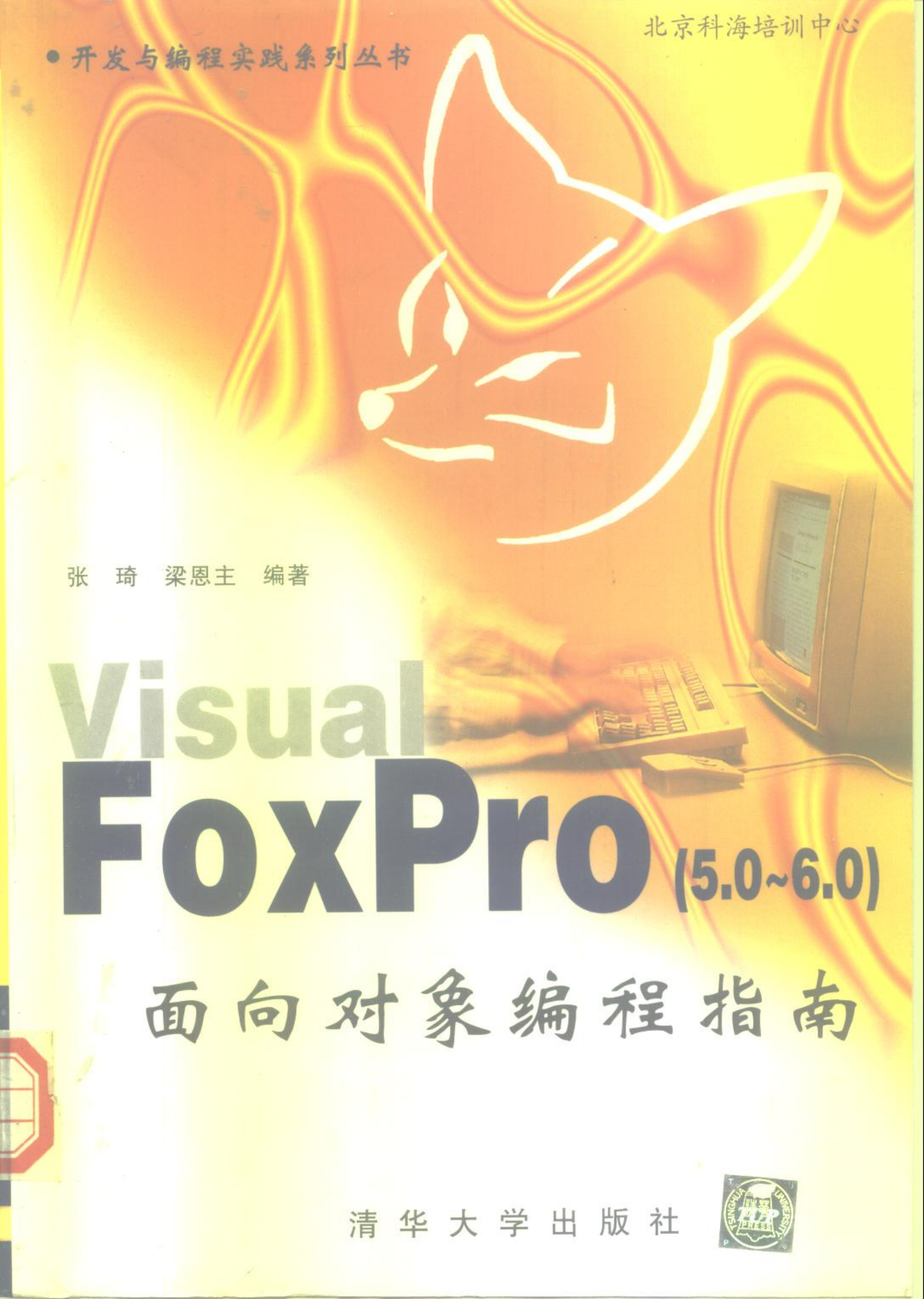


北京科海培训中心

• 开发与编程实践系列丛书

张 琦 梁恩主 编著



Visual FoxPro (5.0~6.0)

