



# מחלקות בשפת ASP

ניר אדר

מסמך זה הורד מהאתר <http://underwar.livedns.co.il>.  
אין להפיץ מסמך זה במדיה כלשהי, ללא אישור מפורש מאת המחבר.  
מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע במסמך, וכן  
לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת, המחבר עשה את מירב המאמצים כדי לספק את  
המידע המדויק והמלא ביותר.

כל הזכויות שמורות לניר אדר

Nir Adar

Email: [underwar@hotmail.com](mailto:underwar@hotmail.com)

Home Page: <http://underwar.livedns.co.il>

אנא שלחו תיקונים והערות אל המחבר.

## תוכן עניינים

2	..... תוכן עניינים
3	..... פתיחה
4	..... פונקציות בונות ופונקציות הורסות
5	..... מתודות ציבוריות ופרטיות
6	..... שדות
7	..... מאפיינים
8	..... הורשה
9	..... דוגמא מסכמת
10	..... רשימת מקורות

## פתיחה

שפת ASP תומכת באבסטרקצית המחלקות הקיימת בשפות תכנות מונחות עצמים שונות. השפה מאפשרת לנו להגדיר מבנה נתונים (המחלקה), תכונות של מבנה הנתונים (מאפיינים) ופעולות שניתן לבצע עליו. במסמך זה אני מניח שהקורא מכיר את הקונספט של תכנות מונחה עצמים, ואראה את הכלים ששפת ASP נותנת לנו לכך.

נפתח בדוגמא המראה שימוש במחלקות. נגדיר מחלקה בשם ClassA אשר לה מתודה PrintStr המקבלת מחרוזת ושולחת אותה ללקוח, בצירוף תו שורה חדשה. על ידי שימוש בהרשאות מסוג public ו-private אנחנו יכולים לקבוע אילו חלקים מהמחלקה שלנו יהיו נגישים למשתמש בה, ואילו יהיו שייכים למימוש הפנימי בלבד.

```
Class ClassA
  Public Sub PrintStr(s)
    Response.Write s & "<BR>"
  End Sub
End Class
```

שימוש לדוגמא במחלקה:

```
Dim objA
Set objA = New ClassA
objA.PrintStr("Hello, World!")
```

בסיום העבודה עם האובייקט, לשחרור המשאבים, נציב בו Nothing:

```
Set objA = Nothing
```

## פונקציות בונות ופונקציות הורסות

פונקציה בונה (Constructor) נקראת מיד לאחר הקצאת הזיכרון עבור אובייקט חדש.  
 פונקציה הורסת (Destructor) נקראת רגע לפני שמשוחרר הזיכרון עבור האובייקט.  
 בשפת ASP, הפונקציה הבונה של מחלקה נקראת תמיד Class\_Initialize והפונקציה ההורסת הינה Class\_Terminate. אובייקט נוצר כאשר אנו משתמשים במילה new. אובייקט נהרס כאשר אנחנו שמים nothing לתוך משתנה ההתייחסות האחרון השומר עליו, או כאשר מסתיימת ריצת הדף.

נביט למשל במחלקה הבאה:

```
Class ClassA
  Public Sub PrintStr(s)
    Response.Write s & "<br>"
  End Sub

  Private Sub Class_Initialize()
    Response.Write "Class Initialize<br>"
  End Sub

  Private Sub Class_Terminate()
    Response.Write "Class Terminate<br>"
  End Sub
End Class
```

כעת אם נריץ את הקוד הבא:

```
Dim objA
Set objA = New ClassA
objA.PrintStr("Hello, World!")
Set ObjA = Nothing
Response.Write("The End<br>")
```

אזי פלט התוכנית שלנו יהיה:

```
Class Initialize
Hello, World!
Class Terminate
The End
```

כעת נריץ את הקוד ללא השמת Nothing למשתנה ההתייחסות:

```
Dim objA
Set objA = New ClassA
objA.PrintStr("Hello, World!")
Response.Write("The End<br>")
```

במקרה זה הפלט של הקוד (בהנחה שזהו הקוד היחיד בדף) הינו:

```
Class Initialize
Hello, World!
The End
Class Terminate
```

- בשפת ASP פונקציות בונות והורסות אינן יכולות לקבל פרמטרים.

## מתודות ציבוריות ופרטיות

על ידי שימוש במילים ו-private ו-public ניתן להגדיר רמות גישה שונות אל המחלקה. אם נגדיר מתודות בהרשאה private – רק מתודות השייכות לאותה מחלקה יוכלו לקרוא להן.

נביט במחלקה הבאה:

```
Class ClassA
Public Sub PublicMethod1()
    Response.Write "This is public method<br>"
End Sub

Private Sub PrivateMethod1()
    Response.Write "This is private method<br>"
End Sub

Public Sub PublicMethod2()
    PrivateMethod1
End Sub
End Class
```

המשתמש במחלקה כזו, כפי שיודגם מיד, יכול לקרוא רק למתודות PublicMethod1 ול-PublicMethod2. אם זאת, ל-PublicMethod2 מותר לקרוא ל-PrivateMethods1 מכיוון שהן שייכות לאותה מחלקה.

```
Dim objA
Set objA = New ClassA
ObjA.PublicMethod1
ObjA.PrivateMethod1
ObjA.PublicMethod2
```

השורה האדומה, אם תכתב, תגרום לשגיאה בעת טעינת הדף.

## שדות

נוכל להגדיר שדות עבור המחלקה שלנו. שדות אלו יכולים להיות ציבוריים או פרטיים. יצירת שדות נעשית על ידי כתיבת שם השדה ולפניו את ההרשאה שאנו רוצים לתת לו. בדוגמא הבא נדגים שדות ציבוריים ושדות פרטיים. יש לזכור כי עקרונות תכנות מונחה עצמים נכונים דורשים לרוב כי כל השדות יהיו פרטיים.

השדות Name, Age הינם ציבוריים בדוגמא זו.

```
Class Person
    Public Age
    Public Name

    Public Sub PrintInformation()
        Response.Write "Person Name: " & Name & ", Age: " & _
            Age & "<br>"
    End Sub
End Class
```

דוגמא לשימוש:

```
Dim objJohn
Set objJohn = New Person
objJohn.Age = 22
objJohn.Name = "John"
objJohn.PrintInformation
Set objJohn = Nothing
```

## מאפיינים

בשפות כמו C++ מקובל לשים את כל השדות בהרשאה private ולכתוב פונקציות get/set המספקות גישה אליהם.

ASP מציעה דרך ההופכת מנגנון זה לשקוף למשתמש – מאפיינים (properties).

מאפיינים, כלפי המשתמש, נראים כמו משתנים, אך למעשה כאשר המשתמש ניגש אליהם נקראת פונקציה המוגדרת על ידי המתכנת.

נשים לב ששפת ASP מציעה טרמינולוגיה מעט שונה מהמקובל בשפות אחרות: get נקרא גם בשפת ASP בשם Get אולם המשמעות המקובלת של set, קביעת ערך חדש למאפיין, מכונה בשפת ASP בשם .Let

לדוגמא נבנה מחלקה המייצגת בן אדם, כאשר שמו וגילו יקבעו ויוחזרו דרך מאפיינים:

```
const ILLEGAL_AGE = 1000

Class Person
  Private mAge
  Private mName

  Public Sub PrintInformation()
    Response.Write "Person Name: " & mName & ", Age: " & _
    mAge & "<br>"
  End Sub

  Public Property Get Age()
    Age = mAge
  End Property

  Public Property Let Age(ByVal NewAge)
    If (NewAge > 0) Then
      mAge = NewAge
    Else
      Err.Raise ILLEGAL_AGE, "Illegal age specified"
    End If
  End Property

  Public Property Get Name()
    Name = mName
  End Property

  Public Property Let Name(ByVal NewName)
    mName = NewName
  End Property
End Class
```

דוגמא לקוד המשתמש במחלקה זו:

```
Dim objJohn
Set objJohn = New Person

' Using the Age Let
objJohn.Age = 22
objJohn.Name = "John"
objJohn.PrintInformation
Response.Write "Age (using Age Get): " & objJohn.Age

Set objJohn = Nothing
```

פלט:

```
Person Name: John, Age: 22
Age (using Age Get): 22
```

כאשר אנו מנסים לקרוא את ערכו של המאפיין, נקראת הפונקציה Get, וכאשר אנו מנסים לקבוע את ערכו נקראת הפונקציה Let, וכל זה באופן שקוף למשתמש.

## הורשה

שפת ASP איננה תומכת בתכונת ההורשה של מחלקות.

## דוגמא מסכמת

נציג ללא הסברים רבים דוגמא נוספת לקוד של מחלקה, השומרת מידע אודות תוכנית טלוויזיה.  
הדוגמא לקוחה מן המאמר "Using Classes within VBScript" מאת Mark Lidstone.

### TVProgramClass.asp

```
<%
  Class TVProgram
    Private internal_StartTime
    Public Property Get StartTime
      StartTime = Hour(internal_StartTime) & _
        ":" & Minute(internal_StartTime)
    End Property
    Public Property Let StartTime(ByVal varTimeIn)
      If IsDate(varTimeIn) Then
        internal_StartTime = varTimeIn
      End If
    End Property

    Private internal_ProgramDate
    Public Property Get ProgramDate
      ProgramDate = Day(internal_ProgramDate) & _
        " " & MonthName(Month(internal_ProgramDate)) & _
        " " & Year(internal_ProgramDate)
    End Property
    Public Property Let ProgramDate(ByVal varDateIn)
      If IsDate(varDateIn) Then
        internal_ProgramDate = varDateIn
      End If
    End Property

    Public ProgramTitle
  End Class
%>
```

קוד לדוגמא המשתמש במחלקה:

```
<!-- #include virtual="TVProgramClass.asp" -->
<%

  Dim objTVShow
  Set objTVShow = New TVProgram

  objTVShow.StartTime = CDate("17:30")
  objTVShow.ProgramDate = DateSerial(1999,9,17)
  objTVShow.ProgramTitle = "The Jerry Springer Show"

  %>
  <%= objTVShow.ProgramTitle %> is on at <%= objTVShow.StartTime %> on
  <%= objTVShow.ProgramDate %>.
```

## רשימת מקורות

1. Mark Lidstone (1999), "Using Classes within VBScript"
2. Microsoft MSDN (2004), "Visual Basic 6.0 - Programming with Objects"
3. James Lindën (2001), "Simple Graphs With ASP"