

“十二五”国家重点图书

# 面向对象的分析与设计

---

邵维忠 杨芙清 著

清华大学出版社

“十二五”国家重点图书

# 面向对象的分析与设计

---

邵维忠 杨芙清 著

清华大学出版社  
北京

## 内 容 简 介

本书是一本全面论述面向对象分析与设计方法的专著。全书分为3部分。第一部分为基础篇,论述面向对象方法的基本思想和主要概念,对不同的分析与设计方法进行综述,对统一建模语言(UML)和本书提出的方法做概要介绍。第二部分为分析篇,详细介绍面向对象分析的全过程,围绕面向对象建模中的需求模型(用况图)、基本模型(类图)、辅助模型(包图、顺序图、活动图以及其他各种UML模型图)和模型规约,对面向对象的分析进行深入讨论,给出详细的过程指导和工程策略,并对相关的UML内容做较详细的介绍。第三部分为设计篇,介绍如何在面向对象分析模型基础上,针对具体的实现条件进行面向对象的系统设计,包括问题域部分、人机交互部分、控制驱动部分和数据接口部分4个主要部分的设计,最后介绍如何进行构件化与系统部署。

本书是作者多年来在面向对象技术领域长期从事科研、教学和工程实践的总结,在广泛借鉴该领域的经典理论和当前最新研究成果的基础上,坚持独立思考、实事求是的科学态度,对该领域的诸多问题提出了新的学术观点和解决方案。

本书可供计算机软件领域的研究人员和工程技术人员,计算机软件专业的教师、研究生和高年级本科生阅读参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

面向对象的分析与设计/邵维忠,杨芙清著.--北京:清华大学出版社,2013.1

ISBN 978-7-302-30120-2

I. ①面… II. ①邵… ②杨… III. ①面向对象语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第217816号

责任编辑:焦虹 战晓雷

封面设计:傅瑞学

责任校对:时翠兰

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:25.75

字 数:616千字

版 次:2013年1月第1版

印 次:2013年1月第1次印刷

印 数:1~3000

定 价:44.50元

产品编号:045438-01

# Abstract

This book expounds methods of object-oriented analysis and design, which consists of three parts. Part I is the first four chapters, covering a description of the basic concepts in object orientation, a review on several different analysis and design methods, a brief introduction to the Unified Modeling Language (UML), and an overview of this book. Part II covers the next six chapters, presenting the whole process of object-oriented analysis that will generate a platform independent OOA model. Chapter 5 introduces the requirements analysis so as to establish system requirements models, i. e., use case diagrams. Chapters 6, 7, and 8 describe the development of main system models, i. e., class diagram denoting the objects of the system, along with their characteristics and the relationships between them. Chapter 9 discusses the creation of auxiliary models if necessary, such as the sequence diagram, the activity diagram, the state machine diagram and so on. Chapter 10 introduces the model specification. Part III covers the last five chapters, which discusses the design of object-oriented systems aiming to a specific platform to establish platform specific OOD models. Chapter 11 addresses the design issues related to objects that come from the problem domain. Chapter 12 addresses the design of the user interface. Chapter 13 introduces the identification of control flows in concurrent systems to represent them with active objects. Chapter 14 describes the design of the data management interface to realize the storage and retrieval of objects from data base management systems or file systems. Finally, Chapter 15 introduces the componentization of systems and the deployment of software artifacts.

The readers of this book could be researchers, engineers and technicians in the field of computer software, as well as professors, postgraduates and senior undergraduates majoring in computer software.

**Keywords:** software engineering, object-oriented analysis and design, UML

# 前言

面向对象方法与技术起源于面向对象的编程语言。自 20 世纪 80 年代以来,该方法首先在编程领域取得巨大成功,进而发展到软件生存周期的分析与设计阶段,相继出现了一大批面向对象的分析与设计方法。此后便以不可逆转之势发展为计算机软件领域的主流技术。20 世纪 90 年代出现的统一建模语言(UML)使各种方法在概念和表示法方面走向统一。然而,UML 本身只是一种建模语言而不是一种建模方法,它独立于方法而不能取代各种方法。关于面向对象方法的研究仍在继续。

