

微机实用新技术丛书

Visual
FoxPro

高级程序设计
技术探密

裘实 魏决 阳光 等编著

国防工业出版社



Visual FoxPro 高级程序 设计技术探密

裘实 魏泱 阳光 等编著

国防工业出版社

·北京·

图书在版编目(CIP)数据

Visual Foxpro 高级程序设计技术探密/裘实等编著. —
北京:国防工业出版社,1999.1
(微机实用新技术丛书)
ISBN 7-118-02007-9

I. V… II. 裘… III. 关系数据库-数据库管理系统, Foxpro-程序设计 IV. TP311.13

中国版本图书馆 CIP 数据核字(98)第 27006 号

 国防工业出版社 出版发行
(北京市海淀区紫竹院南路 23 号)
(邮政编码 100044)
涿州一中印刷厂印刷
新华书店经售

开本 787×1092 1/16 印张 20 456 千字
1999 年 1 月第 1 版 1999 年 1 月北京第 1 次印刷
印数:1—4000 册 定价:27.00 元

(本书如有印装错误,我社负责调换)

编 者 的 话

在当今计算机技术突飞猛进发展的时代,如何快速、准确地吸收和采用新技术,推动国民经济现代化的发展,是我们每个科技工作者的责任。一方面是铺天盖地、眼花缭乱的新名词、新软件,一方面是无数刚刚接触计算机概念,缺乏专业知识,而又在工作中急需使用最新软件技术的各行各业工作人员及计算机爱好者。为了更快、更好地架起这两者之间的桥梁,几位有识之士在国防工业出版社阎瑞琪主任的倡导下,组织起来,克服困难,认真负责、快速编写了这套微机实用新技术丛书,并将进一步把握新技术的脉搏,推出更多、更好的书籍奉献给读者。

本套丛书内容新颖、实用,语言简练,选题着眼读者最广泛的软件与技术,力求使读者在短时间内掌握、吸收,并能灵活运用。

参加丛书编撰的全部是具有博士或硕士研究生学位,并在计算机领域从事实用技术研究和开发的专家及学者。

丛书全体编者衷心企望本丛书能对广大读者的学习和工作提供有力的帮助,并衷心感谢阎瑞琪主任的支持和帮助。

由于时间仓促,本丛书无论在选题策划还是在编写细节上,都可能不足之处,恳切希望广大读者批评指正。

前 言

作为第一个正式具有 Windows 95 兼容标志的应用软件, Visual FoxPro 无疑是 FoxPro 产品历史上最受欢迎的版本, 其各种卓越的改进功能, 包括从面向字符式数据库转向面向对象风格, 完全支持 Windows 事件模式以及数据字典的创建等等, 都将深刻地影响 Xbase 技术的发展趋势。Microsoft 同时针对其他 Visual 开发工具系列(如 Visual BASIC、Visual C++ 等), 宣布了 Visual FoxPro 5.0 专业版功能的增强和完善, 使这一强劲的 32 位面向对象的快速应用程序开发环境, 在客户/服务器的应用能力和 FoxPro 自身的性能上都得以显著地提高。随着桌面操作系统由 Windows 3.1 逐渐向 Windows 95 升级的发展潮流, Visual FoxPro 必将成为今后数据库产品中的主流。

本书作者都是长期从事 Visual FoxPro 数据库软件开发的技术人员和大学教授, 具有深厚的技术背景。本书内容深入, 适合具有专业水平的读者参考, 对开发大中型数据库软件, 要求用 FoxPro 与 Oracle、Sybase、Informix 等联合开发的读者更有吸引力。

本书共分十五章, 内容重点放在客户/服务器程序设计等大型及高级程序设计技术的介绍上, 对已经熟悉数据库软件开发、具有较高专业水平的读者非常实用, 对希望提高 Visual FoxPro 程序设计水平, 了解大中型数据库设计开发方法和了解客户/服务器应用程序开发技术的读者, 可以通过阅读本书, 实现数据库系统的快速升级, 实现与大型数据库系统的连接。

