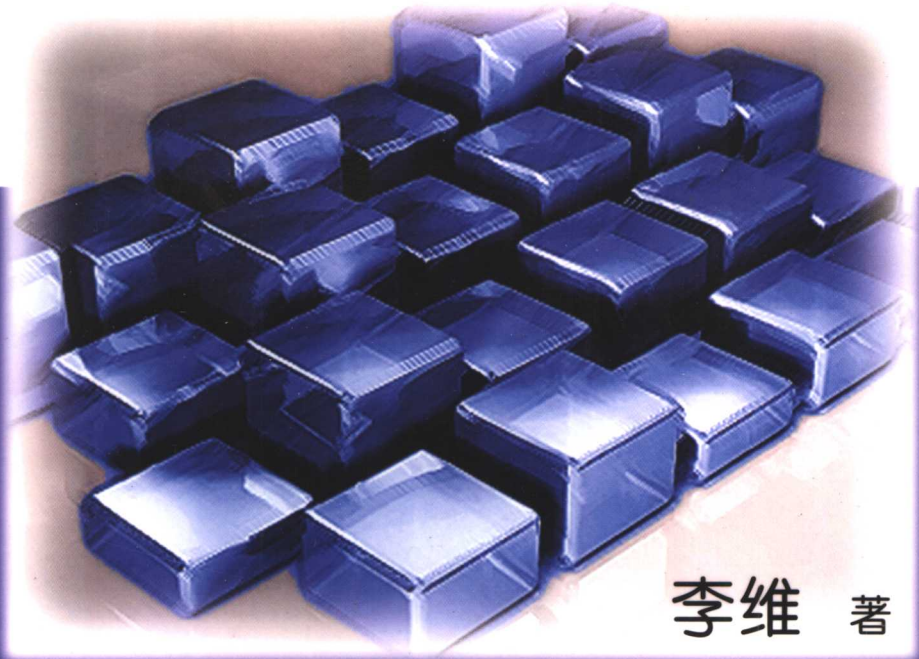
POOD Series

Broadview
WWW.BROADVIEW.COM.CN

面向对象开发实践之路

Practical Object-Oriented Development with C#

—— C#版



李维 著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

CD-ROM included

面向对象开发实践之路
——C#版

Practical Object-Oriented Development with C#

李 维 著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

《面向对象开发实践之路》有 C#和 Delphi 两个版本，本书为 C#版，主要介绍了利用主流开发方法学和技术技巧进行面向对象开发的原则与实践，通过完整剖析一个实际应用程序的设计、开发与实现，深入浅出地阐述 OOD（面向对象开发）、OOP（面向对象程序设计）、TDD（测试驱动开发）、UT（单元测试）等开发方法学与最佳实践的应用与技术技巧，全面展现深厚技术实践经验的精髓。

本书适合于习惯使用 RAD 方式而想学习如何使用 OOA/OOD 以及 XP、TDD 开发软件的 C#、Delphi、VB 以及 PowerBuilder 开发人员阅读。

图书在版编目（CIP）数据

面向对象开发实践之路. C#版 / 李维著. —北京：电子工业出版社，2005.8

ISBN 7-121-01643-5

Ⅰ. 面… Ⅱ. 李… Ⅲ. ①软件工具—程序设计 ②C 语言—程序设计 Ⅳ. ①TP311.56 ②TP312

中国版本图书馆 CIP 数据核字（2005）第 093726 号

责任编辑：周 筠 陈元玉

印 刷：北京智力达印刷有限公司

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

经 销：各地新华书店

开 本：787×980 1/16 印张：30.5 字数：570 千字

印 次：2005 年 8 月第 1 次印刷

印 数：7 000 册 定 价：49.00 元（含光盘 1 张）

序

作为一位IT人员，您阅读什么样的专业书籍呢？是喜欢偏向程序语言和系统的书还是喜欢目前流行的软件工程书籍？是喜欢讨论程序技巧的书籍还是设计架构的书呢？说实在的，这些笔者都喜欢，尤其是工作需要时一定得掌握所有必要的信息，不过一个有趣的问题是：在阅读每一本技术书籍时您会有什么样的感觉？

