

TURING

图灵计算机科学丛书

PEARSON
Prentice
Hall

Object-Oriented Modeling and
Design with UML Second Edition

UML

面向对象建模与设计

(第2版)

[美] Michael Blaha 著
James Rumbaugh
车皓阳 杨眉 译



人民邮电出版社
POSTS & TELECOM PRESS

TURING

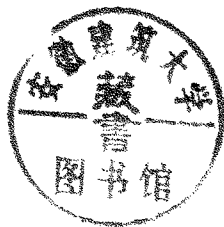
图灵计算机科学丛书

Object-Oriented Modeling and Design with UML
Second Edition

UML 面向对象建模与设计

(第 2 版)

[美] Michael Blaha 著
James Rumbaugh
车皓阳 杨眉 译



 人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目 (CIP) 数据

UML 面向对象建模与设计: 第2版 / (美) 巴拉赫, (美) 兰宝著; 车皓阳, 杨眉译.
—北京: 人民邮电出版社, 2006.1
(图灵计算机科学丛书)

ISBN 7-115-14223-8

I. U... II. ①巴...②兰...③车...④杨... III. 面向对象语言, UML—程序设计—教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 148127 号

内 容 提 要

本书是“面向对象建模与设计”领域的经典著作。全书由四个部分组成。第一个部分以一种高层的、独立于语言的方式描述面向对象的概念, 以及 UML 表示法; 第二部分从问题陈述到分析、系统设计和类设计, 一步一步地描述了软件开发的面向对象方法学; 第三部分用面向对象语言和关系数据库描述了面向对象设计的实现; 第四部分描述了成功的面向对象开发所需要的软件工程实践。本书还配有丰富的习题, 覆盖了一系列应用领域以及实现目标, 而且在书的后面给出了部分习题的答案。

本书可以作为高年级本科生或研究生软件工程或面向对象技术课程的教材, 也可以供相关技术人员参考

图灵计算机科学丛书

UML 面向对象建模与设计 (第 2 版)

- ◆ 著 [美] Michael Blaha James Rumbaugh
译 车皓阳 杨 眉
责任编辑 杨海玲
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 25.5
字数: 621 千字 2006 年 1 月第 1 版
印数: 1—5 000 册 2006 年 1 月北京第 1 次印刷

著作权合同登记号 图字: 01-2005-5234 号

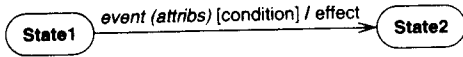
ISBN 7-115-14223-8/TP · 5105

定价: 45.00 元

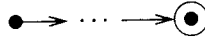
读者服务热线: (010) 88593802 印装质量热线: (010) 67129223

状态模型表示法

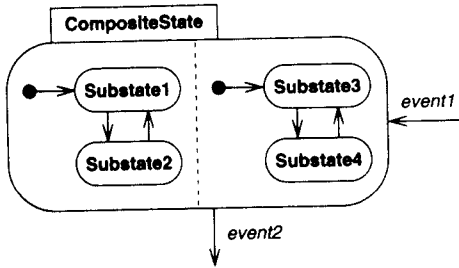
引发状态间迁移的事件:



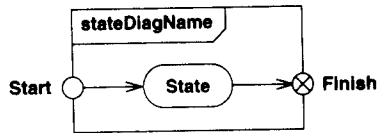
初始状态和终止状态:



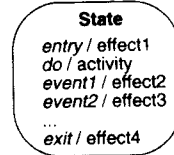
一个对象内的并发:



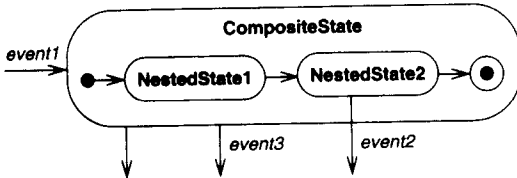
进入点和退出点:



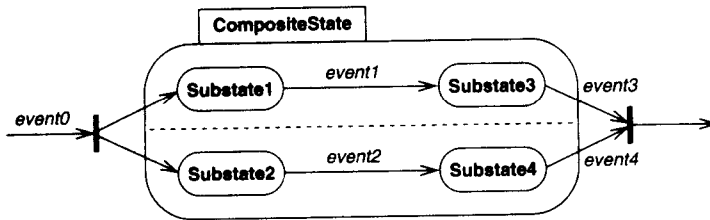
一个状态中的活动:



