

مذكرة الوظائف الاشرافية



Visual Basic

المعام الدراسي 2026/2025م

Information Communication Technology

التوجيه الفني العام للحاسوب

جميع الحقوق محفوظة 2026/2025م



دولة الكويت
State of Kuwait



التوجيه الفني العام للحاسوب
General Supervisory of Computing



✉ ictgeneralsupervisor@moe.edu.kw

🌐 hasob.moe.edu.kw



— بعض دوال الفيجوال
البيسك المتقدمة

الدوال الرياضية Mathematics Functions .

الدالة Abs	الوظيفة	تعيد القيمة المطلقة للعدد
الدالة Sqrt	مثال	كتابة في تحميل النموذج <pre>Dim myny As single myny = Math.Abs(-45) Textbox1.Text = myny</pre> فإن نتيجة تنفيذ الدالة هي <pre>MyNumber=45</pre>
	الوظيفة	تحديد الجذر التربيعي لرقم معين
الدالة Log	مثال	<pre>Dim MyNumber As Integer MyNumber = Math.Sqrt(4) TextBox1.Text = MyNumber</pre> فإن نتيجة تنفيذ الدالة هي 2
	الوظيفة	تحديد قيمة اللوغاريتم العشري لرقم
الدالة Exp	مثال	<pre>Dim MyNumber As String MyNumber = Math.Log(20) TextBox1.Text = MyNumber</pre> فإن نتيجة تنفيذ الدالة هي <pre>MyNumber=2.9957327</pre>
	الوظيفة	تحديد القيمة (e) وهي قاعدة اللوغاريتم الطبيعي مرفوعة بقوة الرقم الذي تتضمنه حيث (e) تساوي تقريباً 2.7182818
الدالة Rnd	مثال	<pre>Label1.Text = Math.Exp(1)</pre>
	الوظيفة	توليد أرقام عشوائية تقع ما بين الصفر وواحد بحد أقصى 15 رقماً عشرياً
	مثال	<pre>Dim myny As Single</pre> كتابة في تحميل النموذج <pre>myny = Rnd(10) Textbox1.Text = myny</pre>
		الدالة Rnd(10) قد تعطي رقماً مثل 0.7055475 وعند تشغيل الدالة مرة أخرى ينتج رقماً آخر مثل 0.533424 وهكذا.

يعمل هذا التصريح مع الدالة Rnd ونستفيد منه لعدم التكرار والحصول على عشوائية أفضل وذلك لأنه يعتمد على ساعة النظام	الوظيفة	التصريح Randomize
Randomize MsgBox Int((10 * Rnd) + 1) توليد أرقام عشوائية من واحد إلى عشرة	مثال	
لحساب الجزء الصحيح فقط من رقم يشتمل على أرقام صحيحة وعشرية أو بعبارة أخرى لحذف الأرقام العشرية الموجودة بعد العلامة العشرية بدون تقريب	الوظيفة	الدالة Int
Dim myny As Single myny = Int(332.54) TextBox1.Text = myny فإن نتيجة الدالة هي 332	مثال	
دالة التقريب التي من خلالها يمكنك تحديد عدد الأرقام العشرية	الوظيفة	الدالة Round
Dim myny As Single myny = math.Round(4.52696,2) Dim myny As Single TextBox1.Text = myny حيث 4.52696 الرقم المراد تقريبه أما الرقم 2 فهو عدد الذي ستقرب بعد العلامة العشرية وسيكون الناتج 4.53	مثال	
تعمل نفس وظيفة الدالة Int مع الأرقام الموجبة و تختلف عنها في حساب الجزء الصحيح للأرقام السالبة	الوظيفة	الدالة Fix
Label1.Text = Fix(4.52696) فإن نتيجة تنفيذ الدالة هي 4	مثال	

دوال سلاسل البيانات String Functions .