本书是一部研究和介绍面向对象分析与设计的著作。在此之前,作者曾在 1998 年和 2003 年先后撰写出版了《面向对象的系统分析》和《面向对象的系统设计》两本学术专著<sup>[59][60]</sup>,并分别于 2006 年和 2007 年再版<sup>[62][63]</sup>。本书在一定意义上可看成是上述两部著作的第 3 版,但是将两本书合并为一本,在结构上做了较大的调整,并根据国际上相关理论与技术的最新发展状况及作者的研究心得重新进行了写作。本书在以下几方面继承了其前身的主要特点。

## 1. 对面向对象基本概念的阐述

以往许多关于面向对象方法与技术的著作都没有花费太多的篇幅对面向对象的基本概念进行解释和论述,其出发点是假设读者已经从其他著作中学到了这些基本知识。然而,这种假设并不符合大多数读者的实际情况;甚至有许多已经用面向对象方法开发过一些软件的读者,对面向对象概念的理解也往往不够深入和准确。为此,本书采用了一章的篇幅来论述面向对象方法的基本思想和主要特点,对其基本概念给出详细的解释,进而在使用这些概念的有关章节进行更深入的论述并给出应用实例。事实证明,这对于读者准确、深入理解和掌握各种具体的面向对象技术是十分必要的。

## 2. 充分运用面向对象方法的基本概念,避免概念复杂化

这里所说的基本概念,是指面向对象方法诞生以来经过长期考验被证明必不可少的那些概念,也就是迄今大部分面向对象编程语言能够直接支持的概念。这些概念总量并不很多,却能解决绝大多数面向对象建模问题。然而,以往有不少面向对象分析与设计方法往往为解决某些建模问题而提出许多扩充概念,这些概念明显地增加了方法的复杂性。UML 更是从各种方法中吸收了大量的扩充概念,其总量达到了空前的程度,使其学习者和用户不堪重负。多年来这个饱受学术界批评和产业界抱怨的问题并没有随着 UML 多次修订而得到解决。实际上,只要把面向对象的基本概念理解得更深入些,运用得更充分些,那些采用扩充概念来解决的问题完全可以用基本概念来解决,而且可以使系统模型更为简练和清晰,

与实现后的源程序对应得更好。本书的宗旨是：基于一个精练的面向对象概念集合来建立方法体系，限制扩充概念的引入，使书中提出的方法更容易学习和使用。

### 3. 加强过程指导

保持建模概念简练的一个基本前提是加强过程指导。本书对于如何运用面向对象概念进行系统分析与设计做了详细阐述。对 OOA 和 OOD 的每个基本活动给出具体的建模策略和引导开发者思考的启示。对于一些典型的困难问题提供了解决问题的技巧与经验。特别是，对那些在其他方法中运用扩充概念解决的问题，给出了运用最基本的面向对象概念自然而有效地解决问题的方案。总的目标是提供一套概念简练、讲究实效、过程策略明确、可操作性强的面向对象分析与设计方法。

### 4. 以类图为中心的开放性模型框架

类图是面向对象建模中最重要的模型，因为类图最集中、最完整地体现了面向对象的概念，并为面向对象的编程提供了最直接的依据。本书给出了以类图为中心的开放性 OOA 模型框架和 OOD 模型框架。该框架以类图作为面向对象建模中的基本模型，在各种模型图中处于核心地位，提供最重要的模型信息；以用例图作为需求模型，用于描述需求，为开展面向对象的建模提供良好的基础；以其他各种模型图作为辅助模型，从不同的视角加强对面向对象模型的表示。书中重点推荐了包图、顺序图、活动图、状态机图和构件图等辅助模型，然而模型框架是开放的，开发者可以根据不同领域的具体要求采用其他各种模型图。在各种模型图之外，通过模型规约对各种模型和其中的元素给出详细的定义与解释，避免将大量的细节都堆积到图中，从而既保持模型图清晰和简洁，又能提供完整、准确的模型语义。

