第二部分　Visual Basic 程序设计实验指导

实验 1　Visual Basic 环境与简单程序设计

一、实验目的

1．掌握在 Visual Basic 环境中创建简单应用程序的方法。
2．掌握在窗体上添加控件的方法以及对控件的调整方法。
3．掌握简单代码的编写。

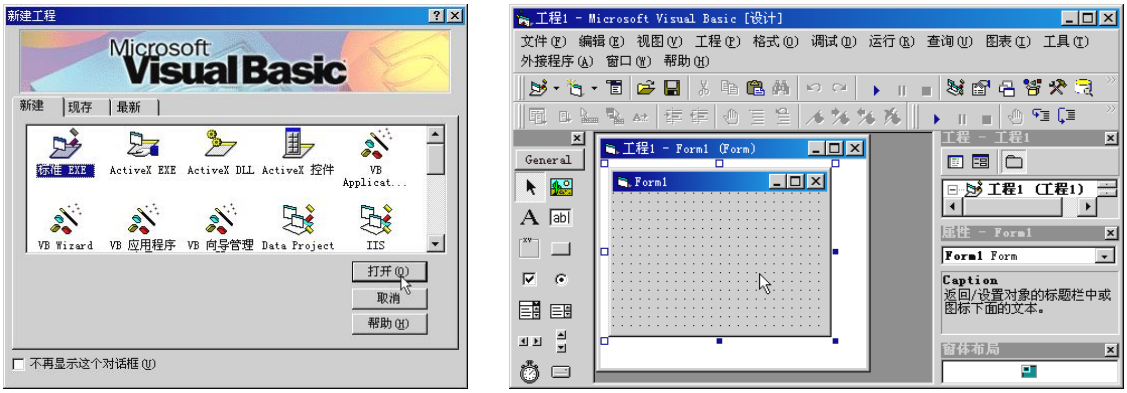
二、实验内容

1．创建一个无代码的简单程序。
2．创建一个含有简单代码的程序。
3．将 Visual Basic 工程编译生成可执行文件。

三、实验步骤与指导

1．创建一个无代码的简单程序

（1）创建工程。启动 Visual Basic，在“新建工程”对话框中选择“标准 EXE”（如图 1.1 所示），单击“打开”按钮，进入 Visual Basic 集成开发环境（IDE）。



（a）新建工程　　（b）Visual Basic 集成开发环境（IDE）

图 1.1　创建工程

（2）设计界面。双击工具箱中的 Label 控件，在窗体上添加一个标签（Label1）。

（3）设置属性。设置标签属性。在界面设计窗口中选定标签，在属性窗口中将标签的 Caption 属性值改为“Hello，World!”。单击 Font 属性右侧的…按钮，在对话框中将字体大小设置为二号。在界面设计窗口中调整标签控件的大小，使“Hello，World!”显示为一行。通过“格式”菜单中的“在窗体中居中对齐”菜单项（如图 1.2 所示）将标签放置在窗体中央。

设置窗体属性。在属性窗口中将窗体的 Caption 属性值改为“我的第一个程序”。

（4）运行程序。单击工具栏中启动按钮▶或按 F5 键运行应用程序，程序运行结果如图 1.3 所示。

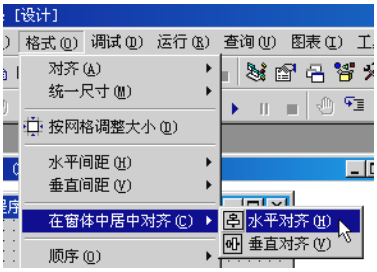


图 1.2　“格式”菜单

图 1.3　第一个程序

2. 创建一个含有简单代码的程序

（1）新建工程。在 Visual Basic 集成开发环境中单击“文件”菜单，选择“新建工程”。

（2）设计界面。单击工具箱中的 CommandButton 控件，在窗体上拖动鼠标“画出”三个命令按钮；单击工具箱 TextBox 控件，在窗体上画出文本框。

（3）设置属性。单击特定对象，然后在属性窗口中进行如下设置：

将三个命令按钮（Command1～Command3）的 Caption 属性分别设为“画圆”、“清除”和“退出”。将文本框的 Text 属性设为空。将窗体的 Caption 属性设为“在窗体上画圆”，FillColor 属性设为浅绿色，FillStyle 属性设为 0-Solid。

（4）编写代码。双击“画圆”按钮，打开代码编辑器窗口，在光标闪动处添加以下代码：

```
Form1.Circle (1100, 1000), 800, vbBlue
Text1.Text = "画一个实心圆"
```

在代码编辑器窗口内左上部的“对象”组合框列表中选定 Command2，在右上部的“过程”组合框列表中选择 Click，然后在光标闪动处添加以下代码：

```
Form1.Cls
Text1.Text = "实心圆消失了"
```

在代码编辑器窗口内左上部的“对象”组合框的列表中选定 Command3，在右上部的“过程”组合框列表中选择 Click，然后在光标闪动处添加以下代码：

```
End
```

单击工具栏中的启动按钮▶或按 F5 键运行应用程序，程序运行效果如图 1.4 所示。

3. 将 Visual Basic 工程编译生成可执行文件

将上述工程保存后，单击“文件”菜单中的“生成….exe”菜单项，在对话框中选择保存

位置并输入文件名，然后单击“确定”按钮。退出 Visual Basic 开发环境，双击已生成的.exe 文件运行。

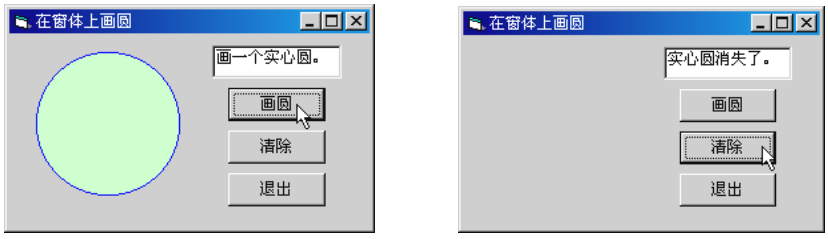


图 1.4　含有简单代码的应用程序

4. 代码快速输入技巧

Visual Basic 代码编辑器具有自动完成关键字的功能。若某些关键字或对象名称较长，或忘记了它们的完整拼写形式，只记得其前几个字母，利用此功能，即可快速、准确地输入关键字或对象名称。

方法 1：用快捷键。先输入关键字或对象名称的前几个字符，然后按 Alt+→键，此时在插入点处将会出现如图 1.5 所示的快速列表，用↓或↑键选中所需关键字或对象名称，然后按 Tab 键或其他分隔符（如空格、圆点“.”、逗号、等号、非字母运算符等），即可准确无误地输入该关键字或对象名称。用鼠标双击快速列表中的关键字或对象名称亦可完成输入。

方法 2：用工具栏按钮。先输入关键字或对象名称的前几个字符，然后单击“编辑”工具栏上的按钮（如图 1.6 所示），也可调出图 1.5 所示的快速列表。若“编辑”工具栏未显示，可以通过单击菜单“视图”→“工具栏”→“编辑”显示该工具栏。

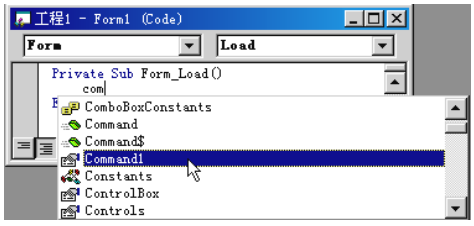


图 1.5　自动完成关键字

图 1.6　编辑工具栏

例如，某命令按钮的名称为 cmdQuestion，要将其 Enabled 属性设为 False，可以按以下步骤操作：

输入 cmdq，按 Alt+→键，在快速列表中选中该项，输入点号“.”，输入“e”，输入赋值号“=”，按 Tab 键。

实验 2　数据与表达式

一、实验目的

1．掌握变量的定义方法。
2．掌握常用内部函数的用法。
3．掌握运算符和表达式的用法。

二、实验内容

1．利用 Chr、Int 和 Rnd 函数随机生成大小写字母。
2．变量的定义和引用；用 Int 和 Rnd 函数生成指定范围的随机整数；用 Randomize 函数初始化随机数发生器；用 Val 函数将数字字符串转换为数值。
3．练习 Left$、Right$、Mid、Len、InStr、Ucase 和 Lcase 等字符串函数的使用。

三、实验步骤与指导

1．随机生成大小写字母

（1）设计界面并设置属性。在窗体上放置两个命令按钮，将其名称分别设为 cmdUcase 和 cmdLcase，将 Caption 属性分别设为“大写字母”和“小写字母”。

（2）编写代码。在两个命令按钮的 Click 事件过程中，分别利用 Chr、Int 和 Rnd 函数组成的表达式生成随机大写和小写字母，并通过 Print 方法在窗体上显示表达式的结果。

生成随机大写字母的表达式：Chr(Int(Rnd * 26) + 65)

生成随机小写字母的表达式：Chr(Int(Rnd * 26) + 97)

程序运行结果如图 2.1 所示。

2．生成指定范围的随机整数

（1）设计界面。在窗体上放置三个文本框，分别用于输入随机数的下界和上界，显示生成的随机整数；三个命令按钮，分别用于执行随机整数的生成和显示，清除文本框的内容，结束程序运行；三个标签，用于对文本框进行简要说明。

（2）编写代码。在命令按钮的 Click 事件过程中定义两个 Integer 型变量，用 Val 函数将输入文本框中的随机数上、下界转换为数值赋予变量。利用 Int 和 Rnd 函数以及变量组成的表达式生成随机整数，并赋值给输出文本框的 Text 属性。

生成指定范围随机整数的表达式为：Int(Rnd * (上界 – 下界 + 1) + 下界)

在窗体的 Load 事件过程中调用 Randomize 函数，对随机数发生器进行初始化。

程序运行结果如图 2.2 所示。

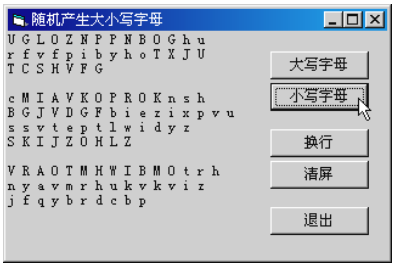


图 2.1　随机生成大小写字母

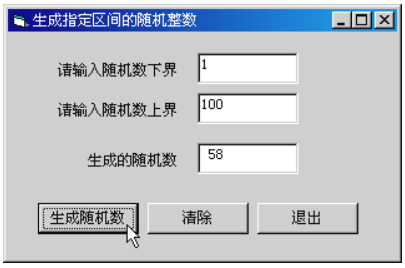


图 2.2　生成指定范围的随机数

3. 字符串函数的使用

（1）设计界面并设置属性。在窗体上放置两个文本框，在属性窗口中将 Text1 的 Text 属性设为 String Functions Demo 作为原始字符串，Text2 用于显示字符串函数的执行结果（若要使文本框显示多行文本，需要在属性窗口中将其 MultiLine 属性设为 True）。

在窗体上放置七个命令按钮，将其 Caption 属性分别设为“Left 函数”、“Right 函数”、“Mid 函数”、“Len 函数”、“Instr 函数”、“UCase 函数”和“LCase 函数”。

（2）编写代码。在代码窗口的“通用-声明”部分定义一个模块级的 String 型变量 strS，在窗体的 Load 事件过程中将原始字符串赋值给该变量。

在各命令按钮的 Click 事件过程中调用对应的字符串函数，并将执行结果显示在 Text2 中。各按钮的功能要求如下：

显示原始字符串的前 3 个字符、最后 3 个字符、第 8～第 11 个字符和总字符数；显示字母“D”在原始字符串中的位置；将原始字符串全部转换为大写或小写。例如：

```
Dim strS1 As String          '定义过程级（局部）变量
' 调用 InStr 函数组成表达式为变量赋值。vbCrLf=Chr(13)+Chr(10)，即回车换行
strS1 = "字母“D”是字符串的" & vbCrLf & " 第 " _& InStr(strS, "D") & " 个字符。"
Text2.Text = strS1           ' 显示函数执行结果
```

程序运行效果如图 2.3 所示。

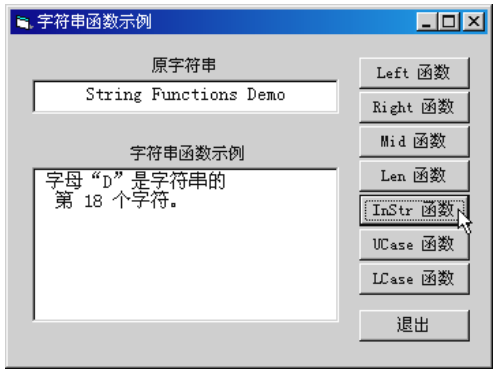


图 2.3　字符串函数示例

4. 在“立即”窗口中练习内部函数、运算符和表达式的使用

用 Print 方法可以在“立即”窗口中获得内部函数调用或表达式运算的结果。问号“?”可以代替 Print 关键字，例如? Time 与 Print Time 等效。

（1）日期/时间函数，例如：Time、Date、Day、Month、Year、Weekday 等。

（2）数学函数，例如：Abs、Sqr、Log 等。

（3）转换函数，例如：Chr、Str、Val 等。

（4）由函数、运算符和数据组成的表达式。

实验 3　Visual Basic 程序设计初步

一、实验目的

1．掌握窗体及三个基本控件（命令按钮、标签和文本框）的常用属性、事件和方法。
2．掌握赋值语句、InputBox 函数和 MsgBox 函数的应用。

二、实验内容

1．窗体常用属性、事件和方法的编程；命令按钮常用事件编程。
2．标签控件的属性设置。
3．文本框常用属性、事件和方法的编程。

三、实验步骤与指导

1．窗体程序设计

通过代码设置窗体的前景色、背景色、字体和字号，用窗体的 Left、Top 属性和 Move 方法移动窗体，用 Print 方法显示窗体的当前位置。

（1）设计界面并设置属性。在窗体上放置两个命令按钮，将 Command1 的 Caption 属性设为“改变属性值移动窗体”，Command2 的 Caption 属性设为“用 Move 方法移动窗体”。将窗体的 MaxButton 属性设为 False（窗体最大化或最小化时，若移动窗体位置将会出错）。设计界面如图 3.1（a）所示。

（2）编写代码。在窗体的 Load 事件中将窗体的 Caption 属性设为“窗体属性、方法示例”。设窗体的前景色 ForeColor 为蓝色(vbBlue)，背景色 BackColor 为白色(vbWhite)，字体 FontName 为“黑体”，字号 FontSize 为 12，Left 和 Top 均为 300。

在 Command1 的 Click 事件中通过改变 Left 和 Top 属性，使窗体右移、下移各 200 缇；在 Command2 的 Click 事件中用 Move 方法使窗体右移、下移各 200 缇。

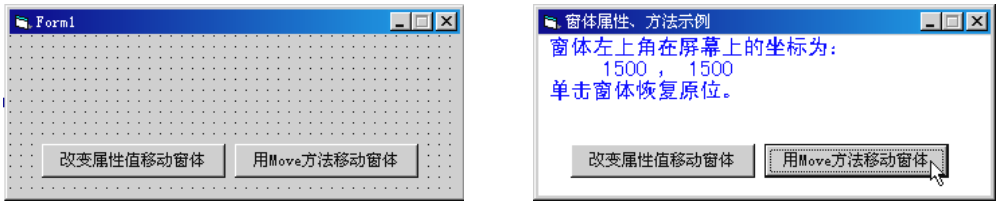
在窗体的 Click 事件中通过改变 Left 和 Top 属性，使窗体恢复原位。

每次移动窗体以及窗体复位时，用 Print 方法在窗体上显示窗体的当前坐标。

程序运行效果如图 3.1（b）所示。

程序代码如下：

```
Option Explicit

Private Sub Command1_Click()
    '改变 Left 和 Top 属性移动窗体
```



（a）设计时界面　　（b）运行时界面

图 3.1　窗体程序设计

```
    Me.Left = Me.Left + 200
    Me.Top = Me.Top + 200
    Cls
    Print "窗体左上角在屏幕上的坐标为："
    Print Tab(6); Me.Left; ", "; Me.Top
    Print " 单击窗体恢复原位。"
End Sub

Private Sub Command2_Click()
    '用 Move 方法移动窗体
    Me.Move Me.Left + 200, Me.Top + 200
    Cls
    Print " 窗体左上角在屏幕上的坐标为："
    Print Tab(6); Me.Left; ", "; Me.Top
    Print " 单击窗体恢复原位。"
End Sub

Private Sub Form_Click()
    Me.Left = 300
    Me.Top = 300
    Cls
    Print " 窗体左上角在屏幕上的坐标为："
    Print Tab(6); Me.Left; ", "; Me.Top
End Sub

Private Sub Form_Load()
    '设置窗体的属性
    Me.Caption = "窗体属性、方法示例"
    Me.FontSize = 12
    Me.FontName = "黑体"
    Me.ForeColor = vbBlue
    Me.BackColor = vbWhite
    Me.Left = 300 '设置窗体位置的初始坐标
    Me.Top = 300
End Sub
```

2. 用标签制作浮雕效果文字

利用两个标签控件，在设计时通过白色/黑色错位叠加，实现如图 3.2 所示的文字浮雕效果。

图 3.2　浮雕效果文字

（1）在窗体上放置两个标签控件，其 Caption 属性均为“VB 6.0 中文版”，字号均为“初号”，字体均为“隶书”。调整标签的宽度和高度，使文字分两行显示。

（2）按表 3-1 设置两个标签的其他属性。

表 3-1　标签控件属性

控件名	背景样式（BackStyle）	边框样式（BorderStyle）	前景色（ForeColor）	左（Left）	顶（Top）
Label1	1 - Fixed Single	1 – Opaque（不透明）	&H00FFFFFF&（白）	324	324
Label2	0 - None	0 – Transparent（透明）	&H00000000&（黑）	384	372

用类似的方法，还可以制作如图 3.3 所示的阴影文字。

图 3.3　阴影文字

3. 文本框中选定内容的复制、剪切和粘贴

在文本框 Text1 中选定文本，通过按钮完成复制、剪切和粘贴操作，将复制或剪切的内容粘贴到文本框 Text2 中。

（1）设计界面并设置属性。在窗体上放置两个文本框 Text1 和 Text2，添加四个命令按钮，分别命名为 cmdCut、cmdCopy、cmdPaste 和 cmdEnd，用于剪切、复制、粘贴和退出。用两个标签对文本框进行简要说明。

（2）编写代码。在代码的“通用-声明”部分定义一个模块级变量 strPaste，用于存放待粘贴的内容。

各按钮的功能要求如下：

程序运行时，在 Text1 中选定文本，单击“复制”按钮，通过文本框的 SelText 属性将选

定的内容存入变量；单击“剪切”按钮，先将选定的内容存入变量，再删除选定内容；单击“粘贴”按钮，将变量中的内容粘贴到 Text2 中。单击“退出”按钮结束运行。

完整的程序代码如下：

```
Option Explicit
Dim strPaste As String              '模块级变量用于存放待粘贴的内容

Private Sub cmdCopy_Click()         '复制
    strPaste = Text1.SelText        '将选中的文本复制到变量中
End Sub

Private Sub cmdCut_Click()          '剪切
    strPaste = Text1.SelText        '将选中的文本复制到变量中
    Text1.SelText = ""              '删除文本框中被选中的文本
End Sub

Private Sub cmdEnd_Click()
    End
End Sub

Private Sub cmdPaste_Click()        '粘贴
    '将变量中的内容粘贴到 Text2 中。若 Text2 中有选定的文本
    '则用变量内容将其覆盖；若无选定文本，粘贴到插入点处
    Text2.SelText = strPaste
End Sub
```

程序运行效果如图 3.4 所示。

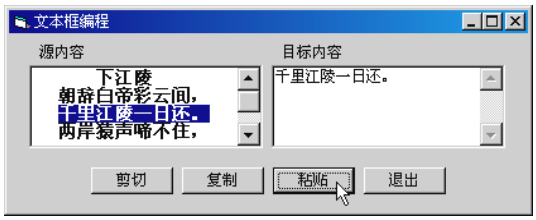


图 3.4　文本框编程

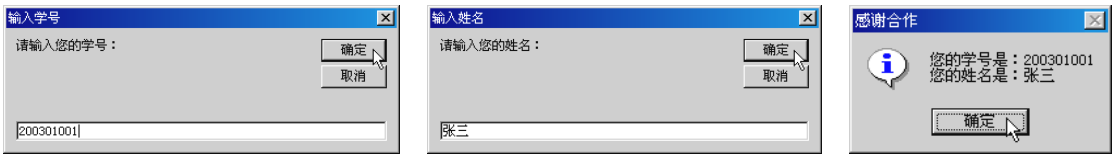
4. InputBox 函数和 MsgBox 函数

在窗体的 Load 事件中两次调用 InputBox 函数，由用户分别输入学号和姓名存入两个变量，然后调用 MsgBox 函数分两行显示输入的学号和姓名。MsgBox 对话框关闭后卸载窗体，即在程序运行期间始终不显示窗体，仅显示三个对话框，如图 3.5 所示。

程序代码如下：

```
Private Sub Form_Load()
    '定义字符串变量用于存放学号、姓名和 MsgBox 函数提示信息
    Dim strNo As String
```

```
    Dim strName As String
    Dim strMsg As String
    '用 InputBox 函数输入学号和姓名
    strNo = InputBox("请输入您的学号：", "输入学号", 200301001)
    strName = InputBox("请输入您的姓名：", "输入姓名")
    'MsgBox 函数的提示信息字符串。vbCr=Chr(13)，即回车
    strMsg = "您的学号是：" & strNo & vbCr & "您的姓名是：" & strName
    '用 MsgBox 函数显示学号和姓名
    MsgBox strMsg, vbInformation, "感谢合作"
    '卸载窗体
    Unload Me
End Sub
```



