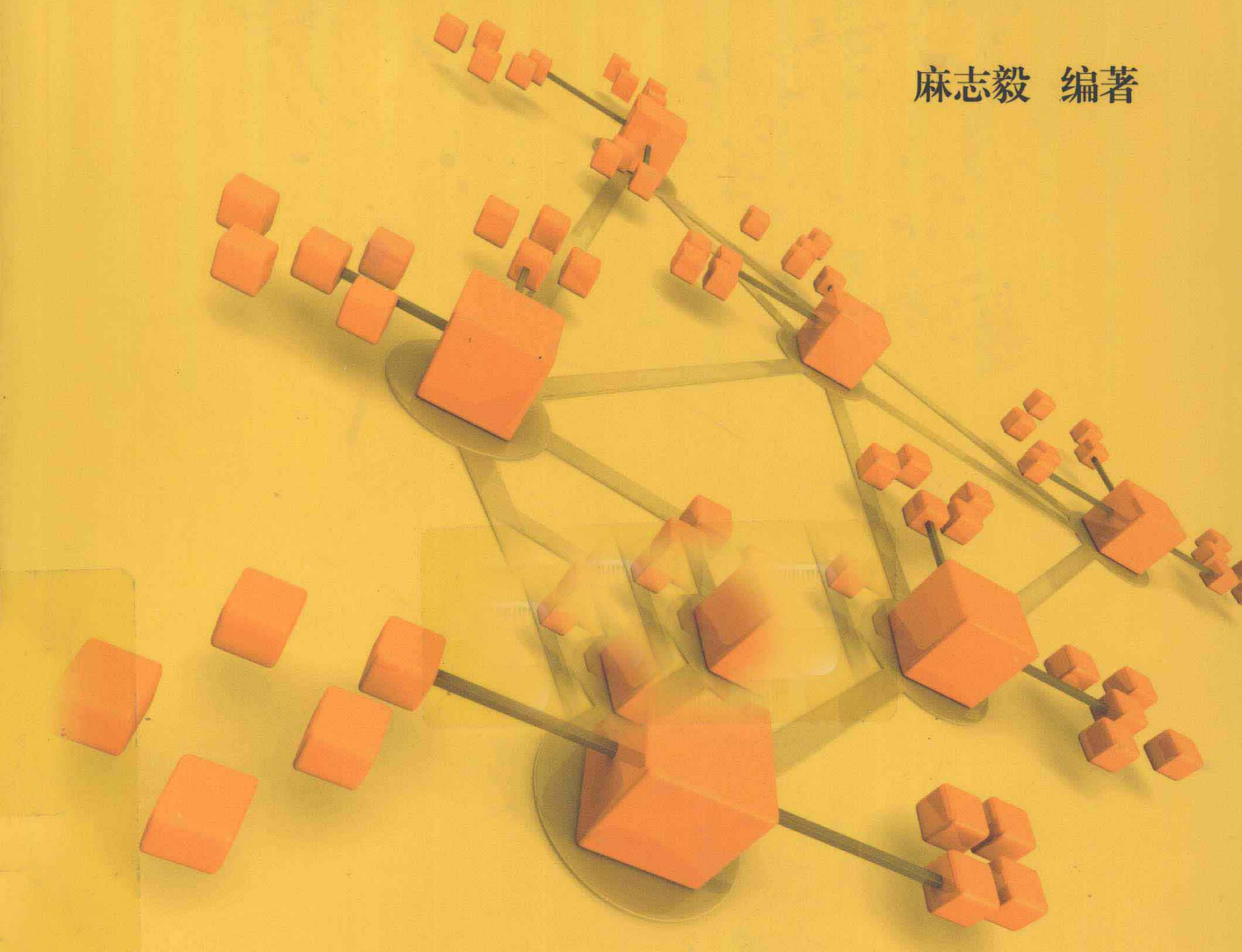
HZ BOOKS
华章教育

面向对象开发方法

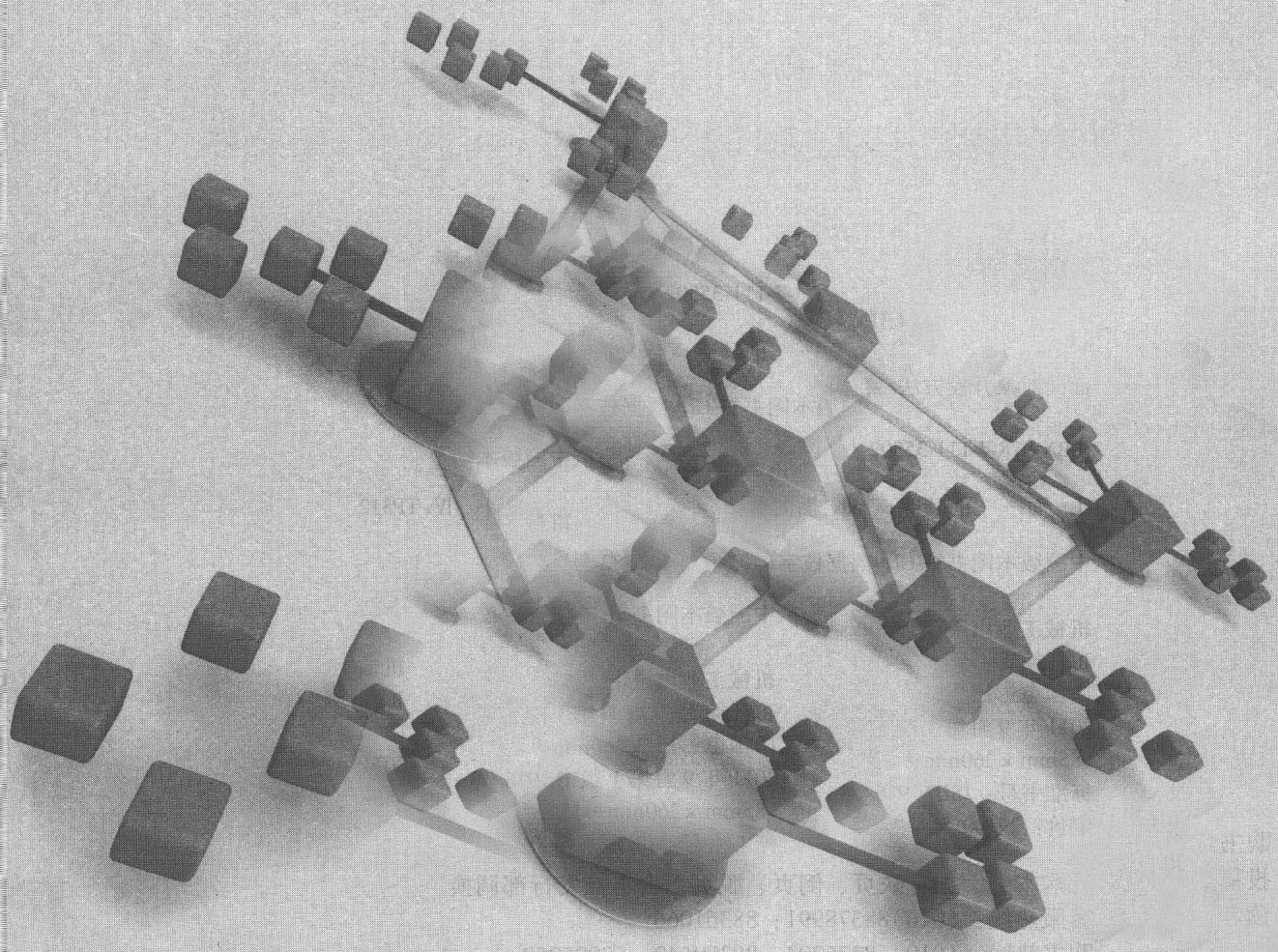
麻志毅 编著



机械工业出版社
China Machine Press

面向对象开发方法

麻志毅 编著



机械工业出版社
China Machine Press

本书是一本关于面向对象方法的教材，讲述了面向对象的基本思想、原则和主要概念，并给出了详细的过程指导。全书主要包括面向对象的分析、设计、实现、测试、系统与模型等。本书注重理论与实践相结合，通过大量的例题、综合案例以及对建模概念的详细剖析，阐明了如何用面向对象方法开发软件系统。

本书的读者对象为高等院校软件学院和计算机学院（或信息学院）软件工程专业以及相关专业的工程硕士、高年级本科生，同时本书也适合从事软件开发的技术人员参考。

图书在版编目（CIP）数据

面向对象开发方法/麻志毅编著. —北京：机械工业出版社，2011.8

ISBN 978-7-111-35502-1

Ⅰ.面…　Ⅱ.麻…　Ⅲ.面向对象语言，UML－程序设计　Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2011）第 151382 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：吴　怡
中国电影出版社印刷厂印刷
2011 年 9 月第 1 版第 1 次印刷
185mm×260mm · 15.75 印张
标准书号：ISBN 978-7-111-35502-1
定价：29.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

