高等学校计算机教材

刘开军　范 颖　编著

C++程序设计

清华大学出版社

C++程序设计

刘开军　范　颖　编著

清华大学出版社

北　京

内 容 简 介

本书是一本面向广大 C++初学者的基础教程。C++是一种高效的面向对象程序设计语言，同时也支持面向过程的程序设计方法，既能够编写系统软件，也可以编写应用软件，因而受到广大软件开发人员的青睐。作者通过详实的例子由浅入深地阐明了 C++的有关概念、程序设计方法和常见问题的处理方法，适合作为大学计算机专业和非计算机专业的程序设计基础教程，也可以供自学者使用。

本书可分为三个部分。第一部分，从第 1 章到第 5 章，讲述了 C++程序设计语言的特点和面向对象的概念、基本语法、程序设计结构、指针和函数等内容，这部分主要是用面向过程的方法，与 C 语言相似。第二部分，从第 6 章到第 9 章，讲述了 C++对面向对象的支持，讲述了类和对象、继承和派生、多态性和虚函数，以及模板等内容，是面向对象程序设计的核心。第三部分即第 10 章，讲述了 C++中标准文件的处理方法和异常处理方法，可以看作 C++的应用实例，有助于读者深入使用 C++语言。

图书在版编目（CIP）数据

C++程序设计/刘开军，范颖编著. —北京：清华大学出版社，2004

ISBN 7-302-08736-9

Ⅰ.C… Ⅱ.①刘… ②范… Ⅲ.C++—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2004）第 051334 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦

http://www.tup.com.cn　　**邮　　编**：100084

社 总 机：010-62770175　　**客户服务**：010-62776969

责任编辑：刘利民

封面设计：钱　诚

版式设计：张红英

印 刷 者：北京季蜂印刷有限公司

装 订 者：三河市金元装订厂

发 行 者：新华书店总店北京发行所

开　　本：185×260　**印张**：28　**字数**：625 千字

版　　次：2004 年 6 月第 1 版　2004 年 6 月第1 次印刷

书　　号：ISBN 7-302-08736-9/TP·6234

印　　数：1～5000

定　　价：34.00 元

前　言

C++语言是在 C 语言的基础上，吸收面向对象程序设计的概念，发展起来的一种高效实用的程序设计语言。C++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，既可以开发系统软件，也可以开发普通的应用软件，因此得到广大软件开发人员的青睐。随着 C++成为 ANSI 标准，C++迅速成为程序员广泛使用的工具。

C++语言最初是由 AT&T 贝尔实验室的 Bjarne Stroustrup 博士设计、实现的，1980 年首次投入使用，当时它只支持系统程序设计技术和数据抽象；1983 年，C++语言增大了对基本的面向对象程序设计的支持；1985 年，C++第一次走向市场；1987 年至 1989 年，增加了对类属程序设计的支持。C++的标准化工作于 1990 年启动，主要由 ANSI(American National Standard Institute)以及后来加入的 ISO(International Standard Organization)负责，1998 年正式发布了第一个国际标准(ISO/IEC14882)。

C++语言在 C 语言的基础上，增加了面向对象变成、类属编程、数据抽象等技术的支持，还对 C 语言进行了面向过程的扩展。使用 C++进行程序设计可以获得可重用性、可靠性、连续性、访问控制、继承性以及多态性等优势。

本书主要面向广大 C++初学者。考虑到读者可能没有 C 语言基础，本书也较为详尽地讲述了 C++面向过程的部分。如果读者已经很熟悉 C 语言，可以快速浏览一下前 5 章，或者直接跳到第 6 章。本书内容深入浅出，对关键知识点讲解透彻，适合于做大学计算机专业和非计算机专业的程序设计基础教程教材。本书的主要特点是：

（1）兼顾 C++的面向过程特性和面向对象特性。C++从 C 进化而来，是 C 语言的超集，因此 C++语言完全可以替代 C 语言进行面向过程的程序设计。但是考虑到现在还有许多的应用领域仍然使用 C 语言，并且 C 语言部分也是学好 C++的基础，本书在前 5 章讲述了 C 和 C++重叠的部分，学习这部分内容也有助于养成良好的程序书写习惯。