### 5. OOA 与 OOD 的关系

书中关于 OOA 和 OOD 的内容构成了一个完整的方法体系。二者采用一致的概念、原则和表示法，但是各有不同的内容、策略和侧重点。OOA 只针对问题域和系统责任，不涉及与具体实现技术有关的问题，目标是建立一个能直接映射问题域、充分体现系统责任、独立于实现的 OOA 模型；OOD 则是在 OOA 模型基础上针对具体的实现条件进行设计，目标是建立一个可在特定平台上实现的 OOD 模型。这种分工观点在对象管理组织(OMG)倡导的模型驱动的体系结构(MDA)中获得了新的生命力。从 MDA 观点来看，OOA 模型是一个平台无关模型，OOD 模型则是一个平台专用模型。有了一个平台无关的 OOA 模型，可以针对多种实现平台分别进行面向对象的设计，从而产生多个平台专用的 OOD 模型。这意味着，OOA 模型可以作为一个整体在 OOD 阶段得到复用。

### 6. 对 UML 的介绍

本书的主要目标是介绍面向对象的分析与设计方法，有关 UML 的内容则是围绕这一主要目标来组织的。书中对 UML 的主要内容进行了详细介绍，对读者在学习时常常感到困惑的概念进行了较为清晰的解释，对 UML 自身存在的一些问题进行了分析和研究，提出了作者的观点。特别是，对于工程实践中如何克服 UML 的复杂内容所带来的困难给出了作者的建议。

本书与 2006 年和 2007 年再版的两部著作<sup>[62][63]</sup>相比,在以下几个方面做出了较大的改进:

- 将以往两本著作的内容合并后,对全书的内容和结构进行了统一布局,避免了某些内容的重复,使布局更为合理,结构更为紧凑。
- 根据国际上面向对象技术和相关标准规范的最新发展以及作者近几年的研究心得,对书中的内容进行了增补和修订。
- 对 UML 的介绍以近期较新而又相对稳定的版本 UML 2.2<sup>[40][41]</sup>为基准,对其中某些问题参照了 UML 2.3 和正在修订中的版本 UML 2.4 进行了核实。
- 对以往两部著作的内容进行了全面的审核、订正和重新写作。减少了若干学术讨论的内容,增加了对一些实际问题的处理策略;删除了关于软件复用的整章内容,增加了构件化和系统部署的内容。在文字表述方面力求“瘦身”,总篇幅缩减至以往两部著作的 70%,使其表达更为准确和精练。

全书共包括 15 章,分为 3 个部分。

第一部分为基础篇。其中第 1 章论述面向对象方法的基本思想,定义和解释其中的基本概念,并介绍其历史与现状。这些基本知识是学习面向对象分析与设计方法的重要基础。第 2 章首先介绍历史上出现的几种经典的分析与设计方法,并与面向对象方法进行比较,然后介绍 4 种影响较大而风格不同的面向对象分析与设计方法。第 3 章对 UML 规范进行简要介绍,包括其背景及发展历史,以及 UML1 与 UML2 先后两个版本系列的基本内容与概况。第 4 章对本书的 OOA 与 OOD 方法进行了提纲挈领式的介绍,是分析篇和设计篇各章内容的总纲。

第二部分为分析篇,详细介绍面向对象分析的全过程。其中第 5 章介绍如何进行需求分析,建立系统的需求模型,即用况图。第 6 章~第 8 章是面向对象系统分析的核心内容,从对象、对象的特征层以及对象之间关系 3 个层次介绍如何建立系统的基本模型,即类图。对如何运用面向对象概念进行类图中每个层次的建模给出详细的过程指导。第 9 章介绍如何根据实际需要有选择地建立系统的各种辅助模型,重点介绍 UML 的包图、顺序图、活动图、状态机图和构件图,对其他各种图只做简要介绍。第 10 章介绍模型规约,讨论如何通过模型规约对系统模型图进行详细而准确的描述,给出一种便于计算机理解(从而提高代码自动生成的比例)并有利于复用的规约组织格式。