前　言

在20世纪80年代，大批面向对象编程语言的出现标志着面向对象技术开始走向繁荣。80年代后期到90年代出现了多种面向对象分析和设计方法，在大多数发达国家的软件开发中得到了相当广泛的运用。在我国的软件产业界，对面向对象技术的学习与应用热潮出现于20世纪90年代的后期，特别是随着统一建模语言(Unified Modeling Language，UML)的出现和不断改进以及面向对象方法的不断完善，使得面向对象技术以其显著的优势成为计算机软件领域的主流技术。

当前，产业界需要大量的掌握面向对象方法与技术的高级应用型开发人才，很多院校开设了相应的课程，旨在使这些人才不仅会使用一种或者几种面向对象编程语言来编程，更重要的是能运用面向对象方法进行系统软件开发，即通过面向对象分析(OOA)和面向对象设计(OOD)建立系统的分析模型和设计模型，进而能够进行编程和测试。

软件开发方法通常要给出概念体系和运用其进行开发软件的过程指导，并要有相应建模工具的支持。

1. 本书概念体系

本书的面向对象概念均遵循UML 2.2。UML是由国际对象管理组织(OMG)发布的建模语言规范，在软件工业界和学术界已经被广泛接受。但UML的内容是庞大和复杂的，多数工程技术人员和读者反映其学习难度很大，这是UML本身的复杂性造成的。UML中的许多内容是用于构造UML元模型的，对于大多数面向应用的软件开发者来说，这些概念一般用不着。还有一些概念在软件系统的建模中很少使用，这是因为UML是各方面成果融合的产物，它尽量地要适合于各领域。特别是UML不仅仅可用于面向对象的软件建模，还可用于对其他方面的建模，例如可用它在建筑业或机器制造业进行建模。

邵维忠教授和杨芙清院士合著的[17]和[18]两本著作，在广泛借鉴国际上各种OOA与OOD方法的同时，根据作者长期的研究与实践形成了自己的方法特色：一是提倡充分运用面向对象方法的基本概念，限制扩充概念的引入，通过加强过程指导来保持建模概念的简练；二是对UML 2.2所采用的与面向对象有关的概念进行了深入的解析，给出了自己的见解；三是其OOD部分更强调用OO概念表达各种全局性的设计决策。

本书选用了 UML 2.2 中常用的面向对象建模概念，以控制概念体系的复杂性。本书遵循了[17]和[18]两本著作提出的宗旨，即加强过程指导来运用基本概念解决各种复杂的分析与设计问题，因此所选用的概念体系仍能保持表达能力的完整性。为了使读者能够理解常常遇到的 UML 概念，尽管这些概念并不是必不可少的，本书也适当地进行了讲解，但是同时也给出了一些运用基本的 OO 概念代替这些概念的方法。除了所采用的概念和表示法与 UML 保持一致外，在中文术语方面，本书与我国的行业规范“面向对象的软件建模规范”[14]完全一致。本书作者作为该规范的主要起草者曾经与国内很多专家、学者和企业界的专业人士进行过反复研究讨论，从而对该规范达成共识。

2. 本书过程指导

过程指导包括开发过程中用到的模型、工具和技术。本书所给出的过程指导，是根据作者单位作为带头承担单位的国家重点科技攻关计划“青鸟工程”的研究成果和国内外的科研及工程实践的经验总结出来的，其中采用了[17]和[18]两本著作中的很多研究成果。像使用其他开发方法一样，用面向对象方法进行软件系统开发的目的是要建立相应的模型，并产生代码。本书讲述了如何建立系统的需求模型、分析模型和设计模型，并针对建立各种模型所使用的图以及其中的一些具体的模型元素，讲述了如何规约。由设计模型产生常见的面向对象编程语言，本书主要通过示例的方式讲述对典型问题的处理技术。此外，基于传统的测试技术，本书讲述了如何针对面向对象软件进行测试。

对于面向对象的软件建模，需要有建模工具的支持。本书从此类工具所应具有的功能上进行了讲述，并介绍了一种具体的面向对象的软件建模工具。

3. 本书特点

在思想体系上，本书在面向对象分析和设计方面秉承了[17]和[18]所提出的宗旨。但这两部学术专著，作为教材适合于理论性强的研究生教学。本书旨在提供一本针对应用型人才培养的教材。本书与[17]和[18]两本著作相比有以下几点不同：

