

高等职业教育
大学专科

计算机系列教材

VB BASIC 程序设计

鲍友文 周海燕 编著
谭浩强 主审

科学出版社

73 87424



10972019

73.87424
0224

高等职业教育
大学专科

计算机系列教材



QBASIC 程序设计

鲍友文 周海燕 编著
谭浩强 主审

科学出版社

1999

T132

内 容 简 介

本书是为第一次学习计算机语言程序设计的初学读者编写的教材,可作为高等职业教育及成人高等教育学生的教学用书,也可作为一些面向实际应用的普通高等教育专业的教材。

全书共包含十二章和八个附录。第一章介绍了程序设计的基本知识,是独立于 QBASIC 语言内容的。第二章介绍了 QBASIC 语言的基本知识,包括 QBASIC 源程序的基本结构和基本数据类型。第三章至第十章讲述了 QBASIC 语言的程序设计语句、数组和记录类型、字符串的有关函数、全局变量与局部变量及静态变量与动态变量等内容。第十一章介绍了 QBASIC 集成环境的一般使用。第十二章对全书涉及到的算法进行了总结,并给出了三个综合的程序设计实例。全书内容由浅入深,循序渐进,通俗易懂,并配备了大量的例题和习题。

本书对于参加“国家信息技术(NIT)证书(QBASIC 程序设计模块)”培训和考试的有关人员,也是一本较为理想的辅导参考书。

图书在版编目(CIP)数据

QBASIC 程序设计/鲍友文,周海燕编著. -北京:科学出版社,1999.2

(高等职业教育大学专科计算机系列教材)

ISBN 7-03-006968-4

I. Q… II. ①鲍… ②周… III. QBASIC 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(1998)第 38683 号

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

北京双青印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1999 年 2 月第 一 版 开本:787×1092 1/16

1999 年 2 月第一次印刷 印张:21 1/2

印数:1—3 400 字数:489 000

定价:28.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

序 言

20 世纪后半叶,随着高新技术的发展和第三产业的兴起,人类社会进入信息时代,知识经济成为新世纪的主题。社会经济发展的多元化以及多层次需求推动了高等教育的改革,一类新型的高等教育——高等技术教育——的诞生是改革的重要成果。联合国教科文组织 1997 年公布的教育分类中,将这类教育称为“高等技术与职业教育”。它大体上可分为本科和专科两个层次,而某些经济发达国家,已开始将这类教育拓展到研究生层次。我国改革开放以来,由于经济的高速发展、产业结构的调整,对这类教育的需求已十分迫切,如对飞行员、海员、计算机高级程序设计员、网络管理员、高级营销员、广告创意策划制作员、国际贸易服务、海关报关员等高级专门人才的培养,沿用传统的普通高等教育模式已很难达到目的。90 年代以来,这类特殊类型的高等教育——高等技术教育——从无到有,在我国得到迅速发展。我国教育部门将这类教育称为高等职业教育。改革的目标是大学专科都将逐步走向高等职业教育,本科已开始试点。发展高等职业教育在教学模式改革后,关键是课程内容的改革和教材建设。这次我们组织北京地区较早进行高等职业教育改革的院校的教授专家共同编写了这套系列教材,以满足我国发展高等职业教育对合适教材的迫切需要。

21 世纪电脑及其网络的使用,将成为改变人类社会生活的重要的文化现象,改变人们已习惯的用笔写字、画图、计算、学习、生活的方式。因此电脑教育将作为人类教育不可缺少的组成部分,也将是任何高等教育中的必修课程。本系列教材共分五本,具体内容是:

- (1)《计算机操作技术基础》
- (2)《FoxPro 数据库技术基础》
- (3)《QBASIC 程序设计》
- (4)《计算机工程制图》
- (5)《Internet/Intranet 基础与应用训练》

本系列教材的特点是:

1. 以学会使用计算机的技术训练为主

对于广大计算机的使用者来说,必须把计算机当作提高脑力劳动生产率的工具,重点是解决如何应用的问题,使计算机为自己的专业需要服务。因此,本套教材是以学会使用计算机为目标而编写的。

