



21世纪高等院校规划教材

C语言程序设计 ——Visual C++ 6.0环境

(第二版)

主 编 张 昕
副主编 魏 玲 杨雪飞 张 昊



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书以高级语言程序设计方法为依据,从实际应用出发,通过典型实例阐述了C语言的概念、语法及程序设计技术,结合实际应用进行项目实训,并兼顾计算机二级考试的要求,实例与考题相结合。程序调试和运行环境为Windows平台下的Visual C++ 6.0,并增加了C++的部分基本内容,使读者能够初步了解C++的基本内容,为以后学习面向对象程序设计语言打下基础。

本书注重理论联系实际,具有实例切入、深入浅出、分散难点、突出重点、阐述简明、分析透彻的特点。全书分为10章,第1章至第9章介绍C语言结构化程序设计知识、语法规则及编程技能,该部分内容安排对应全国计算机等级二级考试。第10章介绍C++面向对象的程序设计基本方法,讲述C++面向对象的主要特征。全书重点突破C语言结构化程序设计内容,指导读者建立结构化程序设计的思想,为后续程序开发课程奠定编程思想基础。

本书除了可用作高等院校本、专科学生的教材外,还兼顾一般读者,可作为计算机软件编程的自学参考书。

本书所配教学资源(电子教案、程序/案例代码等)可从中国水利水电出版社与万水书苑网站免费下载,网址为:<http://www.waterpub.com.cn/softdown/>和<http://www.wsbookshow.com>。

图书在版编目(CIP)数据

C语言程序设计: Visual C++ 6.0环境 / 张昕主编

— 2版. — 北京: 中国水利水电出版社, 2012.1

21世纪高等院校规划教材

ISBN 978-7-5084-9303-9

I. ①C… II. ①张… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第271003号

策划编辑: 雷顺加 责任编辑: 宋俊娥 加工编辑: 李元培 封面设计: 李 佳

书 名	21世纪高等院校规划教材 C语言程序设计——Visual C++ 6.0环境(第二版)
作 者	主 编 张 昕 副主编 魏 玲 杨雪飞 张 昊
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 68367658(发行部)、82562819(万水) 北京科水图书销售中心(零售) 电话: (010) 88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	184mm×260mm 16开本 17.5印张 426千字
版 次	2005年6月第1版第1次印刷 2012年1月第2版 2012年1月第1次印刷
印 数	0001—4000册
定 价	30.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换

版权所有·侵权必究

第二版前言

目前, C 语言已成为计算机及其相关专业首选的程序设计入门语言, 作为教材, 本书有选择地介绍了 C 语言中的主要功能和内容, 介绍 C 语言中的语法, 更重要的是讲授如何在 C 环境下进行程序设计, 如何同第四代计算机语言的程序设计接轨。

本教材融入了编者多年的教学经验, 充分考虑到初学者的能力、认知水平、知识结构等因素, 遵照循序渐进、由浅入深的原则, 文字叙述清晰扼要、通俗易懂, 理论阐述简明科学, 选例经典实用、分析透彻浅显, 使读者在要点把握时感到具体生动, 而不抽象枯涩。在内容安排上分为两个部分, 第一部分由第 1~9 章组成, 介绍 C 语言结构化程序设计知识、语法规则及编程技巧, 该部分内容安排对应全国计算机等级二级考试。第二部分即第 10 章, 介绍 C++ 面向对象的程序设计基本方法, 讲述 C++ 面向对象的主要特征。为更加突出教材实际应用性, 体现现代教育理念, 融入编者和读者对教材使用的经验和体会, 编者对该书第一版进行了修订再版。在第二版中对部分章节进行编辑修改, 精选调整了部分实训案例, 有利于学习者编程技术的掌握和编程能力的提高; 在各章还增加了项目实训环节, 更加强化学习者运用程序设计语言工具解决实际问题的能力。

本书由张昕任主编, 魏玲、杨雪飞、张昊任副主编, 主要编写人员分工如下: 张昕、杨雪飞、张昊编写第 1、2、4 章及附录, 宋丽敏编写第 3 章, 许友丽编写第 5、6 章, 魏玲编写第 7、8 章, 黄丙根编写第 9、10 章。参加本书案例选择、代码调试等工作的还有: 佟超、邵斌、胡丽霞、邱春丽、刘学普、邢继昕、刘红娟。全书由张昕统稿并定稿, 由金桂兰主审。

本书除了可用作高等院校本、专科学生的教材外, 还兼顾一般读者, 可作为从事计算机软件编程人员的自学参考书。

注: 章节前加 “*” 的内容为 C++ 提供的功能。

全书的实例程序代码确保在 Visual C++ 6.0 和 Turbo C 2.0 环境下调试通过。

由于编写时间仓促, 水平有限, 本书难免有错误之处, 恳请广大读者不吝赐教。

编者

2011 年 10 月

第一版前言

C 语言是应用最为广泛的一种高级程序设计语言,以它独到的优势和特点赢得了编程人员的青睐和信任。C 语言功能丰富,表达能力强,使用灵活方便,程序效率高,是结构化的程序设计语言。C 语言具有低级语言的许多特点,可直接处理字符,进行位运算和指针运算等。因此,C 语言具有很强的实用性,既可用于编写应用软件,也适合于编写系统软件。现在,几乎所有高等院校都将“C 语言程序设计”作为计算机专业必修课程和非计算机专业首选程序设计语言课程。C 语言程序设计也是全国和各省的计算机等级考试的重要考试内容。

