

[普通高等学校计算机基础教育“十二五”规划教材·精品系列]

C语言程序设计基础与 实验指导

C YUYAN CHENGXU SHEJI JICHU YU SHIYAN ZHIDAO

顾玲芳 主 编 ■
杨 娜 副主编 ■

普通高等学校计算机基础教育“十二五”规划教材·精品系列

C 语言程序设计基础与 实验指导

主 编 顾玲芳

副主编 杨 娜

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书以程序设计为主线,以全国计算机等级考试二级 C 语言考查点为线索,以编程应用为驱动,详细讲述了顺序结构、选择结构、循环结构的三种程序设计方法,以及数组、函数、指针、结构体、文件等概念和应用。本书注重程序设计能力的培养,安排了读者熟悉的学生成绩管理系统实验案例,使读者在学习理论知识的同时,明确应该掌握的实践要求,真正掌握程序设计的基本方法,并做到触类旁通、一通百通。

本书针对初学者的特点精心策划,定位准确,概念清晰,体系合理,例题丰富,深入浅出,是学习 C 语言的理想教材。本书适合作为普通高等学校学生的教学用书,也可以作为全国计算机等级考试二级 C 语言的参考用书和对 C 语言程序设计感兴趣者的自学用书。

图书在版编目(CIP)数据

C 语言程序设计基础与实验指导 / 顾玲芳主编. —

北京:中国铁道出版社,2015.2

普通高等学校计算机基础教育“十二五”规划教材.

精品系列

ISBN 978-7-113-19625-7

I. ①C… II. ①顾… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 018874 号

书 名: C 语言程序设计基础与实验指导

作 者: 顾玲芳 主编

策 划: 魏 娜

读者热线: 400-668-0820

责任编辑: 周海燕 徐盼欣

封面设计: 刘 颖

封面制作: 白 雪

责任校对: 汤淑梅

责任印制: 李 佳

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街8号)

网 址: <http://www.51eds.com>

印 刷: 北京新魏印刷厂

版 次: 2015 年 2 月第 1 版 2015 年 2 月第 1 次印刷

开 本: 787 mm×1 092 mm 1/16 印张: 16 字数: 394 千

书 号: ISBN 978-7-113-19625-7

定 价: 36.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659

前言

程序设计是计算机科学教育的第一门专业性课程,它的主要目标首先是理解和掌握一门程序设计语言,其次是读懂别人已经编好的程序,从中体会和启发自己的逻辑思维能力,进而自行编制程序解决实际问题,为在计算机领域中深入学习打下坚实的基础。C语言是使用最广的程序设计语言之一,包含了程序设计需要的主要机制,它的实用性、灵活性及可持续性都是人们公认的。因此,C语言一直是计算机专业程序设计课程的首选语言。

本书分三篇。第1篇为基础知识,按顺序先后安排了10章内容,各章强调解决问题的方法、规律,强调知识点与二级考查点相匹配,重点讲解C语言中重要的具有一般性的语法知识。每个知识点先给出知识点的概念,注重规范,再通过经典的实例将语法知识点化整为零。第2篇为实验指导,在介绍实验一般步骤及实验环境之后,安排了学生成绩管理系统实验案例。先介绍了这个管理系统,然后按第1篇各章先后次序,将大实验拆分成与第1篇各章教学内容相匹配的9个实验,每个实验中包含2~3个实验题。实验题题题相扣,只有完成了前一个实验题,才能展开下一个实验题,直到完成最后一个实验题,即完成整个实验案例。第3篇为习题,包括综合练习题和模拟试题及其参考答案,收集了大量涵盖C语言主要知识点的练习题,是与全国计算机等级考试二级C语言接近的考题,力求重点突出、难点突破。题型包括考试常见的题型,如选择题、填空题、判断题、分析程序写运行结果题等,为学生学习提供指导,帮助读者验证自己对C语言的掌握程度,发现自身的长处和不足。

