

C 语言程序设计

(第2版)

◎主编 梅创社 李培金

CYUYAN
CHENGXU SHEJI
(2nd Edition)



014062121

TP312C-43
466-2

C 语言程序设计

(第 2 版)

主 编 梅创社 李培金
副主编 李 俊 刘引涛 殷锋社



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



北航

C1749172

TP 312C-43
466-2

151280910

内 容 简 介

本书围绕程序设计思想训练这个主题,充分吸纳项目教学法的思想,每一章节的内容都利用小型的项目案例进行引导,训练学生的逻辑思维和分析问题以及程序文本的读写能力。由于C语言的语法特点,本书依然沿用知识体系的编写方法,但教材内容的组织按照项目教学法的思想组织教学内容,全书共分为三个部分:语言基础知识、语言设计知识和项目案例库。每章内容都指明了学习内容、知识教学目标和技能培养目标,本书强调实际编程能力的培养,章节内包含了大量的程序设计案例供学习者参考,并在每章后附有大量的习题与实训供读者练习。

本书可作为计算机专业、电子类专业的学习教材,也可供爱好C语言程序设计的读者自学使用。

版权专有 侵权必究

图书在版编目(CIP)数据

C语言程序设计 / 梅创社, 李培金主编. —2版. —北京: 北京理工大学出版社, 2014. 6
ISBN 978-7-5640-9295-5

I. ①C… II. ①梅… ②李… 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115635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)
82562903 (教材售后服务热线)
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 / 三河市天利华印刷装订有限公司

开 本 / 787毫米×1092毫米 1/16

印 张 / 19

字 数 / 437千字

版 次 / 2014年6月第2版 2014年6月第1次印刷

定 价 / 44.00元

责任编辑 / 张慧峰

文案编辑 / 张慧峰

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

Preface

C 语言功能丰富、使用灵活、可移植性好，既具有高级语言的优点，又具有低级语言的许多特点。C 语言程序还可直接对计算机的硬件进行控制，既适合开发系统软件，又适合开发应用软件，因此，C 语言成为计算机专业和电子信息类专业计算机程序设计的首选语言，是国内外广泛使用的计算机编程语言，深受广大程序设计者的喜爱。

本书在对前期版本存在问题进行修正完善的基础上，结合编者多年的 C 语言教学实践，充分考虑教学的特点，突出对学生编程能力的培养，以程序设计为主线，在深入浅出讲解 C 语言语法规则和程序设计基本结构的同时，通过精心筛选的例题，着重介绍 C 程序设计的基本方法和算法的基本要领，通过对项目案例的分析和训练，可使学生在掌握 C 语言程序设计基本结构的同时，增强对程序的读写和思维能力。

教材编写继续沿用知识体系的编写方法，充分将项目案例教学法内容渗透到各章节中。教材编写体系分为以下三部分。

第一部分：语言基础知识。主要介绍 C 语言基础知识以及与 C 语言在编程中密切相关的其他接口软件。

第二部分：语言设计知识。主要介绍 C 程序设计的三种基本结构、函数、数据构造类型和指针的基本知识与应用。

第三部分：项目案例库。提供若干个小型的项目案例让学生熟悉程序设计的流程和步骤，供教师授课引用和学生学习参考。

本教材有以下两个方面的特点。

(1) 每章节后的项目案例，给出了程序的实际分析及参考的程序代码，紧紧围绕训练学生程序设计能力这个主题。

(2) 教材在第三部分增加了项目案例库，可供教师授课引用和学生课外自主学习，培养学生熟悉程序设计开发流程，增强实践性。

本书在编写过程中参考了国内外出版的大量 C 语言程序设计教材，在此对参考文献的作者一并表示感谢！

由于作者水平有限，书中错误在所难免，敬请专家和广大读者批评指正。

编 者

目录

Contents

第一部分 语言基础知识

第1章 C语言概述	3
1.1 C语言的发展和特点	3
1.1.1 C语言的发展	3
1.1.2 C语言的特点	4
1.1.3 C语言的应用领域	4
1.2 C语言程序的结构与书写规则	5
1.2.1 C语言程序的基本结构	5
1.2.2 C语言函数的一般结构	7
1.2.3 C语言函数的语法规则	7
1.3 C语言的编译环境	9
1.4 本章小结	17
习题与实训	18
第2章 数据类型、运算符与表达式	21
2.1 C语言的数据类型	21
2.2 标识符、常量和变量	22
2.2.1 标识符	22
2.2.2 常量	23
2.2.3 变量	23
2.3 整型数据	24
2.3.1 整型常量	24
2.3.2 整型变量	25
2.4 实型数据	26
2.4.1 实型常量	26
2.4.2 实型变量	26

2.5	字符型数据	27
2.5.1	字符常量	27
2.5.2	字符变量	28
2.5.3	字符串常量	29
2.6	算术运算与算术表达式	30
2.6.1	算术运算	30
2.6.2	算术表达式	31
2.7	赋值运算与赋值表达式	32
2.8	自增、自减与逗号运算	33
2.8.1	自增(++)自减(--)运算	33
2.8.2	逗号运算及其表达式	34
2.9	本章小结	34
	习题与实训	36

第二部分 语言设计知识

第3章	顺序结构程序设计	41
3.1	程序设计概述	41
3.1.1	算法的概念	41
3.1.2	算法的特征	42
3.1.3	算法的描述	42
3.1.4	结构化程序设计方法	44
3.2	C语言语句	45
3.3	格式化输出函数 printf()	46
3.3.1	printf()函数的一般格式	46
3.3.2	格式字符	47
3.3.3	使用说明	49
3.4	格式化输入函数 scanf()	49
3.4.1	scanf()函数的一般格式	50
3.4.2	格式字符	50
3.4.3	使用说明	51
3.5	单个字符的输入/输出函数	52
3.5.1	putchar()函数	52
3.5.2	getchar()函数	52
3.6	顺序结构程序设计项目案例	54
3.7	本章小结	56
	习题与实训	56

第4章 选择结构程序设计	61
4.1 关系运算符及关系表达式	61
4.1.1 关系运算符及其优先级	61
4.1.2 关系表达式	62
4.2 逻辑运算符和逻辑表达式	63
4.2.1 逻辑运算符及其优先级	63
4.2.2 逻辑表达式	64
4.3 if 语句和条件运算符	65
4.3.1 if 语句	65
4.3.2 条件运算符	68
4.4 switch 语句	69
4.5 选择结构程序设计项目案例	71
4.6 本章小结	72
习题与实训	73
第5章 循环结构程序设计	77
5.1 循环语句概述	77
5.1.1 引例	77
5.1.2 循环结构和循环语句	78
5.2 for 语句和 while 语句	79
5.2.1 for 语句	79
5.2.2 while 语句	81
5.3 直到型循环 do-while 语句	83
5.3.1 引例	83
5.3.2 do-while 语句	84
5.4 循环结构的嵌套	85
5.5 break 语句与 continue 语句	86
5.5.1 break 语句	86
5.5.2 continue 语句	87
5.6 循环结构程序设计项目案例	89
5.7 本章小结	91
习题与实训	92
第6章 函数	98
6.1 函数的定义与调用	98
6.1.1 函数的定义	99
6.1.2 函数的返回值与函数类型	101
6.1.3 对被调用函数的声明和函数原型	103

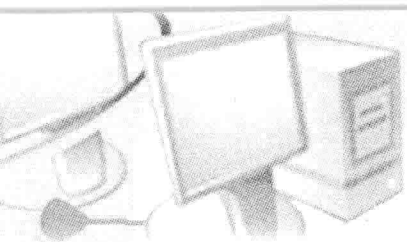
6.1.4	函数的调用	105
6.1.5	函数的形参与实参	107
6.2	函数的嵌套调用和递归调用	108
6.2.1	函数的嵌套调用	108
6.2.2	函数的递归调用	111
6.3	内部变量与外部变量	113
6.3.1	内部变量	113
6.3.2	外部变量	114
6.4	内部函数和外部函数	116
6.4.1	内部函数	117
6.4.2	外部函数	119
6.4.3	多个源程序文件的编译和连接	119
6.5	变量的动态存储与静态存储简介	120
6.5.1	动态存储与静态存储的概念	120
6.5.2	内部变量的存储方式	120
6.5.3	外部变量的存储方式	123
6.6	本章小结	124
	习题与实训	126
第7章	数组	132
7.1	一维数组	132
7.1.1	一维数组的定义	132
7.1.2	一维数组元素的引用	133
7.1.3	一维数组的初始化	134
7.1.4	一维数组应用举例	135
7.2	二维数组	137
7.2.1	二维数组的定义	137
7.2.2	二维数组元素的引用	138
7.2.3	二维数组的初始化	138
7.2.4	二维数组应用举例	139
7.3	字符数组与字符串	140
7.3.1	字符数组的定义和基本操作	140
7.3.2	字符数组的整体操作	141
7.3.3	常用字符串处理函数	144
7.4	数组作为函数参数	145
7.4.1	数组元素作为函数参数	145
7.4.2	数组名作为函数参数	146
7.5	本章小结	149
	习题与实训	150

第 8 章 编译预处理	155
8.1 宏定义	155
8.1.1 无参宏定义	155
8.1.2 有参宏定义	158
8.2 文件包含	160
8.3 本章小结	161
习题与实训	161
第 9 章 指针	163
9.1 指针和指针变量的概念	163
9.2 指针变量的定义与应用	164
9.2.1 指针变量的定义与相关运算	164
9.2.2 指针变量作函数参数	168
9.3 数组的指针和指向数组的指针变量	170
9.3.1 概述	170
9.3.2 通过指针引用数组元素	170
9.3.3 数组名作函数参数	174
9.3.4 二维数组的指针及其指针变量	176
9.4 字符串的指针和指向字符串的指针变量	179
9.4.1 字符串的表示和引用	179
9.4.2 字符串指针作函数参数	181
9.5 返回指针值的函数	183
9.6 指针数组与主函数 main() 的形参	184
9.6.1 指针数组	184
9.6.2 主函数 main() 的形参	185
9.6.3 指向指针的指针	186
9.7 函数的指针和指向函数的指针变量	187
9.8 本章小结	188
习题与实训	189
第 10 章 结构型、共用型和枚举类型	193
10.1 结构类型与结构型变量的定义	193
10.1.1 结构类型定义	193
10.1.2 结构型变量的定义	195
10.2 结构型变量的引用与初始化	196
10.2.1 结构型变量的引用与赋值	196
10.2.2 结构型变量的初始化	197
10.3 结构数组	197

10.4	指向结构类型数据的指针	199
10.4.1	指向结构型变量的指针	199
10.4.2	指向结构型数组的指针	201
10.4.3	指向结构数据的指针作函数参数	202
10.5	链表处理——结构指针的应用	203
10.5.1	链表的创建	203
10.5.2	对链表的插入操作	205
10.6	共用型和枚举型	206
10.6.1	共用型	206
10.6.2	枚举型	208
10.7	用户自定义类型	209
10.8	本章小结	210
	习题与实训	211
第 11 章	位运算	213
11.1	数值在计算机中的表示	213
11.2	位运算	214
11.2.1	位运算符	214
11.2.2	位运算符使用方法	215
11.2.3	应用举例	218
11.3	本章小结	219
	习题与实训	219
第 12 章	文件	222
12.1	文件概述	222
12.2	文件的打开与关闭	223
12.2.1	文件的打开——fopen()函数	224
12.2.2	文件的关闭——fclose()函数	225
12.3	文件的读写操作	225
12.3.1	读/写文件中的一个字符——fgetc()和 fputc()函数	226
12.3.2	读/写一个字符串——fgets()函数和 fputs()函数	229
12.3.3	读/写一个数据块——fread()和 fwrite()函数	231
12.3.4	对文件进行格式化读/写——fscanf()和 fprintf()函数	231
12.3.5	读/写函数的选用原则	232
12.4	位置指针与文件定位	232
12.5	本章小结	233
	习题与实训	234

第三部分 项目案例库

第 13 章 C 语言程序项目案例	239
13.1 火车订票系统	239
13.1.1 系统功能总体描述	239
13.1.2 详细设计	239
13.1.3 调试分析	240
13.2 掷骰子游戏	253
13.3 24 点扑克牌游戏	255
13.4 强大的通讯录	264
13.5 竞技比赛打分系统	270
附录	279
附录 A 标准 ASCII 字符编码表	279
附录 B 运算符的优先级和结合方向	280
附录 C 标准库函数	281



第一部分

语言基础知识

第 1 章 C 语言概述

C 语言是一种面向过程的并且很灵活的程序设计语言。在计算机日益普及的今天，C 语言的应用领域依然很广泛，几乎各类计算机都支持 C 语言的开发环境。掌握了 C 语言后，再学 C++、VC++、Java 等其他语言就比较容易了。

【本章内容】

- C 语言的发展和特点
- C 语言程序结构和书写规则
- C 语言的编译环境

【知识教学目标】

- C 语言的发展与特点
- C 语言程序的基本结构
- C 语言程序的基本符号与规则
- C 语言程序的集成开发环境

【技能培养目标】

- 掌握 C 语言程序的基本结构
- 学会使用 C 语言的集成开发环境
- 熟悉 C 语言程序的调试运行过程

1.1 C 语言的发展和特点

1.1.1 C 语言的发展

在 C 语言诞生以前，系统软件（例如操作系统）主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件，其可读性和可移植性都很差；但一般的高级语言又难以实现对计算机硬件的直接操作（这正是汇编语言的优势），于是人们盼望有一种兼有汇编语言和高级语言特性的新语言。C 语言就是在这种背景下应运而生的。

C 语言是贝尔实验室于 20 世纪 70 年代初期研制出来的，并随着 UNIX 操作系统的日益广泛使用，迅速得到推广。后来，C 语言又被多次改进，并出现了多种版本。20 世纪 80 年代初（1983 年），美国国家标准化协会（ANSI），根据 C 语言问世以来各种版本对 C 语言的发展和扩充，制定了 ANSI C 标准（1989 年再次做了修订）。本书以 ANSI C 标准来介绍。

目前，在微机上广泛使用的 C 语言编译系统有 Microsoft C（简称 MSC）、Turbo C（简称



TC)、Borland C (简称 BC) 等。虽然它们的基本部分都是相同的,但还是有一些差异,所以请读者注意自己所使用的 C 编译系统的特点和规定 (可参阅相应的手册了解)。本书选定的上机环境是 Visual C++ 6.0。

下面是一个计算圆面积的 C 语言程序段 (其中: /*...*/ 称为注释行, “/*” 与 “*/” 之间的内容称为注释内容):

```
void main()                /*告诉编译器 C 程序由此开始执行*/
{                            /*程序片段执行开始*/
    float r,s;              /*定义圆半径 r 与面积 s 为实型数据*/
    r=5.356;                /*给半径 r 赋值*/
    s=3.14159*r*r;          /*计算面积 s*/
    printf("%f\n",s);       /*输出面积 s 的值*/
}                            /*程序执行结束*/
```

由上述 C 语言程序可以看出: C 语言是一种面向过程的程序设计语言。

1.1.2 C 语言的特点

C 语言是近年来较流行的高级程序设计语言之一,许多大型软件均是用 C 语言编写的 (如 UNIX 操作系统)。C 语言同时具有汇编语言和高级语言的双重特性。具体来说, C 语言的主要特点如下:

(1) C 语言是一种模块化的程序设计语言。模块化的基本思想是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、容易理解的小程序。

(2) C 语言简洁,结构紧凑,使用方便、灵活。C 语言一共只有 32 个关键字、9 条控制语句,源程序书写格式自由。

(3) 运算极其丰富,数据处理能力强。C 语言一共有 34 种运算符,例如,算术运算符、关系运算符、自增 (++) 和自减 (--) 运算符、复合赋值运算符、位运算符及条件运算符等。同时, C 语言又可以实现其他高级语言较难实现的功能。

(4) 可移植性好。C 语言程序基本上可以不作任何修改,就能运行于各种不同型号的计算机和各种操作系统环境上。

(5) 可以直接调用系统功能,实现对硬件的操作。这是其他高级语言所不具备的。

当然, C 语言本身也有其弱点,例如: C 语言的语法限制不太严格,在增加程序设计灵活性的同时,在一定程度上降低了某些安全性,这就对程序设计人员提出了更高的要求。

1.1.3 C 语言的应用领域

C 语言是一种通用的、面向过程的编程语言,广泛应用于系统软件与应用软件的开发。因为 C 语言既具有高级语言高效、灵活和可移植性等特点,又同时具有汇编语言可以对计算

机硬件进行管理的特点,因此C语言有广泛的应用领域。

下面列举一些C语言常见的应用领域:

(1) 系统软件。许多著名的系统软件,如DBASE III PLUS、DBASE IV都是由C语言编写的。

(2) 应用软件。Linux操作系统中的应用软件都是使用C语言编写的,这样的应用软件安全性非常高。

(3) 科学计算。相对于其他编程语言,C语言是数字计算能力超强的高级语言。

(4) 图形处理。C语言具有很强的绘图能力和数据处理能力,可以用来制作动画、绘制二维图形和三维图形等。

(5) 嵌入式应用开发。手机、PDA、电子字典等时尚消费类电子产品内部的应用软件、游戏等很多都是采用C语言进行嵌入式开发的。

要学好任何一门计算机语言都不是一件很容易的事,学习C语言也不例外。但掌握了C语言后,再学其他语言就比较容易了,所以对有志于从事计算机编程的人而言,C语言是一门要认真加以钻研的语言。

1.2 C语言程序的结构与书写规则

1.2.1 C语言程序的基本结构

下面通过几个简单的示例来介绍C语言程序的基本构成和书写格式,使读者对C语言程序有一个基本的了解。在此基础上,再进一步了解C语言程序的语法和书写规则。

【例 1.1】求三个数平均值的C语言程序。

```
/*功能: 求三个数的平均值*/
void main()                                /* main() 称为主函数*/
{float a,b,c,ave;                          /* 定义 a, b, c, ave 为实型数据*/
a=7;
b=9;
c=12;
ave=(a+b+c)/3;                            /*计算平均值*/
printf("ave=%f\n",ave);                  /* 在屏幕上输出 ave 的值 */
}
```

程序运行结果:

```
ave=9.333333
```

【例 1.2】输出两个数中较大值的C语言程序。

```
/*功能: 输出两个数中的较大值 */
void main()                                /*主函数*/
{int num1,num2;                          /* 定义 num1、num2 为整型变量 */
```



```

scanf("%d,%d",&num1,&num2);          /* 由键盘输入 num1、num2 的值 */
printf("max=%d\n",max(num1,num2));    /* 在屏幕上输出调用 max 的函数值 */
}

/*以下是用户自己设计的函数 max()*/
int max(int x,int y)                  /* x 和 y 分别取 num1 和 num2 传递的值*/
{if(x>y) return x;                    /* 如果 x>y,将 x 的值返回给 max */
 else return y;                       /* 如果 x>y 不成立,将 y 的值返回给 max */
}

```

程序运行情况:

5,8↵ ("↵"表示按回车键,以下相同)

max=8

在以上两个示例中,例 1.1 所示的 C 语言程序,仅由一个 `main()` 函数构成,它相当于其他高级语言中的主程序;例 1.2 所示的 C 语言程序由一个 `main()` 函数和一个其他函数 `max()`(用户自己设计的函数)构成,函数 `max()` 相当于其他高级语言中的子程序。由此可见,一个完整的 C 语言程序结构有以下两种表现形式。

(1) 仅由一个 `main()` 函数(又称主函数)构成。结构如图 1-1(a) 所示。

(2) 由一个且只能有一个 `main()` 函数和若干个其他自定义函数结合而成。结构如图 1-1(b) 所示。其中:自定义函数由用户自己设计。

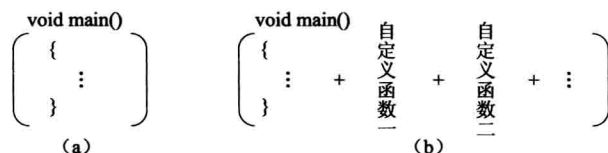


图 1-1 C 语言程序结构示意图

结合以上示例,可以看出 C 语言程序结构有以下基本特点。

(1) C 语言程序是由函数(如: `main()` 函数和 `max()` 函数)组成的,每一个函数完成相对独立的功能,函数是 C 语言程序的基本模块单元。`main` 是函数名,函数名前面的类型表示函数返回值类型, `void` 表示函数没有返回值。`main` 函数名后面的一对圆括号“`()`”是写函数的参数的,参数可以有,也可以没有(本程序没有参数),但圆括号“`()`”不能省略。

(2) 一个 C 语言程序总是从 `main()` 函数开始执行。主函数执行完毕,程序执行结束。

(3) C 语言编译系统区分字母大小写, C 语言把大小写字母视为两个不同的字符,并规定每条语句或数据说明均以分号“`;`”结束,而且分号是语句不可缺少的组成部分。

(4) 主函数 `main()` 既可以放在 `max()` 函数之前,也可以放在 `max()` 函数之后。习惯上,将主函数 `main()` 放在最前面。

(5) C 语言程序中所调用的函数,既可以使用由系统提供的库函数,也可以由设计人员自己根据需要而设计。例如,在例 1.2 中, `printf()` 函数是 C 语言编译系统库函数中的一个函数,它的作用是在屏幕上按指定格式输出指定的内容; `max()` 函数是由用户自己设计的函数,它的作用是计算两个数中的较大值。