

高级汉字BASIC和中西文DBASEⅢ编程技术

高级汉字

BASIC

和

中西文

DBASEⅢ

编程技术

徐文

气象出版社

高级汉字 BASIC 和中西文 dBASEⅢ编程技术

徐 文 编

气 象 出 版 社

(京) 新登字 046 号

内 容 简 介

本书是作者多年实践经验的积累和总结。本书通过大量的应用实例,披露了一些鲜为人知的编程秘密。另外,本书结构新颖,一改以往同类书籍的编排方式,在试图揭示编程技巧的基础上,将 BASIC 和 dBASE III 合二为一,并展示了两者的相互进行数据通讯的技术诀窍。本书所提供的大量实用程序,均具有极高的通用性,任何读者均可方便地参考使用。

本书适合众多的计算机使用者、编程人员及大专院校的有关师生阅读参考。

2008/08

高级汉字 BASIC 和中西文 dBASE III 编程技术

徐 文 编

责任编辑 林雨晨

(北京海淀区白石桥路 46 号 邮政编码 100081)

北京振华胶印厂印刷

新华书店总店科技发行所发行 全国各地新华书店经售

* * *

开本: 787×1092 1/16 印张: 12.5 字数: 314

1992 年 2 月第一版 1992 年 2 月第一次印刷

印数: 1—5000 定价: 9.00 元

ISBN 7-5029-0750-5 / TP.0031

前 言

目前有关计算机编程技术书籍虽然不少,但真正能使计算机程序编制人员掌握编程技术的书籍却是不多,高级汉字 BASIC 和中西文 dBASEⅢ是目前用得最多的高级语言软件系统,为此,非常有必要编著一部实用的编程技术书籍,本书就为此而作。

本书是作者自 1987 年以来在程序设计实践中,总结出来的一部实用编程技术总集。全书共分五大部分 20 章。第一部分,讨论了高级汉字 BASIC 语言的程序设计、数据处理、报表编制和文件开发技术,最后将已出现的 BASIC 程序库分成三类,并分别作编程方法分析。第二部分介绍了 dBASEⅢ各种命令(含未正式公布命令,在其它资料上是查不到的)、变量(分类与互换)、函数、文件(各类文件的建立方法及其应用)以及作者的 dBASEⅢ编程方法论。第三部分介绍了 dBASEⅢ的实用编程技术,包括屏幕输入输出技术、菜单程序技术、子程序技术、数据库操作技术、报表编制技术和系统保密技术。第四部分则专门研究了 BASIC 和 dBASEⅢ之间的数据通讯技术(混合编程技术、与字处理软件通讯技术和相互之间的通讯技术)。第五部分列出了几个必要的附录,举了一个编程实例: FACAS(氧化铝成本核算系统)系统简介,该附录,作者曾在《贵铝科技》1989 年第 4 期上予以介绍。

本书系作者许多程序设计的独创技术和一些先进实用的编程方法的汇集,书中大量引用来自由作者独自开发的“财务管理系统”(Financial Control System, 书中简称 FCS)的例子程序,披露了许多鲜为人知的编程秘密,并以其高度的通用性,使之可方便地成为读者自己开发的应用程序的一部分。

BASIC 语言和 dBASEⅢ联合应用是一项具有开创性的工程,本书的编写方法也是首次性的,它避开了高级语言命令或语句的冗长介绍,而注重应用技术。

本书读者可分四类:

1. 已成为计算机用户的各企事业单位的实际工作人员;
2. 探讨高级语言的编程技巧和专门从事计算机应用程序开发的科研单位和个人;
3. 大专院校计算机软件与应用专业以及其它如经济信息、企业管理、系统工程、管理工程等专业的广大师生;
4. 其它试图应用高级汉字 BASIC 和中西文 dBASEⅢ编制实用程序系统的各类人员。

