

大师签名系列

Contributing to Eclipse  
Principles, Patterns,  
and Plug-Ins

# Contributing to Eclipse

中文版

MASTER SIGNATURE  
大师  
签名系列  
SERIES

[美] Erich Gamma Kent Beck 著  
熊节 译

■ 大师级作者联手打造的 Eclipse  
开山之作



中国电力出版社  
www.infopower.com.cn



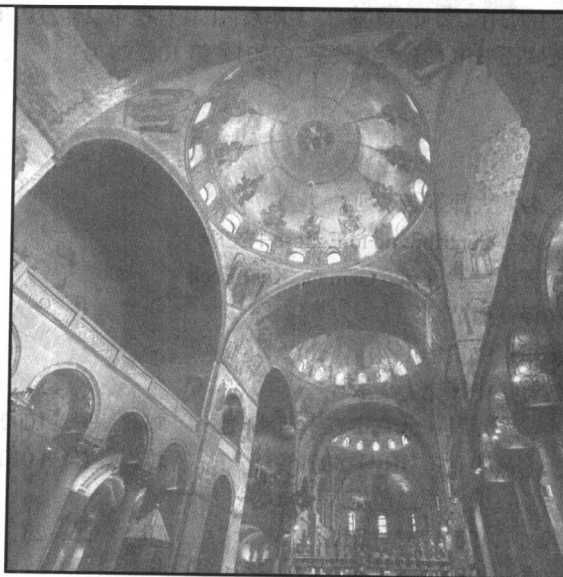
大师签名系列

Contributing to Eclipse  
Principles, Patterns, and Plug-Ins

# Contributing to Eclipse

中文版

[美] Erich Gamma Kent Beck 著  
熊节 译



中国电力出版社

[www.infopower.com.cn](http://www.infopower.com.cn)

Contributing to Eclipse: principles, patterns, and plug-ins (ISBN 0-321-20575-8)

Erich Gamma, Kent Beck

Copyright © 2004 by Pearson Education, Inc.

Original English Language Edition Published by Pearson Education, Inc.

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: 01-2004-5251

图书在版编目(CIP)数据

Contributing to Eclipse 中文版 / (美)甘莫 (Gamma,E.), (美)贝克 (Beck,K.) 著;熊节译.

—北京:中国电力出版社,2004

(大师签名系列)

ISBN 7-5083-2749-7

I.C... II.①甘...②贝...③熊... III.软件工具—程序设计 IV.TP311.56

中国版本图书馆 CIP 数据核字 (2004) 第 107298 号

丛 书 名: 大师签名系列

书 名: Contributing to Eclipse 中文版

编 著: (美) Erich Gamma, Kent Beck

翻 译: 熊节

责任编辑: 牛贵华

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

开本尺寸: 190×233 1/16

印 张: 23

字 数: 529千字

书 号: ISBN 7-5083-2749-7

版 次: 2005 年 1 月北京第 1 版

2005 年 1 月第 1 次印刷

定 价: 39.80 元

版权所有 翻印必究

# Eclipse 家族法则

---

## 扩展者

- 贡献法则 (Contribution Rule): 一切皆是贡献。
- 遵循法则 (Conformance Rule): 插件必须遵循预期的接口。
- 共享法则 (Sharing Rule): 增加, 不要取代。
- 有样学样法则 (Monkey See/Monkey Do Rule): 遇到问题时, 首先复制类似插件的结构。
- 相关性法则 (Relevance Rule): 只有在操作有可能成功时才显示你所贡献的操作。
- 整合法则 (Integration Rule): 要整合, 不要分裂。
- 责任法则 (Responsibility Rule): 明确指出你开发的插件是问题的源头。
- 针对 API 契约编程法则 (Program To API Contract Rule): 首先检查 Eclipse API 契约, 然后针对契约编程。
- “其他”法则 (Other Rule): 让用户可以选择所有东西, 但把那些通常不用于当前视角的选项放在 Other...对话框中。
- IResource 适配法则 (Adapt To IResource Rule): 应该尽量为领域对象定义 IResource 适配器。
- 分层法则 (Strata Rule): 将语言无关的功能与特定于具体语言的功能分开, 将核心功能与 UI 功能分开。
- 使用连贯性法则 (User Continuity Rule): 在多次会话之间, 应该保持用户界面状态一致。

