

中等专业学校计算机应用专业教材系列

郭启全 主编

QBASIC 程序设计

赵树忠 王贺艳 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.com.cn>

中等专业学校计算机应用专业教材系列

郭启全 主编

QBASIC 程序设计

赵树忠 王贺艳 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

本书较为详细地介绍了 MS-DOS 5.0 及其以上版本所提供的 QBASIC 语言环境下进行结构化程序设计的有关内容。

全书共分 12 章,包括程序设计的基本知识和 QBASIC 语言的基础知识;在 QBASIC 环境下进行结构化程序三种基本结构设计、使用数组和过程、进行字符串处理、文件操作、图形设计等方面的内容以及有关 QBASIC 程序开发环境的情况介绍。本书的体系合理、概念清楚、例题和习题丰富、图文并茂、通俗易懂。

本书是一本面向中等专业学校学生的教材,同时也可以作为程序设计语言爱好者的学习参考书。

图书在版编目(CIP)数据

QBASIC 程序设计/郭启全主编;赵树忠,王贺艳编著.

-北京:电子工业出版社,1998.11

中等专业学校计算机应用专业教材系列

ISBN 7-5053-4726-8

I . Q… I . ①郭… ②赵… ③王… III . BASIC 语言-程序设计

IV . TP312

中国版本图书馆 CIP 数据核字(98)第 24610 号

丛 书 名:中等专业学校计算机应用专业教材系列

书 名:QBASIC 程序设计

主 编:郭启全

编 著 者:赵树忠 王贺艳

责任编辑:应月燕

排版制作:电子工业出版社计算机排版室

印 刷 者:民族印刷厂

出版发行:电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:13.75 字数:352 千字

版 次:1998 年 11 月第 1 版 1999 年 10 月第 4 次印刷

书 号:ISBN 7-5053-4726-8
G·375

定 价:18.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。若书店售缺,请与本社发行部联系调换。电话:68279077

《计算机应用专业教材》序

随着计算机技术的迅速发展和普及应用,许多中等专业学校、技工学校和职业高中为了培养出适合社会需要的专门人材,设置了计算机应用专业。

计算机应用专业以培养计算机软件应用,计算机硬件及常用办公设备的应用、故障检测与维修的专业人材为目标。要求学生除了掌握必要的理论基础知识外,主要掌握计算机应用基础,电工及电子技术,办公自动化方面的软件,程序设计语言,计算机辅助设计与绘图,三维动画的制作,微机及其他常用办公设备(如打印机、复印机、传真机)的应用、故障检测与维修,计算机专业英语,多媒体技术,网络技术,数据库等专业知识。

计算机应用专业注重培养学生使用、保养与维修办公自动化设备的能力,注重培养学生熟练使用有关的计算机软件的能力。该专业的培养目标具有鲜明的特点,适应社会对人材的需求。由于毕业生主要面向厂、矿、企事业基层单位,因此能较快地解决基层单位计算机应用专业人材缺乏的问题。

计算机应用专业招收应届初中毕业生,学制三年。学生在校期间按照教学计划要学习以下5种类别的总计27门课程,并通过一定学时的实践教学,使学生既有扎实的理论基础,又有较强的动手能力。教学计划中突出了实践性教学,突出了课程设置的实用性。

教学计划中开设的课程如下:

1. 公共课

包括:数学、物理、语文、建设有中国特色社会主义理论与实践、道德与法律、英语、体育。

2. 专业基础课

包括:计算机专业英语、电工基础、计算机类电子电路基础学、计算机应用基础、中文 Windows 3.2/95、工程制图。

3. 专业课

包括:数据库原理与应用、QBASIC 语言程序设计、C 语言程序设计、中文 Word、中文 Excel、磁盘工具软件精选。

4. 实践教学

包括:微机的故障检测与维修、打印机(复印机、传真机)的故障检测与维修、微机操作训练。

5. 选修课

包括:三维动画设计、多媒体实用技术、计算机网络技术、计算机辅助设计与绘图。

在参与完成了计算机应用专业教学计划之后,有关部门委托我组织编写一套适合于该专业特点的系列教材。实用的教材是完成专业教学计划的保障。由于该专业设立的时间较短,市面上还没有与之配套的适合于这个办学层次的教材。在电子工业出版社的大力支持和帮助下,经过出版社领导、编辑们与作者的共同努力,使这套教材得以及时出版。

