

<<测试驱动的面向对象软件开发>>

图书基本信息

书名：<<测试驱动的面向对象软件开发>>

13位ISBN编号：9787111304258

10位ISBN编号：711130425X

出版时间：2010年8月

出版时间：机械工业出版社

作者：Steve Freeman, Nat Pryce

页数：253

译者：王海鹏

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<测试驱动的面向对象软件开发>>

前言

随着软件发布周期越来越短，一个困境出现了：如何在更少的时间里发布更多的软件，并持续地发布下去？

我们需要新的观点来解决这个困境。

只有技术的转变是不够的。

本书展示了这样一个新观点——假如软件不是“做”出来的，不像我们做纸飞机那样，叠好后就飞出去，情况会怎样呢？

假如我们像对待有价值、有产出的农作物那样（培育、剪草、收割、施肥和灌溉）对待软件，情况会怎样呢？

几十年或几个世纪以来，传统的农夫知道如何让农作物高产。

如果我们以同样的方式对待程序，软件开发会有什么不同呢？

我印象最深的是，本书同时提供了这种观点转变的哲学思想和实现机制。

本书的作者是善于写代码的实践者，他们也善于教别人写代码。

您从中可以学到如何保持高产，以及如何重新审视您的程序。

这里展示的测试驱动开发的方式不同于我所使用的方式。

虽然我还不能清晰表达其中的区别，但我已从作者清楚、自信的技巧介绍中受益。

方言的多样化已经为我进一步优化开发提供了思考的新源泉。

本书展示了一个有条理的、一致的开发体系，其中的不同技巧相互支持。

我建议您阅读本书，通过书中的例子，了解作者如何思考编程，如何实践编程。

这种体验将丰富您的软件开发方式，有助于您编程，而且同样重要的是，有助于您以不同的观点来看待您的程序。

<<测试驱动的面向对象软件开发>>

内容概要

本书采用通俗易懂的比喻,众所周知的编程语言,短小精悍的工作实例,深入浅出的分析处理——仿佛在和几位世界级的编程高手一边喝茶,一边聊天,循序渐进地让读者在不知不觉中进入编程的最高境界。

即使是刚刚入门的初学者,也会从中找到读书的乐趣,因为可以从一开始就找到开启面向对象开发大门的钥匙;随着经验的积累,编程水平的提高,再来看这本书,用不同的视角重新审视程序,又会体会到更深层的编程哲学。

本书是编程爱好者的启蒙指南,更是系统分析人员、测试人员、程序设计人员、软件开发人员以及面向对象程序研究人员等专业人士革新编程思想的必备手册。

<<测试驱动的面向对象软件开发>>

作者简介

Steve Freeman是一名独立咨询师，擅长领域是敏捷软件开发（<http://www.m3p.co.uk>）。他与Nat Pryce一同赢得了2006年敏捷联盟的Gordon Pask奖。他是伦敦极限星期二俱乐部（London Extreme Tuesday Club）的创建成员，也是第一任伦敦XP日（London XP Day）的主席，还经常在国际

<<测试驱动的面向对象软件开发>>

书籍目录

对本书的赞誉 译者序 序 前言 作者简介 致谢 第一部分 简介 第1章 测试驱动开发的要点 第2章 测试驱动开发与对象 第3章 工具介绍 第二部分 测试驱动开发过程 第4章 启动测试驱动循环 第5章 保持测试驱动循环 第6章 面向对象风格 第7章 实现面向对象设计 第8章 基于第三方代码构建 第三部分 工作的例子 第9章 委托开发一个拍卖狙击者 第10章 可行走的骨架 第11章 通过第一个测试 第12章 准备竞拍 第13章 狙击者发出竞拍出价 第14章 狙击者赢得拍卖 第15章 迈向真正的用户界面 第16章 狙击多项物品 第17章 分解Main 第18章 填充细节 第19章 处理失败 第四部分 可持续的测试驱动开发 第20章 聆听测试 第21章 测试可读性 第22章 构造复杂的测试数据 第23章 测试诊断 第24章 测试的灵活性 第五部分 高级主题 第25章 测试持久性 第26章 单元测试与线程 第27章 测试异步代码 后记 模拟对象简史 附录A jMock2速查手册 附录B 编写Hamcrest Matcher 参考文献

<<测试驱动的面向对象软件开发>>

章节摘录

这意味着在实践中我们可以把系统分成两个“世界”：值和对象。

值以函数式的方式对待，对象则实现了系统有状态的行为。

在第三部分，您会看到书中的编码风格是如何根据我们面对的世界而变化的。

在本书中，我们将使用术语“对象”来指具有标识符、状态和处理过程的实例，而不用于指值。

似乎没有其他可以接受的术语同时包含这两种含义（如实体或过程）。

2.3 对象通信 只有当对象设计容易组合时，我们才能从这种高层次的、声明式的方式中获益。

在实践中，这意味着它们遵循相同的通信模式，也意味着它们之间的依赖关系是显式的。

通信模式是一组规则，决定了一组对象相互交谈的方式。

这些规则包括它们扮演的角色、它们可以发送什么消息、何时可以发送等。

在Java这样的语言中，我们可以利用（抽象的）接口来标识对象的角色，而不是利用（具体的）类——

虽然接口不能定义任何我们想说的内容。

根据我们的观点，领域模型就在这些通信模型之中，因为正是它们为对象间可能的关系赋予了意义。

以动态的、互相通信的结构来思考一个系统，这对于静态的分类来说是思考方式的巨大转变，而我们中的大多数人在刚接触对象时学到的都是静态的分类。

领域模型甚至都不是明显可见的，因为在我们使用的编程语言中，通信模式没有明确表达出来。

本书希望展示测试和模拟对象如何帮助我们更清楚地看到对象之间的通信。

这里有一个小例子，说明关注对象间的通信是怎样指导设计的。

在一个视频游戏中，对象可能包括：参与者（如玩家和敌人）、场景（玩家飞过的环境）、障碍（玩家遇上就会撞毁）、效果（如爆炸和烟雾）。

还有一些脚本，它们在游戏过程中在后台大量生成对象。

从玩家的视角来说，这是一种很好的分类，因为它支持我们在玩游戏时需要做出的决定，也就是从外界与游戏进行交互时做决定。

但是，对于实现这个游戏来说，这不是很有用的分类。

游戏引擎必须显示可见的对象，告诉对象随时间产生动画效果，检测实体对象间的碰撞，并在实体对象发生碰撞时，把决定代理给碰撞处理器，如图2-2所示。

正如您在图2-2中看到的，这两种视角是不同的，一个是从游戏引擎的角度来看，另一个是从参与对象实现的角度来看。

例如，障碍是可见的、实体的，而脚本是碰撞处理器和动画的，但不是可见的。

游戏中的对象扮演了不同角色，取决于引擎当时对它们的需要。

这种静态分类与动态通信之间的不匹配意味着，我们不太可能对游戏对象得到一个整齐的分类层次结构，同时又能适合引擎的需要。

在最好的时候，分类层次结构代表了应用的一个维度，它提供了一种机制，让对象之间能共享实现细节。

例如，我们可能有一个基类来实现基于帧的动画的共同特征。

在最坏的时候，我们曾看到许多代码库（包括我们自己的），充满了复杂性和重复，因为使用一种机制来表达多个概念。

<<测试驱动的面向对象软件开发>>

媒体关注与评论

“终于有一本书用丰富的代码揭示TDD和OOD之间的共生关系。
本书值得保存。

” ——Robert C.Martin

“如果您想成为当前TDD领域的一名专家，就需要理解本书中的思想

。”
” ——Michael Feathers

<<测试驱动的面向对象软件开发>>

编辑推荐

《测试驱动的面向对象软件开发》对于更快交付更好的软件，测试驱动开发（TDD）现在是一种已经确立的技术。

TDD基于一个简单的思想：在写产品代码之前先写它的测试代码。

但是，这个“简单”的思想需要一些技能和判断才能做得好。

现在有了一本TDD实践指南。

让您深入理解那些基本概念，并向您展示了如何让测试来引导开发，“培育”出一致的、可靠的、可维护的软件。

作者描述了他们使用的过程，努力实现的设计原则。

以及完成工作的一些工具。

通过一个详细实现的例子，您可以看到TDD如何在各个层次上工作，如何利用测试来驱动特征开发和代码的面向对象结构，如何利用模拟对象来发现并描述对象之间的关系。

在这个过程中。

《测试驱动的面向对象软件开发》系统地讨论了开发团队在使用TDD时遇到的挑战——从过程中集成TDD到测试最难的特征。

旧代码带来的新思维，Robert C.Martin高度评价,Kent Berk作序推荐。

《测试驱动的面向对象软件开发》包括以下内容：如何有效实现TDD：启动，然后在整个项目中保持你们的冲劲。

如何创建更干净的、更有表现力的、更可维护的代码。

如何利用测试。

对可持续的品质保持最严格的关注。

理解在真实软件开发的环境中，TDD、模拟对象、面向对象设计如何交织在一起。

如何利用模拟对象来指导面向对象设计。

如何在TDD困难的地方取得成功：管理复杂的测试数据。

测试持久层和并发。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>