## 促成者

- 邀请法则 (Invitation Rule): 尽可能地邀请别人为你的作品作出贡献。
- 懒加载法则 (Lazy Loading Rule): 只有在真正需要的时候才加载插件。
- 安全平台法则 (Safe Platform Rule): 作为扩展点的提供者, 你必须保护好自己, 不要让扩展者的误操作给你造成损失。
- 公平竞赛法则 (Fair Play Rule): 所有使用者遵守同样的游戏规则, 包括我自己。
- 明确扩展法则 (Explicit Extension Rule): 明确说明平台的什么地方可供扩展。
- 发散性法则 (Diversity Rule): 一个扩展点接纳多个扩展。
- 良好防御法则 (Good Fences Rule): 如果要交出程序的控制权, 首先保护好你自己。

- 用户决定法则 (User Arbitration Rule): 如果有多个选择, 由用户决定使用哪一个。
- 明确 API 法则 (Explicit API Rule): 将 API 与插件内部使用的类分开。
- 稳定性法则 (Stability Rule): 如果你已经开始邀请其他人作出贡献, 就不要再改变游戏规则。
- 保守 API 法则 (Defensive API Rule): 只暴露你有信心的 API, 但同时也应该做好准备暴露更多的 API, 因为使用者会要求你这样做。

## 发布者

- 许可法则 (License Rule): 每项贡献品都应该提供许可证。

# 译序：Eclipse 的 Meta-Knowledge

---

在这个表达泛滥的年代，“meta-”前缀的构词法常常也属于被滥用之列。一个平凡的词汇加上“meta-”前缀，似乎就凭添了几分高深和神秘——例如我们熟悉的“metadata（元数据）”和不那么熟悉的“metaphysics（玄学）”。不过，尽管有落俗之嫌，我仍然愿意说：在你手上的这本《Contributing to Eclipse》，它介绍的恰好是关于 Eclipse 的 meta-knowledge——那些“under the hood”的知识，那些支撑 Eclipse 世界的支柱。

大多数人对于 Eclipse 的了解是从 Java IDE 开始的。在这方面，Eclipse 仅仅是一个流行的（甚至算不上最流行的）产品，学会使用它对于聪明的程序员也不是什么难事。与 JBuilder、IDEA 甚至 Visual Studio 们相比，Eclipse 真正令人无法错认的特点在于它的插件机制。正如读者即将看到的，Eclipse 采用了“微核心+插件”的体系结构，这使它在两方面对程序员具有难以抗拒的吸引力：其一，用户可以比较容易地开发自己的插件，实现自己需要的功能——实际上，Eclipse 社群欢迎这种扩展；其二，这样的体系结构需要优雅而灵活的核心设计，于是 Eclipse 的核心代码就成了一个充满诱惑的设计宝库。由 Erich Gamma 和 Kent Beck 两位作者——实在无需再对这两位作者多加介绍了——为读者讲述插件开发的技术，阐释 Eclipse 的设计蓝图，本书提供的内容应该能吸引对 Eclipse 感兴趣的程序员。

然而，在我看来，这本书想要提供给它的读者的 meta-knowledge，甚至还在“模式地图”的“后面”。当你翻开这本书的封面，首先映入你眼帘的就是一串“Eclipse 家族法则”——贡献法则、遵循法则、共享法则、邀请法则……或许你要问了：“我只是对 Eclipse 的技术感兴趣，为什么要跟我讲 Eclipse 社群的工作方式？”那是因为，Eclipse 不仅是一个技术高手们展现技艺的项目，它更是一个工业级的产品。全靠了这些“家族法则”，世界各地的开源程序员们才能如此成功地协作，开发出如此成功的产品。

