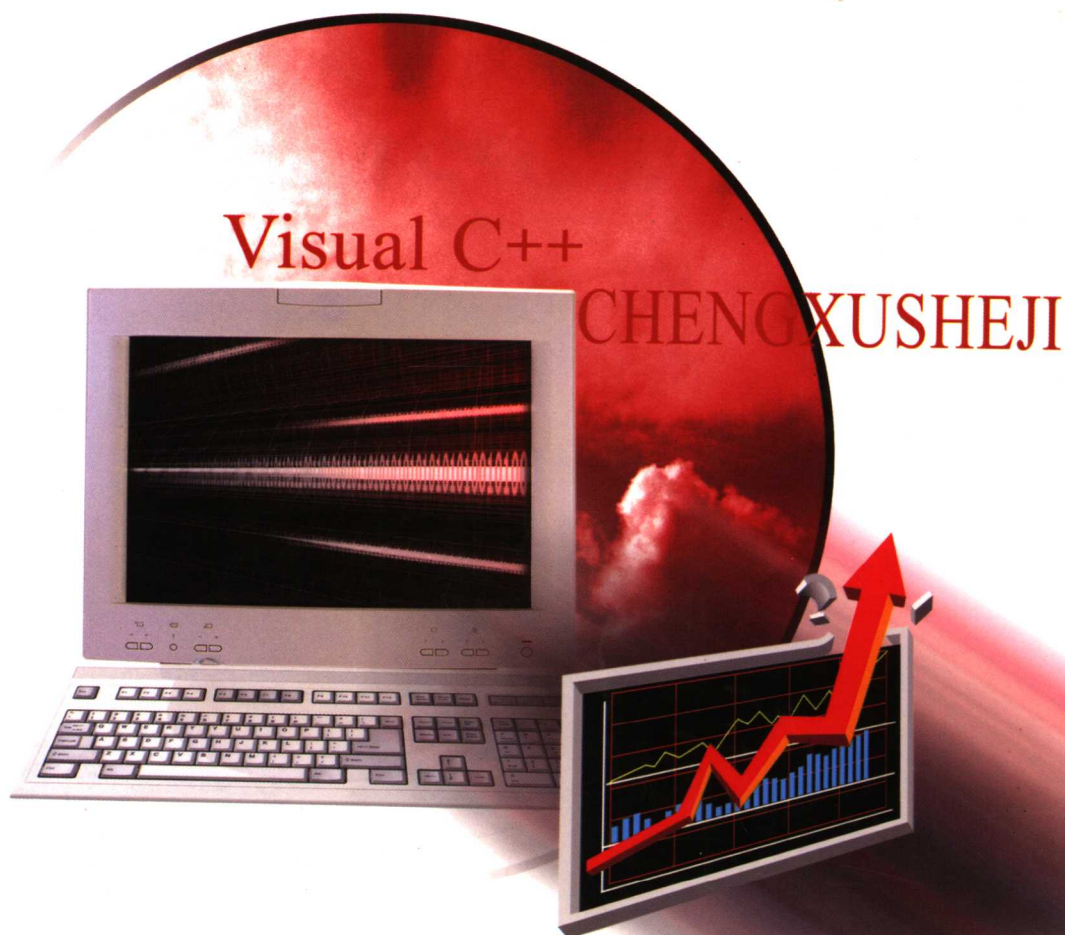


十五规划计算机系列教材



# Visual C++程序设计

主 编：张岳新

副主编：刘宇君 马 锐

兵器工业出版社

# Visual C++程序设计

主 编 张岳新

副主编 刘宇君 马 锐

主 审 张 宏

兵器工业出版社

## 内 容 简 介

C++是一种面向对象的程序设计语言。本书以没有程序设计基础，直接学习C++语言的读者为对象，重点介绍了C++语言的概念、语法、程序设计的思想和面向对象的程序设计方法。为了便于读者学习，每一章后面备有不同类型的习题。

本书的前8章讲述VC++语言的基础，后7章介绍面向对象的程序设计，最后一章介绍MFC程序设计的基本方法。本书由浅入深，通俗易懂，重点突出，重于实用。本书可作为大专院校理工科类学生的C++语言程序设计的教材，也可作为自学C++语言的教材。

