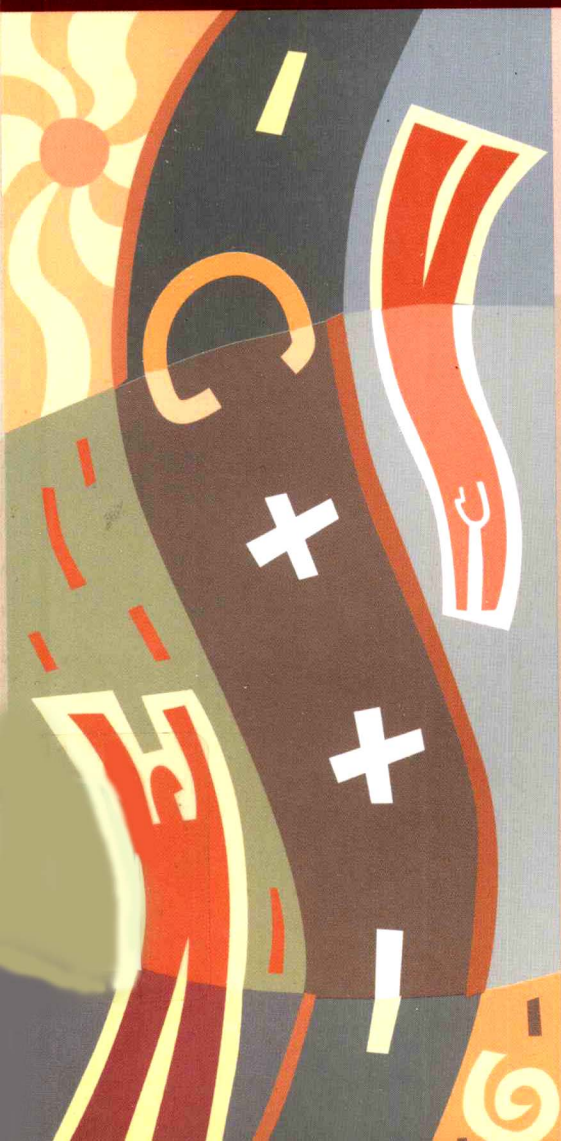
面向21世纪 高职高专计算机专业教材

C++面向对象程序设计

刘丽华 主编



人民交通出版社

面向21世纪 高职高专计算机专业教材

C++ Mianxiang Duixiang Chengxu Sheji

C++面向对象程序设计

刘丽华 主编

人民交通出版社

内 容 提 要

C++语言是目前最流行的面向对象程序设计语言。它既支持传统的面向过程的程序设计方法，也支持新的面向对象的程序设计方法。作为一种混合语言，C++保持了与C语言的兼容，C程序员仅需学习C++语言的特征，就可很快地用C++编写程序。

全书共分15章。第1章至第7章系统讲述了C++语言的基础知识，主要介绍C++程序设计语言、程序结构和过程化基础。第8章至第14章，详尽地讲述了C++语言面向对象的重要特征：类与对象、继承和多类、特殊成员函数、运算符重载，I/O流、模板、异常处理等内容。第15章是专门开辟的"面向对象的程序设计"，旨在培养学生解决实际问题的能力，使学生对C++编程有一个整体认识，并初步掌握实用程序的编制方法及大程序的设计方法，为课程设计打下基础。本书配有丰富的例题，各章均附有与内容相对应的习题。

本书适合作为高职高专学生学习C++语言课程的教材，同时也适合于作为大专院校及各种培训类教材，并可供广大计算机工作者自学之用。

图书在版编目(CIP)数据

C++面向对象程序设计/刘丽华主编.—北京：人民交通出版社，2004.1
ISBN 7-114-04926-9

Ⅰ.C … Ⅱ.刘… Ⅲ.C语言-程序设计
Ⅳ.TP312

中国版本图书馆CIP数据核字(2003)第126130号

面向21世纪高职高专计算机专业教材
C++面向对象程序设计
刘丽华 主编
正文设计：彭小秋 责任校对：王静红 责任印制：杨柏力
人民交通出版社出版发行
(100013 北京和平里东街10号 010 64216602)
各地新华书店经销
三河市宝日文龙印务有限公司印刷
开本：787×1092 1/16 印张：24 字数：589千
2004年3月 第1版
2004年3月 第1版 第1次印刷
印数：0001—3000册 定价：37.00元
ISBN 7-114-04926-9

