

面向 **21** 世纪

高等学校计算机类专业系列教材

# 面向对象程序设计教程

*A Course in Object-Oriented Programming*

龙昭华 编著

王小民 主审

西安电子科技大学出版社  
<http://www.xduph.com>

面向 21 世纪高等学校计算机类专业系列教材

# 面向对象程序设计教程

A Course in Object-Oriented Programming

龙昭华 编著

王小民 主审

西安电子科技大学出版社

2003

## 内 容 简 介

本书立足面向对象方法学,系统地介绍了面向对象编程思想。为了使读者掌握一定的实际编程能力,本书还介绍了面向对象编程语言基础。为了满足高级编程的需要并使读者了解面向对象技术的发展,本书还专门介绍了 STL 和 UML。

本书力求以面向对象方法学为指导,各章节以面向对象软件开发的必要过程进行安排。本书可作为本科生、研究生的教材使用,也可供从事计算机软件开发的技术人员、相关专业的教师阅读。教师在授课时,可根据需要对部分内容进行适当取舍。

★本书配有电子教案,任课教师可免费索取。

### 图书在版编目(CIP)数据

面向对象程序设计教程=A Course in Object-Oriented Programming / 龙昭华编著.

—西安:西安电子科技大学出版社,2003.8

(面向 21 世纪高等学校计算机类专业系列教材)

ISBN 7-5606-1278-4

I. 面… II. 龙… 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(2003)第 056455 号

责任编辑 马武装 王素娟

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)8242885 8201467 邮 编 710071

http://www.xduph.com

E-mail: xdupfb@pub.xaonline.com

经 销 新华书店

印 刷 西安兰翔印刷厂

版 次 2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 17.5

字 数 408 千字

印 数 1~4 000 册

定 价 19.00 元

ISBN 7-5606-1278-4/TP·0675(课)

**XDUP 1549001-1**

\*\*\*如有印装问题可调换\*\*\*

本社图书封面为激光防伪覆膜,谨防盗版。

# 序

第三次全国教育工作会议以来，我国高等教育得到空前规模的发展。经过高校布局和结构的调整，各个学校的新专业均有所增加，招生规模也迅速扩大。为了适应社会对“大专业、宽口径”人才的需求，各学校对专业进行了调整和合并，拓宽专业面，相应的教学计划、大纲也都有了较大的变化。特别是进入21世纪以来，信息产业发展迅速，技术更新加快。面对这样的发展形势，原有的计算机、信息工程两个专业的传统教材已很难适应高等教育的需要，作为教学改革的重要组成部分，教材的更新和建设迫在眉睫。为此，西安电子科技大学出版社聘请南京邮电学院、西安邮电学院、重庆邮电学院、吉林大学、杭州电子工业学院、桂林电子工业学院、北京信息工程学院、深圳大学、解放军电子工程学院等10余所国内电子信息类专业知名院校长期在教学科研第一线工作的专家教授，组成了高等学校计算机、信息工程类专业系列教材编审专家委员会，并且面向全国进行系列教材编写招标。该委员会依据教育部有关文件及规定对这两大类的教学计划和课程大纲，对目前本科教育的发展变化和相应系列教材应具有的特色和定位以及如何适应各类院校的教学需求等进行了反复研究、充分讨论，并对投标教材进行了认真评审，筛选并确定了高等学校计算机、信息工程类专业系列教材的作者及审稿人。这套教材预计在2004年春季全部出齐。

审定并组织出版这套教材的基本指导思想是力求精品、力求创新、好中选优、以质取胜。教材内容要反映21世纪信息科学技术的发展，体现专业课内容更新快的要求；编写上要具有一定的弹性和可调性，以适合多数学校使用；体系上要有所创新，突出工程技术型人才培养的特点，面向国民经济对工程技术人才的需求，强调培养学生较系统地掌握本学科专业必需的基础知识和基本理论，有较强的本专业的基本技能、方法和相关知识，培养学生具有从事实际工程的研发能力。在作者的遴选上，强调作者应在教学、科研第一线长期工作，有较高的学术水平和丰富的教材编写经验；教材在体系和篇幅上符合各学校的教学计划要求。

