

山东省高校统编教材

C语言程序设计教程

主编 吴振国 张建华



中国石油大学出版社

山东省高校统编教材

C 语言程序设计教程

主 编 吴振国 张建华

副主编 邱俊杰 李 莹 祝凌云

中国石油大学出版社

图书在版编目 (CIP) 数据

C 语言程序设计教程/吴振国, 张建华主编. —东营: 中国石油大学出版社, 2012.6
ISBN 978-7-5636-3709-6

I. ①C… II. ①吴… ②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 (2012) 第 121125 号

书 名: C 语言程序设计教程

主 编: 吴振国 张建华

副主编: 邱俊杰 李 莹 祝凌云

责任编辑: 刘玉兰 (0532-86981535)

出 版 者: 中国石油大学出版社 (山东 东营, 邮编 257061)

印 刷 者: 沂南县汇丰印刷有限公司

电子邮箱: eyi0213@163.com

发 行 者: 中国石油大学出版社 (电话 0532-86983011)

开 本: 185 mm × 260 mm 印张: 17 字数: 435 千字

版 次: 2012 年 6 月第 1 版第 1 次印刷

定 价: 29.60 元

版权所有, 翻印必究。举报电话: 0532-86981535

本书封面覆有带中国石油大学出版社标志的激光防伪膜。

本书封面贴有带中国石油大学出版社标志的电码防伪标签, 无标签者不得销售。

前言

为配合计算机应用基础教学改革,以适应能力为主线,强调理论教学与实验、实训紧密结合教学需求,基于全国高等院校计算机基础教育研究会研究制定的《中国高等院校计算机基础教育课程体系 2008》和《中国高职院校计算机教育课程体系2007》,结合全国计算机等级考试C语言考试大纲和山东省高校非计算机专业计算机等级考试C语言考试大纲,在系统分析国内外有关C语言程序设计教材的优势和不足的基础上,组织编写了本教材。本教材以美国国家标准C语言(ANSI C)为基本内容,以当前广为使用的Visual C++6.0 编译系统为实现环境,全面系统地介绍了C语言程序设计的基本知识和相关技术。

C语言是一种通用的程序设计语言,它的结构简单,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,能够有效地用来编制各种系统软件和应用软件,是当今最为流行的计算机编程入门语言之一。通过本课程的学习,使学生掌握初步的软件开发技术,具有初步的程序设计能力。同时,强化学生使用计算机解决实际问题的意识和提高学生计算思维的能力,为进一步学习其他计算机相关课程打下良好的基础。

参加本教材编写的人员都是长期工作在教学第一线从事C语言课程教学,教学经验十分丰富的教师。作为一本C语言的入门教材,在本教材的编写中,本着重基础、重实用的原则,力求使教材通俗易懂,不过多地在细节上展开叙述,使读者能尽快学会C语言程序设计的基础知识和基本方法。

本教材具有三个主要特点:一是采用问题驱动模式。先明确每一个知识点的目标和任务,从现实生活中有针对性的问题入手,导入知识点的学习,使学生了解程序设计的意义,树立学生的编程信心,最终通过所学的知识解决提出的问题。二是内容生动灵活,实例丰富。从问题分析到算法表示,从程序代码编写到运行结果分析,对整个程序设计的过程作了详细的讲解,这样可以帮助初学者快速提高C语言程序设计的能力。书中通过大量的例题培养学生的学习兴趣,实例由易到难,对所学的内容进行逐步扩展,使学生开阔思路,对问题的解决做到举一反三,为开发实际应用项目打好基础。三是理论适度,深入浅出,实用性强。本书内容适度,结构严谨,知识安排循序渐进,符合认知规律;语言简洁,通俗易懂;每章的思考题可强化学生对基本概念的理解,巩固所学知识,拓宽解决问题的思路。

为便于教师教学和学生自学,本书提供了PPT教学课件、例题的源程序、习题及参考答案,并配有《C语言程序设计实训教程》一书,两本书配套使用,从而使本书知识体系的认知与实践相得益彰。

本书共分 11 章,第 1、5、6 章由吴振国编写,第 2、3、4 章由李莹编写,第 7、10、11 章由邱俊杰编写,第 8、9 章由祝凌云编写,全书由吴振国、张建华统稿。