嵌套状态:



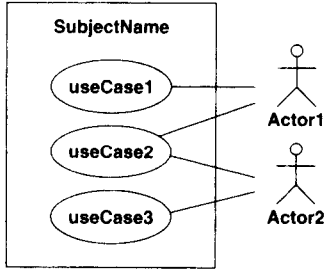
拆分控制:



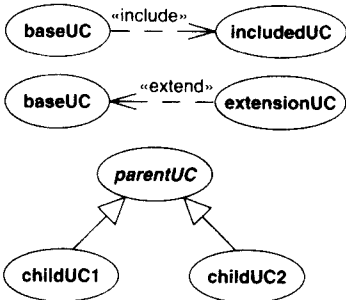
合并控制:

交互模型表示法

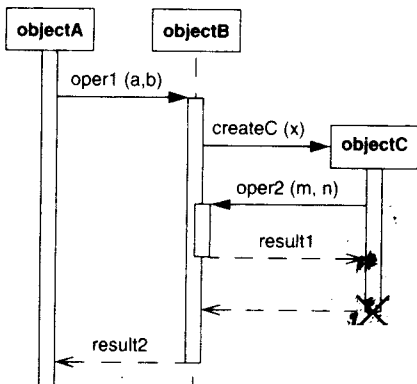
用例图:



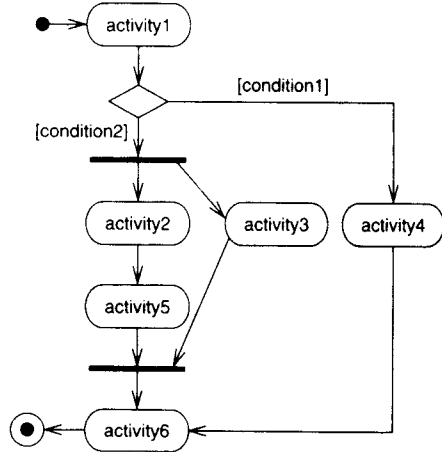
用例关系:



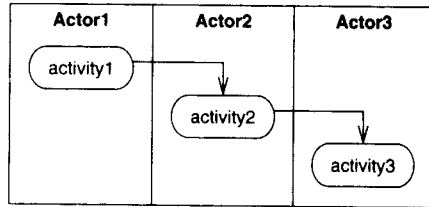
顺序图:



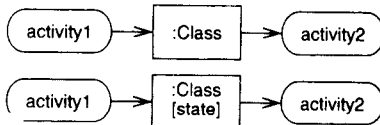
活动图:



带泳道的活动图:



带对象流的活动图:



版权声明

Authorized translation from the English language edition, entitled *Object-Oriented Modeling and Design with UML, Second Edition*, 0130159204 by Michael Blaha and James Rumbaugh, published by Pearson Education, Inc., publishing as Prentice Hall, Copyright © 2005, 1991 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD. and POSTS & TELECOM PRESS Copyright © 2006.

本书中文简体字版由 Pearson Education Asia Ltd. 授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

对本书的赞誉

“由 James Rumbaugh 和 Michael Blaha 及其同事编写的《面向对象建模和设计》一书的初版已经被奉为经典。此书对我的影响远甚于任何其他建模相关的书籍。在过去 10 年中，我曾成功地把他们的思想应用到大型高校课程项目当中，我很高兴看到这本标志性图书最新版本。它肯定会影响新一代软件设计师和建模人员的思维习惯。”

——Bernd Bruegge，慕尼黑技术大学

“Blaha & Rumbaugh 王者归来了。他们为当今时代更新了其经典之作，通过简明直接的讲解，精深的洞察力，以及关键的示例和练习，再次显示了采用面向对象方法其实并不复杂。任何一名实践者都应该拥有、阅读和研究本书。”

——Michael J. Chonoles

“我们信息系统专业的硕士和博士课程采纳了《UML 面向对象建模与设计 (OOMD)》一书中的方法学。这本书由本领域内两名卓越的专家所编写，涵盖了 OOMD 的所有层面，充满真知灼见和最新的示例。它使我们的教学获益良多。”

——Peter H. Chang，劳伦斯技术大学

“如果你正在寻找一本介绍 UML 的图书，其中包括简单实用的面向对象分析和设计过程，以及面向对象概念的重要细节知识，那么我强烈推荐你学习这本出色的著作。”

