

高等院校·高职高专电子商务系列教材

算法与程序设计

SUANFAYUCHENGXUSHEJI

南志红 主编

中国物资出版社

图书在版编目(CIP)数据

算法与程序设计/南志红主编.—北京:中国物资出版社,2002.11

ISBN 7-5047-1871-8

I.算... II.南... III.①电子计算机—算法理论 ②程序设计—理论
IV.①TP301.6 ②TP311

中国版本图书馆 CIP 数据核字(2002)第 065072 号

责任编辑 沈兴龙

封面设计 彩奇风

责任印制 沈兴龙

责任校对 马思奇

中国物资出版社出版发行

网址: <http://www.clph.com.cn>

社址:北京市西城区月坛北街 25 号

电话:(010)68392746 邮政编码:100834

全国新华书店经销

中国农业出版社印刷厂印刷

开本:787×1092mm 1/16 印张:18.5 字数:446千字

2002年11月第1版 2002年11月第1次印刷

书号:ISBN 7-5047-1871-8/G·0424

印数:0001—5000册

定价:28.00元

(图书出现印装质量问题,本社负责调换)

新理念 新概念

《现代市场营销系列教材》 编审委员会 《电子商务系列教材》

- 主任委员** 哈尔滨商业大学教授 刘北林
杭州商学院教授 丁正中
- 副主任委员** 中国物资出版社总编辑 李舒东
哈尔滨商业大学教授 张守文
北京工商大学教授 唐立军
山西财经大学教授 马尚才
中国物流信息中心高级工程师 戴定一
- 委 员** 北京工商大学 李永波
北京工商大学 杨树新
北京工商大学 万江洪
北京工商大学 王 曼
北京工商大学 李书友
北京工商大学 张海燕
北京工商大学 高丽华
山西财经大学 贾 伟
山西财经大学 李淑琴
山西财经大学 杨慧刚
山西财经大学 南志红
哈尔滨商业大学 周 游
哈尔滨商业大学 韩 平
哈尔滨商业大学 白以恩
哈尔滨商业大学 项义军
哈尔滨商业大学 赵 炎
杭州商学院 顾春梅
杭州商学院 盛 亚
杭州商学院 胡永铨

杭州商学院 杨坚红
杭州商学院 江 辛
杭州商学院 易开刚
杭州商业职业技术学院 胡燕燕
温州职业技术学院 俞吉兴
温州职业技术学院 徐育裴
安徽商贸职业技术学院 方光罗
山东商业职业技术学院 匡奕珍
吉林建工学院职业技术学院 申荣季
武汉大学继续教育学院
(省贸科技学校教学站) 吴元佑

总 策 划 沈兴龙

前 言

随着我国社会主义市场经济体制的建立、世界经济一体化进程的加快和科学技术的飞速发展,尤其是我国加入WTO后,现代企业如何在汹涌的经济大潮中求生存与发展,在全球化的世界经济中占有一席之地,已成为我国经济学术界、企业界的一个十分令人关注的问题。提升现代企业营销理念,吸收现代市场营销、营销战略、管理方法,充分运用现代营销技术、现代物流技术、电子商务技术、网络营销、绿色营销、企业营销战略管理等,是提高中国现代企业竞争能力和国际市场占有份额的有力保证。在充分了解中国企业发展和人才需求的基础上,根据教育部颁发的普通高等院校、高职高专专业目录要求,以及面向二十一世纪课程教材的要求,中国物资出版社策划、组织编写了《现代市场营销系列教材》、《电子商务系列教材》。这两套系列教材由北京工商大学、山西财经大学、哈尔滨商业大学、杭州商学院和有关职业技术学院、企业界70名专家、教授联合编写。教材编写队伍庞大,许多编写人员在学术界、教育界、企业界具有较高的知名度。两套系列教材编写体例力求完整性、科学性和合理性,内容充分体现时代性和超前性,充分运用最新理论研究成果、新技术和成功案例,为未来企业家、现职营销人员和企业经营管理者提供了坚实的理论知识,并为掌握制订营销方案、营销策略、运用现代电子商务技术的方法和技巧奠定了基础。《现代市场营销系列教材》、《电子商务系列教材》具有鲜明的时代性、可读性和可操作性,是当代教材(图书)市场上不可多得的系列教材,可作为普通高等院校、高职高专的市场营销、企业经营管理、电子商务等经济类专业和计算机专业教材,也可作为企业人员培训、各层次成人教育教材,还可作为广大企业员工必备的自学参考读物。

《现代市场营销系列教材》 编审委员会
《电子商务系列教材》

编写说明

由于C语言自身的灵活性和C语言具有其他高级语言所不具备的“低层”功能,可以方便地访问硬件系统资源,使得C语言成为一种在计算机软件设计和计算机程序设计教学中备受欢迎的程序设计语言。

本书使用C语言为工具介绍程序的设计方法和编程技术、技巧。全书共分十四章。第一章介绍结构化程序设计方法和算法的基本概念、表示方法。第二章C语言的数据类型、运算符和表达式,介绍C语言的语法知识。第三、四章介绍判断与选择的分支算法、多路分支算法和嵌套分支算法、循环程序(有确定循环次数的循环、无确定循环次数的循环和循环嵌套)的算法设计。第五章介绍一维数组、二维数组和字符数组,并通过实例介绍如何使用这些数组实现相应的算法。第六章介绍使用C语言进行文本查询编辑的算法。第七章介绍几种程序设计中常用算法(递推、倒推、穷举)的实现及其应用。第八章介绍函数调用的内部实现机制,函数声明与函数定义,全局变量、静态局部变量和局部变量之间的区别,递归函数。第九章介绍指针的概念,使用指针给函数传递参数的方法,指针、数组和字符串之间的联系,字符串数组,命令行参数的用法和含义,函数指针的概念与使用。第十章介绍文件的概念、文件的打开、关闭、读写、定位等操作。第十一章介绍结构声明、结构变量定义与访问结构成员的方法,结构作为参数传递与返回结构的函数方法,链表结构的各项基本操作,联合数据类型的含义,各种位操作的方法。第十二章介绍文件的使用,多文件结构和编译预处理的概念。第十三章介绍在西文环境下汉字的显示方法。第十四章介绍C语言调用ROM BIOS的各种服务,实现对硬件的管理的方法。本书理论与实际相结合,既有理论介绍,又有具体的应用示例,叙述深入浅出,便于学以致用。书中的全部程序在Turbo C 2.0下调试通过。

本书是集体合作的结果,本书由南志红任主编,具体编写分工:第一章至第五章由郭鲜风编写,第六章由尹耀兰编写,第八、九、十一、十二章由温雅丽编写,第七、十、十三、十四章由南志红编写。全书由南志红统稿,山西财经大学信息管理学院贾伟院长审阅了全稿并修改定稿。本书可作为普通高等院校、高职高专的教材,也可作为各层次成人教育、企业人员培训教材,以及广大电子计算机爱好者的自学读物。

在本书的编写过程中得到了多方帮助,在此一并表示感谢。

由于编者学术水平所限,编写时间仓促,书中难免存在一些不足之处,恳请广大读者对本书提出批评、指正。

编者

目 录

第一章 程序设计与算法	(1)
第一节 结构化程序设计概述	(1)
第二节 算法	(9)
第二章 C 语言的数据类型、运算符和表达式	(21)
第一节 常量和变量	(21)
第二节 C 语言的数据类型	(24)
第三节 运算符和表达式	(29)
第四节 位运算	(35)
第五节 数据类型转换	(38)
第六节 基本的输入输出操作	(40)
第七节 简单的 C 程序设计	(43)
第八节 Turbo C 上机操作简介	(46)
第三章 基本算法(I)	(54)
第一节 程序分支算法	(54)
第二节 多路分支算法	(58)
第三节 嵌套分支算法	(62)
第四节 语法总结	(69)
第四章 基本算法(II)	(76)
第一节 有确定循环次数的循环程序算法设计	(76)
第二节 无确定循环次数的循环程序算法设计	(80)
第三节 循环嵌套	(83)
第四节 语法分析	(91)
第五章 数组	(101)
第一节 一维数组	(101)
第二节 二维数组	(105)
第三节 字符数组	(109)
第六章 数据查询和编辑算法	(123)
第一节 查询算法	(123)
第二节 数据编辑算法	(127)
第七章 递推、穷举、倒推算法	(135)
第一节 递推算法	(135)
第二节 穷举算法	(137)
第三节 倒推算法	(140)
第八章 函数与变量	(144)
第一节 函数	(144)

第二节 变量	(155)
第九章 指针	(169)
第一节 指针变量	(169)
第二节 数组与指针	(176)
第三节 字符串与指针	(180)
第四节 函数与指针	(183)
第五节 指针数组与指向数组的指针	(189)
第十章 文件操作	(201)
第一节 文件的打开、关闭	(201)
第二节 文件的读写	(204)
第三节 文件的定位	(210)
第四节 错误处理	(211)
第十一章 结构体与共用体	(215)
第一节 结构体	(215)
第二节 结构体与指针	(223)
第三节 动态存储分配与链表	(228)
第四节 位段	(235)
第五节 共用体	(237)
第十二章 编译预处理	(243)
第一节 宏定义	(243)
第二节 文件包含	(247)
第三节 条件编译	(249)
第四节 多文件结构	(250)
第十三章 C 程序中的汉字显示处理	(254)
第一节 汉字编码	(254)
第二节 16 点阵汉字的输入输出方法	(255)
第三节 24 点阵汉字的修饰和放大处理	(258)
第四节 创建应用系统自用汉字字库	(263)
第十四章 微机键盘、光标、鼠标、视频操作程序设计	(267)
第一节 Turb C 系统对 BIOS 中断功能的引用	(267)
第二节 光标操作算法	(269)
第三节 键盘操作算法	(270)
第四节 视频操作算法	(274)
第五节 鼠标操作算法	(277)
第六节 扬声器操作算法	(282)
参考文献	(286)

第一章 程序设计与算法

程序设计需要采用一定的程序设计方法,在程序设计中,算法占有极其重要的地位,算法是程序的灵魂。本章主要介绍:结构化程序设计方法,结构化程序的基本结构,结构化程序的特征与风格;算法和算法的特性,算法的描述,算法的 C 语言表示,模块化程序设计的基本准则。

著名计算机科学家 N·沃思提出:

$$\text{程序} = \text{数据结构} + \text{算法}$$

这一公式简洁明了地概括出了一个程序所包括的两方面的内容。

事实上,程序设计的目的是把人工求解问题的过程转换为计算机语言程序,让计算机执行该程序,以获得问题的解或指出问题无解。对于任何一个问题,无论是科学计算方面的问题还是信息管理方面的问题,其求解过程实质上最终都归结为对数据进行加工处理。因此,在程序中应对所处理的数据做出描述,指定数据的类型和数据的组织形式,即数据结构,同时还应给出对操作的描述,给出操作步骤,亦即算法。

程序设计的实质就是采用结构化程序设计方法针对所要求解的问题描述其数据结构和算法,然后用某种计算机语言(如 C 语言)表达上述算法。算法是程序的灵魂,在程序设计中,算法占有极其重要的地位。因此,在这章重点介绍有关算法的基本知识,以便为后面各章的学习奠定基础。同时,程序设计需要采用合适的方法,这章将介绍结构化程序设计方法。

第一节 结构化程序设计概述

一、结构化程序设计的发展

计算机程序和程序设计方法是与计算机同时产生的。自从计算机诞生至今,程序设计的方法有了很大的变化。这种改变的基本原因是为了适应不断增长的程序复杂性。

最初的程序设计依赖于计算机硬件,使用机器指令代码或与之密切相关的汇编语言。因此,程序设计是十分繁杂困难的,编制的程序往往只针对某一机器。另外,由于当时的计算机内存小,运行速度慢,于是设计人员总是想方设法使自己设计出来的程序构思精巧,运行速度快,尽量少占用内存,甚至有些人为了少用一些内存,挖空心思地想出种种绝招,加之早期的程序设计大都是设计人员以独立的方式进行的,所以,程序的设计风格迥异。同样一个算题,其设计思路五花八门。因此,早期的程序很像手工制造的工艺品,没有统一的标准。

随着计算机技术的发展,应用领域的不断扩大,程序设计的复杂程度越来越高,一个大型软件再也不是一、二个人能独立完成的事,需要几十人或者几百人共同完成。这样,程序设计人员不能再像以前那样“我行我素”,而应该有一个统一的要求与标准,能够适应大型软件设计中的“工作量大,头绪繁杂,容易出错”的新局面。程序的研制与生产,要求

能像工业产品那样标准化、规范化、工程化。设计出来的程序的优劣标准也从原先的“短(程序短)、巧(使用种种技巧)、快(运行速度快)”,转到要求程序“结构清晰,容易阅读,容易修改,容易验证”,亦即希望程序的结构要好这样一个新的标准。因此,计算机科学界的知名学者 E·W·Dijkstra 于 20 世纪 60 年代后期两次建议取消计算机高级语言中的 GOTO 语句,由此引起一场争论。这场争论实质是一场关于程序设计方法的争论。通过争论,人们认识到,关键不在于 GOTO 语句的使用与否,简单地取消 GOTO 语句并不是解决问题的好办法,而是需要创立一种新的程序设计思想、方法和风格,来提高软件的生产率,降低软件的生产成本和维护的费用。于是结构化程序设计就诞生了。它是一整套关于程序设计的准则,保证程序具有合理的结构,使程序易于理解和便于维护,保证程序的正确性。

目前,程序设计方法有两大类,一个是面向过程的结构化程序设计方法,另一个是面向对象的程序设计方法。面向对象的程序设计是为适应不断增长的程序复杂性而发明的一种新的程序设计方法。这一技术利用软件对象来进行软件开发,在强调软件组件的重用方面,这一技术同标准的工程设计规律有更多的相同之处。但是,面向对象技术并没有抛弃传统的做法,它利用传统的方法来模拟建立对象。对象是在传统的数据结构中加入了相关的过程,因而赋予了对象以动作。

本书基于结构化程序设计思想,采用“自顶向下,逐步求精”方法,进行问题分析和程序设计。

二、结构化程序设计方法

结构化程序设计方法是基于模块化的方法,其基本思想是:

1. 分层设计(自顶向下,逐步求精)

对实际问题进行分析和编程时,先从总体出发,把一个大问题或较复杂问题分解为若干个相对独立的子问题,对每个子问题可以再细化为若干个低一层的子问题,直到子问题便于在计算机上实现,这就是自顶向下、逐步求精的解题方法。

2. 模块化

在计算机上实现各个子问题的求解,就是设计相应功能的子程序,即功能模块。这些模块,功能独立且专一,便于用相关算法描述。当用 C 语言来表达算法时,就是功能独立的 C 函数。

3. 模块内采用基本控制结构编程

每个模块有一个入口一个出口。在模块内采用顺序结构、选择结构和循环结构三种基本结构来编程。下面用这种方法求解一个问题,并给出分析求解的过程。以下例子仅用来说明结构化程序设计方法,读者不必深究程序细节,只需总体理解即可。具体的 C 语言程序设计会在以后各章中详细介绍。

例 1-1 设计一个程序,对输入的一行正文,先分离出其中的英文单词,再统计包含元音的单词个数和包含四个不同元音的单词个数。

例如,若输入的文本行是:program main(input, output);应取出四个英文单词,其中包含元音的单词是四个,包含四个不同元音的单词为 0 个。

解题思路:输入文本到 buf 中,从 buf 中分离出一个英文单词时,先确定单词首字母,然后顺序取字母且放入数组 word 内,遇到非字母字符时,即为一个单词。重复上述操作,直到 buf 中文本被取完。若数组 word 内共放进 na 个单词,再测试每个单词内是否包含元

音和是否包含四个不同元音,测试时,用 nb 计数包含元音的单词数,用 nc 计数包含四个不同元音的单词数。

假设输入的文本是小写的字母,buf 和 word 的说明为:

```
# define N 50
# define M 15
char buf[80],word[N][M];
int na,nb,nc;
```

1. 总体框架分析

- (1) 将待测文本输入 buf。
- (2) 从 buf 中分解出 na 个单词存入 word。
- (3) 检测 word 中单词,统计出包含元音的单词 nb 个、包含四个不同元音的单词 nc 个。

2. 细化总体框架

- (1) 从 buf 中分解出 na 个单词的操作可细化为:

```
令 na = 0, pos = 0
while buf[pos]不是结尾标志
    { 搜索 buf,确定第 na 个单词首字母位置 pos;
      依据 pos,从 buf 中取出第 na 个单词存入 word;
    }
```

- (2) 测试 word 中含元音单词和含四个不同元音单词的操作可细化为:

```
for( i = 1; i <= na; i++ )
{
    测试第 word[i]个单词是否包含元音,若是则 nb 计数一次;
}
for( i = 1; i <= na; i++ )
{
    测试第 word[i]个单词是否包含四个不同元音,若是则 nc 计数一次;
}
```

3. 模块化

函数 ScanPos()确定 buf 中某个单词首字母位置 pos,或者文本已分析完。

```
ScanPos( char buf[],int *pos)
{
    while( ! ( buf[ *pos] >= 'a' && buf[ *pos] <= 'z') && buf[ *pos] != '\0')
        (*pos)++;
}
```

函数 GetWord()依据 pos 从 buf 中取字符拼第 na 个单词,且存入 word。

```
GetWord( char buf[],int *pos)
{ int j;
  j = 0; na++;
  while( buf[ *pos] >= 'a' && buf[ *pos] <= 'z')
      { word[na][j] = buf[ *pos];
```

```

        j++;(*pos)++;
    }
}

```

函数 VowelWord() 对 word 中的 na 个单词逐个测试, 当包含元音时则用 nb 计数。

```

VowelWord()
{ char tag[5];
  int i,j,k,flag;
  for( i=1;i<=na;i++)
  { flag=0;
    for( j=0;j<m;j++)
    { strcpy( tag,"aeiou");
      for(k=0;tag[k]!='\0';k++)
        if( word[i][j]==tag[k])
          { flag=1;
            goto next;}
    }
    next:if(flag)nb++;
  }
  printf("带元音的单词为%d个\n",nb);
}

```

函数 FourVowel() 对 word 中的 na 个单词逐个检查, 仅当包含四个不同元音时 nc 计数。

```

FourVowel()
{ char tag[5];
  int i,j,count;
  for( i=1;i<=na;i++)
  { strcpy(tag,"aeiou");count=0;
    for( j=0;word[i][j]!='\0';j++)
      if( FindDelete( tag,word[i][j])) count++;
    if( count==4) nc++;
  }
  printf("带四个不同元音的单词是%d个\n",nc);
}

```

函数 FindDelete() 的功能: 若当前 word[i][j] 中是元音, 则将它从 tag 中清除, 并返回删除标志值 flag。

```

FindDelete( char tag[],char ch)
{ int i,flag;
  flag=0;
  for( i=0;tag[i]!='\0';i++)
    if( tag[i]==ch)
      { tag[i]='';flag=1;}
  return( flag);
}

```

用主函数 main()组织调用上述模块。

```
# include < stdio.h>
# include < string.h>
# include < stdlib.h>
# define N 50
# define M 15
char word[N][M];
int na,nb,nc;
DispWord(0
    { int i;
      for( i=1;i <= na;i + + )
        printf( "%18s \n",word[ i] );
    }
EnterText( char * buf)
    {
      gets(buf);
    }
main()
    { char buf[80];
      int i,option,pos = 0;
      do{
        printf("1_读待测试的文本 2_读字符拼单词 3_查看所拼单词 \n");
        printf("4_检索带元音单词 5_带四个不同元音的单词 6_结束 \n");
        scanf("%d",&option);
        switch( option)
            { case 1: EnterText(buf);
              break;
              case 2: na = 0;
                while( buf[pos] != '\0')
                    { ScanPos(buf,&pos);
                      if(buf[pos] == '\0')break;
                      GetWord( buf,&pos);
                    }break;
              case 3: DispWord();
                break;
              case 4: nb = 0;
                VowelWord();
                Break;
              Case 5: nc = 0;
                FourVowel();
                Break;
            }
        }while(option != 6);
    }
```

三、结构化程序的基本结构

一个结构化程序,总是由一些基本控制结构,按照一定的逻辑规则而有序地组织在一起的。这些基本结构为:顺序结构、选择结构、循环结构,并且实现单入口单出口。结构化程序设计,使程序的控制流程线性化,就是说程序动态执行的顺序符合静态书写结构,增强了程序可读性,方便调试和错误排除。

同样,表述问题的“计算逻辑”,或者说表述解决问题的方法和步骤,即描述算法时,也用这三种基本结构。这三种基本结构是表示一个良好算法的基本单元。这一点将在下一节介绍算法时看到。

(一) 顺序结构

如图 1-1 和图 1-2 所示。程序执行按照书写的先后顺序,自上而下地逐条执行,这种控制结构即为顺序结构。其中,S1、S2 和 S3 可以是一个语句,也可以是一个基本结构或一个模块。

顺序结构是最简单的一种基本结构。

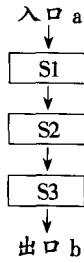


图 1-1

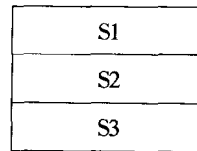


图 1-2

(二) 选择结构

如图 1-3 和图 1-4 所示。根据给定的条件 C 是否成立来选择执行 S1 或 S2。这种控制结构即为选择结构。其中 S1 或 S2 可以有一个是空的,即不执行任何操作。T 表示条件成立,C 的值为真;F 表示条件不成立,C 的值为假。

选择结构也称为分支结构,它用于解决需要分支或分档处理的问题。

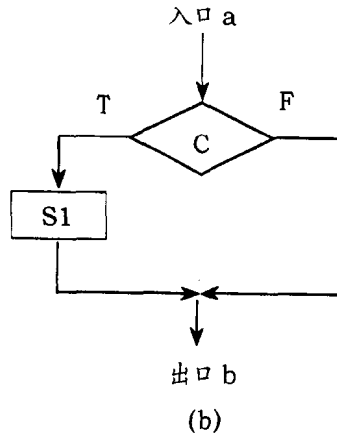
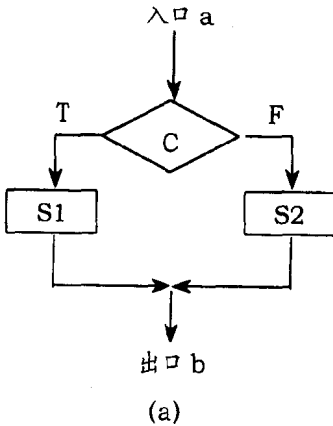


图 1-3



图 1-4

(三) 循环结构

如图 1-5 和图 1-6 所示。根据给定条件 C 成立与否决定是否重复执行 S, 这种控制结构即为循环结构。其中, (a) 图表示当型循环, 当条件成立即 C 的值为真(T)时, 重复执行 S, 当条件不成立即 C 的值为假(T)时, 结束循环; (b) 图表示直到型循环, 执行 S, 直到条件成立即 C 的值为真才结束循环。

循环结构用于解决需要重复计算或重复处理的问题。

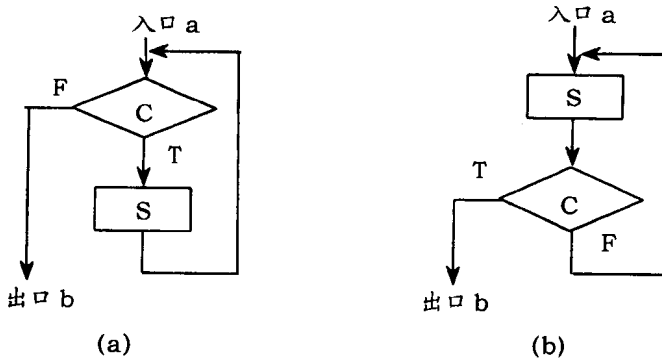


图 1-5

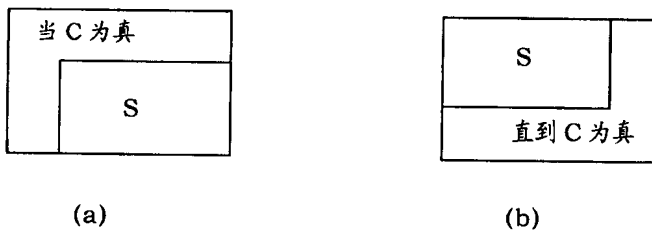


图 1-6

可以看到, 上述三种基本结构, 具有以下三个共同特点。

1. 只有一个入口, 只有一个出口。这里的入口与出口指的是整个基本结构的入口和出口, 即上面各图中的入口 a 和出口 b, 不要与菱形判断框有两个出口相混淆。
 2. 没有“死语句”。每个结构内的任何一部分都有机会被执行到, 亦即对结构内每一个语句或模块而言, 都有一条从入口到出口的路径通过。
 3. 没有“死循环”, 即循环次数是有限的, 不存在永远执行不完的循环。
- 具有以上特点的结构称为“良结构”, 结构化程序是由一些良结构单元构成的程序。

在对实际问题做结构化程序设计时,上述三种基本结构常常是联合使用的。由基本结构所构成的程序是结构化程序,它不存在无规律的转向,只在本基本结构内才允许存在分支和向前或向后的跳转。由以上三种基本结构顺序组成的结构化程序可以解决任何复杂的问题。

(四) 结构化程序的特征与风格

通过前面的介绍可以得出一个结构化程序应具有以下特征:

(1) 任何一个程序单位(例如一个函数)都只能由有限个顺序、选择和循环三种基本结构所组成;

(2) 任何一个程序单位都由若干程序模块组成;

(3) 每个程序模块都只能有一个入口和一个出口;

(4) 一个基本结构可看作是一个程序块,由若干程序块按顺序排列所构成的结构也视为一个程序块。所以,一个结构化程序实质上是由有限个顺序、选择和循环三种基本结构经排列、嵌套而成。

程序风格是对程序结构的限制和要求,也是开发程序所遵守的规则。程序风格包含程序书写规范和程序的结构规则。

结构化程序的书写规范要保证各基本结构之间的排列和嵌套关系清晰。具体方法如下:

(1) 程序书写层次化。为了使程序层次清楚,应当使程序写成锯齿形,使处于同一层次的语句从同一列开始;

(2) 一行一句。尽管许多语言允许一行写多句,但我们推荐一行一句的风格;

(3) 标识符的命名应当见名知意;

(4) 适当地使用注释;

结构化程序的结构规则应付合如前所述的结构化程序的特征。此外还应注意以下两点:

(1) 要保证结构的完整性,不能出现基本结构之间的交叉。循环结构与循环结构、选择结构与选择结构、循环与选择结构的交叉都是不准许的,如图 1-7 所示。(a)图所示的结构符合完整性要求,而(b)图的结构不符合完整性要求;

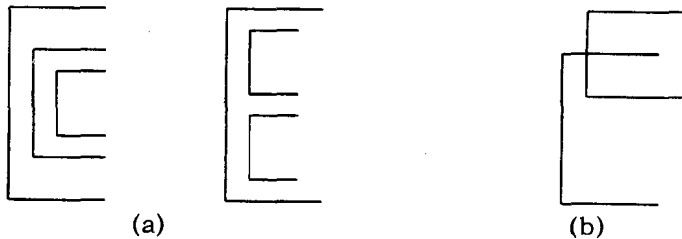


图 1-7

(2) 要保证操作上的整体性。一个基本结构是一个完整操作单位,只能从入口进入,出口退出,不能从其外部进入内部,也不能从内部终止或跳至外部,即要整体执行。

五、C 语言与结构化程序设计

C 语言是目前国际上广泛流行的一种通用的结构化程序设计语言,它不仅是开发系

统软件的理想工具,而且也是开发应用软件的理想程序设计语言。C语言具有如下特点。

1. C语言是介于高级语言和汇编语言之间的一类语言,它比其他高级语言更接近硬件系统。它既具有像汇编语言那样直接访问硬件的功能(如它能直接对内存进行位操作),又具有一般高级语言所具有的面向用户的优点。

2. C语言是一种结构化的程序设计语言,它具有顺序、选择和循环三种基本结构,使程序设计人员便于使用自顶向下逐步求精的结构化程序设计技术。使用它来编写的程序易读。易维护。

3. C语言具有类型丰富和使用灵活的运算符,有多达34种运算符,它不仅具有一般高级语言所具有的算术运算和逻辑运算符,而且还具有位运算和复合运算符。

4. C语言具有丰富的数据类型,它不仅具有如字符型(8bit)、单精度整数(16bit)、双精度整数(32bit)、单精度和双精度实数等基本数据类型;而且还有数组、结构和联合等复合数据类型;有指针数据类型。这使C语言具有很强的数据处理能力。

5. C语言是便于模块化程序设计的程序设计语言,C语言程序是由一个个函数所组成,这种函数结构便于把一个大型程序划分为若干相对独立的模块,模块间通过函数调用来实现相互连接。

6. C语言程序可以通过预编译功能来使用“宏定义”、实现外部文件的读取和合并、实现条件预编译等,提高开发效率。

7. C语言程序具有较高移植性。这是因为C语言不同于其他语言,它不包含依赖硬件的输入/输出机制,其输入/输出功能是由独立于C语言的库函数来实现。这样就使C语言程序本身不依赖于硬件系统,便于在不同的机器系统间移植。

由C语言的特点可知,C语言不仅可移植性好、硬件控制能力高、表达和运算能力强,而且由于它具有结构化的控制语句并以函数作为程序的模块单位,从而使得利用这种语言可以很好地实现结构化程序设计。因此,C语言符合结构化编程风格的要求,是一种理想的结构化程序设计语言。

第二节 算 法

一、算法的基本概念

所谓算法就是处理问题的方法和步骤,确切地说,算法是为了解决特定的问题而要一步一步执行的有穷操作的描述。

例如,求解一元二次方程 $a * x^2 + b * x + c = 0$,其中 a, b, c 是不等于0的实数,可按如下方法步骤求解:

S1:首先计算 $d = b^2 - 4 * a * c$

S2:若 $d > = 0$,则执行(3),否则执行(4)

S3:计算两个实根:

$$x1 = (-b + \text{sqrt}(d)) / (2 * a)$$

$$x2 = (-b - \text{sqrt}(d)) / (2 * a)$$

结束计算。

S4:计算两个复根: