

软件设计 技法

The background of the cover is a vibrant, futuristic digital scene. A rocket is shown launching upwards, leaving a massive, bright orange and yellow plume of fire and smoke at its base. The rocket itself is white with orange and black details. The sky is a deep blue, filled with numerous binary digits (0s and 1s) that appear to be floating or falling. Some of these digits are larger and more prominent than others. In the upper right corner, there is a bright, starburst-like light source that radiates outwards, creating a sense of energy and innovation. The overall composition suggests a theme of technological advancement and digital design.

李铁民 杨 静 张素华 著

中国林业出版社

软件设计技法

李铁民 杨静 张素华 著

中国林业出版社

图书在版编目 (CIP) 数据

软件设计技法/李铁民, 杨静, 张素华著. —北京: 中国林业出版社, 2000. 6

ISBN 7-5038-2579-0

I. 软… I. ①李…②杨…③张… II. 程序语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2000) 第 28133 号

中国林业出版社出版

(100009 北京西城区刘海胡同 7 号)

北京地质印刷厂印刷

2000 年 6 月第 1 版 2000 年 6 月第 1 次印刷

开本: 850mm×1168mm 1/32 印张: 9.25

字数: 210 千字 印数: 500 册

定价: 20.80 元

前 言

第一章通过几个 BASIC 和 FOXBASE 程序的设计过程,引导读者克服“万事开头难”的畏惧心理,“迈开编程第一步”。软件设计领域纵然如大海般辽阔、深邃,毕竟也是由一滴滴水组成的,并不是那么神秘,那么遥远,那么不可捉摸的事情。

第二章介绍了几个有一定实用价值的小型程序,并作了详细的分析说明。通过本章的讲解和练习,你会进一步提高学习的兴趣和程序设计的能力。

本书第三章以后是作者多年来编制的若干应用软件及其分析,通过这些软件的分析研究,读者可以了解更多的关于 BASIC、FOXBASE、PROLOG 等各种计算机语言的编程方法,实际应用软件的设计方法。软件设计首先是对整个软件的功能、采用的语言、软件的大致结构框架、数据的结构、风格等等的设计。所以本书在介绍各个软件设计技法的时候,比较详尽地介绍了软件的功能设计、数据结构的设计、使用方法等等,以便读者对照分析。

有别于一般的计算机书籍全面介绍所有细节的做法,本书是从实用途径出发,引导初学者自学编程的,目的是引导初学者在缺乏全面的计算机知识,面对众多“拦路虎”的情况下,先瞄准一个目标,解决一个实际问题。至于在取得初步的胜利后,再逐步向纵深发展,取得更大的成绩,那就是读者自己的事了。那时你尽可以查阅需要的任何资料细节,举一反三,触类旁通,逐步提高。

编著者

2000 年 5 月

目 录

前言

第一章	迈开编程第一步.....	(1)
第二章	几个实用小型程序的设计	(15)
第三章	林业样地调查数据处理软件	(66)
第四章	科技项目与成果管理系统的设计.....	(109)
第五章	通用检索表查询系统设计.....	(188)
第六章	一个灵活实用的月工资管理软件.....	(199)

主要参考文献

第一章 迈开编程第一步

1 学会编制简单的 BASICA 程序

BASIC 是为初学者设计的一种程序设计语言，BASICA 是它的高级版本。一般的 DOS 系统都配备有 BASICA 语言。使用时，在 DOS 提示符（一般是“C>”）下键入 BASICA，屏幕上即显示出 BASICA 的提示符“OK”，表示 BASICA 已作好准备接受你的指令了。我们现在就来编制一个最简单的程序。每一个程序都有它的目标，简单程序亦不例外。我们这个简单程序的目标是：在紧接着程序执行命令的第一个显示行显示一个星号“*”，在以下两行里分别显示两个和三个星号。请跟我输入以下程序：

```
10 PRINT " * "  
20 PRINT " * * "  
30 PRINT " * * * "  
40 END
```