### 图书在版编目(CIP)数据

Visual C++程序设计 / 张岳新主编. —北京:  
兵器工业出版社, 2004. 1  
ISBN 7-80172-171-3

I. V... II. 张... 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(2004)第000777号

出版发行：兵器工业出版社  
责任编辑：张小洁  
责任技编：魏丽华  
邮编社址：100089 北京市海淀区车道沟10号  
经 销：各地新华书店  
印 刷：军事医学科学院印刷厂  
版 次：2006年2月第1版第3次印刷  
印 数：3501—6500

封面设计：李 晖  
责任校对：王 绛 全 静  
责任印制：王京华  
开 本：787×1092 1/16  
印 张：25  
字 数：634千字  
定 价：35.00元

(版权所有 翻印必究 印装有误 负责调换)

## 计算机系列教材编辑工作委员会

主 任：闫达远

副主任：胡星光 李晓梅

委 员：（按姓氏笔画为序）

马星国 孔令德 王复兴 王 琰 李凤霞

李 梁 张 华 张岳新 陈立潮 苏春辉

梁建民 梁国栋 崔广才 薛 虹

# 前 言

C++语言继承了C语言的简洁、高效、源程序的可移植性好等特点；同时克服了C语言的类型检查机制薄弱和不适合开发大型程序的缺点。C++语言为应用程序开发者提供了良好的程序开发环境，能产生模块化程度高、重用性和可维护性好的程序。当前，C++语言已得到广泛的应用，它不仅可应用于一般程序设计语言应用的场合，而且特别适用于开发中型和大型程序项目，也适合于计算机系统程序的设计与开发。

本书是针对普通高校计算机专业的学生学习程序设计语言课程编写的。全书分为16章，其中第2章（数据类型、运算符和表达式）、第4章（C++流程控制语句）、第5章（函数与编译预处理）、第8章（指针和引用）、第10章（构造函数与析构函数）、第11章（继承和派生类）、第12章（类的其他特性）和第13章（运算符重载）是本书的重点章节，而流程控制语句、函数、指针、虚函数是学习的难点。

本书系统地介绍了C++语言的基本概念、语法、语义，重点介绍了程序设计的思想和方法。针对初学者在学习过程中经常遇到的困难和问题，结合例题进行分析论述。本书的特点是由浅入深，通俗易懂，例题丰富，侧重实用，适合自学。本书主要介绍VC++程序设计语言，除做特别说明外，书中介绍的内容均适用于任何一种C++语言。

尽管本书的所有例题是在VC++6.0版本的编译系统下运行的，除第15章和第16章外，所有的例题在其他的C++编译系统下也能正确运行。每一章后均提供了多种类型的练习题，可供读者复习时参考。

本书除适用于大专院校计算机专业学生用做教材外，也能作为非计算机专业的学生学习C++程序设计的教材，还可作为教师的教学参考书或自学者学习用书。若教学时数较少时，第16章可以不讲授。为便于教学和实践，同时出版一本与该教材配套的学习指导与实验用书。

本书第1、5、12、13、14、16章由张岳新编写，其余章节由刘宇君、马锐编写，由张岳新统编全书。张宏教授审阅了全书。在本书编写与出版过程中，许多领导和同仁对本书提出了许多宝贵的意见，并给予支持和帮助，在此表示衷心的感谢。

由于编者的水平有限，错误和疏漏在所难免，敬请广大读者提出宝贵意见。

编 者

2003年12月

# 目 录

|                          |      |
|--------------------------|------|
| 第 1 章 C++语言概述 .....      | (1)  |
| 1.1 C++语言发展历史 .....      | (1)  |
| 1.2 C++的特点 .....         | (2)  |
| 1.3 简单的 C++程序介绍 .....    | (2)  |
| 1.4 面向对象的程序设计概述 .....    | (5)  |
| 1.5 VC++程序的上机过程 .....    | (6)  |
| 练习题 .....                | (7)  |
| 第 2 章 数据类型、运算符与表达式 ..... | (8)  |
| 2.1 C++的数据类型 .....       | (8)  |
| 2.2 基本运算符 .....          | (13) |
| 2.3 表达式语句和空语句 .....      | (24) |
| 练习题 .....                | (26) |
| 第 3 章 简单的输入输出 .....      | (29) |
| 3.1 标准输入流对象 cin .....    | (29) |
| 3.2 标准输出流 cout .....     | (31) |
| 3.3 其他的输入输出函数 .....      | (34) |
| 练习题 .....                | (34) |
| 第 4 章 C++的流程控制语句 .....   | (38) |
| 4.1 C++语言的语句和程序结构 .....  | (38) |
| 4.2 选择结构语句 .....         | (39) |
| 4.3 循环结构语句 .....         | (44) |
| 4.4 控制执行顺序的语句 .....      | (49) |



