



万水 Visual FoxPro 5.0 从入门到精通系列

# Visual FoxPro 5.0 编程宝典 中文版

周予滨 主编

 中国水利水电出版社

3

1 ZYB/1

万水 Visual FoxPro 5.0 中文版从入门到精通系列

# Visual FoxPro 5.0 中文版编程宝典

周予滨

主编

冯 峰 徐和钦

编著

刘海涛 陈永智

中国水利水电出版社

## 内 容 提 要

Visual FoxPro 是目前应用最广泛的桌面数据库系统,而中文版 5.0 则是其最新的版本。这是一个真正 32 位的微机版数据库系统开发工具,特别是它提供了数据库系统的 WEB 功能,使它成为数据库软件的新宠。

本书由浅入深地介绍了 Visual FoxPro 5.0 中文版编程技术,并附有大量图片和应用实例,不仅是初学者的必备参考书,即使对高级开发人员,本书也会在提高编程技巧方面有所启发。

## 图书在版编目 ( CIP ) 数据

Visual FoxPro 5.0 中文版编程宝典 / 周予滨主编. —北京: 中国水利水电出版社, 1998.4

(万水 Visual FoxPro 5.0 从入门到精通系列)

ISBN 7-80124-717-5

I. V… II. 周… III. 关系数据库-数据库管理系统-Foxpro5.0-程序设计 IV. TP311.13

中国版本图书馆 CIP 数据核字 ( 98 ) 第 09645 号

|       |                                                                       |  |
|-------|-----------------------------------------------------------------------|--|
| 书 名   | Visual FoxPro 5.0 中文版编程宝典                                             |  |
| 作 者   | 周予滨 李腾等                                                               |  |
| 审 校   | 齐侨创作室                                                                 |  |
| 出版、发行 | 中国水利水电出版社 (北京市三里河路 6 号 100044)<br>北京万水电子信息有限公司 (北京市车公庄西路 20 号 100044) |  |
| 排 版   | 北京万水电子信息有限公司                                                          |  |
| 印 刷   | 水利电力出版社印刷厂印刷                                                          |  |
| 规 格   | 787×1092 毫米 16 开本 24.75 印张 554 千字                                     |  |
| 版 次   | 1998 年 6 月第一版 1998 年 6 月北京第一次印刷                                       |  |
| 印 数   | 0001 — 5000 册                                                         |  |
| 定 价   | 36.00 元                                                               |  |

# 前 言

Visual FoxPro 同 Visual Basic、Visual C++、Visual J++一起组成了 Microsoft 公司 Visual 系列开发工具，并与其它一些开发工具一起以 Visual Developer Studio 的形式推向市场，获得了极大的成功。Visual FoxPro 5.0 则是 FoxPro 系列中的最新版本，这是一个真正 32 位的微机版数据库管理系统开发工具。它不仅兼容 FoxPro 系列的所有前期版本，又比 3.0 版本在面向对象编程方面有了新的改进，更令人兴奋的是 Visual FoxPro 5.0 紧跟时代潮流，提供了数据库系统的 Web 功能，使得开发出的管理系统可以方便、迅速地实现在 Internet 和 Intranet 上的应用。所有这些优点都使得 Visual FoxPro 5.0 成为数据库软件中的新宠。

为了方便中国用户学习、掌握、运用这一最新的数据库开发工具，我们基于 Visual FoxPro 5.0 中文版编写了这套丛书，旨在全面、详尽的介绍软件功能与操作方法，同时又注意满足高级开发人员的需求，着重介绍了面向对象编程这一重要部分。书中提供了大量的应用实例，可以帮助读者迅速掌握这一软件。

本套丛书由周予滨和徐臻青组织策划，李腾统稿，文字上力图通俗易懂、细致全面，尽可能满足不同层次的读者需要。包括：

《 Visual FoxPro 5.0 中文版使用指南》

《 Visual FoxPro 5.0 中文版编程宝典》

《 Visual FoxPro 5.0 中文版命令与函数大全》

《 Visual FoxPro 5.0 中文版类库对象大全》

本书为《 Visual FoxPro 5.0 中文版编程宝典》。循序渐进、由浅入深地讲述了 Visual Foxpro 中的编程技术，在介绍了 Visual Foxpro 编程简介后，第二章从程序结构讲起，使用户了解程序的组成，初步掌握程序设计的步骤。在第三章中介绍 Visual Foxpro 的数据库操作命令，在程序设计中要经常用到这些命令。第四章介绍了 Visual Foxpro 的常用函数的用法。Visual Foxpro 支持面向对象的程序设计方法，利用面向对象的程序设计方法可以编制出功能强大、灵活多变的应用程序。在讲述了基本的程序设计知识后，介绍面向对象的程序设计（ OOP ）方法，然后综合前面所讲述的程序设计技术，通过实例讲述用面向对象的方法快速创建应用程序。最后介绍了通过对象链接与嵌入（ OLE ）实现 Visual Foxpro 与其他应用程序交流的方法以及网络环境下的开发技术。

