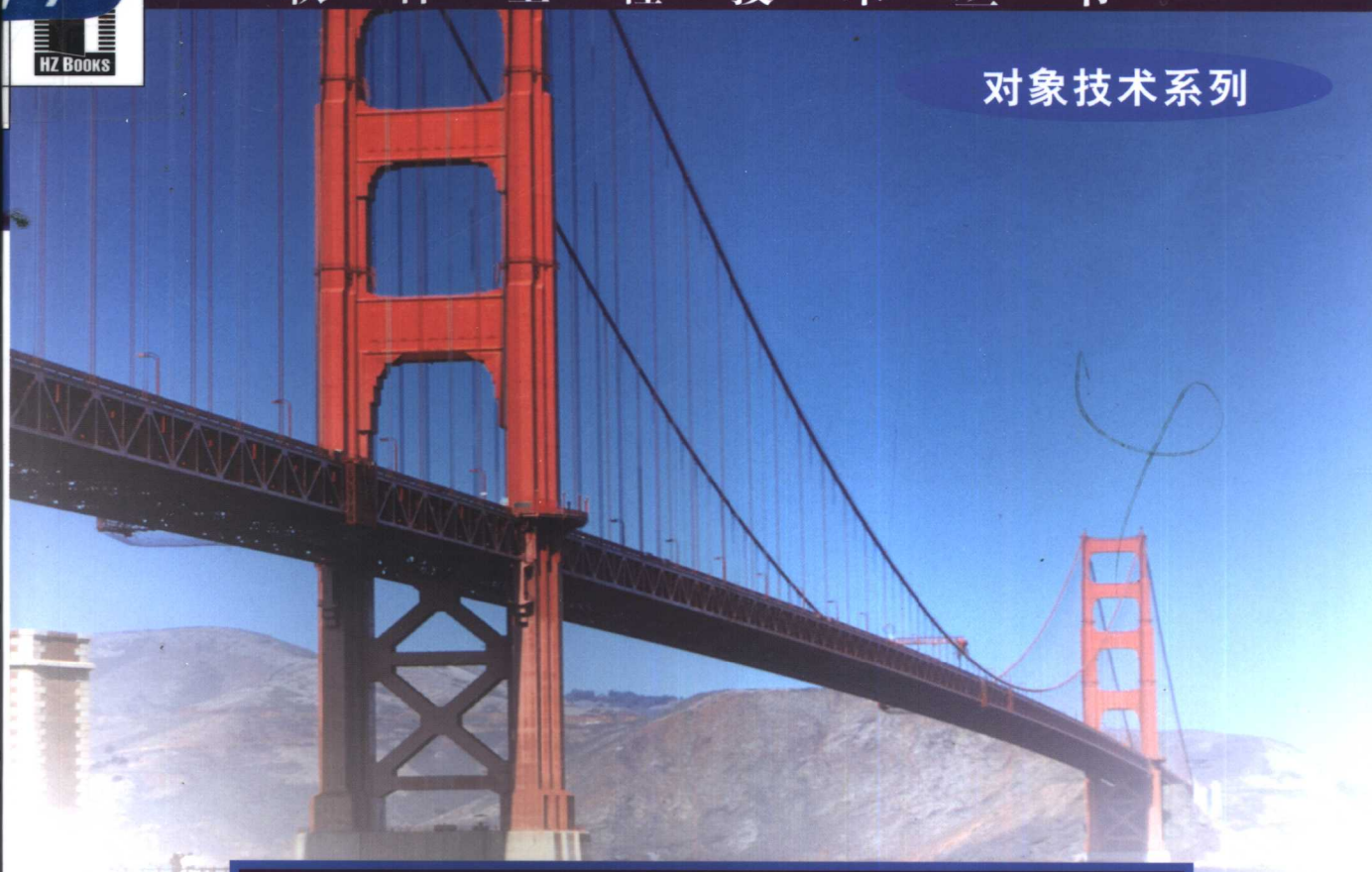
软 件 工 程 技 术 丛 书

HZ BOOKS

对象技术系列



面向对象方法原理与实践

（原书第3版）

Object-Oriented Methods Principles & Practice

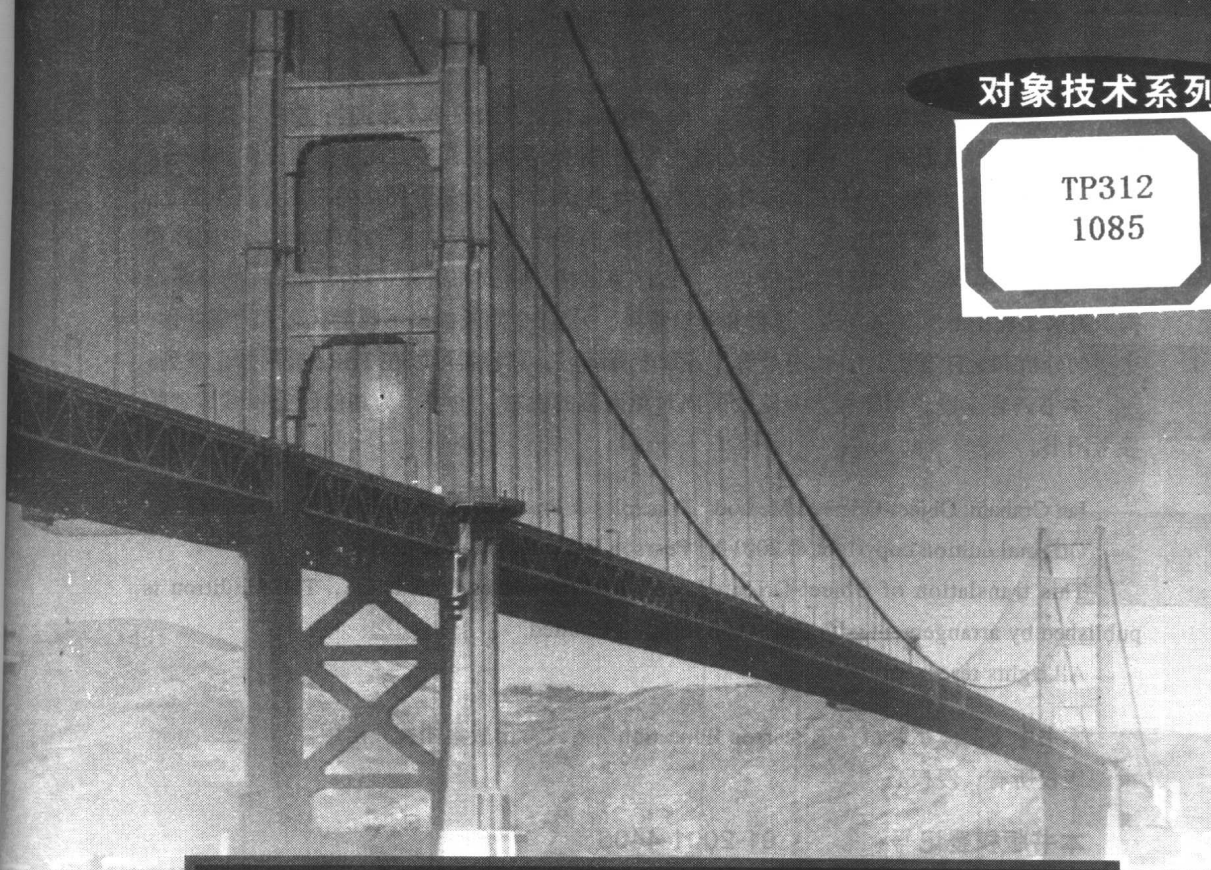
(Third Edition)

