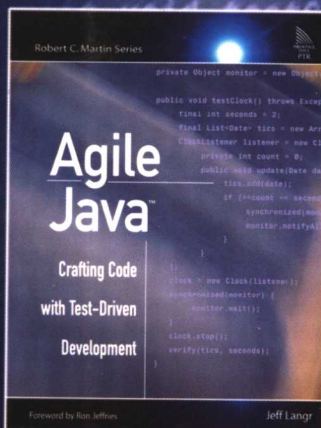


Robert C. Martin Series



Agile Java

中文
版

[美] Jeff Langr 著

涂波
孙勇

译

测试驱动开发的编程技术

Crafting Code
with Test-Driven
Development

Foreword by Ron Jeffries

Agile Java 中文版

——测试驱动开发的编程技术

Agile Java Crafting Code with Test Driven Development

[美] Jeff Langr 著

涂波 孙勇 译



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内容简介

本书将当前流行的敏捷开发方法融入到了 Java 语言的实践中, 给了您学习并掌握 Java5.0、面向对象方法的机会, 同时您还将学习测试驱动开发方法。作者很好地将三者融合在一起, 全力教读者如何使用 Java5.0 开发专业的软件。

本书语言浅显易懂, 颇有趣味, 覆盖了上述三方面的内容, 准确地展示了如何将 Java 和 TDD 进行有效的整合; 帮助开发者在整个软件生命周期中使用这些方法, 以适应现代软件行业对高开发速度和高准确性的要求, 对于希望使用 Java 5.0 作为开发工具的开发人员而言, 本书是一本很好的入门书籍。

从市场来看, 现在的软件业竞争非常激烈, 软件的业务需求变化快, 经常会出现变更, 传统的软件开发方法已经很难适应这种快速变化, 在这种高速变化的情况下, 显得捉襟见肘; 而融合了敏捷方法的 Java 可以从某些方面应对这种变化, 这使得本书成为 Java5.0 学习者一本优秀的参考书。

Authorized translation from the English language edition, entitled Agile Java™: Crafting Code with Test-Driven Development, 1st Edition, 0131482394 by Langr, Jeff, published by Pearson Education, Inc, publishing as Prentice Hall PTR, Copyright©2005 Pearson Education, Inc.

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 PUBLISHING HOUSE OF ELECTORNICS INDUSTRY Copyright ©2006.

本书简体中文版由电子工业出版社和 Pearson Education 培生教育出版亚洲有限公司合作出版。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

本书简体中文版贴有 Pearson Education 培生教育出版集团激光防伪标签, 无标签者不得销售。

版权贸易合同登记号: 图字: 01-2005-0910

图书在版编目 (CIP) 数据

Agile Java 中文版: 测试驱动开发的编程技术 / (美) 兰格 (Jeff.L.) 著; 涂波, 孙勇译. —北京: 电子工业出版社, 2006.9
书名原文: Agile Java: Crafting Code with Test Driven Development
ISBN 7-121-02704-6

I .A... II .①兰...②涂...③孙... III.JAVA 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字 (2006) 第 054418 号

责任编辑: 周 筠

印 刷: 北京智力达印刷有限公司

装 订: 北京中新伟业印刷有限公司

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

开 本: 787×1092 1/16 印张: 42 字数: 680 千字

印 次: 2006 年 9 月第 1 次印刷

定 价: 69.00 元

凡所购买电子工业出版社图书有缺损问题, 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;
邮购电话: (010) 88254888。

质量投诉请发邮件至 zhs@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

对 Agile Java 的赞誉

“这是我阅读过的最实用、最高效、并且令人愉悦的编程书籍。此书具有结果导向的特点。”

Steve Bartolin, President and CEO, The Broadmoor

“终于有了这样一本书——书中提供了把 Java 和测试驱动开发整合在一起的、作为扎实起点的课程！我知道，如果在最初学习 Java 的时候，就能读到 Jeff 的这本书，那么大多数的软件开发人员都能避免某些给您带来多年痛苦的编程习惯。”

Steven A. Gordon, Ph.D., Manager, Arizona State University Software Factory (<http://sf.asu.edu/>)

“Jeff 利用他在 Java 和敏捷过程方面的特别深入的专业知识，向我们展示了一种强大的方法。利用这种方法，我们能够构建干净的、结构良好的、并且经过彻底测试的 Java 程序。强烈向您推荐这本书。”

Paul Hodgetts, Founder and CEO, Agile Logic, Inc.

