

Java领域最有影响力和价值的著作之一，与《Java编程思想》齐名，10余年全球畅销不衰，广受好评

根据Java SE 7全面更新，全面、深入地讲解Java中所有高级特性和开发技巧，包含大量案例，实践性强

PEARSON



Java 核心技术 卷II

高级特性 (原书第9版)

Core Java Volume II—Advanced Features
(Ninth Edition)

(美) Cay S. Horstmann Gary Cornell 著
陈昊鹏 王浩 姚建平 等译



机械工业出版社
China Machine Press



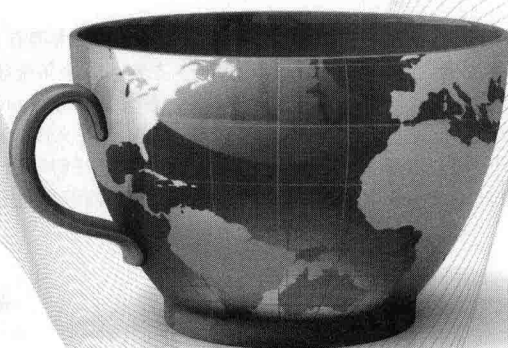
Java

核心技术 卷II

高级特性 (原书第9版)

Core Java Volume II—Advanced Features
(Ninth Edition)

(美) Cay S. Horstmann Gary Cornell 著
陈昊鹏 王浩 姚建平 等译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Java 核心技术 卷 II 高级特性 (原书第 9 版)/(美) 霍斯特曼 (Horstmann, C. S.), (美) 科内尔 (Cornell, G.) 著; 陈昊鹏等译. —北京: 机械工业出版社, 2013.12

(Java 核心技术系列)

书名原文: Core Java Volume II—Advanced Features (Ninth Edition)

ISBN 978-7-111-44250-9

I. J… II. ①霍… ②科… ③陈… III. Java 语言—程序设计—英文 IV. TP312

中国版本图书馆 CIP 数据核字 (2013) 第 235224 号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2013-4456

Authorized translation from the English language edition, entitled *Core Java, Volume II—Advanced Features, Ninth Edition*, 978-0-13-708160-8 by Cay S. Horstmann, Gary Cornell, published by Pearson Education, Inc., Copyright © 2013 Oracle and / or its affiliates.

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 Pearson Education, Inc.

Chinese Simplified language edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2014.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括中国台湾地区和香港、澳门特别行政区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

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

Java 领域最有影响力和价值的著作之一, 由拥有 20 多年教学与研究经验的资深 Java 技术专家撰写 (获 Jolt 大奖), 与《Java 编程思想》齐名, 10 余年全球畅销不衰, 广受好评。第 9 版根据 Java SE 7 全面更新, 同时修正了第 8 版中的不足, 系统全面讲解 Java 语言的核心概念、语法、重要特性和开发方法。本书全面覆盖 Java 技术的高级主题, 包括流与文件、XML、网络、数据库编程、国际化等, 详细描述了图形与 GUI 编程, 还涉及安全、远程方法、注解处理、本地方法等。本书对 Java 技术的阐述精确到位, 叙述方式深入浅出, 并包含大量示例代码, 能够帮助读者充分理解 Java 语言并灵活应用。

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 吴怡

藁城市京瑞印刷有限公司印刷

2014 年 2 月第 1 版第 1 次印刷

186mm × 240mm · 53.5 印张

标准书号: ISBN 978-7-111-44250-9

定 价: 139.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294

88379649 68995259

读者信箱: hzjsj@hzbook.com

译者序

《Java 核心技术》又推出了新版第 9 版！可见这套书在广大 Java 程序员和爱好者中心的地位。本书覆盖面广，几乎囊括了 Java 标准版的所有方面。这一版增加了有关 Java SE 7 的内容，并且几乎重新编写了全书所有的示例程序，使得用户能够更加容易地掌握 Java 的精髓。

《Java 核心技术，卷 II：高级特性》面向的是已经熟读并掌握了《Java 核心技术，卷 I：基础知识》内容的读者，或者是已经对 Java 语言的基本特性相当熟悉的读者。卷 II 的内容包含了流和文件、XML、网络、数据库编程、国际化、高级 AWT、高级 Swing、JavaBean 构件、安全、脚本以及注解、分布式对象和本地方法等内容，把读者引入了 Java 世界的更深处。

我们翻译《Java 核心技术，卷 II：高级特性》始于第 7 版，由于水平有限，书中存在不少问题。如果读者对比之前的版本，一定能够发现我们订正了书中的许多错误，使得全书的阅读变得更加流畅了。当然，我们不敢说翻译中的谬误已经全部被订正，因此恳请读者能够批评指正，让我们共同创造精良的译本。

