

Test-Driven Development

A Practical Guide

测试驱动开发 实用指南

[美] David Astels 著
Ron Jeffries 序
崔凯 译



本年度美国
《软件开发》杂志震撼大奖得主

- 最具实用性的测试驱动开发指南: 问题、方案、代码
- 使用 Java 和 JUnit 测试框架从头开始创建项目
- 针对 C++、C#/.NET、VB6、Python 等的 TDD 框架
- 面向所有对 TDD 感兴趣的开发人员和项目经理



Test-Driven Development

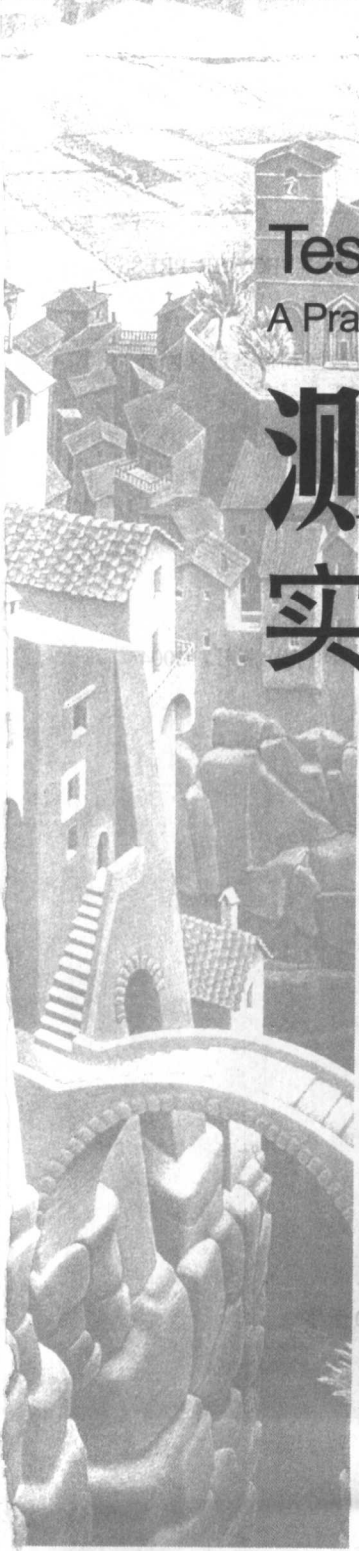
A Practical Guide

测试驱动开发 实用指南

[美] David Astels 著
Ron Jeffries 著
崔凯 译



中国电力出版社
www.infopower.com.cn



ISBN 7-5083-2432-X
定价: 28.00元
2004年9月北京第1版
2004年9月第1次印刷
印张: 31
字数: 880千字
印数: 10004
地址: 北京市三里河路6号
电话: (010) 88212918
邮编: 100044
中国版本图书馆CIP数据核字(2004)第067132号
ISBN 7-5083-2432-X
《测试驱动开发: 实用指南》(美)艾斯特(Astels, David)著, 崔凯译. — 北京: 中国电力出版社, 2004
I. ①测... II. ①艾... ②崔... III. ①测试驱动开发—指南 IV. TP311.52-62
图例的出版号(CIP)数据

Test-Driven Development: A Practical Guide (ISBN 0-13-101649-0)

David Astels

Copyright © 2004 Prentice Hall PTR.

Original English Language Edition Published by Prentice Hall PTR.

All rights reserved.

Translation edition published by PEARSON EDUCATION ASIA LTD and CHINA ELECTRIC POWER PRESS,
Copyright © 2004.

本书翻译版由 Pearson Education 授权中国电力出版社在中国境内（香港、澳门特别行政区和台湾地区除外）
独家出版、发行。

未经出版者书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 Pearson Education 防伪标签，无标签者不得销售。

