普通高等教育“十一五”国家级规划教材

北京市高等教育精品教材立项项目

C++面向对象程序设计教程

（第3版）

陈维兴 林小茶 编著

清华大学出版社

普通高等教育“十一五”国家级规划教材

北京市高等教育精品教材立项项目

C++面向对象程序设计教程

（第3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，全面地讲述了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题，以通俗易懂的语言讲解复杂的概念和方法，以期帮助读者尽快地迈入面向对象程序设计的大门。

本书自2000年出版第1版以来，深受读者欢迎。第2版被评为北京高等教育精品教材，第3版被评为普通高等教育“十一五”国家级规划教材。

本书内容全面、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++的入门教材，也适合作为初学者自学的教材。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3版)习题解答与上机指导》。

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．—3版．—北京：清华大学出版社，2009.6

ISBN 978-7-302-20007-9

Ⅰ．C…　Ⅱ．①陈…　②林…　Ⅲ．C语言－程序设计－教材　Ⅳ．TP312

中国版本图书馆CIP数据核字(2009)第061018号

责任编辑：柳　萍　曾　洁

责任校对：赵丽敏

责任印制：何　芊

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座

http://www.tup.com.cn　　**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 刷 者：清华大学印刷厂

装 订 者：北京市密云县京文制本装订厂

经　　销：全国新华书店

开　　本：185×260　　**印　张**：22.25　　**字　数**：533千字

版　　次：2009年6月第3版　　**印　次**：2009年6月第1次印刷

印　　数：1～8000

定　　价：32.00元

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中效果比较好的例题，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(IOS/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

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

目　录

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。

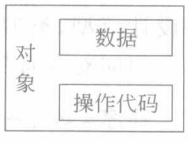

图1.1 对象结构示意图

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

间的相互作用、联系和连接，才构成了世间各种不同的系统。以实际生活为例，我们每一个人可以为他人服务，也可以要求他人为自己服务。当我们需要别人为自己服务时，必须告诉他们，我们需要的是什么服务。也就是说，要向其他对象提出请求，其他对象接到请求后，才会提供相应的服务。

在面向对象程序设计中，对象之间也需要联系，称为对象的交互。面向对象程序设计技术必须提供一种机制允许一个对象与另一个对象的交互。这种机制叫消息传递。一个对象向另一个对象发出的请求称为“消息”。当对象接收到发向它的消息时，就调用有关的方法，执行相应的操作。例如，有一个教师对象张三和一个学生对象李四，对象李四可以发出消息，请求对象张三演示一个实验，当对象张三接收到这个消息后，确定应完成的操作并执行。

一般情况下，称发送消息的对象为发送者或请求者，接收消息的对象为接收者或目标对象。对象中的联系只能通过消息传递来进行，接收对象只有在接收到消息时，才能被激活，被激活的对象会根据消息的要求完成相应的功能。

消息具有以下 3 个性质：

(1) 同一个对象可以接收不同形式的多个消息，做出不同的响应；

(2) 相同形式的消息可以传递给不同的对象，所做出的响应可以是不同的；

(3) 对消息的响应并不是必需的，对象可以响应消息，也可以不响应。

在面向对象程序设计中的消息传递实际是对现实世界中的信息传递的直接模拟。调用对象中的函数就是向该对象传送一个消息，要求该对象实现某一行为(功能、操作)。对象所能实现的行为(操作)，在程序设计方法中称为方法，它们是通过调用相应的函数来实现的，在C++ 语言中方法是通过成员函数来实现的。

方法包括界面和方法体两部分。方法的界面给出了方法名和调用协议(相对于C++ 中成员函数的函数名和参数表)；方法体则是实现某种操作的一系列计算步骤，也就是一段程序(相对于C++ 中成员函数的函数体)。消息和方法的关系是：对象根据接收到的消息，调用相应的方法；反过来，有了方法，对象才能响应相应的消息。

