

“十一五”国家重点图书

计算机科学与技术学科前沿丛书

计算机科学与技术学科研究生系列教材（中文版）

面向对象的系统分析

（第2版）

邵维忠 杨芙清 著



清华大学出版社

“十一五”国家重点图书

计算机科学与技术学科前沿丛书

计算机科学与技术学科研究生系列教材（中文版）

TP312

1126

:2

2006

面向对象的系统分析

（第2版）

邵维忠 杨芙清 著

清华大学出版社

北京

内 容 简 介

本书是一本论述面向对象分析方法的专著,其第1版于1998年由清华大学出版社出版,被国内许多大学用作研究生或高年级本科生教材,并被许多软件开发单位作为工程技术用书,至今每年仍在增印。本次再版,根据国内外面向对象领域理论与技术的最新发展,并参照统一建模语言 UML2.0 的最新版本,做了大量的补充、修改和完善工作,同时在概念解释、文字陈述与图解等方面也做了不少改进。

全书共10章,前3章系统地介绍面向对象方法的基本知识,论述面向对象的分析(OOA)的主要特点和优势,概要地介绍本书的 OOA 方法之主要特色。第4~第9章详细介绍 OOA 的全过程,围绕面向对象建模中的基本模型(类图)、需求模型(用况图)、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,对每个开发活动进行深入讨论,给出详细的过程指导和工程策略。第10章集中讨论软件复用及其与 OOA 的关系,给出按软件复用的要求组织 OOA 过程的策略。

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 13501256678 13801310933

图书在版编目(CIP)数据

面向对象的系统分析/邵维忠,杨芙清著. —2版. —北京:清华大学出版社,2006.12

(计算机科学与技术学科研究生系列教材)

ISBN 7-302-13677-7

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

中国版本图书馆 CIP 数据核字(2006)第 099483 号

责任编辑:焦虹 徐跃进

责任校对:时翠兰

责任印制:孟凡玉

出版发行:清华大学出版社

<http://www.tup.com.cn>

c-service@tup.tsinghua.edu.cn

社总机:010-62770175

投稿咨询:010-62772015

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮购热线:010-62786544

客户服务:010-62776969

印 刷 者:北京嘉实印刷有限公司

装 订 者:三河市漂源装订厂

经 销:全国新华书店

开 本:185×260 印 张:20.5 字 数:504 千字

版 次:2006 年 12 月第 1 版 印 次:2006 年 12 月第 1 次印刷

书 号:ISBN 7-302-13677-7/TP·8253

印 数:1~4000

定 价:33.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。
联系电话:010-62770177 转 3103 产品编号:022392-01

“十一五”国家重点图书 计算机科学与技术学科前沿丛书

计算机科学与技术学科研究生系列教材

编
委
会

■ 名誉主任：陈火旺

■ 主 任：王志英

■ 副 主 任：钱德沛 周立柱

■ 编委委员：(按姓氏笔画为序)

马殿富 李晓明 李仲麟 吴朝晖

何炎祥 陈道蓄 周兴社 钱乐秋

蒋宗礼 廖明宏

■ 责任编辑：马瑛珺

本书责任编委：钱乐秋

序

未来的社会是信息化的社会,计算机科学与技术在其中占据了最重要的地位,这对高素质创新型计算机人才的培养提出了迫切的要求。计算机科学与技术已经成为一门基础技术学科,理论性和技术性都很强。与传统的数学、物理和化学等基础学科相比,该学科的教育工作者既要培养学科理论研究和基本系统的开发人才,还要培养应用系统开发人才,甚至是应用人才。从层次上来讲,则需要培养系统的设计、实现、使用与维护等各个层次的人才。这就要求我国的计算机教育按照定位的需要,从知识、能力、素质三个方面进行人才培养。

硕士研究生的教育须突出“研究”,要加强理论基础的教育和科研能力的训练,使学生能够站在一定的高度去分析研究问题、解决问题。硕士研究生要通过课程的学习,进一步提高理论水平,为今后的研究和发展打下坚实的基础;通过相应的研究及学位论文撰写工作来接受全面的科研训练,了解科学研究的艰辛和科研工作者的奉献精神,培养良好的科研作风,锻炼攻关能力,养成协作精神。

高素质创新型计算机人才应具有较强的实践能力,教学与科研相结合是培养实践能力的有效途径。高水平人才的培养是通过被培养者的高水平学术成果来反映的,而高水平的学术成果主要来源于大量高水平的科研。高水平的科研还为教学活动提供了最先进的高新技术平台和创造性的工作环境,使学生得以接触最先进的计算机理论、技术和环境。高水平的科研也为高水平人才的素质教育提供了良好的物质基础。