北京市版权局著作权合同登记号 图字：01-2004-2529

图书在版编目（CIP）数据

测试驱动开发：实用指南 /（美）艾斯特尔斯（Astels, D.）著；崔凯译. —北京：中国电力出版社，2004
（开发大师系列）

ISBN 7-5083-2482-X

I. 测... II. ①艾...②崔... III. 软件开发—指南 IV. TP311.52-62

中国版本图书馆 CIP 数据核字（2004）第 067132 号

丛 书 名：开发大师系列

书 名：测试驱动开发：实用指南

编 著：（美）David Astels

翻 译：崔 凯

责任编辑：姚贵胜

出版发行：中国电力出版社

地址：北京市三里河路 6 号 邮政编码：100044

电话：（010）88515918 传 真：（010）88518169

印 刷：北京丰源印刷厂

开 本：787×1092 1/16

印 张：31 字 数：680 千字

书 号：ISBN 7-5083-2482-X

版 次：2004 年 9 月北京第 1 版

2004 年 9 月第 1 次印刷

定 价：58.00 元

版权所有 翻印必究

序言

我写这篇序言的目的就是帮你判断是否应当阅读本书。如果你对改善程序质量和提高编程技能感兴趣的话，那么本书会对你有所帮助。

测试驱动开发（Test-Driven Development, TDD）是一种能够改善程序质量的编程方法。如果你像我一样，采用本书所提供的技术，那么将会发现编写程序更加容易，编写出来的代码更加清晰，而且与以往相比程序的瑕疵也会更少。

我可不是在说测试驱动开发方法就是一种灵丹妙药；恰恰相反，TDD 并不神奇，你自己也能做得到。通过集中精力先编写测试，你就会站在用户的立场上设计程序。通过每次只处理一个测试，你就会创作出切实针对问题的简单设计。随着这些小规模测试程序的开发与完善，你就能够将绝大多数代码瑕疵拒之门外。否则，如果没有这样做，那么就会在不知不觉中引入这些臭虫（bug）。最终，通过留存这些测试，就可以使程序更易于维护，随着时间的推移还可以不断地对程序进行完善。

本书中的测试驱动开发范例非常丰富，而且还提供了一个贯穿整个 TDD 开发过程的十分详尽的例子。书中包含许多小的范例，展示了现有的大多数与 TDD 有关的工具的用法。尽管本书的重点主要集中在用 Java 编写的例子上，但是也为大多数语言提供了采用 TDD 开发的例子。本书没有枯燥的理论，而是使用真实的代码范例来阐述 TDD 的实践方法。

等等！下面还有！作者还很好地介绍了重构（Refactoring）和意图导向的编程（Programming by Intention），并且介绍了模拟对象（Mock Object），这是测试和测试驱动开发中一种功能强大的高级技术。此外，书中还有一节介绍了以测试优先（test-first）方式创建 GUI 的内容，以测试优先方式创建 GUI 是测试驱动开发中较为棘手的领域之一。

你将会在本书中发现有关极限编程（eXtreme Programming）的精辟总结、与敏捷建模（Agile Modeling）有关的介绍以及丰富的与书中论题有关的联机资源列表。

所有这些内容都非常精彩，也是本书值得你购买的原因。本书最有价值、最值得回味的内容就是书中的代码。测试驱动开发是一种我们编程时使用的技术。无论我们在开始编程之前进行了怎样的设计和建模，TDD 都有助于我们提高代码质量。如果你有机会阅读本书并在实践中运用书中所讲授的知识，那么测试驱动开发肯定会对你有所帮助。

与其他许多程序员一样，测试驱动开发也使我编写出了更好的程序。这是一种值得掌握的技术。本书会帮助你（程序员）更上一层楼，这也是我推荐本书的原因。

Ron Jeffries
www.XProgramming.com
Pinckney, Michigan

前言

本书并不是一本有关测试的书籍。

这是一本有关使代码变得简单、清晰、健壮的编程方法的书籍。它可以使代码易于设计、编写、阅读、理解、扩展和维护。

这是一本有关思考、设计、沟通以及编程的书籍。我们最终将会得到一套详尽的¹测试集，这是非常好的副产品。

本书探讨的内容是测试驱动开发（Test-Driven Development, TDD）或测试优先编程（Test-first Programming），不管怎么称呼它，它都是指一种先编写测试，然后再编写让测试获得通过的代码的方法。具体而言，就是以尽可能小的步伐推进：编写足以失败的测试代码，编写足以让测试通过的产品代码，通过重构来整理为使测试通过而引入的混乱代码。

