

محتوى العدد

المقالات التقنية للربع الثاني
2010

مقالات منتدى فيجوال بيسك



VB.NET

C#.NET

ADO.NET

SQLSERVER

Asp.net

مقالات التقارير والطباعة

www.vb4arab.com

فيجوال بيسك العرب مجتمع المبرمجين العرب الأول



المحتويات :

مدير المجلة : الأستاذ عبد الله العتيق
فريق العمل : إدارة منتدى فيجوال بيسك العرب
www.vb4arab.com

أقسام المجلة لهذا العدد :

VB.NET

C#.NET

ADO.NET

SQLSERVER

ASP.NET

Reports

قسم أخبار وجديد المنتدى

قسم استراحة المبرمجين

تطالعون في هذا العدد :

مقالات قسم VB.NET

أسلوب الاستدعاء الذاتي و مدخل إلى تطبيقات أكثر كفاءة و عملية الوافي في تحريم البرامج للدوت نت إنشاء **odbc** برمجيا

برمجة سعيدة مع **Implicit Line Continuation**

جديد **2010** اكتب الكود الذي سيستخدم فنتك ثم قم بإنشائها شرح الوافي في حماية البرنامج عبر التسجيل والتفعيل كشف رقم المتصل عن طريق فيجوال بيسك دوت نت مقال أفكار في الدوت نت **OutAttribute Class** مقال أفكار في الجرافكس مقال مقدمة إلى حماية البرامج

مقالات قسم C#.NET

ال **Delegates** مع ال **C#**

فهم كيفية بناء شبكة عصبية صناعية للتعرف على حرف أو رقم..

مقالات قسم ADO.net

درس بالصور ::: كيفية التعامل مع الجداول المترابطة في قواعد البيانات كيفية حفظ القيمة المنطقية (**True Or False**) في قاعدة البيانات

مقالات قسم SQL SERVER

معلومة ::: البحث الصوتي مع الدالة **Difference & soundex** ::: مهم جداً

كيفية اعتماد واستخدام ال **TRANSACTIONS** داخل **SQL Server**

كيفية التعامل مع المتغيرات داخل **SQL Server**

كيفية استخدام **IF** داخل **SQL Server**

مقالات قسم Asp.net

درس كيفية معرفة البلاد لزوار الموقع شرح عمل تصويت بتقنية **ajax**

مقالات التقارير والطباعة في NET

التعامل مع الصور في تقارير الـ .net
التقارير الفرعية في الكريستال
السر في تصميم الاتصال بالكريستال للقواعد الغير مدعومة
حل مشاكل الكريستال مع 2010 vs
درس مبسط تمرير القيم من برامجنا لتقارير الكريستال ريبورت
طريقه عمل تقرير وفاتورة بفلتر بكل في داتا جريد ولييل وداتا بيز اكسس
لا لتسجيل الدخول بعد الآن في الكريستال الجزء الأول
معلومة مشكلة كريستال ريبورت مع فيجوال استوديو 2010

جديد المنتدى
إعلانات و مسابقات

استراحة العدد
عن البرمجة نتكلم
تعرف على العادات السيئة للمبرمجين (برمجياً) عادة سيئة للمبرمجين (أعطنا
تجربتك)

المقدمة : بسم الله الرحمن الرحيم

السلام عليكم ورحمة الله وبركاته:

كان من المفترض إطلاق هذه المجلة في النصف الثاني من العام ٢٠١٠ ولكن تأخر الإصدار
لأسباب عديدة وبعدها توقف .

والأهم من ذلك أن منتدى فيجوال بيسك العرب أيضاً توقف كما توقف في سابق عهده.
لذلك أثرت أن أنشر المجلة :

- لتكون أولاً بما فيها من علوم التقنية والبرمجة مفيدة لكل من يقرأها.
 - لتكون التذكار الأخير لي مع هذا المنتدى.
- على أمل أن يعود ولو بشكل من الأشكال أو يجد من يرثه ليستمر النهج العلمي المميز الذي
كان يتبعه المنتدى.

مع تمنياتي للجميع بالتوفيق نور نبهان ٢٠١٣/٠١/٠١

المقالات

البرمجية

أسلوب الاستدعاء الذاتي و مدخل إلى تطبيقات أكثر كفاءة و عملية

الكاتب : [egyption-coder](http://egyption-coder.com)

[رابط المقال و النقاش حوله](#)

صغر حجم الكود .. استهلاك اقل للذاكرة :

اعتقد أن هذه الأمور من أهم ما يشغل المطورين و المبرمجين في جميع اللغات ناهيك عن برامج لا يمكن تطبيقها إلا بهذا الأسلوب مثل مستكشف الويندوز (**windows Explorer**) أو برامج مضادات الفيروسات أو برامج الكراك أو الألعاب.. الخ بالطبع الموضوع قديم و لكن نظرا لإمكانية تجاهله فقد تجاهله الكثير من المبرمجين و بعضهم يتجنبه أما البعض الآخر فيكتفي بنظرة خاطفة على الموضوع ثم يتحول عنه

إن أسلوب الاستدعاء الذاتي قد بدأ استخدامه منذ الإصدارات الأولى من الفيجوال بيزيك و قد أصبح من الصعب تجنبه بعد ظهور الفيجوال دوت نت و هو ذلك الجزء الذي يقوم باستدعاء نفسه بنفسه

و في هذا المقال إن شاء الله سأحاول أن أوضح كيف يمكن أن نستخدم هذا الأسلوب في إنشاء مستكشف للملفات و عمل مسح تلقائي لجميع الملفات على الهارد و من ثم تصنيفها مما يمكننا بعد ذلك من قراءة أنواع معينة منها ثم مقارنتها بنماذج قياسية لمعرفة هل هي ملفات ضارة أم لا و من ثم حذفها أو تعطيلها و في الحقيقة إن الجزء الصعب و المهم في الموضوع هو النصف الأول حيث أنك إذا ما تمكنت من مسح جميع الملفات و فك المضغوط منها و تحديد أنواع الامتدادات التي تريدها فإن فعلت ذلك تلقائيا فأنك تستطيع القيام بالإجراءات التي تريدها من مقارنات و حذف و تغيير في خصائص تلك الملفات

و سوف أقوم كذلك بشرح أسلوب تعامل الذاكرة مع أسلوب الاستدعاء الذاتي

و سيكون مرجعي في ذلك الماسترنج بتصريف مني لأصل إلى ما ذكرت من نتائج و الله المستعان

أولا لكي نعرف الفكرة التي يقوم عليها الاستدعاء التلقائي أريدك أن تضع تصورا لما نريد

إننا نريد أن ندخل إلى القرص الأول و نتجول بين المجلدات فأن وصلنا إلى مجلد نوقف الدالة ثم نفتح المجلد لنتجول بنفس الدالة بين ملفات المجلد و نحصل على بياناتها فأن وجدنا مجلد نوقف عمل الدالة و نفتح المجلد الثاني لنتجول بين ملفاته حتى إذا وجدنا نوع ملف نريده أوقفنا عمل الدالة ثم طبقنا بعض الأوامر ثم نستكمل حتى إذا انتهينا عدنا إلى الدالة السابقة فاستكملنا باقي المجلدات و هكذا إذا انتهينا رجعنا إلى الأولى ثم ننهي الدالة عندما نمسح جميع ملفات الدرايف أو محرك القرص الذي نمسحه

لعل الأمر يبدو معقدا قليلاً خاصة للمبتدئين و لكنه ليس كذلك و سيظهر كم الأمر بسيطاً بالمثال التالي

تخيل أنك تبحث عن كتاب في منزل و المنزل به عدد من الغرف و كل غرفة بها عدد من الصناديق و الكتب بعض الكتب خارج الصناديق و بعض الصناديق يحتوي على صناديق اصغر بالإضافة إلى الكتب

لا بد أنك ستدخل الغرفة الأولى و تفتح أول صندوق ثم تبحث في محتويات الصندوق فأن كانت كتب قرأت عناوينها و إن وجدت صندوق آخر توقفت عن القراءة ثم قمت بنفس العملية مع الصندوق الثاني

و هنا لنا وقفة..... إن قلنا أن فتح الصندوق و قراءة عناوين الكتب هو دالة

فأننا نستدعي تلك الدالة كلما وجدنا صندوقا حتى إذا لم نجد أي صناديق انهينا كل الدوال المفتوحة واحدة تلو الأخرى لنعود خارج الصندوق الكبير و قد استكشفنا كل ما فيه من كتب

حسنا لا بد من مثال بسيط يوضح لنا الأمر

هل تذكر ما هو المعامل

إن معامل الرقم n و الذي يرمز له بـ $n!$

يساوي $n \times (n-1)!$ (إلا معامل الصفر فهو دائما يساوي 1)
و بما أن معامل $(n-1)$
يساوي $(n-1) \times (n-2)!$ يساوي $(n-1) \times (n-2)!$
و هكذا فإن معامل الرقم n و الذي يرمز له بالرمز $n!$ يكون كالتالي :
 $4! = 4 \times 3!$
 $= 4 \times 3 \times 2!$
 $= 4 \times 3 \times 2 \times 1!$
 $= 4 \times 3 \times 2 \times 1 \times 0!$
 $= 4 \times 3 \times 2 \times 1 \times 1$
24

اعتقد انك فهمت ما اقصده تقريبا

إن المعادلة هي
معامل $n = n \times$ معامل $(n-1)$
و إذا كان $n = 0$ إذن معامل $n = 1$

و لعمل دالة تحسب معامل n
كود:

```
Function facNO(ByVal n As Integer) As Double
    If n = 0 Then
        facNo = 1
    Else
        facNo = n * facNo(n - 1)
    End If
End Function
```

كبرناها و هي صغيرة

إن الاستدعاء التلقائي هنا هو في السطر

كود:

```
facNo = n * facNo ( n-1)
```

في الواقع هنا سيتم الانتهاء من المعامل الخاص بالصفر أولا
ثم معامل الواحد ثم معامل الاثنين و هكذا

و السبب هو أن معامل الأربعة ينتظر و تنطلق الدالة التي تم استدعائها نفس الدالة تستدعي بطلب من نفسها)
و هي الدالة التي تحسب معامل الثلاثة ثم تنتظر و تنطلق دالة الاثنين و هكذا
حتى تنتهي دالة حساب معامل الصفر من عملها فتتبعها دالة حساب معامل الواحد و هكذا

و لكن ما علاقة هذا بمسح المجلدات

حسننا سنقوم بذلك في المرة القادمة حيث سنترج في الأمر حتى نضع برنامج يقوم بمسح جميع ملفات النظام و تصنيفه و فك ضغط الملفات المضغوطة و فحصها و تصنيفها بضغط زر واحد إن شاء الله

من الأفضل قبل كتابة أي إجراء تلقائي أن نكتب تصور عام للكود الخاص به
و بما أننا لن نحتاج إلى أن نعود بقيمة في التطبيق الذي سنبدأ به
و هو تطبيق يقوم بمسح الملفات على الجهاز
بما أننا لن نحتاج إلى أن نعود بقيمة في هذا التطبيق فأننا سنحتاج إلى روتين فرعي
و يعمل هذا الروتين على مسح محتويات مجلد و يكون الروتين كالآتي
sub ScanFolder (current_folder)
'بعض العمليات على ملفات هذا المجلد

كود:

```
if current_folder contains subfolders
for each subfolder
ScanFolder ( current_folder)
next
end if
```

هذا هو التصور المبدئي للكود
و لكي نقوم بتحويل هذا التصور إلى كود
سنحتاج إلى إنشاء مشروع تطبيق ويندوز
و نضيف زر و نسميه سكان و نضيف ليست بوكس لعرض محركات الأقراص التي ستظهر تباعا طبقا للمحرك الذي يتم عمل
سكان له
و نضيف ليست بوكس آخر للمجلدات و آخر للملفات

و الآن سنحتاج إلى القيام بالكود الآتي :
كود الزر سكان
كود:

```
for each drive in my pc
scanFolder(this drive)
next
```

طبعا هذا تصور للكود

و الآن مع الروتين **scanFolder**
كود:

```
for each directory in this drive
ScanFolder ( current_folder)
next
```

أما عن الروتين **ScanFolder** فقد وضعنا تصوره سابقا

و الآن مع الأكواد

الآن مع الكود المبدئي و لا اعتقد انه يحتاج إلى شرح و مع ذلك

كود:

```
Public Class Form1
```



```

Public curDrive As String
Public dir As String

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    'بالتتالي عليها العمل نبدأ و الأقراص محركات نستدعي هنا'

    Dim drives() As String
    drives = System.IO.Directory.GetLogicalDrives
    Dim iDrive As Integer

    'الملفات مع نتعامل كيف بعد نحدد لم لأننا نضعها '
    'المضغوطة و المغلقة'
    For iDrive = 0 To drives.GetUpperBound(0)

        curDrive = drives(iDrive)
        ListBox1.Items.Add(curDrive)
        'now scan folders
        scanFOLDER()
    Next

End Sub
Sub scanFOLDER()
    'بالتتالي مسحها نبدأ و محرك لكل الرئيسية المجلدات نستدعي هنا'
    Dim directories() As String
    directories = System.IO.Directory.GetDirectories(curDrive)

    'الملفات مع نتعامل كيف بعد نحدد لم لأننا نضعها '
    'المضغوطة و المغلقة'

    For Each dir In directories
        ListBox2.Items.Add(dir)
        'now scan files or sub folders
        scanFiles(dir)
    Next

End Sub

Sub scanFiles(ByVal ddir As String)
    'رئيسي مجلد لكل الملفات و المجلدات لمسح الذاتي الاستدعاء نستخدم هنا'
    Dim rupt As Boolean

    If Button1.Text = "scan" Then
        Button1.Text = "stop scan"
        rupt = False
    Else

        Button1.Text = "scan"
        rupt = True
    End If

    If rupt Then
        Exit Sub
    End If
    Dim ddirr As String
    Dim ffile As String
    Dim fi As IO.FileInfo
    'الملفات مع نتعامل كيف بعد نحدد لم لأننا نضعها '
    'المضغوطة و المغلقة'

```

```
For Each ffile In System.IO.Directory.GetFiles(ddir)
    fi = New IO.FileInfo(ffile)
    ListBox3.Items.Add(fi.Name & vbTab & fi.Length & vbTab &
fi.CreationTime)

Next
For Each dirr In System.IO.Directory.GetDirectories(ddir)
    Application.DoEvents()
    scanFiles(dirr)
Next

End Sub
End Class
```

بالطبع هذا المثال ليس هو المثال النهائي
و لكننا نطمح أن نجرى بعض العمليات على أنواع معينة من الملفات أثناء المسح
فبدائية لابد من فك ضغط الملفات المضغوطة و التعامل مع ملفات القراءة فقط
ثم تحديد أنواع الامتدادات المرئية و عرضها على قاعدة بيانات أو ملف يحتوى على اكواد الفيروسات المعروفة لدينا و من ثم
مقارنتها بهذه الاكواد و تحديد إذا كانت فيروسات أم لا
و من ثم حذفها أو تعديل اكوادها إن أمكن
و في الحقيقة إننا لا اعرف أن كنت سأستمر حتى نهاية الموضوع أم لا
و لكن على أي حال سأضع فكرة مبدئية عن كيفية التعامل مع الملفات و يمكن لكم استكمال الموضوع إن شاء الله

[ScanFiles1.rar](#)

الوافي في تحريم البرامج للدوت نت

الكاتب : [mero5000](#)

رابط المقال و النقاش حوله : [الجزء الأول](#) [الجزء الثاني](#)

سوف اشرح طريقه تحريم البرامج وعمل تنصيب لمشاريعنا سواء كانت vb.net أو # C ودمج الملف التنفيذي وقاعدة البيانات و ملفات dll والاسكين وغيرها بمشروعك وغيرها :



إنشاء odbc برمجيا

الكاتب : shenosheno2008

[رابط المقال و النقاش حوله](#)

إنشاء odbc عند عمل الاتصال بالبرنامج برمجيا

أولا:

كود:

```
Imports Microsoft.Win32
```

ثانيا قم بتعريف المكتبة الدالة التالية

كود:

```
Private Declare Function SQLConfigDataSource Lib "ODBC32.dll" (ByVal  
hwndparent As Integer, ByVal frequest As Integer, ByVal lpszDriver As String,  
ByVal lpszAttributes As String) As Integer
```

من خلال الإجراء التالي يتم إنشاء odbc سواء قاعدة البيانات على هذا الجهاز أم على السيرفر

trusted

or

not trusted

هذا هو الإجراء

كود:

```
Function create_odbc_dsn(ByVal server_name As String, ByVal dsn_name As String,  
ByVal driver_name As String, ByVal database_name As String, ByVal description As  
String, ByVal user_id As String, ByVal trusted As Boolean) As Boolean  
Dim atrbute As String  
atrbute = ""  
atrbute = atrbute + "SERVER=" &  
atrbute = atrbute + server_name &  
atrbute = atrbute + Chr(0) &  
atrbute = atrbute + "DSN=" &  
atrbute = atrbute + dsn_name &  
atrbute = atrbute + Chr(0) &  
atrbute = atrbute + "DESCRIPTION=" &  
atrbute = atrbute + description &  
atrbute = atrbute + Chr(0) &  
atrbute = atrbute + "DATABASE=" &  
atrbute = atrbute + database_name &  
atrbute = atrbute + Chr(0) &  
If trusted = False Then  
atrbute = atrbute + "TRUSTED_CONNECTION=NO"  
Else  
atrbute = atrbute + "TRUSTED_CONNECTION=YES"End If
```

```

End If
atrbyte = atrbyte + Chr(0)
If SQLConfigDataSource(0, 4, driver_name, atrbyte) = 1 And
Me.add_userid(user_id, dsn_name, trusted) = True Then
    Return True
Else
    Return False
End If
End Function

```

في حالة **not trusted** يحتاج إلى تعريف مستخدم كالتالي :

كود:

```

Private Function add_userid(ByVal userid As String, ByVal dsn_name As String,
ByVal trusted As Boolean) As Boolean
    Try
        If trusted = True Then
            Return True
        End If
        Dim regkey, regsubkeysw As RegistryKey
        Dim regsubkeyodbc, regsubkeyodbcini As RegistryKey
        Dim regsales As RegistryKey
        regkey = Registry.LocalMachine
        regsubkeysw = regkey.OpenSubKey("SOFTWARE")
        regsubkeyodbc = regsubkeysw.OpenSubKey("ODBC")
        regsubkeyodbcini = regsubkeyodbc.OpenSubKey("ODBC.INI")
        regsales = regsubkeyodbcini.OpenSubKey(dsn_name, True)
        regsales.SetValue("LastUser", "sh")
        Return True
    Catch ex As Exception
        Return False
    End Try
End Function

```

الكلاس في الرابط المرفق يمكنك في شاشة الاتصال تعريف نسخة من الكلاس وتمير متغيرات الاتصال إلى الإجراء

create_odbc_dsn

[Cls create_odbc.rar](#)

برمجة سعيدة مع Implicit Line Continuation

الكاتب : [النبهاني](#)

[رابط المقال و النقاش حوله](#)

السطر يصبح سطرين بدون "_" : ذكاء أم تشويه للغة ؟

منذ البداية ، كان الفرق الجوهرى في الصيغة Syntax بين لغتنا الحبيبة Visual Basic وكل أخواتها من عائلة Basic واللغات الأخرى ك C/C++,Java وغيرها ، هو طريقة التعدد في كتابة السطر الواحد في سطور مختلفة ، فلا حاجة إلى إنهاء كل سطر بالعلامة " ; " لكي يعرف سيادة المترجم Compiler بأن السطر انتهى هنا ، بينما في Visual Basic فلا داعي لذلك ، كل سطر ينتهي بنفسه افتراضيا طالما لم ينته بالعلامة "_" والتي تدل على أن السطر مقسم على اسطر متعددة كهذا الكود مثلا:

كود:

```
x = x ^ 2 + 3 * x _  
+ 3
```

تلك السمة كانت العلامة الفارقة دوما ، لكن جنون مايكروسوفت قادنا إلى أمر جديد كالعادة ! ما رأيك بهذا:

كود:

```
x =  
x ^ 2 + 3 * x + 3
```

لسطر الثاني والثالث صحيحان ١٠٠% وبدون أخطاء ابتداء من الإصدار ٢٠١٠ !! المزيد من الأسطر الغريبة بالنسبة لك تجدها هنا:

كود:

```
<Extension()>  
Function FilterByCountry(  
    ByVal customers As IEnumerable(Of Customer),  
    ByVal country As String) As IEnumerable(Of Customer)  
    Dim query =  
    From c In customers()  
    Where(c.Country = country)  
    Select <Customer>  
        <%=  
            c.Name &  
            ", " &  
            c.Country  
        %>  
</Customer>  
    Return query  
End Function
```

الكود السابق وبأكمله لا يوجد به ولا خطأ !

إذا ماذا عملت مايكروسوفت في هذا الإصدار ؟ كما يتضح من الكود السابق ، أصبح بإمكانك أن تتغاضى عن إدراج العلامة
"_" في الحالات التالية:

-بعد المواصفات **Attributes** كما يتضح في السطر الأول

-بعد علامة "(" عند تمرير قيم البارامترات.

-بعد علامة "," عند تمرير قيم البارامترات.

-بعد معامل المساواة "="

-قبل علامة ")"

-بعد علامة "<%" وقيل علامة ">%" في صيغ الاستعلامات. **Queries Expressions**

-بعد المعامل "&" في استعلامات **XML**.

قد يبدو هذا كثيرا ، لكن الأمر لن ينتهي عند هذا الحد ، فالمزيد من الحالات التي يمكنك فيها توديع علامة المتابعة "_"
موجودة وهي جاهزة وملخصه في الرابط التالي:

<http://msdn.microsoft.com/en-us/library/865x40k4.aspx>

برمجة سعيدة مع **Implicit Line Continuation** لكن هل تم تشويهه اكواد **Visual Basic** بهذا الشكل ؟ فعلا هي و
Lambda Expressions و **LINQ** غيرت الكثير من معالم الأكواد!

جديد : 2010 اكتب الكود الذي سيستخدم فنتك ثم قم بإنشائها :

الكاتب : [samerselo](#)
[رابط المقال و النقاش حوله](#)

من المميزات الجديدة المقدمة ضمن فيجوال أستوديو 2010 هي إمكانية كتابة الكود حسب احتياج برنامجك ثم إنشاء الفئة اعتمادا على ذلك الكود وهذا ضمن لغتي فيجوال بيسك و سي شارب على حد سواء
أنشئ مشروعا جديدا وضع على نافذة المشروع صندوق نصوص و زر ثم انقر نقرا مزدوجا على الزر للانتقال إلى محرر الكود
اكتب الكود التالي في إجراء الحدث :

كود:

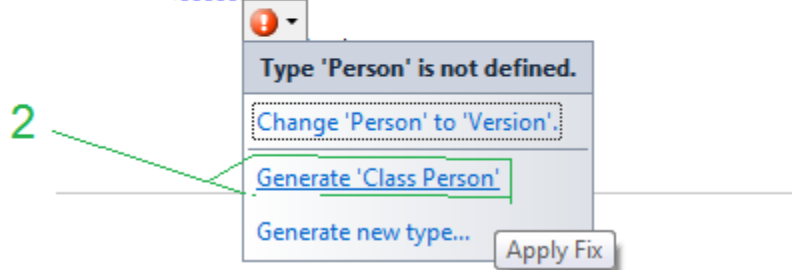
```
Dim sa As New Person(PerName:="Samer", Age:=38)
```

ستلاحظ ظهور خط أزرق متعرج تحت **Person** وذلك لأن الفئة **Person** غير موجودة قف قليلا فوق **Person** بمؤشر الماوس فيظهر لك قائمة تصحيح الأخطاء انقر فوق الخيار **Generate Class Person** حتى تقوم بيئة التطوير بإنشاء الفئة من أجلك حسب طريقة استخدامك لها :

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click  
Dim sa As New Person(PerName:="Samer", Age:=38)
```



```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click  
Dim sa As New Person(PerName:="Samer", Age:=38)
```

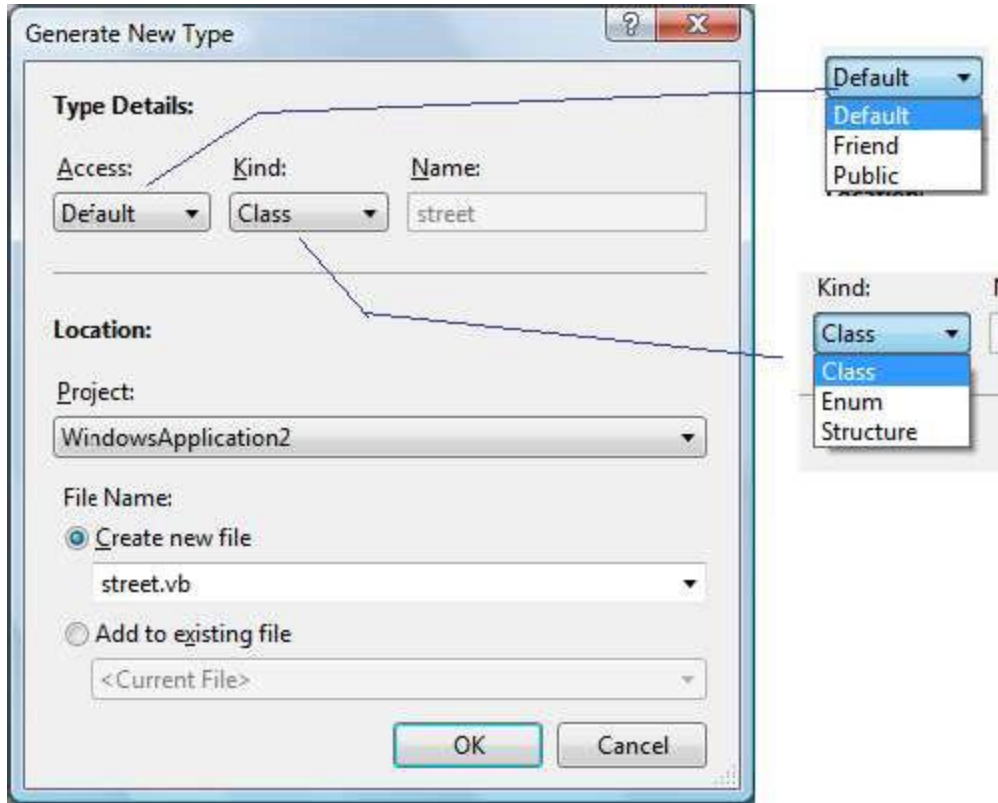


سيتم الآن إضافة الملف **Person.vb** لمشروعك وهو يحتوي على الفئة الجديدة افتح الملف وانظر للكود الذي قامت بيئة التطوير بتوليده من أجلك :
كود:

```
Class Person  
Private _age As Integer  
Private _perName As String  
Sub New(ByVal PerName As String, ByVal Age As Integer)  
' TODO: Complete member initialization  
_perName = PerName  
_age = Age  
End Sub  
End Class
```

يمكننا توليد الفئة من الكود وذلك باختيار **Generate new type** بدلا من **generate class person** عندها قبل أن تُم إنشاء الفئة .

يتم إظهار صندوق حوار يمكننا من اختيار نوع الفئة وخصائصها التي نريد قبل أن تقوم بيئة التطوير بعملية الإنشاء الفعلي للفئة



نريد الآن إضافة وظيفة للفئة تعيد سنة ميلاد الشخص فنعود لمحرر كود الزر الخاص بالنافذة ونضيف تحت الكود السابق السطر التالي :

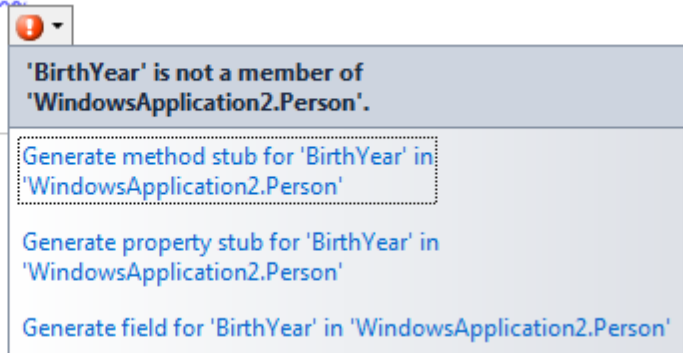
كود:

```
Me.TextBox1.Text = sa.BirthYear
```

وبنفس الطريقة نختار الأمر.... **Generate method stub** فيتم إنشاء الوظيفة المناسبة

```
Me.TextBox1.Text = sa.BirthYear
```

```
Sub  
End Sub
```



انتقل للملف **person.vb** ولاحظ الكود المتولد

كود:

```
Function BirthYear() As String  
    Throw New NotImplementedException  
End Function
```

عدل الوظيفة السابقة حتى تصبح كمايلي :
كود:

```
Function BirthYear() As String
    Return DateAdd(DateInterval.Year, (-1 * _age), Now)
End Function
```

أو نريد إنشاء خاصية **Address** لفننتنا نعود أيضا لمحرر كود الزر في النموذج ونضيف السطر التالي :

كود:

```
sa.Address = "Damascus Sysria"
```

وبنفس الطريقة نختار **Generate Property ...** فيتم إضافة الخاصية للفئة **Person** وهذا سرد للفئة **Person** حتى هذه النقطة

كود:

```
Class Person
    Private _age As Integer
    Private _perName As String
    Sub New(ByVal PerName As String, ByVal Age As Integer)
        ' TODO: Complete member initialization
        _perName = PerName
        _age = Age
    End Sub
    Property Address() As String
    Function BirthYear() As String
        Return DateAdd(DateInterval.Year, (-1 * _age), Now)
    End Function
End Class
```

شرح الوافي في حماية البرنامج عبر التسجيل و التفعيل

الكاتب : [نور نبهان](#)
[رابط المقال و النقاش حوله](#)

تحية طيبة وبعد :

سنتناول في هذا الموضوع طريقة تسجيل البرنامج وتفعيله و الذي تم تناوله سابقا في الفيچوال بيسك ٦ ورغم الوقت بين الموضوعين لم أجد موضوع يطرح نفس الفكرة تماما وإنما بتشابه وبعض التعقيد لذلك سأعيد بناء الفكرة و أطرحها في هذا الموضوع إن شاء الله من جديد بأسلوب بسيط ومطول قليلا من الناحية النظرية ليتم استيعابها بالشكل الكافي .

بداية وقيل كل شيء الطريقة مطروحة لأسباب تعليمية وطبعا لن أناقش فعالية الحماية أو قوة الكسر لان الأمر مفروغ منه ولا أحب الولوج فيه لكن إن شاء الله سيتم **تغطية** بعض الجوانب الوقائية لهذا الموضوع .

بسم الله نبأ:

سنعتمد في برنامجنا الذي سنقوم بتصميمه طريقة الحماية عبر تسجيل وتفعيل البرنامج وهي أن برنامجنا عند تشغيله يظهر للمستخدم رقم ويطلبه بإدخال الرقم المقابل له وهذا الرقم طبعا سيكون فريد أو مميز أي يختلف من جهاز إلى آخر ولنا الحرية في حال اعتماد رقم قطعة في الجهاز أو دالة معينة وبعد ظهور الرقم للمستخدم يتم إرساله لك لتوليد الرقم المقابل له و إرساله للعميل ليعمل البرنامج إذا كان الرقم مطلوب بمجرد التشغيل أو ليقوم بتفعيل كامل الميزات إذا كنت عرضته بطريقة تعطيل بعض النواظ أو الخصائص.

إذا الفكرة الأساسية أن برنامجنا يطلب رقم ليصبح نسخة كاملة .

بداية نستخرج رقم فريد من جهاز الحاسوب بحيث يكون هذا الرقم يختلف بين كل حاسب و آخر كرقم القرص الصلب أو المعالج أو ... ونقوم بعمليات تمويه له لكي لا يظهر مكشوف كما هو فينتج رقم التسجيل للمستخدم.

الآن نقوم بعمليات تمويه أخرى لرقم التسجيل لنشتق رقم التفعيل فالبرنامج يحوي رقم القطعة أي إحدى عتاد الحاسوب ورقم تسجيل ورقم تفعيل فرق القطعة لا علاقة لنا به لأننا بعد استخراج عدلناه ليظهر بطريقة أخرى والنتائج سيكون رقم التسجيل الآن نعيد نفس الكرة ونعدل التسجيل لاستخراج رقم التفعيل وبذلك برنامجنا يجوي الأرقام الثلاثة و الظاهر منها هو رقم التسجيل فقط لدى العميل.

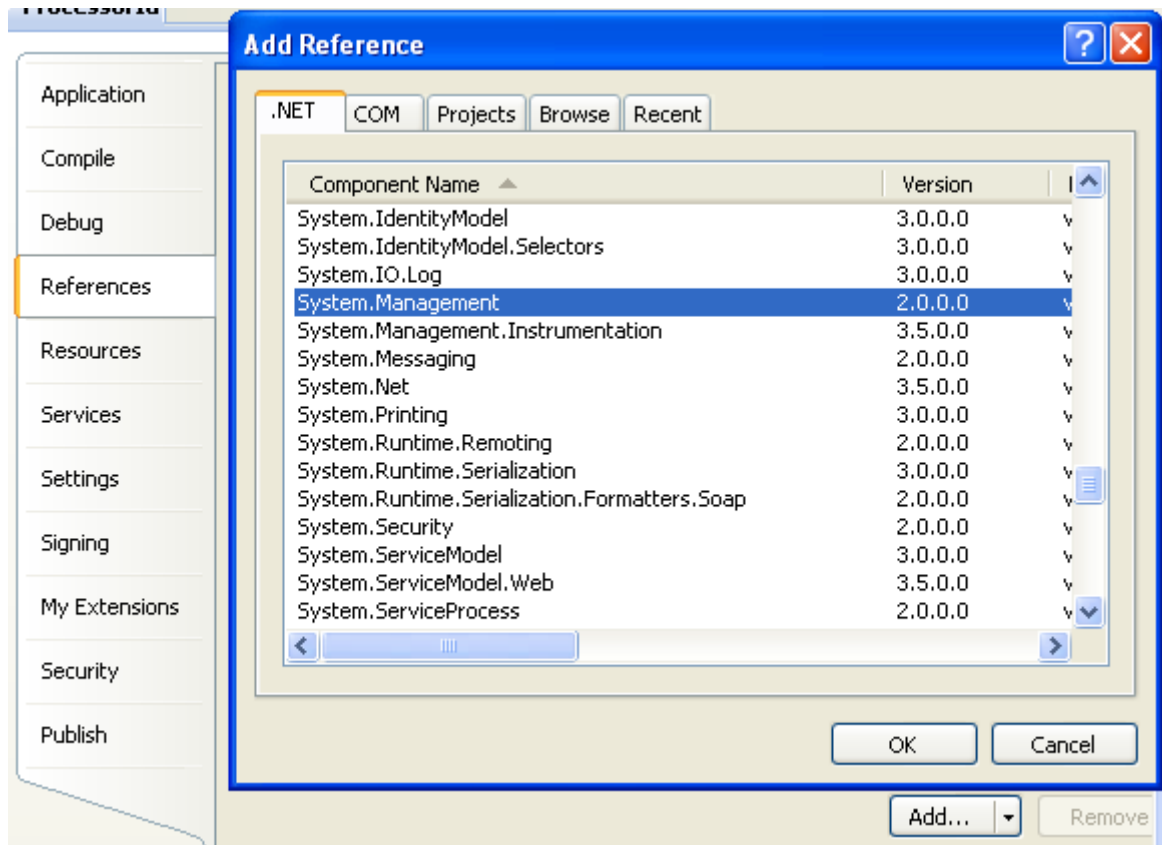
سيقوم العميل بإرسال رقم التسجيل لنا لنستخرج رقم التفعيل بنفس الطريقة التي استخدمناها في برنامجنا ونرسله له وعندها تتم المقارنة بين الرقمين وكسر البرنامج من قبل المهندس العكسي عفوا أقصد المقارنة بين الرقمين وتفعيل البرنامج في حال التساوي.

الطريقة التي سنتبعها في درسنا هي رقم المعالج البر و سيسيور.

زمن vb6 كنا نستخدم مكتبة ميكروسوفت **WbemScripting** لاستخراج رقم المعالج وفي الدوت نت سنستخدم **System.Management.dll** لاستخراج الرقم و المكتبة نستطيع من خلالها استخراج مواصفات عتاد الجهاز لكم كبير من هذا العتاد ما يهمنا في موضوعنا هو رقم المعالج كونه فريد ونريد الاعتماد عليه في تغيير الرقم لكل مستخدم لبرنامجنا.

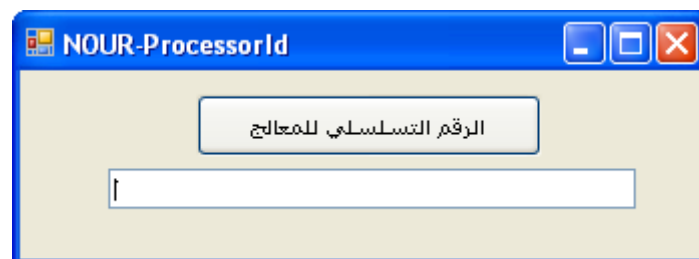
افتح **Microsoft Visual Studio 2008** و أنشأ مشروع جديد وسمه ما شئت

توجه الى نافذة **Solution Explorer** واضغط عل المشروع بالزر الأيمن واختر الخصائص **Properties** ادخل إلى التبويب **References** و أضف **Add References** المكتبة **System.Management** كما في الصورة التالية:



بعد إضافة المكتبة توجه للنموذج أي الفورم و أضف صندوق نص تحت مسمى **TextBoxId** وزر أمر اسمه **GetIdButton**

التصميم :



الآن ضغطتین متتاليتين على زر الأمر لإدخال الكود المناسب لاستخراج رقم المعالج ووضعه في صندوق النص :
وبالأحرى هذا الكود كاملا أولا استيراد لوظائف المكتبة وبعدها كود الزر:

كود:

```
Imports System.Management
Public Class Form1
    Private Sub GetIdButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GetIdButton.Click
        Dim Searcher As ManagementObjectSearcher
        Searcher = New ManagementObjectSearcher("Select ProcessorId From Win32_Processor")
        For Each Device As ManagementObject In Searcher.Get
            For Each Prop As PropertyData In Device.Properties
                TextBoxId.Text = (Prop.Value.ToString)
            Next
        Next
    End Sub
End Class
```

End Sub
End Class

قم بتشغيل البرنامج و أضغط على الزر وستجد بإذن الله رقم المعالج يطبع في صندوق النص.

بالنسبة لشرح الكود السابق :

في البداية استيراد فضاء المكتبة وفي كود الزر قمنا بالبحث في مواصفات المعالج **Processor** عن قيمة **Value** الخاصة بـ **ProcessorId** بواسطة حلقة **For Each** وعند إيجادها نستخرجها في صندوق النص.

ربما يسأل البعض لماذا لا نعرض الرقم مباشرة بدون الحلقة والبحث فيها فالجواب سيكون لان رقم الخاصية بين خصائص المعالج لن تكون الأولى وربما يختلف ترتيبها من جهاز لآخر لذلك نحن نبحث عنها ضمن المواصفات كلها وعند إيجادها يتم قنص القيمة و وضعها في صندوق النص.

الآن نستطيع استخدام رقم المعالج كما هو لعمليات التسجيل والتفعيل بحيث يكون الرقم مباشرة رقم تسجيل ورقم التفعيل يكون بإضافة بعض أرقام أو معادلة له لكن بهذه الطريقة إذا كان المستخدم لديه بعض البديهة فسينتبه إلى الرقم ويكتشف انه رقم المعالج وعندها يعرف التغير الذي حصل لرقم التفعيل بحذف رقم المعالج من السلسلة لذلك سوف نقوم بتشفير الرقم و إظهاره بطريقة تجعل المستخدم لا يعرف ما هو الرقم الذي اعتمدنا عليه لذلك الآن نتجه لإنتاج رقم التسجيل والتفعيل .

أدرج صندوقي نص جديدين تحت مسميات **TextBoxReg** و **TextBoxAct** ليصبح التصميم:



الآن سنتجه لاستخراج رقم التسجيل والتفعيل عبر تحويل أو تشفير السلسلة النصية التي تعبر عن رقم المعالج و التي تظهر في صندوق النص الأول **TextBoxId** سأستخدم دالة التحويل إلى الاسكي (النتاج سيكون رقمي) وهذه طريقة سهلة لتلقين رقم التفعيل للعميل كونها محصورة بالشكل الرقمي من ٠ إلى ٩ فقط وبعيدة عن الرموز أو الأحرف

أضف الدالة التالية لبرنامجك :

كود:

```
Function Str2Int(ByVal InStrng As Object) As String
    Dim StrLn As Integer
    Dim Cntr As Integer
    Dim NewStr As String

    Str2Int = ""
    StrLn = Len(InStrng)
    If StrLn = 0 Then Exit Function
    NewStr = ""
```

```

For Cntr = 1 To StrLn
    Select Case Mid(InStrng, Cntr, 1)
        Case "0" To "z"
            NewStr = NewStr & Asc(Mid(InStrng, Cntr, 1))
    End Select
Next Cntr
Str2Int = NewStr
End Functio

```

و أضف في آخر كود الزر السطرين التاليين :

كود:

```

TextBoxReg.Text = Str2Int(TextBoxId.Text)
TextBoxAct.Text = Str2Int(TextBoxReg.Text)

```

أي يصبح الكود كاملاً :

كود:

```

Imports System.Management
Public Class Form1
    Private Sub GetIdButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GetIdButton.Click
        Dim Searcher As ManagementObjectSearcher
        Searcher = New ManagementObjectSearcher("Select ProcessorId From
Win32_Processor")
        For Each Device As ManagementObject In Searcher.Get
            For Each Prop As PropertyData In Device.Properties
                TextBoxId.Text = (Prop.Value.ToString)
            Next
        Next
        TextBoxReg.Text = Str2Int(TextBoxId.Text)
        TextBoxAct.Text = Str2Int(TextBoxReg.Text)
    End Sub
    Function Str2Int(ByVal InStrng As Object) As String
        Dim StrLn As Integer
        Dim Cntr As Integer
        Dim NewStr As String

        Str2Int = ""
        StrLn = Len(InStrng)
        If StrLn = 0 Then Exit Function
        NewStr = ""
        For Cntr = 1 To StrLn
            Select Case Mid(InStrng, Cntr, 1)
                Case "0" To "z"
                    NewStr = NewStr & Asc(Mid(InStrng, Cntr, 1))
            End Select
        Next Cntr
        Str2Int = NewStr
    End Function
End Class

```

شغل البرنامج وجرب الضغط على الزر وسيظهر أمامك الشكل التالي :

فالرقم الأول هو رقم المعالج سنخفيه عن العميل

الرقم الثاني هو رقم التسجيل مشتق بتحويل رقم المعالج إلى قيم الاسكي وهو سيظهر للعميل ويرسله لنا

الرقم الثالث رقم التفعيل مشتق من رقم التسجيل بنفس الطريقة الأولى وهو سيكون مخفي للعميل وهذا الرقم سيكون لدينا أيضا بالقيام بعملية تحويل التسجيل إلى اسكي أي قيم رقمية لإرساله إلى العميل وعندما يدخله تتم المقارنة بينه وبين رقم التفعيل المخفي عنه إذا تساوى الرقمين تتم العملية وإلا فلا

وهذا الشكل يوضع العملية:

فالمربعات ذات اللون الأسود لن تظهر للعميل وإنما سيظهر فقط رقم التسجيل الذي سيرسله لنا أي المربع الأبيض الأول ومربع يطلب منه إدخال رقم التفعيل المرسل له من قبلنا وهو المربع الأخير

وبعد إدخال الرقم تتم المقارنة بين الرقمين في المربعين الأخيرين في حال التساوي تتم العملية وإلا فلا

بالنسبة لطريقة التشفير فكما تلاحظون حاولنا استخدام دالة التحويل لقيم الاسكي للمفاتيح ويمكنكم اعتماد أي طريقة لتشفير الرقم حيث هناك طرق لا حصر لها يمكننا إجراء غريبة للرقم أو الإقتصاص منه أو الزيادة عليه أو تعديله بإضافة واحد أو أكثر أو تشويش السلسلة كلها أو أخذ مواضع محددة من السلسلة أو.....الخ

...

الآن طريقة المقارنة بين رقم التفعيل المخفي و الرقم الذي أدخله العميل

طبعا وجود المقارنة بحسب معلومات المهندس العكسي يسهل تجاوز البرنامج وكسره

الطريقة المحلية هي الآتي :

أضف زر أمر وسمه **ButtonOk** وصندوق النص الرابع الذي سيدخل العميل فيه الرقم سمه **TextBoxUser**

و أضف كود الزر **ButtonOk** التالي:

كود:

```
Private Sub ButtonOk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonOk.Click
    If TextBoxUser.Text = TextBoxAct.Text Then
        MsgBox("لك شكراً البرنامج تفعيل تم")
    Else
        MsgBox("خاطى المدخل الرقم")
    End If
End Sub
```

الآن جرب لترى النتيجة :



نحن اعتمدنا إظهار رسالة للعميل تفيد بأن الرقم صح أو خطأ أما بالشكل العملي فالأمر سيكون بتفعيل فتح نوافذ واستخدام إمكانيات كالتباعة مثلاً أو فتح البرنامج إذا كان التفعيل يظهر عند إقلاع البرنامج مباشرة لذلك نحن نستعوض بعبارة

كود:

```
MsgBox("لك شكراً البرنامج تفعيل تم")
```

بالإجراء المطلوب تنفيذه عند إدخال القيمة الصحيحة وإلا لن يظهر له شيء لان المهندس العكسي ينصحنا بذلك

كود:

```
If TextBoxUser.Text = TextBoxAct.Text Then
    MsgBox("لك شكراً البرنامج تفعيل تم")
End If
```

ومن نصائحه لنا هي أن لا نستخدم **MsgBox** ونحن أيضاً سنلتزم بأوامره فكما ذكرت سابقاً الرسالة لإيضاح نجاح العملية أما الأصل فهو استخدام الإجراء المناسب كفتح البرنامج أو تفعيل الميزات المحجوبة

ربما يسأل البعض ما دخل المهندس العكسي في الموضوع أقول أنه ربما يفكر في الكسر لذلك نحن نريده أن يكسب الجائزة بعرق جبينه وليس بسهولة لذلك سنجعل عملياتنا مخفية عليه بعض الشيء لنكسبه بعض الخبرة ويكون الكسر تم بتعبه وعرق جبينه و إلا فنحن نتحرى الخير في الناس وفي عملاتنا أنهم لا يحاولوا كسر برامجنا ويتركونا نتسبب.

نتابع مع بعض النصائح : لا تجعل مكان التفعيل هو نفسه مكان المقارنة وإعطاء الصلاحيات

ما المقصود بذلك ؟

المقصود أحبتي : أننا عند تفعيل برنامجنا بشكله الصحيح فإنه لا يتم تخزين الرقم الخاطئ و يتم تخزين الرقم لأنه صحيح فقط أما تفعيل الميزات فحتاج لمقارنة أخرى في فورم أخر بعيدا عن مكان التفعيل بشرح بسيط لو أدخل المستخدم رقم التفعيل الصحيح ستظهر رسالة تفيد بذلك وبنفس الوقت نخزن الرقم الصحيح في مكان معين (البرنامج- الريجستري-ملف) وعند إقلاع البرنامج يتم المقارنة مرة أخرى بنفس الطريقة لان البرنامج ربما يتم تجاوزه من خلال الهندسة العكسية ويتم تدمير قسم المقارنة لذلك المقارنة الأساسية تتم في مكان أخر وواجهة التفعيل هي فقط لتخزين الرقم الصحيح وعند إقلاع البرنامج تتم المقارنة بين رقم التفعيل والرقم المخزن من جديد صح يفعل الميزات وإلا فلا وهكذا نقوم بحشو برنامجنا بعشرة مقارنات ونضع عشرة مقارنات أخرى احتياط مثل بعد التشغيل الخامس للبرنامج في اليوم الأول من كل شهر وبعد شهرين بذلك نضمن أن قفزات الجمل الشرطية مهما حطم منها يبقى شيء مفعوله موجود

الهدف : بهذه الطريقة التمويهية نجعل أمر تجاوز المقارنات صعب بالتالي لن يكسر وطبعاً هنا لن نسبية بطريقة القفز وتدمير المقارنات ماذا لو فكر بتدمير جميع المقارنات فالأمر صعب لان برنامجك ربما يحوي مقارنات عادية كالبحت وغيره بالتالي لن يعمل البرنامج بشكل صحيح فليس أمامه في طريقة القفز إلا العمل على البرنامج لمدة سنة كاملة وبعدين يعطينا نسخة مكسورة تعمل بطلاقة .

الطريقة الثانية للمهندس العكسي : هي استنتاج الرقم طبعا سيكون هذا الأمر مجدي له أكثر بخلاف العادة فبمجرد الوصول لأول مقارنة بين رقم التفعيل و الرقم المدخل سيكشفه إذا كان ذكي قليلا حسب معلوماتي المتواضعة عن هذه الأمور.

لكن هذه الطريقة تفتح أمامه مجالين :

الأول : أنه سيكشف الرقم للجهاز الذي يحاول الكسر عليه بالتالي يجب عليه الذهاب لكل جهاز والكسر عليه بهذه الحالة نضمن أن النسخ المكسورة ستكون أقل ما يمكن لأنه مش معقول يحمل العدة ويوجب على المستخدمين ياهو كسر برنامج فلان

الثاني : هو الخطير لو استطاع كشف المعادلة بالتالي يستطيع عمل مولد أرقام مثل الذي لدينا وهذا سنحاول تجنبه بتعقيد المعادلة المخصصة لاستخراج رقم المعالج وتشفيره بحيث لا يستطيع فك عمليات التشفير على الرقم.

أتوقع كافي نظري حول تجنب الكسر و لا أقول لا أو انه صعب الكسر فالسوق أكبر دليل لكن كان سعبي لتوليد فكرة حول الأمر للجميع حتى يناقشوه في برامجهم ويستخدموه بالشكل الأمثل الذي يناسبهم.

نعود لكودنا في الزر طبعا الدالة يمكنك أن ترميها في موديول وتخلص من رؤيتها أمامك لأنها جاهزة ولن نعدل عليها وإنما سنستدعيها فقط لذلك أول شيء قم بقص الدالة و الصقها في الموديول .

الآن سنجعل رقم المعالج يظهر بمجرد فتح الفورم طبعا فقط ننقل الكود من حدث ضغط الزر `GetIdButton_Click` إلى تحميل الفورم `Form1_Load` ولسنا بحاجة لأكثر من صندوقين نص واحد تسجيل والأخر تفعيل والقيم الوسطى نخزنها ضمن متغيرات `idp2 idp1 idp` فيصبح الكود كالآتي :

كود:

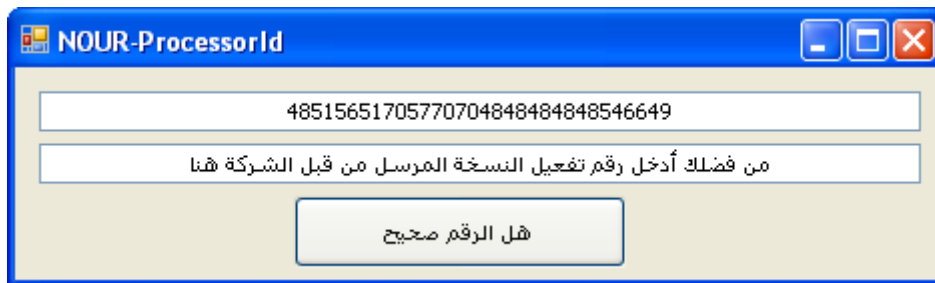
```

Public Class Form1
    Dim idp As String ' المعالج رقم
    Dim idp1 As String ' التسجيل رقم
    Dim idp2 As String ' التفعيل رقم

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim Searcher As ManagementObjectSearcher
        Searcher = New ManagementObjectSearcher("Select ProcessorId From
Win32_Processor")
        For Each Device As ManagementObject In Searcher.Get
            For Each Prop As PropertyData In Device.Properties
                idp = (Prop.Value.ToString)
            Next
        Next
        idp1 = Str2Int(idp)
        idp2 = Str2Int(idp1)
        TextBoxReg.Text = idp1
    End Sub
    Private Sub ButtonOk_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonOk.Click
        If TextBoxUser.Text = idp2 Then
            MsgBox("لك شكراً البرنامج تفعيل تم")
        End If
    End Sub
End Class

```

لاحظ أننا الآن بحاجة لصندوق نص هما **TextBoxReg** يظهر فيه رقم التسجيل و **TextBoxUser** لإدخال رقم التفعيل من قبل العميل أما البقية والزر لا تلزم لأن القيم صارت ضمن متغيرات ولا ننسى الدالة ألصقتها بالموديول من قبل فشكل نافذة التفعيل الآن هو الآتي :



لو انتبهت للكود فجملة **If** تناقش الحالة الصحيحة وإذا خطأ لن نقوم بأي عمل لأننا بحاجة توليد مسح بوكس أنه خطأ نسهل العملية بحسب النصائح السابقة وكما ذكرت أنفا المسح بوكس يجب الاستغناء عنه لأنه أولاً نقطة ضعف وثانياً لن يفيدنا تخيل انه المستخدم فعل البرنامج خلاص أغلق البرنامج وهو يشغله ويراه مفعّل أو أغلق فورم التفعيل و خزن الرقم أو حتى خزن الرقم وبس. وما في ضرورة نلقمه إياها بالملقعة ببب مسح بوكس ببب تم التفعيل يا عميل خلص أغلق فورمه التفعيل ورح من هنا ببب شكراً لك هيه صار عندك نسخة كاملة هذا الحشو مش مهم .

طيب الآن ربما يسأل السائل ربما يستنتج العميل أو حبيينا العكسي أن الرقم المستخرج هو رقم المعالج وبنظرة بسيطة بالمقارنة بين رقمي التسجيل و رقم المعالج سينتبه أن الثاني مشتق من الأول بالتحويل الرقمي كيف لتشابه أرقام في المعالج مع المقابل الرقمي في حال التكرار مثلاً لدي يظهر

رقم المعالج:

0383F9FF00006B1

رقم التسجيل:

48515651705770704848484848546649

فربما يفهم العملية لذلك سنحاول جعله لا يفهم العملية بالقيام بتشويش السلسلة النصية ومن ثم تحويلها إلى رقم وبذلك حتى الرقمين المتشابهين سيكون بالتشويش لديها قيم مختلفة :

لذلك أول شيء أذهب للموديول و الصق الدالة التالية أسفل الدالة السابقة صاحبة الاسكي ليصبح الموديول كاملا كالتالي:

كود:

```
Module Module1
    'الاسكي الى التحويل'
    Function Str2Int(ByVal InStrng As Object) As String
        Dim StrLn As Integer
        Dim Cntr As Integer
        Dim NewStr As String

        Str2Int = ""
        StrLn = Len(InStrng)
        If StrLn = 0 Then Exit Function
        NewStr = ""
        For Cntr = 1 To StrLn
            Select Case Mid(InStrng, Cntr, 1)
                Case "0" To "z"
                    NewStr = NewStr & Asc(Mid(InStrng, Cntr, 1))
            End Select
        Next Cntr
        Str2Int = NewStr
    End Function

    'التشويش دالة'
    Public Function Obfuscate(ByVal origText As String) As String
        Dim textBytes As Byte() =
System.Text.Encoding.Unicode.GetBytes(origText)
        For counter As Integer = 0 To textBytes.Length - 1
            If (textBytes(counter) > 31) And (textBytes(counter) < 127) Then
                textBytes(counter) += CByte(counter Mod 31 + 1)
                If (textBytes(counter) > 126) Then textBytes(counter) -=
CByte(95)
            End If
        Next counter
        Return System.Text.Encoding.Unicode.GetChars(textBytes)
    End Function
End Module
```

نشوش بالدالة بهذه الطريقة

ldp1 = Obfuscate(ldp)

الآن رقم المعالج هو

0383F9FF00006B1

بالتشويش بدالة Obfuscate يصبح

16=:ODSUACEGIQ_P

الآن نحوله إلى الرقمي

49546158796883856567697173819580

لاحظ من جديد هذا الرقم يختلف عن القيمة السابقة لأننا قمنا بتشويش رقم المعالج قبل تحويله إلى قيم الاسكي طبعاً أنت تستطيع التفعيل بالقيم الحرفية لكن الرقمية تبقى أسهل للتلقين كما ذكرنا أول الموضوع

حاول المقارنة بين اللونين الأحمرين في رقم المعالج ورقم التسجيل لن تستنتج شيء لان التشويش قضى على هذا الأمر. بينما كان الأمر جلي قبل التشويش.

لاحظ الفرق السابق فلا علاقة واضحة أبداً خمسة أصفار وكل شفرة لها مقابل مختلف هكذا يصبح الأمر مشفر أكثر
عد للنموذج وعدل الكود كالاتي :

كود:

```
Imports System.Management

Public Class Form1
    Dim Idp As String
    Dim Idp1 As String
    Dim idp2 As String

    Private Sub FrmActiv_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'للمعالج التسلسلي الرقم'
        Dim Searcher As ManagementObjectSearcher
        Searcher = New ManagementObjectSearcher("Select ProcessorId From Win32_Processor")
        For Each Device As ManagementObject In Searcher.Get
            For Each Prop As PropertyData In Device.Properties
                '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%'
                'الرقم استخراج'
                Idp = (Prop.Value.ToString)
            Next
        Next
        '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%'
        'الرقم تشويش'
        Idp = Obfuscate(Idp)
        'اسكي الى تحويله'
        Idp = Str2Int(Idp)
        'التسجيل رقم'
        TextBoxReg.Text = Idp
        '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%'
        'الرقم تشويش'
        Idp1 = Obfuscate(Idp)
        'اسكي الى تحويله'
        'التفعيل رقم'
        Idp1 = Str2Int(Idp1)
    End Sub

    Private Sub ButtonOk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonOk.Click
        'جمالي متغير'
        idp2 = Idp1
        If TextBoxUser.Text = idp2 Then
            MsgBox("لك شكراً البرنامج تفعيل تم")
        End If
    End Sub
End Class
```

اتضح الأمر أكثر الآن فنحن نقوم باستخراج رقم المعالج وتشويشه وتحويله إلى أسكي فينتج رقم التسجيل ونطبعه للمستخدم في صندوق النص ونطالبه بالمقابل الذي هو عبارة عن رقم التسجيل مشوش ومحول إلى أسكي من جديد وبعدها المقارنة في حدث الزر لن أكرر المسح بوكس بس للتوضيح حول المقارنة وليس للاستخدام.

الآن ربما تتلملم من أن رقم المعالج ١٦ رقم ينتج عنها رقم تسجيل ٣٢ رقم ورقم التفعيل سيكون بالتالي ٦٤ رقم فهذا كثير مع أنني أفضل طول الرقم لكننا سنكتفي بطول رقم التسجيل ٣٢ رقم ونقوم بتصغير رقم التفعيل إلى ١٤ رقم مثلاً وذلك بعملية اقتصاص من رقم التفعيل كالاتي وذلك بإضافة السطر التالي في آخر الفورم لوود (اقتصاص سلسلة نصية)

كود:

```
رقم 14 اقتصاص'  
Idpl = (Idpl.Substring(0, 14))
```

طبعا لك الحرية في اختيار طول الرقم بتغير الرقم ١٤ إلى أي رقم آخر لكن طول الرقم يصعب عملية الاشتقاق أصلاً.

كذلك يمكنك اختيار نقطة البدء من أي حرف .

الآن يسأل السائل إذا قام العميل بالتفعيل فهل سأقول له مبروك فقط أما ماذا يحدث

نبدأ من جديد :

عند إدخال الرقم الصحيح سنقوم بتخزينه وأكرر لك الحرية في اختيار موضع التخزين ولا يهم ظهور الرقم لأنه صحيح ونحن من زودناه به لذلك سألجأ إلى **Settings**

نتجه إلى **Solution Explorer** ونضغط على المشروع بالزر الأيمن ونختار **Properties** ونذهب لتبويب **Settings** ونضيف إليه متغير جديد سأسميه **nameuser** وليس **numact** شوية تمويه يبدو التجنيد مؤثر بي كثير.

اسم المتغير

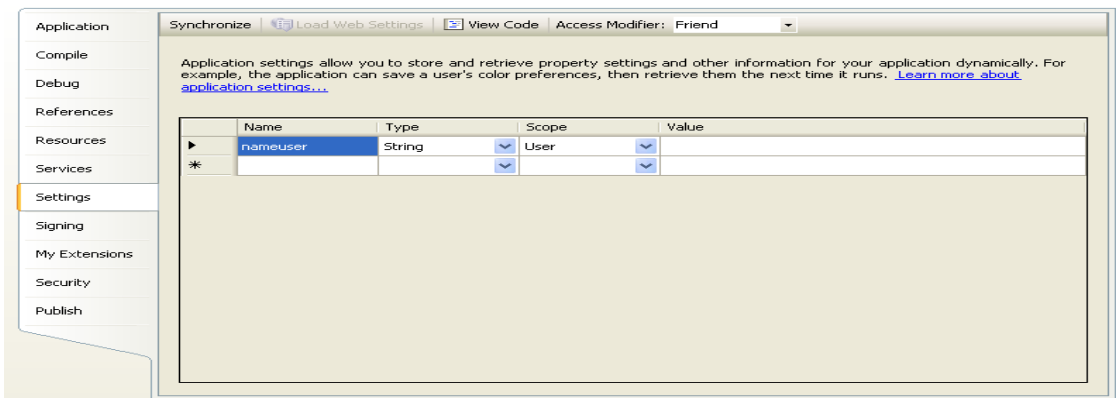
Name = nameuser

نوع المتغير

Type = String

قيمة المتغير

Value =



الآن سنتجه لكوود الزر ونغيره إلى

كوود:

```
Private Sub ButtonOk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonOk.Click
    جمالي متغير
    idp2 = Idp1
    If TextBoxUser.Text = idp2 Then
        My.Settings.nameuser = TextBoxUser.Text
        My.Settings.Save()
    End
End If
End Sub
```

لاحظ الكود السابق في حال التساوي فإنه يسند الرقم إلى الإعدادات ثم يحفظها ففضية التفعيل كلها هنا هي هل الرقم متساوي إذا كان الجواب نعم فعندها لا يفعل الميزات بل يحفظ الرقم فقط وعملية التفعيل في أمكنة أخرى بعيدا عن مركز اختبار الرقم الظاهر للعميل و بذلك لن يظهر للعكسي الأمر واضح كالعادة .

هل تم الأمر لا خطوة أخيرة فقط :

الآن أصبح كود مشروعنا الكامل هو الآتي :

الموديول :

كوود:

```
Module Module1
    ' الاسكي الى التحويل
    Function Str2Int(ByVal InStrng As Object) As String
        Dim StrLn As Integer
        Dim Cntr As Integer
        Dim NewStr As String

        Str2Int = ""
        StrLn = Len(InStrng)
        If StrLn = 0 Then Exit Function
        NewStr = ""
        For Cntr = 1 To StrLn
            Select Case Mid(InStrng, Cntr, 1)
                Case "0" To "z"
                    NewStr = NewStr & Asc(Mid(InStrng, Cntr, 1))
            End Select
        Next Cntr
        Str2Int = NewStr
    End Function

    ' التشويش دالة
    Public Function Obfuscate(ByVal origText As String) As String
        Dim textBytes As Byte() = System.Text.Encoding.Unicode.GetBytes(origText)
        For counter As Integer = 0 To textBytes.Length - 1
            If (textBytes(counter) > 31) And (textBytes(counter) < 127) Then
                textBytes(counter) += CByte(counter Mod 31 + 1)
                If (textBytes(counter) > 126) Then textBytes(counter) -= CByte(95)
            End If
        Next counter
        Return System.Text.Encoding.Unicode.GetChars(textBytes)
    End Function
End Module
```

End Module

الفورم:

كود:

```
Imports System.Management

Public Class Form1
    Dim Idp As String
    Dim Idp1 As String
    Dim idp2 As String

    Private Sub FrmActiv_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'للمعالج التسلسلي الرقم
        Dim Searcher As ManagementObjectSearcher
        Searcher = New ManagementObjectSearcher("Select ProcessorId From Win32_Processor")
        For Each Device As ManagementObject In Searcher.Get
            For Each Prop As PropertyData In Device.Properties
                '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                'الرقم استخراج
                Idp = (Prop.Value.ToString)
            Next
        Next
        '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        'الرقم تشويش
        Idp = Obfuscate(Idp)
        'اسكي الى تحويله
        Idp = Str2Int(Idp)
        'التسجيل رقم
        TextBoxReg.Text = Idp
        '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        'الرقم تشويش
        Idp1 = Obfuscate(Idp)
        'اسكي الى تحويله
        'التفعيل رقم
        Idp1 = Str2Int(Idp1)
        'رقم 14 اقتصاص
        Idp1 = (Idp1.Substring(0, 14))
    End Sub

    Private Sub ButtonOk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonOk.Click
        'جمالي متغير
        idp2 = Idp1
        If TextBoxUser.Text = idp2 Then
            'الاعدادت حفظ
            My.Settings.nameuser = TextBoxUser.Text
            My.Settings.Save()
        End If
    End Sub
End Class
```

والتصميم التالي:

NOUR-ProcessorId

قم بنسخ رقم التسجيل و إرساله للشركة
وسيتم إرسال رقم التفعيل للبرنامج
لتحصل على كامل مميزات البرنامج
رقم الهاتف: نور نبهان

رقم تسجيل البرنامج :

49546158796883856567697173819580

رقم تفعيل البرنامج :

من فضلك أدخل رقم تفعيل النسخة

قم بإدخال رقم التفعيل الذي حصلت عليه
في الصندوق السابق ثم اضغط على زر تفعيل النسخة

تفعيل النسخة

انتهى أمر التفعيل إذا خطأ لن يقوم بأي إجراء منعا لإجراء التجارب من قبل المعتدين و إذا صح يحفظ الرقم فقط
وتتم المقارنة لاحقاً.

الآن طريقة المقارنة بعيدا عن فورم التفعيل كالاتي

اذهب لأي فورم و ألصق كود الرقم

كود:

```
Imports System.Management

Public Class Form1
    Dim Idp As String
    Dim Idp1 As String
    Dim idp2 As String

    Private Sub FrmActiv_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'للمعالج التسلسلي الرقم'
        Dim Searcher As ManagementObjectSearcher
        Searcher = New ManagementObjectSearcher("Select ProcessorId From Win32_Processor")
        For Each Device As ManagementObject In Searcher.Get
            For Each Prop As PropertyData In Device.Properties
                '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%'
                'الرقم استخراج'
                Idp = (Prop.Value.ToString)
            Next
        Next
        '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%'
        'الرقم تشويش'
        Idp = Obfuscate(Idp)
        'اسكي الى تحويله'
```



```

Idp = Str2Int(Idp)
'التسجيل رقم'
TextBoxReg.Text = Idp
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%'
'الرقم تشويش'
Idp1 = Obfuscate(Idp)
'اسكي الى تحويله'
'التفعيل رقم'
Idp1 = Str2Int(Idp1)
'رقم 14 اقتصاص'
Idp1 = (Idp1.Substring(0, 14))
End Sub

```

ففي أي فورم أو فورما محددة على مزاجك سيتم استنتاج الرقم كما تعلمنا سابقا في فورم التفعيل وحفظ الرقم في متغير

كما تلاحظ من الكود أخر قيمة كانت رقم التفعيل المكون من ١٤ رقم والمتغير هو **Idp1**

المقارنة كالتالي

كود:

```

If My.Settings.nameuser = Idp1 Then
'do thing
Else
'end
End If

```

والآن ضع المقارنة أينما تريد في زر الطباعة في زر البحث في تحميل فورم المبيعات في الخ
واختر الإجراء المناسب سواء بإغلاق البرنامج أو بتعطيل بعض الأزرار أو بإخفاء نوافذ أو الخ.....
وكل ما كثرت المقارنات صار البرنامج أبطأ عفواً أقصد صار الأمر مبهم أكثر (ملح وفلفل)
عداك عند وضع مقارنات زمنية. (شهرية ... ساعية ... بعد ١٠ استعمالات ... الخ) بمحض إرادتك .
مداخلة :

حاول أيضا تفحص قيمة المتغير الأخير الذي سنخزن فيه رقم التفعيل بحيث إذا كانت فارغة الخروج من الإجراء وعدم التفعيل أو استخدم Try.

لأنه ربما بعد عشرين سنة يخرج نظام ويندوز لا يدعم المكتبة فيكون استخراج رقم المعالج خالي وتتم المقارنة مع قيمة خالية فتتساوى القيمة ويتم التفعيل أو يولد خطأ .

إلى هنا ننتهي من قسم التسجيل ونذهب الآن لقسم التفعيل:

إن برنامج التفعيل يجب أن يتواجد لديك فقط أو للأشخاص الاعتباريين الذي يحق لهم توليد أرقام التفعيل بحسب رغبتك.

لأن برنامج التفعيل هو الكيجن لبرنامجك إن صح التعبير

طريقة توليد أرقام التفعيل :

هنا الأمر بسيط جدا لأنه جزء صغير مما سبق

وهو جزء من مشروعنا فهو رقم التسجيل مشوش ومحول لاسكي ومقتص ١٤ رقم
ننشأ مشروع جديد ونضيف فيه موديول نضع فيه الدوال السابقة كما في التسجيل كالموديول السابق تماماً
وننشأ فورم جديد عليه زر أمر وصندوقي نص **TextBoxUser TextBoxReg ButtonOk**

التصميم :

كود زر التفعيل :

كود:

```
Public Class Form1
    Dim XXX As String
    Private Sub ButtonOk_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonOk.Click
        'التسجيل رقم
        XXX = TextBoxReg.Text
        '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        'الرقم تشويش
        XXX = Obfuscate(XXX)
        'اسكي الى تحويله
        'التفعيل رقم
        XXX = Str2Int(XXX)
        'رقم اقتصاص 14
        XXX = (XXX.Substring(0, 14))
        'التفعيل رقم استنتاج
        TextBoxUser.Text = XXX
    End Sub
End Class
```

طبعا لا حاجة للتشرح فالأمر واضح مما سبق ومن الكود (فقط بعض التعديلات على رقم التسجيل)

الآن ما عليك إلا الضغط على زر رقم التفعيل بعد إدخال رقم التسجيل وسيظهر في الصندوق الثاني رقم التفعيل لترسله للعميل
ويمكن من تفعيل البرنامج بالشكل الصحيح .

وبذلك نكون قد ناقشنا الموضوع بطريقة مطولة بعض الشيء لذلك معذرة مني إليكم .

[VB_Activate.zip](#)

وفي الختام:

- الغرض من برنامجنا هو غرض تعليمي.
- إذا كان برنامجك يستحق الحماية فقم بحمايته.
- لا تعرض برنامجك مباشرة و إنما قم بنشر نسخة محدودة المهام أصلاً أي البرمجة غير كاملة بمثال عدم وجود البحث أبداً أو الطباعة أو وعند الطلب أرسل للعميل فقط نسخة كاملة تحتاج للتفعيل بإحدى الطرق ويفضل أن تتوسع في البحث عن حمايات أقوى مما قدمناه.

C#

حاولت طرح الموضوع للإخوة المستخدمين للسي شارب

وجدت المثال التالي:

حمل المثال الأصلي بالضغط هنا

وبعد التعديل يصبح المثال كالتالي:

في الموديول WMI_Processor_Information.cs

الكود:

كود:

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Management;

Namespace WMI_ProcessorInformation
{
    Public Class WMI_Processor_Information
    {

        //المعالج رقم, Cpu serial
        public static string GetCpuId()
        {
            try
            {
                ManagementObjectSearcher searcher = new
ManagementObjectSearcher("root\\CIMV2",
"SELECT * FROM Win32_Processor");

                foreach (ManagementObject queryObj in searcher.Get())
                {
                    return queryObj["ProcessorId"].ToString();
                }
            }
            catch (ManagementException e)
            {
                return null;
            }
            return null;
        }

        //التشويش
        public static string Obfuscate(string origText)
        {
            byte[] bytes = Encoding.Unicode.GetBytes(origText);
            int num2 = bytes.Length - 1;
            for (int i = 0; i <= num2; i++)
```

```

        {
            if ((bytes[i] > 0x1f) & (bytes[i] < 0x7f))
            {
                int index = i;
                bytes[index] = (byte)(bytes[index] + ((byte)((i % 0x1f) +
1)))));
                if (bytes[i] > 0x7e)
                {
                    index = i;
                    bytes[index] = (byte)(bytes[index] - 0x5f);
                }
            }
        }
        return new string(Encoding.Unicode.GetChars(bytes));
    }
    // اسكي
    // ('
}
}

```

في الفورم

أضف الأدوات : TextBox3 TextBox2 TextBox1 Button1

: Button1 وضع الكود التالي للزر

كود:

```

{
    // المعالج رقم
    textBox1.Text =
(WMI_ProcessorInformation.WMI_Processor_Information.GetCpuId());
    // الرقم تشويش 1
    textBox2.Text =
(WMI_Processor_Information.Obfuscate(textBox1.Text));
    // الرقم تشويش 2
    textBox3.Text =
(WMI_Processor_Information.Obfuscate(textBox2.Text));
}

```

[حمل المثال بعد التعديل بالضغط هنا](#)

يمكنكم إكماله كما تريدون انطلاقاً مما سبق في شرحنا السابق

سأعطي الآن مثال يوضح السماح بالدخول للبرنامج أو عدم السماح إذا لم يتم التسجيل وسأعتمد طريقة حفظ القيمة في **الريجستري من باب التوسع :**

طبعاً لن نعيد الشرح السابق كاملاً فالأمر ما هو إلا تعديل بسيط :

كود الموديل كما هو

كود الفورم سوف نغيره كما يلي:

كود:

```

Imports System.Management

Public Class Form1
    Dim Idp As String
    Dim Idp1 As String

    Private Sub FrmActiv_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'للمعالج التسلسلي الرقم
        Dim Searcher As ManagementObjectSearcher
        Searcher = New ManagementObjectSearcher("Select ProcessorId From
Win32_Processor")
        For Each Device As ManagementObject In Searcher.Get
            For Each Prop As PropertyData In Device.Properties
                '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                'الرقم استخراج
                Idp = (Prop.Value.ToString)
            Next
        Next
        '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        'الرقم تشويش
        Idp = Obfuscate(Idp)
        'اسكي الى تحويله
        Idp = Str2Int(Idp)
        'التسجيل رقم
        TextBoxReg.Text = Idp
        '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        'الرقم تشويش
        Idp1 = Obfuscate(Idp)
        'اسكي الى تحويله
        'التفعيل رقم
        Idp1 = Str2Int(Idp1)
        'رقم اقتصاص 14
        Idp1 = (Idp1.Substring(0, 14))
        '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        'الرقم من التحقق عملية
        ' بوكس التيكست في ووضعها الريج من القيمة قراءة
        TextBoxUser.Text =
My.Computer.Registry.GetValue("HKEY_CURRENT_USER\Software\Nour", "num", "0")
        'مباشرة البرنامج يظهر سوف صح اذا الان
        If TextBoxUser.Text = Idp1 Then
            MAIN.Show()
            Me.Close()
            'الظاهرة هي التسجيل نافذة ستبقى والا
        End If
    End Sub

    Private Sub ButtonOk_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonOk.Click
        If TextBoxUser.Text = Idp1 Then
            'الريجستري في بحفظه سيقوم صح التفعيل رقم كان اذا
            My.Computer.Registry.SetValue("HKEY_CURRENT_USER\Software\Nour",
"num", TextBoxUser.Text)
            MsgBox("البرنامج تشغيل أعد رجاء ... صحيح الرقم")
        End
    End If
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        End
    End Sub
End Class

```

و التغيير الحاصل هو :

كود زر التفعيل إلى :

كود:

```
Private Sub ButtonOk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonOk.Click
    If TextBoxUser.Text = Idp1 Then
        'الريجستري في بحفظه سيقوم صح التفعيل رقم كان اذا'
        My.Computer.Registry.SetValue("HKEY_CURRENT_USER\Software\Nour", "num", TextBoxUser.Text)
        MsgBox("البرنامج تشغيل أعد رجاء ... صحيح الرقم")
    End
End If
End Sub
```

حيث سيتم تفحص رقم التفعيل إذا كان صحيحا فسيتم حفظه داخل الريجستري كما هو في مسار تحده كما تريد .

الآن في حدث تحميل الفورم أضفنا الكود التالي أسفل الكل

كود:

```
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%'
'الرقم من التحقق عملية'
' بوكس التيكست في ووضعها الريح من القيمة قراءة '
    TextBoxUser.Text =
My.Computer.Registry.GetValue("HKEY_CURRENT_USER\Software\Nour", "num", "0")
'مباشرة البرنامج يظهر سوف صح اذا الان'
    If TextBoxUser.Text = Idp1 Then
        MAIN.Show()
        Me.Close()
'الظاهرة هي التسجيل نافذة ستبقى والا'
    End If
```

حيث سيتم قراءة القيمة المخزنة في الريجستري إذا كانت صحيحة فسوف تظهر لنا شاشة البرنامج الرئيسية وتغلق نافذة التفعيل وغير ذلك ستبقى نافذة التسجيل هي الظاهرة .

طبعا أضفنا فورم رئيسي يسمى MAIN كتمثيل للنافذة الرئيسية لبرنامجنا.

[حمل المثال المرفق 01registry.rar](#)

كشف رقم المتصل عن طريق فيجوال بيسك دوت نت

الكاتب : [husamsamra](#)

[رابط المقال و النقاش حوله](#)

السلام عليكم ورحمة الله وبركاته
إخواني الكرام ، بعد معاناة كبيرة في البحث في هذا الموضوع ، تم بحمد الله حله والوصول إلى معلومات وافية عنه ، لذلك أردت أن يستفيد إخواني في هذا المنتدى مما حصلت في هذا الجانب.
ولكن قبل كل شيء ، أقول أن كل شيء بثمن ، لذلك من يدفع يأخذ. ولكن أيضا من لا يدفع سيأخذ.
والثمن هو دعوة صالحة في ظهر الغيب.

الموضوع: إن كنت مبرمجا في فيجوال بيسك دوت نت وتريد أن تستقبل بيانات من كرت المودم فافعل التالي (الكود مجرب وأكثر من رائع على الإصدار **VISUAL STUDIO 2010** حيث صار هناك حدث معتمد للسيريرال بورت هو **Data Recived** فأقول على بركة الله:

- 1- تأكد من أن لديك كرت مودم معرف بشكل صحيح.
 - 2- اعرف اسم المنفذ لكرت المودم (من إدارة الأجهزة)
 - 3- قم بإنشاء مشروع جديد.
- أضف للنافذة أداة **Serial Port** وليكن اسمها **SerialPort1** واجعل خاصية **PortName="COM3"** حيث **COM3** هو منفذ كرت المودم ، وربما يكون المنفذ لديك غير ذلك.
أكتب الكود التالي عند الحدث: **Form_Load**

كود:

```
SerialPort1.open
```

سيتم فتح المنفذ عند بدء تشغيل البرنامج ، إذا كنت تستعمل برنامجا لكشف الرقم وكان البرنامج قيد التشغيل فسيفشل بدء البرنامج لأن المنفذ مستخدم من قبل برنامج آخر.

عد إلى وضع البرمجة وأضف زرا جديدا ، واكتب في الحدث **Click** للزر الكود التالي:

كود:

```
SerialPort1.Write("AT#CID=1" & vbCrLf)
```

يقوم هذا الكود بإرسال أمر إلى السيريرال بورت يعلمه بأننا نريد استقبال بيانات المتصل عن طريق البورت ، وإذا حاولت قراءة البيانات الناتجة عن ذلك فيجب أن تحصل على "OK" :
إذا لم تحصل على **OK** فربما يكون المودم لديك لا يكشف الرقم أو أنك أخطأت في كتابة التعليمة.

بهذا نكون قد جهزنا البرنامج لاستقبال الاتصال.

في الحدث **DataReceived** أكتب الكود التالي:

كود:

```
Dim MyStr As String  
MyStr = SerialPort1.ReadExisting()  
msgbox (MyStr)
```

عند تشغيل البرنامج سوف يقوم بفتح المنفذ والتهيؤ لاستقبال المكالمات ، فإذا رن الهاتف فستظهر رسالة فيها **"RING"** ثم بعدها رسالة فيها بيانات المتصل.

أنا كنت أود أن أرفق مثالا عن ذلك ولكن لغة البرمجة لدي **VB2010** ولا أظنها ستعمل على الإصدارات السابقة ، ولكني أتمنى أن أكون قد وفقت في الشرح ، وسنبقى على تواصل إن شاء الله.

طبعا بالنسبة للإجراءات الأخرى من العمل على البيانات على المودم فسوف تحتاج إلى المكتبة **TAPI** بأحد إصداراتها وهذه مكتبة معقدة نوعا ما ، و تعقبا على المقال فقد وجدت أن بعض المقاسم لا تتجاوب مع التعليمات السابقة

AT#CID=1

التي ترسلها إلى المودم ليكون جاهزا لاستقبال رقم المتصل من المقسم ، فطبيعة التردد ووقت إرسال البيانات يختلف من مقسم إلى آخر ، علما أن التعليمات السابقة هي الشائعة ، وهي تعمل مع معظم المقاسم ، وقد وجدت في مقال باللغة الإنكليزية النص التالي ومفاده أن معظم المقاسم تعمل على التعليمات الأولى ، وهناك تعليمات أخرى تستطيع تجربتها ، فقط استبدل التعليمات بها عند إرسالها إلى منفذ المودم أول مرة ، لتعرف أي منها يناسب المقسم لديك.

Modem initialization Strings

For most modems, the string **AT#CID=1** will enable Caller ID. On some voice based modems the string is **AT#CLS=8#CID=1**. Look in the user's manual for your modem to see if any setup string is referenced or mentioned. So the most common setup strings are:

AT#CID=1
AT#CLS=8#CID=1
AT#CID=2
AT%CCID=1
AT%CCID=2
AT+VCID=1
AT+VCID=2
AT#CC1
AT*ID1

أنا شخصيا لم يعمل معي على المودم سوى الأول والثالث والسادس والسابع و بالنسبة لبيانات كشف الرقم التي تظهر لك فهي حصرا على الشك التالي:

DATE=0618
TIME=2353
NMBR=0933300097
DDN=6212013
MESG = 110101
MESG = 16010A

هذه الأسطر الستة تتضمن على التوالي:

التاريخ (شهر ويوم)
الوقت (ساعة ودقيقة)
رقم المتصل
رقمك أنت

والسطين الأخيرين هما عبارة عن رسالة لست متأكدا من ماهيتها حتى الآن

بالنسبة لاستخراج الرقم من النص الناتج فإنا أستخدم هذه العبارة:

على اعتبار أن البيانات قد وضعت في متحول **MyStr** أكتب الكود التالي :

رمز:

كود:

```
Dim MyStart As Integer = InStr(LCase(MyStr), "nmb")
Dim MyEnd As Integer = InStr(LCase(MyStr), "ddn")
Dim CallerIDNumber As String
CallerIDNumber=MyStr.Substring(MyStart + 4, MyEnd - MyStart - 5).Trim
```

وبإمكانك استخدام الطريقة التي تناسبك لاستخلاص الرقم من النص

لاحظ أن الأسطر مفصولة بتعليمة **VBCrLf**

وكما ذكرت في البداية ، كشف الرقم يتطلب أن يكون مفعلاً من المقسم ، فإذا لم يكن الرقم يظهر أصلاً عندك على الهاتف ، فلن تحصل عليه على الحسب ، بل عليك الاشتراك في المقسم إذا كان هذا الأمر متاحاً ، لأن كشف الرقم لدينا في سوريا ميزة تشترك بها على خطك .

هذا المثال إخواني على **vb 2010** أرجو أن يفيدكم

الملفات المرفقة : [VB 2010 Modem Sample.rar](#)

مقال : أفكار في الدوت نت OutAttribute Class

الكاتب : [silverlight](#)
[رابط المقال و النقاش حوله](#)

مقدمة:

يوجد في الدوت نت كلاس اسمه **OutAttribute** وهو موجود تحديدا في **System.Runtime.InteropServices** NameSpace وفي موضوعنا هذا سنوضح كيفية استخدام هذا الكلاس ولمزيد من المعلومات عن هذا الكلاس يمكنكم زيارة هذا [الرابط](#)

المثال الأول:

لنفرض أننا لدينا **String Array** تم تعريفها كالتالي وكما ستلاحظون فهي تحتوي على أرقام فقط

كود:

```
Dim arr As String() = {10, 100, 50, 30, 40, 200, 5, 12}
```

الآن نحن نريد أن نحصل علي أعلى قيمة وأقل قيمة للأرقام الموجودة داخل هذه **Array** غالبا هنا ما نقوم بكتابة دالتين للحصول علي أقل قيمة و أعلى قيمة وهاتان الدالتان سيكون الكود الخاص بهما كالآتي :

كود:

```
Private Function GetMaximumValue(ByVal StringArr As String()) As Integer
    Dim arr As New HashSet(Of Integer)
    For i As Integer = 0 To StringArr.Count - 1
        arr.Add(StringArr(i))
    Next
    Return arr.Max
End Function

Private Function GetMinimumValue(ByVal StringArr As String()) As Integer
    Dim arr As New HashSet(Of Integer)
    For i As Integer = 0 To StringArr.Count - 1
        arr.Add(StringArr(i))
    Next
    Return arr.Min
End Function
```

والكود التالي يوضح كيفية استخدام هاتان الدالتان

كود:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim arr As String() = {10, 100, 50, 30, 40, 200, 5, 12}
    Dim min As Integer = GetMinimumValue(arr)
    Dim max As Integer = GetMaximumValue(arr)
    Me.Text = min + max
End Sub
```

ملحوظة هامة علي المثال الأول : الكود الموجود بالمثال أعلاه يحتاج الفيچوال أستوديو ٢٠٠٨ على الأقل والسبب في ذلك هو أنني استخدمت الكلاس **HashSet** وهو موجود فقط في الدوت نت ٣.٥ فما فوق

المثال الثاني:

الآن كيف نستخدم الكلاس **OutAttribute** للحصول على نفس النتائج الموجودة بالمثال الأول ولكن يجب أن نتفهم أننا وعند استخدام الكلاس **OutAttribute** فإن البيانات المرتبطة به يتم تمريرها باستخدام **ByRef** فقط حيث أننا لو مررنا البيانات باستخدام **ByVal** فإننا لن نحصل على شيء.

ولمزيد من المعلومات عن الفرق بين **ByVal** و **ByRef** يوجد بالمنتدى موضوعا كتبه الأخ الفاضل **مروان** ويمكنك الاطلاع عليه في هذا الرابط [الفرق بين ByVal و ByRe](#)

الآن لنعيد صياغة الدالتان الموجودتان بالمثال الأول وبدلا من كتابتهما كدالتان سنكتب فقط روتين أو **Sub** والكود التالي يوضح ذلك :

كود:

```
Private Sub GetMaximumMinimumValue(ByVal StringArr As String(), <Out()> ByRef
max As Integer, <Out()> ByRef min As Integer)
    Dim arr As New HashSet(Of Integer)
    For i As Integer = 0 To StringArr.Count - 1
        arr.Add(StringArr(i))
    Next
    max = arr.Max
    min = arr.Min
End Sub
```

نلاحظ في الكود أعلاه كيفية استخدام الكلاس **OutAttribute Class** وستلاحظ عزيزي القارئ أننا مررنا القيمتان **min** و **max** باستخدام **ByRef** والهدف من استخدامنا الكلمة **<Out()>** قبل كل من المتغير **min** و المتغير **max** في الروتين أعلاه هو أننا نحتاج إلى أن تكون النتائج المراد الحصول عليها هما هذان المتغيران وكل ما فعلناه هو ربطهم بالمتغير **StringArr** فقط لا غير

ولكي نستخدم الروتين أو الكود أعلاه علينا أن نقوم بتعريف المدخلات كالتالي:

تعريف متغير يعبر عن **String Array** كالتالي

كود:

```
Dim arr As String() = {10, 100, 50, 30, 40, 200, 5, 12}
```

تعريف متغيران عبارة عن **Integer** ولا نعطي لهم أي قيمة

كود:

```
Dim min As Integer
Dim max As Integer
```

الآن نمرر هذان المتغيران إلى الروتين و سنلاحظ هنا أننا مررنا المتغيران بالرغم من أن قيمتهم غير معرفة

كود:

```
GetMaximumMinimumValue(arr, max, min)
```

السطر التالي من الكود هو ما نحتاجه لحصول على قيم فعلية للمتغيرات **min** و **max** حيث هنا نعيد مساواة القيم ثانية مع ما يناظرها

كود:

```
max = max
min = min
```

الآن لنستخدم القيمان التي حصلنا عليهم كما نريد والكود التالي يوضح الشكل النهائي للكود :

كود:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim arr As String() = {10, 100, 50, 30, 40, 200, 5, 12}
    Dim min As Integer
    Dim max As Integer
    GetMaximumMinimumValue(arr, max, min)
    max = max
    min = min
    Me.Text = min + max
    max = Nothing
    min = Nothing
End Sub
```

ملحوظة هامة علي المثال الثاني : الكود الموجود بالمثال أعلاه يحتاج الفيچوال أستوديو ٢٠٠٨ علي الأقل والسبب في ذلك هو أنني استخدمت الكلاس **HashSet** وهو موجود فقط في الدوت نت (فيما فوق

طبعا ولأنني أعشق الجرافكس فمن الصعب أن أترك المقال بدون أن أضع أمثلة توضح كيفية استخدام الكلاس **OutAttribute** في **+GDI**

المثال الثالث:

لنفرض أننا نريد أن نحصل علي نقطة مركز المستطيل فغالبا هنا علينا أن نمرر متغير يشير إلى هذا المستطيل وتكون النتيجة التي نحصل عليها عبارة عن نقطة والكود التالي يوضح شكل الروتين المطلوب كتابته

كود:

```
Private Sub GetRectCenterPoint(ByVal r As Rectangle, <Out()> ByRef pt As Point)
    pt.X = ((r.Left + r.Right) / 2)
    pt.Y = ((r.Top + r.Bottom) / 2)
End Sub
```

ثم في الحدث **Paint** يمكن أن نستخدم الروتين بالشكل التالي :

كود:

```
Protected Overrides Sub OnPaint(ByVal e As System.Windows.Forms.PaintEventArgs)
    MyBase.OnPaint(e)

    ' المستطيل تعريف
    Dim boundRect As New Rectangle(10, 10, 100, 100)
    ' قيمة اي اعطاؤها بدون ما نقطة تعريف
    Dim pt As Point
    ' أعلاه الروتين الي النقطة تمرير
    GetRectCenterPoint(boundRect, pt)
    ' النقطة لهذه الفعلية القيمة علي الحصول
    pt = pt
    ' المستطيل مثل ما شيئا نرسم ثم
    e.Graphics.DrawRectangle(Pens.Black, boundRect)
    ' نقطة الي المستطيل منتصف من خطا لنرسم عليها حصلنا التي النقطة نسخدم ثم
    الفورم علي المستطيل بداية
```

```
e.Graphics.DrawLine(Pens.Red, pt, boundRect.Location)
```

```
' رغبتنا حسب شئ لا او صفر الي النقطة قيمة نعيد ان يمكننا هنا  
pt = Nothing
```

```
End Sub
```

المثال الرابع:

ويمكن إعادة كتابة الروتين أعلاه والموجود بالمثال الثالث بالشكل التالي أيضا :

كود:

```
Private Sub GetRectCenterPoint(ByVal r As Rectangle, <Out()> ByRef x As Integer,  
<Out()> ByRef y As Integer)  
    x = ((r.Left + r.Right) / 2)  
    y = ((r.Top + r.Bottom) / 2)  
End Sub
```

ثم في الحدث **Paint** أيضا يمكن أن نستخدم الروتين بالشكل التالي :

كود:

```
Protected Overrides Sub OnPaint(ByVal e As System.Windows.Forms.PaintEventArgs)  
    MyBase.OnPaint(e)  
  
    ' المستطيل تعريف  
    Dim boundRect As New Rectangle(10, 10, 100, 100)  
    ' قيمة اي اعطاؤها بدون ما متغيران تعريف  
  
    Dim Xcenter As Integer  
    Dim Ycenter As Integer  
  
    ' الروتين الي القيمتان نمرر  
    GetRectCenterPoint(boundRect, Xcenter, Ycenter)  
    ' الفعلية القيم علي الحصول  
    Xcenter = Xcenter  
    Ycenter = Ycenter  
    ' نريد ما لنرسم البيانات نستخدم ثم  
    ' المستطيل مثل ما شيئا نرسم  
    e.Graphics.DrawRectangle(Pens.Black, boundRect)  
  
    ' نقطة الي المستطيل منتصف من خطا لنرسم عليهما حصلنا التي القيمتان نستخدم ثم  
    ' الفورم علي المستطيل بداية  
    e.Graphics.DrawLine(Pens.Red, New Point(Xcenter, Ycenter),  
boundRect.Location)  
  
End Sub
```

بالنهاية ومن المؤكد أنه يمكن أن نكتب الكثير من الأفكار المماثلة للأمثلة أعلاه بحيث نستفيد من الكلاس **OutAttribute** بأكثر من أسلوب وتحديدًا في **Data Marshalling** وخاصة مع **Managed Code** والآن الأمر متروك لك عزيزي القارئ لتكتشف كيفية عمل ذلك بنفسك

مقال: أفكار في الجرافكس (ثلاثة أجزاء)

الكاتب : [silverlight](#)

رابط المقال و النقاش حوله [٣](#) [٢](#) [١](#)

الجزء الأول

مقدمة:

غالبا ما تكون الأفكار الكبيرة ما هي إلا مجموعة من الأفكار الصغيرة التي تتجمع معا لتشكل هذه الأفكار الكبيرة وفي هذه الأجزاء عن الجرافكس سنحاول أن نغطي مجموعة من الأفكار الصغيرة التي من الممكن أن نستفيد منها لاحقا في بناء فكرة كبيرة

عالم GDI+

المهتمين بالرسم أو بالجرافكس بشكل عام يعلمون جيدا معنى كلمة **GDI+** وأيضا يعلمون جيدا كيف يستخدمون الأوامر الخاصة بمكتبة **GDI+** لرسم الأشكال الهندسية المختلفة أو رسم الصور أو أي شيء آخر علي الفورم أو علي الكونترول لذلك نحن هنا لن نقاش كيف نرسم بل سوف نقاش كيف نستفيد ونطور من عمليات الرسم المختلفة وبالتالي نستفيد منها في بناء ما نريده

الفكرة الأولى:

الفكرة الأولى التي سوف نتحدث عنها اليوم هي رسم أي شكل هندسي ومن ثم نقوم بتحريك هذا الشكل الهندسي باستخدام الفارة في أي اتجاه نريده والمجموعة التالية من الأمثلة سوف توضح كيفية عمل ذلك

المثال الأول:

لكي نرسم أي شكل هندسي علي الفورم ولناخذ المستطيل كمثال غالبا ما نحتاج إلى تعريف نقطة بداية عملية الرسم وأيضا أبعاد هذا المستطيل أي أننا نحتاج إلى تعريف **Location** وأيضا **Size** ومن ثم نقوم برسم هذه المستطيل علي سطح الفورم أو الكونترول باستخدام أوامر **GDI+** وغالبا ما يتم ذلك في الحدث **Paint** وهذا الحدث موجود في كل من الفورم و الكونترول والكود التالي يوضح الفكرة :

كود:

```
Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Me.Paint
    ' الرسم بداية نقطة تعريف
    Dim startPoint As Point = New Point(10, 10)
    ' المستطيل أبعاد تعريف
    Dim RectangleSize As Size = New Size(100, 100)
    ' المستطيل تعريف
    Dim Rect As Rectangle = New Rectangle(startPoint, RectangleSize)
    ' المستطيل تلوين
    e.Graphics.FillRectangle(Brushes.Red, Rect)
    ' المستطيل رسم
    e.Graphics.DrawRectangle(Pens.Blue, Rect)
End Sub
```

أيضا من الممكن الرسم عن طريق عمل **Overrides** للحدث **Paint** والكود التالي يوضح الفكرة

كود:

```
Protected Overrides Sub OnPaint(ByVal e As System.Windows.Forms.PaintEventArgs)
    MyBase.OnPaint(e)
    ' الرسم بداية نقطة تعريف
    Dim startPoint As Point = New Point(10, 10)
    ' المستطيل أبعاد تعريف
    Dim RectangleSize As Size = New Size(100, 100)
    ' المستطيل تعريف
    Dim Rect As Rectangle = New Rectangle(startPoint, RectangleSize)
    ' المستطيل تلوين
    e.Graphics.FillRectangle(Brushes.Red, Rect)
    ' المستطيل رسم
    e.Graphics.DrawRectangle(Pens.Blue, Rect)
End Sub
```

المثال الثاني:

في المثال الأول قمنا بتثبيت القيم الخاصة بالمستطيل مثل بداية نقطة الرسم و أيضا أبعاد المستطيل وبالتالي سيظل هذا المستطيل مرسوما بشكل دائم علي الفورم ولكي نستطيع أن نتحكم في الشكل المرسوم بشكل أفضل نحتاج إلى أن يتم تعريف جميع المتغيرات الخاصة به بشكل عام والكود التالي يوضح ذلك :

كود:

```
Public Class Form1
    Private startPoint As Point = New Point(10, 10)
    Private RectangleSize As Size = New Size(100, 100)
    Private Rect As Rectangle

    Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Me.Paint
        ' المستطيل تعريف
        Rect = New Rectangle(startPoint, RectangleSize)
        ' المستطيل تلوين
        e.Graphics.FillRectangle(Brushes.Red, Rect)
        ' المستطيل رسم
        e.Graphics.DrawRectangle(Pens.Blue, Rect)
    End Sub
End Class
```

أيضا من الممكن الرسم عن طريق عمل **Overrides** للحدث **Paint** والكود التالي يوضح الفكرة :

كود:

```
Public Class Form1

    Private startPoint As Point = New Point(10, 10)
    Private RectangleSize As Size = New Size(100, 100)
    Private Rect As Rectangle

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)
        ' المستطيل تعريف
        Rect = New Rectangle(startPoint, RectangleSize)
        ' المستطيل تلوين
        e.Graphics.FillRectangle(Brushes.Red, Rect)
        ' المستطيل رسم
        e.Graphics.DrawRectangle(Pens.Blue, Rect)
    End Sub
```

المثال الثالث:

في هذا المثال سنحاول تحريك المستطيل عن طريق استخدام الماوس ولكي نفعل ذلك نحتاج إلى تعريف بعض المتغيرات الإضافية حتى نستطيع أن نقوم بعمل ارتباط بين المستطيل ومكان الماوس على الفورم أو على الكونترول

والكود التالي يوضح المتغيرات التي نحتاجها مع توضيح للهدف من كل من متغير قمنا بتعريفه بالإضافة إلى كيفية استخدام كل متغير في الأحداث الخاصة بالفورم مثل الحدث **MouseDown** و **Paint** و **MouseMove** وأيضا **MouseUp**