面向对象编程指南

清华大学出版社



北京科海培训中心

• 开发与编程实践系列丛书

Visual FoxPro(5.0~6.0) 面向对象编程指南

张琦 梁恩主 编著

清华大学出版社

(京)新登字 158 号

内 容 提 要

本书以详实的内容,丰富的实例全面介绍了 Visual FoxPro 5.0~6.0 应用程序的设计、开发过程。

全书兼顾 Visual FoxPro 的应用及系统开发两个方面,系统地介绍了数据库的实用操作、应用软件的分析、设计过程,并且对 VFP 新增的面向对象的设计及可视化设计两大特点作了充分地阐述。本书除了讲述清晰、说明透彻外,特别针对开发应用而编制了大量的程序,许多程序都可以直接应用在实际的数据库系统中。

本书对于初学者来说,可以循序渐进地掌握 VFP 的应用及开发,但本书更适合希望用 VFP 开发出高速、强健而灵活的数据库系统的中、高级开发人员。

对于正在或将要学习和使用数据库的读者,本书都将是一个最佳的选择。

版权所有,盗版必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得进入各书店。

书 名: Visual FoxPro(5.0~6.0)面向对象编程指南

作 者: 张琦 梁恩主

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 北京市朝阳区科普印刷厂

发 行: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张: 27.625 字数: 672 千字

版 次: 1999 年 3 月第 1 版 1999 年 10 月第 3 次印刷

印 数: 10001~13000

书 号: ISBN 7-302-03471-0/TP·1890

定 价: 36.00 元

前 言

随着计算机技术的飞速发展,利用计算机来获得和处理信息是当今信息管理的一大特色。随着计算硬件的快速发展,有关信息管理的软件——数据库系统管理软件也在迅猛发展。

八十年代中期,Fox 公司推出了 FoxBase 系统,其他公司也推出了一些独立的系统。这些都是采用类似的接口和语言,从而形成 xBase 这一事实上的工业标准,xBase 产品占据了整个 PC 机数据库市场的一半以上。

现在 Microsoft 公司推出的 Visual FoxPro 5.0~6.0 可以进行可视化程序设计,同时引入了面向对象编程的思想,可以说这是 xBase 质的飞跃。Visual FoxPro 5.0 以上的版本所提供的速度、能力和灵活性,是在其他数据库管理系统中难以达到的。它将带领我们进入一个 xBase 的新时代,全新的对象和事件模型使得创建和修改应用程序比以往任何时候都要快速便捷。

Visual FoxPro 5.0 以上版本为我们编写信息管理系统软件提供了极大的方便,本书是编者在用 VFP 开发成功多个应用软件的基础上将不断总结的经验加以汇编,以期让读者体会并掌握 Visual FoxPro 面向对象的编程方法和技巧。全书内容共分为三大部分:

第 1 部分 数据库操作篇,本篇从第 1 章至第 6 章

在这一篇中我们主要介绍 Visual FoxPro 的使用和操作技巧,其中包括数据库表格和索引的建立、VFP 中的查询与报表、VFP 中的记录维护与锁定,以及更高级的数据库管理等内容。我们在讲述操作的同时都给出如何使用程序来实现相应操作的方法,以便读者阅读后就能编出相应的程序来实现所需的功能。

第 2 部分 VFP 面向对象程序设计篇,本篇从第 7 章至第 10 章

Visual FoxPro 是一种真正的面向对象的程序设计语言。通过面向对象程序设计方法,我们可以构建出既稳定又灵活的应用程序。在这一篇中我们将全面介绍 VFP 面向对象程序设计的方法,同时也详细讲解了用来设计表单的各种控件。以便读者掌握面向对象的设计方法和开发技巧。

第 3 部分 VFP 应用实践篇,本篇从第 11 章至第 15 章

本篇内容我们将介绍在实际应用软件开发过程遇到的主要问题和解决方案,同时针对开发应用编制了大量的示例程序,并讲述这些程序的调试方法以及如何发布应用程序。

本书实用性强,同时兼顾基础理论的阐释。它不仅是一本介绍 VFP 入门的书籍,而且所介绍的内容更适合于具有一定基础的中高级用户。特别针对用 VFP 进行应用程序开发的读者,本书具有较高的参考价值。

