C 语言程序设计

主　编　韩光柱
副主编　许桂芬　齐艳波　吴　华　李桂秋

哈尔滨地图出版社

C语言程序设计

C YUYAN CHENGXU SHEJI

主　编　韩光柱

副主编　许桂芬　齐艳波

吴　华　李桂秋

编　委　褚洪彦　张晓龙

赵迎军

哈尔滨地图出版社

·哈尔滨·

图书在版编目（CIP）数据

C语言程序设计/韩光柱主编. —哈尔滨：哈尔滨地图出版社，2004. 1
ISBN 7-80529-725-8

Ⅰ. C... Ⅱ. 韩... Ⅲ. C语言-程序设计
Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2004）第 002928 号

哈尔滨地图出版社出版、发行
（地址：哈尔滨市南岗区测绘路 2 号 邮政编码：150086）
黑龙江省教育厅印刷厂印刷
开本：787 mm×1 092 mm 1/16 印张：15.375 字数：384 千字
2004 年 1 月第 1 版 2004 年 1 月第 1 次印刷
印数：1～1 000 定价：28.00 元

前　言

C语言是近年来国内外广泛流行并得到迅速推广使用的一种计算机程序设计语言。C语言功能丰富、表达能力强、使用灵活方便，目前高等院校计算机专业和其他相关专业已普遍开设了C语言课程。在国家教委考试中心颁布的计算机等级考试和各省、市组织的计算机考试中，C语言是一个重要的语种。

本书重点放在培养读者良好的程序设计风格和习惯，力求法做到科学性、实用性和通俗性的统一。根据作者数年C语言程序设计方面的教学经验，对读者易于混淆的概念做了重点说明，采用循序渐进，简明扼要、重点突出的编写方式。使读者逐步掌握C语言的基本知识。为便于读者自学，每章都配有精选的例题和习题，供读者选用。

本书共12章：第一章程序设计C语言概述，第二章数据类型、运算符与表达式，第三章最简单的C程序设计—顺序程序设计，第四章选择结构，第五章循环控制，第六章数组，第七章函数，第八章指针，第九章预处理命令，第十章结构体与共用体，第十一章位运算，第十二章文件。基本覆盖了教育部考试中心制定的计算机等级考试（二级——C语言程序设计）考试要求。

学习计算机语言的目的是利用它来编写程序解决实际中出现的问题，因此，希望读者不应把主要精力花费的死记硬背语言的规则上，而应把重点放在程序设计上，即学会使用C语言去进行程序设计。

书中所有程序均在微型计算机的Turbo C环境下调试运行通过。

参加本书编写的有：韩光柱（第一章、第九章）、许桂芬（第二章）、齐艳波（第三章、第十一章）、赵迎军（第十二章）均为伊春职业学院老师；吴华（第四章、第五章）、李桂秋（第六章、第七章）、褚洪彦（第八章）、张晓龙（第十章）等均为黑龙江农业职业技术学院老师。

由于编者水平有限，不妥之处在所难免，恳请读者批评指正。

编者

2003.12.1

内 容 提 要

C语言是一种结构化的计算机程序设计语言，它既具有高级语言的特点，又具有低级语言的功能。本书是学习C语言程序设计的基础，采取循序渐进的内容安排，通俗易懂的讲解方法，并辅以大量便于说明问题的例题，使凡学过一门高级语言的读者都能通过学习本书掌握C语言的基本内容，并会应用它编写程序。

本书主要内容包括：C语言的各种数据类型和运算符，各种表达式，语句结构，函数及库函数，指针，数组，字符串，变量的作用域及存储类，结构体及共用体，文件等。本书内容精炼，结构合理，便于自学，对读者可能遇到的难点做了十分系统、清楚和详细的阐述，极大地减少了读者学习C语言的困难。

本书可作为大专院校计算机专业和非计算机专业的教材，也可供计算机培训班或其他自学者使用。

目　录

第一章　程序设计C语言概述

本章分两部分：第一部分主要介绍了程序设计的基本概念，包括什么是程序、什么是计算机语言、如何进行程序设计以及程序与算法之间有什么关系等，弄懂这些问题有助于从宏观上掌握程序设计的一般思路；另外，我们还介绍了在程序设计和算法表达时都要用到的“流程图”。本章的第二部分主要介绍了C语言的结构及特点，包括C语言的风格、C语言的元素及其构成，程序的编辑、编译、调试和运行等。许多内容都是程序设计中的普遍问题，也是学习C语言的基础。