——Mikael Berndtsson，舍伍德 (Skövde) 大学

前 言

欢迎阅读本书第2版！继本书第1版在1991年写成以后，技术领域已有诸多变化。回想当年，那时，面向对象（OO）技术还相当新颖，初始的兴奋和激情过后，人们对于OO技术究竟是真实可用还是只是一种短暂的时尚，仍怀有疑虑。俱往矣，想一想十多年来发生的变化吧！

- OO语言。C++已成为主流，Java也得到普遍应用。目前具有主导地位的编程语言都是OO的。
- OO数据库。令人惊奇的是，OO数据库已经过气，但现在的关系数据库已经融入一些OO的特性。
- OO建模。对象管理组织（Object Management Group, OMG）制定的统一建模语言（Unified Modeling Language, UML）标准已经合并和统一了各种相互竞争的表达法。
- OO方法论。目前所有开发方法论都毫无例外地融合了OO的思想和概念。

OO技术已经真正成为计算机技术主流的一部分，它不再是什么“异类”，已经成为通常的实践。

本书内容

本书为软件开发提供了一种面向对象的方法，它的基础是，对来自真实世界的对象进行建模，随后使用这些模型围绕对象构造独立于语言的设计。面向对象建模与设计有助于更好地理解需求，进行更清晰的设计，创建更具维护性的系统。我们描述了一组面向对象的概念和一种独立于语言的图形表示法，它们可被用来分析问题需求，并设计问题的解决方案，之后通过编程语言或数据库加以实现。我们介绍的方法能够在整个软件开发过程中运用相同的概念和表示法。软件开发人员不需要在每个开发阶段都转换不同的表示法。

我们阐释了如何在整个软件生命周期（从分析、设计到实现）中使用面向对象的概念。本书并非以讲述面向对象语言或编程为主。相反，我们强调编程是开发过程中的最后阶段，整个过程包括陈述问题，理解需求，规划解决方案，用特定语言实现程序。好的设计技术会将实现细节推延至设计之后的阶段完成，以保持灵活性。开发过程之前的错误会对最终的产品和产品的完成时间产生巨大的影响。我们还描述了如何用面向对象语言和关系数据库来实现面向对象设计。

本书强调的面向对象技术并不只是一种编程方法，更重要的是，它是一种利用真实世界的概念而非计算机概念来抽象思考的方法。我们发现对于有些人来说，这种转换过程是很困难的。那些强调面向对象编程的书籍往往无法帮助程序员学会抽象思维。我们发现图形表示法有助于软件开发者直观想象问题，而不是过早地去寻求实现的方法。

我们还阐释了对于大多数应用而言，无论其最终是用什么语言实现的，面向对象技术提

供的都是一种高效实用的软件开发方法。本书采用了一种非形式化的方法；没有证明或希腊字母表示的形式化定义。借助面向对象技术所捕获的直觉，并且通过提供能够系统应用于实际问题的表示法和方法论，希望形成一种注重实效的解决问题的方法。我们将会同时提供好的和坏的设计技巧和示例，帮助软件开发者避免常见的陷阱。

本书的读者对象

本书面向的读者是软件专业人士和学生。读者可以了解到如何将面向对象的概念应用到软件开发生命周期的全部阶段。我们假定读者事先并不具备任何面向对象概念方面的知识。我们还假定读者熟知基本的计算概念，但不必有太多的形式化背景知识。甚至熟悉面向对象的程序员也会因懂得如何系统地设计程序而大受裨益；他们会吃惊地发现某些常见的面向对象编程实践恰恰违反了良好设计的原则。

数据库设计人员也会对本书感兴趣。尽管面向对象程序设计语言已经受到最大的关注，面向对象数据库设计也还是很引人注目且非常实用的。我们用整整一章的篇幅讲述如何使用关系数据库来实现面向对象模型。

本书可以作为高年级本科生及研究生软件工程或面向对象技术课程的教材。它也可以用作数据库或程序设计语言课程的补充教材。使用本书的前提是掌握现代程序设计语言，了解基本计算机学术语和概念，如语法、语义、递归、集合、过程、图和状态；详细的形式化背景并不是必需的。

