

WUTP



普通高等学校计算机科学与技术专业新编系列教材

Object Oriented Programming and Methodology

面向对象程序设计与方法



李陶深 赵文静 主编

```
#include <stdio.h>
#include <stdio.h>
void main()
void main()
{
    void swap(int * ptr1,int * ptr2);
    void swap(int * ptr1,int * ptr2);
    int x,y, * ptr1, * ptr2;
    int x,y, * ptr1, * ptr2;
    printf("input x,y:");scanf("%d,%d",&x
    printf("input x,y:");scanf("%d,%d",&x
    printf("%d\t%d\n",x,y);ptr1=&x;ptr2=
    printf("%d\t%d\n",x,y);ptr1=&x;ptr2=
    if(x<y)
    if(x<y)
    swap(ptr1,ptr2);
    swap(ptr1,ptr2);
    printf("%d\t%d\n",x,y);
    printf("%d\t%d\n",x,y);
}
void swap(int * ptr1,int * ptr2)
void swap(int * ptr1,int * ptr2)
```

武汉理工大学出版社

Wuhan University of Technology Press



普通高等学校计算机科学与技术专业新编系列教材

Object Oriented
Programming and Methodology

面向对象程序设计与方法

李陶深 赵文静 主编

武汉理工大学出版社

Wuhan University of Technology Press

MJS83/01

内 容 提 要

本书以面向对象程序设计方法贯穿始终,阐述了面向对象的程序设计思想和方法,系统地介绍了 C++ 语言的特征、基本知识和编程方法。全书共分 12 章,前 5 章是面向对象方法学部分,主要介绍面向对象程序设计的基本概念与基本思想、统一建模语言(UML)、面向对象分析与设计方法、面向对象编程与测试;第 6 章至第 12 章是面向对象程序设计语言部分,主要介绍 C++ 语言的基本概念、程序结构、面向对象程序编程方法、异常与出错处理、Windows 应用程序开发,并通过大量简洁易懂的实例,生动而详尽地介绍了 C++ 语言中许多关键的概念。

本书可作为大专院校计算机科学与技术专业的本科生教材,也可作为面向对象程序设计初学者和专业软件设计人员进一步提高编程能力的自学教材和参考书。

图书在版编目(CIP)数据

面向对象程序设计与方法/李陶深,赵文静主编. —武汉:武汉理工大学出版社,2003.7

普通高等学校计算机科学与技术专业(本科)新编系列教材

ISBN 7-5629-1950-X

I. 面… II. ①李… ②赵… 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(2002)第 106858 号

出版发行:武汉理工大学出版社(武汉市武昌珞狮路 122 号 邮编:430070)

HTTP://cbs.whut.edu.cn

E-mail: wutp02@163.com wutp@public.wh.hb.cn

经 销 者:各地新华书店

印 刷 者:武汉理工大印刷厂

开 本:787×960 1/16

印 张:19.75

字 数:387 千字

版 次:2003 年 8 月第 1 版

印 次:2003 年 8 月第 1 次印刷

印 数:1—5000 册

定 价:27.00 元

凡购本书,如有缺页、倒页、脱页等印装质量问题,请向出版社发行部调换。本社购书热线电话:(027)87397097 87394412

普 通 高 等 学 校
计算机科学与技术专业新编系列教材
编 审 委 员 会

顾问:

卢锡城 周祖德 何炎祥 卢正鼎 曾建潮
熊前兴

主任委员:

严新平 钟 珞 雷绍锋

副主任委员:

李陶深 鞠时光 段隆振 王忠勇 胡学钢
李仁发 张常年 郑玉美 程学先 张翠芳
孙成林

委员:(以姓氏笔画为序)

王 浩	王景中	刘任任	江定汉	朱 勇
宋中山	汤 惟	李长河	李临生	李跃新
李腊元	李朝纯	肖俊武	邱桃荣	张江陵
张继福	张端金	张增芳	陈和平	陈祖爵
邵平凡	金 聪	杨开英	赵文静	赵跃华
周双娥	周经野	钟 诚	姚振坚	徐东平
黄求根	郭庆平	郭 骏	袁 捷	龚自康
崔尚森	蒋天发	詹永照	蔡启先	蔡瑞英
谭同德	熊盛武	薛胜军		