الدالة Array	الوظيفة	تحويل عدة بيانات مدخلة كنصوص تفصلها فاصلة إلى مصفوفة يكون رقم أول عنصر فيها (Lower Bound) هو (0)
	مثال	<pre>Dim myArray() = {"first", "second", "third"} MsgBox(myArray(0)) MsgBox(myArray(1)) MsgBox(myArray(2)) MsgBox(myArray(3))</pre> النتيجة هي: First Second Third
الدالة UCase	الوظيفة	تحويل حالة النص المدخل من حالة الأحرف الصغيرة إلى حالة الأحرف الكبيرة
	مثال	<pre>Dim myny = "kuwait"</pre> كتابة في تحميل النموذج <pre>myny = UCase("kuwait") Textbox1.Text = myny</pre> النتيجة هي: KUWAIT
الدالة Format	الوظيفة	تنسيق رقم أو سلسلة حروف أو التاريخ/الوقت الموجود في التعبير تبعاً لتعليمات التنسيق الموجودة في الدالة نفسها .
	مثال	<pre>Dim mydate As String mydate = Format(Now(), "dd-mm-yyyy") MsgBox(mydate) MsgBox(Format(DateString, "short date"))</pre>
الدالة Trim	الوظيفة	حذف الفراغات الموجودة في الجهة اليسرى واليمنى من نص
	مثال	<pre>MyText = Trim(" Kuwait ")</pre>

<pre> T e x t1.t e x t = MyT e x t Dim myny = (" kuwait ") myny = Trim("kuwait ") Textbox1.Text = myny </pre> كتابة في تحميل النموذج		
حذف الفراغات الموجودة في الجهة اليمنى من نص	الوظيفة	الدالة Rtrim
<pre> MyT e x t = Rtrim("Kuwait ") </pre>	مثال	
حذف الفراغات الموجودة في الجهة اليسرى من نص	الوظيفة	الدالة Ltrim
<pre> MyT e x t = Ltrim(" Kuwait ") </pre>	مثال	
وتقوم هذه الدالة سلسلة فراغات محدده بالعدد الموجود بين الأقواس	الوظيفة	الدالة Space
<pre> Dim myny = ("kuwait ") myny = "lm" + Space(5) + ("kuwait ") Textbox1.Text = myny </pre> كتابة في تحميل النموذج	مثال	

دوال الوقت والتاريخ Date and Time

الدالة Now	الوظيفة	تعرض تاريخ اليوم والوقت الحالي باستخدام ساعة الحاسب
	مثال	Label1.Text= Now()
	مثال	لعرض الشهر الحالي Label1.Text= Now().Year
	الوظيفة	تعرض الوقت الحالي حسب ما هو مسجل بساعة الحاسب
الدالة Time	مثال	Label1.Text = TimeOfDay
الدالة Date	الوظيفة	تعرض التاريخ الحالي حسب ما هو مسجل بالحاسب
	مثال	Label1.Text = Date.today
الدالة Day والدالة Month	الوظيفة	تعرض الدالة Day() اليوم بدلالة رقمية يقع بين 1 ، 31 يمثل ترتيب اليوم المقابل لقيمة التاريخ، بينما تعرض الدالة Month() الشهر بدلالة رقمية يقع بين 1 ، 12 يمثل ترتيب الشهر المقابل لقيمة تاريخية
	مثال	Label1.Text = Now.Day Label1.Text = Now.Month Label1.Text = Now.DayOfWeek.ToString لعرض اسم اليوم
	الوظيفة	تعرض التاريخ المسلسل المقابل لليوم والشهر والسنة المدون بين الأقواس
الدالة DateSerial	مثال	Dim GetDate As Date GetDate = DateSerial(2003, 4, 11) MsgBox(GetDate)
	الوظيفة	تعرض الوقت المسلسل المقابل للساعة والدقيقة والثانية المدونة بين الأقواس
الدالة TimeSerial	مثال	Dim Get_time As Date Get_time = timeSerial(12,59,20) MsgBox(Get_time)



مهارات متقدمة

أنواع البيانات Data Types

توجد في فيجوال بيسك العديد من أنواع البيانات، فالأعداد لها أنواع تخزن بها قيم صحيحة وأنواع تخزن بها قيم كسرية، وكذلك توجد أنواع بيانات نصية ومنطقية وتاريخ ووقت وغير ذلك الكثير.

