



普通高等教育“十一五”国家级规划教材



北京高等教育精品教材
BEIJING GAODENG JIAOYU JINGPIN JIAOCAI

C++面向对象

程序设计教程

(第4版)

陈维兴 林小茶 编著

非外借

清华大学出版社





普通高等教育“十一五”国家级规划教材



北京高等教育精品教材

BEIJING GAODENG JIAOYU JINGPIN JIAOCAI

C++面向对象 程序设计教程 (第4版)



陈维兴 林小茶 编著

清华大学出版社
北京

内 容 简 介

本书是为具有 C 语言基础的读者编写的,主要介绍 C++ 面向对象程序设计的基本知识和编程方法,全面讲述了 C++ 面向对象的基本特征。针对初学者的特点,本书力求通过大量的例子,以通俗易懂的语言讲解复杂的概念和方法,以帮助读者尽快迈入面向对象程序设计的大门。

本书主要内容包括类、对象、继承、派生类、多态性、虚函数、运算符重载、模板、输入和输出流类库、异常处理和命名空间、STL 标准模板库和面向对象程序设计方法与实例等。

第 4 版教材在第 3 版教材的基础上将全部程序在 Visual C++ 2010 调试环境下重新进行了调试,除了个别的程序,大部分程序可以在 Visual C++ 6.0 和 Visual C++ 2010 同时调试成功。

为了帮助读者进一步理解和掌握所学的知识,同时出版了与本书配套的辅导教材《C++ 面向对象程序设计教程(第 4 版)习题解答与上机指导》。

本书在编写时力求做到内容全面、语言通俗、例题丰富,同时配有大量习题,适合作为高等院校各专业学生学习 C++ 的基础教材,也适合初学者自学使用。

本书第 3 版被评为普通高等教育“十一五”国家级规划教材和北京高等教育精品教材,第 2 版被评为北京高等教育精品教材。

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 13701121933

图书在版编目(CIP)数据

C++ 面向对象程序设计教程 / 陈维兴, 林小茶编著. —4 版. —北京: 清华大学出版社, 2018
ISBN 978-7-302-50371-2

I. ①C… II. ①陈… ②林… III. ①C++ 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 118076 号

责任编辑: 柳 萍

封面设计: 何凤霞

责任校对: 王淑云

责任印制: 丛怀宇

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市铭诚印务有限公司

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 23.75

字 数: 575 千字

版 次: 2000 年 1 月第 1 版 2018 年 10 月第 4 版

印 次: 2018 年 10 月第 1 次印刷

定 价: 55.00 元

产品编号: 077007-01

第 4 版前言

本教材的前 3 版自出版以来获得读者和同行好评,成为不少高校的教材,并作为考研参考书受到读者欢迎。通过在教学工作中不断地进行总结和归纳,同时听取了专家和读者的意见后,我们决定在第 3 版的基础上对教材进行修订。

本教材是根据教学需要编写的,希望最大限度地做到定位准确、取材合适、深度适宜。目前,C++ 教材很多,但大多数都是为没有学过 C 语言的学生编写的。据作者了解,当前无论在大学里还是在社会中,都有相当一批人已经学过 C 语言。很多高校的培养计划,仍是先开设 C 语言程序设计,随后再开设 C++ 语言程序设计(必修或选修)。本教材就是为那些已经学过 C 语言,且具有一定程序设计基础的大学本科生编写的。因此,本教材是符合高校的教学需要的。在取材方面,舍去了 C 语言中的内容,只讲 C++ 面向对象程序设计部分的内容。这样既节省了教学时间,也减轻了学生的经济负担。

本教材将“以学生为中心”作为编写理念,内容叙述力求通俗易懂,由浅入深,符合认知规律,特别注意做到多讲实例,循序渐进地引出概念,尽量将复杂的概念用简洁、浅显的语言来讲述。力求教学内容富有启发性,便于学生学习。本教材还配有大量的例题、应用举例和习题,利于学生举一反三,从中学习程序设计方法和技巧,注重培养学生的创新能力。