编者

1998 年 10 月

目 录

第 1 章 VFP 表格、数据库、索引和排序	(1)
1.1 为应用程序创建表格	(1)
1.1.1 数据类型和字段类型	(1)
1.1.2 拟定数据表内容	(3)
1.1.3 表设计器的使用	(4)
1.1.4 使用表向导	(5)
1.1.5 记录的字段	(6)
1.2 维护数据表结构	(9)
1.2.1 插入字段	(9)
1.2.2 删除字段	(11)
1.2.3 字段重命名	(11)
1.2.4 重定义字段	(11)
1.2.5 变更字段顺序	(12)
1.3 数据表记录输入	(12)
1.3.1 通过浏览或编辑屏幕录入记录	(12)
1.3.2 通过编程添加记录	(13)
1.3.3 从其它表格添加记录	(14)
1.4 数据索引	(15)
1.4.1 创建索引	(15)
1.4.2 使用索引	(18)
1.4.3 维护索引	(19)
1.4.4 删除索引	(19)
1.4.5 复杂的表达式索引	(20)
1.4.6 在运行时选择活动索引	(24)
1.5 数据排序	(27)
1.5.1 创建永久的排序	(27)
1.5.2 使用索引创建虚拟排序	(28)
1.6 创建数据库	(29)
1.6.1 创建一个数据库容器	(29)
1.6.2 将已存在的表格加入数据库	(31)
1.6.3 从数据库删除表格	(31)
1.7 通过编程修改表的属性	(32)
1.8 Visual FoxPro 6.0 新增功能	(33)
1.8.1 访问和赋值的方法程序	(33)
1.8.2 Active 文档	(33)
1.8.3 部件仓库	(33)

1.8.4	Coverage 和 Profiler 应用工具	(33)
1.8.5	GIF 和 JPEG 的支持	(34)
1.8.6	HTML 类型的帮助	(34)
1.8.7	增强的编程语言元素	(34)
1.8.8	OLE 拖放	(34)
1.8.9	新增的向导	(34)
1.9	小结	(34)
第 2 章	使用 BROWSE 查看数据	(35)
2.1	打开浏览器窗口	(35)
2.2	自定义浏览器窗口	(39)
2.2.1	改变字段的宽度	(39)
2.2.2	移动字段显示位置	(40)
2.2.3	分割浏览窗口	(41)
2.2.4	去除格线	(42)
2.3	增加、修改和删除数据	(43)
2.4	在命令窗口中使用 BROWSE 从句	(46)
2.4.1	使用 FIELDS 列表	(46)
2.4.2	使用 FOR 和 KEY 浏览表格子集	(49)
2.4.3	LAST 及 PREFERENCE 从句	(50)
2.4.4	WHEN VALID 和 ERROR 从句	(50)
2.4.5	TIMEOUT 从句	(51)
2.5	浏览相关的表格	(51)
2.5.1	与浏览相关的表格有关的主要命令	(51)
2.5.2	浏览一对一关系表	(55)
2.5.3	浏览一对多关系表	(55)
2.6	小结	(57)
第 3 章	查询与视图	(58)
3.1	建立查询文件	(58)
3.1.1	打开查询设计器	(58)
3.1.2	选取要查询的数据文件	(58)
3.1.3	设置查询设计器各种条件与属性	(59)
3.2	输出查询结果	(66)
3.2.1	输出到游标	(67)
3.2.2	输出到表格	(67)
3.2.3	输出到图形	(67)
3.2.4	输出到屏幕	(70)
3.2.5	输出到报表	(71)
3.2.6	输出到标签	(71)
3.3	在程序和表单中使用查询	(72)
3.4	SQL SELECT 语句	(73)

3.5	BETWEEN 和 IN 从句	(75)
3.5.1	使用 BETWEEN 从句过滤一个范围的值	(75)
3.5.2	使用 IN 从句过滤选定的值	(76)
3.6	使用视图设计器设计视图	(76)
3.6.1	设计本地视图	(76)
3.6.2	设计远程视图	(78)
3.7	使用视图	(81)
3.7.1	在视图中使用多个表	(81)
3.7.2	在视图中使用多个远程表	(82)
3.7.3	在视图中合并本地和远程数据	(82)
3.8	以编程方式创建和更新视图	(83)
3.8.1	创建及维护视图	(83)
3.8.2	创建参数化视图	(87)
3.8.3	打开视图的多个实例	(88)
3.8.4	在视图中更新数据	(89)
3.8.5	视图集成	(92)
3.9	小结	(92)
第 4 章	报表与标签	(93)
4.1	设计报表	(93)
4.2	调整报表文件	(96)
4.2.1	定义报表版面	(96)
4.2.2	设定数据来源	(99)
4.2.3	加入控件对象	(100)
4.2.4	加入计算字段	(102)
4.3	创建报表模板	(103)
4.4	在报表中使用表达式和变量	(104)
4.5	使用模板构建报表	(106)
4.6	使用程序初始化报表	(109)
4.7	标签向导	(113)
4.8	小结	(116)
第 5 章	高级的数据库管理	(117)
5.1	数据规范化	(117)
5.1.1	函数依赖	(118)
5.1.2	码	(118)
5.1.3	第一范式(1NF)	(119)
5.1.4	第二范式(2NF)	(119)
5.1.5	第三范式(3NF)	(120)
5.1.6	何时可违反规范化规则	(120)
5.2	命名规则	(121)
5.2.1	应用程序中的命名问题	(121)

5.2.2 命名表格中的字段	(121)
5.2.3 命名对象	(122)
5.3 设计数据库	(122)
5.3.1 数据库设计步骤	(123)
5.3.2 示例数据库图解	(134)
5.4 VFP 数据库容器的其他高级功能	(135)
5.4.1 使用记录级有效性检验	(135)
5.4.2 使用字段级有效性检验	(138)
5.4.3 维护参照的完整性	(141)
5.4.4 构造和删除永久性关系	(141)
5.4.5 使用参照完整性生成器	(143)
5.4.6 使用触发器	(146)
5.4.7 创建存储过程	(148)
5.5 小结	(149)
第 6 章 数据辞典	(150)
6.1 将数据库容器用作数据辞典	(150)
6.1.1 基本概念	(150)
6.1.2 如何定制和增强数据库容器的功能	(153)
6.1.3 取回和设置属性值	(156)
6.1.4 使用数据库容器工具	(160)
6.1.5 从数据库容器生成文档	(163)
6.2 设计数据辞典	(165)
6.3 小结	(168)
第 7 章 VFP 编程结构	(169)
7.1 VFP 程序设计基础	(169)
7.1.1 常量、变量、变量类型和变量作用域	(169)
7.1.2 运算符、运算符的优先级和表达式	(174)
7.1.3 数组	(176)
7.2 建立逻辑表达式	(185)
7.2.1 建立复合逻辑表达式	(186)
7.2.2 确定逻辑表达式的运算顺序	(187)
7.2.3 简化逻辑表达式	(187)
7.3 条件结构	(188)
7.3.1 IF...ENDIF 结构	(188)
7.3.2 CASE...END CASE 结构	(191)
7.3.3 何时使用 IF...ENDIF 或 DO...CASE 结构	(194)
7.4 循环结构	(195)
7.4.1 使用 DO WHILE...ENDDO 结构	(196)
7.4.2 使用 FOR...ENDFOR 结构	(198)
7.4.3 使用 SCAN...ENDSCAN 结构	(199)

7.5 执行外部程序	(200)
7.5.1 使用过程和用户自定义函数	(200)
7.5.2 使用 DO 命令执行外部程序	(203)
7.5.3 使用 SET PROCEDURE TO 调外部程序	(205)
7.5.4 使用 SET LIBRARY TO 调外部程序	(206)
7.6 小结	(207)
第 8 章 VFP 面向对象编程	(208)
8.1 对象及对象的特性	(208)
8.1.1 对象的定义	(208)
8.1.2 对象的特性	(209)
8.2 类和对象的关系	(211)
8.2.1 类	(211)
8.2.2 类与对象的区别	(212)
8.2.3 创建对象与释放对象	(216)
8.3 类和对象与代码重用	(218)
8.4 基类与子类	(221)
8.4.1 Visual FoxPro 基类	(221)
8.4.2 扩展 Visual FoxPro 类	(223)
8.4.3 类的属性、方法与事件	(224)
8.5 事件处理	(227)
8.5.1 为事件编写代码的规则	(227)
8.5.2 监视和追踪事件序列	(229)
8.6 创建类	(233)
8.6.1 用类设计器创建类	(233)
8.6.2 以编程方式定义类	(234)
8.6.3 定义子类	(235)
8.6.4 使用“类设计器”	(235)
8.6.5 类成员属性	(237)
8.6.6 向容器加控件	(239)
8.7 引用对象	(242)
8.7.1 相对引用	(242)
8.7.2 引用对象数组	(242)
8.7.3 使用域变换运算符引用方法	(243)
8.7.4 在容器层次中引用对象	(244)
8.8 创建类库	(245)
8.8.1 使用类库	(245)
8.8.2 修改库中的类定义	(246)
8.8.3 从类库中删除类定义	(247)
8.8.4 类复制	(247)
8.9 面向对象编程的特点	(248)
8.10 小结	(251)

