

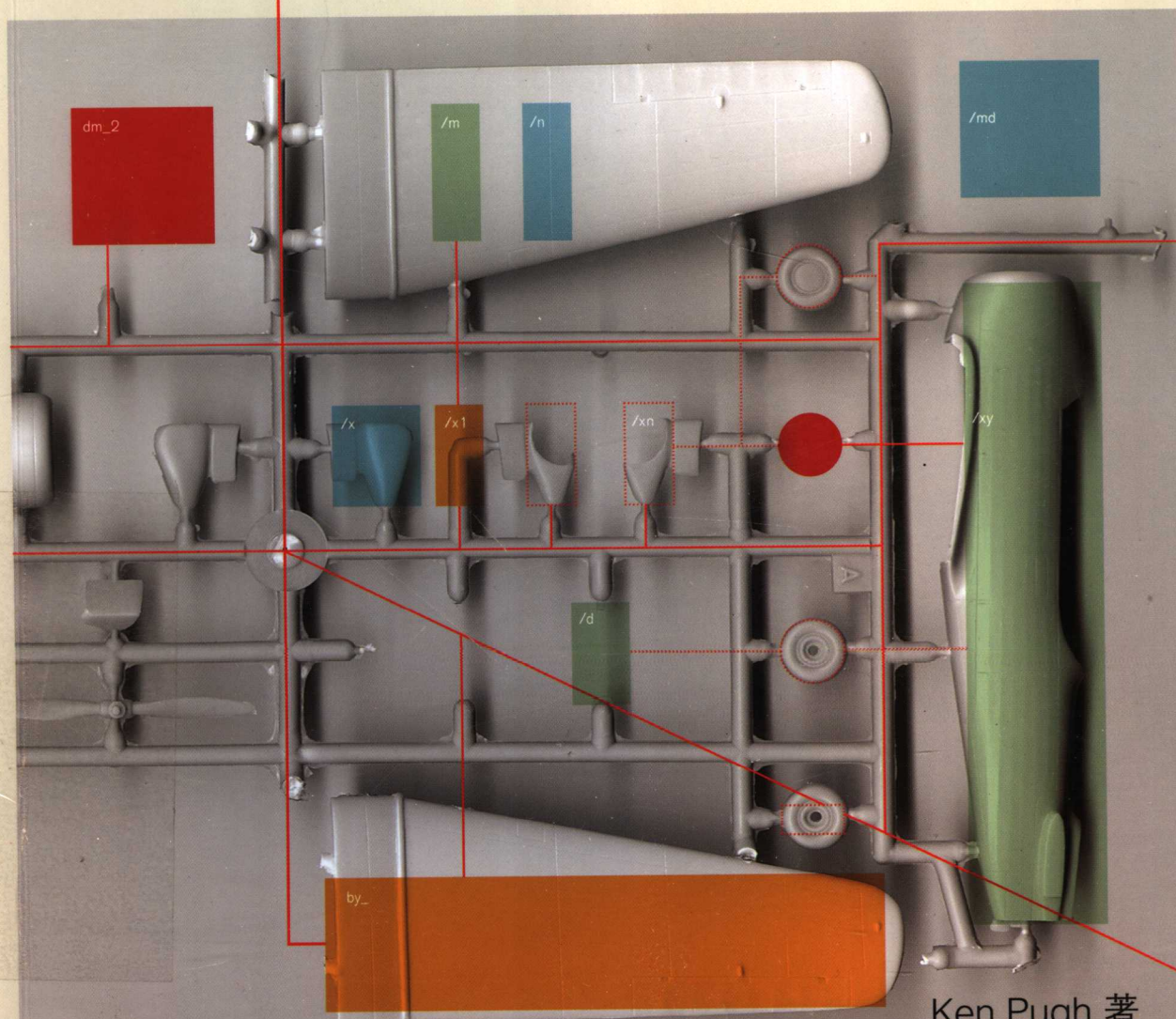
/THEORY/IN/PRACTICE

O'REILLY

Prefactoring

软件预构艺术 (中文版)

