

SAMS

[美] Anil Hemrajani 著
韩坤 徐琦 译

Java敏捷开发

——使用Spring、Hibernate
和Eclipse



 人民邮电出版社
POSTS & TELECOM PRESS

Java敏捷开发

——使用 Spring、Hibernate 和 Eclipse

[美] Anil Hemrajani 著
韩坤 徐琦 译

人民邮电出版社
北京

图书在版编目 (CIP) 数据

Java 敏捷开发: 使用 Spring、Hibernate 和 Eclipse / (美) 赫姆瑞贾尼 (Hemrajani, A.) 著; 韩坤, 徐琦译. —北京: 人民邮电出版社, 2007.1

ISBN 978-7-115-15428-6

I. J... II. ①赫...②韩...③徐... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 125588 号

版 权 声 明

Anil Hemrajani: Agile Java Development with Spring, Hibernate and Eclipse

ISBN: 0-672-32896-8

Copyright © 2006 by Sams Publishing.

Authorized translation from the English language edition published by Sams.

All rights reserved.

本书中文简体字版由美国 Sams 出版公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

Java 敏捷开发——使用 Spring、Hibernate 和 Eclipse

◆ 著 [美] Anil Hemrajani

译 韩 坤 徐 琦

责任编辑 李 际

◆ 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: 20.25

字数: 386 千字

2007 年 1 月第 1 版

印数: 1 - 5 000 册

2007 年 1 月北京第 1 次印刷

著作权合同登记号 图字: 01-2006-4179 号

ISBN 978-7-115-15428-6/TP · 5782

定价: 39.00 元

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

内容提要

本书介绍了一种将极限编程 (XP)、敏捷模型驱动、重构等方法相结合的敏捷软件开发过程和软件开发需要的 3 种重要的技术。全书紧密地结合一个实际范例程序的开发,展示了敏捷开发的完整生命周期。除此之外,本书还引入了很多先进的技术和工具,便于读者进一步的学习和使用。本书中介绍的技术和方法对进行高效的、敏捷的 Java 企业级应用程序开发有极大的帮助。

本书深入浅出,适合于广泛的软件开发的相关群体,无论是具备了一定 Java 基础知识的学习者、具有一定经验的开发者、高水平的软件架构师,还是软件项目技术负责人或项目经理,都可以从本书中受益。

序 言

序言的真实目的是为了说服人们去买这本书。所以，我猜你现在有 3 个选择。第一，你会节约时间现在不读这篇序言，只是接受我的建议，即买这本书是一种很正确的投资。第二，你会不相信我说的话，坦率地说这样的做法是很明智的（因为我是一个顾问），但是在你继续读这篇序言时，我所说的一些话使你产生了共鸣，促使你买下了这本书。第三，你决定不买这本书，这的确不是一个好的主意，因为我认为我所见到的绝大多数 Java 程序员都能从这本书中真正受益。

在我看来，Java 社区中真正需要的是另外一种介绍那些“很酷”的 Java 技术的书。现已有数百种图书、数千篇杂志文章和数万个网页在介绍这些“很酷”的 Java 技术，但幸运的是，那些内容都不是这本书所要介绍的。本书是为数不多的能教你终生受用的技能的好书之一。的确，本书包括 Spring、Hibernate、Ant 还有其他很多你取得成功所需要的技术的基础知识。但是，更重要的是，它能在技术的基础上介绍敏捷开发，而这种敏捷开发是来源于极限编程（XP）和敏捷建模（AM）的，它会使你在现代软件开发中获得成功。

大部分程序员听说过极限编程，很多已经采用了一些极限编程的技术，比如测试驱动的设计（TDD）、重构，甚至结对编程。这些都是很好的开始，但是还不够。在这本书中，Anil 总结了他多年的经验，描述了如何将它们运用于实际的项目开发中。其他的书常常只能提供理论性的内容。但是我们都知，理论和实际常常是不一样的。

当 Anil 第一次请我担任本书的技术评审时，本书给我留下最深刻的印象是它简单和高效的 Java 项目的建模方式。事实上，你可能想现在就快速翻阅这本书，快速浏览其中的一些模型。我想你会发现书中的图例与你在实际项目中自己开发时所使用的很相似——相比现在很多的建模书籍中推荐的方式，它给人一种耳目一新的变化。你还会注意到 Anil 是如何介绍将这些模型转化为日常编写的复杂代码的。我认为这正表现了本书具有强大的力量：它提供了符合实际的建议，而这正是顶级程序员在现实中所做的。