2. 面向问题

一般教材编写的规律是面向概念,具体写法是首先引出概念,然后解释概念,最后举例说明。由于本套教材强调应用,因此写作中是从读者在实际应用中可能遇到的问题出发,采用提出问题、案例分析、解决问题、最后导出必要的概念的写法,以更贴近读者的实际需要。

3. 边学边练

使用电脑是靠练会的,而不是靠看书看会的。因此本套教材将给出大量的训练习题,

只要读者按照这些习题由浅入深进行练习,就一定能达到学会使用的目的。

本系列教材可作为高等职业教育、成人高等教育学生的教学用书,也可作为一些面向实际应用的普通高等教育专业的教材。由于最近教育部开考的“国家信息技术(NIT)证书”培训和考试也是一类面向应用的技术考试,本套书中的部分教材也可作为 NIT 考试的理想参考书。

前 言

随着计算机科学技术的迅速发展,学习计算机知识已成为各行各业人们的迫切需求。为了适应不同工作领域中、具有不同计算机基础的计算机语言学习者的需要,我们编写了这本 QBASIC 语言程序设计教材。

本教材遵循循序渐进、深入浅出、难点分散的编写原则,故更适合于初学计算机程序设计语言的读者。本书可作为大学专科、高等职业教育本科及专科、成人夜大学专科等层次学生的教材用书;也可作为非计算机专业大学本科和自学人员的参考书。此外,对于参加“国家信息技术(NIT)证书(QBASIC 程序设计)”培训和考试的有关人员,本书也是一本较为理想的辅导参考书。

全书主要内容可归为两大组成部分。第一部分指的是第二章至第十章中有关 QBASIC 语言基本语法内容的介绍,包括:QBASIC 语言的特点、QBASIC 源程序的基本组成、QBASIC 语言的数据类型、QBASIC 语言的基本语句及简单编程。第二部分主要由各章的应用举例、自测习题、第一章(程序设计的基础知识)和第十二章(程序设计综合举例)等内容组成,读者通过第二部分内容的学习和实践,可以比较全面地了解算法设计的重要性和结构化及模块化程序设计的基本概念,掌握程序设计的基本功。

考虑到本书的读者将是计算机语言程序设计的初学者,故本书具有以下几个特点:

(1)不过分强调 QBASIC 语言的语法细节,也不强求语法内容的面面俱到。而强调语法知识的学习主要是为培养读者的程序设计能力服务,并强调 QBASIC 语言程序设计课程的实践性。故全书各章用了较大篇幅介绍应用实例,并在实例的讨论中,从问题分析和算法设计入手,以使读者掌握程序设计的完整过程,尽快具备独立分析问题并能够使用一门计算机语言解决问题的能力。

(2)为便于读者巩固和检查所学过的每章主要内容,各章配备了大量的自测习题。习题量之大在同类教材中是不多见的。

(3)为便于读者自学,本书自第二章至第十章,在每章最后都专门设置一节,对全章的要点与难点及学习本章内容时容易出现的常见错误进行总结。

(4)为了强调结构化的程序设计风格,本书删减了 QBASIC 中的一些非结构化语句内容的介绍,如:ON-GOTO 语句、ON GOSUB-RETURN 语句等。

在第十一章介绍了 QBASIC 集成环境的一般使用,读者可根据需要把本章中各节内容分散到前面相关章中学习。比如,学习第三章时应同时学习第 11.1~11.4 节的内容,以掌握 QBASIC 语言环境的基本使用。又比如,学习第八章时应配合学习第 11.6 节的内容,以掌握 SUB 过程和 FUNCTION 过程的上机操作过程。本章中最后一节内容(关于 QBASIC 程序调试的常用方法)可放到读者已经具备了使用 QBASIC 语言编程的基本能力之后再进行学习。

书中标有“*”的个别例题或习题,涉及到了高等数学的知识,读者在学习时可根据具体情况决定取舍。

在本书的筹划和编写过程中,得到了高林教授和田淑清教授的热情帮助和指导,在此,我们对他们表示衷心的感谢。

由于作者水平有限,且时间仓促,书中难免会存在着错误和疏忽之处,恳请专家和读者予以指正。

作 者

1998. 11

高等职业教育
大学专科 计算机系列教材

编写委员会

主 编 高 林 汪继祥

副主编 王启智

编 委 (按姓名笔画排列)

