第 6 章　数组

迄今为止介绍过的数据类型均为基本类型，一个基本类型的变量可以用来存放一个数值型数据或者字符串，然而在实际应用中，经常需要处理大批量相关的数据，有些数据本身还比较复杂（例如学生信息的描述），这时再采用基本类型就显得有些力不从心了。VB 语言提供了数组和自定义类型，数组是一组相同类型变量的集合，自定义类型是由多个简单类型聚合而成，用来描述复杂数据。本章主要讲解数组的相关知识，包括一维数组、二维数组、动态数组和控件数组，介绍自定义数据类型和字符串的处理方法，以及列表框控件和组合框控件。

6.1　一维数组

在引入数组的概念之前，先考虑一个实际的问题：某班有 30 位学生，统计该班 VB 语言考试的平均成绩。如果用以前的方法来解决这个问题，必然要先定义 30 个整型变量，再从键盘输入每一位学生的成绩，分别存放在这些变量里，然后利用循环结构计算成绩总和，再除以人数得到平均成绩。

如果进一步研究其编程实现的过程，会发现有一些难以处理的环节。首先分别定义 30 个变量就显得十分繁琐，如果学生的人数迅速膨胀，定义变量将是一件无法忍受的工作；其次在语法上没有体现这些变量之间的关联性，使得在循环体中无法表示任意一位学生的成绩。

上面这个问题涉及了相同类型的大量相关数据的处理，在程序中需要用到数组。数组是具有相同类型的相关数据的集合，利用数组可以较为方便地解决大量数据处理的问题。数组按结构来划分，可以分为一维数组、二维数组和多维数组，其中一维数组是基础。

6.1.1　一维数组的定义

一维数组的定义方式如下：

```
Dim 数组名([下界 To]上界) As 类型
```

例如：

```
Dim a(1 To 5) As Integer
```

表示定义了一个有 5 个元素的整型数组 a，一个元素相当于一个普通的整型变量，每个元素可以存放一个整型数据。数组的元素在内存中按顺序存放，数组所占据的字节数是各元素所占字节数之和，显然数组 a 在内存占 5×2=10（字节）。

说明：

（1）数组名应该是一个合法的标识符，数组中所有元素的数据类型都相同。

（2）下界和上界均为整型常量表达式，它们规定了元素下标的取值范围。下界最小可以是-32768，上界最大可以是 32767。应该满足下界≤上界，一维数组的长度即元素的个数为上界-下界+1。

（3）对于没有赋初值的数组元素，如果是数值型，系统都自动赋以 0；如果是字符型，

系统都自动赋以空串；如果是逻辑型，系统都自动赋以 False。

（4）如果定义数组时省略了下界，则下界默认是 0。也可以使用 Option Base 语句设置数组下界的默认值，例如：

```
Option Base 1
Dim b(10) As Integer
```

定义了一个有 10 个元素的整型数组 b，它的上界是 10，下界则是默认值 1。需要指出的是，Option Base 语句的参数只能是 0 或 1，而且该语句在一个模块中只能出现一次。

6.1.2　数组元素的引用

引用元素必须要在定义数组之后，元素引用的形式如下：

数组名(下标)

例如：

```
a(4)=a(1)*a(3)+a(2)
```

说明：在引用数组的元素时，应注意下标值不要超过数组的范围。假如某个数组的下界为 0，上界为 10，则其下标值的范围应该是 0～10。超过数组范围的现象称为下标越界，系统会予以报错。

下标从下界开始，到上界结束，它实际上是数组元素的序号，表示该元素在数组中的相对位置。例如数组 a 的第一个元素是 a(1)，最后一个元素是 a(5)，a(3)的后一个元素是 a(4)，前一个元素是 a(2)等。数组 a 在内存的存储结构如图 6-1 所示。

0	a(1)
0	a(2)
0	a(3)
0	a(4)
0	a(5)

图 6-1　一维数组的存储结构

6.1.3　数组的处理

数组是包含大量同类型相关数据的集合，在程序中显然应该通过循环结构来处理数组。那么如何实现呢？请注意下标，它表示元素在数组中的相对位置，数组名加上( )以及下标，就能够表示数组中的任意一个元素。我们可以用 For-Next 语句的循环变量存放下标，它从下界开始，不断加 1，在达到上界之后结束。用循环变量作为数组下标，可以按顺序访问每一个数组元素，这是处理数组的通用做法。通常在程序中设置 3 个循环结构，第一个用于输入数组中的元素，第二个用于处理数组元素，最后一个则用于输出数组中的所有数据。

【例 6.1】某班有 30 位学生，分别输入全班学生的 VB 语言成绩，计算平均成绩并输出。

分析：定义一个长度为 30 的整型数组，用来存放全班学生的成绩。采用 For-Next 语句进行处理，下标初始为 1，每次循环不断加 1，到 30 为止。在循环体中累加每一位学生的成绩，最后除以人数得到平均成绩。

```
Const N As Integer = 30
Private Sub Command1_Click()
Dim a(1 To N) As Integer, i As Integer, sum As Integer, aver!
For i = 1 To N '输入学生成绩
 a(i) = Val(InputBox("请输入第" & i & "位学生的成绩"))
Next i
sum = 0
For i = 1 To N '累加学生成绩
 sum = sum + a(i)
Next i
aver = sum / N '计算平均成绩
Picture1.Print "平均成绩是"; aver
End Sub
```

运行程序，结果如图 6-2 所示。

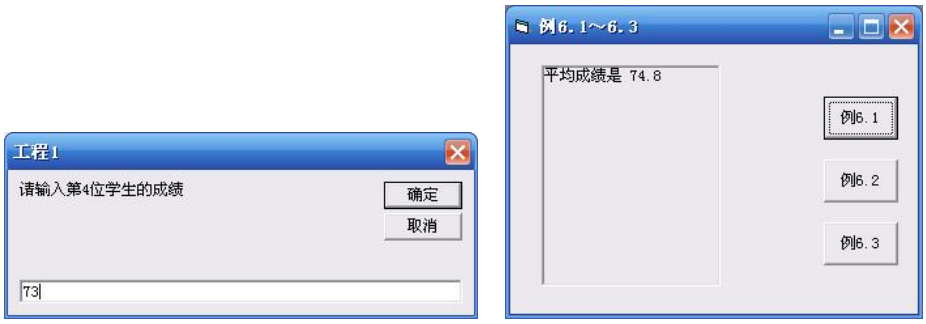

图 6-2　例 6.1 的运行结果

说明：在程序的第一个 For-Next 语句中，反复调用 InputBox 函数输入学生成绩，并分别存放在数组 a 的各个元素中。为增加程序的通用性，在事件过程的外部定义了一个符号常量 N，表示数组的上界。如果需要修改数组的上界，则只需修改符号常量的初值即可。

【例 6.2】某班有 30 位学生，任意输入一个姓名，查询该班是否存在与该姓名对应的学生。

分析：定义一个长度为 30 的字符串数组，用来存放全班学生的姓名。采用 For-Next 语句进行处理，在循环体中穷举所有学生的姓名，用 If 语句依次判断，看看是否存在被查询的该姓名，最终得到判断结果。

```
Const N As Integer = 30
Private Sub Command2_Click()
Dim a(1 To N) As String, i%, j%, flag As Boolean, name$
For i = 1 To N '输入学生姓名
 a(i) = InputBox("请输入第" & i & "位学生的姓名")
Next i
Do
 name = InputBox("请输入要查询的学生姓名")
 flag = False
 For i = 1 To N
  If a(i) = name Then
   flag = True '找到，改变标志
```

```
        Exit For
       End If
      Next i
      If flag = True Then
       Picture1.Print "找到姓名为"; name; "的学生"
      Else
       Picture1.Print "没有找到姓名为"; name; "的学生"
      End If
      j = MsgBox("还要继续查询吗？", vbYesNo + vbquestin)
    Loop While j = 6 '如果按下“是”按钮，则继续循环
    End Sub
```

运行程序，结果如图 6-3 所示。

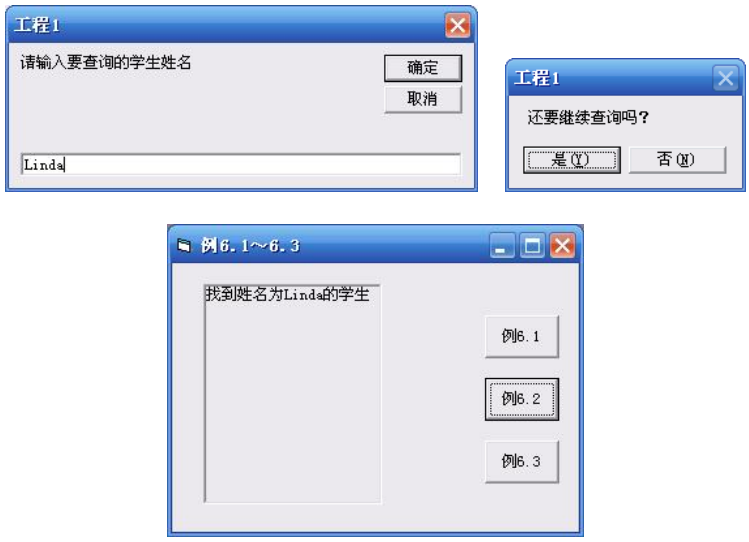

图 6-3　例 6.2 的运行结果

说明：在程序中用 Do-Loop 语句实现了连续查询，当完成一次查询工作之后，弹出一个消息对话框请用户做出选择。如果用户按下“是”按钮，则继续循环，进行另一次查询工作；如果用户按下“否”按钮，就退出循环，结束查询工作。

本题采用的查找算法是顺序查找法，在 6.9 节会给出效率较高的折半查找法。

【例 6.3】某班有 30 位学生，分别输入全班学生的 VB 语言成绩，统计其最高分和最低分，并输出结果。

分析：定义一个长度为 30 的整型数组，用来存放全班学生的成绩。采用 For-Next 语句进行处理，在循环体中用选择排序法比较每一位学生的成绩，最后得到最高分和最低分。

```
Const N As Integer = 30
Private Sub Command3_Click()
Dim a(1 To N) As Integer, i As Integer, max As Integer, min As Integer
For i = 1 To N '输入学生成绩
 a(i) = Val(InputBox("请输入第" & i & "位学生的成绩"))
Next i
max = a(1) '假定第一位学生的成绩是最高分
```

```
    min = a(1) '假定第一位学生的成绩是最低分
    For i = 2 To N
     If max < a(i) Then
      max = a(i) '确保 max 是当前最高分
     End If
     If min > a(i) Then
      min = a(i) '确保 min 是当前最低分
     End If
    Next i
    Picture1.Print "最高分是"; max
    Picture1.Print "最低分是"; min
    End Sub
```

运行程序，结果如图 6-4 所示。

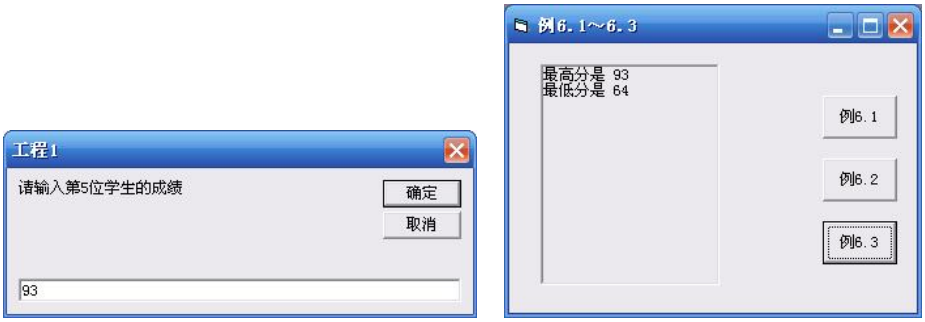

图 6-4　例 6.3 的运行结果

6.2　二维数组

二维数组主要用于描述二维的物理对象，以及保存具有二维逻辑结构的数据集合。在引入二维数组的概念之前，先考虑一个实际的问题：计算机专业有 3 个班，每班有 30 位学生，输入每位学生的 VB 语言考试成绩，输出该专业的 VB 语言考试平均成绩以及各班的最高分。

读者自然会想到用刚学过的一维数组的知识来解决，先定义一个长度为 90 的整型数组，输入各位学生的 VB 语言考试成绩，然后利用一重循环进行处理。但是进一步研究编程实现的过程，又会发现遇到了新问题。班与班之间如何划分？如何判断某一位同学属于哪个班？这些问题用一维数组很难得到解决。如果把班作为行，学生作为列，定义二维数组来描述计算机专业三个班的学生成绩，这样做就显得很自然，处理起来也比较方便。

6.2.1　二维数组的定义

二维数组的定义方式如下：

```
Dim 数组名([下界 To]上界,[下界 To]上界) As 类型
```

例如：

```
Dim a(1 To 2,1 To 2) As Integer
```

表示定义了一个 2 行 2 列的二维整型数组 a，它的逻辑结构如表 6-1 所示。数组 a 有 4 个

元素，分别是 a(1,1)、a(1,2)、a(2,1)和 a(2,2)，每个元素可以存放一个整型数据。

表 6-1　数组 a 的逻辑结构

a(1,1)	a(1,2)
a(2,1)	a(2,2)

说明：

（1）通常把二维数组的第一个下标形象地称为行下标，第二个下标称为列下标。

（2）二维数组的元素个数为行的长度×列的长度，行或者列的长度为各自的上界-下界+1。

（3）类似地还可以定义多维数组。例如：

```
Dim a(1 To 2,1 To 2,1 To 2) As Integer   '共有 8 个元素的三维数组
```

思考：三维数组的元素个数是多少？

6.2.2　二维数组的处理

在学习一维数组时，我们已经知道了用循环变量控制下标，通过 For-Next 语句使下标不断加 1，实现访问数组全部元素的目的。二维数组的处理方法与一维数组相似，只不过有两个循环变量用来分别控制行下标和列下标，通过二重循环结构实现访问数组全部元素的目的。

【例 6.4】求两个 3×3 矩阵的和。

分析：矩阵求和公式是 C=A+B，$c_{ij}=a_{ij}+b_{ij}$，即两个矩阵之和仍然是一个矩阵，其元素值是 A、B 两个矩阵相应位置的元素值之和。首先应定义 3 个 3 行 3 列的二维数组，分别用来表示 A、B 和 C 三个矩阵。然后采用二重循环结构，行下标与列下标都从 1 开始，分别不断地加 1，实现对矩阵各个元素的访问。

```
Private Sub Command1_Click()
Const N As Integer = 3
Dim a(1 To N, 1 To N) As Integer, b(1 To N, 1 To N) As Integer
Dim c(1 To N, 1 To N) As Integer, i As Integer, j As Integer
For i = 1 To N
      For j = 1 To N
       a(i, j) = Val(InputBox("输入 a(" & i & "," & j & ")")) '输入数据存入
数组 a
      Next j
Next i
MsgBox ("矩阵 A 的数据输入完毕！")
For i = 1 To N
  For j = 1 To N
        b(i, j) = Val(InputBox("输入 b(" & i & "," & j & ")")) '输入数据存
入数组 b
       Next j
Next i
MsgBox ("矩阵 B 的数据输入完毕！")
Picture1.Print "开始输出矩阵 C 的数据"
For i = 1 To N
       For j = 1 To N
```

```
            c(i, j) = a(i, j) + b(i, j) '矩阵求和
        Next j
    Next i
    For i = 1 To N
        For j = 1 To N
            Picture1.Print Tab(j * 4); c(i, j); '输出数组 c
        Next j
        Picture1.Print      '输出一行数据，另换一行
    Next i
    End Sub
```

运行程序，结果如图 6-5 所示。

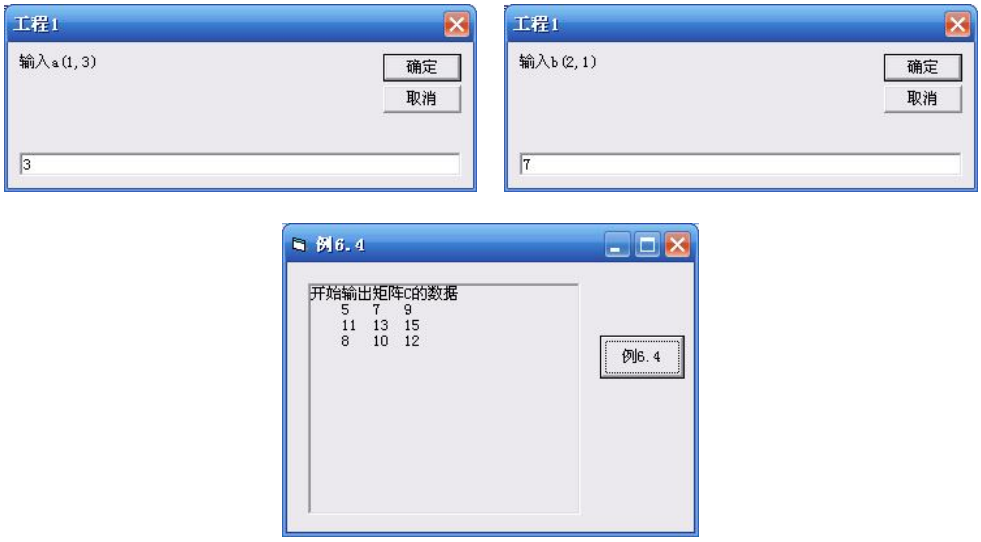

图 6-5　例 6.4 的运行结果

说明：通常把二维数组的行下标放在外层循环，而把列下标放在内层循环。

【例 6.5】计算机专业有 3 个班，每班有 30 位学生，输入每位学生的 VB 语言考试成绩，输出计算机专业 VB 语言考试的平均成绩以及各班的最高分。

分析：首先定义一个 3 行 30 列的二维整型数组，用于存放计算机专业各班学生的 VB 语言考试成绩，再定义一个一维数组用来存放各班的最高分。对二维数组的处理与例 6.4 相似，在内层的循环体中用选择排序算法寻找各班的最高分，并累加所有学生的成绩。

```
Private Sub Command1_Click()
Const M As Integer = 3, N As Integer = 30
Dim a(1 To M, 1 To N) As Integer, max(1 To M) As Integer, i%, j%
Dim sum As Integer, aver As Single
For i = 1 To M
 For j = 1 To N
  a(i, j) = Val(InputBox("输入 a(" & i & "," & j & ")")) '输入成绩存入数组 a
 Next j
Next i
MsgBox ("学生成绩输入完毕！")
```

```
sum = 0
For i = 1 To M
  max(i) = a(i, 1)        '假定各班第一位学生是各班最高分
  For j = 1 To N
   If max(i) < a(i, j) Then
    max(i) = a(i, j)
   End If
   sum = sum + a(i, j)
  Next j
Next i
aver = sum / (M * N)
Picture1.Print " 平均成绩是"; Format(aver, "##.##") '输出平均成绩
For i = 1 To M
  Picture1.Print i; "班的最高分是"; max(i) '输出各班最高分
Next i
End Sub
```