这次修订保持了第 3 版语言通俗、层次清晰、理论与实例相结合的特点,使读者能尽快迈入面向对象程序设计的大门,迅速掌握 C++ 程序设计的基本技能和面向对象的概念和方法,并编写出具有良好风格的程序。本教材在第 3 版的基础上,在以下几个方面进行了修订:

(1) 考虑到一些学校已经采用 Visual C++ 2010 作为调试环境,因此,本书中所有程序重新在 Visual C++ 2010 中调试通过。当然,对于仍在使用 Visual C++ 6.0 作为调试环境的读者,大部分程序都能运行,而有些程序的第一行必须改为“`#include <iostream.h>`”程序才能通过调试,书中有相应的提示。

(2) 增加了“STL 标准模板库”一章,对于有一定程序设计基础的读者来说,掌握这部分内容可以提高编程效率,因为标准模板库中提供了对常用数据结构的操作,如表、栈和队列等。程序设计者在编写实用程序时可以直接调用在这些数据结构上操作的函数,而不用再编写相关的基础程序了。当然,作者也要强调,在学习过程中,数据结构的基础程序还是需要学习者好好研究并自己编写的,只是,在已经掌握了相关内容后,在编写实用程序的时候,可以考虑使用标准模板库。

(3) 删掉了一些不是十分必要的内容和案例,增加了一些新的、更实用的案例和内容,从而使本书更具有实用性。

C++ 是一门实践性很强的课程,只靠听课和看书是学不好的,必须多做题、多编程、多上机。我们编写了与本教材配套的《C++ 面向对象程序设计教程(第 4 版)习题解答与上机

指导》(由清华大学出版社同期出版),请读者参阅。

最后,向各位使用本教材的老师和读者表示衷心的感谢,欢迎您对本书的内容和编写方法提出批评和建议。

言 明 上 策

编 者

2018 年 3 月

第3版前言

面向对象程序设计是不同于传统程序设计的一种新的程序设计范型。它对降低软件的复杂性,改善其重用性和维护性,提高软件的生产效率,有着十分重要的意义。因此面向对象的程序设计被认为是程序设计方法学的一场实质性的革命。

C++ 语言是在 C 语言基础上扩充了面向对象机制而形成的一种面向对象程序设计语言,它除了继承了 C 语言的全部优点和功能外,还支持面向对象程序设计。C++ 是介绍面向对象程序设计的重要语言。学习 C++ 不仅可以深刻理解和领会面向对象程序设计的特点和风格,掌握其方法和要领,而且可以掌握一种十分流行和实用的程序设计语言。

许多高等院校将面向对象程序设计及面向对象技术正式列入教学计划,作为必修课或选修课。

本书第 1 版于 2000 年出版以来,颇受读者欢迎,不少高校用其作为教材或考研参考书,取得了很好的教学效果,第 2 版于 2004 年出版,被评为北京高等教育精品教材。在多年教学实践的基础上,作者听取了专家和读者的意见,并结合本人的教学经验,对原书作了认真的修改。

这次修订保持了原书语言通俗、层次清晰、理论与实例结合的特点,力求做到深入浅出,将复杂的概念用简洁浅显的语言来讲述。使读者尽快地迈入面向对象程序设计的大门,迅速掌握 C++ 程序设计的基本技能和面向对象的概念和方法,并能编写出具有良好风格的程序。本次修订,本书在以下几个方面对第 2 版做了较大的修改补充:

(1) 为了使教师能够更好地组织和实施教学过程,使读者能够更容易地接受和理解课程的内容,对部分章节的内容和讲解方法进行了改进,力求从实例出发循序渐进地引出概念,对概念和例题的分析讲解更加细致、透彻,更有利于读者自学。

(2) 对原书的内容作了十分慎重的斟酌,删掉了部分不是十分必要的内容,增加了一些新的更有用的内容,使本书更具实用性。增加了第 8 章面向对象程序设计方法与实例,以帮助读者进一步了解面向对象程序设计方法,提高解决实际问题的能力。

(3) 更新或增加了一些在实践教学效果比较好的例题,帮助读者举一反三,从中学习方法和技巧,从而更快地掌握 C++ 程序设计的方法和要领。

(4) 对习题部分作了较大的修订,大幅度地增加了题型和题量,帮助读者通过练习题检查对所学内容掌握的情况。

(5) 为了与 C++ 国际标准(ISO/IEC 14882)相一致,使用标准 C++ 的头文件改写了所有源程序。系统头文件不带后缀“.h”,使用系统库时使用命名空间 std。

作为本书的辅导教材,将同时出版《C++ 面向对象程序设计教程(第 3 版)习题解答与上机指导》一书,给出了教材中所有习题的参考答案,介绍了 C++ 上机操作方法,提供了上机实验题与参考解答,以供教师授课与学生学习时参考。

本书第8章由林小茶编写,6.4节和7.6节由周涛编写,各章的习题由陈昕编写,其他章节由陈维兴编写。全书由陈维兴组织编写并统稿。书中所有程序都经作者在 Visual C++ 6.0 上调试通过。

在本书的编写和出版过程中还得到了郑玉明、陈宝福、杨道沅、李春强、孙若莹等老师的帮助和支持,在此表示诚挚的感谢。

最后,借用本书再版的机会,向选择使用本书的老师和读者表示衷心的感谢,欢迎对本书的内容和编写方法提出批评和建议。

编者

2009年5月

第 2 版前言

本书第 1 版于 2000 年 3 月出版以来,颇受读者欢迎,到 2003 年底已重印 13 次。不少高校用其作为教材或考研参考书,取得了很好的教学效果。作者在多年教学和科研实践的基础上,听取了专家和读者的意见,并结合本人的教学经验,对原书作了认真修订。

这次修订保持了原书语言通俗、层次清晰、理论与实例结合的特点,力求做到深入浅出,将复杂的概念用简洁浅显的语言来讲述。使读者尽快地迈入面向对象程序设计的大门,迅速掌握 C++ 程序设计的基本技能和面向对象的概念和方法,并能编写出具有良好风格的程序。本书在以下几个方面对第 1 版进行了修订:

(1) 更换或增加了一定数量的例题,帮助读者举一反三,从中学习方法和技巧,从而更快地掌握 C++ 程序设计的方法和要领。

(2) 对部分章节(主要是第 4 章和第 5 章)的讲解方法进行了改进,力求从实例出发循序渐进地引出概念,对概念和例题的分析讲解更加细致、透彻,更有利于读者自学。

(3) 对原书的内容作了十分慎重的斟酌,删掉了部分不是十分必要的内容,增加了一些对编写程序有用的新内容,从而使本书更具有实用性。

(4) 对习题部分作了较大的修订,增加了题型和题量,帮助读者通过练习题检查自己对所学内容掌握的情况,从而积累编程经验。

我们将出版本教程的习题解答和实验指导书,作为辅导教材供教师和学生授课与学习时参考。

在本书的出版过程中,陈昕博士给予了很大的帮助,在此表示诚挚的感谢。

最后,借用本书再版的机会,向选择使用本书的老师和读者表示衷心的感谢,欢迎您对本书的内容和编写方法提出批评和建议。

编 者

2004 年 6 月

第 1 版前言

本书为全国高等学校电子信息类规划教材,由计算机教学指导委员会编审、推荐出版。北京信息工程学院陈维兴担任主编,张学群主审,李逊林任责任编委。

本教材的参考学时为 54 学时,其中授课 36 学时,上机 18 学时。

面向对象程序设计是不同于传统程序设计的一种新的程序设计范型。它对降低软件的复杂性,改善其重用性和维护性,提高软件的生产效率,有着十分重要的意义。因此面向对象的程序设计被普遍认为是程序设计方法学的一场实质性的革命。