本套教材的作者均具有较丰富的教学经验和科研能力,其中有的同志编著过多本计算机应用方面的书籍,他们处于教学和科研的第一线,深知如何去编好这套教材。

本套教材结合了作者的教学、科研经验,适用性强,语言精练,通俗易懂。书中带有实用的

习题、实验题目、操作指导等。本套教材面向中专、技校、职高的广大学生,面向计算机的初、中级应用人员。由于水平所限,书中不足之处,望专家、读者指正。

郭启全

计算机应用专业教材编委会

主任 郭启全

委员 寇 森 裴桐松 高 嵩 刘 舒 刘 雄

刘文杰 袁德立 赵树忠 高松龄 刘浩宇

前 言

现代计算机技术正在以惊人的速度发展着。在计算机的各种应用中,程序设计是一个重要的工作内容。根据不同的应用特点,应用程序可以用不同的程序设计语言来编写,但有些程序设计语言不易于初学者掌握。对于初学者来说,最好先从较为简单、易于理解和掌握的一种程序设计语言开始学习。QBASIC 语言正是这样的一种程序设计语言,其特点是:用户可以方便地获得 QBASIC 系统(在 5.0 及以上版本的 DOS 操作系统中都提供了 QBASIC 系统);QBASIC 语言中的关键字均用人们比较习惯的英文或英文缩写表示,便于对关键字功能的理解;用 QBASIC 语言进行程序设计时可以基本上不考虑计算机的硬件组成;QBASIC 语言是一种结构化程序设计语言,符合现代程序设计的要求。学好 QBASIC 语言,可以为日后学习 QB(Quick BASIC)和 VB(Visual BASIC)以及其它程序设计语言奠定坚实的基础。

本书着重介绍在 QBASIC 语言环境下进行程序设计的基本知识和基本的程序设计方法。全书共分为 12 章。第 1~3 章主要介绍了计算机语言和程序设计的基本知识、QBASIC 程序开发环境和有关 QBASIC 语言的一些基础知识;第 4~6 章分别论述了在 QBASIC 环境下进行结构化程序的三种基本结构(顺序结构、选择结构和循环结构)设计的主要语句及编程方法;第 7、8 两章分别就数组、过程这两个重要的概念以及它们的使用做了论述;第 9~11 章分别介绍了在 QBASIC 环境下进行字符串处理、图形设计和文件操作的基本方法;第 12 章就数据库管理过程中的两个重要环节——排序与查寻做了简要介绍。

本书以使读者掌握最基本的 QBASIC 程序设计知识和程序设计方法为编写依据。全书图文并茂、文字通俗易懂,对各种语句、标准函数做了比较详细的说明,其中的部分内容是编者多年从事 QBASIC 程序设计的经验和体会。为配合有关内容的学习,各章中均安排了较为丰富的例题和习题,在例题中给出了程序清单、运行时的情况、运行结果以及必要的表示算法的 N-S 图,并对大部分例题中的程序设计思路做了比较透澈的分析。书中所有的例题均用 Pentium-133 机器在 DOS6.22 系统中的 QBASIC 环境下运行通过。

学习 QBASIC 程序设计语言时,根据编者的经验,读者应着重解决好以下三个环节:深入理解各种语句和标准函数的功能及使用方法;通过大量的编程练习不断提高编程技术;保证充分的上机训练时间,加强对有关知识的感性认识。

本书为了一本面向中等专业学校学生的教材,同时也可以作为其它学校相同课程的教材和程序设计语言爱好者的学习参考书。

由于编者水平有限,书中难免出现一些错误,敬请读者批评指正。

赵树忠 王贺艳

1998 年 4 月

目 录

第 1 章 程序设计基本知识	(1)
1.1 程序设计语言概述	(1)
1.1.1 计算机语言的层次	(1)
1.1.2 常用高级语言简介	(2)
1.1.3 QBASIC 语言的特点	(3)
1.1.4 高级语言的执行方式	(4)
1.2 算法及其表示	(5)
1.2.1 算法的概念	(5)
1.2.2 算法的表示	(6)
1.3 结构化程序设计概述	(8)
1.3.1 结构化程序设计的概念	(8)
1.3.2 结构化程序设计的方法	(9)
1.3.3 软件和软件生命期	(11)
习题	(12)
第 2 章 QBASIC 程序开发环境	(13)
2.1 开发环境基本情况介绍	(13)
2.1.1 QBASIC 环境的进入	(13)
2.1.2 QBASIC 的工作屏幕	(14)
2.1.3 程序的编辑与运行	(15)
2.1.4 联机屏幕帮助	(17)
2.1.5 QBASIC 环境的退出	(17)
2.2 常用菜单命令	(17)
2.2.1 File 菜单	(18)
2.2.2 Edit 菜单	(20)
2.2.3 View 菜单	(21)
2.2.4 Search 菜单	(23)
2.2.5 Run 菜单	(24)
2.2.6 Debug 菜单	(25)
2.2.7 Options 菜单	(26)
2.2.8 Help 菜单	(27)
习题	(29)
第 3 章 QBASIC 基础知识	(30)
3.1 源程序的结构与源程序的构成规则	(30)
3.1.1 QBASIC 源程序的结构	(30)
3.1.2 QBASIC 源程序的构成规则	(31)
3.2 数据结构与数据类型	(31)
3.2.1 数据结构与数据类型的概念	(31)