第 9 版的勘定由陈昊鹏完成，本书之前版本由陈昊鹏、王浩、姚建平、龚斌、楼钢、李伟、郭嘉和方小丽等共同完成。

前 言

致读者

本书是按照 Java SE 7 完全更新后的《Java 核心技术，卷 II：高级特性（原书第 9 版）》。卷 I 主要介绍了 Java 语言的一些关键特性；而本卷主要介绍编程人员进行专业软件开发时需要了解的高级主题。因此，与本书之前的版本一样，我们仍将本书定位于用 Java 技术进行实际项目的编程人员。

请注意：如果你是一个经验丰富的开发人员，能够灵活运用像内部类和泛型这样的高级语言特性，那么你就不需要阅读完卷 I 再学习本卷。不过，本卷的某些地方引用了卷 I 的有关内容，（我们希望读者会购买或者已经购买了卷 I，）当然，读者也可以在任何一本综合介绍 Java 平台的书中找到所需的背景知识。

最后要说明的一点是，编写任何一本書籍都难免会有一些错误或不准确的地方。我们非常乐意听到读者的意见。当然，我们更希望对本书问题的报告只听到一次。为此，我们创建了一个 FAQ、bug 修正以及应急方案的网站 <http://horstmann.com/corejava>。你可以在 bug 报告网页（该网页的目的是鼓励读者阅读以前的报告）的末尾处添加 bug 报告，以此来发布 bug 和问题并给出建议，以便我们改进本书将来的版本。

内容提要

本书中的章节大部分是相互独立的。你可以研究自己最感兴趣的主体，并可以按照任意顺序阅读这些章节。

第 1 章的主题是输入输出处理。在 Java 中，所有 I/O 都是通过所谓“流”来处理的。流使你可以按照统一的方式来处理与各种数据源之间的通信，例如文件、网络连接或内存块。我们对各种读入器和写出器类进行了详细的讨论，它们使得对 Unicode 的处理变得很容易。我们还向你展示了：如何使用对象序列化机制从而使保存和加载对象变得容易而方便，及其背后的原理。最后，我们讨论了 Java SE 7 的正则表达式和 NIO2 类库，它们使某些通用操作变得非常便捷，例如读取文件中的所有行。

第 2 章介绍 XML，介绍怎样解析 XML 文件，怎样生成 XML 以及怎样使用 XSL 转换。在一个实用示例中，我们将展示怎样在 XML 中指定 Swing 窗体的布局。我们还讨论了 XPath API，它使得“在 XML 的干草堆中寻找绣花针”变得更加容易。

第 3 章介绍网络 API。Java 使复杂的网络编程工作变得很容易实现。我们将介绍怎样创

连接到服务器上，怎样实现你自己的服务器，以及怎样创建 HTTP 连接。

第 4 章介绍数据库编程，重点讲解 JDBC，即 Java 数据库连接 API，这是用于将 Java 程序与关系数据库进行连接的 API。我们将介绍怎样通过使用 JDBC API 的核心子集，编写能够处理实际的数据库日常操作事务的实用程序。（如果要完整介绍 JDBC API 的功能，可能需要编写一本像本书一样厚的书才行。）最后我们简要介绍了层次数据库，探讨了一下 JNDI（Java 命名及目录接口）以及 LDAP（轻量级目录访问协议）。

第 5 章讨论了一个我们认为其重要性将会不断提升的特性——国际化。Java 编程语言是少数几种一开始就被设计为可以处理 Unicode 的语言之一，不过 Java 平台的国际化支持则走得更加深远。因此，你可以对 Java 应用程序进行国际化，使得它们不仅可以跨平台，而且还可以跨越国界。例如，我们会展示怎样编写一个退休金计算器的 Applet，对它可以根据本地浏览器的情况使用英语、德语或者汉语进行浏览。

第 6 章涵盖了没有纳入卷 I 的所有 Swing 知识，尤其是重要但很复杂的树型构件和表格构件。随后我们介绍了编辑面板的基本用法、“多文档”界面的 Java 实现、在多线程程序中用到的进度指示器，以及诸如闪屏和支持系统托盘这样的“桌面集成特性”。我们仍着重介绍在实际编程中可能遇到的最为有用的构件，因为对 Swing 类库进行百科全书般的介绍可能会占据好几卷书的篇幅，并且只有专门的分类学家才感兴趣。