（a）输入学号　　（b）输入姓名　　（c）显示学号和姓名

图 3.5　InputBox 函数和 MsgBox 函数

实验 4　常用控件

一、实验目的

1．掌握图片框、图像框、定时器、单选按钮和复选框的功能和应用。
2．掌握列表框、组合框和框架的使用。
3．熟悉多窗体的基本操作，掌握窗体操作的常用语句和方法。

二、实验内容

1．图片框、图像框、单选按钮、复选框和框架的综合应用。
2．用定时器制作秒表。
3．列表框及组合框程序设计。
4．多窗体程序设计。

三、实验步骤与指导

1．图片框、图像框、单选按钮、复选框和框架的综合应用

要求：窗体加载时为图片框和图像框载入图片；用框架对单选按钮进行分组；用单选按钮改变图片框和图像框的大小；用复选框指定图片框是否自动改变大小以显示全部图形，图像框是否缩放图形以适应控件大小。

（1）设计界面。在窗体上放置两个框架，其中各放置三个单选按钮。在框架 Frame1 中添加一个图片框，在 Frame2 中添加一个图像框。在窗体上添加两个复选框和四个标签。

（2）设置属性。图像框的 BorderStyle 属性设为 1-Fixed Single，以便在程序运行时使图像框的边界可见。各标签的属性均采用默认值。其他控件的属性设置如表 4-1 所示。

表 4-1　控件属性

对象	名称	属性名	属性值	对象	名称	属性名	属性值
OptionButton	OptEnlargeI	Caption	放大	OptionButton	optEnlargeP	Caption	放大
	OptReduceI	Caption	缩小		optReduceP	Caption	缩小
	OptRevertI	Caption	还原		optRevertP	Caption	还原
		Value	True			Value	True
CheckBox	Check1	Caption	AutoSize	Frame	Frame1	Caption	图片框
	Check2	Caption	Stretch		Frame2	Caption	图像框

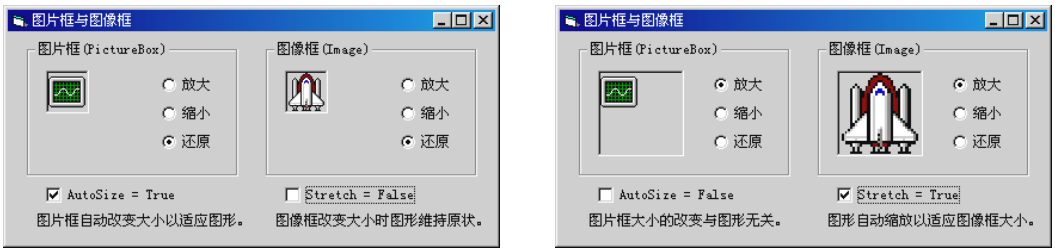
（3）编写代码。在窗体的 Load 事件中用 LoadPicture 函数为图片框和图像框载入图片，并将图片框和图像框的 Width 和 Height 属性存入模块级变量，以备还原之用。

通过各单选按钮的单击事件分别完成图片框和图像框的放大、缩小和还原（即改变控件的 Width 和 Height 属性）。

在复选框 Check1 的单击事件中将该复选框的选中状态（Value 属性值，整型）转换为逻辑型赋值给图片框的 AutoSize 属性，指定图片框是否自动改变大小以显示全部图形，并通过复选框的 Caption 属性显示图片框 AutoSize 属性的当前值（如：AutoSize = True）。

在复选框 Check2 的单击事件中将该复选框的选中状态赋值给图像框的 Stretch 属性，指定图像框是否缩放图形以适应控件大小，并通过复选框的 Caption 属性显示图片框 Stretch 属性的当前值（如：Stretch = False）。

程序的部分运行效果如图 4.1 所示。



（a）原始图形　　（b）放大图片框和图像框

图 4.1　图片框与图像框

程序代码如下：

```
Option Explicit
'模块级变量用于存放图片框和图像框的原始大小
Dim PicW As Integer, PicH As Integer
Dim ImgW As Integer, ImgH As Integer

Private Sub Check1_Click()
    '将复选框的选中状态（Value 属性值）转换为
    '逻辑值后赋值给图片框的 AutoSize 属性
    '决定图片框是否自动改变大小以显示全部图形
    Dim blnV As Boolean
    blnV = Check1.Value
    Picture1.AutoSize = blnV
    Check1.Caption = "AutoSize = " & blnV
    '根据复选框的选中状态确定显示哪个标签
    Label1.Visible = blnV
    Label2.Visible = Not blnV
End Sub

Private Sub Check2_Click()
```

```
    '将复选框的选中状态（Value 属性值）转换为
    '逻辑值后赋值给图像框的 Stretch 属性
    '决定图形是否自动缩放以适应图像框大小
    Dim blnV As Boolean
    blnV = Check2.Value
    Image1.Stretch = blnV
    Check2.Caption = "Stretch = " & blnV
    '根据复选框的选中状态确定显示哪个标签
    Label3.Visible = blnV
    Label4.Visible = Not blnV
End Sub

Private Sub Form_Load()
    '用 LoadPicture 函数为图片框和图像框加载图形
    '两个图形文件与当前工程位于同一文件夹
    'App 是 Visual Basic 预定义的全局对象，代表当前应用程序（工程）
    'Path 是 App 对象的属性之一，指当前应用程序（工程）文件所在的路径
    Picture1.Picture = LoadPicture(App.Path & "\Sinewave.ico")
    Image1.Picture = LoadPicture(App.Path & "\Rocket.ico")
    '设置属性
    Picture1.AutoSize = True
    Image1.Stretch = True
    '存储原始图形大小
    PicH = Picture1.Height
    PicW = Picture1.Width
    ImgH = Image1.Height
    ImgW = Image1.Width
    '设置标签的显示内容和位置
    'Caption 属性是标签的默认属性，可以省略
    Label1 = "图片框自动改变大小以适应图形。"
    Label2 = "图片框大小的改变与图形无关。"
    Label3 = "图形自动缩放以适应图像框大小。"
    Label4 = "图像框改变大小时图形维持原状。"
    Label2.Left = Label1.Left
    Label4.Left = Label3.Left
End Sub

Private Sub optEnlargeI_Click()     '放大图像框
    '根据复选框选中状态设置 Stretch 属性
    Image1.Stretch = Check2.Value
    Image1.Width = ImgW * 2
    Image1.Height = ImgH * 2
End Sub

Private Sub optEnlargeP_Click()     '放大图片框
    Picture1.Width = PicW * 2
```

```
    Picture1.Height = PicH * 2
    '根据复选框选中状态设置 AutoSize 属性
    Picture1.AutoSize = Check1.Value
End Sub

Private Sub optReduceI_Click()      '缩小图像框
    '根据复选框选中状态设置 Stretch 属性
    Image1.Stretch = Check2.Value
    Image1.Width = ImgW / 2
    Image1.Height = ImgH / 2
End Sub

Private Sub optReduceP_Click()      '缩小图片框
    Picture1.Width = PicW / 2
    Picture1.Height = PicH / 2
    '根据复选框选中状态设置 AutoSize 属性
    Picture1.AutoSize = Check1.Value
End Sub

Private Sub optRevertI_Click()      '图像框还原
    Image1.Width = ImgW
    Image1.Height = ImgH
End Sub

Private Sub optRevertP_Click()      '图片框还原
    Picture1.Width = PicW
    Picture1.Height = PicH
End Sub
```

2. 使用定时器控件制作秒表

（1）设计界面。在窗体上放置一个定时器 Timer1，再添加一个框架，在框架中添加一个标签和五个命令按钮。

（2）设置属性。定时器的属性采用默认值。窗体和其他控件的属性设置如表 4-2 所示。

表 4-2 窗体和控件属性

对象	名称	属性名	属性值	对象	名称	属性名	属性值
CommandButton	cmdStart	Caption	开始	Label	Label1	BorderStyle	1
	cmdPause	Caption	暂停			BackColor	黑色
	cmdContinue	Caption	继续			ForeColor	浅绿色
	cmdStop	Caption	停止	Frame	Frame1	Appearance	0-Flat
	cmdReset	Caption	重置			BackColor	灰色
Form	frmTimer	Caption	秒表			Caption	秒表

（3）编写代码。制作秒表的几个关键环节是：

1）记录开始计时的时间，可以通过调用 Visual Basic 内部函数 Timer 为变量赋值来实现。该函数返回从午夜零点开始至当前时刻的总秒数（Single 型数据，精度为 7 位）。

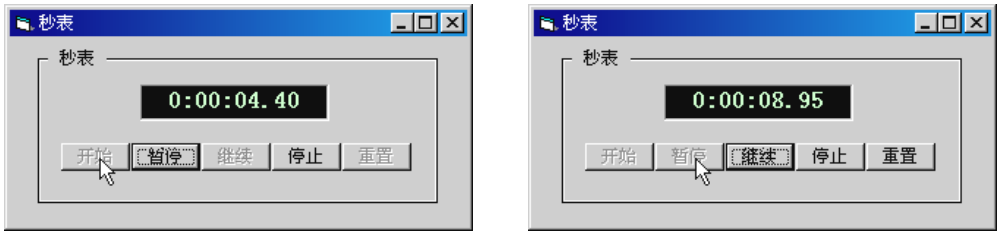
2）计算开始计时至当前时刻的时间差，用 Timer 减去开始计时的时刻即可获得该时间差。

3）在系统允许的最短时间间隔内将时间差以“时:分:秒.xx”的形式显示。适当设置定时器控件的 Interval 属性，在定时器的 Timer 事件中将时间差总秒数转换为时、分、秒，并调用 Format 函数以特定的时间格式显示。

4）为避免发生误操作，应该在单击某一按钮后，对其他按钮的有效性（Enabled 属性）进行设置。

为完成上述功能，需要设置若干变量，用于存储和计算有关的时间数据。具体功能的实现可以参考后面的程序代码。

程序运行结果如图 4.2 所示。



（a）计时状态　　（b）暂停状态

图 4.2　用定时器制作秒表

程序代码如下：

```
Option Explicit
'定义用于存放时、分、秒、总秒数的变量
Dim strH As String, strM As String
Dim strS As String, strSs As String
Dim sngT As Single
Dim intT As Integer
Dim sngStart As Single

Private Sub cmdContinue_Click()      '继续
    Timer1.Enabled = True            '定时器有效
    '“开始”、“继续”、“重置”按钮无效，“暂停”、“停止”按钮有效
    cmdStart.Enabled = False
    cmdPause.Enabled = True
    cmdContinue.Enabled = False
    cmdStop.Enabled = True
    cmdReset.Enabled = False
```

```
End Sub

Private Sub cmdPause_Click()    '暂停
    Timer1.Enabled = False      '定时器无效
    '“继续”、“停止”、“重置”按钮有效，“暂停”按钮无效
    cmdContinue.Enabled = True
    cmdStop.Enabled = True
    cmdReset.Enabled = True
    cmdPause.Enabled = False
End Sub

Private Sub cmdReset_Click()    '重置
    Form_Load                   '执行窗体加载过程中的各初始化语句
End Sub

Private Sub cmdStart_Click()    '开始
    sngStart = Timer     '调用 Timer 函数返回午夜以来总秒数（Single）作计时初值
    cmdContinue_Click    '执行“继续”按钮单击事件过程中的各语句
End Sub

Private Sub cmdStop_Click()   '停止
    Timer1.Enabled = False    '关闭定时器
    '“重置”按钮有效，其他按钮无效
    cmdStart.Enabled = False
    cmdPause.Enabled = False
    cmdContinue.Enabled = False
    cmdStop.Enabled = False
    cmdReset.Enabled = True
End Sub

Private Sub Form_Load()     '窗体加载，各控件初始化
    '定时器时间间隔定为 10ms，但是由于系统限制
    '在 Windows 9x 下，实际最短间隔仅能达到 1/18 秒（约 55ms）
    '在 Windows 2000 下，实际最短间隔可达 10ms
    Timer1.Interval = 10
    Timer1.Enabled = False  '定时器无效
    Label1.Caption = "0:00:00.00"
    '“开始”按钮有效，其他按钮无效
    cmdStart.Enabled = True
    cmdPause.Enabled = False
    cmdContinue.Enabled = False
    cmdStop.Enabled = False
    cmdReset.Enabled = False
End Sub
```

```
Private Sub Timer1_Timer()        '定时器事件
    '根据定时器控件 Interval 属性设置的时间间隔
    '将计时开始后度过的总秒数换算为时、分、秒（取 2 位小数）显示
    sngT = Timer - sngStart       '计时开始后的总秒数（7 位精度）
    strSs = Format(sngT * 100 Mod 100, "00")     '取小数点右侧 2 位
    intT = Int(sngT)                             '总秒数取整
    strS = Format(intT Mod 60, "00.")            '秒
    strM = Format(intT \ 60 Mod 60, "00:")       '分
    strH = Format(intT \ 3600, "0:")             '时
    Label1.Caption = strH & strM & strS & strSs '显示
End Sub
```

3. 列表框及组合框

用列表框和组合框实现员工概况的输入及显示。

（1）设计界面。在窗体上添加一个列表框、两个组合框、两个文本框、四个命令按钮、四个标签。界面布局如图 4.3 所示。

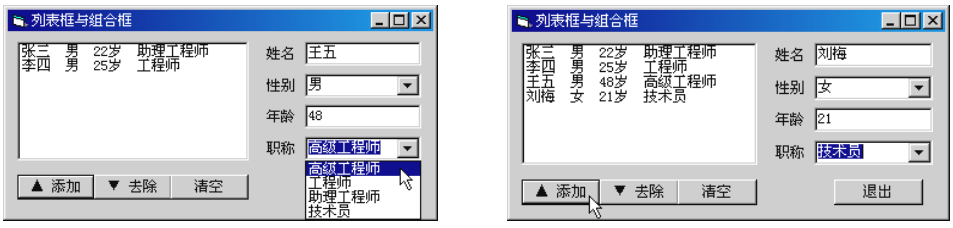
（2）设置属性。列表框属性采用默认值，四个标签的 Caption 属性分别设为“姓名”、“性别”、“年龄”和“职称”。其他控件的属性设置如表 4-3 所示。

表 4-3　控件属性

对象	名称	属性名	属性值	对象	名称	属性名	属性值
ComboBox	cboSex	Text		CommandButton	cmdAdd	Caption	▲添加
	cboPost	Text			cmdRemove	Caption	▼去除
TextBox	txtName	Text			cmdClear	Caption	清空
	txtAge	Text			cmdEnd	Caption	退出

（3）编写代码。在窗体的 Load 事件中用组合框的 AddItem 方法对职称和性别组合框进行初始化。在“添加”、“去除”和“清空”按钮的单击事件中分别调用列表框的 AddItem、RemoveItem 和 Clear 方法实现相应的功能。具体功能的实现可以参考后面的程序代码。

程序运行结果如图 4.3 所示。



（a）设置项目　　（b）添加项目

图 4.3　列表框及组合框

程序代码如下：

```
Option Explicit
Private Sub cmdAdd_Click()      '添加
   If Trim$(txtName.Text) = "" Or Trim$(txtAge.Text) = "" Then Exit Sub
   Dim strAdd As String
   '将在文本框和组合框中输入或选择的内容插入空格后连接成一个字符串
   strAdd = Trim$(txtName.Text) & "  " & cboSex.Text _
      & "  " & Trim$(txtAge.Text) & "岁  " & cboPost.Text
   List1.AddItem strAdd         '在列表框中添加项目
End Sub

Private Sub cmdClear_Click()    '清空
   List1.Clear     '清空列表框
End Sub

Private Sub cmdExit_Click()     '退出
   Unload Me
End Sub

Private Sub cmdRemove_Click()   '去除
   '去除列表框中选定的项目
   If List1.ListIndex <> -1 Then List1.RemoveItem List1.ListIndex
End Sub

Private Sub Form_Load()
   '窗体加载时用组合框的 AddItem 方法对职称和性别组合框进行初始化
   '对一个对象执行多个动作可以使用 With…End With 语句简化代码
   With cboPost
      .AddItem "高级工程师"
      .AddItem "工程师"
      .AddItem "助理工程师"
      .AddItem "技术员"
      .ListIndex = 0
   End With
   cboSex.AddItem "男"
   cboSex.AddItem "女"
   cboSex.ListIndex = 0
End Sub
```

4. 多窗体编程

（1）设计主窗体界面。新建工程，将 Form1 改名为 frmMain，设窗体的 Caption 属性为“主窗体”。在窗体上添加四个命令按钮，将按钮的 Caption 属性均设为空，Style 属性均设为 1-Graphical，分别为每个按钮的 Picture 属性设置一幅图片。添加四个标签，分别为每个按钮进行文字说明。界面布局如图 4.4（a）所示。

（2）添加窗体。单击工具栏中的“添加窗体”按钮，在“添加窗体”对话框中选择“现

存”选项卡，将在本实验前三个工程中建立的窗体文件添加到本工程中。

注意：在一个工程中不允许有同名（Name）的窗体，因此应该在添加窗体前将原工程中的默认窗体名称 Form1 改为其他名称。

（3）编写代码。为主窗体中的各命令按钮的单击事件编写代码。显示其他窗体的语句格式为：

```
窗体名称.Show
```

注意：除主窗体外，其他窗体的代码中均不应出现 End 语句，否则将结束程序。

程序运行效果如图 4.4 所示。

（a）主窗体　　（b）显示多窗体

图 4.4　多窗体

主窗体的程序代码如下：

```
Option Explicit

Private Sub cmdEnd_Click()          '结束
    End
End Sub

Private Sub cmdList_Click()         '显示“列表框与组合框”窗体
    frmLstCbo.Show
End Sub

Private Sub cmdPic_Click()          '显示“图片框与图像框”窗体
    frmPicImg.Show
End Sub

Private Sub cmdTimer_Click()        '显示“秒表”窗体
    frmTimer.Show
End Sub
```

实验 5　选择结构程序设计

一、实验目的

1．掌握逻辑表达式的正确书写形式。

2．掌握 If…Then 语句（单分支）、If…Then…Else 语句（双分支）、If…Then…ElseIf 语句（多分支）以及 If 语句嵌套程序设计。

3．掌握 Select Case 语句程序设计。

二、实验内容

1．用 If…Then 语句及其嵌套实现三个数字的排序。

2．用 If…Then、If…Then…Else 语句检查用户名和密码，进行用户登录检测。

3．制作一个摇奖机，用 If…Then…Else 语句使其中的摇奖按钮成为摇奖机的开关按钮。

4．综合运用 If…Then、If…Then…Else、If…Then…ElseIf 语句和 Select Case 语句编制算术考试程序。

三、实验步骤及指导

1．三值排序

程序运行时，在三个文本框中输入数字，单击“排序”按钮后按从大到小排序。

（1）设计界面并设置属性。在窗体上放置三个文本框，Text 属性均设为空。添加四个标签，Caption 属性分别设为“请输入三个数字：”、“x”、“y”、“z”。添加三个命令按钮，Caption 属性分别为“排序”、“清除”、“退出”。界面布局如图 5.1 所示。

（2）编写代码。在“排序”按钮的单击事件中，将三个文本框中的数字分别赋予三个变量 x、y、z，用 If 语句判断其大小，根据排序要求确定变量中的数据是否需要交换。若需要交换，则借助中间变量 t 进行。这是两个变量进行数据交换最常用的算法。例如，以下语句可以完成 x 和 y 之间的数据交换：

```
t = x: x = y: y = t
```

排序完成后，用窗体的 Print 方法在窗体上显示排序结果。

程序运行结果如图 5.1 所示。

程序代码如下：

```
Option Explicit

Private Sub Command1_Click()
```

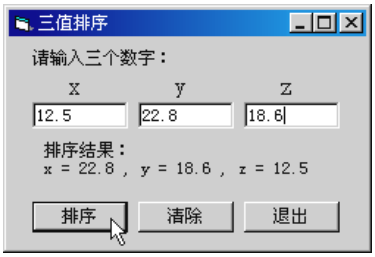


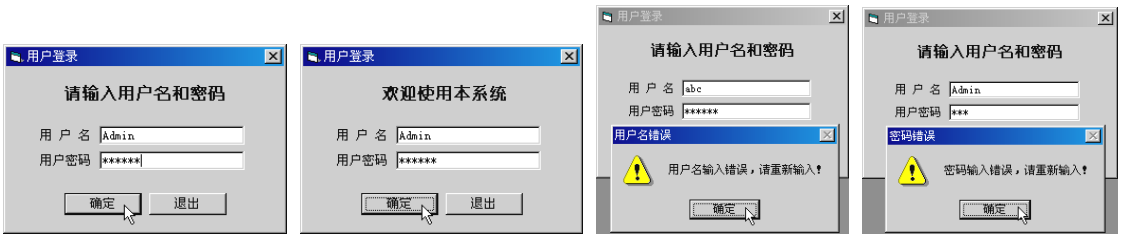
图 5.1　三值排序

```
    'x、y、z 用于存放在文本框中输入的数字
    Dim x As Single, y As Single, z As Single
    Dim t As Single              't 作为中间变量，用于 x、y、z 之间的数据交换
    x = Val(Text1.Text)          '将文本框中的数字分别赋予变量
    y = Val(Text2.Text)
    z = Val(Text3.Text)
    If x < y Then                '若 x < y， x 与 y 交换，使 x>y
       t = x: x = y: y = t
    End If
    If y < z Then                '若 y < z， y 与 z 交换，使 y>z
       t = y: y = z: z = t
       If x < y Then             '再判断 x 与 y
          t = x: x = y: y = t
       End If
    End If
    Cls
    CurrentY = ScaleHeight / 2   '为 Print 方法设置垂直坐标（窗体内部高度的 1/2）
    Print Tab(5); "排序结果："; '显示结果
    Print Tab(5); "x ="; x; ", y ="; y; ", z ="; z
End Sub

Private Sub Command2_Click()     '结束
    End
End Sub

Private Sub Command3_Click()     '清除
    Cls                          '清除窗体上的打印文字或图形
    Text1.Text = ""              '清空各文本框
    Text2.Text = ""
    Text3.Text = ""
    Text1.SetFocus               '设置焦点
End Sub

Private Sub Form_Load()          '窗体加载
    Caption = "三值排序"
```

```
    Command1.Default = True        '回车键默认按钮为“排序”按钮
    Command2.Cancel = True         'Esc 键默认按钮为“退出”按钮
End Sub
```