我们的关注点与面向对象编程界的一些人有所不同，但与信息建模与设计方法论界一致。我们强调面向对象构造是真实事物的模型，而不单是用于编程的技术。我们把对象之间的关系提高到和类相同的语义层次，而不是将其隐藏为对象内部的指针。在继承和方法上我们关注得较少。对继承机制的精确细节不予过多强调。我们所大力强调的是类型、类、建模和高级规划。我们还演示了如何将面向对象的概念应用于状态机。

本书由四部分组成。第一部分是有一种高层的、独立于语言的方式描述面向对象的概念。尽管一开始会忽略高级主题，但这些概念对于本书的其余部分都是基础。第一部分还介绍了本书中所使用的 UML 表示法。第二部分从问题陈述到分析、系统设计和类设计，一步一步地描述了软件开发的面向对象方法学。在这个方法学中除了最后阶段以外所有阶段都是独立于语言的。第三部分用面向对象语言和关系数据库描述了面向对象设计的实现。尽管并不是想替代面向对象编程的书籍，但它还是描述了可应用于不同环境的一些考虑。第四部分描述了成功的面向对象开发所需要的软件工程实践。

作者曾经接触过许多应用，有着多年面向对象分析、设计、编程和数据库建模的经验。我们热衷于面向对象的方法，发现它几乎适用于各类应用。我们也发现使用面向对象的概念以及图形表示法和开发方法学，可以大大地提高软件的质量、灵活性和可理解性。我们希望本书会有助于传达这些信息。

本书配有丰富的习题，覆盖了一系列应用领域以及实现目标。我们建议读者随着阅读的进度，试着完成其中一些习题。OO 技术是无法通过阅读来学习的，必须通过实践。本书的后面包含了部分习题的答案。

本书特色

市场上有许多关于面向对象技术的图书。本书不同于这些书的地方在于，它讲授的是如何思考面向对象建模，而不仅仅是给出编程语言或建模表示法的技术细节。

许多面向对象方面的书籍常常会从某种语言的角度切入，大多都是讲述编程问题的。在这些书当中有一些也的确讨论了设计问题，但仍然以关于编程为主。很少有书是关注于面向对象分析或设计的。我们阐释了面向对象概念可以也应该在整个软件生命周期内使用。

除此之外，有很多书也描述了 UML 的概念。本书的不同之处在于，它不仅解释了概念，还解释了它们的基本用途，以及该如何用它们来构建软件。我们没有解释每一个概念和个中细微差别，但我们集中精力阐述了 UML 的核心——这足以帮助读者学会如何使用 UML 来构建更好的软件。

自第 1 版以来的变化

自我们完成本书的第 1 版以来，已有 14 年的时间。其间技术上有诸多改进，因此这一版也有许多变化。

- **表示法。**我们把 OMT 表示法替换成了 UML 表示法 (UML 2.0)。UML 现在已占统治地位，是 OO 建模的标准语言。
- **过程。**第 2 版在软件开发过程中增加了更多的内容。现在，我们把领域分析和应用分析区分开来。我们还加入了实现建模。我们特意把过程弄得简单并且“轻量级”，这样更适合学生掌握。本书的过程是“重量级”过程 (如 IBM Rational 的 RUP) 的子集。
- **三个模型。**我们延续第 1 版的重点——“三个模型”，因为我们相信它有助于 OO 建模教学和学习。但是，我们取消了功能模型，因为它并没有像我们期望的那样有用。取而代之，我们增加了交互模型来合并用例和顺序图并给出对对象之间的行为的更完整的认识。
- **软件工程。**第四部分包含了几种重要的软件工程专题：迭代开发、模型的管理以及遗留系统的处理。
- **编程语言。**在过去十年当中，编程语言发生了巨大的变化。Smalltalk 渐渐远去，C 和 Fortran 的重要性也慢慢减弱。现在，C++ 和 Java 是占主导地位的 OO 编程语言，因此，我们也相应把重点放在它们身上。
- **数据库。**OO 模型不仅为程序设计编码提供了一个可靠的基础，也为关系数据库提供了一个可靠的基础。本书用完整的一章描述如何从 UML 模型构建有效的、正确的和可扩展的数据库。
- **案例研究。**在第 1 版出版的时候，我们感觉有必要验证 OO 技术的合理性，因此我们包括了几个案例研究。如今，文献当中可以找到许多案例研究，因此我们在本书中就把它剔除。