1)增强了对如何运用概念的讲解，减少了理论阐述和对不同学术观点的讨论。

2)着重讲述了面向对象的应用技术，增加了实现和测试以及体系结构建模技术的讲述

3)在各章的正文部分增加了大量例题，在各章之后给出了习题。

4)通过案例，讲述了如何运用面向对象方法进行分析与设计。

由于以上几个特点，本书具有更强的普及性，适合于更广大的读者群。本书既是一本教材，也可作为从事软件开发的工程技术人员的参考书。

4. 本书主要内容

以下简要地介绍本书的概貌，使读者对它有一个提纲挈领的了解。

第 1 章 集中介绍面向对象方法的基本思想和原则，论述它的主要优点，并简介它的

发展历史和现状。最后，对与本书密切相关的 UML 2.2 进行了简介。

第 2 章 首先概述面向对象分析所面临的问题，然后对其进行综述。在综述中阐述了面向对象的分析模型和过程模型。

第 3 章 全面地讲解建立需求模型所需要使用的概念及其表示法，并详述如何使用它们来建立需求模型。

下面的第 4 章与第 5 章详述了如何建立面向对象分析部分的结构模型和行为模型。

第 4 章 详细地讲述了类图中所使用的概念与表示法，并详述了如何使用它们来建立类图。

第 5 章 讲述建立辅助模型所用到的几种图：顺序图、通信图、状态机图、活动图和包图。其中详细地讲述这些图中所用的概念与表示法以及运用它们建立辅助模型的技术。

下面的第 6 章 ~ 第 12 章详述了如何进行面向对象设计。

第 6 章 阐述面向对象的设计模型和过程模型。

第 7 章 详述如何针对实现条件，把分析模型作为问题域进而进行补充与调整。

第 8 章 详述进行人机交互设计所需要考虑的因素，并从分析和设计两个方面详述了如何进行人机交互设计。

第 9 章 详述什么是控制流，如何识别与定义控制流，以及如何协调控制流之间同步。

第 10 章 讲述对数据管理部分的设计。首先对数据库进行简介，然后详细地讲解如何使用关系数据库系统对永久对象及其间的关系进行存储与检索。

第 11 章 首先讲述针对设计模型进一步开发软件所产生的制品以及如何针对典型建立制品图，然后讲述了用于对系统的网络拓扑结构建模的部署图以及制品如何在其上部署的技术。

第 12 章 从耦合、内聚和复用等方面讲述如何评价面向对象的设计模型。

基于所建立的模型，第 13 章和第 14 章分别讲述了如何对所开发的系统进行面向对象的编码实现与测试。

第 13 章 为面向对象的编程实现。该章首先对面向对象的编程语言进行简介，然后用大量的例题，阐述了面向对象的概念在面向对象语言中的体现以及典型的编程技术。该章还对数据管理部分和状态机图的实现进行了具体的详述。

第 14 章 基于传统的测试技术，针对面向对象软件的特点，讲述了测试面向对象软件的典型技术。

第 15 章 首先讲述如何把一个较为复杂的系统划分成一组子系统，然后说明了如何对系统或子系统进行建模，以及从哪些方面建立系统的模型。此外还阐述如何保证模型的一致性。

第 16 章 通过一个具体的案例分析，说明如何用面向对象方法进行建模。对于该章

给出的综合性习题，建议学生在学完第3章后，根据所选择的习题的难易程度组成3～5人小组，随着课程的进展逐步完成。

最后本书给出了一个附录。该附录给出了对用面向对象方法进行软件系统建模所生成的文档的主要编制要求。

作者长期在北京大学计算机系邵维忠教授领导下开展研究工作，本书的研究工作得到了邵教授的大力支持。邵教授具有严谨的治学态度、深厚的学术功底以及敏锐的洞察力，他的指导使我受益良多。在此致以衷心的感谢！

