



21世纪高等学校计算机
基础实用规划教材

C语言程序设计

(第4版)

◎ 张磊 编著



清华大学出版社

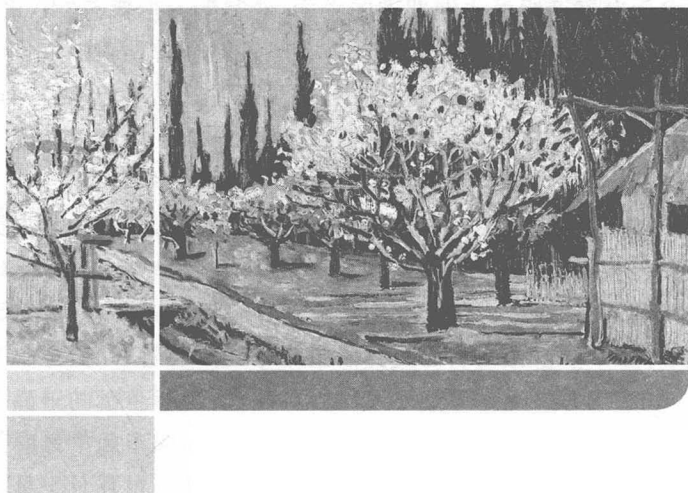




21世纪高等学校计算机
基础实用规划教材

C语言程序设计 (第4版)

◎ 张磊 编著



清华大学出版社
北京

内 容 简 介

本书是面向程序设计初学者的 C 语言基础教材,以培养大学生的逻辑思维能力和程序设计能力为编写指导思想,综合运用案例教学、比较教学、任务驱动等多种教学方法,系统介绍 C 语言程序设计的基本理论、基本方法和基本过程。本书内容组织注重基础,突出应用,兼顾提高,强化主干知识,弱化细枝末节;实例设置注重易学性、趣味性和系列化,易教易学。

全书共 10 章,内容包括程序设计概述、简单程序设计、选择结构程序设计、循环结构程序设计、数组程序设计、函数程序设计、指针程序设计、结构体程序设计、文件程序设计以及综合程序设计。

本书配有教学课件、例题及习题程序源代码等教学资源,并有辅导教材《C 语言程序设计(第 4 版)实验指导与习题解答》(ISBN9787302495932)。

本书适合作为高等院校“C 语言程序设计”课程的教材,也可用作程序设计从业人员及程序设计爱好者的自学参考书。

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 语言程序设计/张磊编著. —4 版. —北京:清华大学出版社,2018

(21 世纪高等学校计算机基础实用规划教材)

ISBN 978-7-302-49601-4

I. ①C… II. ①张… III. ①C 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 028911 号

责任编辑:付弘宇 王冰飞

封面设计:刘 键

责任校对:焦丽丽

责任印制:李红英

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>, 010-62795954

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:20

字 数:484 千字

版 次:2005 年 1 月第 1 版

2018 年 8 月第 4 版

印 次:2018 年 8 月第 1 次印刷

定 价:49.80 元

产品编号:078055-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生的知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要真实实行审稿程序,确保出书质量。

II

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机基础实用规划教材

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前 言

承蒙广大师生的厚爱和清华大学出版社的支持,近几年来在清华大学出版社出版了多种版本的 C 语言程序设计教材,实验指导与习题解答的内容以简化版形式附在主教材中合并出版。本次应广大师生要求编写了辅导教材《C 语言程序设计(第 4 版)实验指导与习题解答》(ISBN9787302495932),并借此机会对《C 语言程序设计(第 3 版)》进行了改版,编写了本书。与前一版本相比,本书主要进行了以下改进。

第一,改正了原有教材中存在的错误和不当之处,力求概念准确,表达恰当。

第二,更新例题,优化代码。一是剔除了不易讲解的例题,更新了不够经典的老例题,充实了系列例题,更易于讲解;二是注重基础例题题目与提高题目相结合,适合不同层次、不同兴趣的学生学习;三是对原有部分例题的程序代码进行了优化,更简明易懂。

