

内 容 简 介

《QBASIC 编程语言基础(第2版)》是2001年出版的《QBASIC 编程语言基础》的修订版。第2版在保持原书基本框架和特色的基础上对部分章节内容进行了修改和增删。

QBASIC 是一种简单实用、功能较强的解释型计算机高级语言,是我国计算机等级考试中的五种程序设计语言之一。

本书为中等职业教育计算机及应用专业国家规划教材。编写中力求体现“宽、浅、用、新”的特点,着重培养学生的逻辑思维能力和使用编程语言解决实际问题的能力。主要内容包括:QBASIC 语言的基本知识;QBASIC 语言程序设计的基本思想和方法;应用 QBASIC 语言解决一些实际问题等。在每章后都有本章小结和习题,在书后有上机实习,帮助学生掌握和巩固所学知识。

本书适用于各类中等职业学校的学生,也可作为各类职业技能培训的教材或自学用书。

本书还配有电子教学参考资料包,包括教学指南、电子教案、习题答案,详见前言。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

QBASIC 编程语言基础/尹作林主编.—2版.—北京:电子工业出版社,2005.1
中等职业教育国家规划教材(计算机及应用专业)
ISBN 7-121-00584-0

I. Q… II. 尹… III. BASIC 语言—程序设计—专业学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 132208 号

责任编辑:李 玮 特约编辑:刘 嘉

印 刷:

出版发行:电子工业出版社

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:12.5 字数:320 千字

印 次:2005 年 1 月第 1 次印刷

印 数:8 000 册 定价:15.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010)68279077。质量投诉请发邮件至 zlts@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

第1章 编程语言概述



1.1 编程语言简介

人和人交流思想和信息要用到自然语言；人和计算机交流信息要用到编程语言。为了让计算机执行解题算法，人们需要将解题的思路和步骤利用编程语言描述成计算机能够理解和执行的语言。

1.1.1 编程语言的发展及特点

随着计算机的出现及其日新月异的发展，编程语言的产生和发展也经历了由简单到复杂、从低级到高级的发展过程。

在计算机刚出现时，人们使用的是机器语言，通过使用二进制代码表示的计算机指令系统来编写程序。这种程序虽然计算机能够理解并直接执行，但对初学者来说不够直观，难学、难记也易懂，出现错误也难检查和修改，而且不便于交流。因此，当时机器语言的应用有很大的局限性。

后来，人们用一些符号来代替每一个具体的指令，即符号语言，也称汇编语言。其实，计算机并不认识汇编语言，它要求有一个“翻译”，即由汇编程序将汇编语言翻译成机器语言，然后让计算机执行。机器语言和汇编语言都与具体的计算机有关，因而被称为面向机器的语言。

人们希望有一种与具体的机器无关的语言，最好是接近于数学公式的描述。这就出现了各种编程语言，也称为高级语言。高级语言编写的程序易读、易修改，但要让计算机执行，必须通过编译程序翻译成机器语言。例如 PASCAL 语言、FORTRAN 语言和 C 语言都必须由编译程序翻译后才能执行，而 BASIC 语言通常靠解释程序边解释边执行。

计算机通过程序指令来自动执行，而程序要靠程序设计人员利用编程语言来编写和调试。程序设计人员根据实际任务提出问题，然后确定计算方法，构造数学模型，画出流程图，选择合适的编程语言，编写程序并上机反复调试，直到满足实际需要为止。这就是用编程语言进行程序设计的过程。

1.1.2 QBASIC 语言的特点

BASIC(Beginner's All-purpose Symbolic Instruction Code)的中文意思是初学者通用符号指令代码。

当前，流行的 BASIC 语言有 BASICA, True BASIC, Turbo BASIC, QBASIC(Quick BASIC)等。BASICA 是解释型语言，简单易学，执行速度慢，一般用于简单科学计算和简单事务管理等小型题目。而 True BASIC, Turbo BASIC, QBASIC 均是编译型语言的集成开发环境，执行速度快，具有结构化编程功能，一般用于编写大型程序。QBASIC 的总体设计比 Turbo BASIC 更完善一些。



QBASIC 是 BASIC 语言发展到近阶段的产物,它的主要特点如下。

- ① QBASIC 采用先进的解释程序,在输入一行语句时,立即进行语法检查,若有错误会立刻给出错误信息,程序运行比一般 BASIC 快得多。
- ② QBASIC 支持鼠标操作并具有良好的用户界面和命令菜单。
- ③ QBASIC 允许用户定义多行的、单独的函数模块,函数中的变量是局部变量。

1.2 QBASIC 语言的简单程序结构

通过下面两个例子来说明 QBASIC 语言的程序结构。

【例 1.1】 已知圆的半径 r ,求这个圆的面积 S 。

```
REM area=?
INPUT r
LET p=3.1415926
LET area=p*r^2
PRINT r;area
END
```

【例 1.2】 计算 $S=1!+2!+3!+\dots+10!$ 。

```
REM this is a program to calculate the sum
DECLARE SUB fact(n,p)
CLS
S=0
FOR I=1 to 10
  Call fact(I,f)
  S=s+f
NEXT I
PRINT "s=";s
End

SUB fact(n,p)
P=1
FOR i=1 to n
  P=p*I
NEXT I
ENDSUB
```

