

高等学校计算机基础教育系列教材

计算机**技术基础**

主 编 刘士杰
副主编 邹光华 王虹桥
孟晓春

煤炭工业出版社

前 言

高等学校非计算机专业学生的计算机教育是高等教育的组成部分,是实现高等教育培养目标的一个重要环节。高等学校的学生应具有基本的计算机文化基础、技术基础和熟练应用计算机的能力。

随着计算机教学改革的不深入,我们对传统的计算机语言教学进行了改革,放弃了过去非计算机专业的学生只学一门结构化语言或可视化语言的做法。我们认为非计算机专业的学生,最好能先学习一门结构化程序设计语言,掌握结构化程序设计的基本思想和方法,为进一步学习可视化程序设计和进行计算机应用奠定坚实的基础。然后,学习一门可以开发 Windows 环境下应用程序的可视化语言,如: Visual Basic, Visual Foxpro 等。基于这一思想,我们在多个班教学实践的基础上编写了这本《计算机技术基础》一书,其目的是满足高等学校非计算机专业理工科学生学习计算机的需要,在介绍一些计算机的软、硬件基础知识的基础上,重点学习一门结构化程序设计语言——QBasic,并在此基础上,进一步学习一门可视化的程序设计语言——Visual Basic。随着计算机应用技术的发展, Basic 程序设计语言在编程环境、结构化程度与功能上也不断发展,而且越来越强,越来越实用。发展到今,结构化和可视化的 Basic 应用十分普及,可以说 Basic 系列是实现程序设计非常好的语言之一。此外, QBasic 与 Visual Basic 兼容,因此,学习完本书后,可将两者有机地结合起来,为本专业的进一步学习和应用程序的开发打下良好的基础。

本书围绕着原国家教委制定的《加强工科非计算机专业计算机基础教学工作的几点意见》的要求,结合 QBasic 与 Visual Basic 中的重点内容,参考了国家计算机等级考试(二级)中 QBasic 的大纲要求,编写了这本具有基础性、先进性、实用性三大特点的《计算机技术基础》教材。本书是在高等学校计算机基础教育系列教材的第一个层次《计算机文化基础》的基础上编写的第二个层次的教材,可供高等学校非计算机专业,特别是理工科专业的学生,以及各类办公人员、工程技术人员学习和使用。

本书避免了繁琐的理论阐述,强调以应用为主,加强了基础知识、基本技能的训练,加大了实用内容和新内容的篇幅。本书具有内容新颖,易学、易用,通俗易懂,图文并茂等特点,并通过典型实例的讲解,使学生能很好地掌握所学知识和操作要领。各章后附有习题和上机操作题,便于学生课后学习和上机操作。

由于本书内容较多,讲授时可以根据课时数和教学对象选择不同的教学内容。

本书由刘士杰任主编,邹光华、王虹桥、孟晓春任副主编,其中第一章、第十二章由刘士杰编写;第二章由张瑾编写;第三章、第五章由邹光华编写;第四章、第十三章由王虹桥编写;第六章由孙改平编写;第七章由王秀和编写;第八章、第十一章由李跃辉编写;第九章由席振元编写;第十章由曹德胜编写。

由于作者水平有限,书中难免存在缺点和错误,殷切希望读者批评指正。

作 者

2000 年 8 月

目 录

第一章 QBASIC 程序设计基础	1
第一节 程序设计与 QBASIC 简介	1
一、程序设计概述	1
二、算法与流程图	2
三、QBASIC 简介	5
第二节 QBASIC 程序的基本结构	6
一、一个简单的 QBASIC 程序	6
二、QBASIC 程序行	7
三、QBASIC 的关键字	8
四、复杂程序的结构	8
第三节 QBASIC 程序设计的基础知识	9
一、QBASIC 的基本数据类型	9
二、常量	11
三、变量	12
四、表达式与运算符	15
五、标准函数	20
六、几个常用的基本语句	26
第四节 QBASIC 的运行环境	31
一、QBASIC 的启动与退出	31
二、QBASIC 的功能菜单	32
三、QBASIC 程序的编辑和运行	34
习题	36
上机操作题	39
第二章 基本控制结构的程序设计	40
第一节 顺序结构程序设计	40
一、顺序结构程序设计的概念	40
二、读数、置数语句(READ...DATA)	40
三、恢复数据指针语句(RESTORE)	42
四、交换变量值语句(SWAP)	43
五、程序暂停语句(STOP)	44
六、清屏语句(CLS)	44
七、注释语句(REM)	44
八、几种赋值语句的比较	45
九、程序举例	46
第二节 选择结构程序设计	47