本教材融入了编者多年的教学经验,充分考虑到初学者的能力、认知水平、知识结构等因素,着力体现以下特色:实例导入,案例教学,分散难点,突出重点,体现结构化程序设计思想;深入研究 C 语言教学大纲和全国计算机二级考试的要求,通过经典实例阐述 C 语言的概念、语法及程序设计技术,实例与二级试题相结合,使学习者能够顺利通过 C 语言二级考试;程序调试和运行环境为 Windows 平台下的 Visual C++ 6.0,初步介绍了面向对象程序设计语言 C++的基本内容,为今后学习 C++做好铺垫。本教材既能满足学习 C 语言程序设计知识、掌握编程技巧、备考和应考的需要,又能培养学生程序开发设计的能力和解决实际问题的能力。

教材遵照循序渐进、由浅入深的原则,文字叙述清晰扼要、通俗易懂,理论阐述简明科学,选例经典实用、分析透彻浅显,使读者在要点把握时感到具体生动,而不抽象枯涩。在内容安排上分为两个部分,第一部分由第 1 章~第 9 章组成,介绍 C 语言结构化程序设计知识、语法规则及编程技巧,该部分内容安排对应全国计算机等级二级考试。第二部分即第 10 章,介绍 C++面向对象的程序设计基本方法,讲述 C++面向对象的主要特征。

