

高 等 学 校 教 材

面向对象方法 基础教程

麻志毅 邵维忠 编著



高等教育出版社

高等学校教材

面向对象方法基础教程

麻志毅 邵维忠 编著

高等教育出版社

内 容 提 要

本书系统讲述了面向对象的基本思想、主要概念以及相应的表示法,并给出了详细的过程指导。主要包括面向对象的分析、设计和实现三部分。本书注重理论与实践相结合,通过给出大量的例题、较为详尽的案例分析以及详述对建模概念的运用,阐明了如何用面向对象方法开发软件系统。

本书可作为高等院校计算机软件及相关专业的教材,也可作为工程硕士、在职人员培训班的教材,对从事软件开发的工程技术人员具有很大的参考价值。

图书在版编目(CIP)数据

面向对象方法基础教程/麻志毅,邵维忠编著. —北京:高等教育出版社,2004.2 (2006 重印)

ISBN 7-04-014236-8

I.面… II.①麻…②邵… III.面向对象语言—程序设计—高等学校—教材 IV.TP312

中国版本图书馆 CIP 数据核字(2004)第 002087 号

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京北苑印刷有限责任公司

开 本 787×1092 1/16
印 张 19
字 数 380 000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2004 年 2 月第 1 版
印 次 2006 年 6 月第 2 次印刷
定 价 26.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 14236-00

前 言

20 世纪 90 年代,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。90 年代中期,该技术在大多数发达国家的软件开发中已经得到相当广泛的运用,在我国也出现了若干成功的应用先例。例如,由北京大学等 20 多家单位于 1995 年研制成功的大型软件开发环境青鸟系统,就是全面采用面向对象技术开发的。我国软件产业界对面向对象技术的学习和应用热潮出现于 20 世纪 90 年代的后期。越来越多的软件企业认识到,要在激烈的市场竞争中立于不败之地,采用面向对象技术是势在必行的选择。

产业界需要大量的掌握面向对象方法与技术的人才。这些人才不仅仅是能够使用一种或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建模,即通过面向对象的分析(OOA)和面向对象的设计(OOD)建立系统的分析模型和设计模型。这一形势对高等院校计算机软件专业的课程设置的教材建设提出了新的要求。为此,北京大学计算机科学技术系在 1991 年就开设了“面向对象的分析与设计”研究生课程。有关的同志翻译了一些国外的著作[2][3]作为教材和参考书,进而撰写了自己的学术著作《面向对象的系统分析》[32]和《面向对象的系统设计》[33]作为研究生课程教材,这些教材和参考书已被国内许多高等院校所采用。

邵维忠、杨芙清合著的[32]和[33]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特色。其中最主要的特点有如下两方面:一是提倡充分运用面向对象方法的基本概念,通过加强过程指导而保持建模概念的简练;二是其 OOD 部分比以往的著作内容更为详细,并且更强调用面向对象概念表达各种全局性的设计决策。

当前,面向对象技术在计算机软件领域中快速普及的形势要求在高等院校的本科教学中也开设相应的课程。[32]和[33]是两部学术专著,比较适合作为研究生教材。尽管目前许多高等院校也在将这两本书作为本科教材使用,但其中有些内容在知识的深度和广度上超出了对本科生的要求。本书旨在提供一本更适合本科教学的教材。在思想体系上,本书继承了[32]和[33]所提出的理论和方法;在内容上,本书的若干章节或段落直接采用了[32]和[33]的内容。但为了更适合于本科教学,本书与[32]和[33]相比有如下特色:

- (1) 适当压缩了面向对象分析与面向对象设计部分的篇幅,增加了面向对象的实现部分的内容;
- (2) 减少了一些理论阐述和不同学术观点的讨论,增加了对如何运用概念的讲解;
- (3) 在各章的正文部分增加了例题,在各章之后给出适量习题;

(4) 通过一个案例,讲述了如何运用面向对象方法进行分析与设计。

本书既可作为教材,也可作为从事软件开发的工程技术人员参考书。由于以上几个特点,本书与[32]和[33]相比具有更强的普及性,适合于更广大的读者群。

作为一种实用的面向对象方法教材和工程技术用书,既要注重概念与表达能力的完整性,又要避免过分复杂。在这方面,本书坚持了[32]和[33]提出的宗旨:采用一个尽可能简练的建模概念集合,通过加强过程指导,解决各种复杂的分析与设计问题。本书所采用的概念和表示法与统一建模语言(Unified Modeling Language, UML)保持一致,但只使用了其中最必要的一部分概念。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”完全一致。我们作为该规范的主要起草者,曾经与国内众多专家、学者和企业界的专业人士进行过反复的研究讨论,并达成共识。

UML是一个由对象管理组织(OMG)采纳的建模语言规范,目前在软件工业界已经被广泛接受。但UML的内容过于庞大和复杂,学习难度很大,这是UML本身的复杂性造成的。UML中的许多内容用于构造UML元模型,对于大多数面向应用的软件开发来说,这些概念是用不着的。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,它要尽量适合于各领域。特别是UML不仅仅是用于面向对象的软件建模语言,还可用于对其他方面的建模,例如在建筑业或机器制造业进行建模。基于上述原因,本书选用了UML中最常用的概念,以控制方法的复杂性。由于本书加强了运用基本概念解决各种复杂的分析与设计问题的过程指导,因此所选用的概念和表示法仍能保持表达能力的完整性。为了使读者能够理解常见的UML概念(尽管这些概念并不是必不可少的),我们适当地进行了讲解,同时也给出了一些运用最基本的面向对象概念代替这些概念的方法。

进行软件开发应该遵循一定的开发过程指导。过程指导为完成软件系统开发的每一步提供详细指导,其中包括具体的模型、工具和技术。本书所讲述的过程指导的思想来自参考文献[32]和[33],即本书所采用的开发过程是在借鉴了较为流行的多种开发过程的基础上,根据青鸟工程的成果和我们进行科研及工程实践的经验总结出来的。

像使用其他开发方法一样,用面向对象方法进行软件开发的目的是要建立相应的模型。总的来讲,本书分为功能需求模型、分析模型、设计模型和实现模型。针对建立各种模型所使用的图以及其中一些具体的模型元素,本书还给出了相应的用于详细说明的模板。

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

下面简要地介绍本书的概貌,使读者对全书有一个提纲挈领的了解。

第一章集中介绍了面向对象方法的基本思想和原则,解释了它的基本概念,论述了它的主要优点,并简介了它的发展历史和现状。

第二章首先概述了面向对象分析所面临的问题,然后对其进行了综述。在综述中阐述

了面向对象分析模型和过程模型。

第三章全面地讲解了建立功能需求模型所需要使用的概念及其表示法,并详述了如何使用它们来建立功能需求模型。

第四章详细地介绍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。

第五章详细地介绍了建立顺序图所使用的概念与表示法,并详述了如何建立顺序图。

第六章讲述 UML 中的几种其他图:协作图、状态图和包图。协作图和状态图与顺序图一样,是用于建立系统动态模型的;包图说明了如何用包组织模型元素以及按组控制模型元素的可见性。

第七章首先简述了面向对象设计的发展史,然后说明了分析与设计的关系,并阐述了面向对象设计模型和过程模型。

第八章详述了如何针对实现条件对分析结果进行补充与调整。

第九章详述了进行人机交互设计所需考虑的因素,并从分析和设计两个方面详述如何进行人机交互设计。

第十章详述了什么是控制流,如何识别与定义控制流,以及如何协调控制流之间的同步。

第十一章讲述对数据管理部分的设计。首先对数据库进行了简介,然后详细地讲解了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。

第十二章针对如何对系统的构件进行描述与构造,详细地讲解了构件图及其应用。本章还对部署图进行了简介。

第十三章从耦合、内聚、复用等方面讲述了如何评价面向对象的设计模型。

第十四章为面向对象的编程实现。本章首先对面向对象的编程语言进行了简介,然后用大量的例题说明了面向对象的概念在具体的面向对象语言中的实现。本章还对数据管理部分的实现和状态图的实现进行了具体的详述。

第十五章首先讲述如何把一个较为复杂的系统划分成一系列子系统,然后说明了如何用包对系统或子系统进行可视化建模,以及从那些方面建立的系统的模型。此外还阐述了如何保证模型的一致性。

第十六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。

最后,本书给出了两个附录。第一个附录讲述了面向对象的软件建模工具的构成,并给出了一个实例。第二个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。

本书在写作过程中得到了杨芙清院士的关心与指导,作者在她所领导的学术队伍中获得了良好的工作条件,本书同时得到了国家自然科学基金项目(60073015)和国家 863 高新

技术研究发展计划项目(2001AA113060 和 2001AA113070)的资助,在此表示衷心感谢!

对书中存在的错误和疏漏之处,恳请各位读者批评指正。

作 者

2003 年 12 月 于北京大学

策划编辑	刘建元
责任编辑	武林晓
封面设计	李卫青
责任印制	毛斯璐

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》,其为人将承担相应的民事责任和行政责任,构成犯罪的,将被依法追究刑事责任。为了维护市场秩序,保护读者的合法权益,避免读者误用盗版书造成不良后果,我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为,希望及时举报,本社将奖励举报有功人员。

反盗版举报电话: (010) 58581897/58581896/58581879

传 真: (010) 82086060

E - mail: dd@hep.com.cn

通信地址: 北京市西城区德外大街 4 号

高等教育出版社打击盗版办公室

邮 编: 100011

购书请拨打电话: (010)58581118

目 录

第一部分 概 述

第1章 面向对象方法概论	(3)
1.1 传统软件开发方法中存在的问题	(3)
1.2 面向对象的基本思想	(5)
1.3 面向对象的主要概念及基本原则	(6)
1.3.1 面向对象的主要概念	(6)
1.3.2 面向对象的基本原则	(9)
1.4 面向对象方法的主要优点	(13)
1.5 面向对象方法与技术的发展史 及现状简介	(17)
习题	(18)

第二部分 面向对象分析

第2章 什么是面向对象的分析	(21)
2.1 分析面临的主要问题	(21)
2.2 面向对象的分析综述	(24)
习题	(28)
第3章 用况图	(29)
3.1 系统边界	(29)
3.2 参与者	(30)
3.2.1 概念与表示法	(31)
3.2.2 识别参与者	(32)
3.3 用况	(32)
3.3.1 概念与表示法	(33)
3.3.2 用况和参与者之间的关系	(35)
3.3.3 用况之间的关系	(36)
3.3.4 捕获用况	(38)
3.3.5 用况模板	(40)
3.4 用况图	(41)
3.5 检查与调整	(41)

3.6 例题	(43)
习题	(46)
第4章 类图	(47)
4.1 对象与类	(47)
4.1.1 概念与表示法	(47)
4.1.2 识别对象与类	(48)
4.1.3 审查与筛选	(50)
4.1.4 抽象出类并进行调整	(52)
4.1.5 认识对象的主动行为 并识别主动对象	(53)
4.1.6 类的命名	(54)
4.1.7 建立类图的对象层	(54)
4.2 定义属性与操作	(55)
4.2.1 定义属性	(55)
4.2.2 定义操作	(58)
4.3 建立关系	(61)
4.3.1 继承	(62)
4.3.2 关联	(72)
4.3.3 聚合	(83)
4.3.4 依赖	(86)
4.4 接口	(86)
习题	(89)
第5章 顺序图	(90)
5.1 概念与表示法	(90)
5.2 建立顺序图	(95)
习题	(96)
第6章 统一建模语言中的几种其他图	(97)
6.1 协作图	(97)
6.1.1 概念与表示法	(97)
6.1.2 建立协作图	(99)
6.2 状态图	(99)

