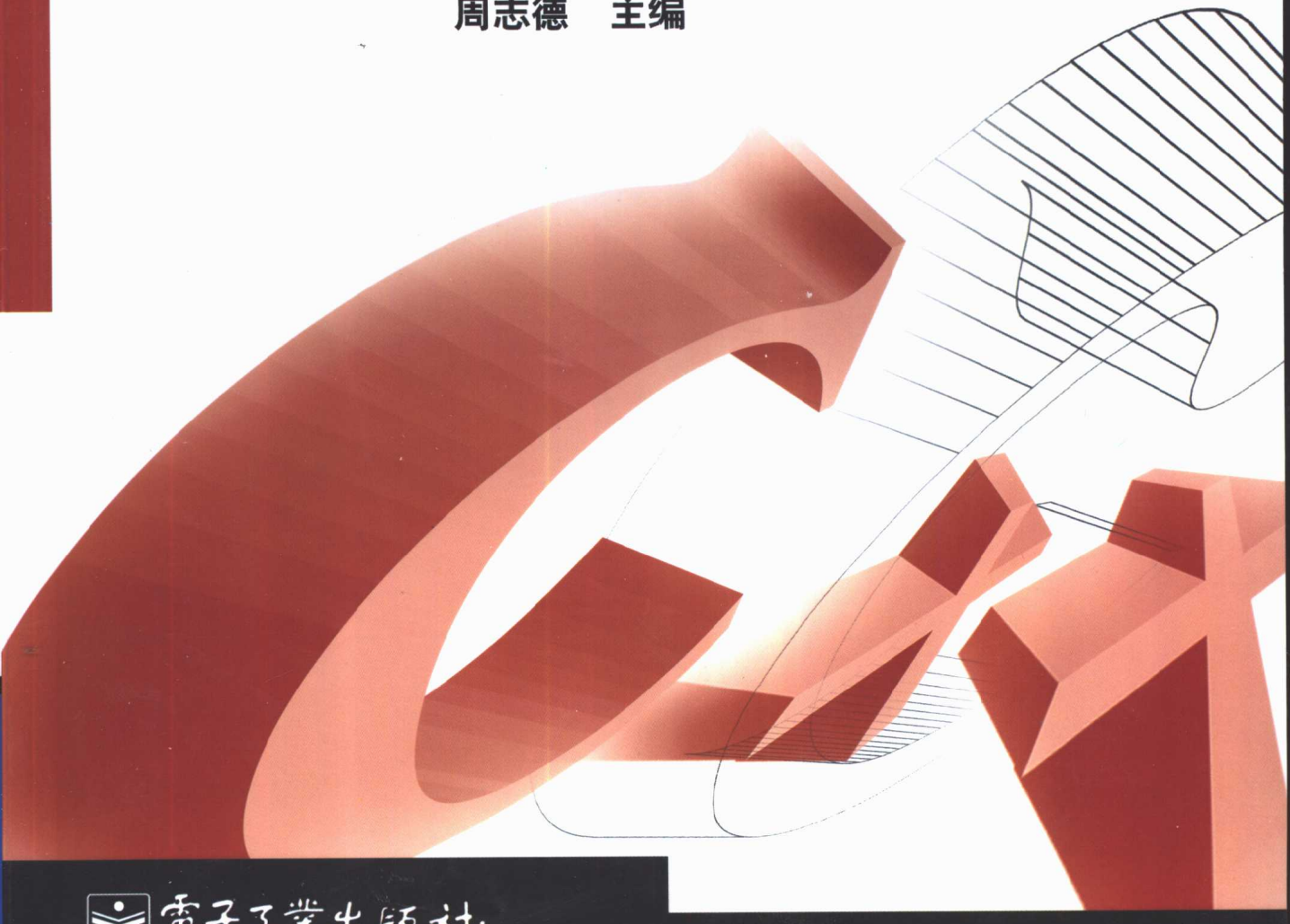
电子·教育

高职高专计算机系列教材

中国计算机学会高职高专教育学组推荐出版

C++程序设计

周志德　主编



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高职高专计算机系列教材

C++程序设计

周志德　主编

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书共12章。第1,2章介绍了C++的概述、上机过程与C++数据类型和表达式;第3,4章叙述了三种基本程序结构、流程控制语句和数组;第5,6章讨论了函数的定义和调用方法、函数的嵌套调用和递归调用、变量的存储类型、内联函数、重载函数、编译预处理中的宏定义、文件包含处理与条件编译;第7章讲解了指针与指针变量、指针与数组、指针数组和指向指针的指针变量、指针与函数、new和delete运算符及引用;第8章介绍了枚举型、结构体、共同体与链表;第9,10章讲述了类和对象、构造函数与析构函数、继承与派生、冲突、支配规则和赋值兼容性与静态成员;第11,12章叙述了友元与运算符重载、多态性与虚函数、流类体系与文件操作。

