A290 Tools for Computing

By:

Dr. Hossein Hakimzadeh
Computer Science and Informatics
IU South Bend
What do we need to learn in order to write computer programs?

- Fundamental programming constructs:
  - Variables,
  - Arithmetic operators,
  - Input and output
  - Conditionals,
  - Loops,
  - Procedures and functions,
  - Arrays (Multi-Dimensional Arrays),
  - Structures, classes and objects,
  - Files
Multi-Dimensional Arrays:

- Arrays can have more than one dimension
- Like a table of values
- Or a Cube of values.
Multi-Dimensional Arrays:

Syntax:

```vba
Dim ArrayName(HighestRowSubscript, HighestColumnSubscript) as Datatype
```
Example-1

- Two Dimensional Array

Dim ScoreBoard(1,8) As Integer

- This declaration creates 18 storage locations (2x9) in which to put Integer values.
Manipulating the Array

- In order to access each variable (array element) we must use two array indexes.

\[
\begin{align*}
\text{Score}(0,0) &= 2 \\
\text{Score}(1,0) &= 5 \\
\text{Score}(1,4) &= 9
\end{align*}
\]
Example-2

- Two Dimensional Array
  
  Dim strName(2, 3) as String

- This declaration creates 12 storage locations (3x4) in which to put **string** values.
Example-3

- Two Dimensional Array

Dim ScoreBoard(9, 4) As Integer

- This declaration creates 50 storage locations (10x5) in which to put Integer values.
Operations on a 2D Array:

- Initializing the array with zeros
- Displaying the cell contents of the array
- Initializing the array with random numbers
Initializing a 2D Array:

'Make a 10x5 storage locations
Dim ScoreBoard(9, 4) As Integer

Dim Row, Col As Integer

For Row = 0 To 9
    For Col = 0 To 4
        ScoreBoard(Row, Col) = 0
    Next Col
Next Row
Let’s write a Procedure to do the same Initialization

Private Sub Initialize(ByRef TheArray() As Integer)
    Dim Row, Col As Integer
    For Row = 0 To TheArray.GetUpperBound(0)
        For Col = 0 To TheArray.GetUpperBound(1)
            TheArray(Row, Col) = 0
        Next Col
    Next Row
End Sub
Private Sub Print(ByVal TheArray(,) As Integer)

    Dim Row, Col As Integer

    For Row = 0 To TheArray.GetUpperBound(0)
        For Col = 0 To TheArray.GetUpperBound(1)
            Console.Write("{0,4}", TheArray(Row, Col))
        Next Col
    Console.WriteLine()

    Next Row

End Sub
Output after calling Initialize()

0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

(c) Copyright 2007, H. Hakimzadeh
Put Random Numbers in the Array

Private Sub InitializeRandom(ByRef TheArray(_,_) As Integer)
    Dim Row, Col As Integer
    Randomize()        ' Seed the random number generator
    For Row = 0 To TheArray.GetUpperBound(0)
        For Col = 0 To TheArray.GetUpperBound(1)
            TheArray(Row, Col) = CInt(Rnd() * 10)   ' return a number in the range 0 to 10
        Next Col
    Next Row
End Sub
Output after calling InitializeRandom()

4 6 8 4 5
6 1 6 10 0
1 7 10 5 5
2 5 4 4 8
8 3 2 3 2
9 8 5 6 1
4 6 4 6 4
5 4 2 3 6
7 9 8 0 5
2 10 8 7 0

Note:
If we call the InitializeRandom() procedure again, the result will be different!
Other Application of 2D Arrays:

- Tic-tac-toe game
- Storing images
- Displaying graphs
Tic-tac-toe game:

'Make a 3x3 storage locations
Dim TicTacToeBoard(2, 2) As char

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>1</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>X</td>
<td>O</td>
</tr>
</tbody>
</table>

(c) Copyright 2007, H. Hakimzadeh
Storing images:
Storing images:
Storing images:
Storing images:
Storing images:
Storing images:
Medical imaging:
Displaying graphs:
Multi-Dimensional Arrays:

- 3-D arrays
Possible Applications of Multi-Dimensional Arrays:

http://www.youtube.com/watch?v=9UCaxkvWe-8
Storing videos:

A form of 3-D array