主　　编：刘丽华（本溪冶金高等专科学校）
副 主 编：彭　斌（江西交通职业技术学院）
参与编写：胡　楠（本溪冶金高等专科学校）
刘　涛（本溪冶金高等专科学校）
律德才（本溪冶金高等专科学校）
郭树岩（本溪冶金高等专科学校）
卢志鹏（本溪冶金高等专科学校）
杜　刚（本溪冶金高等专科学校）

本书策划组成员名单

白　崤　翁志新　张　景　黄景宇

前言 FOREWORD

根据21世纪高等职业教育的新趋势和计算机专业学科建设的要求，结合目前众多高职高专院校的教学计划，人民交通出版社组织全国十几所高职高专院校的多年从事一线教学、实践能力强且具有丰富教材编写经验的教师，编写了这套“面向21世纪高职高专计算机专业教材”，共21本（书目附后），涵盖了高职高专计算机及相关专业的主要课程。在编写过程中认真贯彻了教育部《关于加强高职高专教育人才培养工作的意见》的精神。内容以必需、够用为度，既注重基础知识的讲解，又注意从实际应用出发，满足社会对计算机类专业人才的需求，突出以能力为本位的高等职业教育的特色。

应当说明的是，凡是高等职业教育、高等专科教育和成人高等教育院校的计算机及其相关专业的师生均可使用本套教材。各学校可以根据实际需要，在教学中适当增删一些内容，从而更有针对性地帮助学生掌握计算机专业知识，并形成相关应用能力。

本套教材的出版，将促进高等职业教育的教材建设，对我国高等职业教育的发展产生积极的影响。同时，我们也希望在今后的使用中不断改进、完善此套教材，更好地为高等职业教育服务。

编　者

目 录 CONTENTS

第1章 C++语言概述

计算机软件开发一直被两大难题所困扰:一是如何超越程序复杂性障碍,二是如何在计算机系统中自然地表示客观世界,即对象模型。基于面向对象程序设计的C++语言是软件工程学中的结构化程序设计、模块化、数据抽象、信息隐藏、知识表示、并行处理等各种概念的积累与发展,是20世纪90年代解决上述两大难题极具前途的方法。

面向对象程序设计是软件开发方法的一场革命,它代表新颖的计算机程序设计的思维方法。该方法与通常的结构程序设计十分不同,它支持一种概念,即旨在使得计算机问题的求解更接近人的思维活动,人们能够利用C++语言充分挖掘硬件的潜在能力,在减少开销的前提下,提供更强有力的软件开发工具。

面向对象程序设计是软件系统的设计与实现的新方法。这种方法通过增加软件可扩充性和可重用性,来改善并提高程序员的生产能力,并能控制软件的复杂性和软件维护的开销。当使用面向对象程序设计方法时,软件开发的设计阶段更加紧密地与实现阶段相联系。在软件设计与实现中,当今有许多方法,而面向对象方法是在实践中超越其他许多方法的潜在的大有前途的方法。在各个应用领域中面向对象程序设计都获得了巨大成功。

从目前现状来看,C++和面向对象程序设计,不仅在尖端技术应用领域(如金融和通信)已经立足,而且C++也逐渐地为企业开发人员所接受。C++产品可以给用户一个很好的起点,用户可在此基础上按照接近人的思维活动,按不同的对象类向前发展,C++和面向对象程序设计在企业信息系统中占领一席之地已是迟早之事。

面向对象方法包含了分析、设计和实现的面向对象方法,这部分是当今软件开发最薄弱的部分,面向对象方法对软件系统开发起着关键作用。本书不仅要在面向对象方法,而且还要在面向对象模型和设计方面加以介绍。面向对象模型和设计能更好地加快对问题需求的了解,使设计更加简洁清晰。特别是分析和设计过程中产生的高质量的产品,能极大地减少在开发后期发现的错误,并能显著地改善系统质量。

C++是目前最流行的面向对象程序设计语言。它在C语言的基础上进行了改进和扩充,增加了面向对象程序设计的功能,更适合编制复杂的大型软件系统。这一章我们将引入面向对象的概念,并通过一个简单的C++程序来加深用户对面向对象程序设计方法的理解。

1.1 C++语言的起源和特点

通过名字可以看出,C++语言是从C语言继承来的,但这种继承只是主要表现在语句形式、模块化程序设计等方面。如果从更重要的方面——概念和思想方面来看,C++源于早期的SIMULA语言,因为C++语言的最大特征是支持"面向对象的程序设计"(面向对象的程序

