



普通高等教育“十二五”规划教材

C语言程序设计与应用

C Programming Language

◎ 李列文 唐宜清 赵娜 著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

普通高等教育“十二五”规划教材

C 语言程序设计与应用

李列文 唐宜清 赵 娜 著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书根据 C 语言程序设计的特点,以培养初学者的编程思维、分析和解决问题的能力、程序设计基本能力为目标,通过丰富的实例详细介绍了 C 语言程序设计的思想及方法。

全书分为 11 章,主要内容包括:C 语言概述、数据类型、运算符与表达式、简单程序设计、输入/输出函数、程序流程控制、数组、函数、指针、结构体、共用体与枚举、位运算、文件和综合应用实例。本书叙述严谨、实例丰富、重点突出、语言简练,每章均配有教学课件和精心设计的习题。

本书可作为高等院校 C 语言程序设计的教材,也可作为从事计算机应用的科学技术人员的参考书及培训教材。

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 语言程序设计与应用/李列文,唐宜清,赵娜著. —北京:电子工业出版社,2015.6
ISBN 978-7-121-26041-4

I. ①C… II. ①李… ②唐… ③赵… III. ①C 语言-程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 097797 号

策划编辑:袁 玺

责任编辑:袁 玺 特约编辑:刘宪兰

印 刷:涿州市京南印刷厂

装 订:涿州市京南印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:15 字数:384 千字

版 次:2015 年 6 月第 1 版

印 次:2015 年 6 月第 1 次印刷

定 价:33.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

C 语言是一门出色的程序设计语言，在国内外都得到广泛的应用。它既具有高级语言程序设计的特点，又具有汇编语言的功能；既能有效进行算法描述，又能对硬件直接进行操作；既适合于开发系统软件，又适合于编写应用程序。目前，C 语言以其功能丰富、表达能力强、目标程序效率高、可移植性好等优点，越来越受到人们的欢迎，在国内外的许多高校，C 语言都被列为学习程序设计的首选语言。

作者长期在高校从事 C 语言课程教学，深知 C 语言学习中的重点和难点，亲身感受到学生在学习过程中的困惑。为了使学生在 C 语言学习过程中抓住重点，始终保持强烈的学习兴趣，掌握并使用 C 语言解决本专业的实际问题，作者对书中的内容和写作方法作了精心考虑。本书以程序设计为主线，以编程应用为驱动，以丰富的实例详细介绍了 C 语言程序设计的思想及方法。本书具有以下特色：

(1) 实践性强。C 语言是一门实践性很强的课程。本书强调上机编写程序是学习 C 语言的最好方法，并且对在 Visual C++ 6.0 和 Turbo C 2.0 两种环境下 C 语言的开发方法都进行了阐述。本书例题已在 Visual C++ 6.0 中上机调试通过。

(2) 系统全面。本书结构清晰、内容全面、示例丰富。既有便于理解的语法内容，又有提高学习兴趣和实际编程能力的实例。考虑到很多程序开发项目需要在 Linux 平台下进行，本书最后一章介绍了 Linux 下的 C 语言开发方法。

(3) 注重综合能力培养。为了加深对所学知识的理解和掌握，本书最后一章介绍了图书管理系统和企业员工信息管理系统的设计过程。这两个综合应用实例涉及了本书中大部分知识点，通过这两个实例可使读者理解和掌握模块化程序设计的基本思路和设计流程，有意识地培养学生设计并实现综合应用系统的能力。