2. 用户登录检测

在各种管理信息系统的用户登录界面，一般需要进行用户名和密码的双重检测，并且规定了出错的最多次数。本实验项目要求设计一个用户登录检测程序，若用户输入的用户名和密码均无误，显示“欢迎使用本系统”，否则提示用户名或密码错误，请用户重新输入。若用户名或密码连续三次输入错误，则提示“对不起，您不是本系统的合法用户”。按照信息服务用户管理的常规，应该先对用户名进行检测（用户名具有惟一性），若用户名输入正确，则不再要求用户重复输入用户名，仅检查密码即可，这样做也符合界面友好的原则。程序运行效果如图 5.2 所示。



（a）输入用户名和密码　　（b）登录成功　　（c）用户名错误　　（d）密码错误

图 5.2　用户登录检测

（1）设计界面并设置属性。在窗体上放置两个文本框，分别命名为 txtUserID 和 txtPassword，Text 属性均设为空。添加三个标签，Caption 属性分别设为“请输入用户名和密码”、“用户名”、“用户密码”。添加两个命令按钮，Caption 属性分别为“确定”和“退出”。界面布局如图 5.2（a）所示。

（2）编写代码。在“确定”按钮的单击事件中，检测用户在文本框中输入的用户名和密码。用 If 语句判断其正误，根据判断结果显示相应的信息。若用户输入有误，除提示出错信息外，应该将焦点返回出错的文本框，并且反相显示出错内容。

程序代码如下：

```
Option Explicit
Private Sub cmdExit_Click()      '退出
    Unload Me
End Sub

Private Sub cmdOk_Click()        '“确定”按钮
    Static intErrUser As Integer        '静态变量累加出错次数
    Static intErrPass As Integer
    '检查用户名
    If UCase$(Trim$(txtUserID.Text)) = "ADMIN" Then      '若用户名正确
```

```
        '检查密码
        If Trim$(txtPassword.Text) = "123456" Then       '若密码正确
            Label1.Caption = "欢迎使用本系统"
            Label1.Left = (Me.Width - Label1.Width) / 2
        Else                                  '若密码错误
            intErrPass = intErrPass + 1   '错误数+1
            If intErrPass = 3 Then        '若出错 3 次，退出系统
                MsgBox "对不起，您不是本系统的合法用户。", vbInformation, "提示"
                Unload Me
            Else                          '若出错不足 3 次，重新输入
                MsgBox "密码输入错误，请重新输入！", vbExclamation, "密码错误"
                With txtPassword          '焦点返回密码框
                    .SelStart = 0
                    .SelLength = Len(.Text)
                    .SetFocus
                End With
            End If
        End If
    Else                                  '若用户名错误
        intErrUser = intErrUser + 1   '错误数+1
        If intErrUser = 3 Then        '若出错 3 次，退出系统
            MsgBox "对不起，您不是本系统的合法用户。", vbInformation, "提示"
            Unload Me
        Else                          '若出错不足 3 次，重新输入
            MsgBox "用户名输入错误，请重新输入！", vbExclamation, "用户名错误"
            With txtUserID            '焦点返回用户框
                .SelStart = 0
                .SelLength = Len(.Text)
                .SetFocus
            End With
        End If
    End If
End Sub
```

3. 制作摇奖机

用 Timer（定时器）结合其他控件制作一个摇奖机，要求“开始摇奖”和“抽奖”（抽取中奖号码）的功能通过一个命令按钮实现。当按钮的标题为“开始摇奖”时，用户在文本框中输入奖券号码，单击该按钮后，程序以定时器允许的最短时间间隔快速产生随机号码（6 位数），显示在标签控件中，同时将按钮标题改为“抽奖”。当用户再次单击此按钮时（抽奖），停止摇奖，显示中奖号码并判断用户是否中奖，同时将按钮标题改为“开始摇奖”。

（1）设计界面并设置属性。在窗体上放置一个 Timer 控件，一个文本框，两个命令按钮，Command1 的 Caption 属性为“开始摇奖”，Command2 的 Caption 属性为“退出”。添加四个标签，Label1 用于显示中奖号码，可以适当设置其背景色、前景色和边框样式，使之更加醒

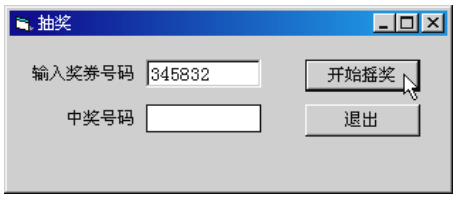
目，另外三个标签用于简要说明和显示中奖信息。

（2）编写代码。程序代码主要是处理 Command1 的单击事件和定时器控件的 Timer 事件。

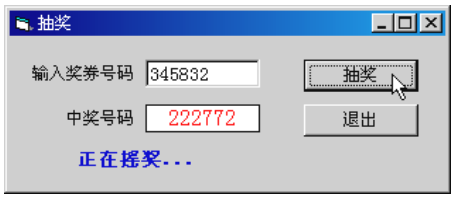
按钮 Command1“一钮两用”，相当于一个开关按钮，在“开始摇奖”和“抽奖”两种状态间切换。单击该按钮时，用 If 语句根据按钮 Caption 属性的当前值决定如何操作。如果按钮 Caption 属性的当前值为“开始摇奖”，则启动定时器产生随机号码，并将按钮的 Caption 属性改为“抽奖”；否则，关闭定时器，显示中奖号码并判断用户是否中奖，同时将按钮的 Caption 属性改为“开始摇奖”。

随机号码通过 Rnd 函数在定时器控件的 Timer 事件中产生，以 6 位数的形式在标签 Label1 中显示。

程序运行效果如图 5.3 所示。请注意（b）图中按钮标题与其他各图不同。



（a）输入号码



（b）正在摇奖

（c）抽奖结果 1

（d）抽奖结果 2

图 5.3　摇奖机

程序代码如下：

```
Option Explicit
Private Sub Command1_Click()      '“开始摇奖”/“抽奖”按钮单击事件
    If Command1.Caption = "开始摇奖" Then  '开始摇奖
        If Trim$(Text1) = "" Then          '如果未输入奖券号码
            MsgBox "请输入奖券号码。", vbInformation, "提示"
            Text1.SetFocus
            Exit Sub
        End If
        '按钮标题更迭，再次单击此按钮时，将执行 Else 后的语句
        Command1.Caption = "抽奖"
        Label4 = "正在摇奖..."
        Randomize     '初始化随机数发生器
        Timer1.Enabled = True   '启动定时器
```

```
      Else      '若按钮 Caption =“抽奖”
        Timer1.Enabled = False  '关闭定时器
        Command1.Caption = "开始摇奖"
        If Trim$(Text1) = Label1.Caption Then
           Label4 = "恭喜! 您中奖了! "
        Else
           Label4 = "谢谢您的参与! "
        End If
    End If
End Sub

Private Sub Command2_Click()   '退出
    Unload Me
End Sub

Private Sub Form_Load()
    '定时器时间间隔定为 10ms，但是由于系统限制
    '在 Windows 9x 下，实际最短间隔仅能达到 1/18 秒（约 55ms）
    '在 Windows 2000 下，实际最短间隔可达 10ms
    Timer1.Interval = 10
    Timer1.Enabled = False       '定时器无效
    Label1.Caption = ""
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Timer1.Enabled = False
End Sub

Private Sub Timer1_Timer()      '定时器事件
    Label1 = Format(Int(Rnd * 999999 + 1), "000000")     '显示 6 位摇奖号码
End Sub
```

4. 算术考试

设计一个小学低年级四则运算算术考试程序。程序中各种运算的操作数均为整数，其中加法、减法和乘法运算的操作数以及除法运算中除数的取值范围为 1～10，被除数为 1～20。操作数和运算符（＋、－、×、÷）均随机产生。减法运算的结果不应为负数，除法运算的结果应为整数。程序出题后，考生在文本框中输入答案并按回车键确认，由程序判断正误并显示结果（√、×）。当出题总数达 10 道题或单击“计分”按钮时，显示考试成绩，并根据成绩显示不同的鼓励信息。单击“重新开始”按钮时重新出题。

（1）设计界面并设置属性。在窗体上放置三个标签，Label1 用于显示题目，将其 Caption 属性设为空，背景色为白色，边框样式（BorderStyle）为 1-Fixed Single，使其看上去像一个文本框。另外两个标签用于对控件的说明。添加一个文本框，Text 属性为空。添加一个图片框，

背景色为白色，用于显示答题结果。添加三个命令按钮，Caption 属性分别为“计分”、“重新开始”和“退出”。

（2）编写代码。程序中信息的基本流程为：

出题→考生输入答案→判断正误并显示结果→统计并显示得分。

出题在Form_Load事件过程中完成。每道题用随机函数Rnd和取整函数Int产生范围为1～10 的两个随机整数作为操作数，若为减法或除法，应将大数作为被减（除）数。为避免生成不能整除的除法题目，可以进行以下处理：

```
If intN1 Mod intN2 Then intN1 = (intN1 \ intN2 + 1) * intN2
```

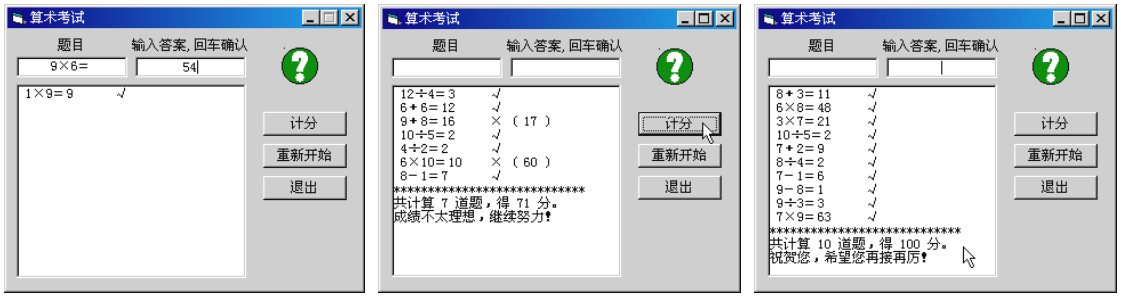
上面语句中的 intN1 为被除数，intN2 为除数，括号中的+1 可以避免被除数与除数相等的题目生成过多，还可以使被除数的取值在 20 以内。

加减乘除四种运算也是随机的，用范围为 1～4 的随机整数作为运算符代码，分别代表＋、－、×、÷。

判断正误在考生输入答案并按回车键后进行，因此，相关代码应置于文本框的按键事件过程中（如 KeyPress、KeyDown、KeyUp）。首先应判断考生是否按了回车键，然后将考生输入的答案与标准答案比较，输出判断结果。同时，分别累加考生答对和答错的题数，以备统计得分之用。考生每做完一题（按回车键），调用 Form_Load 事件过程生成下一题。

统计得分在“计分”按钮的单击事件过程中处理，答对题数/总题数*100 即为分数。

程序界面和运行效果如图 5.4 所示。



（a）出题和输入答案　　（b）单击“计分”按钮　　（c）出题总数达到 10 道题

图 5.4　算术考试

程序代码如下：

```
Option Explicit
'模块级变量存放计算结果和正误题数
Dim intResult As Integer
Dim intOK As Integer
Dim intErr As Integer

Private Sub Command1_Click()    '计分按钮
   Dim intTotal As Integer
```

```
    Dim intGrade As Integer
    intTotal = intOK + intErr   '总题数
    If intTotal = 0 Then
       MsgBox "您还没有答题，不能计分。", vbInformation, " 提示"
       Text1.SetFocus
       Exit Sub
    End If
    intGrade = Int(intOK / intTotal * 100)      '得分
    Label1 = ""
    Picture1.Print String(28, "*")
    Picture1.Print "共计算 " & intTotal & " 道题，"; _
        "得 " & intGrade & " 分。"
    '根据得分情况显示不同内容
    Select Case intGrade
       Case 100
          Picture1.Print "祝贺您，希望您再接再厉！"
       Case 90 To 99
          Picture1.Print "再加把劲，争取得 100 分！"
       Case 80 To 89
          Picture1.Print "成绩不错，但还需努力！"
       Case 60 To 79
          Picture1.Print "成绩不太理想，继续努力！"
       Case Else
          Picture1.Print "不要灰心，刻苦学习！"
    End Select
End Sub

Private Sub Command2_Click()        '重新开始
    Label1 = ""
    Text1 = ""
    Picture1.Cls
    Text1.SetFocus
    intOK = 0
    intErr = 0
    Form_Load      '调用 Form_Load 事件过程
End Sub

Private Sub Command3_Click()    '退出
    End
End Sub

Private Sub Form_Load()
    Dim intN1 As Integer, intN2 As Integer  '操作数
    Dim intOp As Integer, strOp As String   '操作代码、操作符
```

```
    Dim intTmp As Integer        '用于两数交换
    Randomize
    intN1 = Int(10 * Rnd + 1)    '随机取操作数和操作符代码
    intN2 = Int(10 * Rnd + 1)
    intOp = Int(4 * Rnd + 1)
    '若为减法或除法，将大数置前
    If intN1 < intN2 And intOp Mod 2 = 0 Then
       intTmp = intN1
       intN1 = intN2
       intN2 = intTmp
    End If
    Select Case intOp    '四则运算
       Case 1
         intResult = intN1 + intN2
         strOp = "+"     '将操作代码转换为操作符
       Case 2
         intResult = intN1 - intN2
         strOp = "－"
       Case 3
         intResult = intN1 * intN2
         strOp = "×"
       Case 4
         '若有余数，将被除数设为除数的整倍数，被除数＜20
         If intN1 Mod intN2 Then
           intN1 = (intN1 \ intN2 + 1) * intN2
         End If
         intResult = intN1 / intN2
         strOp = "÷"
    End Select
    Label1 = intN1 & strOp & intN2 & "＝"    '标签显示题目
End Sub

'按回车键后，在图片框显示结果
Private Sub Text1_KeyPress(KeyAscii As Integer)
     '若按键为回车键（ASCII 码=13），判断正误并显示
     If KeyAscii = 13 Then
       If Val(Text1) = intResult Then   '若答题正确
          Picture1.Print " "; Label1; Text1; Tab(15); "√"
          intOK = intOK + 1     '答对题数+1
       Else     '若答错
         Picture1.Print " ";Label1;Text1;Tab(15);"×";" (";intResult;")"
          intErr = intErr + 1   '答错题数+1
       End If
       Text1 = ""
```

```
        Text1.SetFocus
        If intOK + intErr = 10 Then  '回答 10 道题后，直接显示计分结果
            Command1_Click    '调用“计分”按钮单击事件过程
        Else      '若不足 10 道题
            Call Form_Load    '执行窗体加载事件过程中各语句生成下一题
        End If
    '若按键非数字键或回删键（ASCII 码=8），则取消按键
    ElseIf Not IsNumeric(Chr(KeyAscii)) And KeyAscii <> 8 Then
        KeyAscii = 0    '将 KeyAscii 参数设为 0 的作用是使本次按键无效
    End If
End Sub
```

实验 6　循环结构程序设计

一、实验目的

1．掌握 For…Next 循环的使用，正确使用循环变量控制 For…Next 循环的起始和结束。

2．掌握 While…Wend 循环的使用。

3．熟悉 Do…Loop 循环的使用。

4．掌握多重循环的应用。

二、实验内容

1．用 For…Next 循环制作可打印的 ASCII 码字符对照表

2．用 For…Next 循环及其嵌套制作九九乘法表，使其以三种方式（全部、下三角、上三角）显示。

3．用 Do…Loop 循环限制用户输入有效数据。

4．用 While…Wend 循环编制一个计算人口增长数据的程序。

三、实验步骤及指导

1．制作 ASCII 码对照表

ASCII 码（美国标准信息交换码）是 7 位二进制字符集，用来表示标准美制键盘上的字母、符号以及控制字符。其中，可打印字符的编码值范围为 32～126（32=空格）。利用 Chr 函数可以将字符代码转换为对应的字符。

（1）设计界面。程序的界面很简单，窗体上无任何控件。将窗体的背景色设为白色，Caption 属性设为“ASCII 码表”。

（2）编写代码。由于可打印字符的 ASCII 码对照表具有明确的起止范围，因此，非常适于用 For…Next 循环制作。单击窗体时，在循环中用 Print 方法将 ASCII 字符及其代码直接显示在窗体上，格式为“字符=字符代码”，每行显示 8 个字符。

程序运行效果如图 6.1 所示。

程序代码如下：

```
Option Explicit
Private Sub Form_Click()
    Dim intASC As Integer, i As Integer
    Cls
    Print
```

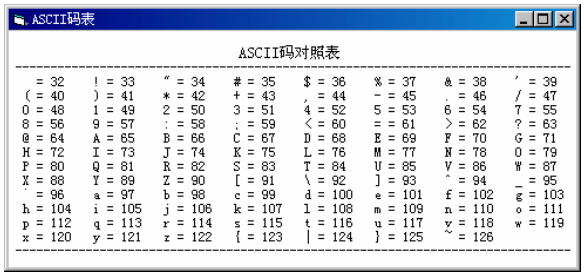


图 6.1　ASCII 码对照表

```
    Me.FontSize = 10
    Print Tab(29); "ASCII 码对照表"
    Me.FontSize = 9
    Print " "; String$(79, "-")      'String 函数返回指定数目的重复字符
    'ASCII 码 32-126 是可打印字符，Chr 函数将 ASCII 码转换为对应字符
    For intASC = 32 To 126           'intASC 为循环变量
       Print Tab(10 * i + 3); Chr(intASC); " ="; intASC;
       i = i + 1
       If i = 8 Then         '每行显示 8 个 ASCII 码
          i = 0
          Print
       End If
    Next intASC
    Print vbCr; " "; String$(79, "-")     'vbCr 代表回车
End Sub
```

2. 制作“九九乘法表”

用三种方式（全部、下三角、上三角）显示“九九乘法表”。

（1）设计界面并设置属性。在窗体上放置一个图片框，设背景色为白色。添加四个命令按钮，Caption 属性分别为“全部”、“下三角”、“上三角”和“结束”。

（2）编写代码。“九九乘法表”具有明显的规律，由 9 行 9 列等式组成，若以变量 i 代表行号，变量 j 代表列号，则所有等式均可表示为：i * j = 乘积。三种显示方式（全部、下三角、上三角）的区别在于行或列中等式的个数不同。在“全部”显示方式下，所有的行、列中均含有 9 个等式。在“下三角”方式下，每行等式的个数等于该行的行号。在“上三角”方式下，每列等式的个数等于该列的列号。对这种具有明显行列规律的问题，通常采用 For…Next 双重循环解决。设外循环的循环变量为行号，内循环的循环变量为列号，在内循环中输出一行中的各列，退出内循环后行号加 1，输出下一行。下面是程序变量定义以及 Caption 属性为“全部”的按钮单击事件过程的代码：

```
Option Explicit
Dim strS As String, i As Integer, j As Integer

Private Sub Command1_Click()    '全部
    Picture1.Cls
    Picture1.Print vbCr; Tab(41); "九九乘法表"
```

```
    Picture1.Print " " & String$(89, "-")
    For i = 1 To 9          '外循环变量为乘法表的“行”
        For j = 1 To 9      '内循环变量为乘法表的“列”
            strS = i & "×" & j & "=" & i * j              '行列相乘
            Picture1.Print Tab((j - 1) * 10 + 3); strS;   '显示
        Next j
        Picture1.Print
    Next i
    Picture1.Print " " & String$(89, "-")
End Sub
```

另外两种显示方式只需将上述代码中第 8 行的“`For j = 1 To 9`”语句进行如下改动：

“下三角”：For j = 1 To i　　'将 9 改为 i

“上三角”：For j = i To 9　　'将 1 改为 i

三种显示方式分别通过单击相应按钮实现。

程序运行界面如图 6.2 所示。

（a）全部

（b）下三角

（c）上三角

图 6.2　九九乘法表

3. 限制用户输入有效数据

“鸡兔同笼”问题是已知鸡和兔的总只数 h 及总腿数 f，求鸡和兔各有多少只。该问题本身并不需要用循环结构解决（当然也可以用循环结构，请读者自己考虑算法），本例只是借“鸡兔同笼”问题练习用 Do…Loop 循环限制用户输入有效数据。程序运行时通过 InputBox 函数由用户依次输入已知条件，即鸡兔总只数 h 和总腿数 f。在输入数据时，应根据用户输入的鸡兔总数 h 限定总腿数 f 的有效范围，即：2h＜f＜4h，且 f 必须为偶数。若输入错误，通过循环控制重新输入。