|                                |              |
|--------------------------------|--------------|
| 4.5 程序举例 .....                 | (52)         |
| 练习题 .....                      | (56)         |
| <b>第5章 函数与编译预处理</b> .....      | <b>(59)</b>  |
| 5.1 函数的定义与调用 .....             | (59)         |
| 5.2 函数的形参、实参、返回值及函数的原型说明 ..... | (61)         |
| 5.3 函数的嵌套与递归调用 .....           | (65)         |
| 5.4 作用域和存储类 .....              | (69)         |
| 5.5 内联函数 .....                 | (77)         |
| 5.6 具有缺省参数值和参数个数可变的函数 .....    | (78)         |
| 5.7 函数的重载 .....                | (82)         |
| 5.8 编译预处理 .....                | (83)         |
| 5.9 程序的多文件组织 .....             | (91)         |
| 练习题 .....                      | (92)         |
| <b>第6章 数组</b> .....            | <b>(99)</b>  |
| 6.1 数组的定义及应用 .....             | (99)         |
| 6.2 字符数组的定义及应用 .....           | (111)        |
| 6.3 字符串处理函数 .....              | (114)        |
| 练习题 .....                      | (118)        |
| <b>第7章 结构体、共同体和枚举类型</b> .....  | <b>(120)</b> |
| 7.1 结构体的定义及应用 .....            | (120)        |
| 7.2 位域 .....                   | (128)        |
| 7.3 共同体的定义及应用 .....            | (130)        |
| 7.4 枚举类型 .....                 | (134)        |
| 7.5 类型定义 .....                 | (136)        |
| 练习题 .....                      | (137)        |
| <b>第8章 指针和引用</b> .....         | <b>(140)</b> |
| 8.1 指针及指针变量 .....              | (140)        |



|                               |              |
|-------------------------------|--------------|
| 8.2 指针与数组 .....               | (144)        |
| 8.3 指针数组与指向指针的指针变量 .....      | (150)        |
| 8.4 指针与函数 .....               | (155)        |
| 8.5 new 与 delete .....        | (160)        |
| 8.6 引用和其他类型的指针 .....          | (163)        |
| 8.7 简单链表及其应用 .....            | (168)        |
| 练习题 .....                     | (177)        |
| <b>第 9 章 类和对象 .....</b>       | <b>(182)</b> |
| 9.1 类 .....                   | (182)        |
| 9.2 对象 .....                  | (189)        |
| 9.3 this 指针 .....             | (192)        |
| 练习题 .....                     | (194)        |
| <b>第 10 章 构造函数和析构函数 .....</b> | <b>(195)</b> |
| 10.1 构造函数 .....               | (195)        |
| 10.2 析构函数 .....               | (202)        |
| 10.3 对象的生存期 .....             | (205)        |
| 10.4 成员对象 .....               | (208)        |
| 10.5 对象数组 .....               | (210)        |
| 练习题 .....                     | (212)        |
| <b>第 11 章 继承和派生类 .....</b>    | <b>(217)</b> |
| 11.1 继承与派生 .....              | (217)        |
| 11.2 访问控制 .....               | (219)        |
| 11.3 派生类的构造函数和析构函数 .....      | (223)        |
| 11.4 二义性问题 .....              | (228)        |
| 11.5 虚基类 .....                | (233)        |
| 11.6 子类型关系 .....              | (236)        |
| 练习题 .....                     | (239)        |