运行程序，结果如图 6-6 所示。

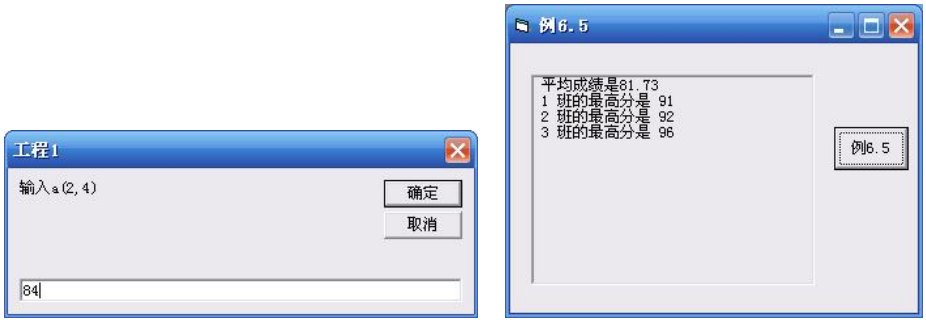

图 6-6　例 6.5 的运行结果

说明：也可以采用 For Each-Next 语句，输出数组 max 的所有元素。这样做的优点是不需要事先了解数组中元素的个数，从而简化了循环条件的设置。部分代码如下：

```
For Each x In max
   Picture1.Print x
Next x
```

思考：三维数组的数据如何输入、输出及处理？

6.3　动态数组

前面介绍的数组都是在定义时指定数组的类型、维数以及每一维的长度，这些数组被称为静态数组。如果无法预知元素的个数和数组的维数，就很难精确地定义静态数组。定义得过大则浪费系统的内存，定义得过小又会满足不了实际的需要。VB 语言允许定义动态数组，以增强程序的灵活性，提高内存使用的效率。

动态数组在程序运行过程中才被分配存储空间，它的定义方式如下：

```
Dim 数组名() As 类型
```

例如：

```
Dim a() As Integer
```

表示定义了一个动态整型数组 a，数组的维数以及元素下标的下界和上界未知。可以用数组名赋值的方式，把一个静态数组中全部元素的值依次赋给一个动态数组中的全部元素。例如：

```
Dim a(1 To 3) As Integer, b() As Integer, i%
For i = 1 To 3 '对静态数组 a 的所有元素赋值
 a(i) = i
Next i
b = a '数组名赋值
For Each x In b '输出动态数组 b 中所有元素的值
 Print x
Next x
```

数组名赋值的方式自动确定了动态数组 b 的维数以及元素下标的下界和上界，它们均与静态数组 a 相同。也可以调用 LBound 和 UBound 函数，分别获得数组的下界和上界。这两个函数的格式如下：

```
LBound(a[,n])
UBound(a[,n])
```

说明：

（1）参数 a 是数组名。参数 n 表示数组 a 的第 n 维，如果省略，则默认是 1。

（2）LBound 函数返回数组 a 第 n 维的下界，UBound 函数返回数组 a 第 n 维的上界。

变体型数组的各个元素能够存放不同类型的数据，如果是动态变体型数组，则可以通过 Array 函数进行初始化，并自动确定动态数组中元素的个数。例如：

```
Dim b(), i%
b = Array(1, 2, 3)
For i = 0 To 2
 Print b(i)
Next i
```

定义数组 b 时，既未指定维数以及元素下标的下界和上界，也未指定数据类型，因此它是动态变体型数组。在 Array 函数中有 3 个参数，作为初值依次赋给了数组 b 的各个元素。由此确定了动态数组 b 中元素个数为 3，下界默认是 0，上界则为 2。

定义了一个动态数组之后，一旦需要即可在程序中使用 ReDim 语句，确定动态数组的维数以及元素下标的下界和上界。其一般形式如下：

```
ReDim [Preserve]数组名([下界 To]上界[,下界 To 上界,…]) [As 类型]
```

说明：

（1）可以多次使用 ReDim 语句对某个动态数组进行设置。

（2）数组的维数以及元素下标的下界和上界都能够改变，甚至下界和上界可以是有了确定值的变量，但是数组的类型不能改变。

（3）每次执行 ReDim 语句之后，数组中所有元素的值将会丢失。如果想保留数组元素的值，则可以使用关键字 Preserve。例如：

```
Dim a() As Integer
  …
ReDim a(2,3) '数组设置为 3 行 4 列
```

```
...
ReDim Preserve a(2,4) '数组设置为 3 行 5 列，并保留数组元素的值
```

在 ReDim 语句中使用关键字 Preserve 时，只能改变动态数组最后一维的上界。

【例 6.6】计算并输出 Fibonacci 数列的前 n 项。

分析：Fibonacci 数列从第三个数开始，每个数都是其前面两个数之和。由于 Fibonacci 数之间存在明显的位置关系，所以用数组来处理最为便利。定义一个动态长整型数组 a，用来存放 Fibonacci 数列。从文本框接收用户输入的 n 值之后，采用 ReDim 语句将数组 a 的长度置为 n。

```
Private Sub Command1_Click()
Dim a() As Long, n As Integer, i As Integer, j%
n = Val(Text1.Text)
ReDim a(1 To n) '设置动态数组的长度
For i = 1 To n
 If i = 1 Or i = 2 Then
  a(i) = 1 '第一项和第二项都是 1
 Else
  a(i) = a(i - 1) + a(i - 2) '每一项是前两项之和
 End If
Next i
j = 0
For i = 1 To n
 Picture1.Print Tab(j * 7); a(i);
 j = j + 1
 If i Mod 5 = 0 Then
  Picture1.Print
  j = 0
 End If
Next i
End Sub
```

运行程序，结果如图 6-7 所示。

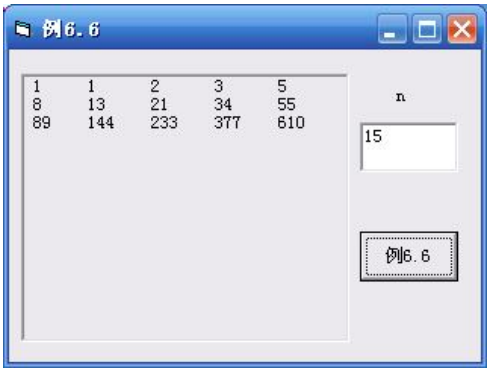

图 6-7　例 6.6 的运行结果

在程序中有时需要对数组重新进行初始化，则可以使用 Erase 语句达到目的，其一般形式如下：

```
Erase 数组名
```

说明：执行 Erase 语句之后，对于静态数组，系统会清除数组中的原有数据，并自动进行初始化；对于动态数组，系统会删除数组的结构，并释放数组所占的内存空间。如果以后想再次使用该动态数组，就必须采用 ReDim 语句重新对其进行设置。例如：

```
Dim a(1 To 5) As Integer, i%
For i = 1 To 5
 a(i) = i
Next i
For i = 1 To 5 '输出数组 a 所有元素的值
 Print a(i);
Next i
Print
Erase a '对数组 a 重新初始化
For i = 1 To 5 '再次输出数组 a 所有元素的值
 Print a(i);
Next i
```

该程序段的运行结果如图 6-8 所示。从图中可以看到，执行 Erase 语句之后，数组 a 所有元素的值都重新初始化为 0。

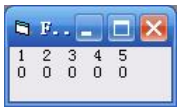

图 6-8　Erase 语句的效果

6.4　控件数组

一组相同类型的相关数据可以用数组来描述和管理，那么一组功能相似的同类控件是否也能够用数组进行组织？回答是肯定的，这样的数组称为控件数组。控件数组由一组同属于一类的控件组成，它们共用一个对象名，依靠索引（Index）属性彼此区分。

如何创建控件数组？主要有以下几种方法：

（1）复制现有的控件，然后粘贴在窗体中。第一次进行粘贴操作时，系统会提示是否创建控件数组，选择“是”即可。此时大多数的可视属性，如颜色、高度和宽度等，将会从源控件即数组中的第一个控件，复制到目标控件即新控件中。

（2）为现有的同类控件取同一个对象名，一般是与第一个控件的名字一致，例如 Text1、Command1 等。这时系统也会提示是否创建控件数组，选择“是”即可。用这种方法创建的控件数组，其控件元素的属性值只是名字（Name）相同，其他属性依然保留最初创建这些控件时的设置。

如何访问控件数组中的元素？利用控件的 Index 属性。与数组的下标相似，Index 表示控件在控件数组中的相对位置，默认从 0 开始，依次加 1。控件元素的访问方法与普通数组的元素基本相同，例如现有控件数组 Text1，要对其中 Index 属性值为 1 的文本框控件，设置 Text 属性值为“VB6.0”，可以写为：

```
Text1(1).Text="VB6.0"
```

在程序中使用控件数组，不仅可以借助循环结构统一处理数组中的控件，而且可以共享同一个事件处理过程。例如设计一个计算器，在窗体中安排 4 个命令按钮，分别完成加减乘除四则运算。考虑到这些命令按钮实现的功能相似，可以创建一个有 4 个元素的命令按钮控件数组，其中每一个命令按钮对应一个控件数组的元素。然后为这个控件数组定义一个单击事件过程，只要用户任意按下 4 个命令按钮中的一个，就会调用这个事件过程。此外还可以在程序中调用 Load 方法，动态创建控件数组中的新元素，达到在程序运行时创建新控件的目的。

【例 6.7】用控件数组改写例 4.6 的程序。

分析：在界面设计阶段分别创建 Option1 和 Check1 两个控件数组，数组 Option1 用于组织 4 个单选按钮，供学生选择系别；数组 Check1 用于组织 5 个复选框，供学生选择爱好。在程序中分别定义两个字符串数组 t1 和 t2，t1 存放所有的系别，t2 存放所有的爱好。在循环结构中判断用户选择了哪些选项，并做相应的处理。

```
Private Sub Command1_Click()
Dim s As String, i As Integer
Dim t1(3) As String, t2(4) As String
t1(0) = "计算机": t1(1) = "汽车"
t1(2) = "机械": t1(3) = "管理"
t2(0) = "足球 ": t2(1) = "围棋 "
t2(2) = "游泳 ": t2(3) = "文学 "
t2(4) = "上网 "
s = s + "姓名：" + Text1.Text + vbCr
s = s + "年龄：" + Text2.Text + vbCr
For i = 0 To 3
 If Option1(i).Value = True Then
  s = s + t1(i) + "系" + vbCr
  Exit For
 End If
Next i
s = s + "爱好："
For i = 0 To 4
 If Check1(i).Value = 1 Then
  s = s + t2(i)
 End If
Next i
MsgBox (s)
End Sub
Private Sub Command2_Click()
 End
End Sub
```

