



工业和信息化普通高等教育“十二五”规划教材立项项目

21世纪高等教育计算机规划教材、21世纪高等学校应用型本科规划教材

C语言程序设计

(第2版)

王洪海 郑利平 主编

薛峰 杜秀全 李凌峰 徐丽萍 邵立 张春光 副主编

- 注重理论联系实际
- 注重可读性与实用性
- 注重读者能力的培养

计算机



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



工业和信息化部普通高等教育“十二五”规划教材立项项目

21世纪高等教育计算机规划教材 21世纪高等学校应用型本科规划教材

C语言程序设计

(第2版)

王洪海 郑利平 ◎ 主编

薛峰 杜秀全 李凌峰 徐丽萍 邵立 张春光 ◎ 副主编

计算机



人民邮电出版社

北京

图书在版编目 (CIP) 数据

C语言程序设计 / 王洪海, 郑利平主编. -- 2版. --
北京: 人民邮电出版社, 2016.2
21世纪高等教育计算机规划教材
ISBN 978-7-115-41000-9

I. ①C… II. ①王… ②郑… 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 (2016) 第015049号

内 容 提 要

本书共分10章, 首先简要介绍程序设计基础, 然后分别系统地介绍C语言的数据类型与运算符、C语言的3种程序结构、数组与函数、指针与文件等相关知识。

本书面向初学者, 内容通俗易懂, 概念清晰, 实践性强。本书提供了大量的实例与习题, 注重各部分知识的综合应用训练。

本书适合作为高等学校本科、高职高专、成人高校的学生和其他初学者学习C程序设计的教材, 也可供参加全国计算机等级考试 (二级C) 的读者选用。

-
- ◆ 主 编 王洪海 郑利平
副 主 编 薛 峰 杜秀全 李凌峰 徐丽萍 邵 立 张春光
责任编辑 邹文波
责任印制 沈 蓉 彭志环
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市潮河印业有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 16.75 2016年2月第2版
字数: 435千字 2016年2月河北第1次印刷
-

定价: 42.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

参编及合作单位

(排名不分先后)

安徽三联学院

合肥工业大学

安徽大学

解放军电子工程学院

前言

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言之一。在对操作系统、系统应用及需要对硬件进行操作的场合中，用 C 语言明显优于用其他高级语言，因此，许多大型应用软件都是用 C 语言编写的。由于 C 语言功能强、使用灵活、可移植性好、目标程序质量好，从而受到编程者广泛的欢迎。

作为长期在教学一线工作的教学工作者，作者与众多优秀老师一起合作，于 2011 年编写并出版了《C 语言程序设计》教材，至今已经走过 4 个年头。《C 语言程序设计》教材自出版后得到了很多高校的专家、教师和学生的支持和关注，他们提出了很多宝贵的建议和意见，我们在此基础上，结合应用型本科高校人才培养需求，重新改写了部分章节，形成了《C 语言程序设计》（第 2 版）。

在新版教材中，我们沿袭了第 1 版中所遵循的知识讲授和能力训练并重的原则，在讲清基本知识的基础上，注意例题的选择，力求理论联系实际和循序渐进，注重培养读者分析问题和程序设计的能力，使读者养成良好的程序设计风格和习惯。另外，在第 2 版中，我们着重更新了部分例题和练习题，目的是逐步引导学生掌握程序设计的方法和技巧，突出应用实践编程能力。

全书共分 10 章，内容包括程序设计基础，C 语言概述，数据类型、运算符及表达式，程序控制结构，数组，函数，指针，结构体与共用体，文件，编译预处理等。与同类书相比，本书注重可读性和实用性，难点分散，努力用人们易于理解的方式清楚地叙述复杂的概念，具有体系合理、逻辑清楚、例题丰富、通俗易懂的特点。本书既可供 C 语言初学者学习使用，也可供参加全国计算机等级考试（二级 C）的读者选用。为了方便读者学习，本书配套教材《C 语言程序设计实验指导（第 2 版）》同期出版，以帮助读者梳理所学知识，掌握应会和必会内容。

本教材由安徽三联学院王洪海、合肥工业大学计算机与信息学院郑利平教授共同担任主编，其他参与编写工作的老师有薛峰、徐丽萍、张春光、李凌峰、杜秀全、邵立。全书由王洪海统稿并定稿。另外，教材的出版也得到了安徽三联学院校领导和人民邮电出版社的大力支持与帮助，在此一并表示衷心的感谢！

