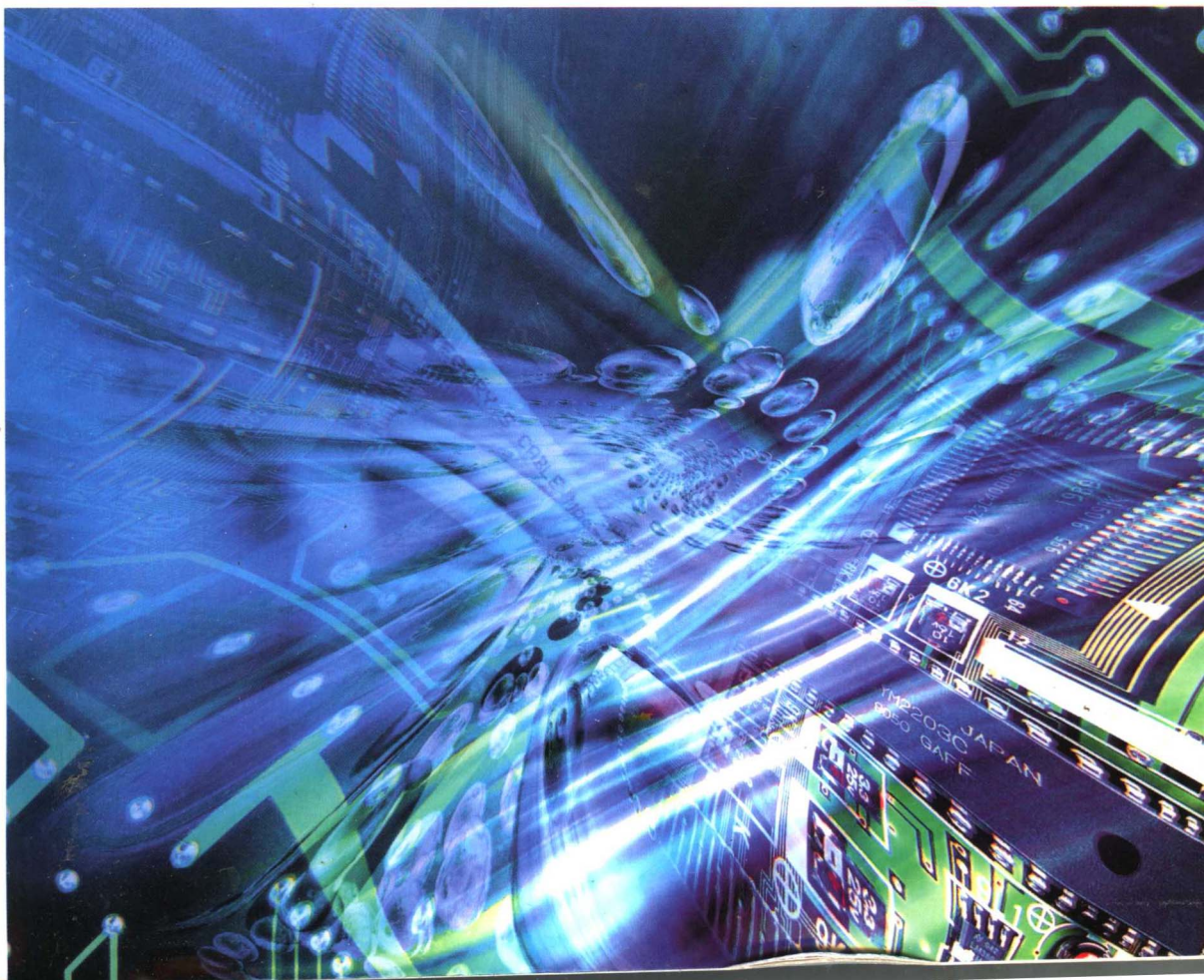
21世纪高等院校十一五规划教材

内蒙古自治区计算机教材编委会 组编

主编 吴 敏 杨国林

C++面向对象程序设计



内蒙古大学出版社

●21世纪高等院校十一五规划教材

C++面向对象程序设计

内蒙古自治区计算机教材编委会　　组编

吴敏　杨国林　主编

吴敏　杨国林　赵永红　刘月峰　金涛　徐广宇　编著

内蒙古大学出版社

内 容 简 介

C + +语言是一种高效实用的编程语言。C + +是在C语言基础上发展演变而来的一种面向对象程序设计语言。它既支持面向过程的程序设计方法,也支持面向对象的程序设计方法。学好C + +语言,再学习其他面向对象软件很容易触类旁通。本书全面系统地讲述了C + +语言的基础知识,基本语法和编程方法。重点讲述了C + +面向对象的重要特征:类和对象,继承性,派生性,多态性和虚函数等内容。本书第1章至第4章是基础部分,比较详细的介绍了C + +与C语言兼容的面向过程部分。第5章至第12章是面向对象程序设计部分,讲述C + +面向对象程序设计方法。本书文字通俗易懂,内容深入浅出,概念系统全面,讲解突出重点,侧重应用。

本书不但适合用于计算机专业作为程序设计基础课教材,也可作为非计算机专业的程序设计课程教材,还可以作为自学C + +语言的指导书和参考书。

图书在版编目(CIP)数据

C + +面向对象程序设计/吴敏,杨国林主编. —呼和浩特:内蒙古大学出版社,2006.7

ISBN 7-81074-977-3

Ⅰ.C... Ⅱ.①吴... ②杨... Ⅲ.C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(2006)第082201号

C + +面向对象程序设计

主　编　吴　敏　杨国林

编　著　吴　敏　杨国林　赵永红

刘月峰　金　涛　徐广宇

内蒙古大学出版社出版发行

内蒙古瑞德教育印务股份有限公司印刷

开本:787×1092/16　印张:23　字数:559千

2006年7月第1版　2006年7月第1次印刷

ISBN 7-81074-977-3/TP·37

定价:28.00元

内蒙古自治区计算机教材编委会

序

