

微型计算机组成
原理及^{语言软件}应用

李友堂 编

程序设计语言，数据库和汉字处理

第三册

张成明 吴宗粹 芮义昆 丁振国 甘圣予

目 录

第一章 IBM PC的BASIC语言	(1)
第一节 IBM PC BASIC 语言概述.....	(1)
第二节 IBM PC BASIC 的命令和语句.....	(3)
第三节 IBM PC BASIC 程序设计.....	(11)
第四节 IBM PC BASIC 的键盘及 BASIC 程序的编辑.....	(18)
第五节 应用实例.....	(21)
附录1.1 IBM PC BASIC 的标准函数.....	(26)
附录1.2 IBM PC BASIC 的出错信息.....	(28)
第二章 IBM FORTRAN 语言	(33)
第一节 概述	(33)
第二节 数据结构和表达式	(37)
第三节 控制语句	(43)
第四节 程序, 函数和子例行程序	(50)
第五节 输入输出语句	(53)
第六节 说明语句	(62)
第七节 内部函数	(66)
第三章 IBM FORTRAN 程序的编译	(69)
第一节 编译需要的系统支持.....	(69)
第二节 启动编译.....	(70)
第三节 编译程序的编译命令.....	(77)
第四节 程序实例的编译清单及说明.....	(81)
第五节 一个程序实例的编译过程.....	(86)
第四章 COBOL 语言基础	(94)
第一节 概述.....	(94)
第二节 COBOL 语言的组成元素.....	(99)
第三节 COBOL 标识部.....	(104)
第四节 COBOL 环境部.....	(105)
第五节 COBOL 数据部.....	(108)
第六节 COBOL 过程部.....	(117)
第五章 COBOL 程序举例	(130)
第一节 COBOL 程序举例	(130)
第二节 COBOL 上机步骤	(134)
第六章 dBASE - II 的基本概念	(137)
第一节 数据库的基本概念.....	(137)

第二节	dBASE - II 数据库的结构	(139)
第三节	dBASE - II 数据库管理系统概述	(140)
第七章	dBASE - II 的使用	(142)
第一节	引导 dBASE - II 的一般过程	(142)
第二节	建立数据库的一般过程	(144)
第三节	简单的检索操作	(147)
第四节	数据入库的令令	(157)
第五节	备份库的建立	(159)
第六节	数据库主文件上的修改操作	(161)
第七节	数据库的再组织	(167)
第八节	定位检索操作	(169)
第九节	报表的编制与使用	(171)
第十节	库文件上的统计操作	(175)
第十一节	数据的连接操作	(176)
第十二节	内存变量的定义与使用	(179)
第十三节	有关命令文件的命令	(180)
第十四节	dBASE - II 命令集	(187)
第八章	dBASE - II 程序设计与举例	(192)
第一节	结构程序设计及其在 dBASE - II 中的实现	(192)
第二节	命令文件的建立	(193)
第三节	命令文件的执行	(197)
第四节	命令文件的修改	(198)
第五节	dBASE - II 命令在命令文件中的使用	(198)
第六节	举例: 工资管理程序	(201)
第七节	dBASE - II 与高级语言之间的数据交换	(209)
第九章	汉字信息处理技术基础知识	(215)
第一节	汉字信息的特点	(216)
第二节	汉字的输入	(221)
第三节	汉字文字库	(230)
第四节	汉字的代码体系	(233)
第五节	汉字显示终端	(235)
第六节	汉字的印字技术	(240)
第七节	汉字处理系统	(242)
第十章	长城 0520 中西文兼容微型机系统	(247)
第一节	长城 0520 微型机主要性能	(247)
第二节	CC - DOS 操作系统的功能和结构	(248)
第三节	长城 0520A 的汉字输入方法	(265)