本书可作为高职高专院校计算机、电子等专业的教材,本书起点低,不要求学过其他程序设计语言,可作为程序设计的入门语言来学习,也可作为从事计算机应用工作的工程技术人员培训和自学的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

C++程序设计/周志德主编. —北京:电子工业出版社,2002.8
高职高专计算机系列教材
ISBN 7-5053-7878-3

Ⅰ.C++… Ⅱ.周… Ⅲ.C++—程序设计—高等学校:技术学校—教材 Ⅳ.TP312

中国版本图书馆CIP数据核字(2002)第056324号

责任编辑:洪国芬
印　　刷:北京东光印刷厂
出版发行:电子工业出版社　http://www.phei.com.cn
　　北京市海淀区万寿路173信箱　邮编100036
经　　销:各地新华书店
开　　本:787×1 092　1/16　印张:23.25　字数:595千字
版　　次:2002年8月第1版　2002年8月第1次印刷
印　　数:8 000册　定价:28.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077

前　言

“C++程序设计”介绍了一种目前流行的面向对象程序设计语言，是学习数据结构等计算机课程的基础，是学习可视化程序设计软件 VC++的基础，是计算机专业及许多工科专业学生应掌握的一门计算机语言课程。本教材是根据“中国计算机学会高职高专教育学组”2001年审定的编写大纲编写的，主要介绍 C++语言中的数据类型、程序结构与控制语句、数组、函数、结构体、指针、类、对象、继承、重载、多态性技术、文件操作的基本概念与方法。本教材有如下特点：

（1）本教材将“C 语言程序设计”与“C++程序设计”综合为一本教材，用 C++的语言来描述原先用 C 语言描述的内容，然后再加上面向对象的程序设计内容。这样做的好处是：

① 学生可以从数据类型、程序结构等基础内容开始由浅入深地进行学习，所以本教材起点低，可作为程序设计的入门语言来学习。

② 由于将两门课程综合为一门课程，所以可减少总的教学时间，在一学期内完成本课程的学习。

③ 可以使学生直接学习面向对象的程序设计方法。

（2）针对高职高专类学生的特点，本书尽可能使用通俗易懂的语言来叙述各章节的内容，并尽可能使用典型例题来说明各章节知识点的概念与使用方法，力求将各章节的重点、难点内容解释清楚，以求多数学生在教师讲课后能看懂教材，学会知识的应用。

（3）针对高职高专类学生理论教学的要求为“知识够用”，将一些理论性较深而不太实用的内容做了删除，降低例题与习题的难度，使多数学生在听完课后能独立完成书中的习题。

（4）由于描述 C++类与对象的程序段一般都较长，因此本书中尽量用同一类型的例题介绍某个系列概念。例如，用描述学生成绩的类讲解类与对象的概念、定义及使用方法；用描述矩形的类介绍构造函数、拷贝构造函数、默认构造函数、析构函数等一系列的概念、定义及使用方法，以减少教师在黑板上的书写工作量，提高课堂的讲课效率。

（5）学生比较难理解的某些章节内容，先通过例题分析，然后引出基本概念，给出定义格式、结论等内容。

（6）高职高专学生在学习 C++程序设计时遇到的困难之一是：一方面要理解 C++中许多比较难的概念，另一方面又要理解复杂的算法。为了解决该问题，本教材在程序的算法上重点抓住常用的一些典型算法，如累加和、累乘积、最大值、最小值、平均值、排序等，并将这些典型算法作为介绍各章节基本概念的例题。这样可减轻学生学习中理解算法的负担，提高课堂的教学效果。一些有较复杂算法的例题只会出现在每章后的程序设计应用举例中。