相信这套精心策划、精心编审、精心出版的系列教材会成为精品教材，得到各院校的认可，对于新世纪高等学校教学改革和教材建设起到积极的推动作用。

系列教材编委会  
2002年8月

# 高等学校计算机、信息工程类专业

## 系列教材编审专家委员会

- 主任：杨震（南京邮电学院副院长、教授）  
副主任：张德民（重庆邮电学院通信与信息工程学院院长、教授）  
韩俊刚（西安邮电学院计算机系主任、教授）  
李荣才（西安电子科技大学出版社总编辑、教授）

### 计算机组

- 组长：韩俊刚（兼）  
成员：（按姓氏笔画排列）  
王小民（深圳大学信息工程学院计算机系主任、副教授）  
王小华（杭州电子工业学院计算机分院副院长、副教授）  
孙力娟（南京邮电学院计算机系副主任、副教授）  
李秉智（重庆邮电学院计算机学院院长、教授）  
孟庆昌（北京信息工程学院教授）  
周娅（桂林电子工业学院计算机系副主任、副教授）  
张长海（吉林大学计算机科学与技术学院副院长、教授）

### 信息工程组

- 组长：张德民（兼）  
成员：（按姓氏笔画排列）  
方强（西安邮电学院电信系主任、教授）  
王晖（深圳大学信息工程学院电子工程系主任、副教授）  
胡建萍（杭州电子工业学院电子信息分院副院长、副教授）  
徐祎（解放军电子工程学院电子技术教研室主任、副教授）  
唐宁（桂林电子工业学院通信与信息工程系副主任、副教授）  
章坚武（杭州电子工业学院通信工程分院副院长、教授）  
康健（吉林大学通信工程学院副院长、教授）  
蒋国平（南京邮电学院电子工程系副主任、副教授）

- 总策划：梁家新  
策划：马乐惠 云立实 马武装 马晓娟  
电子教案：马武装

# 前 言

自 20 世纪 80 年代以来,面向对象程序设计(Object-Oriented Programming)被普遍认为是程序设计方法学及软件领域的实质性革命。面向对象思想及其软件编程实践在软件领域中得到了很大的发展和完善。近年来,高等院校有关专业均陆续开设了这门课程。C++来源于 C,是支持多种范型的程序设计语言。许多高校在讲授面向对象程序设计课程时,都将 C++作为重点内容讲述,这无疑是一种好的选择。

在计算机软件领域,很多新的方法与技术都首先是从编程阶段出现,进而发展到软件生命周期的前期阶段。人们对软件本身的认识,开始也只是着眼于编程,后来才形成了分析、设计、编程、测试以及维护等一整套的软件工程体系。但是,面向对象的方法与技术发展到软件生命的前期阶段后有着更为深刻的意义。因为面向对象方法的本质,就是强调从客观世界中固有的事物出发来构造系统,用人类在现实生活中常用的思维方式来认识、理解和描述客观事物,突出最终建立的系统使之能够映射问题域。本书也是围绕这一方式来编写的,目的是使读者能够很清晰地把握面向对象的精髓,更好地掌握面向对象方法。

本书的编者一直从事计算机语言和面向对象课程的教学,本书的整体框架是从编者第一手教案中整理出来的,不求对 C++语言语法细节的罗列介绍。从未接触过 C++语言和 Java 语言的读者,可以用较快的速度浏览一下其基本语法(本书第 3 章)。

本书在编写过程中做到了以下几点:

(1) 既照顾面向对象思想又照顾其编程实现。这对初学者掌握面向对象方法有很大的好处。

(2) 本书考虑到学生是在学习完结构化程序设计语言(如 C 语言)后接着学习 C++语言,会用结构化程序设计的思想理解 C++,从而淡化了 C++作为优秀的多范型语言的特点的实际情况,对本书内容进行了有的放矢的编写。

