

C YUYAN CHENGXU SHEJI JIAOCHENG

C 语言程序设计教程

韩明 主编



电子科技大学出版社
University of Electronic Science and Technology of China Press

C语言程序设计教程

主编 韩 明

 电子科技大学出版社
University of Electronic Science and Technology of China Press

• 成都 •

图书在版编目(CIP)数据

C语言程序设计教程 / 韩明主编. -- 成都: 电子科技大学出版社, 2018.3

ISBN 978-7-5647-5848-6

I. ①C… II. ①韩… III. ①C语言-程序设计-教材
IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第044574号

C语言程序设计教程

韩 明 主编

策划编辑 李述娜

责任编辑 李 倩

出版发行 电子科技大学出版社

成都市一环路东一段159号电子信息产业大厦九楼 邮编 610051

主 页 www.uestcp.com.cn

服务电话 028-83203399

邮购电话 028-83201495

印 刷 定州启航印刷有限公司

成品尺寸 185mm × 260mm

印 张 15

字 数 336千字

版 次 2019年1月第一版

印 次 2019年1月第一次印刷

书 号 ISBN 978-7-5647-5848-6

定 价 57.00元

版权所有, 侵权必究

前 言

C 语言已经有较长的历史了，很多语言都是在 C 语言基础上发展起来的。因此，C 语言的学习十分重要。通过学习 C 语言可以了解面向过程的编程思想并学会把实际问题进行建立模型、拆分；使用 C 语言要面对许多底层的東西，如指针、字符串等，对使用更高级的语言很有帮助。C 语言功能丰富，表达力强，使用灵活方便，应用面广，目标程序效率高，可移植性好，特别适合来编写软件程序。C 语言诞生后，许多用汇编语言编写的软件，都可以用 C 语言进行编写，而学习 C 语言要比学习汇编语言简单得多。Turbo C、Visual C++ 是 C 语言常用的开发环境，Turbo C 最新版本为 Turbo C 3.0，Visual C++ 最新版本为 Visual C++ 2017，读者根据需求选择 C 语言开发环境。本书选择以 Turbo C 3.0 开发环境讲解 C 语言。

全书遵循程序设计与案例教学的基本思想，以应用为导向，以实用为原则，以能力提升为目标，以典型代码、经典实例、完整案例为依托，既利于程序设计基本思想与方法以及 C 语言编程技术的掌握，又利于计算思维与问题求解能力的培养。本书共分为 11 章，介绍 C 语言程序设计的有关技术与相关应用以及结构化程序设计的基本思想与方法，内容包括 C 语言概述、输入/输出函数、编程基础、程序控制结构、数组、函数、指针、结构类型、编译预处理、文件操作、俄罗斯方块游戏，附有大量的各类习题以及相应的实验指导。

本书内容全面，实例翔实，案例丰富，编排合理，循序渐进，语言流畅，通俗易懂，准确严谨，解析到位，注重算法设计与应用开发能力的培养，既可作为高校本科计算机、电子商务、信息管理与信息系统等相关专业程序设计基础、C 语言程序设计等课程的教材或教学参考用书，也可作为 C 语言程序设计人员的技术参考书以及初学者的自学教程。

由于时间仓促、作者水平有限，书中难免有不足和疏漏，欢迎读者提出宝贵意见和建议，以供再版时进行改进。

目 录

第 1 章 C 语言概述 / 001

- 1.1 程序与程序设计语言 / 001
- 1.2 C 语言程序的结构特点 / 003
- 1.3 C 语言程序设计步骤 / 009
- 1.4 综合实例：输出“您好！欢迎学习 C 语言”文字 / 011
- 1.5 本章小结 / 011
- 1.6 本章习题 / 012

第 2 章 输入输出函数 / 013

- 2.1 格式输入/输出函数 / 013
- 2.2 字符输入/输出函数 / 022
- 2.3 综合实例：不同格式的整数输出 / 024
- 2.4 本章小结 / 024
- 2.5 本章习题 / 025

第 3 章 编程基础 / 026

- 3.1 数据类型 / 026
- 3.2 常量和变量 / 027
- 3.3 C 语言的常用数据类型 / 031
- 3.4 不同数据类型间的转换 / 038
- 3.5 运算符和表达式 / 041
- 3.6 位运算 / 052
- 3.7 综合实例：输出平均值和最小值 / 056
- 3.8 本章小结 / 056
- 3.9 本章习题 / 057

第 4 章 程序控制结构 / 058

- 4.1 程序设计概述 / 058
- 4.2 条件语句 / 061
- 4.3 循环语句 / 067

- 4.4 跳转语句 / 073
- 4.6 综合实例：编写程序计算三角形面积 / 076
- 4.7 本章小结 / 077
- 4.8 本章习题 / 077

第 5 章 数组 / 080

- 5.1 数组简介 / 080
- 5.2 一维数组 / 081
- 5.3 多维数组 / 083
- 5.4 字符数组与字符串 / 086
- 5.5 综合实例：按字母排序国家名字 / 091
- 5.6 本章小结 / 092
- 5.7 本章习题 / 093

第 6 章 函数 / 097

- 6.1 函数概述 / 097
- 6.2 函数调用 / 102
- 6.3 局部变量和全局变量 / 109
- 6.4 变量的存储类别 / 112
- 6.5 综合实例：打印图案 / 114
- 6.6 本章小结 / 115
- 6.7 本章习题 / 115

第 7 章 指针 / 117

- 7.1 地址指针 / 117
- 7.2 变量指针 / 118
- 7.3 数组指针 / 128
- 7.4 字符串指针 / 136
- 7.5 函数指针与指针型函数 / 138
- 7.6 指针数组 / 141
- 7.7 综合实例：建立多项式链表 / 146
- 7.8 本章小结 / 150
- 7.9 本章习题 / 150

第 8 章 结构类型 / 154

- 8.1 结构体 / 154

8.2	结构指针	/	161
8.3	联合	/	167
8.4	枚举	/	169
8.5	综合实例：编写输入人员数据，显示出表格	/	172
8.6	本章小结	/	173
8.7	本章习题	/	173
第9章 编译预处理 / 175			
9.1	宏定义	/	175
9.2	文件包含与条件编译	/	179
9.3	综合实例：用带参的宏求两个数值的余数	/	181
9.4	本章小结	/	181
9.5	本章习题	/	182
第10章 文件操作 / 183			
10.1	文件	/	183
10.2	文件的打开与关闭	/	185
10.3	文件的读写	/	187
10.4	文件的随机读写	/	190
10.5	文件检测函数	/	192
10.6	综合实例：统计文件中字母出现次数	/	194
10.7	本章小结	/	195
10.8	本章习题	/	195
第11章 俄罗斯方块游戏 / 197			
11.1	游戏概述	/	197
11.2	游戏的概要设计	/	198
11.3	游戏程序设计	/	199
11.4	本章小结	/	209
11.5	本章习题	/	209
习题参考答案 / 210			
参考文献 / 232			

第 1 章 C 语言概述

学习目标



1. 掌握程序与程序设计语言的特点。
2. 掌握 C 语言程序的结构特点。
3. 掌握 C 语言程序的设计步骤。
4. 认识 C 语言开发环境。

学习难点



1. 掌握 C 源程序结构。
2. 掌握 C 语言字符集。
3. 掌握 C 语言程序的编译、连接与运行。

1.1 程序与程序设计语言

Turbo C、Visual C++ 是 C 语言常用的开发环境，Turbo C 最新版本为 Turbo C 3.0，Visual C++ 最新版本为 Visual C++ 2017，读者根据需求选择 C 语言开发环境。本书选择以 Turbo C3.0 开发环境讲解 C 语言。在计算机行业，程序是指编程语言按照流程执行的过程。程序由程序设计语言组合而成，例如，C 语言、Java 等。本节介绍程序以及程序设计语言的基本概念。

1.1.1 程序简介

程序 (program) 是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合。程序是由序列组成的, 用于告诉计算机如何完成一个具体的任务。可以将程序看作为实现预期目的而进行操作的一系列语句和指令, 一般分为系统程序和应用程序两大类, 计算机中的程序在港澳台地区称为程式。程序就是为使电子计算机执行一个或多个操作, 或执行某一任务, 按序设计的计算机指令的集合。由程序计数器 (program counter) 控制。程序 (program) 是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合, 是用汇编语言、高级语言等开发语言编制出来的可以运行的文件, 在计算机中称可执行文件 (后缀名一般为 .exe)。



提示:

现在我们所玩的游戏就是应用程序 (现在后缀名多为 .swf 的 Flash 影片类的游戏也较流行)。

1.1.2 程序设计语言

程序设计语言, 通常简称为编程语言, 是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧, 用来向计算机发出指令。一种计算机语言让程序员能够准确地定义计算机所需要使用的数据, 并精确地定义在不同情况下所应当采取的行动。在过去的几十年间, 大量的程序设计语言被发明、被取代、被修改或组合在一起。尽管人们多次试图创造一种通用的程序设计语言, 却没有一次尝试是成功的。之所以有那么多种不同的编程语言存在的原因是, 编写程序的初衷各不相同; 新手与老手之间技术的差距非常大, 而有许多语言对新手来说太难学; 其次, 不同程序之间的运行成本 (runtime cost) 各不相同。有许多用于特殊用途的语言, 只在特殊情况下使用。例如, PHP 专门用来显示网页; Perl 更适合文本处理; C 语言被广泛用于操作系统和编译器 (所谓的系统编程) 的开发。高级程序设计语言 (也称高级语言) 的出现使得计算机程序设计语言不再过度地倚赖某种特定的机器或环境。这是因为高级语言在不同的平台上会被编译成不同的机器语言, 而不是直接被机器执行。最早出现的编程语言之一 FORTRAN 的一个主要目标, 就是为了实现平台独立。虽然大多数的语言可以既可被编译 (compiled) 又可被解译 (interpreted), 但大多数只在一种情况下能够良好运行。在一些编程系统中, 程序要经过几个阶段的编译, 一般而言, 后阶段的编译往往更接近机器语言。这种常用的使用技巧最早在 1960 年用于 BCPL, 编译程序先编译一个“0 代码”的转换程序 (representation), 然后再使用虚拟机转换到可以运行于机器上的真实代码。这种成功的技巧之后又用于 Pascal 和 Pcode 以及 Smalltalk 和二进制码, 虽然在很多时候, 中间过渡的代码往往是解译, 而不是编译的。如果所使用的翻译的机制是将所要翻译的程序代码作为一个整体翻译, 并之后运行内部格式, 那么这个翻译过程就称为编译。因此, 一个编

译器是一个将可阅读的程序文本(源代码)作为输入的数据,然后输出可执行文件(object code)。所输出的可执行文件可以是机器语言,由计算机的中央处理器直接运行,或者是某种模拟器的二进制代码。

提示:

如果程序代码是在运行时才即时翻译,那么这种翻译机制就被称作解译。经解译的程序运行速度往往比编译的程序慢,但往往更具灵活性,因为它们能够与执行环境互相作用。

1.2 C语言程序的结构特点

C语言是世界上广泛流行的计算机高级程序设计语言,它于1978年由美国贝尔实验室设计发布。由于C语言同时具备高级语言的优点和低级语言的功能,而且拥有很好的可移植性,使它成为程序员最喜欢的编程语言之一。它以功能强大、数据结构丰富、目标代码质量高、程序运行效率高、可移植性好等特点成为众多程序员学习程序设计的首选语言。

1.2.1 C语言简介

20世纪70年代,C语言问世。1978年由美国电话电报公司(AT&T)贝尔实验室正式发表了C语言。同时由B. W. Kernighan和D. M. Ritchie合著了“THE C PROGRAMMING LANGUAGE”(简称《K&R》或《K&R》标准)。但是,在《K&R》中并没有定义一个完整的标准C语言,后来由美国国家标准学会在此基础上制定了一个C语言标准,于1983年发表,通常称之为ANSI C。

早期的C语言主要是用于UNIX系统。由于C语言的强大功能和各方面的优点逐渐为人们认识,到了20世纪80年代,C开始进入Windows、Linux操作系统,并很快在各类大、中、小和微型计算机上得到了广泛的使用,成为当代最优秀的程序设计语言之一。

C语言是一种结构化语言。它层次清晰,便于按模块化方式组织程序,易于调试和维护。C语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型,便于实现各类复杂的数据结构。它还可以直接访问内存的物理地址,进行位(bit)一级的操作。由于C语言实现了对硬件的编程操作,因此C语言集高级语言和低级语言的功能于一体。既可用于系统软件的开发,也适合于应用软件的开发。此外,C语言还具有效率高,可移植性强等特点。因此广泛地移植到了各类各型计算机上,从而形成了多种版本的C语言。

提示:

目前,C语言有以下几种:① Microsoft C或称MS C;② Borland Turbo C或称Turbo C;③ 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做了一些扩充,使之更加方便、完美。

在 C 的基础上，1983 年又由贝尔实验室的 Bjarne Stroustrup 推出了 C++。C++ 进一步扩充和完善了 C 语言，成为一种面向对象的程序设计语言。C++ 目前流行的版本是 Borland C++、Symantec C++、Microsoft Visual C++。C++ 提出了一些更为深入的概念，它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间，为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。因而也增加了整个语言的复杂性，掌握起来有一定难度。C 是 C++ 的基础，C++ 语言和 C 语言在很多方面是兼容的。因此，掌握了 C 语言，再进一步学习 C++ 就能以一种熟悉的语法来学习面向对象的语言，从而达到事半功倍的目的。

1.2.2 C 源程序的结构

C 源程序的结构一般由以下几部分组成：

- (1) 一个 C 语言源程序可以由一个或多个源文件组成。
- (2) 每个源文件可由一个或多个函数组成。
- (3) 一个源程序不论由多少个文件组成，都有一个且只能有一个 main 函数，即主函数。
- (4) 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。
- (5) 每一个说明，每一个语句都必须以分号结尾。但预处理命令，函数头和花括号“}”之后不能加分号。

关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符，也可不再加空格来间隔。

下面用实例说明注释。

【例 1-1】求输入数据的正弦值。

```
#include<math.h> /*include 称为文件包含命令文件 */
#include<stdio.h> /* 扩展名为 .h 的文件也称为头文件或首部 */
main()           /*main 函数开始 */
{
    double x, s; /* 定义两个实数变量，以被后面程序使用 */
    printf("input number:\n"); /* 显示提示信息 */
    scanf("%lf", &x);          /* 从键盘获得一个实数 x */
    s=sin(x);                  /* 求 x 的正弦，并把它赋予变量 s */
    printf("sine of % lf is % lf\n",x,s) /* 显示程序运算结果 */
} /*main 函数结束 */
```

程序的功能是从键盘输入一个数 x，求 x 的正弦值，然后输出结果。在 main() 之前的两行称为预处理命令（详见后面）。预处理命令还有其他几种，这里的 include 称为文件包含命令，其意义是把引号“ ”或尖括号<>内指定的文件包含到本程序来，成为本程序的一部分。

被包含的文件通常是由系统提供的，其扩展名为 .h。因此也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型。因此，凡是在程序中调用一个库函数时，都必须包含该函数原型所在的头文件。在【例 1-1】中，使用了三个库函数：输入函数 scanf，正弦函数 sin，输出函数 printf。sin 函数是数学函数，其头文件为 math.h 文件，因此在程序的主函数前用 include 命令包含了 math.h。scanf 和 printf 是标准输入输出函数，其头文件为 stdio.h，在主函数前也用 include 命令包含了 stdio.h 文件。

提示：

需要说明的是，C 语言规定对 scanf 和 printf 这两个函数可以省去对其头文件的包含命令。所以在本例中也可以删去第二行的包含命令 #include。同样，在【例 1-1】中使用了 printf 函数，也省略了包含命令。

在【例 1-1】中的主函数体中又分为两部分，一部分为说明部分，另一部分为执行部分。说明是指变量的类型说明。例题中未使用任何变量，因此无说明部分。C 语言规定，源程序中所有用到的变量都必须先说明，后使用，否则将会出错。这一点是编译型高级程序设计语言的一个特点，与解释型的 BASIC 语言是不同的。说明部分是 C 源程序结构中很重要的组成部分。本例中使用了两个变量 x，s，用来表示输入的自变量和 sin 函数值。由于 sin 函数要求这两个量必须是双精度浮点型，故用类型说明符 double 来说明这两个变量。说明部分后的四行为执行部分或称为执行语句部分，用以完成程序的功能。执行部分的第一行是输出语句，调用 printf 函数在显示器上输出提示字符串，请操作人员输入自变量 x 的值。第二行为输入语句，调用 scanf 函数，接受键盘上输入的数并存入变量 x 中。第三行是调用 sin 函数并把函数值送到变量 s 中。第四行是用 printf 函数输出变量 s 的值，即 x 的正弦值。程序结束。

例如：

```
printf("input number:\n");
scanf("%lf",C10F10&x);
s=sin(x);
printf("sine of %lf is %lf\n",C10F10x,s);
```

运行本程序时，首先在显示器屏幕上给出提示串 input number，这是由执行部分的第一行完成的。用户在提示下从键盘上键入某一数，如 5，按下回车键，接着在屏幕上给出计算结果。

为了进一步说明 C 语言源程序结构的特点，先看以下程序。该程序由简到难，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 1-2】在屏幕上输出“Hello,World!”。

```
//example1.1 The first C Program
#include <stdio.h>
void main()
{
    printf("Hello,World!\n");
}
```

程序分析:

(1) main 函数——从哪里开始,到哪里结束。main()表示“主函数”,每个 C 语言程序都必须有且只能有一个 main 函数,它是每一个 C 语言程序执行的起始点(入口点)。程序是从 main 函数的第一条语句开始执行,然后顺序执行 main 函数中的其他语句。main 函数执行结束,整个程序结束,而不论 main 函数书写在程序中的任何位置。

void 表示该主函数没有返回值。

用 {} 括起来的是主函数 main 的函数体。main 函数中的所有操作(或语句)都包含在这一对花括号之间,即 main 函数的所有操作都在 main 函数体中。

(2) 如何在屏幕上显示文字——函数调用。本程序的“主函数”main 中只有一条函数调用语句——printf()是 C 语言的库函数,用于程序中数据的输出(显示在屏幕上)。本例是将一个字符串“Hello,World!\n”输出,即在屏幕上显示“Hello,World!”。

(3) 程序的最小独立单元——语句。每条语句用“;”结束。

(4) 编译器如何认识 printf 函数——#include 预处理命令。C 语言规定,函数在调用前必须声明。printf 函数的声明就包含在头文件“stdio.h”中,头文件是每一个 C 程序必不可少的组成部分,因为一个 C 程序至少要包含输入或输出函数。

(5) 程序里的说明书——注释。“/* example1.1 The first C Program*/”为注释,注释是为了改善程序的可读性(提示、解释作用),在编译、运行时不起作用(编译时会跳过注释,目标代码中不会包含注释)。注释可以放在程序的任何位置,并允许占用多行,只是需要注意“/*”“*/”匹配,不要嵌套注释。

也可以使用“//”作为注释符,一个“//”只能注释一行,而“/*”和“*/”配对可以注释多行。

注释与软件的文档同等重要,要养成书写注释的良好习惯,这对软件的维护相当重要。因为程序是要给别人看的,自己也许还会看自己几年前编写的程序,清晰的注释有助于读者理解算法和程序的思路。

在程序开发过程中,还可以用注释帮助调试程序,即暂时屏蔽一些不需要运行的语句,以后可以方便地恢复。

【例 1-3】求两个数的较大值。

```
#include <stdio.h>
```

```

int max(int x,int y) /* 计算两数中较大的数 */
{int z; /* 声明部分, 定义变量 */
    if(x>y)
        z=x;
    else
        z=y;
    return z; /* 将 z 值返回, 通过 max 带回调用处 */
}
void main( )
{int a,b,c; /* 声明部分, 定义变量 */
scanf("%d,%d",&a,&b);
c=max(a,b); /* 调用 max, 将调用结果赋予 c */
printf("max=%d",c);
}

```

程序分析:

(1) 本程序包括两个函数。其中, 主函数 main 仍然是整个程序执行的起点, 函数 max 计算两数中较大的数。

(2) 主函数 main 调用 scanf 函数, 获得两个整数, 分别存入 a、b 两个变量中, 然后调用函数 max, 获得两个数中较大的数, 并赋予变量 c。最后输出变量 c 的值(结果)。

(3) max 是用户自定义的函数, int max(int x,int y) 是函数入口, 表示此函数运行时需要获得两个整数值, 数据处理结束后会返回一个整数值。

(4) 函数 max 同样也用 { 和 } 将函数体括起来。max 的函数体是函数功能的具体实现, 它从参数表获得数据, 将处理后得到的结果存储于 z 中, 然后将 z 返回调用函数 main。

(5) 本例表明函数除了调用库函数外, 还可以调用用户自定义的函数。

综合上述两个例子, 我们对 C 语言程序的基本组成和程序结构有了一个初步了解:

(1) C 程序由函数构成(函数是 C 程序的基本单位), 所有的 C 语言程序都由一个或多个函数构成。其中, main 函数必须有且只能有一个。

(2) 被调用的函数可以是系统提供的库函数, 也可以是用户根据需要自己设计编写的函数。程序的全部工作由各个函数完成。编写 C 程序就是编写一个个的函数。

(3) main 函数(主函数)是每个程序执行的起始点。无论 main 函数在程序中的哪个位置, 一个 C 程序总是从 main 函数开始执行, 并且也是从 main 函数结束。

(4) 一个函数由函数首部和函数体两部分组成。

① 函数首部: 一个函数的第一行, 由函数类型、函数名、参数类型和参数名组成。一般格式为

函数类型函数名 (参数类型及参数列表)

② 函数体：函数首部下方用一对花括号括起来的部分。函数体一般包括声明和执行两部分。

例如函数 max：

函数类型	函数名称	参数类型	参数名	参数类型	参数名
↓	↓	↓	↓	↓	↓
int	max	(int	x,	int	y) /* 函数首部 */
{					/* 函数开始 */
int z;					/* 声明部分, 定义变量 */
if(x>y)					
z=x;					
else					
z=y;					/* 函数体, 执行部分 */
return z;					
}					/* 函数结束 */

(5) C 程序书写格式自由，一行可以写几条语句，一条语句也可以写在多行上。每条语句的最后必须以一个分号“;”表示语句的结束。

(6) 可以使用“/* */”和“//”对 C 程序中的任何部分进行注释。注释可以提高程序的可读性，使用注释是编程人员必须养成的良好习惯。一个较大的系统往往由多人合作开发，程序文档、注释是其中重要的交流工具。

(7) C 语言本身不提供输入/输出语句，输入/输出操作是通过调用库函数（scanf、printf 等）来完成的。

输入/输出操作涉及具体计算机硬件，把输入/输出操作放在函数中处理，可以简化 C 语言和 C 编译系统，便于 C 语言在各种计算机上实现。不同的计算机系统需要对函数库中的函数做不同的处理，以便实现同样或类似的功能。

1.2.3 书写 C 程序时应遵循的规则

从书写清晰，便于阅读、理解、维护的角度出发，在书写程序时，应遵循以下规则。

(1) 一个说明或一个语句占一行。

(2) 用 {} 括起来的部分，通常表示了程序的某一层结构。{} 一般与该结构语句的第一个字母对齐，并单独占一行。

(3) 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写。以便看起来更加清晰，增加程序的可读性。在编程时应力求遵循这些规则，以养成良好的编程风格。

1.2.4 C 语言的字符集

(1) 字母

小写字母 a ~ z 共 26 个, 大写字母 A ~ Z 共 26 个。

(2) 数字

0 ~ 9 共 10。

(3) 空白符

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, 编译程序对它们忽略。因此在程序中使用空白符与否, 对程序的编译不发生影响, 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。