第三,增加综合程序设计,强化应用能力培养。本书增加了“第 10 章 综合程序设计”,该章以通讯录程序设计为实例,体现软件工程思想,针对 C 语言结构化程序设计的特点,详细介绍了 C 语言应用程序的设计方法与过程。

第四,以新颖性、趣味性和系列化为重点对课后习题进行了优化。一是对编程题目进行了较大幅度的更新,提高题目的趣味性和吸引力;二是提高练习题目的系列化程度,便于知识的连贯性学习和系统训练。例如,在第 5~9 章增加了 Josephus 环报数游戏程序设计系列习题,并作为实验必做题目列入了实验指导内容。

本书是面向程序设计初学者的 C 语言基础教材,突出 C 语言程序设计的应用性、实践性特点,突出主干知识教学,注重逻辑思维能力和基本程序设计能力的培养,适合程序设计初学者学习使用。

本书以实例引领教学内容,符合认知规律。凡是适合以程序实例开始的新知识均通过程序实例和程序说明予以引导,首先建立感性认识,然后进行相关知识的系统介绍。

本书注重理论与实践相结合,讲解重点突出。通过大量设计性实例培养学生的程序设计能力,按照问题分析与算法设计、程序实现、程序说明及进一步讨论等内容进行系统讲解,注重算法设计、关键语句、关键程序段以及程序讨论的分析说明,重点、难点讲解透彻,而且富有启发性。

全书共 10 章,内容包括程序设计概述、简单程序设计、选择结构程序设计、循环结构程序设计、数组程序设计、函数程序设计、指针程序设计、结构体程序设计、文件程序设计以及综合程序设计。

本书有配套的教学课件、例题程序源代码、习题程序源代码以及《C 语言程序设计(第 4 版)实验指导与习题解答》等教学资源。

冯伟昌、王宗江、黄忠义、刘海慧、张莹、李竹健、张元国、王桂东、魏建国、王金才、张文、

高永存、王涛、薛莹、徐英娟、马明祥、滕秀荣、张敏、魏军、徐兴敏、周金玲、彭玉忠、潘振昌、徐思杰等参与了本书的编写并做了大量素材整理、程序调试、书稿审校等工作,在此表示感谢!

清华大学出版社付弘宇编辑和她的同事们为本书的编辑、出版做了大量严谨细致的工作,在此一并致谢!

作者的联系邮箱为 Mail16300@163.com。

编者

2018 年 3 月

目 录

第 1 章 程序设计概述	1
1.1 程序设计语言	1
1.2 算法	2
1.2.1 算法概念与算法描述.....	2
1.2.2 算法的逻辑结构.....	4
1.2.3 算法的特性.....	5
1.2.4 算法评价.....	5
1.3 程序设计与实现	6
1.3.1 程序设计的基本过程.....	6
1.3.2 编辑运行 C 语言程序	7
1.4 C 语言程序的基本结构.....	10
1.4.1 程序的函数化结构	10
1.4.2 标识符与保留字	12
1.4.3 程序风格	13
小结	13
习题一	14
第 2 章 简单程序设计	17
2.1 数据类型、常量与变量	17
2.1.1 数据类型	17
2.1.2 常量	17
2.1.3 变量	19
2.2 数据的输入与输出.....	21
2.2.1 用 printf() 函数输出数据	21
2.2.2 用 scanf() 函数输入数据.....	25
2.2.3 字符的输入与输出	27
2.3 简单运算.....	30
2.3.1 算术运算	30
2.3.2 赋值运算	31

