

بسم الله الرحمن الرحيم

دورة البرمجة المتقدمة للمهندسين الكيميائيين  
**Advanced Programming for Chemical  
Engineers**

**APChE**

4



تحت رعاية موقع البترول العربي

[www.3rboil.com](http://www.3rboil.com)



إعداد/ مهندس أيمن رسلان

بكالوريوس هندسة تكرير البترول والبتروكيمياء – جامعة قناة السويس

"as\_raslan@yahoo.com"

0182050922

**تحذير:** هذه الدورة مجانية من موقع البترول العربي ولا يجوز استغلالها في الأعمال التجارية والربحية دون الرجوع إلى المؤلف وإلا يعد ذلك من أعمال النصب والاحتيال.

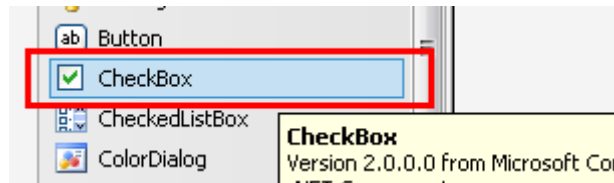
## المزيد من الأدوات..Tools

يحتوي الفيجزال بيزيك على العديد من الأدوات ... منها ما هو معروف للكثيرين ... منها ما هو مباشر الاستخدام ... منها ما يتم التعامل معه مباشرة ... ومنها ما يتم التعامل معه بطريقة خاصة. ولكل أداة خصائصها التي تميزها ... واستعمالاتها الخاصة.

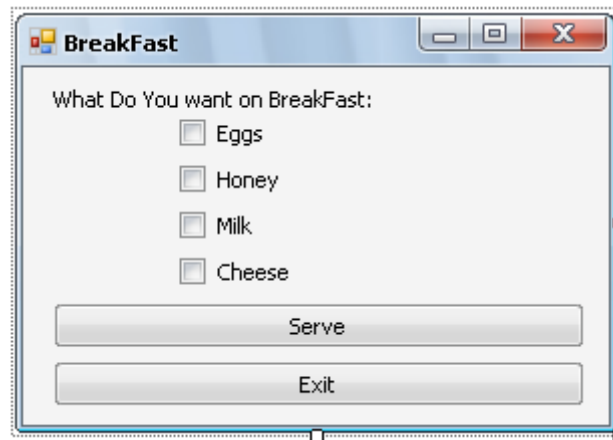
فمثلاً تعرفنا فيما سبق أن الـ Button يستخدم لتنفيذ وظيفة معينة عند الضغط عليه ... والـ TextBox يستخدم كأداة لعرض المدخلات والمخرجات Inputs and Outputs وعرفنا أن الفورم Form يمثل الواجهة أو النافذة التي نعرض فيها محتويات البرنامج من أدوات وكتابات وغيرها.

الآن سنتعرض لأدوات جديدة باختصار ونسرد الخصائص التي نتعامل معها برمجياً للاستفادة من هذه الأداة:

أولاً: الـ CheckBox : وهذه الأداة مشهورة ومعروفة لمستخدمي الانترنت وتستخدم لتحديد أكثر من خيارين في نفس الوقت والخاصية المهمة فيها هي Checked أقصد CheckBox.Checked والقيمة التي تحملها هي قيمة منطقية Boolean إما True وإما False ... هذه هي الأداة وإليك تطبيق عليها:



أنشئ مشروعاً جديداً وأعطه اسم Breakfast ثم قم بإدراج الأدوات التالية ... مع تغيير قيمة الخاصية Text للأدوات لتصبح بالشكل التالي:



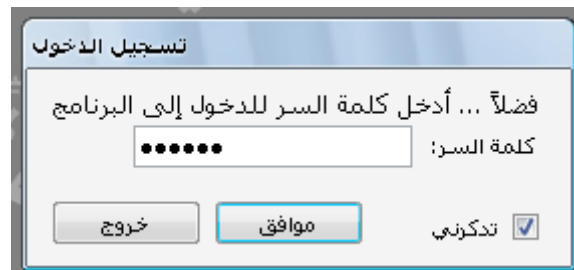
ثم قم ببرمجة الزر Serve بالكود التالي:

```

Start Page Form1.vb Form1.vb [Design]
Button1 Click
Public Class Form1
    Dim X As String
    Dim Z As String
    Dim Y As String
    Dim M As String
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        If CheckBox1.Checked = True Then
            X = CheckBox1.Text
        End If
        If CheckBox2.Checked = True Then
            Y = CheckBox2.Text
        End If
        If CheckBox3.Checked = True Then
            Z = CheckBox3.Text
        End If
        If CheckBox4.Checked = True Then
            M = CheckBox4.Text
        End If
        MsgBox("We Will get to you: " + X + " - " + Y + " - " + Z + " - " + M + " now")
    End Sub
End Class

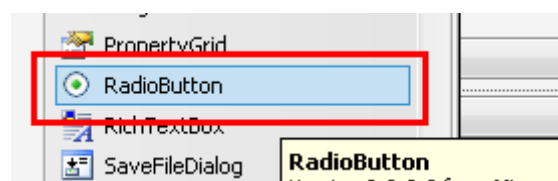
```

ولا تنس أننا قمنا بتعريف المتغيرات في الـ **Public Class** حتى تكون عالمية. الآن قم بتجربة البرنامج باختيار كل أو بعض من أنواع الطعمة التي يعرضها لك البرنامج ولتري النتيجة ... بالفعل البرنامج يعرض ما تم اختياره فقط. ومن أشهر استعمالان هذه الأداة في فورم تسجيل الدخول للبرنامج بحيث عند تحديد خيار تذكرني فإن البرنامج عندما يفتح يعرض كلمة السر مكتوبة أمام المستخدم ولا حاجة لكتابتها ... هكذا مثلاً:



والكثير من الاستخدامات والحيل للاستفادة من هذه الأداة ... استكشف الباقي بنفسك.

ثانياً: الـ **RadioButton**: وهي أداة مشابهة للـ **CheckBox** إلا إنها لا تسمح إلا بخيار واحد فقط من مجموعة خيارات عندما يقتضي الأمر أن نحدد للمستخدم خيار واحد فقط وأيضاً الخاصية البرمجية فيها هي **RadioButton.Checked** مثلها مثل الـ **CheckBox** .. هذه هي الأداة وإليك تطبيق عليها:



أنشئ مشروعاً جديداً وأعطه اسم **Fruit** واصنع الواجهة التالية:

ثم قم ببرمجة الزر Serve أيضاً كالتالي:

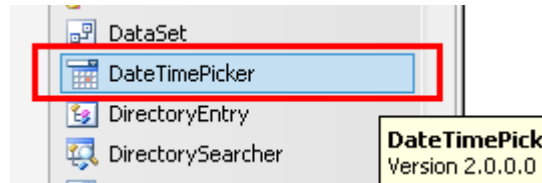
```

Start Page Form1.vb* Form1.vb [Design]*
Button1 Click
Public Class Form1
    Dim X As String
    Dim Z As String
    Dim Y As String
    Dim M As String
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        If RadioButton1.Checked = True Then
            X = RadioButton1.Text
        End If
        If RadioButton2.Checked = True Then
            Y = RadioButton2.Text
        End If
        If RadioButton3.Checked = True Then
            Z = RadioButton3.Text
        End If
        If RadioButton4.Checked = True Then
            M = RadioButton4.Text
        End If
        MsgBox("We Will get to you: " + X + Y + Z + M + " now")
        X = ""
        Y = ""
        Z = ""
        M = ""
    End Sub
End Class

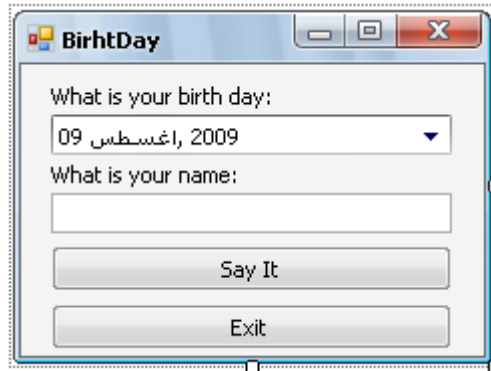
```

واشهر ما تستخدم فيه هذه الأداة كما قلنا هو تحديد اختيار واحد فقط من مجموعة خيارات ... وكذلك تغيير محتويات الواجهة للتعامل مع أدوات جديدة عند الضغط على الأداة فمثلاً في برنامج الـ Z-Factor ممكن أن نحدد خيارين أحدهما حساب الـ Z-Factor بالطريقة الأولى رياضياً .. فعندما يختار المستخدم هذا الخيار تظهر له الأدوات المطلوبة ... وإذا اختار الطريقة الثانية (Loops) فإنه الأدوات تختفي وتظهر مكانها أدوات جديدة طبقاً للاختيار الجديد ... وهكذا كل هذه أفكار برمجية والتطبيق الشامل سيأتي فيما بعد.