本书第一章到第三章由裘实、刘秀芳、朱志坚合作编写, 第四章由刘松海编写, 第五章到第七章由魏泱、李宁合作编写, 第八章由邱桂清编写, 第九章和第十章由徐洁、李纪洪合作编写, 第十一章到第十五章由阳光、王仲谋、农军、吴青合作编写。全书由顾铁成、裘实统稿, 曹春光审校。王玲玲为本书的编排工作付出了艰辛的劳动。

目 录

第一章 Visual FoxPro 面向对象的程序设计	1
1.1 从结构化程序设计进步到面向对象的程序设计	1
1.2 Visual FoxPro 中的类和对象	2
1.2.1 类与对象	2
1.2.2 Visual FoxPro 类的层次	3
1.3 对象的属性、方法和事件	6
1.3.1 在容器层次中引用对象	6
1.3.2 设置属性	7
1.3.3 调用方法程序	8
1.3.4 响应事件	8
1.4 编写类定义	9
1.4.1 保护类成员	9
1.4.2 在容器类中添加对象	10
1.4.3 指派方法程序代码和事件代码	10
1.4.4 定义自定义类	18
1.4.5 创建对象引用	18
1.4.6 创建成员数组	19
1.4.7 创建对象数组	20
1.4.8 使用对象存储数据	20
第二章 使用 Visual FoxPro 的向导	22
2.1 使用应用程序向导	23
2.1.1 应用程序向导的具体使用步骤	23
2.1.2 应用程序向导的对象方法程序	25
2.1.3 应用程序向导的对象属性	27
2.1.4 应用程序向导的对象成员	27
2.2 使用远程视图向导	28
2.3 使用文档向导	29
2.3.1 文档向导的具体使用步骤	29
2.3.2 定制文档向导的附加选项	31
第三章 开发 Visual FoxPro 应用程序	33
3.1 设计 Visual FoxPro 应用程序	33
3.1.1 规划应用程序	33
3.1.2 创建数据库	34

3.1.3 使用类简化应用程序	35
3.1.4 提供交互能力	35
3.1.5 提供交互信息	35
3.1.6 测试和调试	35
3.2 创建 Visual FoxPro 应用程序	35
3.2.1 构造应用程序	36
3.2.2 向项目中添加文件	39
3.2.3 连编应用程序	40
3.2.4 运行应用程序	41
3.3 测试和调试应用程序	42
3.3.1 程序测试和调试计划	42
3.3.2 调试的准备工作	42
3.3.3 快速确定测试中的问题位置	44
3.3.4 显示输出结果	50
3.3.5 记录代码的覆盖范围	50
3.3.6 错误处理	51
第四章 优化 Visual FoxPro 应用程序	58
4.1 Rushmore 技术概述	58
4.1.1 使用 Rushmore 技术优化对数据的访问	58
4.1.2 关闭 Rushmore	59
4.2 使用 Rushmore 技术优化表达式	60
4.2.1 创建基本可优化表达式	60
4.2.2 组合基本可优化表达式	61
4.2.3 组合复杂表达式	61
4.3 使用 Rushmore 技术优化表单和控件	62
4.4 提高 Visual FoxPro 的性能	64
第五章 使用远程视图	68
5.1 定义指定数据源的连接	68
5.1.1 命名连接与数据源优先级	69
5.1.2 显示 ODBC 注册提示	69
5.1.3 使用已有的连接	69
5.2 创建远程视图	69
5.3 设置视图属性和连接属性	70
5.4 下载远程视图时默认数据类型的变化	71
5.5 通过视图更新数据	72
5.5.1 如何更新视图	72
5.5.2 在视图中更新多个表	74
5.6 在视图的字段或记录上创建规则	75
5.7 如何集成多个视图	76

