第3章

复合数据类型

学习目标

通过本章的学习，能够学会：

(1) 数组的定义及使用；

(2) 结构的定义及使用；

(3) 集合的定义及使用；

(4) 枚举的定义及使用。

在第2章中介绍了存储单一信息的基本数据类型，如整型、字符型、浮点型、日期型等，而在实际应用中，有些数据是由若干相关数据构成的，这些数据无法用基本数据类型来存储。Visual Basic.NET 提供了可以存储复杂数据的几种复合数据类型，其中包括数组、结构、集合和枚举。

3.1 数组

数组是同类变量的有序集合。数组中的变量称为数组元素，数组元素具有相同的名字(数组名)和数据类型，通过数组下标(索引)来标识并区分它们。数组可以是一维或多维的。

3.1.1 一维数组

1. 数组的声明

数组必须先声明后使用。声明数组就是让系统在内存中分配一个连续的区域，用来存储数组元素。

一维数组是指只有一个下标的数组。声明一维数组的一般格式如下：

【格式】

```
Dim 数组名(下标上界) As 数据类型
```

【说明】

(1) 声明一个名为“数组名”的数组，该数组的下标上界由“下标上界”指定，数组元素的数据类型由“数据类型”确定。

(2) “数组名”为任何合法变量名。

(3) 数组元素的下标上界只能是常数，不能是变量或表达式。

(4) 数组元素的下标下界为 0，不能改变。

(5) “数据类型”可以是基本数据类型，也可以是 Object 类型。

例如：

```
Dim a(10) As Integer      '11 个元素，从 a(0)到 a(10)
```

2. 数组的初始化

在使用数组时，通常要求数组元素有初始值。在定义数组时指定数组元素的初始值，称为数组的初始化。一维数组的初始化一般格式如下：

【格式】

```
Dim 数组名() As 数据类型={ 值 1,值 2,…,值 n }
```

【说明】

不允许对指定了上界的数组进行初始化，因此“数组名”后的括号必须为空，系统将根据初始值的个数确定数组的上界。

例如：

```
Dim a() As Integer={1, 3, 5, 7, 9}
```

声明了一个 Integer 类型的一维数组 a，该数组有 5 个初值，所以数组的上界为 4。经过上述的定义和初始化后，各数组元素的值依次为：

```
a(0)=1
a(1)=3
a(2)=5
a(3)=7
a(4)=9
```

又如：

```
Dim a() As String={"金", "木", "水", "火", "土"}
```

声明了一个 String 类型的一维数组 a，该数组有 5 个初值，所以数组的上界为 4。经过上述的定义和初始化后，各数组元素的值依次为：

```
a(0)="金"
a(1)="木"
a(2)="水"
a(3)="火"
a(4)="土"
```

3. 数组元素的使用

数组变量声明后，就可以像使用普通变量一样来使用各数组元素了。一维数组元素的引用格式如下：

【格式】

```
数组名(下标)
```

【说明】

下标可以是整型常量或表达式。

例如：

```
Dim a(10) As Integer
a(1)=5
a(2)=2*a(1)+3
```

声明了一个数组就相当于声明了一组变量，使用数组的好处就是可以利用循环语句对数组元素进行依次访问。

例如，使用循环语句就可以很方便地给数组元素赋值：

```
Dim a(100), i As Integer
For i=0 To 100
  a(i)=i
Next
```

注意：通常数组中所有元素的值类型应与数组定义的类型相同，但是如果数组的类型是 Object，则可以在数组中混合使用各种数据类型。

例如：

```
Dim a(2) As Object
a(0)=10                  '整型
a(1)="计算机"            '字符串
a(2)=#10/25/2005#        '日期型
```

