



Professional Java JDK 6 Edition

Java

高级编程(JDK 6版)

W. Clay Richardson
[美] Donald Avondolio 著
Scot Schrager
Mark W. Mitchell
Jeff Scanlon
黄湘情 黄亚葵 译

Java

高级编程(JDK 6版)

W. Clay Richardson
Donald Avondolio
[美] Scot Schrager 著
Mark W. Mitchell
Jeff Scanlon
黄湘情 黄亚葵 译

人民邮电出版社
北京

图书在版编目 (CIP) 数据

Java 高级编程: JDK 6 版 / (美) 理查森 (Richardson, W.C.) 等著; 黄湘情, 黄亚葵译. —北京: 人民邮电出版社, 2007.10

ISBN 978-7-115-16424-7

I. J… II. ①理…②黄…③黄… III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 090580 号

版 权 声 明

W.Clay Richardson, Donald Avondolio, Scot Schrager, Mark W. Mitchell, and Jeff Scanlon
Professional Java®, JDK®6 Edition

Copyright © 2007 by Wiley Publishing, Inc., Indianapolis, Indiana.

All right reserved. This translation published under license.

Authorized translation from the English language edition published by Wiley Publishing, Inc..

本书中文简体字版由 W Publishing 公司授权人民邮电出版社出版, 专有版权属于人民邮电出版社。

Java 高级编程 (JDK 6 版)

- ◆ 著 [美] W. Clay Richardson Donald Avondolio
Scot Schrager Mark W. Mitchell Jeff Scanlon
- 译 黄湘情 黄亚葵
- 责任编辑 陈 昇
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 800×1000 1/16
印张: 45.75
字数: 1 137 千字 2007 年 10 月第 1 版
印数: 1-4 000 册 2007 年 10 月北京第 1 次印刷

著作权合同登记号 图字: 01-2006-4031 号

ISBN 978-7-115-16424-7/TP

定价: 99.00 元

读者服务热线: (010)67132705 印装质量热线: (010)67129223

内容提要

本书全面而深入地探讨了 Java 技术的高级特性，包括许多与 Java 语言相关的开源技术，如 Ant、Jmeter、Hibernate 等。全书共 14 章，其中前 3 章介绍了高效 Java 开发人员所需的特定思维方式，包括关键的 Java 语言特性和库、用于开发 Java 解决方案的各种工具和技术，以及 Java 开发模式。后面部分详细探讨了开发 Java 解决方案的各种工具和技术，包括 JFC、持久性、以页面为中心的开发技术、以组件为中心的开发技术、EJB 3、组件通信、面向服务的集成、Java 安全性、打包和部署等。

本书是 Java 语言初级教材、开发方法学书籍与专业 Java API 书籍之间的桥梁。书中提供了各种技术和工具的具体解决方案，并有大量的书上代码和在线代码可供读者进行实际操作使用。本书适合于具备 Java 入门知识的读者、Java 开发人员和高级 Java 解决方案设计人员阅读。

关于作者

W. Clay Richardson 是一位软件咨询顾问，主要关注高度专业化业务流程的敏捷 Java 解决方案。他曾经涉足过多个 Java 解决方案，担任过高级架构师、开发负责人和程序经理。他是 *More Java Pitfalls*、*Professional Portal Development with Open Source Tools* (Wiley 出版) 和 *Professional Java, JDK5 Edition* 的合著者。他还是弗吉尼亚工学院 (Virginia Tech) 计算机科学系的副教授，讲授有关 Java 面向对象开发的研究生课程。他拥有弗吉尼亚工学院和弗吉尼亚军事学院的学位。

Donald Avondolio 目前是某个企业开发项目的主要架构师/开发人员。在业余时间，Donald 喜欢用假蝇钓鱼、观看棒球和曲棍球比赛、参加三项全能运动 (成绩不是很好)，以及坐在家里抱怨一些事情。

Scot Schrager 在制药、供应链管理和国家安全市场等领域提供过广泛的咨询服务。他曾经领导并参与了各种使用 Java 及面向对象分析和设计技术的项目团队。最近，Schrager 主要关注使用 J2EE 技术的分布式应用程序体系结构。

