

21世纪高等学校计算机规划教材

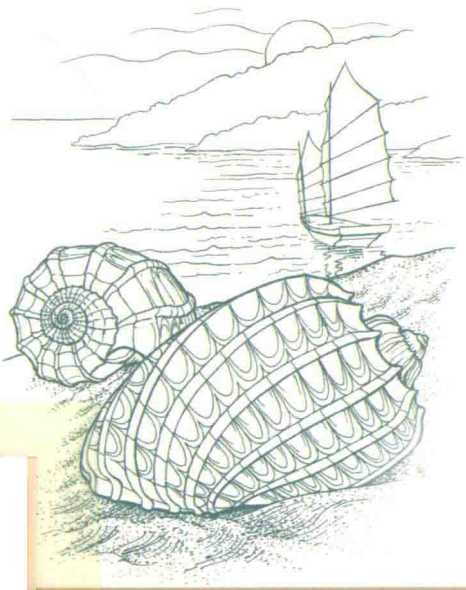
21st Century University Planned Textbooks of Computer Science

C语言 程序设计教程

The C Programming Language

王继鹏 陈希 主编

杨玉蓓 方洁 叶杨 副主编



高校系列



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

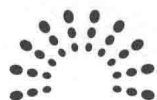
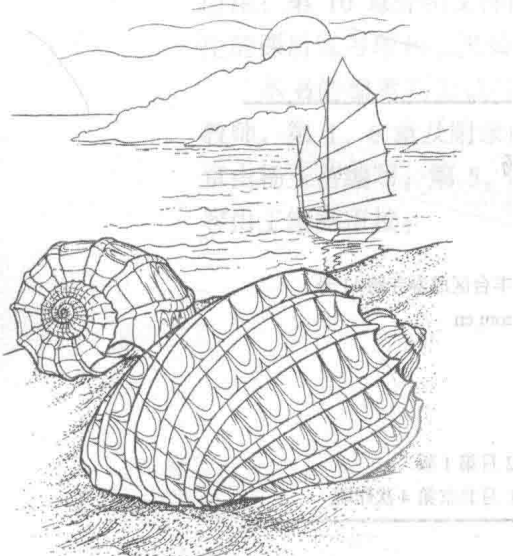
21st Century University Planned Textbooks of Computer Science

C语言 程序设计教程

The C Programming Language

王继鹏 陈希 主编

杨玉蓓 方洁 叶杨 副主编



高校系列

人民邮电出版社

北京

图书在版编目 (CIP) 数据

C语言程序设计教程 / 王继鹏, 陈希主编. — 北京: 人民邮电出版社, 2018. 12 (2019. 8重印)
21世纪高等学校计算机规划教材
ISBN 978-7-115-49168-8

I. ①C… II. ①王… ②陈… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第210225号

内 容 提 要

本书是为将C语言作为入门语言的程序设计课程的初学者所编写的, 以培养读者程序设计的基本能力为基本目标。

本书全面系统地介绍了C语言的语法规则和结构化程序设计的方法, 并用大量的实例剖析了C语言的重点和难点。全书内容包括概述、基本数据类型、运算符和表达式、结构化控制语句、数组、函数和指针、结构体以及文件。

本书结构合理、概念清晰、实例典型, 适合作为高等院校计算机及相关专业学习C语言的教材, 也可以作为对C语言程序设计感兴趣的读者的自学用书。

-
- ◆ 主 编 王继鹏 陈 希
副 主 编 杨玉蓓 方 洁 叶 杨
责任编辑 邹文波
责任印制 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>
北京市艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 17 2018年12月第1版
字数: 424千字 2019年8月北京第4次印刷
-

定价: 55.00 元

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

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号

目 录

第 1 章 概述.....1

1.1 C 语言的发展.....1	1
1.1.1 C 语言的发展简史.....1	1
1.1.2 C 语言的特点.....1	1
1.1.3 C 语言的版本.....2	2
1.1.4 C 和 C++.....2	2
1.2 认识 C 语言程序.....2	2
1.2.1 简单的 C 语言程序.....2	2
1.2.2 C 语言源程序的结构特点.....5	5
1.2.3 C 语言的字符集.....6	6
1.2.4 C 语言的词汇.....6	6
1.3 使用开发工具编写 C 语言程序.....8	8
1.3.1 C 语言程序的开发步骤.....8	8
1.3.2 使用 Microsoft Visual C++ 6.0 编写 C 语言程序.....9	9
1.3.3 编写 C 语言程序时应遵循的良好习惯.....15	15
习题.....16	16