ثالثاً: أداة الـ DateTimePicker وكما هو واضح من اسم الأداة فإنها تستخدم ليلتقط بها المستخدم تاريخ معين بهدف تسجيل بيانات خاصة بهذا التاريخ مثلاً ... إليك الأداة والتطبيق:



أنشئ مشروعاً جديداً وأعطه اسم Birthday واصنع هذه الواجهة:



ثم قم ببرمجة الزر Say It هكذا:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal  
    MsgBox("Happy Birth Day To You " + TextBox1.Text)  
    MsgBox("Your Birht Day is: " + DateTimePicker1.Text)  
End Sub
```

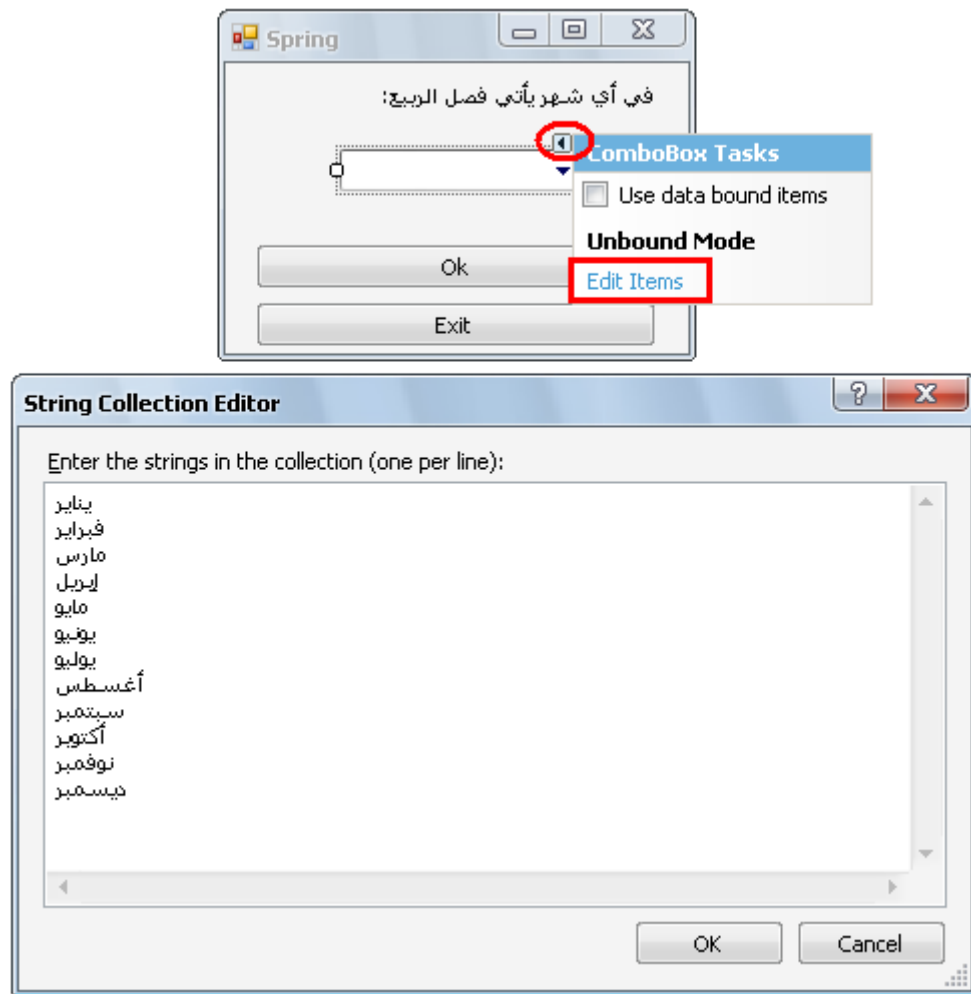
رابعاً: الـ ComboBox : وهو عبارة عن قائمة منسدلة يختار منها المستخدم خيار معين ..



أنشئ مشروعاً جديداً وأعطه اسم Spring أي الربيع واصنع الواجهة التالية:



ولا تنس تغيير قيمة الخاصية RightToLeft الخاصة بـ Label1 حتي يعرض بطريقة صحيحة ... الآن قم بتعبئة محتويات القائمة بالأشهر التالية ... كما في الشكل:

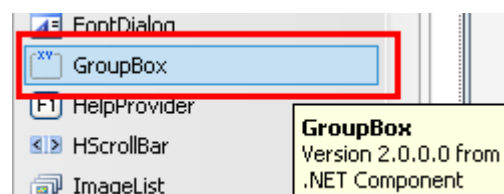


ثم قم ببرمجة الزر OK كالتالي:

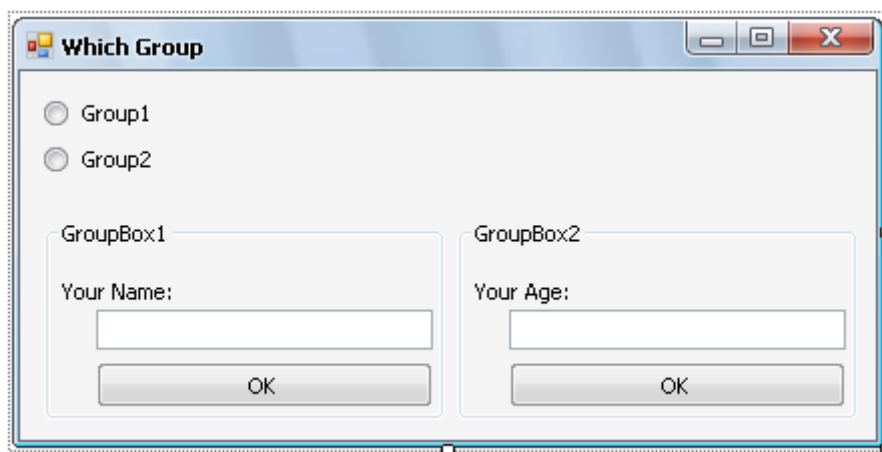
```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal
    If ComboBox1.Text = "مارس" Then
        MsgBox("إجابة صحيحة")
    Else
        MsgBox("غلط يا رايي اجل")
    End If
End Sub
```

ثم قم بتجربة البرنامج.

خامساً: الـ **GroupBox** : وتستخدم لتجميع مجموعة من الأدوات والتعامل معها ككتلة واحدة وهي أداة مساعدة ومفيدة جداً ...