6.2.1 概念与表示法	(100)	的界面成分	(147)
6.2.2 建立状态图	(110)	习题	(150)
6.3 包图	(111)	第 10 章 控制驱动部分的设计	(151)
6.3.1 概念与表示法	(111)	10.1 什么是控制驱动部分	(151)
6.3.2 如何划分包	(113)	10.2 控制流	(152)
习题	(114)	10.3 如何设计控制驱动部分	(153)
第三部分 面向对象设计		10.3.1 识别控制流	(153)
第 7 章 什么是面向对象设计	(117)	10.3.2 审查	(155)
7.1 面向对象设计的发展	(117)	10.3.3 定义控制流	(155)
7.2 OOA 与 OOD 的关系	(118)	10.3.4 进程间和线程间的通信	(156)
7.3 面向对象设计模型和过程	(120)	10.3.5 控制流间的同步	(158)
习题	(122)	习题	(160)
第 8 章 问题域部分的设计	(123)	第 11 章 数据管理部分的设计	(161)
8.1 为复用类而增加结构	(123)	11.1 什么是数据管理部分	(161)
8.2 增加一般类以建立共同协议	(124)	11.2 数据库和数据库管理系统	(161)
8.3 按编程语言调整继承	(125)	11.2.1 关系数据库	(162)
8.4 提高性能	(128)	11.2.2 面向对象数据库	(163)
8.5 为实现永久对象的存储增补属性 与操作	(130)	11.3 如何进行数据管理部分设计	(164)
8.6 对复杂关联的转化并决定关联的 实现方式	(131)	11.3.1 针对关系数据库的数据 存储设计	(164)
8.7 调整与完善属性	(132)	11.3.2 针对面向对象数据库的数据 存储设计	(171)
8.8 构造或优化算法	(134)	11.3.3 针对文件的数据存储设计	(172)
8.9 决定对象间的可访问性	(134)	习题	(172)
8.10 定义对象实例	(135)	第 12 章 构件及其部署的设计	(173)
8.11 其他	(136)	12.1 构件图	(173)
习题	(136)	12.1.1 构件	(173)
第 9 章 人机交互部分的设计	(137)	12.1.2 构件图	(175)
9.1 什么是人机交互部分	(137)	12.2 部署图	(177)
9.2 人机交互部分的分析	(138)	12.2.1 节点	(177)
9.3 人机界面的设计准则	(140)	12.2.2 部署图	(179)
9.4 人机交互的面向对象设计	(142)	习题	(179)
9.4.1 设计输入与输出	(142)	第 13 章 OOD 的评价准则	(180)
9.4.2 命令的组织	(144)	13.1 耦合	(180)
9.4.3 用面向对象概念表达所有		13.2 内聚	(181)
		13.3 复用	(181)

13.4 其他评判准则·····	(182)
习题·····	(183)

第四部分 面向对象的编程实现

第 14 章 面向对象的编程实现·····	(187)
14.1 面向对象的程序设计·····	(187)
14.1.1 面向对象程序设计语言 简介·····	(187)
14.1.2 用非面向对象程序设计语言 实现 OOD 模型·····	(191)
14.1.3 为实现 OOD 模型选择程序 设计语言·····	(191)
14.2 用具体的面向对象语言实现 面向对象概念·····	(193)
14.3 数据管理部分的实现·····	(205)
14.4 状态图的实现·····	(211)
习题·····	(215)

第五部分 系统与模型

第 15 章 系统与模型·····	(219)
15.1 系统与子系统·····	(219)
15.1.1 对体系结构模式建模·····	(220)
15.1.2 划分子系统·····	(221)
15.2 模型·····	(223)
15.2.1 模型的含义·····	(224)
15.2.2 模型的视图·····	(224)
15.2.3 模型的层次性·····	(226)
15.2.4 模型间的一致性检查·····	(226)
习题·····	(227)

第六部分 建模实例

第 16 章 案例:会议文件审批系统·····	(231)
16.1 系统的功能需求·····	(231)
16.2 需求捕获·····	(233)
16.2.1 建立界面原型·····	(233)
16.2.2 识别参与者·····	(235)

