高等教育计算机学科应用型规划教材

C/C++ 程序设计教程
——面向对象分册

郑秋生　主编
王黎明　主审

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高等教育计算机学科应用型规划教材

C/C++程序设计教程
——面向对象分册

郑秋生　主　编
宋宝卫　吴庆涛　夏敏捷　副主编
王黎明　主　审

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内容简介

《C/C++程序设计教程》系列教材分为面向过程和面向对象两个分册，适合分两个学期讲授。本书为面向对象分册。

本书阐述了C++语言中面向对象程序设计的语法和思想，主要内容包括类与对象、继承与派生、虚函数与多态性、异常处理、模板和STL标准模板库等内容。书中通过流行的UML工具描述C++类，并且内容讲解清晰、实例丰富、力避代码复杂冗长，注重程序设计思想。简短的实例和UML图特别有助于初学者更好地理解、把握解决问题的精髓，帮助读者快速掌握面向对象程序设计的基本方法。

本书的特点是实例丰富，重点突出，叙述深入浅出，分析问题透彻，既有完整的语法，又有大量的实例，突出程序设计的思想和方法，将C语言程序设计和C++程序设计有机地进行统一。

本书适合作为本科、专科院校计算机学科的C语言程序设计和C++语言程序设计课程教材，也可作为其他相关专业的教材和技术人员的自学参考书。

图书在版编目(CIP)数据

C/C++程序设计教程. 面向对象分册/郑秋生主编. —北京：电子工业出版社，2008.2
高等教育计算机学科应用型规划教材
ISBN 978-7-121-05730-4
Ⅰ. C… Ⅱ. 郑… 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(2007)第206166号

策划编辑：张 旭
责任编辑：张燕虹
印 刷：北京市海淀区四季青印刷厂
装 订：北京牛山世兴印刷厂
出版发行：电子工业出版社
北京市海淀区万寿路173信箱 邮编 100036
开 本：787×1092 1/16 印张：20.25 字数：531千字
印 次：2008年2月第1次印刷
定 价：29.80元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：(010)88258888。

编 委 会

出版说明

高等教育的教学改革及课程建设总是伴随着科技的进步与生产的发展而发展的。当前高等教育既要培养理论基础扎实、高素质的科研型人才，也要培养具有一定的理论基础更具有较高工程能力的应用型人才。为了满足普通高等院校面向应用的需求，进一步提高高等院校教学质量和教学水平，电子工业出版社与河南省计算机学会共同组织了高等教育计算机学科应用型规划教材的编写和出版工作。

高等教育计算机学科应用型规划教材根据培养目标和对象不同，总结教学改革和教材建设经验，在基础理论方面做出了合理的取舍，同时融入了现代科技应用的成果。这是一种理论与实践、基础知识与现代技术有机结合的教材。

本套教材定位于国内普通高等院校本科、专科的学生，也适用于高职高专、成人教育的学生。教材内容充分考虑学生的知识水平、理解能力和教学要求，遵循由浅入深、循序渐进的原则，适合学生自学和教师教学。

本套教材符合相应教学大纲的基本要求，结合案例（实例）展开教学内容，侧重应用，突出实践，强调理论与实践结合。

本套教材努力从学习者（学生、自学者）的角度阐述理论知识，充分利用图表进行形象化表达，适当补充相关知识内容，引导读者阅读相关书籍。教材内容的选取注重帮助读者建立完整的知识结构，而不是仅仅掌握某个知识单元。教材内容关注计算机技术的迅猛发展，及时补充最新技术。

本套教材努力提供丰富的教学辅助资源，建立师生交流平台，以便于教师、学生使用。读者可以通过电子工业出版社的华信教育资源网站（www. huaxin. edu. cn 或 www. hxedu. com. cn）了解本套教材的出版和服务的动态信息。

河南省计算机学会

电子工业出版社

前　言

本书的主要作者都是具有丰富教学经验的一线教师，从事 C/C++ 程序设计课程教学多年，深知学生在学习 C++ 程序设计这门课程后，对程序设计方法、算法设计、调试程序、习题解答的茫然和问题。因此本书在介绍理论知识、相关概念和语言语法时，始终强调其在程序设计中的作用，使语言语法与程序设计相结合。同类书籍大部分偏重于对语言语法和概念的介绍，虽然在书中有针对一个语法和知识点的程序实例，但学生对每章内容在实际程序设计中的作用缺乏了解，而本书每章节后都附有针对性较强的应用实例分析，尽可能使初学者在学习每章的内容后，拿到题目，既能够独立设计程序、解决实际问题，又不至于无从下手。

