杨云 编著

C++程序设计

C++ CHENGXU SHEJI

陕西科学技术出版社

C++程序设计

杨　云　编著

陕西科学技术出版社

内 容 简 介

C++是当今流行最为广泛的面向对象的程序设计的语言。

本书全面、系统地介绍了C++语言的基本概念、语法、结构和编程技巧以及面向对象程序设计的方法。第1～4章介绍了C与C++的共同点及C++对C的扩充，第5～10章介绍了面向对象的基本特性，包括类和对象、继承与派生、多态性、模板及I/O流库等。书中所用实例由易到难且具有实用性。

本书语言简洁，通俗易懂，层次分明，偏重应用，每章后均有挑战性问题且附有答案。可作为大学本科、专科学生及研究生面向对象程序设计的基础教材，也可供C++语言爱好者自学和程序设计人员参考。

图书在版编目(CIP)数据

C++程序设计/杨云主编．—西安：陕西科学技术出版社，2003.3
ISBN 7-5369-3620-6

Ⅰ.C...　Ⅱ.杨...　Ⅲ.C语言—程序设计
Ⅳ.TP312

中国版本图书馆CIP数据核字(2003)第012975号

出 版 者： 陕西科学技术出版社
西安北大街131号　邮编710003
电话(029)7211894　传真(029)7218236
http://www.snstp.com
发 行 者： 陕西科学技术出版社
电话(029)7212206　7260001
印　　刷： 陕西科技大学印刷厂
规　　格： 787mm×1092mm　16开本
印　　张： 22.125
字　　数： 450千字
印　　数： 1—1000
版　　次： 2003年3月第1版
2003年3月第1次印刷
定　　价： 25.00元

前　言

传统的面向过程的程序设计语言，在编写中小型软件还是显得非常方便的，但是一到大型软件中，也就力不从心了。一个是数据易造成混乱，另一个就是软件的重用性差，造成了大量人力、物力的浪费。在这样的背景下，面向对象的程序设计语言便应运而生了，它全新的设计思想给人们开发大型软件带来了很大的方便，从某种意义上说，面向对象的程序设计语言是为开发大型软件而出世的。

C＋＋语言是目前面向对象程序设计语言中使用最广泛的一种，它是在 C 语言的基础上发展起来的，在具有其自身面向对象编程特征的同时，全面兼容 C 语言，这种平滑的过渡一方面可以照顾 C 语言程序员，另一方面又可使早期在 C 语言编程上的投资不至于立即失效。

本书全面地讲解了 C 语言和 C＋＋语言的共性以及 C＋＋语言面向对象编程的基本特征。全书共分为十章，前四章主要讲述 C＋＋中的 C，其中包括 C 语言的一些基本语法及概念，第五到第十章较为详细地讲述了面向对象编程的一些特性，包括类、对象、继承、派生、多态性、运算符重载、虚函数、模板、流等。

我们力求用简洁明快的语言阐明 C＋＋语言的基本知识，同时书中所用实例也做到了精简但又不乏实用性与趣味性。因此，书中前面讲过的知识，后面一般不会再讲，读者若是把前面的知识遗忘请回头再复习一遍。

为了照顾各方面的读者，本书讲授基本知识点时，所用实例都较为浅显，在基本知识授完的基础上，各章都有深入性的例题详解。同时，课后某些习题也具有一定的挑战性。

本书全部例题都在 Visual C＋＋6.0 上运行过。在编写过程中，我们查阅了大量国内外同类书籍，在此对这些书籍的作者和译者表示衷心的感谢。由于水平有限，书中难免出现疏漏和错误，恳请广大读者批评指正。

编　者

2002 年 10 月

目　录

第 1 章　C++语言概述

C++是一门广泛应用的程序设计语言，它在 C 语言的基础上扩展了面向对象程序设计的特点。学习本章后，要求理解面向对象程序设计的基本概念，了解 C++语言对面向对象程序设计方法的支持，掌握 C++程序的开发过程及 C++程序的基本结构。

1.1　面向对象程序设计的基本思想

面向对象方法是正在快速发展并成为主流的软件系统开发方法，包括面向对象的系统分析、面向对象的系统设计和面向对象的程序设计。系统分析、系统设计和程序设计是软件开发的三个基本步骤，但从面向对象方法的发展过程来看，面向对象程序设计是其中最先发展起来的，在技术上也最为成熟。

面向对象程序设计是一种围绕真实世界的概念来组织模型的程序设计方法，它采用对象来描述问题空间的实体。对象能体现现实世界物体的基本特征的抽象实体，反映在软件系统中就是一些属性和方法的封装体。从程序设计角度来看，对象就是“数据+作用于数据上的操作（或方法）”。