Eclipse 是一个开源项目。但和很多不明内情的人所想的不同，Eclipse 使它的社群广泛地获得了利益——而且主要是经济上的。不用说 IBM 了，Clover、Clay、WindowBuilder 等插件都为各自的公司带来了源源不断的利润。对于开源与商业之间的矛盾，本书的两位作者在“前言”的第一句话就让我们豁然开朗：“每个人都需要别人的关怀和帮助，每个人也同样需要关怀和帮助别人。”表面上看来，开源与商业是一对矛盾，其实它们之间的张力恰好是程序员职业的乐趣所在：我们可以用自己的代码赚钱，我们也可以用自己的代码去帮助别人。两者并不是矛盾，而是一种权衡。

软件开发也充满了各种各样的矛盾：灵活与效率的矛盾，松散耦合与紧密集成的矛盾，

大型产品与微小核心的矛盾……软件开发的核心工作就是在这些矛盾之间权衡，在不同的设计方案之间权衡，在不同的特性之间权衡。从这个意义上，本书就显得更有价值：两位经验丰富的作者没有用更多的力气去深究 Eclipse 的所有细节，却常常不厌其详地比较几种备选方案的优劣，说明选择一种方案的理由。再加上那些“家族法则”（以及随处可见的旁注），Erich Gamma 和 Kent Beck 将自己（以及 Eclipse 社群的其他贡献者）的工作方式、自己编程时的每个习惯和念头都呈现给了他们的读者。至少对于我来说，这些从字里行间渗透出的信息，才是本书最可宝贵的部分。即使我永远不再写任何与 Eclipse 相关的程序，这些成就了 Eclipse 的 meta-knowledge 对我的价值也将无可限量。

所以，对于书名中的“contribute”，我更愿意把它理解成“投身于……”：你不一定真正为社群贡献一个开源的插件，但 Eclipse 是一个充满乐趣与吸引力的世界，只要你投身其中、享受它的秩序与美，对于你自己和整个社群都将有所助益。而这本书就像两位作者在“前言”里所说的，就是带领你进入这个陌生世界的第一位导游。这位导游不会把美景逐一指给你看——有兴趣的游客自然会去观赏；他会陪在游客身边，跟游客聊起那些眼睛看不到的奇闻佚事。如果你当真有心领略这里的全部风光，我想你会喜欢这位导游，和他给你讲的那些故事——那些 meta-knowledge。

## 致谢

在译序中感谢原作者，这看起来多少有些古怪。但无论如何，我应该感谢 Erich Gamma 先生和 Kent Beck 先生。两位作者总共解答了我 14 个问题，如果没有他们及时而热心的回信，这个译本的质量将大打折扣——而那将是我一个人的责任。所以，从某种意义上，是他们好心帮我摆脱了这种尴尬境地，在此向他们表示最诚挚的谢意。

我还应该感谢中国电力出版社的编辑团队。“Eclipse”而又“Contributing”，这个书名就已经足以让国内大多数出版商退避三舍了。中国电力出版社多少有些理想主义地引进了这本书，才让我（和众多热心的读者们）有了一个机会。

现在，亲爱的读者，请做好准备，进入这次艰难而愉快的旅行吧。

谨以此书献给Karin、Jeremy和Jill，你们是我生命中的阳光。

——E. G.

每本书的背后都会有一些人默默地奉献。读者不认识他们，但读者手中的书有他们一份功劳。于我，其中的最重要者便是我的妻子 Cindee。她在我陷入困境时给我支持、鼓励和爱，她为本书付出了辛勤的劳动，她毫不吝惜地给我赞美和批评，并且在我工作时独自照看我们的孩子。也许你永远看不到她做的这一切，但如果没有她，我根本无法完成这本书。谢谢你，亲爱的 Cindee。

——K. B.