作者的目的是让更多的人编制出更加完美的实用程序,如本书在某个方面或某一问题上对读者有所帮助,那么,作者将为此而高兴。

作者

1990 年 12 月于贵州铝厂

目 录

前言

第一部分 BASIC 编程技术

第1章 BASIC 概述	(1)
§ 1.1 BASIC 语言发展史	(1)
§ 1.2 BASIC 语言特点	(1)
§ 1.3 BASIC 起动与操作命令	(2)
第2章 BASIC 程序设计	(4)
§ 2.1 程序设计(I): 屏幕输出设计	(4)
§ 2.2 程序设计(II): 陷阱设计技术	(9)
§ 2.3 程序设计(III): 功能段子程序设计.....	(13)
§ 2.4 程序设计(IV): 模块子程序设计.....	(16)
§ 2.5 程序设计(V): 机器语言子程序设计.....	(18)
第3章 BASIC 数据处理.....	(22)
§ 3.1 BASICA 命令行参数	(22)
§ 3.2 BASIC 数据	(23)
§ 3.3 数据存取技术	(23)
§ 3.4 随机数据存取技巧	(27)
§ 3.5 数据排序技术	(30)
§ 3.6 数据检索技术	(38)
第4章 BASIC 报表编制.....	(41)
§ 4.1 表头设计	(41)
§ 4.2 表体设计技巧	(42)
§ 4.3 动态报表设计	(44)
§ 4.4 报表脱机打印技术	(50)
第5章 BASIC 文件开发.....	(52)
§ 5.1 数据库文件	(52)
§ 5.2 索引文件	(54)
§ 5.3 执行文件	(54)
§ 5.4 文本系统	(55)
第6章 BASIC 常用程序库.....	(58)
§ 6.1 科学计算程序库	(58)
§ 6.2 系统工程分析程序库	(59)
§ 6.3 简单游戏程序库	(61)

第二部分 dBASEⅢ编程基础

第7章 dBASEⅢ概述	(66)
§ 7.1 dBASEⅢ发展概述	(66)
§ 7.2 dBASEⅢ性能指标	(67)
§ 7.3 dBASEⅢ系统功能指标	(68)
§ 7.4 dBASEⅢ操作键	(68)
第8章 dBASEⅢ命令	(71)
§ 8.1 dBASEⅢ命令格式和分类	(71)
§ 8.2 dBASEⅢ辅助命令	(72)
§ 8.3 dBASEⅢ基本命令	(72)
§ 8.4 dBASEⅢ SET TO 命令	(73)
§ 8.5 dBASEⅢ SET ON / OFF 命令	(74)
§ 8.6 dBASEⅢ扩展命令	(75)
第9章 dBASEⅢ变量与函数	(76)
§ 9.1 dBASEⅢ数据	(76)
§ 9.2 dBASEⅢ数组变量	(78)
§ 9.3 dBASEⅢ函数	(80)
第10章 dBASEⅢ文件及其建立	(83)
§ 10.1 文件的分类	(83)
§ 10.2 数据文件的建立	(84)
§ 10.3 命令文件的建立	(87)
§ 10.4 打印文件的建立	(88)
第11章 dBASEⅢ编程方法	(89)
§ 11.1 结构化程序设计	(89)
§ 11.2 最优化程序设计	(90)
§ 11.3 程序调试与纠错方法	(92)
§ 11.4 程序设计备忘录	(94)

第三部分 dBASEⅢ编程技术

第12章 屏幕输入、输出技术	(95)
§ 12.1 声音、色彩和图象程序设计	(95)
§ 12.2 数据输入设计	(101)
§ 12.3 数据输出设计	(102)
§ 12.4 数据输入、输出设计	(106)
§ 12.5 数据输出描述符	(108)
§ 12.6 多窗口编程技术	(109)
第13章 菜单程序设计技术	(113)

§ 13.1 菜单工作方式	(113)
§ 13.2 “DO CASE”菜单技术	(113)
§ 13.3 宏代换菜单技术	(115)
§ 13.4 光标引导菜单技术	(117)
§ 13.5 色彩引导菜单技术	(118)
§ 13.6 功能键菜单技术	(119)
§ 13.7 库菜单技术	(120)
§ 13.8 混合编程菜单技术	(124)
第 14 章 子程序技术	(128)
§ 14.1 子程序的概念	(128)
§ 14.2 屏幕格式子程序	(128)
§ 14.3 模块功能子程序	(128)
§ 14.4 过程子程序	(129)
§ 14.5 外部子程序	(130)
第 15 章 数据库操作技术	(132)
§ 15.1 库结构操作技术	(132)
§ 15.2 数据修改与更新技术	(134)
§ 15.3 数据追加技术	(135)
§ 15.4 数据删除技术	(136)
§ 15.5 数据复制技术	(137)
§ 15.6 数据排序技术	(140)
§ 15.7 数据检索技术	(142)
§ 15.8 数据统计汇总技术	(156)
§ 15.9 多工作区操作技术	(157)
第 16 章 报表输出技术	(160)
§ 16.1 制表基础技术	(160)
§ 16.2 报表设计方法	(161)
§ 16.3 dFORMAT 仿真表格程序设计	(161)
第 17 章 系统保密技术	(165)
§ 17.1 文件名加密技术	(165)
§ 17.2 口令设置保密技术	(165)
§ 17.3 程序加、解密技术	(169)
§ 17.4 dBASE III 编译技术	(171)