第 7 章介绍 Java 2D API，你可以用它来创建实际的图形和特殊的效果。该章还介绍了抽象窗口操作工具包（AWT）的一些高级特性，这部分内容看起来过于专业，不适合在卷 I 中介绍。虽然如此，这些技术还是应该成为每一个编程人员工具包的一部分。这些特性包括打印和用于剪切粘贴及拖放的 API。

第 8 章介绍了用于 Java 平台的构件 API——JavaBean。你将会看到怎样编写自己的 Bean，以及其他编程人员怎样在集成构建环境中对它们进行操作。最后我们展示怎样使用 JavaBean 的持久性，以某种与对象序列化不同的适用于长期存储的格式来存储自己的数据。

第 9 章继续介绍 Java 安全模型。Java 平台一开始就是基于安全而设计的，该章会带你深入内部，查看这种设计是怎样实现的。我们将展示怎样编写用于特殊应用的类加载器以及安全管理器。然后介绍允许使用消息、代码签名、授权以及认证和加密等重要特性的安全 API。最后，我们用一个使用 AES 和 RSA 加密算法的示例进行了总结。

第 10 章讨论了三种处理代码的技术。脚本机制和编译器 API 允许程序去调用使用诸如 JavaScript 或 Groovy 之类的脚本语言编写的代码，并且允许程序去编译 Java 代码。可以使用注解向 Java 程序中添加任意信息（有时称为元数据）。我们将展示注解处理器怎样在源码级别或者在类文件级别上收集这些注解，以及怎样运用这些注解来影响运行时的类行为。注解只有在工具的支持下才有用，因此，我们希望我们的讨论能够帮助你根据需要选择有用的注解处理工具。

第 11 章介绍分布式对象。我们详细介绍了 RMI（远程方法调用）。这个 API 可以让你运行分布在多台机器上的 Java 对象。


第 12 章介绍本地方法，这个功能可以让你调用为微软 Windows API 这样的特殊机制而


编写的各种方法。很显然，这种特性具有争议性：使用本地方法，那么 Java 平台的跨平台特性将会随之消失。虽然如此，每个为特定平台编写 Java 应用程序的专业开发人员都需要了解这些技术。有时，当你与不支持 Java 平台的设备或服务进行交互时，为了你的目标平台，你可能需要求助于操作系统 API。我们将通过展示如何从某个 Java 程序访问 Windows 注册表 API 来阐明这一点。


所有章节都按照最新版本的 Java 进行了修订，过时的材料都删除了，Java SE 7 的新 API 也都详细地进行了讨论。


约定

我们使用等宽字体表示计算机代码，这种格式在众多的计算机书籍中极为常见。各种图标的含义如下：

 **注意：**需要引起注意的地方。

 **提示：**有用的提示。

 **警告：**关于缺陷或危险情况的警告信息。

 **C++ 注意：**本书中有许多这类提示，用于解释 Java 程序设计语言和 C++ 语言之间的不同。如果你对这部分不感兴趣，可以跳过。

API Application Programming Interface 1.2

Java 平台配备有大量的编程类库或者应用编程接口（API）。当第一次使用某个 API 时，我们在每一节的末尾都添加了一个简短的描述。这些描述可能有点不太规范，但是比官方在线 API 文档更具指导性。接口的名字都是斜体的，就像许多官方文档一样。类、接口或方法名后面的数字是 JDK 的版本，表示在该版本中才引入了这些特性。

本书示例代码以程序清单的形式列举了出来，例如：

程序清单 1-1 ScriptTest.java

可以从网站 <http://horstmann.com/corejava>[⊖] 下载示例代码。

致谢

写书总是需要付出极大的努力，而重写也并不像看上去那么容易，特别是在 Java 技术方面，要跟上其飞快的发展速率，更是如此。一本书的面世需要众多有奉献精神的人共同努力，我非常荣幸地在此向整个《Java 核心技术》团队致谢。

⊖ 也可登录华章网站 (<http://www.hzbook.com>) 下载相关代码。——编辑注

Prentice Hall 出版社的许多人都提供了颇有价值的帮助，但是他们甘愿居于幕后。我希望他们都能够知道我是多么感谢他们付出的努力。与以往一样，我要热切地感谢我的编辑，Prentice Hall 出版社的 Greg Doench，他对本书从编写到出版进行全程掌舵，并使我可以十分幸福地根本意识不到幕后那些人的存在。我还非常感谢 Julie Nahil 在撰写上的支持，以及感谢 Dmitry Kirsanov 和 Alina Kirsanova 对手稿的编辑和排版。