第一章 IBM PC的BASIC语言

关于IBM PC机系统的硬件结构及系统软件,在前两册书中已有详细论述,本章主要介绍它的 BASIC 语言。我们编写本章的宗旨是希望初学者通过本章内容的学习,能熟悉IBM PC BASIC的基本命令及语句,初步掌握程序设计方法,能编写一些简单的程序,在IBM PC机上操作运行。根据这个要求,在内容安排上,我们首先介绍IBM PC XT的开机过程及BASIC的启动,其次介绍IBM PC BASIC的功能、基本命令及语句。在此基础上,再进一步介绍BASIC的程序设计方法及程序编辑方法。最后一节给出了几个应用实例。此外,考虑到编程及调试程序的需要,在附录1.1及1.2中给出了IBM PC BASIC的标准函数(内部函数)及出错信息。为了便于初学者学习掌握基本内容,我们插入了一些例子,尽可能用实例来说明问题。

对于本章所述内容,为可靠起见,除绘图语句部分外,我们都尽可能地在IBM PC XT计算机上作了实际验证。

第一节 IBM PC的 BASIC 语言概述

1.1 IBM PC XT的开机过程及BASIC的启动

当系统连接无误并将电源插头接于220伏交流电源之后,分下述几种情况:

1、直接调用硬磁盘系统的开机

打开机箱右后边的电源开关,则系统加电并开始自检,将屏幕显示器右下侧亮度旋钮调到适当位置,则屏幕上显示出被检查的内存容量数,每次查16KB,最后显示出所配RAM的总容量后自检结束,喇叭发出一声响后随机转入读磁盘工作状态,先启动软盘,查无信息时又转到硬盘调入DOS,屏幕上将显示:

```
Current date is Tue 1-01-1980
```

```
Enter new date,
```

此时直接按回车键,或从键盘上输入月-日-年后按回车键,接着显示:

```
Current time is 0:00:34.21
```

```
Enter new time,
```

(注:上面显示的日期及时间仅是举例)

按回车键,或从键盘上输入时:分:秒后按回车键,则显示:

```
The IBM Personal Computer DOS
```

```
Version 2.00 (C) Copyright IBM Corp 1981, 1982, 1983
```

```
C> _
```

这表明已进入DOS命令状态,C>为DOS提示符。此时,打入BASIC或BASICA,则显示:

```
The IBM Personal Computer Basic
```

```
Version D2.00 Copyright IBM Corp.1981, 1982, 1983
```

61330 Bytes free

OK

OK为BASIC状态的提示符，在屏幕最下边还将显示出10个功能键的定义。这时即可从键盘输入相应的 BASIC 程序，或对盘上原有程序进行编辑、运行等操作，关于编辑方法，后面再叙述。

2、调用软盘系统的开机

如要调用软盘，可将相应的软盘片插入到驱动器中，然后按 `Ctrl + Alt + Del` 键，则系统重新启动，并将软盘的信息调入内存。

3、如果在软盘驱动器中插入的是诊断程序（Diagnostics）盘片，则系统进入诊断状态，可对系统各部分进行检查、诊断，以确定故障的原因及所在。

系统进入诊断状态时，屏幕上将显示：

The IBM Personal Computer DIAGNOSTICS

Version 2.00 (C) Copyright IBM Corp 1981, 1982

SELECT AN OPTION

0 - RUN DIAGNOSTIC ROUTINES

1 - FORMAT DISKETTE

2 - COPY DISKETTE

3 - PREPARE FIXED DISK FOR RELOCATION

9 - EXIT TO SYSTEM DISKETTE

ENTER THE ACTION DESIRED

? _

当打入 0，则进入诊断状态，可按屏幕提示符继续往下作；打入 1，则进入软盘的格式化，按屏幕提示符往下作；打入 2，则进行软盘拷贝，按屏幕提示符往下作。

4、如果插入的是操作系统盘片，则进入相应的操作系统工作状态。当采用 PC-DOS 2.0 时，其显示同于上述调用硬盘的情况，进入 BASIC 或 BASICA 的状态也相同。