أنواع البيانات الرقمية الصحيحة في فيجوال بيسك:

النوع	الحجم بالبايت	القيمة الصغرى	القيمة العظمى
Byte	١	٠	٢٥٥
SByte	١	١٢٨-	١٢٧
Short	٢	٣٢٧٦٨-	٣٢٧٦٧
UShort	٢	٠	٦٥٥٣٥
Integer	٤	٢ ١٤٧ ٤٨٣٦ ٤٨-	٢ ١٤٧ ٤٨٣٦ ٤٧
UInteger	٤	٠	٤٢ ٩٤٩ ٦٧٢ ٩٥
Long	٨	٩ ٢٢٣٣ ٧٢ ٠ ٣٦٨ ٥٤٧ ٧٥٨ ٠ ٧-	٩ ٢٢٣٣ ٧٢ ٠ ٣٦٨ ٥٤٧ ٧٥٨ ٠ ٧
ULong	٨	٠	١٨ ٤٤٦ ٧٤٤ ٠ ٧٣٧ ٠ ٩ ٥٥١ ٦١٥

أنواع البيانات الرقمية العشرية:

النوع	الحجم بالبايت	القيمة الصغرى	القيمة العظمى
Single	٤	-3.4028235E38	3.4028235E38
Double	٨	1.79769313486231E308-	1.79769313486231E308
Decimal	١٦	يستخدم لتمثيل الأعداد الصحيحة والعشرية ويمكن تحديد الدقة العشرية من صفر إلى ٢٨ خانة عشرية	

أنواع البيانات النصية:

النوع	الحجم بالبايت	القيمة
Char	٢	يخزن به حرف واحد بترميز Unicode
String	غير محدود	يخزن به سلسلة حرفية من صفر إلى ٢ مليار حرف بترميز Unicode

كما يوجد أيضا أنواع أخرى من البيانات أهمها Boolean ويخزن به قيمة منطقية True/False ونوع Date ويخزن به التاريخ والوقت.

التفرع:

تقدم لنا لغة فيجوال بيسك مجموعة من التعليمات التي تقوم بتنفيذ تعليمة أو مجموعة من التعليمات بناء على قيمة منطقية لشرط معين، والقيمة المنطقية هي إما True أو False والشرط هو أي تعبير له قيمة منطقية فيمكن أن يكون الشرط ناتج مقارنة بين متغيرين باستخدام عوامل المقارنة مثل $A > b$ فهذا يسمى تعبير شرطي تكون نتيجته قيمة منطقية إما True إذا تحقق الشرط وإما False إذا لم يتحقق الشرط، والتعليمات اتخاذ القرار في فيجوال بيسك هي:

- If Then
- If Then Else
- Select Case
- Try Catch Finally

If Then

في حالة تنفيذ سطر واحد فقط من التعليمات البرمجية تكتب جملة If على الصورة التالية:

```
If [Conditional Expression] Then [Statement]
```

تكتب الجملة على سطر واحد بحيث تبدأ بالكلمة المحجوزة If ثم يأتي بعدها تعبير شرطي ثم يليه الكلمة المحجوزة Then ثم الجملة المطلوب تنفيذها.

Conditional Expression

هو تعبير شرطي يمكن أن يكون عملية مقارنة بين قيمتين أو متغيرين أو أكثر ويمكن أن يكون متغير من نوع Boolean، المهم أن تكون النتيجة في النهاية إما True أو False

Statement

هي تعليمة واحدة من التعليمات البرمجية أما إذا أردنا تنفيذ مجموعة من التعليمات تكتب الجملة على الصورة التالية:

```
If [Conditional Expression] Then [Statement 1] [Statement 2] [Statement 3]
End If
```

تكتب التعليمات على عدة أسطر بعد كلمة Then ثم تكتب الكلمة المحجوزة End If لإنهاء هيكل الشرط

مثال (١):

```
Dim X, Y As Short
X = 10
If X >= 10 Then Y = X * 2
```

التعبير الشرطي هو $(X \geq 10)$ وهو عملية مقارنة بين متغير وقيمة مجردة باستخدام عامل المقارنة $\geq$ ، نتيجة هذا التعبير ستكون True لأن قيمة X تساوي ١٠ إذن تحقق الشرط وسيتم تنفيذ الجملة $(Y = X * 2)$ المكتوبة بعد كلمة Then والتي تعني إسناد القيمة الناتجة عن حاصل ضرب المتغير X في العدد ٢ إلى المتغير Y لتصبح قيمة المتغير Y تساوي ٢٠

مثال (٢):

```
Dim X, Y, Z As Integer
Z = 1
If Z <> 9 Then
X += Z
Y *= X
End If
```

في هذا المثال سيتم تنفيذ تعليمتين عند تحقق الشرط ($z < 9$) وهما $x += z$ و $y *= x$ سيكون قيمة المتغير x تساوى 1 و قيمة المتغير y هي صفراً لأن المتغير الرقمي يأخذ القيمة الافتراضية صفر إذا لم تُسند له قيمة.

$y *= x$ يعادل $y = y * x$

تعليمية If Then Else

تُستخدم هذه الجملة إذا أردت تنفيذ تعليمات معينة في حالة عدم تحقق الشرط، وتكتب هذه التعليمات بعد الكلمة المحجوزة

Else

مثال على هذه الجملة: برنامج حساب النسبة المئوية لمجموع طالب
أنشئ مشروع جديد من Console Application ثم اكتب التعليمات البرمجية التالية:

```
Dim Student_Mark As Single
Student_Mark = Console.ReadLine
If Student_Mark >= 50 Then
    Console.WriteLine("Student Succeeded")
Else
    Console.WriteLine("Student failed") End If
```

تم الإعلان عن متغير لدرجة الطالب من نوع Single

السطر `Student_Mark = Console.ReadLine`

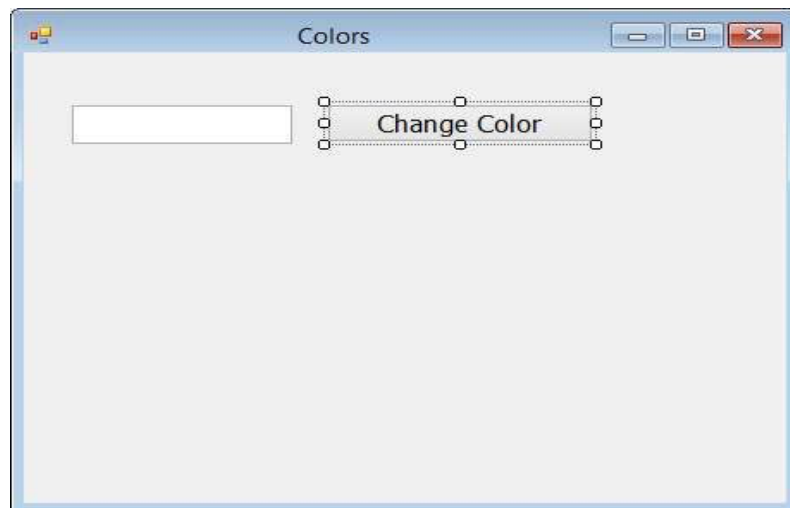
يعنى أن البرنامج سيقراً القيمة التي سيدخلها المستخدم في سطر الأوامر ويخزنها في المتغير Student_Mark بعد ذلك تأتى جملة الشرط التي تختبر درجة الطالب إذا كانت أكبر من أو تساوى ٥٠ يطبع البرنامج الجملة **Student Succeeded** وهي تعنى أن الطالب نجح، وإن لم يتحقق الشرط سيطبع البرنامج الجملة **Student failed** وهي تعنى أن الطالب فشل.

