

计算机科学大众丛书



BASIC 语言及程序设计

刘克武 陈万方 编著



电子工业出版社

BAI 表 程序设计

此书请在下列时间内

78

刘克武 陈万方 编著

電子工業出版社

内 容 提 要

本书介绍 BASIC 语言及程序设计, 打破了以往讲解计算机语言的传统模式, 以实际问题为序安排章节。一章围绕一个中心问题讲解, 每学完一章后就可以动手编程序及上机实习。每一章基本上分四部分: “基础与概念”、“语句及语法”、“实践与训练”、“复习与测试”, 学以致用, 克服了“背着包袱”走路的讲解方法。

本书可以作为中学、职业学校以及大专院校教材, 也适合中等文化程度的读者自学。对于软件人员水平考试(初级程序员级), 计算机等级考试(二级)也是一本较好的学习参考书。

《计算机科学大众》丛书编委会

主 任 林定基

副主任 曹东启 丁嘉种

编 委 (以姓氏笔划为序)

刘兆毓 刘彦明 刘克武 朱家维

宋玉升 郑锡珪 彭裕禄 秦志斌

秘 书 袁 玫 徐海波

BASIC 语言及程序设计

刘克武 陈万方 编著

责任编辑: 宋玉升

*

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经销

北京科技大学印刷厂印刷

*

开本: 787×1092 毫米 1/16 印张: 23.375 字数: 550 千字

1996 年 7 月第 1 版 1996 年 7 月第 1 次印刷

印数: 1—4000 册 定价: 32.00 元

ISBN 7-5053-3379-8/TP·1299

前 言

BASIC 语言作为计算机的一种基本的高级语言,在微型计算机广泛普及应用的今天,应该说是一种适合初学者学习、上机使用、建立程序设计逻辑思维、打下计算机语言基础的一种便捷地语言工具,因为 BASIC 语言容易学习而且能很快掌握,可以通过人-机对话的形式直观地学习它,使用它解决计算问题及数据处理问题。

以往介绍 BASIC 语言的书籍已有不少,这无疑为推动计算机语言的学习和应用起了非常重要的作用。渴望学习 BASIC 语言的读者还在与日增多,编者愿意把十余年在教学中积累起来的经验整理编写出来献给读者朋友们。

对于初学计算机而又没有用过计算机的读者来说,面临着学习计算机语言的难度和使用计算机的难度,如果面对的实际问题又是自己不太熟悉的数学问题,这就像有三座大山挡住了自己进入计算机世界的去路,使初学者产生畏难情绪。能不能让初学者的学习热情与急切愿望不受损伤地去掌握计算机呢?答案是肯定的,这也是编写本书的初衷。

为了分散学习难度,本书打破了传统地讲授计算机语言的模式,不采用概念在前、语句为序的编写形式,而是从实际问题出发,语言为解决实际问题的方式介绍语言、使用语言,把读者引向面对实际问题运用语言,学以致用,不再背着包袱走路的学习方法。

本书每章构成一个学习周期,学完一章就可以马上试着编程和上机实践,由浅入深,循序渐进,步步深入。每一章都是结合一个实际问题分为“基础与概念”、“语句及语法”、“实践与训练”、“复习与测试”进行讲解。教学实践证明,这种讲解方式符合人们认识事物的规律。

全书分为两篇;第一篇共七章,由刘克武编写;第二篇是程序设计例题,由刘克武、陈万方共同命题,陈万方进行程序设计及编写,全书由徐志坚和丁嘉种先生审定。

本书可作为中学、职业学校、大专院校的教材;也可作为软件人员水平考试(初级程序员级)、计算机等级考试(二级)的培训教材。对于自学 BASIC 语言的读者也是一本较好的学习参考书。

