

<<编写有效用例>>

图书基本信息

书名：<<编写有效用例>>

13位ISBN编号：9787121166723

10位ISBN编号：7121166720

出版时间：2012-5

出版时间：电子工业出版社

作者：科伯恩

页数：340

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>

<<编写有效用例>>

前言

前言 为了描述行为需求、软件系统或业务流程，用例正在被越来越多的人所使用。初听起来，编写用例似乎是件很容易的事--只需要写清楚如何使用系统就可以了。但是，当提笔编写时，首先碰到的问题就是：“到底应该写些什么？多写一些，还是少写一些，详细程度如何？”

这些问题确实很难回答。

编写用例就像写散文一样，全部困难在于既要采用一般的写作方式，又要具有完美的表达能力。很难说什么样的用例是好用例，但是我们却可以去考虑：如何去写，才能写出好用例。

本书给出了我在用例编写和培训中总结出的一些指导性原则：应该如何思考、观察需要的内容，最终写出更好的用例和用例集。

本书还给出了大量的例子，既有好的用例，也有不好的用例，以及在不同情况下的编写方式。最重要的是要记住，我们需要的是有用的用例，而不必是一个最好的用例。

有时即使是很平常的用例也很有用，甚至远胜于编写许多需求文档。

所以无须太紧张，只要写出的东西可读，就说明你已经做出了贡献。

本书的读者 本书主要是针对工业界专业人员自学而编写的，因此在组织方式上也有所侧重，就像一本自学指南。

本书由浅入深，每一部分都包括概念、示例、提示和练习（部分答案）。

编写用例的培训教师应给出适当的解释和示例，课程设计人员可以围绕本书设计课程资料，在必要时作为指定读物发布（由于我对许多练习都给出了答案，因此他们必须编写属于自己的考试材料）。

本书的组织 本书首先概要地介绍了用例的概念，之后对用例体部分进行了详细阐述，然后是常见的问题，对忙于编写用例的人的提示，以及最后的注意事项。

“引言”部分对重要的概念进行了介绍，依次讨论了以下问题：“用例是什么样的？”

“何时编写用例？”

“什么样的用例是合法的？”

。

所有问题的答案都是根据具体情况（如时间、地点、人员及编写的理由等）来确定的。

关于这几个问题的讨论虽从本部分开始，但实际上贯穿本书。

“第一部分，用例体部分”，由多个章节组成，分别阐述了需要掌握的重要概念，以及应编写的模板。

具体章节为“用例是规范行为的契约”、“范围”、“项目相关人员和执行者”、“三个命名的目标层次”、“前置条件、触发事件和保证”、“场景和步骤”、“扩展”、“技术和数据的变化”、“连接用例”和“用例格式”。

“第二部分，经常讨论的主题”，列举了反复出现的特殊问题：“什么时候才算完成”、“扩展到多个用例”、“CRUD和参数化用例”、“业务过程建模”、“遗漏的需求”、“用例在整个过程中的作用”、“用例概述和极端编程”和“错误改正”。

“第三部分，对忙于编写用例的人的提示”，包含了一组提示，针对那些已经阅读完本书，或者已经了解这些资料，想要回顾一下关键概念的人们。

这部分结构如下：“对每个用例的提示”、“对用例集的提示”和“处理用例的提示”。

本书有4个附录：附录A讨论了“UML中的用例”，附录B包括“部分习题的答案”。

最后是附录C（术语表）和附录D（参考文献），即撰写本书时使用的资料列表。

<<编写有效用例>>

中文版书中订口处的“ ”表示英文版原书页码，便于读者与原书对照阅读，本书的索引所列页码为原英文版页码。

本书概念的来源 在20世纪60年代后期，Ivar Jacobson在爱立信公司电话系统工作时提出了后来成为众所周知的“用例”。

在20世纪80年代后期，他将用例引入了面向对象编程领域，在这里人们认识到用例可以填补需求分析过程中一个明显的空白。

我在20世纪90年代初参加了Jacobson的课程，他和他的小组没用我所使用的目标（goal）和目标失败（goal failure）这两个词，但最后我终于知道他们曾经使用过类似的概念。