5.8 游离数据与游离视图	77
5.8.1 如何创建游离视图	78
5.8.2 游离数据的显示和修改	78
5.8.3 如何更新联机数据	79
5.9 通过设置视图属性优化视图性能	81
第六章 设计共享访问程序	85
6.1 数据访问操作的控制	85
6.1.1 访问数据的方式	85
6.1.2 共享访问与锁定数据的方式	86
6.1.3 如何解锁数据	89
6.1.4 了解并使用数据工作期	90
6.1.5 使用数据缓冲技术	92
6.1.6 用缓冲技术对数据进行更新	95
6.2 通过事务更新数据	96
6.2.1 了解控制事务处理的命令	97
6.2.2 事务的规则及完成的操作	97
6.2.3 事务处理的嵌套方法	98
6.2.4 确保远程更新的安全性	101
6.3 访问冲突的管理	102
6.3.1 如何管理缓冲冲突	102
6.3.2 管理用视图更新数据时发生的冲突	106
6.3.3 管理冲突的规则	109
6.4 如何提高多用户应用程序的性能	109
第七章 客户/服务器解决方案的设计	110
7.1 客户/服务器应用设计方法	110
7.1.1 客户/服务器应用程序的设计目标	110
7.1.2 如何提高系统性能	110
7.1.3 加快应用程序的开发速度	115
7.1.4 数据完整性的考虑	116
7.2 利用“升迁向导”实现数据库升迁	117
7.2.1 原型化方法	117
7.2.2 构造应用程序的本地原型	118
7.2.3 使用升迁向导实现数据库升迁	118
7.2.4 将本地数据库升迁到 SQL Server 上	118
7.2.5 将本地数据库升迁到 Oracle 上	131
7.3 实现客户/服务器应用程序	131
7.3.1 使用 SQL pass-through 技术	132
7.3.2 用 SQL pass-through 处理远程数据	144
7.3.3 处理 SQL pass-through 错误	153

7.4 优化客户/服务器应用程序的性能	153
7.4.1 优化连接技术	154
7.4.2 加速数据检索	154
7.4.3 加速查询和提高视图的性能	157
7.4.4 加速表单的运行	158
7.4.5 提高更新和删除的性能	159
第八章 设计 OLE 应用程序	162
8.1 如何设计 OLE 应用程序	162
8.1.1 如何链接或嵌入 OLE 对象	162
8.1.2 添加绑定型或非绑定型 OLE 对象	164
8.2 添加 OLE 对象	164
8.2.1 在表中添加 OLE 对象	164
8.2.2 在表单中添加 OLE 对象	166
8.3 用 ActiveX 控件管理对象	168
8.3.1 向表单中添加 ActiveX 控件	169
8.3.2 绑定型 ActiveX 控件的管理方法	169
8.3.3 如何自动管理对象	169
8.4 通过派生对象定制新对象	173
8.5 从其他应用程序中控制 Visual FoxPro	174
8.5.1 Application 对象模型简介	175
8.5.2 如何通过集合属性访问对象	175
8.6 如何创建 OLE 服务程序	176
8.6.1 创建 OLE 服务程序	176
8.6.2 如何注册 OLE 服务程序	179
8.6.3 使用 OLE 服务程序	180
8.6.4 引起错误或从 OLE 服务程序返回错误	180
8.7 实现远程自动化方案	180
8.7.1 配置服务器和本地计算机	181
8.7.2 系统安全策略选项	182
8.7.3 在远程自动化中使用权限	182
8.7.4 远程自动化	183
第九章 应用程序帮助系统的设计	184
9.1 确定帮助系统的特性	184
9.2 设计对联机帮助的访问	184
9.2.1 设计一个帮助菜单	185
9.2.2 如何添加上下文相关性	185
9.2.3 实现“这是什么?”形式的帮助	186
9.3 通过编程实现帮助特性	188
9.3.1 使用 WinHelp 函数设计帮助系统	188

9.3.2 指定 WinHelp 参数的取值	189
9.3.3 保留 F1 键供帮助系统	190
9.3.4 如何在表单中设计“帮助”按钮	190
9.3.5 退出图形样式帮助	191
第十章 Visual FoxPro 应用程序的发布	192
10.1 应用程序的发布过程	192
10.2 发布应用程序前的准备工作	192
10.2.1 如何选择连编类型	192
10.2.2 综合考虑硬件、内存和网络问题	193
10.2.3 确保运行时的行为正确	194
10.2.4 如何在应用程序中包含资源	195
10.2.5 删除受限制的 Visual FoxPro 功能和文件	197
10.3 如何定制要发布的应用程序	198
10.3.1 源代码的保护及文档化	198
10.3.2 调用错误处理和终止系统运行例程	199
10.3.3 添加帮助信息	199
10.3.4 修改应用程序的外观	199
10.3.5 备份并保存源代码	200
10.3.6 生成应用程序	201
10.4 理解磁盘发布的方法	201
10.4.1 了解发布过程	201
10.4.2 尽量利用安装向导	203
第十一章 调用应用程序编程接口 (API)	206
11.1 使用外部库扩展 Visual FoxPro 的功能	206
11.1.1 使用外部库	206
11.1.2 如何访问 ActiveX 控件	207
11.1.3 如何访问动态链接库	208
11.1.4 访问 Visual FoxPro 库	210
11.2 访问 Visual FoxPro 应用程序编程接口	210
11.2.1 创建库或 ActiveX 控件的方法	210
11.2.2 调用 Visual FoxPro API 例程	215
11.2.3 参数的传递和接收	216
11.2.4 将值返回给 Visual FoxPro 的方法	220
11.2.5 将参数传递给 Visual FoxPro API 函数	222
11.2.6 访问 Visual FoxPro 变量和字段的方法	224
11.2.7 Visual FoxPro 的内存管理	225
11.2.8 连编和调试 ActiveX 控件及库	227
第十二章 Visual FoxPro 的企业级应用	230
12.1 企业级开发方法	230

