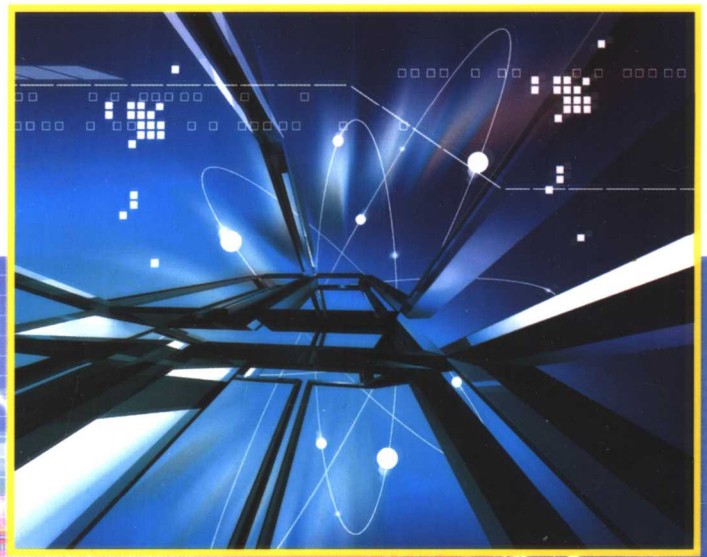
PEARSON
Addison Wesley

程序设计的模式语言·卷2

Pattern Languages of Program Design 2



John M. Vlissides
James O. Coplien 编
Norman L. Kerth

詹文军 周毅 等 译

清华大学出版社

程序设计的模式语言·卷2

John M. Vlissides
James O. Coplien　编
Norman L. Kerth

詹文军　周毅　等译

清华大学出版社
北京

图书在版编目(CIP)数据

程序设计的模式语言·卷2/(美)维利斯德斯(Vlissides, J. M.),(美)科普林(Coplien, J. O.),(美)科兹(Kerth, N. L.)编；詹文军，周毅等译. 北京：清华大学出版社，2006.4

书名原文：Pattern Languages of Program Design 2

ISBN 7-302-12442-6

Ⅰ. 程… Ⅱ. ①维… ②科… ③科… ④詹… ⑤周… Ⅲ. 程序设计语言学 Ⅳ. TP311.1

中国版本图书馆 CIP 数据核字(2006)第 007285 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969

组稿编辑：常晓波
文稿编辑：李　晔
印 刷 者：北京四季青印刷厂
装 订 者：三河市春园印刷有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×230　**印张**：33　**字数**：732 千字
版　　次：2006 年 4 月第 1 版　2006 年 4 月第1 次印刷
书　　号：ISBN 7-302-12442-6/TP·7981
印　　数：1～3000
定　　价：59.80 元

前　言

一种思想就仿佛风中的一粒种子：种子在落地、生根、开始成长之前并没有多大意义。模式(pattern)的思想在软件领域出现20年了，并已经最终落地和开始结果了。1994年编程模式语言会议(Pattern Languages on Programming conference，PLoP '94)作为关于软件模式的第一次会议标志着软件模式活动的萌芽。接下来的一年中我们看到了含有大量模式文章的几本模式著作、杂志的出版，模式已经成为有关软件工程和面向对象技术的会议的主流内容。

然后是PLoP '95，它与前次会议有所不同，就如同一株秧苗与一棵小树之间的不同：它更强壮了，它的根扎得更深了，并且它与周围的环境越来越协调了。关注的重点已经从"模式究竟是什么?"转变到了"如何使模式变得有效，使一个有效的模式变得更有效?"这是一个意义深远的转变，是一个人类关于模式的思维快速成熟的信号。这种转变从自省转向一种健康的积极行为。许多人现在都赞同这样的观点：一个模式的价值并不是在于它的发现，也不是在于它的定义，而是在于它的关联、它的质量，以及它的影响。软件中的模式就是它们所代表的有关技术的最重要的文化。