2.3.3 变量自增和自减运算	32
2.3.4 逗号运算	32
2.4 编译预处理命令简介	33
2.5 简单程序设计举例	37
2.6 表达式中数据类型的转换	41
* 2.7 定义数据类型别名	42
* 2.8 const 常量	42
小结	43
习题二	44
第 3 章 选择结构程序设计	49
3.1 if 选择结构	49
3.1.1 if 选择结构程序示例	49
3.1.2 关系表达式	50
3.1.3 逻辑表达式	51
3.1.4 if 命令	52
3.1.5 条件运算	59
3.2 switch 选择结构	61
3.3 选择结构程序举例	63
小结	68
习题三	69
第 4 章 循环结构程序设计	73
4.1 循环结构控制命令	73
4.1.1 while 命令	73
4.1.2 do-while 命令	75
4.1.3 for 命令	76
4.2 循环体中的控制命令	78
4.2.1 break 命令	78
4.2.2 continue 命令	80
4.3 循环嵌套	81
4.4 goto 命令	83
4.5 循环结构程序举例	84
小结	95
习题四	95
第 5 章 数组程序设计	101
5.1 一维数组程序设计	101

5.1.1	一维数组程序示例	101
5.1.2	一维数组的定义及元素引用	102
5.1.3	数值型一维数组的输入和输出	103
5.1.4	数值型一维数组的初始化	105
5.1.5	字符型一维数组的初始化	107
5.1.6	一维数组的存储	107
5.2	字符串操作	108
5.2.1	字符串的输入和输出	108
5.2.2	多字符串操作函数	110
5.3	二维数组程序设计	113
5.3.1	二维数组的定义及元素引用	113
5.3.2	二维数组的输入和输出	114
5.3.3	二维数组的初始化	116
5.3.4	二维数组的存储	117
5.4	数组应用程序举例	118
	小结	128
	习题五	129
第 6 章	函数程序设计	134
6.1	函数概述	134
6.2	函数定义及调用	136
6.2.1	函数定义	136
6.2.2	函数值和 return 命令	137
6.2.3	函数调用	138
6.3	函数嵌套和递归函数	144
6.3.1	函数嵌套	144
6.3.2	递归函数	145
6.4	数组与函数	149
6.4.1	数组元素作函数参数	149
6.4.2	一维数组名作函数参数	150
6.4.3	二维数组与函数	154
6.5	函数应用程序举例	155
6.6	变量的作用域和存储类型	162
6.6.1	变量的作用域	162
6.6.2	变量的存储类型	164
*6.7	编译连接多个源文件的 C 程序	166
	小结	169

习题六	170
第 7 章 指针程序设计	175
7.1 指针概述	175
7.2 指针变量的定义和使用	176
7.2.1 指针变量程序示例	176
7.2.2 定义指针变量	176
7.2.3 使用指针变量	177
7.3 指针与数组	179
7.3.1 指针与一维数组	180
7.3.2 指针与二维数组	183
7.3.3 指针与字符串	185
7.3.4 指针数组	187
7.4 指针作函数参数	188
7.4.1 简单变量指针作函数参数	188
7.4.2 指向数组的指针作函数参数	190
7.4.3 字符串指针作函数参数	191
7.4.4 指针数组作函数参数	192
*7.4.5 使用带参数的 main() 函数	195
7.5 指针函数	196
7.6 指针应用程序举例	197
小结	199
习题七	200
第 8 章 结构体程序设计	207
8.1 结构体数据概述	207
8.2 结构体类型和结构体变量	207
8.2.1 结构体程序示例	208
8.2.2 定义结构体数据类型	208
8.2.3 结构体变量的定义及使用	209
8.3 结构体数组	214
8.3.1 结构体数组的定义及元素引用	214
8.3.2 结构体数组的初始化	215
8.3.3 结构体数组应用实例	215
8.4 结构体指针变量	216
8.4.1 结构体指针变量的定义及使用	216
8.4.2 结构体指针作函数的参数	218

8.5	使用链表存储数据	219
8.5.1	使用链表存储数据示例	219
8.5.2	链表的特点	220
8.5.3	动态内存管理函数	220
8.5.4	定义链表结构	221
8.6	链表的基本操作	223
8.6.1	链表结点的插入	223
8.6.2	链表结点的删除	226
8.6.3	链表结点的查找	228
8.7	结构体应用程序举例	231
8.8	动态数组	237
	小结	238
	习题八	239
第9章	文件程序设计	245
9.1	文件概述	245
9.1.1	文件的概念	245
9.1.2	文件的分类	246
9.1.3	文件的一般操作过程	246
9.1.4	文件类型指针	247
9.2	文件的基本操作	247
9.2.1	打开和关闭文件	247
9.2.2	文件的字符读写	249
9.2.3	文件结束状态测试	251
9.2.4	文件的数据块读写	252
9.3	文件的其他操作	255
9.3.1	文件位置指针的定位	256
9.3.2	文件的格式化读写	259
9.3.3	文件的字符串读写	260
9.4	文件应用程序举例	261
	小结	264
	习题九	264
第10章	综合程序设计	269
10.1	软件开发流程	269
10.2	通讯录程序设计	270
10.2.1	通讯录程序需求分析	270
10.2.2	通讯录程序功能设计	271
10.2.3	通讯录程序数据设计	272
10.2.4	通讯录程序函数设计	273

10.2.5 函数编码及测试	274
附录 A C 语言经典保留字	295
X 附录 B 常用 C 语言库函数	296
附录 C 字符与 ASCII 码对照表	299
附录 D C 语言的运算符	300
附录 E “学生数据处理”系列例题(习题)简表	302
参考文献	304

设计出较先进的语言并取名为 B,最后 C 语言问世。1983 年,美国国家标准化协会(ANSI)根据 C 语言问世以来的各种版本对 C 的发展和扩充制定了 C 的标准,称为 ANSI C。1987 年 ANSI 又公布了新的标准——87 ANSI C,目前流行的 C 编译系统都是以它为基础的。

在 C 语言的基础上,1983 年贝尔实验室的 Bjarne Stroustrup 推出了 C++ 语言。C++ 语言作为 C 语言的继承和发展,不仅保留了 C 语言的高度灵活、高效率和易于理解等诸多优点,还包含了几乎所有面向对象的特征,成为一种面向对象的程序设计语言。C++ 语言所支持的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的结构化程序设计不同的思维方式和编程方法。

下面是一个用 C 语言编写的计算机程序,它通过累加的方法计算 1~100 的所有自然数的和。

```
#include <stdio.h>
int main()
{
    int i = 1, s = 0;
    while(i <= 100)          /* 循环控制 */
    {
        s = s + i;          /* 数据累加 */
        i = i + 1;          /* 生成下一个要累加的数 */
    }
    printf("sum = %d\n", s); /* 输出结果 */
    return 0;
}
```

当然,计算 1~100 的所有自然数的和还有其他更加高效的方法,例如等差数列求和法。如何选用和设计有效的算法解决问题是程序设计的重要内容。

1.2 算 法

瑞士科学家、Pascal 语言发明者 Niklaus Wirth 对计算机程序给出了一个著名的定义,即程序 = 数据结构 + 算法。该定义归结了计算机程序的两个核心问题,强调了算法在程序中的重要性。

1.2.1 算法概念与算法描述

1. 算法的概念

算法是为计算机处理问题所设计的具体步骤,算法的最终实现是计算机程序。程序设计人员只有将算法转变为计算机程序才能利用计算机解决问题。

算法的建立通常会经过由粗略到细化的过程,先把解决问题的基本过程表达出来,确立粗略的算法框架,然后添加必要的细节,形成解决问题的有效算法。

由于一个具体问题可以有不同的解决方法,自然就能设计出解决问题的不同算法。因此,即便使用同一种计算机语言,在解决同一个问题时也可能有多个不同的计算机程序,认识这一点对学习程序设计是非常重要的。

2. 算法的描述方法

算法的描述方法多种多样,可以使用自然语言描述,也可以使用专门的算法表达工具进行描述。为了使算法的表达更清晰,更容易实现程序编写,在进行程序设计时通常使用专门的算法表达工具对算法进行描述,例如流程图、N-S图、PAD图、伪代码等。以下是使用自然语言和流程图描述算法的实例说明。