设计的概念见 1.2 节)。

SIMULA 语言被广泛地用于系统仿真,设计它的目的主要是模仿现实世界的真实个体,而使用的主要手段是构造计算机领域的对象来表述现实的客体。由于 SIMULA 语言的应用领域并不十分广泛,更重要的一点是它缺乏强有力的开发工具支持,因此并没有受到很大的重视。随后推出的另外一种面向对象语言 SMALLTALK 也没有取得太大的成功,很多人认为它没有提供给自己足够的灵活性和如同 C 或 BASIC 语言那样丰富的功能,关键原因还在于它和人们早已得心应手的语言并不兼容。比如说,一个 C 程序员可能会对它的新特性退避三舍,因为 C 语言的特性对他是十分熟悉和亲切的,同时 C 语言的确是功能强大的,大多数人不愿放弃这些。

C++的产生正是为了解开这样的一个"情结"。面对越来越大、越来越复杂的系统,使用 C 语言已经感到力不从心了,但 C 语言作为应用领域最为广泛的程序设计语言之一,又不能轻易被抛弃。因此需要一种面向对象的程序设计语言。要求它对 C 语言有很高的兼容性,使得 C 程序员只需在原有的知识基础上进行一定的扩充,就能够方便地进行面向对象的程序设计。

从 1980 年起,贝尔实验室的 Bjarne Stroustrup 博士及其同事开始为这个目标对 C 语言进行改进和扩充。由于这种被扩充和改进的 C 语言的大量特性与类(Class)相关,它最初被开发者称为"带类的 C"。但很快人们就认识到这个称呼太片面了,这个"扩展了的 C"不仅以标准 ANSI C 作为子集,保留了 C 语言的全部精华,同时又吸收了 SIMULA 67,ALGOL 68 和 BCPL 语言的许多特性,它已远远超过了 C 语言。随着这种语言的广泛应用和在各个领域取得成果的增多,它给程序设计带来的全新概念和思想表现出远大的前景,它的开发者因此将 C++这一名字赋予它。

与过去的面向过程的程序设计语言比较,C++的最大特征在于它是面向对象的程序设计语言。所谓对象就是指现实世界中的实体,例如桌子、电视接收机、张三等。具有共同行为和特征的实体的集合,可以被归纳成一类,因此每个对象都是属于某个类的对象,例如,人是一个类,而每个具体的人则是人这个类中的一个对象。面向对象的程序设计是程序设计的一种新思想,该思想认为程序是相互联系的离散对象的集合。面向对象的程序设计语言即是支持这种思想的程序设计语言。

1.2 面向对象的基本概念

在面向对象的程序设计方法出现之前,占据主流的是结构化程序设计方法。对于复杂的问题,结构化程序设计采用模块化、自顶向下逐步求精的设计原则,因此,结构化的程序往往清晰、易读。典型的结构化程序设计语言有 C 语言、Pascal 语言等,著名的 UNIX 操作系统的大部分代码就是用 C 语言编写的。

随着软件技术的发展,需要开发的系统越来越复杂。人们逐渐发现,对于大型软件系统来说,如果采用结构化的设计方法,设计、编程、测试和维护等工作都非常困难,而且有许多问题是结构化设计自身无法解决的。在这种背景下,产生了面向对象的方法,而面向对象的程序设计语言(Object-Oriented Programming Language,简称 OOPL)也应运而生。

那么,什么是面向对象的方法呢?

面向对象方法的出发点和基本原则，就是使分析、设计和实现一个系统的方法尽可能地接近我们认识一个系统的方法。形象一点来说，就是使得描述问题的问题空间和解决问题的方法空间在结构上尽可能地一致。这样说可能太抽象，但随着我们对C++语言学习的深入，会逐步体会到这一点。

下面介绍面向对象方法的几个重要概念，作为后面学习C++语言的基础。

对象(Object)：是由信息和对它进行处理的描述所组成的包，其结构如图1-1所示。

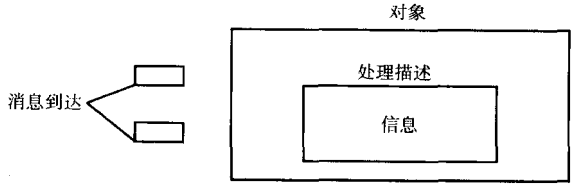


图1-1 对象的图解

• 消息(Message)：是对某个对象进行处理的说明。

方法(Method)：是类似于过程的一个实体，是对对象接受了某一消息后所采取的一系列操作的描述。

• 类(Class)：是对具有共同特征的对象的描述。

• 类的封装(Encapsulation)：封装是把一类对象的状态(用数据表示)和方法(用函数来表示)封闭起来，装入对象体中，形成一个能动的实体。OOPL的封装机制模仿现实生活中的封装技术，把一类对象的数据和函数封闭起来，并提供访问它们的机制。外界只有调用对象的共有成员函数才能和对象交换信息，这样就达到了封装的目的。