在这一版中，我们延续了第 1 版的风格和注重实用的思想，同样给出了许多示例和习题。

网站

我们把与本书相关的资料放在了网站 www.modelsoftcorp.com 上。

致谢

在过去这些年当中，我们曾与许多人一起共事，他们将其所长传授给我们，给我们机会尝试自己的思想，帮助我们在这个过程中学习。我们要感谢这些人，从以前 GE 研发中心管理层直到最近我们业务上的朋友和客户。

第 1 版还有另外三个人参加编写，他们是 Bill Premerlani、Fred Eddy 和 Bill Lorensen，本版却只有我们两个人。感谢他们过去所做的贡献，那是第 2 版的基础。感谢他们在我们编写第 2 版的过程中给予的支持和鼓励。

Chris Kelsey 在第 2 版中担任了重要的角色，应该特别提出。她是第 18 章（OO 编程语言）的主要作者，同时也是积极的审稿人。

感谢其他审稿人（Mikael Berndtsson、Peter Chang、Bill Premerlani 以及 John Putnam）花时间阅读我们的书稿，并给予极赋见地的批评意见。

最后，感谢我们的家人和同事，感谢在本书编写的过程中，他们对诸多不便所显示出的耐心。

Michael Blaha

于密苏里州切斯特菲尔德

blaha@computer.org

James Rumbaugh

于加利福尼亚州库珀蒂诺

目 录

第 1 章 简介	1	3.1.4 值和属性	20
1.1 什么是面向对象?	1	3.1.5 操作和方法	21
1.2 什么是 OO 开发?	3	3.1.6 类表示法小结	22
1.2.1 要建模的是概念, 而非实现	3	3.2 链接和关联的概念	22
1.2.2 OO 方法论	3	3.2.1 链接和关联	22
1.2.3 三种模型	4	3.2.2 多重性	24
1.3 OO 主题	5	3.2.3 关联终端名	25
1.3.1 抽象	5	3.2.4 排序	27
1.3.2 封装	5	3.2.5 包和序列	27
1.3.3 组合数据和行为	5	3.2.6 关联类	27
1.3.4 共享	6	3.2.7 限定关联	29
1.3.5 强调对象的本质	6	3.3 泛化和继承	30
1.3.6 协同 (synergy)	6	3.3.1 定义	30
1.4 关于 OO 开发有效性的证据	7	3.3.2 泛化的用途	32
1.5 OO 建模历史	7	3.3.3 覆盖特征	33
1.6 本书的组织结构	8	3.4 类模型的一个示例	33
参考文献注释	8	3.5 类模型导航	35
参考文献	9	3.5.1 遍历类模型的 OCL 制品	36
习题	9	3.5.2 构建 OCL 表达式	37
		3.5.3 OCL 表达式的示例	37
第一部分 建模的概念		3.6 实践技巧	39
第 2 章 建模是一种设计技术	13	3.7 小结	40
2.1 建模	13	参考文献注释	40
2.2 抽象	13	参考文献	41
2.3 三种模型	14	习题	42
2.3.1 类模型	14	第 4 章 高级类建模	49
2.3.2 状态模型	14	4.1 高级对象和类的概念	49
2.3.3 交互模型	15	4.1.1 枚举	49
2.3.4 模型间的关系	15	4.1.2 多重性	50
2.4 小结	15	4.1.3 作用域	50
参考文献注释	15	4.1.4 可见性	51
习题	16	4.2 关联终端	51
第 3 章 类建模	18	4.3 n 元关联	52
3.1 对象和类的概念	18	4.4 聚合	54
3.1.1 对象	18	4.4.1 聚合与关联	54
3.1.2 类	18	4.4.2 聚合与组合	55
3.1.3 类图	19	4.4.3 操作的传播	55

4.5 抽象类	56	6.1 嵌套状态图	88
4.6 多重继承	57	6.1.1 平面状态图的问题	88
4.6.1 多重继承的种类	57	6.1.2 扩展状态	88
4.6.2 多重分类	58	6.2 嵌套状态	89
4.6.3 应对方案	59	6.3 信号泛化	91
4.7 元数据	61	6.4 并发	92
4.8 具体化	61	6.4.1 聚合并发	92
4.9 约束	62	6.4.2 对象内的并发	93
4.9.1 对象上的约束	62	6.4.3 并发活动的同步	93
4.9.2 泛化集上的约束	62	6.5 状态模型示例	94
4.9.3 链接上的约束	63	6.6 类和状态模型的关系	99
4.9.4 使用约束	63	6.7 实践技巧	99
4.10 派生数据	63	6.8 小结	100
4.11 包	64	参考文献注释	101
4.12 实践技巧	65	参考文献	101
4.13 小结	65	习题	101
参考文献注释	66	第7章 交互建模	105
参考文献	67	7.1 用例模型	105
习题	67	7.1.1 执行者	105
第5章 状态建模	72	7.1.2 用例	106
5.1 事件	72	7.1.3 用例图	107
5.1.1 信号事件	72	7.1.4 用例模型的准则	108
5.1.2 变更事件	73	7.2 顺序模型	109
5.1.3 时间事件	73	7.2.1 场景	109
5.2 状态	74	7.2.2 顺序图	109
5.3 迁移和状态	75	7.2.3 顺序模型的准则	111
5.4 状态图	76	7.3 活动模型	112
5.4.1 状态图示例	76	7.3.1 活动	113
5.4.2 单触发状态图	77	7.3.2 分支	113
5.4.3 状态图的基本表示法小结	78	7.3.3 初始和终止	114
5.5 状态图行为	79	7.3.4 并发活动	114
5.5.1 活动效应	79	7.3.5 可执行活动图	114
5.5.2 do 活动	79	7.3.6 活动模型的准则	115
5.5.3 进入和退出活动	80	7.4 小结	115
5.5.4 完成迁移	81	参考文献注释	115
5.5.5 发送信号	81	参考文献	116
5.5.6 状态图示例	82	习题	116
5.6 实践技巧	82	第8章 高级交互建模	118
5.7 小结	83	8.1 用例关系	118
参考文献注释	83	8.1.1 包含关系	118
参考文献	84	8.1.2 扩展关系	118
习题	84	8.1.3 泛化	119
第6章 高级状态建模	88	8.1.4 用例关系的组合	120

8.1.5 用例关系的准则·····	121	11.4 小结·····	144
8.2 过程化顺序模型·····	121	习题·····	144
8.2.1 带有被动对象的顺序图·····	121	第12章 领域分析·····	146
8.2.2 带有临时对象的顺序图·····	122	12.1 分析概述·····	146
8.2.3 过程化顺序模型的准则·····	123	12.2 领域类模型·····	147
8.3 活动模型的特殊制品·····	123	12.2.1 寻找类·····	148
8.3.1 发送和接收信号·····	123	12.2.2 保留正确的类·····	149
8.3.2 泳道·····	124	12.2.3 准备数据词典·····	150
8.3.3 对象流·····	125	12.2.4 寻找关联·····	150
8.4 小结·····	125	12.2.5 保留正确的关联·····	151
参考文献·····	126	12.2.6 寻找属性·····	155
习题·····	126	12.2.7 保留正确的属性·····	155
第9章 概念小结·····	129	12.2.8 使用继承来细化·····	156
9.1 类模型·····	129	12.2.9 测试访问路径·····	158
9.2 状态模型·····	129	12.2.10 迭代类模型·····	159
9.3 交互模型·····	129	12.2.11 变换抽象的层次·····	161
9.4 模型之间的关系·····	130	12.2.12 把类组织成包·····	162
9.4.1 泛化·····	130	12.3 领域状态模型·····	162
9.4.2 聚合·····	131	12.3.1 确定具有状态的类·····	163
第二部分 分析和设计		12.3.2 寻找状态·····	163
第10章 过程概述·····	135	12.3.3 寻找事件·····	163
10.1 开发阶段·····	135	12.3.4 构建状态图·····	164
10.1.1 系统构思·····	135	12.3.5 评价状态图·····	164
10.1.2 分析·····	136	12.4 领域交互模型·····	165
10.1.3 系统设计·····	136	12.5 将分析迭代·····	165
10.1.4 类设计·····	137	12.5.1 细化分析模型·····	165
10.1.5 实现·····	137	12.5.2 重述需求·····	166
10.1.6 测试·····	137	12.5.3 分析和设计·····	166
10.1.7 培训·····	137	12.6 小结·····	166
10.1.8 部署·····	137	参考文献注释·····	167
10.1.9 维护·····	137	参考文献·····	167
10.2 开发生命周期·····	138	习题·····	167
10.2.1 瀑布式开发·····	138	第13章 应用分析·····	174
10.2.2 迭代开发·····	138	13.1 应用程序交互模型·····	174
10.3 小结·····	138	13.1.1 确定系统边界·····	174
参考文献注释·····	139	13.1.2 寻找参与者·····	175
习题·····	139	13.1.3 寻找用例·····	175
第11章 系统构思·····	140	13.1.4 寻找初始和终止事件·····	176
11.1 形成系统概念·····	140	13.1.5 准备普通场景·····	176
11.2 阐释概念·····	140	13.1.6 增加变化和异常场景·····	177
11.3 准备问题陈述·····	142	13.1.7 寻找外部事件·····	177
		13.1.8 编制复杂用例的活动图·····	180
		13.1.9 组织参与者和用例·····	180