Extreme Abstraction • Extreme Separation • Extreme Readability



東南大學出版社

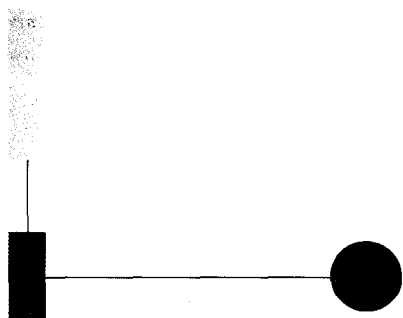
Ken Pugh 著
O'Reilly Taiwan公司 编译

TP311.52/165

2007

软件预构艺术

Extreme Abstraction • Extreme Separation • Extreme Readability



Ken Pugh 著

O'Reilly Taiwan 公司 编译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目 (CIP) 数据

软件预构艺术: 中文版 / (美) 普夫 (Pugh, K.) 著;
O'Reilly Taiwan 公司编译. —南京: 东南大学出版社,
2007.9

书名原文: Prefactoring

ISBN 978-7-5641-0906-6

I. 软 ... II. ①普 ... ②O... III. 软件开发 IV. TP311.52

中国版本图书馆 CIP 数据核字 (2007) 第 126870 号

江苏省版权局著作权合同登记

图字: 10-2006-126 号

©2005 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2005. Authorized translation of the English edition, 2005 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2005。

简体中文版由东南大学出版社出版 2005。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

书 名 / 软件预构艺术

责任编辑 / 张烨

封面设计 / MENDEDESIGN, www.mendeddesign.com, 张健

出版发行 / 东南大学出版社

地 址 / 南京四牌楼 2 号 邮编 210096

印 刷 / 扬中市印刷有限公司

开 本 / 787 毫米 × 980 毫米 16 开本 15 印张 252 千字

版 次 / 2007 年 9 月第 1 版 2007 年 9 月第 1 次印刷

印 数 / 0001-5000 册

书 号 / ISBN 978-7-5641-0906-6/TP · 148

定 价 / 26.00 元 (册)

O'Reilly Media, Inc. 介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权东南大学出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的 50 本书之一）到 GNN（最早的 Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的 Web 服务器软件），O'Reilly Media, Inc. 一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为 O'Reilly Media, Inc. 紧密地与计算机业界联系着，所以 O'Reilly Media, Inc. 知道市场上真正需要什么图书。

作者简介

Ken Pugh在软件分析和设计领域有相当广泛的经验。参与过的系统包括山羊血清流程控制、金融分析以及卫星轨道噪音记录等。他之前写过书讨论C和Unix，以前也替《C/C++ Users Journal》写专栏。他在威斯利学院、夏威夷大学以及许多公司教授程序设计课程，也时常在全国会议上发表演说。他担任个人顾问已经超过20年，从伦敦到悉尼都有他的客户。身为专家级证人，他也在民事诉讼和刑事案件中提供证词。

不工作时，他喜欢滑雪、冲浪、骑自行车，还喜欢到阿巴拉契亚山徒步旅行。

目录

前言	1
第一章 预构简介	9
何谓预构?	9
三个极致	10
探索方针	11
本书的情境	13
第二章 系统纪实	15
和 Sam 会谈	15
避免重复创造	19
名称里有什么?	19
碎块 vs. 整块	21
凝结成块	23
抽象化	24
原型值千言	29
第三章 一般开发议题	31
从大蓝图开始	31
接口契约	32
验证	34
程序代码传达的信息	35
一致性就是简易性	39
预构的态度	40
不要自我重复	41
把假设和决策形成说明文档	41

