计算机实用软件丛书

Visual FoxPro 5.0

中文版面向对象程序设计详解及实例

彭江平　王毅　李静　倪芳　编著

人民邮电出版社
PEOPLE'S POSTS & TELECOMMUNICATIONS PUBLISHING HOUSE

计 算 机 实 用 软 件 丛 书

Visual FoxPro 5.0 中文版
面向对象程序设计详解及实例

彭江平 王毅 李静 倪芳 编著

人民邮电出版社

图书在版编目(CIP)数据

Visual FoxPro 5.0 中文版面向对象程序设计详解及实例/彭江平等编著
—北京：人民邮电出版社，1998.8
（计算机实用软件丛书）
（ISBN 7-115-06913-1）

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

中国版本图书馆 CIP 数据核字(98)第 10979 号

内 容 提 要

本书主要讨论 Visual FoxPro 5.0 中文版面向对象的程序设计，共分八章，第一章与第二章介绍有关面向对象程序设计的基础知识及与面向对象程序设计技术紧密相关的事件驱动程序的基础——事件模型；第三章介绍如何设计菜单及在菜单设计中事件循环机制的应用；第四章介绍了应用 Visual FoxPro 5.0 的基类对象进行面向对象程序设计的方法；第五章与第六章介绍如何建立自定义类并应用自定义类简化应用程序设计；第七章讨论了开发应用程序时应该注意的若干问题；第八章则重点讨论如何对应用系统进行调试、优化及如何发布应用程序。

本书对面向对象的程序设计进行了较深入的讨论，语言通俗，实例丰富，是一本学习 Visual FoxPro 5.0 面向对象程序设计的较系统的参考书，对初学面向对象程序设计者及有经验的程序设计人员都有一定的参考价值。

计算机实用软件丛书

Visual FoxPro 5.0 中文版
面向对象程序设计详解及实例
Visual FoxPro 5.0 Zhongwenban
Mianxiang Duixiang Chengxu Sheji Xiangjie Ji Shili

◆ 编　　著　彭江平　王毅　李静　倪芳
责任编辑　李振广

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：22.75
字数：570 千字　　1998 年 9 月第 1 版
印数：1－6 000 册　　1998 年 9 月北京第 1 次印刷

ISBN 7-115-06913-1/TP·602

定价：30.00 元

“计算机实用软件丛书”编委会

丛书前言

随着计算机、通信和信息技术的迅速发展与广泛应用，人类正在进入信息化社会。计算机技术的应用与推广，将直接推动社会信息化的发展；而计算机技术的应用与推广，实质上取决于计算机软件的应用和推广，可以说，没有软件，就没有计算机的应用；学习、使用计算机，从根本上讲就是学习和掌握软件的使用。

为了适应当前计算机技术发展的需要，满足读者学习、使用计算机软件的需求，人民邮电出版社约请有关专家编写出版了这套“计算机实用软件丛书”。

这套丛书的特点是：普及兼顾提高，应用兼顾开发，各书独立成册形成系列，并注重其相关性，使丛书成为广大计算机应用和开发人员学习使用计算机的必备用书。

这套丛书的内容包括：程序设计语言、操作系统技术、数据库技术、软件开发技术及工具、网络技术、多媒体技术等。

在计算机技术飞速发展的今天，软件产品更新快，经常有新产品或新版本问世，因此我们不但介绍当前流行和优秀的软件，而且力求尽快把国内外最新的软件产品也介绍给读者。

我们将全心全意为读者服务，也热切期待广大读者对丛书提出宝贵意见，以进一步提高丛书的质量。让我们共同努力，为提高我国的计算机开发、应用水平做出贡献。

“计算机实用软件丛书”编委会

前言

中文 Visual FoxPro 5.0 既继承了以前 FoxPro 2.X 的易学的特点，又具有图形化、可视化的界面设计的能力，同时还支持面向对象的程序设计，无论是对初学者，还是有经验的开发人员，它都是一个非常有力的工具。它必将成为最流行的关系数据库系统之一。

在作者看来，对于初学面向对象程序设计的人员， Visual FoxPro 5.0 是一个最好的工具，它既有大多数面向对象程序设计语言的功能，又直观易学，远没有 C++语言等其它面向对象程序设计语言那样复杂；随着它与大型数据库系统的集成功能的加强，完全可能成为开发其它大型数据库应用系统的优秀的前端开发工具，因而对于有经验的开发者，使用它也是一个值得考虑的选择。

本书内容分为四个部分。

第一章与第二章介绍有关面向对象程序设计的基本概念及与面向对象程序设计紧密相关的事件驱动程序设计的基本内容。

第三章与第四章介绍 Visual FoxPro 5.0 的面向对象程序设计的机制。第三章介绍了菜单设计器的使用及事件循环的实现；第 4 章用大量实例介绍了应用 Visual FoxPro 5.0 的基类对象进行面向对象的程序设计过程。