“这是一种优秀的学习 Java 的方式。Agile Java 不止讲述 Java 语言的基础、配置、以及工具，而且还针对测试驱动开发的概念、重构、以及面向对象编程，提供了卓越的指导。我们整个工程师团队都要求阅读这本书。”

Andrew Masters, President and CIO, 5two8, Inc.

“任何开发 Java 应用的程序员都必须阅读这本书。书中包含了关于如何比传统方法更快、更有效率地编写高质量、可扩展的 Java 程序的思想，这些思想是创新的、富有洞察力的。初学者和有经验的程序员都能从本书中汲取大量的营养。”

Bret McInnis, Vice President eBusiness Technologies, Corporate Express

Robert C. Martin 系列

本系列的目的是提高软件开发的技术水平。系列中的所有书籍都是关于技术的，并且注重实践、内容充实。所有作者都是经验丰富的软件大师，他们都专注于实践，而不是理论。通过阅读，您可以知道他们都做过什么，而不是他们想了什么。如果系列中的某本书是关于编程，那么书中会有很多代码。如果系列中的某本书是关于管理，那么会有大量来自于实际项目的案例学习。

任何严肃的从业者都会把下面这些书籍摆放在自己的书架上。这些书籍与众不同，并且可以指导从业者成长为软件大师。

Agile Java™: Crafting Code with Test-Driven Development

Jeff Langr

Working Effectively with Legacy Code

Michael Feathers

UML For Java Programmers

Robert C. Martin

Agile Software Development: Principles, Patterns, and Practices

Robert C. Martin

Agile Software Development with SCRUM

Ken Schwaber and Mike Beedle

Extreme Software Engineering: A Hands on Approach

Daniel H. Steinberg and Daniel W. Palmer

Agile Java 中文版

译者序

测试驱动开发是极限编程（Extreme Programming）的重要特点。它以持续测试来推动代码的开发，既简化了代码，又保证了软件质量。测试驱动开发起源于 XP 方法中提倡的测试优先实践。测试优先实践重视单元测试，强调程序员除了编写代码，还应该编写单元测试代码。在开发的顺序上，它改变了以往先编写代码，后编写测试的过程；而是采用先编写测试，后编写代码来满足测试的方法。这种方法在实践中能够起到非常好的效果，使得测试工作不仅仅是单纯的测试，而且成为设计的一部分。

测试驱动开发是近年来“敏捷革命”中最热的话题之一。但是，市面上鲜见与此相关的有深度的、和 Java 开发紧密结合的作品。本书的出版填补了这一空白，能够满足众多 Java 程序员的需要。本书的特点是：

- 一本结合 Java、描述测试驱动开发方法的非常实用的指南：真实的问题，真实的解决方案，真实的代码。
- 本书包含了一个完整的用 Java 编写并使用 JUnit、Ant 等工具的软件项目。
- 本书适合每一位对测试驱动开发方法感兴趣的 Java 开发人员和项目经理。

快乐工作的基础就是对自己有信心，对自己的工作成果有信心。测试驱动开发提供的测试集可以作为您信心的源泉。相信您在阅读本书后，也一定会为这种敏捷的开发过程所感染，喜欢上 Java，喜欢上测试驱动开发的技术。

非常荣幸能有机会和孙勇先生合作，翻译这本书。由于本人才疏学浅，翻译中难免存在错误，衷心的欢迎读者指正。特别要感谢聂涛先生和魏泉先生。他们对本书的初译进行了非常细致的技术审阅，我从中学到了很多。没有聂涛先生和魏泉先生的辛苦工作，就不会有本书现在的翻译质量。另外，要感谢我的父母，还有我的妻子吕玫，他（她）们的支持永远是我继续向前的动力。