第三部分为设计篇,介绍如何将平台无关的 OOA 模型设计为平台专用的 OOD 模型。OOD 模型包括 4 个主要部分。第 11 章介绍问题域部分的设计,根据编程语言、复用支持和硬件性能等实现条件,对来自问题域类进行修改和调整,使之适用于选定的实现平台并满足性能等方面的要求。第 12 章介绍人机交互部分的设计,主要解决人机交互过程的提取、交互命令组织和人机界面的设计问题。第 13 章介绍控制驱动部分的设计,主要解决并发系统和分布式系统中的对象分布和控制流的设计问题。第 14 章介绍数据接口部分的设计,主要解决对象的持久存储问题,针对文件系统、关系数据库管理系统和面向对象数据库管理系统给出不同的设计策略。第 15 章介绍如何将 OOD 模型构件化以及如何进行系统部署。

本书的研究和写作得到国家重点基础研究发展计划(973 计划)课题“领域特定的虚拟计算环境程序设计”(编号 2011CB302604)的支持。高可信软件技术教育部重点实验室和

北京大学信息学院为本书开展的研究工作提供了良好的条件。在写作过程中,我们曾就书中的学术思想与许多校内的同事和校外的同行进行了讨论、交流和印证,从中受益良多。许多博士生和硕士生帮助我们查阅、收集了大量文献资料。北京大学软件研究所建模工具研究小组研制的面向对象建模工具为本书提出的建模方法提供了强有力的支持。在此,我们谨向上述单位和个人致以衷心的感谢!最后,我们恳切希望各位读者对本书可能存在的错误与疏漏给予批评指正。

#### 作者联系信息

联系人:邵维忠

通信地址:北京大学信息科学技术学院; 邮政编码:100871

电话:010-62751790; 传真:010-62751792

电子邮件:wzshao@pku.edu.cn

作 者

2012年10月于北京大学



# 目 录

## 第一部分 基础篇

|                       |    |
|-----------------------|----|
| 第 1 章 面向对象方法概论        | 3  |
| 1.1 什么是面向对象           | 3  |
| 1.2 从认识论看面向对象方法的形成    | 5  |
| 1.2.1 软件开发——对事物的认识和描述 | 5  |
| 1.2.2 语言的鸿沟           | 5  |
| 1.2.3 编程语言的发展使鸿沟变窄    | 5  |
| 1.2.4 软件工程学的作用        | 8  |
| 1.2.5 面向对象是软件方法学的返璞归真 | 12 |
| 1.3 面向对象方法的基本概念       | 12 |
| 1.3.1 对象              | 13 |
| 1.3.2 类               | 14 |
| 1.3.3 封装              | 16 |
| 1.3.4 继承              | 17 |
| 1.3.5 聚合              | 18 |
| 1.3.6 关联              | 20 |
| 1.3.7 消息              | 21 |
| 1.3.8 多态性             | 22 |
| 1.3.9 其他              | 22 |
| 1.3.10 术语对照           | 24 |
| 1.4 面向对象方法的历史及现状      | 26 |
| 1.4.1 历史              | 26 |
| 1.4.2 发展到软件生存周期全过程    | 27 |
| 1.4.3 渗透到计算机软件各个领域    | 30 |
| 第 2 章 不同的分析与设计方法      | 33 |
| 2.1 功能分解法             | 33 |
| 2.2 结构化方法             | 34 |
| 2.3 信息建模法             | 35 |