C++ 语言是在 C 语言基础上扩充了面向对象机制而形成的一种面向对象的程序设计语言,它除了继承了 C 语言的全部优点和功能外,还支持面向对象程序设计。C++ 现在已成为介绍面向对象程序设计的首选语言。学习 C++ 不仅可以深刻理解和领会面向对象程序设计的特点和风格,掌握其方法和要领,而且可以使读者掌握一种十分流行和实用的程序设计语言。

近年来,许多高等院校纷纷将面向对象程序设计及面向对象技术正式列入教学计划,作为必修课或选修课。由于其重要意义,许多有识之士也纷纷把目光转向面向对象程序设计。

鉴于以上情况,我们在多年教学和科研的基础上编写了这本教材,旨在使读者迅速迈入面向对象程序设计的大门,掌握 C++ 程序设计的基本技能和面向对象的概念与方法,并能编写出具有良好风格的程序。

本教材共分 7 章。第 1 章概述了面向对象程序设计的基本概念。第 2 章介绍了 C++ 对 C 语言在非面向对象方面的扩充。第 3 章至第 7 章详述了 C++ 支持面向对象程序设计的基本方法,包括类、对象、派生类、继承、多态性、模板、流类库等。

本教材第 1 章由林小茶老师编写,其余章节由陈维兴老师编写。全书由陈维兴老师主编并统稿。张学群教授仔细审阅了全书并提出了许多宝贵的意见,在此表示诚挚的感谢。由于编者水平有限,书中难免还存在一些缺点和错误,殷切希望广大读者批评指正。

编 者

1999 年 6 月

目 录

第 1 章 面向对象程序设计概述	1
1.1 什么是面向对象程序设计	1
1.1.1 一种新的程序设计范型	1
1.1.2 面向对象程序设计的基本概念	2
1.1.3 面向对象程序设计的基本特征	4
1.2 为什么要使用面向对象程序设计	8
1.2.1 传统程序设计方法的局限性	8
1.2.2 面向对象程序设计方法的主要优点	9
1.3 面向对象程序设计的语言	11
1.3.1 面向对象程序设计语言的发展概况	11
1.3.2 几种典型的面向对象程序设计语言	12
习题	13
第 2 章 C++ 概述	14
2.1 C++ 的起源和特点	14
2.1.1 C++ 的起源	14
2.1.2 C++ 语言的特点	15
2.2 C++ 源程序的构成	15
2.2.1 简单的 C++ 程序	15
2.2.2 C++ 程序的结构特性	17
2.2.3 C++ 程序的编辑、编译、连接和运行	18
2.3 C++ 在非面向对象方面的扩充	18
2.3.1 注释行	19
2.3.2 C++ 的输入输出	19
2.3.3 灵活的局部变量说明	21
2.3.4 结构名、联合名和枚举名可直接作为类型名	22
2.3.5 const 修饰符	22
2.3.6 函数原型	25
2.3.7 内联函数	27
2.3.8 带有默认参数的函数	30
2.3.9 函数的重载	31
2.3.10 作用域运算符“::”	33

2.3.11 无名联合.....	34
2.3.12 强制类型转换.....	35
2.3.13 运算符 new 和 delete	35
2.3.14 引用.....	38
习题	44

第3章 类和对象	48
3.1 类与对象的基本概念.....	48
3.1.1 结构体与类	48
3.1.2 成员函数的定义	53
3.1.3 对象的定义及使用	56
3.1.4 类的作用域和类成员的访问属性	59
3.2 构造函数与析构函数.....	60
3.2.1 对象的初始化和构造函数	60
3.2.2 用成员初始化列表对数据成员初始化	65
3.2.3 构造函数的重载	67
3.2.4 带默认参数的构造函数	71
3.2.5 析构函数	72
3.3 对象数组与对象指针.....	75
3.3.1 对象数组	75
3.3.2 对象指针	78
3.3.3 this 指针	80
3.4 string 类	83
3.5 向函数传递对象.....	85
3.5.1 使用对象作为函数参数	85
3.5.2 使用对象指针作为函数参数	86
3.5.3 使用对象引用作为函数参数	87
3.6 对象的赋值和复制.....	88
3.6.1 对象赋值语句	88
3.6.2 拷贝构造函数	89
3.7 静态成员.....	96
3.7.1 静态数据成员	96
3.7.2 静态成员函数.....	101
3.8 友元	105
3.8.1 友元函数.....	105
3.8.2 友元类.....	110
3.9 类的组合	112
3.10 常类型.....	117
3.10.1 常引用.....	117