Ian Graham 著　　袁兆山　等译

机械工业出版社
China Machine Press

软 件 工 程 技 术 丛 书

对象技术系列



面向对象的方法原理与实践

(原书第3版)

Object-Oriented Methods Principles & Practice

(Third Edition)

Ian Graham 著　　袁兆山　等译

机械工业出版社
China Machine Press

本书是面向对象领域的经典名著之一，将面向对象方法的基本原理与软件工程实践很好地结合起来，覆盖面广，可读性强，是一本集理论与实践及其多方面应用于一体的好书。

本书全面、准确地阐述了面向对象方法。全书分10章，分别介绍面向对象的基本概念、面向对象程序设计和方法、面向对象和基于对象的程序设计语言、分布式计算、中间件和迁移、数据库技术、面向对象分析与设计、体系结构、模式和组件、软件和系统的体系结构、需求工程中的SOMA方法、过程和项目管理。另有3个附录给出不确定状态下的继承性、主要的分析和设计方法、UML符号等。各章配有练习，其答案可以在 TriReme网站上找到。

本书内容丰富，可作为大专院校面向对象课程的教材，也可作为其他研究领域读者的参考用书。

本书版权登记号：图字：01-2001-4400

图书在版编目（CIP）数据

面向对象方法：原理与实践/格雷厄姆（Graham, I.）著；袁兆山等译. -北京：机械工业出版社，2003.3
（软件工程技术丛书）
书名原文：Object-Oriented Methods Principles & Practice, Third Edition
ISBN 7-111-11186-9

Ⅰ.面… Ⅱ.①格…②袁… Ⅲ.面向对象语言－程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字（2002）第092139号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）
责任编辑：杨 文
北京瑞德印刷有限公司印刷·新华书店北京发行所发行
2003年3月第1版第1次印刷
787mm×1092mm 1/16·42.75印张
印数：0 001-5 000册
定价：69.00元

译者序

OO（面向对象，Object-Oriented）方法，其本质是强调从客观世界固有的事物出发来构造系统，用人类习惯的思维方式来认识世界、理解和描述客观事物，强调最终建立的软件系统能够映射问题域，对象及对象之间的关系能够如实反映问题域中事物及其关系。尽管OO思想在某种程度上已经非常成功，OO技术风靡全球，涵盖分析、设计、编码，以至整个软件工程，C++、VB、VC、J++等OO语言正取代早期的结构化语言。但是，在很多领域，OO技术仍然不够成熟，OO技术的理论落后于实践。从OOPL开始就缺乏形式化描述工具，已经发布的OOSD可用的规范，例如UML等，还需进一步完善，OOA的许多问题还有待研究和解决。需要人们不断地探索和研究。我们认为，只有对OO方法与技术以及OO原理与实践的来龙去脉进行彻底的研究，才能从真正意义上对OO技术有一个全局的了解。

本书是Ian Graham从软件开发角度全面介绍OO软件开发方法：原理与实践的著作。Ian Graham是一位国际公认的对象技术和业务过程建模方面的顾问专家。他有着超过20年的IT实践者和产业顾问的经验，是TriReme 国际有限公司当前的主要顾问。他在OO软件开发方法、专家系统和商务建模方面造诣颇深，并具有丰富的实践经验。本书的内容丰富，把OO方法、基本原理与软件工程实践联系起来，覆盖面广，可读性好，是一本协调方法、理论与实践以及多方面应用的书。

本书面向整个对象技术领域，全面、详细、准确地阐述了面向对象方法，共10章、3个附录。它涵盖面向对象的基本概念、面向对象程序设计、面向对象设计、面向对象分析、面向对象数据库和其他相关技术，并反映最新的技术和方法，广泛涉及到分布式计算、体系结构、中间件和迁移、模式和组件、Java和UML等新思想，揭示了运用规则集合思想的面向对象概念建模、需求工程以及开发过程SOMA（语义对象建模方法）。本书通篇使用UML,并基于Catalysis和SOMA描述了最好的面向对象分析和设计实例。在关于面向对象方法的各章节自始至终将SOMA与Catalysis结合在一起，并使用UML符号。它将对象建模解释为一种普通的知识表示形式，而不只是一种描述计算机程序的方法。它提供单一的资料和全面的、独立于语言的介绍，无论从开发者，还是从管理者的角度，都涉及到了对象技术的各个方面。本书从概念模型设计者的角度，而不是从程序员或设计者的角度，强调在商业环境中使用面向对象技术的实际问题。它将面向对象、人工智能和数据建模结合起来处理IT中的主要问题。它介绍并评价面向对象的语言、中间件、数据库及方法，同时将它们分别与传统技术联系起来，首次简练地解释功能强大的基于组件开发的Catalysis方法（D'Souza and Wills, 1999）。以足够的深度向学生和专业人员提供充分的参考资料，以引导其进入该领域。