（1）设计界面。将窗体背景色设为白色，在窗体上放置三个标签，BackStyle 属性均设为 0-Transparent，用于对鸡兔同笼问题进行简要说明。添加一个命令按钮，设其 Caption 属性为“我来出题”。

（2）编写代码。在按钮 Command1 的单击事件中，通过 InputBox 函数由用户输入鸡兔总数 h。将输入鸡兔总腿数 f 的 InputBox 函数放在 Do…Loop 循环中，若输入的数据有效，则通过 Exit Do 语句退出循环，否则报告错误，并在循环中继续调用 InputBox 函数，直至输入正确或单击对话框中的“取消”按钮为止。

程序运行界面如图 6.3 所示。

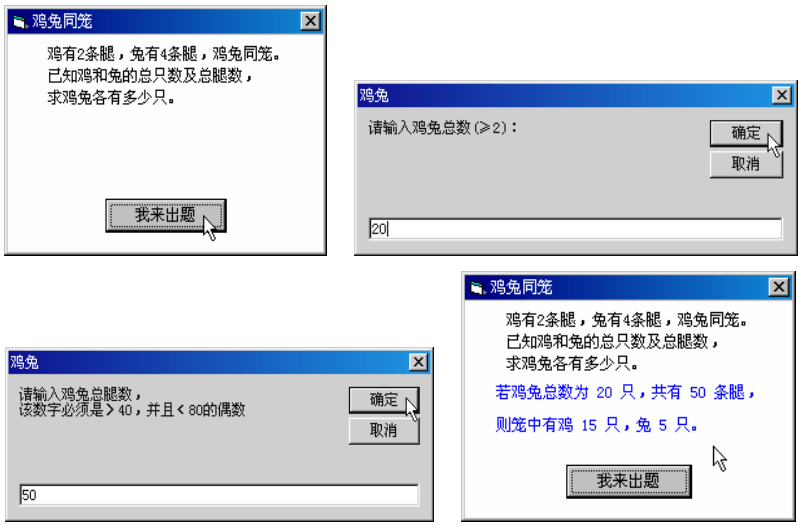


图 6.3　鸡兔同笼

程序代码如下：

```
Option Explicit
Private Sub Command1_Click()
    Dim C%, R%, h%, f%        '鸡数、兔数、总只数、总腿数变量
    h = Val(InputBox("请输入鸡兔总数（≥2）："))
    If h < 2 Then Exit Sub        '用户单击“取消”按钮或鸡兔总数<2，退出本过程
    Do      '无条件循环，在循环中设置退出循环的语句
```

```
        f = Val(InputBox("请输入鸡兔总腿数，" & vbCr _
            & "该数字必须是大于" & 2 * h _
            & "，并且小于" & 4 * h & "的偶数"))
        If f = 0 Then Exit Sub        '用户单击“取消”按钮退出本过程
        '输入错误时，重新输入，正确则退出循环
        If f <= 2 * h Or f >= 4 * h Or f Mod 2 Then
            MsgBox "输入错误，请重新输入！", vbCritical
        Else
            Exit Do    '退出循环
        End If
    Loop
    R = (f - 2 * h) / 2      '求兔数
    C = h - R     '鸡数
    Cls
    Me.CurrentY = Me.Height / 3      '为 Print 方法设置坐标
    Print Tab(5); "若鸡兔总数为"; h; "只，共有"; f; "条腿，"
    Print
    Print Tab(5); "则笼中有鸡"; C; "只，"; "兔"; R; "只。"
End Sub
```

4. 计算人口增长数据

据统计，2002 年末全国总人口为 12.8453 亿，人口自然增长率为 6.45‰。以这两项统计数字为基数，编制一个计算人口增长数据的程序，使之可以计算若干年后我国人口可能达到的数字，也可计算达到一定人口数所需的时间。

（1）设计界面。在窗体上放置一个框架，框架内添加两个单选按钮，用于选择计算项目；添加两个文本框，用于输入年数或人口数；添加两个标签，用于对计算项目进行简要说明。在窗体上添加一个图片框，背景色设为白色，用于显示计算结果。添加两个命令按钮，分别用于计算和退出。再添加一个标签，显示 2002 年末人口的统计数据。

（2）编写代码。人口增长数可以按以下公式计算：

$$P2=P1(1+r)^{y}$$

式中指数 y 为年数，P1 为人口基数，P2 为 y 年后将达到的人口数，r 为人口自然增长率。例如，以 2002 年末的统计数字为基数，计算 10 年后人口总数（亿）的 Visual Basic 表达式为：

```
12.8453*(1+0.00645)^10
```

达到一定人口数所需的时间（年数）可以根据人口增长公式用 Log 函数计算：

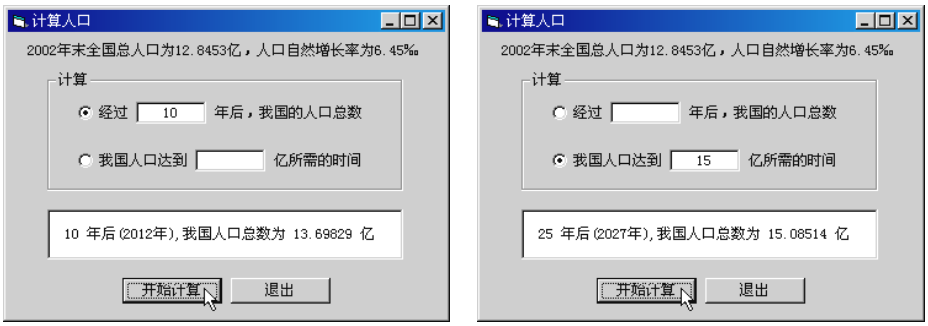
```
y = Log(P2 / P1) / Log(1 + r)
```

也可用 While…Wend 循环求得。例如，以下代码计算人口达到 15 亿所需的时间 y：

```
Dim P1 As Single, y As Single
P1 = 12.8453
y = 0
While P1 < 15
  P1 = P1 * 1.00645
  y = y + 1
```

```
Wend
```

本实验项目要求用 While…Wend 循环或 Do…Loop 循环计算达到一定人口数所需的时间。程序运行界面如图 6.4 所示。



（a）若干年后的人口数　　（b）达到一定人口数所需的时间

图 6.4　计算人口增长数据

完整的程序代码如下：

```
Option Explicit
Private Sub Command1_Click()      '开始计算
    Dim sngP1 As Single, sngP2 As Single     '人口变量
    Dim sngY1 As Single, sngY2 As Single     '年数变量
    sngP1 = 12.8453      '2002 年末的人口（亿）
    If Option1 Then      '如果选择计算若干年后的人口数
        If Val(Text1) <= 0 Then      '排除无效数据
            Picture1.Cls
            Picture1.Print "无效数据！"
            Exit Sub
        End If
        sngY1 = Val(Text1)
        sngP2 = sngP1 * 1.00645 ^ sngY1  '计算人口数（人口自然增长率为 6.45‰）
    Else     '若选择计算达到一定人口数所需的时间
        If Val(Text2) < sngP1 Then
            Picture1.Cls
            Picture1.Print "无效数据！"
            Exit Sub
        End If
        sngP2 = Val(Text2)
        sngY1 = 0
        While sngP1 < sngP2       '计算人口数达到 sngP2 亿所需的时间
            sngP1 = sngP1 * 1.00645
            sngY1 = sngY1 + 1
        Wend
        sngP2 = sngP1
```

```
    End If
    '处理年份
    If Int(sngY1) < sngY1 Then       '若用户输入带小数的年数
       sngY2 = Int(sngY1) + 1
    Else
       sngY2 = sngY1
    End If
    '在图片框显示计算结果
    Picture1.Cls
    Picture1.Print vbCr; Tab(2); sngY1; "年后(" & _
       2002 + sngY2; "年),我国人口总数为"; sngP2; "亿"
End Sub

Private Sub Command2_Click()        '退出
    End
End Sub

Private Sub Form_Load()
    Label1 = "2002 年末全国总人口为 12.8453 亿，人口自然增长率为 6.45‰"
End Sub

Private Sub Option1_Click()         '单击单选按钮时，焦点转至对应的文本框
    Text1.SetFocus
End Sub

Private Sub Option2_Click()
    Text2.SetFocus
End Sub

Private Sub Text1_Click()       '单击文本框时，选中对应的单选按钮
    Option1.Value = True
End Sub

Private Sub Text2_Click()
    Option2.Value = True
End Sub
```

实验 7　数组

一、实验目的

1．掌握一维数组、二维数组和可调（动态）数组的建立和使用方法。
2．掌握控件数组的建立和使用方法。

二、实验内容

1．制作一个将十进制整数转换为二进制、八进制和十六进制数的程序。
2．用含有 16 个元素的单选按钮控件数组设置文本框的字体颜色和背景色。
3．综合运用一维数组、二维数组、动态数组和控件数组的有关知识编写程序，要求能输入学生的学号、姓名、性别、年龄等个人简况，输入的学生人数不限，并可以按学号或姓名查询。

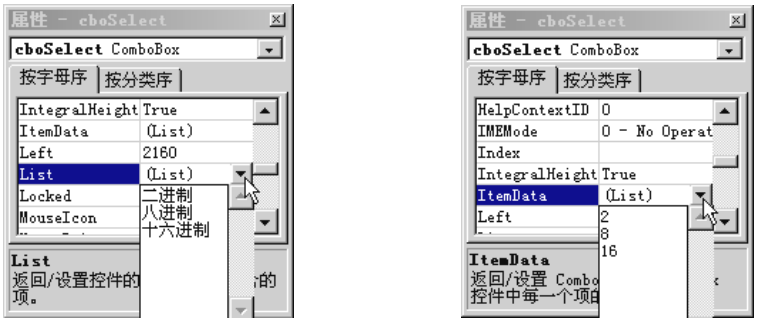
三、实验步骤及指导

1．数制转换

制作一个将十进制整数转换为二进制、八进制和十六进制数的程序。十进制整数转换为其他（N）进制数的基本方法是“N 除取余，逆排序”，即用 N 去除十进制整数，取其余数，再用 N 去除商，取余数，如此反复，直至商为零。将每次所得余数逆序排列，即为要转换的 N 进制数。在十六进制数中有“A”～“F”6 个特殊数码，与十进制数 10～15 相对应，为了便于转换，可以将“0”～“F”16 个数码存入一个下界为 0、上界为 15 的静态字符型数组中。转换时按照数组元素的下标取出对应的字符即可。该数组同样适用于向二进制和八进制的转换。

（1）设计界面并设置属性。在窗体上放置两个文本框，用于输入十进制数和显示转换结果，名称分别为 txtInput 和 txtResult，将 Text 属性均设为空。添加一个组合框，将名称改为 cboSelect。如图 7.1 所示，选择该控件属性窗口的 List 属性，单击右侧的下拉按钮，在弹出的下拉列表框中输入三项内容：“二进制”、“十进制”和“十六进制”（每项输入完成后按 Ctrl+Enter 组合键换行）；再选择属性窗口的 ItemData 属性，单击右侧的下拉按钮，在弹出的下拉列表框中输入三项内容：“2”、“8”和“16”，分别与 List 属性中的各项相对应。再添加两个命令按钮，Caption 属性分别为“转换”和“结束”。

说明：组合框的 ItemData 属性是一个长整型数的数组，它的项目数（元素个数）与控件的 List 属性（本质是字符串数组）的项目数相等。ItemData 属性中的各项与 List 属性中的各项一一对应。在本程序中，当用户选择待转换的数制（如“二进制”）时，可以直接利用 ItemData 属性中对应的数值进行计算。



（a）设置 List 属性　　（b）设置 ItemData 属性

图 7.1　设置组合框属性

（2）编写代码。定义一个模块级的字符串型静态数组 Char(15)，默认下界为 0。在窗体的 Form_Load 事件中将字符 0～9、A～F 赋予该数组的各元素，以备调用。

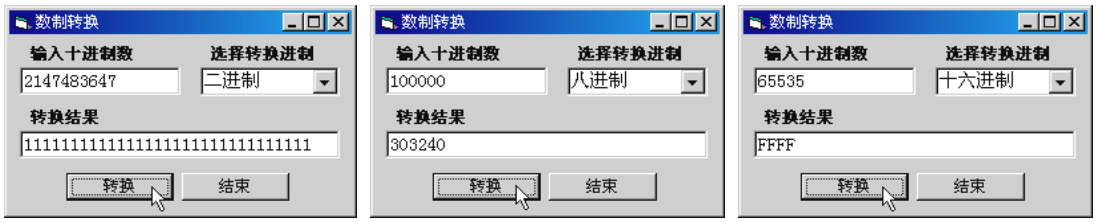
在组合框的单击事件中，将用户选择的进制通过组合框的 ItemData 属性获取进制基数存入变量以备计算。

在“转换”按钮的单击事件中，通过循环进行“N 除取余，逆排序”，将用户输入的十进制数转换为其他进制。在循环中，根据每次所得的余数，取 Char 数组中对应位置的“0～F”字符，即转换为特定进制的数码字符。将每次所得数码字符按以下方式存入字符串变量即可实现“逆排序”：

字符串变量 = 数码字符 & 字符串变量

注意：数码字符应该在“&”连接符之前。

程序运行效果如图 7.2 所示。



（a）十进制转二进制　　（b）十进制转八进制　　（c）十进制转十六进制

图 7.2　数制转换

程序代码如下：

```
Option Explicit
Dim n As Integer          '存数制
Dim Char(15) As String       '静态字符串数组存放 0～F 数字字符

Private Sub cboSelect_Click()  'cboSelect 为数制组合框
    'ItemData 为组合框中项目的代号，分别为 2、8、16
    'ListIndex 为组合框中项目的索引，代表当前被选中的项目
```

```
    '单击时将所选数制存入 n
    n = cboSelect.ItemData(cboSelect.ListIndex)
End Sub

Private Sub cmdEnd_Click()      '“结束”按钮
    Unload Me
End Sub

Private Sub cmdStart_Click()    '“转换”按钮
    On Error GoTo ErrInfo       '出错时（如果数字太大，溢出）转向 ErrInfo 语句
    Dim strResult As String     '存转换结果
    Dim lngNum As Long          '存待转换数字
    Dim r As Integer            '存余数
    lngNum = Val(txtInput)      '将输入框中的十进制数存入 lngNum
    Do Until lngNum = 0         '循环除 n 取余，直到原十进制数商=0 为止
        r = lngNum Mod n        '除 n 取余
        '根据余数，取 Char 数组中对应位置的 0～F 字符
        '转换为特定进制的数字符号
        '将余数逆序排列（注意“&”前后变量的位置）赋予 strResult
        strResult = Char(r) & strResult
        lngNum = lngNum \ n   '除 n 取整，准备下一次循环
    Loop
    txtResult = strResult    '显示结果
    Exit Sub     '关键语句，避免未出错时也执行后面的语句
'错误处理程序段（避免溢出）
ErrInfo:
    MsgBox "请输入 ≤2,147,483,647 的数字。", vbInformation, " 提示"
    With txtInput
        .SelStart = 0
        .SelLength = Len(.Text)
        .SetFocus
    End With
End Sub

Private Sub Form_Load()     '窗体加载
    Dim i As Integer
    '将 0～9 赋值给 Char 数组
    For i = 0 To 9
        Char(i) = i
    Next
    '将字符 A～F 赋值给 Char 数组元素(10)～(15)。65 是“A”的 ASCII 码
    For i = 0 To 5
        Char(10 + i) = Chr(65 + i)
    Next
```

```
    n = 2          '预设转换数制为二进制
    cboSelect.ListIndex = 0       '设组合框第 1 项（二进制）为默认选项
    cmdStart.Default = True       '“转换”按钮为回车默认按钮
End Sub

Private Sub txtInput_KeyPress(KeyAscii As Integer)
    '若按键非数字键或回删键，取消按键
    If Not IsNumeric(Chr(KeyAscii)) And KeyAscii <> 8 Then
       KeyAscii = 0
    End If
End Sub
```

2. 用单选按钮控件数组设置文本框的字体颜色和背景色

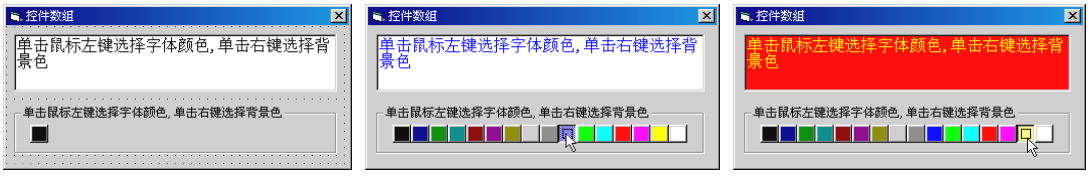
创建一个含有 16 个元素的单选按钮控件数组，代表 QBColor 函数可返回的 16 种颜色，用鼠标左键单击其中任一单选按钮，将文本框的字体颜色（即前景色）设置为对应颜色，右键单击时，将文本框的背景色设置为对应颜色。

（1）设计界面。在窗体上放置一个文本框和一个框架。在框架中添加一个单选按钮 Option1，将 Option1 的 Index 属性设为 0，Style 属性设为 1-Graphical，Caption 属性设为空，BackColor 设为黑色，Height 和 Width 均设为 270。界面设计如图 7.3（a）所示。

（2）编写代码。在窗体的 Form_Load 事件中，通过 For 循环用 Load 语句添加单选按钮控件数组元素（1）～（15），同时调用 QBColor 函数为新添加的单选按钮设置背景色（QBColor 函数的参数值与单选按钮的下标相等），新添加的单选按钮从左向右依次排列。注意将新的控件数组元素设为可见（Visible=True）。

在单选按钮 Option1 的 MouseDown 事件中，要用到由系统提供的该事件过程传送的两个参数：Button 和 Index。Button 参数用于判断用户按下的是哪一个鼠标键，Index 参数用于判断用户操作的是哪一个单选按钮。根据对这两个参数的判断，即可将选中的单选按钮的背景色赋值给文本框的前景色或背景色。

程序设计和运行界面如图 7.3 所示。



（a）设计时界面　　（b）运行界面 1　　（c）运行界面 2

图 7.3　控件数组

程序代码如下：

```
Option Explicit
Private Sub Form_Load()
    Dim i As Integer
```

```
    For i = 1 To 15
        Load Option1(i)        '加载下标为 i 的控件数组元素
        With Option1(i)        '设置控件数组元素属性
            .Left = Option1(i - 1).Left + .Width      '从左向右依次排列
            .BackColor = QBColor(i)        'QBColor 函数返回 16 种颜色之一
            .Visible = True
        End With
    Next
End Sub

'单选按钮的鼠标按下事件，可以区分按下的是左键还是右键
Private Sub Option1_MouseDown(Index As Integer, Button As Integer, _
        Shift As Integer, X As Single, Y As Single)
    'Button 参数传送的是引起该事件的鼠标键
    If Button = vbLeftButton Then        '若按下左键，设置文本框前景色
        Text1.ForeColor = Option1(Index).BackColor  'Index=被选中的单选按钮下标
    Else         '若非左键，设置文本框背景色
        Text1.BackColor = Option1(Index).BackColor
    End If
End Sub
```

3. 数组的综合应用

编写程序，要求能通过文本框控件数组输入学生的学号、姓名、性别、年龄等个人简况，输入的学生人数不限，并可以按学号或姓名查询。

分析：上述学生个人信息类似于数据库中的一个二维表格，每名学生的信息是一条“记录”（行），学号、姓名等是记录的“字段”（列），因此可以将学生信息存入一个二维数组中以备查询。由于要求输入的学生人数不限，即数组中“行”的上界是浮动的，所以应该将二维数组定义为动态数组，当增加新记录时，用 ReDim 语句改变数组“行”的上界。为了在改变数组大小时保留数组中的原有信息，需要在 ReDim 语句中使用 Preserve 关键字。按照习惯，一般是将二维数组中的第一维作为“行”，第二维作为“列”。但是，Visual Basic 规定，使用 Preserve 关键字保留数组原有信息时，只能对数组最后一维的大小重新定义。因此，需要改变通常的做法，即将第一维作为“列”，第二维作为“行”。

（1）设计界面并设置属性。在窗体上放置一个文本框 Text1，设 Text 属性为空。选定该文本框后单击工具栏上的“复制”按钮，再单击“粘贴”按钮，在弹出的对话框中单击“是”按钮，然后再单击两次“粘贴”按钮，即创建了含有 4 个元素的文本框（Text1）控件数组，用于输入学生信息。添加一个标签 Label1，按上述方法，创建含有 4 个元素的标签（Label1）控件数组。Label1(0)～Label1(3)的 Caption 属性分别为“学号”、“姓名”、“性别”、“年龄”，用于对文本框的说明。添加一个文本框 Text2，用于显示查询结果。添加 4 个命令按钮，Caption 属性分别为“输入”、“查询”、“全部”、“退出”。

