

## 第 2 章 Visual Basic 程序设计初步



- 掌握 Visual Basic 编程的关键性概念，掌握编程的一般步骤。
- 掌握窗体、标签、文本框和命令按钮的使用。
- 掌握程序调试的基本概念和方法。

### 2.1 面向对象编程的基本概念

Visual Basic 作为一种面向对象的程序设计语言 (OOP)，程序的核心是对象。VB 中的窗体和控件都是对象，数据库也是对象，理解对象的概念对 VB 程序设计十分重要。本节将在详细介绍对象和类的概念的基础上，通过具体编程实例介绍 Visual Basic 程序设计的基本方法、步骤和相关的概念。

#### 2.1.1 对象和类的概念

在现实生活中，任何一个可见的实体都可以视为一个对象 (Object)。如一个气球是一个对象，一台电脑也是一个对象，也可以说每一个可见的实体都是一个对象。整个世界就是由大大小小的、各种不同的对象构成的。每个对象都有自己的特征、行为和作用在该对象上的活动。如某台具体的电梯对象，它具有载客量、运行速度等特征，还具有上升、下降等行为，以及外界作用在该电梯对象上的动作，如“按了上升按钮”。

在现实生活中，许多对象具有相同或相似的特征，执行相同的操作，称之为同一类对象。类是对一组对象中共性的属性和行为进行的抽象。如人类是“人”的抽象。

在 VB 中，工具箱中包含了许多工具，例如文本框、标签框、命令按钮等，它们以不同的图标形式排列在工具箱中，用这些工具可以在窗体上画出各种各样的控件对象。工具箱中每一个图标就代表了一个控件 (Control) 类。这些控件类就是一种“对象”的抽象。在程序设计过程中，把一个控件放在窗体上就是创建一个具体的控件对象。一个具体的窗体也是一个对象。窗体除了本身是对象外，它还是其他控件的容器，因为创建的控件对象必须放在窗体上。

#### 2.1.2 对象的属性

每个对象都有一组特征，称之为属性。不同的对象有不同的属性，如小孩玩的气球所具有的属性包括可以看到的一些性质，如它的直径、颜色以及描述气球状态的属性 (充气的或未充气的)。还有一些不可见的性质，如它的寿命等。通过定义，所有气球都具有这些属性，当然这些属性也会因气球的不同而不同。

在可视化编程中，每一种对象都有一组特定的属性。有许多属性可能为大多数对象所共



有,如 BackColor 属性定义对象的背景色。还有一些属性只局限于个别对象,如只有命令按钮才有 Cancel 属性,该属性用来确定命令按钮是否为窗体默认的取消按钮。

每个对象属性都有一个默认值,如果不明确地改变该值,程序将使用它。通过修改对象的属性能够控制对象的外观和操作。对象属性的设置一般有两条途径。

(1) 选定对象,然后在属性窗口中找到相应的属性直接设置。选择这种方法的特点是简单明了,每当选择一个属性时,在属性窗口的下部就显示该属性的一个简短提示,缺点是不能设置所需的所有属性。

(2) 在代码中通过编程设置,格式如下:

对象名.属性名=属性值

例如在程序界面设计中,可以直接选定窗体 Form1,然后在属性窗口中将其 Caption 属性改为“例题”;也可以在代码窗口的程序中添加一条语句来实现:

```
Form1.Caption = "例题"
```

### 2.1.3 对象的事件与事件过程

事件(Event)就是对象上所发生的事情。比如一个吹大的气球,用针扎它一下,结果是圆圆的气球变成了一个瘪壳。把气球看成一个对象,那么气球对刺破它的事件响应是放气,对气球松开手事件的响应是升空。

在 VB 中,事件是预先定义好的、能够被对象识别的动作,如单击(Click)事件、双击(Db1Click)事件、装载(Load)事件、鼠标移动(MouseMove)事件等,不同的对象能够识别不同的事件。当事件发生时,VB 将检测两条信息,即发生的是哪种事件和哪个对象接收了事件。

每种对象能识别一组预先定义好的事件,但并非每一种事件都会产生结果,因为 VB 只是识别事件的发生。为了使对象能够对某一事件做出响应,就必须编写事件过程。

事件过程是一段独立的程序代码,它在对象检测到某个特定事件时执行(响应该事件)。一个对象可以响应一个或多个事件,因此可以使用一个和多个事件过程对用户或系统的事件作出响应。程序员只需编写必须响应的事件过程,而其他无用的事件过程则不必编写,例如在下面的例 2-1 中,窗体 Form1 可以识别许多事件,但我们只编写了“单击”(Click)事件的事件过程,因此,程序运行时,它只对“单击”(Click)事件作出响应。

事件过程的名字是由对象名和事件名两部分组成的,两者之间用下划线连接,其一般格式为:

```
Private Sub 对象名_事件名()  
.....  
End Sub
```

其中,Sub 是定义过程开始的语句,End Sub 是定义过程结束的语句,关键字 Private 表示该过程是局部过程。具体编程时,用户可以在过程开始语句和结束语句之间添加实现具体功能的程序代码。在下一节的例 2-1 中,正是在单击事件过程中添加了在窗体上输出字符串的功能语句:“Print "我喜欢 VB 编程!"”。

VB 采用的是面向对象的事件驱动编程机制,其基本思路是:当一个程序开始运行后,若没有发生任何事件,程序就处于等待状态;只有发生了某个可以识别的事件,程序才会运行所对应的事件过程。



### 2.1.4 对象的方法

VB 中的对象除拥有自己的属性和事件外，还拥有属于自己的方法（Method）。所谓“方法”实际上是 VB 为程序设计人员提供了一种特殊的过程和函数，用来完成一定的操作或实现一定的功能。在 VB 中已将一些通用的过程和函数编写好并封装起来，作为方法供用户直接调用，这给用户带来了很大的方便。例如 Print 就是一种方法，是用来输出信息的专用过程。

“方法”是面向对象的，所以调用“方法”的形式与调用一般的过程或函数不同，应该指明是哪个对象调用的。其调用格式如下：

对象名.方法名[可选参数项]

例如，Form1.Hide 表示由 Form1 对象调用 Hide 这个方法，其执行结果是将窗体 Form1 隐藏起来。同样，“Form1.Print "我的第一个 VB 程序!"”的作用是将字符串“我的第一个 VB 程序!”显示在窗体 Form1 内，其中“我的第一个 VB 程序!”就是可选参数。如写成“Printer1.Print "我的第一个 VB 程序!””则是要在打印机上打印出该字符串。