处理误差和错误	42
速度	47
电子表格难题	48
工具只是工具——要善用	51
第四章 大蓝图	53
系统续论	53
过程	55
初始设计	56
全局规划，局部设计	59
测试功能	60
测试品质	64
安全	65
第五章 撰写类	67
范畴和类	67
声明 vs. 执行	71
适当的继承	71
以文本通信	74
不止一个	75
第六章 再谈类	77
尊重类箴言	77
对象的三条定律	79
需求决定类	81
多态	83
一件小差事	86
原则 vs. 实现	88
极致命名	89
函数的重载	89
第七章 快完成了	91
我们在哪儿？	91

关注点的隔离	94
迁移至新系统	99
第八章 第一个版本	103
实践见真章	103
回顾时间	104
系统实情	104
操作接口	105
抽象数据类型	106
配置	106
测试	106
处理误差和错误	109
预构一下	110
第一轮发布的初版	111
实践和理论偶尔不相吻合	111
剩余的类	113
第九章 关联性和状态	115
Sam 的新需求	115
以谁为主?	116
对象的状态	118
第十章 接口和调整	122
目录搜索用例	122
设计接口	124
接口开发	127
接口测试	127
接口分割	128
有东西能用	129
第十一章 邮政编码和接口	132
调整	132
把钱交出去	136

未写的程序代码	137
间接性	139
日志记录	142
范式互斥	142
第十二章 更多的报表	144
花哨的报表	144
修改发生了	146
导出	149
第十三章 发票、信用卡、折扣	151
下一步	151
客户的语言	155
安全和隐私	158
第十四章 Sam 在扩张事业	160
第二家店	160
新的开发工作	162
第三家店	164
再见了, Sam	167
通用性	168
第十五章 打印服务器范例	169
简介	169
系统	170
消息	171
168	173
测试	177
日志记录	178
还可以隔离一点	179
后记	181

第十六章 反垃圾邮件范例	182
情境	182
检查垃圾邮件	186
ReceivingMailServer 类	187
ReceivedMailExaminer 类	194
完整流程	199
第十七章 后记	202
附录一 方针与原则	203
附录二 源代码	213

前言

预构 (prefactoring) 是让你运用经验累积而得的洞察力以及他人的经验开发出新项目软件的艺术 (译注 1)。本书之名源自于重构 (refactoring) 这个术语, 而重构之名得以远播起因于 Martin Fowler 所著的《Refactoring: Improving the Design of Existing Code》(Addison-Wesley Professional, 1999 年出版)。重构是修改程序代码的实践, 借此改良其内部结构, 而不会改变其外在行为。

本书描述设计、编码以及测试时的预构方针。运用本书论及的方针, 虽然无法保证不需重构你的设计或程序代码, 然而可能会减少必要的重构工作量。

许多方针推衍自多年来众多开发人员的经验。分析程序代码在初始阶段时应该如何开发以减少重构需求, 从而得出其他方针。如极致编程 (extreme programming, 简称 XP) 那样, 有些看起来也很像 XP 方针。许多方针都围绕着极致抽象化、极致隔离以及极致可读性这些概念。

译注 1: “prefactoring” 为作者自创的新词, 开宗明义地定义在前言的开始。本书的书名比较精准的翻译应为: 软件的预先架构, 但为求内文流畅, 均以“预构”一词贯穿本书。

有些方针涉及到设计模式 (*design patterns*)。设计模式是软件设计中常见问题的标准解法。软件设计模式的概念因《Design Patterns: Elements of Reusable Object-Oriented Software》这本书而大为流行 (Erich Gamma、Richard Helm、Ralph Johnson 以及 John Vlissides 合著, Addison-Wesley Professional, 1995 年出版)。该书讨论对象和类的模式, 它们形成了许多常见问题的基本解法 (注 1)。

有些方针在模式圈子里其实可称作设计模式。根据《Design Patterns》一书的说法: “某人的模式是另一人的设计原理。”就此书而言, 我称其为“方针” (guideline), 因为这些概念会在众多场合中不断重现, 而非纯粹的设计模式。这些方针没有整理成模式的形式, 因为这些方针只建议了开发程序时可以遵守的实践原则。