本书由顾玲芳任主编,编写了第1篇中的第1~3、7、8章和第2篇及附录;由杨娜任副主编,编写了第1篇中的第4~6、9、10章和第3篇。本书涉及的内容有一定的深度和广度,既能满足编程初学者的需要,也能适应能力较强读者深入探讨的愿望,亦可作为全国计算机等级考试二级C语言的参考用书。

本书编者都是应用型本科院校计算机教育一线的教师,他们最了解学生的特点和需求,也最能有针对性进行教学内容、教学方法和教学手段方面的改革。虽然本书凝聚了编者的思考和经验,但由于水平有限,书中难免存在不足及疏漏之处,敬请专家、读者和同行不吝指正。

编者

2014年12月

第1篇 基础知识

第1章 C语言概述与程序设计基础	1
1.1 C语言发展简史	1
1.2 C语言的特点	1
1.3 简单C语言程序举例	2
1.4 程序设计基础	3
1.4.1 算法概述	4
1.4.2 算法的常用表示方法	4
1.5 Visual C++ 6.0 集成开发环境	6
小结	7
习题	7
第2章 数据类型、运算符与表达式	8
2.1 C语言的基本符号	8
2.1.1 标识符	8
2.1.2 数据类型	9
2.1.3 常量	10
2.1.4 变量	13
2.2 运算符和表达式	13
2.2.1 赋值运算符和赋值表达式	14
2.2.2 算术运算符和算术表达式	16
2.2.3 关系运算符和关系表达式	17
2.2.4 逻辑运算符和逻辑表达式	18
2.2.5 逗号运算符和逗号表达式	20
2.2.6 条件运算符和条件表达式	21
2.2.7 自增运算和自减运算	21
2.3 数据类型转换	22
2.3.1 不同数据类型的数据间的混合运算	23
2.3.2 强制类型转换	24
小结	25
习题	25
第3章 顺序结构程序设计	26
3.1 语句	26

3.2 数据的输入和输出	28
3.2.1 简单格式的输入与输出	28
3.2.2 复杂格式的输入与输出	32
3.2.3 字符数据的输入与输出	36
3.3 顺序结构应用举例	37
小结	38
习题	38
第 4 章 选择结构程序设计	40
4.1 if 语句	40
4.1.1 if 语句的三种形式	40
4.1.2 if 语句的嵌套	44
4.2 switch 语句	47
4.3 选择结构应用举例	48
小结	51
习题	52
第 5 章 循环结构程序设计	53
5.1 while 循环语句	53
5.2 do...while 循环语句	54
5.3 for 循环语句	55
5.4 循环嵌套	58
5.5 break 语句与 continue 语句	60
5.5.1 break 语句的使用	60
5.5.2 continue 语句的使用	61
5.6 循环结构应用举例	61
小结	64
习题	65
第 6 章 数组及其使用	66
6.1 一维数组	66
6.1.1 一维数组的定义和引用	66
6.1.2 一维数组的初始化	67
6.1.3 一维数组程序举例	68
6.2 二维数组	70
6.2.1 二维数组的定义和引用	70
6.2.2 二维数组的初始化	70
6.2.3 二维数组程序举例	71
6.3 字符数组	73
6.3.1 字符数组的定义	73

6.3.2	字符数组的初始化	73
6.3.3	字符数组的引用	73
6.3.4	字符数组与字符串	74
6.3.5	字符数组的输入和输出	74
6.3.6	字符串处理函数	75
6.3.7	字符数组应用举例	78
小结		79
习题		80
第 7 章	函数及其应用	81
7.1	函数的基本概念	81
7.1.1	函数的概念	81
7.1.2	函数的定义	83
7.1.3	函数的调用	85
7.1.4	函数的参数	87
7.1.5	函数的返回值	88
7.1.6	函数的原型声明	89
7.2	数组作为函数参数	91
7.2.1	数组元素作为函数参数	91
7.2.2	一维数组作为函数参数	92
7.2.3	二维数组作为函数参数	97
7.3	函数的嵌套调用和递归调用	98
7.3.1	函数的嵌套调用	98
7.3.2	函数的递归调用	99
7.4	变量的作用域和存储方法	103
7.4.1	局部变量和全局变量	103
7.4.2	变量的存储方法	106
7.5	预处理命令	110
7.5.1	宏定义	110
7.5.2	文件包含	113
7.5.3	条件编译	114
7.6	函数应用举例	115
小结		119
习题		120
第 8 章	指针及其应用	122
8.1	地址和指针的概念	122
8.1.1	地址的概念	122
8.1.2	指针的概念	124