1.1.1　C++ 的发展史

C++ 源于 C 语言，1972～1973 年间，美国贝尔实验室的 D.M.Ritchie 设计出了 C 语言。最初的 C 语言只是为了描述和实现 UNIX 操作系统提供一种工作语言而设计的，由于其简单、灵活和使用方便等特点，C 语言很快就被用于编写各种不同类型的程序，从而成为世界上最为流行的语言之一。

但是，C 语言是一个面向过程的语言。随着软件开发技术的进步，程序员们最终实现把数据和施加在其上的操作结合起来，会得到更易于理解的程序，由此产生了面向对象的程序设计思想。于是，20 世纪 80 年代初，贝尔实验室的 Bjarne Stroustrup 博士设计并实现了 C 语言的扩充、改进版本，最初称为“带类的 C”，多次修改后起名为 C++，于 1993 年正式取名为 C++。C++改进了 C 的不足之处，支持面向对象的程序设计，改进的同时保持了 C 语言的简洁性和高效性。

目前，C++语言越来越受到重视并已得到了广泛的应用，许多软件公司为 C++设计编译系统，提供不同应用类别的类库和越来越方便的开发环境，如：Microsoft 公司的 Visual C++ 开发工具。利用 C++设计并实现应用系统变得日益简单和快捷。因此我们每一个编

程人员对C++的学习都要引起重视。

1.1.2 面向对象程序设计的基本概念

1 什么是对象

对象是变量和相关方法的软件组合。对象是理解面向对象技术的关键。你可以看看你的周围，会看到许多实现对象的例子：你的狗，你的桌子，电视机，自行车等。

这些实现对象都有两个性质：状态和行为。例如：自行车有状态（变速齿轮、两个轮子等），行为（刹车、加速和减速）。

软件的对象是仿照现实对象建立的，它们也有状态和行为。软件的对象在一个或多个变量中维护它的状态，变量是一个由标识符命名的数据项。软件对象用方法实现它的行为，方法是与一个对象相关的函数。

对象的核心是对象的变量的组成。方法围绕在对象核心的周围并将对象核心对于程序中的其他对象隐藏起来，将对象的变量置于它的方法的保护之下，这被称为封装。（我们将在后面面向对象的三大特征中讨论它）。

2 什么是消息

单独一个对象一般没什么作用。相反，对象常常在一个包括许多其他对象的大程序中作为组件出现。通过这些对象之间的交互，程序员可以实现更高级的程序功能和更复杂的行为。例如，自行车只在另一个对象与它进行交互时才有用，也就是说有一个骑车的人骑它时。

软件对象之间的相互交互和通信是通过相互发送消息进行的。当一个对象希望其他的对象执行时，这个对象发送一个消息给其他对象。

消息在程序中提供了两个重要的好处。

对象的行为通过它的方法显露出来，所以消息传送支持对象之间所有可能的交互。

对象不需要在相同的进程内，甚至不需要在相同的机器上，也能够相互发送和接收消息。

3 什么是类

类是蓝图或原型，它定义了所属的某种类的对象所具有的变量和方法。

在现实世界中，常常有相同类型的许多对象。例如：生产汽车时厂家利用了“汽车具有共同的性质”这一情况，用同一张蓝图生产了许多汽车。如果每生产一辆汽车就要画一张蓝图，那就非常低效了。

在面向对象的软件中，也可以有许多相同类型的对象，它们具有相同的性质，如：矩形，雇员记录，视频片断等等。类提供了可重用性，如：汽车厂家可以反复的使用相同的蓝图来生产大量的汽车，软件程序员可以反复使用相同的类（相同的代码）创建许多对象。

4 什么是抽象

抽象就是从许多事物中舍弃个别的、非本质的特征。抽象原则具有两方面的意义：一方面有时问题很复杂但我们并不需要了解和描述它们的一切，只需要分析研究其中与系统目标有关的事物及其本质特性；对于那些与系统无关的特征和许多具体的细节，即使有所了解，也应舍弃。另一方面通过舍弃个体在细节上的差异，抽取其共同特征而得到一批事物的抽象概念。面向对象程序开发主要有过程抽象和数据抽象。