# 序\*

从很多种意义上来说，Eclipse 都是一件杰作。Eclipse 是一种技术，是一个可扩展的工具集成平台、还包括在这个平台之上构建起来的众多工具。Eclipse 还是一个开源项目，正是这个项目为世界奉献了这项卓越的技术。Eclipse 更是一个社群，这里的人们为开源项目做出贡献，同时也开发基于 Eclipse 的商业工具。这个社群涵盖了形形色色的人，从基于 Eclipse 产品的使用者，到编写 Eclipse 扩展工具的开发者，再到探索 Eclipse 潜能的研究者，他们都在为这个社群添砖加瓦。

xiii

Eclipse 是一项雄心勃勃的事业：它要为各种不同的工具提供一个互操作的平台——工具之间交互的方式甚至超出作者最初的想像。为了实现这一目标，我们选择了一种开放、平等、协作的开发过程——任何人都可以为这个平台做出贡献，但凡高质量的作品都会被采纳。正如“贡献法则”告诉我们的，“一切皆是贡献”。获得如此众多的贡献，一切皆有可能。

Eclipse 平台本身就由无数的插件组成，如果再把网撒得开一些，你还会找到数不胜数的插件。本书将帮助你跳入这个不断扩展的生态系统，亲身成为一名工具编写者。这本书是一位忠实的向导，它将带领你跨过第一道门槛，进入 Eclipse 开发的世界，然后把这个世界的地图交到你手上，任你自由翱翔。书中为你准备了一个详尽的范例，带你一步步走过一个插件的整个发展历程——从最基本的实现直到成熟的软件产品。另外，本书还将告诉你扩展 Eclipse 时应当遵循的原则，让你的插件开发不致偏离正道。即使读过其他介绍 Eclipse 的资料，相信你同样会欣赏这本书，因为作者匠心独运的指导能够帮助你加深对 Eclipse 的理解，并且让你从一个全新的角度审视那些过去很少体验的领域，获得新的认识。在很多章的结尾处，作者还给出了“前行的路标”，这些路标将帮助你找到更多相关的信息。

不能不提到本书的两位作者，他们对自己写作的主题有着深刻的理解。Erich 是 Java 开发工具领域当之无愧的权威，在编著本书之前，他就一直影响着 Eclipse 平台的发展和成长。Kent 对软件开发过程的理解无人能出其右，而且善于深入浅出地阐释技术思想。这一次，他们两人携手与读者分享他们在 Eclipse 方面的经验，并指导读者如何在这条路上获得成功。这是一本卓尔不群的实用书籍，它将指导你如何深入 Eclipse 的世界，我相信你一定会喜欢它。

xiv

\* 本书页边方框中的页码为英文原书之页码，索引中之页码即为此页码。——编者注

在这里，我乐于和你分享一些个人的感受。在 IDE 技术领域里，Erich 和我已经合作了好些年。回想起来，我们从前的很多工作都只是学习的台阶——有时甚至还没来得及完成一项工作，我们就不得不进入又一次新的迭代循环。但我们一直在学习。而在 Eclipse 这里，我们终于拥有了一个稳定而又欣欣向荣的技术平台。Eclipse 值得你投入精力和热情。请把 Eclipse 作为你工作的左膀右臂，我相信你会找到为 Eclipse 作出贡献的新途径！

IBM Eclipse 平台领导者

John Wiegand

2003 年 9 月

# 前言

---

每个人都需要别人的关怀和帮助，每个人也同样需要关怀和帮助别人。作为一名程序员翱翔在 Eclipse 的天空能够同时满足这两方面的需要，所以才会有那么多人乐此而不疲。当然，这里也有那些常见的宣传词，例如“提高生产力”、“紧跟市场”、“占据技术优势”——对于那些在其中工作的人，Eclipse 是一个能够大幅提高生产力的环境，人们为 Eclipse 作出的贡献使它更臻完美；如果找到了合适的插件，产品上市的时间将大大提前，你将获得巨大的技术优势。但是，尽管这些都是事实，但它们都不是重点，它们都不足以解释：为何开发者们如此钟爱 Eclipse。

xv

我们喜欢在 Eclipse 环境中工作，是因为它给了我们无微不至的关怀：每当遇到编程问题时，总会有一项 Eclipse 功能帮助我们解决它。我们喜欢为 Eclipse 作出贡献，是因为做这件事给了我们助人的快乐——既是帮助自己，更是帮助别人。如果我们作出的贡献还为别人留下了发展的空间，使他们能够在此之上作出更多的贡献，一个正反馈循环就出现了，每个人都会在其中获得快乐。

