高等学校教育改革推荐教材



Visual FoxPro 8.0 数据库程序设计

温济川　李才有　魏　民　温怀玉　编著
郝文化　主审

中国电力出版社
www.infopower.com.cn

需要全本请访问 www.ertongbook.com

内容提要

本书是按照高等本科院校并结合高职高专学生的培养目标和基本要求，结合作者多年来的教学经验和工程实践，为实施教学改革，使 Visual FoxPro 数据库程序设计面向应用实践，而编写的一本 Visual FoxPro 数据库程序设计基础教材。本书详细介绍了 Visual FoxPro 数据库程序设计的基础知识和应用技术。全书的内容包括 Visual FoxPro 安装与设置、项目管理器、查询、视图、Visual FoxPro 程序设计语言、面向对象程序设计的基本概念、表单设计、报表设计、菜单与工具栏、应用程序的编译与调试、Visual FoxPro Web 功能等，同时在本书附录中给出了 Visual FoxPro 的常用命令以方便读者在使用 Visual FoxPro 进行数据库程序设计时使用。并在每章后都配有上机实战和理论巩固题以实现教与学的统一。

本书语言通俗易懂，内容丰富翔实，突出了以实例为中心的特点，既可作为大学本科和高等学院非计算机专业的教学用书，也可作为广大 Visual FoxPro 数据库程序设计爱好者自学 Visual FoxPro 数据库程序设计技术时的参考用书。

图书在版编目（CIP）数据

Visual FoxPro 8.0 数据库程序设计 / 温济川等编著. —北京：中国电力出版社，2005
高等学校教育改革推荐教材
ISBN 7-5083-2960-0
I.V... II.温... III. 关系数据库-数据库管理系统，Visual FoxPro 8.0-程序设计-高等学校-教材
IV.TP311.138
中国版本图书馆 CIP 数据核字（2004）第 127869 号

丛 书 名：高等学校教育改革推荐教材
书　　名：Visual FoxPro 8.0 数据库程序设计
出版发行：中国电力出版社
地　　址：北京市三里河路 6 号　　邮政编码：100044
电　　话：（010）68358031(总机)　　传　　真：（010）68316497，88383619
本书如有印装质量问题，我社负责退换
服务电话：（010）88515918（总机）　　传　　真：（010）88518169
E-mail：infopower@cepp.com.cn
印　　刷：北京丰源印刷厂
开本尺寸：185×260　　**印　　张**：15　　**字　　数**：335 千字
书　　号：ISBN 7-5083-2960-0
版　　次：2005 年 2 月北京第 1 版
印　　次：2005 年 2 月第 1 次印刷
印　　数：0001—5000 册
定　　价：21.00 元
版权所有，翻印必究

序

党的十六大以来，举国上下都在贯彻人才强国战略。特别是在我国加入 WTO、面临经济转轨的形势下，我国高等教育事业紧扣世界教育发展的脉搏，已从精英教育走向大众教育，进入迅猛发展的时期。

21 世纪是信息技术高度发展的信息时代，要求大学生具有更加丰富的信息技术知识和更强的应用信息技术的能力。选好一本教材，对提高计算机类专业的教学质量至关重要。在确定课程体系以后，最重要的工作就是根据教学要求编写出高质量的教材。在教材建设中，我们提倡百花齐放、推陈出新，经过实践考验，形成公众认可的精品，从而以推动教学质量的提高。

受出版社委托，我们邀请了一些相关高校的教师，召开了一次别具风格的“新形势下高等院校系列教材建设研讨会”，共同研究了国内外高等教育的教学现状与发展趋势，介绍了当前国内高等院校计算机类专业的教学状况与教材建设情况，探讨了新形势下高等院校的教材建设问题，强调了教材要“体现实用性，突出院校自身教学特点，老师易教，学生易学”的编写原则。

在有关专家、教授的亲切指导和热心支持下，在相关高校计算机学院（系）院长（主任）与骨干教师的热情参与下，教材编委会和众多作者在总结了教材建设上多年探索和实践的基础上，广泛汲取了各类成功教材的有益经验，分析了广大学生的承受能力和学习心得，并听取了计算机界教育专家们建议，博采各家所长，精心打造本套教材。本套教材以培养学生的应用能力为目的，突出实用性，突破了传统教材中理论与实践脱节、偏深、偏难的现象，易于实施教学，发掘学生的潜在学习积极性，能让学生在最短的时间内，全面系统地掌握计算机技能。

本套教材作为“新形势下高等院校系列教材”，突出体现了以下特色：

（1）各高校领导十分重视，热心于本套教材的建设，并鼓励教师积极参与，热切希望本套教材真实地反映各高校教学水平、教学特点和最新教研成果；

（2）教材的整个编写过程，自始至终得到有关专家和教授热心、真诚的指导与关怀；

（3）根据各高校与老师的实际需要，与出版社共同规划，共同建设、共同完善教材体系与内容，体现开放与互动交流的宗旨。

这套教材在写法上体现了理论与实践相结合，相关的知识点讲解清晰、透彻，注重教学实际，力求科学实用，符合教学习惯，语言通俗易懂，内容丰富翔实，既注重基本理论及使用方法的深入剖析，又注重实例与技巧的融会贯通。各章后附有课后理论与上机练习，满足学生需要，亦为教师的课堂教学及上机指导提供了有益的参考与帮助。