أنشئ مشروعاً جديداً وأعطه اسم **Which Group** ثم اصنع الواجهة التالية:



ولا تنس أثناء سحب الـ Label والـ TextBox وإدراجه أن تجعله على الـ GroupBox وبذلك يصبح جزءاً منها حتى إذا قمت بتحريك الـ GroupBox وتغيير مكانها على الفورم فإنك ستجد أن المجموعة كلها تتحرك ككتلة واحدة ... المهم الآن نريد أن نضيف الكود الذي يزيد من قيمة البرنامج ... ولكن قبل ذلك قم بتغيير الخاصية Visible الخاصة بـ GroupBox2 إلى False ثم افعل التالي ... اضغط على الـ RadioButton الأول المسمى Group1 وعدل الخاصية Checked من قائمة الخصائص لتصبح True وبذلك عند فتح البرنامج يكون هذا الخيار هو المحدد ... بعدها اضغط على الـ RadioButton هذا المسمى Group1 وأضف فيه الكود التالي:

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As
    If RadioButton1.Checked = True Then
        GroupBox1.Visible = True
        GroupBox2.Visible = False
    Else
        GroupBox1.Visible = False
        GroupBox2.Visible = True
    End If
End Sub
```

ما الذي يعنيه هذا الكود ... بالفعل عندما تتغير حالة الـ RadioButton1 أي عندما يتم اختياره أو إلغاء اختياره باختيار الاختيار الآخر فإن البرنامج يختبر حالته فإذا كان هو الخيار المختار فإن GroupBox1 تختفي وتظهر GroupBox2 ... يمكنك فعل نفس الشيء مع الـ RadioButton الثاني المسمى Group2 هكذا ... إلا أنه لا حاجة لذلك فلا يوجد لدينا سوي خياران اثنان فقط.