3.2.2 QBASIC 的数据类型	(32)
3.3 常量与变量	(34)
3.3.1 常量	(34)
3.3.2 变量	(35)
3.4 标准函数	(37)
3.5 运算符和表达式	(40)
3.5.1 算术运算符和算术表达式	(40)
3.5.2 字符串运算符和字符串表达式	(42)
3.5.3 关系运算符和关系表达式	(42)
3.5.4 逻辑运算符和逻辑表达式	(43)
习题	(45)
第4章 顺序结构程序设计	(48)
4.1 数据输入语句	(48)
4.2 数据输出语句	(54)
4.3 顺序结构中的其它语句	(61)
习题	(65)
第5章 选择结构程序设计	(68)
5.1 两分支选择结构	(68)
5.2 多分支选择结构	(72)
5.3 选择结构的嵌套	(79)
5.3.1 行 IF 语句的嵌套	(79)
5.3.2 块 IF 语句的嵌套	(80)
习题	(83)
第6章 循环结构程序设计	(84)
6.1 循环控制语句	(84)
6.2 循环的嵌套	(95)
6.2.1 循环嵌套的形式	(95)
6.2.2 循环嵌套的执行过程	(96)
6.2.3 循环嵌套的使用注意事项	(98)
6.3 循环的中途退出	(101)
6.3.1 用 EXIT 语句退出循环	(101)
6.3.2 通过给循环变量赋终值退出循环	(102)
6.3.3 通过改变循环语句中条件表达式的值退出循环	(102)
习题	(103)
第7章 数组	(106)
7.1 有关数组的基本概念	(106)
7.1.1 数组与数组元素	(106)
7.1.2 数组的表示方法	(107)
7.1.3 数组的分类	(108)
7.1.4 数组的存储结构	(108)
7.2 数组的定义与说明	(109)
7.2.1 说明语句 DIM	(109)
7.2.2 其它相关语句和函数	(111)
7.3 数组的引用	(113)

7.3.1 数组元素的赋值	(113)
7.3.2 数组元素的输出	(114)
7.4 数组应用综合举例	(115)
习题	(119)
第8章 过程	(121)
8.1 概述	(121)
8.2 FUNCTION 过程	(122)
8.2.1 FUNCTION 过程的定义	(122)
8.2.2 FUNCTION 过程的调用	(123)
8.3 SUB 过程	(125)
8.3.1 SUB 过程的定义	(125)
8.3.2 SUB 过程的调用	(126)
8.4 使用过程时的数据传递与变量的属性	(128)
8.4.1 数据传递	(128)
8.4.2 变量的属性	(130)
8.5 过程的嵌套调用与递归调用	(133)
8.5.1 过程的嵌套调用	(133)
8.5.2 过程的递归调用	(134)
8.6 过程应用综合举例	(135)
习题	(139)
第9章 字符串	(140)
9.1 概述	(140)
9.1.1 字符串的概述	(140)
9.1.2 字符串常量	(140)
9.1.3 字符串变量和字符串数组	(141)
9.2 有关字符串的基本操作	(141)
9.2.1 字符串的连接	(141)
9.2.2 字符串变量的赋值	(142)
9.2.3 字符串的比较	(143)
9.2.4 字符串的修改	(144)
9.2.5 字符串的输出	(145)
9.3 字符串处理函数	(145)
9.4 字符串应用综合举例	(150)
习题	(152)
第10章 数据文件	(154)
10.1 文件的基本概念	(154)
10.1.1 文件的概念	(154)
10.1.2 文件的组织方式	(154)
10.1.3 文件号和文件名	(155)
10.1.4 文件的分类	(155)
10.1.5 文件的基本操作	(157)
10.2 有关文件操作的函数	(157)
10.3 文件的打开与关闭	(160)
10.4 顺序文件的读写	(162)