本书有以下五个鲜明特点：

（1）改变了传统的教学模式。先介绍 C 语言程序设计，再介绍 C++ 对 C 语言的扩展、面向对象的程序设计。本教材将 C/C++ 语言的学习很好地融合在一起，让读者把面向过程和面向对象的程序设计方法有机地结合在一起，面向过程和面向对象两分册统一使用 Visual C++ 6.0编译器。

（2）克服了传统教材以语言、语法学习为重点的缺点，本教材从基本的语言、语法学习上升到程序的“设计、算法、编程、调试”层次。为了让学生更好地掌握程序开发思想、方法和算法，书中提供了大量简短精辟的代码，有助于初学者学习解决问题的方法和诀窍。在每章后都有单独一节讲述关于程序综合设计的内容，有一个或多个较大的程序，以帮助学生更好地掌握程序设计方法和解决实际问题的能力。

（3）教材强调程序的设计方法，大量例题有流程图、N-S 图和 UML 图，即突出程序的算法和设计，而不仅仅是语法和编程，培养学生程序设计能力和程序调试技能，养成好的编程习惯，为专业程序员的培养打下良好的基础。

（4）培养学生面向对象程序设计的能力，引导学生建立程序设计的大局观，帮助学生掌握从客观事物中抽象出 C++ 类的方法。通过系统的学习，使学生的编程能力上一个台阶，具备解决复杂问题的程序设计能力。

（5）根据当前实际大型软件项目开发的需要，加大了异常处理、模板等内容，新增了 STL 标准模板库，并通过流行的 UML 工具设计 C++ 类。

本教材的编写充分考虑了目前应用型本科 C/C++ 语言程序设计课程教学的实际情况和存在的问题：

（1）学生在大一阶段的基础课程较多，不可能投入过多的精力来学习本门课程。

（2）大学生对这门课学习的期望值很高，但对学习时可能遇到的困难估计不足。

（3）大学生现有的上机实践条件大大改善，特别有利于贯彻先进的精讲多练的教学思想。

（4）学生学会了语言的语法，仍不具备解决实际问题的能力，学生的程序设计、算法设计、编程、调试的能力相对较差。

本教材正是考虑了学生的这些实际问题，而精心编写了这套面向应用型本科的C/C++程序设计教程，特别适合于分两个学期系统讲授C/C++程序设计。第1学期讲授面向过程分册，第2学期讲授面向对象分册。

本书共8章，第1章到第3章主要阐述面向对象程序设计的重要概念：类和对象，继承与派生、虚函数与多态性。第4章介绍输入/输出流技术。第5章主要介绍异常的概念、异常的产生以及异常的处理机制。第6章和第7章介绍模板和STL标准模板库。第8章主要讲述面向对象的分析与设计方法，并以实例形式详细介绍如何用C++语言进行程序设计。

为了方便使用本教材的教师备课，我们还提供了配套的电子教案，公开放在网站上，供任课教师自由下载使用。相信我们多年的教学经验会对广大师生的教和学有所帮助。建议本分册的教学学时为60个学时，其中理教为44学时，课内上机实践为16学时，课外上机不少于32学时。

本教材的编写得到了河南省计算机学会的大力支持，该学会组织了河南多所高校编写了高等教育计算机学科应用型规划教材。参编本教材的高校有中原工学院、郑州大学、洛阳师范学院、河南工程学院、河南科技大学、郑州轻工业学院。

本书由郑秋生任主编，由宋宝卫、吴庆涛、夏敏捷任副主编，由王黎明任主审。第1章由夏敏捷和潘惠勇编写，第2章由罗菁编写，第3章由程传鹏编写，第4章由刘钺编写，第5章由郑秋生编写，第6章由张明川编写，第7章和附录A由吴庆涛编写，第8章和附录B由宋宝卫编写。全书由郑秋生修改并统稿。郑州大学王黎明和钱晓捷老师为本书提出改进意见，在此谨向他们表示衷心的感谢。

由于编者水平有限，时间仓促，书中难免有错，敬请广大读者批评指正，在此表示感谢。

作者E-mail：zqs@zzti.edu.cn

作　者

目　录

绪　论

C++语言不仅支持面向过程的程序设计，也支持面向对象的程序设计。在《C/C++程序设计教程》系列教材的面向过程分册中，介绍的是C++在面向过程的结构化程序设计中的应用。

传统的面向过程的结构化程序设计的基本思想是采用自顶向下、逐步细化的设计方法。程序为处理数据的一系列过程，由三种基本控制结构构成，它们分别是顺序结构、选择结构和循环结构。在结构化程序设计中，自顶向下对任务进行模块划分的基本单位是函数。

面向过程的程序设计方法可以归结为“程序＝算法＋数据结构”；这种设计方法的着眼点是面向过程的，特点是算法（数据处理）和数据结构是分离的，适合开发规模较小的程序。但是，在20世纪80年代末，这种设计方法逐渐暴露出以下缺陷：

（1）难以适应大型软件的设计。

（2）程序可重用性差。

当程序规模较大时，相应的数据和算法也变得复杂，程序之间的耦合问题突出，这种设计方法就已经接近或达到面向过程的结构化程序设计的工作极限。为了解决这个问题，面向对象的程序设计方法便应运而生。

为什么要引入面向对象的程序设计方法？面向对象的设计方法与面向过程的设计方法有什么关系？

面向对象程序设计，是针对开发较大规模的程序而提出来的，目的是提高软件开发的效率。面向过程程序设计的缺点的根源在于数据与数据处理分离；而面向对象程序设计方法是软件程序设计中的一种新的思想，将数据和对数据的操作方法放在一起，形成一个相对独立的整体——对象（object），相似的对象还可抽象出共性，形成类（class）。一个类中的数据通常只能通过本类提供的方法进行处理，这些方法成为该类与外部的接口。对象之间通过消息（message）进行通信。面向对象程序设计模拟人们认识自然界和处理事物的方法，设计思路与人们日常生活中处理问题的思路相似，使程序设计更能贴近现实。面向对象程序设计的三大特点是封装、继承和多态。

当然，面向对象的程序设计方法是在面向过程程序设计的基础上发展而来的，面向对象程序设计吸取了面向过程的结构化程序设计的先进思想。面向对象的程序设计方法就是将需要完成的程序分解成一些不可分解的对象，而且尽可能与现实中的对象相联系；然后抽象成相应的类来封装和复用。就一个对象来说，它的数据结构和对这些数据的算法的复杂程度不会很大，解决了面向过程的结构化程序设计方法的弊端。

最后，注意不要把面向对象和面向过程对立起来，面向对象和面向过程不是矛盾的，而是各有用途、互为补充的。

第1章

类和对象

C++是一种面向对象程序设计语言，为面向对象技术提供全面的支持。学习C++语言，首先要认识类和对象，掌握它面向对象的特性和实现面向对象的方法。类是构成实现面向对象程序设计的基础，也是C++封装的基本单元，对象是类的实例。本章主要介绍类和对象的基本概念、类的定义和使用、类的一些特性。

通过本章学习，应该重点掌握以下内容：

- 面向对象程序设计的基本特性
- 类和对象的定义和使用
- 构造函数和析构函数
- 拷贝构造函数和堆对象、对象数组
- 友元函数和友元类
- 类的静态成员

1.1 面向对象程序设计概述

面向对象程序设计(Object Oriented Programming,OOP)是软件系统设计与实现的新方法,这种新方法既吸取了结构化程序设计的全部优点,又考虑了现实世界与面向对象空间的映射关系,所追求的目标是使现实世界的问题求解尽可能简单化。在自然世界和社会生活中,一个复杂的事物总是由很多部分组成的。例如,一个人由姓名、性别、年龄、身高、体重等特征描述;一辆自行车由轮子、车身、车把等部件组成;一台计算机由主机、显示器、键盘、鼠标等部件组成。当人们生产一台计算机的时候,并不是先生产主机,然后生产显示器,最后生产键盘、鼠标,即不是顺序执行的;而是分别生产主机、显示器、键盘、鼠标等,最后把它们组装起来。这些部件通过事先设计好的接口连接在一起,以便协调地工作。例如,通过键盘输入可以在显示器上显示字或图形。这就是面向对象程序设计的基本思路。