• 类的继承(Inheritance)：是指新的类继承原有类的全部数据、函数和访问机制，并可以增添新的数据、函数和访问机制。这样产生的新类叫子类或派生类，原来的类叫父类或基类，这种产生新类的方法叫类的派生，也叫类的继承。如“汽车”是一个类，“轿车”就是它的子类，而某辆实际的轿车就是这个子类的一个对象。

• 多态性：是指相似而实质不同的操作可以有相同的名称。例如，“和”的操作，可以是“整数和”也可以是“矢量和”，在C++中，这两种和的操作都可以简单地称为“和”。C++的多态性使得C++与人的思维习惯更趋一致，用C++编制的程序也更方便阅读。面向对象的方法还有许多特征和概念，如虚函数等，会在后面具体介绍。

1.3 C++对面向对象程序设计的支持

C语言产生于1972年，最早用来编写UNIX操作系统，经过完善与改进后，它发展成为一种通用的计算机语言。1983年进行标准化(ANSI C)后，其发展更为迅速，几乎所有操作系统都提供对C语言的支持。

1985年，AT&T的贝尔实验室在C语言的基础上，吸收了OOPL的特点，形成了面向对象的程序设计语言C++。

C++是一种灵活高效、可移植的面向对象程序设计语言。C++诞生以后，发展极其迅

速，很多公司都研制了自己的 C++ 版本，并推出了多种 C++ 的集成开发环境。

C++ 支持基本的面向对象的概念，如对象、类、方法、消息、子类和继承性以及多态性等。表 1-1 给出了 C++ 对这些概念的命名约定。

C++ 的概念及命名约定　　表 1-1

面向对象的概念	C++ 的命名约定	面向对象的概念	C++ 的命名约定
对象	对象	消息	函数调用
类	类	子类	派生类/子类
方法	成员函数	继承	派生/继承
实例变量	成员		

C++ 程序设计语言在 C 语言的基础上扩充了类、内联函数、运算符重载、变量类型、引用和自由存储管理运算符等语法。我们将在后面通过简单的例子来了解 C++ 语言的特点，在第 2 章、第 3 章再系统地学习 C++ 语言的语法与编程方法。

1.4 C++ 语言与 C 语言的关系

C 语言也诞生在 AT&T 的贝尔实验室，是 1972 年由 Dennis Ritchie 在 B 语言的基础上开发出来的一种高级语言，今天 C 语言的使用已遍及计算机的各个领域。

C 语言有以下三个显著的特点：

(1)它是一种结构化语言，要求一个程序由众多的函数组成，程序的逻辑结构由顺序、选择和循环三种基本结构组成，适宜于大型程序的模块化设计。

(2)它可以部分取代汇编语言，同时具有很高的可移植性，这使得 C 语言程序在保证支持不同硬件环境的前提下，具有较高的代码效率。

(3)它提供了丰富的数据类型和运算，具有较强的数据表达能力，因而在许多不同的场合广泛应用。

总之，C 语言反映了设计者追求高效、灵活，支持模块化程序设计，从而支持大规模软件开发的愿望。

C++ 语言保留了 C 语言设计者的良好愿望，并使得 C 语言语句成为 C++ 语言的一个子集。一般来说，用 C 语言编写的程序可直接在 C++ 编译器下编译。

1.4.1 C++ 语言与 C 语言的主要区别

首先，C++ 提出了类的概念。类是数据和函数的集合，数据用来描述类所属对象的状态，函数用来描述此类对象的行为。例如，大学生代表在大学读书的一类人，即大学生是个类，每个具体的大学生都是这个类中的对象。大学生这个类中的数据可以是学生的姓名、性别、年龄、学校、专业、入学时间等，描述此类对象的行为可以是入学、改换专业、毕业等。

C 语言中的结构只是数据的集合，这种结构也可在 C++ 语言中使用。不同的是 C++ 语言将 C 语言中的“结构”概念扩充成近似于上述“类”的概念，即 C++ 语言中的结构既可以有数据，也可以有函数。

C++ 语言沿用了 C 语言中的结构，概念上没有变化。其次，下列关键字是 C++ 语言新