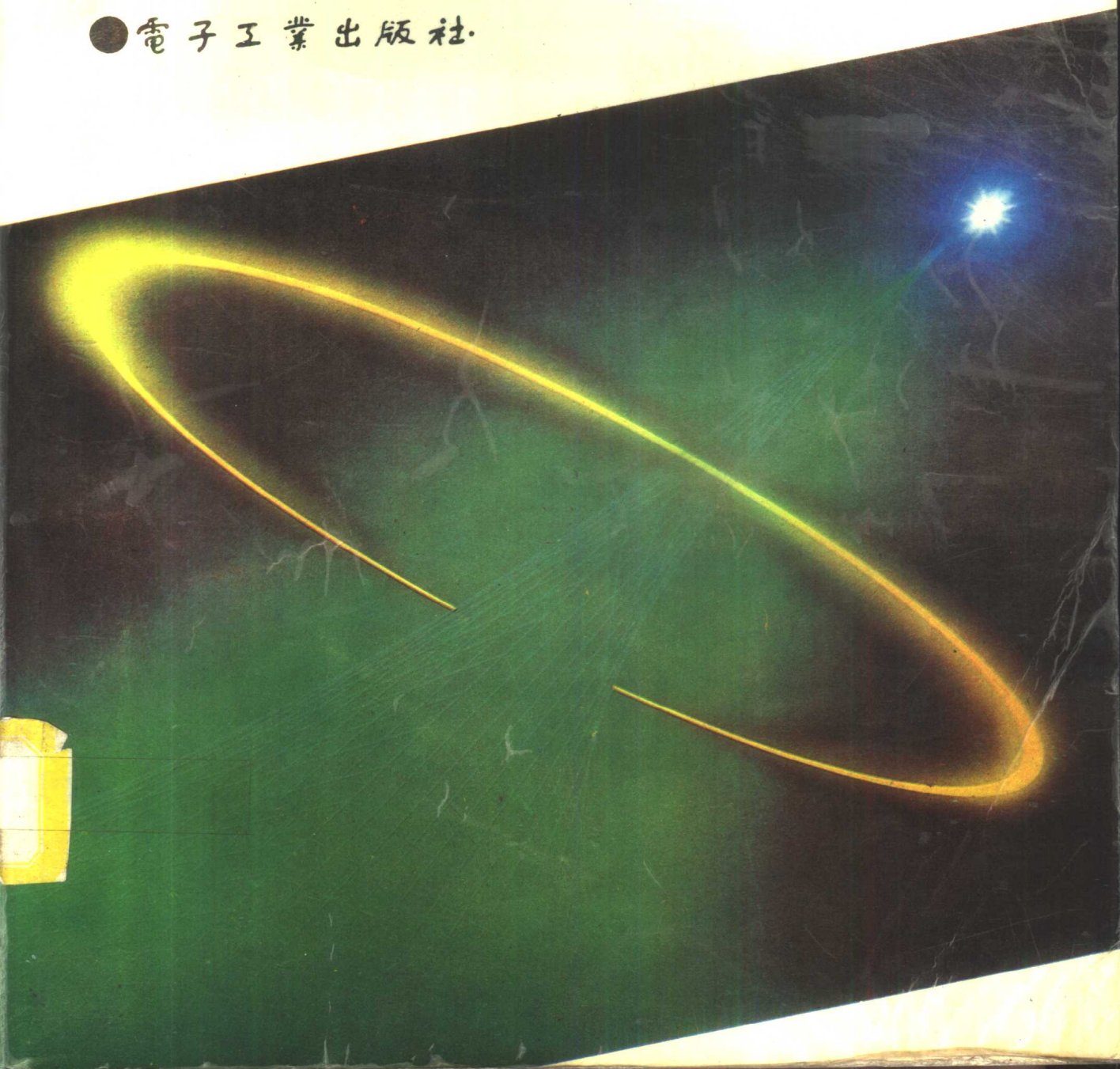


PC系列机 工具箱(五)

最新
出版

● 胡春芳等 编

● 电子工业出版社



PC 系列机工具箱(五)