在上面的调用格式中，如果省略了对象名，则隐含指当前对象。如在例 2-1 的 Form\_Click 事件过程中是直接使用：

```
Print "我喜欢 VB 编程!"
```

调用 Print 方法的。在这里就隐含表示要在窗体 Form1 上输出该字符串。为了清晰和不致混淆，最好都在“方法名”之前加上“对象名”。另外，还应注意每一种对象所能调用的“方法”是不完全相同的。

在代码编写窗口中，如果输入窗体上已有对象名的名称后，再输入一个“.”，VB 就会智能地显示出这个对象所有的属性名和方法名。属性名前面有一个  图标，而方法名的图标是 。

## 2.2 窗体

窗体（Form）也就是平时所说的窗口。它既是 Visual Basic 编程中最常见的对象，又是所有其他控件的容器，各种控件对象必须建立在窗体上。也就是说，窗体既是一个对象，也是其他对象的容器。窗体就像一个可调整大小的绘图板，能够以“所见即所得”的方式，利用控件工具在其上面直观地创建各种对象，形成美观的用户界面。每个应用程序一般至少有一个窗体。在一个 VB 工程中也可以添加多个窗体。窗体有自己的属性、事件和方法，决定着窗体的外观和行为。

### 2.2.1 窗体的属性

窗体的属性决定了窗体的外观和内在结构特征，其中一些属性是其他控件也具有有的。常用的窗体属性如表 2-1 所示。表 2-1 中只罗列了窗体的部分常用属性，其他属性将在具体应用程序的编写过程中逐步介绍。

对于初学者来说一下子记住这些属性是有一定困难的，要熟悉并应用这些属性，最好的办法是上机实践。在“属性”窗口中更改窗体的一些属性，然后运行该应用程序并观察修改的效果，从而体会并记住这些属性的用途。



表 2-1 窗体的常用属性

属性	用途
名称 (Name)	决定窗体的名称, 在程序代码中通过该属性引用、操作对象
Caption	返回/设置窗体标题栏显示的文本
BackColor	返回/设置窗体的背景颜色
ForeColor	返回/设置窗体的前景色
BorderStyle	返回/设置窗体的边框风格
ControlBox	用于确定窗体是否具有控制菜单
Enabled	用于确定窗体是否对用户事件作出响应
Height	返回/设置窗体的高度尺寸
Width	返回/设置窗体的宽度尺寸
Left	返回/设置窗体距屏幕左边的距离
Top	返回/设置窗体距屏幕顶部的距离
MaxButton	决定窗体是否具有最大化按钮
MinButton	决定窗体是否具有最小化按钮
Moveable	决定程序运行时窗体是否能够移动
StartPosition	指定窗体初次出现的位置

### 2.2.2 窗体的事件

VB 采用了事件驱动的编程机制。当没有事件发生时, 程序处于等待状态, 只有事件发生时, 程序才会运行。窗体的事件较多, 根据事件的来源, VB 中的事件可分为鼠标事件、键盘事件和系统事件三种。本节只介绍几个常用的事件, 其余的将在第 7 章再介绍。

最常用的鼠标事件有 Click 事件和 DblClick 事件。Click 事件是程序运行时, 在窗体上单击鼠标左键发生的事件; 而 DblClick 则是在窗体上双击鼠标所发生的事件。

最常用的系统事件有 Initialize 事件、Load 事件与 Unload 事件、Activate 事件与 Deactivate 事件。当窗体第一次创建时会触发 Initialize 事件, 一般将窗体的初始化代码存放在该事件过程中。当一个窗体被装载时触发 Load 事件, 运行程序后, 如果关闭窗口, 就会触发 Unload 事件。当一个窗体变为活动窗体时, 就会触发 Activate 事件。而当另一个窗体或应用程序被激活时, 原窗体不再是活动窗体时会触发 Deactivate 事件。

请注意窗体的 Initialize、Load、Activate 三个事件触发的先后顺序: 先触发 Initialize 事件完成窗体的初始化, 该事件是运行程序创建一个窗体时触发, 运行一次就触发一次, 接着执行 Load 事件, 完成窗体、建立在窗体上的控件对象及变量的初始化, 并显示窗体, 最后触发 Activate 事件。有些操作在三个事件中都能执行, 如修改窗体的标题; 有些操作只能在特定的事件中完成。如 Print 方法, 第一个事件中不能执行, 在第二个事件中执行但不显示结果 (需要设置窗体的 AutoRedraw 为真值), 在第三个事件中可直接执行。

对初学者来说, 可以先掌握基本的 Load()、Click()、DblClick() 事件的应用, 其他事件在以后需要的时候再理解掌握。



如果我们针对某个事件编写了一段程序（事件过程），那么发生某个事件后，该程序段就会被执行；如果发生了某个事件，但没有相应的事件过程，则什么也不会发生。这就是所谓的事件驱动的编程机制。

### 2.2.3 窗体常用的方法

前面提到，所谓方法实际上是 VB 提供的特殊过程和函数。在窗体操作中常用的方法有 Print 方法、Cls 方法、Show 方法、Hide 方法和 Move 方法等。

(1) Print 方法。使用 Print 方法可以在窗体、图片框或打印机等对象中输出文本字符串或表达式的值。这里先用该方法在窗体上输出文字，其详细的使用格式将在第 3 章中再做介绍。

(2) Cls 方法。此方法用来清除由 Print 方法在窗体或图片框中显示的文本或用作图方法在窗体或图片框中显示的图形。Cls 方法的使用格式是：

[对象名.] Cls

对象指窗体或图片框，默认为窗体。

(3) Show 方法和 Hide 方法。这两个方法用于显示或隐藏窗体。

Show 方法用于显示指定的窗体，如果指定的窗体没有装载，VB 将自动装载该窗体。其一般的调用格式如下：

[对象名.] Show [Style]

若 Show 方法前面没有指明对象，隐含指当前窗体。参数 Style 是一个可选的整数，它决定显示的窗体是有模式还是无模式的。若 Style 为 1，则窗体显示是有模式的。此时，Show 后面的代码暂停执行，直到该窗体被隐藏或卸载时才执行。如果 Style 为 0，则窗体显示为无模式的，Show 后面的代码被紧接着执行。

Hide 方法用于隐藏指定的窗体，但不能将其卸载。如果调用 Hide 方法时窗体还没有加载，Hide 方法将加载该窗体但不显示它。Hide 方法的一般格式为：

[对象名.] Hide

若 Hide 方法前面没有指明对象，默认指当前窗体。隐藏窗体时，Hide 方法将指定窗体的 Visible 属性设置为 False，使窗体不可见，但并没有将其卸载。

### 2.2.4 添加窗体与设置启动窗体

通过前面章节的学习我们知道，在建立新工程时系统会自动创建一个窗体。但除了简单的练习题外，一般实际使用的应用程序通常需要使用多个窗体。因此，在当前工程中添加新窗体是应用程序开发过程中经常需要做的工作。

添加新窗体的步骤如下：

(1) 从“工程”菜单中选择“添加窗体”菜单项。

(2) 默认情况下系统将显示如图 2-1 所示的“添加窗体”对话框。

(3) 该对话框的“新建”选项卡用于创建一个新窗体，列表框中列出了各种新窗体的类型，其中选择“窗体”选项时，建立一个空白的窗体，选择其他选项时则建立一个预定义了某些功能的窗体。

(4) 单击“打开”按钮，一个新的空白窗体被加入到当前工程中，同时会显示在屏幕上。



图 2-1 “添加窗体”对话框

添加了新窗体后，它的大小、背景颜色、标题及窗体名称等特征可根据应用程序的要求在属性窗口中设置完成，也可以在程序运行时由程序代码进行设置。

在多窗体情况下，如果事先没有特别的设定，应用程序的第一个窗体被默认为启动窗体，也就是当应用程序开始运行时，先运行该窗体。如果要改变系统默认的启动窗体，需要通过“工程属性”对话框的设置进行调整。例如，如果希望程序启动后首先出现的是 Form2，就应该按下述步骤对工程属性进行设置：

- (1) 从“工程”菜单中选取“工程属性”命令，出现如图 2-2 所示的“工程属性”对话框。



图 2-2 “工程属性”对话框

- (2) 选取该对话框中的“通用”选项卡。
- (3) 在“启动对象”下拉列表框中，选取作为新启动窗体的 Form2。
- (4) 单击“确定”按钮，完成设置。

有时，也许要求应用程序在启动时不加载任何窗体。在这种情况下，可以在标准模块中创建一个名为 Main 的子过程。然后将 Sub Main 过程设为启动对象。可从“工程”菜单中选取“工程属性”命令，然后在出现的对话框中选取“通用”选项卡，再从“启动对象”下拉列表选定 Sub Main。

下面让我们编写两个简单的窗体应用程序。

**例 2-1** 设计一个程序，程序进入运行状态后，自动将窗体的大小设置为屏幕大小的一半并使窗体居中显示，并且在窗体上装载一幅图片。单击窗体时窗体上的图片消失，窗体的背景色会随机发生变化。



进入 VB 编程环境后，根据题目要求在代码窗口中编写窗体的 Load 事件过程和窗体的 Click 事件过程，如下所示：

```
Private Sub Form_Load()
    Form1.Width = Screen.Width * 0.5           '设置窗体的宽度
    Form1.Height = Screen.Height * 0.5        '设置窗体的高度
    Form1.Left = (Screen.Width - Width) / 2    '在水平方向上居中显示
    Form1.Top = (Screen.Height - Height) / 2   '在垂直方向上居中显示
    Form1.Picture = LoadPicture("c:\windows\clouds.bmp") '装载图片
End Sub

Private Sub Form_Click()
    Form1.Picture = LoadPicture("")           '卸载图片
    Form1.BackColor = QBColor(Rnd * 15)      '随机改变窗体背景颜色
End Sub
```

Load 是窗体能够响应的一个事件，只要启动应用程序，窗体被装入内存，就会触发 Load 事件。Load 事件过程通常用来对对象的属性和变量进行初始化。

程序中 LoadPicture 是 VB 的一个函数，用于将指定的图片文件调入内存。LoadPicture 的一般使用格式是：

```
[对象].Picture = LoadPicture ("文件名")
```

对象是指窗体、图片框、图像框等，默认为窗体。括号中双引号里的内容是图形文件名（一般应写完整路径）。本例中是指定装载 C 盘 Windows 目录中的 clouds.bmp 图形文件。如果双引号中为空，则表示对象不装载图形。

窗体的 Picture 属性决定窗体的背景图形，该属性也可以在属性窗口中设置。VB 支持以下几种图像文件格式：位图文件 (\*.bmp)、GIF 压缩位图文件 (\*.gif)、JPEG 压缩位图文件 (\*.jpg)、图元文件 (\*.wmf) 和图标文件 (\*.ico)。

要想改变窗体的尺寸，就需要对窗体的 Height 和 Width 这两个属性进行设置。Height 指窗体的高度，Width 指窗体的宽度，单位为 twip（1 英寸约等于 1440 twip）。本例中是把窗体的高度和宽度都设为屏幕窗口尺寸的一半。

在第 1 章中提到在程序代码中设置对象的属性时，应指明是哪一个对象的属性。在程序中引用某个对象的属性时也应和被引用属性的前面加上对象名。如果不指定对象名（缺省），就是指当前窗体名。

**例 2-2** 设计一个程序，程序包括两个窗体，在第一个窗体上添加一个标签框，内写“进入动物世界”。运行时单击该窗体，显示第二个窗体，窗体上有一动物图片，如果再单击窗体 2，窗体 1 又会显示出来，如图 2-3 所示。



图 2-3 例 2-2 运行结果图



本例题需要设置的对象属性如表 2-2 所示。

表 2-2 例 2-2 对象属性设置

对象	属性名称	设置
窗体 1	名称	Form1
	Caption	窗体 1
窗体 2	名称	Form2
	Caption	窗体 2
标签	名称	Label1
	Caption	进入动物世界
	Font	隶书
	FontSize	小二号
	BackStyle	Transparent

新建立一个工程，系统会自动建立一个窗体（Form1），在 Form1 上添加一标签（Label）控件，根据前面所讲述的在工程中添加新窗体的方法，在本工程中再添加一个窗体（Form2）。然后根据表 2-2 设置对象的属性。到此我们应注意到，本应用程序已包含了两个窗体文件（Form1.frm、Form2.frm）。