一、选择结构程序设计的概念	47
二、单行结构 IF 语句	48
三、块结构 IF 语句	49
四、单条件的嵌套语句 (IF...ELSEIF...END IF)	51
五、多分支选择语句 (SELECT CASE)	53
第三节 循环结构程序设计	58
一、循环结构程序设计的概念	58
二、固定次数的循环语句 (FOR...NEXT)	58
三、条件循环语句 (WHILE...WEND)	67
四、先判断条件的 DO...LOOP 循环语句	70
五、先执行循环, 再判断条件的 DO...LOOP 循环语句	71
六、三种循环结构的比较	74
第四节 综合习题分析	75
习题	79
上机操作题	83
第三章 QBasic 数组、子程序与过程	85
第一节 数组	85
一、数组和数组单元的概念	85
二、数组的建立	86
三、一维数组和二维数组	87
四、静态数组和动态数组	90
五、应用举例	92
第二节 子程序与过程	94
一、自定义函数	95
二、子程序	96
三、过程	98
习题	109
上机操作题	112
第四章 屏幕控制与作图	114
第一节 屏幕显示模式与屏幕控制	114
一、屏幕显示模式简介	114
二、屏幕控制语句	114
三、显示模式控制语句 (SCREEN)	116
四、设置颜色语句 (COLOR)	117
第二节 画图	120
一、几种基本画图语句	120
二、填充颜色语句 (PAINT 语句)	127
三、综合举例	128
习题	129
上机操作题	131

第五章 文件	133
第一节 文件基本概论	133
一、文件说明	133
二、记录	133
三、文件的分类.....	134
四、文件号	135
第二节 文件操作语句和函数	135
一、打开文件语句(OPEN)	135
二、关闭文件语句(CLOSE)	136
三、文件操作语句	137
四、文件操作函数	138
第三节 顺序文件	139
一、顺序文件的输出(顺序写文件)	139
二、顺序文件的输入(顺序读文件)	140
三、顺序文件中数据的添加、检索、排序和修改	142
第四节 随机文件	145
一、随机文件的输出(随机写文件)	145
二、随机文件的输入(随机读文件)	147
习题	149
上机操作题	150
第六章 Visual Basic 编程基础	152
第一节 Visual Basic 概述	152
一、什么是 Visual Basic	152
二、Visual Basic 的特点	153
三、软硬件要求	154
四、Visual Basic 版本简介	154
第二节 Visual Basic 中文版的集成开发环境	155
一、运行 Visual Basic 中文版	155
二、Visual Basic 的窗体设计器	155
三、工程资源管理器	156
四、工具箱窗体	157
五、属性窗口	158
六、代码窗口	159
七、窗体布局窗口	161
第三节 如何使用 Visual Basic 建立第一个应用程序	162
一、简单应用程序设计	162
二、Visual Basic 的相关概念	170
三、输入、输出对话框	173
四、QBasic 与 Visual Basic 中的部分区别	176
第四节 调试与错误处理	181

一、程序错误	181
二、Visual Basic 调试工具	183
三、错误捕获及处理	184
习题	186
上机操作题	186
第七章 设计应用程序的界面	188
第一节 设计窗体	188
一、窗体的建立、添加和删除	188
二、窗体的属性和事件	189
三、向窗体中添加控件	193
四、启动窗体的设置	194
五、窗体的生命周期	196
第二节 VB 中常用的控件	198
一、VB 的控件简介	198
二、VB 的内部控件	198
三、VB 的 ActiveX 控件	230
第三节 设计多文档界面(MDI)	233
一、界面样式	233
二、多文档界面(MDI)中窗体之间的关系	233
三、创建 MDI 应用程序的方法与步骤	234
第四节 发行 Visual Basic 应用程序	235
一、创建安装程序	235
二、使用 Visual Basic 安装向导	236
习题	239
上机操作题	239
第八章 设计菜单和工具栏	244
第一节 菜单编辑器窗口简介	244
第二节 设计菜单	246
一、菜单栏设计	246
二、添加代码	248
三、建立快捷菜单	250
四、运行时操作菜单	254
第三节 设计工具栏	255
一、工具栏的常用属性、方法和事件	255
二、创建工具栏	256
三、为工具栏的各按钮编写处理过程	259
四、应用实例	260
习题	263
上机操作题	263

第九章 设计图形和多媒体程序	265
第一节 图形控件介绍	265
第二节 线条和形状控件	265
一、线条控件	265
二、形状控件	266
第三节 图像框和图片框控件	268
一、图像框控件	268
二、图片框控件	272
第四节 图形控件的应用	276
一、建立绘制正弦曲线的程序	276
二、图形查看实例	278
第五节 使用 Animation 控件	281
一、Animation 控件的相关属性、方法和事件	281
二、使用 Animation 播放 AVI 文件的方法	281
第六节 多媒体集成控件 MCIWnd	282
一、MCIWnd 控件的属性和事件	282
二、使用 MCIWnd 控件制作播放器	283
三、完善媒体播放器	283
第七节 利用 API 函数编写多媒体应用程序	284
一、如何添加 API 函数	284
二、利用 mciExecute() 函数建立简单的多媒体程序	285
习题	286
上机操作题	287
第十章 创建文件应用程序	288
第一节 文件系统概述	288
第二节 文件的读写	288
一、顺序文件	288
二、随机文件	291
三、二进制文件	297
第三节 与文件相关控件的介绍	298
一、驱动器列表框	298
二、目录列表框	299
三、文件列表框	300
习题	301
上机操作题	301
第十一章 数据库技术	302
第一节 用数据控件访问数据库	302
一、数据库的基本概念	302
二、使用数据控件访问数据库	303
三、用 Data 控件创建简单的数据库程序	304

四、使用 DbGrid 浏览数据库	305
五、记录的自动添加	306
第二节 使用可视化数据管理器	307
一、创建新的数据库及表	307
二、数据库记录的添加、删除和查询	309
第三节 用程序代码操作和创建数据库	311
一、用程序代码操作数据库	311
二、用程序代码创建数据库	314
第四节 SQL 语言简介	315
一、SQL 语言的组成	315
二、利用 SQL 语言进行查询	316
习题	318
上机操作题	318
第十二章 计算机软件基础知识	320
第一节 数据结构基础	320
一、线性表	320
二、栈和队列	322
三、树和二叉树	323
四、查找和排序	325
第二节 软件的工程设计方法	326
一、软件工程的基本概念	326
二、软件的生命周期	327
三、软件开发方法	328
习题	332
第十三章 微型计算机硬件技术基础	333
第一节 微机与微机系统	333
一、微处理器、微型计算机和微型计算机系统	333
二、微机的特征和特点	334
三、单片微型计算机	335
第二节 微机中的信息、接口和总线	338
一、微机中的信息	338
二、CPU 与外设之间的信息传送方式	339
三、微机中接口的组成、功能和指标	340
四、微机中的总线结构标准及扩展槽	341
第三节 CMOS 中 BIOS 参数的设置	343
一、AWARD BIOS 参数的设置	343
二、AMI BIOS 参数的设置	346
第四节 使用微机时常遇到的问题及处理措施	348
一、关于主机板与 CPU 的问题	348
二、关于外设方面的问题	350

三、关于内存方面的问题	352
四、关于显示卡和显示器的问题	353
五、关于多媒体方面的问题	354
习题	358
附录 I 常用字符与 ASCII 码对照表	359
附录 II 全国计算机等级考试二级考试新大纲	360

第一章 QBasic 程序设计基础

第一节 程序设计与 QBasic 简介

一、程序设计概述

(一) 计算机的工作原理与程序的概念

计算机工作时,只有当用户向计算机输入一定的信息,它才能按照用户的要求进行工作并得到所需要的结果。目前的计算机的运行机制基本是:

