

## اعلان متغیرهای شی

قبل از اینکه نمونه ای از یک مولفه را بسازیم، شما باید یک متغیر شی اعلان کنید که به شی COM اشاره می کند. شما یک متغیر عمومی را به یکی از دو شکل عمومی و یا از نوع خاص اعلان می کنید و این به چگونگی نقشه شما برای استفاده از آن متغیر بستگی دارد. در بعضی حالتها شما در زمان طراحی (Design Time) نوع شی را که در برنامه استفاده خواهید کرد، نمی دانید. در این مواقع می توانید از متغیرهای شی عمومی برای نگهداری اشاره گری به هر نوع شی استفاده کنید. برای مثال ممکن است شما بخواهید تابعی بنویسید که با اشیایی از کلاسهای مختلف کار کند. در این حالت باید متغیرهایی از نوع Object تعریف کنید. کد زیر نوع داده Object را برای اعلان یک متغیر شی عمومی استفاده می کند:

```
Dim MyObj As Object
```

وقتی شما از این نوع اعلان استفاده می کنید، در زمان طراحی اطلاعات مشخصی درباره شی وجود ندارد. ویژوال بیسیک باید کارهای اضافی برای دسترسی به شی را در زمان اجرا انجام دهد. این مسائل اثرات منفی روی کارآیی برنامه های مشتری (Client) دارند. این نوع اعلان متغیرها الحاق دیر (Late Binding) نامیده می شود و فقط زمانی باید از آن استفاده کنید که ناچار باشید.

اگر شما در زمان طراحی نوع شی را که در زمان اجرا ساخته خواهد شد بدانید، متغیری از نوع خاص را اعلان کنید. کد زیر یک متغیر که اشاره گری را به شی Microsoft Excel می کند، اعلان کرده است:

```
Public xlApp As Excel.Application  
Private xlChart As Excel.Chart  
Static xlSheet As Excel.Worksheet
```

این کد برای کلاسهای ثبت شده دیگر مانند ماژولهای کلاس پروژه نیز صدق می کند:

```
Static obj1 As Project1.Class1
```

ویژوال بیسیک ترکیب فراخوانی را که برای متغیرهای از نوع خاص استفاده می شود، بررسی می کند. کمپایلر می تواند کارآمدترین کد را برای دسترسی به شی در زمان اجرا تولید کند. در زمان طراحی، ویژوال بیسیک لیست خودکار اعضا را فراهم می کند که یک لیست باز شونده از اعضای در دسترس را (مانند خواص و متدها) نمایش می دهد. این نوع الحاق، الحاق زود نامیده می شود.

## نمونه سازی یک شی

وقتی شما یک متغیر شی را اعلان می کنید، ویژوال بیسیک حافظه کافی برای نگهداری یک شی را به آن اختصاص می دهد. وقتی یک شی نمونه سازی شده است، ویژوال بیسیک یک مولفه در حافظه می سازد و متغیر شی به آن نمونه از مولفه اشاره می کند. همانطور که در بخش قبل توضیح دادیم، شما باید مولفه را قبل از آنکه نمونه ای از آن را بسازیم، بطور صحیح ثبت کنید، همچنین در برنامه ها بیشتری باید به کتابخانه شی مولفه ارجاع داده باشید.

در ویژوال بیسیک، سه راه برای ساختن نمونه ای از یک شی میزبان COM وجود دارد:

۱- استفاده از تابع `GetObject`

۲- استفاده از تابع `CreateObject`

۳- استفاده از لغت کلیدی `New` با یک دستور `Set`

## • استفاده از تابع GetObject

اگر نمونه ای از یک شی وجود دارد و یا اگر بخواهید نمونه ای از یک شی از یک فایل موجود را بسازید از تابع GetObject استفاده کنید. اولین پارامتر تابع GetObject پارامتر PathName است که مسیر کامل و نام فایلی را که حاوی شی مورد نظر شماست می باشد. اگر شما پارامتر PathName را حذف کنید، پارامتر دوم که Class می باشد ضروری است. پارامتر Class یک مقدار رشته ای است که کلاس شی را نشان می دهد. کلاس الگویی است از یک نمونه از شی در زمان اجرا. کلاس خواص شی و متدهای استفاده شده برای کنترل کردن رفتار شی را تعریف می کند. خواص و متدها نیز در لیست اعضای خودکار نمایش داده می شوند. مثال زیر با استفاده از Microsoft Word Document ، MyDocument.doc را ساخته و آن را در حالت پیش نمایش چاپ نشان می دهد:

```
Sub ShowDocument()  
    'Creating a specific object variable  
    Static wdDoc As Word.Document  
    'Opens Microsoft Word and displays MyDocument.doc  
    Set wdDoc = GetObject("C:\MyDocument.doc", "Word.Document")  
    'By default new objects are not visible  
    wdDoc.Parent.Visible = True  
    wdDoc.PrintPreview  
End Sub
```

کد زیر نمونه ای از برنامه مرتبط با پسوند xls ساخته و شی را با فایل مورد نظر فعال می کند:

```
Dim xl As Object  
'Do not specify a class type  
Set xl = GetObject("C:\MyFiles\Earnings.xls")
```

## • استفاده از تابع CreateObject

اگر نمونه ای در حال حاضر نیست و شما نمی خواهید وقتی شی ساخته شد فایلی را باز کنید، از تابع CreateObject استفاده کنید. در ادامه اگر باید در برنامه بدلیل اینکه نوع خاص شی را تا زمان اجرا (الحاق دیر) نمی دانید، یک متغیر شی عمومی استفاده کنید، تابع CreateObject را برای نمونه سازی یک کلاس استفاده کنید. اگر یک شی مجبور باشد خودش را بصورت نمونه تکی ثبت شی کند، ویژوال بیسیک فقط یک نمونه از شی را می سازد، در غیر این خصوص بسیاری از زمانها شما CreateObject را فراخوانی می کنید. مثال زیر تابع CreateObject را برای ساختن یک نمونه از Microsoft Excel بکار می برد. آن مرجعی را برای دسترسی به خواص آشکار Microsoft Excel استفاده می کند و سپس متد Quit را برای بستن شی استفاده می کند. سرانجام خود مرجع نیز با استفاده از لغت کلیدی Nothing آزاد می گردد.

```
Dim xlApp As Object  
Set xlApp = CreateObject("Excel.Application")  
  
With xlApp  
    'Set Visible property to True to see the application  
    .Visible = True  
    [statements]  
    'When you finish, use the Quit method to close the application  
    .Quit  
End With  
'Release the reference  
Set xlApp = Nothing
```

## استفاده از لغت کلیدی New با استفاده از Set

اگر شما می خواهید مرجعی را به یک نوع کتابخانه ای برای مولفه خارجی مقاردهی کنید و می توانید یک متغیر شی مخصوص (الحاق زودرس) را برای آن استفاده کنید، برای ساختن نمونه ای از کلاسی که می خواهید در برنامه خودتان بکار ببرید، از لغت کلیدی New به همراه عبارت Set استفاده کنید.

لغات Dim، Private، Public، ReDim و Static فقط یک متغیر را که به یک شی اشاره دارد، اعلان می کنند. تا زمانی که از عبارت Set برای اشاره کردن به یک شی خاص استفاده نکرده اید واقعاً هیچ عمل نمونه سازی انجام نمی گیرد. مثال زیر از لغت کلیدی New به همراه عبارت Set برای ساختن نمونه ای از شی Microsoft Word استفاده کرده است:

```
Dim wd As Word.Application
Set wd = New Word.Application
With wd
.Visible = True
'Add a new document to the Word application
.Documents.Add
'Add text to the active document
.Selection.TypeText Text:="This text was added"
'Print the current document
.ActiveDocument.PrintOut
End With
```

نویسنده: **بادوگر ویرتوال بیسیک**

آدرس وبلاگ: **بادوی ویرتوال بیسیک** <http://vbLog.persianblog.com>



پست الکترونیک: [mam\\_programmer@yahoo.co.uk](mailto:mam_programmer@yahoo.co.uk)