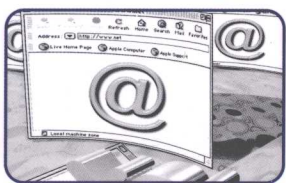
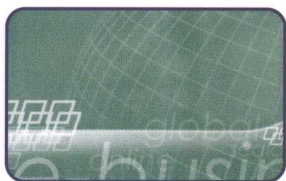




普通高等教育“十一五”规划教材

主 编 王洪海 何广龙 陈向阳 盛 魁

C语言程序设计



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



普通高等教育“十一五”规划教材

C 语言程序设计

王洪海 何广龙 陈向阳 盛 魁 主 编
薛 峰 袁 琴 王德正 徐丽萍 副主编
丁 亮 赵翠荣 方 飞

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

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。本书系统介绍了 C 语言程序设计的方法。全书共分 10 章, 首先简单介绍了程序设计基础, 然后介绍了 C 语言的数据类型与运算符、C 语言的三种程序结构、数组与函数、指针与文件等相关知识。

参与本书编写任务的作者都是教学一线的教师, 他们结合自己多年的教学经验, 针对 C 语言程序设计的教学思路, 在介绍 C 语言的同时, 注重讲解如何应用 C 语言来编写程序, 试图帮助读者克服畏难情绪, 在轻松、愉快的气氛中探索 C 程序设计的奥秘。

本书适合作为大学本科学学生学习 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。

图书在版编目(CIP)数据

C 语言程序设计 / 王洪海等主编. —北京: 中国铁道出版社, 2009. 11

ISBN 978-7-113-10552-5

I. C… II. 王… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 201723 号

书 名: C 语言程序设计

作 者: 王洪海 何广龙 陈向阳 盛 魁 主编

策划编辑: 严晓舟 王安新

责任编辑: 黄园园 杜 鹃

封面设计: 唐韵设计

版式设计: 于 洋

编辑部电话: (010) 63583215

封面制作: 白 雪

责任印制: 李 佳

出版发行: 中国铁道出版社(北京市宣武区右安门西街 8 号 邮政编码: 100054)

印 刷: 河北省遵化市胶印厂

版 次: 2009 年 11 月第 1 版 2009 年 11 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 18.25 字数: 453 千

书 号: ISBN 978-7-113-10552-5/TP·3576

定 价: 31.00 元

版权所有 侵权必究

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

参 编 单 位

(排名不分先后)

安徽三联学院

安徽黄山学院

安徽职业技术学院

安徽文达信息技术职业学院

安庆医药高等专科学校

亳州职业技术学院

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。

本书适合作为大学本科学学生学习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。和同类书相比,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、通俗易懂等特点。在编写过程中,编者遵循了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,注意了例题的选择,增加了例题和习题的数量、类型。讲述中力求理论联系实际和循序渐进,注重培养读者分析问题和设计程序的能力,使读者养成良好的程序设计风格和习惯。

本书特点:

① 面向初学者。书中实例充满趣味性和实用性,语言叙述通俗易懂,难点分散,概念清晰,层次分明。

② 实践性强。本书实例与习题众多,注重各部分知识的综合应用训练。

③ 所有章节后的习题均有参考答案。

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。其中,第1章、第9章由王洪海编写,第2章由陈向阳、丁亮编写,第3章、第5章、第6章由何广龙、袁琴编写,第4章由徐丽萍编写,第7章由盛魁编写,第8章由王德正编写,第10章由赵翠荣编写,全书由王洪海统稿和定稿,由中国科学技术大学原教授赵守忠主审。

为了方便读者学习,本书配有实验指导教材《C语言程序设计实验指导》(王洪海等主编),帮助读者梳理所学知识,掌握应会和必会的内容。

由于编者水平有限,书中难免存在疏误之处,诚挚地希望广大读者提出宝贵意见和建议,邮件请发至 sanlian_whh@163.com。

编 者

2009年9月

目 录

第 1 章 程序设计基础.....	1	3.4.2 变量的初始化	23
1.1 程序设计的基本概念.....	1	3.5 运算符及其表达式	26
1.1.1 程序概述	1	3.5.1 算术运算符和算术 表达式	27
1.1.2 算法与数据结构	2	3.5.2 赋值运算符和赋值 表达式	30
1.2 程序设计的一般步骤.....	6	3.5.3 关系运算符和关系 表达式	32
1.3 结构化程序设计方法.....	6	3.5.4 逻辑运算符和逻辑 表达式	33
本章小结	7	3.5.5 条件运算符和条件 表达式	36
习题 1	7	3.5.6 逗号运算符和逗号 表达式	37
第 2 章 C 语言概述	8	3.5.7 位运算	38
2.1 C 语言的发展及其特点	8	3.5.8 指针运算符、sizeof 运算符	40
2.1.1 C 语言的发展简史	8	3.6 运算符的优先级和结合性	41
2.1.2 C 语言的特点	8	3.7 表达式中数据类型的转换	42
2.2 简单的 C 语言程序介绍	9	本章小结	46
2.2.1 简单的 C 语言程序	9	习题 3	47
2.2.2 C 语言源程序的结构特点	11	第 4 章 结构化的 C 语言程序设计	49
2.2.3 C 语言源程序的书写规则	11	4.1 结构化程序设计概述	49
2.3 运行 C 语言程序的基本步骤	11	4.2 顺序结构程序设计	50
本章小结	12	4.2.1 赋值语句、复合语句	50
习题 2	12	4.2.2 字符输出函数	50
第 3 章 数据类型、运算符及表达式	13	4.2.3 字符输入函数	51
3.1 C 语言的数据类型	13	4.2.4 格式化输出函数	52
3.2 标识符与关键字	14	4.2.5 格式化输入函数	54
3.3 常量	14	4.2.6 顺序结构程序设计举例	56
3.3.1 整型常量	14		
3.3.2 实型常量	16		
3.3.3 字符型常量	16		
3.3.4 字符串常量	18		
3.3.5 符号常量	20		
3.4 变量	21		
3.4.1 变量的定义	21		

4.3 选择结构程序设计	58	5.3.4 字符串	93
4.3.1 简单的 if 结构	58	5.3.5 字符串的输入/输出	94
4.3.2 if...else 语句结构	59	5.3.6 字符串操作函数	97
4.3.3 if 结构的嵌套	60	5.3.7 字符数组应用举例	98
4.3.4 if...else if 语句结构	61	本章小结	101
4.3.5 switch 结构	62	习题 5	102
4.3.6 选择结构程序设计举例	63	第 6 章 函数	107
4.4 循环结构程序设计	64	6.1 函数概述	107
4.4.1 goto 语句	64	6.1.1 函数的定义	109
4.4.2 while 语句	65	6.1.2 函数的调用	111
4.4.3 do...while 语句	66	6.1.3 变量的生命期及作用域	116
4.4.4 for 循环	67	6.2 函数参数传递	124
4.4.5 三种循环语句的比较	68	6.2.1 值传递	127
4.4.6 循环嵌套	68	6.2.2 地址传递	129
4.4.7 break 语句	69	6.3 递归函数	137
4.4.8 continue 语句	69	6.4 内部函数与外部函数	143
4.4.9 空语句	70	本章小结	145
4.4.10 循环结构程序设计举例	70	习题 6	146
本章小结	72	第 7 章 指针	152
习题 4	73	7.1 指针与指针变量的概念	152
第 5 章 数组	78	7.1.1 指针的概念	152
5.1 一维数组的定义和引用	78	7.1.2 指针变量	152
5.1.1 一维数组的定义	78	7.2 指针变量的定义和引用	154
5.1.2 一维数组元素的引用	79	7.2.1 指针变量的定义	154
5.1.3 一维数组的初始化	81	7.2.2 指针变量的赋值	155
5.1.4 一维数组程序举例	82	7.2.3 指针变量的引用	155
5.2 二维数组	85	7.3 指针和地址运算	158
5.2.1 二维数组的定义	85	7.4 指针与数组	160
5.2.2 二维数组元素的引用	86	7.4.1 指针变量与数组	160
5.2.3 二维数组的初始化	86	7.4.2 指针变量在一维数组中 的应用	162
5.2.4 二维数组程序举例	87	7.4.3 指针变量在多维数组中 的应用	163
5.3 字符数组	91	7.5 指针与字符串	168
5.3.1 字符数组的定义	91	7.6 指针数组与多级指针	171
5.3.2 字符数组的初始化	92		
5.3.3 字符数组的引用	93		

7.6.1 指针数组的定义	171	8.4 动态内存分配	206
7.6.2 指针数组的使用	171	8.5 共用体	208
7.6.3 多级指针	173	8.5.1 共用体类型的定义	209
7.7 指针变量与函数	176	8.5.2 共用体变量的定义	209
7.7.1 函数的操作方式与指针 变量	176	8.5.3 共用体变量的引用和 初始化	210
7.7.2 指针型函数的定义与 使用	176	8.6 枚举类型	211
7.7.3 函数指针的定义与使用	177	8.6.1 枚举类型的说明	211
7.7.4 与指针有关的函数参数 传递方式	179	8.6.2 枚举型变量的定义	212
7.7.5 带参数的 main()函数和 命令行参数	184	8.7 用 typedef 定义类型	214
7.8 指针与动态内存分配	185	本章小结	215
7.8.1 动态存储的概念	185	习题 8	216
7.8.2 C 语言的动态存储管理 方式	186	第 9 章 文件	221
本章小结	187	9.1 文件概述	221
习题 7	188	9.1.1 文件的概念	221
第 8 章 结构体与共用体	194	9.1.2 文件类型指针	223
8.1 结构体类型和结构体变量	194	9.2 文件的打开与关闭	224
8.1.1 结构体类型的定义	194	9.2.1 文件的打开	224
8.1.2 结构体变量的定义	195	9.2.2 文件的关闭	226
8.1.3 结构体变量的引用	197	9.3 文件的读/写	227
8.1.4 结构体变量的初始化	198	9.3.1 文件尾测试函数	227
8.2 结构体数组	198	9.3.2 读/写字符函数	228
8.2.1 结构体数组的定义	198	9.3.3 读/写字符串函数	231
8.2.2 结构体数组的引用	199	9.3.4 读/写数据函数	232
8.2.3 结构体数组的初始化	200	9.3.5 格式读/写函数	234
8.3 结构体类型指针	202	9.4 其他函数	236
8.3.1 指向结构体变量的指针	202	9.4.1 文件头定位函数	236
8.3.2 指向结构体数组的 指针	203	9.4.2 文件随机定位函数	237
8.3.3 结构体变量和指向结构体 指针作为函数参数	204	9.4.3 出错检测函数	238
		本章小结	239
		习题 9	239
		第 10 章 编译预处理	240
		10.1 宏定义	240
		10.1.1 不带参数的宏定义	240
		10.1.2 带参数的宏定义	242

10.2 文件包含处理	244	附录 C Turbo C 常见英文错误提示	
10.3 条件编译	247	及含义	255
本章小结	250	附录 D C 语言的库函数	259
习题 10	251	附录 E 习题参考答案	266
附录 A ASCII 码表	252	参考文献	284
附录 B C 语言的关键字	254		

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。计算机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 程序概述

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，步骤如下：

第一步：输入圆的半径 R 。

第二步：利用公式 $S=3.14 \times R \times R$ ，求出圆的面积 S 。

第三步：输出结果 S 。

以上步骤就是求解任一圆面积的程序。

【例 1.2】求 1~100 的和。

第一步：初始化变量 $S=0$ 和计数器 $I=1$ 。

第二步： $S=S+I$

$I=I+1$

第三步：判断计数器 I 的值有没有大于 100，如果没有，返回到第二步执行；否则执行第四步。

第四步：输出结果 S 的值。

以上步骤就是求 1~100 和的程序。

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, 算法比较成熟。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, 在学习程序设计的过程中, 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】累加求和 $1+3+5+\cdots+99$ 。

第一步: $1+3 \rightarrow S$ (结果)。

第二步: $S+5 \rightarrow S$ (结果)。

第三步: $S+7 \rightarrow S$ (结果)。

.....

第五十步: $S+99 \rightarrow S$ (结果)。

这样的算法虽然正确, 但太烦琐。

改进的算法:

第一步: 初始化变量 $S=0$ 和计数器 $I=1$ 。

第二步: $S=S+I$

$I=I+2$

第三步: 判断计数器 I 的值有没有大于 99, 如果没有, 返回到第二步执行; 否则执行第四步。

第四步: 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。用它表示算法直观、形象、易于理解。

一个流程图包括:表示相应操作的框,带箭头的流程线,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。

从上述例子可以看出,流程图表示算法,直观形象,易于理解,是表示算法的良好工具。一个流程图包括以下几部分:

- ① 表示相应操作的框。
- ② 带箭头的流程线。
- ③ 框内外必要的文字说明。

注意,画流程线时,不要忘记画箭头,因为它是反应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。

根据结构化程序设计的思想,任何一个程序都由顺序、循环、选择三种基本结构组成,其流程图结构如图 1-3 所示。

① 顺序结构: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。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 (a) 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 (b) 所示。

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:

- 当 (while) 型循环结构。当型循环结构如图 1-5 (a) 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。

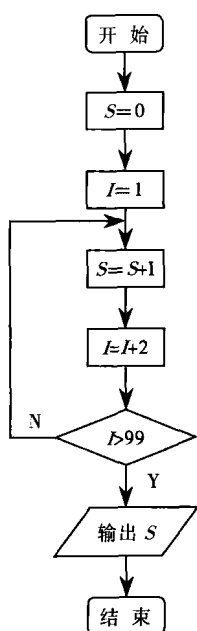


图 1-2 流程图示例

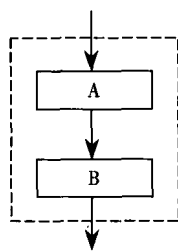
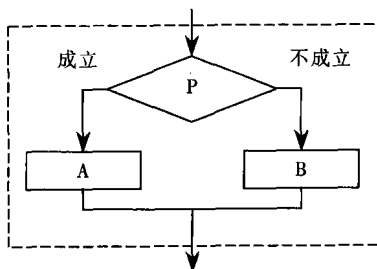
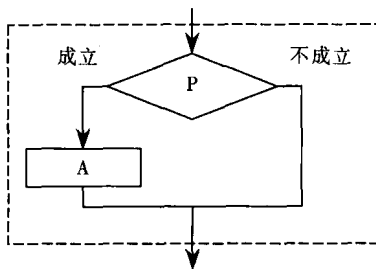


图 1-3 顺序结构



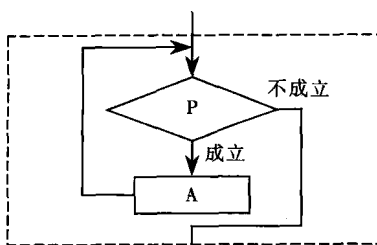
(a)



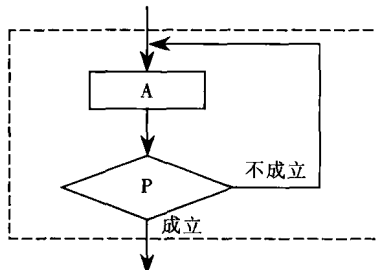
(b)

图 1-4 选择结构

- 直到 (until) 型循环结构。直到型循环结构如图 1-5 (b) 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，然后再对条件 P 做判断，如条件 P 仍不成立，又执行 A……如此反复执行 A，直到给定的条件 P 成立为止，此时不再执行 A，而是脱离循环结构。



(a)



(b)

图 1-5 循环结构

以上三种基本结构，具有以下的共同特点：

- ① 只有一个入口。
- ② 只有一个出口。
- ③ 结构内的每一部分都有机会被执行到。
- ④ 结构内不存在“死循环”。

(3) 用 N-S 流程图表示算法

流程线表示算法优点是直观形象、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：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 图都应掌握其使用方法。

- ① 顺序结构流程图如图 1-6 所示。
- ② 选择结构流程图如图 1-7 所示。

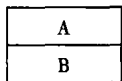


图 1-6 N-S 流程图顺序结构

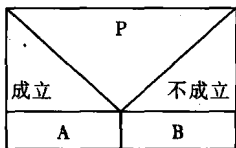
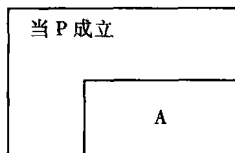
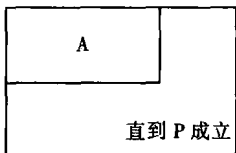


图 1-7 N-S 流程图选择结构

- ③ 循环结构流程图如图 1-8 所示。



(a)



(b)

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的，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。

程序代码如下：

```
#include<stdio.h>
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三个基本步骤：

第一步：分析问题。

作为一名程序设计者，要弄清楚用户的具体需求，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。

第二步：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。

第三步：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以后，该源程序应以文件的形式固定保留下来，便于维护和修改。

1.3 结构化程序设计方法

结构化程序设计是由迪克斯特拉（E.W.Dijkstra）在 1969 年提出的，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、选择和循环三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。

本章小结

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三个基本步骤?
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.卡尼汉和丹尼斯.M.利奇对 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, 出现了版本不同的 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 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 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。