（7）每章前都有学习本章目的、要求，每章后有本章小结，并配有一定量的习题，便于教师教学和学生自学。各章内容充实，安排合理，衔接自然。

在本书的编写过程中，参考了目前国内比较优秀的有关 C++语言程序方面的书籍资料，在此谨向有关作者表示感谢。

本书第 1~6 章由侯正昌编写，第 7~12 章由周志德编写。全书由周志德统编，南京邮电学院唐瑞庭副教授审阅。在全书的编写过程中，张燕、倪卫东、严惠琴、王得燕等老师对本书提出了非常宝贵的意见，在此，仅对上述老师表示感谢。

本书若有错误及不足之处，恳请读者给予指正。

编　者

2002 年 5 月 15 日

此为试读，需要完整PDF请访问：www.ertongbook.com

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第1章　C++概述

通过本章的学习，应了解 C++语言的起源及其特点。掌握 C++程序的基本结构，会编写极简单的 C++程序。了解 C++程序的开发步骤，熟悉 Visual C++集成环境，初步掌握 C++程序的上机操作方法。

1.1　C++的起源

C++语言是在 C 语言的基础上逐步发展和完善起来的，因此介绍 C++ 的起源不妨首先回顾一下 C 语言的发展史。

1967 年，Martin Richards 为编写操作系统软件和编译程序开发了 BCPL 语言（Basic Combined Programming Language）；1970 年，Ken Thompson 在继承 BCPL 语言的许多优点的基础上开发了实用的 B 语言；1972 年，贝尔实验室的 Dennis Ritchie 在 B 语言的基础上做了进一步的充实和完善，开发出了 C 语言。当时，设计 C 语言是为了编写 UNIX 操作系统，以后 C 语言经过多次改进，逐渐开始流行。目前常用的 C 语言版本基本上都是以 ANSIC 为基础的。

C 语言具有许多优点，比如语言简洁、灵活，运算符和数据结构丰富，具有结构化控制语句，程序执行效率高，同时具有高级语言和汇编语言的优点等。与其他高级语言相比，C 语言具有可以直接访问物理地址的优点，与汇编语言相比又具有良好的可读性和可移植性。因此，C 语言得到了极为广泛的应用。

随着 C 语言应用的推广，C 语言存在的一些缺陷或不足也开始暴露出来，并受到大家的关注。比如 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；随着软件工程规模的扩大，难以适应开发特大型程序。同时 C 语言毕竟是一种面向过程的编程语言，已经不能满足运用面向对象的方法开发软件的需要。C++便在 C 语言基础上，为克服 C 语言本身存在的缺点，同时为支持面向对象的程序设计而研制出来的一种通用的程序设计语言，它是在 1980 年由贝尔实验室的 Bjarne Stroustrup 创建的。

研制 C++的一个重要目标是使 C++首先是一个更好的 C，所以 C++ 根除了 C 中存在的问题。C++的另一个重要目标就是面向对象的程序设计，因此在 C++ 中引入了类的机制。最初的 C++被称为“带类的 C”，1983 年正式命名为 C++（C Plus Plus）。以后经过不断完善，形成了目前的 C++。

当前运用得较为广泛的 C++有 Microsoft 公司的 Visual C++（简称 VC++）和 Borland 公司的 Borland C++（简称 BC++）。本书以 Microsoft Visual C++ 6.0 集成环境为例介绍 C++语言。

1.2　C++的特点

C++ 语言的主要特点表现在两个方面，一是全面兼容 C 语言，二是支持面向对象的程序

设计方法。

（1）C++是一个更好的C，它保持了C语言的优点，大多数的C程序代码略做修改或不做修改就可在C++的集成环境下调试和运行。这对于继承和开发当前已在广泛使用的软件是非常重要的，可以节省大量的人力和物力。

（2）C++是一种面向对象的程序设计语言。它使得程序的各个模块的独立性更强，程序的可读性和可移植性更强，程序代码的结构更加合理，程序的扩充性更强。这对于设计、编制和调试一些大型的软件尤为重要。

1.3 简单C++程序

C++集成环境不仅支持C++程序的编译和调试，而且也支持C程序的编译和调试。通常，C++集成环境约定：当源程序文件的扩展名为.c时，则为C程序；而当源程序文件的扩展名为.cpp时，则为C++程序。本书中，所有例题程序的文件扩展名均为.cpp。

【例1.1】 文本的原样输出。文件名为example1_1.cpp。

```
//文本原样输出程序
#include <iostream.h>
void main(void)
{    cout<<"Welcome to C++!\n";
}
```

该程序经编译和链接后，运行可执行程序时，在显示器上显示：

```
Welcome to C++!
```

该程序中，main( )表示主函数，每个C++程序必须有且只能有一个主函数，C++程序总是从主函数开始执行的。main( )函数之前的void表示main( )函数没有返回值，main( )函数后括号内的void表示main( )函数没有形式参数。在花括号内的部分是函数体，函数体由语句组成，每个语句由分号结束。cout是C++中的一个输出流，与符号“<<”结合使用可以输出常量、变量的值及原样输出双引号中的字符串。“\n”是换行符，即输出上述信息后换行。

程序中的#include是C++编译预处理中的文件包含命令，“iostream.h”是头文件，为了能使用输出流cout和输入流cin，程序开头必须用#include命令将文件iostream.h中的内容包含到本文件中来。

程序中以“//”开头的是注释，注释是对程序的说明，用来提高程序的可读性，可以放在程序的任何位置，对程序的编译和运行不起作用。

【例1.2】 求两个整数的和。

```
/*求两个整数和的程序*/
#include <iostream.h>
void main(void)
{    int x,y,sum;                          //说明变量x,y,sum为整型数
     cout<<"Input first integer:";         //显示提示信息
     cin>>x;                               //从键盘上输入变量x的值
     cout<<"Input second integer:";        //显示提示信息
     cin>>y;                               //从键盘上输入变量y的值
```

```
    sum=x+y;                                    //求和
    cout<<"Sum is "<<sum<<endl;                 /*输出结果*/
}
```

该程序经编译和链接后，运行可执行程序时，在显示器上显示：

```
Input first integer：5
Input second integer：4
Sum is 9
```

该程序中的语句 int x,y,sum 用来说明变量 x,y,sum 为 int（整型）变量。程序中的语句 sum=x+y 是一个赋值语句，表示将 x 和 y 的值相加，其结果送给变量 sum。在“/*”和“*/”之间的部分也表示注释。“endl”是换行符，相当于'\n'。

【例1.3】 输入两个整数a和b，输出其中较大的一个数。

```
#include <iostream.h>
void main(void)
{   int max(int x,int y);
    int a,b,m;
    cout<<"Input a,b:";
    cin>>a>>b;
    m=max(a,b);
    cout<<"max="<<m<<endl;
}
int max(int x,int y)
{   int z;
    if (x>y) z=x;
    else z=y;
    return(z);
}
```

该程序经编译和链接后，运行可执行程序时，在显示器上显示：

```
Input a,b：3  5
max=5
```

该程序由两个函数组成：主函数 main( )和被调用函数 max( )。函数 max( )的作用是找出 x 和 y 中的较大者，并通过 return 语句返回给主函数。主函数用来输入两个变量 a 和 b 的值，调用 max 函数找出其中的较大者，并输出结果。

通过以上例题，可以看出 C++程序的结构有以下特点：

（1）C++程序通常由包括 main( )在内的一个或多个函数组成，函数是构成 C++程序的基本单位。其中名为 main( )的函数称为主函数，可以将它放在程序的任何位置。但是，不论主函数放在程序的什么位置，一个 C++程序总是从主函数开始执行，由主函数来调用其他函数。所以，任何一个可运行的 C++程序必须有一个且只能有一个主函数。被调用的其他函数可以是系统提供的库函数，也可以是用户自定义的函数。例如，例 1.3 的 C++ 程序就是由主函数 main( )和用户自定义函数 max( )组成的。

（2）C++的函数由函数的说明部分和函数体两部分组成。

① 函数的说明部分（又称为函数头）。函数的说明部分包括函数名、函数类型、函数参数（形式参数）及其类型。函数类型为函数返回值的类型。例如：

```
int max(int x,int y)
```

表示定义了一个函数，函数名为 max，函数值的类型为 int（整型），该函数有两个形式参数 x,y，其类型均为 int（整型）。

无返回值的函数是 void 类型（无值类型）。main( )函数是一个特殊的函数，可看做是由操作系统调用的一个函数，其返回值是 void 类型或 int 类型。函数参数可以没有，但函数名后面的括号不能省略。

② 函数的执行部分（又称为函数体）。函数说明部分下面的花括号括起来的部分称为函数体。例如：

```
{   int z;
    if (x>y) z=x;
    else z=y;
    return(z);
}
```

如果一个函数内有多对花括号，则最外层的一对花括号为函数体的范围。函数体一般包括变量说明和执行语句两部分。在某些情况下可以没有变量定义，甚至可以既无变量定义又无执行语句（即空函数）。例如：

```
void dump(void )
{   }
```

（3）C++中每个语句和数据说明必须以分号结束。分号是 C++语句的必要组成部分。例如 int z，又如 return(z)。

（4）C++程序的书写格式比较自由，一行内可以写多个语句，一个语句也可以分成几行来写。例如，一个 if 语句写在两行上：

```
if (x>y) z=x;
else z=y;
```

也可以将一个 if 语句写在一行上，即写成：

```
if (x>y) z=x; else z=y;
```

但是，为了便于程序的阅读、修改和相互交流，程序的书写应该符合以下基本规则：

① 同一层次的语句从同一列开始书写，同一层次的开花括号与对应的闭花括号在同一列上；

② 属于内一层次的语句，缩进几个字符，通常缩进两个、四个或八个字符的位置；

③ 任一函数的定义均从第一列开始书写。

（5）C++语言没有专门的输入/输出语句，输入/输出操作是通过输入/输出流 cin 和 cout 来实现的。例如：

```
cin>>a>>b;
```

用来输入变量 a 和 b 的值。又如：

```
cout<<"max="<<m<<endl;
```

用来输出变量 m 的值。

（6）在 C++中，严格区分字母的大小写。例如 int a,A 表示定义两个不同的变量 a 和 A。

（7）在 C++ 程序的任何位置处都可以插入注释信息。注释方法有两种：一种方法是用“/*”和“*/”把注释内容括起来，它可以用在程序中的任何位置。例如：

/*求两个整数和的程序*/

另一种方法用两个连续的“/”字符，它表示从此开始到本行结束为注释内容。例如：

//说明变量 x,y,sum 为整型数

（8）以“#”开头的行称为编译预处理命令。例如：

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

表示本程序包含有头文件 iostream.h。

以上所述的有关函数、输入/输出流等概念将在以后的章节中详细介绍。

1.4 C++ 上机操作

1.4.1 C++ 程序的开发步骤

C++语言是一种编译性的语言，设计好一个 C++ 源程序后，需要经过编译、链接，生成可执行的程序文件，然后执行并调试程序。一个 C++ 程序的开发可分成如下几个步骤：

（1）分析问题。根据实际问题，分析需求，确定解决方法，并用适当的工具描述它。

（2）编辑程序。编写 C++ 源程序，并利用一个编辑器将源程序输入到计算机中的某一个文件中。文件的扩展名为.cpp。

（3）编译程序。编译源程序，产生目标程序。文件的扩展名为.obj。

（4）链接程序。将一个或多个目标程序与库函数进行链接后，产生一个可执行文件。文件的扩展名为.exe。

（5）运行调试程序。运行可执行文件，分析运行结果。若有错误进行调试修改。

在编译、链接和运行程序过程中，都有可能出现错误，此时要修改源程序，并重复以上过程，直到得到正确的结果为止。

1.4.2 C++ 程序上机操作方法

Visual C++ 为用户开发 C++ 程序提供了一个集成环境，这个集成环境包括：源程序的输入和编辑，源程序的编译和链接，程序运行时的调试和跟踪，项目的自动管理，为程序的开发提供各种工具，并具有窗口管理和联机帮助等功能。

使用 Visual C++ 集成环境上机调试程序可分成如下几个步骤：启动 Visual C++ 集成环境；生成项目；生成和编辑源程序，把一个或多个源程序送到各自的文件中；将源程序文件加入到项目中；根据需要改变项目的设置；最后编辑、链接和运行程序。下面以例 1.1 为例说明 C++ 程序的上机操作方法，程序的文件名为 example1_1.cpp。

1. *启动 Visual C++*

当在桌面上建立了 VC++ 的图标后，可通过鼠标双击该图标启动 VC++；若没有建立相应的图标，则可以通过菜单方式启动 VC++，即用鼠标单击“开始”按钮，选择“程序”，选择“Microsoft Visual Studio 6.0”，选择“Microsoft Visual C++ 6.0”，启动 VC++。

