

C语言程序设计实用教程

齐景嘉 蒋巍 主编
郭海霞 蔡庆平 郭海龙 副主编
郭川军 主审



清华大学出版社

C语言程序设计实用教程

齐景嘉 蒋巍 主编
郭海霞 蔡庆平 郭海龙 副主编
宋达 李钟隽 编著

内 容 简 介

本书共分 12 章: C 语言概述,数据类型,基本语句与结构化程序设计,数组,函数,指针,编译预处理,结构体、共用体和枚举类型,文件以及位运算。

本书汇集编者多年讲授 C 语言程序设计课程的教学经验。全书体系完整,内容由浅入深,条理清晰,语言流畅;实例丰富,紧扣知识点,并以 Visual C++ 6.0 为程序平台,可操作性强;注重编程方法与技巧的讲解,重视对编程能力的培养。

本书适合作为高等学校计算机及相关专业 C 语言程序设计课程的教材,也可作为计算机等级考试参考书,还可供从事计算机软件开发人员参考使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计实用教程/齐景嘉等主编.--北京:清华大学出版社,2014

21 世纪高等学校规划教材·计算机科学与技术

ISBN 978-7-302-35515-1

I. ①C… II. ①齐… 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(2014)第 032447 号

责任编辑:付弘宇

封面设计:傅瑞学

责任校对:时翠兰

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京四季青印刷厂

装 订 者:三河市溧源装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:16.75

字 数:410 千字

版 次:2014 年 4 月第 1 版

印 次:2014 年 4 月第 1 次印刷

印 数:1~2000

定 价:29.50 元

产品编号:058365-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程”(简称“质量工程”),通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人: 魏江江

E-mail: weijj@tup.tsinghua.edu.cn

前言

本书是黑龙江省高等教育教学改革项目(项目名称:计算机科学与技术专业金融应用人才培养模式的研究与实践;项目编号:JG2012010240)的部分研究成果。

C语言作为一门最通用的程序设计语言,在过去很流行,将来依然会如此。从C语言的诞生到现在,它已经成为最重要和最流行的编程语言之一。熟练掌握C语言是每一位计算机技术人员必备的基本功之一。C语言功能丰富,表达能力强,目标程序效率高,可移植性好,既具有高级语言的强大功能,又有很多直接操作计算机硬件的功能(这些都是汇编语言的功能)。因此,C语言既可用于编写应用软件,又可用于编写系统软件。学习和掌握C语言,既可以增进对计算机底层工作机制的了解,又为进一步学习其他高级语言打下坚实的基础。因此,C语言是人们学习程序设计的首选语言,是计算机及相关专业的一门专业基础课程和必修课程。

C语言程序设计课程的教学目标是培养学生的逻辑思维能力和程序设计能力,使学生掌握C语言的语法规则及程序设计方法,培养学生用C语言编写程序的能力。本书汇集编者多年讲授C语言程序设计课程的教学经验,全书体系完整,内容由浅入深,概念清晰,重点突出,语言流畅;实例丰富,紧扣知识点,并以Visual C++ 6.0为程序平台,可操作性强;注重编程方法与技巧的讲解,重视对学生编程能力的培养。

全书共分12章,内容安排如下:

第1章主要介绍C程序的基本结构、C程序的开发过程及C程序开发平台Visual C++ 6.0的使用。

第2章主要介绍C语言的数据类型、常量和变量、运算符和表达式。

第3章主要介绍算法、结构化程序设计方法、格式化输入/输出函数。

第4章主要介绍选择结构控制语句。

第5章主要介绍循环控制语句、循环的嵌套。

第6章主要介绍一维数组、二维数组和字符数组及其应用。

第7章主要介绍函数的定义、调用、参数传递、递归调用。

第8章主要介绍指针的概念、指针变量的定义、指针与数组、指针与函数的关系。

第9章主要介绍C语言常用的三种预处理指令:宏定义、文件包含、条件编译。

第10章主要介绍结构体、共用体、枚举类型的定义及应用。

第11章主要介绍文件的基本操作。

第12章主要介绍位运算符及其运算规则。

本书各章都列有该章的教学目标、该章小结,除第1章外,每章的最后一节都精选了涉及该章内容的综合案例,使学生通过阅读和实践案例,掌握知识点的应用,提高程序设计的能力,既方便学生的学习,又方便教师的教学。

本书适合作为高等学校计算机及相关专业C语言程序设计课程的教材,也可作为计算

机等级考试的参考书,还可供从事计算机软件开发人员参考使用。本书同期配套出版《C 语言程序设计习题与实训教程》,既便于教师教学,又便于学生练习。

本书由齐景嘉、蒋巍任主编,郭海霞、蔡庆平、郭海龙任副主编,参加本书编写工作的还有李凌霞、于淑华。各章编写分工如下:第 1、12 章由郭海霞编写;第 2、6 章及附录由李凌霞编写;第 3、4 章由蒋巍编写;第 5 章由齐景嘉编写;第 7 章由蔡庆平编写;第 8、9 章由于淑华编写、第 10、11 章由郭海龙编写。全书由齐景嘉统稿,郭川军审核了全书。

本书编者都是多年从事 C 语言程序设计课程教学的教师,但由于编者水平有限,书中不妥与疏漏之处在所难免,敬请广大读者指正。

本书配套课件等资源可以从清华大学出版社网站 www.tup.com.cn 下载,如果在下载或使用中遇到问题,请联系 fuhy@tup.tsinghua.edu.cn。

编 者

2013 年 12 月

目 录

第 1 章 C 语言概述	1
1.1 程序与程序设计语言	1
1.1.1 程序	1
1.1.2 程序设计语言	1
1.2 C 语言的发展和特点	2
1.2.1 C 语言的发展史	2
1.2.2 C 语言的特点	3
1.2.3 C 和 C++	4
1.3 C 语言程序的基本结构	5
1.3.1 简单的 C 程序	5
1.3.2 C 语言程序的基本结构	7
1.3.3 C 语言程序的书写规则	7
1.4 程序的调试	7
1.4.1 调试步骤	7
1.4.2 Visual C++ 6.0 调试环境的使用	7
本章小结	12
第 2 章 C 语言的基本知识	13
2.1 标识符、常量和变量	13
2.1.1 标识符	13
2.1.2 常量和变量	14
2.2 C 语言的数据类型	16
2.2.1 为什么要讨论数据类型	16
2.2.2 C 语言有哪些数据类型	16
2.2.3 基本数据类型	17
2.3 整型数据	17
2.3.1 整型常量	17
2.3.2 整型变量	18
2.3.3 整型数据的内存表示	20
2.4 实型数据	21
2.4.1 实型常量	21
2.4.2 实型变量	22

2.4.3	实型数据的内存表示	23
2.5	字符型数据	23
2.5.1	字符常量	23
2.5.2	字符串常量	25
2.5.3	字符变量	25
2.5.4	字符的内存表示	27
2.6	数据类型转换	27
2.6.1	自动类型转换	27
2.6.2	强制类型转换	28
2.7	运算符和表达式	29
2.7.1	算术运算符及其表达式	30
2.7.2	自增、自减运算符	31
2.7.3	赋值运算符与赋值表达式	32
2.7.4	关系运算符和逻辑运算符	34
2.7.5	条件运算符及其表达式	37
2.7.6	逗号运算符及其表达式	38
2.7.7	运算符的优先级与结合性	39
	本章小结	40
第3章	顺序结构程序设计	41
3.1	算法	41
3.1.1	算法的概念	41
3.1.2	算法的描述	42
3.2	结构化程序设计	43
3.2.1	结构化程序设计思想的产生	43
3.2.2	结构化程序设计的三种基本结构	45
3.3	C语言基本语句	46
3.4	C语言中数据的输入和输出	47
3.4.1	数据的格式化输入输出	48
3.4.2	字符数据的非格式化输入输出	57
3.5	顺序结构程序设计举例	59
	本章小结	61
第4章	选择结构程序设计	62
4.1	if 语句	62
4.1.1	单分支 if 语句	62
4.1.2	双分支 if 语句	63
4.1.3	多分支 if 语句	63
4.1.4	if 语句的嵌套	65

4.2	switch 语句	66
4.3	选择结构程序设计举例	68
	本章小结	71
第 5 章	循环结构程序设计	72
5.1	问题提出与程序示例	72
5.2	while 语句	73
5.2.1	while 语句的一般格式	73
5.2.2	while 语句的执行过程	73
5.3	do...while 语句	75
5.3.1	do...while 语句的一般格式	75
5.3.2	do...while 语句的执行过程	76
5.4	for 语句	77
5.4.1	for 语句的一般格式	77
5.4.2	for 语句的执行过程	77
5.5	break 语句和 continue 语句	80
5.5.1	break 语句	80
5.5.2	continue 语句	81
5.6	循环结构嵌套	82
5.7	综合应用举例	84
	本章小结	89
第 6 章	数组	90
6.1	问题提出	90
6.2	一维数组	91
6.2.1	一维数组的定义与初始化	91
6.2.2	一维数组的引用	93
6.2.3	一维数组应用举例	94
6.3	二维数组	96
6.3.1	二维数组的定义与初始化	96
6.3.2	二维数组的引用	98
6.3.3	二维数组应用举例	99
6.4	字符串与字符数组	101
6.4.1	字符数组的定义与初始化	101
6.4.2	字符数组的输入输出	103
6.4.3	字符串处理函数	106
6.4.4	字符数组应用举例	110
6.5	综合应用举例	112
	本章小结	115

第 7 章 函数	117
7.1 问题的提出	117
7.2 函数定义	117
7.2.1 函数定义的一般形式	117
7.2.2 函数的返回值	119
7.3 函数调用	120
7.3.1 函数调用的一般形式	120
7.3.2 函数的嵌套调用	121
7.3.3 函数的递归调用	122
7.4 函数说明	124
7.5 参数传递	126
7.5.1 形参和实参	126
7.5.2 参数传递的形式	127
7.6 数组作为函数参数	130
7.6.1 数组元素作为函数参数	130
7.6.2 数组名作为函数参数	131
7.7 局部变量和全局变量	133
7.7.1 局部变量	133
7.7.2 全局变量	134
7.8 变量的存储属性	137
7.9 内部函数和外部函数	141
7.9.1 内部函数	141
7.9.2 外部函数	142
7.10 综合应用举例	143
本章小结	145
第 8 章 指针	147
8.1 指针和指针变量	147
8.1.1 指针的基本概念	147
8.1.2 指针数据类型	148
8.1.3 指针运算符 & 和 * 的使用	149
8.1.4 指针变量作为函数参数	151
8.2 空间的动态分配与指针运算	152
8.2.1 问题提出与程序示例	152
8.2.2 存储器申请	153
8.2.3 存储器释放	154
8.2.4 指针的算术运算	155
8.3 指针与一维数组	157

8.3.1	问题提出与程序示例	157
8.3.2	指针用于数组处理	158
8.3.3	指针与字符串	159
8.3.4	指向数组的指针作为函数参数	162
8.4	指针与二维数组	164
8.4.1	指针与二维数组的关系	164
8.4.2	向函数传递二维数组	168
8.5	指针数组和指向指针的指针	169
8.5.1	指针数组的定义与使用	169
8.5.2	指针数组与字符串数组	171
8.5.3	指向指针的指针	172
8.5.4	main()函数的形参	174
8.6	指针与函数	176
8.6.1	返回指针值的函数	176
8.6.2	指向函数的指针	177
8.7	综合应用举例	180
	本章小结	184
第 9 章	编译预处理	186
9.1	宏定义	186
9.1.1	无参宏定义	186
9.1.2	带参宏定义	188
9.2	文件包含	190
9.3	条件编译	191
9.4	案例分析	192
	本章小结	195
第 10 章	结构体、共用体和枚举类型	196
10.1	结构体	196
10.1.1	问题提出与程序示例	196
10.1.2	结构体类型的定义	197
10.1.3	结构体变量的初始化	199
10.1.4	结构体成员的引用	200
10.1.5	程序举例	201
10.2	结构体数组	202
10.2.1	结构体数组的定义与初始化	202
10.2.2	程序举例	203
10.3	指向结构体的指针	205
10.3.1	指向结构体变量的指针	205

10.3.2	指向结构体数组的指针	208
10.4	用 typedef 定义类型	209
10.5	结构体的应用——链表	210
10.5.1	链表的概念	210
10.5.2	链表结点的定义	210
10.5.3	链表的建立	211
10.5.4	链表的基本操作	211
10.6	共用体类型	213
10.6.1	共用体的定义	213
10.6.2	程序举例	214
10.7	枚举类型	216
10.8	综合应用举例	218
	本章小结	218
第 11 章	文件	219
11.1	问题提出与程序示例	219
11.2	文件操作的基本方法和相关概念	221
11.2.1	数据文件	221
11.2.2	文件类型指针	222
11.2.3	文件的打开(fopen 函数)	223
11.2.4	文件的关闭(fclose 函数)	224
11.3	文件的读写操作	225
11.3.1	fputc()函数与 fgetc()函数	225
11.3.2	fprintf()函数与 fscanf()函数	226
11.3.3	fread()函数与 fwrite()函数	228
11.3.4	fgets()函数与 fputs()函数	229
11.4	文件的定位	230
11.4.1	文件的顺序存取和随机存取	230
11.4.2	rewind()函数	230
11.4.3	fseek()函数	231
11.5	综合应用举例	232
	本章小结	233
第 12 章	位运算	234
12.1	位运算符和位运算	234
12.1.1	按位与运算	235
12.1.2	按位或运算	236
12.1.3	按位异或运算	236
12.1.4	取反运算	237

12.1.5 左移运算	237
12.1.6 右移运算	238
12.2 位运算符的运算功能	238
12.3 应用举例	238
本章小结	239
附录 A 常用字符与 ASCII 码对照表	240
附录 B C 语言运算符的优先级和结合性	242
附录 C Visual C++ 6.0 环境下 C 程序的基本开发过程	243
附录 D C 语言常见出错信息	250
参考文献	254

第1章

C语言概述

教学目标

- ◇ 了解程序及程序设计语言相关的基本概念。
- ◇ 掌握 C 语言的基本结构。
- ◇ 掌握利用 Visual C++ 6.0 集成开发环境调试 C 语言程序的基本方法。

1.1 程序与程序设计语言

1.1.1 程序

程序(program)是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合,也可理解成为实现预期目的所需执行的一系列语句和指令。程序一般可分为系统程序和应用程序两大类。

一个程序应该包括以下两方面的内容。

① 对数据的描述:在程序中要指定数据的类型和数据的组织形式,即数据结构(data structure)。

② 对操作的描述:即操作步骤,也就是算法(algorithm)。

著名计算机科学家尼古拉斯·沃思(Niklaus Wirth)提出一个公式:数据结构+算法=程序。实际上,一个程序除了以上两个主要的要素外,还应当采用合适的程序设计方法进行设计,并且采用一种计算机语言来表示。因此,编写程序时算法、数据结构、程序设计方法和语言工具四个方面缺一不可。

1.1.2 程序设计语言

计算机程序设计语言通常简称为编程语言,是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧,用来向计算机发出指令。计算机语言让程序员能够准确地定义计算机需要使用的数据,并精确地定义在不同情况下计算机所应当采取的行动。在过去的几十年间,大量的程序设计语言被发明、被取代、被修改或组合在一起。尽管人们多次试图创造一种通用的程序设计语言,却没有一次尝试是成功的。之所以有那么多不同的编程语言存在,原因是多方面的:不同的编程者编写程序的初衷各不相同;不同层次的编程者之间会存在一定的技术差距,例如许多编程语言对初学者来说太难;不同程序之间的运

行成本(runtime cost)各不相同。此外,许多编程语言有其特殊的用途,例如,PHP 专门用来显示网页,而 C 语言被广泛用于操作系统和编译器的开发(所谓的系统编程)。因此,想要创造一种通用的程序设计语言绝非易事。

程序设计语言按照语言级别可以分为低级语言和高级语言。低级语言又有机器语言和汇编语言之分。低级语言与特定的机器环境有关,而且操作复杂、烦琐、费时、易出差错。机器语言是计算机能够直接识别的语言,是表示成二进制代码形式的机器基本指令集,也可以说是操作码经过符号化的基本指令集。汇编语言的实质和机器语言是相同的,只不过汇编语言是机器语言中地址部分符号化的结果。高级语言的表示方法要比低级语言更接近人的思维习惯,更易于描述需要解决的问题。高级语言的特点是在一定程度上与具体机器无关,易学、易用、易维护。高级语言的出现使得计算机程序设计语言不再过多地倚赖某种特定的机器或环境。这是因为高级语言在不同的平台上会被编译成不同的机器语言,而计算机可以直接执行机器语言,却不能直接执行高级语言。因此,可以把高级语言理解为是独立于平台的。FORTRAN 语言是最早出现的高级编程语言之一,当初开发它的一个主要目标就是要实现程序设计语言的平台独立性。同时,高级语言并不是特指某一种具体的语言,常用的高级语言有很多种,例如 FORTRAN、COBOL、BASIC、PASCAL、C、FOXPRO、DELPHI 等,使用这些语言编程时,它们的语法格式、命令格式都是各不相同的。

1.2 C 语言的发展和特点

1.2.1 C 语言的发展史

C 语言是一种高级语言,它的发展与 UNIX 操作系统有着紧密的联系。

- ◇ C 语言的原型是 ALGOL 60 语言(也称为 A 语言)。
- ◇ 1963 年,英国剑桥大学将 ALGOL 60 语言发展成为 CPL(Combined Programming Language)语言。
- ◇ 1967 年,剑桥大学的 Martin Richards 对 CPL 语言进行了简化,于是产生了 BCPL 语言。
- ◇ 1970 年,美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改,并为它起了一个有趣的名字“B 语言”。意思是将 CPL 语言煮干,提炼出它的精华。并且他用 B 语言写了第一个 UNIX 操作系统。
- ◇ 1973 年,美国贝尔实验室的 Dennis M. Ritchie 在 B 语言的基础上最终设计出了一种新的语言,他取了 BCPL 的第二个字母作为这种语言的名字,这就是 C 语言。C 语言既保持了 BCPL 语言和 B 语言的精练、接近硬件的优点,同时又克服了它们过于简单的缺点。为了推广 UNIX 操作系统,Ken Thompson 和 Dennis M. Ritchie 合作把 UNIX 系统的 90% 以上用 C 语言进行了改写,并加入了多道程序设计的功能,从此开创了 UNIX 操作系统发展的新纪元。
- ◇ 1976 年,UNIX 第六版颁布后,C 语言得到了计算机界的普遍认可。
- ◇ 1977 年,Dennis M. Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。

- ◇ 1978 年, Brian W. Kernighian 和 Dennis M. Ritchie 出版了名著《The C Programming Language》, 从而使 C 语言成为当时世界上流行最广泛的高级程序设计语言, 从此, C 语言与 UNIX 操作系统一起互相促进并迅速发展。
- ◇ 1988 年, 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, 使得这些版本的 C 语言之间出现了一些不一致的地方。为了改变这种情况, 美国国家标准研究所(ANSI)为 C 语言制定了一套 ANSI 标准, 成为现行的 C 语言标准的主要依据。

1.2.2 C 语言的特点

C 语言是一种计算机程序设计语言。它既具有高级语言的特点, 又具有汇编语言的特点。它可以作为系统程序设计语言, 编写系统应用程序, 也可以作为应用程序设计语言, 编写不依赖计算机硬件的应用程序。它的应用范围广泛, 具备很强的数据处理能力, 不仅仅用于软件开发, 而且各类科研都需要用到。C 语言适于编写系统软件、三维、二维图形和动画等。概括起来, C 语言的主要特点如下。

- ◇ 语言简洁紧凑, 运用灵活、方便。C 语言一共只有 32 个关键字和 9 种控制语句, 程序书写形式自由, 区分大小写。它是把高级语言的基本结构和语句与低级语言的实用性结合起来。虽然 C 语言也是强类型语言, 但它的语法比较灵活, 允许程序编写者有较大的自由度。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, 因此它既具有高级语言的功能, 又具有低级语言的许多功能。
- ◇ 具有丰富的数据类型。C 语言不但具有字符型、整型、浮点型、数组类型、指针类型等基本数据类型, 而且具有结构体、共用体、枚举等数据类型。使用这些数据类型能够实现各种复杂的数据结构的运算。C 语言的计算功能、逻辑判断功能强大, 而且为用户提供了自定义数据类型的功能。
- ◇ C 语言具有强大的图形处理功能, 支持多种显示器和驱动器, 具有强大的数据处理能力, 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, 如二进制的位运算、复合运算等。C 语言可以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。
- ◇ C 语言具有适用范围广、可移植性好的特点。C 语言所提供的语句中没有直接与硬件相关的语句和与硬件直接相关的操作, 输入/输出功能不依赖于特定的计算机硬件, 而是通过调用系统的库函数的方式来实现。用 C 语言编写的程序可以很容易地从一种计算机环境移植到另一种计算机环境中。同时, C 语言也适合多种操作系统, 如 DOS、UNIX、Windows XP、Windows 7 等。
- ◇ 具有强大的预处理能力。在编译源程序之前, 系统首先执行预处理命令, 这些预处理命令的使用大大提高了软件开发的工作效率, 并为程序的组织和编译提供了便利。
- ◇ 控制流程结构化。C 语言提供了一整套循环、条件判断语句和转移语句, 实现了对程序逻辑流程的有效控制, 有利于结构化程序设计。结构式语言的显著特点是代码及数据的分隔化, 即程序的各个部分之间除了必要的信息交流外彼此独立。这种结构