经过几次比较，他和我都发现我们两个模型之间没有显著的区别。

我慢慢地根据最近的想法扩展了他的模型。

在1994年我为IBM顾问组编写用例指南时，建立了执行者（actor）和目标（goal）概念模型。

从而把用例中一些难以理解的事情解释清楚了，并且为如何构造和编写用例提供了指导。

从1995年开始，执行者和目标模型就在<http://members.aol.com/acockburn>及后来的www.usecases.org中正式使用，最后出现在1997年Journal of Object Oriented Programming上我写的一篇文章中，题为“构造带目标用例”。

从1994到1999年，尽管在理论上还有几个松散的分支，但用例模型的概念基本保持稳定。

在教授和指导过程中，我终于明白了人们为什么在简单的概念上花费了大量时间（我竟然从来没想到，我第一次尝试时也犯过很多同样的错误！）。

这些想法，加上“执行者和目标”模型中一些不合理之处，产生了本书及项目相关人员（stakeholder）和利益（interest）模型，该模型是本书提出的一个新概念。

统一建模语言（UML）对这些概念没什么影响，反之亦然。

Jacobson以前的一位同事Gunnar Overgaard编写了大量UML用例的资料，并且保持了Jacobson的风格。

然而UML标准开发组受画图工具的影响很大，以至于用例的文本特征在标准中消失了。

Gunnar Overgaard和Ivar Jacobson讨论了我提出的概念，并向我保证这些关于用例的大部分概念都适合放在UML的一个椭圆图中，因此既不会影响UML，也不会被UML标准所提出的概念影响。

这意味着你可以使用本书的概念，并与UML1.3用例标准兼容。

另一方面，如果你只读了UML标准，由于它根本没有讨论用例的内容及如何去编写，那么你也知道用例到底是什么、如何使用，并且还可能产生一个危险的想法，即用例由图形而不是文本构成的。

本书的目的是告诉你如何编写有效的用例，而标准很少谈及这些，因此我把对UML的评述单独放在附录A中。

本书所用的实例 本书编写的实例尽可能取自实际项目，并且有些实例看起来可能不太完美。我所要说明的是，这些实例对项目组的需求来说已经足够了，并且用例编写过程中的那些不完美之处是在误差和经济允许的范围内的。

Addison-Wesley的编辑们说服我把这些实例从原有形式中整理出来，以强调它们正确的形式，而不是它们实际的和可用的形式。

希望你通过阅读这些实例发现一些有用之处，并了解项目中的实际编写过程。

你可以应用我在这些实例中采用的一些原则，并找到改进它们的方法。

这种事情经常发生，因为改进一个人编写的实例是一件永无止境的事情，所以我接受这个挑战和任何批评意见。

《软件专业人员集锦》中的用例 《软件专业人员集锦》（The Crystal Collection for Software Professionals）只是图书选集之一，强调了一直被忽略的、发挥人能动性的软件开发技术。

其中一些书讨论了某种技术，一些书讨论了项目中的角色，一些书讨论了开发组协作的观点。

“集锦”遵循下面两个原则。

软件开发是创造和交流相互作用的游戏。

当我们提高开发人员的个人技术和开发组协作效率时，它得到了改进。

不同的项目有不同的需求。

<<编写有效用例>>

系统有不同的特征，由大小不同的开发组研制，并且开发组成员有着不同的价值观，优先考虑的事情也不同。

因此不可能定义一个最好的软件开发方法。

“集锦”中的基础读物--Software Development as a Cooperative Game，详细描述了这样一些观点：软件开发是一种合作游戏，方法学是一种合作文化，以及方法学系列化。