在我 35 年的职业生涯里, 我用过的计算机语言有 IBM 360、汇编语言、C、C++、Java、C#、HTML 以及 XML。我也用过不少软件框架, 包括 J2EE、Struts 以及 MFC。一路走来, 我和好几百名开发人员共事与互动过。这些开发人员为我提供了在本书中讨论到的众多方针的相关经验。

同时, 我也有机会从软件开发领域里的多位知名实践家处学得不少知识。本书结合了许多他们在文章和谈话中讨论到的理论, 以及我和他们交流到的有关我在职业生涯中发展出的实践。这些人包括 Scott Ambler、Chuck Allison、James Bach、Kent Beck、Grady Booch、Alister Cockburn、Larry Constantine、Jim Coplien、Ward Cunningham、Gary Evans、Bruce Eckel、Martin Fowler、James Grenning、Payson Hall、Allen Holub、Andy Hunt、Jason Hunter、Eric Jackson、Ivar Jacobsen、Ron Jeffries、Cem Kaner、Joshua Kerievsky、Robert Martin、Bret Pettichord、P. J. Plauger、James Rumbaugh、Dan Saks、Jim Short、Joel Spolsky、Bjarne Stroustrup、Dave Thomas、Bill Venner、Gerry Weinberg、Karl Wieggers 以及 Rebecca Wirfs-Brock。

我试着把书中每项方针的创始人都列举出来, 然而许多方针的来源都不详。如果你知道那些无人认领的方针是谁发明的, 请让我知道。有些方针列在《201 Principles of Software Development》(Alan M. Davis, McGraw-Hill, 1995 年)、《Code Complete》(Steve McConnell, 2nd, Microsoft Press, 2004 年) 以及网站 (<http://c2.com/cgi/wiki>) 上。

列举方针时如果没有情境配合, 就好像在谈 “curling” 时没有明讲是 “hair” 或 “ice” (curling 可指卷发或冰上游戏)。因此, 本书在介绍方针时, 会落实在一套 CD 出租系统

注 1: 参见《深入浅出设计模式》一书, 由 Elisabeth Freeman、Eric Freeman、Bert Bates 以及 Kathy Sierra 合著 (影印版由东南大学出版社于 2005 年出版), 这是另一本讨论模式之书。

的开发上。就我所知，这样的店并不存在。Martin Fowler 写的那本关于重构的书中有个音像店实例，这激起我这个构想。

我喜欢户外休闲活动。我特别喜爱冲浪、背包旅行、滑雪以及徒步旅行。Bruce A. Tate 写的《Bitter Java》（Manning 出版，2002 年）让我有这个想法：在描述编程时以运动来模拟。本书到处都有一些和讨论主题相关的户外故事。

人与人是不同的

不同的人看事情的角度会有些不同。常用文字处理器中出现文档活页夹可能很恼人，也可能有娱乐效果。一个针对操作界面看似了不起的想法，可能因为系统用户和系统开发人员有完全不同的视角，反而使结果变得没那么了不起了。

众人看待问题和解决问题的方法各有不同。我喜欢以纵观全局的视角查看某种解法，这样才能欣赏问题是怎么解决的。查看主要类和少数序列图的概况能够快速提供信息，比涉足程序代码更快。

不是每个人都像我这样做。有些人喜欢去读程序代码，观看类图对他们毫无用处，只会造成干扰。无论采用何种手段，都没有谁比谁好、谁比谁差的问题。当你碰上某人和你的做法不同而你又不懂得欣赏时，你才会碰上麻烦。

同样，开发人员对于从强类型（strongly typed）语言到非强类型（nontyped）语言之间的喜好范围也是各有不同，对于事前从大处着眼以及事前不从大处着眼也是如此。在藩篱一端的人，应该欣赏另一端的人所付出的代价以及所面临的情境问题。