本书由张昕主编,肖荣,金桂兰任副主编,参加本书编写的有:张昕、肖荣、金桂兰、宋丽敏、许友丽、魏玲、黄丙根、姬秀荔等。全书由张昕统稿并定稿,由安志远主审。

本书除了可用作高等院校本、专科学生的教材外,兼顾一般读者,可作为从事计算机软件编程人员的自学参考书。

注:章节前加“*”的内容为 C++提供的功能。

全书的实例程序代码确保在 Visual C++ 6.0 和 Turbo C 2.0 环境下调试通过。

由于编写时间仓促,水平有限,本书难免有错误之处,恳请广大读者不吝赐教。

编者

2005 年 1 月

目 录

第二版前言

第一版前言

第1章 C语言绪论..... 1

1.1 Visual C++ 6.0 环境运行的C程序实例..... 1

1.1.1 C程序实例..... 1

1.1.2 C程序的开发过程..... 2

1.1.3 Visual C++ 6.0 集成环境..... 3

1.2 C程序的基本结构..... 6

1.2.1 C程序的结构特点..... 6

1.2.2 C程序的书写格式..... 8

1.3 C语言的词法..... 8

1.3.1 C语言的字符集..... 8

1.3.2 C语言的词法..... 8

1.4 C语言数据输入和输出..... 9

1.4.1 C语言格式输入/输出..... 10

1.4.2 C语言字符数据输入/输出..... 13

1.4.3 C++的输入输出流..... 14

1.5 程序设计语言的发展..... 15

1.5.1 程序与程序设计语言..... 15

1.5.2 高级语言与C语言..... 16

本章小结..... 17

习题一..... 17

项目实训..... 18

第2章 数据类型及表达式..... 19

2.1 关于数据类型及表达式的C程序实例..... 19

2.2 C语言的数据类型..... 21

2.2.1 基本数据类型..... 21

2.2.2 常量..... 22

2.2.3 变量及其类型定义..... 24

2.3 运算符及表达式..... 25

2.3.1 运算符和表达式..... 25

2.3.2 算术运算符和算术表达式..... 26

2.3.3 关系运算符和关系表达式..... 27

2.3.4 逻辑运算符和逻辑表达式..... 27

2.3.5 条件运算符和条件表达式..... 28

2.3.6 赋值运算符和赋值表达式..... 28

2.3.7 逗号运算符和逗号表达式..... 29

2.3.8 求字节运算符..... 30

2.3.9 类型转换..... 30

本章小结..... 31

习题二..... 32

项目实训..... 34

第3章 结构化程序设计..... 35

3.1 顺序结构程序设计..... 35

3.1.1 顺序结构的C程序实例..... 35

3.1.2 顺序结构语句..... 35

3.1.3 程序实训..... 36

3.2 选择结构程序设计..... 38

3.2.1 选择结构的C程序实例..... 38

3.2.2 if 语句..... 38

3.2.3 switch 语句..... 45

3.2.4 程序实训..... 47

3.3 循环结构程序设计..... 49

3.3.1 循环结构的C程序实例..... 49

3.3.2 while 语句..... 50

3.3.3 do-while 语句..... 51

3.3.4 for 语句..... 52

3.3.5 循环的嵌套..... 54

3.3.6 break 语句和 continue 语句..... 55

3.3.7 程序实训..... 57

3.4 结构化程序设计方法与综合实训..... 59

3.4.1 结构化程序设计方法..... 59

3.4.2 综合实训..... 59

本章小结..... 63

习题三..... 64

项目实训..... 68

第4章 数组..... 69

4.1 数组应用的C程序实例..... 69

4.2 一维数组的定义和引用	70	5.8 编译预处理	113
4.2.1 一维数组的定义	70	5.8.1 宏定义	114
4.2.2 一维数组元素的引用	70	5.8.2 “文件包含”处理	119
4.2.3 一维数组的初始化	71	5.8.3 条件编译	119
4.3 二维数组的定义和引用	71	5.9 综合实训	120
4.3.1 二维数组的定义	71	本章小结	122
4.3.2 二维数组元素的引用	72	习题五	123
4.3.3 二维数组的初始化	72	项目实训	128
4.4 字符数组与字符串	73	第6章 指针与引用	129
4.4.1 字符数组的定义	73	6.1 关于指针的C程序实例	129
4.4.2 字符数组的初始化	74	6.2 指针	130
4.4.3 字符数组的引用	75	6.2.1 地址和指针的概念	130
4.4.4 字符数组的输入输出	75	6.2.2 指向变量的指针变量	132
4.4.5 字符串处理函数	76	6.3 指针与数组	133
4.5 综合实训	78	6.3.1 指向数组元素的指针	133
本章小结	82	6.3.2 通过指针引用数组元素	134
习题四	83	6.3.3 数组名作函数参数	137
项目实训	86	6.3.4 指针与字符数组	141
第5章 函数与预处理	87	6.3.5 数组指针	144
5.1 函数应用的C程序实例	87	6.4 指针与函数	145
5.2 函数的定义	88	6.4.1 指针作函数参数	145
5.2.1 函数的定义	89	6.4.2 函数指针	146
5.2.2 函数原型	91	6.4.3 指针函数	149
5.3 函数调用	91	6.5 多级指针与指针数组	150
5.3.1 函数调用	91	6.5.1 多级指针	150
5.3.2 函数传值调用	92	6.5.2 指针数组	151
5.3.3 函数返回值	95	6.5.3 main函数的参数	152
5.4 函数的嵌套调用和递归调用	98	*6.6 引用	153
5.4.1 函数的嵌套调用	98	*6.7 内存管理	156
5.4.2 函数的递归调用	99	6.8 综合实训	158
*5.5 内联函数和重载函数	103	本章小结	158
5.5.1 内联函数	103	习题六	160
5.5.2 重载函数	103	项目实训	165
5.6 作用域	105	第7章 结构体、联合体和枚举类型	166
5.6.1 作用域	105	7.1 结构体	166
5.6.2 局部变量和全局变量	106	7.1.1 结构体与结构变量	166
5.7 变量及函数的存储类型	108	7.1.2 结构成员的引用	169
5.7.1 变量的存储类型	108	7.1.3 结构数组	170
5.7.2 函数的存储类型	113	7.1.4 结构指针与链表	173

7.2 联合体	186	9.4.2 文件位置指针的检测	223
7.2.1 联合体和联合变量	186	9.5 文件的出错检测	224
7.2.2 联合体的引用及特点	187	9.5.1 文件出错检测函数 <code>ferror()</code>	224
7.3 枚举类型	188	9.5.2 出错标志置零函数 <code>clearerr()</code>	224
7.3.1 枚举类型及变量的定义	188	9.6 综合实训	225
7.3.2 枚举元素的引用	189	本章小结	227
7.4 自定义类型	190	习题九	227
7.5 综合实训	191	项目实训	228
本章小结	194	*第10章 C++面向对象程序设计	229
习题七	194	10.1 C++面向对象的程序实例	229
项目实训	199	10.1.1 C++面向对象的程序实例	229
第8章 位运算	200	10.1.2 C++语言概述	231
8.1 位运算的C程序实例	200	10.2 类与对象	232
8.2 二进制位运算	200	10.2.1 面向对象的基本概念	232
8.2.1 二进制位运算	201	10.2.2 类的构造与封装	232
8.2.2 位复合赋值运算符	205	10.2.3 创建对象	235
8.3 位段	206	10.2.4 友元	236
8.4 综合实训	208	10.2.5 模板	239
本章小结	210	10.2.6 程序实训	240
习题八	210	10.3 继承与派生	241
项目实训	211	10.3.1 继承与派生的概念	241
第9章 文件	212	10.3.2 派生类	242
9.1 文件操作的C程序实例	212	10.3.3 派生类的构造函数与析构函数	243
9.1.1 文件操作的C程序实例	212	10.3.4 虚函数	245
9.1.2 C文件概述	214	10.3.5 程序实训	246
9.1.3 文件类型指针	215	10.4 运算符重载	248
9.2 文件的打开与关闭	215	10.4.1 重载运算符	248
9.2.1 文件打开函数 <code>fopen()</code>	215	10.4.2 运算符重载规则	249
9.2.2 文件关闭函数 <code>fclose()</code>	217	10.4.3 类型转换与转换函数	249
9.3 文件的读写	217	10.5 综合实训	251
9.3.1 单个字符读写函数	217	本章小结	253
9.3.2 字符串读写函数	218	习题十	254
9.3.3 数据块读写函数	219	项目实训	254
9.3.4 格式化读写函数	220	附录A ASCII字符编码一览表	256
9.3.5 整数读写函数	221	附录B C库函数	257
9.4 文件的定位	222	附录C Turbo C 2.0集成环境	261
9.4.1 文件的定位	222	参考文献	269

第1章 C语言绪论

本章导读

C语言是广泛流行的计算机高级语言，它适宜作为系统描述语言。本章通过C程序实例分析，使读者从宏观上整体把握和了解C程序的基本结构、C程序的开发调试过程，对C程序开发设计有较深入的感性认识，进而强化对计算机语言和程序设计的理解，为进一步学习C语言程序设计打下基础。本章学习的主要内容为：

- Visual C++ 6.0 环境运行的C程序实例
- C程序的基本结构
- C语言的词法和输入/输出
- 程序设计语言的发展

1.1 Visual C++ 6.0 环境运行的C程序实例

1.1.1 C程序实例

下面介绍几个简单的C程序，并在Visual C++ 6.0 环境下进行调试和运行。

【例 1.1】输出一行信息：This is a C program.

```
/*程序名为 11_1.cpp*/
#include "stdio.h"                /*文件包含*/
void main()                      /*主函数*/
{   printf("This is a C program.\n"); /*输出信息*/
}
```