当笔者阅读有关偏向程序语言和系统的书籍，例如Delphi/C#/Java/.NET Framework等时，脑中却常常在想如何与软件工程以及设计架构结合在一起，因为这些书籍讨论的主要是实现技术，而许多专家不是告诉我们使用专业的开发流程和设计架构更是影响软件开发的重要因素吗？

当笔者阅读有关软件工程方面的书籍时，也常常觉得不同的软件工程似乎适合不同风格的IT人员和组织使用，有的IT人员非常严谨，他们喜欢使用拥有正式、一定流程的软件工程来发展软件；而有的IT人员则非常喜欢自由的风格，开发的软件也非常具有弹性，他们喜欢极为灵活的软件工程，然而有更多的IT人员介于这两者之间。不同的IT人员使用不同的程序语言，因为不同的IT人员有着不同的应用，或者对于程序语言有着不同的喜好，这是很自然的事情。软件工程也应该一样，不同的组织适用于不同的软件工程或是喜好不同的软件工程。例如RUP对于笔者个人而言感觉太沉重，而完全使用XP又觉得有些单薄，因此还是喜欢进行先期的基础设计。当笔者阅读系统架构的书籍时，经常发现有专业人员询问，要“多少设计才足够？”。就笔者在工作中的经验来说，过多的设计的确会造成失败的结果，请注意这里说的“失败”是指软件无法在限制的时程中完成。

最后，一个笔者最常思考的问题是：我们应该如何结合不同的程序语言、系统架

构、软件工程等来开发软件呢？例如当笔者阅读XP的书籍时觉得很棒，因此接下来的软件开发就自然受到XP的影响，但是笔者以前使用、喜欢和接受的OOA/D呢？TDD(Test-Driven Development)很酷，改变了笔者的想法、视野以及开发软件的方法，但是TDD一定只能和XP使用吗？不能和设计导向的开发方式一起使用吗？这似乎也不尽然。

对于许多使用Delphi/VB/PowerBuilder的开发人员来说，可能习惯了使用RAD开发模式，因此许多Delphi/VB/PowerBuilder开发人员并不熟悉如何使用面向对象的方式来开发软件。使用RAD并没有问题，RAD如果加上良好的架构仍然可以开发出很棒的软件，RAD加上TDD可以产生快速而质量优秀的软件，然而不可否认的是，现在我们身处的开发环境几乎都是面向对象的框架(Framework)、程序语言和集成开发环境了，因此开始试着结合RAD、面向对象、面向对象分析/面向对象设计、TDD等应该能够让Delphi/VB/PowerBuilder的开发人员学习到更多的概念和技术，不但可以提高生产力、增加软件的质量，也可以了解C++/Java那边的开发人员如何开发软件。结合RAD和面向对象能够提供更强大的力量，至少笔者是如此认为的。

在笔者阅读许多的IT书籍时，最喜欢看的是结合技术/理论从头设计软件的书籍，这个场景软件无须很大，只要完整地讨论从分析/设计，以及如何实现出来即可。这个发展过程应该说明设计是如何做出的，也就是说，类图(Class Diagram)之中的类、类架构、类中的PME(Property，Method和Event)是如何找出来的，也要说明如何根据设计架构实现出软件，在实现过程中会发生什么事？实现如何结合设计等细节。可惜的是这样的书很少，大多的书不是偏重设计讨论，就是用一堆程序代码展示程序代码技巧。笔者相信在许多OOA/D中漂漂亮亮、设计良好的类图或是设计架构绝不是第一次就能画/设计出来的，一定是经过某些刺激/化学作用才出现的，类中的PME也是一样，问题是这些刺激/化学作用是什么？

还是言归正传，这本书主要讨论的内容是什么呢？很简单，本书将以一个笔者以前使用RAD方式开发的小工具为起点，讨论如何使用OOA/D并且结合XP/TDD(Test-Driven Development)来实现这个工具。在整个设计和实现流程中读者将会实际看到如何使用面向对象技术以及敏捷开发方式来完成软件开发的工作。读者将会学习到如何把用户需求借助OOA/D转换为观念上的设计，当然这会产生用户案例(Use Case)、类图、循序图(Sequence Diagram)甚至是活动图(Activity Diagram)，不过