书写模式听起来挺简单，但实际上不是。这个努力过程有两部分：认知我们自己在重现设计问题方面的智慧，以及有效地与该部分智慧沟通。

本书各章的作者们都熟知这个努力过程。它开始于对人们已经解决了的重现问题的反省；然后描述重现问题解决方案的本质特性；接着将描述的特性书写下来。这仅仅是开始。如果一个模式没有经过评判来改正自身，那么它总是无用的，这也就是为什么要举行PLoP。提交到PLoP的论文都要进入"指导过程"，在这里评审者将与作者对论文进行反复讨论，以提高论文的质量。一次提交的论文也许需要经过多次反复讨论之后才能确定是被接受或是被拒绝。

努力过程在PLoP继续着。论文并没有按照传统的方式被介绍。实际上，每篇论文都要在一个"作者的研讨会"中被评审，在那里作者倾听读者对自己的论文的讨论意见。作者不能参与讨论——这将会妨碍这个过程，使得编辑难以向其他读者灌输文章的实际信息。研讨会的目标是帮助作者提高论文水平，而不是维护它。

作者带着一个或者多个修改意见返回，并在将论文提交进行最后评审之前对论文进行修改。清除了最后一个障碍的论文将被包含到本书中。每一篇论文都必须经过相同的荆棘、暴风雨，以及犀利言辞的洗礼，然后才能成为本书中的一章。

对于种子和新的思想，你永远都不会准确了解它们是如何成长的。许多分支从模式的种子中萌芽，每个分支都沿着两条路径中的一条独立成长起来。一些分支按照Christopher Alexander[Alexander+77，Alexander+79]所宣称的路径成长，反映了他关于建筑体系的

模式的开创性工作。Alexander 将他的模式层次结构化成被称为“模式语言”的一个系统。但与藤蔓植物的分枝会自行发现自己的路径一样，这些 Alexandrian 风格的文章都采用了适合自己需要的形式：当一个路径合适时它们就转到该路径。

其他的分支则根据“四人组”(Gang of Four，GOF)所描述的路径前行，这个绰号来自于软件模式第一本书的作者们[Gamma＋95]。这些先驱者们发现了可以在面向对象软件设计中捕获智慧的被隔离模式的价值。他们的模式非常大，并且高度结构化，比 Alexander 的模式相互依赖性更小。他们帮助了更多的软件开发者认知和开发软件模式的优点，现在由于书写他们自己模式的努力他们已经受到更多的尊敬。

本书列举出了有关这些路径的大量范例。当读者研究本书的每一章时，请询问自己哪一章最适合自己，为什么适合。仔细查看风格、格式、被提出的问题、所解决的领域、结果以及每个模式的缺点。

然后开始行动。一个充满活力的模式社区需要每一个人的参与。你是否已经发现模式的方式适合自己的工作？如果是，请将它们提炼出来。你是否在其中发现了可以与同事共享的智慧？那么请让他们知道这些。你能否识别那些自己已经获取但是需要归档的智慧？请将它写下来，让其他人评论它。同时请考虑将它提交到下一次 PLoP!

没有哪一本书可以不经过幕后人们的帮助而成文，但是 PLoP 付出了额外的努力。Hillside 集团的成员组织了这次会议。来自于伊利诺伊 Urbana-Champaign 大学的 Brian Foote 和他的学生、同事们为汇编这些文章、反复复制和分发它们，以及为使本次会议运作更好而提供了非常宝贵的帮助。这些带头人如同无私的牧羊人看管着自己的羊群一般名副其实。那些 PLoP 非作者群的参与者们所做的贡献一样需要致谢，这使得 PLoP 不再是一个被动的倾听经历；阅读多篇文章，然后给出仔细斟酌、建设性的反馈是一种艰苦的工作。在这里向所有参与者表示诚挚的感谢!