本书着重使用 Java 编程语言，通篇都是 Java 编写的例子。我们假定读者至少对 Java 有中等程度的了解（如果想要亲自动手尝试一下书中的例子的话，还要有一套能够工作的 Java 系统）。大家可以从我的 Web 站点[URL 54]获得范例代码以及其他一些辅助资料。

尽管我们将重点放在 Java 上，但是本书的第 4 部分还是介绍了其他一些重要的针对几种流行编程语言的 xUnit 家族成员。我们选取第 10 章中的第一项任务，分别用各种语言重新予以实现，从中可以比较各种不同的测试框架。

极限编程

测试驱动开发是由 Kent Beck 形式化（formalized）的一种称做极限编程（eXtreme Programming, XP）的敏捷软件开发过程（agile process）的核心部分。XP 很可能是敏捷软件开发过程中最“敏捷”的部分了，它开销极低，繁缛极少，然而却是一种高度自律、非常高效的软件开发方法，对变化有着难以置信的适应能力。

也就是说，为了运用测试驱动开发方法进行开发并从中获益，你不必采用极限编程（XP）。单就测试驱动开发本身而言也是值得的，你的产品代码质量一定会改善。当然如果采用极限编程的话，那么真正熟练地使用测试驱动开发方法对于选用这种开发过程来说才是值得的。

测试驱动开发是极限编程中一项主要的设计工具²。如前所述，产品开发终了所拥有的整套测试集是非常好的副产品。因为有了那些测试，当我们在代码修改前后运行测试并且通过的话，就可以确信自己在此期间没有意外地引入使产品无法正常运行的错误。反之，如果某项测试在代码修改以后无法运行通过，那么我们就知道究竟是哪个地方出了问题，从而能够方便地找出问题并且修正

1 究竟有多么详尽要看你对 TDD 掌握得有多好。

2 另一项是重构。

它。惟一可能造成错误的位置就是在上次测试运行通过以后修改的地方。

所有这一切都意味着由于测试的存在，我们可以安全地使用极限编程的另一种方法：重构（Refactoring）。正如我们将要在第 2 章看到的，重构就是在不改变代码外部行为的情况下对代码的结构进行调整。我们有测试可以保证行为没有被改动。这样你就有了修改（有时这种改动是翻天覆地的）工作代码所必需的自信，而我们最终的代码也会更清晰，可扩展性更好，可维护性更强而且更易于理解。

附录 A 更多地谈到了有关极限编程的内容。要想更多地了解有关这方面的知识，你可以浏览一下参考文献或访问附录 C 中所列出的极限编程联机资源。

谁应该阅读本书？

你应当读这本书吗？我写这篇前言的目的就是要问答这个问题。以前在雅虎极限编程讨论组（XP Yahoo Group）中有一项关于图书的前言应当服务于什么目的的调查。大家总体的意见是通过阅读前言应当能够比较清楚地知道自己是否需要购买或阅读某一本书。希望我在此很好地履行了这一义务。

如果你想采用极限编程，那么就请阅读本书。如前所述，要想能够较好地掌握极限编程，那么花些时间和精力来较好地掌握测试驱动开发是值得的。测试驱动开发在极限编程中处于核心位置，因此较好地掌握了测试驱动开发就可以使整个极限编程过程的效率大为提高。

如果你想编写出更加清晰、健壮、扩展性强而又尽可能精练（而不是臃肿）的代码，那么请阅读本书。

如果你对开始编写代码之前花费数周或数月时间绘制各种设计图例心存异议的话，那么请阅读本书。

最后，如果你想知道如何才能使编程工作再度充满乐趣的话，那么就请阅读本书。

在阅读本书之前，就所应当了解的知识而言，如果你对 Java 至少有中等程度以上的了解，那么会更好一些。如果你以前对其他几种面向对象编程语言（如 Smalltalk、C++、Python 或 Ruby）有较好的掌握的话，那么就可以从本书中汲取到更多有用的知识。

在本书付梓的时候，市面上还有另外一本有关测试驱动开发的书[9]（我知道大家对这本书肯定趋之若鹜）³在本书编写大半的时候，我了解到还有这样一本书也正在编写中，所以作为对那本书的补充则一直是本书所要达到的部分目标。通过阅读那本书，你能够领悟测试驱动开发本质和抽象层面的东西，那本书也包含了足够的实用范例。如果你倾向于那本书，那么我鼓励你先阅读它。现在你手里拿的这本书，顾名思义，是一本有关测试驱动开发的实用指南。本书把重点放在一种语言（虽然不是最好的语言，但无可争辩的是一种非常流行的同时也是对测试驱动开发支持良好的语言）上，不仅介绍了测试驱动开发的概念和原则，同时也对各种工具和技术进行了清晰的阐述。

本书的结构

本书分为四个部分：

3 本书中文版《测试驱动开发》及英文影印版已由中国电力出版社出版发行。——编者

第 1 部分 背景知识介绍 在第 1 部分我们考察了一些与书中的主要内容（即利用 Java 来进行测试驱动开发）有关的话题。我们开始先介绍测试驱动开发，然后分章节讲述有关重构（refactoring）和意图导向编程（programming by intention）的知识。这两种技术在极限编程中也是非常重要的，是测试驱动开发所要求的，同时又是测试驱动开发才使之成为可能的。

第 2 部分 工具与技术 在第 2 部分我们深入探讨了各种对使用 Java 来进行测试驱动开发有用的工具以及如何来使用这些工具。我们先是提供了一篇介绍 JUnit 的教程，JUnit 是 Java 测试驱动开发框架事实上的标准。接着我们探讨了一些 JUnit 的标准或非标准扩展。随后探讨了一些支持使用 JUnit 的工具以及其他一些完全独立于 JUnit 但是能够与之很好配合的工具。这一部分的最后几章考察了某些特定的技术问题和相关工具。

第 3 部分 一个彻头彻尾的采用 TDD 开发的 Java 软件项目 这是一本重在亲身实践的实用指南，所以在第 3 部分，我们围绕如何开发一个真实的系统来展开。这可不是一个随随便便的例子，我们将采用测试驱动开发的方法来开发这个项目。在此过程中，我们将会用到书中前几部分所讲的内容。

第 4 部分 xUnit 一族 JUnit 只是巨大的且正在成长中的程序员测试框架（programmer test framework）家族中的一个成员。我们将在第 4 部分查看一下这个家族中的其他一些成员。我们不会把所有的成员讲解一遍，但是会讲到几种较为流行的编程语言的测试框架。我们针对每种测试框架，根据同一组描述（即需求），使用不同的语言重新编写了程序，这样它们之间也就有了一个很好的比较。具体而言，这几个描述就是 Java 项目中开始的那几个描述。同时这也可以使我们拿各种测试框架与 JUnit 作一比较。

本书有四个附录：

A 极限编程 这个附录简明扼要地介绍了有关极限编程的知识。

B 敏捷建模 该附录概括描述了有关敏捷建模的内容。

C 联机资源 在本书讲解过程中我引用了一些 Web 站点，大家可以在这些站点上找到一些相关的信息或下载资源。该附录提供了附带注解的分类资源列表。

D 练习答案 书中的许多章节都包含练习题。该附录包含了书中的全部练习题和参考答案。

本书所使用的约定

为了便于大家阅读，我在书中使用了不少印刷字体和排版约定，用以区分不同种类的信息。

源代码 本书包含了大量的源代码，采用代码行缩进并以 sans-serif 字体来表示，如下所示：

```
public int getAverageRating() {  
    return totalRating / numberOfRatings;  
}
```

总体而言，在引用某个方法的时候，我们没有把参数也包括在内，而是使用空括号，这样就可以清楚地表示这是一个方法而不是某种其他类型的标识符，例如 aMethod()。

我们一般会省略成块代码中的 package 和 import 语句。

文件系统及主控 I/O 我们使用等宽 serif 字体来表示指代文件系统的内容。这些内容包括文件名（filter.properties）、命令及其输出等：

```
$ java \  
> -classpath bin:/usr/local/java/lib/MockMaker.jar \  
> mockmaker.MockMaker \  
> com.saorsa.tddbook.samples.mockobjects.IntCalculator \  
> >src/com/saorsa/tddbook/samples/mockobjects/MockIntCalculator.java
```

建议与旁注 我使用了多种不同的表现形式来强调那些值得注意或是有趣但因为某种原因不适合在正文中叙述的重要信息。



这本书中穿插了一些小知识，也许你会觉得特别有用，所以我们把这些内容单独分出来，本段内容就是出于这个目的。

我使用旁注来把那些与正文没有直接关系的小段文字分离出来。这些旁注的位置通常位于每一页的顶端。这里就有一个旁注的例子。

旁注样例

这是一段非常简短的旁注的例子；大多数旁注都只有半页或一页的篇幅。

术语 我是在 Smalltalk 下学习有关面向对象的知识的，所以从开始起我所使用的就是 Smalltalk 的术语。我列出了几个我所使用的术语以及它们在 Java 和 C++ 中对应的含义，以免你对 Smalltalk 不熟悉：

实例变量 (instance variable) 这种变量的作用域是对象。每个对象都有该变量的独立拷贝。
(Java: 域或成员变量; C++: 数据成员。)

类变量 (class variable) 这种变量的作用域是类。类的所有实例都共用该变量的一个拷贝。
(Java: 静态域; C++: 静态数据成员。)

方法 (method) 类的函数成员。(Java: 方法; C++: 成员函数。)

把消息发送给一个对象 这是一种更为抽象的指代调用对象方法的表达。

发送者 发送特定消息的方法，也就是调用特定的方法（通常称做方法的引用）。

致谢

一篇没有向促成本书出版的人们致谢的前言是不完整的。作为作者，你就好像站在金字塔尖之上，大家以各种各样的形式向你提供帮助（这样本书才可能出版）。借此机会我要向所有帮助我的人表示感谢，并向所有我知道姓名的人点名致谢。

感谢 Kent Beck，是他让测试驱动开发和极限编程家喻户晓——至少在我们家是这样的，感谢他对本书的支持。

感谢 Miroslav Novak，最早是他引发了我对这种崭新的编程方式的兴趣，这在当时还只是由一群聪明人在一种称做 Wiki 的站点上讨论的内容。也许从编程时间上讲 Miroslav 还属于晚辈，但是有时不得不承认我从他那里学到了更多的东西。

有几件事情要感谢 Patrick Wilson-Welsh：当我在细节问题上裹足不前时，Patrick 总是会提醒我

要着眼于大的方面。他是一位作者所能找到的最好的讨论伙伴和文字编辑。感谢他鼓足勇气离开华盛顿特区过惯了的生活，搬迁到“偏僻的”的加拿大，成为我的共同创始人和首位见习生。

感谢《The Pragmatic Programmers (程序员修炼之道)》[URL 55]一书[25]的作者 Dave Thomas，感谢他让我使用他为《The Pragmatic Programmers》一书编写的 LaTeX 宏。那本书的布局和排版让人赏心悦目，而该书也是让我的编程思想发生转折的催化剂。

与《The Pragmatic Programmer》有同等震撼力的就是 Pete McBreen 所著的《Software Craftsmanship (软件工艺)》[34]。我其实是一字不落地读完了此书。Pete 就软件是一种手工艺品进行了精彩的介绍和讨论。绝好的一本书！它是使我职业生涯发生转折的又一重磅因素（第三个是极限编程）。感谢 Pete。

感谢 Peter Coad，他以各种方式向我提供帮助而且帮助我启动了这个项目，真是感激不尽。还要感谢他让我事先推出了 TDD 版的《The Coad Letter》[URL 61]⁴。

感谢 Paul Petralia 及与他一起工作的全体工作人员，Paul 是 Prentice Hall 出版公司的组稿编辑。感谢让我们说服你这不是一本有关“测试”的书以及在说服你之后对此毫不动摇的态度。

在此必须要提到 Craig Larman，感谢他的鼓励、支持和建议。我仍然清晰地记得那天在达拉斯郊外 Craig 的家里讨论 UML 和 Together[URL 34]并一起饮用家酿 Chai 酒的情景。

非常感谢 Ron Jeffries 为本书作序，多谢他一如既往地支持我在极限编程相关方面的尝试，尤其是对测试驱动开发一书以及《TDD Coad Letter》的支持。还要感谢他在极限编程方面所做的大量工作，没有他的努力，XP 不会有今天的成就。

我要对 TDD Yahoo! 讨论组的成员致以特殊的谢意，感谢他们把与 JUnit 有关的建议发送给我，由衷地感谢 Darren Hobbs、J. B. Rainsberger 和 Derek Weber。

向通过写作以及允许我使用其资料（他们才是这些方面的专家）等方式作出贡献的人们致以特殊的谢意（按照出现顺序排列）：

Mike Clark 有关 JUnitPerf 一节的内容；

Jens Uwe Pipka 有关 Daedalus 扩展一节的内容；

Tim Bacon 有关 xmlUnit 一节的内容；

Mike Bowler 有关 Gargoyle 扩展一节的内容；

Bryan Dollery 有关 IDEA 一节的内容；

James Newkirk 有关 NUnit 一章的内容；

Bob Payne 在最后一刻带来了有关 PyUnit 一章的内容；

Kay Pentecost 有关 vbUnit 一章的内容；

Scott Ambler 有关敏捷建模的附录。

感谢极限编程团体所有向我提供反馈意见的人们（排名不分顺序）：Kay Pentecost、Edmund Schweppe、Aldo Bergamini、Mike Clark、Francesco Cirillo 以及我的朋友、同事和以往的合作者：Randy Miller 和 Miroslav Novak。像所有的作者一样，肯定有所遗漏的人，对此表示歉意。

还要感谢本书的复审人员：Alan Francis 和 William Wake。

对，没错，正像 Kent Beck 在他的 TDD 一书[9]的前言里所说的，感谢家庭成员是老生常谈，但

4 这是以 Peter Coad 命名的一种网络刊物。——译者

是他们完全值得我去感谢。感谢我的妻子 Kate，感谢你说的“我来打扫厨房，你去写书吧”。感谢我的孩子们，感谢 Tasha 和 Jason，感谢你们在我不得不写书的时候对我的理解，感谢你们认为有一个能够写书的爸爸是一件多么酷的事儿。最后感谢我最小的孩子 Leah，她太小了，还不知道我在做什么，感谢她在看到我的时候露出的微笑以及当我抱起她时给我的拥抱。

本书是用各种开源软件制作的。我所有的计算机跑的都是红帽子 Linux。手稿是用 GNU Emacs 准备的，排版用的是 LaTeX，图片处理用的是 Gimp，其间大量使用 xdvi 进行预览。本书的 PDF 版是用 dvips 和 ps2pdf 创建的。与 LaTeX 一起使用的还有几个软件包，一些是现成的 (lgrind、draftcopy 和 fixme)，有几个是 Dave Thomas 荣誉提供的 (用于制作练习、扩展的参考引用支持和 URL 引用)，一些是我自己编写的 (用于编排引用语、需求描述/任务/测试管理、旁注和建议等)。

谨以此书献给我的父母，
尽管他们并不总是赞同和理解我所做的一切，
但还是放手让我去尝试。

目录

序 言
前 言

第 1 部分 背景知识介绍

第 1 章 测试驱动开发	3
什么是测试驱动开发?	4
让计算机来告诉你	5
一个简短的例子	6
小结	8
第 2 章 重构	10
什么是重构?	10
何时进行重构?	10
如何进行重构?	19
一些重要的重构方法	19
通过重构来引入模式 (Refactoring to Patterns)	33
小结	34
第 3 章 意图导向的编程	35
名字	35
简单	38
要做有根据的假设 (warranted assumption)	39
如何进行意图导向的编程?	39
“不要注释”	41
小结	43

第 2 部分 工具与技术

第 4 章 JUnit	47
体系结构概览	47
断言	48
编写一个 TestCase	51
运行测试	52
使用 setUp()和 tearDown()	56
使用 TestSuite	57

相互关联	58
测试的归属问题	62
一些建议	63
小结	66
第 5 章 JUnit 扩展	67
标准扩展	67
使用 MockObjects 补充的断言	71
使用 JUnitPerf 来测试性能与可伸缩性	72
Daedalus JUnit 扩展	78
使用 XMLUnit 编写基于 XML 的测试	87
Gargoyle 软件公司的 JUnit 扩展	95
第 6 章 与 JUnit 有关的工具	105
Jester	105
NoUnit	112
Clover	114
Eclipse	116
IDEA	118
第 7 章 模拟对象	120
模拟对象	120
例子解说	121
模拟对象的用途	127
如果能这样该有多好?	129
一个普通的例子	129
MockObjects 框架	130
MockMaker	134
EasyMock	136
小结	141
第 8 章 以测试优先的方式开发一个 GUI 应用	142
例子	142
AWT 机器人	143
蛮力测试 (Brute Force)	143
JFCUnit	149
Jemmy	156
超瘦 (Ultru-Thin) GUI	160
小结	167
第 3 部分 一个彻头彻尾的采用 TDD 开发的 Java 软件项目	
第 9 章 项目描述	171
概述	171

用户描述及相关任务	171
第 10 章 电影列表	175
为电影创建一个容器	175
为电影列表创建 GUI	186
在 GUI 中增加一个电影	192
回顾	199
第 11 章 可以为电影更名	200
支持对电影名进行编辑	200
对电影更名的 GUI	203
回顾	210
第 12 章 电影是惟一的	211
电影是惟一的	211
发现不惟一时的出错信息	215
回顾	222
第 13 章 评分	223
给电影打分	223
在 GUI 中显示评分	225
对评分进行编辑	231
回顾	237
第 14 章 类别	238
增加一种类别	238
在 GUI 中显示类别	242
提供可供选择的类别	245
回顾	248
第 15 章 根据类别进行筛选	249
获得基于类别的子列表 (sublist)	249
支持一种称做 ALL (全部类别) 的类别	252
在 GUI 上添加类别选择器 (category selector)	253
处理电影类别的变更	260
界面修整	264
回顾	266
第 16 章 持久性存储	267
将数据输出到平直文件中	267
GUI 中的 Save-As 菜单项	271
GUI 中的 Save 菜单项	280
从平直文件中读取数据	286
加载到 GUI 中	290
回顾	294

第 17 章 排序	295
电影比较.....	295
对电影列表进行排序.....	299
从 MovieListEditor 中获取排序后的列表.....	304
在 GUI 中增加排序操作.....	305
回顾.....	309
第 18 章 多重评分	310
多重评分.....	310
评分来源.....	316
修正后的持久性存储.....	324
在 GUI 中显示电影的多重评分.....	339
在 GUI 中添加增加评分的功能.....	344
删除单一评分域.....	349
回顾.....	349
第 19 章 评论	350
为评分附加评论.....	350
保存评论.....	352
加载评论.....	354
显示评论.....	355
增加评论.....	364
回顾.....	366
第 20 章 项目回顾	367
设计.....	367
测试代码与应用代码的比较.....	370
测试质量.....	370
模拟对象的使用.....	374
总体评述.....	374
调试.....	375
测试清单.....	376
小结.....	382

第 4 部分 xUnit 一族

第 21 章 RubyUnit	385
框架描述.....	385
范例.....	386
第 22 章 SUnit	390
框架描述.....	390
范例.....	391

第 23 章 CppUnit.....	396
框架描述.....	396
范例.....	397
第 24 章 NUnit.....	403
框架描述.....	403
范例.....	404
第 25 章 PyUnit.....	412
框架描述.....	412
范例.....	414
第 26 章 vbUnit.....	420
框架描述.....	420
范例.....	421
附录 A 极限编程.....	429
敏捷革命.....	429
极限编程.....	430
四种变化量.....	430
价值观.....	432
实践方法.....	433
小结.....	436
附录 B 敏捷建模.....	437
与建模有关的一些荒诞论调.....	437
敏捷建模 (AM) 介绍.....	439
什么是敏捷模型?.....	443
附录 C 联机资源.....	446
论坛.....	446
有关敏捷开发过程的信息.....	446
有关极限编程的信息.....	447
与 JUnit 有关的软件.....	447
与 JUnit 有关的信息.....	449
工具.....	450
其他的 xUNIT 家族成员.....	451
公司.....	452
杂类.....	453
附录 D 练习答案.....	455
参考文献.....	476

第 1 部分

背景知识介绍