本书共 11 章，第 1 章介绍了 C 语言的基本知识，包括 C 语言的特点，C 语言程序的基本结构及 Visual C++ 6.0 和 Turbo C 2.0 的使用。第 2 章介绍了 C 语言数据类型、运算符与表达式。第 3 章介绍了简单的程序设计，包括 C 语句概述、顺序结构和数据的输入/输出。第 4 章介绍了控制流与程序结构，包括算法语句、选择语句、循环语句和跳转语句。第 5 章介绍了数组，包括一维数组、多维数组、数组的排序算法和字符串处理函数。第 6 章介绍了函数，包括函数概述、函数的定义、函数的调用、变量的作用域与存储类型、内部函数和外部函数及预处理指令。第 7 章介绍了指针，包括指针相关概念、数组与指针、指针与函数和动态存储分配。第 8 章介绍了结构体、共用体与枚举类型。第 9 章介绍了位运算，包括数值在计算机中的表示和位运算。第 10 章介绍了文件，包括文件的概述、文件的打开与关闭、文件的读/写和文件的定位函数。第 11 章系统地介绍了图书管理系统和企业员工信息管理系统的设计过程，包括系统的功能分析、总体设计、功能模块的设计、数据结构设计到实现的全过程，并对 Linux 下 C 语言程序设计方法进行了讲解。

本书第1章、第2章、第4章、附录C、附录D以及第9章的部分内容由唐宜清编写，第3章、第5章、第10章、第11章、附录A和附录B由赵娜编写，第6章、第7章、第8章、第9章由李列文编写。

在本书的编写过程中，彭高丰、王民意对本书提出了许多宝贵意见，在此一并表示衷心的感谢。

由于作者水平有限，书中错误和不当之处敬请读者批评指正。

作 者
2015年6月

目 录

第 1 章	绪论	1
1.1	C 语言的发展历史	1
1.2	C 语言的特点	1
1.3	简单的 C 语言程序介绍	2
1.3.1	程序快速简介	2
1.3.2	程序细节	2
1.4	C 语言程序的格式	2
1.5	使用 C 语言的 7 个步骤	3
1.6	开发环境	5
1.6.1	Visual C++ 6.0	5
1.6.2	使用 Turbo C 2.0 编程	15
	小结	18
	习题	18
第 2 章	数据类型、运算符与表达式	19
2.1	数据类型	19
2.2	变量	19
2.2.1	变量的类型	20
2.2.2	类型修饰符	20
2.2.3	变量的定义	21
2.2.4	存储单元的基本概念	23
2.3	常量	23
2.3.1	整型常量	23
2.3.2	实型常量	24
2.3.3	字符常量	24
2.3.4	字符串常量	25
2.3.5	符号常量	25
2.4	运算符与表达式	26
2.4.1	运算符的优先级和结合性	26
2.4.2	算术运算符和算术表达式	27
2.4.3	关系运算符和关系表达式	30
2.4.4	逻辑运算符和逻辑表达式	31
2.4.5	位运算符	32
2.4.6	赋值运算符和赋值表达式	33
2.4.7	逗号运算符和逗号表达式	33

2.4.8	条件运算符和条件表达式	34
2.4.9	求字节数运算符	35
2.5	不同类型数据之间的转换	35
2.5.1	自动类型转换	35
2.5.2	强制类型转换	36
	小结	37
	习题	37
第3章	简单的程序设计	39
3.1	C 语句概述	39
3.2	顺序结构	40
3.3	输入/输出	41
3.3.1	字符数据输入/输出	41
3.3.2	格式化输入/输出	42
	小结	45
	习题	45
第4章	控制流与程序结构	47
4.1	算法概述	47
4.1.1	算法	47
4.1.2	算法的主要特征	48
4.1.3	算法的表示	48
4.2	选择语句	50
4.2.1	if 语句	51
4.2.2	switch...case 语句	56
4.3	循环语句	58
4.3.1	while 语句	58
4.3.2	do...while 语句	59
4.3.3	for 语句	60
4.3.4	循环语句的嵌套	63
4.4	跳转语句	64
4.4.1	goto 语句	64
4.4.2	break 语句	65
4.4.3	continue 语句	67
	小结	68
	习题	68
第5章	数组	70
5.1	数组概述	70
5.1.1	初始化	71
5.1.2	指定初始化项目	72
5.1.3	为数组赋值	72
5.1.4	数组边界	73