第 2 章 数据类型、运算符和表达式..... 18

2.1 C 语言的数据类型.....18	18
2.2 常量与变量.....19	19
2.2.1 常量.....19	19
2.2.2 变量.....20	20
2.3 整型数据.....21	21
2.3.1 整型常量.....21	21
2.3.2 整型变量.....21	21
2.3.3 整型常量的类型.....23	23
2.4 浮点型数据.....24	24
2.4.1 浮点型常量.....24	24
2.4.2 浮点型变量.....24	24
2.4.3 浮点型常量的类型.....26	26

2.5 字符型数据.....26	26
2.5.1 字符型常量.....26	26
2.5.2 字符型变量.....28	28
2.5.3 字符串常量.....29	29
2.6 C 语言的运算符和表达式.....30	30
2.6.1 C 语言运算符和表达式简介.....30	30
2.6.2 算术运算符和算术表达式.....31	31
2.6.3 赋值运算符和赋值表达式.....33	33
2.6.4 逗号运算符和逗号表达式.....35	35
2.7 C 语言数据类型转换.....36	36
习题.....38	38

第 3 章 顺序结构..... 41

3.1 顺序程序设计.....41	41
3.2 C 语句概述.....42	42
3.3 赋值语句.....43	43
3.4 字符数据的输入/输出.....44	44
3.4.1 字符输出函数 putchar.....44	44
3.4.2 字符输入函数 getchar.....45	45
3.5 格式输入与输出.....45	45
3.5.1 格式输出函数 printf.....45	45
3.5.2 格式输入函数 scanf.....50	50
3.6 顺序结构程序设计举例.....53	53
习题.....56	56

第 4 章 选择结构..... 60

4.1 关系运算符与关系表达式.....60	60
4.1.1 关系运算符.....60	60
4.1.2 关系表达式.....61	61
4.2 逻辑运算符与逻辑表达式.....61	61
4.2.1 逻辑运算符.....61	61
4.2.2 逻辑表达式.....62	62
4.3 if 结构.....63	63
4.3.1 单 if 语句.....63	63

4.3.2 if-else 语句	65	6.3.5 字符串处理函数	115
4.3.3 if-else if-else 语句	68	6.3.6 字符数组的程序举例	118
4.4 嵌套 if 结构	70	习题	119
4.5 switch 结构	72	第 7 章 函数	125
4.6 多重 if 结构和 switch 结构的比较	73	7.1 函数的定义	127
4.7 条件运算符	74	7.1.1 函数的分类	127
4.8 应用举例	75	7.1.2 函数的定义	127
习题	77	7.1.3 函数的参数和返回值	128
第 5 章 循环结构	81	7.2 函数的调用	131
5.1 循环	81	7.2.1 函数调用的形式	131
5.2 while 语句	81	7.2.2 对被调函数的声明	132
5.3 do-while 语句	83	7.3 函数的嵌套调用和递归调用	134
5.4 for 语句	85	7.3.1 函数的嵌套调用	134
5.4.1 基本的 for 语句	85	7.3.2 函数的递归调用	135
5.4.2 各种特殊形式的 for 语句	87	7.4 数组作为函数的参数	137
5.5 三种循环语句的比较	88	7.4.1 数组元素作为函数的参数	137
5.6 break 语句	89	7.4.2 数组名作为函数的参数	137
5.7 continue 语句	90	7.5 变量的作用域和生存期	138
5.8 嵌套循环	91	7.5.1 变量的作用域	138
5.9 应用举例	95	7.5.2 变量的存储类别和生存期	142
习题	96	习题	146
第 6 章 数组	100	第 8 章 指针	149
6.1 一维数组	101	8.1 指针的概念	149
6.1.1 一维数组的定义	101	8.2 指针变量的定义与引用	151
6.1.2 一维数组的初始化	102	8.2.1 指针变量的定义	151
6.1.3 一维数组的引用	103	8.2.2 指针变量的引用	152
6.1.4 一维数组的程序举例	104	8.3 指针与数组	155
6.2 二维数组	108	8.3.1 指向一维数组的指针	155
6.2.1 二维数组的定义	108	8.3.2 指向多维数组的指针	159
6.2.2 二维数组的初始化	109	8.3.3 指针数组和数组指针	162
6.2.3 二维数组的引用	109	8.4 指针与字符串	163
6.2.4 二维数组的程序举例	109	8.5 指针与函数	165
6.3 字符数组与字符串	111	8.5.1 指针变量作为函数参数	165
6.3.1 字符数组的定义	112	8.5.2 数组的指针作为函数参数	166
6.3.2 字符数组的初始化	112	8.5.3 指向函数的指针	167
6.3.3 字符数组的引用	113	8.5.4 返回指针值的函数	169
6.3.4 字符数组的输入/输出	113	8.6 指向指针的指针	170

8.7 指针与动态内存分配	171	10.1.3 文件类型指针	205
8.8 应用举例	173	10.2 文件的操作	206
习题	175	10.2.1 文件的打开	206
第9章 结构体	180	10.2.2 文件的关闭	208
9.1 结构体概述	180	10.2.3 文件的重命名	208
9.1.1 声明结构体	180	10.2.4 文件的删除	208
9.1.2 定义结构体变量	181	10.3 文件的读写	209
9.1.3 引用结构体变量	182	10.3.1 字符的读取和写入	209
9.1.4 结构体变量的初始化	183	10.3.2 字符串的读取和写入	210
9.2 结构体数组	183	10.3.3 按格式读取和写入	211
9.2.1 定义结构体数组	184	10.3.4 数据块存取函数	212
9.2.2 结构体数组的初始化	184	10.4 文件的定位	214
9.2.3 引用结构体数组元素	185	10.4.1 rewind 函数	214
9.3 结构体指针	187	10.4.2 fseek 函数和随机读写	215
9.3.1 指向结构体变量的指针变量	187	10.4.3 ftell 函数	215
9.3.2 指向结构体数组的指针	188	10.5 文件状态检测	216
9.3.3 用结构体变量和指向结构体的 指针作函数参数	189	10.6 综合应用	217
9.4 共用体	192	习题	222
9.4.1 共用体的定义	192	附录 A 基本控制字符/字符与 ASCII 代码对照表	224
9.4.2 共用体变量的引用方式	193	附录 B 运算符的优先级和 结合性	225
9.4.3 共用体类型数据的特点	194	附录 C C 语言中的关键字	227
9.5 枚举类型	196	附录 D C 语言的常用库函数	228
9.5.1 枚举类型的声明	196	附录 E C 语言程序编译常见提示 信息	234
9.5.2 枚举类型变量的定义	197	附录 F C 语言开发常见错误	248
9.5.3 枚举类型变量的引用	197	附录 G Visual C++ 6.0 环境下的 C 语言程序调试	260
9.6 自定义类型标识符	200		
习题	201		
第10章 文件	204		
10.1 C 文件概述	204		
10.1.1 缓冲文件系统	204		
10.1.2 文件的存储方式	205		

第1章

概述

本章主要介绍 C 语言的发展与特点、C 语言和源程序的结构特点以及开发应用程序的主要步骤。通过本章的学习，读者将对 C 语言有概括性的了解，能够依照例题编写一些简单程序。

1.1 C 语言的发展

1.1.1 C 语言的发展简史

C 语言的原型是 ALGOL 60 语言。1963 年，剑桥大学将 ALGOL 60 语言发展成为 CPL (Combined Programming Language)。1967 年，剑桥大学的马丁·理查兹对 CPL 进行了简化，开发出了 BCPL (Basic Combined Programming Language)。1970 年，美国贝尔实验室的肯·汤普森将 BCPL 进行了修改，并为它起了一个新名字“B 语言”，意思是提炼出 CPL 的精华，并且他用 B 语言写了第一个 UNIX 操作系统。而在 1972 年，B 语言也被“提炼”了一下，美国贝尔实验室的丹尼斯·里奇在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二字母作为这种语言的名字，这就是 C 语言。

为了推广 UNIX 操作系统，1977 年丹尼斯·里奇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》。1978 年布莱恩·科尔尼干和丹尼斯·里奇出版了名著《C 语言程序设计》*The C Programming Language*，从而使 C 语言成为目前世界上使用最广泛的高级程序设计语言。1988 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，这些 C 语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准研究所 (ANSI) 为 C 语言制定了一套 ANSI 标准，成为现行的 C 语言标准。

早期的 C 语言主要是用于 UNIX 系统。C 语言的强大功能和各方面的优点使其逐渐为人们认识，到了 20 世纪 80 年代，C 开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到了广泛应用，成为当代最优秀的程序设计语言之一。C 语言发展迅速，而且成为最受欢迎的语言之一，主要因为它具有强大的功能。许多著名的操作系统软件，如 DOS、Windows、Linux 都是由 C 语言编写的。

1.1.2 C 语言的特点

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，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。它的应用范围广泛，具备很强的数据处理能力，不仅仅是在软件开发上，而且各类科研都需要用到C语言，适于编写系统软件及三维、二维图形和动画，例如，单片机以及嵌入式系统开发。此外，C语言还具有效率高、可移植性强等特点，被广泛移植到各类计算机上。

1.1.3 C 语言的版本

1989年，ANSI发布了第一个完整的C语言标准——ANSI X3.159—1989，简称“C89”，不过人们也习惯称其为“ANSI C”。C89在1990年被国际标准化组织（International Organization for Standardization, ISO）采纳，所以也有“C90”的说法。

1999年，在做了一些必要的修正和完善后，ISO发布了新的C语言标准，并将其命名为ISO/IEC 9899:1999，简称“C99”。C99被ANSI于2000年3月采用。2011年12月8日，ISO正式发布了新的C语言的新标准C11，之前被称为C1X，官方名称为ISO/IEC 9899:2011。新的标准提高了对C++的兼容性，并增加了一些新的特性。

1.1.4 C 和 C++

C++语言是在C语言的基础上开发的一种集面向对象编程、泛型编程和过程化编程于一体的编程语言。在C语言的基础上，1983年又由贝尔实验室的本贾尼·斯特劳斯特卢普推出了C++语言。C++语言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++语言提出了一些更为深入的概念，它支持的这些面向对象的概念容易将问题空间直接映射到程序空间，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。C++语言应用较为广泛，是一种静态数据类型检查的、支持多重编程的通用程序设计语言。C++语言支持过程化程序设计、数据抽象、面向对象设计、制作图标等多种程序设计风格。因而也增加了整个语言的复杂性，掌握起来有一定难度。C++语言目前的最新标准是2014年8月18日经过C++语言标准委员投票通过的C++14。

C语言是一个结构化语言，它的重点在于算法与数据结构。C程序的设计首要考虑的是如何通过一个过程，对输入（或环境条件）进行运算处理得到输出（或实现过程（事物）控制）。C++首要考虑的是如何构造一个对象模型，让这个模型能够契合与之对应的问题域，这样就可以通过获取对象的状态信息得到输出或实现过程（事物）控制。因此，C语言和C++语言的最大区别在于它们解决问题的思想方法不一样。

C语言是C++语言的基础，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++语言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 认识 C 语言程序

1.2.1 简单的 C 语言程序

为了说明C语言程序结构的特点，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语言程序的基本部分和书写格式。

【例 1.1】在屏幕上输出“Hello world!”。

程序实现：

```
#include <stdio.h>           //头文件包含命令
void main( )                 //main 是主函数的函数名
{                             //用{表示函数体的开始
printf("Hello world!\n");    //输出“Hello world!”到屏幕
}                             //用}表示函数体的结束
```

运行结果：**Hello world!**
Press any key to continue

程序分析：

本例的第 1 行#include <stdio.h>称为预处理命令。预处理命令是以#开头的代码行。#必须是该行除了任何空白字符外的第一个字符。#后是命令关键字，在关键字和#之间允许存在任意个数的空白字符。整行语句构成了一条预处理命令，该命令将在编译器进行编译之前对源代码做某些转换。常见的预处理命令有两种：#include 文件包含命令和#define 宏定义命令。#include 文件包含命令，其意义是把尖括号<>或双引号" "内指定的文件包含到本程序中，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，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。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。在本例中，stdio.h 是标准的输入输出头文件，输出函数 printf 是该文件中的标准输入输出函数，表示输出双引号里的内容，其中“\n”表示换行。

运行结果最下方的“Press any key to continue”是由操作系统自动添加的，意思是“按任意键继续”，不是程序运行结果的组成部分。此时按下键盘上的任意一个键后，将返回到 C 语言编辑开发环境。

【例 1.2】计算两数 a、b 之和，并输出结果 c。

程序实现：

```
#include <stdio.h>
void main( )
{
    int a,b,c;                //定义三个整型变量 a,b,c
    a=1; b=2;                //对变量 a、b 赋值
    c=a+b;                   //求 a、b 的和，并赋值给 c
    printf("c=%d\n",c);      //按照整型值输出 c 的值
}
```

运行结果：**c=3**
Press any key to continue

程序分析：

本例的主函数体分为两部分，前半部分为定义部分，后半部分为执行部分。定义是对变量的定义。C 语言规定，程序中所有用到的变量都必须先定义，然后才能使用，否则将会出错。程序第 4 行是定义语句，定义了整型变量 a、b、c。

定义部分后为执行部分或称为执行语句部分，用以完成程序的功能。第 5 行是赋值语句，使 a 和 b 的值分别为 1 和 2。第 6 行计算 a+b，并将结果赋值给 c。第 7 行是输出语句，以 %d 的格式输出 c 的值，其中%d 表示“十进制整数类型”。在执行输出时，“c=”按原样输出，

在%d的位置上代以一个十进制整数，printf函数中括号内逗号右端的c是要输出的变量。运行时将会把c的值以十进制整数的格式输出在%d的位置上。

【例 1.3】 从键盘输入一个实数，求它的正弦值，然后输出结果。

程序实现：

```
#include <stdio.h>
#include <math.h>           //包含头文件 math.h
void main( )
{
    double x,s;             //定义两个双精度浮点型变量 x,s
    printf("input number x:\n");
    scanf("%lf",&x);       //调用 scanf 函数从键盘输入一个双精度浮点型值到变量 x
    s=sin(x);              //调用 sin 函数计算变量 x 的正弦值，并把结果赋给变量 s
    printf("sin(%lf)= %lf\n",x,s); //按照双精度浮点型输出 x 对应的正弦值 s
}
```

运行结果：

```
input number x:
3.14
sin(3.140000)= 0.001593
Press any key to continue
```

程序分析：

本例的前两行是预处理命令，预处理命令根据需要可以有多行。在本例中，使用了三个 C 语言提供的库函数：正弦函数 sin、输入函数 scanf 和输出函数 printf。正弦函数 sin 是数学函数，其所在的头文件为 math.h 文件，因此在程序的预处理部分用 include 命令包含了 math.h。另两个函数 scanf 和 printf 是标准输入/输出函数，其所在的头文件为 stdio.h，所以在主函数前也用 include 命令包含了 stdio.h 文件。

本例中使用了两个变量 x 和 s，分别用来表示输入的自变量和正弦函数值。由于 sin 函数要求这两个量必须是双精度浮点型，故第 5 行定义部分用类型标识符 double 来定义这两个变量。第 6 行是执行部分的输出语句，调用 printf 函数在屏幕上输出提示信息“input number:”，通知用户输入自变量 x 的值。第 7 行为输入语句，调用 scanf 函数，接收用户从键盘上输入的数并存入变量 x 中。另外，要注意 sin 函数要求输入的参数 x 为弧度值。第 8 行是调用 sin 函数并把结果函数值送到变量 s 中。第 9 行是用 printf 函数输出变量 s 的值（即 x 的正弦值）给用户。

本例程序执行步骤如下。

- (1) 定义变量 x、s。
- (2) printf 函数在屏幕上显示提示信息，通知用户输入一个数 x。
- (3) 用户输入一个数（如 3.14），然后按下【Enter】键，输入结束。
- (4) scanf 函数接收到这个数，并送入变量 x 中。
- (5) 调用 sin 函数，把 x 的值作为实际参数传送给 sin 函数。
- (6) 在 sin 函数中计算 x 的正弦值，计算完将结果值返回给主函数。
- (7) 主函数收到返回值后赋给变量 s。
- (8) 在屏幕上输出 sin(x)=s 的值。

【例 1.4】 输入两个整数，比较后输出其中较大的数。

程序实现：

```
#include <stdio.h>
int max(int a, int b);     //声明一个整型函数 max，有两个整型参数 a, b
```

```

void main( )
{
    int x,y,z;
    printf("input two numbers:\n");
    scanf("%d%d",&x,&y);           //调用 scanf 函数从键盘输入两个整型值到变量 x, y
    z=max(x,y);                     //调用 max 函数, 并传递参数 x,y. 然后将返回值赋给 z
    printf("max=%d\n",z);
}
int max(int a, int b)               //定义 max 函数, 有两个整型参数 a、b
{
    if(a>b)
        return a;
    else
        return b;
}

```

运行结果:

```

input two numbers:
3 5
max=5
Press any key to continue

```

程序分析:

本例的程序由主函数和 max 函数两个函数组成。C 语言规定, 程序中所有用到的自定义函数都必须先声明或定义, 然后才能使用, 否则将会出错。本例中主函数调用了 max 函数, 而 max 函数是一个用户自定义函数, 因此应在主函数调用 max 函数前进行声明或定义, 本例中在程序的第 2 行进行了声明。max 函数的功能是: 如果传入的参数 a 的值大于 b 的值, 则返回 a 的值; 否则返回 b 的值。

本例程序执行步骤如下。

- (1) 定义变量 x、y、z。
- (2) printf 函数在屏幕上显示提示信息, 通知用户输入两个数。
- (3) 用户输入两个数 (如 3 和 5), 然后按下【Enter】键, 输入结束。
- (4) scanf 函数接收这两个数, 并送入变量 x、y 中。
- (5) 调用 max 函数, 把 x、y 的值作为实际参数传送给 max 函数的形式参数 a、b。
- (6) 在 max 函数中比较 a、b 的大小, 将大者的值返回给主函数。
- (7) 主函数收到返回值后赋给变量 z。
- (8) 在屏幕上输出 max=z 的值。

1.2.2 C 语言源程序的结构特点

- 一个 C 语言源程序可以由一个或多个源文件组成。每个源文件可由一个或多个函数组成。
- C 语言源程序的基本组成单位是函数; 一个 C 程序可由若干个函数组成, 其中必须有且仅有一个以 main 命名的主函数, 其余的函数名称可由编程者自行设定。
- C 程序的执行总是由 main() 函数的第一个可执行语句开始, 到 main() 函数的最后一个可执行的语句结束; 而其他函数都是在 main() 函数开始执行以后, 通过函数的调用才得以运行。
- C 语言系统提供了丰富的库函数, 用户在程序中需要调用某个库函数时, 必须用预处理命令 #include 将描述该库函数的头文件包含进去。预处理命令通常应放在源程序的最前面。

- 每一条语句都必须以分号结尾。但预处理命令、函数头和大括号“}”之后不能加分号。
- 一条语句可以分开放在任意多行内；一行中也可以有多个语句。
- 标识符及关键字彼此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。

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。

- 字母。小写字母 a~z 共 26 个，大写字母 A~Z 共 26 个。
- 数字。0~9 共 10 个。
- 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，编译程序会忽略它们。因此在程序中使用空白符与否，对程序的编译不产生影响，但在程序中适当的位置使用空白符将使程序更具清晰性和可读性。
- 标点和特殊字符。

1.2.4 C 语言的词汇

在 C 语言中使用的词汇分为 6 类：标识符、关键字、运算符、分隔符、常量、注释符等。

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，标识符只能是由字母 A~Z、a~z，数字 0~9 和下划线“_”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, 3x,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 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 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) 在标识符中大小写敏感，即是有区别的。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，命名应尽量有相应的意义，以便阅读理解。

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。如前面例题中用到的 `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【例 1.4】中用到的 `if-else` 就是条件语句的语句定义符。

(3) 预处理命令字

预处理命令字用于表示一个预处理命令。如前面各例中用到的 `include`。

ANSI C 提供的 32 个关键字如下。

auto	break	case	char	const
continue	default	do	double	else
enum	extern	float	for	goto
if	int	long	register	return
short	signed	sizeof	static	struct
switch	typedef	unsigned	union	void
volatile	while			

Turbo C 扩充的 11 个关键字如下。

asm	_cs	_ds	_es	_ss	cdecl
far	huge	interrupt	near	pascal	

注意：在 C 语言中，关键字都是小写的。

3. 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。运算符共有 34 种。C 语言把括号、赋值、逗号等都作为运算符处理。从而使 C 语言的运算类型极为丰富，可以实现其他高级语言难以实现的运算。

C 语言的 34 种运算符分类如下。

- (1) 算术运算符：+、-、*、/、%、++、--。
- (2) 关系运算符：<、<=、==、>、>=、!=。
- (3) 逻辑运算符：!、&&、||。
- (4) 位运算符：<<、>>、~、|、^、&。
- (5) 赋值运算符：=及其扩展。
- (6) 条件运算符：?:。
- (7) 逗号运算符：,。
- (8) 指针运算符：*、&。
- (9) 求字节数：sizeof。
- (10) 强制类型转换：(类型)。
- (11) 分量运算符：.、->。
- (12) 下标运算符：[]。
- (13) 其他：()。

注意：各种运算符混合使用时，优先级与结合方法是难点。

4. 分隔符

在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用在变量定义、声明以及函数参数表中，分隔各个变量。空格一般用作间隔符，如在关键字、标识符之间必须要有一个以上的空格符作间隔，否则将会出现语法错误，例如，把“int a;”写成“inta;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、转义字符等多种。

6. 注释符

C 语言的注释符是“/*...*/”以及“//”。在“/*”和“*/”之间的所有内容都是注释。“//”后面的一行是注释。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，使编译程序跳过此处不做处理，待调试结束后再去掉注释符。

7. 控制语句

C 语言中提供了 9 种控制语句，在程序中起到流程控制作用。

if-else	switch	for	while	do-while
continue	break	goto	return	

1.3 使用开发工具编写 C 语言程序

1.3.1 C 语言程序的开发步骤

C 语言程序的开发分为 4 个步骤：编辑、编译、连接、运行，如图 1.1 所示。

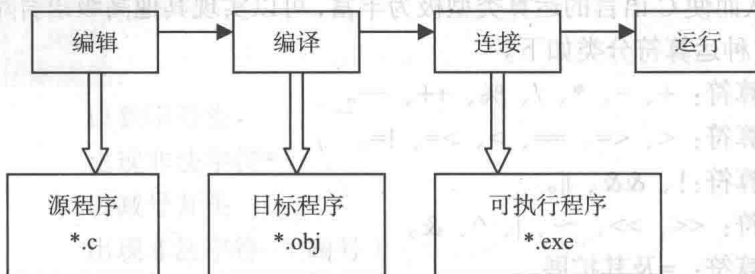


图 1.1 C 语言程序的开发步骤

1. 编辑

编辑源程序，可以用文字处理软件编写源程序，也可以用集成化的编译开发程序设计软件，如 Microsoft VC++。C 语言源程序的扩展名为.c。

2. 编译

源程序编写好之后，可以进行编译。编译是将源程序转换成二进制文件，即目标文件，扩展名为.obj。在编译过程中可以发现在源程序编写过程中出现的错误，这种错误一般是由书写错误造成的，因此这种错误称为语法错误，这种错误是易于修改的。必须在此阶段将所有的语法错误修改完成才能进入下一步，语法错误大多可以根据编译工具的提示加以修正。

3. 连接

编译成功后的文件并不能运行,因为这种程序虽然称之为目标文件,但仍是半成品,不能执行。在目标程序中还没有为函数、变量等安排具体的地址,因此也称为浮动程序。所以连接就是将若干目标文件加以归并、整理,为所有的函数、变量分配具体地址,同时将库函数连接到目标文件中,生成可执行程序,扩展名为.exe。

在连接的过程中也可能发现错误,这种错误可能是由于设计不足或缺陷引起的,这种错误称为逻辑错误。逻辑错误是不易被发现的,应尽可能地加以避免。逻辑错误的修正往往需要对程序进行跟踪调试才能完成。

4. 运行

根据运行的不同目的,运行可分为调试运行、测试运行和应用运行。

(1) 调试运行:是专门为验证某些函数的正确性而进行的,被运行的主函数通常就是一个调试程序,运行时通过输入一些特定的数据,观察它是否产生预期的输出。如果发现任何不正常的情况,应配合使用程序跟踪等手段,观察程序是否按预期的流程运行,程序中的某些变量的值是否如预期的那样变化,从而判定出错的具体原因和位置,以便加以纠正。

(2) 测试运行:是应用运行前的试运行,是为了验证整个应用系统的正确性,如果发现错误,应进一步判断错误的原因和产生错误的大致位置,以便加以纠正。

(3) 应用运行:是指程序正式投入使用后的运行,目的是通过程序运行完成预先设定的功能,从而获得相应的效益。

注:以上提到的源程序、目标程序、可执行程序的区别如表 1-1 所示。

表 1-1 源程序、目标程序、可执行程序的区别

	源程序	目标程序	可执行程序
内容	高级语言	机器语言	机器语言
是否可执行	否	否	是
文件扩展名	.c	.obj	.exe

1.3.2 使用 Microsoft Visual C++ 6.0 编写 C 语言程序

Microsoft Visual C++ 6.0,简称 VC 6.0,是微软推出的一款 C++编译器,将“高级语言”翻译为“机器语言”的程序。Visual C++是一个功能强大的可视化软件开发工具,自 1993 年 Microsoft 公司推出 Visual C++1.0 后,随着其新版本的不断问世,Visual C++已成为专业程序员进行软件开发的首选工具。

Visual C++ 6.0 不仅是一个 C++编译器,而且是一个基于 Windows 操作系统的可视化集成开发环境(Integrated Development Environment, IDE)。Visual C++ 6.0 由许多组件组成,包括编辑器、调试器以及程序向导 AppWizard、类向导 Class Wizard 等开发工具。这些组件通过一个名为 Developer Studio 的组件集成为和谐的开发环境。

