



全国高等职业学校•高等专科学校教材

数据库应用程序 开发技术

上海市教育委员会 组编

刘念祖 周 琛

1.132-43

高等教育出版社

第一章 Visual Basic 基础

本章首先介绍 Visual Basic 6.0 的集成开发环境、基本概念和程序结构,并通过一个实例来了解 Visual Basic 的程序开发过程。然后介绍 Visual Basic 的数据类型、变量、常量、数组、过程、运算符、常用内部函数和程序控制结构等知识。掌握这些基本内容,有利于更好地开发数据库应用程序。

1.1 集成开发环境

Visual Basic 6.0 的集成开发环境(Integrated Development Environment,缩写为 IDE)是系统自身的界面,它集成了开发应用程序所需的所有功能,例如界面设计、代码编辑、程序编译和调试等。用 Visual Basic 6.0 进行程序设计,就必须先了解这个工作环境。

1.1.1 启动集成开发环境

在 Windows 9X 中,有多种启动 Visual Basic 6.0 的方法。最基本的方法是单击 Windows 9X 任务栏上的“开始”按钮,把鼠标指针移到“程序”命令上,将弹出一个级联菜单;再把鼠标指针移到“Microsoft Visual Basic 6.0 中文版”,将出现 Visual Basic 6.0 程序组;最后单击“Microsoft Visual Basic 6.0 中文版”,即可进入 Visual Basic 6.0 编程环境。如果已在桌面上建立了启动 Visual Basic 6.0 的快捷方式,那么只要在桌面上双击 Visual Basic 6.0 的快捷

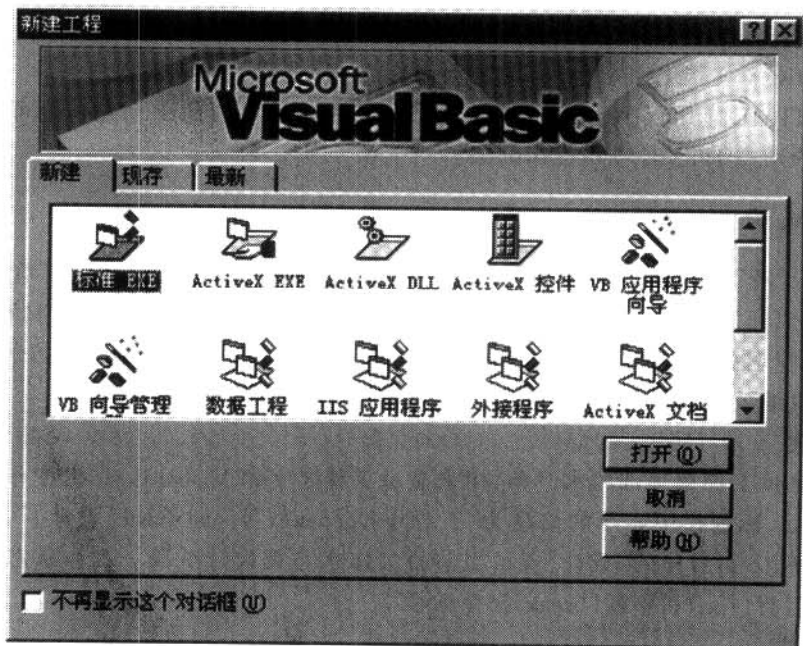


图 1-1-1 “新建工程”对话框

图标,就可进入 Visual Basic 6.0 编程环境。

当启动 Visual Basic 6.0 时,屏幕上将出现“新建工程”对话框,如图 1-1-1 所示。这个对话框有三个选项卡:“新建”选项卡用于让用户选择并创建新的工程类型;“现存”选项卡用于打开已经存在的工程;“最新”选项卡用于打开最近使用的工程。“新建工程”对话框还可通过执行“文件”菜单中的“新建工程”命令来显示,但此时没有“现存”和“最新”选项卡。

Visual Basic 6.0 的工程类型有如下几种:

- (1) 标准 EXE:用于创建一个标准的可执行文件,它是一种较常用的工程类型
- (2) ActiveX EXE(只能在专业版或企业版中建立):它是一个包含公共类的、可以被其他应用程序访问或自己运行的应用程序。
- (3) ActiveX DLL(只能在专业版或企业版中建立):功能与 ActiveX EXE 应用程序一样,但不能单独运行。
- (4) ActiveX 控件(只能在专业版或企业版中建立):用于创建自定义的 ActiveX 控件(OCX),它可以在 Visual Basic 程序或任何支持 ActiveX 的应用程序中使用。
- (5) VB 应用程序向导:用于在开发环境中直接建立新的应用程序框架,然后定制应用程序,以适合用户的需要。
- (6) 数据工程:提供开发数据报表应用程序的框架,选择该图标后,将自动打开数据环境设计器和数据报表设计器。
- (7) IIS 应用程序:用于创建一个用来连接 Windows NT 的 Internet 信息服务器(Internet Information Server,缩写为 IIS)的程序,它是一个在网络环境中运行的工程。
- (8) 外接程序:为 Visual Basic 自身提供了其他功能,并在开发环境中自动打开联接设计器。可视化数据管理器是一个外接程序的实例。
- (9) ActiveX 文档 DLL:用于创建一个能被运行在 Microsoft Internet Explorer (IE)中的应用程序使用的动态链接库。
- (10) ActiveX 文档 EXE:用于创建一个可以在 IE 中运行的应用程序。
- (11) DHTML 应用程序(只能在专业版或企业版中建立):用于创建一个可以在网络浏览器上运行的动态 HTML 应用程序框架。

1.1.2 集成开发环境

在“新建工程”对话框中选择要创建的工程类型(如“标准 EXE”),然后单击“打开”按钮,即进入集成开发环境,如图 1-1-2 所示。它由标题栏、菜单栏、工具栏、工具箱、窗体设计器、工程资源管理器、窗体布局窗口和属性窗口等元素组成。为了能看清每个部分,图中有些部分是通过缩放或移动重新排列的,可能与实际情况有些差异。

1. 标题栏

标题栏是位于屏幕顶部的水平条,用来显示工程名字和 Visual Basic 的工作状态。启动 Visual Basic 后,标题栏中显示的信息为“工程 1-Microsoft Visual Basic[设计]”。其中“工程 1”是工程名字,方括号中的“设计”表示当前的工作状态是设计阶段。工作状态一般分为设计(design)、运行(run)和中断(break)三个阶段。