在例 1.1 中, QBASIC 源程序由主模块组成,没有子模块。例 1.2 中的 QBASIC 源程序由两个模块组成,前面为主程序模块,后面为 SUB 子程序模块。从这两个例子中我们可以看出 QBASIC 源程序的基本结构是:

源程序 { 主程序模块(1 个)
 { 子程序模块(0 个、1 个或多个) { FUNCTION 过程
 SUB 过程

关于 QBASIC 程序有如下说明。请访问: www.ertongbook.com



① 一个 QBASIC 程序由一系列语句行(程序行)组成,一个语句行可以有一个或多个语句。当一个语句行有多个语句时,语句之间用冒号(:)隔开。

② 语句行可以有行号,也可以没有行号。行号的目的是作为转移的目标,而不是代表执行顺序。

③ 解释语句也是可选的,它的功能是对语句行(程序行)或程序段的功能、意义进行解释,以增加程序的可读性。

④ 一个语句一般包括两部分:语句定义符和语句体。语句定义符是用来标识该语句的性质,而语句体是用来表示该语句执行的对象。例如在例 1.1 中的“LET p=3.1415926”语句,“LET”是语句定义符,也可省略。“p=3.1415926”是语句体。个别语句没有语句体,例如“END”和“CLS”。

⑤“END”是结束语句,一个程序可以有多个 END 语句。如果程序中只有一个 END 语句,应放在整个程序的最后。

⑥ 一个 QBASIC 源程序有且必须有一个主程序模块。

⑦ 一个源程序可以有若干个子程序模块,在主程序模块中可以调用一个或多个子程序模块。子程序之间可以相互调用,但子程序不能调用主程序。

1.3 QBASIC 程序的上机调试和运行

在 MS-DOS 5.0 及以上版本中都包含 QBASIC 系统,它包含两个文件:

QBASIC.EXE QBASIC 系统的可执行文件(不可缺少)

QBASIC.HLP QBASIC 系统的帮助文件

上述文件默认在 DOS 子目录中。若机器中无上述两个文件,可从其他机器上拷贝到软盘或硬盘上。若用户想使用汉字或鼠标,应在进入 QBASIC 环境之前运行汉字操作系统 UC-DOS、汉字输入法、鼠标驱动程序。

本节简单介绍 QBASIC 语言的上机调试和运行方法。

1.3.1 QBASIC 的进入

首先进入 QBASIC 两个文件所在的目录,在 DOS 提示符下输入:

```
QBASIC ↵
```

1.3.2 QBASIC 的工作窗口

在进入 QBASIC 后按 Esc 键,就可进入 QBASIC 的工作窗口,如图 1.1 所示。

可以看到,QBASIC 提供了两个工作窗口。

上面一个窗口叫程序窗口,用来输入和编辑程序。窗口中的“Untitled”是尚未命名的意思,即新程序还没有命名。若调入一个有文件名的程序,则此处为该文件名。

下面一个窗口叫直接窗口(Immediate),在将程序语句输入到程序窗口以前,可以在此窗口进行测试。当在此窗口输入一个语句并按回车键后,此语句被立即执行并输出结果。

这两个窗口中只能有一个窗口在工作,可以使用 F6 键或鼠标进行切换。当前工作的窗口称为活动窗口,活动窗口的标题是高亮度显示的。

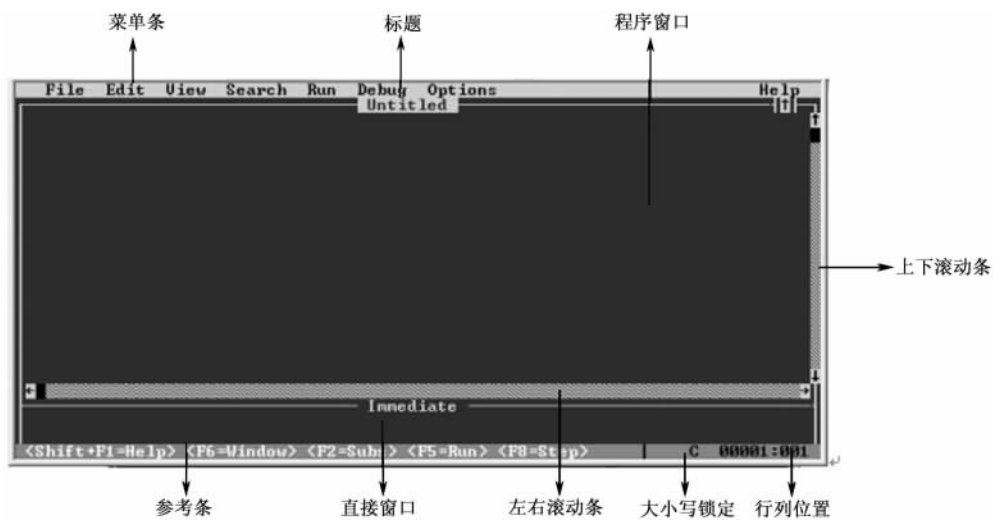


图 1.1 QBASIC 的工作窗口

1.3.3 菜单功能

程序窗口的上端有八个菜单,其功能见表 1.1。

表 1.1 菜单功能

菜 单	功 能
File	文件菜单,处理与文件有关的操作
Edit	编辑菜单,处理与程序编辑有关的操作
View	查看菜单,用于查看程序的有关部分
Search	查找菜单,用于查找所需要的内容
Run	运行菜单,用于运行程序
Debug	调试菜单,用于调试程序
Option	选择菜单,用于屏幕的前景、背景颜色的设置等
Help	帮助菜单,用于选择帮助信息

我们可以使用键盘或鼠标来选择菜单和执行命令。

① 使用键盘时,首先按 Alt 键激活菜单,然后用方向键←↑→↓选择所需要的命令,在选定后按回车键执行所选命令。

② 使用鼠标时,将鼠标移到所选菜单上按左键,弹出下拉菜单,再移动鼠标,在选择所需要的命令后按左键执行该命令。

1.3.4 QBASIC 程序的输入、编辑和运行

1. 输入 QBASIC 程序

使程序窗口成为活动窗口,输入以下程序:

```
REM area=?
INPUT r
LET p=3.1415926
```



```
LET area=p*r^2
PRINT r;area
END
```



注意

在输入程序时,如果保留字(即语句定义符)是小写,在这条语句输完按回车键后,屏幕上将自动转换为大写,如图 1.2 所示。

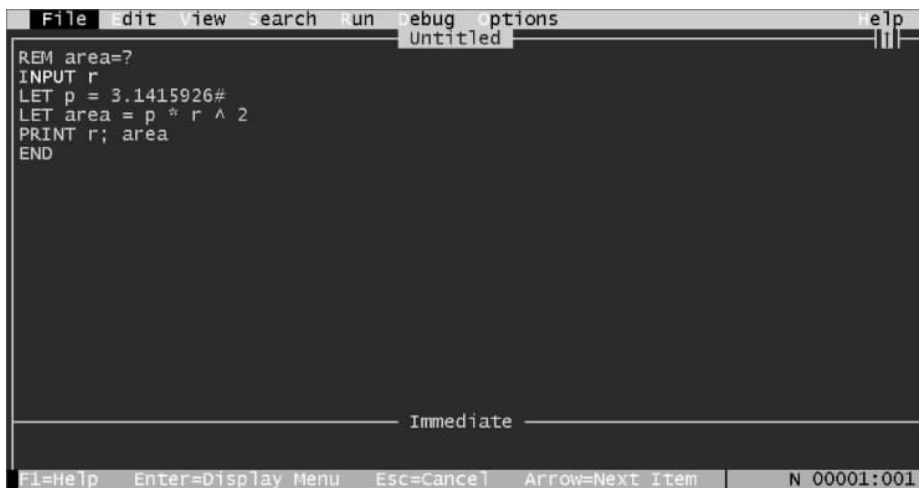


图 1.2 输入程序

2. 装入外存中已有的 QBASIC 程序

打开“File”下拉菜单,选择“Open”命令,弹出一个对话框,如图 1.3 所示。可用以下两种方法打开所需要的文件:

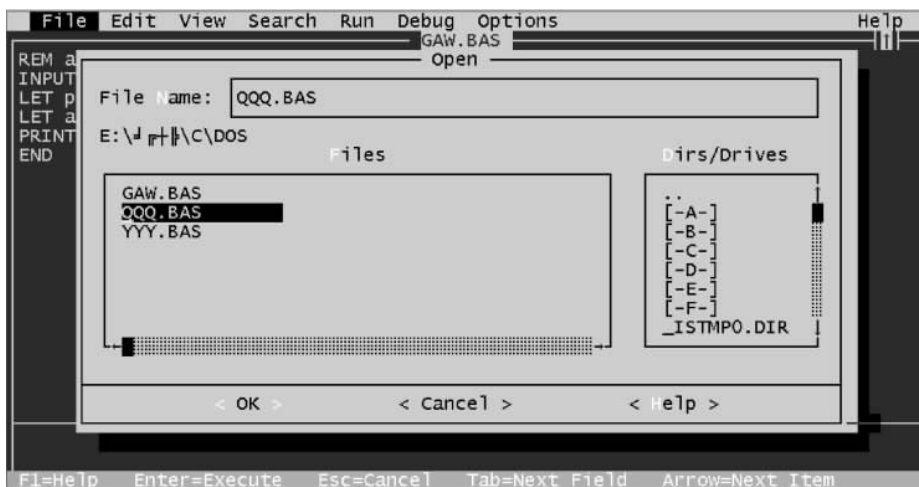


图 1.3 “Open”对话框

- ① 在“File Name”输入框中输入文件名(默认扩展名为.BAS,必要时还需加上盘符和路径)。
- ② 用 Tab 键或鼠标切换磁盘及目录,用光标键在框内移动,直到欲打开的文件被选中为止。确定文件后接下回车键或单击“OK”按钮,即可装入所选择的文件。



3. 编辑 QBASIC 程序

① 使用←↑→↓方向键将闪动的光标移动到需要插入、删除或修改的字符位置,按下 Del 键、Delete 键或 Backspace 键删除一个字符,使用 Ins 键或 Insert 键来插入一个字符。按下 Ins 键可以在插入状态和修改状态之间进行切换。

② 按下 Ctrl+Y 组合键,可以删除光标所在行。

③ 若要插入一行,先将光标移到插入处的行首,按回车键插入一个空行,再输入所需内容。

4. 运行 QBASIC 程序

运行 QBASIC 程序,可以使用以下三种方法之一。

① 选择“Run”菜单中的“Start”命令,如图 1.4 所示。



图 1.4 运行程序

② 直接按下功能键 F5。

③ 使直接窗口成为活动窗口,键入命令“run”,回车并运行程序。

5. 保存 QBASIC 程序

选择“File”菜单中的“Save”命令,可在原保存路径以原文件名保存当前 QBASIC 程序;选择“Save as”命令,可改变文件名和保存路径来保存当前 QBASIC 程序,如图 1.5 所示。



图 1.5 保存文件



6. 退出 QBASIC 程序

选择“File”菜单中的“Exit”命令,可退出 QBASIC。如果没有保存当前 QBASIC 程序,将弹出一个对话框,提醒保存当前 QBASIC 程序,如图 1.6 所示。

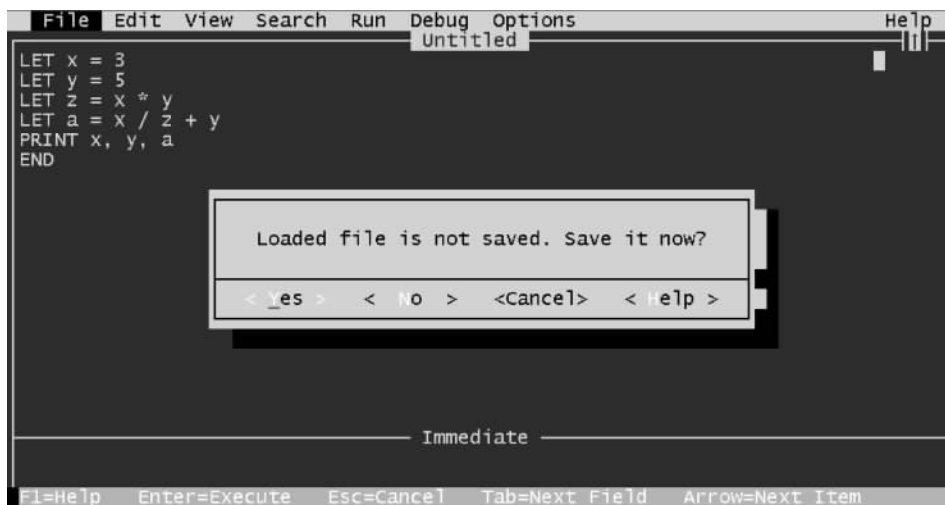


图 1.6 退出 QBASIC



本章小结

通过对本章的学习,主要了解编程语言及 QBASIC 语言的发展和特点,初步掌握 QBASIC 语言简单程序的结构,掌握 QBASIC 的进入方法、工作窗口和菜单功能操作过程,学会简单的 QBASIC 程序的输入、编辑、保存和运行。



习题 1

1. QBASIC 语言有哪些主要特点?
2. 上机练习
 - (1) 进入、退出 QBASIC;
 - (2) 按照书中例题输入一段小程序,学会输入和保存程序;
 - (3) 从计算机硬盘中装入一个小程序,学会装入和编辑程序;
 - (4) 运行输入或装入的程序。
3. QBASIC 语言系统包含哪两个文件? 如果在你的 DOS 系统中没有上述两个文件,应如何处理?
4. 扩展题: 如果想在 QBASIC 语言环境中输入汉字,应如何处理?

第2章 数据类型、运算符和表达式



2.1 QBASIC 使用的各种数据类型

任何一种编程语言都是为了编写程序而设计的。程序是具有特定功能的用某种编程语言表示的语句序列。程序是根据算法编写的,同时程序又是算法的一种表示形式和实现算法的手段。程序的功能概括地说就是对一组数据进行描述和操作(加工或处理),得到一组结果数据并输出。对数据的描述是由数据类型完成的,而对数据的操作是由语句完成的。本节主要介绍数据类型。

数据类型主要解决这样几个问题:数据占用内存的字节数,数据的表示范围,数据能进行的运算及数据的输出形式。此外,对于实型数据还存在有效数位(精度)问题。

QBASIC 的基本数据类型见表 2.1。

表 2.1 QBASIC 的基本数据类型

类 型	类 型 符	类 型 名	占内存字节数	表示范围
整型	%	INTEGER	2	-32768~32767
长整型	LONG	4	-2147483648~2147483647	
单精度型	!	SINGLE	4	-M1~-N1, N1~M1
双精度型	#	DOUBLE	8	-M2~-N2, N2~M2
字符串型	\$	STRING	实际字符个数	0~32767 个字符
定长字符串型		STRING * N	N	0~N 个字符(0≤N≤32767)

在表 2.1 中, $M1 = 3.402823E38$, $N1 = 1.401298E-45$, $M2 = 1.797693134862315D308$, $N2 = 4.940656458412465D-324$ 。

由此可见,单精度型的有效数位为 7 位,双精度型的有效数位为 16 位。当单精度数的绝对值小于 $N1$ 或双精度数的绝对值小于 $N2$,称为下溢。计算机将下溢的数据作为 0 来处理。当单精度数的绝对值大于 $M1$ 或双精度数的绝对值大于 $M2$ 时,称为上溢。计算机将给出溢出的信息。