本书还教授如何完成那些我们需要做的适合软件开发全景的一般性的工作，例如，验收测试、单元测试、对象关系映射、系统集成以及重构。本书从“5000 英尺”的过程观点开始，然后逐渐降低到地面，深入浅出地介绍在实际工作中怎样使用工具。大多数的书只注重问题的一方面，但是本书做得非常好，兼顾到两个方面。如果你花费一些时间看完本书，我想你会理解我所说的这些话。

——Scott W. Ambler
Agile Modeling 主管

这本书很难定位。就让我来解释一下为什么它不同寻常，为什么值得你去关注。

容易定位的书在业界举不胜举。它们往往是那些介绍一种专门的产品或 API 的书籍。有的书很好，但有的很差。你可以通过封面、字体、出版社、作者的可信度来选择要看的书——你已经做出了一个非常重要的选择：你知道什么是你想要的书。这样一本好的书籍能使你在某个领域更具实力，但是，它并不能改变你的工作方式。

不易定位的书就少得多。它们与作者的关联更加紧密，潜在地，与你的关联也就更多。

这本书告诉了我们，这样的书是多么有价值。Anil Hemrajani 提炼出他作为架构师和开发者的广泛的经验，写成了这本充满了一个成功的从业者的实际观点的书。和所有最好的书一样，它是一种读者和作者的有效沟通。具备了这样的有效沟通，书中的主题就不局限于可预测的范围，使得内容非常生动：Anil 采用了非常自然的写作方式，使读者能从阅读中得到快乐。

这本书所涉猎的范围非常广。它将一些原本很少放在一起介绍的主题融合在了一起，但这样做是非常必要的。我们所使用的软件开发过程不可避免地与我们构造代码的方式和编写代码的工具联系在一起。没有一个高效的开发者在工作中会将它们彼此分割——很多决定之间是相互联系的——但是绝大多数的书籍并没有为我们描述这样一个完成任务所需要的蓝图。

现在，如果你想开发出好的 Java 软件产品，就需要理解一些关键的概念，例如对象关系映射、依赖注入，你还需要了解为什么使用和怎样使用单元测试和自动构建项目这样的技术，并且——同样重要的——你要知道完成任务所需的最好的工具，例如框架和集成开发环境（IDE）。你还需要知道，应避免哪些隐藏的陷阱，以及为什么有时使用软件的技巧和对技术的掌握一样重要。

Anil 所做的工作之所以卓越，是因为他将这些东西一起写入了本书，为很多人感到迷惑的地方提供了清晰的解答。本书介绍的范围广阔，但却从未偏离目标——帮助读者完成成功的项目。

在我看来，本书是一幅蓝图。它为你清晰地描绘了通往 Java 企业级开发的成功之路。沿途中，你可以为图中的某个领域增加更加细致的描绘，但你将永远从图中所提供的方向中获益。

我向今天所有想成为 Java 企业级开发实践家的人推荐这本书。它可能会更好地改变你的工作方式。

——Rod Johnson
Interface21 公司 CEO
Spring 框架的创建人

关于作者

Anil Hemrajani 从 1995 年底开始，作为开发者、创业者、作者和培训师一直致力于 Java 技术的研究。他是 Isavix 公司（现在的 Inscope Solutions）的创始人，该公司是一个很成功的 IT 服务公司；他也是 isavix.net（现在的 DeveloperHub.com）的创始人，这是一个获奖的在线开发者社区，现注册用户超过 10 万人。他有超过 20 年服务于信息技术社区的经历，其间与很多财富 100 强企业或小企业合作过，他在业界知名的杂志上发表过大量文章，在很多会议中发表过演讲，被 Sun Microsystems 公司授予“Java 社区发展杰出贡献奖”，并在 JavaOne 上获 BackOnline（一种基于 Java 的客户端/服务器的产品）的“优秀 Java 客户端产品”奖，还以一个免费的在线文件存储服务获得 Computerworld-Smithsonian 奖的提名。他最近的一个项目是 visualpatterns.com 网站。



本书献给我可爱的、细致入微的妻子，她总是耐心地支持我（容忍我的一些古怪的方面）。谢谢你在我忙碌于各种项目（比如这本书）的时候为我照料好一切。没有你就不可能有这本书！当然还有我最亲爱的孩子们，是你们照亮了我的每一天。