كود:

```
Public Class Form1
    ' الرسم بداية نقطة
    Private startPoint As Point = New Point(10, 10)
    ' المستطيل أبعاد
    Private RectangleSize As Size = New Size(100, 100)
    ' المستطيل تعريف
    Private Rect As Rectangle
    ' المستطيل تحريك نقطة مكان لتحديد تستخدم سوف نقطة تعريف
    Private emptyPoint As Point = Point.Empty
    ' حركة أثناء الأفقي الاتجاه في المستطيل رسم نقطة الي يشير رقمي متغير تعريف
    الماوس
    Private x As Integer = startPoint.X
    ' حركة أثناء الرأسى الاتجاه في المستطيل رسم نقطة الي يشير رقمي متغير تعريف
    الماوس
    Private y As Integer = startPoint.Y

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)
        Rect = New Rectangle(x, y, RectangleSize.Width, RectangleSize.Height)
        Using lgb As New Drawing2D.LinearGradientBrush(Rect, Color.FromArgb(255,
Color.DarkBlue), Color.FromArgb(255, Color.Cyan), 90, True)
            e.Graphics.FillRectangle(lgb, Rect)
        End Using
    End Sub

    Protected Overrides Sub OnMouseDown(ByVal e As
System.Windows.Forms.MouseEventArgs)
        MyBase.OnMouseDown(e)
        If e.Button = Windows.Forms.MouseButtons.Right Then Return
        ' طريق عن المستطيل تحريك نقطة أو مكان الي يشير الذي للمتغير قيمة تمرير
        الماوس مع ربطها
        emptyPoint.X = e.Location.X
        emptyPoint.Y = e.Location.Y
    End Sub

    Protected Overrides Sub OnMouseMove(ByVal e As
System.Windows.Forms.MouseEventArgs)
        MyBase.OnMouseMove(e)
        ' بعض يمتلك بالفعل الماوس تحريك نقطة الي يشير الذي المتغير أن نتأكد هنا
        القيم
        If emptyPoint = Point.Empty Then Return
        ' الأفقي الاتجاه في المستطيل رسم بداية لنقطة جديدة قيم نمرر هما
        x = e.X - emptyPoint.X + startPoint.X
        ' الرأسى الاتجاه في المستطيل رسم بداية لنقطة جديدة قيم نمرر هنا
        y = e.Y - emptyPoint.Y + startPoint.Y
        ' المستطيل رسم يعيد أن الفورم من نطلب هنا
    End Sub
End Class
```



```

        Me.Invalidate()
    End Sub

    Protected Overrides Sub OnMouseUp(ByVal e As
System.Windows.Forms.MouseEventArgs)
        MyBase.OnMouseUp(e)
        ' الماوس بيسار الخاص الباتون باستخدام قام المستخدم بأن نتأكد هنا
        If e.Button = Windows.Forms.MouseButtons.Right Then Return
        ' المستطيل رسم باية نقطة الب جديدة قيمة نمرر هنا
        startPoint.X += e.X - emptyPoint.X
        startPoint.Y += e.Y - emptyPoint.Y
        ' الأصلية قيمته الي المستطيل تحريك مكان الي يشير الذي المتغير نعيد هنا
        ' شئ لا وهي
        emptyPoint = Point.Empty
    End Sub
End Class

```

المثال الرابع:

تحريك المستطيل في الاتجاه الأفقي فقط والكود التالي يوضح ذلك

كود:

```

Public Class Form1

    ' الرسم بداية نقطة
    Private startPoint As Point = New Point(10, 10)
    ' المستطيل أبعاد
    Private RectangleSize As Size = New Size(100, 100)
    ' المستطيل تعريف
    Private Rect As Rectangle
    ' المستطيل تحريك نقطة مكان لتحديد تستخدم سوف نقطة تعريف
    Private emptyPoint As Point = Point.Empty
    ' حركة أثناء الأفقي الاتجاه في المستطيل رسم نقطة الي يشير رقمي متغير تعريف
    ' الماوس
    Private x As Integer = startPoint.X

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)
        Rect = New Rectangle(x, startPoint.Y, RectangleSize.Width,
RectangleSize.Height)
        Using lgb As New Drawing2D.LinearGradientBrush(Rect, Color.FromArgb(255,
Color.DarkBlue), Color.FromArgb(255, Color.Cyan), 90, True)
            e.Graphics.FillRectangle(lgb, Rect)
        End Using
    End Sub

    Protected Overrides Sub OnMouseDown(ByVal e As
System.Windows.Forms.MouseEventArgs)
        MyBase.OnMouseDown(e)
        If e.Button = Windows.Forms.MouseButtons.Right Then Return
        ' طريق عن المستطيل تحريك نقطة أو مكان الي يشير الذي للمتغير قيمة تمرير
        ' الماوس مع ربطها
        emptyPoint.X = e.Location.X
    End Sub

    Protected Overrides Sub OnMouseMove(ByVal e As
System.Windows.Forms.MouseEventArgs)
        MyBase.OnMouseMove(e)

```

```

' بعض يمتلك بالفعل الماوس تحريك نقطة الي يشير الذي المتغير أن نتأكد هنا
القيم
If emptyPoint = Point.Empty Then Return
' الأفقي الاتجاه في المستطيل رسم بداية لنقطة جديدة قيم نمرر هما
x = e.X - emptyPoint.X + startPoint.X
' المستطيل رسم يعيد أن الفورم من نطلب هنا
Me.Invalidate()
End Sub

Protected Overrides Sub OnMouseUp(ByVal e As
System.Windows.Forms.MouseEventArgs)
MyBase.OnMouseUp(e)
' الماوس بيسار الخاص الباتون باستخدام قام المستخدم بأن نتأكد هنا
If e.Button = Windows.Forms.MouseButtons.Right Then Return
' المستطيل رسم باية نقطة الب جديدة قيمة نمرر هنا
startPoint.X += e.X - emptyPoint.X
' الأصلية قيمته الي المستطيل تحريك مكان الي يشير الذي المتغير نعيد هنا
شئ لا وهي
emptyPoint = Point.Empty
End Sub

End Class

```

المثال الخامس:

تحريك المستطيل في الاتجاه الرأسي فقط والكود التالي يوضح ذلك

كود:

```

Public Class Form1

' الرسم بداية نقطة
Private startPoint As Point = New Point(10, 10)
' المستطيل أبعاد
Private RectangleSize As Size = New Size(100, 100)
' المستطيل تعريف
Private Rect As Rectangle
' المستطيل تحريك نقطة مكان لتحديد تستخدم سوف نقطة تعريف
Private emptyPoint As Point = Point.Empty
' حركة أثناء الرأسي الاتجاه في المستطيل رسم نقطة الي يشير رقمي متغير تعريف
الماوس
Private y As Integer = startPoint.Y

Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
MyBase.OnPaint(e)
Rect = New Rectangle(startPoint.X, y, RectangleSize.Width,
RectangleSize.Height)
Using lgb As New Drawing2D.LinearGradientBrush(Rect, Color.FromArgb(255,
Color.DarkBlue), Color.FromArgb(255, Color.Cyan), 90, True)
e.Graphics.FillRectangle(lgb, Rect)
End Using
End Sub

Protected Overrides Sub OnMouseDown(ByVal e As
System.Windows.Forms.MouseEventArgs)
MyBase.OnMouseDown(e)
If e.Button = Windows.Forms.MouseButtons.Right Then Return
طريق عن المستطيل تحريك نقطة أو مكان الي يشير الذي للمتغير قيمة تمرير
الماوس مع ربطها
emptyPoint.Y = e.Location.Y

```

```

End Sub

Protected Overrides Sub OnMouseMove (ByVal e As
System.Windows.Forms.MouseEventArgs)
    MyBase.OnMouseMove (e)
    ' بعض يمتلك بالفعل الماوس تحريك نقطة الي يشير الذي المتغير أن نتأكد هنا
القيم
    If emptyPoint = Point.Empty Then Return
    ' الرأسى الاتجاه في المستطيل رسم بداية لنقطة جديدة قيم نمرر هنا
    y = e.Y - emptyPoint.Y + startPoint.Y
    ' المستطيل رسم يعيد أن الفورم من نطلب هنا
    Me.Invalidate()
End Sub

Protected Overrides Sub OnMouseUp (ByVal e As
System.Windows.Forms.MouseEventArgs)
    MyBase.OnMouseUp (e)
    ' الماوس بيسار الخاص الباتون باستخدام بأن نتأكد هنا
    If e.Button = Windows.Forms.MouseButtons.Right Then Return
    ' المستطيل رسم باية نقطة الب جديدة قيمة نمرر هنا
    startPoint.Y += e.Y - emptyPoint.Y
    ' الأصلية قيمته الي المستطيل تحريك مكان الي يشير الذي المتغير نعيد هنا
شئ لا وهي
    emptyPoint = Point.Empty
End Sub

End Class

```

في الكود الخاص بالمثال الثالث إلى الخامس أعلاه بعض القصور وهي كالتالي:

- 1- اهتزاز الرسم أثناء عملية تحريك المستطيل
- 2- عند الضغط علي الفورم في أي مكان بالمثال الثالث يتم تحريك المستطيل
- 3- عند الضغط علي الفورم في يمين أو يسار المستطيل بالمثال الرابع يتم تحريك المستطيل
- 4- عند الضغط علي الفورم في أعلي أو أسفل المستطيل بالمثال الخامس يتم تحريك المستطيل
- 5- عند تحريك المستطيل خارج أبعاد الفورم يبدأ المستطيل في الاختفاء

حاول عزيزي القارئ أن تجد حلولاً لمثل هذه المشاكل وحاول أيضاً أن تستخدم شكلاً هندسياً آخر غير المستطيل وذلك قبل أن تقرأ المشاركات التالية والخاصة بالفكرة التي نتحدث عنها في هذا الجزء الأول

ملحوظة :-

ألا يذكر الكود الموجود أعلاه بأي كمنترول تستخدمه علي سبيل المثال **TrackBar** أو **ScrollBar** وهذا يعود بنا إلي بداية المقال وهو أن مجموعة أفكار صغيرة سوف تؤدي إلي بناء فكرة كبيرة

في المشاركات السابقة من الجزء الأول كان هناك بعض القصور في الكود الذي استخدمناه في المثال الثالث إلى الخامس وفي المشاركة هذه وأيضاً في المشاركات التالية سوف نحاول التخلص من هذا القصور في الكود

- 1- لكي نتخلص من **Flicker** أو من الاهتزازات التي نراها أثناء عملية تحريك الشكل الهندسي علينا أن نقوم إما بضبط خاصية **DoubleBuffered** بحيث نجعل قيمتها تساوي **True** وهذا يحدث في مرحلة **DesignTime** أو من الممكن أن نغير القيمة برمجياً

حل آخر وهو أن نستخدم **SetStyle** لضبط القيم التي تؤثر علي **DoubleBuffering** والكود التالي يوضح كيفية عمل ذلك

كود:

```
Public Sub New()  
    InitializeComponent()  
  
    Me.SetStyle(ControlStyles.OptimizedDoubleBuffer, True)  
    Me.SetStyle(ControlStyles.AllPaintingInWmPaint, True)  
    Me.SetStyle(ControlStyles.UserPaint, True)  
  
End Sub
```

2-تحريك المستطيل في الاتجاه الأفقي فقط

المثال السادس:

لكي نتخلص من أوجه القصور الموجودة بالمثال الرابع أعلاه يجب علينا أن نعيد صياغة المتغيرات قليلا بحيث تتناسب مع احتياجات الفكرة التي نتحدث عنها وبالتالي سوف نحتاج إلى إضافة متغيرات جديدة وبشكل مختلف قليلا وهذه المتغيرات كالتالي:

- 1-تعريف متغير يشير إلى الشكل المراد تحريكه وهو عبارة عن **Rectangle**
- 2-تعريف متغير يشير إلى عرض هذه المستطيل وهو عبارة عن **Integer**
- 3-تعريف متغير يشير إلى ارتفاع المستطيل وهو عبارة عن **Integer**
- 4-تعريف متغير يشير إلى أقصى مسافة يمكن أن يتحركها المستطيل وهو عبارة عن **Integer**
- 5-تعريف متغير يشير إلى أقل مسافة يمكن أن يتحركها المستطيل وهو عبارة عن **Integer**
- 6-تعريف متغير يشير إلى نصف ارتفاع المستطيل وهو سيكون عبارة عن **Integer**
- 7-تعريف متغير يشير إلى الفارق بين أقصى مسافة يستطيع أن يتحركها المستطيل مطروحا منها عرض المستطيل وهو سيكون عبارة عن **Integer**
- 8-تعريف متغير يشير إلى أن الماوس موجود بالفعل داخل المستطيل وهو سيكون عبارة عن **Boolean**
- 9-تعريف متغير يشير إلى أن الماوس علي يمين المستطيل وهو سيكون عبارة عن **Boolean**
- 10-تعريف متغير يشير إلى أن الماوس علي يمين المستطيل وهو سيكون عبارة عن **Boolean**

والكود التالي يوضح شكل هذه المتغيرات

كود:

```
Public Class Form1  
  
    Private rect As Rectangle  
    Private IsMouseInRectLeft As Boolean = False  
    Private IsMouseInRectRight As Boolean = False  
    Private IsMouseInsideRect As Boolean = False  
    Private rectWidth As Integer = 22  
    Private rectHeight As Integer = 22  
    Private maximumHorizontal As Integer = 0  
    Private minimumHorizontal As Integer = 0  
    Private x As Integer = 0  
    Private MiddleOfRectWidth As Integer = 0  
  
End Class
```

الكود التالي يوضح كيفية ربط المتغيرات التي تم تعريفها وإعطاؤها قيما أولية لكي تتم عملية رسم المستطيل بشكل سليم

كود:

```
Private Sub SetGraphicsData()  
    rect = New Rectangle(Me.ClientRectangle.X, Me.ClientRectangle.Y, rectWidth,  
rectHeight)  
    MiddleOfRectWidth = rectWidth / 2  
    maximumHorizontal = Me.ClientRectangle.Right  
    minimumHorizontal = Me.ClientRectangle.X  
    x = maximumHorizontal - rectWidth  
End Sub
```

نضبط **Flicker** أو الاهتزازات الناتجة عن تحريك المستطيل كما سبق وأوضحنا في المشاركة السابقة وأيضا نمرر القيم الأولية لكي تحدث عملية الرسم بشكل صحيح والكود التالي يوضح ذلك

كود:

```
Public Sub New()  
    InitializeComponent()  
  
    Me.SetStyle(ControlStyles.OptimizedDoubleBuffer, True)  
    Me.SetStyle(ControlStyles.AllPaintingInWmPaint, True)  
    Me.SetStyle(ControlStyles.UserPaint, True)  
  
    SetGraphicsData()  
End Sub
```

نعيد صياغة الكود الخاص بأحداث الماوس وهي **MouseDown** و **MouseUp** و **MouseMove** بحيث نربط المتغيرات التي قمنا بتعريفها مع القيم الخاصة بمكان الماوس على الفورم والكود يوضح ذلك

كود:

```
Protected Overrides Sub OnMouseDown(ByVal e As MouseEventArgs)  
    MyBase.OnMouseDown(e)  
  
    If rect.Contains(e.Location) Then  
        IsMouseInsideRect = True  
        MiddleOfRectWidth = e.Location.X - rect.X  
    Else  
        If e.Location.X < Me.rect.X Then  
            IsMouseInRectLeft = True  
        Else  
            IsMouseInRectRight = True  
        End If  
    End If  
  
    Invalidate()  
End Sub  
  
Protected Overrides Sub OnMouseUp(ByVal e As MouseEventArgs)  
    MyBase.OnMouseUp(e)  
  
    If IsMouseInsideRect Then  
        IsMouseInsideRect = False  
        If e.Location.X > minimumHorizontal + MiddleOfRectWidth And _  
            e.Location.X < x + MiddleOfRectWidth Then  
            rect.X = e.Location.X - MiddleOfRectWidth  
            IsMouseInsideRect = False  
        End If  
    End If  
End Sub
```

```

        End If

    ElseIf IsMouseInRectLeft Then
        IsMouseInRectLeft = False

    ElseIf IsMouseInRectRight Then
        IsMouseInRectRight = False
    End If

    Invalidate()
End Sub

Protected Overrides Sub OnMouseMove(ByVal e As MouseEventArgs)
    MyBase.OnMouseMove(e)
    If IsMouseInsideRect Then
        If e.Location.X <= minimumHorizontal + MiddleOfRectWidth Then
            rect.X = minimumHorizontal
        ElseIf e.Location.X >= x + MiddleOfRectWidth Then
            rect.X = x
        Else
            rect.X = e.Location.X - MiddleOfRectWidth
        End If
        Invalidate()
    End If
End Sub

```

الآن لنرسم المستطيل علي الفورم من خلال الحدث **Paint** ويمكن هنا أن نقوم برسم أي شيء نريده بشرط أن نرسمه قبل أن نرسم المستطيل المراد تحريكه وهنا قمت برسم مستطيل سيعمل كخلفية للمستطيل المراد تحريكه والكود التالي يوضح ذلك

كود:

```

Protected Overrides Sub OnPaint(ByVal e As PaintEventArgs)
    MyBase.OnPaint(e)

    Dim backRectangle As New Rectangle(Me.ClientRectangle.X,
    Me.ClientRectangle.Y, Me.Width, rect.Height)
    Using lgbBack As New Drawing2D.LinearGradientBrush(backRectangle,
    Color.Orange, Color.Yellow, 90, True)
        e.Graphics.FillRectangle(lgbBack, backRectangle)
    End Using

    Using lgb As New Drawing2D.LinearGradientBrush(rect, Color.FromArgb(255,
    Color.Cyan), Color.FromArgb(255, Color.DarkBlue),
    Drawing2D.LinearGradientMode.Vertical)
        e.Graphics.FillRectangle(lgb, rect)
        e.Graphics.DrawRectangle(New Pen(Color.DarkBlue, 1.5F), rect)
    End Using
End Sub

```

آخر شيء يجب علينا أن نفعله هو أن نتأكد دائما أن عملية رسم المستطيل ستتم بشكل جيد وذلك عندما تتغير أبعاد الفورم أو الكونترول والحدثان المسؤولان عن ذلك هما الحدث **OnResize** وأيضا الحدث **OnSizeChanged** والكود التالي يوضح ذلك

كود:

```

Protected Overrides Sub OnResize(ByVal e As System.EventArgs)
    MyBase.OnResize(e)
    SetGraphicsData()

```

```

        Invalidate()
    End Sub

    Protected Overrides Sub OnSizeChanged(ByVal e As EventArgs)
        MyBase.OnSizeChanged(e)
        SetGraphicsData()
    End Sub

```

الكود التالي يوضح الكود النهائي والخاص بكيفية تحريك المستطيل في الاتجاه الأفقي

رمز:

كود:

```

' Horizontal Movement
Public Class Form1

    Private rect As Rectangle
    Private IsMouseInRectLeft As Boolean = False
    Private IsMouseInRectRight As Boolean = False
    Private IsMouseInsideRect As Boolean = False
    Private rectWidth As Integer = 22
    Private rectHeight As Integer = 22
    Private maximumHorizontal As Integer = 0
    Private minimumHorizontal As Integer = 0
    Private x As Integer = 0
    Private MiddleOfRectWidth As Integer = 0

    Public Sub New()
        InitializeComponent()

        Me.SetStyle(ControlStyles.OptimizedDoubleBuffer, True)
        Me.SetStyle(ControlStyles.AllPaintingInWmPaint, True)
        Me.SetStyle(ControlStyles.UserPaint, True)

        SetGraphicsData()

    End Sub

    Private Sub SetGraphicsData()
        rect = New Rectangle(Me.ClientRectangle.X, Me.ClientRectangle.Y,
rectWidth, rectHeight)
        MiddleOfRectWidth = rectWidth / 2
        maximumHorizontal = Me.ClientRectangle.Right
        minimumHorizontal = Me.ClientRectangle.X
        x = maximumHorizontal - rectWidth
    End Sub

    Protected Overrides Sub OnPaint(ByVal e As PaintEventArgs)
        MyBase.OnPaint(e)

        Dim backRectangle As New Rectangle(Me.ClientRectangle.X,
Me.ClientRectangle.Y, Me.Width, rect.Height)
        Using lgbBack As New Drawing2D.LinearGradientBrush(backRectangle,
Color.Orange, Color.Yellow, 90, True)
            e.Graphics.FillRectangle(lgbBack, backRectangle)
        End Using

        Using lgb As New Drawing2D.LinearGradientBrush(rect, Color.FromArgb(255,
Color.Cyan), Color.FromArgb(255, Color.DarkBlue),

```

```

Drawing2D.LinearGradientMode.Vertical)
    e.Graphics.FillRectangle(lgb, rect)
    e.Graphics.DrawRectangle(New Pen(Color.DarkBlue, 1.5F), rect)
End Using

End Sub

Protected Overrides Sub OnMouseDown(ByVal e As MouseEventArgs)
    MyBase.OnMouseDown(e)

    If rect.Contains(e.Location) Then
        IsMouseInsideRect = True
        MiddleOfRectWidth = e.Location.X - rect.X
    Else
        If e.Location.X < Me.rect.X Then
            IsMouseInRectLeft = True
        Else
            IsMouseInRectRight = True
        End If
    End If

    Invalidate()
End Sub

Protected Overrides Sub OnMouseUp(ByVal e As MouseEventArgs)
    MyBase.OnMouseUp(e)

    If IsMouseInsideRect Then
        IsMouseInsideRect = False
        If e.Location.X > minimumHorizontal + MiddleOfRectWidth And _
            e.Location.X < x + MiddleOfRectWidth Then
            rect.X = e.Location.X - MiddleOfRectWidth
            IsMouseInsideRect = False
        End If

    ElseIf IsMouseInRectLeft Then
        IsMouseInRectLeft = False

    ElseIf IsMouseInRectRight Then
        IsMouseInRectRight = False
    End If

    Invalidate()
End Sub

Protected Overrides Sub OnMouseMove(ByVal e As MouseEventArgs)
    MyBase.OnMouseMove(e)
    If IsMouseInsideRect Then
        If e.Location.X <= minimumHorizontal + MiddleOfRectWidth Then
            rect.X = minimumHorizontal
        ElseIf e.Location.X >= x + MiddleOfRectWidth Then
            rect.X = x
        Else
            rect.X = e.Location.X - MiddleOfRectWidth
        End If
        Invalidate()
    End If
End Sub

Protected Overrides Sub OnResize(ByVal e As System.EventArgs)
    MyBase.OnResize(e)
    SetGraphicsData()
    Invalidate()
End Sub

```



```

Protected Overrides Sub OnSizeChanged(ByVal e As EventArgs)
    MyBase.OnSizeChanged(e)
    SetGraphicsData()
End Sub

End Class

```

3-تحريك المستطيل في الاتجاه الراسي فقط

المثال السابع:

الكود والمتغيرات لا تختلف كثيرا عن الكود الخاص بتحريك الفورم في الاتجاه الأفقي والموجودة بالمثال السادس أعلاه عليك عزيزي القارئ أن تلاحظ الفارق فقط في ربط المتغيرات مع حركة الماوس الراسية في أحداث الماوس أو كيفية الرسم في الاتجاه الأفقي أو كيفية ربط أبعاد المستطيل المراد تحريكه مع أبعاد الفورم والكود النهائي لكيفية تحريك المستطيل في الاتجاه الراسي كالتالي

كود:

```

' Vertical Movement
Public Class Form3

    Private rect As Rectangle
    Private rectWidth As Integer = 22
    Private rectHeight As Integer = 22
    Private y As Integer = 0
    Private MiddleOfRectHeight As Integer = 0
    Private maximumVertical As Integer = 0
    Private minimumVertical As Integer = 0
    Private IsMouseAboveRectangle As Boolean = False
    Private IsMouseUnderRectangle As Boolean = False
    Private IsMouseInsideRect As Boolean = False

    Public Sub New()
        InitializeComponent()

        Me.SetStyle(ControlStyles.OptimizedDoubleBuffer, True)
        Me.SetStyle(ControlStyles.AllPaintingInWmPaint, True)
        Me.SetStyle(ControlStyles.UserPaint, True)

        SetGraphicsData()

    End Sub

    Private Sub SetGraphicsData()
        rect = New Rectangle(Me.ClientRectangle.X, Me.ClientRectangle.Y,
rectWidth, rectHeight)
        MiddleOfRectHeight = rectHeight / 2
        maximumVertical = Me.ClientRectangle.Bottom
        minimumVertical = Me.ClientRectangle.Y
        y = maximumVertical - rectHeight
    End Sub

    Protected Overrides Sub OnPaint(ByVal e As PaintEventArgs)
        MyBase.OnPaint(e)

        Dim backRectangle As New Rectangle(Me.ClientRectangle.X,
Me.ClientRectangle.Y, rect.Width, Me.Height)
        Using lgbBack As New Drawing2D.LinearGradientBrush(backRectangle,
Color.Orange, Color.Yellow, 180, True)

```

```

        e.Graphics.FillRectangle(lgbBack, backRectangle)
    End Using

    Using lgb As New Drawing2D.LinearGradientBrush(rect, Color.FromArgb(255,
Color.Cyan), Color.FromArgb(255, Color.DarkBlue),
Drawing2D.LinearGradientMode.Horizontal)
        e.Graphics.FillRectangle(lgb, rect)
        e.Graphics.DrawRectangle(New Pen(Color.DarkBlue, 1.5F), rect)
    End Using

End Sub

Protected Overrides Sub OnMouseDown(ByVal e As MouseEventArgs)
    MyBase.OnMouseDown(e)

    If rect.Contains(e.Location) Then
        IsMouseInsideRect = True
        MiddleOfRectHeight = e.Location.Y - rect.Y
    Else
        If e.Location.Y < Me.rect.Y Then
            IsMouseAboveRectangle = True
        Else
            IsMouseUnderRectangle = True
        End If
    End If

    Invalidate()
End Sub

Protected Overrides Sub OnMouseUp(ByVal e As MouseEventArgs)
    MyBase.OnMouseUp(e)

    If IsMouseInsideRect Then
        IsMouseInsideRect = False
        If e.Location.Y > minimumVertical + MiddleOfRectHeight And _
            e.Location.Y < y + MiddleOfRectHeight Then
            rect.Y = e.Location.Y - MiddleOfRectHeight
            IsMouseInsideRect = False
        End If

    ElseIf IsMouseAboveRectangle Then
        IsMouseAboveRectangle = False

    ElseIf IsMouseUnderRectangle Then
        IsMouseUnderRectangle = False
    End If

    Invalidate()
End Sub

Protected Overrides Sub OnMouseMove(ByVal e As MouseEventArgs)
    MyBase.OnMouseMove(e)
    If IsMouseInsideRect Then
        If e.Location.Y <= minimumVertical + MiddleOfRectHeight Then
            rect.Y = minimumVertical
        ElseIf e.Location.Y >= y + MiddleOfRectHeight Then
            rect.Y = y
        Else
            rect.Y = e.Location.Y - MiddleOfRectHeight
        End If
        Invalidate()
    End If
End Sub

```

```

Protected Overrides Sub OnResize(ByVal e As System.EventArgs)
    MyBase.OnResize(e)
    SetGraphicsData()
    Invalidate()
End Sub

Protected Overrides Sub OnSizeChanged(ByVal e As EventArgs)
    MyBase.OnSizeChanged(e)
    SetGraphicsData()
End Sub

End Class

```

بالنسبة لتحريك المستطيل في جميع الاتجاهات سوف اترك ذلك للقارئ ليقوم بكتابة الكود الخاص بذلك وبشكل عام هو سيكون خليطا من الكود المستخدم في المثال السادس بالمشاركة السابقة والمثال السابع أعلاه

وبذلك نكون قد انتهينا من الجزء الأول لهذا المقال

مقال: أفكار في الجرافكس الجزء الثاني

مقدمة :

كما يعلم القارئ أن جميع المبرمجين يحرصون علي أن تظهر البرامج التي يصنعونها بشكل جذاب بل ويحاولون إضفاء لمسات جمالية بطرق مختلفة حتى يجذبون المستخدم لبرامجهم لذلك ستجد الجميع يهتمون بمثل هذه الأمور حيث أنها تمثل سوقا تجاريا لا بأس به وبشكل عام هذا ما سنحاول أن نوضح بعضا منه في هذا الموضوع وكما سبق وكتبت في الجزء الأول أننا هنا لا نناقش كيف نرسم أو كيف نستخدم أوامر الرسم المختلفة بل سوف نناقش كيف نستفيد ونطور في أوامر **+GDI** الموجودة في الدوت نت بحيث نستفيد منها بشكل أفضل لذلك أفترض أن قارئ هذا المقال لديه إلمام جيد بأوامر **+GDI** وكيفية استخدامها

الفكرة الثانية:

الفكرة التي سوف نتحدث عنها اليوم هي كيف نرسم **Ribbon** أو كيف نرسم **Glow**

المثال الأول:

كيف يمكن أن نستفيد من الأوامر الخاصة برسم الأقواس بالإضافة إلى رسم المسارات أو ما يعرف باسم **GraphicsPath**الكود التالي يوضح كيف نرسم قوس من دائرة في داخل مستطيل

كود:

```

Public Class Form1

    Private DarkColor As Color = Color.DarkOrange
    Private LightColor As Color = Color.Orange
    Private rect As New Rectangle(10, 10, 150, 150)
    Private angle As Integer

    Protected Overrides Sub OnPaint(ByVal e As System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        شكل سيكون كيف لتوضيح فقط كخلفية يعمل لكي المستطيل هذا رسم من الغرض '
        نرسمه الذي القوس
        e.Graphics.FillRectangle(Brushes.Black, rect)
        angle = 180
    End Sub
End Class

```

```

        Using Path As Drawing2D.GraphicsPath = New Drawing2D.GraphicsPath()
        Path.AddArc(Rect, angle, 180)
        Using p As Pen = New Pen(DarkColor)
            Using lgb As New LinearGradientBrush(Path.GetBounds(),
Color.FromArgb(84, LightColor), Color.FromArgb(128, LightColor),
LinearGradientMode.Vertical)
                e.Graphics.FillPath(lgb, Path)
            End Using
            e.Graphics.DrawPath(p, Path)
        End Using
    End Using
End Sub

End Class

```

أرجو عزيزي القارئ أن تلاحظ قيمة المتغير **Angle** وقيمهته بالمثل أعلاه تساوي صفر حاول عزيزي القارئ أن تقوم بتغيير قيمة هذا المتغير لكي يساوي ١٨٠ أو ٩٠ أو ٢٧٠ ولاحظ الفرق في كل مرة ستجرب فيها الكود ستكتشف أن اتجاه القوس المرسوم داخل المستطيل يتغير تبعاً للزاوية التي نحددها.....

المثال الثاني :

الآن لنرسم نفس القوس ولكن هذه المرة سوف نقوم بعمل **Blend** للألوان المستخدمة في رسم القوس وسنكرر الرسم فقط من أجل المقارنة بين ما رسمناه في المثال الأول والمثال الثاني

كود:

```

Public Class Form1

    Private DarkColor As Color = Color.DarkOrange
    Private LightColor As Color = Color.Orange
    Private rect As New Rectangle(10, 10, 150, 150)
    Private angle As Integer

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        Dim m_blend As New Blend
        m_blend.Factors = New Single() {0.0F, 0.4F, 0.8F, 1.0F}
        m_blend.Positions = New Single() {0.0F, 0.3F, 0.4F, 1.0F}

        ' شكل سيكون كيف لتوضيح فقط كخلفية يعمل أن هو المستطيل هذا رسم من الغرض '
        ' نرسمه الذي القوس
        e.Graphics.FillRectangle(Brushes.Black, Rect)

        angle = 180
        Using Path As Drawing2D.GraphicsPath = New Drawing2D.GraphicsPath()
            Path.AddArc(rect, angle, 180)
            Using p As Pen = New Pen(DarkColor)
                Using lgb As New LinearGradientBrush(Path.GetBounds(),
Color.FromArgb(84, LightColor), Color.FromArgb(128, LightColor),
LinearGradientMode.Vertical)
                    lgb.Blend = m_blend
                    e.Graphics.FillPath(lgb, Path)
                End Using
                e.Graphics.DrawPath(p, Path)
            End Using
        End Using
    End Using
End Class

```

```

شكل سيكون كيف لتوضيح فقط كخلفية يعمل أن هو المستطيل هذا رسم من الغرض '
نرسمه الذي القوس
Dim rect1 As Rectangle = New Rectangle(Rect.X + Rect.Width + 5, Rect.Y,
Rect.Width, Rect.Height)
e.Graphics.FillRectangle(Brushes.Black, rect1)

غير لا فقط المقارنة أجل من الجزء هذا رسم إعادة يتم '
Using Path As Drawing2D.GraphicsPath = New Drawing2D.GraphicsPath()
    Path.AddArc(rect1, angle, 180)
    Using p As Pen = New Pen(DarkColor)
        Using lgb As New LinearGradientBrush(Path.GetBounds(),
Color.FromArgb(84, LightColor), Color.FromArgb(128, LightColor),
LinearGradientMode.Vertical)
            e.Graphics.FillPath(lgb, Path)
        End Using
        e.Graphics.DrawPath(p, Path)
    End Using
End Using
End Sub
End Class

```

لو نظرت جيدا إلى ما رسمناه ستجد أن هناك فارقا قد لا يبدو ظاهرا للعين لأول وهلة لكن لو ركزت عينيك علي ما رسمناه ستجد فارقا في تدرج الألوان بسبب استخدام **Blend** حاول تغيير قيم **Positions** و **Factors** الخاصة بالكلاس **Blend** لتكتشف بنفسك أهمية هذا الكلاس في عملية التحكم في تدرج الألوان

المثال الثالث:

سنعيد رسم المثال الثاني كما هو ولكن بدلا من أن نستخدم مستطيلا كخلفية للأقواس المرسومة سنرسم **Ellipse** جرب المثال ولاحظ الفارق

كود:

```

Public Class Form1

    Private DarkColor As Color = Color.DarkOrange
    Private LightColor As Color = Color.Orange
    Private rect As New Rectangle(10, 10, 150, 150)
    Private angle As Integer

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        Dim m_blend As New Blend
        m_blend.Factors = New Single() {0.0F, 0.4F, 0.8F, 1.0F}
        m_blend.Positions = New Single() {0.0F, 0.3F, 0.4F, 1.0F}

        ' السابق المثال في الموجود المستطيل من بدلا بيضاوي شكل رسم تم هنا '
        e.Graphics.FillEllipse(Brushes.Black, rect)

        angle = 180
        Using Path As Drawing2D.GraphicsPath = New Drawing2D.GraphicsPath()
            Path.AddArc(rect, angle, 180)
            Using p As Pen = New Pen(DarkColor)
                Using lgb As New LinearGradientBrush(Path.GetBounds(),
Color.FromArgb(84, LightColor), Color.FromArgb(128, LightColor),
LinearGradientMode.Vertical)
                    lgb.Blend = m_blend
                End Using
            End Using
        End Using
    End Sub
End Class

```

```

        e.Graphics.FillPath(lgb, Path)
    End Using
    e.Graphics.DrawPath(p, Path)
End Using
End Using

' غير لا فقط المقارنة أجل من الجزء هذا رسم إعادة يتم '
Dim rect1 As Rectangle = New Rectangle(rect.X + rect.Width + 5, rect.Y,
rect.Width, rect.Height)
' السابق المثال في الموجود المستطيل من بدلا بيضاوي شكل رسم تم هنا '
e.Graphics.FillEllipse(Brushes.Black, rect1)

Using Path As Drawing2D.GraphicsPath = New Drawing2D.GraphicsPath()
    Path.AddArc(rect1, angle, 180)
    Using p As Pen = New Pen(DarkColor)
        Using lgb As New LinearGradientBrush(Path.GetBounds(),
Color.FromArgb(84, LightColor), Color.FromArgb(128, LightColor),
LinearGradientMode.Vertical)
            e.Graphics.FillPath(lgb, Path)
        End Using
        e.Graphics.DrawPath(p, Path)
    End Using
End Using
End Sub
End Class

```

لو نظرت إلى الرسم جيدا ستجد أن حواف الأقواس المرسومة بها عيوب ظاهرة للعين لذلك يفضل أن نضبط قيمة **SmoothingMode.AntiAlias** ونجعل قيمتها تساوي

حاول أيضا عزيزي القارئ أن تقوم بتغيير الألوان المستخدمة في الرسم لكي تلاحظ الفوارق المختلفة في الأمثلة السابقة

المثال الرابع:

قد يتساءل البعض كيف نتحكم في ارتفاع القوس وهنا سنحاول أن نرسم نفس الكود أعلاه ولكن بأسلوب مختلف قليلا

كود:

```

Public Class Form1

    Private DarkColor As Color = Color.Black
    Private LightColor As Color = Color.Orange
    Private rect As New Rectangle(10, 10, 150, 150)
    Private startAngle As Integer
    Private sweepAngle As Integer = 100

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        e.Graphics.SmoothingMode = SmoothingMode.AntiAlias

        Using path = New GraphicsPath
            startAngle = ((180 - sweepAngle) / 2)
            path.AddArc(rect, CSng(startAngle), CSng(sweepAngle))
            Dim pt As Point = Point.Round(path.PathData.Points(0))
            Using lgb = New LinearGradientBrush(New Point(Rect.Left,
Rect.Bottom), New Point(Rect.Left, (pt.Y - 1)), DarkColor, Color.FromArgb(50,
DarkColor))
                e.Graphics.FillPath(lgb, path)
            End Using
        End Using
    End Sub
End Class

```

```

        End Using
    End Using

    End Sub

End Class

```

أرجو أن تلاحظ عزيزي القارئ كيف يمكن التحكم في ارتفاع القوس عن طريق الربط بين قيمة **StartAngle** و **SweepAngle** معا حيث أن الفارق بين القيمتين يعبر عن الفارق في الارتفاع بين النقطة التي تمثل مركز المستطيل والنقطة التي يبدأ عندها رسم القوس

أيضا أرجو أن تلاحظ كيف تم تحديد أول نقطة موجودة بالمسار **GraphicsPath** عن طريق تعريف متغير عبارة عن **Point** وكيف تم ربطها مع **PathData** وهو عبارة عن **ReadOnly Property** خاصة بالمسار وهي عبارة عن **Array** تحدد النقط الكلية الخاصة بشكل المسار **GraphicsPath** حيث باستخدامها يمكن استرجاع أول وآخر نقطة بالمسار فقط لا غير أما باقي النقاط التي تمثل شكل المسار فهناك استحالة تامة لاسترجاع باقي القيم الخاصة بها

المثال الخامس :

سنعيد رسم الفكرة الموجودة بالمثال السابق مع إضافة مستطيل آخر إلى المسار وأرجو أن تلاحظ هنا ما سيحدث

كود:

```

Public Class Form1

    Private DarkColor As Color = Color.Black
    Private LightColor As Color = Color.Orange
    Private rect As New Rectangle(10, 10, 150, 150)
    Private startAngle As Integer
    Private sweepAngle As Integer = 130

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        e.Graphics.SmoothingMode = SmoothingMode.AntiAlias

        Using path = New GraphicsPath
            startAngle = ((180 - sweepAngle) / 2)
            path.AddArc(rect, CSng(startAngle), CSng(sweepAngle))
            Dim pt As Point = Point.Round(path.PathData.Points(0))

            Dim r As Rectangle = rect
            r.Inflate(-1, -1)
            path.AddArc(r, CSng(startAngle), CSng(sweepAngle))

            Using lgb = New LinearGradientBrush(New Point(rect.Left,
rect.Bottom), New Point(rect.Left, (pt.Y - 1)), DarkColor, Color.FromArgb(50,
DarkColor))
                e.Graphics.FillPath(lgb, path)
            End Using
        End Using

    End Sub

End Class

```

ستلاحظ وبالرغم من أننا نحاول أن نملأ المسار بأحد الألوان فإن ذلك لا يحدث وبدلا من ذلك يظهر لنا قوس عبارة عن جزء من محيط دائرة وبالرغم من ذلك يمكنك أن تستخدم هذا المثال في رسم إطار علي شكل قوس

أرجو أن تحاول عزيزي القارئ أن تجرب جميع الأمثلة أعلاه لكي يتضح لك مدى أهمية كلا من الأقواس **Arcs** و المسار **GraphicsPath** في طرق أو عمليات الرسم **+GDI** المختلفة

في الأمثلة التالية سنحاول أن نغطي كيفية التحكم في الألوان حيث يمكن أن نجعل أي لون نختاره بأن نجعله **Light** أو **Dark** أي و ببساطة شديدة سنحاول أن نجعل اللون الذي نحدده إما غامقا أو فاتحا ومن ثم نحاول أن نستفيد من هذه الميزة في عمليات الرسم

عموما هناك أفكار كثيرة لكي نتحكم في الألوان وهي ميزة هامة جدا في عمليات **+GDI** المختلفة حيث أن التحكم في الألوان ودرجاتها وقيمة **alpha** الخاصة بها ستساعدك كثيرا عزيزي القارئ في الخروج بنتائج رائعة لكي تجعل ما ترسمه يأخذ أشكالا رائعة

المثال السادس:

تخفيف الألوان والمثال التالي يوضح ذلك

لكن عليك أن تلاحظ عزيزي القارئ أهمية استخدام هذا السطر الموجود بالكود **Igb.SetSigmaBellShape(0.9)** كود:

```
Public Class Form1

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        ' الأصلي اللون الي يشير متغير تعريف
        Dim baseColor As Color = Color.Orange
        Dim rect As New Rectangle(10, 10, 150, 150)

        ' الأصلي اللون لتخفيف نحتاجها التي الدرجة تحدد سوف القيمة هذه
        ' 100 عن تزيد لا أن ويجب صفر عن تقل لا أن يجب القيمة هذه
        Dim percent As Integer = 80

        ' الجديد باللون الخاصة القيم تحديد
        Dim alpha As Integer = baseColor.A
        Dim red = baseColor.R + CInt(((255 - baseColor.R) / 100) * percent)
        Dim green = baseColor.G + CInt(((255 - baseColor.G) / 100) * percent)
        Dim blue = baseColor.B + CInt(((255 - baseColor.B) / 100) * percent)
        ' الجديد اللون تعريف
        Dim lightColor As Color = Color.FromArgb(alpha, red, green, blue)

        ' معا الجديد واللون الأصلي اللون باستخدام شي أي رسم
        Using lgb As New Drawing2D.LinearGradientBrush(rect, baseColor,
lightColor, Drawing2D.LinearGradientMode.Vertical)
            lgb.SetSigmaBellShape(0.9)
            e.Graphics.FillEllipse(lgb, rect)
        End Using

    End Sub

End Class
```

المثال السابع:

تعميق الألوان والمثال التالي يوضح ذلك

كود:

```
Public Class Form1

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        ' الأصلي اللون الي يشير متغير تعريف
        Dim baseColor As Color = Color.Orange
        Dim rect As New Rectangle(10, 10, 150, 150)

        ' الأصلي اللون لتغميق نحتاجها التي الدرجة تحدد سوف القيمة هذه
        ' 100 عن تزيد لا أن ويجب صفر عن تقل لا أن يجب القيمة هذه
        Dim percent As Integer = 80

        ' الأصلي اللون من غامق لون علي الحصول
        Dim alphaDark As Integer = baseColor.A
        Dim redDark = baseColor.R - CInt(((baseColor.R) / 100) * percent)
        Dim greenDark = baseColor.G - CInt(((baseColor.G) / 100) * percent)
        Dim blueDark = baseColor.B - CInt(((baseColor.B) / 100) * percent)
        ' الجديد اللون تعريف
        Dim darkColor As Color = Color.FromArgb(alphaDark, redDark, greenDark,
blueDark)

        ' معا الجديد واللون الأصلي اللون باستخدام شي أي رسم
        Using lgb As New Drawing2D.LinearGradientBrush(rect, baseColor,
darkColor, Drawing2D.LinearGradientMode.Vertical)
            lgb.SetSigmaBellShape(0.9)
            e.Graphics.FillEllipse(lgb, rect)
        End Using

    End Sub

End Class
```

المثال الثامن:

هنا سنحاول استخدام المثال السادس والسابع لنخرج بكود مختلف قليلا وعليك عزيزي القارئ أن تلاحظ الفارق حاول أن تتحكم في قيمة المتغير الذي يؤثر علي درجة تفتيح أو تغميق اللون

كود:

```
Public Class Form1

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        ' الأصلي اللون الي يشير متغير تعريف
        Dim baseColor As Color = Color.Orange
        Dim rect As New Rectangle(10, 10, 150, 150)

        ' الأصلي اللون لتغميق نحتاجها التي الدرجة تحدد سوف القيمة هذه
        ' 100 عن تزيد لا أن ويجب صفر عن تقل لا أن يجب القيمة هذه
        Dim percent As Integer = 20

        ' الأصلي اللون من غامق لون علي الحصول
        Dim alphaDark As Integer = baseColor.A
        Dim redDark = baseColor.R - CInt(((baseColor.R) / 100) * percent)
        Dim greenDark = baseColor.G - CInt(((baseColor.G) / 100) * percent)
        Dim blueDark = baseColor.B - CInt(((baseColor.B) / 100) * percent)
```

```

' الجديد اللون تعريف
Dim darkColor As Color = Color.FromArgb(alphaDark, redDark, greenDark,
blueDark)

' الفاتح اللون درجة في نتحكم حتي المتغير هذا قيمة تغيير
percent = 80
' الأصلي اللون من خفيف لون علي الحصول
Dim alphaLight As Integer = baseColor.A
Dim redLight = baseColor.R + CInt(((255 - baseColor.R) / 100) * percent)
Dim greenLight = baseColor.G + CInt(((255 - baseColor.G) / 100) *
percent)
Dim blueLight = baseColor.B + CInt(((255 - baseColor.B) / 100) *
percent)
Dim lightColor As Color = Color.FromArgb(alphaLight, redLight,
greenLight, blueLight)

' معا الفاتح واللون الغامق اللون باستخدام شي أي رسم
Using lgb As New Drawing2D.LinearGradientBrush(rect, lightColor,
darkColor, Drawing2D.LinearGradientMode.Vertical)
    lgb.SetSigmaBellShape(0.9)
    e.Graphics.FillEllipse(lgb, rect)
End Using
End Sub

End Class

```

أعتقد أننا نستطيع أن نبني من الأفكار أعلاه دالتين أحدهما تستخدم في تخفيف اللون والأخرى في تغميق اللون

جرب وحاول بنفسك عزيز القارئ أن تبني مثل هاتين الدالتين

المثال التالي سنحاول أن نخلط بين جميع الأمثلة السابقة لكي نخرج بالشكل النهائي لما نريد أن نرسمه ومن ثم نحاول أن نكتب روتين نستخدمه فيما بعد في أي مشروع

المثال التاسع:

كود:

```

Imports System.Drawing.Drawing2D

Public Class Form1

    Private DarkColor As Color = Color.DarkOrange
    Private LightColor As Color = Color.Orange
    Private rect As New Rectangle(10, 10, 150, 150)
    Private angle As Integer = 180

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        Dim DarkColor As Color = Color.Orange
        Dim LightColor As Color = Color.Yellow
        Dim rect As New Rectangle(10, 10, 100, 100)
        Dim WhiteColor As Color = Color.White
        Dim darkRatio As Integer = 16
        Dim scale As SizeF = New SizeF(CSng(rect.Width / 100), CSng(rect.Height
/ 100))
        Dim sigmaBlend As New Blend
        sigmaBlend.Factors = New Single() {0.0F, 0.4F, 0.8F, 1.0F}
        sigmaBlend.Positions = New Single() {0.0F, 0.3F, 0.4F, 1.0F}

```

```

Dim darkerColor As Color = DarkColor

Dim mode As SmoothingMode = e.Graphics.SmoothingMode
e.Graphics.SmoothingMode = SmoothingMode.AntiAlias

Dim darkerColorAlpha As Integer = darkerColor.A
Dim darkerColorRed As Integer = darkerColor.R - CInt(((darkerColor.R) /
100) * darkRatio)
Dim darkerColorGreen As Integer = darkerColor.G - CInt(((darkerColor.G)
/ 100) * darkRatio)
Dim darkerColorBlue As Integer = darkerColor.B - CInt(((darkerColor.B) /
100) * darkRatio)

Dim glowColor As Color = Color.FromArgb(darkerColorAlpha,
darkerColorRed, darkerColorGreen, darkerColorBlue)

Using Path As Drawing2D.GraphicsPath = New Drawing2D.GraphicsPath()
    Path.AddEllipse(rect)
    Dim pgb As New Drawing2D.PathGradientBrush(Path)
    pgb.CenterPoint = New PointF(rect.Left + (rect.Width / 2),
rect.Bottom + (33 * scale.Height))
    pgb.CenterColor = Color.FromArgb(255, LightColor)
    ' Creates a gradient falloff based on a bell-shaped curve.
    pgb.SetSigmaBellShape(0.85)

    If darkerColor.GetBrightness >= 0.3 Then
        darkerColor = glowColor
    End If

    pgb.SurroundColors = New Color() {Color.FromArgb(255, darkerColor)}
    e.Graphics.FillEllipse(pgb, rect)
End Using

Using Path As Drawing2D.GraphicsPath = New Drawing2D.GraphicsPath()
    Path.AddArc(rect, angle, angle)
    Using p As Pen = New Pen(glowColor)
        Using lgb As New LinearGradientBrush(Path.GetBounds(),
Color.FromArgb(64, WhiteColor), Color.FromArgb(84, WhiteColor),
LinearGradientMode.Vertical)
            lgb.Blend = sigmaBlend
            e.Graphics.FillPath(lgb, Path)
        End Using
        e.Graphics.DrawPath(p, Path)
    End Using
End Using

Dim blender As Drawing2D.Blend = New Drawing2D.Blend()
blender.Positions = New Single() {0, 0.9, 1}
blender.Factors = New Single() {0, 1, 1}

Dim sweepRectangle As Rectangle = rect
sweepRectangle.Height -= (rect.Height / 2)
sweepRectangle.Inflate(-16 * scale.Width, -3 * scale.Height)
Using lgb As New Drawing2D.LinearGradientBrush(sweepRectangle,
WhiteColor, Color.Empty, 90, True)
    sweepRectangle.Height -= scale.Height / 2
    lgb.Blend = blender
    e.Graphics.FillEllipse(lgb, sweepRectangle)
End Using

Using borderPen As New Pen(glowColor, 2)
    Dim borderRect As Rectangle = rect
    e.Graphics.DrawEllipse(borderPen, borderRect)

```

```
End Using
```

```
e.Graphics.SmoothingMode = mode
```

```
End Sub
```

```
End Class
```

بالتأكيد يمكننا أن نرسم ما نريد أو نعدل بالكود كيفما شئنا ولكنك سوف تلاحظ عزيزي القارئ أننا لم نأتي بشيء جديد كل ما في الأمر أننا استخدمنا أوامر **+GDI** المختلفة والموجودة بالفعل بالدوت نت ولكن بصورة أكثر فعالية حتى نحصل علي الشكل النهائي لما نريد

وفي الكود أعلاه حاول أن تلاحظ كيفية استخدام الكلاسات **Blend** و **LinearGradientBrush** و **PathGradientBrush** ولاحظ أيضا كيف نتأكد من **Brightness** الخاصة بلون ما فإذا كانت أقل من ما نريد فإننا نقوم بتغيير اللون إلى لون آخر قاتم قليلا ولكنه يأتي من نفس اللون

الكود التالي يوضح شكل الروتين النهائي الذي من الممكن أن نستخدمه في أي مشروع بحيث نمرر له بعض المتغيرات مثل **Graphics Object** و **Rectangle** وأيضا لونين مختلفين أحدهما سيمثل اللون الغامق والأخر عبارة عن اللون الفاتح

كود:

```
Private Sub DrawGlowRibbon(ByVal g As Graphics, ByVal rect As Rectangle, ByVal darkColor As Color, ByVal lightColor As Color)

    Dim mode As SmoothingMode = g.SmoothingMode
    g.SmoothingMode = SmoothingMode.AntiAlias

    Dim darkRatio As Integer = 16
    Dim whiteColor As Color = Color.White

    Dim sigmaBlend As New Blend
    sigmaBlend.Factors = New Single() {0.0F, 0.4F, 0.8F, 1.0F}
    sigmaBlend.Positions = New Single() {0.0F, 0.3F, 0.4F, 1.0F}

    Dim scale As SizeF = New SizeF(CSng(rect.Width / 100), CSng(rect.Height / 100))
    Dim darkerColor As Color = darkColor

    Dim darkerColorAlpha As Integer = darkerColor.A
    Dim darkerColorRed As Integer = darkerColor.R - CInt(((darkerColor.R) / 100) * darkRatio)
    Dim darkerColorGreen As Integer = darkerColor.G - CInt(((darkerColor.G) / 100) * darkRatio)
    Dim darkerColorBlue As Integer = darkerColor.B - CInt(((darkerColor.B) / 100) * darkRatio)
    Dim glowColor As Color = Color.FromArgb(darkerColorAlpha, darkerColorRed, darkerColorGreen, darkerColorBlue)

    Using Path As Drawing2D.GraphicsPath = New Drawing2D.GraphicsPath()
        Path.AddEllipse(rect)
        Dim pgb As New Drawing2D.PathGradientBrush(Path)
        pgb.CenterPoint = New PointF(rect.Left + (rect.Width / 2), rect.Bottom + (33 * scale.Height))
        pgb.CenterColor = Color.FromArgb(255, lightColor)
        pgb.SetSigmaBellShape(0.85)
        If darkerColor.GetBrightness >= 0.3 Then
            darkerColor = glowColor
        End If
        pgb.SurroundColors = New Color() {Color.FromArgb(255, darkerColor)}
        g.FillEllipse(pgb, rect)
    End Using
End Sub
```

```

End Using

Using Path As Drawing2D.GraphicsPath = New Drawing2D.GraphicsPath()
    Path.AddArc(rect, 180, 180)
    Using p As Pen = New Pen(glowColor)
        Using lgb As New LinearGradientBrush(Path.GetBounds(),
Color.FromArgb(64, WhiteColor), Color.FromArgb(84, WhiteColor),
LinearGradientMode.Vertical)
            lgb.Blend = sigmaBlend
            g.FillPath(lgb, Path)
        End Using
    End Using
    g.DrawPath(p, Path)
End Using
End Using

Dim blender As Drawing2D.Blend = New Drawing2D.Blend()
blender.Positions = New Single() {0, 0.9, 1}
blender.Factors = New Single() {0, 1, 1}

Dim sweepRectangle As Rectangle = rect
sweepRectangle.Height -= (rect.Height / 2)
sweepRectangle.Inflate(-16 * scale.Width, -3 * scale.Height)
Using lgb As New Drawing2D.LinearGradientBrush(sweepRectangle, WhiteColor,
Color.Empty, 90, True)
    sweepRectangle.Height -= scale.Height / 2
    lgb.Blend = blender
    g.FillEllipse(lgb, sweepRectangle)
End Using

Using borderPen As New Pen(darkerColor, 2)
    Dim borderRect As Rectangle = rect
    g.DrawEllipse(borderPen, borderRect)
End Using

g.SmoothingMode = mode
End Sub

```

التطبيق النهائي:

في المثال التالي سنحاول أن نبنى باتون شكله لطيف وهي فكرة بسيطة جدا لكيفية استخدام الأفكار أعلاه

كود:

```

Imports System.Drawing.Drawing2D

Public Class RibbonButton
    Inherits Control

    Private DarkColor As Color
    Private LightColor As Color
    Private rect As Rectangle
    Private Path As GraphicsPath
    Private m_state As ButtonState = ButtonState.Normal

    Public Sub New()

        Me.SetStyle(ControlStyles.OptimizedDoubleBuffer, True)
        Me.SetStyle(ControlStyles.AllPaintingInWmPaint, True)
        Me.SetStyle(ControlStyles.UserPaint, True)
        Me.UpdateStyles()

        Me.SetNormalstate()
    End Sub

```

```

        Me.SetGraphicsData()

    End Sub

    Protected Overrides ReadOnly Property DefaultSize() As System.Drawing.Size
        Get
            Return New Size(50, 50)
        End Get
    End Property

    Private Sub SetGraphicsData()
        Path = New GraphicsPath()
        Path.AddEllipse(Me.ClientRectangle)
        Dim rgn As New Region(Path)
        Me.Region = rgn
        rect = New Rectangle(Me.ClientRectangle.X + 1, Me.ClientRectangle.Y + 1,
Me.ClientRectangle.Width - 2, Me.ClientRectangle.Height - 2)
    End Sub

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        Select Case m_state

            Case ButtonState.Default, ButtonState.Disabled, ButtonState.Normal
                SetNormalstate()
                DrawGlowRibbon(e.Graphics, rect, DarkColor, LightColor)

            Exit Select

            Case ButtonState.Hot
                SetHoverstate()
                DrawGlowRibbon(e.Graphics, rect, DarkColor, LightColor)
            Exit Select

            Case ButtonState.Pressed

                SetPressedState()
                DrawGlowRibbon(e.Graphics, rect, DarkColor, LightColor)
        End Select

    End Sub

    Protected Overrides Sub OnMouseEnter(ByVal e As System.EventArgs)
        MyBase.OnMouseEnter(e)
        m_state = ButtonState.Hot
        Invalidate()
    End Sub

    Protected Overrides Sub OnMouseDown(ByVal e As
System.Windows.Forms.MouseEventArgs)
        MyBase.OnMouseDown(e)

        m_state = ButtonState.Pressed
        Invalidate()

    End Sub

    Protected Overrides Sub OnMouseUp(ByVal e As
System.Windows.Forms.MouseEventArgs)
        MyBase.OnMouseUp(e)
        OnMouseEnter(e)
    End Sub

```

```

Protected Overrides Sub OnMouseMove(ByVal e As
System.Windows.Forms.MouseEventArgs)
    MyBase.OnMouseMove(e)

    If (e.Button And MouseButton.Left) = MouseButton.Left And Not
rect.Contains(e.Location) And m_state = ButtonState.Pressed Then
        OnMouseLeave(e)
    End If
End Sub

Protected Overrides Sub OnMouseLeave(ByVal e As System.EventArgs)
    MyBase.OnMouseLeave(e)

    m_state = ButtonState.Default
    Invalidate()

End Sub

Protected Overrides Sub OnResize(ByVal e As System.EventArgs)
    MyBase.OnResize(e)
    SetGraphicsData()
    Invalidate()
End Sub

Protected Overrides Sub OnSizeChanged(ByVal e As EventArgs)
    MyBase.OnSizeChanged(e)
    SetGraphicsData()
End Sub

Private Sub DrawGlowRibbon(ByVal g As Graphics, ByVal rect As Rectangle,
ByVal darkColor As Color, ByVal lightColor As Color)

    Dim mode As SmoothingMode = g.SmoothingMode
    g.SmoothingMode = SmoothingMode.AntiAlias

    Dim darkRatio As Integer = 16
    Dim WhiteColor As Color = Color.White

    Dim sigmaBlend As New Blend
    sigmaBlend.Factors = New Single() {0.0F, 0.4F, 0.8F, 1.0F}
    sigmaBlend.Positions = New Single() {0.0F, 0.3F, 0.4F, 1.0F}

    Dim scale As SizeF = New SizeF(CSng(rect.Width / 100), CSng(rect.Height
/ 100))
    Dim darkerColor As Color = darkColor

    Dim darkerColorAlpha As Integer = darkerColor.A
    Dim darkerColorRed As Integer = darkerColor.R - CInt(((darkerColor.R) /
100) * darkRatio)
    Dim darkerColorGreen As Integer = darkerColor.G - CInt(((darkerColor.G)
/ 100) * darkRatio)
    Dim darkerColorBlue As Integer = darkerColor.B - CInt(((darkerColor.B) /
100) * darkRatio)
    Dim glowColor As Color = Color.FromArgb(darkerColorAlpha,
darkerColorRed, darkerColorGreen, darkerColorBlue)

    Using Path As Drawing2D.GraphicsPath = New Drawing2D.GraphicsPath()
        Path.AddEllipse(rect)
        Dim pgb As New Drawing2D.PathGradientBrush(Path)
        pgb.CenterPoint = New PointF(rect.Left + (rect.Width / 2),
rect.Bottom + (33 * scale.Height))
        pgb.CenterColor = Color.FromArgb(255, lightColor)
        pgb.SetSigmaBellShape(0.85)
        If darkerColor.GetBrightness >= 0.3 Then

```

```

        darkerColor = glowColor
    End If
    pgb.SurroundColors = New Color() {Color.FromArgb(255, darkerColor)}
    g.FillEllipse(pgb, rect)
End Using

Using Path As Drawing2D.GraphicsPath = New Drawing2D.GraphicsPath()
    Path.AddArc(rect, 180, 180)
    Using p As Pen = New Pen(glowColor)
        Using lgb As New LinearGradientBrush(Path.GetBounds(),
Color.FromArgb(64, whiteColor), Color.FromArgb(84, whiteColor),
LinearGradientMode.Vertical)
            lgb.Blend = sigmaBlend
            g.FillPath(lgb, Path)
        End Using
    End Using
    g.DrawPath(p, Path)
End Using
End Using

Dim blender As Drawing2D.Blend = New Drawing2D.Blend()
blender.Positions = New Single() {0, 0.9, 1}
blender.Factors = New Single() {0, 1, 1}

Dim sweepRectangle As Rectangle = rect
sweepRectangle.Height -= (rect.Height / 2)
sweepRectangle.Inflate(-16 * scale.Width, -3 * scale.Height)
Using lgb As New Drawing2D.LinearGradientBrush(sweepRectangle,
WhiteColor, Color.Empty, 90, True)
    sweepRectangle.Height -= scale.Height / 2
    lgb.Blend = blender
    g.FillEllipse(lgb, sweepRectangle)
End Using

Using borderPen As New Pen(darkerColor, 2)
    Dim borderRect As Rectangle = rect
    g.DrawEllipse(borderPen, borderRect)
End Using

g.SmoothingMode = mode
End Sub

Private Sub SetNormalstate()
    DarkColor = Color.Blue
    LightColor = Color.Cyan
End Sub

Private Sub SetHoverstate()
    DarkColor = Color.Maroon
    LightColor = Color.LightYellow
End Sub

Private Sub SetPressedState()
    DarkColor = Color.DarkOrange
    LightColor = Color.Yellow
End Sub

#Region " Enum "

<Flags()> _
Private Enum ButtonState
    Normal = &H0
    Hot = &H1
    Pressed = &H2
    Disabled = &H4

```



```
[Default] = &H8
End Enum

#End Region

End Class
```

بالمرفقات ستجدون مثالا لكيفية استخدام الكلاس أعلاه وهو سيكون مثل أي باتون آخر ولكن بشكل مختلف قليلا

المثال في [رابط المرفقات](#) بنسخة الفيچوال أستوديو ٢٠٠٨
أتمنى أن تكون الأفكار التي تحدثنا عنها مفيدة للبعض منكم بشكل ما أعتقد أنكم يمكنكم تطوير الكلاس مثل رسم **Text**
أو رسم صورة ما داخل **Ribbon** أو إضافة الكثير من **Property** له ليتناسب مع ما احتياجاتكم

[GlowButtonExample.rar](#)

مقال: أفكار في الجرافكس الجزء الثالث

مقدمة :

في الأجزاء السابقة ناقشنا كيفية الاستفادة من **GDI+** بأكثر من أسلوب و في الجزء الثاني تحديدا ناقشنا كيفية رسم **Glow**
وفي هذه الجزء سنناقش كيفية رسم **Glow** بطريقة أخرى مختلفة.

الفكرة الثالثة:

رسم **Glow** بطريقة ثانية مختلفة.

أولا قبل مناقشة كيفية رسم **Glow** لنناقش بعض الأفكار البسيطة التي ستكون مفيدة بشكل عام لاحقا وسنعطي أمثلة علي ذلك.

من المؤكد أن المهتمين بعمليات الرسم يعلمون جيدا كيف يرسمون مستطيلا له حواف دائرية لذلك فإن الكود التالي ليس جديدا عليهم

المثال الأول : رسم مستطيل له حواف دائرية

كود:

```
Protected Overrides Sub OnPaint(ByVal e As System.Windows.Forms.PaintEventArgs)
    MyBase.OnPaint(e)

    ' المناسبة بالأبعاد مستطيل تعريف
    Dim rect As New Rectangle(10, 10, 100, 100)
    ' صفر يساو أو من أقل دائما يكود لا أن عل بشرط قطر نصف تعريف
    Dim raduis As Integer = 10
    ' الرسم مسار تعريف
    Dim path As New GraphicsPath

    ' المستطيل بداية ومكان المستطيل أبعاد عن تعبر التي المتغيرات تعريف بعض
    Dim x As Integer = rect.X
    Dim y As Integer = rect.Y
    Dim w As Integer = rect.Width
```

```

Dim h As Integer = rect.Height

' المستطيل يسار أعلى اقصى في قوس رسم
path.AddArc(x, y, raduis, raduis, 180.0F, 90.0F)
' المستطيل يمين أعلى اقصى في قوس رسم
path.AddArc((x + w) - raduis, y, raduis, raduis, 270.0F, 90.0F)
' المستطيل يمين أسفل اقصى في قوس رسم
path.AddArc((x + w) - raduis, ((y + h) - raduis), raduis, raduis, 0.0F,
90.0F)
' المستطيل يسار أسفل اقصى في قوس رسم
path.AddArc(x, ((y + h) - raduis), raduis, raduis, 90.0F, 90.0F)

' الرسم عن الناتجة التشوهات إخفاء
' تحديدا الرسم في التشوهات بعض ستلاحظ الكود من التالي السطر بإلغاء قمت لو
الدائرية الحواف عند
e.Graphics.SmoothingMode = SmoothingMode.AntiAlias

' التالية الطرق من بوحدة المسار نرسم النهاية في

' الأولى الطريقة
' واحد لون بإستخدام المسار رسم
Using sb As New SolidBrush(Color.Red)
    e.Graphics.FillPath(sb, path)
End Using

' الثانية الطريقة
' لونين بإستخدام المسار رسم
Using lgb As New Drawing2D.LinearGradientBrush(rect, Color.DarkRed,
Color.Red, 90, True)
    e.Graphics.FillPath(lgb, path)
End Using

' الثالثة الطريقة
' لون من أكثر بإستخدام المسار رسم
Dim pgb As New Drawing2D.PathGradientBrush(path)
pgb.SurroundColors = New Color() {Color.DarkCyan, Color.RoyalBlue,
Color.SteelBlue, Color.Cyan}
pgb.CenterColor = Color.SteelBlue
e.Graphics.FillPath(pgb, path)
path.Dispose()

End Sub

```

في واقع الأمر إن مقدرة المبرمج علي تحويل أجزاء من الكود إلى دوال **Functions** أو الي روتينات **Methods/Sub** يجعل الكود أسهل وأقصر وأيضا يمكن الاستفادة من مثل هذه الدوال أو الروتينات بشكل عام في أي برامج وبشكل لا يحتاج فيه المبرمج إلى إعادة صياغة وكتابة الكود أكثر من مرة لذلك من الأفضل تحويل الكود أعلاه إلى دالة **Function**

الكود التالي يوضح شكل الكود بعد تحويل جزء إلى الي دالة أو أكثر

وستلاحظون أنه قد تم وضع بعض الشروط لكي نتأكد دائما أن نصف القطر لا يقل عن أو يساوي صفرا

كود:

```

Public Class Form1

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        ' المناسبة بالأبعاد مستطيل تعريف

```

```

Dim rect As New Rectangle(10, 10, 100, 100)
' الرسم مسار تعريف
Dim path As GraphicsPath = GetRoundedPath(rect, 10)

' الرسم عن الناتجة التشوهات إخفاء
' الرسم في التشوهات بعض ستلاحظ الكود من التالي السطر بإلغاء قمت لو
' الدائرية الحواف عند وتحديد
e.Graphics.SmoothingMode = SmoothingMode.AntiAlias

' التالية الطرق من بوحدة المسار نرسم النهاية في

' الأولى الطريقة
' واحد لون بإستخدام المسار رسم
Using sb As New SolidBrush(Color.Red)
    e.Graphics.FillPath(sb, path)
End Using

' الثانية الطريقة
' لونين بإستخدام المسار رسم
Using lgb As New Drawing2D.LinearGradientBrush(rect, Color.DarkRed,
Color.Red, 90, True)
    e.Graphics.FillPath(lgb, path)
End Using

' الثالثة الطريقة
' لون من أكثر بإستخدام المسار رسم
Dim pgb As New Drawing2D.PathGradientBrush(path)
pgb.SurroundColors = New Color() {Color.DarkCyan, Color.RoyalBlue,
Color.SteelBlue, Color.Cyan}
pgb.CenterColor = Color.SteelBlue
e.Graphics.FillPath(pgb, path)
path.Dispose()

End Sub

Private Function GetRoundedPath(ByVal rect As Rectangle, _
                                ByVal TopLeftRaduis As Integer, _
                                ByVal TopRightRaduis As Integer, _
                                ByVal BottomLeftRaduis As Integer, _
                                ByVal BottomRightRaduis As Integer) _
As GraphicsPath

    If ((TopLeftRaduis = 0) OrElse (TopLeftRaduis < 0)) Then
        TopLeftRaduis = 1
    End If
    If ((TopRightRaduis = 0) OrElse (TopRightRaduis < 0)) Then
        TopRightRaduis = 1
    End If
    If ((BottomLeftRaduis = 0) OrElse (BottomLeftRaduis < 0)) Then
        BottomLeftRaduis = 1
    End If
    If ((BottomRightRaduis = 0) OrElse (BottomRightRaduis < 0)) Then
        BottomRightRaduis = 1
    End If

    Dim x As Integer = rect.X
    Dim y As Integer = rect.Y
    Dim width As Integer = rect.Width
    Dim height As Integer = rect.Height
    Dim path As New GraphicsPath

    path.AddArc(x, y, TopLeftRaduis, TopLeftRaduis, 180.0F, 90.0F)

    If (TopRightRaduis <> 1) Then

```

```

        path.AddArc((x + width) - TopRightRaduis), y, TopRightRaduis,
TopRightRaduis, 270.0F, 90.0F)
    Else
        path.AddLine((x + width), y, (x + width), width)
    End If

    If (BottomRightRaduis <> 1) Then
        path.AddArc((x + width) - BottomRightRaduis), ((y + height) -
BottomRightRaduis), BottomRightRaduis, BottomRightRaduis, 0.0F, 90.0F)
    Else
        path.AddLine(CInt((x + width)), CInt((y + height)), CInt((x +
width)), CInt((y + height)))
    End If

    path.AddArc(x, ((y + height) - BottomLeftRaduis), BottomLeftRaduis,
BottomLeftRaduis, 90.0F, 90.0F)
    path.CloseFigure()

    Return path

End Function

Private Function GetRoundedPath(ByVal rect As Rectangle, ByVal radius As
Integer) As GraphicsPath
    Return GetRoundedPath(rect, radius, radius, radius, radius)
End Function

End Class

```

المثال الثاني : استخدام الدالة **GetRoundedPath** لرسم المستطيل بحواف دائرية وبأكثر من أسلوب وفي الكود أدناه تم رسم المستطيل بحيث تكون حوافه اليسرى فقط لها شكل دائري

كود:

```

Public Class Form1

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        ' المناسبة بالأبعاد مستطيل تعريف
        Dim rect As New Rectangle(10, 10, 100, 100)
        ' الرسم مسار تعريف
        Dim path As GraphicsPath = GetRoundedPath(rect, rect.Width / 2, 0,
rect.Width / 2, 0)

        ' الرسم عن الناتجة التشوهات إخفاء
        ' الرسم في التشوهات بعض ستلاحظ الكود من التالي السطر بإلغاء قمت لو
الدائرية الحواف عند وتحديد
        e.Graphics.SmoothingMode = SmoothingMode.AntiAlias

        ' واحد لون بإستخدام المسار رسم
        Using sb As New SolidBrush(Color.Red)
            e.Graphics.FillPath(sb, path)
        End Using

    End Sub

End Class

```

يمكنك عزيزي القارئ أن تجرب تمرير قيم إلى الدالة بأن تجعل القيم جميعها صفرا أو تجعل بعضها صفرا ولاحظ الفرق في شكل المستطيل وبالتأكيد ليس شرطا أن تكون قيمة نصف القطر متساوية حيث يمكنك استخدام أرقام مختلفة لنصف القطر

المثال الثالث: رسم Glow

لكي نرسم Glow بشكل جيد نحتاج إلى تعريف الكثير من المتغيرات والمثال التالي يوضح ذلك ولقد قمت بتوضيح الهدف من كل جزء من الكود داخل المثال

كود:

```
Public Class Form1

    Protected Overrides Sub OnPaint(ByVal e As
System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        ' الرئيسي المستطيل عن يعبر متغير تعريف
        Dim rect As New Rectangle(20, 20, 100, 25)

        ' الرئيسي المستطيل مع جميعها وربطها المستطيلات بعض تعريف

        ' للمستطيل مساوية ابعاده تكون بحيث الأول المستطيل عن يعبر متغير تغير
        ' الرئيسي المستطيل ارتفاع نصف يساوي ارتفاعه يكون ان علي الرئيسي
        Dim rect1 As New Rectangle(rect.X, rect.Y, rect.Width, (rect.Height /
2))

        ' الأول المستطيل يساوي ما هو وغالبا الثاني المستطيل عن يعبر متغير تعريف
        ' الأول المستطيل ارتفاع عن ضئيل وبمقدار قليلا اكبر يكون ارتفاعه ولكن
        Dim rect2 As New Rectangle(rect.X, rect.Y, rect.Width, ((rect.Height /
2) + 1))

        ' أبعاده من جزء أن هنا وستلاحظون الثالث المستطيل عن يعبر متغير تعريف
        ' الرئيسي المستطيل نطاق خارج ستكون المستطيل
        Dim rect3 As New Rectangle((rect.X - 10), (rect.Y + (rect.Height / 2)),
(rect.Width + 20), rect.Height)

        ' من السفلي النصف في رسمه سيتم وهو الرابع المستطيل عن يعبر متغير تعريف
        ' الارتفاع في جدا ضئيل بمقدار سيزيد ولكنه الرئيسي المستطيل
        Dim rect4 As New Rectangle(rect.X, (rect.Y + (rect.Height / 2)),
rect.Width, ((rect.Height / 2) + 1))

        ' الدائرية للحواف القطر نصف تعريف
        Dim radius As Integer = 5

        ' المستطيلات نملاً لكي المستخدمة الألوان تعريف
        Dim colorLightLight As Color = Color.LightYellow
        Dim colorDarkDark As Color = Color.Orange

        Dim colorDark As Color = Color.OrangeRed
        Dim colorLight As Color = Color.Yellow

        ' تشوهات بدون سيحدث الرسم أن من التأكد
        Dim mode As SmoothingMode = e.Graphics.SmoothingMode
        e.Graphics.SmoothingMode = SmoothingMode.AntiAlias
        e.Graphics.SmoothingMode = SmoothingMode.HighQuality

        ' Lineargradient Brush عن يعبر متغير تعريف
```

```

        Using lgb As New Drawing2D.LinearGradientBrush(rect1, colorLightLight,
colorDarkDark, 90.0F, True)
        ' Lineargradient Brush عن يعبر آخر متغير تعريف
        Using lgb1 As New Drawing2D.LinearGradientBrush(rect1, colorDark,
colorLight, -90.0F, True)
        ' ابعاد من أضغر التدرج كان لو حالة في اللون تدرج شكل تحديد
المستطيل
        lgb1.WrapMode = WrapMode.TileFlipXY
        lgb.WrapMode = WrapMode.TileFlipXY
        ' الرئيسي المستطيل من العلوي النصف يمثل وهو الأول المسار تعريف
دائرية فقط اليسري حوافه نجعل وأيضا
        Dim path1 As Drawing2D.GraphicsPath = GetRoundedPath(rect2,
radius, radius, 0, 0)
        ' الأول المسار رسم
        e.Graphics.FillPath(lgb, path1)
        ' المستطيل من السفلي الجزء سيمثل وهو الثاني المسار تعريف
        Dim path2 As New GraphicsPath
        path2.AddRectangle(rect3)
        ' Pathgradient Brush عن يعبر متغير تعريف
        Dim pgb As New PathGradientBrush(path2)
        ' الفرشاة لهذه الألوان إضاف
        pgb.SurroundColors = New Color() {colorDark, colorDark,
colorLight, colorLight}
        ' سيعمل الذي وهو المسار منتصف في للرسم المستخدم اللون تحديد
اللامع اللون بمثابة
        pgb.CenterColor = colorLight
        ' الثاني المسار رسم
        e.Graphics.FillPath(pgb, GetRoundedPath(rect4, 0, 0, radius,
radius))

        End Using
    End Using

    ' للمستطيل الخارجية الحدود رسم

    Dim outerBorderColor As Color = Color.DarkOrange
    Dim innerBorderColor As Color = Color.Cornsilk

    ' المستطيل عن ضئيل بمقدا اقل ابعاده وستكون مستطيل عن يعبر متغير تعريف
الرئيسي
    Dim rect5 As New Rectangle((rect.X + 1), (rect.Y + 1), (rect.Width - 2),
(rect.Height - 2))

    ' الداخلية الأبعاد رسم
    Using innerBorderPen As Pen = New Pen(innerBorderColor)
        Using path As GraphicsPath = GetRoundedPath(rect5, radius)
            e.Graphics.DrawPath(innerBorderPen, path)
        End Using
    End Using

    ' الرئيسي المستطيل أبعاد رسم
    Using outerBorderPen As Pen = New Pen(outerBorderColor)
        Using path As GraphicsPath = GetRoundedPath(rect, radius)
            e.Graphics.DrawPath(outerBorderPen, path)
        End Using
    End Using

    e.Graphics.SmoothingMode = mode

End Sub

```

```

Private Function GetRoundedPath(ByVal rect As Rectangle, _
                                ByVal TopLeftRaduis As Integer, _
                                ByVal TopRightRaduis As Integer, _
                                ByVal BottomLeftRaduis As Integer, _
                                ByVal BottomRightRaduis As Integer)
As GraphicsPath

    If ((TopLeftRaduis = 0) OrElse (TopLeftRaduis < 0)) Then
        TopLeftRaduis = 1
    End If
    If ((TopRightRaduis = 0) OrElse (TopRightRaduis < 0)) Then
        TopRightRaduis = 1
    End If
    If ((BottomLeftRaduis = 0) OrElse (BottomLeftRaduis < 0)) Then
        BottomLeftRaduis = 1
    End If
    If ((BottomRightRaduis = 0) OrElse (BottomRightRaduis < 0)) Then
        BottomRightRaduis = 1
    End If

    Dim x As Integer = rect.X
    Dim y As Integer = rect.Y
    Dim width As Integer = rect.Width
    Dim height As Integer = rect.Height
    Dim path As New GraphicsPath

    path.AddArc(x, y, TopLeftRaduis, TopLeftRaduis, 180.0F, 90.0F)

    If (TopRightRaduis <> 1) Then
        path.AddArc((x + width) - TopRightRaduis, y, TopRightRaduis,
TopRightRaduis, 270.0F, 90.0F)
    Else
        path.AddLine((x + width), y, (x + width), width)
    End If

    If (BottomRightRaduis <> 1) Then
        path.AddArc((x + width) - BottomRightRaduis, ((y + height) -
BottomRightRaduis), BottomRightRaduis, BottomRightRaduis, 0.0F, 90.0F)
    Else
        path.AddLine(CInt((x + width)), CInt((y + height)), CInt((x +
width)), CInt((y + height)))
    End If

    path.AddArc(x, ((y + height) - BottomLeftRaduis), BottomLeftRaduis,
BottomLeftRaduis, 90.0F, 90.0F)
    path.CloseFigure()

    Return path

End Function

Private Function GetRoundedPath(ByVal rect As Rectangle, ByVal radius As
Integer) As GraphicsPath
    Return GetRoundedPath(rect, radius, radius, radius, radius)
End Function

End Class

```

في المشاركة التالية سنقوم بتحويل الكود أعلاه إلى دالة أو أكثر من دالة بحيث يسهل استخدامها ولا نحتاج إلى تكرار كتابة الكود مرارا وتكرارا وبالتالي يمكن الاستفادة منها بشكل أفضل

الكود التالي يوضح شكل الكود الموجود بالمثل الثالث وذلك بعد تحويل جزء من الكود إلى دوال **Functions** و روتينات **Method/Sub**

وستلاحظ عزيز القارئ أن الكود الموجود في الحدث **Paint** قد أصبح أقل بكثير عن سابقة في المثل الثالث وهنا تظهر أهمية تحويل أجزاء من الكود إلى دوال أو روتينان.

المثال الرابع :

كود:

```
Public Class Form1

    Dim colorLightLight As Color = ControlPaint.Light(Color.SteelBlue)
    Dim colorDarkDark As Color = (Color.Blue)
    Dim colorDark As Color = (Color.MediumBlue)
    Dim colorLight As Color = ControlPaint.Light(Color.Cyan)
    Dim colorFrameOut As Color = ControlPaint.Dark(Color.Navy)
    Dim colorFrameIn As Color = ControlPaint.Light(Color.LightSteelBlue)
    Dim rect As New Rectangle(20, 20, 100, 25)
    Dim rect1 As New Rectangle(20, 50, 100, 25)

    Protected Overrides Sub OnPaint(ByVal e As System.Windows.Forms.PaintEventArgs)
        MyBase.OnPaint(e)

        DrawGlow(e.Graphics, rect, colorLightLight, colorDarkDark, colorDark, colorLight, 5)
        DrawGlow(e.Graphics, rect1, colorLightLight, colorDarkDark, colorDark, colorLight, colorFrameOut, colorFrameIn, 5)

    End Sub

    Private Function GetBorderPath(ByVal rect As Rectangle, ByVal displacement As Single) As GraphicsPath

        rect.Width -= 1
        rect.Height -= 1
        Dim path As New GraphicsPath

        path.AddLine((rect.Left + displacement), CSng(rect.Top), (rect.Right - displacement), CSng(rect.Top))
        path.AddLine((rect.Right - displacement), CSng(rect.Top), CSng(rect.Right), (rect.Top + displacement))
        path.AddLine(CSng(rect.Right), (rect.Top + displacement), CSng(rect.Right), (rect.Bottom - displacement))
        path.AddLine(CSng(rect.Right), (rect.Bottom - displacement), (rect.Right - displacement), CSng(rect.Bottom))
        path.AddLine((rect.Right - displacement), CSng(rect.Bottom), (rect.Left + displacement), CSng(rect.Bottom))
        path.AddLine((rect.Left + displacement), CSng(rect.Bottom), CSng(rect.Left), (rect.Bottom - displacement))
        path.AddLine(CSng(rect.Left), (rect.Bottom - displacement), CSng(rect.Left), (rect.Top + displacement))
        path.AddLine(CSng(rect.Left), (rect.Top + displacement), (rect.Left + displacement), CSng(rect.Top))

        Return path

    End Function

    Private Function GetBorderPath(ByVal rect As Rectangle, ByVal exclude As
```



```

Rectangle, ByVal displacement As Single) As GraphicsPath

    If exclude.IsEmpty Then
        Return GetBorderPath(rect, displacement)
    End If

    rect.Width -= 1
    rect.Height -= 1

    Dim list As New List(Of PointF)

    Dim x As Single = rect.X
    Dim y As Single = rect.Y
    Dim right As Single = rect.Right
    Dim bottom As Single = rect.Bottom

    Dim pt1 As Single = (rect.X + displacement)
    Dim pt2 As Single = (rect.Right - displacement)
    Dim pt3 As Single = (rect.Y + displacement)
    Dim pt4 As Single = (rect.Bottom - displacement)
    Dim pt5 As Single = IIf((displacement = 0.0F), 1.0F, displacement)

    If ((rect.Y >= exclude.Top) AndAlso (rect.Y <= exclude.Bottom)) Then

        Dim pt6 As Single = ((exclude.X - 1) - displacement)
        Dim pt7 As Single = (exclude.Right + displacement)

        If (pt1 <= pt6) Then
            list.Add(New PointF(pt1, y))
            list.Add(New PointF(pt6, y))
            list.Add(New PointF((pt6 + displacement), (y - pt5)))
        Else
            pt6 = (exclude.X - 1)
            list.Add(New PointF(pt6, y))
            list.Add(New PointF(pt6, (y - pt5)))
        End If

        If (pt2 > pt7) Then
            list.Add(New PointF((pt7 - displacement), (y - pt5)))
            list.Add(New PointF(pt7, y))
            list.Add(New PointF(pt2, y))
        Else
            pt7 = exclude.Right
            list.Add(New PointF(pt7, (y - pt5)))
            list.Add(New PointF(pt7, y))
        End If

    Else
        list.Add(New PointF(pt1, y))
        list.Add(New PointF(pt2, y))
    End If

    list.Add(New PointF(right, pt3))
    list.Add(New PointF(right, pt4))
    list.Add(New PointF(pt2, bottom))
    list.Add(New PointF(pt1, bottom))
    list.Add(New PointF(x, pt4))
    list.Add(New PointF(x, pt3))

    Dim path As New GraphicsPath

    For i As Integer = 1 To list.Count - 1
        path.AddLine(list.Item((i - 1)), list.Item(i))
    Next

```

```

        path.AddLine(list.Item((list.Count - 1)), list.Item(0))
        Return path

    End Function

    Private Function GetBorderPath(ByVal rect As Rectangle) As GraphicsPath
        Return GetBorderPath(rect, 0)
    End Function

    Private Function GetBorderPath(ByVal rect As Rectangle, ByVal radius As
Integer) As GraphicsPath
        Dim path As New GraphicsPath
        If (radius = 0) Then
            path.AddRectangle(rect)
            Return path
        End If
        Return GetRoundedPath(rect, radius)
    End Function

    Private Function GetRoundedPath(ByVal rect As Rectangle, ByVal radius As
Integer) As GraphicsPath
        Return GetRoundedPath(rect, radius, radius, radius, radius)
    End Function

    Private Function GetRoundedPath(ByVal rect As Rectangle, _
                                     ByVal TopLeftRaduis As Integer, _
                                     ByVal TopRightRaduis As Integer, _
                                     ByVal BottomLeftRadius As Integer,
-
                                     ByVal BottomRightRadius As Integer)
As GraphicsPath

        If ((TopLeftRaduis = 0) OrElse (TopLeftRaduis < 0)) Then
            TopLeftRaduis = 1
        End If

        If ((TopRightRaduis = 0) OrElse (TopRightRaduis < 0)) Then
            TopRightRaduis = 1
        End If

        If ((BottomLeftRadius = 0) OrElse (BottomLeftRadius < 0)) Then
            BottomLeftRadius = 1
        End If

        If ((BottomRightRadius = 0) OrElse (BottomRightRadius < 0)) Then
            BottomRightRadius = 1
        End If

        Dim x As Integer = rect.X
        Dim y As Integer = rect.Y
        Dim width As Integer = rect.Width
        Dim height As Integer = rect.Height
        Dim path As New GraphicsPath

        path.AddArc(x, y, TopLeftRaduis, TopLeftRaduis, 180.0F, 90.0F)

        If (TopRightRaduis <> 1) Then
            path.AddArc((x + width) - TopRightRaduis, y, TopRightRaduis,
TopRightRaduis, 270.0F, 90.0F)
        Else
            path.AddLine((x + width), y, (x + width), width)
        End If
    End Function

```

```

        If (BottomRightRadius <> 1) Then
            path.AddArc((x + width) - BottomRightRadius, ((y + height) -
BottomRightRadius), BottomRightRadius, BottomRightRadius, 0.0F, 90.0F)
        Else
            path.AddLine(CInt((x + width)), CInt((y + height)), CInt((x +
width)), CInt((y + height)))
        End If

        path.AddArc(x, ((y + height) - BottomLeftRadius), BottomLeftRadius,
BottomLeftRadius, 90.0F, 90.0F)
        path.CloseFigure()

        Return path

    End Function

    Private Sub DrawBorder(ByVal g As Graphics, _
        ByVal rect As Rectangle, _
        ByVal borderColor As Color, _
        ByVal TopLeftRadius As Integer, _
        ByVal TopRightRadius As Integer, _
        ByVal BottomLeftRadius As Integer, _
        ByVal BottomRightRadius As Integer)

        Using borderPen As Pen = New Pen(borderColor)
            Using path As GraphicsPath = GetRoundedPath(rect, TopLeftRadius,
TopRightRadius, BottomLeftRadius, BottomRightRadius)
                g.DrawPath(borderPen, path)
            End Using
        End Using
    End Sub

    Private Sub DrawBorder(ByVal g As Graphics, ByVal rect As Rectangle, ByVal
borderColor As Color)
        DrawBorder(g, rect, borderColor, 0)
    End Sub

    Private Sub DrawBorder(ByVal g As Graphics, ByVal rect As Rectangle, ByVal
borderColor As Color, ByVal radius As Integer)

        Using borderPen As Pen = New Pen(borderColor)
            Using path As GraphicsPath = GetBorderPath(rect, radius)
                g.DrawPath(borderPen, path)
            End Using
        End Using

    End Sub

    Private Sub DrawGlow(ByVal g As Graphics, _
        ByVal rect As Rectangle, _
        ByVal colorLightLight As Color, _
        ByVal colorDarkDark As Color, _
        ByVal colorDark As Color, _
        ByVal colorLight As Color, _
        ByVal colorFrameOut As Color, _
        ByVal colorFrameIn As Color, _
        ByVal radius As Integer)

        Dim mode As SmoothingMode = g.SmoothingMode
        g.SmoothingMode = SmoothingMode.AntiAlias

        Dim r As New Rectangle(rect.X, rect.Y, rect.Width, (rect.Height / 2))

        Using lgb As LinearGradientBrush = New LinearGradientBrush(r,

```

```

colorLightLight, colorDarkDark, 90.0F, True)

    Using lgb1 As LinearGradientBrush = New LinearGradientBrush(r,
colorDark, colorLight, -90.0F, True)

        lgb1.WrapMode = WrapMode.TileFlipXY
        lgb.WrapMode = WrapMode.TileFlipXY

        Dim r1 As New Rectangle(rect.X, rect.Y, rect.Width,
((rect.Height / 2) + 1))
        g.FillPath(lgb, GetRoundedPath(r1, radius, radius, 0, 0))

        Dim path As New GraphicsPath

        Dim r2 As New Rectangle((rect.X - 10), (rect.Y + (rect.Height /
2)), (rect.Width + 20), rect.Height)
        Dim r3 As New Rectangle(rect.X, (rect.Y + (rect.Height / 2)),
rect.Width, ((rect.Height / 2) + 1))

        path.AddRectangle(r2)

        Dim pgb As New PathGradientBrush(path)
        pgb.SurroundColors = New Color() {colorDark, colorDark,
ControlPaint.Light(colorLight), ControlPaint.Light(colorLight)}
        pgb.CenterColor = ControlPaint.Light(colorLight)
        g.FillPath(pgb, GetRoundedPath(r3, 0, 0, radius, radius))
        path.Dispose()
    End Using
End Using

    Dim r4 As New Rectangle((rect.X + 1), (rect.Y + 1), (rect.Width - 2),
(rect.Height - 2))

    DrawBorder(g, r4, colorFrameIn, radius)
    DrawBorder(g, rect, colorFrameOut, radius)

    g.SmoothingMode = mode

End Sub

Private Sub DrawGlow(ByVal g As Graphics, _
                    ByVal rect As Rectangle, _
                    ByVal colorLightLight As Color, _
                    ByVal colorDarkDark As Color, _
                    ByVal colorDark As Color, _
                    ByVal colorLight As Color, _
                    ByVal radius As Integer)

    Dim mode As SmoothingMode = g.SmoothingMode
    g.SmoothingMode = SmoothingMode.AntiAlias
    Dim r As New Rectangle(rect.X, rect.Y, rect.Width, (rect.Height / 2))
    Using lgb As LinearGradientBrush = New LinearGradientBrush(r,
colorLightLight, colorDarkDark, 90.0F, True)

        Using lgb1 As LinearGradientBrush = New LinearGradientBrush(r,
colorDark, colorLight, -90.0F, True)

            lgb1.WrapMode = WrapMode.TileFlipXY
            lgb.WrapMode = WrapMode.TileFlipXY

            Dim r1 As New Rectangle(rect.X, rect.Y, rect.Width,
((rect.Height / 2) + 1))
            g.FillPath(lgb, GetRoundedPath(r1, radius, radius, 0, 0))

```

```

        Dim path As New GraphicsPath

        Dim r2 As New Rectangle((rect.X - 10), (rect.Y + (rect.Height /
2)), (rect.Width + 20), rect.Height)
        Dim r3 As New Rectangle(rect.X, (rect.Y + (rect.Height / 2)),
rect.Width, ((rect.Height / 2) + 1))

        path.AddRectangle(r2)

        Dim pgb As New PathGradientBrush(path)
        pgb.SurroundColors = New Color() {colorDark, colorDark,
ControlPaint.Light(colorLight), ControlPaint.Light(colorLight)}
        pgb.CenterColor = ControlPaint.Light(colorLight)
        g.FillPath(pgb, GetRoundedPath(r3, 0, 0, radius, radius))
        path.Dispose()
    End Using
End Using

    g.SmoothingMode = mode

End Sub

End Class

```

التطبيق :

في الكود الموجود بالمرفقات ستجدون فكرة عن كيفية استخدام الأفكار أعلاه في بناء باتون يختلف كثيرا عن الباتون العادي والموجود في الدوت نت وهو يختلف عن المثال الموجود في الجزء الثاني من سلسلة المقالات الخاصة بالجرافكس حيث في الجزء الثاني كان الباتون دائريا أما في هذا المثال سيكون الباتون علي شكل مستطيل له حواف دائرية إضافة إلى ذلك ستجدون فكرة مبسطة عن كيفية الاستفادة من نفس الأفكار أعلاه في بناء **Panel** بسيط ولكن بشكل مختلف

أتمنى أن يكون الموضوع مفيدا لكم بشكل ما

بالمرفقات ستجدون نسخة من الكود بنسخة الفيجوال ٢٠١٠ و أيضا بنسخة الفيجوال ٢٠٠٨

[GaphicsIdeas Part3 VS2008.rar](#)

[GaphicsIdeas Part3 VS2010.rar](#)

مقال : مقدمة إلى حماية البرامج

الكاتب : [silverlight](#)

[رابط المقال و النقاش حوله](#)

مقدمة:

غالبا وبعض قضائك أيام وليال في بناء برنامجك ثم وعند الانتهاء من كتابة اكواد البرنامج كاملا هنا فقط تبدأ في التفكير في كيفية حماية البرنامج من السرقة أو من النسخ بصورة غير شرعية

ولكن قبل أن تبدأ في كتابة أي اكواد لحماية برامجك عليك أن تتذكر شيئا هاما جدا ألا وهو أنه لا يوجد برنامج أمن من الكسر أو النسخ مائة بالمائة وأكبر دليل علي ذلك هو أن برامج مايكروسوفت نفسها مازالت معرضة للكسر أو النسخ بصورة غير شرعية لذلك لا تحزن عزيزي القارئ عندما يتم نسخ برامجك أو كسر حمايتها لأنه لو حدث هذا معك فذلك يعني أنك قمت ببناء برنامج رائع وبالتالي أصبح برنامجك محط اهتمام الهاكرز

عموما عزيزي القارئ ولكي تصل إلى مرحلة أن تحمي برامجك مائة بالمائة من النسخ أو من الكسر أو أن تجعل كل من يستخدمون برامجك يطلبون ترخيصا مسبقا منك قبل تشغيلها فمن الأفضل أن تسمح ما كتبت من اكواد أو أن تكتفي بأن تصنع برامج لاستخدامك الشخصي فقط وبهذا تستطيع حمايتها مليون بالمائة وهذا ما أفعله غالبا

وفي واقع الأمر إنني لا أهتم وبصفة شخصية بموضوع حماية البرامج من النسخ أو الكسر بصورة غير شرعية خاصة أنني لا أعمل في حقل البرمجة لذلك لم أحاول أن أتطرق إلى مثل هذا الموضوع قبل الآن ولكني لاحظت أن الكثيرين في المنتدى يهتمون بذلك وبالتأكيد هذا حق مشروع لأي مبرمج ومع أن الكثير من الزملاء الأفاضل بالمنتدى قد قاموا بتغطية الكثير من الموضوعات التي تناقش أو تتناول فكرة المقال فمازال الكثير من القراء يتساءلون عن كيفية حماية برامجهم عموما وبشكل عام فإن هذا المقال سيوضح وبشكل بسيط كيفية حماية البرامج وبأسلوب مبسط بقدر الإمكان.

ماذا تحتاج لكي تحمي برامجك:

أعتقد أنك لكي تحمي برامجك بشكل مناسب فأنت تحتاج إلى فهم ودراسة المسميات التالية:

1- تحتاج إلى أن تدرس وبغناية فائقة كيفية أن تقوم بتشفير وفك التشفير للبيانات لذلك أنت تحتاج إلى فهم كيفية التعامل مع الكلاسات الموجودة في [System.Security.Cryptography.dll](#)

2- تحتاج إلى تحديد أين ستقوم بتخزين **Serial** الخاص ببرنامجك أو أين ستقوم بحفظ أي معلومات أخرى هامة مطلوبة للتأكد من أن برنامجك قد تم شراؤه بشكل قانوني لذلك أنت تحتاج إلى فهم ودراسة كيفية التعامل مع **Registry** أو مع الملفات **Files**

3- تحتاج إلى فهم ودراسة **Dates & Timer Classes** لأن هذا هام جدا في تحديد الفترات الزمنية التي تمنحها لتجربة برنامجك

4- بالتأكيد ستحتاج إلى أن تكون ملما ببعض الأشياء الأخرى مثل التعامل مع **System.Text** أو **MemoryStream** أو

5- تحتاج إلى أن تتعلم كيفية بناء الكلاسات وتحويل الكلاسات إلى مكتبات **libraries** حتى تستطيع أن تستخدمها من داخل برنامجك أو من داخل **Setup** الخاص ببرنامجك

6- بالنهاية عليك أن تكون ملما بمشاكل نظام التشغيل **Operating System** الذي سوف يعمل عليه برنامجك وملما أيضا بجميع **Security** التي يحتاجها نظام التشغيل حتى يعمل برنامجك بالشكل المناسب.

الفكرة:

الفكرة التي يعتمد عليها الموضوع هي بسيطة جدا في مضمونها ويمكن تقسيمها إلى عدة مراحل كالآتي:

المرحلة الأولى:

بناء برنامج صغير تمرر به اسم المستخدم فيقوم بتشفير الاسم وإعطاء **Serial** يناسب فقط اسم المستخدم وهذا الجزء سوف تحتفظ أنت به ولا تعطيه للعميل وهذه المرحلة يمكن أن نطلق عليها **Key Generation**

المرحلة الثانية:

بناء برنامج صغير يقوم فيه مستخدم برنامجك بتمرير اسمه مع **Serial** الذي تفضلت بإعطائه له لتفعيل البرنامج وغالبا ما يكون مثل هذا الملف موجودا مع **Setup** الخاص ببرنامجك وهذه المرحلة يمكن أن نطلق عليها **Serial and User Validation**

المرحلة الثالثة:

بناء برنامج صغير يقوم بالتأكد وبصورة دائمة من أن البرنامج مرخص وبشكل قانوني لأي عميل وبالتالي يستمر البرنامج في العمل علي جهاز العميل بالشكل المطلوب أما في حالة لو أن العميل لا يمتلك **Serial** صحيح للبرنامج فهنا تظهر رسالة توضح له الفترة الزمنية المتبقية لتجربة البرنامج ومن ثم عدم تشغيل البرنامج بشكل نهائي بعد انتهاء فترة التجربة وهذه المرحلة يمكن أن نطلق عليها **Registration Checking** وغالبا ما تكون هذه المرحلة جزء من البرنامج الرئيسي

قبل أن نبدأ في كتابة الأكواد الخاصة بالمرحلة الثلاثة أعلاه علينا أن نكتب مجموعة من الكلاسات ولقد قمت بكتابة جميع الكلاسات علي شكل **library** باسم **RiverNile** وهي تتكون من الكلاسات كالآتي:

الكلاس الأول :

سنطلق عليه **RiverNileUtility** وهذا الكلاس ستجدون به جميع الدوال التي ربما نحتاج أن نستخدمها من خلال جميع الكلاسات الأخرى.

الكلاس الثاني:

هذا الكلاس سنطلق عليه **RiverNileData** وهو سيمثل أو سيعبر عن البيانات التي نحتاج بأن نعمل لها **Encrypt** و **Decrypt** لاحقا.

الكلاس الثالث:

هذا الكلاس سنطلق عليه اسم **RiverNileEncryptor** وهو سيكون مسئولاً بشكل رئيسي عن عمليات التشفير المختلفة أثناء تنفيذ البرنامج وهذا الكلاس بشكل عام يعتمد علي كلاس موجود بالفعل في الدوت نت وهو **RijndaelManaged Class** وهو مشتق أصلاً من **SymmetricAlgorithm Class** وهناك بعض الكلاسات المماثلة والتي هي مشتقة من **SymmetricAlgorithm Class** وهي علي التوالي **RC2CryptoServiceProvider** و **DESCryptoServiceProvider** و **TripleDESCryptoServiceProvider** وبذلك يمكنكم التعديل علي الكود الأصلي إن أردتم أن تستخدموا أي **Algorithm** أخر تجدونه مناسباً لكم أكثر لكي يقوم بالتشفير وفك التشفير

الكلاس الرابع:

هذا الكلاس سنطلق عليه **RiverNileDates** وهو عبارة عن **Structure** وهو سيمثل أو سيعبر عن التواريخ التي سنحتاج الرجوع إليها للتأكد من تاريخ تركيب البرنامج علي كمبيوتر العميل

لقد كتبت الكلاسات الأربعة علي شكل **library** حتى تكون فعالة بحيث يمكن استخدامها ليس فقط في حماية البرامج ولكن من الممكن أن تستفيدوا منها في تشفير أي بيانات داخل الملفات مثل ملفات الداتا بيز و خلفه

في رابط المرفقات ستجدون مثالا يوضح جميع الخطوات التي تمثل فكرة المقال وهي بنسخة الفيچوال ٢٠٠٨ أتمني أن تكون مفيدة لكم بشكل ما

كيف تستخدم المثال الموجود بالمرفقات:

قم بتشغيل المثال سيظهر لك الفورم الرئيسي وهو الهدف منه اختبار المراحل التي تحدثنا عنها أعلاه فقط لا غير

وفي هذا الفورم الرئيسي ستجد ثلاثة باتون وعند الضغط علي الباتون الأول سيظهر لك الفورم الخاص بالمرحلة الأولى وهو **KeyGeneration** قم بإدخال اسمك أو أي اسم آخر في **TextBox** الأول وهنا سيتم ظهور السيريرال المناسب لهذا الاسم أو توماتيكيا في **TextBox** الثاني

ثم قم بالضغط علي الزر الثاني بالفورم الرئيسي وهنا سيظهر لك الفورم الخاص بالمرحلة الثانية وهو **Validation** يمكنك الآن كتابة أو قص ولصق الاسم والسيريرال إلى هذا الفورم وهنا لو نفذت ذلك بدقة فستظهر لك رسالة تؤكد أنه قد تم حفظ السيريرال ولكن لو أدخلت البيانات بشكل خاطئ فستظهر لك رسالة تؤكد أنك يجب ان تشتري سيريرال للبرنامج ومن ثم يتم إغلاق **Validation Form** والبرنامج

ثم عليك الضغط علي الباتون الثالث وهنا ستظهر لك علي الفورم رسالة إما توضح أن البرنامج مرخص بشكل صحيح أو توضح الفترة الزمنية المتبقية لك لاستخدام البرنامج

عملية التسجيل الخاصة بالسيريرال واسم المستخدم بالإضافة إلى بعض البيانات الأخرى يتم تسجيلها في الريجستري الخاص بنظام التشغيل جرب تشغيل برنامج **RegEdit** ثم اذهب إلى **HKEY_CURRENT_USER\Software** ستجد أن البرنامج اسمه **RiverNile** ومسجل به بعض البيانات ولكنها **Decrypted**

أرجو أن تدرسوا الأكواد بشكل جيد وكما ترون سيكون من الصعوبة أن أقوم بتوضيح كل سطر في أي كلاس وعموما لقد كتبت فوق كل جزء من الكود توضيح بسيط للهدف من كل جزء بالكلاس .

أعتقد عزيزي القارئ أنك لو استطعت أن تتفهم المثال الموجود بالمرفقات بالشكل المناسب ومن ثم تبني كلاسات شبيهة به ومن ثم استطعت أن تدمجها مع برنامجك فربما تمكنت من أن تحمي برامجك أو علي الأقل تجعل برامجك تمتلك ما يكفي من الحماية التي ربما تجبر المستخدم علي أن يطلب منك شراء البرنامج وبشكل قانوني.

الملفات المرفقة : [SoftwareEvaluationMonitor.rar](#)

ال Delegates مع ال C#

الكاتب : **Mohammed Asaad**

[رابط المقال و النقاش حوله](#)

موضوع ال Delegates موضوع أكبر من هذا المقال وهذا جزء بسيط يوضح فكرة عن ماهيتها.

بالنسبة لمصطلح Delegates ففضلت إبقائه كما هو، سأحدث في هذا المقال عن ال Delegates إن شاء الله يمكن تعريف ال Delegates بأنها مؤشرات أو متغيرات تشير لطرق داخل الفئة الحالية أم في فئة أخرى وهي مشابهة للمؤشرات Pointers في لغة C أو C++ إلا أن هناك فروق فيما بينهما منها:

- أن ال Pointers تتعامل مع إجراءات ستاتيكية فقط (Static Procedures) بينما ال Delegates تتعامل مع إجراءات ستاتيكية أو إجراءات تابعة لفئات .
- ال Delegates أفضل من ناحية أنها تحصر نوعية الإجراءات بحيث تتوافق وسيطاتها - سيتم توضيح هذه النقطة لاحقاً

سيتم شرح النقاط التالية باستخدام مشروع Console ويتصرف ال Delegates كتصرف الإجراء الذي يشير إليه كما أنه يرجع قيمة كما يرجع الإجراء المشار إليه

استدعاء إجراءات بدون استخدام ال: Delegates

لنصنع فئة (Class) تحتوي على إجراء ما
كود:

```
class MyTestClass
{
    public void MyTestMethod
    {
        Console.WriteLine("Method without delegate");
    }
}
```

إذن قمنا بصنع فئة (MyTestClass) وتحتوي على الإجراء (MyTestMethod) والذي لا يرجع قيمة باستخدام الكلمة المحجوزة void

والآن لتنفيذ الإجراء نقوم بتعريف كائن جديد من MyTestClass وننفذ بواسطته الإجراء MyTest Method

كود:

```
{
MyTestClass TestClass = new MyTestClass();
TestClass.MyTestMethod();
}
```

والآن لننتقدم قليلا في موضوعنا هذا،

الشكل العام لل Delegates وخطواتها ومثال بسيط عنها:

كود:

```
Modifier [Returned Value Type] Identifier (Parameters);
```

- **Modifier**: محدد الوصول
- **Returned Value Type**: نوع القيمة المرجعة ويجب أن تتوافق مع الإجراء المشار إليه بهذا ال Delegate
- **Identifier**: اسم ال Delegate
- **Parameters**: الوسائط ويجب أيضا أن تتوافق مع وسيطات الإجراء المشار إليه بهذا ال Delegate

أما خطوات تعريف Delegate ما فهي كالتالي:

١. التصريح (Declaration)
٢. إنشاء نسخة جديدة منه (Instantiation)
٣. التنفيذ (Invocation)

كود:

```
namespace MyTestProject
{
    //التصريح ( Declaration)
    delegate bool IsLongString(string MyString);

    class Program
    {
        public static bool IsLongStr(string String)
        {
            bool Result;
            if (String.Length > 10)
            {
                Result = true;
                Console.Write(String + ": Is long string");
            }
            else
            {
                Result = false;
                Console.Write(String + ": Is not long string");
            }
            return Result;
        }

        static void Main(string[] args)
        {
            //إنشاء نسخة إنشاء)
        }
    }
}
```

```

        IsLongString TestDelegatedObject = new IsLongString(IsLongStr);
        //التنفيذ (Invocation)
        TestDelegatedObject(("Welcome from C# :));
        TestDelegatedObject(("Welcome"));
    }
}

```

لنراجع الكود السابق:
 قمنا في البداية بالتصريح عن **Delegate** كالتالي:

كود:

```

delegate bool IsLongString(string MyString);

```

وبعد ذلك قمنا بتصميم إجراء تتوافق وسيطاته وقيمتها المرجعة مع ال **delegate**

كود:

```

bool IsLongString(string MyString)

```

كود:

```

public static bool IsLongStr(string String)
{
    bool Result;
    if (String.Length > 10)
    {
        Result = true;
        Console.WriteLine(String + ": Is long string");
    }
    else
    {
        Result = false;
        Console.WriteLine(String + ": Is not long string");
    }
    return Result;
}

```

وبعد ذلك قمنا بإنشاء نسخة جديدة من ال **delegate IsLongString**

كود:

```

IsLongString TestDelegatedObject = new IsLongString(IsLongStr);

```

وفيما بعد قمنا بمرحلة التنفيذ
 كود:

```
TestDelegatedObject ("Welcome from C# :");
TestDelegatedObject ("Welcome");
```

علما بأن الطريقة

كود:

```
TestDelegatedObject ("Welcome");
```

توازي الطريقة

كود:

```
TestDelegatedObject.Invoke ("Welcome");
```

الدمج المتعدد: (MultiCasting)

علمنا أنه بالإمكان تفويض متغير للإشارة لإجراء ما، لكن هل يمكن تفويض لتنفيذ أكثر من إجراء؟ الإجابة: نعم يمكن، ليكن لدينا أكثر من إجراء وهذه الإجراءات لها نفس الوسيطات ونفس القيمة المرجعة وتنفيذ جميع هذه الإجراءات يتم بشكل متزامن، إذن يمكن اللجوء للـ **Delegates** للقيام بذلك عبر **MultiCasting** الآن سننبني أربع إجراءات لها المخرج نفسه والمدخل نفسه

كود:

```
static void Multi(double x, double y)
{
    Console.WriteLine(x * y);
}

static void Add(double x, double y)
{
    Console.WriteLine(x + y);
}
static void Devide(double x, double y)
{
    Console.WriteLine(x / y);
}
static void subtract(double x, double y)
{
    Console.WriteLine(x - y);
}
```

ونصرح الآن عن **Delegate** يتوافق مع الوسيطات والقيمة المرجعة:

كود:

```
delegate void Calculate(double x, double y);
```

لتنفيذ الإجراءات الأربعة السابقة نحتاج لأربع **Delegates**

كود:

```
Calculate DelMulti = new Calculate(Multi);
Calculate DelAdd = new Calculate(Add);
Calculate DelDevide = new Calculate(Devide);
Calculate Delsubtract = new Calculate(subtract);
```

الآن لتنفيذ ال **Delegates** السابقة يكون على الشكل:

كود:

```
DelMulti(4, 5);
DelAdd(4, 5);
DelDevide(4, 5);
Delsubtract(4, 5);
```

لكن يمكن دمج جميع ال **delegates** السابقة بواحد فقط عبر ربط ال **delegates** الأخرى معه وهذا ما يسمى

MultiCasting

ويتم ذلك باستعمال معاملات الربط والإزالة (+ و-)

كود:

```
Calculate AllCalculations = DelAdd + DelDevide+ DelMulti+Delsubtract;
AllCalculations(4,5);
```

ويمكن إزالة ارتباط **Delegates** تم ربطها بالمعامل -

كود:

```
AllCalculations=AllCalculations- ( DelDevide + DelMulti + Delsubtract);
```

تجدر الإشارة إلى أن ال **Delegates** تعتبر من الكائنات غير المتغيرة (**Immutable**)
بمعنى أنه عندما قلنا:

كود:

```
AllCalculations += DelAdd;
```

فإن ذلك لا يغير **AllCalculations** بل ينشئ نسخة جديدة لها

الأحداث:(Events) :

الأحداث هي أيضا مؤشرات مفوضة لإجراءات تقليدية تستدعى عند إطلاق هذا الحدث ويتم التصريح عنها باستخدام الكلمة
المحجوزة **event**

وخطوات إنشاء حدث وإطلاقه كالتالي:

١. الإعلان عن **delegate** ما .
٢. الإعلان عن حدث (**event**) مبني على هذا ال **delegate**.
٣. إطلاق الحدث .

مثال مبني على ما سبق:

كود:

```
public static event IsLongString CheckText; //Step 2 CheckText based on
IsLongString
```

وقبل إطلاق الحدث يجب عمل **Multicasting** للحدث حتى لا تقع في استثناء,

وكمثال لإطلاق الحدث فيتم كالتالي:

كود:

```
static void Main(string[] args)
{
    IsLongString TestDelegatedObject = new IsLongString(IsLongStr);
    CheckText += TestDelegatedObject;
    CheckText("Welcome from C# :)"); // The event is being
fired
}
```

الإجراءات المجهولة:(Anonymous Methods)

في **C# 1** كان الطريق الوحيد للتصريح عن **Delegates** هو استخدام إجراءات مسماة ، لكن ابتداء من **C# 2** وصعودا أصبح بالإمكان استخدام إجراءات مجهولة,(**Anonymous Methods**)

لنرى المثال التالي:

كود:

```
delegate bool IsLongString(string MyString);
static void Main(string[] args)
{
    IsLongString AnonMethod = delegate(string String)
    {
        bool Result;
        if (String.Length > 10)
        {
            Result = true;
            Console.Write(String + ": Is long string");
        }
        else
        {
            Result = false;
            Console.Write(String + ": Is not long string");
        }
    }
}
```

```
return Result;
};
AnonMethod("Welcome from c# :) ");
}
```

صرحنا عن **Delegates** وسميناه **IsLongString**
كود:

```
delegate bool IsLongString(string MyString);
```

بعد ذلك صرحنا عن نسخة من النوع **IsLongString** باسم **AnonMethod** نلاحظ أننا ربطنا إجراء غير مسمى بال **AnonMethod** وتم تنفيذه بشكل عادي.

Lambda Expressions

يمكن القول باختصار أن ال **Lambda Expressions** شبيهة بال **Anonymous Methods** والمثال التالي يوضح استخدام بسيط لها:
كود:

```
delegate bool IsLongString(string MyString);
static void Main(string[] args)
{
    IsLongString AnonMethod = String =>
    {
        bool Result;
        if (String.Length > 10)
        {
            Result = true;
            Console.Write(String + ": Is long string");
        }
        else
        {
            Result = false;
            Console.Write(String + ": Is not long string");
        }
        return Result;
    };
    AnonMethod("Welcome from c# :) ");
}
```


فهم كيفية بناء شبكة عصبية صناعية للتعرف على حرف أو رقم..

الكاتب : [رجب النتشة](#)
[رابط المقال و النقاش حوله](#)

مقدمة :

هذا الموضوع عبارة عن شرح لمثال بسيط يسهل عملية فهم كيفية بناء شبكة عصبية صناعية مكونة من خلية عصبية واحدة في المخرجات تقوم بالتعرف على الحرف أو الرمز المعلم في مرحلة التعليم مع ملاحظة أن عملية التعليم للشبكة العصبية تكون يدوية...
وتعطينا نتيجة أن الحرف هو صحيح أو أنه خطأ ...

الأهداف :

- 1- تعلم منطق العمل مع الشبكات العصبية وتنفيذها برمجياً.
- 2- تعلم مبادئ التعليم والفحص للشبكة العصبية.
- 3- التسلسل في بناء شبكة عصبية تلائم مسألة ما.
- 4- فهم أهمية خوارزميات التعليم بعد أن نقوم بتعليم الشبكة يدوياً.

تحمية

أنظر إلى الصورة التالية...

الصورة غير متوفرة

من هذه الشاشة يمكن أن نرى أنه تم إدخال قيمة ٠ في النص الخاص بالـ **Threshold** وثمانية سالبة -٨ في النص الخاص بالانحراف **Bise** و بعد ذلك تم إدخال اسم للحرف أو الرمز الذي علمناه وهو **One** بعد ذلك تم إدخال قيمة ١ في اللوحة

المخصصة لإدخال قيم الأوزان (وهي اللوحة البيضاء) فترى أن اللوحة تحتوي على أصفار ما عدا المنطقة التي نريد أن نعلمها لتستقبل الواحد..

بعد ذلك ننتقل إلى مرحلة الفحص **Testing** فنقوم برسم الرمز أو الحرف باللون الأزرق في اللوحة الخضراء ، بعد ذلك تكبس على كبسة **Is it a one** فيظهر لنا رسالة تقول لنا **Yes, it is a One** في حالة أن شكل الحرف يشبه شكل الحرف في التعليم ... فإذا لم يكن ذلك يقول البرنامج لنا **No , it not a One**

الفرضية...

لنفترض أننا نريد أن نعلم الشبكة كي تتعرف على الرقم واحد ١ ..
فتريد أن نتعرف على الرقم ١ من خلال رسمه مكونة من ٧ في ٧ ...
كما في الشكل ١ ...

الصورة غير متوفرة

الشكل ١ : احتمالين لرسم الرقم واحد في مصفوفة مكونة من ٧ في ٧

فهنا يكون عدد المدخلات إلى الشبكة هو ٤٩ أي إذا ضربنا ٧*٧ يكون الناتج ٤٩ ...

بما أن الشبكة العصبية هي عبارة عن نظام مكون من عدد من الخلايا العصبية

نريد أن نقوم ببناء شبكة عصبية صناعية مكونة من ٩ ٤ خلية عصبية للمدخلات و خلية عصبية واحدة على المخرجات ...
ف نقوم بإدخال المدخلات إلى الخلايا العصبية ال ٩ ٤ و نقوم بعمل المطلوب كي يكون المخرج للخلية العصبية واحد إذا كان الشكل المرسوم هو عبارة عن واحد أما إذا كان غير ذلك فإنه يتم إخراج صفر ، وبالنظر إلى الشكل ٢ ترى التخييل للشبكة بالرسم.. مع المعادلات...

الصورة غير متوفرة

الشكل ٢ : الشبكة العصبية الصناعية التي تحوي ٩ ٤ خلية عصبية للمدخلات **X** و خلية عصبية **Y** للإخراج أما ال **W** فهي لتخزين الأوزان

الفئة الممثلة للشبكة Class ANN

ما نحتاجه لبناء تمثيل للشبكة العصبية الصناعية هو بناء فئة **Class** ممثلة للشبكة .. يمكن أن نتخيل التمثيل بلغة ال **UML** لها كما في الشكل ٣ ..

الصورة غير متوفرة

الشكل ٣ : التمثيل للشبكة **ANN** باستخدام لغة **UML** المحتاج بنائها لإتمام العمل..

سوف أضع هنا تمثيل لها بلغة سي شارب...
أنظر إلى الكود التالي...

كود:

```
/*      Name           : Rajab Natshah
 *      E-mail          : rajab_n@yahoo.com
 *      Website         : www.rajab.natshah.com
 *      Date            : 22/12/2005
 */

using System;

namespace OneCharNetwork
{
    /// <summary>
    /// Artificial Neural Network
    /// </summary>
    public class ANN
    {
        public ANN(int N)
        {
            this.N = N;
            X = new double[N];
            W = new double[N];
        }

        /// <summary>
        /// The Size N of the input Neurons
        /// </summary>
        public int N;
    }
}
```

```

// the Input matrix of the network
public double[] X;

// the Wight matrix of the network
public double[] W;

// the bise
public double Bise;

// the threshold
public double threshold;

// the output
public double Y;

/// <summary>
/// the y_in of the Network
/// </summary>
public double y_in()
{
    // the summation of multiply
    double sum = 0;

    for (int i = 0; i < N; i++)
        sum += X[i] * W[i];

    // y_in = Bise + sum

    sum += Bise;

    return sum;
}

/// <summary>
/// the Activation Function
/// </summary>
public bool f(double y_in)
{
    if (y_in >= threshold)
    {
        Y = 1;
        return true;
    }
    else
    {
        Y = 0;
        return false;
    }
}

}
}

```

ونقوم بتهيئة البيئة البرمجة المكونة من مرحلتين...

مرحلة التعليم Learning أو.. Training

في مرحلة التعليم نحتاج أن نحدد حجم الشبكة ... أي قيمة.. **N** وهي هنا ٩٤ لأننا نستخدم لوحة لرسم الحرف مكونة من ٧ في ٧ ...
فما نحتاجه في مرحلة التعليم هو أن نقوم بتحديد قيمة الأوزان **Weights** وهو ما نرمز له بمصفوفة **W** ..
ونريد أن نحدد قيمة العتبة **Threshold** وقيمة الانحراف **Bise** ...
فذلك تم تصميم المرحلة الأولى في البرنامج كما في الشكل ٤ ..
الصورة غير متوفرة

الشكل ٤ : مرحلة التعليم في البرنامج التنفيذي

هنا عندما نقوم بتحديد قيمة الانحراف **Bise** بقيمة محده تمثل الانحراف للحرف في الشبكة ..
والمربعات البيضاء تمثل المحتوى في الأوزان **W** فلو قمنا مثلاً بإدخال قيم للأوزان وقيمة العتبة **Threshold** صفر
وقيمة الانحراف **Bise** بثمانية سالبة -٨ كما في الشكل ٥ .
الصورة غير متوفرة

الشكل ٥ : التمثيل للعملية نظرياً.

و لإتمام ذلك نقوم بإدخال القيم في المنطقة المخصصة لها ... كما في الشكل ٦
الصورة غير متوفرة

الشكل ٦ : تعليم الشبكة لكي تتعرف على الرقم واحد

مرحلة الفحص... Testing

بالنظر إلى المنطقة المخصصة لمرحلة الفحص .. نقوم بإدخال 0 في المنطقة التي لا نريد أن نرسم الحرف
ونضع واحد 1 في المنطقة التي تمثل رسمة الحرف ... انظر الشكل ٧ .
الصورة غير متوفرة

الشكل ٧ : عملية الفحص للشبكة..

يمكن أن نسأل هنا نعم ممكن أن أفهم أي أدخل واحد في المنطقة الممثلة للرسم للحرف الذي نريد أن نعلمه
ولكن لماذا وضعنا قيمة -٨- لتحديد قيمة الانحراف **Bise** ...؟

هنا نقول لو قمت بعمل جمع لقيم الواحدات في اللوحة الممثلة للأوزان .. فإنك تجد أنها تساوي ثمانية ..
هنا نقول أAAAAAAAAAAAAهنا هو الهدف من الانحراف **Bise** يعني نقوم بعمل ضرب للمدخلات مع الأوزان
فإذا كان مجموعهم ثمانية نقوم بجمع قيمة الانحراف **Bise** التي هي -٨- مع المجموع فيكون الناتج صفر
وقيمة صفر تساوي الـ **Threshold** فنقوم بإرجاع واحد أي أن الحرف هو واحد ... يعني ما يتم هو كما في المعادلة التالية
...

الصورة غير متوفرة

لو نظرنا إلى عملية المجموع نجد أنها ترجع ٨ وذلك من خلال عملية الضرب
للمدخلات **X** مع الأوزان أنظر الشكل ٨ .

الصورة غير متوفرة

الشكل ٨ : عملية ضرب المخلات X بالقيمة المقابلة لها في الأوزان W وجمعها مع بعضا..

الصورة غير متوفرة

لقد تم تنفيذ هذه الخطوة باستخدام هذا الكود...

كود:

```
/// <summary>
/// the y_in of the Network
/// </summary>
public double y_in()
{
    // the summation of multiply
    double sum = 0;

    for(int i=0;i<N;i++)
        sum += X[i] * W[i];

    // y_in = Bise + sum

    sum += Bise ;

    return sum;
}
```

حسننا الآن قيمة مجموع المضاريب مجموعة مع قيمة الانحراف **Bise** تساوي صفر أي أن قيمة المدخلات للشبكة **y_in** تساوي صفر..

نقوم بعدها استدعاء الاقتران الممثل لتنشيط الخلية .. وهو **F** كما في تنفيذنا فإذا كان قيمة **y_in** أكبر أو يساوي قيمة ال **Threshold** وهي صفر

فإن القيمة الراجعة تكون واحد أي نعم لقد تم التعرف على الحرف وهو صحيح ... أما إذا كانت قيمة ال **y_in** أقل من صفر نقوم بإرجاع صفر وهذا يعني

رفض الشبكة له و أنها لم تتعرف على الشكل المدخل .. والتمثيل الرياضي للعملية هو..

الصورة غير متوفرة

والتمثيل الذي قمت ببنائه له كما في الكود التالي ...

كود:

```
/// <summary>
/// the Activation Function
/// </summary>
public bool f(double y_in)
{
    if(y_in >= threshold)
    {
        Y = 1;
        return true;
    }
    else
    {
        Y = 0;
        return false;
    }
}
```

تطوير وتحسين...

لقد قمنا بعمل تعليم للشبكة وعمل فحص لها ...
 لكن ماذا لو قمنا بتغيير طريقة التعليم هل من الممكن أن تتغير النتائج ...
 نقول نعم ذلك الهدف من خوارزمية التعليم ...
 تخيل أننا علمنا الشبكة بطريقة أن تتقبل الرقم واحد بأشكال متعددة ...
 ولا تقبل أي شوائب خارجية ..
 فأقوم باقتراح القيم التالية للتعليم اليدوي والممل .. كما في الشكل ٩

الصورة غير متوفرة

الشكل ١٠ : التعرف على الرقم واحد الأصلي...

الصورة غير متوفرة

الشكل ١١ : التعرف على الرقم واحد مع بعض النواقص..

الصورة غير متوفرة

الشكل ١٢ : التعرف على الرقم واحد مع نقصان من الأعلى..

الصورة غير متوفرة

الشكل ١٣ : التعرف على الرقم واحد مع نقصان من الأسفل.

الشكل ١٤ : رفض الشكل المدخل لأنه لا يشبه الواحد..

ما أريده عند هذه النقطة أن يظهر السؤال التالي...

كيف أحدد قيمة الأوزان W وال Threshold وال خاص في شبكتي ؟

والجواب هو ... أنت على الطريق الصحيح ...
 هذا هو هدفنا من تعليم الشبكة العصبية أوتوماتيكيا ..
 أي أننا نقوم بعمل برنامج يقوم بالبحث عن أفضل القيم الخاصة بتعليم شبكتي
 التعليم الصحيح لكي تعطيني نتائج صحيحة ...
 وهذا هو العمل في علم الشبكات العصبية الصناعية ...

نقوم بتحديد مسألة ما..

بعدها نقوم بتصميم شبكة عصبية مناسبة للمسألة ...

ثم نقوم بتصميم خوارزمية أوتوماتيكية تقوم بتعليم الشبكة العصبية ...

ثم نقوم بفحص الشبكة هل هي تقوم بالعمل الصحيح أم لا ...

بعد ذلك نقوم باستخدامها في التطبيقات الحقيقية ...

وذلك بحفظ قيم الأوزان W وال Bise وال Threshold

الخلاصة:

العمل مع الشبكات العصبية ممتع وله صعوبات معينة ...

وفي هذا الموضوع لقد قمنا بتعلم منطق العمل مع الشبكات العصبية وتنفيذها برمجياً، ولو بشكل بسيطة وهدف ومسألة بسيطة

وتعلمنا مبدأ التعليم والفحص للشبكة العصبية. والتسلسل في بناء شبكة عصبية في برنامج تطبيقي، و أكبر هدف توصلنا له هو فهم أهمية خوارزمية التعليم بعد أن نقوم بتعليم الشبكة يدوياً.

أرجو أن يكون الدرس مفيداً لمن يريد أن يفهم و يطبق في علم الشبكات العصبية...

أضع لكم المشروع في الملف المرفق .. وهو مشروع من نوع سي شارب

[OneCharNetwork.zip](#)

هذا نفس المثال ولكن باستخدام لغة فجوال بيسك ..
هذا كان أول مثال أقدمه لأبي العلمي الدكتور حمزة عباس السوادي

[ANN 1.0.zip](#)

درس بالصور ::: لكيفية التعامل مع الجداول المترابطة في قواعد البيانات

الكاتب : lo2i

[رابط المقال و النقاش حوله](#)

درسنا اليوم سنتحدث قليلاً عن كيفية برمجة برنامج مرتبط بقاعدة بيانات ذات جداول مترابطة فيما بينها.

المثال التالي يحتوي على التالي ..

- كيفية التعامل مع قواعد بيانات اكسيس
- الاتصال بقاعدة البيانات
- التعامل مع ال **TransAction**
- التعامل مع الوضع المنفصل بقاعدة البيانات
- طريقة استخدام ال **moudle** لتعريف سلسلة الاتصال بقاعدة البيانات

أولاً : كيفية التعامل مع قواعد بيانات اكسيس:

نقوم بفتح برنامج الاكسيس لإنشاء قاعدة البيانات.
ونقوم بإنشاء الجدول الأول وهو بإسم (**Students**) وسوف يكون به الحقول التالية :

id وهو من نوع number
name وهو من نوع text
age وهو من نوع text
Class وهو من نوع text

كما توضح الصورة التالية..

Field Name	Data Type
id	Number
name	Text
age	Text
class	Text

ننتقل الآن للسجل الثاني..

وسيكون بإسم (**informations**) وسوف يكون به الحقول التالية :

id وهو من نوع number
tell وهو من نوع text
fax وهو من نوع text
mobile وهو من نوع text
email وهو من نوع text

كما توضحه الصورة التالية :

Field Name	Data Type
id	Number
tell	Text
fax	Text
mobile	Text
email	Text

بعد أن قمنا بإنشاء الجداول السابقة ننتقل الآن لنعمل على ال (Relationships) العلاقات بين الجداول لنقوم بربط الجدول الأول بالجدول الثاني من خلال الحقل ID لدى الجدولين.

كما هو موضح في الصورة التالية..

The screenshot shows the Microsoft Access 2007 ribbon with the 'Relationships' task pane open. The task pane has a red border and contains the following text:

Information about data
C:\Users\lo2i\Desktop\related tables\related tables\related tables\bin\Debug\data.mdb

Prepare Database for Web
Publish your application to a server
This will lock down your tables to allow only web-legal schema. Web-visible objects can then be added to your database and you database can be published to a web site. This operation cannot be reversed.

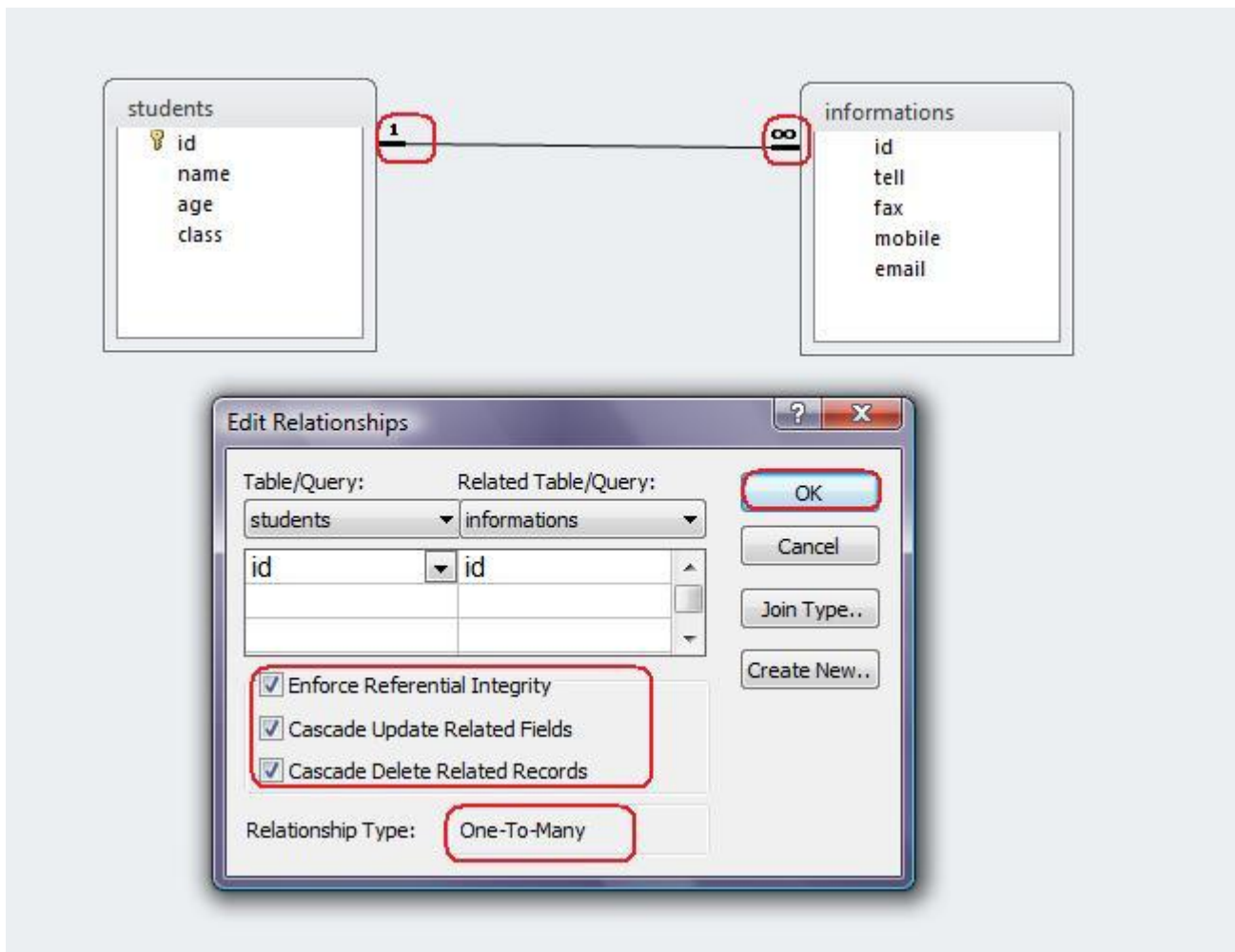
Relationships
Want to define how the data in tables is related?
Define how the data in tables is related, such as ID fields or name fields in different tables that should match.

نقوم بفتح العلاقات من خلال برنامج Microsoft access 2010 و Microsoft access 2007

نقوم بإضافة الجدولين لعمل العلاقة بينهم..

ونقوم بسحب ال id من خلال الجدول students إلى ال ID المرتبط بالجدول informations

فتظهر لنا الشاشة التالية..



قم بعمل نفس الظاهر لك في الصورة كي يعمل البرنامج معك بطريقة صحيحة..

ولاحظ أن نوع العلاقة بين الجدولين **one to many**

ملاحظة :- عند إعطائك خطأ في عملية الربط هذا يدل على أنك أخطأت في بناء الجدول من خلال أنواع الحقول..

بعد أن تقوم بعملية الربط كما هو موضح في الصورة السابقة تقوم بحفظ قاعدة البيانات.

ومن ثم تقوم بفتح الجدول الرئيسي المسمى بـ (**Students**) وقم سجل جديد عليه

ستلاحظ أنه عند اكتمال تعبئة الجدول سوف يقوم بإعطاء الخيار لك بتعبئة الجدول الثاني المسمى بـ (**informations**) هو موضح في الصورة التالية..

students				
id	name	age	class	Add New Field
1	lo2i	20	3rc	
	tell	fax	mobile	email
1111111	2222222	3333333		vb@c#.asp
*				
*				

كما هو مبين لك في الصورة السابقة أن أصبح الحقل ID هو الرابط بين الجدولين..

ننتقل الآن للعمل على الvisual studio 2008

نقوم بفتح البرنامج وإضافة التالي:

- شاشة أب (حاضنة) Mdiparent
- Form1
- Form2
- Moudle

نفتح ال **moudle** ونكتب به الكود التالي..

كود:

```
Public con As New OleDbConnection("provider=microsoft.jet.oledb.4.0; data source =" & Application.StartupPath & "\data.mdb")
```

بعد استدعاء فضاء الأسماء التالي :

كود:

```
Imports System.Data.OleDb
```

ننتقل الآن إلى الفورم الأول:

The screenshot shows a Windows form titled "الإضافة" (Add) with a standard Windows window border. The form contains eight text input fields stacked vertically, each with a label to its right. The labels are: "رقم الهوية" (ID Number), "الاسم" (Name), "العمر" (Age), "الصف" (Class), "رقم الهاتف" (Phone Number), "رقم الفاكس" (Fax Number), "رقم الجوال" (Mobile Number), and "البريد الإلكتروني" (Email). At the bottom center of the form is a button labeled "إضافة" (Add).

ونقوم بكتابة هذا الكود في الزر للإضافة لقاعدة البيانات

كود:

```

If Trim(TextBox1.Text) = "" Or _
Trim(TextBox2.Text) = "" Or _
Trim(TextBox3.Text) = "" Or _
Trim(TextBox4.Text) = "" Or _
Trim(TextBox5.Text) = "" Or _
Trim(TextBox6.Text) = "" Or _
Trim(TextBox7.Text) = "" Then
    MsgBox("الفارغة الخانات بملاً قم", MsgBoxStyle.Critical)
    Exit Sub
End If

Try
Dim dr As New OleDbDataAdapter("select * from students where id=" &
Trim(TextBox1.Text), con)
Dim dt1 As New DataTable
' =====
Dim cmd As New OleDbCommand("insert into students(id,name,age,class) values
(?,?,?,?)", con)
    cmd.Parameters.Clear()
    cmd.Parameters.AddWithValue("@id", OleDbType.Integer).Value =
Trim(TextBox1.Text)
    cmd.Parameters.AddWithValue("@name", OleDbType.VarChar).Value =
Trim(TextBox2.Text)
    cmd.Parameters.AddWithValue("@age", OleDbType.VarChar).Value =
Trim(TextBox3.Text)
    cmd.Parameters.AddWithValue("@class", OleDbType.VarChar).Value =
Trim(TextBox4.Text)
' =====

    If con.State = ConnectionState.Open Then
        con.Close()
    End If
    con.Open()
    dr.Fill(dt1)
    If dt1.Rows.Count <> 0 Then
        MsgBox("البيانات قاعدة في مكرر الهوية رقم",
MsgBoxStyle.Exclamation)
        TextBox1.SelectAll()
        TextBox1.Focus()
        Exit Sub
    End If
    tran = con.BeginTransaction
    cmd.Transaction = tran
    cmd.ExecuteNonQuery()
' =====
Dim cmd1 As New OleDbCommand("insert into informations(id,tell,fax,mobile,email)
values (?,?,?,?,?)", con)
    cmd1.Parameters.Clear()
    cmd1.Parameters.AddWithValue("@id", OleDbType.Integer).Value =
Trim(TextBox1.Text)
    cmd1.Parameters.AddWithValue("@tell", OleDbType.VarChar).Value =
Trim(TextBox5.Text)
    cmd1.Parameters.AddWithValue("@fax", OleDbType.VarChar).Value =
Trim(TextBox6.Text)
    cmd1.Parameters.AddWithValue("@mobile", OleDbType.VarChar).Value =
Trim(TextBox7.Text)
    cmd1.Parameters.AddWithValue("@email", OleDbType.VarChar).Value =
Trim(TextBox8.Text)
' =====
    cmd1.Transaction = tran
    cmd1.ExecuteNonQuery()
    tran.Commit()
    MsgBox("بنجاح البيانات بقاعدة الحفظ تم", MsgBoxStyle.Information)
    Me.Dispose()

```

```
Catch ex As Exception
    MsgBox("الحفظ عملية أثناء خطأ حدث", MsgBoxStyle.Critical)
    tran.Rollback()
Finally
    con.Close()
End Try
```

كما يجب عليك أن تعرف الكائن التالي
كود:

```
Public tran As OleDbTransaction
```

وهو إشارة إلى الكائن **transaction**

وطبعاً يجب عليك استدعاء
كود:

```
Imports System.Data.OleDb
```

في أعلى شاشة الكود

ثم تضغط على ال **textbox1** الذي سوف يدخل فيه المستخدم (رقم الهوية) لو تذكرت معي قليلاً إنا جعلنا حقل ال **id** في جدول ال **students** من نوع رقم..

فماذا هنا لو قام المستخدم بإدخال أحرف في الحقل ،، طبعاً سوف يحصل لك خطأ في البرنامج

لهذه سوف نجعل صندوق النص لا يقبل إلا أرقام فقط

كود:

```
Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
    Dim c As Char
    c = e.KeyChar
    If Not (Char.IsNumber(c)) Then
        e.Handled = True
    End If
End Sub
```

ننتقل الآن إلى ال Form2

رقم الهوية	الاسم	العمر	الصف	الهاتف	الفاكس	الجوال	البريد
1	lo2i	20	3rc	1111111	2222222	3333333	vb@c#.asp

البحث برقم الهوية

البحث بالاسم

عدد السجلات الناتجة عن البحث = 1

البحث

كل البيانات

بداية نقوم بتعريف فضاء الأسماء التي عرفناها في الفورم السابق

ومن ثم نقوم بإضافة الإجراء التالي :

كود:

```

Sub search(ByVal sql As String)
    Try
        Dim data As New OleDbDataAdapter(sql, con)
        If con.State = ConnectionState.Open Then
            con.Close()
        End If
        ds.Clear()
        con.Open()
        data.Fill(ds, "sss")
        con.Close()
        With DataGridView1
            .DataSource = ds.Tables("sss")
            .Columns(0).HeaderText = "الهوية رقم"
            .Columns(1).HeaderText = "الاسم"
            .Columns(2).HeaderText = "العمر"
            .Columns(3).HeaderText = "الصف"
            .Columns(4).HeaderText = "الهاتف"
            .Columns(5).HeaderText = "الفاكس"
            .Columns(6).HeaderText = "الجوال"
            .Columns(7).HeaderText = "البريد"
            .RowsDefaultCellStyle.BackColor = Color.Bisque
            .AlternatingRowsDefaultCellStyle.BackColor = Color.Beige
        End With
        Label1.Text = String.Format("البحث عن الناتجة السجلات عدد = {0}",

```

```
ds.Tables("sss").Rows.Count)
Catch ex As Exception
MsgBox("الاستعلام تنفيذ على قادر غير")
Finally
con.Close()
End Try
```

وفي الحدث لود نقوم بكتابة الكود التالي :

كود:

```
search("select
students.id,students.name,students.age,students.class,informat
ions.fax,informations.mobile,informations.email from students,informations where
(students.id=informations.id)")
```

وفي زر (كل البيانات نقوم بكتابة هذا الكود)

كود:

```
search("select
students.id,students.name,students.age,students.class,informations.tell,informat
ions.fax,informations.mobile,informations.email from students,informations where
(students.id=informations.id)")
```

وفي زر البحث نقوم بكتابة هذا الكود

كود:

```
If RadioButton1.Checked = True Then
If TextBox1.Text = "" Then
MsgBox("الفراغ ملأ يجب", MsgBoxStyle.Information)
TextBox1.Focus()
Else
search("select
students.id,students.name,students.age,students.class,informations.tell,informat
ions.fax,informations.mobile,informations.email from students,informations where
(students.id=informations.id) and (students.id=" & Trim(TextBox1.Text) & ")")
End If
End If
If RadioButton2.Checked = True Then
If TextBox2.Text = "" Then
MsgBox("الفراغ ملأ يجب", MsgBoxStyle.Information)
TextBox2.Focus()
Else
search("select
students.id,students.name,students.age,students.class,informations.tell,informat
ions.fax,informations.mobile,informations.email from students,informations where
(students.id=informations.id) and (students.name='" & Trim(TextBox2.Text) &
"')")
End If
End If
```

طبعاً الكود السابق يكتب اعتماداً على وجود كامل ال **controls** الموجودة في الصورة السابقة.

رابط المشروع

[related tables.rar](#)

كيفية حفظ القيمة المنطقية (True Or False) في قاعدة البيانات

الكاتب : [lo2i](#)

[رابط المقال و النقاش حوله](#)

كثرت في الآونة الأخيرة الأسئلة عن كيفية حفظ القيمة المنطقية true أو false في حقل ما في جدول ما في قاعدة ما

أولاً : سنبدأ مع قاعدة البيانات Access

ننشئ جدولاً واحداً بإسم users

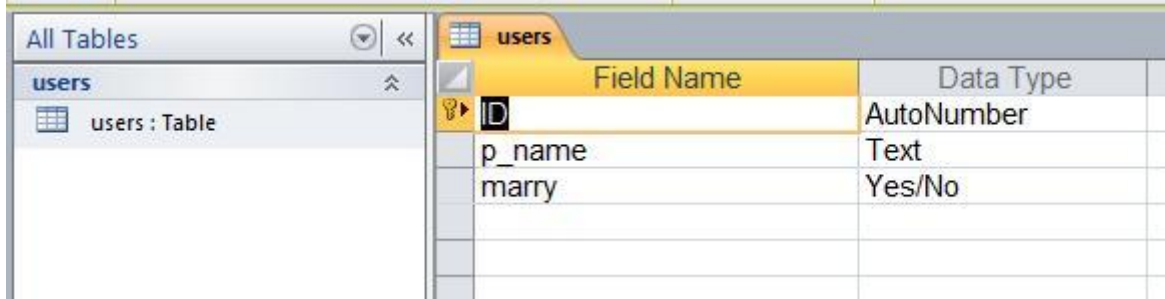
وننشئ بداخله الحقول التالية

<< id للترقيم التلقائي للسجلات auto number >>

<< p_name إضافة اسم الشخص text >>

<< marry إضافة حالة الشخص (متزوج ، أعزب) حقل منطقي yes/no >>

وهذه صورة الجدول بعد إضافة الحقول عليه..



Field Name	Data Type
id	AutoNumber
p_name	Text
marry	Yes/No

نذهب للبرمجة

ونصمم شاشة مبسطة



Form1

اسم الشخص

حاليه

متزوج

غير متزوج

حفظ

بعد تعريف نص الاتصال بقاعدة البيانات (لسنا في صدد شرحها الآن)

نكتب الكود التالي في زر الحفظ

كود:

```
Dim cmd As New OleDbCommand("insert into users(p_name,marry) values (?,?)", con)
cmd.Parameters.Clear()
cmd.Parameters.AddWithValue("@p_name", OleDbType.VarChar).Value =
Trim(TextBox1.Text)
cmd.Parameters.AddWithValue("@marry", OleDbType.Boolean).Value =
RadioButton1.Checked
con.Open()
cmd.ExecuteNonQuery()
con.Close()
MsgBox("تام بنجاح الحفظ تم")
```

ويكون شكل الكود كاملاً بهذا الشكل

كود:

```
Imports System.Data.OleDb
Public Class Form1
    Dim con As New OleDbConnection("provider=microsoft.jet.oledb.4.0;data
source=" & Application.StartupPath & "/bool.mdb")
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim cmd As New OleDbCommand("insert into users(p_name,marry) values
(?,?)", con)
        cmd.Parameters.Clear()
        cmd.Parameters.AddWithValue("@p_name", OleDbType.VarChar).Value =
Trim(TextBox1.Text)
        cmd.Parameters.AddWithValue("@marry", OleDbType.Boolean).Value =
RadioButton1.Checked
        con.Open()
        cmd.ExecuteNonQuery()
        con.Close()
        MsgBox("تام بنجاح الحفظ تم")
    End Sub
End Class
```

اعتقد أن الكود سهل وواضح جداً

وتكمن فكرة تخزين القيمة المنطقية في القاعدة بهذا الكود

كود:

```
cmd.Parameters.AddWithValue("@marry", OleDbType.Boolean).Value =
RadioButton1.Checked
```

وهو تعريف باراميتر في جملة الإضافة وتعين قيمة له خارجها

ولاحظ أن `oledbtype=boolean` = قيمة منطقية

المشروع في المرفقات

[save bool.rar](#)

معلومة ::: البحث الصوتي مع الدالة ::: Difference & soundex مهم جداً

الكاتب : BADRMEDIA

[رابط المقال و النقاش حوله](#)

موضوع البحث الصوتي من الأمور الهامة التي يجب أن نتعرف عليها ، لتطوير محركات البحث داخل برامجنا

الفكرة ببساطة:

في حالة تحويل نطق كلمة من لغة إلى لغة أخرى

بمعنى تحويل كلمات عربية وكتابتها بحروف لاتينية مثل " أحمد " تحول لحروف لاتينية هكذا " **Ahmed** "

نجد قواعد تحويل مزاجية وهوائية ترجع لعيب حركي في اللسان أو الإقامة في وسط له لهجة خاصة ككسر الحروف أو فتح الحروف دائما وهكذا

فنجد كلمة مثل هاني مثلا نجد من يكتب ممكن أن يكتبها على هذا النحو

Hani
Hany
Hanee
Honey
Heni
Haani
Haany
Hane

كود:

```
SELECT FirstName, LastName  
FROM FriendsData
```

ننظر للصورة الآتية:

```

SELECT      FirstName, LastName
FROM        FriendsData

```

	FirstName	LastName
1	Ahmed	Badr
2	Ahmad	Gamal
3	Yusef	Badr
4	Yousef	Badr
5	Hani	Badr
6	Hany	Badr
7	Hanee	Badr
8	Honey	Badr
9	Mohammed	Badr
10	Muhammed	Badr
11	Mahammed	Badr
12	Mahamed	Badr
13	Mohamed	Badr
14	Ahmed	ElsSalous
15	Ahmed	Elkourdy
16	Sameer	Abdelwahed

تحويلات كثيرة وغريبة :

هب أنك تريد البحث عن الحاج هاني الآن ماذا ستفعل ؟
لا أعتقد أنه بهذا الوضع أنك تستطيع الوصول لصديقنا هاني وأنت ستلجأ لهذا الحل :
أنك ستفرز الكلمات بترتيب الحروف وتبحث عن أي كلمات تبدأ بحرف " H " والأمر لله لأنه الحروف الوحيد لكونه بداية للكلمة صعب تغييره إلا لو كان هناك قبائل تقلب الحرف الآخر وهذا موضوع ليس في يدي!!!!!!!!!!!!