“为 Eclipse 作贡献”甚至还可能成为一个精彩的商业计划——“贡献”并不表示你不能让你的成果赚钱。你可以在 Eclipse 基础上发布功能完备的产品，也可以发售针对现有产品的扩展插件。

Eclipse 非常强大——不论从个人角度、专业角度还是社会角度来说都是如此。但是，要加入贡献者的圈子，你必须首先掌握大量的知识。我们希望本书能帮助你跨过这道门槛，开始为 Eclipse 作出贡献的旅程。

对于那些喜欢为别人编写开发工具的人来说，Eclipse 既带来了好消息，也带来了坏消息。好消息是：Eclipse 强大得令人难以置信，而且它的内部开发者都遵循创建 Eclipse Java 开发工具 (JDT) 时制订的那些准则，这使得它的结构清晰而优美。坏消息则是：Eclipse 拥有一套完整的世界观，如果你希望在 Eclipse 的沙箱里玩得开心，就必须首先牢记它的准则。

打个不恰当的比方，初涉 Eclipse 的感觉多少有点像孤身一人在曼谷上空跳伞（这个比喻不适用于泰国程序员）。平安着陆之后，你需要食物和住所，但如何去得到呢？你应该如何利用身边的资源满足自己迫切的需要？

xvi

帮助你克服这种迷失感正是本书的首要目标。如果你和一名向导一起降落在曼谷，你就可以对他说：“我饿了。”而他会告诉你：“你可以在某地吃顿饭。”同样，如果你说：“我想

实现这样一个功能。”本书会告诉你：“这应该是一个视角，那应该是一个 Object Contribution，而那个功能则需要一个新的编辑器。”

即便读完本书，你还是无法获得 Eclipse 世界的完整地图，但对于任何基本的问题，你至少会知道一条解决的途径。你还会了解到一些准则，只有遵守它们，你才能和别人友好相处。这就好像我们为你画出一幅曼谷地图，在上面标出六条大街、一家饭店和一家宾馆。你无法从这张图上了解曼谷的一切，但靠着它已经足以在这个城市生存下来，足以开始了解更多的东西。

在学习 Eclipse 的过程中，你花在阅读代码上的时间会比编写代码多得多。你必须习惯于一天用六个小时读代码，再用一个小时写代码——你会发现这样的一天效率同样高得惊人。在习惯了 Eclipse 的文化之后，你会自然而然地“知道”如何解决越来越多的问题，但你同样会经常照搬其他方案的结构来解决类似的问题——可能是模仿 Eclipse 的结构，也可能是模仿你自己以前的解决方案。

当我们与你一道走进书中的范例时，我们不会装出一副无所不知的圣贤模样——实际上，我们也记不住所有的细节。不过，我们会告诉你如何找到可供模仿的结构。学会充分利用 Eclipse 的搜索工具，这也是成为 Eclipse 贡献者所必需的一项技能。

本书不是为想要学习使用 Eclipse 的初学者准备的，我们希望读者至少熟悉 Eclipse 的常用术语——视图、编辑器，凡此种种。使用 Eclipse 一段时间之后，你很可能产生对它进行扩展的想法，那时这本书就该派上用场了。

在筹备这本书时，我们发现可以讲的概念数不胜数。可是，如果我们把 Eclipse 所有的 2000 个知识点都写进去，恐怕就再也没人愿意看这本书了。为了留住你的目光，我们决定只介绍其中的 50 个知识点——对于刚迈进这个世界的你来说，这应该是最重要的 50 件事。在很多章节的最后，你可以找到我们为你准备的“前行的路标”。你可以跟随它的指引在代码中找到书中概念的延伸。实际上，在向别人传授 Eclipse 知识时，我们最常用的办法就是：“不妨去看看 `org.eclipse.core.runtime` 的源代码”。

