

第3章

Visual Basic 程序设计初步

设计一个 VB 应用程序主要包括设计用户操作界面和设计程序代码。VB 中所有用户界面都是以窗体为基础。使用 VB 程序设计语言要遵循有关的语法规定。

3.1 习题解答

【习题 3.1】 以下哪些变量名是 Visual Basic 的合法变量名？

`Sin, 6a, a+b, x_y, a%, b$, Print`

【解】

合法的变量名是：`x_y, a%, b$`。

按照 Visual Basic 的语法规定，变量名不能使用系统保留字，`Sin` 和 `Print` 是系统函数或方法，不能作为变量名。变量名的第一个字符必须是字母，`6a` 不符合这条规定。变量名的第二个字符及其后的字符可以是字母、数字及下划线，`a+b` 是表达式，含有系统不允许用作标识符的字符。

【习题 3.2】 VB 中主要的数据类型有哪些？关键字和类型说明符是什么？

【解】

在 VB 中主要有两大类数据类型：一种是基本数据类型，包括数值类型、字符类型、布尔类型、日期类型、变体类型等；一种是用户自定义数据类型。

数据类型及说明符如表 3.1 所示。

表 3.1 数据类型及说明符

数据类型	类型说明符	数据类型	类型说明符
整型(Integer)	%	字符串类型(String)	\$
长整型(Long)	&	字节(Byte)	
单精度实型(Single)	!	布尔型(Boolean)	
双精度实型(Double)	#	日期类型(Date)	无
货币类型(Currency)	@	变体类型(Variant)	

【习题 3.3】 声明变量的命令是什么？如果变量没有被声明，其数据类型是什么？

【解】

根据变量的作用范围，声明变量的方式有不同：

Dim 声明变量；

Public 声明公共变量；

Private 声明私有变量，变量的作用范围只在声明变量的过程或模块内；

Static 声明静态变量。

如果变量没有被声明，则默认其数据类型为变体类型。

【习题 3.4】 把以下表达式写成合法的 VB 表达式。

(1) $[(3x+y)^2 + \cos 45^\circ] \times 8$

(2) $10 + 5 \times (c+d) / (a+b)$

(3) $2e^2 + \sin(30^\circ) \ln x$

【解】

(1) 符合 Visual Basic 语法的表达式为： $((3 * x + y)^2 + \cos(45 * 3.14159/180)) * 8$ 。

VB 中要求三角函数的自变量是一个以弧度为单位的数值表达式，因此需要将表达式的参数转换为弧度。

(2) 符合 Visual Basic 语法的表达式为： $10 + (5 * (c+d)) / (a+b)$ 。

(3) 符合 Visual Basic 语法的表达式为： $2 * \exp(2) + \sin(30 * 3.14159/180) * \ln x$ 。

【习题 3.5】 说明以下语句所定义的数组有多少数组元素。

(1) Dim a1(-1 To 4) As Integer

(2) Dim a2(0 To 1, -1 To 2) As Integer

(3) Dim a3(1 To 3, 0 To 2) As Integer

【解】

(1) a1 是一维数组，有 6 个数组元素。

(2) a2 是二维数组，有 8 个数组元素。

(3) a3 是二维数组，有 9 个数组元素。

【习题 3.6】 窗体的 Name 属性与 Caption 属性有何区别？

【解】

窗体有 Name 属性、Caption 属性。

窗体的 Name 是窗体的名称属性，其作用是使程序能够识别窗体。这个属性值在编程时使用，用户看不到。

窗体的 Caption 属性是窗体的标题。Caption 属性值是显示在窗体标题栏上的文字。

【习题 3.7】 结合实际，举例说明什么是属性、事件、方法。

【解】

属性是对象的特征，例如命令按钮的长度、宽度等是表示命令按钮形状特征的属性值。

事件是由系统预先设置好、能够被识别和响应的动作，例如，单击命令按钮是由系统

设置并可被识别的动作,是命令按钮的事件。

方法是 VB 提供的一种特殊的子程序,用来完成预定的操作,例如在屏幕上显示文字可以用窗体的 Print 方法实现。

【习题 3.8】 当用鼠标单击窗体时,会触发哪些事件?

【解】

当用鼠标单击窗体时,会触发多个事件,包括 Click、MouseUp、MouseDown 等。

【习题 3.9】 设计一个程序,当单击窗体时,在窗体上显示任意一幅图片。

【解】

根据题目要求,单击窗体装入图片,即单击窗体时触发 Form_Click()事件过程,把图片显示在窗体上。该事件过程的代码如下。

```
Private Sub Form_Click()  
    Form1.Picture=LoadPicture("e:\flower.bmp")  
End Sub
```

为了在窗体上装入图片,程序中使用了 LoadPicture()函数。LoadPicture()函数的作用是将指定的图形文件调入内存。这里是将 e 盘中位于根目录下的图形文件 flower. bmp 调入内存,并将此值赋给窗体 Form1 的 Picture 属性。

运行程序,单击窗体,执行 Form_Click()事件过程。运行结果如图 3.1 所示。



图 3.1 运行习题 3.9

3.2 实验导引

本章实验的目的是了解窗体的构成、窗体的事件,了解 VB 的变量、函数、表达式基本的概念,通过练习,能够编写简单的程序,掌握常用函数的使用方法。

按照要求编写程序,并上机调试、运行。只有多练习才能提高程序设计的能力。

【实验题 3.1】 设计一个程序。窗体背景设置为蓝色,窗体的标题为“显示文字”。单击窗体,在窗体上显示一串文字“VisualBasic 程序设计”,文字的颜色为红色。

【解】

窗体上无须添加控件。按照表 3.2 设置控件属性。

表 3.2 实验题 3.1 对象属性设置

控 件	属 性	设 置
窗体	(名称)	Form1
	Caption	显示文字
	BackColor	&H00FF8080&

编写单击窗体的事件过程如下。

```
Private Sub Form_Click()  
    Form1.ForeColor=RGB(255, 0, 0)  
    Print "VisualBasic 程序设计"  
End Sub
```

ForeColor 属性表示窗体的前景色,也就是显示在窗体上的字体颜色,RGB(255,0,0)函数表示红颜色。该赋值语句将表示红颜色的函数值 RGB(255, 0, 0)赋值给 ForeColor 属性。

【实验题 3.2】 设计一个窗体,外观如图 3.2 所示。要求窗体的标题为“显示图片”。窗体上有一个图片框和两个命令按钮,其标题分别是“显示”、“清除”。运行程序,单击“显示”命令按钮,在图片框中显示一幅图片。单击“清除”按钮,则清除图片框中的图片(提示:使用 LoadPicture()函数,图片文件名为空,即“”)。



图 3.2 实验题 3.2 窗体外观

【解】

参照图 3.2 在窗体上添加控件,并按照表 3.3 设置各控件的属性。

表 3.3 实验题 3.2 对象属性设置

控 件	属 性	设 置
窗体	(名称)	Form1
图片框	(名称)	Picture1
命令按钮 1	(名称)	cmdShow
	Caption	显示
命令按钮 2	(名称)	cmdCls
	Caption	清除

分别编写两个命令按钮的单击事件过程,代码如下。

```
Private Sub cmdShow_Click()  
    Picture1.Picture=LoadPicture("e:\flower1.bmp")  
End Sub  
Private Sub cmdCls_Click()  
    Picture1.Picture=LoadPicture("")  
End Sub
```

cmdShow_Click()事件过程中用 LoadPicture()函数将图片文件赋值给 Picture1.Picture,图片显示在图片框中。cmdCls_Click()事件过程中,LoadPicture()函数的参数为空值,其效果是使图片框控件的 Picture 属性值为空,即图片框没有图片。

【实验题 3.3】 表达式练习。

设计一个程序,当单击“计算”命令按钮时,计算以下表达式的值,并将计算结果显示在窗体上。

$$(1) (a * a) + (b + d) / 3$$

$$(2) a > b \text{ And } c = d$$

$$(3) 2e^2 + \sin 30^\circ * \ln b$$

【解】

设计窗体,有两个命令按钮,分别为“计算”和“退出”,如图 3.3 所示。

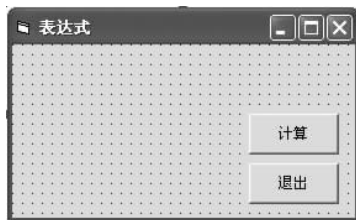


图 3.3 实验题 3.3 窗体外观

设 $a=3, b=5, c=8, d=7$, 事件过程代码如下。

```
Private Sub cmdCala_Click()  
    a=3: b=5: c=8: d=7  
    Print "(1)"; (a * a) + (b + d) / 3  
    Print "(2)"; a > b And c = d  
    Print "(3)"; 2 * Exp(2) + Sin(30 * 3.14159 / 180) * ln b  
End Sub
```

运行程序,变量赋值后,在 Print 语句中计算表达式的值并显示在窗体上。

“退出”命令按钮的单击事件过程如下。

```
Private Sub cmdExit_Click()  
    End  
End Sub
```

【实验题 3.4】 字符串函数使用练习。

自己设计窗体界面及程序,完成如下功能。

设有字符串 VisualBasicProgramming,请使用函数分别取出该字符串的三个子字符串 Visual、Basic、Programming,并将 Visual 的字母都变为大写,Basic 的字母都变为小写。