8.2	指针变量	124
8.2.1	指针变量的定义	124
8.2.2	指针变量的操作	125
8.2.3	指针变量作为函数参数	128
8.3	指针指向数组	131
8.3.1	指向数组元素的指针	131
8.3.2	指针的运算	132
8.3.3	通过指针引用数组元素	133
8.3.4	指向二维数组元素的指针	136
8.4	指针指向字符串	136
8.4.1	字符指针与字符串指针	137
8.4.2	字符串的表示形式	137
8.4.3	字符指针作为函数参数	138
8.4.4	对使用字符指针变量和字符数组的讨论	139
8.5	指针数组和二级指针	141
8.5.1	指针数组	141
8.5.2	二级指针	143
8.6	动态内存分配与指向它的指针变量	144
8.6.1	什么是内存的动态分配	145
8.6.2	如何实现动态的分配与释放	145
8.7	指针应用举例	147
小结		149
习题		150
第 9 章	结构体与枚举	151
9.1	结构体类型与结构体变量	151
9.1.1	结构体类型的声明	151
9.1.2	结构体变量的定义	152
9.1.3	结构体变量的引用	153
9.2	结构体数组	154
9.2.1	结构体数组的定义	154
9.2.2	结构体数组的初始化	155
9.2.3	结构体数组的引用	155
9.2.4	结构体数组应用举例	155
9.3	结构体指针	156
9.3.1	指向结构体变量的指针	156
9.3.2	指向结构体数组的指针	157
9.4	结构体类型数据在函数间的传递	157

9.4.1 结构体变量作为函数参数	157
9.4.2 结构体指针变量作为函数参数	158
9.5 枚举类型的使用	159
9.5.1 枚举类型和枚举变量	160
9.5.2 枚举类型变量的赋值和使用	160
9.6 类型定义符 typedef.....	161
小结	162
习题	163
第 10 章 文件	164
10.1 文件概述	164
10.1.1 文件的概念	164
10.1.2 文件的分类	164
10.1.3 文件缓冲区	165
10.1.4 文件类型指针	165
10.2 文件的打开与关闭	165
10.2.1 打开文件	166
10.2.2 关闭文件	167
10.3 文件的顺序读写	167
10.3.1 格式化读写函数 fscanf()和 fprintf()	167
10.3.2 字符方式读写函数 fgetc()和 fputc()	168
10.3.3 数据块读写函数 fread()和 fwrite()	170
10.3.4 字符串读写函数 fgets()和 fputs()	172
10.4 文件的定位与随机读写	174
10.4.1 文件指针重定位函数 rewind().....	175
10.4.2 随机读写函数 fseek()	175
10.4.3 其他相关函数	176
小结	176
习题	177

第 2 篇 实 验 指 导

第 1 章 实验一般步骤及实验环境	178
1.1 实验的一般步骤	178
1.2 实验环境 VC++ 6.0 使用指南	179
第 2 章 学生成绩管理系统实验案例	185
2.1 实验案例分析	185
2.2 实验内容及要求	191

第 3 篇 习题及参考答案

综合练习题及参考答案	199
模拟试题及参考答案	221

附 录

附录 A 常用字符及其 ASCII 码对照表	228
附录 B 关键字	229
附录 C 运算符的优先级和结合性	230
附录 D 常用库函数	231
附录 E 常用格式说明符	235
附录 F VC++ 6.0 编译错误信息	237
附录 G 实验案例源程序	240
参考文献	246