5.1.5	指定数组大小	73
5.2	多维数组	74
5.2.1	二维数组	74
5.2.2	更多维数的数组	75
5.3	数组的排序算法	76
5.3.1	冒泡法排序	76
5.3.2	插入法排序	77
5.3.3	折半法排序	80
5.4	字符串处理函数	82
5.4.1	字符串复制函数	82
5.4.2	字符串连接函数	82
5.4.3	字符串比较函数	83
5.4.4	获得字符串长度函数	84
小结		84
习题		84
第6章	函数	86
6.1	函数概述	86
6.2	函数的定义与调用	87
6.2.1	函数定义	87
6.2.2	函数调用	89
6.2.3	形式参数与实际参数	91
6.2.4	函数的值	93
6.3	函数的嵌套与递归	94
6.3.1	嵌套调用	94
6.3.2	递归调用	95
6.4	变量的作用域	98
6.4.1	局部变量	98
6.4.2	全局变量	99
6.5	变量的存储类型	100
6.5.1	自动变量	100
6.5.2	静态变量	101
6.5.3	寄存器变量	101
6.5.4	外部变量	102
6.6	内部函数和外部函数	103
6.6.1	内部函数	103
6.6.2	外部函数	103
6.7	预处理指令	104
6.7.1	文件包含	104
6.7.2	宏定义	105
6.7.3	条件编译	107

小结	108
习题	109
第7章 指针	112
7.1 指针相关概念	112
7.1.1 地址与指针	112
7.1.2 变量与指针	113
7.1.3 指针变量	113
7.1.4 指针自加自减运算	114
7.2 数组与指针	115
7.2.1 一维数组与指针	115
7.2.2 二维数组与指针	116
7.2.3 字符串与指针	118
7.2.4 字符数组	121
7.3 指针与函数	122
7.3.1 函数返回值是指针	122
7.3.2 指向函数的指针	122
7.4 动态存储分配	124
小结	126
习题	127
第8章 结构体、共用体与枚举	129
8.1 结构体	129
8.1.1 结构体类型的定义	129
8.1.2 结构体变量的定义和初始化	129
8.1.3 结构体变量的引用	132
8.1.4 结构体数组	133
8.1.5 结构指针	135
8.1.6 结构体与函数	137
8.1.7 链表的基本操作	138
8.2 共用体	140
8.2.1 共用体类型的定义	140
8.2.2 共用体变量的说明和引用	140
8.3 枚举类型	141
8.3.1 枚举类型的定义	141
8.3.2 枚举变量的定义和引用	142
8.4 用户定义类型	144
8.5 程序举例	145
小结	148
习题	148
第9章 位运算	150
9.1 数值在计算机中的表示	150

9.1.1	二进制位与字节	150
9.1.2	数的表示主要形式	150
9.2	位运算	151
9.2.1	按位与运算	151
9.2.2	按位或运算	151
9.2.3	按位异或运算	151
9.2.4	按位取反运算	152
9.2.5	左移运算	152
9.2.6	右移运算	152
9.3	位域	153
9.3.1	位域的定义和位域变量的说明	153
9.3.2	位域的使用	154
	小结	155
	习题	155
第 10 章	文件	157
10.1	文件的概述	157
10.1.1	文件的基本概念	157
10.1.2	文件指针	157
10.2	文件的打开与关闭	158
10.2.1	文件的打开	158
10.2.2	文件的关闭	159
10.3	文件的读/写	159
10.3.1	字符读/写函数	159
10.3.2	字符串读/写函数	160
10.3.3	格式化读/写函数	161
10.3.4	数据块读/写函数	162
10.3.5	判断文件结束函数	162
10.4	文件定位函数	162
10.4.1	fseek 函数	162
10.4.2	rewind 函数	162
	小结	162
	习题	163
第 11 章	C 语言项目实践	164
11.1	图书管理系统	164
11.1.1	系统功能与系统总体设计	164
11.1.2	功能模块的设计	164
11.1.3	数据结构设计	165
11.1.4	函数功能描述	165
11.1.5	程序演示	166
11.1.6	程序清单	167
11.2	企业员工信息管理系统	187

11.2.1	系统功能总体设计	187
11.2.2	功能模块的设计	187
11.2.3	数据结构设计	188
11.2.4	函数功能描述	189
11.2.5	程序演示	189
11.2.6	程序清单	191
11.3	Linux 下 C 语言程序开发	207
11.3.1	Linux 简介	207
11.3.2	Linux 安装与常用命令	208
11.3.3	Linux 常用命令	214
11.3.4	编辑器 vi 的使用	215
11.3.5	Linux 下的 C 语言开发流程	217
	小结	218
附录 A	常用 ASCII 码对照表	219
附录 B	C 语言中的关键字	220
附录 C	运算符的优先级和结合性	221
附录 D	常用 ANSI C 语言标准库函数	223
参考文献		229

第 1 章 绪 论

1.1 C 语言的发展历史

C 语言是在 20 世纪 70 年代初问世的。1973 年，美国贝尔实验室的 Dennis M. Ritchie 设计发明了 C 语言，正式改写了 UNIX 操作系统的核心并在 DEC PDP - 11 计算机上使用。1978 年，Brian W. Kernighian 和 Dennis M. Ritchie 合作出版了名著《The C Programming Language》的第一版，该书中介绍的 C 语言标准也被 C 语言程序员称作“K&RC”，也称为《K&R》标准。

然而，在《K&R》中并没有定义一个完整的标准 C 语言，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，美国国家标准研究协会（American National Standards Institute: ANSI）在 1988 年为 C 语言制定了一套 ANSI 标准，成为现行的 C 语言标准，从此 C 语言发展迅速，成为了目前世界上最广泛流行的高级程序设计语言。

1.2 C 语言的特点

C 语言发展如此迅速，而且成为最受欢迎的语言之一，主要因为它具有强大的功能。许多著名的系统软件，如 PC - DOS、DBASE IV 都是用 C 语言编写的。归纳起来 C 语言主要具有下列六个特点。

(1) C 是结构式语言。结构式语言的显著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、维护及调试。C 语言是以函数形式提供给用户的，这些函数可方便地调用，并具有多种循环、条件语句控制程序流向，从而使程序完全结构化。

(2) C 语言允许直接访问物理地址，对硬件进行操作。

由于 C 语言允许直接访问物理地址，可以直接对硬件进行操作，因此既具有高级语言的功能，又具有低级语言的许多功能，能够像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元，可以用来写系统软件。

(3) C 语言程序生成代码质量高，程序执行效率高，一般只比汇编语言编写的程序生成的目标代码效率低 10% ~ 20%。

(4) C 语言适用范围大，可移植性好。它有一个突出的优点就是适合于多种操作系统，如 DOS 和 UNIX，也适用于多种机型。

(5) 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、结构体类型和共用体类型等。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，使程序运行效率更高。另外 C 语言具有强大的图形功能，支持多种显示器和驱动器，且计算功能、逻辑判断功能强大。

(6) 运算符丰富。C 语言的运算符包含的范围很广泛，共有 34 种运算符。C 语言把括号、赋值、强制类型转换命令等都作为运算符处理。从而使 C 语言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。

1.3 简单的 C 语言程序介绍

学习一门新的编程语言的最好方法是运用它来编写程序。对于所有的初学者来说，编写的第一个程序几乎都是相同的，可从例 1.1 中初步了解组成一个 C 语言源程序的基本部分和书写格式。

1.3.1 程序快速简介

【例 1.1】

```
#include <stdio.h>           //引入标准头文件
int main()                   //主程序,程序进入点
{
    printf("Hello,world!\n");  /* 输出文字与换行符 */
    return 0;                 //程序正常结束
}
```

该程序的作用是在屏幕上输出一行信息：“Hello, world!”。

1.3.2 程序细节

(1) 预处理指令：在 C 语言中，以 # 开头，并且放在源文件开头的指令属于预处理指令，表示在编译器编译程序之前需要进行处理的命令。预处理命令通常应放在源文件或源程序的最前面。

(2) 头文件：表示所要调用的函数或其他内容在哪个头文件中，在程序的开始部分就应加上相应的头文件。在例 1.1 这个程序中，包含了标准输入/输出头文件“stdio.h”，用来交互处理输入/输出的信息。

(3) 注释：C 语言支持两种方式的注释，一种是单行注释，在一行中以“//”开始，后续的内容为注释；另一种是多行注释，也叫块注释，以“/*”开始，以“*/”结束。注意：程序不执行注释部分。

(4) main 函数：一个 C 语言程序的运行是从 main 函数开始的。本例中，int main() 表示的是主函数。函数是 C 语言程序的基本单位，一个 C 语言程序可由一个或多个函数构成，但有且只能有一个 main 函数，这也是程序运行的入口点。

(5) 返回语句：表示程序运行结束后，要返回到程序运行的起始点。一般操作系统，会告诉程序运行是否正常结束。例 1.1 中的“return 0”语句就是完成这个功能的。

(6) 函数体：表示程序完成的主要功能的代码部分，用一对大括号（{}，必须配对使用）括起来。函数体一般由说明语句和可执行语句两部分构成。{} 在一个 C 语言程序中是必不可少的，如果函数体内有多个 {}，则最外层是函数体的范围。

1.4 C 语言程序的格式

(1) C 语言程序一般是一行写一个语句，也可以一行写几个语句，还可以一个语句写几行。

一般情况下，一个语句写在多行上不用续行符。但是，有的编译系统有时需要用续行符进行续行，即在续行的前一行末尾加上反斜线（\）。另外，C 语言的语句不需加行号，只有在 goto 语句转向到的语句前需加语句标号。

(2) 每个语句（包括说明语句和执行语句）的末尾必须有一个分号（;）。分号是语句的组成部分，不可缺少，即使程序中最后一个语句也要加分号。要注意：如果一个语句没有结束，在换行前不要加分号。

(3) 一个 C 语言程序是由函数构成的。程序中 main 函数就是其中的主函数，当然，在程序中是可以定义其他函数的。如果将所有的执行代码全部放入 main 函数，程序也可行，但是如果将其分成块，每一块使用一个函数进行表示，在这些自定义函数中进行特殊的操作，使得函数完成特定的功能，那么整个程序看起来就具有结构性，并且易于观察和修改。

(4) 函数体的内容在“{}”中，C 语言程序使用大括号来表示程序的结构层次范围。一个完整的程序模块要用一对大括号括起来，以表示该程序模块的范围。编程时要注意左右大括号要对应使用。

(5) 在使用 C 语言开发程序时，习惯上使用英文小写字母书写，这样易于观察，当然也可以用大写字母，通常定义常量时会使用大写字母，或在定义函数时，有时也会将首字母大写。

(6) 在书写程序中可使用“/* ... */”或“//”对任何部分注释，以增加程序的可读性。

(7) 为了增强程序的可读性，可以使用适当的空格和空行。但是，变量名、函数名和 C 语言保留字中间不能加入空格。除此之外的空格和空行可以任意设置，C 语言编译系统会忽略这些空格和空行符。

1.5 使用 C 语言的 7 个步骤

C 语言是一种编译性语言。一般来说，C 语言程序开发要经历以下 7 个步骤：

- (1) 定义程序目标；
- (2) 设计程序；
- (3) 编写代码；
- (4) 编译与链接；
- (5) 运行程序；
- (6) 调试程序；
- (7) 维护和修改程序。

要注意的是，这只是理想化的状态。在实践中，特别是在较大的项目中，可能需要做一些反复工作，用后一步骤所了解到的内容来改进前一个步骤的内容。

第 1 步：定义程序目标

在开始进行 C 语言程序开发时，应对需要程序做什么有一个清晰的想法。要考虑程序需要的信息，程序需要进行的计算和操作，以及程序的输出信息。

第 2 步：设计程序

在对程序需要完成的目标有一个概念性的规划后，应该决定程序要如何完成它，如用户界面应该是怎么样的，程序应该如何组织，目标用户是谁等问题。

另外，还需要确定在程序中如何表示数据，以及用什么方法来处理数据。在刚开始学习 C

语言编程时，这些选择是比较简单的，但是在处理复杂的情况时，就会需要对这些问题进行更多的思考。选择一个好的方式来表示信息通常可以使程序和数据处理变得更容易。

第 3 步：编写代码

在程序有了清晰的设计后，就可以通过编写代码来实现它。使用字处理软件或编辑工具将源程序以文本文件形式保存到磁盘，源程序文件名由用户自己选定，但扩展名必须为“.c”。现在常用的编程环境如 VC++ 和 Turbo C 等，都支持直接编辑文件。

第 4 步：编译与链接

编译器将已编辑好的源程序翻译成二进制的目标代码，并对源程序进行检查，如果源程序没有语法错误将产生一个与源程序同名、以“.obj”为扩展名的目标程序。

编译后产生的目标程序不能直接运行，应该把目标文件与其他目标文件、系统提供的库函数和操作系统的资源链接到一个可执行的文件中，生成以“.exe”为扩展名的可执行程序。

第 5 步：运行程序

可执行程序生成后，就可以在操作系统的支持下运行。可以在编程环境中单击“运行”命令来运行，也可以在操作系统中单击相应的文件直接运行。

第 6 步：调试程序

如果程序运行后，没有达到预期的目标，则要进一步检查，通过调试来发现并修正程序中的错误（包括语法错误、运行错误和逻辑错误等）。编译器可以找出多种错误。我们可以根据编译器的提示来找出错误并更正。

第 7 步：维护和修改程序

在创建完程序后，如果需求有变化或是硬件环境有变化，则需要对程序进行部分或全部修改，如修正错误、新增功能、提升性能或改变其他属性等。如果前期对程序作了清楚的文字注释并采取良好的设计做法，则所有这些任务都会大大简化。

总结

编程工作通常不像上面讲述的过程那样简单，有时必须在不同步骤间来回反复。例如，当编写代码时，可能会发现之前定义的程序目标是不切实际的；或者在看到程序的运行后，会发现更好的程序实现方式，让人有了改变程序设计的想法。因此，在编程的过程中需要做好记录，以有助于在各阶段之间反复改动。

很多初学者都往往会忽视第 1 步和第 2 步（定义程序目标 and 设计该程序）而直接到第 3 步（编写程序）。这对极简单的程序似乎可以，但是随着程序变得更长、更复杂时，错误将变得难以发现，从而导致程序质量不高，程序开发效率低下等问题。因此，养成在编写代码前先进行规划的习惯是非常必要的，程序越长越复杂，需要的计划工作量就越大。

1.6 开发环境

常用 C 语言程序开发的工具有 Visual C++ 6.0 和 Turbo C 2.0。下面将对这两种开发工具的使用方法进行具体介绍。

1.6.1 Visual C++ 6.0

1. Visual C++ 6.0 简介

在使用一个开发工具前,首先需要熟悉开发工具所提供的开发环境,这样就可以利用开发环境提供的功能快速开发应用程序。Visual C++ 6.0 是微软公司推出的一个功能强大的可视化软件开发工具,它集程序的代码编辑、程序编译、链接和调试等功能于一身,这使得开发过程更快捷和方便。

Visual C++ 6.0 开发环境界面如图 1-1 所示。

1) 菜单栏

Visual C++ 6.0 开发环境的菜单栏中包括 File、Edit、View、Insert、Project、Build、Tools、Window 和 Help 菜单,下面分别进行介绍。

☑ File 菜单

File 菜单中包含用于对文件进行各种操作的命令,如图 1-2 所示。

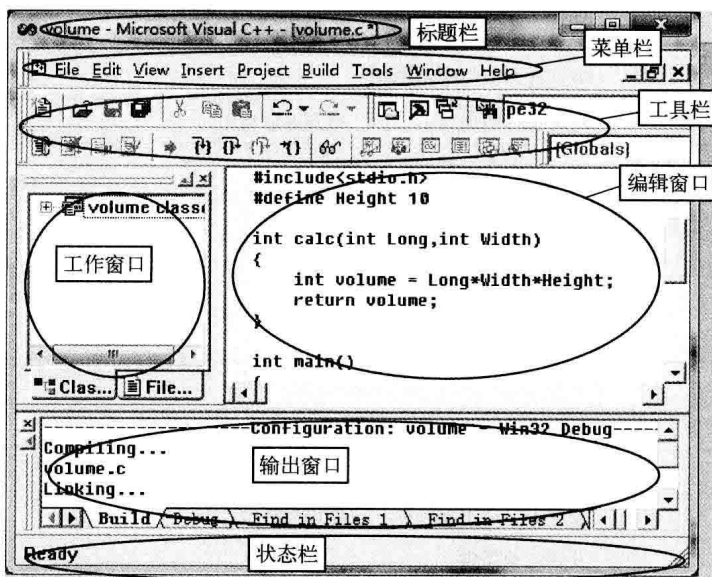


图 1-1 Visual C++ 6.0 开发环境界面

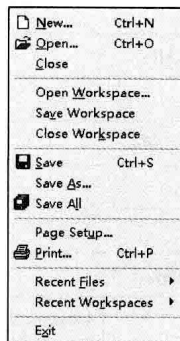


图 1-2 File 菜单

File 菜单中各命令的功能说明如表 1-1 所示。

☑ Edit 菜单

Edit 菜单中包含所有与文件编辑有关的命令,如复制、粘贴等,如图 1-3 所示。Edit 菜单中各命令的功能说明如表 1-2 所示。

表 1-1 File 菜单中各命令的功能说明

命 令	功 能 说 明
New	创建新的文件、工程和工作区
Open	打开一个已存在的文件、工程和工作区
Close	关闭当前打开的文件
Open Workspace	打开一个已存在的工作区
Save Workspace	保存当前打开的工作区
Close Workspace	关闭当前打开的工作区
Save	保存当前打开的文件
Save As	将当前文件另存为一个新文件名
Save All	保存所有打开的文件
Page Setup	对输出文件进行页面设置，在输出机安装后方有效
Print	输出文件的全部或选定的部分，在输出机安装后方有效
Recent Files	最近打开的文件列表，用户可以查看或重新打开
Recent Workspaces	最近使用的工作区，用户可以查看或重新打开
Exit	退出开发环境

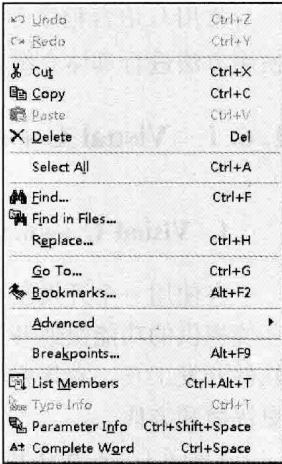


图 1-3 Edit 菜单

表 1-2 Edit 菜单中各命令的功能说明

命 令		功 能 说 明
Undo		撤销上一次的编辑操作，即使保存了文件，该操作仍然有效
Redo		恢复被取消的编辑操作
Cut		将所选择的内容剪切掉，移到剪贴板中
Copy		将所选内容复制到剪贴板中
Paste		在当前位置插入剪贴板中最新的内容
Delete		删除被选择的内容
Select All		选择当前窗口中的全部内容
Find		查找指定的字符串
Find in Files		在多个文件中查找指定的字符串
Replace		替换指定的字符串
Go To		将光标移到指定的位置
Bookmarks		设置书签或书签导航，方便以后查找
Advanced	Incremental Search	开始向前搜索
	Format Selection	对选择对象进行快速缩排
	Untabify Selection	在选择对象中用空格代替跳格
	Tabify Selection	在选择对象中用跳格代替空格
	Make Selection Uppercase	把选择部分的字母改成大写
	Make Selection Lowercase	把选择部分的字母改成小写
	a - b View Whitespace	显示或隐藏空格