1) 用自然语言描述算法

问题: 计算 1~100 的所有自然数的和。

最直观的理解,计算 1~100 的所有自然数的和(以下称为“自然数累加”问题),就是求以下代数式的值:

$$1+2+3+4+\dots+99+100$$

显然可以采用逐个自然数累加的方法求和。算法的粗略描述如下:

假如用 i 表示当前要加的数, i 开始取值为 1,每加一次, i 的值增加 1;用 s 表示已经累加取得的结果,开始取值为 0。那么,问题求解的过程就是不断地将 i 加到 s 中,直到 i 的值超过 100 时结束累加过程,并将累加的结果显示在计算机屏幕上。

上面一段文字对问题的求解方法进行了基本描述,但作为算法还不够完整,还需要更明确地表达出求解问题的步骤。下面是包含了执行步骤的算法描述,是用自然语言对算法进行描述的常见形式。

步骤① i 和 s 赋初值,使 $i=1, s=0$;

步骤② 判断 i 的值,若 $i \leq 100$,则执行步骤③,否则转步骤⑤;

步骤③ s 加上 i ;

步骤④ i 加上 1,转步骤②;

步骤⑤ 输出 s 的值,结束。

按照上述算法确定的 5 个步骤即可求解“自然数累加”问题。若选用一种计算机语言正确描述该算法,就会得到求解“自然数累加”问题的计算机程序,执行程序,将得到“自然数累加”问题的计算结果。

2) 用流程图描述算法

流程图是人们经常使用的一种算法描述工具,其特点是绘制简单、结构清晰、逻辑性强、便于描述、容易理解。表 1-1 列出了常用的流程图符号及其功能。

表 1-1 常用的流程图符号及其功能

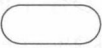
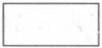

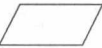

流程图符号	符号的功能
	开始、结束
	处理
	判断
	输入、输出
	流程方向

图 1-1 是利用流程图符号对“自然数累加”问题的算法进行表达的算法流程图。其中, $1 \rightarrow i$ 表示使 i 的值为 1, $s+i \rightarrow s$ 表示将 s 的值加上 i 。

1.2.2 算法的逻辑结构

顺序结构、选择结构和循环结构是算法的 3 种基本逻辑结构, 这 3 种结构互相结合可以实现任何逻辑控制。

1. 顺序结构

顺序结构的算法, 其各个步骤由前到后依次执行, 每个步骤都被执行一次, 其逻辑结构如图 1-2 所示。顺序结构的算法用程序实现后, 程序中的每一个语句将按照排列顺序由前到后依次执行, 直到最后一个语句被执行, 程序结束。图 1-3 所示为顺序结构算法的一个实例, 其中 s_1, s_2 表示一个学生的两门课程的成绩, $(s_1 + s_2)/2 \rightarrow ave$ 表示使 ave 的值为 $(s_1 + s_2)/2$ 。该算法实现的功能是输入一个学生的两门课程的成绩, 计算并输出平均成绩。

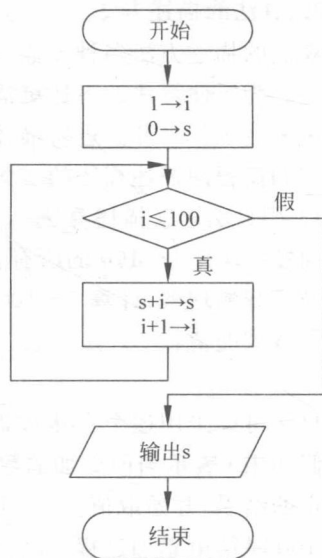


图 1-1 “自然数累加”问题的算法流程图

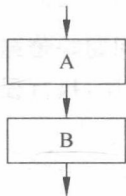


图 1-2 顺序结构逻辑图



图 1-3 顺序结构算法实例

2. 选择结构

选择结构的算法的某些步骤是否能够执行要视当前的条件而定, 在同一次处理中有的步骤可能执行不到, 其逻辑结构如图 1-4 所示。这种算法用程序实现后, 程序中的有些语句就会被有选择地执行。图 1-5 所示为选择结构算法的一个实例, 其中 s_1 和 s_2 的含义与上