书中不足之处,恳请专家和读者予以指正。

编 者

一九九五年六月

目 录

第一篇 BASIC 语言及程序设计

第一章 算术运算及整数运算	(1)
§1.1 基础与概念	(1)
1. BASIC 语言	(1)
2. BASIC 语言中的算术运算	(1)
3. BASIC 语言中的数	(2)
4. 数在 BASIC 中的存贮	(3)
5. 算术运算符及算术表达式	(4)
§1.2 语句及语法	(5)
1. BASIC 语言中的字符集	(5)
2. BASIC 语句的结构	(5)
3. 算术赋值语句 (LET 语句)	(6)
4. 输出语句 (PRINT 语句)	(7)
5. 程序结束语句 (END 语句)	(8)
§1.3 实践与训练	(8)
1. 变量、表达式在语句中的作用	(8)
2. 程序阅读训练	(11)
3. 程序设计初步	(12)
4. 流程图基础 (1)	(15)
5. 上机实验	(17)
§1.4 复习与测试	(18)
第二章 字符操作及输入、输出	(20)
§2.1 基础与概念	(20)
1. 字符常数及其操作	(20)
2. 字符型变量及字符表达式	(21)
3. 数据的输入与输出	(22)
4. 分隔符在输出中的作用	(23)
5. 空行及定位输出	(24)
6. 数及程序输入	(25)
§2.2 语句及语法	(26)
1. 键盘输入语句 (INPUT 语句)	(26)
2. 读数语句及数据语句 (READ/DATA 语句)	(26)
3. 数据指针回退语句 (RESTORE 语句)	(27)
4. 无条件转移语句 (GO TO 语句)	(28)

5. 暂停语句 (STOP 语句)	(28)
§2.3 实践与训练	(28)
1. 三种常数与三种变量	(28)
2. 溢出及误差	(31)
3. 程序阅读与分析	(32)
4. 流程图基础 (2)	(33)
5. 程序设计例题	(34)
6. 上机操作基础知识	(37)
§2.4 复习与测试	(40)
第三章 关系运算及逻辑运算	(45)
§3.1 基础与概念	(45)
1. 关系运算	(45)
2. 逻辑运算	(46)
§3.2 语句及语法	(47)
1. 条件转移语句 (IF.....THEN 语句)	(47)
2. 循环语句 (FOR/NEXT 语句)	(48)
§3.3 实践与训练	(50)
1. 认识关系表达式与逻辑表达式	(50)
2. 条件转移语句的形式及分支程序	(51)
3. 程序阅读与分析	(52)
4. 流程图基础 (3)	(55)
5. 程序设计例题	(57)
6. 操作命令与出错信息	(60)
§3.4 复习与测试	(63)
第四章 数据的成组操作及累加、累乘	(66)
§4.1 基础与概念	(66)
1. 怎样对大量数据进行累加	(66)
2. 怎样进行累乘	(67)
3. 数组的提出及使用基础	(69)
§4.2 语句及语法	(72)
1. 数组说明语句 (DIM 语句)	(72)
2. 程序说明语句 (REM 语句)	(73)
§4.3 实践与训练	(74)
1. 数据的成组输入	(74)
2. 数据的成组输出	(76)
3. 数组元素的使用及操作	(78)
4. 数组使用中的维、界和下标	(85)
5. 流程图基础 (4)	(91)

6. 上机实验	(94)
§4.4 复习与测试	(96)
✓第五章 数学函数及自定义函数的使用	(99)
§5.1 基础与概念	(99)
1. BASIC 语言的标准函数	(99)
2. 自定义函数	(100)
§5.2 语句及语法	(100)
1. 自定义函数语句 (DEF 语句)	(100)
2. 开关转移语句 (ON-GO TO 语句)	(102)
§5.3 实践与训练	(104)
1. 三角函数、对数函数的使用	(104)
2. 取整函数的功能与运用	(107)
3. 随机函数及其利用	(110)
4. 符号函数的作用	(115)
5. 程序设计例题	(118)
§5.4 复习与测试	(125)
第六章 字符函数及数据处理	(128)
§6.1 基础与概念	(128)
1. 字符函数在数据处理中的作用	(128)
2. BASIC 语言中的字符函数	(128)
§6.2 语句及语法	(129)
1. 测长函数的功能及使用	(129)
2. 取子串函数的作用	(130)
3. 左、右取子函数	(131)
4. 字符与代码的变换	(132)
5. 数据类型的转换	(135)
§6.3 实践与训练	(136)
1. 字符函数的功能及运用	(136)
2. 字符函数与信息变换	(141)
3. 字符函数与信息筛选	(143)
4. 数值问题非数值化处理	(146)
§6.4 复习与测试	(150)
第七章 程序的结构分析	(153)
§7.1 基础与概念	(153)
1. 子程序与主程序	(153)
2. 常用的程序结构	(154)
§7.2 语句及语法	(161)
1. 转子语句 (GOSUB 语句)	(161)

2. 返主语句 (RETURN 语句)	(161)
3. 开关转子语句 (ON-GOSUB 语句)	(162)
§7.3 实践与训练	(162)
1. 子程序的设计与试验	(162)
2. 程序结构的转化试验	(166)
§7.4 复习与测试	(168)

第二篇 基础程序设计例题及题解

第一章 数及数列	(173)
§1.1 数的分类	(173)
§1.2 寻找某些特定的自然数	(176)
§1.3 自然数的验证	(190)
§1.4 数列的形成	(193)
第二章 多项式及近似计算	(197)
§2.1 求多项式之值	(197)
§2.2 近似计算	(202)
第三章 解方程	(212)
§3.1 求方程 $f(x)=0$ 的根	(212)
§3.2 求方程组的解	(217)
§3.3 解不定方程	(219)
第四章 极值与排序	(227)
§4.1 极值	(227)
§4.2 排序	(231)
第五章 数组运算	(239)
§5.1 形成指定数组	(239)
§5.2 数组元素的查找及变换	(245)
第六章 数制转换	(254)
§6.1 十进制整数翻成二进制整数	(254)
§6.2 二进制整数翻成十进制整数	(256)
§6.3 十进制小数翻成二进制小数	(260)
§6.4 二进制小数翻成十进制小数	(262)
第七章 字符图形	(265)
§7.1 几何图形	(265)
§7.2 函数图形	(282)
第八章 表格输出	(290)
§8.1 输出数表	(290)
§8.2 直方图的制作	(302)

第九章 字符串操作	(307)
§9.1 字符串处理	(307)
§9.2 字符串应用	(318)
第十章 简单计算与应用	(325)
§10.1 基础计算	(325)
§10.2 实用计算	(332)
参考文献	(364)

第一篇 BASIC 语言及程序设计

第一章 算术运算及整数运算

本章的中心是要解决简单的计算问题。为此，要介绍数的使用、数的存放、算术运算的符号以及使用数和运算符号描写计算问题的方法，进而介绍组成计算问题程序的有关语句，最后达到编写简单程序的目的。

学完本章后即可在微型计算机（IBM-PC 及兼容机）上实践自己所编写的程序。

§ 1.1 基础与概念

1. BASIC 语言 (IBM-PC 及兼容机)

计算机语言是人和计算机交换信息的工具，计算机语言可分为机器语言、汇编语言及高级语言等三大类，BASIC 语言是高级语言的一种，在微型计算机上使用的 BASIC 语言，可以说它是一种会说话型语言。使用者可以操作着微型机的键盘，注视着微型机的显示屏幕把自己编写的程序录入到计算机中，计算机可以帮助你检查录入的程序是否正确，如发现错误，计算机可以在显示屏幕上提示出产生错误的原因和位置，待使用者修正无误后，就可以执行这个程序并在显示器或打印机上输出正确结果。整个过程是在人一机会话中进行的。

BASIC (Beginner's All-purpose Symbolic Instruction Code) 是“初学者通用符号指令代码”的英文缩写。BASIC 语言于 1964 年问世，早年多用于教学。由于这种语言语句少、容易学，在微型计算机广泛使用的今天，它仍然不失为初学者进入计算机世界首先必学的一种计算机语言。

随着 BASIC 语言的使用与发展，它的功能不断增加。使用该语言不仅可以编写计算问题的程序，也可以编写数据处理的程序。此外，在作图、动画、发声等方面也有相当能力。在 BASIC 语言中可以使用汉字，从而可以应用于人们的事务处理。在 BASIC 语言的发展中也形成了许多功能不同的版本。例如 True BASIC，它具有结构化特点；QuickBASIC；Turbo BASIC 是一种有窗口功能，可以与汇编语言、操作系统联用的 BASIC 语言。本书主要介绍只有十七个语言的基本 BASIC。

2. BASIC 语言中的算术运算

在数学中，一般把四则运算说成是算术运算。在 BASIC 语言中，算术运算的范围就宽多了，它不仅包括加、减、乘、除还包括指数运算、对数运算以及求三角函数值的运算等。可以说，凡是在 BASIC 语言中求数值性结果的运算都可以看成是算术运算。例如，求下列各式的值，

例 1. $1+2\times 3+4=?$

例 2. $(a+b)^2 - c = ?$

例 3. $2\pi R + cd = ?$

例 4. $absinx = ?$

例 5. $e^2 + \log 5 = ?$

例 6. $\sqrt{x} + |x| = ?$

都可以由算术运算来解决。其中函数的使用将在第五章介绍。

在数学中，求整数运算结果的方法是采用四舍五入。比如，计算结果为 38.5，用四舍五入的方法取值为 39。这种运算在 BASIC 语言中称为整数运算。整数运算可以自动完成，只要使用者需要求出整数结果，不管参加运算的数是整数还是非整数，都可以在 BASIC 语言中方便地求出结果来。

3. BASIC 语言中的数

在数学中为了研究数的属性和定义域，给数起了很多名字，诸如，整数、小数、分数、百分数；正数、负数、实数；自然数、奇数、偶数、素数（质数）；有理数、无理数、虚数；指数、对数、三角函数等等。由于 BASIC 语言本身描写数的能力所限，上面所列的数，有的可以直接使用，有的在 BASIC 语言中并不把它作为数来对待，因此不能直接使用。例如： $\frac{5}{8}$ ，20%，12‰， 3^2 ， $\sqrt{2}$ ， $\sin 30^\circ$ ， 2π ， $|-5|$ 等均不能直接使用。再如，会计记帐时常使用带有撇分节的数：12,346,78 以及开发票使用的大写数字，也不能直接用来参与计算。以上介绍，目的在于提醒读者注意，在 BASIC 语言中（或其它计算机语言），对于使用的数另有其定义。

BASIC 语言中的数有两大类，一类是数值性的数，称为数值常数或数值型常数；另一类是非数值性的数，称为字符常数或字符型常数，也叫字符串（在以后介绍）。数值型常数主要用于算术运算，字符型常数主在用于数据处理。下面介绍数值型常数的写法。

数值常数在 BASIC 语言中有两种表示方法，一种是不带指数的表示形式，例如：26，-48，+35，-4.6，+7.2，0，0.0，0.5，.5，-0.25，.75，-.125，+.375 等都可以在 BASIC 语言中直接使用。我们可以概括出数的结构规则：

① 在 BASIC 语言中使用的数值型常数是一种十进制数。

② 数由两部分构成，一部分是数的符号，在数最左边；另一部分是数的尾数，紧跟符号之后。正数用“+”表示，负数用“-”表示。尾数是由 0—9 及小数点排列而成。

③ 组成数的所有成分都排在同一横排上，而数的成分只包括 +、-、0~9、. 共 13 个。

根据上述规定，在 BASIC 语言中下列常数都是错误的（即不合乎语法）

例 1. $\frac{1}{2}$ （未在一横排上，横线不是构数成分）

例 2. 48\$（“\$”不是构数成分）

例 3. 80%（百分号不是构数成分）

例 4. $\sqrt{2}$ （未排在同一横排上，“ $\sqrt{\quad}$ ”不是构数成分）

例 5. 123 45.67（空格为构数成分）

例 6. 4,567.89 (逗号不是构数成分)

例 7. 8-3 (负号应在数的最左边)

例 8. 38^0 (未排在同一横排上)

第二种表示数值型常数的方法是附带指数的表示形式,这种表示形式是科学记数法在 BASIC 语言中的应用。当数的位数过多时,例如,1160000000,用科学记数法可写为 11.6×10^8 ,在 BASIC 语言中可写为 11.6E+8。英文字母 E 代表的是底数 10,其后的 +8 表示的是指数之值。再如,0.000000006 可以记为 6×10^{-9} ,在 BASIC 中写为 6E-9。由此不难看出,当数的位数过多时,不管它的值是大是小,都可以用带指数的形式表示在 BASIC 语言中。

需要指出的是,无论是什么样的计算机,它所能表示数的范围总是有限度的,这个范围称为数的界。当数过大(越过上界)、过小(越过下界)都会使计算机无法表示而出现“溢出”,正如杯子中的水过多而流出一样。大多数“溢出”是在运算中产生的。一旦出现“溢出”,计算机将中止当前的工作,这种现象称为“中断”。

4. 数在 BASIC 中的存贮

怎样把参加运算的数以及运算的中间结果和最后结果保留起来,这就是数的存贮问题。在计算机中设有内存贮器,内存贮器中备有很多“房间”,用来存贮数据及其它信息。数的存贮通常是使用“房间”的代号,称为地址,按照地址存贮想要保留的数据。使用 BASIC 语言或其它高级语言,数的存贮就十分简单了,使用者不必和内存贮器中的单元打交道,而是使用称为“变量”的空间来存贮数据和其它信息。

变量是学习 BASIC 语言(或其它语言)的重要概念。使用变量首先要给变量命名。下面介绍变量命名的规定及变量的特点。

(1) 变量命名的规定

在 BASIC 语言中,变量是以大写英文字母开头,后跟字母或数字,长度不大于 6 个字母或数字,最短的变量名是一个字母。例如: A, AB, AB1, ABCDEF, A1, B2 等都是符合规定(即合法)的变量名。使用这些变量名,就可以存贮数值型常数。至于怎么存贮又怎样把存贮的数据取出来使用,在介绍语句时将详细说明。

(2) 整型变量与实型变量

在 BASIC 语言及其它高级语言中,变量的实质是存贮空间的代号。需要存贮数据就得按照规定定义变量。为了存贮属性不同的数据,定义和使用的变量也不相同。例如: 49.6 和 32 这两个数的属性就不同,前者是实数,在 BASIC 中称为实型数; 2 是整数,亦称为整型数。对应于两类不同类型的数据,在 BASIC 语言中也有两种变量,一类叫实型变量,另一类叫整型变量。为了区分这两类不同的变量,可使用类型说明符号。“%”为整型变量的类型说明,实型变量不加类型说明,因此, A%, AB%, AB1%, ABCDEF%, A1%, B2%, 都是整型变量。不加类型说明的上述变量为实型变量。

(3) 整型变量的取整作用

当我们在计算机中需要求出整数结果时,可以借助整型变量的取整作用来实现。在 BASIC 语言中,整型变量有一个特殊功能,凡是存入到整型变量的数据可自动变为整数。下面举两个例子。

例 1. 将 48.4 放入 A% 中, 输出 A% 中的值时其结果为 48.

例 2. 将 -34.5 放入 B% 中, 输出 B% 中的值时其结果为 -35.

不难看出, 整数变量可以自动取整, 而且整型变量在取整时还能进行四舍五入 (IBM-PC 上). 从而保证了结果的精度.

有关变量的使用及变量本身的其它特点将在以后介绍.

5. 算术运算符及算术表达式

(1) 算术运算符及优先级

在 BASIC 语言中, 基本的算术运算有加、减、乘、除、乘方等五种, 并有规定的五种运算符如下:

BASIC 中的算术运算符				
加号	减号	乘号	除号	乘方号
+	-	*	/	^

上列五种运算的优先顺序 (即优先级), 乘方最高, 接下来是乘除, 优先级最低的是加减. 在同级运算中, 按由左至右的运算顺序. 为了改变运算的顺序, 在 BASIC 语言中可以使用括号, 但只设有小括号, 没有中括号和大括号. 括号的优先级高于上述五种运算.

(2) 算术表达式

使用 BASIC 语言或其它高级语言描写要进行的算术运算, 一般来说不能把数学上的算式照搬到 BASIC 语言中, 而需要按照某种规定对数学式进行变换或改写. 这种. 由数学式按照规定变换而来的, 且又完全能描写原来数学式的新运算式称为算术表达式. 下面是几个有代表性的变换实例.

数学式变为 BASIC 算术表达式	
范 例	变 换 规 则
(1) $1+2-3\times 4+5^2$ 表达式: $1+2-3*4/5^2$	(1) 数字照用. (2) 使用规定的算术运算符. (3) 所有字符都写在同一横排上.
(2) $x^2+2x-ab$ 表达式: $x^2+2*x-A*B$	(1) 英文字母大写. (2) 乘号不能省略.
(3) $\{[(5+4)\times 3]^2-2\}^3-1$ 表达式: $((5+4)*3)^2-2)^3-1$	(1) 只能使用小括号, 且成对出现.
(4) $y^{-2}+\frac{a}{-b}$ 表达式: $y^(-2)+A/(-B)$	(1) 两个运算符不能连在一起. (2) 必要时加括号.
(5) $\frac{(a+b)^2}{\pi}$ 表达式: $(A+B)^2/3.1416$ 或 $(A+B)^2/PI$	(1) 不能使用非 (大写) 英文字母. (2) 同一个数学式可以写出一个以上的算术表达式.

从以上举例我们可以概括出书写算术表达式的规则: 组成表达式的所有成分都要排在

同一横排上，且要使用 BASIC 语言规定的字符（见 BASIC 字符集）；乘号不能省略；必要时可使用括号；同一个数学式可以对应一个以上的算术表达式。

§ 1.2 语句及语法

在高级语言中，语句是使用者向计算机发出的指令。一个语句包含着一种或多种操作内容。将众多语句有机地排列起来就构成了一个程序，使用这个程序就可以让计算机完成一个特定的计算或处理问题。语法是由语句构成程序的原则及规定，不符合语法规定的语句属于错误语句或称非法语句。当程序中出现非法语句时，计算机将作出相应的提示，供使用者进行修正。学习语句要了解语句的功能及其含义，掌握语句的语法。

1. BASIC 语言的字符集

计算机语言，都有组成该语言的基本字符。BASIC 语言字符集是使用在 BASIC 语言中的全体字符。其中包括十个数字，26 个大写英文字母，运算符号，标点符号及专用符号等，详见下表。

BASIC 语言字符集	
数字	0 1 2 3 4 5 6 7 8 9
大写英文字母	A B C D E X Y Z
算术运算符 及连接符	加号 减号 乘号 除号 乘方号 左括号 右括号 + - * / ^ ()
关系运算符	大于 等于 小于 大于等于 小于等于 不等于 < = > >= <= <>
标点符号	逗号 分号 冒号 问号 圆点 省字号 双引号 叹号 , ; : ? . ' " !
专用符号	百分号 货币号 井号 空格号 % \$ # (空出一个位置)

字符集里的每一个字符都可以通过计算机的键盘键入计算机。每按一个键，显示器就显示一个相应的字符。按空格键，显示器不显示任何符号，而只是留出一个空位。

BASIC 语言的字符，有的一个字符具有多种功能，例如，加、减号既可以作加、减法的运算符，又可以作数的正、负号。凡不包含在 BASIC 字符集里的字符均不能使用，并称为非法字符。

2. BASIC 语句的结构

BASIC 语句的结构很简单，每个语句大都由语句标号、语句定义符、语句体三部分组成，这三部分的排列如下，

BASIC 语句的结构		
语句标号	语句定义符	语句体

使用者按照这个框架，根据自己的需要来组织语句。下面先介绍这三个部分的含义。

(1) 语句标号

语句标号也叫行号，用以表示该语句在一个程序中所处的位置。因此，行号的作用是用来规定语句的前后次序。行号规定使用正整数，数的范围与计算机内存的大小有关，一般在 0 ~ 60000 左右。

一个程序由若干语句组成，每个语句开头都有一个行号。一般都习惯于将第一个语句行号用 10，行号间隔也用 10。例如：

行号	语句定义符	语句体	
10	LET	A=3	(第一个语句)
20	LET	B=3	(第二个语句)
30	PRINT	A*B	(第三个语句)
40	END		(第四个语句)

行号间隔是为了便于修改程序而设置的。当需要增补新语句时，可以使用留出的间隔插入一个新语句，而不必改动原有的语句行号。

(2) 语句定义符

语句定义符的作用是表明该语句的功能，让计算机完成某种操作。因此，也可把语句定义符说成是予約语，用以完成人一机之间予約好的某种操作。

在 BASIC 语言中，语句定义符大都是英语中的单词或它的缩写，例如：

语句定义符	含 义
PRINT	表示输出的意思
END	表示结束的意思
INPUT	表示输入的意思
READ	表示读的意思
DATA	表示数据的意思

这些定义符也称为关键字，可见它们在语句中的作用是重要的。

(3) 语句体

语句体是 BASIC 语句中的主体，它描写一个语句的具体操作内容。例如：

10 PRINT 3+5-6

在这个语句里的定义符是 PRINT，而 3+5-6 就是语句体。这个语句的功能是完成算术运算后输出其结果为 2。语句体是由使用者自己编写的。

3. 算术赋值语句 (LET语句)

(1) 句型

行号	LET	变量 =	$\begin{cases} \text{常数} \\ \text{算术表达式} \\ \text{变量} \end{cases}$
----	-----	------	--

语句定义符是 LET (“让”的意思)，接下来的所有内容都是语句体。这里还有一个符号 “=”。该符号形式上是个等号，在此称为赋值号，它的作用是“让”符号右端的内容赋与 (即存入) 符号左端的变量中。

(2) 例句

10 LET A=5.6
20 LET B=2.5*3
30 LET C=A

执行这三个语句后，变量 A、B、C 中存入的结果如下，

A 中为: 5.6 B 中为: 7.5 C 中为: 5.6

如果我们想使用 A, B, C 中已存入的值时, 可以写出以下的赋值语句:

40 LET D=A*B+C

该语句可以自动取出 A, B, C 中的值, 进行算术运算。然后, 将计算结果 425.6 存入 D 中。

当我们想求出计算结果的整数值时, 可以使用以下语句:

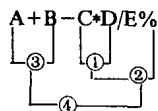
50 LET E%=3.4*1.2

执行该语句, 存入 E% 中的内容是 3.4 乘 1.2(4.08) 的取整结果, 即 4。

如果利用 A, B, C, D, E% 中已经存入的结果, 可以使用下面的赋值语句:

60 LET F%=A+B-C*D/E%

执行该语句, 首先是计算表达式的值, 表达式中包括加、减、乘、除, 根据运算的优先级别, 计算机顺序如下:



(先乘除, 后加减; 同级运算由左至右)

- ① $C * D$, 即 5.6×425.6 , 结果为 2383.36;
- ② $2383.36 / E\%$, 即 $2383.36 \div 4$, 结果为 595.84;
- ③ $A + B$, 即 $5.6 + 7.5$, 结果为 13.1;
- ④ $13.1 - 595.84$, 结果为 582.74。

完成上述四步运算后, 求出了表达式的值, 接着把此值存入变量 F% 中。由于 F% 是整型变量, 有取整作用, 当 582.74 存入 F5 时, 被取整, 在 F% 中的值为 583。这就是该语句的执行结果。

(3) 功能

赋值语句的第一个功能是存贮, 把参与运算的常数暂贮在一个变量中备用。

第二个功能是计算, 可以把算术表达式的值求出来, 又存贮到一个变量中去。

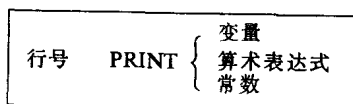
赋值语句的第三个功能是传送, 它可以把一个变量中的值传送到另一个变量中去, 利用此功能, 我们可以保留计算的中间结果和最后结果。

在实用中, 该语句的定义符 LET 可以省略。例如, 赋值语句可以简写为:

```
10 A=5.6
20 B=2.5*3
30 C=A
40 D=A*B+C
50 E%=3.4*1.2
60 F%=A+B-C*D/E%
```

4. 输出语句 (PRINT 语句)

(1) 句型



PRINT是语句定义符，含义是输出的意思。PRINT之后是语句体，也叫输出项。输出项可以单独出现，也可以混合出现，下面通过例子来介绍。

(2) 例句

10 PRINT A (输出A中已存入的内容)

20 PRINT 3*5+6 (输出表达式的值)

30 PRINT 1234.5 (直接输出常数)

以上三个例句，输出项均为一个。当需要一个以上的输出项时，两个输出项之间要用分隔符，逗号(或分号)隔开。例如：

40 PRINT A, B (输出A, B中的内容)

50 PRINT 1+2, 3*4 (同时输出两个表达式的值)

(3) 功能

输出语句的第一个功能是输出。当变量未存入任何值时，输出结果为0。

输出语句的第二个功能是计算。当输出项为表达式时，先计算，然后再输出。

5. 程序结束语句 (END 语句)

(1) 句型

行号	END
----	-----

END既是语句的定义符，又是语句体。

(2) 例句

① 40 END (表示程序在第40行结束)

② 60 END (表示程序在第60行结束)

(3) 功能

该语句的功能是使程序结束，一般放在程序的结尾。有时也不在结尾，而是放在程序的“逻辑结尾”。所谓逻辑结尾是指符合某种条件的结尾，将在后面介绍。

§ 1.3 实践与训练

1. 变量、表达式在语句中的作用

(1) 变量的特点

通过前面的学习，我们已经知道变量是内存单元的代号，使用变量可以存贮参加运算的初始数据、中间结果和最后结果；使用变量还可以从中取出已存入的内容参与新的运算。简单地说，利用变量既可以存入数据，又可以取出数据。那么，在什么情况下是存入数据，在什么情况下是取出数据呢？这就要看变量在语句中出现的位置了。请看下面的实例，

例1. 在变量A, B, 中存入数据。

10 A=3

20 B=5

30 END

在这个程序中，变量A, B都是接收来自赋值号右侧的常数。我们把这个程序再加入一个赋值语句，见例2。