|                                   |              |
|-----------------------------------|--------------|
| <b>第 12 章 类的其他特性 .....</b>        | <b>(243)</b> |
| 12.1 友元函数 .....                   | (243)        |
| 12.2 虚函数 .....                    | (247)        |
| 12.3 静态成员 .....                   | (257)        |
| 12.4 const、volatile 对象和成员函数 ..... | (260)        |
| 12.5 指向类成员的指针 .....               | (264)        |
| 练习题 .....                         | (267)        |
| <b>第 13 章 运算符重载 .....</b>         | <b>(271)</b> |
| 13.1 运算符重载 .....                  | (271)        |
| 13.2 特殊运算符的重载 .....               | (279)        |
| 13.3 一个字符串类 .....                 | (290)        |
| 练习题 .....                         | (295)        |
| <b>第 14 章 输入输出流类库 .....</b>       | <b>(298)</b> |
| 14.1 概述 .....                     | (298)        |
| 14.2 C++的基本流类体系 .....             | (299)        |
| 14.3 标准设备的输入/输出 .....             | (304)        |
| 14.4 文件流 .....                    | (309)        |
| 练习题 .....                         | (321)        |
| <b>第 15 章 模板与异常处理 .....</b>       | <b>(323)</b> |
| 15.1 模板 .....                     | (323)        |
| 15.2 异常处理 .....                   | (331)        |
| 练习题 .....                         | (337)        |
| <b>第 16 章 MFC 程序设计 .....</b>      | <b>(340)</b> |
| 16.1 Windows 和 MFC 编程 .....       | (340)        |
| 16.2 文档与视图结构 .....                | (349)        |
| 16.3 MFC 的数组类 .....               | (354)        |
| 16.4 鼠标使用实例 .....                 | (355)        |



---

|                      |       |
|----------------------|-------|
| 16.5 GDI 与文本处理 ..... | (367) |
| 16.6 CString 类 ..... | (370) |
| 16.7 文本信息的输出 .....   | (371) |
| 16.8 菜单的制作 .....     | (377) |
| 附录 I ASCII 码表 .....  | (382) |
| 附录 II 常用的库函数 .....   | (383) |
| 参考文献 .....           | (387) |



# 第 1 章 C++语言概述

本章介绍了 C++语言的起源、发展概况及其特点, C++程序的基本结构, 面向对象程序设计的基本概念, 简单的上机操作过程。

## 1.1 C++语言发展历史

自从 1946 年第一台电子数字计算机 ENIAC 问世以来, 随着计算机应用领域的不断扩大, 促进了计算机技术的高速发展, 尤其是近年来计算机的硬件和软件都是日新月异。作为应用计算机的一种工具——程序设计语言, 得到不断的充实和完善。每年都有新的程序设计语言问世, 老的程序设计语言不断地更新换代。

C++语言是在 C 语言的基础上逐步发展和完善的, 而 C 语言是吸收了其他高级语言的优点逐步成为实用性很强的语言。

20 世纪 60 年代, Martin Richards 开发了 BCPL 语言(Basic Combined Programming Language)。1970 年, Ken Thompson 在 BCPL 语言的基础上发明了实用的 B 语言。1972 年, 贝尔实验室的 Dennis Ritchie 和 Brian kernighan 在 B 语言的基础上, 做了进一步的充实和完善, 设计了 C 语言。以后, C 语言经过多次改进, 得到计算机界的认可并开始流行。目前, 国际上标准的 C 语言是 87ANSI C, 常用的有 Microsoft C, Turbo C, Quick C 等等。每一种不同的版本略有不同, 但基本的部分是兼容的。

1. C 语言具有以下的特点:

(1) C 语言是一种结构化的程序设计语言。语言本身比较简洁、使用比较灵活方便。

(2) 它具有一般高级语言的特点, 又具有汇编语言的特点。除了提供对数据进行算术、逻辑运算外, 还提供了二进制整数的位运算。用 C 语言开发的应用程序, 不仅其结构性较好, 且程序执行效率高。

(3) 程序的可移植性好。在某一种计算机上用 C 语言开发的应用程序, 其源程序基本上可以不做修改, 在其它型号和不同档次的计算机上重新编译连接后, 就完成应用程序的移植。

(4) 程序的语法结构不够严密, 程序设计的自由度大。精通 C 语言的程序设计者正是利用这一特点, 设计出高质量的通用的应用程序。但对于初学者来说, 掌握 C 语言并不是一件容易的事。往往是源程序编译时容易通过, 程序运行时出错, 且这种错误不易解决。

随着 C 语言应用的不断推广, C 语言存在的一些不足也开始流露出来。例如, C 语言对数据类型检查的机制比较弱; 缺少支持代码重用的结构; 随着计算机应用面的推广和软件工程规模的扩大, 难以适应开发特大型的程序; 软件维护困难等。

1980 年, 贝尔实验室的 Bjarne Stroustrup 博士及其同事对 C 语言进行了改进和扩充。在



保持了C语言简洁、高效前提下，克服了C语言存在的不足。并把Simula 67中类和对象的概念引入到C语言中。1983年由Rick Maseitti提议将改进后的C命名为C++（C Plus Plus）。后来，又把运算符的重载、引用、虚函数等功能加入到C++中，使C++的功能日趋完善。

当前用得较为广泛的C++有：VC++（Visual C Plus Plus）、BC++（Borland C Plus Plus）、AT&T C++等。

## 1.2 C++的特点

近年来，C++得到广泛的应用。C++除了继承C语言的一些特点之外，还具有以下特点：

1. C++是C语言的一个超集，它基本上具备了C语言的所有功能

用C语言开发的源程序代码可以不做修改或略做修改后，就可在C++的集成环境下编译、调试或运行。这对于推广或进一步开发目前仍有使用价值的软件是极为重要的，可节省人力和物力。

2. C++是一种面向对象的程序设计语言

面向对象的程序设计可大大增强程序的可读性和可理解性，使得各个模块的独立性强，程序代码的结构性更加合理。这对于设计和调试大的应用软件是非常重要的。

3. 用C++语言开发的应用程序，其扩充性强，可维护性好

在应用软件的开发过程中，对要解决的实际问题有一个认识、理解，再进一步认识和理解，直至客观地弄清楚问题的本质。这种认识和理解的过程，往往伴随着可能需要改变程序的结构或功能，这就要求开发工具具有较强的可扩充性。其次，对于任何一个已开发的应用软件，随着时间的推移和应用的深入，常要求增加或扩充新的功能、改进某些功能或发现程序故障。这均要求所设计的程序具有可扩充性和可维护性的特点。对于设计大的应用程序，这一特点是非常重要的。

## 1.3 简单的C++程序介绍

当前常用的几种C++语言略有不同，但其语法和C++的基本内容都是相同的。本书以学习Visual C++程序设计为主，除做特殊说明的章节外，均适用于任何一种C++语言。上机实习和实验的环境为Visual C++ 6.0。

C++的集成环境不仅可编译和调试用C++语言设计的程序，也可编译和调试用C语言设计的程序。C++的集成环境为了区分是用C语言还是用C++语言设计的程序，约定当源程序文件的扩展名为“.C”时，则为C语言程序；文件的扩展名为“.CPP”时，则为C++程序。在文件扩展名中不区分大小写字母。本书中除做特殊说明外，所有源程序文件的扩展名均为“.CPP”。

### 1.3.1 一个简单的C++程序

首先通过一个简单的实例，说明C++程序的基本结构及其特点。

例 1-1 根据输入的半径，求出一个圆的面积，并输出计算结果。

//源程序文件名为：EX1\_1.cpp

```
#include <iostream.h>
```



```
void main(void)
{
    float r,area;                                //说明两个变量：半径 r 和圆面积 area

    cout << "输入半径 r=";                        //显示提示符
    cin>>r;                                        //从键盘上输入半径变量 r 的值
    area = 3.1415926 * r * r;                    //计算圆面积
    /*输出半径和圆面积*/
    cout << "半径=" <<r<<"\n";                  // 输出变量 r 的值
    cout << "圆面积=" <<area<<"\n";
}
```

使用 C++ 提供的集成环境，将以上程序输入到文件 EX1\_1.cpp 中。将这程序编译和连接后，运行可执行程序时，在显示器上显示：

输入半径 r=

此时等待用户输入一个半径的实数值。设从键盘上输入 3.5 时，则显示器上显示：

半径=3.5

圆面积=38.4845

以上程序的设计步骤是：先输入半径 r，根据输入值求出圆面积 area，最后输出半径值和圆面积。对该程序的基本结构及各语句的作用说明以下几点：

#### (1) 注解或说明信息

在 C++ 程序的任何位置可以插入注解。所谓注解是指对程序的执行或功能来说是不起作用的信息，它是可有可无的。加入注解可增加程序的可读性和可理解性，说明程序的设计思想和方法。在 C++ 语言中有两种方法进行注解，第一种是用 “/\*” 和 “\*/” 把注解信息括起来。这种注解方式可以出现在程序中的任何位置。第二种方法是用两个连续的 “/” 字符，它表示从此开始到本行结束为注解。