(3) 本书以“由点到线,再到面”这一逻辑思维为出发点,紧紧围绕面向对象思想进行阐述。

因此,本书从基本概念、基本语法到模型的建立,力求体现面向对象软件工程的整个体系,并对标准模板库(STL)和标准建模语言(UML)作了重点阐述。本书的整个编写过程清晰,提倡既不要陷入具体语言中,又努力做到联系实际:既寻求规律,又有所创新,使读者对面向对象有一个比较全面的认识,并从本书的学习中受到启迪。

在本书的完成整理过程中,教师蒋贵全、范时平,学生周敏、周大刚、张贵和秦志胜等参与了相关章节的编写录入和校对工作。

本书作为本科教材时,教学重点是第 1~5 章,带\*号章节为选修内容;作为研究生教材时,教学重点放在第 6、7 章。本书也可供从事软件设计的技术人员参考。

本书得到了西安电子科技大学出版社马武装老师的大力支持。本书由深圳大学王小民博士审定，重庆邮电学院李秉智教授阅读了本书的初稿。

由于时间关系加上编者水平有限，疏漏和错误之处敬请谅解，并诚恳期待读者批评指正。

编著者

2003年5月

# 目 录

第 1 章 绪论 .....	1
1.1 面向对象简述 .....	1
1.1.1 对象定义 .....	1
1.1.2 面向对象 .....	2
1.1.3 面向对象范型 .....	5
1.2 面向对象程序设计范型 .....	7
1.2.1 软件工程 .....	7
1.2.2 面向对象理论及历史 .....	8
1.2.3 面向对象编程 .....	14
习题一 .....	16
第 2 章 面向对象分析 .....	17
2.1 面向对象分析中的术语 .....	17
2.1.1 抽象 .....	17
2.1.2 对象 .....	19
2.1.3 类 .....	19
2.1.4 封装 .....	20
2.1.5 继承 .....	20
2.1.6 消息 .....	21
2.1.7 结构与联系 .....	21
2.1.8 多态性 .....	22
2.1.9 主动对象 .....	23
2.2 各种分析方法 .....	23
2.2.1 功能分解法 .....	23
2.2.2 数据流法 .....	24
2.2.3 信息建模法 .....	24
2.2.4 面向对象法 .....	25
2.3 支持软件复用 .....	25
2.3.1 贯穿软件生命周期全过程的一致性 .....	25
2.3.2 实用性 .....	26
2.3.3 有利于用户参与 .....	26
2.4 面向对象分析设计的主要过程 .....	26
2.4.1 发现对象 .....	26
2.4.2 定义属性 .....	27
2.4.3 定义行为 .....	28
2.4.4 定义结构 .....	28



2.4.5 细分题目 .....	31
2.5 具有支持 OO 特征的建模语言 .....	33
2.5.1 OOA 建模语言的评价 .....	33
2.5.2 OOA 模型的一般机制 .....	33
2.5.3 OOA 视图和提交 .....	34
2.6 面向对象设计初步 .....	34
2.6.1 OOD 模型 .....	34
2.6.2 优良的 OOD .....	35
2.6.3 OOD 中对象标识设计 .....	35
2.6.4 复杂对象的构造设计 .....	35
习题二 .....	36
<b>第 3 章 面向对象程序设计语言基础 .....</b>	<b>37</b>
3.1 面向对象程序设计语言 .....	37
3.1.1 程序 .....	37
3.1.2 语言及其分类 .....	37
3.1.3 C++是面向对象程序设计语言 .....	38
3.2 C++语言基础 .....	38
3.2.1 C++的发展史 .....	38
3.2.2 C++基础语法 .....	39
3.2.3 常数 .....	41
3.2.4 输入/输出 .....	44
3.2.5 表达式和语句 .....	45
3.2.6 函数 .....	50
3.2.7 函数的重载 .....	52
3.2.8 C++中的类 .....	52
3.2.9 写 C++类代码的相关基础常识 .....	60
3.2.10 继承 .....	72
3.2.11 运算符重载 .....	87
3.2.12 虚函数 .....	95
3.2.13 模板 .....	99
3.3 Java 的基础知识 .....	102
3.3.1 Java 的发展史 .....	102
3.3.2 Java 的基本语法 .....	102
3.3.3 运算符 .....	105
3.3.4 表达式 .....	108
3.3.5 流程控制语句 .....	110
3.3.6 数组和字符串 .....	110
3.3.7 Java 环境的一些特征 .....	112
习题三 .....	113