本书的研究工作和写作得到了杨芙清院士所领导的学术队伍的支持，得到了国家重点基础研究发展规划(2011CB302604)的资助。在此谨向上述单位表示衷心的感谢！

对书中存在的错误和疏漏之处，恳请各位读者给予批评指正。

作　者

2011年6月于北京大学

教学建议

章	教学要求	学时
第1章 面向对象方法概论	初步掌握面向对象方法的主要概念和基本思想，领会面向对象的基本原则，理解面向对象方法的主要优点，了解面向对象方法发展的简史和现状	2～4
第2章 什么是面向对象分析	深刻地认识面向对象分析所面临的问题，重点掌握面向对象分析模型和过程模型	2
第3章 建立需求模型——用况图	掌握用况图中所使用的概念与表示法，能熟练地运用它们建立软件系统的需求模型	3～4
第4章 建立基本模型——类图	掌握类图中所使用的概念与表示法，能熟练地运用它们来建立软件系统的基本模型。该章的常用概念较多，要从多个角度进行理解，多进行练习。掌握该章的内容是学习面向对象开发方法的关键	8
第5章 建立辅助模型	掌握顺序图、通信图、状态机图、活动图和包图中的概念与表示法，能熟练地运用它们来建立软件系统的辅助模型。其中的顺序图、状态机图和包图是重点	6～8
第6章 什么是面向对象设计	明确面向对象的分析与设计的关系，掌握面向对象设计模型和过程模型	1～2
第7章 问题域部分的设计	掌握该章所讲述的对典型问题的处理方法	4
第8章 人机交互部分的设计	掌握用面向对象概念表达界面成分以及人机交互部分与问题域部分的衔接机制	2～3
第9章 控制驱动部分的设计	理解什么是控制流，掌握如何设计控制驱动部分。进程间和线程间的通信以及控制流间的同步是其中的难点	2～3
第10章 数据管理部分的设计	重点掌握针对关系数据库的数据存取设计，了解针对面向对象数据库的数据存取设计和针对文件的数据存取设计	3～4
第11章 制品及部署部分的设计	掌握如何用制品以及制品间的关系对软件开发过程中产生的典型物理事物建模，以及如何使用部署图对系统的部署进行设计	1～2 可选
第12章 OOD的评价准则	了解如何从耦合、内聚和复用等方面评价面向对象的设计模型	1～2 可选
第13章 面向对象的编程实现	了解典型的面向对象程序设计语言，重点掌握用具体的面向对象程序设计语言实现设计模型	2～4
第14章 面向对象测试	了解面向对象测试的概念，掌握典型的面向对象测试技术。对该章的学习，应该具有一般的软件测试基础知识	2 可选
第15章 系统与模型	掌握系统、子系统、模型和视图概念，以及相关概念间的关系；重点掌握运用这些概念划分系统和对系统进行建模的技术	2～3 可选
第16章 案例：网上会议文件审批系统	学会用面向对象方法进行系统建模的整体环节以及一些图的具体用法。建议随着课程的进度按相关章的内容分开讲解该章的内容。对于该章给出的综合性习题，建议学生在学完第3章后，根据所选择习题的难易程度组成2～4人小组，随着课程的进展逐步完成习题	4

注："可选"表示若学时不足，可少讲或不讲该章内容。各章都附有习题，教师可以根据情况留一些习题作为课外作业。建议安排数次习题课，重点讲解作业中存在的普遍性问题。针对学生普遍关心的问题，可安排1～2次课堂讨论课。

目 录

CHAPTER 1

第 1 章 面向对象方法概论

计算机软件的地位在近几十年来发生了巨大的变化，它几乎已经深入到了社会生活中各个领域，并发挥着重要的作用。随着计算机硬件技术、网络技术以及社会其他领域的迅速发展，人们对计算机软件的要求也越来越高，这致使开发出来的软件越来越复杂，其质量、成本和工期也越来越难以控制。面对这种状况，人们也越来越深刻地认识到，要用工程化的方法开发计算机软件。

软件工程是应用计算机科学理论和技术加上工程管理原则和方法，按预算和进度实现满足用户要求的软件产品的工程，或以此为研究对象的学科[1]。其主要内容包括软件开发方法和计算机辅助软件工程工具等。

在软件工程的发展历程中，流行过功能分解法、结构化方法和信息建模方法等。这些方法几乎都只注重于系统的某方面，对系统的其他方面建模的能力都很弱。例如，功能分解法集中于将功能作为系统的构造块，对数据组织的功能较弱；即使是在结构化方法中对数据组织的支持也不强；在信息建模方法中的构造块是实体，强调对数据组织，但在该方法中忽略了系统功能。此外，上述方法都没有较强的描述系统的动态行为的能力。这些方法曾发挥着重要的作用，但随着对软件的正确性、易维护性和功效性等的要求越来越高，这些方法显现出了局限性。

软件学术界和产业界尝试了数十年，一直在寻找开发较为复杂的软件系统的有效方法。经过坚持不懈的努力，形成了面向对象方法。面向对象方法是一种把面向对象的思想运用于软件开发过程中，指导开发活动的系统方法。面向对象方法是在传统方法的基础上发展起来的，例如仍使用抽象和模块化等概念。然而，面向对象方法与传统方法相比发生了根本性的变化，主要在于面向对象方法具有把所建立的模型与问题域进行完整且直接地映射的能力，在整个开发过程中均采用一致的概念和表示法，采用诸如封装、继承和消息等原则使得问题域的复杂性在模型上得以控制。