提要: 本题是使用字符串函数,从一个字符串中取出子字符串的练习。相关函数有: Left、Mid 和 Right。请按照题目的要求自行设计窗体界面。界面设计的基本要求是能显示字符串处理的结果,并有能够触发处理字符串的事件过程。

以下是主要的程序代码段。

```
str0="VisualBasicProgramming"  
str1=UCase(Left(str0, 6))  
str2=LCase(Mid(str0, 7, 5))  
str3=(Right(str0, 11))
```

具体的事件过程、结果显示方式等请自行设计。

【实验题 3.5】 数值函数使用练习。

自己设计窗体界面及程序,完成如下功能:分别使用 Int、Fix、CInt 等取整数的函数将一个实数转换为整数,并总结这几个函数的异同。

提要: Int、Fix、CInt 等函数的作用都是将实数转换为整数,但有一些不同。设计程序,测试这几个函数的差异。需要特别注意的是设计测试数据。考虑函数对四舍五入的

处理,测试数据应包含小于0.5、等于0.5、大于0.5几种情况;考虑正负数的影响,测试数据应包含正实数、负实数。然后根据测试结果总结出这几个函数的异同。

【实验题 3.6】 颜色函数使用练习。

设计一个含有三个命令按钮的窗体,命令按钮的标题分别是“红色”、“蓝色”及“绿色”。单击某个命令按钮,窗体的背景色改变为命令按钮上文字所标的颜色。如单击“红色”命令按钮,窗体的背景色改为红色。

【解】

在窗体上添加三个命令按钮,名称分别是 cmdRed、cmdGreen、cmdBlue。单击某个命令按钮,事件过程中改变窗体的背景色。

“蓝色”按钮的单击事件过程如下。

```
Private Sub cmdBlue_Click()  
    Form1.BackColor=QBColor(9)  
End Sub
```

“绿色”按钮的单击事件过程如下。

```
Private Sub cmdGreen_Click()  
    Form1.BackColor=QBColor(10)  
End Sub
```

“红色”按钮的单击事件过程如下。

```
Private Sub cmdRed_Click()  
    Form1.BackColor=QBColor(12)  
End Sub
```

第4章

常用控件

在设计 VB 程序时,首先需要从工具箱中选择要使用的控件,并添加到窗体上,调整布局,完成界面设计,然后进行程序设计。本章主要掌握命令按钮、文本框、标签等常用控件的特点和使用方法。

4.1 习题解答

【习题 4.1】 文本框、标签控件的功能与特点各是什么?

【解】

文本框控件能够显示、编辑数据。标签控件用于显示数据。两者都能显示数据。但是标签不可以编辑数据。文本框具有标签显示数据的功能,同时具备编辑数据的功能。当把文本框的 Locked 属性设置为 True,文本框的功能和标签一样。

文本框能够获得及失去焦点,即能够触发 GetFocus 和 LostFocus 事件。标签不能触发这两个事件。

【习题 4.2】 分析文本框的 KeyPress 事件。文本框的 Change、KeyPress 事件有何不同?

【解】

(1) 分析文本框的 KeyPress、KeyDown、KeyUp 等三个事件的特点。

在某个控件获得焦点的情况下,按下键盘上的按键时触发该控件的 KeyPress 事件。KeyPress 主要用来获取单个字符(大小写字母、数字键)的 ANSI 码值。

在某个控件获得焦点的情况下,按下键盘上任意键时触发该控件的 KeyDown 事件,在松开按键时触发 KeyUp 事件。KeyDown 和 KeyUP 事件可以响应不被 KeyPress 识别的按键操作,例如,功能键(F1~F12)、编辑键、定位键以及这些键和键盘换档键的组合等。KeyDown 和 KeyUP 事件将大小写字母视为具有相同的 KeyCode,即不能直接判断单个字符的大小写,需要利用 Shift 的值确定所按下的按键。

当发生按键操作时,这三个事件发生的先后次序是 KeyDown→KeyPress→KeyUp。

KeyPress 将每个字母的大写、小写形式作为不同的键代码,即作为两种不同的字符。而 KeyDown 和 KeyUp 用两个参数确定字母的大写、小写形式。KeyCode 表示键盘上物理按键(即在键盘上 A 和 a 是同一个键),Shift 指明 Shift 键的状态,从而确定所按下的是大写还是小写字母。

(2) 比较文本框的 Change、KeyPress 事件

文本框的 Change 事件是当文本框 Text 属性发生改变时触发的事件。当用户按下并释放键盘上的一个按键时触发焦点所在文本框控件的 KeyPress 事件,该事件返回一个 KeyAscii 参数,表明所按的按键。

实际上,当在文本框中输入一个字符时,能够分别触发这两个事件。输入字符时,先触发 KeyPress 事件,完成输入操作且文本框的 Text 属性值发生改变时,触发 Change 事件。

【习题 4.3】 设计一个程序,试用标签的 WordWrap 属性。总结 WordWrap 属性的作用与特点。

【解】

在窗体上添加一个名称为 lblTest 的标签。

编写如下两个事件过程。

```
Private Sub Form_Load()  
    lblTest.AutoSize=True  
    lblTest.WordWrap=True  
    lblTest.Caption="VisualBasic Programming"  
End Sub
```

单击窗体时,先将标签的 Width 属性设置为一个比较小的值(字符串不能完全显示在其中),再改变 WordWrap 的属性。

```
Private Sub Form_Click()  
    lblTest.Width=1000                                '设置 lblTest 的宽度  
    lblTest.WordWrap=Not lblTest.WordWrap             '转换"自动换行"属性  
End Sub
```

WordWrap 属性的作用是当标签中所显示的内容超出标签的长度时,折行显示。若要使标签保持水平方向的大小不变,且能够显示全部文本,应将 WordWrap 和 AutoSize 属性设置为 True。

如果希望标签控件只在水平方向扩展,则应将 WordWrap 设置为 False。若不希望改变标签大小,应将 AutoSize 设置为 False。

图 4.1 是上述两个事件过程分别对应的单击窗体时的情况。



图 4.1 运行习题 4.3

【习题 4.4】 命令按钮的 Default 和 Cancel 属性的作用是什么？

【解】

一个窗体中有一个命令按钮可以被设置为缺省命令按钮。在程序运行时按下回车键,相当于选择该缺省命令按钮,激活该命令按钮的单击事件。将某个命令按钮的 Default 属性设置为 True 时,该按钮即为缺省的命令按钮,窗体中其他命令按钮的 Default 属性被自动设置为 False。这样,当该命令按钮所在窗体为当前窗体时,按回车键就可选中该命令按钮。

一个窗体中可以有一个命令按钮控件被设为取消按钮。当一个命令按钮控件被设为取消按钮时,按 Esc 键的效果与单击该按钮的效果相同,触发相同的事件过程。将命令按钮的 Cancel 属性设置为 True,该按钮被设置为取消按钮,窗体中其他命令按钮控件的 Cancel 属性自动地被设置为 False。

【习题 4.5】 若将文本框的 ScrollBars 属性设置为 0、1 或 2,是否可以将文本框设置为显示多行文本？

【解】

若仅仅设置文本框的 ScrollBars 属性,无法使文本框显示多行文本。需要将 Multiline 属性设置为 True。

【习题 4.6】 设计一个程序,试用文本框的 Locked 属性,总结 Locked 的作用。

【解】

文本框 Locked 属性的含义是锁定。当 Locked 属性值为 True 时,文本框被锁定,不可以编辑文本框中的内容。

文本框的 Enable 属性值为 False 时,也不能向文本框中输入数据。但是两者有不同。文本框被锁定(Locked 属性值为 True)时,背景不是灰色,文本框依然能够获得焦点,仅仅就是不能编辑、输入。而 Enable 属性值为 False 时,文本框的背景色变为灰色,不可以响应任何事件,不能获得焦点,也不能进行编辑。

为测试文本框的 Locked 属性,设计如图 4.2 所示的窗体。窗体上有名称为 Text1 的文本框和两个命令按钮,分别为 Locked=True 和 Locked=False。

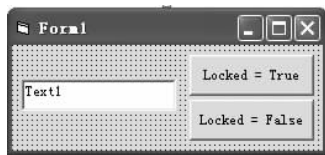


图 4.2 习题 4.6 窗体外观

程序代码如下。

```
Private Sub cmdLockedFalse_Click()  
    Text1.Locked=False  
    Text1.SetFocus  
End Sub  
  
Private Sub cmdLockedTrue_Click()  
    Text1.Locked=True  
    Text1.SetFocus  
End Sub
```

【习题 4.7】 控件焦点的含义是什么？文本框的 SetFocus 方法的作用是什么？

【解】

控件的焦点就是将光标设置在某个对象上,使该对象成为当前对象。

一个对象成为焦点,才能接受输入等相关操作。如 `Text1.SetFocus` 就是设置 `Text1` 成为焦点。用鼠标单击某对象也会使其获得焦点。

【习题 4.8】 图片框和图像框的特点各是什么? 在使用上有什么区别? 什么情况下可以互相替代?

【解】

图像框控件和图片框控件相似,都可用来显示应用程序中的图形,图形的加载方法也相同。在作为图片的容器时,两者可以互换。

图片框 `PictureBox` 和图像框 `Image` 控件的主要区别如下。

- (1) 图像框控件比图片框控件占内存少,且重新绘图的速度快。
- (2) 图片框控件可以作为容器控件,图像框控件则不能作为容器控件使用。
- (3) 图像框控件所具有的属性和方法比图片框控件要少。
- (4) 图像框控件可以伸展图片的大小,使之适应控件的大小。图片框控件不支持这种特性。

【习题 4.9】 哪些类型的图片可以装入图片框? 哪些类型的图片可以装入到图像框?

【解】

图片框和图像框控件可以使用以下类型的图形文件。

- (1) 位图(bitmap),文件后缀为 .bmp。
- (2) 图标(icon),文件后缀为 .ico。
- (3) Windows 元文件(metafile),文件后缀为 .wmf。
- (4) 后缀为 .jpg 及 .gif 的压缩图形文件。

图片框除了可以显示图形,还可以作为其他控件的容器,具有画图的属性和图形方法,能够作为容器控件。而图像框不具备上述特点。

【习题 4.10】 怎样在图片框中显示文本信息?

【解】

在图片框中显示文本信息的最直接方法是使用 `Print` 方法。若要指定显示文本的位置,可在 `Print` 方法之前设定当前坐标的位置。

另一种方法是图片框作为容器,在其中增加文本框控件,然后将信息显示在文本框中,其效果类似于显示在图片框中。

4.2 实验导引

通过本章的实验,了解命令按钮、文本框、标签、滚动条、图片框、图像框、计时器等控件的部分属性。通过编写简单的程序,练习并掌握文本框、标签和命令按钮等控件的基本使用方法。同时,进一步了解在 VB 集成环境中设计界面、编辑程序、运行程序的一般方法。

了解常用控件的命名规则,在程序中尽可能按照规范使用变量名和控件名,培养良好的编程风格。练习编写程序,通过上机调试、运行程序,进一步理解程序设计的思路和方法。

【实验题 4.1】 设计程序,窗体如图 4.3 所示。要求在“输入数据”框中输入一个数字。当单击“计算”命令按钮时,将输入的数据乘以 10 后显示在另一个标签中。



图 4.3 实验题 4.1 窗体外观

【解】

按照图 4.3 在窗体上添加一个文本框、三个标签、两个命令按钮,并按照表 4.1 设置各个控件的属性值。

表 4.1 实验题 4.1 对象属性设置

对 象	属 性	设 置
标签 1	(名称)	Label1
	Caption	输入数据
标签 2	(名称)	Label2
	Caption	输入数据 * 10
标签 3	(名称)	lblResult
	Caption	(置空)
	BorderStyle	1
文本框	(名称)	txtInput
	Text	(置空)
命令按钮 1	(名称)	cmdStart
	Caption	计算
命令按钮 2	(名称)	cmdExit
	Caption	退出

“计算”命令按钮的事件过程代码如下。

```
Private Sub cmdStart_Click()  
    x=Val(txtInput.Text)                                '字符转换成数字  
    lblResult.Caption=Str$ (x * 10)                     '数值型数据换成字符,并显示  
End Sub
```

程序中第一行是将文本框中的输入内容赋给变量 x 。由于文本框中所接受的内容为字符型数据,而算术运算要求参与运算的是数值型数据,所以语句中使用 Val 函数将文本框 txtInput 中的数字字符转换成数值,然后赋给 x 。第二行语句的作用是先将 x 的值乘上 10,再使用 Str\$ 函数将其转换成字符型数据,最后赋值给 lblResult 的 Caption 属性。

以下过程是单击“退出”按钮时执行的事件过程。

```
Private Sub cmdExit_Click()
    End
End Sub
```

【实验题 4.2】 在实验题 4.1 的基础上,进一步考虑对输入数据的校验,要求只能输入数值数据。如果输入字母或其他非数值数据,应给出提示,并提示重新输入。

【解】

我们考虑一个问题,当运行实验题 4.1 的程序时,向文本框中输入一个字母或其他字符,例如 a,再单击“计算”按钮,则程序运行结果如图 4.4 所示,而这是不符合实际情况的。字符 a 乘以 10 没有任何意义。造成这种错误的原因是由于语句 $x = \text{Val}(\text{txtInput.Text})$ 中使用了 Val 函数。Val 函数的作用是将数值字符转换为数值。当函数的参数是非数值型的字符时,函数不能正确使用。



图 4.4 实验题 4.1 对输入 a 的响应

为避免出现这类问题,增强程序的健壮性,应限制输入到文本框中的字符只能是 0~9 这 10 个数字。为此可以在 KeyPress 事件过程中判断输入的字符,程序代码如下。

```
Private Sub txtInput_KeyPress(KeyAscii As Integer)
    If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
        MsgBox "请输入数值数据!", 0+48, "提示"
        KeyAscii = 0
        txtInput.SetFocus
    End If
End Sub
```

当文本框 txtInput 获得输入焦点时,按下键盘上任意按键,都将触发 txtInput_KeyPress() 事件过程。括号中的 KeyAscii 是由系统返回的按键所对应的 ASCII 字符代码。如果输入的不是 0~9 之间的字符,即按键所对应的 ASCII 字符 KeyAscii 小于字符 0 的 ASCII 码 Asc("0"),或大于字符 9 的 ASCII 码 Asc("9"),则用消息框给出提示,如图 4.5 所示。



图 4.5 错误提示

由于输入字符后才触发 txtInput_KeyPress() 事件过程,因此,不论输入的字符是否正确,都会将该输入字符显示在文本框中。使用 KeyAscii=0 语句,可以从文本框中清除所输入的不满足条件的非数值字符。大家可以试着去掉 KeyAscii=0 语句,观察程序运行的效果,体会这条语句的作用。

对实验题 4.1 中单击命令按钮的事件过程 cmdStart_Click() 进行一些修改,使得该事件过程能够在文本框为空时给出提示。程序如下。

```
Private Sub cmdStart_Click()
    If Trim(txtInput.Text) = "" Then
        MsgBox "输入不能为空!", 0+48, "提示"
```

```
txtInput.SetFocus
Else
    x=Val(txtInput.Text)
    lblResult.Caption=Str$ (x * 10)
End If
End Sub
```

“退出”按钮的事件过程如下。

```
Private Sub cmdExit_Click()
    End
End Sub
```

经过上述修改,增强了程序的健壮性。

【实验题 4.3】 设计一个程序,窗体中有两个文本框、两个命令按钮,窗体外观如图 4.6 所示。

当在第一个文本框中输入信息时,立刻在第二个文本框中显示相同的内容;或在第二个文本框中输入信息时,立刻在第一个文本框中显示相同的内容。

单击“清除”按钮时,清除两个文本框中的信息。单击“退出”按钮,结束程序的运行。

【解】

程序要求在第一个文本框中输入信息时,立刻在第二个文本框中显示相同的内容;或在第二个文本框中输入信息时,立刻在第一个文本框中显示相同的内容。

按照图 4.6 在窗体上添加文本框和命令按钮控件,并按照表 4.2 设置各个控件的属性。



图 4.6 实验题 4.3 窗体外观

表 4.2 实验题 4.3 窗体及相关对象属性设置

对 象	属 性	设 置
窗体	(名称)	Form1
	(名称)	txtFirst
文本框 1	Text	(置空)
	(名称)	txtSecond
文本框 2	Text	(置空)
	(名称)	cmdClear
命令按钮 1	Caption	清除
	(名称)	cmdExit
命令按钮 2	Caption	退出

当向文本框 txtFirst 输入字符时,触发 txtFirst 的 Change 事件。在该事件过程中,将文本框 txtFirst 中的内容赋值给 txtSecond 的 Text 属性,使 txtSecond 的内容与 txtFirst 的内容相同,程序如下。

```
Private Sub txtFirst_Change()  
    txtSecond.Text=txtFirst.Text  
End Sub
```

类似地,txtSecond 的 Change 事件过程如下。

```
Private Sub txtSecond_Change()  
    txtFirst.Text=txtSecond.Text  
End Sub
```

当单击“清除”按钮时,将两个文本框中的内容清除,并将光标的焦点设置到文本框 txtFirst 上。

```
Private Sub cmdClear_Click()  
    txtFirst.Text=""  
    txtSecond.Text=""  
    txtFirst.SetFocus  
End Sub
```

“退出”按钮的事件过程如下。

```
Private Sub cmdExit_Click()  
    End  
End Sub
```

【实验题 4.4】 修改教材第 4 章例 4.4 的程序,使文本框中能放入更多的信息。由于文本过长,文本框不能直接把所有内容全部显示出来,因此,应该用滚动条将原来在文本框范围以外的内容顺序移到文本框中显示出来。

【解】

按照图 4.7 在窗体上添加一个文本框、两个命令按钮控件,并按照表 4.3 设置各控件的属性。



图 4.7 实验题 4.4 窗体外观

表 4.3 实验题 4.4 窗体及相关对象属性设置

对 象	属 性	设 置
窗体	(名称)	Form1
窗体	(名称)	Form1
	Caption	带滚动条的文本框
文本框	(名称)	txtText
	Text	(置空)
	MultiLine	True
	ScrollBars	2

对 象	属 性	设 置
命令按钮 1	(名称)	cmdShowText
	Caption	显示文本
命令按钮 2	(名称)	cmdExit
	Caption	退出