(4) 标点和特殊符号

标点和特殊字符包括常用的标点和一些特殊字符, 和其他文档中的相同。

C 语言词汇主要包括标识符、关键字、运算符、分隔符、常量、注释符 6 部分内容。其中, C 语言的注释符是以 “/*” 开头并以 “*/” 结尾的串。在 “/*” 和 “*/” 之间的即为注释。程序编译时, 不对注释做任何处理。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。在调试程序中对暂不使用的语句也可用注释符括起来, 使翻译跳过不做处理, 待调试结束后再去掉注释符。

1.3 C 语言程序设计步骤

计算机之所以能够产生如此大的影响, 其原因不仅在于人们发明了机器本身, 更重要的是, 人们为计算机开发出了不计其数的能够指挥计算机完成各种工作的程序。正是这些功能丰富的程序给了计算机无尽的生命力, 是程序设计工作者智慧的结晶。

所谓程序, 是指用计算机语言描述的、为解决某一问题、满足一定语法规则的语句序列, 而程序设计就是用某种程序语言编写解决这些问题的步骤的过程。

要设计出一个程序, 首先要明确要处理问题中的数据结构 (即数据和数据之间的关系); 其次要描述出对问题的处理方法和步骤 (即算法)。因此, 数据结构与算法是程序设计过程中密切相关的两个方面。著名计算机科学家 Niklaus Wirth 教授提出了关于程序的著名公式 (沃思公式): 程序 = 数据结构 + 算法。这个公式说明了程序设计的主要任务。

那么, 如何用 C 语言进行程序设计呢? 一般包含以下步骤。

1.3.1 分析问题

使用计算机解决具体问题时, 首先要对问题进行充分的分析, 确定问题是什么, 解决问

题的步骤又是什么。针对所要解决的问题，找出已知的数据和条件，确定所需的输入、处理及输出对象。

1.3.2 确定数据结构和算法

在分析求解问题的基础上，确定程序中数据的类型和组织存储形式，即确定存放数据的结构。针对问题的分析和确定的数据结构，选择合适的算法加以实现。注意，这里所说的“算法”泛指解决某一问题的方法和步骤。

1.3.3 编写与编辑源程序

根据确定的数据结构和算法，用 C 语言把该数据结构和算法严格地用符合 C 语言语法规则的代码描述出来，也就是编写出源程序代码。将源程序输入计算机，并以约定的扩展名“.c”的文本文件形式保存在外存上，例如 file1.c、t.c 等。

用于编辑源程序的软件是编辑程序。编辑程序是提供给用户书写程序的软件环境，用来输入和修改源程序。

1.3.4 编译、连接与运行

编译是把 C 语言源程序翻译成计算机能识别的二进制指令形式的目标程序。编译过程由编译程序完成。编译程序自动对源程序进行句法和语法检查，当发现错误时，将错误的类型和所在的位置显示出来，提供给用户，以帮助用户修改源程序中的错误。如果未发现句法和语法错误，对目标代码进行优化后生成目标程序。目标程序的文件扩展名是“.obj”。

目标程序尽管计算机能够读懂，但仍不能执行，这是因为其中还缺少一些内容（如库函数、其他目标程序、各种资源等的二进制程序），需要把它们与目标程序“组合起来”，这个过程称为连接。经过连接之后，就生成了可执行程序，可执行程序的扩展名为“.exe”。连接过程是由连接程序（也称链接程序或装配程序）完成的。运行程序是指将可执行程序投入运行，得到程序处理的结果。如果程序运行结果与预测结果不一致，必须重新回到第 3 步（也有可能从第 1 步开始重新分析问题），对程序进行编辑修改、编译和运行，直到得到正确的结果为止。

与编译、连接不同，运行可以脱离语言处理环境，直接在操作系统环境下进行。

1.3.5 写出程序的文档

程序是提供给用户使用的，如同正式的产品应当提供产品说明书一样，正式提供给用户使用的程序，必须向用户提供程序说明书。内容应包括：程序名称、程序功能、运行环境、程序的装入和启动、需要输入的数据以及使用注意事项等，为程序的使用、维护做好基础工作。

很多软件能够完成从第 3 和第 4 步的全部过程，这种软件称为 C 语言集成开发系统，如 CodeBlocks、Visual C++ 等。在集成开发系统中开发 C 程序的过程可以用如图 1-1 所示。