为了保持原书的风采，本书的封面、版式、字体等各方面均维持原书的风格。关于本书中文版的文字或技术问题，您可以发送 email 到：javatdd@gmail.com。

涂波

2006 年 4 月于北京

关于作者

Jeff Langr 是一名有着多年开发经验的独立软件咨询师。他通过自己的公司 Langr Software Solutions (<http://www.LangrSoft.com>), 向客户提供关于软件开发、软件设计, 以及敏捷过程的专业咨询。

Langr 在极受尊敬的 Object Mentor 工作了两年, 这是第一家针对 XP 编程的咨询公司, 公司的老板是 Bob Martin。之后, Langr 又在多家世界 500 强公司任职。当然, 也在倒闭了的互联网公司工作过。

Langr 有在大学里教授 Java 课程的背景。他成功地培养了数百个在 Java、TDD、XP, 以及面向对象开发等方面都非常专业的学生。Langr 也多次在国际会议和地区社团会议上, 发表关于软件开发的演讲。

Langr 编写了入门书籍 Essential Java Style (Langr2000), 这是一本关于如何构建高质量 Java 代码的指南。五年以后, 您依然能在 blog 中发现, 某些人发誓说他们已经把这本书看得折了角。在 Software Development、C/C++ Users Journal, 以及包括 Developer.com 在内的多个在线杂志中, 都能找到 Langr 的关于 Java 和 TDD 的文章。您可以从这里找到关于这些文章的链接: <http://www.langrsoft.com/resources.html>。

他想补充说: 他生活在软件开发的世界里, 希望他对代码进行精雕细刻的热爱, 能够贯穿在本书之中。Langr 居住在 Colorado Springs, 和他的妻子, 以及三个孩子 Katie、Time、Anna 生活在一起。

前言

Jeff Langr 从 agile (敏捷) Java 的角度, 写了一本非常有趣的 Java 书籍: 利用测试驱动开发的技术来雕琢代码。本书的目的是教会初学者如何使用 Java 语言, 以及他和我都知道的最佳的开发方法——测试驱动开发 (Test-Driven Development, 简称 TDD)。TDD 承诺可以带来巨大的潜在价值, Jeff 已经证明了这一点。我非常荣幸为这本书作序, 也非常荣幸向您推荐这本书。

本书不仅对初学者很有用, 对于有经验的程序员而言, 这也是一本能够带给您新的内容, 帮助提升 Java 语言水平的好书。我不打算说这是一本认证指南, 或者和其他很多“Java 大全”相类似的书籍。这不是本书的要点。本书的要点在于帮助您熟练掌握如何使用 TDD。TDD 对于您将来的学习和日常工作, 都会有很大帮助。

本书以面向对象的概念和思想开始阐述。如果您了解对象的概念, 那将有助于阅读这本书。如果您不了解对象的概念, 那就要在继续之前, 去熟悉面向对象。接着, 每前进一步, 您都会使用测试驱动开发技术。如果您没有用过 TDD, 可能在一开始的时候会觉得有些不适应。但是, 如果您象我们一样, 尝试了 TDD, 那 TDD 就会成为您的开发工具箱中常用的工具。

如果您已经安装了 Java 和 JUnit, 请继续。如果没有, 请一定阅读“搭建环境”, 在开始真正的例子之前, 正确地配置您的环境。一旦能够正确编译和运行一个简单的 Java 程序, 就意味着准备就绪了。

Jeff 要求您敲入测试和例子代码, 我也赞同 Jeff 的要求。TDD 的原则是通过实践来学习, 而不只是阅读。您需要形成自己的关于开发节奏的看法。此外, 敲入编程书籍中的例子是最好的学习方法。

在本书中, Jeff 帮助您构建两个应用。一个是学生信息系统, 另一个是国际象棋。在阅读所有章节的过程中, Jeff 向您介绍了 Java 的基础。或许更重要的是, 您会接触到某些最重要的深入的知识, 包括接口、多态、模拟对象、反射、多线程, 以及泛型。

第 10 课是关于 Java 的数学特性的, 在这节课中, 我发现了利用测试来学习语言和库的新特性的方法。很容易理解类似 BigDecimal 的内容, 您可能会想: “我已经明白了。”也许, 暂时您的确是明白了。但是, 当把学到的知识变成测试的时候, 两件事情发生了: 第一, 阅读会丢失一些信息, 通过编写测试可以掌握这些遗漏的知识。第二, 测试记录了我们的学习过程, 以

及在学习中的思考过程。因为我已经养成了保留测试用例的习惯，所以我可以参考它们，唤起我的记忆。我甚至常常把书籍的页码索引到某些测试，当我回过头来希望进一步深入的时候，这些测试提供了注解。

第 11 课关于 I/O，提供了一个我不太熟悉但是非常好的例子。因为我在 Java 方面从事的工作比较少，而且我常用的语言中没有和嵌套类相对应的概念，但是 Jeff 提供了一个非常好的例子，帮助我很好的使用和测试嵌套类。

当我写这份前言的时候，我真的进入了这本书，因为 Jeff 把我带到了我从没有到过的地方。我喜欢那样。第 12 课关于模拟对象，第一个例子是敏捷软件开发者经常会遇到的问题：针对一个定义良好，但是还没有实现的外部 API，我们如何实现增量开发？Jeff 向我们展示如何通过接口定义来实现增量开发——必要的时候，接口定义来自文档——然后构建一个表示您对 API 的理解的模拟对象。编写针对模拟对象的测试，通过这些测试我们可以确信：API 和我们所期望的、最终得到的实际代码一样，可以正常工作。这又是一个优秀的方法。

Jeff 是一个相当好的老师。Jeff 要求我们思考和做练习。他知道如果您和我打算开始学习，那么就必须实践：我们必须做练习。书中每一章都有练习。他建议我们思考这些练习，并且完成和我们不熟悉的主题相关的练习。在键盘上敲入代码，让这些练习深入您的大脑，并且在完成这些有趣的例子的时候，参照 Jeff 的指导。您会非常满意所做的工作。

本书至少可以带给您三个好处：学会可能此前并不了解的 Java 知识，即使您不是一个 Java 初学者。学会在多种情况下，如何应用测试驱动开发（某些情况下，您可能发现自己很难找到合适的方法）。最后，在掌握本书的技巧之后，您可以把这种颇有价值的技术放进自己的专业工具箱里。

我很喜欢这本书，而且发现它非常值得一读。希望您也会喜欢这本书！

Ron Jeffries

www.XProgramming.com

Pinckney, Michigan

November 2, 2004

致 谢

非常感谢所有想象力丰富的程序员，以及正在学习中的程序员，给予了本书很多富有洞察力的意见。

感谢我的前任雇主——Object Mentor 的 Bob Martin，Bob 是一个伟大的导师，同时也是该丛书的编辑。Bob 早期的复审，使得本书更加精确。非常荣幸我的这本书能够成为 Robert Martin 系列的一部分。也非常感谢 Ron Jeffries 为拙作说了很多赞扬的话。

ThoughtWorks 的 Jeff Bay 做了非常重要的工作，他给了我想要的东西——残酷但是真实的反馈。Jeff 负责提供每节后面的练习。

非常感谢 Steve Arneil、Dave Astels、Tim Camper、Dan D'Eramo、Michael Feathers、Paul Holser、Jerry Jackson、Ron Jeffries、Bob Koss、Torri Lopez、Andrew Masters、Chris Mathews、Jim Newkirk、Wendy、David Peterson、Michael Rachow、Jason Rohman、Tito Velez 和 Jeroen Wenting。他们校对了这本书，提出了大量的修改意见。