第 9 章 表单及其控件	(252)
9.1 表单设计器	(252)
9.1.1 表单对象	(253)
9.1.2 属性窗口	(253)
9.1.3 窗口选项	(253)
9.1.4 数据环境	(255)
9.1.5 表单控件工具栏	(257)
9.1.6 布局工具栏	(258)
9.1.7 调色板工具栏	(258)
9.2 表单控件概述	(259)
9.3 常用的表单控件	(259)
9.4 向表单加入控件	(262)
9.4.1 深入了解容器和控件对象	(263)
9.4.2 添加 Visual FoxPro 容器	(263)
9.4.3 在表单中添加用户自定义对象	(264)
9.4.4 确定表单中的控件数目	(266)
9.4.5 包含预定义常量	(266)
9.5 处理表单对象	(267)
9.5.1 在设计时设置属性	(267)
9.5.2 定义表单行为	(268)
9.5.3 编辑事件代码和方法程序代码	(269)
9.5.4 保存表单为文件或类定义	(270)
9.5.5 运行表单	(271)
9.5.6 在运行时设置属性	(273)
9.5.7 在运行时调用方法程序	(274)
9.5.8 对事件作出响应	(275)
9.5.9 处理对象的示例	(275)
9.6 添加新的属性和方法程序	(278)
9.6.1 创建新属性	(278)
9.6.2 创建数组属性	(278)
9.6.3 创建新方法程序	(279)
9.7 设计一个 SDI 和 MDI 表单	(280)
9.7.1 指定表单类型	(280)
9.7.2 显示位于顶层表单中的子表单	(281)
9.7.3 隐藏 Visual FoxPro 主窗口	(281)
9.7.4 在顶层表单中添加菜单	(282)
9.8 小结	(282)
第 10 章 高级表单设计控件	(283)
10.1 使用列表框和下拉列表框	(283)
10.1.1 常用的列表框属性和方法程序	(283)

10.1.2	填充列表框或组合框	(284)
10.1.3	创建具有多列的列表框	(287)
10.1.4	允许用户选择列表框中的多项	(287)
10.1.5	允许用户在列表框中添加项	(288)
10.1.6	将图片添加到列表项	(288)
10.2	使用组合框	(289)
10.2.1	下拉组合框	(289)
10.2.2	常用组合框的属性	(290)
10.2.3	关于动态设置 RowSourceType 的示例	(290)
10.3	使用页框	(294)
10.3.1	将页框添加到表单	(295)
10.3.2	将控件添加到页面上	(296)
10.3.3	管理“页面”选项卡上的长标题	(296)
10.3.4	在程序中更换页面	(296)
10.3.5	常用的页框属性	(296)
10.3.6	关于页框的示例	(297)
10.4	定义表格	(299)
10.4.1	设置表格的属性	(300)
10.4.2	使用表格控件创建一对多表单	(301)
10.4.3	在表格列中显示控件	(301)
10.4.4	在表格中进行有条件格式编排	(303)
10.4.5	常用的表格属性	(303)
10.4.6	在表格中添加控制的示例	(304)
10.5	ActiveX 控件	(305)
10.5.1	Sysinfo 控件	(306)
10.5.2	ProgressBar 控件	(306)
10.6	小结	(307)
第 11 章	VFP 的项目管理	(308)
11.1	将项目管理器用作应用程序界面	(308)
11.1.1	从项目管理器中运行程序和表单	(308)
11.1.2	预览报表	(309)
11.1.3	打开数据库和浏览表格	(310)
11.2	在项目管理器中使用快捷方式	(312)
11.2.1	将标签作为主要类目的快捷方式	(312)
11.2.2	使用项目信息对话框	(313)
11.2.3	折叠和展开项目管理器	(315)
11.3	建立 VFP 的 .APP 和 .EXE 文件	(317)
11.3.1	设置主文件	(318)
11.3.2	在 .APP 和 .EXE 中包含和排除文件	(321)
11.3.3	从命令窗口建立项目	(322)
11.3.4	定制应用程序的运行状态	(325)
11.4	在工作组开发中使用 Visual SourceSafe	(327)