秘书长:田道全

总责任编辑:段 超 徐秋林

出版说明

当今世界已经跨入了信息时代,计算机科学与技术正在迅猛发展。尤其是以计算机为核心的信息技术正在改变整个社会的生产方式、生活方式和学习方式,推动整个人类社会进入信息化社会。为了顺应时代潮流,适应计算机专业调整及深化教学改革的要求,充分考虑到不同层次高校的教学现状,满足广大高校的教学需求,武汉理工大学出版社经过广泛调研,与国内近 30 所高等院校的计算机专家进行探讨,决定组织编写“普通高等学校计算机科学与技术专业新编系列教材”。

我们在组织编写新编本套系列教材时,以培养现代化高级人才为重任,以提高学生综合素质、培养学生应用能力和创新能力为目的,以面向现代化、面向世界、面向未来为准绳,注重系列教材的特色和实用性,反映最新的教学与科研成果,体现本专业的时代特征。同时,面对教育改革的需要、人才的需要和社会的需要,在编写本教材时,借鉴、学习国外一流大学的先进教学体系,结合国内的实际需要,吸取具有先进性、实用性和权威性的国外教材的精华,以更好地促进国内教材改革顺利进行。从时代和国际竞争要求的高度来思考,为打造一套高起点、高水平、高质量的系列教材而努力。

本套教材具有以下特色:

与时俱进,内容科学先进——充分体现计算机学科知识更新快的特点,及时更新知识,确保教材处于学科前沿,以拓宽学生知识面,培养学生的创新能力。

紧跟教学改革步伐,体现教学改革的阶段性成果——符合全国高校计算机专业教学指导委员会、中国计算机学会教育委员会制订的“计算机学科教学计划 2000”的内容要求。

实现立体化出版,适应教育方式的变革——本套教材努力使用和推广现代化的教学手段,凡有条件的课程都准备组织编写、制作和出版配合教材使用的实验、习题、课件、电子教案及相应的程序设计素材库。

本套教材首批 25 种预定在 2003 年秋季全部出齐。我们的编审者、出版者决不敢稍有懈怠,一定高度重视,兢兢业业,按最高的质量标准工作。教材建设是我们共同的事业和追求,也是我们共同的责任和义务,我们诚恳地希望大家积极选用本套教材,并在使用过程中给我们多提意见和建议,以便我们不断修订、完善全套教材。

武汉理工大学出版社

2002 年 10 月

前 言

面向对象方法(简称 OO 方法)是一种把面向对象的思想应用于软件开发过程中,指导开发活动的系统方法,它所追求的目标是尽可能地运用人类的自然思维方式来建立问题空间的模型,构造尽可能直观、自然地表达求解方法的软件系统。运用面向对象方法开发的软件系统易于维护,支持软件的复用,其体系结构易于理解、扩充和修改。面向对象方法的出现很快受到了计算机软件界的青睐,成为了目前软件开发的主流开发方法。

面向对象程序设计以对象为核心,利用类的概念和继承等机制,设计类以及由类构造程序的方法与过程,用计算机对象模拟现实世界对象。面向对象程序设计的最大优点是软件具有可重用性、可维护性和可管理性。

C++是由美国 AT&T 公司的 Bell 实验室最先设计和实现的一种应用比较广泛的面向对象程序设计语言,它不仅继承了 C 语言的全部优点和功能,而且具有支持面向对象的机制、保持与 C 语言兼容、程序可读性好、代码结构合理、代码可重用性和可扩充性好、程序比较容易维护等特点,适用于大中型软件的开发。C++已经成为目前最受欢迎的面向对象程序设计语言之一。

近年来,国内外许多高校都把面向对象方法和面向对象程序设计作为必修课或选修课,列入了计算机相关专业的教学计划。鉴于面向对象程序设计技术已经逐渐成为软件开发的主流趋势,本书以面向对象程序设计方法贯穿始终,阐述了面向对象程序设计的基本思想和方法,系统地介绍了使用 C++ 语言进行面向对象程序设计的基本原理和编程方法。

本书的内容由两大部分组成:前 5 章是面向对象方法学部分,主要介绍面向对象方法的基本概念与基本建模思想、统一建模语言(UML)、面向对象分析与设计方法、面向对象实现与测试;第 6 章至第 12 章是面向对象程序设计部分,包括 C++语言的基本概念、基本语法、面向对象程序编程方法、C++中的输入输出处理、异常与出错处理、Windows 应用程序开发等内容。全书理论与实际相结合,循序渐进,通俗易懂,通过大量简洁易懂的实例,生动而详尽地讨论了面向对象技术中类与对象、封装、抽象、继承与派生类、多态性、虚函数等特征,以及这些特征在面向对象程序设计中的具体应用。通过本书的系统学习,读者可以建立起面向对象程序设计的新思维,比较全面地掌握利用 C++语言进行面向对象程序设计的基本技能和编程要领,初步具备利用面向对象技术分析和解决实际问题的能力。