在本系列前面的著作[Coplien＋95]中，我们已经避免了干预每个作者成果形式的细节以及布局，正如我们渴望排版上的一致一样。因此我们在 Addison-Wesley 的朋友们由于达到了排版方面的成就而倍受欢迎。Deborah Lafferty 通过文雅但是坚决的方式引导出版的效果；她的保持项目按计划进行的能力令人吃惊。Rojean Wagner 以及新英格兰 Editorial Services 的员工们接受了大量电子格式以及不停涌现的要求所带来的挑战。向 Tom Stone 表示特别感谢，感谢他冒着风险给一个初出茅庐的人以支持的声音。

最后，我们要认识对于本书成功来说最重要的人们——模式作者们。他们已经选择了同我们其他人共享自己的智慧，他们在这个过程中做出了巨大牺牲。花费时间来揭示什么才能更好地实现有竞争性的优势是需要勇气和奉献的。我们代表所有从其劳动中获益的人向他们致谢!

J. M. V
J. O. C
N. L. K

软件模式系列

本系列编辑：*John M. Vlissides*

软件模式系列(the Software Patterns Series，SPS)形成了对于软件开发者来说有持续重要意义的模式文化。软件模式记录了所有软件相关领域可重现问题的通用解决方案，从软件技术本身，到开发和发行软件的组织，以及使用软件的人。本系列书籍从一个或者多个领域汲取经验，形成软件专业人员可以快速应用的形式。关联(relevance)和影响(impact)是 SPS 的基本原则。关联是指每一本书都提供了解决真实问题的各种模式。模式是真正与其名称内在关联的；它们出自于实践者的经验之谈，而不是理论或者推测的结果。模式在改变人们如何工作得更好的方面有重要影响。一本书成为一个系列的一部分不仅仅是因为它符合这些基本原则，而且还因为它向读者展示了它的实现过程。

介绍－1

Ralph Johnson 和 Ward Cunningham

PLoP 已经创造了一种新的文化。这意味着 PLoP 的奠基者们对于已经存在的语言模式文化有些不满意，这是一个事实。这些奠基者，是面向对象编程领域中的一些知名人士，已经开始认识到自己领域内的优势受限于领域自身文化中存在的偏见。这种偏见，是科学出版物的传统产物，对于那些超越常规的、最近的发明和发现有偏好，而不管其是否有用。奠基者们对于常规现象的兴趣也许源自于他们对软件重用的研究。或者也许是源自于他们对那些采用最新技术但是由于缺乏常规解决方案而失败的项目的观察。这使得所有的奠基者都同意将他们的注意力集中到解决方案的传播上。PLoP 会议就是实施方案传播的一个结果。

本卷是系列书籍中致力于“模式形式”(pattern form)的第 1 卷，这是我们认为的共享解决方案的最佳方式。我们请求作者按照“模式形式”提交论文，但没有确切地说明该形式是什么。Christopher Alexander 有效表述了术语模式语言(pattern language)，并且在他的著作 *The Timeless Way of Building* 中很好地解释了形式这个概念。许多作者对于这本书都很熟悉，更多的人由于一系列的 OOPSLA 研讨会以及 Internet 讨论组而知道这本书。即使是这样，我们认为作者们仍然有必要不受约束地考虑将 Alexander 的形式概念纳入到计算机编程领域。我们坚持的一件事情是每一篇论文都要描述一个解决方案，一个能够用于解决一个问题的方案。

读者在阅读本卷时会注意到本书作者提供的解决方案覆盖范围非常大。这意味着读者也许会发现本书并不是每一章都特别有趣。我们期望随着 PLoP 的成长和成熟，PLoP 自身会细分到不同的传统应用中。将来的卷本将不需要读者具备当前阅读本卷所需具备的多面手要求。现在我们请求读者深入到本卷书的每一章中。即使没有打算直接应用自己阅读的内容，这种阅读方式也会使读者对于如何表述模式带来新的灵感，也会提供给读者一个关于开发软件为何如此困难的视角。