|              |                           |           |
|--------------|---------------------------|-----------|
| 2.4          | 面向对象方法                    | 36        |
| 2.4.1        | 什么是 OOA                   | 36        |
| 2.4.2        | 什么是 OOD                   | 37        |
| 2.4.3        | 面向对象方法的主要优点               | 38        |
| 2.4.4        | 几种典型的面向对象方法               | 41        |
| 2.4.4.1      | Booch 方法                  | 43        |
| 2.4.4.2      | Coad/Yourdon 方法           | 45        |
| 2.4.4.3      | Jacobson 方法               | 48        |
| 2.4.4.4      | Rumbaugh 方法               | 52        |
| <b>第 3 章</b> | <b>统一建模语言 UML 简介</b>      | <b>56</b> |
| 3.1          | UML 的背景与发展历史              | 56        |
| 3.2          | UML1 概况                   | 58        |
| 3.2.1        | UML1 规范的主要组成部分            | 58        |
| 3.2.2        | 语言体系结构和定义方式               | 59        |
| 3.2.3        | UML1 的各种图和扩展机制            | 62        |
| 3.3          | UML2 概况                   | 66        |
| 3.3.1        | 从 UML1 到 UML2             | 66        |
| 3.3.2        | UML2 的 4 个规范              | 67        |
| 3.3.2.1      | UML 基础结构                  | 67        |
| 3.3.2.2      | UML 上层结构                  | 68        |
| 3.3.2.3      | 对象约束语言                    | 69        |
| 3.3.2.4      | UML 图交换                   | 70        |
| 3.3.3        | UML2 的各种图                 | 70        |
| <b>第 4 章</b> | <b>本书的 OOA&amp;D 方法概貌</b> | <b>74</b> |
| 4.1          | 引言                        | 74        |
| 4.2          | 主要概念                      | 74        |
| 4.2.1        | 主要模型元素                    | 75        |
| 4.2.2        | 面向对象建模的主要原则               | 76        |
| 4.3          | 模型及其规约                    | 80        |
| 4.3.1        | 基本模型——类图                  | 80        |
| 4.3.2        | 需求模型——用况图                 | 81        |
| 4.3.3        | 辅助模型——其他各种图               | 82        |
| 4.3.4        | 模型规约                      | 82        |
| 4.3.5        | OOA 模型框架                  | 83        |
| 4.3.6        | OOD 模型框架                  | 83        |
| 4.4          | 建模过程                      | 85        |
| 4.4.1        | OOA 过程                    | 85        |

|                 |                           |            |
|-----------------|---------------------------|------------|
| 4.4.2           | OOD 过程                    | 87         |
| 4.5             | OOA 与 OOD 的关系             | 88         |
| 4.5.1           | 一致的概念与表示法                 | 88         |
| 4.5.2           | 不同的目标、内容和抽象层次             | 89         |
| 4.5.3           | 在软件生存周期中的位置               | 90         |
| 4.5.4           | 两种不同的分工观点                 | 90         |
| 4.5.5           | 从模型驱动的体系结构看 OOA 与 OOD 的关系 | 92         |
| <b>第二部分 分析篇</b> |                           |            |
| <b>第 5 章</b>    | <b>建立需求模型——用况图</b>        | <b>97</b>  |
| 5.1             | 需求分析和系统分析                 | 97         |
| 5.2             | 基于用况的需求分析基本思路             | 98         |
| 5.3             | 系统边界与参与者                  | 98         |
| 5.3.1           | 系统边界                      | 98         |
| 5.3.2           | 参与者                       | 99         |
| 5.4             | 用况                        | 101        |
| 5.4.1           | 什么是用况                     | 101        |
| 5.4.2           | 用况的内容与格式                  | 103        |
| 5.4.3           | 如何定义用况                    | 104        |
| 5.5             | 用况图                       | 105        |
| 5.5.1           | 模型元素及表示法                  | 105        |
| 5.5.2           | 几个值得商榷的问题                 | 107        |
| 5.6             | 开发过程与建议                   | 110        |
| <b>第 6 章</b>    | <b>发现对象、定义对象类</b>         | <b>113</b> |
| 6.1             | 对象和类的概念及其运用               | 113        |
| 6.1.1           | 概念                        | 113        |
| 6.1.2           | 类的语义                      | 114        |
| 6.1.3           | 在类的抽象层次建模                 | 115        |
| 6.1.4           | 如何运用对象和类的概念               | 117        |
| 6.2             | 表示法                       | 118        |
| 6.3             | 发现对象                      | 118        |
| 6.3.1           | 研究问题域                     | 118        |
| 6.3.2           | 正确地运用抽象原则                 | 119        |
| 6.3.3           | 策略与启发                     | 120        |
| 6.3.4           | 审查与筛选                     | 122        |
| 6.4             | 对象分类                      | 124        |
| 6.4.1           | 将对象抽象为类                   | 124        |
| 6.4.2           | 审查与调整                     | 124        |

|            |                   |            |
|------------|-------------------|------------|
| 6.4.3      | 类的命名              | 125        |
| <b>第7章</b> | <b>定义对象的属性和操作</b> | <b>127</b> |
| 7.1        | 属性和操作             | 127        |
| 7.2        | 表示法               | 128        |
| 7.3        | 定义属性              | 129        |
| 7.3.1      | 策略与启发             | 129        |
| 7.3.2      | 审查与筛选             | 130        |
| 7.3.3      | 推迟到 OOD 考虑的问题     | 131        |
| 7.3.4      | 属性的命名和定位          | 131        |
| 7.4        | 定义操作              | 132        |
| 7.4.1      | 行为分类              | 132        |
| 7.4.2      | 策略与启发             | 133        |
| 7.4.3      | 审查与筛选             | 134        |
| 7.4.4      | 认识对象的主动行为         | 134        |
| 7.4.5      | 描述操作流程            | 134        |
| 7.4.6      | 操作的命名和定位          | 135        |
| 7.5        | 接口的概念及其用途         | 136        |
| <b>第8章</b> | <b>建立对象间的关系</b>   | <b>140</b> |
| 8.1        | 一般-特殊结构           | 140        |
| 8.1.1      | 相关概念              | 140        |
| 8.1.2      | 表示法               | 142        |
| 8.1.3      | 如何发现一般-特殊结构       | 143        |
| 8.1.4      | 审查与筛选             | 145        |
| 8.1.5      | 一般-特殊结构的简化        | 147        |
| 8.1.6      | 多继承及多态性问题         | 150        |
| 8.2        | 整体-部分结构           | 152        |
| 8.2.1      | 相关概念              | 152        |
| 8.2.2      | 表示法               | 153        |
| 8.2.3      | 如何发现整体-部分结构       | 155        |
| 8.2.4      | 审查与筛选             | 156        |
| 8.2.5      | 整体-部分结构的高级应用技巧    | 156        |
| 8.2.5.1    | 简化对象的定义           | 156        |
| 8.2.5.2    | 支持软件复用            | 157        |
| 8.2.5.3    | 表示数量不定的组成部分       | 158        |
| 8.2.5.4    | 表示动态变化的对象特征       | 159        |
| 8.2.6      | 两种结构之间的变通         | 162        |
| 8.3        | 关联                | 163        |

|            |                    |            |
|------------|--------------------|------------|
| 8.3.1      | 相关概念               | 163        |
| 8.3.2      | 表示法及实现技术           | 165        |
| 8.3.2.1    | 基本表示法              | 165        |
| 8.3.2.2    | 二元关联的实现            | 166        |
| 8.3.3      | 运用简单的关联概念解决复杂的关联问题 | 169        |
| 8.3.3.1    | 带有属性和操作的关联         | 169        |
| 8.3.3.2    | 三元关联和 $n$ 元关联      | 171        |
| 8.3.3.3    | 一个类在关联中多次出现        | 173        |
| 8.3.3.4    | 关联端点的复杂情况          | 174        |
| 8.3.4      | 如何建立关联             | 180        |
| 8.4        | 消息                 | 184        |
| 8.4.1      | 什么是消息              | 184        |
| 8.4.1.1    | 顺序系统中的消息           | 184        |
| 8.4.1.2    | 并发系统中的消息           | 185        |
| 8.4.1.3    | 消息对面向对象建模的意义       | 187        |
| 8.4.2      | 消息的表示法             | 187        |
| 8.4.3      | 如何建立消息             | 191        |
| 8.5        | 关于依赖关系             | 193        |
| 8.5.1      | 什么是依赖              | 193        |
| 8.5.2      | 依赖的定义方式和表示法        | 194        |
| 8.5.3      | 依赖关系对面向对象建模的作用     | 197        |
| <b>第9章</b> | <b>建立辅助模型</b>      | <b>199</b> |
| 9.1        | 类图与其他模型图之间的关系      | 199        |
| 9.2        | 包图                 | 201        |
| 9.2.1      | 包的概念、用途及表示法        | 201        |
| 9.2.2      | 包之间的关系及表示法         | 202        |
| 9.2.3      | 如何建立包图             | 205        |
| 9.2.3.1    | 将模型元素打包            | 205        |
| 9.2.3.2    | 包的命名               | 208        |
| 9.2.3.3    | 组织嵌套的包             | 209        |
| 9.2.3.4    | 减少包的嵌套层次           | 210        |
| 9.2.3.5    | 建立包之间的关系           | 210        |
| 9.2.4      | 用包图控制系统的视野         | 212        |
| 9.3        | 顺序图                | 212        |
| 9.3.1      | 顺序图及其用途            | 212        |
| 9.3.2      | 主要概念及表示法           | 213        |
| 9.3.3      | 顺序图的组织机制与复用        | 217        |
| 9.3.4      | 关于顺序图的若干问题讨论       | 219        |