（2）编写代码。在窗体的“通用-声明”部分声明一个字符串型动态数组 arStu( )（用于

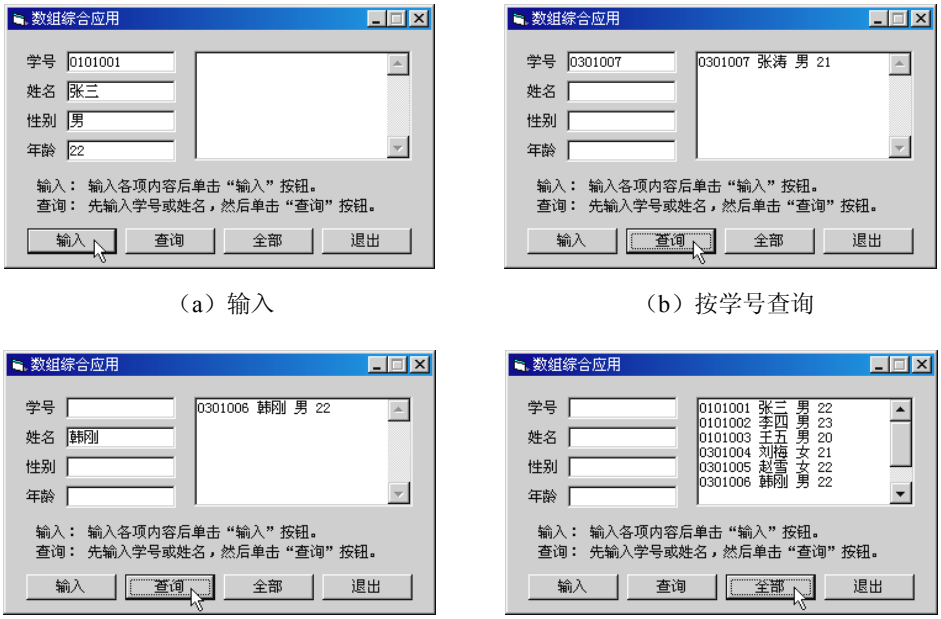
存储学生信息）和一个整型变量（用于存储已输入的学生数）。

在“输入”按钮的单击事件中，定义一个静态整型变量 i，用于累加学生数。通过循环检查各输入文本框，若均有内容，则令 i = i+1，用 ReDim Preserve 语句重新定义动态数组的第二维上界为 i（数组第一维上界始终为 3），即：ReDim Preserve arStu(3, i)。然后用 For 循环将文本框控件数组中的数据存入动态数组(文本框控件数组各元素的下标与动态数组各元素第一维下标相对应)。

在“查询”按钮的单击事件中，用 For 循环按第二维遍历动态数组元素，查找符合条件的学号或姓名。若找到，在 Text2 中显示，若未找到，用 MsgBox 函数给出提示。

在“全部”按钮的单击事件中，用 For 双重循环（外循环变量为第二维下标，内循环变量为第一维下标）将全部学生的信息显示在 Text2 中。

程序运行结果如图 7.4 所示。



（a）输入　　（b）按学号查询

（c）按姓名查询　　（d）全部显示

图 7.4　数组的综合应用

完整的程序代码如下：

```
Option Explicit
Dim arStu() As String          '动态数组
Dim intNum As Integer          '存记录数
Private Sub cmdAll_Click()       '“全部”按钮，显示全部记录
   If intNum = 0 Then Exit Sub      '若记录数为零，退出过程
   Dim i As Integer, j As Integer
   Dim strAll As String
```

```
    For i = 1 To intNum          'i 为数组第二维下标
       For j = 0 To 3            'j 为数组第一维下标
          '以 i 为行，j 为列，将数组元素存入变量
          strAll = strAll & arStu(j, i) & " "
       Next j
       strAll = strAll & vbCrLf      '行尾加回车换行符
    Next i
    Text2.Text = strAll           '在文本框中显示
End Sub
Private Sub cmdEnd_Click()        '“结束”按钮
    End
End Sub
Private Sub cmdInput_Click()     '“输入”按钮
    Static i As Integer     '静态变量存放记录数
    Dim j As Integer
    For j = 0 To 3        '检查各文本框是否为空
       If Trim$(Text1(j).Text) = "" Then     '若第(j)个文本框无内容
          '利用标签控件数组对应元素的 Caption 属性提示应输入哪一项内容
          MsgBox "请输入" & Label1(j).Caption, vbInformation, " 提示"
          Text1(j).SetFocus
          Exit Sub
       End If
    Next
    i = i + 1       '静态变量累加记录数
    ReDim Preserve arStu(3, i)        '根据记录数重新定义动态数组末维上界
    '将各文本框的数据存入数组，同时清空文本框以备输入下一个记录
    For j = 0 To 3
       arStu(j, i) = Trim$(Text1(j).Text)
       Text1(j).Text = ""
    Next
    intNum = i     '保存记录数
    Text1(0).SetFocus    '焦点返回学号文本框
End Sub
Private Sub cmdQuery_Click()         '“查询”按钮
    If intNum = 0 Then Exit Sub      '若记录数为零，退出过程
    Dim i As Integer, j As Integer
    Dim strQ As String
    If Trim$(Text1(0)) <> "" Then    '若已输入学号，则按学号查找
       For i = 1 To intNum
          '若找到，在查询结果文本框中显示记录内容并退出本过程
          If arStu(0, i) = Trim$(Text1(0)) Then
             For j = 0 To 3
                '以 i 为行，j 为列，将数组元素存入变量
                strQ = strQ & arStu(j, i) & " "
```

```
                Next
                Text2.Text = strQ      '显示
                Exit Sub      '找到退出
            End If
        Next
    ElseIf Trim$(Text1(1))<>""Then    '若未输入学号，但输入了姓名，则按姓名查找
        For i = 1 To intNum
            '若找到，在查询结果文本框中显示记录内容并退出本过程
            If arStu(1, i) = Trim$(Text1(1)) Then
                For j = 0 To 3
                    strQ = strQ & arStu(j, i) & " "
                Next
                Text2.Text = strQ
                Exit Sub     '找到退出
            End If
        Next
    Else    '若学号、姓名均未输入
        MsgBox "请先输入学号或姓名，再单击“查询”按钮。", vbInformation, " 提示"
        Text1(0).SetFocus
        Exit Sub
    End If
    '若未找到
    MsgBox "对不起，没有您要查询的学生。", vbInformation, " 查询结果"
End Sub
```

实验 8　过程

一、实验目的

1．掌握过程的定义和说明的方法。
2．掌握过程的参数传递方法及规则。
3．掌握过程的调用方法。
4．掌握鼠标事件过程和键盘事件过程的应用。

二、实验内容

1．编写一个函数过程（Function 过程），实现摄氏温标与华氏温标之间的相互转换。

2．编写一个用辗转相除法求两个数的最大公约数的子过程（Sub 过程），通过多次调用该子过程，求出多个数（超过两个数）的最大公约数。

3．综合运用子过程和键盘事件过程，编写一个扩展组合框功能的程序，使组合框能进行列表项自动匹配，能添加不重复的新项目。

三、实验步骤及指导

1．温标转换

编写一个函数过程（Function 过程），实现摄氏温标与华氏温标之间的相互转换。两种温标转换的公式如下：

华氏温度（℉）= 摄氏温度×9/5＋32

摄氏温度（℃）=(华氏温度－32)×5/9

（1）设计界面并设置属性。在窗体上放置一个文本框 Text1，设 Text 属性为空。添加 5 个标签，设 Label1 的 Caption 属性为空，BorderStyle 属性为 1-Fixed Single，背景色为白色；设 Label2 和 Label3 的 Caption 属性均为空；设 Label4 和 Label5 的 Caption 属性分别为“输入温度”和“转换温度”。添加两个命令按钮，Caption 属性分别为“摄氏→华氏”和“华氏→摄氏”。

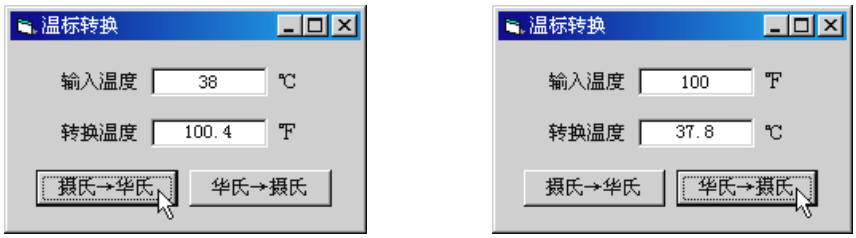
（2）编写代码。创建一个自定义函数 TransTh，用于两种温标的相互转换。该函数含有两个参数（形参），分别为单精度型和逻辑型，用于传送待转换温度和温标类型标志。函数的返回值为字符串型。在函数过程中，根据温标类型标志进行换算，将换算结果用 Format 函数保留一位小数后返回。

在两个命令按钮的单击事件中以文本框中的温度和温标类型标志作为实参调用 TransTh

函数。若转换类型为摄氏→华氏，则设温标类型标志为 True，否则为 False。调用函数后将返回值显示在 Label1 中，并根据温标转换类型，将 Label2 和 Label3 的 Caption 属性分别设为“℃”或“℉”。

此外，在文本框的键盘事件（KeyPress）过程中，需要对用户按键进行过滤，若按键不是数字键或回删键，则使按键无效。

程序运行界面如图 8.1 所示。



（a）摄氏→华氏　　（b）华氏→摄氏

图 8.1　温标转换

程序代码如下：

```
Option Explicit
Private Sub cmdCtoF_Click()      '摄氏→华氏
    If Trim$(Text1) = "" Then Text1 = "0"
    '调用自定义函数
    Label1.Caption = TransTh(Val(Text1.Text), True)
    Label2 = "℃"
    Label3 = "℉"
End Sub
Private Sub cmdFtoC_Click()      '华氏→摄氏
    If Trim$(Text1) = "" Then Text1 = "0"
    '调用自定义函数
    Label1.Caption = TransTh(Val(Text1.Text), False)
    Label2 = "℉"
    Label3 = "℃"
End Sub

Private Sub Form_Load()
    Label2 = ""
    Label3 = ""
End Sub
'自定义函数：摄氏温标与华氏温标相互转换
Private Function TransTh(sngT As Single, blnCtoF As Boolean) As String
    If blnCtoF Then
        TransTh = Format(sngT * 9 / 5 + 32, "0.#")    '摄氏→华氏
    Else
```

```
        TransTh = Format((sngT - 32) * 5 / 9, "0.#")      '华氏→摄氏
    End If
End Function
Private Sub Text1_KeyPress(KeyAscii As Integer)      '文本框键盘事件过程
    '若按键不是数字键或回删键，则取消按键
    If Not IsNumeric(Chr(KeyAscii)) And KeyAscii <> 8 Then
        KeyAscii = 0
    End If
End Sub
```

2. 求多个数的最大公约数

编写一个用辗转相除法求两个数的最大公约数的子过程（Sub 过程），通过多次调用该子过程，求出多个数（超过两个数）的最大公约数。

分析：对多个数求最大公约数时，先求出前两个数的最大公约数，将所得最大公约数与第三个数求最大公约数，依此类推。在计算过程中，只要出现最大公约数为 1，即不必再对后续的其他数求公约数。

辗转相除法是以两数中的小数作除数，大数作被除数，相除取余。若余数为零，则除数即为最大公约数；若余数不为零，则将除数改作被除数，余数改作除数，继续相除，直至余数得零为止。在最后一次相除时所用的除数（即最后一个不为零的余数）就是所求两数的最大公约数。整个过程可以用循环实现。

（1）设计界面并设置属性。在窗体上添加一个图片框 Picture1，设背景色为白色，AutoRedraw 属性为 True。添加三个命令按钮，Caption 属性分别为“开始”、“清除”和“退出”。

（2）编写代码。创建一个子过程 GCD，该子过程含有两个参数（形参）m 和 n，采用传址方式，均为 Long 型，用于传送拟求最大公约数的两个数。在子过程中始终以 n 作为除数，通过循环用辗转相除法求最大公约数，计算结束时，n 即为最大公约数。由于采用传址方式，形参 n 与对应的实参占用同一内存地址，实参值随形参改变，因此当子过程结束时，主调过程中与形参 n 对应的实参值就是求得的最大公约数。

在“开始”按钮的单击事件中，定义一个动态数组，用 InputBox 函数获取用户要求最大公约数的数字个数，以该数值定义动态数组的上界，再通过 For 循环，用 InputBox 函数将每个数存入数组，并在图片框中显示每个数。计算多个数的最大公约数时仍然采用 For 循环（循环次数＝数字个数−1），在循环中依次取出动态数组中的数字，调用 GCD 子过程计算。计算结束后将计算结果显示在图片框中。

程序运行结果如图 8.2 所示。

程序代码如下：

```
Option Explicit
'自定义子过程，参数传递为传址方式
Private Sub GCD(ByRef m As Long, ByRef n As Long)
    '用辗转相除法求两数的最大公约数
    Dim r As Long, t As Long
```

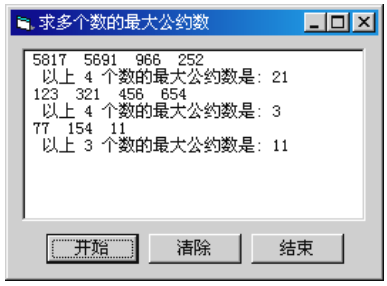


图 8.2　最大公约数

```
    If m < n Then t = m: m = n: n = t          'm 中存大数（被除数），n 为除数
    Do
        r = m Mod n                  '相除取余，r 为余数
        If r = 0 Then Exit Do        '若余数为 0，n 即为最大公约数，退出循环
        m = n                        '否则，除数改作被除数
        n = r                        '余数改作除数
    Loop
End Sub
Private Sub Command1_Click()     '开始
    Dim Ar() As Long    '动态数组用于存放拟求最大公约数的数字
    Dim n%, i%, n1&, m1&    '% = Integer, & = Long
    n = Val(InputBox("求几个数的最大公约数？"))
    If n < 2 Or n > 20 Then Exit Sub
    ReDim Ar(n)     '重新定义数组上界
    For i = 1 To n  '输入 n 个数，为求其最大公约数做准备
        '将第 i 个数存入数组
        Ar(i) = Val(InputBox("输入第" & i & "个数："))
        If Ar(i) <= 0 Then          '若输入数≤0 或单击“取消”按钮
            Picture1.Cls
            Exit Sub
        End If
        Picture1.Print Ar(i);     '在图片框中显示输入的数
        '超过图片框宽度的 4/5 时换行
        If Picture1.CurrentX > Picture1.Width * 0.8 Then Picture1.Print
    Next
    Picture1.Print
    n1 = Ar(1)                  '将第 1 个数存入 n1
    For i = 2 To n              'n 个数调用 n-1 次 GCD 过程求最大公约数
        m1 = Ar(i)              '将第 i 个数存入 m1
        '调用 GCD 过程求 m1 和 n1 的最大公约数。由于采用传址方式
        'GCD 过程结束时，n1 中的数字即为两数的最大公约数
        Call GCD(m1, n1)
        If n1 = 1 Then      '只要本次求得的最大公约数为 1，不再继续
            Exit For
        End If
```

```
    Next
    Picture1.Print "  以上"; n; "个数的最大公约数是:"; n1
End Sub
Private Sub Command2_Click()    '清除
    Picture1.Cls
End Sub
Private Sub Command3_Click()    '结束
    End
End Sub
```

3. 扩展组合框的功能

组合框由单行文本框和列表框组合而成。为了与普通文本框相区别，可以将组合框的文本框部分称为“编辑框”。本实验项目要求对组合框的功能进行扩展：①当用户在编辑框中输入内容时，自动检索已有的列表项，将相匹配的列表项显示在编辑框中；②当用户在编辑框中输入内容并按回车键后，在组合框中添加不重复的列表项。将这两项扩展功能编制成 Public 自定义过程，放在标准模块（.bas）中，供其他各模块调用。

（1）设计界面并设置属性。在窗体上放置两个组合框，Combo1 的属性采用默认值，设 Combo2 的 Style 属性为 1-Simple Combo，适当调整其大小。添加四个标签，用于对组合框的样式及其功能进行简要注释。

（2）编写代码。在当前工程中添加一个标准模块（工程→添加模块），模块名称自定。在模块中添加两个作用范围为 Public 的自定义子过程，分别命名为 AutoMatch 和 AddCboItem，用于实现列表项自动匹配和添加新项目。为了增强代码的通用性，为两个子过程各设置一个组合框对象类型的形参。两个子过程的起始行如下：

```
Public Sub AutoMatch(cboX As ComboBox)
Public Sub AddCboItem(cboX As ComboBox)
```

当调用子过程时，用需要执行上述功能的特定组合框的名称作为实参即可，例如：

```
Call AutoMatch(Combo1)      '以组合框对象为参数调用自动匹配子过程
Call AddCboItem(Combo2)     '以组合框对象为参数调用添加项目子过程
```

在列表项自动匹配子过程（AutoMatch）中，将用户已输入的字符与组合框中已有的项目逐项进行模糊比较（用 Like 运算符），若有匹配者，则在编辑框中显示。

在添加新项目子过程（AddCboItem）中，将用户已输入的内容与组合框中已有的项目逐项进行准确比较（不区分大小写，可以用 UCase 或 LCase 函数转换），若列表中无该项目，则添加到列表中。

在窗体的 Form_Load 事件中，向两个组合框中添加若干项目以便测试。

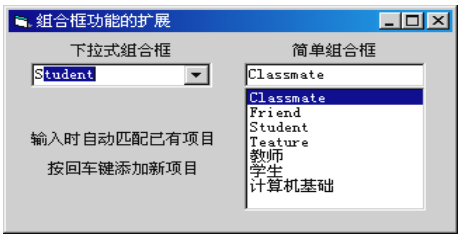
上述两个扩展组合框功能的子过程需要在组合框的不同事件中调用。

AutoMatch 子过程在组合框的 Change 事件中调用。为了避免用户按删除键或回删键时也进行自动匹配操作，需要设置一个模块级的逻辑型标志变量，在组合框的 KeyDown 事件中对用户按键进行判断，若用户按了删除键或回删键，将标志变量设为 False，否则为 True。在组合框的 Change 事件中调用 AutoMatch 子过程之前，先检查该标志变量，若为 False，则不调用

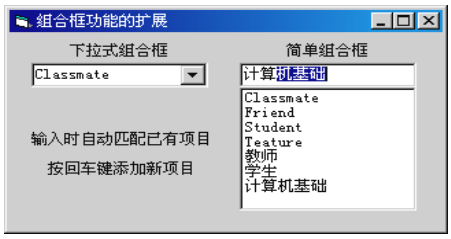
AutoMatch 子过程，即不进行自动匹配处理。

AddCboItem 子过程在组合框的 KeyDown 事件中调用。若用户按了回车键，则调用该过程，添加新的列表项。

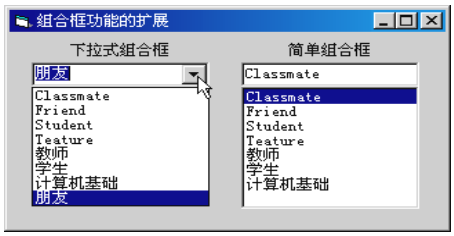
程序运行界面如图 8.3 所示。



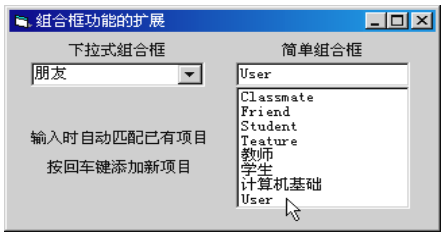
（a）在下拉式组合框中输入“s”



（b）在简单组合框中输入“计算”



（c）在下拉式组合框中输入“朋友”后按回车键



（d）在简单组合框中输入 User 后按回车键

图 8.3　组合框功能扩展

完整的程序代码如下：

```
'标准模块的程序代码
Option Explicit
'自定义子过程：在组合框中添加不重复的列表项
'可以在组合框的键盘事件（按回车键）或 LostFocus 事件中调用
'子过程的参数为组合框对象变量
Public Sub AddCboItem(cboX As ComboBox)
    Dim i As Integer
    Dim blnOld As Boolean   '项目存在标志
    Dim sNew As String

    blnOld = False
    sNew = Trim$(cboX.Text) '取用户在组合框的编辑框中输入的项目

    For i = 0 To cboX.ListCount - 1
        '若列表中已有该项目（不区分大小写），则设置标志，退出循环
        If LCase$(cboX.List(i)) = LCase$(sNew) Or sNew = "" Then
            blnOld = True
            Exit For
        End If
```

```
    Next

    If blnOld = False Then       '若列表中无该项目，添加
        cboX.AddItem sNew
    End If
End Sub

'自定义子过程：自动查找与组合框的编辑框中的内容相匹配的列表项
'可以在组合框的 Change 事件中调用
'子过程的参数为组合框对象变量
Public Sub AutoMatch(cboX As ComboBox)
    Dim intIns As Integer
    Dim strInput As String
    Dim i As Integer

    intIns = cboX.SelStart      '存放编辑框中的插入点
    strInput = Left$(cboX.Text, intIns)      '存放插入点左侧已输入的子字符串
    For i = 0 To cboX.ListCount - 1     '循环遍历组合框列表项
        '用 Like 运算符进行模糊比较
        '若有与编辑框中已输入的字符相匹配的列表项（不区分大小写）
        If LCase$(cboX.List(i)) Like LCase$(strInput) & "*" Then
            cboX.ListIndex = i        '定位并退出循环
            Exit For
        End If
    Next
    If cboX.ListIndex = -1 Then      '若无匹配的列表项，保留当前输入内容
        cboX.Text = strInput
    End If
    cboX.SelStart = intIns      '恢复插入点
    cboX.SelLength=Len(cboX.Text)-intIns      '选定插入点后面的内容（反相显示）
End Sub

'窗体模块的程序代码
Option Explicit
Dim blnAutoMatch As Boolean     '组合框列表项自动匹配标志

Private Sub Combo1_Change()     '组合框的编辑框的内容改变时
    If blnAutoMatch Then         '若自动匹配标志为 True
        Call AutoMatch(Combo1)   '以组合框对象为参数调用自动匹配子过程
    End If
End Sub

'组合框按键事件
Private Sub Combo1_KeyDown(KeyCode As Integer, Shift As Integer)
    '若按键为删除键或回删键，设自动匹配标志为 False，不进行自动匹配
    If KeyCode = vbKeyDelete Or KeyCode = vbKeyBack Then
```

```
        blnAutoMatch = False
    Else
        blnAutoMatch = True
    End If
    If KeyCode = vbKeyReturn Then    '若按键为回车键
        Call AddCboItem(Combo1)      '以组合框对象为参数调用添加项目子过程
    End If
End Sub

Private Sub Combo2_Change()
    If blnAutoMatch Then             '若自动匹配标志为 True
        Call AutoMatch(Combo2)       '以组合框对象为参数调用自动匹配子过程
    End If
End Sub

Private Sub Combo2_KeyDown(KeyCode As Integer, Shift As Integer)
    '若按键为删除键或回删键，设自动匹配标志为 False，不进行自动匹配
    If KeyCode = vbKeyDelete Or KeyCode = vbKeyBack Then
        blnAutoMatch = False
    Else
        blnAutoMatch = True
    End If
    If KeyCode = vbKeyReturn Then    '若按键为回车键
        Call AddCboItem(Combo2)      '以组合框对象为参数调用添加项目子过程
    End If
End Sub

Private Sub Form_Load()      '窗体加载
    Dim i As Integer
    '在组合框中添加若干项目以便测试
    With Combo1
        .AddItem "Classmate"
        .AddItem "Friend"
        .AddItem "Student"
        .AddItem "Teacher"
        .AddItem "教师"
        .AddItem "学生"
        .AddItem "计算机基础"
        .ListIndex = 0
    End With
    For i = 0 To Combo1.ListCount - 1   '将 Combo1 中的项目复制到 Combo2 中
        Combo2.AddItem Combo1.List(i)
    Next
    Combo2.ListIndex = 0
End Sub
```

实验 9　界面设计

一、实验目的

1．熟悉菜单、通用对话框、多文档界面（MDI）、工具栏等界面设计的方法。
2．掌握界面的设计并对对象编写代码。

二、实验内容

1．设计制作普通菜单和弹出式菜单。
2．设计制作工具栏。
3．利用“应用程序向导”制作多文档界面（MDI）应用程序。

三、实验步骤及指导

1．制作菜单

将实验 4（常用控件）和实验 7（数组）中的部分实验项目合并为一个多窗体工程，在工程中设计一个含有菜单栏的主窗体，通过菜单统一管理各实验项目程序的运行。

（1）设计主窗体界面。新建工程，将窗体 Form1 的名称改为 frmMain 作为主窗体，以“菜单.vbp”为名称保存工程。用菜单编辑器为主窗体设计菜单，菜单结构如表 9-1 所示。注意主菜单标题“弹出式菜单”下的菜单项为菜单控件数组，名称为 mnuSub，索引为 0，标题为空。

表 9-1　主窗体菜单结构

标题	名称	索引	标题	名称	索引
常用控件(&C)	mnuControl		....数制转换(&N)	mnuNS	
....图片框与图像框(&P)	mnuPicImg		....控件数组(&C)	mnuCtlArr	
....秒表(&T)	mnuTimer		退出(&X)	mnuExit	
....列表框与组合框(&L)	mnuLstCbo		弹出菜单	mnuPop	
数组(&A)	mnuArray		.... (空)	mnuSub	0

（2）添加各实验项目窗体。单击工具栏上的“添加窗体”按钮，在“添加窗体”对话框中选择“现存”选项卡，根据上述菜单结构，将实验 4（常用控件）和实验 7（数组）中有关实验项目的窗体添加到本工程中，注意不能有名称（Name）相同的窗体。

建议：为了对工程中的文件进行统一管理，减少程序移植运行时出错的机会，最好将本工程用到的所有文件存放在同一文件夹下。

（3）编写代码。在菜单栏中，用于启动各实验项目窗体的菜单项分别位于主菜单标题“常用控件”和“数组”之下。程序代码的任务主要是响应菜单栏中各菜单项的单击事件以及弹出式菜单的创建和激活。

响应菜单项单击事件。菜单项单击事件过程的代码很简单，只需显示对应的窗体即可。例如，在“秒表”菜单项的单击事件 mnuTimer_Click 过程中加入以下代码即可显示秒表窗体：

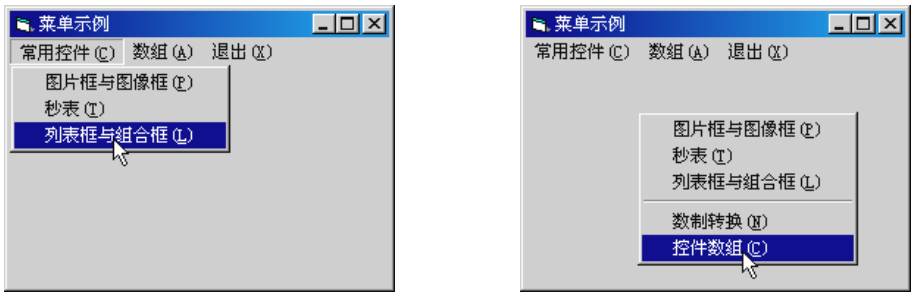
```
frmTimer.Show
```

创建弹出式菜单。弹出式菜单只能显示位于同一个主菜单标题下的菜单项，要想在一个弹出式菜单中显示来自不同菜单标题的菜单项，需要进行特殊处理。处理的方法很多，本例采用动态创建弹出式菜单项的方法。在设计菜单时，已将主菜单标题“弹出式菜单”下的菜单项设为菜单控件数组（仅有一项，索引为 0），窗体加载时，通过循环为菜单控件数组添加成员，然后根据“常用控件”和“数组”菜单下各菜单项的标题（Caption）设置菜单控件数组各成员的标题。

激活弹出式菜单。在窗体的 Form_MouseUp 事件中进行判断，若用户单击了鼠标右键，则调用窗体的 PopupMenu 方法显示弹出式菜单。

响应弹出式菜单项单击事件。在弹出式菜单项的单击事件 mnuSub_Click 中，用 Select Case 语句根据菜单控件数组索引（Index）执行菜单栏的相应菜单命令。

程序运行界面如图 9.1 所示。读者可以增加菜单标题，将更多的实验项目添加到本工程中。



（a）菜单栏　　（b）弹出式菜单

图 9.1　创建菜单

完整的程序代码如下：

```
Option Explicit
Private Sub Form_Load()      '窗体加载
    Dim i As Integer
    '设计时在主菜单标题“弹出式菜单”下仅设一个菜单项，将该菜单项
    '设为菜单控件数组，名称为 mnuSub，索引为 0，标题为空
    '窗体加载时添加菜单控件数组成员
    mnuPop.Visible = False        '将“弹出式菜单”设为不可见
    For i = 1 To 5
        Load mnuSub(i)        '加载菜单控件数组成员
```

```
    Next
    '设置菜单项标题
    mnuSub(0).Caption = mnuPicImg.Caption
    mnuSub(1).Caption = mnuTimer.Caption
    mnuSub(2).Caption = mnuLstCbo.Caption
    mnuSub(3).Caption = "-"    '菜单项分隔符
    mnuSub(4).Caption = mnuNS.Caption
    mnuSub(5).Caption = mnuCtlArr.Caption
End Sub

Private Sub Form_MouseUp(Button As Integer, Shift As Integer, _
        X As Single, Y As Single)
    If Button = vbRightButton Then   '右击窗体
        PopupMenu mnuPop, 2      '显示弹出式菜单
    End If
End Sub

Private Sub mnuSub_Click(Index As Integer)      '单击弹出式菜单项
    '根据菜单控件数组索引执行菜单栏的相应菜单命令
    Select Case Index
        Case 0
            mnuPicImg_Click
        Case 1
            mnuTimer_Click
        Case 2
            mnuLstCbo_Click
        Case 4
            mnuNS_Click
        Case 5
            mnuCtlArr_Click
    End Select
End Sub

Private Sub mnuCtlArr_Click()       '控件数组
    frmCtlArray.Show
End Sub

Private Sub mnuExit_Click()         '退出
    End
End Sub

Private Sub mnuLstCbo_Click()       '列表框与组合框
    frmLstCbo.Show
End Sub
```

```
Private Sub mnuNS_Click()          '数制转换
   frmDtoN.Show
End Sub

Private Sub mnuPicImg_Click()      '图片框与图像框
   frmPicImg.Show
End Sub

Private Sub mnuTimer_Click()       '秒表
   frmTimer.Show
End Sub
```

2. 设计制作工具栏

为上一个实验项目（制作菜单）添加工具栏，工具栏中的按钮与菜单中的菜单项相对应，实现相关菜单项的功能。

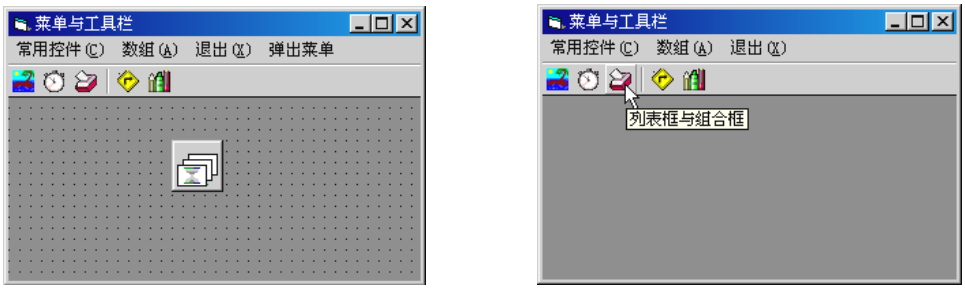
（1）准备工作。打开前面制作的“菜单.vbp”工程，单击“文件”菜单，选择“工程另存为...”命令，在弹出的对话框中以新名称“工具栏.vbp”保存工程。在属性窗口将窗体 frmMain 改名为 frmTlb，在工程资源管理器窗口中右击该窗体名称，在弹出的菜单中选择“frmMain.frm 另存为(A)...”命令，在弹出的对话框中以新名称 frmToolBar.frm 保存窗体。右击工具箱，在弹出的菜单中选择“部件(O)...”命令，在对话框“控件”选项卡的列表中选中 Microsoft Windows Common Controls 6.0，单击“确定”按钮。此时工具箱中新增的扩展控件中含有制作工具栏所需的 ToolBar 控件和 ImageList 控件。

（2）添加 ToolBar 控件和 ImageList 控件。显示 frmTlb 窗体设计器，双击工具箱中的 ToolBar 控件和 ImageList 控件图标，将其添加到窗体上，默认名称分别为 ToolBar1 和 ImageList1。

（3）在 ImageList 控件中添加图片。在本实验项目中，ImageList 控件的作用是为工具栏按钮提供所需的图片。右击 ImageList 控件，在弹出的菜单中选择“属性”菜单项，打开“属性页”对话框。在“通用”选项卡中选中“16×16”单选按钮。在“图像”选项卡中单击“插入图片”按钮打开“选择图片”对话框，查找并打开所需图标或位图文件，将图片添加到控件中，图片的索引号自动按顺序递增。重复插入图片，直至所需的全部图片均添加到控件中。为了便于访问，可以为每幅图片设置一个关键字。

（4）在 ToolBar 控件中添加工具栏按钮。首先要在 ImageList 控件和 ToolBar 控件之间建立关联。右击 ToolBar 控件，在弹出的菜单中选择“属性”菜单项，打开“属性页”对话框。在“通用”选项卡的“图像列表”组合框中选择 ImageList1 即可在二者间建立关联。建立关联后，在属性页的“按钮”选项卡中单击“插入按钮”按钮可以为工具栏添加按钮。对各按钮的设置内容主要有三项：①为按钮指定图片，在“图像”输入框中输入当前按钮所需图片在 ImageList 控件中的索引号或关键字；②在“样式”组合框中选择按钮样式，如图 9.2（a）所示，设计界面中共有 6 个按钮，其中第 4 个按钮的样式为分隔符（3 - tbrSeparator），其他按钮

均为默认样式（0 - tbrDefault）；③在“工具提示文本”输入框中输入提示内容，程序运行期间，当鼠标指针在按钮上作短暂停留时，将会显示该提示内容，如图 9.2（b）所示。如果工具栏按钮较多，最好为每个按钮设置一个关键字。



（a）设计时界面　　（b）运行时界面

图 9.2　制作工具栏

（5）为工具栏按钮编写代码。单击工具栏中的按钮（分隔符和占位符样式的按钮除外）时，将会引发 ToolBar 控件的 ButtonClick 事件。此事件由所有按钮共享，系统为该事件提供了一个 Button 对象参数，利用 Button 对象的 Index 或 Key 属性可以识别被单击的按钮。通常用 Select Case 语句为工具栏按钮编写代码。本实验项目复用了“菜单.vbp”工程的所有代码，由于弹出式菜单的菜单项包含了工具栏按钮的所有功能，而且采用了控件数组的形式，其索引与工具栏按钮的索引具有对应关系，因此可以使代码简化。在 ToolBar 控件的 ButtonClick 事件过程中，仅用一行代码调用弹出式菜单项的单击事件过程，同时传送一个索引参数即可。代码如下：

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
    '工具栏按钮索引（下界为 1）与弹出式菜单的菜单项索引（下界为 0）具有对应
    '关系。单击工具栏按钮时调用弹出式菜单项单击事件过程可以简化代码
    Call mnuSub_Click(Button.Index - 1)
End Sub
```

其他代码请参考“菜单.vbp”工程。

3. 使用“应用程序向导”制作多文档界面应用程序

使用“应用程序向导”可以自动生成含有菜单栏、工具栏、状态栏和通用对话框的多文档界面（MDI）应用程序。具体操作如下：

新建工程，在对话框中选择“VB 应用程序向导”，如图 9.3 所示。单击“确定”按钮后按照向导逐步操作。若无特殊需要，在向导的第二步直接单击“完成”按钮即可按照默认方式生成程序。图 9.4 是利用应用程序向导按默认方式生成的多文档界面应用程序的运行界面。在该程序中，所有界面的设计均由向导自动生成，工具栏所有按钮的功能代码和大部分菜单项的功能代码也已自动生成，只有少数菜单项的功能代码需要进行补充和修改。

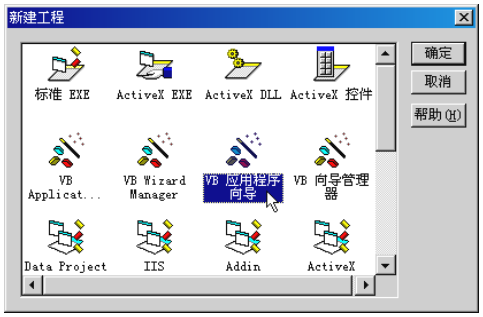


图 9.3　利用应用程序向导新建工程

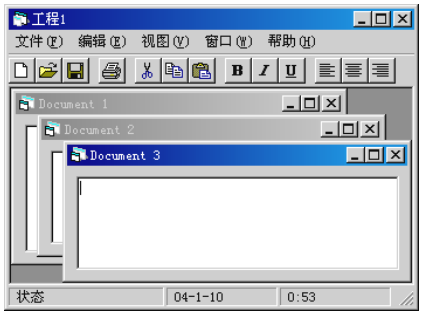


图 9.4　应用程序向导生成的程序的运行界面

实验 10　图形操作

一、实验目的

1．熟悉图形坐标系。
2．掌握绘图属性的用法。
3．掌握图形控件的用法。
4．掌握图形方法的应用。

二、实验内容

1．用图形控件制作奥运五环。
2．用绘图方法画坐标轴和动态正弦曲线。
3．用绘图方法画三维饼图。
4．用 PaintPicture 方法使图像翻转和缩放。

三、实验步骤及指导

1．用图形控件制作奥运五环

用 Shape 控件数组制作奥运五环，在五环下用 Print 方法显示“2008 北京”。

（1）设计界面并设置属性。设窗体的背景色为白色。在窗体上添加一个 Shape 控件，设其 Index 属性为 0，Shape 属性为 3-Circle，BorderWidth 属性为 7。

（2）编写代码。在窗体的 Form_Load 事件中用 Load 方法添加 Shape 控件数组成员，设置其位置和颜色（BorderColor 属性），构成奥运五环。在五环下面，用 Print 方法显示“2008 北京”字样。

程序运行界面如图 10.1 所示。

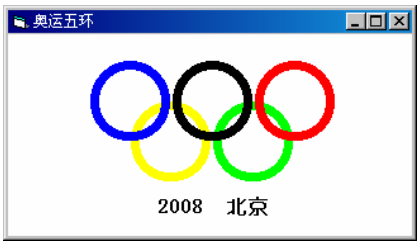


图 10.1　奥运五环

本实验项目更简单的做法是在设计界面时在窗体上放置五个 Shape 控件，适当调整其位

置，按奥运五环的规定设置各控件的颜色（BorderColor 属性）。再添加一个标签，设 BackStyle 属性为 1-Transparent，Caption 属性为“2008 北京”。用这种方法不必编写代码。

运行时加载 Shape 控件数组成员以及在窗体特定位置打印文字的程序代码如下：

```
Private Sub Form_Load()
    Dim iTop%, iLeft%, i%, j%, k%
    iTop = Shape1(0).Top          '环顶端坐标
    iLeft = Shape1(0).Left + Shape1(0).Width     '下一个环的左边坐标
    For i = 1 To 2                '外循环为行
        For j = 1 To 2            '内循环为列
            k = (i - 1) * 2 + j   '求控件数组下标
            Load Shape1(k)        '添加控件数组对象
            Shape1(k).Top = iTop  '定位
            Shape1(k).Left = iLeft
            Shape1(k).Visible = True      '显示
            iLeft = iLeft + Shape1(0).Width       '下一列位置
        Next j
        iTop = iTop + Shape1(0).Width / 2    '下一行位置
        iLeft = Shape1(0).Left + Shape1(0).Width / 2
    Next i
    Shape1(0).BorderColor = vbBlue       '设置颜色
    Shape1(2).BorderColor = vbRed
    Shape1(3).BorderColor = vbYellow
    Shape1(4).BorderColor = vbGreen
    Me.AutoRedraw = True
    Me.CurrentX = Shape1(3).Left + Shape1(3).Width / 3       '设置打印坐标
    Me.CurrentY = Shape1(3).Top + Shape1(3).Height + 200
    Print "2008  北京"
End Sub
```

2. 用绘图方法画坐标轴和动态正弦曲线

在图片框中用 Line 方法画出带有箭头的 x 轴和 y 轴。单击“正弦信号”按钮时，在图片框中用 PSet（画点）方法画出沿 x 轴波动的动态正弦曲线。通过单选按钮调整画曲线的速度。通过“停止/继续”按钮停止或继续画曲线的过程。

（1）设计界面并设置属性。在窗体上放置一个图片框，一个定时器，三个命令按钮，三个单选按钮（控件数组）。控件属性设置如表 10-1 所示。

表 10-1 控件属性

对象	名称	属性名	属性值	对象	名称	属性名	属性值
OptionButton	optRate(0)	Caption	快速	CommandButton	cmdSin	Caption	正弦信号
	optRate(1)	Caption	中速		cmdStop	Caption	停止
	optRate(2)	Caption	慢速		cmdEnd	Caption	退出
PictureBox	picSin			Timer	tmrSin		

（2）编写代码。程序代码的主要任务是画坐标轴和画正弦曲线。

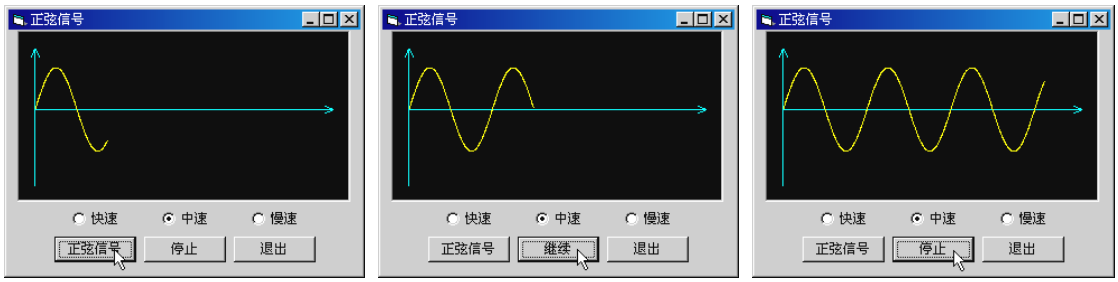
用 Line 方法画坐标轴。创建一个以图片框对象为形参的自定义子过程，用于在图片框中画坐标轴。在图片框中用 Line 方法分别画出 x 轴和 y 轴（上下左右各留出 200 缇的空间）。在 x、y 坐标轴末端的两侧用 Line 方法各画一条短斜线，即构成“箭头”。调用该子过程的语句置于“正弦信号”按钮的单击事件过程中。

用 PSet（画点）方法由点连线画出正弦曲线。为了画出动态的正弦曲线，将画曲线的语句置于定时器的 Timer 事件（Interval=50）过程中。画正弦曲线时应注意两个要点：首先要确定画曲线的起始坐标；其次要求出所画新点的 x 坐标和 y 坐标（用 Sin 函数根据 x 求 y），该坐标是新点距起始坐标的偏移量。

用单选按钮选择画曲线的速度。由于操作系统的限制，定时器控件的 Interval 属性值即使达到系统允许的最短间隔，画曲线的速度仍然显得非常缓慢。解决这一矛盾的方法是在定时器事件中添加一个 For 循环，使每次 Timer 事件多画若干点。定义一个模块级的变量，用于存放循环终值，在单选按钮的单击事件中改变循环终值的大小，即可调整画曲线的速度。

画曲线的暂停和继续。通过“停止”按钮实现暂停和继续切换的功能。

程序运行效果如图 10.2 所示。



（a）开始　　（b）单击“停止”按钮后　　（c）单击“继续”按钮后

图 10.2　正弦信号

程序代码如下：