有这么一个故事：有个博士生马上要进行答辩，答辩的主题是“果实”。可是时间已经太紧迫了，他只来得及去了解黄瓜这一种果实。答辩开始了，第一个问题是：“请谈谈你对西红柿的了解。”“你看，”他说道，“西红柿是一种果实——跟黄瓜一样。黄瓜含有 80% 的水分，它的表皮能抵挡病虫害，可以用黄瓜来做沙拉。”就这样，尽管他不了解别的果实，但他成功地把每个问题都扯到了黄瓜上面。

我们也了解黄瓜。除了 Eclipse 之外，我们还有两条“黄瓜”：模式和 JUnit。所以，贯穿这本书的始终，你会不断发现类似于模式的建议——实际上，一开始写这本书的时候，我们总觉得有点不舒服，直到我们开始写下一些“法则”；有了“法则”的概念之后，我们的写作就进展得顺利多了。说到底，这些法则其实就是微型的模式。

JUnit——我们的第二条“黄瓜”——则是运行所有示例的基础。1997 年，在去亚特兰大的飞机上，我们俩花了几个小时开发出了 JUnit 的第一个版本。从那以后，我们的整个技

术生活几乎都围绕着它。对于本书中的示例代码，JUnit 是一个最理想的运行环境：你可以非常方便地运行所有测试，并一目了然地看到结果。

但是，我们必须给你一个忠告：如果你已经听厌了模式和 JUnit，也许这本书并不适合你。

## 格式约定

本书将使用如下的格式约定：

**粗体**——用于表示用户界面元素，例如菜单、按钮、标签、文本框等。

*斜体*——用于表示文件名和 URL。另外，新出现的术语也用斜体字强调。

Courier ——用于所有示例代码及正文中引用的代码元素。插件名称和 XML 文件中的元素也用这种字体表示。

**Courier 加粗**——用于强调示例代码中的某一部分，特别是插入或修改的代码。

~~Courier 加删除线~~——用于指出示例代码中应当删掉的部分。

在第二篇和第三篇的示例中，我们用下列图标表示当前的动作：



搜索——表示我们正在搜索、查看 Eclipse 的代码。



模仿——表示我们正在展示出自 Eclipse 的代码或 XML 标记。



测试——表示我们正在展示出自 JUnit 测试用例的代码。

## 在线范例

本书的网站是 [www.awprofessional.com/titles/0321205758](http://www.awprofessional.com/titles/0321205758)，在那里可以下载本书所有的示例源码。此外，该网站还提供一份纠错清单，以及其他与本书相关的消息。

xviii

如果要使用这些示例代码，你必须首先安装 Eclipse SDK（2.1.1 版）。请访问 [www.eclipse.org/downloads/](http://www.eclipse.org/downloads/)，在那里可以找到适合你的操作系统平台的 Eclipse SDK。

## 致谢

在撰写本书的过程中，我们从一个巨大的审阅者群体那里得到了大量热情洋溢而又无微不至的支持——只有很少的意见让我们不快。每当采纳了审阅者的建议时，我们总是尽量记住把那个人的名字加入下面的名单中。如果恰好漏掉了你的名字，请提醒我们，我们会在下次印刷时加上。我们记得的审阅者有：Frank Sauer、Jan Schulz、Dave W、Bob Foster、Binyan、Jeff Duska、Tom Ayerst、Richard Kuo、Wirianto Djunaidei、Don Estberg、Andreas Guthier、Vincent Massol、John Pickler、The Silicon Valley Patterns Group、Russ Rufer、Tracy Bialik、

