

C语言程序设计

任务驱动教程

燕居怀 ■ 主编

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

C 语言程序设计任务驱动教程

主 编 燕居怀
副主编 崔雪梅 胡彩霞



图书在版编目(CIP)数据

C 语言程序设计任务驱动教程/燕居怀主编. —北京:北京理工大学出版社,2017.7
ISBN 978-7-5682-4355-1

I. ①C… II. ①燕 III. ①C 语言-程序设计-教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 168070 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010)68914775(总编室)

(010)82562903(教材售后服务热线)

(010)68948351(其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 北京泽宇印刷有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 14

字 数 / 330 千字

版 次 / 2017 年 7 月第 1 版 2017 年 7 月第 1 次印刷

定 价 / 57.00 元

责任编辑 / 王玲玲

文案编辑 / 王玲玲

责任校对 / 周瑞红

责任印制 / 李志强

图书出现印装质量问题,请拨打售后服务热线,本社负责调换

前 言

C 语言是当今影响和使用最为广泛的程序设计语言。它结构简单,数据类型丰富,运算灵活方便,用它编写的程序,具有速度快、效率高、代码紧凑、可移植性好等优点。它兼有高级语言和低级语言的特点,既可用于编写应用软件,又可用于编写系统软件,是除汇编语言外执行效率较高的计算机程序设计语言。

根据高等院校教学改革、课程建设的需要,针对高等教育的培养目标和受教育者的特点,学习国外先进的职业教育理念,借鉴先进的教学模式,本教材采用“任务驱动式”,从日常生活中的典型任务入手,将 C 语言的基本知识和技能根据工作任务的需要进行重新整合,精心设计工作任务,由浅入深、逐步推进,使学生能够轻松掌握 C 语言的基本语法知识,逐步提高阅读程序、调试程序、编写程序的技能,真正实现“做中学,学中做”的知识传授。

本教材各章内容按以下形式组织。

学习提示:引出本章的主要学习知识点。

能力目标和知识目标:为教师授课和学生的学习制定了明确的教和学的目标。

教学内容:根据每章知识点的不同,每章引入不同的任务讲解所对应的知识点。每个任务采用“任务说明 + 任务分析 + 知识详解 + 任务实施”的四位一体教学组织模式。其中知识详解模块设置了例题,使学生可以巩固练习所学知识点。任务实施模块采用“算法设计 + 程序代码 + 程序运行结果”,有利于学生分析任务、理解任务,从而激发学生学习兴趣和积极性,也能够培养学生分析问题、解决问题的能力,能够尽快独立地设计 C 程序。

本章小结:总结本章重要的知识点,帮助学生整理复习。

习题:对本章所学内容进行练习和巩固。

本教材的特点是通俗易懂、任务丰富、目标明确、重难点突出、重在应用,以任务驱动为主线,使读者能够轻松愉快地学到相应的知识和技能。本教材不仅可作为高等院校 C 语言课程的教材,还可以作为广大计算机爱好者学习 C 语言或者备考全国计算机等级考试(二级 C)的教材和参考书。

由于编者水平和经验有限,书中难免存在一些不妥之处,恳请广大读者批评指正。

编 者

目 录

第1章 走进C语言	1
任务 制作自己的名片	1
1.0.1 了解计算机语言的发展历史	2
1.0.2 C语言的发展历史	2
1.0.3 C语言的特点	2
1.0.4 C语言程序的结构	2
1.0.5 C语言程序的结构特点	3
1.0.6 C语言的开发过程	4
1.0.7 掌握 Visual C++6.0 上机使用方法	4
1.0.8 程序代码	7
1.0.9 程序运行结果	7
知识拓展	7
本章小结	8
习题1	8
第2章 数据类型、运算符和表达式	10
任务 2.1 计算圆柱体的体积——整型与实型数据,变量与常量	10
2.1.1 基本字符集、关键字、标识符	11
2.1.2 数据类型	12
2.1.3 常量	13
2.1.4 变量	16
2.1.5 算法设计	17
2.1.6 程序代码	17
2.1.7 程序运行结果	17
任务 2.2 分离数字问题——运算符与表达式	18
2.2.1 运算符与表达式	18
2.2.2 算术运算符与算术表达式	18
2.2.3 赋值运算符与赋值表达式	21
2.2.4 位运算与位运算符	22
2.2.5 逗号运算符和逗号表达式	24
2.2.6 数据类型转换	24
2.2.7 算法设计	26
2.2.8 程序代码	26