胡春芳等编

电子工业出版社

(京)新登字 055 号

内 容 提 要

本书共分六个部分,介绍在 PC 系列机上数据库技术的应用技巧。主要内容有数据输入,菜单技术,数据查询与统计,屏幕编辑与显示技术,打印技术以及数据库综合应用技术。利用书中介绍的技巧和经验,能解决实际工作中的很多问题,是计算机用户的助手和朋友。

本书适合于从事计算机方面工作的工程技术人员、大专院校计算机专业师生阅读。

PC 系列机工具箱(五)

胡春芳等编

马晓晓审

责任编辑 张丽华

*

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经销

山东电子工业印刷厂印刷

开本:787×1092 毫米 1/16 印张:22 字数:560 千字

1995 年 6 月第 1 版 1995 年 6 月第 1 次印刷

印数:5000 册 定价:23.00 元

ISBN 7-5053-2901-4/TP·970

前 言

很久就想编一本书,一本有关计算机方面的工具性很强的书,因考虑不成熟,始终未能如愿。经几年实践积累,现推出笨拙之作奉献给广大读者,但愿它的诞生能给从事计算机方面工作的人员以帮助,成为他们的得力助手,充分发挥“工具箱”的作用。

书中所包含的内容是很多工程技术人员经验和智慧的结晶,也是我们编者辛勤劳动的结果,所有程序都进行了调试和验证。无可非议,此书的出版发行,将对我国计算机事业的发展有很大的帮助。

当然,因现在 PC 系列机种类繁多,难免有些软件因机型不同而产生不完全兼容的问题,很可能在一种机器上运行完好的程序,在另一种机器上实现不了。因此,希望读者在使用书中的技巧时,一定要注意其运行环境。

在本书的编写过程中,得到刘晓、王海东、时伟、徐立颖、王洁、张雪芳、王则青、张悦、黄卫民等同志的热心帮助,在此表示深深的感谢。因水平有限,书中可能会有一此疏漏之处,请广大读者给予指正,并诚恳地希望能将问题反馈给我们,以便及时修正,更希望同我们交流经验。

编 者

1994 年 8 月于北京

目 录

第一部分 数据输入	(1)
1. 在编译 dBASE III 中实现数据输入和修改	(1)
2. 加快数据输入的方法和技巧	(2)
3. 利用词典库实现 C-dBASE 的窗口输入	(2)
4. 利用“\$”进行自校	(4)
5. dBASE III 屏幕格式输入的设计方法	(5)
6. 综合检索、统计时汉字条件的输入	(6)
7. 如何加快录入速度	(9)
8. 数据录入正确性的程序检测	(10)
9. 代码录入正确性的判断程序	(12)
10. 充分利用屏幕方便数据录入	(14)
11. 学会运用@输入/输出格式命令	(17)
12. dBASE III 数据分屏输入法浅析	(17)
13. 一种较完善的 dBASE III 输入程序设计方法	(20)
14. 一种新颖、灵活的屏幕录入格式	(21)
15. 一屏多记录数据录入、编辑技巧	(25)
第二部分 菜单技术	(29)
1. 一种滚动式显示菜单	(29)
2. 通用的 dBASE III PLUS 菜单程序	(30)
3. 光条技术在 dBASE III 菜单程序中的应用	(33)
4. 用 C-dBASE III 实现下拉式菜单	(35)
5. 一种新颖的垂挂式菜单	(37)
6. 通用菜单程序的设计与实现	(40)
7. 一种方便实用的菜单生成器	(43)
8. C-dBASE III 实现下拉弹出式菜单	(44)
9. 在 dBASE III 中实现多级菜单的路径选择	(47)
10. 中文 dBASE III Compiler 菜单函数通用程序	(49)
11. 在菜单中设置密码	(51)
12. 利用 GRP 图形设备实现的屏幕菜单程序设计	(52)
13. 一种 dBASE III 菜单系统生成器的实现方法	(55)
14. 具有翻页功能的通用光条菜单程序	(58)
15. 全自动菜单生成程序	(61)
第三部分 数据查询与统计	(64)
1. 解决连续查询内存不够用的问题	(64)
2. 超宽报表屏幕查询的设计	(65)
3. 在 dBASE III 数据库中记录姓名对照表的提示查询	(66)

