器设置 HTML 文本	(200)	16.3 JAR 档案和安全性	(241)
14.5 开始开发 Java 小应用程序	(201)	16.3.1 声明文件	(242)
14.5.1 理解 HelloWorld——构造小应用 程序	(203)	16.3.2 私有密钥、公共密钥和证书	(243)
14.6 探究小应用程序的生命周期	(203)	16.3.3 密钥工具	(243)
14.6.1 编译 InitStartStop 小应用程序	(206)	16.3.4 jarsigner	(244)
14.6.2 理解 InitStartStop 小应用程序	(207)	16.4 java.util.zip 包	(246)
14.6.3 Java 动画小应用程序	(209)	16.4.1 类	(247)
14.7 使用控件的小应用程序	(213)	16.4.2 有计划地读取 JAR 文件	(247)
14.7.1 理解 InternetApplet 小应用 程序	(218)	16.5 JAR 文件格式	(249)
14.7.2 探究 init() 方法	(218)	第 17 章 小应用程序和应用程序的 比较	(250)
14.7.3 ActionCommand 和 ActionListener	(219)	17.1 应用程序探究	(251)
14.7.4 actionPerformed() 方法	(219)	17.2 应用程序的优点	(251)
14.7.5 探究 paint() 方法	(220)	17.3 开发 Java 应用程序	(252)
第 15 章 高级小应用程序代码	(222)	17.3.1 HelloWorld 应用程序	(252)
15.1 使用 <PARAM> 标记	(223)	17.3.2 为应用程序传递参数	(253)
15.1.1 理解 starPainter 的源代码	(224)	17.3.3 避免空指针异常	(254)
15.1.2 使用 getSize() 方法	(225)	17.3.4 由 main() 的静态特性带来的 限制	(255)
		17.4 把小应用程序转换为应用程序	(257)
		17.4.1 为什么要把小应用程序转换为 应用程序	(257)
		17.4.2 把小应用程序代码修改成应用程 序代码	(257)
		17.4.3 解释其它小应用程序的特性	(262)
		17.4.4 缺省	(262)
		17.4.5 加入 <param> 信息的第二种	

办法	(264)	19.7.3 使用列表框	(303)
17.4.6 使用窗口来结束应用程序 的工作	(265)	19.8 文本域和文本区	(307)
17.4.7 检查所有的小应用程序的 方法	(269)	19.8.1 创建文本域	(307)
17.5 使用 Zip 文件格式将用户的应用程 序打包	(270)	19.8.2 创建文本区	(308)
17.6 把应用程序转换成小应用程序	(270)	19.8.3 公共文本部件的特性	(308)
17.6.1 最简单的转换	(271)	19.8.4 文本域特性	(309)
17.6.2 处理命令行参数	(271)	19.8.5 文本区特性	(309)
17.6.3 维护应用程序的一个单一实例	(271)	19.8.6 使用文本域和文本区	(310)
17.6.4 在应用程序向小应用程序转换 过程中更难的问题	(272)	19.9 滚动条	(313)
第 18 章 管理应用程序	(273)	19.9.1 创建滚动条	(313)
18.1 安装应用程序	(274)	19.9.2 滚动条特性	(313)
18.1.1 从 .class 文件中安装应用程序	(274)	19.9.3 使用滚动条	(314)
18.1.2 结束安装	(275)	19.10 画布	(315)
18.1.3 结束 UNIX 操作系统下的应用 程序安装	(275)	19.11 部件的公共方法	(317)
18.1.4 结束 Windows 操作系统下的应 用程序安装	(277)	19.11.1 部件显示方法	(317)
18.1.5 从 .jar 文件中安装应用程序	(280)	19.11.2 部件的位置和尺寸设置	(318)
18.2 在同一个系统中维护多个应用程序	(281)	19.11.3 部件布局 and 送显的方法	(320)
第三部分 用户界面	(283)	19.11.4 部件输入事件	(321)
第 19 章 java.awt: 部件	(285)	第 20 章 深入研究异常和事件	(323)
19.1 使用 java.awt 构造图形用户界面	(286)	20.1 Java 的异常	(324)
19.2 按钮	(287)	20.1.1 抛出异常	(326)
19.2.1 创建按钮	(287)	20.1.2 联合方案	(327)
19.2.2 使用按钮	(287)	20.1.3 异常的类型	(328)