الحل برمجيا:

سنتعرف بداية على دالة (SOUNDEX()) ونتعرف على دورها وكيف تعمل

ببساطة شديدة ، هذا الدالة تكون أربع خانات تأخذ أول حرف من الكلمة والثلاث أرقام الأخرى عبارة عن أرقام تمثل النطق الصوتي

كود:

```

SELECT
    FirstName,
    SOUNDEX(FirstName) AS SoundTest
FROM
    FriendsData

```

ننظر للصورة التالية

```

SELECT
    FirstName,
    SOUNDEX(FirstName) AS SoundTest
FROM
    FriendsData

```

	FirstName	SoundTest
1	Ahmed	A530
2	Ahmad	A530
3	Yusef	Y210
4	Yousef	Y210
5	Hani	H500
6	Hany	H500
7	Hanee	H500
8	Honey	H500
9	Mohammed	M530
10	Muhammed	M530
11	Mahammed	M530
12	Mahamed	M530
13	Mohamed	M530
14	Ahmed	A530
15	Ahmed	A530
16	Sameer	S560

بعد ذلك سنتعرف على الدالة DIFFERENCE

تقوم هذه الدالة بحساب الفرق بين قيمتين ومن هنا تقوم هي بوضعها في اعتبارها أو رفضها

كود:

```

SELECT
    FirstName,
    SOUNDEX(FirstName) AS SoundTest,
    DIFFERENCE(FirstName, 'Hani') As Differ
FROM
    FriendsData

```

كما في الصورة

```

SELECT
    FirstName,
    SOUNDEX(FirstName) AS SoundTest,
    DIFFERENCE(FirstName, 'Hani') AS Differ
FROM
    FriendsData

```

	FirstName	SoundTest	Differ
1	Ahmed	A530	2
2	Ahmad	A530	2
3	Yusef	Y210	1
4	Yousef	Y210	1
5	Hani	H500	4
6	Hany	H500	4
7	Hanee	H500	4
8	Honey	H500	4
9	Mohammed	M530	2
10	Muhammed	M530	2
11	Mahammed	M530	2
12	Mahamed	M530	2
13	Mohamed	M530	2
14	Ahmed	A530	2
15	Ahmed	A530	2
16	Sameer	S560	2

سنلاحظ الحاج هاني كتبنا اسمه بنطق مختلف والنتيجة للصوتيات جيدة

نلاحظ أن مع الحاج هاني لما طلبنا من الدالة **DIFFERENCE** ترجع بالفرق عادت بالقيمة 4

إذن نتكلم قليلا عن هذه الدالة والمعامل **X** و **Y** اللذان يرسلان لها

عندما ننادي علي الدالة **DIFFERENCE(x, y)** تعود هي بقيمة من ٠ : ٤ كنتيجة لمستوي الفرق الصوتي بين الكلمة **x** و الكلمة **y** وظهور الرقم ٤ يعني أن الكلمتان **x** و **y** هما لكن مدخل البيانات واحد من الصعيد وواحد من وجه بحري وآخر من سيناء

أما لو ظهرت القيمة بصفر بـ ٠ فهذا النتيجة أنه لا يوجد تشابه صوتي بين هذه الكلمات

و عملية حساب الفرق تتم بهذه الطريقة

يتم تحويل الكلمة الموجودة في المعامل **X** بالدالة **SOUNDEX** إلي القيمة **SX** التي تحدثنا عنها سابقا وتتكون من أربع خانات

كذلك يتم تحويل الكلمة الموجودة في المعامل **Y** بالدالة **SOUNDEX** إلي القيمة **SY** التي تحدثنا عنها سابقا وتتكون من

أربع خانات

لو كانت الخانات الأربعة للقيمة المنتجة **SX** تساوي لخانات الأربعة للقيمة **SY** فإذن النتيجة ؛

لو كانت الخانة الأولى للقيمة المنتجة **SX** غير متساوية مع الخانة الأولى للقيمة **SY** فإذن النتيجة ٠ صفر ومات الكلام

في حالة الخانة الأولى للقيمة المنتجة **SX** متساوية مع الخانة الأولى للقيمة **SY** فالمعالج سيدخل على الخانات التالية وبناء على التوافق والتضاد للخانات التالية ستكون النتيجة إما ١ أو 2 أو ٣ ولو تطابقوا نصل للقيمة ؛ كتشابه تماثل بينهم

بناء على ما أثقلت به دماغك في الشرح السابق رغم سهولته ، يمكن أن تجعل النتائج تظهر بناء على مدى التقارب الذي تريده من ٠ : ؛

كمثال:

نبحث عن الحاج هاني بقية متطابقة ؛

أمور هامة حول الموضوع

هذه التقنية وخروج هذه الدالة موجودة من إصدار ٢٠٠٠ وليست مرتبطة بـ **SQL 90** بل مرتبطة بـ **T-SQL** لذا ستجدها في كلمة بيانات التطوير التي تدعم **T-SQL**

طبعاً القاعدة للدالة **SOUNDEX** تكون بتكون أربع خانات تبدأ بأول حرف من الكلمة والباقي أرقام

لا بد أن تعلم أن الدالة **SOUNDEX** تقوم بإغفال حروف العلة **Vowel Letters** الانجليزية (a e i o u) بالإضافة إلى الحرفين **h** و **y** ، مع العلم أن الحرف الأول لا يتم إغفاله حتى لو كان حرف علة

بناء على الكلام الذي ذكرته به و صدمتك به كخبر سيء

نجد أن كلمة مثل **ayna** تساوي **ana**

لأن **SOUNDEX** للكلمة **ayna** هي القيمة **A500** كذلك القيمة لـ **ana** هي **A500** طبعاً لأن حرف العلة تم إغفاله خارجاً

لكن لو كانت الحروف التالية بعد الحرف الأول كلها حروف علة زي الكلمة (**Dai**) فستحول الرموز إلي أصفار وتكون النتيجة لدالة حرف **D000**

أعتقد أنك فاض الكيل بك بسبب الجرائم المذكورة للدالة
أزيدك صدمات

بحكم أن الأرقام المكونة تكون لثلاث خانات فقط بعد الحرف الأول وهذا هو منهاج الدالة فستكون هناك كلمات طولها وحروفها الباقية لا قيمة لها مثل

كلمة عبد الملك **AbdullMalek** هي مثل كلمة عبد الباسط **AbdullBaset** نفس القيمة **A134**

يبدو أن من وضع الدالة كان لا يعرف غير الكلمات الثلاثية والرباعية **ToTo** و **LoLo** و **SoSo**

مشكلة:-

مصيبة أخرى الدالة **SOUNDEX** لا تعتبر صوتية بشكل منطقي فهي لا تميز الحروف الساكنة **Silent Letters** بصورة ٥٠% من حقها فاللغة الألمانية كحروف لاتيني لا تعرف الحروف الساكنة

فكلمة **Empty** تنطق في الانجليزية **ايمتي** وفي الألمانية **ايمبتي**

لكن بالنظرة الأخرى مسار التطوير في اللغة اللاتينية الاعتماد على الانجليزية البريطانية إذن عدم التعرف على الحروف الساكنة يجعل كلمة ليست مرتبطة لا من قريب ولا من بعيد متساوية في النطق مختلفة في القيمة مثال:-

كلمة **K500** قيمتها **K500** وكلمة **Now** نفس النطق قيمها **N000** (لا تنسي الأصفر تظهر بسبب حروف العلة)

وجه مظلم آخر حول الدالة أنها لا تفرق بين الحرف **M** والحرف **N** أثناء توليد الأرقام

فمثلا كلمة **Seem** هي نفس قيمة الكلمة **Seen** القيمة **S500**

وجه مظلم آخر

الحقول التي بها كلمتين لا تصلح مع هذه الدالة ، كما أنها تسير حرف حرف كما وضحنا إذن الكلمات الكبيرة تقع في مشكلة كما أن الكلمات التي بها حروف خاصة مثل كلمة

آل بدر **UI-Badr**

سيحصل على الحرف الأول والثاني وعندما يصل للشرطة سيتجاهلها ويتجاهل بقية الحقل

رغم هذا السوء أرى أن هذا الدالة مفيدة بنسبة كبيرة وتفيد المستخدم ويجب على المطورين التعامل بها بجانب تعرفهم على الخاصية **FullTextSearch** الهامة و التي يتمحور حولها مفهوم وعلم البحث الحديث

وأرى أنه يجب على ميكروسوفت أن تبدأ في التطوير فمن إصدار ٢٠٠٠ حتى الآن هذه الدالة محلك سر

هذا الاسكريبت لإنشاء الجدول والبيانات التي ضربنا عليها المثال

كود:

```
SET NUMERIC_ROUNDABORT OFF
GO
SET ANSI_PADDING, ANSI_WARNINGS, CONCAT_NULL_YIELDS_NULL, ARITHABORT,
QUOTED_IDENTIFIER, ANSI_NULLS ON
GO
IF EXISTS (SELECT * FROM tempdb..sysobjects WHERE
```

```

id=OBJECT_ID('tempdb..#tmpErrors')) DROP TABLE #tmpErrors
GO
CREATE TABLE #tmpErrors (Error int)
GO
SET XACT_ABORT ON
GO
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
GO
BEGIN TRANSACTION
GO
PRINT N'Creating [dbo].[FriendsData]'
GO
CREATE TABLE [dbo].[FriendsData]
(
[ID] [int] NOT NULL IDENTITY(1, 1),
[FirstName] [nvarchar] (50) COLLATE Arabic_CI_AS NOT NULL,
[LastName] [nvarchar] (50) COLLATE Arabic_CI_AS NOT NULL
)
GO
IF @@ERROR<>0 AND @@TRANCOUNT>0 ROLLBACK TRANSACTION
GO
IF @@TRANCOUNT=0 BEGIN INSERT INTO #tmpErrors (Error) SELECT 1 BEGIN TRANSACTION
END
GO
PRINT N'Creating primary key [PK_FriendsData] on [dbo].[FriendsData]'
GO
ALTER TABLE [dbo].[FriendsData] ADD CONSTRAINT [PK_FriendsData] PRIMARY KEY
CLUSTERED ([ID])
GO
IF @@ERROR<>0 AND @@TRANCOUNT>0 ROLLBACK TRANSACTION
GO
IF @@TRANCOUNT=0 BEGIN INSERT INTO #tmpErrors (Error) SELECT 1 BEGIN TRANSACTION
END
GO
IF EXISTS (SELECT * FROM #tmpErrors) ROLLBACK TRANSACTION
GO
IF @@TRANCOUNT>0 BEGIN
PRINT 'The database update succeeded'
COMMIT TRANSACTION
END
ELSE PRINT 'The database update failed'
GO
DROP TABLE #tmpErrors
GO
SET NUMERIC_ROUNDABORT OFF
GO
SET XACT_ABORT, ANSI_PADDING, ANSI_WARNINGS, CONCAT_NULL_YIELDS_NULL,
ARITHABORT, QUOTED_IDENTIFIER, ANSI_NULLS, NOCOUNT ON
GO
SET DATEFORMAT YMD
GO
-- Pointer used for text / image updates. This might not be needed, but is
declared here just in case
DECLARE @pv binary(16)

BEGIN TRANSACTION

-- Add 16 rows to [dbo].[FriendsData]
SET IDENTITY_INSERT [dbo].[FriendsData] ON
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (1,
N'Ahmed', N'Badr')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (2,
N'Ahmad', N'Gamal')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (3,

```

```

N'Yusef', N'Badr')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (4,
N'Yousef', N'Badr')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (5,
N'Hani', N'Badr')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (6,
N'Hany', N'Badr')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (7,
N'Hanee', N'Badr')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (8,
N'Honey', N'Badr')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (9,
N'Mohammed', N'Badr')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (10,
N'Muhammed', N'Badr')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (11,
N'Mahammed', N'Badr')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (12,
N'Mahamed', N'Badr')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (13,
N'Mohamed', N'Badr')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (14,
N'Ahmed', N'ElsSalous')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (15,
N'Ahmed', N'Elkourdy')
INSERT INTO [dbo].[FriendsData] ([ID], [FirstName], [LastName]) VALUES (16,
N'Sameer', N'Abdelwahed')
SET IDENTITY_INSERT [dbo].[FriendsData] OFF

COMMIT TRANSACTION
GO

```

والله الموفق لكل خير

كيفية اعتماد واستخدام الـ TRANSACTIONS داخل SQL Server

الكاتب : **Mr. Ahmed Negm**

[رابط المقال و النقاش حوله](#)

ماذا لو احتجت يوماً ما أثناء كتابتك لـ Stored Procedure داخل SQL Server إلى أن تقوم بتنفيذ ثمة عمليات في نفس الوقت ???

تعال مع نوضح بمثال جيد من الواقع العملي!! ...

أنت دخلت إلى بنك ، وتريد تحويل مبلغاً من المال (١٠٠ دولار) إلى (أحمد نجم) ... وهنا سيقوم الموظف المختص بعمليتان:

١. سحب (100) دولار (من حسابك الشخصي) .
٢. إيداع (100) دولار (في حساب) أحمد نجم.

ولكن ماذا إذا كانت هذه العملية تتم على (الكمبيوتر) ، وبعد سحب المبلغ من الحساب الأول ، صادف البرنامج خطأ ما أثناء التسجيل في الحساب الثاني ???

هناك في الحالات البدائية وضع من اثنين:

١. أن يكون هناك إجراء تحقق من عدم التسجيل ، وبناءاً على ذلك يقوم بإجراء برمجي آخر بإرجاع المبلغ المسحوب للحساب الأول .
٢. أو يكون البرنامج عقيم التقنية وبالتالي تضيع الـ (١٠٠ دولار) على صاحبها ، ولا تصل كذلك للحساب المحول له .

كذلك :: ماذا إن لم تكن عمليتي سحب وإيداع ... فلنفترض إنها مثلاً ١٠ عمليات سحب ، ٥ إيداع ، 5 تحويل وتسوية ... الخ ؟؟ ... أظن سيزداد الوضع سوءاً خاصة إذا كنت تبحث عن تزامن التنفيذ وعدم فقدان أي عملية من العمليات المحدد ، فإما يلغي التنفيذ للجميع ، وإما يُنفذوا بالكامل.

في درسنا اليوم سنتعامل مع الـ **TRANSACTION** ، وهي الخاصة بضبط تلك المسائل ضبطاً تاماً

الشكل العام للجملة:

كود:

```
BEGIN TRANSACTION

-- الأولى العملية -- INSERT INTO
-- الثانية العملية -- UPDATE
-- الثالثة العملية -- DELETE
-- الخ --

COMMIT TRANSACTION
```

راجع المثال التالي كي تلاحظ الكود بشكل أكبر ، علماً بأن قاعدة البيانات باسم (BANK) ، وجدول العملاء باسم (CUSTOMERS)، وحقل الرصيد باسم (BALANCE)

كود:

```
USE [BANK]

BEGIN TRANSACTION

-- First Code --
UPDATE [CUSTOMERS] SET [BALANCE] = [BALANCE] - 100 WHERE [NAME] = 'Abdo'
-- Second Code --
UPDATE [CUSTOMERS] SET [BALANCE] = [BALANCE] + 100 WHERE [NAME] = 'Ahmed'

COMMIT TRANSACTION
```

وبذلك تضمن تماماً عدم تنفيذ أي من العمليات والأكواد المكتوبة إلا إذا تم تنفيذ الجميع بشكل سليم ، وأكد ستتساءل عن كيفية عمل مثل هذه العمليات فنياً داخل محرك السيكوال سيرفر ، فكيف يقوم المحرك ذاته بالتراجع عن إجراء قد قام بتنفيذه بالفعل ، وماذا إذا كان هذا الإجراء معقداً نوعاً ما ... وهنا سأجيبك بأن هذا كله يتم تنفيذه داخل ملف الـ LOG الخاص بقاعدة البيانات ، وحين يصل الكومبايلر للسطر COMMIT ، فحينها يتم تطبيق تلك السطور الكودية على قاعدة البيانات ذاتها

كيفية التعامل مع المتغيرات داخل SQL Server

الكاتب : **Mr. Ahmed Negm**

[رابط المقال و النقاش حوله](#)

كثيراً ما تود العمل على إجراء عملية حسابية داخل **SQL Stored Procedure** أو **SQL Functions** ، مثل عمليات الجمع والطرح وما إلى ذلك ، أو أحيانا تود إسناد قيمة ما لمتغير معين لتعيد استخدام تلك القيمة فيما بعد فيما تريد. إليك الآتي:

الشكل العام للجملة:

كود:

```
<المتغير نوع> <المتغير اسم> @ DECLARE
```

مثال:

كود:

```
DECLARE @MyVar INT
```

حيث (**MyVar**) هو اسم المتغير ، ولا بد أن يكون مسبقاً دائماً بعلامة (@) في أي تعامل مع **SQL Server** ، وتمثل (**INT**) نوع المتغير لديك

أما عن إسناد القيم لهذا المتغير فتكون مثلاً كالآتي:

الشكل العام للجملة هو:

استخدم **SET** أو **SELECT ...** لوضع قيمة ما داخل المتغير المحدد كما يلي :

كود:

```
SET | SELECT @<VAR NAME> = VALUE
```

والمثال كما يلي:

كود:

```
SET @MyVar = 50
```

وكي تلاحظ الاستفادة من تلك المعلومة ، فقم بنسخ الكود التالي في محرر **SQL Server** وقم بالتجربة الآن:

كود:

```
DECLARE @MyVar INT  
SELECT @MyVar = 7  
PRINT @MyVar
```

أو المثال التالي ، والذي يرجع بعدد سجلات جدول معين ويضع العدد داخل جملة معينة بتنسيق معين كما يلي:

كود:

```
USE [ESHOP]  
DECLARE @MyVar INT  
SELECT @MyVar = (SELECT COUNT(*) FROM PRODUCTS)  
PRINT 'This table contains ' + CONVERT(VARCHAR,@MyVar) + ' records'
```

حيث (**ESHOP**) هو اسم قاعدة البيانات التي سأعمل عليها ، و (**PRODUCTS**) هو اسم الجدول المراد إرجاع عدد سجلاته.

كيفية استخدام IF داخل SQL Server

الكاتب : **Mr. Ahmed Negm**

[رابط المقال و النقاش حوله](#)

كثيراً ما تود العمل على الجمل الشرطية داخل **SQL Stored Procedure** أو **SQL Functions** ، وذلك لفحص نتائج معينة والقيام بتنفيذ مهام معينة بناءً على تلك النتائج والقيم وما إلى ذلك ... إليك الآتي:

الشكل العام للجملة:

كود:

```
IF <الفحص محل الشرط>  
<النتيجة>
```

أما عن الجمل الشرطية ، والمتبوعة بنتيجة شاملة لما يخالف الشرط الأساسي فبنائها كما يلي:

كود:

```
IF <الفحص محل الشرط>  
<النتيجة>  
ELSE  
<الأخرى النتيجة>
```

لاحظ الأمثلة التالية:

كود:

```
USE [ESHOP]  
  
DECLARE @MyVar INT  
  
SELECT @MyVar = (SELECT COUNT(*) FROM PRODUCTS)  
  
IF @MyVar = 0  
PRINT 'ZERO'
```

أو المثال التالي:

كود:

```
USE [ESHOP]  
  
DECLARE @MyVar INT
```



```

SELECT @MyVar = (SELECT COUNT(*) FROM PRODUCTS)

IF @MyVar = 0
PRINT 'ZERO'
ELSE
PRINT 'Not ZERO'

```

سنتجه لمعالجة مشكلة أخرى وهي كما يلي:

ماذا لو أردنا تنفيذ مجموعة من السطور المختلفة كنتيجة لتحقق شرط ما ، فمثلاً لو نظرنا للكود السابق وأردنا أن نطبع السطر (ZERO) وكذلك السطر (ZERO INDICATES THAT YOU HAVE NO RECORDS) تحت بعض ، تعال نستخدم الكود التالي ونرى النتيجة:

كود:

```

USE [ESHOP]

DECLARE @MyVar INT

SELECT @MyVar = (SELECT COUNT(*) FROM PRODUCTS)

IF @MyVar = 0
PRINT 'ZERO'
PRINT 'ZERO INDICATES THAT YOU HAVE NO RECORDS'

```

ستجد أن النتيجة الخاصة بتنفيذ هذه الجمل هي طباعة السطر الأخير ، ولم تظهر كلمة (ZERO) في النتائج ، ولكن دعونا نصلح الخطأ ونضيف رونقاً برمجياً باستخدام البلوك أو التركيب الكودي المسمى **BEGIN ... END** كما يلي:

كود:

```

USE [ESHOP]

DECLARE @MyVar INT

SELECT @MyVar = (SELECT COUNT(*) FROM PRODUCTS)

IF @MyVar = 0
BEGIN
    PRINT 'ZERO'
    PRINT 'ZERO INDICATES THAT YOU HAVE NO RECORDS'
END
ELSE
BEGIN
    PRINT 'Not ZERO'
    PRINT 'YOU ALREADY HAVE AT LEAST OR MORE ONE RECORD'
END

```

وبذلك تمكنا من التغلب على المشكلة

درس كيفية معرفة البلاد لزوار الموقع

الكاتب : [programer1](#)

[رابط المقال و النقاش حوله](#)

طلب مني احد العملاء تصميم موقع تجاري هذا الموقع يتعامل مع البلاد خارج الدولة فقط .. وقال لي أريد أن اعرف مكان الشخص الذي سيرسل رسالة لي طلب الخدمة

(طبعا لو كان من نفس الدولة شركة منافسة له ستعرف أسعار الخدمات عنده)
طبعا أنا قلت المثال لتتضح أهمية معرفة بلد الزائر أو الشخص طالب الخدمة للموقع وطبعا لها أهميات ثانية

كود:

```
using System.Globalization;
```

طبعا أنا فتحت مكتبة **Globalization** الخاصة بالتعامل مع اللغات والثقافات المختلفة والبلاد

كود:

```
string name = RegionInfo.CurrentRegion.DisplayName ;  
Response.Write (name) ;
```

شغل الصفحة الآن وستعرض لك البلد التي أنت منها

طبعا في خصائص كثيرة جميلة في هذه الفئة مثل

كود:

```
RegionInfo.CurrentRegion.EnglishName ;
```

لعرض اسم البلد باللغة الانجليزية

كود:

```
RegionInfo.CurrentRegion.NativeName ;
```

لعرض اسم البلد بنفس اللغة الأولى لتلك البلد : مصر مثلا ستعرض مصر عشان اللغة الأولى فيها العربية وهكذا

كود:

```
RegionInfo.CurrentRegion.ThreeLetterISORegionName
```

ستعرض أول ٣ حروف باللغة الانجليزية
مصر مثلا ستكون **Egy**

كود:

```
RegionInfo.CurrentRegion.CurrencyEnglishName
```

ستعرض عملة البلد .. مثلا مصر العملة الجنيه .. الإمارات العملة الدرهم .. وهكذا
طبعا تلك الفئة مليئة بالأشياء الهامة والشيقة وللمزيد راجع الآتي :

<http://msdn.microsoft.com/en-us/libr...8vs.71%29.aspx>

ولكن هذه الطريقة تعرض البيانات من خلال الفئة الخاصة باللغات في جهاز المستخدم
وهذا عيب أكثر منها ميزة : لأنه لو المستخدم غير مكان البلد من **change location** في الويندوز سيتغير البلد فعلا
توجد طريقة أخرى أكثر أمانا وهي عن طريق الآي بي الخاص بالجهاز

شرح عمل تصويت بتقنية ajax

الكاتب : أحمد AHMAD

رابط المقال و النقاش حوله

أقدم شرح جميل ومميز يشرح كيفية عمل تصويت باستخدام تقنية الأجاكس العملية الشرح موضح بالصور حتى النهاية
رابط تحميل الكتاب من هنا

http://www.ahmad.somee.com/FileUpload/o_ajax_Book.zip

ورابط تحميل المشروع الجاهز من هنا

http://www.ahmad.somee.com/FileUpload/o_ajax.zip

التعامل مع الصور في تقارير الـ.net

الكاتب : [al-raheeb](#)

[رابط المقال و النقاش حوله](#)

معلومات الدرس:

نوع الدرس : **الدرس عبارة عن ملف فيديو**

المدة : **22 دقيقة تقريبا**

الحجم : **54 ميغا بايت**

قسمت الدرس إلى رابطين لملف واحد مضغوط (يجب تحميل الرابطين ليتمكن فتح الضغط عن الملف)

الرابط الأول : [التعامل مع الصور في التقارير part1.rar](#)

الرابط الثاني : [التعامل مع الصور في التقارير part2.rar](#)

المشروع الذي كان عليه الدرس : لغة فيجول بيسك ٢٠٠٨ (ستجدون المشروع في المرفقات)

بعد مشاهدة الدرس أضمن لك أنك ستكون قادرا على:

- 1- دمج أو تضمين صور دخل التقرير ومن ثم استعراضها
- 2- عرض صور من قاعدة البيانات داخل التقرير
- 3- عرض صور من ملف خارجي داخل التقرير وغير ذلك.

الملفات المرفقة: [ImageAlbum.rar](#)

التقارير الفرعية في الكريستال (فن و إبداع) الجزء الثالث

الكاتب : [ابن النيل](#)

[رابط المقال و النقاش حوله](#)

كيف أصمم تقرير فرعي وكيف أربطه بالتقرير الرئيسي و ما هي أهميته ؟

اعلم أن هذه التساؤلات تأتي من فترة لآخري في ذهن البعض وما أهميتها بالنسبة له . أنا معك تماما حيث إنني شخصا لم أحاول تعلمها والسيطرة عليها إلا لما احتجتها وبشده . وهنا تأتي المقولة الشهيرة (الحاجة أم الاختراع) إذا كيف يكون شكله النهائي :

هذا هو شكله:

ابن النيل

أسماء الإدارات والموظفين الموجودين فيها

الكود	أسم الإداره	أدارة المشتريات
1.000		

id	name	tel	title
8.000	محمد ١		
9.000	محمد ٢		

الكود	أسم الإداره	أدارة المبيعات
2.000		

id	name	tel	title
1.000	hamada	0	eg
2.000	hamada1	00	eg
3.000	hamada2	000	eg
4.000	hamada3	0000	eg
5.000	hamada4	00000	eg

وكيف أصممه اتبع معي الصور لتعرف كيف تصممه وبدقه وباحتراف

تابع الصور :

أسماء الإدارات والموظفين الموجودين فيها

Section3 (Details)

management

اسم الإدارة

id

لعدد

Section4 (Report Footer)

Section5 (Page Footer)

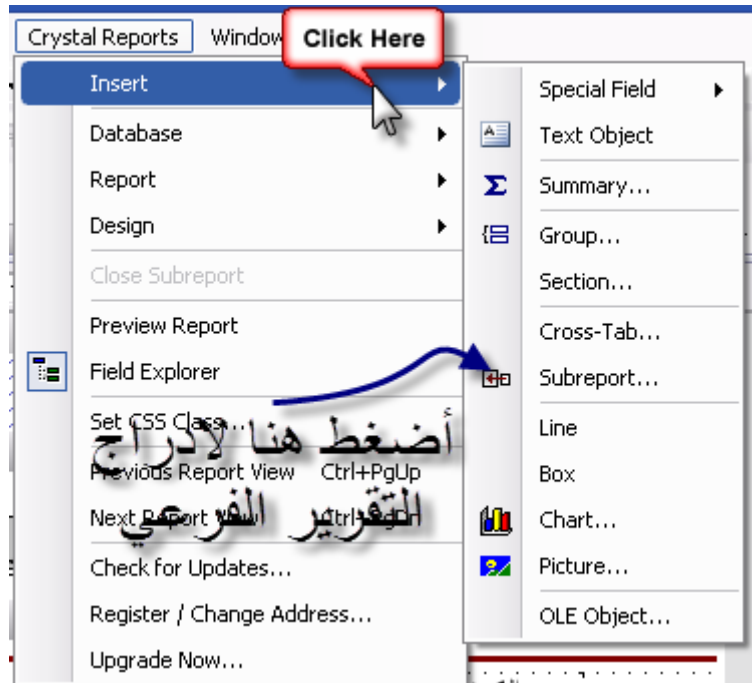
Page N of M

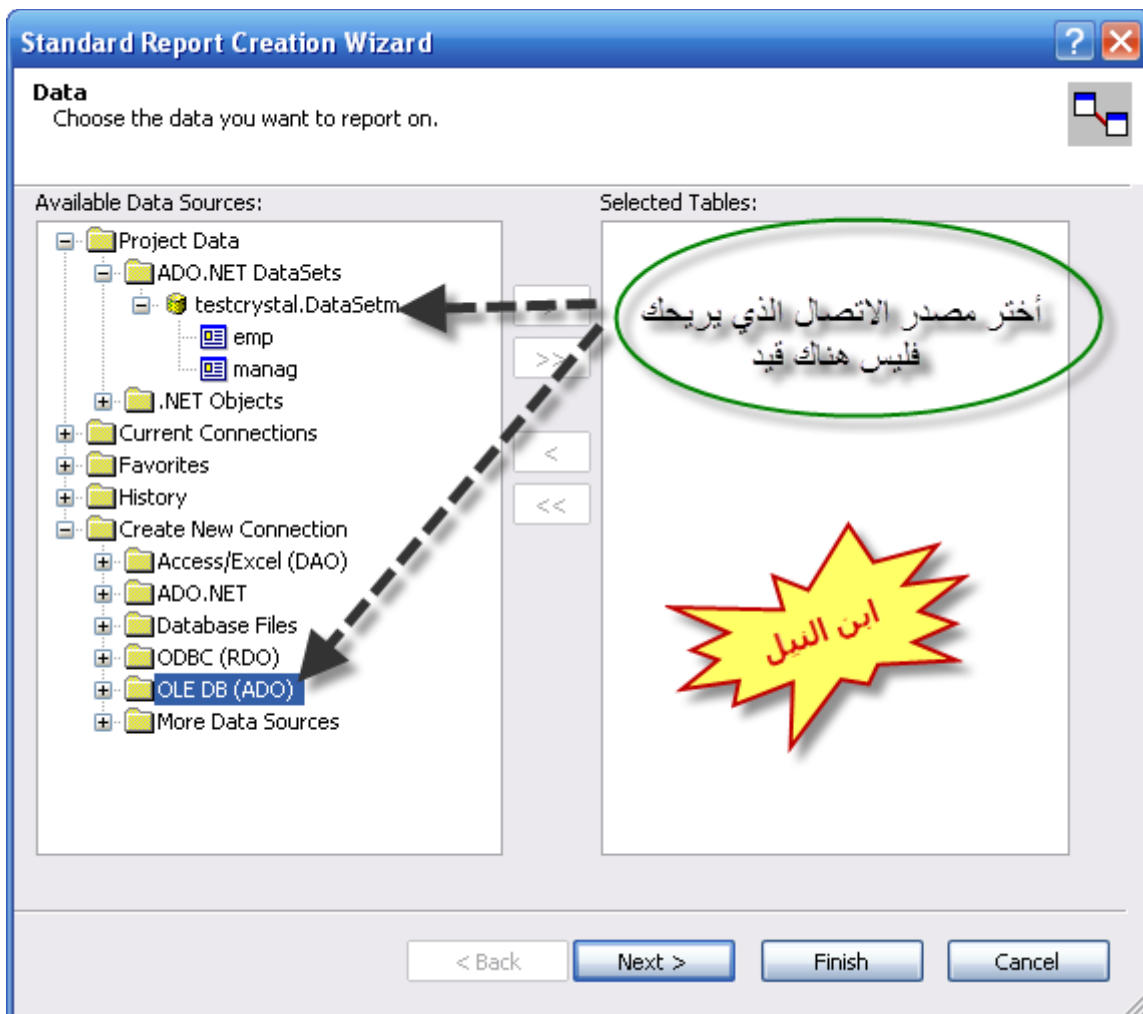
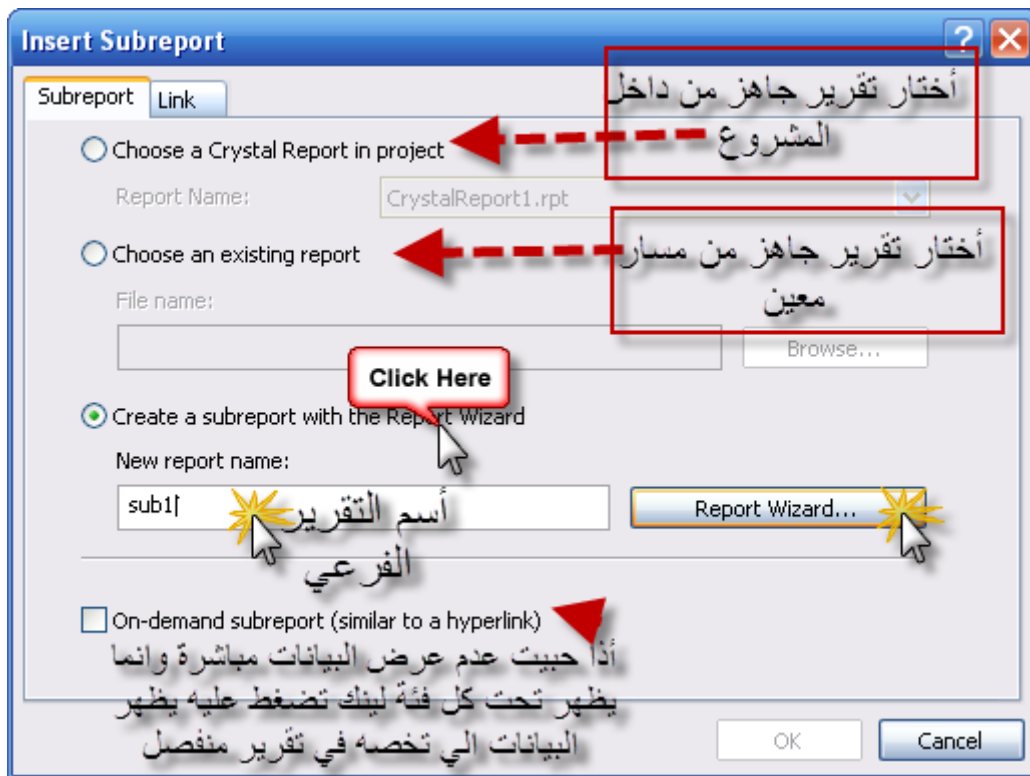
الشكل العادي الذي تعودنا عليه جميعاً ونستطيع تصميمه ويسمى التصميم الأب
يجب ان يكون التقرير الابن مصمم على جدول مرتبط بجدول التقرير الاب
بعلاقة رأس لأطراف (علاقه منطقيه)

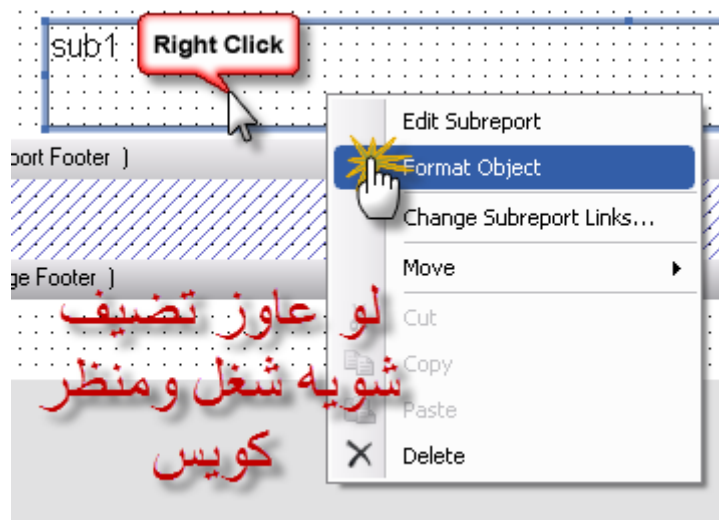
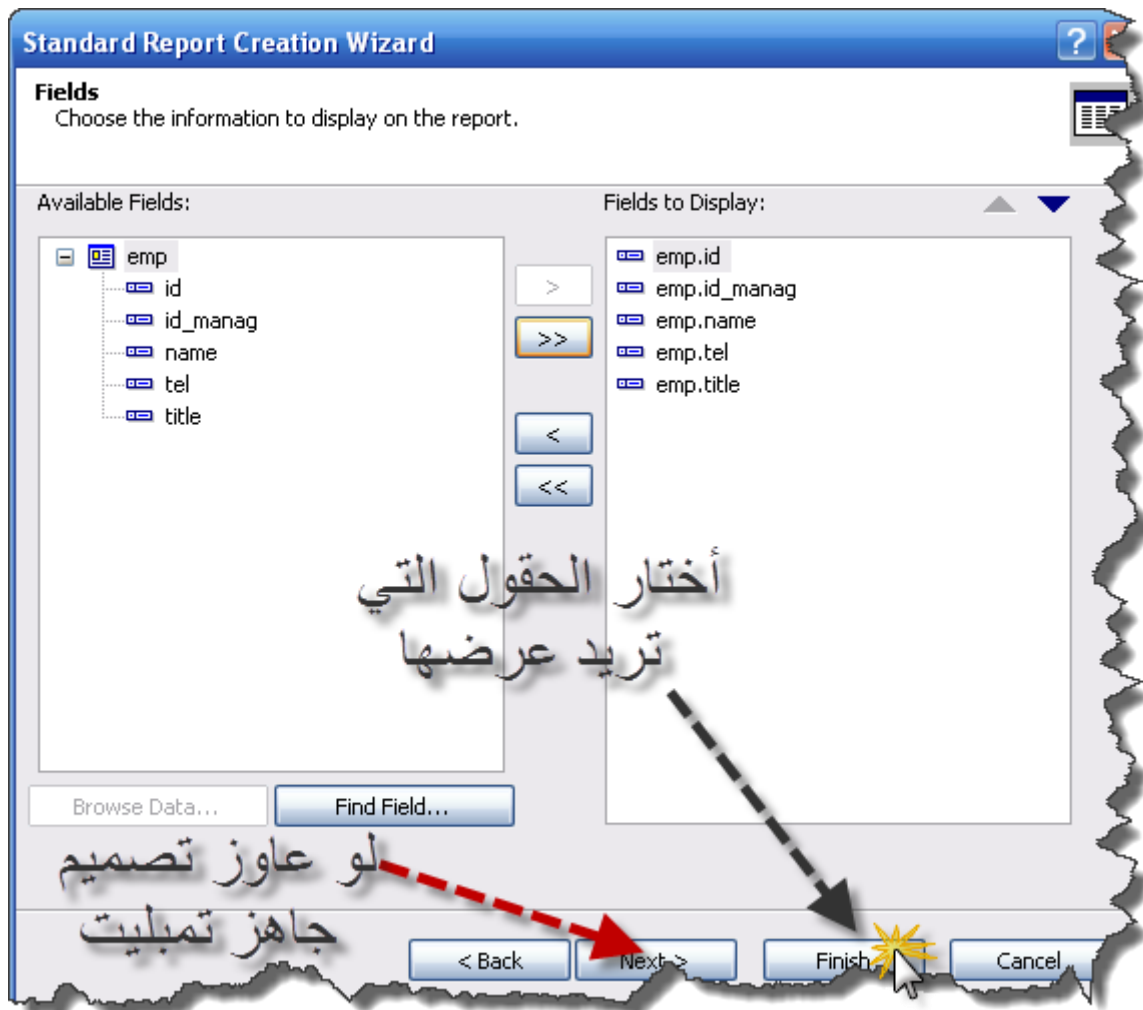


أضغظ هنا لإدراج التقرير
الفرعي وتصميمه

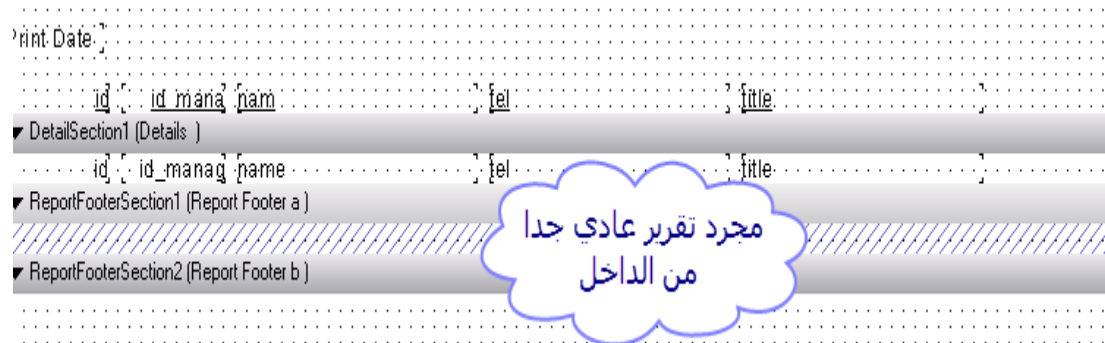
أو من القوائم ممكن أن تختاره أيضا

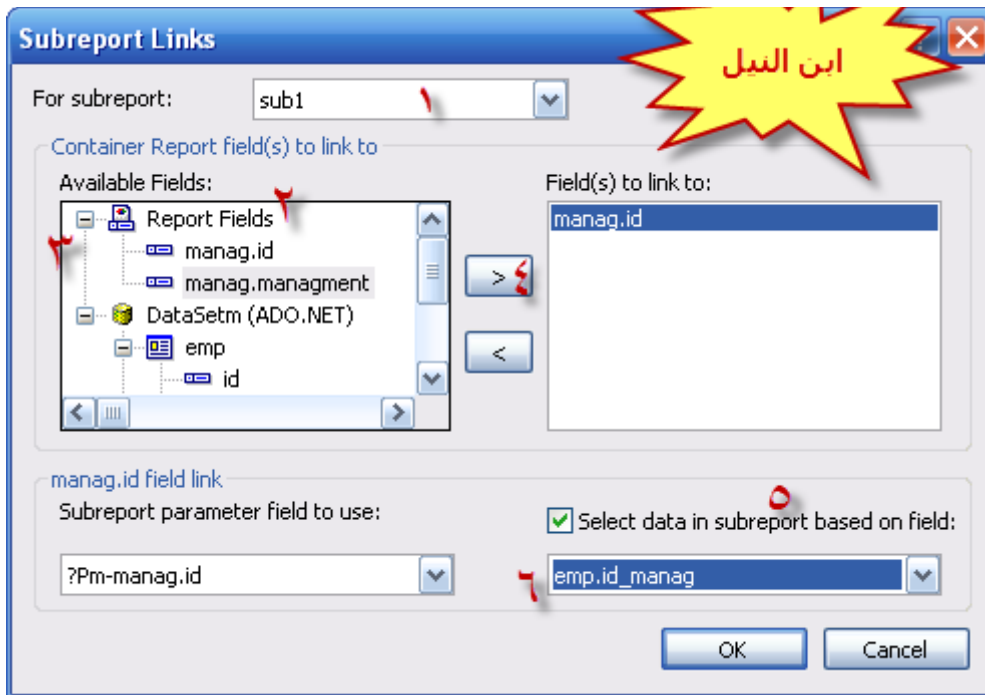
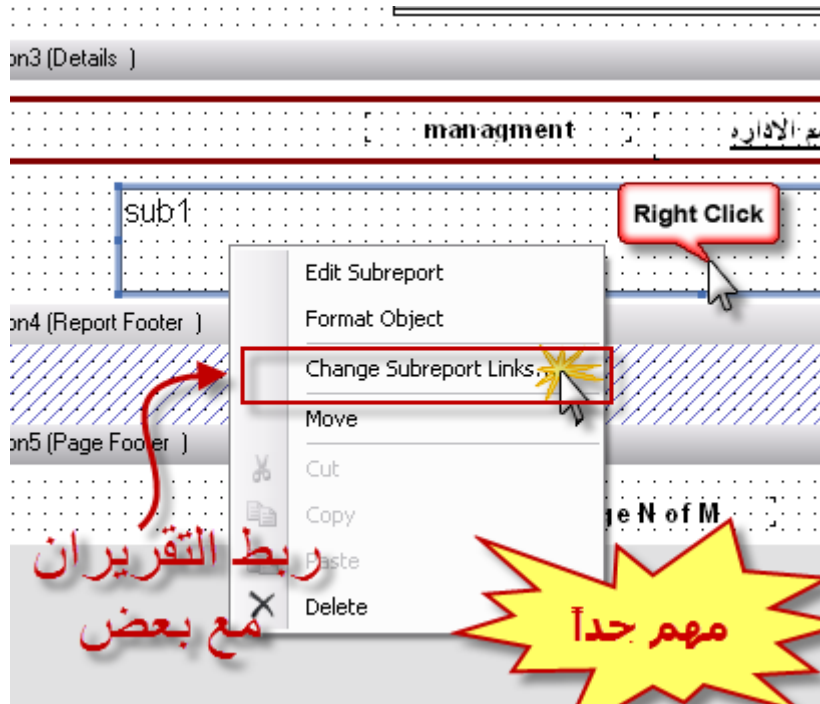












شرح الأرقام في الصورة الخيرة:

تعد هذه الشاشة من أهم الشاشات لأنها هي المسؤولة عن ضبط التقرير فبدونها سوف تظهر كل النتائج الفرعية تحت كل الفئات الرئيسية في حالة من التكرار والفوضى

لذلك :

- 1 - تحديد التقرير الفرعي الذي نتعامل معه
- 2 - من هذا القسم قسم حقول التقرير نختار حقل الكود الذي يربط الجدولين مع بعض والذي نختاره كود الجدول الرئيسي
- 3 - نفس الشرح أو ممكن نقول رقم زايد في الصورة
- 4 - انقل الحقل إلى حدها إلى الجهة الأخرى لاحظ في الصورة أنا نقلت حقل id من جدول الإدارات فقط وهو الصحيح فقط ولا ننقل أي سجلات أخرى

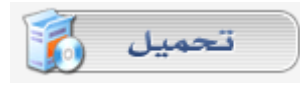
- 5 - ضع علامة صح ليفتح لك الصندوق إذا لم يكن موجود
6 - أختار حقل الربط في جدول الموظفين وهو الحقل الذي نحفظ فيه رقم إدارة هذا الموظف حتى أثناء العرض يلتزم كل موظف بإدارته

ثانياً:

كيف أبرمجه وأرسل له البيانات؟ لكي أجيب علي هذه النقطة ببساطه راجع المقال السابق من هنا

<http://www.vb4arab.com/vb/showthread.php?t=60513>

المرفقات نفس الملف في الجزئين السابقين ومره أخرى الرابط هنا



السر في تصميم الاتصال بالكريستال للقواعد الغير مدعومة 2

الكاتب : [ابن النيل](#)

[رابط المقال و النقاش حوله](#)

السؤال الأول كيف نصمم التقرير على هذا النوع من قواعد البيانات وكيف نتصل به من الكريستال ؟
كيف نصمم تقرير على الكريستال لأي قاعدة بيانات نريدها وكيف نتصل بها ؟

الجواب :

إن من يفهم الكريستال حق الفهم يعلم كيف صممت البنية التحتية وهيكله . بمعنى من أهم وسائل تصميم أي تقرير هو أولاً تحديد اتصال مع هيكل القاعدة سواء الحقيقية أو الوهمية (الملفات الوصفية XML لهيكل قاعدة البيانات مثل Linq و Entity framework و Data set وملفات ORM و الملفات التي تصف قاعدة البيانات التي يتم كتابتها بشكل يدوي)
لمزيد راجع كتاب [سامر سلو](#) في كتابة كلاسات اللينك يدويًا.

يعني ماذا سأكتب توصيف للقاعدة واكتب خصائصها وحقولها والربط , مستحيل طبعاً

وهذا رأيي, ما هو الحل تابع الصور وأنت ستعرف السر

أنا كنت قبل هذا الوقت أقول في دروسي إن هناك فروق مهمة بين كلاً من الكريستال xi والكريستال الحر كريستال ١٠

والآن هو الوقت لنكشف السر الذي بحث عليه أناس كثير قبل الآن و عطل مشاريع وأرغم أناس على تغيير قاعدة بيانات من أجل عدم قدرتهم على بناء التقارير المناسبة لقاعدة بياناتهم ومن أهمهم

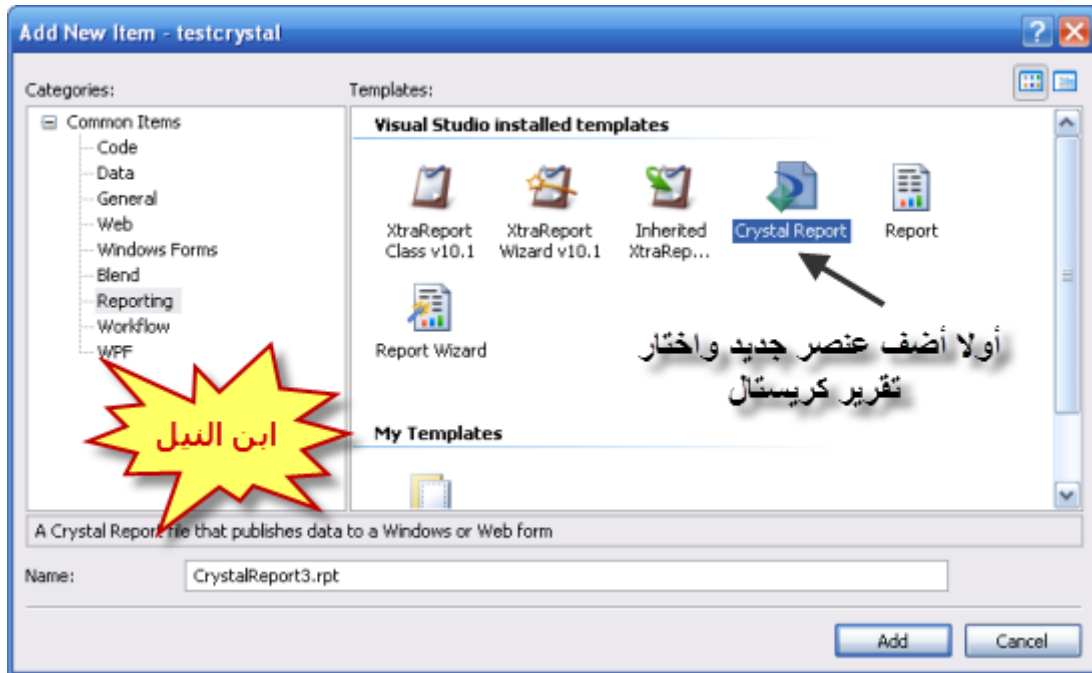
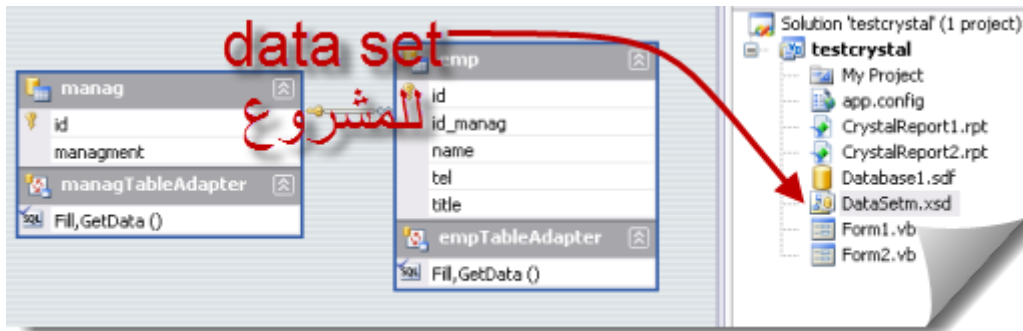
MYSQL و قطتي الصغيرة SQLITE

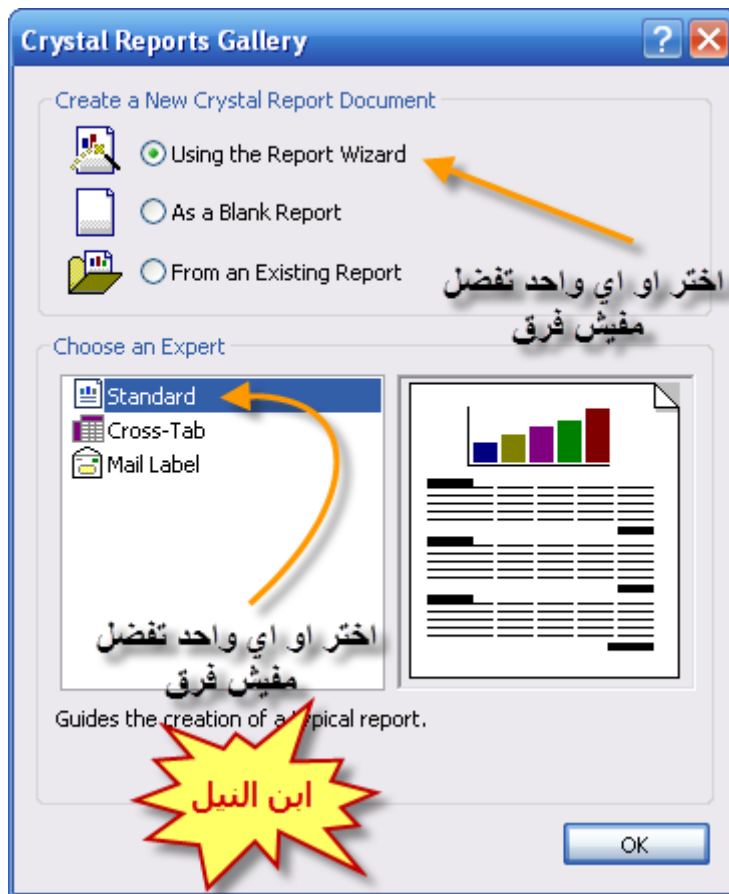
لحل هذه المشكلة يجب استخدام كريستال ريبورت xi المدمج في الفيچوال أستوديو وتوافر DataSet لتكون هي ملف الموصف لقاعدة البيانات في شكل XML

ملاحظة: أي قاعدة بيانات يمكن الاتصال بها في بيئة الدوت نت ويمكن صنع لها DataSet من النوع الجاهز إذ يمكن تصميم تقرير لها بدون حتى أي اتصال مع القاعدة كل ما نحتاجه هيكل القاعدة المعروف في لغة ال SQL ب DDL

نبدأ العمل علي بركة الله

[لاحظ الصور](#)





Standard Report Creation Wizard هو أداة مهمة جدا

Data
Choose the data you want to report on

السريع في
crystal XI

اسم Dataset في مشروعك

Available Data Sources:

- Project Data
 - ADO.NET DataSets
 - testcrystal.DataSetm
 - emp
 - manag
 - .NET Objects
- Current Connections
- Favorites
- History
- Create New Connection

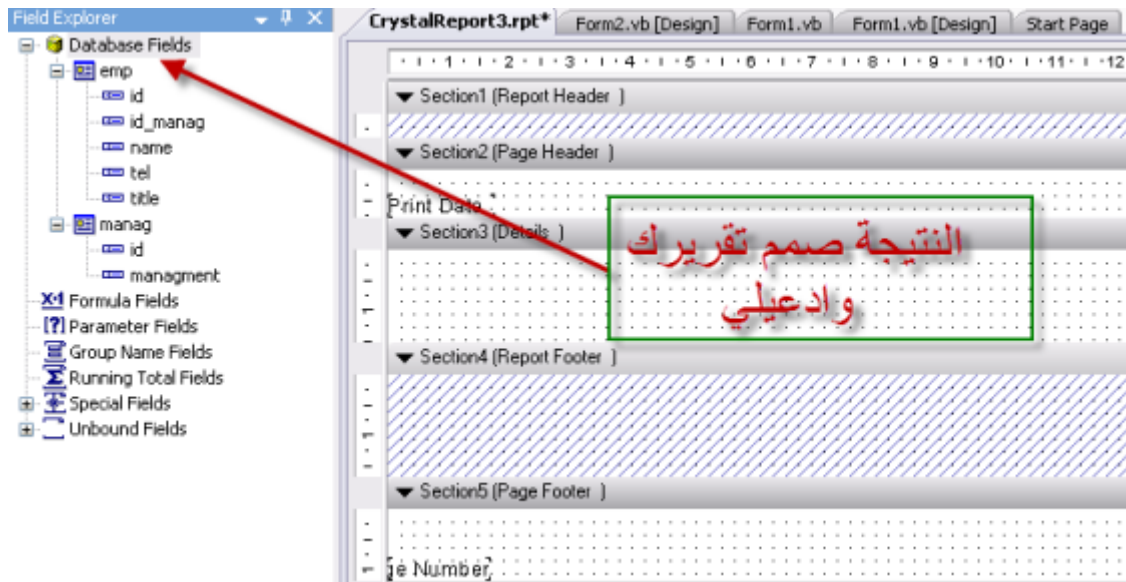
Selected Tables:

اسماء الجداول الموصوفه داخل الداتل ست

مهمه جدا

ابن النيل

< Back Next > Finish Cancel



الجزء الأول

حل مشاكل الكريستال مع 2010 vs

الكاتب : [ابن النيل](#)
[رابط المقال و النقاش حوله](#)

المشكلة الأولى :

كريستال ريبورت غير موجود أساسا مع فيجوال ستوديو ٢٠١٠ سواء أي Name Space أو المصمم للتقارير أو أداة عرض التقارير

الحل:

يتم تحميل الحزمة التالية التي بها كل هذه النواقص في فيجوال استوديو

<http://www.businessobjects.com/jump/...10/default.asp>

هذه المشكلة ظهرت لان شركة مايكروسوفت قامت بالفصل بين كلا من لغاتها والكريستال فعلي من يريد استخدامه إضافته من هنا (الرابط السابق)

المشكلة الثانية:

بالرغم من إنزال هذا البرنامج أو الحزمة لم يتم ظهور أداة عرض التقارير crystal viewer وأيضا عند أضافه ملف كريستال للمشروع تكون غير قادر علي تشغيل المشروع أو تنفيذ عملية البناء له

الحل :

المشكلة تكمن في أن كريستال ريبورت كان جاري بنائه في نفس الوقت مع فيجوال استوديو ٢٠١٠ أي يعتمد في بنائه علي framework 4 beta2 مما أدى عند نزول النسخة النهائية من framework4 أصبحت مختلفة في أشياء عن سابقتها في أشياء معينة وحل لمشاكل ما بشكل أفضل

ولحل المشكلة اتبع التالي:

- 1- افتح المشروع صاحب المشكلة
- 2- من solution explorer اختر my project للتعامل مع خصائص المشروع
- 3- اختر من الجانب الأيسر compile option
- 4- ثم اختر آخر زر بالأسفل وهو advanced compile option
- 5- سوف تجد نوع المشروع متوافق مع framework 4 client profile قم بتغييره إلى framework 4 فقط وسوف تلاحظ ظهور الأداة وأنتك أصبحت قادر علي تشغيل المشروع دون أخطاء

درس مبسط : تمرير القيم من برامجنا لتقارير الكريستال ريبورت

الكاتب : [نور نبهان](#)

[رابط المقال و النقاش حوله](#)

مقدمة:

غالباً ما تكون محتويات التقرير من قواعد البيانات المرفقة مع البرنامج. ولكن بفرض أننا نريد وضع جملة في أعلى التقرير تتغير برغبة المستخدم لبرنامجنا ولتكن اسم الشركة المستخدمة للبرنامج أو مثلاً لو أردنا نسخ ذاتية شخص معين بإدخال معلوماته الشخصية فقط للطباعة الفورية فهل نحن مضطرون لحفظ هذه القيم في قاعدة البيانات ومن ثم استيرادها طبعا لا وكذلك الأمر في حال جمع محتوى حقل معين فنحن نظهره في صندوق نص تحت بند المجموع الكلي فكيف نستطيع طباعة هذه المعلومات و المدخلات في نماذج برنامجنا وبدون استخدام قواعد البيانات هذا ما سنتناوله في درسنا اليوم إن شاء الله

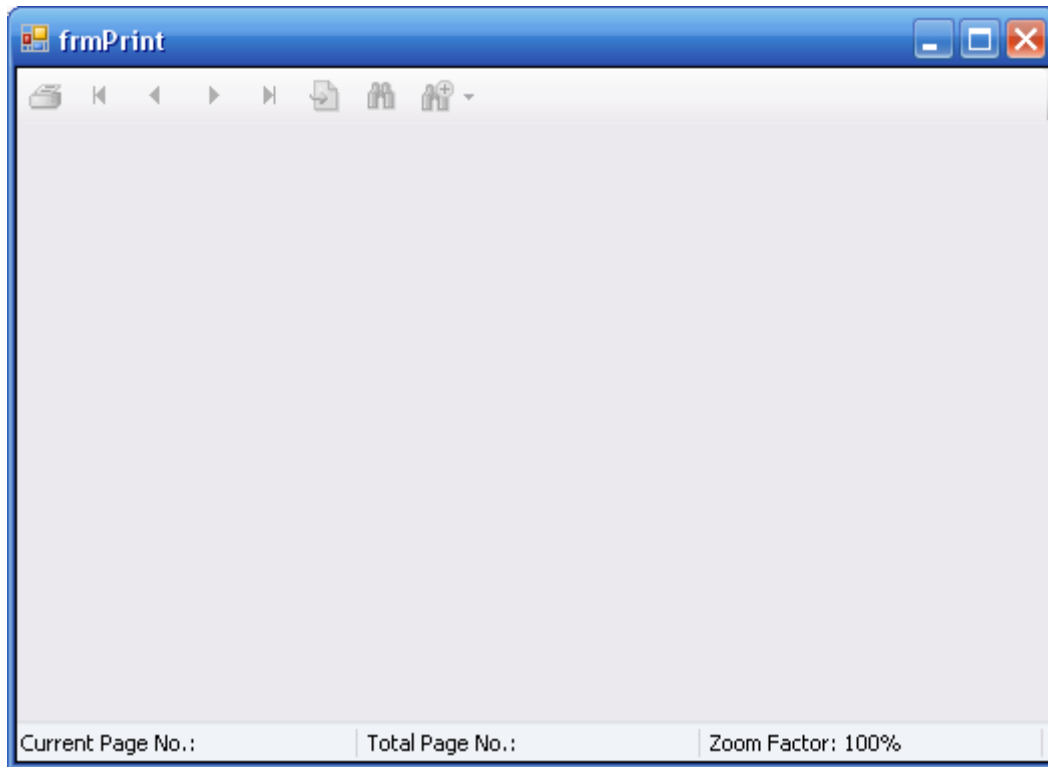
الدرس:

افتح Microsoft Visual Studio 2008 و أنشأ مشروع جديد تحت المسمى الذي تريد أذهب للنموذج Form1 وأضف عليه زر أمر Button1 وبعض الليبل Label و صناديق النص TextBox وهذا هو التصميم المبني للفورم الأول:

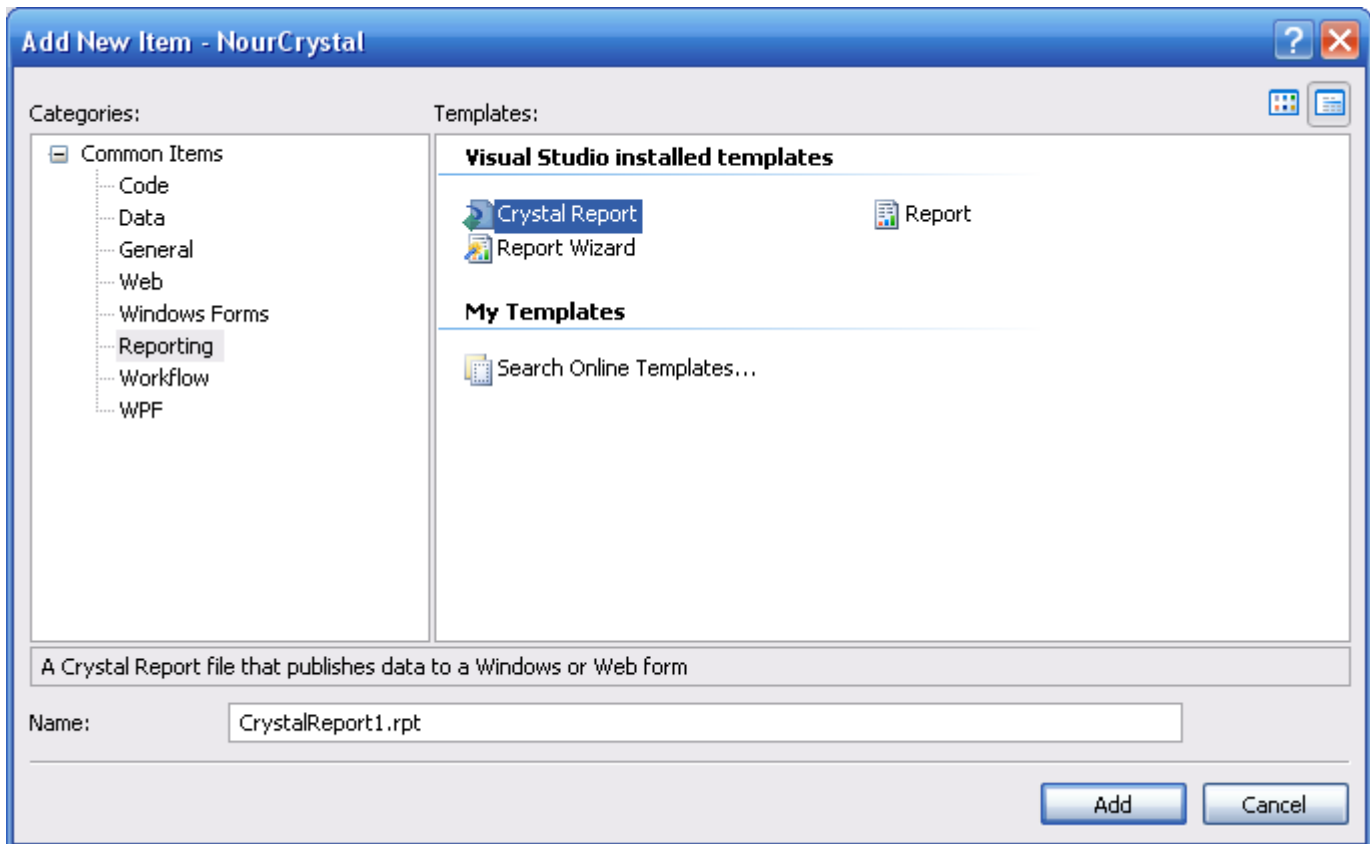
فيجوال بيسك العرب	اسم الشركة
نور نبهان	اسم المستخدم
85	العلامة الاولى
75	العلامة الثانية
160	المجموع

طباعة المدخلات

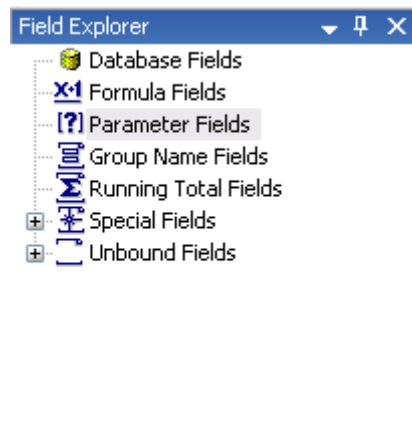
الآن فورم ثاني frmPrint أضف له CrystalReportViewer1 من شريط الأدوات وهذا شكل التصميم:



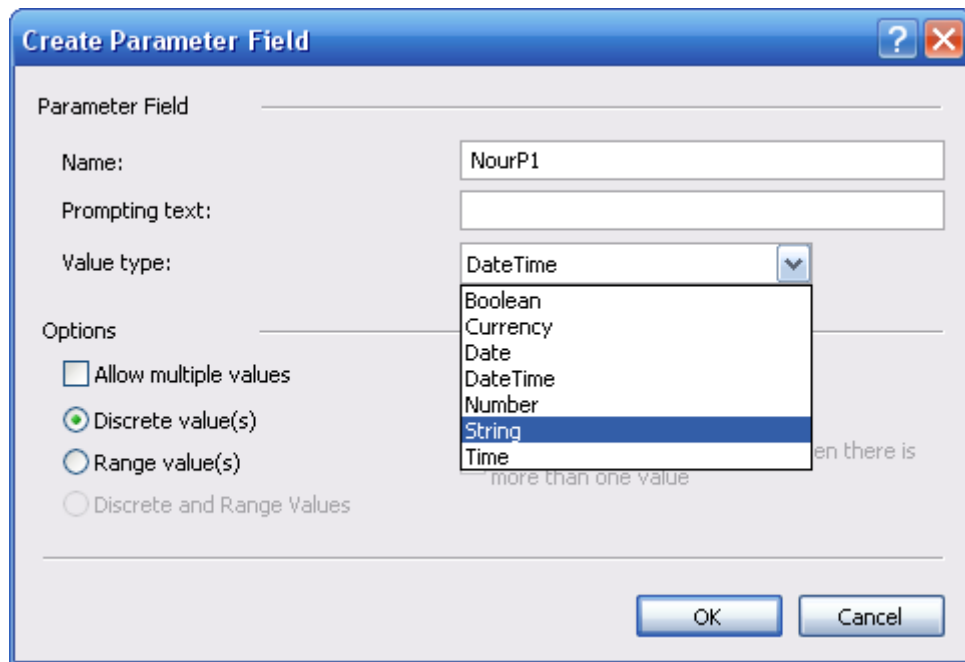
الآن من قائمة Project1 اختر Add New Item واختر تبويب Reporting وعلم على CrystalReport كما في الصورة التالية:



واختر Add
سيتم فتح التقرير في وضع التصميم:
اذهب لنافذة Field Explorer في الجهة اليسرى واختر Parameter Field



ثم اضغط بالزر الأيمن واختر NEW جديد. لإضافة متغير جديد للقائمة:



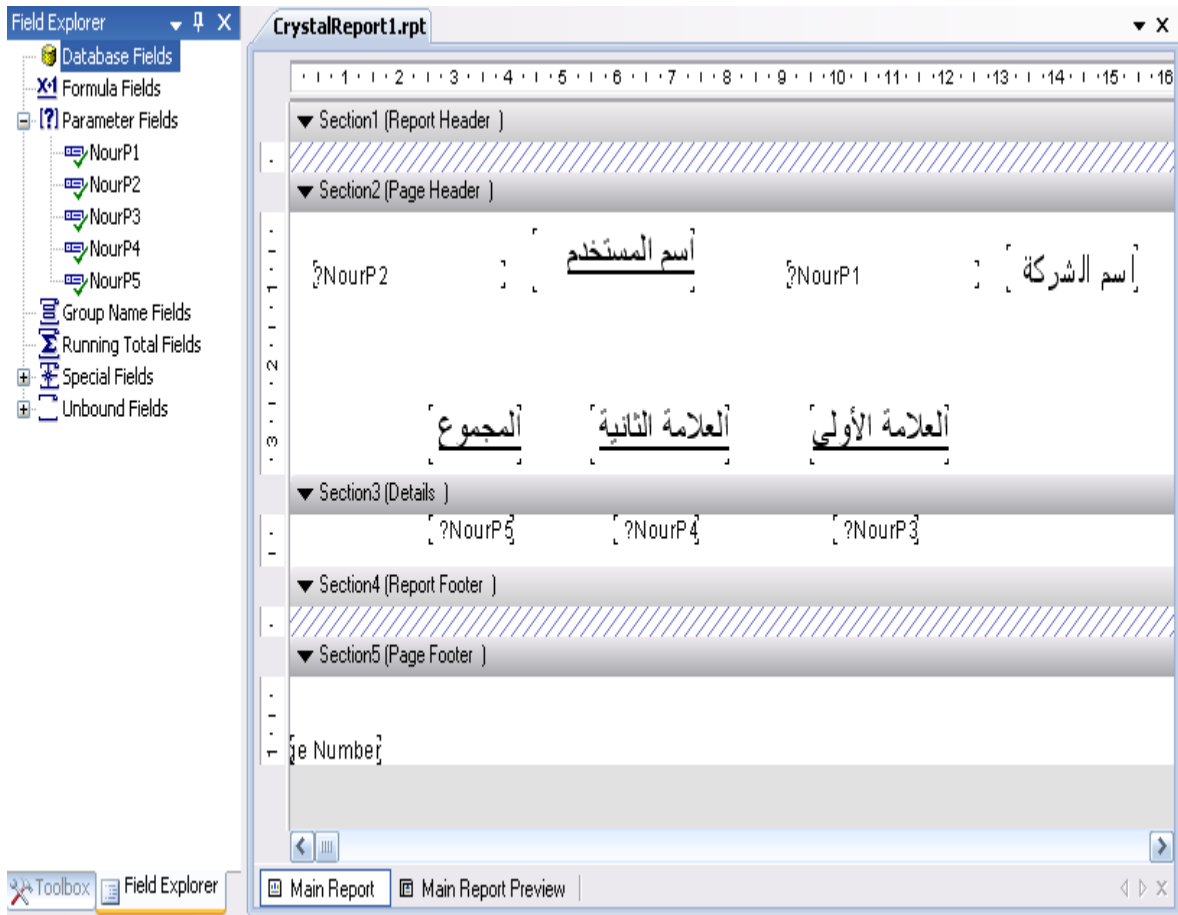
اختر أي اسم تريد وكذلك نوع المتغير من القائمة بحسب المطلوب وكرر ذلك بحسب ما تريد من متغيرات
أضف لمثالنا:

2 متغير من نوع String

3 من نوع Number وسمها مثلا:

NourP1,NourP2,NourP3,NourP4,NourP5

بعد إضافة المتغيرات قمت بسحبها بالفأرة إلى التقرير وإفلاتها في المكان الذي تريد كما في الصورة
للتوضيح :



لاحظ الأسماء الكبيرة هي فقط نصوص للتوضيح و المتغيرات هي ذات الخط الصغير:

بذلك انتهينا من تصميم النماذج والتقارير ونتجه إلى جوهر الموضوع وهو كتابة الكود.
أذهب للفورم الأول وضغطين على Button1 زر الطباعة واكتب الكود التالي:

كود:

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    ' الطباعة زر
    ' طباعة تقرير احداث
    Dim NourCrystal As New CrystalReport1
    ' المصدر
    frmPrint.CrystalReportViewer1.ReportSource = NourCrystal
    ' العرض حجم
    frmPrint.CrystalReportViewer1.Zoom(100%)
    ' التقرير تحديث
    frmPrint.CrystalReportViewer1.Refresh()
    ' الطباعة فورم عنوان تغيير
    frmPrint.Text = "صالحة دعوة من لاتنسونا"
    '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    ' الشركة اسم
    NourCrystal.SetParameterValue(0, TextBox1.Text)
    ' المستخدم اسم
    NourCrystal.SetParameterValue(1, TextBox2.Text)
    ' الاولى العلامة
    NourCrystal.SetParameterValue(2, TextBox3.Text)
    ' الثانية العلامة
    NourCrystal.SetParameterValue(3, TextBox4.Text)
    ' المجموع
    NourCrystal.SetParameterValue(4, TextBox5.Text)

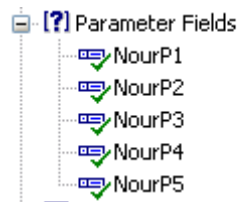
```

```
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%'  
frmPrint.Show()  
End Sub
```

طبعا الكود مشروح ضمناً
لكن لاستيعاب الأمر بالنسبة لتخزين قيمة المتغيرات:
لاحظ هذا السطر من الكود وهو كل موضوعنا
كود:

```
NourCrystal.SetParameterValue(0, TextBox1.Text)
```

فهنا قمنا بتمرير محتوى TextBox1 إلى المتغير الأول حيث يبدأ الترقيم من الصفر والمتغير في مثالنا كان NourP1
فمتغيرات التقرير التي أضفناها هي كما في الصورة:



ولكي نسند قيمة لأحدها نستخدم الكود السابق:
كود:

```
NourCrystal.SetParameterValue(x, y)
```

حيث قيمة x هو ترتيب المتغير ففي الصورة السابقة

NourP1=0
NourP2=1
NourP3=2
NourP4=3
NourP5=4

و قيمة y هي ما نريد وضعه في المتغير (نص رقمالخ)

يمكن أن نسند قيم صندوق نص كما سبق
أو قيمة متغير مخزن فيه عملية حسابية مثال

كود:

```
Dim x As Integer  
x = Val(TextBox3.Text) + Val(TextBox4.Text)  
NourCrystal.SetParameterValue(4, x)
```

ومن الإعدادات المخزنة:
كود:

```
NourCrystal.SetParameterValue(4, My.Settings.namecop)
```

أو بالحساب المباشر:
كود:


```
NourCrystal.SetParameterValue(4, Val(TextBox3.Text) + Val(TextBox4.Text))
```

أو أي شرط نريده.
فقط تذكر أن هذا كل ما تريد:

NourCrystal.SetParameterValue(القيمة, ترتيب المتغير)

[NourCrystal.zip](#)

طريقه عمل تقرير وفاتورة بفلتر بكل سهولة : في داتا جريد و ليبل وقاعدة أكسس

الكاتب : [el3ashe2](#)

[رابط المقال و النقاش حوله](#)

الهدف إن شاء الله عمل فاتورة على شكل معين وقابله للطباعة

الاستعلام سيكون عن طريق رقم الفاتورة

وسنطبع في الأخير على طباعة عاديه أو PDF لنحفظه على الجهاز لو أحببنا ولكن يجب تكون الطباعة الافتراضية لأننا لن نختارها ولو أحببنا سنضيف

بسم الله نبدأ المطلوب:

- 1- قاعدة بيانات أكسس و بها جدولين واحد للعملاء والأخر للبضاعة التي سنضيفها في فاتورة البيع
- 2- فيجوال ستوديو ٢٠٠٨ (٢٠٠٥ لن يضر سيعمل بنفس الأكواد إن شاء الله)
- 3- متابعه جيده لخطورة خطوة

كود:

```
Imports System
Imports System.Data.OleDb
Imports System.Drawing.Imaging
Imports System.Drawing.Printing
```

استدعاء بالطبع سنستخدمها لاختصار الأكواد وللتعامل مع قواعد البيانات وأيضا للتعامل مع الطباعة

سنستخدم التالي في الپابلك كلاس:

كود:

```
Public Class Form2
    Dim b As Boolean
    Dim Con As New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0 ;Data
Source=" & Application.StartupPath & "\Store.mdb")
    Dim sqlstr1 As String

    Dim cmd As OleDbCommand
    Dim dr As OleDbDataReader
    Dim dt1 As OleDbDataAdapter
    Dim ID As Integer
    Dim DataSet1 As New DataSet
    Dim DataSet2 As New DataSet
```

عمل الاتصال وسنسميه **Con**

الباقى معروف أساسيات وبعض المتغيرات المستخدمة

في الحدث **Load** لن سنستخدمه غير انه يعود بالتاريخ بالكود هذا طبعا

كود:

```
Label23.Text = Date.Now
```

صورة للفاتورة

رقم الملف الضريبي	شركه اسم الشركة طبعا هنا	رقم التسجيل الضريبي	هنا رقم الفاتورة
			الى حستعلم بيه
		فاتوره رقم:	<input type="text"/>
		بيانات العميل	
-	التليفون:	-	اسم العميل:
-	فاكس:	-	رقم العميل:
-	التاريخ:	-	العنوان:

<http://dvd4arab.com>
شرح الفاتورة
by : El3ashe2

نرجو ذكر المصدر عند النقل لمنتدى اخر
منتديات فيجوال بيزك العرب

شروط الدفع:

- نقدا او بيشيك قابل للدفع
- ضريبه مبيعات 10%
- اجمالي القيمة المطلوبة

دى حساب السعر
والضريبة والاجمالي

لطباعه في النهاية 

نحتاج إلى ٢٣ لبيبل و 1 تكست بوكس و 2 زر أمر و ١ داتا جريد فيو

الفكرة في سحب بيانات الفاتورة برقم الفاتورة من قاعدة البيانات

وعرضها في الداتا جريد

وسحب من نفس الجدول بيان رقم العميل واستخدامه في سحب بيانات العميل من جدول العميل في الأعلى

ثم أخيرا الطباعة وهذا كود جاهز

نكمل الأكواد

أول أزرار البحث في الداتا بيز باستخدام رقم الفاتورة في التكتست بوكس سنضغط على الزر مرتين لفتح نافذة الكود الخاص فيه

ونضع فيه الكود التالي أنا وضعت فيه اقتباسات للشرح

كود:

```
'===== الفاتورة برقم البحث =====  
  
DataSet1.Clear()  
If TextBox1.Text = "" Then  
    MsgBox("الفرغ ملأ يجب", MsgBoxStyle.Information)  
    TextBox1.Focus()  
Else  
    sqlstr1 = "SELECT Quntity As الكمية,Price As السعر,inv As  
الفاتورة_رقم,ItemCode As الاجهزة_سيريال,ItemName As المواصفات,ClientCode As  
العميل_رقم From out WHERE inv LIKE '%" & TextBox1.Text & "%'"  
    Try  
        Con.Open()  
    Catch ex As Exception  
    End Try  
    cmd = New OleDbCommand(sqlstr1, Con)  
    dr = cmd.ExecuteReader  
    Do While dr.Read  
        TextBox1.Text = dr("الفاتورة_رقم")  
    Loop  
    Con.Close()  
    dt1 = New OleDbDataAdapter(sqlstr1, Con)  
    dt1.Fill(DataSet1, "out")  
    DataGridView1.DataSource = DataSet1  
    DataGridView1.DataMember = "out"  
End If  
  
'===== الأسعار جمع =====  
Dim sum As Double  
For i = 0 To DataGridView1.RowCount - 1  
    sum += DataGridView1.Rows(i).Cells(1).Value  
Next  
Label17.Text = sum  
Label14.Text = Label17.Text * 10 / 100  
Label16.Text = Label14.Text + +Label17.Text  
===== للعميل الليبل بيانات سحب =====  
  
Try  
    Label19.DataBindings.Add("Text", DataSet1, "Out.العميل_رقم")  
Catch ex As Exception  
  
End Try  
Call clientdata()
```

انتهى الكود في الزر بالكود هذا

كود:

```
Call clientdata()
```

هذا معناه أن فيه حدث عمل مستقل سنستدعيه ونضعه لوحده تحت الكود للزر الأول ولكن طبعا بعد

كود:

```
End Sub
```

كود:

```
Private Sub clientdata()  
    '===== bayanat el3amel fe labels  
    sqlstr1 = "select ClientName As العميل_اسم,Phone AS التليفون_رقم,Mobile AS  
    رقم_الموبايل, Fax AS الفاكس_رقم,Address As العنوان From Client WHERE Code LIKE  
    '%" & Label19.Text & "%'"  
    Try  
        Con.Open()  
    Catch ex As Exception  
    End Try  
    cmd = New OleDbCommand(sqlstr1, Con)  
    dr = cmd.ExecuteReader  
    Do While dr.Read  
        Label18.Text = dr("العميل_اسم")  
        Label22.Text = dr("التليفون_رقم")  
        Label20.Text = dr("الفاكس_رقم")  
        Label21.Text = dr("العنوان")  
    Loop  
    Con.Close()  
    'dt1 = New OleDbDataAdapter(sqlstr1, Con)  
    dt1.Fill(DataSet1, "Client")  
End Sub
```

المهم هناك أمر آخر مهم ونستخدمه لحسب بيانات العميل
وسنضعها في حدث منفصل مثل التي قبلها قبل **End Class** وبعد **End Sub**

كود:

```
Private Sub getdata()  
    Con.Open()  
    Dim DataAdapter1 As New OleDbDataAdapter(sqlstr1, Con)  
    DataAdapter1.Fill(DataSet2, "Out")  
  
    Con.Close()  
End Sub
```

سنسمي زر الطباعة **btnPrint**

ونضع الكود التالي لنطبع الفاتورة :

كود:

```

Private Declare Auto Function BitBlt Lib "gdi32.dll" (ByVal hdcDest As IntPtr,
ByVal nXDest As Integer, ByVal nYDest As Integer, ByVal nWidth As Integer, ByVal
nHeight As Integer, ByVal hdcSrc As IntPtr, ByVal nXSrc As Integer, ByVal nYSrc
As Integer, ByVal dwRop As System.Int32) As Boolean
Private Const SRCCOPY As Integer = &HCC0020

Private Function GetFormImage() As Bitmap
    ' Get this form's Graphics object.
    Dim me_gr As Graphics = Me.CreateGraphics

    ' Make a Bitmap to hold the image.
    Dim bm As New Bitmap(Me.ClientSize.Width, Me.ClientSize.Height, me_gr)
    Dim bm_gr As Graphics = me_gr.FromImage(bm)
    Dim bm_hdc As IntPtr = bm_gr.GetHdc

    ' Get the form's hDC. We must do this after
    ' creating the new Bitmap, which uses me_gr.
    Dim me_hdc As IntPtr = me_gr.GetHdc

    ' BitBlt the form's image onto the Bitmap.
    BitBlt(bm_hdc, 0, 0, Me.ClientSize.Width, Me.ClientSize.Height, _
        me_hdc, 0, 0, SRCCOPY)
    me_gr.ReleaseHdc(me_hdc)
    bm_gr.ReleaseHdc(bm_hdc)

    ' Return the result.
    Return bm
End Function

' Variables used to print.
Private m_PrintBitmap As Bitmap
Private WithEvents m_PrintDocument As PrintDocument

' Print the picture.
Private Sub btnPrint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrint.Click
    ' Copy the form's image into a bitmap.
    m_PrintBitmap = GetFormImage()

    ' Make a PrintDocument and print.
    m_PrintDocument = New PrintDocument
    m_PrintDocument.Print()
End Sub

' Print the form image.
Private Sub m_PrintDocument_PrintPage(ByVal sender As Object, ByVal e As
System.Drawing.Printing.PrintPageEventArgs) Handles m_PrintDocument.PrintPage
    ' Draw the image centered.
    Dim x As Integer = e.MarginBounds.X + _
        (e.MarginBounds.Width - m_PrintBitmap.Width) / 2
    Dim y As Integer = e.MarginBounds.Y + _
        (e.MarginBounds.Height - m_PrintBitmap.Height) / 2
    e.Graphics.DrawImage(m_PrintBitmap, x, y)

    ' There's only one page.
    e.HasMorePages = False
End Sub

Private Sub btnPrintForm_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrint.Click
    ' Copy the form's image into a bitmap.
    m_PrintBitmap = GetFormImage()

    Make a PrintDocument and print.

```

```
m_PrintDocument = New PrintDocument
m_PrintDocument.Print()
End Sub
End Class
```

لا لتسجيل الدخول بعد الآن في الكريستال (الجزء الأول)

الكاتب : [ابن النيل](#)

[رابط المقال و النقاش حوله](#)

لكل الذين يستخدمون برنامج الكريستال ويعانون بشده من شاشة الدخول السؤال الأهم لماذا تظهر شاشة الدخول من حيث المبدأ . أنها تظهر لوجود جداول داخل التقرير لم يحصلوا على بيانات مما يجعلهم منزعين ويطالبون المستخدم بشاشة تسجيل الدخول للحصول على نصيبتهم من البيانات . هذه ببساطه هي المشكله و يوجد لها كثير من الحلول.

1-إرسال معلومات الاتصال إلى التقرير عند استدعائه ويتولى هو جلب البيانات بمعرفته ويعرضها دون أي تحكم منا فيها

2-إرسال البيانات إلى ملف XML ثم توجيه التقرير للملف ليجمع بياناته بنفسه

3-تعينة DataSet أو Data Table بالبيانات وتصبح هي مصدر البيانات بدل من قاعدة البيانات ويمكن طبعا بسهولة فائقة التحكم في البيانات الموجودة داخلهما مما يوفر عدد كبير جدا من التقارير الواجب تصميمها . وذلك لأنك تتحكم في البيانات المرسله فلا داعي مثلا لإلتقرير واحد مثلا لأسماء الموظفين واعرض فيه الموظفين حسب أي شرط قد يخطر في بالك كأنك صممته ١٠٠ مره أو أكثر (أنت وأفكارك)

الحل الأمثل من وجهة نظري التحكم في مصدر بيانات التقرير ألا وهو الحل رقم ٣ وهو ينقسم إلى ٣ حلول حسب الحالة

الأول:

في حالة الجداول المنفردة أو أن التقرير مصمم علي جملة استعمال واحده أو مبني علي جدول واحد الموضوع ببساطه هو تعينة Data Table بنفس جملة الاستعلام أو الجدول الذي بني عليه التقرير سواء بشرط أو لا أو لو حتى الناتج بدون بيانات مما يجعل أن جميع حقول التقرير حصلت علي بياناتها فتعرضها مباشرة دون الاهتمام بمعلومات الدخول إلى نفس المصدر الذي صمم عليه التقرير.

مثال:




```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Try
        Dim rep1 As New CrystalReport1
        Dim dt As New DataTable
        dt = DataSetm.manag
        dt.TableName = "manag"
        rep1.SetDataSource(dt)
        Form2.CrystalReportViewer1.ReportSource = rep1
        Form2.ShowDialog()
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub

```

حجز data table
وتعيينه بأي مصدر للبيانات أنت حر يارب تجيب في ورقه . ههه

خطوه مهمه ليتعرف التقرير على الجدول

إرسال التقرير للعرض

إضافة مصدر بيانات للتقرير

ابن النبل

عارض التقارير ...

ابن النبل

أسماء الإدارات	الكود	أسم الإداره
	1.000	إداره المبيعات
	2.000	إدارة المشروبات
	3.000	إداره الملاك

هذا شكل تقرير بأسماء الإدارات

الثاني:

أما في حالة أكثر من جدول فهنا الوضع مختلف حيث أن المطلوب هذه المرة بيانات مختلفة وكثيرة من أكثر من جدول إذا نظرية تعبئة Data Table الواحد لن تفلح هذه المرة الحل هو التصريح عن Data Set ثم تعبئتها بالجدول وكل جدول به

البيانات الخاصة به + يجب أن يكون كل جدول يحمل اسما وإلا كما يقال» كأنك يا بو زيد ما غزيت) »
 لأن التقرير سيبحث داخل ال Data SET عن أسم الجدول ليأخذ ما به من بيانات فلا يجده وهنا المشكلة

وهذا الوضع ينطبق أيضا علي الجداول الفرعية حيث أنها في الحقيقة مجرد تقرير بأكثر من جدول وهذا مثال علي ذلك.

```
Dim rep1 As New CrystalReport2
```

```
Dim ds As New DataSet
Dim dt1 As New DataSet.managDataTable
Dim dt2 As New DataSet.empDataTable
```

التصريح عن المتغيرات اللازمها وهم مهم هي
dataset
 وذلك لتعبئتها بالجدول وارسالها

```
For Each dr1 As DataRow In DataSet.manag.Rows
Dim row1 As DataSet.managRow
row1 = dt1.NewRow
With row1
.id = dr1("id").ToString
.managment = dr1("managment").ToString
End With
dt1.Rows.Add(row1)
Next
```

المهم هنا تعبئة
datatable

بالبيانات من اي مكان انت والي
 تحبه انا مثلا عيشه من داتا تابل
 جاهز

```
For Each dr2 As DataRow In DataSet.emp.Rows
Dim row2 As DataSet.empRow
row2 = dt2.NewRow
With row2
.id = dr2("id").ToString
.id_manag = dr2("id_manag").ToString
.name = dr2("name").ToString
.tel = dr2("tel").ToString
.title = dr2("title").ToString
End With
dt2.Rows.Add(row2)
Next
```



```
dt1.TableName = "manag"
dt2.TableName = "emp"
ds.Tables.Add(dt1)
ds.Tables.Add(dt2)
```

مهمه جدا فدونها لن يتعرف التقرير علي
 الجدول الذي صمم عليه

الحمد لله

```
rep1.SetDataSource(ds)
```

تعبئة الجداول قبل الارسال



```
Form2.CrystalReportViewer1.ReportSource = rep1
Form2.ShowDialog()
```



هذا تقرير اب واين وهو
تقرير الموظفين كلا حسب
ادارته

يعني ٢ تقرير رئيسي وهو
الادارات واخر فرعي وهو
الموظفين

أسماء الادارات والموظفين الموجودين فيها

الكود 1.000 أسم الاداره ادارة المشتريات

id	name	tel	title
8.000	محمد ١		
9.000	محمد ٢		

الكود 2.000 أسم الاداره ادارة المبيعات

id	name	tel	title
1.000	hamada	0	eg
2.000	hamada1	00	eg
3.000	hamada2	000	eg
4.000	hamada3	0000	eg
5.000	hamada4	00000	eg

هذا ليس الحل الوحيد ولكنه الأسهل والأسرع والأدق هناك أشياء أخرى يمكن حل المشكلة

وهذا مثال علي الحالتين في المرفقات

[حمل المثال من هنا](#)

معلومة ::: مشكلة كريستال ريبورت مع فيجوال استوديو 2010

الكاتب : **BADR MEDIA**

[رابط المقال و النقاش حوله](#)

لقد لاحظ البعض عدم وجود كريستال ريبورت مع فيجوال ستوديو ٢٠١٠ ولا وجود لفضاء الاسم **Name Space** واختفاء **Viewer**

لحل هذه المشكلة

قم بتحميل الملف من الشركة المطورة للكريستال ريبورت

<http://www.businessobjects.com/jump/...10/default.asp>

مايكروسوفت فصلت الكريستال لتجعلنا نتجه لتقاريرها ومن يريد الكريستال يحمله لا مانع

بعد تحميل الملفات ستجد أن أداة **Crystal Report** لم تظهر بعد

-من قائمة **Project** اختار خصائص المشروع

-اختار **Tab == compile**

-اختر من الأسفل **Advanced compile option**

-ستجد نوع المشروع متوافق مع **framework 4 client profile** قم بتغييره إلى **framework 4** سنظهر الأداة

وتصبح قادر علي تشغيل المشروع

والله الموفق لكل خير

قسم أخبار المنتدى

تم بحمد الله افتتاح قسمين جديدين للأسئلة والمقالات المخصصة ل

WPF, WCF, SilverLight , XAmI



مسابقة دائمة لأفضل مقال تقني



إصدار نظام الترشيح للأعضاء الخاص بمنتدى فيجوال بيسك العرب



إعلان نتائج المسابقة الأولى



تلميحات حول تقييم المواضيع والمشاركات المميزة



قسم المسابقات

نتائج مسابقة المقالات للربع الثاني من العام 2010



نتيجة ل طرح التصويت العام على مسابقة المقالات في تصويتين مستقلين الأول لعموم أعضاء المنتدى والثاني لأعضاء الإدارة والمشرفين وكبار الشخصيات

توضح الصورة التالية جدولاً كاملاً بالنتائج النهائية لمسابقة المقالات

العلامة النهائية	علامة تصويت الأعضاء	علامة تصويت المشرفين	تصويت الأعضاء	تصويت المشرفين	اسم المقال	
٣٠	١٠	٢٠	٢٥	٨	معلومة ::: البحث الصوتي مع الدالة ::: Difference & soundex ::: مهم جداً	١
٢٦	٨	١٨	٢١	٦	فهم كيفية بناء شبكة عصبية صناعية للتعرف على حرف أو رقم ..	٢
٢٣	٧	١٦	١٨	٣	التقارير الفرعية في الكريستال (فن وابداع) الجزء الثالث	٣
٢١	٥	١٦	١٥	٣	مقال: أفكار في الجرافكس الجزء الثالث	٤
٢١	٩	١٢	٢٤	١	درس بالصور ::: لكيفية التعامل مع الجداول المرابطة في قواعد البيانات :::	٥
٢٠	٤	١٦	١٣	٣	لا لتسجيل الدخول بعد الان في الكريستال (الجزء الاول)	٦
٢٠	٦	١٤	١٦	٢	شرح عمل تصويت بتقنية ajax	٧
١٥	٣	١٢	١١	١	اسلوب الأستدعاء الذاتي و مدخل الى تطبيقات أكثر كفاءة و عملية	٨
١٤	٢	١٢	٩	١	مقال : أفكار في الدوت نت OutAttribute Class	٩
١٣	١	١٢	٧	١	ال Delegates مع C#	١٠

المركز الأول

معلومة ::: البحث الصوتي مع الدالة ::: Difference & soundex :::
الكاتب : BADRMEDIA

المركز الثاني

فهم كيفية بناء شبكة عصبية صناعية للتعرف على حرف أو رقم ..
الكاتب : رجب النتشة

المركز الثالث

والمقال التقارير الفرعية في الكريستال (فن و ابداع) الجزء الثالث
الكاتب : ابن النيل

ألف مبروك لجميع الفائزين
مع تمنياتنا للجميع بالتوفيق

مع الشكر لكل من ساهم بأي شكل في مسابقة المقالات
ولجميع من صوت في كلا التصويتين

وبدء المشوار من جديد :



ويستطيع الجميع المشاركة في مسابقة المقالات بطرح مقالته في أحد الأقسام المخصصة
للمقالات وهي:

- قسم مقالات دوت نت العام

- قسم مقالات VB .net

- قسم مقالات #C

- قسم مقالات ADO .net

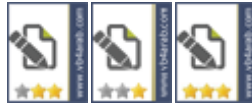
- قسم مقالات ASP .net

- قسم مقالات SQL Server

- قسم مقالات التقارير والطباعة في

- قسم مقالات WPF, WCF, SilverLight , Xml

وسيتم إجراء المسابقة في نهاية الربع وتكرر هذه المسابقة في كل ربع من السنة بشكل دوري نتمنى
للجميع المشاركة مع أطيب المنى لهم بالتوفيق .



قسم استراحة العدد

عن البرمجة نتكلم

الكاتب : **EGY Tiger**

السلام عليكم ورحمة الله وبركاته

أخواني المبرمجين، والمهتمين بعالم البرمجيات والتصميم، تحية طيبة مباركة وبعد:

فإنني بإذن الله بصدد ابتداء سلسلة جديدة من المقالات تحت عنوان (عن البرمجة نتكلم) وفيها سأضع عصير خبراتي البرمجية، وفيها بعض المقالات التي يمكن أن تهتم البعض، راجيا من الله أن تكون أولا خالصة لوجهه سبحانه وتعالى، وثانيا أن تكون نافعة لمن يقرأها بإذن الله،،،

وسأبدأ بمقال عن خطوات التفكير في إنشاء برنامج جديد، وهي تسبق خطوات إنشاء برنامج جديد بخطوة، وهي خطوة التفكير والتخطيط،،،

فإنشاء برنامج جديد - أي أصدقائي - ليس بقرار سهل، أو يمكن عمله كل يوم، بل يتطلب اتخاذ قرار بإنشاء برنامج جديد عدد من المنطلقات التي تساعدك في البدء بتنفيذ أولى خطوات عمل برنامج جديد،،،

وأولى هذه المنطلقات هو: الحالة النفسية الملائمة لتنفيذ القرار، فلا بد وأنت تتخذ القرار أن تتأكد أنك في حالة نفسية مناسبة، فلا تكون حالتك النفسية متحمسة جدا، وبالتالي سيكون أول مسمار في نعش البرنامج هو فتور الحماس التدريجي، ولا تكون نفسيتك متشائمة جدا وبالتالي ستجد صعوبة قبل تنفيذ أي خطوة أو فكرة جديدة،،،

وثاني هذه المنطلقات هو: لا تنفرد برأيك لأنك لن تستطيع السباحة ضد التيار، تكلمنا من قبل في أماكن مختلفة عن أن المبرمج يرتبط ببرنامج، وبأفكاره ارتباطا يشبه ارتباط الأب بابنه، وبالتالي فلن تكون وجهة نظرك متعادلة عند نقد هذا البرنامج، أو هذه الفكرة بل على العكس قد تكون متحمس جدا لفكرة تافهة، ولكن لأنك تنظر لها بمنظور قريب يجعل الفكرة في عينك كأنها درة الأفكار، وغير قابلة للنقد، وطالما واجهت كمبرمج الإحباط من العملاء عندما أجدهم لا يولون اهتماما لفكرتي - الرائعة - وعلى العكس يبذون تجاهلهم أو تافههم من الفكرة، ورأيي أن تناقش في الفكرة أحد أصدقاءك أو معارفك المقربين على ألا يكون مبرمج، أو مبرمج جديد في عالم البرمجة، وستجد أن أفكاره قد تختلف تماما - كمستخدم نهائي - عن أفكارك كمطور أو مستخدم غير نهائي.

وثالث هذه المنطلقات هو: لا تعمل على برنامجك إلا إذا كانت الظروف مواتية، فلا تفكر مثلا في وقت ما: لدي فراغ مقارب لنصف الساعة فلاقضيها في عمل كذا وكذا من برنامجك، وهذا خطأ جدا لسببين الأول أن هذه النصف ساعة لن تكون كافية لأن العمل تحت الضغط - نصف ساعة فقط - لا يشابه العمل في حرية- ملائمة للبرمجة-، والسبب الثاني هو اقتناعك بعد انقضاء النصف ساعة أنك (عملت شيئا) قد يتسبب في تكاسلك فيما بعد عن العمل في الوقت المناسب.

ورابع هذه المنطلقات هو: اقتناعك بالفكرة، فأنت كمبرمج علاقتك بالبرنامج تختلف عن علاقة الإنسان بالآلة، وتختلف عن علاقة المستخدم بأي برنامج، ولذلك فإنك إن لم تكن مقتنع بالفكرة فلن تستطيع تكوين التناغم المناسب للعمل مع برنامجك، ولعمل علاقة شرعية مع برنامجك صدد التطوير.

وخامس هذه المنطلقات هو: عمل خطوات للسير عليها، فكما تعلم أن عمل برنامج جديد أفضل وأسهل ١٠٠ مرة من التعديل في برنامج قديم، ولهذا فيجب عليك اتخاذ الخطوات المناسبة لتقلل من التعديلات المستقبلية قدر الإمكان، وهناك طريقة سهلة، وقوية، وهي الأفضل ويعتمدها العدد الأكبر من المبرمجين، وهي تصميم كامل البرنامج على هيئة نوافذ وقوائم فقط بدون كود، وبعد الانتهاء من التصميم تماما وبعد أن يكون البرنامج واضح أمامك سيكون عليك استخدام قدراتك الرفيعة كمبرمج ولتنفيذ هذه الرسومات البلهاء على هيئة سطور وأوامر.

وسادس هذه المنطلقات هو: لا تعمل برنامج جديد لتجربة شيء جديد، وبدلا من ذلك اعتمد على المكتبات **Libraries**، فيجب عليك تكوين مكتبة خاصة بك فيها العديد من الأفكار والتجارب الخاصة بك، وعند الحاجة لهذه الفكرة أو تلك استخدم المكتبة الجاهزة لتنفيذها.

المقال الثاني من السلسلة:

وبعد أن فكرنا في إنشاء برنامج، واتخذنا القرار بالبدء في إنشاء برنامج، علينا الانتقال للخطوة الثانية وهي (البدء في إنشاء برنامج جديد) ولكي نصل لهذه الخطوة لابد وأن نتأكد من أننا:

- فكرنا جيدا في فكرة البرنامج
- اقتنعنا بالفكرة وبأنها مجزية
- هيأنا الوضع المناسب للبرمجة
- وضعنا الجدول الزمني للعمل في البرنامج
- تأكدنا من أننا مبرمجين ولسنا هنا بالصدفة

والآن نبدأ في خطوات عمل البرنامج، وهنا ستجد أنني في بعض الخطوات أتفرع منها لطرق فرعية مختلفة لأنني لن أنشر رأيي فقط، ولكن سأنشر رأيي وتجارب الآخرين في الموضوع:

1-الجدول الزمني:

يجب عليك أن تصمم جدولا زمنيا للعمل في برنامجك أو فكرتك مع مراعاة أوقات فراغك وتفرغك في الفترات القادمة، وأن تلتزم بهذا الجدول بقدر الإمكان، مع العلم بأنك يجب أن تترك في الجدول مساحات فارغة للتعويض عن الوقت بدل الضائع، ووقت لما لم تكن تضعه في الحسبان، وذلك حتى لا يتراكم عليك الجدول.

2-الخيارات المبدئية:

هناك بعض الخيارات المبدئية التي يجب أن تضعها في البرنامج أو كما تسمى (معايير خاصة) لك ولهذا البرنامج فقط، وتلتزم بهذه الخيارات لأنها كما ستري مستقبلا ستوفر عليك الكثير من الوقت والجهد.

3-الخطوة الأولى:

هنا يمكن أن يحدث الاختلاف بين وجهات نظر، وجهة نظر المعظم، أو الأغلبية وهي التي أختلف معها والتي تقول إن أول خطوة هي تصميم النماذج Forms في برنامجك فارغة بدون أي أكواد وبعد الانتهاء تبدأ في وضع الكود، وهي طريقة ناجحة وتوفر الوقت والجهد ولكني لا أتوافق معها، أما طريقتي (وبعض المبرمجين) فهي أنك تقسم البرنامج إلى أجزاء وتبدأ في تطوير جزء جزء مثلا الجزء الخاص بالمخزون مثلا مكون من ١٠ نماذج وهي (كارت صنف - إذن تسوية - جرد فعلي -.... الخ) أبدأ في تصميمها حتى أنتهي منها واحدة واحدة ثم أنتقل للجزء التالي.

4-اتعب أكثر في الشاشات الهامة:

اجعل دائما في جدولك الزمني وقتا كبيرا للشاشات الهامة والأكثر استخداما في برنامجك فمثلا شاشة الكاشير في أي برنامج مبيعات تعتبر من أكثر الشاشات استخداما ولو لم تكن مريحة للعميل فسيفشل برنامجك حتى لو كانت باقي الشاشات سليمة مائة بالمائة.

5-أشرك أصدقاءك:

لا شك أن تفكير عقليين أفضل من تفكير عقل واحد، خاصة وإن كانت هذه العقول الجديدة محايدة (مستخدم نهائي مثلا) وذلك

بعد تصميم كل جزء من برنامجك وليس بعد تصميم البرنامج ككل.

6- اختر بدقة:

إن دقة الاختيار في البداية توفر عليك الكثير من الجهد في المستقبل، مثل اختيار قاعدة البيانات المناسبة لك، واختيار طرق الاتصال بها، وطرق نقل البيانات وعرضها كل ذلك يجب عليك بذل بعض الوقت في اختياره مع دراسة كل مميزاته وعيوبه.

7- لا تتسرع في العرض على العميل:

صدقني إن ظهور خطأ **BUG** واحد فقط أمام العميل يقلل من قيمة برنامجك كثيرا وبدون شعورك تجد أن العميل (يستتفه أو يستقل) مجهودك في البرنامج، بل ويناقشك في مدى تعبك عليه ومدى صعوبته.

8- لا تعد بما لا تستطيع أن توفي به:

لا تعد - نفسك أو العميل - بما لا تستطيع الوفاء به لأن ذلك يثقل عليك كثيرا.

9- اختبر برنامجك جيدا:

وهناك موضوع جيد عن اختبار البرامج قبل عرضها على العميل [هنا](#)

10- أمن برنامجك جيدا:

لا تنس وجود الكراكز في كل مكان،

وهكذا لم يبق لك في برنامجك إلا التسعير والتسويق، وهناك موضوع رائع عن التسعير [هنا](#)، أما التسويق فسيكون هو موضوع المقال الثالث والأخير في هذه السلسلة بإذن الله.

تستطيع أن تشاركنا في هذا الموضوع بالدخول على الرابط التالي : [اضغط هنا](#)

تعرف على العادات السيئة للمبرمجين (برمجياً) عادة سيئة للمبرمجين (أعظنا تجربتك)

تركي العسيري Microsoft MVP

سؤال (يوفر علي كتابة مقال) : كم مرة أعدت كتابة نفس الشيفرة المصدرية **Source Code** لانجاز مهمة معينة؟

كلنا نتغنى ليل نهار بمبدأ **إعادة الاستخدام Code Reusability** ولكن مشكلتنا القاتلة للوقت إننا نحن المبرمجين لا نرضى بما قدمته أناملنا، ونلولو دائماً: لو أني فعلت كذا كان أفضل، أو لو أني كتبتها بالطريقة تلك لكان أجمل، أو لو أني وضعت بعين اعتباري الحالة هذه لكان أكثر مرونة، لو.. لو.. لو..

لو تعود إلى ماضيك وتبدأ في محاسبة أصابعك، ستكتشف أنك قمت بتكرار كتابة عشرات الوظائف مئات المرات، وأنا أيضاً (رغم أنني شخص كثير الإزعاج حول مبدأ إعادة الاستخدام وكثير النصح له) إلا أنني أجد نفسي (أعيد) كتابة عشرات الوظائف في مشاريع مختلفة.. ما هو الحل؟

في الحقيقة لا يوجد حل شامل لهذه الظاهرة المشاعة بين المبرمجين، ولكن أقول سدّدوا وقاربوا!

بالنسبة لي، فمنذ صدور الإصدار **Visual Studio 2002** حتى هذه اللحظة، قمت ببناء إطار عمل **Framework** خاص بي وبمشاريعي، يحتوي على ما يقارب **200** فئة **Class** تقوم بعمل الوظائف المشتركة لمختلف المشاريع، كإدارة الخادم **Server** والاتصال به، قواعد البيانات **Database**، الأمان وإدارة المستخدمين **Security and User Management**، إدارة الإعدادات **Settings**، إدارة الملفات والمرفقات... **Attachments** الخ من المهام التي لا يخلو مشروع من مشاريعي إلا ويستخدمها.

صحيح أنني (في الغالب) اضطر إلى تعديل أجزاء كبيرة من إطار العمل الخاص بي، ولكن (بحق) وفرت علي لن أقول ساعات بل أياماً وشهوراً كنت سأضطر إلى (إعادة) كتابة شفراتي من جديد.

ولو سألتك سؤال بسيطاً، كم مرة قمت بكتابة كود لإدارة المستخدمين والحسابات والصلاحيات في مشاريعك المختلفة؟ ولماذا لا تقم بعمل إطار عمل **Framework** خاص بك تستخدمه في أكثر من مشروع؟

كما قلت ليس لدي حل شامل لهذه المشكلة فالكاتب مبتلى بهذه العادة السيئة **و لكن أقول لك بشكل نقاط:**

• عندما تكتب فئة **Class**، فكر دائماً في إمكانية إعادة استخدامها في مشروع آخر (حتى لو كنت تعتقد أنك لن تستخدمها لاحقاً)

• اجعل الفئة مغلقة قدر المستطاع **Encapsulated** ولا تعتمد على أي فئات أو كائنات أخرى.

• سأبتر أناملك إن علمت أنك قمت بكتابة ثوابت حرفية **String Constants** في شفراتك المصدرية، بمعنى آخر: لا تقم بتضمين الثوابت النصية في الكود **String Hardcoding**، وإنما عرفها على شكل ثوابت مسماة باستخدام الأمر **const** أو دعها تقرأ من ملفات مصادر **Resource Files**.

• اجعلها قابل للعمل بأمان مع مسارات التنفيذ المتعددة. **Thread Safe Classes**.

• احذر احذر احذر (كل الحذر) : لا تجعل هذه الفئة أبداً تظهر أي مخرجات **Outputs** على الشاشة سواء كانت رسائل **MessageBoxes** أو كود **HTML** فهذه الفئة قد نستخدمها لاحقاً في مشروع من أنواع مختلفة كـ **Windows Applications, WPF Application, ASP.NET Applications ...** الخ. وإنما اجعلها تعود بقيم أو ترمي استثناءات **Throwing Exceptions** إن حدث فيها أخطاء.

• المخرجات لا يشترط أن تكون على الشاشة فقط، بل حتى على شكل ملفات أو طابعات، لا تجعلها أبداً تعتمد على شيء من هذا القبيل (فقد يتم تنفيذها في نوع من المشاريع لا يعطي صلاحيات لها)، لذلك دعها تعود على شكل كائنات **Stream** إن

كنت تنوي حفظ ملفات، أو كائنات GDI+ للطباعة أو الرسم على الشاشة.

• إن كانت الفئة ستخرج مخرجات نصية **Strings** ، فحاول قدر الإمكان توفير طريقة للتحكم في شكلها **Format** ، فمثلا، لو كانت الفئة ستظهر قيم للتاريخ أو إعداد، أعط فرصة للمبرمج للتحكم في شكل هيئة المخرجات. **Output Format**.

• حتى لو كان هدف برنامجك عربي بشكل حصري، حاول فصل اللغة عن الكود، فلا تدري لعكك يوما من الأيام تود إضافة لغة جديدة.

• عندما تصمم فئة، تخيل دائما (وهذا ما افعله) إن المبرمج سيستخدمها من أنواع مشاريع مختلفة كـ **Windows** أو **Web**. ليس هذا فقط، بل ضع بعين الاعتبار أنها ستعمل في أجهزة الخادم **Server** أو العملاء **Clients**.

• تذكر أن بناء الفئة فن ولا يبدعه إلا القليل، أعط لنفسك وقتا للتفكير في الإبداع وانظر إلى الفئة من أبعاد وزوايا مختلفة قبل البدء في بنائها، (حدد هدفها) بشكل مفصل من البداية، واجعلها (محصورة) في نطاق هذا الهدف، وللمتطلبات الأخرى (قد تحتاج إلى بناء فئة أخرى أو تطبيق مبدأ الوراثة. **Inheritance**).

هذا موضوع للنقاش (وليس مقالا حول إعادة الاستخدام) لذلك يسعدني أن يقوم كل شخص منكم بعرض تجاربه وماذا استطاع تقديمه لقتل هذه العادة السيئة بيننا (نحن معشر المبرمجين)

تركي

((Vb4Arab)) ((Vb4Arab))

Mohammed Asaad

السلام عليكم ورحمة الله وبركاته
بما أنك فتحت باب النقاش حول هذه القضية سأتكلم بما كان يحصل معي إلى اليوم لم أقم بإنتاج مشروع واحد لأن " شبح عدم الرضا " بما أنجزت يلاحقني دوما قدمت بعض النصائح ونصائح ممتازة أيضا عدم التخطيط الجيد والعمل العفوي يزيد من هذا التوتر لي عودة بعد أن أقوم بتجميع فكرة حول الموضوع بشكل موسع إن شاء الله

((Vb4Arab)) ((Vb4Arab))

تركي العسيري **Microsoft MVP**

من أجمل ما قرأت! ما شاء الله عليك.. شيء جميل جدا، وهذا بالضبط ما يفترض أن نكون عليه، وإلا لن نتقدم البشرية.

بخصوص عبارتك، صدقتي كلنا نشاركك في هذا الإحساس ولكن عليك أن تأخذ بشكل ايجابي ((وليس سلبي)) ليكون دافع لك للتطوير أكثر وأكثر!

ولكن أريد أن أقول لك شيئا، هناك فكرة في عالم البرامج اسمها الإصدار **Version** ، فعندما تبدأ بتصميم مشروع عليك تعريف المزايا **Features** التي فيه، وعندما تبدأ بالعمل **قف ولا تتعدى حدود هذه المزايا أبدا** ، استمر في عملك **والا لن تنتهي أبدا من إخراج المشروع** ! يعني باختصار، أجل الأفكار الإضافية إلى الإصدار القادم.

دعنا نعود إلا الوراء مع برنامج **Microsoft Word** ، تخيل لو أن **Microsoft** قالت: امممم نرى أن علينا تأجيل طرح الإصدار وإضافة مزايا إضافية، لو سارت على هذا النهج لكننا (حتى هذه اللحظة) لم نتعرف على منتج اسمه **Microsoft Word!** فالأفكار ليس لها نهاية وليس لها حدود.

تعريف المزايا **Defining Features** أمر مهم جدا لتحديد مجال المشروع **Project Scope** ، وعلينا كمطورون الانضباط والتركيز عليه دون تشتيت أفكارنا، دعني أصدمك من مصدر موثوق في **Microsoft** ، الكثير من المزايا كانت ستظهر مع **Visual Studio 2003** ، ولكن **Microsoft** أجلتها حتى الإصدار **2008!**

أما لو كل شخص قاس عمله وإنتاجه على ملكوت فكره وبحار الهاماته، فلن ينتج ولن يخرج أي شيء! قدم ما تستطيع الآن، والبقية تأتي بإذن الله.

(((Vb4Arab))) (((Vb4Arab)))

Islam Ibrahim

شكراً لك أستاذي الفاضل ، هذا هو الأسلوب الذي لطالما حلمت به ،نصائح من ذهب وأنا من عشاق التوسيع **Extensibility** وهذا الأسلوب هو المعتمد في التطبيقات المتعددة الطبقات وهو الأفضل طبعاً، لأنه يسهل مهام صيانة الكود وتطويره.

هناك أمور أخرى يهملها المطورون وهي التوثيق **XML Documenting** في الكود ، ولا يدركون أهميته وحتى إن كانوا يعتمدون التعليقات في أكوادهم فهم يستخدمون الطريقة التقليدية .

(((Vb4Arab))) (((Vb4Arab)))

Marwan

شكرا أستاذ تركي على طرح هذا الموضوع المهم

أود أن أشارك ببعض ما لدي بخصوص **Code reusibility**

1-عندما تقوم ببناء مكتباتك الخاصة لا تجعلها تعتمد بشكل مباشر على مكتبات ثاوية **Third party libraries** بل قم بتغليف الوظائف **Encapsulte** التي تستخدمها من هذا المكتبات بواسطة فئات من إنتاجك أنت و اجعل شيفرتك تتعامل مع هذه الواجهة -الفئات فإذا قمت مستقبلاً بالاستغناء عن هذا المكتبات الثانوية لن تضطر إلى تتبع كل الأماكن التي استخدمتها فيها لتقوم بتغيير اللازم و لكن بدل ذلك ستتوجه مباشرة إلى الفئات التي قامت بتغليف هذه الوظائف و تغير ما يجب تغييره.

2-قدم لنا الدوت نت ٣,٥ ميزة جميلة جدا لتسهيل إعادة استخدام الشيفرة و هي الطرق المضافة **Extension Methods** مثلا إذا كان لدينا فئة اسمها **StringUtilities** نريد أن نضع فيها طرق نستخدمها بكثرة للتعامل مع البيانات النصية. قبل الدوت نت ٣,٥ الطريقة المثلى هي جعل هذه الطرق **Static** حتى يمكن استدعائها بسهولة.

مثال:

كود:

```
public class StringUtilities
{
    public static bool IsWebURL(string target)
    {
        Regex WebUrlExpression = new Regex(@"(http|https)://([\w-]+\.)+[\w-]+(/[\w- ./?%&=]*)?", RegexOptions.Singleline | RegexOptions.Compiled);
        return !string.IsNullOrEmpty(target) &&
        WebUrlExpression.IsMatch(target);
    }
}
```

الطريقة تقوم بالتأكد أن النص هو عبارة عن عنوان ويب بواسطة **Regular expression**

عندما نريد أن نستخدم هذه الطريقة نكتب ببساطة:

كود:

```
StringUtilities.IsWebURL("http://vb4arab.com");
```

لكن أن قمن بكتابتها كطريقة إضافية فسيكون الكود أوضح:

كود:

```
public static class StringUtilities
{
    public static bool sWebURL(this string target)
    {
        Regex WebUrlExpression = new Regex(@"(http|https)://([\w-]+\.)+[\w-
]+(/[ \w- ./?%&=]*)?", RegexOptions.Singleline | RegexOptions.Compiled);
        return !string.IsNullOrEmpty(target) &&
WebUrlExpression.IsMatch(target);
    }
}
```

الآن يمكن أن نستخدمها بهذا الشكل:

كود:

```
"http://vb4arab.com".IsWebURL()
```

المثال مأخوذ من مشروع [KIGG](#)

لذلك أنصحك دائماً أن تقوم بتحويل أي طريقة **Method** مفيدة إلى **Extension Method** و تضيفها إلى مكتبة الخاصة حتى يسهل عليك استخدامها فيما بعد للمزيد عن الطرق المضافة:

<http://vb4arab.com/vb/showthread.php...ension+methods>
<http://vb4arab.com/vb/showthread.php...ension+methods>

(((Vb4Arab))) (((Vb4Arab)))

أحمد جمال

ربما لا أواجه نفس المشكلة بنفس الصورة ، يعني في العادة أغلب مشاريعي هي كوكتيل من مشاريع سابقة بطريقة النسخ واللصق المباشر.

المشكلة التي أواجهها تكمن نوعاً ما في عدم تقسيم الفئات بشكل يسمح لي لاحقاً بإضافة فئة أو **dll** كما هو البدء بالاستخدام مباشرة ، عادة ما تحتوي فئاتي على أقل أو أكثر من المطلوب ، لذا ومع كل مشروع جديد أتذكر أين كتبت كود هذه المهمة لأقوم بفتح المشروع ونسخها منها مباشرة ، أبسط أمثلة ذلك أكواد قواعد البيانات التي تنسخ كما هي عادة.

أما لماذا لا أقوم بوضعها في فئات ، للأسف هذه المشكلة مردها إلى أنه قبل البداية لم أصمم البرنامج بصورة صحيحة أصلاً ، وبعد انتهائه لا أجد عادة الوقت الكافي لإعادة استخراج مجموعة من الأكواد المفيدة وتجميعها في فئات ، لذا فقط اعتمد على ذاكرتي في تذكر أين كتبت هذا الكود من قبل . هذا وإن كان كافياً لي نظرياً في الفترة الحالية - حتى موعد تدهور خلايا الذاكرة - لكنه لا يساعد فعلياً أي شخص يعمل معي في أي مشروع وكثير من أكواده يضطر لكتابتها رغم أنها لدي بصورة متكاملة.

الحل لا أعرف ، ولكن للأسف واقع العمل أحياناً يفرض عليك اختيارات غير مناسبة تماماً .

(((Vb4Arab))) (((Vb4Arab)))

محمد الناشرى

الحاجة أم الاختراع

تماماً مثل ما يفعل الأستاذ / احمد جمال
نوعية المشاريع التي اعلم عليها بسيطة جدا وهي في الغالب لأداء بعض المهام التي احتاجها في العمل وغالبها يتعامل مع قواعد البيانات أو الملفات...

لذا ارجع كثيراً لمشاريعي السابقة واخذ منها ما احتاج إليه (لتسريع العملية فقط) ولا ننسى **Add Existing item** هناك نقطة أخرى بالنسبة للويب وهي الـ **userControl** مفيدة جدا .. واستخدمها بكثرة وهي واحدة من الحلول. هذا الكلام لايعطينا من التفكير جدياً (التفكير الجدي يعنى العمل) بمبدأ إعادة الاستخدام..
ولكن
تبقى الحاجة أم الاختراع

(((Vb4Arab))) (((Vb4Arab)))
The X man

إلى اللحظة .. أستعمل طريقة النسخ واللصق من مشاريع أخرى .. ولكن مع مرور الأيام ! رأيت (وأقول رأيت فعلاً) أنني استخدمت عدة أكواد مرات كثيرة داخل البرنامج نفسه ، ورأيت أنني أستعملها نفسها (أو مع بعض التعديل) في برامج أخرى . ولكني أتحدث مع نفسي وأقول : لو أنك جعلت من هذه الطريقة (أو من هذه الأوامر) طريقة عامة أو أوامر عامة ، تنشئها مرة واحدة وتستخدمها عدة مرات ، لكنت وفرت على نفسك الكثير من الوقت .
وأبسط مثال عليها : أضفت أداة **ComboBox** إلى النافذة وكتبت أوامر بحيث تسمح بكتابة الأرقام فقط ، ومن الأرقام : مسموح له أن يكتب خانتين فقط ، وفي الخانتين مسموح له أن يكون الرقم بين ١ إلى ٣١
استخدمت عدد ٨ **ComboBox** في البرنامج ، ومن كثر الذكاء أعدت كتابة الأكواد ٨ مرات!!!!
مش لو قمت بإنشاء أداة على شكل **ocx** أو ملف **dll** وكتبت بها هذه الأكواد مرة واحدة ، وكل ما احتجتها قمت بعملية (سحب وإفلات) للأداة من صندوق الأدوات لكان أفضل؟؟؟
(هذا الكلام كله صار أول ما تعلمت البرمجة ، ما كان عندي أي خلفية عن كيفية اختصار الجهد والوقت)
هذه هي تجربتي مع (تكرار كتابة نفس الشيفرة المصدرية)

(((Vb4Arab))) (((Vb4Arab)))
DreamNet

معظمنا إن لم أقل نسبة كبيرة جدا منا نحن مبرمجو ال **vb** تراكمت لدينا عادات سيئة ولدها لدينا العمل تحت إطار ضيق وبإمكانيات محدودة في السابق وبمفاهيم أخذنا وقتاً طويلاً لنستوعبها
لذلك لا زلنا نحمل هذه العادات التي تؤثر دائماً على نوعية العمل الذي نقوم به

في البدء لم نكن نعطي أدنى اهتمام حتى للمتغيرات وأنواعها ومع كبر المشاريع وكثرة المشاكل يتولد لدينا شعور بعدم الثقة والرضي في نفس الوقت وهذا الشعور العشوائي يدفع الكل لعدم التفكير في تصحيح الأخطاء بل بتغيير طرق التفكير

والأهم هنا هو طريقة تفكير مبرمجي ال **vb** فنحن لم نتعود إلى وقت قريب على التعامل مع الفئات إلا عند الدخول إلى الدوت نت والخطأ الفادح هو أننا نستعمل الدوت نت بعقلية الإصدار القديم والقليل من تحرر من هذا المشكل

النقطة الأهم هنا هي تنمية عقلية المبرمج بالعمل الجماعي وستجد أن أغلبنا لم يشارك فعليا في عمل جماعي يصقل فيه مهاراته ويتخلص به من عاداته السيئة لان العمل الجماعي هو الحل الأنسب لكل مبرمج للتفكير بعقلية منظمة وليست عشوائية

كل المشاكل التي نعاني منها سببها الوحيد هو العشوائية في التفكير

(((Vb4Arab))) (((Vb4Arab)))

Mohammed Asaad

كنت سأطرق لهذه النقطة لكن خشيت أن أكون خرجت بها عن إطار الموضوع على كل: كل ما تفضلت به عن العمل الجماعي صحيح ويدعو للتوحيد والتنظيم بالعمل بعيدا عن العفوية والعشوائية أيضا تشكيل فريق جماعي يلزم جميع أعضاء الفريق على المضي على نحو واحد بعد تخطيط جيد للعمل القادم والاستفادة من تجارب سابقة

أضف لذلك:

العمل الفردي 99% لا يصيب بقدر ما يصيب **العمل الجماعي** بمعنى أن العمل الفردي كثير ما يتصف بالتردد وعدم الثبات وهذا ما يدعو لإعادة صياغة البرنامج أكثر من مرة وهذا ما كان يحدث معي كل فترة أنظر للبرنامج وأرى أنه ليس على قدر من التنظيم فأعيد صياغته مرة أخرى ومؤخرا تطرقت لمشاريع أخرى بغية اكتساب مهارات تنظيمية إضافة لنصائح من هنا وهناك شعرت بأن هناك نوع من التنظيم

بالنسبة لأمر صنع مكتبة خاصة وجعل الفئات مغلقة بقدر الإمكان أعجبتني الفكرة لكن لا أشعر أنني سأعطي جميع الاستثناءات والأخطاء التي قد ترد

(((Vb4Arab))) (((Vb4Arab)))

Islam Ibrahim

هناك نقطة أخرى أرغب في إثارتها وهي التوافقية مع باقي لغات البرمجة، وأعني بذلك، أنني أحيانا قد أعرض مكتباتي المبرمجة بفيجوال بيسك على ميرمج سي شارب، لذلك لا بد من أن تكون مكتبتي متوافقة مع معايير **CLS** و **CTS** أي تتجنب استخدام كود غير متوافق. **Non-CLS Compliant Code**

وعليه يجب الابتعاد قدر الإمكان عن استخدام المكتبات والفئات الموجودة داخل المكتبة **Microsoft.VisualBasic.dll** (بالنسبة للفيجوال بيسك) واستبدالها بما يكافئها داخل مكتبات الفريم وورك.

يعني مثلاً بدل استخدام **vbCrLf** نستخدم **Environment.NewLine** فهي حسب ما قرأت أكثر توافقية حتى في أنظمة التشغيل غير الويندوز.

وحتى عند تحويل الكود بين أكثر من لغة برمجة ستكون العملية أكثر مرونة.

كخلاصة قول لا بد من الاستقلالية قدر الإمكان والتحرر من قيود اللغة التي أبرمج بها، حتى تكون مكتباتي أكثر شمولية وأكبر قابلية للاعتماد من قبل لغات برمجة أخرى.

(((Vb4Arab))) (((Vb4Arab)))

samerselo

أنا قمت أكثر من مرة ببناء مكتبة خاصة بي تحتوي على كثير من الأكواد المكررة ولكن فكرتي تفشل في الغالب ربما بسبب أنني ميرمج هاوي أقوم ببرمجة مشاريعي الخاصة عندما أحتاجها وقد تمضي شهور أو سنوات قبل أن أقدم على كتابة مشروع جديد وفكرة المشروع تأخذ مني وقتاً طويلاً حتى تبدأ بالتنفيذ ربما لأنني أحب أن أنجز الأمور بأفضل صورة ممكنة وربما بسبب الوقت المخصص للبرمجة يقل يوماً بعد يوم على كل حال مازالت محاولاتي لبناء مكتبة خاصة بي موجودة **كاأكواد مصدرية** أستخدمها بطريقة النسخ واللصق غالباً وأنا ممن لا يحبون كتابة الكود أكثر من مرة اكتبه وأتقنه من أول مرة واجعله عاماً قدر الإمكان بحيث لا تضطر للتعديل عليه لاحقاً إن أمكن ذلك وخرزته في مكتبة واستخدمه لاحقاً متى ما احتجت إليه هذه هي نظريتي بخصوص الموضوع

(((Vb4Arab))) (((Vb4Arab)))

MrVisualBasic

لطالما عانيت من نفس المشكلة
تجربتي كانت مع لغة **php** حيث كنت أكتب عشرات الأسطر البرمجية و أكررها (خاصة شيفرات التعامل مع قواعد البيانات)
لكن بعدها فكرت جيدا و قلت لما لا أبني فنة خاصة بالتعامل مع قواعد البيانات و الحمد لله قمت ببنائها و أنا الآن استعملها في
أكثر من مشروع
و خلاصة القول إن هذه الفنة:

- 1- وفرت الكثير من الوقت و الجهد
- 2- الشيفرة المصدرية أصبحت واضحة و مختصرة في نفس الوقت

(((Vb4Arab))) (((Vb4Arab)))

أبو عابد

نعم
لدي مجموعة من الفئات بنيتها منذ أكثر من سنة وهي تعل في مشاريعي القديمة ، ولكن المشكلة هي تراحم الأفكار الجديدة
لدرجة لا تستطيع معها أن تطور شيئا!!!

(((Vb4Arab))) (((Vb4Arab)))

soso.cis

بسم الله الرحمن الرحيم

من أكثر العادات السيئة للتي أمر بها هي الشعور بالضيق وسط هذا الكم الهائل من المعلومات التي اجهلها لا ادري إن كانت
عاده سيئة أم هذا لكوني مبتدئه في طور التعليم ولكن وكما قال البعض حتى يكتمل المشروع يجب التقاضي عن المزايا والعمل
على الأساسيات للمشروع وإلا سيبقى العمل في الظلام ينتظر لحظه ميلاده

ليس لدي الكثير لأضيفه لكم و كلن أحببت أن أشارككم الحوار عل الحديث يزيح عني بعض الضغوط

(((Vb4Arab))) (((Vb4Arab)))

محمد أسامة جندي

وضعت يدك على الجرح أخي تركي

من تجربتي أجد هناك نوعين من المبرمجين بشكل عام:

1- النوع الأول أكثر ما يهمله هو أن يعمل البرنامج ... يتحايل على المشاكل لحلها بحيث يصل لهدفه بأسرع الطرق فكل ما
يهمله هو الكم (إنجاز برنامج أسرع وقت ممكن) وهذا النوع يميزه الالتزام بالمواعيد وإنجاز عدد لا بأس به من البرامج
ولكن يعيبه التحديث والتطوير فهو يخاف من التقنيات الجديدة ولا يحب تعلمها إلا عندما يجبر عليها لأنها تخفض من إنتاجيته
(أول فترة)

2- النوع الثاني أكثر ما يهمله هو أن يعمل البرنامج بأفضل شكل ممكن وأفضل أداء فتراه يصرف جزء كبير من وقته في تعلم

التقنيات الجديدة ومحاولة تطبيقها في برنامجه وبيحث عن أفضل الحلول لمشاكله وليس أسرعها حتى يحصل على أفضل أداء ويحاول إضافة كل ما يخطر في باله للبرنامج وهذا النوع يميزه أنه يتعلم التقنيات الجديدة دانما ولكن على حساب الإنتاجية فلا ترى له برامج في السوق لأنه يقضي وقته في تطوير برامجه وليس إنهاؤها.

الحل كما تفضل أخونا تركي في:

1- ضبط مميزات برنامجك قبل كتابة أي حرف في الكود ثم عدم تطوير أي جزئية مهما صغرت إلا بعد إنزال النسخة إلى السوق أو الزبون.

2- تطوير مكتبة خاصة بك تحوي جميع الخدمات والأكواد التي تستخدمها وإضافة التوثيق لهذه المكتبة حتى تستخدمها بشكل جيد.

3- محاولة مجارات التقنيات الجديدة والإطلاع عليها كأقل تقدير حتى تتعلمها إن احتجت إليها (مثال: في العادة عند البحث ضمن قائمة أو غيره نحتاج إلى حلقة for وأن نفحص قيم القائمة ضمن هذه الحلقة ولكن مع Lambda أصبح كل ما تحتاجه سطر واحد للبحث في قائمة - تخيل إنتاجية من يستخدم ومن لا يستخدم هذه التقنية)

(((Vb4Arab))) (((Vb4Arab)))

تركي العسيري

ما أود أن أضيفه هنا على نقاطي في مشاركتي الأولى قليلا من الأكواد حتى تتضح الصورة أكثر، فقلت أن عليك أن تجعل الفئة قابلة لإعادة الاستخدام (Reusable قدر المستطاع) في أكثر من مشروع، وأسهل مثال (للمبتدئين) أعرضه هي الفئة الجميلة DBHelper والتي (كما يتضح من اسمها) تسهل حياتك البرمجية عند التعامل مع قواعد البيانات، فمنذ سنين لم أستخدم أبرز كائنات ADO.NET كConnection ، أو Command ... غيرها) بل حتى قد نسيت طرق استخدامها، فعلاقتي مع قواعد البيانات أصبحت تتم عن طريق الفئة DBHelper العزيزة على قلبي (استخدمتها في عشرات المشاريع ومئات الأمثلة في السنين الخالية)، حيث تبدأ حياة هذه الفئة بإنشاء كائن منها وإرسال نص الاتصال Connection String مع تحديد نوع المزود:

كود:

```
Dim dbAccess As New DBHelper("connection string", _
    DataAccessProviderENM.SQLServer)
```

والآن – وبمجرد كتابة سطر واحد فقط أستطيع أن أفعل كل شيء مع قواعد البيانات، فمثلا لو أردت استخدام عبارة SELECT، فيمكنني استدعاء الطريقة (GetDataSet()) بكل سهولة:

كود:

```
ds = dbAccess.GetDataSet ("SELECT * FROM [TableName]")
```

مع العلم أن السطر السابق سيقطع الاتصال مع قواعد البيانات، فأغلب طرق DBHelper تقوم بقطع الاتصال Disconnect تلقائيا دون تدخل المبرمج، مما يزيل القلق على مصادر النظام. System Resources

وحتى إن كنت من المتورطين في الوضع المتصل Connected Mode ، فهناك إمكانية للحصول على SqlDataReader أيضا:

كود:

```
dr = dbAccess.GetDataReader ("SELECT * FROM [TableName]")
...
dr.Close()
```

وإن أردت تنفيذ جمل استعلامية (لا تعود بقيم)، فلن تجد أسرع من الطريقة: ExecuteSqlCommand()

كود:

```
dbAccess.ExecuteNonQuery("UPDATE [TableName] SET [Field] = 'Value'")
```

المزيد أيضا، قمت بإعادة تعريف **Overload** الطريقة السابقة لتقبل أكثر من جملة استعمال في نفس الوقت، لتقوم بتنفيذها على شكل عملية واحدة: **Singe Transaction**

كود:

```
dbAccess.ExecuteNonQuery("UPDATE [TableName] SET [Field] = 'Value'", _  
"UPDATE [TableName2] SET [Field] = 'Value'")
```

ليس هذا فقط، فالفئة **DBHelper** مرنة جدا وتوفر دعم إضافي للمبرمج لوضع لمساته الخاصة، فمثلا يستطيع المبرمج في أي وقت من الحصول على كائن اتصال **Connection** جديد بسطر واحد فقط:

كود:

```
Dim cn As SqlConnection  
cn = CType(dbAccess.CreateConnectionObject(), SqlConnection)
```

ويمكنه أيضا من الحصول على كائن الأوامر **Command** بنفس الأسلوب:

كود:

```
Dim cmd As SqlCommand  
cmd = CType(dbAccess.CreateCommandObject(), SqlCommand)
```

عودة إلى الطريقة (**GetDataSet**) ، هذه الطريق تعود بكائن جديد من نوع **DataSet** ، ولكن بما أنني لا أعرف طريقة تفكير المبرمج وما هي احتياجاته (فقد يكون لديه كائن **DataSet** جاهز يريد تعبئته)، وفرت له إمكانية إرسال كائن **DataSet** حتى أقوم بتعبئته عوضا عن إنشاء كائن جديد:

كود:

```
Dim ds As New DataSet  
...  
...  
dbAccess.GetDataSet ("SELECT * FROM [TableName]", ds)
```

وإن كان المبرمج يعتمد على كائنات **Command Parameters** بشكل مكثف، فيمكنه إرسال كائن **Command** أيضا:

كود:

```
Dim cmd As New SqlCommand  
...  
...  
ds = dbAccess.GetDataSet (cmd)
```

المزيد أيضا، الطريقة (**GetPagedSQL**) مثيرة جدا، فهي تقوم بتوليد جمل استعمال **SELECT** التي تطبق مبدأ التصفح **Paging** عرض النتائج الطويلة على صفحات متفرقة:

كود:

```
dbAccess.GetPagesSQL(1, 10, "ID", "TableName", _  
"*", "[CDate] Desc")
```

أخيراً، تذكر أن هذه الفئة ليست محصورة على **SQLServer** ، بل يمكنها دعم مزود **OLEProvider** أيضاً (لعشاق) **MS-Access**

كود:

```
Dim dbAccess As New DBHelper("connection string", _
    DataAccessProviderENM.OLEDBProvider)
```

كما رأيتم يا أصدقائي الأعضاء، بمتوسط عدد (٢-٣) سطور برمجية هي كل ما تحتاجه للاتصال والتعامل مع قواعد البيانات! ولو تدقق في كود الفئة من الداخل، ستجد أنها مغلفة **Encapsulated** بقوة ولا تعتمد على كائنات خاصة أخرى، مما يعني سهولة إعادة استخدامها في أي مشروع من مشاريع **Visual Studio**.

بعيدا عن الفئة **DBHelper** ، فقد قلت أنني سأبتر أناملك إن قمت بضمين ثوابت نصية: **Hardcoding Strings**

كود:

```
Class TestClass
    Function TestMethod() As String
        Return "Hello World"
    End Function
End Class
```

استخدم الثوابت المسماة باستخدام: **Const**

كود:

```
Class TestClass
    Const HELLO_MSG As String = "Hello World"
    Function TestMethod() As String
        Return HELLO_MSG
    End Function
End Class
```

وبخصوص الأخطاء، فلا تظهر شيئا كما هنا:

كود:

```
Const MAX_VALUE As Integer = 100
Sub TestMethod(ByVal v As Integer)
    If v > MAX_VALUE Then
        MsgBox("Value is greater than MAX")
    End If
End Sub
```

بل اجعلها ترمي استثناءات **Exceptions** ، لنعطي المبرمج حرية في كيفية التصرف عند حدوث أخطاء:

كود:

```
Const MAX_VALUE As Integer = 100
Sub TestMethod(ByVal v As Integer)
    If v > MAX_VALUE Then
        Throw New Exception("Value is greater than MAX")
    End If
End Sub
```

بالنسبة للمخرجات، سواء ملفات أو كائنات رسم، فلا تجعل الفئة تنشئ سياق رسم جديد خاص بها، بل أعطي فرصة للمبرمج من تحديد سياق الرسم، فنحن لا ندري أين يريد عرض المخرجات (على النافذة، على أداة، حفظها كصورة في ملف)

كود:

```
Sub New(ByVal gr As Graphic)
    ...
End Sub
```

عندما تعرف طرق **Methods** في فئة، ويوجد بارامترات **Parameters** مشتركة:

كود:

```
Dim obj As New Class
obj.Method(True, 1, "string")
obj.Method2(True, 1, "string")
obj.Method3(True, 1, "string")
```

فيمكن الاعتماد على الخصائص **Properties** حتى تسهل الحياة على المبرمج:

كود:

```
Dim obj As New Class
obj.Property1 = True
obj.Property2 = 1
obj.Property3 = "string"

obj.Method()
obj.Method2()
obj.Method3()
```

وعلى ذكر الخصائص، فكن ذكياً قدر الإمكان، فوفر على المبرمج تحديد العمر عن طريق تعريف خاصية للقراءة فقط **ReadOnly** تعتمد على خاصية الميلاد:

كود:

```
Class Person
    ReadOnly Property Age() As Integer
        Get
            Return (Now - Me.BirthDate).Years
        End Get
    End Property
    ...
End Class
```

ولتحقيق أكثر لمبدأ إعادة الاستخدام، فالأحداث **Events** جدا مثيرة، فعندما يتم وصول رسالة جديدة إلى برنامجك، دع هذه الفئة تفجر حدث يدركه المبرمج لاحقاً:

كود:

```
Class MessgeManager
    Event NewMsg()
    Sub Checking()
        RaiseEvent NewMsg()
    End Sub
    ...
End Class
```

الأمثلة كثير جدا جدا (لاحظوا أنني لم اعرض أمثلة لاستخدام الواجهات **Interfaces** ولا الوراثة **Inheritance** ولا أدوات التحكم)، ولكن تذكر كلمتي السابقة، إن بناء الفئات فن لا يبدع فيه إلا قليل من المبرمجين، كلما أبدعت في بناء فئاتك وجعلتها **ذكية** و **مرنة** أكثر، كلما وفرت على نفسك وقت وجهد أكثر.

أخيرا، بالنسبة لمن يسأل عن بناء إطار عمل خاص **Framework** وطريقة بناء المكتبات **Libraries** ، فهو موضوع كبير جدا ومتشعب، وفي الحقيقة هو نفس سؤال: كيف أكتب برنامج!! نعم، فقد أضدمك إن قلت لك أن تطوير المكتبة اعقد وأصعب من تطوير برنامج، فالبرنامج تكتبه بأي طريقة المهم أنه يعمل (هدفك هو المستخدم النهائي)، بينما المكتبة هدفك هو مبرمج لا يرضى إلا بعمل دقيق وفعال ومرن وقابل للتحكم وتحت السيطرة) أتمنى أن تكون قد فهمت ما أقصده

بالنسبة لإطار العمل **Framework** الخاص بي، فهو معقد جدا (٢٠٠ فنة)، وأجريت عليه (ولا أزال) تعديلات كثيرة جدا، ولا يمكن التحدث عنه، خذ مثلا هذه صورة مبسطة لأبرز الفئات لمجالات أسماء: **Namespaces**



تذكر أن هذه صورة مبسطة جدا للهرم الكائني **Object Hierarchy** الخاص بي، وهناك مئات الفئات التي لم أتمكن من عرضها، فهدفي إعطائك فكرة مبسطة عن مدى جدية وتعقيد الموضوع!

معذرة على الإطالة في ردي، فمنذ فترة طويلة لم أكتب مقالات تقنية، ويبدو أن أناملي اشتاقت للكتابة

الملفات المرفقة : [DBHelper Class.zip](#)

(((Vb4Arab))) (((Vb4Arab)))
النهائي

عندما بدأت البرمجة - على يدك - بعد قراءاتي لكتاب **Visual Basic** للجميع ، نحو برمجة كائنية التوجه ، كان موضوع البرمجة الكائنية موضوعا كئيبا وكرهيا بالنسبة لي ، حتى إنني عندما كنت أقوم بتحميل الأمثلة الجاهزة من المواقع الأجنبية ، كنت أتجنب تلك التي تعرض الأفكار على شكل فئات **Classes** ، لكن وقبل حوالي ٤ سنوات من الآن ، تطور الوضع لدي كثيرا وأصبحت لا اكتب برامجي إلا بفئات - حتى لو كانت برامجي لأتسلى بها لوحدي ، تطور الوضع بعدها كثيرا فبدأت بكتابة مكتبات كاملة خاصة بي ، وكم كنت سعيدا عندما أصبحت أطبق مقولتك بأن اهتمامك سينصب في إنشاء فئات وتقسيمها وتحقيق مبدأ إعادة الاستخدام دون التفكير في الكود نفسه ، فأصبح البرنامج بالنسبة لي ينحصر في إنشاء فئات بالطريقة الصحيحة وليس في عمله لدى المستخدم!!

المشكلة هنا هي العادات السيئة للمبرمجين الغير متمرسين والتي يصرون عليها!

(((Vb4Arab))) (((Vb4Arab)))

AHMAD

أنا حاول أن افعل ما تحدثتم عنه وقد وضعت بعض الفئات التي استخدمها غالبًا ولكن الطريقة التي حتى الآن استخدمها بنسبة كبيرة عندما يتوجب على عمل وظيفة في برنامج قمت بعملها في برنامج آخر مسبقًا في أن أتذكر البرنامج وأذهب إلى السورس وانسخ الشفرات وأقوم بلصقها في المشروع الحالي مع بعض التعديل هذه الطريقة بالطبع متعبة أكثر من الطريقة الأولى ولكنها توفر على تحكم أقوى ،،وقد يكون السبب ضعف في قدرتي على بناء الفئات بشكل جيد

وبمناسبة هاجس عدم الرضى فقد حصل معي مرة أن قمت بإعادة تحزيم برنامجي سبع مرات دون أن يراه احد !!

(((Vb4Arab))) (((Vb4Arab)))
EGY Tiger

لا أشعر أنني سأضيف جديد بعد ما ذكر وللأسف فقد رأيت المقال متأخرا بحيث استطاع الأخوة الكرام تغطية كافة الجوانب التي يمكن التكلم عنها، ولكني (بناء على الطلب الصريح من الأستاذ تركي (أعظنا تجربتك) فسأدلي بدلوي في الموضوع.

المهم ، وبعيدا عن المقدمات فإنني لا أستخدم الفئات الجاهزة إلا في الأدوات التي أبرمجها بمعنى **Controls** ولكنني في ذات الوقت لا أكرر الكود الذي أكتبه إلا إذا شعرت أن تكرار الكتابة يمكن أن يؤدي لنتيجة أفضل، أو أنني غير واثق من الكود القديم،،،

ولكني أستخدم خاصية **Copy** و **Paste** ولي كلمة كثيرا كنت أقولها لعدد من أصدقائي وهي " لو أنك أعطيتني مشروع جديد فارغ وطلبت مني بناء مشروع جديد من الصفر فقد يستغرق ذلك وقتا طويلا جدا، أو قد لا أنجح على الإطلاق" ، وذلك أخواني لأنني شخص كسول جدا، ولا أحب أن أكتب الكود ذاته مرتين

وأذكر في بداية تعلمي للبرمجة كنت إذا عملت نافذتين تؤديان نفس الوظيفة، أو أداتين تؤديان نفس الوظيفة لوجدت الكود الذي كتبته في كل منهما مختلف عن الآخر، وأني أعتبر هذا عيبا خطيرا في لأنني أعتبر هذا حالة من غياب المعايير الموحدة لكتابة الكود.

المهم فإنني أستخدم خاصية النسخ واللصق للوظائف وليس للفئات وهذا هو حلي المتواضع للمشكلة

(((Vb4Arab))) (((Vb4Arab)))
usamasobh

أما عن تجربتي أن الشخصية فمثلا قال بعض الأخوة مشاريع صغيرة يستخدمون فيها القص واللصق أما أنا فلا أستخدم القص واللصق لأنني أحب الابتكار في الأكواد فلا مانع عندي أن أكتب الكود أكثر من مرة حيث أنني في كل مرة أكتب الكود أري أشياء لم أكن أراها من قبل مثلما قلت حضرتك أننا نحن المبرمجين لا نرضى بما قدمته أناملنا، وهذا ليس عيبا وإنما هو من وجهة نظري طريق للإبداع فكل مرة أكتسب فيها خبرة وأفهم أشياء لم أكن أفهمها كمثل الحال إذا كنت أستخدم فيجوال بيسك ٦ وتطورت لأستخدم الفيجوال بيسك دوت نت نفس الطريقة الكود أول مرة أكتبه ليس كأخر مرة وصلت إليه

تحياتي لك وأرجو أن تكون قد وضحت وجهة نظري مع أنني اتفق معك في جميع أرائك في هذا الموضوع ولكن ينقصنا أن نفتتح بالطريقة التي تقنع غرور المبرمج وتهديده لمصاعب البرمجة

(((Vb4Arab))) (((Vb4Arab)))
نور نبهان

السلام عليكم

أكثر ما اعتبره عادة غير محبذة بالنسبة لي

إنني عندما أنتهي من الكود وطبعاً يكون التصميم خرائط أقوم بعملية تنسيق الواجهات بترتيب العناصر إلى هنا الأمور تمام وهذا المطلوب البرنامج يعمل .

لكن عندما أريد تجميل الواجهة بالصور أو أدوات الزينة أغير رأبي ٤٠ مرة خلال يوم أو يومين وذلك لضعف الخيال التصميمي

(((Vb4Arab))) (((Vb4Arab)))

تستطيع أن تشاركنا في هذا الموضوع بالدخول على الرابط التالي : [اضغط هنا](#)

خاتمة :

الحمد لله رب العالمين

نرجو من الله عز وجل التوفيق و السداد وان نكون قد وفقنا في اغناء معلوماتكم البرمجية ولو بشيء يسير من فيض العلوم .

تستطيعون الاطلاع على جديد المقالات التقنية والبرمجية من خلال المنتدى ففي كل يوم ترى الجديد بسبب سير الركب في علم الحاسب و البرمجة .

إدارة منتدى فيجوال بيسك العرب

www.vb4arab.com

تقبلوا خالص تحياتي و لا تنسوا إخوانكم في سوريا من الدعاء

نور نبهان

www.nourafamia.com

nourafamia@hotmail.com

00963-966993175

أين الذين تهزهم أحزاني ؟

الحمد لله رب العالمين