5、如既不用软盘也不用硬盘时，在开机自检通过后，系统直接将固化的 BASIC 解释程序调入内存。此时屏幕的显示及键盘操作均同于上述的 BASIC 状态。

1.2 IBM PC BASIC的功能

IBM PC 机可以提供磁盘 BASIC 与高级 BASIC（即 BASICA）两种不同形式的 BASIC，它们的功能是向上兼容的，即高级 BASIC 不仅具有磁盘 BASIC 的全部功能，而且还扩充有一些其它功能。这两种 BASIC 的解释程序都存放在 IBM PC 的 DOS 磁盘上，磁盘 BASIC 要求 RAM 容量 32KB 以上，高级 BASIC 要求 RAM 容量 48KB 以上。

IBM PC BASIC 除具有一般 BASIC 的功能外，还扩充有一些特殊的功能，现列举如下：

1、可以显示 255 个扩充的字符集。除了一般常用的字母、数字及专用符号外，还可显示希腊字母以及其它在数学及科技文献中常用的各种符号。

2、绘图能力。配上彩色/图形显示器适配器，可画出点、线甚至绘制出完整的图形。

3、特殊的 I/O 设备。IBM PC 除用磁盘作 I/O 设备外，还配有扬声器可用以发

声，它还支持光笔及游戏操纵杆，增加用户兴趣和用途。

4、有一个内部“时钟”，可用以跟踪日期和时间。

5、有一个串行通讯接口 (RS232) 可供使用。

除此以外，在高级BASIC中，还扩充有以下特殊的功能。

6、事件陷阱程序可应某个特定事件的发生而陷阱（自动转移）到规定的程序行。这些事件包括通讯线路激活、功能键按下、游戏操纵杆按钮按下以及光笔激活。

7、先进的绘图能力。由于扩充了CIRCLE, PUT, GET, PAINT以及DRAW等绘图语句，可以借助彩色/图形监视器适配器创造更复杂的图形。

8、先进的音乐支持。用PLAY语句可以使扬声器发出音乐声调。

第二节 IBM PC BASIC 的命令及语句

本节介绍IBM PC BASIC 的命令及语句，供作程序设计参考。如上所述，IBM PC BASIC，特别是高级BASIC (BASICA) 的功能很强，包含的命令及语句很多，共达一百多条。为节省篇幅起见，我们只对少数较难懂的命令（或语句）举例作较详细地说明，对其它命令（或语句）只作简略介绍。读者如欲了解命令及语句的详细信息，建议参阅：

Personal Computer Hardware Reference Library "IBM BASIC"

2.1 命令（或语句）格式

典型的命令（或语句）格式如下：

CLOSE [[#] 文件号 [, [#] 文件号] ...]

DEF类型字母 [-字母] [, 字母 [-字母]] ...

DIM变量 (下标) [, 变量 (下标)] ...

INPUT [,] ["提示符" ,] 变量 [, 变量] ...

[LET] 变量 = 表达式

在上述例子中，

1、大写字母是关键字，应按所示送入；

2、小写斜体字的任何项都应提供；

3、方括号中的项目为选择项；

4、省略号 (...) 指出前项可以按需要重复任意多次；

5、方括号以外的所有标点符号（例如逗号，括号，分号，连字符与等号）应在所示地方引入。

2.2 命令

命 令	功 能
AUTO [n, m]	自动给出行号
BLOAD 文件名, 偏移量	将二进制数据（机器语言程序）读入内存
BSAVE 文件名, 偏移量	存贮二进制数据