2.2.9 程序运行结果	26
任务 2.3 求解一元二次方程	27
2.3.1 常用的数学函数	27
2.3.2 算法设计	28
2.3.3 程序代码	28
2.3.4 程序运行结果	29
本章小结	29
习题 2	29
第 3 章 顺序结构	32
任务 3.1 制作算术计算器	32
3.1.1 C 语言的基本语句	32
3.1.2 格式输出函数 printf()	34
3.1.3 格式输入函数 scanf()	37
3.1.4 算法设计	39
3.1.5 程序代码	39
3.1.6 程序运行结果	40
任务 3.2 密码的破解——字符运算	40
3.2.1 字符输出函数	40
3.2.2 字符输入函数	41
3.2.3 算法设计	42
3.2.4 程序代码	42
3.2.5 程序运行结果	42
知识拓展	43
本章小结	44
习题 3	44
第 4 章 选择结构程序设计	46
任务 4.1 判断闰年——if 单分支语句	46
4.1.1 条件判断表达式	46
4.1.2 关系运算符和关系表达式	47
4.1.3 逻辑运算符和逻辑表达式	48
4.1.4 单分支 if 语句	49
4.1.5 算法设计	49
4.1.6 程序代码	50
4.1.7 程序运行结果	50
任务 4.2 判断最大值——if - else 双分支语句	50
4.2.1 双分支 if 语句	50
4.2.2 条件运算符和表达式	51
4.2.3 算法设计	51



4.2.4	程序代码	52
4.2.5	程序运行结果	52
任务 4.3	学生学习成绩评定——多分支 if 语句	53
4.3.1	多分支 if 语句	53
4.3.2	算法设计	54
4.3.3	程序代码	54
4.3.4	程序运行结果	55
任务 4.4	计算分段函数——if 语句的嵌套	55
4.4.1	if 语句的嵌套	56
4.4.2	算法设计	56
4.4.3	程序代码	57
4.4.4	程序运行结果	58
任务 4.5	switch 语句	58
4.5.1	switch 语句的格式	58
4.5.2	算法设计	60
4.5.3	程序代码	60
4.5.4	程序运行结果	61
本章小结	61
习题 4	61
第 5 章	循环结构设计	65
任务 5.1	高斯定理	65
5.1.1	while 语句	65
5.1.2	算法设计	66
5.1.3	程序代码	67
5.1.4	程序运行结果	67
任务 5.2	求平方和	67
5.2.1	do - while 语句	67
5.2.2	算法设计	68
5.2.3	程序代码	69
5.2.4	程序运行结果	69
任务 5.3	统计农场的鸡兔个数	69
5.3.1	for 循环语句	70
5.3.2	while、do - while 和 for 三种循环的比较	72
5.3.3	算法设计	73
5.3.4	程序代码	73
5.3.5	程序运行结果	73
任务 5.4	九九乘法表	74
5.4.1	循环的嵌套	74



5.4.2	算法设计	76
5.4.3	程序代码	77
5.4.4	程序运行结果	78
任务 5.5	break 语句	78
5.5.1	break 语句	78
5.5.2	算法设计	79
5.5.3	程序代码	79
5.5.4	程序运行结果	79
任务 5.6	continue 语句	80
5.6.1	continue 语句	80
5.6.2	算法设计	80
5.6.3	程序代码	81
5.6.4	程序运行结果	81
本章小结		81
习题 5		82
第 6 章	数组	84
任务 6.1	数据的排序——一维数组的定义与引用	84
6.1.1	一维数组的定义	85
6.1.2	一维数组元素的引用	85
6.1.3	一维数组的初始化	87
6.1.4	冒泡排序	87
6.1.5	算法设计	88
6.1.6	程序代码	89
6.1.7	程序运行结果	89
任务 6.2	成绩的管理——二维数组的定义与引用	90
6.2.1	二维数组的定义	90
6.2.2	二维数组的引用	91
6.2.3	二维数组的初始化	92
6.2.4	算法设计	93
6.2.5	程序代码	93
6.2.6	程序运行结果	94
任务 6.3	用字符数组实现密码验证——字符数组与字符串	94
6.3.1	字符数组的定义	95
6.3.2	字符数组的引用	95
6.3.3	字符数组的初始化	95
6.3.4	字符串	96
6.3.5	算法设计	97
6.3.6	程序代码	97



6.3.7 程序运行结果	98
知识拓展	98
本章小结	100
习题6	100
第7章 函数	102
任务 7.1 打印字符图形——函数定义及调用	102
7.1.1 函数的功能及分类	103
7.1.2 函数的定义	104
7.1.3 函数的调用	106
7.1.4 形参与实参	107
7.1.5 算法设计	107
7.1.6 程序代码	108
7.1.7 程序运行结果	108
任务 7.2 构造三角形——有参函数	109
7.2.1 函数的值	109
7.2.2 参数的传递方法	109
7.2.3 算法设计	112
7.2.4 程序代码	112
7.2.5 程序运行结果	113
任务 7.3 排序 - 地址传递	114
7.3.1 参数地址传递方式	114
7.3.2 函数的声明	115
7.3.3 算法设计	117
7.3.4 程序代码	117
7.3.5 程序运行结果	118
任务 7.4 汉诺塔问题——函数的嵌套与递归调用	118
7.4.1 函数的嵌套调用	118
7.4.2 函数的递归调用	120
7.4.3 变量的作用域	122
7.4.4 算法设计	124
7.4.5 程序代码	124
7.4.6 程序运行结果	125
本章小结	125
习题7	125
第8章 指针	128
任务 8.1 交换两个变量的值——指针的定义	128
8.1.1 内存地址与变量	128
8.1.2 指针变量的定义	130



8.1.3	指针变量的初始化	131
8.1.4	指针变量的引用	131
8.1.5	算法设计	131
8.1.6	程序代码	132
8.1.7	程序运行结果	133
任务 8.2	三个数排序——指针变量作为函数参数	133
8.2.1	指针变量作为函数参数	133
8.2.2	算法设计	134
8.2.3	程序代码	134
8.2.4	程序运行结果	135
任务 8.3	字母放大镜——通过指针访问数组	136
8.3.1	数组的指针	136
8.3.2	字符串的指针	137
8.3.3	算法设计	137
8.3.4	程序代码	138
8.3.5	程序运行结果	139
	知识拓展	139
	本章小结	141
	习题 8	142
第 9 章	构造类型	144
任务 9.1	学生信息统计——结构体	144
9.1.1	结构体类型的定义	145
9.1.2	结构类型变量的定义	146
9.1.3	结构变量成员的引用方法	147
9.1.4	结构变量的初始化	148
9.1.5	算法设计	148
9.1.6	程序代码	149
9.1.7	程序运行结果	149
任务 9.2	建立同学通讯录——结构数组和结构指针	149
9.2.1	结构数组	150
9.2.2	结构指针变量的定义和使用	151
9.2.3	结构指针变量作函数参数	155
9.2.4	算法设计	156
9.2.5	程序代码	157
9.2.6	程序运行结果	157
任务 9.3	共用体类型的应用	157
9.3.1	共用体类型及共用体变量的定义	158
9.3.2	共用体变量的引用	160



9.3.3	算法设计	162
9.3.4	程序代码	163
9.3.5	程序运行结果	164
任务 9.4	枚举类型的应用	164
9.4.1	枚举类型及枚举变量的定义	164
9.4.2	算法设计	165
9.4.3	程序代码	166
9.4.4	程序运行结果	167
本章小结	167
习题 9	167
第 10 章	文件	170
任务 10.1	文件的认识	170
10.1.1	文件概述	170
10.1.2	文件的分类	171
10.1.3	文件操作	172
10.1.4	任务设计	175
10.1.5	程序代码	175
10.1.6	程序运行结果	176
任务 10.2	文件的读写与保存	176
10.2.1	向文件读写一个字符	177
10.2.2	向文件读写一个字符串	177
10.2.3	文件格式化读写	177
10.2.4	算法设计	178
10.2.5	程序代码	179
10.2.6	程序运行结果	179
本章小结	180
习题 10	180
习题答案	181
附录 1	C 语言关键字	195
附录 2	ASCII 码表	197
附录 3	C 语言运算符的优先级与结合性	198
附录 4	常用的 C 语言标准库函数	200
参考文献	208

第 1 章

走进 C 语言

学习提示:

在众多的编程语言中,C 语言以其结构化、模块化风格,层次清晰,数据处理能力强,运算符和数据类型丰富,良好的移植性等特点,被广泛应用到各个领域,适用于编写系统软件及应用软件,是风靡全球的计算机语言。

Visual C++6.0 是目前使用最普遍的 C 语言软件,具有方便、直观、易用的界面和丰富的库函数,向用户提供了集成环境,把程序的编辑、编译、连接、运行等操作集中在一个界面进行,使用十分方便。

能力目标:

- 掌握使用 VC++6.0 开发 C 程序的过程
- 使用 VC++6.0 编程环境编写一个简单的 C 程序
- 处理程序中的异常情况

知识目标:

- 了解计算机语言的发展历史
- 了解 C 语言的发展历史及特点
- 掌握 C 语言程序的基本结构及特点
- 掌握 C 语言程序的开发过程

任务 制作自己的名片

● 任务说明

编写一个程序,制作出自己的名片,程序输出信息如下:

```
*****  
*           * 名片 *           *  
*           姓名:小明           *  
*           班级:船电二班       *  
*           电话:123456789       *  
*****
```

● 任务分析

为了创建出第一个 C 语言程序,应首先了解 C 语言的程序结构语句的用法,并熟悉相应的



开发工具的使用。

● 知识详解

1.0.1 了解计算机语言的发展历史

计算机产生后,它最初接受的是由“0”和“1”组成的指令码,这种指令码序列称为机器语言。用机器语言编写的程序,计算机能直接理解并执行,且执行效率高。但是由于机器语言不容易被人理解和记忆,带来了许多不便,所以不易推广。后来又产生了用助记符描述的指令系统,它比机器语言容易理解和记忆,这就是汇编语言。汇编语言与机器语言一样,对机器的依赖性很强,这也束缚了其发展和应用。那么,能否创造一种既接近硬件,又不依赖于机器的类型,同时又使用灵活、功能强大的高级语言呢? C语言就承担了这样的历史重任,并慢慢发展起来了。

1.0.2 C语言的发展历史

C语言是一种过程化的程序设计语言。它的前身是 Martin Richards 于 20 世纪 60 年代开发的 BCPL 语言,这是一种计算机软件人员开发系统软件时作为记述语言使用的程序语言。1970 年,美国贝尔实验室的 Ken Thompson 和 Dennis Ritchie 完成了 UNIX 的初版。与此同时,他们还改写了由 Martin Richards 开发的 BCPL 语言,形成了一种称为 B 的语言。此后, B 语言又进一步被改进和完善,形成了称为 C 的语言。C 语言的发展历史如图 1.1 所示。

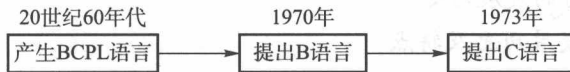


图 1.1 C语言的发展阶段

1.0.3 C语言的特点

与其他语言相比,C语言具有以下主要特点。

①C语言简洁、紧凑,而且程序书写形式自由,使用方便、灵活。

②C语言具有面向用户、可读性强、容易编程和维护等优点,又具有面向硬件和系统并可以直接访问硬件的功能。

③C语言也是一种结构化的程序设计语言。结构化语言的显著特点是程序与数据独立,从而使程序更通用。这种结构化方式可使程序层次更清晰,更便于调试、维护和使用。

④C语言是一种模块化的程序设计语言。所谓模块化,是指将一个大的程序按功能分割成一些模块,使每一个模块都成为功能单一、结构清晰、容易理解的函数,适合大型软件的研制和调试。

⑤C语言的可移植性好。C语言是面向硬件和操作系统的,但它本身并不依赖于机器硬件系统,从而便于在硬件结构不同的机器间和各种操作系统间实现程序的移植。

1.0.4 C语言程序的结构

例如,输出“Hello World! Hello C!”,程序如下:



```
#include <stdio.h>           //文件包含标准输入/输出头文件
int main()                   /* 主函数 */
{
printf("Hello World! Hello C!\n"); //输出
}
```

①语句#include 是预处理命令, <stdio.h> 是标准输入/输出库的头文件,stdio.h 头文件是程序员已经设置好的函数。例如本题中的“printf”,计算机也不知道 printf 是什么意思,要先声明一下,计算机才知道 printf 是要输出括号里面的文字。

文件包含的一般形式如下:

```
#include <文件名> 或者写成 #include "文件名"
```

②main 函数是主函数的函数名。C 语言程序只能有一个 main 主函数,从 main 主函数开始执行,到 main 主函数结束。“{...}”代表函数体,这是计算机要执行的命令部分,每条语句要加“;”结束,main 后面的“()”不可以漏掉,而且绝对不可以加分号。

③函数体中有一条输出语句“printf("Hello World! Hello C!\n");”,其功能是将双引号中的内容“Hello World! Hello C!”原样输出到计算机屏幕上。后面“\n”属于转义字符,其功能是换行,即在输出字符串后自动跳转到下一行(就像键盘上的 Enter 键一样)。

④“//”是单行注释,其主要作用是帮助读者阅读程序,计算机并不执行注释部分的内容,也不检查其语法是否正确,注释可写在程序中的任意位置。除了单行注释,还有多行注释,多行注释主要针对一行写不完,需进行下一行书写的内容,用“/*”和“*/”引起来的内容是注释部分。

1.0.5 C 语言程序的结构特点

通过对 C 程序的认识,总结出 C 程序的如下一些特点。

①C 程序是由函数构成的,一个 C 源程序至少包含一个 main() 函数,也可以包含一个 main() 函数和若干个其他函数。一个 C 程序总是从 main 函数开始执行,到 main 主函数结束。

②一个函数由函数的首部和函数体两部分组成。函数的首部即函数的第一行,包括函数名、函数类型、函数参数名、参数类型;函数体是用一对花括号“{}”括起来的用于完成某种功能的语句的集合,函数体一般包括声明部分和执行部分。

③C 语言中每条语句以分号“;”结束。

④C 程序的书写格式自由,一行内可以写几个语句。

⑤C 语言本身没有输入/输出语句,输入/输出是由函数完成的。

⑥标识符、关键字之间必须至少加一个空格,表示分隔。若已有明显的分隔符,也可以不加空格。

⑦可以用“/*”和“*/”对 C 程序中的任何部分进行注释,而“//注释内容”只能注释一行。

⑧C 语言严格区分大小写字母。C 语言对大小写字母非常敏感,如 main、MAIN、Main 是不同的。在 C 语言中,常用小写字母表示变量名、函数名等,常用大写字母表示符号常量等。



1.0.6 C 语言的开发过程

开发一个 C 程序,一般要经历编辑、编译、连接和运行 4 个步骤,如图 1.2 所示。

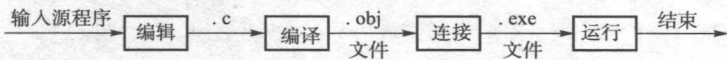


图 1.2 C 语言开发过程

1. 编辑

用户通过编译器,将自己开发的 C 语言程序输入计算机的过程,称为 C 程序源文件的编辑。编辑产生的文件以文本的形式储存,扩展名为 .c,也称为 C 的源程序。源程序文件以 ASCII 码形式存储,计算机不能直接执行。

2. 编译

计算机把 C 的源程序翻译成计算机可以识别的二进制形式的目标代码文件,这个过程称为编译,由 C 的编译程序完成。

C 的编译器程序在编译的同时,还对源程序的语法和程序的逻辑结构等进行检查,当发现错误时,将会列出错误的位置和种类,此时需要重新编译源程序。如果编译成功,则生成目标文件,文件名同源程序文件名,扩展名为 .obj。

当编译生成的目标文件不包含程序运行所需要的库函数等“资源”时,计算机仍然不能直接执行。

3. 连接

连接程序将目标程序和其他程序模块,以及提供的 C 库函数等进行连接,并生成可执行文件的过程,称为连接。连接生成的可执行文件的文件名同源程序文件名,扩展名为 .exe。

4. 运行

双击可执行文件名,都可以获得运行结果。如果运行结果有误,则需要重新编辑源程序,再进行编译、连接、运行,直到得到满意的结果。

1.0.7 掌握 Visual C++6.0 上机使用方法

1. 启动 Visual C++6.0

双击桌面上的 Visual C++6.0 图标,或者选择“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Visual C++6.0”菜单命令,启动 Visual C++6.0,进入 Visual C++6.0 集成环境,如图 1.3 所示。

2. 创建工程

创建工程的步骤如下:

启动 Visual C++6.0 后,选择“文件”→“新建”菜单命令,打开“新建”对话框。在“新建”对话框中选择“工程”选项卡,如图 1.4 所示。然后选择“Win32 Console Application”类型,在“工程名称”文本框中可以输入当天日期,单击“位置”框右边的小按钮,在弹出的对话框中选择“D:\HCQ”目录(为了方便维护,建议在磁盘上建立自己的工作目录,以便存放所开发的源程序文件),单击“确定”按钮。然后出现一个新的对话框,选择“一个空工程”,单击“完成”按



钮,然后单击“确定”按钮,Visual C++6.0 将创建自己的工程。

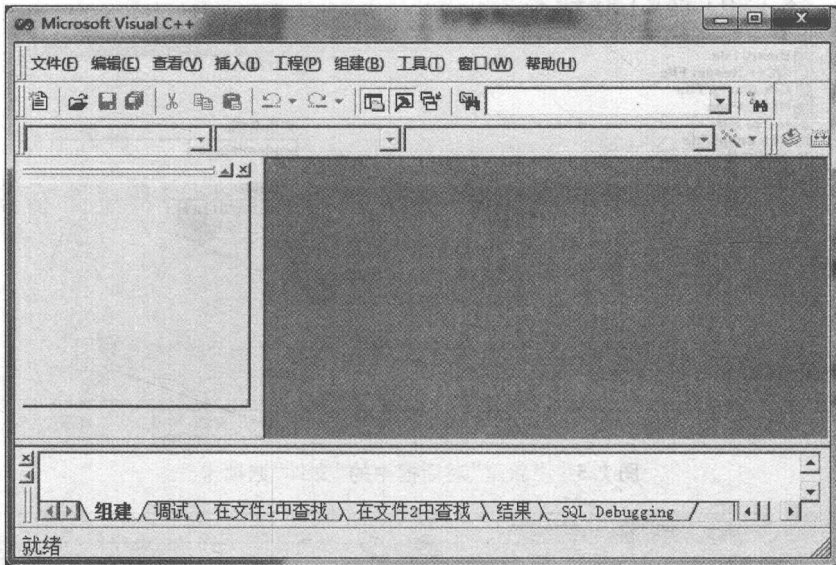


图 1.3 Visual C++6.0 界面

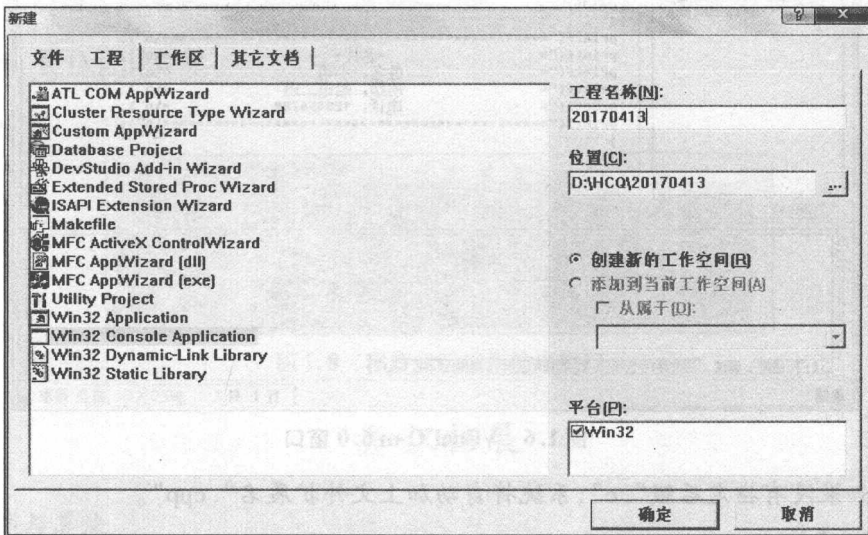


图 1.4 “新建”对话框中的“工程”选项卡

3. 创建、编辑源程序文件

在创建好的工程中,添加源程序文件。操作步骤如下:

选择“文件”→“新建”菜单命令,打开“新建”对话框,选择“文件”选项卡,如图 1.5 所示。选中对话框右侧的“添加到工程”复选框,即把当前要创建的文件加入工程 20170413 中。在“文件”列表框中选中“C++ Source File”,在“文件名”文本框中输入文件名“wodemingpian.c”。单击“确定”按钮,wodemingpian.c 文件将被添加到工程中,即可编辑程序,如图



1.6 所示。

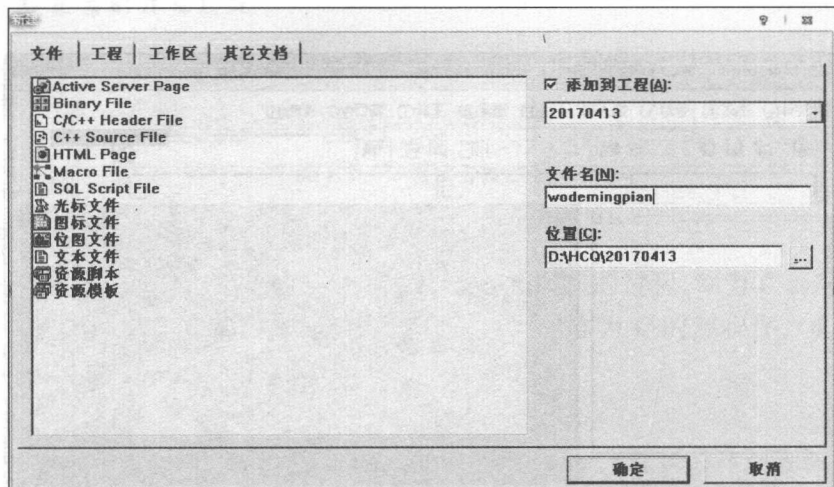


图 1.5 “新建”对话框中的“文件”选项卡

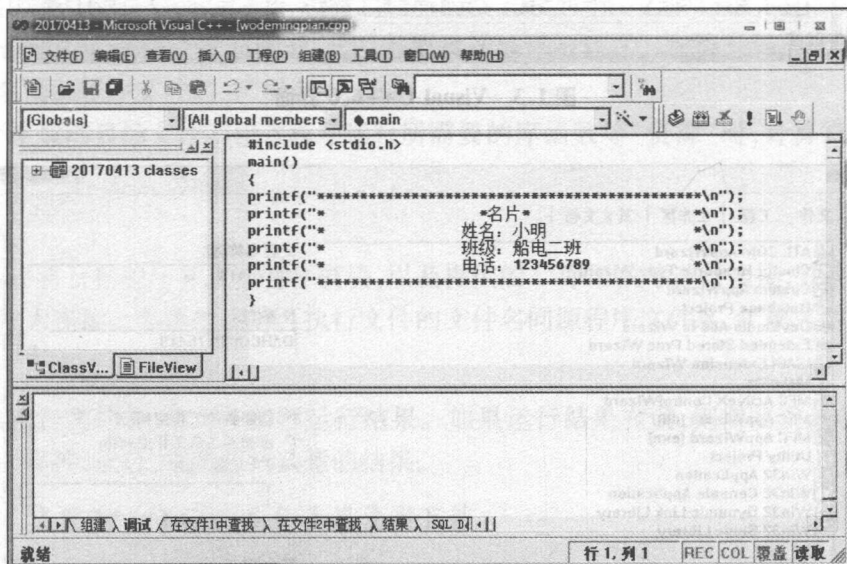


图 1.6 Visual C++ 6.0 窗口

注意:如果没有指定后缀“.c”,系统将自动加上文件扩展名“.cpp”。

4. 编译、连接

选择“组建”→“编译”,查看无错误后运行程序,如图 1.7 所示。也可以使用快捷键 Ctrl + F7。

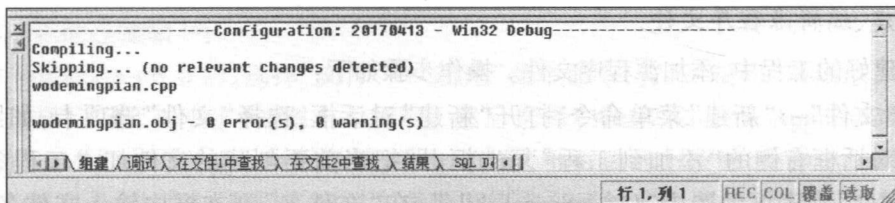


图 1.7 编译、连接的输出窗口