自 20 世纪 80 年代中后期，面向对象方法得到了迅速的发展，现已经成为一种成熟的软件开发方法，并已经深入到了几乎计算机的所有领域。面向对象的开发方法不仅为人们提供了良好的开发范型，而且在提供软件的生产率、可靠性、易用性和易维护性等方面有明显的效果，已经成为当代计算机界最为关注的一种开发方法[1]。

1.1　面向对象的基本思想

面向对象方法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问题求解，以及如何进行软件系统构造的软件方法学。面向对象方法作为一种方法学，有其自己的基本思想。

较为完善的面向对象的分析与设计方法出现在20世纪80年代末期。我们已经知道，传统的方法只注重从某方面构造系统，而很少涉及其他方面。为了克服传统开发方法的不足，面向对象方法的思路是从现实世界中的客观对象(如人和事物)入手，尽量运用人类的自然思维方式从多方面来刻画并进而实现软件系统，这与传统开发方法构造系统的思想是不一样的。

在面向对象方法中，把一切都看成是对象。下面以开发一个开发票的程序为例，说明这种观点。发票的简化样本如图1-1所示。

编号	商品名称	规格	单位	数量	单价	金额
合计						

图1-1　发票样本

按非面向对象思路，要定义数据结构，然后根据数据结构编写函数或过程来计算购买单项商品的金额和合计金额。而按面向对象思路，把发票看成是一个对象，其中有若干属性，如编号、商品名称和规格等，还有若干操作，如计算购买一种商品金额的操作即“单项金额计算”，和计算合计金额的操作即“发票金额合计”。

面向对象方法是一种运用对象、类、继承、聚合、关联、消息和封装等概念和原则来构造软件系统的开发方法。具体地讲，面向对象方法的基本思想如下：

1)客观世界中的事物都是对象(object)，对象之间存在一定的关系，如复杂对象由简单对象构成。面向对象方法要求从现实世界中客观存在的事物出发来建立软件系统，强调直接以问题域(现实世界)中的事物为中心来认识问题，并根据这些事物的本质特征和系统责任，把它们抽象出来并表示为系统中的对象，作为系统的基本构成单位。这可以使系统的构成能直接映射到问题域，在其中保持问题域中的事物及其相互关系的本来面貌。

2)用对象的属性(attribute)表示事物的数据特征；用对象的操作(operation)表示事物的行为特征。

3)对象的属性与操作结合为一体，成为一个独立的、不可分的实体，实体对外屏蔽其内部细节。

4)通过抽象对事物进行分类。把具有相同属性和相同操作的对象归为一类，类

(class)是这些对象的抽象描述，每个对象是它的类的一个实例。

5)复杂的对象可以用简单的对象作为它的构成部分。

6)通过在不同程度上运用抽象的原则，可以得到较一般的类和较特殊的类，特殊类继承一般类的属性与操作。

7)对象之间通过消息进行通信，以实现对象之间的动态联系。

8)通过关联表达类之间的静态关系。

图 1-2 为上述部分思想的一个示意图。

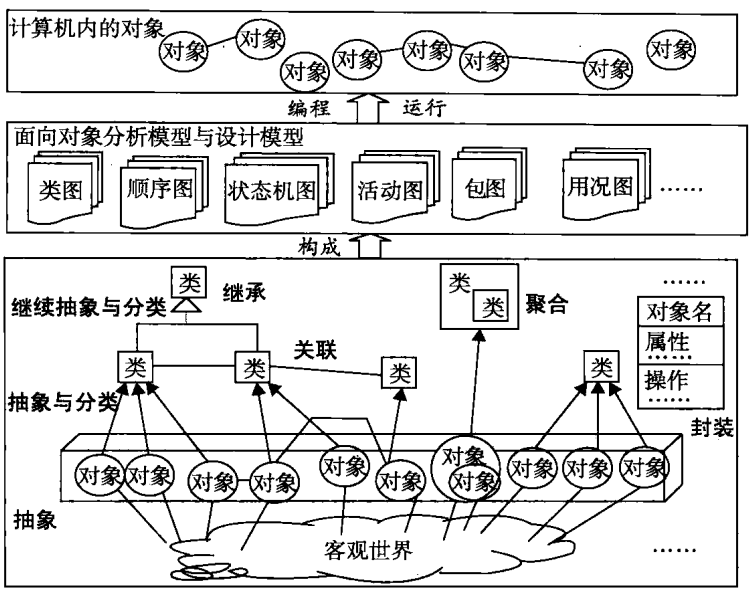


图 1-2　面向对象基本思想(部分)的示意图

利用抽象原则从客观世界中发现对象以及对象间的关系，其中包括整体对象和部分对象，进而再把对象抽象成类，把对象间的关系抽象为类之间的关系。通过继续运用抽象原则，确定类之间存在着的继承关系。上述简略地说明了建立系统的静态结构模型的思想，对于系统的其他模型的建立原则也与此类似，这些内容将是本书讲述的重点。通过以图形的方式作为建模的主要方式之一，分别建立系统的分析与设计模型，进而以此得到可运行的程序。正是通过面向对象建模，对所要解决的问题有了深刻且完整的认识，进而把其转换成可运行的程序，使得程序所处理的对象是对现实世界中的对象进行抽象的结果。

从上述可以看出，面向对象方法强调充分运用人类在日常的逻辑思维中经常采用的思想、方法与原则，例如抽象、继承、聚合、封装和关联等。这使得软件开发者能更有效地思考问题，并以其他人也能看得懂的方式把自己的认识表达出来。

面向对象方法是多种多样的，尽管各种面向对象方法不同，但都是以上述的基本思想为基础的。

一种方法通常要包含一组概念和相应的表示法以及运用它们构造系统的过程指导，面向对象方法也不例外。贯穿于本书的面向对象概念及表示法均取自 UML (Unified

Modeling Language)的2.2版本；至于过程指导，国内外尚无统一标准，本书给出的过程指导是基于特定活动组织的，其特点是易学易用，且不失应用的普遍性。

1.2　面向对象的基本原则

面向对象的基本原则主要有抽象、分类、封装、消息通信、多态性、行为分析和复杂性控制等。

(1)抽象

抽象(abstraction)是指从事物中舍弃个别的、非本质的特征，而抽取共同的、本质特征的思维方式。在面向对象方法中，可从以下几个方面来运用抽象：

- 编程语言的发展呈现抽象层次提高的趋势。例如，用C++编程，不用考虑CPU寄存器和堆栈中存放的内容。目前的大多数编程语言都是从这些细节中抽象出来的。在面向对象编程语言中，存在着过程抽象和数据抽象(抽象数据类型)。对于一个完成确定功能的语句序列，其使用者可把它作为单一的实体(如函数)，尽管实际上它可能是由一系列更细节的运算完成的，这种抽象就是过程抽象。在类的范围内，使用过程抽象来形成操作。数据抽象是指把数据和在数据上进行的操作结合在一起，形成一种新的数据类型。类就是一种数据抽象，栈也是一种数据抽象。
- 在面向对象方法中，对象是对现实世界中的事物的抽象，类是对对象的抽象，一般类是对特殊类的抽象。这种抽象是根据开发的需要进行的。例如，就对象是对现实世界中的事物的抽象而言，高校中的学籍管理系统和伙食管理系统中所使用的学生的信息是不一样的，在不同系统中所抽象出来的对象肯定是不同的；再如，一个现实事物，可能要担任很多角色，只有与问题域和系统责任有关的角色，在系统中才予以考虑。
- 在面向对象的不同开发阶段需要进行不同程度的抽象。典型地，在面向对象分析阶段，先定义类的属性和操作，而与实现有关的因素在设计阶段再考虑。例如，对自动售货机建模，在分析阶段先定义一个类“自动售货机”，根据其收钱和发货的职责定义其属性和操作，其中对外提供的操作为收钱口、选择按钮和发货口(三者形成一个接口)，而对于如何根据实现条件来设计它的内部细节是设计阶段的任务。这与现实生活中一样，我们可以在较高的抽象层次上分析与解决问题，然后再逐步地在较低抽象层次上予以落实。

从上述的自动售货机例子中能看出，使用抽象至少有如下好处：一是便于访问，外部对象只需知道有限的几个操作(作为接口)即可使用自动售货机对象；二是便于维护，如自动售货机的某部分有变化而其接口没有发生变化，只需在自动售货机的内部对该部分进行修改。甚至可用更优的具有相同接口的自动售货机对其进行替换，而不影响使用

者的使用方式。

(2)分类

分类(classification)的作用是按照某种规定划分出事物的类别，以有助于认识复杂世界。

在 OO 中，分类就是把具有相同属性和相同操作的对象划分为一类，用类作为这些对象的抽象描述。如果一个对象是分类(类)的一个实例，它将符合该分类的总体模式。分类实际上是把抽象原则运用于对象描述时的一种表现形式。在 OO 中，还可以进一步地运用分类原则，通过不同程度的抽象，形成一般/特殊结构。

运用分类原则，清楚地表示了对象与类的关系，以及特殊类与一般类的关系。

(3)封装

封装(encapsulation)有两个含义：1)把描述一个事物的性质和行为结合在一起，对外形成该事物的一个界限。面向对象方法中的封装就是用对象把属性和操纵这些属性的操作包装起来，形成一个独立的实体单元。封装原则使对象能够集中而完整地描述并对应一个具体的事物，体现了事物的相对独立性。2)信息隐蔽，即外界不能直接存取对象的内部信息(属性值)并不能访问对象隐藏起来的内部操作，外界也不用知道对象操作的内部实现细节。在原则上，对象对外部仅定义其什么操作可被其他对象访问，而其他的对象不知道所要访问的对象的内部属性和隐藏起来的内部操作以及它是如何实现操作的。

如上所述，封装使得在对象的外部不能随意存取对象的内部数据和操作，而只允许通过由对象提供的外部可用的操作来访问其内部，这就降低了对象间的耦合度，还可避免外部错误对它的“交插感染”。另外，通过封装使得对象的内部修改对外部的影响变小，减少了修改对外引起的“波动效应”。图 1-3 所示的是封装的示意图，其中的一部分操作是外部可用的。

严格的封装也会带来问题，如必须为外部要访问的私有属性编写读写函数，这致使编程麻烦，也有损于执行效率。有些语言不强调严格的封装，而使用可见性控制，以此来解决问题。例如，C++ 和 Java 就是这样的语言，通过定义对象的属性和操作的可见性，它们都能对外规定其他对象对其属性和操作的可访问性，一个对象也可以提供仅局限于特定对象可访问的属性和操作，这可以通过把相应的可见性指定为受保护的或私有的来达到目的。

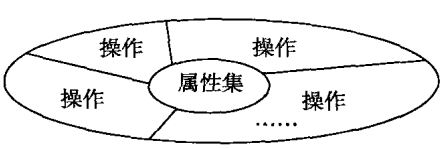


图 1-3　封装的示意图

(4)消息通信

对象之间要通过**消息**(message)进行交互。消息必须直接发给特定的对象，消息中包含所请求服务的必要信息，且遵守所规定的通信规约。一条消息应包括：消息名、入口参数和返回参数。一个对象可以是消息的发送者，也可以是消息的接收者，还可以作为消息中的参数。

(5)多态性

多态性(polymorphism)是指在具有继承关系的类层次结构中可以定义同名的操作或属性，但这些操作或属性具有不同的含义，即表现出不同的行为或具有不同的数据类型。这样，针对同一个消息，不同的对象对其进行响应，所体现出来的行为是不同的。

(6)行为分析

关系机制为我们提供了用关联、继承和聚合等组织类的方法。很多面向对象学者把模型的这部分结构称为静态模型，也有的称其为结构模型。通常，对系统还需要进行**行为分析**：

- 对于一个对象，由于其内的属性值在不断地在发生着变化，按一定的规则可把对象划分为不同的状态。在一个状态内，在请求对象操作时，可执行该状态内规定的操作，也可能会使对象的状态发生改变。通过状态机图可以分析对象在各状态内可执行的操作以及状态变迁情况。
- 系统中的对象是相互协作的，通过发消息共同完成某项功能。这种协作的交互性，可以用交互图来进行描述。
- 很多系统具有并发行为。从事物的并发行为的起因上看，每个并发行为是主动发生的。体现在对象上，就是有一种对象是主动的，它代表着一个进程或线程。在交互图上也能体现出对象间的并发行为。

(7)复杂性控制

为了控制系统模型的复杂性，引入**包**(package)的概念。使用包可以把模型元素组织成不同粒度的系统单元，也可以根据需要用包来组织包。例如，用分析包和设计包来分别组织分析模型和设计模型，以显式地描述不同抽象层次的模型；在类图中也可以根据类之间关系的紧密程度(如类之间存在着继承结构)用包来组织类。

1.3　面向对象方法的主要优点

我们首先从认识论的角度和软件工程方法的角度看一下面向对象方法带来的益处，再把面向对象方法与传统方法进行比较，看面向对象方法有什么优点。

(1)从认识论的角度，面向对象方法改变了开发软件的方式

面向对象方法从对象出发认识问题域，对象对应着问题域中的事物，其属性与操作分别刻画了事物的性质和行为，对象之间的继承、聚合、关联和依赖关系如实地表达了问题域中事物之间实际存在的各种关系。因此，无论系统的构成成分，还是通过这些成分之间的关系而体现的系统结构，都可直接地映射到问题域。这使得使用面向对象方法有利于正确理解问题域及系统责任。

(2)面向对象语言使得从客观世界到计算机的语言鸿沟变窄

图1-4为一个示意图，说明了面向对象语言如何使得从客观世界到计算机的语言鸿