程序的运行结果与例 4.6 完全相同。

说明：与例 4.6 的程序相比，本例程序中 If 语句的数量以及分支的个数都明显减少了，结构也更为紧凑。

思考：在程序的第一个 For-Next 语句中，为何出现 Exit For？而在第二个 For-Next 语句中，为何又没有出现 Exit For？

6.5 自定义类型

利用数组可以将一群相同类型的数据组织在一起，但在实际应用中，经常会遇到由多种不同类型数据组成的实体。例如，描述一个学生的数据实体包括学号、姓名、性别、年龄和成绩等数据项，这些类型不同的数据项是相互联系的，组成一个有机的整体。如果用独立的简单数据项分别表示它们，就不能体现数据的整体性，也不便于进行整体操作；又不能用普通数组来存放这些类型不同的数据。

VB 语言允许程序员自定义数据类型，这种自定义的类型又称为记录类型，它由一些基本类型的成员所组成。定义记录类型的关键字是 Type，其一般形式如下：

```
Type 记录类型名
 成员表列
End Type
```

说明：

（1）对成员表列中的所有成员都应进行类型声明。成员声明的形式如下：

```
成员名 As 类型
```

（2）记录类型只是刻画了一个数据结构的模型，并没有定义实例，也不要求分配实际的内存空间。在程序中使用记录类型时，必须定义记录变量。

例如，学生信息可以用记录类型描述为：

```
Type Student
  sno As Long        '学号
  name As String     '姓名
  sex As String      '性别
  score As Integer   '成绩
End Type
```

Student 类型有 4 个成员，分别表示学生的学号、姓名、性别和成绩，这些成员的类型可以不相同。需要指出的是，通常在程序的标准模块（.bas）中定义记录类型。在“工程”菜单中选择“添加模块”命令，即可创建标准模块。如果在窗体模块中定义记录类型，则必须用关键字 Private 进行声明。

先定义记录类型，再定义记录变量。记录变量所占内存空间的长度，是其各个成员所占空间的长度之和。定义记录变量的方法与定义普通变量基本相同，只不过数据类型是记录类型。例如：

```
Dim s1 As Student,s2 As Student '定义 s1 和 s2 为 Student 类型的变量
```

访问一个记录变量的目的通常是引用它的成员，例如登记学生的姓名、统计学生的成绩等。引用记录变量成员的形式如下：

```
记录变量名.成员名
```

s1.sno 表示引用记录变量 s1 中的 sno 成员，它可以像普通变量一样使用，能够进行赋值等合法的运算。例如：

```
s1.sno=2051226 '将 2051226 赋给 s1 变量的成员 sno
```

一个 Student 类型的记录变量可以存放某个学生的一组相关信息，如果有多个学生的数据需要处理，则应使用记录数组。例如：

```
Dim s(1 To 10) As Student
```

定义了一个数组 s，它有 10 个元素，每个元素都相当于一个 Student 类型的记录变量。如何访问记录数组元素的成员？其一般形式如下：

```
记录数组名(下标).成员名
```

例如：

```
s(2).sno=2051227
Text1.Text= s(6).name
```

【例 6.8】用记录类型改写例 6.3。要求不仅输出最高分和最低分，还要输出拥有这些分数的相关学生的姓名。

分析：在标准模块中定义 Student 记录类型，成员有姓名和成绩，成员的类型分别为字符串和整型。

```
Type Student
   name As String
   score As Integer
End Type
```

在窗体模块中定义一个长度为 30 的 Student 型数组，用来存放全班学生的姓名和成绩。采用 For-Next 语句进行处理，具体过程与例 6.3 的程序十分相似。

```
Const N As Integer = 30
Private Sub Command1_Click()
Dim a(1 To N) As Student, max As Student, min As Student, i%
For i = 1 To N '输入学生的姓名和成绩
 With a(i) 'With 语句
  .name = InputBox("请输入第" & i & "位学生的姓名")
  .score = Val(InputBox("请输入第" & i & "位学生的成绩"))
 End With
Next i
max = a(1)  '假定第一位学生的成绩是最高分
min = a(1)  '假定第一位学生的成绩是最低分
For i = 2 To N
 If max.score < a(i).score Then
  max = a(i) '确保 max 是当前成绩最高的学生
 End If
 If min.score > a(i).score Then
  min = a(i) '确保 min 是当前成绩最低的学生
 End If
Next i
Picture1.Print "最高分是"; max.name; max.score; "分"
Picture1.Print "最低分是"; min.name; min.score; "分"
End Sub
Private Sub Command2_Click()
End
End Sub
```

运行程序，结果如图 6-9 所示。

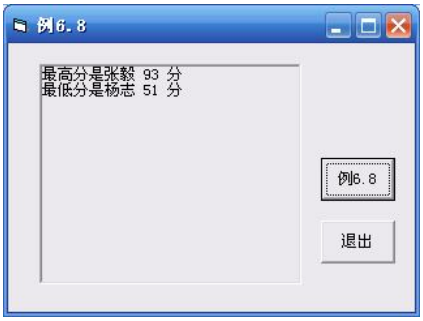

图 6-9　例 6.8 的运行结果

说明：在程序中使用了 With 语句，以简化语句的书写。如果用 With 语句对某个记录变量作出声明，则在该语句的作用域中访问这个记录变量的成员时，可以省略记录变量名。

6.6　字符串的处理

文本是日常生活中司空见惯的一种数据形式，也是经常需要借助计算机进行处理的对象，记事本和 Word 等程序就是专门用于文本编辑的软件。VB 语言以字符串表示文本数据，在程序中除了可以用运算符进行字符串的连接和比较等基本操作之外，还可以调用内部函数完成字符串的查找、截取等一些高级操作。

1. 格式转换

字符串的格式转换是指字符串与数值的相互转换，以及字符与编码的相互转换等操作。

（1）Val 函数。Val 函数的格式如下：

```
Val(s)
```

说明：该函数的功能是把字符串 s 转换为一个数值。例如 Val("1234")得到的值是 1234。

（2）Str 函数。Str 函数的格式如下：

```
Str(n)
```

说明：该函数的功能是把数值 n 转换为一个字符串。例如 Str(1234)得到的值是"1234"。

（3）Asc 函数。Asc 函数的格式如下：

```
Asc(s)
```

说明：该函数的功能是，把字符串 s 中的第一个字符转换为相应的 ASCII 码。例如 Asc("ab")得到的值是 97，即字符 a 的 ASCII 码。

（4）Chr 函数。Chr 函数的格式如下：

```
Chr(n)
```

说明：该函数的功能是把数值 n 即 ASCII 码转换为所对应的字符。例如 Chr(97)得到的值是"a"，而 Chr(Asc("a")-32)得到的值是"A"，因为小写字母的 ASCII 码比相应大写字母的 ASCII 码大 32。

思考：如何随机产生一个小写字母？

（5）UCase 函数。 UCase 函数的格式如下：

```
UCase(s)
```

说明：该函数的功能是把字符串 s 中的小写字母转换为大写形式。例如 UCase("aBCdE")得到的值是"ABCDE"。

（6）LCase 函数。LCase 函数的格式如下：

```
LCase(s)
```

说明：该函数的功能是把字符串 s 中的大写字母转换为小写形式。例如 LCase("AbCdE")得到的值是"abcde"。

2. 统计长度

函数 Len 用于统计字符串的长度即所包含字符的个数，其格式如下：

```
Len(s)
```

例如，Len("VB6.0 环境")得到的值是 7。

3. 删除多余的空格

函数 LTrim 删除字符串中前面的空格，函数 RTrim 删除字符串中后面的空格，函数 Trim 则删除字符串中前后两边的空格。它们的格式如下：

```
LTrim(s)
RTrim(s)
Trim(s)
```

例如，LTrim("　VB6.0　")得到的值是"VB6.0　"，RTrim("　VB6.0　")得到的值是"　VB6.0"，而 Trim("　VB6.0　")得到的值是"VB6.0"。

4. 生成字符串

字符串的生成是指按照要求产生一个字符串。

（1）String 函数。String 函数的格式如下：

```
String(m,s|n)
```

说明：该函数的功能是，产生一个由 m 个重复的字符组成的字符串。该字符是字符串 s 中的第一个字符，或者是 ASCII 码为数值 n 的字符。例如 String(3,"ab")得到的值是"aaa"，String(3,97)得到的值也是"aaa"。

（2）Space 函数。Space 函数的格式如下：

```
Space(n)
```

说明：该函数的功能是产生一个由 n 个空格组成的字符串。例如"VB"+Space(3)+"6.0"得到的值是"VB　6.0"。

5. 查找和替换

字符串的查找是指在一个字符串中查找另一个指定的字符串，字符串的替换是指在一个字符串中把一个子串替换为另一个子串。

（1）InStr 函数。InStr 函数的格式如下：

```
InStr([n,]s1,s2)
```

说明：该函数的功能是，在字符串 s1 中查找字符串 s2 首次出现的位置。参数 n 是字符串 s1 的起始查找位置，如果省略则默认值是 1，表示从头开始查找。如果找到，函数的返回值是字符串 s2 在字符串 s1 中第一次出现的位置；如果未找到，则函数的返回值是 0。例如 InStr("我是中国人","中国")得到的值是 3，InStr(4,"VB6.0 环境","环境")得到的值是 6，而 InStr("欲穷千里目，更上一层楼","千里眼")得到的值是 0。

思考：InStr(4,"我是中国人","中国")得到的值是什么？

（2）Replace 函数。Replace 函数的格式如下：

```
Replace(s1,s2,s3[,m][,n][,…])
```

说明：该函数的功能是，在字符串 s1 中把子串 s2 替换为子串 s3。参数 m 是字符串 s1 的起始查找位置，此时在函数的返回值中会删除位置 m 之前的字符。如果省略 m 则默认值是 1，表示从头开始查找。参数 n 是进行替换操作的最大次数，如果省略 n 则默认值是-1，表示替换所有符合条件的子串。例如 Replace("aAAbAAc","AA","BB") 得到的值是"aBBbBBc"，Replace("aAAbAAc","AA","BB",3) 得到的值是"AbBBc"，Replace("aaAAbAAc","AA","",2,1) 得到的值是"abAAc"，相当于从第 2 个位置开始，删除第一次出现的子串"AA"。

思考：Replace("aAAbAAc","AA","BB",,1) 得到的值是什么？

6. 截取子串

字符串的截取是指从字符串中连续抽取若干个字符，组成一个新的字符串即子串。

（1）Left 函数。Left 函数的格式如下：

```
Left(s,n)
```

说明：该函数的功能是从字符串 s 的左边取出 n 个字符，组成一个子串。例如 Left("VB6.0 环境",5)得到的值是"VB6.0"。

（2）Right 函数。Right 函数的格式如下：

```
Right(s,n)
```

说明：该函数的功能是从字符串 s 的右边取出 n 个字符，组成一个子串。例如 Right("VB6.0 环境",2)得到的值是"环境"。

（3）Mid 函数。Mid 函数的格式如下：

```
Mid(s,m[,n])
```

说明：该函数的功能是从字符串 s 的第 m 个字符开始，取出 n 个字符，组成一个子串。如果省略 n，则表示取出从第 m 个字符开始的所有字符。例如 Mid("VB6.0 环境",3,3)得到的值是"6.0"，Mid("VB6.0 环境",3) 得到的值是“6.0 环境”。实际上 Mid(s,1,n)的作用等价于 Left(s,n)，Mid(s,Len(s)-n+1)的作用等价于 Right(s,n)。

（4）Split 函数。Split 函数的格式如下：

```
Split(s[,d][,n][,…])
```

说明：该函数的功能是从字符串 s 中取出 n 个子串，子串之间的分隔符是参数 d。如果省略 d，默认分隔符是空格；如果省略 n 则默认值是-1，表示取出所有的子串。通常把函数的返回值赋给一个动态字符串数组，数组的每一个元素依次存放一个子串。

以前介绍的数组输入方法，是采用循环结构反复调用 InputBox 函数，这种方法未免有些单调。Split 函数可以用来一次性地给一个数组赋初值，提高数据输入的效率。例如用户先在文本框中输入所有的数据，数据之间以事先约定的字符进行分隔，然后调用 Split 函数取出所有的数据，依次存入数组的各个元素。读者可能会问，在这种情况下数组的下界是多少？上界又是多少？我们可以调用 LBound 函数得到数组的下界，调用 UBound 函数得到数组的上界。

【例 6.9】改写例 6.2。要求在文本框中输入所有学生的姓名。

分析：在窗体中安排一个文本框，用于输入全班所有学生的姓名。定义一个动态字符串数组 a，调用 Split 函数，从文本框中读取姓名，存放到数组 a 中。具体的查找过程与例 6.2 的程序基本相同。

```
Private Sub Command1_Click()
```

```
Dim a() As String, i%, j%, flag As Boolean, name$
a = Split(Text1.Text) '输入学生姓名
Do
 name = InputBox("请输入要查询的学生姓名")
 flag = False
 For i = LBound(a) To UBound(a)
  If a(i) = name Then
   flag = True '找到，改变标志
   Exit For
  End If
 Next i
 If flag = True Then
  Picture1.Print "找到姓名为"; name; "的学生"
 Else
  Picture1.Print "没有找到姓名为"; name; "的学生"
 End If
 j = MsgBox("还要继续查询吗？", vbYesNo + vbquestin)
Loop While j = 6
End Sub
Private Sub Command2_Click()
End
End Sub
```

运行程序，结果如图 6-10 所示。

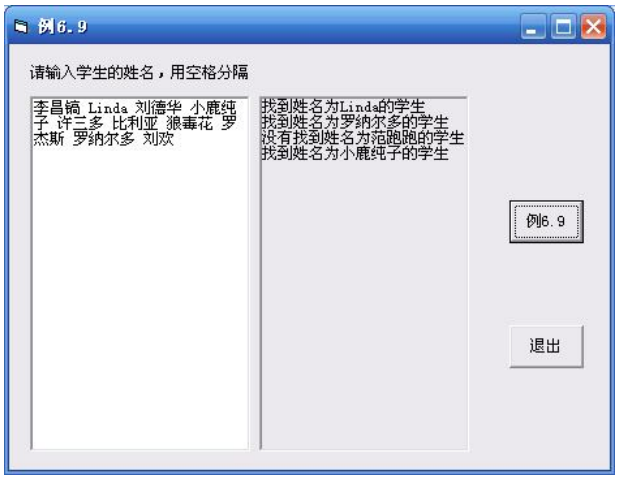

图 6-10　例 6.9 的运行结果

说明：文本框的 MultiLine 属性值设置为 True，使得输入时能够自动换行。约定输入学生的姓名时，以空格进行分隔。

【例 6.10】统计一个字符串中数字、字母以及其他字符的个数。

分析：定义三个计数器记录相关字符的个数。在循环语句中调用 Mid 函数，依次取出各个字符，用 If 语句的 ElseIf 结构判断字符的特征，相应计数器则不断加 1。

```
Private Sub Command1_Click()
```

```
Dim a As String, b As String, i%, k1%, k2%, k3%
a = Text1.Text
k1 = 0 '计数器清 0
k2 = 0
k3 = 0
For i = 1 To Len(a)
 b = Mid(a, i, 1) '取出字符串中第 i 个字符
 If b >= "A" And b <= "Z" Or b >= "a" And b <= "z" Then
  k1 = k1 + 1 '字母的个数增 1
 ElseIf b >= "0" And b <= "9" Then
  k2 = k2 + 1 '数字的个数增 1
 Else
  k3 = k3 + 1 '其他字符的个数增 1
End If
Next i
Picture1.Print "字母的个数为"; k1
Picture1.Print "数字的个数为"; k2
Picture1.Print "其他字符的个数为"; k3
End Sub
```

运行程序，结果如图 6-11 所示。

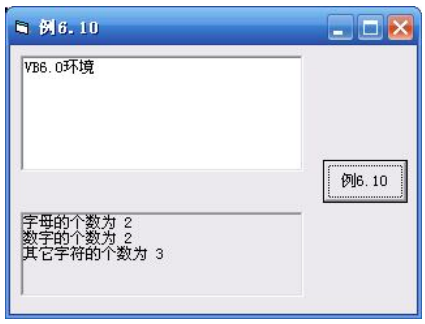

图 6-11　例 6.10 的运行结果

说明：Mid(a, i, 1)的作用是从字符串 a 的第 i 个位置开始，取出 1 个字符，即得到第 i 个字符。随着 i 不断加 1，就可以得到字符串 a 的每一个字符。

6.7　列表框

如果可供用户选择的项目较少，一般采用一组单选按钮或复选框。但是如果存在大量的选项时，采用单选按钮和复选框就显得十分繁琐，而采用列表框或者组合框则不失为一个较好的解决方案。列表框（ListBox）控件能够显示一个项目列表，用户可以从中选择一个或多个项目。如果项目列表中的项目过多而无法一次全部显示，则列表框将自动出现滚动条。在 VB 的工具箱中，列表框控件的图标如图 6-12 所示。

图 6-12　列表框图标

1. 属性

表 6-2 列出了列表框控件的常用属性。

表 6-2　列表框的常用属性

属性	作用
Name	设置列表框的对象名
Text	确定用户当前所选的项目，该属性不能在属性窗口中设置，只能在程序中设置或引用
List	设置列表框所显示的项目列表
ListCount	确定列表框中项目的总数，该属性只能在程序中设置或引用
ListIndex	确定当前选中的项目在项目列表中的索引值，该属性只能在程序中设置或引用
Selected	确定项目列表中某个项目是否被选中，该属性只能在程序中设置或引用
MultiSelect	确定列表框是否允许多选
Style	设置列表框的外观，默认值是 0，表示标准方式；如果是 1，则项目的左边有复选框

说明：

（1）程序第一个列表框控件的默认对象名是 List1，第二个列表框控件的默认对象名是 List2，依此类推。

（2）List 是列表框控件最重要的属性之一，其属性值是一个字符串数组，每一个元素存放项目列表其中的一个项目。List 数组的下标从 0 开始，例如输出列表框 List1 的第 2 个项目，则可以写为：

```
Print List1.List(1)
```

向列表框中添加项目有两种方法。第一种方法是在属性窗口中选中列表框的 List 属性，单击下拉按钮，输入一个项目后按下 Ctrl+Enter 组合键，在下一行继续输入新项目。第二种方法是在程序中调用 AddItem 方法，在列表框中添加项目。

（3）在程序中 ListIndex 和 ListCount 往往与 List 属性配合使用。如果用户未选择任何项目，ListIndex 的值是-1；如果用户选中项目列表中的第一项，ListIndex 的值是 0；如果用户选中项目列表中的最后一项，则 ListIndex 的值是 ListCount-1。

（4）Selected 的属性值是一个逻辑型数组，其每一个元素与项目列表中的每一个项目一一对应。如果某个项目被用户选中，Selected 数组相应元素的值是 True；如果未被选中，则相应元素的值是 False。例如 List1.Selected(2)的值是 True，表示列表框 List1 的第 3 个项目被选中。

（5）MultiSelect 的属性值有 3 个，默认值是 0，如表 6-3 所示。

表 6-3　MultiSelect 属性值

常量	值	含义
None	0	不允许多选
Simple	1	简单多选，可以用鼠标单击或按空格键进行选择
Extended	2	扩展多选，可以借助 Shift 键或 Ctrl 键进行选择