支芬和	牛大刚	刘璇	吴悦	张俊玲
陈文博	陈南	周立	周海燕	郑坚
胡景凡	鲍友文	樊月华		

目 录

序言

前言

第一章 程序设计的基础知识	(1)
1.1 算法的基本概念	(1)
1.1.1 什么是算法	(1)
1.1.2 计算机算法的特性	(2)
1.2 常用的算法描述工具	(3)
1.2.1 使用自然语言描述算法	(3)
1.2.2 使用流程图描述算法	(4)
1.2.3 使用 N-S 图描述算法	(5)
1.3 算法设计举例	(7)
1.4 程序设计的基本概念	(11)
1.4.1 程序设计的基本步骤	(11)
1.4.2 结构化程序的三种基本结构	(12)
1.5 本书的约定	(14)
1.5.1 语句格式中的符号约定	(14)
1.5.2 QBASIC 保留字及其他名字的约定	(15)
1.5.3 程序书写形式的约定	(15)
1.6 本章要点与难点	(15)
自测习题	(16)
第二章 QBASIC 语言的基础知识	(18)
2.1 BASIC 语言的发展与 QBASIC 语言的特点	(18)
2.2 QBASIC 源程序的基本结构	(19)
2.3 数据类型	(20)
2.4 常量和变量	(20)
2.4.1 常量	(20)
2.4.2 变量	(22)
2.5 数学库函数	(25)
2.6 算术运算符和算术表达式	(26)
2.7 本章要点与难点	(28)
2.7.1 不同类型的数值常量的正确表示及取值范围	(29)
2.7.2 正确使用变量定义的三种形式	(29)
2.7.3 算术表达式的正确表示和求值	(29)
自测习题	(30)
第三章 顺序结构的实现	(32)
3.1 问题的引出	(32)
3.2 数据输出语句	(33)

3.2.1 PRINT 语句	(33)
3.2.2 PRINT USING 语句	(36)
3.2.3 LPRINT 语句	(37)
3.2.4 WRITE 语句	(38)
3.3 赋值语句(LET 语句)	(38)
3.4 数据输入语句	(39)
3.4.1 键盘输入语句(INPUT 语句)	(40)
3.4.2 读数/置数语句(READ/DATA 语句)	(42)
3.4.3 恢复数据指针语句(RESTORE 语句)	(44)
3.5 终止程序运行语句(END 语句)	(45)
3.6 暂停程序运行语句(STOP 语句)	(46)
3.7 注释语句	(46)
3.8 清屏语句(CLS 语句)	(46)
3.9 变量值交换语句(SWAP 语句)	(47)
3.10 符号常数说明语句(CONST 语句)	(48)
3.11 应用举例	(49)
3.12 本章要点与难点	(51)
3.12.1 LET 语句、INPUT 语句和 READ/DATA 语句的使用比较	(51)
3.12.2 正确理解使用 READ/DATA 语句时的数据指针概念	(52)
3.12.3 利用 PRINT 语句控制输出格式	(52)
3.12.4 利用 PRINT USING 语句控制输出格式	(53)
自测习题	(54)
第四章 选择结构的实现	(58)
4.1 问题的引出	(58)
4.2 条件的描述	(59)
4.2.1 关系运算符和关系表达式	(59)
4.2.2 逻辑运算符和逻辑表达式	(59)
4.3 用单行 IF 语句和块 IF 语句实现选择结构	(60)
4.3.1 两分支选择结构的实现	(60)
4.3.2 使用嵌套的单行 IF 语句和嵌套的块 IF 语句实现多分支选择结构	(63)
4.4 使用 SELECT CASE 语句实现多分支选择结构	(67)
4.5 应用举例	(72)
4.6 本章要点与难点	(75)
4.6.1 关系表达式和逻辑表达式的正确使用	(75)
4.6.2 嵌套 IF 语句的正确使用	(76)
4.6.3 SELECT CASE 语句的正确使用	(76)
自测习题	(77)
第五章 循环结构的实现	(82)
5.1 问题的引出	(82)
5.2 用 WHILE-WEND 语句实现循环结构	(85)
5.2.1 WHILE 循环的结构	(85)
5.2.2 WHILE 循环的执行过程	(86)
5.2.3 WHILE 循环的使用	(87)