第一节　程序设计概述

一、程序的概念与计算机的工作原理

计算机是一个能高速运算、具有存储与记忆能力、用程序控制的电子装置。计算机在工作时，只有当操作者向计算机输入一定的信息（这种信息必须是计算机能接受的）时，它才能按照操作者的要求进行工作并且得到所需的结果。目前计算机的基本运行机制具有如下特点：

1. 将需要计算机完成的任务编成一条一条的指令，输入计算机，存放在计算机的内存储器中（称为“编程序”）。

2. 计算机工作时，从内存中取出指令，然后再执行它们（称为“运行程序”）。

计算机通过一条一条的指令来完成工作，人们以一条一条的指令来控制计算机，使它按照人们的要求工作。用计算机术语来说，指令的序列被称为程序。程序是人们意志的体现，它表明了要计算机“做什么”和“怎么做”；同时，程序又是计算机处理问题的灵魂，只有当操作者向计算机输入一定的程序，计算机才能按照程序规定的步骤工作。失去了程序的控制，计算机便无法发挥其作用。

二、计算机语言

程序是计算机指令的集合。我们做一件事时，总要按照一定的步骤一步一步地进行。计算机也是一样的，执行一个程序时总是一条指令一条指令地执行。例如，要求两个数A，B之和S，就可以按照以下的“指令”来进行。

1. 输入A；

2. 输入B；

3. 求和S＝A＋B；

4. 输出S的值。

那么，怎样把以上1～4所确定的步骤以一种计算机能够接受的形式输入计算机呢？这就必须解决下述问题。

首先，计算机并不认识汉字或者其他的自然语言，如英语、法语等，若强行把以上1～4以某种自然语言形式输入计算机，那么计算机就无法“执行”这样的“程序”。其次，自然语言的意义往往和具体的语言环境有关。一个句子、一个词语在不同的场合的含义往往不同，具有“二义性”，而计算机不能容忍这种“二义性”，因为它根本不具备智能，只能机械地、死板地按照你的规定步骤去做。如果你的程序中存在着“二义性”，计算机将不知所措。

正是因为以上的原因，我们期待着一种能够准确无误地表达我们所编制程序含义的、能够

被计算机和人们所接受的、相当严谨且不具有二义性的表达方法的出现。这种表达方法正是我们所说的“计算机语言”。“计算机语言”是人与计算机之间进行通信的工具，是一种计算机能够接受的信息，它由一些简单的单词、符号、数字和严谨的语法组成，能够准确无误地表达要完成的具体操作。它是专门用于人们与计算机之间信息交流的一种特殊语言，在人与计算机之间建立起了一座信息的桥梁。目前计算机语言种类很多，总的来说，它可以分成机器语言、汇编语言和高级语言三大类。

三、程序设计的基本步骤与任务

人们通过程序让计算机工作起来，以便处理各种信息，解决各种问题，让计算机为人类服务，所以“程序”是人们意志的体现。反过来说，要让计算机按自己的“意志”进行工作，就必须编制程序。编制程序就是用户通过程序设计语言对信息进行加工、处理并输出预期结果的方法。编制程序是本书的核心内容，下面简要地说明编制程序的过程。

1. 分析问题：这是编程序的第一步，因为任何程序都是为了解决一定的实际问题。编程序时不能无的放矢，而是要认真考虑实际问题，找出解决问题的基本思路。

2. 提出算法：把第 1 步中的分析问题的思路进一步明确化、详细化，建立解题需要的数学或物理模型，为下一步用计算机语言来表达这些方法奠定基础。

3. 编写程序：根据第二步的方案用某种计算机语言把程序写出来。

4. 上机调试：对编好的程序进行实际检验，发现其中的错误之处，不断加以改正，直到程序能达到预期的目的。

5. 运行程序：将程序投入运行，并输出结果。程序设计的基本步骤如图 1－1 所示。

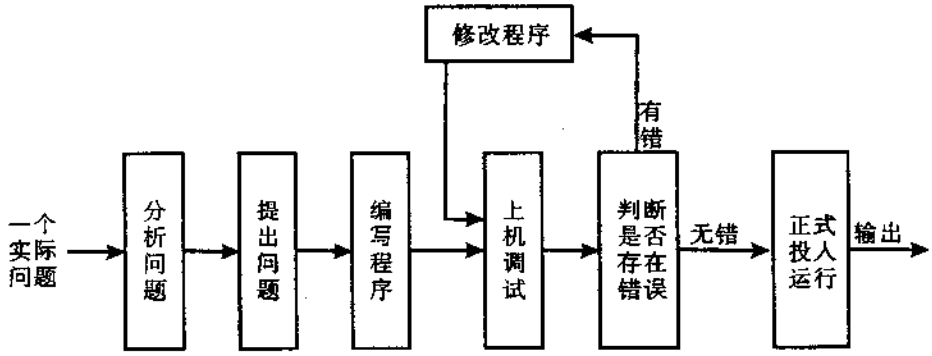

图 1－1　程序设计的基本步骤

第二节　算法与流程图

在系统地学习 C 语言程序设计知识之前，先介绍算法与流程图，它们是进行程序设计的重要工具。

一、算法

算法就是对问题求解方法的精确描述。在进行程序设计时，最关键的问题是算法的提出。因为它直接关系到编写出来的程序的正确性、可靠性。如果没有认真地研究实际的问题就草率地提出一些不成熟的算法，那么编写出来的程序就可能出现错误或疏忽。算法作为对解题步骤的精确描述，应具备如下性质。

1.有穷性：一个算法必须在有限步骤之后结束，而不能无限制地进行下去。因此，在算法中必须给出一个结束的条件。我们不能指望计算机算出圆周率的准确值。因为 π 是一个无穷不循环小数，无法求出它的准确值。

2.明确性：一个算法中的任何步骤都必须意义明确，不能模棱两可、含混不清，即不允许有二义性，不能在计算机中使用诸如"老张对老李说：他的儿子考上了清华大学"这种有歧义的表达方法，到底是老张的儿子上了清华大学还是老李的儿子上了清华大学？无法确知。

3.可执行性：所采用的算法必须能够在计算机上执行，因此，在算法中所有的运算必须是计算机能够执行的基本运算。要计算机执行的步骤，计算机应该能够实现，不能提出像"让计算机去煮饭，煮完饭之后再炒菜"之类的算法，至少它在目前无法实现。

4.有一定的输入与输出：要计算机解决问题时，总是需要输入一些原始的数据；计算机向用户报告结果时，总是要输出一些信息。因此，一个算法中必须有一定的输入与输出。以上是算法的基本性质，在此基础上就可以学习描述算法的基本工具——流程图。

