

新编 BASIC 语言和常用 DOS 命令 上机指导

王树亮 王里平 王二平 编著

天津科学技术出版社

新编 BASIC 语言和常用 DOS 命令
上 机 指 导

王树亮 王里平 王二平 编著

天津科学技术出版社

津新登字 (90) 003号

责任编辑: 刘万年

新编BASIC语言和常用DOS命令

上机指导

王树磊 王里平 王二平 编著

天津科学技术出版社出版

天津市张自忠路189号 邮编 300020

河北省雄县胶印厂印刷

新华书店天津发行所发行

开本 787×1092毫米 1/16 印张 8.25 字数 194 000

1993年9月第1版

1993年9月第1次印刷

印数: 1-- 10 200

ISBN 7-5308-1416-8/TP·43 定价: 4.50元

前 言

BASIC 语言是广泛流行和使用的一种计算机高级语言，世界上几乎所有的计算机都配有该语言，这为 BASIC 语言实验提供了极为有利的条件。BASIC 语言是一门实践性很强的课程，离开实验去学习等于纸上谈兵。随着计算机语言的发展，对 BASIC 语言的教学提出了新的要求。1989 年 7 月底在山西忻州召开了全国性的“BASIC 发展与前景研讨会”。会议指出，BASIC 语言在我国计算机教育普及当中作出了巨大贡献，尽管传统的 BASIC 语言存在版本繁杂、“方言”众多、程序结构性差、运行速度慢、语言特性过多依赖硬件等，但受我国目前国情，特别是微型机装备情况的制约，BASIC 语言还会继续存在下去。而且 BASIC 语言自身也在不断发展，在今后较长时间内，BASIC 语言还会在我国大专院校和中等专业学校中作为计算机入门课程。正确的态度应该是不断改进 BASIC 语言教学，不断修订 BASIC 语言教材。北京大学教授李大友曾说，可用 20 个学时讲授 BASIC 语言的全部内容。本书就是根据上述精神，为改进 BASIC 语言教学，编写的一本强化上机实验教材，它与汪明霓、王树亮等编写的《新编 BASIC 语言结构化程序设计教程》以及其它的《BASIC 语言》教材配套使用。书中以 APPLE-II、IBM-PC 及兼容机作为实验机型，详细讨论了十三个 BASIC 语言的上机实验和简略介绍了微机的安装、启动及常用 DOS 命令的使用。在第一章 BASIC 语言上机实验中，第一节主要介绍了关于数据的输入与输出过程；第二、三、四、五、六节主要讲述了程序的转移、循环以及利用数据、函数、字符串编程的设计方法；第七、八两节分别介绍了 APPLE-II 机和 IBM-PC 机的图形绘制；第九节专门介绍了程序的设计方法；第十、十一、十二节

介绍了文件的操作。在第二章常用 DOS 命令的简介中，第一、二、三节以 IBM-PC 机为例，介绍了微机的安装、启动、键盘、专用组合键的使用和常用 PC-DOS 命令的使用格式及其功能；第四、五节以 APPLE-II 机为例，介绍了 APPLE-II 机的安装、启动、键盘的使用和常用 DOS3.3 命令的格式及其功能；第六节主要介绍了打印机的使用方法。

本书可作为大专院校、中等专业学校、职业学校及成人继续教育的微机上机实验教材，也可供广大自学青年阅读参考。

本书第一章第一、二、三、四、五、六节由王树亮编写；第一章第七、八、九、十、十一、十二节由王里平编写；第二章由王二平编写，最后由王里平负责对全书各章进行修改、统稿。

在本书编写过程中，参考了汪明霓、张世英、陶龙芳等人的著作，在此一并表示感谢。

由于编者学识水平和实验教学经验的限制，编写过程又较仓促，错误和不足之处在所难免，希望广大读者批评指正。

编 者

目 录

第一章 新编 BASIC 语言上机指导	(1)
第一节 数据的输入与输出	(3)
第二节 程序流向的转移	(10)
第三节 程序的循环	(20)
第四节 数组	(27)
第五节 函数	(34)
第六节 字符串	(43)
第七节 PC 机图形绘制	(51)
第八节 APPLE-II 机上的图形绘制	(57)
第九节 子程序	(63)
第十节 PC 机顺序文件的应用	(67)
第十一节 PC 机随机文件的应用	(74)
第十二节 APPLE-II 机的文本文件	(82)
第二章 常用 DOS 命令上机指导	(95)
第一节 PC 机的安装与启动	(97)
第二节 PC-DOS 下的键盘使用	(100)
第三节 常用 PC-DOS 命令简介	(102)
第四节 APPLE-II 微机的安装、启动和键盘操作	(106)
第五节 APPLE-II 机的 DOS 命令简介	(108)
第六节 打印机操作	(111)
附录	(115)
附录一 APPLE-II 机 BASIC 错误信息表	(117)
附录二 IBM-PC 机 BASIC 错误信息表	(119)
附录三 ASCII 码表	(123)

第 一 章

新编 BASIC 语言上机指导

第一节 数据的输入与输出

一、实验目的

- 1.掌握常见输入、输出语句的用法;
- 2.掌握常见的几种显示格式的控制方法;
- 3.学会清屏、暂停、结束和注释语句的用法。

二、实验内容

(一) 显示语句 (PRINT)

格式: <行号> PRINT <显示内容>

功能: PRINT 语句可将<显示内容>显示在屏幕上。

输入程序 1-1

```
10 PRINT 5
20 PRINT -15
30 PRINT 5+2
40 PRINT "ABC"
50 PRINT "5+2"
```

运行程序

RUN (RUN 的用法, 请参阅 DOS 命令一节)

```
5
-15
7
ABC
5+2
OK
```

程序的 30 行执行后显示出的是 5+2 的值 7, 而 50 行显示出的为 5+2, 请注意“ ” (双引号) 在显示语句中的用途。

(二) 显示格式

1. 标准显示

格式: <行号> PRINT <显示内容 1>, <显示内容 2>,
.....

或

```
<行号> PRINT <显示内容 1>,  
<行号> PRINT <显示内容 2>,
```

.....

功能：这种显示法是把屏幕分成均等的 5 栏（APPLE-II 为 3 栏），各显示项显示在每栏的开头，用逗号分隔各显示项是分栏显示的控制方法。

2. 紧凑显示

格式：<行号> PRINT <显示内容 1>; <显示内容 2>;

.....

或

<行号> PRINT <显示内容 1>;

<行号> PRINT <显示内容 2>;

.....

功能：这种显示法是使用后一个显示项紧跟着前一个显示项显示，用分号隔开各显示项是紧凑显示的控制方法。

3. 换行显示

格式：<行号> PRINT <显示内容 1>

<行号> PRINT <显示内容 2>

或

<行号> PRINT <显示内容 1>; PRINT <显示内容 2>

功能：这种显示方式是使<显示内容 1>在上一行开头显示后，<显示内容 2>在下一行开头显示。

这种语句的格式中如果某个显示项后没有逗号或分号，那么语句结束符（回车或冒号）起换行显示的作用。

4. 空行显示

格式：<行号> PRINT <显示内容 1>

<行号> PRINT

<行号> PRINT <显示内容 2>

功能：这种显示方式，使<显示内容 2>与<显示内容 1>空一行显示，若要空二行显示<显示内容 2>，可在中间再写入：

<行号> PRINT

以上的几种显示方式可混合使用。

输入程序 1-2

```
10 PRINT 5, "3+2", "ABC"
```

```
20 PRINT "XYZ",
```

```
30 PRINT -3.14
```

```
40 PRINT
```

```
50 PRINT "3+2"; "="; 5, "5-2";
```

```
60 PRINT "="; 3
```

```
70 PRINT
```

```
80 PRINT "ABC": PRINT 3+2
```

```
90 PRINT "THE END"
```

运行程序

RUN

5 3+2 ABC

XYZ -3.14

3+2=5

5-2=3

ABC

5

THE END

OK

(三) 打印语句 (LPRINT)

格式: <行号> LPRINT <打印内容>

功能: LPRINT 语句与 PRINT 语句的功能、格式控制等都相同, 只是将 <打印内容> 命令用打印机打印在纸上, 而不是显示在屏幕上。

大家可将程序 1-1 中的 PRINT 改为 LPRINT, 在打印机上打印出结果来。在 APPLE-II 机上无 LPRINT 语句, PRINT 语句具有打印功能。

(四) 赋值语句 (LET)

格式: <行号> LET <变量> = <常量、变量或表达式>

功能: 执行该语句时, 计算机先计算表达式的值, 再将此值赋给“=”左边的 <变量>。这里的 LET 可以省略。

输入程序 1-3

10 LET A = 5

20 LET B = 7

30 C = 9

40 C = C+1: B = B+5

50 D = A+B

60 PRINT "A=";A,"B=";B,"C=";C,"D=";D

70 PRINT H

80 A\$="ABCD"; B\$="****"

90 PRINT A\$,B\$

运行程序

RUN

A=5 B=12 C=10 D=17

0

ABCD * * *

OK

这里注意 $C=C+1$ 、 $B=B+5$ 这样的赋值形式在以后会经常用到。另外在 60 行中 H 没有被赋值，显示出的是 0。

(五) 交换语句 (SWAP)

格式: <行号> SWAP <变量 1>, <变量 2>

功能: 执行该语句后, 将两个变量的值进行互换。进行交换的两个变量的类型必须相同, 即数值型变量只能与数值型变量互换, 字符串型变量只能与字符串型变量互换。

输入程序 1-4

```
10 A = 5: B = -4
```

```
20 A $ = "ABC": B $ = "XYZ"
```

```
30 SWAP A, B
```

```
40 SWAP A $, B $
```

```
50 PRINT "A ="; A, "B ="; B, "A $ ="; A $, "B $ ="; B $
```

运行程序

RUN

```
A = -4      B = 5      A $ = XYZ      B $ = ABC
```

OK

在 APPLE-II 机上不能运行 SWAP 语句, 要交换两变量的值时可用下面赋值方式:

```
10 A = 5: B = -4
```

```
20 A $ = "ABC": B $ = "XYZ"
```

```
30 C = A: A = B: B = C
```

```
40 C $ = A $: A $ = B $: B $ = C $
```

```
50 PRINT "A ="; A, "B ="; B, "A $ ="; A $, "B $ ="; B $
```

RUN

```
A = -4      B = 5      A $ = XYZ      B $ = ABC
```

这里 30 行中的 $C=A:A=B:B=C$ 就是进行 A 与 B 的值之间互换; 同样 $C $ = A $:A $ = B $:B $ = C $$ 是 A \$ 与 B \$ 之间的值互换, 请熟悉这种互换方式。

(六) 键盘输入语句 (INPUT)

格式: <行号> INPUT <"提示">; <变量 1>, <变量 2>,

功能: INPUT 语句也是给变量赋值。在执行该语句时会停下来等待你由键盘直接键入常量数据再打一个回车, 变量便被赋与键入的值, 程序才会接着运行, 否则会一直等待。

变量既可以是数值型, 又可以是字符串型的。由键盘键入数据时, 必须按顺序一一对应变量的个数和类型, 数据与数据间也要用逗号隔开。在运行程序时, “”(双引号)中的内容

原文显示在屏幕上，当“提示”后接“;”时，则显示“?”号；若接“,”时，则无“?”号。另“提示”可以省略。

```
输入程序 1-5
10 INPUT "A=";A
20 INPUT "R$=";R$
30 INPUT B,S$,C,T$
40 PRINT "A=";A,"R$=";R$,"B=";B
50 PRINT "S$=";S$,"C=";C,"T$";T$
```

运行程序

```
RUN
A=? 5
R$ = * * *
? 10, 3+5, 15, ABC
A=5          S$ = * * *          B=10
S$ =3+5      C=15                T$ =ABC
OK
```

(七) 读数语句和置数语句 (READ 和 DATA)

格式: <行号> DATA <常量 1>, <常量 2>,
<行号> READ <变量 1>, <变量 2>,

功能: DATA 语句和 READ 语句也是给变量赋值的语句。DATA 语句为非执行语句(可放在程序的任何地方,但一般是放在程序的最前面或最后面)。READ 语句是执行语句,当执行时 READ 语句中所列的变量将从 DATA 语句中逐个读取常量来赋给变量。

READ 语句中只能是变量,它们既可以是数值型变量,又可以是字符串型变量,两变量间需用逗号隔开,句末无逗号。DATA 语句中只能是常量,常量间也需用逗号隔开,常量的个数、类型必须与 READ 语句中变量的个数、类型和顺序一一对应,字符串型常量的双引号可省略。

```
输入程序 1-6
10 DATA -3.5, ABC, DEF, 7
20 READ A,B$,C$,D
30 DATA "5+2","=", 7
40 READ E$,F$,G
50 PRINT A,B$,C$,D
60 PRINT E$;F$;G
```

运行程序

```
RUN
```

-3.5 ABC DEF 7

5+2=7

OK

在执行 READ 语句时，有一个读指针指向 DATA 语句中第一个常量，所以第一个变量被赋予的是 DATA 语句的第一个常量，然后读指针移向第二、三……常量，按顺序一一读取而不会错乱，从而使 READ 语句的变量读完一条 READ 语句后，又移向下一条 DATA 的开头。

(八) 恢复数据区语句 (RESTORE)

格式: <行号> RESTORE <行号>

功能: 当执行该语句时，使 RESTORE 后的 READ 语句中的变量，不再对应地读取 DATA 语句的下一个常量值，而是读取 RESTORE 语句指定 <行号> DATA 语句中的第一个常量值。RESTORE 后面的 <行号> 省略后，则读指针指向第一条 DATA 语句中的第一个常量值。

输入程序 1-7

```
10 DATA 1,3,5,7,9
20 DATA 2,4,6,8,10
30 READ A,B,C,D,E,F,G,H,I,J
40 RESTORE 20
50 READ K,L,M
60 RESTORE
70 READ N,O,P,Q,R
80 PRINT A,B,C,D,E,F
90 PRINT F,G,H,I,J
100 PRINT K,L,M
110 PRINT N,O,P,Q,R
```

运行程序

RUN

```
1   3   5   7   9
2   4   6   8  10
2   4   6
1   3   5   7   9
```

OK

(九) 清屏语句 (CLS)

格式: <行号> CLS

功能: 该语句执行后，清除屏幕上显示的内容，将光标置于屏幕的左上角，但不清除

内存中的程序。

大家可直接打入 CLS 观察其清屏功能。在 APPLE-II 机中清屏语句用 HOME，其格式与功能同 CLS。

(十) 暂停语句 (STOP)

格式: <行号> STOP

功能: STOP 语句被执行时, 可使程序运行暂停, 直到键入继续运行命令 CONT。在程序 1-7 的第 90 行与第 100 行之间加入

```
95 STOP
```

运行程序时, 当显示出

```
Break in 90
```

再键入 CONT 命令可继续运行。

(十一) 结束语句 (END)

格式: <行号> END

功能: END 语句表示程序执行的结束, 并关闭所有的文件, 使内存各变量释放且返回到 BASIC 操作状态。END 并不一定是程序的最后一条语句。

(十二) 注释语句 (REM)

格式: <行号> REM <注释内容>

功能: REM 为非执行语句, 注释内容只是对程序的目的、程序设计或程序的其它问题做一些说明, 计算机在运行程序时对注释内容是不理会的。

第二节 程序流向的转移

一、实验目的

- 1.熟悉无条件转向语句,条件分支语句和条件执行语句的格式和功能;
- 2.掌握计数器和累加器在程序流向中所起的作用。

二、实验内容

(一) 无条件转向语句的使用

格式: <行号> GOTO <行号>

功能: 当执行该语句时, 将无条件地转到 GOTO 后指定的那个 <行号> 继续运行。

例 2-1 计算在不同时间内自由落体下降的高度 ($h = 1/2gt^2$)

输入程序 2-1

```
10 INPUT "T="; T
20 H=9.8 * T * T / 2
30 PRINT "H="; H
40 GOTO 10
```

试分别求出 1 秒、1.8 秒、3.5 秒、6 秒时的 H 值。

运行程序

RUN

T=? 1

H = 4.9

T=? 1.8

H = 15.876

T=? 3.5

H = 60.025

T=? 6

H = 176.4

T=?

由运行结果可见, 每当程序运行到第 40 行语句时, 就无条件地转向第 10 行语句继续运行。要终止无条件转向语句继续执行, 可同时按 CTRL 键和 BREAK 键 (或 CTRL+C 键)。

(二) 条件分支语句的使用

格式: <行号> IF <关系表达式> THEN <行号> ELSE <行号>

功能：当 IF 后的 <关系表达式> 的条件成立时，程序转向 THEN 后指定的那个 <行号> 去运行；若 IF 后的 <关系表达式> 条件不成立时，则程序转向 ELSE 后指定的 <行号> 运行。ELSE 部分可省略，省略后当条件不成立时，程序转向执行 IF 后的下一行语句继续运行。

在 APPLE-II 机上，条件分支语句的格式应为：

<行号> IF <关系表达式> THEN <行号>

无 ELSE 部分。功能同上句省略 ELSE 部分相同。

例 2-2 某单位对用户每月用水按下述规定计费

① 当一户用水量平均每人每月不超过 2 吨时，每人收费 0.1 元；

② 当一户用水量平均每人每月多于 2 吨时，超过 2 吨的部分每吨按 0.16 元计费。

设 P 为某户当月的用水量，N 为某户人数，W 为某户当月水费。

输入程序 2-2

```
10 INPUT "P,N=",P,N
20 IF P / N > 2 THEN 40 ELSE 30
30 W = 0.1 * N : GOTO 50
40 W = 0.1 * N + (P - 2 * N) * 0.16
50 PRINT "W=" ; W : GOTO 10
```

试按下表分别输入 P 和 N 数据，求出 W 值。

P	N
5	3
6.7	2
8.2	4
12.8	5

运行结果如下

RUN

P,N= 5,3

W=0.3

P,N= 6.7, 2

W=0.632

P,N= 8.2, 4

W=0.432

P,N= 12.8, 5

W=0.948

P,N=