5.3 用 DO-LOOP 语句实现循环结构	(88)
5.3.1 DO 循环的格式及执行过程	(89)
5.3.2 由基本 DO 语句和 EXIT DO 语句构成的循环结构	(90)
5.3.3 带 WHILE 子句的 DO 循环	(90)
5.3.4 带 UNTIL 子句的 DO 循环	(93)
5.4 用 FOR-NEXT 语句实现循环结构	(94)
5.4.1 FOR 循环的构成	(94)
5.4.2 FOR 循环的执行过程	(96)
5.4.3 FOR 循环的使用	(97)
5.5 循环的嵌套	(99)
5.6 应用举例	(102)
5.7 本章要点与难点	(111)
5.7.1 循环条件的正确使用	(111)
5.7.2 DO 循环的正确使用	(112)
5.7.3 循环嵌套的正确使用	(113)
5.7.4 与循环有关的算法设计	(113)
自测习题	(114)
第六章 数组和记录	(121)
6.1 一维数组的引出	(121)
6.2 一维数组的建立和元素的引用	(122)
6.2.1 一维数组的建立	(122)
6.2.2 一维数组元素的引用	(124)
6.3 一维数组应用举例	(126)
6.3.1 查找	(126)
6.3.2 插入	(131)
6.3.3 删除	(133)
6.3.4 排序	(136)
6.3.5 其他	(140)
6.4 二维数组的引出	(141)
6.5 二维数组的建立和元素的引用	(142)
6.5.1 二维数组的建立	(142)
6.5.2 二维数组元素的引用	(143)
6.6 静态数组和动态数组	(144)
6.7 二维数组应用举例	(147)
6.7.1 查找	(147)
6.7.2 矩阵运算	(148)
6.7.3 特殊矩阵的生成	(148)
6.8 记录	(150)
6.8.1 记录类型的引出	(150)
6.8.2 记录类型的定义	(151)
6.8.3 记录类型变量和数组的定义	(151)
6.8.4 记录中域的表示方法	(152)
6.8.5 记录类型数据的应用举例	(152)

6.9 本章要点与难点	(153)
6.9.1 正确理解和使用数组定义、重定义语句	(153)
6.9.2 数组元素的正确引用	(154)
6.9.3 有关数组应用的算法设计	(155)
6.9.4 注意区分记录类型名和记录变量名	(155)
自测习题	(155)
第七章 字符串和字符串数组	(162)
7.1 问题的引出	(162)
7.2 字符串常量和字符串变量	(163)
7.2.1 字符串常量	(163)
7.2.2 字符串变量	(163)
7.3 有关字符串的操作	(165)
7.3.1 字符串变量的赋值	(165)
7.3.2 字符串的连接	(167)
7.3.3 字符串的比较	(168)
7.4 字符串数组	(169)
7.5 用于字符串处理的库函数	(171)
7.5.1 求字符串长度的函数(LEN)	(171)
7.5.2 子字符串处理函数(LEFT\$、RIGHT\$、MID\$)	(171)
7.5.3 字符串与数值间的转换函数(VAL、STR\$)	(173)
7.5.4 字符与 ASCII 码之间的转换函数(ASC、CHR\$)	(174)
7.5.5 大、小写字母转换函数(LCASE\$、UCASE\$)	(174)
7.5.6 生成由相同字符组成的字符串函数(STRING\$、SPACE\$、SPC)	(175)
7.5.7 字符(串)专用读入函数(INKEY\$、INPUT\$)	(175)
7.5.8 日期和时间函数及语句(DATE\$、TIME\$)	(176)
7.6 字符串应用举例	(177)
7.7 本章要点与难点	(182)
自测习题	(183)
第八章 过程	(187)
8.1 问题的引出	(187)
8.2 子程序过程(SUB 过程)	(188)
8.2.1 定义一个子程序过程	(188)
8.2.2 子程序过程的调用	(190)
8.2.3 调用 SUB 过程时的参数传递	(191)
8.3 函数过程(FUNCTION 过程)	(195)
8.3.1 定义一个函数过程	(196)
8.3.2 函数过程的调用	(196)
8.3.3 调用 FUNCTION 过程时的参数传递	(197)
8.4 过程的嵌套调用及递归调用	(198)
8.4.1 过程的嵌套调用	(198)
8.4.2 过程的递归调用	(200)
8.5 关于过程中变量使用的进一步讨论	(203)
8.5.1 全局变量	(203)