输入完成后，接着按 F2 功能键或 RUN，屏幕上便显示出我们预想的结果来。现在这个程序就保留在计算机的内存里了，你只要一按 F2 键，该程序就被执行一次。但是当你要编写新的程序时或者是关机时，程序就丢失了。为了保存这个程序，请键入：SAVE" P1 ，“P1”是我们为这个程序起的名字，计算机就把该程序以 P1 为文件名保存在磁盘上。保存文件时 BASICA 还自动加上一个扩展名“.BAS”，因而实际

的文件名是“P1. BAS”。你可以不去理会这个扩展名，以后如要使用这个程序，键入 LOAD" P1 即可调入。

以上是编制一个 BASICA 程序的全过程。至此，你已经学会编制一个最简单的程序了。实际上，再复杂的程序也是由一个一个像这样很简单的程序片段构成的。只要循序渐进，不断努力，一定能逐渐掌握更多的程序设计技法，实现独立设计较复杂的应用软件的理想。

现在我们来分析一下这个程序的结构。

在我们这个程序中，每一行都是以一个数字开头的，这是 BASICA 要求的行号。规定的行号最小为 0（一般不用 0 而用 1 或由 10 开始），最大是 65529。行号用来规定 BASICA 程序执行的顺序，除非遇到转移语句，程序就是按照行号从小到大的顺序被执行的。编程时，一般以 10, 20, 30, ……为行号，如果需要在第一行与第二行之间增加一条语句，则可以 15 为行号。如果原程序的行号为 1, 2, 3, ……那么，要达到在第一行后边增加一程序的目的，就得先将原来的“2”行改为“3”、“3”改为“4”，……最后将新增加的一行作为第二行。可见行号之间预留间隔完全是为了修改程序时的方便。

在我们这个简单程序中，前 3 行都含有一个“PRINT”，这是 BASICA 的一个“语句”名，作用是在屏幕上显示信息。本程序中是显示其后双引号中的字符。“PRINT”是英语“印”的意思。BASICA 的语句名都是英语字或英语缩写，其字意反映了该语句的作用，便于记忆。语句名可以写成大写、小写或大小写混合的方式都是可以的，但是为了阅读方便，编程者可根据自己的习惯选择大写或是小写，在一个程序中最好使用统一的格式。BASICA 中“?”是与 PRINT 语句名等

同的，所以本程序中的 PRINT 都可以用“?”来代替。另外输入 PRINT 语句名还有一种简便方法，即在按住 Alt 键的同时按下“P”字母键，便得到语句名 PRINT。

“40”行使用了另一个语句“END”，其英语意义为结束，这里亦表示结束程序执行，回到调用状态。虽然在本例中即使没有该语句程序也能照常运行。但作为一个良好的编程习惯，每一个程序逻辑上的结束处都应写上这个语句。

现在介绍一下输入、修改、运行、存储、调用 BASICA 程序的一般方法。

BASICA 程序的输入：

①在刚进入 BASICA 编辑状态的情况下，直接输入新程序即可。

②如果已编辑或调用过别的程序，再需输入新的程序时，要使用 NEW 命令。该命令清除内存中保留的程序内容，供用户输入新的程序。输入新程序时并不需要为该程序命名，BASICA 只在内存中保留一个唯一的程序，修改和运行都是针对这一程序的。

③可使用 AUTO 命令由 BASICA 自动提供由 10 开始且间隔为 10 的规范的行号序列，开始号和间隔亦可通过在 AUTO 命令后附加参数的方法由程序员确定。

④在使用自动行号功能输入的时候，不要移动光标去修改别的程序行，那样的修改会导致错误。

⑤程序输入结束后或发现已输入行中的错误欲修改时需按下 Ctrl+C（即按住 Ctrl 键的同时按下 C 键）退出。

⑥输入一行程序后按回车键，该行程序才被记录。这时光标移至下一行起始处，在使用自动行号时在下一行开始处显示新的行号，表示这一行程序已经正确输入。

⑦以屏幕上显示的程序行为模本输入新的程序行。例如在输入本程序时，当输完第一行时，光标移至下一行左端。这时可用光标键移动光标至上一行，将“10”行的“1”改成“2”，再移至星号前，加入一个星号后按回车，这样就完成了第二行的输入。原来输入的“10”行程序并未改变，只是作为输入第“20”行的一个模本，简化了输入。在语句较复杂且存在较相似模本的情况下此法很有用。显然，这是在未使用自动行号的情况下进行的。