第 1 篇 基础知识

第 1 章 C 语言概述与程序设计基础

【本章学习重点】

- (1) 了解 C 语言的发展。
- (2) 理解 C 语言的特点。
- (3) 了解一个 C 源程序的组成。
- (4) 了解什么是算法，理解算法的常用表示方法。

1.1 C 语言发展简史

C 语言是在 20 世纪 70 年代初问世的。1978 年，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 合著了著名的“The C Programming Language”一书。通常简称为 K&R，也有人称之为 K&R 标准，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之一。但是，在 K&R 中并没有定义一个完整的标准 C 语言。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，美国国家标准学会（American National Standards Institute）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，成为现行的 C 语言标准。

目前最流行的 C 语言版本有 Microsoft C、Turbo C 与 AT&TC。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作了一些扩充，使之更加方便、完美。

1.2 C 语言的特点

C 语言发展迅速，是最受欢迎的语言之一，主要因为它具有强大的功能。许多著名的系统软件，如 PC-DOS、DBASE IV 都是由 C 语言编写的。C 语言加上一些汇编语言子程序，更能显示 C 语言的优势。归纳起来，C 语言具有下列特点：

1. C 语言是中级语言

C 语言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。C 语言可以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。

2. 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。

3. C 语言简洁紧凑

C 语言使用方便、灵活。ANSI C 一共只有 32 个关键字, 9 种控制结构, 生成目标代码质量高, 程序执行效率高。语法限制不太严格, 程序设计自由度大, 可移植性好。

4. C 语言功能齐全

C 语言具有各种各样的数据类型, 并引入了指针概念, 可使程序效率更高。另外, C 语言也具有强大的图形功能, 支持多种显示器和驱动器。而且计算功能、逻辑判断功能也比较强大, 可以实现决策目的。

5. C 语言适用范围大

C 语言还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。

1.3 简单 C 语言程序举例

为了说明 C 语言源程序结构的特点, 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 源程序的基本部分和书写格式。

【例 1-1】在屏幕上显示一行信息: “您好, 欢迎来到 C 语言的世界!”。

```
#include <stdio.h>
int main()
{
    printf("您好, 欢迎来到 C 语言的世界! \n");
    return 0;
}
```

(1) main 是主函数的函数名, 由 4 个小写字母组成。每一个 C 源程序都必须有且仅有一个主函数 (main() 函数)。程序从 main() 函数开始执行, main() 函数称为程序的入口。

(2) main 前的 int 是整数类型说明符, int main() 表示主函数的类型是整型, 主函数执行后会产生一个整数。

(3) 程序第 5 行的 return 0; 的作用就是如果程序正常运行, 在结束前将整数 0 作为函数值; 如果执行出现异常, 程序就会中断, 不执行 return 0; , 此时函数值是个非零的整数。

(4) 函数调用语句, printf() 函数的功能是把要输出的内容送到显示器显示。printf() 函数是一个由系统定义的标准函数, 是在头文件 stdio.h 中声明的, 所以需要用 #include 命令把它包含进来, 如 #include <stdio.h>。

【例 1-2】求圆的周长与面积。

```
#include <stdio.h>
int main()
{
    int r;
    double c, s;
    printf("input number:");
    scanf("%d", &r);
    c=2*3.14*r;
    s=3.14*r*r;
    printf("r=%d, c=%f, s=%lf\n", r, c, s);
}
```

```

    return 0;
}

```

- (1) `#include` 称为文件包含命令，扩展名为.h 的文件称为头文件。
- (2) 定义一个整型变量和两个实型变量，以被后面程序使用。
- (3) 显示输入提示信息。
- (4) 从键盘获得一个整数 `r`。
- (5) 求以 `r` 为半径的圆周长与圆面积，分别赋给变量 `c` 与 `s`。
- (6) 显示程序运算结果。
- (7) `main()` 函数结束。

程序的功能是从键盘输入一个数 `r`，求以 `r` 为半径的圆周长与圆面积，然后输出结果。

在 `main()` 函数之前的这一行语句称为预处理命令（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，其意义是把尖括号 `<>` 或引号 `" "` 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，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。在本例中，使用了两个库函数：输入函数 `scanf()` 和输出函数 `printf()`。`scanf()` 和 `printf()` 是标准输入/输出函数，其头文件为 `stdio.h`，所以在主函数前用 `include` 命令包含了 `stdio.h` 文件。

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，在本例中可以删去第 1 行的包含命令 `#include <stdio.h>`。同样，在例 1-1 中使用了 `printf()` 函数，也可省略头文件包含命令。

一般情况下，主函数体分为两部分：一部分为说明部分，或称为声明部分；另一部分为执行部分。每一个说明，每一条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。说明是指变量的类型说明。例 1-1 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。说明部分是 C 源程序结构中很重要的组成部分。例 1-2 中使用了三个变量 `r`、`c` 和 `s`，用来表示半径、周长和面积。半径的类型可以由用户来定，本例中以整型为例，而因为圆周率是个实型数，故 `c` 与 `s` 这两个变量用类型说明符 `double` 来说明。说明部分后的 5 行语句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 1 行是输出语句，调用 `printf()` 函数在显示器上输出输入提示字符串，请操作人员输入半径 `r` 的值。第 2 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 `r` 中。第 3、4 行是计算圆周长与圆面积分别赋给 `c` 与 `s`。第 5 行是用 `printf()` 函数输出变量 `r`、`c`、`s` 的值。

1.4 程序设计基础

一个程序应包括：

(1) 对数据的描述。在程序中要指定数据的类型和数据的组织形式，即数据结构（`data structure`）。

(2) 对操作的描述，即操作步骤，也就是算法（`algorithm`）。

Nikiklaus Wirth 提出的公式：

程序 = 算法 + 数据结构

中国计算机普及和高校计算机基础教育的开拓者谭浩强教授认为：

程序=算法+数据结构+程序设计方法+语言工具和环境

这 4 个方面是一个程序设计人员所应具备的知识。本书的目的是使学生知道怎样编写一个 C 程序，进行编写程序的初步训练，因此，只介绍算法的初步知识。

1.4.1 算法概述

算法是指解题方案准确而完整的描述。算法是一个十分古老的研究课题，人们对于算法的研究已经有数千年的历史。计算机的出现，为这个课题注入了新的青春活力，人们可以将算法编写成程序交给计算机执行，使许多原来认为不可能完成的算法变得实际可行。值得注意的是，算法不同于程序，也不等于计算方法，程序的编制不可能优于算法的设计。

一个算法应该具有以下 5 个重要的特征：

- (1) 有穷性：一个算法必须保证执行有限步之后结束。
- (2) 确切性：算法的每一步骤必须有确切的定义。
- (3) 输入：一个算法有零个或多个输入，以刻画运算对象的初始情况。
- (4) 输出：一个算法有一个或多个输出，以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。
- (5) 可行性：算法原则上能够精确地运行，通过有限次运算后即可完成。

算法由两种基本要素构成：一是对数据对象的运算和操作，二是算法的控制结构。一般的计算机系统中都包括算术运算、逻辑运算和关系运算；算法的控制结构包括顺序结构、选择结构和循环结构。算法设计的基本方法有列举法、归纳法、递推法、递归法、回溯法等。不同的方法间存在着联系，在实际应用中，不同方法通常会交叉使用。

通常评价一个算法，主要从算法的工作量及占用的内存空间这两个方面来度量，即时间复杂度和空间复杂度。对于程序设计人员，必须会设计算法，并根据算法写出程序。

1.4.2 算法的常用表示方法

算法可以用各种描述方法来进行表示，最常用的是伪代码和流程图。伪代码使用介于自然语言和计算机语言之间的文字和符号来描述算法。下面简单介绍流程图的相关知识。

流程图表示算法的好处是：用图形表示流程，直观形象，各种操作一目了然，而且不会产生歧义。正因为如此，流程图成为描述算法的标准工具。图 1-1-1 给出了流程图的 5 种组成元素。

- (1) 起止框：表示算法的开始和结束。
- (2) 输入/输出框：用来表示算法的输入/输出操作。
- (3) 判断框：用来根据给定的条件决定执行几条路径中的哪一条路径。

(4) 处理框：主要用来表示“赋值”“加减乘除”等操作。

(5) 流程线：用来表示程序流程的方向。

1973 年，美国学者提出了一种新型流程图：N-S 图。

C 语言是一种结构化程序设计语言。程序主要由以下三种基本控制结构组成。

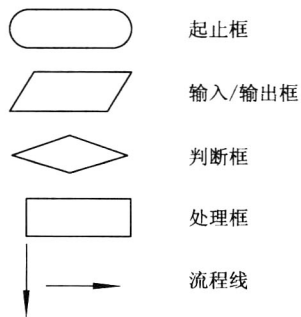


图 1-1-1 流程图符号

1. 顺序结构

顺序结构是程序最基本的控制结构。本篇第2章的赋值语句及第3章的输入/输出语句都可构成顺序结构。当执行由这些语句构成的程序时，将按这些语句在程序中的先后顺序逐条执行，没有分支，没有转移。顺序结构可用图1-1-2所示的流程图表示。

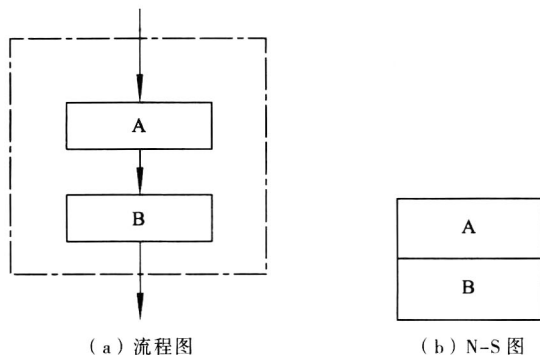


图 1-1-2 顺序结构

2. 选择结构

本篇第4章中将介绍的 if 语句、switch 语句都可构成选择结构。当执行由这些语句构成的程序时，将根据不同的条件去执行不同的分支中的语句。选择结构可用图1-1-3所示的流程图表示。

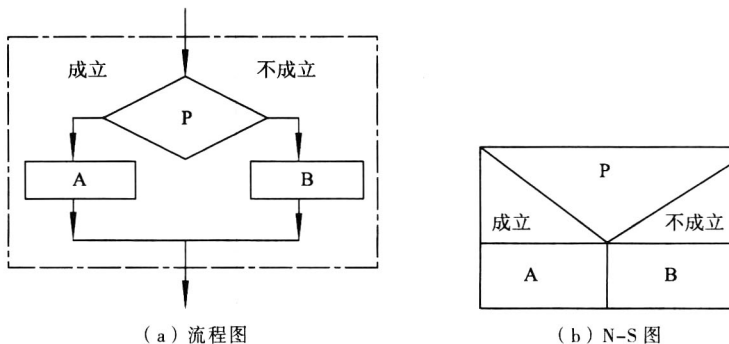


图 1-1-3 选择结构

3. 循环结构

本篇第5章中将介绍不同形式的循环结构。它们将根据各自的条件，使同一语句重复执行多次或一次也不执行。循环结构的流程图可用图1-1-4和图1-1-5所示的流程图表示。图1-1-4是当型循环，该循环的特点是：当指定的条件满足（成立）时，就执行循环体，否则就不执行。图1-1-5是直到型循环，该循环的特点是：执行循环体直到指定的条件满足（成立），就不再执行循环体。