这套教材是“新形势下高等院校系列教材”，是一种新的尝试。“新”，就会有许多值得修改的地方。本套教材面向各高校，对有志于参与本套教材编写或修改的教师来说是开放的，各学校可以根据自己的特点和教师的特长加以修订和补充。我们热烈欢迎更广泛的学校、教师或作者共同热心参与，更好地规划和完善这套“新形势下高等院校系列教材”。

全国高等学校计算机教育研究会理事长 袁开榜

前　言

Visual FoxPro 数据库是一种典型的关系型数据库管理系统，它最突出的优点在于将可视化设计界面和关系数据库合二为一，用户不须借助任何第三方开发工具就可开发出功能强大的数据库应用系统。Visual FoxPro 提供了丰富的设计工具，如项目管理器、报表设计器、表单设计器、表达式生成器、查询设计器、Web 生成器等，这些为用户开发各种数据库管理系统提供了极大的方便，所以自 Visual FoxPro 问世以来，受到广大用户的欢迎，已经成为一种通用的数据库管理软件。Visual FoxPro 数据库程序设计逐渐成为高等院校各专业的一门重要专业基础课程或选修课程，针对高等院校非计算机专业学生，作者们经过认真收集和整理素材，精心编写了这本与 Visual FoxPro 发展同步的《Visual FoxPro 8.0 数据库程序设计》一书。本书以注重数据库管理系统设计基础知识，注重实际操作，注重数据库管理系统应用为中心，主要讲授 Visual FoxPro 数据库程序设计的基本概念及分类，介绍目前应用较多的 Visual FoxPro 数据库程序设计技术及应用，目的是让非计算机专业学生掌握和了解 Visual FoxPro 数据库程序设计的概念、基本原理及应用技术，能够利用 Visual FoxPro 作为数据库管理系统的开发工具。

主要内容

本书共分为 11 章。其中，第 1 章～第 4 章主要介绍了如何利用菜单进行用户环境设置，如何利用系统提供的各种可视化设计工具设计库、表、查询、视图、报表、表单、菜单等数据库对象。这一部分内容对于初学者学习 Visual FoxPro 系统设计来说具有很大的帮助。如果读者对 Visual FoxPro 已有一定的了解可以略去对此部分内容的学习，直接转入对本书后面知识的学习。第 5 章～第 10 章，主要介绍了 Visual FoxPro 的程序结构、语言语法，使用命令、函数、类、内部变量等知识，这些知识对于学习 Visual FoxPro 数据库程序设计来说极其重要，希望读者在学习此部分知识时认真理会与掌握。第 11 章，介绍了 Visual FoxPro 的网络功能，介绍了如何在网络中发布数据，以及设置网络服务。在本书附录中给出了 Visual FoxPro 的常用命令，以方便读者在使用 Visual FoxPro 进行数据库程序设计时使用。在每章的开头都列出了本章所要求掌握的知识点，在每章结尾都对本章的所学内容进行了相应的总结。为了方便读者在学完本章后，检验学习成果并加深对本章内容的理解，在每章的最后都给出了相应的实践检验题目。

特点

本书用模块化方式深入浅出地讲解了 Visual FoxPro 数据库程序设计的基础知识，以及使用 Visual FoxPro 进行数据库程序设计时的重点与难点。全书重点突出、主次分明、结构清晰、逻辑性强，每章都有知识点、概述、小结、实践检验等配套内容。通过对本书的学习，读者能够在充分掌握 Visual FoxPro 数据库程序设计基础知识的同时，掌握 Visual FoxPro 数据库程序设计技术，并将其尽快运用到实际工作中，从而实现理论与实践的结合、统一。

本书语言通俗易懂，内容丰富翔实，突出了以实例为中心的特点，既可作为大学本科和高等院校非计算机专业的教学用书，也可作为广大 Visual FoxPro 数据库程序设计爱好者自学 Visual FoxPro 数据库程序设计技术时的参考用书。

编写分工

本书由温济川负责前期策划和后期统稿工作。由温济川、李才有、魏民、温怀玉等担任全书的编写工作。郝文化负责全书的审订工作。其中，第 1～3 章、第 9 章由温济川编写，第 4～5 章由李才有编写，第 6～8 章由魏民编写，第 10～11 章由温怀玉编写。同时，参与本书工作的还有邹素琼、王安贵、陈郭宜、程小英、谭小丽、卢丽娟、刘育志、吴淬砺、赵明星、贺洪俊、李小平、史利、张燕秋、周林英、黄茂英、李力、李小琼、李修华等，陈长忆老师为我们提供了宝贵的意见，在此表示衷心感谢。

配套服务

为充分展现本书编写特点，帮助读者深刻理解本书编写意图与内涵，进一步提高对本书教学的使用效率，我们建立了本书使用指导联络方式，它是读者与编者之间交流沟通的桥梁。欢迎读者将图书使用过程中的问题与各种探讨、建议反馈给我们，作者将竭诚为你服务，联系方式 E-mail：bojia@bojia.net。