在输入屏幕上修改程序的方法：

①首先要将所输入的程序显示出来，这要使用 LIST 命令。对于较长的程序，还可指明显示的起止行。如 LIST 100—150，只显示 100—150 之间的程序行。如果已明确要修改的程序行，即可移动光标至该行进行修改，要修改的程序行不在屏幕上时，可使用 EDIT 命令调出。例如 EDIT 50，即可显示出该程序且光标停留在行首位置，便于修改。

②插入一程序行。插入一程序行的方法是将新插入的程序行确定一个正确的行号，该行号的数值界于欲插入的前后两程序行的行号之间。当输入该程序后再使用 LIST 命令，可看到它已处在应该插入的位置了。

③删除程序行。删除一程序行，只需输入该程序行的行号再按回车键即可。而 DELETE 命令则可删除一行或多程序行。如 DELETE 120，删除行号为 120 的程序行，DELETE 70—130 则删除掉行号为 70 和 130 以及两行之间的所有程序。

④插入或删除字符：移动光标至拟插入位置的下一个字符，按 Ins 键，再输入插入的内容，一个或多个字符不限。删除字符时，移动光标至所要删除的字符上，按 Del 键。删除相

连的几个字符，则连续按该删除键即可。另有一个退格删除键，用来删除光标位置前的字符。

⑤当需要增加的程序行过多而必须改变其后的程序行号时，有一条 RENUM 命令能很容易地把内存中现有程序的行号全部改成以 10 开始且间隔为 10 的行号序列。

储存程序：SAVE 命令用于将内存中的程序储存在磁盘上。SAVE 命令至少需要一个参数，即程序的名字。程序名一般以字母开头由 8 个字符组成，例如我们的第一个程序名是 P1，储存这个程序的方法是键入 SAVE" P1。

调用程序：使用 LOAD 命令可将磁盘上储存的程序调入内存执行或修改，例如键入 LOAD" P1 即可调入我们已存入磁盘的那个程序。

运行程序：使用 RUN 命令。

下面我们再给出若干简单的 BASICA 程序，并简要说明该程序的设计目标、主要语句或新出现的语句和技术要点。初学者可逐一输入并运行这些程序，并根据提示对程序作一些尝试性的修改。

程序 P2

```
10 CLS
20 LOCATE 1, 1
30 PRINT " 1"
40 LOCATE 1, 80
50 PRINT " 2"
60 LOCATE 24, 1
70 PRINT " 3";
80 LOCATE 24, 79
```

```
90 PRINT " 4";  
100 LOCATE 12, 39  
110 PRINT " 0"  
120 END
```

设计目标：在屏幕左上角、右上角、左下角、右下角及中央位置分别显示数字 1、2、3、4 和 0。

要点：“10”行用来清除屏幕。语句 CLS 是英语 “Clean screen” 的缩写，清除屏幕的意思。LOCATE 是“定位”的意思，用来指定光标的位置，紧随其后的 PRINT 语句就由该位置开始显示。LOCATE 语句要求两个参数即屏幕的横竖坐标。BASICA 中文本状态下屏幕的坐标是这样定义的：屏幕左端的横坐标为 1，最右端为 80，即每一行 80 个显示位置。每一屏显示 25 行，竖坐标由上而下依次为 1, 2, …, 25。其中第 25 行用作 BASICA 的提示行，所以实际可使用的只有 24 行。故左上角的坐标是 1, 1，右上角是 1, 80，左下角是 24, 1，右下角是 24, 80。但是“80”行中使用的参数是 24, 79。这是因为 24, 80 是屏幕的最后一个显示位置，程序执行完这个位置的显示后要向上“滚屏”，而无法看到第一行的显示。“70”“90”行末尾的分号也是为了防止滚屏的，不可去掉。