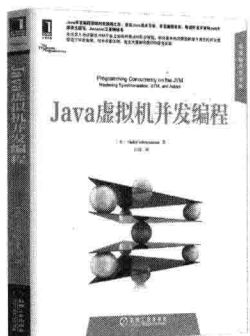
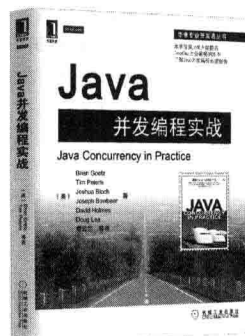
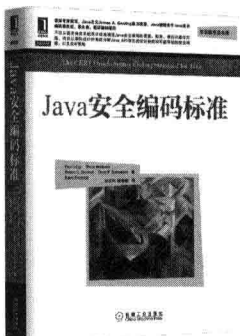
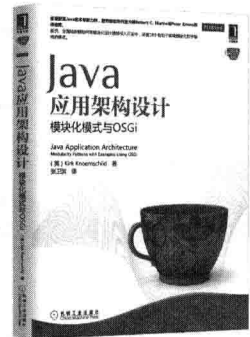
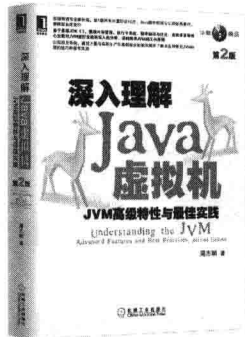
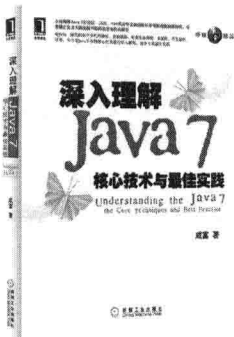
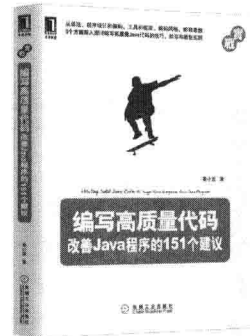
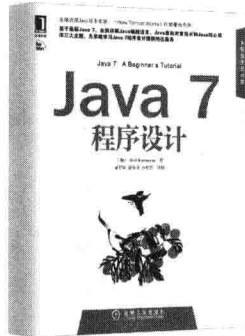
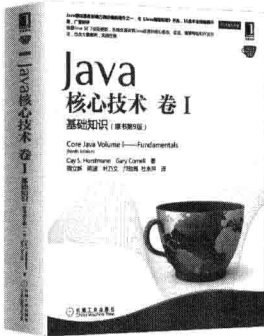
我非常感谢找到了很多令人尴尬的错误并提出了许多颇具创见性的建议的早先版本的读者。我特别要感谢十分出色的评审团队，他们用令人惊异的眼睛仔细浏览了所有原稿，并将我从许多令人尴尬的错误中拯救了出来。

这一版及以前版本是由以下人员评审的：Chuck Allison (特约编辑,《C/C++ Users Journal》)、(Lance Anderson (Oracle))、Alec Beaton (PointBase, Inc.)、Cliff Berg (iSavvix Corporation)、Joshua Bloch David Brown、Corky Cartwright、Frank Cohen (PushToTest)、Chris Crane (devXsolution)、Dr. Nicholas J. De Lillo (曼哈顿学院)、Rakesh Dhoopar (Oracle)、Robert Evans (资深教师, 约翰·霍普金斯大学应用物理实验室)、David Geary (Sabreware)、Jin Gish(oracle) Brian Goetz (首席顾问, Quiotix Corp.)、Angela Gordon、Dan Gordon、Rob Gordon、John Gray (Hartford 大学)、Cameron Gregory (olabs.com)、Marty Hall (约翰·霍普金斯大学应用物理实验室)、Vincent Hardy、Dan Harkey (圣何塞州立大学)、William Higgins (IBM)、Vladimir Ivanovic (PointBase)、Jerry Jackson (ChannelPoint Software)、Tim Kimmet (Preview Systems)、Chris Laffra、Charlie Lai、Angelika Langer、Doug Langston、Hang Lau (McGill 大学)、Mark Lawrence、Doug Lea (SUNY Oswego)、Gregory Longshore、Bob Lynch (Lynch Associates)、Philip Milne (顾问)、Mark Morrissey (俄勒冈研究生院)、Mahesh Neelakanta (佛罗里达大西洋大学)、Hao Pham、Paul Phillion、Blake Ragsdell、Ylber Ramadani (Ryerson 大学)、Stuart Reges (亚利桑那大学)、Rich Rosen (Interactive Data Corporation)、Peter Sanders (ESSI 大学, Nice, France)、Dr. Paul Sanghera (圣何塞州立大学和布鲁克学院)、Paul Sevinc (Teamup AG)、Devang Shah、Richard Slywczak (NASA/Glenn 研究中心)、Bradley A. Smith、Steven Stelting、Christopher Taylor、Luke Taylor (Valtech)、George Thiruvathukal、Kim Topley (《Core JFC 第 2 版》的作者)、Janet Traub、Paul Tyma (顾问) Peter van der Linden、Burt Walsh、Joe Wang(Oracle) 和 Dan Xu(Oracle)。