（2）对重要的知识点，本书除用实例讲解外，还插入专门提示，详细讲解难点所在，使读者迅速掌握 C++语言的本质特点。C++语言是中级语言，既有低级语言的高效，又兼有高级语言的易用性，这使得 C++语言具有高度的灵活性，但是部分内容难于理解，尤其是涉及到内存管理的知识点。本书的重点讲解是作者多年来的心得总结，将含糊晦涩的术语，用通俗易懂的语言解说，这有助于读者牢固掌握 C++语言。

（3）重点突出 C++语言的面向对象特点。面向对象是 C++语言的灵魂，也是 C++强大生命力的所在。面向对象程序设计是一种新的程序设计思想，通过数据抽象，实现代码的可重用性和可靠性。C++语言全面支持面向对象程序设计，通过类的概念完成数据抽象，通过虚函数完成多态，通过派生和继承完成继承。但是由于 C++支持多继承和派生，在 C++程序中容易发生二义性等错误，这成为 C++中最难于理解的部分。本书用了大量的篇幅专门讲解 C++语言中的多态性、虚函数和抽象类等内容，方便读者全面理解面向对象的概念。

（4）内容全面，几乎覆盖了 C++语言的所有方面。C++体系非常庞杂，应该说把 C++

所有的方面全都讲透远不是一本书能完成的。本书把主要精力放在C++的常用内容和精华部分上，帮助读者掌握C++语言的核心。但对生僻的知识点也会提及，给读者一个C++体系的全貌，在读者以后需要进一步学习时，可以起到入门的作用。

纵观当前，C++的发展领导了软件程序设计的潮流。在教学上，它以其灵活性和面向对象特性逐渐取代了原来的PASCAL语言和C语言；在科学计算领域，它比Fortran更为可靠和方便；在系统开发方面，C++更是首选，当前的Windows操作系统和Unix/Linux操作系统的内核都是用C/C++编写。目前，软件开发领域出现了一些新的语言，如Java、Delphi等，这些语言支持RAD，能够快速开发出大型应用程序，显示出很强的发展势头。应该说，在开发应用软件（如数据库应用软件、多媒体应用软件等）方面，C++确实有开发速度慢、开发难度大等缺点，但是在开发与系统接合紧密的软件时，C++的地位至今无法替代。

C++的编译器以Borland C++和Visual C++为典型。目前Borland的Borland C++系列已经停止开发，取而代之的是Borland C++ Builder。Microsoft的Visual C++目前已经推出了8.0（即Visual C++.Net）。需要说明的是，不论是Borland C++ Builder还是Microsoft Visual C++都不是纯粹的C++语言的“编译器”，它们都是IDE(Integrated Development Environment)，提供从源文件编辑到程序调试的全程支持。编译器只是IDE的一个部分，并且这个部分总是在后台工作，初学者很难注意到它的存在。

学好C++语言是学习其他语言的基础，新出现的语言（如Java、C#等）都借鉴了C++的精华，语法上很像C++，学完C++语言之后再学习这些语言就很容易入门。本书除了向读者讲解C++语言，还希望读者通过学习本书，能够养成良好的程序书写习惯，从开始接触编程就保持严谨的编程风格。

作　者

2004年4月7日

目 录

第 1 章　C++语言概述

C++语言是一种优秀的面向对象程序设计语言，它在 C 语言的基础上发展而来，但它比 C 语言更容易为人们学习和掌握。C++以其独特的语言机制在计算机科学的各个领域中得到了广泛的应用。面向对象的设计思想是在原来结构化程序设计方法基础上的一个质的飞跃，C++完美地体现了面向对象的各种特性。

在学习 C++之前，我们先介绍 C++的历史、面向对象的基本概念以及一个基本的 C++程序的结构，最后讲解在 Microsoft Visual C++ 6.0 下如何编写 C++程序，并编译执行。本章的学习重点是：