若文本框允许显示多行文本,通过设置 ScrollBars 属性,可以为文本框添加水平滚动条或垂直滚动条,也可以同时添加水平和垂直两个方向的滚动条。

单击文本框时,在文本框中显示一段文本,代码如下。

```
Private Sub cmdShowText_Click()
```

txtText.Text="如果文本框太小,不能容纳所有内容,为了查阅方便,可以为文本框加上滚动条,通过移动滚动条,能够查看全部文本内容。文本框的 ScrollBars 属性用来设置是否加上滚动条。它共有 4 个属性值: 0 (None)——不加滚动条(缺省); 1(Horizontal)——只加水平滚动条; 2(Vertical)——只加垂直滚动条; 3(Both)——加垂直和水平滚动条。

```
End Sub
```

由于将 ScrollBars 属性设置为 2,即为文本框添加了垂直滚动条,可以移动滚动条,显示文本的内容。

“退出”按钮的事件过程如下。

```
Private Sub cmdExit_Click()
```

```
End
```

```
End Sub
```

【实验题 4.5】 窗体上有一个标签、一个文本框、两个命令按钮。在文本框中输入密码,然后单击“核对密码”命令按钮,如果密码不正确,在文本框中显示“重新输入密码”的信息;如果密码正确,在文本框中显示“欢迎!”的文字。窗体外观如图 4.8 所示。

【解】

按照图 4.8 在窗体上添加控件,并按照表 4.4 设置各个控件的属性。

假设系统已经设置的密码为 BASIC。当在文本框中输入一个字符串并单击“核对密码”按钮时,将文本框中输入的字符串与程序中预先设定的密码 BASIC 进行比较,如果相同,则通过密码验证,显示提示信息。“核对密码”命令按钮的事件过程如下。



图 4.8 实验题 4.5 窗体外观

表 4.4 实验题 4.5 窗体及相关对象属性设置

对 象	属 性	设 置
窗体 1	(名称)	frmFirst
文本框	(名称)	txtPWD
	PasswordChar	*
标签 1	(名称)	Label1
	Caption	请输入密码
命令按钮 1	(名称)	cmdCheck
	Caption	核对密码
命令按钮 2	(名称)	cmdExit
	Caption	退出

```
Private Sub cmdCheck_Click()  
    Static Counter As Integer  
    password$ =txtPWD.Text  
    If UCase(password$ )="BASIC" Then  
        MsgBox "密码输入正确!"  
    Else  
        Counter=Counter+1  
        lblTitle.Caption="请重新输入密码!"  
        txtPWD.Text=""  
        txtPWD.SetFocus  
        If Counter=3 Then          '三次输入错误则退出  
            End  
        End If  
    End If  
End Sub
```

由于输入的字符可能有大小写之分,可以在程序中使用 UCsae 函数将所有输入的字符改为大写,再与程序中设定的密码进行比较。

题目要求当输入的密码不正确时,给出提示,提示重新输入。但若连续三次输入错误,则退出系统。程序中用变量 Counter 统计输入的次数。程序中将 Counter 设为静态变量,这样,在每次执行命令按钮的事件过程之后,该变量的值仍然保存。

请读者考虑一个问题: 如果不将 Counter 设置为静态变量,将会出现什么情况?
“退出”按钮的事件过程如下。

```
Private Sub cmdExit_Click()  
    End  
End Sub
```

【实验题 4.6】 窗体上有一个图片框、一个滚动条。在图片框中装入一个图片。通过单击滚动条的操作,改变图片框的大小。滚动条的变化范围是 0~10,每次单击滚动条时,图片框增加或缩小的尺寸为 30(缇)。

【解】

按照图 4.9 在窗体上添加控件,按照表 4.5 设置窗体及各控件的属性。



图 4.9 实验题 4.6 窗体外观

表 4.5 实验题 4.6 窗体及相关对象属性设置

对 象	属 性	设 置
图片框	(名称)	PictureBox
	Autosize	False
命令按钮	Caption	退出
	(名称)	cmdExit
水平滚动条	(名称)	Hscroll1
	Max	10
	Min	0

运行程序,窗体被装入,触发 Load 事件,执行 Form_Load()事件过程,将图片装入图片框。程序代码如下。

```
Private Sub Form_Load()  
    pictureBox.Picture=LoadPicture("c:\Amhappy.wmf")  
End Sub
```

改变滚动条中滑块的位置时,触发 HScroll1_Change()事件过程。代码如下。

```
Private Sub HScroll1_Change()  
    Static position As Integer  
    If position<HScroll1.Value Then  
        pictureBox.Height=pictureBox.Height+30  
        pictureBox.Width=pictureBox.Width+30  
    Else  
        pictureBox.Height=pictureBox.Height-30  
        pictureBox.Width=pictureBox.Width-30  
    End If  
    position=HScroll1.Value  
End Sub
```

HScroll1_Change()事件过程中定义了一个静态变量 position,作用是当过程执行结束时,保存当前滚动条位置的值。

程序进入运行状态后,单击水平滚动条的滚动箭头或移动滚动块,滚动条的值发生变

化,触发滚动条的 Change 事件,执行 HScroll1_Change()事件过程语句。

程序中的静态变量 position 的初始值为 0,当单击一次滚动条右侧的箭头时,滚动条 HScroll1 的 Value 属性值由 0 变化到 1,此时 position 值小于 Value,If 语句的条件为真,图片框 picBox 的高和宽各加 30,分别赋给 picBox 的 Height 属性和 Width 属性,图片框放大,由于图片框的 Autosize 属性值为 False,图片框中的图片自动调整大小随图片框变大。然后执行 End If 下面的语句,将滚动条当前的 Value 值赋给 position 变量。

再次单击滚动箭头,Value 的值由 1 变化到 2,由于在过程中 position 被定义为静态变量,所以 position 保留了上一次过程结束时的值 1,position 值 1 仍小于 Value 值 2,满足 If 语句的条件,图片框 picBox 再次放大。然后将滚动条的当前 Value 值赋给 position。结束事件过程。如此操作下去,当 Value 的值变化到最大(10)时,position 中存放的值是 9,图片框继续放大,最后 Value 的值 10 赋给 position,过程结束,position 将 10 保留下来。由于我们事先限制了滚动条的最大变化范围为 10,所以当 Value 的值为 10 时,滚动箭头已处在了滚动条的最右端,如果要再次触发 Change 事件,只能单击左端的滚动箭头。

我们单击左端的滚动箭头,Value 的值由 10 变化到 9,上一次过程结束时 position 中保留的值为 10,所以不满足 If 语句的条件,执行 Else 后面的语句,图片框的高和宽各缩小 30。如此操作下去,图片框不断缩小,直到 Value 的值变化到 0。

如果在过程中将 position 变量定义成普通变量,而不是静态变量,那么过程结束后 position 不能保留上一次的值,也就是说,每当触发 Change 事件,执行过程代码时,position 的值都为 0,position 值永远小于 Value 值,会发生什么情况,请读者想一想。

单击“退出”按钮,执行如下过程,结束程序的运行。

```
Private Sub cmdExit_Click()  
    End  
End Sub
```

【实验题 4.7】 设计一个计时秒表,能够显示当前日期、时间,以及是上午还是下午。界面自行设计。

【解】

设计窗体如图 4.10 所示。

按表 4.6 设置窗体及各个控件的属性。

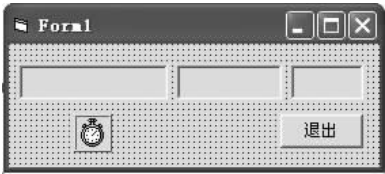


图 4.10 实验题 4.7 窗体外观

表 4.6 实验题 4.7 窗体及相关对象属性设置

对 象	属 性	设 置
计时器	(名称)	Timer1
	Interval	1000
命令按钮	(名称)	cmdExit
	Caption	退出

对 象	属 性	设 置
标签 1	(名称)	lblDate
	Caption	(置空)
	BorderStyle	1
标签 2	(名称)	lblTime
	Caption	(置空)
	BorderStyle	1
标签 3	(名称)	lblAP
	Caption	(置空)
	BorderStyle	1

运行程序。设每一秒钟数字表改变一次显示的内容。定时执行的操作在 Timer1 控件的 Timer1_Timer() 事件过程中实现,程序代码如下。

```
Private Sub Timer1_Timer()
    lblDate.Caption=Date$
    lblTime.Caption=Time$
    If Val(Left(Time$, 2))<12 Then
        lblAP.Caption="上午"
    Else
        lblAP.Caption="下午"
    End If
End Sub
```

Timer1_Timer() 事件过程的前两句是将日期函数值和时间函数值分别赋给标签控件 lblDate 和 lblTime 的 Caption 属性。

运行程序后的结果如图 4.11 所示。

我们知道,时间函数 Time\$ 的值是以 hh:mm:ss 形式表示当前时间的,所以 If 条件中判断小时数,应该取时间的前两位,Left(Time\$, 2) 的作用是取出 Time\$ 值最左侧的两个数字,即小时数。例如,21:29:38,取出最左面的两个字符就是“21”。用取出的小时数与“12”进行比较,如果小于“12”,表示是上午,将“上午”字符串赋给标签控件 lblAP 的 Caption 属性;反之,如果超过“12”,表示是下午,将“下午”字符串赋给标签控件 lblAP 的 Caption 属性。

单击“退出”按钮,执行如下过程,结束程序的运行。

```
Private Sub cmdExit_Click()
    End
End Sub
```

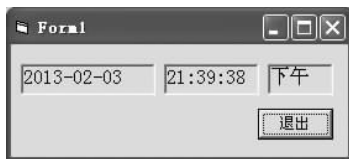


图 4.11 运行实验题 4.7