

第 1 章

窗体与控件

- 对象概述
- 认识 Visual Basic 窗体
- 窗体的属性
- 窗体的方法
- 窗体的事件
- 窗体与控件

1.1 对象概述

1.1.1 对象的概念

对象是基于对现实世界的描述，对象是程序设计的基本实体。在 Visual Basic（简称 VB）中，对象分为系统预定义对象和用户自定义的对象两类。窗体和控件共同构成 Visual Basic 中预定义的对象。工具箱中的控件是“空对象”，当这些控件被建立在窗体上后才成为真正的对象，即实体，如图 1.1 所示。



图 1.1 窗体中的对象

在面向对象的程序设计中，把现实世界中的所有事物都看作对象，并且每种事物或物体都可能属于同一个类，也可能属于不同的类，如交通工具是一种类型，而交通工具又可分为汽车、火车、飞机和轮船等不同的类型。

用户对对象的操作用事件来表示。建立一个对象后，其操作可以通过对该对象的属性、事件及方法来描述。

1.1.2 “编辑器”设置

利用“选项”对话框中的“编辑器”选项卡，用户可设置代码编辑器的特性。选择菜单栏中的“工具”→“选项”命令，就会弹出“选项”对话框，如图 1.2 所示。

在“编辑器”选项卡中设置编辑功能，其中：

- “自动语法检测”用于设置在输入一行代码时是否自动进行语法检查；
- “要求变量声明”用于设置在模块中是否要求显式声明变量；
- “自动列出成员”用于设置是否自动列出成员；
- “自动显示快速信息”用于设置是否显示有关函数及其变量的说明；
- “自动显示数据提示”用于在中断模式下是否在代码窗口中的光标位置显示变量或对象属性值。

设置了“自动列出成员”之后，在编写代码时，输入“对象名。”系统会自动显示对象的属性、方法、事件，如在代码编辑器中输入文本框名 Text1.时，系统就自动显示如图 1.3 所示的属性界面。

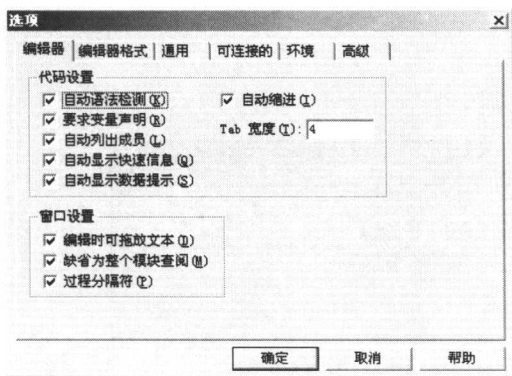


图 1.2 设置“编辑器”特性

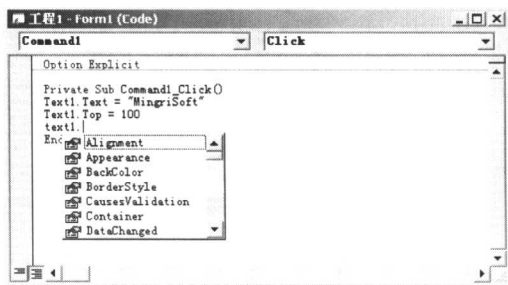


图 1.3 设置属性

1.1.3 对象的属性

属性是指一个对象的性质和特性，不同的对象拥有不同的属性。在 Visual Basic 中，常见的属性有标题 (Caption)、名称 (Name)、颜色 (Color)、字体大小 (FontSize)、是否可见 (Visible) 等，这些属性决定了对象在界面中具有的外观及功能。不同的对象有不同的属性，在程序设计时，需要合理、正确地设置对象的属性值。

在 Visual Basic 中设置对象的属性可以通过属性窗口来完成，也可以在代码窗口中通过输入语句来进行设置。

语法

对象名.属性名称=新设置的属性值

示例

将窗体 frm_main 的 Caption 属性值设置为“采购管理系统”，其对应的语句为：

```
frm_main.Caption = "采购管理系统"
```

1.1.4 设置对象的属性

对象属性可以在代码窗口的程序代码中设置，也可以在程序设计阶段通过属性窗口设置。若想在属性窗口中设置，首先要激活该窗口，然后单击属性窗口或按 F4 键。

在属性窗口中设置属性常用的方法有以下 3 种。

(1) 直接键入新属性值。例如，设置 frm_xgmm 窗体的 Caption 属性为“操作员密码修改”，如图 1.4 所示，设置后的结果如图 1.5 所示。



图 1.4 在 Caption 属性中直接键入属性值

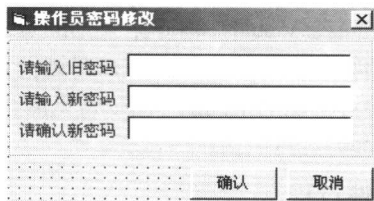


图 1.5 设置后的结果 1

(2) 选择输入选项，通过下拉列表选择所需要的属性值。例如，设置 frm_xgmm 窗体的

BorderStyle 属性为 3-Fixed Dialog, 如图 1.6 所示, 设置后的结果如图 1.7 所示。

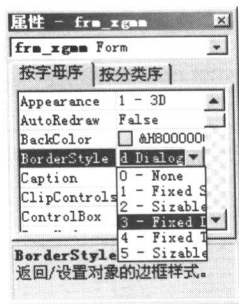


图 1.6 选择窗体的属性值图

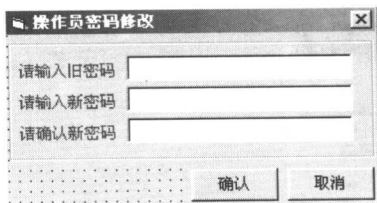


图 1.7 设置后的结果 2

(3) 利用对话框设置属性值。例如, 设置 Label1 控件的 Font 属性为“宋体”, 单击“...”按钮 (如图 1.8 所示), 在弹出的“字体”对话框中设置字体、大小等 (如图 1.9 所示), 然后单击“确定”按钮, 设置后的结果如图 1.10 所示。