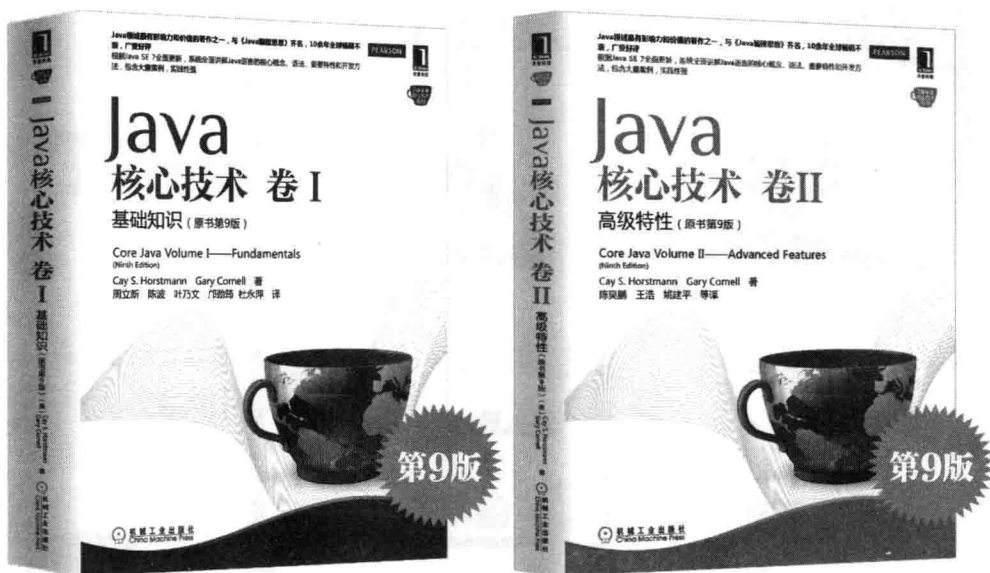
Cay Horstmann

于加州旧金山 2012 年 12 月

推荐阅读



推荐阅读



Java核心技术 卷I和卷II (第9版)

Java领域最有影响力和价值的著作之一，拥有20多年教学与研究经验的资深Java技术专家撰写（获Jolt大奖），与《Java编程思想》齐名，10余年全球畅销不衰，广受好评。第9版根据Java SE 7全面更新，同时修正了第8版中的不足，系统全面讲解Java语言的核心概念、语法、重要特性和开发方法，包含大量案例，实践性强。

目 录

译者序
前言

第 1 章 流与文件	1	1.7.1 缓冲区数据结构	62
1.1 流	1	1.7.2 文件加锁机制	64
1.1.1 读写字节	1	1.8 正则表达式	66
1.1.2 完整的流家族	3	第 2 章 XML	76
1.1.3 组合流过滤器	7	2.1 XML 概述	76
1.2 文本输入与输出	10	2.2 解析 XML 文档	81
1.2.1 如何写出文本输出	10	2.3 验证 XML 文档	91
1.2.2 如何读入文本输入	13	2.3.1 文档类型定义	92
1.2.3 以文本格式存储对象	13	2.3.2 XML Schema	98
1.2.4 字符集	16	2.3.3 实用示例	101
1.3 读写二进制数据	20	2.4 使用 XPath 来定位信息	113
1.4 ZIP 文档	27	2.5 使用命名空间	119
1.5 对象流与序列化	29	2.6 流机制解析器	121
1.5.1 理解对象序列化的文件格式	34	2.6.1 使用 SAX 解析器	122
1.5.2 修改默认的序列化机制	39	2.6.2 使用 StAX 解析器	126
1.5.3 序列化单例和类型安全的 枚举	40	2.7 生成 XML 文档	129
1.5.4 版本管理	42	2.7.1 不带命名空间的文档	130
1.5.5 为克隆使用序列化	44	2.7.2 带命名空间的文档	130
1.6 操作文件	46	2.7.3 写出文档	131
1.6.1 Path	46	2.7.4 示例：生成 SVG 文件	131
1.6.2 读写文件	48	2.7.5 使用 StAX 写 XML 文档	133
1.6.3 复制、移动和删除文件	49	2.8 XSL 转换	140
1.6.4 创建文件和目录	50	第 3 章 网络	151
1.6.5 获取文件信息	51	3.1 连接到服务器	151
1.6.6 迭代目录中的文件	53	3.1.1 套接字超时	155
1.6.7 ZIP 文件系统	55	3.1.2 因特网地址	156
1.7 内存映射文件	56	3.2 实现服务器	157
		3.2.1 为多个客户端服务	160
		3.2.2 半关闭	163
		3.3 可中断套接字	164
		3.4 获取 Web 数	171