说明

基本数据类型是 QBASIC 预先定义好的,用户可以使用它们但不能改动它们。

2.2 常量、变量和标准函数

2.2.1 常量

常量是指在程序执行过程中其值不能改变的量。在 QBASIC 中,常量分为数值常量和字符串常量,还有符号常量的形式。



1. 数值常量

(1) 整型常量(整数)

整数是由1~5个数字组成的序列,其范围在整型的表示范围内,数字前可以有符号位(符号“+”可不写或以空格代替)。整数的后面可以带类型符号“%”,但也不能超出表示范围。例如-1,0,123,32767%都是合法的整数,而32768%就不是合法的整数。

(2) 长整型常量(长整数)

长整数是由1~10个数字组成的序列,其范围在长整型的表示范围内,数字前可以有符号位。长整数后面可以带类型符“&.”,但也不能超出表示范围。为了与整数区别,对于在整数范围内的长整数规定数后必须带类型符“&.”。

例如32768,2147483647,-1&.都是合法的长整数;而-32768,-1,2147483648都不是合法的长整数。

(3) 单精度实数

单精度实数有小数形式和指数形式两种。

小数形式是由1~7个数字组成的序列,数字前面或数字之间必须有一个小数点,最前面可以有符号位。用小数形式表示的单精度实数,后面可以加类型符“!”。例如1.234567,123.4567!都是合法的单精度实数。

指数形式是由数字部分和指数部分组成的序列。其中,数字部分用小数形式表示;指数部分用E后跟两位带符号的整数(符号“+”可不写)表示10的若干次幂。用指数形式表示的单精度实数不得超出单精度的表示范围。例如1.234567E7,2.345678E-10都是合法的单精度实数;而1.234567E7!就不是合法的单精度实数(用指数形式表示的单精度实数后面不能加类型符“!”)。

(4) 双精度实数

双精度实数也有小数形式和指数形式两种。

小数形式是由1~16个数字组成的序列,数字前面或数字之间必须有一个小数点,最前面可以有符号位。用小数形式表示的双精度实数后面可以加类型符“#”;对于数字位不超过7位的用小数形式表示的双精度数在数后必须加类型符“#”,以便和单精度数区别。例如1.2345678,1.234567#都是合法的双精度实数;而1.234567就不是合法的双精度实数。

指数形式是由数字部分和指数部分组成的序列,其中数字部分用小数形式表示,指数部分用D后跟两三位带符号的整数(符号“+”可不写)表示10的若干次幂。用指数形式表示的双精度实数不得超出双精度的表示范围。例如1.2345678D12,1.2345D-100都是合法的双精度实数;而1.2345D-100#就不是合法的双精度实数(有了D,就不能有#)。

2. 字符串常量

由一对双引号括起来的零个或多个合法字符(通常为ASCII字符)称为字符串常量。例如“CHINA”,“QBASIC”,“DOS 6.0”等是合法的字符串常量。



注意

“”为空串,不含任何字符,“ ”为含一个空格字符的字符串常量,两者是不同的。在书写字符串常量时,两端的双引号不可省略,而在输出时两端的双引号并不输出,例如PRINT“QBASIC”的输出结果为QBASIC。



3. 符号常量

符号常量就是用一个符号代替一个常量,用 CONST 语句来定义符号常量。例如:

```
CONST PI= 3.141593
CONST PRICE= 8.2
CONST QB="QBASIC"
```

使用符号常量有如下三个优点。

① 程序的可读性好。符号常量用的符号应该“见名知义”,使人在阅读程序时容易了解该常量是什么含义。

② 便于程序的修改。如果程序多处用到某个常量,若不用符号常量,修改时容易漏改,而用符号常量便可以做到“一改全改”。

③ 便于程序的书写。如果程序多处用到某个位数较多的常量时,书写麻烦还容易出错,而使用符号常量只在定义时书写一次就行了。例如 CONST PI= 3.141592653589793。

2.2.2 变量

在程序执行过程中,其值可以改变的量称为变量。

1. 变量名和变量的值

变量名以字母打头,后面的字符可以是字母、数字和小数点,长度为 1~40 个字符,不区分大小写。QBASIC 的保留字(例如语句定义符和函数名等)不能作为变量名。必要时,变量的类型符可以写在变量名的后面。

例如 A,B,M,N,A1,B1,YEAR,LEAP 等都是合法的变量名;而 2ND,CHANG-LI,WANG YING,SIN 不是合法的变量名。

变量名应尽量“见名知义”,建议变量名最好不使用小数点,长度尽量不超过 8 个字符。

每一个变量都在内存中占有一定的存储单元。变量名实际上就是变量的符号地址,而变量的值就是以该变量名为符号地址的存储单元的内容。例如 LET A= 1.2345 表示将实数 1.2345 存入单精度型变量 A 所标识的内存单元(4 个字节)中。

2. 变量的类型

每一个变量都是用来存储数据的,依数据的类型不同,变量也相应地分为不同的类型。就基本类型来分,包括整型变量、长整型变量、单精度型变量、双精度型变量、字符串型变量、定长字符串型变量。

说明或定义变量类型的方法有两种:

(1) 在变量名的后面加类型符

例如 I%,N&,G!,D#,S\$ 分别是整型、长整型、单精度型、双精度型和字符串型变量。其中,规定单精度型变量的类型符可省略不写,即 G,A,B 等都是单精度型变量,这种情况用得最多。另外,定长字符串型变量不能用加类型符的方法定义,只能用 DIM 语句定义。

(2) 用 DIM 语句定义变量的类型

DIM 语句的格式为

```
DIM 变量名 1 AS 类型名 1 [,变量名 2 AS 类型名 2.....]
```

例如

```
DIM I AS INTEGER, J AS INTEGER, N AS LONG
DIM A AS SINGLE, D AS DOUBLE, S AS STRING, FS AS STRING * 20
```



在用 DIM 语句定义变量的类型时,变量名后面不能加类型符;而在引用 DIM 语句定义过的变量时,变量名后面可以加类型符。例如,上面 I 已被 DIM 语句定义为整型,在使用时 I 和 I%是一个意思。又如,NUM 没有被 DIM 语句定义为整型变量,则 NUM 和 NUM%是两个不同类型的变量,其中 NUM 为单精度型变量(建议尽量不要这样使用变量)。



说明

① 单(双)精度常数在表示形式上有小数形式与指数形式之分,但单(双)精度型变量没有上述两种形式之分,在内存中一律以规格化的浮点形式存储。

② 在用 DIM 语句定义定长字符串型变量时,应事先估计出实际字符串的最大长度。例如姓名的拼音一般不超过 20 个字符。在使用时,如果赋给定长字符串型变量的字符个数多于指定的长度,则右端多余部分被截去而不做溢出处理。

③ BASIC 以简单易学著称,变量可以不先说明而直接使用。用得最多的是单精度型变量(QBASIC 中变量的默认类型为单精度实型,故可省写类型符“!”),只在需要的场合才使用(长)整型、双精度型及字符串型变量。

2.2.3 标准函数

QBASIC 提供的标准函数包括数值函数、字符串函数和系统函数等。本节只介绍数值函数。表 2.2 中列出了一些最常用的数值函数(或数学函数),有关这些函数的使用方法见第 6.2 节。

表 2.2 常用数值函数

序号	函数	功能
1	SGN(X)	符号函数 $\text{sgn } x = -1(x < 0), 0(x = 0), 1(x > 0)$
2	ABS(X)	绝对值函数 $ x $
3	INT(X)	取整函数 $\text{int } x = \text{不大于 } x \text{ 的最大整数}$
4	FIX(X)	截尾函数,即截去小数部分(不考虑四舍五入),取整数部分
5	SQR(X)	平方根函数 \sqrt{x}
6	EXP(X)	以 e 为底的指数函数 e^x
7	LOG(X)	以 e 为底的对数函数 $\ln x$
8	SIN(X)	正弦函数 $\sin x, x$ 的单位为弧度
9	COS(X)	余弦函数 $\cos x, x$ 的单位为弧度
10	TAN(X)	正切函数 $\tan x, x$ 的单位为弧度
11	ATN(X)	反正切函数 $\arctan x, x$ 的单位为弧度
12	RND[(X)]	随机函数,产生一个在(0,1)之间的随机数



说明

① 函数的自变量 X 是个形式参数,在调用函数时要用实际参数代替形式参数。实际参数可以是一个常量、变量或表达式,例如 INT(SQR(3))等。

② 调用一个函数不能单独作为一个语句,只是作为一个表达式或表达式的一部分。例如



$\text{SIN}(3.141593/3), A + \text{INT}((B - A + 1) * \text{RND})$ 等。

2.3 运算符和表达式

QBASIC 的运算符分为算术运算符、关系运算符和逻辑运算符,相应地有算术表达式、关系表达式和逻辑表达式。

2.3.1 算术运算符和算术表达式

1. 算术运算符

QBASIC 的算术运算符(按优先级从高到低)如下:

\wedge (乘方) \rightarrow $-$ (取相反数) \rightarrow $*$ (乘), $/$ (除) \rightarrow \backslash (整除) \rightarrow MOD (求余) \rightarrow $+$ (加), $-$ (减)

对于 $*$, $/$, \backslash 运算,要注意除数不得为零。对运算符 \wedge ,需说明如下。

① $A \wedge 0 = 1$,其中 $0 \wedge 0 = 1$ 是对数学的补充定义。

② $0 \wedge B = 0 (B > 0)$ 。

③ 当 B 为正整数时, $A \wedge B = (A * A * \dots * A)$ (B 个 A); B 为负整数时,要求 $A \neq 0$; $A \wedge B = 1 / (A * A * \dots * A)$ ($|B|$ 个 A)。

④ 当 B 为非零实数时,要求 $A \geq 0$; $A \wedge B = \text{EXP}(B * \text{LOG}(A))$ 。

对于 \backslash 和 MOD ,要求余数的符号和被除数的符号相同,例如:

```
5\2 = 2, 5 MOD 2 = 1
-5\2 = -2, -5 MOD 2 = -1
5\ -2 = -2, 5 MOD -2 = 1
-5\ -2 = 2, -5 MOD -2 = -1
```

MOD 称为求余,而不称为求模。因为求模要求除数为正整数而余数为非负数。另外,当两个实数进行 \backslash 或 MOD 运算时,则先按四舍五入变成整数,再整除或求余。

如果还有函数参与运算,则函数优先于乘方;另外,可以用括号改变运算顺序,即括号最优先。对于同一优先级的运算符(例如 $*$, $/$ 或 $+$, $-$),按从左到右的顺序进行。

2. 算术表达式

表达式的基本概念是用运算符(可以使用括号)连接运算量形成的式子。用算术运算符(可以使用括号)连接数值运算量(包括常量、变量、函数等)形成的式子称为算术表达式。例如:

```
-8*(1/3)/(9\4) + INT(SQR(3) * SIN(ATN(1))) - 7 MOD 3
```

是一个算术表达式。按照运算符的优先级进行运算可以求得该表达式的值为 -1 。

2.3.2 关系运算符和关系表达式

1. 关系运算符

QBASIC 提供的关系运算符(又称关系比较符)有以下六种:

$<$ (小于), $>$ (大于), \leq 或 $=<$ (小于等于), \geq 或 $=>$ (大于等于), $=$ (等于), $<>$ 或 $><$ (不等于)

关系运算符的优先级是相同的。

2. 关系表达式

关系表达式是指由一个关系运算符连接两个算术表达式(或两个字符串表达式)形成的式



子。例如： $A+B>C$ ， $I\leq B$ ， $B*B-4*A*C\geq 0$ ， $A<>0$ ， $CNT \text{ MOD } 10=0$ ，“ABC”<“ACB”，“ABC”<>“ABC”都是合法的关系表达式。

所谓字符串表达式是指用字符串连接运算符“+”连接字符串量(字符串型的常量、变量、函数和数组元素等)形成的式子，例如“STR”，“A”+“BOOK”，“I”+“LOVE”+“CHINA”。关于字符串的比较详见第7章。

关系表达式的值是一个逻辑值，即“真”或“假”。例如 $1<2$ ， $1\leq 2$ ， $1<>2$ 的值均为“真”；而 $1>2$ ， $1\geq 2$ ， $1=2$ 的值均为“假”。

在QBASIC中，没有专门设置逻辑型常量和逻辑型变量，而是用数值来代表逻辑值。规定以“0”代表“假”，以“-1”代表“真”。例如 $1<2$ 的值为-1，而 $1>2$ 的值为0。

2.3.3 逻辑运算符和逻辑表达式

1. 逻辑运算符

QBASIC 提供三个逻辑运算符，按优先级从高到低书写如下：

NOT(逻辑非)→AND(逻辑与)→OR(逻辑或)

其中，NOT 为一元运算，其真值表见表 2.3；AND 和 OR 都是二元运算，其真值表见表 2.4。

表 2.3 逻辑非真值表

A	NOT A
假	真
真	假

表 2.4 逻辑与、逻辑或真值表

A	B	A AND B	A OR B
假	假	假	假
假	真	假	真
真	假	假	真
真	真	真	真

2. 逻辑表达式

一个逻辑表达式是用逻辑运算符连接关系表达式(或其他逻辑量)形成的式子。例如 NOT ($1>2$)， $(N\geq A) \text{ AND } (N\leq B)$ ， $(X<A) \text{ OR } (X>B)$ 等都是合法的逻辑表达式。

关系表达式是逻辑表达式的特例，例如 $A<>0$ 是一个关系表达式，也可以说是一个逻辑表达式。这和“1，A，SQR(2)都是算术表达式的特例”的说法是类似的。但应注意，关系表达式没有特例。例如 $A+B>C$ 是一个关系表达式，不能说 $A+B$ 或 C 是关系表达式的特例。又如“ABC”<>“ACB”是一个关系表达式，不能说“ABC”或“ACB”是关系表达式的特例。它们都是字符串表达式的特例。

2.3.4 不同类型数据的混合运算

一般地说，同类型的数据进行运算，结果的类型相同。但关系运算不同，两个数值量或两个字符串经过关系运算后，结果是一个逻辑值。

1. 不同类型数值量的算术运算

两个不同类型的数值量进行算术运算，结果的类型是两者中占字节数较多的那种类型。例如整型和长整型算术运算，结果为长整型；整型与单(双)精度型算术运算，结果为单(双)精度型；长整型与单(双)精度型算术运算，结果为双精度型；单精度型与双精度型算术运算，结果为双精度型。

这里要特别注意，长整型与单精度型(都占4个字节)算术运算，结果为双精度型(8个字节)。这是因为尽管长整型的表示范围在单精度型的表示范围之内，但单精度型的有效数位却比长整型的有效数位少。为了兼顾表示范围和有效数位，所以结果的类型为双精度型。



2. 不同类型数值量的逻辑运算

逻辑运算的对象通常是关系表达式,前面已说明“假”用“0”表示,“真”用“-1”表示。事实上,QBASIC 的逻辑运算是对整数按位进行的,因此,数值量也能进行逻辑运算,此时结果的类型是占字节数较少的那种类型。具体来说,整型和长整型逻辑运算,结果为整型(长整型的高位无效)。单(双)精度型四舍五入后如果在整型(长整型)的表示范围内,结果为整型(长整型),否则不能进行逻辑运算(字符串型也不能进行逻辑运算)。一般来讲,不宜使用难于理解的逻辑运算。例如

```
NOT 0 = -1, NOT 1.6 = -3, NOT 32767 = -32768
NOT 2.1474836466D9 = -2147483648
1 AND 2 = 0, 5 OR 6 = 7, 3.27666E4 AND 7 = 7
```

3. 不同类型运算符的优先次序

一个逻辑表达式内可能包括算术运算符、关系运算符和逻辑运算符,这三类运算符的优先级从高到低为:

算术运算符→关系运算符→逻辑运算符

例如,有一个逻辑表达式为

```
Y MOD 4 = 0 AND Y MOD 100 <> 0 OR Y MOD 400 = 0
```

这是一个计算闰年的表达式。当 $Y=2000$ 时,上述表达式的值为 -1 ; $Y=2001, 2002, 2003$ 时,表达式的值均为 0 ; 当 $Y=2004$ 时,表达式的值又为 -1 。

为了便于阅读程序,可以将逻辑运算的对象关系表达式的两侧加括号,例如上面的逻辑表达式可书写如下:

```
(Y MOD 4 = 0) AND (Y MOD 100 <> 0) OR (Y MOD 400 = 0)
```

这样写不但醒目,而且也是一种好的习惯。



本章小结

本章首先介绍了 QBASIC 的各种基本数据类型。这些数据类型在编程时经常用到。另外,需要掌握的是常量和变量的定义,要学会使用各种运算符及表达式。



习题 2

- QBASIC 的基本数据类型有哪几种?
- 对于 QBASIC 语言,下列常数哪些是合法的? 它们各属于什么类型?

(1) 12345	(2) -34789%
(3) 1234567890	(4) -12345678901&
(5) .1234567	(6) -1.23E100
(7) 2.718281828459046	(8) D-100
(9) "STRING"	(10) 'ABC'
- 下列符号名中哪些可以作为 QBASIC 的变量名?

(1) .ABC	(2) A2
----------	--------



- | | |
|---------------|---------------|
| (3) 3A | (4) A. 4 |
| (5) A_5 | (6) CNT1 |
| (7) CHEN YING | (8) WANG, LEI |
| (9) ATN | (10) AVERAGE |

4. 将下列数学表达式写成 QBASIC 的算术表达式。

- | | |
|------------------------------------|---|
| (1) $ x $ | (2) $-\frac{b}{a}$ |
| (3) $\sqrt{a^2+b^2}$ | (4) $\sqrt{s(s-a)(s-b)(s-c)}$ |
| (5) $\frac{-b+\sqrt{b^2-4ac}}{2a}$ | (6) $\frac{n(n+1)(2n+1)}{6}$ |
| (7) $e^{-a^2x^2} \cos bx$ | (8) $\frac{\sqrt{\pi}}{2a} e^{-(\frac{b}{2a})^2}$ |
| (9) $n \log_2 n$ | (10) $\sin \frac{\pi}{4}$ |

5. 设 $A=4, B=5, C=-2, D=9$, 写出下列表达式的值。

- | | |
|---|--|
| (1) $A < B$ | (2) $A + B <= C + D$ |
| (3) $A = B$ | (4) $A = D - B$ |
| (5) $A * B > = -C * D$ | (6) $A < B \text{ AND } C < D$ |
| (7) $B > C \text{ OR } C > D \text{ AND NOT } C > 0$ | (8) $(C + D) / (A - B) \text{ MOD } 2 \text{ OR } C + D > A$ |
| (9) $A \wedge 2 \setminus C > D \text{ MOD } A \text{ OR } B > = C$ | (10) $A > B = C \text{ AND } A < B > = D$ |

6. 将下列条件写成表达式。

- (1) 判断 x 是否为偶数。
- (2) 以 a, b, c 三条线段为边长, 能否构成一个三角形。
- (3) x 是否能同时被 3 和 7 整除。
- (4) 某年是否为闰年。(闰年的条件: 能被 4 整除但不能被 100 整除以及那些能被 400 整除的年份为闰年。)

第3章 基本语句、输入输出语句



3.1 赋值语句(LET 语句)

3.1.1 赋值语句的一般形式

赋值语句的一般形式为：

```
[LET]变量名=表达式
```

变量分为数值变量和字符串变量两大类,表达式也分为数值表达式(算术表达式、关系表达式、逻辑表达式)和字符串表达式两大类,变量的类型应与表达式的类型一致。也就是说,赋值语句分为两大类,即

```
[LET]数值变量名=数值表达式
```

```
[LET]字符串变量名=字符串表达式
```

例如

```
LET a = 3.141593/4
```

```
LET S$ = "a" + " book"
```

都是合法的赋值语句,而下面的写法是非法语句

```
LET a = "boy"
```

3.1.2 赋值语句的作用

赋值语句的作用是计算赋值号右边表达式的值并赋给赋值号左边的变量。所谓把“值”赋给“变量”,就是把值存到变量名所标识的存储单元中。变量存值的特点是,在定义变量时(或首次引用变量时)变量的初值为0(对于数值变量)或空串(对于字符串变量),当该变量被赋值时,新值就取代了旧值,而旧值将不复存在了。如果在赋值号右边的表达式中出现变量名,而它又不是赋值号左边的变量名,尽管该变量参与运算但它的值并不改变。

【例 3.1】 赋值语句中变量值的变化情况。

```
a = 4
b = SQR(a):PRINT a;b
b = a-1:PRINT a;b
b = a:PRINT a;b
a = a+1:PRINT a;b
END
```

输出结果为：

```
4 2
```

```
4 3
```