该书分别对方法学的不同侧面、技术与活动、工作产品和标准进行了讨论。

用例所需要讨论的精髓出现在本书的1.2节“你的用例不能作为我的用例”。

本书是一本技术指南，描述了用例编写过程。

尽管你可以在几乎所有项目中使用这些技术，但是必须根据每个项目的实际需要选择模板和编写标准

。

<<编写有效用例>>

内容概要

Jolt大奖素有“软件业之奥斯卡”的美称，本丛书精选自Jolt历届获奖图书，以植根于开发实践中的独到工程思想与杰出方法论为主要甄选方向。

本书作者Alistair

Cockburn，凭借自己在面向对象领域的丰富经验，并参考其他专家的建议，扩展了典型的用例处理方法，为软件开发人员编写用例提供了一种“基本、具体和实用的”指南。

本书完整地叙述了有关用例的初、中、高级概念，并提供了大量的、正反两方面的用例编写实例，是一本概念清晰、结构完整、内容丰富的专业图书。

本书荣获2001年Jolt世界图书大奖，适用于不同知识层次的软件工作、研究人员和用例编写人员。

<<编写有效用例>>

作者简介

作者：（美国）科伯恩（Alistair Cockburn）译者：王雷 张莉 科伯恩（Alistair Cockburn）是用例方面的一位著名专家。

他是Humans and Technology的资深顾问，在那里他负责帮助雇员在面向对象项目上获得成功。

在保险、零售和电子商务公司，以及在大公司，例如挪威中央银行和IBM，他有二十多年主持硬件和软件开发项目的经验。

他也是Surviving Object—Oriented Projects（Addison—Wesley，1998）一书的作者。

王雷，北京航空航天大学计算机学院副教授，从事操作系统、软件工程和过程工程等方面的研究工作。

曾获部级科技进步二等奖、三等奖各一项。

张莉，北京航空航天大学计算机学院教授，软件工程研究所副所长。

计算机学会软件工程专家委员会委员、教育专家委员会副主任、中国电子学会云计算专家委员会委员、国际信息处理联盟（IFIP）成员、欧洲国际企业互操作虚拟实验室（V—Lab）成员。

<<编写有效用例>>

书籍目录

- 第1章 引言1
 - 1.1 用例是什么 (梗概) 1
 - 用例1 通过网络购买股票 3
 - 用例2 汽车交通事故索赔 5
 - 用例3 对运到的包装箱进行登记 6
 - 1.2 你的用例不能作为我的用例7
 - 用例4 买东西 (非正式版本) 10
 - 用例5 买东西 (完整正式版本) 10
 - Steve Adolph: 在新领域中“发掘”需求14
 - 1.3 需求和用例15
 - 图1-1 “轮轴和轮辐”需求模型17
 - 用例作为项目连接结构18
 - 1.4 用例的增值点18
 - 1.5 合理安排你的精力19
 - 1.6 先用使用叙述做热身21
 - 1.7 练习22
- 第1部分 用例体部分
- 第2章 用例是规范行为的契约27
 - 2.1 具有目标的执行者之间的交互27
 - 执行者具有目标27
 - 图2-1 一个具有目标的执行者请求另一个执行者履行职责28
 - 目标可能失败29
 - 交互是复杂的30
 - 用例聚集场景33
 - 图2-2 条形裤: 成功和失败场景33
 - 图2-3 在条形裤中展示子目标的小条形裤34
 - 2.2 涉及利益的项目相关人员之间的契约35
 - 图2-4 SuD为主执行者提供服务, 同时维护幕后项目相关人员的利益36
 - 2.3 图形模型37
 - 图2-5 执行者和项目相关人员38
 - 图2-6 行为38
 - 图2-7 用例是职责的激发者39
 - 图2-8 作为组合的交互39
- 第3章 范围41
 - 表3-1 “内/外”列表41
 - 3.1 功能范围42
 - “执行者?目标”列表42
 - 表3-2 “执行者?目标”列表43
 - 用例简述43
 - 表3-3 用例简述44
 - 3.2 设计范围44
 - 一个简短而真实的故事45
 - 图3-1 设计范围的大小是任意的46
 - 用图标来突出设计范围46

<<编写有效用例>>

设计范围示例47

(1) 企业系统范围47

用例6 增加新服务 (企业) 48

用例7 增加新服务 (Acura) 49

(2) 一个应用程序对应多台计算机49

用例8 输入和修改请求 (联合系统) 50

用例9 添加新服务 (给Acura添加) 50

用例10 通知新服务请求 (BSSO中) 51

用例11 更新服务请求 (BSSO中) 51

用例12 通知更新后的服务请求 (Acura中) 51

(3) 基本用例51

图3-2 Acura-BSSO的用例图52

图3-3 Acura-BSSO的一组用例图52

用例13 资源的串行存取 53

用例14 实施资源锁转换政策 54

用例15 实施存取兼容性政策 55

用例16 实施存取选择政策 56

用例17 令服务客户等待获得资源存取权限 56

3.3 最外层用例57

3.4 使用范围确定的工作产品59

3.5 练习60

第4章 项目相关人员和执行者61

4.1 项目相关人员61

一个简短而真实的故事62

4.2 主执行者62

主执行者为什么有时是不重要的 (而有时又是重要的) 63

在开始用例编写时64

在用例编写和设计过程中64

设计之后, 准备配置系统时66

执行者与角色66

统一建模语言 (UML) 图和执行者/角色规格说明67

刻画主执行者的特点67

表4-1 “执行者概况” 表示例68

4.3 辅助执行者68

4.4 被讨论系统68

4.5 内部执行者和白盒用例69

4.6 练习69

第5章 三个命名的目标层次71

图5-1 用例层次72

5.1 用户目标 (蓝色, 海平面) 72

一个简短而真实的故事74

蓝色的两个层次74

5.2 概要层次 (白色, 云朵/风筝) 75

用例18 操作保险单 + 75

重温最外层用例的内容76

5.3 子功能 (靛青色/黑色, 海平面以下 蛤) 77

目标层次总结78

<<编写有效用例>>

- 5.4 利用图标来突出目标层次78
- 5.5 找出正确的目标层次79
 - 找出用户目标80
 - 提升和降低目标层次80
 - 图5-2 通过问“为什么”的问题来转换层次81
- 5.6 一个较长的编写实例：“处理索赔”的多层次示范81
 - 用例19 处理索赔（业务）82
 - 用例20 评估工作补偿索赔84
 - 用例21 处理索赔（系统）+86
 - 用例22 损失注册88
 - 用例23 查找……（问题陈述）92
- 5.7 练习93
- 第6章 前置条件、触发事件和保证95
 - 6.1 前置条件95
 - 6.2 最小保证97
 - 6.3 成功保证98
 - 6.4 触发事件99
 - 6.5 练习100
- 第7章 场景和步骤101
 - 7.1 主成功场景101
 - 常见的环境结构101
 - 场景主体103
 - 7.2 执行步骤104
 - 准则104
 - 准则1：使用简单的语法104
 - 准则2：明确地写出“谁控制球”105
 - 准则3：从系统外部的角度来编写用例105
 - 准则4：显示过程向前推移106
 - 准则5：显示执行者的意图，而不是动作107
 - 准则6：包含“合理”的活动集108
 - 图7-1 一个事务由4个部分组成109
 - 准则7：“确认”而不是“检查是否”110
 - 准则8：可选择地提及时间限制111
 - 准则9：习惯用语：“用户让系统A与系统B交互”111
 - 准则10：习惯用语：“循环执行步骤x到y，直到条件满足”112
 - 编号或不编号113
 - 7.3 练习114
- 第8章 扩展117
 - 8.1 扩展的基础117
 - 8.2 扩展条件118
 - 集中讨论所有可能的失败和可选择的过程120
 - 准则11：用“检测到什么”的方式来编写条件121
 - 一个真实的、令人不快的小故事122
 - 关于集中讨论列表123
 - 扩展列表的合理化123
 - 逐层合并失败124
 - 8.3 扩展处理125

<<编写有效用例>>

准则12：条件处理的缩排方式127
失败的嵌套128
从扩展中创建新用例129
8.4 练习130
第9章 技术和数据的变化131
图9-1 在UML中使用具体化方式表现技术变化132
第10章 连接用例133
10.1 子用例133
10.2 扩展用例133
图10-1 扩展用例的UML图135
什么时候使用扩展用例136
10.3 练习137
扩展用例137
第11章 用例格式139
11.1 供选择的格式139
完整正式的用例格式139
用例24 完整正式的用例模板<名字>139
非正式用例格式140
用例25 实际登录（非正式版本）140
单列表格格式141
表11-1 用例的单列表格格式141
双列表格格式142
表11-2 双列表格142
RUP格式143
用例26 登记课程 144
条件语句格式147
Occam格式147
图形方式148
UML用例图149
11.2 影响用例书写格式的因素149
矛盾的因素：业务环境、社会作用、不同文化150
理解层次150
项目相关人员的要求150
经验与格式151
覆盖面151
一致性151
复杂度152
冲突152
完整性152
目标与任务——完成什么与怎样完成153
资源153
其他因素153
11.3 5种项目类型的标准153
需求了解阶段用例154
用例27 需求了解用例模板——Oble a New Biscum 154
业务过程建模用例155
用例28 业务过程用例模板——Symp a Carstromming 155