12.2 用 Visual FoxPro 作为前端应用程序开发环境	231
12.2.1 如何扩展 Visual FoxPro 中可视化设计工具的功能	231
12.2.2 如何集成其他程序的功能	232
12.2.3 如何扩展 Visual FoxPro 的数据存储能力	233
12.2.4 如何升迁 Visual FoxPro 数据	234
12.3 使用 Visual FoxPro 作为数据源	235
12.3.1 在其他应用程序中使用 Visual FoxPro 的数据	235
12.3.2 在其他应用程序中使用 Visual FoxPro 对象和命令	235
12.3.3 用 Visual FoxPro 建立数据仓库	236
12.3.4 把 Visual FoxPro 当作 WWW 查询引擎	237
第十三章 Visual FoxPro 随机示例简介	239
13.1 Visual FoxPro Solutions 示例简介	239
13.2 Tasmanian Traders 示例简介	240
13.2.1 Tasmanian Traders 总体设计概述	240
13.2.2 如何设计 TASTRADE 数据库	241
13.2.3 如何设计与建立 Tasmanian Trader 类库	243
13.2.4 Tasmanian Trader 程序的编码、测试和调试	244
13.2.5 Tasmanian Traders 类库介绍	249
13.3 理解客户/服务器示例	255
13.3.1 示例应用程序类介绍	255
13.3.2 示例程序数据库介绍	257
13.3.3 在示例应用程序中选择一个临时数据表	258
13.3.4 在示例应用程序中更新数据	258
13.3.5 在示例应用程序中解决数据冲突问题	260
13.3.6 在示例应用程序中实现商务规则	261
13.4 OLE 服务程序示例介绍	262
13.4.1 Pool Manager 服务程序示例	262
13.4.2 Gopher OLE 服务程序示例	265
13.4.3 Fox ISAPI OLE 服务程序示例	267
13.5 示例类库介绍	273
13.6 多媒体类示例介绍	277
13.7 Visual FoxPro 部分实用程序简介	278
13.7.1 ADDLABEL.APP	279
13.7.2 CPZERO.PRG	279
13.7.3 GENDBC.PRG	280
13.7.4 转换器	280
13.7.5 代码分析器	281
13.7.6 HexEdit	281
13.7.7 远程自动化连接管理器	281

第十四章 使用 FPOLE.DLL 和配置文件	283
14.1 使用 FPOLE.DLL	283
14.1.1 FoxDoCmd()函数	283
14.1.2 FoxEval()函数	284
14.1.3 SetErrMode()函数	284
14.1.4 SetOLEObject()函数	285
14.1.5 CloseIt()函数	285
14.1.6 GetLastError()函数	285
14.1.7 FPOLE 示例	286
14.2 使用配置文件	289
14.2.1 创建配置文件	289
14.2.2 自动启动应用程序或程序	290
14.2.3 修改默认配置文件	290
14.2.4 指定配置文件	291
第十五章 大型 Visual FoxPro 应用程序开发技术	292
15.1 了解集体开发技术	292
15.2 了解源代码管理技术	292
15.2.1 在 Visual FoxPro 中使用源代码管理工具	293
15.2.2 在源代码管理系统下管理 Visual FoxPro 项目	295
15.3 在源代码管理项目中管理文件	300
15.3.1 多文件组件简介	300
15.3.2 如何签出文件	301
15.3.3 如何签入文件	301
15.3.4 如何得到文件的最新版本	303
15.3.5 比较文件或项目的方法	303
15.3.6 检查表单、报表和其他表文件的区别	304
15.3.7 如何显示文件和项目信息	305
15.4 集体开发与数据库修改技术	306
15.4.1 处理视图和连接的方法	306
15.4.2 集体开发类库的方法	306