内蒙古自治区的高等教育事业起步于20世纪50年代初。经过近50年的发展，我区的高等教育无论从规模上，还是质量上都取得了长足的发展。特别是近些年来，全区高等院校的招生数量成倍增长，部分院校的合并使得一些高校的办学规模迅速壮大，形成了几所万人大学。与此同时，各高校对各自的专业及课程设置都做了较大的调整，以适应当今日益发展变化的高等教育事业。面向21世纪，在科学技术日新月异，社会对人才的知识结构、层次要求越来越高的新形势下，我们的高等教育的教学水平，特别是教材建设都应有一个更新更高的要求。

回顾50年来的发展，虽然我区高等教育的教学科研水平有了较大的提高，但与之相应的教材建设的现状还不尽如人意，绝大多数主干课程的教材还沿用一些传统教材，有些甚至是20世纪七八十年代的版本。有些院校的教材选用则有一定的随机性，在几种版本的教材之中换来换去。其间，虽然部分院校也组织力量编写了一些基础课及专业课教材，但大都是各成体系，缺乏院校间的协作与交流，形不成规模，质量亦无法保证，常常滞后于学科的发展与课程的变化。这都与我区高等教育的发展极不协调。诚然，区外部分地区高校的教学科研水平比我区要高，一些教材的质量好，我们可以直接利用，但这并不能成为我们不搞教材建设的理由。好的教材还需要相应的教育资源条件与之相对应才能取得良好的教学效果，从而达到促进教学质量提高之目的。应当承认，由于经济发展的相对落后，我区高校所招学生的基础和学校的教学条件比起全国重点名牌大学相对要差一些。因而，我们高校的教材也应从实际出发，结合自己学校和学生的特点，逐步探索、建立一套适合自治区教育资源条件的教材体系，促进自治区高校教学科研水平的提高，多出人才，出好人才。

值得欣喜的是，随着自治区教育科学水平的提高，我区高校教育领域的一些有识之士逐渐认识到，面向21世纪，未来高校之间的竞争就是学校的产品——学生质量的竞争。要想培养出高水平、高素质的学生，使我区的高校在这种竞争中立于不败之地，除各高校应努力提高自身的教学组织管理水平、提高教师的素质外，还应积极主动地加强与区内外高校的协作、交流，取长补短，走联合发展的道路，使我区高等教育的整体水平能够在较短的时间内得到提高。为此，在有利于规范高校教材体系，促进高校教育质量的提高，加强各高校教学科研人员之间的协作与交流的原则下，由自治区教育厅牵头，内蒙古大学出版社组办、资助，联合全区高等院校的有关专家、学者共同组建成立一些相关专业的教材编委会，以求编写适合我区高等教育特点的教材，逐步建立、完善自治区高等教育的教学、教材体系，并开展一些与教学相关的科研工作。我们希望，通过教材编委会这种工作模式，建设一批高质量的教材，带出一支高水平的师资队伍，培养出大批高素质的人才。

我坚信，在自治区教育厅的指导下，在编委会各位专家、学者的辛勤工作下，在各院校的相互理解、相互协作、相互支持下，我们一定能够克服发展过程中的困难，逐步推出一批高质量、高水平的教材，为推进内蒙古自治区高等教育事业做出重要的贡献。

李建国

2002年3月19日

前　言

C++语言是目前公认的热门编程语言之一。C++是在C语言基础上发展演变而来的一种面向对象程序设计语言。它既支持面向过程的程序设计方法，也支持面向对象的程序设计方法。C++全面兼容了C语言，但是C++本身也是一个完整的程序设计语言。它在面向过程方面不但对C语言进行了扩展，而且提供了比C更严格、更安全的语法要求。同时C++增加了面向对象编程、数据抽象、类属编程等技术支持，通过继承和多态性，使程序具有很高的可重用性，使软件的开发和维护更为方便。C++语言既可以用于开发系统软件，也普遍用于开发应用软件，同时也广泛地应用于科研和教学。C++的国际标准早在1998年就已经制定并发布。它是目前应用最广的面向对象程序设计语言之一。

本书主要面向广大C++语言的初学者，考虑到读者可能没有C语言基础，本书的前几章比较详细的介绍了C++与C语言兼容的面向过程部分。如果读者对C语言比较熟悉，可以快速阅读第2、3、4章C++对C语言的扩充部分，然后直接进入第5章学习。本书内容深入浅出，概念全面。针对初学读者和自学读者的特点，力求将复杂的概念用通俗易懂、简洁浅显的语言进行讲述，不但适用于计算机专业作为程序设计基础课教材，也可作为电子信息、通信、自动化等相关专业的程序设计课程教材。

本书共计12章，分为四大部分：

(1)C++面向过程部分。这部分内容是学习后面各章的基础知识，分别在第2、3、4章讲述。第2章介绍了C++的数据类型、各种表达式以及程序的三种基本结构。第3章讲述了C++的函数。在面向对象程序设计中，函数是模块划分的基本单位，函数也为重用技术提供了支持。本章从应用角度讲述了函数的定义和使用方法。第4章讲述了数组、指针和字符串的概念，以及相关内存管理的知识。

(2)C++面向对象程序设计的基本概念。这部分内容是在第5、6章。第5章讲述了类与对象，这是面向对象程序设计的核心概念。本章讲述类与对象的定义，以及用类和对象的概念编程来解决实际问题。第6章讲述了使用类的静态成员和友元实现数据共享的概念。

(3)C++面向对象程序设计的深入讨论。这部分内容是C++面向对象程序设计的重点和精华部分，充分体现出面向对象的可重用性、访问控制、继承与派生和多态性的特点。第7章介绍运算符重载。运算符可以看成特殊的函数，运算符重载是函数重载的一种形式，它是C++程序设计的重要部分。第8章为继承与派生，C++利用继承和派生机制，解决了对编写程序无法重复使用而造成资源浪费的缺点。第9章为多态性，讲述类的另一个重要特性。所谓多态性是指发出的消息被不同类型的对象接受时导致不同的行为，是对类的特定成员函数——虚函数的再抽象。第10章为模板，模板是C++的高级特性，是建立具有类型安全的类库和函数库的技术支持。

(4)C++I/O流类库和异常处理概念。C++与C语言一样没有输入/输出语句，其输入/输出可以通过I/O流类库实现。在本书第11章介绍了流的概念，并介绍了流类库的结构和使用。C++的异常处理是对所能预料到的运行错误进行处理的一套机制，通过这套机制，程序能更好地从这些异常事件中恢复过来。本书第12章讲述了异常处理的概念。

本书第1、5、10章由吴敏编写，第4、6章由杨国林编写，第2、3章由赵永红编写，第7章

由刘月峰编写，第8、9章由金涛编写，第11、12章由徐广宇编写。本书所有例题都在Visual C++6.0版本的编译系统下调试通过。全书由吴敏、杨国林修改定稿。

在本书的编写过程中，得到了内蒙古自治区计算机教材编委会的各级领导、专家的大力支持和帮助，在此不胜感谢。经由内蒙古自治区计算机教材编委会审定，本书作为内蒙古自治区高校计算机系列教材之一正式出版。另外，特别感谢内蒙古大学出版社编辑部主任呼和老师，在本书的编写过程中给予了的许多具体、及时的支持和帮助。

由于作者水平有限，错误和不足之处在所难免，恳请读者批评指正。

作　者

2006年7月4日

目　录

第1章 概 述

1.1 面向对象程序设计思想

C++语言是一种面向对象的程序设计语言，使用C++语言可以实现面向对象的程序设计。但是，由于C++是C语言的扩展，它分享了C语言的许多技术特征、属性和优点。同时提供了对面向对象程序设计的全面支持。由此，C++语言包括了面向过程的过程性部分和面向对象的类部分。在介绍面向对象的概念之前，我门需要了解一下过程的结构化程序设计与面向对象程序设计的联系与区别。

1.1.1 结构化程序设计

结构化程序设计强调程序设计的风格和程序结构的规范化。它的基本思路是：把一个复杂问题的功能逐步分解，求解过程分阶段进行。大问题拆分为一系列较小的功能模块，直到这些自完备的子任务小到易于理解、易于处理的范围内。结构化程序设计的基本思想具体是：

1. 自顶向下。

2. 逐步细化。

3. 模块化设计：将整个程序划分为若干功能相对独立的子模块，并要求这些子模块的关系尽可能简单；子模块还可以继续划分，直至最简。

4. 结构化编码：每一个模块最终都可以用顺序、选择、循环三种基本结构来实现。

从以上四步可以看出，结构化程序设计可以有效地将一个复杂的问题分解为若干易于处理的子任务，而每一个子任务都可以独立编写程序来解决，从而将整个系统划分成多个子模块，程序中的子模块在C语言中通常用函数实现。

结构化程序设计方法有许多优点，如程序的各个模块可以独立编写，程序便于阅读、修改和维护，从而减少了程序出错的机会。功能独立的子模块可以构成函数库，有利于实现软件的复用。为了处理复杂的问题，结构化程序设计成功地提供了有力的手段，立即为广大的程序设计人员接受和使用，成为当时程序设计的主要方法。

虽然结构化程序设计具有很多优点，但它仍是一种面向过程的程序设计方法。在结构化程序设计中，把解决问题的过程(如函数)作为程序的重点。程序中的数据与处理数据的过程是分离的，当数据量或数据结构发生变化时，所有进行相关处理的过程都要进行修改。每一个相对老的程序系统，若采用新的方法或结构进行修改和扩充都会带来额外的开销。程序的可重用性差。当然，自顶而下的结构化程序设计，其思路是将一个复杂的程序分解为相互联系的，但又彼此独立的程序模块，这种分而置之的方法对于小程序和初学者来说，是一种比较有效的技术。随着软件技术的发展，如图形用户界面的应用，软件的使用越来越方便，也就要求程序的数据处理能力越来越强。同时程序的数据量也变得越来越大。然而结构化程序设计中，数据与处理数据的过程(如函数)分离，使得程序的开发和维护变得更复杂。同时，大规

模的软件系统也很难用过程来描述和实现。这种面向过程的设计方法，使系统的开发、维护工作变得越来越困难。为解决软件设计危机，提出了面向对象程序设计方法。

1.1.2 面向对象的概念

面向对象是一种软件开发设计方法，它是在面向过程的结构化程序设计方法面临困难的背景下，应运而生的一种新的软件开发方法。

什么是面向对象的方法呢？下面先来介绍一下数据类型，数据类型可以分为三种：简单数据类型，用户自定义数据类型，抽象数据类型。

1. 简单数据类型：语言本身提供的，或称为系统预定义类型，如 C 语言中的整型、实型、字符型等。

2. 用户自定义数据类型：以简单数据类型为基础，它包含的数据成分是多个简单数据类型的数据（或是已定义的其它自定义数据类型的数据），如 C 语言中的结构体类型、共用体类型。

对于以上两种数据类型，它们的数据定义和对数据进行的操作是分开的。一般是要先定义数据，后允许对数据进行操作。

3. 抽象数据类型：在定义数据时，必须同时定义对数据进行的操作，用抽象数据类型定义的数据就称为对象（或实例）。

那么，面向对象方法中的对象是将“一组数据和对该数据的操作方法放在一起，作为一个相互依存、不可分离的整体”。对同类型对象抽象出共同特性，形成抽象数据类型——类。面向对象程序设计就是一种应用对象、类、封装、聚合、继承、消息传递、多态等概念来构造系统的软件开发方法。这里提出了一些新的概念，这些新概念描述了面向对象方法的特点。当然我们不期望通过简单的介绍就要求读者完全理解这些概念，在本书的后续章节中，将不断地帮助读者加深对这些概念的理解，以达到熟练运用的目的。

（1）对象是面向对象程序设计中的基本单位

对象可以说是现实世界中的一个实际事物，比如一个学生或一辆汽车。对象是构成世界的一个独立单位，它具有自己的静态特性（可以用数据来描述）和动态特征（行为或功能）。对象是面向对象语言中构成程序的主要成员，它是程序中用来描述客观事物的一个实体。它由属性（静态特性）和行为（动态特征）构成。

（2）类是面向对象方法的核心

类是对某些具有相同属性的对象的描述，它是对客观世界中一类事物的高度抽象。比如说对一个专业的全部学生的描述。它为属于该类的全部对象提供了抽象的描述，其内部也包括了属性（静态特性）和行为（动态特征）两部分的描述，而对象就是某个类的实例。

（3）对象的属性和行为结合为一个独立实体称为封装

对象的属性一般用数据来表示，而对象的行为用方法或函数来表示，二者结合成一个独立的系统单位，并尽可能隐藏对象的内部细节，这称为一个封装体。封装体对外保留有限的接口与外部发生联系。封装是面向对象方法的一个重要原则。

（4）复杂对象可以由若干简单对象构成

在面向对象程序设计中，将一个复杂对象化解为若干个简单对象的集合，这种关系称为组合。也就是说，一个复杂的对象可以由比较容易理解和实现的简单对象组装而成。

(5) 派生类继承基类的属性和行为——继承

继承是面向对象技术能够提高软件开发效率的主要原因之一。一个新类具有原有类的属性和行为,而且自己还可以具有新的属性和行为,称新类为派生类,原有类为基类,两类之间关系称为继承。继承具有重要的实际意义,它简化了人们对事物的认识和描述。类的继承关系对于软件重用是十分有用的。

(6) 消息传递

在面向对象的方法中,对象之间是使用消息进行通信的。消息是向某个对象提出执行该对象具有的某项服务的申请,这里的消息主要是指对类的成员函数的调用。不同对象之间通过发送消息(调用类的成员函数)向对方提出服务请求。接收到消息的对象,经过解释,然后予以响应,这种通信机制称做消息传递。

(7) 多态性提供了面向对象方法的设计灵活性

多态性是面向对象程序设计中的另一重要特性,简单地说,多态性是指同一个消息被不同类型的对象或相同类型的对象接收时产生不同的操作或行为。多态可以在编译时实现,也可以在运行时实现。

1.2 面向对象语言的核心概念

对象的封装性、继承性和多态性是面向对象方法最重要的三大特征。这三大特征是相互联系的,封装性是面向对象方法的基础,继承性是面向对象方法的技术支持,多态性提供了面向对象方法设计的灵活性。

1.2.1 对象与类

在现实世界中,对象可以是一切有形的事物和抽象的概念。例如:一个教师,一个学生,一本书,一个商店,一个学校,学校的校规,企业的规则,图书馆的借书规则等。从上述举例可以看出对象具有如下特征:

1. 用一个名字来标识一个对象。

2. 每一个对象都会有其状态特征(或静态特征)。

3. 有一组操作用来实现对象的动态特征。

例如:有一个学生对象,其姓名为李林,性别为男,本科,本学期的课程为计算机原理、面向对象程序设计等。在这里,对象名可以为李林,性别、学历、课程名是这个学生的状态特征,也可称为属性(或静态特征)。而要统计其本学期的总分和平均分,就是这个对象要实现的操作,反映出这个对象的动态特征。