本书更深远的目标旨在清楚、有条理地回答如下问题：什么是面向对象的方法？其优点、缺陷及可能的代价是什么？有哪些可用的语言、方法和工具，它们究竟如何？开始采用这项技术时，需要做些什么？面向对象分析和设计方法的任务和作用是什么？如何获得面向对象系统

的需求？如何运用面向对象？需要什么特别技巧？与信息技术(IT)的其他领域有何联系？什么才算是合适的应用？等等。本书在附录中给出了模糊对象：不确定状态下的继承性、主要的分析和设计方法、UML符号等。各章都有练习，其答案可以在 TriReme网站上找到。

本书主要由合肥工业大学计算机与信息学院袁兆山教授翻译。参加翻译工作的还有合肥工业大学的苗沛荣、张艳明、袁晓靖、刘珊珊、袁晓辉、张国宁、张健、钟金琴、李莹莹、方睿、李心科、李向上、刘栋、王珏、李宏芒等。刘宗田、张艳明、袁晓辉参加审校，由袁兆山教授审校定稿。

由于水平有限，加之时间仓促，译文难免有不足之处，欢迎广大读者不吝指正。

译　者

2002年6月于合肥

第3版前言

他放下手中的书，双脚伸到壁炉中的灰烬上，双手合抱于头后，兴奋之余，他已不局限于考虑某个特定对象，而是考虑这本书对我们的作用。

乔治·埃利奥特《Middlemarch》

当本书第1版于1991年问世时，面向对象技术引起了IT界和学术界的极大兴奋和兴趣，几乎每周都举行公众的研讨会。人们创建了新的刊物和例会，特别兴趣小组的成员人数迅速增加。第2版问世时，人们的兴趣已达到顶峰，并且许多商业组织已对这种技术的优劣有了初次体验。尽管还有人对这种技术大惊小怪，但随着（面向对象）语言的逐步稳定，即使统一建模语言(UML)的出现已经在很大程度上解决了表示法方面的争论，人们依然对方法和生命周期问题愈发狂热。目前人们所关心的问题主要集中在跨企业应用集成和基于组件的开发上。自从第2版问世以来，对象技术(OT)可能已经在出版的著作和涉及的主题的规模和范围上增长了三倍。当我开始着手写第3版时，其前景是令人畏惧的，因为此书的初衷只是提供一个全面的概括，而不是将它写得过于庞大。另一方面，许多新开发的特征只是在1994年左右出现的主题上的变化：Jim Coplien 的C++思想广受欢迎，初步暗示着人们对设计模式兴趣的巨大探索。早期的基于对象管理组(OMG)体系结构模型的对象请求代理已经成熟，并已形成产品；新的、更好的面向对象程序设计语言已经出现；面向对象数据库现已进入日常的商业应用——尽管范围还很有限。除此之外，在分析和设计方法领域还有着彻底的变化。这就是我重写本书的重要原因。

主题

实际上本书是对整个对象技术领域的综述。它涵盖面向对象程序设计、面向对象设计、面向对象分析、面向对象数据库和若干相关技术。现在有许多涉及到特定语言和方法的关于对象技术的好书，然而对于更广范围的知识，它们只是一带而过。这些书在总体上高度地审视面向对象的原理和好处，但是使用了大量的尤其是关于程序设计的材料，这会使读者感到迷惑，况且这些书依赖于特定语言的语法。另一个极端是，现在有一些很好的针对管理人士的综述性图书，但是对专业人员和学生们而言，这些通常不够深入。如果读者想深入理解面向对象中那些与程序设计无关的方面，他将不得不求助于研究文献、会议记录、大量的专论或者高科技论文。如果读者想快速地对整个领域作大致的了解，或者评估对象技术将来的作用，却苦于没有连贯的、全面的资料。因此，我的目标旨在通过以下途径填补文献上的缺陷。

- 提供单一的资料和全面的、独立于语言的介绍，无论从开发者还是从管理者的角度，都涉

及到了对象技术的各个方面。

- 本书主要是从概念模型设计者的角度，而不是从程序员或设计者的角度，强调在商业环境中使用面向对象技术的实际问题。
- 宣传如下观点：必须将面向对象、人工智能和数据建模结合起来（而不是各自分开），来处理IT中的主要问题。
- 介绍并评价面向对象的语言、中间件、数据库及方法，同时将它们分别与传统技术联系起来。特别是首次简练地解释功能强大的基于组件开发的Catalysis方法(D'Souza and Wills, 1999)。
- 在保持乐观地评价实际应用中的对象技术的时候，试图戳穿一些有关对象技术的神话。
- 以足够的深度向学生和专业人员提供充分的参考资料，以引导其进入该领域。

本书更远的目标与前两版相同，旨在明确地回答如下问题。

- 什么是面向对象的方法?
- 其优点、缺陷及可能的代价是什么?
- 有哪些可用的语言、方法和工具，以及人们对它们的评价如何?
- 开始采用这项技术时，应做些什么?
- 面向对象分析和设计方法的作用是什么?
- 如何获得面向对象（OO）系统的需求?
- 如何运用面向对象?
- 需要什么特别技巧?
- 与信息技术(IT)的其他领域有什么联系?
- 什么是合适的应用?