19.3 按照 1.1 版本的事件模式使用按钮	(290)	20.1.4 确定要处理的异常	(330)
19.4 标签	(291)	20.1.5 捕捉运行时刻异常	(333)
19.5 复选框和单选按钮	(292)	20.1.6 处理异常	(334)
19.5.1 创建复选框	(292)	20.1.7 创建用户自己的异常类	(336)
19.5.2 检查和设置复选框的状态	(293)	20.2 Java 的 Error 类	(340)
19.5.3 监听复选框中的变化	(293)	20.3 Java 的事件	(340)
19.5.4 创建单选按钮	(295)	20.4 事件处理技术	(341)
19.5.5 使用单选按钮	(296)	20.4.1 事件监听器	(341)
19.6 选择框	(299)	20.5 键盘和鼠标事件	(345)
19.6.1 创建选择框	(299)	20.5.1 键盘事件	(345)
19.6.2 使用选择框	(300)	20.5.2 Java 2 中的修饰键	(348)
19.7 列表框	(301)	20.5.3 Java 2 中的鼠标事件	(349)
19.7.1 创建列表框	(301)	20.6 Java 1.0 的事件模式	(350)
19.7.2 列表框特性	(301)	20.6.1 Event 类	(350)
		20.6.2 事件的起源	(353)
		20.6.3 键盘	(355)
		20.6.4 直接处理事件	(356)
		20.6.5 覆盖 handleEvent() 方法	(356)
		20.6.6 发送用户自己的事件	(358)
		第 21 章 容器和布局管理器	(362)
		21.1 组织部件	(363)

21.2 容器	(363)	22.9 实用图形类	(411)
21.3 布局管理器	(363)	22.9.1 Point 类	(411)
21.4 再谈容器	(364)	22.9.2 Dimension 类	(412)
21.5 容器基础	(364)	22.9.3 Rectangle 类	(412)
21.6 面板	(365)	22.10 Color 类	(414)
21.7 框架	(366)	22.11 裁剪	(416)
21.7.1 创建框架	(366)	22.12 动画技术	(417)
21.7.2 框架特性	(367)	22.13 打印	(419)
21.7.3 使用框架使用的小应用程序 单独运行	(368)	第 23 章 JFC——Java 基础类	(421)
21.7.4 为框架增加菜单	(369)	23.1 Java 基础类	(422)
21.7.5 使用菜单	(370)	23.2 JFC:概览	(424)
21.7.6 弹出式菜单	(373)	23.2.1 设置 JFC	(424)
21.8 对话框	(373)	23.3 Helloworld	(424)
21.8.1 创建对话框	(373)	23.3.1 编译 HelloworldJFC	(425)
21.8.2 可重用的 OK 对话框	(374)	23.3.2 运行中的 HelloworldJFC	(425)
21.9 滚动窗格	(378)	23.3.3 理解 HelloworldJFC	(426)
21.10 再谈布局管理器	(378)	23.3.4 窗格分层	(426)
21.10.1 流式布局	(379)	23.4 改进 Helloworld	(426)
21.10.2 网格布局	(380)	23.4.1 JLabel	(427)
21.10.3 方位布局	(381)	23.4.2 增加图标	(428)
21.10.4 网格块布局	(382)	23.4.3 关闭窗口	(428)
21.11 镶边	(386)	23.5 使用 JFC 增加按钮	(429)
21.12 空布局管理器	(387)	23.5.1 理解 ButtonExample	(430)
21.13 从 Sun 公司的产品进一步扩展	(387)	23.5.2 设置快捷键和助记符	(430)
第 22 章 图形	(388)	23.5.3 监听来自于按钮的动作	(430)
22.1 Java 图形	(389)	23.6 增加工具提示和图标	(431)
22.2 paint 方法、Update 方法和 repaint 方法	(389)	23.7 使用弹出式菜单	(432)
22.3 Graphics 类	(389)	23.7.1 理解 PopupExample	(434)
22.3.1 坐标系	(390)	23.8 边	(435)
22.3.2 画线	(390)	23.8.1 理解 BorderedButtons	(436)
22.3.3 画矩形	(391)	23.8.2 更多的边	(436)
22.3.4 画三维矩形	(391)	23.9 复选框和单选按钮	(437)
22.3.5 画圆角矩形	(393)	23.9.1 理解 CheckBoxPanel	(439)
22.3.6 画圆和椭圆	(394)	23.9.2 使用 ActionListeners 和设置 事件	(439)
22.3.7 画多边形	(395)	23.9.3 使用箱和箱式布局	(439)
22.4 Polygon 类	(396)	23.10 使用 CheckBoxPanel 改变文本对齐 方式	(440)
22.5 画文本	(397)	23.10.1 理解 CheckBoxPanel	(442)
22.5.1 Font 类	(399)	23.10.2 改变对齐方式	(442)
22.5.2 FontMetrics 类	(402)	23.11 标签窗格	(442)
22.6 画模式	(403)	23.11.1 理解 JTabbedPane	(443)
22.7 绘制图像	(406)	23.11.2 其它的 JTabbedPane 能力	(444)
22.8 MediaTracker 类	(408)	23.12 滑杆	(444)

23.12.1 理解 SliderExample	(446)	25.1 在屏幕上绘制图像	(489)
23.12.2 设置刻度标志	(446)	25.2 生产者、消费者和观察者	(492)
23.12.3 捕捉滑杆的变化	(446)	25.3 图像过滤器	(494)
23.13 进度条	(446)	25.4 从内存拷贝图像	(496)
23.13.1 理解 ProgressBarExample	(449)	25.5 将图像拷贝到内存	(498)
23.13.2 创建和控制进度条	(449)	25.6 颜色模式	(506)
23.13.3 ProgressThread	(450)	25.6.1 DirectColorModel 类	(507)
23.13.4 RprogressMonitor	(450)	25.6.2 IndexColorModel 类	(508)
23.13.5 ProgressMonitorInputStream	(453)	25.6.3 RGBImageFilter 类	(508)
23.13.5 ProgressMonitorInputStream	(453)	25.6.4 通过颜色循环产生动画	(513)
第 24 章 高级 JFC	(456)	第 26 章 Java 二维图形	(520)
24.1 模型-视-控制——JFC 设计	(457)	26.1 Graphics2D 对象	(521)
24.1.1 比较 MVC 和 AWT 设计	(457)	26.2 Java 二维应用编程接口中的坐标	(521)
24.1.2 这种模式的部件用在哪里	(458)	26.3 画图	(522)
24.1.3 深入研究 MVC 部件是如何工 作的	(458)	26.3.1 画线	(522)
24.1.4 为什么要理解 MVC	(459)	26.3.2 绘制矩形	(523)
24.1.5 使用 JFC 的可插入的外观-和 -感觉	(460)	26.3.3 绘制圆角矩形	(523)
24.1.6 修改外观-和-感觉	(461)	26.3.4 绘制椭圆和圆	(523)
24.2 菜单和工具条	(462)	26.3.5 画弧	(523)
24.2.1 理解 MenuBar 示例	(465)	26.3.6 画曲线	(524)
24.2.2 加速键	(466)	26.3.7 绘制任意形状	(525)
24.3 列表框和组合框	(467)	26.4 不同的笔划	(525)
24.3.1 理解 ListCombExample	(468)	26.5 自定义填充	(527)
24.3.2 列表框模型	(468)	26.6 转换	(530)
24.4 使用表格	(470)	26.7 绘制文本	(532)
24.4.1 理解 TableExample	(471)	26.7.1 有风格设置的字符串	(532)
24.4.2 表格模式	(471)	26.7.2 文本布局	(533)
24.4.3 单元编辑器	(473)	26.7.3 字符属性	(534)
24.5 树	(475)	26.8 绘制图像	(535)
24.5.1 树结点	(475)	26.8.1 缓冲区中的图像	(535)
24.5.2 树模型	(477)	26.8.2 将图像拷贝到 BufferedImage 中	(536)
24.5.3 理解新的 TreeExample	(480)	26.8.3 过滤缓冲区里的图像	(536)
24.5.4 removeSelectedNode	(480)	26.8.4 操作缓冲区图像	(542)
24.5.5 AddTextFieldValue()	(480)	26.9 透明	(546)
24.6 使用 JEditor 显示 HTML 文件	(481)	26.10 裁剪	(547)
24.6.1 理解 HTMLView	(483)	第四部分 输入输出	(551)
24.6.2 监听超链事件	(483)	第 27 章 文件和流	(553)
24.6.3 设置新页	(483)	27.1 什么是流	(554)
24.6.4 JTextField	(484)	27.2 基本输入输出类	(555)
24.7 创建图标	(484)	27.2.1 InputStream 类	(555)
24.8 JFC 小应用程序	(487)	27.2.2 OutputStream 类	(557)
第 25 章 图像	(488)	27.2.3 System.in 对象和 System.out 对象	(557)

27.2.4 PrintWriter 类	(558)	28.3.4 操作 StringBuffer 对象	(604)
27.3 处理文件	(560)	28.4 使用 StringToKenizer 类	(605)
27.3.1 文件安全	(560)	28.5 字体处理	(608)
27.3.2 FileInputStream 类	(560)	28.5.1 获取字体属性	(608)
27.3.3 使用 FileOutputStream 类	(561)	28.5.2 获取字体度量	(610)
27.3.4 使用 File 类	(563)	28.5.3 创建字体	(611)
27.3.5 RandomAccessFile 类	(567)	28.5.4 使用字体	(611)
27.4 使用管道	(569)	第 29 章 使用多国语言实现	(614)
27.4.1 PipedInputStream 类和 PipedOutputStream 类	(569)	29.1 为什么使用多国语言实现	(615)
27.4.2 PipeApp 应用程序	(570)	29.2 什么是多国语言实现	(615)
27.4.3 探究 main() 方法	(573)	29.3 Java 对多国语言实现的支持	(615)
27.4.4 探究 changeToY() 方法	(575)	29.3.1 Locale 类	(616)
27.4.5 探究 changeToZ() 方法	(576)	29.3.2 将场所敏感数据打包	(618)
27.4.6 探究 YThread 类	(577)	29.3.3 Java.util 包中的其它与多国语 言实现有关的部分	(621)
27.4.7 过滤流	(578)	29.4 多国语言实现的输入和输出(I/O)	(621)
27.5 缓冲区里的流	(578)	29.4.1 字符集转换器	(622)
27.6 数据流	(578)	29.4.2 Reader 和 Writer	(623)
27.6.1 DataInput 接口	(579)	29.5 新的 java.text 包	(624)
27.6.2 DataOutput 接口	(579)	29.6 示例: InternationalTest	(625)
27.6.3 DataInputStream 类和 DataOutputStream 类	(580)	第 30 章 通信和网络	(629)
27.7 字节数组流	(580)	30.1 TCP/IP 概述	(630)
27.8 字符数组流	(581)	30.1.1 OSI 参考模型	(630)
27.9 字节和字符之间的转换	(581)	30.1.2 TCP/IP 网络模型	(630)
27.10 StringBufferInputStream	(582)	30.2 TCP/IP 协议	(633)
27.11 对象流	(582)	30.2.1 Internet 协议(IP)	(633)
27.12 其它流	(585)	30.2.2 传输控制协议(TCP)	(633)
27.12.1 LineNumberReader 类	(585)	30.2.3 用户数据报协议(UDP)	(634)
27.12.2 SequenceInputStream 类	(586)	30.3 统一资源定位器(URL)	(634)
27.12.3 PushbackInputStream 类	(586)	30.3.1 URL 语法	(634)
27.13 StreamTokenizer 类	(587)	30.3.2 一般 URL 格式	(634)
第 28 章 使用字符串和文本	(590)	30.4 Java 和 URL	(635)
28.1 字符串简介	(591)	30.4.1 URL 类	(635)
28.2 使用 String 类	(592)	30.4.2 连接到 URL 上	(636)
28.2.1 获取关于 String 对象的信息	(593)	30.4.3 以 HTTP 为中心类	(637)
28.2.2 比较字符串	(596)	30.4.4 示例: 定制 AltaVista 搜索引擎	(637)
28.2.3 字符串抽取	(599)	第 31 章 TCP 套接字	(644)
28.2.4 字符串操作	(602)	31.1 TCP 套接字基础	(645)
28.3 使用 StringBuffer 类	(603)	31.1.1 什么是套接字	(645)
28.3.1 创建 StringBuffer 对象	(603)	31.1.2 Java TCP 套接字类	(647)
28.3.2 获取关于 StringBuffer 对象的 信息	(604)	31.1.3 定制套接字行为	(649)
28.3.3 StringBuffer 对象	(604)	31.2 创建 TCP 客户/服务器应用程序	(650)

31.2.1 设计一个应用程序协议	(650)	33.9.3 编写一个服务器程序	(702)
31.2.2 开发股票客户端程序	(651)	33.10 InetAddress 类	(706)
31.2.3 开发股票报价服务器程序	(656)	33.10.1 转换名字为地址	(706)
31.2.4 运行客户端和服务器应用程序	(663)	33.10.2 检验 InetAddress	(707)
第 32 章 UDP 套接字	(664)	33.10.3 获得一个 Java 小程序的起始 地址	(707)
32.1 UDP 信息概览	(665)	33.11 DatagramSocket 类	(707)
32.1.1 UDP 套接字的特性	(665)	33.12 DatagramPacket 类	(709)
32.1.2 Java UDP 类	(666)	33.12.1 广播数据报	(709)
32.2 创建 UDP 服务器程序	(667)	33.12.2 一个简单的数据报服务器 程序	(710)
32.2.1 启动服务器	(671)	33.13 多点发送套接字	(712)
32.2.2 StartServing()方法处理请求	(671)	第 34 章 Java 安全性深析	(714)
32.2.3 getTimeBuffer()方法创建字节 数组	(671)	34.1 Java 安全性的必要性	(715)
32.2.4 运行时间服务器程序	(671)	34.2 Java 安全性框架	(716)
32.3 创建 UDP 客户端程序	(671)	34.2.1 第一部分:语言所提供的安全 措施	(716)
32.3.1 启动 TimeCompare	(676)	34.2.2 第二部分:Java 编译器	(717)
32.3.2 getTimeTimes()方法:TimeCompare 的执行路径	(677)	34.2.3 第三部分:检验器	(717)
32.3.3 printTimes()方法:显示比较 结果	(677)	34.2.4 第四部分:类装载器	(718)
32.3.4 运行应用程序	(677)	34.2.5 第五部分:建立安全性策略	(718)
32.4 使用 IP 多点发送	(678)	34.2.6 总结	(718)
32.4.1 Java 的多点发送	(679)	34.3 Java 小程序的限制	(719)
32.4.2 多点发送应用程序	(680)	34.3.1 Java 小程序与应用程序	(719)
第 33 章 java.net	(686)	34.3.2 SecurityManager 类	(720)
33.1 URL 类	(687)	34.3.3 Java 浏览器的安全性策略	(720)
33.1.1 获取 URL 内容	(688)	34.4 Java 安全性问题	(723)
33.1.2 获取 URL 信息	(688)	34.4.1 已知的缺陷	(724)
33.2 URLConnection 类	(689)	34.4.2 拒绝服务攻击	(725)
33.3 HTTPURLConnection 类	(691)	34.5 Java 安全性 API:扩展 Java 小程序 边界	(725)
33.4 URLEncoder 类	(691)	34.5.1 均衡密码技术(Symmetric Cryptography)	(726)
33.5 URLDecoder 类	(692)	34.5.2 公共密钥加密	(727)
33.6 URLStreamHandler 类	(692)	34.5.3 资格授权	(728)
33.7 ContentHandler 类	(693)	34.5.4 实现了什么	(729)
33.8 Socket 类	(693)	34.5.5 密钥管理	(729)
33.8.1 发送和接收 Socket 数据	(694)	34.5.6 为一个 JAR 文件数字签名	(730)
33.8.2 获取 Socket 信息	(695)	34.5.7 定义策略	(731)
33.8.3 设置 Socket 选项	(695)	34.5.8 运行 Java 小程序	(731)
33.8.4 关闭 Socket 连接	(696)	34.6 安全性 API	(731)
33.8.5 等待到来的数据	(696)	34.6.1 公共密钥和私人密钥类	(731)
