汉字dBASEⅢ

编程解难

平 金维克 董小国
苏士俊 冒巳锋 编著

科学普及出版社

汉字 dBASEⅢ编程解难

余 平 金维克 董小国 苏士俊 冒巳锋 编著

科 学 普 及 出 版 社

内 容 提 要

本书以专题讨论的方式介绍汉字 dBASEⅢ的编程经验和技巧，涉及 9 个方面：系统模块与菜单设计、屏幕输入输出、打印输出、查询、程序文件、技巧、应用程序、数据库文件的结构和编译 dBASEⅢ。对于较难掌握的宏代换函数，也在“怎样进行随机查询?”和“怎样灵活地打印报表?”等专题中给出了较复杂的实例。书中的程序均在计算机上调试通过。书后附有 dBASEⅢ的运算符、函数、命令、英汉对照的错误及提示信息汇总表。

本书可以为经验不多的读者排忧解难，也能供经验丰富的程序员查阅参考。

汉字 dBASEⅢ编程解难

余 平　金维克　董小国　苏士俊　冒巳锋　编著

*　*　*

责任编辑：茹勇夫

封面设计：金维克

*

科学普及出版社出版(北京海淀区白石桥路 32 号)

新华书店北京发行所发行　各地新华书店经售

北京市燕山联营印刷厂印刷

*

开本 787×1092 毫米　1/16　印张：10.875　字数：252 千字

1991 年10月第 1 版　1991 年10月第 1 次印刷

印数:1-6 000 册　定价：5.80 元

ISBN 7-110-02042-8/TP · 35

前　言

dBASEⅢ是目前微机上流行最广的一种数据库管理系统，特别适合中小型数据库管理工作。自从它引进我国并成功地汉化以来，短短的几年间已经开发出成千上万个应用软件。尤其是财务会计、物资管理、人事档案、图书管理、计划统计等方面，使用的软件大多数都是用 dBASE 开发的。由于 dBASE 深受用户欢迎，一些软件公司又相继推出了兼容产品，如 dBASEⅢ PLUS、编译 dBASEⅢ、dBASEⅣ、FOXBASE 等等。

dBASE 的语句精炼、易学易用，即使是没有学过其它高级语言的人，也能通过自学和实践在较短的时间内掌握简单程序的编写方法。国内仅公开出版的 dBASE 教科书、手册就有一、二十种。然而，dBASE 的强大功能，仅仅为数据库的管理提供了一个良好的工作环境。要想用它编制出可靠性好、速度快、效率高、通用性强的实用软件，还要求编程人员具有丰富的经验和熟悉各种编程技巧。

许多人都有一个共同的体会：在编程过程中常常遇到一些问题，但是在用户手册中找不到解决的方法。处理这类问题要化费很多时间和精力，而最后找到的方法却可能十分简单。我们认为，有必要把这一类方法汇编成书。因此，本书既不是一本教科书，也不是用户手册。也就是说，既不系统地介绍程序设计方法，也不详细讲解每一条命令和函数，而专门探讨编程经验和技巧，针对一个一个专题进行讨论。例如："怎样解决多次重复输入汉字的问题?"、"怎样进行模糊查找?"、"怎样使程序能在具有不同汉字显示行数的机器上运行?"等等。此外，对于较难掌握的宏代换函数，虽然没有作为专题分析，但在"怎样进行随机查询?"和"怎样灵活地打印报表?"等专题中都有较复杂的实例，可以供读者参考。本书主要是我们实际工作经验的汇集，也吸取了他人的一些经验。总之，我们希望本书既能为经验不多的读者排忧解难，也能供经验丰富的程序员查阅参考。

dBASE 的兼容软件比较多，本书以汉字 dBASEⅢ为主，也涉及一些兼容软件，主要是编译 dBASEⅢ。全书共 122 个小题，各个题目的内容基本独立。书中没有划分章节，只是把相关的题目放在一起以便读者查找，大致分为 9 类：系统模块与菜单设计、屏幕输入输出、打印输出、查询、程序文件、技巧、应用程序、数据库文件的结构和编译 dBASEⅢ。书中的程序均在计算机上调试通过。为了便于读者使用，书后附有 dBASEⅢ的运算符、函数、命令、英汉对照的错误及提示信息汇总表。由于 dBASE 的应用日益广泛，广大程序员创造的新思路、新方法层出不穷，本书提供的经验仅仅是九牛一毛而已。书中的方法不一定是最好的方案，但都是可行的和实用的方法。本书难免有缺点和错误，我们恳切地希望读者批评指正。