致 谢

虽然我写完了这本书，但是我知道，如果没有那么多卓越的人的支持，这绝对是不可能的。我从内心深处真诚地感谢列在下面的每一个人。

- **Cliff Berg**——首先，也是最重要的，我要感谢 Cliff，我的老朋友和老同事，是你一直鼓励我写这本书，验证并支持我的想法，给了我很大的信心。
- **Scott W. Ambler**——谢谢你同意和我一起写本书的序言，并对它一章一章地做了细致入微的审阅。Scott 已经成为了这个领域里我的一个偶像，他致力于敏捷建模和敏捷数据方面的工作，至今我还对他能在百忙中答应帮助我感到惊喜。我还要感谢 Scott 的 www.agilemodeling.com 和 www.agiledate.org 网站，它们对于我的帮助是无价的。Scott，谢谢你。
- **Rod Johnson**——当我第一次遇见 Rod，我还不知道我将要写这本书，更不知道会邀请他为我写序言——我只知道我们的观点相同，我对他所做的工作表示敬佩。如今 Spring 框架如此受欢迎，我惊讶于他竟然有时间睡觉，所以他能为我的书写序言对于我来说尤其意义重大（谢谢你）。我还想感谢他整合出了 Spring 框架，我非常喜欢使用它。
- **Anil Singh**——我情不自尽地要感谢你的帮助，那些漫长的写书的日子，那些无价的反馈意见，那一个个我们一起讨论章节内容的深夜（时常是凌晨3点）——谢谢你为我做的一切！我尤其感激你几乎可以在任何时候来和我讨论我的书。然而，我所能回报你的只是一句由衷的话：谢谢你！
- **Dan Shellman**——他的快速、细致、真诚的反馈使得这本书变得好了很多，谢谢你在写书的过程中宽容地接收了我的 100 多封邮件。尤其感激你的反馈和电话交谈，无论是在周末还是在与家人的假期中！Dan 已经成为了我的长期的同事和老朋友，我希望我们的友谊能够长久保持下去。
- **Haresh Lala**——谢谢他的细致的、长期不懈的反馈！感谢他对我书中代码的测试（两次），并且阅读了我最早的、最粗糙的书稿。最重要的，感谢当他的生活中出现了新的生命的时候，在那样的忙碌中对我的不懈帮助。
- **Hernando Vera**——幸亏有了 Hernando 的敏锐的思考、提出的想法和对技术的巧

妙阐述。否则，我真是寸步难行。10年来，Hernando已经成为了我求助的对象，当我怀疑的时候，我知道可以从他那里获得答案，无论是在开发、设计、搭建架构、构建等哪个过程中，我想我还没有见过有谁能像他一样，集智慧、创造性于一身。

- **Martin Rimmelzwaal**——Martin 和我是最近认识的，但是我已经把他当做了我最亲密的朋友。谢谢他对本书前几章所做的评审。我尤其要感谢他对于我的那些没完没了的关于各种技术观点、方法和相关问题的邮件的回复。我希望以后还能和 Martin 在各种项目中合作。
- **Spring** 框架的团队——首先，你们非常厉害！对于具体的人，我想感谢 Alef Arendsen 对本书第 6 章和第 7 章的评论，还有他在 Spring 的各个方面给予我的帮助，正是在他的帮助下，第 6 章和第 7 章得到了很大的改善。我还要感谢 Juergen Hoeller 对于声明性事务管理和后面章节中的接口问题的帮助。考虑到 Spring 团队的白天（帮助客户）和晚上（架构方面的工作）的忙碌工作，同时还要不可思议地发布新的版本——我能说的只有：各位，谢谢！
- **Madhu Siddalingaiah**——感谢他对本书大纲的起草的指导（还有其他一些出版方面的问题）以及对于本书第 8 章的宝贵的反馈意见。
- **Dave Berman**——感谢 Dave 对于本书第 2 章的深刻的评论和他关于敏捷方法的讨论，使得本书的敏捷/XP 方面的阐述更加坚固细致。
- **Jeff Nielsen**——感谢他对本书第 2 章和第 3 章以及 XP+AMDD 部分滑稽的图片系列的及时的反馈意见，使我能在时间有限的情况下解决很多主要的错误。谢谢 Jeff！
- **Ramanand Singh**——谢谢你和我进行的关于 Spring 的最初讨论，还有你对第 6 章的反馈意见。
- **Pearson** 出版社的工作人员——我要感谢 Pearson 出版社的职员（Songlin、Mandie、Kim、Mark、Barbara 和其他幕后人员）出版了这本书，尤其感谢 Jenny 自始至终的参与，她帮助我渡过了平凡的和令我沮丧的日子。我还要感谢 Boris，他给了我对于本书深入的、直接的、有价值的评价——如果没有他的帮助，本书不可能是现在的样子。
- 对于我的朋友 Peter，感谢你时常在思想上对我进行鼓励，这让我能够以一种独特的方式来完成这本书。还有 Andy 和 Missy，和你们在一起的时候充满了欢声笑语，使我愉快地度过了那段时间（尤其是在持续几周、每天写书 14~15 小时的情况下）。
- 感谢 Greenberry 咖啡店的员工们，他们为我提供了舒适的环境、高速的网络和美味的咖啡和食物。这使我能在一天中很长时间投入地写书。