C++的特点
面向对象的基本概念
C++程序的结构和实现

1.1　C++语言的历史

1.1.1　C++的出现

自1946年世界上第一台电子计算机问世以来，计算机科学及其应用发展十分迅猛，计算机已将人类带入了一个新的时代——信息时代。计算机由硬件系统和软件系统两大部分构成，硬件是指计算机的物理设备，而软件可以说是计算机的灵魂，有了软件，计算机才能灵动起来，成为一台真正的“电脑”。所有的软件，都是用计算机语言编写的。计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的历程。现在各种语言种类繁多，总的来说可以分为下述几类。

1．汇编语言

汇编语言由机器语言发展而来，两者都属于低级语言。机器语言编写的程序代码是由二进制的“0”和“1”组成的，可由机器识别的指令序列。机器语言直接面向计算机硬件，人们很难读懂，因而现在几乎不用了。汇编语言将CPU指令用英文单词代替，将机器指令翻译成人们可以读懂的指令形式。汇编语言是现在使用效率最高的的计算机语言，功能也最为强大。但用汇编语言编写的程序很长，工作量极大。现在汇编语言只用于在功能较弱的处理机（如单片机）上编写程序，或用在对效率要求很高的程序段中。

2．解释型语言

解释型语言编写的程序不会被全部翻译成机器指令形式而后执行，它由一个解释程序实时地处理每个要执行的语句。由于对于程序中的循环语句要反复解释多次，程序翻译成

机器语言后难以有效地优化，所以解释型语言的效率都不高。典型的解释型语言是COBOL、BASIC语言等。

3．编译型语言

现在应用最为广泛的基本上都是编译型语言，例如 Fortran、Pascal、C、C++、Java、Delphi 等。编译型语言通过编译程序将程序员编写的程序代码翻译成机器指令，生成可执行文件。编译型语言编写的程序执行效率很高，编译程序的构造也较解释程序简单，但一般情况下其语法的限制比解释型语言更为严格。

注意：计算机语言（程序语言）的分类。除机器语言和汇编语言属于低级语言之外，其他语言都属于中高级语言（中级和高级的划分并不是非常严格）。所有的语言最终都要翻译成机器代码后才能由计算机执行。“翻译”可分为解释型和编译型两种，解释型语言（如 Basic）是边执行边翻译，编译型语言则是在全部程序代码都翻译成机器代码、生成目标程序后才开始执行。

C 语言是一种高效的编译型结构化程序设计语言。C 语言最早由贝尔实验室的 Dennis Ritchie 在 B 语言的基础上开发出来，并于 1972 年在一台 DEC PDP-11 计算机上首次实现。C 语言是作为 UNIX 操作系统的开发语言开始为人们所接受的，现代的系统级软件基本上都是用汇编语言和 C 语言编写的。

C 语言通常称为中级计算机语言。中级语言并没有贬义，不意味着它功能差、难以使用或者比 BASIC、Pascal 那样的高级语言原始；也不意味着它与汇编语言相似，会给使用者带来类似的麻烦。C 语言之所以被称为中级语言，是因为它把高级语言的成分同汇编语言的功能结合起来了。

在过去 20 年里，C 语言已经能够应用于绝大多数类型的计算机上了，同时 C 语言的发展也导致不同的 C 语言版本的出现。这些不同版本的 C 语言通常是不兼容的。为了明确定义一种与机器无关的 C 语言，1989 年美国国家标准协会制定了 C 语言的标准（ANSI C）。C 语言具有以下优良的特点使得它得以风靡全球：

（1）由于 C 语言的严谨设计，使得用 C 语言编写的程序具有很好的可移植性。一般认为 C 语言与硬件无关。

（2）语言简洁、紧凑，使用方便、灵活。与其他语言相比，用 C 语言编写的代码更为简练，程序的书写更为自由。

（3）C 语言有极为丰富的数据类型和运算符。C 语言提供指针，可以直接访问内存，能进行位操作，从而使其能够胜任开发操作系统的工作。

（4）生成的目标代码质量高，程序执行效率高。各种要求较高的算法和系统软件大都用 C 语言编写。

C 语言在盛行时也暴露出了它的局限性：

（1）C 语言类型检查机制较弱，这使得程序中的一些错误不能在编译时被发现。

（2）C 语言本身几乎没有支持代码重用的机制，这使得各个程序的代码很难为其他程序所用。

（3）对大型的软件项目，程序员很难控制程序的复杂性。

为解决日益增长的软件需求，避免 C 语言的不足，1980 年贝尔实验室的 Bjarne Stroustrup 开始对 C 语言进行改编，1983 年正式命名新的语言为 C++语言，在经历了 3 次 C++版本的修订后，于 1994 年制定了 C++的标准 ANSI C++标准的草案。目前 C++仍在不断的发展中。