<b>第 4 章 发现对象及其属性、行为定义</b> .....	117
4.1 怎样发现对象 .....	117
4.1.1 从问题领域发现对象 .....	117
4.1.2 从系统责任发现对象 .....	118
4.1.3 对象的审查和筛选 .....	118
4.2 定义属性 .....	121
4.2.1 发现对象的属性 .....	121
4.2.2 按一般常识确定属性 .....	121
4.2.3 本问题领域中的属性 .....	121
4.2.4 结合系统软件 and 将开发的软件确定属性 .....	121
4.2.5 与其他对象相联系的属性 .....	121
4.3 确定行为 .....	122
4.3.1 对象的状态和状态转移图 .....	122
4.3.2 根据一般常识确定行为 .....	124
4.3.3 根据状态转移图确定行为 .....	125
4.4 C++编程 .....	125
4.5 Java 编程 .....	127
*4.6 主动对象识别及其程序设计 .....	129
4.6.1 主动对象识别 .....	129
4.6.2 主动对象程序设计基础 .....	131
4.6.3 主动对象程序设计 .....	133
习题四 .....	137
<b>第 5 章 对象系统设计步骤、对象关系及其实现</b> .....	139
5.1 面向对象程序设计步骤 .....	139
5.1.1 构造一个解题模型 .....	140
5.1.2 标识对象 .....	140
5.1.3 标识对象间的关系 .....	141
5.1.4 建立对象的型构 .....	141
5.1.5 实现各对象 .....	141
5.2 对象之间、类之间的关系 .....	142
5.2.1 IS-A 关系 .....	142
5.2.2 HAS-A 关系 .....	144
5.2.3 USES-A 关系 .....	146
5.2.4 IS-LIKE-A 关系 .....	146
5.2.5 CREATES-A 关系 .....	146
5.3 面向对象设计和开发示例 .....	147
5.3.1 构造一个解题模型 .....	147
5.3.2 标识对象 .....	147
5.3.3 标识对象间的关系 .....	148

5.3.4 建立对象的型构 .....	149
5.3.5 实现各对象 .....	152
习题五 .....	166
<b>*第 6 章 C++标准模块库(STL)及其程序设计 .....</b>	<b>167</b>
6.1 STL 简介 .....	167
6.1.1 初识 STL .....	167
6.1.2 STL 和 HP 公司 .....	167
6.1.3 大众化的 STL .....	167
6.1.4 STL 总览 .....	168
6.1.5 STL 基本组件 .....	168
6.1.6 其他 STL 组件 .....	168
6.1.7 完整的 STL 程序包 .....	168
6.2 vectors .....	169
6.2.1 vector 程序实例 .....	169
6.2.2 初始化 .....	171
6.2.3 vector 容器的方法 .....	172
6.2.4 对 vector 可执行的其他操作 .....	173
6.3 STL 与模板 .....	173
6.3.1 template 关键字 .....	174
6.3.2 模板语法 .....	174
6.3.3 模板函数 .....	175
6.3.4 模板类 .....	175
6.3.5 STL 与模板的比较 .....	176
6.4 迭代器 .....	176
6.4.1 迭代器的使用 .....	177
6.4.2 迭代器类型 .....	180
6.5 算法 .....	182
6.5.1 算法的定义 .....	182
6.5.2 算法的使用 .....	184
6.5.3 泛型算法的使用 .....	193
6.5.4 函数对象 .....	193
6.6 容器 .....	198
6.6.1 容器的定义 .....	198
6.6.2 适配器的定义 .....	223
6.7 总结 .....	225
习题六 .....	226
<b>*第 7 章 标准建模语言及其应用 .....</b>	<b>227</b>
7.1 标准建模语言(UML)概述 .....	227
7.1.1 UML 语言的特征 .....	228

7.1.2 UML 语言的目标 .....	228
7.1.3 UML 语言 .....	228
7.2 UML 的模型、视图、图与系统架构建模 .....	229
7.2.1 视图 .....	229
7.2.2 图 .....	230
7.2.3 UML 中的模型元素 .....	230
7.3 用例建模 .....	233
7.3.1 用例图 .....	233
7.3.2 确定行为者 .....	234
7.3.3 确定用例 .....	235
7.3.4 用例之间的关系 .....	236
7.4 类和对象建模 .....	236
7.4.1 类图和对象图 .....	236
7.4.2 确定类 .....	237
7.4.3 UML 中类之间的关系 .....	239
7.4.4 包 .....	244
7.4.5 模板 .....	244
7.4.6 一个类图的实例 .....	245
7.5 动态建模 .....	246
7.5.1 消息 .....	246
7.5.2 状态图 .....	246
7.5.3 时序图 .....	253
7.5.4 协作图 .....	254
7.5.5 活动图 .....	256
7.6 物理体系结构建模 .....	258
7.6.1 逻辑体系结构和物理体系结构 .....	258
7.6.2 构件图 .....	260
7.6.3 部署图 .....	261
7.7 使用 UML 的过程 .....	263
7.7.1 UML 过程的基础 .....	263
7.7.2 面向对象方法的一般过程 .....	264
7.7.3 Rational Objectory 过程 .....	266
习题七 .....	267
参考文献 .....	268



# 第1章 绪 论

面向对象的程序设计就是一种软件构造的思想，它的重要意义在于用比较接近人类自然的思维去分析问题、解决问题。本章将集中介绍面向对象方法的基本原理与概念，并简要介绍它的发展历史与现状。本章的目的是使读者在学习面向对象的分析与设计之前，首先了解面向对象方法的主要内容，并从理论上掌握其基本知识，作为进一步学习的基础。

本书中面向对象的程序设计所依托的语言主要是 C++，有时为便于理解和与各种语言相互比较，也使用了 Java。C++和 Java 语言怎么去表现面向对象思想将在第 4、5 章介绍。没有学习过 C++和 Java 语言的人，可以仅在思维形态上去理解。本章和第 2 章涉及到的相关程序及其概念在无法理解时，可以先跳过去。

## 1.1 面向对象简述

由于面向对象方法已经发展到计算机科学的许多领域，因此，若想从一般意义上给出“面向对象方法”严格的、清晰的定义，是一件很困难的事情。随着科学技术的发展，人们在不同时期对某一事物的认识也不同。比如，我们在 20 世纪 80 年代初期以前去讨论这个问题，则可以说，面向对象是一种新兴的程序设计方法，或者说是一种新的程序设计范型(Paradigm)，其基本思想是使用对象、类、继承、封装、消息等基本概念进行程序设计。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，即使我们在“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在向计算机软件以外的一些领域发展，例如计算机体系结构和人工智能等。

### 1.1.1 对象定义

#### 1. 对象的一般定义

从一般意义上来讲，现实世界中的任何事物都可以称为对象，但面向对象分析(Object-Oriented Analysis, OOA)只注意那些与问题域和责任有关的对象。通过分析、认识这些对象，抽象出它们的主要特征，并在软件系统中用程序来表示这些对象。从这个意义上讲，对象的定义是：

对象是系统中用来描述客观事物的一个实体，是构成系统的一个基本单位，由一组属性和对这些属性进行操作的一组服务构成。

人类在认识现实世界的思维活动时并不是逐个地认识和描述每一个对象个体的，而是通过抽象，把具有共同特征的对象归结为一类，形成一般概念。

面向对象方法是基于客观世界，并使对象思想模型化的软件开发方法。在面向对象程



序设计中,所谓对象,是指一个属性(数据)集及其操作(行为)的封装体。作为计算机模拟真实世界的抽象,一个对象就是问题域、一个物理的实体或逻辑的实体。在计算机程序中,可将对象视为一个“基本程序模块”,因为它包含了数据结构和对象所提供的相关操作功能。关于对象中的“方法”、“服务”、“行为”、“状态”等名词,其主要概念是:

对象的属性是指描述对象的数据。这些数据可以是系统或用户定义的数据类型,也可以是一个对象的数据类型。对象的属性值的集合称为对象的状态(State)。

对象的行为是定义在对象属性上的一组操作方法(Method)的集合。方法是响应消息而完成的算法,表示对象内部实现的细节,对象方法集合体现了对象的行为能力。对象的属性和行为是对象定义的组成要素,它们统称为对象的特征。无论对象是有形的还是抽象的,简单的还是复杂的,一般都具有以下特征:

- (1) 具有一个状态,由与其相关联的属性值集合表征;
- (2) 具有惟一标示名,可以区别于其他对象;
- (3) 有一组操作方法,每个操作决定对象的一种行为;
- (4) 对象状态只能被自身的行为所改变;
- (5) 对象的操作包括自操作(施于自身)和它操作(施于其他对象);
- (6) 对象之间以消息传递的方式进行通信;
- (7) 一个对象的成员仍然可以是一个对象。

在上面列举的特征中,严格地讲,前3条是对象的基本特征,后4条是特征的进一步定义说明。在面向对象系统中,人们并不是逐个描述各个具体的对象,而是将注意力集中于具有相同特征的一类对象,并抽象出一类对象的共同结构和行为,进行一般描述,从而避免数据冗余。

## 2. 对象的广义定义

在不同的方面,对象定义是不同的。在韦氏字典中是这样来解释对象的: Object, something mental and physical which thought(人认知的某些物质、精神的事物)。照此定义,对象既可以是人和物,也可以是其直接进行活动或思想的“存在的东西”。可见或可以触摸的任何事物(例如一种产品或物质)被理解为对象是容易接受的,但要注意的是,活动或思想的“存在的东西”被理解为对象常常被人们忽视。

### 1.1.2 面向对象

客观世界的问题都是由客观世界的实体及其相互之间的联系构成的。我们把客观世界的实体称为问题对象。对象都有自己的运动状态及运动规律,不同对象之间的相互作用和相互通信构成了完整的客观世界。

#### 1. 人类从世界的本源入手去认识、描述对象

从认识论的角度看,整个软件开发过程可以归结为两项主要活动:①人们对所要解决的问题及其相关的事物的认识;②基于这种认识所进行的描述。软件开发也蕴涵有人类认识、描述世界,分析问题,解决问题的思想。

所谓“认识”,是指在系统所要处理的问题域范围内,通过人的思维对该问题域内客观存在的事物以及所要解决的问题产生正确的认识和理解,包括弄清事物的属性、行为以及



彼此之间的关系，并找出解决问题的方法。

所谓“描述”，是指用一种语言把人们对问题域范围中事物的认识和对问题及其解决方法的认识表现出来。描述的方式有许多，其中从问题域中对象的固有特征进行描述的方式，使对问题域的描述的准确性与世界的本源更加接近。最终的描述必须使用一种被机器读得懂的语言，即编程语言。

大致地划分认识与描述，可以把分析和设计看作对问题及其解决方案的认识，把编程看作对解决方案的描述。细致地划分，则分析和设计阶段本身也包括描述，即按一定的软件开发模型中分析及设计阶段的表示方法，产生分析文档和设计文档；编程也包括一定的认识和理解活动，特别是在传统的软件开发方法中，分析文档和设计文档不能够很好地映射问题域，程序员在书写程序之前，往往需要在分析、设计文档的帮助下，对程序所要描述的事物进行再认识。

在软件工程学出现以前，“软件开发”几乎就是指“编程”。但即使在那个时期，编程工作实际上也包括认识事物和描述事物两项内容：首先要将所要解决的问题在头脑中想清楚，即产生正确的认识，然后再动手编写程序，即产生正确的描述。对问题域的正确认识是软件开发工作的首要任务。没有对问题域的正确认识，就不可能产生一个正确的系统。描述只是把开发人员对问题域的认识表达出来，最终产生一个机器能够理解和执行的系统实现。

## 2. 面向对象(Object-Oriented)

“面向对象”是认识论和方法学的一个基本原则。面向对象分析的目的是完成对问题空间的分析并建立系统模型，具体任务是确定和描述系统中的对象、对象的静态特征和动态特征、对象间的关系以及对象的行为约束等。人对客观世界的认识和判断常常采用由一般到特殊(演绎法)和由特殊到一般(归纳法)两种方法，这实际上是对需要认识判断的问题域对象进行分解和归类的过程。面向对象有两方面的含义：① 问题域中的客观对象总是被以自然思维的方式来认识、描述；② 认识、描述这些客观对象总是以预先归纳好的一定标准特征模式进行。

基于上述面向对象思想的软件工程方法分为三个步骤：面向对象分析(Object-Oriented Analysis, OOA)、面向对象设计(Object-Oriented Design, OOD)、面向对象编程(Object-Oriented Programming, OOP)。本书尽可能将这三部分内容融合为一个整体进行介绍，以使读者有一个完整认识。一般而言，OOA 的内容偏重于“认识”，OOD 则侧重于对认识清楚的对象进行“细致的描述”。两者研究的对象是基本相同的，其主要区别是后者较前者更加细致。本书侧重从 OOA 出发进行讨论。

## 3. 面向对象分析

OOA 是面向对象软件工程方法的第一环节，包括概念原则、过程步骤、表示方法、提交文档等规范要求。面向对象分析主要遵循以下几个原则：

### 1) 构造和分解相结合的原则

构造是指由基本对象组装复杂对象的过程；分解是指对大粒度对象进行精化从而完成系统模型的细化的过程，这两者的结合是软件工程中系统分析的基本方法。



## 2) 抽象化和具体化相结合的原则

抽象强调实体的本质和内在的属性，而忽视与问题无关的属性，是在决定如何实现对象之前确定对象的意义和行为。使用抽象可以尽可能避免过早考虑一些细节数据。抽象将数据对象及作用在其上的操作抽象成对象，它是分析的核心和灵魂，也是组织和建立系统规格说明和设计说明的基础。

具体化是指在精化过程中，对对象的必要细节加以刻画，这有助于确定系统对象，加强系统模型的稳定性。

抽象化和具体化使具体对象可以直接从抽象对象的定义中获得已经有的性质和特征，而不必重复定义它们。在分析中只需一次性地指定公共属性和操作，然后具体化并且扩充这些属性及其操作。

## 3) 封装的原则

封装是指将对象的各种独立的外部特征与内部实现细节分开。从外部只需知道它做什么，而不必知道它如何去做，也不必知道其内部数据。对象接口定义要尽可能地与内部工作状态相分离。封装有助于最大限度地减少由于需求的改变而对整个系统所造成的影响。

## 4) 相关的原则

相关是指系统中的对象之间存在着各种关联。在分析时要考虑与问题相关的对象间的各种关联，这些关联是对象协作的基础。相关包括静态结构的关联，比如整体和部分；也包括动态特征的关联，比如消息传递等。

## 5) 行为约束的原则

对象的语义特征是通过行为约束刻画的。行为约束包括静态行为约束和动态行为约束，它表示对象合法存在和对象的操作合法执行应该满足的约束条件。行为约束有助于深刻地理解对象和系统。

面向对象分析的主要内容包括静态结构分析和动态行为分析。下面来看看他们各自的表述以及联系。

### • 静态结构分析

静态结构描述对象、类以及类间的相互关系。静态结构首先要确定对象和对象类。

对象是一种具有简明界面及应用意义的实体的抽象。所有对象都是惟一标识和可以区分的。

对象类描述的是一组具有共同属性、操作和语义特征的对象。属性是类中对象的性质，操作是类中对象的行为。

在确定了对象和对象类的基础上，还要确定对象类之间的关系。对象类之间的关系主要有以下几种：

(1) 一般化与特殊化关系。一般化与特殊化是面向对象分析中的基本概念。在分析过程中，我们可以这样说：如果类 A 具有类 B 的全部服务，而且具有自己特有的某些属性或服务，则类 A 叫做类 B 的特殊类，类 B 叫做类 A 的一般类。根据类的共性以及个性将类组织成不同层次，子类通过继承机制来获得父类的属性和操作。在 OOA 模型中建立一般化与特殊化模型，是为了使系统模型更加清晰地映射问题域中事物的分类关系。

(2) 聚合关系。聚合关系是对象之间的组合构造关系，根据这种关系可以将对象组织成不同层次。高层次的对象是容器对象，称为聚合对象；低层次的对象是内含对象，称为组





成对象。聚合对象通过组成对象的操作来实现自身的操作。

(3) 关联关系。关联关系表达了对象之间的引用和消息传递等关系。关联有其多重性,包括一对一、一对多和多对多等。

- 动态行为分析

动态行为描述了系统中对象的合法状态序列。对象的动态行为通常用动态模型来表示,它包括两方面的内容:一是单个对象自身的生命周期演化;一是整个系统中对象间的消息传递和协同工作。

对象生命周期演化主要包括三个组成部分:一是对象在生命周期中的可能状态;二是对象发生状态转换时要执行的动作;三是导致对象从一个状态到另外一个状态的转换的事件,事件是控制状态转换的条件。这三部分一起构成了对象生命周期中的状态转换图,用于描述对象内部的动态行为。

对象的动态行为可以通过关系由其子对象继承,子对象继承祖先的状态和转换,同时子对象状态图也是父对象状态图的细化。

在对象系统中,对象之间通过消息传递来协同工作。对应于系统的每一项任务,都有一组消息传递以及动作来完成该任务。每个对象都要完成一组任务,而每个任务都有一组事件序列与之对应,因此,系统中对象协作的动态性质可以由一组事件序列来描述。动态行为与静态结构密切相关并且受其约束。状态是对象属性和关联的具体取值,而静态结构限制了对象状态的取值范围。对象状态的变化是通过对象的操作来完成的。动态行为并行性也与静态结构相关。系统中的对象操作可以是并发的;在聚合关系中,聚合对象的状态可以是各组成对象状态的组合,而各组成对象有其自身的独立状态,这些状态的变化可以是并发的;另外,同一对象的各个动作之间也可以是并发的。

#### 4. OOA 与 OOD

OOA 的任务是采用面向对象方法,把对问题论域和系统的认识理解,正确地抽象为规范的对象(包括类和继承层次)和消息传递联系,形成面向对象模型,为后续的面向对象设计和面向对象编程提供指导。而且,OOA 与 OOD 能够自然地过渡和结合,也是面向对象方法值得称道的一个优点。

#### 5. 基于对象

很多人没有对“面向对象”和“基于对象”(Object-Based)两个不同的概念加以区分。一般说来,面向对象的三大特征(封装,继承,多态)缺一不可。“面向对象”和“基于对象”都实现了“封装”的概念,但是面向对象实现了“继承和多态”,而“基于对象”没有实现“继承和多态”。“面向对象”和“基于对象”的中文含义,常常容易被混淆。

### 1.1.3 面向对象范型

在软件设计和实现中,传统的被人们广泛使用的方法是面向过程的程序设计方法。近几年来,在计算机软件业,面向对象程序设计的思想已经被越来越多的软件设计人员所接受。面向对象程序设计是建立在面向对象范型基础之上的,下面先来了解一下相关的概念。