Visual Basic for Applications

Programming

Damiano SOMENZI

School of Economics and Management

Advanced Computer Skills

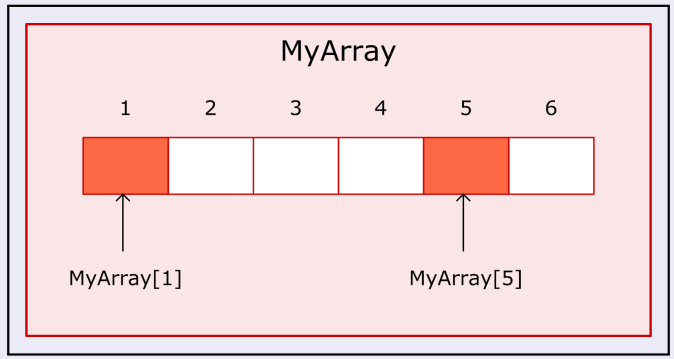
damiano.somenzi@unibz.it

Week 8

Outline

Array

An array holds a set of values of the **same data type**. An array is a **single variable** with many compartments to store values, while a typical variable has only one storage compartment in which it can store only one value

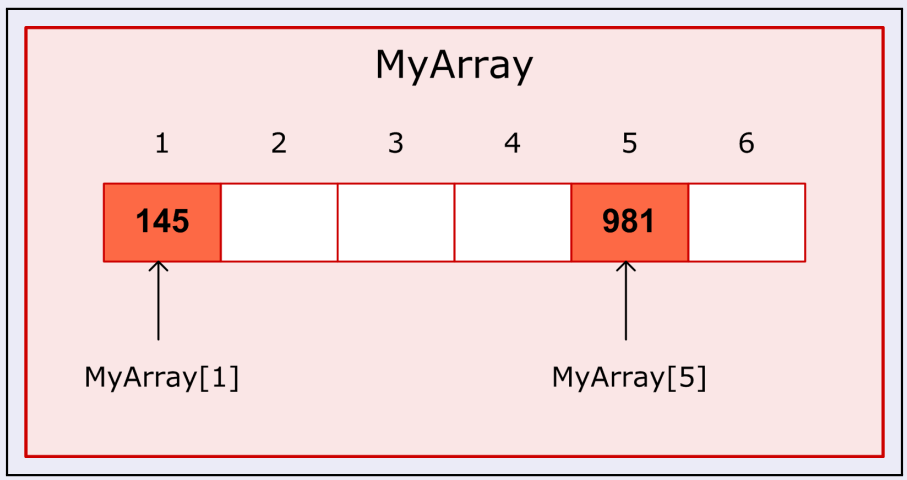


- An array variable is a collection of variables that uses the same name but are distinguished by an index value (`MyArray[1]`,`MyArray[2]`)
- To get all the values that the array holds we refer to it as a whole (`MyArray`)
- To get an identified value of an array we refer to its individual elements (`MyArray[5]`)

Array

Example

To store six numbers (integers) you can create an array variable with six compartments, rather than creating six different variables. Each compartment contains one value



Array Declaration

Array variables are declared the same way as other variables, using the `Dim` statements. **Generally** it is specified the size of the array (as we look at)

- An array holding 10 integers (fixed-size) is declared as

  ```
  Dim Numbers(1 To 10) As Integer
  ```

  ... create an array having 10 compartments ...

  ... the name and an index (an integer between the lower and upper bound) specifies a compartment ...

  ... data could be read from the array or stored into it ...

  1. The array variable is named `Numbers`
  2. It has size 10 (compartments)
  3. The Lower Bound of the array is **1** and the Upper Bound is **10**
  4. `Numbers(1),Numbers(2),..,Numbers(10)` variables are Integer data type variables

Using Arrays

Code	Array Content	Display
`Dim A(1 To 5) As Integer`	0 0 0 0 0	
`A(1) = 123`	123 0 0 0 0	
`A(2) = 98`	123 98 0 0 0	
`A(3) = 341`	123 98 341 0 0	
`MsgBox(A(2))`	123 98 341 0 0	98
`MsgBox(A(4))`	123 98 341 0 0	0
`MsgBox(A(6))`	123 98 341 0 0	Out of Range

Array
Examples

```
Sub arrExample()

   'it declares an array of 4 integers
   'only two integers are stored in it, therefore two compartments hold the value zero

   Dim A(1 To 4) As Integer
   A(1) = 88
   A(4) = 122
   MsgBox (A(1) & " - " & A(2) & " - " & A(3) & " - " & A(4) )
End Sub
```

```
Sub arrExample()

   'the array is the best data structure
   'in order to store 4 integers typed by user

   Dim A(1 To 4) As Integer
   Dim i As Integer
   For i = 1 To 4
        A(i) = Val(InputBox("Type an Integer"))
   Next i
   MsgBox (A(1) & " - " & A(2) & " - " & A(3) & " - " & A(4) )
End Sub
```