8.5.2 局部变量	(204)
8.5.3 静态变量与动态变量	(206)
8.6 用户自定义函数与块内子程序	(207)
8.6.1 用户自定义函数	(207)
8.6.2 块内子程序	(208)
8.7 应用举例	(209)
8.8 本章要点与难点	(218)
8.8.1 按地址传递方式中的形参对实参的影响	(218)
8.8.2 全局变量与局部变量的不同	(219)
8.8.3 动态变量与静态变量的不同	(220)
自测习题	(221)
第九章 文件	(229)
9.1 文件的概念	(229)
9.1.1 文件的分类	(229)
9.1.2 数据文件的用途及组成	(230)
9.1.3 文件的读写和文件缓冲区	(230)
9.2 顺序文件	(231)
9.2.1 顺序文件的打开与关闭	(231)
9.2.2 顺序文件的读写	(232)
9.2.3 顺序文件的修改	(236)
9.3 随机文件	(238)
9.3.1 随机文件的打开与关闭	(238)
9.3.2 随机文件的读写	(239)
9.3.3 在随机文件中使用记录变量的读写方法	(241)
9.4 其他常用文件和目录操作语句	(247)
9.5 本章要点与难点	(248)
自测习题	(249)
第十章 屏幕控制和简单作图	(252)
10.1 显示模式和屏幕坐标系	(252)
10.1.1 文本模式和图形模式	(252)
10.1.2 屏幕坐标系	(253)
10.2 屏幕控制语句和函数	(253)
10.2.1 屏幕定义(SCREEN)语句	(253)
10.2.2 清除屏幕(CLS)语句	(254)
10.2.3 宽度设置(WIDTH)语句	(254)
10.2.4 光标定位(LOCATE)语句	(254)
10.2.5 CSRLIN 和 POS 函数	(255)
10.3 简单作图语句	(255)
10.3.1 画点	(256)
10.3.2 画线、矩形框、矩形块	(256)
10.3.3 画圆、椭圆和圆弧	(258)
10.4 图形的颜色	(259)
10.4.1 色彩设置(COLOR)语句	(259)

10.4.2 图形着色(PAINT)语句	(260)
10.5 图形窗口	(261)
10.5.1 视窗(VIEW)语句	(261)
10.5.2 窗口(WINDOW)语句	(262)
10.6 程序设计举例	(263)
10.7 本章要点与难点	(265)
自测习题	(267)
第十一章 QBASIC 的集成使用环境	(269)
11.1 QBASIC 环境的一般使用	(269)
11.1.1 QBASIC 的启动与退出	(269)
11.1.2 QBASIC 工作窗口的主要成分	(270)
11.2 QBASIC 程序的建立、存盘和打开	(272)
11.2.1 建立一个新程序	(272)
11.2.2 程序的存盘	(273)
11.2.3 打开一个已存在的文件	(274)
11.3 QBASIC 程序的运行和编辑	(274)
11.3.1 程序的运行	(274)
11.3.2 QBASIC 源程序的编辑	(275)
11.4 使用 QBASIC 的“联机帮助”	(278)
11.5 有关 SUB 过程和 FUNCTION 过程的操作	(279)
11.5.1 建立一个过程	(279)
11.5.2 对程序(包含主程序和过程)的一般操作	(280)
11.5.3 在屏幕上同时显示主程序和过程的方法	(281)
11.6 QBASIC 程序调试的常用方法	(282)
11.6.1 使用 PRINT 或 STOP 语句	(282)
11.6.2 使用 QBASIC 的调试功能	(285)
自测习题	(288)
第十二章 程序设计实践	(291)
12.1 全书算法总结	(291)
12.2 程序设计综合举例	(293)
自测习题	(313)
附录	(314)
附录 A 字符与 ASCII 代码对照表	(314)
附录 B 常用的 ASCII 控制字符	(315)
附录 C QBASIC 保留字	(316)
附录 D QBASIC 语句一览表	(318)
附录 E QBASIC 函数一览表	(322)
附录 F PRINT USING 语句的格式字符	(325)
附录 G QBASIC 支持的键盘编辑命令一览表	(326)
附录 H QBASIC 环境中的键盘快捷键一览表	(328)
参考文献	(330)