第五章与第六章重点介绍如何在基类的基础上建立自定义类，并应用它们来简化应用程序的设计的多种方法。

第七章与第八章主要介绍应用系统的设计有关的几个方面。

参加本书的编写人员有：彭江平、王毅、李静与倪芳。其中，第一、二、三章由李静编写，第四章由王毅编写,第五、六、七章由彭江平编写，第八章由倪芳编写。本书由彭江平主编，王毅担任副主编。

由于作者水平有限，再加上时间仓促，书中错误和不足之处，敬请读者批评指正。

作者

目录

第一章

面向对象程序设计基础

面向对象技术可以说是软件工程学方面的一次重大革命，它为日益困难的软件维护工作提供了很好的解决方法，同时也为软件设计者提供了重用组件的解决方案。本章首先简要地介绍有关面向对象程序设计的发展历程及一些基本概念，然后针对Visual FoxPro 5.0,讨论了在这一特定语言环境下的一些相对应的概念。

1.1 面向对象技术的形成与发展

1.1.1 软件危机

随着软件系统规模的扩大和复杂性的增加，软件的开销（开发时所耗费的人力、物力和运行时所占用的硬件资源和运算时间）也在惊人地增加，软件系统的可靠性和可维护性却在明显地降低。人们把这种情况称为软件危机。产生这种危机的原因，在于我们设计开发系统的方法和认识系统的方法不一致。

1.1.2 认识系统的方法与面向对象技术

人们认识世界是一个渐进的过程，是在继承了以往的有关知识的基础上、多次迭代往返而逐步深化的，既包括了从一般到特殊的演绎思维过程，也包括了从特殊到一般的归纳思维过程。而在面向对象技术形成之前，用于分析、设计和实现一个系统的过程和方法大部分是“瀑布”型的，这种方法强调自顶向下按步就班地完成软件开发工作。因此，当越接近系统设计的后期时，对系统设计前期的结果进行修改困难就越大，所以我们的系统开发设计经常出现一些“遗憾工程”。造成这种“遗憾”的原因是由于我们认识系统的方法和设计系统的方法的不同。那么，是否可以按照人们认识系统的过程去设计系统呢？

面向对象的基本方法学认为：客观世界是由许多各种各样的对象所组成，每种对象都有各自的内部状态和运动规律，不同对象间的相互作用和相互联系就构成了各种不同的系统，构成了我们面对的客观世界。

面向对象技术所使用的思维方法已在很多领域中充分使用，如一台机器由若干个集成部件组成，每一部件都有自己的内部状态和操作规则，各部件之间的也存在着相互的作用和联系。这种思维方法在计算机软件中的应用，给计算机软件的发展带来了质的飞跃，它几乎遍及计算机软件技术的各个领域，不仅有面向对象的程序设计，而且有面向对象的语言、面向对象的数据库和面向对象的操作系统等。

1.1.3 传统结构化方法和面向对象方法的比较

传统的结构化方法强调“过程”，虽然一个系统可以由若干个模块组成，但每一个模块都是一个过程，模块与模块之间的联系也是一个过程，并且这些模块的独立性较差，模块之间存在着相互的依赖性。这种方式以“过程”为中心，而这种过程使得模块缺乏灵活性。

面向对象的方法是以“对象”为中心，一个系统是由若干个对象构成。虽然我们也可以将每一个对象看作一个模块，但这种模块与结构化方法中的模块有着很大的区别。它的每一个对象是一个可以独立存在的实体，对象的内部状态以及实现功能的方法是隐藏的，不受外界的影响。由于模块之间的联系通过消息传递和接受实现，所以使得各模块之间依赖性很小。

结构化程序设计与面向对象程序设计有许多不同之处。

1. 设计方式

所谓设计方式，是指对问题处理的程序设计方式。一般在过程性的设计思想中，是针对问题采用顺序处理方式来解决问题。这种设计方式，采用了如下所述的标准处理过程：

(1) 采用顺序性的程序处理问题；

(2) 采用逻辑概念设计程序文件；

(3) 在解决问题的过程中，常采用调用子过程文件。

而面向对象程序设计方法采用对象为设计中心，也就是说：

(1) 不采用顺序性的处理方式，而采用了对象本身的属性与方法来解决问题；

(2) 在处理问题过程中，可以直接在对象中设计事件程序，直接触发问题。

2. 数据处理方式

过程化程序设计在进行数据处理时，基本上是通过程序对数据处理，简单地说，就是程序与数据是分开的，数据完全暴露在外面，程序处理完后，再将结果显示出来。

若采用面向对象的设计方法，我们可以将记录的字段内容连接在编辑对象中，所以数据是隐藏的。

3. 运行方式

过程化设计方法对程序的处理方式，是采用单一程序文件方式，也就是说常常通过程序之间的调用完成程序运转。而面向对象程序设计方法的程序的运行是不同对象之间的消息传递。

4. 使用方式

这里指的使用方式，是表示该程序是否可以重用。一般在过程性程序中，除非设计成公用模块，否则其重用性并不很高。但是若设计成公用程序模块，几乎很难再扩展为更复杂的处理方式，除非再修改形成一个公用模块。

在面向对象程序设计中，类可以重用。对象或类本身具有继承性，并且在可视化工具中，直接用控制点取方式快速定义重复的对象与方法，这是在过程性方法中非常难设计的一种结构，几乎不可能。

面向对象程序设计对象的可重用，大大减少了重复设计的时间，让系统开发更具有生产率。

5. 处理顺序

过程性方法的处理具有顺序性，也就是说采用了程序本身的“直线型”处理，因此使用者必须在程序员所设计的程序下，进行操作，无法自由导航。面向对象程序的结构，虽然在设计过程中，将对象进行编号，但是这种设计，只是为了让用户用Tab或Enter键操作时，能快速地在顺序编号上移动。

为了更好地说明这两类程序设计方法的区别，我们用一个简单而又不失一般性的例子来说明两类方法。整个工作任务是维护一组数据，而且假定要对数据执行如下的操作：

（1）打印数据；

（2）获取数据项数目；

（3）获取数组的最大和最小者；

（4）按从大到小的顺序排序。

如果用结构化的程序设计方法，需要用一个数组来存储这组数据，用几个过程（或函数）来分别实现上述各项功能。这就需要把该数组作为一个参数传递给每一个过程，而在每一个过程中都要有完成这一任务的代码。如果必要，还要把结果作为返回值。这个设计的过程大体如图1.1所示。在这里，该组数据和对它的操作的各个过程是相互独立的。

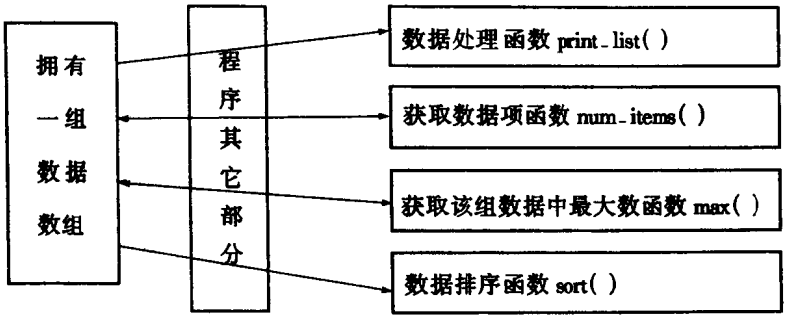

图1.1 维护一组数据的结构化程序设计框图

一个面向对象的程序需要定义一个类，该类包括存储这组数据的结构和处理这组数据的方法（即要执行的操作）。这组数据仍然以数组的形式存储，并且完成各种动作的代码与在结构化程序设计中的代码没有太大的区别。主要的区别在于：

数据与操作数据所需的代码并不是独立的，它们是同一对象的组成部分。一个类中的数据变量称为实例变量，各个过程称方法。图1.2给出了该例的面向对象的结构。

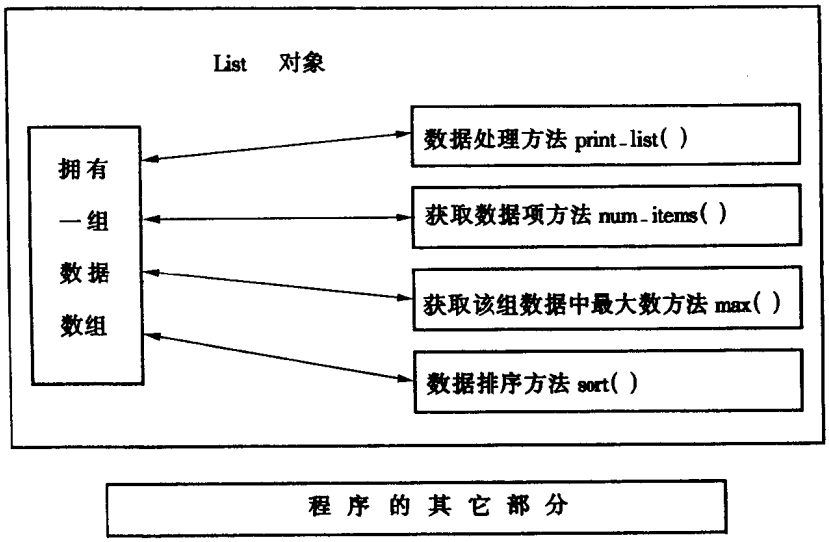

图1.2 维护一组数据的面向对象程序设计结构

1.2 面向对象的有关概念

面向对象技术为软件开发提供了一种新的方法学,引入了许多新的概念,这些概念是理解和使用面向对象技术的基础和关键。

1.2.1 对象（Object）

对象是面向对象的软件技术中最基本、最核心的概念，对象就是针对某一状态的一组操作的集合，也就是说，它是由描述该对象内部状态的数据以及可以对这些数据施加的所有操作封装在一起构成的统一体。

在应用领域中凡是有意义的、与所要解决的问题有关系的任何事物都可以作为对象，它可以是具体的物理实体的抽象，也可以是人为的概念，或是任何有明确边界和意义的东西，例如，一名职工、一家公司、一座图书馆、一本书、贷款、借款等等，都可以作为一个对象。总之，对象是对问题域中某个客体的抽象，设立某个对象就反映了软件系统保存其有关的信息并且与之进行交互的能力。

通常，描述对象内部状态的数据又称为对象的静态属性，对这些数据可以施加的操作又称为对象的动态行为。有时我们把对对象的操作称为方法或事件。

1.2.2 封装(Encapsulation)

面向对象技术将数据与处理代码组合在一个类的定义中，这种组合方式称为“封装”。它是面向对象设计的核心技术。一个对象（即为类的一个实例）中的数据和代码相对于程序的其余部分是不可见的，它能防止那些非期望的交互和非法的访问。只有那些在类的声明中说明为Public（公有的）类型的变量或方法才可以被程序的其它部分访问。公有的方法或变量是一个类与外部的接口，程序的其它部分通过这个接口使用这个类的功能。

因为程序的其它部分只知道类的接口，所以不必担心改变类的其它部分会对整个程序产生非期望的影响。只要保持类的接口不变，该类的内部工作就可以随意改变，把它插入到程序中去，而不必担心会产生副作用。

在Visual FoxPro 5.0中，系统本身内置了一定数量的基类，这些类基本上能满足应用系统的需要。另一方面，还允许用户在这些类的基础上创建自定义类。

1.2.3 继承（Inheritance）

在面向对象的程序设计中，继承这个词指的是在已有类的基础上建立新类的能力。新类自动地拥有父类的所有元素：方法与实例变量，然后再根据需要添加新任务所需的方法或实例变量。

继承是一个有力的工具，使用得当可以不必重写代码。一旦在一个类的定义中实现了一个特别的功能，那么在它的派生类中就可以重复地使用这一功能。不但可以建立已有基类的派生类，而且可创建基于自己所创建类的派生类。

基类是指不是由其它类而派生的类。一个派生类的最近父类叫做该类的父类或超类。从某个类派生的类叫该类的子类。例如，如果A类是由B类派生的，则B是A的超类，而A是B的子类。两者之间的关系是相互的。

并不强迫子类不加改变地使用超类的所有方法，子类可以重置超类的任何方法。这一特征使程序设计者可以根据实际的需要定制子类。当然，这并不意味重置越多越好，因为如果重置太多的话，还不如建立一个新类。

1.2.4 多态性(Polymorphism)

在面向对象程序设计中，多态性是指用同一个名称可以调用不同的方法。具体调用哪一个方法取决于所传递对象（作为参数）的类型。例如，可能有两个方法都称为Print()，一个是在屏幕上显示字符，另一个是用于显示一个位图对象。当调用Print()方法的时候，到底激活哪一个方法取决于传递的参数类型是字符对象还是位图对象。

多态性有时也称为方法的重载。方法重载是指同一个方法名（在上面的例子中就是Print()方法）在上下文中有不同的含义。

1.2.5 面向对象

为了回答“什么是面向对象的软件技术？”这个问题，首先应该了解面向对象方法学的以下几个要点：

（1）客观世界是由各种对象组成的，任何事物都是对象，复杂的对象可以由比较简单的对象以某种方式组合而成。按照这种观点，可以认为整个世界就是一个复杂的对象。因此，面向对象的软件系统是由对象组成的，软件中的任何元素都是对象，复杂的软件对象由比较简单的对象组合而成。

（2）把所有对象都划分成各种对象类（简称为类），每个对象都定义了一组数据和一组方法。数据用于表示对象的静态属性。因此，每当建立该对象类的一个新实例时，就按照类中对数据的定义为这个新对象生成一组专用的数据，以便描述该对象独特的属性值。例如，屏幕上不同位置显示的半径不同的几个圆，虽然都是 Circle 类的对象，但是，各自