同时，为了便于多媒体教学，我们提供了本书配套的电子教案，为老师教学提供有益的参考和帮助，只要登录网址：http://www.bojia.net，即可在网站的下载专区免费下载。

作　者

2004 年 12 月

需要全本请访问 www.ertongbook.com

目　录

序
前　言

第 1 章　Visual FoxPro 8.0 安装与设置 …… 1

1.1　Visual FoxPro 8.0 系统安装 …… 1
1.2　设置系统环境 …… 4
1.3　Visual FoxPro 文件扩展名及文件类型 …… 18
1.4　疑难问题解答 …… 19
1.5　本章小结 …… 20
1.6　实践检验 …… 20

第 2 章　项目管理器 …… 21

2.1　创建项目 …… 21
2.2　数据库 …… 23
2.3　表 …… 25
2.4　索引 …… 32
2.5　联系 …… 35
2.6　记录的输入和编辑 …… 38
2.7　上机指导 …… 43
2.8　疑难问题解答 …… 48
2.9　本章小结 …… 49
2.10　实践检验 …… 49

第 3 章　查询 …… 50

3.1　创建查询 …… 50
3.2　创建多表查询 …… 59
3.3　创建交叉表查询 …… 63
3.4　查询的输出 …… 67
3.5　SQL 查询语言 …… 70
3.6　疑难问题解答 …… 75
3.7　本章小结 …… 76
3.8　实践检验 …… 76

第 4 章　视图 …… 77

4.1　创建带参数的视图 …… 77
4.2　远程视图 …… 79
4.3　疑难问题解答 …… 81
4.4　本章小结 …… 82
4.5　实践检验 …… 83

第 5 章　Visual FoxPro 程序设计语言 …… 84

5.1　变量、函数、表达式 …… 84
5.2　Visual FoxPro 的命令格式 …… 89
5.3　数据库的操作命令 …… 92
5.4　表的基本操作 …… 93
5.5　表记录的操作 …… 95
5.6　表的其他操作 …… 107
5.7　多表操作 …… 112
5.8　程序结构及设计 …… 114
5.9　疑难问题解答 …… 123
5.10　本章小结 …… 124
5.11　实践检验 …… 125

第 6 章　面向对象程序设计的基本概念 …… 127

6.1　面向对象程序设计的基本概念 …… 127
6.2　类 …… 131
6.3　对象设计的语法 …… 133
6.4　创建类及对象 …… 135
6.5　疑难问题解答 …… 142
6.6　本章小结 …… 142
6.7　实践检验 …… 143

第 7 章　表单设计 …… 144

7.1 表单设计工具 …… 144
7.2 基本控件 …… 149
7.3 疑难问题解答 …… 176
7.4 本章小结 …… 177
7.5 实践检验 …… 177

第 8 章 报表设计 …… 179

8.1 用报表向导和“快速报表”设计报表 …… 179
8.2 报表的设计 …… 182
8.3 疑难问题解答 …… 184
8.4 本章小结 …… 184
8.5 实践检验 …… 185

第 9 章 菜单与工具栏 …… 186

9.1 菜单设计 …… 186
9.2 设计工具栏 …… 192
9.3 疑难问题解答 …… 195
9.4 本章小结 …… 195
9.5 实践检验 …… 195

第 10 章 应用程序的编译与调试 …… 197

10.1 主程序的管理 …… 197
10.2 应用程序的文件管理 …… 199
10.3 优化应用程序 …… 201
10.4 Visual FoxPro 的调试技术 …… 203
10.5 软件的编译及制作安装盘 …… 206
10.6 疑难问题解答 …… 209
10.7 本章小结 …… 210
10.8 实践检验 …… 210

第 11 章 Visual FoxPro 8.0 Web 功能 …… 212

11.1 数据的 Web 发布 …… 212
11.2 Web 浏览器和 Web 页的创建 …… 216
11.3 WWW 搜索页 …… 220
11.4 发送电子邮件 …… 221
11.5 疑难问题解答 …… 223
11.6 本章小结 …… 223
11.7 实践检验 …… 223

附录 Visual FoxPro 常用命令 …… 225

第 6 章　面向对象程序设计的基本概念

知识点：

- 了解面向对象的思想
- 理解对象、类、封装、继承等基本概念
- 了解面向对象的特征
- 掌握 Visual FoxPro 中面向对象程序设计的基本方法

本章概述：

在本章中将介绍面向对象程序设计的基本概念，以及面向对象程序设计方法在 Visual FoxPro 中的实现。

利用 Visual FoxPro 开发程序，自觉不自觉地就在使用面向对象的程序设计方法，有的读者觉得面向对象的程序设计不如面向过程的程序设计（如 C 语言）方便，甚至难于控制，其原因在于他们对面向对象程序设计的基本概念不清晰，理解不深刻。因此，读者在学习时应重点掌握和深刻理解对象、类、消息、继承等面向对象程序设计的基本概念。

6.1　面向对象程序设计的基本概念

面向对象（Object-Oriented）是人们认识客观世界的一种方法，是一种以对象为中心的思维方式。人们在认识和理解现实世界的过程中，普遍运用以下 3 个法则来认识各种对象：