#### (2) 包含文件或编译预处理指令

用 # 开头的行称为编译预处理指令，如程序中的第二行。有关包含文件或编译指令的具体用法及规则将在第 5 章中介绍。头文件 iostream.h 是一个标准输入输出流的头文件。程序中要从键盘上输入数据，或将计算结果送到显示器上输出时，要包含该头文件。

#### (3) 主函数 main

程序中的 main 函数称为主函数。任一 C++ 程序均有一个且只能有一个主函数。该函数可出现在源程序中的任何位置，执行程序时，系统总是从 main 函数开始执行的。

#### (4) 花括号对 “{}”

任一函数体均以 “{” 开始，并以 “}” 结束。花括号对 “{}” 称为函数括号或语句括号。注意，花括号一定要配对使用。

#### (5) 任一 C++ 程序均由一个或多个函数所组成

主函数 main 是必不可少的，其余函数可有可无。通常，程序中用到的函数可以是库函数





或用户自定义的函数。在本例中，`cin` 和 `cout` 分别用于输入数据和输出数据。任一函数由若干个语句所组成，每一个语句均以“;”结束。

#### (6) 程序的书写规则

按照 C++ 的语法规则，程序的书写形式是自由的，可以将一个语句写成若干行（不能在一个基本语法单位之间换行），也可将若干个语句写在一行内。但为了方便程序的阅读和相互交流，程序的书写格式及源程序文件中的格式必须符合以下规则：

1) 对齐规则。同一层次的语句必须从同一列开始，闭花括号与同一层次的开花括号对齐或与开花括号所在的语句对齐。

2) 缩进规则。为了方便程序的阅读和理解，体现程序的层次结构，属于内一层次的语句，必须缩进若干个字符。通常缩进两个、四个或八个字符的位置。

3) 每一函数的定义(包括第一个开花括号)必须从第一列开始书写。

(7) C++ 语言不提供专门的输入输出语句。数据的输入与输出是通过函数来实现的。

(8) 严格区分大小写字母。在 C++ 程序中，除注解中的字母外，是严格区分大小写字母的。如 `B` 与 `b` 表示两个不同的标识符。在输入源程序或编辑源程序时，特别要注意这一点。

### 1.3.2 程序的基本要求

用 C++ 语言解决实际问题时，对所设计的程序是有质量要求的。通过本课程的学习，所设计的程序应达到以下几方面的基本要求：

#### 1. 程序的正确性

首先要求程序正确无误，包括语法和语义正确，算法描述正确。这是对程序的最基本的要求。

#### 2. 程序的可读性和可理解性好

设计的程序被他人阅读时，要易于读懂，容易理解程序的设计思想和设计方法。为了保证程序的可读性和可理解性，第一，要求程序的结构性好，应采用软件工程的程序设计方法来设计程序；第二，在程序中增加足够的注解，说明程序设计的思想和方法；第三，程序的书写格式必须规范。

#### 3. 程序的可维护性好

这要求程序易于修改，易于增加新的功能。

#### 4. 程序的结构好，而且执行速度快

必须指出，要想设计出高质量的程序，仅学习本课程的知识是不够的，还要掌握数据结构、算法设计与分析、软件工程及程序设计方法学等知识。本课程特别强调实践，只有把理论学习与大量的上机实践相结合，才能学好本课程，才可能设计出高质量的程序。

### 1.3.3 C++ 程序的开发步骤

针对一个实际问题，用 C++ 语言设计一个实用程序时，通常要经过如下五个开发步骤：

#### 1. 用户需求分析

根据要解决的实际问题，分析用户的所有需求，并用合适的方法、工具进行详细描述。

#### 2. 根据用户需求，设计 C++ 源程序



利用 C++ 的集成环境或某一种文本编辑器将设计好的源程序输入到计算机中一个文件中, 文件的扩展名为 .cpp。

### 3. 编译源程序, 并产生目标程序

在编译源程序文件时, 若发生语法或语义错误时, 要修改源程序文件, 直到没有编译错误为止。编译后, 为源程序产生了目标文件。在 PC 机上, 目标程序文件的扩展名为 .obj。