3.4.1	URL 和 URI	171	4.9.3	高级 SQL 类型	243
3.4.2	使用 URLConnection 获取 信息	173	4.10	Web 与企业应用中的连接 管理	244
3.4.3	提交表单数据	181	第 5 章	国际化	246
3.5	发送 E-mail	187	5.1	Locales	246
第 4 章	数据库编程	191	5.2	数字格式	251
4.1	JDBC 的设计	191	5.3	日期和时间	257
4.1.1	JDBC 驱动程序类型	192	5.4	排序	264
4.1.2	JDBC 的典型用法	193	5.4.1	排序强度	265
4.2	结构化查询语言	194	5.4.2	分解	265
4.3	JDBC 配置	198	5.5	消息格式化	270
4.3.1	数据库 URL	199	选择格式	272	
4.3.2	驱动程序 JAR 文件	199	5.6	文本文件和字符集	273
4.3.3	启动数据库	199	源文件的字符编码	274	
4.3.4	注册驱动器类	200	5.7	资源包	275
4.3.5	连接到数据库	201	5.7.1	定位资源包	275
4.4	执行 SQL 语句	203	5.7.2	属性文件	276
4.4.1	管理连接、语句和结果集	206	5.7.3	包类	277
4.4.2	分析 SQL 异常	207	5.8	一个完整的例子	278
4.4.3	组装数据库	209	第 6 章	高级 Swing	293
4.5	执行查询操作	212	6.1	列表	293
4.5.1	预备语句	212	6.1.1	JList 构件	293
4.5.2	读写 LOB	218	6.1.2	列表模式	298
4.5.3	SQL 转义	219	6.1.3	插入和移除值	302
4.5.4	多结果集	220	6.1.4	值的绘制	304
4.5.5	获取自动生成键	221	6.2	表格	307
4.6	可滚动和可更新的结果集	222	6.2.1	简单表格	307
4.6.1	可滚动的结果集	222	6.2.2	表格模型	310
4.6.2	可更新的结果集	224	6.2.3	对行和列的操作	314
4.7	行集	228	6.2.4	单元格的绘制和编辑	328
4.7.1	构建行集	228	6.3	树	338
4.7.2	被缓存的行集	229	6.3.1	简单的树	339
4.8	元数据	231	6.3.2	节点枚举	352
4.9	事务	240	6.3.3	绘制节点	353
4.9.1	保存点	241	6.3.4	监听树事件	356
4.9.2	批量更新	241	6.3.5	定制树模型	362

6.4 文本构件	370	7.12.3 打印预览	520
6.4.1 文本构件中的修改跟踪 ...	371	7.12.4 打印服务程序	528
6.4.2 格式化的输入框	373	7.12.5 流打印服务程序	531
6.4.3 JSpinner 构件	388	7.12.6 打印属性	532
6.4.4 用 JEditorPane 显示 HTML ...	395	7.13 剪贴板	538
6.5 进度指示器	401	7.13.1 用于数据传递的类和接口 ...	539
6.5.1 进度条	401	7.13.2 传递文本	540
6.5.2 进度监视器	404	7.13.3 Transferable 接口和数据 风格	543
6.5.3 监视输入流的进度	407	7.13.4 构建一个可传递的图像 ...	545
6.6 构件组织器和装饰器	411	7.13.5 通过系统剪贴板传递 Java 对象	548
6.6.1 分割面板	411	7.13.6 使用本地剪贴板来传递 对象引用	552
6.6.2 选项卡面板	414	7.14 拖放操作	552
6.6.3 桌面面板和内部框体	419	7.14.1 Swing 对数据传递的支持 ...	553
6.6.4 级联与平铺	422	7.14.2 拖曳源	557
6.6.5 否决属性设置	425	7.14.3 放置目标	559
第 7 章 高级 AWT	440	7.15 平台集成	566
7.1 绘图操作流程	440	7.15.1 闪屏	566
7.2 形状	442	7.15.2 启动桌面应用程序	571
7.3 区域	456	7.15.3 系统托盘	576
7.4 笔划	458	第 8 章 JavaBean 构件	581
7.5 着色	465	8.1 为何使用 Bean	581
7.6 坐标变换	466	8.2 编写 Bean 的过程	583
7.7 剪切	471	8.3 使用 Bean 构造应用程序	585
7.8 透明与组合	473	8.3.1 将 Bean 打包成 JAR 文件 ...	585
7.9 绘图提示	481	8.3.2 在开发环境中组合 Bean ...	586
7.10 图像的读取器和写入器	486	8.4 Bean 属性与事件的命名模式 ...	591
7.10.1 获得图像文件类型的 读取器和写入器	487	8.5 Bean 属性的类型	594
7.10.2 读取和写入带有多个 图像的文件	488	8.5.1 简单属性	594
7.11 图像处理	496	8.5.2 索引属性	594
7.11.1 构建光栅图像	496	8.5.3 绑定属性	595
7.11.2 图像过滤	502	8.5.4 约束属性	596
7.12 打印	510	8.6 BeanInfo 类	603
7.12.1 图形打印	510	8.7 属性编辑器	607
7.12.2 打印多页文件	518		