13.1.10 检查领域类模型	180	14.9 选择软件控制策略	204
13.2 应用类模型	181	14.9.1 过程驱动型控制	204
13.2.1 确定用户界面	181	14.9.2 事件驱动型控制	205
13.2.2 定义边界类	182	14.9.3 并发控制	205
13.2.3 确定控制器	182	14.9.4 内部控制	205
13.2.4 检查交互模型	183	14.9.5 其他范型	205
13.3 应用状态模型	183	14.10 处理边界条件	206
13.3.1 使用状态来确定应用类	184	14.11 设定权衡优先级	206
13.3.2 寻找事件	184	14.12 常见的架构风格	207
13.3.3 构建状态图	184	14.12.1 批处理转换	207
13.3.4 检查其他状态图	186	14.12.2 连续型转换	208
13.3.5 检查类模型	187	14.12.3 交互式界面	209
13.3.6 检查交互模型	188	14.12.4 动态仿真	209
13.4 增加操作	188	14.12.5 实时系统	210
13.4.1 来自类模型的操作	188	14.12.6 事务管理器	210
13.4.2 来自用例的操作	188	14.13 ATM 系统的架构	210
13.4.3 购物清单操作	188	14.14 小结	211
13.4.4 简化操作	189	参考文献注释	212
13.5 小结	190	参考文献	213
参考文献注释	190	习题	213
参考文献	191	第 15 章 类设计	218
习题	191	15.1 类设计概述	218
第 14 章 系统设计	194	15.2 填补空白区	219
14.1 系统设计概述	194	15.3 实现用例	220
14.2 估算性能	195	15.4 设计算法	221
14.3 制订复用计划	195	15.4.1 选择算法	221
14.3.1 库	195	15.4.2 选择数据结构	222
14.3.2 框架	196	15.4.3 定义内部类和操作	223
14.3.3 模式	196	15.4.4 把操作分配给类	223
14.4 将系统拆分成子系统	197	15.5 向下递归	225
14.4.1 分层	198	15.5.1 功能分层	225
14.4.2 分区	198	15.5.2 机制分层	225
14.4.3 组合分层和分区	198	15.6 重构	226
14.5 确定并发性	199	15.7 设计优化	226
14.5.1 识别内部的并发性	200	15.7.1 为了高效访问而增加冗余的 关联	227
14.5.2 定义并发任务	200	15.7.2 为了效率而重新调整执行 顺序	228
14.6 分配子系统	200	15.7.3 将派生值保存下来以避免 重新计算	228
14.6.1 估算硬件资源需求	200	15.8 行为具体化	229
14.6.2 权衡硬件和软件	201	15.9 调整继承	230
14.6.3 给处理器分配任务	201	15.9.1 重新调整类和操作	230
14.6.4 确定物理连通性	202		
14.7 管理数据存储	202		
14.8 处理全局资源	203		

15.9.2 提取公共行为	230	18.1.3 比较C++ 和 Java	256
15.9.3 使用委托来共享行为	231	18.2 节选 ATM 模型	257
15.10 组织类设计	233	18.3 实现结构	257
15.10.1 信息隐藏	233	18.3.1 数据类型	258
15.10.2 实体的内聚性	233	18.3.2 类	260
15.10.3 微调包	234	18.3.3 访问控制	260
15.11 ATM 示例	234	18.3.4 泛化	262
15.12 小结	235	18.3.5 关联	265
参考文献注释	236	18.4 实现功能	267
参考文献	236	18.4.1 创建对象	268
习题	237	18.4.2 对象的生存期	270
第 16 章 过程小结	241	18.4.3 对象销毁	271
16.1 系统构思	241	18.4.4 链接创建	272
16.2 分析	242	18.4.5 链接销毁	274
16.2.1 领域分析	242	18.4.6 派生属性	275
16.2.2 应用分析	242	18.5 实践技巧	275
16.3 设计	242	18.6 小结	276
16.3.1 系统设计	242	参考文献注释	277
16.3.2 类设计	242	参考文献	277
第三部分 实现		习题	277
第 17 章 实现建模	245	第 19 章 数据库	281
17.1 实现概述	245	19.1 简介	281
17.2 微调类	245	19.1.1 数据库的概念	281
17.3 微调泛化	247	19.1.2 关系数据库的概念	282
17.4 实现关联	248	19.1.3 范式	283
17.4.1 分析关联遍历	249	19.1.4 选择 DBMS 产品	283
17.4.2 单向关联	249	19.2 节选 ATM 模型	284
17.4.3 双向关联	249	19.3 实现结构——基础	284
17.4.4 高级关联	250	19.3.1 类	284
17.4.5 ATM 示例	251	19.3.2 关联	285
17.5 测试	251	19.3.3 泛化	287
17.5.1 单元测试	251	19.3.4 标识	289
17.5.2 系统测试	251	19.3.5 RDBMS 实现的基本 规则小结	289
17.6 小结	252	19.4 实现结构——高级	290
参考文献注释	252	19.4.1 外键	290
参考文献	253	19.4.2 检查约束	291
习题	253	19.4.3 索引	291
第 18 章 OO 语言	254	19.4.4 视图	292
18.1 简介	254	19.4.5 小结 RDBMS 实现的高 级规则	292
18.1.1 C++ 简介	254	19.5 为 ATM 示例实现结构	292
18.1.2 Java 简介	255	19.6 实现功能	295

19.6.1 将程序设计语言耦合到数据库中	295	22.2 模型的类型	323
19.6.2 数据转换	297	22.3 建模的缺陷	324
19.6.3 封装与查询优化	297	22.4 建模会话	325
19.6.4 使用SQL代码	298	22.4.1 密室建模	325
19.7 面向对象数据库	298	22.4.2 轮转建模	326
19.8 实践技巧	299	22.4.3 实况建模	326
19.9 小结	300	22.5 组织人员	327
参考文献注释	300	22.6 学习技术	328
参考文献	300	22.7 教授技术	329
习题	301	22.8 工具	329
第20章 程序设计风格	305	22.8.1 建模工具	329
20.1 面向对象的风格	305	22.8.2 配置管理工具	330
20.2 可复用性	305	22.8.3 代码生成器	330
20.2.1 可复用性的类别	305	22.8.4 模拟工具	330
20.2.2 可复用性的风格准则	306	22.8.5 库	331
20.2.3 使用继承	307	22.9 估算建模工作量	331
20.3 可扩展性	308	22.10 小结	331
20.4 健壮性	309	参考文献注释	332
20.5 大规模程序设计	310	参考文献	332
20.6 小结	312	第23章 遗留系统	334
参考文献注释	313	23.1 逆向工程	334
参考文献	313	23.1.1 逆向工程与正向工程	334
习题	313	23.1.2 逆向工程的输入	334
		23.1.3 逆向工程的输出结果	335
		23.2 构造类模型	335
第四部分 软件工程		23.2.1 实现复原	334
第21章 迭代开发	317	23.2.2 设计复原	336
21.1 迭代开发概述	317	23.2.3 分析复原	336
21.2 迭代开发与瀑布式开发	317	23.3 构造交互模型	336
21.3 迭代开发与快速原型法	318	23.4 构造状态模型	337
21.4 迭代的适用范围	318	23.5 逆向工程的技巧	337
21.5 执行一次迭代	319	23.6 包装	337
21.6 规划下一次迭代	320	23.7 维护	338
21.7 建模和迭代开发	320	23.8 小结	339
21.8 识别风险	321	参考文献注释	339
21.9 小结	321	参考文献	339
参考文献注释	322	附录A UML图形化表示法	341
参考文献	322	附录B 术语表	343
第22章 管理建模	323	部分习题答案	355
22.1 管理建模概述	323	索引	379