### 4. 将目标文件连接成可执行文件

将一个或多个目标程序与库函数进行连接后, 产生一个可执行文件。在 PC 机上, 可执行文件的扩展名为 .exe。

### 5. 调试程序

运行可执行程序文件, 输入测试数据, 并分析运行结果。若运行结果不正确, 则要修改源程序, 并重复以上的过程, 直到得到正确的结果为止。

## 1.4 面向对象的程序设计概述

面向对象的程序设计 (Object Oriented Programming, 简称 OOP) 方法是近年来十分流行的一种程序设计方法, 它试图用客观世界中描述事物的方法, 来用程序描述要解决的事情。在 C++ 中通过引入类和对象的概念, 增加了程序模块的独立性和可扩展性。

### 1.4.1 对象及面向对象的程序设计

对象 (Object) 是 OOP 中最重要的概念之一。简单地说, 对象是对一个客观实体的具体描述。它是既有数据又有对数据进行操作的代码的一个逻辑实体。数据用来描述客观实体的属性, 而操作代码用来描述客观实体的行为或动作。

面向对象的程序设计 (OOP), 就是用 OOP 来描述客观世界中所需说明的有关事物。这种程序设计方法有利于提高软件的可靠性、可理解性和可维护性。

### 1.4.2 面向对象程序设计的要素

C++ 是一种面向对象的程序设计语言, 它支持面向对象程序设计的几个要素: 封装性、继承和派生性、重载性和多态性。下面对这些要素分别做简要的介绍。

#### 1. 封装性

将描述一个对象的数据及对这些数据进行操作处理的代码有机地组成一个整体, 形成一个模块。对该模块中的数据及代码的存取权限加以限制后, 可使模块完全独立。对象的这种特性称为封装性。经封装后, 只能通过对象中的代码来处理描述对象的数据, 而在模块外的程序代码均不能访问对象的数据。这种封装性可通过定义类来实现, 对象是类的一个实例。

封装性有利于程序的调试和维护。

#### 2. 继承和派生性

一个类可以派生出新的类, 原来的类称为基类, 而由基类派生出来的类称为派生类。派生类可以部分或全部地继承基类的数据或代码。这种特性称为继承和派生性。派生类又可以作为一个基类, 再次派生出新的类, 这样逐次地继承和派生下去, 可形成一棵树状的类结构图。





利用这种特性, 首先, 在基类正确的前提下, 只要调试派生类中增加的数据或代码, 从而可减少程序设计的错误; 其次, 可加快和简化程序设计, 提高工作效率。

### 3. 重载性

同一个运算符, 根据不同的运算对象可以完成不同运算; 或同一个函数名, 根据其不同的参数可完成不同的功能。这种特性称为重载性。例如, “+”运算符可以完成两个整数或两个实数的求和运算, 也可以完成两个复数的加法运算, 还可以完成两个字符串的拼接运算。在 C++中有两种重载, 一种是运算符的重载, 另一种是函数的重载。例如, 使用相同的函数名 `abs`, 可以分别求出整数、实数或双精度实数的绝对值。

这种特性, 为简化程序设计提供了方便。

### 4. 多态性

协同完成某一任务的若干个对象之间往往需要进行信息传递。按一定格式传递的信息称为消息。C++系统提供了实现信息传递的机构。同一个消息发送给不同的对象时, 可以导致完全不同的动作或行为, 这种特性称为多态性。多态性的重要性在于允许一个类体系的不同对象, 能够以不同的方式响应同一个消息, 实现了“同一接口, 多种方法”的目的。

利用这种特性, 可增加程序设计的灵活性和通用性, 也可减轻类体系使用者的记忆负担。

## 1.5 VC++程序的上机过程

VC++为用户开发 C++或 C 程序提供了一个集成环境, 这个集成环境包括: 实现源程序的输入、编辑和修改, 源程序文件的保存与打开, 源程序的编译和连接, 程序运行期间的调试与跟踪, 工程项目的自动管理, 提供应用程序的开发工具, 多窗口管理, 提供联机帮助等。