John Brewer、Jerry Lewis、Jeff Miller、Patrick Manion、Carol Thistlethwaite、Pei-wei Wu、Chris Lopez、Ken Hejmanowski、Walter Vannini、Azad Bolour、Thomas Roche、Phil Goodwin、Carsten Heyl、Ed Burnette、Joe Bowbeer、David Loeffler、Jim Sawyer、Alex Blewitt、Laurent Bendel、Tom Killa、Bill Schauweker、Jan Ploski、Jan Looney、Rich Smith、Andy Farley、Debbie Utley、Ivan Moore、Geoff Gibbs、Mark Barkan、Florian Hawlitzek、Steve Blass、Mayuresh Kadu、Ricardo Lecheta、Stefan Baramov、Arne Hänle、Peter Friese、Randy Gordon、Frank Sauer、Andy Yang、Eric Pieters、Tobias Widmer、Bernard Gaffney、James Howe、Matt Dickie、Lee R. Nackman 和 Wim Engels。

Joel Rosi-Schwartz 和 Betty Rosi-Schwartz 还审阅了我们的手稿，Bard Bloom 及时指出了我们在“第零循环”中留下的一些漏洞，在此向他们表示感谢。如果本书中还有其他错误，责任全在我们二人。

我们要特别感谢 Eclipse.org 的提交者（committer）们，他们给了我们及时而详尽的技术反馈。André Weinand、Dirk Bäumer、John Wiegand 和 Kai-Uwe Mätzel 不止一次地对我们的书稿提供深入的反馈，Jim des Rivières、Nick Edgar、Philippe Mulet 和 Steve Northover 专门审阅了“第三循环”的各章节，Daniel Megert 和 Martin Äschlimann 对全书的源码和文字做了详细检查，也向他们表示感谢。

Mike Hendrickson 和 Eric Evans 为我们提供了及时的技术帮助。

Kent 在此要特别感谢 Erich，本书能够得以付梓，Erich 当居首功。我无比珍视我们之间的专业关系。

我们要感谢 Cindee Andres，她不辞辛劳的工作大大提升了我们文字的质量。我们还要感谢 Joëlle Andres-Beck，她一直关注着本书的写作，找出了“前行的路标”中的不少问题。

最后，我们要感谢我们的出版团队——Lynda D' Arcangelo、Tyrrell Albaugh 和 John Fuller。Mike Hendrickson 鼓励我们开始撰写本书，我们的编辑 John Wait 和 John Neidhart 总在最需要的时候给我们支持，Chris Guzikowski 领军的市场团队及时而准确地将本书的信息传达给购书人，向他们致以诚挚的谢意。

# 目录

---

译序: Eclipse 的 Meta-Knowledge

序

前言

|                      |   |
|----------------------|---|
| 第 1 章 全景 .....       | 1 |
| 1.1 本书的目标 .....      | 3 |
| 1.2 插件 .....         | 4 |
| 1.3 Eclipse 简介 ..... | 4 |

## 第一篇 第零循环: Hello World

|                                |    |
|--------------------------------|----|
| 第 2 章 为 Eclipse 插件开发做好准备 ..... | 9  |
| 2.1 设置工作区 .....                | 9  |
| 2.2 源码的浏览和搜索 .....             | 10 |
| 2.3 前行的路标 .....                | 11 |
| 第 3 章 Hello World .....        | 12 |
| 3.1 声明与实现的分离 .....             | 13 |
| 3.2 Hello 按钮 .....             | 14 |
| 3.3 向世界问好 .....                | 20 |
| 3.4 前行的路标 .....                | 26 |

## 第二篇 第一循环: 基本的插件

|                          |    |
|--------------------------|----|
| 第 4 章 我们的计划 .....        | 29 |
| 4.1 例示 JUnit .....       | 29 |
| 4.2 JUnit 集成 .....       | 31 |
| 第 5 章 创建一个执行测试的菜单项 ..... | 32 |
| 5.1 前行的路标 .....          | 37 |
| 第 6 章 创建一个执行测试的菜单 .....  | 38 |
| 6.1 前行的路标 .....          | 47 |
| 第 7 章 显示测试结果 .....       | 48 |