在完成上述目标的同时，本书还揭示了我在运用规则集合思想的面向对象概念建模、需求工程以及开发过程——以上统称为语义对象建模方法（Semantic Object Modelling Approach, SOMA)上所做的初步工作。在关于方法的所有几章里，我自始至终将SOMA与Catalysis结合在一起，并使用了UML表示法。在阅读这部分材料时，读者应意识到我的方法的特色，它将对象建模解释为一种普通的知识表示形式，而不只是一种描述计算机程序的方法。

第3版的主要变化

第3版对第2版进行了实质性的修改和扩展，反映了自前两版问世以来在该领域所发生的主要变化。自我写第1版以来，人们如此迅速接受了对象技术，还有那些万分热情地为这项技术作宣传的人，都令我感到吃惊。所以在过去三年左右的时间中，该技术所发生的迅速变化就不再那么令人吃惊了。不但产品变化了，方法的数量增多了，而且专业人员，实际上只要是一个细心观察的人，都会得出与他们在1991年或1994年所得出的相差甚远的结论。一个最显著的变化是整个业界普遍地接受OMG及其为对象技术颁布的各种标准。另一方面，还有许多方面没有变化。因此，尽管这本书的目标没有变，但实现的手段必须在本质上有所不同。这一版含有所有

最新的确定材料，并符合最新出现的标准，改进了可用的产品和方法的描述，而且在新的事实的基础上得出了新的结论。

主要变化如下：与本书第1版相比，第1章稍作修改，以反映出业界的术语已变得更明确、更规范。这一章得益于我讲授的课程和向许多人传授这个主题的过程中所学会的教学技巧，我希望它们将会更好、更成熟。新的第4章包括中间件和迁移策略，更多的内容是关于OMG标准的。第5章中关于面向对象数据库的内容已经完全被更新了，以反映在该领域出现的日渐成熟的新产品。这部分内容的最大变化是关于面向对象分析和设计及其管理。至于大约六年前就存在的50种或更多种方法，它们的综述已被归入附录中，这主要是由于这些方法具有显著的历史价值。本书现在通篇使用UML，基于主要Catalysis和SOMA的角度，第6章和第7章描述了最好的面向对象分析和设计实践。用一个新的附录总结UML表示法。新的第7章包括软件体系结构、模式和基于组件的开发。第8章详细描述了需求工程中的SOMA方法。关于管理的第9章，为了更加便于清楚地解释，以及为了对推荐的开发过程提供更明确的描述，故作了相当大的重新组织。这部分现在包括了关于用户界面设计的指导。所有其他各章和附录A都稍作修订和改进，以反映该领域新的发展，并修正了作者在第2版中发现的错误。

这几年我逐渐地意识到，为了帮助众多的读者，我在大多数章的末尾增加了练习。精选的答案可以在 TriReme站点找到。我知道那里有答案。参考文献进行了很大的扩充，以反映这一版中参考文献数量和新内容的大量增加。词汇表也被更新和改进。

尽管存在这些大的改动，但本书的主要目标依然未变，我只是希望本书与它未作修改之前相比，是一本更全面的、详细的、最新的和准确地概述面向对象方法的书籍。

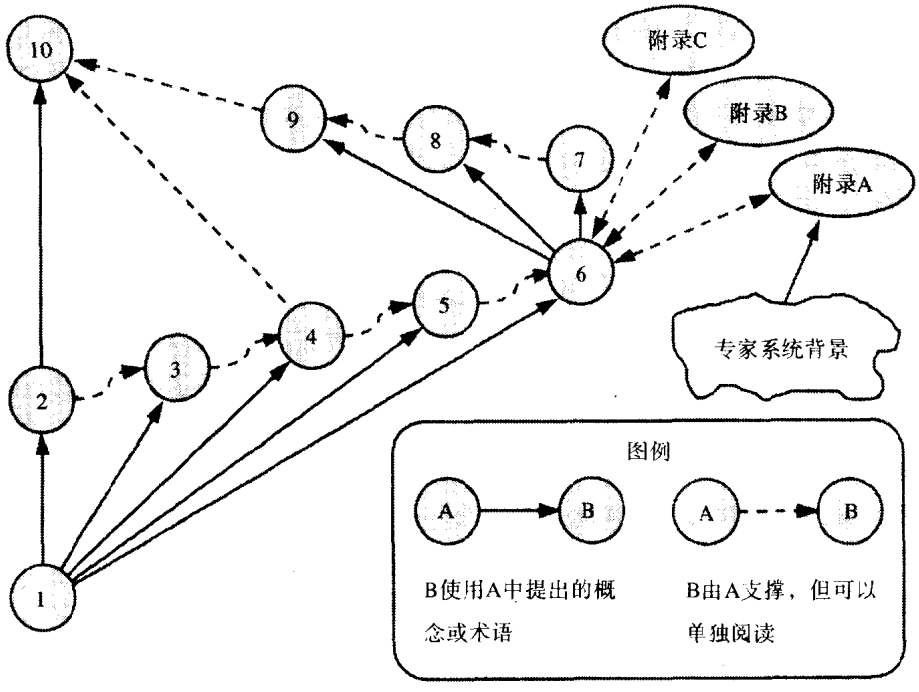
适用读者

本书适用于没有高深的计算机科学理论知识或数学知识的读者，但同时，也试图以一定的深度准确处理重要的问题。本书提供了关于如何最有效地在实际工作中运用面向对象方法的建议。

本书早期版本的主要读者是从事IT和DP(数据处理)职业的人：软件工程师和一般的使用计算机工作的人员，无论是用户机构、咨询服务机构、大学，还是制造商。尽管这种状况在这一版还存在，但很明显，这本书已在大学中拥有一批忠实的读者。在那里，本书被用做讲述对象技术课程的大学生的入门教材，开设在信息技术或软件工程课程之内，可能还需要补充一门关于面向对象程序设计的课程。本书对教授计算机科学、商务系统分析和人工智能的老师大有用处。研究人员将会对这本综述书籍及其原稿感兴趣。他们可能还会发现散列本书的一些评论是令人不解的，甚至是自相矛盾的。管理者和项目规划者应该阅读本书，以理解这项技术是如何影响他们的业务实践并能对变动作更有效的规划。咨询人员、项目经理、系统分析员和设计员可以阅读本书以进行评估，并跟上这项技术的发展，同时，我希望他们在日常工作中使用该技术。程序员应该阅读这本书，以拓宽他们在这个领域的视野。

在本书成型的过程中，本书的内容已在各种会议、公众研讨会、内部培训课程等多种场合陆续呈现给了读者。

阅读图



这本书提供的内容有多种可选的阅读途径。在阅读图中表示了两种主要的阅读步骤。管理人员和那些欲从高层来审视这个课题的人应该走“高的路径”，阅读第1章、第2章和第10章，项目经理还应该阅读第9章。对分析方法感兴趣的人应该走“低的路径”，因为第6至第8章是以前面所有几章为基础的。然而，即使是对面向对象程序设计相当熟悉的人，也应该阅读第1章，因为该章介绍了一些不同于其他著作所介绍的术语。这些术语强调了我已经提及的概念模型观点。

如果读者依序阅读本书，我希望这本书让人感觉是在讲述一个故事。务必注意留心一些关键的论题。这些构成了系统设计人员与概念建模者之间观点的差异：前者需要面向对象以借鉴吸收其他计算领域的技术和思想，而后者需要向已经被人们精心理解的面向对象的优点（可重用性和可扩展性）增加更加丰富的语义。

Ian Graham
2000年11月于Balham
(ian@trireme.com)

目 录

译者序
第3版前言
第1章 基本概念 ……1
1.1 历史背景 ……2
1.2 什么是面向对象方法 ……7
1.3 基本术语和思想 ……7
1.3.1 抽象与封装 ……12
1.3.2 继承 ……19
1.3.3 封装、继承和面向对象 ……24
1.4 小结 ……26
1.5 书目注释 ……27
1.6 练习 ……27
第2章 面向对象程序设计和方法的好处 …29
2.1 好处 ……30
2.2 一些问题和缺陷 ……42
2.3 实例研究 ……45
2.4 采用策略 ……46
2.5 小结 ……49
2.6 书目注释 ……51
2.7 练习 ……51
第3章 面向对象和基于对象的程序设计语言……52
3.1 面向对象语言 ……52
3.1.1 Simula ……52
3.1.2 Smalltalk及其同源语 ……54
3.1.3 C扩展 ……56
3.1.4 Eiffel……59
3.1.5 Java ……61
3.1.6 Object-COBOL ……62
3.2 其他具有面向对象特征的语言 ……63
3.3 函数式语言和应用式语言 ……64
3.4 基于AI的系统 ……68
3.4.1 Lisp扩展 ……69
3.4.2 其他基于人工智能的开发系统 ……71
3.5 对象库、应用框架和面向对象的第四代语言 ……72
3.6 其他的开发技术 ……74
3.6.1 其他语言 ……74
3.6.2 类型理论和面向对象程序设计 ……76
3.6.3 通过传统语言来实现面向对象程序设计 ……77
3.7 选择一种面向对象语言 ……78
3.8 方向和趋势 ……79
3.9 小结 ……80
3.10 书目注释 ……82
3.11 练习 ……83
第4章 分布式计算、中间件和迁移……84
4.1 分布式计算和客户/服务器计算 ……85
4.1.1 网络和体系结构的问题 ……91
4.2 对象请求代理和中间件 ……93
4.2.1 XML的角色 ……100
4.3 企业应用集成 ……101
4.4 迁移策略 ……104
4.4.1 面向对象的系统与传统IT的协同工作 ……105
4.4.2 用于包装的数据管理策略 ……107
4.4.3 迁移的实际问题 ……109
4.4.4 重用现有的软件组件和软件包 ……110
4.4.5 用面向对象分析作为一个出发点……111
4.4.6 面向对象的分析和基于知识的原型设计 ……113
4.4.7 对象技术本来是一种迁移策略 ……114
4.5 小结 ……116
4.6 书目注释 ……118

4.7 练习 ……119
第5章 数据库技术 ……120
5.1 数据模型的断续历史 ……120
5.1.1 早期数据库的缺点 ……122
5.1.2 关系模型及其作用 ……125
5.1.3 语义数据模型和数据分析方法 ……134
5.2 关系模型的缺点 ……139
5.2.1 规范化 ……140
5.2.2 完整性规则和业务规则 ……141
5.2.3 空值 ……141
5.2.4 抽象数据类型和复杂对象 ……141
5.2.5 递归查询 ……142
5.3 实体-关系数据库和演绎数据库 ……142
5.3.1 实体-关系数据库 ……143
5.3.2 演绎数据库 ……143
5.4 对象-关系数据库 ……144
5.5 查询语言 ……147
5.6 什么是面向对象的数据库 ……148
5.7 面向对象数据库的好处 ……154
5.7.1 使用面向对象程序设计所带来的好处 ……154
5.7.2 丰富语义能力所带来的好处 ……154
5.7.3 面向对象数据库本身的好处 ……155
5.7.4 使用面向对象数据库方面的问题 …157
5.8 OODB产品综述 ……158
5.8.1 商用面向对象数据库 ……159
5.8.2 其他有影响的产品和项目 ……164
5.9 对象数据库的参照完整性 ……167
5.10 面向对象数据库的应用 ……169
5.11 战略性考虑 ……172
5.12 小结 ……172
5.13 书目注释 ……174
5.14 练习 ……176
第6章 面向对象分析与设计 ……177
6.1 面向对象分析与设计方法的历史 ……177
6.2 软件工程 ……181
6.2.1 职责驱动方法与数据驱动方法 ……185
6.2.2 翻译式方法与细化方法 ……186
6.3 使用UML的面向对象分析与设计 ……186
6.3.1 对象结构 ……190
6.3.2 使用用况来发现类型 ……197
6.3.3 不变量与规则集 ……203
6.3.4 不变量和封装 ……212
6.3.5 状态模型 ……219
6.3.6 转向组件设计 ……222
6.3.7 设计过程 ……228
6.3.8 编制模型文档 ……229
6.3.9 实时扩展 ……229
6.4 标识对象 ……231
6.4.1 知识与分类理论的基本原理 ……233
6.4.2 任务分析 ……236
6.4.3 Kelly网格 ……240
6.5 CASE工具 ……243
6.6 小结 ……244
6.7 书目注释 ……245
6.8 练习 ……245
第7章 体系结构、模式和组件 ……247
7.1 软件和系统的体系结构 ……247
7.2 模式、体系结构和去耦设计 ……259
7.3 设计组件 ……278
7.3.1 要求灵活性的组件 ……280
7.3.2 大规模的连接器 ……281
7.3.3 将业务模型映像到实现 ……282
7.3.4 业务组件和库 ……283
7.4 小结 ……286
7.5 书目注释 ……287
7.6 练习 ……287
第8章 需求工程 ……289
8.1 需求工程的研究方法 ……289
8.2 需求工程与系统规格说明 ……294
8.3 缩小大型问题的范围——任务网格 ……302
8.4 发现业务目标和优先权 ……304
8.5 代理、会话和业务过程 ……305
8.5.1 业务过程模型 ……306

8.5.2 活动图和业务过程建模 ……311
8.6 从会话到任务及用况 ……312
8.7 从任务对象模型到业务对象模型 ……320
8.8 无缝性 ……325
8.9 用况生成的三段论模式 ……329
8.10 保证场景的完整性 ……330
8.11 任务关联集和顺序图 ……331
8.12 可执行的规格说明和模拟 ……336
8.13 组织和举行需求研讨会 ……338
8.13.1 研讨会接纳的角色 ……339
8.13.2 哪些人员应该参加研讨会 ……340
8.13.3 选择一个场所 ……342
8.13.4 研讨会后勤 ……342
8.13.5 研讨会组织者和助理人员的一览表 ……344
8.13.6 会议助理人员的技巧 ……346
8.13.7 谁应该做会议记录 ……347
8.13.8 举办一个研讨会 ……348
8.13.9 在研讨会的环境中使用面谈技术 …351
8.14 小结 ……352
8.15 书目注释 ……352
8.16 练习 ……353
第9章 过程和项目管理 ……355
9.1 为什么要遵循一个过程 ……355
9.2 一种面向对象方法必须做些什么 ……357
9.3 经典的生命周期模型 ……360
9.3.1 瀑布模型、V模和X模型 ……360
9.3.2 螺旋模型 ……361
9.3.3 喷泉模型和MOSES ……362
9.3.4 分形、海螺和弹子机 ……364
9.4 研讨会、时间框和演化开发 ……364
9.5 过程和产品生命周期模型 ……369
9.5.1 面向对象生命周期模型 ……370
9.5.2 Objectory和RUP ……372
9.5.3 OPEN过程框架 ……374
9.6 一个契约驱动的过程模型 ……374
9.7 契约驱动过程的细节 ……381
9.7.1 项目启动阶段和活动 ……382
9.7.2 需求活动 ……383
9.7.3 分析加工活动 ……388
9.7.4 时间框规划活动 ……390
9.7.5 一个时间框内的开发：构造活动 …392
9.7.6 设计活动 ……394
9.7.7 程序设计活动 ……397
9.7.8 测试活动 ……398
9.7.9 用户评审和UAT活动 ……399
9.7.10 合并、协同、重用和文档编制 ……400
9.7.11 评估和重用评估活动 ……402
9.7.12 实现规划活动 ……405
9.7.13 开发规划和资源规划活动 ……406
9.7.14 领域建模和中心库管理活动 ……409
9.7.15 故障修补活动 ……410
9.7.16 一般的项目管理任务和问题 ……411
9.7.17 项目角色和职责 ……417
9.8 重用管理 ……419
9.9 度量和过程改进 ……422
9.9.1 度量 ……422
9.9.2 过程改进 ……429
9.10 用户界面设计 ……430
9.10.1 设计HCI ……431
9.10.2 认知心理学的基本原理 ……434
9.10.3 HCI设计原则 ……436
9.10.4 用户界面设计的指导方针 ……441
9.11 测试 ……449
9.12 小结 ……450
9.13 书目注释 ……450
9.14 练习 ……452
第10章 应用 ……453
10.1 Web应用 ……453
10.2 其他商业应用 ……455
10.2.1 图形用户界面 ……455
10.2.2 模拟 ……456
10.2.3 地理信息系统 ……456
10.2.4 并发系统和并行硬件 ……458

10.2.5 其他应用 ……460
10.3 专家系统、人工智能和智能代理 ……462
10.3.1 黑板和参与者系统 ……465
10.3.2 神经网络和并行计算 ……467
10.3.3 智能代理 ……471
10.4 前景展望 ……478
10.5 小结 ……484
10.6 书目注释 ……484
附录A 模糊对象：不确定状态下的继承性 ……486
附录B 基本的分析和设计方法 ……515
附录C UML表示法摘要 ……576
术语表 ……586
参考文献与参考书目 ……598
名字索引 ……628
主题索引 ……635

第1章
基本概念

我的目的是让所有的人得到提高，我将及时为此作出贡献。

W.S.Gilbert《日本天皇》

20世纪90年代，“面向对象”在IT界几乎是现代、美德、价值的同义词。90年代末期，“组件”开始取代“对象”，至少对记者来说这只改变了很少的内容。因为我们很可能夸大了这项技术的好处，所以最好能用一个更折中的观点。首先，这本书将用纯理论术语讲述面向对象和基于组件开发的真相。其次，本书将说明在建立有用的计算机系统时怎样用这些抽象的思想推动实际应用的进展。本书将独立于任何特定的程序设计语言，而且避开语法细节，以便在选择一个合适的开发环境和方法对出现的业务和技术问题进行高层次评估。在处理面向对象的事物时，我们还必须意识到，即使现在已经有了一些标准存在，但我们正在进入的仍是一个不受太多标准束缚并且研究仍不够完善的领域。我们不得不考虑当前关心的许多其他领域，例如软件体系结构、分布式开放系统、数据库、计算机辅助软件工程（CASE）技术、专家系统等等，我们偶尔也会进入相对未知的领域。

通常，面向对象的提供者承诺的主要好处是可重用性和可扩展性。这就是说，花很少的功夫就可以把预先写好的组件装配成面向对象系统，而且不需要改动这些可重用的组件就可以很容易地扩展系统。我们将在下一章考察这些好处，并且具体体验面向对象系统在什么范围内能够实际获得这些好处。

本书中使用的面向对象方法（OBJECT-ORIENTED METHODS）不仅仅是指面向对象程序设计，它还包含系统开发的一整套原理：程序设计、知识引导、需求分析、业务建模、系统设计、数据库设计和其他相关问题。贯穿本书的重点将放在基本原理方面，以及如何通过它来解决构造信息系统时出现的问题。例如，重用和扩展大部分代码不会限制可重用性和可扩展性这样的好处；设计和分析文档可以保存在库中，而且可以一次又一次地重用和扩展。当然，进行这些操作的前提是潜在的用户可以很容易地找到这些文档。

我已经说过，面向对象这个词已经被赋予了太多的意思。为了澄清这个问题，在这本书中，将把“面向对象的”、“面向组件的”、“基于对象的”、“基于类的”程序设计、设计和分析区别开来。在后面的几章中我们将会看到，很少有商业系统符合纯面向对象的概念，如果有，那么这个系统可能有其他的缺点。但是面向对象系统确实具有作用，一个系统或语言是不是面向对象的并不重要，重要的是怎样才是面向对象的以及用什么办法实现相关的好处。

本章将介绍面向对象方法的基本概念和术语，我们以一段简短的历史介绍作为开始。第2章将讨论面向对象的方法和程序设计的动机和好处。

1.1 历史背景

面向对象的兴起从整体上反映和概括了计算机发展的历史。在20世纪40年代后期，最早的计算工作就是我们现在所想像的程序设计。后来，设计和分析作为单独的问题产生了。同样，首先引起人们注意的是面向对象程序设计，后来是面向对象设计，近来是面向对象分析。所以本书先讲面向对象程序设计，再讲设计和分析。但是，我们更关心设计和分析。

虽然Ten Dyke and Kunz（1989）声称“民兵”（Minuteman）导弹的设计者早在1957年就使用了初期的面向对象技术，但实际上面向对象程序设计的历史是从1967年在挪威开发的Simula语言开始的，并且随着Smalltalk语言的开发而贯穿于70年代。Simula语言的基础是ALGOL语言和早期的离散事件模拟语言Simula 1。Smalltalk语言几乎把对象的概念奉若神明。这期间有影响的语言包括Alphard（Wulf *et al.*，1976）和CLU（Liskov *et al.*，1977）。值得注意的是，面向对象语言Simula早于任何结构化程序设计的概念。从那时起，很多受这些发展所鼓舞而产生的语言都声称自己能用于“面向对象”。

模拟建模对于传统的第三代语言程序员来说是一个很难的问题。它要求程序员把这些语言中常用的控制函数流改成用复杂对象进行更自然描述的扩展流，这些对象随时会改变状态，并且影响事件。面向对象程序设计中，在对象之间传递的消息能改变对象的状态，它取代了函数流。因此，面向对象程序设计是一个很自然的方法，因为程序的结构直接反映了问题的结构。此外，通常在模拟问题时，对象是什么是很清楚的：街道上的汽车、生产线上的机器。它们是真实的，而不是抽象对象，而且很容易鉴别。但遗憾的是，在商业应用中经常不是这样。

SMALLTALK语言和GUI

随着程序设计语言Smalltalk的到来，面向对象这个词终于进入程序设计语言。Smalltalk大部分是在Palo Alto的Xerox研究中心（PARC）开发的，但是它不仅来源于Simula，而且来源于Alan Kay在Utah大学的博士论文。根据Rentsch（1982）的记录，这篇论文以一个到处都有的小的个人计算机为基础。个人计算机能处理各种信息管理问题，能供各种人使用。它最早的版本是PARC的Flex机器，人们称之为Dynabook。Smalltalk基本上是Dynabook的软件部分，Simula语言的类和继承概念与Lisp的结构化特征对Smalltalk语言的影响很大[⊖]。虽然Smalltalk和Lisp是不同的语言，但是Smalltalk把Simula的类概念和Lisp的函数抽象特性结合了起来。

大约在80年代，人们突然开始对用户界面（UI）感兴趣。最著名的商业先驱Xerox和后来的Apple创造了后来得到广泛应用的WIMP[⊖]界面。Smalltalk语言的许多思想都与这些发展有关。一方面，面向对象程序设计支持这些界面的开发，特别是Apple Lisa和Macintosh。另一方面，WIMP对面向对象语言的风格影响很大。最明显的影响是，存在很多适用于界面开发的程序库对象，而其他领域受到的影响很少。面向对象程序设计成功的一个原因是，这些界面非常复杂，

⊖ Lisp是LISt Processing的缩写。它是John McCarthy 在1958 开发的，早期的人工智能研究使用了它。

⊖ WIMP 代表Windows（窗口），Icons（图标），Menus（有时是Mice）和Pointers（指针），它指的是一种利用了这四者的图形用户界面风格。

建立它们要花很大的代价。如果面向对象代码没有内在的可重用性，那么就不可能如此大规模地建立界面。例如，据估计，Apple Lisa（Macintosh的前身）是超过200人年努力的结果，其中很大一部分是用来开发界面。在这个时期，WIMP的影响无处不在：所有的工作站都逐渐配备了WIMP前端处理机，例如微软的Windows或类似的东西。朝着标准化、开放的、基于UNIX系统的转移也与用户界面有关。其中，OpenLook、OSF Motif 和其他类似产品在用户界面方面的竞争在一段时间以来被看成是市场成功与否的关键决定因素。从这种意义上说，面向对象在1993年已经广泛应用于世界各地。 3

人工智能的影响

从70年代中后期开始，面向对象设计和人工智能研究与开发之间的互相影响引发了人工智能（AI）语言若干有用的扩充，主要是Lisp。因此我们得到了如Lisp和Flavors、Loops以及CLOS（Common Lisp Object System）等这些语言。人工智能程序设计环境（通常是Lisp的扩充）的设计，例如KEE和ART，受面向对象思想的影响很大。基于语义网络和框架的知识表示理论强烈地影响了这些系统。这些表示用模式化对象的网络的形式表达了关于真实世界对象和概念的知识。这些对象可以从更普通的对象那里继承特征。人工智能对面向对象的主要贡献在于，它对继承理论的改进。面向对象方法完全吸收了这些经验。第3章将深入讨论人工智能语言。

连同对并行计算的研究，人工智能世界的另一个研究流派创建了参与者（Actor）的概念。与黑板系统（见Englemore and Morgan，1988）一样，Actor系统（Agha，1986）试图模型化协作工作人员池或专家池。一个Actor是比对象更拟人化的概念，它定义了任务、需求和关于协作的知识。Actor语言常用于实时和并行系统。一个相关的概念是智能agent（代理）。值得注意的是，现代基于组件的开发环境，例如COM+，需要组件知道它们的合作者。第6章和第7章将会涉及这方面的更多知识。

新语言

与商业应用相比，用户界面开发不会引起重大的数据管理问题。由于商业应用从模拟和用户界面设计问题而不是从数据库管理中获得经验，所以在其他类型的应用中使用早期的面向对象语言时会出现性能问题，这导致了新语言的开发，例如Eiffel。现有的有效的常用语言，如Ada、C和PASCAL也进行了扩充。面向对象程序设计语言不能方便地处理反复出现的对象、并行情况等等。开发面向对象数据库就是为了解决这个问题。第5章将详细讨论相关内容。第3章概述了各种面向对象和基于对象的程序设计语言。 4

大量用户的需求，特别是一些重要的财政用户的需求例如美国国防部（DoD），使计算机行业供应商改变作法。整个60年代和70年代，美国国防部一直要求做三件重要的事情：把表示建立在所谓的结构化方法上的软件开发实践的工程方法，可重用的软件组件（模块化）以及开放系统。很多主要的IT供应商采用的系统开发方法可以解决第一个需要。通过使用80年代后期的CASE工具，这些方法的使用达到了顶峰。UNIX、X/Windows和Ada在某种程度上解决了美国国防部关于开放系统的要求。也就是说，无论我们用什么软件和硬件，通过最小的努力就能使系统协同工作。Ada还能提高模块化。后面我们将探讨Ada面向对象的程度，更重要的是将讨论面向对象程序设计是否促进解决以上三个重要问题。面向对象程序设计也致力于美国国防部的三

个要求之一。它有可能使得系统开发者通过利用可重用组件来装配系统，因此能把模块化和软件工程结合在一起。随着各种面向消息的中间件和支持标准的消息代理产品的出现如CORBA（见第4章），互操作性的可能性变大了，特别是对于真正的分布式系统。

随着面向对象程序设计的逐渐成熟，人们的兴趣转移到了面向对象设计方法和面向对象分析或说明。可重用性和可扩展性的好处能用于设计、说明和编码。Biggerstaff and Richter（1989）、Prieto-Diaz and Freeman（1987）和Sommerville（1989）认为，在一个更普遍的软件工程背景下，重用的程度越高，好处就越大。在这个领域中存在一些重要的问题，例如，面向对象设计是否必须用面向对象语言实现，现在的设计方法是否与特定的语言有关。

新的数据库和CASE

到90年代早期，当关系数据库开始受到人们重视的时侯，我们看到，主要的供应商开始引入各种各样扩展了的关系数据库产品，这些产品来源于专家系统、函数程序设计和后来的面向对象程序设计。现在已经出现了面向对象和半面向对象数据库商业产品，但面向对象的理论还必须考虑一些典型的数据库问题。例如，如何有效地处理永久对象、缓存和对象版本管理。我们可以把这种发展解释为发生在这一时期的面向对象概念规则化的一个部分。它也引发了一系
5 列与基于对象—标识方法相比的说明性关系查询语言的相对效率问题。具有讽刺意味的是，根据以往的情况，不仅面向对象数据库，而且早期网状和层次数据库系统都使用面向对象属性。最近，人们越来越清楚，面向对象方法的数据库基本分成两种：纯粹的面向对象数据库和混合的对象—关系数据库。实际上，基于这两种数据库的查询语言分别是OQL和SQL3。第5章将详细讨论数据库理论和面向对象数据库及其问题。

计算机辅助软件工程（CASE）在商业系统的开发中变得越来越重要，有的人热衷于它，有的人对它持怀疑态度。现在已经出现了很多面向对象分析和设计的方法，还有一些支持它们的CASE工具。使用它们到底有什么好处呢？第6章到第8章将讨论面向对象分析和设计技术。在第6章和第7章我们也会讨论支持面向对象方法的CASE工具。

分布式系统和WEB

到了90年代，开发新软件的商业压力增大了，计算机的价格变得更低，而功能变得更强大了。这些因素使得这个领域变得更为成熟，产生了在GUI和AI方面的一系列广泛应用。虽然关系数据库曾经扮演和继续扮演着重要的角色，现在的作用仍然很重要，但是，分布式和客户/服务器计算成为可能而且变得重要，特别是随着所谓的三层客户/服务器系统的出现，对象技术成为很多开发过程的基础。新的应用程序和更好的硬件导致主流组织采用面向对象程序设计，而且开始关注面向对象的设计与分析。在这十年中，面向对象数据库也变得成熟了，并开始应用于商业中。因特网的出现和流行最终提供了解决方法，那就是面向对象的解决方法。因为web能提供各种多媒体（文本、图形、声音、视频），从而人们不必再依靠关系数据库来实现那些需要用来存储、检索复杂结构的应用所需要的性能。处理这些多媒体的自然的程序设计方式是面向对象。第一个著名的用于web的语言Java就是完全面向对象的。处理繁忙的web站点的公司，例如Microsoft和IBM，必须使用面向对象数据库，例如运用Versant和ObjectStore来提供复制、版本控制、速度及其需要的适应性。网络计算、瘦客户和代理都要求一个包含理论框架和软件工程的方法。目前对