<<编写有效用例>>

- 确定系统需求用例规模156
- 用例29 确定系统需求用例规模模板——
Burple the Tramling 156
- 短期、高强度的项目用例157
- 用例30 高强度项目用例模板——Kree a Ranfath 157
- 详细功能需求用例158
- 用例31 用例名字——Nathorize a Permion 158
- 11.4 总结159
- 11.5 练习159
- 第2部分 经常讨论的主题
- 第12章 什么时候才算完成163
- 关于“正在完成”164
- 第13章 扩展到多个用例165
- 简单描述每个用例（低精度表示）165
- 创建用例簇165
- 第14章 CRUD和参数化用例167
- 14.1 CRUD用例167
- 用例32 管理报表用例 168
- 用例33 存储报表用例 170
- 14.2 参数化用例173
- 第15章 业务过程建模177
- 15.1 建模与设计177
- 从核心业务178
- 图15-1 核心业务黑盒179
- 图15-2 白盒用例中的新业务设计179
- 从业务过程到技术179
- 图15-3 白盒用例中的新业务设计（又一次）180
- 图15-4 带黑盒系统用例的新业务过程180
- 从技术到业务过程181
- 15.2 业务用例和系统用例181
- Rusty Walters：业务建模和系统需求183
- 第16章 遗漏的需求185
- 16.1 数据需求的精度186
- 16.2 从用例到其他需求的交叉链接188
- 图16.1 翻新图1.1，“轮轴和轮辐”需求模型188
- 第17章 用例在整个过程中的作用191
- 17.1 用例在项目组织中的作用191
- 通过用例标题进行组织191
- 表17-1 规划表示例192
- 一个真实的小故事192
- 跨版本处理用例193
- 交付完整场景194
- 一个短而真实的集成实例194
- 17.2 从用例到任务或特征列表194
- 用例34：获得折扣 196
- 表17-2 “获得折扣”任务列表197
- 17.3 从用例到设计197

<<编写有效用例>>

- 一个真实的小故事199
- 面向对象（OO）设计者特别注意199
- 17.4 用例到用户界面（UI）设计201
- 17.5 用例到测试用例202
- 用例35：订购商品，产生发货单（测试例子）202
- 表17-3 主要成功场景测试（好信用）203
- 表17-4 主要成功场景测试（坏信用）203
- 17.6 实际用例编写203
- 分工合作过程204
- 第1阶段：制定粗略的系统功能图204
- 第2阶段：制定详细用例视图206
- 用例需要的平均时间208
- 从大型团队中收集用例208
 - Andy Kraus：从庞大、不同层次的团队收集用例209
- 第18章 用例概述和极端编程213
- 第19章 错误改正215
 - 19.1 没有系统215
 - 19.2 没有主执行者216
 - 19.3 过多的用户接口细节217
 - 19.4 过低的目标层次218
 - 19.5 目标和内容不符220
 - 19.6 用户接口描述过多的改进实例221
- 用例36：寻找一种解决方案——修改前 221
- 用例37：寻找可能的解决方案——修改后 226
- 第3部分 对忙于编写用例的人的提示
- 第20章 对每个用例的提示233
 - 提示1：每个用例都是一篇散文233
 - 提示2：使用例易于阅读233
 - 提示3：仅用一种句型234
 - 提示4：“包含”子用例235
 - 提示5：谁控制着球235
 - 提示6：正确地得到目标层236
- 图20-1 问“为什么”以提高层次237
- 提示7：不考虑GUI237
- 提示8：两个结局238
- 提示9：项目相关人员需要的保证238
- 提示10：前置条件240
- 提示11：对用例进行通过/失败测试240
- 表20-1 对用例进行通过/失败测试241
- 第21章 对用例集的提示243
 - 提示12：一个不断展开的故事243
 - 提示13：业务范围和系统范围244
 - 提示14：核心价值 and 变化244
- 核心价值245
- 适当的改变246
- 不合适的改变247
- 提示15：用例集中的质量问题248

