

MANNING
覆盖 JUnit 4.8

JUnit
IN ACTION
SECOND EDITION

JUnit 实战 (第2版)

[美] Petar Tahchiev Felipe Leme 著
Vincent Massol Gary Gregory
王魁 译

人民邮电出版社
POSTS & TELECOM PRESS

JUnit

实战

(第2版)

JUnit
IN ACTION
SECOND EDITION

[美] Petar Tahchiev Felipe Leme 著
Vincent Massol Gary Gregory
王魁 译

人民邮电出版社
北京

图书在版编目 (CIP) 数据

JUnit实战 : 第2版 / (美) 塔凯文 (Tahchiev, P.)
等著 ; 王魁译. — 北京 : 人民邮电出版社, 2012. 4
ISBN 978-7-115-27475-5

I. ①J… II. ①塔… ②王… III. ①JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第014947号

版 权 声 明

Original English language edition, entitled *JUnit in Action (Second Edition)* by Petar Tahchiev, Felipe Leme, Vincent Massol, Gary Gregory, published by Manning Publications Co., 209 Bruce Park Avenue, Greenwich, CT 06830. Copyright ©2011 by Manning Publications Co.

Simplified Chinese-language edition copyright ©2012 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Manning Publications Co.授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

JUnit 实战 (第 2 版)

-
- ◆ 著 [美] Petar Tahchiev Felipe Leme
Vincent Massol Gary Gregory
- 译 王 魁
- 责任编辑 杜 洁
- ◆ 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 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 29.25
字数: 640 千字 2012 年 4 月第 1 版
印数: 1—3 000 册 2012 年 4 月北京第 1 次印刷

著作权合同登记号 图字: 01-2009-3551 号

ISBN 978-7-115-27475-5

定价: 79.00 元

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

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第 0021 号

内容提要

本书从认识 JUnit、不同的测试策略、JUnit 与构建过程、JUnit 扩展 4 个方面，由浅入深、由易到难地对 JUnit 展开了系统的讲解，包括探索 JUnit 的核心、软件测试原则、测试覆盖率与开发、使用 stub 进行粗粒度测试、使用 mock objects 进行测试、容器内测试、从 Ant 中运行 JUnit 测试、从 Maven2 中运行 JUnit 测试、持续集成工具、表示层的测试、Ajax 测试、使用 Cactus 进行服务器端的 Java 测试、测试 JSF 应用程序、测试 OSGi 组件、测试数据库访问、测试基于 JPA 的应用程序、JUnit 的其他用法等内容。

本书不仅仅介绍了使用 JUnit 框架测试项目必须掌握的核心概念，还指导读者如何使用 JUnit 框架编写自己的测试用例，并针对在编写代码的过程中如何使得代码可测试给出建议。本书还介绍了基础的软件开发原则，如测试驱动开发（TDD）等，以便指导用户如何使用不同的工具来测试典型 Java EE 应用程序的每一层。此外，本书也提供了几个附录，以帮助读者快速转换到最新版本的 JUnit，并且能够轻松地集成自己喜欢的 IDE。

本书适合于已具有一定 Java 编程基础的读者，以及在 Java 平台下进行各类软件开发的开发人员、测试人员。对于单元测试学习者和编程爱好者来说，本书则具有极高的学习参考价值。

对本书第一版的赞誉

本书是关于单元测试 Java EE 组件的最佳指导手册。如果你正在寻找一些更加令人振奋的内容，那么你可以挑选其他图书中的任意一本，但是当你准备坐下来敲出一些代码时，你就会希望有这么一本书在你身旁。

——JavaRanch.com

我肯定会向任何对测试代码感兴趣的人推荐《JUnit 实战》……它是一本结构清晰、阅读流畅的图书，并且将技术理论和实际应用进行了完美的结合。

——TheServerSide.com

对于想要了解如何正确地构建 Java EE 应用程序的中级 Java 程序员，这是一本必备指南……清晰、简单并且有趣……这是到目前为止我所看到的最好的一本 JUnit 指南……这本书的确详细地介绍了 mock objects 和 stub 的使用，进一步扩展了你对基础软件设计的理解……我强烈推荐它。

