

## تقرؤون في هذا العدد

- تقنية WCF - الجزء الأخير
- البرمجة الكائنية OOP
- أمير على عرش نفسه
- المختصر النافع في لغة XML
- آلية النسخ واللصق برمجيا
- كواليس ناجح دوخهم
- المختصر النافع في Enum
- من ASP إلى MVC و Web API
- ليس بين الرفوف

مجلة تقنية شهرية

# مجلة الرابط

ومنافع أخرى

1. إن سألت عن مساعدي، فهو ساعدي
2. اعمل جميلا، ولا تنتظر جميلا
3. عَيْني الْمَرْمُودَةُ وَلَسْتُ أَجْرِبًا
4. بذل الجهود في تقليل العهود

**مبرمجوكا**



خالد السعداني

## حقل الواقع

الارتقاء على ظهور الناس لبلوغ منزلة ما، أشبه بحركة البهلوان الذي يتمشى على الحبل ويدها متعلقتان بثابت، فإن تحرك الثابت، أو خذل الناس سقط البهلوان، وهويت أنت أرضا.

## مجلة الرابط؟

مجلة تقنية شهرية تهتم بمجالات البرمجة والتطوير بشكل خاص، وبالمعلومات بشكل عام، وتتطرق بشكل ثانوي لقضايا الأدب والدين.

## المحتويات

- افتتاحية الرابط -س: لماذا؟ ج: لهذا !
- تقنية WCF - الجزء الأخير
- البرمجة الكائنية OOP
- أمير على عرش نفسه
- المختصر النافع في لغة XML
- آلية النسخ والالصق برمجيا
- كواليس ناجح دوخهم
- المختصر النافع في Enum
- قهوة الرابط
- نظرة شمولية عن ASP.NET
- كلمة ختامية: ليس بين الرفوف

# مجلة الرابط

## سين: لماذا؟ جيم: لهذا!



خالد السعداني

ما علق بذاكرتي من فصول الدراسة، قصة عن عنتر بن شداد كان قد رواها لنا أحد الأساتذة، تقول أن رجلا قدم إلى عنتر يسأله عن أسباب بأسه وأسرار انتصاره في كل نزالاته، فكان جواب عنتر للرجل عمليا:  
إذا أردت أن تفهم فهات اصبعك لأعضه، وخذ إصبعي لتعضه، ولنرى من سيستسلم الأول!

وافق الرجل على التحدي، وتبادلا عض الأصبع، وظل كل منهما يكابد الألم إلى أن صاح الرجل:  
توقف...أنا أستسلم يا عنتر!

فابتسم عنتر بكبرياء وقال: لهذا أنتصر دوما!

فسأله الرجل مستفسرا: كيف ذلك يا عنتر؟

فرد عنتر: لو صبرت قليلا لاستسلمت قبلك، فأنا أيضا تألمت بشدة لكنني كنت أسلي نفسوي وأهيؤها لصراخك واستسلامك في أية لحظة، وكذلك كان.

العبرة من هذه القصة أن النجاح ليس سهلا، وفي الوقت نفسه ليس صعبا، جوهر المسألة شيء من الصبر وشيء من الوقت، ولكل نجاح سبب، فإن سألت أحد الناجحين: لماذا نجحت؟ فلن يجيبك: أمطرت السماء علي ذهبا، وإنما سيجيبك بالأسباب: لهذا نجحت!

آمن بهذه المسألة، واصبر في عملك صبورا جميلا حتى تقطف ثماره، دون أن تتكاسل أو تستعجل، فما يأتي بالسهل يمشي بالسهل كما يقول المثل الإنجليزي Easy Come Easy Go

لو ركنا إلى زاوية الذرائع لما صدر هذا العدد الذي بين يديك عزيزي القارئ، ولو اعتذرنا لناسبنا الاعتذار، بحكم أن شهري نونبر وديسمبر كانا حافلين بالأعمال والمشاغل، لكنني خاملت على نفسي لأخرج هذا العدد المزدوج في محاولة مني لسد الخلل ومواصلة السير في هذا الدرب النبيل.

أرجو من الله العلي القدير أن ينفع بهذا الإصدار كل مهتم بالشأن البرمجي، وكل متابع لإنتاجات أكاديمية المبرمجين العرب.

دام لكم البشر والفرح!

# تقنية WCF: ثورة الخدمات

(الجزء الثالث)

## إنشاء مشروع تطبيقي بتقنية WCF

خالد السعداني

لماذا سنحتاج إلى إنشاء تطبيق في حاجة إلى التواصل مع تطبيق آخر؟ وبصيغة أخرى:

متى نحتاج إلى برمجة تطبيق لا يكتمل عمله إلا في ظل تواصله مع تطبيق آخر؟ الجواب عن هذا السؤال وغيره من الأسئلة المماثلة (أسميها الروافد التي تصب في آخر المطاف في بحر دواعي استخدام الأنظمة الموزعة) يمكن تنويحه حسب الحاجة البرمجية.

فإن الحاجة إلى أنظمة موزعة وتطبيقات تتواصل فيما بينها تكون مُلِحَّةً حينما نريد تركيز البيانات في نقطة واحدة متعددة الفروع. كأن يتم التواصل بين الأنظمة العملية الموزعة على عدة أجهزة. وبين النظام المركزي المثبت على جهاز السيرفر. أو حينما نريد إنشاء تطبيق في حاجة إلى موارد خارجية لا يمكنه الوصول إليها إلا عبر

رأينا في الجزء السابق من سلسلة "ثورة الخدمات" كيف نقوم بإنشاء أول مشروع بتقنية WCF على بيئة الفيجوال ستوديو. وتطرقنا في غضون الحصة إلى شرح مختلف الملفات التي تأتي مع هذه النوعية من المشاريع. بعد أن قدمنا للنموذج ABC ((Address, Binding, Contract الذي تتبناه تقنية WCF.

في حصة اليوم إن شاء الله سوف نستمر مع هذه التقنية القوية. التي باتت تلعب دورا بارزا في صناعة الأنظمة الموزعة. لكن قبل مباشرة ذلك تعالوا بنا نجيب على سؤال ينبش دماغ كل مقبل على هذا الصنف من البنات:

وسيط شبكي (أنترنت، شبكات داخلية أو خارجية)، أو حينما نريد تقسيم المهام بين عدة تطبيقات لرفع جودة النظام وتسهيل عمليات الصيانة والمتابعة، أو حينما نريد استثمار منافع تطبيقات جاهدة ونستعملها في تطبيقنا بدل إعادة برمجتها من الصفر وربما بجودة نقل عن تلك التي تتصف بها التطبيقات الجاهدة التي تلبى حاجتنا، أو حينما نسعى إلى التحكم في تطبيق معين عبر تطبيق آخر، وقس على ذلك...

إذن فدواعي تبني هذا النمط البرمجي ليست ترفاً بقدر ما هي حاجة ماسة تفرضها طبيعة الأنظمة المراد إحداثها وكذلك سرعة تطور تكنولوجيا المعلومات التي باتت ترى في الأنماط البرمجية الكلاسيكية بُنيَّةً لم تعد تصلح لمواكبة مستجدات العصر.

الآن تعالوا بنا أحبتي في الله، لنواصل المشوار الذي بدأناه مع تقنية WCF. وفي هذا الجزء الثالث بحول الله، سنقوم بإنشاء تطبيق من نوع خدمة، وسنقوم باستضافته، ثم بعد ذلك سنقوم بالاتصال به من تطبيق آخر لكي نظفر بما يقدمه من وظائف، في محاولة بسيطة منا إلى التدليل على هذا النوع من البنيات المعلوماتية المتقدمة، عن البنيات الموجهة خدماتياً SOA أحدث.

المثال الذي سنعمل عليه، هو تطبيق لحساب الوزن المثالي للذكور والإناث، بحيث سننشئ تطبيقاً يطلب من المستخدم إدخال جنسه وطوله بالسنتيمتر، وبناء على هذه المدخلات، سيتم استدعاء دالة موجودة على Service مستضاف على سيرفر، فتستقبل هاته الدالة القيم المرسله وتحسب الوزن المثالي.

قبل أن نباشر العمل، ينبغي أن نعلم أن صيغة حساب الوزن المثالي للأشخاص يكون على الشكل التالي:

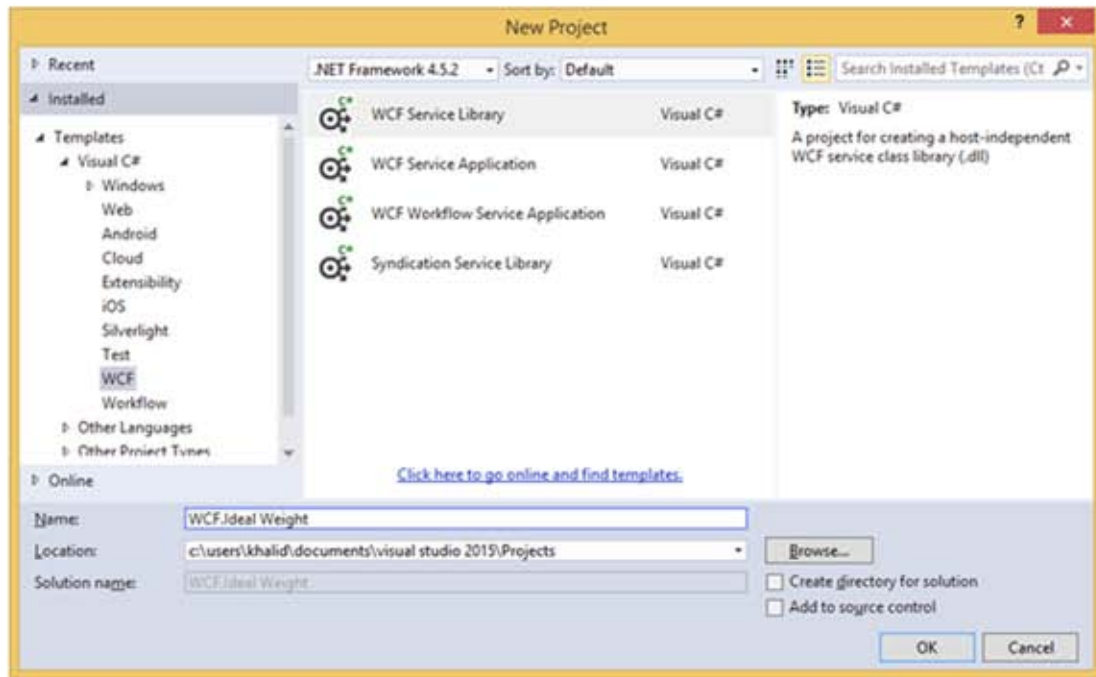
1. إذا كان الجنس ذكراً، فإن الوزن المثالي يساوي الطول ناقص 100 في 0.90 أي

$$\text{Ideal Weight} = (\text{Height} - 100) * 0.90$$

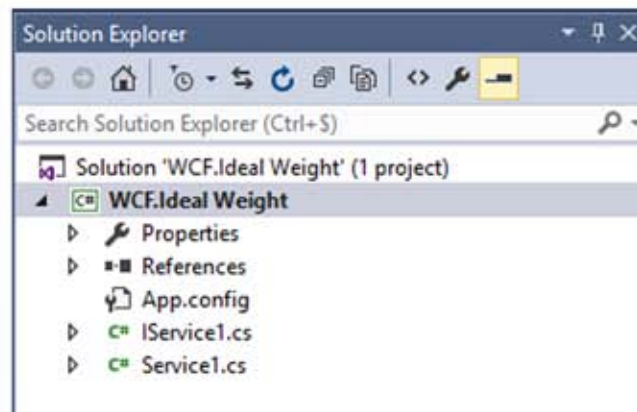
2. إذا كان الجنس أنثى، فإن الوزن المثالي يساوي الطول ناقص 100 في 0.85 أي

$$\text{Ideal Weight} = (\text{Height} - 100) * 0.85$$

تعالوا بنا ننشئ مشروعاً جديداً على الفيجوال ستوديو من نوع WCF Service Library. ولنقم بإعطائه الاسم WCF.Ideal Weight مثلاً:



بعد أن يتم إنشاء المشروع ستلاحظون ظهور ملف الواجهة `IService1.cs`. وملف الخدمة `Service1.cs` التي ستقوم بتطبيق عناصر الواجهة. وقد تقدم شرحهما في الجزء السابق من السلسلة.



الآن. سنأتي إلى ملف الواجهة Service1.cs. وسنقوم بإزالة كل ما فيه من تعريفات للوظائف. وسنقوم بتعريف الوظيفة التي ستقوم بحساب الوزن المثالي. وصيغتها كما يلي:

```
[OperationContract]
string IdealWeight(double height, char sex);
```

الوظيفة كما تلاحظون. تستقبل برامتين، أولهما عشري double من أجل استقبال الطول. والثاني حرفي char من أجل استقبال الجنس إما (m) male أو (f) female. وفيما يلي المحتوى الكامل للواجهة Service1.cs:

```
using System.ServiceModel;
```

```
namespace WCF.Ideal_Weight
```

```
{
    [ServiceContract]
    public interface IService1
    {
        [OperationContract]
        string IdealWeight(double height, char sex);
    }
}
```

بعد ذلك، سندخل إلى ملف Service1.cs. ونقوم بإزالة ما به من وظائف قديمة ناجمة عن التعريفات السابقة داخل الواجهة. وسنقوم بكتابة الكود الخاص بالوظيفة التي قمنا بتعريفها على الشكل التالي:

```
public string IdealWeight(double height, char sex)
{
    double Weight = sex == 'm' ? (height - 100) * 0.90: (height - 100) * 0.85;

    return string.Format("The ideal Weight is: {0} - This function is called from a WCF Service", Weight);
}
```

في الدالة IdealWeight قمنا بحساب الوزن المثالي بناء على نوع الجنس والطول. أما بشأن السطر التالي:

```
double Weight = sex == 'm' ? (height - 100) * 0.90: (height - 100) * 0.85;
```

فقد استخدمنا الرابط الشرطي الثلاثي. ويمكننا استبداله بالبنية الشرطية العادية باستخدام if كما يلي:

```
double Weight;
if (sex == 'm')
    Weight = (height - 100) * 0.90;
else Weight = (height - 100) * 0.85;
```

لتصبح الدالة على الشكل التالي عند استخدام الكلمة if:

```
public string IdealWeight(double height, char sex)
{
    double Weight;
    if (sex == 'm')
        Weight = (height - 100) * 0.90;
    else Weight = (height - 100) * 0.85;

    return string.Format("The ideal Weight is: {0} - This function is called from a WCF
Service", Weight);
}
```

أو على الشكل التالي. في حال استخدمنا الرابط الثلاثي:

```
public class Service1 : IService1
{
    public string IdealWeight(double height, char sex)
    {
        double Weight = sex == 'm' ? (height - 100) * 0.90: (height - 100) * 0.85;
        return string.Format("The ideal Weight is: {0} - This function is called from a WCF
Service", Weight);
    }
}
```



بهذه الكيفية أحبتي في الله، نكون قد انتهينا من إنشاء الواجهة، وملف الخدمة، وفيما يلي الكود الشامل لكل منهما:  
الكود الخاص بالواجهة :Service1.cs

```
using System.ServiceModel;

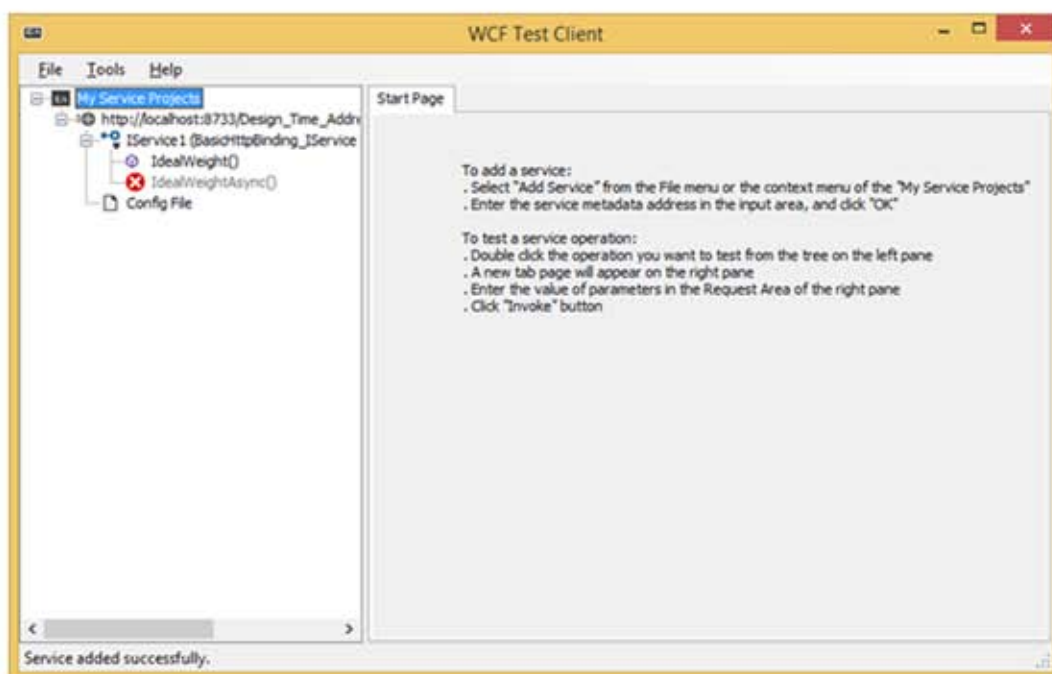
namespace WCF.Ideal_Weight
{
    [ServiceContract]
    public interface IService1
    {
        [OperationContract]
        string IdealWeight(double height, char sex);
    }
}
```

الكود الخاص بالخدمة :Service1.cs

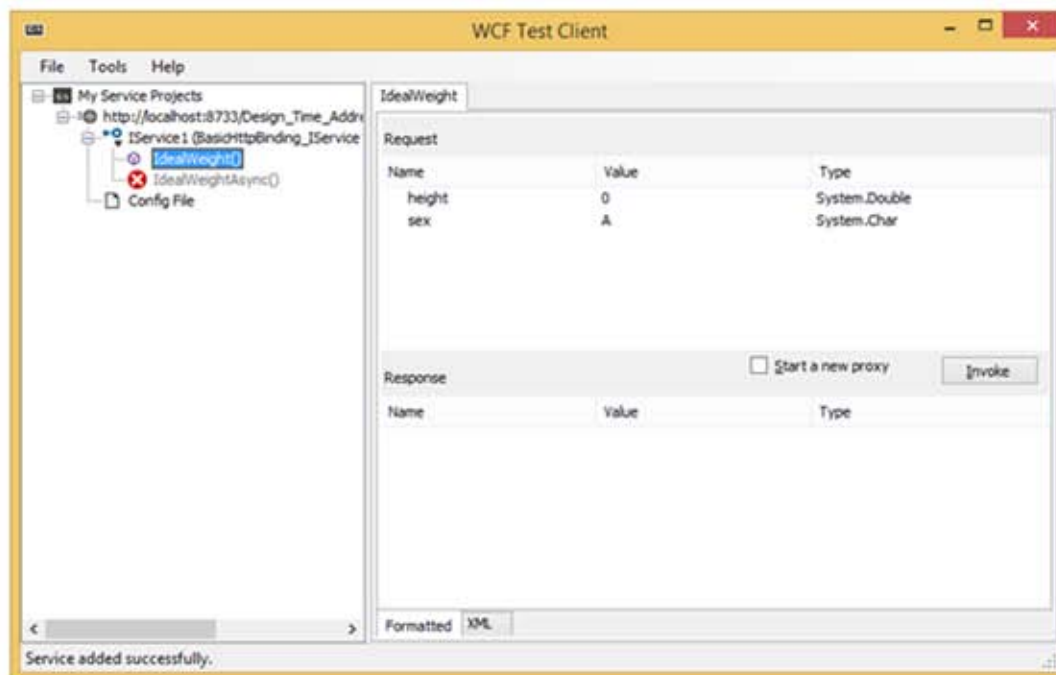
```
namespace WCF.Ideal_Weight
{
    public class Service1 : IService1
    {
        public string IdealWeight(double height, char sex)
        {
            double Weight = sex == 'm' ? (height - 100) * 0.90: (height - 100) * 0.85;

            return string.Format("The ideal Weight is: {0} - This function is called from a WCF Service", Weight);
        }
    }
}
```

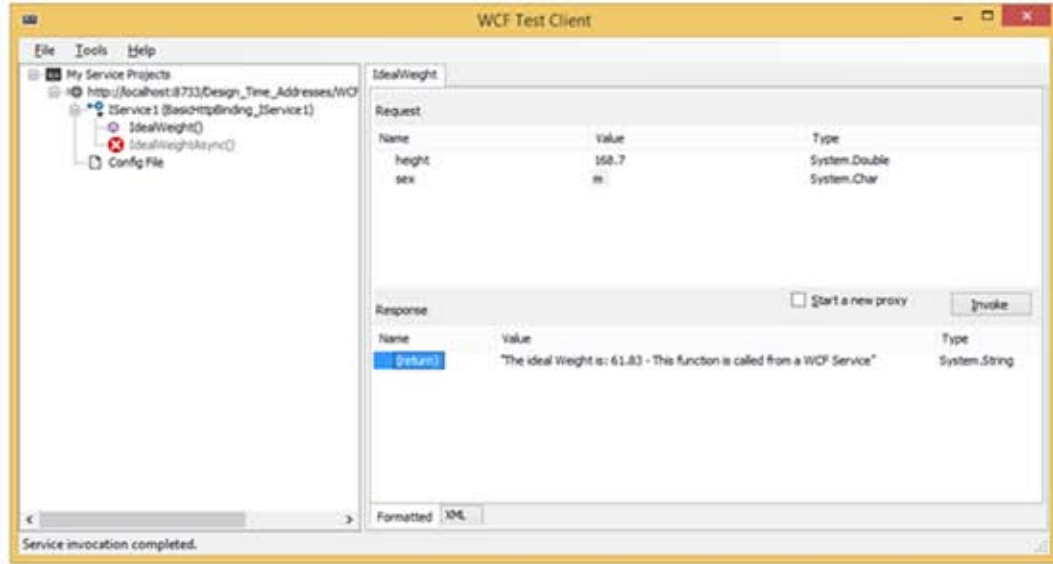
لنقم الآن بالضغط على F5 لاختبار الخدمة، ستظهر لنا واجهة WCF Test Client التالية:



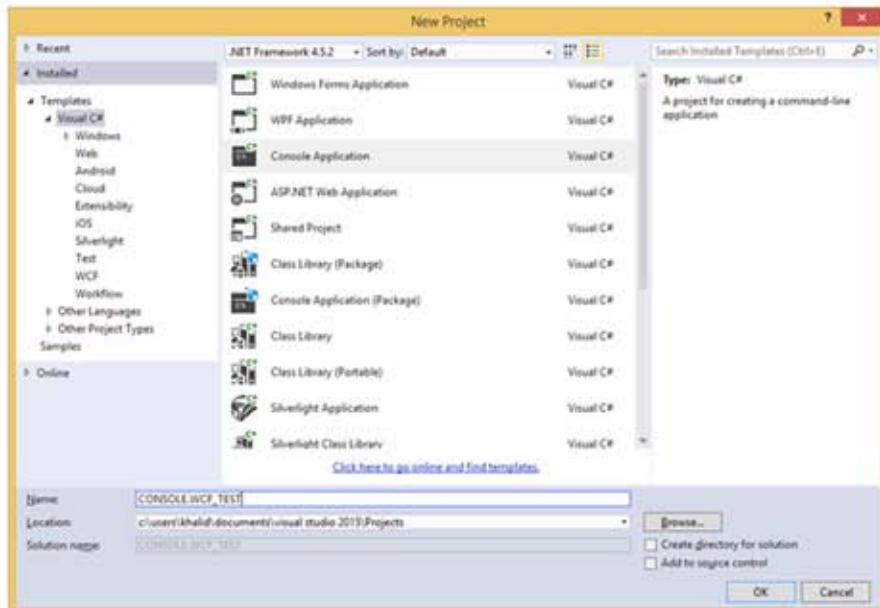
لو ضغطنا مرتين على اسم الدالة IdealWeight فسوف تظهر لنا الصفحة على الشكل التالي:



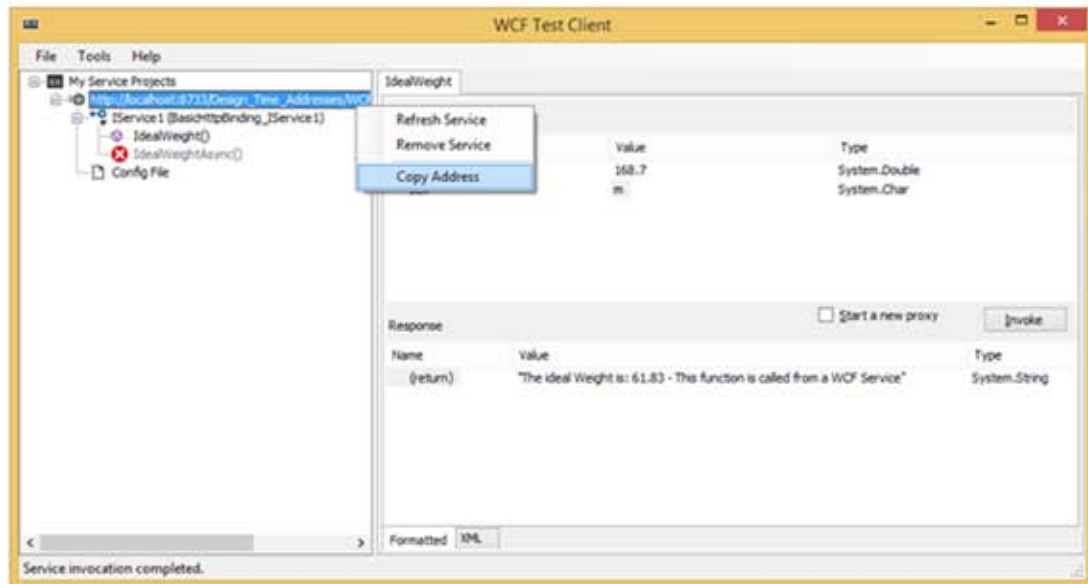
يمكننا أن نجرب الدالة من هنا، لنقم بإدخال قيمة عشرية للطول، وقيمة حرفية للجنس ونضغط على الزر Invoke. ستلاحظون ظهور النتيجة التالية:



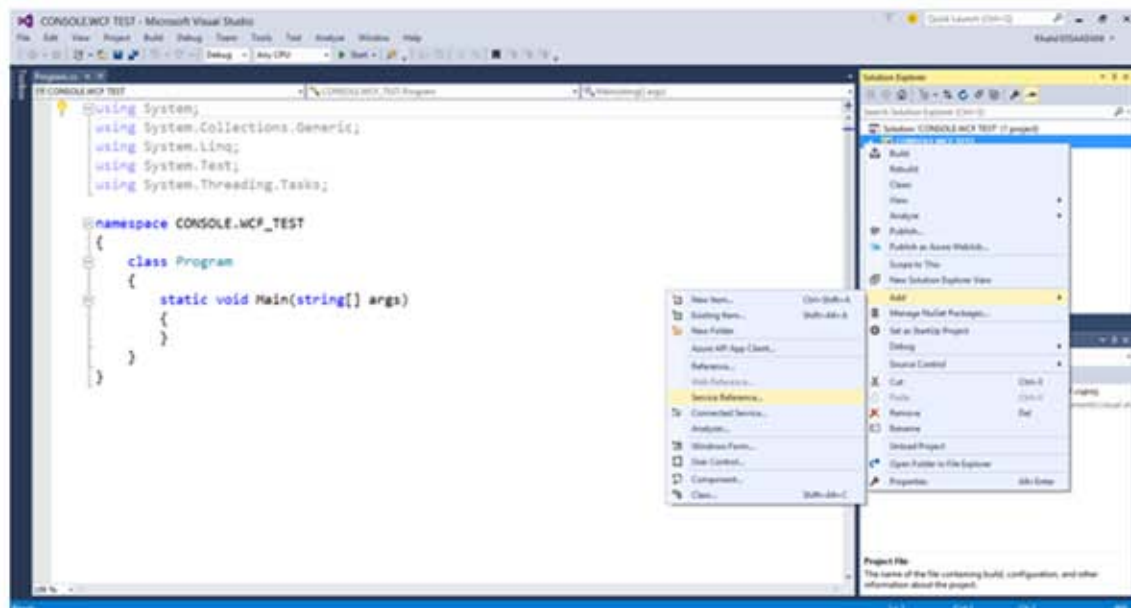
أدخلنا في الطول القيمة العشرية 168.7 سنتمتر، وفي الجنس الحرف m، فتم إرسالهما إلى الدالة وقامت بحسابهما وتم إرجاع النتيجة الظاهرة في الأسفل. إذن ولله الحمد، التطبيق يعمل بنجاح، يمكننا اختباره من مكان آخر، سننشئ مشروعاً آخر بمعزل عن هذا المشروع، وسنقوم بالاستفادة من هذه الدالة التي يقدمها هذا السيرفيس. تعالوا بنا نفتح الفيچوال ستوديو وننشئ مشروعاً جديداً، من أي نوع نريد سواء Windows Forms Application، أو ConsoleApplication، وليكن مثلاً من نوع Console لكي نركز في الكود فقط:



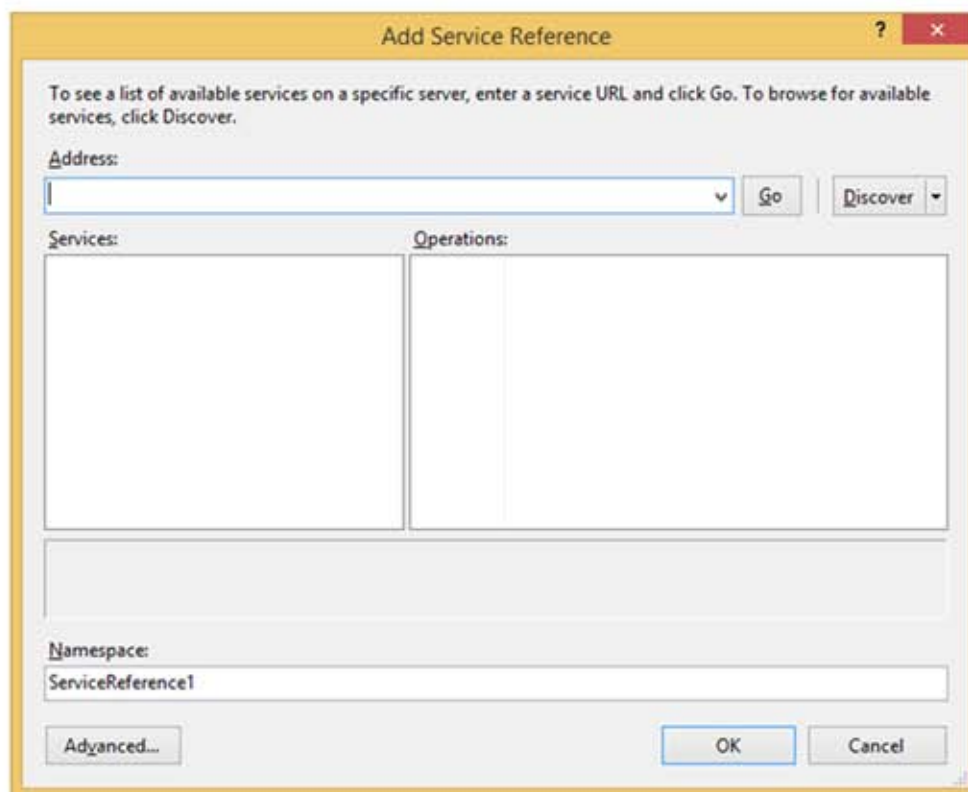
بعد أن يتم إنشاء المشروع، سنقوم بربطه مع رابط الخدمة، كيف نحصل على هذا الرابط، المسألة بسيطة وقد رأينا ذلك في الجزء المنصرم، إذ يكفي أن نعود إلى تطبيق WCF Test Client، ونقوم بنسخ الرابط كما يلي:



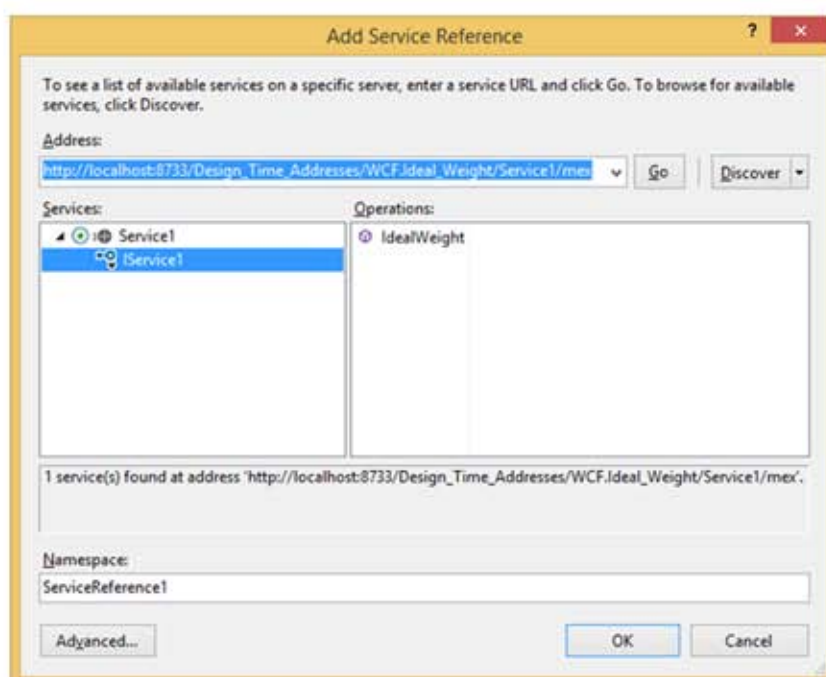
ثم نعود إلى المشروع الذي أنشأناه في Console، ونضغط بيمين الفأرة على المشروع ونقوم باختيار الأمر Service Reference كما توضح الصورة أدناه:



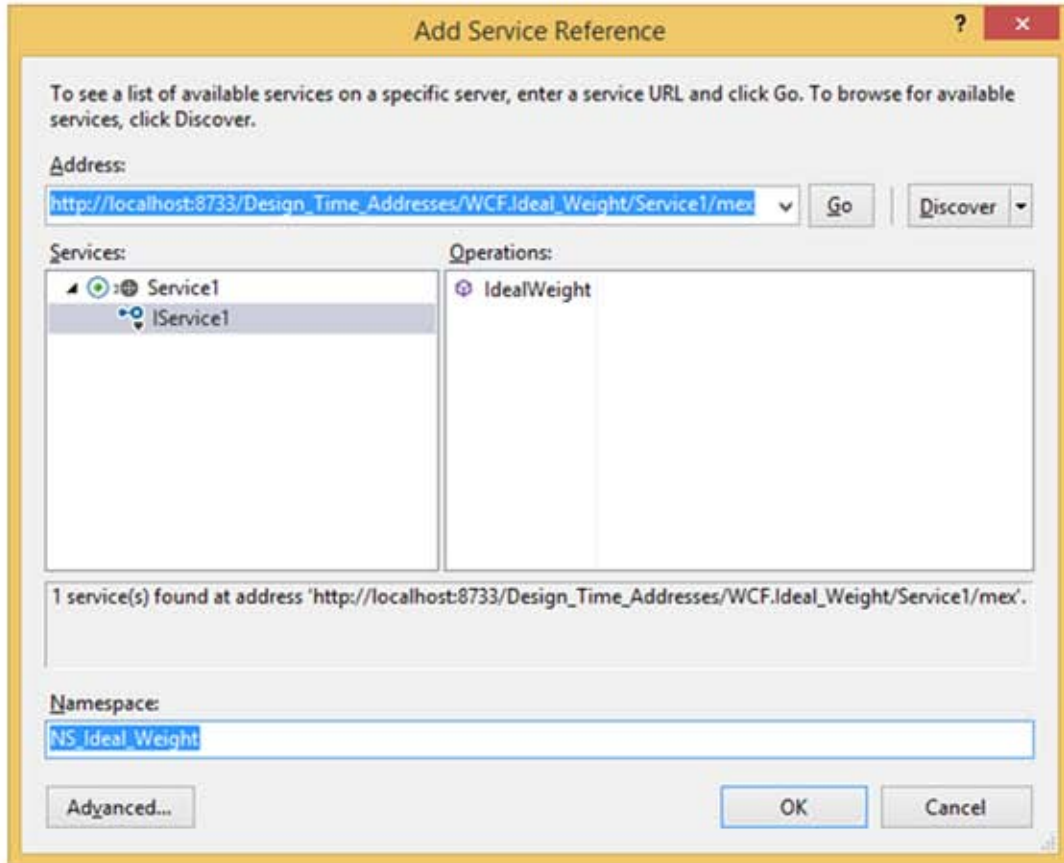
لتطالعنا الواجهة التالية:



في مربع النص Address سنقوم بلصق رابط الوصول إلى Service ونضغط على الزر Go ليتم البحث عن Service ثم جلب كافة وظائفه كما يلي:



يمكننا أن نغير اسم مجال الأسماء Namespace من القيمة ServiceReference1 إلى اسم آخر يكون أكثر وضوحا وليكن مثلا NS\_Ideal\_Weight:



بعد ذلك، يمكننا استخدام عناصر الخدمة بكل سهولة، لكن يشترط في السيرفر الذي يستضيف الخدمة أن يكون شغالا إذا أردنا أن يتم عمل الوظائف بنجاح وإلا فإن العملية ستفشل. إذ لا يمكن استدعاء وظائف موجودة على سيرفيس مستضاف على سيرفر غير شغال.

هكذا نكون قد قطعنا الأشواط المهمة، بقي علينا فقط أن نأتي وننشئ برنامجا من خلاله نستدعي الدالة IdelaWeight، ونقوم بتجربتها، لنأت إذن إلى محرر الكود ولنكتب الشفرة التالية:

---

```
using System;

namespace CONSOLE.WCF_TEST
{
    class Program
    {
        static void Main(string[] args)
        {
            double height;
            char sex;

            Console.WriteLine("Enter your height (cm): ");
            height = double.Parse(Console.ReadLine());

            Console.WriteLine("Enter your sex (m / f): ");
            sex = char.Parse(Console.ReadLine());

            NS_Ideal_Weight.Service1Client service = new NS_Ideal_Weight.Service1Client();

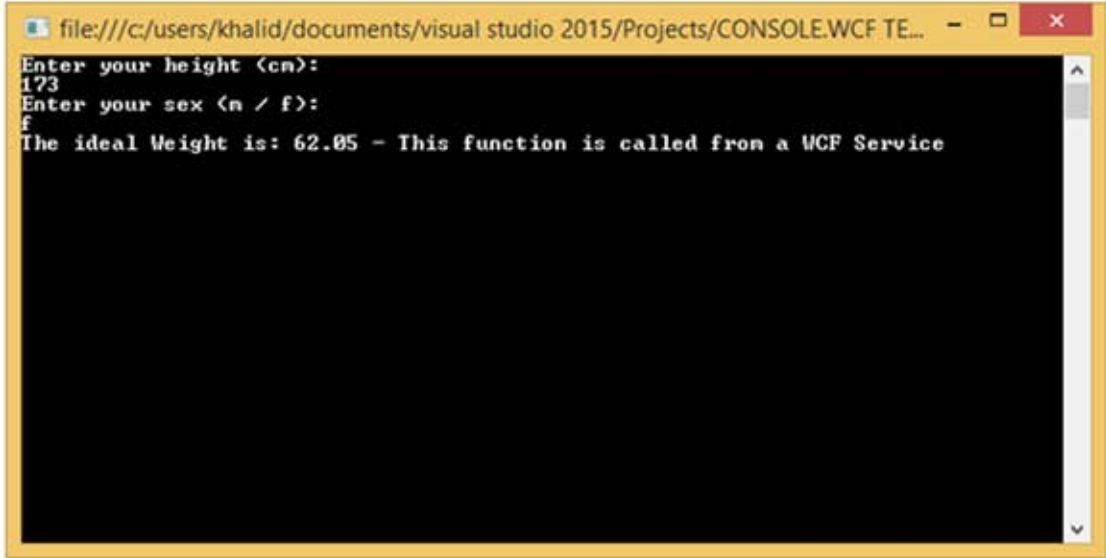
            string result = service.IdealWeight(height, sex);

            Console.WriteLine(result);

            Console.Read();
        }
    }
}
```

في المثال أعلاه، طلبنا من المستخدم إدخال طوله وجنسه، ثم استدعينا الدالية IdealWeight الموجودة في الكلاس Service1 والتي توجد بدورها في مجال الأسماء NS\_Ideal\_Weight.

عند التنفيذ سنحصل على النتيجة التالية:

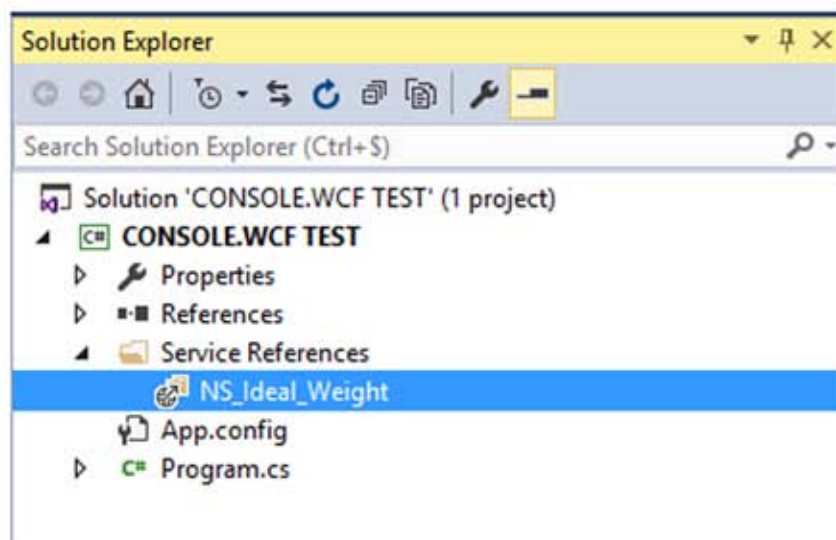


```

file:///c:/users/khalid/documents/visual studio 2015/Projects/CONSOLE.WCF TE...
Enter your height <cn>:
173
Enter your sex <n / f>:
f
The ideal Weight is: 62.05 - This function is called from a WCF Service

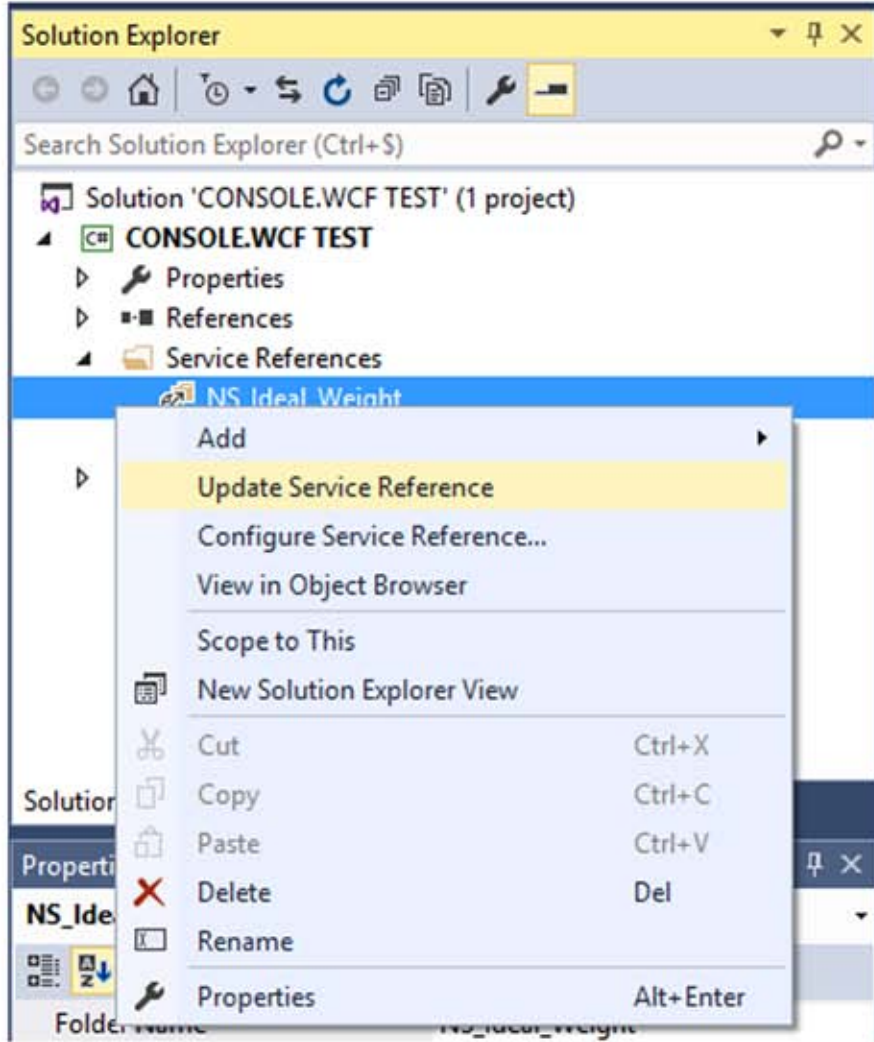
```

وهذا يعني أن الاتصال بالسيرفيس تم بنجاح. إذا قمنا بالتعديل على ملف الخدمة Service وأردنا تحديث العناصر في المشروع المستفيد، يكفي أن نأتي إلى مجلد الخدمة الناتج عن عملية الربط الظاهر في الصورة أسفله:





ثم نضغط عليه بيمين الماوس ونختار الأمر Update Service Reference الذي سيقوم بعملية التحديث:



إلى هنا أحبتي في الله نكون قد انتهينا من سلسلة "ثورة الخدمات – WCF نموذجاً". على أمل أن أكون قد وفقت في شرح هذه التقنية. أدعو لكم بالتوفيق والسداد ودام لكم البشر والفرح.

- خالد السعداني

قصة ظهور  
قاصمة الظهر  
(البرمجة الكائنية  
التوجه)

oop

الجزء الثالث

# قصة ظهور قاصمة الظهور

## البرمجة الكائنية التوجه

## Object Oriented Programming

خالد السعداني

في الجزء السابق من سلسلة "قصة ظهور قاصمة الظهور"، تعرضنا إلى أساسيات البرمجة الكائنية التوجه، حيث تعرفنا على كيفية إنشاء الفئات Creating Classes واستنساخ الكائنات منها Instantiating Objects. ومررنا عبر شرح محددات الوصول Access Modifiers والفروق الماثلة فيما بينها، ثم ختمنا بالشرح المفصل لمفهوم الخصائص Properties بقسميها Getter و Setter. في حصة اليوم بحول الله، سوف نتعرف على مفهوم الوظائف Methods، وكذلك على مفهوم آليات بناء وهدم الكائنات، أو ما يعرف بالمشيدات والمهدمات Constructors and Destructors.

### الوظائف Methods:

وهي المهام والعمليات التي تقوم بها الفئة، وتكتب على شكل إجراءات ودوال حسب الدور الذي تؤديه.

ونحن نتحدث عن فئة الحاسوب يمكننا اعتبار العمليات التي يقدمها الحاسوب، مثل الاشتغال والتوقف وإعادة الاشتغال، وتخزين ومعالجة البيانات بمثابة وظائف Methods.

Class CLS\_COMPUTER

'Fields / Attributes

Private Model As String

Private Price As Decimal

```
Private Date_of_Manufacture As Date
```

```
'Properties الخصائص
```

```
Public Property Model_Property As String
```

```
Get
```

```
Return Model
```

```
End Get
```

```
Set(value As String)
```

```
Model = value
```

```
End Set
```

```
End Property
```

```
Public Property Price_Property As Decimal
```

```
Get
```

```
Return Price
```

```
End Get
```

```
Set(value As Decimal)
```

```
Price = value
```

```
End Set
```

```
End Property
```

```
Public Property Date_of_Manufacture_Property As Date
```

```
Get
```

```
Return Date_of_Manufacture
```

```
End Get
```

```
Set(value As Date)
```

```
Date_of_Manufacture = value
```

```
End Set
```

```
End Property
```

'الوظائف Methods

Public Sub Run()

'Do something

End Sub

Public Sub ShutDown()

'Do something

End Sub

Public Sub Restart()

'Do something

End Sub

Public Function ProcessData() As DataType

'Return something

End Function

Public Sub StoreData()

'Do something

End Sub

End Class

يمكننا استدعاء الوظائف بنفس طريقة استدعاء الخصائص. بحيث نستطيع الوصول إليها من خلال اسم الكائن المستنسخ من الفئة. كما يلي:

'Instantiate new object from Computer Class

Dim Computer As New CLS\_COMPUTER

'Calling Class Methods

Computer.Run()

Computer.StoreData()

Dim Result = Computer.ProcessData()

Computer.Restart()

Computer.ShutDown()

## المشيدات Constructors:

المشيدات هي إجراءات نكتبها داخل الفئة من أجل تشييد الكائن عند عملية استنساخه عبر إسناد قيم بدئية له، أو القيام بعمليات معينة أثناء عملية استنساخ الكائن. لأنها أول ما ينفذ عند إنشاء الكائن.

لإنشاء مشيد في لغة الفيجوال بيسك، يكفي أن نكتب الكلمة New () بعد صيغة إنشاء الإجراءات التي هي Sub كما يلي:

```
Sub New()
```

```
End Sub
```

في المثال أعلاه أنشأنا مشيدا بالصيغة الافتراضية. بمعنى أنه حتى لو لم نقم بكتابته بذلك الشكل فلا مشكلة لأنه موجود ضمنا.

للإعلان عن كائن من الفئة التي تحتوي على مشيد بهذه الصيغة الافتراضية نكتب ما يلي:

```
Dim myComputer As New CLS_COMPUTER()
```

الأقواس فارغة في نهاية اسم الفئة تدل على أن المشيد لا ينتظر أية برامترات. يمكننا إنشاء مشيدات حسب حاجتنا، كأن نقوم بتمرير قيم بدئية لحقول الفئة عند استنساخ كائن منها كما يعرض لنا المثال التالي:

### المشيدات Constructors'

```
Sub New(_model As String, _price As String, _Date_of_Manufacture As Date)
```

```
Model = _model
```

```
Price = _price
```

```
Date_of_Manufacture = _Date_of_Manufacture
```

```
End Sub
```

للتمييز بين حقول الفئة وبين البرامترات الوافدة للمشيد نقوم بتغييرات طفيفة كأن نضيف رمز الأندرسكور ( \_ ) أمام اسم البرامتر كما في الحالة أعلاه.

بالنسبة لحقول الفئة وجميع عناصرها من خصائص ووظائف فيمكننا كتابة الكلمة Me ثم نقطة بعدها وستظهر لنا كلها ونختار منها ما نشاء. كما يوضح لنا المثال التالي الذي يؤدي نفس دور المثال السابق:

### المشيدات Constructors

```
Sub New(_model As String, _price As String, _Date_of_Manufacture As Date)
    Me.Model = _model
    Me.Price = _price
    Me.Date_of_Manufacture = _Date_of_Manufacture
End Sub
```

الآن أصبح بإمكاننا إسناد قيم بدئية عند استنساخ أي كائن من الفئة CLS\_COMPUTER بحيث سيطلب منا إدخال قيم البرامترات كما تعرض الصورة التالية:



عند إنشاء أي كائن جديد. المفروض أن يكون على النسق التالي:

```
((Dim myComputer As New CLS_COMPUTER("Compaq", 888.6, New Date(2014, 08, 16
```

يمكنك إنشاء المشيدات بالعدد الذي تريد. احرص فقط أن تكون البنية Signature مختلفة. والمقصود بالبنية أن تكون البرامترات مختلفة في كل المشيدات إما من حيث عددها. أو من حيث نوع بياناتها. أو من حيث ترتيبها. فكل هذه الأشكال من المشيدات مقبولة:

### المشيد الأول'

```
Sub New(_model As String, _price As String, _Date_of_Manufacture As Date)
    Me.Model = _model
    Me.Price = _price
    Me.Date_of_Manufacture = _Date_of_Manufacture
End Sub
```

## المشيد الثاني'

Sub New()

Me.Model = "Undefined"

Me.Price = 00.0

Me.Date\_of\_Manufacture = Date.Now

End Sub

## المشيد الثالث'

Sub New(ByVal computer As CLS\_COMPUTER)

Me.Model = computer.Model

Me.Price = computer.Price

Me.Date\_of\_Manufacture = computer.Date\_of\_Manufacture

End Sub

المشيد الأول يقوم باستقبال مجموعة من البرامترات ويسندها إلى حقول الفئة. بينما يقوم المشيد الثاني بإسناد قيم بدئية دون الحاجة إلى استخدام برامترات بحيث القيم محددة مسبقا. أما المشيد الثالث والأخير فإنه يقوم باستقبال كائن من نفس نوع الفئة ومن خلال بياناته سيتم إنشاء الكائن الجديد.

الآن تعال بنا نرى مثلا على إنشاء ثلاث كائنات من هذه الفئة باستخدام هذه المشيدات الثلاثة:

## نموذج على استخدام المشيد الأول'

Dim Computer1 As New CLS\_COMPUTER("Toshiba", 755.1, New Date(2012, 07, 27))

في المثال أعلاه سيتم إسناد القيم الممررة إلى حقول الكائن Computer1.

## نموذج على استخدام المشيد الثاني'

Dim Computer2 As New CLS\_COMPUTER()

في المثال أعلاه سيتم إسناد القيم الأساسية التي ذكرناها داخل المشيد إلى حقول الكائن Computer2.



نموذج على استخدام المشيد الثالث'

```
Dim Computer1 As New CLS_COMPUTER("Toshiba", 755.1, New Date(2012, 07, 27))
```

```
Dim Computer3 As New CLS_COMPUTER(Computer1)
```

في المثال أعلاه، سيتم بناء الكائن Computer3 بنفس البيانات التي تم إسنادها للكائن Computer1 لأننا مررناه على شكل برامتر إليه.

### ملحوظة مهمة:

إذا أنشأت مشيدا واحدا فقط داخل فئة معينة له برامترات، فعند كتابتك لهذا الاستنساخ سيعتبر خطأ:

```
Dim myComputer As New CLS_COMPUTER()
```

لأن عليك إنشاء مشيد فارغ ليسمح بهذا النوع من الاستنساخ، أي عليك إضافة مشيد داخل الفئة كما يلي بالضبط:

```
Sub New()
```

```
End Sub
```

## المهدمات :Destructors

المهدمات تقوم بنقيض الدور الذي تقوم به المشيدات. فإن كانت هذه الأخيرة تقوم ببناء الكائن أثناء عملية الاستنساخ من الفئة. فإن المهدمات تقوم بإنهاء الكائن وختم مهامه عبر تحرير الذاكرة الرئيسية منه أثناء خروج البرنامج من مجال الوصول Scope إلى هذا الكائن.

علما أننا لسنا مطالبين بكتابة أي كود من أجل تأليف المهدمات لأن دورها أصلا منوط بآلية التنظيف التلقائي Automatic Garbage Collector التي توفرها CLR (Common Language Run-time). هذه الأخيرة التي تعتبر مثل الآلة الافتراضية التي نشغل عليها برامج الدوت نيت. فكلما انتهى كائن ما تقوم CLR بتحرير الذاكرة منه عبر آلية التنظيف التلقائي GC.

لكن إن أردنا أن ننفذ كودا معيناً عند انتهاء مهام أي كائن. كأن نقطع اتصال. أو نخطر كائنات أخرى بأن الكائن لم يعد موجوداً على الذاكرة. أو غير ذلك من العمليات. فيمكننا أن نعيد تعريف الإجراء Finalize () الذي يمكننا من افتناص عملية إنهاء الكائن ونكتب فيه ما نشاء من كود كما يلي:

```
Protected Overrides Sub Finalize()
```

```
    MsgBox("تم إنهاء الكائن")
```

```
End Sub
```

# أمير على عرش الجائشة أو نرجسي واندثر

بُشْرَاكُمْ إِن أَنَا أُدْخَلْتُمْ فِي الْمَقَارِنَةِ،  
فَعَزَّ الذَّلِيلُ التَّفَاتَةَ مِنْ عَزِيزٍ، هَكَذَا  
أَرَانِي وَأَرَاكُمْ، وَدُونَنَا عَشْرَةَ أَفْجَاجٍ  
عَمِيقَةٍ يَتَسَفَّلُهَا مَحِيطٌ لَا قَاعَ لَهُ، فَمَنْ  
دَنَا مِنْ شَمْوُخِي تَلَقَّفَتْهُ الْهَوَّةُ فَسَقَطَ  
مِنْ غَيْرِ رَجْعَةٍ..  
سَقُوطٍ مَحِيرٍ..

سَتَهْلِكُونَ فِي فِرَاقِ الْهَوَّةِ وَحَلْمِ  
الْإِرْتِطَامِ بِالْقَاعِ يَخَامِرْكُمْ..

# أمير على عرش نفسي

خالد السعداني

## أو نرجسي واندثر

لم يكن الجو جميلاً يومها ولا وجوه البشر. سحب مُكْرِهَمَةٌ ووجوه مُكْفَهَرَةٌ. سيطر علي العبوس بدوري وأنا أخطو بساقي النحيلتين في شارع الأمل. عن أي أمل تتحدثون؟ على الأقل إذا أردتم أن تكذبوا فأحسنوا الكذبة !  
كنت أنفُسُ عن عبوسي بصوت كَسْرِ الصقيع وأنا أدوسه. شعور نبيل فعلا. على الأقل أفضل من كسر الخواطر.

كان صديقي "المامون" يتمشى إلى جانبي وهو يغمغم مرددا أغنية غامضة لم أتبين ألفاظها. كان من حين إلى آخر يسرق تركيزي لأرخي بسمعي إليه فأعود خائبا. فلا الأغنية عرفتها ولا غمغماته تطربني.

بين كسر الصقيع وغمغمات المامون. تراءى لي مشهد زلزل أركانني..  
إنه صديقي القديم "سليمان" يتوسد الرصيف في عز البرد القارس. فركت عيني في غير تصديق ثم أوقفت المامون بغلظة. لم أنتبه إلى ردة فعله لأن تركيزي كله قد سُلبَ وأنا أنظر إلى حال "سليمان" المأساوي.

كان يبعد عنا ببضعة أمتار. لا يفصل بيننا سوى الشارع. نحن وُقُوفٌ أمام المقهى وهو نائم على الرصيف المقابل..

أهذا هو نفسه ذلك الفتى الوسيم الذي كان يغضب أشد الغضب إذا عبث أحد بخصلات شعره؟  
أهذا هو "سليمان" الذي حير الفصل بذكائه ونباهته؟

أهذا هو سليمان ملك النرجسية وسلطان حب الذات. الذي قهرنا بمدح نفسه في كل مناسبة؟  
المشهد مؤلم صراحة. لقد أخبرني بعض الأصدقاء قبل مدة أنه فقد عقله. لكن لم أكن أتوقع أن أراه بهذه الحال. سبحان مبدل الأحوال !

سألني "المامون" وهو يتابع نظراتي:

- هل تعرف هذا المجنون؟

رمقته بفضفاضة وهتفت به:

- لا تتكلم بهذا الأسلوب. فهذا صديق قديم!

أطرق المامون برأسه لهنيهات. ثم عاد يسألني:

- وما الذي حدث له يا ترى؟

فأجبت به بكل ثقة:

لقد فقد عقله بسبب النرجسية وحب الذات. هذا ما أخبرني به من أبلغوني بنبا جنونه. أتدري يا

"المامون" أن هذا المسكين كان لا يكف أبدا عن سب الناس وهرمطة أعراضهم و التقليل من شأنهم؟ أ

تدري أنه كان يتحدث عن نفسه بإطراء ومبالغة؟

كان عاشقا لنفسه. وبقدر عشقه لها. كان كارها للناس...

هذا المسكين ضحية النرجسية. أذكر في إحدى المرات. أنني وجدت قصاصة ورق كان قد كتب عليها

بعض خواطره. فلما أخذتها لأقرأها وجدتها عامرة بعبارات تقديس الذات وطافحة بالتنقيص والطعن

في الناس.

القصاصة ما تزال عندي وقد حفظتها من كثرة ما أعدت قراءتها لأن أسلوبها أعجبني. وإن كان

محتواها لا يروق إلا لمرضى نفسي. ثم بدأت أتلو كلمات صديقي النرجسي على المامون:

"سأنطوي على نفسي ففيها من الخير ما لا يدركه سواي...

سأهوءُ بها وأسمو بها إلى المعالي حتى لا تطالها خديعة أدمي..

ألم أختلط بالناس صالحهم وطالحهم؟ ألم أعد من مخالطتهم بوجع الرأس إلى حد السهاد والأرق؟

فَلَاكُفَّ إِذْن. فلست صبيا أرضع أصابعي. أو أحمقا يجوب الشوارع عريانا..

إنني وإن أبى الناس فهمي لأمير على عرش نفسي..

أمير على عرش الجائشة..

وحدني من يُقدِّرني ويعرف كُنْهِي وجوهري. أما انطباعات الناس عني فإنها أهون عندي من قَعْيِيسٍ على

عمته. لا تعرفون من هو قعيس؟ سأخبركم عنه. سَأَمْدُكُمُ بالمعرفة من بطون الكتب أما من من

أفواهكم فلتحتفظوا بها عندكم. فلست في حاجة إليها.

إن قَعْيِيسًا يا جماعة. صبي يتيم. تبنته عمته. وترعرع في كوخها الصغير. وكان لهذه العمه اللئيمة

كلب. فحدث أن هطل المطر تَهْطَاطًا في إحدى الليالي. فَعَزَّ في نفسها أن يبتل الكلب. و كذلك الكلاب

ترأف على بعضها. فأدخلته كوخها وأخرجت قعيسا. ليجده الناس صباح الغد قد مات متجمدا.

هوان قعيس على عمته دون هوان انطباعاتكم عني..

لقد جريت الحب القادم منكم فاستحليته كالعسل. فإذا آخره مرارة دونها مرارة العلقم..

أنتم هكذا دائما. لا تحبون إلا لمصلحة. وعليها تَوَادُّونَ وَحَادُّونَ..

فإن أنا جَارِيَتُكُمْ كنت أشبه بكم. وما لذلك حبيبت ولا لأجله أحيي..

أنا زهرة الريحان التي فاح عبقها فما استروحته أنوف الجيف، وكيف لها أن تستروح طيب عبقها وقد  
زكمت أنوفها رائحة الموت.

نرجسي أنا إلى حد حبي لنفسي بقدر كرهى لكم،  
بُشْرَاكُمْ إِن أَنَا أَدْخَلْتَكُمْ فِي الْمَقَارِنَةِ، فَعِزُّ الدَّلِيلِ الْتَفَاتَةٌ مِنْ عَزِيْزٍ هَكَذَا أَرَانِي وَأَرَاكُمْ، وَدُونَا عَشْرَةٌ  
أَفْجَاجٌ عَمِيْقَةٌ يَنْسَفِلُهَا مُحِيْطٌ لَا قَاعَ لَهُ، فَمَنْ دَنَا مِنْ شَمْوَخِي تَلَقَّفَتْهُ الْهُوَّةُ فَسَقَطَ مِنْ غَيْرِ رَجْعَةٍ..  
سَقُوْطٌ مَحِيْرٌ..

ستهلكون في فراغ الهوة وحلم الارتطام بالقاع يخامركم..

وأنى لكم ذلك وأنا أنا، وأنتم أنتم!"

ما كدت أكمل الكلمات حتى وجدت المامون يرمقني بنظرات شزراوات، وكأنه يقول لي:

وما العجب، لقد كان صديقك مجنوناً بشدة، فقلت وثيرة جنونه..

أن يجوب الشوارع فاقدًا لعقله أهون عليه من أن يعشق نفسه إلى حد الجنون..

على الأقل، أن يكسر الصقيع أفضل من أن يكسر قلوب الناس!

# المختصر النافع في لغة XML

## أسباب الظهور ودواعي وضوابط الاستعمال

خالد السعداني

كما هو معلوم فهناك ثلاثة أصناف من اللغات في المعلومات. الصنف الأول وهو لغات البرمجة Programming Languages وهي لغات نستخدمها من أجل إنشاء برامج مكتبية Desktop Programs. تطبيقات هواتف Mobile Applications. مواقع ديناميكية Dynamic Websites. ألعاب Games. و هلم جرا. مثل جافا. سي بلس بلس. سي شارب. فيجوال بيسيك... أيضا هنالك صنف ثان يسمى لغات الاستعلام Query Language. وهي لغات نستخدمها للتواصل مع حاويات البيانات. ونقصد بحاويات البيانات كل بنية تمكننا من تخزين البيانات بشكل منظم. ولنأخذ على سبيل المثال لغة SQL التي نستخدمها للتواصل مع قواعد البيانات العلائقية / الترابطية. وأيضا هنالك مثال آخر: لغة xQuery التي تستخدم للتواصل مع البيانات المخزنة في ملفات XML. وهنالك صنف ثالث من اللغات في الإعلاميات: وهو اللغات الوصفية التي

### تقديم عام:

اليوم في عالم البرمجة من النادر جدا أن تجد شخصا لم يسمع قط بكلمة XML. فهي صارت شديدة الذيوع والانتشار مع مختلف الأنظمة والتطبيقات والبرامج. لكن مع شهرتها إلا أنها تبقى غير مفهومة بشكل جيد عند العديد من الناس.

بداية ينبغي أن نعلم أن XML هي اختصار ل Extensible Markup Language وهي لغة قوية جدا تعتمد على أسلوب الأوسمة التشعبية Generic Tags (مثل لغة HTML) وتمكننا من تنظيم البيانات وتنسيقها لتسهيل قراءتها من قبل الإنسان العادي ولتسهيل قراءتها أيضا من قبل البرامج.

نستخدمها فقط لعرض البيانات وفق قواعد وضوابط معينة ولعل أشهر لغات الوصف: لغة HTML التي نستخدمها لهيكله صفحات الويب، وأيضا لغة XML التي نستخدمها لتخزين البيانات بشكل تراتبي وتشعبي.

### نبذة تاريخية:

في بداية عهد الانترنت، كانت الحواسيب والبرامج تتبادل البيانات فيما بينها عبر استخدام الملفات Files، وكان هذا النوع من التبادل شاق جدا لأن تخزين البيانات في الملفات يستلزم الانضباط لمجموعة من قواعد مثل فصل البيانات بواسطة عوارض أو فواصل منقوطة حسب نوع الملف، ويمكنكم الاستفاضة في طرق التخزين عبر العودة إلى الفصل الأول من كتاب "الشرح الوافي لتعلم لغة SQL من نبعها الصافي" حيث ستجدون مواصفات كل نوع من أنواع التخزين في الملفات بالإضافة إلى عيوب كل نوع.

المشكل في هذا النوع من الملفات هو أن على كل الأنظمة التي تريد استخدام البيانات المحفوظة في هذه الملفات الانضباط لقواعده، وهذه القواعد ليست موحدة، بحيث قد تجد ملفا على سبيل المثال يفصل البيانات بعارضة وآخر يفصلها بنقطة وهكذا...

تم حل هذا المشكل عبر إنتاج أول لغة امتدادية التي كانت تحمل اسم SGML وهو اختصار ل Stan-dard Generalized Markup language، وكانت لغة قوية جدا لأنها تعتمد نفس أسلوب الأوسمة في تنسيق البيانات، لكن كان عيبها هو صعوبتها وتعقيداتها التي جعلت من الصعب جدا تبادل البيانات من خلالها.

هنا اجتمع فريق من المعلوماتيين ذوي الخبرة مع لغة SGML وقاموا بإطلاق لغة جديدة تعتمد على أسس SGML لكنها سهلة الاستخدام وسهلة القراءة من طرف البرامج والأنظمة، هذه اللغة هي XML، التي تم اعتماد نسختها XML 1.0 من طرف W3C في العاشر من فبراير من عام 1998. بعد ذلك، تم إطلاق النسخة XML 1.1 في الرابع من فبراير من عام 2004، لكن تم تجاهلها وتم الاعتماد على النسخة XML 1.0 التي ما تزال هي المستخدمة حتى الوقت الذي كتبت فيه هذه الكلمات.

### الغاية من لغة XML:

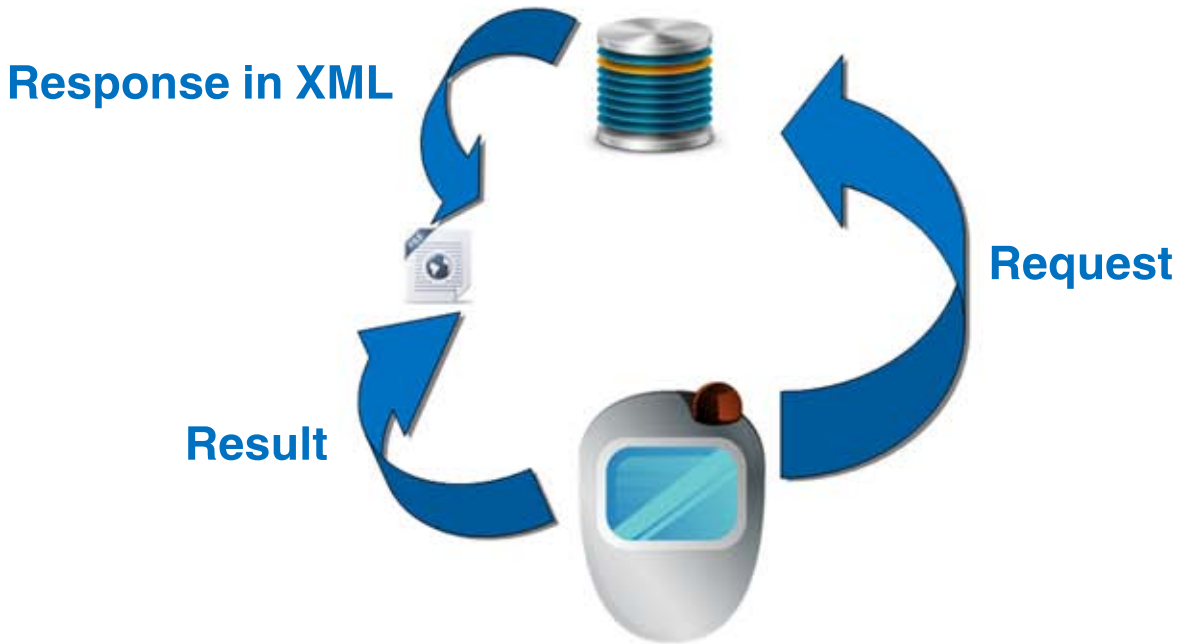
الهدف الأساسي وراء إنشاء لغة XML هو ابتكار تقنية متطورة لتبادل البيانات بين البرامج والأنظمة المختلفة.

وغالبا ما يتم استخدام XML لتبادل البيانات بين السيرفر وبين البرنامج العميل، أو بين البرامج فيما بينها.



و بعيدا عن هذا الكلام النظري، دعونا نتعرف على مثال حقيقي يوضح لنا كيف نستخدم XML لتبادل البيانات بين Server وبين Program.

هنالك بعض التطبيقات الهاتفية التي تحتاج إلى تحديث لبياناتها باستمرار، ولناخذ مثلا تطبيق الأذان الذي ينصب على الهواتف بغرض التذكير بمواقيت الأذان. هذا البرنامج يحتاج دائما إلى تحديث الأوقات فلا يعقل أن يؤذن يوميا في نفس الوقت، إذن كيف يتم ذلك، يقوم التطبيق المنصب على الهاتف بإرسال استعلام للسيرفر Request بغية تنبيهه لإرسال آخر التحديثات، فيقوم السيرفر بتهيئة الجواب Response وتخزينه في ملف XML بشكل مرتب ومنسق وسهل على التطبيق قراءته، وهذه الصورة توضح كيف يتم ذلك:



## الأدوات اللازمة:

لإنشاء ملف XML سنحتاج إلى أي محرر شفرات مثل مفكرة NotePad أو برنامج Adobe Dream- weaver أو الفيجوال ستوديو أو PHP Designer وغيرها.

## قواعد لغة XML:

وينبغي الحذر عند إنشاء ملف XML إذ يلزم أن يكون بتكوين جيد Well-Formed. ولن يكون كذلك إلا إذا كان يحترم قواعد وضوابط XML. بالنسبة للأوسمة Tags. فهي عبارة عن أسماء نكتبها بين علامتي أصغر من < وأكبر من > لكي نعرف جزء من البيانات. مثلا الوسم <name> قد نستخدمه لتعريف الأسماء المراد تخزينها. ويشكل الوسم من بدايته إلى نهايته ما يعرف بالعنصر Element. طريقة كتابة العنصر Element في XML هي كتابة وسم الافتتاح ثم نكتب قيمة العنصر. وبعد ذلك نقوم بكتابة وسم الإغلاق. الفرق بين العنصر Element والوسم Tag كما يلي:



فيما يلي بعض القواعد التي ينبغي الانتباه إليها جيدا عند إنشاء ملف XML:

- كل عنصر في XML يحتاج إلى وسم الفتح ووسم الإغلاق.
- ملف XML ينبغي أن يحتوي على جذر Root واحد فقط.
- الالتزام بقواعد تسمية العناصر. بحيث لا يقبل بدء الإسم إلا بحرف من الأحرف المعروفة. ولا يقبل الإسم الذي يبدأ برقم أو رمز باستثناء رمز - الأندر سكور. يمكنك كتابة الأرقام بعد اسم العنصر كما تشاء

• لا يسمح بتضمين أسماء العناصر فراغات

<full name>Khalid ESSAADANI </full name>

• لا يسمح ببداية اسم العنصر بالكلمة المحجوزة XML

• احذر أيضا من ترك أي فراغ سواء في بداية الوسم أو في نهايته

<tagname > text </tagname>

• يرجى احترام ترتيب العناصر بحيث لا يمكن كتابة العناصر بشكل متداخل كما يلي:  
<name><age>Khalid</name>26</age>

بل ينبغي إغلاق كل عنصر قبل فتح عنصر آخر.

• إذا كان هنالك عنصر يحتوي قيمة فارغة فيمكنك تمثيله بالشكل التالي:  
</ name>

• أو تقوم بكتابة العنصر ببدايته ونهايته لكن في المحتوى لا تكتب شيئا  
• الأوسمة حساسة جدا لحالة الأحرف بحيث يلزمك كتابة وسم البداية بنفس هيئة وسم  
النهاية.

• الكتابة التالية صحيحة وكل عنصر يعتبر مختلفا عن العنصر الآخر حتى وإن كانوا يحملون  
نفس الإسم لغويا. لأن حالة الأحرف مختلفة:

<Country> Morocco </Country>

<country> Morocco </country>

فيما يخص النص المكتوب داخل وسمي الفتح والإغلاق XML TEXT فلا مشكلة في طول النص  
ويمكنك كتابة النص بالطول الذي تشاء.

• تستطيع أيضا تخزين بيانات ثنائية في عناصر XML. مثلا محتوى أي ملف (صورة أو مقطع  
صوتي...)

• كما يمكنك كتابة نص بأية لغة تشاء داخل العنصر لأن XML عالمية.

• بعض الرموز لا يسمح بكتابتها داخل نص العنصر وهي كما يلي :

الرمز	البديل
<	&lt ;
>	&gt ;
«	&quot ;
'	&apos ;
&	&

## الخصائص Attributes:

يوجد أيضا مفهوم شبيه بالعناصر وهو الخصائص Attributes. والذي نكتبه بالشكل التالي:

```
<element attribute = « » > Content </element>
```

يمكننا إضافة خصائص Attributes للعناصر حسب احتياجاتنا. وفيما يلي مثال على كيفية استخدام الخصائص:

```
<?xml version="1.0" encoding="utf-8"?>
<books>
  <book id="BK1" price="30">
    <title>The Western Stories</title>
    <author>Alex Martel</author>
    <Publish_Date>12/10/2009</Publish_Date>
  </book>
</books>
```

قيمة الخاصية تكتب دائما بين مزدوجتين. ولا يسمح بتكرار الخاصية أكثر من مرة في نفس الوسم. وقواعد تسمية الخصائص هي نفسها التي مرت معنا حينما تحدثنا عن العناصر.

## مكونات ملف XML:

الآن نعرفنا على مفهوم العناصر Elements ومفهوم الخصائص Attributes ومفهوم الأوسمة Tags وكيفية كتابة التعليقات. لكننا لم نتعرف بعد على كيفية إنشاء ملف XML. لذلك دعونا نبدأ الجانب العملي بحول الله.

ملف XML يتكون من جزئين رئيسيين وهما المقدمة Prolog والمحتوى Content. أول شيء نبدأ به عند كتابة XML هو سطر الإعلان (XML Declaration) XML (prolog). والذي يكون بهذا الشكل:

```
<? « xml version = « 1.0? >
```

للإشارة فقط فسطر الإعلان منصوص به لكنه ليس ضروريا في نسخة XML 1.0 لأنه يمكننا الإستغناء عنه وكتابة محتوى XML مباشرة.

لكن في نسخة XML 1.1 فإن تعريف المقدمة إجباري ويعتبر ملف XML من دون المقدمة في هذه النسخة ملفا مختلا Not well-formed.

في سطر الإعلان تعتبر الخاصية version هي الخاصية الوحيدة والمهمة والتي ينبغي تضمينها في سطر الإعلان لتحديد نسخة XML المستخدمة.

يمكننا كتابة مجموعة من الخصائص الأخرى ونعطيها قيمة لكي نسهل على XML Parser عملية معالجة محتوى ملف XML. كأن نكتب خاصية الترميز encoding attribute يوجد العديد من صيغ الترميز أشهرها US-ASCII إلا أنه محدود جدا لأنه يشتغل مع اللغة الإنجليزية ولا يشتغل جيدا مع العديد من اللغات الأخرى. لذلك يبقى الخيار الأفضل هو الترميز الأحادي Unicode الأشهر UTF-8 أو الترميز UTF-16. والفرق بينهما فقط في عدد البايتات التي يستعملها كل ترميز. بحيث الترميز UTF-8 يستخدم بايت واحد لكل حرف وأحيانا إثنان بايت لبعض الحروف. بينما يستعمل الترميز UTF-16 إثنان بايت لكل الحروف.

أيضا توجد الخاصية standalone التي يمكننا تضمينها في سطر الإعلان. والتي تأخذ افتراضيا القيمة no في دلالة إلى أن ملف XML مرتبط بملفات أخرى؛ إن كان ملف XML مستقلا بذاته وغير مرتبط بأية ملفات خارجية يمكننا إعطاء القيمة yes للخاصية standalone. مثل جميع اللغات المشهورة يمكننا كتابة التعليقات داخل ملف XML. والتعليقات كما هو معروف هي عبارات يتم تجاهلها بواسطة XML Parser وتستخدم فقط للعنونة أو كتابة توضيحات. وصيغتها كما يلي:

```
<!-- This is a comment -->
```

ويمكننا كتابة تعليقات متعددة الأسطر بنفس الطريقة.

إذا قمت بالانضباط لجميع القواعد التي ذكرناها فإنك ستنشئ ملف XML مهيكلا جيدا وهو ما يصطلح عليه بالإنجليزية Well-formed. أي أن ملفك منسق ومراعي لقواعد XML وبالتالي يمكنك مشاركته مع مختلف الأنظمة والتطبيقات.

إذن باختصار. فإن ملف XML يعتبر ملفا جيدا الهيكلة well-formed إذا كان يحتوي على جذر وحيد. وإذا كان يحتوي على عناصر وخصائص خاضعة لقوانين التسمية. وإذا كانت كل الأوسمة المفتوحة تم إغلاقها. وإذا كانت قيم الخصائص مكتوبة بين علامتي التنصيص. وألا يكون هنالك تداخل في الأوسمة. بحيث لا يتم فتح وسم إلا بعد أن يغلق الذي قبله.

## XML و XHTML أية علاقة؟

XHTML وهي اختصار ل Extensible HyperText Markup Language. وهي النسخة القياسية لل HTML التقليدي. بحيث تعتمد تقريبا نفس ضوابط لغة XML التي رأيناها آنفا (مع استثناءات طبعاً: مثل حالة الأوسمة Tags Case ففي XML ينبغي الانتباه لحالة الأحرف لأن XML حساسة جدا لحالة الأحرف case-sensitive. بينما في XHTML لا يوجد مشكل في حالة الأحرف not case-sensitive). وهذا يعني أن XHTML هو نفسه HTML لكن بضوابط وقواعد XML.

من بين قواعد كتابة XHTML هي أن كل الخصائص ينبغي أن تحتوي قيمة ولا ينبغي أن تكون من دون قيمة، ففي لغة HTML العادية يعتبر هذا الكود سليماً:

```
<input type= "text" autofocus />
<select>
  <option selected> value </option>
</select>
```

بينما في XHTML يلزم إعطاء قيمة للخاصية autofocus وإلا فإن صفحة XHTML تعتبر غير سليمة not validated. والأصح أن تكتب بهذا الشكل:

```
<input type= "text" autofocus= "true" />
<select>
  <option selected= "selected" > value </option>
</select>
```

### خاتمة:

هكذا نكون قد تطرقنا إلى الركائز الأساسية في لغة XML. يمكنكم الاستفاضة في رواتب لغة XML كتقنيات DTD و XSL وبعض الأمور الأخرى المتعلقة بلغة XML عبر العودة إلى كتابنا: "مدخل إلى لغة XML وما يترتب عليها من نوابغ" وهو موجود على النت بصيغة PDF. كما يمكنك أيضاً متابعة دورتنا المسجلة على شكل فيديوهات على قناتنا في اليوتيوب.

دام لكم البشر والفرح والسلام عليكم ورحمة الله وبركاته.

خالد السعداني

# أشياء نراها تافهة

## كيف تتم عملية النسخ واللصق برمجياً؟

خالد السعداني

• الملفات النصية Text File : ويكون محتوى الملف عبارة عن بيانات نصية.  
• الملفات الثنائية Binary File : تكون على شكل بيانات ثنائية Binary Data. أي عبارة عن متتاليات وسلاسل من الأصفار والآحاد التي بعد أن تتم قراءتها ومعالجتها تتحول إلى شيء معين (صورة، فيديو، مقطع صوتي....)

### عملية النسخ:

عملية النسخ في الأصل تقوم بأخذ البيانات الثنائية للملف وتضعها في ملف جديد فارغ فيتم إنشاء نسخة مماثلة للملف الأول. لأن الملفان معا يتوفران على نفس المحتوى الثنائي.

### النسخ واللصق برمجياً:

لنأخذ أي ملف موجود على حاسوبنا.

### عملية النسخ واللصق Copy and Paste

وتعد من أشهر العمليات التي يقوم بها مستخدمو الحواسيب. بحيث لا غنى لنا عنها. كم من مرة يعجبك مقطع صوتي معين أو صورة معينة أو فيديو معين على جهاز صديقك فتقوم بنسخه ولصقه على حاسوبك. لكن هل سألت نفسك يوماً: مالذي يحدث حينما أقوم بنسخ الملف وكيف يتم تكوين ملف جديد بنفس محتوى الملف المنسوخ؟

في درس اليوم إن شاء الله سوف نتعرف على عملية النسخ واللصق. لكن دعونا نتعرف في الأول على ماهية الملفات Files.

الملف File هو مجموعة من البيانات التي تنتمي إلى نفس النوع. وتنقسم الملفات إلى نوعين:

سنعلن عن مسار الملف الأصلي في متغير نصي:

```
string file1 = @"c:\picture.png;
```

وبالمثل سنعلن عن مسار الملف الثاني أي الملف الذي سينتج عن عملية النسخ:

```
string file2 = @"c:\copy.png;
```

الآن سنقوم بفتح الملف الأول لقراءة بياناته الثنائية:

```
FileStream fs1 = new FileStream(file1, FileMode.Open);
```

ثم نقوم بإنشاء ملف فارغ خالٍ من أية بيانات ثنائية على مسار الملف الثاني:

```
FileStream fs2 = new FileStream(file2, FileMode.Create);
```

الملف الأول موجود عندنا مسبقاً لذلك سنقوم بقراءة بياناته الثنائية بواسطة Binary Reader. والملف الثاني ليس له محتوى ثنائي سنقوم بتزويده بالبيانات عبر Binary Writer.

```
BinaryReader br = new BinaryReader(fs1);
```

```
BinaryWriter bw = new BinaryWriter(fs2);
```

الآن هيأنا كل شيء. بقي فقط تعبئة الملف الثاني بالبيانات الثنائية للملف الأول. وذلك عبر تكرار يذهب من أول بايت إلى آخر بايت في الملف الأول. وكلما وصل إلى بايت معين يقوم بكتابتته في الملف الثاني. كما يلي:

```
for (int i = 0; i <= br.BaseStream.Length - 1; i++)
{
    bw.Write(br.ReadByte());
}
```

في الأخير نقوم بإغلاق القارئ BinaryReader والكاتب BinaryWriter.

```
br.Close();
```

```
bw.Close();
```

#### ملخص عملية النسخ واللصق:

1. فتح الملف المراد نسخه.
2. إنشاء ملف فارغ.
3. كتابة بيانات الملف الأول في الملف الثاني.
4. النتيجة ملفان متشابهان.



# كَوَالِيْسُ نَاجِحٍ دَوَّخَهُمْ

بقلم: خالد السعداني

إن كنت تنتظر بعد أدائك الحسن شكورا  
فقد خطوت أول خطوة في درب الفشل،  
لأن شكر الناس مقرون بدوام فضلك  
عليهم، فإن نقصته نقموا، وإن أزلته  
شتموا، فلا تركز إلى محاسن أقوالهم  
فيك فإنها كشعر الغزل، يلزم الحسنة ما  
بقيت حسنة فإن بزغ الشيب من خصلاتها  
استحال الجميع حُطِيَّةً، اعمل الجميل في  
الناس ولا تنتظر منهم شكورا فإن العمل  
الحسن يُتَغَى به وجه المولى تبارك  
وتعالى فإن خلا من الإخلاص ذرته رياح  
الرياء فلا الناس أقروا لك بخيرك، ولا الأجر  
اُحْتَسِبَ لك.



# كَوَالِيْسُ نَاجِحِ دَوَظَهُمْ

خالد السعداني

زعموا أن رجلا بلغ من النجاح كل مبلغ، فكان لا يضع يده على شيء إلا حالفه السداد، ولا يرى في مسألة رأيا إلا واطأه الصواب، فاحتر الناس في أمره أشد الحيرة، وبذلوا لمعرفة حَبِيَّتِهِ كل جُهدٍ ولأَيِّ، فلما فطن لذلك دفعته نفسه الصافية إلى شرح أساليبه للناس ليستفيد منها كل من له بها حاجة.

فصعد ذات يوم ربوة عالية، وصاح في الناس يجمعهم حوله، فلما خَلق حوله جمع غفير خطب فيهم قائلاً:

يا معشر الناس، إن الحديث عن النجاح حديث ذو شجون، ولا يمكننا حصره في هذا الملتقى، لكنني سَأُجِملُ وَأُخْصُّ، عسى أن أصيب بهذه العصاراة بيت القصيد فتتضح الصورة في أذهانكم ويستحيل التراب في أيديكم دَهَبًا.

إن النجاح قمة علياء، وما بلغتها عن كسل مني أو تواكل، وإنما بلغتها بجهد وفقني المولى تبارك وتعالى إليه، ألا وإن الناس يظنون النجاح مرهونا بطول الزمن، ولا يعلمون أنه مرهون بدفع الثمن، فمن شاء أن ينجح، فليقدم من وقته أَثْمَنَهُ، ومن فكره أَحْسَنَهُ، وَلَا يُجَارِيَنَّ السَفْهَاءَ بِخُلُقٍ يَجْرُهُ إِلَى حَبِيَّتِهِمْ كما يَجْرُ الطعم السمك، فَيَخْسِرَ حياته مقابل دودة، لِيَخْسِرَ مُجَارِي السَفْهَاءَ جَاحَهُ مقابل نقاش عقيم.

يا معشر الناس، إنني سلكت في درب النجاح أربعة مسالك، فاسلكوها تظفروا بما ظفرت به.

## المسلك الأول: إن سألت عن مساعدي، فهو ساعدي:

إن الذي يرى في جهد الآخرين وسيلة لبلوغ ذروة نجاحه إنسان واهم، فالنجاح مقرون بالجهد الذاتي، ومن بلغ ذروة النجاح مرتقيا ظهور غيره كان سقوطه أشبه بسقوط البهلوان الذي يمشي على الحبل وذراعه تمسكان بثابت، فإن غاب هذا الأخير أو تهادى، هوى إلى الأرض.

**المسلك الثاني: اعمل جميلاً، ولا تنتظر جميلاً:**

إن كنت تنتظر بعد أدائك الحسن شكورا فقد خطوت أول خطوة في درب الفشل. لأن شكر الناس مقرون بدوام فضلك عليهم. فإن نقصته نقموا. وإن أزلته شتموا. فلا تركز إلى محاسن أقوالهم فيك فإنها كشعر الغزل. يلزم الحسنة ما بقيت حسنة فإن بزغ الشيب من خصلاتها استحال الجميع حطيئة. اعمل الجميل في الناس ولا تنتظر منهم شكورا فإن العمل الحسن يُبتَغَى به وجه المولى تبارك وتعالى فإن خلا من الإخلاص ذرته رياح الرياء فلا الناس أقرؤا لك بخيرك. ولا الأجر احتسب لك.

**المسلك الثالث: عيني المرموذة ولست أجرباً:**

لم أزل أوظن نفسي على إحسان الظن بالناس ولو أسأؤوا. فلست موكلاً بالتنقيب في السرائر. أو إجلاء مكنونات الضمائر. ما أنا إلا إنسان بسيط أشغل نفسي بعيوبها فأربح كامل وقتي لإصلاحها بدل تضييعه في تتبع عيوب الغير فلا أنا أصلحت نفسي ولا ربحت وقتي ولا أرحت الناس من شري و مساوئي.

لقد قدرني المولى تبارك وتعالى بعد أن أخلصت نيتي إليه. وسلمت مقاليد شؤوني إليه. على أن أتهم عيني بالرمد. بدل أن أصم جلد غيري بالجرب ولو كان كذلك. تعلموا أن تنسبوا عيب الآخر إلى نفوسكم. لتستشعروا مرارة ظلمكم. فإن لم تقدرُوا فانسخوا عيب غيركم بجميل ظنكم.

**المسلك الرابع: بذل الجهود في تقليل العهود:**

اعلموا أن أصل النجاح في البخل بتوزيع العهود. لأن كثير العهود إما أن يخذل الناس أو يخذل نفسه. إن بقي على عهده ووفى لكل الناس ضيع نفسه بضياع وقته. وإن نكث عهده خسر الناس وإن فاز بكل وقته.

تعلموا أن تكونوا بخلاء في توزيع العهود. وأدركوا أن النجاح كل النجاح في الحسم والصدق. لا جامل أحدا على حساب وقتك. فإن العهود سيوف على رقبتك نازلة نازلة. سَوَاءً أَوْفَيْتَ أَمْ نَكَنْتَ. لذلك لا تعد أحدا بشيء إلا إن كنت قادرا عليه. فإن توزيع الوعود في طعمه كتوزيع الورود. لكنه عند الأداء أشد وطأ من القيود.

# المختصر النافع في Enumerations

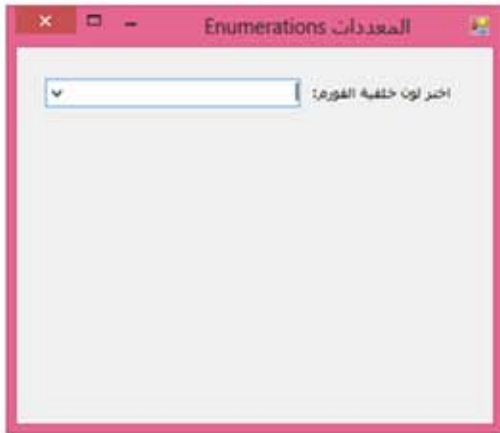
## ماهي المعدادات Enum؟ لماذا وكيف نستعملها؟

خالد السعداني

. أسماء فواكه، ألوان...، فيفضل أن نقوم بتجميعها داخل المعدادات ليصبح استخدامها لهذه الأنواع مرنا بدل تشتيت الكود في الإعلان عن مجموعة من الثوابت.

### مثال عملي:

قم بإنشاء مشروع جديد من نوع Windows Forms Application وقم بتصميم الواجهة كما يلي:



الصورم أعلاه يشمل الأدوات الواردة في الصفحة التالية:

### تقديم عام:

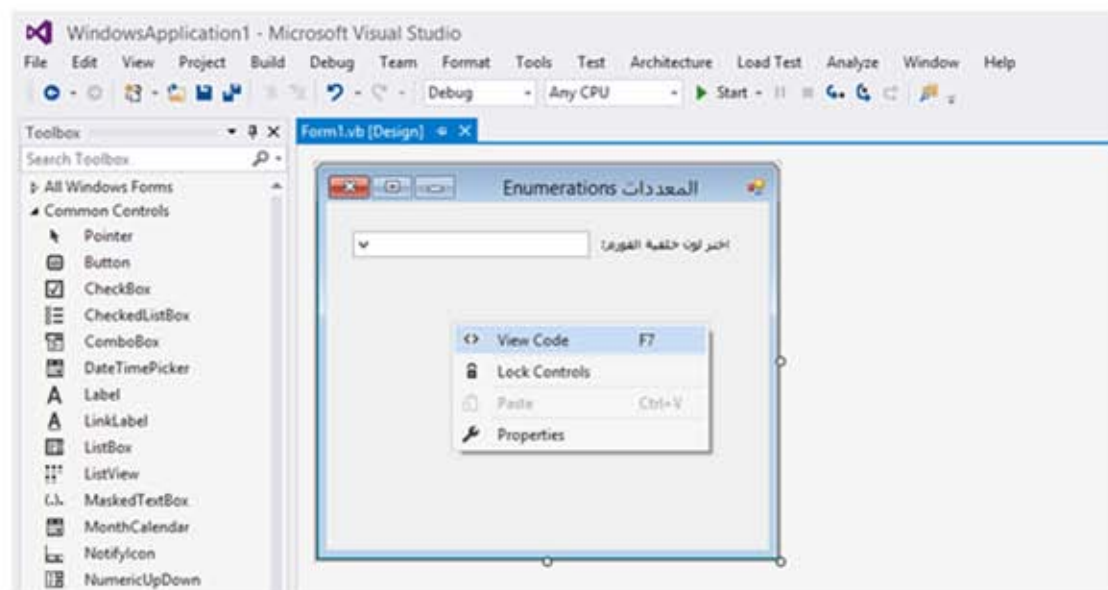
المعدادات Enumerations والمعروفة اختصارا ب Enum هي مفهوم برمجي يسمح لنا بتجميع القيم الثابتة في مكان واحد. مع تسهيل عملية الوصول إلى هذه القيم الثابتة عبر التعامل مع المعدادة كما لو أنها نوع بيانات.

نستخدم المعدادات إذا كنا نريد تحديد قيم محددة ثابتة ليتم تخزينها في متغير ما. مثلا حينما نريد حفظ أيام الأسبوع في برنامجنا، فنحن نعلم مسبقا أن مجال أيام الأسبوع محدد والقيم معروفة لذلك يمكننا استخدام المعدادات بدل الإعلان عن مجموعة من الثوابت Constants.

حينما يكون لدينا مجموعة من الثوابت التي تنتمي إلى نفس المجال، مثلا أيام أسبوع.

الأداة	اسمها	دورها
Label	lblColor	من أجل عرض النص
ComboBox	cmbColor	من أجل اختيار لون خلفية الفورم

الآن سنقوم بإنشاء معدة Enumeration خاصة بأسماء الألوان التي نريد تعبئة الكومبوكس بها. قم بالضغط بيمين الماوس على الفورم واختر View Code كما يلي:



بعد ذلك قم بإنشاء المعددة كما يلي:

#### VB.NET CODE

```
Enum EnumColors
```

```
    Red
```

```
    Blue
```

```
    Green
```

```
    Purple
```

```
    Orange
```

```
End Enum
```

## C#.NET CODE

```
enum EnumColors
{
    Red,
    Blue,
    Green,
    Purple,
    Orange
}
```

المعددة EnumColors تحتوي على مجموعة من الثوابت التي لها رتب تبدأ من صفر وتنتهي برتبة آخر عنصر ناقص واحد. كما يلي:

الثابت	قيمه
Red	0
Blue	1
Green	2
Purple	3
Orange	4

يمكننا التحكم في رتب العناصر المكونة للمعددة عبر إسناد القيمة الرقمية لكل ثابت من الثوابت كما يلي:

## VB.NET CODE

```
Enum EnumColors
    Red = 4
    Blue = 2
    Green = 1
    Purple = 3
    Orange = 0
End Enum
```

## C#.NET CODE

```
enum EnumColors
{
    Red = 4,
    Blue = 2,
    Green = 1,
    Purple = 3,
    Orange = 0
}
```

عند التعامل مع عناصر هذه المعددة يكون الترتيب مبنيًا على القيم الرقمية التي تم إسنادها لكل عنصر.

الآن سنقوم بتعبئة الكومبوكس الذي أسميناه cmbColor بأسماء الألوان إما يدويًا أو من خلال إنشاء مصفوفة. قم بالولج إلى الحدث Form\_Load الخاص بالفورم من خلال النقر عليه مرتين واكتب الشفرة الآتية:

## VB.NET CODE

```
Dim ArrayColors() As String = {"Red", "Blue", "Green", "Purple", "Orange"}
```

```
Me.cmbColor.Items.AddRange(ArrayColors)
```

## C#.NET CODE

```
string[] ArrayColors = {"Red", "Blue", "Green", "Purple", "Orange"};
```

```
this.cmbColor.Items.AddRange(ArrayColors);
```

بعد ذلك، قم بالنقر مرتين على أداة ComboBox التي أسميناه cmbColor من أجل الانتقال إلى الحدث SelectedIndexChanged الذي يصدر عند اختيار عنصر من عناصر الكومبوكس. ثم اكتب الشفرة التالية:

## VB.NET CODE

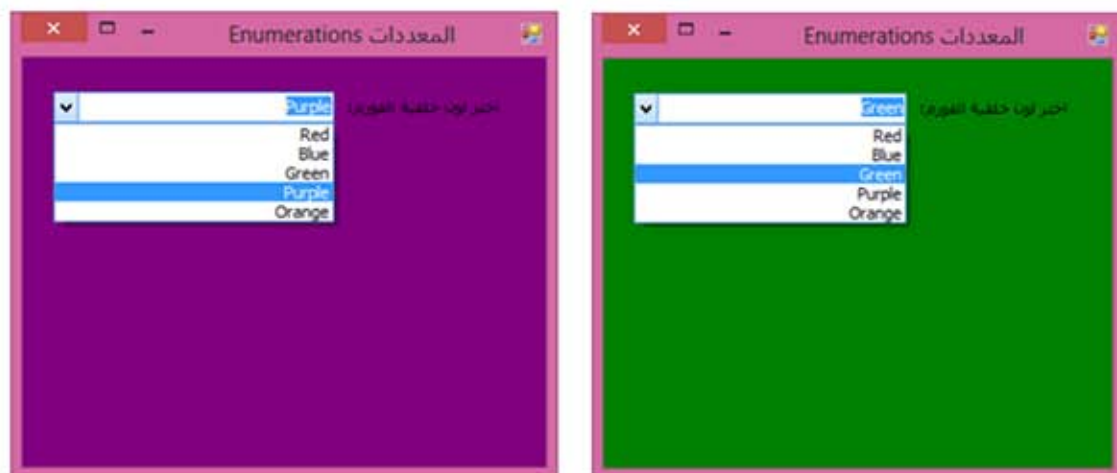
```
Select Case cmbColor.SelectedIndex
    Case EnumColors.Red
        Me.BackColor = Color.Red
    Case EnumColors.Blue
        Me.BackColor = Color.Blue
    Case EnumColors.Green
        Me.BackColor = Color.Green
    Case EnumColors.Purple
        Me.BackColor = Color.Purple
    Case EnumColors.Orange
        Me.BackColor = Color.Orange
End Select
```

## C#.NET CODE

```
switch(cmbColor.SelectedIndex)
{
    case (int)EnumColors.Red:
        this.BackColor = Color.Red;
        return;
    case (int)EnumColors.Blue:
        this.BackColor = Color.Blue;
        return;
    case (int)EnumColors.Green:
        this.BackColor = Color.Green;
        return;
    case (int)EnumColors.Purple:
        this.BackColor = Color.Purple;
        return;
    case (int)EnumColors.Orange:
        this.BackColor = Color.Orange;
        return;
}
```



الكود السابق يتحقق من رتبة القيمة المحددة في الكومبوبوكس، فإن كانت تساوي صفر وهي نفس رتبة اللون الأحمر في المكددة EnumColors يتم تغيير لون خلفية الفورم إلى الأحمر. وهكذا دواليك مع باقي الألوان، عند تنفيذ الكود أعلاه سوف نحصل على النتيجة التالية:



إلى هنا نكون قد تعرفنا على المعدّات Enumerations في الدوت نيت، دورها، وكيفية استخدامها. نكتفي بهذا القدر مع متمنّياتي لكم بالتوفيق والسداد.

دام لكم البشر والفرح.

# قهوة الرابط

فضاء للاستراحة

# النميمة والبرمجة

إذا أردت أن تقتل النميمة والغيبة في بلاد فَعَلِّمْ  
أهلها البرمجة والتصميم، ولاحظ أن الحديث  
سينتقل من فلان وعلان إلى أكواد و ألوان.

## متعة البرمجة

متعة البرمجة في ارتكاب الأخطاء وإصلاحها،  
ومواجهة الصعوبات وتجاوزها، و المبرمج الذي لا  
يتعب ذهنه، ليس سوى كاتب عمومي.

## انتكاس الفطرة البرمجية

كان أحد الحكماء قديما يبرمج بلغة الفيجوال بزيك  
٦، ولما تقدم الزمن جمع طلابه وقال لهم: سأقص  
عليكم لغزا فمن أصاب مغزاه جعلته على خزائن  
الأكواد المصدرية.

أوما الطلاب برؤوسهم موافقة، ثم تابع الحكيم:  
ليست شيئا مملوسا، وأحالت مجدنا مملوسا.

مفاهيمها كلها لؤم، وذكر اسمها نذير شؤم.

سكت الطلاب جميعا، فانبى من الخلف شاب قد

قارب سن العشرين ثم قال: عرفتھا يا سيدي، إنك

تتحدث عن البرمجة الكائنية OOP.

اندهش الحكيم ومعه الطلاب، ثم قال:

– كيف عرفت يا ولد؟

ضحك الغلام وهو يقول: من اعتاد البرمجة الإجرائية

رأى في البرمجة الكائنية انتكاسة للفطرة السليمة.

## التنمية البشرية في حكم المماليك

كان معروفًا على أبي الأقباس الأندلسي حبه الشديد  
لمجامع البرمجة وتعلقه بأهلها، وفي إحدى جولاته مر من  
أمام بيت فسمع الكلمات التالية:  
برمجة، تحليل، الثوابت، ...  
فاستبشر خيرا وفرح بذلك قائلا: يا سلام هنا يوجد  
مبرمجون !  
فدخل دون استئذان ليجد أحد مدربي التنمية البشرية  
يخاطب جمهوره قائلا:  
رددوا في أعماقكم:  
نحن أقوىاء !  
قوموا بتحليل أفكاركم !  
حافظوا على الثوابت !  
لا تسمحو لأحد ببرمجتكم...  
غضب أبو الأقباس غضبا شديدا وصاح بهم:  
لو ذات سوار لظمتني، غيروا قاموسكم يرحمكم الله !

## نعيم الدنيا

إذا أردت أن تعيش نعيم الدنيا، اعتزل السياسة،  
وقاطع الصحافة، ولا تتخذ حيا لك قدوة، عليك  
بالكتب، وصاحب أحد الموتى الصالحين فإنه لن  
يخذلك وقد أفضى إلى ما قدم.

## الأميرة والمبرمج الساخج

كبرت الأميرة كارولين فأمر الملك هيكتور بقرع الأجراس لإعلان جاهزيتها للزواج لكي يتقدم خيرة شباب المملكة إلى خطبتها، وفي صباح اليوم الموالي بدأ الناس يتوافدون إلى القصر ليعرضوا أنفسهم للأميرة.

فتقدم شاب من أغنى أغنياء المملكة وقدم إليها عقدا من الألماس مدبجا بالذهب، فأعرضت عنه.

ثم تقدم شاب شاعر وبدأ يمدح الأميرة بكلمات عذبة، فرفضته. ثم تقدم مبرمج لا يحمل بين يديه شيئا، ويرتدي ملابس مهلهلة بالثقوب، وعيناه مثقلتان بالنوم من شدة السهر، سألته الأميرة: ماذا لديك لتقدمه إلي؟ قال: لا شيء.

ضحكت الأميرة ساخرة:

- أمجنون أنت؟

فرد عليها بهدوء:

- إن تزوجتني سأسترجع لك حسابك على الفيسبوك.

نظرت إليه باندهاش:

- وكيف عرفت أن حسابي على الفيسبوك مسروق.

ضحك المبرمج بجنون:

- لأنني أنا من سرقه. فهل تقبلين بي زوجا؟

كان يظن أن النهاية ستكون مثل الأفلام، وستضحك الأميرة

وتتزوجه ويعيشان في سعادة وسلام، لكنه صدم حينما بدأت

تنادي الأميرة:

- أيها الجنود أمسكوا هذا اللص لقد سرق حسابي على

الفيسبوك.

# ألف ليلة وليلة (نسخة برمجية)

استلقى شهريار على سريرهِ وعيناه لا تفارقان عيني شهرزاد،  
ثم خاطبها بشدة:

– أعتقد بأن قصصك قد انتهت، أخاف عليك – إن لم تسردي  
علي قصة هذه الليلة – من المصير المشؤوم، تعلمين جيداً  
ماذا لحق البنات قبلك.

طأطأت رأسها في حياء وهي تقول:

– في جعبتي حكاية يا مولاي.

أشار شهريار بيده إليها قائلاً:

– تفضلي يا شهرزاد.

رفعت رأسها وهي تهمس:

كان ياما كان في قديم الزمان، مبرمج مهووس بتخصص  
الحماية والأمان، أوتي من كل لغة برمجية حظاً، لكنه كان غليظ  
الطباع فظاً، ضئيل الصبر كالماء إذا نضب، سريع الهيجان  
كالليث إذا غضب، وبينما هو منشغل في أحد البرامج، حصل  
معه خطأ bug فصار كالثور الهائج، بحث في غوغل فلم يجد  
إجابة، وحاول مع الكود ففقد صوابه،

كان الكود سليماً من الشوائب، وكان الخطأ من أغرب الغرائب،  
وبعد جهد طويل، وبين صياح وعويل، لم يجد لخطئه تفسيراً،  
فخبط الحاسوب وكسره تكسيراً، وبعد أن انتهى من التكسير  
والدوس، تذكر أنه كان قد نسي في الكود كتابة القوس، فندم  
على فعلته أشد الندم، وقرر ترك البرمجة واشتغل ببيع العمم.

رفعت شهرزاد رأسها فوجدت شهريار يغط في نوم عميق،

فاستغلت الفرصة وأخذت حاسوبه لتنشئ صفحة على

الفيسبوك بعنوان:

"أنقذوا شهرزاد من بطش أمير البلاد".

# نظرة شمولية حول تقنية ASP.NET

## من ASP إلى MVC و Web API

خالد السعداني

مواقع و خصائص مكونات الصفحة.  
 3. لغة جافاسكريبت: تسمح بجعل الصفحات مرنة وقادرة على التجاوب مع المستخدم. لأنها لغة برمجية وتوفر للمبرمج كل ما يحتاجه للقيام بعمليات برمجية على الصفحة. كاستقبال القيم من المستخدم والتعامل معها. وإجراء العمليات الحسابية والمنطقية. والتحقق من العبارات. وتكرار الأوامر عبر آلية الحلقات. واقتناص الأحداث. وتحديد عناصر HTML عبر ترانبيه DOM وغير ذلك من المزايا القوية التي توفرها لغة جافاسكريبت.  
 كل اللغات الأنفة. تنفذ على مستوى المتصفح. أي على حاسوب المستخدم / الزائر. ما يضمن سرعة استعراض الصفحات. لأن هذه السرعة غير مقسمة من طرف كل الزوار. فهي متعلقة بجهاز الزائر. على خلاف لغات السيرفر. التي كلما ازداد عدد الزوار والمستخدمين. ازداد الضغط على السيرفر مما قد يبطئ عملية استعراض الموقع أو التطبيق حسب كفاءة جهاز السيرفر.

### تقديم عام:

تقنية ASP.NET باختصار هي إطار عمل Framework أنشأته شركة ميكروسوفت للسماح للمبرمجين بإنشاء صفحات الويب الديناميكية. فكما نعلم جميعا لا بد من الإلمام بمجموعة من التقنيات واللغات لكي نستطيع إنشاء صفحات الويب.

ولعل أبرز ما نحتاجه:

1. لغة HTML: باعتبارها اللغة التي تسمح لنا ببناء الهيكل الأساسي لمواقع وتطبيقات الويب. ومن دونها لا نستطيع إنشاء الصفحات. إذ هي التي تسمح لنا بالقيام بأول خطوة وهي عرض المحتوى وبناء الهيكل الرئيسي للصفحات.
2. لغة CSS: تسهل عملية التصميم والتحكم في شكل الصفحات وتحديد

لكن اللغات السابقة وحدها لا تكفي لبناء تطبيق ويب متكامل. يسمح بإجراء العمليات على مستوى السيرفر، كإرسال الرسائل، أو التعامل مع قواعد البيانات وما إلى ذلك. لذلك وجب استخدام إحدى اللغات الديناميكية التي تخول لنا القيام بهذه العمليات وغيرها. توجد لغات وتقنيات كثيرة، تسمح ببناء أنظمة ديناميكية على الويب. لعل أبرزها لغة PHP و JSP و التقنية الرهيبة التابعة لمنصة الدوت نيت ASP.NET. ظهرت تقنية ASP.NET عام 2000 تحت اسم ASP اختصاراً لـ Active Server Pages، وجاءت كحل من ميكروسوفت لتمكين المبرمجين من بناء صفحات ويب ديناميكية، وكان ما يميز هذا الإصدار القديم أنه يسمح بكتابة أوامر ASP داخل أوامر HTML بين الرمزين `<% %>`. فتقدم الزمن بعد ذلك، وتطورت تقنية ASP ليصبح اسمها ASP.NET، وصارت جزء لا يتجزأ من إطار العمل دوت نيت فريموورك، وكان من مستجدات التقنية، أنها صارت تعتمد على الترجمة -Compilation- لا على التفسير Interpretation كما كان الحال مع تقنية ASP الكلاسيكية، إضافة إلى تقسيم الكود إلى صفحتين، ما يتعلق بـ HTML يكتب في صفحة وما يتعلق بلغة البرمجة يكتب في صفحة تعرف باسم Code Behind.

تسمح تقنية ASP.NET بإنشاء صفحات ويب ديناميكية بعدة طرق، لعل أبرز هذه الطرق:

## 1. طريقة Web Forms:

وهي طريقة قوية جداً، شبيهة جداً بنمط برمجة الواجهات في الويندوز، إذ بدورها تعتمد على آلية السحب والإفلات Drag and Drop لوضع الأدوات على الويب فورم، واقتناص الأحداث بطريقة ماثلة جداً لبرمجة الواجهات، وهذه الطريقة هي الطريقة التي اشتهرت بها تقنية ASP.NET نظراً لسهولة تعلمها، وكفاءتها العالية.

## 2. طريقة MVC:

اختصاراً لـ Model, View and Controller وهي طريقة حديثة جداً في تقنية ASP.NET علماً أنها نمط قديم يعود إلى سبعينيات القرن المنصرم، لكن لم تمثلها ميكروسوفت إلا حديثاً بسبب قوة هذا النمط البرمجي الذي ينتمي في الأصل إلى نماذج التصميم Design Patterns، وهو نمط يقضي بتقسيم المشروع إلى مجزوءات، كل مجزوءة تتناول جانباً من المشروع، وهي كما يلي:

مجزوءة النموذج Model، وهي تشمل كل ما من شأنه بناء الفئات وإنشاء خصائصها

مجزوءة العرض View، وهي خاصة بعرض النتيجة للمستخدم وإمداده بحصيلة تنفيذ Action معين.

مجزوءة المراقبة Controller، وهي تلعب دور الخادم الذي يتنقل بين View و Model لتحديد ما الواجب استدعاؤه وما الواجب عرضه.



أمام هذا التقسيم، تصبح عملية بناء صفحات الويب سلسلة، والتركيز ينصب على كل مجزوءة بما تقتضيه بدل الانكباب الشامل على الوحدات الثلاثة في صفحة واحدة كما هو الحال في طريقة Web Forms. هذا بالإضافة إلى ما يتيح النمط MVC من يسر وسهولة في عملية إجراء الاختبارات .Testing

### 3. طريقة Web pages:

وهي طريقة سهلة لبناء الصفحات الديناميكية، تعتمد بالأساس على الشكل الكلاسيكي لتقنية ASP، إذ يتم الجمع بين كود السيرفر وكود العميل في صفحة واحدة. كود العميل معلوم حاله وهو عبارة عن أوسمة HTML وما يترتب عنها من اللغات والتقنيات التي تعمل عند العميل. وكود السيرفر هو عبارة عن كود مكتوب بطريقة حديثة اختارت لها ميكروسوفت اسم Razor. وهي طريقة تنبني في الأساس على كتابة الأوامر البرمجية بعد الرمز @. كما تسمح أيضا باستخدام لغات البرمجة فيجوال بيسيك و سي شارب مع أوامر Razor حسب نوع الصفحة. إن كانت صفحة العرض بامتداد cshtml فأوامر Razor تدعم لغة سي شارب. وإن كانت بامتداد vbhtml فهي تدعم لغة فيجوال بيسيك.

### 4. طريقة Web API:

وهي طريقة حديثة بدورها، مدارها بالأساس على بناء تطبيقات من نوع RESTfull أي أنها عبارة عن تطبيقات قريبة من مفهوم خدمات الويب Web Services. تتم برمجتها بتقنية ASP.NET بالاعتماد على النمط MVC لتصير قابلة للاستهلاك من قبل تطبيقات أخرى متصلة بالسيرفر الذي يستضيفها. وهي تدخل فيما يعرف ببناء الأنظمة المتصلة Connected Systems. أي مجموعة من التطبيقات المتصلة فيما بينها عبر وسيط شبكي.

طريقة Web API تعكس ميول شركة ميكروسوفت للاستحواذ على مجال صناعة خدمات الويب. إذ من الملحوظ أن ميكروسوفت أخذت منحى تصاعديا في هذا المجال. كان مبدؤه مع خدمات الويب في إصدارات ASP.NET القديمة. مرورا بتقنية WCF التي أتت كحل متكامل لبناء تطبيقات موزعة خدميا (SOA) (Services Oriented Architecture). وانتهاء بتقنية Web API التي - في نظري - تتفوق على كل ما تقدمها من تقنيات بسبب سهولة استعمالها في إنشاء تطبيقات من نوع RESTfull.

# كلمة ختامية

## ليس بين الرفوف!

لقد وضعت الكتاب هنا. على هذا الرف. بين كتاب "المفضليات" وكتاب "ضرائر الشعر". فما لي لا أجده الآن؟

علي العودة إلى فصله السابع لأطلع عليه من جديد. وإلا لضاعت جهودي كلها... أنا متأكد أنني وضعته هنا. وذاكرتي آخر من يخذلني. إذن فأحدهم أخذه من غير إذني.. لكن من يا ترى؟ لا أحد يطرق مكتبتي لأشك في عبثه بالرفوف. إذن فأين ضاع هذا الكتاب؟ ضاع الكتاب. وضاع معه العمل المرهون به. وكله بسبب سوء التنظيم. فإن كنت معاتباً فلأعاتب نفسي. لأن التنظيم شأن ذاتي لا دخل ليد العابثين به. والكتاب من غير شك موجود لكن مجهول الأين.

القصة أعلاه ليست حادثة شخصية. وإنما تمثيل لحالات كثيرة نقع فيها - نحن معشر المبرمجين - إذ نتعامل مع الأدوات التي نستخدمها بسوء تنظيم. فحاجتنا إليها حاجة لحظية. وما بعد اللحظة لا يدخل في اعتباراتنا. حتى إذا ما بلغت الحاجة إلى الأدوات مبلغها. صارت في عداد المفقودات. ليتوقف العمل الذي كان متوقفاً عليها. وكله بسبب سوء التنظيم.

حدثت في وقت سابق في كتاب "ماذا بعد الأساسيات" عن أهمية التنظيم في الحقل البرمجي. وكنا قد ركزنا الحديث عن تنظيم المشاريع والأكواد المصدرية. والحرص على إعطائها أسماء تعبر عن دورها. وأحببت إثارة هذا الموضوع من جديد لما له من أهمية بالغة في هذا الحقل. لذلك أنصح جميع الإخوة والأخوات العاملين بالحقل البرمجي أن يقوموا بتنظيم البيانات على أقراصهم الصلبة. عبر إنشاء مجلدات رئيسية. على سبيل المثال (أعمال برمجية، أكواد مصدرية، موارد معرفية...) وداخل كل مجلد رئيسي نضع مجلدات فرعية. مثلاً مجلد الموارد المعرفية ننشئ داخله مجلدات من قبيل (كتب، فيديوهات، ..) وهكذا دواليك.

بهذه الكيفية. سيكون قرصك الصلب منظماً. كلما احتجت شيئاً ما. قم فقط بتحديد طبيعته وعد إلى المجلد المعني بالأمر.

دون أن تغفل عن درء الكسل لنقل الملفات المهمة من سطح المكتب إلى المجلد الخاص بها. فالكسل هنا مدخل من مداخل سوء التنظيم. ولو ركنت إليه. ستصبح عند الحاجة: ليس بين الرفوف! توجد أمور كثيرة تدخل في عملية التنظيم يقصر المقام عن سردها كلها. وقد تناولها في مقالات قادمة إن شاء الله.

دام لكم البشر والفرح.