|                             |    |
|-----------------------------|----|
| 7.1 前行的路标.....              | 52 |
| 第 8 章 定义扩展点 .....           | 53 |
| 8.1 前行的路标.....              | 61 |
| 第 9 章 向扩展发出通知.....          | 62 |
| 第 10 章 发布 .....             | 64 |
| 10.1 将插件打包.....             | 64 |
| 10.2 将插件打包成零件.....          | 67 |
| 10.3 贡献.....                | 70 |
| 10.4 前行的路标.....             | 72 |
| 第 11 章 结束第一循环.....          | 73 |
| 第 12 章 间奏：测试驱动的插件开发 .....   | 74 |
| 12.1 PDE JUnit .....        | 75 |
| 12.2 测试项目装置.....            | 79 |
| 12.3 测试我们贡献的 JUnit 插件 ..... | 81 |
| 12.4 现在.....                | 85 |

### 第三篇 第二循环：插件的其他部分

|                           |     |
|---------------------------|-----|
| 第 13 章 查看测试结果 .....       | 89  |
| 13.1 贡献一个视图.....          | 89  |
| 13.2 监听测试进程.....          | 92  |
| 13.3 改变颜色.....            | 94  |
| 第 14 章 贡献菜单 .....         | 100 |
| 14.1 创建上下文菜单.....         | 100 |
| 14.2 贡献菜单项.....           | 105 |
| 14.3 前行的路标.....           | 108 |
| 第 15 章 把失败的测试当作编译错误 ..... | 110 |
| 第 16 章 将失败的测试变成标记 .....   | 112 |
| 16.1 标记的测试.....           | 112 |
| 16.2 传递项目信息.....          | 115 |
| 16.3 创建标记.....            | 117 |
| 16.4 删除标记.....            | 120 |
| 16.5 标记的图片.....           | 122 |
| 16.6 标记处理.....            | 123 |
| 第 17 章 找出所有的测试 .....      | 127 |
| 第 18 章 构建器和项目特性 .....     | 131 |



|        |                                |     |
|--------|--------------------------------|-----|
| 18.1   | 资源监听器与构建器.....                 | 131 |
| 18.2   | 利用项目特性配置构建器.....               | 133 |
| 18.3   | 前行的路标.....                     | 139 |
| 第 19 章 | 自动测试的属性设置.....                 | 140 |
| 19.1   | 前行的路标.....                     | 147 |
| 第 20 章 | 异常处理.....                      | 148 |
| 20.1   | IStatus 和 CoreException.....   | 148 |
| 20.2   | 在错误对话框中展现异常.....               | 150 |
| 20.3   | Logging Errors.....            | 151 |
| 第 21 章 | 插件的跟踪调试.....                   | 154 |
| 21.1   | 前行的路标.....                     | 157 |
| 第 22 章 | 标记解决以及调用耗时操作.....              | 158 |
| 22.1   | 测试标记的解决办法.....                 | 158 |
| 22.2   | 构建并再次执行.....                   | 159 |
| 22.3   | 显示构建进度.....                    | 160 |
| 22.4   | 重新执行测试.....                    | 164 |
| 22.5   | 前行的路标.....                     | 165 |
| 第 23 章 | 测试报表视图——使用 JFace.....          | 166 |
| 23.1   | TestResult.....                | 167 |
| 23.2   | 测试.....                        | 168 |
| 23.3   | 视图.....                        | 170 |
| 23.4   | TestReportLabelProvider.....   | 173 |
| 23.5   | TestReportContentProvider..... | 176 |
| 23.6   | 处理事件.....                      | 182 |
| 23.7   | 前行的路标.....                     | 186 |
| 第 24 章 | 一个简单的编辑器.....                  | 188 |
| 24.1   | 贡献一个编辑器.....                   | 190 |
| 24.2   | 扩展一个 contributor.....          | 194 |
| 24.3   | 前行的路标.....                     | 199 |
| 第 25 章 | 再论 ResultView——监视变化.....       | 201 |
| 25.1   | 检查颜色.....                      | 201 |
| 25.2   | 监视变化.....                      | 202 |
| 25.3   | 响应变化.....                      | 204 |
| 25.4   | 前行的路标.....                     | 207 |
| 第 26 章 | 视角.....                        | 209 |