在本书的编写过程中,得到了山东省教育厅高教处的大力支持,也得到了山东省高校一些计算机教学专家的具体指导,在此一并表示衷心感谢。

限于编者的水平,本教材在内容及文字方面可能存在许多不足之处,希望使用者批评指正,以使本教材在再次修订时得到完善和提高。

编者

2012 年 3 月

目 录

第1章 C语言概述..... 1	2.6.2 算术表达式..... 37
1.1 C语言的发展过程..... 1	2.6.3 自增、自减运算符..... 38
1.2 C语言的特点..... 3	2.6.4 复合的赋值运算符..... 39
1.3 C语言源程序的结构..... 3	2.7 逗号运算符和逗号表达式..... 40
1.4 C程序上机步骤..... 6	本章小结..... 40
1.4.1 C语言程序的执行过程..... 6	课后练习..... 41
1.4.2 使用 Visual C++6.0 编辑和运行程序..... 7	第3章 顺序结构程序设计..... 42
1.5 算法简介..... 12	3.1 C语句概述..... 42
1.5.1 算法的概念..... 12	3.2 C语言中输入、输出功能的实现..... 44
1.5.2 算法的表示..... 13	3.3 单字符输入、输出函数..... 45
本章小结..... 18	3.3.1 putchar()函数(字符输出函数)..... 45
课后练习..... 18	3.3.2 getchar()函数(字符输入函数)..... 46
第2章 数据类型、运算符及表达式..... 19	3.4 printf()函数(格式化输出函数)..... 48
2.1 C语言的数据类型..... 19	3.4.1 格式化输出函数的一般形式..... 48
2.1.1 数据和数据类型..... 19	3.4.2 格式化输出函数的格式控制符..... 50
2.1.2 C语言的数据类型..... 20	3.4.3 格式化输出函数的使用说明..... 54
2.2 常量和变量..... 20	3.5 scanf()函数(格式化输入函数)..... 55
2.2.1 标识符、关键字和保留标识符..... 21	3.5.1 格式化输入函数的一般形式..... 55
2.2.2 常量和符号常量..... 21	3.5.2 格式化输入函数的格式控制符..... 56
2.2.3 变量..... 22	3.5.3 格式化输入函数的使用说明..... 57
2.3 基本数据类型..... 24	3.6 顺序结构程序设计举例..... 59
2.3.1 整型数据..... 24	3.7 编译预处理..... 63
2.3.2 实型数据..... 26	3.7.1 宏定义..... 63
2.3.3 字符型数据..... 28	3.7.2 文件包含..... 65
2.4 各类数值型数据间的混合运算..... 32	3.7.3 条件编译..... 66
2.5 赋值运算符和赋值表达式..... 34	本章小结..... 69
2.5.1 C语言运算符的分类..... 34	课后练习..... 69
2.5.2 赋值运算符..... 35	第4章 选择结构程序设计..... 71
2.5.3 赋值表达式..... 35	4.1 关系运算符和关系表达式..... 71
2.5.4 类型转换..... 35	4.1.1 关系运算符..... 71
2.6 算术运算符和算术表达式..... 37	4.1.2 关系表达式..... 72
2.6.1 算术运算符..... 37	4.2 逻辑运算符和逻辑表达式..... 73
	4.2.1 逻辑运算符..... 73

4.2.2 逻辑表达式	75	6.3.5 字符数组的应用举例	154
4.3 if 语句	76	本章小结	156
4.3.1 单分支 if 语句	76	课后练习	157
4.3.2 双分支 if 语句	78	第 7 章 函数	158
4.3.3 多分支 if 语句	79	7.1 函数的定义与调用	159
4.3.4 if 语句的嵌套	80	7.1.1 函数的定义	159
4.3.5 条件运算符和条件表达式	84	7.1.2 函数的返回值与函数类型	160
4.4 switch 语句	86	7.1.3 对被调用函数的说明和函数原型	162
4.5 选择结构程序设计举例	90	7.1.4 函数的调用	164
本章小结	94	7.1.5 函数的形参与实参	164
课后练习	95	7.2 函数的嵌套调用和递归调用	167
第 5 章 循环结构程序设计	96	7.2.1 函数的嵌套调用	167
5.1 循环概述	96	7.2.2 函数的递归调用	168
5.2 while 语句	97	7.3 数组作为函数参数	170
5.3 do while 语句	99	7.3.1 数组元素作函数参数	170
5.4 for 语句	104	7.3.2 数组名作为函数的形参和实参	171
5.5 三种循环的比较	108	7.4 局部变量与全局变量	176
5.6 循环语句的嵌套	111	7.4.1 局部变量	176
5.7 break 语句和 continue 语句	115	7.4.2 全局变量	177
5.7.1 break 语句	115	7.5 变量的存储类别	180
5.7.2 continue 语句	117	7.5.1 局部变量的存储方式	181
5.8 程序应用举例	118	7.5.2 全局变量的存储方式	185
本章小结	124	7.6 内部函数和外部函数	186
课后练习	125	7.6.1 内部函数	186
第 6 章 数组	126	7.6.2 外部函数	186
6.1 一维数组的定义和引用	126	本章小结	187
6.1.1 一维数组的定义	126	课后练习	187
6.1.2 一维数组元素的引用	127	第 8 章 指针	188
6.1.3 一维数组的初始化	129	8.1 指针的概念	188
6.1.4 一维数组应用举例	133	8.1.1 地址的概念	188
6.2 二维数组	136	8.1.2 指针变量	189
6.2.1 二维数组的定义与使用	136	8.2 指针变量的定义与应用	189
6.2.2 二维数组的初始化	137	8.2.1 指针变量的定义与相关运算	189
6.2.3 二维数组应用举例	139	8.2.2 指针作函数参数	192
6.3 字符数组	141	8.3 指针与数组	195
6.3.1 字符数组的定义	142	8.3.1 通过指针访问一维数组	195
6.3.2 字符数组的初始化	142	8.3.2 通过指针访问二维数组	199
6.3.3 字符数组的输入与输出	144	8.3.3 指向数组的指针作函数参数	200
6.3.4 字符串处理函数	149		

8.4 指针与字符串	202	9.6 共用体.....	227
8.4.1 字符串的表示与引用	202	9.7 枚举	229
8.4.2 指向字符串的指针作函数参数.....	203	9.8 类型定义符 typedef.....	229
8.5 指针数组与指向指针的指针	204	本章小结	230
8.5.1 指针数组	204	课后练习	231
8.5.2 指向指针的指针变量	207	第 10 章 位运算.....	232
8.5.3 命令行参数	207	10.1 位运算符和位运算.....	232
8.6 指针与函数	209	10.2 应用举例.....	236
8.6.1 返回指针值的函数	209	本章小结	237
8.6.2 指向函数的指针	209	课后练习	237
本章小结	210	第 11 章 文件.....	238
课后练习	210	11.1 C 语言文件概述	238
第 9 章 结构体及其他数据类型	211	11.2 文件的打开与关闭	239
9.1 结构体类型的声明.....	211	11.2.1 文件的打开——fopen() 函数	239
9.2 结构体变量的定义与引用.....	212	11.2.2 文件的关闭——fclose() 函数	241
9.2.1 定义结构体变量	212	11.3 文件的读写操作	241
9.2.2 结构体变量的初始化	213	11.3.1 fgetc() 函数与 fputc() 函数	241
9.2.3 访问结构体成员	213	11.3.2 fwrite() 函数与 fread() 函数	244
9.3 结构体数组	214	11.3.3 其他文件读写函数	246
9.3.1 定义结构体数组	214	11.4 位置指针与文件定位	250
9.3.2 结构体数组的初始化	214	11.4.1 rewind() 函数.....	250
9.3.3 标识结构体数组的成员.....	215	11.4.2 fseek() 函数	250
9.3.4 结构体数组的应用	215	11.5 文件检测.....	252
9.4 指向结构体的指针.....	216	本章小结	252
9.4.1 结构体指针的定义与应用.....	217	课后练习	252
9.4.2 通过指针访问结构体数组.....	218	附录 I 《C 语言程序设计教程》考试大纲.....	253
9.4.3 结构体指针作函数参数.....	219	附录 II 常用字符与 ASCII 码对照表	256
9.5 链表	220	附录 III 关键字表	258
9.5.1 链表概述	220	附录 IV 运算符的优先级和结合性	259
9.5.2 动态存储分配	221	附录 V C 语言常用库函数	260
9.5.3 链表的基本操作	222	参考文献.....	264

第1章 C语言概述

本章内容提要

- (1) 计算机语言的发展
- (2) C语言的特点
- (3) C语言程序的基本结构
- (4) 算法和流程图的基本应用
- (5) Visual C++6.0平台的使用

提出问题

问题 1: 什么是计算机语言? 什么是程序?

问题 2: C语言程序如何在计算机中运行?

相关知识

1.1 C语言的发展过程

在现实生活中,做任何一件事情都要有一定的操作流程,通常我们所说的“按程序办事”就是指要按照这一流程一步步去做。程序是指完成某一任务所采用的方法和过程。可以将程序看做一系列动作的执行过程的描述。例如,到银行取款这个任务,我们可以把完成任务的过程描述为:

带上存折去银行→填写取款单→将存折和取款单交给银行工作人员→银行工作人员办理取款手续→拿到钱→离开银行

人们使用计算机,就是要利用计算机处理各种各样的问题。计算机是一个机器,需要有人告诉他们工作任务和完成工作的步骤和方法。为了接受人的指挥,人们为计算机提供了一套完整的指令系统,其中每一条指令对应着计算机能执行的一个基本动作。计算机程序(program)就是为实现特定目标或解决特定问题而编写的指令序列的集合。

人类和计算机之间的沟通需要搭建一个沟通平台,通过这个平台,人们把自己的想法告诉计算机,计算机完成相应的工作任务。计算机语言(Computer Language)就是用于人与计算机之间交流信息的语言系统。为了使计算机进行各种工作,就需要有一套用以编写计算机程序的数字、字符和语法规则,由这些数字、字符和语法规则组成计算机的各种指令(或语句)。这些就是计算机能接受的计算机语言。

计算机程序是通过计算机程序设计语言编写出来的,计算机程序设计语言通常分为三类:机器语言、汇编语言和高级语言。

1. 机器语言

机器语言是用二进制代码表示的计算机能直接识别和执行的一种机器指令的集合。它是计算机的设计者通过计算机的硬件结构赋予计算机操作功能。机器语言具有灵活、直接执行和速度快等特点。

用机器语言编写程序,编程人员要首先熟记所用计算机的全部指令代码。编写程序时,程序员自己处理每条指令和每一数据的存储分配和输入输出,还要记住编程过程中每步所使用的工作单元处在何种状态。这是一件十分繁琐的工作,编写程序花费的时间往往是实际运行时间的几十倍或几百倍。而且,编出的程序全是些 0 和 1 的指令代码,直观性差,还容易出错。现在,除了计算机生产厂家的专业人员外,绝大多数程序员已经不再学习机器语言。

2. 汇编语言

为了克服机器语言难读、难编、难记和易出错的缺点,人们采用和指令代码实际含义相近的英文缩写词、字母和数字等符号来取代指令代码(如用 ADD 表示运算符“+”的机器代码),于是就产生了汇编语言。汇编语言是一种用助记符表示的面向机器的计算机语言,亦称符号语言。由于采用了助记符来编写程序,汇编语言比用机器语言的二进制代码编程要方便些,在一定程度上简化了编程过程。汇编语言的特点是用符号代替了机器指令代码,而且助记符与指令代码一一对应。

由于使用了助记符,用汇编语言编制的程序送入计算机,计算机不能像用机器语言编写的程序那样直接识别和执行,必须通过预先放入计算机的“汇编程序”的加工和翻译,才能变成能够被计算机识别和处理的二进制代码程序。用汇编语言等非机器语言书写好的程序称为源程序,运行时汇编程序要将源程序翻译成目标程序。目标程序是机器语言程序,它能被计算机的 CPU 直接识别和执行。

汇编语言像机器指令一样,仍然是面向机器的语言,使用起来还是比较繁琐费时,通用性也差。但是,汇编语言用来编制系统软件和过程控制软件,其目标程序占用内存空间少,运行速度快,有着高级语言不可替代的用途。

3. 高级语言

机器语言和汇编语言都是面向机器的语言,对机器非常依赖,要求使用者对硬件结构及其工作原理都十分熟悉,这是非计算机专业人员难以做到的,对于计算机的推广应用是不利的。计算机事业的发展,促使人们去寻求一些与人类自然语言相接近且能为计算机所接受的规则明确、自然直观和通用易学的计算机语言。这种与自然语言相近并为计算机所接受和执行的计算机语言称为高级语言。高级语言是面向用户的语言。无论何种机型的计算机,只要配备上相应的高级语言的编译或解释程序,则用该高级语言编写的程序就可以通用。

C 语言就是目前众多计算机语言中最受欢迎的计算机高级语言之一。C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的,并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL (Basic Combined Programming Language) 发展演变而来。1970 年,AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出 B 语言,1971 年在 PDP-11/20 上实现了 B 语言,并编写了 UNIX 操作系统。但 B 语言过于简单,功能有限。1972 年至 1973 年间,贝尔实验室的 Dennis Ritchie 在 B 语言的基础上设计出了 C 语言(取 BCPL 的第二个字母)。C 语言既保持了 BCPL 和 B 语言的优点(精练、接近硬件),又克服了它们的缺点(过于简单、数据无类型等)。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的,随着微型计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些 C 语

言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准协会(ANSI)为C语言制定了一套ANSI标准,成为现行的C语言标准。

目前,在微型计算机上有许多C语言系统的版本可以使用,最流行的有以下几种:

- Borland Turbo C 或称 Turbo C
- Microsoft C 或称 MS C
- Microsoft Visual C++

本教材采用 Microsoft Visual C++6.0 作为C语言系统平台。

1.2 C语言的特点

C语言是一种结构化语言,它层次清晰,便于按模块化方式组织程序,易于调试和维护。C语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型,便于实现各类复杂的数据结构,还可以直接访问内存的物理地址,进行位(bit)一级的操作。由于C语言实现了对硬件的编程操作,因此,C语言集高级语言和低级语言的功能于一体,既可用于系统软件的开发,也适合于应用软件的开发。此外,C语言还具有效率高、可移植性强等特点。

1. 简洁紧凑、灵活方便

C语言一共只有32个关键字、9种控制语句,程序书写自由,主要用小写字母表示。C语言可以像汇编语言一样对位、字节和地址进行操作。

2. 运算符丰富

C语言把括号、赋值、强制类型转换等都作为运算符处理,从而使C的运算类型极其丰富,表达式类型多样化,灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

3. 数据结构丰富

C语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等,能用来实现各种复杂的数据类型的运算,并引入了指针概念,使程序效率更高。另外,C语言具有强大的图形功能、计算功能、逻辑判断功能。

4. C语言是结构化程序设计语言

结构化程序设计语言的特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。另外,C语言源程序是以函数的形式呈现给用户的,C语言系统还提供了丰富的系统库函数,这些函数可以供用户方便地调用。C语言有完整的选择结构和循环控制结构语句,使程序流程控制完全结构化。

5. C语言应用范围广,可移植性好

C语言的一个突出的优点就是适合于多种操作系统,如DOS、UNIX,也适用于多种机型。

1.3 C语言源程序的结构

下面通过几个示例来了解C语言的基本结构,对C语言程序形成一个感性认识。

例1.1 在显示器上输出字符串“I am a student”。

源程序代码如下:

```
#include <stdio.h>

main()
```

```

printf("I am a student\n");
}

```

这是一个简单的 C 语言程序，执行后在用户屏幕上的输出结果是：

例 1.2 从键盘输入三个学生的考试成绩，计算其平均分并输出。

源程序代码如下：

```

#include <stdio.h>
main()
{
    float a,b,c,d;           /*定义 4 个实型变量*/
    scanf("%f%f%f",&a,&b,&c); /*通过 scanf() 函数实现数据的输入*/
    d=(a+b+c)/3;             /*计算 3 个变量的平均值并保存至变量 d 中*/
    printf("average=%f\n",d); /*调用 printf() 函数输出 3 个数的平均值*/
}

```

程序运行结果如下：

例 1.3 输入一个 x 的值，计算 x 对应的正切值。

分析：求一个数的正切值，首先需将其转化为对应的弧度，然后调用 `tan()` 函数便可求得这个数的正切值。

源程序代码：

```

#include <stdio.h>
#include <math.h>
main()
{
    float x;                 /*定义一个实型变量 x*/
    scanf("%f",&x);           /*调用 scanf() 函数进行数据的输入*/
    x=(3.14159*x)/180;        /*将数值 x 转换为对应的弧度*/
    printf("tan(x)=%f\n",tan(x)); /*调用 tan() 函数计算 tan(x) 的值*/
}

```

运行结果为：

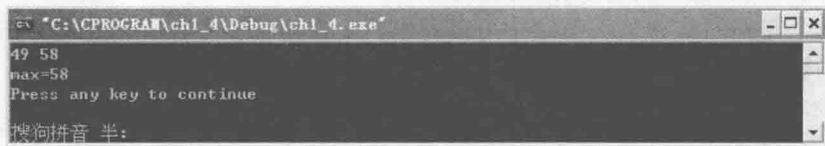
例 1.4 输入两个数，输出其中的大者。

程序源代码如下:

```
#include <stdio.h>
int max(int a,int b)    /*用户自定义函数 max()*/
{
    int c;              /*定义一个整型变量 c*/
    if(a>b) c=a;         /*若变量 a 的值大于 b, 将变量 a 赋值给变量 c*/
    else c=b;           /*若变量 a 的值小于或等于 b, 将变量 b 的值赋给变量 c*/
    return(c);          /*函数调用结束时返回变量 c 的值*/
}

main()
{
    int x,y,z;           /*定义 3 个变量 x、y 和 z*/
    scanf("%d%d",&x,&y); /*调用 scanf() 函数实现数据的输入*/
    z=max(x,y);          /*调用 max() 函数, 并将函数值赋给变量 z*/
    printf("max=%d\n",z);
}
```

程序运行结果如下:



说明:

(1)#include: 以#开始的程序行称为预处理命令行, 在程序的编译之前对这些命令进行处理, 然后将处理的结果和源程序一起再进行通常的编译处理, 得到目标代码。#include 是 C 语言中经常使用的三种预处理命令之一, 它的作用是把一些文件包含到程序中。例如, #include <stdio.h>是把 stdio.h 头文件包含到程序中。

(2)<stdio.h>: 以 h 为扩展名的文件被称为头文件, 它可以是 C 语言系统中的标准库的头文件, 也可以是自定义的文件。在前面的例题中, 我们使用了 stdio.h 和 math.h 两个头文件, stdio.h 文件中包含了标准输入输出函数的声明, 而 math.h 文件中包含了与数学相关的函数声明。在 C 语言中, 调用标准函数库中的函数时, 必须事先在程序中将该函数所在函数库的头文件包含到源程序中。

(3)一个 C 语言程序可以由主函数和其他若干个函数组成, 因此, 我们把 C 语言称为“函数化”的计算机语言。

(4)main(): 主函数是整个程序的入口, 即每个程序运行都是从主函数开始执行的。主函数可以出现在程序的任何位置。

注意: 对于主函数的格式, 目前存在一些争议, 比较常见的格式有以下几种:

- ① main()
 {.....}
- ② void main()
 {.....}
- ③ int main(void)

```
{.....;
    return 0;
}
```

这里存在着 C89 标准和 C99 标准的差异,我们这个教材的所有程序是在 Visual C++6.0 系统下调试运行的,以上三种形式在 Visual C++6.0 中都能正常运行。结合全国计算机等级考试的内容和目前大部分教材上所采用的形式,本教材选用最简单的第一种形式作为主函数的格式。

(5) `int max(int a, int b)`: 用户自定义函数,其中, `int` 表示函数的类型,也就是函数返回值的类型。`max` 是函数的名字, `(int a, int b)` 是函数的形式参数(函数的自变量)。

(6) C 语言中可以包含注释,注释内容可以使程序的可读性大大提高,便于修改和维护,程序编译运行时,编译器并不处理这些注释。注释分为单行注释和多行注释,单行注释用“//”,多行注释使用“/*”和“*/”表示开始和结束。

(7) 从书写清晰,便于阅读、理解和维护的角度出发,在 C 语言程序书写时应注意以下三个方面:

- ① 尽量一行一条语句,使程序简洁。
- ② 编写 C 语言代码一般使用小写字母,符号常量等一些特殊表示用大写字母。
- ③ 注意代码的缩进,使程序有层次感。

从上面几个完整的 C 语言源程序中可以看出, C 语言程序是由函数组成的,任何一个函数(包括主函数 `main()`)都由函数说明和函数体两部分组成。其一般结构如下:

```
[函数类型]  函数名(函数参数表)      函数说明部分
{ 说明语句部分;
  执行语句部分;                      函数体部分
}
```

(1) 函数说明。函数说明由函数类型(可缺省)、函数名和函数参数表三部分组成,其中函数参数表的格式为:

数据类型 形参 [, 数据类型 形参 2.....]

如例 1.4 中的函数 `max()`, 其函数说明中的各部分如图 1-1 所示。

函数类型	函数名	函数参数表
↓	↓	↓
<code>int</code>	<code>max</code>	<code>(int x, int y)</code>

图 1-1 函数说明部分结构图

(2) 函数体。函数体在函数说明部分的下面,是花括号(必须配对使用)内的部分。函数体一般由说明语句和可执行语句两部分构成。其中,说明语句部分由变量定义、类型定义、函数声明、外部变量说明等组成;可执行语句一般由若干条可执行语句(可以是函数调用语句)构成。

1.4 C 程序上机步骤

1.4.1 C 语言程序的执行过程

通过前面的学习,我们对 C 语言已有了初步了解,对 C 语言源程序结构有了总体的认识,

那么如何在机器上运行C语言源程序呢?任何高级语言源程序都要“翻译”成机器语言,才能在机器上运行。“翻译”的方式有两种,一种是解释方式,即对源程序解释一句执行一句;另一种是编译方式,即先把源程序“翻译”成目标程序(用机器代码组成的程序),再经过连接装配后生成可执行文件,最后执行可执行文件而得到结果。

C语言是一种编译型的程序设计语言,它采用编译的方式将源程序翻译成目标程序(机器代码)。运行一个C程序,从输入源程序开始,要经过编辑源程序文件(.C)、编译生成目标文件(.obj)、连接生成可执行文件(.exe)和执行四个步骤,如图1-2所示。

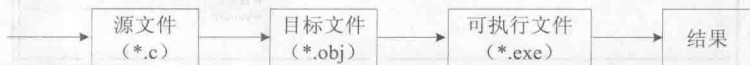


图 1-2 C 语言执行过程

1.4.2 使用 Visual C++6.0 编辑和运行程序

为便于程序的编写、调试和运行,目前计算机语言系统通常以集成开发环境(Integrated Development Environment, 简称 IDE)的形式提供给用户,IDE 系统一般采用窗口菜单技术,提供了专供编程用的编辑环境,通过菜单方式提供编译、连接,以及启动可执行程序的命令。利用 IDE 编写程序,会使开发过程中的各种工作都变得很方便,从而提高编程效率。Visual C++6.0 是目前国内比较流行的一种 C 语言源程序编译系统,使用该系统可以直接编辑和运行 C 语言源程序和 C++源程序。下面具体介绍如何使用 Visual C++6.0 编辑和运行 C 语言程序。

用 Visual C++6.0 创建的 C 语言程序被存储为一个独立的工程,每个工程会新建一个文件夹,工程中包含一组文件,其中的部分文件是由 Visual C++6.0 自动创建的,这组文件组合在一起构成了一个完整的应用程序(也可以建立一个单独的源程序文件。对于简单的 C 语言程序,一般创建一个文件就可以)。创建一个工程涉及的过程和步骤如下:

(1) 打开 Microsoft Visual C++ 软件。

(2) 使用 Microsoft Visual C++ 不仅可以创建控制台应用程序,也可以创建 Windows 应用程序,在这里我们选择创建一个控制台应用程序。在 Microsoft Visual C++ 中选择菜单栏中的“文件”→“新建”选项,弹出如图 1-3 所示界面。

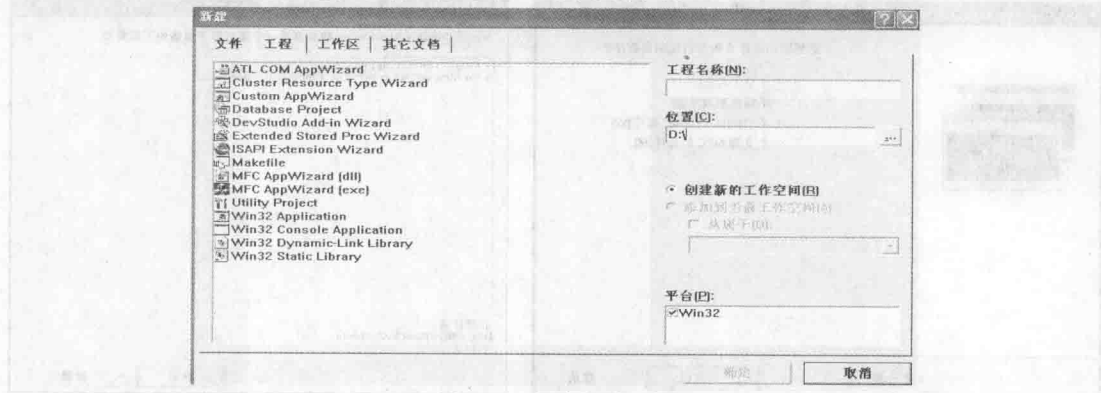


图 1-3 “新建”对话框

(3) 打开“工程”选项卡,单击“Win32 Console Application”选项,如图 1-4 所示。

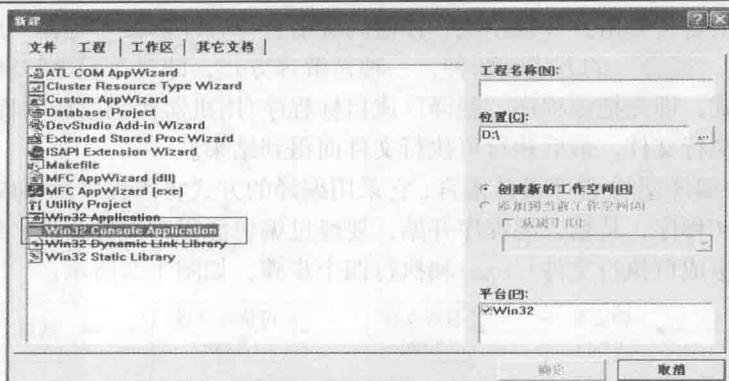


图 1-4 “Win32 Console Application”选项

(4)单击“位置”输入框右侧的按钮，将弹出图 1-5 所示的“选择目录”对话框，选择创建工程的文件夹，单击“确定”按钮。



图 1-5 “选择目录”对话框

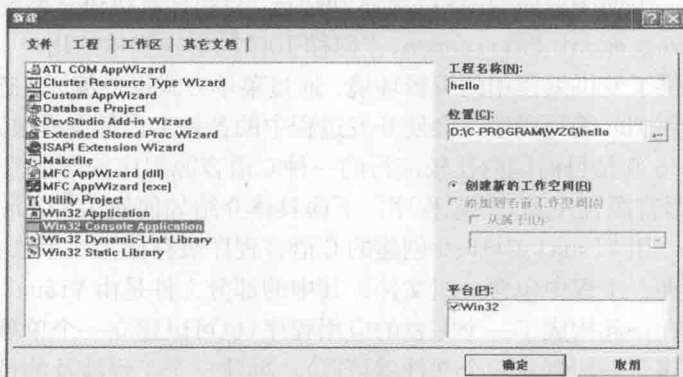


图 1-6 键入工程名称

(5)在“工程名称”输入框中输入名称“hello”，如图 1-6 所示。

(6)单击“确定”按钮，将显示“win32 console application”对话框。选择“一个空工程”单选按钮，如图 1-7 所示。

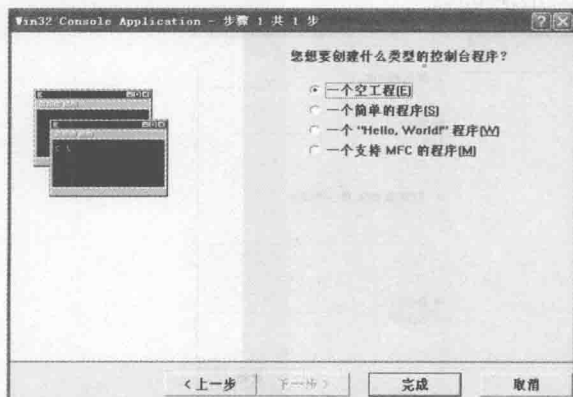


图 1-7 “win32 console application”对话框

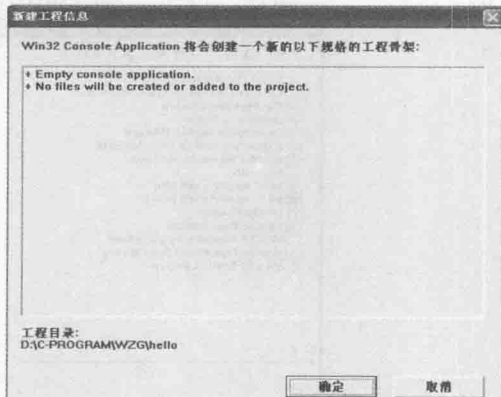


图 1-8 “新建工程信息”对话框

(7)单击“完成”按钮，将显示“新建工程信息”对话框，提示将建立一个空白控制台应用程序，工程中不包含任何文件，如图 1-8 所示。

(8)单击“确定”按钮，将打开图 1-9 所示的 Visual C++ IDE。

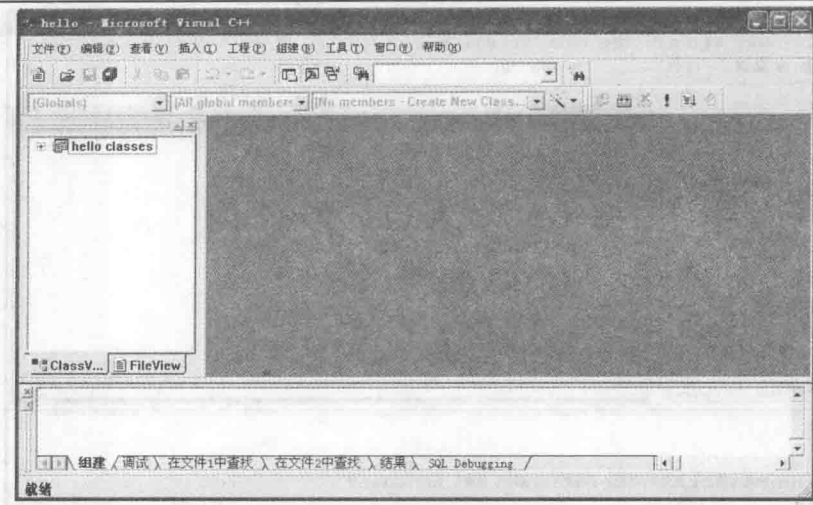


图 1-9 Visual C++ IDE

(9) 选择“文件”→“新建”，打开“文件”选项卡，单击“C++ Source File”选项，如图 1-10 所示。

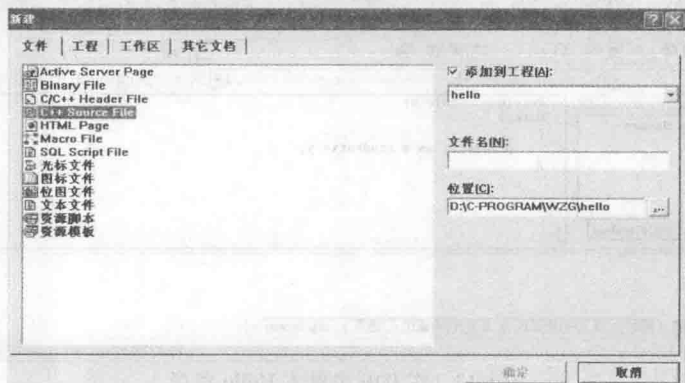


图 1-10 “文件”选项卡

(10) 在“文件名”文本框中键入文件名 hello.c，如图 1-11 所示。

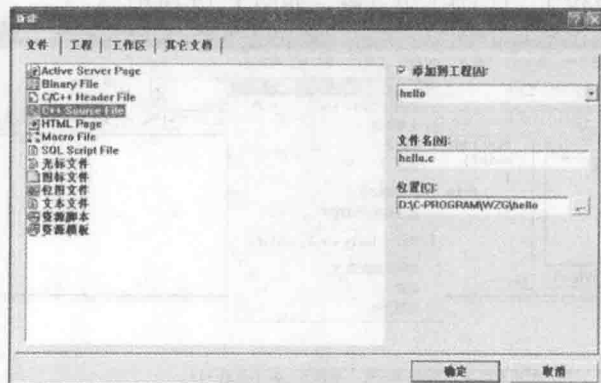


图 1-11 输入文件名 hello.c

(11) 单击“确定”按钮，将显示 Visual C++ IDE 界面，如图 1-12 所示。