Arrays and Indexing

A common way to refer to many instances of the same thing is to give them a single name and index them. So we have Super Bowl XX, Pope John 23, Taco Bell Franchise 229, etc. Indexing is handy in programming.

Indexing, The Basic Idea

- Motivation: When there is a large number of similar things that must be referenced and manipulated, it can be inconvenient to think up a unique name for each, and to refer to them by the name

  + For example: Each of the Seven Dwarfs has a name, but who can remember them?
  + Also, it is difficult to refer to them in a loop since there is no way to enumerate them

- Indexing names the items by associating a base name and a number — the index — with each one

- Computer notation: Dwarf(5) = Happy

Indexing Particulars

- In everyday indexing, it is common to begin the indexing with 1, e.g. May 1, SuperBowl I, Elizabeth I
- The number at which indexing begins is its origin
- Some computer languages use 1 as the origin, but others, including Visual Basic 6.0, use 0 as the index origin. (Gripes: a decent language lets you decide, but whatever.)

- Some computer languages use 1 as the origin, but others, including Visual Basic 6.0, use 0 as the index origin. (Gripes: a decent language lets you decide, but whatever.)

- Notice x0 and x1 are variable names, while x(0) and x(1) are different elements of array x

Arrays

- Arrays are declared like any other variable using a Dim statement

  Dim dwarf(6) As String

- Arrays are used for representing collections of data values, e.g. integers, strings, etc.

  For example:

  dwarf(0) = “Sneezy”
  dwarf(1) = “Dopey”
  dwarf(2) = “Grumpy”
  ...

- Elements of an array must all be of the same type

- The index of an array element is also known as a subscript.

Arrays In VB6.0

- Arrays are declared like any other variable using a Dim statement

  Dim dwarf(6) As String

- Notice

  + The syntax is just like a normal declaration except for the parenthesis pair
  + In the parentheses is the largest desired index
  + The total number of elements of the array will be one more than the largest index, since the origin is 0
  + The type applies to all of the elements

Indexing Arrays

- To refer to different elements of the array, it is necessary only to change the index …

  + The index value must be an integer constant (1), a variable (myIndex) or expression (myIndex+1)

  + A loop can sweep through all elements

The dwarf array values were initialized in event handler Form_Load()
Combining Indexing, Arrays, Loops

A common error is to index beyond the end of the array...

Mini-Exercise #1

Declare a variable that holds 20 doubles (i.e., numbers with a decimal place) that represent weights.

Mini-Exercise #2

Initialize the array weights with values of 100, 200, 300, … 2000

Mini-Exercise #1 -- Answer

Declare a variable that holds 20 doubles (i.e., numbers with a decimal place) that represent weights

Dim weights(19) As Double

Mini-Exercise #2 -- Answer

Initialize the array weights with values of 100, 200, 300, … 2000

Dim weights(19) As Double
Dim i As Integer
i = 0
Do While i < 20
weights(i) = (i + 1) * 100
i = i + 1
Loop

Mini-Exercise #3

Print out the contents of the array of weights
Mini-Exercise #3 -- Answer
Print out the contents of the array of weights

Dim i As Integer
i = 0
Do While i < 20
    Print weights(i)
i = i + 1
Loop

Practice Using Arrays

- Draw a 10-segment “inch worm” on the screen and move it forward
- Use arrays to keep the positions of the segments
- Write procedures to initialize worm and draw it
- Goals of exercise:
  - Practice with arrays
  - Practice with indexing
  - Practice writing procedures
  - Notice how arrays are passed as parameters

Worm code

❖ The code for this example is available on the class web under “Example Code”

Worm Programming – Global Variables

❖ The first step is to declare two arrays. These hold the x and y coordinates of the center of each of the circles that make up the body of the worm.

There are 10 circles in all.

Dim inchwormX(9) As Integer, inchwormY(9) As Integer

Worm Programming – Initialization

❖ Initialize the worm on Form load:

Private Sub Form_Load()
    FillColor = vbGreen
    FillStyle = 0
    Call initializeWorm(inchwormX(), inchwormY())
    Call drawWorm(inchwormX(), inchwormY())
End Sub

initializeWorm procedure

Private Sub initializeWorm(xSeg As Integer, ySeg() As Integer)
    Dim index As Integer
    index = 0  ' Index range 0 thru 9
    Do While index < 10  ' Process all array elements
        xSeg(index) = 1000 + 100 * index
        If (index Mod 2) = 0 Then  ' Test of index odd/even
            ySeg(index) = 1000  ' Even segments
        Else
            ySeg(index) = 900  ' Odd segments
        End If
        index = index + 1  ' Move to next index
    Loop
End Sub
drawWorm procedure

Private Sub segment(x As Integer, y As Integer)
    Circle (x, y), 100, vbGreen
End Sub

Private Sub drawWorm(x1() As Integer, y1() As Integer)
    Dim iterate As Integer
   Cls
    iterate = 0
    Do While iterate < 10
        Call segment(x1(iterate), y1(iterate))
        iterate = iterate + 1
    Loop
    Line (x1(9), y1(9))-(x1(9) + 100, y1(9) - 200), vbGreen
End Sub

Making the worm crawl

Private Sub Form_Click()
    Dim index As Integer
    index = 0
    ' make the worm crawl 200 twips in the x direction
    Do While index < 10
        inchwormX(index) = inchwormX(index) + 200
        index = index + 1
    Loop
    Call drawWorm(inchwormX(), inchwormY())
End Sub