参加本书编写的还有杨彬、董渊、王翔、顾克勤等人。感谢李浩、郭阳提供了大量应用实例。

由于时间仓促，加上水平有限，书中的错误与疏漏，恳请读者批评指正。

# 目 录

|                                       |    |
|---------------------------------------|----|
| 第一章 Visual Foxpro 编程简介 .....          | 1  |
| 1.1 为什么要编程 .....                      | 1  |
| 1.2 VisualFoxpro 编程的特点 .....          | 2  |
| 1.3 Visual Foxpro 中设计程序的步骤 .....      | 5  |
| 1.4 本书特色和内容提要 .....                   | 6  |
| 第二章 Visual Foxpro 的程序结构 .....         | 8  |
| 2.1 数据的输入输出命令 .....                   | 8  |
| 2.1.1 ? /? ? .....                    | 8  |
| 2.1.2 wait .....                      | 10 |
| 2.1.3 TextBox 控件 .....                | 11 |
| 2.2 内存变量 .....                        | 11 |
| 2.2.1 内存变量的声明和赋值 .....                | 11 |
| 2.2.2 数组的声明和赋值 .....                  | 12 |
| 2.2.3 内存变量的操作命令 .....                 | 12 |
| 2.2.4 系统内存变量的操作使用 .....               | 18 |
| 2.3 程序控制结构 .....                      | 18 |
| 2.3.1 分支判断结构: IF...ELSE...ENDIF ..... | 18 |
| 2.3.2 多重判断结构 .....                    | 19 |
| 2.3.3 循环结构 .....                      | 20 |
| 2.4 命令文件的建立与执行 .....                  | 22 |
| 2.4.1 命令文件的建立与编辑 .....                | 23 |
| 2.4.2 命令文件的执行 .....                   | 24 |
| 2.5 过程和自定义函数 .....                    | 24 |
| 2.5.1 过程和函数概述 .....                   | 24 |
| 2.5.2 过程和函数的执行 .....                  | 25 |
| 2.6 项目管理器的使用 .....                    | 27 |
| 第三章 FoxPro 的数据库操作命令 .....             | 31 |
| 3.1 数据库的操作命令 .....                    | 33 |
| 3.1.1 创建数据库 .....                     | 33 |
| 3.1.2 打开数据库 .....                     | 34 |
| 3.1.3 设定当前数据库 .....                   | 36 |
| 3.1.4 关闭数据库 .....                     | 37 |

|            |                              |           |
|------------|------------------------------|-----------|
| 3.1.5      | 删除数据库 .....                  | 37        |
| 3.2        | 数据表的操作命令 .....               | 38        |
| 3.2.1      | 创建数据表 .....                  | 39        |
| 3.2.2      | 将数据表加入到数据库 .....             | 40        |
| 3.2.3      | 将数据表移出数据库 .....              | 41        |
| 3.2.4      | 打开数据表 .....                  | 43        |
| 3.2.5      | 关闭数据表 .....                  | 46        |
| 3.2.6      | 删除数据表 .....                  | 47        |
| 3.2.7      | 显示数据表的字段定义 .....             | 48        |
| 3.2.8      | 修改数据表的定义 .....               | 49        |
| 3.3        | 记录数据的操作命令 .....              | 52        |
| 3.3.1      | 记录的显示 .....                  | 52        |
| 3.3.2      | 记录的定位 .....                  | 53        |
| 3.3.3      | 记录的查找 .....                  | 55        |
| 3.3.4      | 记录的增加 .....                  | 58        |
| 3.3.5      | 记录的修改 .....                  | 60        |
| 3.3.6      | 记录的编辑 .....                  | 61        |
| 3.3.7      | 记录的删除 .....                  | 63        |
| 3.3.8      | 记录的索引 .....                  | 67        |
| 3.3.9      | 记录的筛选 .....                  | 70        |
| 3.3.10     | 记录的查询 .....                  | 71        |
| 3.3.11     | 记录数据的统计计算 .....              | 73        |
| 3.4        | 数据库操作命令的应用 .....             | 77        |
| 3.5        | 总结 .....                     | 82        |
| <b>第四章</b> | <b>FoxPro 的常用函数的用法 .....</b> | <b>84</b> |
| 4.1        | 函数的使用方法 .....                | 84        |
| 4.2        | 字符处理函数 .....                 | 85        |
| 4.2.1      | 去掉字符表达式中的头尾空格 .....          | 85        |
| 4.2.2      | 查找字符函数 .....                 | 86        |
| 4.2.3      | 替换字符函数 .....                 | 88        |
| 4.2.4      | 截取字符表达式中的一段 .....            | 88        |
| 4.2.5      | 计算字符表达式长度 .....              | 89        |
| 4.2.6      | 判断字符类型的函数 .....              | 90        |
| 4.2.7      | 判断字符大小写的函数 .....             | 90        |
| 4.2.8      | 字符大小写转换函数 .....              | 91        |
| 4.2.9      | 其他关于字符的函数 .....              | 91        |
| 4.3        | 数值处理函数 .....                 | 92        |
| 4.3.1      | 三角函数 .....                   | 93        |

