

C语言程序设计

主编 蒲先祥 张舜尧 何风琴



中国海洋大学出版社
CHINA OCEAN UNIVERSITY PRESS

C 语言程序设计

主 编 蒲先祥 张舜尧 何风琴

中国海洋大学出版社

图书在版编目 (CIP) 数据

C 语言程序设计 / 蒲先祥, 张舜尧, 何风琴主编. — 青岛:
中国海洋大学出版社, 2017.12
ISBN 978-7-5670-1566-1

I . ① C… II . ①蒲… ②张… ③何… III . ① C 语言—
程序设计 IV . ① TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 228116 号

出版发行 中国海洋大学出版社
社 址 青岛市香港东路 23 号
出 版 人 杨立敏
网 址 <http://www.ouc-press.com>
电子信箱 sjyybook@163.com
订购电话 010-60739092
责任编辑 王积庆
印 制 廊坊市广阳区九洲印刷厂
版 次 2017 年 12 月第 1 版
印 次 2017 年 12 月第 1 次印刷
成品尺寸 185mm × 260mm
印 张 14
字 数 327 千字
定 价 65.00 元

邮政编码 266071

电 话 0532-85902349

印 数 1-3000



PREFACE

前言

C语言是一种结构化程序设计语言，它功能丰富、表达能力强、使用灵活、应用面广、可移植性好，既具有高级语言的特点，又具有低级语言的许多功能；既可以用来编写系统软件，也可以用来编写应用软件。C语言作为一门较为通用的编程语言，得到越来越广泛的应用。目前，C语言程序设计被许多学校列为程序设计课程的首选，全国各省市区的计算机等级考试都将C语言列入了考试范围。

本书是针对非计算机专业初学者的特点编写的，把提高编程能力、阅读程序的能力放在重要地位，避免在程序设计教学过程中陷入程序设计语言繁杂的语法和格式的学习，而是将主要精力集中在所要解决的实际问题上，从知识点以及具体问题出发，重点讲述如何通过程序设计来解决问题。

本书是根据作者多年的教学经验和教学改革成果编写而成的。全书以程序设计为主线，以编程应用为出发点，通过案例和问题引入内容，重点讲解程序设计的思想和方法，并介绍相关的语言知识，注重培养读者分析问题和程序设计的能力，注重培养良好的程序设计风格和习惯。本书讲述力求理论联系实际、循序渐进，全书通过大量例题验证语法和说明程序设计方法，并且为了使读者更好地掌握各章节内容，各章末配有精选的习题。通过本书的学习和解题练习，读者既能迅速掌握C语言的基础知识，又能很快地学会C语言的编程方法，提高解决实际问题的能力。

为了更好地理解C语言编程的思想，本书将所学知识整合贯通，从而提高学生综合分析问题和解决问题的能力，为以后各种编程语言的学习打下坚实的基础。本书还安排了一个综合案例——学生成绩管理系统，综合案例的设计几乎涵

盖了C语言的所有知识点，是对C语言的综合应用和所学知识的总结，旨在帮助读者掌握完整的课程脉络。

本书具有以下特点：

1. 由表及里，逻辑性强

本书的开始部分介绍了C语言程序的基本结构和开发环境，使读者可以从感性上认识C语言程序的基本组成。后边的章节以此为基础，分模块对深层的程序设计要素进行了介绍，能够有效地引导学生逐步学习和掌握C语言程序设计。

2. 案例丰富，层次感强

本书十分重视内容的实践性，在理论阐述的基础上，通过大量案例的分析介绍，明确了C语言程序设计的要点。案例引入层次鲜明、生动形象，能够有效地启发学生对所学知识进行拓展运用。

本书在编写过程中参考了大量C语言程序设计相应成果，限于篇幅未能一一列出，这里对这些作者和学者表示诚挚的歉意。由于编者水平有限，书中难免存在疏漏和不足，希望广大读者及时批评指正，以便本书的日后完善。

编 者



目录 CONTENTS

第 1 章 C 语言概述

1.1 C 语言的历史背景及特点	2
1.2 C 语言程序的基本构成	6
1.3 C 语言运行步骤	10
实训与习题	12

第 2 章 基本的数据类型及运算符

2.1 C 语言的数据类型	16
2.2 C 语言的运算符和表达式	25
2.3 数据的混合运算	30
实训与习题	31

第 3 章 顺序结构程序设计

3.1 算法与结构化程序设计	36
3.2 C 语言的语句	41
3.3 数据输出	43
3.4 数据的输入	47
3.5 顺序结构	51
实训与习题	52

第 4 章 选择结构程序设计

4.1 选择结构	56
4.2 条件的表示	56
4.3 if 语句	61
4.4 用 switch 语句实现多分支选择结构	67

实训与习题	70
-------------	----

第 5 章 循环结构程序设计

5.1 循环结构及其算法	74
5.2 用 while 语句实现循环结构	76
5.3 用 do-while 语句实现循环结构	77
5.4 用 for 语句实现循环	79
5.5 循环的嵌套	82
5.6 break 语句和 continue 语句	85
实训与习题	87

第 6 章 数 组

6.1 一维数组的定义和使用	92
6.2 二维数组的定义和使用	99
实训与习题	102

第 7 章 函 数

7.1 函数的作用	106
7.2 函数的类别	107
7.3 函数的定义	108
7.4 函数的调用	109
7.5 数组与函数	118
7.6 局部变量和全局变量	123
实训与习题	129

第 8 章 指 针

8.1 指针的基本概念	134
8.2 变量与指针	135
8.3 数组与指针	142
8.4 字符串与指针	152
8.5 函数与指针	158
8.6 返回指针值的函数	161
8.7 指针数组及双重指针	163

实训与习题	168
-------------	-----

第 9 章 结构体与链表

9.1 结构体	172
9.2 结构体指针变量	177
9.3 用 typedef 命名已有数据类型	182
9.4 动态内存分配	184
9.5 链表的构造与处理	185
实训与习题	190

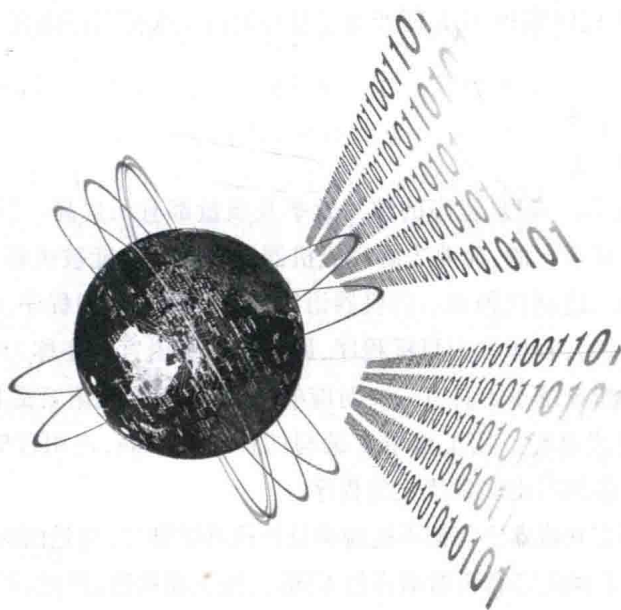
第 10 章 文 件

10.1 概述	196
10.2 文件的打开和关闭	197
10.3 文件的读写	200
10.4 文件的定位操作	208
实训与习题	210
附 录	214
参考文献	216

第 1 章

C 语言概述

本章介绍了计算机程序设计语言的功能、算法的概念及其描述、C 语言的发展历史、C 语言的特点、C 语言的程序结构和上机步骤。本章学习重点要求掌握算法的描述、C 语言的程序结构和上机运行 C 语言的程序结构。学完本章之后,读者将对程序设计以及 C 语言有一个初步的了解。





1.1 C 语言的历史背景及特点

C 语言是国际上广泛流行的高级程序设计语言,是目前程序设计领域中具有较大影响的程序设计语言之一,它既可以用来编写系统软件,又可以用来编写应用软件。

1.1.1 C 语言的历史背景

语言是人们交换思想的工具,我们在日常生活中使用的汉语、英语等称为自然语言。计算机诞生以后,人们要指挥计算机工作,就产生了计算机语言。用于程序设计的计算机语言基本上可分为三种:机器语言、汇编语言和高级语言。

1. 机器语言

计算机诞生的初期,人们使用的计算机语言仅由计算机能够识别的 0 和 1 代码组成,被称为机器语言。下面是某 CPU 指令系统中的两条指令:

1 0 0 0 0 0 0 0 (进行一次加法运算)

1 0 0 1 0 0 0 0 (进行一次减法运算)

用机器语言编程序,就是从所使用的 CPU 的指令系统中挑选合适的指令,组成一个指令序列。这种程序虽然可以被机器直接理解和执行,却由于它们不直观、难记、难认、难理解、不易查错,只能被少数专业人员掌握,并且由于编写程序的效率很低,质量难以保证,这使计算机的推广使用受到了极大的限制。