VC++启动成功后，就产生如图 1.1 所示的 VC++集成环境。

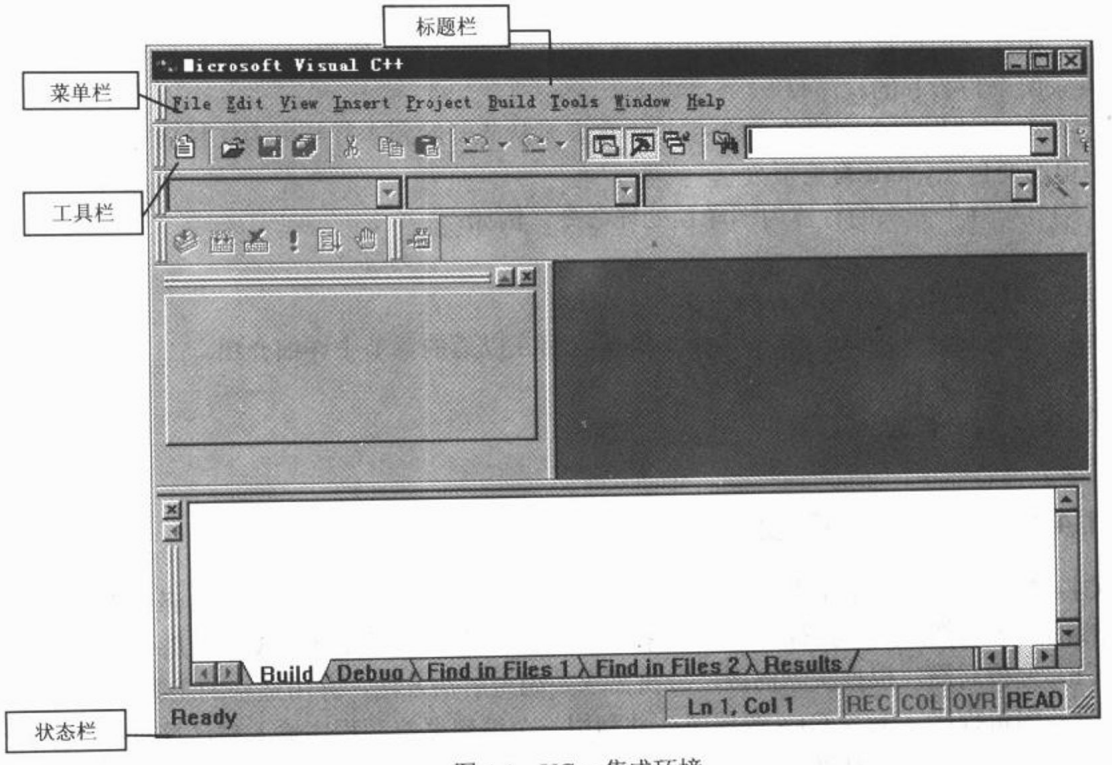


图 1.1　VC++集成环境

VC++集成环境是一个组合窗口。窗口的第一部分为标题栏；第二部分为菜单栏，其中包括“File（文件）”、“Edit（编辑）”、“View（视图）”、“Insert（插入）”、“Project（项目）”、“Build（编译、链接和运行）”、“Tools（工具）”、“Windows（窗口）”、“Help（帮助）”等菜单。第三部分为工具栏，其中包括常用的工具按钮；第四部分为状态栏。还有几个子窗口。

2. 生成项目

通常都是使用项目的形式来控制和管理 C++程序文件的，C++的项目中存放特定程序的全部信息，包含源程序文件、库文件、建立程序所用的编译器和其他工具的清单。C++的项目以项目文件的形式存储在磁盘上。

生成项目的操作步骤为：

（1）选择集成环境中的“File”菜单中的“New”命令，产生“New”对话框，如图 1.2 所示。

（2）选择对话框中的“Projects”标签，以便生成新的项目。在产生新项目时，系统自动生成一个项目工作区，并将新的项目加入到该项目工作区中。

（3）在项目类型清单中，选择“Win32 Console Application”项目，表示要生成一个 Windows 32 位控制台应用程序的项目。

（4）在“Location”文本框中输入存放项目文件的文件夹路径，例如 D:\C++。

（5）在“Project Name”文本框中输入项目名，例如 example1_1。

（6）检查“Platforms”文本框中是否已显示“Win32”，表示要开发 32 位的应用程序。

（7）单击“New”对话框中的“OK”按钮，这时就产生了一个项目文件。系统自动加上文件扩展名“.dsw”。例如，系统在文件夹“D:\C++\example1_1”中产生了一个项目文件

此为试读，需要完整PDF请访问：www.ertongbook.com