HZ BOOKS

BITTER JAVA

中文版

（美）Bruce A. Tate 著
苏金国　等译

NNING

机械工业出版社
China Machine Press

BITTER JAVA

中文版

（美）Bruce A. Tate 著
苏金国　等译

机械工业出版社
China Machine Press

本书系统地介绍了常见的服务器端Java编程错误，以及这些错误产生的原因和解决方案。书中涵盖了基本Java和J2EE概念的反模式，如servlet、JSP、EJB、企业连接模型和可扩展性等，通过代码示例展示了Java编程中常见的陷阱，还提供了重构代码，并解释了为什么新方案是安全的。本书适合中级水平的Java程序员、分析员或架构师阅读，通过研究书中介绍的反模式，可以吸取别人的经验教训，在工作中少走弯路。

Bruce A. Tate: Bitter Java (ISBN 1-930110-43-X).

Original English edition published by Manning Publications Co., 209 Bruce Park Avenue, Greenwich, Connecticut 06830.

本书版权登记号：图字：01-2003-2788

图书在版编目（CIP）数据

Bitter Java中文版 /（美）塔特（Tate, B. A.）著；苏金国等译. －北京：机械工业出版社，2006.2
书名原文：Bitter Java
ISBN 7-111-18315-0

Ⅰ. B…　Ⅱ. ① 塔…　② 苏…　Ⅲ. JAVA语言－程序设计　Ⅳ. TP312

中国版本图书馆CIP数据核字（2006）第000791号

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：盛思源　刘立卿
北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行
2006年2月第1版第1次印刷
787mm × 1020mm 1/16 · 16.25印张
印数：0 001 -4 000册
定价：35.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

序

说实在的，很少有计算机的书能把我迷住。有时，作者的睿智和对技术的精通不能不让我折服，Guy Steele的《Common LISP: The Language》就是这样，我记得很清楚，我是躺在风和日丽的夏威夷海滩上一口气把它看完的。有人可能会笑话我这么“滑稽”，也许他们笑我是有道理的，不过对我来说，Steele的每一章都写得那么引人入胜，让人忍不住想再看下一章。这本书真是让我爱不释手。

再就是一本看似小说的书，其中却不断的有令人惊奇的新发现。一方面它会牢牢地吸引住读者，让人沉迷其中，忘却周遭的现实世界；另一方面又用深刻的事实提醒着你，这些事实是你一直以来迫切搜寻的，稍纵即逝，尽管你知道应该认识到这些问题，但往往出于某种原因忽视了它们的存在，这本书则举起大锤让你警醒。Tom DeMarco的《The Deadline: A Novel About Project Management》就是这样一本书。同样让我不忍释卷。

对我来说，Bruce Tate写的《Bitter Java》又是一个奇迹，它同样让我着迷。与DeMarco所说的绑架到摩罗维亚（译者注：这是作者所杜撰的一个奇境国家）一样，Bruce自己的种种“极限运动”：皮划漂流、在崎岖的山路上骑车，还有热气球冒险旅行，这些无不让我随着他们在急流中“上下颠簸”，“疾速挥桨”想看到下一个模式或陷阱。就像Steele的书一样，每看完一章后，我简直等不及看下一章；DeMarco的书让我手不释卷，看《Bitter Java》的感受也是一样，同样是舍不得放下。

要问我的建议？很简单，如果你没办法放下手边的事情，不能安排出全天的时间来静心地看这本书，那你先别开始看。如果天色已经很晚了，那我很同情你，因为你会忍不住熬夜把它看完，这样你明天肯定会疲惫不堪。如果你躺在夏威夷海滩上看这本书，最好用SPF 99的防晒霜。在你津津有味看书的时候，你会忘记时间，那么长时间地曝晒在烈日之下，对你的皮肤来讲实在是够受的。

《Bitter Java》确确实实把我震住了，希望你也一样。如果你要开发软件，或者要与开发软件的人共事，会发现每一章的内容都那么精辟，你会把它们引为经典，我的下一个设计评论中就打算引用Bruce的观点。《Bitter Java》里凝聚着作者的智慧和经验，这是每一个优秀软件工程人员梦寐以求的。我觉得第9章有关Java编码标准的内容可以自成一本书，要是有这样一本书，衷心希望所有Java程序员都能仔细阅读，精心钻研。

按照Bruce的观点，显然软件工程与在危险的急流上飞流直下可有一比，尽管不像在标准的4级急流中那样有生命危险，但是软件工程的失败对于你的生活（以至你所领导的人们的生活）来讲，可能同样是一场灾难。所以建议你仔仔细细地好好研究这本绝妙的指南。Bruce在这本短小精悍的书中，凝聚了一个能真正驾驭急流的高手的真知灼见，这些经验不可多得，讲述得清晰明了，读起来轻松有趣。

做好准备，尽可能地去体会这些危险之旅带来的震撼吧！不过别被吓倒了，遇到危险就狼狈得丢盔弃甲！

祝你愉快！

Hays W. "Skip" McCormick III，《AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis》的作者之一

参加本书翻译的人员：苏金国、林琪、任岗、张野、刘鑫、杨健康、王宇、陈波、王小振、杨明、伊瑞海、朱涛江、张文静、荆涛。

前　言

到了夏天，得克萨斯河水就几近干涸。为了寻找急流，漂流者不得不跟着暴风雨的脚步走。那是1996年夏天的一天，我和一个同伴晚上8点离开奥斯汀，冲入狂暴的风雨中，一直来到阿肯色州的Cossatot河。等我们到了那里，坏天气好像跟我们开了个残酷的玩笑，居然绕过这条河径直走了。我们筋疲力尽，失望之极，只好在河岸上扎营休息。那天晚上我们根本没有听到一丝雨声。

到了早上，我还是垂头丧气，头昏眼花地走出帐篷，然后几乎跌倒……在河里。Cossatot河素来就有涨水极快的坏名声，仅仅因为上游10英里处下了2个小时的雨，水位就突涨了6英尺。现在我们倒是可以漂流了，但是水位又太高。我们决定等第二天早上再来对付难度大的这一段，先到相对容易一些的上游漂流。原来水流迟缓的1级水域现在已经变成了汹涌的3级急流。指南上说这一段要漂流“4个小时”，但我们只花了20分钟就飞驰而下。“中间”河段更糟糕：湍急的河水已经达到4级，猛烈地咆哮着。经过仔细侦察，我们轮流在河岸上执守，一个人在河水中漂流时，要系着安全带，由另一个人在岸上监视情况。然后我们把皮划艇放在营地，徒步走下去，看看下面的河段情况怎么样。让我们惊讶的是，居然有几十个当地人在河岸旁放着躺椅，像看风景一样看着河面。以往他们看到的只是4级瀑布，如今这条河已经完全被可怕的大漩涡所笼罩。此前，我们很少看到当地人，他们在这里只是看有没有人出风头，有没有惊险的事情发生。这种景象让我们目瞪口呆，所以我和同伴也各自坐在一块大石头上开始看热闹。

追溯到2000年，尽管当时我的职位已经不低，而且待遇优厚，但我还是离开了IBM，去加入一家名叫allmystuff的创业型（startup）公司。当时经济已经开始衰退，但是在奥斯汀其他创业型公司纷纷垮台之际，这家公司却刚刚拿到了赞助。这家公司的企业运作并不取决于广告收入，所以尽管广告收入日薄，似乎也影响不大，另外allmystuff里集结了许多精兵良将。我加入公司之时，它的银行资金有一千万美元，不仅有固定客户，而且拥有着高新技术，这一切都预示着它很可能成为炙手可热的成功企业。我见过许多朋友都离开IBM大旗，转而投奔其他公司，虽然新公司的待遇不能比，也没有安全感，但是很有冒险性。我想反正在必要时还可以回去，所以面对着即将到来的黑暗，我迎头冲入到这场风暴中。

在奥斯汀新闻笑谈曾经红极一时的创业型公司都纷纷落马之时，allmystuff也开始在困境中挣扎。我们日以继夜地工作，为很少的几个客户部署解决方案。尽管我们的质量一流，有让人自豪的业绩记录，但最后还是被衰退的经济所累。风险投资者决定最好还是关门大吉，再找寻一种适应这种经济衰退局势的新概念东山再起。尽管这件事本身让人很难受，但是在我的职业生涯中，那个时期我学到的东西却是任何其他时候都比不上的。

就像Cossatot河岸上的当地人一样，如果一个冒险故事是我们身边发生的真事，很刺激，甚至很危险，那我们大多数人都无法抵挡它的吸引力。不论是看一个久负盛名的希腊悲剧故事，

还是看像电视剧“生存者”（Survivor）这样一些最新的流行节目，我们的猎奇思想永无止境。程序员也不例外。我们很喜欢聊最近发生的冒险事情（我们把这称为“实境谈话”（merc talk）），这有很多原因。我有许多鲜活的工作记忆就是在allmystuff的乒乓球台边留下的。在那里我们讨论过管理哲学；讨论过代码基是不是已经失控；另外还讨论过，与日益复杂的JSP模型相比，有专门浏览器的XML是不是一种更简单的方案。我们还讨论过，眼看着进度不断推迟，图形化设计人员能不能把他设计的用户界面映射为越来越复杂的Java命令。正是这些讨论燃起了我的热情，促使我离开了原本安稳的职位，做着百万富翁的梦，投向这个待遇更低、没有安全感的新工作。这些经验使我成为一个更好的程序员、管理人员和架构师。

不记得在哪个场合下，前IBM主席John Akers曾说过，太多的人“整天都只是喝水聊天，无所事事”。我记得听了这话我们都很生气，他不知道，往往在喝水（或喝酒）时或者在乒乓球台边听到的东西会决定一个项目（甚至一个公司）的成败。在这里，必须听些程序员的故事，受些熏陶，因为这些会影响一生。我准备把其中一些故事放到《Bitter Java》这本书里。

再回到早先，那时我还没有加入allmystuff，正准备在一个会议上演讲。我报告的题目是“Bitter Java”。在会议期间，我遇到了一位著名的Java程序员，JSP的创始人之一。他告诉我曾经在Pamplona参加过“公牛奔跑”（run with the bulls）活动（译者注：这是一个很经典的活动，人们拼命地在公牛前面奔跑，一路上公牛会踩伤、踢伤或用角刺伤很多人，但人们还是乐此不疲，认为这是勇敢者的游戏），还被刺伤了。他还很起劲地给我解释他在参加公牛奔跑时的策略。我对他讲的不以为然。在Pamplona，早有无数的人告诉过我怎样避免被刺伤。我还向O. J. Simpson咨询过这方面的问题。每年都有数万个热衷于此的人参加，但只有十几个被刺伤。不过，慢慢地我有了想法：如果我要参加公牛奔跑，那我就会和他讨论。我想知道他是怎么计划的，他又怎么实施他的计划，哪里出了问题。这些信息我能用得上。后来发现，这个被刺伤的程序员正是allmystuff工程部的副总裁，他招募我帮助建立他的服务机构。再来说我的报告，尽管讲这个Pamplona故事可能会让我失去在allmystuff工作的机会，但我还是决定用这个故事来开始我的演讲。它充分体现了《Bitter Java》中的概念。要说能帮助避免一个微妙的圈套或陷阱，一个关于失败的故事往往抵得上10个成功的故事。这个故事牢牢地吸引住了听众，而且……我也得到了我的工作。

像许多程序员一样，我很喜欢极限运动。我们曾划着小艇遭遇危险，有时甚至遇到生命危险。William Neely曾经讲过漂流界很有名的一个法则：你注视一个急流的时间与它吞掉你的危险往往成正比。换句话说，如果看上去漩涡大到能吃掉你，它很可能就会吃掉你。漂流者有自己的一套办法来描述如何沿河下行。漂流指南中会指出一条路线，还会指出路线沿线以及路线之外的一些危险地方。指南中可能说，“接下来，你会看到中间有一块大石头，你要向左。如果误撞到右边，那这个急流就会成为‘终结者’，用残酷的方式告诉你错了。”我很清楚，即使你没有真正了解一条河的威力，你也很想知道哪些地方可能出问题。我想知道水下有没有岩石，有没有陷阱等着我。我想知道哪里可能遇到漩涡，怎么躲过瀑布底下的大石头。我想知道，是不是有人在这条河上丧生，那是怎么回事。如果没有足够的了解，就算有高超的技术，我往往也会回避，甚至顶着小船沿河岸一路走下去，就是不下水。

程序员（包括我）也是一样。我要了解应用和项目会在哪里失败。我要知道是不是与某个

接口的通信太多了，所用的技术能不能解决这个问题。我得明白一个技术可能在哪里出问题，这个技术能不能扩展。

我深信，要想成功，软件开发中总少不了失败，而且势必要从中学习。我还没有看到哪个组织能系统地从错误中学习，并以正规、系统的方式仔细分析为什么要修改一个有问题的设计模式或过程。我曾经看过许多代码，但并不是所有代码都很好。我已经领会到“bitter Java”的魅力，希望你也一样。

致谢

Bitter Java概念于3年前萌生，最早是我在一个1小时的演讲中提出的。Braden Flowers帮助我完成了最初的定型，对于这一点，我对他实在是感激不尽。Bitter Java网站于2年后推出。尽管经过了一段时间的讨论（甚至争吵），Manning工作人员、诸位审校人员、一些朋友、另外一些作者还有我，最后还是建立了一个紧密协作的写作小组，大家横跨三大洲，终于在很短的期限内使这本书得以问世，不仅如此，如此高质量也超乎了我的想像。尽管我想逐个地提到每一个人的名字，但是由于太多的人做出了贡献，伸出过援手，实在无法一一列出大家的名字。

我原来还曾写过另外两本技术书，能得到Manning提供的世界一流的支持，真让我很受感动，也为之欣喜万分。我要感谢Marjan Bace两次：首先要感谢你耐心的指导，在整个过程的每一步都坚持力求完美，其次要感谢你热心地帮助我完善写作风格，帮助我修改草稿。你对文字娴熟的驾驭能力激励着我，你对别人的理解（包括你的顾客、你的竞争对手，还有你旗下的作者们）对我的影响也无以言表。

要特别感谢Ted Kennedy，他提供了卓越而专业的审校，而且在我最需要的时候给予了支持。还有许许多多的人花费了大量时间审阅这本书，这里要感谢你们大家：Jon Eaves、John Crabtree、John Mitchell、Steve Clafin、Sean Koontz、Max Loukianov、Greg Wadley、Tim Sawyer、Braden Flowers、Ted Neward、Eric Baker、Jon Skeet，还要特别感谢Juan Jimenez，你以后肯定会成为一个非常棒的编辑。你们的意见都有非凡的意义，你们的批评（无论尖锐还是温和）对我都有很大帮助，所以我才能把这本书写得更好。希望你们知道，尽管并非每条建议我都采纳，但我确确实实仔细地考虑过每一条意见，意见真的是太多了。真希望哪一天也能对你们写的书提提我的意见。

接下来要感谢我们的出版团队。首先要感谢Liz Welch，谢谢你的热心和高效率，还有高超的编辑水平（特别是你对正确性和一致性的执着）；还要感谢Susan Capparelle、Helen Trimes和Mary Piergies，凭着你们非凡的能力，得心应手地推动了这个出版项目的前进；另外要感谢Dan Barthel一直支持着这个项目，并“发现了”Braden和我；感谢Leslie Haimes制作了这么精良的封面；感谢Bruce Murray和Chris Hillman出色的技术支持；感谢Lee Fitzpatrick 在我们的团队不断调整的时候一直给予支持，最后要感谢Tony Roberts让这本书最终成型。

在此还要向为这本书贡献过思想和内容的人表示感谢：Brian Dainton和他的Contextual团队贡献了风格指南；Tony Leung提供了“堵住内存泄漏”（Plugging Memory Leaks）中的示例代码；Amandeep Singh提供了读/写锁代码；Mike Conner提供了“利用缓存扩展电子商务应用”

（Scaling Up E-business Applications with Caching）中所用的图。还要感谢Mike Conner建立的团队很好地研究了“三角形”（Triangle）模式，这正是《Bitter Java》中的核心模式。另外要感谢Mark Wells为我指点了读/写锁思想。特别要感谢Skip McCormick III，不光是因为他作为始创者最早提出了反模式，并让这种思想发扬光大，还因为他为这本书写了一篇绝妙的序，如我所愿，充分地反映了这本书的精神。另外再一次感谢你，Braden Flowers，谢谢你贡献的EJB例子。

2001年对整个国家、整个高技术产业，以及我的家庭都是很困难的一年。尽管如此，亲爱的Maggie，你一直在我身旁，支持我、鼓励我、督促着我。你是我的编辑、同事、策划、支持者和特殊的朋友。真心地感谢你，对你的爱天长地久。

关于本书

反模式的研究和应用是程序设计领域新的前沿课题之一。在《Bitter Java》中，我们将利用个人的经历和特定的例子探讨这一主题。这里包含有许多代码，我还对同一个例子做了多次重构，希望能进一步改善它的性能和可读性，直到最后的例子确实很好为止。

本书的组织结构

这本书包括11章，分为3个部分，另外还有一个附录。有关本书组织和结构的详细信息请见第1章的1.5节。以下是一个概述。

第一部分涵盖了设计模式、反模式和支持Internet服务器端开发有关标准的基础知识。

- 第1章将反模式与其他使用类似概念的行业做了比较。例如，医药行业会采用预防措施（设计模式）和治疗手段（修正），但是最好的医生能诊断出根本病因，比如体重过高或压力太大等等（反模式）。我还会在这一章更深入地介绍这本书的组织结构。
- 第2章将介绍基本的Internet标准以及Java，还提供了一个很简单的过程介绍：即实现服务器端Java开发需要些什么。这不能算是一个全面的教程，但是它能让你知道在哪些方面还需要进一步的学习。

第二部分详细地介绍了服务器端反模式，首先从一个很糟糕的公告栏应用入手，并对它重构，分别解决各个反模式。

- 第3章介绍了基本的服务器端反模式，并定义了三角形（Triangle）设计模式，这个设计模式在allmystuff（现在的Contextual）公司和IBM很常用。这里的核心反模式是“神奇servlet”（Magic Servlet），也就是企图仅由一个servlet做所有的工作。
- 第4章涉及JSP中的反模式。由于JSP有一种标记语言，所以这一章与其他章感觉上有一些不同。
- 第5章建立了一个缓存用例，显示出在企业的多个层次进行缓存可以呈数量级地提高性能。
- 第6章指出导致内存泄漏的Java问题，并讨论了故障恢复技术。
- 第7章处理的是与连接两个系统有关的反模式。讨论紧耦合系统时，关键问题是连接抖动；XML和Web服务则构成了松耦合系统的基础。
- 第8章介绍了EJB。在这里我提供了EJB的基础知识，并给出了一个更糟糕的设计，在此把公告栏应用的模型组件实现为EJB实体bean。

第三部分介绍了公告栏应用例子之外的更高层细节。

- 第9章讨论了编程卫生。这里给出了Contextual公司实际使用的一个编码标准指南，并提供了一些让代码更可读、更好理解的建议。
- 第10章将谈到性能反模式，从实现到处理再到调优都有涉及。在此还讨论了如何建立体系

结构来实现可扩展性。

- 第11章是全书的总结，将介绍如何在你的项目中、你的职业生涯中以及你的企业中应用反模式。

附录中分别按名字、出现领域和症状提供了反模式的三个参照表。

如何使用本书

建议读者先把全书按章节顺序看一遍，起码要略读一遍，以便对反模式有个总体的印象，然后再返回去有针对性地查看某几章和例子作为参考。

如果读者还需要另外的帮助，或者有问题想问作者，请访问我们提供的在线资源（见下面的介绍）。

面向读者

《Bitter Java》面向的是中级服务器端Java程序员或架构师，不过其他人也能从中获益。作为程序员，如果你是新手，一定会喜欢我们简洁易懂的语言，并接受我们的思想，也就是要全力理解和掌握反模式。如果你是高级程序员，则会从中发现有关XML的一些新的反模式，另外，能从一个新的角度研究模式，这对你也很有好处。

除了从头到尾阅读这本书，你还可以参考其他有关“Bitter Java”的资源。任何一本书都不可能独立存在，读者应当充分利用第1章所列的各种在线资源和参考书，当然还要好好利用http://www.manning.com 和http://www.bitterjava.com上的在线资源。这本书的最后还给出了一个参考文献，分门别类地列出了本书参考的资源。

尽管Java会带来苦痛，但我希望阅读这本书能让你有一个顺利而愉快的体验。

源代码

这本书包含了大量源代码例子，其中大多数都是基于servlet的服务器端例子。这本书还包含有Braden Flowers提供的EJB编程例子。

源代码例子可以从http://www.bitterjava.com或出版商的网站http://www.manning.com/tate在线获得。

在线资源

请读者充分利用两个最重要的Internet资源：

- Manning的作者在线（Author Online）网页为Manning已出版的所有书都提供了一个内部论坛（只有购买了图书的人方可进入）。让浏览器指向http://www.manning.com/tate，就可以访问到Manning的《Bitter Java》论坛。然后可以按照说明来订购和访问这个论坛。
- 《Bitter Java》网站（http://www.bitterjava.com）针对Java反模式和示例程序提供了一些开放的论坛。通过这些论坛，读者可以参与一般的反模式讨论，甚至可以对书中的某些主题做出分析，提出见解。《Bitter Java》出版后不久，你就能下载程序例子了。另外不要忘记

注册《Bitter Java》email时讯，我会每隔一个月发布一次，如果我有你感兴趣的信息，也会随时发出时讯邮件。你也可以申请得到以前的时讯。

你可以利用这些资源问问题，对《Bitter Java》做评论，还可以得到其他读者和作者本人的帮助。我们已经搭建起在线群体的基础架构，能不能建立起在线群体，剩下的就要看你们的了。

关于封面插图

《Bitter Java》封面上是一个Azanaghi阿拉伯人，他们在现在的毛利塔尼亚最北部生活过几个世纪。关于这些人的生活和居住地点，详细情况不详，相关历史已经找不到了，而且艺术家没有画出他的脸，这更增添了他的神秘感。这个插图选自1799年Madrid首次出版的西班牙民族服饰纲目。这本书的首页上写着:

Coleccion general de los Trages que usan actualmente todas las Nacionas del Mundo desubierto, dibujados y grabados con la mayor exactitud por R.M.V.A.R.

Obra muy util y en special para los que tienen la del viajero universal

我们尽力遵从原意，将其直译为:

对于现今世界中多个国家的多种不同服饰，R.M.V.A.R设计并印制了一些惟妙惟肖的图片。这对于周游世界的旅行者来说尤其有用。

这些图片完全是手工绘制，尽管我们对这些图片的设计者、刻版者和着色者一无所知，但是从图片本身可以清楚地看到他们的工作确实是“惟妙惟肖”。“神秘的Azanaghi阿拉伯人”只是所收录的众多精美图片中的一幅。纲目中收录的各种各样的图片生动地反映了200年前世界各地各自鲜明的特色。在那个时期，即使两地仅相距数十英里，两地的服饰也可能是迥异的，单从人们的服饰就可以清楚地区分出他们是哪里人。这些图片可以活灵活现地反映出当时存在的隔离感和距离感，其实除了当今这个剧烈动荡的时期外，历史上所有阶段都存在着这种距离感。

从那以后，服饰语言已经发生了很大变化，原先各个地域都有多种多样的服饰，这种情况也在逐步消失。现在即使是不同大洲的居民，通常也很难加以区别。如果从好的一方面看，我们也许是在用文化的多样性来换取个人更为多样化的人生，也可以理解为换取一种更加多姿多彩、更有意思的才智人生。

图片中描绘的情景栩栩如生，这种封面让人回望过去，我们对Manning出版公司在计算机图书中采用两个世纪前的各色地域风情作为封面的首创性和娱乐性也很满意。

目　录

第三部分 全 景 图

第一部分 基础知识

1993年，我第一次在一条4级河流上漂流而下：这就是阿肯色州的Cossatot河。为了做好充足的准备，我在得克萨斯州一些难度较低的2级河流上与一些有经验的漂流者们练习了好几个小时。我学到了一些常用的术语，还有一些重复性动作，要想在一次新的漂流中安然无恙，这些动作必须掌握。尽管这些河流很容易应付，不过我充分利用了这段时间来做准备，以便有足够的信心面对以后到来的挑战，这是令人愉快的几个小时。

第1章和第2章建立了模式、反模式和服务器端Java编程的基础。在第1章中，我们讨论了设计模式对当前程序设计的影响，还说明了为什么反模式是一个很有必要的补充研究内容。这一章提供了有关检测工作的一些提示，针对一些重复出现的Java问题，分析了可以采用哪些方法来建立相应的模式。在第2章中，我们讨论了服务器端Java编程当前的状况，从基本的Internet标准和语言，到方法论的改进都有谈到。在此过程中，我们会搜寻反模式可能藏身的隐蔽之处，并寻求改进，来帮助我们与这些反模式抗争。