

jisuanji jishu jichu



高等学校“十五”规划教材



计算机技术基础 (上册)

主编 张 瑾 刘开南

中国矿业大学出版社

高等学校“十五”规划教材

计算机技术基础

(上册)

主 编 张 瑾 刘开南

中国矿业大学出版社

内容提要

本书是“高等学校‘十五’规划教材”之一。全书分上、下两册。上册为《计算机技术基础》理论教学的教程:前5章为QBasic语言教学,包括QBasic程序设计基础、顺序结构程序设计、选择结构程序设计、循环体系程序设计和过程;后6章为Visual Basic语言教学,包括Visual Basic编程基础、窗体设计与常用控件、应用程序界面设计、文件系统、图形与多媒体及数据库技术。下册分为两部分:第一部分为计算机基础实验指导;第二部分为全国计算机等级考试(二级 Visual Basic)应试指导。另外,本书附有正式出版的多媒体光盘,通过视频和动画演示,使学生进一步理解课程内容。

本书是高等学校非计算机专业的本科、专科学生学习计算机技术基础的教材,也可供各类工程技术人员和计算机爱好者学习参考。

图书在版编目(CIP)数据

计算机技术基础 / 张瑾, 刘开南主编. —徐州: 中国矿业大学出版社, 2005. 2
ISBN 7-81070-930-5/TP·47
I. 计... II. ①张... ②刘... III. 电子计算机—基本知识 IV. TP3
中国版本图书馆 CIP 数据核字(2005)第 008240 号

书 名 计算机技术基础(上、下)
主 编 张 瑾 刘开南
责任编辑 刘社育 瓮立平
出版发行 中国矿业大学出版社
(江苏省徐州市中国矿业大学内 邮编 221008)
网 址 <http://www.cumtp.com> E-mail: cumtpvip@cumtp.com
印 刷 北京京科印刷有限责任公司
经 销 新华书店
开 本 787×960 1/16 印张 42.25 字数 782 千字
版次印次 2005 年 2 月第 1 版 2005 年 2 月第 1 次印刷
印 数 1~4100 册
定 价 62.00 元(分上、下册,并附多媒体光盘 1 张)
(图书出现印装质量问题,本社负责调换)

前 言

高等学校非计算机专业学生的计算机基础教育是高等教育的一个重要的教学内容,计算机技术基础层次的教学目标是使学生掌握一门程序设计语言,具备结合专业方向应用计算机的能力。从 1999 年起,我们开始进行非计算机专业学生的计算机基础课程教学改革探索,通过教学实践总结出:学习程序设计语言,最好能从面向过程的结构化程序设计语言入手,掌握程序设计的基本思想和方法,然后在学习可视化程序设计语言时,才能很快进入较深层次的学习。为此,我们在近年的计算机技术基础教学过程中,将 QBasic 语言和 Visual Basic 语言放在同一门课程中讲授,取得了很好的效果。本书就是基于这一思想进行编写的。

本书分为上、下两册。上册为计算机技术基础的理论教学的教程:前 5 章为 QBasic 语言教学;后 6 章为 Visual Basic 语言教学。下册分两部分:第一部分为计算机技术基础实验指导;第二部分为全国计算机等级考试(二级 Visual Basic)应试辅导。另外,本书附有《计算机技术基础》多媒体教材(光盘)。通过视频和动画演示,使学生进一步理解课程内容。

从 2005 年开始,全国计算机等级考试将进行重大改革,目前已推出了新的考试大纲。本书是按照新版大纲对 Visual Basic 考试的要求编写的。教材中选编了历年 Visual Basic 二级考试试题中的部分模拟试题;多媒体教材中有 Visual Basic 的模拟上机考试界面,为学生参加全国计算机等级考试(二级 Visual Basic)提供了大量辅导资料。书中还配备了大量的习题和实验指导。本书可作为高等学校非计算机专业的本科、专科学生的教材,也可供各类计算机爱好者自学程序设计时参考。

本书由张瑾、刘开南任主编,孙改平任副主编;由耿子林教授担任主审。

本书的编者全部是多年从事计算机基础课程教学、具有丰富教学

经验的教师,其具体分工为:上册第一章、第七章由刘士杰编写;第二章、第三章、第四章、第五章由张瑾编写;第六章、第八章由孙改平编写;第九章、第十一章由刘开南编写;第十章由曹德胜编写。下册中第一章、第二章、第三章由耿子林编写;第四章、第五章、第六章由张瑾编写;第七章、第八章由刘开南编写;第九章、第十章由鞠宏军编写;第十一章、第十二章、第十三章由郭海文编写;第十四章、第十五章、第十六章、第十七章、第十八章、第十九章、第二十章和第二十一章由曹德胜编写。

本书所附的《计算机技术基础》多媒体光盘(煤炭工业音像出版社出版)由李芙玲任主编,张瑾、孙改平任副主编,刘开南担任主审。其他编制人员有曹德胜、郭海文、冀晓乐、田小琴。

由于编者水平所限,书中难免存在着缺点和错误,衷心希望读者批评指正。

联系方式:Email ylxqs375@sina.com.cn

编 者
2005年1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)
第二节 QBasic 程序的基本结构	(7)
第三节 QBasic 程序设计的基础知识	(10)
习 题	(33)
第二章 顺序结构程序设计	(36)
第一节 赋值语句(LET)	(36)
第二节 输出语句(PRINT)	(38)
第三节 键盘输入语句(INPUT)	(41)
第四节 其他语句	(42)
第五节 顺序程序综合举例	(43)
习 题	(46)
第三章 选择结构程序设计	(47)
第一节 选择结构程序设计的概念	(47)
第二节 单行结构 IF 语句	(47)
第三节 块结构 IF 语句	(49)
第四节 单条件的嵌套语句(IF...ELSEIF...END IF)	(51)
第五节 多分支选择语句(SELECT CASE)	(53)
第六节 GOTO 和 ON...GOTO 语句	(59)
习 题	(62)
第四章 循环结构程序设计	(64)
第一节 循环结构程序设计的概念	(64)
第二节 固定次数的循环语句(FOR...NEXT)	(65)
第三节 条件循环语句(WHILE...WEND)	(73)
第四节 DO...LOOP 循环语句	(76)
第五节 三种循环结构的比较	(80)
第六节 多重循环	(81)
第七节 综合习题分析	(83)

习 题	(94)
第五章 过程	(106)
第一节 概述	(106)
第二节 FUNCTION 过程	(109)
第三节 SUB 过程	(112)
第四节 过程的参数传递	(113)
第五节 过程的嵌套与递归	(116)
第六节 综合举例	(120)
习 题	(125)
第六章 Visual Basic 编程基础	(129)
第一节 Visual Basic 概述	(129)
第二节 Visual Basic 的集成开发环境	(132)
第三节 建立第一个 Visual Basic 应用程序	(140)
第四节 数据输入与输出	(153)
第五节 调试与错误处理	(160)
习 题	(166)
第七章 窗体设计与常用控件	(167)
第一节 设计窗体	(167)
第二节 Visual Basic 中常用的控件	(176)
第三节 数组与数组控件	(216)
第四节 发行 Visual Basic 应用程序	(232)
习 题	(235)
第八章 应用程序界面设计	(237)
第一节 Visual Basic 模块设计	(237)
第二节 公共对话框控件	(258)
第三节 菜单设计	(268)
第四节 多文档界面	(278)
第五节 工具栏	(286)
第六节 界面上的用户事件	(293)
习 题	(307)
第九章 文件系统	(309)
第一节 文件系统控件	(309)
第二节 文件的读写	(312)
第三节 常用的文件操作语句和函数	(325)

习 题	(327)
第十章 图形与多媒体	(328)
第一节 图形控件	(328)
第二节 图形方法	(338)
第三节 图形控件的应用	(343)
第四节 多媒体	(345)
习 题	(351)
第十一章 数据库技术	(352)
第一节 数据库的基本概念	(352)
第二节 数据库管理器	(353)
第三节 数据控件	(359)
第四节 用 Data 数据控件创建简单的数据库程序	(364)
第五节 数据库记录的添加、修改、查找和删除	(365)
第六节 ADO 数据控件	(368)
习 题	(371)
主要参考文献	(372)

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,计算机的基本运行机制是:

(1) 将需要计算机完成的任务编成 1 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

人们通过程序使计算机处理各种信息,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。编制程序就是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)步提出的算法,用一定的计算机语言编写程序。

(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,最关键的问题是提出算法,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,即不允许有二意性。

(3) 可行性。所采用的算法必须能在计算机上执行,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。计算机解决问题时,一般需要输入一些原始数据;计算机向用户报告结果时,一般要输出一些信息。因此,一个算法中一般应必须有一定的输入与输出。

(二) 流程图

流程图是一种用图形、文字和符号来比较直观、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。常用的流程图图形符号如图 1—1 所示。各流程图图形符号表示的意义如下:

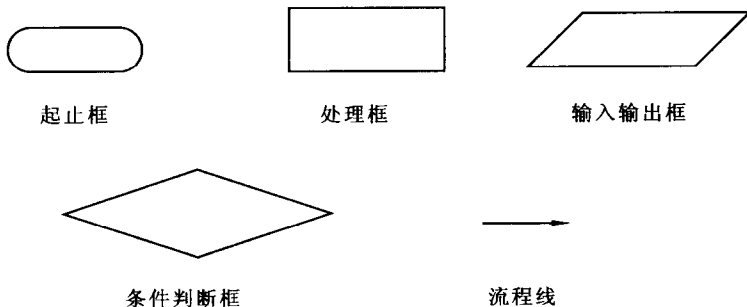


图 1—1 常用的流程图图形符号

(1) 起止框。它代表一个算法的开始与结束。

(2) 处理框。它表示算法中对问题处理的 1 个或若干个步骤,例如:赋值、计算等,这些步骤不涉及输入与输出。

(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。

(4) 条件判断框。它表示算法中需要根据条件判断程序走向的步骤。

(5) 流程线。它表示算法的每 1 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,流程图使其设计思路清楚、层次分明,不仅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(一般看懂 1 个程序的流程图,要比看懂 1 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—1] 用流程图描述如下算法:向计算机输入 2 个不相等的数 a 、 b ,判断两数中哪个大,输出大数。本算法流程图如图 1—2 所示。

分析:从图 1—2 中可以看出,本流程图以起始框表示算法开始;用输入框输入 2 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,若不成立, 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。

(三) N—S 流程图

前面介绍的流程图是传统流程图。由于传统流程图有 1 个流程线($\rightarrow$),用它可以表示 GOTO 语句(无条件转移语句)。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变得混乱,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 流程图。按照盒图规定,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,全部算法都由一些基本的矩形框图顺序排列而组成的 1 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。

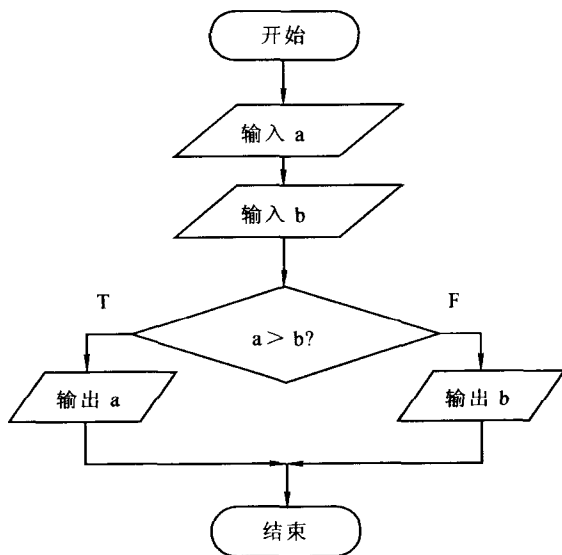


图 1—2 例 1—1 算法的流程图

人们经过长期的实践,将程序的基本流程归纳为 3 种基本的简单结构,即顺序结构、选择结构和循环结构。这 3 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,无论多么简单或多么复杂,都可以由这 3 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。下面对结构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 流程图的顺序结构如图 1—3(b)所示,其功能与图 1—3(a)传统流程图相对应。图 1—3(b)中,由块 a 和块 b 2 个框组成 1 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 流程图如图 1—4(b) 所表示,其功能与图 1—4(a) 传统流程图对应。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 1 次选择结构时,块 a 和块 b 只有 1 个块被执行。

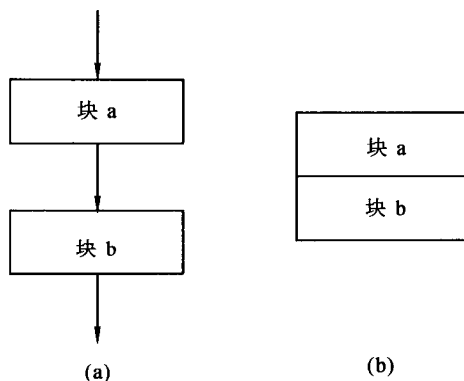


图 1—3 顺序结构流程图

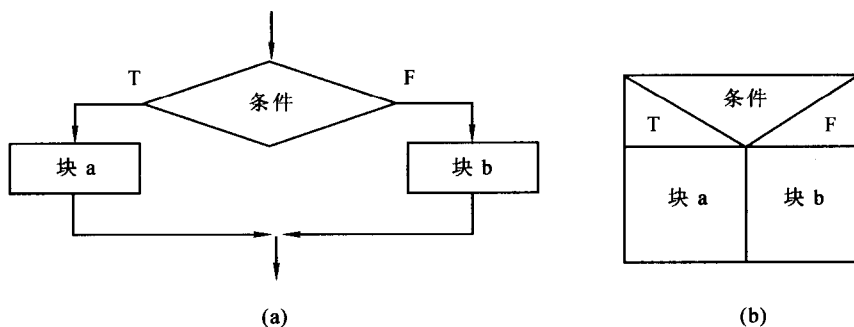


图 1—4 选择结构流程图

3. 当型循环结构

循环结构是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。当型循环结构是在程序进入循环前,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。当型循环结构的 N—S 流程图如图 1—5(b) 所示,其功能与图 1—5(a) 传统流程图对应。当“条件”成立时,重复执行块操作,直到“条件”不成立为止。其特点是:先判断,后执行块。

【例 1—2】 用 N—S 流程图做例 1—1。

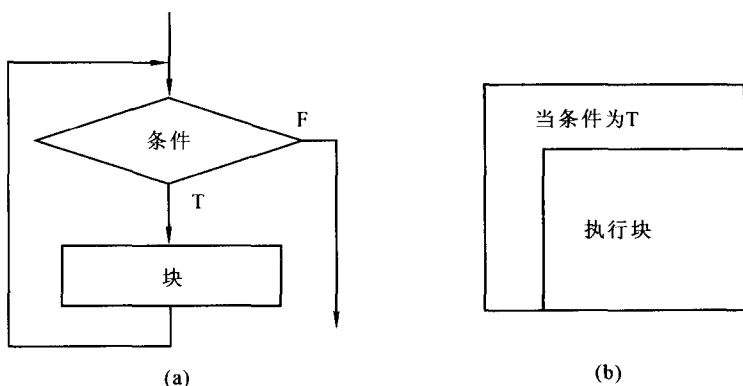


图 1—5 循环结构流程图

分析：本例算法涉及顺序结构与选择结构，其 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 语言与其他高级语言相比并不逊色，它既吸取了其他高级语言的诸多特点，又很好地保持了 Basic 语言易于学习和

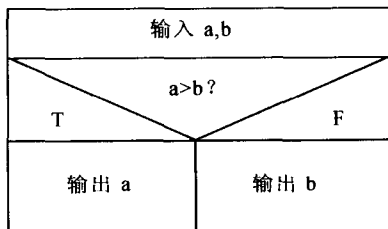


图 1—6 例 1—2 算法的 N—S 流程图

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。目前流行的 Visual Basic，就是在 QBasic 基础上发展起来的。与其他高级语言相比，QBasic 具有以下特点：

(1) 功能很强，编程效率高。它是一种模块化结构程序设计语言，支持递归。它适用于科学计算，有丰富的画图功能，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 快得多。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,特别在编写代码方面会遇到较大的困难。

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 再过渡到学习 Visual Basic 是很方便的。本书将在学习 QBasic 的基础知识后,重点学习 Visual Basic 的有关知识,便于很好地理解和掌握面向对象的、可视化的程序设计。

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 1—3] 编 1 个程序,计算任一半径圆的面积。

分析:程序的流程图如图 1—7 所示。

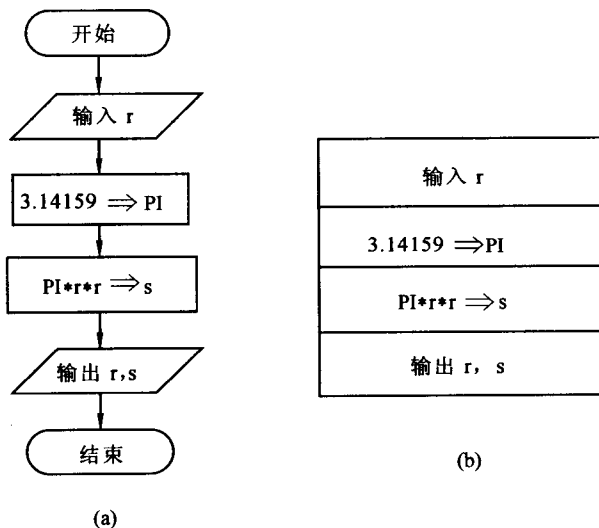


图 1—7 例 1—3 程序流程图

(a)传统流程图;(b)N—S 流程图

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
```

程序的第1行为注释语句,通过它对程序作简短的解释说明;第2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3行是1个赋值语句,其作用是给变量Pi赋 π 的具体值;第4行为计算圆的面积,该句也是1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赋值号左边的变量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) 1个QBasic程序由若干个语句组成。QBasic的1个程序行中,既可以输入1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。如例1—3中可以将第3行和第4行写在1个程序行中,具体书写方法是:

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

(2) 1个语句包括以下2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,在例1—3中,第4行LET s=PI*r*r中,LET就是1个“语句定义符”,表示该语句执行“赋值”功能;再如,第5行中的“PRINT”也是1个“语句定义符”,表示该语句执行输出功能。1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。如例1—3第4行中的s=PI*r*r就是1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5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。应当说明,有些语句可以没有语句体,如结束语句END就没有语句体。

(3) 1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 程序行的一般格式为:

(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2种。行号可以是1到40位的正数,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,且必须以字母开头,以“:”结束,例如: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。在同一个源程序中行号和行标号也可以混合使用,但 1 个程序行前,只能放 1 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
```

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:下面的语句是正确的:

```
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 2 种。执行语句可以完成某种特定的操作,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,一般不用作表示其他含义的专用符号,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 1 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 1 个程序中,因为这样的程序结构既不清晰,调试起来也相当困难。可取的方法是将问题划分成若干个功能独立的部分,分别编制,这