Mark W. Mitchell 在企业级应用集成方面 (特别是 Java 与 Microsoft 平台之间的 Web 服务集成方面) 有着丰富的经验。他曾经开发和部署过多个任务关键型 Web 应用程序。Mitchell 拥有弗吉尼亚大学的计算机科学学位。

Jeff Scanlon 是来自弗吉尼亚的一位软件开发顾问。他拥有 Sun Certified Java Developer (Sun 认证的 Java 开发人员) 和 Microsoft Certified Solutions Developer (微软认证的解决方案开发人员) 证书，并且在 *Software Development* 杂志中发表过文章。

致 谢

首先,如果没有我的贤妻 Alicia 的支持,我就不可能有任何机会来完成此书。她和我的女儿 Jennifer 是我一生的幸福所在,我期望可以有更多的时间同她们在一起。我无法用语言来描述我是如何地爱她们两个人。Stephanie,我们爱你,并且将永远不会忘记你。感谢我们的技术编辑 David Parks,感谢他对此项目所做的杰出贡献,在我怂恿你参加此项工作时,你一点也不知道将要做的事。我的合作伙伴们——Donnie、Mark、Scot 和 Jeff 在一个要求苛刻的项目中付出了惊人的艰苦工作。我对你们中的每一位为本书作出的贡献而感激不尽。感谢 Bob Elliott 和 Brian Herrmann,感谢他们在同我们一起为本书所付出的辛苦和持之以恒的工作。感谢我的各级领导,Joe Duffy、Bruce Feldman、Jim Moorhead、Don Heginbotham、Jon Gransmeder 和 Augie Dunheimer,感谢他们为了实现“为正确的人做正确的事”这个简单理念而作出的努力。在一个体现各方共赢局面的团体中工作是令人振奋的。我还要感谢我的父母 Bill 和 Kay、我的岳父母 Stephen 和 Elaine Mellman、我的姐姐 Kari、我的姐夫 Grayson、我的弟弟 Morgan 以及我的继父 Dave,感谢他们对我的一贯支持。感谢我的祖母和外祖母 Vivian 和 Sophie,感谢她们对我的关爱。

感谢我的团队成员,感谢他们每天都在从事着使世界变得更美好的伟大工作:Joe Simasek、Rob Brown、Keith Berman、Mauro Marcellio、Terry Trepel(欢迎从伊拉克归来!）、Marshall Sayen、Joe Sayen、Hanchol Do、Scot Schrager、Don Avondolio、Brian Stearns、Cliff Toma、Mike Clarke、Brad Philips、Jeff Lane、Nhon Pham、Julia Lee、Vic Fraenckel(欢迎你死里逃生!）、Morgan Ruther、Lonni Haaland、George Burgin 和 Mark (Mojo) Mitchell。Matt Tyrrell,我原想写一些幽默或者有趣的事情,但是我想 Jennifer 这句话更恰当“Matt 叔叔什么时候上我们家来?”。

——WCR

首先,我要感谢这些在工作中一直激发我的人:Swati Gupta、Chi Louong、Bill Hickey 和 Chiming Huang。感谢费吉尼亚工学院计算机科学/信息技术系的所有权威教授们:Shawn Bohner、Tarun Sen、Stephen Edwards 和 John Viega。我的学生们通过他们的奉献、努力学习和洞察力教会我许多东西,洞察力使得我在本书中使用他们的开发理念作为指导。同样感谢 The Great Cow Harbor Run (Northport, NY) 和 The Columbia Triathlon (Columbia, MD) 的发起人、志愿者和组织者们,感谢他们组织了我喜欢参与的世界级活动,更重要的是感谢他们使我成为了一个更富有纪律性、更专注的人。尤其要感谢我的朋友们:Wieczoreks、Devaneys、Keanes、O'Donohoes、Howards 和 Pujols。