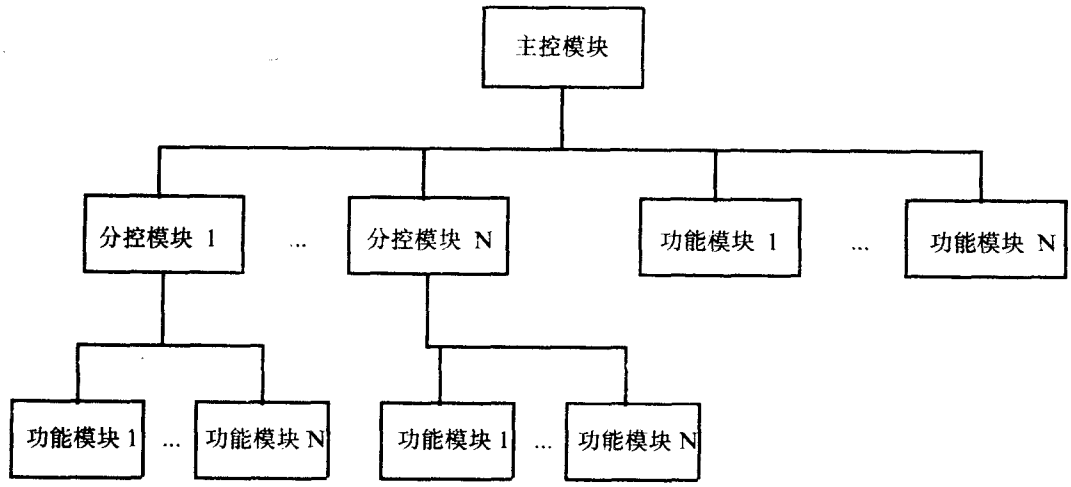
本书由余平、金维克、董小国、苏士俊和冒已锋等在共同研究的基础上分头编写，最后由余平和金维克对全书进行统稿。在统稿时没有追求统一的风格，以便保留每个撰写人各自的特色。如果两个人讨论同一个专题，只要给出的解决方法不同，统稿时也未进行合并，而是让读者选择自己喜爱的解法。

1991.1.31

目　　录

1. 如何设计数据库系统的系统结构?

根据结构化程序设计的原则，一个数据库系统应该由若干个程序模块组成。各种模块按工作性质划分，大致可以分成两类。一类是主控模块和分控模块，它们在程序中并不完成具体的操作，而用于控制各功能模块之间的调用。另一类是功能模块，通常一个功能模块完成一项具体的操作，比如查询记录，修改、删除记录，打印报表等等，它们在控制模块的调度下，根据用户的需要，实现对数据库的各种操作。系统结构大致如下图所示:



注:

(1) 每个模块都是一个子程序，用 RETURN 结束（主控模块也可用 QUIT 结束)。

(2) 根据用户的功能要求，用 DO ＜子程序名＞调用各个子程序。

(3) 编程可以使用 MODIFY COMMAND 命令或 WORDSTAR 等字处理软件。

(苏士俊)

2. 如何设计数据库系统的主控模块?

主控模块是数据库系统的控制中心，一般由 4 部分组成:

第 1 部分是系统各参数的设置，可称为系统初始化。

第 2 部分是用户使用权限的审查，通常采取口令的形式。即: 请用户输入口令，程序判断口令是否正确，以便防止非法用户对数据库进行操作（口令也叫保密字)。

第 3 部分为主菜单的功能选择，请用户根据需要选择相应的序号。

第 4 部分是分支控制，例如使用 CASE 语句，根据用户的选择调用不同的功能模块。

因此主控模块的程序结构组成如下:

```
{ 系统的初始化 }
{ 核定使用权限 }
DO WHILE ＜条件＞
   CLEAR
   { 显示主菜单 }
   { 根据主菜单进行选择 }
```

```
    { 根据选择执行相应的功能模块 }
  ENDDO
  CLEAR ALL
  RETURN
```

关于如何审定用户使用权限的问题请参见"76. 如何实现口令的保密?"。

菜单的设计方法很多，本书中也提供了几种，这里就不介绍了。下面只介绍如何用条件控制循环以及如何根据用户的选择调用不同的程序模块。

```
DO WHILE .T.
  { 显示主菜单 }
  { 请选功能序号. 即由用户输入一个数字给变量 C }
  DO CASE
    CASE C=0
        EXIT
    CASE C=1
        DO <过程 1>
    CASE C=2
        DO <过程 2>
    ......
  ENDCASE
ENDDO
CLEAR ALL
RETURN
```

程序模块在其它高级语言中叫子程序，dBASE 中叫过程。参见"64. 过程、过程文件和命令文件"。

(苏士俊)

3. 如何设计数据库系统的输入模块?