(1) 将需要计算机完成的任务编成一条条指令,输入计算机,存放在计算机的内存存储器中,称为编程序。

(2) 计算机工作时,从内存中取出指令,然后再执行它们,称为运行程序。

因此,计算机是通过一条一条的指令来完成工作的。人们以一条一条的指令来控制计算机,使它按照人们的要求工作。用计算机术语来说,指令的序列被称为程序。也可以说,程序是为了解决某一问题或完成某项任务的指令的有序集合。只有当用户向计算机输入一定的程序,计算机才能按照程序规定的步骤进行工作。失去程序控制,计算机将无法发挥其作用。

(二) 程序设计的基本步骤

人们通过程序让计算机工作,以便处理各种信息,解决各种问题,让计算机为人们服务。所以“程序”是人们意志的体现。编程序中用户通过程序设计语言对信息进行加工、处理并输出预期的结果。编制程序的基本过程如下:

(1) 分析问题。为了解决一定的实际问题,编程序时不能无的放矢,而要认真考察和分析实际问题,确定有哪些给定的数据,需要进行哪种处理,需要输出哪些数据等等,找出解决问题的大致思路。

(2) 提出算法。把第(1)步中分析问题的思路进一步明确化、详细化,建立解决问题所需要的数学或物理模型。

(3) 编写程序。根据第(2)步的方案用一定的计算机语言把程序编写出来。

(4) 上机调试。先录入程序,然后对编好的程序运行调试,进行实际检验。调试过程中,对其中的错误不断加以改正,直到程序能达到预期的结果为止。

(5) 运行程序。将程序投入运行,并输出结果。

二、算法与流程图

在系统地学习 QBasic 语言程序设计知识之前，先介绍算法与流程图，它们是程序设计的重要工具。

(一) 算法

算法就是对问题求解方法的精确描述。在程序设计时，最关键的问题是算法的提出，因为它直接关系到编写出来的程序的正确性、可靠性和完整性。一个计算机能实现的算法应具有以下特征：

(1) 有穷性。一个算法应包含有限的操作步骤，而不能是无限的。因此，在算法中必须给出结束的条件。

(2) 确定性。一个算法中的任何步骤都必须意义明确，不能模棱两可、含糊不清，即不允许有二义性。

(3) 可行性。所采用的算法必须能在计算机上执行，因此，在算法中所有的运算必须是计算机能够执行的基本运算。例如， $A \neq 0, B = 0$ ，执行“ A / B ”就是一个不可执行的操作，算法中不应出现。

(4) 有一定的输入与输出。要计算机解决问题时，总是需要输入一些原始数据；计算机向用户报告结果时，总是要输出一些信息。因此，一个算法中必须有一定的输入与输出。

(二) 流程图

流程图是一种用图形、文字和符号来比较形象地描述“算法”的工具。它对于编制程序很有帮助。

1. 流程图的图形符号

流程图又称为框图，它是由几种不同的图形符号组合而成的。流程图常用图形符号如图 1-1 所示。图中，各图形表示的意义如下：

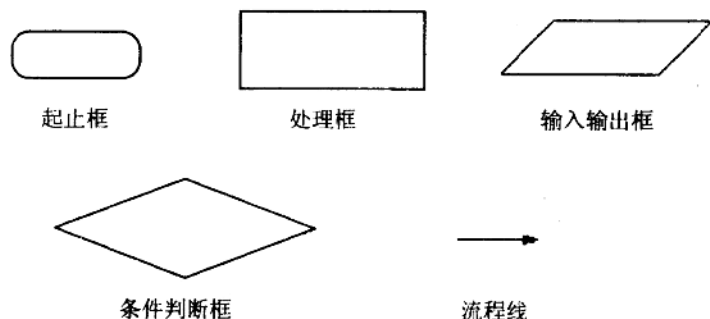


图 1-1 流程图常用图形符号

(1) 起止框，它代表一个算法的开始与结束之处。

(2) 处理框，它表示算法中对问题处理的一个或若干个步骤，如：赋值、计算等，这些步骤不涉及输入与输出。它用矩形框表示。

(3) 输入输出框，它表示算法中需要进行输入或输出处理的步骤。它用平行四边形表示。

(4) 条件判断框，它表示算法中需要根据条件判断的步骤。它用菱形框表示。

(5) 流程线，它表示算法的每一个步骤之间的先后顺序，标示着程序的走向。

2. 流程图的作用

流程图在程序设计中起着重要作用，主要有以下几点：

(1) 流程图既是程序设计者解决问题时逻辑思维的具体反映，也是计算机算法的形象描述。

(2) 用流程图描述程序的设计思想，可以使设计思想明确、具体、直观和更精确化。

(3) 对程序设计人员来说，有了流程图，思路清楚、层次分明，可以减少编程中出现的错误，而且也有利于程序的检查和修改。

(4) 有利于程序的交流。由于流程图直观性强，便于阅读(一般来说，看懂一个程序的流程图，要比看懂一个程序容易得多)，因此，借助于流程图，可以较快地弄清别人编制的程序。

(5) 适应性强。流程图所表示出来的是程序的计算步骤和流程，各框内都是用文字或符号表示程序的功能。它只是程序的骨架，并不是程序的一部分。因此，它适于在不同的机器上用不同的语言进行程序设计。

对于复杂问题，在编写程序前都要先画出流程图。正确地使用流程图，可以使程序的设计思想更加清晰和条理化，从而有助于开发出高质量的程序来。

例 1. 用流程图描述如下算法：

向计算机输入两个不相等的数 A、B，判断两数中哪个数大，输出大数。

本例算法的流程图如图 1-2 所示。

从图 1-2 中可以看出，本流程图从起始框开始，表示算法开始；用输入框输入两个不相等的数 A、B；向下用条件判断框判断“ $A > B$ ”这个条件是否成立，若成立，则 A 为大值，输出 A 的值，否则不成立，B 为大值，输出 B 的值；最后流程线达到终止框，表示算法结束。

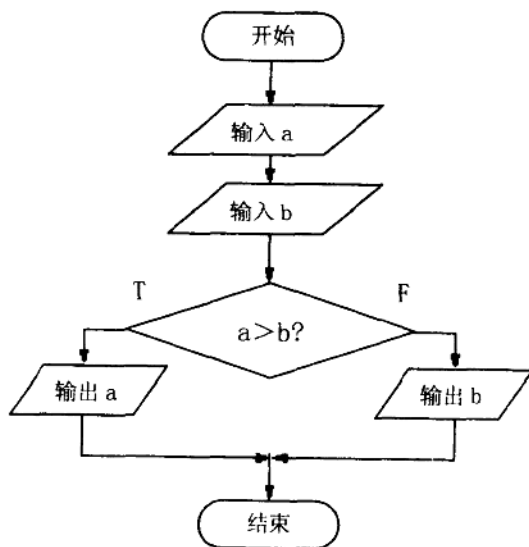


图 1-2 例 1 算法的流程图

(三) N-S 流程图

前面介绍的流程图是传统流程图。由于它有一个流程线(→)，用它可以表示 GOTO 语

句。但程序在使用 GOTO 语句时, 有很大的随意性, 使用多了也可使程序结构混乱, 可靠性降低。因此, 对于结构化程序设计, 传统流程图不太适用。1973 年, I.Nassi 和 B.Schneiderman 推出了盒图, 也称 N-S 图。按盒图规定画出的算法图都是结构化的, 所以也称为结构化盒图。N-S 流程图的主要特点是全部算法由一些基本的矩形框图顺序排列而组成的一个大矩形表示, 即不允许程序任意转移, 只能顺序执行, 从而使程序结构化。

人们经过长期的实践, 将程序的基本流程归纳为三种基本的简单结构, 即顺序、选择和循环。这三种基本结构的流程图称为结构流程图的基本图或基本结构元素。对于一个算法也可认为, 无论多么简单或多么复杂, 都可以由这三种基本结构组合构造而成。

N-S 盒图的上边框表示入口, 下边框表示出口, 它的每个基本图形都符合“单入口, 单出口”的要求, 而且和结构化程序设计的基本控制结构的约定一样。本书将用 N-S 图表示算法, 下面对流程图的基本图作进一步说明。

1. 顺序结构

顺序结构中, 各语句是按照在算法中排列的先后次序执行的。N-S 流程图的顺序结构用图 1-3b 形式表示, 它与传统流程图的图 1-3a 对应。块 a 和块 b 两个框组成一个顺序结构。它表示程序执行顺序是从块 a 到块 b。每个块可以是某个语句或语句序列。

2. 选择结构

选择结构是根据某种条件选择性地执行算法的某一部分。在选择结构中先判断条件是否成立, 如果成立则执行某个语句或语句序列, 否则执行另一个语句或语句序列。选择结构 N-S 流程图用图 1-4b 形式表示, 它与传统流程图的图 1-4a 对应。当“条件”成立时执行块 a 操作, 否则执行块 b 操作。注意, 在执行一次选择结构时, 块 a 和块 b 只有一个块被执行。

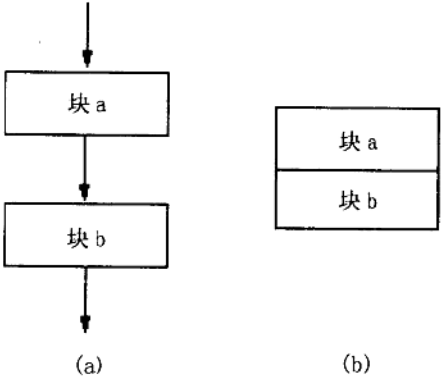


图 1-3 顺序结构流程图

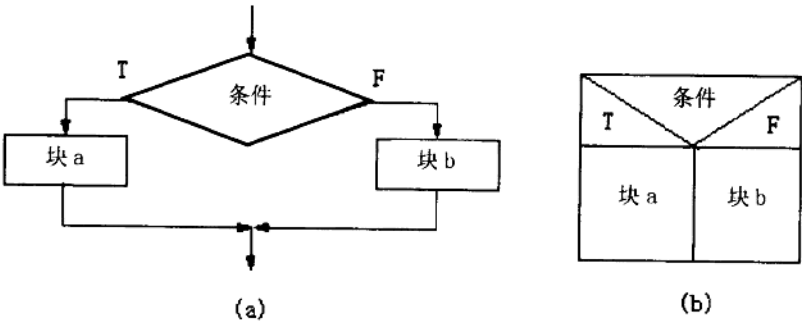


图 1-4 选择结构流程图

3. 当型循环结构

循环结构根据某种条件重复性地执行算法所规定的某一部分, 重复执行的语句或语句

序列称为循环体。在进入循环前要判断所说明的条件是否成立，只有在条件成立时才执行循环体；每次循环体执行后再判断条件是否成立，条件成立则再一次执行循环体，如此循环重复直到条件不成立为止，从而达到重复性地执行算法所规定的某一部分的目的。N-S 当型循环结构用图 1-5b 形式表示，它与传统流程图的图 1-5a 对应。当“条件”成立时，重复执行块操作，直到“条件”不成立为止。特点是：先判断，后执行块。

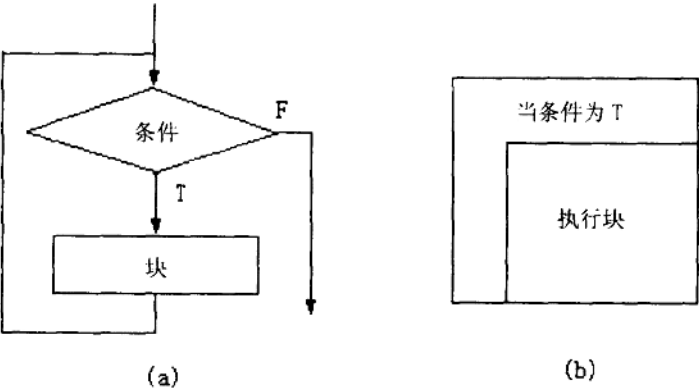


图 1-5 循环结构流程图

例 2. 用 N-S 流程图作例 1。
本例算法的 N-S 流程图如图 1-6 所示。

三、QBasic 简介

QBasic 程序语言是美国微软 (Microsoft) 公司开发的一种现代化、结构化的 Basic 程序设计语言。QBasic 程序设计语言随微软公司 MSDOS 5.0 及以上版本提供给用户。QBasic 语言与其它高级语言相比并不逊色，它既吸取了其它高级语言的诸多特点，又很好地保持了 Basic 语言易于学习和使用的特点，尤其适用于初学者，同时又具有很强的实用价值，这对许多熟悉 Basic 语言的用户提供了很大的方便。QBasic 是全国高等院校计算机基础教育研究会提出推广的语言，也是日前全国计算机等级考试的二级考试中规定的五种语言中的一种。

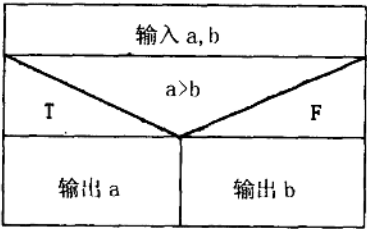


图 1-6 例 2 算法的 N-S 流程图

QBasic 和其它语言相比，具有以下特点：

- (1) 功能很强，编程效率高。它是模块化结构程序设计语言，支持递归。它适用于科学计算，又有丰富的画图功能，用它可以编写各种数据处理程序及高效率的管理程序等。
- (2) 用户界面好，使用方便。它有当代流行的菜单式图形用户界面，有会话式的编程方式和使用方便的联机帮助，易编程，易调试。
- (3) QBasic 采用先进的解释程序，使程序运行比一般 Basic 快很多。当输入一行 QBasic 语句时，系统会立即将该行上所有的关键字改为大写，并进行语法检查，如有错会立即给出出错信息，因此，特别适合初学者。

(4) QBasic 系统涉及的知识广度和深度适中, 有利于普及和提高。学会了 QBasic, 再去学习 Visual Basic 是水到渠成的事。否则, 直接学习 Visual Basic 是比较困难的。

还应指出, 在 Windows 环境下运行的 Visual Basic 与 QBasic 基本兼容, 也就是说, 用 QBasic 所写的程序一般能在 Visual Basic 环境下顺利运行。因此, 掌握了 QBasic 再过渡到 Visual Basic 是很方便的。本书将在学习 QBasic 后介绍 Visual Basic 的有关知识。

第二节 QBasic 程序的基本结构

一、一个简单的 QBasic 程序

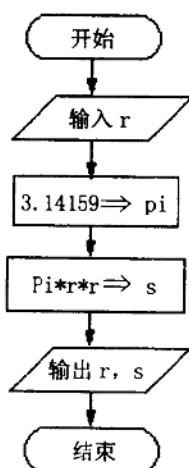
首先用一个简单的 QBasic 程序实例来说明 QBasic 程序的基本结构。

例 3. 编一个程序, 计算任一半径圆的面积。

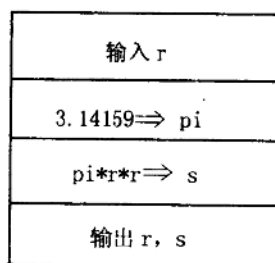
程序的流程图如图 1-7 所示。

程序如下:

```
REM 计算圆的面积
INPUT "半径 r="; r
LET pi=3.14159
LET s=pi*r*r
PRINT "R="; r, "S="; s
END
```



传统流程图



N-S 流程图

图 1-7 例 3 流程图

程序的第一行为注释语句, 通过它对程序作简短的解释说明; 第二行为输入语句, 把

圆的半径值通过键盘送给变量 r ，执行该语句时，屏幕先显示引号中的提示信息，等待用户输入半径，用户输入半径并按回车键后，程序继续执行；第三行是一个赋值语句，其作用是给变量 π 赋 π 的具体值；第四行为计算圆的面积，该句也是一个赋值语句，机器先计算出赋值号(=)右边的表达式的值，然后将计算结果赋给左边的变量 S ；第五行为输出语句，用于将圆的半径和计算结果(圆的面积)从外部设备输出；最后一行为结束语句，用于终止程序的执行。

从上面的例子中可以看出：

(1) 一个 QBasic 程序由若干个语句组成。QBasic 中一个程序行中既可以输入一个语句，也可以输入多个语句。当输入多个语句时，各语句间必须用“:”分开。比如，上例中可以将第四行和第五行写在一个程序行中。具体书写方法是：

```
LET pi=3.14159 : LET s=pi*r*r
```

(2) 一个语句包括以下两个部分：

① 语句定义符。用来规定该语句执行何种操作功能。例如上例中，第三行 $\text{LET } s=\pi*r*r$ 中，LET 就是一个“语句定义符”，表示该语句执行“赋值”功能；再如第五行中的“PRINT”表示输出功能。一个语句如果没有语句定义符是不合法的，机器也不执行该语句。

② 语句体。它在语句定义符的后面，表示语句执行的对象。如上例第三行中的 $s=\pi*r*r$ 就是一个语句体，表示 LET 执行的对象，即将圆的面积值计算后赋值给变量 s ；第四行中的“ $R=$ ”； r ，“ $S=$ ”； s 表示输出的对象，它们是“ $R=$ ”及 r 的值，“ $S=$ ”及 s 的值。需要说明的是，有些语句可以没有语句体，如结束语句 END 就没有语句体。

(3) 一个程序一般应以 END 语句结束。执行程序时，遇到 END 语句，便停止执行。

二、QBasic 程序行

QBasic 程序行的一般格式如下：

(行标识符) 语句 (: 语句)……(语句) (注释)

(一) 行标识符

行标识符是对程序进行标识的符号。QBasic 的行标识符分为行号和行标号两种。行号可以是 0 到 65529 范围内的整数。由于 0 在某些语句(如 ON ERROR)中有特殊含义，因此，最好不用 0 作行号，若用将可能出错。QBasic 中的行号不表示程序执行的顺序。行标号由字母和数字组成，最多不超过 40 个字符。它必须以字母开头，以“:”结束，如：TEST2A:、B3: 等。

行标识符是程序行中的可选项，一般不用。通常只有当语句中表示需要转向到某个程序行时，才在要转向的程序行前加行标识符。同一个源程序中行号和行标号，也可以混合使用，但一个程序行前，只能放一个行标识符。

一般来说，行号与行标号具有相同作用，可以相互代替，但是 IF~THEN 语句例外。在这个语句中，如果 THEN 后面需要行标识符，则只能用行号。例如，下面的语句是正确的：

```
IF X>=10 THEN 50
50 PRINT X
```

而下面的语句是错误的:

```
IF X>=10 THEN LL
```

```
LL: PRINT X
```

如果非要在 IF ~THEN 语句中使用行标号,则必须在 THEN 后面加上 GOTO 语句,例如,下面的这个语句就是正确的:

```
IF X>=10 THEN GOTO LL
```

(二) 语句

语句是程序行中不可缺少的主体,分为执行语句和非执行语句两种。执行语句可以完成某种特定的操作,例如:读、写、赋值、输出、转移控制等,它使程序按照流程顺序进行。非执行语句只是完成为变量分配存储单元,说明和定义变量类型以及指明源文件中被所有过程共享的变量等任务。

(三) 注释

程序行最后的注释是用来说明程序的操作和使用的。它可以含有任何字符,是非执行语句。注释只能以单引号或 REM 开头。一般比较复杂的不易理解的程序行后才加注释,简单的程序行不必加注释。在程序的开始部分也可写入若干注释语句,以说明程序的名称、功能、提示说明等。

三、QBasic 的关键字

关键字又称保留字,是 QBasic 中预先规定的,一般不用作表示其它含义的专用符号,如语句标识符、标准函数等。关键字在使用时不可更改,也不允许定义用作用户的变量、行标号等。

QBasic 中,对关键字的大小没有严格规定,即大写或小写均可。QBasic 编辑器能自动将关键字转换成大写字母。

四、复杂程序的结构

要编制一个解决复杂问题的程序,通常需要上千甚至上万条语句。不可能将这么多条语句都罗列在一个程序当中,因为这样程序结构既不清晰,调试起来也相当困难。好的方法是将问题划分成若干个功能独立的部分分而制之,这样整个程序形成了良好的层次结构,而且各部分可以分别调试,最后再进行统调,从而大大减轻了调试的工作量和困难程度。

按独立功能部分从大到小的顺序,有程序→模块→过程→程序行的包含关系。

1. 模块

模块是 QBasic 程序所包含的有某一或某些功能的独立程序块,是独立存盘并可以分别处理的源文件。任何 QBasic 语句都可出现在模块中。

模块可以包含 SUB 或 FUNCTION 过程,以及不直接属于这两类过程中的语句。不属于 SUB 或 FUNCTION 中的语句称为模块级代码。模块级代码一般包括像 DIM 和 TYPE~ENDTYPE