1.1.2　C 和 C++的关系

C++包含了整个 C，C 是建立 C++的基础，因而可以认为 C 是 C++的一个子集。C++包含了 C 的全部特征、属性和优点，同时增加了面向对象编程的完全支持。

1．C++保持与 C 语言的兼容

这种兼容性最明显的表现是大部分 C 程序不需修改即可在 C++的环境下编译执行。用 C 语言编写的许多库函数和应用软件都可用于 C++。一个 C 语言的程序员可以很快成为一名 C++程序员，只要掌握了 C++中面向对象的成分。

但是，这种兼容性使得 C++不是一种纯粹的面向对象程序设计语言。C 语言是一种面向过程的结构化程序设计语言，C++与 C 兼容就意味着 C++也要支持面向过程的程序设计。C++中的数据并不全部都包装成对象，允许用户使用简单数据类型和不属于任何类的函数和数据。这种特性使得 C++比一般的面向对象语言更灵活，但也容易造成风格上的紊乱，使得初学者感到困惑。

2．C++增加了很多新的概念

C++保持了 C 语言的简洁、高效、灵活等优点，同时又对 C 语言的不足和问题作了很多改进：

（1）增加了一些新的运算符，使得 C++应用起来更为方便灵活，例如：::，new，delete，->* 等，这使得 C++表达能力更强；并且在 C++中允许运算符重载，程序员可以定义已有运算符在特定对象上的运算规则，另外还可以定义新的运算符。

（2）C++语言更为灵活。C++解除了 C 中对变量说明的限制。在 C 语言中，函数中的变量必须在执行语句前说明；而在 C++中，可以在函数中的任何位置说明变量，甚至可以在语句内说明变量。标识符的作用域种类更多，变量的生存期最小可以是块级。输入和输出可以采用流方式，形式更为简洁。函数中允许设置默认函数参数，函数调用更为方便，尤其是当函数参数较多时。

（3）引进了引用类型。引用是对指针的一种改进，这样函数的参数传递就有传值和传地址两种方式，同时给函数返回值带来很大方便。

（4）支持面向对象编程。C++全面支持面向对象的各种技术，支持类和对象。允许类的继承、虚函数和纯虚类、函数重载。这些措施提高了编程的灵活性，减小了程序员的负担。又引进了内联函数的概念，提高了程序的效率。

3．C++是一种面向对象程序设计语言

C++和 C 的本质差别在于：C++是面向对象的，而 C 是面向过程的。因此，C++是在

对 C 语言改进的基础上，又增添了支持面向对象的特性。但是 C++并不是完全抛开了 C 的成分，面向对象是对原来设计方法的一种发展。

在本书中前 5 章讲述的内容主要是与 C 语言兼容的部分，其后各章主要讲述面向对象的编程方法。对于 C 语言程序员可以直接从第 6 章开始学习。

1.2 面向对象程序设计的基本思想

1.2.1 面向对象的由来和发展

20 世纪 60 年代的 Simula 67 是面向对象语言的鼻祖，首次明确提出了类和对象的概念。对象代表着待处理问题中的一个实体，在处理问题过程中，一个对象可以以某种形式与其他对象通信。从概念上讲，一个对象是既包含数据又包含处理这些数据操作的一个程序单元。类用来描述特性相同或相近的一组对象的结构和行为。该语言还支持类的继承，可将多个类组成为层次结构，进而允许共享结构和行为。

后来出现的 Smalltalk 语言是第一个比较成功的面向对象语言，对后来面向对象语言的发展产生过重大影响。该语言丰富了 Simula 中类和对象的概念，信息也更加隐蔽，程序设计就是向对象发送信息。

