

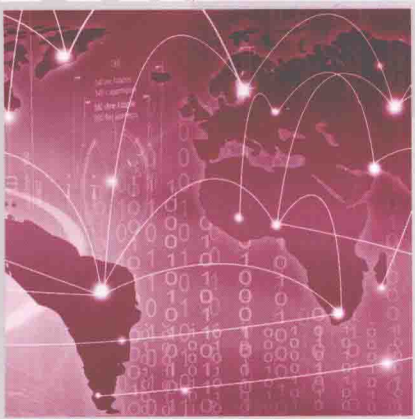


高职高专新课程体系规划教材 ·

计算机系列

C程序设计与 项目开发

普清民◎主编



- 以项目为载体，以任务驱动的方法组织教学
- 语言通俗易懂，实例丰富、有趣
- 理论与实践有机结合
- 配套教学资源：教学课件、习题答案和案例程序

课件下载地址：www.tup.com.cn

清华大学出版社



高职高专新课程体系规划教材 ·

计算机系列

C 程序设计与 项目开发

常州大学图书馆
藏书章

普清民◎主 编

丁 犇 何业军 李 勇 常润梅◎副主编



清华大学出版社

北 京

内 容 简 介

本书根据高等职业教育人才培养目标的要求,结合作者多年的教学经验,按照项目导向、任务驱动、教学做一体化的思路组织编写。所设置的项目有顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组的应用、指针的应用、函数与编译预处理、构造数据类型、文件操作等。每个项目根据工作过程又划分成若干个任务。教材的内容体系包括任务说明、任务实施、任务拓展、相关知识、典型例题、单元习题等,采用Visual C++ 6.0作为C语言的开发环境。

本书可作为应用型本科和高职高专院校电子、计算机、通信、自动化等专业C语言程序设计课程的教材,也可作为成人教育、职工培训和相关工程技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C程序设计与项目开发/普清民主编. —北京:清华大学出版社,2015
高职高专新课程体系规划教材·计算机系列
ISBN 978-7-302-39993-3

I. ①C… II. ①普… III. ①C语言-程序设计-高等职业教育-教材 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第077240号

责任编辑:苏明芳

封面设计:刘 超

版式设计:郑 坤

责任校对:王 颖

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62788951-223

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:13.25

字 数:322千字

版 次:2015年7月第1版

印 次:2015年7月第1次印刷

印 数:1~2500

定 价:28.00元

前言

Foreword



C 语言程序设计是学习程序设计的入门课程,也是高职高专院校很多工科类专业必修的技术基础课程。通过对本课程的学习,使学生掌握 C 语言程序设计的基本知识、程序结构、算法和结构化程序设计的方法,培养学生的编程思维、程序设计能力以及分析和解决问题的能力。

本书按照“项目导向、任务驱动”的教学原则,结合 C 语言程序设计自身的特点,用通俗易懂的方法组织编写。本书具有以下特点:

(1) 打破以知识传授为主要特征的传统课程教学模式,以项目为载体,由任务驱动组织课程内容。每个任务的完成包括“任务说明→任务实施(任务功能、编程思路、源程序、调试运行)→任务拓展”等步骤,分步完成。

(2) 在教材编写的过程中,力求把复杂的问题简单化。用通俗易懂的语言讲授课程内容,用简单易上手的实例降低学习的难度,强调实用性和趣味性。让学生在解决问题的过程中获得成就感,增强自信心。

(3) 加强能力训练。C 语言程序设计课程传统的教学方法是理论→实验→理论的教学模式。通过变革,将课程内容设计为“教、学、做一体化”的模式,在实训室或职场完成课程内容的教学训练任务,将理论和实践融为一体。

(4) 教学效果评价采用过程评价与结果评价相结合的方式,重点评价学生的职业能力。

(5) 书中所有的源程序均在 Visual C++ 6.0 集成开发环境下调试通过,所有实例的输出结果均为截屏获得,确保了源程序的正确性。

(6) 每个项目都安排有一定的编程练习,便于学生训练和提高。

本书配有课件、习题答案和案例程序,便于教师教学和读者自学使用。

本书由中山职业技术学院的普清民、何业军、李勇,佛山职业技术学院的丁犇老师,以及内蒙古电子信息职业技术学院的常润梅老师合作编写。参编人员均为教学一线的专职教师,有多年承担本课程教学和项目开发的经验。在本书编写的过程中参阅了大量的书籍和资料,在此向这些书籍和资料的原作者表示衷心的感谢!

由于编者水平有限,书中难免有不足之处,恳请广大读者批评指正!

编者

目 录

Contents



项目 1 初识 C 语言	1
任务 1.1 C 语言的发展历程与主要特点	1
任务 1.2 C 语言程序的结构	3
任务 1.3 程序的算法	5
1.3.1 算法的概念	6
1.3.2 算法的描述	6
任务 1.4 C 语言程序的运行环境	9
1.4.1 C 语言程序的实现过程	9
1.4.2 Visual C++ 6.0 集成开发环境简介	10
思考与练习	16
项目 2 顺序结构程序设计	18
任务 2.1 数据的类型与运算	18
2.1.1 数据类型	18
2.1.2 常量与变量	19
2.1.3 运算符与表达式	22
任务 2.2 C 语句	28
任务 2.3 数据的输入 / 输出	30
2.3.1 格式输入 / 输出函数	31
2.3.2 字符输入 / 输出函数	33
思考与练习	36
项目 3 选择结构程序设计	40
任务 3.1 用 if 语句实现的选择结构	40
任务 3.2 用 switch 语句实现的多分支结构	48
思考与练习	52
项目 4 循环结构程序设计	55
任务 4.1 用 while 语句实现的循环结构	55
任务 4.2 用 do···while 语句实现的循环结构	58



Note

任务 4.3 用 for 语句实现的循环结构.....	61
任务 4.4 循环嵌套.....	65
任务 4.5 break、continue 语句的应用.....	68
思考与练习.....	71
项目 5 数组的应用.....	74
任务 5.1 一维数组.....	74
任务 5.2 二维数组.....	79
任务 5.3 字符数组.....	83
5.3.1 字符数组的定义、初始化与引用.....	83
5.3.2 字符数组的输入 / 输出.....	85
5.3.3 字符串处理函数.....	86
思考与练习.....	91
项目 6 指针的应用.....	94
任务 6.1 指针与指针变量.....	94
任务 6.2 通过指针访问一维数组.....	98
任务 6.3 通过指针访问二维数组.....	102
任务 6.4 通过指针访问字符串.....	105
6.4.1 指向字符串的指针变量.....	105
6.4.2 字符指针变量与字符数组的比较.....	107
思考与练习.....	110
项目 7 函数与编译预处理.....	113
任务 7.1 函数的定义与调用.....	113
任务 7.2 函数间的参数传递.....	120
任务 7.3 变量的作用域和存储类别.....	126
7.3.1 变量的作用域.....	126
7.3.2 变量的存储类别.....	127
任务 7.4 内部函数和外部函数.....	132
任务 7.5 编译预处理.....	136
7.5.1 宏定义.....	136
7.5.2 文件包含.....	138
7.5.3 条件编译.....	139
思考与练习.....	142
项目 8 构造数据类型.....	146
任务 8.1 结构体类型与结构体变量.....	146
8.1.1 结构体类型的定义.....	147



8.1.2 结构体变量的应用.....	147
任务 8.2 结构体数组.....	152
任务 8.3 结构体指针.....	157
任务 8.4 共用体类型.....	163
任务 8.5 枚举类型.....	166
任务 8.6 用 typedef 定义类型.....	169
思考与练习.....	172
项目 9 文件操作	177
任务 9.1 文件的打开与关闭.....	177
任务 9.2 文件的读写	181
任务 9.3 文件的定位和检测	186
思考与练习.....	190
参考文献	192
附录 A 常用字符与 ASCII 码对照表	193
附录 B C 语言关键字	194
附录 C 运算符的优先级和结合性.....	195
附录 D 常用的库函数.....	197



项目 1

初识 C 语言

C 语言是一种用途广泛、功能强大、使用灵活的编程语言。它既具有高级语言接近人类自然语言便于学习和开发的特点，又具有低级语言能直接操作计算机硬件的特点；既可用于编写应用软件，又可用于编写系统软件，同时它还具有效率高、可移植性强等特点。本项目主要是了解 C 语言的发展历史、特点、程序结构和开发工具等。

本项目的主要内容

- (1) C 语言的发展历程与主要特点。
- (2) C 语言程序的结构。
- (3) 程序的算法。
- (4) C 语言程序的运行环境。

重点与难点

- (1) C 语言程序的结构。
- (2) C 语言集成开发环境的使用方法。

任务 1.1 C 语言的发展历程与主要特点

1. 计算机语言

人与计算机交流信息使用的语言称为计算机语言或程序设计语言。人们就是利用计算机语言来编写程序，然后交给计算机去执行的。计算机语言可分为机器语言、汇编语言和高级语言 3 大类。

早期人们使用的程序设计语言是用二进制代码 0 和 1 表示的，不必通过任何翻译处理，就能被计算机硬件识别和执行，这种语言称为机器语言。它是一种低级语言，与人类的自然语言差别较大，不便于记忆、阅读和使用。在机器语言的基础上，人们设计出了汇编语言，它将机器语言用人们便于记忆和阅读的助记符表示，如 ADD、SUB、MOV 等。汇编语言必须经过汇编程序软件翻译为机器语言后，才能被计算机识别与执行。汇编语言和机器语言都是直接面向计算机硬件编程的，依赖于硬件的结构，因此二者被称为低级语言。

随着计算机技术的发展，出现了高级语言。它是一种与硬件结构和指令系统无关、表达方式接近于人类自然语言和数学语言的程序设计语言。人们用它来编写计算机程序，比使用机器语言和汇编语言显然要方便得多。用高级语言编写的程序称为“源程序”，它也



需要翻译成机器语言才能被计算机识别和执行。C 语言是一种高级语言，它使用接近于人类的思维和表达方式来描述问题、编写计算机程序。

2. C 语言的发展历程

C 语言的发展历程如表 1-1 所示。

表 1-1 C 语言的发展历程

时 间	C 语言发展概述
1967 年	英国剑桥大学的 Martin Richards 推出了没有数据类型的 BCPL (Basic Combined Programming Language) 语言
1970 年	美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，设计出了简单且接近硬件的 B 语言（取 BCPL 的第一个字母）
1972、1973 年	美国贝尔实验室的 D.M.Ritchie 在 B 语言基础上设计出了 C 语言。C 语言既保持了 BCPL 和 B 语言的优点（精炼，接近硬件），又克服了它们的缺点（过于简单，无数据类型）
1973 年	最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的。1973 年，Ken Thompson 和 D.M.Ritchie 两人合作把 UNIX 的 90% 以上用 C 语言改写
1978 年	Brain W.Kernighan 和 Dennis M.Ritchie（合称 K&R）合著了影响深远的名著 <i>The C Programming Language</i> ，这本书介绍的 C 语言被称为后来广泛使用的 C 语言版本的基础，它实际上是第一个 C 语言标准
1983 年	美国国家标准协会（ANSI）成立了一个委员会，根据 C 语言问世以来各种版本对 C 语言的发展和扩充，制定了第一个 C 语言标准草案（83 ANSIC）
1989 年	美国国家标准协会（ANSI）公布了一个完整的 C 语言标准——ANSI X3.159-1989（又称 ANSI C 或 C89）
1990 年	国际标准化组织 ISO（International Standard Organization）接收 C89 作为国际标准 ISO/IEC 9899:1990（简称 C90），它和 ANSI 的 C89 基本上是相同的
1995 年	ISO 对 C90 做了一些修订，即 1995 基准增补 1（ISO/IEC 9899/AMD1:1995）
1999 年	ISO 又对 C 语言标准进行修订，在基本保留原来 C 语言特性的基础上，针对应用的需要增加了一些功能，尤其是 C++ 中的一些功能，命名为 ISO/IEC 9899:1999。2001 年与 2004 年先后进行了 2 次技术修正。ISO/IEC 9899:1999 及其技术修正被称为 C99

3. C 语言的主要特点

与其他语言相比，C 语言主要有以下特点。

- 语言简洁、紧凑，使用方便、灵活。C 语言共有 32 个关键字（见附录 B）、9 种控制语句，程序书写自由，C 语言的关键字主要用小写字母表示。
- 运算符丰富，表达式类型多样化。C 语言有 34 种运算符（见附录 C），括号、赋值、强制类型转换等都作为运算符处理。
- 数据类型丰富。C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。C99 又扩充了复数浮点类型、超长整型（long long int）、布尔类型（bool）等。
- 模块化结构。具有结构化的控制语句，如 if...else 语句、while 语句、do...while 语句、switch 语句、for 语句等。用函数作为程序的基本单位，便于实现程序的模块化。
- 语法限制不太严格，程序设计自由度大。例如对数组下标越界不做检查，对变量



的类型使用比较灵活。因此,编程过程中不能完全依赖编译查错,程序员需要养成严谨的工作习惯。

- 允许直接访问物理地址,可以直接对硬件进行操作。
- 生成的目标代码质量高,程序执行效率高。
- 用C语言编写的程序可移植性好,基本上不做修改就能用于各种型号的计算机和操作系统。



Note

任务 1.2 C 语言程序的结构

1. 几个简单的 C 语言程序

为了使读者对C语言有一个初步的认识,下面通过几个简单实例来分析C语言程序的结构。

【例 1-1】一个最简单的C语言程序,在屏幕上显示“Hello, world!”。

```
#include <stdio.h>          /* 包含系统头文件 */
void main()                 /* 主函数 */
{
    printf("Hello, world! \n"); // 输出信息
}
```

程序运行的结果为:

```
Hello, world!
```

这是世界上第一个C程序,整个程序只有一个主函数main()。主函数中只有一条语句,该语句放在一对大括号内,其功能是使用系统提供的标准库函数printf()在屏幕上输出信息“Hello,world!”。#include <stdio.h>是一条编译预处理命令,表明该程序使用了文件stdio.h中定义的函数printf()。“/*”与“*/”之间或“//”之后的文字都表示注释,是对程序所做的说明,在程序编译和运行时不起作用。“\n”是换行符,即输出信息“Hello, world!”后,将当前光标位置移到下一行开头。

【例 1-2】求两个整数之和。

```
#include <stdio.h>          /* 包含系统头文件 */
void main()                 /* 主函数 */
{
    int a,b,sum;             /* 定义 3 个整型变量 */
    a=25;                   /* 给变量 a 赋值 */
    b=32;                   /* 给变量 b 赋值 */
    sum=a+b;                /* 对 a 和 b 求和,并赋值给 sum */
    printf("sum=%d \n",sum); /* 输出 a 和 b 之和 */
}
```



程序运行的结果为：

```
sum=57
```



Note

本程序的功能是求两个整数 a 和 b 之和。整个程序只有一个主函数 `main()`，主函数的大括号“{}”内有对 a 、 b 、 sum 变量的定义（声明）和 4 条语句。“`sum=a+b`”是 C 语言中的算术表达式（后加半角分号即构成语句），表示将变量 a 和 b 的值相加然后赋给变量 sum 。“`printf("sum=%d \n",sum);`”是调用标准库函数 `printf()` 输出 sum 的值。“`%d`”为格式控制符，表示将 sum 的值以十进制形式输出。

【例 1-3】输入长方形的长和宽，求长方形的面积。

```
#include <stdio.h>                                /* 包含头文件 */
float area(float a,float b);                       /* 函数原型声明 */
void main()
{
    float a,b,s;                                  /* 定义 3 个实型变量 */
    printf(" 输入两个实型数据: ");               /* 提示输入 a 和 b 的值 */
    scanf("%f,%f",&a,&b);                          /* 从键盘上输入变量 a 和 b 的值 */
    s=area(a,b);                                  /* 调用 area() 函数 */
    printf("s=%.2fn",s);                          /* 输出 s 的值 */
}
float area(float x,float y)                        /* 定义 area() 函数，形参 x、y 为实型 */
{
    float z;                                       /* area() 函数中的声明部分，定义变量 z */
    z=x*y;                                         /* 求长方形的面积 */
    return(z);                                    /* 将 z 的值返回到调用处 */
}
```

程序运行的结果为：

```
输入两个实型数据：4.5,6.2<Enter>
s=27.90
```

本程序包括两个函数，即主函数 `main()` 和被调用函数 `area()`。`area()` 函数的作用是求 x 与 y 的乘积，并将结果赋给变量 z ，`return` 语句将 z 的值返回给主调函数 `main()`。`main()` 函数中的 `scanf()` 是系统提供的标准库函数，其功能是从键盘输入 a 和 b 的值。`&a` 和 `&b` 中“`&`”的含义是取地址，即将两个数值分别输入到变量 a 和 b 的地址所标明的单元中，也就是将两个输入的数值分别赋给变量 a 和 b 。

2. C 语言程序的结构

一个 C 语言程序通常包括 3 大部分，即编译预处理、全局声明和函数定义。

(1) 编译预处理。编译预处理包括宏定义（由 `#define` 实现）、文件包含（由



#include 实现)和条件编译。

(2) 全局声明。全局声明包括全局变量声明和函数声明。在函数外部定义的变量称为全局变量,在函数内部定义的变量称为局部变量。全局变量声明就是指在函数外部进行变量声明(或定义)。C语言程序规定,当一个函数调用另一个函数时,被调用函数必须先声明。被调用函数的声明既可以在调用函数内部声明,也可以在调用函数之前进行声明。

(3) 函数定义。函数是C语言程序的基本单位,即C语言程序是由若干个函数构成的。一个C语言程序可以由一个主函数 main() 组成,也可以由一个主函数 main() 和若干个其他函数组成,即一个C语言程序必须有一个而且只能有一个主函数 main()。主函数 main() 可以通过直接书写语句或调用其他函数来完成任务。被调用的函数可以是系统提供的库函数(如 printf() 函数、scanf() 函数),也可以是用户根据需要自己编写的函数(如 max() 函数)。当程序中出现多个函数时,主函数 main() 可以位于其他函数之前(此时被调用函数必须先声明),也可以位于其他函数之后。但无论主函数 main() 处于什么位置,C语言程序总是从主函数 main() 开始执行,主函数 main() 执行完程序即结束。由于C语言程序的全部工作是由各个函数来完成的,因此编写C语言程序的实质就是编写一个个函数。

一个函数由两部分组成,即函数首部和函数体。

① 函数首部。函数首部即为函数的第一行,包括函数类型、函数名、函数参数类型、函数参数名等。一个函数名后面必须跟一对圆括号“()”,这也是函数的标志。函数参数可以省略,如 main()。

② 函数体,即函数首部下面的大括号“{}”。如果一个函数内有多个大括号,则最外层的一对大括号“{}”是函数体的范围。函数体包括声明部分和语句部分。对变量的定义和所调用函数的声明属于声明部分。执行部分则由若干条语句组成,指定在函数中所进行的操作。在实际应用中,有时不需要声明部分,甚至既无声明部分也无语句部分。

(4) C语言程序的书写格式。C语言程序的书写格式自由,可以在一行内写几条语句,也可以把一条语句写在多行。每条语句和变量定义(声明)、函数声明都必须以分号“;”结束,但预处理命令、函数首部和的大括号“{}”之后不允许添加分号“;”。

(5) 注释。注释不是C语言程序所必需的,只是为了便于阅读而设置的。C语言程序的注释有以下两种形式。

① 以“//”开始的单行注释。这种注释可以单独占一行,也可以出现在一行中其他内容的右侧。

② /* ... */。“/*”和“*/”必须成对使用,且“/”和“*”以及“*”和“/”之间不能有空格,否则都出错。这种注释可以包含多行内容。

(6) 实际编程中,一般用适当的缩进格式书写C语言程序,以提高程序的可读性和清晰性。

任务 1.3 程序的算法

编写程序时,一般需要考虑以下两方面的问题。

(1) 数据结构。数据结构是对数据的描述,即在程序中指定数据的类型和组织形式。



Note

(2) 算法。算法是对操作的描述,即在程序中指定计算机处理数据对象时的操作步骤。数据是操作对象,操作的目的是对数据进行加工处理。作为程序设计人员,必须认真考虑和设计数据结构及操作步骤(即算法)。瑞士著名的计算机科学家沃思(Nikiklaus Wirth)将程序归纳为:

程序 = 数据结构 + 算法

数据是程序处理的对象,而算法是程序的灵魂。算法是解决“做什么”和“怎么做”的问题。程序中的操作语句,实际上就是算法的体现。

实际上,一个过程化的程序除了以上两个主要因素之外,还应当采用结构化的程序设计方法来设计程序,并且用一种计算机语言来描述。因此,算法、数据结构、程序设计方法和计算机语言 4 个方面是一个程序设计人员所应具备的知识。

1.3.1 算法的概念

广义地说,为解决一个问题而采取的方法和步骤就称为算法。

计算机程序的算法可分为两大类:数值运算算法和非数值运算算法。数值运算的目的是求数值解,例如求平方根、求圆柱体的体积等,都属于数值运算范畴。非数值运算包括的面非常广,最常见的是用于事务管理领域,例如学生档案管理、职工工资管理、图书管理等。

数值运算往往有现成的模型,可采用数值分析的方法,因此对数值运算算法的研究比较深入,算法比较成熟。对各种数值运算都有比较成熟的算法可供选用,如计算机程序系统中的数学程序库,C 语言编译系统中的头文件 math.h 中就包含了许多数学运算算法。

非数值运算的种类繁多,要求各异,难以做到全部都有现成的答案,只有一些典型的非数值运算算法(如排序、查找搜索算法)有现成、成熟的算法可供选用。大多问题需要程序设计者参照已有的类似算法思路,自行设计相关问题的算法。

1.3.2 算法的描述

算法可以采用自然语言、流程图、N-S 图等来描述。

1. 用自然语言描述算法

自然语言就是人们日常生活中使用的语言,如汉语、英语或其他语言。用自然语言描述算法通俗易懂,但文字冗长,容易出现歧义。一般情况下,不建议使用自然语言描述算法。

2. 用流程图描述算法

流程图是用一些图形框来表示各种操作,直观形象,易于理解。美国国家标准化协会 ANSI 规定了一些常用的流程图符号,已为世界各国程序工作者普遍采用,流程图中各种符号及名称如图 1-1 所示。

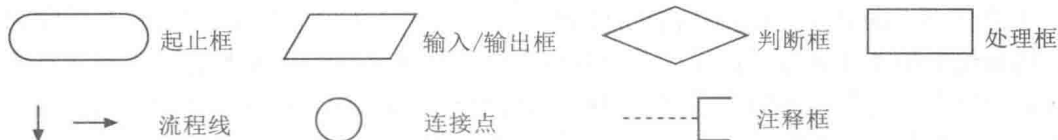


图 1-1 流程图中各种符号及名称

结构化程序设计有 3 种基本结构，有关它的流程图描述如下。

(1) 顺序结构。如图 1-2 所示，虚线框内是一个顺序结构。在执行完 A 框所指定的操作后，接着执行 B 框所指定的操作，即 A 和 B 两个框是顺序执行的。

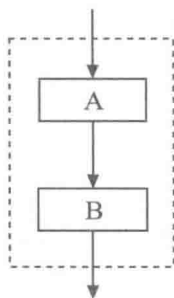
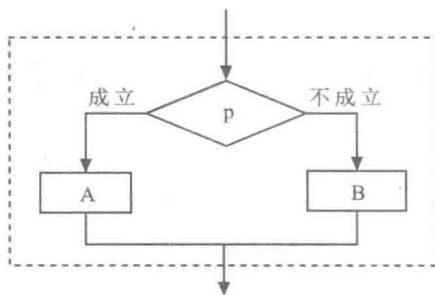
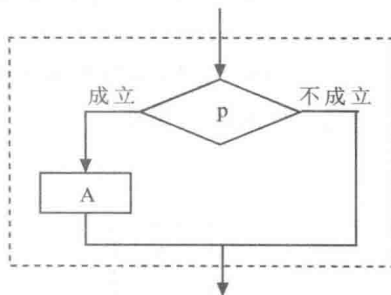


图 1-2 顺序结构流程图

(2) 选择结构。选择结构又称为分支结构，如图 1-3 (a) 所示。虚线框内是一个分支结构，此结构中必须包含一个判断框，且 A 框和 B 框是相互独立的。当给定的条件 p 成立时，选择执行 A 框；否则，选择执行 B 框。A 框或 B 框可以省略，即不执行任何操作，如图 1-3 (b) 中省略了 B 框。



(a) 双分支选择结构



(b) 单分支选择结构

图 1-3 选择结构流程图

(3) 循环结构。循环结构又称重复结构，即重复执行某一部分操作。它可分为当型循环结构和直到型循环结构两大类。

① 当型 (while 型) 循环结构。当型循环结构如图 1-4 (a) 所示。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，若成立继续执行 A 框操作。如此反复执行 A 框操作，直至条件 p 不成立退出循环结构。

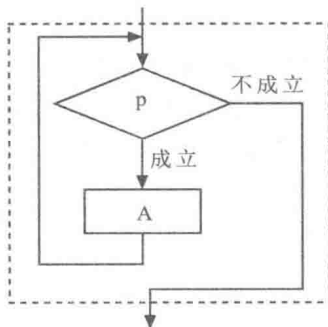




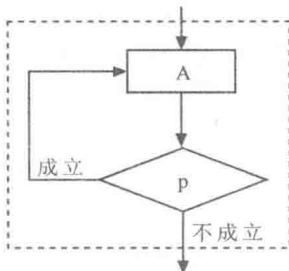
② 直到型 (until 型) 循环结构。直到循环结构如图 1-4 (b) 所示。先执行 A 框操作, 然后判断给定的条件 p 是否成立。若条件 p 成立, 则重复执行 A 框操作, 然后再判断条件 p 是否成立。若条件 p 仍然成立, 则再一次执行 A 框操作, 如此反复执行, 直至给定条件 p 不成立退出循环结构。



Note



(a) 当型循环结构



(b) 直到型循环结构

图 1-4 循环结构流程图

3. 用 N-S 图描述算法

1973 年, 美国学者 I.Nassi 和 B.Shneiderman 提出了一种新的流程图形式。在这种流程图中, 完全去掉了带箭头的流程线, 全部算法都是在一个矩形框内, 在该框内还包含其他一些从属于该矩形框的框, 或者说该框由一些基本的框组成, 这种流程图称为 N-S 图。

下面用 N-S 图描述结构化程序的 3 种基本结构。

(1) 顺序结构。顺序结构 N-S 流程图如图 1-5 所示, 由 A 框和 B 框组成一个顺序结构。

(2) 选择结构。选择结构 N-S 流程图如图 1-6 所示, 当条件 p 成立时执行 A 操作, 当条件 p 不成立时执行 B 操作。

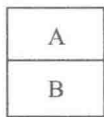


图 1-5 顺序结构的 N-S 流程图

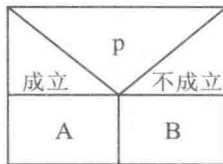


图 1-6 选择结构的 N-S 流程图

(3) 循环结构。当型循环结构 N-S 流程图如图 1-7 (a) 所示, 当条件 p 成立时反复执行 A 操作, 直至条件 p 不成立为止。直到型循环结构 N-S 流程图如图 1-7 (b) 所示, 先执行 A 的操作, 再判断条件 p, 只要条件 p 成立反复执行 A 操作, 直到条件 p 不成立为止。

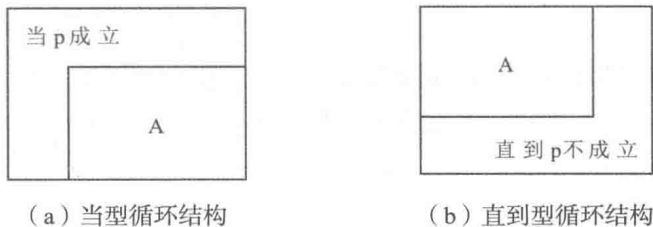


图 1-7 循环结构的 N-S 流程图

N-S 图的最大特点是将一个模块所完成的功能全部放在一个矩形框中, 这更符合结构化程序设计思想。

算法设计好后必须通过计算机语言编写程序代码实现其功能, 所以用计算机语言来实现算法是算法设计的最终目的。

任务 1.4 C 语言程序的运行环境

1.4.1 C 语言程序的实现过程

按照 C 语言的语法规则编写的程序称为 C 源程序。C 源程序是不能直接被计算机识别和执行的, 必须经编译系统把 C 源程序翻译成二进制形式的目标程序, 然后再将该程序与系统的函数库以及其他目标程序连接起来, 形成可执行的目标程序。即 C 源程序必须经过编辑、编译、连接和执行 4 个环节, 才能输出结果。其操作流程如图 1-8 所示。

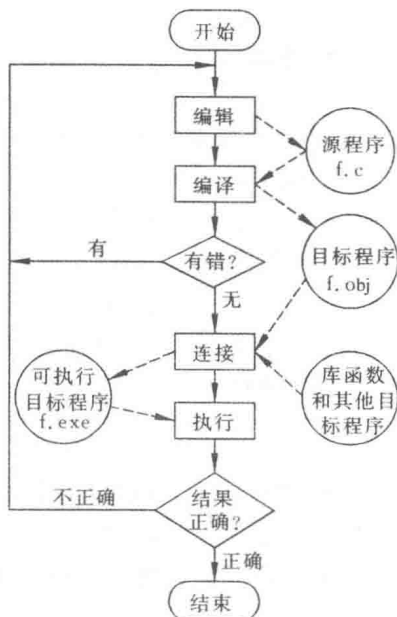


图 1-8 C 语言程序操作流程





1. 编辑

编辑是在一定的环境下进行程序输入和修改的过程。C 源程序可以事先在纸上写好,然后输入到编辑环境中。也可以直接在编辑环境中进行输入和修改,编辑环境有 Turbo C 和 Visual C++ 6.0 等。

2. 编译

编辑好的 C 源程序不能直接被计算机所执行,必须经过编译,生成计算机能够识别的机器代码。编译是将 C 源程序转换成二进制代码的过程,即目标代码文件,其扩展名为 .obj。

在编译过程中将发现在 C 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,因此这种错误我们形象地叫做语法错误,这种错误是易于修改的。

3. 连接

编译后生产的目标代码还不能直接在计算机上执行,因为在目标代码中还没有为函数、变量等安排具体的地址,因此也称其为浮动程序。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 .obj 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,这种错误一般不易发现,我们称这种错误为逻辑错误。逻辑错误是不易被发现的,应尽可能地加以避免。

4. 执行

将生成的可执行文件运行,在屏幕上显示运行的结果。如果运行正确,程序编写成功,如果不正确,将提示错误。开发者可以修改后再运行,如此循环往复,直到得到正确的结果。

1.4.2 Visual C++ 6.0 集成开发环境简介

对 C 语言程序的调试是通过 C 语言编译系统来完成的。早期主要采用 Turbo C,现在多数采用 Visual C++ 6.0 集成开发环境。Visual C++ 6.0 集成开发环境不仅支持 C++ 程序的编译和运行,而且也支持 C 语言程序的编译和运行。Visual C++ 6.0 集成开发环境约定,当源程序文件的扩展名为 .cpp 时,编译和运行 C++ 程序,而当源程序文件的扩展名为 .c 时,编译和运行 C 程序。启动 Visual C++ 6.0 后,一般先新建一个工程,再向工程中添加文件,然后编辑、编译、连接、执行程序,最后查看程序执行的结果。如果在创建文件之前,没有先建工程,系统在编译时会提示是否要创建工程。

1. 启动 Visual C++ 6.0

Visual C++ 6.0 有绿色版、企业版等不同的版本,根据用户的需要,选择一个版本安装即可。安装 Visual C++ 6.0 绿色中文版后,选择“开始”|“程序”|Visual C++ 6.0|Visual C++ 6.0 命令,可启动 Visual C++ 6.0 集成开发环境,启动后的界面如图 1-9 所示。