此外，本书介绍的方针可能和你当前开发的案例相去甚远。将之和你自己的方针互相比较，有助于了解你的案例所付出的代价，说不定有助于鞭策你做些方针上的修正。

设计范例

本书描述系统开发，整合了过去已开发过的某些系统的众多功能。开发那些系统的经验构成了某些开发新系统的方针。开发人员并不完美。我们无法读取客户的心思，而且他的心思可能随时改变。本书概述的开发故事展现出决策之处会发生的意想不到的修正。

本书介绍的多种可能的设计方法之一可供解决客户的需求。评估设计的优良与否没有绝对的衡量方法。显然，设计有好坏的分别，但灰色地带也相当多。计算软件中方法的数量、程序代码行数或者类的数量，似乎是衡量的客观方式，但最后得到的数字通常没有特殊意义，除了讲求极致化以外。我是以一种内在实质的感觉来衡量设计。这就好像“嗅

出”程序是否需要重构一样。系统的整体设计给我的感觉范围可从“窝心”到“心烦意乱”。

读者

本书的读者对象是了解面向对象设计基本概念的人。本书会提出建议方针，以创建更具可读性及维护性的程序代码。

本书假设读者熟悉基本 UML 图表（类、顺序及状态），此外也懂得某种面向对象语言，如 Java、C++、C#、Ruby 或 Python。

本书内容

本书分成 17 章，各章的安排依照范例系统开发的流程。第三章另外探讨常见开发议题，而第六章探讨面向对象设计。范例系统开发在第十四章结束。第十五章描述了一个使用预构方针设计的实际系统，而第十六章谈的是反垃圾邮件系统的设计。

以下是各章内容的说明：

第一章 预构简介

这是对预构各个方面的简介。

第二章 系统纪实

我们和客户 Sam 碰面，了解他想要的系统全貌。我们会讨论创建一套共享的语汇以便于沟通，然后用到某些极致抽象化的概念。

第三章 一般开发议题

我们要讨论开发系统常见的议题，包括大蓝图（big picture）、接口契约、以程序代码交流、简易性、处理错误以及电子数据表谜题。

第四章 大蓝图

我们持续和 Sam 交谈，以便进一步了解整体需求，也就是大蓝图。然后，我们开始创建系统的设计内容。

第五章 撰写类

我们做出系统轮廓并开发实现类，接着探索单一类或数个类如何表达概念。

第六章 再谈类

我们要谈论一般的面向对象设计议题。查看内聚和耦合（cohesion and coupling）的类的箴言，另外还有三条对象定律。多态行为（polymorphic behavior）会以继承和接口双双展示。