16.2.3 识别用况·····	(235)
16.2.4 对需求进行捕获与描述·····	(236)
16.3 分析·····	(245)
16.3.1 寻找类·····	(245)
16.3.2 绘制状态图·····	(246)
16.3.3 建立类图·····	(247)
16.3.4 建立顺序图·····	(251)
16.4 设计·····	(253)
习题·····	(263)

附 录

附录 A 面向对象的软件建模工具·····	(267)
A.1 为什么需要软件建模工具·····	(267)
A.2 建模工具简介·····	(268)
A.3 面向对象的软件建模工具所应 具有的功能·····	(269)
A.4 面向对象工具遵循的标准·····	(272)
A.5 面向对象的软件建模工具 JBOO 3.0 简介·····	(273)
A.5.1 JBOO 3.0 是一种面向对象的 软件建模工具·····	(273)
A.5.2 JBOO 3.0 的功能·····	(274)
附录 B 文档编制指南·····	(280)
B.1 总体说明文档·····	(280)
B.2 用况图文档·····	(280)
B.3 类图文档·····	(281)
B.4 顺序图文档·····	(282)
B.5 协作图文档·····	(282)
B.6 状态图文档·····	(283)
B.7 活动图文档·····	(283)
B.8 构件图文档·····	(284)
B.9 部署图文档·····	(284)
B.10 包图文档·····	(284)
附录 C 英汉术语对照表·····	(286)
参考文献·····	(288)

第一部分

概 述

第 1 章 面向对象方法概论

本章首先简要地回顾一下传统软件开发方法中存在的问题,以此引出面向对象方法是怎样对软件开发进行改进的。随后重点讲述了面向对象的基本思想、主要概念和基本原则,并论述了面向对象方法的主要优点。最后简要地介绍了面向对象技术的发展史及现状。通过对本章的学习,首先了解面向对象方法的主要内容,掌握基本知识,为进一步学习利用面向对象方法进行软件开发打下基础。

1.1 传统软件开发方法中存在的问题

在 20 世纪 60 年代以前,软件开发构造的软件系统大多都是较小的,且相对简单。那时所用的编程语言及编程环境也相对简单,常见的编程语言有汇编语言、FORTRAN 和 COBOL 等。当时,人们认为软件开发是一项很强的依赖个人技巧和技术能力的艺术性劳动,崇尚程序员的个人技能,没有认识到需要使用什么方法,因此,当时明显地注重个人的作用。那时产生的代码,按现在的人们所形容的,是意大利细面条式的,因为代码中含有较多的 GOTO 语句。随着软件复杂性的增长,那样的随心所欲的方法就不再是可接受的了,因为这样的代码是很难维护的。高层次语言的引入有助于解决一些与复杂性有关的问题,但这些语言并不是充分的,因为软件开发也需要方法。

在随后出现了多种开发方法,这些开发方法都能解决一些问题,但也都存在着一定的局限性。下面对两种典型的软件开发方法进行简要的分析,以找出其中存在的主要问题。

1. 结构化方法

结构化方法是结构化分析、结构化设计和结构化编程的总称。结构化方法的思想萌发于 20 世纪 60 年代, Tom DeMacro 和 Ed Yourdon 等人于 20 世纪 70 年代系统地提出了结构化方法思想。该方法强调开发方法的结构合理性以及所开发软件的结构合理性。其目的在于,提供一种有序的机制,通过抽象来理解待开发的系统,试图产生结构良好的软件系统。

在那时,人们都认为这种方法非常自然,因为它是按功能组织的。当时的计算机应用还不是很普及,只是特定的用户按照自己的需要对软件系统提出功能性的要求。结构化方法的发明,在相当大的程度上解决了以前存在的问题,且与模块化编程结合使用后,使得开发效率有很大的提高。特别是模块化编程删除了遭到人们一致反对的 GOTO 语句,使软件得

到了有效的维护。这些是结构化方法在当时大受欢迎的主要原因。

结构化分析和设计注重于处理功能,在数据与对数据的操作上结合得不紧密。此外,分析与设计阶段所采用的概念与表示法存在着不一致。

当时用结构化方法实现了很多软件系统,但由于上述原因,其在系统的开发与维护中的问题也日益显现出来。对于功能稳定的应用领域,如某些科学计算,结构化方法是很适用的。但对于众多的领域而言,其功能是易变的,如企业管理和商业管理领域就是如此,因为这些领域要随着市场的变化不断地进行调整。对于大型、复杂的软件系统,由于其中内在的高耦合、低内聚而导致了系统的成功率不高,特别是当时团体的开发与管理方法的不足,使得在 20 世纪 70 年代的软件危机情况更加严重。为了解决软件危机,人们对结构化方法进行了一定的改进,对编程语言也进行了革新,如产生了用于软件开发的 4GL、CASE 工具、原型技术和代码生成器等。这些努力取得了一定的成就,但并没有从根本上解决问题。

总之,在软件开发中使用经典的结构化方法,对于较为复杂的系统而言,容易导致模块的低内聚和模块间的高耦合,从而使得系统缺乏灵活性和可维护性。这就需要更好的软件开发方法,它应能管理系统的复杂性,满足用户易变的需求,即所构造的系统应该具有灵活性和可维护性。

2. 信息建模方法

20 世纪 70 年代末, Peter Chen 提出了实体 - 关系法。1981 年, M. Flavin 将实体 - 关系法与 Ed Codd 的关系数据库理论相结合,提出了一种新的软件开发方法,即信息建模方法。它以为实体的数据集合作为系统的构造块,以数据结构为中心来开发软件。因为有相当多的人都认为实体是稳定的,并且关系数据库有一个相当好的数学基础,所以当时的大多数公司都使用信息建模方法开发软件。

对于数据及其关系比较复杂的系统来说,信息建模方法很有用。但它也存在弱点,且正好与结构化方法相反,即它对功能的处理很弱。这种方法从理论上讲,所有的功能可以通过与一阶谓词演算相符的语言结构来定义,但现实是,大多数需要解决的问题不能通过使用一阶谓词得到解决。

几乎所有的传统方法都只注重于系统的一个方面,对系统的其他方面建模的能力却很弱。例如,结构化方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,即使是在数据流图中,对数据组织的支持也不很强;在信息建模方法中的构造块是实体,强调对数据的组织,但该方法忽略了系统功能。此外,这两种方法都没有较强地描述系统的动态行为的能力。

到 20 世纪末,软件工程师和软件研究人员已经尝试了至少 30 年,试图寻找有效地开发较为复杂的软件系统的方法。为了克服传统方法的不足,人们经过坚持不懈的努力,终于形成了目前的面向对象方法。

面向对象方法是在传统方法的基础上发展而来的,例如,它仍然使用抽象和模块化等概念。面向对象方法对信息建模方法进行了发展,例如,类就是对实体的发展;但与结构化方法相比,面向对象方法发生了根本性的变化,它所建立的模型对问题域进行了完整与直接的映射,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得到了控制。

1.2 面向对象的基本思想

面向对象方法已深入到计算机软件领域的几乎所有分支。它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 年代中后期。传统的软件开发方法注重从一个方面构造系统,而对其他方面考虑得较少。为了克服传统开发方法的不足,面向对象方法解决问题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构造软件系统,这与传统的结构化方法注重用功能构造系统和信息建模方法用信息实体构造系统是不一样的。

在面向对象方法中,把一切都看成是对象。例如,编写一个开发票程序,需要对发票的明细栏进行计算,如图 1.1 所示。

编号	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,要定义数据结构以及利用数据结构进行计算的函数或过程。而按面向对象思路,把发票看成是一个对象,其中有若干属性和操作,例如,把其中的一个操作定义为“发票总计”,用该操作来合计各明细栏的金额。

目前,人们普遍地认为面向对象方法是一种运用对象、类、继承、封装、聚合、关联、消息和