2. 菜单栏

菜单栏位于标题栏的下面,其菜单项有文件、编辑、视图、工程、格式、调试、运行、查询、

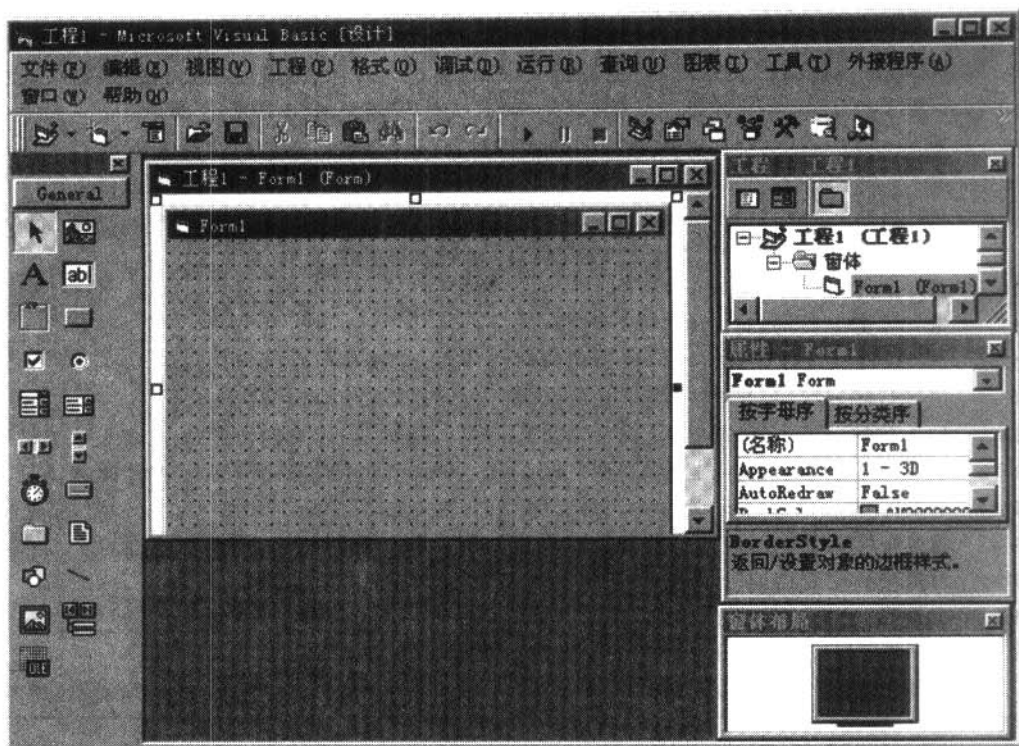


图 1-1-2 集成开发环境

图表、工具、外接程序、窗口和帮助。

3. 工具栏

Visual Basic 6.0 提供编辑、标准、窗体编辑器和调试等 4 种工具栏,方便用户执行菜单中的命令。标准工具栏位于菜单栏的下面,提供常用菜单命令。其他工具栏可以通过“视图”菜单中的“工具栏”命令打开或关闭。标准工具栏图标如图 1-1-3 所示,其名称和作用见表 1-1。

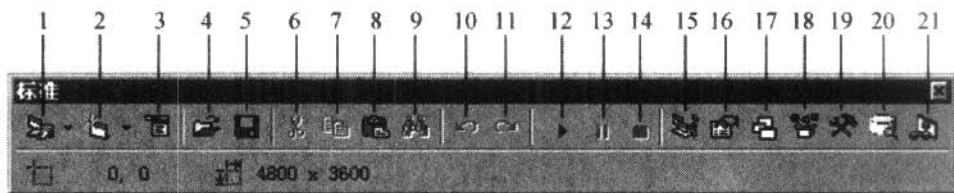


图 1-1-3 标准工具栏图标

表 1-1 标准工具栏详解

序号	名称	作用
1	添加工程	添加一个新工程,等于“文件”菜单中的“添加工程”命令
2	添加窗体	在工程中添加一个新窗体,等于“工程”菜单中的“添加窗体”命令
3	菜单编辑器	打开菜单编辑器,等于“工具”菜单中的“菜单编辑器”命令
4	打开工程	打开已存在的工程文件,等于“文件”菜单中的“打开工程”命令

(续表)

序号	名 称	作 用
5	保存工程	保存当前的工程文件,等于“文件”菜单中的“保存工程”命令
6	剪 切	把选定的内容剪切到剪贴板上,等于“编辑”菜单中的“剪切”命令
7	复 制	把选定的内容复制到剪贴板上,等于“编辑”菜单中的“复制”命令
8	粘 贴	把剪贴板的内容粘贴到当前的插入位置,等于“编辑”菜单中的“粘贴”命令
9	查 找	打开“查找”对话框,等于“编辑”菜单中的“查找”命令
10	撤 消	撤消当前操作
11	重 复	对“撤消”的反操作
12	启 动	用来运行一个应用程序,等于“运行”菜单中的“启动”命令
13	中 断	暂停正在运行的程序,等于“运行”菜单中的“中断”命令
14	结 束	结束程序运行并回到设计窗口,等于“运行”菜单中的“结束”命令
15	工程资源管理器	打开工程资源管理器,等于“视图”菜单中的“工程资源管理器”命令
16	属性窗口	打开属性窗口,等于“视图”菜单中的“属性窗口”命令
17	窗体布局窗口	打开窗体布局窗口,等于“视图”菜单中的“窗体布局窗口”命令
18	对象浏览器	打开对象浏览器,等于“视图”菜单中的“对象浏览器”命令
19	工具箱	打开工具箱,等于“视图”菜单中的“工具箱”命令
20	数据视图窗口	打开数据视图窗口,等于“视图”菜单中的“数据视图窗口”命令
21	组件管理器	管理系统中的组件,等于“视图”菜单中的“Visual Component Manager”命令

4. 工具箱

工具箱一般位于屏幕的左侧,内有构成用户界面所需的各种控件,每个控件有一个图标,用户只要单击所需的控件图标,就能使用该控件。工具箱中的控件分为两类:一类是标准控件(又称内部控件),另一类是 ActiveX 控件。启动 Visual Basic 6.0 后,工具箱中只有常用的标准控件。标准控件图标如图 1-1-4 所示,其名称和作用见表 1-2。

表 1-2 标准控件的作用

序 号	名 称	作 用
1	指针(Pointer)	这不是一个控件,选择它,才能改变窗体中控件的位置或大小
2	图片框(PictureBox)	用于显示图片,也可做为其他控件的容器
3	标签(Label)	显示(输出)用户不能编辑的文本信息
4	文本框(TextBox)	既能输入也能输出文本信息,并可对文本进行编辑
5	框架(Frame)	作为其他控件的容器,可组合相关的控件
6	命令按钮(CommandButton)	单击此按钮,可向应用程序发命令,执行指定的操作
7	复选框(CheckBox)	又称检查框,用于多重选择
8	选项按钮(OptionButton)	又称单选按钮,只能在一组项目选择一个项目
9	组合框(ComboBox)	由文本框和列表框组成,既可在列表中选择,又可直接输入
10	列表框(ListBox)	在列表中显示项目并供用户选择
11	水平滚动条(HScrollBar)	用户可根据水平滚动条中滑块的位置,设置相应的值
12	垂直滚动条(VScrollBar)	用户可根据垂直滚动条中滑块的位置,设置相应的值
13	定时器(Timer)	定时触发事件
14	驱动器列表框(DriveListBox)	显示驱动器列表,供用户选择使用

(续表)

序 号	名 称	作 用
15	目录列表框(DirListBox)	显示当前驱动器上的目录列表,供用户选择使用
16	文件列表框(FileListBox)	显示当前目录中的文件列表,供用户选择使用
17	图形(Shape)	在窗体中绘制图形,如矩形、圆和椭圆等
18	直线(Line)	在窗体中画直线
19	图像(Image)	显示图像,可作为背景或装饰的图像元素
20	数据(Data)	提供对数据库的链接
21	OLE 容器(OLE)	用于对象的连接与嵌入

5. 窗体设计器

窗体设计器简称为窗体(Form),在设计阶段用来存放各种所需的控件(如文本框、命令按钮等),以构成用户能见到并与之交流的界面。实际上,窗体不仅包含用户能看到的对象,还包含用户不能看到的程序代码。

当启动一个新的“标准 EXE”工程时,屏幕就显示如图 1-1-5 所示的窗体。窗体好像是一块画布,程序员可根据程序界面的要求,从工具箱中选择所需的控件放在画布上,这就是 Visual Basic 应用程序设计的第一步工作。窗体上的点状网格供程序员调整界面布局之用,其点距等特性的调整,可通过“工具”菜单中“选项”命令中的“通用”选项卡来实现,如图 1-1-6 所示。

6. 工程资源管理器

工程资源管理器列出了当前工程或工程组中的所有窗体和模块文件,工程是指用于创建一个应用程序的文件集合。工程资源管理器中的文件分为窗体文件(*.frm)、程序模块文件(*.bas)、类模块文件(*.cls)、工程文件(*.vbp)和工程组文件(*.vbg)。

(1) 窗体文件

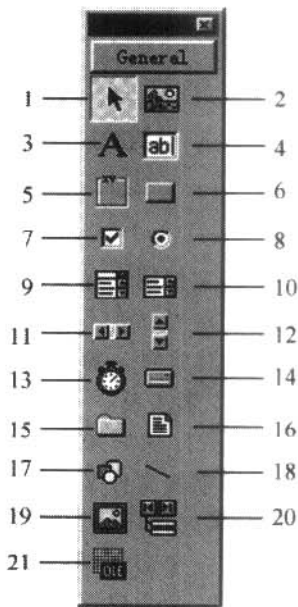


图 1-1-4 标准控件图标

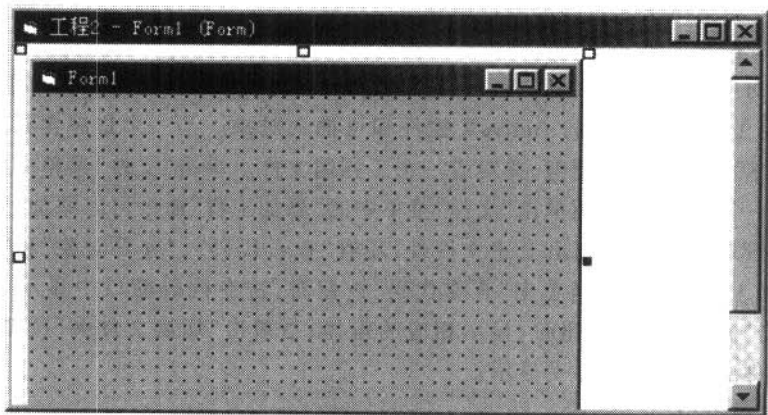


图 1-1-5 窗体

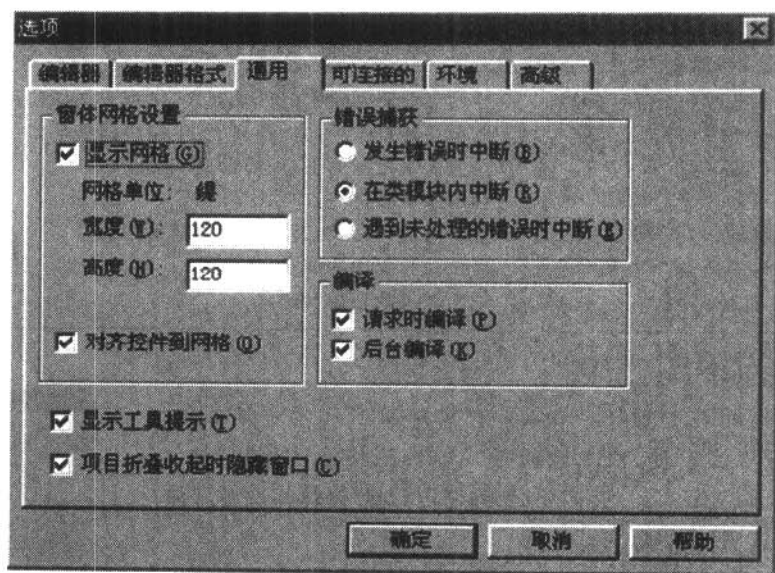


图 1-1-6 用“通用”选项卡设置窗体网格特性

窗体文件用来存放窗体及其控件的属性和其他信息。每个窗体对应一个窗体文件,一个应用程序可以有不超过 255 个的窗体。当添加一个窗体时,工程资源管理器中会自动增加一个窗体文件。窗体的添加可单击工具栏中的“添加窗体”按钮或“工程”菜单中的“添加窗体”命令来实现。

(2) 程序模块文件

程序模块文件又称标准模块文件。程序模块文件主要用来声明全局变量和定义一些通用过程,属纯代码性质的文件,它不包含任何控件、图形或其他可视信息。程序模块文件不属于任何一个窗体,但可以被不同窗体的程序调用。程序模块文件可通过“工程”菜单中的“添加模块”命令建立。

(3) 类模块文件

每个类模块文件用来保存一个类。Visual Basic 提供了大量预定义的类,同时也允许用户根据需要通过类模块来定义自己的类。类模块可通过“工程”菜单中的“添加类模块”命令建立。

(4) 工程文件和工程组文件

每个工程对应一个工程文件。工程文件是该工程所涉及的窗体文件和模块文件的集合,它还能引用工程中所使用的 ActiveX 控件和可插入对象。当一个程序含有两个以上的工程时,这些工程就构成一个工程组,并对应一个工程组文件。单击“文件”菜单中的“新建工程”命令可以建立一个新的工程;而单击“文件”菜单中的“添加工程”命令可以添加一个工程。

工程资源管理器窗口除列出上述文件外,还有“查看代码”、“查看对象”和“切换文件夹”三个按钮,如图 1-1-7 所示。当单击“查看代码”按钮时,将在代码窗口中显示当前文件的相应代码;当单击“查看对象”按钮时,显示相应的窗体;当单击“切换文件夹”按钮时,显示各类文件所在的文件夹或取消文件夹显示。

7. 窗体布局窗口

窗体布局窗口允许使用表示屏幕的小图像来布置应用程序中各窗体的位置。

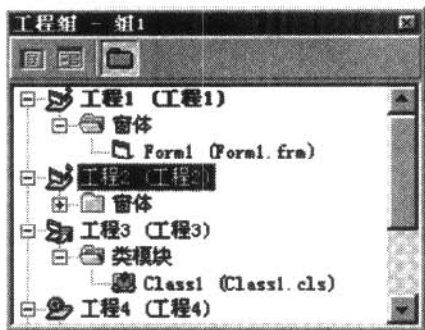


图 1-1-7 工程资源管理器



图 1-1-8 属性窗口

8. 属性窗口

属性窗口用来设置窗体和控件的属性值。属性窗口(如图 1-1-8)除窗口标题外,还有对象框、属性显示方式、属性列表和当前属性的简单解释四个部分。

在属性窗口中,单击对象框旁的下拉列表按钮,能列出应用程序中每个对象的名字及其类型,供用户选择。属性显示方式有按字母序和按分类序两种,可以通过单击相应的选项卡来实现。属性列表可以滚动显示当前对象的所有属性,以便查看或设置当前对象的每个属性值。当前属性的简单解释部分主要是显示当前属性的名称和功能说明。

除上面介绍的几种窗口外,在集成开发环境中还有其他一些窗口,如代码编辑窗口、立即窗口、本地窗口和监视窗口等。

1.1.3 集成开发环境的设置

Visual Basic 具有很大的灵活性,可以通过设置工作环境来满足个人风格的需要。例如可以选择单文档界面(SDI)或多文档界面(MDI),可以调节集成开发环境中各种元素的大小和位置。

1. SDI 或 MDI 界面

Visual Basic 集成开发环境有两种不同的界面,即单文档界面和多文档界面。SDI 界面的所有窗口可在屏幕上任何地方自由移动;而 MDI 界面的所有子窗口包含在一个大小可调的父窗口内。

在缺省情况下,Visual Basic 集成开发环境为 MDI 界面方式,如图 1-1-2 所示。若要将 MDI 界面切换成 SDI 界面,可单击“工具”菜单中的“选项”命令来打开“选项”对话框,然后选择“高级”选项卡中的“SDI 开发环境”复选框,如图 1-1-9 所示,最后单击“确定”按钮。这样在下次启动 Visual Basic 时,集成开发环境就成为 SDI 界面,如图 1-1-10 所示。

2. 可连接的窗口

为了便于充分利用屏幕显示区域,集成开发环境中的许多窗口能相互连接排列,如图 1-1-2 中工程资源管理器、属性窗口和窗体布局窗口被连接放置在屏幕的右侧。可连接放置

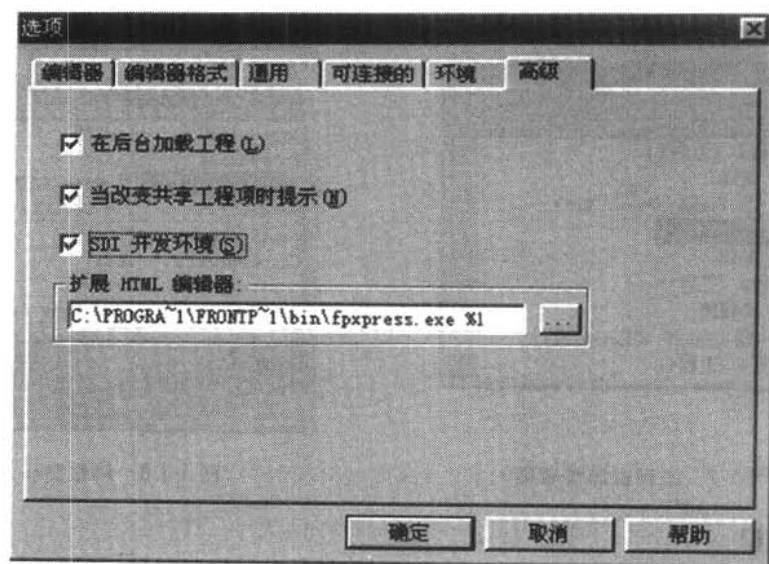


图 1-1-9 设置 SDI 开发环境

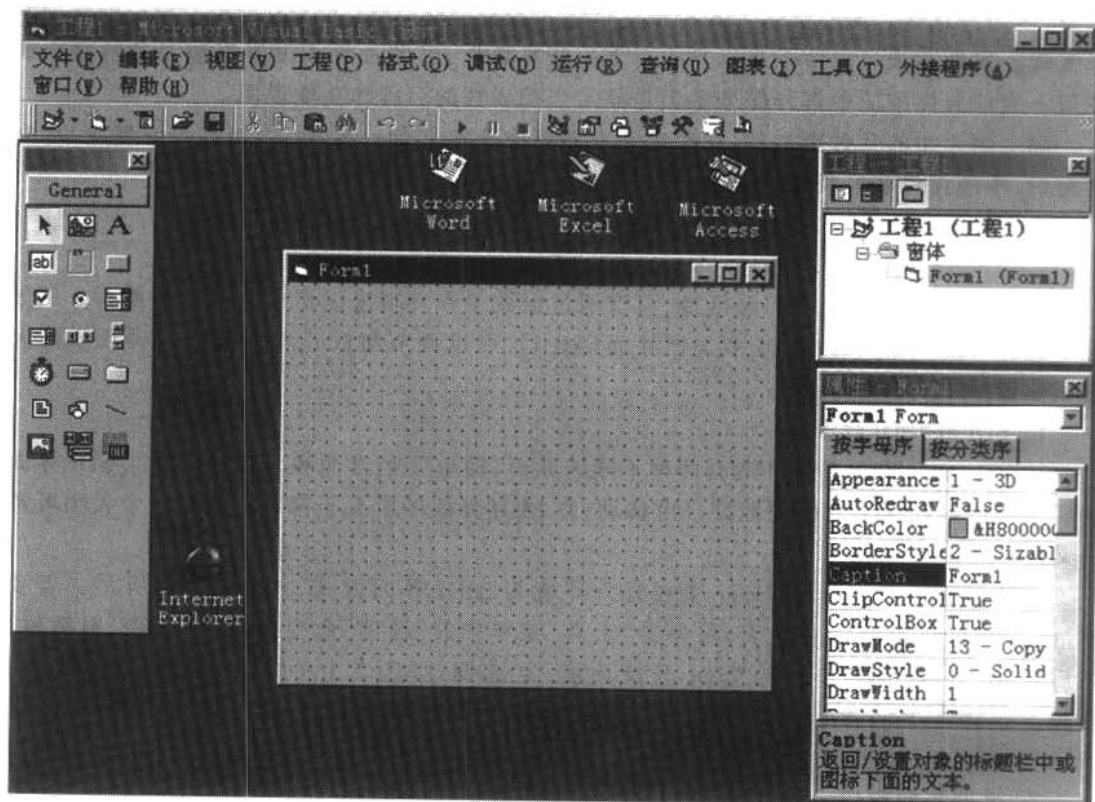


图 1-1-10 SDI 界面

的窗口有立即窗口、本地窗口、监视窗口、工程资源管理器、属性窗口、对象浏览器、窗体布局窗口、工具箱和调色板。

连接窗口的设置,可单击“工具”菜单中的“选项”命令来打开“选项”对话框,然后在“可连接的”选项卡中选中要连接窗口的复选框,如图 1-1-11 所示,最后单击“确定”按钮。

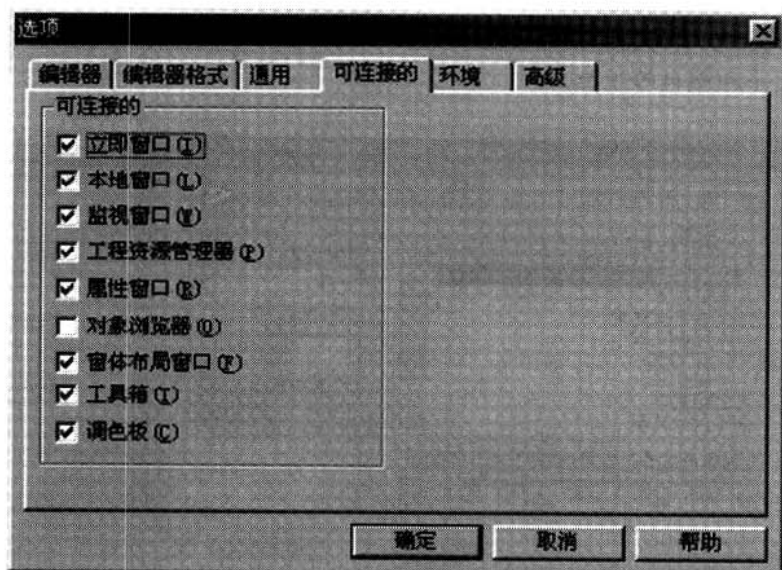


图 1-1-11 设置可连接的窗口

3. 设置编辑器

单击“工具”菜单中的“选项”命令,打开“选项”对话框,在“编辑器”选项卡中可进行编辑器的代码设置和窗口设置,如图 1-1-12 所示。

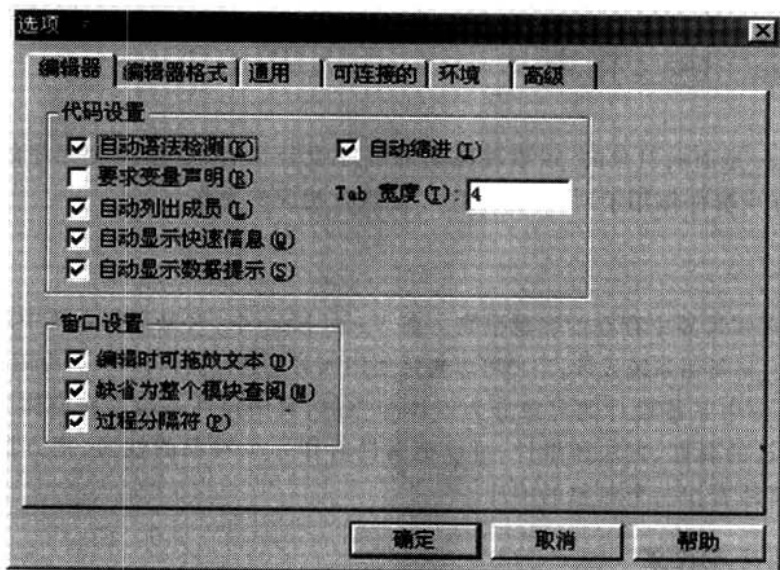


图 1-1-12 设置编辑器特性

代码设置包括在键入一行代码时是否要进行自动语法检测,在模块中是否要求显式变量声明,是否自动列出成员,是否自动显示有关函数及其变量的说明信息,在中断模式中是否要自动显示有关数据提示,换行时是否自动缩进,以及[Tab]键的宽度设置。

窗口设置包括是否支持在编辑时可拖放文本,是否将新模块设置到缺省状态,以及是否在代码编辑窗口显示过程分隔符。

在“选项”对话框中,选择“编辑器格式”选项卡(如图 1-1-13),用户可以设置代码中各种文本的字体、大小和颜色等属性。

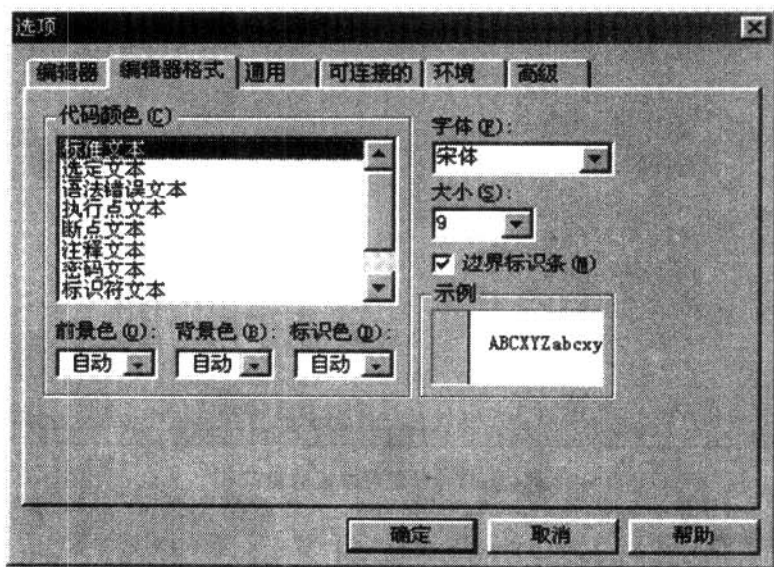


图 1-1-13 “编辑器格式”选项卡

1.2 Visual Basic 概述

Visual Basic 是面向对象的、以事件驱动的程序设计语言,这意味着程序的界面是由对象组成的,只有当事件作用于对象时,程序才会执行相应的操作。

1.2.1 对象

对象是指现实世界中存在的客观事物。在 Visual Basic 中,对象是指系统中的基本运行实体,它分为两类:一类由系统设计,用户可以直接使用或对其进行操作;另一类由用户自己定义。

Visual Basic 中系统设计的对象分为窗体和控件,其中控件是包括在窗体内的对象。这些对象都有自己的属性、方法和事件,可以把属性看作一个对象的性质,把方法看作一个对象的动作,把事件看作一个对象的响应。

1.2.2 属性及其设置

属性是指对象的特性,不同的对象有不同的属性。常见的对象属性有 Caption、Name、

Color、Fontsize 和 Visible 等。

对象的属性可以在属性窗口中设置,方法是先选择要设置属性的对象,然后激活属性窗口进行设置。Visual Basic 中激活属性窗口有以下几种方法:

- (1) 直接按[F4]键。
- (2) 单击“视图”菜单中的“属性窗口”命令。
- (3) 单击工具栏中的“属性窗口”按钮。

设置不同的属性,要使用不同的设置方式,在 Visual Basic 中通常有以下三种方式。

1. 直接键入属性值

对于有些属性,用户必须直接键入其属性值。如 Caption 属性,一般要求用户选择有意义的名称作为其属性值,这样可以提高程序的可读性。

2. 下拉列表中选择属性值

有些属性(如 BorderStyle、ControlBox、DrawStyle 和 DrawMode 等)的取值范围有限,一般通过下拉列表选择所需要的属性值,如图 1-2-1 所示。



图 1-2-1 属性窗口(下拉列表)

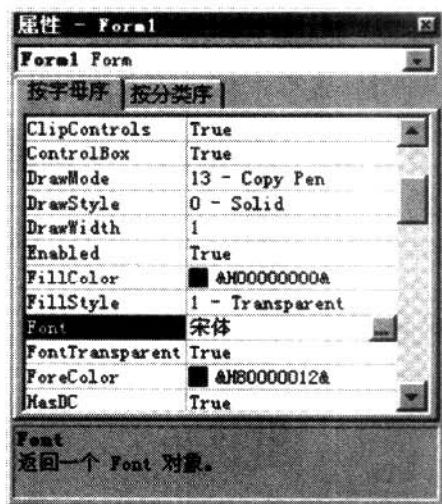


图 1-2-2 属性窗口(省略号按钮)

3. 使用对话框设置属性值

对于与 Picture、Icon 或 Font 有关的属性,设置框的右边会显示省略号按钮,如图 1-2-2 所示。单击此按钮,屏幕上将显示一个对话框,此时用户就可以根据需要设置属性了。

对象的属性除了在属性窗口中设置外,还可以在程序中用语句设置,其一般格式为:

对象名.属性名 = 属性值

1.2.3 方法

方法是指用户可以对对象进行的操作,它是封装在对象中的程序函数,定义了对对象可以执行的任务。不同的对象有不同的方法,但大多数对象都有 Drag 和 Move 等共同的方法。调用对象方法的格式为:

对象名.方法名

如果在调用方法时省略对象名, Visual Basic 则认为所调用的方法是当前对象的方法。

1.2.4 事件及事件过程

事件是预先设置的、能被对象识别的动作。例如 Click、DblClick、Load、MouseMove 等。不同的对象能识别的事件也不同,当事件由用户触发(如 Click)或由系统触发(如 Load)时,对象就会对该事件作出响应。

事件过程是指对象响应事件时所执行的程序代码。一个对象可以识别一个或多个事件,因此可以使用一个或多个事件过程对用户或系统的事件作出响应。事件过程的程序代码可以在代码窗口中编写或修改,打开代码窗口有以下三种方法:

- (1) 双击已建立的对象(窗体或控件)。
- (2) 单击“视图”菜单中的“代码窗口”命令,或直接按[F7]键。
- (3) 在工程资源管理器中,单击“查看代码”按钮。

代码窗口打开时,系统将自动给出事件过程的开头“Private Sub 对象名_事件名()”和结尾“End Sub”,如图 1-2-3 所示。事件过程名由对象名和该对象的事件名组成,事件过程名不能由用户直接输入,只能根据需要在对象框和过程框中任意选择组合。事件过程名确定后,用户可以在事件过程的开头和结尾之间编写程序,来设定响应事件的代码。

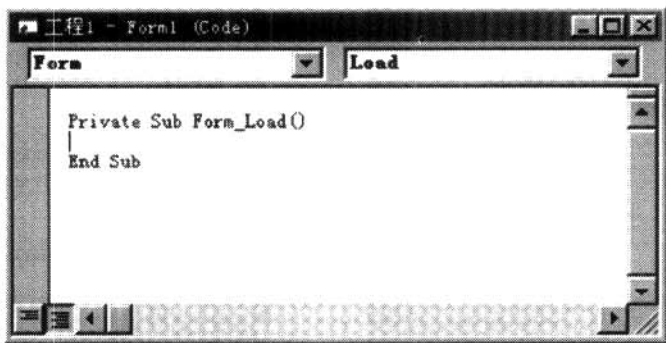


图 1-2-3 代码窗口

1.2.5 属性、方法和事件的相互作用

本节通过一个实例,来介绍对象的属性、方法和事件之间的相互作用。

在日常生活中,小孩玩的气球(Balloon)可看作一个对象,它具有属性、方法和事件。气球的属性包括一些可见的性质,如颜色(Color)、直径(Diameter)和充气(Inflated)状态等,也有不可见的性质,如它的寿命。通过定义,所有气球都具有这些属性,这些属性也会因气球的不同而不同。气球还具有本身所固有的方法和动作,如充气(Inflate)方法、放气(Deflate)方法和上升(Rise)方法等,所有的气球都具备这些能力。气球还有预定义的对某些外部事件的响应,例如气球对刺破它的事件响应是放气,对放手事件的响应是升空。

如果能对气球这一对象进行编程的话,则其 Visual Basic 程序代码可如下表示。设置气球属性的代码是:

```
Balloon.Color = Red
```

```
Balloon.Diameter = 10
Balloon.Inflated = True
```

调用气球的方法是：

```
Balloon.Inflate
Balloon.Deflate
Balloon.Rise 5
```

可以看出,对象名是一个名词,方法名是一个动词,它们之间用圆点“.”分隔。在第三行代码中有一个附加项,称为参数,表示气球上升的高度。一个方法可以有一个或多个参数,它们对执行的动作做进一步的描述。

气球对一个事件的响应如下所示：

```
Private Sub Balloon_Puncture()
    Balloon.Deflate
    Balloon.MakeNoise "Bang"
    Balloon.Inflated = False
    Balloon.Diameter = 1
End Sub
```

本例事件过程中的代码描述了当刺穿(Puncture)事件发生时,气球的行为是首先调用 Deflate 方法,然后以“Bang”(发出的声响)为参数调用 MakeNoise 方法。因为气球已不再是充气状态,所以把 Inflated 属性置为“False”,并给 Diameter 属性设置一个新值。

尽管实际上对气球不能编程,但是用户可以对 Visual Basic 的窗体或控件进行编程。程序员自己可以决定应更改哪些属性,调用哪些方法,以及对哪些事件作出响应,从而得到希望的外观和行为。

1.3 Visual Basic 程序结构

Visual Basic 的程序结构与传统的面向过程的程序结构不同,它采用事件驱动的编程机制,这样能大大降低编程的难度和工作量,提高程序的开发效率。

1.3.1 Visual Basic 程序的结构

计算机应用程序只不过是计算机为完成一个特定任务而执行的一系列指令。应用程序的结构是组织指令的方法,也就是指令存放的位置和指令的执行顺序。对于只有一行代码的程序来说,其组织结构并不重要,应用程序越复杂,对其组织结构的要求也越高。除了对应用程序的执行进行控制外,如何在应用程序中方便地查找特定指令,结构也起着很重要的作用。

Visual Basic 应用程序通常由窗体模块、标准模块和类模块组成。

1. 窗体模块

Visual Basic 应用程序是基于对象的,所以应用程序的代码结构就是该程序在屏幕上的物理表示的模型。根据定义,对象包含数据和代码。在屏幕上看到的窗体对象是由其属性规定的,这些属性定义了窗体的外观和内在特性。在 Visual Basic 中,一个应用程序含有一

个或多个窗体模块(其文件扩展名为“frm”),每个窗体模块都包含事件过程,即代码部分,其中有为响应特定事件而执行的指令。窗体可包含控件,这样在窗体模块中就有事件过程集来对应窗体上的每一个控件。除了事件过程,窗体模块还可包含通用过程,它可以被窗体模块中的任何事件过程调用。

2. 标准模块

标准模块(其文件扩展名为“bas”)完全由代码组成,这些代码与具体的窗体或控件没有关系,它用来声明全局变量、定义函数过程或子程序过程,其中的过程可被不同对象中的事件调用

3. 类模块

类模块(其文件扩展名为“cls”)可看作是没有物理表示的控件。用类模块创建的对象可被应用程序内的过程调用。标准模块只包含代码,而类模块既包含代码又包含数据。

1.3.2 Visual Basic 程序的工作方式

事件是可以由窗体或控件识别的动作。在响应事件时,事件驱动应用程序执行指定的代码。Visual Basic 的每一个窗体和控件都有一个预定义的事件集,如果其中有一个事件发生,并在相关联的事件过程中存在代码,那么 Visual Basic 将执行该代码。

尽管 Visual Basic 中的对象能自动识别预定义的事件集,但还是要通过代码来判定它们是否能响应具体事件以及如何响应具体事件。代码(即事件过程)与每个事件有关,为了让窗体或控件响应某个事件,就必须把代码写入这个事件的事件过程之中。

对象所能识别的事件类型多种多样,但多数类型为大多数对象所共有,例如大多数对象都能识别 Click 事件。如果单击窗体,则执行窗体的单击事件过程中的代码;如果单击命令按钮,则执行命令按钮的单击事件过程中的代码。

事件驱动应用程序的典型动作序列为:

- (1) 启动应用程序,装载和显示窗体;
- (2) 窗体或窗体上的控件接受事件,事件可以由用户(例如键盘操作)或系统(例如定时器事件)引发,也可以由代码间接引发(例如当代码装载窗体时的 Load 事件);
- (3) 如果在相应的事件过程中存在代码,就执行该代码;
- (4) 应用程序等待下一次事件。

注意,有些事件会伴随其他事件的发生。例如在 DblClick 事件发生时,MouseDown、MouseUp 和 Click 事件也会发生。

1.3.3 Visual Basic 程序开发的一般步骤

Visual Basic 的对象已被抽象为窗体和控件,大大简化了程序设计过程,因而在 Visual Basic 环境下,能以较快的速度和较高的效率开发具有良好的用户界面的应用程序。Visual Basic 应用程序的开发一般需要以下三步:

(1) 建立可视用户界面

用户界面由窗体和控件组成,所有的控件都放在窗体上(一个窗体最多可容纳 255 个控件),程序中所有的信息都要通过窗体显示出来,它是应用程序的用户界面。在应用程序中要用到哪些控件,就在窗体上建立相应的控件。程序运行后,将在屏幕上显示由窗体和控件组成的用户界面。

如果要建立新的窗体,可以单击“工程”菜单中的“添加窗体”命令来实现,此时窗体的缺省名为“Form1”,在这个窗体上就可以设置用户界面。

(2) 设置可视界面特性

在建立用户界面后,就可以设置窗体及其控件的属性了。在实际的应用程序设计过程中,建立界面和设置属性可以同时进行,即每画完一个控件,接着就设置该控件的属性。当然,也可以在所有对象都建立之后,再回过头来设置每个对象的属性。究竟采用哪种方法,可以根据自己的习惯而定。

(3) 编写事件驱动代码

我们已经知道 Visual Basic 采用事件驱动编程机制,因此大部分程序都是针对窗体中各个控件所能支持的方法或事件编写的,这样的程序称为事件过程。例如,命令按钮可以接收鼠标事件,如果单击该按钮,鼠标事件就调用相应的事件过程来作出响应。

在多数情况下,特别是小型应用程序,所编写的程序通常都由事件过程组成,即针对控件或窗体的事件编写代码。

1.3.4 Visual Basic 程序的开发例子

1. 简单的 Visual Basic 程序

这个简单的 Visual Basic 应用程序要求在屏幕上开辟一个窗口,窗口的上部有一个文本框,下部有三个命令按钮,其中左边按钮上标有“显示按钮”,中间按钮上标有“清屏按钮”,右边按钮上标有“结束按钮”。当用鼠标单击“显示按钮”时,在文本框中显示“数据库应用程序开发技术”;如果单击“清屏按钮”,则清除文本框中显示的内容;如果单击“结束按钮”,则结束程序。

下面是开发这个应用程序的具体步骤。

(1) 建立可视用户界面

为了建立一个应用程序,先要新建一个工程。单击“文件”菜单中的“新建工程”命令,在“新建工程”对话框中双击“标准 EXE”图标,建立一个新的工程,同时生成一个名为“Form1”的窗体。

一个工程应包括对象和代码,其中对象是指窗体及其中的控件,窗体是程序运行时的背景窗口和对话框,控件是在窗体中的具体对象,而代码是控制运行的程序。

建立可视用户界面实际上就是在窗体上画出所需的各个控件对象。本例除了一个窗体对象外,还需要在窗体上建立四个控件对象,即一个文本框和三个命令按钮,具体操作如下:

① 单击工具箱中的文本框图标,然后在窗体的上方画出文本框控件,此时文本框内自动显示该控件的缺省标题“Text1”。

② 单击工具箱中的命令按钮图标,在窗体的适当位置画出第一个命令按钮,此时在该命令按钮上显示按钮的缺省标题“Command1”,另外两个命令按钮可以按同样的方法画出,它们的缺省标题分别为“Command2”和“Command3”。

四个控件对象画完后,还要根据要求对它们的大小、位置作适当的调整,这样可视用户界面就设计完毕,如图 1-3-1 所示。

(2) 设置可视界面特性

建立可视用户界面后,每个对象都有自己的缺省标题,它们分别为 Form1、Text1、

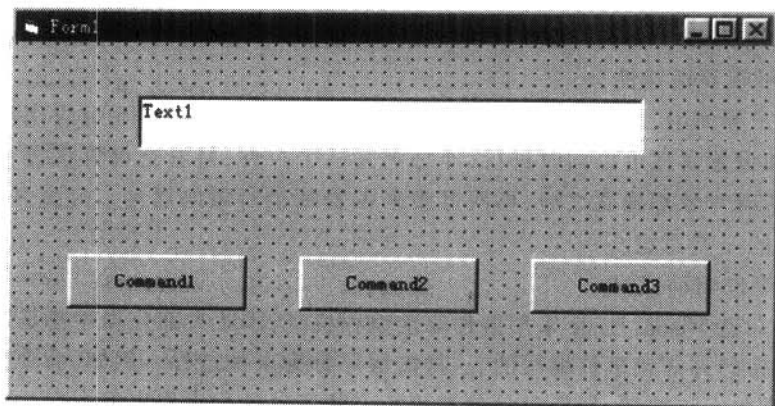


图 1-3-1 用户界面

Command1、Command2 和 Command3。标题(Caption)是对象的一种属性,为了使对象的标题符合设计要求,应该对每个对象的 Caption 属性进行修改。

窗体的缺省标题为“Form1”,要求程序运行时窗体的标题为“Visual Basic 应用程序设计示例”,可按如下步骤修改:

① 单击窗体中没有控件的地方,这时窗体周围出现 8 个小方块,表明窗体已被激活,成为当前活动的对象。

② 在属性窗口中单击 Caption 属性,输入“Visual Basic 应用程序设计示例”(引号不要输入)替代其缺省标题“Form1”。

文本框是用来显示信息的,应该把它变为空白,其修改步骤为:

① 单击文本框,将其激活。

② 删除属性窗口中 Text 属性内的内容。

本例子中要求左边的命令按钮标题为“显示按钮”,可按如下步骤修改:

① 单击左边的命令按钮,将其激活。

② 在属性窗口中单击 Caption 属性,然后输入“显示按钮”,来替代其缺省标题“Command1”。

③ 输入文字后,可以对文字的字体、字体样式和大小进行设置。方法是单击属性窗口中的 Font 属性,此时在该属性条的右端出现一个带有省略号的按钮,再单击这个按钮,即打开“字体”对话框,如图 1-3-2 所示。在本例中把字体设为“隶书”,字体样式设为“粗体”,大小设为“三号”。

如前所述,对象可以有多种属性,以上操作只涉及到 Text、Caption 和 Font 属性。图 1-3-3 是设置属性后的窗体。

对象还有一个常用的属性是 Name 属性,它表示对象的名称,其缺省值与 Caption 属性的缺省值一样。本例中窗体、文本框和三个命令按钮的 Name 属性缺省值分别为 Form1、Text1、Command1、Command2 和 Command3。要注意 Caption 属性和 Name 属性是两个完全不同的属性,Caption 是对象的标识,而 Name 是对象的名称,所以在属性窗口中,Name 的属性条为“(名称)”,并放在属性窗口的第一条,当然在程序代码中,还是使用“Name”。

(3) 编写事件驱动代码