由三种基本控制结构组成的算法结构可以解决任何复杂的问题。由三种基本控制结构构成的程序称为结构化程序。

结构化程序的特点是：

- (1) 只有一个入口。
- (2) 只有一个出口。

(3) 结构内的每一部分都有机会被执行到。

(4) 结构内不存在“死循环”。

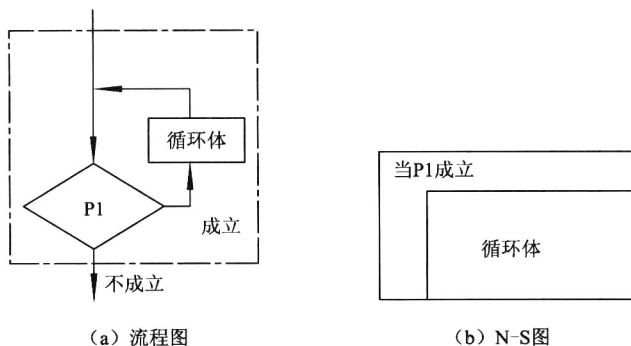


图 1-1-4 当型循环结构

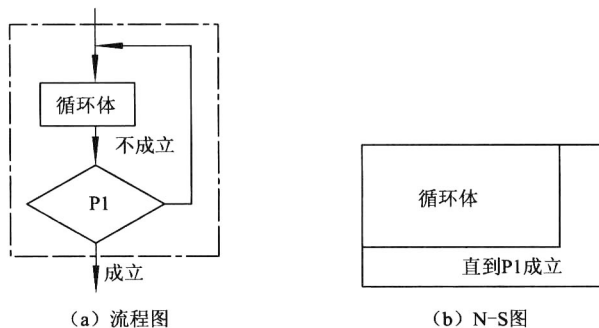


图 1-1-5 直到型循环结构

1.5 Visual C++ 6.0 集成开发环境

Visual C++ 6.0 (简称 VC++ 6.0) 是微软公司推出的目前使用极为广泛的基于 Windows 平台的可视化集成开发环境, 它和 Visual Basic、Visual FoxPro、Visual J++ 等其他软件构成了 Visual Studio 程序设计软件包。Visual Studio 是一个通用的应用程序集成开发环境, 包含文本编辑器、资源编辑器、工程编译工具、增量连接器、源代码浏览器、集成调试工具, 以及一套联机文档。使用 Visual Studio, 可以完成创建、调试、修改应用程序等各种操作。

VC++ 6.0 提供面向对象技术的支持, 它能够帮助使用 MFC 库的用户自动生成一个具有图形界面的应用程序框架。用户只需在该框架的适当部分添加、扩充代码就可以得到一个满意的应用程序。

(1) VC++ 6.0 除了包含文本编辑器、C/C++混合编译器、连接器和调试器外, 还提供了功能强大的资源编辑器和图形编辑器, 利用“所见即所得”的方式完成程序界面的设计, 从而可以大大减轻程序设计的劳动强度, 提高程序设计的效率。

(2) VC++ 6.0 的功能强大, 用途广泛, 不仅可以编写普通的应用程序, 还能很好地进行系统软件设计及通信软件的开发。

VC++ 6.0 的安装相对简单, 只要运行 Visual Studio 软件中的 setup.exe 程序, 选择安装 Visual

C++ 6.0, 然后按照安装程序的指导完成安装过程。安装完成后, 在“开始”菜单的“程序”菜单中有 Microsoft Visual Studio 6.0 图标, 选择其中的 Microsoft Visual C++ 6.0 即可运行 (也可在 Windows 桌面上建立一个快捷方式, 以后双击即可运行)。

利用 VC++ 6.0 提供的一种控制台操作方式, 可以建立 C 语言应用程序。本书涉及所有例题程序都在 VC++ 6.0 中验证通过。

小 结

本章主要讲述 C 语言的基础知识、程序设计的概念及算法。主要掌握以下几点:

- (1) 程序的构成、main()函数。
- (2) 算法的概念。
- (3) 流程图。

习 题

1. 为什么要学习 C 语言? 它与其他高级语言相比, 有什么特点?
2. 概述 C 语言程序的构成。
3. 什么是算法? 算法有什么特点?
4. 概述算法的常用表示方法。
5. 编程实现一个 C 程序, 输出以下信息。以此熟悉上机方法与步骤。

```
#####  
Welcome!  
#####
```