——Killersites.com

本书不是一本 JUnit 教程，它深入地展示了 JUnit 的方方面面。它还解释了 JUnit 在软件开发过程的环境中的重要性。在此向广大的 JUnit 初学者和 JUnit 高级用户强烈推荐这本品质精良的图书。

——Denver JUG

阅读了大量的 Manning 图书之后，我发现自己开始以不同的方式来思考问题了，这也促使我成为一名更优秀的开发者，《JUnit 实战》对我而言就是这样一本书。一开始，它打乱了我的开发节奏，因为为了测试我的代码，我不得不对它们进行修改，但是，随后我就发现所做的这些修改使得代码在整体上变得更好。现在我的代码充斥着各种工厂方法和类似的模式。你们这些家伙正在做一些好东西。

——Joshua Smith，一位读者

示例清晰，具有很强的实用性。作者们解决了单元测试 EJB 和 Web 应用所面临的各种复杂问题。他们并没有回避测试这些应用程序所带来的真正问题。

——Wade Matveyenko，一位读者

前言

作为一个屡获殊荣的数学家，我无法容忍平庸。这就是数学所教给我的——不要停止，直到你把它完成，并且不仅要用好的方法，而且要用最好的方法。

当我开始编写软件时，我发现这条原则同样适用。我认识一些同事，他们对自己的工作非常马虎随便，并且我也看到了他们因为自己的疏忽大意而深受其害。他们不耐烦地完成他们的任务，从不担心他们开发的软件质量，更不用说寻找可能的最佳解决方案。对于那些家伙，重用相同的代码就意味着，在任何他们需要相同代码的地方简单地复制和粘贴它即可。我看到了不耐烦地尽快完成任务是如何导致相同的任务一次又一次地被重新打开，因为编写代码时所产生的 bug 和问题。

值得庆幸的是，那类同事已经很少见了。大多数我的朋友都是值得我向他们学习的人。我曾有机会为惠普公司工作，不仅与技术团队合作，而且还与各级项目经理共事，并从他们那里我学到了提供高质量软件产品的秘密。

后来，我加入了 Apache 软件基金会 (Apache Software Foundation, ASF)，在那里我有机会与一些全球最顶级的软件开发人员一起工作。我研究了他们编写代码、编写测试用例以及在我们之间分享信息的最佳做法和习惯，并且我可以为惠普的一些最大客户将我学习到的知识和技能应用到各个项目中。

我逐渐对确保软件产品的可持续质量的问题发生了兴趣。然后，我在 2008 年春天遇到了 Vincent Massol 和 Felipe Leme。我曾在 ASF 与他们俩一起为了 Cactus 框架工作。Vince 建议我为他 5 年前撰写的畅销书编写最新的修订版。虽然这个计划是确定的，但是我还需要一些兴趣相投的伙伴来帮我一起实现它。这时候我联系了 Felipe Leme 和 Gary Gregory。他们俩都愿意帮忙编写一些章节。

在这之后事情的进展就顺利多了，带着修订 Vince 作品的最初目的，我们花了一年半的时间来编写新版本。如果有人在一开始就告诉我这项工作会有多么艰难，我肯定以为他在吓唬我。这就是为什么我觉得我需要向 Manning 团队表示衷心感谢的原因——他们使得整个写书的过程轻松了许多。

现在这本书已经出版，并且你可以把它捧在手中，我希望你会喜欢它。为了完成这本书，我们经历了一段艰难的旅程，但是现在它终于完成了。我知道你会从我们的书中学到很多新的东西，因为我确信你会提高你的软件质量——你已经迈出了第一步。

Petar Tahchiev

关于本书

欢迎阅读《JUnit 实战》第 2 版！如果你拿起这本书，那么你就有可能是一名关注软件产品质量的 Java 开发人员。或许你使用过早期版本的 JUnit 框架，或许你使用过其他的测试框架，亦或许这是你第一次接触软件测试。无论你之前拥有什么样的经历，你都可能对提高软件过程和软件质量感兴趣。本书旨在帮助读者掌握软件测试的基础知识和相关技能。软件测试包括了许多解决具体任务的项目、测试应用的不同组件和层。在软件测试中，最核心的角色是 JUnit 框架。大约 10 年前，Erich Gamma 和 Kent Beck 编写了这个框架，事实上它已经成为 Java 测试领域的测试标准。JUnit 框架最新的 4.x 版本不仅仅是 3.x 版本的修订版，还带来了更多的变化。如果你还没有听说过 JUnit 框架，那么根据书名，你可能会期待找到一个行之有效的老框架的新版本。但事实并非如此。不同于 JUnit 的早期版本，4.x 版本中引入一种新的方法并重新编写了整个框架。因此，也就有必要对本书的第一版进行修订了。

在《JUnit 实战》第 2 版中，我们介绍了使用 JUnit 框架测试项目必须掌握的核心概念。但是，这并不是本书全部的内容！本书不仅教用户如何使用 JUnit 框架编写自己的测试用例（test case），而且还针对在编写代码的过程中如何使得代码可以测试给出建议。本书还介绍了基础的软件开发原则，如测试驱动开发（TDD）等。它也将逐步指导用户如何测试每一个典型 Java EE 应用的每一层，例如，前层，使用诸如 Selenium 和 JSFUnit 此类的扩展工具；业务层，使用诸如 Cactus、mock objects 和 stub 此类的工具；最后的数据库和 JPA 层，使用诸如 DBUnit 之类的工具。

本书分为几个部分，其目的是按照由浅入深、由易到难的顺序对 JUnit 进行讲解。第 1 部分包括了头几章，介绍了 JUnit 框架的技术背景，高度概括了其构架，并展示了一个非常简单的 HelloWorld 示例应用程序，以帮助读者搭建和运行自己的测试环境。在简短介绍 JUnit 后，我们又利用几章的篇幅逐一介绍了所有重要的概念和框架组件，并深入讨论了每一个组件的功能。

本书的第 2 部分介绍了不同的测试技术：mock 方法和容器内方法，还介绍了一

些新工具，以便创建我们需要的模拟对象。本书的第 3 和第 4 部分详细讨论了第三方工具/JUnit 扩展，我们使用它们来测试不同应用的不同层。此外，本书也提供了几个附录，以帮助读者快速转换到最新版本的 JUnit，并且能够轻松地集成自己喜欢的 IDE。

本书结构

第 1 章帮助你快速入门。本章通俗易懂地介绍了测试是什么、如何有效地进行测试、如何编写自己第一个测试用例。通过对这一章的阅读，你会充分意识到测试是自然而然产生的，它总是在开发期间发生。

第 2 章深入讨论了 JUnit 构架及其构成。在这一章中，介绍了大部分的 JUnit 常用功能。

在第 3 章中，我们开始创建一个应用程序实例。你将掌握几种设计模式并使用它们创建应用程序。在后面的章节中，我们还将阐述如何利用第 2 章介绍的 JUnit 功能有效地测试应用程序。

第 4 章着眼于几个重要方面：单元测试的必要性、已有软件测试的不同特点、不同测试类型之间的区别。我们也针对如何搭建不同的开发和测试环境给出了有效的建议。

在第 5 章中，我们将讨论测试质量的话题。我们继续探讨几个关键问题，如如何改善测试、如何提高测试覆盖面、如何设计应用程序构架以便使应用程序更易于测试。本章的最后简要地介绍了测试驱动开发（TDD）方法。

第 6 章详细介绍了 stub 技术，通常在无法获得系统资源时使用这种技术来模拟系统资源。在这一章中，我们提供了一个示例：通过使用 Jetty 嵌入式 servlet 容器来替换 servlet 容器。

第 7 章阐述了另一种在测试中加入模拟对象的方法：mock objects。当你处于封闭的 API 下进行编程，并且你无法修改或实例化可用的资源时，这种技术就非常有用。在这一章中，我们提供了一个示例：使用两大最主流的框架 EasyMock 和 JMock 来模拟一个 Servlet 并测试它。

第 8 章简要地介绍了当缺少重要的系统对象时我们所能采用的最后一种方法：容器内测试（in-container testing）。我们在这一章中介绍这种方法，以便我们在第 14~16 章中能够展开这个话题，讨论真实环境下的容器内测试示例。作为本书第 2 部分的总结章，第 8 章中也比较了前面章节所讨论的几种方法：stub、mocks 和容器内测试。

第 9 章是本书第 3 部分的开篇之章。在这一部分中，我们的重点是 JUnit 与不同构建框架的集成。特别是在这一章中，我们介绍了 Ant 构建框架。我们告诉你如何自动执行测试，以及如何使用执行结果生成有效的、美观的报告。最后，我们使用 Ant 框架运

行前面章节中的一些示例。

第 10 章继续讨论引入构建框架并使 JUnit 与之集成的方法，但在这一章中我们来看一下 Maven。

第 11 章专门介绍持续集成（Continuous Integration，CI）理论——以一种持续的方式构建项目并执行测试，以确保没有任何变化会影响项目。我们进一步介绍了两大实践持续集成的主流软件项目：CruiseControl 和 Hudson，我们也借此机会，将前面章节中的一些示例导入到这两个工具中，设置并执行它们。

从第 12 章开始进入本书的最后一部分。这一部分讨论了各种 JUnit 扩展，借助它们可以增强测试框架，从而就能完成一些通常不可能完成的具体任务。在这一部分中，我们还介绍了典型应用程序的所有层，并解释了如何测试这些层。第 12 章重点关注 Web 应用程序的表示层，所以我们介绍了 HtmlUnit 和 Selenium 工具，并明确说明了如何使用它们。

第 13 章继续关注 Web 应用程序的表示层，但在这一章中，我们将聚焦于最难的话题之一：Ajax。我们不但详细介绍了 Ajax 是什么、为什么它很难测试，还阐述了各种测试方案。最后，我们介绍了 JsUnit 项目，并就测试 Google Web Toolkit（GWT）应用程序给出了一些特别的建议。

第 14 章探讨了另一种测试表示层的方法：即已在第 8 章中提到的容器内测试（in-container testing）。为此，我们介绍了有史以来的第一个容器内测试框架：Apache Cactus 项目。

第 15 章介绍了几种特别适用于测试 JSF 应用程序的方法。这一章详细说明了如何使用另一种称为 JSFUnit 的新兴工具——一种新的容器内测试框架，它基于 Apache Cactus，专门用来测试 JSF 应用程序。

第 16 章是专门为那些对 OSGi 应用程序感兴趣的读者而准备的。在这一章中，首先简单地介绍了 OSGi 的概念，然后介绍了 JUnit 的扩展 JUnit4OSGi，并提供了几种使用模拟和容器内测试来测试 OSGi 应用程序的方法。

从第 17 章开始的最后 3 章主要讨论了数据库测试。在第 17 章中，我们全面详细地介绍了一个称为 DBUnit 的项目。我们还深入剖析了几种测试数据库的方法，无论采用了什么持久化技术。

第 18 章揭开了 JPA 测试的所有秘密：测试多层应用程序和 JPA 持久层应用程序。

第 19 章是本书的最后一章。在这一章中，我们讨论了几种提高测试效率的方法。我们还介绍了一个新项目来帮助你测试 Spring 应用程序：Unitils。

本书约定

以下印刷约定适用于全书：

- Courier 字体用于表示所有的代码列表;
- Courier 字体用于正文中表示某些代码;
- 黑体用于强调和介绍新术语;
- 注解用于代码中进行内部注释, 突出显示重要的概念或代码区域。如果注解带有编号 (如 ❶), 那么其编号将随后出现在正文中。

此外, 在代码列表中, 你可能偶尔还会看到**粗体代码**。

- **粗体代码**——这里使用粗体有两个目的: 一是突出显示一些 Java 关键字 (便于读者阅读), 二是突出显示两个或多个代码列表之间的不同。

代码下载

读者可以通过 Manning 公司网站上的图书主页中的链接 www.manning.com/JUnitinActionSecondEdition 或 www.manning.com/tahchiev, 下载到本书相关的源代码。该下载页面包含了一个目录结构, 列出了不同章节对应的所有子模块。每个子文件夹包含了一个用来编译和打包的构建脚本, 读者可以执行与之相关的测试。关于如何安装应用程序的说明已包含在下载文件中的 README 文件中。

关于源代码, 有几点要特别说明一下。起初我们想采用一个大型应用程序, 可以全面展示应用层中的各种测试方法。后来, 我们意识到使用这样一个大型应用程序所要面临的各种困难, 于是我们转而采用目录结构表示法, 每一章都拥有与之对应的源代码示例。全书的源代码示例根据章名被划分到不同的子文件夹中。所有的源代码示例都包含一个 Maven 构建脚本, 并且其中有一些源代码示例还包含了一个 Ant 构建脚本。为了运行书中的所有示例, 读者需要在计算机中安装 Maven2。

作者在线交流

购买《JUnit 实战 (第 2 版)》也就意味着你获得了免费访问 Manning 出版公司所运营的一个私有论坛的权利, 在这个论坛上你可以对本书做出评价, 可以提出技术问题, 并从作者和其他读者那里获得帮助。你可以从 www.manning.com/JUnitinActionSecondEdition 注册并访问该论坛。这个页面提供了注册后如何访问论坛的信息, 也说明了你可以获得什么帮助以及论坛的行为规则。

Manning 对读者的承诺是, 提供一个让读者之间以及读者和作者之间得以有效沟通的场所, 但并不承诺作者们的参与程度, 作者们对本书论坛的参与仍然是自愿的 (无偿的)。我们建议你向作者们提出一些具有挑战性的问题, 这样他们才会有兴趣经常参与。只要本书尚未绝版, 读者就可以通过出版商的网站访问作者在线 (Author Online) 论坛

和以前讨论的存档。

关于书名

In Action 丛书的特点是理论（介绍、概述等）与实践（教你如何动手的各种示例）相结合，其目的在于帮助读者学习和记忆。根据认知科学方面的研究，人们记住的事物往往都是他们在自我探索过程中发现的事物。

虽然 Manning 公司内部没有一位是认知科学家，但是我们深信，要使得学习的效果永久，就必须经历探索、实践以及复述学到了什么这几个阶段。人们理解并记住新事物，这也就是说，只有在积极探索新事物之后人们才能掌握它们。人类在实践中学习。“实例驱动”是 In Action 丛书的一个基本要素。它鼓励读者尝试，实践新代码，探索新构想。

关于本书的书名，还有一个更现实的理由：我们的读者都很忙。他们因工作需要或者为了解决问题而寻求书籍的帮助。他们需要的图书是这样的：不仅可以让他们快速上手，而且还可以让他们在想学的时候只学他们想学的内容。他们需要可以在实践（In Action）中帮助他们的图书。该丛书就是专门为这类读者设计的。

关于作者

Petar Tahchiev 是一名软件工程师，担任了 Apache 软件基金会（Apache Software Foundation）的 Jakarta 项目管理委员会（PMC）成员。多年来，他一直是 Jakarta Cactus 的开发主力，也参与了 Apache Maven 开发团队。此外，他也是 JCP 成员和 Bulgarian Java User Group（BGJUG）的负责人，经常在 OpenFest、ApacheCON、CommunityONE 和许多其他会议上发表演讲。Petar 在保加利亚出生并长大，并以优异的成绩毕业于索菲亚（Sofia，保加利亚首都）大学的数学系。他曾在德国和荷兰的公司（如 Unic、惠普）任职多年，现在他回到可爱的索菲亚，重点发展他自己的公司 Phamola，他的公司主要为客户在如何通过技术赢得优势方面提供帮助和建议。Petar 撰写了本书的第 1~11 章、第 14~16 章以及附录 A~D。

Felipe Leme 是一名软件工程师，非常热爱 TDD（Test-Driven Development）、Java 和电脑。11 岁时，他得到了他人生的第一台电脑，1996 年他开始学习 Java，2000 年他编写了他的第一个 JUnit 测试用例。自从 1997 年在 Campinas 州立大学（Unicamp）获得计算机工程学士学位，他就几乎只使用 Java 了，同时他也在许多方面为社区做出了贡献：开源项目（如 DbUnit）的提交者（Committer）、会议（如 JavaOne）演讲者、JCP 个人成员、java.net 上的博客和撰稿人。Felipe 撰写了本书的第 17~第 19 章。经常往返于巴西的圣保罗和美国的加利福尼亚之间后，他终于定居在旧金山的湾区（Bay Area），与他的妻子、孩子们和寄居蟹生活在一起。

Vincent Massol 在利用几年的晚上时间创建了 Jakarta Cactus 和 Codehaus Cargo 并参与 Apache Maven 开源项目后，现在正享受着全职开发 XWiki 的乐趣，XWiki 是一个提供最顶级企业维基服务的开源项目。Vincent 也是 XWiki SAS 公司的 CTO，这家公司主要提供 XWiki 开源项目方面的服务。他是本书第一版的主要作者。Vincent 生活在法国巴黎，可以通过 www.massol.net 在线联系他。

Gary Gregory 拥有超过 20 年的面向对象语言的经验，包括 Smalltalk、Java、XML、各种数据库技术。Gary 曾任职于 Ashton-Tate、ParcPlace-Digitalk 和其他几家软件公司，

目前他在 Seagull 软件公司从事现有系统集成 (Legacy Integration) 应用程序服务器的开发。他是 Apache 软件基金会和 Apache Jakarta 项目管理委员会的积极参与者，定期为各种 Apache Commons 项目贡献力量。Gary 在法国巴黎出生和长大，获得洛杉矶加利福尼亚大学的语言学和计算机科学学士学位。Gary 撰写了本书的第 12~第 13 章和附录 E。他和他的妻子、儿子伴随着高尔夫俱乐部、各类冲浪板生活在洛杉矶。可以通过 <http://www.garygregory.com> 联系他。

关于封面图片

《JUnit 实战（第 2 版）》封面上的图片名叫“Burco de Alpeo”，选自 1799 年初版于西班牙马德里的一本各地服饰习俗汇编。同样的图片出现在本书的第 1 版封面上，但在接下来的几年里，尽管已要求我们的第 1 版读者来帮忙，我们还是没能成功地破译图片名的准确含义。欢迎读者在作者在线（Author Online）论坛中发布任何关于第 2 版的新建议。

那本汇编的扉页上写道：

Coleccion general de los Trages que usan actualmente todas las Naciones del Mundo desubierto, dibujados y grabados con la mayor exactitud por R.M.V.A.R. Obra muy util y en especial para los que tienen la del viajero universal

我们尽可能照字面意思翻译：

搜集了目前在已知的世界各国中使用的服饰，由 R.M.V.A.R 提供高精度的设计和印刷。本书特别适合那些准备畅游世界各国的人士。

虽然我们对这张图片的设计师、刻版师以及为这张图片手工上色的工人一无所知，但他们在此图中投入的“精度”却是显而易见的，而这只是这本彩色汇编中众多图片中的一张。这些图片的多样性生动地表明了 200 年前世界各地的城镇和区域的独特个性。在那个时候，即使两个地方相隔几十公里，根据不同的服饰风格就能区分出哪些人属于哪些地方。

这本汇编让我们真切地感受到了那个时代（以及任何其他历史时期，除了沟通极为方便的现代以外）的隔离与距离。自那以后，服饰风格已发生了改变。那时各地如此丰富的多样风格，在今天也逐渐模糊同化了。现在通常很难依靠衣着分辨出哪个人居住在哪个洲了。或许，客观地看，我们使用文化和视觉上的多样性换取了个人生活的多样性，或者换取了更丰富也更有精神世界与科技生活。

我们使用基于两个世纪以前的多样化的地区生活来做图书封面，让这本汇编中的图片重现于世，并借此来赞美计算机业的创意、进取和乐趣。

第 1 版前言

到目前为止，测试仍然是人们所能找到的能确保交付的软件正常运行的最好办法。本书是 4 年来在测试领域研究和实践的成果。实践来自我的 IT 咨询背景——我曾先后在 Octo Technology 公司和 Pivolis 公司任职；研究则来自我在晚上和周末所从事的开源开发工作。

自从 1982 年我的早期编程生涯开始，我就对编写能帮助开发者写出更好代码并且提高开发速度的工具感兴趣了。这一兴趣引导我进入了软件咨询、质量改进等领域。这些日子，我在建立持续构建（continuous-build）平台，同时也在探索最佳开发实践。这两者都需要完善的测试集来支持。测试与编码活动越接近，从代码获得反馈就越快速，所以我对单元测试很感兴趣。单元测试与编码活动如此接近，以至于现在单元测试像编写代码一样成为开发活动的一部分。

这个背景使得我参与了以下这些与软件质量有关的开源项目：

- Cactus，用于对 J2EE 组件进行单元测试（<http://jakarta.apache.org/cactus/>）；
- Mock Objects，用于对任何代码进行单元测试（<http://www.mockobjects.com/>）；
- Gump，用于持续构建（<http://jakarta.apache.org/gump/>）；
- Maven，用于构建和持续构建（<http://maven.apache.org/>）；
- Pattern Testing，使用 AOP（Aspect-Oriented Programming）来检查构架和设计规则，这是一个概念验证性质的项目（<http://patterntesting.sf.net/>）。

由于参与了这些项目，也就有了《JUnit in Action》这本书的面世。

没有人想编写糟糕的代码。我们都希望写出可以正常工作的代码——我们都以自己的代码为荣。但是，常常事与愿违。你是否经常听到这样的话：“我们是想要些测试的，但我们在压力下没有足够的时间去写。”“我们一开始是写了单元测试的，但两个星期后我们就没有动力继续下去了，再往后我们就放弃了编写单元测试。”

本书将向你展示为了编写高质量的代码所需要掌握的各种工具和技能。本书手把手

地教你如何高效地运用工具，避免常见的陷阱。本书将使你具备编写可以正常工作的代码的能力，还将帮你把单元测试引入日常开发活动，并建立起一套步骤，使你可以按部就班地写出稳健的代码。

最重要的是，本书向你展示了如何控制你的软件的熵（译注：指随着软件的增长或修改，软件变得复杂和紊乱的程度），而不是反过来被它控制。我想起了拉丁作家 Lucretius（公元前 94 年至公元前 55 年）在他的《On the Nature of Things》作品中写到的一些诗句（我就不列出拉丁文原文了）：

It is lovely to gaze out at the churning sea from the safety of the shore when someone else is out there fighting the waves, not because you're enjoying their troubles, but because you yourself are being spared.

这正是当你意识到自己被精良的测试集“武装”起来时你将体验到的感觉。你会看到，别人还在蹒跚挣扎；并且你会感到欣慰，因为你拥有测试，可以用来阻止任何人（包括你自己）破坏你的应用程序。

Vincent Massol