3.10.2	常对象	118
3.10.3	常对象成员	119
	习题	122
第4章	派生类与继承	135
4.1	派生类的概念	135
4.1.1	为什么要使用继承	135
4.1.2	派生类的声明	137
4.1.3	派生类的构成	138
4.1.4	基类成员在派生类中的访问属性	139
4.1.5	派生类对基类成员的访问规则	140
4.2	派生类的构造函数和析构函数	148
4.2.1	派生类构造函数和析构函数的执行顺序	149
4.2.2	派生类构造函数和析构函数的构造规则	150
4.3	调整基类成员在派生类中的访问属性的其他方法	157
4.3.1	同名成员	157
4.3.2	访问声明	159
4.4	多重继承	163
4.4.1	多重继承派生类的声明	163
4.4.2	多重继承派生类的构造函数与析构函数	165
4.4.3	虚基类	169
4.5	基类与派生类对象之间的赋值兼容关系	178
4.6	应用举例	181
	习题	187
第5章	多态性	197
5.1	编译时的多态性与运行时的多态性	197
5.2	运算符重载	198
5.2.1	在类外定义的运算符重载函数	198
5.2.2	友元运算符重载函数	202
5.2.3	成员运算符重载函数	208
5.2.4	成员运算符重载函数与友元运算符重载函数的比较	213
5.2.5	“++”和“--”的重载	215
5.2.6	赋值运算符“=”的重载	220
5.2.7	下标运算符“[]”的重载	224
5.3	类型转换	226
5.3.1	系统预定义类型间的转换	226
5.3.2	类类型与系统预定义类型间的转换	227
5.4	虚函数	234

5.4.1	虚函数的引入	235
5.4.2	虚函数的定义	238
5.4.3	纯虚函数和抽象类	248
5.5	应用举例	249
	习题	254
第 6 章	模板与异常处理	259
6.1	模板的概念	259
6.2	函数模板与模板函数	259
6.3	类模板与模板类	265
6.4	异常处理	272
6.4.1	异常处理概述	272
6.4.2	异常处理的方法	273
6.5	应用举例	277
	习题	282
第 7 章	C++ 的流类库与输入输出	285
7.1	C++ 为何建立自己的输入输出系统	285
7.2	C++ 流的概述	286
7.2.1	C++ 的输入输出流	286
7.2.2	预定义的流对象	288
7.2.3	输入输出流的成员函数	288
7.3	预定义类型的输入输出	291
7.3.1	插入运算符与提取运算符	291
7.3.2	输入输出的格式控制	293
7.4	用户自定义类型的输入输出	301
7.4.1	重载插入运算符	301
7.4.2	重载提取运算符	303
7.5	文件的输入输出	304
7.5.1	文件的打开与关闭	305
7.5.2	文件的读写	308
7.6	命名空间和头文件命名规则	316
7.6.1	命名空间	316
7.6.2	头文件命名规则	318
7.7	应用举例	319
	习题	322
第 8 章	STL 标准模板库	325
8.1	容器、算法和迭代器的基本概念	325

8.2 容器	326
8.2.1 vector 容器	326
8.2.2 list 容器	333
8.2.3 容器适配器	338
8.2.4 deque 容器	342
8.2.5 set、multiset、map 和 multimap 容器	344
本章小结	345
习题	346
第9章 面向对象程序设计方法与实例	349
9.1 面向对象程序设计的一般方法和技巧	349
9.1.1 问题分析和功能定义	350
9.1.2 对象(类)设计及实现	350
9.1.3 核心控制设计	352
9.1.4 编码与测试	352
9.1.5 进化	352
9.2 设计实例:模拟网上购书的结账功能	353
9.2.1 问题分析与功能定义	353
9.2.2 对象(类)设计	353
9.2.3 核心控制设计	355
9.2.4 编码与测试	356
习题	361
参考文献	362