二、流程图

流程图是一种能够比较形象地描述"算法"的工具，它对编制程序很有帮助。流程图又称为框图，是由几种不同的图形组合而成的，如图1－2所示。

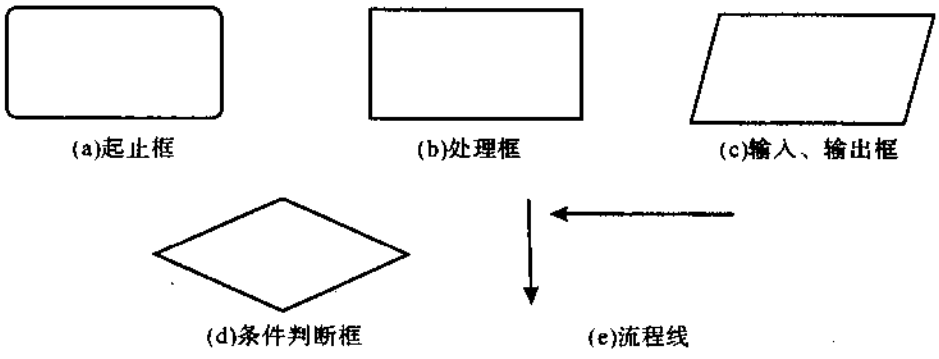

图1－2　组成流程图的基本图形

1.起止框：如图1－2(a)，它代表一个算法的开始与结束之处。

2.处理框：如图1－2(b)，它可以代表算法中的一个或若干个步骤，这些步骤不涉及输入与输出。

3.输入输出框：如图1－2(c)，它表示一个算法中需要进行输入或输出处理的步骤。为了与一般的处理步骤区别开，输入输出框采用了平行四边形的形式。

4.条件判断框：如图1－2(d)，当一个算法中需要依据某一条件来决定后续操作时，就采用此框。

5.流程线：图1－2(e)，它表明每一步骤之间的先后顺序，标识着一个算法的走向。

例1.用流程图来描述如下算法：向计算机输入一个数X，若X≥0；则显示X的值。

解：流程图如图1－3、图1－4所示。开始时先遇到起始框，表示算法的开始，之后随着箭头指向输入框，要求你从键盘上向计算机输入一个数X。再向下，遇到判断框，判断"X≥0"这个条件是否成立，若"X≥0"成立，则打印出X的值。否则，沿着标有"不成立"的那条流程线到达终止框，该算法结束。

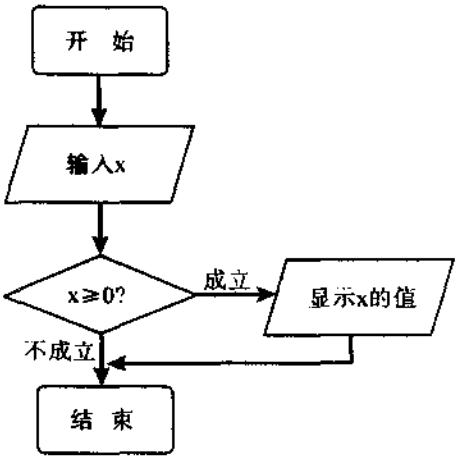

图 1－3 中文流程图

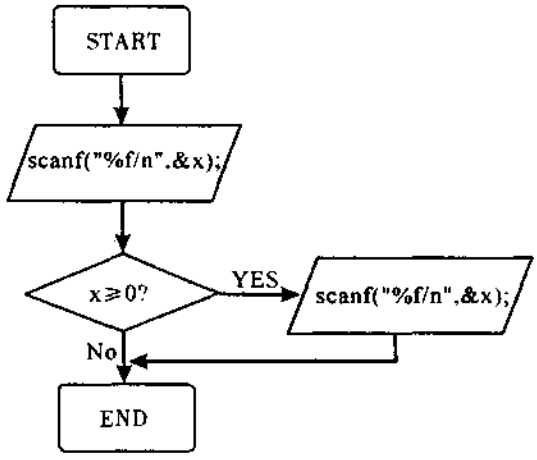

图 1－4 英文流程图

第三节 语言的识别与程序的执行

前面叙述仅说明了计算机的程序要通过一定的计算机语言来表达，而未说明计算机是如何识别用某种语言编写的程序，如何执行程序，这是本节将要讨论的内容。

一、计算机最终能执行的是机器语言程序

计算机不能识别与执行人类自然语言。计算机内部存储数据和指令采用二进制（“0”和“1”）方式。计算机只能接受和识别由“0”和“1”组成的二进制信息。每一类型的计算机都分别规定了由若干个二进位的信息（即若干个“0”或“1”组成的信息）组成一条指令。例如，某种计算机规定以 1011011000000000 这样的指令作为“加法”指令，遇到这样的指令，计算机就执行一次加法。

这种计算机能直接识别和执行的二进制形式的指令称为“机器指令”。例如，前面介绍的 1011011000000000 就是一条机器指令。一条机器指令产生一个相应的机器操作。每一种计算机确定有若干种指令（例如加法指令、减法指令、传送指令、取数指令、存数指令、输出指令……）以实现不同的操作。一种计算机的指令的集合称为该计算机的机器语言（machine language），或者说成是该计算机的指令。正如同用算盘算题一样，每一条珠算口诀就是一条“指令”，算盘全部口诀之和就是“珠算语言”。也就是说，“语言”是全部指令的总和。人们为了解决某一问题，可以从该“语言”中选择所需的指令，组成一个指令序列，这个指令序列称为“机器语言程序”。用机器语言编写出的程序，计算机能直接识别和执行，执行效率比较高；但是，难学、难记、难写、难检查、难调试、难以推广也是它显而易见的缺点。

二、C 语言是一种高级语言

为了弥补机器语言的上述缺点，人们创造了一种各类计算机都通用的接近于人类“自然语言”和“数学语言”的程序设计语言。譬如可以写出下面一条 C 语言语句：

```
printf("%d\n",sin(x+y)+cos(x-y)+3.14);
```

其中，“printf”是一个 C 语言的函数，功能是“按双引号指定的格式打印”。“sin(x＋y)＋cos(x－y)＋3 14”是一个数学式子，它的数学含义是“分别计算(x＋y)的正弦值和(x－y)的

余弦值，指导它们相加之后再加3.14"。以上是一条接近自然语言（英文）和数学语言的指令。如果计算机能识别这样的指令，将为使用者提供极大的方便。

这种人工创造的语言称为"高级语言（high－level－language）"，相对而言，机器语言称为"低级语言（low－level language）"。所谓"低级"，指它直接贴近机器，"高级"指离机器远一些，不是直接面向机器的。高级语言在各种计算机中都通用。

三、翻译程序

计算机不能直接识别高级语言，需要有一个"翻译"，把用高级语言编写的程序翻译成用二进制形式表示的机器语言程序（即由若干条机器指令组成的指令序列），如图1－5所示。这个"翻译"工作由一个计算机软件实现。人们在"创造"一种语言的同时，必须设计出一个翻译系统。高级语言程序在运行之前，都要通过这个翻译系统逐条被翻译成机器语言程序，然后再由计算机执行。

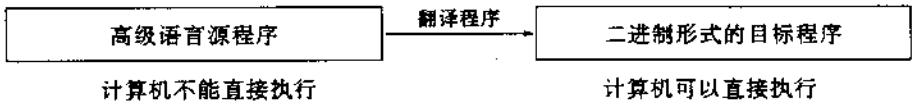

图1－5　翻译程序

高级语言程序称为"源程序"，翻译后得到的机器语言程序称为"目标程序"（或"目的程序"）。目前，国内外使用的高级语言种类很多，不下几百种，最常用的也有十几种，它们的运用的范围各不相同。每一种高级语言都有自己相应的翻译程序，翻译程序属于系统软件范畴。

四、翻译程序的分类

目前的翻译程序有解释与编译两种方式。

1.编译（compile）方式：用一个称为"编译程序"的软件进行编译工作。编译过程包括翻译和查错两个功能。还包括词法分析、语法和语义分析、生成目标程序、优化目标程序。如果发现语法有错，就报告出错信息，不生成目标程序，这时用户必须修改程序，再次进行编译。如还有错，还要修改和编译，直到不出现语法错误为止，此时便生成目标程序（见图1－6）。

2.解释（interpret）方式：解释也是将高级语言源程序翻译为机器指令的一种方式，但它与编译方式不同，不是把整个高级语言源程序一起翻译成一个目标程序。而是翻译一句，执行一句，不产生整个的目标程序。如果程序没有错误，则一直进行到全部执行完毕。如果在运行过程中发现程序有错，则马上中止"解释"工作，然后修改程序后再重新运行（见图1－7）。

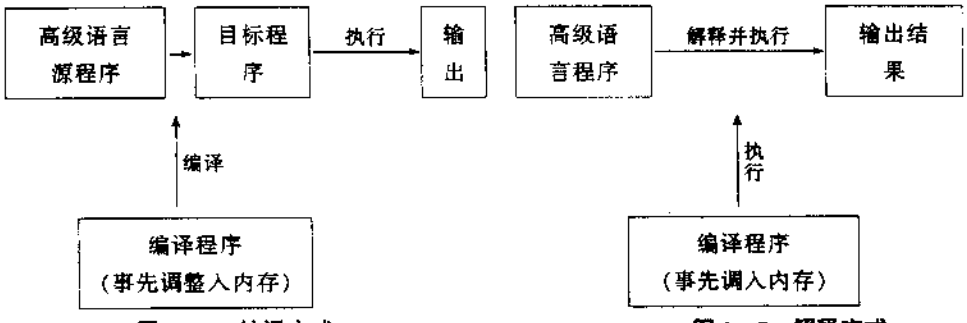

图1－6　编译方式

图1－7　解释方式

3.解释与编译的区别：编译方式相当于笔译，得到一篇完整的译文；解释方式相当于口译，说一句译一句，发现说得不对就停止翻译并且前功尽弃。解释方式使用方便，便于逐条调试语句，但占机器时间多，效率较低。编译方式得到的目标程序经过优化，执行效率高，但占内存

