

بسم الله الرحمن الرحيم

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

APChE

3



تحت رعاية موقع البترول العربي

www.3rboil.com



إعداد/ مهندس أيمن رسلان

بكالوريوس هندسة تكرير البترول والبتروكيمياء – جامعة قناة السويس

"as_raslan@yahoo.com"

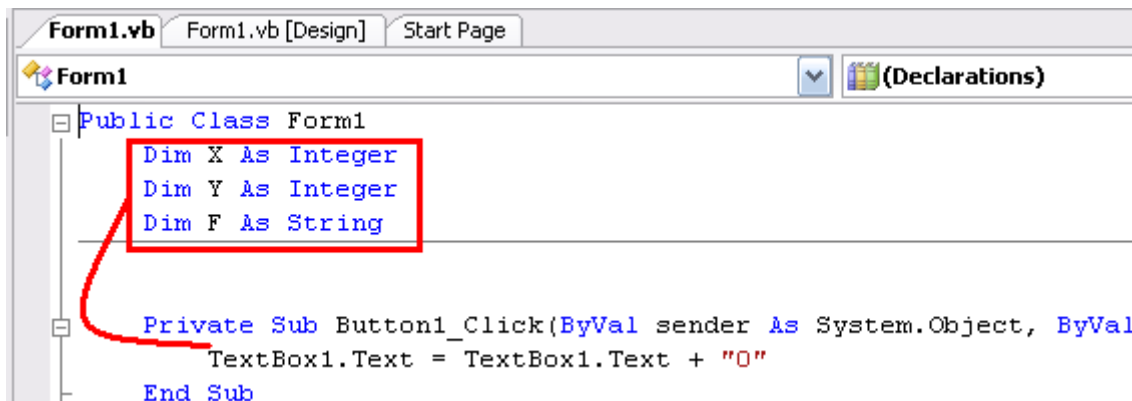
0182050922

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.

المتغيرات المحلية والعالمية ... Local and global variables

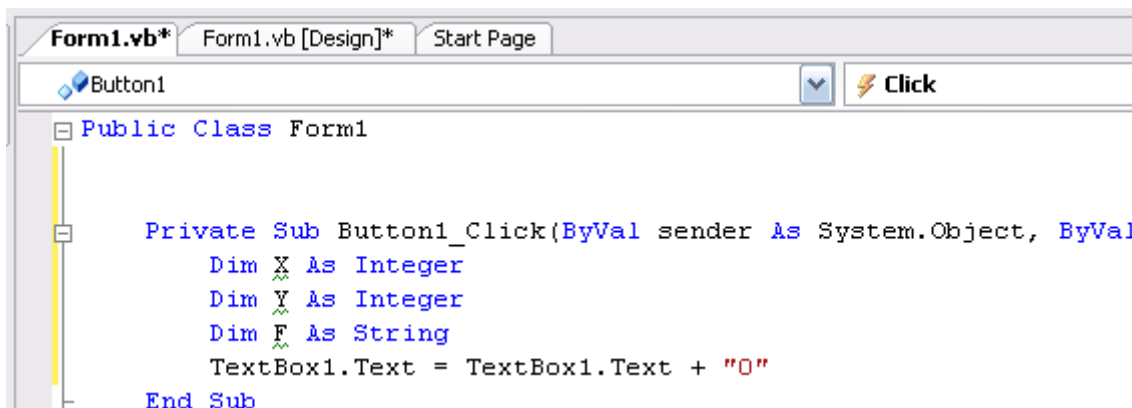
في الحلقة السابقة تعرضنا لما تعنيه كلمة متغير **variable** وطبقنا عليها بمثال بحيث كانت المتغيرات كما قلنا عالمية **global**. إذن فما الفرق بين المتغير المحلي **local** والعالمي **global**. المتغيرات المحلية هي التي تعرف فقط في محلها أو في منطقة معينة فقط من البرنامج أما المتغيرات العالمية هي التي تعرف على كامل البرنامج أي أنها تكون معروفة ويفهمها الفيجوال بيزيك في أي مكان تكتب فيه الكود.

وللتوضيح أكثر دعنا نجري القليل من التخريب على برنامج الـ **Genius Calculator** الذي أنشأناه في الحلقة السابقة ... افتح المشروع بالفيجوال بيزيك ... ثم اذهب منطقة الكود (يمكنك الوصول إلى منطقة كتابة الكود بالضغط كليك يمين في أي مكان في البرنامج ثم **View Code**) ثم قم بقص أسطر تعريف المتغيرات الثلاثة من منطقة الـ **Public Class** والصقها في مكان الـ **Button1** قبل السطر البرمجي الموجود فيه بالفعل ما الذي تلاحظ



```
Public Class Form1
    Dim X As Integer
    Dim Y As Integer
    Dim F As String

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal
        TextBox1.Text = TextBox1.Text + "0"
    End Sub
```



```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal
        Dim X As Integer
        Dim Y As Integer
        Dim F As String
        TextBox1.Text = TextBox1.Text + "0"
    End Sub
```

وإذا نزلت لأسفل ستلاحظ أخطاء كثيرة ظهرت بالبرنامج تحتها خطوط زرقاء، وإذا قمت بتشغيل البرنامج فإن البرنامج لن يعمل:

```

Private Sub Button12_Click(ByVal sender As System.Object, ByVal
X = TextBox1.Text
TextBox1.Text = ""
F = "+"
End Sub

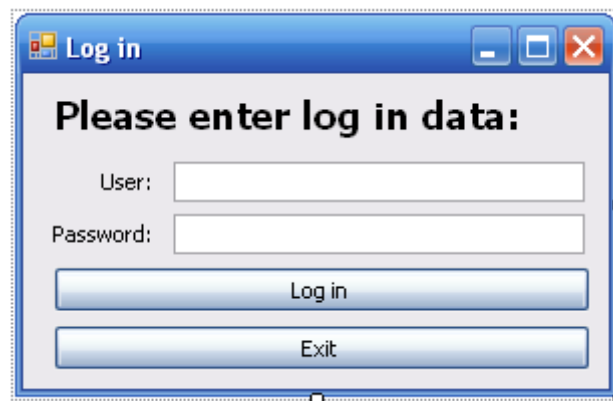
Private Sub Button13_Click(ByVal sender As System.Object, ByVal
X = TextBox1.Text
TextBox1.Text = ""
F = "-"
End Sub

```

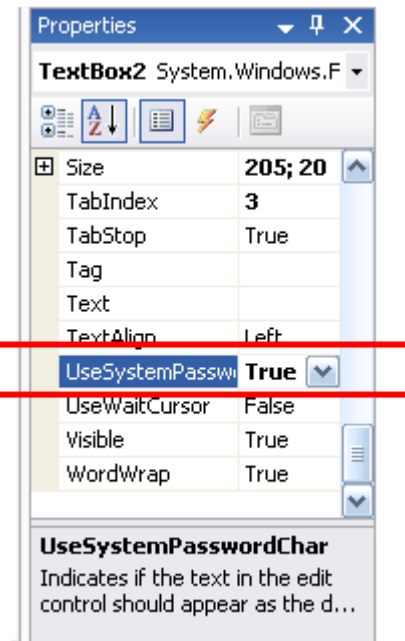
ما السبب في ذلك؟ ... السبب في ذلك أن تعريف المتغيرات أصبح في منطقة الـ **Button1** وهذه منطقة محلية (وهكذا بالنسبة لأي أداة أخرى غير **Button1**) أي أن البرنامج لا يعرف المتغيرات إلا في هذه المنطقة ... ولن يفهمها في منطقة الـ **Button2** وغيرها ... أما لتعريف المتغير بحيث يكون مفهوم ومستخدم في جميع مناطق وأجزاء البرنامج فلا بد من تعريف في منطقة الـ **Public Class** كما فعلنا في الحلقة السابقة.

استكمالاً لـ If Blocks

عرفنا أن جملة **if** الشرطية تستخدم لوضع قيود معينة على البرنامج بحيث لا يتم تنفيذ الحدث إلا إذا تم هذا الشرط ... وإن لم يتم الشرط فلن يحدث شيء. ويمكن توسيع دائرة الشرط بحيث أنه إذا تم الشرط يتم الحدث وإذا لم يتم الشرط يتم حدث آخر (لاحظ أنه سيتم حدث في الحالتين)، قم مثلاً بإنشاء برنامج جديد وأعطه اسم **Log in** واصنع الواجهة التالية:



فهذه واجهة مشهورة لتشفير البرامج وحمايتها باسم المستخدم وكلمة السر. بعد ذلك سنحتاج لأن نقوم بتعديل خصائص الـ **TextBox** الثاني بحيث أنه عندما يكتب المستخدم شيء فيه لا تظهر الكتابة وإنما تظهر نقاط تشفير كلمة السر المعروفة، قم بتعديل هذه الخاصية من قائمة الـ **Properties** كالتالي:



ثم اضغط على Button1 مرتين بالماوس للدخول إلى منطقة الكود ... واكتب الكود التالي:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal
    If TextBox1.Text = "ayman" And TextBox2.Text = "123" Then
        MsgBox("مرحباً يا أيمن")
    Else
        MsgBox("اسم مستخدم أو كلمة سر خاطئة")
    End If
End Sub
```

هذا الكود فيه الكثير لنشره ... ففيه الشكل المطول لجملته if ، أقصد أنها تكون على الشكل التالي:

If ... شرط آخر ... and ... الشرط

:

الحدث في حين ثبوت الشروط

:

Else

:

حدث آخر في حين انتفاء الشروط

:

End if

فيمكنك الجمع بين أكثر من شرط باستخدام كلمة And كما ترى ... بحيث أن الحدث لن يتم إلا بحدوث كلا الشرطين ... وهناك كلمات أخرى غير And أشهرها Or بمعنى أن الحدث لن يحدث إلا بحدوث أي من الشرطين ... إما هذا أو ذاك، إذاً افعل كذا ... وهكذا فإننا في مثالنا هذا

نطلب من البرنامج أن يختبر القيمتان اللتان ادخلهما المستخدم في TextBox1 و TextBox2 فإذا كانت قيمة "ayman" TextBox1.text وكذلك قيمة "123" TextBox2.text في نفس الوقت فإن البرنامج يظهر رسالة ترحيبية للمستخدم msgbox وإذا لم يصح أي من الشرطين Else فإن البرنامج سوف يظهر رسالة خطأ للمستخدم.

الآن قم بإضافة كود إغلاق البرنامج إلى الزر Exit كما تعرف ... ثم قم بتشغيل البرنامج لتختبر وظيفته.

وبهذه الطريقة يمكننا تشفير البرامج بكلمة سر واسم مستخدم بحيث لا يفتحها أي شخص ... وتوجد طرق وتقنيات أكثر تقدماً سنتعرف عليها فيما بعد.

Z-Factor

بالفعل حان الوقت لإنشاء أحد التطبيقات المتخصصة ... هل تعرف الـ Z-Factor إذا كنت لا تعرفه فأنت بالتأكيد لست من دارسي الهندسة الكيميائية وهذه الدورة لا تهدف إليك مباشرة ... المهم ابدأ مشروعاً جديداً وأعطه اسم الـ Z-Factor ودعنا الآن نفكر ما الذي سنصنعه.

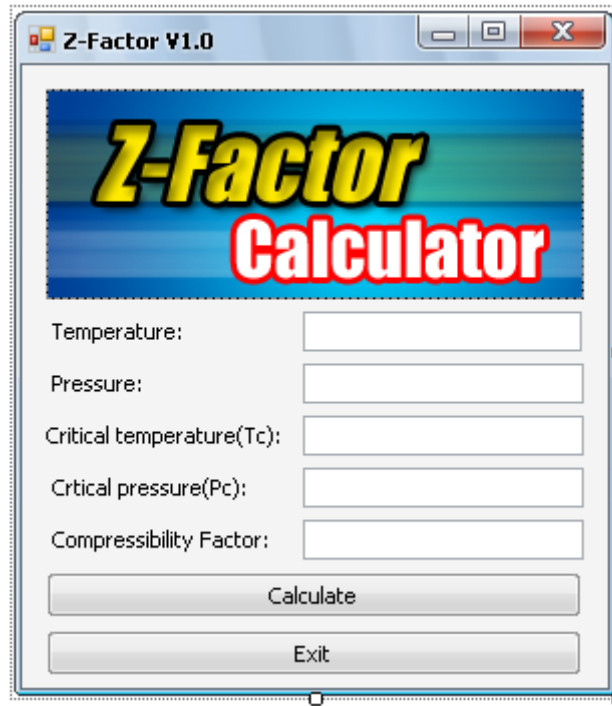
هناك طرق كثيرة لحساب الـ Z-Factor (أقصد طرق رياضية حسابية)، ودائماً في مثل حالتنا هذه نلجأ إلى المراجع والكتب لاختيار الطريقة المناسبة ... والكتب كثيرة ...

القاعدة الذهبية الأولى لإنشاء برامج الهندسة الكيميائية ... الجأ إلى المراجع والكتب لاختيار المعادلات الـ Empirical والصيغ الرياضية التي تحتاج إليها

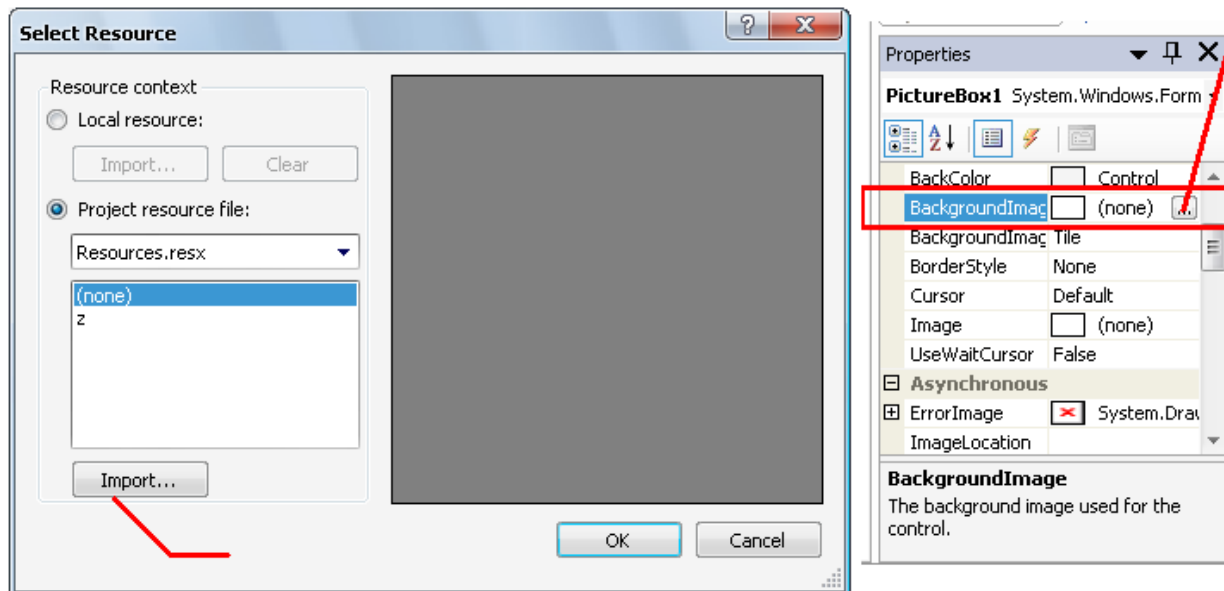
ومن الطرق المشهورة لحساب الـ Z-factor هي حسابه باستخدام الـ T_c و P_c بطريقة مباشرة Direct Method ومنها طريقة Papay Method ... هكذا:

$$Z = 1 - (P_{pr}/T_{pr}) [0.36748758 - 0.04188423 (P_{pr}/T_{pr})]$$

إذن الآن نحن نحتاج إلى مربع نص TextBox ليدخل فيه المستخدم قيمة الـ Temperature وآخر لإدخال الـ Pressure وآخر للـ T_c وآخر للـ P_c وخامس لتعرض فيه قيمة الـ Z-Factor بعد حسابها ... نحتاج كذلك إلى زر لتنفيذ الحسابات عند الضغط عليه ... وزر للإغلاق كالعادة ... وأضف إلى ذلك مربع صورة Picture Box لإضفاء شكل جمالي على البرنامج ... المهم اصنع الواجهة التالية:



الأدوات المستخدمة لا جديد فيها ... سوى مربع الصورة PictureBox ويمكنك إدراجه من صندوق الأدوات، ستجد أنه فارغ قم بعد ذلك بتعبئته بالصورة التي تراها (الصورة مرفقة مع الدرس) من خلال قائمة الخصائص Properties بتغيير خاصية Background Image كالتالي:



يتبقى لنا أنا نبرمج Button1 ليقوم بالحسابات المطلوبة – أقصد حساب قيمة الـ Z-Factor قم بكتابة الكود التالي:

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal
    Dim T As Double = TextBox1.Text
    Dim P As Double = TextBox2.Text
    Dim Tc As Double = TextBox3.Text
    Dim Pc As Double = TextBox4.Text
    Dim Tr As Double = T / Tc
    Dim Pr As Double = P / Pc
    Dim Z As Double
    Z = 1 - (Pr / Tr) * (0.36748758 - 0.04188423 * (Pr / Tr))
    TextBox5.Text = Z
    TextBox5.BackColor = Color.Yellow
End Sub

```

لقد قمنا بتعريف مجموعة من المتغيرات ثم استخدمنا هذه المتغيرات كما ترى في العملية الحسابية، وفي نهاية الكود بعد حساب الـ Z طلبنا من البرنامج أن يقوم بتغيير لون الـ TextBox5 إلى الأصفر لجذب الانتباه إليه. قم بتجربة البرنامج الآن.

لا تسخر مني

في برنامجنا الأخير هذا توجد الكثير من نقاط الضعف، فقد لا يقوم المستخدم بكتابة أي شيء في مربعات النصوص أصلاً ويضغط على زر الحساب Calculate وهذا نوع من السخرية والاستهزاء وعندها سيعطب البرنامج عن العمل ولا بد من قفله وإعادة تشغيله ... ولا بد من تحجيم مثل هذه الأمور ولو حتى على سبيل تفادي الأخطاء من البرنامج.

يتم ذلك عن طريق اختبار القيمة المدخلة في كل من مربعات النصوص المطلوبة ... فإذا كانت فارغة تظهر رسالة خطأ للمستخدم ولا يقوم البرنامج بالحساب ... إذاً يصبح الكود السابق كالتالي:

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1
    If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox4.Text = "" Then
        MsgBox("خطأ ... من فضلك أدخل جميع البيانات المطلوبة")
    Else
        Dim T As Double = TextBox1.Text
        Dim P As Double = TextBox2.Text
        Dim Tc As Double = TextBox3.Text
        Dim Pc As Double = TextBox4.Text
        Dim Tr As Double = T / Tc
        Dim Pr As Double = P / Pc
        Dim Z As Double
        Z = 1 - (Pr / Tr) * (0.36748758 - 0.04188423 * (Pr / Tr))
        TextBox5.Text = Z
        TextBox5.BackColor = Color.Yellow
    End If
End Sub

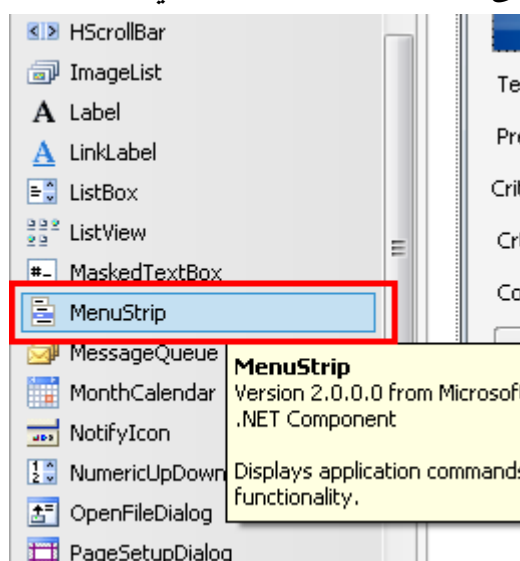
```

وهكذا تم التغلب على مثل هذه الأمور السخيفة. وهناك المزيد من السخافات ... فمن الممكن أن يقوم المستخدم بكتابة حروف في مربعات النصوص بدلاً من الأرقام ... هنا يجب أن نمنع

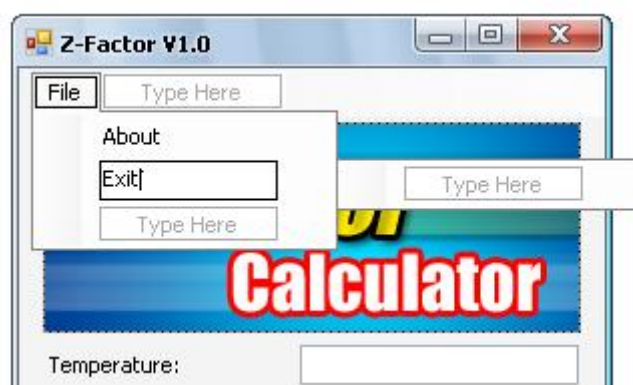
المستخدم أصلاً من كتابة الحروف ونسمح له بكتابة الأرقام فقط ... سندع هذا إلى مرحلة لاحقة.

اجعله يضيح بالحياة

من أسباب قبول البرنامج هو شكله ومرونته ... ولعلك لاحظت أنه عندما أضفنا مربع الصورة إلى البرنامج أصبح البرنامج أكثر احترافية (قم باستخدام برامج الديزايين الخاصة لصناعة صورك مثل الفوتوشوب ... وراعي دوماً استخدام الألوان التجارية كالأزرق والأحمر ... وابتعد عن الألوان المبهرجة) ... الآن سنضيف قائمة إلى البرنامج Menu مثلما ترى في البرامج الأخرى ... المهم أدرج الأداة من صندوق الأدوات ثم قم بإضافة قائمة File وأضف تحت قائمة File قائمة فرعية تحتوي على About و Exit كالتالي:



اسحب الأداة وأدرجها على الفورم (ربما يكون مربع الصورة قد غطى عليها لذا قم بتغيير مكان جميع محتويات الفورم للأسفل قليلاً) وستلاحظ أن الأداة قد نزلت تحت إلى ما يسمى بالـ Tray أو الصينية (بعض الأدوات لا يظهر مباشرة على الفورم ويتم التعامل معه من خلال الـ Tray) اضغط على الأداة في الـ Tray مرة واحدة لتحديد ما ستلاحظ ظهورها أعلى الفورم قم بتعديلها لتصبح كالتالي:



ثم ارجع إلى كلمة About واضغط عليها مرتين لنقوم ببرمجتها كالتالي:

```
Private Sub AboutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal  
    MsgBox("Z-Factor Calculator Version 1 Programmed By Ayman Raslan")  
End Sub
```

وبنفس الطريقة قم ببرمجة Exit كالعادة لإغلاق البرنامج. الآن أصبح البرنامج ملئ بالأدوات مقبول ومألوف من قبل المستخدم ويبدو أكثر احترافية ... قم بتجربة البرنامج الآن.

Z-Factor Version 2.0

ما رأيك الآن أن نبحث على طريقة أخرى لحساب ال-Z-Factor، لقد عثرت على طريقة في أحد الكتب المختصة بحسابات الهندسة الكيميائية تقوم بحساب ال-Z-Factor **لبخار الكحول الأيزوبروبيلي** عن طريق ال-Iteration حيث تكون خطوات الحساب كالتالي:

Step1: Assume Z and Calculate h where:

$$h = 0.0867 (P/P_c) / Z_{\text{assumed}} (T/T_c)$$

Step2: Recalculate Z and check for error where:

$$Z = 1 / (1 - h) - 4.9343 (T_c/T)^{1.5} (h / (1 + h))$$

Then: error = Z - Z_{assumed} < 0.01

إذاً فإن الحل يكون على خطوتين إلى أن يتحقق شرط الخطأ المقبول فإن لم يتحقق نعيد تكرار الحل مرة أخرى وهكذا ... إذاً نحتاج لمعرفة شئ جديد ... إنها الحلقات التكرارية

الحلقات التكرارية ... Loops

تستخدم الحلقات التكرارية بهدف عمل ال-Iteration وهي من أساسيات أي لغة برمجية ... ومنها أنواع كثيرة مثل حلقة For وحلقة Do و... وغيرها. سنكتفي بحلقة تكرارية واحدة وهي حلقة for والتي تأخذ الشكل التالي:

For i=1 to 10

:

الحدث

:

Next

فهذه الحلقة ستنكرر فيها تنفيذ الحدث عشر مرات، من المرة الأولى التي تساوي فيها i واحد إلى أن تصبح قيمة i تساوي عشرة ... وتكون الزيادة في i افتراضاً هي الوحدة ... ويمكن التحكم في مقدار الزيادة والنقصان في i بكتابة ذلك في الحلقة فيصبح السطر الأول كالتالي:

For i=1 to 10 Step 3

ففي هذه الحالة لن تتكرر الحلقة عشرة مرات ... وإنما أربعة مرات فقط ... في المرة الأولى حيث تكون $i=1$ وفي الثانية $i=4$ وفي الثالثة $i=7$ وفي الرابعة $i=10$.

إذاً كيف سننفذ هذه الحلقة على برنامجنا الجديد Z-Factor Version 2.0 .. قم بتصميم الواجهة التالية:



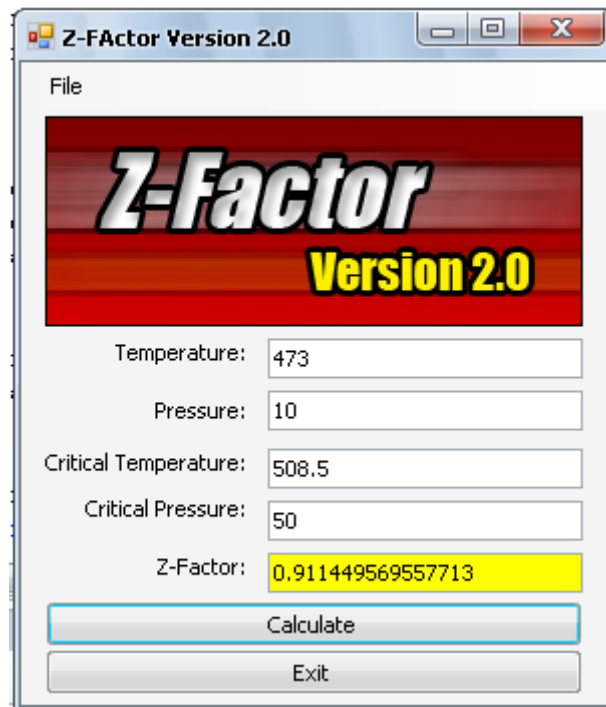
ثم قم ببرمجة زر Calculate كالتالي:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1
    If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox4.Text = "" Then
        MsgBox("خطأ ... من فضلك أدخل البيانات كاملة")
    Else
        Dim T As Double = Val(TextBox1.Text)
        Dim p As Double = Val(TextBox2.Text)
        Dim Tc As Double = Val(TextBox3.Text)
        Dim Pc As Double = Val(TextBox4.Text)
        Dim Zassumed As Double
        Dim h As Double
        Dim Z As Double
        Dim err As Double
        For Zassumed = 2 To 0 Step -0.01
            h = 0.0867 * (p / Pc) / (Zassumed * (T / Tc))
            Z = 1 / (1 - h) - 4.9343 * (Tc / T) ^ 1.5 * (h / (1 + h))
            err = ((Z - Zassumed) ^ 2) ^ 0.5
            If err <= 0.01 Then
                TextBox5.Text = Z
                TextBox5.BackColor = Color.Yellow
                Exit For
            End If
        Next
    End If
End Sub
```

طبعاً كما فعلنا في البرنامج السابق ... لقد قمنا بوضع شرط if إذا كان مربعات النصوص خالية لتفادي السخافات ... وبعدها قمنا بتعريف المتغيرات التي نحتاج إليها وأسندنا لكل منها قيمته ومن أين يأخذها ...

نأتي الآن إلى For loop فقد بدأناها من $Z_{assumed}=2$ وهي أقصى قيمة متوقعة لـ Z-Factor والقيمة النهائية هي الصفر ... أما الـ Step فقد جعلناها تتناقص بمقدار 0.01 بعد ذلك قمنا بوضع الصيغ الرياضية المطلوبة ... وعند حساب الـ err كتبنا الصيغة الرياضية التي تراها حتى تأتي بالقيمة الكتلقة للخطأ دون مراعاة الإشارة سواء أكانت موجبة أم سالبة ... وبعدها وضعنا if لاختبار قيمة الـ err فإذا كانت أقل من أو تساوي 0.01 فإننا نكتفي بذلك فنأمر البرنامج بأن يكتب قيمة الـ Z-Factor في مربع النص الخامس ويجعله لونه أصفر ... ثم يقوم بكسر الـ Loop عن طريق Exit For لمنع البرنامج من الاستمرار في تنفيذ الحلقة والخروج منها نهائياً.

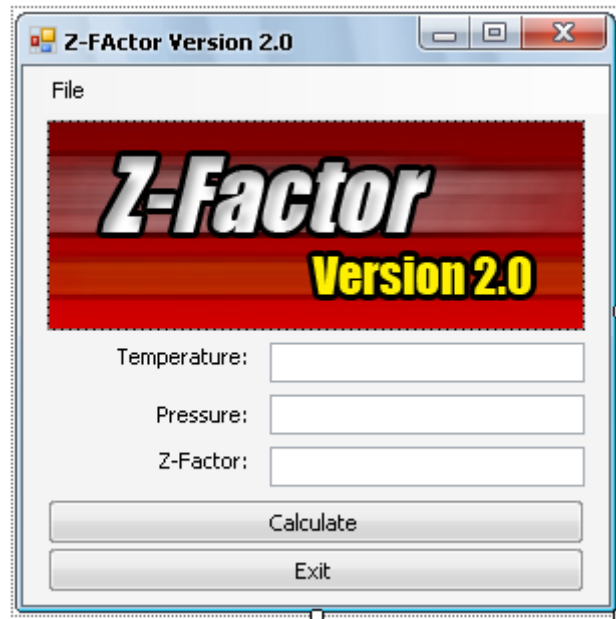
الآن دعنا نختبر عمل البرنامج ... في الكتاب الذي حصلت منه على هذه المعادلات يقدم الكاتب مثلاً بحيث قيمة $T=473\text{ k}$ وقيمة $P=10\text{ atm}$ وقيمة $T_c=508.5\text{ k}$ وقيمة $P_c=50\text{ atm}$ فتكون نتيجة الحسابات لديه أن $Z=0.91$... دعنا نرى القيمة التي يخرجها برنامجنا هذا:



بالطبع قيمة قريبة جداً مما حصل عليه ... بل وأدق منها ... ويزيد على ذلك أنه أصبح لديك برنامج جاهز لحساب الـ Z-Factor لبخار الكحول الأيزوبروبيلي عند أي ضغط وأي درجة حرارة ويمكنك الحصول منه على Data كثيرة جداً تستخدم في رسم المخططات Charts.

وكان من الممكن اختصار البرنامج أكثر من ذلك ... حيث إن قيمة T_c و P_c ثابتة ولا حاجة لكتابتها في كل مرة ... ومن الأفضل التعويض عنهما كثوابت في معادلات الحساب ... قم

بتعديل البرنامج ليصبح يحتوي على ثلاث مربعات نصوص فقط أحدهم للحرارة الآخر للضغط والثالث للـ Z-Factor ... قم بتعديل البرنامج الآن.



```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.  
    If TextBox1.Text = "" Or TextBox2.Text = "" Then  
        MsgBox("خطأ ... من فضلك أدخل البيانات كاملة")  
    Else  
        Dim T As Double = Val(TextBox1.Text)  
        Dim p As Double = Val(TextBox2.Text)  
        Dim Zassumed As Double  
        Dim h As Double  
        Dim Z As Double  
        Dim err As Double  
        For Zassumed = 2 To 0 Step -0.01  
            h = 0.0867 * (p / 50) / (Zassumed * (T / 508.5))  
            Z = 1 / (1 - h) - 4.9343 * (508.5 / T) ^ 1.5 * (h / (1 + h))  
            err = ((Z - Zassumed) ^ 2) ^ 0.5  
            If err <= 0.01 Then  
                TextBox5.Text = Z  
                TextBox5.BackColor = Color.Yellow  
                Exit For  
            End If  
        Next  
    End If  
End Sub
```

انتهت الحلقة

في هذه الحلقة تعرفنا على مبادئ جديدة ... ربما لم نكثّر من التطبيقات لكننا أكثرنا من المبادئ ... فقد عرفنا ما تعنيه المتغيرات المحلية والعالمية ... وكذلك توسعنا أكثر في جملة if وكذلك تعرفنا على أداة الـ PictureBox والـ MenuStrip وختمنّاها بالـ For Loop وأنشأنا تطبيق الـ Z-Factor ... في الحلقة القادمة ستعرف المزيد والمزيد