|               |              |            |
|---------------|--------------|------------|
| 9.3.5         | 如何建立顺序图      | 221        |
| 9.4           | 活动图          | 223        |
| 9.4.1         | 活动图及其用途      | 223        |
| 9.4.2         | 主要概念及表示法     | 224        |
| 9.4.3         | 一个活动图的例子     | 228        |
| 9.4.4         | 关于活动图的若干问题讨论 | 230        |
| 9.4.5         | 如何建立活动图      | 232        |
| 9.5           | 状态机图         | 235        |
| 9.5.1         | 状态机图与状态建模    | 235        |
| 9.5.2         | 主要概念及表示法     | 236        |
| 9.5.3         | 一个状态机图的例子    | 241        |
| 9.5.4         | 如何建立状态机图     | 242        |
| 9.6           | 构件图          | 245        |
| 9.6.1         | 构件图的由来及用途    | 245        |
| 9.6.2         | 主要概念及表示法     | 245        |
| 9.6.3         | 如何建立构件图      | 248        |
| 9.7           | UML 的其他几种模型图 | 249        |
| 9.7.1         | 对象图          | 249        |
| 9.7.2         | 组合结构图        | 250        |
| 9.7.3         | 通信图          | 251        |
| 9.7.4         | 交互概览图        | 252        |
| 9.7.5         | 定时图          | 254        |
| 9.7.6         | 部署图          | 257        |
| <b>第 10 章</b> | <b>模型规约</b>  | <b>259</b> |
| 10.1          | 相关概念及问题讨论    | 259        |
| 10.1.1        | 什么是模型规约      | 259        |
| 10.1.2        | 模型规约是给谁看的    | 260        |
| 10.1.3        | 描述语言的选择      | 260        |
| 10.1.4        | 模型规约的组织方式    | 261        |
| 10.1.5        | 关于对象约束语言 OCL | 262        |
| 10.2          | 类图的规约        | 264        |
| 10.2.1        | 目标与措施        | 264        |
| 10.2.2        | 类规约的内容       | 266        |
| 10.2.3        | 类规约的格式       | 268        |
| 10.3          | 其他模型图的规约     | 270        |
| 10.4          | 模型规约的建立过程    | 270        |

## 第三部分 设计篇

|                               |     |
|-------------------------------|-----|
| <b>第 11 章 问题域部分的设计</b> .....  | 275 |
| 11.1 什么是问题域部分 .....           | 275 |
| 11.2 实现条件对问题域部分的影响 .....      | 276 |
| 11.3 设计过程 .....               | 277 |
| 11.3.1 设计准备 .....             | 277 |
| 11.3.2 设计内容及策略 .....          | 277 |
| 11.3.2.1 针对编程语言支持能力的调整 .....  | 277 |
| 11.3.2.2 增加一般类,提供共同协议 .....   | 282 |
| 11.3.2.3 为实现复用采取的设计策略 .....   | 283 |
| 11.3.2.4 提高性能 .....           | 285 |
| 11.3.2.5 为实现对象持久存储所做的修改 ..... | 290 |
| 11.3.2.6 完善对象的细节 .....        | 290 |
| 11.3.2.7 定义对象实例 .....         | 295 |
| 11.3.2.8 修改或补充辅助模型及模型规约 ..... | 296 |
| 11.3.3 建立与 OOA 文档的映射 .....    | 297 |
| <b>第 12 章 人机交互部分的设计</b> ..... | 300 |
| 12.1 什么是人机交互部分 .....          | 300 |
| 12.2 人机交互部分的需求分析 .....        | 300 |
| 12.2.1 分析参与者——与系统交互的人 .....   | 301 |
| 12.2.2 从用况分析人机交互 .....        | 302 |
| 12.2.3 分析处理异常事件的人机交互 .....    | 305 |
| 12.2.4 命令的组织 .....            | 306 |
| 12.2.5 输出信息的组织 .....          | 309 |
| 12.3 人机界面的设计准则 .....          | 310 |
| 12.4 人机界面的面向对象设计 .....        | 312 |
| 12.4.1 选择界面支持系统 .....         | 312 |
| 12.4.2 选用实现交互界面元素 .....       | 314 |
| 12.4.3 用面向对象概念表示界面元素 .....    | 316 |
| 12.5 可视化编程环境下的人机界面设计 .....    | 320 |
| 12.5.1 问题的提出 .....            | 320 |
| 12.5.2 所见即所得的界面开发 .....       | 320 |
| 12.5.3 设计的必要性 .....           | 323 |
| 12.5.4 基于可视化编程环境的设计策略 .....   | 324 |
| <b>第 13 章 控制驱动部分的设计</b> ..... | 330 |
| 13.1 什么是控制驱动部分 .....          | 330 |

