

高等院校“十三五”规划教材

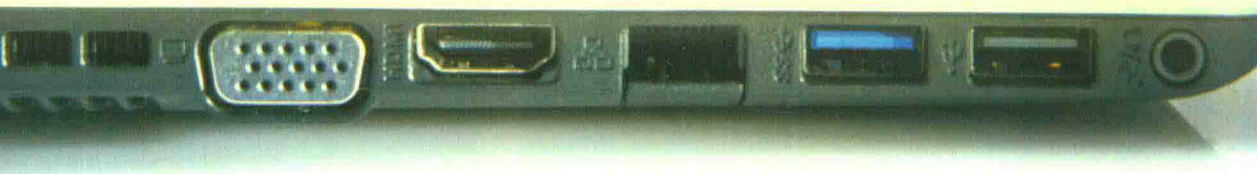
C语言

程序设计案例教程

主 编 高 璐 张 娟



 南京大学出版社



高等院校“十三五”规划教材

C语言

程序设计案例教程

主 审 仲崇俭 裴 杭
主 编 高 璐 张 娟
副主编 徐 鉴 李 强

图书在版编目(CIP)数据

C 语言程序设计案例教程/高璐, 张娟主编. —南京: 南京大学出版社, 2016. 8

高等院校“十三五”规划教材

ISBN 978-7-305-17482-7

I. ①C… II. ①高… ②张… 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(2016)第 196841 号

出版发行 南京大学出版社

社 址 南京市汉口路 22 号 邮编 210093

出 版 人 金鑫荣

丛 书 名 高等院校“十三五”规划教材

书 名 C 语言程序设计案例教程

主 编 高 璐 张 娟

责任编辑 王秉华 吴 汀 编辑热线 025-83595860

照 排 南京理工大学资产经营有限公司

印 刷 丹阳市兴华印刷厂

开 本 787×1092 1/16 印张 17 字数 415 千

版 次 2016 年 8 月第 1 版 2016 年 8 月第 1 次印刷

ISBN 978-7-305-17482-7

定 价 36.00 元

网 址: <http://www.njupco.com>

微信服务号: njyuexue

微信服务号: njyuexue

销售咨询热线: (025)83594756

* 版权所有, 侵权必究

* 凡购买南大版图书, 如有印装质量问题, 请与所购
图书销售部门联系调换

前 言

C 语言是目前国际上广泛流行的一种结构化程序设计语言,它以其功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高等优点深受广大程序开发人员青睐,不仅适合于开发系统软件,而且也是开发应用软件和进行大规模科学计算的常用程序设计语言,因此成为程序设计语言的常青树。

本书第 1、2 章介绍了 C 语言的发展及特点和 C 语言的基本数据类型;第 3~5 章介绍了结构化程序设计的三种基本结构(顺序结构、选择结构和循环结构);第 7 章介绍了函数的定义和使用;第 6、9、10 章介绍了数据的构造类型(包括数组、字符串、结构体和共同体等)和指针类型;第 8、11 章介绍了编译预处理和文件的概念及文件的基本操作;第 12 章介绍了位运算的基本知识。本书的目标是让读者深刻理解和掌握 C 语言的语法规则及编程特点,克服“懂 C 语言,但不会做题,不会编程序”的通病。

本教材具有如下特色:

1. 作为一门专业基础课教材,主线上必须保留或沿袭理工科课程以“学科体系”为线索的指导思想,即在教材内容的知识结构上,依然以概念、定律、定理为线索的编写体系,有别于专业技能课教材。

2. 为了满足“以能力为中心”的培养目标要求,本教材改变传统基础课教材的编写方法,在掌握必须的理论知识的基础上,突破技术的综合应用能力培养,加强实践操作和技能训练,每章都设计了一个小型案例。充分考虑到学生特点和教学规律,精心设计经典有趣案例,使学生的学习重心从“学会知识”扩展到“学会学习、掌握方法和培养能力”上。

3. “算法”是程序设计的灵魂,是程序设计方法的核心内容。本教材突破传统的知识内容归属问题,将“算法设计”始终贯穿、渗透到教材的每一个案例中,从而培养程序设计能力,掌握程序设计方法。