8.8	定制器	616	10.1.1	获取脚本引擎	716
8.9	JavaBean 持久化	624	10.1.2	脚本赋值与绑定	717
8.9.1	JavaBean 持久化可用于 任何数据	628	10.1.3	重定向输入和输出	719
8.9.2	一个 JavaBean 持久化的 完整示例	633	10.1.4	调用脚本的函数和方法	720
第 9 章	安全	644	10.1.5	编译脚本	722
9.1	类加载器	645	10.1.6	一个示例：用脚本处理 GUI 事件	722
9.1.1	类加载器的层次结构	646	10.2	编译器 API	727
9.1.2	将类加载器作为命名空间	647	10.2.1	编译便捷之法	727
9.1.3	编写你自己的类加载器	648	10.2.2	使用编译工具	727
9.2	字节码校验	653	10.2.3	一个示例：动态 Java 代码 生成	732
9.3	安全管理器与访问权限	657	10.3	使用注解	737
9.3.1	Java 平台安全性	658	10.4	注解语法	743
9.3.2	安全策略文件	661	10.5	标准注解	746
9.3.3	定制权限	667	10.5.1	用于编译的注解	747
9.3.4	实现权限类	669	10.5.2	用于管理资源的注解	748
9.4	用户认证	674	10.5.3	元注解	748
9.5	数字签名	687	10.6	源码级注解处理	750
9.5.1	消息摘要	688	10.7	字节码工程	756
9.5.2	消息签名	691	第 11 章	分布式对象	765
9.5.3	校验签名	693	11.1	客户与服务器的角色	765
9.5.4	认证问题	695	11.2	远程方法调用	767
9.5.5	证书签名	697	11.3	RMI 编程模型	769
9.5.6	证书请求	698	11.3.1	接口与实现	769
9.6	代码签名	699	11.3.2	RMI 注册表	770
9.6.1	JAR 文件签名	699	11.3.3	部署程序	774
9.6.2	软件开发者证书	702	11.3.4	记录 RMI 活动的日志	776
9.7	加密	704	11.4	远程方法中的参数和返回值	778
9.7.1	对称密码	705	11.4.1	传递远程对象	778
9.7.2	密钥生成	706	11.4.2	传递非远程对象	779
9.7.3	密码流	710	11.4.3	动态类加载	781
9.7.4	公共密钥密码	711	11.4.4	具有多重接口的远程引用	785
第 10 章	脚本、编译与注解处理	716	11.4.5	远程对象与 equals、hashCode 和 clone 方法	786
10.1	Java 平台的脚本	716	11.5	远程对象激活	786

第 12 章 本地方法	792	12.6.4 替代方法调用	815
12.1 从 Java 程序中调用 C 函数 ...	793	12.7 访问数组元素	816
12.2 数值参数与返回值	797	12.8 错误处理	820
12.3 字符串参数	799	12.9 使用调用 API	824
12.4 访问域	804	12.10 完整的示例: 访问 Windows 注册表	829
12.4.1 访问实例域	804	12.10.1 Windows 注册表概述 ...	829
12.4.2 访问静态域	808	12.10.2 访问注册表的 Java 平台 接口	830
12.5 编码签名	809	12.10.3 以本地方法方式实现 注册表访问函数	830
12.6 调用 Java 方法	810		
12.6.1 实例方法	810		
12.6.2 静态方法	813		
12.6.3 构造器	814		

第 1 章 流与文件

- ▲ 流
- ▲ 文本输入与输出
- ▲ 读入和写出二进制数据
- ▲ ZIP 文档
- ▲ 对象流与序列化
- ▲ 使用文件
- ▲ 内存映射文件
- ▲ 正则表达式