- 最后，但是同样重要，这本书的成功完成是建立在业界很多人的创新性的工作之上的，可以说他们已经成为了我的偶像，我要感谢他们，因为是他们宝贵而惊人的想法间接地帮助了我。这些人包括：Martin Fowler、Kent Beck、Eric Gamma、Ward Cunningham 等人。

为读者提供的服务

访问 Sams 出版公司的网站 www.sampublishing.com，根据本书英文版书号(0672328968)搜索本书网页，这样你就可以方便地下载本书中使用的源代码及示例文件。

前言

我从 1995 年底开始从事 Java 技术方面的工作，也就是在 Java 开发包（JDK）1.0 正式发布之前不久。在这之前，我已经使用 C 和 C++ 编程很多年了。我很喜欢 Java 所提供的特性，例如跨平台性、简单的语法（相比于 C 和 C++）、面向对象特性、安全性、丰富的 API 等等。

在 20 年的职业生涯中，我学到了很多。其中我认为最重要的是简单性。每当看到复杂的东西时，我就会怀疑它是否是正确的解决方法。而这种复杂性正是在大约 2000 年 J2EE 开始成为主流的时候我对 Java 的感受。注意，从现在开始，我所说的 JEE 是指 J2EE，因为其中的“2”最近被 Sun Microsystems 公司去掉了。

当看到抽象层给 JEE 带来了不必要的复杂性的时候，我渐渐地失去了对 Java 的兴趣。我开始认为 Sun Microsystems 公司（Java 的创造者）将 Java 和 JEE 致力于解决很复杂的企业级的应用，而在一定程度上忽略了不太复杂的、中小型的应用程序。甚至我看到了人们将这种小题大做的方式当作了常识来接受，我曾经在非分布式的处理中使用企业级的 JavaBean（EJB）的方式运行，例如本地登录。我对此的反对意见相当大，以至于 2000 年在 JavaWorld.com 上（<http://www.javaworld.com/javaworld/jw-10-2000/jw-1006-soapbox.html>）写了一篇短文，题为“你真的需要企业级的 JavaBean 吗？”（大约 5 年以后，我看到 EJB 3.0 规范被编写得简单了许多，更加简化了开发。）这个故事把话题引到本书和我写本书的原因上。

我最近被一个财富 100 强公司聘请为顾问，建立一个在集群的环境中运行的企业级应用程序。通过在网上对各种解决方案的调研和与一些精通技术的人的讨论，我决定选择这样一种解决方案：它采用 Spring 的 MVC 框架、Hibernate 的对象—关系（OR）映射框架、Eclipse IDE、JUnit 测试框架、Ant 构建工具、多种标签库，还有一些其他的工具。（所有这些产品以及我选择这些技术的原因都将在本书后面的章节中介绍。）

我非常喜欢使用 Spring 和 Hibernate，主要因为它们能让我使用 POJO（Plain-Old Java Object，普通 Java 对象）并且避免了很多 EJB 所带来的不便。当然，使用 Eclipse IDE 也是相当不错的。我越来越惊喜地发现这个产品是如此的好，这就是我决定使用本书的一整章来介绍 Eclipse 的原因。我认为当 Java 受欢迎的地位逐渐被微软的 .NET、LAMP

(Linux、Apache、MySQL 和 PHP 或 Python/PEAL) 和 Ruby on Rails 所威胁的时候, 前面说到的这些产品的出现给 Java 注入了新的生命力。