由于编者水平有限，书中难免存在错误和不妥之处，敬请广大读者批评指正。

编 者

2015 年 12 月

目 录

第 1 章 程序设计基础1

1.1 程序设计的基本概念1

1.1.1 程序1

1.1.2 算法与数据结构2

1.2 程序设计的一般步骤6

1.3 结构化程序设计方法6

小结7

习题 17

第 2 章 C 语言概述8

2.1 C 语言的发展及其特点8

2.1.1 C 语言的发展简史8

2.1.2 C 语言的特点8

2.2 简单的 C 程序介绍9

2.2.1 简单的 C 程序9

2.2.2 C 源程序的结构特点11

2.2.3 C 源程序的书写规则11

2.3 运行 C 语言程序的基本步骤11

小结12

习题 212

第 3 章 数据类型、运算符 及表达式13

3.1 C 语言的数据类型13

3.2 标识符与关键字14

3.3 常量14

3.3.1 整型常量14

3.3.2 实型常量15

3.3.3 字符型常量16

3.3.4 字符串常量17

3.3.5 符号常量19

3.4 变量20

3.4.1 变量的定义20

3.4.2 变量的初始化22

3.5 运算符及其表达式24

3.5.1 算术运算符和算术表达式25

3.5.2 赋值运算符和赋值表达式28

3.5.3 关系运算符和关系表达式30

3.5.4 逻辑运算符和逻辑表达式31

3.5.5 条件运算符和条件表达式33

3.5.6 逗号运算符和逗号表达式34

3.5.7 位运算35

3.5.8 指针运算符、sizeof 运算符37

3.6 运算符的优先级和结合性38

3.7 表达式中数据类型的转换39

小结43

习题 344

第 4 章 结构化的 C 语言

程序设计47

4.1 结构化程序设计概述47

4.2 顺序结构程序设计47

4.2.1 赋值语句、复合语句47

4.2.2 字符输出函数47

4.2.3 字符输入函数48

4.2.4 格式化输出函数49

4.2.5 格式化输入函数51

4.2.6 顺序结构程序设计举例53

4.3 选择结构程序设计55

4.3.1 简单的 if 结构55

4.3.2 if...else 语句结构56

4.3.3 if 结构的嵌套57

4.3.4 if...else...if 语句结构58

4.3.5 switch 结构59

4.3.6 选择结构程序设计举例60

4.4 循环结构程序设计61

4.4.1 goto 语句61

4.4.2 while 语句61

4.4.3 do...while 语句62

4.4.4 for 语句63

4.4.5 3 种循环语句的比较	65	小结	133
4.4.6 循环嵌套	65	习题 6	134
4.4.7 break 语句	66	第 7 章 指针	139
4.4.8 continue 语句	66	7.1 指针与指针变量的概念	139
4.4.9 空语句	67	7.1.1 指针的概念	139
4.4.10 循环结构程序设计举例	67	7.1.2 指针变量	139
小结	69	7.2 指针变量的定义和引用	141
习题 4	69	7.2.1 指针变量的定义	141
第 5 章 数组	74	7.2.2 指针变量的赋值	141
5.1 一维数组的定义和引用	74	7.2.3 指针变量的引用	142
5.1.1 一维数组的定义	74	7.3 指针和地址运算	145
5.1.2 一维数组元素的引用	75	7.4 指针与数组	147
5.1.3 一维数组的初始化	77	7.4.1 指针变量与数组	147
5.1.4 一维数组程序举例	77	7.4.2 指针变量在一维数组中的应用	149
5.2 二维数组	80	7.4.3 指针变量在多维数组中的应用	150
5.2.1 二维数组的定义	80	7.5 指针与字符串	154
5.2.2 二维数组元素的引用	81	7.6 指针数组与多级指针	157
5.2.3 二维数组的初始化	81	7.6.1 指针数组的定义	157
5.2.4 二维数组程序举例	82	7.6.2 指针数组的使用	157
5.3 字符数组	86	7.6.3 多级指针	159
5.3.1 字符数组的定义	86	7.7 指针变量与函数	161
5.3.2 字符数组的初始化	86	7.7.1 函数的操作方式与指针变量	161
5.3.3 字符数组的引用	87	7.7.2 指针型函数的定义与使用	161
5.3.4 字符串	87	7.7.3 函数指针的定义与使用	162
5.3.5 字符串的输入/输出	88	7.7.4 与指针有关的函数参数传递方式	164
5.3.6 字符串操作函数	91	7.7.5 带参数的 main 函数和命令行参数	168
5.3.7 字符数组应用举例	92	7.8 指针与动态内存分配	169
小结	95	7.8.1 动态存储的概念	169
习题 5	96	7.8.2 C 语言的动态存储管理方式	170
第 6 章 函数	100	小结	171
6.1 函数概述	100	习题 7	172
6.1.1 函数的定义	102	第 8 章 结构体与共用体	177
6.1.2 函数的调用	104	8.1 结构体类型和结构体变量	177
6.1.3 变量的生命周期及作用域	108	8.1.1 结构体类型的定义	177
6.2 函数参数传递	115	8.1.2 结构体变量的定义	178
6.2.1 值传递	118	8.1.3 结构体变量的引用	179
6.2.2 地址传递	119	8.1.4 结构体变量的初始化	180
6.3 递归函数	126		
6.4 内部函数与外部函数	131		