虽然存在各种不同看法，但在 PLoP 作者以及参与者中存在一些令人惊讶的相同意见。举例来说，大多数人都发现作为模式的本质所在，一个解决方案能够很容易超越它的表达式所显示的确切自然特性。一个模式最终还必须驻留于自己的服务目的上。因此各种书写样式——从标志模板的标记节到一种更具叙述性样式的运行段——对于一个成功模式来说，都并不比仅具备一些基本元素的样式作用更大。这些基本元素包括建立一个问题以及它的上下文，分析解决方案的有效性，并且，最重要的是，提供一个具体的解决方案。包括这些基本元素的模式是成功的。

PLoP '94的一个重要特征就是作者的研讨会。作者们聆听一小组人讨论自己的论文并

辩论论文的作用和缺陷，而不是仅仅向听众表述自己的论文。这使得作者不仅可以有机会知道大家是如何交流的，而且可以发现如何学习大家正在讨论的技术。作者们的研讨会，源自于几十年前那些具有创造性的创作社团，是一种非常重要的论坛，向新作者提供了解技术以及向有经验的作者提供打磨自己思想的机会。我们对 Richard Gabriel 在 1994 年春天将我们引入到作者研讨会深表感谢。据我们所知，这是研讨会第一次被使用到一个技术社团，而且看起来效果非常好。

正是这种远见和过程导致了本卷的诞生。我们非常高兴出现这种结果，相信读者也有同感。对于我们来说这是一个完全超速前行的产物。每一个星期我们都会看到一些新的证据证明我们和作者们已经得出的结论将会对我们编写程序的方式产生意义深远和持久的影响。这些证据也将例证我们关于软件工程将来发展的看法。

介绍—2

Richard Gabriel

周而复始(repetition)——一个傅科勒特摆、圣克鲁斯的波涛、年轻姑娘的迷人曲线——吸引着我们——一段稳定的R&B敲击、一些午夜摇篮曲。周而复始——押韵诗、拂晓前的咖啡——定义了我们的生活——晚饭、周末、夜晚的流逝。

当我们出生时,世界是新鲜的,并没有包含周而复始印象。一个婴儿第一次回应自己母亲的微笑并不是由于母亲自身,而是由于母亲重复的表示;最后他微笑是因为母亲的反应显示她就是母亲。每天早上8点20分我领着我的女儿来到公共汽车站,当我们停下后她会等一会儿,然后会注视大路看看公共汽车是否已经来了。她希望公共汽车在特定时间抵达,这样当汽车如期抵达时她会感到安全和舒服——而如果汽车没有如期抵达她会感到不高兴和紧张。

我们通过重复各种方式和内容来理解所有我们应该理解的事物。语法是我们叽里咕噜的正则表示,歌曲是我们制造噪音的风格体现,人类社会则是我们模式化基因素材的表达。我们通过认知以及根据一次又一次发生的事情来使用重现。寻觅不到重现而变得无法预测是令人害怕的。

重现可以实施是因为行为、特征以及关系都是显而易见地可以重复的。每一次行为——每一件事情——都不仅包含了显然的可重复部分,而且也包含了随之而来的可变部分。突出重现部分是因为我们的思维(我们的大脑?)就是按这种方式构造的。

周而复始仿佛像我们学习2×8=16,3×8=24,4×8=32时的口头禅一般。当我在五年级拼写"stillness"错误时,我会在黑板上将它拼写100次。当我打壁球出现误击或者将球打错方向时,我会喊"完成击拍,完成击拍,你这个笨蛋"。这是简单事物的一种口头禅。这种喊叫的神奇是——有魔力的;低吟般的歌咏——是一种口头禅。当开始周而复始的韵律时,身体开始摇摆;当喊叫的周而复始继续时,有时喊叫者不会停止。周而复始是一种力量。