特别感谢 Paul Petralia 以及他在 Prentice Hall 的全体员工，是他们使我尽可能顺利地完成此书。

再次感谢我的姐姐 Christine Langr，她提供了本书封面的图片，并且帮助我解决了很多设计问题。

同时，感谢 San Antonio 的 Jim Condit 为我提供了 Nerf play 和住所。并且，感谢我的妻子 Kathy Langr 提供了例子（而且陪伴我度过整个艰苦的写作过程）。

引言

我是一名软件工匠¹。我的开发生涯中，大部分时间都在致力于快速构建解决问题的方案。与此同时，我也竭尽全力确保我的代码经过了细心的雕琢。我一直为完美的代码而努力，但是我知道这个目标是无法达到的，特别是为了发布产品，公司持续给你压力的时候。面对自己每天写的代码，我会有一丝自豪，然而当我回过头看昨天的代码，我经常 would 感到困惑：“昨天我到底是怎么想的？”这种挑战使我不断改进自己的代码——永远热切的盼望着下次做得更好一点，比上次痛苦更少一些。

本书为您描绘出一条成功学习和掌握 Java 开发的捷径。本书基于我在教授编程和自学一门新的编程语言中逐渐总结出来的方法：测试驱动开发（TDD:test-driven development²），这是一种引入大量底层反馈的技术。这些反馈使您很快看到行动的结果。使用 TDD，您将学会如何雕琢您的 Java 代码，从而得到稳定的面向对象设计和高可维护、高质量的系统。

我已经使用 TDD 的方法开发软件超过四年，然而时至今日我仍然惊讶于这种方法带给我的好处。TDD 提高了我编写代码的质量，而且每周都教给我新的东西，使我变得更富有效率。我也曾经使用 TDD 的方法在我的公司和 Object Mentor（该公司一直用这种方法教授课程）编写和教授编程语言。

在学习 TDD 之前，我花费了超过十五年的时间采用“传统”方法来学习、使用和教授开发语言，没有使用测试驱动的开发。学生编写并执行一个例子程序，通过程序的输出获得反馈。尽管这是一种有效的方法，但是我过去的经验表明这种方法会导致学生对语言细节不能很好的理解。

◀ 1

与之相对照的是，TDD 所带来的大量迅速的反馈会持续地强迫您编写正确的代码，并且很快指出有问题的代码。经典的“编码—执行—观察”的方法可以提供反馈，但是反馈的速度要慢得多。不幸的是，当前的教学将其作为一种占主导地位的方法。

已经有人尝试了革新的方法来教学。上个世纪九十年代，Adele Goldberg 开发了一种用来教授年轻学生的软件——LearningWorks。这种软件通过让用户动态执行小段代码来直接操作可视化对象，用户可以立刻看到执行的结果。最近有一个 Java 培训工具采用了类似的方法，该工

¹ See [McBreen2000].

² 译注:后面的内容中多数地方直接使用 TDD 表示测试驱动开发。

具可以让学生执行小段代码，在活动对象上产生可视的结果。

这些方法的问题是局限在了学习环境中。一旦您完成了培训，您依然必须去了解在不使用这些受限工具的情况下，如何从头构建您自己的系统。通过使用 TDD，您被教会了一种不受限制的技术，您可以继续将其应用在您的职业软件开发生涯中。

本书尽最大可能采用了面向对象的方法，学习 Java 的困难部分在于需要“循序渐进”。为了写出一组有实际意义的类，您必须掌握的 Java 语言的最小集合是什么呢？

大多数书籍用“Hello world”作为第一个 Java 例子程序，不过这个程序似乎并不适合作为最初的例子：`class Hello { public static void main (String[] args) { System.out.println ("hello world"); } }`。这个简短的程序至少包含了一打您必须要学习的概念。更糟的是，除了这一打概念，至少还有三个您最好马上掌握的非面向对象的概念。

在这本书中，您将学到正确的从头开始编写代码的方法，之后您将会回过头来完全理解“Hello world”³。使用 TDD，您可以很快编写出出色的面向对象的代码。在学习之初您仍然会有很大的障碍需要去克服，但是 TDD 使您免于在一开始就必须去理解一些非面向对象的概念，例如静态方法和数组。您可以适时的掌握核心的 Java 概念，但是您最初的重点应该放在对象上面。

本书向您呈现出一种与以往截然不同的解决问题的方法。它允许您假定从来没有一种叫 C 的语言，C 是 Java 语言的基础，存在了有大约三十年的时间。C 语言给 Java 语言打上了自己的烙印，妨碍您构建出色的面向对象的系统。使用本书，您可以在理解 Java 语言中这些来自 C 的遗产之前，学到正确的构建面向对象系统的方法。

2

预期的读者

我为想把 Java 作为第一门语言的编程初学者设计了本书；这本书同样会对那些熟悉 TDD 但对 Java 不了解的程序员有所帮助。有经验的 Java 开发者会发现，本书向您展现了一种崭新的，更好的实现目标的方法。

本书涵盖了 Java 2 标准版 (J2SE) 5.0。

Sun 提供了许多类库和 API (应用程序编程接口)，这些类库或者 API 增强了核心 Java 语言的功能。例如：JMS (Java 消息服务) 提供了标准的基于消息解决方案的定义。EJBs (企业 Java Beans) 提供了一种针对大型企业应用、构建基于组件的软件的方法。JDBC (Java 数据库连接) 提供了与关系型数据库交互的标准接口。许多这样的高级 APIs 集合在一起，被称为 J2EE (Java

³ 不要着急，尽管如此，为了保证您可以编译和执行 Java 代码，您实际上还是以“hello world”程序开始。但是您不必马上完全理解它。

2 企业版)。很多 APIs 需要一整本书来阐述。现在已经出版了许多有关 J2EE 的书籍。

本书只针对一小部分扩充 APIs 进行了入门介绍。本书用敏捷 Java 向您展现大多数企业应用都会广泛使用的技术，比如日志、JDBC，以及 Swing。某些内容（例如日志）会带给您大多数应用都需要掌握的知识。还有些内容（例如 Swing 和 JDBC）会让您对该技术有基本的理解。所有的内容会提供给您起步所必须的知识，并且会告诉您到何处去寻找更多的信息。

假如您正在开发移动应用，您或许正在使用 J2ME（Java 2 微型版）。J2ME 是一个面向资源限制型环境的 Java 版本，例如手机。相对 J2SE，J2ME 有很多显著的限制。本书不针对 J2ME 作专门的讨论，但是，本书讨论的大多数 Java 核心技术和概念同样适用于 J2ME 环境。

在使用任何 Java 技术扩展之前，您必须理解 J2SE 中提供的核心语言和开发库。本书将帮助您掌握这些知识。

3

本书所不能提供的

本书不能对 Java 语言的每个方面都进行详尽地论述。它提供了一种敏捷学习 Java 语言的方法。不是授之以鱼，而是教您如何钓鱼，以及在什么地方发现鱼。本书将教会您绝大多数的 Java 语言的核心概念。的确，当您完成本书的 15 节课程，您将能够编写出有质量的 Java 代码。但是，这一目标要求您了解本书所没有提及的一些语言功能和细微之处。

熟悉所有 Java 语言细节的方法是仔细阅读 Java 语言规格说明(JLS)。您可以从这里得到 JLS 的第二版：<http://java.sun.com/docs/books/jls>。第二版的 JLS 涵盖了 J2SE5.0 之前的版本，但不包括 J2SE5.0。我写这本书的时候，JLS 第三版正在编写之中，您可以从这里得到它的维护检查版本：http://java.sun.com/docs/books/jls/java_language-3_0-mr-spec.zip。

如果您要深入了解扩展 Java 类库，Java API 文档和源代码是最好的材料。

本书不是一本认证指南，它不能帮助您通过认证考试。一本好的认证教材会教您如何应付考试。本书会帮助您破译代码质量低劣的原因，教会您如何编写专业的 Java 代码。

本书不会试图去溺爱你。学习编程是一种很大的挑战。编程包括思考和解决问题——不是容易到连白痴和傀儡都能够胜任的。我努力避免在本书中侮辱您的智慧，也就是说，我努力使全书更有趣和易于阅读。它是您和我之间的对话，更是在您将来职业开发生涯中和我的对话，同时它也是您和您的计算机之间的对话。

TDD 不承诺

对 TDD 有经验的开发者也许注意到他们的方法和我在本书中的方法有风格上的不同。有很多种方法来实践 TDD。没有一种技术是完美的，或者绝对正确的。按照对您最有利的方式去做，

4

按照最有意义的方式去做，只要不违背本书中的基本原则。

读者一定会发现需要改进代码的地方⁴；即使是初学者，也请不要犹豫，告诉我您看到的您不喜欢的代码。把您的建议发送给我，我会在下一版中修正。您能改进几乎所有的代码或者技术。

在您学完第 1 课后，会发现大量的测试，好像这些代码是一气呵成，可事实上并不这样。每个测试都是由逐个断言构建而成，采用了比这本书所能忍受的还要小的增量一点一点完成的。请您一定要记住，使用 TDD 编写代码的时候，最重要的是采用尽量小的步伐并得到经常的反馈。当我说“小”的时候，我的意思是非常小。如果您认为您正在采用小的步伐，请尝试采用更小的步伐。

如何使用这本书

本书的核心包括 15 节课程，每一课大约 30 页。您将像幼儿蹒跚学步一样，开始您的 Java、TDD、面向对象的旅程，结束的时候您将拥有扎实的基础以从事专业 Java 开发。

15 节的核心课程是循序渐进的。您从第 1 课开始，完成每一课之后再学习后面的课程。一旦您学完了所有核心课程，您将对如何编写健壮的 Java 代码有深刻的理解。

假如您没有完成 15 节核心课程，就不能假设您知道如何编写优秀的 Java 代码！（即使您完成了这些课程，您依然不能算是一个专家）。每一节课程都基于前面的课程。假如您在完成所有课程之前就停止您的脚步，您将缺乏完整的理解，这也许会导致您写出拙劣的代码。

每一节课程的开始，都有本节所要讨论主题的一个概要。后面紧接着详细的论述。我用文字介绍每一个语言特性，用测试代码详细说明它。我用相应的代码实现来展现每一个语言特性。点缀其间的是关于 TDD 技术、面向对象、优秀开发实践的讨论。

5

此外，我提供了 3 节附加课，以覆盖更多的 Java 主题。其中的两节课程介绍 Swing，这是一个专注于用户界面开发的 Java 工具。这两节课程给您提供充分的信息，足以帮助您开始用 Java 构建健壮的用户界面程序。但是我最大的愿望是给您一些怎样用 TDD 构建它们的思路。第三节课简要介绍了一些大多数 Java 程序员都想了解的 Java 主题。

为了最有效的学习，您应该输入并执行本书中的每一段测试和实现代码。您可以下载这些代码（请看下一段），但是我强烈建议您亲自输入每一段代码。正确运用 TDD 依赖于您清晰地理解测试和编码之间来回转换所包含的韵律。假如您只是下载代码然后去执行它，您将几乎不会学到太多。键盘的触觉似乎传递了大量的知识。

但是，我又算什么，能要求您按照指定的方式去学习？您可以选择从这个地址下载代码：

⁴稳定的成对持续的结对编程组合会有帮助。

<http://www.langrsoft.com/agileJava/code>。我按照课程章节组织好了这些代码，也提供了代码的执行结果，也就是课程结束的时候它的状态。这将易于您在任何一点拾起课程，特别是很多例子一直沿用到了后续的章节。

练习

15 节核心课程的每一节要求您一点一点构建一个大学学员信息系统。我选择这个普通的主题有助您增量开发和扩展现有代码。每一节课程的最后都有若干练习。不同于学员信息系统，这些练习由 Jeff Bay 提供，要求您一步一步构建一个象棋软件。

其中的一些练习非常有挑战性，但是我强烈建议您完成每一个练习。通过这些练习，您可以在没有我的帮助的情况下，实战性的学习如何用 Java 解决问题。所有的练习给您又一次深入理解每一节课程的机会。

本书中的约定

代码紧随文字之后，目的是使其成为交流的一部分。假如我说“以下代码”，那就意味您接下来会读到一段代码；而我说“以上代码”，那就意味着代码会出现在相邻的文字上方。 6

代码（下面）用等宽字体显示：

```
this.isCode();
```

大段代码中较少的部分用黑体显示，黑体表明这是当前例子中新增加的代码，或者是需要特别的关注。

```
class B {
    public void thisIsANewMethod() {
    }
}
```

直接出现在文字中的代码，例如 `this.someCode()`，同样用等宽字体。但是类的名字，例如 `CourseSession`，和其余的文字使用相同的字体。

我经常在例子中使用省略号。表明这部分代码存在，但是与当前的讨论或者例子无关。

```
class C {
    private String interestingVariable;
    ...
}
```

```
private void someInterestingMethod () {
}
...
```

交替使用文字和代码来表达思想。例如，也许我会说需要取消工资支票，也可能会说您应该发送一个 `cancel` 消息给 `PayrollCheck` 对象。这种方法有助于您在理解需求和用代码实现需求二者之间建立必要的联系。

类名约定使用单个名词，例如 `Customer`：“您使用 `Customer` 类创建多个 `Customer` 对象”。为了增加可读性，我有时用类名的复数形式代指这些 `Customer` 对象：“这个集合包含了所有从文件系统加载的 `Customers`”。

新术语第一次出现用斜体。绝大多数的术语在术语表中（附录 A）。

贯穿全书，您都将根据规格说明并把它们转换成 Java 代码。按照敏捷过程的传统，我会用非正式的语言来描述这些规格说明。这些规格说明是用户需求，也叫 *stories*：一个 *story* 是对多次对话的约定。您经常为了获取进一步的需求细节，需要和 *story* 的讲述者（有时候称其为客户）进行对话。在本书中，如果一个 *story* 没有理解，请试着往下读，读相应的测试，看其如何解释这个 *story*。

7



我在本书中用一个“讲故事”图标来强调 *stories*。该图标强调口头的非正式的 *stories*。

学习开发艺术的最好途径是和一位有实际经验的人一起工作。您将会发现很多编程的秘密。一些秘密是您为了掌握开发所必须知道的基本概念。其他的秘密向您展现需要密切注意的易犯的错误。您必须赢得这些挑战，记住您所学的，努力成为一名成功的 Java 程序员。



我用桥梁图标来标记这样的关键点（桥梁跨越了易犯的错误）。很多关键点在任何语言中都是有效的，不只是 Java。

8

“敏捷”综述

这一节简要介绍本书中的一些核心概念，包括 Java，面向对象编程，以及测试驱动开发（TDD）。您将会得到下面问题的答案：

- 什么是“敏捷”？
- 什么是 Java？
- 什么是面向对象编程？
- 为什么要面向对象？
- 什么是对象？
- 什么是类？
- 为什么要采用 UML？
- 什么是继承？
- 为什么要测试驱动开发（TDD）？

什么是“敏捷”

这本书的名字是 *Agile Java*。“敏捷 (agile)” 是描述一些软件开发方法学的新的标志性词汇。如果给一个宽松的定义，一个方法就是构建软件的一种过程。目前有一系列方法可以用来构建我们的软件，其中有很多已经被规范化并被命名为某种方法。现在有许多主要的过程被投入应用，或许您已经听说过其中的一些：瀑布、RUP（统一软件过程）、XP（极限编程），以及 Scrum。

随着我们对如何有效工作了解的增多，方法学也在不断的发展。瀑布法是一种古老的方法，该方法提倡软件开发过程中编写大量的文档，严格地事先（例如在项目刚启动的时候）定义需求和系统设计，并且将项目开发分成若干个串行的阶段。该方法的名字来源于这样的图：开发过程自顶向下，从一个阶段到下一个阶段。

尽管瀑布法适合某些场合，但是它的局限性在其他软件项目中会引发严重的问题。用瀑布