1.2 面向对象方法的基本特征

面向对象程序设计方法提出了一些全新的概念,如类和对象、封装与数据隐藏、继承与重用、多态性等,下面分别加以讨论。

1.2.1 对象和类

客观世界中的事物可分为两大部分:意识和物质。意识形态描述的是一个抽象的概念,物质形态表达具体的事物。例如"人"和"小王这个人","人"是意识,是一个抽象的概念,是对现实世界中事物的一种抽象概括;而"小王这个人"是物质,是客观存在的、具体的。现实世界中的事物与面向对象系统可以相对应,即现实世界中任何一个物质对应面向对象系统中的对象,实现世界中的意识对应面向对象系统中的抽象概念——类。现实世界与面向对象系统的对应关系如图 1-1 所示。

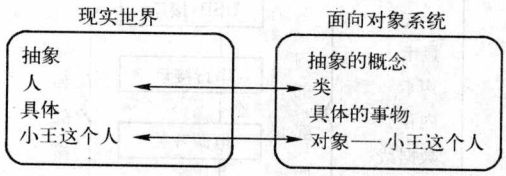

图 1-1 现实世界与面向对象系统的对应关系

在面向对象系统中,类是一种复杂的数据类型,它将不同类型的数据、与这些数据相关的操作封装在一起。类是对现实世界客观事物的抽象。对象是类的一个实体,又称为实例。一个类可以有多个对象。类有两方面的要素:属性(attribute)和行为(behavior)。属性描述的是事物的静态特征,如一个人的姓名、性别、年龄、身高等;行为又称为方法(method),描述的是事物的动态特征,如一个人可以走路、说话、学习、开车等。对象与类的关系如图 1-2 所示。如果想控制一个人的行为,就要从外部给他一个指令(例如,上课铃响了,某人开始学习),这个指称为消息(message)。面向对象程序设计中,将数据(属性)和数据的操作(行为)封装在一起,作为一个不可分割的整体,而各个对象之间通过发送不同的消息来实现协调地工作。在 C++

中，每个对象都是由数据（表示对象属性）和函数（操作代码，表示对象的行为）这两部分组成的，调用对象中的函数就是向该对象传送一个消息，要求该对象实现其中功能（行为）。对象之间的关系如图 1-3 所示。

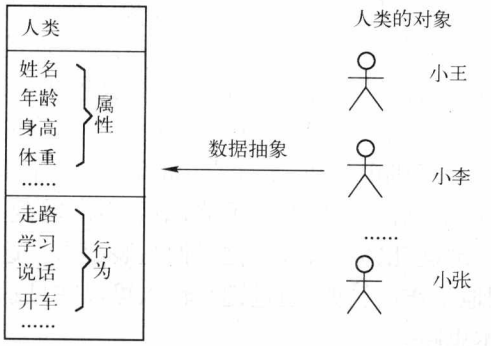

图 1-2　对象与类的关系

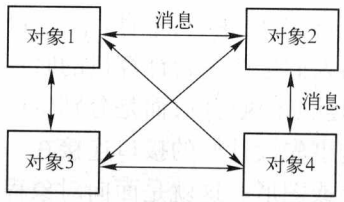

图 1-3　对象之间的关系

1.2.2　封装（encapsulation）与数据隐藏

封装是指将对象的数据与这个数据相关的操作放在一起，形成一个实体——对象。各个对象之间相互独立，互不干扰。对象只留少量的接口，以便与外部联系。从外部看，对象就像一个“黑匣子”，数据和方法是隐蔽的、看不见的。当用户使用对象时，不必知道对象的具体实现细节，只需根据对象提供的外部接口访问对象即可。例如，计算机的主机中包括了主板、显卡、声卡、数据线、电源线等部件，如图 1-4 所示。在使用计算机主机时，不必关心内部部件是如何工作的（例如，显卡是如何显示图像的等），只要会操作主机面板上的开关或提供接口就可以了。