本书中, 尽管 Spring、Hibernate 和 Eclipse 的地位是非常重要的, 但能从技术和软件过程的角度提供一整套的解决方案是本书最重要的目标之一。从技术的角度出发, 我提供了一个端到端的解决方案来实现一个完整的 Web 应用范例程序, 它可以使用事务管理并且适用于集成环境。从过程的角度出发, 我近来交替使用 Rational 提供的统一过程 (RUP, Rational Unified Process) 和敏捷模型驱动开发 (AMDD: agilemodeling.com) 与极限编程 (XP: extremeprogramming.org) 相结合的过程方法。所以在本书中你将会发现如下的概念和文档, 例如用户故事、发布计划、CRC 卡片等。目的是为你提供一个快速开发和部署企业级应用的解决方案。

还要再对我的背景做一点说明, 我作为程序员已经大约 20 年的时间了, 主要使用 C/C++、Java、企业相关的数据库、应用服务器、UNIX、微软的 Windows 等核心技术。然而在 1998 年的时候, 我走了大约 5 年的弯路, 这段时间我在经营自己创立于 1996 年的公司 (我做着最少量的编程工作)。后来我卖掉了这个公司, 重新回到了我的开发工作中。即使是我在做公司 CEO 的时候, 手下有几个为我工作的人, 我仍然抓住一切机会在 7 年的时间里约见了 100 多个开发者, 与他们一起讨论技术。离开了公司, 我又创建了一个在线的 Java 开发者社区, 它的会员数量超过了 10 万人, 为此赢得了几个奖项。我希望我的经历能够给本书加入独特的视角。

总之, 真心希望你觉得本书有用并喜欢它。

谁该读这本书

本书认为你已经具备了 Java 和相关数据库 (包括 SQL) 的基础知识, 并且具有命令行执行程序的经验, 除此以外, 下列读者可以使用本书。

- 软件开发人员和架构师——开发人员和架构师阅读本书都可以有所收获, 因为本书包括了高水平的软件开发流程、程序的设计和深入而完整的对 Java 技术的研究, 还包括一个具有很好的功能的企业级应用程序的范例。
- 技术负责人和经理——具有一定编程经历的, 愿意使用 Java 或类似语言的并对多种使用技术的项目的构建有深入理解的负责人和项目经理, 本书介绍的知识能够帮助你的项目进行计划并解决项目上的技术难题 (也许会在你受挫的时候给你精神上的支持)。你也可以选择具体的一章 (例如, 第 5 章), 理解如何将具体的技术融入到你的项目中去。

除此之外, 作为读者, 你可能会从本书中得到构建强大的 JEE 企业级应用的其他好的方式。如果你不熟悉敏捷建模或极限编程, 或正在寻找敏捷软件开发的方法, 本书会

使你对迭代渐进式的软件开发过程有个全面的了解。

本书的目标

以下是本书的目标。

- **敏捷开发**——本书最初的也是最重要的目标是要告诉你怎样快速开发企业级的 Java 应用程序。完成这样的目标，需要做好很多方面的工作。要有一个敏捷小巧的软件开发流程，要有简单的设计（适度运用设计模式和抽象层），使用方便的技术（例如 Spring 和 Hibernate），使用 POJO 而不是远程对象，一般来说尽可能地使用稳定的开源技术。总之，要能使用 Java 简单快速地开发企业级的应用程序。
- **完整的解决方案**——紧密相联的还有本书的第二个目标，就是从技术和开发流程的角度提供一套完整的解决方案。读完本书后你应该可以建立一个完整的应用程序，不仅仅是从技术上，还能使用本书介绍的开发流程。除此以外，有些技术我无法深入介绍，但会提供进一步研究它的参考资源（网站）。本书在所介绍的技术上比较好的一点是你会有一个完整的系统，从用户界面到嵌入的数据库，使用任务调度功能（感谢 Spring 框架），所有的这些都包括在一个单独的 Web 应用文件（.war）中！当然你可以把这里所介绍的技术改换为其他你选择的技术（例如，选用 Oracle 数据库而不使用 HSQLDB）。总之，你会有完整的解决方案来做到这点——从流程上和技术上。
- **仅使用一种开源技术的解决方案并不是本书的目标**——尽管本书完全基于开源的框架、工具和产品，但介绍仅使用一种开源的解决方案并不是本书的目标。例如，你可以利用 Java 的产品兼容性用商业产品替换这里介绍的任何产品。然而开源项目已经变得越来越成功，技术的健壮性和详细的文档给我留下了非常深刻的印象。例如 Eclipse SDK 和 Hibernate 就比它们那些很有争议性的商业对手做得好。你完全可以只使用本书的所有技术作为你的解决方案，它们会像介绍的那样出色。事实上，最近我为一个大公司提出了一个解决方案，就是使用 Spring 框架、Hibernate、Eclipse、JUnit、Ant 和本书中所提到的其他工具。我们也使用了像 BEA WebLogic、Oracle 数据库服务器这样的商业产品。这个公司（还有其他我知道的）就是采用基于本书中提到的开源技术的解决方案。
- **快速阅读**——本书特意写得比你通常见到的 600 多页的 Java 书薄很多。这使你能很快地阅读本书并将解决办法运用于实际中去。我努力将本书要介绍的内容设计为流线型，每一节点都还包括很多内容。为了不把本书写成一本非常厚的书，我需要做一些艰难的选择，放弃对一些内容的介绍。但我努力将所有你需要的与过程、技术相关的 Java 敏捷开发内容都包括了进来（如前所述的一套完

整的解决方案)。

- 简单性——我会尽一切可能采用简单的方案来实现那些用复杂方式来实现的目标。例如，本书的范例程序就采用了最少的抽象层次来实现我们的目标。对于抽象层次，这里我指的是过度地使用设计模式、接口和对程序过多的分割。其中的每一个都是很有意义的，但是对每种东西适度地使用才是好的实践，也是我在软件开发的过程中喜欢遵循的。除此之外，我还认为这种简单性同样应该扩展到设计中，在这个过程中我愿意适度地使用 UML，但是为了简单化，常常使用像 OpenOffice.org、PowerPoint、Visio 这样的工具做出一些形式灵活的视图，而不用像 Rational Rose 那样笨重的工具画图。
- 富于技巧——正如你可能已经知道的，当使用工具时，各种技巧不仅能使你更有效地使用工具，而且增加了你使用时的乐趣。本书中介绍了很多使用技术的技巧。附录中也包括了一些有用的内容，比如范例开发过程一览表和一些有用工具列表。
- 可选择性——贯穿本书（虽然没有详细介绍），我都会尝试提供一些推荐的解决方案。我知道一种解决方案并不能适用于所有的人。例如，你可能会使用 Sun 公司的 NetBeans 或 JetBrains 的 IntelliJ 作为开发工具，并不想换用 Eclipse。这种情况是被预料到的并且是可以理解的。本书在内容结构上考虑到了这样的情况，你仍然可以从本书的剩余部分获得宝贵的信息。将本书中介绍的技术替换为你自己的选择（例如，JDO，而不是 Hibernate）。

未介绍的内容

本书认为你已经具备了 Java 的相关知识并且对 JEE 有了较好的理解，认为你对软件开发流程、相关数据库的知识、多层架构等知识也有一定的认识，从这点出发，本书直接就开始讨论构建范例程序所需要的具体技术。而且我提供了相关技术安装（和它的高级特性）的说明网站，而不是把它们照搬到本书中，以免信息过时。

本书认为你对所介绍的主要技术并不了解，例如 Spring、Hibernate、Eclipse 等。所以，本书提供了这些技术的基础知识，以便你能将它们结合起来使用。本书也会进一步对这些技术提供的高级特性进行简单的介绍。这里没有提到的内容均不属于本书要介绍的范围。因为有很多书籍对本书所提到的技术进行了全面的阐述。

本书的内容

本书的内容更侧重于与开发相关的问题，而对基础环境的构建关注得较少。换句话

说，我更关注于介绍应用程序开发的技术，例如 Spring、Hibernate 和 Eclipse，而对像应用服务器（例如 JBoss）或数据库（例如 MySQL）方面的内容介绍得很少。在不确定的情况下，我会采用容易安装的产品。我坚定地认为首先应该实现程序中的功能，在这之后才会考虑选择什么样的重构方式和实现技术对它进行优化。本书的内容已经在实际的项目中成功地运行过了（其中一些是在集群的应用服务器中运行的）。所以你不要认为我们完全忽略了基础环境。本书的一个很主要的目的就是要保持它的短小精悍、恰如其分。所以我们紧紧地围绕着一个设计优秀、大小适当的应用程序讨论。

考虑到 Java 的跨操作系统性和产品兼容性，理论上，如果你的应用程序是可以发布的话，那么你可以将它部署到一个更加强大的 Web 应用服务器和数据库上。比如，你可以使用本书中使用的较低档的环境（Apache 的 Tomcat 和 HSQLDB），然后把它们升级到 JBoss 应用服务器和 MySQL 数据库中，或者进一步升级到 BEA 的 WebLogic 和 Oracle 服务器上。这就是 Java 的可爱之处：不仅跨越操作系统，而且使用的产品是兼容的。

还有一点要对本书中介绍的核心技术（Spring、Hibernate 和 Eclipse）说明的是，尽管它们最近被我使用，在我的书中着重介绍，但是我也提到了其他的一些与之并列的可选技术，建议你对它们也能了解一下。如果你使用 JDO 而不是 Hibernate，你仍然可以从本书的各章中获得想要的知识。

本书的组织方式

本书中的每一章都是建立在前一章的基础之上的。但因为各章在内容上是相互分离的，所以你也可以直接跳到某章（例如第 6 章），阅读想要了解的内容。如果对所介绍的内容不感兴趣的话，你也可以跳过这一部分（例如你可能想要使用 NetBeans 而不是 Eclipse，你就可以跳过第 8 章）。

第 1 章预览和概述本书将要用到的技术和开发流程。第 2 章主要定义了范例程序的业务需求，也概览了 AMDD 和 XP 的软件开发方法。第 3 章介绍了范例程序的设计。第 4 章介绍环境搭建的问题。从这一章之后，我们就开始进入了快速的代码开发阶段，我们会在第 5 章描写 Hibernate 编程。第 6 章和第 7 章主要介绍 Spring 框架。第 7 章和第 8 章是本书最精彩的部分，因为在这两章中用到了以前各章中的知识，你会明白我们为什么要以这种方式来介绍前面的章节，你会知道在这些技术背后到底是怎么回事，从而为 Spring、Hibernate、Ant 和 JUnit 等技术打下坚实的基础。从这部分以后，我们开始介绍一些高级概念，之后在本书的附录中还会有很多有用的东西。

还有，对于使用命令行开发而不是使用用户图形界面（例如 Eclipse SDK）的开发者，我在前面的章节中有意使用了命令行，以便让读者对怎样使用 Java 工具有一个基本的了解。以后当你在 Eclipse 这样的 IDE 中使用 Java 工具（例如 Ant 和 JUnit）时，就会明白

界面的背后到底正在进行什么样的工作。当你使用的 IDE 不能满足需要的时候，这尤为重要。

关于本书中的代码

本书是关于 Java 编程的，会有很多与范例程序相关的代码作为实际的例子要在本书中构建。所有范例的源代码和文件都可以从 Sams 出版社的网站中下载得到，这些电子版的代码会方便你自己浏览和测试。

源码中的一部分将会在本书各部分中引用和说明。除此以外，我们的代码中有两个约定：

- 为了便于阅读，在某些地方，将很长的单行代码（即多于 82 个字符）分隔为两行。
- 代码本身是以特定的字体显示的。

注意：

本书的所有代码都可以在 Sams 出版社的网站下载得到。请访问 www.sampublishing.com，根据本书英文版书号（0672328968）探索本书网页，点击“Downloads”下载。

关于本书的插图

一幅好的图比使用很多文字更能说明问题，所以在本书中你将会发现大量的界面截图和示意图。对于本书中的 UML 图，我更倾向于使用形式自由的图，而不是那些使用转换工具从工程中转换而来的。我也没有花费很多的时间来处理很正式的 UML 图，因为很多图在起到了它们的作用后就可以扔掉，大多数的项目中不会在之后更新它们。我坚信程序的源码和数据库脚本才是最重要的工件（由软件开发过程所生成或使用的文档、模型、说明或软件）。

总结一下，我觉得示意图应该做到以下几点：

- 它必须是有效的；
- 本身具有很强的说明问题的能力（例如，使用说明性的语言而不是晦涩难懂的符号，还要使用图例）；
- 很简单，恰如其分，让人一看就知道要传达的信息；
- 与标准的表示符号一致（例如 UML）或是使用组织要求的形式，这样可以很容易地将一个系统转移给别人或团体，也很容易生成代码，而且，阅读的人更愿意使用标准的表示法。