【例 1.2】输入两个整数，输出它们的和。

```
/*程序名为 11_2.cpp*/
#include "stdio.h"                /*文件包含*/
void main()                      /*主函数*/
{   int a,b,c;                  /*声明定义变量 a, b 和 c*/
    scanf("%d%d",&a,&b);        /*输入变量 a 和 b 的值*/
    c=a+b;                      /*计算 c 等于 a 与 b 的和值*/
    printf("c=%d\n",c);         /*输出变量 c 的值*/
}
```

【例 1.3】输入两个整数，通过调用函数计算它们的和并输出和值。

```
/*程序名为 11_3.cpp*/
#include "stdio.h"                /*文件包含*/
int sum(int x,int y)             /*定义 sum 函数*/
{   int z;                      /*声明定义变量 z*/
    z=x+y;                      /*计算 z 等于 x 与 y 的和值*/
    return (z);                 /*将 z 的值返回，通过 sum 带回调用处*/
}
void main()                     /*主函数*/
```

```

{   int a,b,c;                        /*声明定义变量 a,b 和 c*/
    scanf("%d%d",&a,&b);             /*输入变量 a 和 b 的值*/
    c=sum(a,b);                      /*调用 sum 函数，将得到的值赋给 c*/
    printf("c=%d\n",c);              /*输出变量 c 的值*/
}

```

以上是三个 C 程序实例的源程序代码。可以看出，每个程序中都有一个名字为 `main` 的函数。

例 1.1 的功能是输出一行信息，输出窗口如图 1-1 所示。

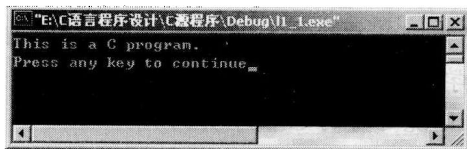


图 1-1 【例 1.1】的输出窗口

其中，“This is a C program.”为程序输出结果，“Press any key to continue”为系统提示，按任意键后返回编辑窗口。

例 1.2 的功能是输入两个整数，输出它们的和。如果输入是：

90<CR> (<CR>表示输入回车键)

7<CR>

输出结果是：

c=97

输出窗口如图 1-2 所示。

其中，90 和 7 分别由键盘输入后按回车键，而“c=97”为输出结果。

例 1.3 的功能是利用调用函数的方法，完成输入两个整数，计算它们的和并输出和值。如果输入是：

90<CR>

7<CR>

输出结果是：

c=97

输出窗口如图 1-2 所示。

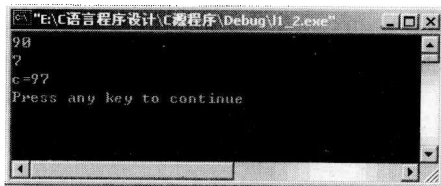


图 1-2 【例 1.2】、【例 1.3】的输出窗口

1.1.2 C 程序的开发过程

C 程序的开发要经过编辑创建源程序、编译源程序、连接目标代码、运行可执行文件等四个过程。

1. 编辑

编辑创建源程序是将 C 语言源程序代码编辑录入到计算机中，形成源程序文件。

录入器是计算机提供的某种文本编辑软件,也可以是C系统提供的编辑器。本书以 Visual C++ 6.0 环境介绍,它提供了一个全屏幕编辑器,使用非常方便。在附录3中介绍了 Turbo C 2.0 的环境情况。在 Visual C++ 6.0 环境中的源程序文件,其扩展名为.cpp,而在 Turbo C 2.0 环境中的源程序文件的扩展名为.c。

2. 编译

由于C语言是一种以编译方式执行的高级语言,C的源程序必须经过编译后才能运行。编译源程序就是由C系统提供的编译器将源程序文件的源代码转换成目标代码的过程。

编译过程主要进行词法分析和语法分析,并将分析结果提示给用户。例如,如果发现错误,错误信息将显示在屏幕上通知用户。经过编译后的目标文件的扩展名为.obj。

3. 连接

连接过程是将编译过程中生成的目标代码进行连接处理,进而生成可执行程序文件的过程。在连接过程中,时常还要加入一些系统提供的库文件代码。经过连接后生成的可执行文件的扩展名为.exe。

4. 运行

运行可执行文件的方法很多,可在C系统下执行“运行”命令,也可以在操作系统下直接执行可执行文件。可执行的程序文件运行后,将在屏幕上显示程序的执行结果。

1.1.3 Visual C++ 6.0 集成环境

Visual C++ 6.0 是目前非常盛行的一种C编译系统,功能十分强大,操作方便,视图界面友好。该系统可在 Windows 9x 及 Windows 更高版本的支持下运行。

1. 启动 Visual C++ 环境

在“开始”菜单中“程序”选项的 Microsoft Visual Studio 6.0 级联菜单下,选择 Microsoft Visual C++ 6.0 命令,启动 Visual C++,进入 Visual C++ 6.0 集成环境。Visual C++ 6.0 主窗口如图 1-3 所示。

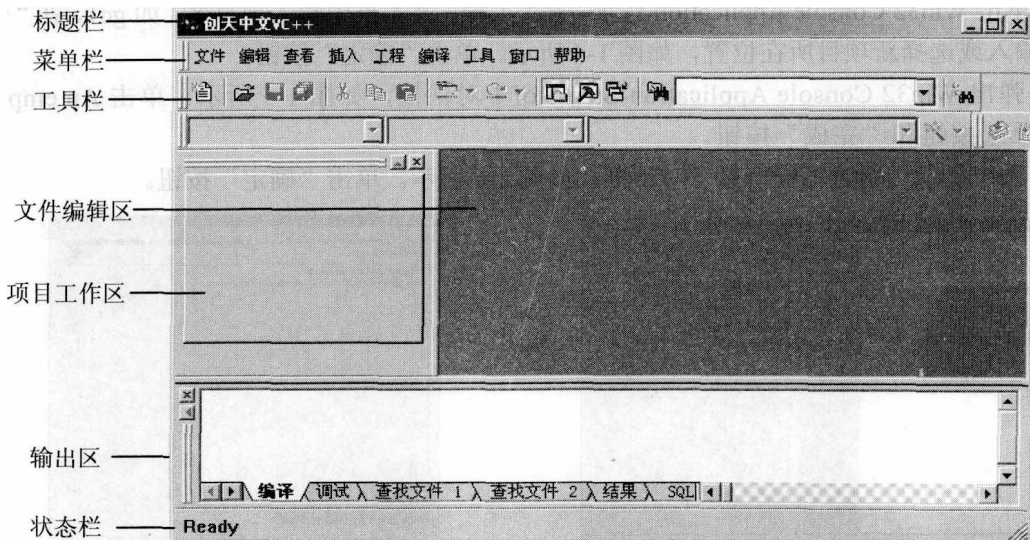


图 1-3 Visual C++ 6.0 集成环境

2. 编辑源程序文件

开发一个新程序需要通过应用程序向导建立新的工程项目，并在项目中添加文件，这些文件可以是源程序文件.cpp 或头文件.h 等，完成源程序的建立编辑。

(1) 建立新工程项目。文件是操作系统处理数据和代码的基本单位，一个 C++ 应用程序由多个文件组成，如源程序文件、头文件、资源文件等。Visual C++ 6.0 用工程化的管理方法把一个应用程序中的所有相互关联的一组文件组织成一个有机的整体，构成一个项目。项目也称为工程。项目以文件夹方式管理所有源文件，项目名作为文件夹名。文件夹中包含源程序代码文件(.cpp.h)、项目文件(.dsp)、项目工作区文件(.dsw)以及项目工作区配置文件(.opt)，还有相应的 Debug（调试）或 Release（发行）、Res（资源）等子文件夹。

建立新工程项目时，程序员通常只编写源程序文件，而项目中其他的文件或文件夹由于使用了系统提供的资源，经过编译、连接由系统自动生成。具体建立过程如下：

- 1) 单击 Visual C++ 6.0 主窗口菜单栏中的“文件”菜单。
- 2) 单击下拉菜单的“新建”命令。弹出“新建”对话框，如图 1-4 所示。

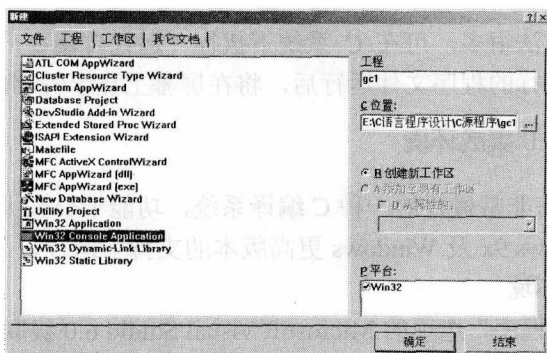


图 1-4 “新建”对话框

- 3) 在“新建”对话框中，单击“工程”选项卡。
- 4) 单击 Win32 Console Application 选项，在“工程”文本框中输入项目名，如 gc1，在“位置”框输入或选择新项目所在位置，如图 1-4 所示。单击“确定”按钮。
- 5) 弹出 Win32 Console Application-Step 1 of 1 对话框，如图 1-5 所示。单击 An empty project 单选按钮和“完成”按钮。
- 6) 系统显示“新建工程信息”对话框，如图 1-6 所示。单击“确定”按钮。