全书的编写由李陶深、赵文静、胡燕、霍亮、谌静、梅传军合作完成,是集体智慧的结晶。其中,本书的第6、9、10、11章由赵文静编写,第7章由胡燕、霍亮编写,第8章由胡燕、谌静编写,第12章的12.1、12.2节由梅传军撰写,第1、2、3、4、5章和第12章的12.3、12.4节由李陶深编写。全书由李陶深统一定稿。

本书的完成得到了广西大学、西安建筑科技大学、武汉理工大学、湖北工学院的有关教师和武汉理工大学出版社的大力支持。广西大学的陈国宁老师认真检查和测试了书中的程序和实例,为本书提出了许多中肯的修改意见。作者一并致以衷心的感谢。

由于作者的水平有限,书中难免有疏漏、欠妥、错误之处,敬请读者批评指正。

编 者

2003年4月



目 录

第一篇 面向对象方法学

1 面向对象方法的基本概念	(2)
1.1 面向对象的基本概念	(2)
1.1.1 面向对象的含义	(2)
1.1.2 对象	(3)
1.1.3 类	(4)
1.1.4 属性	(5)
1.1.5 操作、方法和服务	(5)
1.1.6 消息	(6)
1.1.7 封装	(7)
1.1.8 继承	(8)
1.1.9 多态性	(10)
1.2 面向对象方法概述	(11)
1.2.1 从结构化方法到面向对象方法	(11)
1.2.2 面向对象方法的主要内容	(13)
1.2.3 面向对象方法的优点	(16)
1.3 面向对象的建模方法	(16)
习题与思考题	(18)
2 面向对象建模	(19)
2.1 对象建模	(19)
2.1.1 对象图	(20)
2.1.2 对象符号	(20)
2.1.3 对象模型的表示	(21)
2.1.4 对象模型的例子	(23)

2.2 动态模型	(24)
2.2.1 事件与状态	(24)
2.2.2 状态图	(25)
2.2.3 行为	(25)
2.3 功能模型	(26)
2.3.1 数据流图	(26)
2.3.2 操作与约束	(28)
2.4 对象模型、动态模型、功能模型之间的关系	(29)
2.4.1 对象模型与动态模型之间的关系	(30)
2.4.2 对象模型与功能模型之间的关系	(30)
2.4.3 动态模型与功能模型之间的关系	(30)
习题与思考题	(31)
3 统一建模语言(UML)	(33)
3.1 统一建模语言(UML)简介	(33)
3.1.1 UML 的概貌	(33)
3.1.2 UML 语言的特点	(34)
3.1.3 UML 的建模步骤	(35)
3.2 UML 的组成	(35)
3.2.1 UML 语义	(35)
3.2.2 UML 的表示方法	(35)
3.2.3 UML 语言的公共机制	(36)
3.2.4 UML 语言的图符表示	(37)
3.2.5 UML 语言的基本视图	(38)
3.3 UML 的静态建模机制	(40)
3.3.1 建立用例图	(40)
3.3.2 类图和对象图的表示	(42)
3.3.3 包图	(44)
3.4 UML 的动态建模机制	(46)
3.4.1 顺序图	(46)
3.4.2 协作图	(47)
3.4.3 状态图	(48)
3.4.4 活动图	(49)
3.5 系统实现的建模机制	(51)
3.5.1 构件图	(51)

3.5.2 配置图	(52)
习题与思考题	(55)
4 面向对象的分析与设计	(56)
4.1 面向对象分析	(56)
4.1.1 面向对象分析的活动与过程	(57)
4.1.2 需求陈述	(59)
4.1.3 面向对象分析过程	(61)
4.2 面向对象的设计	(70)
4.2.1 面向对象设计概述	(70)
4.2.2 系统设计	(71)
4.2.3 对象设计	(76)
4.2.4 面向对象设计的一个实例	(77)
习题与思考题	(81)
5 面向对象实现与测试	(82)
5.1 面向对象设计的实现	(82)
5.1.1 面向对象程序设计的准则	(82)
5.1.2 面向对象语言的主要特性	(85)
5.1.3 面向对象程序语言概述	(86)
5.2 软件测试	(88)
5.2.1 软件测试的基本概念	(88)
5.2.2 软件测试方法	(90)
5.2.3 软件测试的策略	(95)
5.3 面向对象软件的测试	(100)
5.3.1 面向对象的单元测试	(100)
5.3.2 面向对象的集成测试	(100)
5.3.3 面向对象的确认测试	(101)
5.4 面向对象的测试用例设计	(101)
5.4.1 测试类的方法	(101)
5.4.2 基于场景的测试设计	(103)
习题与思考题	(105)

