

21 世纪计算机编程指南系列



本书配套光盘内容包括：
与本书配套的电子书



北京希望电脑公司 总策划
李春葆 编 著

中文版 Visual FoxPro 6.0 高级实例开发指南

21 世纪计算机编程指南系列



本书配套光盘内容包括：
与本书配套的电子书



北京希望电脑公司 总策划

李春葆 编 著

光盘

中文版

Visual FoxPro 6.0

高级实例开发指南

TP31-62/143

内容简介

本书是作者在长期实际软件开发的基础上编写的,力求让需要了解 Visual FoxPro 基本编程技术的读者快速升级到专业级开发水平。本书内容包括面向对象的程序设计、基本控件设计、系统封面设计、数据表操作设计、报表设计、菜单设计和应用实例设计等7章。

本书具有以下几个特点:(1)从系统开发的角度进行讨论。我们知道,一个管理信息系统通常由系统封面、菜单、数据表基本操作(记录输入、修改、删除和恢复)、数据查询和报表等几部分组成,本书直接从这几方面入手,展开介绍每部分的基本设计技术。(2)从使用者的角度进行介绍。本书由浅入深,先给出一些基本设计方法,再一步一步地添加新的功能,使读者随着内容的深入进一步提高编程水平。(3)从设计者的角度进行阐述。本书实际上是精选的实例汇粹,相当一部分是作者由大型软件开发中的一些技巧性强的应用程序改编而来的。书中所有实例都在中文 Visual FoxPro6.0 环境下调试通过,并给出了详细的代码,读者可以改进应用到自己的应用程序中。

本书条理清晰,讲解透彻,实例丰富,可操作性和实用性强。本书不但是从事 Visual FoxPro 6.0 的规划、开发与应用的广大从业人员的指导书和范例样板;同时也可作为广大大专院校相关专业师生的教学自学用书,以及科研机构参考用书和馆藏读物。尤其对急需使用 Visual FoxPro 进行数据库应用程序设计的开发人员,本书的实例具有重要的实际参考价值。

本书配套光盘内容包括:与本书配套的电子书。

系 列 书: 21 世纪计算机编程指南系列
书 名: 中文版 Visual FoxPro 6.0 高级实例开发指南
总 策 划: 北京希望电脑公司
文 本 著 者: 李春葆
责 任 编 辑: 龙启铭
CD 制 作 者: 希望多媒体创作中心
CD 测 试 者: 希望多媒体测试部
出版、发行者: 北京希望电子出版社
地 址: 北京海淀路 82 号, 100080
网址: www.bhp.com.cn E-mail: lwm@hope.com.cn
电话: 010-62562329,62541992,62637101,62637102
010-62633308,62633309 (图书发行,技术支持)
010-62613322-215 (门市) 010-62531267 (编辑部)

经 销: 各地新华书店、软件连锁店
排 版: 希望图书输出中心
CD 生 产 者: 北京中新联光盘有限公司
文 本 印 刷 者: 北京媛明印刷厂
开本 / 规格: 787×1092 毫米 16 开本 17.625 印张 407 千字
版次 / 印次: 2000 年 6 月第 1 版 2001 年 4 月第 2 次印刷
印 数: 5000~8000 册
本 版 号: ISBN7-900044-45-0/TP·45
定 价: 35.00 元 (ICD, 含配套书)

说明: 凡我社光盘配套图书若有缺页、倒页、脱页、自然破损,本社发行部负责调换

前 言

Visual FoxPro 6.0 是 Microsoft 公司最新推出的功能强大的关系数据库管理系统,其突出的优点是:

- 支持完全可视化的面向对象的程序设计;
- 强大的数据库 P 管理功能;
- 强大的数据查询功能;
- 支持 ActiveX 技术;
- 支持客户机/服务器体系结构;
- 新增支持 GIF 和 JPEG 图像格式;
- 新增 OLE 拖放。

由于诸多的优点,使 Visual FoxPro 6.0 成为专业级软件开发工具。学会使用该系统进行软件开发成为众多计算机爱好者的目标之一。

本书是作者在长期实际软件开发的基础上编写的,力求让了解 Visual FoxPro 基本编程的读者快速升级到专业级开发水平,本书具有以下特点:

- 从系统开发的角度进行讨论。我们知道,一个管理信息系统通常由系统封面、菜单、数据表基本操作(记录输入、修改、删除和恢复)、数据查询和报表等几部分组成,本书直接从这几方面入手,展开介绍每部分的基本设计技术。
- 从使用者的角度进行介绍。本书由浅入深,先给出一些基本设计方法,再一步一步地添加新的功能,使读者随着内容的深入进一步提高编程水平。
- 从设计者的角度进行阐述。本书实际上是精选的实例汇粹,相当一部分是作者大型软件开发中的一些技巧性强的应用程序改编而来的。书中所有实例都在中文 Visual FoxPro6.0 环境下调试通过,并给出了详细的代码,读者可以进一步改进。

为了简便,书中的 Visual FoxPro 除特殊指定外都是指 Visual FoxPro 6.0 中文版系统。

由于作者水平所限,可能存在不足之处,敬请读者指正,并一起交流提高,作者的 E-mail 地址为:licb@public.wh.hb.cn。

作者
2000 年 4 月



目 录

第1章 面向对象的程序设计 1	
1.1 对象和类..... 1	
1.2 Visual FoxPro 类层次..... 5	
1.3 类的设计..... 6	
1.4 对象的操作..... 13	
1.5 通过编程定义类..... 16	
1.6 Visual FoxPro 事件..... 25	
1.7 表单的使用..... 28	
1.8 一个面向对象的程序例子..... 37	
第2章 基本控件设计 41	
2.1 标签..... 41	
2.2 文本框..... 43	
2.3 编辑框..... 45	
2.4 命令钮..... 47	
2.5 命令组..... 50	
2.6 选项组..... 53	
2.7 复选框..... 56	
2.8 组合框..... 58	
2.9 列表框..... 61	
2.10 微调..... 65	
2.11 表格..... 67	
2.12 图像..... 71	
2.13 计时器..... 73	
2.14 页框..... 75	
2.15 OLE 绑定型控件..... 78	
2.16 线条..... 80	
2.17 形状..... 83	
第3章 系统封面设计 86	
3.1 静态封面设计..... 86	
3.2 动态封面设计..... 99	
第4章 数据表操作设计 122	
4.1 数据表基本操作设计..... 122	
4.2 数据表查询设计..... 152	
第5章 报表设计 200	
5.1 基本报表设计..... 200	
5.2 复杂报表设计..... 203	
第6章 菜单设计 214	
6.1 基本菜单设计..... 214	
6.2 用表单设计菜单..... 216	
第7章 应用实例设计 225	
7.1 数据表设计..... 225	
7.2 表单设计..... 225	
7.3 系统运行..... 271	

第 1 章 面向对象的程序设计

从 Visual FoxPro 3.0 开始就引入了面向对象的程序设计, 至 Visual FoxPro 6.0, 面向对象的程序设计技术日臻完善。在面向对象的程序设计中, 程序员把精力放在“对象”的设计上, 即如何构造对象是程序设计的重点, 而不像以往采用结构化程序设计那样重点如何考虑编写程序代码。在结构化程序设计中的分析和设计阶段, 要以“功能”划分整个系统, 分别构成各类功能模块, 各个功能模块如有可能还必须划分为更为具体的小功能模块, 直至功能模块能以一个函数或过程来实现为止。这种方法称为自下而上的程序设计方法, 它具有思路清楚、功能单一等特点, 但不便维护和修改。与结构化程序设计方法不同的是, 面向对象的程序设计方法采用自上而下的程序设计方法。它一般要经历“具体-抽象-具体”三个阶段。它以“对象”为处理中心, 在系统分析和设计时必须首先确定所要处理的对象, 然后确定对象的属性以及对象之间的相互关系。

1.1 对象和类

1.1.1 对象

对象是面向对象程序设计的基本单元, 是一种将数据和操作过程结合在一起的数据结构。在 Visual FoxPro 中, 对象有自己的属性、事件和方法。

属性

属性用于描述对象所具有的性质和特点。属性是对象本身的物理特性, 它规定对象的形状、位置和显示方式等诸多因素。比如, 命令按钮对象的大小、颜色和标题等都是命令按钮对象的属性。对象中的每个属性都具有一定的含义, 可以赋予一定的值。例如, Visual FoxPro 中命令按钮对象就具有以下基本属性:

- Caption 设置对象标题中显示的文本。
- Enabled 设置对象有效或无效。T. (缺省) 为有效; F. 为无效。
- FontName 设置对象的显示字体。
- FontSize 设置对象显示字体的大小。
- Height 设置对象的显示高度。
- Left 设置对象显示时距父对象左边的距离。
- Name 设置对象的名称。
- Picture 设置对象中显示的位图文件、图标文件或通用字段。
- SpecialEffect 设置对象的显示样式。0 三维; 1 (缺省) 为平面。
- Top 设置对象显示时距父对象上方的距离。
- Visible 设置对象是否可见。取值为 T. (缺省) 或 F. 。

- Width 设置对象显示时的宽度。

不同的对象有着许多共同的属性，但只要通过对象名称来调用某一属性，就能够唯一确定属性的使用范围。对象的属性可以在设计时在属性窗口中设置，也可以在运行时通过编程方式来设置。

事件

事件是能被对象识别和响应的动作。它是一种预先定义好的特定动作。每一个对象都有与之相关联的事件。比如，对于电话这个对象，当拿起电话听筒时将触发一个事件，开始拨号时也将触发一个事件。在 Visual FoxPro 中，用户单击鼠标、移动鼠标、按下键盘上的某个键时都会触发事件，初始化一个对象或者遇到导致错误的代码时也会启动事件。单击鼠标左键时，将会产生一个 Click 事件，而单击鼠标右键时，将会产生一个 RightClick 事件，移动鼠标时会产生一个 MouseMove 事件，等等。多数情况下，事件是由用户操作而引起的。

在 Visual FoxPro 中，对象的事件可由用户的动作产生，如 Click 事件；也可以由系统自动产生，如 Timer 事件。但要注意的是，尽管对象可以响应的事件的范围很广，但对象的事件集是由系统预定义的，用户不能扩充。例如，Visual FoxPro 提供了几个常用事件如下：

- Init 对象创建时触发。
- Click 用鼠标左键单击对象时触发。
- Dblclick 用鼠标左键双击对象时触发。
- RightClick 用鼠标右键单击对象时触发。
- KeyPress 按下并松开键盘上的一个键时触发。

方法

使用对象时若响应了某一个事件，总希望能够执行指定的任务，这就必须通过“方法”来实现。对象能够执行的一个操作称为方法。方法常常与事件有着紧密的联系，一个事件必定有一个与之相对应的方法。用户的动作能够激活事件，但响应事件必须通过对应的方法来实现。比如，为了响应 Click 事件，用户可以为 Click 事件编写相对应的方法程序。

对象、事件和方法之间的关系

在一个对象中，属性用于描述一个对象，说明对象的特征以及对象所要完成的功能，然而却没有说明对象应当如何去做。因此，对象还有与其相关的过程，这些过程就称为方法。方法不同 Visual FoxPro 中的过程，方法总是局限于一个对象内，不同于 Visual FoxPro 普通过程的调用。

事件可以有与之相关的方法，可以为某个事件编写一段方法代码，事件发生时就动这段代码执行。例如，可以为 Visual FoxPro 中的 Click 事件编写一段方法代码，当用户单击鼠标触发 Click 事件时，就执行为它编写的这段代码 Click 方法。此外，Visual FoxPro 中的方法可以独立于任何事件存在而不和任何事件相关联。因此，这些方法的调用执行必须在代码中按显式方式用，Visual FoxPro 中的事件基本上是固定的，不能创建新的事件，而方法则不同，可以无限扩充。

对象、事件和方法之间的关系如图 1.1 所示。

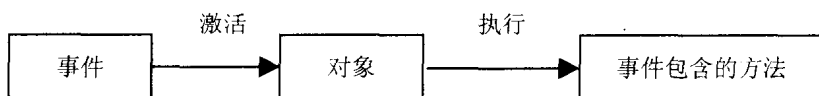


图 1.1 对象、事件和方法之间的关系

例如，有一个命令钮对象具有以下属性：

```

Name="Command1"    &&指定名称
Caption="上一记录" &&指定标题
Top=10             &&指定左上角位置
Left=20            &&指定左上角位置
Height=100         &&指定高度
Width=200          &&指定宽度
  
```

该命令钮对象有一个 Click 事件，该事件包含的方法如下：

```

SKIP -1            &&上移记录指针
IF BOF()           &&如果到了跳过了第一个记录，则指向第一个记录
    GO TOP
ENDIF

THISFORM.REFRESH  &&刷新表单数据
  
```

当用户单击该对象时，激活该对象的 Click 事件，然后执行 Click 事件包含的方法，从而实现相应的功能即将当前记录指针上移。这样通过 Click 包含的方法完成“上一记录”的操作。

从该例看出，对象是程序设计的主体，通过属性指定对象的“状态”。当对象的“状态”指定后，便确定该对象的操作，也就是哪些事件激活该对象，最后编写实现这些事件的方法程序。

1.1.1.2 类

类和对象密切相关，类是创建对象的关键。所有对象的属性、事件和方法都是在类定义中确定的。类就像是一个蓝图或模板，对象都是由它生成的。类确定了由它生成的对象所具有的属性、事件和方法。反过来，类的功能只有通过产生一个对象并且通过引用该对象才能实现。在 Visual FoxPro 中，通过使用对象的属性、事件和方法来引用一个对象。因而面向对象程序设计方法的核心部分是类的设计和处理。

子类

开发一个类并不纯粹是为了引用，而是要在此基础上派生出更多的子类。每个子类可以具有已存在类的全部属性和方法，并且可以任意添加新的控件、属性和方法。这样做的结果一方面可以提高程序代码的重用性，减轻工作量；另一方面亦可满足各个不同层次的需求。

类的封装性

对象可以是真实世界中的具体事物，而从对象中归纳抽象出的类自然具备许多详细的内外部特征，具有许多的行为和功效。就以电话为例，安装一部电话时，并不会关心这部电话呼号和拨号时的内部机制，也不会关心电话是如何与话路相连的，更不用关心按键如何转换

成电信号的。相反，我们只需要知道拿起听筒、拨出电话号码、与接电话的人谈话即可，完全没有必要了解电话本身的内部技术细节。但并不是说，电话没有了这些内部细节，而是将这些细节作了处理，将它们隐藏在电话的内部。同样，对于一个对象内的属性和方法，也可以进行抽象处理，将它们封闭在一个对象的内部，使得当引用一个对象或者创建一个新的对象时，该对象就具有了一定的属性和方法，这就是对象的封装。

在使用 Visual FoxPro 提供的基类时，不必关心这些基类的内部是用什么代码和方法实现其特有的功能，所要做的工作就只要声明一下，就可直接调用其所固有的功能。封装的特性使得代码的修改和维护变得比以前容易多了，因为对类的某一个方法或属性进行修改时，只会在该类内部起作用，不会影响其他类的正常操作。

类的继承性

为了方便，Visual FoxPro 中提供有许多基类。由于很容易就可以从一个类中派生出另一个子类，这个子类不但具有父类的属性和方法，而且可以根据需要修改这些属性和方法，或添加新的属性和方法。

通过继承父类的属性和方法，如果某个类中发现问题时，就不需要逐个修改其子类，而是将这个类本身作适当修改后，这个变化就能够自动反映到所有子类中。

Visual FoxPro 基类

在 Visual FoxPro 中，可以从 Visual FoxPro 基类中创建对象或者派生出子类。表 1.1 列出的是 Visual FoxPro 所提供的基类名。

表 1.1 Visual FoxPro 的基类

ActiveDoc	ProjectHook
CheckBox	Label
Column*	Line
ComboBox	ListBox
CommandButton	OLEControl
CommandGroup	OLEBoundControl
Container	OptionButton *
Control	OptionGroup
Cursor	Page*
Custom	PageFrame
DataEnvironment	Relation
EditBox	Separator *
Form	Shape
FormSet	Spinner
Grid	TextBox
Header*	Timer
Hyperlink	ToolBar
Image	

*表示该类为容器类中的必需部分，它不能派生子类

所有的 Visual FoxPro 基类都具有以下的最小事件集：

- Init 创建一个对象时发生。
- Destroy 从内存中释放一个对象时发生。

- Error 无论何时，只需类的事件或者方法代码中发生错误时发生。
- 此外，所有的 Visual FoxPro 基类不具有以下的最小属性集：
- Class 用于表示类的类型。
 - BaseClass 表示派生该类的基类名。
 - ClassLibrary 存储该类的类库。
 - ParentClass 表示派生出当前类的类，如果类直接从 Visual FoxPro 基类派生，则 ParentClass 属性与 BaseClass 属性相同。

注意 如果对象是从一个基类上派生的，那么该对象的 ParentClass 属性就是空的。

1.2 Visual FoxPro 类层次

Visual FoxPro 中的类分为两种类型，即容器类和控件类，下面分别进行讨论，图 1.2 给出 Visual FoxPro 中的类的层次示意图。

Visual FoxPro 对象	
控件类	容器类
CheckBox(复选框)	Container(容器)
ComboBox(组合框)	FormSet(表单集)
CommandButton(命令按钮)	Form(表单)
Control(控件)	Grid(表格)
Custom(自定义)	Column(列)
EditBox(编辑框)	PageFrame(页框)
Header(标头)	Page(页)
Image(图像)	ToolBar(工具栏)
Label(标签)	OptionButtonGroup(选项按钮组)
Line(线条)	CommandButtonGroup(命令按钮组)
ListBox(列表框)	
OLE Control(OLE 控件)	
Shape(形状)	
Spinner(微调)	
TextBox(文本框)	
Timer(计时器)	

图 1.2 Visual FoxPro 中的类的层次

1.2.1 容器类

容器类可以包含其他对象，并允许访问所包含的对象。例如，如果创建一个包含有两个对表框和两个命令按钮的容器类，然后将一个基于该类的对象添加到表单中，则在运行和设计时每个单独的对象都是可操作的，而且可以改变列表框的位置或者命令按钮的标题。因此，容器类提供了一种将多个对象进行组合的功能。表 1.2 列出了 Visual FoxPro 中的每个容器类所能包含的对象。

表 1.2 Visual FoxPro 中的容器类所能包含的对象

容器类	可以包含的对象
CommandButtonGroup	命令按钮
Container	任意控件
Control	任意控件
Custom	任意控件、页框、容器、自定义
FormSet	表单、工具栏
Form	任意控件、页框、容器、自定义
Grid 中的 Column	标头和除表单、表单集、工具栏、计时器、其他表格列外的任意对象
Grid	表格列
OptionButtonGroup	选项按钮
PageFrame	页
Page	任意控件、容器、自定义
ToolBar	任意控件、页框、容器

1.2.2 控件类

控件类比容器类封装得更完全。对于由控件类创建的对象，在设计和运行时是作为一个单元对待的，构成控件对象的各部分不能单独修改或者操作。所有控件类都没有 AddObject 方法，即不能向控件对象中添加其他对象。

Visual FoxPro 中的对象根据它们所基于的类的性质可分为两类：

- 容器可作为其他对象的父对象。如当表单作为容器时，它是放在其上面的复选框的父对象。
- 控件可包含在容器中，但不能作为其他对象的父对象。如复选框。

1.3 类的设计

类的设计是面向对象编程中极为重要的内容。当将大量的属性、方法、事件等封装到一个类中之后，所要做的工作就变成了如何在这个类的基础上派生出一个对象，并将这个对象加入到应用的程序中，然后再确定如何运行这个应用程序。

1.3.1 类设计的考虑

类的定义要面临许多不同的应用背景，在类的设计当中，首要的任务是确定用什么样的类来完成一项具体的工作。在应用程序中，经常会碰到具有部分相同功能的对象，换句话说，这些对象都具有一定的共性，这时就可以针对这个共性来设计一个类。例如，可以设计命令按钮来移动表中的记录指针、关闭表单、获取帮助等。所有这些按钮都可以作为一个类加以存储并且可以随时添加到表单中。

确定要设计的类后，接着可以继续设计类中所包含的属性和方法，以及对 Windows 标准事件的响应。在 Visual FoxPro 中，可以在基类之上创建几种子类，可以在 CREATE CLASS 命令的 AS 子句或者“新类”对话框中指定要创建的类。

1.3.2 类库文件

使用类设计器创建一个类时，必须指定类所在的类库。每个新建的类都要存入一个扩展名为.VCX的类库中，为了创建一个类库，可以使用如下的三种方法：

- 在类设计器中设计一个类时，在“新类”对话框的“存储于”框中指定一个新的类库文件名。
- 使用 CREATE CLASS 命令的 OF 子句来指定新的类库名。例如：

```
CREATE CLASS myclass OF new_lib AS CUSTOM
```

- 使用 CREATE CLASSLIB 命令。其基本格式为：

```
CREATE CLASSLIB <新的类库>
```

例如，以下命令建立一个类库文件 USERLIB.VCX：

```
CREATE CLASSLIB USERLIB
```

定义类库文件后，就可以很容易地实现类库之间类的拷贝和删除。要将一个类从一个类库文件拷贝到另一个类库中，步骤如下：

- (1) 确保这两个类库都在一个项目中。
- (2) 在项目管理器中选择“类”选项卡。
- (3) 单击要拷贝的类所在类库中左边的加号。
- (4) 将选中的类从原来的类库中拖到新的类库中即可。

通过类拷贝，可以把一个类及其所有的子类都保存在同一个类库文件中。这样，当我们要用到该类及其子类时，只要打开一个类库文件即可。否则，如果一个类包含有来自不同类库文件中的元素，则必须同时打开多个类库文件。

装入到类库文件中的类，还可以从中移去。Visual FoxPro 提供有两种移去类的方法。

一是在项目管理器中选择要删除的类，然后再单击“移去”按钮；另一种是使用 REMOVE CLASS 命令。此外，还可以用 RENAME CLASS 命令来改变类库文件中的类名，但是必须记住，改变类库文件中的类名后，在其他.VCX 类库文件中引用某个类名的类和子类时，如果不作相应的修改，就会发生错误。

1.3.3 创建新类

可以在类设计器中创建自己的新类并显示设计的每个对象。为了定义一个新类，即弹出“新类”对话框后完成类的定义，可以有以下四种方法：

- 在项目管理器中选择“类”选项卡，然后单击“新建”按钮。
- 从“文件”菜单选择“新建”命令，然后选择“类”，再单击“新建文件”按钮。
- 使用 CREATE CLASS 命令。
- 将表单或表单上的部分对象以类存储。

我们以第四种方法讨论创建新类的过程。建立一个如图 1.3 所示的表单，其中只有一个命令按钮 COMMAND1，其上建立一个 MOUSEMOVE 过程，该过程包含的代码如下：

```
THIS.MOUSEPOINTER=99
```

```
THIS.MOUSEICON="POINT.CUR"
```

这个过程的功能是当鼠标指针移动到该命令按钮之上时，鼠标指针变成一个手指形状。

设成一个类后，一个应用中所有命令钮都从该类派生，从而节省代码。

当设计好该表单后，选择其中的命令钮(因为只将命令钮设计成类)，然后选择主菜单中的“文件”，再选择“另存为类”，这时出现如图 1.4 所示的“另存为类”对话框，在“类名”中输入 USERCOMM，在“文件”中选择类库文件 USERLIB.VCX，单击“确定”即可。

现在查看 USERLIB 类库中的该类，选择主菜单中的“工具”，再选择“类浏览器”，打开类库文件 USERLIB，如图 1.5 所示，看到该类库包含有 USERCOMM 类。单击该类，出现该类的“类设计器”，用户可以进行修改操作。

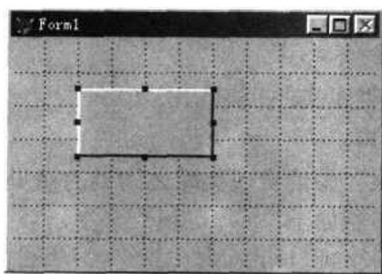


图 1.3 一个表单

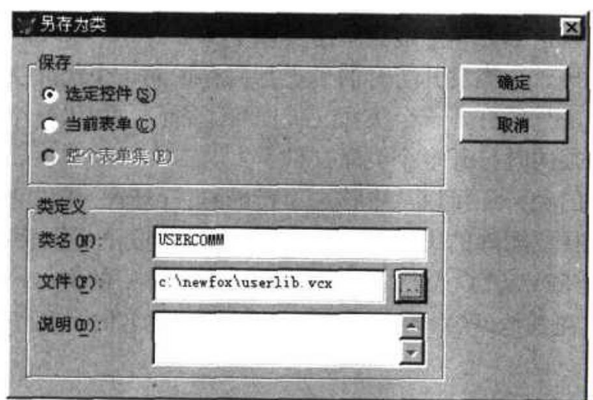


图 1.4 “另存为类”对话框

现在再看如何从类派生新的对象。使用 CREATE FORM FORM1 创建一个新表单 FORM1，选择“表单控件工具栏”中的“查看类”，然后单击“添加”，在出现的“打开”对话框中选择类库 USERLIB.VCX，这时出现新的“表单控件工具栏”，如图 1.7 所示。单击其中的命令钮，将鼠标指针放到表单 FORM1 上，便产生一个命令钮，打开其“属性”窗口，我们看到该对象是从 USERCOMM 类派生，因此，它继承了 USERCOMM 类的方法和属性，也就是说，当鼠标指针移动到该命令钮之上时，鼠标指针变成一个手指形状。

在“类浏览器”中选择 USERCOMM 类后，单击“查看类代码”显示该类的定义代码，如图 1.9 所示，用户可以使用主菜单的“文件”菜单对其进行存储操作。从该类代码可以清楚地看到该类的定义过程和派生关系。

我们还可以向这个类中添加属性方法。

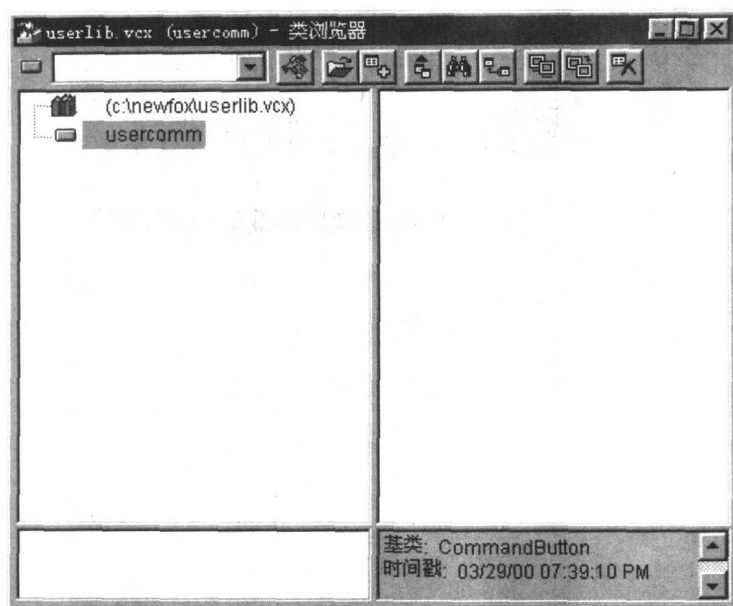


图 1.5 “类浏览器”窗口

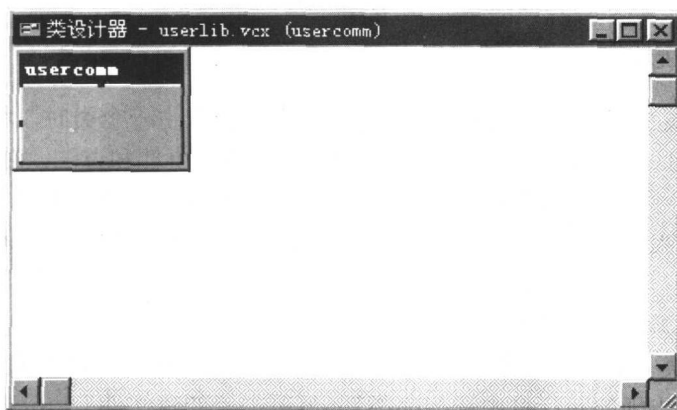


图 1.6 “类设计器”窗口



图 1.7 新的“表单控件工具栏”



图 1.8 派生的命令钮的“属性”窗口

1.3.4 添加类属性与方法

属性和方法是属于某个类的，不能独立于类而存在。派生一个类时，将继承基类或者父类的属性和方法。可以从“属性”窗口选择要继承的属性、事件和方法。对于属性，选中时，如果是可以修改的，则可以在属性窗口中输入属性值。例如，对于图 1.8 中从 USERCOMM 类派生的对象 USERCOMM1，可以将其 Caption 属性设置为“返回”。然后在其 Click 事件中添加的如下代码：

```
THISFORM.RELEASE
```

下面新建一个类，并讨论在其中添加属性和方法。从主菜单选择“文件”，从下拉菜单中选择“新建”，在出现的对话框中选择“类”，再单击“新建文件”，出现类似于图 1.4 所示的对话框。在“类名”文本框中输入“FORM1”，从“派生于”组合框选择“Form”，在“存储于”中选择 USERLIB.VCX 类库文件，单击“确定”，进入该类的设计屏幕，其中的操作与表单设计基本相似。选择主菜单项“类”，如图 1.10 所示。选择“新建属性”菜单项出现如图 1.11 所示的“新建属性”对话框，其中“名称”文本框键入属性名称，“说明”文本框中键入这个属性的有关描述，“可视性”框中指定属性是否受保护、是否隐藏。现在，在“名称”文本框中输入 JLS，单击“添加”，便为该新建了一个属性 JLS。采用同样的方法，新建一个 DELJLS 属性。

新属性创建后，缺省值设置为假(.F.)。为了给新属性设置不同的缺省值，可以使用“属性窗口”来完成。在“属性窗口”中选择“其他”选项卡，然后单击新属性再设置其缺省值。

完成类中的属性定义后，还可以将新的方法添加到类中。为此，从类设计器的“类”菜单中选择“新建方法”命令，在“新建方法”对话框中键入方法名和描述信息后，再指定方

法是受保护、是否隐藏。添加方法后，还必须在代码编辑窗口中添加方法的代码(与表单中的对象设计方法完全相同)。这里建立一个新方法 DISP。



图 1.9 生成的类代码

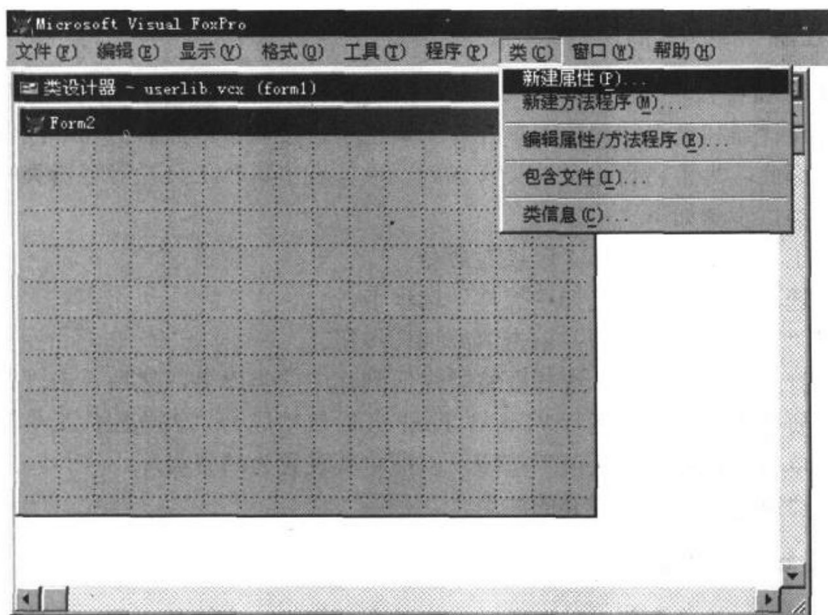


图 1.10 创建类的屏幕

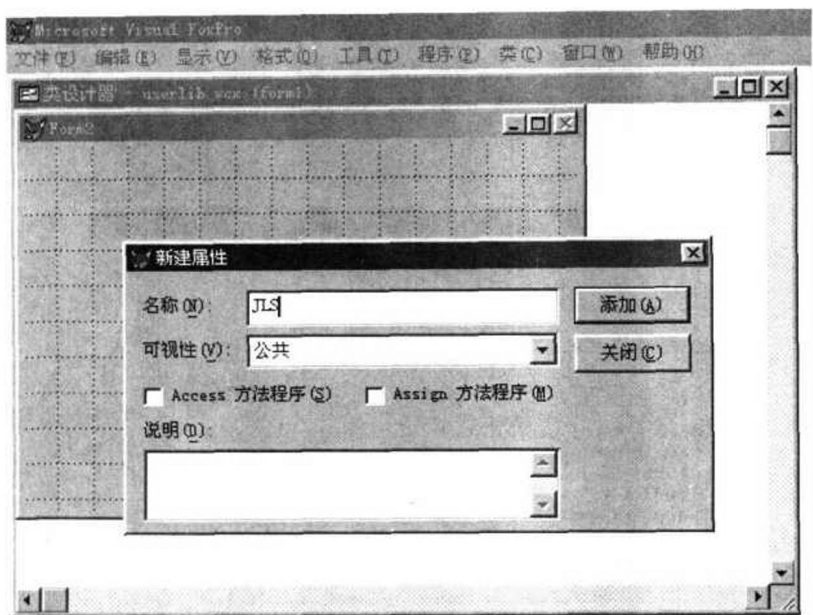


图 1.11 类的“新建属性”对话框

注意，类定义中属性和方法缺省为公共的(public)，表示可以在其他类或者过程中进行引用。如果属性和方法设置为受保护的(protected)，则仅可以被类定义中的其他方法或者类的子类访问。如果属性和方法设置为隐藏的(hidden)，则只能被类定义的其他成员访问，类的子类不能访问或者引用被隐藏的成员。

1.3.5 类的注册

可以把类从项目管理器中拖动到表单设计器或者类设计器中，然后进行类的注册，以便可以在表单设计器和类设计器的表单控件工具栏中直接显示，然后像标准控件那样添加到容器对象中。因此，为了将自定义的类添加到表单、表单集、工具栏等容器类中，首先必须注册新定义的类，步骤如下：

- (1) 从“工具”菜单中选择“选项”命令。
- (2) 在“选项”对话框中选择“控件”选项卡。
- (3) 选择“可视类库”选项并单击“添加”按钮。
- (4) 在“打开”对话框中，选择要将类注册到其中的类库名，然后单击“打开”按钮。
- (5) 如果要使类库在以后运行 Visual FoxPro 时自动成为“表单控件工具栏”中的可用类库，还必须单击“设置为默认值”按钮，以便将其设置为缺省类库。
- (6) 单击“确定”按钮完成类的注册。

1.3.6 调用父类中的方法

创建一个类时，这个类将自动继承父类中的全部属性、方法和事件。在面向对象的程序设计中，一个类既可以是 Visual FoxPro 基类的派生类，也可以用作定义其他类的基类。类的这种关系就是类的继承关系。创建的子类不但可以继承父类的全部属性、方法和事件，