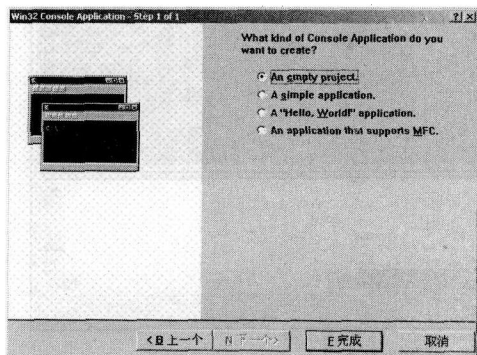


图 1-5 Win32 Console Application-Step 1 of 1 对话框

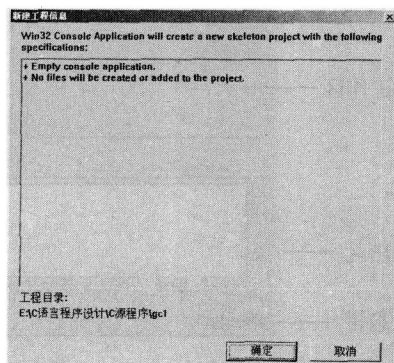


图 1-6 “新建工程信息”对话框

7) 系统自动返回 Visual C++ 6.0 主窗口。项目工作区窗口的内容如图 1-7 所示。

(2) 建立新项目中的文件。若在新建立的项目中创建文件，如源程序文件.cpp 或头文件.h，操作如下：

1) 打开相应的项目文件，选择“文件”菜单下的“新建”命令，在弹出的“新建”对话框中选择“文件”选项卡。

2) 单击 C++ Source File 选项或 C/C++ Header File 选项，在“文件”文本框中输入文件名，如“tl_1”，单击“确定”按钮。

3) 系统自动返回 Visual C++ 6.0 主窗口，并显示文件编辑区窗口。

4) 在文件编辑区窗口输入源程序文件或头文件的内容，如图 1-8 所示。

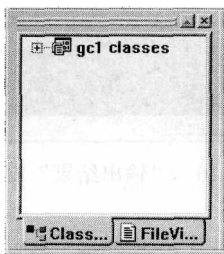


图 1-7 项目工作区窗口

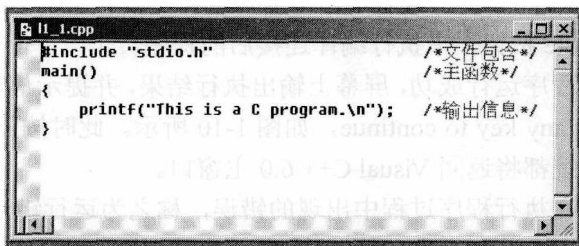


图 1-8 文件编辑区窗口

5) 可重复上述步骤，直到所有文件输入完毕。

若要在项目中添加已经完成的源程序代码文件，具体操作如下：

1) 单击主窗口菜单栏中“工程”菜单的“添加工程”的级联菜单 Files 命令，弹出 Insert Files into Project 对话框，如图 1-9 所示。

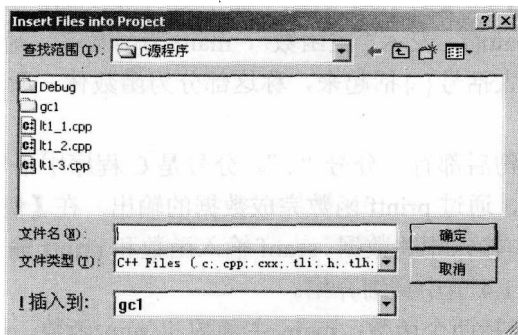


图 1-9 Insert Files into Project 对话框

2) 在“查找范围”框中选择要添加文件所在的位置，单击需要添加的文件，并单击“确定”按钮。或双击需要添加的文件。

3) 可重复上述步骤，直到所有文件添加完毕。

创建编辑完源文件后，还需要将其保存。一种方法是直接单击工具栏中的 Save 按钮 ；另一种方法是单击菜单栏的“文件”菜单下的“保存”命令，实现对源文件的保存。

3. 编译和连接

源程序文件编辑完成后，要进行编译、连接，生成可执行目标代码文件。具体操作如下：

(1) 选择主窗口菜单栏中的“组建”菜单项。

(2) 系统弹出下拉菜单, 选择“编译”命令, 系统开始对程序文件进行编译和连接, 并生成以项目名称命名的可执行目标代码文件.exe。

另一种方法是: 用户可单击主窗口编译工具栏上的 Build 按钮进行编译和连接。

在编译和连接过程中, 系统如果发现程序有语法错误, 则在输入区窗口中显示错误信息, 给出错误的性质、出现位置和错误的原因等。如果双击某条错误, 编辑区窗口右侧出现一个箭头, 指示再现错误的程序行。用户据此对源程序进行相应的修改, 并重新编译和连接, 直到通过为止。

4. 执行

执行程序时, 选择主窗口菜单栏中“编译”菜单中的“执行”命令, 或者单击主窗口编译工具栏上的 Build Execute 按钮来执行编译连接后的程序。

程序运行成功, 屏幕上输出执行结果, 并提示信息: Press any key to continue, 如图 1-10 所示。此时按任意键系统都将返回 Visual C++ 6.0 主窗口。

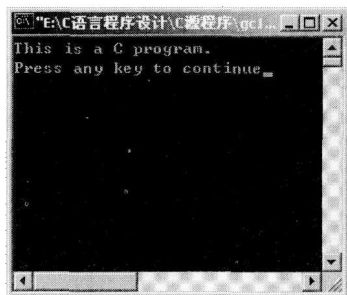


图 1-10 “输出结果”窗口

在执行程序过程中出现的错误, 称之为运行错误, 如出现负数开平方、内存不足等。此时, 用户就要修改源程序文件并且重新编译、连接和执行。

1.2 C 程序的基本结构

1.2.1 C 程序的结构特点

由【例 1.1】、【例 1.2】和【例 1.3】这几个简单 C 程序实例, 分析一下 C 程序的特点。

每一个 C 程序都有 main(), 表示“主函数”, main 为函数名。函数名后必须有一对圆括号 ()。其后面的内容用一对大括号 {} 括起来, 称这部分为函数体。所以每一个 C 程序都必须有一个 main 函数。

函数体中的每一条语句后都有一分号“;”。分号是 C 程序中语句的结束标志。

程序中都有输出语句, 通过 printf 函数完成数据的输出。在【例 1.2】和【例 1.3】中都使用了 scanf 函数来向程序中输入相关数据。scanf 输入函数和 printf 输出函数都是 C 语言提供的标准输入输出函数, 将在 1.4 节中详细介绍。

在【例 1.3】程序中包括两个函数: main 主函数和 sum 函数。main 函数的功能是完成原始数据 a 和 b 的输入, 同时完成函数 sum 的调用以及变量 c 的输出。sum 函数的功能是将 x 和 y 的和值赋给变量 z, 通过 return 语句将 z 的值返回给主函数 main。返回值通过函数名 sum 带回到 main 函数的调用处。