4. 使用宏代换实现交互式组合条件查询	(70)
5. 通用数据库查询程序	(72)
6. 在 dBASE III 状态下查找库文件的技巧	(74)
7. 词语索引的简单方法	(76)
8. 卡片式查询程序	(80)
9. 姓氏索引表的建立	(82)
10. 通用多重模糊查询	(83)
11. 一种实现通用多种模糊查询的方法	(86)
12. 如何正确实现 C-dBASE III 提供的索引功能	(89)
13. 加快超多库查找的速度	(89)
14. 复项随机检索简易实现法	(90)
15. 数据库多重条件查询一法	(93)
16. 用 SEEK 实现不等条件定位	(95)
17. 巧妙利用 Seek 和 Locate 命令实现数据库的快速检索	(97)
18. 快速查询索引文件结构	(98)
19. 多条件任意组合查询统计功能的实现	(99)
20. 如何完成有特殊要求的排序功能	(100)
21. 利用 Seek 命令代替 Count 命令计数	(102)
22. 利用 TOTAL 语句完成成批 COUNT 语句的功能	(103)
23. 通用 dBASE III 双重分类求和子程序	(104)
24. 自动计算某类文件总长度的方法	(105)
25. dBASE III 内存变量比较时出现的问题及解决方法	(106)
26. 巧用 TOTAL 命令实现对多个关键字段的同类合计	(107)
27. 统计磁盘文件长度、剩余空间的方法	(109)
28. 用数据库索引文件弥补 SORT 排序命令的不足	(110)
29. dBASE 实现统计图形	(111)
30. 在 C-dBASE III 中巧用宏替换函数实现快速统计汇总	(112)
31. 万能统计程序	(113)
第四部分 屏幕编辑、显示技术	(117)
1. CCDOS2.13H 特殊显示技术应用	(117)
2. 如何使屏幕显示更多的信息	(120)
3. 在 FoxBASE 下如何实现菜单的翻页显示	(121)
4. 用 GRD 图形设备实现屏幕表格的改进	(122)
5. 利用结构描述文件使显示和打印效果美观	(123)
6. 发掘 C-dBASE III 的动态画面功能	(125)
7. 编译 dBASE III 显示方式的转换	(125)
8. dBASE III PLUS VER 1.10 显示行数的修改	(126)
9. 能适应任何数据库的显示格式	(127)
10. 数据库字段名全自动显示	(128)
11. 占用拼音显示行一法	(130)
12. 联想式微机保持屏幕基本色彩的方法	(130)
13. 在窗口内显示移动信息	(131)
14. 微机字幕移动设计技巧	(133)

15. 谈一行字的屏幕移动显示	(134)
16. 在 dBASE III 中实现全屏显示代码库内容的方法	(136)
17. C-dBASE III 显示图形的方法	(137)
18. dBASE III 放大汉字的另一方法及程序	(139)
19. 编译 dBASE III 实现浏览编辑功能的方法	(141)
20. 消除 C-Clipper 屏幕跳跃的方法	(144)
21. 用 dBASE III 实现动态画面	(145)
22. dBASE III 清屏与显示方式种种	(146)
23. 在 dBASE III 下对屏幕显示颜色的选择	(148)
24. 一种屏幕动态显示汉字的方法	(149)
25. 如何使屏幕显示更清楚直观	(150)
26. 用 dBASE III 的 @ 和 COLOR 命令创造多彩的屏幕格式	(152)
27. 一种屏幕动态显示汉字的方法	(154)
28. 定做任意屏幕格式而无需建立屏幕格式文件法	(155)
29. 数据库立体直方图实时显示	(158)

第五部分 打印技巧..... (163)