CLEAR, n, m	清除程序变量, 任意设置存储区及堆栈
CONT	继续执行程序
DELETE n-m	删除程序行 n 到 m
EDIT n	显示所要改变的程序行 n
FILES 文件名	列出磁盘上与文件名相符的文件目录
KILL 文件名	删除一个磁盘文件
LIST n-m	在屏幕上显示程序行 n-m
LIST 文件名	在屏幕上显示文件清单
LLIST n-m	在打印机上打印程序行 n-m
LLIST 文件名	打印文件清单
LOAD 文件名	装入一个程序文件。可以包括 R 选择, 以便运行此文件
MERGE 文件名	将保存在磁盘上的程序与内存中的程序合并
NAME 文件名 AS 文件名	更改磁盘上的文件名
NEW	清除当前程序和变量
RENUM i, j, k	重编程序行。从 j 行开始, 开始行号为 i, 增量为 k
RESET	重新初始化磁盘信息, 与 CLOSE 相似
RUN 文件名	执行一个程序。应用 R 选择使文件为打开的
RUN n	从指定行 n 开始运行内存程序
TRON, TROFF	接通跟踪, 断开跟踪
SAVE 文件名	用给出的文件名将一个内存中程序存盘。A 或 P 选择是用 ASCII 代码或保护格式存贮
SYSTEM	结束 BASIC。关闭全部文件并返回 DOS 状态

2.3 非输入/输出语句

语 句	功 能
CALL 数字变量名(变量表)	调用一个机器语言程序
CHAIN 文件名 [, [行号] [, [ALL] [, [DELETE 范围]]]	调用一个程序并从当前程序传递变量于它。其它选择包括使用复盖, 开始运行指定行, 传递所有变量或删除一个复盖。

COM (n) ON	启动通讯联络
COM (n) OFF	停止通讯联络
COM (n) STOP	通讯联络被抑制
COMMON 变量表	把变量传递给链接程序
DATES=X\$	设置日期
DEF FN函数名 (宗量表) =表达式	定义一个数字函数或字符串函数
DEF类型字母范围	定义不需指明的变量类型。类型包括 INT, SNG, DEL或STR
10 DEF DBL L-P	定义L, M, N, O, P字母开头的变量为双精度型变量
DEF SEG=地址	定义当前的存储器段
DIF USRn=偏移量	定义机器语言子程序n的起始地址
DIM 下标变量表	说明数组的最大下标值并给它们分配存储空间
END	停止程序, 关闭所有文件并返回到命令级
ERASE数组表	从程序中删除数组
ERROR n	模拟出错代号n
FOR V=X TO Y[STEP Z] : NEXT V	循环次数控制。与NEXT语句构成循环
GOSUB n : RETURN	调用从指定行号n开始的子程序, 遇到RETURN语句, 从子程序返回
GOTO n	转移到指定行n
IF 逻辑表达式 [,] THEN语句 [ELSE语句]	如表达式为真 (非0), 执行THEN或GOTO语句, 否则执行ELSE语句或转下一行
IF 逻辑表达式 [,] GOTO n [ELSE语句]	
KEY (n) ON/OFF/STOP	启动、封锁或停止功能键或光标控制键的入陷转移
KEY ON/OFF	显示或关闭功能键
KEY LIST	显示全部功能键的值, 每个值的15个字符全被显示出来

KEY <i>n</i> , X\$	置功能键 <i>n</i> 为字符串 X\$ 的值
[LET] 变量 = 表达式	将表达式的值赋给变量
MID\$ (X\$, <i>n</i> , <i>m</i>) = Y\$	用字符串 Y\$ 替换 X\$ 的一部分, 从第 <i>n</i> 个字符开始, 替换 <i>m</i> 个字符

例:

```

10 A$ = "MARATHON, GREEN"
20 MID$ (A$, 10) = "FLORIDA"
30 PRINT A$
RUN
MARTHON, FLORIDA

```

MOTOR [状态]	用程序使盒式磁带机 (或磁盘) 接通或关闭
------------	-----------------------

例:

```

10 MOTOR 1 'Turn motor on
20 MOTOR 0 'Turn motor off
30 MOTOR 'Turn motor back on

```

ON COM (<i>n</i>) GOSUB <i>m</i>	对通讯活动启动陷阱程序
------------------------------------	-------------

ON ERROR GOTO <i>m</i>	从 <i>m</i> 行开始启动出错陷阱程序
------------------------	------------------------

例:

```

10 ON ERROR GOTO 100
20 LPRINT "This goes to the Printer"
30 END
100 IF ERR=27 THEN PRINT "Turn Printer on"
    , RESUME

```

这个例子表明如何陷阱一个常见的出错 - 忘记接通打印机

ON 表达式 GOSUB 行号表	转向表达式确定的子程序
------------------	-------------

ON KEY (<i>n</i>) GOSUB 行号	对指定的功能键或光标控制键启动陷阱程序
------------------------------	---------------------

ON PEN GOSUB 行号	对光笔启动陷阱程序
-----------------	-----------

ON STRIG (<i>n</i>) GOSUB 行号	对游戏操纵柄启动陷阱程序
--------------------------------	--------------

OPTION BASE <i>n</i>	指定数组下标的最小值, <i>n</i> 为 1 或 0
----------------------	------------------------------

PEN ON/OFF	启动或封锁光笔功能
------------	-----------

POKE <i>n</i> , <i>m</i>	将 <i>m</i> 个字节写入由 <i>n</i> 确定地址的内存区, <i>m</i> 范围 0~255, <i>n</i> 范围 0~65535
--------------------------	---

RANDOMIZE n	再激发随机数发生器
REM 注释	在程序中插入注释
RESTORE [行号]	恢复DATA 指针位置, READ 语句可重读。如指定行号则下一个READ语句读取被指定的 DATA 语句中的第一项
RESUME	错误改正之后, 继续执行程序
RESUME [0]	从出错语句重新开始执行
RESUME NEXT	从出错语句下一句重新开始执行
RESUME 行号	从指定行号重新开始执行
STOP	停止执行程序, 打印中断信息并返回到命令级
STRIG ON/OFF	启动或封锁游戏操纵柄功能
STRIG (n) ON/FF	启动或封锁游戏操纵柄陷阱
SWAP 变量1, 变量2	交换两个变量值
例: 10 A\$ = "ONE", B\$ = "ALL", C\$ = "FOR" 20 PRINT A\$, C\$, B\$ 30 SWAP A\$, C\$ 40 PRINT A\$, C\$, B\$ RUN ONE FOR ALL ALL FOR ONE	
TIME\$ = X\$	置时间
WAIT	在规定通道进展到指定的位模式之前, 暂停程序执行, 以便监视机器人入口端状态
WHILE 逻辑表达式 : (循环语句) : WEND	逻辑表达式为真 (非 0) 时, 执行一个循环 WEND 为循环终结语句

2.4 输入输出语句

BEEP	发喇叭声
CIRCLE (X, Y), r	画圆, 圆心为 (X, Y), 半径为r。其它选择, 允许画圆的一部分, 或改变纵横比, 画一个椭圆

CLOSE [[#]文件号[, [#]
文件号] ...]

关闭文件或设备

例:

CLOSE # 1, # 2, # 3

关闭与文件号 1, 2, 3 相关联的文件或设备

CLOSE

关闭所有打开的文件与设备

CLS

清屏

COLOR 前台, 背景, 边界

在文本模式中, 设置前台, 背景及屏幕边界的颜色

COLOR 背景, 调色板

在绘图模式中, 设置背景颜色及前台颜色的调色板

DATA 常数表

建立 READ 语句所用的数据表

DRAW 字符串

按字符串的规定画图

FIELD [#] 缓冲区编号; 宽
度 AS 串变量 [, 宽度 AS 串变
量] ...

在随机文件缓冲区中, 对变量分配空间

例:

FIELD 1, 20 AS N\$, 10 AS D\$

在 1 号随机文件缓冲区中, 把前 20 个字节分配给变量 N\$, 其后 10 个字节分给 D\$

GET [#] 文件号 [, m]

从随机文件读记录 m 到随机缓冲区

GET(X1, Y1)-(X2, Y2), 数组

从屏幕上读图形信息

INPUT "提示符" [,] 变量表

从键盘读数据

INPUT # 文件号, 变量表

从文件读数据

LINE (X1, Y1)-(X2, Y2)

在屏幕上画一直线, 其它参数允许画一个方框及其中内容

LINE INPUT "提示符"

从键盘读一行, 包括逗号及其它分割符

[,] 字符串变量

LOCATE 行, 列

定义光标位置, 其它参数允许规定光标的大小及它是否闪烁

LPRINT 表达式清单

在打印机上打印数据

LPRINT USING V\$

按 V\$ 指定格式打印数据

表达式清单

LSET 字符串变量=X\$	在区段中, 字符串X\$ 靠左排列, 用空格填满右面
RSET 字符串变量=X\$ 例: 10 A\$=SPA\$ (20) 20 RSET A\$=N\$ 按20个字符区段向右排列N\$ SPACE\$ (20) 函数产生20个空格的字符串	在区段中, 字符串X\$ 靠右排列, 左边留空格
OPEN 文件名 FOR 方式 AS [#] 文件号[记录长度]	为特定方式打开文件, 选择参数用于设置随机文件的长度
OPEN 方式, [#] 文件号, 文件名[, 记录长度]	OPEN 的另一格式
例: 10 OPEN "DATA" FOR OUTPUT AS # 1 或 20 OPEN "O", # 1, DATA 都是以顺序输出方式打开文件 "DATA" 30 OPEN "B: SSFILE" AS 1 LEN=256 或 40 OPEN "R", 1, "B: SSFILE", 256 都是为随机输入与输出方式打开 B盘上文件 "SSFILE", 记录长度为256	
OUT n, m	向通道n输出字节m
PAINT (X, Y), 颜色, 范围	给屏幕上规定范围设置颜色
PLAY 字符串	按字符串规定的声调奏乐
PRINT 表达式清单	在屏幕上显示数据
PRINT USING V\$, 表达式清单	按V\$ 指定格式在屏幕上显示数据
PRINT #f, 表达式清单	将表达式清单数据写入文件f
PRINT #f USING V\$, 表达式清单	按V\$ 规定的格式将表达式清单数据写入文件 f
PRESET (X, Y), 颜色	用背景颜色在屏幕规定位置画点
PSET (X, Y) [, 颜色]	在屏幕规定位置画点
PUT [#] 文件号 [, 记录长]	从随机文件缓冲区向文件写数据
PUT (X, Y), 数组, 操作	在屏幕上显示图形信息
READ 变量表	从 DATA语句建立的数据表中检索信息

SCREEN 方式, 成组, n页, v页	设置屏幕方式, 颜色接通/断开, 显示页和有效页
SOUND 频率, 持续时间	用扬声器发声
WIDTH 大小	置屏幕宽度。其它选择允许指定一个打印机或通讯文件的宽度
WRITE表达式清单	在屏幕上显示数据

例:

```
10 A=80, B=90, C$="THAT'S ALL"
```

```
20 WRITE A, B, C$
```

```
RUN
```

```
80, 90, "THAT'S ALL"
```

由此看出, WRITE与PRINT相似。但WRITE在显示项目之间插入逗号, 字符串两边加入双引号, 正数前不留空格

WRITE #f, 表达式清单 向文件 f 输出数据

2.5 PC BASIC 的语句关系图

为了帮助读者能较直观地理解 PC BASIC 的语句, 我们根据信息传递的观点, 绘制了下面的 PC BASIC 语句关系图。

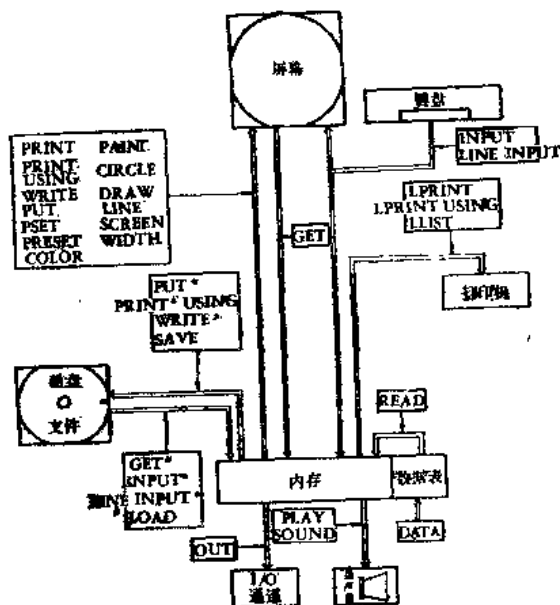


图1.1 PC BASIC的语句关系图

图中以内存 (RAM) 为中心, 外部设备包括屏幕, 键盘, 打印机, 磁盘, 喇叭及 I/O 通道。图中双箭头表示信息传递的方向, 单箭头表示语句 (命令) 的作用。由图

可清楚地看出，各语句（命令）的功能。例如：

1、INPUT, LINE INPUT 语句的作用是从键盘读取信息送到内存，同时送屏幕监视；

2、LPRINT, LPRINT USING及LLIST的作用是从内存把信息送到打印机打印；

3、READ语句从DATA语句建立的数据表中把信息读入内存；

4、GET语句从屏幕把信息读入内存；

5、PLAY和SOUND 语句从内存把信息送到喇叭奏乐；

6、GET #, INPUT #, LINE INPUT #语句及LOAD命令由磁盘把文件信息读入内存；

7、PUT #, PRINT #, PRINT # USING语句及SAVE命令从内存把信息读入磁盘作为文件存储；

8、OUT语句从内存把信息送入I/O通道；

9、PRINT, WRITE, PRINT USING, PUT, PSET, PRESET, COLOR, PAINT, CIRCLE, DRAW, LINE, SCREEN及WIDTH 语句的作用，都是从内存把信息送到屏幕处理。其中PRINT, WRITE及PRINT USING用于显示数据，DRAW, CIRCLE, LINE, PSET及PRESET用于绘制图形，COLOR PAINT用于在屏幕上设置颜色。

由图还可看出，在PC BASIC系统中，键盘仅用作输入。打印机及喇叭仅用作输出。而屏幕及磁盘，既用作输入，又用作输出。

第三节 IBM PC BASIC 程序设计

IBM PC个人计算机的BASIC与其它微型机BASIC虽然基本相同，但有其特点，在编写程序及在IBM PC机上运行其它BASIC程序时，必须注意这个问题。现简述如下。

3.1 行格式

BASIC中的程序行有如下格式：

nnnnnBASIC语句[* BASIC语句...] ['注释]用Enter键结束程序行。

“nnnnn”表示行号，为0~65529间一到五位数。

BASIC语句分为可执行语句与不可执行语句。可执行语句是若干程序指令，在执行程序时告诉BASIC作什么。不可执行语句如DATA或REM，当BASIC遇见它们时，对程序没有任何影响。一行中允许包括若干个语句，中间用冒号分开，但总字符不得超过255个。

注释可以加在末尾，用‘（单引号）将注释与该行的其余部分分开。

每个BASIC程序行开始要带行号。行号用于说明存储器中程序行的次序，而且还是程序转移和编辑的依据。

3.2 字符集

BASIC字符集包括字母、数字及专用字符共255个，这些字符BASIC都可识别它

们。在 BASIC 中，字母可用大写或小写，数字是从 0 到 9 的数字。

下列专用符号在 BASIC 中有特殊含义：

符号	名称	符号	名称
	空格		
=	等号或赋值号	!	感叹号或单精度型说明符
+	正号或并置号	&	意为“and”
-	负号	,	逗号
*	星号或乘法符号	.	小数点
/	斜线或除号	,	分号
\	倒斜线或整除号	:	冒号或语句分隔符
^	脱字符或取幂符号	"	单引号
(	左括号	?	问号或 PRINT 简化符
)	右括号	<	小于号
#	号数或双精度型说明符	>	大于号
%	百分号或整型说明符	"	双引号或字符串分隔符
\$	美元号或字符串说明符	_	字下线

3.3 保留字

某些字在 BASIC 中具有特殊的意义，称这些字为保留字。保留字包括所有 BASIC 命令、语句、函数名和运算符。保留字不能用作变量名。在程序中，用空格或其它语法允许的符号使保留字与其它部分分隔开，以使 BASIC 能识别它们。

3.4 常数

常数是执行期间 BASIC 使用的实际值。常数包括字符串常数和数字常数。字符串常数是包括在双引号中的一个字符序列，如“HELLO”等，最多可包括 255 个字符。数字常数有正数或负数。BASIC 中的数字常数不能包含逗号。数字常数有五种形式：

整数	- 32768 ~ + 32767
定点数	正实数或负实数
浮点数	以指数形式表示的正负数
十六进制数	用前缀 %H 表示，如 %H7E, %H32F
八进制数	用前缀 %O 表示，如 %O364

3.5 数的精确度

数字常数可以以整数、单精度数或双精度数的形式存储。以整数、十六进制或八进制输入的常数，存储在两个字节中，并且作为整数解释。单精度数，可以 7 位数字存储，以 7 位数字打印，但只有 6 位数字是精确的。双精度数以 17 位数字存储，并以多达 18 位数字打印出来。

3.6 变量及变量名

变量是 BASIC 中表示值的名字。包括数字变量及字符串变量。

变量名长度不限，但如超过 40 个字符，只有 40 个字符有效。变量名中字符可以是字母、数字、小数点，但必须以字母开头。变量名不应为保留字，但可以包括被嵌入的保

“留字”，例如：

```
10 EXP=5
```

不正确，因EXP是保留字。但

```
10 EXPONENT=5
```

是正确的，因EXP仅是变量名的一部分。

3.7 变量类型说明

字符串变量用符号\$说明，例如：

```
A$="SALES REPORT"
```

数字变量类型及所需存储字节数如下：

% 整型变量（2字节）

! 单精度型变量（4字节）

双精度型变量（8字节）

也可用BASIC语句说明变量类型，例如，DEF INT, DEF SNG, DEF DBL, DEF STR分别说明变量为整型，单精度，双精度，字符串变量。

如果变量类型未明显说明，则为单精度型。例如，

PI# 说明一个双精度变量

MINIMUM! 说明二个单精度变量

LIMIT% 说明一个整型变量

N\$ 说明一个字符串变量

ABC 表示一个单精度变量

3.8 数组

数组名及其类型用DIM语句说明，例如，

```
10 DIM B$(5)
```

产生一个名为B\$的一维数组，它的所有元素是不同长度的字符串并且初始值为空，即不包含任何元素。

```
20 DIM A(2, 3)
```

产生一个名为A的二维数组，由于没有说明类型，是单精度型的。所有数组元素的初始值为0。

除非指明用OPTION BASE语句改变数组最小下标，则最小下标为0。数组下标最大值为32767，数组最大维数为255。

如果你应用一个事先没定义的数组，则即自动定为一维的且最大下标为10。

3.9 BASIC如何改变数的精度

如果需要，BASIC可以改变一个数的精度。例如，

1、一个数赋给不同精度的变量时，这个数将以目标变量的精度存储，例如，

```
10 A%=23.42
```

```
20 PRINT A%
```

```
RUN
```