- 区分不同对象类，如动物与植物是不同类别的生物；
- 区分整体对象及其组成部分，如人的四肢是人这个对象的组成部分；
- 区分对象及其属性，如人的身高是任何人都具有的属性。

面向对象这种认识问题的方法不仅使人们区分了不同的事物，还使人们认识到了事物的本质，即将客观事物从物质和意识两个方面进行认识。物质指的是具体的事物，意识描述的是某一个抽象的概念。例如“张三”和“人”，“张三”是物质，它是具体的客观存在；“人”是意识，它是一个抽象的概念，是对客观存在的事物的一种概括。

6.1.1　面向对象程序设计方法概述

面向对象程序设计（Object-Oriented Programming，OOP）是将人们认识客观世界的方法应用到计算机程序来描述实际问题的思路，因此是一种直观、高效率地解决问题的方法。

面向对象程序设计技术追求的是在软件系统中实现对现实世界的重现，尽量实现将现实世界中的事物直接映射到软件系统中。它将数据及对数据的操作放在一起，作为一个相互依存、不可分割的整体来处理，它采用数据抽象和信息隐藏技术，将对象及对象的操作抽象成一种新的数据类型——类，并且考虑不同对象之间的联系和对象类的重用性。

面向对象程序设计是软件系统设计与实现的新方法，这种新方法既汲取了结构化程序设计的优点，又充分考虑了现实世界与计算机世界的映射关系，其目的是通过增加软件可扩充

性和可重用性，来改善并提高程序员的生产能力，同时降低维护软件的难度。即面向对象程序设计主要是为了解决两个问题：

- 程序代码的重复使用，由此种机制可以增加程序开发的速度；
- 降低维护负担。

1. 面向对象编程方法的基本特征

面向对象的编程方法具有4个基本特征：

（1）抽象。抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不是将对象的所有属性全部提取出来，而是只将感兴趣的内容映射到计算机的解空间。比如，要设计一个学生成绩管理系统，考查学生这个对象时，只关心他的班级、学号、成绩等，而不用去关心他的身高、体重这些信息。抽象包括两个方面，一是过程抽象，二是数据抽象。过程抽象是指任何一个明确定义功能的操作都可被使用者看作单个的实体看待，尽管这个操作实际上可能由一系列更低级的操作来完成。数据抽象定义了数据类型和施加于该类型对象上的操作，并限定了对象的值只能通过使用这些操作修改和观察。

（2）继承。继承是实现代码重用的一种重要方法，当对象的一个新类从现有的类中派生出来后，新类就继承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和变量，它可以修改或增加新的方法使之更适合特殊的需要。这一过程模拟了自然界中生物的遗传特性。比如子女继承（遗传）了父母作为人的一切特性，以及区别于其他父母子女的特性（高矮、相貌等），同时子女也可能具有父母不具备的个性。

（3）封装。封装是面向对象的主要特征之一。封装是把过程和数据包装起来，这时对数据的访问只能通过已定义的界面。面向对象始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口与其他对象进行交流。比如电视机这一对象，其各种功能被封装在机壳中，人们通过特定的按钮实现频道转换、音量调节等功能。对一般用户而言，电视机内部是禁止操作的，这称为信息隐藏。封装保证了模块具有较好的独立性，使得程序维护修改较为容易。对应用程序的修改仅限于类的内部，因而可以将应用程序修改带来的影响减少到最低限度。

（4）多态性。多态性是对继承的一种扩展，它描述的是同一个消息可以根据发送消息对象的不同采用多种不同的行为方式。比如矩形类和圆类都是图形类的子类，由于矩形类和圆类定义了不同的绘图方法，当都收到绘图消息时，两个类产生了不同的结果。又比如，同样的编辑/粘贴操作，在字处理程序和绘图程序中有不同的效果。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好地解决了应用程序函数同名问题。

2. 面向对象程序设计的优点

面向对象程序设计具有许多优点：

（1）开发时间短，效率高，可靠性高，所开发的程序更强壮。由于面向对象编程的可重用性，可以在应用程序中大量采用成熟的类库，从而缩短了开发时间。

（2）应用程序更易于维护、更新和升级。继承和封装使得应用程序的修改带来的影响更加局部化。

6.1.2 对象

对象是面向对象系统中的核心概念，如果不能正确地认识和定义它，就无法掌握面向对

象的理论。

对象具有两方面的含义，在现实世界中存在的一切事物都可称为对象，例如一个人、一件物品，甚至一个地球都是对象。除去这些可以触及的事物是对象之外，还有一些无法整体触及的抽象事件，例如一次演出、一场球赛、一次借书，也都是对象。

一个对象既可以非常简单，又可以非常复杂，复杂的对象往往是由若干个简单对象组合而成的。

所有的这些对象，除去它们都是现实世界中所存在的事物之外，它们都还具有各自不同的特征。例如一粒米，它首先是以一粒米客观存在的，其次它还具有颜色、体积、重量，再其次它还可以被食用，可以用作原料等。也就是说一粒米是一个具有自身状态和自身功能的客观实体。再例如一个人，首先它是一个客观实体，具有一个名字来标识，其次它具有性别、年龄、身高、体重等这些体现他自身状态的特征；再其次他还具有一些技能，例如会说英语、会修电器等。

接下来给“对象”下一个定义，即对象是现实世界中的一个实体，它具有如下特性：

- 用一个名字来将其与其他对象加以区别；
- 有一个状态用来描述它的某些特征；
- 有一组操作，每一个操作决定对象的一种功能或行为；
- 对象的操作可分为两类：一类是自身所承受的操作，一类是施加于其他对象的操作。

对象是具有自身状态特征及可以对这些状态施加的操作结合在一起所构成的独立实体。例如有一个人名叫张三，男性，身高 1.70m，体重 65kg，会打篮球、开汽车，这个“对象”用以下方式描述：

对象名：张三

对象的状态：

性别：男

身高：1.70 m

体重：65kg

对象的功能（可做的操作）：

打篮球

开汽车

而计算机中的对象是一些可操作的、可见的类的实体。它既具有各方面特征的数据（即属性），又具有完成某种操作的事件代码（即方法程序），是一个带有操作程序的逻辑单元。对象具有它特定的名称。

对象将数据和方法封装在一起，使这些个体变得独立（封装性），当外界与它发生关系时，必须通过预先设计好的渠道进行，这个渠道就是方法。

Visual FoxPro 中的对象是以预定义类及其实例的形式出现，将在 6.2 节“类”和 7.2 节“基本控件”中对 Visual FoxPro 的对象进行详细讲解。

6.1.3 类

类是对象的抽象，是具有共同属性、共同方法、共同事件的对象的集合。

在现实世界中，人们通过对一组对象的观察，将该组对象所具有的共同特征（包括结构特征和行为特征）集中起来，以说明该组对象的特征和性质。比如，通过对许多不同大小、

肤色的马加以归纳总结，抽象出各种马的共同特征，从而得到马类。

在面向对象程序系统中将具有相同操作功能和相同数据格式（属性）的对象的集合称为类。类形成了一个具有特定功能的模块和一种代码共享的手段。它为程序员提供了一种工具，使得他们可以方便地建立所需要的任何类型，并可以像一般类型那样方便地使用。

类与对象的区别：类与实例之间的关系可以看成是抽象与具体的关系。类是抽象的，不能在程序中具体存在，在程序中存在的是对象。但对象是由类产生的，没有相应的类就不可能产生相应的对象。因此，类是产生对象的模板，而由类产生的对象则称为类的实例。

例如苹果是一个抽象的概念，但当人们看到一个红苹果时就会说："这是一个苹果。"把它换成面向对象系统的语言来叙述："这是一个苹果类的实例。"再如，张三是一个学生，学生是一个类，而张三作为一个具体的对象，是学生类的一个实例。

对于同一类的不同实例之间，必定具有如下特点：

- 相同的操作集合;
- 相同的属性集合;
- 不同的对象名。

6.1.4 方法

方法是对象所具有的动态特性，是对象内在的运动规律。

当对象通过接口接收到消息的时候，对象自动调用相应的方法对消息进行处理。比如电视机这一对象，当通过遥控器上的按钮进行频道选择（发送消息）时，选择信号通过电视机上的红外线接口传送给电视机的接收装置，接收装置自动启动频道选择电路（调用方法），改变当前频道。

计算机中的每个对象都是为了一定的目的而设计的，其方法是经过精心设计的程序，由用户或系统在特定条件下激活。

同一类对象的方法是一样的，但用户可以为每一个对象自定义方法。

6.1.5 属性

属性是对象所具有的静态特性，是对象的外部表现。

所谓属性就是对象的大小、颜色、位置等能让人感知的、让人识别的区别于其他对象的特性。比如电视机的屏幕尺寸为 29in、34in，电视机的颜色等是电视机的属性。

同一类的不同对象其属性不能完全相同，而不同类的对象的部分属性可以相同。

计算机中对象的属性可以根据用户的需要进行添加。

6.1.6 事件

对象的方法是由外界或自身在某种条件下激活的，这一条件就称为对象的事件。

例如，对一台电视机来说，当用户按下遥控器的按钮时，便激发了一个事件，按下电视机的关闭按钮就激活了电视机结束工作的方法。

计算机中对象的事件是一种预先定义好的特定动作，由用户或系统激活。每个对象都可以对其进行识别和响应。大多数情况下，事件是通过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、移动鼠标和按键等。还有一些事件是由系统激发的，如窗体初始化、窗体关闭等。

需要全本请访问 www.ertongbook.com

6.2 类

从 6.1 节的分析知道了类是面向对象程序设计的核心，以及类的基本概念和类的特点，本节介绍 Visual FoxPro 中的类。

6.2.1 基类

基类（Base Class）即基本的类，所有用户自定义的类都是由基类派生出来的，所有对象都是基类及其派生类的实例。

基类由 Visual FoxPro 预定义。即安装完 Visual FoxPro，基类就已经在计算机中，用户可通过 Visual FoxPro 使用这些类。

根据类不同的性质，一般将类分为容器类和控件类两大类，图 6-1 显示了 Visual FoxPro 基类的情况。其中，可视类表示在程序运行过程中其类的实例（对象）是可以看见的，而不可视类则不能看见，它们完全由方法和属性组成。

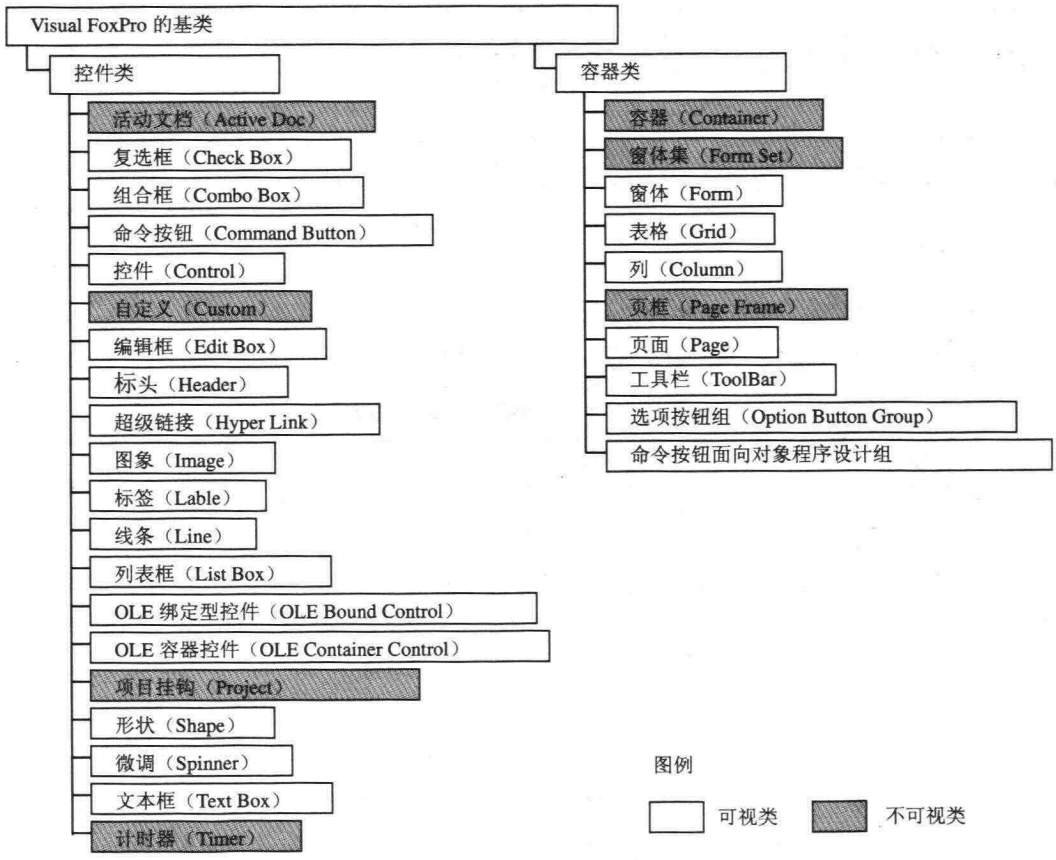

图 6-1 Visual FoxPro 中的类

6.2.2 容器类

容器类是指可以包含其他对象的类，即盛放其他对象的容器，Visual FoxPro 中所有的容

器类如图 6-1 所示。但是不同容器类所能包含的对象是不同的，表 6-1 列出了每种容器类所能包含的对象。

表 6-1　容器及其所包含的对象

容　器	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
自定义	任意控件、页框、容器和自定义对象
窗体集	窗体、工具栏
窗体	页框、任意控件、容器或自定义对象
表格列	表头和除窗体集、窗体、工具栏、计时器和其他列以外的其余任一对象
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
项目	文件、服务程序
工具栏	任意控件、页框和容器

6.2.3　控件类

控件类是指可以包含在容器类中的 Visual FoxPro 基类。

表 6-2 列出了 Visual FoxPro 中常用控件类的作用。

表 6-2　Visual FoxPro 中的控件类

控 件 类	说　明
复选框	创建一个复选框
组合框	创建一个组合框
命令按钮	创建一个单一的命令按钮
编辑框	创建一个编辑框
选项按钮	创建一个单一的选项按钮
文本框	创建一个一个文本框
微调	创建一个微调钮
标签	创建一个用于显示正文内容的标签
图象	创建一个显示.BMP 文件的图像控件
线条	创建一个能显示水平线、垂直线或斜线的控件
形状	创建一个显示方框、圆或椭圆形状的控件
计时器	创建一个能够按规则执行代码的计时器

6.3 对象设计的语法

程序开发的过程其实质上就是对象设计和操作的过程，即对对象的方法、属性进行定义。对象的设计和操作可以在设计时刻和运行时刻两种方式下进行。“设计时刻方式”是指在Visual FoxPro 的编程环境下对控件的属性、方法进行定义；“运行时刻方式”是指在用户程序运行过程中，通过对象的方法来定义控件的属性。

6.3.1 对象引用的语法

对象的引用就是使用对象，是操作对象的基础。

要使用一个对象，需要知道它相对于容器分层结构的关系。例如，如果要在窗体集中处理一个窗体的控件，则需要引用窗体集、窗体和控件。在容器层次中引用对象恰似给 Visual FoxPro 提供这个对象地址。例如，给他人讲述自己的详细住址时，需要依次指明自己所在的国家（地区）、省份（州）、城市、街道，甚至具体的门牌号码，否则将引起混淆。如图 6-2 表示了一种可能的容器嵌套方式。

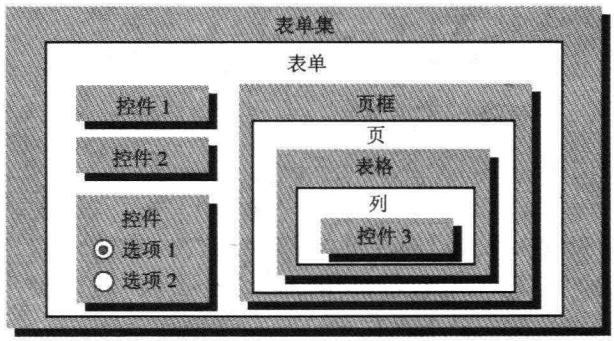

图 6-2 容器嵌套方式

例如要引用图 6-2 中的控件 3，其引用语法如下：

语法：窗体集名.窗体名.页框名.页名.表格名.列名.控件 3

以上语句中对象名之间的“.”称为成员操作符，表明对象之间的层次关系，比如 Form1.Command1 表示“窗体 Form1 中的按钮 Command1”。

由于对对象的引用总是在某一对象的方法程序或事件代码中实现的，因此可以更加简明地引用对象，这种引用方式叫“相对引用”。相对引用用到的关键字及含义如表 6-3 所示。

表 6-3 相对引用关键字

关 键 字	引 用
PARENT	该对象的直接容器
THIS	该对象
THISFORM	包含该对象的窗体
THISFORMSET	包含该对象的窗体集

【例 6-1】对象引用。

下面一段程序说明了引用对象的方法及引用语句存在的位置。

```
THISFORMSET.frm1.cmd1.Caption = "OK"    &&引用窗体集中 frm1 窗体的 cmd1 控件的
                                        &&Caption 属性，语句位置在此窗体集的任意
                                        &&窗体的任意控件的事件或方法程序代码中
THISFORM.cmd1.Caption = "OK"            &&引用窗体中 cmd1 控件的 Caption 属性，语
                                        &&句位置在 cmd1 所在的同一窗体的任意控件的
                                        &&事件或方法程序代码中
THIS.Caption = "OK"                     &&引用控件自身的 Caption 属性，语句位置在
                                        &&控件的事件或方法程序代码中
THIS.Parent.BackColor = RGB(192,0,0)    &&引用控件的容器对象的 BackColor 属性，语
                                        &&句位置在窗体的一个控件的事件或方法程序
                                        &&代码中
```

6.3.2 对象属性设置语法

引用的目的之一就是对对象的属性进行定义，既可以在设计时刻又可以在运行时刻进行属性定义。设计时刻对象属性的改变，将在 7.2 节“基本控件”中讲解。

属性定义如下所示：

语法：容器.对象.属性=属性值

【例 6-2】下列语句设置 frmPhoneLog 窗体中 txtDate 文本框的各种属性：

```
frmPhoneLog.txtDate.Value = DATE()                  && 文本框的内容为显示当前日期
frmPhoneLog.txtDate.Enabled = .T.                   && 设置文本框控件有效
frmPhoneLog.txtDate.ForeColor = RGB(0,0,0)          && 设置文字颜色为黑色文本
frmPhoneLog.txtDate.BackColor = RGB(192,192,192)    && 设置文本框为灰色背景
```

当需要为一个对象设置多个属性时，可以利用 WITH … ENDWITH 语句结构。

【例 6-3】在窗体集的窗体中，要设置表格列的多个属性，可以使用以下的语法结构：

```
WITH THISFORMSET.frmForm1.grdGrid1.grcColumn1
.Width = 5
.Resizable = .F.
.ForeColor = RGB(0,0,0)
.BackColor = RGB(255,255,255)
.SelectOnEntry = .T.
ENDWITH
```

6.3.3 事件执行语法

事件是系统定义的激活对象方法的某种条件，一般情况下这些条件由用户交互触发，如用户单击鼠标触发 Click 事件、击键触发 KeyPress 事件等；或由系统自动触发，如调用一个窗体会触发该窗体的 Activate 事件、关闭窗体会触发该窗体的 Destory 事件等。当事件被触发，事件的对应代码（既可以从父类继承来，又可以由用户自己编写）将被自动执行。

在需要的情况下，可以用编程方式来模拟事件的触发，可以模拟事件的命令有如下几条：

1. MOUSE 命令

MOUSE 命令用于产生 Click、DblClick、MouseMove 和 DragDrop 事件，模拟单击、双击、移动和拖动鼠标。

在下面的示例中，第一个命令在 Visual FoxPro 主窗口的第 3 行第 3 列单击鼠标；第二个

命令在 Visual FoxPro 主窗口的第 3 行第 3 列单击鼠标，然后把鼠标指针拖动到 Visual FoxPro 主窗口的第 10 行第 10 列；第三个命令把鼠标指针从当前位置拖动到第 20 行第 20 例；第四个命令在当前位置双击鼠标，然后把鼠标指针拖动列第 30 行第 30 列；最后一个命令把鼠标指针从当前位置拖动到第 10 行第 10 列，然后拖动到第 30 行第 30 列。

```
MOUSE CLICK AT 3,3
MOUSE CLICK AT 3,3 DRAG TO 10,10
MOUSE DRAG TO 20,20
MOUSE DBLCLICK DRAG TO 30,30
MOUSE DRAG TO 10,10,20,20,30,30
```

2. ERROR 命令

ERROR 命令用于产生 Error 事件，产生一个错误信息；

下面示例生成“密码错误”错误信息。

```
ERROR '密码错误'
```

3. KEYBOARD 命令

KEYBOARD 命令用于产生 KeyPress 事件，模拟击键。

下面示例模拟按下 Enter 键。

```
KEYBOARD 'ENTER'
```

6.3.4 方法引用语法

如果对象已创建，便可以在应用程序的任何一个地方调用这个对象的方法程序。其调用语法如下：

语法：容器.对象.方法

下例调用方法程序来显示窗体，并将焦点设置在文本框上：

```
frsFormSet.frmForm1.Show                    &&显示窗体集 frsFormSet 中的 frmForm1 窗体
frsFormSet.frmForm1.txtGetText1.SetFocus &&将焦点放置到 frmForm1 窗体 txtGetText1
```

在表达式中，有返回值的方法程序必须以圆括号结尾。例如，下列语句将用户自定义的 GetNewCaption 方法程序的返回值设置成窗体 Form1 的标题：

```
Form1.Caption = Form1.GetNewCaption()
```

传递给方法程序的参数必须放在方法程序名后面的圆括号中。例如下列语句将变量 nStyle 传递给窗体 Form1 的 Show 方法程序代码：

```
Form1.Show(nStyle)
```

6.4 创建类及对象

Visual FoxPro 提供了丰富的基类，用户可以利用它来完成大多数的应用程序的设计过程，但这些基类的属性、事件方法是固定的，用户对其控制不够灵活，因此创建用户自己的类可以按用户自己的意愿来设置类的属性、方法，并且针对某些特定的操作，可以通过类重用代码，提高编程效率。我们可以将系统提供的基类比作建造房屋的水泥、钢筋、砖等原材料，而自定义类可比作各种预制件，可以使房屋建造更加方便、快捷。

6.4.1 自定义类

自定义类可以通过编程的方法来进行，也可以利用类设计器和窗体设计器可视地进行。

1. 编程创建自定义类

创建类的命令是 DEFINE CLASS，以下是其命令格式：

```
DEFINE CLASS 新类名 AS 父类名 [OF ClassLibrary] [OLEPUBLIC]
   [IMPLEMENTS cInterfaceName [EXCLUDE]
IN TypeLib | TypeLibGUID | ProgID ]
    [[PROTECTED | HIDDEN] 属性名 1,属性名 2…]
       [对象.]属性名 = 设置的属性值…]
    [ADD OBJECT [PROTECTED] 对象名 AS 对象的类名
       [NOINIT][WITH 对象属性列表]]…
    [[PROTECTED | HIDDEN] FUNCTION | PROCEDURE 过程名[_ACCESS |_ASSIGN]|
    THIS_ACCESS [NODEFAULT]
       过程代码
    [ENDFUNC | ENDPROC]]…
    [PEMName_COMATTRIB = nFlags | DIMENSION PEMName_COMATTRIB[ 5 ]
       [PEMName_COMATTRIB[1] = nFlags
       PEMName_COMATTRIB[2] = cHelpString
       PEMName_COMATTRIB[3] = cPropertyCapitalization
       PEMName_COMATTRIB[4] = cPropertyType
       PEMName_COMATTRIB[5] = nOptionalParams]]
ENDDEFINE
```

命令中的的关键字用大写英文表示，常用子句的参数部分用汉字标出，即汉字部分是定义者需要输入的部分。下面通过几个例子来理解该命令的使用。

【例 6-4】类定义 1。

```
DEFINE CLASS MyClass1 AS FORM
    Caption = "我的窗体"
    Width = 400
    Height = 300
ENDDEFINE
```

该例定义了一个基于窗体的子类，同时还定义了该窗体类的默认名称和尺寸。

提示：该命令定义的是类而不是对象，其属性值是类的默认值，当由该类定义对象时，属性值可以改变。

【例 6-5】类定义 2。

```
DEFINE CLASS MyClass2 AS Custom
    PROTECTED Score
    Name = SPACE(6)
    Sex = .F.
    Age = 20
    Score = 80
ENDDEFINE
```

该例定义了一个基于自定义类的子类，这一自定义类指一个不可见的用户自定义类。并且属性 Score 是保护属性，使其在运行时不能被改变。其他属性的值可以在运行时和设计时

需要全本请访问 www.ertongbook.com