A201 Object Oriented Programming with Visual Basic .Net

By:

Dr. Hossein Hakimzadeh
Computer Science and Informatics
IU South Bend

(c) Copyright 2007 - 2014, H. Hakimzadeh
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
Why do we need Structures?

- A **structure** provides the means to group together related data elements.

- A **structure** is heterogeneous aggregate of data elements.

- Visual Basic allows programmers to define their own complex data types using the so called ‘Structure’ statement.
Structures

- Syntax:

```plaintext
[Public|Private] Structure StructureName

Dim Field1 As Datatype
Dim Field2 As Datatype
...

End Structure
```

- The name of the new structure
- Elements of the structure
- Accessibility

(c) Copyright 2007 - 2014, H. Hakimzadeh
Defining a new Structure

- To define a new data type called Employee.

```vbnet
Public Structure Employee
    Dim name As String
    Dim HourlyWages As Double
    Dim HoursWorked As Double
    Dim GrossPay As Double
End Structure
```
Declaring a variable using the new Employee Structure

DIM employee1 As Employee

A new variable of type Employee is created
Accessing the new Structure

- Fields are referenced using the dot notation

```plaintext
employee1.name = "John Smith"
employee1.HourlyWages = 8.75
employee1.HoursWorked = 38
```

Now we can store values into the elements of the structure.
Example 1

Employee Structure
Example 1

Module Module1

Public Structure Employee
    Dim name As String
    Dim HourlyWages As Double
    Dim HoursWorked As Double
    Dim GrossPay As Double
End Structure

Sub Main()
    Dim employee1 As Employee
    employee1.name = "John Smith"
    employee1.HourlyWages = 8.75
    employee1.HoursWorked = 38
End Sub

End Module

(c) Copyright 2007 - 2014, H. Hakimzadeh
Example 2

Module Module1

Public Structure Employee
   Dim name As String
   Dim HourlyWages As Double
   Dim HoursWorked As Double
   Dim GrossPay As Double
End Structure

Sub Main()
   Dim employee2, employee3, employee4 As Employee
   employee2.name = "Mary Thomas"
   employee3.name = "Sue Boom"
   employee4.name = "Jill Anderson"
End Sub

End Module
Creating an Array of Structures

Let’s create an array of 5 employees:

Dim WorkGroup(4) As Employee
Creating an Array of Structures

- Let’s create an array of 5 employees:

  Dim WorkGroup(4) As Employee

Note that Visual Basic actually allocates array elements 0 through 4, which gives you 5 elements.
Storing employee information in the array of employees:

Sub Main()

Dim WorkGroup(4) As Employee ' Create an array of 5 employees

WorkGroup(0).name = "John Smith"
WorkGroup(0).HourlyWages = 8.75
WorkGroup(0).HoursWorked = 38

WorkGroup(0).GrossPay = WorkGroup(0).HoursWorked * WorkGroup(0).HourlyWages

....

....

End Sub
Storing employee information in the array of employees:

Sub Main()

Dim WorkGroup(4) As Employee ' Create an array of 5 employees

WorkGroup(0).name = "John Smith"
WorkGroup(0).HourlyWages = 8.75
WorkGroup(0).HoursWorked = 38
WorkGroup(0).GrossPay = WorkGroup(0).HoursWorked * WorkGroup(0).HourlyWages

WorkGroup(1).name = "Mary Thomas"
WorkGroup(1).HourlyWages = 12.75
WorkGroup(1).HoursWorked = 20

....

....

End Sub
Storing employee information in the array of employees:

```vba
Sub Main()
    Dim WorkGroup(4) As Employee ' Create an array of 5 employees
    WorkGroup(0).name = "John Smith"
    WorkGroup(0).HourlyWages = 8.75
    WorkGroup(0).HoursWorked = 38
    WorkGroup(0).GrossPay = WorkGroup(0).HoursWorked * WorkGroup(0).HourlyWages

    WorkGroup(1).name = "Sue Thomas"
    WorkGroup(1).HourlyWages = 12.75
    WorkGroup(1).HoursWorked = 20

    WorkGroup(2).name = "Sue Boom"
    WorkGroup(2).HourlyWages = 10.25
    WorkGroup(2).HoursWorked = 25
    WorkGroup(2).GrossPay = WorkGroup(1).HoursWorked * WorkGroup(1).HourlyWages
    ....
    ....
End Sub
```

(c) Copyright 2007 - 2014, H. Hakimzadeh
Operation on Employee Structure

What are some common operations that one may perform on an employee structure?

1. Read information into the structure (e.g. insert/update)
2. Print the employee information
3. Calculate the Gross Pay for the employee
4. Retrieve information from the structure
   - GetName()
   - GetHourlyWage()
   - GetHoursWorked()
   - GetGrossPay()
Reading Information Into a Structure:

Private Sub ReadEmployeeInfo(ByRef TheEmployee As Employee)

    Console.Write("Please enter Employee Name :")
    TheEmployee.name = Console.ReadLine()

    Console.Write("Please enter Employee Hourly Wages:")
    TheEmployee.HourlyWage = CDbl(Console.ReadLine())

    Console.Write("Employee Hours Worked : ")
    TheEmployee.HoursWorked = CDbl(Console.ReadLine())

End Sub
Print Procedure

Private Sub Print(ByVal TheEmployee As Employee)
    Console.WriteLine("Employee Name : {0}", TheEmployee.name)
    Console.WriteLine("Employee Hourly Wages: {0,4:c}", TheEmployee.HourlyWages)
    Console.WriteLine("Employee Hours Worked: {0}", TheEmployee.HoursWorked)
    Console.WriteLine("Employee Gross Pay : {0,6:c}", TheEmployee.GrossPay)
End Sub
Calculate the Gross Pay:

Private Sub CalculateGrossPay(ByRef TheEmployee As Employee)

    TheEmployee.GrossPay = TheEmployee.HourlyWage * TheEmployee.HoursWorked

End Sub
Get the Employee Name:

Private Function GetName(ByVal TheEmployee As Employee) As String
    Return (TheEmployee.name)
End Function
Get the Employee’s Hourly Wage:

Private Function GetHourlyWage(ByVal TheEmployee As Employee) As Double
    Return (TheEmployee.HourlyWage)
End Function
Get the Employee’s Hours Worked:

Private Function GetHoursWorked(ByVal TheEmployee As Employee) As Double
    Return (TheEmployee.HoursWorked)
End Function
Get the Employee’s Gross Pay:

Private Function GetGrossPay(ByVal TheEmployee As Employee) As Double
    Return (TheEmployee.GrossPay)
End Function
Putting it all together....

☐ Next we will create a complete program using structures
Option Explicit On
Option Strict On

Module Module1

Public Structure Employee
  Dim Name As String
  Dim HourlyWage As Double
  Dim HoursWorked As Double
  Dim GrossPay As Double
End Structure

Sub Main()
  Dim Employee1 As Employee

  ReadEmployeeInfo(Employee1)
  Print(Employee1)

  CalculateGrossPay(Employee1)
  Print(Employee1)

  Console.WriteLine(GetName(Employee1))
  Console.WriteLine(GetHourlyWage(Employee1))
  Console.WriteLine(GetHoursWorked(Employee1))
  Console.WriteLine(GetGrossPay(Employee1))
End Sub

' The procedures and functions are inserted here......

End Module