第一章 Visual FoxPro 面向对象的程序设计

简单地讲,程序设计就是为了完成某一具体任务而编写的一系列指令,而 Visual FoxPro 程序设计实际上就是对存储数据的操作。

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。这些指令包含在以下地方:

- ① 命令窗口;
- ② 程序文件;
- ③ 表单设计器的事件/方法代码窗口;
- ④ 菜单设计器的过程代码窗口;
- ⑤ 报表设计器的过程代码窗口。

1.1 从结构化程序设计进步到面向对象的程序设计

20 世纪 80 年代,结构化程序设计是最主要、最通用的程序设计方法。结构化程序设计是将一个程序用以下三种控制流程来编写:

① 顺序:这是最基本的控制流。顺序控制意味着命令将按代码出现的先后顺序逐条执行;

② 选择:使用 IF 和 ELSE 从一组选项中选择一项。如果条件为真,程序执行一种操作,否则执行其他操作;

③ 循环:只要某一个条件为真,则反复执行相同的命令。

结构化程序设计还意味着将一个程序分解为多个函数或过程。自顶向下逐步求精的结构化程序设计方法是把一个问题分成若干子问题,依次进行下去,直到子问题很容易把握为止。体现在程序设计语言中,即为高层函数包含低层函数,而低层函数又包含更低层函数。结构化程序设计就是通过函数来体现设计的层次性。将程序分解为独立的过程还有利于程序的维护,因为只需对一个过程进行修改,而对程序的其他部分不会产生不可预期的改动。而且,每次测试和调试程序中的一个过程要比一次测试和调试整个程序容易得多。

随着工程规模的扩大,软件的复杂性也大幅度增加了。为了缩短软件的研制时间,提高软件开发的效率,一种新的编程方法应运而生,它就是面向对象的程序设计方法(Object-Oriented Programming),简称 OOP。面向对象的程序设计是一种试图模仿人们建立现实世界模型的程序设计方法,它利用了人们对事物分类的自然倾向,引进了类的概念,具有数据抽象、继承性等特点。

Visual FoxPro 不但仍然支持标准的 Xbase 结构化程序设计,而且在语言上还进行了

扩展,提供了面向对象程序设计的强大功能和更大灵活性。

面向对象的程序设计方法与编程技术不同于标准的结构化程序设计。程序设计人员在进行面向对象的程序设计时,不再是单纯地从代码的第一行一直编到最后一行,而是考虑如何创建对象,利用对象来简化程序设计,提供代码的可重用性。对象可以是应用程序的一个自包含组件,一方面具有私有的功能,供自己使用;另一方面又提供公用的功能,供其他用户使用。

1.2 Visual FoxPro 中的类和对象

在 Visual FoxPro 中,表单、表单集以及控件等都是应用程序中的对象。用户通过对象的属性、事件和方法程序来处理对象。

Visual FoxPro 面向对象的语言扩展部分为应用程序中的对象提供了更多的控件,同时也使得创建和维护可重用代码库更为容易。它有如下优点:

- ① 使代码更紧凑;
- ② 在应用程序中可更容易地加入代码,而不必精心确定方案的每个细节;
- ③ 减少了不同文件代码集成为应用程序的复杂程度。

面向对象程序设计基本上是一种包装代码,代码可以重用而且维护起来很容易。其中最主要的包装概念称为类。

注意,表单(Form)也就是 Visual Basic 中的窗体。在表单上可以添加各种控件,经常用于创建输入输出界面。表单的实例可以在各种对话框中找到。表单集(Formset)就是表单的集合,可以统一操作表单集中的表单。

1.2.1 类与对象

类和对象关系密切,但并不相同。类包含了有关对象的特征和行为信息,它是对象的蓝图和框架。例如,楼房的结构和设计布局可以是一个类,而这个类的实例——对象便是一栋楼房。

1. 对象具有属性

每个对象都有属性。在 Visual FoxPro 中,创建的对象也具有属性,这些属性由对象所基于的类决定。例如,表 1.1 列出了一个文本框可能有的属性。

表 1.1 文本框可能具有的属性

属 性	说 明
BackColor	文本框的背景颜色
Enabled	用户能否在文本框中输入文本
Font Name	文本的字体
Left	文本框左边的位置
MousePointer	在文本框内鼠标指针的形状
Top	文本框顶边的位置
Visible	文本框是否可见

某些属性值既能在设计阶段也可在运行阶段进行设置,例如文本框的 Left 属性;某些属性值则不能进行设置,是只读的,例如文本框的 Class 属性;属性值也有特定的数据类型,例如文本框的 Visible 属性只能取 .T. 或 .F.,而 Name 属性可以是字符型值。

2. 对象具有与之相关联的事件和方法程序

每个对象都可以对称为事件的动作进行识别和响应。事件是一种预先定义好的特定动作,由用户或系统激活。通常,事件是由用户的交互操作产生的。在 Visual FoxPro 中,可以激发事件的用户动作包括单击鼠标、移动鼠标和按键。

方法程序是与对象相关联的过程,但又不同于一般的 Visual FoxPro 过程。方法程序紧密地和对象连接在一起,并且与一般 Visual FoxPro 过程的调用方式也有所不同。

事件可以具有与之相关联的方法程序。例如,为 Click 事件编写的方法程序代码将在 Click 事件出现时被执行。方法程序也可以独立于事件而单独存在,此类方法程序必须在代码中被显式地调用。

事件集合虽然范围很广,但却是固定的,您可以根据需要为某些事件编制代码。您不能创建新的事件,然而方法程序集合却可以无限扩展。表 1.2 列出了与复选框相关联的一些事件。

表 1.2 与复选框相关联的事件

事 件	说 明
Click	用户单击复选框
GotFocus	用户选择复选框
LostFocus	用户选择其他控件

表 1.3 列出了与复选框相关联的一些方法程序。

表 1.3 与复选框相关联的一些方法程序

方法程序	说 明
Refresh	复选框中的值被更新,以反映隐含数据源的数据变化
SetFocus	焦点被置于复选框,好像用户刚使用 TAB 键选中复选框

1.2.2 Visual FoxPro 类的层次

如果您正在创建 Visual FoxPro 用户自定义类,图 1.1 可以帮助您理解 Visual FoxPro 类的层次。

1. Visual FoxPro 基类

从下述 Visual FoxPro 基类中,可以派生子类或创建对象,如表 1.4 所列。

表 1.4 Visual FoxPro 基类

复 选 框	编 辑 框	列 表 框	形 状
列 *	表单	OLE 绑定型控件	微调
命令按钮	表单集	OLE 容器控件	文本框
命令组	表格	选项按钮 *	计时器

(续)

复选框	编辑框	列表框	形 状
组合框	标头 *	选项组	工具栏
容器	图像	页面 *	
控件	标签	页框	
自定义	线条	分隔符 *	

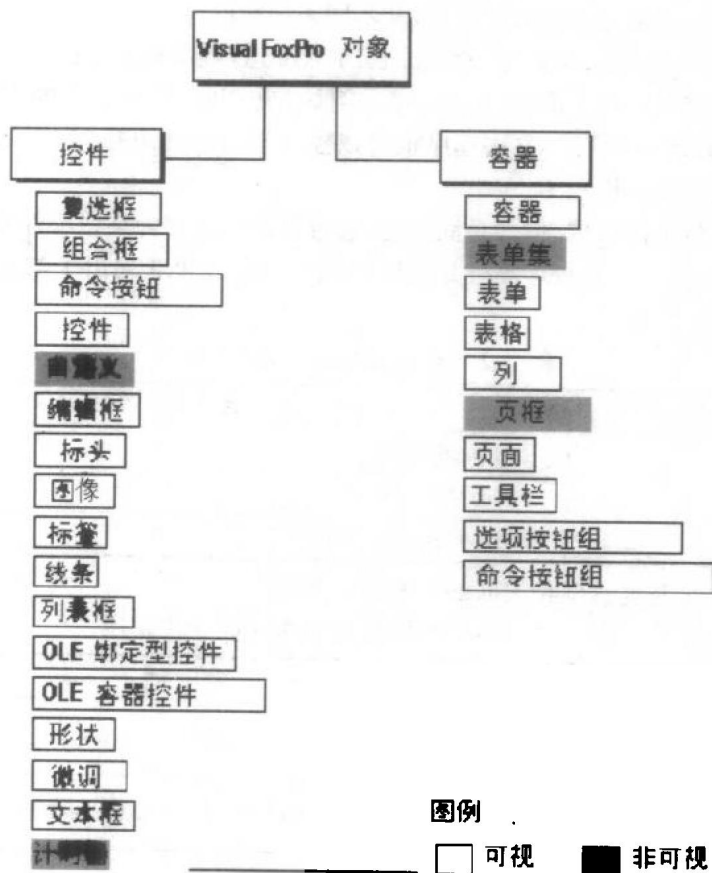


图 1.1 Visual FoxPro 类的层次

其中，标有 * 的这些类是父容器的集成部分，在类设计器中不能基于它们创建子类。所有 Visual FoxPro 基类有如下的最小事件集，如表 1.5 所列。

表 1.5 Visual FoxPro 基类的最小事件集

事 件	说 明
Init	当对象创建时激活
Destroy	当对象从内存中释放时激活
Error	当类中的事件或方法程序过程中发生错误时激活

所有 Visual FoxPro 基类有如下的最小属性集，如表 1.6 所列。

表 1.6 Visual FoxPro 基类的最小属性集

属 性	说 明
Class	该类属于何种类型
BaseClass	该类由何种基类派生而来,例如 Form、Commandbutton 或 Custom 等
ClassLibrary	该类从属于哪种类库
ParentClass	对象所基于的类。若该类直接由 Visual FoxPro 基类派生而来,则 ParentClass 属性值与 BaseClass 属性值相同

注意,如果对象从基类派生而来,那么基类对象的 ParentClass 属性值为空。

2. 容器类与控件类

Visual FoxPro 的类有两大主要类型,因此 Visual FoxPro 对象也分为两大类型:容器类和控件类。

图 1.2 显示了容器类和控件类的区别。

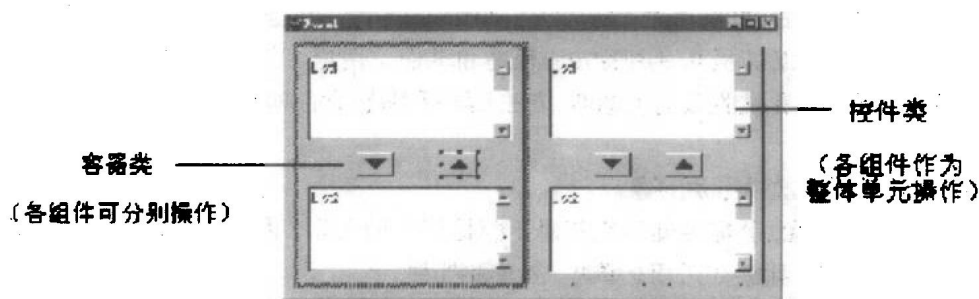


图 1.2 容器类与控件类

容器类可以包含其他对象,并且允许访问这些对象。例如,若创建一个含有两个列表框和两个命令按钮的容器类,而后将该类的对象加入表单中,那么无论在设计时还是在运行时,都可以对其中任何一个对象进行操作。不仅可以轻松地改变列表框的位置和命令按钮的标题,也可以在设计阶段给控件添加对象。例如,可以给列表框加标签,以标明该列表框。

表 1.7 列出了每种容器类所能包含的对象。

表 1.7 每种容器类所能包含的对象

容 器	能包含的对象
命令按钮组	命令按钮
容器	任意控件
控件	任意控件
表单集	表单、工具栏
表单	页框、任意控件、容器或自定义对象
表格列	标头对象以及除表单、表单集、工具栏、计时器和其他列对象以外的任意对象
表格	表格列
选项按钮组	选项按钮
页框	页面
页面	任意控件、容器和自定义对象
工具栏	任意控件、页框和容器