8.2 结构体数组	181	9.3.2 读/写字符函数	206
8.2.1 结构体数组的定义	181	9.3.3 读/写字符串函数	208
8.2.2 结构体数组的引用	181	9.3.4 读/写数据函数	210
8.2.3 结构体数组的初始化	182	9.3.5 格式读/写函数	212
8.3 结构体类型指针	183	9.4 其他函数	213
8.3.1 指向结构体变量的指针	183	9.4.1 文件头定位函数	213
8.3.2 指向结构体数组的指针	185	9.4.2 文件随机定位函数	214
8.3.3 结构体变量和指向结构体 指针作为函数参数	185	9.4.3 出错检测函数	215
8.4 动态内存分配	187	小结	215
8.5 共用体	189	习题 9	215
8.5.1 共用体类型的定义	189	第 10 章 编译预处理	217
8.5.2 共用体变量的定义	190	10.1 宏定义	217
8.5.3 共用体变量的引用和初始化	190	10.1.1 不带参数的宏定义	217
8.6 枚举类型	191	10.1.2 带参数的宏定义	219
8.6.1 枚举类型的说明	192	10.2 文件包含处理	221
8.6.2 枚举型变量的定义	192	10.3 条件编译	223
8.7 用 typedef 定义类型	194	小结	226
小结	195	习题 10	227
习题 8	196	附录 A C 语言的字符集 -ASCII 字符表	228
第 9 章 文件	200	附录 B C 语言的关键字	229
9.1 文件概述	200	附录 C C 语言的库函数	230
9.1.1 文件的概念	200	附录 D Turbo C 常见英文 错误提示及含义	237
9.1.2 文件类型指针	202	附录 E 习题参考答案	241
9.2 文件的打开与关闭	203	参考文献	258
9.2.1 文件的打开	203		
9.2.2 文件的关闭	205		
9.3 文件的读写	206		
9.3.1 文件尾测试函数	206		

第 1 章

程序设计基础

1.1 程序设计的基本概念

电子计算机是 20 世纪人类最伟大、最杰出的科技发明之一，对人类社会的发展具有极其深远的影响。目前，计算机已经广泛而深入地渗透到人类社会的各个领域，极大地增加了人类认识世界和改造世界的能力。

一个完整的计算机系统由硬件系统和软件系统两大部分所组成。硬件是计算机系统物理装置的总称，它可以是电子的、机械的、光/电的元件或装置；软件是指在计算机硬件上运行的各种程序、数据和一些相关的文档、资料等。如果没有程序，计算机什么也不会做。

1.1.1 程序

从自然语言的角度来说，程序是对解决某个问题的方法步骤的描述；从计算机的角度来说，程序是用某种计算机能识别并可执行的计算机语言描述解决问题的方法步骤。

程序的特点是有始有终、每个步骤都能操作、所有步骤执行完对应问题要能得到解决。

例 1.1 求解任一圆的面积。

操作步骤如下。

第 1 步：输入圆的半径 R 。

第 2 步：利用公式 $S=3.14 \times R \times R$ ，求出圆的面积 S 。

第 3 步：输出结果 S 。

以上操作步骤就是求解任一圆面积的程序。

例 1.2 求 1~100 的和。

操作步骤如下。

第 1 步：初始化变量 $S=0$ 和计数器 $I=1$ 。

第 2 步： $S=S+I$

$I=I+1$

第 3 步：判断计数器 I 的值有没有大于 100，如果没有，返回到第 2 步执行；否则执行第 4 步。

第 4 步：输出结果 S 的值。

以上操作步骤就是求 1~100 的和程序。

1.1.2 算法与数据结构

著名计算机科学家沃斯 (N.Wirth) 曾经提出过一个经典公式: 程序=算法+数据结构。

1. 算法

程序设计的关键是解题的方法与步骤——算法, 它反映了计算机的执行过程, 是对解决特定问题操作步骤的一种描述。

算法可分为两大类: 数值运算算法和非数值运算算法。数值运算算法即求数值解, 通过运算得出一个具体值。数值运算一般有现成的模型, 算法较成熟。非数值运算算法: 用于事务管理, 如图书检索、人事管理等, 具体对语言而言, 其语法就是工具, 是算法的一个具体实现和描述。所以, 在程序设计的学习中, 一方面要熟练掌握高级语言的语法, 因为它是算法实现的基础; 另一方面, 还要掌握程序设计的基本方法, 更重要的是必须认识算法的重要性, 加强思维训练, 达到写出高质量程序的目的。下面先看两个常用的简单算法。

例 1.3 累加求: $1+3+5+\cdots+99$ 。

操作步骤如下。

第 1 步: $1+3 \rightarrow S$ (结果)

第 2 步: $S+5 \rightarrow S$ (结果)

第 3 步: $S+7 \rightarrow S$ (结果)

⋮

第 50 步: $S+99 \rightarrow S$ (结果)

这样的算法虽然正确, 但太烦琐。

改进的算法操作步骤如下。

第 1 步: 初始化变量 $S=0$ 和计数器 $I=1$ 。

第 2 步: $S=S+I$

$I=I+2$

第 3 步: 判断计数器 I 的值有没有大于 99, 如果没有, 返回到第 2 步执行; 否则执行第 4 步。

第 4 步: 输出结果 S 的值。

一个优秀的算法应该具备以下特性。

(1) 有穷性: 一个算法在合理的范围内应包含有限的操作步骤, 而不能是无限的。

(2) 确定性: 算法中每一个步骤应当是唯一的和确定无误的, 而不能出现含糊、模棱两可而产生歧义的。

(3) 有零个或多个输入: 在执行算法时, 需从外界得到必要的信息。

(4) 有一个或多个输出: 算法的目的是为了求解, 解就是得到的输出。一个算法得到的结果就是该算法的输出, 没有输出的算法是没有任何意义的。

(5) 有效性: 算法中每一个步骤应当能有效地执行, 并得到确定的结果。

对于程序设计人员, 必须会设计算法, 并根据算法写出程序。

2. 常用算法的表示方法

算法的表示方法有很多种, 常用的有自然语言描述、伪代码、流程图、N-S 图等。在这里重点介绍流程图和 N-S 图。

(1) 用自然语言表示算法。

用自然语言表示算法的优点是通俗易懂, 但文字冗长, 易产生歧义。除了很简单的问题, 一

般不用自然语言表示算法。

(2) 用流程图表示算法。

流程图是一种传统的算法表示法,它利用几何图形框来表示各种不同性质的操作,用流程线来指示算法的执行方向。

一个流程图包括:表示相应操作的框、带箭头的流程线以及框内外必要的文字说明。几何图形框的含义如图 1-1 所示。



图 1-1 几何图形框的含义

例 1.4 用流程图表示例 1.3,如图 1-2 所示。

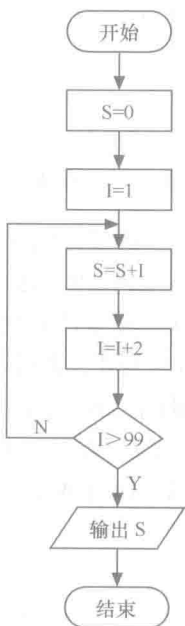


图 1-2 流程图示例

从上述例子可以看出,用流程图表示算法直观形象,易于理解,流程图是表示算法的良好工具。

注意,画流程线时,不要忘记画箭头,因为它是反映流程的执行先后次序的,如不画出箭头就很难判断各框的执行次序了。

根据结构化程序设计的思想,任何一个程序都由顺序、循环、选择 3 种基本结构组成,其流程图结构介绍如下。

① 顺序结构:如图 1-3 所示,虚线框内是一个顺序结构,其中 A 和 B 两个框是顺序执行的,即先执行 A,再执行 B。顺序结构是最简单的一种基本结构。

② 选择结构:选择结构又称分支结构。如图 1-4 所示,虚线框内是一个选择结构。注意,无论 P 条件是否成立,只能执行 A 框或 B 框之一,不可能两者都执行,如图 1-4 左图所示。其中, A 或 B 两个框中可以有一个是空的,即不执行任何操作,如图 1-4 右图所示。

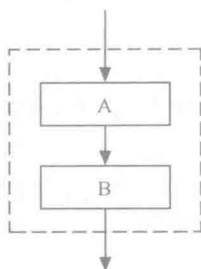


图 1-3 顺序结构

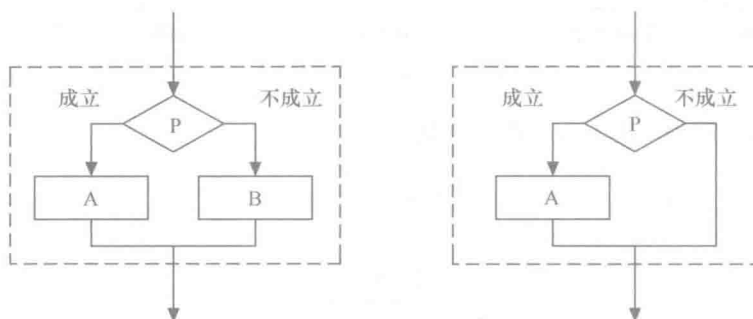


图 1-4 选择结构

③ 循环结构：循环结构又称重复结构，即反复执行某一部分的操作。循环结构有两种类型。

a. 当 (while) 型循环结构。当型循环结构如图 1-5 左图所示。它的功能是：当给定的条件 P 成立时，执行 A 框操作，执行完 A 后，再判断条件 P 是否成立，如果仍然成立，再执行 A 框，如此反复执行 A 框，直到某次 P 条件不成立为止，此时不再执行 A 框，而是脱离循环结构。

b. 直到 (until) 型循环结构。直到型循环结构如图 1-5 右图所示。它的功能是：先执行 A 框，然后判断给定的条件 P 是否成立，如果条件 P 不成立，则再执行 A 框，然后再对条件 P 作判断，如条件 P 仍不成立，又执行 A 框，如此反复执行 A 框，直到给定的条件 P 成立为止，此时不再执行 A 框，而是脱离循环结构。

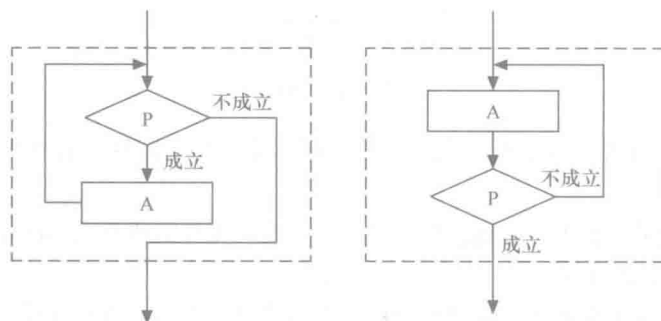


图 1-5 循环结构

以上 3 种基本结构，具有以下共同特点。

- 只有一个入口。
- 只有一个出口。
- 结构内的每一部分都有机会被执行到。

- 结构内不存在“死循环”。

(3) 用 N-S 流程图表示算法。

用流程图表示算法的优点是直观形象、表示清晰、易于理解，缺点是流程图占篇幅较多，当算法复杂时，画流程图费时且不方便。1973 年美国学者提出了一种新型流程图，即 N-S 流程图。N-S 流程图适合于结构化程序设计，因而作为编程人员，传统流程图和 N-S 流程图的使用方法都应掌握。用 N-S 流程图表示顺序结构、选择结构和循环结构分别如图 1-6、图 1-7 和图 1-8 所示。

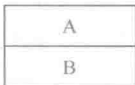


图 1-6 N-S 流程图顺序结构

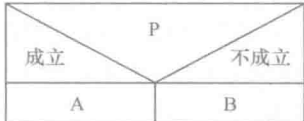


图 1-7 N-S 流程图选择结构

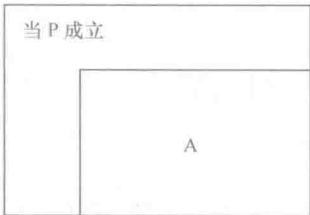


图 1-8 N-S 流程图循环结构

(4) 用伪代码表示算法。

伪代码就是用介于自然语言和计算机语言之间的文字和符号来描述算法，它不必遵守严格的语法规则。

(5) 用计算机语言表示算法。

我们的任务是用计算机解题，就是用计算机实现算法。用计算机语言表示算法必须严格遵守所用的语言的语法规则，这是和伪代码不同的。下面将对前面介绍过的例 1.4 的算法用 C 语言表示。

例 1.5 用 C 语言表示例 1.4。

```
(1_5.c)
#include<stdio.h>
void main()
{
    int i,s;
    i=2;
    s=0;
    do
    {
        s=s+i;
        i=i+2;
    }
    while(i<=99);
    printf("sum is %d\n",s);
}
```

数据结构是对参与运算的数据及它们之间的关系进行的描述，算法和数据结构是程序的两个重要方面。

1.2 程序设计的一般步骤

程序设计过程一般包括 3 个基本步骤。

第 1 步：分析问题。

作为一名程序设计者，首先要弄清楚用户的具体需求，在这里要进行用户需求分析（弄清楚用户要解决什么问题）、数据及处理分析（解决问题时需要哪些原始数据、怎么获得以及怎样处理这些数据等）、可行性分析（确定用户的需求在现有的条件下是否可解）、运行环境分析（计算机的硬件和软件是否能满足设计的需求）等。

设计者在以上分析的基础上，将实际的问题抽象化并建立相应的数学模型，最终确定设计方案。

第 2 步：确定算法。

根据选取的数学模型和确定的设计方案，设计出可操作的具体步骤，并以流程图的形式将算法清晰、直观地表示出来。

第 3 步：编程实现。

程序设计的最后一步就是选择某种计算机语言编写源程序。对于源程序，应包含注释的语句，以描述程序各个部分做何种工作。此外，编程实现还应包含程序的调试与运行。调试主要是为了测试程序在运行过程中出现的一些错误。一旦程序运行情况良好，该源程序应以文件的形式固定保留下来，以便于维护和修改。

1.3 结构化程序设计方法

结构化程序设计是以模块化设计为中心，将待开发的软件系统划分为若干个相互独立的模块，是以模块化设计为中心，将待开发的软件系统划分为若干个相互独立的模块，这样的每一个模块不会受到其他模块的牵连，因而可将原来较为复杂的问题简化为一系列简单模块的设计，使完成每一个模块的工作变得单纯而明确，为设计一些较大的软件打下了良好的基础。结构化程序设计的具体方法如下。

- (1) 自顶向下。
- (2) 逐步细化。
- (3) 模块化设计。
- (4) 结构化编码。

结构化程序设计的基本思想是采用“自顶向下，逐步求精”的程序设计方法和“单入口单出口”的控制结构。自顶向下、逐步求精的程序设计方法从问题本身开始，经过逐步细化，将解决问题的步骤分解为由基本程序结构模块组成的结构化程序框图；“单入口单出口”的思想认为，一个复杂的程序，如果它仅是由顺序、选择和循环 3 种基本程序结构通过组合、嵌套构成，那么这个新构造的程序一定是一个单入口单出口的程序。据此就很容易编写出结构良好、易于调试的程序来。

小 结

本章简要介绍了程序设计的基本概念、什么是程序以及程序设计的一般步骤、结构化程序设计方法等。本章的难点是给出一个问题，怎样设计出合适的算法并表示出来。

习 题 1

1. 什么是程序?
2. 程序设计过程一般有哪 3 个基本步骤?
3. 结构化程序设计的方法是什么?

第 2 章

C 语言概述

2.1 C 语言的发展及其特点

2.1.1 C 语言的发展简史

1972 年, 贝尔实验室的布朗·W.卡尼汉 (Brian W.Kernighan) 和丹尼斯·M.利奇 (Dennis M.Ritchie) 对 B 语言进行了完善和扩充, 在保留 B 语言强大的硬件处理能力的基础上, 扩充了数据类型, 恢复了通用性, 实现了最初的 C 语言。此后, 为了让 C 语言成为在任何计算机上都能运行的通用计算机语言, 1977 年, 由两人合写了著名的《THE C PROGRAMMING LANGUAGE》一书。

随着微型机的普及, 出现了不同版本的 C 语言, 为了统一标准, 美国国家标准协会 (American National Standards Institute) 于 1987 年制定了一个 C 语言标准, 通常称之为 ANSI C。

C 语言是当今最流行的程序设计语言之一, 它适合作为系统描述语言, 既可以用来编写系统软件, 也可以用来编写应用软件。目前最流行的 C 语言有以下几种。

- Microsoft C 或称 MS C。
- Borland Turbo C 或称 Turbo C。
- AT&T C。

这些 C 语言版本不仅实现了 ANSI C 标准, 而且在此基础上各自作了一些扩充, 使之更加方便、完美。

2.1.2 C 语言的特点

C 语言之所以发展迅速, 而且成为最受欢迎的语言之一, 主要是因为它具有强大的功能。许多著名的系统软件, 如 UNIX/Linux、Windows、dBASE III PLUS、dBASE IV, 都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序, 就更能显示 C 语言的优势, 像 PC-DOS、WordStar 等就是用这种方法编写的。

归纳起来, C 语言具有下列特点。

(1) 语言简洁、紧凑, 使用方便、灵活。C 语言一共只有 32 个关键字, 9 种控制语句, 程序书写形式自由, 主要用小写字母表示, 压缩了一切不必要的成分。

(2) 运算符丰富。C 的运算符包含的范围很广泛, 共有 34 种运算符。C 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,具有现代化语言的各种数据结构。C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结构(如链表、树、栈等)的运算。

(4) 具有结构化的控制语句(如if...else语句、while语句、do...while语句、switch语句、for语句)。用函数作为程序的模块单位,便于实现程序的模块化。C是良好的结构化语言,符合现代编程风格的要求。

(5) 语法限制不太严格,程序设计自由度大。例如,对数组下标越界不作检查,由程序编写者自己保证程序的正确;对变量的类型使用比较灵活,如整型数据与字符型数据可以通用;一行可以写多个语句等。

一般的高级语言语法检查比较严,能检查出几乎所有的语法错误;而C语言允许程序编写者有较大的自由度,因此,放宽了语法检查。程序员应当仔细检查程序,保证其正确,而不要过分依赖C编译程序去查错。“限制”与“灵活”是一对矛盾,限制严格,就失去灵活性;而强调灵活,就必然放松限制。对于一个对C语言使用不熟练的编程人员,编一个正确的C程序可能会比编一个其他高级语言程序难一些。也就是说,对于用C语言的人,要求对程序设计更熟练一些。

(6) C语言能进行位(bit)操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此,C既具有高级语言的功能,又具有低级语言的许多功能,可用来写系统软件。C语言的这种双重性,使它既是成功的系统描述语言,又是通用的程序设计语言。

(7) 生成目标代码质量高,程序执行效率高,与汇编语言相比,用C语言写的程序可移植性好。C语言一般只比汇编程序生成的目标代码效率低10%~20%,基本上不用修改就可以用于各种计算机系统。

2.2 简单的C程序介绍

2.2.1 简单的C程序

为了说明C语言源程序结构的特点,先看以下几个程序。这几个程序由简到难,表现了C语言源程序在组成结构上的特点。虽然有关内容还未介绍,但可从这些例子中了解到组成一个C源程序的基本部分和书写格式。

例 2.1 输出指定字符串。

```
(2_1.c)
#include<stdio.h>
void main()
{
    printf("Hello, this is a c programming language.\n");/*输出C程序语言字符串*/
}
```

程序运行结果:

```
Hello, this is a c programming language.
```