10.4.1 向顺序文件写数据	(162)
10.4.2 向顺序文件追加数据	(163)
10.4.3 从顺序文件读数据	(163)
10.5 随机文件的读写	(165)
10.5.1 随机文件记录的定义	(165)
10.5.2 向随机文件写数据	(166)
10.5.3 从随机文件读数据	(167)
习题	(168)
第 11 章 图形	(169)
11.1 图形设计的基本知识	(169)
11.1.1 视频显示器与视频显示适配器	(169)
11.1.2 屏幕模式与屏幕坐标	(169)
11.2 屏幕控制语句	(171)
11.3 绘图语句	(175)
11.4 图形应用综合举例	(183)
习题	(185)
第 12 章 排序与查寻	(186)
12.1 排序与查寻的概念	(186)
12.1.1 排序	(187)
12.1.2 查寻	(187)
12.2 排序算法与程序	(187)
12.2.1 冒泡排序	(188)
12.2.2 插入排序	(189)
12.2.3 快速排序	(190)
12.3 查寻算法与程序	(192)
12.3.1 顺序查寻	(192)
12.3.2 二分查寻	(193)
习题	(195)
附录	(196)
附录 A 标准 ASCII 字符集	(196)
附录 B 键盘扫描码	(198)
附录 C QBASIC 的保留字	(198)
附录 D QBASIC 的语句命令及功能	(200)
附录 E QBASIC 的标准函数及功能	(205)
附录 F QBASIC 的限制	(207)
附录 G QBASIC 运行错误代码、信息及含义	(210)
参考文献	(210)

第1章 程序设计基本知识

1.1 程序设计语言概述

1.1.1 计算机语言的层次

语言可以分成两大类,即自然语言和人工语言。人类所用的语言一般称为自然语言,它们是人类在自身发展过程中为表达思想、交流经验、互通信息而自然形成的语言。人工语言是人们为某种目的而专门设计的语言。计算机是人类科技发展史上的一个进步,可以看成是人脑的延伸,但目前它还不能直接理解人类的自然语言,只能接受特定的机器代码。欲使计算机按人的意愿去处理某件事情,必须由人通过某种途径来生成计算机所能操作的代码。计算机语言就是用来生成这些可为计算机直接识别的代码的人工语言之一。计算机语言可以分成三个层次。

1. 机器语言

计算机基本的物理构成是复杂的电子线路。这些线路可以在不同的电信号作用下,通过各种装置产生不同的动作。计算机的核心部分(CPU、存储器等)只能识别高、低电平两种状态,即二进制的0、1。由0、1的不同组合所形成的可为计算机直接识别的二进制指令代码的集合就称为机器语言。计算机在接收到这些二进制代码后,由译码器产生各种控制信号送往有关的单元,从而产生相应的操作。

机器语言是最基本的、出现最早的计算机编程语言,是唯一可为计算机直接执行的语言。用机器语言编写的程序冗余小,执行效率高,节省内存,运行速度快,可以直接控制计算机的硬件。但不同类型计算机的硬件结构可能有所不同,它们各自具有不同的指令系统,因此用机器语言编程对程序设计者的水平要求很高,他们必须对所用计算机的硬件工作原理以及线路连接关系十分清楚,而且一般在某种机器上编写的程序不能在另一种机器上运行,因此人们称之为“面向机器”的程序设计语言。此外,用机器语言编写的程序可读性差、容易出错,出错后很难查找,因此,人们往往不直接用机器语言进行编程,而是借助于其它更为简单、编程效率更高的计算机语言,通过间接的方法来产生可为计算机直接接受的代码。

2. 汇编语言

为克服机器语言的缺点,便于一般人员进行编程,50年代初出现了汇编语言。其特点是引入了一些助记符来表示操作,用一些特定的符号来表示计算机中的某些单元,如用ADD表示加运算,用AL、AH表示某一寄存器等。这些助记符比较接近英文,其它符号也比较简单,便于记忆、理解,从而简化了编程过程。

用助记符和有关符号编写的程序称为汇编语言程序。由于计算机只能识别二进制代码,不能识别这些符号,因此,还必须通过某种方法将汇编语言程序“翻译”成相应的二进制代码,由这些二进制代码组成的程序称为目标程序,“翻译”的过程称为汇编。汇编的方法有人工汇

编和机器汇编两种。人工汇编是由人根据机器的指令系统通过查表进行转换,机器汇编则是借助于专门的汇编程序由计算机来完成转换。

用汇编语言进行编程也要求编程人员有丰富的计算机硬件知识,这是因为汇编语言仍然是一种“面向机器”的程序设计语言,编程过程离不开机器的硬件结构,助记符、符号及汇编过程要依赖于机器的特定指令系统。所以汇编语言程序的通用性仍然较差,应用范围受到一定的限制。但汇编语言程序仍然具有运行速度快、能直接控制硬件等特点,在实时性要求较高的计算机自动控制系统中或对计算机硬件设备要求特殊管理、编写系统软件的情况下仍得到了广泛的应用。

3. 高级语言

机器语言和汇编语言都是“面向机器”的程序设计语言,习惯上称它们为“低级语言”。由于它们难学、难记、程序难调试、可移植性差,在非计算机专业人员中不易推广。随着计算机的普及和人们解决日益增加的实际问题的需要,出现了各种形式的高级语言。

高级语言是普及型的计算机程序设计语言,其各种命令的形式接近于自然语言(英语)和数学算式的格式表示。它们各自有着严格的语法规则,便于记忆、书写、阅读和修改。用高级语言编写出的程序的每一条命令从字面上就能看出其含义。高级语言基本上摆脱了机器类型的影响,程序设计者在进行程序设计时可以不考虑机器的硬件结构,只需要掌握应用问题的解决方法和有关的算法,按照语言的语法规则书写命令,就可以编出程序。用某种高级语言编写出来的程序称为该语言的源程序。各种高级语言有它们各自的特点,但它们的源程序都不能直接被计算机所执行,也必须经过一个翻译过程才能转换成可为计算机直接执行的目标程序。

用高级语言编程的方法和思路很接近人与人之间的自然语言交流和数学描述,因此编程效率高,编程时基本上不涉及计算机的硬件知识,便于普及,程序的通用性好。但高级语言程序不如机器语言简练,翻译转换后生成的目标程序冗余大,运行时占用内存多,速度较慢。

1.1.2 常用高级语言简介

自 1954 年诞生了第一种高级语言——FORTRAN 语言以来,到目前为止已陆续出现了上百种计算机高级语言。高级语言的发展趋势是标准化、模块化、结构化。这些高级语言各有各的特点,所侧重的应用领域不同,除少数几种高级语言外,基本上是不同的领域使用不同的高级语言。比较常用、流行的高级语言主要有以下五种。

1. FORTRAN 语言

FORTRAN 语言是世界上最早出现的高级语言,它特别适合于科学和工程中的数值计算。它所支持的数据类型比较丰富,可以进行复数运算。此外,它的数据输入输出格式也很灵活。

2. COBOL 语言

COBOL 语言是专门为处理商业事务而设计的一种通用语言,适用于商业和管理领域。其特点是针对商务和管理过程中要处理大量记录形式的数据的特殊需要,侧重于数据项和输入输出记录的处理。

3. PASCAL 语言

PASCAL 语言是最早出现的结构化程序设计语言,它特别强调程序的可靠性和易于验证性。用它编写出来的程序结构性好,程序易读。这种语言广泛用于一些计算机课程的教学,也可用于科学计算。

4. C 语言

C 语言是一种通用的结构化程序设计语言,特别是在软件工程编写系统软件方面应用很广。它既有高级语言的诸多特点,也具有汇编语言控制硬件的能力,还可比较方便地调用其它语言编写的程序,因此人们习惯把它称为“中级语言”,其应用日益广泛。随着 Microsoft Windows 的出现,近年来又出现了可视化的 Visual C 语言。

5. BASIC 语言

BASIC 语言是一种易学易用、易推广又有较高使用价值的高级语言,特别适合于初学者学习使用。1964 年产生了最早的第一代 BASIC 语言,此后又出现了非结构化的第二代 BASIC 语言,其代表是 GWBASIC、BASICA 等。70 年代出现了结构化程序设计的思想以后,又推出了一些结构化的第三代 BASIC 语言,主要有 True BASIC、Quick BASIC、Turbo BASIC 以及随 DOS5.0 版本问世的 QBASIC 等。第三代 BASIC 语言的特点是易学易懂,编程完全符合结构化程序设计的要求。1991 年,Microsoft 公司又隆重推出了在 Windows 环境下运行的具有强大可视性的 Visual BASIC。它可以充分享用 Windows 的资源,可以在图形界面下开发具有良好用户界面的 Windows 应用程序,并且高度兼容 Quick BASIC、QBASIC 等早期 BASIC 语言编写的程序。Visual BASIC 是第四代 BASIC 语言的典型代表。

1.1.3 QBASIC 语言的特点

QBASIC 是由 Microsoft 公司开发的,它作为 MS-DOS 5.0 以上版本的一个组成部分出现,即安装有 MS-DOS 5.0 以上的用户都可方便地使用 QBASIC 进行编程。QBASIC 是一种结构化、模块化的程序设计语言,它采用了先进的解释方式执行程序,在保留以前 BASIC 语言的特点的基础上,又扩充了许多功能。主要特点如下:

(1) 变量和常量的数据类型更加丰富

QBASIC 允许变量名最长可达 40 个字符,所支持的变量和常量的数据类型有整型、长整型、单精度实型、双精度实型、字符串型、定长字符串型以及用户自定义型,比以前的 BASIC 语言更为丰富。

(2) 提供了块 IF 语句

在此之前的 BASIC 语言只有行 IF 语句,其功能有限,结构不清晰,不利于结构化程序设计。块 IF 语句使选择结构的使用更加灵活,便于进行结构化程序设计。此外,QBASIC 提供的多分支选择语句 SELECT CASE 语句的功能也比以前 BASIC 语言的 ON...GOTO 语句有了增强。

(3) 循环控制语句灵活多样

除具有以前 BASIC 语言的 FOR...NEXT 循环以外,QBASIC 又增加了 WHILE 和 DO 循环语句,使循环控制更加细腻。此外,QBASIC 语言程序可以不使用行号(也可以使用),限制使用 GOTO 语句进行流程的跳转,这些都保证了程序设计更符合结构化的要求。

(4) 提供了独立的 SUB 子程序模块和 FUNCTION 函数模块

这些模块也称为过程。它们的独立性很强,相互之间可以调用,模块中定义的变量都是局部变量,多个模块可以同时驻留内存,大大方便了程序的模块化设计,增加了程序的通用性。

(5) 兼容旧的基本版本

用 BASIC、BASICA、GW BASIC 编写的程序,不加修改就可以在 QBASIC 环境下运行。

(6) 具有灵活的在线(联机)帮助功能

QBASIC 语言提供了灵活的在线帮助系统,可以为每个语句提供语法检查,随时可为保留字、菜单命令、直接键和 ASCII 码提供帮助信息。

(7) 具有良好的编辑环境和独特的交互式编辑系统

QBASIC 提供了全屏幕程序编辑窗口以及一个可以随时显示指定命令(或表达式)结果的命令窗口。此外,还提供了菜单功能,用户既可以使用键盘也可以使用鼠标进行各种编辑、调试及运行操作。编辑时,每输入一个程序行,系统马上就对其进行语法检查,若有错误则给出出错信息,否则会自动将关键字转换成大写字母,同时自动调整语句中各项间的空格,并将程序行翻译成可执行代码。

(8) 具有较强的图形功能

QBASIC 可以支持 CGA、EGA、VGA、MCGA 等多种图形显示适配器和多种图形方式。它所提供的图形控制语句比较丰富,可以设置多种显示模式、颜色、线型、填充等。

1.1.4 高级语言的执行方式

用高级语言编写出来的源程序是不能直接被计算机执行的,必须要通过特定的翻译程序将其转换成机器语言才能为计算机所执行。对高级语言源程序进行翻译的方式有两种,即编译执行方式和解释执行方式。大部分高级语言只有一种执行方式。

1. 编译执行方式

源程序编辑好以后,利用高级语言系统自身所具备的编译工具(功能),一次将源程序统一翻译成机器语言程序,该程序称为源程序的目标程序,该过程称为编译。如果源程序有错误,则编译时给出所有的出错信息,不生成新的目标程序,只有当用户将全部错误改正以后,编译才生成最终的目标程序文件。该目标程序是由二进制代码组成的机器语言文件,可以被计算机执行。

编译执行方式有如下的特点:

(1) 一次将源程序翻译成目标程序后,该目标程序可多次运行而无需再进行翻译。

(2) 编译方式所形成的目标程序的装载和运行速度比一般的解释方式要快。

(3) 编译后所形成的目标程序一般可脱离相应的语言环境运行。

(4) 编译后的目标程序的运行不需要源程序,有利于源程序的保密。

(5) 不同语言环境下生成的目标程序(包括汇编得到的目标程序)可以互相调用,有利于充分发挥各种语言的长处。

(6) 编译过程中可一次发现程序中的全部错误,且修改方便(有些错误是互相关联的,可能修改一处就可消除其它的一些错误)。

(7) 源程序有任何错误都不能生成目标程序。

(8) 源程序变动后必须重新编译才能生成新的目标程序。

由于编译方式具有很多优点,所以大部分高级语言都采用这种执行方式,如 Quick BASIC、Turbo C 等。

2. 解释执行方式

这种执行方式是直接运行源程序。运行时逐句将源程序翻译成机器语言(称为解释)调入内存加以运行,解释一句,运行一句,不在磁盘上生成永久的目标程序。若程序中的某些语句有错误,则执行到出错处即停止运行,并给出出错的位置及出错的类型(有些系统还给出出错的原因),便于及时修改。

解释执行方式的优点是不需要编译工具故整个语言系统较小,出错前的程序可正常运行,可以得到部分运行结果,便于用户调试。不足之处是由于解释一句运行一句,致使一般运行速度较编译方式要慢,每次运行程序都需要计算机进行解释,较大程序的调试比较麻烦。早期的 BASIC 语言都采用解释执行方式。QBASIC 采用的是较为先进的解释方式,其环境下的运行速度与一般的解释方式相比是比较快的。

1.2 算法及其表示

欲使计算机做某件事情,首先必须由人为其设计出一系列操作内容和步骤,然后借助于计算机语言将其表示出来,这就是程序设计。程序最终以二进制机器语言的形式为计算机接受,从而产生一系列相应的操作。为了有效地进行程序设计,一方面要掌握解决问题(包括数值问题和非数值问题)的方法和步骤,另一方面还要能熟练地用某种计算机算法语言将其正确地表示出来。

1.2.1 算法的概念

所谓算法,就是人们为解决具体问题所采取的方法和步骤。请看下面的例题。

【例 1.1】将两个变量 X、Y 的值进行互换。

分析:计算机是在内存中用一些存储单元来存放某一类型的变量值的。两个变量值的互换是不能直接进行的,必须要借助于第三个变量,采取间接交换的方法实现。为此,我们可以在内存中开辟出一些存储单元,记为中间变量 Z。本题的一种算法如下:

步骤 1 将变量 X 的值暂存于中间变量 Z 中;

步骤 2 将变量 Y 的值传送给变量 X;

步骤 3 将中间变量 Z 中暂存的变量 X 的原值传送给变量 Y。

解决某一问题的算法不是唯一的,上面的例题还有其它的算法,读者不妨可以试写一下。对于程序设计者来说,一方面要善于利用现有的一些算法,更重要的是会设计出新的优化算法。所谓优化算法,指的是人机开销较少的算法,即算法和程序的设计费用较低、运行时间较短、占用内存较少的算法。

计算机算法可以分成两大类:数值运算算法和非数值运算算法。数值运算的最终目的是为了得到某一数值表达式的值,如求乘积、积分值等。非数值运算包括的范围很广,最常见的是事务管理,如图书检索、人事工资管理等,计算机在这方面的应用远远超过其在数值运算方面的应用。【例 1.2】给出的就是学生成绩管理中的一种非数值运算算法。

【例 1.2】有 30 个学生的成绩,要求把成绩大于等于 60 分的学生姓名、成绩打印出来。

分析:本题有两个特点:一是 30 个学生的姓名、成绩打印与否取决于其成绩是否大于等于 60 分;二是对 30 个学生的成绩都要做同样的判断。结合这些特点,可以采用循环处理方法,在循环中依据给定的条件进行是否打印的判断。若用 $i(i=1,2,\dots,30)$ 表示学生的序号, N_i 表示第 i 个学生的姓名, S_i 表示第 i 个学生的成绩,则算法可表示如下:

步骤 1 预置学生序号为 1 ($1 \rightarrow i$);

步骤 2 对第 i 个学生的成绩 S_i 进行判断,若 $S_i \geq 60$,则打印该学生的姓名 N_i 、成绩 S_i ,否则不打印;

步骤 3 学生序号加 1 ($i+1 \rightarrow i$);

步骤 4 如果学生序号超过 30 ($i > 30$, 表示已经处理完 30 个学生),则算法结束,否则 ($i \leq 30$) 返回步骤 2 处理下一个。

算法是操作步骤的有穷集合。一个算法,应该具有如下的特性:

(1) 有穷性。一个算法必须只包括有穷个操作步骤。

(2) 确定性。算法中的每个步骤都必须有明确具体的解释,不能模棱两可。

(3) 可行性。算法中的每个步骤都必须能有效地执行,并可得到确定的结果。

(4) 输入。一个算法有零个或多个输入作为操作对象。输入可以是各种类型的,如数值、字符或字符串等。

(5) 输出。即问题的结果。一个算法可以有一个或一个以上的输出,它们往往是与输入具有某种特定关系的。

1.2.2 算法的表示

算法要用适当的方式表示出来,以便程序设计者可以按照其中的步骤编写出程序。算法的表示方法对编程的质量有很大的影响,在程序设计过程中,经常采用的算法表示方法主要有三种。

1. 用自然语言表示算法

自然语言就是人们日常使用的语言。【例 1.1】和【例 1.2】中的算法就是用自然语言表示的。使用自然语言描述算法的最大优点是人们比较习惯,通俗易懂,即使没有学过计算机的人也很容易接受,缺点是文字冗长,语义、语法灵活多变,所表示的含义往往不太严格,容易出现“歧义”。特别是在表示具有多分支选择和循环的算法时不是很方便。因此,除一些比较简单的算法外,一般不用自然语言描述算法。

2. 用传统流程图表示算法

传统的流程图是用一些图框、流向线、文字说明来表示算法一种图形。有关用传统流程图表示算法的一些基本图框、图框的意义和长度比例等都已由美国标准化协会 ANSI(American National Standard Institute)做了具体的规定,现已为各国计算机界所普遍采用。我国颁布的国家标准(GB ISO5807-85)中所做的部分规定如图 1.1 所示。

起止框表示算法的开始或结束,它只有一个出口(起始框)或一个入口(结束框)。处理框有一个入口和一个出口,其中可以用简明的文字说明具体的处理操作。判断框内用文字注明判断的条件,它有一个入口和两个出口 Y(条件满足时的流向)、N(条件不满足时的流向)。输入/输出框、准备框也都具有一个入口和一个出口,框中注明具体的操作内容。注释框可以引

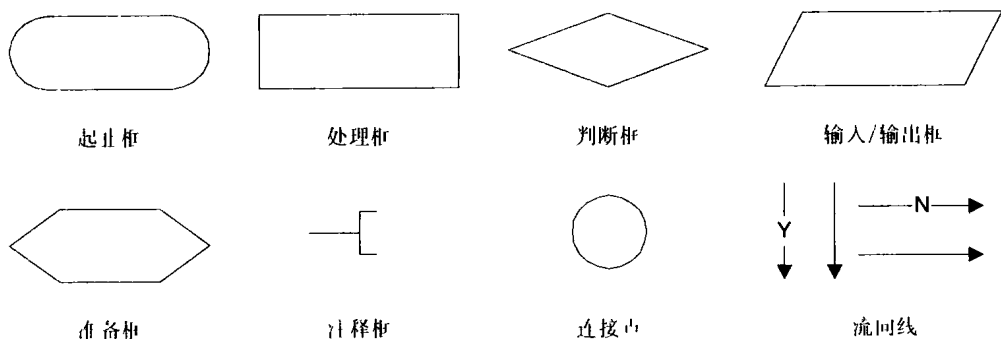


图 1.1

出对算法中的某些部分所做的补充说明。连接点用于表示画在两页或两处的同一流程图之间的连接关系，一般在对应连接点的圆框中注上同一字母。带箭头的流向线表示算法的流向。图 1.2 给出了表示【例 1.2】算法的传统流程图。

用这种传统流程图表示算法，比用自然语言表示算法明确简练，形象直观，流向清楚，有利于用计算机语言加以实现，因此广泛用于各种高级语言的程序设计中。它的缺点是，流向线的长短、方向任意性太大，对于比较复杂的问题，要详细地画出这种流程图，图形就显得特别长，流向线较多，给人以一种眼花缭乱的感觉。随着结构化程序设计方法的推广，这些缺陷日益明显。最近，国际标准化组织 ISO(International Standard Organization)在 ANSI 提出的符号的基础上又做了一些改进，主要是增加了表示循环的符号，改进了多分支处理的符号。

3. 用 N-S 图表示算法

尽管传统的流程图有一定的优点并对其做了部分改进，但对于结构化程序设计仍嫌不足。经验表明，一般的程序都可以由顺序结构、选择结构和循环结构三种基本结构组成。1973 年，美国学者 Nassi 和 Schneiderman 提出了一种新的绘制流程图的方法，并以二人名字的首字母命名为 N-S 图。三种基本结构的 N-S 图画法如下：

(1) 顺序结构 由若干个矩形框组成，画法如图 1.3 所示。其含义是：先执行块 A，再执行块 B。框中填写具体的操作内容。

(2) 选择结构 由若干个分支构成。图 1.4 给出了两分支结构的 N-S 图。条件 C 满足(真)时执行块 A 的操作，不满足(假)时执行块 B 的操作。

(3) 循环结构 循环结构有两种：当(WHILE)型循环结构和直到(UNTIL)型循环结构。它们的 N-S 图的画法分别如图 1.5(a)和图 1.5(b)所示。当型循环先判断循环条件是否满足，满足时执行循环体的操作，然后返回继续判断循环条件是否满足，若满足则再次执行循环体的操

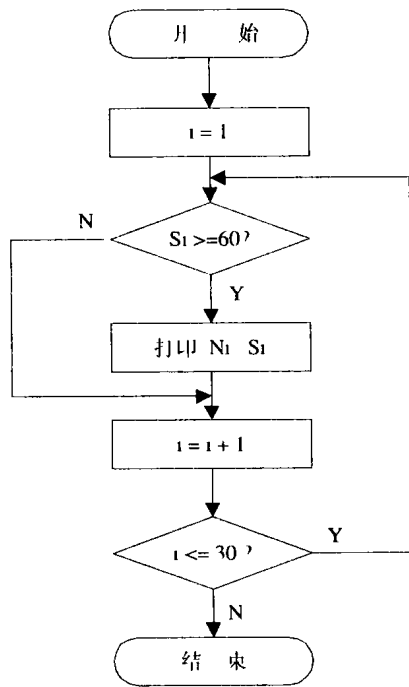


图 1.2