作者根据课程要求,结合 C 语言程序设计的教学经验,并参考了大量相关同类教材,对多年来的讲稿进行了精心的总结、修改和整理,编成本书。在编写过程中,哈尔滨信息工程学院软件学院的全体教师提出了许多宝贵意见和建议,本书的出版得到了编者所在学校的大力支持,在此一并向他们表示衷心的感谢。

本书可作为高等院校 C 语言程序设计课程的教材,也可供准备参加计算机等级考试和考研的读者阅读参考,同时也可作为工程技术人员和计算机爱好者的参考用书。

尽管我们在本教材的编写方面做了很多努力,但由于作者水平所限,不当之处在所难免,恳请各位读者批评指正。

目 录

第1章 C语言概述	1
1.1 C语言的发展及特点	1
1.1.1 C语言的发展	1
1.1.2 C语言的特点	2
1.2 初步认识C语言程序	2
1.2.1 C语言程序的格式	4
1.2.2 C语言程序的构成	4
1.2.3 C语言程序的基本要求	5
1.3 软件开发方法	5
1.3.1 软件开发方法	5
1.3.2 算 法	5
1.3.3 结构化程序设计	6
1.4 C语言程序上机调试步骤和方法	7
1.5 小 结	10
习 题	10
第2章 C语言程序设计的初步知识	12
2.1 C语言的数据类型	12
2.2 标识符	12
2.2.1 系统定义标识符	13
2.2.2 用户定义标识符	13
2.3 常 量	13
2.3.1 整型常量	14
2.3.2 实型常量	14
2.3.3 字符常量	15
2.3.4 字符串常量	16
2.3.5 符号常量	16
2.4 变 量	17
2.4.1 变量名称	17
2.4.2 变量的类型	17
2.4.3 变量的赋值	18

2.5 运算符与表达式	19
2.5.1 算术运算符和算术表达式	19
2.5.2 赋值运算符和赋值表达式	20
2.5.3 关系运算符和关系表达式	21
2.5.4 逻辑运算符和逻辑表达式	21
2.5.5 条件运算符和条件表达式	22
2.5.6 逗号运算符和逗号表达式	23
2.6 自增(自减)运算符及 C 语言运算符的优先级别	23
2.6.1 自增、自减运算符	23
2.6.2 C 语言运算符的优先级与结合性	24
2.7 不同类型数据间的混合运算(包含数据类型转换)	25
2.7.1 自动类型转换	25
2.7.2 强制类型转换	25
2.8 小结	26
习 题	26
第 3 章 顺序结构程序设计	28
3.1 任务 1——将英里转换为公里	28
3.2 C 语句	30
3.3 数据的输出	33
3.3.1 数据输入输出的概念及在 C 语言中的实现	33
3.3.2 字符输出函数(putchar 函数)	34
3.3.3 格式输出函数(sprintf 函数)	35
3.4 数据的输入	38
3.4.1 字符输入函数(getchar 函数)	38
3.4.2 格式输入函数(scanf 函数)	39
3.5 小型案例	43
3.6 小结	46
习 题	47
第 4 章 选择结构程序设计	49
4.1 任务 2——有节约要求的水费问题	49
4.2 if 语句	51
4.2.1 if 语句的三种形式	51
4.2.2 if 语句的嵌套	55
4.3 switch 语句	57
4.4 小型案例	60

4.5 小 结	62
习 题	63
第5章 循环结构程序设计	67
5.1 任务3——公司员工薪水计算	67
5.2 while 语句	69
5.3 do-while 语句	70
5.4 for 语句	73
5.5 循环嵌套	77
5.6 goto 语句以及用 goto 语句构成循环	81
5.7 break 语句和 continue 语句	82
5.7.1 break 语句	82
5.7.2 continue 语句	83
5.8 小型案例	84
5.9 小 结	89
习 题	90
第6章 数 组	94
6.1 任务4——用冒泡法对10个数排序	94
6.2 一维数组	96
6.2.1 一维数组的定义	96
6.2.2 一维数组元素的引用	97
6.2.3 一维数组的初始化	97
6.2.4 一维数组的应用	98
6.3 二维数组	98
6.3.1 二维数组的定义	98
6.3.2 二维数组元素的引用	99
6.3.3 二维数组的初始化	100
6.3.4 二维数组的应用	101
6.4 字符数组	103
6.4.1 字符数组的定义及初始化	103
6.4.2 字符数组元素的引用	104
6.4.3 字符串数组	105
6.4.4 字符数组的输入与输出	107
6.4.5 字符串数组处理函数	108
6.5 小型案例	112
6.6 小 结	115

习 题.....	115
第7章 函 数	118
7.1 任务 5——从身份证号码中间提取出生日期、性别、年龄等信息	118
7.2 函数的概念	122
7.2.1 库函数的使用	122
7.2.2 函数的定义	122
7.2.3 函数的声明	124
7.3 函数的参数和返回值	124
7.3.1 函数的参数	124
7.3.2 函数的返回值	125
7.4 函数的参数传递方式	125
7.4.1 值的传递	126
7.4.2 地址的传递	126
7.5 函数的调用	128
7.5.1 函数的一般调用	128
7.5.2 函数的嵌套调用	128
7.5.3 函数的递归调用	129
7.6 变量的作用域和存储类型	130
7.6.1 变量的作用域	130
7.6.2 变量的存储类型	130
7.7 函数的作用范围	133
7.7.1 内部函数	133
7.7.2 外部函数	133
7.8 小型案例	133
7.9 小 结	135
习 题.....	136
第8章 编译预处理	138
8.1 任务 6——求解某校园圆形花坛的圆周长及圆面积	138
8.2 宏定义	140
8.2.1 无参宏	140
8.2.2 带参宏	140
8.2.3 终止宏定义	141
8.3 文件包含命令	141
8.4 条件编译	143
8.4.1 带 #if、#else 和 #endif 标识的	143

8.4.2 带 #ifdef 标识的	143
8.4.3 带 #ifndef 标识的	144
8.5 小型案例	145
8.6 小 结	146
习 题.....	146
 第9章 指 针	148
9.1 任务7——利用指针变量完成税率的调整并打印前后结果	148
9.2 指针变量	150
9.2.1 指针变量的定义	151
9.2.2 指针变量的引用	151
9.2.3 指针变量的运算	154
9.3 指针与数组	159
9.3.1 指针与一维数组	159
9.3.2 指针与二维数组	161
9.3.3 指针数组	164
9.4 指针与字符串	166
9.4.1 用字符数组存放一个字符串	166
9.4.2 字符指针指向一个字符串	166
9.4.3 字符数组和字符指针变量的区别	169
9.5 指针与函数	171
9.5.1 指向函数的指针	171
9.5.2 指针函数	172
9.5.3 指针变量作为函数的参数	173
9.5.4 数组名作为函数的参数	174
9.5.5 指向函数的指针作为函数的参数	176
9.6 指向指针的指针变量	177
9.7 main()函数的形参和 void 指针	178
9.7.1 指针数组作为 main 函数的形参	178
9.7.2 指向 void 的指针变量	180
9.8 小型案例	180
9.9 小 结	182
习 题.....	182
 第10章 构造数据类型.....	187
10.1 任务8——输出一批学生的基本信息	187
10.2 结构体.....	190

10.2.1 结构体定义	190
10.2.2 结构体变量	193
10.2.3 结构体变量的使用	195
10.3 结构体与函数	196
10.3.1 结构变量与数组结构作为函数的参数	197
10.3.2 结构变量作为函数的返回值	197
10.4 结构体与指针	198
10.4.1 结构体变量指针	198
10.4.2 结构体数组指针	199
10.5 链表	201
10.5.1 链表概述	201
10.5.2 链表的基本操作	202
10.6 共用体	202
10.7 枚举类型	204
10.8 typedef 类型声明	206
10.9 小型案例	207
10.10 小结	208
习题	210
第 11 章 文件	212
11.1 任务 9——磁盘文件信息复制	212
11.2 文件类型指针	216
11.3 文件的基本操作	217
11.3.1 文件的打开	217
11.3.2 文件的关闭	219
11.3.3 文件读函数	220
11.3.4 文件写函数	226
11.4 文件的定位函数	232
11.4.1 rewind() 函数	232
11.4.2 fseek() 函数和随机读写	232
11.5 文件出错检测函数	234
11.5.1 ferror() 函数	234
11.5.2 clearerr() 函数	235
11.6 小型案例	235
11.7 小结	238
习题	239

第 12 章 位运算	242
12.1 位的运算	242
12.1.1 “按位与”运算符(&)	242
12.1.2 “按位或”运算符()	243
12.1.3 “异或”运算符(^)	243
12.1.4 “取反”运算符(~)	244
12.1.5 左移运算符(<<)	245
12.1.6 右移运算符(>>)	245
12.1.7 位运算赋值运算符	246
12.1.8 不同长度的数据进行位运算	246
12.2 位 段	246
12.3 小型案例	249
12.4 小 结	252
习 题	253
附 录	256
参考文献	259

第 1 章 C 语言概述

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种类型的计算机上,从而形成了多种版本的 C 语言。

1.1 C 语言的发展及特点

1.1.1 C 语言的发展

C 语言的发展颇为有趣。它的原型是 ALGOL 60 语言。

1963 年,剑桥大学将 ALGOL 60 语言发展成为 CPL(Combined Programming Language)语言。

1967 年,剑桥大学的 Martin Richards 对 CPL 语言进行了简化,于是产生了 BCPL 语言。

1970 年,美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改,并为它起了一个有趣的名字“B 语言”。意思是将 CPL 语言“煮干”,提炼出它的精华。并且他用 B 语言写了一个 UNIX 操作系统。

1973 年,B 语言也被人“煮”了一下,美国贝尔实验室的 Dennis M. Ritchie 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。

为了使 UNIX 操作系统推广,1977 年,Dennis M. Ritchie 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,Brian W. Kernighan 和 Dennis M. Ritchie 出版了名著《The C Programming Language》,从而使 C 语言成为目前世界上最流行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,使得这些 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。C 语言发展迅速,而且成为最受欢迎的语言之一,主要是因为它具有强大的功能。许多著名的系统软件如 DBASE III PLUS、DBASE IV 都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序,就更能显出 C 语言的优势了,如 PC-DOS、WORDSTAR 等就是用这种方法编写的。

1.1.2 C 语言的特点

1. 简洁紧凑、灵活方便

C 语言一共只有 32 个关键字,9 种控制语句,程序书写自由,主要用小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合了起来。

2. 运算符丰富

C 语言的运算符包含的范围很广泛,共有 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理,从而使 C 语言的运算类型极其丰富,表达式类型多样化。灵活使用各种运算符,可以实现在其他高级语言中难以实现的运算。

3. 数据结构丰富

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,能用来实现各种复杂的数据类型的运算,并引入了指针的概念,使程序效率更高。另外,C 语言具有强大的图形功能,支持多种显示器和驱动器,且计算功能、逻辑判断功能强大。

4. C 语言是结构式语言

结构式语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C 语言是以函数形式提供给用户的,这些函数可方便地调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

5. C 语言语法限制不太严格,程序设计自由度高

一般的高级语言语法检查比较严,能够检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度。

6. C 语言允许直接访问物理地址,可以直接对硬件进行操作

C 语言既具有高级语言的功能,又具有低级语言的许多功能,能够像汇编语言一样对位、字节和地址进行操作,而这三者是计算机最基本的工作单元,可以用来写系统软件。

7. C 语言程序生成代码质量高,程序执行效率高

C 语言程序生成的代码一般只比汇编程序生成的目标代码效率低 10%~20%。

8. C 语言适用范围大,可移植性好

C 语言有一个突出的优点就是适合于多种操作系统,如 DOS、UNIX,也适用于多种机型。

1.2 初步认识 C 语言程序

为了帮助同学们对 C 语言程序形成完整的认识,这里准备了一个较为完整的程序。下面的程序用来演示在 C 语言程序设计中使用的—般程序设计方法和基本要素。编写程序,用来从文件中读取所有的数据,并输出所读取的数的平方。

```
/*
```

```
程序执行时在 D 盘的根目录下创建一个文件取名为 a.dat,  
文件中输入 10 个数"1 3 2 4 5 6 7 8 9 10"
```

程序运行时输入: Demo01.exe d:\\a.dat

```
* /
#include <stdio.h>          /* 包含 stdio.h, malloc.h, stdlib.h 三个头文件 * /
#include <malloc.h>
#include <stdlib.h>

#define N 10                /* 定义一个不带参数的宏 N * /
#define f(x) x * x          /* 定义一个带参数的宏 f(x) * /

int a[N], * b;              /* 定义全局变量: 数组 a 和指针变量 b * /
int input();                /* 对函数 input 作声明 * /
void main(int argc, char * argv[]) /* 主函数 * /
{
    /* 下面是主函数的函数体 * /
    void output();          /* 声明 output 函数 * /
    int n, i;                /* 定义局部变量 n, i * /
    if(argc != 2)            /* 如果程序运行携带了参数, 不携带参数时只有文件名
                                demo01.exe * /
    {
        printf("语法: Demo01 <filename>\n"); /* 输出提示信息 * /
        return;                               /* 程序运行错误, 直接退出程序 * /
    }
    n = input(argv[1], a); /* 调用函数 input * /

    b = (int *) malloc((sizeof(int) * n)); /* 为指针变量 b 申请内存空间 * /
    for(i = 0; i < n; i++)                /* 循环 n-1 次 * /
    {
        b[i] = f(a[i]);                  /* 把数组 a 中每个元素都求出平方值, 送到数组
                                            b 中对应位置 * /
    }
    output(b, n);                        /* 调用函数 output, 输出所有 b 中元素 * /
}
/* 下面函数的功能是把文件中的数据逐个读出, 存放到数组 a 中 * /
int input(char * filename, int a[]) /* input 函数的首部 * /
{
    FILE * fp;                          /* 定义指向文件的指针 fp * /
    int i = 0;                           /* 定义整型变量 i, 用来存放读取出来的整数的个数 * /
    fp = fopen(filename, "r");            /* 调用系统函数 fopen 打开文件 filename * /
```

```

if(fp = NULL)          /* 如果打开文件失败,提示出错并退出程序 * /
{
    printf("\n 无法打开文件 %s",filename); /* 提示错误信息 * /
    exit(0);          /* 调用系统函数 exit,退出程序 * /
}

while(!feof(fp)&& i<N)  /* 当文件读取未结束时继续读取 * /
{
    fscanf(fp, "%d",&a[i + +]);
                                /* 读取文件 fp 中的一个整数放入数组 * /
}
fclose(fp);              /* 关闭文件 * /
return (i);              /* 返回读取到的数据的个数 * /
}

/* 下面函数的功能是输出数组 c 中的 n 个元素 * /
void output(int c[], int n) /* 函数 output 的首部 * /
{
    int i;                /* 定义循环用的整型变量 i * /
    for(i = 0; i<n; i + +) /* 循环 n 次 * /
    {
        printf("%8d",c[i]); /* 输出数组元素 c[i]的值,保留 8 位宽度 * /
        if(i % 10 == 0)     /* 每输出 10 个数据换一行 * /
            printf("\n");
    }
}

```

程序输出结果为:

```

Demo01.exe d:\a.dat <回车>
1  9  4  16  25  36  49  64  81  100

```

1.2.1 C 语言程序的格式

C 语言中格式书写比较自由,一行可以写多条语句,一条语句也可以书写在多行上。C 语言识别大小写字母,如“A”和“a”是两个不同的字母。C 语言语句后必须有分号,只有分号的语句为空语句。为了增强程序的可读性,应该避免在一行中书写多条语句,并使用锯齿形书写程序代码,还可以通过空行来增强可读性。

1.2.2 C 语言程序的构成

C 程序的基本结构是函数,一个或多个 C 函数组成一个 C 程序,若干个 C 语句构成 C

语言函数,若干个基本单词形成C语句。C语言中使用的函数有两类,一类是系统定义的函数,如 printf 和 fclose 等,称为标准库函数,可以直接在程序中使用。另一类是用户自己定义的函数,如 demo01.c 中的 output()函数,必须由用户自己编写源程序代码。

函数的基本格式如下:

```
[函数类型]函数名([函数形参表])    /* 函数首部 */  
{                                     /* 函数体 */  
    [变量定义和声明语句;]  
    可执行语句部分;  
}
```

1.2.3 C语言程序的基本要求

(1) 在整个程序文件中,函数可以出现在任意位置。主函数不一定出现在程序的开始处,但不管主函数位于程序何处,程序总是在主函数中开始,也在主函数中结束。

(2) 每个程序行中的语句数量任意。

(3) 为了对程序进行必要的描述,可以给程序进行说明,说明必须写在/*和*/之内。

1.3 软件开发方法

1.3.1 软件开发方法

20世纪60年代出现了软件危机,其现象表现为软件开发费用和精度失控,软件的可靠性差,生产出来的软件难以维护。为了解决软件危机,在60年代末期提出了软件工程的概念,并在以后不断发展、完善。与此同时,软件研究人员也在不断探索新的软件开发方法,至今已形成8类软件开发方法,主要有Parnas方法、SASA方法、面向数据结构的软件开发方法、问题分析法、面向对象的软件开发方法、可视化开发方法等。

1.3.2 算法

算法定义:算法是规则的有限集合,是为了解决特定问题而规定的一系列操作。

算法的特性:算法必须具备5个基本的特性。

(1) 有限性:有限步骤之内正常结束,不能形成无限循环。

(2) 确定性:算法中间的每个代码行必须有确定含义。

(3) 输入:可以有0个或者多个输入。

(4) 输出:至少有一个输出。

(5) 可行性:原则上能够精确运行,操作可以经过已经实现的基本运算通过有限次执行完成。

算法的表示方法:算法的表示方法有多种,下面介绍算法的流程图表示方法。传统的流程图由如图1-1所示的几种基本框组成。使用这些框和流程线组成的流程图表示算法,形象直观,简单方便。在设计算法的时候对于整理设计思路很有帮助。

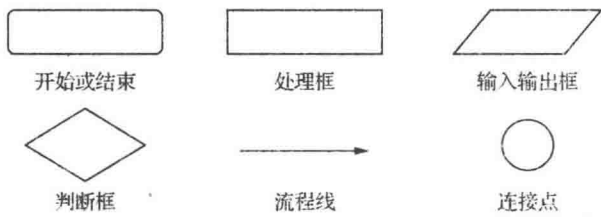


图 1-1 流程图的基本框图

1.3.3 结构化程序设计

结构化程序设计的思想最早是由著名计算机科学家 E. W. Dijkstra 提出的。1966 年, Bohm 和 Jacopin 证明了只用三种基本结构就能实现任何一个入口、一个出口的程序;1977 年, IBM 公司的 Mills 又进一步提出:“程序应该只有一个入口和一个出口”。在长期程序设计的实践中, 结构化程序设计方法不断得以完善, 使之成为开发传统应用领域应用系统的主要方法之一。

结构化程序设计由三种基本结构组成, 分别是: 顺序结构、选择结构和循环结构。

1. 顺序结构

顺序结构是最简单的一种结构, 可以由赋值语句和输入、输出语句构成。当执行由这些语句构成的程序时, 将按照这些语句在程序中的先后顺序逐条执行。流程图如图 1-2(a) 所示。

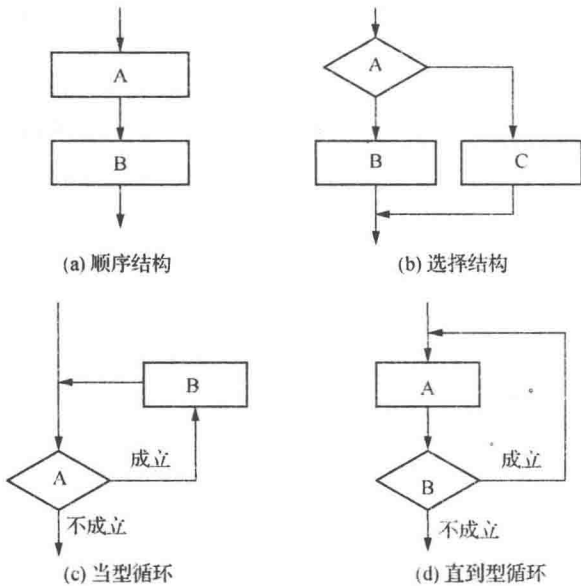


图 1-2 程序的基本结构框图

2. 选择结构

选择结构也称分支结构。当执行该结构中的语句时, 程序根据不同的条件执行不同分支中的语句。如图 1-2(b) 所示。

3. 循环结构

循环结构是指根据各自的条件, 使同一组语句重复执行多次或者一次也不执行。循环