多，使用不大方便。如果一处程序要反复多次运行，则用解释方式很不经济，因为每次都要重新进行翻译工作。而用编译方式能得到一个可以保存的目标程序，且编译完后再次执行程序时，编译系统可以撤出内存，只需直接运行该目标程序即可，不必再重新编译，也就是"一次编译，多次运行"（当然，如果源程序有修改，就需要重新编译以便得到一个新的目标程序）。编译或解释工作都是在操作系统的管理下进行的。现在，绝大多数语言的翻译工作都采用编译方式。早期的 C 语言采用编译方式（例如 Turbo C），后来也有采用解释方式运行的 C 语言。

第四节　C 语言的风格和特点

一、C 语言的概述

C 语言最早诞生于 1972 年，它的发展是同著名的 UNIX 操作系统紧密联系在一起的。1977 年，美国电话电报公司（AT&T）贝尔实验室正式发表了 C 语言，出现了独立于机器的 C 语言编译文本《可移植 C 语言编译程序》。同时，B. W. Kernighan 和 D. M. Ritchie 合著了著名的《The C Programming Language》一书，通常简称为《K&R》。《K&R》并没有定义一个完整的标准 C 语言，但却成为事实上的标准。在这之后，美国国家标准协会 ANSI 制定了 C 语言标准，于 1983 年发表，通常称之为 ANSCI C。国际标准化组织 ISO 也制定了 C 语言的国际标准，称为 ISO 标准，由于该标准的某些改进优于 ANSI 标准，所以 ANSI 也接受这个国际版本，因此统称为 ANSI/ISO C 标准，这就是我们今天看到的 C 语言。

C 语言是当今最优秀的程序设计语言之一，早期主要应用于 UNIX 操作系统，UNIX 是一个多用户分时操作系统，除了一个由汇编语言编写的被其他程序频繁调用的核心外，其他程序都是用 C 语言编写的。UNIX 操作系统具有简洁、通用和方便等优点，在各类大、中、小和微型计算机上得到了非常广泛的应用，这对促进 C 语言的推广应用起了很大的作用。同其他高级程序设计语言，如 BASIC，FORTRAN 及 PASCAL 相比，C 语言具有以下特点：

1. C 语言是一种结构化语言，它主要由一些功能相对独立的函数构成。用 C 语言编写出来的程序层次清晰，便于按模块化方式组织，易于调试和维护。

2. C 语言具有丰富的运算符和数据类型，表现能力和处理能力极强。它不仅可以实现各类复杂的数据结构，还允许几乎所有的数据类型进行转换，这些均给编程人员提供了极大的方便。

3. C 语言可以对硬件进行编程操作，可以通过指针直接访问内存的物理地址，进行位一级的操作，它集高级语言和汇编语言的功能于一体，既可用于应用软件的开发，也适合于系统软件的开发，这是它最突出的优点。

此外，C 语言还具有生成代码效率高、可移植性强等优点，因此被广泛地移植到了各种类型计算机上，形成了多种 C 语言版本。

目前在微机上以 MS－DOS 操作系统为平台的 C 语言主要分为以下两种：Microsoft C 或称 MS C，Borland Turbo C 或称 Turbo C。Turbo C 是 Borland 公司开发的一个以 MS－DOS 作为操作系统，用于微机上的 C 编译系统，它具有友好的用户界面和丰富的标准库函数。全国计算机等级考试——二级 C 语言上机考试系统以 MS－DOS5.0 和 Turbo C2.0 为软件考试环境，以下主要结合 Turbo C 介绍开发 C 程序的过程。

二、C 程序的结构特点

程序结构是指程序的组织形式。程序结构乃是程序的骨架，程序本身可以改变，但是程序

的组织形式不能随意变动。就像一本书，总由封面、扉页、目录、正文和封底等组成一样，那么，C程序是怎样组织的呢？

先举一个简单的程序实例，看看C程序是什么样子。从实例开始介绍C程序的一些基本要素，然后从具体到一般，给出C程序的一般组织形式。

【例1-1】从键盘输入一个整数，以此整数为半径求圆面积，并将结果显示在屏幕上。

```
1: # include<stdio.h>          /*用预处理命令指定一个包含文件*/
2: # define PI 3.1416          /*替换命令*/
3: main()                      /*主函数*/
4: {                           /*main()函数体开始标记*/
5: float are();                /*函数说明*/
6: int r;                      /*整型内部变量说明*/
7: float a;                    /*实型内部变量说明*/
8: scanf("%d",&r);             /*调用scanf库函数接受键盘输入*/
9: a=area(r);                  /*调用area函数，返回值赋给a*/
10: printf("s=%f\n",a)         /*调用printf库函数，结果*/
11: }                          /*main()函数体结束标记*/
12: float area(x)              /*求圆面积函数定义*/
13: int x;                     /*函数形式参数说明*/
14: {                          /*area()函数体开始标记*/
15: float y;                   /* 实型内部变量说明*/
16: y=PI*x*x;                  /*求圆面积*/
17: return(y);                 /*返回圆面积值*/
18: }                          /*area()函数体结束标记*/
```

以上是一个简单的然而却是完整的C语言程序，示例程序中共有18行，为了叙述方便，每行都已编上序号，但千万要注意，这些行的序号并非C语言所要求，C语言是不使用行序号的。另外，每一命令行右边是注释部分，注释部分不参加编译，仅起注解作用。要注意这两点。

下面列出C语言程序的主要结构特点：

(1)所有的C语言程序都是由一个或多个叫做"函数"的程序模块构成的。例如，上述程序由两个函数组成：第3～11行是一个函数；第12～18行是另一个函数。如果程序较大，可能由更多的函数组成，所以C语言程序从总体上看，它是函数的集合体，如图1-8所示。

(2)在多个函数中，必须有一个而且只能有一个叫做main()的函数(例如第3行)。在英文中"main"是"主要的"意思，所以称它为主函数(相当于其他高级语言的主程序)；除主函数外，其他函数(相当于其他高级语言子程序)用户可以自己起名。例如，第12行的ares()函数就是编程者自己取的名字。函数名要取得有意义，应有"见名识义"之功效。

(3)各个函数在程序中的位置并不十分重要，不是说主函数main()一定要放在其他函数之前，也可以放在别的位置，但程序总是从main函数的第一条语句开始执行。函数主要是由函数名及其后面的一对大括号及包含的若干语句组成的。函数有所谓"有参函数"和"无参函数"之分，如果是无参函数，那么函数名后的"形参"下一行的"形参说明"部分自然就不存在了。但是，任何情况下，函数名后的一对圆括号都不能省略。不妨对示例中的第12～18行构成的求圆面积的有参函数进行考察。看看它是否与图1-9的函数的一般格式相符。

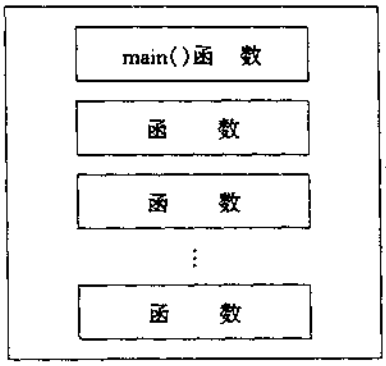

图 1-8 C语言程序是函数的集合体序

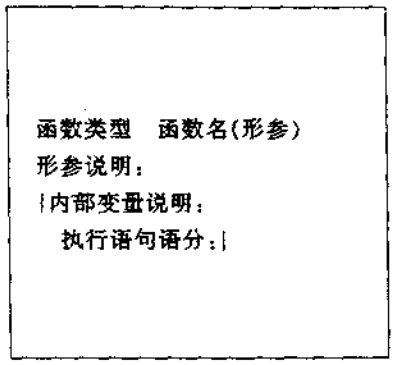

图 1-9 C语言函数的一般格式

第 12 行: float area (x) ——定义了一个名为 area 的实型有参函数

函数类型 函数名 形参

第 13 行:int x; ——说明形参 x 是整型变量

第 14 行:{ ——函数体开始标记

第 15 行:float y; ——内部变量说明

第 16 行:y=PI * x * x; ——执行语句

第 17 行:retrun(y); ——执行语句

第 18 行:} ——函数体结束标记

C 语言引入函数这一概念,是与数学中的函数概念相类似的。

在数学中,函数 $y=f(x)$,对应于自变量 x 的每一个给定值,总有一个函数值 y 与之相对应。上例中:函数 $y=f(x)=3.1416\times2$(即函数 area()中的 y=PI * x * x)

调用该函数时:

当 x=2,有 y=f(2)=3.1416·22=12.5664

当 X=3,有 y=f(3)=3.1416·32=28.2774……

第五节 C语言的元素及其构成

构成 C 语言的元素分为字符集、标识集、关键字、运算符、常量、变量和注释符共 7 类。

1. 字符集:某种语言所采用的字母、数字、特殊符号集合了字符集,C 语言中允许表 1-1 所示符号在程序中存在。

2. 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。除标准库函数中的函数名由系统定义外,其余都由用户自己定义。C 语言规定:标识符是由英文字母(A~Z,a~z)、数字(0~9)、下划线____组成的字符串,并且第一个字符必须是字母或下划线。以下标识符是合法的:x, a, x^3y, _parl, sum1。

以下标识符是非法的:3y 以数字开头;x * y 出现非法字符(*);-3X 以非法字符减号开头(-);boy\1 出现非法字符(\)。

使用标识符必须注意以下几点:

(1)标准 C 语言不限制标识符的长度,但它受各种版本 C 语言编译系统的限制,同时也受到具体机器的限制。例如,TURBO C 中标识符的有效长度可以在编译系统中设置,缺省值为

32位。

表1-1　C语言字符集

类别名称	具体符号	总计个数
英文字母(大写)	A,B,C,…X,Y,Z	共26个
英文字母(小写)	a,b,c,…x,y,z	共26个
数字	0,1,2,…,9	共10个
运算符号	+,-,*,/,^	共5个
标点符号	,,",:,;,',?	共7个
空白符号	空格,制表符	共2个
括号	(,),[,],{,}	共6个
关系运算符	>,<,=	共3个
特殊符号	$,%,|,#,\,&,!	共7个

(2)C语言标识符区别大小写,例如INDEX和index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,命名应尽量有一定的意义,便于阅读理解。

3.关键字:关键字是由C语言规定的具有特定意义的字符串,也称为保留字。用户定义的标识符不应与关键字相同。C语言的关键字均为小写,可分成以下几类:

(1)类型说明符:定义和说明变量、函数或其他数据结构的类型。例如:

①数据类型:char,double,float,int,long,short,struct,union,unsigned;②存储方式:auto,extern,static,register;③数据类型定义:typedef。

(2)语句定义符:表示语句的功能。例如:

①选择语句:if,else,switch,case,default;②循环语句:do,while,for;③转移语句:goto;④返回语句:return;⑤其他语句:continue,break。

(3)预处理命令:表示预处理命令。例如:

①宏定义:#define,#undef;②文件包含:#include;③条件编译:#if,#ifedf,#ifndef,#else,#endif。这些字符集的含义如表1-2所示。

表1-2　C语言中的关键字

类别	拼写	意义	拼写	意义	拼写	意义	拼写	意义
ANSI与&R共同规定的	auto	自动	double	双精度	int	整型	struct	结构
	break	中止	else	否则	long	长整型	switch	开关选择
	case	情况	extern	外部的	register	寄存器	typedef	类型定义
	char	字符	float	实型	return	返回	union	联合
	continue	连续	for	对于	short	短整型	unsigned	无符号的
	default	默认	goto	转移	sizeof	尺寸大小	while	当……
	do	做	if	如果	static	静态的	enum	枚举
新增	拼写	意义	拼写	意义	拼写	意义	拼写	意义
	const	常数	signed	有符号的	void	空类型的	volatile	优化

4.运算符:C语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式,表示各种运算功能。运算符由一个或多个字符组成。