在关系型数据库中，增加数据是以"记录"为单位进行的。为了输入多条记录，应该使用循环语句，每次循环输入一条记录。

对数据库中的记录进行输入和编辑时，dBASEⅢ本身提供一种普通的全屏幕输入格式，不用编程，每个字段名和相应的数据占用一行。但用户常常希望屏幕格式更美观。因此，一般采用格式化输入方法，即由@...GET 语句和 READ 语句配合输入数据。这种方法可以在指定的位置上显示汉字提示，并在指定的位置上接收相应的数据，还能对数据的格式和范围加以限制。输入模块的程序结构组成如下:

```
{ 打开数据库文件 }
{ 某些内存变量赋初值 }
DO WHILE <条件>
  APPEND BLANK
  @ 行号,列号 SAY "<提示信息>" GET 字段名 / 变量名
```

```
.....
@ 行号,列号 SAY "<提示信息>" GET 字段名/变量名
READ
REPLACE...WITH...
{ 接受新的循环控制信息 }
ENDDO
{ 关闭数据库 }
RETURN
```

几点说明:

(1) @语句中的变量要求事先有初值，因此 APPEND BLANK 是必须的，这样可使各字段变量的初值为空或为 0。READ 语句也是必须的，它是@...GET 语句或格式文件的激活语句。

(2) 由若干个@语句组成的格式描述，可以象上面那样直接写在输入模块中，也可以把它们以格式文件的形式存入磁盘，在输入模块中打开格式文件后再调用它。

(3) REPLACE 语句用来计算有依赖关系的字段值，或者将 GET 得到的变量值（常为简单字符）转换为相应的字段值（常为较复杂的字符串或汉字）。

例 1: REPLACE 实发工资 WITH 基本工资+工龄工资+副食补贴

例 2: 将 GET 得到的变量 X 的值（"1"和"2"）转为性别

```
IF X="1"
   REPLACE 性别 WITH "男"
ELSE
   REPLACE 性别 WITH "女"
ENDIF
```

(4) 程序还应该有拒绝接受错误数据的能力，请参见"27. 数据输入时如何进行数据有效性检验和正确性判断?"

(5) 所谓接收新的循环控制信息，就是每输入一条记录之后，都要给循环控制条件一个新值，以决定是否结束输入工作。例如:

```
S=" "
@ 行号,列号 SAY "继续增加记录吗 (Y/N) ? " GET S
READ
```

这时 DO WHILE 语句中的条件已具体化为: DO WHILE UPPER(S)="Y"。注意，在循环语句之前一定要有 S="Y"语句，否则无法进入循环。

(苏士俊)

4. 如何用 TEXT...ENDTEXT 语句编制屏幕菜单?

程序设计中经常要编制屏幕菜单。最简单的菜单是将各个功能模块的功能集中显示于屏幕上，并冠以相应的选择序号。当用户从键盘按序号选择后，程序进入相应的模块进行处理。处理之后又回到这一级菜单。

可以用 TEXT...ENDTEXT 语句编制菜单: 直接将菜单放在 TEXT 与 ENDTEXT 之

间．程序运行时，TEXT 至 ENDTEXT 间的全部内容将原封不动地显示在屏幕上。在 ENDTEXT 语句之后，再使用输入语句进行相应的屏幕选择。然后用 DO CASE 进入各自的分支程序。

下面的程序输出一个汇总用的菜单，其中涉及 3 个功能模块，分别由 3 个过程完成。

```
DO WHILE .T.
  CLOSE DATABASE
  STORE 0 TO CHOOSE
  CLEAR
  TEXT
    ————————— 输 出 汇 总 菜 单 —————————
    | 1. 中国北京第一 XXXXXX 厂科协统计调查综合汇总表 |
    | 2. 中国北京第二 XXXXXX 厂科协统计调查综合汇总表 |
    | 3. 中国北京第三 XXXXXX 厂科协统计调查综合汇总表 |
    | 0. 退出                                        |
    ————————— 请选择功能号: —————————
  ENDTEXT
  @ 6,39 GET CHOOSE PICTURE "9" RANGE 0,3
  READ
  IF CHOOSE=0
    EXIT
  ENDIF
  DO CASE
    CASE CHOOSE=1
      DO MSHZ1
    CASE CHOOSE=2
      DO MSHZ2
    CASE CHOOSE=3
      DO MSHZ3
  ENDCASE
ENDDO
RETURN
```

（冒已锋）

5. 用 TEXT...ENDTEXT 语句显示菜单时，如何确定菜单的位置？

TEXT 语句与 ENDTEXT 语句之间的菜单或一段文字可以直接显示在屏幕上，显示的位置与当前的光标位置有关。因此，盲目使用 TEXT 语句，好象菜单位置是随机的。

为解决这个问题，可以在 TEXT 语句之前，先用@语句确定光标的位置（只对行有效，对列无效），然后再将菜单显示在屏幕上。程序如下：

```
CLEAR
```

```
@ 行号，列号
TEXT
  { 菜单内容 }
ENDTEXT
```

(苏士俊)

6. 如何利用菜单技术减少汉字输入?

为了避免直接输入汉字，可以借助于菜单技术：把某些常用的汉字词组，按其类别制成菜单并配以相应的数字代码，当需要输入汉字时，用户只需输入相应的词组代码，由程序自动地将代码替换成汉字。这样不仅提高了输入速度，而且可以降低出错率。

比如在输入职工档案时，将各部门的汉字名称以菜单形式显示在屏幕上，请用户选择相应的代码，然后用 REPLACE 语句完成替换工作：

```
@ 行号,列号
TEXT
    1. 厂长办公室
    2. 设计科
    3. 工具科
    4. 工艺科
ENDTEXT
WAIT "请选择相应的编码:" TO 部门
DO CASE
   CASE 部门 ="1"
        REPLACE 部门 WITH "厂长办公室"
   CASE 部门 = "2"
        REPLACE 部门 WITH "设计科"
   .....
ENDCASE
```

(苏士俊)

7. 如何动态生成菜单，使系统具有记忆功能?

菜单可以减少汉字输入。比如在录入学生档案时，把民族的汉字名称和编码写成菜单，就能通过输入编码来输入民族。参见"6. 如何利用菜单技术减少汉字输入?"

但是，简单的菜单有时满足不了用户的要求。例如对于某大学的学生档案程序，如果要在菜单中列出世界上所有的民族，不仅增加了选择的困难，也影响屏幕美观，而实际上许多民族可能根本用不上。因此需要使用**动态菜单**生成技术。

技术问题是这样解决的：专门设置一个民族编码库，用它来支持系统运行，库内只存放民族名称和对应的编码。在需要输入民族时，打开这个数据库，将其内容按一定格式显示在屏幕上，并询问有无需要的民族。如果屏幕上有所需的民族，可以输入相应的编码；如果没有所需要的民族，只需键入 N，即可转去向民族编码库中增加一条新记录（新的民族名称和对应的编码）。该民族的名称和编码一旦存入数据库，下次再显示菜单时，新民族和编码

会永远出现在菜单中。这种有"记忆"功能的菜单会极大地方便用户。

下面是完成该功能的一个示范程序，凡是用户使用过的民族都自动出现在菜单中。菜单内容会随系统运行而动态变化。为简化问题，本例中采用记录号作为民族的编码

存放民族名称的 MZTSK.DBF 库已有 7 条记录，其库结构和数据如下:

```
数据库结构 — 数据库    : C:MZTSK.dbf
数据库中的数据记录个数 :          7
数据库的最后更新日期   : 01/23/91
 字段  字段名       类型        宽度     小数
    1  民族名称     字符型         4
** 总计 **                       5
Record#  民族名称
       1  汉族
       2  回族
       3  苗族
       4  壮族
       5  满族
       6  白族
       7  傣族
```

动态生成菜单的程序如下:

```
SET TALK OFF
RK=" "
Y12=" "
DO WHILE .T.
  CLEAR
  @ 3,8 SAY "**民族提示**"
  USE MZTSK
  SET HEADING OFF
  LIST
  SET HEADING ON
  IF RK="Y"
    EXIT
  ENDIF
  @ 1,3 SAY "如果没有需要的民族，"
  @ 1,23 SAY "请按 N 键，否则按其它键! " GET Y12  PICTURE "! "
  READ
  *增加新民族:
  IF Y12 <> "N"
    @ 1,0 SAY SPACE(80)
    EXIT
```

```
      ELSE
        APPEND BLANK
        RK="N"
        DO WHILE RK <>"Y"
          @ 2,10 SAY "新民族: " GET 民族
          READ
          @ 2,30 SAY "认可吗(Y/N)?" GET RK  PICTURE "! "
          READ
          @ 2,0 SAY SPACE(80)
        ENDDO
      ENDIF
    *  认可后会返回循环体重新显示菜单
    ENDDO
    N=0
    @ 1,8 SAY "请选择民族编码:  " GET N  PICTURE "99"  RANGE 1,RECNO( )-1
    READ
    @ 1,40 SAY "以下操作略 ..."
    USE
    SET TALK ON
    RETURN
```

注:

(1) 下面是用户操作时显示的一个屏幕画面:

```
如果没有需要的民族，请按 N 键，否则按其它键!
      * * 民族提示 * *
        1 汉族
        2 回族
        3 苗族
        4 壮族
        5 满族
        6 白族
        7 傣族
```

(2) 为了同时显示几十个民族并使屏幕美观，可直接用下面的程序段代替上述程序中的 LIST 语句，这样每行将显示 5 个民族:

```
    @ 3,8 SAY "* * * * * * * * * *  民 族 提 示  * * * * * * * * * *"
    DO WHILE .NOT.EOF( )
      K=1
      STRING =""
      DO WHILE .NOT.EOF( ) .AND. (K<=5)
```

```
        STRING = STRING + STR( RECNO( ),3,0 ) + ″: ″ + 民族 +″ ″
        K=K+1
        SKIP
      ENDDO
      @ ROW( )+1,8 SAY STRING
    ENDDO
```

(3) 下面是程序自动生成的一个动态菜单，最初含 7 个民族，现在含 8 个民族：

```
* * * * * * * * * * 民 族 提 示 * * * * * * * * * *
 1: 汉族   2: 回族   3: 苗族   4: 壮族   5: 满族
 6: 白族   7: 傣族   8: 布依
```

(苏士俊)

8. 怎样自动建立菜单程序?

在任何管理系统中，菜单是经常使用的，有时一个管理系统中要用到几十个菜单。编写菜单程序虽然不复杂，但每个菜单程序都要反复调试，很是浪费时间。

下面我们看一个简单的自动建立菜单程序，它的特点是：

(1) 菜单永远自动居于屏幕中央，左右对称，这符合中国人一般的审美观点。

(2) 菜单的许多项目可以按给定的列数自动排列。

(3) 可以给定菜单的颜色（文字的颜色）。

(4) 有错误提示，并可以反复设计，直到满意为止。

(5) 生成的菜单程序可以方便地调用或直接运行。

自动建立菜单程序 CD.PRG 如下：

```
SET TALK OFF
SET SAFETY OFF
SET FUNCTION 9 ″MODI COMM CD″
SET FUNCTION 10 ″DO CD;″
H=″————————————————————————————————————————————————————————————″
DO WHILE .T.
  CLEAR
  @ 4,1
  WAIT ″                    输入、修改菜单题目和项目吗? (Y / N) ″ TO XG
  CLEAR
  IF UPPER(XG)< >″Y″
    EXIT
  ELSE
    DO CD0
  ENDIF
ENDDO
USE CD1
```

```
L=0
DO WHILE .NOT.EOF( )
   K=LEN(TRIM(菜单项目))
   IF K>L
      L=K
   ENDIF
   SKIP
ENDDO
COUNT TO JL
CLEAR
WJM="          "
@ 3,25 SAY "请送菜单文件名" GET WJM
READ
WJM=TRIM(WJM)+".PRG"
DO WHILE .T.
   GO 1
   CLEAR
   LS=1
   ZS="14"
   @ 4,25 SAY "请送菜单列数" GET LS PICTURE "##"
   @ 5,25 SAY "请送菜单色码" GET ZS
   READ
   CLEAR
   IF LS*(L+4)>75
      @ 4,25 SAY "列数过多，请重新给菜单列数! "
      WAIT ""
   ELSE
      DO CD1
      CLEAR
      UI=" "
      @ 4,25 SAY "重新设计菜单吗? (Y/N)" GET UI
      READ
      CLEAR
      IF UPPE(UI)<>"Y"
         EXIT
      ENDIF
   ENDIF
ENDDO
USE
```