1. dBASE 中一种新的报表打印方法	(163)
2. 标准信封编辑和打印	(169)
3. 利用回车指令解决打印锁机的问题	(171)
4. 在 dBASE III 下巧用 2463 打印机打印大字	(171)
5. 用数据库在打印机上分页输出源程序及文本文件	(172)
6. 用 dBASE III 进行文本的文稿格式打印	(172)
7. 用 dBASE III 设计打印直方图	(173)
8. dBASE III 利用汉字库放大打印输出浅析	(175)
9. dBASE III 换页打印的四种方法	(176)
10. 如何解决使用函数 PROW() 打印时发生锁机问题	(179)
11. 源程序快速分页自动打印	(179)
12. 在 dBASE III 中报表输出行超长时的解决方法	(182)
13. dBASE III 下 CHR(0) 的输出方法	(182)
14. dBASE III 中打印输出技巧	(184)
15. 打印机的驱动程序	(184)
16. 使用 CCED 将 dBASE 数据库进行横向报表输出的方法	(185)
17. 二维表格的通用打印程序	(187)
18. 稿纸格式打印	(189)
19. 如何实现多个 dBASE III 数据库结构的连续打印	(191)
20. 单字节制表符及其输入方法	(192)
21. 利用结构描述文件使显示和打印效果美观	(192)
22. 如何在同一行中打印大小不同字	(194)
23. AR 系列打印机控制方法	(195)
24. 用虚拟盘简化 dBASE III 打印程序的设计	(197)
25. 浮打半间断打印中防止打印机锁死	(197)
26. 自制一张区位码表	(198)
27. 规则表格的自动生成打印程序	(199)

28. 打印区位码和 ASCII 码对照表	(202)
29. 连续自动打印文本文件	(203)
30. 备注字段的表格打印	(205)
31. 如何实现打印程序的可移植	(206)
32. 解决不同版本 CCDOS 下打印机死锁问题	(207)
33. dBASE III 打印换页的实现方法	(209)
34. 用微机写带邮编的信封	(210)
35. 一种使 dBASE 数据库表格打印输出实现全自动的方法	(211)
36. 彻底解决打印机死锁问题的方法	(212)
37. 字型变换时制表符的连接问题	(214)
38. 打印 dBASE III 源程序清单程序	(214)
39. 用 dBASE III 实现分页分列打印的方法	(216)
40. 也谈用 dBASE III 建立通用中文资料打印程序	(219)

第六部分 综合技术..... (223)

1. 通用数据复核校验程序	(223)
2. 利用 FoxBASE 建立一库对多库的联系	(227)
3. 快速启动 dBASE III	(228)
4. 一个色彩斑斓的引导画面	(229)
5. 在微机上实现“卡拉 OK”	(230)
6. PRG 文件的结构化处理	(234)
7. 通用制表去零方法	(235)
8. 超多库数据登录程序的实现	(237)
9. 消除内存变量滚雪球现象	(241)
10. 设计、调试报表程序的一种方法	(242)
11. 数据库的转置方法	(243)
12. 微型计算机用户保密的一种方法	(247)
13. ZAP 命令删除数据库记录的恢复	(249)
14. 如何利用 dBASE III 的 F1 功能	(251)
15. 程序清单自动编排	(253)
16. 运用 dBASE III 对数据库进行压缩处理时应注意的问题	(255)
17. 数据库文件起始和结束标志的判定条件	(257)
18. 如何实现数据库中字符的替换	(258)
19. 汉字 dBASE III 应用技巧	(259)
20. 巧用 ALTERNATE 命令	(259)
21. 自动给程序加行号的方法	(260)
22. 一次性删除磁盘中所有 BAK 文件的方法	(261)
23. 批量备份 dBASE III 库结构简易方法	(262)
24. 一种简化 dBASE III 命令的方法	(263)
25. dBASE III PLUS 与 LOTUS 软件接口的汉化	(264)
26. dBASE III WAIT 命令在调试中的应用	(265)
27. dBASE III 中字符串前导空格的剔除	(266)
28. 获取 dBASE III 数据库结构值的简便方法	(267)
29. dBASE III 的窗口	(268)

30. 一种简便可靠的动态加密方法	(270)
31. 将子目录中的文件连续拷贝到软盘的程序	(271)
32. 关于 dBASE III 数据库的修复	(273)
33. 实现 dBASE III 数据报表中汉字布阵的一种简单方法	(274)
34. 提高 dBASE III 数据库删除操作效率的方法	(276)
35. 在 dBASE III 环境中实现屏幕表格	(276)
36. 编译 dBASE III 数据库中插入记录的实现	(277)
37. dBASE III 命令状态下对子目录文件的四种操作方法	(278)
38. WAIT 语句的几种功能	(279)
39. dBASE III 中的 LOCATE CONT 语句嵌套使用时的缺陷及其补救方法	(280)
40. dBASE III 中功能键的动态定义	(282)
41. 用 BASIC 语言处理 DBF 文件时异常问题的处理方法	(284)
42. 浅谈加速程序调试的方法	(286)
43. 利用日期清理硬盘的有效办法	(287)
44. 为系统设计一个引导画面	(290)
45. Readkey() 及 Inkey() 函数的简易取值	(291)
46. 谈 C-dBASE III 的动态画面功能	(292)
47. 如何使 dBASE III 具有产生随机数的功能	(293)
48. dBASE III 部分命令失效的补救	(293)
49. 巧用 DOS 重定向 I/O 功能实现大批量的数据库结构备份	(294)
50. 十六进制数与十进制数的相互转换	(295)
51. 文件工作区自动管理程序	(297)
52. 将非数据库结构文件传送到数据库的指定字段和指定记录中	(299)
53. 利用 FIND 命令检查程序错误	(301)
54. 巧妙处理字符“0”	(301)
55. 数据库记录的移动	(302)
56. 为零数值型字段的处理	(303)
57. 不通过赋值实现内存变量的存储	(304)
58. 当日文件自动备份	(305)
59. 对全屏编辑命令 BROWSE 做一点补充	(307)
60. 巧用单个功能键仿真 BROW 命令中的组合功能键	(308)
61. 如何恢复数据库中的数据	(311)
62. 数据库应用系统的分级保护	(312)
63. 巧用 peek 命令建立保密口令	(313)
64. SET EXACT OFF/ON 在编程中的应用	(314)
65. 在 dBASE III 中设置颜色的技巧	(316)
66. 浅析 dBASE III 数据库的重新索引	(317)
67. 编译 dBASE III 程序的几个问题	(318)
68. dBASE III 中 & 函数的应用	(320)
69. dBASE III 中 RUN/! 命令的使用	(321)
70. 巧用 dBASE III 进行文字编辑	(322)
71. 使用 Config.db 定义功能键	(323)
72. 删除 PRG 文件的前导空、备注行和空行的方法	(323)
73. 如何提高汉字 dBASE III 程序的执行速度	(325)

74. 运用编译 dBASE III 实现数据间条件合并、纵向迭加	(327)
75. 在 dBASE 中实现精确计算数据量的方法	(329)
76. 辅助字典在 dBASE III 中的应用	(331)
77. 字段宽度的动态生成方法	(333)
78. 用 C-dBASE III 作高精度直方图	(334)
79. 数据从 dBASE III 输送到 Informix 数据库的简便方法	(335)
80. 数据库中信息替换	(337)
81. 在 dBASE III 中 CONFIG. DB 文件的使用	(337)

第一部分 数据输入

1. 在编译 dBASE III 中实现数据输入和修改

编译 dBASE III 不支持解释 dBASE III 中 APPEND、CHANGE、EDIT 等直接对数据库记录进行输入和修改的命令。由于不同的数据库其库结构长度、宽度不同,在实现屏幕编辑时带来很多不方便。下面一段程序,可实现任意数据库记录的输入和修改。其中,A 区打开需进行输入或修改数据的主库,B 区打开的是一临时库。

该程序实现了记录的输入。只要将第四行语句改为记录指针定位语句(如 GO、FIND、LOCATE 等),即可实现特定记录的修改。

```

SELE A
USE ZK
COPY TO JK STRU EXTE
APPE BLANK
SELE B
USE JK
N1=1
M1=1
DO WHILE. NOT. EOF()
CLEA
N2=1
SELE A
@ 0,5 SAY '记录号:' +STR(RECNO(),4)
DO WHIL EN2<10
SELE B
D= 'ZD' -STR(N1,M1)&D=TRIM(FIELD NAME)
DD=&D
SKIP
SELE A
@ N2,5 SAY &D
@ ROW(),COL() SAY ':' GET &DD
N2=N2+1
N1=N1+1
IF N1>=10
M1=2
ENDIF
ENDDO
READ
SELE B
ENDDO
```

2. 加快数据输入的方法和技巧

一个数据库文件的结构建好以后,接着要做的工作就是输入数据,特别是在汉字信息较多的情况下,输入速度慢的问题较为突出。通过实践,加快数据库文件中的数据输入可以从以下几个方面着手:

(1) 正确使用 SET CARRY ON 命令,避免相同数据的重复输入

在为数据库文件输入记录时,常常遇到相邻记录的某些数据项相同的情况,对于这些相同的数据项实际是作了重复的输入操作,对于某些数据项是汉字的就更增加了输入负担和时间,如果能正确使用 SET CARRY ON 命令,就可避免重复操作和节省输入数据的时间。

该命令是在为数据文件追加(APPEND)或插入(INSERT)一个记录时,将当前文件的前一个记录复制到当前记录中,这样当前记录与前一个记录的相同之处可不再输入,只需要修改不同之处,一般情况下系统设置该命令为 OFF 状态。

(2) 利用编码代替汉字的输入方法尔后使用替换命令 REPLACE

在许多情况下,输入的数据是有规律的,而对汉字的处理要比字母及数字麻烦得多,尤其在输入汉字时更为突出。如果用数字或字母组成的代码代替汉字串的输入,不仅可以大大减少击键次数,提高输入效率,而且还可以减少出错率。所以使用 REPLACE 命令修改一批有规律的数据是很有效的,在用编码代替汉字后,库文件相应的数据项是编码,而在实际应用中我们所需要的是汉字信息,这时,可用 REPLACE 命令以汉字替换编码。

(3) 使用 SET FUNCTION 命令

该命令是用来定义键盘上 F2~F10 的功能,除 F1 不能定义外,F2~F10 均可由用户用 SET FUNCTION 命令来定义。每个功能键的功能可定义为含 1—35 个字符的序列,这些字符可以是数字、汉字和各种符号,所需字符串可以预先分别定义在 9 个功能键上,在需要时按下定义后的某功能键,便可直接输入该字符串,而且,功能键所定义的内容也可根据实际需要进行更换。

3. 利用词典库实现 C-dBASE 的窗口输入

对于较大的应用系统,有时有必要建立较大的词典库,对于这样的词典库采用编码输入法可实现窗口输入。比如对于籍贯项目可建立全国市县名词典库,并将其内容以拼音字头进行编码放置在对应的字段中。当输入籍贯时,只要键入市县名的拼音字头(比如北京市 BJ)程序即可在窗口中列出所有的重码市县名,编号从零到 kk(<10)供选择。若窗口中没有所需的内容,您可键入数字 kk+1,程序将脱离词典库接受用户的键盘输入。

利用词典库窗口输入比使用 DOS 词组节省内存且可提高输入速度,保证数据输入的准确,参见程序。

```
use zjk
```

```
appe blank
```

```
@2,17 say "籍贯"
```

```

sele 2
use JG inde ijj
bm= ' '
do while . t.
@2,1 say '输入编码' get;
bm pict '!!!!'
read
bm=trim(bm)
bm1= "" +bm+ ""
find &bm1
if eof()
loop
else
jlh=recn()
exit
endi
endd
yy=0
do while . t.
tt=0
do while tt<9
@tt+1,50 say str(tt+1)+JG
tt=tt+1
skip
if hzbmd<>bm
exit
endi
endd
cb=tt+2
@8,1 say "请输入编号,回车;选择" get cb rang 1,tt+2
read
if cb=tt+1
@2,17 say "籍贯" get jg
read
exit
endi
if cb<>tt+1
aa=yy*9+cb-1
go jlh
skip aa
sele 1
repl JG with jg->JG
@2,22 say JG
exit

```

```

endi
@0,50 clea
yy=yy+1
if hzbmd<>bm
go jlh
yy=0
endi
endd

```

杨根令

4. 利用“\$”进行自校

利用 C-dBASE III 编写应用程序时,输入数据是否正确的自校一般采用以下两种方法:

(1)采用 RANGE 短语自校;

(2)在循环程序 DO WHILE . T. 中用条件控制 IF 和 LOOP 语句配合进行自校。

方法 1 适合有规律的数字型数据的输入;方法 2 既适用于数字型,又适用于字符型。但在可供选择的字符型数据较多时,方法 2 组合条件复杂,程序较长。如果采用包含运算符“\$”进行自校能弥补其不足之处。下面结合一个通用建库模块说明其应用方法。在建立库结构时,字段类型代码有五种可供选择,即 C、N、D、L、M。若选此范围以外的字母时,条件 .NOT. FT \$ “CNDLM”成立,等待重新输入数据。否则,.NOT. FT \$ “CNDLM”不成立,即输入正确,退出循环,继续执行后面程序,从而起到了自校功能。

说明:程序中 JG. DBF 是一个结构库文件,它的四个字段就是一般数据库文件中每个字段的四个属性。

程序清单

* JG. PRG 建立库结构模块

SET TALK OFF

SET SAFE OFF

CLEAR

ACCE “请输入待建立结构的数据库名:” TO KM

IF .NOT. FILE("&KM.DBF")

USE JGK

COPY STRU EXTE TO &KM. JG

ENDI

@5,10 SAY “请输入数据库结构:”

@6,10 SAY “说明:1——字段名必须以字母开头”

@7,10 SAY “ 2——字段类型代码:”

@8,10 SAY “C—字符 N—数字 D—日期 L—逻辑 M—备注”

USE &KM. JG

APPE BLAN

X=1

FT=“ ”

DO WHILE . T.

XS=STR(X)

@10,10 SAY “结构 ‘+XS+’ :”

```

@11,10 SAY "字段名:" GET FIELD-NAME
READ
DO WHILE .NOT. FT$ "CNDLM"
    @12,10 SAY "字段类型:" GET FT PICT "A"
    READ
ENDDO
FIELD TYPE=FT
:
ENDDO

```

CREAT & KM FROM & KM. JG

CLOSE DATA

RETU

数据库结构 :A:JGK.dbf

记录个数 : 1

最后更新日期:10/17/90

字段	字段名	类型	宽度	小数
1	FIELD-NAME	Character	10	
2	FIELD-TYPE	Character	1	
3	FIELD-LEN	Character	3	
4	FIELD-DEC	Character	3	
* * 总计 * *				18

申全洲

5. dBASE III 屏幕格式输入的设计方法

dBASE III 为用户提供了丰富的输入/输出语句,并且都有一定的格式设计能力,其中功能最强的一类输入语句是@ say...get...和 READ 语句的结合。使用它可设计出所需要的屏幕格式。但是,对于下列问题,是否也能很好地解决呢?

在一个数据库中有 M 个字段,是用户输入数据的存放区,用户希望在某个时候只输入其中的 X 个字段,另一时候只输入其中的 Y 个字段,要求 M-X 或 M-Y 个字段处于输入范围之外,从而提高输入速度,降低输入的冗余度,特别是减少数据串项的可能性。

如果按照传统的程序设计方法,要解决这个问题,还得为每一种可能的输入方法编制一个程序,那么对于 M 个字段就要编写 2^M 个程序,可见几乎是不可能的。

避开传统的设计方法,采用程序自动生成的原理,较理想地解决了这个问题。

本方法的主要思想是:

(1)针对含有 M 个字段的数据库,建立一个有 M 个记录的数据库,取名为 DBSTRU. DBF。该库中至少含有存放 M 个字段段名的字段 MSTR 和一个标志字段 BZ。除此之外,根据需要可增加输入提示信息字段等其它字段。

(2)增设一个数据库 DBCOMM. DBF。该库只有一个字符型的字段 SAYCOM,用于存放自动生成的@ SAY...GET...命令序列,一个记录存放一条命令。

(3)编写程序一(XGBZ. PRG),该程序首先列出用于输入的 M 个字段的名称,并各给出一个编号,用户依编号选择本次要输入的和不要输入的字段,然后程序根据用户的选择,修改数据库 DBSTRU 中的 BZ 字段。

(4)编写程序二(AUTOPROG.PRG)。该程序根据 DBSTRU 中的 BZ,自动生成@ SAY ...GET 命令作为一字符串,放入 DBCOMM.DBF 中,然后用 COPY TO MENU.AUTSDF 命令生成文件 MENU.AUT。

(5)在有选择输入的用户程序段中,用下面的命令串 1:

DO MENU.AUT

READ

或命令串 2:

SET FORMAT TO MENU.AUT

CHANGE * 或 EDIT *

CLOSE FORMAT

就可以实现前面所讲的任选 X 或 Y 个字段进行输入或修改。

曹岩松

6. 综合检索、统计时汉字条件的输入

在使用 dBASE III 编的管理信息系统中,常有综合检索和综合统计两种功能,且检索及统计条件常含有汉字。由于使用对象是非专业计算机人员且带有各地方言,这样就造成汉字输入困难,而由于工作原因,他们不可能花去大量时间学习拼音及其他汉字输入法。基于上述原因,在程序中采用窗口提示一些固定字段,而非固定汉字取若干个连续关键字作为条件。

采取上述方法,即给使用者醒目提示,又免去输入汉字这一繁琐步骤,后面附程序清单。

在程序 RY2 中不难看出,检索条件中“部门、性别、民族、学历、行政职务及专业职务”为固定汉字字段,则可执行相对应的窗口程序 RY23n(n 即检索条件选号),如 RY235(学历提示)。其中 RYK 为学历库,只要按下相应的数字,汉字会自动填到空格中,其它类似。

在 RY5 中,可输入若干个关键字,其中 117 为判断标志。

程序清单

.TYPE RY235.PRG

@ 11,50 SAY '请选择'

S= ' '

@ 12,40 SAY '1—文盲;2—小学;3—职初;4—初中;'

@ 13,40 SAY '5—职高;6—高中;7—中技;8—中专;'

@ 14,40 SAY '9—专科;10—本科;11—双学士;12—研究生;'

@ 15,40 SAY '13—硕博;14—博研'

@ 15,56 GET S

READ

S=VAL(S)

IF S>=1.AND. S<=14

USE RYK1

GO S

STOR C_N TO XL

USE RYK_1

GO VAL(IB0)

REPL DATA WITH '&XL'

@ 7+IB,23 SAY DATA

```

RETU
. TYPE RY2. PRG
SET TALK OFF
CLEAR
IB=1
USE RYK .1
REPL ALL MARK WITH . F.
REPL ALL DATA WITH ' '
REPL ALL SIGN WITH ' '
GO TOP
DO WHILE .NOT. EOF()
    STORE ' ' TO IB0
    SET COLO TO 0/2
    @ 1,28 SAY '请输入检索条件'
    @ 2,5 SAY '1. 部门      2. 姓名      3. 性别      4. 民族      5. 学历'
    @ 3,5 SAY '6. 出生年月  7. 毕业年月  8. 基职工资  9. 原学专业  10. 从事专业'
    @ 4,5 SAY '11. 行政职务 12. 专业职务 13. 聘任时间 14. 毕业学校 15. 备注'
    @ 5,5 SAY '0. 检索'
    SET COLO TO 2/0
    @ 5,55 GET IB0
    READ
    IF IB0= ' '
        RETURN
    ENDIF
    IF IB0= '0'
        DO RY21
        RETU
    ENDIF
    IB0=VAL(IB0)
    IF IB0>=1 . AND. IB0<=15
        IF IB0=2
            DO RY24
            RETU
        ENDIF
        IF IB0=14
            DO RY25
            RETU
        ENDIF
        GO IB0
        REPL MARK WITH . T.
        IF IB0>=6 . AND. IB0<=8
            IF MARK
                IF DATA # ' '
                    SKIP 10

```