实际上在 VB 中，一个应用程序中可以包含构成这个应用程序的所有窗体文件（\*.frm）、标准模块文件（\*.bas）、类模块文件（\*.cls）、包含 ActiveX 控件的文件（\*.ocx）等多个文件。这一系列文件的集合就是工程文件。当工程中所有部分的设计都完成后，可以把工程文件编译为可执行文件。有关生成可执行文件的问题已在第 1 章中介绍，此处不再赘述。

现在本程序有了两个窗体，运行时如何进行切换呢？请看下面的程序代码。

（1）程序运行时，用户单击窗体 1，就触发了 Form1 的 Form\_Click 事件，系统就会执行下列程序代码：

```
Private Sub Form_Click()
    Form1.Hide      '将窗体 1 隐藏
    Form2.Show      '显示窗体 2
End Sub
```

Hide 和 Show 是在多窗体程序中使用的两个方法。Hide 方法的作用是将指定的窗体隐藏起来，使其不可见，即不在屏幕上显示。Show 方法是将指定的窗体显示在屏幕上。

（2）显示 Form2 时，系统会运行下述窗体装载事件过程：

```
Private Sub Form_Load()
    Form2.Picture = LoadPicture("d:\office97\clipart\popular\lion.wmf")
End Sub
```

该过程把指定的图片文件显示在 Form2 上。

（3）单击第二个窗体，则应再显示 Form1，隐藏 Form2，在 Form2 的代码窗口中编写如下过程便可以实现这一要求：

```
Private Sub Form_Click()
    Form1.Show      '显示窗体 1
    Form2.Hide      '隐藏窗体 2
End Sub
```



End Sub

从表面上看，这个过程和前面单击 Form1 的过程名字都是 Form\_Click，运行时会不会发生混乱呢？不会的，因为过程代码是从属于窗体的，在 Form1 中的代码窗口编写的前一个 Form\_Click 过程，它只影响对 Form1 的单击事件；而在 Form2 中的代码窗口编写的另一个 Form\_Click 过程，它只影响对 Form2 的单击事件，两者不会发生混淆。

## 2.3 标签、文本框和命令按钮

标签、文本框和命令按钮是 VB 编程中使用最频繁的三个基本控件。

### 2.3.1 标签

标签（Label）控件是用来显示文本的控件，该控件和文本框控件都是专门对文本进行处理的控件，但标签控件没有文本输入的功能。

标签控件在界面设计中的用途十分广泛，它主要用来标注和显示提示信息，通常是标识那些本身不具有标题属性的控件。例如，可用 Label 控件为文本框、列表框、组合框的控件添加描述性的文字。另外希望在窗体某个位置上输出处理结果、事件进程等信息，Label 比 Print 方法要更容易控制位置。标签控件的主要属性如表 2-3 所示。

表 2-3 标签控件的主要属性

属性	说明
名称 (Name)	设置标签名字，供程序使用
AutoSize	设置标签的大小是否能自动调整以完整显示文本内容，False 为不能调整
BorderStyle	设置标签有无边框，0 - 无边框（默认值）；1 - 有边框
Caption	设置标签上要显示的文本内容
Left	设置标签距窗体左边界的距离
Top	设置标签距窗体上边界的距离
WordWrap	设置标签中所显示的文本是否能自动折行

标签最常用的属性是 Caption，常用来改变 Label 控件中显示的文本。Caption 属性允许文本的长度最多为 1024 个字符。默认情况下，当文本超过控件宽度时，文本会自动换行，但是如果 Caption 内容为西文字符，当文本超过控件宽度时，文本不会自动换行。AutoSize 属性用来设置标签是否会随标题内容的多少自动调整大小，Font 属性用来设置文字字体和大小，ForeColor 属性用来设置标签中的文字颜色，而 BackColor 属性用来设置背景颜色。

标签的主要事件有 Click、DbClick 和 Change 等，但一般标签都是用于标注界面或输出信息，不需编写事件过程。

### 2.3.2 文本框

文本框（TextBox）是一种通用控件，用于显示用户输入的信息，作为接受用户输入数据的接口；或者在设计或运行时，通过对文本框的 Text 属性赋值，作为信息输出的对象。显然



文本框既可输入信息，又可输出信息。在程序设计或运行期间都可以在文本框中输入、编辑、修改和显示文本。

控制文本框外观的属性大部分与窗体和标签相同，文本框所特有的一些主要属性见表 2-4。这里要强调的是文本框没有 Caption 属性，这一点与很多常用的控件不同。

表 2-4 文本框的部分属性

属性	用途
名称 (Name)	决定文本框的名称，在程序代码中通过该属性引用、操作文本框
Text	文本框中的文本内容
MultiLine	该属性值为 True 时，可以接收多行文本
ScrollBars	0—没有滚动条，1—水平滚动条，2—垂直滚动条，3—同时有水平和垂直滚动条
PasswordChar	指定显示在文本框中的替代字符，主要用于密码的输入
MaxLength	指定显示在文本框中的最大字符长度，超出部分不接收
Locked	决定文本框中的文本是否可以编辑，默认为 False，可以编辑
SelStart	定义在文本框中选择文本的起始位置
SelLength	定义选中字符的个数
SelText	当前所选择的字符串内容

文本框的主要事件有：Change 事件、KeyPress 事件、GotFocus 事件和 LostFocus 事件等。

当用户在文本框中输入新的信息或在程序运行时将文本框的 Text 属性设置为新值时，触发 Change 事件。用户每向文本框输入一个字符就引发一次该事件。因此，Change 事件常用于对输入字符类型的实时检测。

当进行文本输入时，每一次键盘输入，都将使文本框接受一个 ASCII 码字符，发生一次 KeyPress 事件，并返回所输入字符的 ASCII 码，因此通过该事件判断用户按了什么键。

一个处于可以接收输入数据状态的对象，被称为具有焦点。当焦点进入文本框时，触发 GotFocus 事件。

当焦点从文本框中离开的时候，触发 LostFocus 事件。通常，可用该事件来检查文本框中用户输入的内容或指定文本框失去焦点后所做事情。

文本框最常用的方法是 SetFocus，该方法是使文本框获得焦点。若希望运行程序就使插入点光标放在窗体上的 Text1 中，可以在窗体的 Activate 事件中添加代码 Text1.SetFocus，但要注意该语句不能出现在窗体的 Load 事件过程中。

### 2.3.3 命令按钮

在 Visual Basic 应用程序中，命令按钮 (CommandButton) 是使用最多的控件对象之一。常常用它来接收用户的操作信息，激发某些事件，实现下一个启动、中断和结束等操作。

命令按钮常用的基本属性有 Name、Caption、Default、Cancel、Enabled、ToolTipText 等。

Name 属性决定命令按钮的名称，在程序代码中通过该属性引用、操作命令按钮对象。

Caption 属性用于设置命令按钮上显示的文本。它既可以在属性窗口中设置，也可以在程



序运行时设置。在运行时设置 `Caption` 属性将动态更新按钮文本。`Caption` 属性最多包含 255 个字符。若标题超过了命令按钮的宽度，文本将会换行。如果内容超过 255 个字符，则标题超出部分将被截去。

`Default` 属性和 `Cancel` 属性的作用分别是：指定一个默认命令按钮，应将其 `Default` 属性设为 `True`，则不管窗体上的哪一个控件有焦点，只要用户按 `Enter` 键，就相当于单击此默认命令按钮。同样，通过 `Cancel` 属性可指定默认的取消按钮。在把命令按钮的 `Cancel` 属性设置为 `True` 后，不管窗体当前哪个控件有焦点，按 `Esc` 键，就相当于单击此默认按钮。显然，一个窗体只能有一个命令按钮的 `Default` 属性设置为 `True`，也只能有一个命令按钮的 `Cancel` 属性设置为 `True`。

`ToolTipText` 属性的作用是以较少的文字解释命令按钮的功能。

命令按钮最重要的事件是单击事件（`Click`），单击命令按钮时将触发单击事件，并调用和执行已写入 `Click` 事件中的代码。多数情况下，主要是针对该事件过程来编写代码。需要注意的是，命令按钮不支持双击（`DoubleClick`）事件。

下面编写几个使用上述基本控件的简单程序。

**例 2-3** 设计一个程序，用户界面由三个命令按钮、一个标签和一个文本框组成。当用户单击其中一个命令按钮，在文本框上显示文本内容，单击另一个命令按钮，清除文本框内容，单击第三个命令按钮，结束程序的运行。

首先进入 VB 编程环境，建立一个新工程，再将一个标签、一个文本框和三个命令按钮添加到窗体 `Form1` 上，在窗体上添加控件的方法是：

把鼠标指针移到工具箱中所需控件图标上，单击鼠标左键，所选择的控件图标即改为浅灰色，然后松开鼠标左键将光标移到窗体上，这时光标由箭头变成十字。将十字光标移到窗体希望的位置上，按住鼠标左键作斜线运动，拖拉成所需要的尺寸，然后释放左键，一个控件就加到窗体上了。

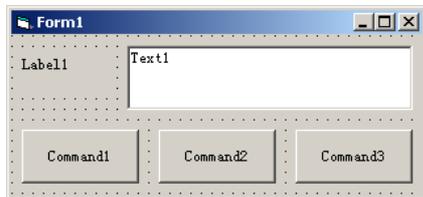


图 2-4 程序界面设计

在窗体上添加了上述几个控件后，调整控件的大小和位置如图 2-4 所示之后，就可以开始设置窗体和各控件的属性了。本例题中需要设置的属性如表 2-5 所示。

表 2-5 例 2-3 对象属性设置

对象	属性	设置
窗体	(名称)	<code>Form1</code>
	<code>Caption</code>	例题 2-3
标签	(名称)	<code>Label1</code>
	<code>Caption</code>	文字显示
	<code>FontName</code> <code>FontSize</code>	宋体 小四号
文本框	(名称)	<code>Text1</code>
	<code>Text</code>	置空
	<code>FontName</code> <code>FontSize</code>	隶书 四号



续表

对象	属性	设置
命令按钮 1	(名称)	cmdShow
	Caption	显示
	FontSize	五号
命令按钮 2	(名称)	cmdClear
	Caption	清除
	FontSize	五号
命令按钮 3	(名称)	cmdEnd
	Caption	退出
	FontSize	五号

在本例中，窗体的 Caption（标题）属性值设置为“例题 2-3”，名称属性使用了系统给定的默认值，故不需要重新设置。

设置控件的属性时，首先单击窗体上的某一控件，使其成为当前控件（请注意：当前控件的四周有 8 个蓝色小方块）。然后在属性窗口中找到需要设置的属性，再指定属性值。例如，要把命令按钮 Command1 的 Caption（标题）属性设为“显示”，就应该先单击命令按钮 Command1，使其成为当前控件，再到属性窗口中找到 Caption 属性栏。可以看到系统事先为命令按钮 Command1 设置的属性值（称为默认值）是“Command1”。单击此行，可以看到此行变为醒目（蓝色）显示。为了改变系统给定的标题，在这里可以删除字符 Command1 并用汉字重新输入“显示”，这时可以看到命令按钮 1 上的文字已由 Command1 改变为“显示”，该命令按钮的 Caption 属性就设置好了。从属性表中找到“名称”属性，单击名称属性，其属性值为 Command1（系统默认值），将其删除，修改为 cmdShow。

为了改变命令按钮上显示文字的大小，可以设置命令按钮的 Font 属性。在属性窗口中找到 Font 属性并单击它，右侧会出现一个有三点（...）的按钮。单击该按钮，会打开一个字体设置对话框（见图 2-5），其中包括有“字体”、“字体样式”、“大小”等。在这里可以把“大小”中的“小五”（系统默认值）改为“五号”，然后用鼠标单击图 2-5 中的“确定”按钮，退出此对话框。



图 2-5 字体设置对话框

用同样的方法按表 2-5 中的要求设置其余两个命令按钮的有关属性。

文本框可以用来显示文字信息。文本框的 Text 属性的值就是文本框中所显示的内容。单



击窗体上的文本框，使其成为当前控件，从属性窗口中找到 **Text** 属性，单击名称属性，其属性值为 **Text1**（系统默认值），将其删除，即清空文本框，使文本框中不显示任何信息。最后将文本框的字体大小属性设置为“四号”，字体为“隶书”。

标签的作用主要是用于一些文字标注，这里把标签的 **Caption** 属性设置为“文字显示”，字体设置为宋体，字号设为小四号。

在这里应注意到属性“标题”（**Caption**）和“名称”的不同作用。“标题”的内容显示在对象上，用户可以在屏幕上看到它，以便识别不同的对象。而“名称”不显示在对象上，它是供程序识别的。例如，本例中第一个命令按钮的 **Caption** 属性值是“显示”，它显示在窗体中的命令按钮上。该命令按钮的“名称”属性值是 **cmdShow**，它在屏幕上是不看到的，只供程序识别，在后面将会看到 **cmdShow** 这个名字将会出现在程序代码中。另外，还应注意文本框控件没有“标题”（**Caption**）属性，文本框中显示的内容是它的 **Text** 属性值。设置完成后的窗体界面如图 2-6 所示。

属性设置完成以后就可以编写事件过程代码了。过程代码也称程序代码，它是针对某个对象事件编写的。本例题要求单击“显示”命令按钮后，在窗体上的文本框内输出字符串。也就是说，需要对该命令按钮的单击事件编写一段代码，以指定用户单击该按钮时要执行的操作。为了编写程序代码，必须使屏幕显示出代码窗口。

可以双击该命令按钮进入代码窗口，也可以按 F7 键进入代码窗口，此时代码窗口就会出现：



图 2-6 设置完成后的窗体界面

```
Private Sub cmdShow_Click()
```

```
End Sub
```

表示对名为 **cmdShow** 的对象（即“显示”按钮）的单击事件（**Click**）进行代码设计。根据题意应在上述两行命令之间输入：

```
Text1.Text = "我的第二个 VB 程序！"
```

即：

```
Private Sub cmdShow_Click()
```

```
Text1.Text = "我的第二个 VB 程序！"
```

```
End Sub
```

其中 **Text1** 是文本框的名称的属性值，即文本框的名字。**Text** 是文本框的文本属性，上面赋值语句的目的是修改文本框的 **Text** 属性的值，即把右边字符串赋给文本框的 **Text** 属性，这样在窗体的文本框中就显示出这个字符串了。它的作用和在属性表中为窗体、命令按钮或文本框等对象设置属性值是一样的。也就是说，可以通过两种方法修改对象的属性值。一是设计阶段在属性窗口中进行，例如在上面所做的控件属性值设置。另外是在程序运行过程中用语句来实现，如本例中的“**Text1.Text="我的第二个 VB 程序！"**”。

下面是另一个事件过程。若用户单击“清除”按钮，就使文本框清空：

```
Private Sub cmdClear_Click()
```

```
Text1.Text= ""
```

```
End Sub
```

要想清除文本框，将 **Text** 属性置空即可，其实在前面的属性窗口中为文本框设置属性时已经实现过清空文本框的功能，这里只是通过程序语句来实现罢了。



“退出”命令按钮的事件过程如下：

```
Private Sub cmdEnd_Click()
    End
End Sub
```

其中的 End 命令将结束程序的运行。

程序代码编写完成后就可以运行调试程序了。选择菜单“运行”中的“启动”命令或按 F5 键，程序开始运行。运行时单击“显示”命令按钮后的结果见图 2-7。若再单击“清除”按钮，文本框中无任何文字显示。单击“退出”按钮，程序就会结束运行。



图 2-7 程序运行界面

**例 2-4** 编写一程序，由用户从键盘上输入两个数，然后求这两个数的和，并将结果显示出来，要求的用户界面如图 2-8 所示。

解题步骤如下：

(1) 首先进入 VB 编程环境，建立一个新工程，再将三个标签、三个文本框和一个命令按钮加到窗体 Form1 上，调整控件的大小和位置如图 2-9 所示。



图 2-8 例 2-4 运行界面

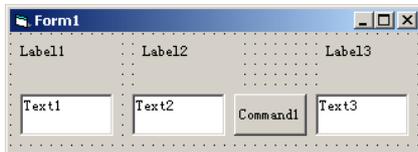


图 2-9 例 2-4 初始设计

(2) 按表 2-6 设置有关属性。

表 2-6 例 2-4 属性设置

对象名称	属性	属性值	说明
Form1	Caption	例题 2-4	窗体标题
Text1	Text		程序启动时文本框内无文字
Text2	Text		程序启动时文本框内无文字
Text3	Text		程序启动时文本框内无文字
Label1	Caption	被加数	标签框标题
Label2	Caption	加数	标签框标题
Label3	Caption	和	标签框标题
Command1	Caption	=	命令按钮标题

另外，为了醒目，可以把窗体上所有控件的 FontSize 属性设置为“小四”。

(3) 编写命令按钮 Command1 的 Click 事件过程代码：



```
Private Sub Command1_Click()
    Text3.Text = Val(Text1.Text) + Val(Text2.Text)
End Sub
```

代码“Text3.Text=Val(Text1.Text)+Val(Text2.Text)”的作用是把文本框 1 和文本框 2 中的数据相加，并将结果放在文本框 3 中。Val()是 VB 内部数据转换函数，作用是把字符串转换为数值。有关内部函数的具体使用方法将在第 3 章介绍。

程序运行时，在文本框 1 和文本框 2 中输入数字后单击命令按钮，文本框 3 中就会显示出计算结果（如图 2-8 所示）。

通过上面的例子可以看出，文本框既可以用于用户输入，也可以用于程序输出；文本框和标签结合使用，可以使应用程序界面更加醒目、清楚。

**例 2-5** 设计一个程序，界面如图 2-10 所示，单击“复制”按钮，将上面文本框中选定的文本复制到下面文本框中，单击“移动”按钮可将上面文本框中选定的文本移动到下面文本框中，单击“清空”按钮将两个文本框中的字符清除。

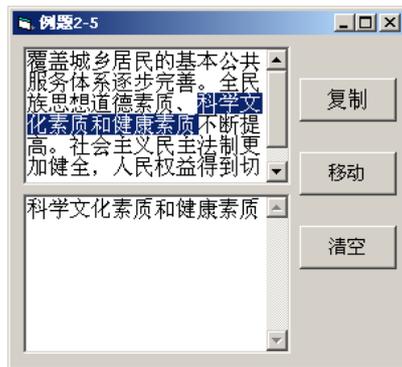


图 2-10 例 2-5 运行界面

解题步骤如下：

(1) 新建一个工程，在窗体上添加两个文本框和三个命令按钮，并调整控件的位置和大小。

(2) 通过属性窗口按表 2-7 设置对象属性。

表 2-7 例 2-5 对象属性设置

对象名称	属性	属性值	说明
Form1	Caption	例题 2-5	窗体标题
Text1	Text		程序启动时文本框内无文字
Text2	Text		程序启动时文本框内无文字
Command1	Caption	复制	命令按钮标题
Command2	Caption	移动	命令按钮标题
Command3	Caption	清空	命令按钮标题

另外，为使文本框内容浏览时方便，把两个文本框的 ScrollBars 属性设置为 2，添加垂直滚动条；把两个文本框的 MultiLine 属性设置为 True，使文本框中的内容能自动换行。



(3) 为三个命令按钮添加代码。单击“复制”按钮将第一个文本框中选定的内容复制到第二个文本框中,就是将第一个文本框中选定的内容连接到第二个文本框中内容的后面,选定内容用其 SelText 属性表示。因此需要在“复制”按钮的 Click()事件中添加代码:

```
Private Sub Command1_Click()
    Text2.Text = Text2.Text + Text1.SelText
End Sub
```

单击“移动”按钮将第一个文本框中选定的内容移动到第二个文本框中,就是将第一个文本框中选定内容连接到第二个文本框内容的后面,并将第一个文本框选定的内容清除,将选定内容清除,就是置 SelText 为空串,即 SelText="",因此需要在“移动”按钮的 Click()事件中添加代码:

```
Private Sub Command2_Click()
    Text2.Text = Text2.Text + Text1.SelText
    Text1.SelText = ""
End Sub
```

单击“清空”按钮,将两个文本框中的字符清除,需要在“清空”按钮的 Click()事件中添加代码:

```
Private Sub Command3_Click()
    Text1.Text = ""
    Text2.Text = ""
End Sub
```

程序运行时,在文本框 1 中输入一段文字,用鼠标选定任意部分文字后单击“复制”按钮,所选的文字就被复制到文本框 2 中;若单击“移动”按钮,选定的文字就会被移动到文本框 2 中;单击“清空”按钮,两个文本框中的文字都被清除。

## 2.4 程序调试

编写程序时,难免会发生错误。查找和修改错误,使得程序能够正常运行的过程称为程序调试。VB 为调试程序提供了一组有效的调试工具。

### 2.4.1 错误类型

可以将程序错误划分为编译错误、运行错误和逻辑错误三大类。

#### 1. 编译错误

在程序代码窗口编写程序时,系统会对输入的程序语句的语法进行自动检查。如果发现语法错误,系统会弹出一个对话框,提示错误信息,如图 2-11 所示。这时,必须单击“确定”按钮,关闭错误提示对话框。出错的那一行语句变为红色,提示进行必要的修改。例如,在图 2-11 所示的程序语句中,少写了一个右括号就回车换行,系统就会提示错误信息。添加右括号后,该语句文字又变为正常的黑色。

另外,有些编译错误要等到系统对程序进行运行前的



图 2-11 编译错误提示



编译时才能发现，如变量未定义、遗漏关键字等。

### 2. 运行错误

运行错误是指程序在编译通过后，在运行过程中发生的错误。这类错误往往是由于程序代码执行了一些非法操作引起的。比如类型不匹配、要求的对象不存在、试图打开一个不存在的磁盘文件等。

例如，窗体中没有添加控件 Text3，而程序中却要对 Text3 的某个属性赋值，程序运行时，就会给出运行错误提示，如图 2-12 所示。单击“调试”按钮，进入中断模式，出现错误的语句高亮显示（见图 2-13）。此时可以对代码进行修改。



图 2-12 运行错误提示

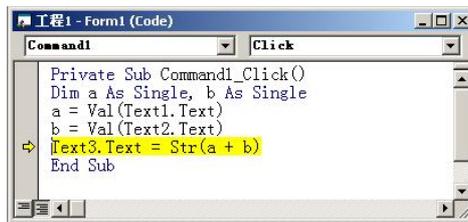


图 2-13 中断模式

### 3. 逻辑错误

程序没有编译或运行错误，但运行后得不到正确的运行结果，这说明程序存在逻辑错误。例如，运算符使用不正确，语句次序不对，算法错误等。通常，逻辑错误没有错误提示信息，只能通过仔细阅读和分析程序来查找错误。

## 2.4.2 调试与排错

为了改正程序中发生的不同错误，VB 提供了各种调试工具。主要是通过设置断点，插入观察变量、逐行执行和过程跟踪等方法，在调试窗口中显示所关注的信息，让程序员发现和排除错误。

### 1. VB 的三种模式

作为一个集编辑、编译和运行于一体的程序集成开发环境，VB 的工作状态可分为三种模式，即设计模式、运行模式和中断模式。在设计模式下可以进行程序界面的设计、对象的属性设置以及程序代码的编写等，此时，窗口的标题栏显示“设计”。在设计模式下不能运行程序，也不能使用调试工具。执行“启动”命令后，系统由设计模式进入运行模式，标题栏显示“运行”。运行模式下，可以查看程序代码，但不可以修改程序。若要修改程序，可以执行“结束”命令，回到设计模式。也可以执行“中断”命令，进入中断模式。中断模式中，程序暂时停止运行，可以查看、修改程序代码和检查数据。修改结束后，再单击“继续”按钮，程序继续执行。

### 2. 插入断点和逐句跟踪

在程序调试时，可以通过人为地设置断点来中断程序的运行，然后逐句跟踪检查相关变量、属性和表达式的值是否在预期的范围之内。

可以在中断模式或设计模式状态下设置或删除断点。把光标移到怀疑有问题的语句处，按 F9 键，就在该条语句处设置了断点。程序运行到断点处会自动停下，进入中断模式。断点



以前的变量、属性和表达式值均可以查看。在中断模式下查看某个变量值的方法是，把鼠标指向该变量，稍停一会儿，鼠标下方就会显示出该变量的值。

若要继续跟踪断点以后语句的执行情况，可以按下 F8 键或选择“调试”菜单中的“逐语句”命令。在断点处再按一次 F9 键，就取消了断点。

将设置断点和逐条语句跟踪相结合进行程序调试，是初学者常用的方法。

### 3. 调试窗口

在中断模式中，除了用鼠标指向某变量显示其值外，还可以通过“立即”窗口、“监视”窗口和本地窗口来观察有关变量的值。可以单击“视图”菜单中的相关命令打开这些窗口。

“立即”窗口是程序调试中使用最方便的窗口。可以在程序中利用 Debug.Print 方法，把输出送到立即窗口，也可以直接在立即窗口中使用 Print 语句或“?”来显示变量的值。

“本地”窗口显示的是当前过程中所有变量的值。当程序从一个过程切换到另一个过程时，本地窗口的内容会发生变化，它只反映当前过程中可用的变量值。

“监视”窗口显示的是事先设定的要监视的表达式值。在设计阶段，可以利用“调试”菜单的“添加监视命令”或“快速监视”命令来添加监视表达式以及设置监视类型。程序运行时，“监视”窗口中就会显示相应的数值。



## 本章小结

Visual Basic 采用的是面向对象的事件驱动编程机制。在 Visual Basic 中对象是窗体和控件的总称。每个对象都有一组特征，称之为属性。事件是预先定义好的、能够被对象识别的动作。对象除拥有自己的属性和事件外，还拥有属于自己的方法。方法实际上是 VB 提供的一种特殊的子程序，用来完成一定的操作或实现一定的功能。

窗体本身是对象，同时它又是其他对象的容器。标签、文本框和命令按钮是程序界面设计时最常用的控件对象。

任何一个程序都有可能存在错误，可以将程序错误划分为编译错误、运行错误和逻辑错误三大类。为了改正程序中发生的不同错误，VB 提供了各种调试工具。主要是通过设置断点、插入观察变量、逐行执行和过程跟踪等方法，在调试窗口中显示所关注的信息，让程序员发现和排除错误。



## 习题 2

### 一、填空题

1. Visual Basic 是一种面向\_\_\_\_\_的可视化程序设计语言，采取了\_\_\_\_\_的编程机制。
2. Visual Basic 的对象主要分为\_\_\_\_\_和\_\_\_\_\_两大类。
3. 在 Visual Basic 中，用来描述一个对象特征的量称为对象的\_\_\_\_\_。
4. 在 Visual Basic 中，设置或修改一个对象的属性的方法有两种，分别是\_\_\_\_\_和\_\_\_\_\_。
5. 在 Visual Basic 中，事件过程的名字由\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_所构成。



6. 程序运行时,若用户单击了窗体 Form1,则此时将被执行的事件过程的名字应为\_\_\_\_\_。
7. 窗体的标题栏显示的内容由窗体对象的\_\_\_\_\_属性决定。
8. VB 工程文件的扩展名为\_\_\_\_\_,标准模块文件的扩展名为\_\_\_\_\_,窗体文件的扩展名为\_\_\_\_\_。
9. 在 VB 中,错误的类型大致可分为三种:\_\_\_\_\_,\_\_\_\_\_和\_\_\_\_\_。
10. VB 有三种工作模式,即设计模式、\_\_\_\_\_和\_\_\_\_\_。

## 二、选择题

1. 在 Visual Basic 中最基本的对象是( ),它是应用程序的基石,也是其他控件的容器。  
A. 文本框      B. 命令按钮      C. 窗体      D. 标签
2. 窗体的标题栏显示的内容由窗体对象的( )属性决定。  
A. BackColor      B. BackStyle      C. Text      D. Caption
3. 若要取消窗体的最大化功能,则可设置( )属性为 False 来实现。  
A. ControlBox      B. MinButton      C. MaxButton      D. Enabled
4. 窗体的边框风格由窗体对象的( )属性来设置。  
A. BackStyle      B. BorderStyle      C. WindowState      D. FillStyle
5. 若要以程序代码方式设置在窗体中显示文本的字体大小,则可用窗体对象的( )属性来实现。  
A. FontName      B. Font      C. FontSize      D. FontBold
6. 若要设置文本的显示颜色,则可用( )属性来实现。  
A. BackColor      B. ForeColor      C. FillColor      D. BackStyle
7. 若要使窗体启动时位于屏幕的中间,则可通过( )属性来设置。  
A. Top      B. Left      C. StartUpPosition      D. WindowState
8. 若要禁止窗体被移动,则可设置窗体的( )属性为 False 来实现。  
A. Move      B. Moveable      C. MinButton      D. ControlBox
9. 若要将窗体隐藏起来,则实现的方法为( )。  
A. Hide      B. UnLoad      C. Show      D. WindowState
10. 命令按钮的标题文字由( )属性来设置。  
A. Text      B. Caption      C. Name      D. (名称)
11. 若要将某命令按钮设置为默认命令按钮,则可设置( )属性为 True 来实现。  
A. Value      B. Cancel      C. Default      D. Enabled
12. 若要设置命令按钮的工具提示,则可通过( )属性来设置。  
A. Caption      B. Text      C. Value      D. ToolTipText
13. 若要使某命令按钮获得控制焦点,则可使用( )方法来设置。  
A. Refresh      B. SetFocus      C. GotFocus      D. Value
14. 若要使标签框的大小自动与所显示的文本相适应,则可通过设置( )属性的值为 True 来实现。  
A. AutoSize      B. Alignment      C. Appearance      D. Visible



15. 若要设置或返回文本框中的文本, 则可通过文本框对象的 ( ) 属性来实现。  
 A. Caption      B. Text      C. (名称)      D. Name
16. 若要使文本框成为只读文本框, 可通过设置 ( ) 属性值为 True 来实现。  
 A. ReadOnly      B. Lock      C. Locked      D. Enabled

### 三、编程题

1. 建立一个新工程, 在窗体上添加一个标签框 (Label)。标签框的边框风格属性 (BorderStyle) 值设为 1, 单击窗体时, 在标签框中显示“我喜欢 VB”的字样, 如图 2-14 所示。



图 2-14 第 1 题的运行界面

2. 建立一个新工程, 在窗体上添加一个标签 (Label1)、一个文本框 (Text1) 和两个命令按钮 (Command1、Command2)。运行窗体后, 单击“显示标题” (Command1) 按钮在标签中显示“Visual Basic 编程”字样, 单击“显示内容” (Command2) 按钮在文本框中显示“我的 Visual Basic 程序”字样, 如图 2-15 所示。



图 2-15 第 2 题的运行界面

3. 在窗体中添加两个命令按钮和一个标签, 程序启动时只显示两个命令按钮, 单击“英文”按钮时显示“Good Bye”, 单击“中文”按钮时显示“再见”, 如图 2-16 所示。



图 2-16 第 3 题的运行界面