第七章 快完成了

我们将说明利用关注点的隔离来创建报表。规划迁移之事会带来其他设计议题。

第八章 第一个版本

我们对所采取的设计手法进行回顾。探索开发过程中解决的问题以及编码阶段新增的其他类。

第九章 关联性和状态

Sam提出新需求。我们探讨如何利用系统中相关联的类实现这些新需求。我们会查看系统中对象状态的表示方式。

第十章 接口和调整

我们创建Sam的目录搜索用例所需的接口。我们会探索如何检测这些接口以及如何调整实现内容以符合这些接口。

第十一章 邮政编码和接口

Sam要让系统可以记录顾客的地址。我们使用一些接口来决定如何识别地址中的邮政编码。

第十二章 更多的报表

Sam认为他需要更花哨的报表以及各种不同的报表。我们以一些介绍过的方针实现其需求。

第十三章 发票、信用卡、折扣

Sam决定是时候新增开立发票给顾客以及以信用卡收取发票开立的金额的功能。我们要探究连接外在信用卡处理器的接口,并以Sam能理解的方式新增顾客的折扣运算。

第十四章 Sam 在扩张事业

Sam想扩张他的事业,他要开更多店,在本地和全球都有。他的店会在网站上显示出来。我们使用先前介绍过的许多方针来开发对此扩张事业所采取的解决手段。

第十五章 打印服务器范例

本章介绍一个案例研究,涉及到图书馆实际使用的系统,它可按由个人计算机打印的文件收费。本章说明采用的方针。

第十六章 反垃圾邮件范例

本章讨论如何收发电子邮件,介绍电子邮件接收器和垃圾邮件侦测器的可行性设计方案。

第十七章 后记

总结一些想法。

封面

封面是 Azur NAA-57 飞机的 1/72 比例的模型图。NAA-57 的前身是北美 T-6 Texan，而 T-6 Texan 是欧洲制造商制造的。MendeDesign 公司的 Jeremy Mende 设计了封面。他的设计和我的喜好很配。我是业余飞行员，虽然近几年没怎么飞过——除了几年前曾飞往阿拉斯加探险。

排版约定

伪代码范例以具体方式说明了许多方针。我试着让范例尽可能通用化，因为这不是语言手册。程序结合了许多语言的惯例。类名称以大写隔离（例如，`ClassName`）。属性、变量以及方法名称则使用下划线及小写（例如，`method_name()`）。另外，还有 C++ STL 以及 Python 的惯例。

如果类需要访问某个对象的属性，我会替其属性指定 `to` 和 `from` 的字眼，好像那些属性是 Eiffel 或 C# 的特性。就其他语言或其他惯例而言，你可能会用 `get` 和 `set` 作为方法名称的开头。

本书使用下列各种字型体例：

黑体字 (**Bold**)

用于表示状态或枚举值。

斜体字 (*Italic*)

用于表示新术语、网址、电子邮件地址、文件名、文件扩展名、路径名称、目录以及 Unix 工具。

等宽字 (Constant Width)

用于表示命令、选项、变量、属性、关键字、函数、类型、类、命名空间、方法、模块、特性、参数、值、对象、事件、事件处理器、XML、HTML 标记、文件内容以及命令的输出。它也用于表示伪代码的类轮廓。

等宽黑体字 (**Constant Bold**)

用于表示用户输入或信息交换。

NOTE

这个设计元素指出提示、建议或一般注意事项。

这个图示为指导方针。

程序范例

这本书是为了协助你把工作做好。一般而言,你可以在你的程序和说明文件中使用本书的程序代码。除非你要重制重要的程序代码,否则无需取得我们的许可。例如,使用本书的片段程序代码写了一个程序并不需要取得我们的许可。但是,把 O'Reilly 书籍的程序范例刻录成光盘贩卖或发布时就需要取得授权。引用本书的文句和范例程序代码来回答案问题时不需要取得许可,但把本书大量的程序范例整合到你的产品的说明文件中,则需要取得授权。

虽然不是必要,但若注明来源,我们会很感谢。注明来源时通常包括书名、作者、出版商以及 ISBN。例如,“*Prefactoring* by Ken Pugh. Copyright 2005 O'Reilly Media, Inc., 0-596-00874-0”。

如果你对书中的程序范例的使用情况有别于上述情况,不用客气,尽量和我们联络:
permissions@oreilly.com。

批评指教

我们已尽力核验本书所提供的信息,尽管如此,我们仍不能保证本书完全没有瑕疵,而网络世界的变化之快也使得本书永不过时的保证成为不可能。如果读者发现本书内容上的错误,不管是赘字、错字、语意不清或技术错误,我们都竭诚虚心接受读者的指教。如果您有任何问题,以下是我们的联络信息:

美国:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国:

100080 北京市海淀区知春路 49 号希格玛公寓 B 座 809 室
奥莱理软件(北京)有限公司

本书的 Web 页上列出了勘误表、示例和任何额外的信息。可通过以下网址查询:

<http://www.oreilly.com/catalog/prefactoring>
<http://www.oreilly.com.cn/book.php?bn=978-7-5641-0906-6>

如果想就本书的技术问题发表评论或咨询,请发邮件至: