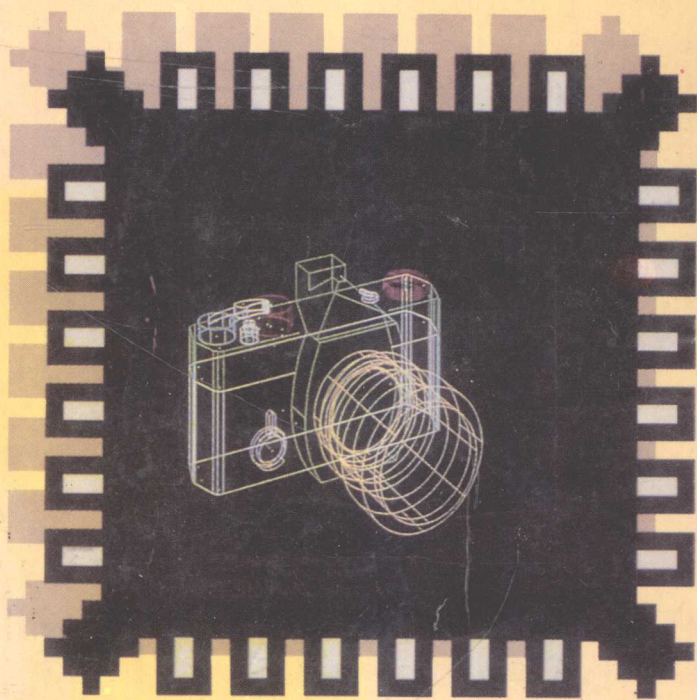


FOXBASE + 通用程序 设计技巧集粹

(2.0 - 2.1)

江代有 编著



《现代电子技术》增刊

陕西电子杂志社

FOXBASE+通用程序设计技巧集粹 (2.0~2.1)

江代有 编著

陕西电子杂志社

陕(新)刊增字第 33 号

FOXBASE+通用程序设计技巧集粹
(2.0~2.1)

江代有 编著

* * * * *

陕西电子杂志社出版

西安东方彩印厂印刷

开本:787×1092 1/16 印张:27.12 字数:651 千字

1995 年 7 月第 1 版 1995 年 7 月第 1 次印刷

印数:1—5000 册

国内统一刊号:CN61—1224/TN

定价:25.80 元

内 容 简 介

针对任意对象的通用程序因其适用面宽,节省开发经费,提高管理水平而大大优于针对特定对象的专用程序。

本书收集了 FOXBASE+ 环境下通用程序设计的诸多宝贵经验, 汇集了行家里手的编程精华。把全书的程序连成一体即可成为一个结构完整, 功能齐全的通用信息管理系统, 每一个程序又可以单独挂接于任意一个专用管理系统下而成其子模块。

作 者

1995 年 6 月

目 录

第一章 通用系统中常用的程序段	(1)
1.1 通用库结构测试程序	(1)
1.2 生成随机选定的字段名表程序	(2)
1.3 任意变量组屏幕显示格式安排程序	(6)
1.4 随机组合的逻辑表达式的生成程序	(7)
第二章 通用系统封面程序设计技巧	(13)
2.1 动态显示封面程序.....	(13)
2.2 特色显示封面程序.....	(15)
2.3 仿 WINDOWS 丰富画面程序	(17)
2.4 屏幕设计技巧.....	(20)
2.5 汉字放大技巧.....	(34)
2.6 竖行显示汉字技巧.....	(37)
第三章 通用总控菜单程序	(40)
3.1 通用下拉式菜单程序.....	(40)
3.2 通用上弹式菜单程序.....	(49)
3.3 通用光亮条菜单程序.....	(52)
3.4 通用传统式菜单程序.....	(55)
第四章 通用初始化模块设计技巧	(58)
4.1 通用自动生成库结构程序.....	(58)
4.2 通用建立数据库结构程序.....	(63)
4.3 通和自动分屏浏览结构程序.....	(65)
4.4 通用修改库结构程序.....	(67)
4.5 通用自动生成屏幕修改格式文件程序.....	(72)
4.6 通用修改格式文件程序.....	(78)
第五章 通用数据维护模块设计技巧	(79)
5.1 通用全屏幕录入程序.....	(79)
5.2 通用数据录入、修改、显示程序.....	(84)
5.3 通用帮助提示录入程序.....	(87)
5.4 通用全屏幕修改程序.....	(91)
5.5 通用携带式录入修改程序.....	(96)
5.6 通用动态修改程序.....	(97)