第二篇 面向对象程序设计

6 面向对象程序设计语言概述	(108)
6.1 程序设计的基本概念	(108)
6.1.1 面向过程程序设计的基本概念	(109)
6.1.2 面向对象程序设计的基本概念	(110)
6.2 面向对象程序设计方法的优点	(110)
6.2.1 可重用性	(110)
6.2.2 可扩充性和可维护性	(111)
6.2.3 可管理性	(111)
6.3 C++语言概述	(111)
6.3.1 C++的发展历史	(111)
6.3.2 C++程序的结构与特点	(113)
6.3.3 C++程序的开发过程	(114)
6.4 其他面向对象程序设计语言简介	(115)
习题与思考题	(117)
7 C++语言基础	(118)
7.1 C++程序的结构	(118)
7.1.1 C++程序的构成	(118)
7.1.2 程序各对象之间的关系	(119)
7.1.3 C++程序的结构分析	(120)
7.2 C++的控制流	(121)
7.2.1 C++的选择结构	(121)
7.2.2 循环控制结构	(123)
7.3 C++在非面向对象方面的一些特性	(125)
7.3.1 注释行	(125)
7.3.2 新的 I/O 流	(125)
7.3.3 new 和 delete	(127)
7.3.4 灵活的局部变量说明	(129)
7.3.5 作用域运算符——::	(129)

7.3.6	const 修饰符	(130)
7.3.7	引用	(135)
7.3.8	结构、联合和枚举名都是类型名	(139)
7.3.9	无名联合	(139)
7.3.10	显式类型转换	(140)
	习题与思考题	(142)
8	函数、类与对象	(144)
8.1	C++ 的函数	(144)
8.1.1	函数原型	(144)
8.1.2	内联函数	(146)
8.1.3	带有缺省参数的函数	(148)
8.1.4	函数重载	(150)
8.1.5	指向函数的指针	(153)
8.2	定义类	(154)
8.2.1	类的定义	(154)
8.2.2	类的实现	(155)
8.2.3	类的作用域	(156)
8.2.4	创建类的实例对象	(157)
8.2.5	结构和类	(157)
8.2.6	联合和类	(159)
8.3	构造函数与析构函数	(160)
8.3.1	构造函数	(160)
8.3.2	析构函数	(162)
8.3.3	重载构造函数	(163)
8.3.4	拷贝构造函数	(164)
8.4	对象数组与对象指针	(166)
8.4.1	对象数组	(166)
8.4.2	对象指针	(168)
8.4.3	this 指针	(171)
8.5	静态成员	(173)
8.5.1	静态数据成员	(173)
8.5.2	静态成员函数	(175)
8.6	友元	(176)
8.6.1	友元函数	(176)

8.6.2 友元成员	(177)
8.6.3 友元类	(178)
习题与思考题	(179)
9 继承与派生、多态性	(182)
9.1 派生类	(182)
9.1.1 派生类的定义	(182)
9.1.2 保护成员的作用	(184)
9.1.3 派生类的构造函数与析构函数	(185)
9.2 继承	(188)
9.2.1 方法的继承	(188)
9.2.2 数据类型转换	(191)
9.2.3 隐藏基类成员	(192)
9.2.4 抽象类	(194)
9.2.5 多重继承	(194)
9.3 多态性与重载	(197)
9.3.1 多态性	(197)
9.3.2 函数重载	(197)
9.3.3 运算符重载	(198)
9.3.4 虚拟函数	(201)
9.3.5 纯虚拟函数	(202)
9.4 应用举例	(203)
习题与思考题	(211)
10 C++中的输入/输出操作	(213)
10.1 C++的流	(213)
10.1.1 C++的预定义流	(213)
10.1.2 流的插入与提取	(214)
10.2 C++的流类库	(215)
10.3 格式化 I/O	(216)
10.3.1 ios 成员函数的格式化输入/输出控制	(216)
10.3.2 使用操作符进行输入输出的格式控制	(220)
10.3.3 用户自定义的输入输出	(222)
10.4 文件 I/O	(223)
10.4.1 文件的打开与关闭	(224)