20 世纪 80 年代以后，面向对象的程序设计语言广泛应用于程序设计，并且有许多新的突破。特别是随着操作系统和软件项目日益庞大，人们日益需要一种更高效的开发方式，这更加推动了面向对象语言的发展。

1.2.2 面向对象的要素

面向对象的系统包含了 3 个要素：对象、类和继承，这 3 个要素反映了面向对象的传统观念。面向对象的语言应该支持这 3 个要素。首先，应该包括对象的概念。对象是状态和操作的封装体，状态是存储操作结果的。满足这一点的语言被认为是基于对象的语言。其次，应该支持类的概念和特征，类是以接口和实现来定义对象行为的样板，对象是由类来创建的。支持对象和类的语言被认为是基于类的语言。最后，应该支持继承，已存在的类具有建立子类的能力，进而建立类的层次。支持上述 3 个方面的语言称为面向对象的语言。按这一标准来衡量，C++是面向对象的语言。

下面将对象、类和继承这些面向对象方法中特别重要的概念解释一下。在本书后面讲解 C++语言的过程中还会反复讲解这些概念，因为它们是理解和掌握面向对象程序设计语言的关键。

1．对象

从概念上讲，对象代表着正在创建的系统中的一个实体。例如，在一个学校管理系统中，像学生、教师、成绩单等都是对象，这些对象对于实现系统的完整功能都是必要的。

从实现形式上讲，对象一个属性（状态）和操作（方法或行为）的封装体。属性是由

对象中变量的内容和值定义的，例如学生有年龄、性别、入学日期等属性。各个对象的属性值互不相同。操作是一系列的实现步骤，它能够完成特定的功能，例如对学生可以有选课、毕业等操作。在C++中，对象的状态由成员变量的值表示，操作由对象的成员函数完成。

对象实现了信息隐藏，对象与外部是通过操作接口联系的，操作的具体实现外部是不可见的。封装的目的就是阻止非法的访问，操作接口提供了这个对象的功能。

对象是通过消息与另一个对象传递信息的，每当一个操作被调用，就有一条消息被发送到这个对象上，消息带来将被执行的这个操作的详细内容。在 C++中，向对象发送消息就是调用对象的成员函数，从而获取对象的状态信息或是对对象的状态进行修改。

2．类

类是对象的模板，它包含所创建对象的状态描述和方法的定义。类的完整定义包含了外部接口和内部算法以及数据结构的形式。

由一个特定的类所创建的对象被称为这个类的实例，因此类是对象的抽象及描述，它是具有共同行为的若干对象的统一描述体。

类是抽象数据类型的实现。一个类的所有对象都有相同的数据结构，并且共享相同的实现操作的代码，而各个对象有着各自不同的状态，即私有的存储。因此，类是所有对象的共同的行为和不同状态的集合体。

3．继承

类提供了说明一组对象结构的机制，再借助于继承扩充类的定义方式，从而体现代码可重用的优越性。

继承提供了创建新类的一种方法，这种方法就是说，一个新类可以通过对已有类进行修改或扩充来满足新类的要求。新类共享已有类的行为，而自己还具有修改的或额外添加的行为。因此，可以说继承的本质特征是行为共享。

从一个类继承定义的新类将继承已有类的所有方法和属性，并且还可以添加所需要的新的方法和属性。新类被称为已有类的子类，而已有类称为父类，又叫基类，新类又叫派生类。

注意：面向对象的 3 个要素是对象、类和继承。在 C++中，类是主要的编程对象，程序员需要设计类的成员变量和成员函数，以及类与外界的接口。对象即类的实例化，类是对象的模板。同一个类的对象具有相同的行为，但状态可以不同。继承是创建新类的一种重要方法，派生类按照继承类型的不同可以从基类继承到不同的成员，同时派生类还可以添加新的成员，通过这种继承和扩充得到新的类。

1.2.3　面向对象的技术

面向对象程序设计的本质是把数据和处理数据的过程当成一个整体——对象。面向对象程序设计的实现需要封装和数据隐藏技术，以及继承和多态性技术。