最后,我要感谢所有的合著者,他们是一群可以一起工作和交往的有趣的人:Jeff、Mark、Scot 和 Clay;感谢我的同事:Mauro Marcellio、Joe、Marshall Sayen、Joe Simasek、Terry Trepel、Hanchol Do、Keith Berman、Rob Brown、Dave Parks、Brian Stearns、Mike Clarke、Morgan Ruther、Cliff Toma、Matt Tyrrell、Thoman 家庭成员(Brettie、Cindy 和 Zoe)、Vic Fraenckel、Nhon Pham、Julia Lee,以及感谢我的钓鱼伙伴 George Burgin。感谢我所有的家庭成员:爸爸、妈妈、Michael、John、Patricia、Keil、

致 谢

Jim、Sue、Reenie、Donna、Kelly、Stephen、Emily、Jack、Gillian、Matt 和 Danielle，你们都是好样的。感谢我的妻子 Van，我爱她胜过一切，感谢她在本书写作期间对我一贯的支持。

——DJA

我要感谢的第一个人是 Heather。我们已经相处七年并且有了一个十一个月大的儿子，你使我成了世界上最幸运的人。感谢你所说的“我愿意”。我还要感谢我的父母。首先感谢我的妈妈，因为她始终站在我一边，支持我和我的决定。还有我的爸爸，因为我希望自己将来某天会成为那种类型的爸爸。我也希望感谢支持我的家庭成员；我的姐姐和他的家庭成员；Fern、Gary 和 Isabel。另外，我要感谢我其他的家庭成员：Joe、Sabina、Robin、Peter、Brandon、Abby、Christiana、Joe Jr.、Chris、Ann、Paige 和 Liam。我还需要感谢我的合作者们，是他们使得每天都成为一个体验。感谢 Clay 指引了方向并使得这项工作充满乐趣。我们不忙于讲述基本的知识，而是解决具体的问题。感谢 Don 容忍我。感谢 Dave 容忍 Don 和我。感谢 Cliff 所做的所有辛苦的工作，还要感谢 Gerry。我还要感谢 Marty、Melinda、Brett、Mike、Mark、Terry、Mauro、Marshall 和 Keith。

——SRS

首先我要感谢我的爱妻 Elisa，感谢她在本书创作期间对我的支持。像本书这样的项目永远是看起来要花的时间比实际花的要少，感谢她在我最乐观的时间估计下对我的支持。感谢 Clay，感谢他给我与他一起编写此书的机会，感谢 Don 通过本书对我的指导。我还要感谢我们的技术和开发编辑 Dave 和 Brian，感谢他们帮助我完成本书中的这些章节。感谢我最近一直亲密合作的同事：Keith、Jeff、Jon、Terry、Nhon、Matt、Marshall、Joe、Brad、Carlton、Todd、Bryan、Hanchol 和 Vic；还要感谢我过去亲密合作过的每个同事。仅仅通过观察和聆听我就学到了很多。没有比作为最低职位人员的工作环境更好的了——每天都可以从每个人身上学到东西。感谢我的父母，感谢爸爸激发我对计算机科学的兴趣，感谢妈妈帮助我如何学习写作。尤其要感谢上帝，因为编写本书是对信仰和信任的一个考验。最后但并非最不重要的是，我要感谢我所有的家庭成员和朋友，感谢他们在本书交稿期间对我的支持和理解。

——MWM

以下这些人应受到我的感谢：感谢 Dave Nelson，是他引导我进入编程的世界并且找到自己的定位；感谢我的父母和我的家庭成员；感谢我们的编辑们：Wiley、Brian Herrmann 和 Robert Elliott；还要感谢我们的技术评审人员 Dave Parks。感谢我最重要的老师 Alfred Martin 和 Paul D'Andrea。最后，感谢 Phil Bickel、Eric Anderton、John Tarcza、Keith Obenschain、Robert Burt、Joseph Kapp、Randy Nguyen、Leo Pak、Mark Orletsky、Randy Shine、David Hu、Min Soo Yi 和 Corey Chang 的支持。

——JS

前言

本书弥补了 Java 语言基本书籍和内容更加详细的书籍（介绍 EJB、JEE、JMX、JMS 等主题）之间的空白。大多数开发解决方案都涉及混合技术的使用，而有关所有这些技术的书籍可以擦上几英尺高。此外，读者只需要这些书中全部内容的一小部分来解决他们的特定问题。本书提供了该技术的背景信息、使用该技术的实践示例以及指引读者找到更多详细信息的一个说明。本书旨在作为 Java 开发人员的专业参考书。

本书适用对象

本书适合以下 3 种类型的读者。

- ❑ 读完 *Beginning Java* 之后的读者，本书可以为他们提供更高级的 Java 解决方案和语言特性。
- ❑ Java 开发人员，他们需要一本好的全面的参考书，作为处理超出其技术经验的新 Java 问题的第一参考。
- ❑ 已经具有某些解决方案经验的开发人员，例如，可以不必考虑是否值得花时间单独阅读有关 JMS 的 500 页来看看 JMS 是否能够适合其解决方案空间。本书能为这些开发人员提供克服难题的技术通行证。

本书包括的内容

本书是建立在 Ivor Horton 所著的 *Beginning Java 2* 之上的，向读者提供一种专家是如何使用 Java 来开发软件解决方案的思路。它从 Java 开发人员的工具和技术开始讨论，接着探讨 Java SDK 更高深和更细微的部分，最后提供几个使用 Java API 和开放源代码工具来构建实际 Java 解决方案的示例。本书全面探讨了高级 Java 开发领域的各个方面，也有对单个 API 的详尽讲述。本书是 Java 语言教材、方法学书籍与专业 Java API 书籍之间的桥梁。例如，一旦读者掌握了基本的 Java 语言，就会不可避免地遇到一个问题（例如构建一个数据库驱动的 Web 站点），这需要使用诸如 JSP 之类的技术和诸如 Hibernate 之类的工具来解决；而本书提供了集成这些技术和工具的一个具体解决方案。图 0-1 描述了本书内容与其他 Java 书籍之间的关系。当读者从 Java 入门书籍着手时，就可以使用本书作为一个解决方案初级读本，以引领读者进入更深层次关于特定主题（比如模式、Web 服务、JDBC）的书籍。

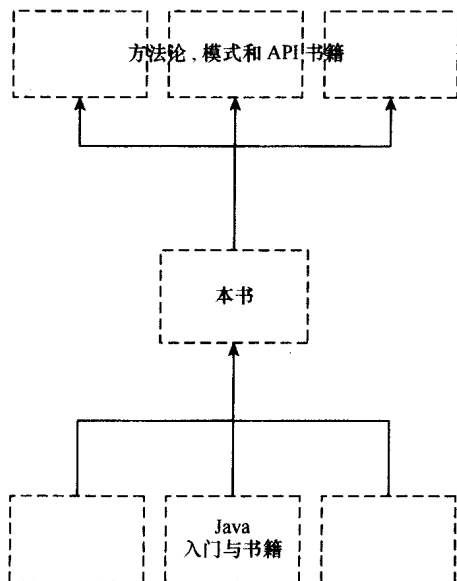


图 0-1

本书的组织结构

作为一名高效的专业 Java 开发人员，需要具备两种主要的能力：像 Java 开发人员一样的思考能力，以及拥有对 Java API、工具 and 技术的广泛理解能力，以解决大量的 Java 问题。浏览一下本书的结构，读者就可以明白本书中的这些章节是如何帮助自己提高这些技能的。

第一部分：像 Java 开发人员一样思考

有经验的 Java 开发人员认识到高效 Java 开发人员具备一种特定的思维方式。本书前 3 章将向读者讲述这个问题，帮助读者形成这种思维方式。

第 1 章：关键的 Java 语言特性和库

任何入门级 Java 书籍都会介绍 Java 编程语言的特性。本章介绍那些书籍中都不会涉及的许多关键的高级 Java 语言特性，例如正则表达式、首选项和 Java 日志。最重要的是，本章介绍了 Derby，这是一个轻量级数据库，是 Java 6 中的新增内容；并且回顾了 Java 2 标准 5.0 版中引入的语言特性，这些特性包括泛型、元数据以及自动装箱等。

第 2 章：用于开发 Java 解决方案的各种工具和技术

一个人从了解 Java 语言到成为一个 Java 开发人员是一个有趣的转变。一般来说，开发人员可以找到讲授语言的书籍和讲授方法学的书籍。而且，方法学书籍通常采用辩论式的书写风格，仿佛正在进行一次学术论文答辩或者开一个讨论会。这些书籍通常对其方法论规定了一些约定的形式，以免读