目前最新版 C++编译器集合在 Microsoft Visual Studio 2017 软件里面,包含 C++(支持面向对象程序设计的大型语言)、Visual Basic(支持名面向对象程序设计的语言)、Java(适用网络的语言)、C#、J#、.net。其中,VC 开发环境的版本已经升级至 Microsoft Visual C++ 2017,对 C++的支持更加全面稳定。

Visual C++不仅支持 C++语言,同时也支持 C 语言,所以我们可以采用 VC 6.0 开发 C

程序。

利用 Visual C++ 6.0 集成环境开发 C 程序的过程如下。

1. 启动 Visual C++ 6.0 (见图 1.2)

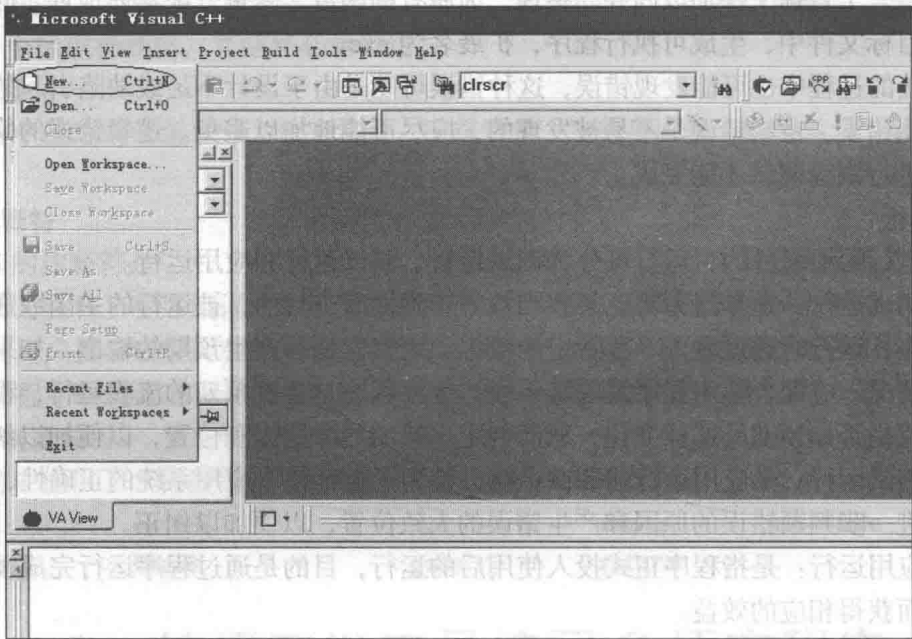


图 1.2 Visual C++ 6.0 界面

方法一：单击 Windows “开始”菜单 → “程序” → “Microsoft Visual Studio 6.0” → “Microsoft Visual C++ 6.0” 命令；

方法二：在桌面上双击 Visual C++ 6.0 图标



2. 新建 C 语言源程序文件，如图 1.3 所示

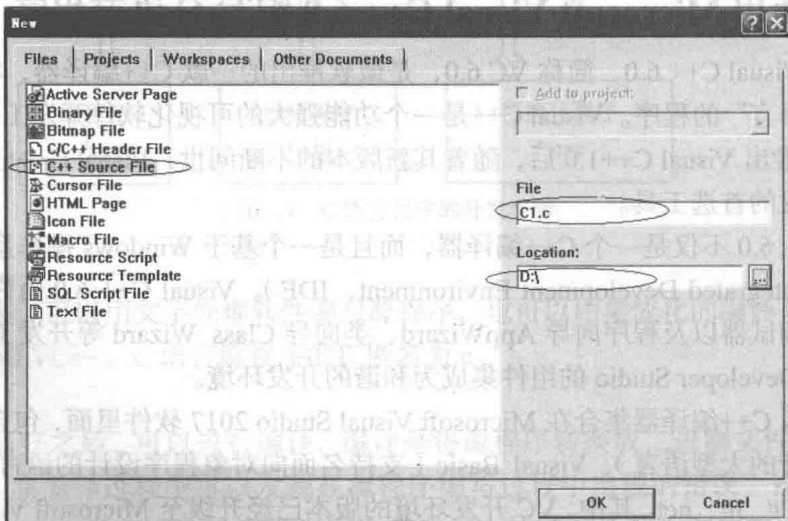


图 1.3 新建 C 语言源程序文件

(1) 单击 Visual C++ 6.0 菜单栏中的“文件 (File)” → “新建 (New)”命令，打开“新