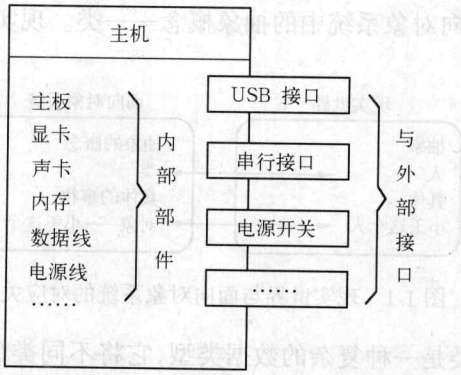

图 1-4　计算机主机数据封装示意图

从上述描述可以看出，封装应具有以下特点：

（1）具有一个清楚的边界，内部部件封装在内部，外部不可以访问。

（2）具有接口，如果将所有的数据和方法都封装在内部，那么就像是一个没有门的房子，房子中的装置再好也没有人能使用，所以必须提供必要的接口。

（3）对象内部的数据和方法是受封装外壳保护的，其他对象不能直接使用。

1.2.3 继承(inheritance)与重用

继承是面向对象程序设计中另一个重要的概念,前面讨论的类是独立的一个实体,各个类之间是平等的,在同一级别上;但在现实世界中,各种类之间存在着复杂的关系。例如,电视机是一个类,彩色电视机也是一个类,电视机和彩色电视机有相似的属性、方法,并没有表现出来。从以上分析可以发现,彩色电视机只不过是在原来电视机的基础上添加了一些新的属性和方法而已。为了解决这类问题,C++引入了继承机制。

假设已经定义了一个电视机类,现在需要定义一个彩色电视机类,有两种方法可供选择:一是重新设计,二是继承原来电视机的属性和方法,再添加彩色电视机自己特有的新的属性和行为。显然第二种方法更合适。在C++中,把"电视机类"称为"父类"或"基类",把"彩色电视机类"称为"子类"或"派生类"。

C++提供的继承机制提供了类之间相互关系的解决方案,使某个类可以继承另外一个类的特征和能力。使用继承符合人们对事物的认识和叙述,大大简化了对问题的描述,提高了程序的可重用性,从而提高了程序设计、修改、扩充的效率,实现软件重用(software reusability)。

1.2.4 多态性(polymorphism)

多态性是面向对象程序设计的一个重要特征。如果一种语言只支持类而不支持多态,那么这种语言只能称为基于对象的语言(如Visual Basic,VB),而不能称为面向对象的语言。多态性描述的是,同一个消息可以根据发送消息对象的不同而采用不同的行为方式。比如学校的上课铃响,不同班级的学生进入不同的教室学习,不同的老师进入不同教室开始讲课,不同的对象会做出不同的响应。可以看到学生和教师在接收到同一消息(上课铃声)时,执行不同的操作,这就是多态的表现。

C++支持多态。C++中的多态性包括静态多态性和动态多态性。静态多态性是在编译的时候就能决定调用哪个函数,比如函数重载,根据参数不同,执行不同操作。动态多态性是由继承产生的,相关的但不同的类对象对同一消息会做出不同的响应。这种情况多是在程序运行过程中才能确定操作的对象,C++中通常使用虚函数(virtual)来实现动态多态性。

1.2.5 消息

面向对象技术的封装使得对象相互独立,各个对象要想相互协作实现系统的功能,则需要对象之间的消息传递机制。消息是一个对象向另一个对象发出的服务请求,进行对象之间的通信,也可以说是一个对象调用另一个对象的方法(method)或称为函数(function)。

通常,把发送消息的对象称为发送者,把接收消息的对象称为接收者。在对象传递消息中只包含发送者的要求,指示接收者要完成哪些处理,但并不告诉接收者应该如何完成这些处理。接收者接收到消息后,要独立决定采用什么方式完成所需的处理。同一对象可接收不同形式的多个消息,产生不同的响应;相同形式的消息可送给不同的对象,不同的对象对于形式相同的消息可以有不同的解释,做出不同的响应。

在面向对象设计设计中,对象是节点,消息是纽带。应注意不要过度侧重如何构建对象及对象间的各种关系,而忽略对消息(对象间的通信机制)的设计。