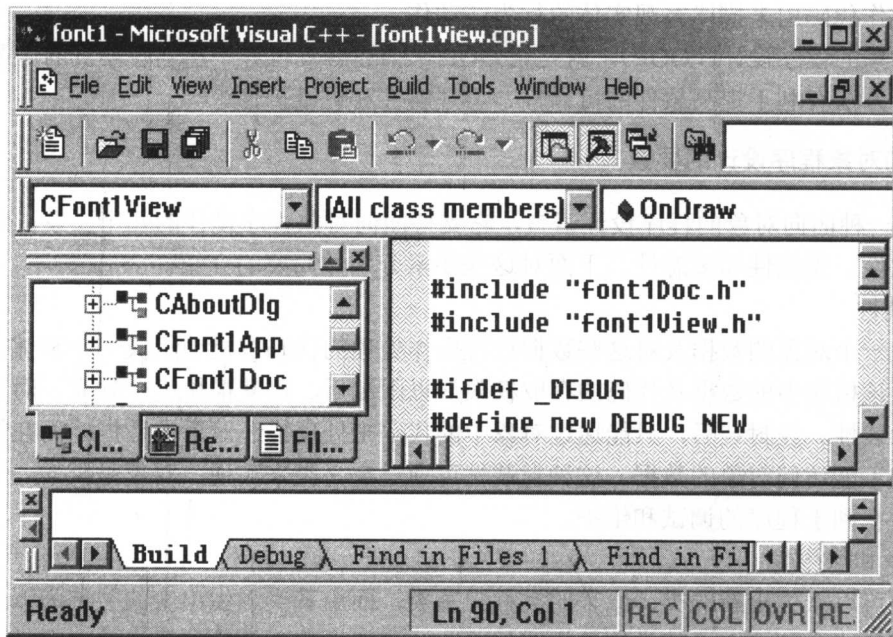


图 1-1 VC++集成环境





该集成环境功能齐全,环境复杂,只有经过较长时间的上机实践、逐步理解和体会,才可能达到能熟练运用集成环境中的各种工具。本节仅介绍简单的上机操作过程,有关上机操作较为详细的说明,请参看与本教材配套的实验指导书。

在 Windows 操作系统下启动 VC++ 的集成环境,则产生图 1-1 所示的一个组合窗口。最上面部分为窗口标题。第二部分为菜单条,菜单有“文件(File)”菜单、“编辑(Edit)”菜单等。第三部分为工具条,有“打开”文件、“剪切”、“复制”、“粘贴”等。主体部分由三个子窗口组成,左边的子窗口为程序结构窗,用于快速定位文件、资源或类;右边的子窗口为源程序编辑窗,用于输入或编辑源程序文件;下边的子窗口为提示信息输出窗,用来显示出错信息或调试程序的有关信息。

有多种建立一个新的源程序文件的方法。一种方法是选择“File”菜单中的“New”命令,这时弹出一个子窗口,它有四个标签,分别是 Files、Projects、Workspaces 和 Other Documents。用鼠标单击 Files 标签,在可选清单中双击“C++ Source File”。这时光标进入源程序编辑子窗口,这时可输入 C++ 源程序了。编辑源程序的方法与 Word 基本相同。

要打开一个已存在的源程序文件时,可选择 File 菜单中的 Open 命令,然后根据提示信息选择相应的源程序文件名。由系统将指定的源程序文件调入程序编辑子窗口,就可以对源程序文件进行编辑了。

当正确地输入源程序或编辑结束后,要将源程序保存到磁盘文件中。可选择 File 菜单中的 Save 命令来实现。源程序文件存盘后,可选择 Build 菜单中的 Build 命令来编译和连接程序。当编译和连接成功后,可选择 Build 菜单中的 Execute 命令来执行程序。

## 练 习 题

1. 程序的基本要求有哪些?
2. 面向对象程序设计有哪几个基本要素?
3. 在 VC++ 集成环境下,从输入源程序到得到正确的结果,要经过哪些步骤?
4. 将本章例题中的程序输入到源程序文件 ex1\_1.cpp,并在 VC++ 集成环境下编译、连接和运行。
5. 在 VC++ 中,有哪两种注解方法?每一种注解方法适用于什么场合?
6. 简要说明 C++ 程序开发的每一个步骤。