者失败。新开发人员会发现此方法令人疲惫不堪，因为很少有人一开始就可以负责指挥一个团队。在本书中，读者将发现开发人员是如何从实际视角来关注方法学和工具的，以便使用工具使开发工作更容易、更高效。

第3章：Java 开发模式

模式向开发人员提供了一个宝贵的资源，以尝试传达通用问题的解决方案。然而，由于软件问题一般是非常抽象的，理解它们的通用解决方案（乃至采用该方法的价值）也就必然是非常重要的。

然而，可以想象，在整个 Java 解决方案领域中存在重复出现的关键问题，因此，框架和 API 构建在模式之上。因而，如果一个人掌握了模式的使用方法，那么他将会无可争辩地成为一个高效的 Java 开发人员。本章将阐明模式非常重要的方面，提供对模式的实际了解，并且示范 Java 世界中常见模式的示例。

第二部分：对 Java API、工具和技术的广泛理解

Java 平台已经针对 3 种不同的平台扩展成了 3 种具有明显区别的版本，这远远超出了最初作为一个简单小应用程序开发语言的目标，不仅仅是平台正在发生着巨大变化，而且开放源代码运动和 Java 社区也增加了大量特性和工具，它们为 Java 开发人员提供更多选择。

因此，人们会发现自己很容易受到打击。本书这部分内容将提供 Java 开发领域中的一系列常见问题。在每个领域，将会介绍一个问题，并提供一个有针对性的解决方案。这些解决方案并不企图全面覆盖所有涉及的 API，只是提供解决该问题所需的一个基本方法。借助这部分内容，读者可以进入更专业的领域。本部分内容的主要目的在于，让读者无需为解决一个常见的开发问题从三尺高的书籍中寻找一个简单的端对端解决方案。

第4章：使用 JFC 开发高效的用户界面

Java 基础类（Java Foundation Class，JFC）通常简称为 Swing，它们提供构建用户界面和桌面应用程序的功能。因为这些类在各种入门 Java 书籍中频繁地生成大量逻辑示例，所以从逻辑开始讲解 Swing 示例是有意义的。然而，本章将更详尽地介绍 Swing 复杂的一面，包括一些类似于布局管理器和 Java 2D 之类的高级主题。

第5章：使用文件持久保存应用程序

对于任意应用程序来说，将能够做到的非常重要的事情之一就是保持（即保存）自己的状态。在本章中，读者将会发现实现保存和恢复功能的技术，这些技术使用两种不同的方法，即 Java 对象序列化和用于 XML 绑定的 Java API（Java API for XML Binding，JAXB）。

第6章：使用数据库持久保存应用程序

文件传统上用来在一个单线程模式中共享数据——一次一个用户。当数据必须在整个企业中共享时，就需要使用数据库。在本章中，读者将会学习到 Java 数据库连接性（Java Database Connectivity，JDBC）API 4.0 的更多高级特性。而且，本章还探讨了非常流行的对象持久性保存框架之一（并且是新 EJB 3.0 规范开发的基础）——Hibernate。

第7章：使用模型 1 体系结构开发 Web 应用程序

长时间从事 Web 应用程序开发的人都认识到以页面为中心的范型，也称为模型 1 体系结构

(Model 1 Architecture), 已经用于跨多种技术平台 (ASP、Code Fusion、Perl 等) 开发 Web 应用程序。Java 通过自己的 Java 服务器页面 (Java Server Page, JSP) 和 Java 标准标签库 (Java Standard Tag Library, JSTL) 规范来支持此范型。本章将介绍使用模型 1 体系结构来开发 Web 应用程序的框架和其他最佳实践。

第 8 章: 使用模型 2 体系结构开发 Web 应用程序

随着 Web 应用程序的发展, 人们已经认识到模型 1 体系结构的以页面为中心的方法存在着一些缺点。在本章中, 读者将了解到这些缺点并了解它们是如何促使模型 2 体系结构产生的, 模型 2 体系结构是以组件为中心的方法。读者将会看到如何使用一个类似于 WebWork 的组件框架来轻松地集成类似于 Hibernate 的其他组件。

第 9 章: 使用 Java 本机接口与 C/C++ 交互

经常会具有一些用非 Java 编程语言编写的应用程序组件, 而且自己的应用程序还需要经常访问这些组件。针对这个问题的解决方法就是 Java 本机接口 (Java Native Interface, JNI)。本章将解释 JNI 错综复杂的内容以及使用过程中可能易犯的错误。

第 10 章: EJB 3 和 Java 持久性 API

企业 JavaBeans (Enterprise JavaBeans, EJB) 是 Java 的分布式组件技术和 Java 2 企业版本 (Java 2 Enterprise Edition, Java 2 EE) 平台的基石。EJB 3.0 通过使用 Java 持久性 API 提供可靠、健壮且透明的对象持久性来利用简单 Java 对象 (Plain Old Java Object, POJO) 范型, 显著改进了 Java 语言。本章将解释 EJB 3 规范和 Java 持久性 API, 并且演示它们开发企业 Java 解决方案的实用程序。

第 11 章: 在 Java 组件和其他平台的组件之间通信

虽然 RMI 已经证明是用于 Java 到 Java 通信的良好解决方案, 但是仍然大大需要访问 (或者提供访问) 其他平台的组件。Microsoft.NET 平台尤其如此。本章将解释进程间通信的基础知识, 讨论进程间通信的几种技术, 并以一个使用 Web 服务的示例结束。

第 12 章: 面向服务的集成

执行许多个机器和平台之间分布式组件的企业应用程序集成时, 通常需要读者能够跨许多个不同步骤分散工作负荷。有两个 API 在此方面尤其有用, 它们就是 Java 消息服务 (Java Message Service, JMS) 和 Java 管理扩展 (Java Management Extensions, JMX)。在本章中, 读者将看到这两个 API 捆绑到一起以提供一个高度有用的体系结构的核心。

第 13 章: Java 安全性

信息安全性对于 Java 开发极其重要。在本章中, 读者将会看到自己的应用程序是如何使用 Java 授权和验证服务 (Java Authorization and Authentication Service, JAAS) 来实现安全性的, 以及自己的数据是如何使用 Java 加密扩展 (Java Cryptography Extensions, JCE) 来实现安全性的。同时本章还详细讲述了 Java 6 中引入的新 XML 数字签名支持, 它在构建 Web 服务安全性中很有用。

第 14 章: 打包和部署 Java 应用程序

不论是企业应用程序还是桌面应用程序, 有关开发 Java 应用程序的最棘手和最痛苦的事情之一就是打包和部署应用程序。在许多 API 中存在着大量的部署描述符和打包规则。包括 JAR、WAR、EAR 等。通常读者可从每本书籍中粗略了解到这些格式和规范。在本章中, 读者将会学习到 Java 中存在的

大量打包机制，以及对每种机制的部署描述符的描述。

使用本书的前提

本书是基于 J2SE 6.0 版本之上的。读者也许会发现拥有一个自己选择的集成开发环境（Integrated Development Environment, IDE）很有用，Eclipse 就是一个非常好并且流行的 IDE（www.eclipse.org）。而且，根据本书内容，读者也许需要使用一个类似于 JBoss（www.jboss.org）或者 Tomcat（<http://jakarta.apache.org/tomcat>）的应用服务器。应用服务器下载以及其他（API 等）的下载需要将会在每章中予以指出。

约定

为了让读者获取最佳阅读效果并记住本书中所讲述的内容，本书中使用了大量的约定。

诸如此类的方框提供了与前后文字直接相关的、不要忘记的重要信息。

对当前讨论内容的技巧、提示、诀窍和旁白像这样用斜体放置。

关于文本样式，本书中遵循如下标准：

- ❑ 在介绍重要词时对它们进行突出显示。
- ❑ 在使用组合按键时使用 Ctrl+A 的形式。
- ❑ 本文中的文件名称、URL 和代码类似于这样：persistence.properties。
- ❑ 代码存在有两种不同的表达方式：

在代码示例中，新的并且重要的代码用灰色背景进行突出显示。

本文中不太重要的代码或者以前已经显示过的代码就不使用灰色背景。

源代码

在使用本书中提供的示例时，读者可以选择手动输入所有代码，也可以选择直接使用本书附带的源代码文件。本书中使用的所有源代码都可以从 www.wrox.com 处下载。进入该站点后，只需要查找本书书名（通过使用 Search 框或者通过使用书名列表），然后单击本书的详细页面上的 Download Code 链接，即可获得本书的所有源代码。

因为许多书的书名与本书的雷同，所以最简单的方式是通过 ISBN 搜索；本书的 ISBN 是 978-0-471-77710-6。

下载了代码之后，读者只要使用自己喜欢的解压工具将代码解压即可。此外，还可以进入主要 Wrox 代码下载网页 www.wrox.com/dynamic/books/download.aspx 来查找本书以及所有其他 Wrox 书籍中

可用的源代码。

勘误

我们尽自己最大的努力来确保书中文字和代码不出现错误。然而，人无完人，因此总会有错误存在。如果读者在我们的某一本书中发现了一个错误（例如拼写错误或者错误代码段），我们将对您的反馈感激不尽。通过发送一个勘误，您就可以解决其他读者多时的困惑，同时有助于我们提供更高质量的信息。

要找到本书的勘误页，可以进入www.wrox.com站点，使用 Search 框或者使用书名列表来查找本书书名。然后，在本书详细页面上，单击 Book Errata 链接。在此页面上，可以查看本书所有已提交的以及 Wrox 编辑已公布的勘误。包括每本书的勘误链接的一个完整书籍列表可以在www.wrox.com/misc-pages/booklist.shtml处获得。

如果在书籍勘误页没有找到您所发现的错误，那么可以访问 www.wrox.com/contact/techsupport.shtml，在那里填写一个表格将您所发现的错误发送给我们。我们将检查您提交的信息，如果确认您的勘误正确，我们就会在本书勘误页面上发布这条消息，并在本书后续版本中修订该错误。

p2p.wrox.com

如果想与作者或者其他同行进行讨论，请加入<http://p2p.wrox.com>处的 P2P 论坛。这些论坛是基于 Web 的系统，读者可就 Wrox 书籍的相关内容和相关技术发表自己的见解，并且可以与其他读者和技术人员进行交流。这些论坛提供了一个电子邮件订阅功能，当论坛上新发布了读者感兴趣的话题内容时，论坛可以将其发送到读者的电子邮箱中。这些论坛当前的用户包括 Wrox 作者、编辑、其他业内专家以及同道读者们。

在<http://p2p.wrox.com>处将会发现许多不同的论坛，这些论坛不仅有助于阅读本书，还有助于开发自己的应用程序。按以下步骤进行操作就可以加入这些论坛。

1. 登录 p2p.wrox.com 站点，单击 Register 链接。
2. 阅读使用条款，单击 Agree。
3. 填写加入论坛所必需的信息以及希望提供的任意可选信息，单击 Submit。
4. 您的电子邮件中将会收到一封确认函，告诉您如何确认自己的账号并完成注册过程。

读者无需加入 P2P 就可以阅读这些论坛的信息，但要发布自己的信息，则必须加入论坛。

一旦加入了论坛，就可以发布自己的新消息，并对其他用户发布的消息进行跟贴。可以随时在 Web 站点阅读消息。如果希望一个特定的论坛发送新消息到自己的电子邮件中，可以单击论坛列表中论坛名称旁边的 Subscribe to this Forum 图标。

有关使用 Wrox P2P 的更多信息，可以阅读 P2P FAQ 获取关于论坛软件如何运行的问题的解答，以及许多特定于 P2P 和 Wrox 书籍的常见问题的解答。要阅读 FAQ，请单击 P2P 页面上的 FAQ 链接。

目 录

第一部分 像 Java 开发人员一样思考

第 1 章 关键的 Java 语言特性和库	3
1.1 介绍 Derby	3
1.2 使用 Derby	3
1.3 添加到 Java 5 中的语言特性	7
1.3.1 泛型	7
1.3.2 增强的 for 循环	16
1.3.3 Java 类库的增强	17
1.3.4 可变参数	18
1.3.5 装箱/拆箱转换	20
1.3.6 静态导入	22
1.3.7 枚举	24
1.3.8 元数据	26
1.4 重要的 Java 实用程序库	35
1.4.1 Java 日志	35
1.4.2 Java 首选项	67
1.5 小结	74
第 2 章 用于开发 Java 解决方案的各种工具和技术	76
2.1 高质量软件开发的原则	76
2.2 进行高效软件开发的习惯	77
2.2.1 交流	77
2.2.2 模型	78
2.2.3 敏捷	78
2.2.4 条理	79

2.2.5	根据需要追溯行为	79
2.2.6	不要害怕编写代码	79
2.2.7	将代码当作设计而非产品	80
2.2.8	不断学习	80
2.2.9	自下而上构建流程	80
2.2.10	配置管理	81
2.2.11	单元测试代码	81
2.2.12	持续集成	81
2.2.13	维护短期迭代	82
2.2.14	度量成果——间接度量	83
2.2.15	追踪问题	84
2.3	开发方法	84
2.3.1	瀑布方法	84
2.3.2	统一流程	85
2.3.3	极限编程	87
2.3.4	关于方法的评价	87
2.4	实际开发场景	88
2.4.1	Ant	89
2.4.2	Maven 2	97
2.4.3	原型	98
2.4.4	递移依赖关系	98
2.4.5	TestNG	103
2.4.6	XDoclet	107
2.4.7	JMeter	114
2.5	小结	117
第3章	Java 开发模式	118
3.1	模式为什么重要	119
3.1.1	理解Java编程语言的关键	119
3.1.2	理解Java开发中所使用工具的关键	120
3.1.3	开发有效Java解决方案的关键	120

3.2 基于设计原则构建模式	121
3.2.1 设计单个类	122
3.2.2 在类之间创建关联	122
3.2.3 创建接口	123
3.2.4 创建继承循环	124
3.3 重要的 Java 模式	125
3.3.1 适配器	125
3.3.2 模型-视图-控制器	128
3.3.3 命令	136
3.3.4 策略	141
3.3.5 组合	146
3.4 小结	149

第二部分 对 Java API、工具和技术的广泛理解

第 4 章 使用 JFC 开发高效的用户界面	153
4.1 布局管理器	154
4.1.1 BorderLayout	154
4.1.2 BoxLayout	161
4.1.3 FlowLayout	169
4.1.4 GridLayout	174
4.1.5 GridBagLayout	186
4.1.6 SpringLayout	191
4.1.7 CardLayout	200
4.1.8 GroupLayout	205
4.2 Mustang 版本桌面增强	213
4.3 管理 Swing 应用程序中的导航流	224
4.4 小结	234
第 5 章 使用文件持久保存应用程序	235
5.1 应用程序数据	235
5.1.1 保存应用程序数据	237

5.1.2	应用程序的示例配置数据模型	237
5.2	Java Serialization: 持久保存对象图	239
5.2.1	关键类	240
5.2.2	序列化对象	240
5.2.3	扩展和自定义序列化	254
5.2.4	使用Java Serialization的时机	258
5.3	JavaBean 长期序列化: XMLEncoder/Decoder	259
5.3.1	设计差异	259
5.3.2	序列化JavaBean	262
5.3.3	使用XMLEncoder/Decoder的时机	267
5.4	灵活的 XML 序列化: 用于 XML 绑定的 Java API(JAXB)	267
5.4.1	用于配置对象的样例XML文档	269
5.4.2	使用XML模式定义XML格式	270
5.4.3	JAXB API关键类	278
5.4.4	编组和解组XML数据	278
5.4.5	使用JAXB生成的类创建新的XML内容	280
5.4.6	在应用程序中使用JAXB生成的类	281
5.4.7	为使用JAXB注释现有Java类	287
5.4.8	使用JAXB的时机	306
5.4.9	JDK中JAXB的用途	306
5.5	小结	307
第 6 章	使用数据库持久保存应用程序	308
6.1	JDBC API 概述	308
6.2	设置环境	309
6.3	JDBC API 的用法	310
6.3.1	理解两层模型	310
6.3.2	理解三层模型	310
6.4	有效使用 JDBC 4.0	311
6.4.1	概述	312
6.4.2	管理连接	312