<<编写有效用例>>

第22章 处理用例的提示249

提示16：仅仅有3章（第4章在哪儿呢？）249

提示17：首先向广度上努力249

图22-1 工作随着细化而增加250

提示18：12步秘诀251

提示19：认识到错误的开销252

提示20：喜欢蓝色牛仔裤252

提示21：处理失败情况253

提示22：前期和后期的工作标题254

提示23：执行者扮演角色255

提示24：大的图画恶作剧255

图22-2 “妈妈，我想回家。” 256

图22-3 椭圆图形式的语境图257

表22-1 语境图的执行者?目标列表257

提示25：大型工具的争论257

提示26：使用标题和简介的项目计划259

附录A UML中的用例261

A.1 椭圆和“小人”图符261

A.2 UML中的包含关系262

图A-1 包含关系的画法262

准则13：将高层目标画得高一点263

A.3 UML的扩展关系263

图A-2 扩展关系的画法264

准则14：将扩展用例画得低一些264

准则15：用不同的箭头形状264

正确地使用扩展265

图A-3 扩展一个基用例的三个中断用例265

扩展点266

A.4 UML的泛化关系267

正确地使用泛化关系267

图A-4 泛化关系的画法。

268

准则16：将泛化目标画得高一点268

泛化的危害269

图A-5 泛化的危害——终止大交易269

图A-6 改正后的终止大交易270

A.5 从属用例与子用例270

A.6 用例图的画法271

准则17：语境图中的用户目标271

准则18：将支持执行者放在右边271

A.7 代之以编写基于文本的用例272

附录B 部分习题的答案273

第3章273

练习3.1273

练习3.2273

<<编写有效用例>>

- 第4章274
- 练习4.2274
- 练习4.3275
- 第5章275
- 练习5.1275
- 练习5.2276
- 第6章276
- 练习6.1276
- 练习6.4277
- 第7章277
- 练习7.1277
- 练习7.2278
- 练习7.4278
- 用例38 使用订单处理系统 279
- 第8章279
- 练习8.1279
- 练习8.5280
- 用例39 在网上买股票 280
- 第11章281
- 练习11.1281
- 用例40 执行清洁火花塞服务 281
- 附录 C 术语表283
- 主要术语283
- 用例类型 (Use Case Type) 285
- 图形286
- 附录D 参考读物289
- 本书参考了以下书籍289
- 本书参考了以下文章289
- 有用的在线资源290
- 索引291

<<编写有效用例>>

章节摘录

版权页：插图：可以采用多种方法来处理这个角色片段，每种方法都各有优缺点。哪种方法都没有明显的优势，你从中选择其一就可以了。

可选策略1。

根据主执行者担当的角色来对它们进行分解。

创建一个“执行者—角色”表，在表中列举出在任一用例中充当主执行者的所有不同的人 and 系统，以及他们担当的所有角色。

在主执行者域使用角色名称。

使用“执行者—角色”表将用例与现实世界中的人和系统对应起来。

这种策略能使编写者不必顾及错综复杂的工作头衔，而只专注于继续用例的编写工作。

有些人，或许是用户界面设计者或软件打包人员，会利用“执行者—角色”表来将用例同其最终用户对应起来。

可选策略1带来的一个不便之处是需要单独维护和阅读一个列表。

可选策略2。

在用例部分前的某个地方，书写：“经理可以执行职员可以执行的任何用例，并且经理还可以执行其他更多的用例。

区域经理可以执行经理可以执行的任何用例，并且区域经理还可以执行其他更多的用例。

因此，每当我们写主执行者是（例如）职员时，应该理解为任何职位比职员高的人——本例中，经理和区域经理都可以执行该用例。

”这种做法比“执行者—角色”表容易维护，因为它变化的可能性不大。

这种方法的缺点是，人们需要花更多的时间来互相提醒当职员作为主执行者时，经理也可以运行该用例。

利用这两种策略，人们都可以获得足够好的结果。

但要说到值不值的问题，我们会采用第2种策略，因为这样可以少去编写、审查和维护一项工作产品。

。

<<编写有效用例>>

编辑推荐

《编写有效用例》完整地叙述了有关用例的初、中、高级概念，并提供了大量的、正反两方面的用例编写实例，是一本概念清晰、结构完整、内容丰富的专业图书。有了这本书作为指南，你将了解编写用例的要点，提高你编写用例的技巧，并且为你下一个项目提供一个有效使用用例的方法。

<<编写有效用例>>

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>