

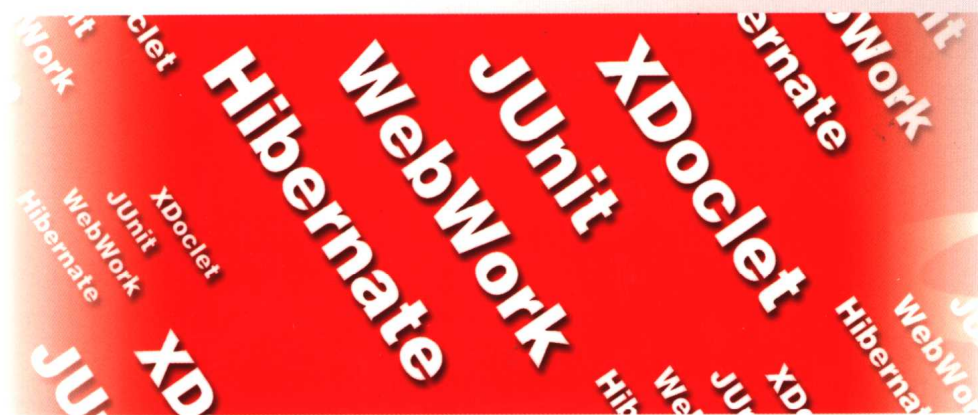
Java

开放源码编程

Java Open Source Programming

[美] Joe Walnes Ara Abrahamian
Mike Cannon-Brookes Pat Lightbody 著

甄山 甄镭 译



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

内 容 简 介

本书介绍了如何通过使用开放源码工具来构建一个现实的应用,以提高开发效率,降低企业开发费用。

全书共分三个部分,使基本概念、使用工具和开发应用三个过程贯穿于读者的学习当中。特别是本书提及的开发者工具箱,包含了诸多的开放源码项目,它们都是与 J2EE 长期协同工作的结晶。而且,书中的实例通过重新创作著名的 Sun Java 蓝图——PetStore 应用,将 JUnit、Ant、Hibernate、WebWork、SiteMesh、XDoclet 以及 Lucene 等工具的应用演绎得淋漓尽致,有利于读者在开发过程中提高可用性、结构简洁性和复用性的能力。

本书适用于希望学习 Java 开放源码技术的程序员阅读。

0-471-46362-0 Java Open Source Programming: with XDoclet, JUnit, WebWork, Hibernate by Joseph Walnes, Ara Abrahamian, Mike Cannon-Brookes, Patrick Lightbody

© 2004 by Wiley Publishing, Inc.

All rights reserved. Authorized translation from the English language edition published by John Wiley & Sons, Inc.

本书简体中文专有翻译出版权由 John Wiley & Sons Inc. 授予电子工业出版社未经许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字: 01-2004-0920

图书在版编目(CIP)数据

Java 开放源码编程 / (美) 沃纳斯 (Walnes, J.) 等著; 甄山, 甄镭译. —北京: 电子工业出版社, 2005.1
(Java 技术大系)

ISBN 7-121-00514-X

I. J… II. ①沃… ②甄… ③甄… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 123403 号

责任编辑: 张 毅 zhangyi@phei.com.cn

印 刷: 北京东光印刷厂

出版发行: 电子工业出版社

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: 22.5 字数: 510 千字

印 次: 2005 年 1 月第 1 次印刷

印 数: 5000 册 定价: 35.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077。质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

译者序

应用软件的目的是解决某一领域的业务问题。然而，在进行应用软件开发时，程序员不得不一方面关注用户的业务需求，一方面关注要解决的技术问题，他们经常发现，用户需求的一点变更，就会导致系统结构性的变化。比如，在一个基于 Web 的应用程序中，用户突然想要增加全文检索功能，而你对全文检索又一无所知，更糟糕的是用户不认为这是一个多么了不起的功能，你会怎么办呢？如果应用系统的每一部分开发都是从零开始的话，项目的风险是极大的，不仅仅是时间和费用的问题，潜在的 Bug 经常会导致系统出现不可预知的问题，甚至是崩溃。

开放源码使开发人员从底层功能开发中解放出来，可以更好地关注于用户的业务和需求。由于有了经过充分测试的代码，系统的成本降低了，周期缩短了，风险减少了。由于总体成本的降低，用户更愿意加大在信息系统上的投入，使信息系统的应用范围更为广泛，这就为开发商提供了更多的商业机会。

然而，如何在实际应用中应用开放源码项目仍然存在问题，这些问题包括如何将这项目与开发过程相结合？如何选择合适的开放源码项目？如何进行单元测试与集成测试？如何更有效地实现程序员之间的交流？幸好，针对这些问题在本书中都可以找到答案。

本书是由 Java 开放源码领域的顶尖高手编写的，展示了如何使用最好的开放源码工具来开发 J2EE 应用，这些工具的组合使用可以减少实际应用中的 Bug，并可以提高整个系统的稳定性和健壮性。本书用一个基于 Web 的宠物商店——PetSoar 的开发作为实例，展示了整个项目的开发过程，在这个过程中，作者更关注于开发易于维护和灵活的应用，而不是仅仅展示最终结果。同时，作者为在开发过程的各个阶段应用开放源码的软件构件提供了一套方法。

本书内容涉及到以下开放源码产品：

- Hibernate——超高性能的对象/关系持久性和查询服务
- OpenSymphony WebWork——开放源码的 Web 应用框架
- JUnit 和 Mock Objects——辅助进行快速、健壮的单元测试
- XDoclet——辅助生成代码和自动配置文件
- Jakarta Lucene——为应用增加类似 Google 的针对数据仓库的智能搜索功能
- OpenSymphony SiteMesh——辅助创建有共同视觉效果和用户体验的站点
- OpenSymphony OSCache——方便地缓存 Web 站点的内容，加快页面的调用速度

由于译者水平有限，加之时间仓促，书中错误和不妥之处在所难免，希望读者多多批评指正。

译者

感 谢

尽管 Martin Fowler 有自己的著作要编写,但还是自始至终地为本书提出了非常宝贵的改进意见和指导。Ben Hogan 和 Erik Hatcher 给我们提供了大量的反馈信息。在这里要特别感谢的是 Dan North,他不仅为本书的校验做了大量的工作,同时还负责撰写本书的其中一章,为最后定稿节省了大量的时间。特别感谢 Gavin King 在本书的最后编写阶段准确而高效地完成了与 Hibernate 有关章节的校验工作。

此外,本书还得到了许多书评作者耐心的帮助,没有他们的帮助,本书将一事无成。他们是:

Andy Pols, Aslak Hellesoy, Damian Guy, Darren Hobbs, Hooman Mehr, Ivan Moore, Jason Carriera, Jeremy Stell-Smith, Lisa Van Gelder, Mathias Bogaert, Matt Ho, Mike Roberts, Mike Royle, Owen Rogers, Rachel Davies, Rachel McConnell, Rebecca Parsons, Scott Farquar, Steve Freeman 和 Tim Mackinnon。

在本书的创作过程中, Rickard Oberg 和 Matt Baldree 也发挥了重要的作用。没有他们的帮助,我们甚至不知从何入手。

同时也感谢我们所使用的开放源码软件工具组成员的支持,他们应作者的要求为本书提供了优秀的技术软件。我们感谢 Bob 和 Wiley 的其他小组提供的技术支持。

Mike 也感谢他的合作者, Atlassian 开发组成员 (Scott, Owen, Anton, Jeff, Bobby, Dave, Edwin) 的合作支持;感谢他的“工作”同事 (Mike, Gavin, Jason, Jeremy, Eugene, Rickard 及其他同事的支持)、他在悉尼的同事 (Niki, Alina (Kins), Sarah, Camilla, Will, Hoey, Tex, Kel, Nuts 等能够想起的同事)。最后,也是最重要的,感谢他的家人 (妈妈、爸爸、Jace, Tora, Andrew, James, Holly 和小 Phoebe) 的支持。他也感谢那些在初稿的编写过程中给予他帮助的人们,由于忘记了他们的名字,在这里请求谅解并表示深深的歉意。

Joe 感谢他所有 ThoughtWorkers 的工作伙伴,同他们在一起工作使他受益匪浅。特别感谢 Dan North 对他的支持和鼓励,使他脚踏实地并不断前进。Paul Hammant 经常来看望他,并把自己在实际工作中获得的经验提供给他。Paul, Dan, Laura Waite 和 Tim Bacon 经常在 Joe 遇到困难的时候及时帮助他解决疑难问题,他们是非常好的良师益友。Charles Lowell, Chris Stevenson, Drew Noakes, Duncan Cragg, Gregor Hohpe, Jeremy Stell-Smith, Jonathan Rasmusson, Martin Fowler, Mike Roberts 和 Tim Mackinnon 一直在工作中相互支持和鼓励,还有其他人,在此不一一列举。感谢 Trevor Mather 的支持。

Joe 感谢他的家人 (妈妈、爸爸 Gay 和 Tilly) 的支持。最重要的是感谢他的未婚妻 Jaimie 自始至终对他工作的支持和帮助。爱情的力量是无限的!

Ara 感谢他的合作者,在本书的编写期间从他们身上学到了许多东西。他也同时感谢他在 Eedé 和 Roxcel 的同事 (Behrang, Nazanin, Iman, Arbi, Nassim, Ali, Ramin, Mohammad, Jarek 和 Mr. Lehmann) 对他工作的支持和帮助;特别对他的启蒙老板

(Hooman) 和 Arsanjani 博士 (他的首位软件开发技术启蒙教练) 以及他最好的朋友 (Teodik, Armond, Farzad, Raymond, Telma, Abtin, Artin, Arin, Nasser 和 Ali), 没有他们的大力支持和帮助, 完成这样一项规模浩大的工程是不可能的。感谢他的家人 (爸爸、妈妈、Razmik, Emma, Alice, Seda 和 Narbeh) 对他工作的支持。他爱你们!

Pat 感谢他在 Cisco 和 Spoke 的所有合作者, 在本书的编写过程中, 他们自始至终地给他提供帮助。特别感谢 Mike Schwartz 在提供大量有价值的信息反馈工作上所做的突出贡献。他同时也感谢 Adam Fleming, Marshal Dhillon, Michael Artamonov 和 Dan Libicki, 因为他们把在工作中获得的第一手资料交给他, 用在本书中。特别感谢 Bill Griswold 教授在本书的编写过程中给予的巨大帮助。最后, 感谢他的家人和朋友在本书的编写过程中所给予的支持。爸爸、妈妈、Chris, 你们是最优秀的。

作者简介

Joe Walnes 是 ThoughtWorks (www.thoughtworks.com) 的技术顾问, 这是一家专门为企业的技术发展提供服务的系统集成公司。Joe Walnes 的学术研究领域集中在为软件开发设计者提供极限编程指导, 为面向对象和面向组件系统提供设计技术支持, 以及简化 J2EE 的开发。在他的(很少的)业余时间中, 致力于开放源码项目的研究工作。他的许多工作项目和论文可在 www.opensymphony.com 和 www.codehaus.org 中找到。

你可以在 <http://joe.truemesh.com> 阅读他的 blog。

Ara Abrahamian 是一名为企业 Java 开发提供技术支持和服务的自由顾问。他参与了遍布全球的各种 J2EE 项目。他的技术专长包括代码生成、面向属性程序设计和软件自动化技术。同时他也积极参与许多开放源码项目的研究工作, 并且是 XDoclet 项目的领导人。他对采用轻量级的方法方面, 如大型分布式开发团队的极限编程 (XP) 也感兴趣。

你可在 http://freeroller.net/page/ara_e 找到有关他的介绍资料。

Mike Cannon-Brookes 是 Atlassian (www.atlassian.com) 公司的创始人, 该公司是一家 J2EE 应用和服务的公司, 位于澳大利亚的悉尼。该公司设计完成了如 JIRA 这样顶级的 J2EE 软件应用开发。JIRA 是一个用于问题追踪和项目管理的领先级系统。Mike 的特长在于构建非常简洁、便于使用的 J2EE Web 应用, 并建立了 JavaBlogs (www.javablogs.com) 网站, 该网站是一个基于 Web 的、关注 Java 的 blog 聚集器。同时他也建立了 OpenSymphony (www.opensymphony.com) 项目研究网站, 并从事 WebWork、SiteMesh 和其他开放源码项目的研究开发工作。

你可在 <http://blogs.atlassian.com/rebelutionary> 找到有关他的个人资料和研究资料。

Pat Lightbody 现居住于美国旧金山, 他在 Palo Alto 公司工作, 该公司最初叫 Spoke Software。这个公司主要开发采用社会网络提高销售水平的企业软件。在这之前, 他作为一名软件工程师在 Cisco Systems 从事研究工作, 并在圣迭哥的加利福尼亚大学就读, 获计算机理学学士学位。同时他也从事各类开放源码项目的研究, 主要从事 WebWork、XWork 和 OSWorkflow 的研究开发工作, 可参考 www.opensymphony.com。

你可在 <http://blogs.atlassian.com/psquad> 中找到有关他的介绍和相关研究资料。

引 言

在当今的 IT 领域, Java 在企业开发中是领先的技术。由于管理需要越来越多的技术, 复杂的结构似乎爆炸式地增长, 留下许多不能与当今世界快节奏同步发展的疑难问题。这些复杂性可以从 Java 2 企业版本 (J2EE) 规范快速更新看出来。这些不必要的复杂性驱使我们寻找简化程序开发的方法。

现在, 不要误解我们: 我们喜欢 J2EE! 但是为什么它要花费那么长的时间才能完成项目呢? 我们很早就是 J2EE 的使用者, 当它们发展的时候, 一直紧跟相关概念和复杂性。这是个棘手的问题, 随着时间的推移, 我们开始利用可复用的开放源码技术, 这明显减少了开发这些应用所需的时间。我们也开始研究哪些技术能实际帮助我们, 哪些已变成我们研究开发上的障碍。

这是一本关于使用开发技术和开放源码工具来降低构建企业应用费用的书。我们的目的是向你展示如何把这些技术应用组成一体来构建一个现实世界的应用, 同时避免了复杂的概念, 而使用简单易懂的理论。我们将帮助你扩大 Java 和 J2EE 框架方面的知识, 以便开始自由使用经过无数次开发和研究得来的成果。

本书不仅教你如何利用开放源码技术, 而且可以使你马上投入工作, 它也将加强和扩大你的研究理论, 像我们一样, 用这样一种方式——很快, 你会挠挠自己的头惊奇地问: “为什么我不早点做这个呢?”

为什么需要这本书

本书的两个目标使你无论是现在还是未来都将受益:

- 当前最近的目标是, 你会对众多令人惊奇的开放源码技术中的几个变得非常熟练。
- 从长远来看, 我们将介绍的简单技术和理论不论是现在还是未来都将适用, 即使现在的技术可能被较好的技术所代替, 也还是如此。

我们使用这些技术和方法来重新创作 Sun 的 Java 蓝图——著名的 PetStore 应用。我们的 PetStore 实现将不是一个精确的副本, 而是在各方面的提高, 包括可用性、结构和简洁性。我们把这个项目称为 PetSoar, 是因为项目刚开始翱翔。在这里只描述了贯穿本书的各种技术和技能的轮廓。

读者对象

正因为本书有两个目标——技术和方法, 本书相应地也适合两种类型的读者 (尽管通常这两组实际包含同样类型的人) 阅读。

第一种是任何希望学习 Java 开放源码技术的人。本书中, 我们将就几个重要的对企业开发有帮助的开放源码项目做了更深一步的讲解。这些技术包括 JUnit、Ant、Hibernate、WebWork、SiteMesh、XDoclet 和 Lucene。如果你对 these 项目不是很熟悉, 或是你已经

对这些技术有所熟悉而准备用它们解决问题的话，我们将用这些工具针对你所遇到的问题进行有针对性的讲解，并提供在现实世界中一步步地使用这些工具的指导方法。

第二种读者是那些已对项目的复杂性有了初步的认识而急于解决问题的人。我们已经发现这些复杂性实际上是开发者所思考的结果，即复杂的需求需要复杂的解决方案。取而代之的是，在本书中，我们将展示如何应用最简单的原则在最短的时间里来达到目的。我们将通过使用测试驱动开发（TDD）方法进行 PetSoar 应用的开发来达到这个目的，这个实践表达出这样一个观点：少即多（Less is More）。

本书约定

在本书中，我们针对不同的原因使用不同的书写格式。例如，当需要你输入某些东西的时候，我们把想要输入的文本加黑，例如“**Type Is-1** 然后按 **Enter**”。当我们提到命令、文件名或库时，使用单字体（或者固定宽度字体）来表示，例如，“正如你能看到的，Is 命令列出两个文件：fileone.txt 和 filetwo.lgz，这两个文件定位在 directoryone 库中。”

本书的结构

本书由三个不同的部分组成，主要内容包括在第 2 部分和第 3 部分。

第 1 部分：介绍

本书的第 1 部分将作为一个导航图来告诉你在本书的其他部分将要讨论的内容。主要介绍将要涉及的工具，也包括有关 PetSoar 的应用概况和结构。

第 2 部分：建立开放源码工具箱

本书的第 2 部分将介绍我们所拥有的许多开放源码项目，这已被包括在过去几年中用 J2EE 所创建的“开发者工具箱”中。我们将讨论使用每项技术的理由，从产品的使用中学习新的东西，以及最终选择的可以提供类似或互补的功能的开放源码项目。

第 3 部分：开发应用

本书的最后部分全面完整地介绍了通过使用这些开放源码项目来实现 PetSoar 的不同特性。在这个部分中，我们也将正式向你介绍有关开发理论的轮廓，并向你展示如何成功地把它们融合到你的专业工作中，并产生令人难以想像的成功应用。这部分展示了技术和方法是如何结合在一起而完成一个应用的。

本书的内容安排是逐步深入的，因为我们知道，我们是在为不同的读者编写此书，有些读者可以直接阅读本书的主要章节或有关部分。如果你已经掌握了第 2 部分所描述的工具，可以跳过这些章节。或者，如果你想学习一个完整的工具，可直接阅读第 2 部分；或者，如果你正在构建一个不是以 Web 为基础的企业应用，你可以跳过第 3 部分的有关章节。

本书之外

本书为你使用开放源码工具并将目标集中在构建简单的实际应用方面提供了一个好的开端，但是帮助并不局限于此，对于初学者来说，可以参考一个相关的网站：

<http://www.wiley.com/compbooks/walnes>

网站内容如下所示：

- 可以下载 PetSoar 应用，并把它安装在你的个人计算机上。
- 对本书内容的增加和有关补遗。
- 一个邮件目录记载了在本书中有关作者和读者使用的工具和方法的讨论纲要。
- 在本书出版后的有关技术更新可使你在工具和方法上与发展同步。

此外，还有一本标题是 *Java Open Source Library* 的书是由 John T. Bell、James 和 Stan Ng (John Wiley&Sons: Hoboken, NJ, 2003) 编写的，主要为“J2EE 开放源码工具箱：用开放源码工具构建一个企业平台”，包括许多在本书中讨论的相同的工具，我们会引用这本书。

目 录

第 1 部分 介 绍

第 1 章 本书概述	2
1.1 使用开放源码技术	2
1.2 了解设计和开发理论	3
1.2.1 先测试	3
1.2.2 少即多	3
1.2.3 总是问傻问题	4
1.3 考察 PetSoar 项目	4
1.4 抓住基础	4
小结	5
第 2 章 应用概况	6
2.1 另一个宠物商店	6
2.2 了解可维护性的重要性	7
2.3 了解 PetSoar 的需求	7
2.4 检查结构和技术	8
2.4.1 查看结构	8
2.4.2 查看实用库	9
2.4.3 使用存留和搜索	10
2.4.4 使用 Web 前端	10
2.5 测试	11
小结	11

第 2 部分 建立开放源码工具箱

第 3 章 用 JUnit 进行单元测试	14
3.1 测试的类型	14
3.2 使用 JUnit	15
3.2.1 JUnit 的特征	15
3.2.2 写一个单元测试	15
3.2.3 运行一个单元测试	17

3.2.4	运行多个测试	18
3.2.5	你需要知道 JUnit 的一切	21
小结		28
第 4 章	用 Mocks 测试对象的交互作用	30
4.1	测试对象的交互作用	30
4.1.1	考察测试状态中的一些易犯错误	30
4.1.2	探索替换物: 测试交互作用	31
4.2	使用模拟对象	32
4.2.1	实例场景	32
4.2.2	了解模拟对象的角色	34
4.2.3	了解模拟对象库	35
4.2.4	使用动态模拟	35
小结		39
第 5 章	用 Hibernate 存储对象	40
5.1	了解持久化的复杂性	40
5.2	用 Hibernate 2 持久化对象	41
5.2.1	创建持久类	41
5.2.2	映射类到数据库中	43
5.2.3	配置 Hibernate	45
5.2.4	获得一个会话	47
5.2.5	在数据库中存储对象	49
5.2.6	从数据库中检索对象	51
5.2.7	查询对象	51
5.2.8	持久化对象间的联系	53
5.3	持久化层次结构对象	58
5.4	了解 Hibernate 工具箱	60
5.5	Hibernate 同其他竞争技术的比较	60
5.5.1	Hibernate 与 EJB	60
5.5.2	Hibernate 与 JDO	61
5.5.3	Hibernate 与 DAO 框架	62
小结		62
第 6 章	使用 WebWork 实现 MVC 模式	63
6.1	了解模型视图控制器	63
6.1.1	检查模型层	63
6.1.2	检查视图层	64
6.1.3	检查控制器层	64

6.1.4	把全部层联系在一起	64
6.1.5	考证使用 MVC 的理由	65
6.2	了解 MVC、WebWork 和 XWork	66
6.2.1	考察 XWork	66
6.2.2	考察 WebWork	66
6.3	进一步了解活动	68
6.3.1	一个简单的 XWork 实例	68
6.3.2	配置 XWork	69
6.3.3	构造活动（活动组成）	69
6.3.4	从 XWork 中调用一个活动	71
6.3.5	使用参数和 ActionContext	72
6.4	应用物理学中的牛顿第三定律	73
6.4.1	了解 XWork 结果和活动链	73
6.4.2	考察 WebWork 结果和 Servlet 环境	74
6.4.3	配置 WebWork	75
6.4.4	了解调度的角色	76
6.4.5	考察 JSP 和 Velocity 中的视图实例	77
6.4.6	基于组件的 Web 开发	80
6.5	表达你自己	84
6.5.1	使用基础表达式	85
6.5.2	使用高级表达式	86
6.5.3	了解 ValueStack	87
6.6	考察类型转换	89
6.6.1	深入研究一个日期实例	89
6.6.2	指定默认转换规则	90
6.6.3	指定新的转换规则	90
6.7	采用截取器分离事务	91
6.7.1	配置截取器栈	91
6.7.2	使用记录截取器	92
6.7.3	构建自己的截取器	93
6.8	Validation——一个强大的截取器	93
6.8.1	考察一个没有使用 XWork 验证框架的实例	93
6.8.2	考察一个采用 XWork 验证框架的实例	94
6.8.3	使用内置的和用户自定义的验证	95
6.8.4	使用验证表达式	97
小结		97

第 7 章 用 SiteMesh 简化布局	98
7.1 识别布局的问题	98
7.2 使用面向对象的解决方案	102
7.2.1 装饰器设计模式	102
7.2.2 组合设计模式	104
7.2.3 结合模式	105
7.3 用 SiteMesh 实现解决方案	106
7.3.1 SiteMesh 的基本原理	106
7.3.2 创建一个装饰器	106
7.3.3 组成网页	109
7.3.4 考察 SiteMesh	112
7.4 使用技巧	115
7.4.1 将装饰器分组	115
7.4.2 不要惧怕引用	115
7.4.3 CSS 是你的朋友	116
7.4.4 最小化 HTML	116
7.4.5 分散关注点	116
小结	117
第 8 章 用 Lucene 添加搜索能力	118
8.1 了解搜索的复杂性	118
8.2 Lucene 简介	119
8.2.1 了解 Lucene 的元素	119
8.2.2 索引一个文件	119
8.2.3 搜索文件	122
8.2.4 重新索引和删除一个已索引的文档	124
8.2.5 使用高级搜索	124
8.2.6 定制标志的进程	125
小结	126
第 9 章 用 XDoclet 生成配置文件	127
9.1 XDoclet 简介	127
9.2 用 XDoclet 了解面向属性编程	127
9.2.1 了解属性的语法	129
9.2.2 运行 XDoclet	130
9.2.3 用 XDoclet 应用高级的 Hibernate 或映射	133
9.2.4 使用 XDoclet 生成更复杂的工件	135
9.2.5 了解 XDoclet 任务和子任务	138

9.2.6 有效地使用 XDoclet	139
小结	141
第 10 章 交流与工具	142
10.1 考察 PetSoar 开发	142
10.2 管理不完善的交流	144
10.2.1 用一切可能的方法交流	144
10.2.2 使用源码交流	144
10.2.3 使用交流作为学习的工具	146
10.3 考察工具箱	146
10.3.1 源代码配置管理——CVS	146
10.3.2 知识管理——Wiki	147
10.3.3 邮件目录——Majordomo	147
10.3.4 实时讨论——IRC 和即时消息	147
10.3.5 IDE-IDEA 和 Eclipse	147
10.3.6 问题追踪和任务管理——JIRA	148
10.4 使用连续集成	148
10.5 把工具连接在一起	149
小结	149
第 11 章 省时工具	151
11.1 了解实用组件	151
11.2 了解 OSCache	152
11.2.1 介绍 SampleNews.com	152
11.2.2 使用 LOOP 测试	152
11.2.3 考察 OSCache 标识符库	154
11.2.4 了解 OSCache 概念	155
11.2.5 缓存时间实例	156
11.2.6 查看高级的 OSCache 特征	157
11.3 了解 Commons Lang	157
11.3.1 考察最有用的类	158
11.3.2 使用 Builder 类	158
11.4 了解 Commons Collections	161
11.5 了解 Commons Logging	162
11.5.1 考察 Commons Logging 的优点	162
11.5.2 一个简单的实例	162
11.6 了解 Commons Digester	163
11.6.1 考察一个 Digester 实例	164

11.6.2 了解 Digester 规则	166
小结	167

第 3 部分 开发应用

第 12 章 建立开发环境	170
12.1 在 IDE 内工作	170
12.1.1 问题: IDE 不是自动化的	171
12.1.2 解决办法: 自动构建工具	172
12.2 将 Ant 用于你全部的构建需要	172
12.2.1 问题: Ant 不是万灵药	172
12.2.2 解决办法: 使用有意义的方法	173
12.3 使用杂交处理方法	173
12.4 组织你的项目	174
12.4.1 通过类型组织文件结构	175
12.4.2 通过部署组织文件结构	176
12.4.3 选择一个结构	177
12.4.4 PetSoar 怎样	177
12.5 管理单元测试	178
12.5.1 了解测试类型	178
12.5.2 考察测试小组、JUnit 和批量测试	179
12.6 使用版本控制	180
12.7 部署 PetSoar	180
小结	182
第 13 章 了解测试驱动开发	183
13.1 为什么先测试	183
13.1.1 先测试与后测试	184
13.1.2 锁定需求	185
13.2 了解测试技术	186
13.2.1 将单元测试与代码放在同一程序包中	186
13.2.2 从不跳过失败的测试	186
13.2.3 使用模拟对象孤立无法测试部分	187
13.2.4 何时使用接口和类	187
13.2.5 保持简洁	187
13.2.6 自顶向下工作	188
13.2.7 使用许多小的对象	188

13.2.8	保证测试组快速运行	188
13.2.9	避免静态和单件	189
13.3	测试 TDD 循环	189
13.3.1	使用 TDD 的实例场景	189
13.3.2	实例场景重访	191
13.3.3	加强功能	192
小结		194
第 14 章	管理生命周期和组件依赖	195
14.1	了解组件和服务	195
14.2	处理依赖	195
14.2.1	使用直接实例化	195
14.2.2	使用工厂	196
14.2.3	使用注册	197
14.2.4	使用容器	197
14.3	了解组件生命周期	198
14.4	了解反转控制	198
14.4.1	考察优点	199
14.4.2	考察缺点	199
14.5	了解分离关注点 (Separation of Concerns)	200
14.6	使用容器来定义范围	200
14.7	使用 XWork 的容器	201
14.7.1	配置容器	201
14.7.2	了解容器是如何工作的	202
14.8	测试 XWork 组件	203
小结		205
第 15 章	定义域模型	206
15.1	考虑域模型的优势	206
15.2	处理层	206
15.2.1	层驱动与特性驱动方法的比较	207
15.2.2	选择自底向上、自顶向下、中间分离	208
15.3	确定当前的目标	209
15.4	向店中添加一个宠物	209
15.4.1	使用 Hibernate 实现 PersistenceManager	212
15.4.2	我们的进展	220
15.5	检索宠物	220
15.5.1	检索一个单一的宠物	223

15.5.2 我们的进展	226
15.6 持续分类目录	227
15.6.1 我们的进展	230
15.6.2 实现 Category-Pet 关联	230
15.6.3 我们的进展	234
15.7 考虑性能和可维护性	235
小结	236
第 16 章 创建基于 Web 的界面	237
16.1 添加一个宠物	237
16.1.1 创建 AddPet 活动	237
16.1.2 为 AddPet 创建视图	240
16.1.3 把全部连在一起试用	242
16.2 显示一个宠物	243
16.2.1 创建一个 ViewPet 活动	243
16.2.2 重构活动	246
16.3 编辑宠物	249
16.3.1 检查有效性	251
16.3.2 把全部连接在一起——第二次试用	251
16.4 把宠物从这里拿走	252
16.5 重构 CRUD	256
16.5.1 消除在 ViewPet 和 RemovePet 中的重复	256
16.5.2 剩下的工作	258
16.5.3 完成最后一个重构	260
16.5.4 解耦验证	261
16.6 更换宠物	264
小结	266
第 17 章 定义导航、布局、视觉和感觉	267
17.1 组件化窗口组件表单	267
17.2 形成更美妙的视感和感觉	270
17.3 使用样式化风格	272
17.4 导航到更好的用户界面	275
小结	277
第 18 章 实现浏览和查询功能	278
18.1 定义应用需求	278
18.2 浏览分类列表	278
18.3 浏览宠物列表	280