为提高高等院校的教学质量,教育部最近实施了精品课程建设工程。由于教材是提高教学质量的关键,必须加快教材建设的步伐。为适应学科的快速发展和培养方案的需要,要采取多种措施鼓励从事前沿研究的学者参与教材的编写和更新,在教材中反映学科前沿的研究成果与发展趋势,以高水平的科研促进教材建设。同时应适当引进国外先进的原版教材,确保所有教学环节充分反映计算机学科与产业的前沿研究水平,并与未来的发展趋势相协调。

中国计算机学会教育专业委员会在清华大学出版社的大力支持下,进行了计算机科学与技术学科硕士研究生培养的系统研究。在此基础上组织来自多所全国重点大学的计算机专家和教授们编写和出版了本系列教材。作者们以自己多年来丰富的教学和科研经验为基础,认真研究和结合我国计算机科学与技术学科硕士研究生教育的特点,力图使本系列教材对我国计算机科学与技术学科硕士研究生的教学方法和教学内容的改革起引导作用。本系列教材的系统性和理论性强,学术水平高,反映科技新发展,具有合适的深度和广度。同时本系列教材两种语种(中文、英文)并存,三种版权(本版、外版、合作出版)形式并存,这在系列教材的出版上走出了一条新路。

相信本系列教材的出版,能够对提高我国计算机硕士研究生教材的整体水平,进而对我国大学的计算机科学与技术硕士研究生教育以及培养高素质创新型计算机人才产生积极的促进作用。

陈火旺

再 版 前 言

面向对象方法与技术起源于面向对象的编程语言(OOPL)。20 世纪 80 年代大批 OOPL 的出现标志着 OO 技术开始走向繁荣和实用。但是正如《软件工程百科全书》^[22] 中所言:“编程并不是软件开发问题的主要根源。需求分析与设计问题更为普遍并且更值得解决。因此面向对象开发技术的焦点不应该只对准编程阶段,而应更全面地对准软件工程的其它阶段。面向对象方法真正远大的目标是它适合于解决分析与设计期间的复杂性并实现分析与设计的复用。”基于这一事实,人们对面面向对象方法的研究重点,从面向对象的编程,转移到面向对象的分析与设计(OOA 与 OOD)。从 20 世纪 80 年代后期到 90 年代相继出现了一大批关于 OOA 与 OOD 的学术论文和专著,提出了多种 OOA 与 OOD 方法,并在工业界被广泛采用,这标志着面向对象方法已经无可置疑地成为计算机软件领域的主流技术。最近十年来,统一建模语言(UML)的出现和不断改进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 OO 方法的研究仍在继续。

本书是一本介绍面向对象分析方法的著作,其第 1 版于 1998 年由清华大学出版社出版,此后每年都在增印,至今已第 9 次印刷,被国内许多大学用作研究生或高年级本科生教材,并被许多软件开发单位作为工程技术用书。在一个技术快速发展的领域,一本书能够在七年多的时间不断地拥有新读者,使我们由衷地感谢读者的厚爱,同时也为书中的内容已经难以反映该领域理论与技术的最新发展而心存遗憾。为此我们在第 1 版基础上,根据国际上在该领域的发展状况,结合自己的研究心得,进行了第 2 版的写作。从第 1 版到本次再版,本书有以下几个主要特点:

1. 对 OO 基本概念的阐述

以往许多关于面向对象分析与设计的著作和面向对象编程语言的著作都没有花费太多的篇幅对 OO 基本概念进行解释和论述,其出发点是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 OO 方法开发过一些软件的读者,对 OO 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 OO 方法的基本思想和主要特点,定义其中的主要概念并给出详细的解释,并且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,这对于形成完整的 OO 理论体系,为读者熟练、深入地掌握各种具体的 OO 技术奠定坚实的理论基础是十分必要的。

2. 充分运用 OO 方法基本概念,限制扩充概念的引入

这里所说的基本概念,是指 OO 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 OO 建模问题。然而,以往有不少 OOA 与 OOD 方法往往为解决某些建模问题而提出许多扩充概念,这些概念在增强方法表达能力的同时也明显地增加了方法的复杂性。统一建模语言(UML)更是从各种方法中吸收了大量的扩充概念,所提供的概念总量十分巨大,使 UML 的许多学习者和用户感到不堪重负。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,反而有继续膨胀之势(目前最新的 UML2.0 提供的建模概念已达到 280 多个)。实际上,只要把 OO 基本概念理解得更深入些,运用得更充分些,其中许多扩充概念所能解决的问题也完全可以用基本概念来解决,而且可达到更好的效果——使 OOA 和 OOD 模型更加清晰,并且与实现后的 OO 源程序形成良好的对应。本书的宗旨是:基于一个精练的 OO 基本概念集合建立 OOA 和 OOD 方法体系,限制扩充概念的引入,使书中提供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。所谓加强,并非强制地要求在工程中遵循一种固定的过程步骤。相反,本书在提出一个建议性的过程模型的同时,强调指出 OOA 过程中的各个活动没有严格的次序,可以灵活地安排各项活动的先后次序,并且可以交叉地进行各项活动。书中对 OOA 过程中的每个活动给出详细地指导。针对每一项活动,对于如何运用 OO 概念来分析和建立系统模型进行了详细阐述。对于常见的问题,给出一般的分析策略和引导分析员思考的启示;对于异常的或较为隐含的问题做了进一步的探讨,提供解决这些问题的技巧与经验。特别是,对于许多在其他方法中运用扩充概念解决的问题,给出了运用 OO 基本概念自然而有效地解决问题的方案。总的目标是提供一种基于精练的 OO 概念集合、讲究实效、过程策略明确、可操作性强的 OOA 方法。

过程指导方面的另一个重要特点是加强对软件复用的引导和支持。在介绍每个 OOA 活动的章节中,一方面强调对以往的 OOA 结果的复用,另一方面给出相应的策略,使分析员定义的各种模型成分更容易在其他系统中复用。此外,在最后一章中论述了 OOA 与软件复用的关系,给出按复用的要求组织 OOA 过程的策略。

4. 以类图为中心的开放性模型框架

在面向对象的建模中类图是最重要的模型,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。除类图之外,在 OO 方法的发展历程中,人们还创造了若干其他形式的模型图,以便从不同的视角对系统建模型。目前 UML 提供的模型图已达 13 种。这为软件开发者提出了一个问题:面对种类如此繁多的模型图,如果每一种图都要使用,则将严重地增加工程开销;如果有选择地采用其中几种,应当如何选择?如何处理各种模型之间的关系?

本书给出一个以类图为中心的开放性模型框架,其中以类图作为面向对象建模中的基本模型,在各种模型中处于核心地位,提供最重要的模型信息;以用况图作为需求模型,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,必要时用来加强对 OO 模型表示。书中重点推荐了包图、顺序图和活动图等辅助模型,然而模型框架是开放的,根据不同领域和开发者的具体要求,其他各种模型图也可以纳入这个框架。此外,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

本书与作者的另外一本专著《面向对象的系统设计》^[50]互为姊妹篇。二者构成一个完整的面向对象分析与设计方法体系。OOA 与 OOD 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,暂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,目标是建立一个满足用户需求、针对具体的技术平台、能够编程实现的 OOD 模型。OOA 与 OOD 的这种分工观点来自 Coad/Yourdon 方法^{[9][10]},在本书第 1 版已对其合理性进行详细地论述。这种观点在当前 OMG 倡导的模型驱动的体系结构(MDA)^[33]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(PIM),它独立于任何实现平台;OOD 模型则是一个平台专用模型(PSM)。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。

在面向对象方法中,分析与设计之间不像结构化方法那样强调严格的阶段划分,但是二者毕竟属于不同的软件生命周期阶段。哪些工作应该在 OOA 中进行,哪些工作应该在 OOD 中进行,哪些工作既可以在 OOA 也可以在 OOD 进行?本书和文献[50]对这些问题做出了回答。

与第 1 版相比,本次再版有以下几点主要改进:

- 对面向对象概念的定义、解释和论述比以往更为准确和深入;对一些重要的建模思想和原则进行了更充分的论述。
- 对书中所有的概念,均采用与 UML 一致的术语和表示法。术语的中文译法依据我国颁布的有关标准。
- 对 UML2.0 的各种模型图都进行了介绍,论述了各种模型图在面向对象建模中的作用,对其中比较重要的模型图的应用给出指导。
- 对 OOA 模型框架做了适当的调整,使之更具有开放性,同时对 OOA 过程模型做了相应的调整。
- 在介绍分析方法中各种策略和技巧的同时,列举了更多的应用实例。对于用 OO 基本概念解决其他方法和 UML 采用扩充概念来解决的问题,给出更多的例子。

全书共分 10 章,内容安排如下:

第 1 章集中地介绍和阐述面向对象方法的基本思想,论述它对软件工程理论与技术进步的重要意义,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 OOA 与 OOD 方法以及其他各种面向对象技术的重要基础。

第 2 章剖析了在软件工程领域系统分析所面临的主要困难,通过对各种分析方法的比

较,阐明面向对象分析方法的主要优点。

第3章概括地介绍了本书给出的 OOA 方法的基本面貌和主要特点。

第4章至第6章详细地介绍 OOA 基本模型(类图)的开发,是全书的重点。每一章针对类图的一个层次(即对象层、特征层和关系层)展开讨论,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7章详细介绍用况的概念和需求模型(用况图)的开发,论述它在面向对象的建模中的作用。

第8章全面介绍 UML 的其他 11 种模型图,论述各种图在面向对象建模中的作用。对包图、顺序图、活动图、状态机图和构件图等 5 种较为有用的图进行了详细介绍,并给出如何用这些图进行建模的策略。对部署图、组合结构图、交互概览图、定时图、对象图、通信图等其余 6 种图做了简要介绍。对各种图中一些值得商榷的问题进行了讨论。

第9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、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10章介绍软件复用的基本知识,讨论软件复用与 OOA 的关系,给出按软件复用的要求组织 OOA 过程的策略。

本书的研究与写作得到国家自然科学基金项目“元模型理论与建模方法研究”(60473064)和国家重点基础研究发展计划(973 计划)课题“虚拟计算环境的程序设计方法学”(2005CB321805)的支持。在写作过程中,我们曾就书中的许多学术思想与南京大学徐家福教授,北京航空航天大学金茂忠教授,北京大学王立福、梅宏、麻志毅、谢冰、王千祥、张世琨等同事,以及其他许多专家和应用领域的朋友进行了讨论和印证,从中受益良多。特别是上海复旦大学钱乐秋教授在百忙中审阅了本书的全稿并提出了宝贵意见。北京大学软件研究所的 OO 建模工具课题组所研制的建模工具集 JBOO4.0 为本书和文献[50]中的 OOA 与 OOD 方法提供了技术支持。我们的许多博士生和硕士生在研究工作和课程学习中使用了本书的内印本,将他们发现的问题及时向我们反映。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

2006 年 7 月于北京大学

目 录

第 1 章 面向对象方法概论	1
1.1 什么是面向对象	1
1.2 从认识论看面向对象方法的形成	3
1.2.1 软件开发——对事物的认识和描述	3
1.2.2 语言的鸿沟	4
1.2.3 编程语言的发展使鸿沟变小	4
1.2.4 软件工程学的作用	6
1.2.5 面向对象是软件方法学的返璞归真	11
1.3 面向对象方法的基本概念	12
1.3.1 对象	12
1.3.2 类	13
1.3.3 封装	15
1.3.4 继承	17
1.3.5 聚合	18
1.3.6 关联	19
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 发展到软件生命周期的前期阶段	27
1.4.3 当前的研究及实践领域	30
第 2 章 为什么需要 OOA	32
2.1 什么是 OOA	32
2.2 分析面临的主要问题	32
2.3 分析方法综述	36
2.3.1 功能分解法	36
2.3.2 数据流法	37

2.3.3	信息建模法	39
2.3.4	面向对象方法	40
2.3.5	分析方法的比较	40
2.3.6	OOA 的主要优点	42
2.4	各种 OOA 方法	43
2.5	关于统一建模语言 UML	45
第3章	本书的 OOA 方法概貌	47
3.1	引言	47
3.2	主要概念及表示法	48
3.2.1	建模元素及表示法	48
3.2.2	OOA 的主要原则	50
3.3	OOA 模型及其规约	55
3.3.1	基本模型——类图	55
3.3.2	需求模型——用况图	56
3.3.3	辅助模型——包图、顺序图、活动图及其他	56
3.3.4	模型规约	58
3.3.5	完整的 OOA 文档及详简程度的控制	58
3.4	OOA 过程	60
3.5	OOA 与 OOD 的关系	62
第4章	发现对象、定义对象类	64
4.1	对象和类	64
4.1.1	对象、主动对象以及它们的类	64
4.1.2	类的语义	66
4.1.3	在类的抽象层次上建模	67
4.1.4	如何运用对象和类的概念	69
4.2	表示法	70
4.3	研究问题域和用户需求	71
4.3.1	研究用户需求,明确系统责任	71
4.3.2	研究问题域	72
4.3.3	确定系统边界	73
4.4	发现对象	74
4.4.1	出发点——问题域和系统责任	74
4.4.2	正确地运用抽象原则	74
4.4.3	策略与启发	75
4.4.4	审查和筛选	78
4.4.5	识别主动对象	80
4.5	对象分类	81

4.5.1 将对象抽象为类	81
4.5.2 检查和调整	81
4.5.3 类的命名	82
4.6 建立类图的对象层	83
第5章 定义属性与操作	84
5.1 对象的属性和操作	84
5.2 表示法	86
5.3 定义属性	86
5.3.1 策略与启发	86
5.3.2 审查与筛选	88
5.3.3 推迟到 OOD 考虑的问题	89
5.3.4 属性的命名和定位	90
5.4 定义操作	90
5.4.1 行为分类	90
5.4.2 策略与启发	91
5.4.3 审查与调整	92
5.4.4 认识对象的主动行为	93
5.4.5 操作过程描述	94
5.4.6 操作的命名和定位	95
5.5 建立类图的特征层	96
5.6 接口的概念及其用途	97
第6章 定义对象间的关系	101
6.1 一般-特殊结构	102
6.1.1 相关概念	102
6.1.2 表示法	104
6.1.3 如何发现一般-特殊结构	105
6.1.4 审查与调整	108
6.1.5 一般-特殊结构的简化	110
6.1.6 多继承及多态性问题	113
6.1.7 调整对象层和特征层	115
6.2 整体-部分结构	116
6.2.1 相关概念	116
6.2.2 表示法	119
6.2.3 如何发现整体-部分结构	120
6.2.4 审查与筛选	121
6.2.5 整体-部分结构的高级应用技巧	122
6.2.5.1 简化对象的定义	122

6.2.5.2	支持软件复用	123
6.2.5.3	表示数量不定的组成部分	124
6.2.5.4	表示动态变化的对象特征	124
6.2.5.5	两种结构之间的变通	127
6.2.6	调整对象层和属性层	129
6.3	关联	129
6.3.1	相关概念	129
6.3.2	表示法及实现技术	131
6.3.2.1	基本表示法	131
6.3.2.2	二元关联的实现	132
6.3.3	运用简单的关联概念解决复杂的关联问题	135
6.3.3.1	带有属性和操作的关联	135
6.3.3.2	三元关联和 n 元关联	138
6.3.3.3	一个类在关联中多次出现	141
6.3.3.4	关联端点的复杂情况	142
6.3.4	如何建立关联	147
6.3.5	对象层、特征层的增补及关联说明	151
6.4	消息	152
6.4.1	什么是消息	152
6.4.1.1	顺序系统中的消息	152
6.4.1.2	并发系统中的消息	153
6.4.1.3	消息对 OOA 的意义	157
6.4.2	消息的表示法	157
6.4.3	如何建立消息	163
6.4.4	消息的规约	166
6.5	关于依赖关系的讨论	166
6.5.1	什么是依赖	166
6.5.2	依赖的定义和表示法中的若干问题	169
6.5.3	依赖对面向对象建模的作用	172
第7章	建立需求模型	175
7.1	需求分析和系统分析	175
7.2	基于用况的需求分析基本思路	176
7.3	系统边界与参与者	176
7.3.1	什么是系统边界	177
7.3.2	什么是参与者	178
7.3.3	系统边界与参与者对 OOA 的意义	180
7.3.4	如何发现参与者	180
7.4	用况	182

7.4.1	什么是用况	182
7.4.2	用况对 OOA 的意义	184
7.4.3	表示形式和例子	185
7.4.4	如何定义用况	188
7.5	用况图	189
7.5.1	用况图中的主要模型元素及表示法	189
7.5.2	关于用况图的若干问题讨论	190
7.6	开发过程和文档组织	193
7.6.1	开发过程小结	193
7.6.2	文档组织	194
7.6.3	几条建议	195
第 8 章	建立辅助模型	197
8.1	类图和其他模型图之间的关系	197
8.2	包图	200
8.2.1	包的概念、用途及表示法	200
8.2.2	包之间的关系及表示法	201
8.2.3	如何建立包图	204
8.2.3.1	将模型元素打包	204
8.2.3.2	包的命名	207
8.2.3.3	组织嵌套的包	208
8.2.3.4	减少包的嵌套层次	209
8.2.3.5	建立包之间的关系	209
8.2.4	用包图控制对系统模型的视野	211
8.3	顺序图	212
8.3.1	顺序图及其用途	212
8.3.2	主要概念及表示法	213
8.3.3	顺序图的组织机制与复用	217
8.3.4	关于顺序图的若干问题讨论	219
8.3.5	如何建立顺序图	222
8.4	活动图	225
8.4.1	活动图及其用途	225
8.4.2	主要概念及表示法	226
8.4.3	一个活动图的例子	230
8.4.4	关于活动图的若干问题讨论	231
8.4.5	如何使用活动图	234
8.5	UML 的其他几种模型图	237
8.5.1	对象图	237
8.5.2	组合结构图	238

8.5.3	通信图	239
8.5.4	交互概览图	241
8.5.5	定时图	243
8.5.6	状态机图	245
8.5.7	构件图	256
8.5.8	部署图	262
第9章	模型规约	264
9.1	相关概念及问题讨论	264
9.1.1	术语问题	264
9.1.2	规约是给谁看的	265
9.1.3	采用形式语言还是自然语言	266
9.1.4	规约的组织方式问题	266
9.1.5	关于对象约束语言 OCL	268
9.2	类的规约	270
9.2.1	目标与措施	270
9.2.2	类规约的内容	272
9.2.3	类规约的组织格式	274
9.3	对其他模型图的规约	276
9.4	模型规约的建立过程	277
第10章	OOA 与软件复用	279
10.1	软件复用	279
10.1.1	概述	279
10.1.2	可复用构件	282
10.1.3	构件库	284
10.1.4	软件复用的根本困难	287
10.2	专向领域的复用及领域分析	290
10.2.1	可复用的软件构架	290
10.2.2	领域分析	291
10.2.3	基于构件/构架的软件开发	292
10.3	OOA 对软件复用的支持	294
10.3.1	支持软件复用的 OO 概念与原则	294
10.3.2	支持软件复用的 OOA 方法	295
10.4	复用技术对 OOA 的支持	297
索引		302
参考文献		305

第 1 章

面向对象方法概论

本章集中介绍面向对象方法的基本原理与概念,论述该方法成为当今计算机软件领域主流技术的原因及其对改进软件开发、促进计算机科学技术发展的重要意义。最后简要介绍其发展历史与现状。本章的主要目的,是使读者在学习面向对象的分析与设计技术之前,先了解面向对象方法的主要内容,掌握基本知识,作为进一步学习的基础。

1.1 什么是面向对象

由于面向对象方法已经发展到计算机科学的许多领域,所以很难从一般意义上给出“面向对象方法”的严格而清晰的定义,使之对这些领域都能适用。如果在 20 世纪 80 年代初期以前讨论这个问题,则可以说:面向对象是一种新兴的程序设计方法,或者说它是一种新的程序设计范型(programming paradigm),其基本思想是使用对象、类、封装、继承、聚合、关联、消息、多态等基本概念来进行程序设计。自 20 世纪 80 年代以来,面向对象方法已深入到计算机软件领域的几乎所有分支,远远超出了程序设计语言和编程技术的范畴。但是,即使在“计算机软件”范围内定义什么是面向对象也仍然是不完整的,因为面向对象方法还发展到计算机软件以外的一些领域,如计算机体系结构和人工智能等。

不过,本书主要是讨论软件开发问题的,而面向对象方法也主要是在计算机软件领域产生了巨大影响,并且发展成较为完整的理论与技术体系,因此本书就在这个范围内讨论什么是面向对象。

面向对象(object-oriented, OO)不仅是一些具体的软件开发技术与策略,而且是一整套关于如何看待软件系统与现实世界的关系,用什么观点来研究问题并进行求解,以及如何进行系统构造的软件方法学。

概括地说,面向对象方法的基本思想包括两个主要方面。一方面,是从现实世界中客观存在的事物(即对象)出发来构造软件系统,并在系统构造中尽可能地运用人类的自然思维方式。开发一个软件是为了解决某些问题,这些问题所涉及的业务范围称为该软件的问题域。面向对象方法强调直接以问题域(现实世界)中的事物为中心来思考问题、认识问题,并根据这些事物的本质特征,把它们抽象地表示为系统中的对象。以对象作为系统的基本构成单位,而不是用一些与现实世界中的事物相差较远,并且没有对应关系的其他概念来构造系统。这可以使系统直接地映射问题域,保持问题域中事物及其相互关系的本来面貌。另一方面,面向对象方法比以往的方法更接近人类的日常思维方式。软件开发方法不应该是