图 1.8 利用对话框设置窗体的属性值

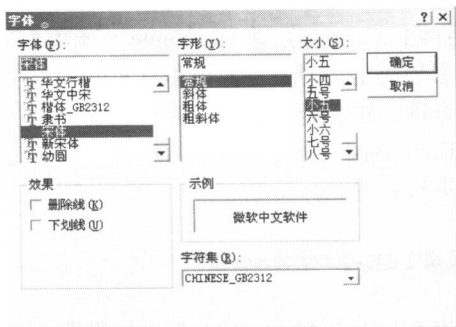


图 1.9 “字体”对话框

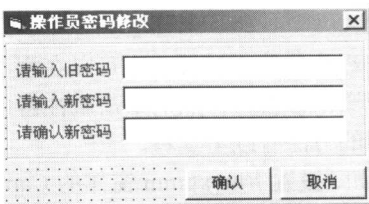


图 1.10 设置后的结果 3

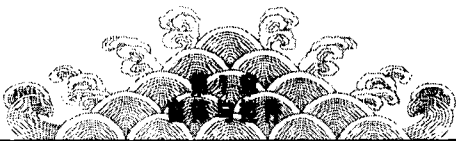
1.1.5 对象的事件

事件是 Visual Basic 预先编辑完成的并且能被对象识别的动作。传统的编程语言是面向过程的机制, 而在 Visual Basic 中采用的是事件驱动的机制, 用户不必考虑程序执行的每一个步骤, 只要编写一些单独的程序段, 在触发到该事件时便会被激活。

不同的对象识别不同的事件, 常见的事件有单击 (Click)、双击 (DbClick)、改变 (Change) 和装载 (Load) 等。

1.1.6 事件过程

当在对象上发生了事件后, 应用程序对这个事件所能做出的反应以及处理步骤就是事件过程。在 Visual Basic 中设计程序的主要工作就是为对象编写事件过程的程序代码。



语法

```
Private Sub 对象名称_事件名称()  
    '事件响应程序代码  
End Sub
```

对象名称：对象的 Name 属性。

事件名称：由 Visual Basic 预先定义好的赋给该对象的事件，并且该事件必须是对象所能识别的。

示例

在鼠标单击事件下实现锁定 Text1 控件，设置“删除”、“保存”等按钮不可用，“增加”按钮可用。其实现的代码如下：

```
Private Sub Cmd_cancel_Click()  
    Text1.Locked = True : Cmd_del.Enabled = False : Cmd_save.Enabled = False  
    Cmd_add.Enabled = True : Cmd_edit.Enabled = False  
End Sub
```

1.1.7 对象方法

方法是 Visual Basic 中的一个术语。所谓方法就是对象所具有的动作和行为，实际上是 Visual Basic 所提供的一些已经封装好的通用子程序。用户可以使用对象名直接调用其方法。

语法


对象名称.方法名称

示例

当用户按 Enter 键时，使用 SetFocus（获取焦点）方法，设置 Text2 获得焦点，代码如下：

```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)  
    If KeyCode = vbKeyReturn Then Text2.SetFocus '使用获取焦点方法使Text2控件获取焦点  
End Sub
```

注意：调用对象方法时，如果省略对象名，那么程序将把当前窗体作为对象。

Visual Basic 对同类对象定义了相同的属性、事件和方法。对象的属性、事件和方法可以通过“对象浏览器”来查找，“对象浏览器”窗口可以通过“视图”菜单中的“对象浏览器”命令，或在工具栏中单击按钮打开。

1.2 认识 Visual Basic 窗体

1.2.1 窗体的概念

窗体是应用程序的一个重要组成部分。在程序设计阶段，窗体是程序员的“工作台”，程序员在窗体上建立应用程序。在程序运行的时候，每一个窗体对应一个窗口。

窗体是 Visual Basic 中一个重要的对象，它可以作为其他控件的“父对象”，也就是说，窗体除了具有自己的属性、方法外，还可以作为其他控件的容器，可以在它上面放置除了窗体之外的其他控件，如文本框、图片框、各种按钮等。当窗体显示时，在它上面的控件是可见的；当窗体移动时它们随之移动；当窗体隐藏时，在它上面的控件也跟着隐藏。

窗体文件的扩展名是 .frm，可以作为文件存放到磁盘上。

1.2.2 窗体样式

VB 中窗体样式主要有单文档窗体（SDI）、多文档窗体（MDI）和资源管理器窗体。

1. SDI 窗体

SDI 窗体是单文档窗体，指在应用程序中每次只能打开一个文档，想要打开另一个文档时，必须先关上已打开的文档。不能将一个窗体包含在另一个窗体中，所有的窗体都可以在屏幕上自由移动。

本书中介绍的大多是 SDI 窗体。

2. MDI 窗体

MDI 窗体是多文档窗体，在应用程序中可以同时打开多个文档。每个文档都有自己的窗口，文档或子窗口被包含在父窗口中，父窗口为应用程序中所有的子窗口提供工作空间。当最小化父窗口时，所有的文档窗口也被最小化，只有父窗口的图标显示在任务栏中。Microsoft Word 和 Microsoft Excel 应用程序就是 MDI 界面。

创建 MDI 窗体的步骤如下。

(1) 在菜单栏中选择“工程”→“添加 MDI 窗体”命令，新添加的窗体的默认名为 MDIForm1 (一个应用程序只能有一个 MDI 窗体)。

(2) 创建一个新窗体 Form1 (或者打开一个存在的窗体)，然后把它的 MDIChild 属性设为 True，将启动对象设置为该窗体。

3. 资源管理器窗体

资源管理器界面越来越流行，资源管理器界面是指包括有两个窗格或者区域的一个单独的窗口。通常左半部分是一个树型的或者层次型的视图，右半部分是一个显示区，这样的界面可用于定位或者浏览大量的文档、图片或者文件。Windows 资源管理器就是这样的界面。

1.2.3 窗体的结构

窗体由标题栏、控制菜单、控制按钮区和窗体组成，如图 1.11 所示。

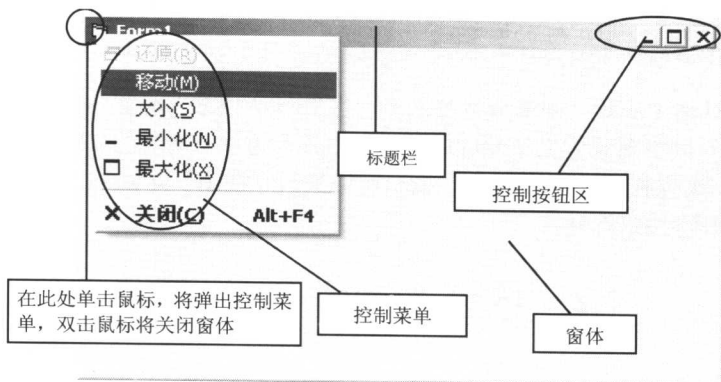


图 1.11 窗体的结构

下面介绍各部分的功能。

1. 标题栏

标题栏是指窗体顶部的长条区域，在标题栏中可以显示窗体的 Caption 属性值。

2. 控制按钮区

窗体的控制按钮在窗体标题栏的最右端，包括最大化、最小化和关闭按钮，其作用是对窗体进行控制。

3. 窗体

用户设计程序外观的操作界面，用户可以在该界面上放置各种控件。

4. 控制菜单

位于窗体的左上角，双击该图标将关闭窗口体，单击该图标，将以下拉方式显示系统命令菜单。

1.2.4 添加和删除窗体

要向工程中添加窗体有两种方法，一种是创建一个新窗体，另一种是打开现有的窗体。

1. 创建一个新窗体

(1) 从菜单栏中选择“工程”→“添加窗体”命令，打开“添加窗体”对话框。

(2) 在“添加窗体”对话框中，打开“新建”选项卡，选择“窗体”图标，然后单击“打开”按钮。

另外，在“新建”选项卡中，还提供了一些已经事先设计好的窗体，如图 1.12 所示，这些窗体添加到工程中即可使用。例如，在工程中添加一个关于窗体，方法为：选择“新建”选项卡中的“‘关于’对话框”选项，然后单击“打开”按钮，这时一个名为 frmAbout 的窗体将被增加到“工程资源管理器”列表中，同时显示在工程中，如图 1.13 所示。

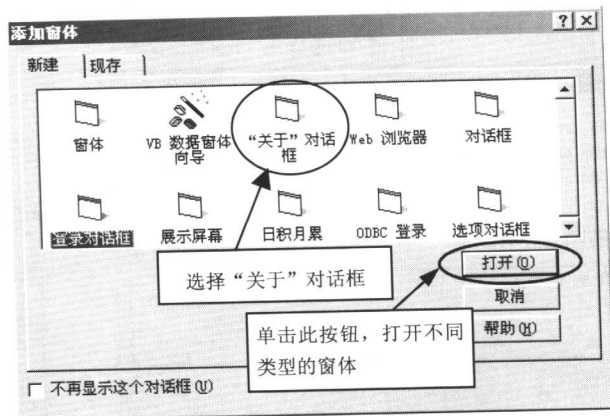


图 1.12 “添加窗体”对话框

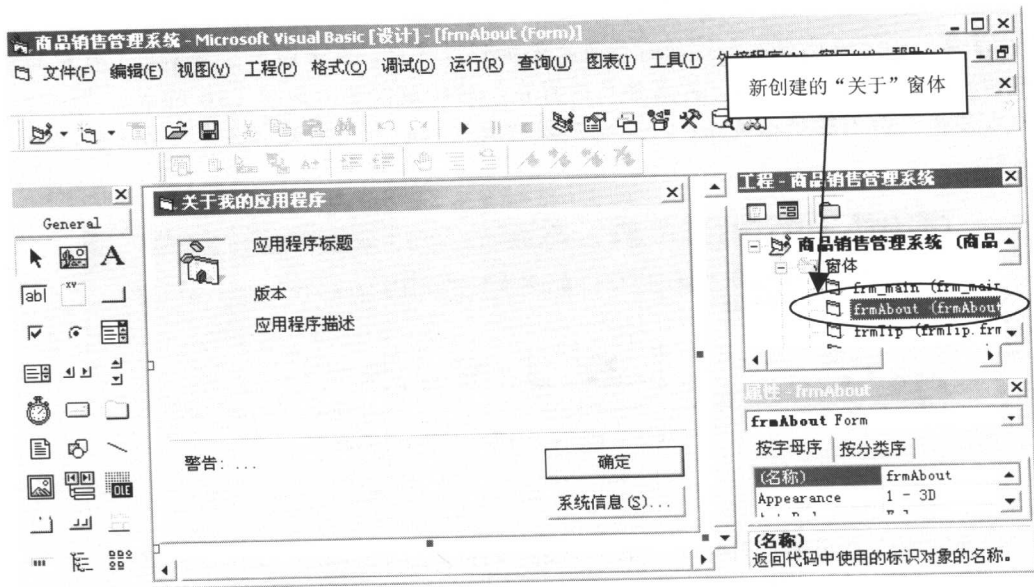


图 1.13 “关于”窗体

2. 从现存窗体中导入窗体

在开发应用程序的过程中，如果在其他工程中，有与本工程中要设计的窗体相同或相类似

的窗体，那么可以将该窗体导入到本工程当中，这样会比重新设计一个窗体要节省很多的时间，从而提高软件的开发速度。例如，在人口管理系统中，已经设计了一个“权限设置”窗体，在酒店管理系统中，也需要设计一个“权限设置”窗体，这种情况下，就可以从人口管理系统当中将“权限设置”窗体直接导入到酒店管理系统的工程中。具体方法如下。

(1) 在“酒店管理系统”工程中，单击“工程”→“添加窗体”菜单命令，打开“添加窗体”对话框。

(2) 在“添加窗体”对话框中，打开“现存”选项卡。

(3) 在“现存”选项卡中选择要导入的窗体文件，然后单击“打开”按钮，如图 1.14 所示，导入后的结果如图 1.15 所示。

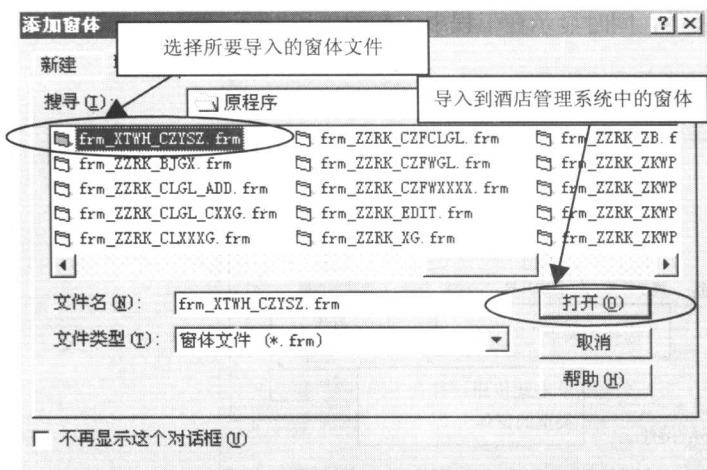


图 1.14 选择现存的窗体文件

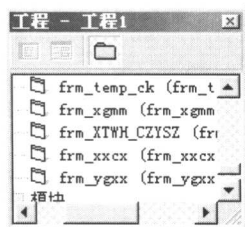


图 1.15 已导入到工程中的窗体

注意：如果将已经添加过的窗体再次添加到工程中，或者添加的窗体与工程中的窗体名称相同，那么系统会提示“此窗体已经存在”的提示信息，如图 1.16 所示。这时如果是因为名称相同而不能导入窗体，那么就应该将要导入的窗体重新命名，然后再重新导入到工程中。

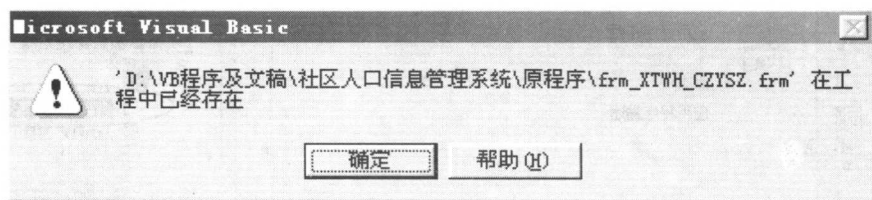


图 1.16 添加相同名称窗体后的提示信息

3. 移除或删除窗体

当工程中的某些窗体不需要时，可以将该窗体移除或彻底删除。操作步骤如下。

(1) 打开 VB 开发环境，在工程资源管理器中选中要移除的窗体。

(2) 选择“工程”→“移除”菜单命令，将窗体从 Visual Basic 6.0 开发环境中移除。另外，要将窗体文件彻底删除，可以在存放工程文件的文件夹中直接删除。

例如，将采购材料管理 (frm_cgclgl) 窗体从采购管理系统中移除，如图 1.17 所示。



图 1.17 移除窗体

1.3 窗体的属性

1.3.1 更换窗体的图标

在 Visual Basic 工程中每个窗体都有一个默认的图标，为了给应用工程赋予实际工程的意义，可以更换窗体的图标。更换图标可以使用窗体的 Icon 属性实现。

具体步骤如下。

- (1) 首先选中需要更换图标的窗体。
- (2) 在窗体的属性窗口列表中找到 Icon 属性，单击按钮 **...**，如图 1.18 所示。
- (3) 在弹出的“加载图标”对话框中选择图标文件，加载图标后的窗体如图 1.19 所示。



图 1.18 选择图标文件

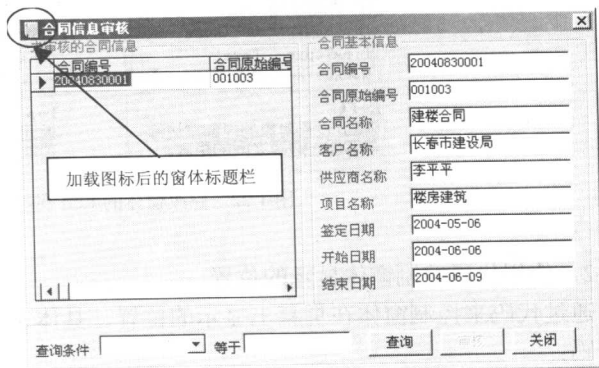


图 1.19 加载图标后的窗体

1.3.2 隐藏窗体的标题栏

每一个窗体中都有一个标题栏，但有些时候，为了使界面更美观，可以将标题栏去掉，如图 1.20 所示。具体方法如下：在窗体属性栏中，找到 BorderStyle 属性，将其值设置为 0-None，如图 1.21 所示。

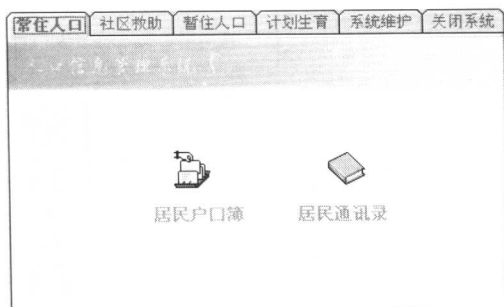


图 1.20 没有标题栏的窗体

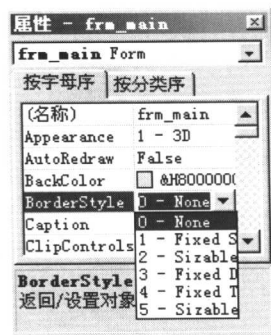


图 1.21 设置 BorderStyle 属性

1.3.3 控制窗体的显示位置

控制窗体在屏幕中的显示位置有两种方法，一种是在属性窗口中设置，另外一种是通过程序代码控制。

1. 在属性窗口中设置窗体显示的位置

在属性窗口中修改窗体的 Left 和 Top 属性值，可以控制窗体在屏幕上显示的位置，如图 1.22 所示，将窗体的 Left 属性值设为 3000，表明窗体在运行的时候距离屏幕左端的距离为 3000；将窗体的 Top 属性值设为 2600，表明窗体在运行的时候距离屏幕顶端的距离为 2600。

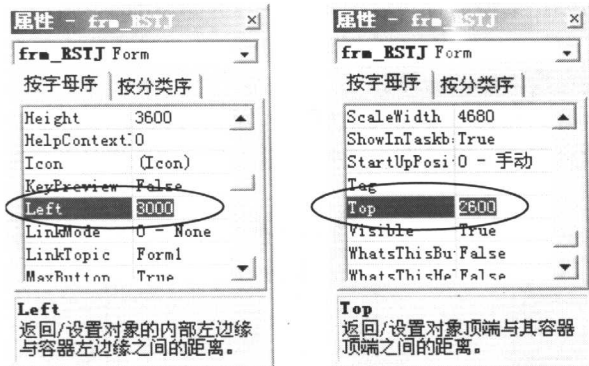


图 1.22 修改窗体的 Left 属性与 Top 属性

2. 使用代码控制窗体显示的位置

通过代码来控制窗体在屏幕上显示的位置，具体实现方法是在窗体的 Load 事件中添加如下代码：

```
frm_RSTJ.Left = 2000
frm_RSTJ.Top = 2600
```

若将窗体显示在屏幕的中间位置，需在窗体的 Load 事件中添加如下代码：

```
Me.Left = (Screen.Width - Me.Width) / 2
Me.Top = (Screen.Height - Me.Height) / 2
```

1.3.4 修改窗体的大小

一般情况下，在工程中可以通过拖动窗体的边缘改变窗体的大小，另外也可以在属性窗口中通过修改窗体的 Height 属性和 Width 属性来改变窗体的大小，如图 1.23 所示。

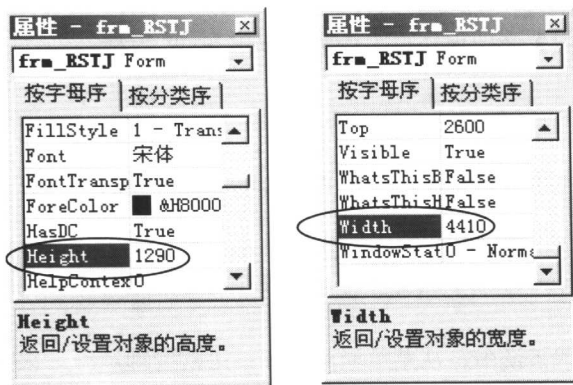


图 1.23 修改窗体的 Height 属性和 Width 属性

1.3.5 设置具有图像背景的窗体

在开发应用程序的时候，为了美化程序界面，常常需要给窗体添加一个图片作为窗体的背景，这种窗体往往用于应用程序的主界面中，如图 1.24 所示。

在窗体中添加图片的具体步骤如下。


- (1) 选中需要加入背景图片的窗体。
- (2) 在“属性”窗口中，找到 Picture 属性，单击按钮, 如图 1.25 所示。在弹出的“加载图片”对话框中选择图片文件。
- (3) 单击“确定”按钮，图片即可被加载到窗体中，结果如图 1.24 所示。



图 1.24 添加图片后的窗体



图 1.25 设置 Picture 属性

1.4 窗体的方法

1.4.1 加载和卸载窗体

1. 加载窗体

在多窗体的应用程序中，经常需要加载或卸载窗体，这时就需要使用 Load 和 Unload 语句。

Load 语句用于将一个窗体装入内存。执行 Load 语句后，可以引用窗体中的控件及各种属性，但此时窗体是不可见的，要想使窗体可见还需要使用 Show 方法。

语法

Load 窗体名

2. 卸载窗体

加载的窗体将占用一部分内存，对于不再使用的窗体应该及时卸载。

语法

Unload 窗体名

注意：如果用 Unload 语句卸载全部窗体，程序将停止执行。

示例

本示例为在采购管理系统中，从菜单栏中选择“采购管理”→“采购员管理”命令，则加载采购员管理窗体 frm_cgygl，如图 1.26 所示，采购员管理窗体中的操作完成后，通过单击“关闭”按钮，卸载该窗体。

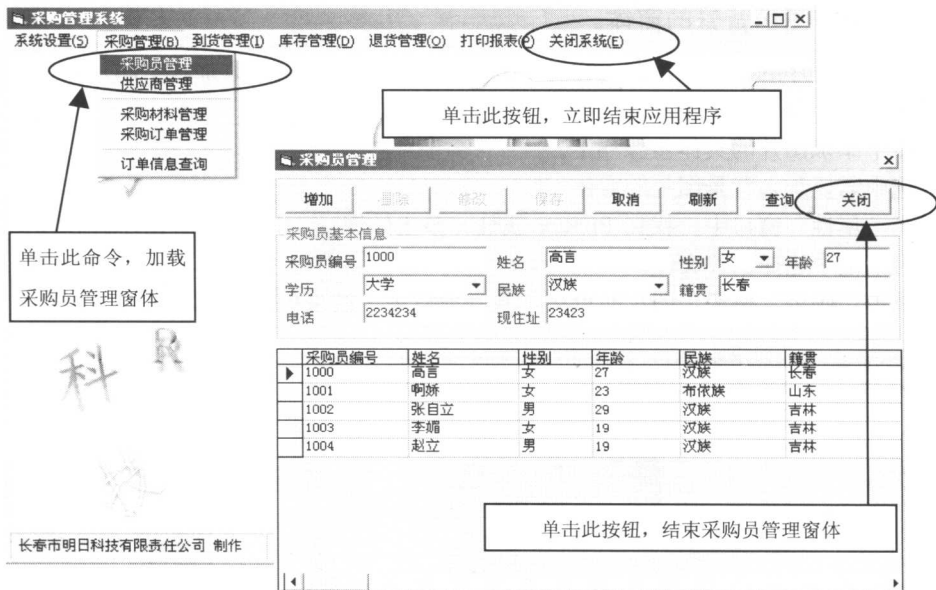


图 1.26 加载和卸载窗体

加载采购员管理窗体的代码如下：

```
Private Sub CGYGL_Click() '采购员管理菜单的单击事件
    Load frm_cgygl '加载采购员管理窗体frm_cgygl
    frm_cgygl.Show '显示frm_cgygl窗体
End Sub
```

卸载采购员管理窗体的代码如下：

```
Private Sub Cmd_exit_Click() '退出按钮的单击事件
    Unload Me '卸载frm_cgygl窗体
End Sub
```

注意：Me 是系统保留字，表示当前窗体。

3. End 语句

End 语句用于立即结束应用程序。一旦应用 End 语句，在其后的代码将不再执行，也不会再有事件发生，对象的各个引用也将被释放。

示例

在图 1.26 所示的采购管理系统主界面中，当用户单击“关闭系统”命令时，将立即结束采

购管理系统程序。具体代码如下：

```
Private Sub END_Click()  
    myval = MsgBox("确认要退出采购管理系统吗?", 33, "采购管理系统")  
    If myval = vbOK Then End  
End Sub
```

1.4.2 窗体的显示与隐藏

1. 显示窗体

Show 方法用来显示一个窗体，且具有装入和显示窗体两种功能，也就是说，在执行 Show 方法时，如果窗体不在内存中，则 Show 方法自动将窗体装入内存，然后显示该窗体。

语法

窗体名称.Show 模式

窗体名称：如果省略窗体名称，则显示当前窗体。

模式：可选的整数，决定窗体是非模态还是模态。如果值为 0，则窗体是非模态的；如果值为 1，则窗体是模态的。

当模式值为 1 时，鼠标只在该窗体内起作用，不能到其他窗体操作，只有关闭该窗体后才能对其他窗体操作；当模式值为 0 时，可以对其他窗体进行操作。

示例

将窗体 frm_mian 显示为模态的代码如下：

```
frm_main.Show 1
```

将窗体 frm_mian 显示为非模态的代码如下：

```
frm_main.Show 0
```

2. 隐藏窗体

Hide 方法使窗体隐藏，即不在屏幕上显示，但仍在内存中。

语法

窗体名称.Hide

如果省略窗体名称，则带有焦点的窗体就是被隐藏的窗体。

示例

本示例为在医院住院管理系统的主界面窗体中单击“用户注册”按钮，将显示用户注册窗体；当在用户注册窗体中单击“隐藏”按钮之后，隐藏用户注册窗体，如图 1.27 所示。

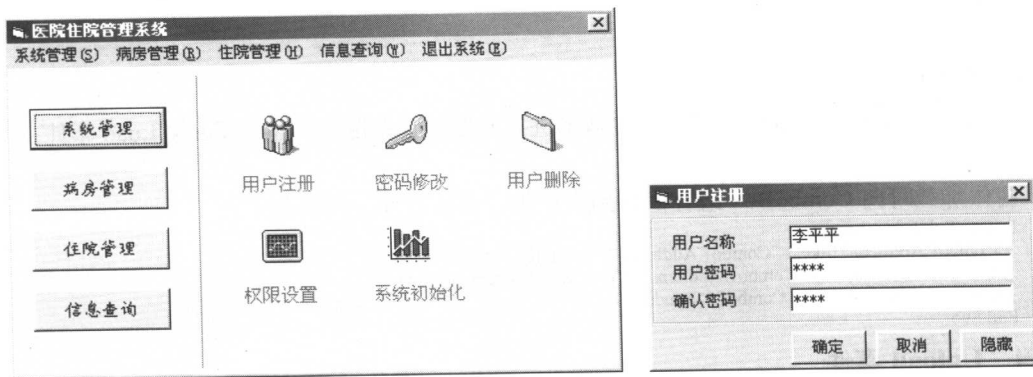


图 1.27 窗体的显示与隐藏示例

在主界面窗体中的“用户注册”按钮的 Click 事件中添加显示用户注册窗体的代码：

```
frm_YHGL.Show
```

在用户注册窗体中的“隐藏”按钮的 Click 事件中添加隐藏窗体的代码：

```
frm_YHGL.Hide
```

1.4.3 将数据输出到窗体上

Print 方法是输出数据、文本的一个重要方法。

语法

窗体名称.Print[<表达式>[, |; [<表达式>]...]]

说明: <表达式>可以是数值或字符串表达式。对于数值表达式,先计算表达式的值,然后输出;如果是字符串则原样输出;如果表达式为空,则输出一个空行。

当输出多个表达式时,各表达式用分隔符(逗号、分号或空格)隔开。若用逗号分隔,将以 14 个字符位置为单位,把输出行分成若干个区段,每区段输出一个表达式的值;表达式之间用分号或空格作为分隔符,则按紧凑格式输出。

一般情况下,每执行一次 Print 方法将自动换行,可以通过在末尾加上逗号或分号的方法,使输出结果在同一行显示。

1.5 窗体的事件

1.5.1 Click 事件

程序运行后,单击窗体的空白部分将触发 Form_Click 事件。若单击窗体上的其他控件,则触发的是控件的单击事件。

示例

单击窗体时设置“确定”按钮(Cmd_yes)不可见,代码如下:

```
Private Sub Form_Click()  
    Cmd_yes.Visible = False  
End Sub
```

1.5.2 DblClick 事件

程序运行后,双击窗体的空白部分将触发 Form_DblClick 事件。

示例

双击窗体空白处时设置“关闭”按钮(Cmd_No)变为可用,其代码如下:

```
Private Sub Form_DblClick()  
    Cmd_No.Enabled = True  
End Sub
```

1.5.3 Load 事件

Load 是窗体的基本事件,只要窗体被加进存储器就自动触发客体的 Form_Load 事件。

示例

窗体加载时向 ComboBox 组合框的列表中加入查询条件,代码如下:

```
Private Sub Form_Load()  
    Combo1.AddItem("Like"): Combo1.AddItem(">"): Combo1.AddItem(">=")  
    Combo1.AddItem("="): Combo1.AddItem("<"): Combo1.AddItem("<=")  
    Combo1.AddItem("<>"): Combo1.ListIndex=0  
End Sub
```

1.5.4 Unload 事件

Unload 事件对应于 Load 事件,是卸载窗体时触发的事件。

示例

关闭客户信息管理窗体时设置主窗体 frm_main 有效,代码如下:

```
Private Sub Form_Unload(Cancel As Integer)  
    frm_main.Enabled = True  
End Sub
```

1.6 窗体与控件

窗体提供了程序窗口，但 Visual Basic 控件却完成程序的大部分任务。这些控件可以显示或编辑文本、显示图片、与数据库交互，并可以完成许多其他任务。由于 Visual Basic 的开放式设计，不仅可以使用 Visual Basic 本身的控件，而且还可以集成第三方编写的控件。正是这个充满生机的第三方控件，人们可以无所不能，做任何想要做的工作，如文字处理、处理流程图、显示图片和创建商业报表等。

1.6.1 向窗体中添加控件

在窗体上添加控件的步骤如下。

- (1) 在工具箱中选取所要添加的控件。
- (2) 将鼠标放置在窗体上，当鼠标指针变成一个十字架时，按住鼠标左键同时拖动鼠标，当达到所需控件的大小时放开鼠标左键，这时控件将被添加到窗体中。

例如，在 Form1 窗体中加入一个 CommandButton 控件，如图 1.28 和图 1.29 所示。

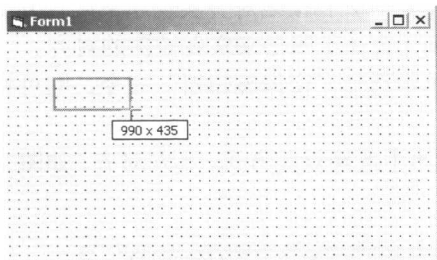


图 1.28 添加控件时鼠标拖动的过程

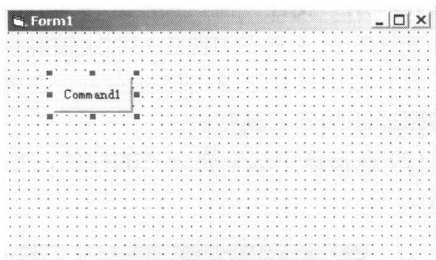


图 1.29 添加完成后的按钮控件

1.6.2 向窗体中添加多个控件

在开发应用程序的过程中，有时需要添加多个相同的控件，这时如果使用上面介绍的方法一个一个的向窗体中添加，会很麻烦。下面介绍两种添加多个控件的快捷方法。

- (1) 按住 Ctrl 键，然后用鼠标选中工具箱中所要添加的控件。
- (2) 选中已添加的控件，单击鼠标右键，在弹出的快捷菜单中选择“复制”命令，然后在窗体的空白处，单击鼠标右键从弹出的快捷菜单中选择“粘贴”命令。

1.6.3 控件对齐

在设计窗体界面的时候，应该尽量将同一类别的控件对齐排列，如将选中的控件左对齐，即控件的 Left 属性相同，这样窗体界面看起来会比较整齐、美观。

1. 利用属性窗口设置控件对齐方式

- (1) 首先选中一个文本控件，然后按住 Ctrl 键，再用鼠标分别选中所要设置的控件，如图 1.30 所示。

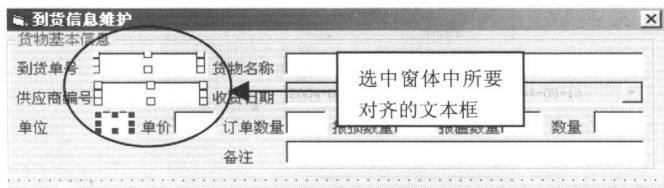


图 1.30 选中窗体中所要对齐的文本框

(2) 在“属性”窗口中选择文本控件的 Left 属性，然后在其右边对应的空白框中输入 Left 属性的属性值，如图 1.31 所示。

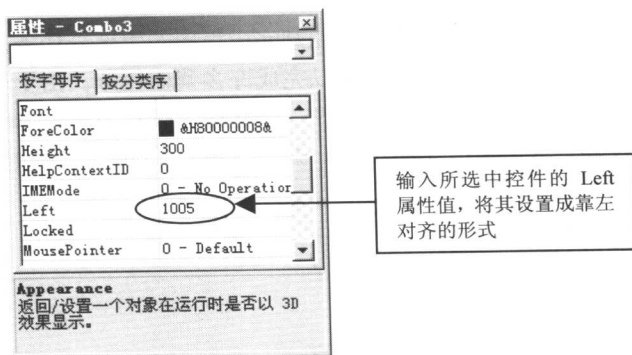


图 1.31 将所选控件设置成靠左对齐的方式

2. 利用窗体编辑器设置控件对齐

在利用“属性”窗口实现控件对齐的过程中，除了左对齐和顶端对齐实现起来比较方便以外，其他的对齐方式实现起来都比较麻烦。下面介绍利用窗体编辑器来实现控件的对齐。

(1) 在工具栏上单击鼠标右键，在弹出的快捷菜单中选择“窗体编辑器”命令，即可将窗体编辑器添加到工具栏中。

(2) 在窗体中选中要对齐的控件，然后在窗体编辑器中选择对其方式，如左对齐，如图 1.32 所示。

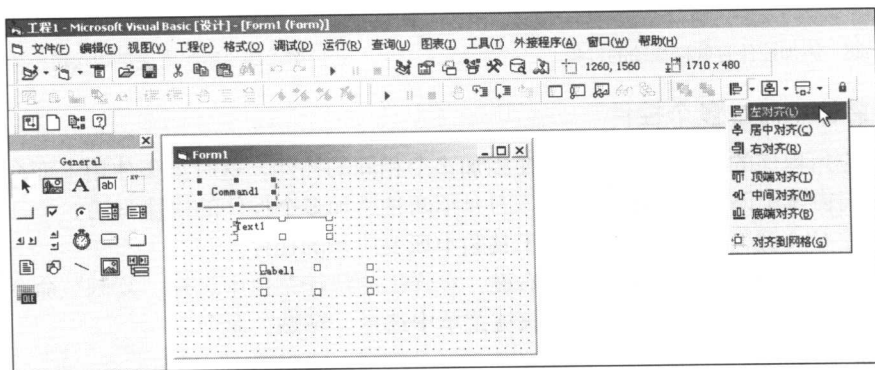


图 1.32 利用窗体编辑器设置控件对齐

或者选中要对齐的控件，在菜单栏中选择“格式”→“对齐”命令，如图 1.33 所示。

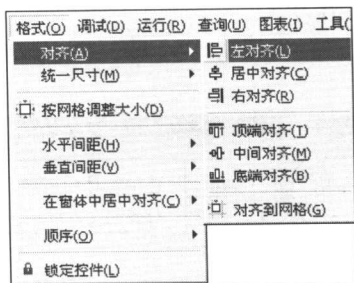


图 1.33 在菜单栏中设置对齐方式

1.6.4 调整光标在控件上移动的先后顺序

可以通过修改控件的 TabIndex 属性值来控制控件的先后顺序。在图 1.34 所示的界面中，将 Text1 控件的 TabIndex 属性值设置成 1，Text2 控件的 TabIndex 属性值设置成 2，Text3 控件的 TabIndex 属性值设置成 3，依次类推。在窗体运行的时候，按下 Tab 键就可以将光标从一个控件移向另一个控件。

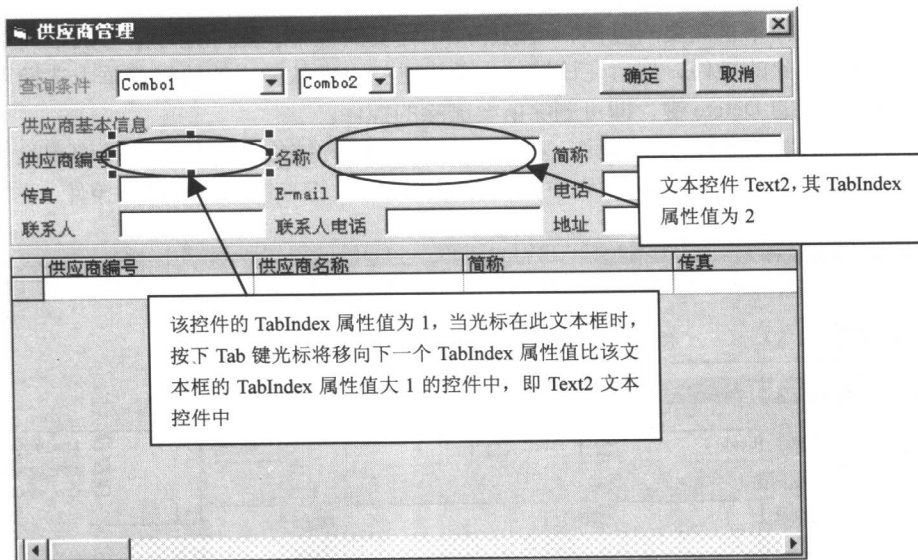


图 1.34 设置控件的先后顺序

也可以用 Enter 键来控制光标在控件上移动的先后顺序。如当按下 Enter 键时光标将从 Text1 文本框中移到 Text2 文本框中，可用下面的代码来实现：

```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = 13 Then Text2.SetFocus
End Sub
```

1.6.5 调整控件在窗体上的前后顺序

当在窗体上的同一层次上放置多个控件的时候，要根据不同的需要设置控件在窗体上摆放的前后顺序。

1. 窗体编辑器设置实现前后顺序

在窗体同一层次控件的上层控件上，单击鼠标右键，在弹出的快捷菜单中设置该控件在窗体上的前后顺序，如图 1.35 所示。

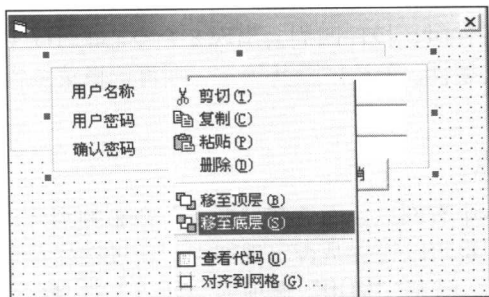


图 1.35 设置窗体中同一层次上控件的前后顺序