重复的内容是重要的并且显示了共性。而不重复的内容是可变的并且能够被突出或者隐藏。周而复始使得我们感到安全,而可变则使我们感到自由。抽象揭示和刻画了周而复始中什么是共同的部分,而抽象的内容则根据重复的部分来定义。剩余的内容,从科学角度看,位于无知的面纱之后,而从计算角度看,剩余的内容则充满了不可知。抽象因为重复的内容具有的神奇魔力以及歌咏式的口头禅而被我们所知。

但是,为什么要念叨"完成击拍,完成击拍"呢?保持韵律是一种记忆术——在古时候规律被表述成口头禅式的韵律——这样记忆和教授就可以通过口头禅式的方式得以实现。这种完成击拍式的规律是很难被忘记的——这是一种规律,尽管,可能会意义不大:它的魔力

显示在何处？在击中一个球之后，所做的操作与击中球之时或之前所发生的事情有所差别，它对此有怎样的建议呢？

这个建议必须生效——因为我们随时都在击中球——但它如何生效？这里解释了建议如何生效：如果你计划在一个特定点触球，你的肌肉在正要达到该点之前会自然变得慢慢松弛——也许是为了防止摇晃。当你试图减缓手臂的向前挥动时，会采取相对运动以抵消向前趋势。你的手臂，就会在“干扰”中移动，上下摇摆，左右画圈；接着向前移动的力量就降低了。由于球拍的长度使得惯性增大。这都是击拍存在的问题。

当你完成击拍动作时，你的手臂瞄准了一个远离球拍能够触到球的位置点，因此这时就不需要减速，“不要减速，不要减速，你这个笨蛋！”这也许是一个能够反复念叨的口头禅，但它不是一个指导你如何做的建议。

这种口头禅产生了我们希望的效果——如同一粒种子生长成一枝花并最终开花结果一样，也如同沙滩上的风吹拂生成沙丘形状并由沙粒形成强大而涟漪的侧面一样。生成行为的力量是周而复始行为的伴生姊妹。如果赋予某人的任务是可完成的，那么这个人就会完成它。科学喜爱聪明的解释——回转力由自行车前轮生成，从而使你可以在转弯时正确地倾斜——但是聪明的解释仅仅是聪明的，它只有对于娱乐性的杂志文章或者 MIT 气概非凡的讲演来说才真正有意义。

生成的口头禅具有宗教仪式的心理力量，同时也具有规则算法的物质力量。宗教仪式构建我们对于周而复始和可预测的需要，宗教仪式也是迷信的基础——如果你用左脚在棒球运动员休息场区的顶端行走，将球棒的击球端树屑擦去，同时挖一个三英寸的壕沟来放住自己的右脚，你将不会出局。习惯认为——宠爱一只狗并将它的血压降低到睡眠水平以下的习惯作法，可以使得口吃者讲话清晰。

习惯性方法、周而复始、重现，这些都是简单的模式，可重复的部分，驯服令人惊异的变异的熟悉路标。你不知道投手将要如何投出球，但是你知道当你走向投球板时自己将要如何准备。类似于完成击球之类的建议，习惯就是人们可以遵从的事情。

软件——它是一类我们可以通过科学研究来学习生成的事物吗？根据好的语义来对系统进行分类将会使生成和维护 5 百万行的电信交换系统变得容易吗？在软件工作空间中通过更高级编程语言的输出数据释放出的生产率是否会增加额外 30％性能？

我并不这么认为。这样的工作会帮助一些职业研究者获得任职并且可以为他们的家人提供一种舒适的生活，但是你也许同样可以告诉一个网球新手关于如何控制肌肉放松并且请他们重复其新击打动作 20 次。

Hillside 集团的工作是提醒我们，对于编写软件的团队和组中的人们，以及在习惯性方法、周而复始、重现的抽象事务中，针对个体行为的每一步都是发明和创造。在几十年间，如果不是几个世纪的话，软件工程并不是，而且也不会是、不能是一门工程，因为我们仍然无法认识可重复部分的重要性。当我们认识了重要性后我们就将有了模式——重现、可预测。

我们仍然无法识别可重复的部分是真的吗？我有几年每隔几个月都要去一趟巴黎，我

的好朋友，Jérôme，总要带我到他已经预订了牡蛎的那家啤酒店去。无论什么时候我品尝这些牡蛎，我能识别的只是盐水、肥肉，有时候还有小颗粒。Jérôme 并不吝啬，许多时候我们都要吃 Locmariaquer 牡蛎，但是我只是尽可能礼貌地吃得最少。

几年后我访问加利福尼亚 Point Reyes Station 的时候，我和我的朋友点了牡蛎——Hog Island Sweetwater 牡蛎。这些牡蛎养殖在 Tomales 海湾，是河流的入海口，它们只是稍微有点咸。突然我能够品尝牡蛎的味道了，甜甜的，有一点怪，肌肉坚实，味道很好。第二天——为了品尝多种风味——我们又吃了其他地方的牡蛎，我又一次喜欢上了它们。一直到第二年我都在随时尽可能地品尝牡蛎，甚至它们不是甜水（sweetwater）牡蛎，我能品尝牡蛎隐含的余味并且能够喜欢它的这种风味——这个可重复的部分。

两年前 Jérôme 和我在波特兰、俄勒冈的时候，我带着他去了 Jake 连锁饭店，我们整整吃了 5 打各类牡蛎。他微笑着为我们俩点了香槟酒。

我需要学习认知可重复的部分——这需要时间和运气。有些东西味道像调味品——我很难学会品尝它。在那之前，牡蛎品尝起来像毒药。

软件品尝起来也像毒药。模式，可能，将会揭示其甜水味道。

模式将软件中的周而复始和重现具体化；它们向孤独的开发者提供惯例和熟悉方式，并且阻止他或者她免于被折磨到最终产生损失。一个模式是一种容易被理解的事物，越容易理解的模式就越像一个组成开发的一部分的宗教仪式。模式以周而复始为来源，提供宗教仪式。模式没有考虑的内容提供了自由空间，这是创造活动的起点。

对于模式来说没有内容——没有人将会创作一篇内容是一个模式或者模式语言的期刊文章或者会议论文——如果有人这么做，他肯定忘了这一点。正如同没有人会写一篇其内容是对完成击拍的建议从而使其可被接受的生物力学论文一样。

如果棒球手的宗教仪式有效的话，那是因为他放松了自己或者给予了自己自信。模式应该类似于这种宗教仪式——人们应该能够遵从这些，也许很少有原因会解释为什么或者这种原因可能是实际科学类论文的主题。内容不是在模式中——它位于其他地方。模式只是拥有宗教仪式部分，模式通过向开发者提供诸如“如果你这么做，那么就有效”之类的知识帮助其完成工作来达到目的。这种模式被称为生成式模式（generative parttern）。模式像宗教仪式般起作用，模式满足了我们对于周而复始和可预测的需求，模式为了人们在自己的生活中使用它而被设计出来——生成式模式像魔力一样起作用。

理解生成式模式不需要计算机科学学位。获取它们的要点并不需要拥有一个数学背景——只要按照它们说的去做就会成功。

那么如何产生生成式模式？不需要——只要去寻找它们。它们位于各种行家里手描述的系统中，它们位于有梦想需要维护的系统中。如果一个伟大的编程者来到你面前对你说，“嘿，看看我找到的代码”。那就是你将能够挖掘出生成式模式的地方。一些人创作或者开发出包含模式的代码，但这种代码很大程度上是与你不相干的：它是一种已经被使用并且被发现适于承载模式的相同的解决方案或者风格。我们无法找到单个的金原子，我们找到