|            |                                     |            |
|------------|-------------------------------------|------------|
| 4.3.2      | 反三角函数 .....                         | 93         |
| 4.3.3      | 指数函数 .....                          | 94         |
| 4.3.4      | 对数函数 .....                          | 95         |
| 4.3.5      | 截断函数 .....                          | 95         |
| 4.3.6      | 绝对值函数 .....                         | 97         |
| 4.3.7      | 最大最小值函数 .....                       | 97         |
| 4.3.8      | 求余函数 .....                          | 98         |
| 4.3.9      | 其他常用函数 .....                        | 98         |
| 4.4        | 日期和时间处理函数 .....                     | 99         |
| 4.4.1      | 系统日期时间 .....                        | 100        |
| 4.4.2      | 日期时间函数 .....                        | 100        |
| 4.4.3      | 转换函数 .....                          | 104        |
| 4.5        | 其他常用函数 .....                        | 107        |
| 4.6        | 总结 .....                            | 111        |
| <b>第五章</b> | <b>Visual FoxPro 5.0 的调试器 .....</b> | <b>114</b> |
| 5.1        | FoxPro 最新版本的调试器 .....               | 114        |
| 5.2        | 调试器菜单 .....                         | 119        |
| 5.2.1      | 文件菜单 .....                          | 119        |
| 5.2.2      | 编辑菜单 .....                          | 120        |
| 5.2.3      | 调试菜单 .....                          | 123        |
| 5.2.4      | 工具菜单 .....                          | 130        |
| 5.2.5      | 窗口菜单 .....                          | 136        |
| 5.2.6      | 帮助菜单 .....                          | 137        |
| 5.3        | 调试器的工具栏、快捷键和快捷菜单 .....              | 138        |
| 5.3.1      | 调试器的工具栏 .....                       | 138        |
| 5.3.2      | 调试器的快捷键 .....                       | 139        |
| 5.3.3      | 调试器的快捷菜单 .....                      | 140        |
| 5.4        | 调试器窗口 .....                         | 143        |
| 5.4.1      | “调用堆栈”窗口 .....                      | 143        |
| 5.4.2      | “调试输出”窗口 .....                      | 144        |
| 5.4.3      | “局部”窗口 .....                        | 146        |
| 5.4.4      | “跟踪”窗口 .....                        | 146        |
| 5.4.5      | “监视”窗口 .....                        | 148        |
| <b>第六章</b> | <b>发挥面向对象的威力 .....</b>              | <b>149</b> |
| 6.1        | OOP 简介 .....                        | 149        |
| 6.1.1      | OOP 术语快速入门 .....                    | 149        |
| 6.1.2      | OOP 的优点 .....                       | 153        |