33.8.6 一个简单的 Socket 客户端程序	(698)	34.6.2 Signature 类	(732)
33.9 ServerSocket 类	(701)	34.6.3 身份及签名者	(732)
33.9.1 接收进入的套接字连接	(701)		
33.9.2 获得服务器套接字地址	(702)		

34.6.4 资格证	(733)	37.3.2 管理对象中定义方法	(772)
34.6.5 IdentityScope 类	(733)	37.3.3 编译管理对象	(773)
第 35 章 对象序列化	(736)	37.3.4 导入管理对象	(773)
35.1 什么是对象序列化	(737)	37.4 Admin View Module	(773)
35.1.1 对象序列化如何运作	(740)	37.4.1 内容管理器和可选择对象	(773)
35.1.2 使用对象引用处理对象	(740)	37.4.2 图标	(777)
35.2 对象序列化示例	(740)	37.4.3 属性册	(780)
35.2.1 一个写 Date 类的应用程序	(740)	37.4.4 任务页	(781)
35.2.2 在 JDK1.02 下运行 DateWrite	(742)	37.4.5 对话框	(784)
35.2.3 编译和运行 DateWrite	(742)	37.4.6 自验证字段	(789)
35.2.4 一个读入 Date 类的简单应用 程序	(742)	37.4.7 新布局管理器	(791)
35.2.5 编译和运行 DateRead	(744)	第五部分 数据库	(793)
35.2.6 使用 Java 小程序读入 Date 类	(744)	第 38 章 数据库介绍	(795)
35.3 写和读自己的对象	(745)	38.1 ODBC 和 JDBC	(796)
35.4 定制对象序列化	(747)	38.2 关系数据库概念	(796)
第 36 章 远程方法调用	(750)	38.2.1 SQL	(798)
36.1 什么是远程方法调用	(751)	38.2.2 连接	(798)
36.1.1 创建远程对象	(751)	38.3 ODBC 技术概述	(801)
36.1.2 一个 RMI 应用程序样本	(751)	38.3.1 ODBC 一致层	(803)
36.1.3 创建远程接口	(752)	38.3.2 ODBC 函数和命令集	(804)
36.1.4 创建实现类	(752)	38.4 高级客户/服务器概念	(805)
36.1.5 编译 RemoteServer	(754)	38.4.1 客户/服务器系统层	(805)
36.1.6 创建主干(Stub)	(754)	38.4.2 事务	(806)
36.1.7 创建客户程序	(755)	38.4.3 数据光标	(807)
36.1.8 启动注册并运行代码	(756)	38.4.4 复制	(809)
36.1.9 绑定 RemoteObject 到注册	(757)	第 39 章 JDBC:Java 数据库连接	(810)
36.1.10 运行客户程序	(757)	39.1 JDBC 概述	(811)
36.2 创建 Java 小程序客户程序	(757)	39.1.1 JDBC 是如何工作的	(811)
36.3 创建定制套接字	(758)	39.1.2 安全模型	(813)
36.3.1 创建一个定制套接字	(759)	39.1.3 JDBC-ODBC 桥接器	(813)
36.3.2 创建定制的 RMISocketFactory	(762)	39.2 JDBC 实现	(813)
36.3.3 在应用程序中指定套接字工厂	(763)	39.2.1 JDBC 类——概述	(814)
36.4 使用激活模型	(765)	39.2.2 JDBC 应用程序剖析	(816)
36.4.1 创建可激活对象	(766)	39.2.3 JDBC 实例	(816)
第 37 章 管理 API	(769)	39.3 Connection 类	(819)
37.1 JMAPI 组件	(770)	39.4 元数据函数	(820)
37.2 JMAPI 小程序	(770)	39.4.1 DatabaseMetaData	(821)
37.2.1 JmapiHome.html	(770)	39.4.2 ResultSetMetaData	(825)
37.2.2 MOContentManagerApplet.html	(771)	39.5 SQLException 类	(826)
37.2.3 MOPropertyBookApplet.html	(771)	39.6 SQLWarning 类	(827)
37.2.4 访问管理对象	(771)	第 40 章 JDBC 考察	(828)
37.3 创建管理对象	(772)	40.1 语句	(829)
37.3.1 管理对象中定义属性	(772)	40.1.1 Statement	(830)