2. 事件

列表框控件能够响应 Click 和 DblClick 等事件。在实际编程中经常针对列表框编写

DblClick 事件过程，使得双击列表框中的某个选项之后，可以对该选项进行相应的操作。例如在“文件”对话框的文件列表框中双击某个文件名，即可直接打开该文件。

3. *方法*

列表框的常用方法如表 6-4 所示。

表 6-4　列表框的常用方法

方法	功能
AddItem	向列表框中添加一个项目
RemoveItem	从列表框中删除一个项目
Clear	清除列表框中所有项目

说明：

（1）AddItem 方法的调用形式如下：

```
对象.AddItem Item[,Index]
```

参数 Item 表示被添加到列表框中的字符串，即新项目。参数 Index 表示新项目在列表框中的索引值，即插入到项目列表中的位置。如果该参数被省略，则将把新项目插入到项目列表的末尾。

（2）RemoveItem 方法的调用形式如下：

```
对象. RemoveItem Index
```

参数 Index 表示被删除的项目在列表框中的索引值。例如删除列表框 List1 中的第一个项目，可以写为：

```
List1.RemoveItem 0
```

6.8　组合框

组合框（ComboBox）控件组合了文本框和列表框的特性，用户既可以在它的文本框部分输入文本以选择项目，也可以在它的列表框部分选择项目。当用户在列表框部分选定某个项目之后，该项目会自动出现在文本框部分。列表框将用户的选择限制在项目列表之内，而组合框则允许用户选择项目列表中所没有的项目。在 VB 的工具箱中，组合框控件的图标如图 6-13 所示。

图 6-13　组合框图标

1. *属性*

组合框控件的大部分属性与列表框控件相同，此外还有一些与文本框相同的属性。表 6-5 列出了组合框控件的常用属性。

表 6-5　组合框的常用属性

属性	作用
Name	设置组合框的对象名
Text	确定用户当前选择的项目或者在文本框部分输入的项目
List	设置组合框所显示的项目列表
ListCount	确定组合框中项目的总数

续表

属性	作用
ListIndex	确定当前选中的项目在项目列表中的索引值
Selected	确定项目列表中某个项目是否被选中
Style	设置组合框的类型

说明：

（1）程序第一个组合框控件的默认对象名是 Combo1，第二个组合框控件的默认对象名是 Combo2，依此类推。

（2）Style 的属性值有 3 个，默认值是 0，如表 6-6 所示。

表 6-6　Style 属性值

常量	值	含义
Dropdown Combo	0	下拉式组合框
Simple Combo	1	简单组合框
Dropdown List	2	下拉式列表框

下拉式组合框如图 6-14 所示。它将文本框和下拉式列表框组合在一起，用户可以直接用键盘在文本框中输入项目，也可以单击下拉按钮，打开列表框进行选择。

简单组合框如图 6-15 所示。它将文本框和列表框简单地组合在一起，列表框的项目列表直接显示在窗体上。

下拉式列表框如图 6-16 所示。它的功能与下拉式组合框相似，但是用户只能从列表框中进行选择，而不能直接在文本框中输入项目。

图 6-14　下拉式组合框

图 6-15　简单组合框

图 6-16　下拉式列表框

思考：当组合框为下拉式列表框类型时，用户能否选择项目列表中所没有的项目？

2. 事件

根据类型的不同，组合框控件能够响应的事件也有所不同。所有类型的组合框都能够响应 Click 事件，但是只有简单组合框（Style 的属性值为 1）才能响应 DblClick 事件。此外下拉式组合框和简单组合框还可以响应 Change 事件。

3. 方法

AddItem、RemoveItem 和 Clear 等方法也同样适用于组合框控件。例如在组合框 Combo1 中添加一个项目“土耳其”，可以写为：

```
Combo1.AddItem "土耳其"
```

例如清空组合框 Combo1 中的所有项目，可以写为：

```
Combo1.Clear
```

【例 6.11】用列表框和组合框改写例 4.6 的程序。

分析：在界面设计阶段分别创建一个组合框控件和两个列表框控件。组合框 Combo1 组织的项目列表供学生选择系别，列表框 List1 组织的项目列表供学生选择爱好，列表框 List2 则存放已经选中的爱好。列表框 List1 的项目列表是在属性窗口中针对 List 属性一一添加的，组合框 Combo1 的项目列表则是在窗体 Load 事件过程中调用 AddItem 方法添加的。

在程序中分别为列表框 List1 和 List2 编写 DblClick 事件过程。在 List1 的 DblClick 事件过程中，先通过 Text 属性读取用户当前选择的项目，然后调用 AddItem 方法把它添加到 List2 的项目列表中。在 List2 的 DblClick 事件过程中，通过 ListIndex 属性获取用户当前选项的索引值，调用 RemoveItem 方法把该项目删除。

在命令按钮 Command1 的 Click 事件过程中，通过组合框 Combo1 的 Text 属性获取用户选择的系，在循环结构中访问列表框 List2 的 List 数组中的元素，获取所有的选项，并做相应的处理。

```
Private Sub Form_Load()
Combo1.AddItem "计算机"
Combo1.AddItem "汽车"
Combo1.AddItem "机械"
Combo1.AddItem "管理"
End Sub
Private Sub Command1_Click()
Dim s As String, i As Integer
s = s + "姓名：" + Text1.Text + vbCr
s = s + "年龄：" + Text2.Text + vbCr
s = s + Combo1.Text  '得到学生所在的系
s = s + "系" + vbCr
s = s + "爱好："
For i = 0 To List2.ListCount - 1 '得到学生所有的爱好
 s = s + List2.List(i) + Space(2)
Next i
MsgBox (s)
End Sub
Private Sub Command2_Click()
End
End Sub
Private Sub List1_DblClick()
Dim s As String
s = List1.Text '从 List1 得到选中的项目
List2.AddItem s '将该项目添加到 List2 中
End Sub
Private Sub List2_DblClick()
List2.RemoveItem List2.ListIndex '删除选中的项目
End Sub
```

运行程序，结果如图 6-17 所示。

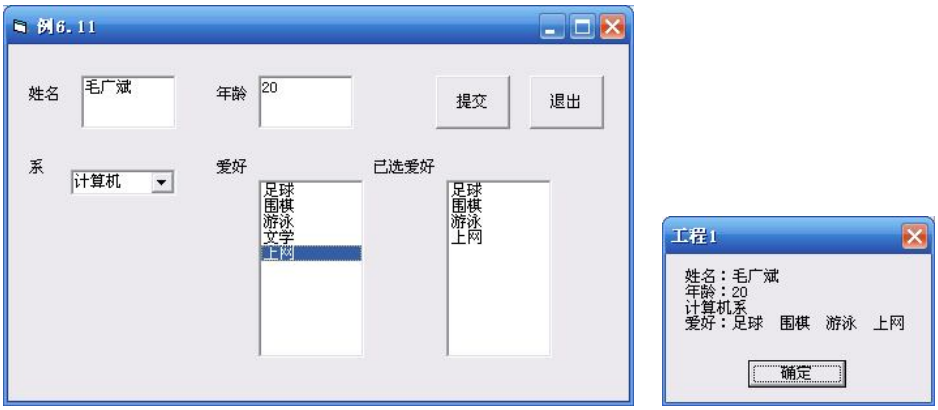

图 6-17　例 6.11 的运行结果

说明：程序运行时，用户可以在组合框的列表框部分选择学生所在的系，也可以在文本框部分输入项目列表中未列出的系。如果用户在“爱好”列表框中双击一个项目，则“已选爱好”列表框中将自动出现该项目，表示用户选中了某个爱好。如果用户在“已选爱好”列表框中双击一个项目，则该项目将自动消失，表示用户放弃了对某个爱好的选择。

该程序有许多需要完善的地方。例如用户在“爱好”列表框中反复双击同一个项目，则该项目就会多次出现在“已选爱好”列表框中，造成对爱好的重复选择。解决方法是在向“已选爱好”列表框中添加项目之前，先判断其中是否已存在该项目。如果是新项目就可以添加，否则不予添加。部分代码如下：

```
Private Sub List1_DblClick()
Dim s As String, flag As Boolean, i%
s = List1.Text '从 List1 得到选中的项目
flag = False
For i = 0 To List2.ListCount - 1
 If s = List2.List(i) Then '在 List2 中查找该项目
  flag = True
  Exit For
 End If
Next i
If Not flag Then '所选项目是以前未选过的项目
 List2.AddItem s '将该项目添加到 List2 中
End If
End Sub
```

思考：如果只安排一个“爱好”列表框，而没有“已选爱好”列表框，并且把列表框控件的 MultiSelect 属性值设置为 1，即允许多选。此时应如何编写程序，使得可以显示用户在“爱好”列表框中选择的多个项目？

6.9　程序举例

【例 6.12】采用选择排序法对 n 个整数按升序排序。

分析：在前面的章节中多次提到并利用了选择排序法，现在彻底阐明以及实现这个算法。

选择排序算法的基本思想是，在每一轮里把该轮第一个数与后面的数依次比较，如果第一个数大于后面的数，则两者进行交换。始终确保在该轮数列中，第一个数是当前最小数，直至所有的数按升序排列。

例如有数列[5,3,4,2,1]，第一轮 5 首先与 3 比较，显然 5>3，于是 5 与 3 交换，此时数列变为[3,5,4,2,1]；然后 3 与 4 比较，数列不变；3 再与 2 比较，发生交换，数列变为[2,5,4,3,1]；该轮最后一次是 2 与 1 比较，发生交换，数列变为[1,5,4,3,2]。经过第一轮排序，确保了最小数在数列的第一个位置。

第二轮排序数列缩小为[5,4,3,2]，5 与 4 交换，数列变为[4,5,3,2]；然后 4 与 3 交换，数列变为[3,5,4,2]；3 再与 2 交换，数列变为[2,5,4,3]。经过第二轮排序，确保了次小数在数列的第二个位置，整个数列变为[1,2,5,4,3]。

第三轮排序数列进一步缩小为[5,4,3]，经过比较和交换后，变为[3,5,4]，整个数列变为[1,2,3,5,4]。第四轮排序数列缩小为[5,4]，经过比较和交换后，变为[4,5]，整个数列变为[1,2,3,4,5]，排序结束。每一轮都是当前排序数列的第一个数依次与后面的数比较和交换，始终确保当前最小数在其正确的位置上。如此循环往复，最终数列成为按升序排列的有序数列。

编程实现时，需要定义一个动态整型数组 a 存放数列。用二重循环来排序，外层控制排序的轮数，内层控制每轮的次数。循环结构编制的关键在于排序的轮和次如何确定，n 个数的排序显然要进行 n-1 轮，因为经过 n-1 轮排序后，依次确定了从最小数到次大数的位置，而剩下的最后一个数自然就是最大数。

那么每一轮中到底比较多少次呢？第一轮由于是第一个数与后面 n-1 个数比较，显然应该比较 n-1 次；第二轮是第二个数与后面 n-2 个数比较，应该比较 n-2 次；经过归纳可以得出，第 k 轮应该比较 n-k 次。这样可以用循环变量 i 控制外层循环，i 的初值是 1，一直到 n-1；j 则从 i+1 开始，一直到 n。在循环体中利用 If 语句和中间变量，完成 a(i)与 a(j)的比较和交换。

```
Private Sub Command1_Click()
Dim a() As Integer, n As Integer, i%, j%, t%
n = Val(Text1.Text)
ReDim a(1 To n)
For i = 1 To n
 a(i) = Val(InputBox("请输入第" & i & "个数"))
Next i
Picture1.Print "输出原数列"
j = 0
For i = 1 To n      '输出原数列
 Picture1.Print Tab(j * 6); a(i);
 j = j + 1
 If i Mod 5 = 0 Then
  Picture1.Print
  j = 0
 End If
Next i
For i = 1 To n - 1
 For j = i + 1 To n
  If a(i) > a(j) Then
```

```
    t = a(i)        'a(i)与a(j)交换
    a(i) = a(j)
    a(j) = t
   End If
  Next j
 Next i
 Picture1.Print "输出排序之后的数列"
 j = 0
 For i = 1 To n     '输出排序之后的数列
  Picture1.Print Tab(j * 6); a(i);
  j = j + 1
  If i Mod 5 = 0 Then
   Picture1.Print
   j = 0
  End If
 Next i
 End Sub
 Private Sub Command2_Click()
 End
 End Sub
```

运行程序，结果如图 6-18 所示。

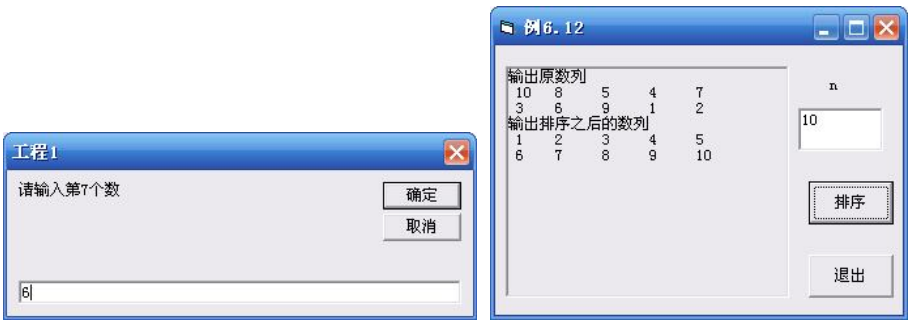

图 6-18　例 6.12 的运行结果

说明：从提高程序的运行效率来看，每一轮中第一个数可以先不与后面的数交换，而是定义一个标志 min 用来记录当前最小数的下标。在该轮结束后，第一个数再与该轮最小的数交换。于是循环语句变为：

```
For i = 1 To n - 1
min=i
 For j = i + 1 To n
  If a(min) > a(j) Then
min=j        '记录当前最小数的下标
  End If
Next j
If i<>min Then  '如果a(i)就是当前最小数，则不必交换
t = a(i)        'a(i)与a(min)交换
  a(i) = a(min)
  a(min) = t
```

```
 End If
Next i
```

思考：外层 For-Next 循环改为 For i = 2 To n 可以吗？

【例 6.13】在一个已排好序（假定为升序）的数列中查找某个数据。

分析：在本例中，可以采取效率较高的折半查找算法。其基本思想是，每次选中有序数列里中间位置的数据 d，与所查数 m 进行比较。如果两者相等，则查找成功；如果 d 大于 m，则在数列的左半区（第一个数到 d）查找；如果 d 小于 m，则在数列的右半区（d 到最后一个数）查找。如此循环往复，直到查找结束。由于每次查找范围都较上次缩小一半，因此查找速度较快。

例如有数列：[-5,-2,1,5,10,21,37,49,82]，要查找的数是 37。首先选定中间数 10，由于 37>10，因此下次查找范围应该在右半区即[21,37,49,82]。需要指出的是，本次的中间位置的数可以从下次的查找范围中排除。在数列[21,37,49,82]中选定中间数 37，经过比较，判定查找成功。如果查找范围不断缩小，最后仍未找到，则判定查找失败。

编程实现时，首先应定义一个动态整型数组 a 用来存放数列。其次应定义 3 个指示器 low、mid 和 high，用来指示查找范围。其中 low 指示查找范围的第一个数，high 指示查找范围的最后一个数，mid 指示查找范围的中间数，mid=(low+high)/2。在循环结构中判断要查找的数 m 与 mid 所指示的中间数 a(mid)是否相等，如果相等则查找成功；如果 m 小于 a(mid)，说明查找范围缩小为左半区，则 low 不变，high 变为 mid-1；如果 m 大于 a(mid)，说明查找范围缩小为右半区，则 low 变为 mid+1，high 不变；如果 low>high，则说明查找范围失效，查找失败。

```
Private Sub Command1_Click()
Dim a() As Integer, i%, j%, m%, n%
Dim flag As Boolean, low%, high%, mid%
n = Val(Text1.Text)
ReDim a(1 To n)
For i = 1 To n
 a(i) = Val(InputBox("请输入第" & i & "个数"))
Next i
Picture1.Print "输出数列"
j = 0
For i = 1 To n      '输出数列
 Picture1.Print Tab(j * 6); a(i);
 j = j + 1
 If i Mod 5 = 0 Then
  Picture1.Print
  j = 0
 End If
Next i
Picture1.Print
Do
 m = Val(InputBox("请输入要查询的数 m"))
 low = 1
 high = n
 flag = False
```

```
    Do While low <= high
     mid = (low + high)/2          '计算中间数的位置
     If a(mid) = m Then            '中间数与 m 比较
      flag = True                  '查找成功
      Exit Do
     ElseIf a(mid) > m Then
      high = mid - 1               '查找范围缩小为左半区
     Else
      low = mid + 1                '查找范围缩小为右半区
     End If
    Loop
    If flag = True Then            '判断查找标志
     Picture1.Print "找到"; m
    Else
     Picture1.Print "没有找到"; m
    End If
    j = MsgBox("还要继续查询吗？", vbYesNo + vbquestin)
  Loop While j = 6
  End Sub
  Private Sub Command2_Click()
  End
  End Sub
```

运行程序，结果如图 6-19 所示。

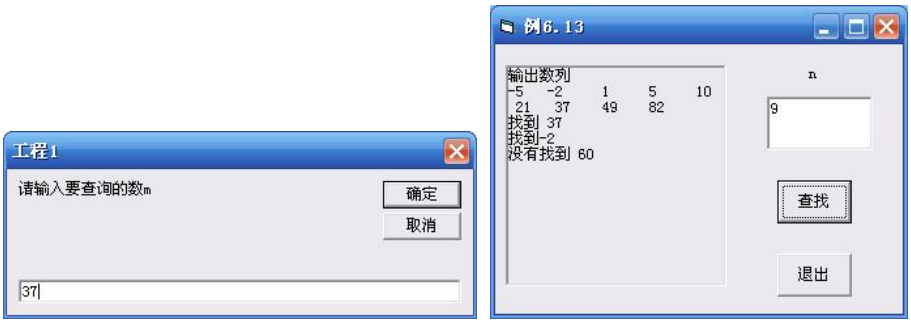

图 6-19　例 6.13 的运行结果

【例 6.14】判断用户输入的文本是否为回文。如果一个文本的逆序与原文完全相同，这样的文本就称为回文，例如“level”、“2002”和“我是我”等。

分析：定义一个字符串变量 s，存放输入的文本。其次定义 2 个指示器 left 和 right，left 初始指示文本的第一个字符，right 初始指示文本的最后一个字符。在循环结构中反复判断 left 和 right 各自指示的字符是否相同，如果不同，显然不是回文；如果相同，则 left 不断加 1 向右移动，而 right 不断减 1 向左移动。

```
Private Sub Command1_Click()
Dim s As String, left%, right%, flag As Boolean
s = Text1.Text
left = 1
```

```
right = Len(s)
flag = True
Do While left < right
 If mid(s, left, 1) <> mid(s, right, 1) Then
  flag = False
  Exit Do
 End If
 left = left + 1
 right = right - 1
Loop
If flag = True Then
 s = s + "是回文"
Else
 s = s + "不是回文"
End If
Picture1.Print s
End Sub
Private Sub Command2_Click()
End
End Sub
```

运行程序，结果如图 6-20 所示。

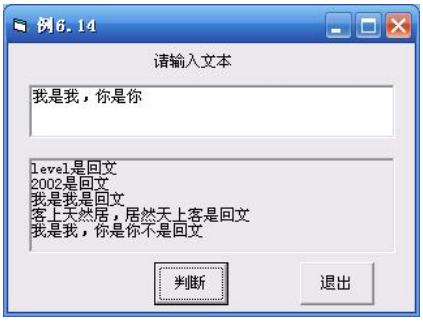

图 6-20　例 6.14 的运行结果

说明：将在第 7 章给出判断回文的递归解法。

【例 6.15】Josephus 环问题。n 个孩子围成一圈，任意假定一个数 m。从第一个孩子开始，以顺时针方向数，每数到第 m 个孩子时，他就离开这个圈子。孩子不断离开，圈子不断缩小，最后剩下的一个孩子便是胜利者。请计算出胜利者的序号。

分析：显然要定义一个动态整型数组 a，存放这些孩子的序号，数组的长度就是孩子的个数 n。该问题有两个难点需要解决。首先孩子是围成一个圈的，而数组元素是顺序排列的，表达了一种线性的逻辑结构。如何在数到数组的末尾时，跳到数组的头部继续往下数？这可以通过对元素下标求余加 1 的技巧解决，当数到数组尾部时，下一个数组元素的下标通过先对 n 求余再加 1，算得为 1，从而回到数组首部继续往下数。

其次是如何表示孩子离开圈子？当孩子离开圈子时，将数组 a 中与其对应的元素值（序号）置为 0，表示这个孩子已离开圈子。这样继续数时如果发现这个孩子对应的数组元素值为 0，

说明他已出圈，就跳过此人不数。

```
Private Sub Command1_Click()
Dim a() As Integer, i%, j%, k%, m%, n%
n = Val(Text1.Text)
ReDim a(1 To n)
For i = 1 To n
 a(i) = i '设置小孩序号
Next i
m = Val(InputBox("请输入 m"))
j = 0
For i = 1 To n - 1  'n-1 个孩子出圈
 k = 1      '从 1 开始数
 Do While k <= m '数 m 个数
  j = j Mod n + 1 '下标后移
  If a(j) <> 0 Then
   k = k + 1
  End If
 Loop
 Picture1.Print a(j); "号孩子出圈"        '显示出圈的孩子
 a(j) = 0 '孩子出圈
Next i
For i = 1 To n
 If a(i) <> 0 Then '找到胜利者
  Exit For
 End If
Next i
Picture1.Print "胜利者是"; a(i); "号孩子"  '显示胜利者
End Sub
Private Sub Command2_Click()
End
End Sub
```

运行程序，结果如图 6-21 所示。

图 6-21　例 6.15 的运行结果

说明：n-1 个孩子出圈之后，在数组 a 中只有一个元素的值不为 0，该元素就对应于胜利者，元素的值则为胜利者的序号。在循环语句中不断地判断数组 a 元素的值是否不为 0，即可

找到胜利者。

【例 6.16】编写程序，输出 n 行杨辉三角形。

分析：杨辉三角形中的各个元素实际上是二项式$(a+b)^n$的展开式中各项的系数，例如 6 行杨辉三角形如下所示：

```
1
1    1
1    2    1
1    3    3    1
1    4    6    4    1
1    5    10   10   5    1
```

从中可以发现，杨辉三角形每一行元素的数量比上一行增加 1，各行的第一列和对角线上的元素值均为 1，而且其余各项的值都是其上一行前一列元素的值与上一行同一列元素的值之和。

编程实现时显然要定义一个动态整型数组 a，以存放杨辉三角形的各个元素。根据用户输入的 n 值，设置 a 为 n 行 n 列的二维数组。注意到对角线上元素的行号和列号相等，首先利用一重循环将数组各行的第一列及对角线上的元素的值置为 1，然后利用二重循环，借助于本行元素与上一行元素之间的规律，计算出所有元素的值，最后再次利用二重循环，输出杨辉三角形所有元素的值。

```
Private Sub Command1_Click()
Dim a() As Integer, n%, i%, j%, k%
n = Val(Text1.Text)
ReDim a(1 To n, 1 To n)
For i = 1 To n
 a(i, 1) = 1 '第一列元素置为 1
 a(i, i) = 1 '对角线元素置为 1
Next i
For i = 3 To n    '从第三行开始
 For j = 2 To i - 1 '从第二列开始，到对角线为止
  a(i, j) = a(i - 1, j - 1) + a(i - 1, j)
 Next j
Next i
For i = 1 To n
 k = 0
 For j = 1 To i
  Picture1.Print Tab(k * 6); a(i, j);
  k = k + 1
 Next j
 Picture1.Print
Next i
End Sub
Private Sub Command2_Click()
End
End Sub
```

运行程序，结果如图 6-22 所示。

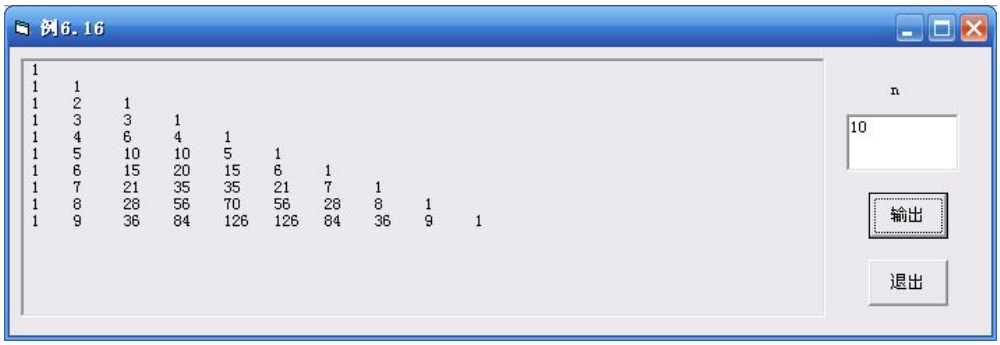

图 6-22　例 6.16 的运行结果

6.10　小结

本章主要讲解了一维数组、二维数组、动态数组以及字符串的处理方法。数组是同类型相关数据的集合，它由一些元素组成。每一个元素可以存放一个数据，下标表示元素在数组中的相对位置。通过循环结构可以对数组的元素进行统一的输入、输出和处理，其中循环控制变量控制元素的下标从下界移动到上界。一般说来，对一维数组的处理应采用一重循环，对二维数组的处理应采用二重循环。

动态数组的维数和长度可以用 ReDim 语句重新进行设置，但是数组的类型不能改变。如果重新设置动态数组时使用了关键字 Preserve，则可以保留元素的值。记录类型是程序员自定义的一种类型，它由一些基本类型的成员所组成，定义记录类型的关键字是 Type。字符串中的各个字符之间存在着较为明显的位置关系，其处理方法与数组有一些相似之处。VB 语言提供了很多字符串处理函数，常用的有 Len 函数、InStr 函数、Replace 函数和 Mid 函数等，分别用于实现字符串的统计长度、查找、替换和截取等操作。

控件数组能够统一地组织和管理一批同属一类的控件，它们共用一个对象名。通过控件对象的 Index 属性，可以实现对控件数组中任意一个控件元素的访问。列表框控件以项目列表的形式显示数据，可以供用户进行选择，实现数据输入的标准化。组合框控件同时具有文本框和列表框的一些特性，它在界面上显示时比列表框节省空间。组合框不仅可以接受标准化数据，也可以接受用户在文本框部分输入的数据，具有很好的灵活性。列表框控件和组合框控件重要的属性是 List、ListIndex、Text 和 Selected，重要的方法是 AddItem、RemoveItem 和 Clear。

此外，在案例程序中还介绍了选择排序和折半查找等常用算法，并对这些算法的原理以及实现方法做了详细的分析。

习题

1．有数组定义语句 Dim a(7,-2 To 3) As Integer，数组 a 有多少个元素？

2．写出单击窗体后，下列程序段的运行结果。

```
Private Sub Form_Click()
```

```
Dim a(3, 5) As Integer, i As Integer, j As Integer
For i = 1 To 3
 For j = 1 To 5
  a(i, j) = a(i - 1, j - 1) + i + j
 Next j
Next i
Print a(3, 4)
End Sub
```

3．将一个长度为 10 的数列头尾颠倒，例如该数列原先为[1,2,3,4,5,6,7,8,9,10]，处理后变为[10,9,8,7,6,5,4,3,2,1]。

4．编写一个模拟掷骰子的程序，要求统计掷 100 次后骰子上各点出现的次数。

5．计算一个 4×4 矩阵的两个对角线之和。

6．完成一个 3×3 矩阵的转置（即行列互换）。

7．输入一个 3×3 矩阵各元素的值，找出每一行最大的数。

8．将一个字符串翻转，例如把字符串“abcd”翻转为“dcba”。

9．把两个已按升序排列的数列合并为一个新数列，该数列仍按升序排列。例如数列 a 是[1,3,6,7,9]，数列 b 是[2,4,5,8,10]，合并之后的新数列是[1,2,3,4,5,6,7,8,9,10]。

10．把一个数插入到一个已按升序排列的数列 a 中，并使该数列仍按升序排列。例如数列 a 是[1,3,6,8,9,10]，要插入的数是 4，合并之后的新数列是[1,3,4,6,8,9,10]。

11．输入一个句子，找出其中最长的单词。

12．找出一个 4×4 矩阵中的“鞍点”。所谓鞍点是指它在本行中的值最大，在本列中的值最小。输出鞍点的行号和列号，如果找不到鞍点，则输出“no found”。

13．某班有学生 30 人，学习语文、数学和英语 3 门课程。输入所有学生各门课程的成绩，输出单科成绩的最高分以及该班每门课的平均成绩。

14．输出魔方阵。所谓魔方阵，是指由自然数 1～n^2（n 为奇数）组成的方阵，其各行、各列以及对角线元素之和均相等。

15．某比赛有 10 位评委给参赛选手打分，满分为 10 分。选手综合得分的计算规则是：去掉一个最高分，再去掉一个最低分，然后计算出该选手的平均得分。编写程序，分别输入 10 位评委给 5 位参赛选手打的分数，然后计算出各位选手的综合得分，并输出其中的最高综合得分。