|                                   |            |
|-----------------------------------|------------|
| 6.2 可视类 .....                     | 154        |
| 6.2.1 通过表单设计器 .....               | 155        |
| 6.2.2 完全通过编程的方法 .....             | 176        |
| 6.3 不可视类 .....                    | 184        |
| 6.4 Visual FoxPro 类的层次 .....      | 187        |
| 6.5 类库操作入门 .....                  | 189        |
| 6.5.1 使用类设计器 .....                | 189        |
| 6.5.2 使用类浏览器 .....                | 191        |
| <b>第七章 用面向对象的方法快速创建应用程序 .....</b> | <b>198</b> |
| 7.1 开发应用程序的过程 .....               | 198        |
| 7.1.1 需求分析 .....                  | 198        |
| 7.1.2 数据库组织 .....                 | 199        |
| 7.2 创建应用程序的骨架——事件循环 .....         | 201        |
| 7.2.1 菜单系统 .....                  | 201        |
| 7.2.2 事件循环(事件驱动) .....            | 204        |
| 7.3 创建应用程序的血肉——表单和报表 .....        | 205        |
| 7.3.1 创建基类 .....                  | 205        |
| 7.3.2 创建表单 .....                  | 211        |
| 7.4 创建报表 .....                    | 221        |
| 7.4.1 创建一对一报表 .....               | 221        |
| 7.4.2 创建一对多报表 .....               | 223        |
| 7.4.3 创建分组/总计报表 .....             | 224        |
| 7.5 创建应用程序的皮肤——完善 .....           | 225        |
| 7.5.1 项目管理器 .....                 | 225        |
| 7.5.2 “项目管理器”按钮 .....             | 226        |
| 7.5.3 将自己的文件加入到项目中 .....          | 227        |
| 7.5.4 为一个项目建立应用程序 .....           | 228        |
| 7.6 小结 .....                      | 229        |
| <b>第八章 高级技术 OLE ActiveX .....</b> | <b>230</b> |
| 8.1 Activex 控件的使用 .....           | 230        |
| 8.1.1 ImageList 控件 .....          | 232        |
| 8.1.2 ProgressBar 控件 .....        | 237        |
| 8.1.3 Slider 控件 .....             | 240        |
| 8.1.4 StatusBar(状态条)控件 .....      | 242        |
| 8.1.5 Toolbar 控件 .....            | 248        |
| 8.1.6 CommonDialog 控件 .....       | 259        |
| 8.1.7 SysInfo 控件 .....            | 269        |

|            |                                       |            |
|------------|---------------------------------------|------------|
| 8.1.8      | TreeView 控件 .....                     | 269        |
| 8.1.9      | Multimedia MCI 控件 .....               | 283        |
| 8.1.10     | HWND 控件 .....                         | 285        |
| 8.1.11     | RichTextBox 控件 .....                  | 291        |
| 8.1.12     | Grid 控件 .....                         | 293        |
| 8.1.13     | TabStrip 控件 .....                     | 299        |
| 8.1.14     | Visual FoxPro Foxtlib 控件 .....        | 303        |
| 8.1.15     | Calendar 控件 .....                     | 306        |
| 8.1.16     | PictureClip 控件 .....                  | 308        |
| 8.1.17     | MAPISession 控件和 MAPIMessages 控件 ..... | 309        |
| 8.1.18     | MSComm 控件 .....                       | 322        |
| 8.2        | OLE (自动化技术) Automation .....          | 328        |
| 8.2.1      | 链接或嵌入 OLE 对象 .....                    | 328        |
| 8.2.2      | 创建服务程序 .....                          | 329        |
| 8.2.3      | 编译服务程序 .....                          | 330        |
| 8.2.4      | 使用 OLE 服务程序 .....                     | 332        |
| 8.2.5      | 使用远程自动化 .....                         | 333        |
| 8.2.6      | 服务器的配置 .....                          | 333        |
| 8.2.7      | 配置客户计算机 .....                         | 335        |
| 8.2.8      | OLE 自动化服务器的建立 (例程和说明) .....           | 335        |
| 8.3        | 小结 .....                              | 340        |
| <b>第九章</b> | <b>网络环境下的数据库开发 .....</b>              | <b>341</b> |
| 9.1        | 多用户管理 .....                           | 341        |
| 9.1.1      | 多用户环境所必须考虑的问题——并发控制 .....             | 341        |
| 9.1.2      | 加密算法 .....                            | 357        |
| 9.2        | 客户/服务器模式 .....                        | 358        |
| 9.2.1      | 什么是客户/服务器模式 .....                     | 359        |
| 9.2.2      | 如何访问服务器的数据 .....                      | 359        |
| 9.2.3      | 示例 s .....                            | 364        |
| 9.3        | 使用升迁向导 .....                          | 377        |
| 9.3.1      | Oracle 升迁向导 .....                     | 378        |
| 9.4        | 小结 .....                              | 384        |

# 第一章 Visual FoxPro 编程简介

## 1.1 为什么要编程

Visual FoxPro 将面向过程的程序设计与面向对象的程序设计结合在一起，用户可以用它创建出功能强大、灵活多变的应用程序。程序就是为了完成某一具体任务而编写一系列指令。

通常，在程序中能够完成的工作都可以通过人工操作来完成，那么为什么还要编程呢？请看下面的例子对比。例如，某公司要在雇员“employee”表中查找一个特定雇员（陈智勇）的信息，可以通过人工执行一系列指令来实现。

人工操作步骤：

- (1) 选择“文件”菜单中的“打开”命令。
- (2) 在弹出的“打开”对话框的“文件类型”列表框中选择“表”。
- (3) 打开相应的表。
- (4) 在“显示”菜单中选择“浏览”命令。
- (5) 滚动浏览表，查找 employee 字段为“陈智勇”的记录。

若要通过编程实现上述目的，可以在“命令”窗口中键入下列命令：

```
USE ...\\...\\employee
LOCATE FOR employee= "陈智勇"
BROWSE
```

在表中找到这个雇员后，就可以执行特定的操作（如将工资提高百分之十）。

若要通过人工操作提高该雇员的工资，步骤如下：

- (1) 选定 salary 字段。
- (2) 将 salary 字段值乘以 0.1，再加上原来的字段值，然后在该字段处键入得到的结果。

若要通过编程实现以上结果，可以在“命令”窗口中键入以下命令：

```
REPLACE salary WITH salary * 1.1
```

如果要修改某个雇员的工资，用人工方式或在“命令”窗口中键入指令都很容易实现，但如果要把所有的雇员的工资都提高百分之十，恐怕就不是一件轻而易举的事情。若还是用人工方式来做，不但费时费力，而且还容易出错。解决此类问题的更好办法是编写一个可执行的程序文件，用该程序可以轻松无误地完成这一工作。

增加所有雇员工资的示例程序：

代码

注释

```
USE ...\\...\\employee
SCAN
```

打开 employee 表。

浏览表中所有记录，针对每条记录执行  
SCAN 与 ENDSCAN 之间的所有指令。

REPLACE salary WITH ;

salary \* 1.1

ENDSCAN

将工资增加百分之十。(分号;)表示续行。)

结束由 SCAN 开始、针对表中所有记录执行的指令序列。

和“命令”窗口中单独键入每条命令相比，运行程序有如下优点：

程序可被修改并重新运行。

可从菜单、表单和工具栏启动程序。

一个程序可调用其他程序。

## 1.2 VisualFoxPro 编程的特点

大多数编程语言具有相似的基本特征。如果熟悉一种编程语言就可以很容易地将这种编程语言的知识应用到其它编程语言中。下面讲述 Visual FoxPro 同几种常用编程语言的比较。

(1) Visual FoxPro 编程中不区分大小写。与流行的 C/C++ 不同，Visual FoxPro 编程中不区分大小写，但为了程序易读易维护，建议用户对标识符使用固定的大小写规则。

(2) Visual FoxPro 不需要明确声明变量。当把值保存到变量而该变量不存在时，Visual FoxPro 隐含地声明它。这种情况对于需要明确声明变量的编程语言来说会产生错误。

(3) 在 Visual FoxPro 中，不能为变量指定数据类型（虽然在命名变量时建议您使用表示其数据类型的前缀）。其他语言在声明变量时，需要为其指定数据类型。

(4) 每种编程语言都有它自己的在程序中添加注释的格式。Visual FoxPro 的注释格式是：

|      |        |
|------|--------|
| *    | 整行注释   |
| USE  | && 结束行 |
| NOTE | 多行注释   |

(5) 赋值语句。Visual FoxPro 中的赋值方式是：nVal = 1 或者 STORE 1 to nVal。在 Visual FoxPro 中，也可以使用 REPLACE 命令给表中的字段赋值。

设计程序时，就是用一系列指令存储数据并操作这些数据。程序设计的原材料是数据和数据的存储容器，而处理这些原材料的工具是命令、函数和操作符。

(6) 存储数据。用户使用的数据可能包括时间、货币数量以及日期、名称和说明等。每个数据都有其数据类型，属于同一类型的数据可以按相似的方法进行处理。您当然可以直接处理数据而不加以存储，但这样做会失去很大的灵活性，而且 Visual FoxPro 提供的许多功能没有发挥作用。为增强处理数据的能力，Visual FoxPro 提供了多种数据存储容器。

(7) 数据类型。数据类型决定了数据的存储方式和使用方式。两个实数可以做乘法运算，但两个字符型数据不能做乘法运算。同样道理，字符可以用大写方式打印，而数字就不存在大小写的问题。下表列出了 Visual FoxPro 的主要数据类型。

| 数据类型 | 类型示例           |
|------|----------------|
| 数值型  | 123 .31415 -71 |

|       |                        |                |
|-------|------------------------|----------------|
| 字符型   | "West String"          | "23" "1/01/95" |
| 逻辑型   | .T. .F.                |                |
| 日期型   | {02/01/98}             |                |
| 日期时间型 | {01/01/98 12:32:00 am} |                |

(8) 数据容器。数据容器允许在多个数据上进行相同的操作。例如, 将一个雇员工作的小时数加起来, 再乘以每小时工作应付的工资, 扣除税款后, 便可知道一个雇员应得的报酬。若对每个雇员都进行这样的操作将非常麻烦, 但若将这些信息保存在数据容器中并对数据容器进行操作, 那么, 通过运行程序就能实现数据的更新。下面列出了 Visual FoxPro 中主要的数据容器:

| 数据容器的类型 | 说明                               |
|---------|----------------------------------|
| 变量      | 在随机存储器 (RAM) 中的单个数据元素            |
| 表记录     | 多行预定义字段, 每个字段包含一条预定义数据, 表存储在磁盘上。 |
| 数组      | 随机存储器中的多元素数据。                    |

(9) 处理数据。数据容器和数据类型构成了处理数据的基础, 而对数据的处理最终要通过操作符、函数和命令来实现。

1) 使用操作符。Visual FoxPro 中最常用的操作符如下:

=、+、! 或 NOT、\*、/

注意, 操作符应与数据类型相匹配。下面的语句将两个数值赋给两个变量, 因为变量名的首写字母为 n, 所以可马上知道该变量保存的是数值型数据, 但这样的规定只是约定的, 实际上, 您可以使用字母、数字及下划线的任意组合为变量命名。

```
nSalary = 123
```

```
nBonus = 456
```

下面的语句将两个字符串赋值给两个变量, 变量名首写字母 c 表明该变量保存的是字符型数据。

```
cName = "Tom"
```

```
cCompany = "Macrohard"
```

因为 cName 是字符型数据, 而 nBonus 是数值型数据, 所以执行下面操作时将出现数据类型不匹配的错误:

```
? cName + nBonus
```

2) 使用函数。函数可返回特定类型的数据。可以通过这些函数所带的文档来了解其返回值类型。

可以用以下五种方法调用 Visual FoxPro 函数:

a. 将函数的返回值赋给某个变量。

下面这行代码使用变量 dToday 保存当前系统日期。

```
dToday = DATE()
```

b. 在 Visual FoxPro 命令中包含函数调用。

下面的命令使用 GETDIR() 函数的返回值设置默认路径。

```
CD GETDIR()
```

c.在活动输出窗口中输出返回值。

下面这行代码在 Visual FoxPro 主窗口中输出当前系统时间。

? TIME()

d.调用函数但不保存其返回值。

下面的函数调用将关闭临时表。

SYS(2002)

e.函数嵌套。

下面这行程序输出今天是星期几。

? DOW( DATE( ) )

3) 使用命令。一条命令即完成一个特定动作的指令。每条命令都有自己特定的语法,用来说明为实现该命令的功能所必须包含的东西。此外,与命令有关的还有一些可选子句,这些关键字可进一步指导命令干些什么。

例如, USE 命令可以打开表或关闭表:

USE 语法

说明

USE

关闭当前工作区中的表。

USE employee

在当前工作区中打开 employee 表,同时关闭当前工作区中所有已打开的表。

USE employee IN 0

在下一个可用工作区中打开 employee 表。

USE employee IN 0 ALIAS sonny

在下一个可用工作区中打开 employee 表,并且指定该工作区的别名为 sonny。

(10) 程序流的控制。Visual FoxPro 中有一类特殊的命令,它们可以反复执行其他命令或函数,并决定这些命令或函数何时执行以及执行次数。这类特殊命令可用来实现两种主要的程序结构:条件分支和循环,它们在程序设计过程中作用很大。将在下一章详细讲述这些命令。

掌握了一些基本概念后,程序设计就变成了一个不断重复的过程。您需要多次重复某些步骤,并在这个过程中不断对代码进行优化。从编写第一行程序开始,便不断进行测试,完成每个“试验—查错”的过程。对语言越熟悉,则编程速度越快,而且更多的初步测试工作将在头脑中进行。

程序设计的基本步骤包括:对问题进行说明;分解问题;编制各模块;测试并完善各模块;组装全部模块;整体测试。

开始程序设计之前,请注意以下几个问题:

(1) 在解决问题之前,必须把问题说明清楚,否则会不断进行修改,丢弃已编好的代码并从头再来,而且最终也不可能得到满意的结果。

(2) 将问题分解成可单独处理的几个步骤,而不是一下子解决全部问题。

(3) 在开发过程中不断测试和调试已编好的代码。通过测试检查代码是否能实现所需的功能;调试是找出代码在哪里出错并纠正这些错误。

(4) 精炼数据和数据存储方式,便于程序对其进行处理。这需要正确构造表格的结构。

## 1.3 Visual FoxPro 中设计程序的步骤

Visual FoxPro 程序由代码组成, 代码包括以命令形式出现的指令、函数或 Visual FoxPro 可以理解的任何操作。这些指令可以包含在: “命令”窗口中、程序文件中、“表单设计器”或“类设计器”的事件或方法程序代码窗口中、“菜单设计器”的过程代码窗口中, 或者“报表设计器”的过程代码窗口中。下面分别介绍:

### 1. 在“命令”窗口中的代码指令

可以在“命令”窗口中键入 Visual FoxPro 命令并按 ENTER 键执行。若要重新执行该命令, 还可以将光标移到此命令所在行并按 ENTER 键。也可以在“命令”窗口中将多行代码象独立程序一样执行, 方法是: 选择要运行的代码行, 然后按 ENTER 键。因为“命令”窗口是一个编辑窗口, 所以在编辑命令时可以使用 Visual FoxPro 提供的编辑工具。在“命令”窗口中可以编辑、插入、删除、剪切、复制和粘贴命令正文。在“命令”窗口中执行命令的优点是: 能够立即执行被键入的命令, 不需要将其保存为文件。

此外, 在菜单和对话框中所作的选择可以马上转化为“命令”窗口中的命令。用户可以将这些命令复制并粘贴至 Visual FoxPro 程序, 然后重复执行这些程序。这样做很容易重复执行成百上千条的命令。

### 2. 编制程序

Visual FoxPro 程序是包含一系列命令的文本文件。在 Visual FoxPro 中, 可以通过以下步骤建立程序:

- (1) 在“项目管理器”中, 选择“代码”标签中的“程序”项。
- (2) 选择“新建”命令。

也可以按以下步骤进行:

- (1) 在“文件”菜单中选择“新建”命令。
- (2) 在弹出的“新建”对话框中选择“程序”。
- (3) 选择“新建文件”按钮。

还可以这样做:

在“命令”窗口中键入“MODIFY COMMAND”, Visual FoxPro 就会打开一个称为“程序 1”的新窗口, 这时就可以键入编写的应用程序了。

### 3. 保存程序

程序编写好并输入后, 请注意保存。若要保存程序, 可以通过在“文件”菜单中选择“保存”命令来保存程序文件。若用户要关闭一个没有保存的程序, 则会弹出相应对话框, 提示用户是保存还是放弃已做出的修改。若用户保存了一个由“项目管理器”创建的程序则保存的程序被加入到当前项目中。若用户保存一个尚未命名的程序, 则会打开“另存为...”对话框, 提示用户为程序指定程序名。程序保存后, 用户可运行或修改它。

#### 4. 修改程序

程序创建后可以修改。要修改程序，先要打开程序文件。若要打开程序文件，按下述方法进行操作：

若程序包含在某个项目中，则在“项目管理器”中选择它，然后选择“修改”命令。也可以在“文件”菜单中选择“打开”命令，弹出“打开”对话框。在“文件类型”列表框中选择“程序”，然后在“文件名”列表中选择要修改的程序的文件，单击“确定”按钮，就可以打开要修改的程序文件了。此外，在“命令”窗口中键入以下命令，也可以打开要修改的程序文件：

**MODIFY COMMAND program-file-name**

如果记不清程序文件的名字，可以在“命令”窗口中键入：

**MODIFY COMMAND ?**

可以弹出“打开”对话框，然后在文件列表中选择要修改的程序，单击“打开”按钮打开要修改的文件。

打开文件之后便可进行修改，修改完毕后不要忘记保存。

#### 5. 运行程序

程序创建完毕后便可以运行。运行程序的步骤是：

若程序包含在一个项目中，则在“项目管理器”中选择该程序，然后单击“运行”按钮。或者在“程序”菜单中选择“运行”菜单项。然后在弹出的程序列表中，选择想要运行的程序，单击“运行”按钮。也可以在“命令”窗口中键入以下命令：

**DO program-file-name**

就可以运行程序了。

#### 6. 使用 Visual FoxPro 设计工具编写代码

通过“表单设计器”、“类设计器”和“菜单设计器”，程序员可以很容易地把程序代码与用户界面连接起来，这样应用程序便可响应用户的输入并执行相应的代码。同样，“报表设计器”将程序代码与报表文件联系起来，以此定制结构复杂并且符合用户要求的报表。

使用上面提到的设计工具可以充分发挥 Visual FoxPro 的强大功能。

## 1.4 本书特色和内容提要

本书循序渐进，由浅入深地讲述了 Visual FoxPro 中的编程技术，在介绍了 Visual FoxPro 编程简介后，第二章从程序结构讲起，使用户了解程序的组成，初步掌握程序设计的步骤。然后在第三章中介绍 Visual FoxPro 的数据库操作命令，在程序设计中要经常用到这些命令。上面讲过，在 Visual FoxPro 的程序设计中还可以使用函数，第四章介绍了 Visual FoxPro 的常用函数的用法。Visual FoxPro 支持面向对象的程序设计方法，利用面向对象的程序设计方法可以编制出功能强大、灵活多变的应用程序。在讲述了基本的程序设计知识后，本

书将介绍面向对象的程序设计（ OOP ）方法，是用户编制的程序可以发挥面向对象的威力。这之后将综合前面所讲述的程序设计技术，通过实例讲述用面向对象的方法快速创建应用程序。最后还要介绍通过对象链接与嵌入（ OLE ）实现 Visual FoxPro 与其他应用程序交流的方法以及网络环境下的开发技术。

程序的开发是离不开测试与调试的， Visual FoxPro5.0 提供了一个强大的调试工具，通过它，用户可以轻松地调试自己的程序，迅速排除错误，优化代码。本书详细讲述了该调试器的使用。

## 第二章 Visual FoxPro 的程序结构

同其他程序设计语言一样，Visual FoxPro 的编程语言中也有专门控制程序流程的语句。它们可以反复执行其他命令或函数，并决定这些命令或函数何时执行以及执行次数。这类特殊命令可用来实现两种主要的程序结构：条件分支和循环，它们在程序设计过程中作用很大。为了更好地理解示例程序，先讲述几个最常用的命令。

### 2.1 数据的输入输出命令

#### 2.1.1 ???

这两个命令求出表达式的值并显示表达式的结果，使用的语法是：

?|?? Expression1

[**PICTURE** cFormatCodes] | [**FUNCTION** cFormatCodes] | [VnWidth]

[**AT** nColumn]

[**FONT** cFontName [, nFontSize] [**STYLE** cFontStyle | Expression2]]

[, Expression3] ...

参数为：

? Expression1 求出表达式 Expression1 的值并在表达式的结果前插入回车换行符。结果显示在 Visual FoxPro 主窗口或者用户定义的活动窗口的下一行，并且靠窗口的左边显示。除非功能代码 cFormatCodes 或者系统变量 `_ALIGNMENT` 指定别的显示方式。

如果省略表达式，则显示一个空行。当包括多个表达式时，在显示的每个表达式之间加入一个空格。

?? Expression1 求出表达式 Expression1 的值并把表达式的结果显示在 Visual FoxPro 主窗口或者用户定义的活动窗口，或者打印机的当前行的当前位置。在表达式的结果前不插入回车换行符。

**PICTURE** cFormatCodes 指定显示 Expression1 结果所使用的图形格式。cFormatCodes 可以包含功能代码、图形代码，或者两者都包括。用户可以使用与 **Format** 和 **InputMask** 属性中可用的相同的代码。

功能代码影响结果的整个格式，而图形代码则只影响结果中的单个字符。如果在 cFormatCodes 中使用功能代码，则它们必须出现在图形代码的前面。而且它们的前面必须有一@符号。无内嵌空格的多个功能代码可以直接跟随在一个@符号后面。最后一个功能代码后面必须有一个或者多个空格。这些空格标志着功能代码的结束和图形代码的开始。

**FUNCTION** cFormatCodes 指定? 和 ??输出中的功能代码。如果包含功能子句，则不要在功能代码前键入@符号。在功能代码包含在 **PICTURE** 中时前面必须有一个@符号。

VnWidth 指定一特殊的功能代码，这个代码使字符表达式的结果垂直拉伸有限列。