最重要的是借助书中的讨论、观察和设计，读者可以真正看到类架构是如何形成的。在形成了初步的设计之后，本书会结合敏捷开发和TDD来快速进入实现阶段，而在使用敏捷开发和TDD的过程中，我们又会发现这个步骤可以更进一步地让我们了解在OOA/D阶段的盲点，并且让我们能够再精致化原先的设计。笔者发现这样的结合非常有效，不但设计架构能够真正地反映实现的程序代码，而且几乎所有的实现程序代码又都能够被TDD所验证，这样的结果使得开发人员对于进行的设计和实现的程序代码都拥有高度的信心，这是笔者在没有这样做之前从来没有过的感觉(如果读者真正开发过大型的项目就可以了解笔者说的，当项目实现的程序代码愈来愈多时，最后就会愈来愈心虚)。

因此，这本书主要是给习惯使用RAD方式而想学习如何使用OOA/D/以及XP、TDD开发软件的Delphi/VB/PowerBuilder/C#开发人员阅读的，如果您已经是OOA/D的专家，那么您就不需要阅读这本书。这本书除了可以让Delphi/VB/PowerBuilder/C#开发人员学习如何使用OOA/D以及XP、TDD之外，在讨论的过程中读者也可以看到设计模式(Design Pattern)如何自然地出现在我们的设计并且实现在程序代码中，本书另外一个附加的功能则是读者也可以在本书中学习到许多.NET程序设计的技巧。

十年前笔者使用C/C++学习OOA/D时也曾经迷惘过，不知道如何设计面向对象应用程序，经过这些年工作的历练，笔者认为OOA/D也与学习程序语言和程序设计一样，只要具备扎实的基础知识，再加上多看、多听、多学、多练，就可以具备一定的技巧和经验，之后就像读者使用Delphi/VB/PowerBuilder一样会发现这些技能都是很自然地就掌握了。开放的学习之心、积极的学习态度和追求更好的愿望可以让开发人员成为IT领域的顶尖人才，笔者在此也鼓励Delphi/VB/PowerBuilder开发人员能够顺利地从RAD进入面向对象的世界，进而结合RAD/面向对象让C#能够发挥比其它程序语言/工具更为强大的开发能力。

李维

2005 年 6 月于台北，新店

Contents

目 录

Practical OO Development with C#

第0章 导读

当您打开本书时，最先浮现在您脑中的问题可能是：“这本书到底在说些什么？”、“这本书对我有没有什么帮助？”，或者“为什么要写这本书？”。如果您会这么想，那您一定是心中有主见的读者，也应该是适合阅读本书的读者。本书《面向对象开发实践之路——C#版》叙述的内容就在于讨论如何完整地使用面向对象技术以及测试驱动开发（Test-Driven Development，TDD）方法来开发一个完整而实际的软件。如果您想了解更多的发展背景，那么请再花一两分钟的时间看看下面的内容，以便决定这本书是否适合您。

缘由

当笔者在十年前第一次为Delphi撰写书籍时，最主要的目的是想把Delphi介绍给当时的开发人员。随着Delphi不断地发展，其最著名的开发模式——RAD（快速应用开发，Rapid Application Development）——也深植于大多数Delphi开发人员的思想之中。在笔者前几年的著作之中比较偏向于讨论Delphi的高级功能及其新奇技术。然而随着面向对象技术逐渐普及于软件开发流程中，笔者心想：现在是到了为Delphi开发人员介绍如何使用Delphi真正强大的开发模式来开发软件的时候了。因此，本书就是要介绍如何使用Delphi，并结合OOA/D的开发模式，以及最新的软件开发技术——XP（极限编程，Extreme Programming）与TDD（测试驱动开发，Test-Driven Development）——来开发软件的。

为什么现在是向Delphi开发人员介绍OOA/D与XP、TDD的成熟时机呢？原因很简单，那就是因为Delphi产品本身已经发展到了适当的时机，软件开发环境也开始强迫开

发人员必须采用新的开发模式。这怎么说呢？让我们看看下面的事情：

- Delphi 开始在产品中加入 OOA/D 的功能，这代表 Delphi 愈来愈适合在 Win32/.NET 下使用面向对象开发模式来开发大/中型应用软件。
- Microsoft .NET 本身即以面向对象为设计思想，.NET 框架更是以接口、类和设计模式为中心，因此，在.NET 平台下的开发人员必须了解和掌握面向对象技术。这也是新的软件平台强迫开发人员进行技术再升级的最好范例。
- 在新的 Delphi 产品中加入了更多更新的软件工程技术，例如重构（Refactoring）、TDD（Test-Driven Development，测试驱动开发）、MDA（Model-Driven Architecture，模型驱动构架）、DDA（Design-Driven Architecture，设计驱动构架）等。这样一来，开发人员如果想彻底发挥 Delphi 的强大功能，就必须学习和使用这些经实践证实为有效的软件工程开发模式。

就像数年前Delphi 2为开发人员介绍了客户端/服务器架构（Client/Server）技术、Delphi 3引入了多层（Multi-tier）技术一样，现在正是Delphi开发人员再次开始学习和采用新思维与新开发模式的大好时机，而这个想法也正是促使笔者撰写本书的火花。

许多开发人员认为，要使用OOA/D开发模式，就一定要使用C++、Java这样的编程语言，其实并非如此。Delphi开发环境所用的Object Pascal语言（在Delphi的最新版本中已经改称为Delphi语言了）就是一个纯粹的面向对象编程语言。和C++、Java比较起来，它不但毫不逊色，而且在面向对象方面还提供了更新、更好的开发机制。然而，Delphi之所以会让许多人误认为只能以RAD的方式开发软件，主要就是因为当以RAD的方式进行开发时，Delphi确实太好用了。另外，目前几乎没有书籍来说明如何以Delphi结合面向对象技术开发软件也是一个原因。

早在Delphi 8面世之际，笔者就开始构思、并准备撰写这本书了。然而由于工作繁忙，一直无法完成这本书。在Delphi 2005进入Beta测试后，当笔者看到其中加入了更多的面向对象技术，重构、TDD以及其他软件工程相关的功能时，心中就想，一定要完成这本书——因为改变的时机已经成熟了，Delphi开发人员也应该像Java社区一样开始使用面向对象技术、重构、设计模式等软件技术来开发质量更好的软件。笔者

希望本书能像数年前笔者的“实战Delphi 5.x”系列书籍一样，能引导Delphi开发人员进入软件开发的新领域，为大家引航指南。更重要的是要让大家意识到，软件技术并没有语言的界限，学习这些软件技术和软件工程知识对于每一个Delphi开发人员而言都是非常重要的。

当笔者于2004年完成了本书Delphi的版本后，把电子文档给一些朋友试读，希望这些朋友能够提供一些宝贵意见。这些朋友中也有同时熟悉好几门程序语言的人，其中有几位好友在看完Delphi版后，立刻问我为什么不再写一本面向对象开发实践之路的C#语言版？随着阅读Delphi电子文档的朋友逐渐增加，提出这个问题的朋友也愈来愈多，因此也让我开始认真思考这个可能性。

我第一次使用C#是在2000年参加MicrosoftVS.NET Beta测试计划时，到目前为止，使用C#也有一段时间了。当写完面向对象开发实践之路的Delphi版之后，我心中思考的问题就是是否只要把Delphi版的内容转换为C#语言就好了，还是应该再加入其它的内容？几经考虑之后，我决定增加一些新的内容，这在本质上是因为Delphi和C#程序语言仍然有所不同，而且使用的工具也不相同。

因此，在面向对象开发实践之路的C#版中，有以下内容是和Delphi版不一样的：

- 尽量使用 C#程序语言的特点。
- 增加程序代码稽核的内容。
- 增加程序代码度量的内容。
- 增加更多设计模式的内容。
- 增加更多程序代码最优化的内容。

我希望这些新增加的内容能够让C#的读者从本书获得更多的收获，当然这些新的内容也会适当地回馈到下一版的面向对象开发实践之路Delphi版之中，而未来在Delphi版中的新内容也可能会再回馈到下一版的C#一书中。如此写书的方式不也是我们软件中迭代式开发的应用范例吗？这也可以让本书有更好的质量和内容。

适合的读者

本书主要的目的是在帮助习惯使用RAD开发模式的读者正式进入OOA/D和XP、