的是矿石和金块——一堆聚集物。模式被发现是因为它们被很好地使用了；它们形成了任何人都可以找到的矿石。

伟大的模式编写者就是矿工，他们除了提供奇妙的解释之外没有创造任何东西——他们是作者，但他们不是科学家，甚至不是工程师。计算机科学系出身的人会认为他们是苦力、抄写员、文书。模式编写的天才们在任何地方都不会获得职位，他们不会向 CEO 们提出建议。一个模式编写者不会在自己的工具包中载着荣誉驶入晚年。

当我们阅读 Christopher Alexander 的模式时，我们看到几乎所有的模式都是有关人们如何与其他人一起以及独自生活在家中、乡镇以及城市中的，包括文化或者缺乏情趣的精神。这些模式很少谈及工程和构造。从物理解释上来说它们很简单——我们无法得知为什么房间中的光线会导致有用的生物反应，只知道这些地方是有生机和舒适的。Alexander 描绘了多个世纪以来的体系结构，他带领将近 12 个人的小组，花费接近 12 年找出了 253 个模式。

我们今天需要在 40 年积累的代码中寻找软件模式。我们已经发现模式了吗？

任何人都可以写下几行有节奏和韵律的话——它是诗歌？模式是一种形式——周而复始的宣称，各种变异的启示——我们仍然在写打油诗吗？

Kent Beck

曾经有一位山区中的国王，他统治的国家经济依赖于耕种。为了使有限的土地获得最好的收成，这个国王建立了学校来教授领地内每一个居民耕种的原理——光合作用、气候、昆虫学。在授课结业以后，每个年轻的小伙子或者姑娘被分配了一块可以进行耕种的土地，同时也得到了种子和工具。每一个人都被允许，甚至鼓励，按照他或者她自己的方式进行耕种。

但这种方式的结局是灾难性的。理解原理并不意味着可以使新的农夫不会犯下严重错误。庄稼在错误的时间被种植，相邻地域的庄稼没有共同协作以抵御害虫的侵害，土壤年复一年地疏于保养。

随着食物供应的骤减，这个国家的居民变得不安起来。他们饥肠辘辘地来到国王面前请求国王找到解决方法。国王则召集了国家中所有最富智慧的智者们，给了他们一个选择：要么解决耕种问题，要么就丢掉自己的脑袋。

经过多次耗费大量刺激性饮料的迟至深夜的会议之后，智者们向国王报告："陛下"，他们说，"对那些未来农夫们的教育使得他们知道了耕种的多种方式。但是那些老农夫们知道，只有一些耕种技术值得考虑。虽然理解原理是重要的，但理解如何能够生效的技术也是同等重要的。请告诉您最有经验和思想的农夫们，让他们在一本书中写下这些内容。然后将这本书提供给那些未来的农夫，这样他们就会避免错误并进行耕种了。"

随着新的耕种技术被认识和记录，这本书的内容快速增长。看起来这本书的内容将会

增长到没有人能够完全理解它。但经过一段时间后，可以发现许多新出现的技术其实是书中技术的翻版，这样被识别出来的新技术数量逐渐减少。这本书也渐渐形成一部内容繁多却目录清晰的巨著。

随着农夫们学习阅读和利用这本书，国王的食物产量变得稳定了。农夫们现在能够利用其他人的智慧了。当他们面临一种问题时，他们并不总是想要发明一种独立的方法来解决它；他们可以找到一种技术来解决这个问题。如果确实没有合适的技术，他们仍然会修改技术而不是自行解决。

农夫们有时会怀念那些疯狂发明技术的日子。但总的来说，国王是满意的。人民吃得很好，所以他们很高兴。国王再也不会整天都听到抱怨了，所以他也很高兴。由于智者们已经解决了问题，所以他们也可以保留他们的头颅来高兴地面对这种状况。

新时代开始了。

目　录

第 1 部分　特定语言的模式与惯用法

第 2 部分　通用模式