第一章 程序设计的基础知识

本章学习目的

- (1)了解算法的一般概念和计算机算法的特性。
- (2)掌握三种算法描述工具:自然语言、流程图、N-S图。
- (3)掌握简单问题的算法设计。
- (4)掌握程序设计的基本步骤和结构化程序的三种基本结构。

1.1 算法的基本概念

虽然计算机能够处理的问题是千变万化的,但是计算机最终只能识别和执行由计算机语言编写的“命令”,即通常所说的“程序”,由此就引出了程序的概念并要求学习计算机知识的人必须掌握程序设计的基本知识。一般地说,为一个实际问题进行程序设计时通常要涉及两个方面的内容:

- (1)为待解决的问题确定正确的解题方法和步骤;
- (2)使用某种计算机语言把已确定的解题步骤表示成计算机能够执行的命令。

第(1)个内容涉及的就是本节将要介绍的关于算法的概念及设计;而第(2)个内容就是全书的主要内容,即如何使用 QBASIC 语言编写程序。有关算法设计与程序设计的讨论涉及一个很大的题目,很多学校都把它专门作为一门课程来开设。限于本书的篇幅,我们只在本章中简要介绍算法设计及程序设计的基本知识,至于这些基本知识的应用,则主要通过后续各章中引入的实例来进行介绍。

1.1.1 什么是算法

人们在处理日常生活中各类事情时,都会有意或无意地按照一定的方法和步骤来进行。例如,某人为了尽量减少上班路途上花费的时间,就会考虑选择一个最佳的交通方案,它可能是:离开住处⇒步行⇒乘地铁⇒步行⇒转乘公共汽车⇒再步行⇒到达目的地。这个交通方案其实就是解决一个特定上班路线问题的“算法”。广义地讲,所谓“算法”就是解决问题时所采用的方法和步骤,换句话说,算法就是对问题处理过程的一种描述。也正是从这个意义上说,处理任何事情都存在着算法。显然,解决同一个问题的算法通常不止一个,比如上班的路线可能有多条。但是一般地说,对一个具体问题而言,总存在着一个最佳算法,人们在解决问题时实际上就是在为问题寻找着一个最佳算法。鉴于本书的内容,我们只讨论用计算机处理问题时所涉及到的算法,即计算机算法。

1.1.2 计算机算法的特性

首先看一个计算机求解 $\sum_{n=1}^{10} \frac{1}{n}$ 的例子。

【例 1.1】计算 $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{10}$ 。

问题分析：计算 $\sum_{n=1}^{10} \frac{1}{n}$ 的过程中存在着两个特点：

- (1) 重复进行加法计算(通常称作“累加”),每加一个数,“和”都要发生变化;
- (2) 加数的变化是有规律的,即每加一次后加数的分母值增 1。

解决这个问题的算法可以写作：

- (1) 设用 sum 表示累加结果,用 n 表示加数的分母;
- (2) 先分别为 sum 和 n 赋初值 0 和 1,表示为: $0 \Rightarrow \text{sum}$ 和 $1 \Rightarrow n$;
- (3) 进行加法运算,每次求和结果赋给 sum,表示为: $1/n + \text{sum} \Rightarrow \text{sum}$;
- (4) 加数的分母增 1,表示为: $n+1 \Rightarrow n$;
- (5) 每次分母增 1 后判断 n 值:当 $n < 10$ 或 $n = 10$ 时,返回第(3)步,重复执行(3)~(5);当 $n > 10$ 时,执行第(6)步;
- (6) 输出 sum 值,sum 值即所求。

以上算法(或称计算机算法)实际就是用计算机解决 $\sum_{n=1}^{10} \frac{1}{n}$ 问题的方法和步骤。显然这个算法不是唯一的,例如计算 $\sum_{n=1}^{10} \frac{1}{n}$ 还可以采用先把 10 个加数通分后再相加的方法。一般而言,计算机算法具有以下 5 个特性:

(1) 有穷性。任一个算法只能包含有限次的执行步骤,即任何算法必须在有限时间内完成。比如例 1.1 中的操作步骤是 6 个,操作的结束条件是 $n > 10$ 。假使要改成计算 n 为非定值时的 $\frac{1}{n}$ 累加和,显然就不能用有限步骤和有限时间来完成,即不满足“有穷性”的要求。

(2) 确定性。算法中的每个步骤都必须有明确的定义,不能有“多义性”。例如把上面算法的第(5)步改成:“每次分母增 1 后判断 n 值:当 $n < 10$ 时,返回第(3)步,重复执行(3)~(5);否则执行第(6)步”就会产生不正确的计算结果。这是因为改写后的第(5)步,仅考虑了 $n < 10$ 的一种情况(即“ $n < 10$ 时返回第(3)步,重复执行(3)~(5)”),而没考虑 $n = 10$ 的情况。故 $n = 10$ 和 $n > 10$ 都只能归结到“否则执行第(6)步”的情况中,显然这就会造成定义不明确。

(3) 有效性。计算机算法中的每个步骤都应当能被计算机有效地执行,并能产生有效的结果。例如,“一个数被零除”就不是有效的算法。再例如,对于一台只能表示的最大整数值为 32767 的计算机,让其表示大于 32767 的整数也是不能被有效执行的算法。

(4) 零个或多个输入。开始执行算法时,可有零个或多个输入值。如:上面例 1.1 的算法就属于零个输入的情况。

(5) 一个或多个输出。算法执行结束后,必须输出算法的执行结果(结果可能不止一

个)。如例 1.1 中有一个输出,即输出 $\sum_{n=1}^{10} \frac{1}{n}$ 的计算结果。

这里需要强调的是,读者在为实际问题进行算法设计时,一定要使算法符合上述算法特性,以保证所设计的算法能被计算机正确地执行并得到预期的结果。此外,初学算法设计的读者除了注意算法的正确性,还应尽量注意算法的简明性,以使算法易于理解、交流和修改。

1.2 常用的算法描述工具

在进行算法设计时,首先要解决的问题是如何把算法正确地表示出来,这就要依赖于算法的描述工具。目前虽然存在着多种形式的算法描述工具,但是都可以归结到以下两类:

- 文字描述工具
- 图形描述工具

在文字描述工具中,常用的描述形式有自然语言和伪代码两种;在图形描述工具中,常用的有流程图、N-S 图、PAD 图等工具。限于篇幅,本书只介绍自然语言、流程图和 N-S 图这三种算法描述工具。

1.2.1 使用自然语言描述算法

所谓“自然语言”指的就是我们日常生活中使用的语言,如汉语、英语或数学表达式等。请看下例。

【例 1.2】用自然语言描述一元二次方程 $ax^2+bx+c=0(a \neq 0)$ 全部根的算法。

问题分析:根据代数知识可知,一元二次方程的根可能存在下面三种情况:

(1) 若 $b^2-4ac>0$,则有两个不相等的实根,即:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

(2) 若 $b^2-4ac=0$,则有两个相等的实根,即:

$$x_{1,2} = \frac{-b}{2a}$$

(3) 若 $b^2-4ac<0$,则有一对共轭复根,即:

$$x_{1,2} = \frac{-b}{2a} \pm \frac{\sqrt{-(b^2 - 4ac)}}{2a}i$$

用自然语言描述算法如下:

(1) 输入 a, b, c 的值;设 $d = b^2 - 4ac$ 。

(2) 根据 d 值的正负,求出根是实根还是复根:

①如果 $d>0$,则得到两个不相等的实根为:

$$x_1 = \frac{-b + \sqrt{d}}{2a}, \quad x_2 = \frac{-b - \sqrt{d}}{2a}$$

②如果 $d=0$,则得到两个相等的实根为: