

```
Public Class condition
    Private p_lhs As Object
    Private p_rhs As Object
    Private p_relOperator As String

    Public Sub New(ByVal lhs As Object, ByVal rhs As Object, ByVal cond As String)
        p_lhs = lhs
        p_rhs = rhs
        p_relOperator = cond
    End Sub

    Property lhs()
        Get
            Return p_lhs
        End Get
        Set(ByVal value)
            p_lhs = value
        End Set
    End Property
    Property rhs()
        Get
            Return p_rhs
        End Get
        Set(ByVal value)
            p_rhs = value
        End Set
    End Property
    Property relOperator()
        Get
            Return p_relOperator
        End Get
        Set(ByVal value)
            p_relOperator = value
        End Set
    End Property

    Public Function isTrue() As Boolean
        Dim result As Boolean = False

        Select Case p_relOperator
            Case "<"
                If p_lhs < p_rhs Then result = True
            Case ">"
                If p_lhs > p_rhs Then result = True
            Case "="
                If p_lhs = p_rhs Then result = True
            Case "<>"
                If p_lhs = p_rhs Then result = True
            Case Else
                result = False
        End Select
        Return result
    End Function
End Class
Imports system.text
Public Class Decode
    Private p_conditions As List(Of condition)
```

C:\Documents and Settings\bkm\My Documents...\NestedIfTester\NestedIfTester\Condition.vb2

```
Public Sub Decode()  
    p_conditions = New List(Of condition)  
  
End Sub  
Public Sub New(ByVal conditionList As List(Of condition))  
    p_conditions = New List(Of condition)  
    For Each st As condition In conditionList  
        p_conditions.Add(st)  
    Next  
End Sub  
Public Function getCode() As String  
    Dim codeString As New StringBuilder  
    For index As Integer = 0 To p_conditions.Count - 1  
        If (p_conditions.Item(index).isTrue) Then  
            codeString.Append("1")  
        Else  
            codeString.Append("0")  
        End If  
  
    Next  
  
    Return codeString.ToString  
  
End Function  
End Class
```