## وهكذا يوجد الكثير من الأدوات

ولكل أداة استخداماتها الخاصة والأصل يرجع للمبرمج وذكائه في استغلال الأداة وبرمجتها بالطريقة الصحيحة التي تجعل منها فائدة عظيمة ... وإليك الآن بعض الإرشادات المهمة:

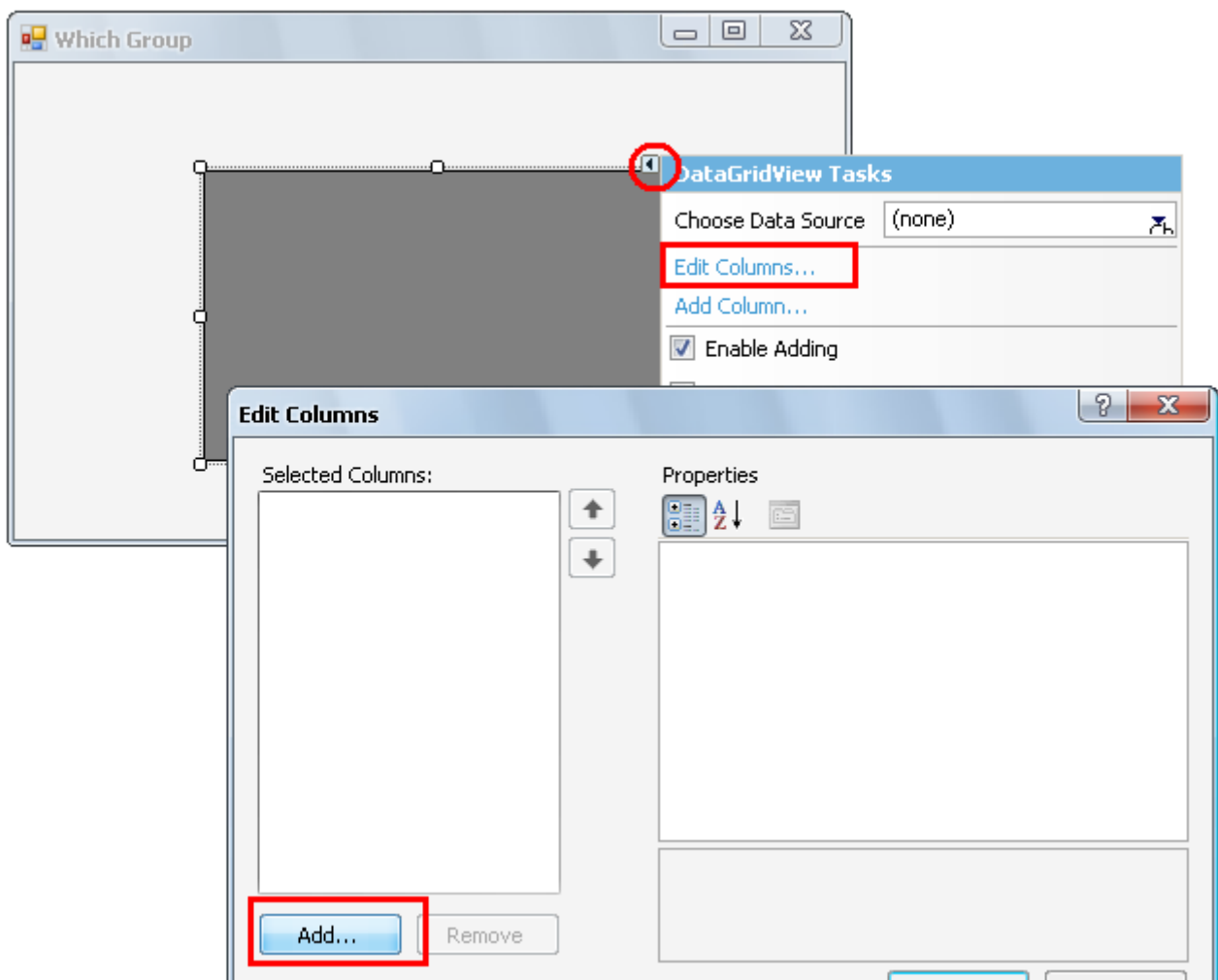
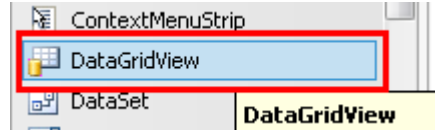
- بعض الأدوات يكون لها خيارات إضافية يمكنك التعامل معها مباشرة بالضغط على علامة الدبوس أو السهم الصغير الذي يظهر أعلى الأداة كما فعلنا مع RadioButton

- بعض الأدوات تظهر على الفورم وتظهر على الـ Tray بالأسفل مثلما كان الحال مع أداة الـ MenuStrip التي تعاملنا معها في الحلقات الماضية

- بعض الأدوات لا تظهر على الفورم وتظهر على الـ Tray مثل الـ Timer والـ ContextMenu ... غيرها ... وإليك مثال على كل مما سبق:

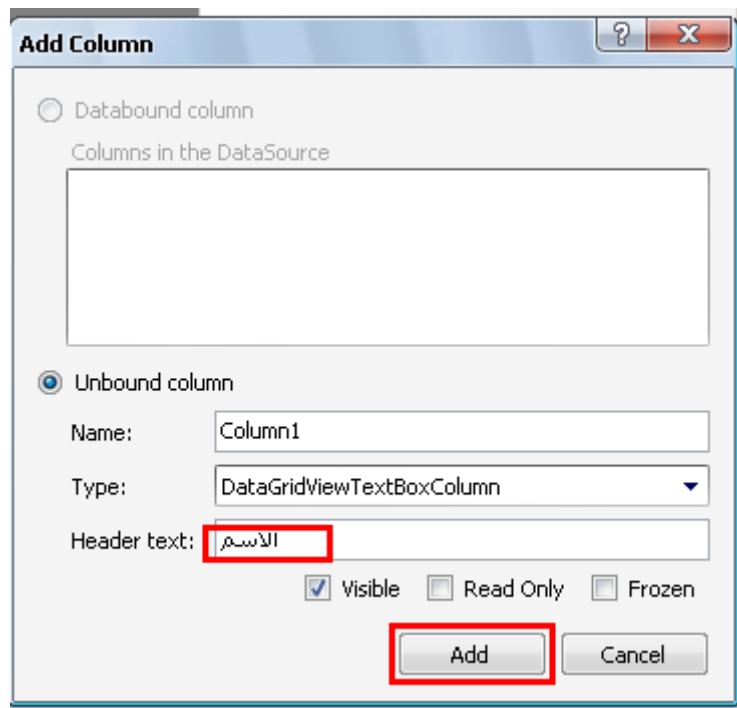
الخيارات الإضافية للأداة:

مثل أداة الـ DataGridView وهي تستخدم لعرض البيانات في جداول للمستخدم ... هكذا:

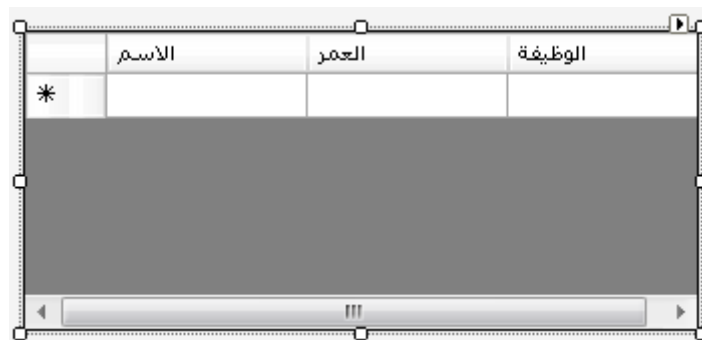


فهكذا نقوم بإضافة أسماء الأعمدة كالتالي مثلاً الاسم والعمر والوظيفة:





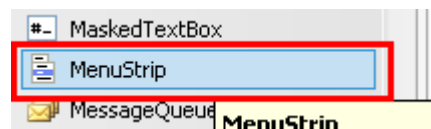
فيصبح شكل الـ DataGridView1 مكتملاً كالتالي:



سندرس تطبيق مهم على هذه الأداة في الحلقة القادمة.

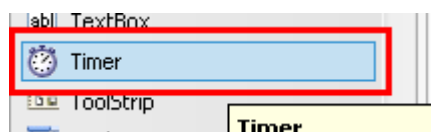
ثانياً: أدوات تظهر على الفورم وتظهر على الـ Tray :

مثل الـ MenuStrip ... وقد درسناها من قبل:

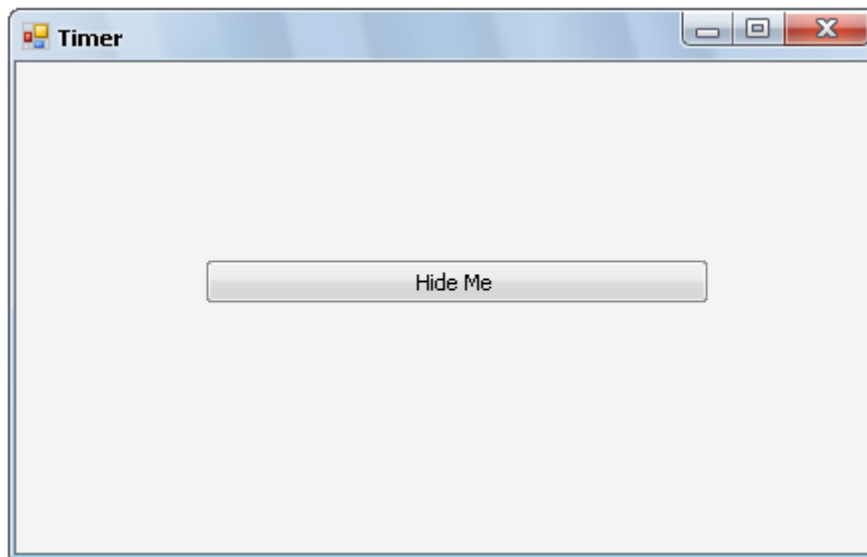


ثالثاً: أدوات تظهر على الـ Tray ولا تظهر على الفورم:

مثل الـ Timer وكما هو واضح من الاسم فهو مؤقت زمني يستخدم لتنفيذ مهمة معينة بعد زمن معين:



أنشئ مشروعاً جديداً وأعطه اسم Timer ... ثم أدرج الـ Timer على الفورم .. ستجد أنه لا يظهر على الفورم وإنما يظهر على الـ Tray بالأسفل ... دعنا الآن مع هذا المثال الرائع ... أضف زرّاً واحداً على الفورم وأعطه اسم Hide Me ...



ثم اذهب إلى الـ Timer على الـ Tray بالأسفل ... واضغط عليه لتفعيله والتعامل معه ... اذهب إلى قائمة الخصائص ... ثم قم بتعديل الخاصية Interval لتصبح قيمتها تساوي 200 ... فهذه هي المدة الزمنية بالملي ثانية التي سينفذ الـ Timer المهمة بعدها ... الآن عد إلى الـ Timer واضغط عليه مرتين وأضف بداخله الكود التالي:

```
Private Sub Timer1_Tick(ByVal sender As System.Object,  
    Me.Opacity = Me.Opacity - 0.1  
    If Me.Opacity = 0 Then  
        End  
    End If  
End Sub
```

ما الذي يعنيه هذا الكود ... إنه يعني أن الـ Timer ينفذ مهمة معينة كل 200 ميلي ثانية وهي أنه يقوم بإنقاص شفافية الفورم بمقدار 0.1 أو 10% إلى أن تصبح شفافية الفورم Opacity مساوية للصفر أي أنه يصبح مختفي تماماً فيغلق البرنامج تماماً .. الآن عد إلى الزر Hide Me واضغ عليه مرتين وأضف إليه الكود التالي لتفعيل الـ Timer للقيام بوظيفته (الخيار الافتراضي للـ Timer أنه يكون غير فعال ولا بد من تفعيله):

```
Private Sub Button1_Click(ByVal sender As  
    Timer1.Enabled = True  
End Sub
```

الآن قم بتجربة البرنامج واختبر وظيفة الزر والتايمر ... لا إنه ليس بسحر ... إنها البرمجة.

وهكذا هنالك المزيد والمزيد من الأدوات ... ولتتعرف أكثر ابحث عن كل أداة منفردة على الويب وتعرف كيف يمكن استخدامها وكيف يمكن برمجتها ... مثلاً اكتب في محرك البحث جوجل **ConTextMenu + Vb.net 2005** وستري الكثير من النتائج

واجب مطلوب منك

مطلوب منك البحث على الويب عن أداة الـ ContextMenu وأن تقوم بعمل أحد التطبيقات  
عليها ... ابحث وتعرف على ذلك بنفسك