本章将介绍 Java 中用于输入和输出的各种应用编程接口 (Application Programming Interface, API)。你将要学习如何访问文件与目录, 以及如何以二进制格式和文本格式来读写数据。本章还要向你展示对象序列化机制, 它可以使存储对象像存储文本和数值数据一样容易。然后, 我们将介绍在 Java SE 1.4 中引入的“新 I/O”包 `java.nio` 所带来的种种改进, 以及 Java 7 的“新新 I/O”改进。最后, 本章将讨论正则表达式, 尽管这部分内容实际上与流和文件并不相关, 但是我们确实也找不到更合适的地方来处理这个话题。很明显, Java 设计团队在这个问题的处理上和我们一样, 因为正则表达式 API 的规格说明隶属于“新 I/O”特性的规格说明。

1.1 流

在 Java API 中, 可以从其中读入一个字节序列的对象称做输入流, 而可以向其中写入一个字节序列的对象称做输出流。这些字节序列的来源地和目的地可以是文件, 而且通常都是文件, 但是也可以是网络连接, 甚至是内存块。抽象类 `InputStream` 和 `OutputStream` 构成了输入/输出 (I/O) 类层次结构的基础。

因为面向字节的流不便于处理以 Unicode 形式存储的信息 (回忆一下, Unicode 中每个字符都使用了多个字节来表示), 所以从抽象类 `Reader` 和 `Writer` 中继承出来了一个专门用于处理 Unicode 字符的单独的类层次结构。这些类拥有的读入和写出操作都是基于两字节的 Unicode 码元的, 而不是基于单字节的字符。

1.1.1 读写字节

`InputStream` 类有一个抽象方法:

```
abstract int read()
```

这个方法将读入一个字节, 并返回读入的字节, 或者在遇到输入源结尾时返回 `-1`。在设计具体的输入流类时, 必须覆盖这个方法以提供适用的功能, 例如, 在 `FileInputStream` 类中, 这个方法将从某个文件中读入一个字节, 而 `System.in` (它是 `InputStream` 的一个

子类的预定义对象)却是从键盘读入信息。

`InputStream` 类还有若干个非抽象的方法, 它们可以读入一个字节数组, 或者跳过大量的字节。这些方法都要调用抽象的 `read` 方法, 因此, 各个子类都只需覆盖这一个方法。

与此类似, `OutputStream` 类定义了下面的抽象方法:

```
abstract void write(int b)
```

它可以向某个输出位置写出一个字节。

`read` 和 `write` 方法在执行时都将阻塞, 直至字节确实被读入或写出。这就意味着如果流不能被立即访问(通常是因为网络连接忙), 那么当前的线程将被阻塞。这使得在这两个方法等待指定的流变为可用的这段时间里, 其他的线程就有机会去执行有用的工作。

`available` 方法使我们可以去检查当前可读入的字节数量, 这意味着像下面这样的代码片段就不可能被阻塞:

```
int bytesAvailable = in.available();
if (bytesAvailable > 0)
{
    byte[] data = new byte[bytesAvailable];
    in.read(data);
}
```

当你完成对流的读写时, 应该通过调用 `close` 方法来关闭它, 这个调用会释放掉十分有限的操作系统资源。如果一个应用程序打开了过多的流而没有关闭, 那么系统资源将被耗尽。关闭一个输出流的同时还会冲刷用于该输出流的缓冲区: 所有被临时置于缓冲区中, 以便用更大的包的形式传递的字符在关闭输出流时都将被送出。特别是, 如果不关闭文件, 那么写出字节的最后一个包可能将永远也得不到传递。当然, 我们还可以用 `flush` 方法来人为地冲刷这些输出。

即使某个流类提供了使用原生的 `read` 和 `write` 功能的某些具体方法, 应用系统的程序员还是很少使用它们, 因为大家感兴趣的数据可能包含数字、字符串和对象, 而不是原生字节。

Java 提供了众多从基本的 `InputStream` 和 `OutputStream` 类导出的类, 这些类使我们可以处理那些以常用格式表示的数据, 而不只是字节。

API java.io.InputStream 1.0

- `abstract int read()`

从数据中读入一个字节, 并返回该字节。这个 `read` 方法在碰到流的结尾时返回 `-1`。

- `int read(byte[] b)`

读入一个字节数组, 并返回实际读入的字节数, 或者在碰到流的结尾时返回 `-1`。这个 `read` 方法最多读入 `b.length` 个字节。

- `int read(byte[] b, int off, int len)`

读入一个字节数组。这个 `read` 方法返回实际读入的字节数, 或者在碰到流的结尾时返回 `-1`。