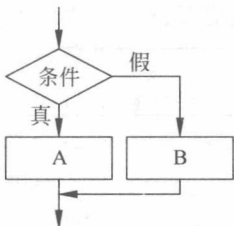


图 1-4 选择结构逻辑图

相同。该算法的功能是输入一个学生的两门课程的成绩, 若平均成绩不低于 90, 则输出“优等生”, 否则输出“加油!”。

3. 循环结构

循环结构的算法中有些步骤会被有条件地重复执行, 被重复执行的步骤每次处理的数据可能发生变化, 其逻辑结构如图 1-6 所示, 算法实例如图 1-1 所示。这种算法用程序实现后, 程序中的有些语句就会被反复地执行。

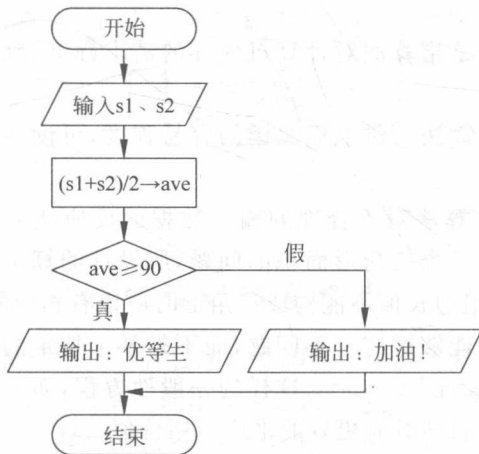


图 1-5 选择结构算法实例

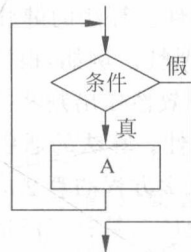


图 1-6 循环结构逻辑图

用计算机语言实现一个具体的算法时,其中的选择结构和循环结构由专门的控制命令予以实现。当一个程序段既没有选择结构也没有循环结构时,该程序段的逻辑结构即为顺序结构。

1.2.3 算法的特性

算法具有如下特性:

(1) 有穷性。一个算法必须经过有限步骤之后结束,对应算法的实现程序,计算机能够在执行有限步骤后给出结果。但在实际应用中,所谓“有限步骤”,既指步骤量有限,同时也要求步骤量合理。

(2) 确定性。一个算法通常由一系列求解步骤来完成,各操作步骤之间有严格的顺序关系,每一个步骤所规定的操作必须是确定的,不能有二义性。例如前面所讨论的“自然数累加”问题的算法,其步骤④是“i 加上 1,转步骤②”,若将其描述为“i 加上 1,转其他步骤”,那么该步骤就是不确定的。

(3) 有效性。一个算法中不能出现无效的步骤,每一个步骤描述的操作必须能够通过已经实现的基本运算有效地执行,并且得到确定的结果。例如,若某个步骤有除法算式 a/b ,若不能保证执行该步骤时 b 的值不会为 0,那么该步骤就不是一个有效的步骤。

(4) 输入和输出特性。每个有意义的算法有零个或多个输入,并且提供一个或多个输出。所谓 0 个输入是指算法本身定义了初始条件,能够提供初始数据;而多个输入是指算法能够从外部获得处理数据。输出是指算法执行后能够产生输出信息,以反映数据处理结果,没有输出的算法是毫无意义的。

1.2.4 算法评价

算法评价涉及多个方面,例如算法的时间复杂性、空间复杂性、可读性、健壮性、通用性和正确性等,简要介绍如下:

(1) 时间复杂性。算法的时间复杂性是指算法执行时间与问题规模的关系,它是反映算法执行效率的一个指标。一个好的算法应该是当问题规模增大时执行算法所需时间的增