第四部分 数据通讯技术

第 18 章 数据通讯基础	(172)
§ 18.1 数据通讯的必要性	(172)
§ 18.2 数据通讯接口文件	(172)

§ 18.3 程序连接技术	(173)
第 19 章 与字处理软件通讯	(174)
§ 19.1 编辑数据文件	(174)
§ 19.2 BASIC 数据追加技术	(174)
§ 19.3 dBASEⅢ数据追加和修改技术	(175)
第 20 章 dBASEⅢ与 BASIC 数据通讯	(177)
§ 20.1 顺序方式读取接口文件	(177)
§ 20.2 随机方式读取接口文件	(177)
§ 20.3 读取 dBASEⅢ接口文件通用程序	(178)
§ 20.4 BASIC 语言直接读取 dBASEⅢ数据库	(179)
附录 FACAS 简介	(183)
参考文献.....	(189)

第一部分 BASIC 编程技术

第 1 章 BASIC 概述

§ 1.1 BASIC 语言发展史

自 1946 年世界上第一台电子计算机“ENIAC”诞生以来，人们经过长期的实践探索，到五十年代末才创造出了一种“程序设计语言”，又称“算法语言”。接着，一批高级语言随之诞生，如 FORTRAN、ALGOL、PASCAL、COBOL 和 PL/1 等语言。

为了尽快地普及计算机，人们需要一种程序结构简单、浅显易懂的计算机语言。于是，人们从 FORTRAN 语言中提炼、简化、扩充而得 BASIC 语言。1960 年它首先出现在美国。BASIC 是 Beginner's All-purpose Symbolic Instruction Code 的字头缩写，意为“初学者通用符号指令代码”。

BASIC 语言常用的有单用户基本 BASIC，高级 BASIC 和多用户分时 BASIC。在 IBM PC 计算机上，迄今为止，BASIC 创始人它为它开发了四种版本，分别是 Cassette BASIC，Disk BASIC，Advanced BASIC，True BASIC。前三种版本的 BASIC 均具有包容性，也就是说 Disk BASIC 涵盖了 Cassette BASIC 全部功能，并增加了一些新的功能。同样，Advanced BASIC 涵盖了 Disk BASIC 全部功能，并增加了 Disk BASIC 所没有的功能。这三种版本，都是解释性的，同时也有编译 BASIC 存在。第四种版本 True BASIC 与前三种版本的 BASIC 也有互通性。但 True BASIC 不仅仅是高级 BASIC 在功能上的增加，它还有许多更加优秀的特点。这里就不赘述，有兴趣的读者可参看有关书籍。

本书第一部分立足于 BASIC 第三版本，介绍 Advanced BASIC 的程序设计技巧。这是因为 Advanced BASIC 是目前计算机上用得最为普及的 BASIC 版本。

§ 1.2 BASIC 语言特点

BASIC 语言有以下几个特点：

- (1) BASIC 语言比较简单。编写程序时，书写格式不很严格，便于掌握和使用。
- (2) BASIC 语言是一种交互式会话语言，它可通过控制台键盘或终端与计算机进行人机对话。用 BASIC 语言，可在计算机上，边算边改，直到得出满意的结果，十分方便。

(3) BASIC 语言不仅可用于科学计算,而且还具有一定的数据处理能力。特别是高级 BASIC 增强了字符串处理和自选打印格式的功能,用于小型数据处理和事务管理是很方便和灵活的。

(4) BASIC 语言可扩充性强。几十年来,许多厂商和用户在最基本的 BASIC 语言(只有十七条语句)基础上,结合实际的需要进行了各种各样的扩充,使得它运用更加灵活,功能更加完善。现在其音响、图形、通讯等方面的功能是其它高级语言无法比拟的。

(5) BASIC 语言也不是十全十美的,它也受到一些限制。例如数的范围、简单变量的个数、数组的维数、自定义函数的自变量个数等等都受到一定限制。象数组,其最大维数不能够超过 255,每维所含的元素个数不能超过 32767。使用过程中,要特别引起注意。

