

第 3 章 实验指导

3.1 实验一 Visual Basic 语言环境实验

【实验目的】

1. 熟悉 Visual Basic 6.0 集成开发环境。
2. 学习用窗体和控件建立界面。
3. 学习用属性窗口设置对象的属性。
4. 掌握窗体的概念及相关操作。
5. 熟练掌握程序开发的基本步骤。

【相关知识】

1. 窗体

窗体是 Visual Basic 中的对象，具有自己的属性、事件和方法。窗体的属性包括：

(1) Name。该属性值即对象的名称，这也是所有对象都具有的属性，用来唯一标识该对象。所有的控件在创建时，由 Visual Basic 自动提供一个默认名称。

(2) Height、Width、Top 和 Left。Height 和 Width 决定了控件的高度和宽度，Top 表示控件到窗体顶部的距离，Left 表示控件到窗体左边框的距离。默认单位为 Twip，1Twip=1/20 点=1/1440 英寸=1/567cm。窗体的位置、大小如图 3-1 所示。

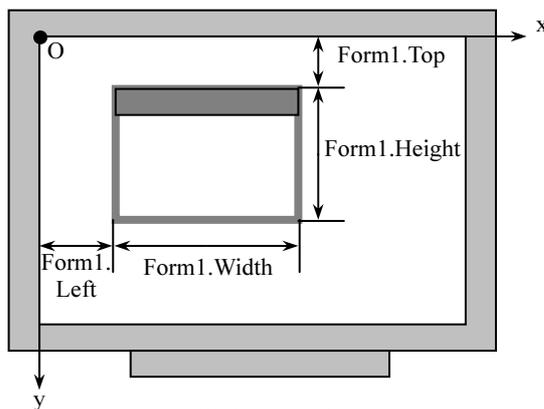


图 3-1 窗体位置、大小

(3) Caption。该属性的值是显示在窗体标题栏上的文字。

(4) BackColor 和 ForeColor。BackColor 用于返回或设置对象的背景颜色，ForeColor 用

于返回或设置在对象里显示的图片 and 文本的前景颜色。

(5) **BorderStyle**。此属性决定边框风格，通过改变 **BorderStyle** 属性值，可以控制窗体如何调整大小。

(6) **MaxButton**、**MinButton** 和 **ControlBox**。**MaxButton** 控制窗体是否可用最大化按钮，**MinButton** 控制窗体是否可用最小化按钮。当 **MaxButton** 或 **MinButton** 属性值为 **True** 时，有最大化按钮或最小化按钮，**False** 则无。**ControlBox** 的值设置为 **True** 时，有控制菜单框；设置为 **False** 时，则无控制菜单框，同时窗体也无最大化按钮和最小化按钮，即使 **MaxButton** 属性和 **MinButton** 属性设置为 **True**。

(7) **Picture**。使用 **Picture** 属性可以指定一幅图片显示在窗体的表面上，作为控件的背景。

(8) **WindowState**。其可能的属性值如下：

- 1) 0: 正常窗口状态，有窗口边界。
- 2) 1: 最小化状态，后台运行，同时在任务栏创建图标。
- 3) 2: 最大化状态，窗口充满整个屏幕。

(9) **Font** 属性组。

FontName 属性是字符型，决定对象上正文的字体（默认为宋体）。

FontSize 属性是整型，决定对象上正文的字体大小（默认为 9 磅）。

FontBold 属性是逻辑型，决定对象上正文是否为粗体（默认为 **False**）。

FontItalic 属性是逻辑型，决定对象上正文是否为斜体（默认为 **False**）。

FontStrikeThru 属性是逻辑型，决定对象上正文是否加一删除线（默认为 **False**）。

FontUnderLine 属性是逻辑型，决定对象上正文是否带下划线（默认为 **False**）。

(10) **AutoRedraw**。该属性决定窗体被隐藏或被另一窗口覆盖之后重新显示时，是否还原该窗体被隐藏部分或覆盖以前的画面。当属性值为 **True** 时，重新还原该窗体以前的画面；当属性值为 **False** 时，则不重画。

窗体常用的事件有 **Load**、**Click**、**DbClick**、**Activate**、**Resize** 等。其中 **Load** 是当窗体被装入工作区时触发的事件，常用来在启动应用程序时对属性和变量进行初始化。**Activate** 是当一个窗体成为活动窗口时所触发的事件，**Resize** 是当窗体大小发生改变时所触发的事件。

窗体常用的方法有 **Print**（打印输出）、**Cls**（清除）、**Show**（显示）、**Hide**（隐藏）以及 **Move**（移动）等。

2. 在窗体上添加控件

窗体是设计 **Visual Basic** 应用程序的一个基本平台，几乎所有的控件都是添加在窗体上的。对于生成的控件，可以调整控件大小、移动控件位置、使用网格精确定位控件；对于多个控件，还可以对它们进行排列、调整大小等操作。另外还允许可视化地创建和编辑控件数组，这对于编组使用选择框、命令按钮等控件提供了很大方便。

3. 设置启动窗体

在默认情况下，应用程序中的第一个窗体被指定为启动窗体。应用程序开始运行时，此窗体就被显示出来。如果想在应用程序启动时显示别的窗体，可以将其他窗体设置为启动窗体。设置方法是在“工程”菜单或者“工程资源管理器”快捷菜单中，选择“属性”命令，弹出“工程属性”对话框，如图 3-2 所示。在“启动对象”下拉列表中选择窗体名称，即可把该窗体设为启动窗体。

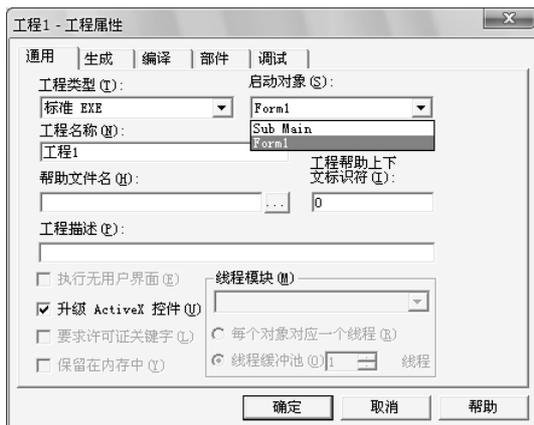


图 3-2 设置启动窗体

【实验例题】

建立一个简单的应用程序，在窗体上输出“大家好！”和“欢迎来到这里！”。

设计思路：通常不直接在窗体上输出文本信息，而是通过窗体上的对象输出文本信息。例如可以用标签输出“大家好！”，用文本框输出“欢迎来到这里！”。因此可以在窗体上放置一个标签、一个文本框和一个命令按钮，单击命令按钮，则在标签上显示“大家好！”，在文本框中显示“欢迎来到这里！”。

设计步骤：

(1) 新建工程。单击 Windows 任务栏中的“开始”→“程序”→“Microsoft Visual Basic 6.0 中文版”→“Microsoft Visual Basic 6.0 中文版”命令，进入 Visual Basic 6.0 集成开发环境，并显示“新建工程”对话框。默认选择是建立“标准 EXE”（即标准工程），单击“打开”按钮，进入设计模式，并自动创建了一个窗体模块 Form1，这就是应用程序的窗体。在这个窗体上添加控件，即可建立应用程序界面。

(2) 设计应用程序界面。

1) 设置窗体属性。

①在属性窗口中双击 Caption 属性条，输入“我的第一个 VB 程序”。

②在属性窗口中选择 BackColor 属性条，然后单击右端的箭头，在所显示的调色板中选择一种颜色（例如浅黄色）。

2) 在窗体上添加控件。

①单击工具箱中的标签（Label）控件类型图标，鼠标指针变为十字形。再在“对象”窗口的窗体上单击并拖动鼠标然后释放，窗体上就会出现一个标签类型的控件对象，同时鼠标指针恢复为箭头形状。

②用同样方法将文本框（TextBox）控件和命令按钮（CommandButton）放置在窗体上。

3) 设置控件属性。在窗体上选中命令按钮，然后在属性窗口中双击 Caption 属性条，输入“显示”。

通过以上的几个简单操作，我们就建立了应用程序界面，如图 3-3 所示。

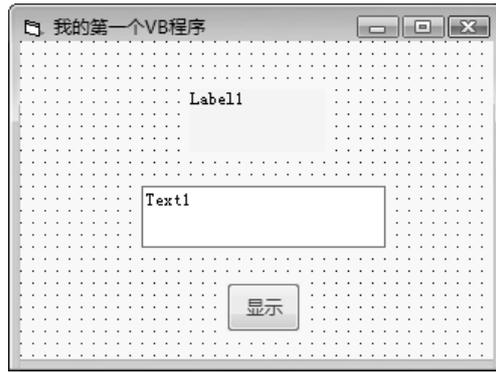


图 3-3 应用程序界面设计

(3) 编写程序代码。程序具有这样的功能：当用户在应用程序窗体中单击“显示”按钮时，窗体中的标签上会显示“大家好！”，文本框中会显示“欢迎来到这里！”。在这里要对“显示”按钮（Command1）进行编程，在窗体上双击“显示”按钮（Command1），弹出代码编辑窗口。在编辑窗口上部左侧的对象列表框中显示对象名为 Command1，在右侧的过程列表框中显示的过程是 Click，如图 3-4 所示。此时在代码编辑窗口中会出现如下代码：

```
Private Sub Command1_Click()

End Sub
```

说明我们是对命令按钮 Command1 的单击事件进行编程。在这两条语句之间输入如下代码：

```
Label1.Caption = "大家好！"
Label1.FontName = "宋体"
Label1.FontSize = 15
Label1.FontBold = True
Text1.Text = "欢迎来到这里！"
Text1.FontName = "黑体"
Text1.FontSize = 16
Text1.FontBold = True
```

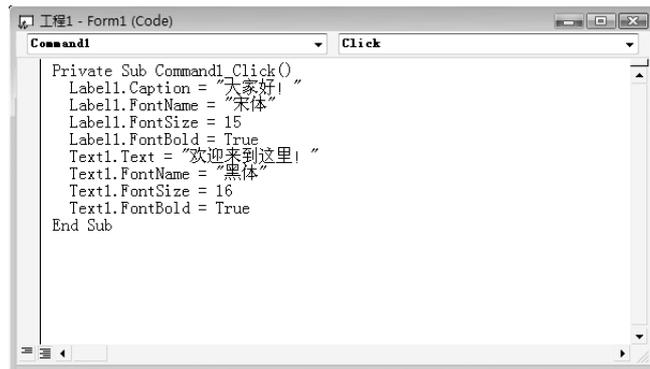


图 3-4 实验一代码编辑窗口

至此，我们完成了对命令按钮 Command1 的编程。单击工具栏中的“启动”按钮，开始

运行程序。单击“显示”按钮，结果如图 3-5 所示。本例仅有一个窗体，因此在保存文件时，只需保存一个窗体文件和工程文件。



图 3-5 实验一运行结果

【实验内容】

1. 设计一个简单的应用程序，要求在窗体上画一个文本框和两个命令按钮，并把命令按钮分别设置为“显示”和“清除”。程序运行后，单击“显示”按钮，在文本框中输出一行文字信息——“我的第一个 VB 应用程序”；单击“清除”按钮，则清除文本框中的内容。

2. 设计一个简易计算器的面板，如图 3-6 所示。



图 3-6 简易计算器面板

3.2 实验二 顺序程序设计

【实验目的】

1. 掌握 Visual Basic 的常用数据类型。
2. 掌握 Visual Basic 的各种运算符及表达式。
3. 学习使用 Visual Basic 内部函数和数据输入、输出方法。
4. 掌握命令按钮、文本框和标签的使用。
5. 设计顺序程序。

【相关知识】

1. 数据类型

Visual Basic 的数据类型分为基本数据类型和自定义类型。基本数据类型有 11 种, 包括字节型、整型、长整型、单精度型、双精度型、货币型、日期型、逻辑型、字符型、对象类型和变体型。按照各数据类型的特点又可以分为数值型、非数值型和变体型。

各个基本数据类型都有确定的表示范围, 在通过赋值语句给各个数据类型的变量赋值时, 要注意所赋予的值要与变量的数据类型保持一致。

2. 常量

常量是指在程序执行的过程中不能改变其值的数据, 分为文字常量和符号常量。文字常量依照其数据类型, 在字符型和日期型文字常量的外面要用相应的定界符括起来。

3. 变量

变量与符号常量的概念类似, 都是用一个标识符表示一个数据。与符号常量不同的是变量的值可以在程序执行的过程中进行修改。

在使用变量之前一般要用声明语句向系统说明其数据类型, 如:

```
Dim x As Integer
```

但是 Visual Basic 也允许用户直接使用变量不进行声明, 这叫做隐式声明, 且所有隐式声明的变量都是 Variant 类型。

为了避免在隐式声明情况下由于变量名输入错误而引起的麻烦, 可以在窗体模块的通用区域中加入语句“Option Explicit”, 以强制要求用户显式地声明变量。

4. 表达式

表达式是用运算符或圆括号将操作数连接起来的一个运算式。Visual Basic 中的表达式可以分为算术表达式、关系表达式、逻辑表达式、字符串表达式、日期表达式和混合表达式。表达式的书写应注意以下问题:

(1) 各个运算符的操作数要完备, 例如 $a+b$ 是错误的而 $3+-2$ 却是正确的。VB 中只有“-” (数值取反运算) 和“Not” (逻辑取非) 是单操作数的运算符。

(2) 乘号以“*”代替且不可省略, 如 3 乘以 x 应写成 $3*x$ 。

(3) 不能出现上下标。例如 x^4 要写成 x^4 , a_1-a_2 应写成 $a1-a2$ 。

(4) 书写繁分式的时候要写在一行。

(5) 用圆括号设定运算的先后顺序, 例如:

$$\frac{b-\sqrt{b^2-4ac}}{2a} \longrightarrow (b-\text{Sqr}(b*b-4*a*c))/(2*a)$$

5. 内部函数

VB 提供了大量的内部函数, 这就给 VB 程序设计带来了极大的方便。内部函数可以分为数学函数、转换函数、字符串函数、日期函数和格式输出函数。使用中要注意各个函数要求的参数数量、参数的数据类型和函数返回值的数据类型。

6. 输入输出数据

使用 VB 控件进行数据的输入输出时要注意各个控件的不同用法, 如标签对象常用其 Caption 属性输出数据; 文本框控件可以利用 Text 属性进行数据的输入或输出。

除了采用控件的方式进行数据的输入和输出外，VB 还提供了 `InputBox` 函数和 `MsgBox` 函数，分别用于输入数据和输出数据。要注意输入输出函数的常用参数和调用格式。

在利用 `Print` 方法实现数据的输出时，一定要明确 `Print` 方法的语法格式：

`[<对象名>].Print [<表达式列表>];;`

其中：

(1) 对象名可以是窗体对象名、图片框对象名、`Printer` 对象、`Debug` 对象，如果缺省该项，则默认对象为当前窗体。

(2) 表达式列表是指一个或多个表达式。不写任何表达式的时候，默认输出一个空行；如果要输出多个表达式，表达式之间要用“,”或“;”分隔符分隔。

如果使用“,”(逗号)分隔符，则分隔符后的表达式将以标准格式在下一个打印区(每行以 14 个字符为单位划分为若干区段，每个区段称之为一个打印区)的起始位置输出。

如果使用“;”(分号)分隔符，则下一个表达式将按紧凑格式紧跟在上一个表达式后面无间隔地输出。

7. 命令按钮控件

命令按钮 (`CommandButton`) 是 Windows 应用程序中最常用的控件，主要是利用其 `Click` 事件作为程序的执行入口。命令按钮控件图标如图 3-7 所示。

(1) 命令按钮的属性。

1) `Caption`。在设计界面时一般重新设置命令按钮的 `Caption` 属性，以说明该按钮的功能。

2) `Enabled`。返回或设置控件是否响应事件，也就是该控件是否可用。

3) `Style`。用来设置命令按钮是标准的还是图形的。

4) `Cancel`。返回或设置一个值，用来指示命令按钮是否为取消命令按钮。

5) `Default`。返回或设置一个值，用来指示命令按钮是否为默认命令按钮。

6) `Visible`。返回或设置一个值，决定控件运行时是否可见。

7) `Value`。把 `True` 赋值给 `Value` 属性，可以引发 `Click` 事件，相当于“虚拟点击”或“软触发”。

8) `ToolTipText`。返回或设置鼠标在命令按钮上停留时的提示文本。这个属性对于图形按钮特别有用，可以提示按钮的功能。

(2) 命令按钮的事件。

命令按钮的常用事件是 `Click` 事件，命令按钮的功能是通过编写 `Click` 事件过程来实现的。用户触发命令按钮事件的方式有以下几种：

1) 单击命令按钮。

2) 在命令按钮获得焦点时，按回车键。

3) 对于设计了访问键的命令按钮，按 `Alt+访问键`。

8. 文本框控件

文本框 (`TextBox`) 控件也称为编辑控件，常用其 `Text` 属性来接收用户输入的信息，也可以输出信息。文本框控件图标如图 3-8 所示。



图 3-7 命令按钮控件图标

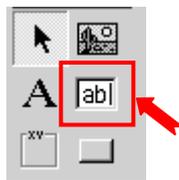


图 3-8 文本框控件图标

(1) 文本框的属性。

1) Text。返回或设置文本框中的文本，该属性是文本框控件最重要的属性之一。可以在设计时设置 Text 属性，也可以在运行时直接在文本框内输入，或通过程序代码对 Text 属性重新赋值。

2) MaxLength。返回或设置在文本框控件中能够输入字符的最大数。

3) MultiLine。返回或设置文本框是否接受多行文本。

4) ScrollBars。返回或设置文本框是否有垂直或水平的滚动条。当文本过长，可能超过文本框的边界时，应该为该控件添加滚动条。

5) PasswordChar。返回或设置一个值，该值指示所键入的字符在文本框中以何种形式显示。

6) 文本编辑属性组。

①SelStart 属性。用来指定选定文本块的起始位置，如果没有选定的文本，则该属性指定光标的位置。若 SelStart 值为 0，所指示的位置是在文本框第一个字符之前；若 SelStart 值等于文本框中文本的长度，所指示的位置是在文本框最后一个字符之后。

②SelLength 属性。用来指定所选的字符个数。

③SelText 属性。用来指定选定的字符，如果没有字符被选定，则是空串。

注意：以上 3 个与文本选定操作有关的属性只能在程序代码中进行操作，设计时不可用。

(2) 文本框的事件。

1) Change。当文本框中的内容（即 Text 属性的值）发生改变时，将触发 Change 事件。

2) KeyPress。在文本框获得焦点并且用户按下了键盘上的按键后，将触发 KeyPress 事件。

9. 标签控件

标签 (Label) 控件主要用来在窗体相对固定的位置上显示文本信息，它不能作为输入信息的界面，通常用于功能注释。标签控件图标如图 3-9 所示。

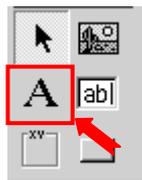


图 3-9 标签控件图标

标签的属性：

(1) Caption。该属性是标签最重要的属性，它用来设定标签上显示的文本内容。

(2) Alignment。决定文本的对齐方式。

(3) Autosize。决定标签是否可以根据文本内容自动调整大小。

(4) WordWrap。值为 True 的时候将根据标题文本内容自动调整标签的高度，但是不会调整标签的宽度；为 False 的时候不会自动调整标签的高度，标签的宽度取决于 AutoSize 属性的设置情况。

若要使 WordWrap 属性生效，必须把 AutoSize 属性的值设为 True。

10. InputBox 函数

VB 程序设计的过程中可以通过 InputBox 函数调用用户输入框，以使用户在程序执行的过程中输入待处理的数据。InputBox 函数的格式为：

InputBox(Prompt[, Title] [, Default] [, XPos] [, YPos] [, HelpFile, Context])

InputBox 函数的返回值就是用户在输入框中输入的内容，但是是字符串型的数据。因此，实际使用的时候常常通过赋值语句把 InputBox()函数的值赋给某个变量。如果这个变量需要的

是一个数值型的值，那么要用 Val()函数把 InputBox 返回的值转换为数值型。

11. MsgBox 函数

通过 MsgBox 函数可以调用消息框，并且可以指定该输入框的提示信息、消息框类型、标题，以及相关的帮助信息，其语法格式如下：

MsgBox(Prompt[, Buttons] [, Title] [, HelpFile, Context])

如果要忽略 MsgBox()函数的返回值，还可以用如下方式调用 MsgBox()函数：

- (1) **MsgBox Prompt[, Buttons] [, Title] [, HelpFile, Context]**
- (2) **Call MsgBox (Prompt[, Buttons] [, Title] [, HelpFile, Context])**

【实验例题】

1. 随机给出两个[10,100]内的整数，让用户计算两个数的和，要求显示用户计算所花费的时间。程序设计界面如图 3-10 所示。

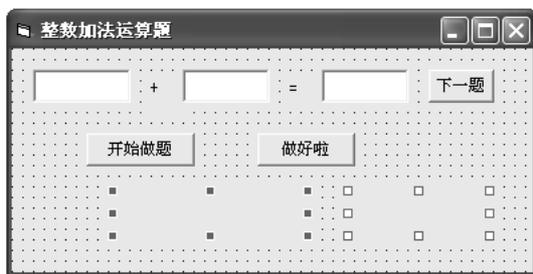


图 3-10 程序界面的布局

设计思路：

使用随机函数 Rnd()可以产生这两个随机整数。记录时间的起始点是在用户单击“开始做题”按钮后，记录时间的结束是在用户单击“做好啦”按钮后，这完全是两个事件过程，要想在“做好啦”按钮单击事件里使用由“开始做题”事件过程所创建的变量，那么这个变量至少要至少是模块级变量（在窗体代码窗口的“通用”区声明）。

在单击“开始做题”按钮后要修改文本框 Text3 的 Locked 属性，使之可以输入数据；在单击“做好啦”按钮后，要把文本框 Text3 的 Locked 属性重新修改为 True，使之不可再次输入；单击“下一题”按钮实际上就是要重新生成数据。

设计步骤：

- (1) 创建新工程，按照图 3-10 的要求添加各控件，各对象的属性设置如表 3-1 所示。

表 3-1 各个对象的属性设置

对象名	属性名	属性值
Form1	Caption	整数加法运算题
Label1	Caption	+
Label2	Caption	=
Label3	Caption	"" (空)
Label4	Caption	"" (空)

续表

对象名	属性名	属性值
Text1	Text	"" (空)
	Locked	True
Text2	Text	"" (空)
	Locked	True
Text3	Text	"" (空)
	Locked	True
Command1	Caption	开始做题
Command2	Caption	做好啦
	Default	True
Command3	Caption	下一题

(2) 编写程序代码如下:

```

Dim start_time As Date, end_time As Date
Private Sub Command1_Click()
    start_time = Now()
    Text3.Locked = False
End Sub

Private Sub Command2_Click()
    Dim total_time As Date
    Text3.Locked = False
    end_time = Now()
    total_time = end_time - start_time
    Label3.Caption = "您总共花费的时间是: " & Format(total_time, "hh:mm:ss")
    Label4.Caption = "正确结果是: " & Val(text1.text)+ Val(text2.text)
End Sub

Private Sub Command3_Click()
    Randomize
    Text1.Text = Int(Rnd * (99 - 10 + 1) + 10)
    Text2.Text = Int(Rnd * (99 - 10 + 1) + 10)
    Text3.Text = ""
    Text3.Locked = True
End Sub

Private Sub Form_Load()
    Randomize
    Text1.Text = Int(Rnd * (99 - 10 + 1) + 10)
    Text2.Text = Int(Rnd * (99 - 10 + 1) + 10)
End Sub

```

程序运行结果如图 3-11 所示。

说明: 本题还不具备自动判断用户计算结果是否正确功能, 请读者在学习到分支结构

时，完善本题，使之能够判断对错。

2. 让用户输入一元二次方程的系数（均不为 0），然后根据这个数据计算一元二次方程的解，程序的界面设计如图 3-12 所示。



图 3-11 程序运行效果

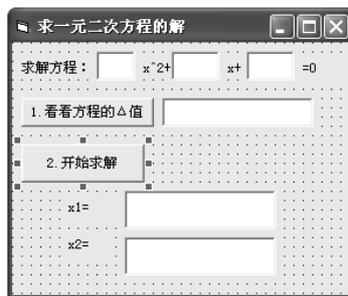


图 3-12 程序的界面布局

设计思路:

(1) 建立数学模型。

本题的数学模型是一元二次方程 $a*x^2+b*x+c=0$ 。

(2) 利用数学模型求解。

用一个变量 d 存放 $\Delta=b^2-4*a*c$ 的值，把这个值显示出来以由用户判断系数是否合理。

根据求根公式可知方程的解为: $x1=(-b+\text{sqr}(d))/(2*a)$; $x2=(-b-\text{sqr}(d))/(2*a)$ 。

设计步骤:

(1) 创建新工程，为窗体添加各控件并设置属性，属性设置如表 3-2 所示。

表 3-2 各属性的设置

对象名	属性名	属性值
Form1	Caption	求一元二次方程的解
Label1	Caption	求解方程:
Label2	Caption	x^2+
Label3	Caption	$x+$
Label4	Caption	$=0$
Command1	Caption	1.看看方程的 Δ 值
Command2	Caption	2.开始求解
Text1	Text	"" (空)
Text2	Text	"" (空)
Text3	Text	"" (空)
Text4	Text	"" (空)
Text5	Text	"" (空)
	Locked	True
Text6	Text	"" (空)
	Locked	true

(2) 为程序设计代码如下:

```
Dim a As Single, b As Single, c As Single, d As Single
Private Sub Command1_Click()
    a = Val(Text1.Text)
    b = Val(Text2.Text)
    c = Val(Text3.Text)
    d = b ^ 2 - 4 * a * c
    Text4.Text = d
End Sub

Private Sub Command2_Click()
    Text5.Text = Format((-b + Sqr(d)) / (2 * a), "0.000")
    Text6.Text = Format((-b - Sqr(d)) / (2 * a), "0.000")
End Sub
```

程序执行的结果如图 3-13 所示。



图 3-13 程序的执行结果

3. 根据用户输入的电压数计算如图 3-14 所示电路中的总电流 I 。已知电路图中电阻 $R_1=150\Omega$, $R_2=230\Omega$, $R_3=260\Omega$ 。程序的界面设计如图 3-15 所示。

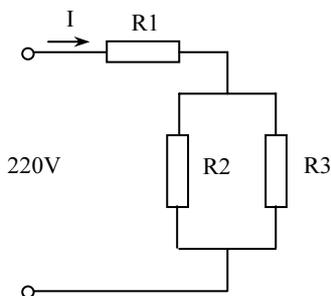


图 3-14 电路图



图 3-15 程序界面布局

设计思路: 首先要做的就是将问题尽可能转化为一个数学模型。这个题目是一个物理问题, 由于电流 $I=U/R$, 电压是用户输入的值, 是已知数, 因此要通过计算求出另一个参数总电阻 R 。由欧姆定律可以推导出总电阻 $R=R_1+R_2*R_3/(R_2+R_3)$ 。

设计步骤:

(1) 创建新工程, 加入两个按钮, 一个用于打开输入对话框, 一个用于开始计算; 输入

的值放在标签里显示，输出结果放在一个文本框里显示。

各个对象的属性设置如表 3-3 所示。

表 3-3 各对象的属性值

对象名	属性	值
Form1	Caption	求电路总电流
Command1	Caption	输入电压值并计算
Text1	Text	"" (空)
Label1	Caption	"" (空)
Label2	Caption	总电流为:

(2) 编写程序代码如下:

```
Private Sub Command1_Click()
    Dim U As Single, R As Single, I As Single
    Dim R1 As Single, R2 As Single, R3 As Single
    R1 = 150
    R2 = 230
    R3 = 260
    U = InputBox("请输入电压 U 的值", "输入电压", 220)
    R = R1 + R2 * R3 / (R2 + R3)
    I = U / R
    Text1.Text = Format(I, "0.000") & "A"
    Label1.Caption = "您输入的电压是" & Format(U, "0.00") & "V"
End Sub
```

程序执行的结果如图 3-16 所示。

4. 设计一个程序，要求用户通过输入对话框输入半径 R ，根据 R 的值，求对应圆的面积、球体的体积（体积公式为 $V=4\pi R^3/3$ ）。程序界面设计如图 3-17 所示。

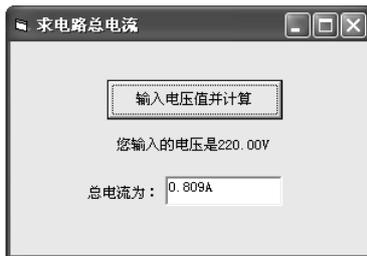


图 3-16 程序运行结果



图 3-17 程序界面布置

设计思路：圆面积的公式为 $S=R^2*\pi$ ；球体积的公式为 $4*\pi*R^3/3$ 。

设计步骤：

(1) 创建新工程，加入用于用户输入的按钮“输入半径并计算”按钮；加入 3 个标签和 3 个文本框控件，分别用于输出结果，其界面布局如图 3-17 所示。

各对象的属性如表 3-4 所示。

表 3-4 各对象的属性设置

对象名	属性名	属性值
Form1	Caption	根据半径求相应值
Command1	Caption	输入半径并计算
Label1	Caption	半径:
Label2	Caption	圆的面积:
Label3	Caption	球的体积:
Text1	Text	"" (空)
	Locked	True
Text2	Text	"" (空)
	Locked	True
Text3	Text	"" (空)
	Locked	True

(2) 编写程序代码如下:

```
Private Const PI As Double = 3.1415926
Private Sub Command1_Click()
    Dim R As Single
    R = Val(InputBox("请输入半径的数值", "输入半径", 15))
    Text1.Text = R
    Text2.Text = PI * R ^ 2
    Text3.Text = 4 * PI * R ^ 3 / 3
End Sub
```

程序执行的结果如图 3-18 所示。



图 3-18 程序执行的结果

【实验内容】

1. 程序界面设计如图 3-19 所示, 编写程序, 把 Text1 中的数据赋给变量 x, Text2 中的

数据赋给变量 y ，并实现变量 x 和 y 值的交换，交换后 x 和 y 的值在各自对应的文本框中显示出来。

2. 编写程序，要求输入同学的姓名和成绩后，在文本框中显示，可以输出成绩的平均值和同学的总人数，并允许在程序界面中调用系统的计算器程序。程序的界面设计如图 3-20 所示。



图 3-19 “交换变量的值”程序的界面布局



图 3-20 “学生成绩处理”程序的界面布局

3. 编写一个华氏温度和摄氏温度之间的转换程序，要求用 `InputBox` 函数输入华氏温度，用 `MsgBox` 函数输出转换后的摄氏温度。转换公式为 $C = \frac{5}{9}(F-32)$ 。F 为华氏温度，C 为摄氏温度。程序界面设计如图 3-21 所示。



图 3-21 “华氏度转换为摄氏度”程序的界面

3.3 实验三 选择程序设计

【实验目的】

1. 掌握单分支结构和双分支结构。
2. 掌握多分支条件语句。
3. 掌握单选按钮、复选框和框架的使用。

【相关知识】

常用分支结构语句的格式如下。

(1) `If...Then...Else` 语句。

If 表达式 Then

语句块 1**Else****语句块 2****End If**

(2) If-Then 语句。

If 表达式 Then**语句块****End If**

(3) If-Then-ElseIf 语句 (多分支结构)。

If 表达式 1 Then**语句块 1****ElseIf 表达式 2 Then****语句块 2**

...

Else**语句块 n****End If**

注意: Else 和 If 之间没有空格。

(4) Select Case 语句 (多分支结构)。

Select Case 表达式**Case 表达式列表 1****语句块 1****Case 表达式列表 2****语句块 2**

...

Case 表达式列表 n**语句块 n****Case Else****语句块 n+1****End Select**

(5) IIf 函数。

result=IIf(条件,True 部分,False 部分)

(6) 单选按钮控件。

单选按钮 (OptionButton) 控件图标如图 3-22 所示, 用来表示一组互斥的选项。单选按钮控件成组出现, 而且只能有一个按钮被选中。

单选按钮控件的常用属性如下:

1) Caption。用来设置单选按钮控件的显示文字。

2) Value。返回或设置单选按钮是否被选中, True 表示选中, False 表示未选中。

3) Style。返回或设置控件的外观。0 表示 Standard, 1 表示

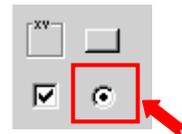


图 3-22 单选按钮控件图标

Graphical。

4) **Alignment**。该属性值为 0 时（默认值），单选按钮在标题文字左边；属性值为 1 时，单选按钮显示在标题文字的右边。

(7) 复选框控件。

复选框 (CheckBox) 控件图标如图 3-23 所示。在同一组选项中，用户可以选择多个选项。

复选框控件的常用属性如下：

1) **Value** 属性。返回或设置复选框的状态信息。0 表示未选中，1 表示选中，2 表示禁止使用。

2) 复选框的 **Caption** 属性、**Style** 属性和 **Alignment** 属性与单选按钮控件相同。

(8) 框架控件。

框架 (Frame) 控件图标如图 3-24 所示。它是一个容器控件，其主要作用是对窗体上的控件进行分组。框架内控件的创建方法有两种：一是单击工具箱上的图标，在框架中的适当位置拖曳出适当大小的控件；二是将控件复制到剪贴板，然后粘贴（按 **Ctrl+V** 键）到框架中。



图 3-23 复选框控件图标



图 3-24 框架控件图标

【实验例题】

1. 从键盘上输入代表年、月、日的 3 个数字，判断该年是否为闰年，并根据月份判断该月有多少天，根据日来判断是星期几。

设计思路：闰年的计算依据是年号能够被 4 整除但不能够被 100 整除，或能够被 400 整除。

设计步骤：

(1) 创建用户界面，在窗体上创建 3 个文本框控件 Text1、Text2、Text3，6 个标签控件 Label1~Label6，以及 2 个命令按钮。

(2) 设置属性。各对象的属性设置如表 3-5 所示。

表 3-5 对象属性设置

对象名	属性	属性值
Form1	Caption	日期判断
Text1	Text	"" (空)
Text2	Text	"" (空)
Text3	Text	"" (空)
Label1	Caption	"" (空)
Label2	Caption	"" (空)
Label3	Caption	"" (空)
Label4	Caption	年份
Label5	Caption	月份

续表

对象名	属性	属性值
Label6	Caption	日期
Command1	Caption	判断
Command2	Caption	清除

(3) 编写程序代码如下:

```

Private Sub Command1_Click()
    Dim year As Integer, month As Integer, day As Integer
    Dim leapyear As Boolean
    year = Val(Text1.Text)
    month = Val(Text2.Text)
    day = Val(Text3.Text)
    '判断是否为闰年
    If (year Mod 4 = 0 And year Mod 100 <> 0) Or (year Mod 400 = 0) Then
        leapyear = True
        Label1.Caption = "该年为闰年"
    Else
        leapyear = False
        Label1.Caption = "该年不是闰年"
    End If
    Select Case month
    '判断月份天数
        Case 1, 3, 5, 7, 8, 10, 12
            Label2.Caption = "31 天"
        Case 4, 6, 9, 11
            Label2.Caption = "30 天"
        Case 2
            If leapyear Then
                Label2.Caption = "29"
            Else
                Label2.Caption = "28"
            End If
    End Select
    dd$ = Str$(year) & "," & Str$(month) & "," & Str$(day)
    '判断星期几
    Label3.Caption = WeekdayName(Weekday(dd$))
End Sub

Private Sub Command2_Click()
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""

```

```

Label1.Caption = ""
Label2.Caption = ""
Label3.Caption = ""
End Sub

```

运行结果如图 3-25 所示。



图 3-25 程序执行的结果

思考：本题还可以根据输入的月份判断是什么季节？（3~5 月为春季、6~8 月为夏季、9~11 月为秋季、11~2 月为冬季。）

2. 设计个人信息输入窗口，使用单选按钮输入性别和学历，使用复选框输入个人爱好。

设计步骤：

(1) 创建用户界面，在窗体上创建 1 个文本框控件 Text1，4 个标签控件 Label1~Label4，4 个单选按钮 Option1~Option4，4 个复选框 Check1~Check4，3 个框架 Frame1、Frame2、Frame3，一个命令按钮 Command1。

(2) 设置属性。各对象的属性设置如表 3-6 所示。

表 3-6 对象属性设置

对象名	属性	属性值
Form1	Caption	个人信息
Text1	Text	"" (空)
Label1	Caption	姓名：
Label2	Caption	性别：
Label3	Caption	民族：
Label4	Caption	个人信息：
Option1	Caption	男
Option2	Caption	女
Option3	Caption	汉族
Option4	Caption	少数民族
Check1	Caption	音乐
Check2	Caption	跑步
Check3	Caption	小说

续表

对象名	属性	属性值
Check4	Caption	上网
Frame1	Caption	"" (空)
Frame2	Caption	"" (空)
Frame3	Caption	爱好
Command1	Caption	确定

(3) 编写程序代码如下:

```

Private Sub Command1_Click()
    Dim s1 As String, s2 As String, s3 As String, s4 As String
    Dim s As String, m As String
    If Text1.Text = "" Then
        m = InputBox("忘记输入姓名", "提示", "输入姓名")
        Text1.Text = m
    End If
    s1 = Text1.Text + "," '存储姓名信息
    s2 = IIf(Option1, "男,", "女,")
    s3 = IIf(Option3, "汉族;", "少数民族;") '存储性别信息
    s4 = "爱好:"
    If Check1.Value = 1 Then '存储个人爱好信息
        s4 = s4 + Check1.Caption + ","
    End If
    If Check2.Value = 1 Then
        s4 = s4 + Check2.Caption + ","
    End If
    If Check3.Value = 1 Then
        s4 = s4 + Check3.Caption + ","
    End If
    If Check4.Value = 1 Then
        s4 = s4 + Check4.Caption
    End If
    s = s1 + s2 + s3 + s4
    Label4.Caption = s
    Text1.SetFocus '光标置于文本框, 准备下一次输入
End Sub
Private Sub Text1_Change()
    Label4.Caption = ""
End Sub

```

运行结果如图 3-26 所示。

【实验内容】

1. 输入三个数进行排序, 运行界面如图 3-27 所示。



图 3-26 程序的执行结果

2. 设计一个简易计算器的程序。要求：①输入两个操作数和一个操作符（+、-、×、/、\、mod、^），根据操作符决定所进行的计算；②如果操作数中没有输入数字，用 MsgBox 提示出错信息，运行界面如图 3-28 所示。

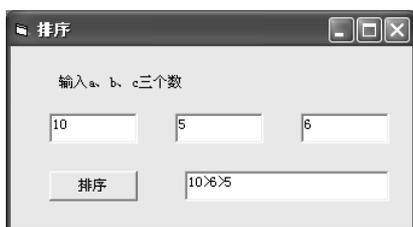


图 3-27 “排序”程序运行结果



图 3-28 “简易计算器”程序运行结果

3. 设计一个应用程序，界面设计如图 3-29 所示。要求当用户在两组不同的单选按钮和一组复选框上作出选择后，文本框中正文的字体、字号和风格能够发生相应的变化。



图 3-29 设置字体运行结果

3.4 实验四 循环程序设计

【实验目的】

1. 掌握用 While-Wend 语句、Do-Loop 语句和 For-Next 语句实现循环的方法。
2. 掌握如何在程序设计中用循环的方法实现一些常用算法。
3. 掌握循环结构程序设计的基本方法，能设计简单的循环结构程序。
4. 学会结合循环语句和图片框、图像框和计时器等控件设计简单的程序。

【相关知识】

常用的循环语句如下：

(1) For 语句。

```
For 循环变量 = 初值 To 终值 [Step 步长]
    语句块
    [Exit For]
    语句块
Next 循环变量
```

(2) Do-Loop 语句。

```
Do While/Until 条件
    语句块
[Exit Do]
语句块
Loop
或者
Do
    语句块
[Exit Do]
语句块
Loop While / Until 条件
```

(3) While-Wend 语句。

```
While <条件>
    语句块
Wend
```

(4) Exit Do 语句。

Exit Do 语句必须放置在 Do-Loop 语句的循环体中。执行到 Exit Do 时，程序会立即结束本层循环，跳到 Loop 后执行下面的语句。

(5) Exit For 语句。

同 Exit Do 类似，Exit For 语句用在 For-Next 循环中，用于强制跳出循环结构。

(6) 循环嵌套。

外循环必须完全包含内循环，不能交叉。循环的转移遵循“允许循环内转移，允许转出，禁止转入”的原则。

(7) 图像框控件

图像框 (Image) 控件用于显示图像，不支持图形方法，也不能当作容器来使用。图像框控件图标如图 3-30 所示。

1) Picture 属性。该属性为图像框控件指定要显示的图像。

2) Stretch 属性。如果属性值为 True，当所显示图像的原始大小与控件大小不相同，会缩放图像来填充整个控件；如果属性值为 False (默认值)，图像会以原始大小显示。

(8) 图片框控件。

图片框 (PictureBox) 控件用于显示图片，此外还支持 Print 方法和绘图方法，可以绘制自定义的函数图、曲线图和图片等，还可以作为其他控件的容器。

图片框控件图标如图 3-31 所示。

常用属性如下：

1) Picture 属性。与图像框控件的 Picture 属性相同。

2) AutoSize 属性。属性值为 True 时，图片框的边界会随着所



图 3-30 图像框控件图标



图 3-31 图片框控件图标

载入图片大小的变化而变化。

(9) 计时器控件。

计时器 (Timer) 控件又称时钟、定时器控件, 时钟控件能在程序运行过程中不断地积累时间, 当达到给定的时间间隔时, 自动引发 Timer 事件。它常常用于编写计时、动画等程序, 在程序运行阶段不可见。计时器控件图标如图 3-32 所示。

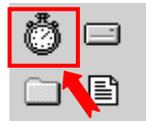


图 3-32 计时器控件图标

1) 常用属性。

①Interval 属性。取值范围为 0~65535, 单位为 ms(0.001s), 表示计时间隔。例如, 如果希望每 1s 产生一个计时器事件, 那么 Interval 属性值应设为 1000。这样每隔 1000ms 引发一次 Timer

事件, 从而执行相应的 Timer 事件过程。若将 Interval 属性设置为 0 或负数, 则计时器停止工作。

②Enabled 属性。只要时钟控件的 Enabled 属性值被设置为 True, 而且 Interval 属性值大于 0, 则计时器开始工作。把属性值设置为 False, 可使时钟控件无效, 即计时器停止工作。

2) 常用事件。

时钟控件只能响应 Timer 事件。该事件以 Interval 属性指定的时间间隔发生, 可以把需要定时执行的操作放在该事件过程中。

(10) 图形和绘图操作。

这里只介绍实验涉及的一些概念, 其他知识可参考相关资料。

(1) RGB 函数。RGB 颜色模型中的任何颜色都是由红 (R)、绿 (G)、蓝 (B) 三种颜色按不同比例混合生成的。因此要设定一种颜色, 只需指定其红、绿、蓝分量的大小即可。RGB(Red,Green,Blue)是内部函数, 返回一个由长整型数表示的颜色值。此函数有三个整型参数, 取值都是 0~255, 如 RGB(0,0,0)返回黑色, RGB(255,0,0)返回红色, RGB(255,255,0)返回黄色。RGB 函数可以返回 $255 \times 255 \times 255$ 种颜色。

(2) Line 方法。Line 方法可以在窗体或图片框上绘制一条直线或一个矩形。语法格式如下:

Object.Line [[step](x1,y1)]-[step](x2,y2),[color],[B|F]]

1) Object 为窗体或图片框对象名。

2) [step](x1,y1)指定起点坐标, [step](x2,y2)指定终点坐标。如果有参数 B, 则绘制以给定两点为对角的矩形, 否则画出以给定两点为端点的直线段。

3) color 指定直线或矩形边框的颜色。如果有参数 F, 则用边框颜色填充矩形。无参数 B 时, 不能使用参数 F。

4) 如果(x1,y1)参数前有 step 关键字, 表示该坐标是相对于当前作图位置的偏移量。如果(x2,y2)参数前有 step 关键字, 表示是以起点为基准的相对坐标。

(3) Circle 方法。Circle 方法可以在窗体或图片框上绘制圆形、椭圆或弧。语法格式如下:

Object.Circle [[step](x,y),radius,[color],[start],[end],[aspect]]

1) [step](x,y)指定圆心或椭圆中心的坐标。

2) radius 指定圆的半径或椭圆的长半轴。

3) color 指定线条颜色。

4) start 与 end 指定弧的起止角度（单位是弧度），如果被省略，则绘制出完整的圆或椭圆。

5) aspect 指定圆度（垂直半轴与水平半轴长度之比），当它为 1 或省略时，绘出一个正圆，其他值时为椭圆。

【实验例题】

1. 求 1~100 之间的所有素数之和。

分析：显然应采用穷举法来解决这个问题。在程序中设置一个二重循环，其中外层循环列举 1~200 之间所有的自然数，内层循环判断是否是素数。

程序代码如下：

```
Private Sub Command1_Click()
    Sum = 0
    n = 0
    For i = 2 To 100
        Flag = True
        For j = 2 To i - 1      'i-1 也可以用"i/2,sqr(i)"代替
            If i Mod j = 0 Then
                Flag = False
                Exit For      '跳出内层循环
            End If
        Next
        If Flag Then
            Print i,
            Sum = Sum + i
            n = n + 1
            If n Mod 5 = 0 Then Print '每输出 5 个数换行
        End If
    Next
    Print "Sum=", Sum
End Sub
```

运行结果如图 3-33 所示。



图 3-33 “求素数之和”程序

2. 利用循环和图像框控件输出一个月的日历（以 2010 年 1 月为例）。

设计步骤：

- (1) 新建工程，在窗体中添加一个命令按钮 Command1 和一个图像框 Picture1。
- (2) 将窗体的 Caption 设置为“日历”。

(3) 将命令按钮的 Caption 设置为“打印”。

(4) 编写程序代码如下：

```
Private Sub Command1_Click()
    Dim month As Integer, day As Integer, week As Integer
    Dim s() As String
    Dim i%
    week = 5      '2010 年 1 月 1 日是星期五
    day = 31     '2010 年 1 月有 31 天
    s = Split("日 一 二 三 四 五 六")
    For i = 1 To 7
        Picture1.Print Tab(4 * i - 1); s(i - 1);
    Next i
    For i = 1 To day
        If (i + week) Mod 7 = 0 Then
            Picture1.Print Tab(4 * 7 - 2); i;
        Else
            Picture1.Print Tab(4 * ((i + week) Mod 7) - 2); i;
        End If
    Next i
End Sub

Private Sub Form_Load()
    Picture1.FontSize = 12
End Sub
```

运行结果如图 3-34 所示。



图 3-34 “日历”程序

3. 编程实现图片的移动和缩放。

设计步骤：

(1) 创建用户界面。新建一个工程，在 Form1 中放置 1 个图片框控件、5 个命令按钮控件。

(2) 设置属性。各对象属性如表 3-7 所示。

表 3-7 对象属性设置

对象	属性	属性值
Form1	Caption	图像操作
Picture1	Picture	d:\图片\pig.bmp
Command1	Caption	左移
Command2	Caption	右移
Command3	Caption	放大
Command4	Caption	缩小
Command5	Caption	还原

(3) 编写程序代码如下:

```
Public cmdIndex As Integer
Private Sub Command1_Click()
    cmdIndex = 1
    Timer1.Enabled = True
End Sub

Private Sub Command2_Click()
    cmdIndex = 2
    Timer1.Enabled = True
End Sub

Private Sub Command3_Click()
    cmdIndex = 3
    Timer1.Enabled = True
End Sub

Private Sub Command4_Click()
    cmdIndex = 4
    Timer1.Enabled = True
End Sub

Private Sub Command5_Click()
    cmdIndex = 5
    Timer1.Enabled = True
End Sub

Private Sub Timer1_Timer()
    Select Case cmdIndex
        Case 1
            Picture1.Move (Picture1.Left - 100)
        Case 2
            Picture1.Move (Picture1.Left + 100)
        Case 3
            Picture1.Height = Picture1.Height * 1.1
            Picture1.Width = Picture1.Width * 1.1
        Case 4
            Picture1.Height = Picture1.Height * 0.9
            Picture1.Width = Picture1.Width * 0.9
        Case 5
            Timer1.Enabled = False
            Picture1.Left = 1320
            Picture1.Width = 2775
            Picture1.Height = 3255
    End Select
End Sub
```

程序运行结果如图 3-35 所示。



图 3-35 图片操作界面

【实验内容】

1. 输出 100~999 之间的水仙花数。水仙花数即个位数的立方加十位数的立方加百位数的立方，正好等于该数本身的数。
2. 设计一条滚动的广告字幕。
3. 设计一个闹钟。要求能显示当前时间，设置定时时间，当设定的时间到来时能发出报警声，程序界面如图 3-36 所示。



图 3-36 “闹钟”程序界面

3.5 实验五 数组程序设计

【实验目的】

1. 熟练掌握静态数组。
2. 掌握动态数组。
3. 了解控件数组。
4. 掌握列表框和组合框。

【相关知识】

1. 固定数组

Visual Basic 有两种类型的数组，即固定大小的静态组和运行时大小可变的动态数组。

数组必须先定义，后使用。固定数组的定义形式为：

Dim 数组名(下标 1 [,下标 2]...)[As 类型]

其中下标必须是常数，格式为：[下界 to]上界。下界的默认值为 0，例如：

```
Dim x(5) as Integer
```

```
Dim score(3,4) As Integer
```

在具体操作时是针对每个数组元素进行的，因此数组的引用单位是元素。

2. 动态数组

动态数组的定义形式为：

Dim 数组名() [As 类型]

此时数组的大小是不确定的，可用 ReDim 语句指定数组的大小。

3. 自定义数据类型

除了标准数据类型之外，用户还可以自定义数据类型。定义形式为：

Type 自定义类型名

成员 1 As 类型

成员 2 As 类型

...

End Type

例如对于一个学生的学号、姓名、性别、年龄、成绩等数据，常常需要把这些数据定义成一个 Student 类型。

```
Type Student
```

```
    Xh As String
```

```
    Xm As String
```

```
    Xb As String
```

```
    Nl As Integer
```

```
    Score As Single
```

```
End Type
```

4. 控件数组

控件数组由一组相同类型的控件组成，它们共用一个控件名，具有相同的属性，但通常有不同的属性值。当建立控件数组时，系统给每个元素赋一个唯一的索引号 (Index)，即下标，下标值由 Index 属性指定。通过属性窗口的 Index 属性，可以知道该控件的下标是多少，例如第 1 个元素下标是 0，第 2 个元素下标是 1，依此类推。

控件数组共享同样的事件过程。例如控件数组 Command 有 3 个命令按钮，不论单击哪个命令按钮，都会调用同一个事件过程。为了区分控件数组中的各个元素，系统会把下标值传送给一个过程。例如单击控件数组 Command 中的任意 1 个命令按钮时，调用的事件过程如下：

```
Private Sub CmdNumber_Click(Index As Integer)
```

```
End Sub
```

5. 列表框

列表框 (ListBox) 是以列表形式显示一系列数据，供用户选择。如果选项较多，超出了列表框的显示区域，会自动加上滚动条。列表框最主要的特点是只能选择，而不能直接添加或修改其中的内容。列表框控件的图标如图 3-37 所示，默认的列表框控件名为 ListX (X 为阿拉伯数字 1、2、3 等)。



图 3-37 列表框控件图标

常用属性:

(1) List。该属性是一个字符数组,存放列表框的项。

(2) ListIndex。属性值表示执行时选中的列表项序号,如果第1个项目被选中,则此属性值为0;第2个项目被选中,则此属性值为1,……如果没有项目被选中,则 ListIndex 的值为-1。

(3) ListCount。属性值表示列表框中项目的数量, ListCount-1 表示列表中最后一项的序号。使用 List 属性和 0~ListCount-1 之间的一个索引号,可以获得任何一个列表项的内容。

(4) Selected。该属性是一个逻辑数组,其元素对应列表框中相应的项,表示对应的项是否被选中。如果 Selected(i)的值为 True,表示第 i+1 项被选中。

(5) Text。属性值是被选中列表项的文本内容。

列表框中的列表项可以在设计状态通过 List 属性设置,也可以在程序中用 AddItem 方法来添加。使用 RemoveItem 方法或 Clear 方法,可以删除列表项。例如在窗体上建立一个列表框 List1,两个命令按钮 Command1、Command2,则用下面的过程可以向列表框添加和删除项目。

```
Private Sub Command1_Click()
    List1.AddItem "One", 0
End Sub
```

```
Private Sub Command2_Click()
    List1.RemoveItem 0
End Sub
```

单击命令按钮 Command1,可以把字符串"One"加到列表框 List1 的开头;而单击命令按钮 Command2,则可删除列表框的第一项。

6. 组合框

组合框(ComboBox)控件具有文本框和列表框的大部分属性,它可以像列表框一样,让用户通过鼠标选择需要的项目;也可以像文本框那样,用键入的方法选择项目。组合框控件的图标如图 3-38 所示。

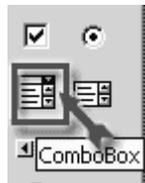


图 3-38 组合框控件图标

组合框与列表框具有相同的属性,例如 List、ListIndex、ListCount、Selected、Sorted、Text、MultiSelect 等。组合框的风格是由 Style 属性值决定的,为 0 时是下拉式组合框(默认状态),为 1 时是简单组合框,为 2 时是下拉列表框。

组合框响应的事件依赖于其 Style 属性。只有简单组合框才能接收 DblClick 事件,下拉组合框与下拉列表框可接收 Click 事件和 DropDown 事件。下拉组合框和简单组合框可以在文本区输入文本,因此能够接收 Change 事件。

7. 直线控件

直线(Line)控件并不常用,它的主要作用是在窗体上显示一条直线段,对控件进行视觉上的分组。

直线控件的常用属性如下:

(1) X1、Y1、X2、Y2。这四个属性决定了直线控件的两个端点在窗体上的位置。

(2) **BorderStyle**。此属性决定直线控件的线形，即线条样式。

8. 形状控件

形状 (Shape) 控件图标如图 3-39 所示，它可以显示多种几何形状。

形状控件常用属性如下：

(1) **Shape**。决定形状控件以何种形状显示，其取值及含义如下所示：

- 0 或 **VbShapeRectangle**：矩形。
- 1 或 **VbShapeSquare**：正方形。
- 2 或 **VbShapeOval**：椭圆形。
- 3 或 **VbShapeCircle**：圆形。
- 4 或 **VbShapeRoundedRectangle**：圆角矩形。
- 5 或 **VbShapeRoundedSquare**：圆角正方形。

(2) **BorderStyle**。该属性定义图形边框样式。

(3) **FillStyle**。用于指定图形的填充样式。

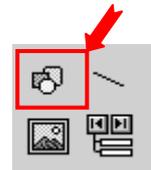


图 3-39 形状控件图标

【实验例题】

1. 通过 **Array** 函数输入 10 位学生的成绩，并输出最高分和最低分。

程序代码如下：

```
Option Base 1
Dim a As Variant
Private Sub Command1_Click()
    Dim max As Integer, min As Integer
    Dim i As Integer
    max = a(1)           '设定初值
    min = a(1)
    For i = 2 To 10
        If max < a(i) Then    '找最高分
            max = a(i)
        End If
        If min > a(i) Then    '找最低分
            min = a(i)
        End If
    Next i
    Picture1.Print "最高分: " + Str(max) + Chr(13) + "最低分: " + Str(min)
End Sub

Private Sub Form_Load()
    a = Array(78, 95, 85, 77, 89, 92, 73, 85, 75, 96)
End Sub
```

程序运行结果如图 3-40 所示。



图 3-40 程序的执行结果

2. 矩阵转置，即将二维数组的行、列元素互换。

设计步骤：

(1) 创建用户界面。新建一个工程，在 Form1 中放置 2 个标签控件、2 个文本框控件和 1 个命令按钮控件，如图 3-41 所示。



图 3-41 “转置操作”程序界面设计

(2) 设置属性。各控件属性如表 3-8 所示。

表 3-8 对象属性设置

对象	控件名称	属性名称	属性值
窗体	Form1	Caption	转置操作
标签 1	Label1	Caption	转置前
标签 2	Label2	Caption	转置后
文本框 1	Text1	Text	"" (空)
		MultiLine	True
文本框 2	Text2	Text	"" (空)
		MultiLine	True
命令按钮 1	Command1	Caption	转置

(3) 编写程序代码如下：

```
Private Sub Command1_Click()
    Dim i, j As Integer
    Dim myString As String
    myString = ""
    Const n = 3
```

```

Const m = 4
Dim a(n, m), b(m, n) As Integer
Randomize
For i = 1 To n
    For j = 1 To m
        a(i, j) = Int(Rnd() * 90) + 10
        myString = myString & Str(a(i, j)) & " "
    Next j
    myString = myString & Chr(13) + Chr(10)
Next i
Text1.Text = myString
For i = 1 To n
    For j = 1 To m
        b(j, i) = a(i, j)
    Next j
Next i
myString = ""
For i = 1 To m
    For j = 1 To n
        myString = myString & Str(b(i, j)) & " "
    Next j
    myString = myString & vbCrLf
Next i
Text2.Text = myString
End Sub

```

运行结果如图 3-41 所示。

3. 利用控件数组完成实验四中的例 3，编程实现图片的移动和缩放。

(1) 创建用户界面。新建一个工程，在 Form1 中放置 1 个图片框控件、1 个命令按钮控件数组，其中包括 5 个按钮。

(2) 设置属性。各对象属性如表 3-9 所示。

表 3-9 对象属性设置

对象	属性	属性值
Form1	Caption	图像操作
Picture1	Picture	d:\图片\pig.bmp
Command1(0)	Caption	左移
Command1(1)	Caption	右移
Command1(2)	Caption	放大
Command1(3)	Caption	缩小
Command1(4)	Caption	还原

(3) 编写程序代码如下：

```

Public cmdIndex As Integer
Private Sub Command1_Click(Index As Integer)
    cmdIndex = Index

```

```

    Timer1.Enabled = True
End Sub

Private Sub Timer1_Timer()
    Select Case cmdIndex
        Case 0
            Picture1.Move (Picture1.Left - 100)
        Case 1
            Picture1.Move (Picture1.Left + 100)
        Case 2
            Picture1.Height = Picture1.Height * 1.1
            Picture1.Width = Picture1.Width * 1.1
        Case 3
            Picture1.Height = Picture1.Height * 0.9
            Picture1.Width = Picture1.Width * 0.9
        Case 4
            Timer1.Enabled = False
            Picture1.Left = 1320
            Picture1.Width = 2775
            Picture1.Height = 3255
    End Select
End Sub

```

程序运行结果同图 3-35，在此省略。

4. 编写程序删除一个字符串中的所有数字字符。

方法一：

```

Private Sub Command1_Click()
    Dim s As String, t As String, i%, j%
    s = Text1.Text
    t = ""
    For i = 1 To Len(s)
        If Not (Mid(s, i, 1) >= "0" And Mid(s, i, 1) <= "9") Then
            t = t + Mid(s, i, 1)
        End If
    Next i
    Picture1.Print t
End Sub

Private Sub Form_Load()
    Text1.Text = ""
End Sub

```

程序运行结果如图 3-42 所示。

方法二：

```

Private Sub Command1_Click()
    Dim s As String, t As String, i%, j%

```



图 3-42 程序的执行结果

```

s = Text1.Text
t = ""
j = Len(s)
i = 1
Do While i <= j
    If Mid(s, i, 1) >= "0" And Mid(s, i, 1) <= "9" Then
        s = Left(s, i - 1) + Right(s, j - i)
        j = Len(s) '重新计算新串的长度
    Else
        i = i + 1 '非数字字符时才将 i 加 1, 使得 i 指向下一个字符
    End If
Loop
Picture1.Print s
End Sub

Private Sub Form_Load()
    Text1.Text = ""
End Sub

```

【实验内容】

1. 设计一个程序, 找出数组中的最大值、最小值, 并计算所有元素的平均值。
2. 设计一个通讯录程序, 用自定义数据类型定义通讯录各项的信息, 如姓名、电话、手机、地址等。在文本框中输入各个信息内容, 在列表框中显示各个人员的信息, 程序界面设计如图 3-43 所示。
3. 设计一个日出和日落的动画场面, 界面设计如图 3-44 所示。



图 3-43 通讯录程序界面布局

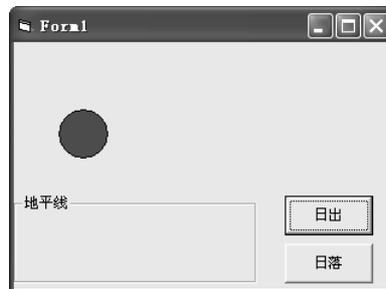


图 3-44 “日出日落”动画程序界面布局

3.6 实验六 过程程序设计

【实验目的】

1. 掌握子过程的定义和调用方法。
2. 掌握函数过程的定义和调用方法。
3. 理解嵌套调用和递归调用。
4. 理解作用域和生存期。

5. 掌握滚动条控件的使用方法。

【相关知识】

1. 子过程的建立

可以被其他程序或主程序调用，并且可以完成特定功能的一段程序称为子过程，建立的语法格式如下：

```
[Public|Private] Sub 子过程名([参数列表])  
    变量定义语句  
    执行语句  
    [Exit Sub]  
End Sub
```

2. 子过程的调用

子过程调用的语法格式如下：

```
Call 过程名([参数列表])  
过程名 [参数列表]
```

3. 函数过程

Visual Basic 不仅有内部函数，还可以编写自己的函数过程。语法格式如下：

```
[Public|Private] Function 过程名([形参列表]) [As 类型]  
    <语句组 1>  
    [函数名=返回值]  
    [Exit Function]  
    <语句组 2>  
    [函数名=返回值]  
End Function
```

4. 函数过程的调用

函数过程的调用的语法格式如下：

```
变量=函数过程名([实参列表])
```

5. 事件过程

事件过程的语法与子过程非常相似，其定义形式如下：

```
Private Sub 对象名_事件名([形参列表])  
    变量定义语句  
    执行语句  
End Sub
```

6. 参数传递的方式

VB 语言的参数传递有传值、传引用和传数组三种方式，其中传数组方式可以归结为传引用方式的一种特例。

值传递是单向的，形参的值发生变化，对实参毫无影响。传引用将实参的地址传给了形参，形参和实参实际上使用的是同一个存储单元，这是默认的参数传递方式。所以传引用方式与传值方式最大的区别在于，传引用调用时形参的值发生变化，会使实参的值也同步发生变化。传数组把数组名作为实参传递给形参数组，使得形参数组和实参数组的起始地址相同。

按引用传递参数使过程用变量的引用去访问实际变量的内容，如果过程中不需改变参数的值，应尽量采用按值传递。

7. 嵌套调用和递归调用

在程序中调用一子过程，而在子过程中又调用另外的子过程，称为嵌套调用，过程在执行期间又调用自己，称为递归调用。递归调用中应注意递归的结束条件和递归的表达式。

8. 作用域和生存期

VB 实体的作用域由小到大可以划分为局部作用域、模块作用域和全局作用域三个层次。变量按照生存期来划分，则可以分为动态变量和静态变量。

9. 滚动条

滚动条控件图标如图 3-45 所示。常用属性如下：



图 3-45 滚动条控件图标

(1) Value。返回或设置滚动条上的滚动滑块所处的位置。

(2) Max 和 Min。返回或设置 Value 属性的最大值和最小值。

(3) LargeChange。返回或设置当用户单击滚动箭头和滑块之间的区域时，Value 属性值的改变量。

(4) SmallChange。返回或设置当用户单击滚动箭头时 Value 属性值的改变量。

当拖动滑块时会触发 Scroll 事件，当改变 Value 属性时会触发 Change 事件。

10. 直线

直线控件用于在窗体上绘制直线，在 VB 的工具箱中，直线控件的图标如图 3-46 所示。

11. 形状

形状 (Shape) 控件用于在窗体上绘制简单的几何图形，它的初始状态是一个矩形。在 VB 的工具箱中，形状控件的图标如图 3-47 所示。



图 3-46 直线图标



图 3-47 形状图标

Shape 是形状控件最重要的属性之一，用来确定具体的图形。

【实验例题】

1. 求 $1!+2!+3!+\dots+n!$ 的值。

设计步骤：

(1) 创建用户界面，添加 2 个标签 Label1、Label2，1 个命令按钮 Command1 和 2 个文本框 Text1 和 Text2。

(2) 设置属性。各对象的属性设置如表 3-10 所示。

表 3-10 对象属性设置

对象名	属性	属性值
Text1	Text	"" (空)
Text2	Text	"" (空)
Label1	Caption	n:
Label2	Caption	$1!+2!+\dots+n!=$
Command1	Caption	计算

(3) 编写事件过程。

```
Option Explicit
Private fact As Long
Private Sub Command1_Click()
    Dim n As Integer, i As Integer
    Dim sum As Double
    n = Val(Text1.Text)
    sum = 0
    For i = 1 To n
        Call fac(i)
        sum = sum + fact
    Next i
    Text2.Text = Str(sum)
End Sub

Private Sub fac(ByVal x As Integer)
    Dim i As Integer
    fact = 1
    For i = 1 To x
        fact = fact * i
    Next i
End Sub
```

运行结果如图 3-48 所示。

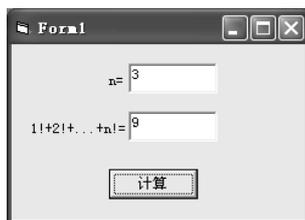


图 3-48 程序的执行结果

2. 编写一个二分法查找函数，在输入的 10 个数构成的数组运用二分法查找再次输入的数，找到则返回该数在数组中的下标位置，找不到则返回-1。

设计思路：将数列有序（递增或递减）排列，查找过程中采用跳跃式方式查找，即先以有序数列的中点位置为比较对象，如果要找的元素值小于该中点元素，则将待查序列缩小为左半部分，否则为右半部分。通过一次比较，将查找区间缩小一半。二分查找是一种高效的查找方法，它可以明显减少比较次数，提高查找效率。但是，二分查找的先决条件是查找表中的数据元素必须有序。

程序代码如下：

```
'添加子过程 sort(), 用于数据排序
Public Sub sort(a() As Integer)
    Dim temp As Integer, i As Integer, j As Integer
    For i = LBound(a) To UBound(a)
```

```

        For j = i + 1 To UBound(a)
            If a(j) < a(i) Then
                temp = a(i)
                a(i) = a(j)
                a(j) = temp
            End If
        Next j
    Next i
End Sub
'添加查找函数 search 函数
Public Function search(a() As Integer, key As Integer) As Integer
    Dim upper As Integer, lower As Integer
    Dim temp As Integer, intindex As Integer
    upper = UBound(a)
    lower = LBound(a)
    Do
        If a((upper + lower) \ 2) > key Then
            upper = (upper + lower) \ 2
        ElseIf a((upper + lower) \ 2) < key Then
            lower = (upper + lower) \ 2
        Else
            search = (upper + lower) \ 2
        End If
        Exit Function
    End If
    If upper = lower Then
        search = -1
        Exit Function
    End If
Loop
End Function
'在窗体的单击事件中添加代码，调用子过程和函数，对输入数据处理
Private Sub Form_Click()
    Dim i As Integer, a(9) As Integer
    Dim str1 As String, index As Integer
    For i = 0 To 9
        str1 = "请输入第" & CStr(i + 1) & "个数: "
        a(i) = CInt(InputBox(str1))
    Next i
    sort a
    index = search(a, CInt(InputBox("请输入要查找的数: ")))
    If index = -1 Then
        Print "你要查找的数不在数组中"
    Else
        Print "你要查找的数: "; a(index); "在数组中的下标为"; index
    End If
End Sub
End Sub

```

通过对话框输入 1~10 之间的数，查找 5，程序运行结果如图 3-49 所示。

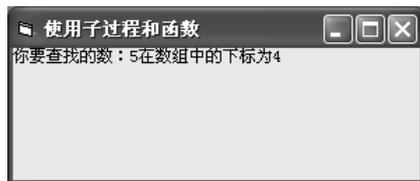


图 3-49 程序的执行结果

【实验内容】

1. 采用递归调用方式求两个自然数的最大公约数。
2. 在同一个窗体上建立两个过程，其中 `area` 过程是通用函数过程计算面积大小，单击窗体中的按钮将计算输入三边长的三角形面积。
3. 编写一个过程，实现随机产生 30 个任意三位正整数，并且从大到小顺序排列。
4. 使用顺序查找方法在数组 {1,3,5,2,4} 中查找指定的元素 `k`。

3.7 实验七 界面设计

【实验目的】

1. 掌握多窗体程序设计的方法。
2. 掌握下拉式菜单的应用。
3. 了解通用对话框。
4. 了解 ActiveX 控件。

【相关知识】

1. 多窗体

在实际应用中往往需要用到多个窗体。建立多窗体 (MultiForm) 的方法是，选择“工程”→“添加新窗体”命令或单击工具栏中的按钮，在同一个工程文件中再建立一个新的窗体 Form2，将这个窗体保存为另一个文件。按照这种方法就能够在同一个工程文件中建立多个窗体。

如果没有特别的设定，多窗体程序的第一个窗体默认为启动窗体，当应用程序开始运行时，先运行这个窗体。如果要改变系统默认的启动窗体，可以通过“工程属性”对话框进行设置。当显示一个窗体后再显示其他的窗体，就要考虑窗体的加载、显示、隐藏和卸载问题。

2. 菜单

大多数 Windows 应用程序都有菜单栏，它处于窗体标题栏的下面，并包含一个或多个菜单标题。单击每个菜单标题都会弹出一个下拉菜单，其中包含菜单项、分隔符和子菜单标题。在包含较多菜单项的菜单上，经常使用分隔符将各项划分成一些逻辑组。

使用菜单编辑器可以创建新菜单或编辑已有的菜单。打开“工具”菜单，执行“菜单编辑器”命令，出现如图 3-50 所示的“菜单编辑器”对话框。其中各主要选项的含义如下：

(1) 标题: 用来输入菜单名, 这些名字出现在菜单栏或菜单中。如果想建立分隔符, 则应输入一个连字符“-”。为了能够通过键盘访问菜单项, 可在一个字母前插入“&”符号。例如“新建(&N)”, 在运行时该字母带有下划线。

(2) 名称: 在程序中以该名称访问菜单项。

(3) 索引: 指定一个数值来确定控件在控件数组中的位置。

(4) 快捷键: 为命令选择快捷键。



图 3-50 “菜单编辑器”对话框

(5) 帮助上下文 ID: 为 ContextID 指定唯一数值, 在 HelpFile 属性指定的帮助文件中, 用该数值查找适当的帮助主题。

(6) 协调位置: 共有 4 个可选项, 它们决定如何在窗体中显示菜单。

(7) 复选: 用来指出切换选项的开关状态。

(8) 有效: 决定是否让菜单项对事件作出响应。

(9) 可见: 决定是否将菜单项显示在菜单上。

(10) 显示窗口列表: 在 MDI 应用程序中, 确定菜单控件是否包含一个打开的 MDI 子窗体列表。

(11) 右箭头: 把选定的菜单项向右移一个等级。

(12) 左箭头: 把选定的菜单项向左移一个等级。

(13) 上箭头: 把选定的菜单项在同级菜单内向上移动一个位置。

(14) 下箭头: 把选定的菜单项在同级菜单内向下移动一个位置。

(15) 下一个: 将选定项移动到下一行。

(16) 插入: 在当前选定项上方插入一行。

(17) 删除: 删除当前选定项。

菜单设计过程可以分两步, 第一步是用菜单编辑器建立菜单, 第二步是为菜单编写代码。首先在菜单编辑器的“标题”文本框中输入菜单项的名称, 然后在“名称”文本框中输入该菜单控件的名称, 此时菜单标题自动显示在菜单控件列表框中。单击“下一个”按钮, 可以再建一个菜单控件; 单击“插入”按钮, 可以在现有的控件之间增加一个菜单控件; 在菜单控件列表框中选一个菜单控件, 单击左箭头或右箭头按钮, 可以改变该控件在列表框中的缩进位置, 决定该控件是菜单标题、菜单项、子菜单标题, 还是子菜单项。位于列表框中左

侧平齐的菜单控件作为菜单标题显示在菜单栏中，列表框中被缩进过的菜单控件成为其前导菜单控件的菜单项。如图 3-51 (a) 所示为设计过程，图 3-51 (b) 为设计后的结果。

每个菜单项实际都是一个对象，标题和名称就是它们的属性。其他属性还有 Enabled、Visible、Checked、Index 等。除了分隔符以外的所有菜单项都能识别 Click 事件。

3. 通用对话框

通用对话框 (CommonDialog) 控件是一个 ActiveX 控件，在使用前应该添加到工具箱中。具体方法是，选择“工程”→“部件”命令，弹出“部件”对话框。在“控件”选项卡中选择 Microsoft Common Dialog Control 6.0 复选框，再单击“确定”按钮关闭“部件”对话框，通用对话框控件就被添加到当前工程的工具箱中，默认名为 CommonDialogX (X 为 1, 2, …)。

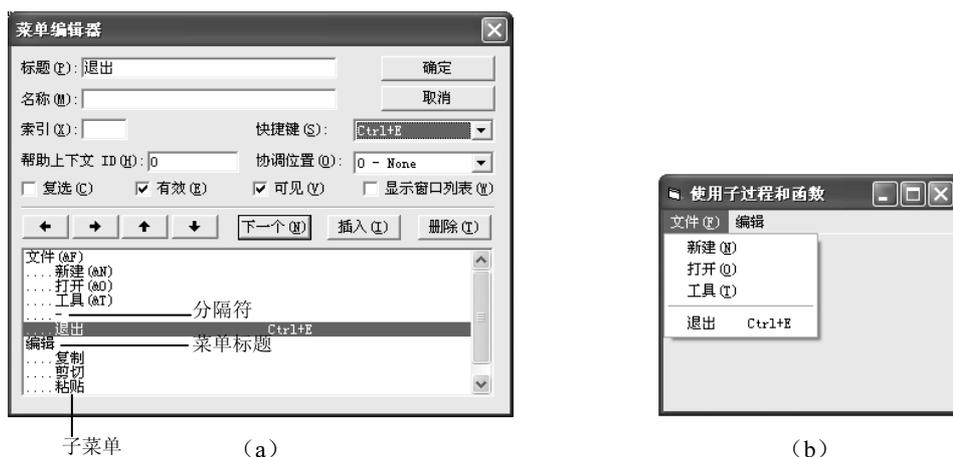


图 3-51 添加的各菜单元素

通用对话框提供 5 种形式的对话框，通过设置 Action 属性值或调用 Show 方法可建立不同类型的对话框。表 3-11 即为各类对话框的 Action 属性值和方法。

表 3-11 Action 属性取值和对应的对话框列表

通用对话框的类型	Action	方法	功能
打开 (Open)	1	ShowOpen	显示“文件打开”对话框
另存为 (Save As)	2	ShowSave	显示“文件保存”对话框
颜色 (Color)	3	ShowColor	显示“颜色”对话框
字体 (Font)	4	ShowFont	显示“字体”对话框
打印机 (Printer)	5	ShowPrinter	显示“打印”对话框
帮助 (Help)	6	ShowHelp	显示“帮助”对话框

每种对话框都有自己特殊的属性，这些属性既可以在属性窗口中设置，也可以在代码中设置，还可以在“属性页”对话框中设置。在窗体上右击通用对话框控件，可以弹出“属性页”对话框，如图 3-52 所示。



图 3-52 通用对话框的“属性页”对话框

通用对话框的基本属性如下：

(1) **Action**: 决定打开何种类型的对话框，该属性只能在使用时赋值，不能在设计时赋值。

(2) **DialogTitle**: 对话框标题，用来设置对话框的标题内容，默认值为“打开”。

(3) **CancelError**: 表示用户在与对话框进行信息交换时，单击“取消”按钮是否产生出错信息。

通用对话框控件有一系列的 **Show** 方法，例如 **ShowOpen**、**ShowSave**、**ShowFont**、**ShowColor**、**ShowPrinter**、**ShowHelp** 等。我们只需在程序中写入 **CommonDialogX.ShowOpen** 语句，就能弹出“打开”对话框。通用对话框控件的属性是和对话框类型紧密相关的，有些属性只适用于某一类对话框，有些属性在不同的对话框中是有差别的。

4. ActiveX 控件

ActiveX 控件是一段可以重复使用的编程代码和数据，在 **Windows** 系统里以文件的形式存在，一般放在 **System** 目录下，对应的文件扩展名为 **.ocx**。**Visual Basic** 标准工具箱里没有 **ActiveX** 控件，需要在“部件”对话框中选中相关控件，把它们添加到工具箱中，然后像标准控件一样使用。常用的 **ActiveX** 控件如表 3-12 所示。

表 3-12 常用 ActiveX 控件

ActiveX 控件	ActiveX 部件	文件名
TabStrip 页框	Windows 通用控件 Microsoft Windows Common Controls 6.0	路径： Windows\system\Mscomctl.ocx
ToolBar 工具栏		
StatusBar 状态栏		
ProgressBar 进度条		
TreeView 分层显示		
ListView 排列显示		
ImageList 图像列表		
Slider 滑块		
ImageCombo 图像组合框		

续表

ActiveX 控件	ActiveX 部件	文件名
CommonDialog 通用对话框	Microsoft Common Dialog Control 6.0	Comdlg32.ocx
MMControl 多媒体	Microsoft Multimedia Control 6.0	Mci32.ocx
MediaPlayer 媒体播放器	Microsoft Media Player	Msdxm.ocx

【实验例题】

1. 设计一个密码登录程序，要求有 3 个窗体。在第 1 个窗体输入用户名和密码，如果密码正确，弹出第二个窗体，显示“欢迎使用本系统”。窗体 2 有 1 个游戏菜单，可以玩游戏；如果密码不正确，弹出第 3 个窗体，显示“对不起，您不能使用该系统！”。

设计步骤：新建一个工程，在 Form1 中放置 3 个标签控件、2 个文本框控件和 2 个命令按钮控件，如图 3-53 所示。建立新窗体 Form2，如图 3-54 所示。再创建窗体 Form3，界面如图 3-55 所示。各窗体控件属性设置如表 3-13 所示。



图 3-53 窗体 1 界面

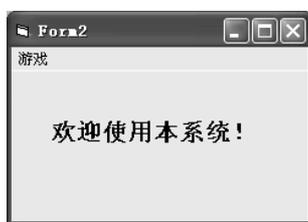


图 3-54 窗体 2 界面



图 3-55 窗体 3 界面

表 3-13 各窗体控件属性值

对象	控件名称	属性名称	属性值
窗体 1	Form1	Caption	Form1
标签 1	Label1	Caption	登录系统
标签 2	Label2	Caption	请输入用户名:
标签 3	Label3	Caption	请输入密码:
文本框 1	Text1	Text1	"" (空)
		PasswordChar	*
文本框 2	Text2	Text2	"" (空)
		PasswordChar	*
命令按钮 1	Command1	Caption	确定
命令按钮 2	Command2	Caption	退出
窗体 2	Form2	Caption	Form2
标签 1	Label1	Caption	欢迎使用本系统!
游戏	Youxi	Caption	游戏
		Enabled	True

续表

对象	控件名称	属性名称	属性值
空档接龙	Jielong	Caption	空档接龙
		Enabled	True
扫雷	Saolei	Caption	扫雷
		Enabled	True
结束	Jieshu	Caption	结束
		Enabled	True
窗体 3	Form3	Caption	Form3
命令按钮 1	Command1	Caption	退出

编写程序代码如下：

(1) 窗体 1 代码。

```
Private Sub Command1_Click()
    If Text1.Text = "user" And Text2.Text = "123456" Then
        Form2.Show
    Else
        Form3.Show
    End If
    Form1.Hide
End Sub
Private Sub Command2_Click()
    End
End Sub
```

(2) 窗体 2 代码。

```
Private Sub Jielong_Click()
    Shell ("c:\windows\system32\freecell.exe"), vbNormalFocus
End Sub

Private Sub Jieshu_Click()
    End
End Sub
```

```
Private Sub Saolei_Click()
    Shell ("c:\windows\system32\winmine.exe"), vbNormalFocus
End Sub
```

(3) 窗体 3 代码。

```
Private Sub Command1_Click()
    End
End Sub
```

程序运行结果如图 3-56 所示。



图 3-56 程序的执行结果

2. 设计应用程序。要求单击“打开”按钮时弹出“打开”对话框，选择图像文件，并将其显示在图像框控件中；单击“字体”按钮时弹出“字体”对话框，选择需要的字体、字形、字号后，在文本框控件中显示文字；单击“颜色”按钮时弹出“颜色”对话框，可选择需要的字体颜色。

设计步骤：

新建一个工程，在 Form1 中放置 4 个命令按钮控件、1 个文本框控件、1 个图像框控件和 1 个通用对话框控件。各控件属性如表 3-14 所示。

表 3-14 控件属性值

对象	控件名称	属性名称	属性值
窗体	Form1	Caption	Form1
文本框	Text1	Text1	字体和颜色对话框
		MultiLine	True
图像框	Image1	Stretch	True
命令按钮	CmdFont	Caption	字体
	CmdColor	Caption	颜色
	CmdExit	Caption	退出
	CmdOpen	Caption	打开

编写程序代码：

```
Private Sub CmdColor_Click()
    On Error Resume Next
    CommonDialog1.CancelError = True
    CommonDialog1.Flags = cdICCRGBInit
    CommonDialog1.ShowColor
    If Err = cdICancel Then Exit Sub
    Text1.ForeColor = CommonDialog1.Color
End Sub

Private Sub CmdExit_Click()
    End
End Sub

Private Sub CmdFont_Click()
    On Error Resume Next
```

```

'当用户单击“取消”按钮时，返回一个错误信息
CommonDialog1.CancelError = True
CommonDialog1.Flags = cdlCFBoth Or cdlCFEffects
CommonDialog1.ShowFont
If Err = cdlCancel Then
    Exit Sub
Else
    Text1.FontName = CommonDialog1.FontName
    Text1.FontBold = CommonDialog1.FontBold
    Text1.FontItalic = CommonDialog1.FontItalic
    Text1.FontSize = CommonDialog1.FontSize
    Text1.FontStrikethru = CommonDialog1.FontStrikethru
    Text1.FontUnderline = CommonDialog1.FontUnderline
    Text1.ForeColor = CommonDialog1.Color
End If
End Sub
Private Sub CmdOpen_Click()
'出现错误时跳到下一语句
On Error Resume Next
CommonDialog1.CancelError = True
'属性 DialogTitle 是要弹出的对话框的标题
CommonDialog1.DialogTitle = "打开文件"
'默认的文件名为空
CommonDialog1.FileName = ""
CommonDialog1.InitDir = _
"C:\Documents and Settings\Administrator\My Documents"
CommonDialog1.Filter = "Bmp File(*.bmp)*.Bmp|Jpg File(*.jpg)*.jpg"
CommonDialog1.FilterIndex = 2
CommonDialog1.Flags = cdlOFNCreatePrompt + cdlOFNHideReadOnly
CommonDialog1.ShowOpen
If Err = cdlCancel Then Exit Sub
UserSelectFile = CommonDialog1.FileName
Image1.Picture = LoadPicture(UserSelectFile)
End Sub

```

程序运行结果如图 3-57 所示。



图 3-57 程序的执行结果

3. 用 ImageList 控件设置几张图片, 程序启动时自动读取每一张图片的 Index 属性, 填入窗体的 ListBox 中。从 ListBox 中选择任何一张图片, 就会显示在窗体上。

设计步骤:

新建一个工程, 在“部件”对话框中选中 Microsoft Windows Common Controls 6.0 选项, 单击“确定”按钮。此时工具箱中出现了一组 ActiveX 控件, 其中就有 ImageList 控件。在 Form1 中放置 1 个图片框控件、1 个列表框控件和 1 个 ImageList 控件, 如图 3-58 所示。

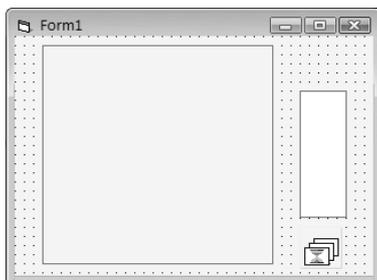


图 3-58 例 3 窗体界面

打开 ImageList 控件的“属性页”对话框, 选择“图像”选项卡, 单击“插入图片”按钮, 弹出“选定图片”对话框。找到要显示图片的路径和名称, 选中要显示的图片, 单击“打开”按钮, 将图片插入到 ImageList 中。用同样方法可以插入多张图片, 每插入一张图片, 索引值加 1。

```
Private Sub Form_Load()
    Dim i As Integer
    For i = 1 To ImageList1.ListImages.Count '把所有图像的“索引”加入到 List1
        List1.AddItem ImageList1.ListImages(i).Index
    Next
End Sub

Private Sub List1_Click()
    Picture1.PaintPicture _
        ImageList1.ListImages(Val(List1.Text)).Picture, 0, 0, 3000, 3000
End Sub
```

程序运行结果如图 3-59 所示。从 ListBox 中选择任何一张图片的索引值, 该图片就会显示在窗体上。



图 3-59 程序的执行结果

【实验内容】

1. 利用通用对话框设计一个简单的文本编辑器，具有打开、保存、字体、颜色功能。要求用菜单实现，有“文件”和“格式”两个菜单，“文件”菜单下包含“打开”、“保存”和“退出”三项。“格式”菜单下包含“字体”和“颜色”两项，打开文件显示在文本框中，各控件的布局如图 3-60 所示。



图 3-60 “文本编辑”程序界面布局

2. 用进度条模拟读文件和写文件的进度状态。单击“开始”按钮，进度条开始移动。利用时钟控件，每隔 100ms 将变量 i 的值加 1 并赋给进度条，直到 i 的值为 100，此时完成进度终止。界面如图 3-61 所示。

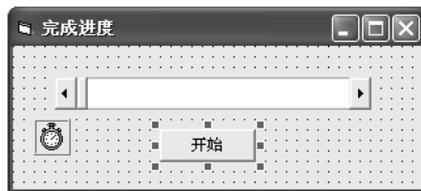


图 3-61 “完成进度”程序界面布局

3. 创建一个多文档界面的程序，把第 1 题中设计的简单文本编辑器窗体作为子窗体，并创建一个新的窗体，设计程序，使得某小球沿着圆周顺时针运动，程序界面如图 3-62 所示。

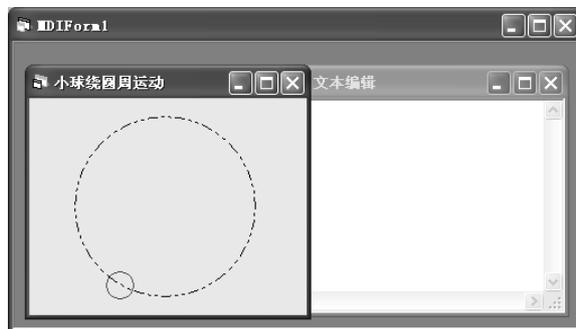


图 3-62 “小球运动”程序界面布局

3.8 实验八 文件

【实验目的】

1. 理解文件的概念，了解文件的结构与分类。
2. 掌握顺序文件的读写方法。
3. 掌握随机文件的读写方法。
4. 掌握文件系统控件的功能和使用方法。

【相关知识】

1. 文件

文件是存储在外部介质上的一段数据流。

2. 文件的分类

(1) 根据数据性质的不同，文件可以分为程序文件和数据文件。

程序文件是存储计算机程序的文件，扩展名为.exe、.frm、.vbp、.vbg 或.bas。

数据文件是存储数据信息的文件。根据数据存取方式和结构的不同，可以分为顺序文件和随机文件，文件结构由若干记录组成，记录由字段组成，字段由字符组成。本章将数据文件统一简称为文件。

(2) 根据数据的存取方式和结构的不同，文件可以分为顺序文件和随机文件。

顺序文件结构简单、占用空间少，文件中的记录一个接一个地存储。在这种文件中只知道第一个记录的存储位置，如果要查找某个记录，只能从文件头开始逐个记录地顺序读取，直到找到该记录为止。顺序文件的这种存储方式使得其不能灵活地存取和增减数据，因此适用于存储有一定规律且不需要经常修改的数据。

随机文件是随机存取文件的简称，文件中的每个记录长度是固定的，且每个记录都有一个记录号。在这种文件中查找某个记录，只需要给出其记录号即可快速找到该记录。在随机文件中可以同时进行读、写操作，能快速灵活地查找和修改每个记录。

3. 文件的操作流程

(1) 打开（创建）文件是为文件在内存中指定一个读写缓冲区，指定文件的打开方式，确定“文件号”，便于后面的操作。

(2) 读文件是将外部介质上的文件数据输入到内存；写文件是将内存中的文件数据输出到外部介质上。

(3) 关闭文件是将内存缓冲区中的所有数据写入文件中，并释放与该文件相关的“文件号”。

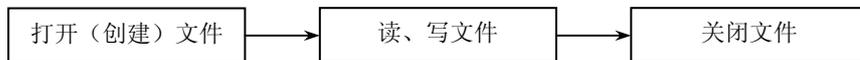


图 3-63 文件操作流程图

4. 文件的打开与关闭

(1) Open 语句。

Open 语句格式如下：

Open 文件名 For [打开方式] [Access 存取类型] As [#] 文件号 [Len=记录长度]

功能：为文件的输入输出分配缓冲区，并确定缓冲区所使用的存取方式。

说明：

- 1) 格式中的 Open、For、Access、As、Len 为关键字。
- 2) 打开方式：指定文件的输入输出方式，默认为 Random。
 - Input：指定顺序输入方式。
 - Output：指定顺序输出方式。
 - Append：指定以追加的方式顺序输出。
 - Random：指定随机存取方式。
- 3) 存取类型：指定访问文件的类型。
 - Read：打开只读文件。
 - Write：打开只写文件。
 - Read Write：打开读/写文件。
- 4) 文件号：是一个整数表达式，取值范围为 1~511。
- 5) 记录长度：是一个整数表达式，取值不超过 32767 字节。

(2) Close 语句。

Close 语句格式如下：

Close [#]文件号 [, [#]文件号]...

功能：结束文件的输入输出操作，释放相应文件号。

说明：

- 1) 格式中的 Close 为关键字。
- 2) 文件号是 Open 语句中使用的文件号。如果指定文件号，则仅关闭指定文件；如果不指定文件号，则关闭所有打开的文件。

5. 顺序文件

(1) 写操作。

Visual Basic 中顺序文件的写操作可使用 Print 语句与 Write 语句。

1) Print 语句格式：**Print #文件号, [Spc(n) | Tab(n)] [表达式表]**

功能：把数据写入文件中。

说明：

- ①格式中的 Print 是关键字。
- ②文件号是使用 Open 语句打开文件时指定的文件号。
- ③Spc 函数的功能是插入空格；Tab 函数的功能是对输出进行定位。
- ④表达式表省略的情况下，将向文件写入一个空行。
- ⑤各数据项之间用逗号隔开。

2) Write 语句。

格式：**Write #文件号, [表达式表]**

功能：把数据写入文件中。

说明：

- ①格式中的 Write 是关键字。

②文件号、表达式说明同 Print 语句。

③Write 语句只能写以 Output 或 Append 方式打开的文件。

④用 Write 语句写文件时，数据以紧凑的格式存放，且在数据项之间自动插入逗号，字符串加上双引号，正数前面没有空格。

(2) 读操作。

Input 语句格式：Input #文件号,变量表

功能：从文件中读出数据。

说明：

①格式中的 Input 是关键字。

②文件号是使用 Open 语句打开文件时指定的文件号。

③变量表由一个或多个变量组成，变量之间使用逗号隔开。

④因为读出的数据将赋值给变量表中的相应变量，因此变量类型与数据类型应一一匹配。

6. 随机文件

(1) 读操作。

1) 随机文件读操作步骤如下：

①定义数据类型。

②打开随机文件。

③用 Get 语句将文件中的数据读到内存中。

④关闭文件。

2) Get 语句。

格式：Get #文件号, [记录号], 变量

功能：实现随机文件的写操作。

说明：

①格式中的 Get 是关键字。

②文件号是使用 Open 语句打开文件时指定的文件号。

③记录号的取值范围为 1~231-1，以 Random 方式打开的文件，记录号是需要读的记录的编号。如果省略记录号，其后的逗号需保留，执行读下一记录的操作。

④变量是除对象变量和数组变量外的任何变量（包括含有单个数组元素的下标变量）。

(2) 写操作。

1) 随机文件写操作步骤如下：

①定义数据类型。

②打开随机文件。

③用 Put 语句将内存中的数据写入文件。

④关闭文件。

2) Put 语句。

格式：Put #文件号, [记录号], 变量

功能：实现随机文件的写操作。

说明：①格式中的 Put 是关键字。

②其他说明与 Get 语句相同。

7. 文件操作语句与函数

(1) 文件操作语句。

1) 复制文件。

格式: **FileCopy 源文件名,目标文件名**

2) 重命名文件。

格式: **Name 原文件名 As 新文件名**

3) 删除文件。

格式: **Kill 文件名**

4) 设置文件属性。

格式: **SetAttr 文件名,属性**

说明: 文件属性包括 vbNormal、vbReadOnly、vbHidden、vbSystem、vbArchive。

(2) 文件操作函数。

1) FreeFile 函数。

格式: **FreeFile**

功能: 返回一个没有使用的文件号。

2) FileDateTime 函数。

格式: **FileDateTime(文件名)**

功能: 返回创建或最后一次修改文件的日期和时间。

3) Loc 函数。

格式: **Loc(文件号)**

功能: 返回由文件号指定的文件当前读写位置。

4) LOF 函数。

格式: **LOF(文件号)**

功能: 返回文件长度。

5) EOF 语句。

格式: **EOF(文件号)**

功能: 判断文件结束状态。

说明: 读写文件时, 当指针到文件尾时, EOF 返回值为 True, 否则返回 False。

8. 文件系统控件

(1) 驱动器列表框 (DriveListBox)。

驱动器列表框控件包括很多常见属性, 其中需要特别注意的是 Drive 属性, 该属性只能通过程序代码设置。

Drive 属性设置格式: **驱动器列表框名称.Drive[=驱动器名]**

功能: 设置或返回所选择的驱动器名。

当重新设置驱动器列表框的 Drive 属性时, 会引发 Change 事件, Change 事件过程开头一般为 Drive1_Change ()。

(2) 目录列表框 (DirListBox)。

在目录列表框中只能显示当前驱动器上的目录, 若要显示其他驱动器上的目录, 则使用 Path 属性改变路径。

Path 属性设置格式：**目录列表框名称.Path=["路径"]**

功能：设置或返回当前路径。

当重新设置目录列表框 Path 属性时，同样会引发 Change 事件。

(3) 文件列表框 (FileListBox)。

文件列表框控件用于显示当前目录下的文件。

1) 文件列表框常用属性。

①Path 属性：设置或返回文件列表框当前目录。

②Filename 属性：返回或设置被选定文件的文件名，不包括路径名。

③Pattern 属性：返回或设置文件列表框所显示的文件类型。

④Archive/Normal/Hidden/System/ReadOnly 属性：设置是否只显示文档/标准/隐藏/系统/只读文件。

⑤List 属性：设置或返回列表框中的某一项目。

⑥ListCount 属性：返回列表框中所列项目的总数。

⑦ListIndex 属性：设置或返回当前列表框上所选的项目的索引值。第一项的索引值为 0，依此类推。

2) 文件列表框常用事件。

①PathChange 事件：Filename 属性指定的文件的 Path 属性改变时触发的事件。

②PatternChange 事件：Filename 属性指定的文件的 Pattern 属性改变时触发的事件。

③Click、DbClick 事件：单击、双击文件名时触发的事件。

【实验例题】

1. 创建一个学生成绩表，录入记录数据，并按分数高低将记录显示在窗体中。

设计思路：顺序文件的读操作使用 Input 语句、写操作使用 Open 语句、数值数据的排序使用冒泡算法。

设计步骤：

(1) 界面设计如图 3-64 所示。



图 3-64 学生成绩表界面设计

(2) 属性设置如表 3-15 所示。

表 3-15 属性设置

对象	控件名称	属性名称	属性值
命令按钮	Command1	Caption	输入成绩
	Command2	Caption	按成绩高低排序

(3) 编写程序代码:

'添加一个标准模块, 在标准模块中声明记录类型, 保存为 sscore.bas

```
Type stu
    snum As String * 5
    sname As String * 8
    sscore As Integer
End Type
'写文件
Private Sub Command1_Click()
    Static stud() As stu
    Open "d:\score.txt" For Output As #1
    n = InputBox("请输入学生总数: ", "数据输入")
    ReDim stud(n) As stu
    For i = 1 To n
        stud(i).snum = InputBox$("请输入学号: ", "数据输入")
        stud(i).sname = InputBox$("请输入姓名: ", "数据输入")
        stud(i).sscore = InputBox("请输入成绩: ", "数据输入")
        Write #1, stud(i).snum, stud(i).sname, stud(i).sscore
    Next i
    Close #1
End
End Sub
'读文件并使用冒泡排序
Private Sub Command2_Click()
    Static stud() As stu
    Static temp() As stu
    Open "d:\score.txt" For Input As #1
    n = InputBox("请输入学生总数: ", "数据输入")
    ReDim stud(n) As stu
    ReDim temp(0) As stu
    For i = 1 To n
        Input #1, stud(i).snum, stud(i).sname, stud(i).sscore
    Next i
    For i = n To 2 Step -1
        For j = 1 To i - 1
            If stud(j).sscore < stud(j + 1).sscore Then
                temp(0).sscore = stud(j + 1).sscore
                temp(0).sname = stud(j + 1).sname
                temp(0).snum = stud(j + 1).snum
                stud(j + 1) = stud(j)
                stud(j) = temp(0)
            End If
        Next j
    Next i
    Close #1
    For i = 1 To n
```

```
Print stud(i).snum, stud(i).sname, stud(i).sscore
Next i
End Sub
```

程序运行结果如图 3-65、图 3-66 所示。



图 3-65 score.txt 文件内容



图 3-66 程序执行结果

2. 创建一个支出统计表，录入数据，总计后显示在窗体中。

(1) 界面设计如图 3-67 所示。



图 3-67 “支出统计”程序界面设计

(2) 属性设置如表 3-16 所示。

表 3-16 属性设置

对象	控件名称	属性名称	属性值
窗体	Form1	Caption	支出统计
标签	Label1	Caption	支出名目:
	Label2	Caption	金额 元
	Label3	Caption	总计 元

续表

对象	控件名称	属性名称	属性值
文本框	Text1	Text	"" (空)
		Name	Titem
	Text2	Text	"" (空)
		Name	Tmoney
	Text3	Text	"" (空)
		Name	Tshow
	Text4	Text	"" (空)
		Name	Tsum
命令按钮	Command1	Caption	添加
		Name	Cadd
	Command2	Caption	显示
		Name	Cshow
	Command3	Caption	总计
		Name	Csum
图形	Shape1	BorderWidth	2

(3) 编写程序代码如下:

'添加一个标准模块, 在标准模块中声明记录类型

```
Type Payout
```

```
    item As String * 20
```

```
    money As Double
```

```
    sum As Double
```

```
End Type
```

'在窗体层中定义通用变量

```
Dim mypay As Payout
```

```
Dim recordnum As Integer
```

```
Dim sum As Double
```

'写随机文件

```
Private Sub Cadd_Click()
```

```
    If Titem <> "" And Tmoney <> "" Then
```

```
        mypay.item = Titem.Text
```

```
        mypay.money = Tmoney.Text
```

```
        Open App.Path & "\mypayout.txt" For Random As #1 Len = Len(mypay)
```

```
        recordnum = LOF(1) / Len(mypay) + 1
```

```
        Put #1, recordnum, mypay
```

```
        Close #1
```

```
        Titem.Text = ""
```

```
        Tmoney.Text = ""
```

```
    End If
```

```
End Sub
```

'读随机文件并计算显示结果

```
Private Sub Csum_Click()
    Open App.Path & "\mypayout.txt" For Random As #1 Len = Len(mypay)
    Print
    Print
    Print
    Print
    Print Spc(1), "支出名目", Spc(12), "金额"
    Do While Not EOF(1)
        Get #1, , mypay
        Print Spc(1), mypay.item, mypay.money
        sum = sum + mypay.money
    Loop
    Tsum.Text = sum
    Close #1
End Sub
```

程序运行结果如图 3-68 所示。



图 3-68 程序的执行结果

3. 创建一个数据文件浏览器。

设计思路：使用驱动器列表框、目录列表框和文件列表框控件创建文件管理器；使用 Line Input 语句读文件。

设计步骤：

(1) 界面设计，如图 3-69 所示。



图 3-69 数据文件浏览器界面设计

(2) 属性设置如表 3-17 所示。

表 3-17 属性设置

对象	控件名称	属性名称	属性值
窗体	Form1	Caption	数据文件浏览器
驱动器列表框	Drive1	Name	Drive1
目录列表框	Dir1	Name	Dir1
文件列表框	File1	Name	File1
命令按钮	Command1	Name	Cmdopen
		Caption	打开数据文件
	Command2	Name	Cmdexit
		Caption	退出程序
文本框	Text1	Name	Txtopen
		Text	"" (空)
		Multiline	True

(3) 编写程序代码如下：

```
'在窗体层中定义通用变量
Dim FName As String
Dim FPath As String
Dim Temp As String
'选择数据文件
Private Sub Dir1_Change()
    File1.Path = Dir1.Path
End Sub
Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
End Sub
Private Sub File1_Click()
    FPath = File1.Path
    FName = File1.FileName
    FName = FPath + FName
End Sub
'读取并显示数据文件
Private Sub Cmdopen_Click()
    Open FName For Input As #1
    Do While Not EOF(1)
        Line Input #1, Temp
        Txtopen.Text = Temp
    Loop
    Close #1
```

```

End Sub
'退出程序
Private Sub Cmdexit_Click()
    End
End Sub

```

程序运行结果如图 3-70 所示。



图 3-70 程序运行结果

【实验内容】

1. 用文件系统控件创建图片浏览器，界面设计如图 3-71 所示。
2. 编写程序，按下列格式输出本月月历，并把结果放入一个文件中，界面设计如图 3-72 所示。

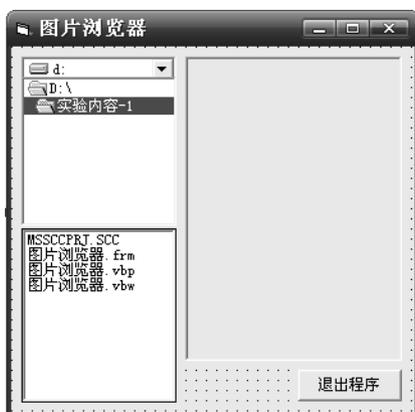


图 3-71 图片浏览器界面设计



图 3-72 月历界面设计

3. 在硬盘上建立一个通讯录，文件中的记录包括编号、用户名、电话号码和地址等 4 项内容。编写一个程序，从文件中查找指定的用户的编号，并在文本框中输出其名字、电话号码和地址，界面设计如图 3-73 所示。



图 3-73 通讯录界面设计

3.9 实验九 数据库应用

【实验目的】

1. 掌握可视化数据管理器（VisData）的使用方法。
2. 了解 VB 数据控件和 ADO 控件的常用属性和方法。
3. 掌握访问关系数据库的方法。

【相关知识】

1. 数据库

数据库（DataBase，DB）是按照数据结构来组织和存储在一起的具有较小冗余度、可共享的相关数据的集合。数据库中数据的存储独立于使用它的程序；对数据库插入新数据，修改和检索原有数据均能按一种公用的和可控制的方式进行。

数据库管理系统（DataBase Management System，DBMS）是一种操纵和管理数据库的大型软件，用于建立、使用和维护数据库。

2. 关系数据库

关系数据库是建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。关系数据库根据表、记录和字段之间的关系进行组织和访问，以行和列的二维表形式存储数据，并通过关系将这些表联系在一起。

（1）表。表是以行和列的形式组织起来的数据的集合。一个数据库包括一个或多个表，一般一个关系对应一个表。

（2）记录。表中的每一行称为一个记录，记录由若干个字段组成，每个记录行用记录号作为标识。

（3）字段。表中的每一列称为一个字段，表是由其包含的各种字段定义的，每个字段描述了它所具有的数据的意义。创建表时，为每个字段分配一个数据类型、定义数据长度和其他属性，因此表的设计就是对表的字段的设计。

（4）关键字。关键字用来确保表中记录的唯一性，通常一个字段或者多个字段可作为一个表的索引字段。对可以唯一标识一个记录的关键字称为主关键字或者简称主键。

(5) 索引。索引是表中单列或多列数据的排序列表，每个索引指向其对应的数据表的某一行。为提高数据的访问效率，可以为表建立多个索引，从而改变表中记录的排列顺序。

(6) 表间关系。一个数据库包含一个以上的表，一般一个关系对应一个表，表间关系把各个表链接起来。表间关系是通过各表中的某一个相同的关键字建立起来的。

(7) 视图。视图是在表的基础上建立的一个虚拟表，它的各项操作与真实的表是相同的，目的在于简化数据的表达。

3. 可视化数据管理器 (VisData)

在 Visual Basic 中借助可视化数据管理器可以创建 Access 格式的数据库。具体操作步骤如下：

(1) 打开 VisData。在 Visual Basic 运行环境中，选择“外接程序”→“可视化数据管理器”命令，打开“可视化数据管理器”窗口。

(2) 创建 Access 格式的数据库。在“可视化数据管理器”窗口中，选择“文件”→“新建”→Microsoft Access→Version 7.0 命令，打开“新建数据库”窗口，创建 Access 格式的数据库。

(3) 创建表结构。在“数据库”窗口中，右击 Properties，在弹出的快捷菜单中选择“新建表”命令，打开一个“表结构”对话框。在“表结构”对话框中，单击“添加字段”按钮，打开“添加字段”对话框，添加字段，单击“生成表”按钮生成表。

(4) 添加表数据。双击“数据库”窗口中创建的表，打开“表”编辑窗口。单击表编辑窗口中的“添加”按钮，开始添加数据，每条记录添加完成后单击对话框中的“更新”按钮确定。所有数据输入结束后即完成表的创建操作。

4. VB 访问数据库的方法

(1) 标准数据控件。通过设置 Data 控件的属性可以访问数据库。

表 3-18 Data 控件部分属性、方法和事件

属性/方法/事件名称	说明
Connect 属性	用于定义控件所要链接的数据库类型，属性以分号结束
DatabaseName 属性	用于返回或设置控件的数据源
RecordSource 属性	用来返回或设置控件打开的数据库表或满足某个 SQL 查询语句的表的记录集
AddNew 方法	用于在数据表中添加一个新记录，例如 Data1.Recordset.AddNew
Update 方法	用于新记录输入完成后更新数据库，例如 Data1.Recordset.Update
UpdateRecord 方法	用于确认对记录的修改，将绑定控件中的数据强制写入数据库中
Refresh 方法	用于打开或刷新记录集，例如 Data1.Refresh
Close 方法	用于关闭数据库，例如 Data1.Recordset.Close
Delete 方法	用于删除当前记录，例如 Data1.Recordset.Delete
Error 事件	当用户单击 Data 控件发生运行错误时，激活该事件
Reposition 事件	当用户使用 Data 控件按钮或 Move 方法/Find 方法进行记录间的移动时，激活该事件
Validate 事件	如果移动数据控件的记录指针，并且绑定控件中的内容已被修改，此时数据库当前记录的内容将被更新，同时触发该事件

表 3-19 数据绑定控件的部分属性、方法和事件

属性名称	说明
DataSource 属性	用于指定一个有效的数据控件，将绑定控件连接到数据库
DataField 属性	用于设置数据库中的字段，使绑定控件与字段建立联系

表 3-20 记录集的部分属性和方法

属性/方法名称	说明
AbsolutePosition 属性	用于返回当前指针值，第 1 条记录指针值为 0，该属性为只读属性
BOF 属性	值为 True 时，记录指针处在记录集的首记录
EOF 属性	值为 True 时，记录指针处在记录集的尾记录
RecordCount 属性	用于返回记录集中的记录数，该属性为只读属性
Move 方法	MoveFirst: 移到第 1 条记录; MoveLast: 移到最后一条记录; MoveNext: 移到下一条记录; MovePrevious: 移到上一条记录
Find 方法	FindFirst: 从记录集首记录开始查找满足条件的第 1 条记录; FindLast: 从记录集尾记录开始查找满足条件的第 1 条记录; FindNext: 从当前记录开始查找满足条件的下一条记录; FindPrevious: 从当前记录开始查找满足条件的上一条记录

(2) 数据库访问对象。Visual Basic 6.0 一共有 3 种数据库访问对象：ADO (ActiveX Data Object)、RDO (Remote Data Objects) 和 DAO (Data Access Objects)。

ADO 是数据库应用程序开发的数据访问接口。ADO 通过它的属性和方法为数据的访问提供统一的接口，经过简单编程实现与各种类型的数据库进行连接。ADO 对象的作用如表 3-21 所示。

表 3-21 ADO 对象说明

对象名称	说明
Connection	连接对象，应用程序通过连接访问数据源
Command	命令对象，从连接到的数据源获取所需数据的命令信息
Recordset	记录集对象，获得一组记录组成的记录集
Error	错误对象，访问数据时从数据源返回的错误信息
Property	属性对象，ADO 控件属性信息
Parameter	参数对象，与命令对象有关的参数
Field	字段对象，包含在记录集中某个字段的信息

ADO 的 4 个集合为错误集 Errors、字段集 Fields、属性集 Properties 和参数集 Parameters。

ADO 的核心是 Connection、Recordset 和 Command 对象。具体应用时先将 Connection 对象与数据库建立连接，再用 Command 对象执行命令，用 Recordset 对象操作和查看操作结果，最后断开连接。

ADO 控件是作为可选项集成在 VB 开发环境中的，在使用前必须先添加。添加 ADO 的

具体操作过程是：打开“工程”菜单，选择“引用”→Microsoft ActiveX Data Objects 2.7 Library 命令，ADO 版本可以根据具体情况进行选择。

(3) 专用数据绑定控件。

Visual Basic 提供了一些专用的数据绑定控件：DBGrid 数据库表格控件、DBList 数据库列表控件、DBCombo 数据库组合控件、DataGrid 数据表格控件、DataList 数据列表控件、DataCombo 数据组合控件等。需要注意的是 DBGrid、DBList、DBCombo 这三个控件只有在 Visual Basic 6.0 企业版中才有，而在 Visual Basic 6.0 专业版中只有与 ADO 数据控件一起配合使用的 DataGrid、DataList、DataCombo 控件。

这些控件并不出现在工具箱中，通过选择“工程”→“部件”命令，在打开的“部件”对话框中选择 Microsoft ADO Data Control 6.0、Microsoft DataList Control 6.0、Microsoft DataGrid Control 6.0、Microsoft Chart Control 6.0 等项，单击“确定”按钮，即可在工具箱中出现相应的图标。

5. SQL 语言

SQL 是结构化查询语言 (Structure Query Language) 的简称，用于存取数据以及查询、更新和管理关系数据库系统。SQL 语言包含查询、定义、操纵和控制 4 个部分，被作为关系型数据库管理系统的标准语言。用户可以使用 SQL 语言在数据库中执行各种操作。

一个 SQL 语句至少包括以下 4 个元素：命令动词；被操作的字段列表；指定要操作的数据表；操作方式。有些情况下，操作方式可以省略。

SQL 语句中的常用命令动词有 9 个，分别是 CREATE、DROP、ALTER、SELECT、INSERT、UPDATE、DELETE、GRANT、REVOKE，用于完成对数据库的操作。

6. 报表制作

在数据库系统中数据报表是数据最常用的输出方式，制作数据报表包括了数据的分组、汇总等。Visual Basic 为报表的制作提供了数据环境设计器和数据报表设计器两个工具。

(1) 数据环境设计器。

数据环境设计器提供了一个交互式的设计环境，通过设置 Connection 对象和 Command 对象属性，快速完成对数据库的链接。

1) Connection 数据连接对象。在数据环境中操作数据之前，必须先建立连接对象。

2) Command 数据命令对象。命令对象定义了将从数据库连接中取回数据的详细信息。

将数据环境设计器中的对象设置完成后，将其拖放到报表上，系统会自动创建并完成数据绑定控件的设置。

(2) 数据报表设计器。

数据报表设计器是一个方便快捷的报表设计工具，根据数据环境设计器提供的数据，能创建出汇总多个数据表数据的报表。

【实验例题】

1. 创建图书信息登记与查阅程序。

设计思路：使用可视化数据管理器创建 Access 数据库，命名为“资料室图书管理.mdb”，包含“图书信息”表；使用数据环境设计器快速生成数据库应用程序，实现对数据库中数据的显示、删除、添加等操作。

设计步骤:

(1) 界面设计如图 3-74 所示。



图 3-74 界面设计

(2) 数据表结构与数据信息如表 3-22、表 3-23 所示。

表 3-22 “图书信息”表结构

字段名称	类型	大小	长度	是否允许零长度
编号	Long	4	固定长度	否
ISBN	Text	25	可变长度	否
书名	Text	50	可变长度	否
作者	Text	25	可变长度	否
出版社	Text	50	可变长度	否
出版日期	Date/Time	8	固定长度	否
关键词	Text	50	可变长度	否
内容简介	Text	200	可变长度	否

表 3-23 图书信息表

编号	ISBN	书名	作者	出版社	出版日期	关键词	内容简介
20060051	7-302-03985-2	Visual Basic 程序设计	谭浩强 薛淑斌 袁玫	清华大学出版社	2001 年 6 月	Visual Basic	...
20070001	7-111-09048	数据挖掘概念与技术	【加】 Jiawei Han Micheling Kamber	机械工业出版社	2006 年 3 月	数据挖掘	...
20070002	7-302-01307-1	人工智能原理	石纯一 黄昌宁等	清华大学出版社	2000 年 12 月	人工智能	...
20070003	7-115-12082-X	从零开始-3ds max 基础培训教程	詹翔 王海英 编著	人民邮电出版社	2004 年 3 月	动画制作	...

(3) 单击“开始”→“程序”→“Microsoft Visual Basic 6.0 中文版”命令，在“新建工程”对话框中选择“标准 EXE”项，进入 Visual Basic 运行环境。

(4) 选择“外接程序”→“可视化数据管理器”菜单项，打开“可视化数据管理器”窗口，如图 3-75 所示。



图 3-75 “可视化数据管理器”窗口

(5) 在“可视化数据管理器”窗口中，选择“文件”→“新建”→Microsoft Access→version 7.0 命令，打开“新建数据库”窗口，创建名为“资料室图书管理”的数据库，如图 3-76 所示。



图 3-76 新建数据库

(6) 在接下来打开的“数据库”窗口中右击 Properties 项，选择快捷菜单中“新建表”菜单项，打开一个“表结构”对话框，如图 3-77、图 3-78 所示。



图 3-77 “数据库”窗口



图 3-78 “表结构”对话框

(7) 在“表结构”对话框中，单击“添加字段”按钮，打开“添加字段”对话框并添加字段，如图 3-79 所示。



图 3-79 “添加字段”对话框

(8) 添加如图 3-80 所示的 8 个字段（字段信息如表 3-22 所示），然后单击“生成表”按钮生成“图书信息”表。



图 3-80 “表结构”对话框

(9) 双击“数据库”窗口中的“图书信息”表，打开“图书信息”表编辑窗口，如图 3-81 所示。

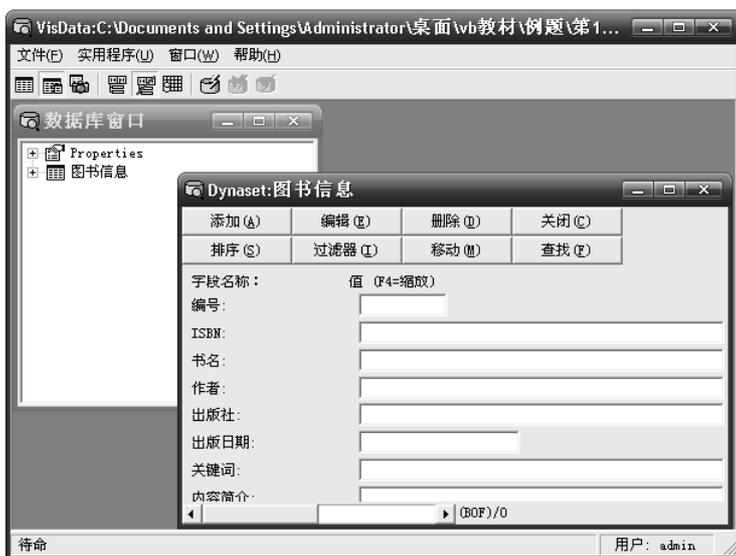


图 3-81 “图书信息”表编辑窗口

(10) 单击“图书信息”表编辑窗口中的“添加”按钮，开始添加图书信息（数据信息如表 3-23 所示），每条记录添加完成后单击对话框中的“更新”按钮确定。所有数据输入结束后即完成表的创建操作。

(11) 在“可视化数据管理器”窗口中，选择“实用程序”→“数据窗体设计器”命令，弹出“数据窗体设计器”对话框，如图 3-82 所示。



图 3-82 “数据窗体设计器”对话框

(12) 在“数据窗体设计器”对话框中，填入窗体名称为“图书信息登记与查阅”，指定记录源为“图书信息”表，即可在可用字段列表中显示“图书信息”表的字段，如图 3-83 所示。

(13) 单击  按钮将可用字段全部添加到“包括的字段”列表框中，如图 3-84 所示。

(14) 单击“生成窗体”按钮生成窗体，如图 3-85 所示。

程序运行结果如图 3-86 所示。



图 3-83 指定记录源



图 3-84 添加窗体包括的字段



图 3-85 生成的“图书信息”窗体



图 3-86 “图书信息”执行程序

2. 编写一个简单的数据库应用程序，浏览和查询“图书信息”数据库中的记录。

设计思路：使用 Adodc 控件与 DataGrid 控件链接与访问数据库；使用 SQL 的 Select 语句查询。

设计步骤：

- (1) 界面设计如图 3-87 所示。
- (2) 属性设置如表 3-24 所示。



图 3-87 “图书信息管理”程序界面设计

表 3-24 属性设置

对象	控件名称	属性名称	属性值
窗体	Form1	Caption	图书信息管理
标签控件	Label1	Caption	请输入要查询的书名
Adodc 控件	Adodc1	ConnectionString	"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path & "\资料室图书管理.mdb" & ";Persist Security Info=False"
		RecordSource	图书信息
DataGrid 控件	DataGrid1	DataSource	Adodc1
文本框	Text1	Name	Tbookname
		Text	(空)
命令按钮	Command1	Name	CmdQuery
		Caption	查询记录
	Command2	Name	CmdShowAll
		Caption	显示全部
	Command3	Name	CmdAdd
		Caption	添加记录
	Command4	Name	CmdDelete
		Caption	删除记录
	Command5	Name	CmdRefresh
		Caption	刷新记录
	Command6	Name	CmdUpdate
		Caption	更新数据
	Command7	Name	CmdExit
		Caption	关闭程序

(3) 编写程序代码如下:

```
'窗体层定义变量
Dim fvalue As String
Dim sql As String
'设置查询
Private Sub CmdQuery_Click()
    fvalue = Tbookname.Text
    sql = "select * from 图书信息 where 书名 =" + fvalue + ""
    Adodc1.CommandType = adCmdUnknown
    Adodc1.RecordSource = sql
    Adodc1.Refresh
    Set DataGrid1.DataSource = Adodc1
End Sub
'其他按钮
Private Sub CmdShowAll_Click()
    Adodc1.RecordSource = "图书信息"
    Adodc1.Refresh
    Set DataGrid1.DataSource = Adodc1
End Sub
Private Sub CmdAdd_Click()
    Adodc1.Recordset.AddNew
End Sub
Private Sub CmdDelete_Click()
    Adodc1.Recordset.Delete
End Sub
Private Sub CmdRefresh_Click()
    Adodc1.Refresh
End Sub
Private Sub CmdUpdate_Click()
    Adodc1.Recordset.UpdateBatch
End Sub
Private Sub CmdExit_Click()
    End
End Sub
```

程序运行结果如图 3-88 所示。



图 3-88 程序运行结果

3. 制作资料室图书信息报表。

设计思路：使用数据环境设计器与数据报表设计器制作报表。

设计步骤：

(1) 打开 Visual Basic 软件，创建新工程。

(2) 链接数据库。

1) 打开数据环境设计器，选择“工程”→“添加 Data Environment”命令，打开“数据环境”窗口，如图 3-89 所示。

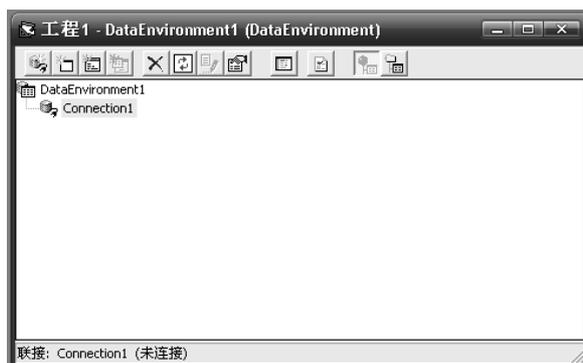


图 3-89 数据环境设计器

2) 在“数据环境”窗口中选择 Connection1 对象，右击，选择“属性”菜单项，打开“数据链接属性”对话框，如图 3-90 所示。



图 3-90 “数据链接属性”对话框

3) 在“数据链接属性”对话框中选择 Microsoft Jet 4.0 OLE DB Provider 项，单击“下一步”按钮，进入“连接”选项卡。

4) 在“连接”选项卡中单击“选择或输入数据库名称”按钮，打开“选择 Access 数据库”对话框，选择“资料室图书管理.mdb”数据库，单击“确定”按钮后返回“数据环境”窗口。

5) 选择 Connection1 对象, 右击, 在弹出的快捷菜单中选择“添加命令”菜单项, 为 Connection1 添加一个 Command1 对象, 右击 Command1, 在弹出的快捷菜单中选择“属性”菜单项, 如图 3-91 所示。



图 3-91 Command1 对象快捷菜单

6) 在弹出的“Command1 属性”对话框的“通用”选项卡中, 设置如图 3-92 所示内容。



图 3-92 “Command1 属性”对话框

(3) 制作报表。

1) 打开数据报表设计器, 选择“工程”→“添加 Data Report”命令, 打开“数据报表 1”窗口, 如图 3-93 所示。

2) 在对应于数据报表 1 的“属性”窗口中, 设置 DataSource 为 DataEnvironment1, DataMember 为 Command1。

3) 在对应于数据报表的工具箱中选择 Rpttable 控件 , 添加至报表标头区, 并将 Caption 属性设置为“资料室图书一览表”。

4) 从环境设计器中将各字段拖放至页标头、细节区域, 并调整文本位置如图 3-94 所示。

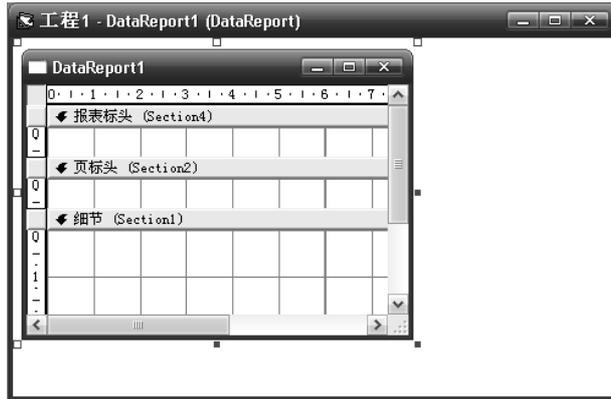


图 3-93 “数据报表 1” 窗口

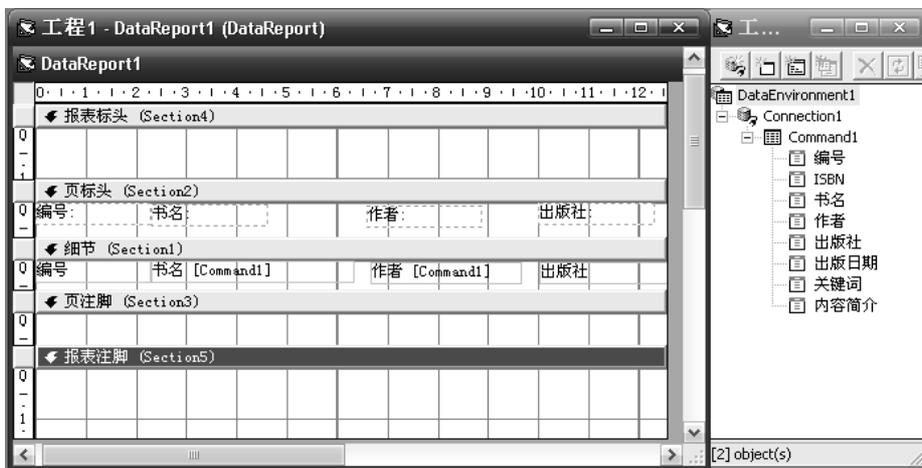


图 3-94 设计数据报表 1

程序运行结果如图 3-95 所示。



图 3-95 程序的执行结果

【实验内容】

1. 为本实验例题 1 的“图书信息登记与查阅”程序添加登录界面，只允许管理者登录，登录验证码保存在数据库的“登录信息”表中，界面设计如图 3-96 所示。



图 3-96 登录界面设计

2. 设计一个列车时刻表查询系统，界面设计如图 3-97 所示。



图 3-97 列车时刻表查询界面设计

3. 制作列车时刻报表，并为报表添加日期和页码。