1.1.3 面向对象程序设计的基本特征

面向对象程序设计方法模拟人类习惯的解题方法，代表了计算机程序设计的新颖的思维方法。这种方法的提出是对软件开发方法的一场革命，是目前解决软件开发面临困难的最有希望、最有前途的方法之一。本节介绍面向对象程序设计的 4 个基本特征。

1. 抽象

抽象是人类认识问题的最基本的手段之一。抽象是将有关事物的共性归纳、集中的过程。在现实生活中，人们能看到的都是一些具体的事物，例如男人、女人、大人、小孩等，这些都是具体的人。假如，把所有具有大学学籍的人归为一类，称为“大学生”，这就是一个抽象。在抽象的过程中，通常会忽略与当前主题目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象是对复杂世界的简单表示，抽象并不打算了解全部问题，而只强调感兴趣的信息，忽略了与主题无关的信息。例如，在设计一个成绩管理程序的过程中，只关心学生的姓名、学号、成绩等，而对他的身高、体重等信息就可以忽略。而在学生健康信息管理系

统中，身高、体重等信息必须抽象出来，而成绩则可以忽略。

抽象是通过特定的实例（对象）抽取共同性质后形成概念的过程。面向对象程序设计中的抽象包括两个方面：数据抽象和代码抽象（或称为行为抽象）。前者描述某类对象的属性或状态，也就是此类对象区别于彼类对象的特征物理量；后者描述了某类对象的共同行为特征或具有的共同功能。正如前面所述，对于一组具有相同属性和行为的对象，可以抽象成一种类型，在C++ 中，这种类型就称为类（class），类是对象的抽象，而对象是类的实例。

抽象在系统分析、系统设计以及程序设计的发展中一直起着重要的作用。在面向对象程序设计方法中，对一个具体问题的抽象分析的结果，是通过类来描述和实现的。

现在以学生管理程序为例，通过对所有学生进行归纳、分析，抽取出其中的共性，可以得到如下的抽象描述：

（1）共同的属性：姓名、学号、成绩等，它们组成了学生类的数据抽象部分。用C++ 语言的数据成员来表示，可以是

```
char * name;                    //姓名
int number;                     //学号
float score;                    //成绩
```

（2）共同的行为：数据录入、数据修改和数据输出等，这构成了学生类的代码抽象（行为抽象）部分，用C++ 语言的成员函数表示，可以是

```
input()                         //数据录入函数
modify()                        //数据修改函数
print()                         //数据输出函数
```

如果我们开发一个学生健康档案程序，所关心的特征就有所不同了。可见，即使对同一个研究对象，由于所研究问题的侧重点不同，就可能产生不同的抽象结果。

2. 封装

在现实世界中，所谓封装就是把某个事物包围起来，使外界不知道该事物的具体内容。在面向对象程序设计中，封装是指把数据和实现操作的代码集中起来放在对象内部，并尽可能隐蔽对象的内部细节。对象好像是一个不透明的黑盒子，表示对象属性的数据和实现各个操作的代码都被封装在黑盒子里，从外面是看不见的，更不能从外面直接访问或修改这些数据及代码。使用一个对象的时候，只需知道它向外界提供的接口而无需知道它的数据结构细节和实现操作的算法。

下面以一台洗衣机为例，说明对象的封装特征。首先，每一台洗衣机有一些区别于其他洗衣机的静态属性，例如出厂日期、机器编号等。另外，洗衣机上有一些按键，如“启动”、“暂停”、“选择”等，当人们使用洗衣机时，只要根据需要按下“选择（洗衣的方式）”、“启动”或“暂停”等按键，洗衣机就会完成相应的工作。这些按键安装在洗衣机的表面，人们通过它们与洗衣机交流，告诉洗衣机应该做什么。我们无法（当然也没必要）操作洗衣机的内部电路和机械控制部件，因为它们被装在洗衣机里面，这对于用户来说是隐蔽的，不可见的。

面向对象的程序中使用一个对象时，只能通过对象与外界的操作接口来操作它，这个接口类似于洗衣机的按键。使用对象类似于使用洗衣机，不必了解洗衣机里面的结构和工作