在程序中可以用/*注释内容*/对程序中的任何部分作注释。注释内容不参与程序的编译、连接和执行。在程序中作适当的注释可增加程序的可读性。

通过以上分析, C 程序结构的特点可总结如下:

(1) C 程序是由函数构成的。一个 C 源程序可由一个 main 函数和若干个其他函数组成, 其中必须有一个且只能有一个 main 函数。main 函数又称主函数。因此, 函数是 C 程序的基本单位。被调用的函数可以是系统提供的库函数, 如 printf 和 scanf 函数, C 的函数库十分丰富; 也可以是用户根据需要而编制设计的函数, 如【例 1.3】中的 sum 函数。可以说 C 语言是函数

式的语言，程序的功能都是通过各个函数功能的实现来完成的。编写 C 程序就是编写一个个函数。

(2) 函数由函数说明和函数体组成。

1) 函数的说明。

函数的说明部分即函数的首部，包括函数名、函数返回值类型、函数的形式参数（形参）名、形式参数类型。

例如，【例 1.3】中的 sum 函数的首部为：

int	sum	(int	x,	int	y)
↓	↓	↓	↓	↓	↓
函数返回值类型	函数名	函数参数类型	函数参数名	函数参数类型	函数参数名

函数名后必须有一对圆括号()。函数的参数可有可无，没有参数的函数称为无参函数，如 main 函数，有参数的函数称为有参函数。若有多个参数时，参数之间用逗号隔开。

2) 函数体。

函数体即函数说明部分下面的用一对大括号括起来的部分{……。如果一个函数内有多重大括号，则最外层的一对{}为函数体的范围。函数体一般包括声明部分和执行部分。

声明部分的功能，一是定义函数中用到的变量，在 C 语言中，一个变量在使用之前必须进行定义，如【例 1.3】main 函数中的“int a,b,c;”；二是对该函数所调用的函数进行声明，将在第 4 章介绍。

执行部分的功能是由若干条语句完成的。

函数体中声明部分可以没有，如【例 1.1】。甚至声明部分和执行部分均没有，如：

```
sub()
{ }
```

这种函数称为空函数。它不完成任何功能，一般是程序员为以后程序的功能完善而预留的。

(3) C 程序总是从 main 函数开始执行。main 函数始终是 C 程序执行的入口处，而不论 main 函数在整个程序中的位置，在【例 1.3】中，main 函数虽然放在 sum 函数之后，但程序从 main 函数处开始执行。

(4) C 程序语句和数据定义必须以分号“;”结束。C 语言中，分号是程序语句的结束标志，也是 C 语句的必要组成部分。分号不可或缺。如果语句后丢失分号，则编译系统认为后面内容仍属于本语句，直到遇见分号才结束该语句。程序编译时，系统就会报出错误信息提示。

(5) C 语言严格区分大小写。C 语言编译系统严格区分字母的大小写，如变量 a 和变量 A 系统认定为两个不同的变量。

(6) C 语言数据输入和输出是由库函数实现的。C 语言本身没有输入输出语句。输入和输出操作是由库函数 scanf 和 printf 等来实现的。这样可以增加程序的可移植性。

(7) C 语言用/*注释内容*/形式进行程序注释。在程序中使用注释能对程序进行注解和说明，目的是便于程序的阅读和分析。在“/*”和“*/”之间的所有字符都为注释符，C 系统不对注释符进行编译，它可以放在程序的任一位置。

在 C++ 中，还可以使用“//”来实现注释，从“//”后面字符开始直到它所在行的行尾，所有字符都为注释符，它只能注释一行信息。

1.2.2 C 程序的书写格式

(1) C 程序书写格式自由。对于短语句可以多条语句书写在同一行，也可以将长语句分写在多行。但需要注意，分行时不能将一个单词分开，也不能将双引号引起来的字符串分开。提倡一行一条语句的风格。

(2) 程序书写采用缩进格式。往往根据语句的从属关系，程序书写时采用缩进格式，使程序语句的层次结构清晰，提高程序的可读性。同一层次语句要左对齐，不同层次的语句要缩进若干个字符，这样程序层次清楚，便于阅读和理解。例如，一个函数的函数体语句，应比函数体定界的左大括号缩进 2 至 4 个字符，并且同一层次语句全部左对齐。

(3) 在程序中适当使用注释信息。在书写程序时，适当地加入注释同样可增加程序的可读性。

1.3 C 语言的词法

1.3.1 C 语言的字符集

C 语言的字符集由 ASCII 字符集组成：

(1) 26 个小写字母：a~z。

(2) 26 个大写字母：A~Z。

(3) 10 个数字：0~9。

(4) 其他符号：+、-、*、/、=、,、.、_、)、(、&、^、%、\$、#、@、!、~、<、>、?、'、;、:、"、)、[、]、{、}、-、\、空格。

用 C 字符集中的字符可以构造各种词法符号。ASCII 字符集见附录 1 的 ASCII 字符编码一览表。

1.3.2 C 语言的词法

C 语言使用一组字符来构造具有特殊意义的符号，称之为词法符号。这主要有关键字、标识符、运算符、分隔符、常量及注释符等。

1. 关键字

关键字又称保留字。关键字是系统预定义的词法符号，具有特定的含义，根据语言版本不同，关键字会有所不同。ANSI C 规定有 32 个关键字，具体包括：

auto	break	case	char	const	continue
default	do	double	else	enum	extern
float	for	goto	if	int	long
register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void
volatile	while				

C++ 又补充了 29 个关键字，具体包括：

bool	catch	class	const_cast	delete	dynamic_cast
explicit	false	friend	inline	mutable	namespace

new	operator	private	protected	public	reinterpret_cast
static_cast	template	this	throw	true	try
typeid	typename	using	virtual	wchar_t	

注意: C 和 C++ 均不允许对关键字重新定义, 即程序员不能对这些关键字再定义其他含义。

2. 标识符

标识符是指用来标识程序中用到的变量名、函数名、类型名、数组名、文件名以及符号常量名的有效字符序列。

C 语言标识符的命名必须符合语法规则: 标识符是以字母或下划线开始, 由字母、数字和下划线组成的符号串。

命名标识符还应注意以下几个问题:

(1) 不能使用关键字作为用户标识符。

(2) C 语言严格区分标识符的大小写字母。例如, `sum` 和 `Sum` 是两个不同的标识符。

(3) 标识符的有效字符个数因 C 语言的编译系统而定。有的编译系统允许使用长达 31 个字符的变量名, 而有的编译系统只取变量名的前 8 个字符作为有效字符, 后面的字符将被编译系统忽略, 不被识别。因此, 标识符不宜定义得太长。

(4) 标识符命名时, 尽可能做到见文知义, 选择能够代表一定意义的单词或缩写来标识, 如 `sum` 代表和值, `day` 代表日期等, 以增加程序的可读性。

3. 分隔符

分隔符是程序中的标点符号, 用来分隔单词或程序正文。

(1) 空格: 作为单词之间的分隔符。

(2) 逗号: 作为变量之间或函数的多个参数之间的分隔符。

(3) 冒号: 作为语句标号与语句间的分隔符以及 `switch` 语句中 `case` 与语句序列之间的分隔符。

(4) 大括号: 作为函数体、复合语句等的定界。

4. 其他

注释符前面已经介绍, 不再赘述。

运算符是对数据进行某种操作的单词, 是系统预定义的函数。运算符的具体内容将在第 2 章详细介绍。

常量是程序中由书写形式决定类型和值的数据。C 语言常量有数字常量、字符常量和字符串常量。各种常量将在第 2 章中做详细介绍。

1.4 C语言数据输入和输出

数据的输入输出是用户和程序交流的接口, 任何高级语言都应具备数据的输入和输出功能。C 语言与其他一些高级语言不同, C 语言本身不提供此功能, 而是编译系统在标准函数库中定义了一些输入输出函数, 如前面提到的使用 `scanf` 函数和 `printf` 函数分别实现数据的输入和输出。对单字符数据输入和输出时, 可以使用 `getchar` 函数和 `putchar` 函数实现。但需要注意, 程序在调用这些库函数时, 必须在源文件的开始处加上头文件的“包含命令”, 格式如下:

```
#include <库函数头文件名>
```

或

#include "库函数头文件名"

scanf 函数和 printf 函数包含在标准输入输出头文件 stdio.h 中。因此，对应的文件包含命令为：

#include <stdio.h>

在 C++ 程序中，除了 C 语言中提供的标准输入输出函数外，还增加了特有的输入和输出操作符，称之为流输入和流输出。相应的操作符为 cin 和 cout，它们被定义在 iostream.h 头文件中。因此，在使用 cout 和 cin 前，必须在源文件中加入相应的文件包含命令：

#include <iostream.h>

下面重点介绍 C 语言程序中数据的输入输出方法，同时介绍有关 C++ 数据输入输出的简单方法。

1.4.1 C 语言格式输入/输出

1. 格式输出函数 printf()

printf 函数在前面程序中多次用到，它是格式控制输出函数，其作用是向输出设备按指定的格式输出若干个任意类型的数据。函数 printf() 的使用格式为：

printf("格式控制", 输出项列表)

(1) “输出项列表”是需要输出的一些数据。数据可以有 0 项、1 项或多个输出项，多个输出项之间用“,”分隔，并且输出项应与格式控制相匹配。

(2) “格式控制”是用双引号括起的一串字符，也称格式控制字符串，包括格式说明、普通字符和转义字符 3 种。格式控制字符串的功能是指定输出数据的格式和类型。

下面均以【例 1.2】和【例 1.3】中的 printf("c=%d\n",c); 进行说明。

其中，“c=”是普通字符，“%d”是格式说明，“\n”是转义字符，而 c 是输出项。

下面分别对“格式控制”的 3 种形式进行说明。

1) 普通字符。普通字符内容按照原样向输出设备输出。如“c=”在屏幕上的输出结果就是：
c=

2) 格式说明。格式说明是以字符“%”开头，加上格式字符组成，表示对数据输出格式的控制，它与后面的数据输出项对应。格式说明与数据输出项的数据个数、数据类型及数据排放次序相匹配对应。如，格式说明“%d”与输出项“c”对应，即控制数据输出项 c 按格式说明“%d”规定的格式（十进制整数形式）输出。因此，据 c 值为 97，输出结果是：97。

如果，格式说明“%d”改为“%c”，则输出项 c 的值将按“%c”规定的格式（字符型数据）输出。因此，据 c 值为 97，输出结果是：a。

'a' 为 97 对应的 ASCII 字符。

常用的格式字符情况如表 1-1 所示。

表 1-1 printf 格式字符

格式字符	输出形式	实例	输出结果
d (或 I)	十进制整数	int a=122;printf("%d",a);	122
x (或 X)	十六进制整数	int a=122;printf("%x",a);	7A
o (或 O)	八进制整数	int a=122;printf("%o",a);	172
U	不带符号的十进制整数	int a=122;printf("%u",a);	122