5.7	通用数据移动程序.....	(99)
5.8	通用快速连接数据库程序	(101)
第六章	通用系统维护模块设计技巧.....	(104)
6.1	限制系统使用时间	(104)
6.2	设置系统口令	(105)
6.3	设置系统口令库的技巧	(109)
6.4	自动生成下拉式菜单程序	(114)
6.5	自动生成弹出式菜单程序	(120)
6.6	快速删除文件名带有一定特征的文件	(124)
6.7	备份大量子目录程序	(126)
第七章	通用查询浏览程序设计技巧.....	(128)
7.1	通用组合查询程序	(128)
7.2	通用模糊查询程序	(132)
7.3	通用全屏幕浏览多记录程序	(134)
7.4	通用浏览超长超宽报表程序	(142)
第八章	通用统计汇总模块设计技巧.....	(145)
8.1	通用统计汇总前的审查纠错程序	(145)
8.2	通用多库统计汇总程序	(148)
8.3	通用统计作图程序	(158)
8.4	通用二维报表统计程序	(165)
第九章	通用报表打印程序.....	(169)
9.1	通用动态制表程序	(169)
9.2	通用打印折行报表程序	(205)
9.3	通用行列转换打印超宽报表程序	(208)
9.4	通用打印实线封闭表格程序	(211)
9.5	通用打印二维报表程序	(214)
9.6	通用打印任意报表程序	(218)
第十章	通用服务功能模块设计技巧.....	(222)
10.1	通用表格数据转换为数据库数据程序.....	(222)
10.2	通用文件数据备份程序.....	(224)
10.3	通用数据库行列转置程序.....	(227)
第十章	通用排序索引模块设计技巧.....	(230)
11.1	通用简单排序程序.....	(230)

11.2	通用复杂排序索引技巧.....	(232)
11.3	通用随机排序程序.....	(235)
第十二章 通用系统中常用的工具程序.....		(238)
12.1	源程序锯齿结构编排工具程序.....	(238)
12.2	源程序结构检查程序.....	(244)
12.3	源程序大小写字母自动转换程序.....	(248)
12.4	阿拉伯数字转换成英文数字程序.....	(250)
12.5	小写金额自动转换为大写金额的程序.....	(253)
12.6	文本文件自动生成左右排版打印的程序.....	(254)
12.7	命令文件的加密与解密程序.....	(260)
12.8	通用跟踪查错程序.....	(263)
12.9	FOXBASE+各程序间相互调用关系的自动分析程序	(265)
12.10	过程文件的自动分解程序	(268)
12.11	显示数据工作区的各种参数程序	(269)
12.12	快速统计系统的命令文件个数、字节数及源程序行数程序.....	(270)
第十三章 其它应用技巧.....		(273)
13.1	系统日历程序.....	(273)
13.2	计算器程序.....	(274)
13.3	调色板程序.....	(277)
13.4	音乐程序.....	(279)
13.5	清屏技巧.....	(285)
13.6	窗口技术.....	(288)
13.7	程序调试技巧.....	(294)
13.8	提高程序运行效率的方法.....	(298)
13.9	容错技巧.....	(300)
13.10	陷阱技术	(305)
13.11	功能键设置技巧	(307)
13.12	在线帮助技巧	(309)
第十四章 仿 PCTOOLS 窗口工具技巧程序		(312)
第十五章 通用系统中常用的函数及命令应用技巧.....		(319)
15.1	宏代换函数(&)的应用技巧	(319)
15.2	INKEY()函数的应用技巧	(321)
15.3	READKEY()函数的应用技巧	(327)
15.4	STUFF()函数的应用技巧	(335)
15.5	BROWSE 命令的使用技巧	(336)

15.6	ON KEY 命令的应用技巧	(337)
15.7	PACK 命令的应用技巧	(340)
15.8	SUBSTR()函数的应用技巧	(341)
15.9	CHR()函数的应用技巧	(345)
15.10	KEY BOARD 命令的应用技巧	(349)
附录一	FOXBASE+命令集	(351)
附录二	FOXBASE+函数集	(368)
附录三	FOXBASE+的配置与调整	(376)
附录四	FOXBASE+光标控制键	(381)
附录五	FOXBASE+出错信息及说明	(390)
附录六	FOXBASE+命令文件的编译和组合	(395)
附录七	ASC II 码表	(396)
附录八	FOXBASE+屏幕设计的应用程序生成器 FOXVIEW	(399)
附录九	FOXBASE+模板方式程序自动生成器 FOXCODE	(406)
附录十	FOXBASE+文档自动生成器 FOXDOC	(411)
附录十一	控制中心 FOXCENTRAL	(418)
附录十二	图形系统 FOXGRAPH	(427)

第一章 通用系统中常用的程序段

1.1 通用库结构测试程序

在通用管理信息系统中,数据的维护、查询、统计等数据处理工作都是离不开数据库的字段名和数据类型的,而不同的数据库又有着不同的字段名和数据类型,针对任意数据库操作的通用系统,只有通过测试被指定的数据库结构,才会取得所需的字段名、字段类型、字段宽度和小数位数。根据具体的需要,常常采用以下两种方法测试被指定的数据库结构。

一、用 COPY 命令测试库结构

在 FOXBASE+ 和 DBASE 中都提供了 COPY TO <文件名>STRUCTURE EXTENDED 命令,这个命令对于测试库文件结构很有用处,通过它可以测得库文件中各个字段名、字段类型、字段宽度和小数位数。下面的源程序清单就是利用该命令生成被指定数据库的结构文件,然后再通过执行循环语句将被测文件的各字段名赋给相应的各内存变量中,以被程序调用。

源程序清单如下:

```
*****用 COPY 命令测试库结构程序*****
SET TALK OFF
SET STAT OFF
SET SCOR OFF
SET MENU OFF
CLEA
SET COLO TO 7/1
FN1=SPAC(10)
FN2=SPAC(10)
@10.25 SAY "请输入欲被测试结构的文件名:" GET FN1
@12.25 SAY "请输入您欲生成的结构文件名:" GET FN2
READ
USE &FN1
COPY TO &FN2 STRU EXTE
USE &FN2
N=11
N1="11"
DO WHILE .NOT. EOF()
    ZDM&N1=FIELD_NAME
```

```

N=N+1
N1=STR(N,2)
SKIP
ENDD
CLOS ALL
RETU

```

二、用 FCOUNT 和 FIELD 函数测试库结构

FOXBASE+提供的 FCOUNT 函数可用于测试被测库文件的字段数，FIELD 函数可用于测试被测库文件的字段名。较之 COPY 命令，这两个函数用起来比较方便，它们无需生成结构文件，但它们只能测得库文件的字段名，而 COPY 命令除能测得字段名外，还可测得相应的字段类型、字段宽度和小数位数。下面给出的源程序清单就是用这两个函数测试库结构的实例。

源程序清单如下：

```

*****用 FCOUNT 和 FIELD 函数测试库结构程序*****
SET TALK OFF
SET STAT OFF
SET SCOR OFF
SET MENU OFF
CLEA
SET COLO TO 7/1
FN=SPAC(10)
@12,25 SAY "请输入欲被测试结构的文件名:" GET FN
READ
USE &FN
ZDS=FCOUNT()
ZD=1
N=11
N1="11"
DO WHILE ZD<=ZDS
    ZDM&N1=FIELD(N-10)
    N=N+1
    N1=STR(N,2)
    ZD=ZD+1
ENDD
CLOS ALL
RETU

```

1.2 生成随机选定的字段名表程序

在管理信息系统中，对数据库及其记录的操作经常要用到字段名。如果在专用程序中，

根据具体的库指定具体的字段名是十分容易的事，在命令语句后面加上条件语句（例如：BROWSE [FIELDS<字段名>] 等）就可以达到预期的目的；但对于通用管理程序来说，库文件的结构是任意的，而选定字段名去组成字段名表也是由用户随机指定的，这对于程序员来说是十分头痛的事。下面给出的程序段就是为解决这一问题而设计的。程序中首先用 FCOUNT 函数测试库结构，并存放在 KXMC&KZ0（款项名称，即字段名）中，同时开辟与款项数相同的内存变量 M&KZ0，用于存放对款项的选择标志或舍弃标志。然后在屏幕上显示这些款项名称，用户用回车键选中，光标键选择，选中的为红色，未选中的呈绿色。

源程序清单如下：

```
* * * * 生成随机选定的字段名表程序 * * * *  
SET TALK OFF  
SET STAT OFF  
SET SCOR OFF  
SET SAFE OFF  
SET ESCA OFF  
CLEA  
SET COLO TO 7/1  
FN=SPAC(10)  
@10.25 SAY "请输入数据库文件名：" GET FN  
READ  
CLEA  
USE &FN  
KXS0=FCOUNT()  
KZ=11  
KZ0="11"  
KXS=0  
DO WHILE KXS<KXS0  
    KXMC&KZ0=FIELD(KZ-10)  
    M&KZ0="X"  
    KZ=KZ+1  
    KZ0=STR(KZ,2)  
    KXS=KXS+1  
ENDD  
KXS=0  
KZ=11  
KZ0="11"  
HZB=10-INT(KXS0/6)  
SET COLO TO 2/0  
DO WHILE KXS<KXS0  
    LZB=15  
    DO WHILE LZB<=65 .AND. KXS<KXS0
```

```

@HZB,LZB SAY KXMC&KZ0
LZB=LZB+20
KXS=KXS+1
KZ=KZ+1
KZ0=STR(KZ,2)
ENDD
HZB=HZB+1
ENDD
HZB=10-INT(KXS0/6)-3
@HZB,0 SAY "回车键选中确认"
@HZB+1,0 SAY "光标键选择"
@HZB,70 SAY " * 结束"+SPAC(5)
DO WHILE .T.
    KXS=0
    JSBZ=.F.
    KZ=11
    KZ0="11"
    HZB=10-INT(KXS0/6)
    DO WHILE KZ<KXS0+11
        LZB=15
        DO WHILE LZB<=65 .AND. KZ<KXS0+11
            IF M&KZ0="X"
                SET COLO TO 2/0
            ELSE
                SET COLO TO 4/0
            ENDI
            @HZB,LZB SAY KXMC&KZ0
            KEY=INKEY(0)
            DO CASE
                CASE KEY=42 .OR. KZ=KXS0+11
                    JSBZ=.T.
                    EXIT
                CASE KEY=24 .OR. KEY=4
                    SET COLO TO 2/0
                    @HZB,LZB SAY KXMC&KZ0
                    M&KZ0="X"
                CASE KEY=13
                    SET COLO TO 4/0
                    @HZB,LZB SAY KXMC&KZ0
                    M&KZ0="O"
            ENDC
            KZ=KZ+1
            KZ0=STR(KZ,2)

```

```

        LZB=LZB+20
    ENDD
    IF JSBZ
        EXIT
    ELSE
        HZB=HZB+1
    ENDI
ENDD
    SET COLO TO 2/0
    SK1=0
    HZB=10-INT(KXS0/6)-3
    DO WHIL SK1<4
        @HZB,70 SAY "OK?" +SPAC(7)
        SK=0
        DO WHIL SK<100
            SK=SK+1
        ENDD
        @HZB,70 SAY SPAC(10)
        SK=0
        DO WHIL SK<100
            SK=SK+1
        ENDD
        SK1=SK1+1
    ENDD
    OK="Y"
    @HZB,70 SAY "OK(Y/N)?" GET OK
    READ
    IF UPPE(OK)="Y"
        EXIT
    ENDI
ENDD
    BLMB=" "
    KZ=11
    KZ0="11"
    KXS=0
    DO WHIL KZ<KXS0+11
        IF M&KZ0="O"
            KXS=KXS+1
            IF KXS=1
                BLMB=BLMB+KXMC&KZ0
            ELSE
                BLMB=BLMB+"."+KXMC&KZ0
            ENDI
        ENDI
    ENDD

```

```

ENDI
KZ=KZ+1
KZ0=STR(KZ,2)
ENDD
GO TOP
IF KXS<>0
  CHAN FIELD &BLMB
ENDI
COLS ALL
SET COLO TO
RETU

```

1.3 任意变量组屏幕显示格式安排程序

在管理信息系统中，经常需要显示或编辑被指定数据库的字段名和字段数据，这就需要将这各字段名或数据以合适的、美观的格式显示在屏幕上。对专用程序而言，字段数和字段名都很具体，易于安排显示格式；但对于通用程序，字段名和字段数都是随机的，让它们在屏幕上以美观的格式显示出来，就要用到 FCOUNT 函数，先测试被指定数据库的字段数，再将各字段名存放于内存变量组中。随后计算出比较合理的显示行列坐标，通过循环语句就可以用美观的格式显示出被指定的库文件各字段名。

源程序清单如下：

```

* * * * * 任意变量组屏幕显示格式安排程序 * * * * *
SET TALK OFF
SET STAT OFF
SET SCOR OFF
SET SAFE OFF
SET ESCA OFF
CLEA
SET COLO TO 7/1
FN=SPAC(10)
@10,25 SAY "请输入数据库文件名：" GET FN
READ
USE &FN
ZDS0=FCOUNT()
ZDS=1
KZ=11
KZ0="11"
DO WHIL ZDS<=ZDS0
  ZDMC&KZ0=FIELD(KZ-10)
  KZ=KZ+1

```

```

KZ0=STR(KZ,2)
ZDS=ZDS+1
ENDD
CLEA
HZB=10-INT(ZDS0/6)
KZ=11
KZ0="11"
DO WHIL KZ<=ZDS+10
    LZB=15
    DO WHIL LZB<=65 .AND. KZ<=ZDS0+10
        @HZB,LZB SAY ZDMC&KZ0
        LZB=LZB+20
        KZ=KZ+1
        KZ0=STR(KZ,2)
    ENDD
    HZB=HZB+1
ENDD
CLOS ALL
SET COLO TO
RETU

```

1.4 随机组合的逻辑表达式的生成程序

在通用管理信息系统中，对任意指定的数据库进行查询、修改、打印、统计等操作中，需要生成各种随机组合的逻辑表达式。如果在专用的管理系统中，针对具体的数据库生成具体的逻辑表达式，是相当容易的事；但对于通用系统来说，就不那么简单了。因为，通用系统中的数据库是随机指定的，不同的数据库有着不同的结构；所以，设计一个随机组合的逻辑表达式生成程序是至关重要的。下面给出的源程序清单就是一个极好的通用逻辑表达式生成程序。执行该程序，只需按上下光标键、空格键或回车键即可完成条件组合。其中：上下光标键用于选择字段，空格键用于选择比较运算符及连接运算符，回车键为选择确认键，对某一字段可进行多次选择组合。

程序说明如下：

[1] 程序中使用 COPY TO <文件名> STRUCTURE EXTE 命令，生成被指定数据库的结构说明文件。

[2] 执行该程序时，首先需要输入需生成逻辑表达式的数据库文件名，然后打开该数据库，生成相对应的结构文件，并打开结构文件，对记录进行统计，产生内存变量数据文件 (LSM)，进入组合条件选择，对结构文件进行操作。

[3] 利用 INKEY () 函数截获键值，使用上下光标键任意选择组合条件所需要的字段名，键入回车键确认选择。

[4] 选择某字段后，程序自动显示比较运算符<比较运算符存于数组文件 (BJ) 中>，按空格键进行选择，键入回车键确认。

[5] 选择比较运算符后，程序自动显示与所选字段名类型、宽度相同的编辑格式，等待输入条件，该条件被存入内存变量数组文件。程序继续询问是否组合其它条件，如回答“Y”，则显示连接运算符，按空格键进行选择，键入回车键确认，然后换行继续显示字段名，再进行以上操作，直到结束，即可完成条件组合。

源程序清单如下：

```
* * * * 生成随机组合的逻辑表达式程序 * * * *  
SET TALK OFF  
SET SCOR OFF  
SET STAT OFF  
SET SAFE OFF  
SET ESCA OFF  
CLEA  
SET COLO TO 7/1  
PRIVA TJ1.JP.PS.ZM1.GH.HS.JLH.JLS.ZFS.CS.RQ.ZSL  
BLK=" "  
REST FROM BJLJ.MEM ADDIT  
DO WHIL . T.  
    STOR 0 TO JP.PS.HS  
    STOR " " TO ZC.ZM1.ZCX  
    STOR 1 TO GH.TJ1  
    FN=SPAC(10)  
    @10.25 SAY "请输入数据库文件名：" GET FN  
    READ  
    USE &FN  
    SET COLO TO N/B  
    @1.32 CLEA TO 23.78  
    SET COLO TO W+/R.B+/W  
    COPY TO FNJGK STRU EXTE  
    USE FNJG  
    JLS=RECC()  
    DIME LSM(JLS)  
    HS=11  
    CLEA  
    @4.35 SAY "请输入组合条件,可按定值,多值,范围组合操作"  
    GO TOP  
    DO WHIL . NOT. EOF()  
        DO WHIL . T.  
            JP=0  
            JLH=RECN()  
            ZM1=" "
```



```

@HS,32 SAY FIELD—NAME
LSM(JLH)=SPAC(FIELD—LEN)
@7,25 SAY "回车键确认,上下光标键选择"
JP=0
DO WHIL JP=0
    JP=INKEY()
ENDD
DO CASE
    CASE JP=13
        HS=HS+1
        EXIT
    CASE JP=5
        SKIP -1
        IF BOF()
            GO BOTT
        ENDI
    CASE JP=24
        SKIP
        IF EOF()
            GO TOP
        ENDI
ENDC
ENDD
DO WHIL . T.
    @18,42 SAY "请按空格键选择比较运算符:"
    @20,34 SAY "等于=,大于>,小于<,大于等于>=,小于等于<="
    @22,40 SAY "不等于<>,字符比较$,回车确认"
    IF GH>7
        GH=1
    ENDI
    @HS-1,40 SAY " "
    @HS-1,40 SAY BJ(1.GH)
    JP=0
    CLEA TYPEA
    DO WHIL JP=0
        JP=INKEY()
    ENDD
    DO CASE
        CASE JP=13
            EXIT
        CASE JP=32
            GH=GH+1
    ENDC

```