|               |                     |            |
|---------------|---------------------|------------|
| 13.2          | 相关技术问题              | 330        |
| 13.2.1        | 系统总体方案              | 330        |
| 13.2.2        | 软件体系结构              | 331        |
| 13.2.3        | 网络环境中的软件体系结构        | 332        |
| 13.2.4        | 系统的并发性              | 334        |
| 13.2.4.1      | 顺序程序和并发程序           | 334        |
| 13.2.4.2      | 进程和线程               | 335        |
| 13.2.4.3      | 应用系统的并发性            | 337        |
| 13.2.4.4      | 术语讨论                | 340        |
| 13.3          | 如何设计控制驱动部分          | 342        |
| 13.3.1        | 选择软件体系结构风格          | 342        |
| 13.3.2        | 确定系统分布方案            | 342        |
| 13.3.3        | 识别控制流               | 347        |
| 13.3.4        | 用主动类表示控制流           | 349        |
| <b>第 14 章</b> | <b>数据接口部分的设计</b>    | <b>351</b> |
| 14.1          | 什么是数据接口部分           | 351        |
| 14.2          | 数据管理系统及其选择          | 352        |
| 14.2.1        | 文件系统                | 352        |
| 14.2.2        | 数据库管理系统             | 353        |
| 14.2.2.1      | 关系数据库和数据库管理系统       | 354        |
| 14.2.2.2      | 面向对象数据库和数据库管理系统     | 356        |
| 14.2.3        | 数据管理系统的选择           | 356        |
| 14.3          | 对象存储方案和数据接口的设计策略    | 358        |
| 14.3.1        | 针对文件系统的设计           | 358        |
| 14.3.1.1      | 对象在内存空间和文件空间的映像     | 358        |
| 14.3.1.2      | 对象存放策略              | 359        |
| 14.3.1.3      | 设计数据接口部分的类          | 361        |
| 14.3.1.4      | 问题域部分的修改            | 362        |
| 14.3.2        | 针对 RDBMS 的设计        | 363        |
| 14.3.2.1      | 对象及其对数据库的使用         | 363        |
| 14.3.2.2      | 对象在数据库中的存放策略        | 366        |
| 14.3.2.3      | 数据接口部分类的设计和问题域部分的修改 | 374        |
| 14.3.3        | 针对 OODBMS 的设计       | 376        |
| <b>第 15 章</b> | <b>构件化与系统部署</b>     | <b>378</b> |
| 15.1          | 面向对象方法与构件技术         | 378        |



|        |                 |     |
|--------|-----------------|-----|
| 15.2   | 面向对象模型的构件化····· | 379 |
| 15.3   | 系统部署·····       | 380 |
| 15.3.1 | 软件制品的组织·····    | 381 |
| 15.3.2 | 系统部署过程与策略·····  | 382 |
| 索引     | ·····           | 385 |
| 参考文献   | ·····           | 389 |