10.4.2	文件的读/写	(226)
10.4.3	二进制 I/O	(227)
10.4.4	随机访问	(228)
10.4.5	I/O 状态	(230)
10.5	应用举例	(231)
	习题与思考题	(237)
11	预处理与异常处理	(238)
11.1	预处理	(238)
11.1.1	预处理标识符声明	(238)
11.1.2	预处理指令	(240)
11.1.3	条件预处理	(242)
11.2	出错与异常处理	(243)
11.2.1	编程中的错误	(244)
11.2.2	异常及异常对象	(245)
11.2.3	异常处理机制	(245)
11.2.4	异常的产生、捕获与处理	(246)
	习题与思考题	(248)
12	Windows 应用程序开发简述	(250)
12.1	Windows 应用程序的编程初步	(250)
12.1.1	一个 Windows 应用程序的组成	(251)
12.1.2	Visual C++ 6.0 开发环境的使用	(252)
12.1.3	创建一个简单的 Windows 应用程序	(253)
12.1.4	AppWizard 产生的源代码	(255)
12.2	Windows 应用程序	(256)
12.2.1	文档模板	(256)
12.2.2	图形用户接口(GUI)	(257)
12.2.3	对话框与控件编程	(260)
12.2.4	设备文本与图形处理	(261)
12.2.5	应用编程举例	(262)
12.3	Windows 的消息机制	(272)
12.3.1	消息与消息处理	(272)
12.3.2	消息的种类	(272)
12.3.3	发送消息与接受消息	(277)

12.3.4 MFC 的消息机制	(281)
12.4 多线程编程	(284)
12.4.1 线程的基本概念	(284)
12.4.2 线程之间的数据共享与同步	(285)
12.4.3 Visual C++ 中多线程的编程方法	(286)
12.4.4 多线程的编程举例	(295)
习题与思考题	(299)
参考文献	(300)

第一篇 面向对象方法学

面向对象方法学

面向对象方法学（Object-Oriented Programming, OOP）是一种编程方法学，其核心思想是将现实世界中的事物抽象为对象，对象之间通过消息传递进行交互。面向对象方法学的主要特点包括：封装性、继承性、多态性和抽象性。封装性是指将数据和方法封装在对象中，对外隐藏内部细节；继承性是指对象可以继承其他对象的属性和方法；多态性是指同一操作作用于不同对象时，可以有不同的解释；抽象性是指通过抽象类或接口来定义一组共同的行为。面向对象方法学广泛应用于软件开发中，特别是在大型复杂系统的开发中，能够有效提高开发效率和代码的可维护性。

面向对象方法学

面向对象方法学（Object-Oriented Programming, OOP）是一种编程方法学，其核心思想是将现实世界中的事物抽象为对象，对象之间通过消息传递进行交互。面向对象方法学的主要特点包括：封装性、继承性、多态性和抽象性。封装性是指将数据和方法封装在对象中，对外隐藏内部细节；继承性是指对象可以继承其他对象的属性和方法；多态性是指同一操作作用于不同对象时，可以有不同的解释；抽象性是指通过抽象类或接口来定义一组共同的行为。面向对象方法学广泛应用于软件开发中，特别是在大型复杂系统的开发中，能够有效提高开发效率和代码的可维护性。



1 面向对象方法的基本概念

本章提要

面向对象方法是当前最有效、最实用和流行的软件开发技术之一,是一种运用对象、类、继承、封装、聚合、消息传递、多态性等概念来构造系统的软件开发方法。本章首先着重讲述面向对象的基本概念和特征,然后介绍面向对象方法和面向对象软件开发的基本过程,以及面向对象的程序设计语言与建模方法。

1.1 面向对象的基本概念

面向对象(Object-Oriented,缩写为OO)方法是20世纪80年代流行起来的一种最有效、最实用的软件开发技术之一,是当前软件方法学的主要研究方向。面向对象的概念和思想由来已久,但是什么是面向对象的概念?面向对象应具备哪些基本特征?为什么一种方法被认为是面向对象的?关于这些问题有不同的观点和论述,目前还没有一个明确、共同的结论。在下面的讨论中,我们试图综合这些观点中最公共的部分。

1.1.1 面向对象的含义

在现实生活中,我们就生活在对象的世界里,这些对象存在于自然中、人造实体中、商业中以及我们使用的产品中,它们可以被分类、描述、组织、组合、操纵和创建。当我们利用计算机来解决客观世界的问题时,就是借助于某种设计方法和某种程序设计语言,创建、运行求解问题的计算机软件,对计算机中的实体施加某种处理,并用处理结果去映射问题的解。面向对象方法就是为计算机软