11.4.1	如何将 VFP 和 SourceSafe 结合	(329)
11.4.2	在 VSS 中管理文件	(331)
11.4.3	如何得到最新版本和显示版本历史	(335)
11.4.4	比较最后版本与 Check Out 修改的内容	(336)
11.5	小结	(337)
第 12 章	应用程序调试	(338)
12.1	程序测试和调试计划	(338)
12.2	在程序存在错误前进行调试	(338)
12.2.1	在 Visual FoxPro 5.0 中一般的命令规则	(338)
12.1.2	建立测试环境	(340)
12.1.3	设置验证信息	(340)
12.1.4	查看事件发生的序列	(341)
12.3	逐步发现错误	(342)
12.3.1	启动调试工作期	(342)
12.3.2	跟踪代码	(342)
12.3.3	停止程序的执行	(343)
12.3.4	查看存储的值	(347)
12.4	显示输出结果	(349)
12.5	记录代码的覆盖范围	(349)
12.6	处理“运行时”的错误	(351)
12.6.1	预防错误	(351)
12.6.2	处理过程中的错误	(352)
12.7	小结	(355)
第 13 章	创建联机帮助	(356)
13.1	FoxPro 所支持的帮助形式	(356)
13.1.1	DBF 样式的帮助	(356)
13.1.2	图形样式的帮助	(357)
13.1.3	Windows 风格帮助的特点	(358)
13.2	联机帮助设计的基本概念	(360)
13.2.1	帮助信息的类型	(360)
13.2.2	主题	(361)
13.2.3	最低限度的编程	(362)
13.2.4	信息分层	(362)
13.3	创建 DBF 形式的帮助系统	(362)
13.4	为应用系统加入 DBF 形式的帮助	(365)
13.4.1	为应用程序指定一个帮助表	(365)
13.4.2	控制“帮助窗口”的位置	(365)
13.4.3	用主题名来选取主题	(366)
13.4.4	创建“上下文相关”的帮助	(366)
13.5	创建 Windows 95 帮助系统	(367)

13.5.1	创建帮助主题文件	(367)
13.5.2	创建帮助文件	(372)
13.5.3	为帮助加入 Contents	(375)
13.5.4	加入图形	(375)
13.5.5	加入带热点的图形	(376)
13.6	为应用系统加入图形样式的帮助	(377)
13.6.1	装入帮助文件	(377)
13.6.2	激活帮助	(378)
13.6.3	建立基本的帮助菜单	(378)
13.6.4	上下文敏感的帮助	(378)
13.6.5	加入表单级的上下敏感帮助	(380)
13.6.6	加入 What's this 帮助	(380)
13.6.7	用编程实现帮助特性	(381)
13.7	小结	(384)
第 14 章	应用程序的发布	(385)
14.1	应用程序的发布过程	(385)
14.1.1	发布过程	(385)
14.1.2	了解“发布树”	(385)
14.1.3	“安装向导”如何工作	(386)
14.2	准备要发布的应用程序	(387)
14.2.1	选择连编类型	(387)
14.2.2	考虑环境问题	(387)
14.2.3	确保运行时的行为正确	(388)
14.2.4	在应用程序中包含资源	(388)
14.2.5	删除受限制的功能和文件	(390)
14.3	使用“安装向导”	(391)
14.4	小结	(400)
第 15 章	开发示例	(401)
15.1	总体介绍	(401)
15.1.1	运行 Tasmanian Traders	(401)
15.1.2	总体设计	(402)
15.2	数据库设计	(403)
15.2.1	创建表的结构和表之间的关系	(403)
15.2.2	实现安全性	(404)
15.2.3	维护数据的一致性	(404)
15.3	类库设计	(406)
15.3.1	About 类库	(407)
15.3.2	Login 类库	(407)
15.3.3	Orders 类库	(408)
15.3.4	tsBase 类库	(409)

15.3.5	tsBaseForm 类库	(409)
15.3.6	tsGen 类库	(413)
15.4	系统实现	(415)
15.4.1	表单	(416)
15.4.2	报表	(417)
15.4.3	菜单和工具栏	(417)
15.4.4	出错处理	(417)
15.4.5	测试和调试	(419)
15.4.6	主程序	(419)
15.4.7	存储程序	(420)
15.5	小结	(421)
附录 A	Visual FoxPro 快捷键	(422)
附录 B	Visual FoxPro 中的常见事件	(424)
附录 C	Visual FoxPro 中常用方法程序	(426)

第1章 VFP 表格、数据库、索引和排序

20 世纪 80 年代中期, Fox 公司推出了 Foxbase 系统, 它兼容于早年推出的 dBase 系统。在同时期, 有许多公司都推出了自己独立的数据库产品, 它们都和 Foxbase 具有类似的接口和语言。所有的这些产品形成了被称之为 xBase 这种事实上的工业标准, 它们占据了整个数据库市场的一半以上。

著名的 Microsoft 公司兼并 Fox 公司以后推出了 FoxPro, 它以快速的查询、灵活的操作迅速占领了市场。紧接着, Microsoft 公司在 FoxPro 产品中引入了可视化程序设计以及面向对象编程的思想, 推出了兼有速度快、能力强、灵活快大的 Visual FoxPro 产品。

从本章开始, 本书将带你进入 Visual FoxPro 的世界。

1.1 为应用程序创建表格

表格在日常生活中使用十分频繁, 也是我们管理数据的有效方法。在程序设计特别是在数据管理系统中, 自然离不开表格。日常生活中的表和计算机中的表既有着密切的联系, 又有很大的区别。但不管怎么说, 计算机中所使用的表离不开现实生活中的表模型。在 Visual FoxPro 中, 数据库和表又有所区别, 我们可以创建数据库表, 也可以创建与数据库无关联的自由表。这一节将介绍为应用程序创建表格, 以及对表格的相关操作。

1.1.1 数据类型和字段类型

1. 数据类型

一个数据库可以存储多种不同类型的数据, 下面我们就先讲述这些数据的类型。

(1) 字符数据类型

当数据需要使用字母、数字、空格、符号以及标点时, 请使用字符型数据。字符型字段或变量可以保存诸如姓名、地址和无需计算的数字等文本信息。例如, 电话号码或邮政编码之类的数据作为字符值保存更便于处理。

(2) 逻辑数据类型

当数据只包含两个值时, 一种有效的存储方式便是使用逻辑数据类型。逻辑型数据的值只可以为“真”(T.)或“假”(F.)。

(3) 货币数据类型

当在保存货币值时, 请使用货币数据类型 (而不是数值类型)。在货币表达式中如果小数位数超过 4 位, 则 Visual FoxPro 在处理表达式之前将它四舍五入到 4 位。货币数据类型一般使用美元符\$表示。如

Money = \$100

MoreMoney = \$10000.43886

在上例中, MoreMoney 变量将在内部四舍五入成 10000.4389。

(4) 日期数据类型

可使用日期数据类型保存不带时间的日期值。日期型变量以“yyyymmdd”字符格式保存。

若要赋日期值, 应将日期放在花括号中:

```
MyBirthday = {10/14/1974}
```

空日期值可使用一对花括号或在花括号中加一个空格或正斜杠表示。例如:

STORE {} to dBlankdate0	&&花括号
STORE { } to dBlankdate1	&&在花括号中加一个空格
STORE {/} to dBlankdate2	&&在花括号中加一个正斜杠

(5) 日期时间数据类型

当保存日期、时间或二者兼有时, 请使用日期时间数据类型。日期时间值分别存储在两个 4 字节整数中, 第一个 4 字节整数保存日期, 第二个 4 字节保存时间。时间从午夜起计算, 以 1/100 秒为最小计时单位, 日期时间值可以包含完整的日期和时间, 也可以只包含二者之一。若用户没有指明日期值, 则 Visual FoxPro 用系统默认值 1899 年 12 月 30 日填入; 若用户没有指明时间值, 则 Visual FoxPro 用默认的午夜时间填入。

下列规则适用于日期型或日期时间型数据:

- {00:00:00AM} 等价于午夜 {12:00:00AM}。
- {00:00:00PM} 等价于中午 {12:00:00PM}。
- {00:00:00} 至 {11:59:59} 等价于 {12:00:00AM} 至 {11:59:59AM}。
- {12:00:00} 至 {23:59:59} 等价于 {12:00:00PM} 至 {11:59:59PM}。

(6) 数值数据类型

数值型数据用来表示数量。它包含数字 0~9, 也可加上一个正负号或小数点。

表 1-1 给出了以上所述数据类型及其范围, 以供用户查找。

2. 字段类型

设计一个数据库时, 通常需要把记录分为几个部分, 每个部分就是一个字段。字段是记录中拥有特定数据类型的命名部分。命名字段在设计阶段由表设计器设置其数据类型, 或者在运行时由 CREATE TABLE 命令来确定。

字段类型可以是 Visual FoxPro 允许的任意数据类型或专用的字段类型。前面已经讲述过的数据类型都可以作为字段的类型, 这里我们还要介绍两个专门的字段类型。

(1) 备注字段类型

备注字段类型可用来存储数据块。备注字段含有一个 10 字节的引用, 它指向真正的备注内容, 备注型数据的实际大小取决于用户输入的数据量。

表 1-1 数据类型

数据类型	说明	大小	范围
字符型	任意文本	每个字符只用一个字节, 最多可有 254 个字符	任意字符
货币型	货币量	8 个字节	从 922337203685477.5808 到 922337203685477.5807
日期型	包含有年、月和日的数据	8 个字节	01/01/100 到 12/31/9999
日期时间型	包含有年、月、日和时间的数据	8 个字节	年、月、日从 01/01/100 到 12/31/9999, 时间从 00:00:00 a.m. 到 11:59:59 p.m.
逻辑型	“真”或“假”的布尔值	1 个字节	“真”(T.) 或“假”(F.)
数值型	整数或分数	在内存中占 8 个字节; 在表中占 1 到 20 个字节	从 .9999999999E+19 到 .9999999999E+20

(2) 通用字段类型

可使用通用字段来保存 OLE 对象。通用字段包含一个 10 字节的引用, 它指向该字段真正的内容, 如电子表格、字处理文档或用另一个应用程序创建的图片等。通用字段的大小仅受可用磁盘空间的限制。

1.1.2 拟定数据表内容

在 FoxPro 中, 表是关系数据库管理系统的基本结构。表格中的特定格式能让用户更容易浏览表格记录, 和找出想要的信息。注意, 应将 Visual FoxPro 的表 and 数据库区别开来。在 Visual FoxPro 中, 表相当于以前所讲的数据库, 但从 Visual FoxPro 3.0 版后, Visual FoxPro 的数据库是以 .DBC 为扩展名, 它可以包含一个或多个表和视图甚至到远程数据源的连接和存储过程。因此, 数据库是个更广的定义, 而一个表是以记录(行)和字段(列)的形式存储数据。数据常常是涉及某一类事物的信息, 如客户、客户订单、雇员、存货信息等等。一个 Visual FoxPro 表可以用 .DBF 扩展名存于自己的文件中, 也可以包含在一个数据库中。

在 Visual FoxPro 中, 设计表的第一步是明确表的目的和使用方法, 也就是说我们需要用表来存储哪些信息。明确目的之后, 就可以确定需要保存哪些主题的信息(即表), 以及每个主题需要保存哪些信息(即表中的字段), 并确定表中的字段以及字段的类型。

下面通过一个实例分析一下如何完成这个任务。假设有一个订货单, 它除了包含一些顾客的信息(如顾客的代号和定货时间等)以外, 还提供了有关订货情况的信息。我们的目的, 就是要保存订单的信息。现实中订单包含以下一些内容:

订单号
顾客号
顾客地址
订货量