(1) 过程抽象

过程抽象是指任何一个完成确定的操作序列，其使用者都可以把它看做一个单一的实体，

尽管实际上它可能是由一系列更低的操作完成的。

运用过程抽象，软件开发者可以把一个功能分解为一些子功能；如果子功能仍然比较复杂可以进一步分解。这使得开发者可以在不同的抽象层次上考虑问题，在较高的层次上思考时可以不关心较低层次的实现细节。过程抽象不是面向对象的界限在全系统的范围内进行功能的描述。但是过程抽象对于在对象范围内组织对象的服务是有用的。

(2) 数据抽象

数据抽象是根据施加于数据之上的操作来定义数据类型，并限定数据的值只能由这些操作来修改和观察。数据抽象是面向对象分析的核心原则。它强调把数据和操作结合为一个不可分割的系统单位，对象的外部只需要知道它做什么，而不必知道它如何做。

5　什么是继承

特殊的对象拥有其一般的全部属性与服务，称作特殊类对一般类的继承。

继承具有重要的实际意义，在软件开发过程中，在定义特殊类时，不需要它的一般类已经定义过的属性和服务重复的书写一遍，只需要说明它是某个类的特殊类，并定义它自己的特殊属性和服务。

继承对于软件的复用是很有益的。在开发一个系统时，使特殊类继承一般类，这本身就是软件的复用，然而其复用意义不仅在于此。如果把用面向对象方法开发的类作为一般类，通过继承而实现复用，从而大大的扩展了复用范围。

1.1.3　面向对象程序设计的三大特征

1　封装性

封装是面向对象方法的一个重要原则。它有两个涵义：把对象的全部属性和全部服务结合在一起，形成一个不可分割的独立单位；尽可能的隐藏对象的内部细节，对外形成一个边界，只保留有限的对外接口使之与外部发生联系。这主要是指对象的外部不能直接的存取对象的属性，只能通过几个容许外部使用的服务与对象发生联系。

封装的信息隐蔽作用反映了事物的相对独立性。当站在对象角度观察一个对象时，只要注意它对外呈现什么行为，而不必关心它们的内部细节。规定了它们的职责之后，就不应该随意从外部插手去改动它的内部信息或干预它们的工作。封装的原则在软件上反映的是：要求对象以外的部分不能随意存取对象的内部数据，从而有效地避免了外部错误对它的“交叉感染”，使软件的错误局部化，因而大大地减少了差错和排错的难度。

2　继承性

继承提供了创建新类的一种方法。例如：当我们继承了上一辈的财产时，我们可以把喜欢的东西留下，不喜欢的送走或者卖掉，看见其他中意的还可以买来放在家里。与此类似，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，而且还能自己修改或者添加行为。

继承对于软件的复用是很有益的。在开发一个系统时，使特殊类继承一般类，这本身就是软件的复用，然而其复用意义不仅在于此。如果把用面向对象方法开发的类作为一般类，通过继承而实现复用，从而大大的扩展了复用范围。

3　多态性

多态即“一个接口，多种算法”。本质上有一个用于一般类活动，而具体活动的选择是由

涉及的数据类型决定的。对象的多态性是指在一般类中定义的属性或服务被特殊类继承之后，可以具有不同的数据类型或表现出不同的行为。这使得同一属性或服务名在一般类及各个特殊类中具有不同的语义。

1.2 C++ 语言程序的特点

1.2.1 C++与C语言的关系

C语言是C++的一个子集，C++包含了C语言的所有功能。

1 C++保持与C的兼容

兼容是指C 语言程序不需要修改就可以在C++语言环境中运行。目前的C语言编写的库函数和应用软件都可以用于C++。如果已经掌握了C语言，在学习C++时只要再了解并掌握C++的新特性就可以了。C++ 不是一个单纯的面向对象的程序设计语言，因为C语言是面向过程的语言，所以C++ 也支持面向过程的程序设计。由此可以看出C++把两种不同的程序设计风格融于一体。我们特别要注意的是，不能用面向过程的思维方法去学习面向对象的技术，然而，在面向对象的程序设计中，对于一个作用函数内部的实现时，要用到面向对象的方法。

2 C++对C作了很多的改进

C++ 保持了C的简洁与接近汇编的特点，同时又有了重要的改进。

(1)增加了一些运算符，如： new，delete， :: 等。

(2)改进了类型系统，增加了限制。C++规定数据类型转换采用强制转换，规定函数的说明必须采用原型；对于缺省函数作了限制；增加了编译系统检查的能力。

(3)引进了引用概念，将引用作为函数参数为程序设计带来很大方便。

(4)容许函数重载，容许函数设置缺省值参数，这些措施提高了程序设计的灵活性；引入了内联函数的概念，提高了程序设计的有效性。

(5)变量更加灵活。在C语言中仅容许变量存在于函数体内或分段程序内，而且对变量必须先说明后使用，不能交叉使用。C++打破了这一规定，可根据要求对变量进行说明。

1.2.2 C++支持面向对象程序设计方法的特点

C++语言的主要特点在于它支持面向对象程序设计。下面说明它和面向对象有关的一些特征。

1 类和数据封装

C++中的类对面向对象程序设计的基本支持，类是数据抽象和信息隐藏的工具，对象是类的具体化、实例化。对象的值和相关的操作被封装在一个类定义中。对象可以被说明为给定类的变量，对象之间可以通过发送或接收消息相互联系，接收消息的对象通过调用类的方法来实现相应的操作。

2 构造函数和析构函数

类可以包含一组构造函数和一个析构函数。构造函数是在每次创建一个特定的对象时由

C++自动执行的类成员函数，析构函数是在特定的类的对象被撤销时自动执行的类成员函数。类的构造函数保证了在类的对象生成时可以自动进行初始化，类的析构函数保证对类的对象正常清除。

3　访问限制和信息隐藏性

类的成员有公有、私有和保护三种控制权限，它们具有不同的访问限制。声明为私有成员只能由类自己的函数访问；保护成员可以由该类及其派生类的成员函数访问；公有成员构成了类的界面，容许所有的函数访问。通过将成员函数设置为具有不同的访问权限，实现了信息隐藏。

4　对象和消息

对象是面向对象程序设计的基本单元，通过向对象发送消息来实现对象的操作，对象根据消息的内容调用相应的方法。C++中的消息传送采用类似函数的调用机制来实现。

5　友元

友元是 C++面向对象的另一个重要特征。通常类的私有成员禁止该类以外的函数和类直接访问，而友元机制容许有选择的突破这些限制。只要在类定义中声明非成员函数为该类的友元函数或其他类为该类的友元类，则这些友元可以直接访问类的私有部分或保护部分。

6　运算符重载和函数重载

C++容许为已有的函数和运算符重新赋予了新的含义，使它们可以用于用户所希望操作的对象。运算符重载和函数重载使得我们能用更自然的表现方式实现对象的操作，提高程序的可读性。

7　继承和派生类

在程序中定义类时，会出现许多两个或多个类享有相似的情况。这时，不必在每个类中复制这些成员，而可以定义一个包含它们公共成员的基类，然后通过继承，从基类派生出其他类。通过继承的类，为其增加新的操作和成员，改写继承类的部分内容，可以得到更实用的类。 派生类的引入有力的支持了面向对象的程序设计思想。

8　虚函数

C++中的虚函数可以支持动态联编，从而也支持多态性。多态性容许在设计中使用高级抽象。它使得更高层次代码只需要写一次，而可以通过提供不同底层服务来满足复用的要求。这样可以大大提高代码复用性。

1.3　简单的 C++程序介绍

首先介绍几个简单的 C++程序，然后从中分析 C++程序的基本结构。

例 1.1

```
#include  <iostream.h>          //头文件
void main()                     //主函数
{
    cout<<"this is a C++ program. \n";  //双引号内的字符串原样输出，
}                                       //"\n" 表示回车换行
```

运行该程序，屏幕显示如下信息：

```
this is a c++ program.
```

例 1.2

```
//求两个整数之和
#include <iostream.h>
void main()
{
    int a,b,c;              //定义三个整型变量
    cin>>a>>b;              //输入两个整数分别给 a 和 b
    c=a+b;
    cout<<"c=a+b="<<c <<endl;  //输出变量 c 的值，endl 输出回车换行
}
```

运行该程序，光标停在屏幕上等待用户从键盘输入数据

1 2 (空格分隔，回车键结束，1 给了 a，2 给了 b)

结果为：

```
c=a+b=3
```

例 1.3

```
#include <iostream.h>
float sum(float x,float y)  //定义求和子函数，函数名为 sum，函数值为实型
{
   float z;                 //形式参数为 x，y
   z=x+y;
   return z;                //将 z 值返回到主调函数调用点
}
void main()
{
   float a,b;               //定义两个实型变量
   cin>>a;cin>>b;           //输入两个实数分别给 a 和 b
   float c=sum(a,b);        //调用 sum 子函数，将被调函数的返回值赋给 c
   cout<<"a+b=";
   cout<<c;
   cout<<endl;
}
```

运行程序，从键盘输入两个实数。

1.2 5.4

结果为：

```
a+b=6.6
```

C++ 程序的基本结构

一个 C++ 程序是由若干个文件(.cpp)和预处理语句组成的，而每个文件是由函数构成