```
Option Explicit
Dim intDeg As Integer
Dim intRate As Integer

'自定义过程：画坐标轴，以图片框对象作形参
Private Sub DrawAxis(picX As PictureBox)
    Dim x1 As Integer, y1 As Integer
    Dim x2 As Integer, y2 As Integer
    Dim y As Integer
    picX.BackColor = vbBlack     '黑色背景
    picX.Cls
    picX.DrawStyle = 0       '画线样式为实线
```

```
    '定义坐标轴起止点
    x1 = 200
    x2 = picX.ScaleWidth - 200
    y1 = picX.ScaleHeight - 200
    y2 = 200
    y = y1 / 2
    '画坐标轴，QBColor(11) = 亮青色
    picX.Line (x1, y1)-(x1, y2), QBColor(11)     'Y 轴，自下而上
    picX.Line (x1, y)-(x2, y), QBColor(11)       'X 轴，从左向右
    '画 Y 轴箭头
    picX.Line (x1 - 50, y2 + 120)-(x1, y2), QBColor(11)
    picX.Line (x1 + 50, y2 + 120)-(x1, y2), QBColor(11)
    '画 X 轴箭头
    picX.Line (x2 - 120, y + 50)-(x2, y), QBColor(11)
    picX.Line (x2 - 120, y - 50)-(x2, y), QBColor(11)
End Sub

Private Sub cmdSin_Click()          '“正弦信号”按钮
    picSin.AutoRedraw = True        '准备画持久图形
    Call DrawAxis(picSin)           '调用自定义过程画坐标轴
    intDeg = 1                      '设正弦曲线初始角度
    tmrSin.Enabled = True           '启动定时器
    cmdStop.Caption = "停止"
End Sub

Private Sub cmdEnd_Click()          '退出
    End
End Sub

Private Sub cmdStop_Click()         '停止或继续
    If cmdStop.Caption = "停止" Then
        tmrSin.Enabled = False
        cmdStop.Caption = "继续"
    Else
        tmrSin.Enabled = True
        cmdStop.Caption = "停止"
    End If
End Sub

Private Sub Form_Load()             '窗体加载
    tmrSin.Interval = 50            '设置定时器
    tmrSin.Enabled = False
    optRate(1).Value = True         '初始速度为中速
    intRate = 8
```

```
End Sub

Private Sub Form_Unload(Cancel As Integer)      '窗体卸载
   tmrSin.Enabled = False
End Sub

Private Sub optRate_Click(Index As Integer)     '单选按钮控件数组单击事件
   Select Case Index    '根据索引
      Case 0            '快速
         intRate = 20
      Case 1            '中速
         intRate = 8
      Case 2            '慢速
         intRate = 2
   End Select
End Sub

Private Sub tmrSin_Timer()      '定时器事件，用画点的方法画正弦曲线
   Dim x As Integer, y As Integer
   Dim scalY As Integer
   Dim i As Integer
   scalY = picSin.ScaleHeight / 4      'Y 轴最大值
   '用循环调整画曲线的速度，循环次数在单选按钮的单击事件中设置
   For i = 0 To intRate
      '设置绘图起始坐标，新点以此坐标+偏移量画出
      picSin.CurrentX = 200
      picSin.CurrentY = (picSin.ScaleHeight - 200) / 2

      x = intDeg / 180 * scalY      '取画点的 x 坐标

      'Sin 函数的参数要求用弧度，下面括号中的表达式将角度转换为弧度
      '即：弧度 = 角度 * π / 180 （intDeg 变量中存放的是角度）
      y = Sin(intDeg * 3.14 / 180) * scalY     '取画点的 y 坐标

      '用 PSet（画点）方法画曲线。Y 轴方向默认向下，为使曲线开始时向上，y 取负值
      '使用 Step 关键字时，x,y 表示所画新点距起始坐标的偏移量
      picSin.PSet Step(x, -y), vbYellow
      intDeg = intDeg + 1      '角度+1
      '若曲线画到图片框右边界，则回到起点重画
      If picSin.CurrentX >= picSin.ScaleWidth Then cmdSin_Click
   Next
End Sub
```

3. 画三维饼图

若以一个圆的面积表示事物的总体，以扇形的面积表示各部分占总体的百分数，则这种

图形称为扇形统计图，又叫做百分比较图。三维饼图是扇形统计图的三维表示形式（圆饼实际上是一个较矮的圆柱），常用于直观地显示事物总体中各部分的百分比。本实验项目要求统计某单位技术人员中初级、中级和高级职称人员所占的百分比，并且用三维饼图显示。

（1）设计界面并设置属性。在窗体上放置一个图片框，背景色为白色。添加一个框架，设 Caption 属性为“输入各级职称人数”。在框架中添加含有三个成员的文本框控件数组，设 Text 属性为空。再添加三个标签，Caption 属性分别为“初级”、“中级”和“高级”。在窗体上添加两个命令按钮，Caption 属性分别为“显示三维饼图”和“退出”。

（2）编写代码。程序代码主要是处理“显示三维饼图”按钮的单击事件。在事件过程中定义一个一维单精度数组 arRank，下界为 0，上界为 2（与各文本框的 Index 属性对应）；定义一个单精度变量 sTotal，用于存放人员总数。通过 For 循环将用户在文本框中输入的数据存入数组，并将累加值存入 sTotal，然后将各类人员数除以人员总数得出人员比例再存入数组中。

为了便于对程序进行修改和调试，同时使读者了解如何创建和使用以数组作为参数的自定义过程，在本实验项目中要求将画饼图的程序段设计为自定义子过程，在按钮的单击事件中以数组 arRank 作为实参调用该过程。在画圆饼的子过程中要解决的主要技术问题是：①要使圆饼（柱）看上去更加真实，必须将圆饼的底面画成椭圆形；②利用循环自下而上画出一系列由不同颜色的扇形拼接构成的椭圆，形成三维圆饼效果；③在圆饼的上底面勾画出扇形的边缘（一般用黑色）；④将圆饼侧面与底面扇形对应部分的颜色设为与底面扇形色彩相近的较深颜色；⑤勾画出整个圆饼的轮廓线以及相邻扇形在圆柱侧面的分隔线（黑色）；⑥显示图例。用 Circle 方法和 Line 方法可以完成上述任务。具体实现步骤请参考程序代码。

程序运行效果如图 10.3 所示。

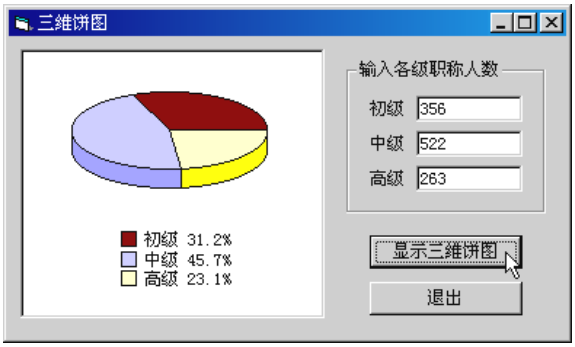


图 10.3　三维饼图

程序代码如下：

```
Option Explicit
'自定义过程：画饼图，形参为数组
Private Sub DrawCake(Ar() As Single)
    Const PI = 3.1415926
    Dim i As Integer, sngRatio As Single
```

```
Dim CX As Single, CY As Single, CR As Single
CX = 1500        '设置圆心坐标和半径
CY = 1000
CR = 1000
sngRatio = 0.4   '椭圆纵横比
Picture1.Cls
Picture1.FillStyle = 0       '0=vbFSSolid，实线
'每次循环根据各职称比例画三个扇形构成椭圆
'自下而上画 200 个同样的图形可以形成三维饼图（圆柱）效果
For i = 1 To 200
   If Ar(0) > 0# Then        '初级
      Picture1.FillColor = RGB(128, 0, 0)       '暗红色填充
      '画扇形，扇形轮廓线的颜色比填充色略深
      Picture1.Circle (CX, CY - i), CR, RGB(100, 0, 0), _
         -2 * PI, -2 * PI * Ar(0), sngRatio
   End If
   If Ar(1) > 0# Then        '中级
      If Ar(0) = 0# Then Ar(0) = 0.0000001
      Picture1.FillColor = RGB(192, 192, 255)       '浅蓝灰色填充
      Picture1.Circle (CX, CY - i), CR, RGB(150, 150, 255), _
         -2 * PI * Ar(0), -2 * PI * (Ar(0) + Ar(1)), sngRatio
   End If
   If Ar(2) > 0# Then        '高级
      Picture1.FillColor = RGB(255, 255, 192)       '浅黄色填充
      Picture1.Circle (CX, CY - i), CR, RGB(240, 240, 0), _
         -2 * PI * (Ar(0) + Ar(1)), -2 * PI, sngRatio
   End If
Next
Picture1.FillStyle = 1       '1=vbFSTransparent，透明
Picture1.Circle (CX, CY), CR, vbBlack, PI, 0, sngRatio   '饼图下缘弧线
CY = CY - 200
'最上面的扇形
Picture1.Circle (CX, CY), CR, vbBlack, -2 * PI, -2 * PI * Ar(0), sngRatio
Picture1.Circle (CX, CY), CR, vbBlack, -2 * PI * Ar(0), _
   -2 * PI * (Ar(0) + Ar(1)), sngRatio
Picture1.Circle (CX, CY), CR, vbBlack, -2 * PI * (Ar(0) + Ar(1)), _
   -2 * PI, sngRatio
'圆柱两侧的竖线
Picture1.Line (500, CY)-(500, CY + 250)
Picture1.Line (2500, CY)-(2500, CY + 250)
'利用椭圆参数方程画相邻扇形在圆柱侧面的分隔线
Dim a As Single
If Ar(0) > 0.5 Then
   a = 2 * PI * Ar(0)
```

```
        CX = CR * Cos(a) + CX
        CY = CY - CR * sngRatio * Sin(a)
        Picture1.Line (CX, CY)-(CX, CY + 210)
    End If
    CX = 1500: CY = 800
    If Ar(0) + Ar(1) > 0.5 Then
        a = 2 * PI * (Ar(0) + Ar(1))
        CX = CR * Cos(a) + CX
        CY = CY - CR * sngRatio * Sin(a)
        Picture1.Line (CX, CY)-(CX, CY + 210)
    End If
    '显示图例
    Ar(0) = Val(Format(Ar(0) * 100, "0.0"))
    Ar(1) = Val(Format(Ar(1) * 100, "0.0"))
    Ar(2) = 100 - Ar(0) - Ar(1)
    Picture1.FillStyle = 0
    Picture1.FillColor = RGB(128, 0, 0)
    Picture1.Line (1000, 2000)-(1150, 1850), , B
    Picture1.Print " 初级 "; Ar(0) & "%"
    Picture1.FillColor = RGB(192, 192, 255)
    Picture1.Line (1000, 2200)-(1150, 2050), , B
    Picture1.Print " 中级 "; Ar(1) & "%"
    Picture1.FillColor = RGB(255, 255, 192)
    Picture1.Line (1000, 2400)-(1150, 2250), , B
    Picture1.Print " 高级 "; Ar(2) & "%"
End Sub

Private Sub Command1_Click()        '“显示三维饼图”按钮
    Dim arRank(2) As Single, sTotal As Single
    Dim i As Integer, j As Integer
    For i = 0 To 2       '将各文本框中的数据存入数组
        arRank(i) = Val(Text1(i).Text)
        If arRank(i) = 0 Then j = j + 1
        sTotal = sTotal + arRank(i)
    Next
    If sTotal = 0 Or j > 1 Then Exit Sub        '至少应有两个数据
    For i = 0 To 2       '将人数转换为比例
        arRank(i) = arRank(i) / sTotal
    Next
    Call DrawCake(arRank)        '调用画图过程，实参为数组
End Sub

Private Sub Command2_Click()    '退出
    End
```

```
End Sub

Private Sub Form_Load()
    Picture1.AutoRedraw = True
End Sub

Private Sub Text1_GotFocus(Index As Integer)
    Text1(Index).SelStart = 0
    Text1(Index).SelLength = Len(Text1(Index).Text)
End Sub

Private Sub Text1_KeyPress(Index As Integer, KeyAscii As Integer)
    If Not IsNumeric(Chr(KeyAscii)) And KeyAscii <> 8 Then
        KeyAscii = 0      '若按键为非数字键或回删键，则取消按键
    End If
End Sub
```

4. 图像翻转

用 PaintPicture 方法复制图像，通过改变 PaintPicture 方法中的参数（坐标、高度和宽度）使图像伸缩，模拟图像水平或垂直“旋转”效果。

（1）设计界面并设置属性。在工程中设置三个窗体，窗体 Form2 和 Form3 不放置控件，用于显示被复制的图像。在 Form1 上添加一个图片框，命名为 picF，通过图片框的 Picture 属性为其加载一幅图片。添加两个框架，每个框架中添加两个命令按钮，Frame1 中按钮的 Caption 属性分别为“水平旋转”和“停止”；Frame2 中按钮的 Caption 属性分别为“垂直旋转”和“停止”。在窗体上添加一个命令按钮，Caption 属性为“退出”。在窗体上添加两个定时器。

（2）编写代码。本程序的核心语句是调用窗体的 PaintPicture 方法实现图像的复制、翻转、伸缩。PaintPicture 方法的格式如下：

```
目标对象.PaintPicture 源图像, dx, dy[, dw, dh, sx, sy, sw, sh, 组合模式]
```

如果只是图像的简单复制，调用 PaintPicture 方法时只带有必选参数（源图像及目标图像的坐标 dx、dy）即可，其他可选参数均忽略。本实验项目要求在复制图像的同时使图像产生水平或垂直旋转的效果，因此，除了必选参数外，还需要用到目标图像的宽度（dw）和高度（dh）两个可选参数。将 dw 或 dy 设为负值，可以实现图像的水平或垂直翻转。动态改变 dx、dy、dw 和 dh 的大小，可以模拟图像水平或垂直“旋转”的效果。为了使图像有规律地“旋转”，可以将 PaintPicture 方法的调用放在定时器控件的 Timer 事件过程中。在程序代码中应该注意控制 dw 和 dh 的极限值，二者的绝对值最好分别控制在图像的原始宽度和高度以内。

程序部分运行效果如图 10.4 所示。

程序代码如下：

```
'Form1 程序代码
Option Explicit
Dim iZoom As Integer
```

（a）原始图像　（b）垂直翻转　（c）水平翻转　（d）正在“旋转”

图 10.4　图像的水平和垂直“旋转”

```
Dim iW As Integer, HorX As Integer, FlagW As Integer
Dim iH As Integer, VertY As Integer, FlagH As Integer

Private Sub Command1_Click()     '“水平旋转”按钮
    Timer1.Enabled = True        '启动计时器 1
End Sub

Private Sub Command2_Click()     '“停止”水平旋转按钮
    Timer1.Enabled = False       '关闭计时器 1
End Sub

Private Sub Command3_Click()     '“垂直旋转”按钮
    Timer2.Enabled = True        '启动计时器 2
End Sub

Private Sub Command4_Click()     '“停止”垂直旋转按钮
    Timer2.Enabled = False       '关闭计时器 2
End Sub

Private Sub Command5_Click()     '“退出”按钮
    End
End Sub

Private Sub Form_Load()          '窗体加载，初始化
    Form2.Show                   '显示窗体 2（水平旋转）
    Form2.AutoRedraw = True      '允许自动重画
    Form3.Show                   '显示窗体 3（垂直旋转）
    Form3.AutoRedraw = True      '允许自动重画
    Timer1.Interval = 50         '设置计时器时间间隔
    Timer2.Interval = 50
    iW = picF.Width                                    '取图片框（picF）原始宽度
    HorX = (Form2.ScaleWidth - picF.Width)/2           '水平旋转前图像左边初始 X 坐标
    FlagW = 1                                          '设图像宽度伸缩标志为 1
```

```
    iH = picF.Height                                    '取图片框（picF）原始高度
    VertY = (Form3.ScaleHeight - picF.Height)/2 '垂直旋转前图像顶端初始 Y 坐标
    FlagH = 1                                           '设图像高度伸缩标志为 1
    Timer1.Enabled = False       '关闭计时器
    Timer2.Enabled = False
End Sub

Private Sub Form_Unload(Cancel As Integer)
    End
End Sub

Private Sub Timer1_Timer()  'Timer1 计时器事件，用于水平旋转
    iZoom = 100         '设置图像水平伸缩值（缇），通过图像伸缩模拟旋转效果
    If iW >= picF.Width Then          '若图像当前宽度≥原始宽度
        FlagW = 1                     '设宽度伸缩标志为 1
    ElseIf iW <= -picF.Width Then     '若图像当前宽度≤原始宽度的负值
        FlagW = 0                     '设宽度伸缩标志为 0
    End If
    If FlagW = 1 Then                 '若宽度伸缩标志为 1
        iW = iW - iZoom               '图像宽度减伸缩值
        HorX = HorX + iZoom / 2       '左边界右移
    Else                              '若宽度伸缩标志为 0
        iW = iW + iZoom          '图像宽度加伸缩值
        HorX = HorX - iZoom / 2 '左边界左移
    End If
    Form2.Cls
    '将图片框中的图像“画”到 Form2 窗体上
    'HorX 和 picF.Top 为图像左上角坐标
    'iW 为图像宽度，若为负值，可以使图像水平翻转
    'picF.Height 为图像高度
    Form2.PaintPicture picF.Picture, _
        HorX, picF.Top, iW, picF.Height
End Sub

Private Sub Timer2_Timer()  'Timer2 计时器事件，用于垂直旋转
    iZoom = 70     '设置图像垂直伸缩值（缇），通过图像伸缩模拟旋转效果
    If iH >= picF.Height Then         '若图像当前高度≥原始高度
        FlagH = 1                     '设高度伸缩标志为 1
    ElseIf iH <= -picF.Height Then    '若图像当前高度≤原始高度的负值
        FlagH = 0                     '设高度伸缩标志为 0
    End If
    If FlagH = 1 Then                 '若高度伸缩标志为 1
        iH = iH - iZoom               '图像高度减伸缩值
        VertY = VertY + iZoom / 2     '顶端下移
    Else                              '若高度伸缩标志为 0
        iH = iH + iZoom               '图像高度加伸缩值
```

```
        VertY = VertY - iZoom / 2     '顶端上移
    End If
    Form3.Cls
    '将图片框中的图像“画”到 Form3 窗体上
    'picF.Left + 50 和 VertY 为图像左上角坐标
    'picF.Width 为图像宽度
    'iH 为图像高度，若为负值，可以使图像垂直翻转
    Form3.PaintPicture picF.Picture, _
        picF.Left + 50, VertY, picF.Width, iH
End Sub

'Form2 程序代码
Option Explicit
Dim F2 As Integer

Private Sub Form_Click()          '单击窗体时停止或继续旋转
    If F2 = 0 Then
        Form1.Timer1.Enabled = False
        F2 = 1
    Else
        Form1.Timer1.Enabled = True
        F2 = 0
    End If
End Sub

Private Sub Form_Load()
    F2 = 0
End Sub

'Form3 程序代码
Option Explicit
Dim F3 As Integer

Private Sub Form_Click()          '单击窗体时停止或继续旋转
    If F3 = 0 Then
        Form1.Timer2.Enabled = False
        F3 = 1
    Else
        Form1.Timer2.Enabled = True
        F3 = 0
    End If
End Sub

Private Sub Form_Load()
    F3 = 0
End Sub
```

实验 11　文件系统

一、实验目的

1．掌握文件系统控件的使用。
2．掌握用 FSO 对象模型对顺序文件的读写以及对驱动器、文件夹和文件的操作。

二、实验内容

1．用文件系统控件和图像框制作图片浏览器。
2．用 FSO 对象模型访问文件系统。

三、实验步骤及指导

1．制作图片浏览器

利用文件系统控件、图像框和单选按钮制作一个简单的图片浏览器。当用户在文件列表框中选择图片文件时，在图像框中显示该图片，用单选按钮可以调整图片的显示比例。

（1）设计界面并设置属性。在窗体上放置驱动器列表框、目录列表框和文件列表框各一个。添加两个框架，设 Frame1 的 Caption 属性为“显示比例”，Frame2 的 Caption 属性为空。在 Frame1 中添加含有四个成员的单选按钮控件数组 Option1(0)～Option1(3)，设其 Caption 属性分别为“25%”、“50%”、“75%”和“100%”。在 Frame2 中添加一个图像框。

（2）编写代码。程序代码的主要任务有三项：获取用户选择的图片文件的位置和文件名；在图像框中显示图片；根据显示比例调整图像框的大小。

三个文件系统控件组合在一起，可以实现对计算机整个文件系统的访问。编程的关键是使三者保持同步，在代码中只要响应驱动器列表框（Drive1）和目录列表框（Dir1）的 Change 事件即可。在三者同步的前提下，通过文件列表框（File1）的 Path 属性和 FileName 属性可以获取图片文件的路径和文件名。需要提醒读者注意的是，应当在驱动器列表框的 Change 事件过程中加入出错处理代码，否则，程序运行时若选择了尚未插入磁盘的软驱（或光驱），程序会因出错而崩溃。

在文件列表框（File1）的单击事件过程中，编写代码获取图片文件的路径和文件名，然后用 LoadPicture 函数将图片显示在图像框中。

单选按钮控件数组的单击事件中，根据索引判断被选择的单选按钮，进而确定显示比例，调整图像框的大小。

提示：本实验项目有一个缺陷，程序中图像框的宽度和高度比例是固定的，往往与实际

图片的宽高比不同，若二者相差较大，则会导致图像显示失真。读者可以根据原始图片的宽高比调整图像框的宽度和高度。以下代码可以获得原始图像的宽高比：

```
Dim sngWvsH As Single
sngWvsH = Image1.Picture.Width / Image1.Picture.Height
```

程序运行效果如图 11.1 所示。

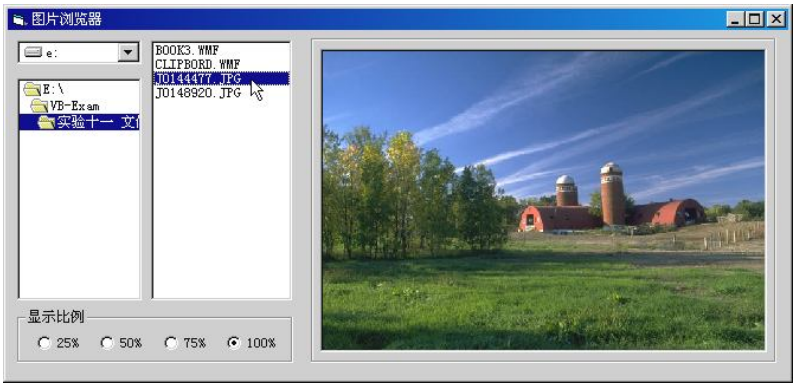


图 11.1　图片浏览器

程序代码如下：

```
Option Explicit
Dim sngWidth As Single, sngHeight As Single

Private Sub Dir1_Change()
    '目录列表框 Path 属性改变时触发 Change 事件
    File1.Path = Dir1.Path        '使文件列表框与目录列表框的 Path 属性同步
End Sub

Private Sub Drive1_Change()
    On Error Resume Next          '出错执行下一句
    '在驱动器列表框选择新驱动器后
    'Drive1 的 Drive 属性改变，触发 Change 事件
    Dir1.Path = Drive1.Drive      '将驱动器盘符赋予目录列表框 Path 属性

    If Err.Number Then            '若有错误发生（如软驱中无磁盘）
       MsgBox "设备未准备好！", vbCritical
    End If
End Sub

Private Sub File1_Click()    '在文件列表框中选择文件
    Dim fName As String
    '取文件全路径
    If Right$(File1.Path, 1) = "\" Then
       fName = File1.Path & File1.FileName
```

```
    Else
        fName = File1.Path & "\" & File1.FileName
    End If
    Image1.Picture = LoadPicture(fName)      '加载图片文件
End Sub

Private Sub Form_Load()
    '设置文件过滤（各种图片文件）
    File1.Pattern = "*.emf;*.wmf;*.jpg;*.jpeg;" & _
        "*.bmp;*.dib;*.gif;*.gfa;*.ico;*.cur"
    Image1.Stretch = True        '使图片适应图像框
    sngWidth = Image1.Width      '存图像框原始宽、高
    sngHeight = Image1.Height
End Sub

Private Sub Option1_Click(Index As Integer) '选择显示比例
    Image1.Width = sngWidth * Val(Option1(Index).Caption) / 100
    Image1.Height = sngHeight * Val(Option1(Index).Caption) / 100
End Sub
```

2. 用 FSO 对象模型访问文件系统

用 FSO 对象模型和文件系统控件编制一个可以读写顺序文件、创建新文件和文件夹的程序。

（1）设计界面并设置属性。在窗体上放置驱动器列表框、目录列表框和文件列表框各一个。添加一个组合框，Style 属性设为 2-Dropdown List。添加一个文本框，设 Text 属性为空。添加四个命令按钮，Caption 属性分别为“新建文件”、“新建文件夹”、“写文件”和“退出”。添加五个标签，用于对其他控件进行简要说明。

（2）添加 FSO 对象引用。要在程序中使用 FSO 对象模型，必须加载该对象的引用。执行“工程”菜单中的“引用”命令，在对话框的列表中选中 Microsoft Scripting Runtime，单击“确定”按钮。

（3）编写代码。程序代码的主要任务是：声明 FSO 对象变量并创建对象；使三个文件系统控件保持同步（参考实验项目 1）；响应文件列表框（File1）的双击事件以读取文件内容；响应各命令按钮的单击事件以完成相应的功能。

在代码的“通用-声明”部分定义一个 FileSystemObject 对象变量和一个 TextStream 对象变量。

在文件列表框（File1）的双击事件过程中用 FileSystemObject 对象的 OpenTextFile 方法以读方式（ForReading）打开文件，用 TextStream 对象的 ReadAll 方法读取文件内容并显示在文本框中。

在“新建文件”按钮的单击事件过程中通过 InputBox 函数获得用户输入的文件名，用 FileSystemObject 对象的 CreateTextFile 方法建立新文件。

在“新建文件夹”按钮的单击事件过程中通过 InputBox 函数获得用户输入的文件夹名称，

用 FileSystemObject 对象的 CreateFolder 方法建立新文件夹。

在“写文件”按钮的单击事件过程中用 FileSystemObject 对象的 OpenTextFile 方法以写方式（ForWriting）打开文件，用 TextStream 对象的 Write 方法将文本框中的内容写入文件。

在上述事件过程中均需进行获取文件路径的操作，因此可以将有关代码编制成一个自定义函数统一调用。

读者可以扩展该程序的功能，对文件和文件夹进行复制、删除和移动等操作。

程序运行效果如图 11.2 所示。

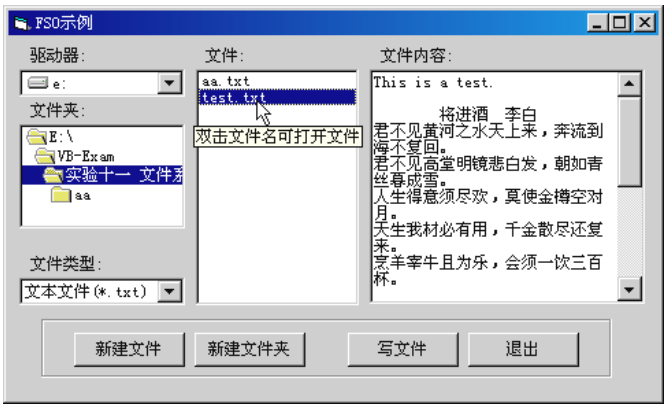


图 11.2　用 FSO 访问文件系统

程序代码如下：

```
Option Explicit
Dim fso As New FileSystemObject 'FileSystemObject 对象变量
Dim tstFile As TextStream       'TextStream 对象变量
Dim sPathName As String         '拟存放文件路径

Private Sub cmdEnd_Click()      '退出
   Unload Me
End Sub

Private Sub cmdFile_Click()     '“新建文件”按钮
   Dim fName As String
   fName = InputBox("请输入文件名", "输入")
   If fName = "" Then Exit Sub      '若文件名为空，退出过程
   sPathName = GetDir       '调用获取路径自定义函数
   fso.CreateTextFile (sPathName & fName)  '新建文件
   File1.Refresh            '刷新文件列表框
End Sub

Private Sub cmdFold_Click()     '“新建文件夹”按钮
   On Error Resume Next
```

```
    Dim sFolderName As String
    sFolderName = InputBox("请输入文件夹名", "输入")
    If sFolderName = "" Then Exit Sub '若文件夹名为空，退出过程
    sPathName = GetDir      '调用获取路径自定义函数
    '新建文件夹
    fso.CreateFolder (sPathName & sFolderName)
    If Err.Number Then MsgBox "文件夹已存在。", vbExclamation, "提示"
    Dir1.Refresh            '刷新目录列表框
End Sub

Private Sub cmdWrite_Click()    '“写文件”按钮
    If File1.FileName = "" Then Exit Sub
    sPathName = GetDir          '调用获取路径自定义函数
    '写方式打开文件
    Set tstFile = fso.OpenTextFile(sPathName & File1.FileName, ForWriting)
    tstFile.Write (Text1.Text)      '将文本框内容写入文件
    tstFile.Close       '关闭文件
End Sub

Private Sub Combo1_Click()          '选择组合框项目时为文件类型赋值
    File1.Pattern = Mid(Combo1.Text, 21)
End Sub

Private Sub Dir1_Change()
    '目录列表框 Path 属性改变，触发 Change 事件
    '使文件列表框与目录列表框的 Path 属性同步
    File1.Path = Dir1.Path
End Sub

Private Sub Drive1_Change()
    On Error Resume Next        '出错执行下一句
    '在驱动器列表框选择新驱动器后
    'Drive1 的 Drive 属性改变，触发 Change 事件
    Dir1.Path = Drive1.Drive    '将驱动器盘符赋予目录列表框的 Path 属性
    If Err.Number Then      '若有错误发生（如 A 驱中无盘）
       MsgBox "设备未准备好！", vbCritical
    End If
End Sub

Private Sub File1_DblClick()    '在文件列表框双击文件名读文件
    On Error Resume Next    '出错执行下一句
    Dim fName As String
    sPathName = GetDir      '调用获取路径自定义函数
    fName = sPathName & File1.FileName
```

```
    If FileLen(fName)=0 Then Text1.Text="":Exit Sub     若为空文件，则退出过程
    '读方式打开文件
    Set tstFile = fso.OpenTextFile(fName, ForReading)
    Text1.Text = tstFile.ReadAll     '将文件内容读至文本框
    If Err.Number Then MsgBox "文件太大，无法打开。", vbExclamation
    tstFile.Close      '关闭文件
End Sub

Private Sub Form_Load()          '窗体加载
    Drive1.Drive = App.Path      '设默认驱动器
    Dir1.Path = App.Path         '设默认目录
    File1.Pattern = "*.txt"      '设默认文件类型
    File1.ToolTipText = "双击文件名可以打开文件"
    cmdWrite.ToolTipText = "将文本框内容写入选定的文件"
    '对文件类型（组合框 Combo1）进行初始化
    Dim sItem As String
    sItem = "文本文件(*.txt)"
    Combo1.AddItem sItem + Space(20 - Len(sItem)) + "*.txt"
    sItem = "所有文件(*.*)"
    Combo1.AddItem sItem + Space(20 - Len(sItem)) + "*.*"
    Combo1.ListIndex = 0    '默认类型为文本文件
End Sub

Private Function GetDir()        '自定义函数：获取路径
    If Right$(Dir1.Path, 1) = "\" Then  '若为根目录
       GetDir = Dir1.Path
    Else    '若为子目录
       GetDir = Dir1.Path & "\"
    End If
End Function

Private Sub Form_Unload(Cancel As Integer)
    Text1.Text = ""         '若文件较大，清空文本框可以加快卸载速度
    Set tstFile = Nothing  '释放对象变量
    Set fso = Nothing
End Sub
```

实验 12　数据控件

一、实验目的

1．掌握建立数据库的一般步骤。
2．掌握用数据窗体向导创建数据库应用程序的步骤。
3．掌握自定义数据访问窗体的创建。

二、实验内容

1．建立 Access 数据库。
2．用数据窗体向导创建数据库应用程序。
3．创建自定义数据访问窗体。

三、实验步骤及指导

1．建立 Access 数据库

（1）在 Visual Basic 环境中创建数据库。在 Visual Basic 中可以使用“可视化数据管理器”直接建立 Access 数据库。操作步骤如下：

1）启动数据管理器。在 Visual Basic 环境中执行“外接程序”菜单中的“可视化数据管理器”命令，打开可视化数据管理器（VisData）。

2）建立数据库。在 VisData 窗口中执行菜单命令“文件”→“新建”→Microsoft Access Ver 7.0 MDB，打开“选择要创建的 Microsoft Access 数据库”对话框，在对话框中输入数据库文件名（如 TestVb）并保存，VisData 窗口的工作区将出现如图 12.1 所示的“数据库窗口”（此时为空库，无表）。

图 12.1　可视化数据管理器-数据库窗口

3）建立数据表。右击数据库窗口的空白处，在弹出的菜单中选择“新建表”菜单项，打开如图 12.2 所示的“表结构”对话框，输入表名称（如“基本情况”）后，单击“添加字段”按钮，打开如图 12.3 所示的“添加字段”对话框，输入字段名称，设置类型和大小（仅 Text 类型）。添加了所有字段后，单击“生成表”按钮即可建立数据表。在一个库中可以建立多个不同名称的表。“基本情况”表的结构定义如表 12-1 所示。

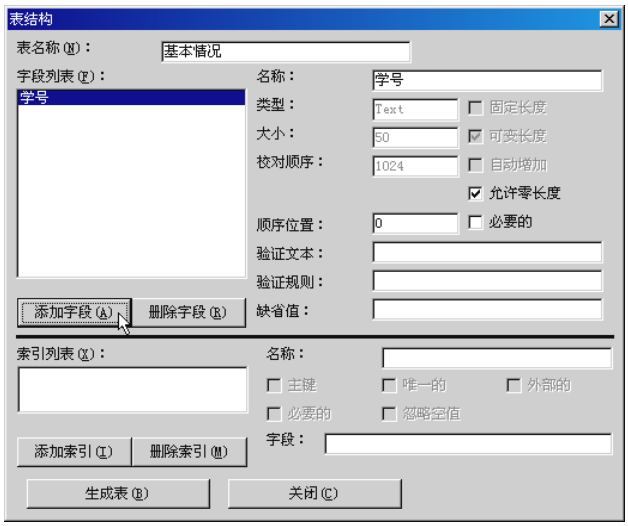


图 12.2　表结构

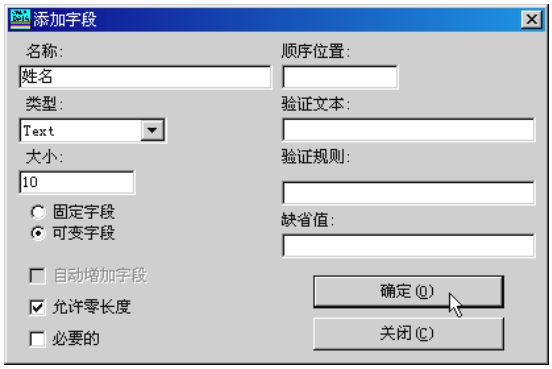


图 12.3　添加字段

表 12-1　基本情况表结构定义

字段名	类型	大小	字段名	类型	大小
学号	文本（Text）	20	出生日期	日期（Date）	
姓名	文本（Text）	10	专业	文本（Text）	20
性别	文本（Text）	2	班级	文本（Text）	20

4）输入记录。双击“数据库窗口”中的表名称，打开如图 12.4 所示的记录操作窗口，可以对记录进行增、删、修改等操作。

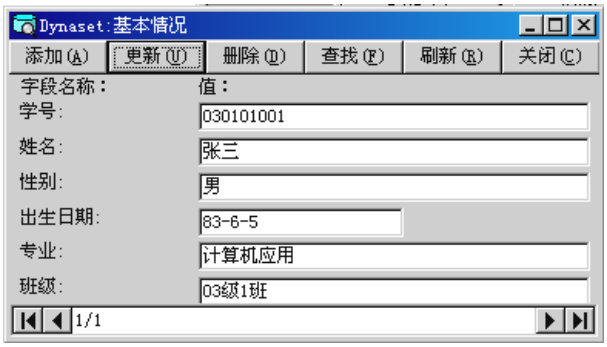


图 12.4　记录操作

（2）用 MS Access 建立数据库。上述“可视化数据管理器”由于版本较早，功能较弱，读者可以用 Microsoft Access 建立数据库及其数据表并输入数据，操作方法请参考 Microsoft Access 的联机帮助。

注意：Visual Basic 的“可视化数据管理器”不能识别用 MS Access 2000 以上版本建立的数据库。对于较高版本的数据库，可以在 Microsoft Access 中通过菜单命令“工具”→“数据库实用工具”→“转换数据库”→“到早期 Access 数据库版本”，将其转换为 Visual Basic“可视化数据管理器”能够识别的数据库。

2. 用数据窗体向导创建数据库应用程序

用“数据窗体向导”可以快速创建一个数据访问窗体。

执行“工程”菜单中的“添加窗体”命令，打开如图 12.5 所示的对话框，在“新建”选项卡中选择“VB 数据窗体向导”，单击“打开”按钮后将会出现向导的第一个对话框。

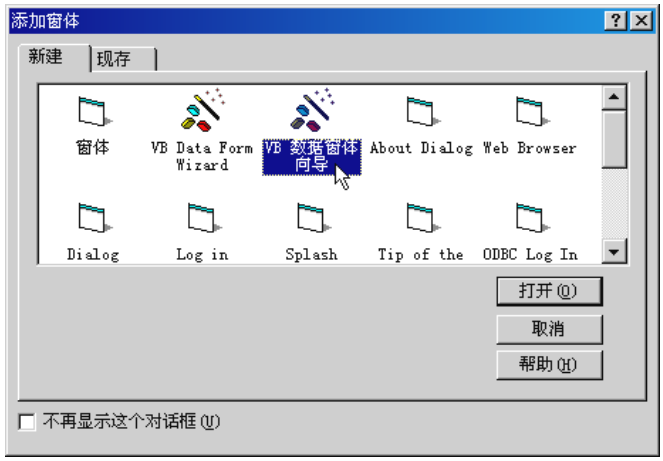


图 12.5　启动数据窗体向导

如果是创建单表访问窗体，数据窗体向导将有七个步骤：介绍、数据库类型、数据库、窗体（Form）、记录源、控件选择和完成，可以根据向导提示操作。

在本实验项目中，数据库类型为 Access，数据库为前面建立的 TestVb.mdb，窗体布局为单个记录，绑定类型为 ADO 数据控件，记录源为“基本情况”表中的所有字段。生成数据窗体后，将其设为启动窗体即可运行，运行界面如图 12.6 所示。

图 12.6　用数据窗体向导创建的数据窗体

3. 创建自定义数据访问窗体

用 ADO 数据控件和 DataGrid 控件创建一个简单的数据访问窗体，可以显示 TestVb.mdb 数据库中基本情况表的内容。

用数据窗体向导创建数据访问窗体虽然快捷，但界面形式单一，利用 ADO 数据控件和数据绑定控件可以创建具有不同风格和功能的数据访问窗体。

创建自定义数据访问窗体的一般步骤如下：

（1）加载 ADO 数据控件和数据绑定扩展控件。ADO 数据控件和数据绑定扩展控件（如 DataGrid 控件）属于 ActiveX 控件，必须将其添加到工具箱中才能使用。右击工具箱，在弹出的菜单中选择“部件”命令，在对话框的“控件”选项卡的列表中选中“Microsoft ADO Data Control 6.0(OLEDB)”和“Microsoft DataGrid 6.0(OLEDB)”，单击“确定”按钮。选择工具箱中新增加的 ADO 数据控件（Adodc）和 DataGrid 控件，将其添加到窗体上。

（2）设置 ADO 数据控件属性。ConnectionString（连接字符串）和 RecordSource（记录源）属性是 ADO 数据控件最重要的两个属性，通常通过属性页进行设置。右击窗体上的 ADO 数据控件，在弹出的菜单中选择“ADODC 属性”命令，打开“属性页”对话框。在“通用”选项卡中选择“使用连接字符串”，单击“生成”按钮，打开“数据链接属性”对话框。在“提供者”选项卡中选择 Microsoft Jet 4.0 OLE DB Provider，单击“下一步”按钮。在“连接”选项卡中指定数据库文件名（如前面创建的 TestVb.mdb），单击“确定”按钮，完成对 ConnectionString 属性的设置，返回“属性页”对话框，在“记录源”选项卡中设命令类型为 2-adCmdTable，然后在“表或存储过程名称”下拉列表中选择数据表（如“基本情况”）。单击“确定”按钮完成设置。

（3）设置数据绑定控件属性。ADO 数据控件本身不能显示记录集，需要通过绑定具有显示功能的其他控件显示记录集，这些控件称为数据绑定控件，如：文本框、标签、图像（片）框、列表框、组合框、复选框、DataGrid 等。

在属性窗口将数据绑定控件的 DataSource(数据源)属性设为 ADO 数据控件(如 Adodc1)。如果是单字段显示控件（如文本框等），还需要将控件的 DataField（数据字段）属性设置为特定字段。DataGrid 控件属于多字段显示控件，没有 DataField 属性。

（4）编写代码。根据需要，为 ADO 数据控件、数据绑定控件和其他控件编写代码。

本实验项目的界面设计和控件使用比较简单，没有编写代码，只是为了让读者熟悉利用 ADO 数据控件和数据绑定控件创建自定义数据访问窗体的一般过程。

自定义的数据访问窗体如图 12.7 所示。

图 12.7　自定义数据访问窗体

说明：本实验中的两个工程（ADO25.vbp 和 DtGrid25.vbp）是在安装了 VB-SP5 的计算机上编制的，在未安装 VB-SP5 的计算机上运行时可能会出错。若发生错误，请进行以下处理：执行“工程”菜单中的“引用”命令，打开“引用”对话框，将列表中 Microsoft ActiveX Data Objects 2.5 Library 左侧的复选标志（√）清除，然后选中列表中的 Microsoft ActiveX Data Objects 2.0 Library，单击“确定”按钮。