【实例 3.1】 声明一个大小为 10 的整型数组。要求：

(1) 倒序输出每个数组元素的值；

(2) 输出数组元素的和；

(3) 输出数组元素的平均值。

程序代码：

```
Module Module1
    Sub Main()
        Dim a() As Integer={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
        Dim i As Integer
        Dim sum As Double=0
        Dim aver As Double
```

```
        '倒序输出每个数组元素的值
        For i=9 To 0 Step-1
            Console.Write(CStr(a(i))+" ")
        Next
        Console.WriteLine()
        '求数组元素的和
        For i=0 To 9
            sum+=a(i)
        Next
        '输出数组元素的和
        Console.WriteLine("和为："+CStr(sum))
        aver=sum / 10  '求数组元素的平均值
        '输出数组元素的平均值
        Console.WriteLine("平均值为："+CStr(aver))
        Console.ReadLine()
    End Sub
End Module
```

运行结果如图 3.1 所示。

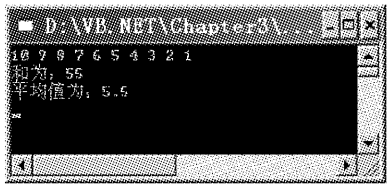

图 3.1　实例 3.1 的运行结果

3.1.2　二维数组

1. 二维数组的声明

二维数组是指具有两个下标的数组。声明二维数组的一般格式如下：

【格式】

Dim 数组名(下标 1 上界,下标 2 上界) As 数据类型

【说明】

a(0,0)	a(0,1)	a(0,2)	a(0,3)
a(1,0)	a(1,1)	a(1,2)	a(1,3)
a(2,0)	a(2,1)	a(2,2)	a(2,3)

图 3.2　二维数组示例

数组元素的下标下界均为 0,不能改变。二维数组可以看作是一个行列矩阵,例如：

```
Dim a(2, 3) As Integer
```

数组元素的排列如图 3.2 所示。

2. 二维数组的初始化

二维数组的初始化一般格式如下：

【格式】

Dim 数组名(,) As 数据类型={ {第 1 行值},{第 2 行值},…,{第 n 行值} }

【说明】

(1)“数组名”后的括号内必须有一个逗号,系统将据此确定数组是二维数组。

(2) 内层花括号的对数确定二维数组的行数，内层花括号中值的个数决定了二维数组的列数。

例如：

```
Dim a(,) As Integer={{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}}
```

声明了一个 Integer 类型的二维数组 a，经过上述的定义和初始化后，数组为 3 行 4 列，各数组元素的值依次为：

a(0,0)＝1　a(0,1)＝2　a(0,2)＝3　a(0,3)＝4

a(1,0)＝5　a(1,1)＝6　a(1,2)＝7　a(1,3)＝8

a(2,0)＝9　a(2,1)＝10　a(2,2)＝11　a(2,3)＝12

3. 二维数组元素的使用

二维数组变量声明后，就可以像使用普通变量一样来使用各数组元素了。二维数组元素的引用格式如下：

【格式】

```
数组名(下标,下标)
```

【说明】

下标可以是整型常量或表达式。

例如：

```
Dim a(2, 3) As Integer
a(0, 0)=1
a(0, 1)=a(0, 0)+2
```

二维数组元素的访问可以通过二重 For 循环来实现。第一重 For 循环来控制访问行，第二重 For 循环来控制访问列。

例如：

```
Dim a(2, 3), i, j As Integer
For i=0 To 2
   For j=0 To 3
      a(i, j)=i+j
   Next j
Next i
```

【实例 3.2】 通过二维数组来存储一个 4×4 矩阵，并求矩阵对角线上元素之和。

程序代码：

```
Module Module1
    Sub Main()
        Dim a(3, 3), i, j As Integer
        Dim sum As Integer=0
        '给数组元素赋值
```

```
        For i=0 To 3
            For j=0 To 3
                a(i, j)=i+j
            Next j
        Next i
        '求矩阵对角线上元素之和
        For i=0 To 3
            For j=0 To 3
                '行标和列标相等说明是对角线上元素
                If i=j Then
                    sum+=a(i, j)
                End If
            Next j
        Next i
        Console.WriteLine("矩阵为：")
        '输出矩阵元素
        For i=0 To 3
            For j=0 To 3
                Console.Write(CStr(a(i, j))+" ")
            Next j
            Console.WriteLine()
        Next i
        Console.WriteLine("对角线上元素之和为："+CStr(sum))
        Console.ReadLine()
    End Sub
End Module
```

运行结果如图 3.3 所示。

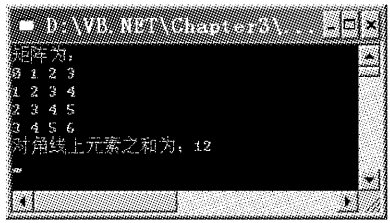

图 3.3　实例 3.2 的运行结果

3.1.3　复制数组

Visual Basic.NET 支持数组间赋值，可以将一个数组赋值给另一个数组。数组类似于对象，使用其 Clone 方法可以实现复制数组。

【格式】

```
数组名.Clone
```

【说明】

Clone 方法仅复制数组元素的值（无论它是引用类型还是值类型），返回数组的浅表副本，即仅复制数组元素的值；如果值是引用类型，不复制这些引用所引用的对象。

【实例 3.3】　复制数组。

程序代码：

```
Module Module1
```

```
    Sub Main()
        Dim a() As Integer={1, 2, 3, 4}
        Dim i As Integer
        Dim b() As Integer
        b=a.Clone()
        Console.WriteLine("b 数组：")
        For i=0 To 3
            Console.Write(" "+CStr(b(i)))
        Next
        Console.WriteLine()
        b(0)=5
        Console.WriteLine("修改后的 b 数组：")
        For i=0 To 3
            Console.Write(" "+CStr(b(i)))
        Next
        Console.WriteLine()
        Console.WriteLine("a 数组：")
        For i=0 To 3
            Console.Write(" "+CStr(a(i)))
        Next
        Console.ReadLine()
    End Sub
End Module
```

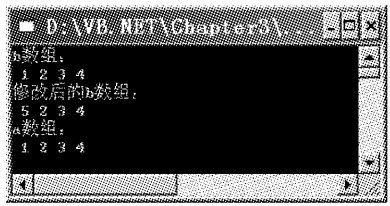

图 3.4　实例 3.3 的运行结果

运行结果如图 3.4 所示。

3.1.4　动态数组和 Ubound()函数

在 Visual Basic.NET 中，所有数组都是变长的，也就是说所有数组都是动态数组。在声明一个数组时，可以指定长度也可以不指定长度，指定的长度也可以使用 ReDim 语句来改变。ReDim 语句的格式如下：

【格式】

```
ReDim [Preserve] 数组名(下标上界)
```

【说明】

(1) ReDim 语句用于重新分配数组变量的存储空间。

(2) 参数 Preserve 为可选项，用来保留现有数组中的数据。

(3)“下标上界”表示重定义数组维度的上界。如果数组是多维的，以逗号分隔多个下标上界。

例如：

```
Dim a() As Integer          '声明一个一维动态数组
ReDim a(5)                  '分配 6 个元素空间
ReDim Preserve a(8)         '分配 9 个元素空间,保留原有元素
```

Ubound()函数可以用于求解数组某维的上界。

【格式】

```
Ubound( 数组名[,维数])
```

【说明】

维数为1(默认)表示第1维,2表示第2维,以此类推。

例如:

```
Dim a(5) As Integer
Dim b(3, 9) As Integer
Dim ua, ub1, ub2 As Integer
ua=UBound(a)              '获取 a 数组上界
ub1=UBound(b, 1)          '获取 b 数组第 1 维上界
ub2=UBound(b, 2)          '获取 b 数组第 2 维上界
```

【实例 3.4】 求斐波那契(Fibonacci)数列的前 n 项,n 从键盘输入。斐波那契数列:1,1,2,3,5,8,…。满足如下条件:$F_1=1$,$F_2=1$,$F_n=F_{n-1}+F_{n-2}$。

程序代码:

```
Module Module1
    Sub Main()
        Dim i, f() As Integer
        Dim n As Integer
        Console.Write("请输入 n: ")
        n=CInt(Trim(Console.ReadLine()))
        If n>0 Then
            ReDim f(n)   '重新分配数组变量的存储空间
            f(0)=1:f(1)=1
            For i=2 To n-1
                f(i)=f(i-1)+f(i-2)
            Next
            For i=0 To n-1
                Console.Write(" "+CStr(f(i)))
                If i Mod 5=0 And i<>0 Then
                    Console.WriteLine()  '输出 5 个数一换行
                End If
            Next
        End If
        Console.ReadLine()
    End Sub
End Module
```

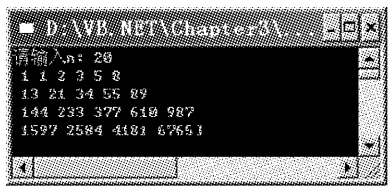

图 3.5 实例 3.4 的运行结果

运行结果如图 3.5 所示。

3.2 结构

结构是一种较为复杂但非常灵活的复合数据类型，一个结构类型可以由若干个称为成员（或域）的成分组成。根据需要，不同的结构类型可以包含不同的成员。在程序设计中，结构有着重要的作用。

3.2.1 结构类型与变量的定义

1. 定义结构类型

【格式】

```
[Public|Friend|Private|Dim] Structure 结构名
    变量声明
    [过程声明]
End Structure
```

【说明】

（1）在模块或类级使用，以声明结构和定义该结构成员的特征。

（2）Public 或 Dim：用 Public 关键字声明的结构具有公共访问权限。公共结构的可访问性没有任何限制。

（3）Friend：用 Friend 关键字声明的结构具有友元访问权限，可以从包含其声明的程序中以及同一程序集中的其他任何地方访问它们。

（4）Private：用 Private 关键字声明的结构具有私有访问权限，只能在同一模块中访问。

（5）结构名：要定义的结构的名称，必须是有效的 Visual Basic. NET 标识符。

（6）变量声明：一个或多个 Dim、Private 或 Public 语句，声明作为结构的数据成员的变量或事件。这些声明与普通变量或事件声明一样，遵循相同的规则。也可以在结构中定义常数或属性，但必须至少声明一个变量或事件。

（7）过程声明：作为结构的方法成员的 0 个或多个 Function、Property 或 Sub 过程的声明。这些声明与在结构外一样，遵循相同的规则。

例如：

```
Public Structure student
    Dim num As Integer
    Dim name As String
    Dim sex As String
    Dim birthday As Date
End Structure
```

2. 定义结构类型变量

结构类型反映了所处理对象的抽象特征。在创建结构后，可将过程级变量和模块级

变量声明为该类型。结构变量的定义格式如下：

【格式】

```
[Dim|Public|Private] 变量名 1,变量名 2,…,变量名 n As 结构名
```

【说明】

定义多个具有相同结构的结构变量。

结构类型与结构变量是不同的概念，定义一个结构类型并不意味着系统要分配一块内存单元来存放各种结构成员，只是指定了这个类型的组织结构，即向编译程序“声明”由程序员自己所定义的结构有哪些成员，其类型是什么，长度多少，反映了数据的抽象属性。只有用它来定义某个具体变量时，才占据存储空间。

例如：

```
Dim s As student
```

3. 结构内的成员

结构内的成员可以是基本数据类型，如 Integer、Date、Double 型等，也可以是构造类型(如数组等)。结构内的成员也可以是已定义的另一个结构类型。

例如：

```
'定义一个结构 myDate
Public Structure myDate
    Dim year As Integer
    Dim month As Integer
    Dim day As Integer
End Structure
'使用结构 myDate 来定义 birthday
Public Structure student
    Dim num As Integer
    Dim name As String
    Dim sex As String
    Dim birthday As myDate
End Structure
```

3.2.2 结构变量的使用

1. 结构变量的引用

在定义结构变量之后，就可以使用该变量了。结构变量是由不同类型的成员组成，通常参加运算的是结构变量的成员，引用成员的格式如下：

【格式】

```
结构变量名.成员名
```

【说明】