练习：读者可修改各个 LOCATE 语句的参数及 PRINT 语句后面引号中的字符，然后运行程序，看看结果是否与你预期的目标相符。另外还可以键入 10 并回车去掉 10 行，看看运行后的结果，理解 CLS 语句的作用。

程序 P3

```
1 REM P3  
5 CLS
```

```
10 FOR I=1 TO 7
20 COLOR I
30 FOR J=1 TO I
40 PRINT STR$ (J);
50 NEXT J
60 PRINT
70 NEXT I
75 COLOR 7
80 END
```

目标：从第一行至第七行分别用 7 种颜色显示与行数数目相同的自然数序列

要点：FOR—NEXT 循环语句，位于 FOR 和 NEXT 之间的语句将被连续执行若干次，FOR 语句指定循环变量并确定循环次数。每循环一次，循环变量自动+1。本例中“10”行至“70”行是一个循环，我们以其循环变量 I 命名之为 I 循环，I 循环内的语句作为一个整体要被执行 7 次，每次循环对应于我们要显示数字的一行，即第一次循环显示第一行，第二次显示第二行……同样 J 循环对应于一行内所要显示自然数的个数，每次循环显一个自然数。“40”行使用了一个函数 STR\$，将循环变量的数值变成一个相应的字符。因为 PRINT 语句在显示数值时要在每个数值前加一个前导空格，变成字符型是为了显示的比较紧凑。该行末尾的分号指示 PRINT 将下一个显示内容在其后间隔一个空格的位置输出。“60”行是一个不带参数的 PRINT 语句指示 BASICA 换行，因为每经过一个 J 循环，就完成了一行的显示，必须换行。COLOR 语句是定义屏幕显示颜色的，每一行使用的色号与该行数相同，即第一行是色号 1，第二行是色号 2，等等。BASICA 默

认的色号是 7，所以在程序中如果使用了改变显示颜色的命令，在结束时应使用 COLOR 7 语句以恢复标准显示。不过本例中最后使用的颜色亦为 7 号色，故此行亦可略去。

练习：去掉“60”行，并运行程序，察看其结果。

去掉“40”行中的 STR\$ 函数，改为 PRINT J，看看运行后显示的结果。

删除“75”行，观察是否对程序结束后的正常显示有影响。

程序 P4

```
10 REM P4
20 CLS
30 FOR I=1 TO 7
40 COLOR I
50 FOR J=1 TO 72
60 LOCATE I, J
70 PRINT " PROGRAM 4"
80 FOR K=1 TO 100 : NEXT K
90 LOCATE I, J: PRINT " ";
100 NEXT J
110 PRINT "          ";
120 NEXT I
130 COLOR 7
140 END
```

目标：移动显示字符“PROGRAM 4”，且每行变换一种颜色。

要点：利用 LOCATE 语句和 I, J 循环，不断改变字符的显示位置，同时清除前一次的显示字符，从而达到“移动”显示

的效果。“80”行利用空循环 K 使显示的字符保留一定时间。“90”行用于抹掉最近一次显示内容的第一个字符。因为第二个字符及其以后部分在下次显示时就被新的显示内容所覆盖而自然消除了，这样就达到了清除旧的显示的目的。

练习：将“80”行中的参数 100 变成 1000，2000 或其它数值看看运行的效果。“110”行是为了在每行的最后一次显示结束后能清除整个字符串，试验一下去掉该行将会发生什么事情。

程序 P5

```
10 REM    P5
```

20 A \$ = " 本程序是用来模拟电视广告中“飘字幕”的方式显示字符串，欲了解该技术的细节请仔细阅读本程序，或直接与程序设计者联系。联系地址：山西省太原市新建南路 185 号，联系电话：0351—7222383"

```
30 CLS
```

```
40 FOR I=1 TO 460
```

```
50 A $ = RIGHT $ (A $ , LEN (A $ ) - 2) + LEFT $ (A $ , 2)
```

```
60 LOCATE 23, 1 : PRINT LEFT $ (A $ , 80);
```

```
70 FOR W=1 TO 900: NEXT W
```

```
80 NEXT I
```

```
90 END
```

目标：模拟“飘字幕”方式显示中文字符串。

要点：显示位置不变、显示的字符串长度不变，但是每次显示的内容不断变化。通过几个字符串操作函数，每经一次循环，把整个字符串的第一个字（2 个字符长度）移到字符串的

末尾组成一个新的字符串，“60”行在一个固定位置显示该字符串的前 80 个字符。由于是在同一位置连续显示，新的显示自然覆盖了旧的内容，故本程序中并不需要清除屏幕的语句。“20”行定义的显示文本汉字字符串包括其后部的空格都必须以中文的“全角”方式输入，否则容易出现显示错误。该字符串末尾的空格是为了下一个字符串的开头不至于紧接住上一个字符串的末尾，使每一次显示之间有一个间歇。“50”行是本程序的核心部分。LEN 函数用来取得 A\$ 字符串的长度（以字符数表示）。RIGHT\$ 取右子字符串函数由 A\$ 中截取除第一个字（两个字符）外的所有字符，其长度是原字符串全长-2，取左子字符串函数 LEFT\$ 则取出第一个字（两个字符），然后加在用 RIGHT\$ 函数所取子字符串的末尾。如此不断循环实现了字符串的动态变化。

练习：①调整“70”行中的空循环参数（程序中为 900），使之适合你的系统和阅读要求。②“40”行中的参数 460 是这样计算的：A\$ 字符串长度为 200 个字符，每个循环移动两个字符，所以每“飘”一次需要 100 个循环，程序按动态显示 5 遍来设计，共需要 500 个循环。但最后一次显示时最末尾的 80 个字符不必“移出”屏幕，故循环次数应减少 40，即总循环次数应为 $100 \times 5 - 40 = 460$ 。试调整该参数使显示次数为 3 次。③增加“75”行：

LOCATE 24, 37; PRINT TIME\$ 使飘字幕的同时在屏幕下方正中显示时间。

2 学会编制简单的 FOXBASE 程序

进入 FOXBASE 的方法是在 DOS 提示符下键入 MFOXPLUS。FOXBASE 的提示符是一个圆点“.”。编制一

个 FOXBASE 程序时必须先为它起个名字,例如我们要编的程序是 f1,这时可键入 MODI COMM f1,即可得到 FOXBASE 的编辑屏幕。在该编辑屏幕输入和修改程序都是很方便的,几乎与一般的文字处理软件一样。输入一行后可按回车键继续下一行的输入亦可按光标键移至别的程序行。所输入的内容都被接受了,而不是像 BASICA 中那样,输入或修改一程序后需按回车键才有效。程序输入结束后,不能马上执行,而是要先按 Ctrl+W 键返回 FOXBASE 的圆点状态,再使用 DO 命令执行。执行 f1 程序的命令是 DO f1。

请依次输入以下几个简单的 FOXBASE 程序,每次只能输入一个,输完一个按 Ctrl+W 返回后,再用 MODI COMM 命令输入下一个。最后使用 DO 命令逐一执行它们。你会发现,这些程序与 BASICA 程序 P1—P5 的运行结果完全一样。

程序 F1

```
? " * "
```

```
? " * * "
```

```
? " * * * "
```

程序 f2

```
@0, 0 SAY" 1"
```

```
@0, 79 SAY" 2"
```

```
@23, 0 SAY" 3"
```

```
@23, 79 SAY" 4"
```

```
@12, 39 SAY" 5"
```

程序 F3

```
SET TALK OFF
```

```
I=1
```

```
DO WHILE I<=7
```

```

C=STR (I)
SET COLOR TO &C
? " 1"
J=2
DO WHILE J<=I
?? STR (J, 2)
J=J+1
ENDDO
I=I+1
ENDDO
SET COLOR TO 7

```

程序 F4

```

SET TALK OFF
CLEAR
I=1
DO WHILE I<=7
C=STR (I)
SET COLOR TO &C
J=0
DO WHILE J<=71
@I, J SAY" PROGRAM 4"
K=1
DO WHILE K<=100
K=K+1
ENDDO
@I, J SAY" "

```