Array
Examples

```
Sub arrExample()

    'the array is the best data structure in order to store
    'data registered in the range of cells "A1:A4" (worksheet 1)

    Dim A(1 To 4) As Integer
    Dim w As Worksheet
    Dim r As Integer, c As Integer
    Set w = Worksheets(1)
    c = 1
    For r = 1 To 4
        A(r) = w.Cells(r, c).Value
    Next r
    MsgBox (A(1) & " - " & A(2) & " - " & A(3) & " - " & A(4))
End Sub
```

Array
Examples

```
Sub arrExample()

    'array is the best data structure to hold integers numbers typed by the user
    'in order to display the numbers in reverse order
    'with respect to the order they are typed

    Dim A(1 To 4) As Integer
    Dim i As Integer
    For i = 1 To 4
        A(i) = Val(InputBox("Tell me an integer"))
    Next i
    For i = 4 To 1 Step -1
        MsgBox ("A(" & i & ") -> " & A(i))
    Next i
End Sub
```

Array
Examples

```
Sub arrExample()

    'data typed by the user and stored into an array
    'are registered into the cells of the range "A1:A4" (worksheet 1)

    Dim A(1 To 4) As Integer
    Dim w As Worksheet
    Set w = Worksheets(1)
    Dim i As Integer
    Dim r As Integer, c As Integer
    For i = 1 To 4
        A(i) = Val(InputBox("Tell me an integer"))
    Next i
    c = 1
    For r = 1 To 4
        w.Cells(r, c).Value = A(r)
    Next r
End Sub
```

Array Data Structure

Exercises

one

We need a sub procedure in order to manage an array of 5 words:

- It asks the user for the 5 words, and then it stores them within the array;
- It displays the stored words one by one as well, but in reverse order with respect to the order they are typed

```
Sub words()

    'it creates an array in order to store 5 words typed by the user
    'it displays the stored words in reverse order
    'with respect to the order they are typed

    Dim words(1 To 5) As String
    Dim i As Integer
    For i = 1 To 5
        words(i) = InputBox("Tell me a word")
    Next i
    For i = 5 To 1 Step -1
        MsgBox ("word-" & i & " -> " & words(i))
    Next i
End Sub
```

Array Data Structure

Exercises

two

Assume to have registered in an excel worksheet data concerning with some ski athletes (name, surname, date of birth, and races). Write a program that determines the average age of the athletes skiing the race specified by the user.
The performed tasks should be:

1. Given the date of birth of an athlete the function procedure named *Age* returns his/her age, assuming to determine the age **today** (the function `DateSerial` could be helpful)
2. Given data about the athletes and the race name specified by the user, the sub procedure named *averageAgebySpeciality* computes the average age, but only for the athletes skiing that race. The program displays, using a clear message, the average value. *We assume to manage 10 athletes maximum a time*

Array Data Structure

Exercises

solution: 1/2

```
Sub averageAgebySpeciality(d As Range, ByVal rc As String)
    '...
    Dim i As Integer, tot As Integer, n As Integer
    Dim A(1 To 10) As Integer
    For i = 1 To 10
       If d.Cells(i, 4).Value = rc Then
            A(i) = age(d.Cells(i, 3).Value)
       End If
    Next i
    tot = 0
    n = 0
    For i = 1 To 10
        If A(i) > 0 Then
            tot = tot + A(i)
            n = n + 1
        End If
    Next i
    If n > 0 Then
        MsgBox ("Race: " & rc & " - Average Age: " & tot / n)
    Else
        MsgBox ("The race name does not exist")
    End If
End Sub
```

Array Data Structure

Exercises

solution: 2/2

```
Function age(byVal birth_date As Date) As Integer
    '...
    If DateSerial(Year(Date), Month(birth_date), Day(birth_date)) <= Date Then
        age = Year(Date) - Year(birth_date)
    Else
        age = Year(Date) - Year(birth_date) - 1
    End If
End Function

Sub Main()
    '...
    Dim sd As Range
    Dim sp As String
    Set sd = Worksheets(1).Range("B3:E10")
    sp = InputBox("Write the speciality")
    Call averageAgebySpeciality(sd, sp)
End Sub
```

Outline

Array as Argument

LBound Function

Returns a number representing the smallest available index of an array. The basic syntax is **LBound**(*arrayname*)

UBound Function

Returns a number representing the largest available index of an array. The basic syntax is **UBound**(*arrayname*)

Array as Argument

When an array is declared to be a procedure argument, the array should be declared without the range of indexes specification, just with rounds. Since the smallest and the largest indexes are not available the **LBound** and **UBound** functions should be used

```
Sub SetOfWords(words() As String)
    Dim i As Integer
    For i = LBound(words) To UBound(words)
      ...
    Next i
End Sub
```

```
Sub SetOfNumbers(num() As Integer)
    Dim i As Integer
    For i = LBound(num) To UBound(num)
      ...
    Next i
End Sub
```

Array as Argument

Array as Argument

The procedures are called as we know

```
Sub Main()
    Dim ArrayOfWords(1 To 5) As String
    Dim ArrayOfNumbers(1 To 20) As Integer
    ...
    Call SetOfWords(ArrayOfWords)
    Call SetOfNumbers(ArrayOfNumbers)
End Sub
```

Outline

Array as Argument

Set of Examples

1/2

```
Sub main()

    'it takes the words from the range "A2:A6" and
    'it stores them within an array of 5 elements
    'then it performs the tasks:
    'it counts the number of words having more than 5 characters
    'it displays all the words through a single message

    Dim SetOfWords(1 To 5) As String
    Dim i As Integer
    Dim r As Range
    Set r = Worksheets(1).Range("A2:A6")
    For i = 1 To 5
        SetOfWords(i) = r.Cells(i, 1).Value
    Next i
    Call longWords(SetOfWords)
    Call printWords(SetOfWords)
End Sub
```

Array as Argument

Set of Examples

2/2

```
Sub longWord(words() As String)

    'it counts the number of words having more than 5 characters

    Dim i As Integer
    Dim c As Integer
    c = 0
    For i = LBound(words) To UBound(words)
        If Len(words(i)) > 5 Then c = c + 1
    Next i
    MsgBox ("Number of words larger than 5 -> " & c)
End Sub
```

```
Sub printWords(words() As String)

    'it displays all the words through a single message

    Dim i As Integer
    Dim s As String
    s = ""
    For i = LBound(words) To UBound(words)
        s = s & i & " - " & words(i) & Chr(13)
    Next i
    MsgBox (s)
End Sub
```

Array as Argument

Set Of Examples

1/2

```
Sub Main()

    'it stores 5 numbers typed by the user into an array of 5 elements
    'the user specifies in which worksheet (index) the set of numbers could be stored,
    'and from which cell (row, column) within the worksheet
    'then it performs the following task:
    'it stores the set of numbers in the range of cells, as specified by the user

    Dim dataSet(1 To 5) As Double
    Dim j As Integer, wr As Integer, wc As Integer, wi As Integer
    Dim ws As Worksheet
    For j = 1 To 5
        dataSet(j) = Val(InputBox("Write a number"))
    Next j
    wi = Val(InputBox("Where? (specify a worksheet index)"))
    wr = Val(InputBox("Row?"))
    wc = Val(InputBox("Column?"))
    Set ws = Worksheets(wi)
    Call setValues(dataSet, ws, wr, wc)
End Sub
```

Array as Argument

Set Of Examples

2/2

```
Sub setValues(arr() As Double, w As Worksheet, ByVal r As Integer, ByVal c As Integer)

    'it stores the set of numbers in the range of cells, as specified by the user

    Dim i As Integer
    For i = LBound(arr) To UBound(arr)
        w.Cells(r + (i - 1), c).Value = arr(i)
        w.Cells(r + (i - 1), c).Interior.ColorIndex = 27
    Next i
End Sub
```

Outline

Array Data Structure

Exercise

Given data about student examinations: name, sex and mark, the program determines the average mark by sex.
The performed tasks should be:

1. Given a range of cells, such that a row refers to a student, his/her sex, and his/her final mark, the sub procedure **AverageMarkBySex**, firstly registers the data into three arrays named respectively Student, Sex and Mark, then it computes the average marks for male students and for female students. *The procedure can analyze 10 student a time*
2. The sub procedure **Main** calls averageMarkbySex and it passes to it the range of data. Data are arranged in the Column B (Student Name), Column C (Student Sex) and Column D (Student Mark)

Exercises

```
Sub averageMarkbySex(st As Range)
    '...
    Dim Student(1 To 10) As String, Sex(1 To 10) As String, Mark(1 To 10) As Integer
    Dim m As Integer, f As Integer, mc As Integer, fc As Integer
    Dim i As Integer
    For i = 1 To 10
        Student(i) = st.Cells(i, 1).Value
        Sex(i) = st.Cells(i, 2).Value
        Mark(i) = st.Cells(i, 3).Value
    Next i
    m = 0
    f = 0
    mc = 0
    fc = 0
    For i = 1 To 10
        If Sex(i) = "male" Then
            m = m + 1
            mc = mc + Mark(i)
        ElseIf Sex(i) = "female" Then
                f = f + 1
                fc = fc + Mark(i)
        End If
    Next i
    MsgBox ("Female (" & f & ") Mark Average: " & fc / f)
    MsgBox ("Male (" & m & ") Mark Average: " & mc / m)
End Sub
Sub Main()
    '...
    Dim data As Range
    Set data = Worksheets(1).Range("B1:D10")
    Call averageMarkbySex(data)
End Sub
```