面向对象方法中的对象概念与现实世界中的对象概念相似。在面向对象方法中,对象是系统中用来描述客观事物的一个实体,它是用来构成系统的一个基本单位,对象是既包括属性数据(静态状态),又包括作用于属性数据上的一组操作(动态特征)的封装体。也就是说,对象是属性和操作的封装体。属性是用来描述对象静态特征的数据项,操作或行为是描述对象的动态特征的操作序列。在C++语言中,属性称做数据成员,操作或行为称为函数成员。

类是一组相似对象的抽象描述,它为属于该类的全部对象提供了抽象的描述。例如:学生对象,每个学生会有不同的名字、性别、年龄、家庭住址等;一个班的学生在一个学期会学习

相同的课程，可是每门课的成绩却是不一样的。因此，每个学生的总成绩或平均成绩也就有区别。那么，学生的姓名、性别、年龄、家庭住址各门课程都是学生的共同静态属性，而求学生的平均成绩和总成绩就是每个学生需要的共同操作。类是对具有相同属性和相同行为（操作）的一组相似的对象的抽象，或者说，类所包含的属性和服务描述了一组对象的共同属性和行为。这样，我们可以先建立一个学生类，再用学生类创建出不同的学生对象，也就是说，类就是创建某个具体对象的模具或模板。

因此，面向对象方法中的类，是具有相同属性和服务的一组对象的集合，它为属于该类的全部对象提供了抽象的描述，其内部包括属性和行为两个主要部分。把一组对象的共同特征加以抽象并表示在一个类中的能力，是面向对象方法最主要的特征之一。

1.2.2 封装

封装是面向对象方法的一个重要原则，就是把对象的一组属性和一组操作组装在一起，形成一个独立的实体，并尽可能地隐藏对象的内部细节。封装具有两个含义：第一个含义是把对象的全部属性和操作组合在一起，形成一个不可分割的封装体；第二个含义是被封装的对象属性和操作必须通过所提供的与外界联系的公共接口才可以被外界访问，即封装体内的对象属性和对象的操作，外界是无法直接访问的，这便是信息的隐蔽性。

在C++语言中，封装性具体体现在类和对象上，对象的属性（对象的静态特征）由一组数据表示，作用于属性数据上的一组操作（对象的动态特征）是对象的函数成员。类的存取权限分为三种，即私有的、保护的和公有的。在私有域声明的数据成员和函数成员构成了类的私有部分。例如，类的私有部分在封装体外一般是无法访问的，只有通过封装体内的成员函数或类的友元函数才可以访问。只有具有公有权限的数据成员或函数成员才可以作为与外界联系的公共接口由外界访问，称为类的外部接口，外界只能通过公有成员这个接口与类发生联系。可以看出，通过封装，使一部分成员作为类与外界的接口，而将其他成员隐藏起来。这样就达到了成员访问权限的合理控制，使不同类之间的相互影响减小到最低限度，同时数据隐藏增强了数据的安全性。属性数据和操作代码封装为一个可重用的程序模块构成了面向对象程序设计方法的基础，在编写程序时可以有效地利用已有的程序模块，通过外部接口按照特定的访问规则去使用封装好的模块，而不必了解类的实现细节，从而降低了大型软件的复杂性。

1.2.3 继承

继承是面向对象方法的另一个重要概念。在客观世界中，存在着整体和部分的关系，一般和特殊的关系。继承的方法实现了一般与特殊的关系，解决了软件的重用和扩充的问题。

继承是在类、子类以及对象之间自动共享数据和共享操作方法的一种机制。继承性允许程序设计人员在设计新类时，只需要考虑与已有的父类所不同的部分，而继承父类的内容作为自己类的内容。子类不仅可以继承父类的行为（操作），也可以继承父类的属性（数据）。如果父类中的某些行为不适用于子类，则设计人员可以重置这些行为，子类还可以增加自己需要的属性和操作。

举一个例子说明继承关系。将一个人当成对象，对于人的约定（如姓名、性别、年龄、住址等）和对人的操作（提取姓名、输出一个人的各项信息等）已经定义封装在人类（person）中了；