§ 1.3 BASIC 起动与操作命令

1.3.1 BASIC 直接起动

BASIC 版本不同,其起动方式也不一样。

由于 Cassette BASIC 是存放在计算机的只读记忆体 (read only memory) 内的,所以,只要一打开计算机不加任何干涉,几秒钟后,即可进入 Cassette BASIC 状态。而 Disk BASIC, Advanced BASIC 和 True BASIC 必须在 DOS 操作系统下才能起动。在“A>”提示符下,如果插入 A 驱动器的软磁盘上有 BASIC 或 BASICA 文件,那么只要键入 BASIC[回车],或键入 BASICA[回车]即可起动。

1.3.2 BASIC 自动起动

一般情况下,计算机先要进入汉字 DOS 操作系统状态,再进入 BASIC 状态,接着可调入 BASIC 程序,最后开始运行。显然过程繁琐。为了减少操作步骤,我们可用批处理方式来解决。例如,要使 A 盘上名为 FCS.BAS 的 BASIC 文件在 DOS 状态下一性起动,则建立一个批处理文件,问题就解决了:

```
A>COPY CON:FCS.BAT[回车]
```

```
A>ECHO OFF[回车]
```

```
A>CLS[回车]
```

```
A>BASIC FCS[回车]
```

```
A>^Z[回车]
```

在 A>提示符下,直接键入 FCS[回车]即可运行 BASIC 程序 FCS.BAS。

1.3.3 几个最重要的 BASIC 操作命令

1. BASICA 命令

理所当然地, BASICA 命令是高级 BASIC 系统中的第一命令。BASICA 命令的格式如下:

```
BASICA [filespec][ / F:files ][ / S:bsize ][ / C:combuffer ][ / M:max workspace]
```

(1) 不作任何选择的情况下, 操作 A>BASICA[回车], 它将计算机置于 BASIC 状态下;

(2) 在 BASICA 后接一个程序名<选择 filespec 项>, 它将 BASIC 程序载入内存并执行;

(3) 置 /F: 项, 设置一个能同时打开的文件数目。省略时, 机定数目为 3, 最大设定值为 15;

(4) 置 /S: 项, 设置随机文件的暂存区大小。机定值为 128, 最大设定值为 32767;

(5) 置 /C: 项, 设置异步通讯处理暂存区的大小。输出暂存区一般定为 128, 最大设定值为 32767。高速传送下, 置 /C: 1024, /C: 0 将解除 RS-232C 的功能;

(6) 置 /M: 项, 设定 BASIC 的最大使用范围。最大设定使用量为 64K, 如省略 /M: 项, 则 64K 的内存空间将全被 BASIC 系统占用。

2.SAVE 命令

该命令将你编制好的, 在内存中的 BASIC 程序存在磁盘上。该命令有三种方式:

- (1) SAVE filename 以压缩二进制方式存入;
- (2) SAVE filename, A 以 ASCII 码方式存入;
- (3) SAVE filename, p 以编码二进制方式存, 并加密。

3.LOAD 命令

该命令将磁盘中的 BASIC 程序载入内存中。它有两种存入方式:

- (1) LOAD filename 调 BASIC 文件入记忆体内;
- (2) LOAD filename, R 调 BASIC 文件入记忆体内, 并执行。

4.RUN 命令

该命令执行内部存储器中的 BASIC 程序。它有五种执行方式:

- (1) RUN 从开头开始执行;
- (2) RUN line 从第 line 行开始执行;
- (3) RUN line, R 从第 line 行开始执行, 数据文件保持开启状态;
- (4) RUN filename 将软磁盘上的 BASIC 文件载入记忆体内, 从头开始执行;
- (5) RUN filename, R 将软磁盘上的 BASIC 文件载入记忆体内, 从头开始执行, 并使数据文件保持开启状态。

5.其它操作命令

BASIC 操作命令有二十多条, 其它如: AUTO, BLOAD, BSAVE, CLEAR, CONT, DELETE, EDIT, FILES, KILL, LIST, LLIST, MERGE, NAME, NEW, RENUM, RESET, SYSTEM, TROF, TROFF 等等, 请读者参阅有关资料, 这里不再细述。

一般地说, 使用计算机, 首先要了解硬件环境, 熟悉 DOS 操作系统, 然后才能应用某种高级语言编制程序。就 BASIC 语言而言, 在非常熟练地掌握 DOS 系统后, 必须了解 BASIC 语言的操作命令, 这样才能进入编程阶段。从下一章开始, 我们将从 BASIC 语言的编程技巧出发, 详细地探讨 BASIC 语言在实际应用中的各种问题。

第二章 BASIC 程序设计

§ 2.1 程序设计 (I): 屏幕输出设计

屏幕输出是程序执行过程中,所有在 CRT (显示器) 上的反映。屏幕输出应是计算机程序设计人员首先要解决的问题。屏幕输出包括音响、图象和文字资料三种。很多程序设计人员,由于忽视屏幕输出的设计工作,只重视打印机输出结果,屏幕输出很混乱,操作非常费劲,常常出错。为此,本节首先研究一下屏幕输出的程序设计。

2.1.1 声 (音乐) 设计

1. BEEP 或者 PRINT CHR \$(7)

该命令使计算机扬声器发出“嘟”声。BEEP 和 PRINT CHR \$(7) 相同。

例如:在一个工作“菜单”下,选择工作路径,可供选择的键为 1~6 六个数字键。在你没有按上这几个键 (错按) 时,让计算机发出叫声,提醒注意。设计方法如下:

```
100 R$ = INPUT$(1) 从键盘接收一个字串[程序 1]
110 H = VAL(R$) 将字串变成数字
120 IF H = 0 OR H > 6 THEN 130 ELSE 140
130 BEEP:GOTO 100 发出嘟声
140 REM 程序段.....
999 END
```

前例中, 130 句可写成: 130 PRINT CHR \$(7):GOTO 100, 效果一样。

2. PLAY

该语句按字符串内容演奏音乐,其用法详见 BASIC 使用手册。下面举例说明用法。后面的程序用 PLAY 语句演奏台湾校园歌曲《蜗牛与黄鹌鸟》。

```
10 REM 《蜗牛与黄鹌鸟》' [程序 2]
20 GOSUB 120
30 PLAY "MB T80 XD$;XE$;"
40 PLAY "MB XF$;XG$;"
50 PLAY "MB XH$;XI$;"
60 PLAY "MB XD$;XE$;"
70 PLAY "MB XF$;XG$;"
80 PLAY "MB XH$;XI$;"
90 PLAY "MB XD$;XE$;"
100 PLAY "MB XF$;XG$;"
110 PLAY "MB XH$;XI$;":GOTO 160
```

```

120 D$ = "O3L8B16BB16BG#16B16EO4C#O3B4"
125 E$ = "B16BB16BG#16F#16EG#F#4"
130 F$ = "F#.G#16BB16B16G#G#16F#16EE"
135 G$ = "F#.G#16EE16C#16O2BO3C#O2B4"
140 H$ = "O3B16BB16BG#16F#16EO4C#O3BB16O4C#16"
145 I$ = "O3EF#EF#G#4F#4E3"
150 RETURN
160 REM 程序段 .....: END

```

程序 3 是一个比较高级的程序，其优美的音调和无限循环而不影响程序执行的特点将使你永远难忘。它是巧用音乐函数 PLAY(n)来完成的。

```

10 REM 《爱的罗曼斯》 [程序 3]
20 CLS:A=0:N=1
30 IF PLAY(0)=0 THEN GOSUB 70
40 A=A+1
50 LOCATE 20,14:COLOR 9,0:PRINT A;
60 GOTO 30
70 ON N GOTO 80,90,100,110,120,130,140,150,160,170,180,190,200,210,220,230,240,250,260
80 PLAY "MB L9 B < BG > B < BG > B < BG > B < BG > A < BG ":N=2:RETURN
90 PLAY "MB > G < BG > G < BG > F < BG > E < BG > E < BG ":N=3:RETURN
100 PLAY "MB > G < BG > B < BG > > E < BG > > E < BG > > E < BG ":N=4:RETURN
110 PLAY "MB > > E < BG > > D < BG > > C < BG > > C < EC BEC ":N=5:RETURN
120 PLAY "MB AEC AEC BEC > C < EC BF+D+":N=6:RETURN
130 PLAY "MB > C < F+D+ BF+D+ > D+ < F+D+ > C < F+D+ BF+D+":N=7:RETURN
140 PLAY "MB B < BG > A < BG > G < BG > G < BG > F < BG ":N=8:RETURN
150 PLAY "MB > E < BG > F < BA > F < BA > F < BA > F < BA ":N=9:RETURN
160 PLAY "MB > G < BA > F < BA > E < BG > E < BG > E < BG. L18 < EB > EGB > E...":N=10:RETURN
170 PLAY "MB L18 G+ < BG > G+ < BG > G+ < BG > G+ < BG > F+ < BG ":N=11:RETURN
180 PLAY "MB > E < BG > E < AF+ > D+ < AF+ > D+ < AF+ > D+ < AF+":N=12:RETURN
190 PLAY "MB > D < AF+ > D+ AF+ > > C+ < F+D+ > C+ < F+D+ > C+ < F+D+":N=13:RETURN
200 PLAY "MB > C+ < F+D+ > D+ < F+D+ > C+ < F+D+ > C+ < G+E BG+E":N=14:RETURN
210 PLAY "MB BG+E BG+E > C+ < G+E > D+ < G+E > E < G+E":N=15:RETURN
220 PLAY "MB > E < G+E > E < G+E > E < G+E > E- < G+E > D < G+E":N=16:RETURN
230 PLAY "MB > C+ < EC+ > C+ < EC+ > C+ < EC+ > C+ < EC+ BEC+":N=17:RETURN
240 PLAY "MB AEC+ G+ < BG+ > G+ < BG+ > G+ < BG+ > G+D+ < A":N=18:RETURN
250 PLAY "MB > AD+ < A > F+D+ < A > E < BG > E < BG > E < BG...":N=19:RETURN
260 PLAY "MB L18 < EB > EGB > E...":N=1:RETURN

```

在程序的适当地方，用 PLAY 语句插一小段音乐，增加程序运行的生动性，是很有意义的。

3.SOUND freq, duration

该语句也使计算机扬声器发出声音。不过，其音域要比前两个语句宽广得多。程序 1 中 130 语句也可改成：

```
130 SOUND 800,1:GOTO 100
```

上语句表示扬声器的发声频率为 800Hz，声音持续时间为一个节拍。因此，语句 SOUND 也可用于演奏乐曲。下面是演奏日本民歌《樱花》的小程序。

```
10 REM 《樱花》' [程序 4]
20 DIM M(100):T=0:PLAY "MB"
30 FOR I=7 TO 88:M(I)=36.8*(2^(1/12))^(I-6):NEXT
40 FOR I=0 TO 6:M(I)=32767:NEXT
50 READ A,B:IF A=-1 THEN 120
60 SOUND M(A),B:SOUND 32767,1:GOTO 50
70 DATA 49,8,49,8,51,12,0,4,49,8,49,8,51,12,0,4,49,8,51,8,52,8,51,8
80 DATA 49,8,51,4,49,4,45,16,44,8,40,8,44,8,45,8
90 DATA 44,8,44,4,40,4,39,16,49,8,49,8,51,12,0,4,49,8,49,8,51,12,0,4
100 DATA 40,8,44,8,45,8,49,8,51,4,49,4,45,8,44,16
110 DATA -1,-1
120 T=T+1:IF T=2 THEN 130 ELSE RESTORE:GOTO 50
130 REM 程序段 .....:999 END
```

2.1.2 图象设计

表 2.1 中分辨率汉字模式下各作图语句的用途

序号	语 句	SCREEN 1	SCREEN 2	用 途
1	CIRCLE	✓	✓	画一椭圆或半椭圆
2	COLOR	✓	×	设置前景、背景颜色
3	DRAW	✓	✓	绘制任意图形
4	GET	✓	×	读屏幕某范围的绘图资料
5	LINE	✓	✓	画线、四方形、四方形面
6	PAINT	✓	×	将屏幕下特定范围涂上色
7	PUT	✓	×	将绘图资料写于屏幕
8	PSET	✓	✓	画一点
9	PRESET	✓	✓	擦一点
10	POINT	✓	✓	求屏幕某一点的颜色代号

与图象设计有关的语句有：

CIRCLE, COLOR, DRAW, GET, LINE, PUT, PAINT, POINT, SCREEN, WIDTH 等。表 2.1 列出了各有关语句的用途。其中，命令 COLOR 当前景色选为 16 至 31 时，显示字符出现闪烁。注意：在高分辨率模式下，各语句使用将有变化，请小心使用。有关图象设计的例子很多，程序 5, 6, 7 都可见到，这里就不再举例。

在中分辨率汉字模式下，由于 11 行显示，有图象不能变色，显得很单调。我们可用下列命令语句实现颜色变化：

OUT 985,n (n 为前景颜色,取值在 0—254 之间,16 种色循环变化,背景色始终为黑色)。

2.1.3 定位显示

光有声音和图象是不起作用的。屏幕输出最重要的是将我们要做的事情或要了解的事用中文在屏幕上显示出来,这就需要定位显示。在 BASIC 中,定位显示是通过 LOCATE 和 PRINT 语句来完成的。例如:

```
400 CLS:LOCATE 4,10:PRINT "企业工资管理系统"
```

上面语句首先清屏,然后在第 4 行第 10 列处显示:“企业工资管理系统”。

定位显示与音响、图象结合使用,将使屏幕显示更为生动,也能够起到突出文字资料的作用。程序 5 便是一例。

```
10 KEY OFF:SCREEN 2:CLS           '[程序 5]
20 LOCATE 3,17:PRINT "Financial Control System"
30 LOCATE 6,16:PRINT "[平行分步法成本核算程序]"
40 FOR I=1 TO 20:LINE (I*2,I+60)-(640-I*2,139-I),1,B:NEXT I:K=0:A=2:NI=0:NJ=640:NK=16
50 LOCATE 25,1:PRINT "按空格键继续...";
60 K=K+1:SOUND 200,1:FOR I=NI TO NJ STEP NK:LINE (318,137)-(I,119),A:NEXT I:A=0:IF INKEY$
=" " THEN 90
70 IF K MOD 2=0 THEN A=2:NI=0:NJ=640:NK=16 ELSE A=1:NI=640:NJ=0:NK=-16
80 GOTO 60
90 REM 程序段 ... :999 END
```

2.1.4 屏幕覆盖技术

屏幕动画的产生,主要是依靠屏幕覆盖技术来完成的。屏幕覆盖技术有三种:跳动、滑动和摆动。

1. 跳动

用背景色覆盖定位内容,然后又重新显示定位内容。或者,是不同内容的互相覆盖。程序 35 游戏程序即采用此种覆盖技术。

2. 滑动

用背景色局部覆盖的办法,起到滑动的效果。局部覆盖分等距覆盖和非等距覆盖两种。要使一个长方形面从左到右滑动,程序 6 是一种设计方法,是一种等距滑动。

```
10 CLS:LOCATE 8,25:PRINT "滑动.....";           '[程序 6]
20 FOR I=1 TO 100:LINE (50+I,160)-(300+I,170),1,BF
30 LINE (49+I,160)-(49+I,170),2
40 NEXT I:END
```

3. 摆动

事实上,摆动是一种非等距覆盖技术。程序 5 在执行过程中有一条“线”在左右摆动,就是如此。

2.1.5 INKEY\$ 变数与动画

覆盖技术反映了程序设计的技巧问题。然而这种技巧要与应用程序相结合,必须依靠 INKEY\$ 变数。在程序运行过程中,当执行碰到 INKEY\$ 时,它要求从键盘读一个字元。由于它有“读没读到一个字元都要往下执行”的特性,我们可结合条件语句、图象、定位显示和覆盖技术,创造出一个生动活泼的“动态”画面。

例如:在一个财务管理系统中,成本费用子系统的设计就反映了这种动画技术的实际应用。程序 7 在运行中,当你键入某种选择后,计算机自动让它跳动起来,允许你作进一步的肯定后,再往下执行。

```
10 CLS:SCREEN 2:A=1:FOR I=1 TO 10:LINE(20+I,1+I)-(90-I,180-I),A,B:NEXT' 程序 7
20 DATA "企业管理费表","车间经费表","生产费用表","退出本系统"
25 FOR I=1 TO 4:READ A$(I):NEXT
30 FOR I=3 TO 8:LOCATE I,7:PRINT MID$("成本费用报表",(I-3)*2+1,2);SOUND137+I,1:NEXT
40 LOCATE 4,20:PRINT "系统菜单":LOCATE 5,16:PRINT "1";A$(1):LOCATE 6,16:PRINT "2";A
$(2):LOCATE 7,16:PRINT "3";A$(3):LOCATE 8,16:PRINT "4";A$(4)
50 LOCATE 9,18:PRINT SPC(18);LOCATE 9,18:PRINT "请选择:";
60 FOR I=100 TO 300 STEP 200:LINE(I,53)-(I,161),1:NEXT:LINE(160,71)-(160,143),1
70 FOR I=161 TO 53 STEP -18:LINE(100,I)-(300,I),1:NEXT
80 R$=INKEY$:IF R$="" THEN 80
90 H=VAL(R$):IF H=0 OR H>4 THEN 100 ELSE 110
100 FOR I=300 TO 1000 STEP 100:SOUND I,5:NEXT:GOTO 50
110 LOCATE 9,18:PRINT "选对了吗[Y/N]";
120 R$=INKEY$:LOCATE H+4,25:PRINT SPC(10);LOCATE H+4,25:PRINT A$(H);IF R$="" THEN
120
130 IF R$="Y" OR R$="y" THEN 150 ELSE IF R$="N" OR R$="n" THEN 50 ELSE 140
140 FOR I=1000 TO 300 STEP -100:SOUND I,5:NEXT:GOTO 120
150 ON H GOTO 170,830,1080,160
160 CLS:END
170 REM 企业管理 ... :820 RUN
830 REM 车间经费 ... :1070 RUN
1080 REM 生产费用 ... :2000 RUN
```

当然,在程序中,一味地追求屏幕输出效果,从而增加程序的长度,占用大量空间,这是不可取的。我们主张在适当的情况下注意一下屏幕输出效果,对用户来说,肯定是有益无害的。千万不要本末倒置,影响主体程序的设计。

§ 2.2 程序设计(Ⅱ): 陷阱设计技术

2.2.1 程序陷阱设计

汉字 BASIC 用其中断陷阱技术, 可随时捕捉到某些可能发生的随机事件。陷阱设计的一般过程是首先编制好续元处理子程序, 再在程序的开头放一条陷阱设计语句, 用以指出续元子程序的入口。当程序运行完每一条语句, BASIC 解释程序都去查找一下期待中断事件是否发生。如果发生, 则转入该事件续元子程序, 处理完后, 再返回到被打断的程序继续执行下去。用户一般允许用户设置的陷阱有:

1. 音乐陷阱, 2. 出错陷阱, 3. 功能键陷阱, 4. 时钟陷阱, 5. 异步通讯陷阱, 6. 光标陷阱, 7. 按钮陷阱。

2.2.2 音乐陷阱设计

设置音乐陷阱, 目的是让程序能边运行边奏音乐, 两者互不干涉。该功能在一般 BASIC 使用手册中是没有说明的。BASIC 语言中专门的音乐陷阱语句为:

ON PLAY(n) GOSUB <line>

开启 PLAY ON 暂停 PLAY STOP

取消 PLAY OFF 其中 n 为缓冲器音符个数 1 ~ 32。

在 § 1.1 节中有一个音乐程序《爱的罗曼斯》, 虽在运行时, 不影响程序继续执行, 但它是通过巧用 PLAY(n) 函数设计的。如程序不在 60 句无条件转向 30 句, 那么音乐将会中断。要使程序真能边运行边演奏音乐, 必须有音乐陷阱语句。程序 3 需作如下修改:

```
10 REM 由程序 3 修改而得
20 CLS:N=1:A=1:PLAY "MB L64 B"
30 ON PLAY(1) GOSUB 70 '陷阱语句
40 PLAY ON
50 LOCATE 6,20:PRINT "继续执行...";A;
60 A=A+1:GOTO 30
70 ON N GOTO 80,90,100,110,120,130,140,150,160,170,180,190,200,210,220,230,240,250,260
80 PLAY "MB L9 B<BG>B<BG>B<BG>B<BG>A<BG":N=2:RETURN
90 PLAY "MB >G<BG>G<BG>F<BG>E<BG>E<BG":N=3:RETURN
100 PLAY "MB >G<BG>B<BG>>E<<BG>>E<<BG>>E<<BG":N=4:RETURN
110 PLAY "MB >>E<<BG>>D<<BG>>C<<BG>>C<EC BEC":N=5:RETURN
120 PLAY "MB AEC AEC BEC >C<EC BF+D+":N=6:RETURN
130 PLAY "MB >C<F+D+ BF+D+>D+<F+D+>C<F+D+ BF+D+":N=7:RETURN
140 PLAY "MB B<BG>A<BG>G<BG>G<BG>F<BG":N=8:RETURN
150 PLAY "MB >E<BG>F<BA>F<BA>F<BA>F<BA":N=9:RETURN
160 PLAY "MB >G<BA>F<BA>E<BG>E<BG>E<BG. L18 <EB>EGB >E...":N=10:RETURN
170 PLAY "MB L18 G+<BG>G+<BG>G+<BG>G+<BG>F+<BG":N=11:RETURN
```