第 1 章 面向对象程序设计概述

20 世纪 90 年代以来面向对象程序设计(object-oriented programming, OOP)异军突起,迅速地在全世界流行,并一跃而成为程序设计的主流技术。现在,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,不仅因为它是一种最先进的、新颖的计算机程序设计思想,更主要的是这种新的思想更接近人的思维活动,人们利用这种思想进行程序设计时,可以很大程度地提高编程能力,减少软件维护的开销。面向对象程序设计方法是通过增加软件的可扩充性和可重用性来提高程序员的编程能力的。这种思想与我们以前使用的方法有很大的不同,并且在理解上有一些难点,希望本章的内容能对读者有所帮助。

1.1 什么是面向对象程序设计

1.1.1 一种新的程序设计范型

面向对象程序设计是一种新的程序设计的范型(paradigm)。程序设计范型是指设计程序的规范、模型和风格,它是一类程序设计语言的基础。一种程序设计范型体现了一类语言的主要特征,这些特征能用以支持应用领域所希望的设计风格。不同的程序设计范型有不同的程序设计技术和方法学。

面向过程程序设计范型是流行很广泛的程序设计范型,这种范型的主要特征是,程序由过程定义和过程调用组成(简单地说,过程就是程序执行某项操作的一段代码,函数是最常用的过程),从这个意义出发,基于面向过程的程序可以用以下的公式来表述:

程序=过程+调用

基于面向过程程序设计范型的语言称为面向过程性语言,如 C、Pascal、Fortran、Ada 等都是典型的面向过程性语言。除面向过程程序设计范型外,还有许多其他程序设计范型。如函数式程序设计范型也是较为流行的程序设计范型,它的主要特征是,程序被看做“描述输入与输出之间关系”的数学函数,LISP 是支持这种范型的典型语言,此外,还有模块程序设计范型(典型语言是 Modula)、逻辑式程序设计范型(典型的语言是 PROLOG)、进程式程序设计范型、类型系统程序设计范型、事件程序设计范型、数据流程序设计范型等。

面向对象程序设计是一种新的程序设计范型。这种范型的主要特征是:

程序=对象+消息

面向对象程序的基本元素是对象,面向对象程序的主要结构特点是:第一,程序一般由类的定义和类的使用两部分组成;第二,程序中的一切操作都是通过向对象发送消息来实现的,对象接收到消息后,启动有关方法完成相应的操作。一个程序中涉及的类,可以由程序

设计者自己定义,也可以使用现成的类(包括类库中为用户提供的类和他人已构建好的类)。尽量使用现成的类,是面向对象程序设计范型所倡导的程序设计风格。

需要说明的是,某一种程序设计语言不一定与一种程序设计范型相对应。实际上具有两种或多种范型特征的程序设计语言,即混合型语言。例如C++就不是纯粹的面向对象程序设计范型的语言,而是具有面向过程程序设计范型和面向对象程序设计范型的混合型程序设计语言。

1.1.2 面向对象程序设计的基本概念

为了掌握面向对象程序设计技术,我们从最基本的概念入手。本节介绍的内容是面向对象程序设计的理论基础,这些内容不依赖于具体的程序设计语言。也就是说,无论使用哪种面向对象语言进行面向对象程序设计,本节内容都有理论意义。

1. 对象

我们从两个方面讨论对象的含义,即在现实世界中的含义和面向对象程序设计中的含义。

在现实世界中,任何事物都是对象。它可以是一个有形的具体存在的事物,例如一张桌子、一个学生、一辆汽车,甚至地球;也可以是一个无形的、抽象的事件,例如一次演出、一场球赛、一次出差等。对象既可以很简单,也可以很复杂,复杂的对象可以由若干简单的对象构成,整个世界可以认为是一个非常复杂的对象。

现实世界中的对象既具有静态的属性(或称状态),又具有动态的行为(或称操作、功能)。例如,一个人就是一个对象,每个人都有姓名、性别、年龄、身高、体重等属性,都有吃饭、走路、睡觉、学习等行为(功能)。所以在现实世界中,对象一般可以表示为:属性+行为,一个对象往往是由一组属性和一组行为构成的。

现实世界中的对象,具有以下特性:

- (1) 每一个对象必须有一个名字以区别于其他对象;
- (2) 用属性来描述对象的某些特征;
- (3) 有一组操作,每组操作决定对象的一种行为;
- (4) 对象的行为可以分为两类:一类是作用于自身的行为,另一类是作用于其他对象的行为。

面向对象的程序设计采用了以上人们所熟悉的这种思路。在面向对象程序设计中,对象是描述其属性的数据以及对这些数据施加的一组操作封装在一起构成的统一体。在C++中每个对象都是由数据和操作代码(通常用函数来实现)两部分组成的,如图1.1所示。在面向对象程序设计中,用数据来体现上面提到的“属性”,用函数来实现对数据的操作,以实现某些功能。

例如一个学生是一个对象,学号、姓名和成绩等数据就是它的属性;输入或输出姓名、学号、成绩等操作就是前面提到的行为。

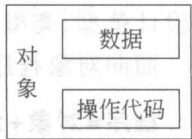


图 1.1 对象结构示意图

2. 类

在现实世界中,“类”是一组具有相同属性和行为的对象的抽象。例如,虽然张三、李四、王五……每个人的性格、爱好、职业、特长等各有不同,但是他们的基本特征是相似的,都具有相同的生理构造,都能吃饭、说话、走路等,于是把他们统称为“人”类,而具体的每一个人是人类的一个实例,也就是一个对象。

类和对象之间的关系是抽象和具体的关系。类是对多个对象进行综合抽象的结果,对象又是类的个体实物,一个对象是类的一个实例。

例如,教师黎明是一个对象。

对象名:黎明

对象的属性:

年龄:45

学历:博士

职称:教授

专业:计算机软件

对象的操作:

走路

吃饭

授课

一个个的像黎明这样的教师就构成教师类。类在现实世界中并不真正存在。例如,在地球上并没有抽象的“学生”,只有一个个具体的学生,如张三、李四、王五……同样,世界上没有抽象的“教师”,只有一个个具体的教师。

在面向对象程序设计中,“类”就是具有相同的数据和相同的操作(函数)的一组对象的集合,也就是说,类是对具有相同数据结构和相同操作的一类对象的描述。例如,“学生”类可由学号、姓名、成绩等表示其属性的数据项和对这些数据的录入、修改和显示等操作代码组成。在C++语言中把类中数据称为数据成员,类中的操作是用函数来实现的,这些函数称为成员函数。

类和对象之间的关系是抽象和具体的关系。类是多个对象进行综合抽象的结果,一个对象是类的一个实例。例如“学生”是一个类,它是由许多具体的学生抽象而来的一般概念。同理,桌子、教师、计算机等都是类。

在面向对象程序设计中,总是先声明类,再由类生成其对象。类是建立对象的“模板”,按照这个模板所建立的一个个具体的对象,就是类的实际例子,通常称为实例。打个比方,手工制作月饼时,先雕刻一个有凹下图案的木模,然后在木模上抹油,接着将事先揉好的面塞进木模里,用力挤压后,将木模反扣在桌上,一个漂亮的图案就会出现在月饼上了。这样一个接着一个地,就可以制造出外形一模一样的月饼。这个木模就好比是“类”,制造出来的糕点好比是“对象”。

3. 消息与方法

现实世界中的对象不是孤立存在的实体,它们之间存在着各种各样的联系,正是它们之