2. 汇编语言

为减轻人们在编程中的劳动强度,20 世纪 50 年代中期人们开始用一些英文助记符号来代替 0、1 代码编程,于是便产生了符号语言(或称汇编语言)。如前面的两条机器指令可以写为:

ADD A, B

SUB A, B

用汇编语言编程,程序的编写效率及质量都有所提高。但是,汇编语言指令是机器不能直接识别和执行的,而要先翻译成机器语言程序才能被机器识别和执行。将汇编语言程序转换成为二进制代码表示的机器语言的程序称为汇编程序,经汇编程序“汇编(翻译)”得到的机器语言程序称为目标程序,原来的汇编语言程序称为源程序。由于汇编语言指令与机器语言指令基本上具有一一对应的关系,所以汇编语言源程序的代换可以由汇编系统以查表的方式进行。用汇编语言编写的程序效率高,占用存储空间小,运行速度快;而且用汇编语言能编写出非常优化的程序。

汇编语言和机器语言都不能脱离具体机器即硬件,均是面向机器的语言。不同类型的计算机所用的汇编语言和机器语言是不同的,缺乏通用性,因此,汇编语言被称为低级语言。用

面向机器的语言编程,可以编出效率极高的程序,但是程序员用它们编程时,不仅要考虑解题思路,还要熟悉机器的内部结构,并且要“手工”进行存储器的分配,因而其劳动强度仍然很大,给计算机的普及推广造成了很大的障碍。

3. 高级语言

1954年出现的FORTRAN语言以及随后相继出现的其他高级语言,开始使用接近人类自然语言的、但又消除了自然语言中二义性的语言来描述程序。高级语言不受具体机器的限制,使用了许多数学公式和数学计算上的习惯用语,非常擅长科学计算。用高级语言编写的程序通用性强,直观、易懂、易学,可读性好。到目前为止,世界上有数百种高级语言,常用的有几十种,如FORTRAN、PASCAL、C、LISP、COBOL等。这些高级语言使人们开始摆脱进行程序设计必须先熟悉机器的桎梏,把精力集中于解题思路和方法上,使计算机的使用得到了迅速普及。

C语言是高级程序设计语言,是由ALGOL 60逐渐演变而来的。早在1963年,英国剑桥大学根据当时流行的高级语言ALGOL 60推出一种接近于硬件的语言CPL (combined programming language)。1967年,英国剑桥大学Martin Richards为编写操作系统软件和编译器,针对当时的CPL语言提出了一种改进的语言,称为BCPL (basic combined programming language) 语言。

20世纪60年代, Bell实验室的Ken Thompson着手开发后来对计算机产生了巨大影响的Unix操作系统。为了描述Unix, Thompson首先将BCPL语言改进为他称为B的语言。1970年, Thompson发表了用汇编语言和B语言写成的在PDP-7上实现的Unix初版。BCPL和B语言都是“数据无类型”语言,即每一个数据项都占用内存中的一个字节,处理数据项的任务落在了程序员的身上。

1971年, Dennis Ritchie开始协助Thompson开发Unix。他对B语言做了进一步的充实和完善,加入数据类型和新的句法,于1972年在一台DEC PDP-11计算机上实现了最初的C语言(取BCPL的第2个字母)。为了推广Unix操作系统,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》。从此, C语言借助Unix操作系统而得以快速传播, Unix操作系统也由于C语言而得以快速移植并落地生根,两者相辅相成。1978年, Brian W. Kernighian和Dennis M. Ritchie出版了名著*The C Programming Language*,从而使C语言成为目前世界上流行广泛的高级程序设计语言之一。以后,又有多种程序设计语言在C语言的基础上产生,如C++、Java、C#等。实际上,当今许多新的重要的操作系统都是用C或C++语言编写的。在过去30多年里, C语言已经能够用在绝大多数计算机上了。C语言发展迅速,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dBASE III PLUS、dBASE IV都是用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。

C语言的形成可简单地描述如下。

ALGOL (1960年) → CPL (1963年) → BCPL (1967年) → B (1970年) → C (1972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 C 语言之间出现了许多不一致的地方。为了改变这种状况,1983 年美国国家标准研究所 (ANSI) 为 C 语言制定了第一个 ANSI 标准,称为 ANSI C (83 ANSI C)。1987 年美国国家标准研究所又公布了新的 C 语言标准,称为 87 ANSI C。这个标准在 1989 年被国际标准化组织 (ISO) 采用,被称为 ANSI/ISO Standard C (即 C89)。Brian W.Kernighian 和 Dennis M.Ritchie 根据这个标准,重写了他们的经典著作,并发表了 *The C Programming Language, Second Edition*。后来流行的各种 C 语言编译系统的版本大多数都是以此为基础的,但是它们彼此又有不同。此后在计算机上使用的 C 语言编译系统如 Microsoft C、Turbo C、Borland C、Quick C 等,都是按标准 C 语言编写的,相互之间略有差异。每一种编译系统又有着不同的版本,版本之间也有差异,主要表现在功能上。版本越高的编译系统所提供的函数越多,编译能力越强,使用越方便,用户界面越友好。

1983 年, Bell 实验室又推出了 C++ 语言,该语言在 C 语言基础上进行了改进和革新。C++ 语言和 C 语言在很多方面是兼容的,然而, C++ 语言是一种面向对象的程序设计语言。掌握了 C 语言,对今后学习 C++ 语言是很有帮助的。1995 年 Bell 实验室又为 C 语言增加了一些新的函数,使之具有 C++ 的一些特征,使 C89 成为 C++ 的子集。1999 年 Bell 实验室推出的 C99 在基本保留 C 语言特征的基础上,增加了一系列面向对象的新特征。从此, C 语言也从面向过程的语言发展成为面向对象的语言。

计算机处理数据的基本单元是计算机指令。单独的一条指令本身只能完成计算机的一个最基本的功能,计算机所能实现的指令的集合称为计算机的指令系统。虽然一条计算机指令只能实现一个简单的功能,而且指令系统的指令数目也是有限的,但是一系列有序的指令组合却能完成很复杂的功能。一系列计算机指令的有序组合就构成了程序。

一般情况下,程序的执行是按照指令的排列顺序一条一条地执行的,但是有的程序往往需要通过判断不同的情况执行不同的指令分支,还有些指令需要被反复执行。

程序在计算机中是以 0、1 组成的指令码 (即机器语言) 来表示的,即程序是 0、1 组成的序列,这个序列能被计算机所识别。一般情况下,程序和数据均存储在存储器中 (这种结构称为冯·诺依曼结构,而程序和数据分开存储的结构称为哈佛结构)。当程序要运行的时候,当前准备运行的指令从内存中被调入 CPU,由 CPU 处理该指令。

1.1.2 C 语言的特点

C 语言是一种推出比较晚的高级语言,它吸取了早期高级语言的长处,克服了某些不足,形成了自己的风格:既具有高级语言的特点,又具有低级语言的一些功能。因此可以说, C 语言是一种很有特色的高级语言。总的来说, C 语言是一种简洁明了、功能强大、移植性好的结构化程序设计语言。

1. C 语言是一种结构化程序设计语言

C 语言适用于结构化编程方法。在 C 语言中,函数是构成结构化程序的最小模块,每个函数实现一个功能,函数之间有相对的独立性,多个函数共同实现一个大功能。C 语言程序大多是由若干个函数组成的,也可以说是由若干个模块构成的,这些模块可放在一个文件中,

也可放在多个文件中。C语言是结构化程序设计语言,它具备构成结构化程序设计的三种基本结构模式的语句。

2. C语言编程简洁紧凑,使用方便、灵活

C语言是一种非常简练的语言,语法限制不太严格,程序设计自由度大,使用C语言编写的程序简洁明了。C语言的简练表现在如下几个方面。

(1) C语言中关键字不多,有些关键字采用了简单的符号来替代。例如,条件语句中,if体和else体的定界符采用一对花括号{ }来标识,如果只有一条语句,则不使用花括号,这样比使用关键字作定界符简单多了。

(2) C语言中数据类型说明符采用缩写形式。例如,整型说明符用int,字符型说明符用char,这比说明符使用英文全称要简练一些。

(3) C语言中运算符很丰富,而且功能很强。在编程中尽量使用表达式,这样要比使用函数调用简练得多。另外,C语言中有一个三目运算符(?:)具有条件语句的功能。编程时使用三目运算符的条件表达式要比使用if-else语句简练得多。

(4) C语言还有一种其他高级语言不曾有过的预处理功能。使用该功能提供的某些预处理命令,会使程序的书写变得清晰简洁。例如,C语言程序常用文件包含命令(#include<文件名>),这样可以少写很多行语句,给程序的编写带来方便。

3. C语言功能强大

C语言功能强大,不仅表现在它具有的高级语言功能上,还表现在它具有的低级语言功能上。C语言具有所有高级语言的功能,包含数值处理功能和非数值处理功能。C语言允许直接访问物理地址,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,不能简单地把C语言说成是高级语言,它是一种介于汇编语言和高级语言之间的中级语言。

另外,C语言还提供了丰富的类型,有较多的数据类型和存储类型,使用起来比较方便灵活。除了C语言系统自有的数据结构外,程序员还可以自创类型,所以C语言使用起来灵活、多样。

4. C语言移植性好

C语言的编译系统较小,又具有一些预处理命令,因而为它的移植带来了一些方便。C语言移植性好表现在两个方面:一是C语言系统只要稍加修改,便可用于各种不同型号的计算机和各种操作系统中,并且生成的目标代码质量高,程序执行效率高;二是用C语言编写的程序可以比较方便地不同系统下运行。因此,C语言能够广泛地应用于各个领域。

5. C语言的不足

C语言具有灵活简练的特点,但在某些方面存在不足。了解C语言中的不足,对在编程中避免出错十分重要。

C语言运算符多,难用难记。C语言共有44种运算符,又分为15个优先级和两种结合性。记住这些运算符的功能,搞清楚不同的优先级,对初学者有一定难度。但是,这些运算符是编程的基础,必须掌握。另外,有些不同功能的运算符使用同一种运算符符号,还应分清它们的

区别。例如，* 号作为单目运算符表示取内容的功能，作为双目运算符表示两个操作数相乘的功能。

C 语言中类型转换比较灵活，在许多情况下不做类型检查，对类型要求不够严格。因此，在 C 语言程序中，对类型处理要谨慎，要尽量避免出现类型不一致的情况。

C 语言中，给数组进行初始化是判界的，越界会发出编译错。但是，数组动态赋值是不判界的，这样会造成数据方面的混乱。

C 语言的编译系统中，允许不同的编译系统对表达式中各个操作数和参数表中各个参数有不同的计算顺序。这对于一般表达式和参数表是没有什么影响的。但是，当表达式或参数表中出现了具有副作用的运算符时，不同计算顺序的编译系统将会造成二义性。



1.2 C 语言程序的基本构成

计算机尽管可以完成许多极其复杂的工作，但实质上这些工作都是按照事先编好的程序进行的，所以人们常把程序称为计算机的灵魂。

1.2.1 简单的 C 语言程序介绍

为了说明 C 语言程序结构的特点，先看以下程序。虽然有关内容还未介绍，但可从这个例子中了解到组成一个 C 语言程序的基本部分和书写格式。

【例 1.2.1】

```
#include<stdio.h>
main ( )
{
    printf (" 同学们,你们好! \n" );
}
```

include 称为文件包含命令，扩展名为 .h 的文件称为头文件。用 include 命令包含之后，在程序中便可以使用头文件中定义的功能函数了。

main 是主函数的函数名，表示这是一个主函数。每一个 C 语言程序都必须有且只能有一个主函数（main 函数）。

函数调用语句 printf 函数的功能是把要输出的内容送到显示器去显示。printf 函数是一个由系统定义的标准函数，可在程序中直接调用。函数原型在 stdio.h 中定义。

这是 C 语言程序中最简单的程序，只有一个文件，文件中只有一个主函数，函数中也只有一条语句，该语句的功能是将一个字符串常量输出显示在屏幕上。

一个完整的C语言程序,是由一个main()函数(又称主函数)和若干个其他函数结合而成的,当然也可以是只由一个main()函数构成的。

通过以上例子,可以概括出C语言程序的结构特点。

(1) 函数与主函数。

程序由一个或多个函数组成;

必须有且只能有一个主函数main(),main函数可以放在程序的最前面,也可以放在程序的最后面,或是放在程序的中间。当然,main函数不能放在其他函数中间,因为C语言函数不能嵌套定义。

程序执行从main开始,在main中结束,其他函数通过嵌套调用得以执行。

(2) 程序语句。

一个C语言程序可以由一个或多个源文件组成。

C程序由语句组成。

用“;”作为语句终止符,每一个说明、每一个语句都必须以分号结尾。但预处理命令、函数头和最后一个花括号{}之后不能加分号。

(3) 注释。

/* */为注释,不能嵌套,如/*.../*...*/...*/是错误的。

注释不产生编译代码。

(4) 编译预处理命令。

源程序中可以有预处理命令(#include命令仅为其中的一种),预处理命令通常应放在源文件或源程序的最前面。

1.2.2 C语言程序的书写格式

尽管C语言语句精练、简洁、语义丰富、格式灵活,然而C语言程序的可读性比较差。为了提高程序的可读性,应该养成良好的书写习惯。

C语言程序在书写格式习惯上有如下要求。

(1) 习惯用小写字母,大小写敏感。

(2) 常用锯齿形书写格式。

(3) {}对齐。

(4) 足够的注释。

1.2.3 C语言的字符集

字符是组成语言的最基本的元素。C语言字符集由字母、数字、空格、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

注意: C语言严格区分大小写字母,即大写ABC和小写abc具有不同的含义。

空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用,在其他地方出现时,只起间隔作用,编译程序时对它们忽略不计。因此在程序中使用空白符与否,对程序的编译不产生影响,但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

1.2.4 C 语言的词汇

在 C 语言中使用的词汇分为六类：标识符、关键字、运算符、分隔符、常量、注释符。

1. 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外,其余都由用户自定义。C 语言规定,标识符只能是字母 (A ~ Z, a ~ z)、数字 (0 ~ 9)、下划线 (_) 组成的字符串,并且其第一个字符必须是字母或下划线。

以下标识符是合法的:

a, x, x3, BOOK_1, sum5

以下标识符是非法的:

3s	以数字开头
s*T	出现非法字符 *
-3x	以减号开头
bowy-1	出现非法字符 - (减号)

在使用标识符时还必须注意以下几点。

(1) 标准 C 语言不限制标识符的长度,但它受各种版本的 C 语言编译系统限制,同时也受到具体机器的限制。例如,在某版本 C 语言中规定标识符前八位有效,当两个标识符前八位相同时,则被认为是同一个标识符。

(2) 在标识符中,大小写是有区别的。例如, BOOK 和 book 是两个不同的标识符。

(3) 标识符虽然可由程序员随意定义,但标识符是用于标识某个量的符号,因此,命名标识符时应尽量使其具有相应的意义,做到顾名思义,以便于阅读和理解。

2. 关键字

关键字是由 C 语言规定的具有特定意义的字符串,通常也称为保留字。用户定义的标识符不应与关键字相同。C 语言的关键字分为以下几类。

①类型说明符。类型说明符用于定义、说明变量、函数或其他数据结构的类型。如前面例子中用到的 int、double 等。

②语句定义符。语句定义符用于表示一个语句的功能。如 if 语句中的 if。

③预处理命令字符。预处理命令字符用于表示一个预处理命令。如前面例子中用到的 include。

3. 运算符

C 语言中含有相当丰富的运算符。运算符由一个或多个字符组成。

4. 分隔符

在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中分隔各个变量。空格多用于语句各单词之间,作间隔符。在关键字和标识符之间必须要有一个以上的空格符作间隔,例如,把 “int a;” 写成 “inta;” 时, C 编译器会把 inta 当成一个标识

符处理,其结果必然出错。

5. 常量

C语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符常量等。

6. 注释符

C语言的注释符是以“/*”开头并以“*/”结尾。在“/*”和“*/”之间的内容即为注释。程序编译时,不对注释作任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序时,对暂不使用的语句也可用注释符括起来,使编译程序跳过不作处理,待调试结束后再去掉注释符。

1.2.5 C语言的语句

与其他高级语言一样,C语言也是利用函数体中的可执行语句,向计算机系统发出操作命令。按照语句功能或构成的不同,可将C语言的语句分为如下五类。

1. 控制语句

控制语句完成一定的控制功能。C语言只有9条控制语句,这9条控制语句又可细分为如下三种。

(1) 选择结构控制语句。

`if() ~ else ~, switch() ~`

(2) 循环结构控制语句。

`do ~ while(), for() ~, while() ~, break, continue`

(3) 其他控制语句。

`goto, return`

2. 函数调用语句

函数调用语句由一次函数调用加一个分号(语句结束标志)构成。例如:

`printf("This is a C function statement.");`

3. 表达式语句

表达式语句由表达式后加一个分号构成。最典型的表达式语句是在赋值表达式后加一个分号构成的赋值语句。

例如,“`num=5`”是一个赋值表达式,而“`num=5;`”却是一个赋值语句。

4. 空语句

空语句仅由一个分号构成。空语句什么操作也不执行。

5. 复合语句

复合语句是由大括号括起来的一组(也可以是1条)语句构成的语句。例如: