

计算机应用基础（二）

教案

信息学院

计算机应用基础课程建设小组

说 明

1. 本教案仅供参考，但每讲要点与进度应保持一致。
2. 讲命令不要纠缠语法规则，主要通过实例介绍基本用法。
3. 上课主要是讲解，并结合数据库实例做演示，但不需要将每个命令都在 COMMAND 窗口演示一遍。
4. 选择有代表性的命令演示命令的菜单操作，但不需对每个命令的菜单操作方式都演示一遍。
5. 函数重在应用示例的讲解
6. 不要引导学生纠缠一些比较复杂和过细的问题，如：
严格比较与模糊比较；While 条件；数组；结构文件；
数据库文件格式转换（SDF）；一对多和多对一问题；
以及对话框中非基本的各种选项；子程序与过程，等。
7. 结构化命令的讲解重在程序解读与演示上，并不断复习已讲过的命令。
8. 推荐数据库实例：（参考）
书本上的实例 - sb.dbf, sb.fpt, bmdm.dbf, zz.dbf, dx.dbf
Hd.dbf, Hd.fpt
教案实例 - E-mail.dbf, E-mail.fpt （电子邮件管理库）
staff.dbf （职工个人资料库）
address.dbf （职工通讯地址库）
payment.dbf （职工工资收入库）
Score.dbf （学生成绩库）
9. 程序（参考）
Say.prg 格式输出演示程序（按姓名查询记录）
Read.prg 格式输入演示程序（添加新记录）
Equatio1.prg 单边选择演示程序（解一元二次方程）
Equatio2.prg 双边选择演示程序（解一元二次方程）
Equatio3.prg 多边选择演示程序（解一元二次方程）
Case.prg 多边选择演示程序（添加记录与查询记录）
Append.prg Do循环演示程序（连续添加记录）
Max_p.prg For ...循环演示程序（求小于等于某个
正整数的最大素数）
@@@@@.prg For...循环演示程序（画棱形图案）
Check.prg Scan 循环演示程序（显示与修改关联
数据库数据）
Modi.prg 输入姓名修改数据库 STAFF.dbf 的记录
Print.prg 打印学生成绩表（SCORE.dbf）
Update.prg 连续修改多个人的综合信息
Wait.prg 使用 WAIT 命令暂停

10. 数据库与程序的缺省路径为 C:\teacher

李大丰电话： 84612984

E-mail: dafengxiao@a-1.net.cn

第一讲 数据库的基本概念和数据浏览与建库

要点：数据库基本概念

Foxpro 概述

数据库浏览

关系型数据库

建库

添加记录

修改记录

- 数据库基本概念 (P1-P6, 补充资料 1)

- **FoxPro 概述**

FoxPro 的安装环境 (P13)

硬件环境

软件环境

网络环境

FoxPro 的启动与退出 (P14)

启动：开始/程序/Foxpro for Windows， 双击桌面狐狸头

退出：命令菜单 File/Exit， 控制菜单/Close，

Quit 命令

FoxPro 的主窗口 (P18-P19, P21)

标题栏：控制菜单、标题、最小化/最大化 (还原) 按钮

菜单栏：File, Edit, Database, Record, Program, Run,

Text, Window, Help

状态栏：命令信息，系统反馈信息，操作状态

Foxpro 的命令窗口

显示：Window/Command

隐藏：Window/Hide

FoxPro 的操作方式 (P24)

菜单方式，命令方式，程序 (命令文件) 方式

- 数据库浏览 (P35-P38)

数据库文件：文件.dbf 命名规则 (P32)

文件的打开与关闭：File/Open, Use

数据库的浏览：

浏览：Database/Browse

转换显示方式：Browse/Change, Browse/Browse

滚动记录（鼠标方式，↑,↓,PageUp,PageDown）

移动字段（鼠标方式，Browse/Move Field）

改变字段宽度（鼠标方式，Browse/Size Field）

窗口的分割（鼠标方式，Browse/Resize Partitions）

窗口的切换（鼠标方式，Browse/Change Partitions）

窗口的关联（Browse/Link Partitions）

其他操作（Browse/Grid, Browse/Font.....,

Browse/Goto....）

关闭浏览窗口：控制菜单/Close, Esc 键

- 关系型数据库（P28-P30）

记录：二维表中的行数据

字段名：二维表中列数据集合的名称，长度不超过 10 的字符串（不含空格）

字段类型：字段数据值的类型

C: 字符型, N: 数字型, F: 浮点型, L: 逻辑型 (T,F)

D: 日期型, Memo: 备注型, General: 通用型

字段宽度:

C: ≤254, N: ≤20, F: ≤20, L: 1, D: 8,

Memo: 10, General: 10

注：备注型字段中的文字与通用型字段中的多媒体文件实际存放在扩展名为 .fpt 的备注文件中。Memo 和 General 字段仅存放链接信息

记录号：在输入记录数据时自动产生的记录序号

记录指针：用于记录定位的指针，数据库管理系统可以移动指针，指向不同的记录

当前记录：记录指针指向的记录

- 创建数据库（P30-P32）

打开表结构对话框：File/New/Table/New

定义表结构：字段名，字段类型，字段宽度，小数位

调整表结构：插入，删除，移动

确定（OK），保存文件到磁盘目录

- 添加记录操作（P33-P34）

打开数据库：File/Open

打开添加记录窗口：Record/Append

输入记录，回车

结束添加：控制菜单/Close, Ctrl+W, Ctrl+Q, Esc

- 修改记录操作

打开数据库: File/Open

打开修改记录窗口: Record/Change

滚动记录: (鼠标方式, ↑, ↓, PageUp, PageDown)

直接修改数据值

结束修改: 控制菜单/Close, Ctrl+W, Ctrl+Q, Esc

补充资料 1

一、基本知识

1. 数据和信息

以下面的档案表为例说明数据和信息的概念。

编号	姓名	性别	出生日期	籍贯	民族	政治面目
1001	李勇	男	42/08/08	上海市	汉族	党员
1002	纽丽萍	女	76/02/28	北京市	满族	团员
1003	李小琳	女	65/01/02	天津市	汉族	群众
1004	李盛国	男	73/05/08	云南省	彝族	群众

第一行是标题行,说明以下各列是何种信息。从第 2 行起,每一行的每一个格子里都是一个数据,每一个数据都反映了一个人的某一方面的信息,一行的 7 个数据就反映了这个人的基本信息。

2. 数据库

数据库通俗的解释是存储数据的仓库。而在计算机里,把组织成二维(行和列)形式的表格称为关系型数据库,简称数据库。上表就是一个数据库的例子。数据库的每一列称为一个字段,第一行为字段名行,称数据库的结构。其余每一行数据称为一个记录。同一字段的数据应具有相同的属性,库中不能有相同的记录。

早期还有两种类型的数据库:层次型和网络型,现已基本淘汰了。

数据库的特点是:

- (1) 数据的独立性。数据库和应用程序独立,和具体的程序无关。
- (2) 数据的共享性。可以为多个用户或多种语言程序使用。
- (3) 数据的冗余度小。即重复的数据少。节省资源且易于维护。
- (4) 数据的结构化。数据库文件之间可通过相同的字段建立联系,可减少重复的数据,节省了存储空间,可防止数据的不一致性。
- (5) 基本保证了数据的安全性和完整性,具有并发控制功能。由于数据的共享,带来了数据的安全性和完整性等问题。

为确保数据的安全性，允许采取安全措施。如规定密码、口令和存取权限，不得随意检索或修改库中的数据。

完整性是指数据的正确性与有效性的问题。为确保数据的完整性，提供了检验措施以控制数据在允许范围内。如当输入的数据越界或与定义的类型不符时发出警告等。

并发控制是为了防止多个用户并行操作数据库，引起库中数据发生不一致的问题。系统对并行操作能进行控制，比如采用封锁技术禁止并发更新。

3. 数据库管理系统

对数据库进行管理的软件。一般具有建库、编辑、修改、增删库中数据等维护数据库的功能；检索、排序、统计等使用数据库的功能；友好的交互输入/输出能力；方便、高效的数据库编程语言；允许多个用户同时访问数据库；提供数据的独立性、安全性和完整性等保障。

目前在微机和小型机上常用的数据库管理系统有以下几种：dBase III、FoxBASE+、Informix、Oracle和FoxPro。

4. 数据库系统

是指计算机系统中引进了数据库后的系统构成。通常可视为由人（用户、数据库管理员）、软硬件设备、数据库和数据库管理系统等组成的一个信息处理系统。

5. 数据库应用系统（程序）

针对特定的应用问题研制出的数据库系统。即以数据库为基础，由有关的人（用户、应用程序员、数据库管理员）、软硬件设备、数据库管理系统及各种应用程序包所组成的信息处理系统。

第二讲 数据库命令操作与内存表达式

要点: **FoxPro** 命令操作的例

FoxPro 命令结构

FoxPro 命令书写规则

内存变量与赋值

表达式及其值

表达式的值显示

● **FoxPro** 命令操作的例

Create C:\mysub\newdata.dbf

创建一个新数据库文件, 定义库结构
(相等于 File/New/Table/New)

Use C:\mysub\newdata.dbf

打开指定的库文件 (相等于 File/Open)

Browse

打开浏览窗口 (相当于 Database/Brows)

Goto....

Append

打开添加记录窗口 (相当于 Record/Append)

Use 关闭数据库

List 在主窗口滚动式显示数据库内容

Display all 在主窗口分页式显示数据库内容

List Structure 在主窗口滚动式显示数据库结构

Display Structure 在主窗口分页式显示数据库结构

Modify Structure 打开数据库表结构对话框,
修改库结构

● **FoxPro** 命令结构 (P38-P39)

命令动词 + 范围子句 + 条件子句 + 字段子句 + 其它

命令动词: List, Display, Browse 等

范围子句:

All 动词作用于所有记录

Next n 动词作用于当前记录及以下共 n 条记录

Record n 动词作用于记录号为 n 的记录

Rest 动词作用于当前记录直至最后一条记录

例: List record 3 显示记录号为 3 的记录

说明：将 3.14 乘以“半径”中的数据值的平方，
然后将计算结果存放到“面积”中去

注意：(1) 字符串值须带引号“”，日期值须带大括号
{ }，逻辑值须带点 . . .。

(2) 内存变量仅存在于内存中，退出 FoxPro 后
不再存在。

(3) 数据库字段名不能出现在等号的左方，但是
可以出现在等号右方，当它出现在等号右方
时代表当前记录中该字段的数据值。

(4) 常量与内存变量的类型：数值型，字符型，
日期型，逻辑型

赋值命令：Store ... to ...

例：Store 3.25 to x,
Store “王小刚” to 姓名
Store {05/23/72} to reg_date
Store .T. to single

● 表达式及其值 (P47-P48)

数值表达式：

运算符： + , - , * , / , ** , ^ , %

例：3.14*半径^2 , (x+y)/2/z , 7%3

运算顺序：先乘方，后乘除，再加减，括号优先

字符表达式：

运算符： + , -

例：楼号="A16 " , 房号="302" ,

楼号 + "-" + 房号 的值： "A16 -302"

例：楼号="A16 " , 房号="302" ,

楼号 - "-" - 房号 的值： "A16-302 "

逻辑表达式：

例：5<3 的值： .F. , "B">"A" 的值： .T.

关系运算符： < , > , = , <= , >= , <> , # , \$

逻辑运算符： .OR. , .AND. , .NOT.

例：judge1=5<3 , judge2="B" \$ "AB" ,

judge=judge1 .OR. judge2 , judge 的值： .T.

judge=judge1 .AND. judge2 , judge 的值： .F.

judge=.NOT. judge1 , judge 的值： .T.

judge=judge1 <> judge2 , judge 的值： .T.

日期的运算:

例: {07/16/99} - {06/20/99} 的值 : 26 (数值)

{06/20/99} + 26 的值: {07/16/99} (日期)

● 表达式的值显示 (P49-P50)

? 表达式, 表达式, ... 换行显示各表达式的值

例: ? {07/16/99} - {06/20/99} 显示: 26

? 5 < 3 .AND. "B" > "A" 显示: F

例: ? "姓名=", name 显示: 姓名=王小刚

注: 字符串在屏幕显示时不出现双引号, 日期在屏幕出
现时不出现大括号, 逻辑值在显示时不出现 . .

? ? 表达式, 表达式, ...

在原光标位置显示各表达式的值

例: ? "姓名=" ? "姓名="

? name ?? name

结果: 姓名= 结果: 姓名=王小刚

王小刚

第四讲 函数及其应用

要点：基本函数

函数应用举例

● 基本函数 (P50-P52)

INT (N) 数值取整

例： ? Int (23.54) : 23

ROUND (N1,N2) 数值 N1 四舍五入保留 N2 位小数

例： ? Round (23.54, 4) : 23.5400

 ? Round (23.54, -1) : 20

SQRT (N) 正数值的算术平方根

例： ? SQRT (9) : 3

MOD (N1,N2) 自然数 N1 被 N2 整除的余数

例： ? Mod (7, 3) : 1

MAX (N1, N2, N3, ...) 求大数

例： ? MAX (3.45, 6.73, 2.41) : 6.73

MAX (C1, C2, C3, ...) 求 ASCII 码最大的字符串

例： ? MAX ('abc', 'abd') : 'abd'

MIN (N1, N2, N3, ...) 求小数

例： ? MAX (3.45, 6.73, 2.41) : 2.41

MIN (C1, C2, C3, ...) 求 ASCII 码最小的字符串

例： ? MAX ('abc', 'abd') : 'abc'

LEN (C) 给出字符串 C 的字节数

例： C="王"+"小刚" ? Len (C) : 6

LTRIM (C) 截去字符串 C 的左空格

例： C=" abc" ? Len (Ltrim (C)) : 3

RTRIM (C) 截去字符串 C 的右空格

例： C="abc -" ? Len (Rtrim (C)) : 6

TRIM (C) 与 RTRIM (C) 相同

ALLTRIM (C) 截去字符串 C 的首尾空格

例： C=" abc -" ? Len (Alltrim (C)) : 8

LEFT (C, N) 取字符串 C 的左 N 位

例: ? Left ("64492552", 4) : "6449"

RIGHT (C, N) 取字符串 C 的右 N 位

例: ? Right ("64492552", 4) : "2552"

SUBSTR (C, N1, N2) 取字符串 C 的从左起第 N1 位
开始的长度为 N2 的子串

例: ? Substr ("001-212-746-2986", 5, 3) : "212"

LOWER (C) 将字符串 C 的字符全部该为小写

例: ? Lower ("aBCD-ef") : "abcd-ef"

UPPER (C) 将字符串 C 的字符全部该为小写

例: ? Upper ("aBCD-ef") : "ABCD-EF"

SPACE (N) 产生 N 个空格的字符串

例: ? Len ("A"+Space (2)+"2597") : 7

STR (N1, N2, N3) 保留 N3 位小数, 改数值 N1
为长度为 N2 的字符串

例: n=23.3464 ? Str (n, 7, 2) : " 23.35"

VAL (C) 将数字形式的字符串转换为数字

例: ? VAL (' 23.35') : 23.35

AT (C1,C2,N) 给出 C1 在 C2 中第 N 次出现时的
字符位

例: ? At ("f", "吉祥村 f 座 5 号", 1) : 8

DATE () 给出机内今日日期

例: ? Date () : 01/30/00

TIME () 给出机内时间

例: ? Time() : 16:54:35

DTOC (D) 将日期 D 转换成字符串

例: ? Dtoc ({01/30/00}) : "01/30/00"

CTOD (C) 将字符串 C 转换成日期

例: ? Ctod ("01/30/00") : 01/30/00

YEAR (D) 返回日期 D 的年份 (数值)

例: ? Year (Date ()) : 2000

MONTH (D) 返回日期 D 的月份 (数值)

例: ? Month ({10/23/99}) : 10

DAY (D) 返回日期 D 的日号 (数值)

例: ? Day ({10/23/99}) : 23

? 代码

Use D:\mysub\Payment.dbf

Replace all total with Int (salary+allowance+award)

第五讲 数据传递与数据库文件操作

要点: 记录与内存变量组的数据传递

从另一数据库添加记录

复制数据库文件

单独复制库文件结构

产生排序文件

磁盘文件操作命令

- 记录与内存变量组的数据传递 (P61)

Scatter Memvar Fields 字段名, 字段名, ...

将当前记录的指定字段数据复制给一组内存变量,

内存变量名相应为 m.字段名 (内存变量自动定义)

例: Scatter Memvar Fields name, sex, birthday

? m.name, m.sex, m.birthday

显示结果: 王小刚 男 02/08/76

Gather Memvar Fields 字段名, ...

将内存变量组的当前值替换当前记录相应字段的值

例: m.name = "李芳"

m.sex = "女"

m.birthday = {03/15/77}

Gather Memvar Fields name, sex, age

? name, sex, age

结果: 李芳 女 03/15/77

例: Append Blank

Scatter Memvar

m.name = "陈军"

m.sex = "男"

m.birthday = {10/23/80}

m.married = .F.

Gather Memvar

结果: 数据库添加一条非空记录

- 从另一数据库添加记录 (P58) (菜单操作: Database/Append From....)

Append From 数据库路径及全名 条件子句 Fields

字段名, 字段名...

将另一数据库符合条件的记录中相应字段的数据逐条添加到当前库中

例: 假设当前库与 D:\mysub\mydb2.dbf 中都包含 name, sex, birthday 字段, 并且 D:\mysub\mydb2.dbf 中有 11 条记录

```
Append From D:\mysub\mydb2.dbf Fields name,
sex, birthday
```

结果: 当前数据库增加 11 条记录, 其中 name, sex, birthday 三个字段非空

注意: 指定的字段必须是两个数据库共有的, 且类型相同, 此外两个库都可以包含不同的其它字段

- 复制数据库文件 (P61-P62) (菜单操作: Database/Copy To...)

Copy To 新数据库路径及全名 范围 条件 Fields 字段名, 字段名, ...

将当前库指定范围符合条件的所有记录的相应字段数据逐条复制到新的数据库中, 新数据库具有指定的字段且字段类型与宽度与当前库相同

例: 假设当前库有 10 条记录, 字段为 name, sex, birthday

```
Copy To D:\mysub\new.dbf All Fields name,
birthday
```

结果: 在 D:\mysub 中产生一个新数据库, 文件名为 new.dbf, 有 10 条记录, 包含两个字段 name 和 birthday

例: Copy To D:\mysub\copy.dbf

结果: 在 D:\mysub 中产生一个与当前库完全相同的数据库, 文件名为 copy.dbf

注意: 若字段子句包含备注或通用字段, 则新库自动包含相应备注文件 (.fpt)

- 单独复制库文件结构 (P62)

Copy Structure To 数据库路径及全名 Fields 字段名, 字段名, ...

在指定磁盘目录中产生指定字段结构的新数据库（无记录）

例：Copy Structure To D:\mysub\newbase.dbf

结果：在 D:\mysub\ 中产生一个新的库文件，结构与当前库完全相同，其中没有记录，文件名为 newbase.dbf

- 产生排序文件（P72）（菜单操作：Database/Sort）

Sort To 新数据库文件路径及全名 On 关键字段名

[/A 或 /D] 范围子句 条件子句 字段子句

产生一个新的数据库文件，包含当前库中指定范围符合条件及指定字段的记录，其中的记录按关键字段值的大小排序（ /A 升序， /D 降序 ）

例：Sort To D:\mysub\sortbase.dbf On name /D

结果：在 D:\mysub 中产生名为 sortbase.dbf 的新数据库，结构与当前库相同，包含所有当前库记录，但记录号顺序按姓名值从大到小排列

- 磁盘文件操作命令（P68）（与相应 DOS 命令相似）

Rename 文件路径及全名 To 新的文件路径及全名

Erase 文件路径及全名

Delete File 文件路径及全名

Copy File 源文件路径及全名 To 目标文件路径及全名

Dir 目录路径

Type 文本文件路径及全名

第六讲 索引与查询（一）

要点：建立复合索引文件

引用复合索引排序

顺序查询

索引查询

为了便于在数据库中查找符合特定条件的记录，往往需要让数据库中的记录按特定的顺序排列（好比字典按字母顺序排列单词一样）。FoxProw 允许用户任意排列记录的顺序，但要求用户事先针对数据库建立索引文件，索引文件自动保存索引关键字值与记录号的对照（好比索引卡片），例如职工姓名值与记录号的对照，设备编号值好与记录号的对照，等。

索引文件与数据库文件不同，它从属与特定的数据库，不能通过 File/Open 或 Use

命令打开和编辑。

- 建立复合索引文件 (P74-75)

Index On 关键字表达式 Tag 关键字标识

[Ascending 或 Descending]

在磁盘上自动建立文件名与当前数据库名相同而扩展名为: .cdx 的复合索引文件, 并在该文件中自动添加关键字值与记录号的对照, Ascending 指定关键字值按升序排列, Descending 指定关键字值按降序排列。Ascending 为缺省选择。

例:

```
Use D:\mysub\Staff.dbf
```

```
Index On name Tag nm Descending
```

```
Index On sex Tag sx Ascending
```

```
Index On birthday Tag bd Descending
```

结果: 建立数据库文件 Staff.dbf 的索引文件, 文件名自动为 Staff.cdx, 该文件分别保存姓名值, 性别值, 生日值与记录号的对照, 其中姓名和生日按降序排列, 性别按升序排列。nm, sx, bd 分别是姓名, 性别, 和生日三个索引关键字的标识符。

注: 必须首先打开数据库文件才能建立索引文件

注: 在第一次使用 Index on 命令时自动建立索引文件, 以后再次使用 Index On 命令时, 仅在索引文件中添加一项索引, 并不产生新的索引文件。

注: 关键字索引标识符由用户自定, 通常按某个字段值建立索引, 标识符可以就是该字段名或其缩写

注: 在打开数据库时自动被调用, 在关闭数据库时自动退出

注: 当数据库添加, 插入或删除记录后, 复合索引文件会自动相应更新

- 引用复合索引排序 (P76)

Set Order To

记录按记录号从小到大排序

Set Order To 索引标识符

记录按事先建立的相应索引关键字顺序排列

例:

Use D:\mysub\Staff.dbf

Set order To nm

List

结果：按姓名大小降序显示记录

Set Order To sx

List

结果：按性别值大小升序显示记录

Set Order To bd

List

结果：按生日大小降序排列记录

Set Order To

List

结果：按记录号大小升序排列记录

注：在 Set Order 命令中使用的关键字标识符应与建立复合索引文件时定义的关键字索引标识符一致。

注：复合索引文件中可能保存多个索引，通过 Set Order 命令确定的索引称为当前控制索引

- 删除已建立的索引 (P77)

Delete Tag 索引标识符, 索引标识符, ...

例：

Delete Tag sx , bd

结果：复合索引表中只剩下姓名索引

- 顺序查询 (P77-78)

Locate For 条件

将记录指针从首记录开始下移到符合条件的第一条记录处。

例：

Use d:\mysub\sb.dbf

Locate for department="12"

? recno()

结果：显示 5

Locate for department="12"

? recno()

结果：显示 5

Continue

将通过 Locate 命令定位的指针继续下移到下一条符合条件的记录处。

例：

```
Use D:\mysub\sb.dbf
```

```
Locate for department="12"
```

```
? Recno()
```

```
结果：显示 5
```

```
Continue
```

```
? Recno()
```

```
结果：显示 6
```

注：Continue 命令必须与 Locate 命令配合使用

注：如果通过 Continue 命令再也找不到符合条件的记录，指针将停在文件结束处

注：Locate 命令既适用于引用了索引排序的情况，也适用于没有引用索引排序的情况

● 索引查询 (P78)

Seek 索引关键字表达式

将记录指针从首记录开始下移到指定关键字值的第一条记录处。

例：

```
Use D:\mysub\Staff.dbf
```

```
Set Order To Tag sx
```

```
Seek "女"
```

```
? Recno()
```

```
结果：显示 2
```

```
Seek "女"
```

```
? Recno()
```

```
结果：显示 2
```

注：Seek 命令只适用于引用了索引排序的文件。

注：Seek 命令中直接写关键字值，而不是逻辑条件

注：如果要将指针移动到关键字值相同的下一条记录，可以使用 skip 命令，因为在排序的情况下，关键字值相同的记录一定是连续排列的。

注：如果用 Seek 命令找不到相应的记录，则指针停在文件结束处。

注：通过 LOCATE 命令或 SEEK 命令查找记录时，若找到，函数 FOUND（）值为 T；若找不到，函数 FOUND（）值为 F。

第七讲 索引与查询（二）

要点：单索引文件的建立，引用与取消和更新

使用 **RQBE** 窗口查询

● 单索引文件的建立，引用与取消和更新（P74-77）

单索引文件与复合索引文件的不同在于：

- (1) 单索引文件只保存一项索引关键字值与记录号的对照，而复合索引文件可以保存多个关键字索引。
- (2) 复合索引文件的名称与数据库名相同，无须在命令中指定，而单索引文件的文件名必须在命令中由用户定义，其扩展名是 .idx 而不是 .cdx
- (3) 复合索引文件在打开数据库时自动被引用，而单索引文件必须通过命令被引用
- (4) 它们的建立与引用的命令也有所不同。

建立单索引文件（P74）

Index On 关键字表达式 To 单索引文件全名

[Ascending 或 Descending]

例：

```
Use D:\mysub\Staff.dbf
```

```
Index On name To D:\mysub\name.idx Descending
```

在磁盘上建立数据库 staff.dbf 的姓名索引文件，文件名为 name.idx，姓名按降序排列

注：可以对一个数据库建立几个不同关键字的单索引文

件，根据不同的需要引用不同的索引。

引用单索引文件排序（P75）

Set Index To 单索引表文件全名

例：

```
Use D:\mysub\Staff.dbf
```

```
Set Index To D:\mysub\name.idx
```

```
Disp all
```

结果：显示的记录按姓名的值从大到小排列

取消单索引排序（P75）

Set Index To

例：

```
Set Index To
```

Disp all

结果：显示的记录按记录号的值从小到大排列

单索引文件的更新（P77）

Reindex

例：

Use D:\mysub\Staff.dbf

Append blank

Replace name with “刘小庆 “ birthday with {02/07/79}
married with .F.

Set Index To name.idx

Disp all

结果：显示记录中没有新添加的记录

Reindex

Disp all

结果：显示记录中有新添加的记录，并且排列在合适的位置

注：对于已经被引用的单索引表，当数据库发生变动时，单索引文件内容会自动相应更新。对于未被使用的索引文件，当数据库发生变动后，必须先通过 Set index 命令被引用，然后通过 Reindex 修改

注：为了保证单索引文件与数据库的一致，最好在数据库更新后，立即打开所有单索引文件，进行 REINDEX。

如：

Set Index To name.idx , sex.idx , birthday.idx , ...

Reindex

- 使用 RQBE 窗口查询记录（P89-93）

RQBE 窗口是 FoxPro 提供的窗口工具，允许用户选择数据库记录的查询方式（显示记录所满足的条件，显示哪些字段值，记录排列顺序，等）。并允许用户将此设置保存为查询文件，将来运行此查询文件即可显示查询结果。

（1）运行 RQBE，完成查询设置：Run / New Query

常用设置：

Selection Criteria 指定显示哪些记录

（字段值是什么或不是什么，考虑不考虑大小写，等）。

Fields... 指定显示哪些字段值

(选择按钮: move, remove, ok, 等)

Order By... 确定排序字段及升降序

(选择按钮: move, remove, ascending, descending, ok, 等)。显示记录按第一关键字, 第二关键字, ...的顺序排列。

(2) 显示查询结果: Do Query

(3) 保存查询文件: File / Save (文件扩展名为 .qpr)

(4) 运行已建立的查询文件: Run / Query

(5) 修改已建立的查询文件: File / Open

注: RQBE 窗口工具用于建立查询文件, 该文件是与特定数据库有关的查询程序。用户可以针对数据库建立多个不同的查询文件供将来使用。

注: 使用 RQBE 窗口设置查询条件时应该首先打开数据库。

注: RQBE 窗口中的 Order By... 设置与数据库是否有索引文件无关

注: 查询文件 (.qpr) 是一个查询程序, 数据库记录修改后, 该查询设置仍可使用。

第八讲 数据库统计

要点：计数命令

求和命令

求平均命令

计算命令

汇总命令

使用 **RQBE** 窗口显示统计结果的例

- 计数命令 (P102) (菜单命令：Database/Count...)

Count 范围子句 条件子句 To 内存变量名

计算指定范围内符合条件的记录的条数，将结果赋予某个内存变量。

例：

```
Use D:\mysub\staff.dbf
Count All For sex="男" To no_male
? no_male
结果显示： 6
```

- 求和命令 (P102-103) (菜单命令：Database/Sum...)

Sum 数值表达式 范围子句 条件子句 To 内存变量名

将指定范围内符合条件的每条记录的相应表达式值求和，将结果赋予某个内存变量。

例：

```
Use D:\mysub\payment.dbf
Sum Int(salary+allowance+award) To total_Pay
? Total_pay
结果显示： 15690.00
```

- 求平均命令 (P103) (菜单命令：Database/Average...)

Average 数值表达式 范围子句 条件子句 To 内存变量名

将指定范围内符合条件的每条记录的相应表达式值求平均，将结果赋予某个内存变量。

例：

```
Use D:\mysub\payment.dbf
Average salary+allowance+award For Title="助教"
To zhujiao_Pay
? zhujiao_pay
结果显示： 1100.00
```

- 计算命令 (P103) (菜单命令：Database/Calculate...)

Calculate 统计函数 范围子句 条件子句 To 内存变量名

对指定范围内符合条件的记录进行统计，将结果赋予某个内存变量。

例：显示最大收入和最小年龄

```
Use D:\mysub\payment.dbf
```

```
Calculate Max(salary+allowance+award) To max_income
```

```
? max_income
```

结果显示： 2420.50

```
Calculate Min(Year(Date())-Year(birthday)) To yanger
```

```
? yanger
```

结果显示： 21.00

注：基本统计函数有

SUM (数值表达式)

AVG (数值表达式)

MAX (表达式)

MIN (表达式)

- 汇总命令 (P104) (菜单命令： Database/Total...)

Total On 汇总字段名 To 汇总文件 范围子句 条件子句

Fields 数值字段名, 数值字段名,

对指定范围内符合条件的记录中的指定数值字段，按汇总字段的值进行分类求和，并产生一个新数据库文件（汇总文件）。汇总数据库的记录条数等于被汇总数据库中汇总字段值的类别数。

例：

```
Use D:\mysub\payment.dbf
```

```
Index On title Tag title
```

```
Set Order to title
```

```
List
```

```
Total On title To D:\mysub\totalpay.dbf
```

```
Fields salary, allowance, award
```

```
Use D:\mysub\totalpay.dbf
```

```
List Fields title, salary, allowance, award Off
```

```
List
```

结果显示：各职务的工资总额，津贴总额，奖金总额

注：使用 Total 命令分类汇总前，数据库必须按汇总字段索引排序

注：汇总数据库包含被汇总数据库的所有字段，其记录的非数值求和字段保留原排序数据库中每个类别记录中的第一条记录的相应值。

- 使用 **RQBE** 窗口显示统计结果的例

使用 RQBE 窗口显示各种收入按职务分类的平均数

File / Open payment.dbf

Run / New Query

Select Criteria: payment.retired Like .F.

Order By...

Title

Group By ...

Title

Fields...

Title

Avg (payment.salary)

Avg (payment.allowance)

Avg (payments.award)

Do Query

第九讲 多数据库操作与 VIEW 窗口设置

要点：多工作区概念

多数据库操作

数据库关联

VIEW 窗口设置

- 多工作区概念 (P79-81)

FoxPro 在内存中设置了多个区域，用以放置多个不同数据库的记录，这些区域叫做工作区。编号为 1 - 225，前十个工作区也可以用 A - J 十个字母作为别名。工作区中打开的数据库名也可以用来作为工作区别名，此外用户还可以根据自己习惯和方便在命令中给工作区定义别名。

在一个工作区打开新数据库文件时，该工作区中原先打开的数据库自动关闭。

- 多个数据库操作 (P80-81)

(1) 在不同工作区打开不同数据库

例：

Select A

Use D:\mysub\staff.dbf

Select 2

Use D:\mysub\address.dbf

Select 0

Use D:\mysub\payment.dbf

注：select 0 表示选择编号最小的尚未打开数据库的工作区

(2) 对不同数据库操作时需要先选定相应的工作区。

例：

Select B

List Fields name, telephone

Select staff

List Fields name birthday

Select 3

List Fields name, salary

(3) 当前库与当前记录的概念

当前选定的工作区中打开的数据库称为当前库，当前库记录指针指向的记录称为当前记录。

例:

Select payment

Locate For title="打字员"

? Recno()

结果显示: 6

Select staff

Locate For name="王小刚"

? Recno()

结果显示: 4

Select payment

Display name, title

结果显示: 6 赵芬 打字员

Select staff

Display name, sex

结果显示: 4 王小刚 男

(4) 在当前工作区使用不同库的数据

例:

Select staff

Locate For name="王小刚"

Select address

Locate For name="王小刚"

Select C

Locate For name="王小刚"

? name, title, sex, telephone

注: 显示 Sex 和 Telephnoe 不存在

? name, title, staff.sex, B.telephone

结果显示: 王小刚 助教 男 84128898

注: 不能写 2.telephone, 1.sex 等

? Locate For name = "赵芬"

? name, tilte, staff.sex, B.telephone

注: staff.sex 和 B.telephone 的值没有变 (指针没有相应移动)

- 数据库关联 (P87-89)

数据库关联指当前库记录指针移动时, 其它几个数据库的指针自动相应移动。建立数据库关联时, 当前库与被关联的数据库必须有相同的关键字段, 并且被关联的数据库必须

事先引用关键字索引。

Set Relation To 关键字 1 Into 被关联库 1,

关键字 2 Into 被关联库 2 ,

建立当前库与另几个库的关联

例:

Select address

Set Order To Tag name

Select payment

Set Order To Tag name

Select staff

Set Relation To name Into address , name Into payment

Locate For name="王小刚"

? name, sex, payment.title, address.telephone

结果显示: 王小刚 男 助教 84128898

Replace sex with "女",

address.telephone with "66666666",

payment.title with "教授"

? name, sex, payment.title, address.telephone

结果显示: 王小刚 女 教授 66666666

Display all name, sex, payment.title, address.telephone

结果显示所有人的姓名, 性别, 职务, 电话

注: 如果当前库与其它工作区中的数据库没有建立指针关联, 结果给出的数据将会牛头不对马嘴。

Copy To D:\mysub\join.dbf Fields name, sex, payment.title,

Address.telephone

Sele 0

Use D:\mysub\join.dbf

List

结果显示新数据库的所有记录, 字段为姓名, 性别, 职务,

电话。Join.dbf 是三个数据库中若干字段的连接。

连接命令:

Join With 工作区名 For 连接条件 To 新数据库

文件全名 Fields 字段名, 字段名, ...

将当前库与另一库的记录按连接条件逐条连接, 并将有关字段数据复制到新表。

例: Join With address For name = address.name
To D:\mydir\join.dbf Fields name, title,
B.telephone

● **VIEW 窗口设置 (P91-83)**

VIEW 窗口是 FoxPro 提供的窗口工具, 在 VIEW 窗口中用户可以:

指定用户文件路径缺省值

在多个工作区打开或关闭数据库

修改数据库索引

设置控制索引

建立数据库的关联

浏览数据库记录

等

(1) 指定用户文件路径缺省值

Window / view 打开 VIEW 窗口

左击文件按钮 (左方第三个图标)

左击 FoxPro path... 按钮, 选择用户文件路径

(2) 在多个工作区打开和关闭数据库

左击观察按钮 (左方第一个图标)

左击工作区编号, 左击 Open 按钮, 打开数据库

左击已打开数据库名, 左击 Close 按钮, 关闭数据库

(3) 修改数据库索引

左击已打开数据库名, 左击 Setup 按钮打开 Setup 窗口

左击 Indexes 线框中的 Modify... 按钮打开 Index 窗口

在 Fields 线框中左击新的关键字段, 并单选升降序, 然后左击 Add to index list 按钮, 完成添加索引关键字

在 Index Key 列表框中左击关键字段名, 然后左击 Remove From List 按钮, 取消该索引

左击 Ok 按钮确认修改

(4) 设置控制索引

左击已打开数据库名, 左击 Setup 按钮打开 Setup 窗口

左击 Indexes 线框中的关键字段, 左击 Set Order 按钮

设置为当前控制索引 (钥匙图标表示当前控制索引)

左击 Indexes 线框中的 No Order 按钮可取消控制索引

左击 Ok 确认设置

(5) 建立数据库的关联

在观察窗口中左击一个数据库名（主库）

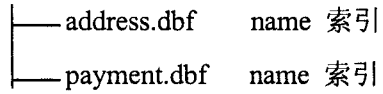
左击 **Relations** 按钮，左击另一个被关联的数据库名

（在右方滚动框中可以看到树状关联图示）

(6) 浏览数据库记录

在观察窗口中左击已打开数据库名，确定为当前库，然后左击 **Browse** 按钮

例：staff.dbf



VIEW 窗口的设置可以通过 **File / Save As...** 保存为扩展名为 **.vue** 的 View 文件。将来通过 **File / Open** 打开此文件时可以自动恢复所有原先的设置（工作区，数据库，索引，关联，等）

第十讲 FoxProw 程序

要点：程序的一般概念

注释语句

状态设置命令

输入输出命令

运行控制命令

● 程序的一般概念 (P109-111)

程序是按一定规则书写的命令和语句的序列。程序文件名由用户按一般文件命名规则自定，扩展名是 .prg。

Fox Prow 能够检查程序的错误，编译程序，并执行程序。

源程序：用户编写的程序，扩展名：.prg

编译程序：FoxProw 完成编译的程序，扩展名：.fxp

编译程序不能打开编辑

(1) 打开/关闭程序编辑窗口：

打开编辑窗口编辑新程序：

File / New / Program 或命令：Modify Command

打开编辑窗口编辑已有源程序：

File / Open 或

命令：Modify Command 程序文件全名

关闭源程序编辑窗口：

File / Close , Esc , 控制菜单/Close 关闭窗口

(2) 保存编辑结果：File / Save , File / Save As...

(3) 运行程序：

编译并运行源程序（不含参数）：

Program / Do 或 命令：Do 文件路径及全名

编译并运行含参数的程序：

命令：Do 文件路径及全名 With 参数 1, 参数 2, ...

一个程序被运行后 FoxProw 在磁盘目录中自动产生扩展名为 .fxp 的编译程序，文件名与源文件名相同。

程序及其编辑运行的例：equatio1.prg

```

parameter a, b, c
delta = b*b - 4*a*c
If delta < 0
? '方程没有实数解'
endif
if delta = 0
x = -b / (2*a)
? '方程有唯一实数解: ', x
endif
if delta > 0
x1 = (-b - sqrt(delta)) / (2*a)
x2 = (-b + sqrt(delta)) / (2*a)
? '方程有两个实数解: ', x1, x2
endif
Cancel

```

注: Parameter 命令用于说明程序运行所需的参数

注: Cancel 命令用于终止程序

- 注释语句 (P111-112)

为了阅读程序的方便, 可以在程序命令序列中插入一些文字说明。当 FoxProw 运行程序时自动跳过这些注释文字。

例:

```

-----
* 解一元二次方程, 参数: a, b, c
parameter a, b, c
delta = b*b - 4*a*c
If delta < 0          && 如果 b2-4ac < 0
.....

```

注: 文字说明前的标志符 * 和 && 不可缺少

- 状态设置命令 (P112-113)

Set Talk Off 要求 FoxProw 在程序运行时不自动显示反馈信息

Set Talk On 命令恢复 FoxProw 的自动对话功能。

Set Print On 要求程序执行结果同时在打印机上输出

Set Print Off 关闭打印机输出功能（仅在屏幕输出）

注：一般在程序开始部分写 Set Talk Off，在程序结束部分写 Set Talk On。在打印命令前写 Set Print On，在打印后写 Set Print Off。

● 输入输出命令

程序运行中所需的一些数据可以通过命令输入，一些运行结果可以通过输出命令显示。

(1) 键盘输入命令 (P115-116)

Input “输入提示” To 内存变量

例：Input “请输入您的出身日期：” To bd

程序执行到此命令时屏幕输出内容为：

请输入您的出生日期：_

此时用户应在光标后输入数字，然后回车。

例：

```
-----  
* 解一元二次方程，参数：a, b, c  
Input “请输入方程的系数 a : ” To a  
Input “请输入方程的系数 b : ” To b  
Input “请输入方程的系数 c : ” To c  
delta = b*b - 4*a*c  
If delta < 0          && 如果 b2-4ac < 0  
.....  
-----
```

此程序成为不含参数的程序（没有 Parameter 语句），必要的计算参数在程序中通过键盘输入命令输入，而不是在 Do With 命令中输入

注：Input 的数据应该是常量，如：数字，字符串，日期，逻辑值，应注意它们的表示方法。

Accept “输入提示” To 内存变量

此命令的功能与 Input 命令相同，只是必须输入

字符串（不带引号）。

例：

```
-----  
.....  
Accept "请输入被查询人姓名：" To nm  
Locate For name = nm  
.....  
-----
```

用户输入不带引号的姓名字符串并回车后，程序执行查询命令。

(2) 格式输出命令 (P117-119)

格式输出命令规定输出数据在屏幕显示的行列位置。

@ 行号, 列号 **Say** 表达式

例: Say.prg 格式输出程序

```
-----  
Accept "被查询人姓名：" To nm  
Locate For name = nm  
@ 6,10 Say "被查询人姓名：" + name  
@ 7,10 Say "被查询人性别：" + sex  
@ 8,10 Say "被查询人生日：" + Dtoc(birthday)  
@ 9,10 Say "被查询人婚否：" +  
@ 9,24 Say married  
-----
```

注：此程序在当前数据库（staff.dbf）中查找一条记录

注：上例表达式为字符型，应用 + 号与输出字符串连接，

日期应转换为字符串。逻辑型表达式及其输入说明

字符的格式输出应分开写。

注：不能将上例中的 + 号写成 , 号。

(3) 格式输入命令 (P117-119)

格式输入命令允许用户在指定屏幕行列位置编辑数据并完成输入。它由格式编辑与读入数据两部分语句组成：

@ 行号, 列号 **Say** "输入提示" **Get** 变量名 1

@ 行号, 列号 **Say** "输入提示" **Get** 变量名 2

.....

Read

变量名：字段名或程序中已定义过的内存变量名。

程序执行到 @ 语句时，在屏幕指定位置出现输入提示及数据编辑框，显示变量的当前值并等待编辑。用户编辑完数据后按回车键确认，程序执行下面的 Read 命令完成数据读入。

例：Read.dbf 格式输入程序

Append Blank

? "请输入姓名，性别，生日，婚否"

@ 6, 15 Say "姓名：" Get name

@ 7, 15 Say "性别：" Get sex

@ 8, 15 Say "生日：" Get birthday

@ 9, 15 Say "婚否：" Get married

Read

注：此程序在当前数据库（staff.dbf）中添加一条记录

● 运行控制命令

(1) 暂停命令 (P124-125)

Wait "等待提示信息"

程序执行此句命令时，暂停继续执行，屏幕出现等待提示信息，用户回车后程序继续。

例：Wait "输入数据类型错，请回车重新输入"

(2) 程序终止命令 (P126)

Cancel 终止执行程序，返回 FoxProw 命令窗口

Quit 终止执行程序，退出 FoxProw，返回 Windows

第十一讲 选择结构命令

要点：单边选择

 双边选择

 多边选择

选择结构命令用于针对不同情况进行不同的数据处理。

● 单边选择 (P127)

If 条件

 命令序列

Endif

.....

如果条件成立，执行 If 下面的命令序列；如果条件不成立，则执行 Endif 之后的命令。

例：equatio1.prg 解一元二次方程

```
-----  
parameter a, b, c  
delta = b*b - 4*a*c  
If delta < 0                              && delta < 0  
? '方程没有实数解'  
endif  
if delta = 0                              && delta = 0  
    x = -b / (2*a)  
    ? '方程有唯一实数解: ', x  
endif  
if delta > 0                              && delta > 0  
    x1 = (-b - sqrt(delta)) / (2*a)  
    x2 = (-b + sqrt(delta)) / (2*a)  
    ? '方程有两个实数解: ', x1, x2  
endif  
Cancel  
-----
```

注：上例三个条件是互相排斥的。

● 双边选择 (P127-128)

If 条件

 命令序列 1

Else

 命令序列 2

Endif

.....

如果条件成立，执行命令序列 1，然后转到 Endif 之后；若条件不成立，执行命令序列 2，然后转到 Endif 之后。

例：equatio2.prg 解一元二次方程

```
-----  
parameter a, b, c  
delta = b*b - 4*a*c  
If delta < 0                              && delta < 0  
    ? '方程没有实数解'  
Else                                      && delta ≥ 0  
    If delta = 0                         && delta = 0  
        x = -b / (2*a)  
        ? '方程有唯一实数解: ', x  
    Else                                 && delta > 0  
        x1 = (-b - sqrt(delta)) / (2*a)  
        x2 = (-b + sqrt(delta)) / (2*a)  
        ? '方程有两个实数解: ', x1, x2  
    Endif  
Endif  
Cancel  
-----
```

注：在程序中 If 和 Endif 必须成对出现

注：允许在结构语句中嵌套结构语句

例：

Use c:\teacher staff.dbf

Clear

nm=Space(6)

@ 4, 1 Say "

Accept Space(19) + '请输入待修改记录的姓名: ' To nm

Locate For name=nm

If found()

@ 7, 25 Say '姓名: ' Get name

@ 8, 25 Say '性别: ' Get sex

@ 9, 25 Say '生日: ' Get birthday

@ 10, 25 Say '婚否: ' Get married

Read

@ 11, 1 Say "

Accept Space(19) + '已完成记录修改, 请回车退出' To k

Else

@ 6, 1 Say "

Accept Space(19) + '查无此人, 请回车退出' To k

Endif

Close all

Cancel

● 多边选择 (P128-129)

Do Case

Case 条件 1

命令序列 1

Case 条件 2

命令序列 2

.....

Case 条件 n

命令序列 n

Otherwise

命令序列 q

Endcase

.....

当条件 1 成立, 执行命令序列 1, 然后转到 Endcase 之后

当条件 2 成立, 执行命令序列 2, 然后转到 Endcase 之后

.....

当条件 n 成立, 执行命令序列 n, 然后转到 Endcase 之后

若以上条件都不成立, 执行命令序列 q, 然后转到 Endcase 之后

例: equatio3.prg

解一元二次方程


```

-----
parameter  a, b, c
delta = b*b - 4*a*c
Do  Case
    Case  delta<0
        ? '方程没有实数解'
    Case  delta=0
        x = -b / (2*a)
        ? '方程有唯一实数解: ', x
    Case  delta>0
        x1 = (-b - sqrt ( delta )) / ( 2*a )
        x2 = (-b + sqrt ( delta )) / ( 2*a )
        ? '方程有两个实数解: ', x1 , x2
Endcase
Cancel

```

注： 如果不可能出现其它情况，则 **Otherwise** 及其命令序列 **q** 可以省略。

例： Case.prg 多任务处理程序

```

-----
Set default To D:\mysub
Use  satff.dbf
Clear
? “添加记录任务代码： 1 ， 查询记录任务代码： 2”
Input “请输入任务代码： ” To k
?
Do  Case
    Case  k=1
        Do  read.prg  && 调用格式输入添加记录程序
    Case  k=2
        Do  say.prg   && 调用格式输出查询记录程序
    Otherwise
        Wait “输入错误，回车退出”
Endcase
Use

```

Cancel

注：FoxProw 允许在一个程序中调用另一个程序，但不得循环调用。

注：Clear 命令用于清除屏幕原有显示

注：Set Default To 命令用于指定路径缺省值

注：程序结束前应注意关闭数据库

注：? 后面没有表达式，表示屏幕光标下移一行

注：命令 @r,c Say "" 的作用是移动屏幕光标位置

第十二讲 循环结构命令

要点: **Do** 循环

For 循环

Scan 循环

在程序中往往需要重复进行一些相同的数据处理, 例如: 对每条记录做相同的处理, 或对不同参数做相同的计算, 等。使用循环命令可以避免在程序中多次出现相同的命令序列。

- **Do** 循环 (P129-132)

Do While 条件

命令序列

Enddo

只要条件成立便反复执行命令序列。若条件不成立则执行 **Enddo** 后面的命令

Do 与 **Enddo** 之间的命令序列成为循环体。

例:

```
-----  
Use D:\mysub\payment.dbf  
Go top  
Do While Eof() <> .T.  
Replace retired With Year(Date())-Year(birthday) > 60  
Replace total With salary + allowance + award  
Skip  
Enddo  
Use
```

注: 如果疏忽了 **Skip** 命令, 将出现死循环

注: 如果没有 **Go Top**, 则循环从当前记录开始

注: 在程序中 **Do ...** 和 **Enddo** 必须成对出现

例: append.dbf 连续添加记录程序

```
-----  
Set Default To D:\mysub  
Use Staff.dbf  
answer = ""  
Do While lower(answer) <> "n"  
    Clear  
    Do read.prg      && 调用格式输入添加记录程序  
    Accept "继续添加记录请直接回车, 否则键入字母  
          N 再回车 " To answer  
  
Enddo  
Use  
Cancel
```

Loop 与 **Exit** 命令在循环体内的运用:

Loop 返回循环结构的开始, 开始新一轮循环

Exit 跳出循环, 执行 **Enddo** 下面的命令

例:

```
-----  
.....  
Do While .T.  
    Clear  
    Do read.prg  
    Accept "继续添加记录请直接回车, 否则键入字母  
          N 再回车 " To answer  
    If Lower(answer) = "n"  
        Exit  
    Else  
        Loop  
    Endif  
Enddo
```

注: 选择结构可以嵌套在循环结构中, 循环结构也可以嵌

套在选择结构中，但是不能出现交叉嵌套，如：

```
Do ....           If .....
  If ....         Do .....
Enddo             Endif
.....
Endif             Enddo
```

● **For** 循环 (P132-133)

For k = k1 To k2 Step K3

命令序列

Endfor

其中：k 是数值型内存变量名，称为循环参数；k1, k2, k3 可以是数值，或数值表达式，分别称为循环参数初值，循环参数终值，和循环步长。该结构表示对参数等于 k1, k1+k3, 直至 k2, 反复执行循环体命令序列，直至 k 大于 k2 (当 k3 > 0), 或 k < k2 (当 k3 < 0)。

注：当 k3 > 0 时 k2 必须大于等于 k1；当 k3 < 0 时 k2 必须小于等于 k1；否则出现死循环。

例：max_p.prg 求大于等于 3 的正整数以内的最大素数

```
-----
Parameter m
For k = m to 3 step -1    && 逐个检验
  For I = k-1 to 2 step -1 && 是否素数?
    n = k / I
    If n = int (n)        && 存在约数
      exit
    Endif
  Endfor
  If n <> int (n)          && 不存在约数
    ? 'm 以内的最大素数是', k
    exit
  Endif
Endfor
Cancel
```

注：一个循环结构可以嵌套在另一个循环结构内，应注意
不能交叉嵌套

例： @@@@.prg 绘制棱型图案的程序

Clear
r0=10 && 棱形中心行号
h=5 && 棱形半高
r1=r0-h && 最高行行号
r2=r0+h && 最低行行号
c0=40 && 棱形中心列号
d=2 && @ 符号的列间隔
@ r1, c0 Say "@"
c1 = c0
c2 = c0
For r=r1+1 To r0 && 从第 r1 行到第 r0 行
 c1 = c1 - d && 列号 c1 每次减 d
 c2 = c2 + d && 列号 c2 每次加 d
 @ r, c1 Say "@"
 @ r, c2 Say "@"
Endfor
For r=r0+1 To r2 && 从第 r0+1 行到第 r2 行
 c1 = c1 + d && 列号 c1 每次加 d
 c2 = c2 - d && 列号 c2 每次减 d
 @ r, c1 Say "@"
 @ r, c2 Say "@"
Endfor
Cancel

注：如果步长等于 1，则可省略 Step 子句

● Scan 循环 (P133-135)

Scan 范围子句 条件子句

命令序列

Endscan

在指定范围中扫描记录，对符合条件的记录执行命令序列。

例：Check.prg 核对并修改关联数据库数据

```
-----  
Clear  
Set Default To D:\mysub  
Select 1  
Use staff.dbf  
Select 2  
Use address.dbf  
Set Order To Tag name  
Select 3  
Use payment.dbf  
Set Order To Tag name  
Select staff  
Set Relation To    name Into address , name Into payment  
@ 4, 10 Say "请核对并修改数据"  
Scan  
    @ 6, 10 Say "姓名: " Get name  
    @ 7, 10 Say "职务: " Get payment.title  
    @ 8, 10 Say "电话: " Get address.telephone  
Read  
Endscan  
Close all  
Cancel  
-----
```

注：在 Scan 命令的循环体中不需要用 Skip 移动指针

注：Close all 命令可以在程序结束前关闭所有文件