وفي بعض الحالات نحتاج إلى اختبار شروط متعددة بحيث أن البرنامج يختبر شرط معين فإذا تحقق هذا الشرط يقوم البرنامج بتنفيذ التعليمات المطلوبة ويخرج من الجملة الشرطية وإذا لم يتحقق الشرط يقوم باختبار شرط آخر وهكذا، نستخدم في هذه الحالة التعليمة التالية:

```
If [Conditional Expression] Then
    [Statements]
ElseIf [Conditional Expression] Then
    [Statements]
Else
    [Statements]
EndIf
```

يمكن توضيح هذه التعليمة في هذا المثال:

أنشئ مشروع جديد من نوع Windows Forms Application وضع أدوات TextBox و Button على Form كما في الشكل التالي



وغير خصائص الأدوات كما يلي:

القيمة	الخاصية	الأداة
btnColor	Name	Button
Change Color	Text	Button
txtColor	Name	TextBox
Colors	Text	Form

- اضغط على الزر btnColor وضغطة مزدوجة بالفأرة لفتح نافذة التعليمات البرمجية واكتب التعليمات البرمجية التالية في الحدث Click

```
Public Class Form1

    Private Sub btnColor_Click(sender As Object, e As EventArgs)
        If txtColor.Text = "Red" Then
            Me.BackColor = Color.Red
        ElseIf txtColor.Text = "Blue" Then
            Me.BackColor = Color.Blue
        ElseIf txtColor.Text = "Green" Then
            Me.BackColor = Color.Green
        End If
    End Sub
End Class
```

- بعد تشغيل البرنامج اكتب في مربع النص الكلمة Blue ثم اضغط على زر Change Color سيتم تغيير خلفية الـ Form لتصبح باللون الأزرق، وإذا كتبت كلمة Red ستتغير الخلفية للون الأحمر، وإذا كتبت كلمة Green ستتغير الخلفية للون الأخضر ما الذي يحدث عندما تضغط على زر Change Color :
- يبدأ البرنامج بتنفيذ جملة الشرط ويختبر الشرط الأول txtColor.Text = "Red" فإذا كانت الكلمة المكتوبة في مربع النص هي Red سيقف البرنامج التعليم التالية Me.BackColor = Color.Red والتي تقوم بتغيير خلفية النموذج إلى اللون الأحمر وينتقل البرنامج بعد ذلك إلى نهاية جملة الشرط End If ولا يختبر باقي الشروط ، اما إذا لم يتحقق الشرط ينتقل البرنامج لاختبار الشرط الثاني txtColor.Text = "Blue" فإذا كانت الكلمة المكتوبة في مربع النص هي Blue يتحقق الشرط وينفذ البرنامج تعليمة Me.BackColor = Color.Blue والتي تغير خلفية النموذج إلى اللون الأزرق ثم يخرج البرنامج من جملة الشرط وهكذا.

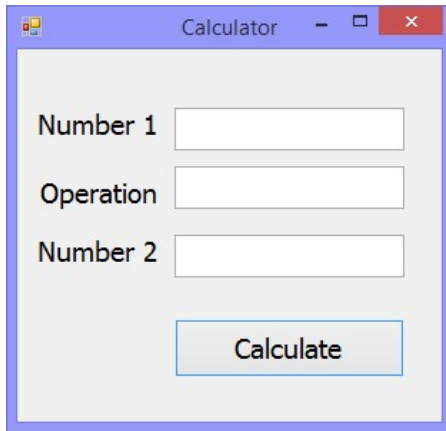
ويمكن كتابة هذا البرنامج بطريقة أخرى بهذا الشكل:

```
If txtColor.Text = "Red" Then Me.BackColor = Color.Red
If txtColor.Text = "Blue" Then Me.BackColor = Color.Blue
If txtColor.Text = "Green" Then Me.BackColor = Color.Green
```

فما الفرق بين الحالتين وأيها أفضل؟

تعليلة Select Case تستخدم عبارة Select Case لمقارنة تعبير أو متغير واحد بقيم متعددة وتختلف عنها عبارة Elself في أنها تقارن تعابير مختلفة في كل مرة.

ويمكن توضيح وظيفة العبارة Select Case من خلال هذا المثال:



◀ أنشئ مشروع جديد باسم SimpleCalculator واجعل واجهة البرنامج بهذا الشكل

◀ غير خاصية Name لأدوات الـ TextBox لتصبح على الترتيب

txtNum2 – txtOperation – txtNum1

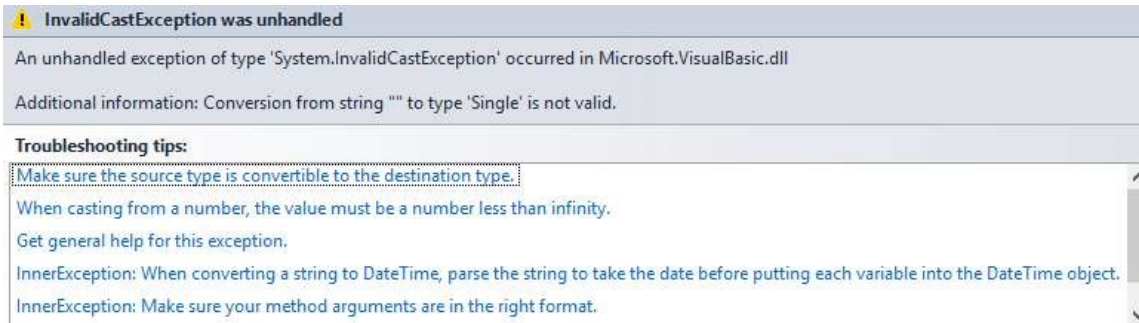
◀ غير خاصية Name للزر لتصبح btnCalc وغير خاصية Text لتصبح Calculate ثم افتح نافذة التعليمات

البرمجية واكتب التعليمات البرمجية التالية في حدث Click للزر btnCalc

```
Private Sub btnClac_Click(sender As Object, e As EventArgs)
    Dim calc_Operator As String = txtOperation.Text
    Dim Num1 As Single = txtNum1.Text
    Dim Num2 As Single = txtNum2.Text
    Dim Result As Single
    Select Case calc_Operator
        Case "+"
            Result = Num1 + Num2
        Case "-"
            Result = Num1 - Num2
        Case "*"
            Result = Num1 * Num2
        Case "/"
            Result = Num1 / Num2
    End Select
    MsgBox(Result)
End Sub
```

شرح التعليمات البرمجية:

- ◀ تم الإعلان عن متغير باسم Calc_Operator من نوع String يُخزن به المعامل الحسابي الذي سيتم إدخاله في مربع النص الثاني txtOperation
- ◀ تم الإعلان عن متغير باسم Num1 من نوع Single يُخزن بداخله العدد الذي سيتم إدخاله في مربع النص الأول txtNum1
- ◀ تم الإعلان عن متغير باسم Num2 من نوع Single يُخزن بداخله العدد الذي سيتم إدخاله في مربع النص الثالث txtNum2
- ◀ تم الإعلان عن متغير باسم Result يُخزن بداخله ناتج العملية الحسابية
- ◀ نبدأ بكتابة عبارة Select Case يليها المتغير calc_Operator الذي نريد اختبار قيمته في عدة حالات
- ◀ ثم نكتب بعد ذلك كل حالة بكلمة Case تليها القيمة المطلوب مقارنتها بقيمة المتغير calc_Operator ثم نكتب التعليمات المطلوب تنفيذها عندما تكون نتيجة المقارنة True أي عندما تتساوى القيمة التي تم إدخالها في مربع النص operationtext مع القيمة المكتوبة بعد كلمة Case
- ◀ وهكذا حتى تنتهي العبارة بكلمة End Select ثم نكتب كود إظهار رسالة تعرض قيمة المتغير Result
- ◀ قم بتشغيل البرنامج وادخل قيم عددية في مربعات النص الأول والثالث وأدخل المعامل الحسابي في مربع النص الثاني ثم اضغط على زر Calculate لإظهار النتيجة في رسالة ملاحظة:
- ◀ في حالة عدم إدخال المعامل الحسابي في مربع النص الثاني ستظهر الرسالة وبها القيمة صفر وذلك لأن عبارة Select Case لن ينفذ منها شيء لأنها لن تتساوى قيمة مربع المعامل الحسابي مع أي من حالات العبارة Select Case
- ◀ أما في حالة عدم إدخال قيم في أي من مربعات النص الخاصة بالأعداد ستظهر رسالة الخطأ التالية:



نص رسالة الخطأ

Conversion from string "" to type 'Single' is not valid

وهذا يعني أن المترجم غير قادر على تحويل القيمة الفارغة الموجودة في مربع النص الأول إلى النوع Single ليتم تخزينها في المتغير Num1

ولتتفادى هذا الخطأ يمكننا استخدام عبارة **Try Catch**

تستخدم العبارة Try Catch لمعالجة الاستثناءات ، بمعنى أنه من الممكن حدوث أخطاء استثنائية سواء كانت هذه الأخطاء متوقعة ومعلومة أو غير معلومة ، كما في المثال السابق إذا أدخل المستخدم القيم المطلوبة بشكل صحيح سيتم تنفيذ البرنامج بدون أخطاء وإذا حدث استثناء ولم يُدخل المستخدم إحدى القيم في هذه الحالة سيحدث خطأ ، فتستخدم العبارة Try Catch لاصطياد تلك الأخطاء ، وتُكتب الكلمة Try تليها التعليمات المطلوب تنفيذها ثم تُكتب كلمة Catch تليها

التعليمات المطلوب تنفيذها في حالة حدوث أخطاء ، ثم تليها كلمة End Try لإنهاء العبارة ويمكن إظهار رسالة للمستخدم بالخطأ وذلك بالإعلان عن متغير بعد الكلمة Catch بالشكل التالي

حيث ex هو اسم المتغير و Exception هو نوعه ex As Exception

ويمكن إظهار نص رسالة الخطأ للمستخدم من خلال الخاصية Message للمتغير ex فيصبح التعليمات البرمجية بالشكل التالي:

```
Private Sub btnClac_Click(sender As Object, e As EventArgs)
    Try
        Dim calc_Operator As String = txtOperation.Text
        Dim Num1 As Single = txtNum1.Text
        Dim Num2 As Single = txtNum2.Text
        Dim Result As Single
        Select Case calc_Operator
            Case "+"
                Result = Num1 + Num2
            Case "-"
                Result = Num1 - Num2
            Case "*"
                Result = Num1 * Num2
            Case "/"
                Result = Num1 / Num2
        End Select
        MsgBox(Result)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
```


التكرار

تُستخدم حلقات التكرار لتنفيذ مجموعة من التعليمات لعدد معين من المرات بناءً على شرط محدد، وتقدم لغة فيجوال بيسك 2008 العديد من حلقات التكرار.

Do ... Loop

تُستخدم حلقة Do ... Loop لتكرار تنفيذ مجموعة من التعليمات لعدد غير محدد من المرات بناءً على القيمة المنطقية للشرط المستخدم، ويستمر تنفيذ الحلقة التكرارية طالما تحقق الشرط أي عندما تكون قيمته True، والشرط كما ذكرنا سابقاً هو أي تعبير تكون له قيمة منطقية. الصورة العامة للتكرار:

Do... Loop

Do While Condition code block loop

مثال:

طباعة الأعداد من ١ إلى ١٠ على الشاشة

افتح مشروع جديد من نوع Console Application وقم بتغيير اسمه إلى 10Numbers1To ، ثم اكتب التعليمات البرمجية التالية:

```
Sub Main()
    Dim N As Byte = 1
    Do While N <= 10
        Console.WriteLine(N)
        N += 1
    Loop
    Console.ReadLine()
End Sub
```

شرح التعليمات البرمجية:

في البداية تم الإعلان عن متغير من نوع Byte ليكون عداد للحلقة وتم إعطائه القيمة الابتدائية ١

```
Dim N As Byte = 1
```

بدأت الحلقة بالكلمات المحجوزة Do تليها كلمة While يليها الشرط المطلوب تكرار تنفيذ الحلقة إذا كانت نتيجته True، والشرط هنا عبارة عن عملية مقارنة متغير بقيمة محددة باستخدام معاملي المقارنة <=، ومعنى هذا أن الحلقة لن تبدأ في التنفيذ إذا كانت نتيجة هذا الشرط False نكتب الجملة التالية لطباعة قيمة المتغير N على الشاشة السوداء

Console.WriteLine(N)

والتي ستطبع القيمة الابتدائية للمتغير N وهي ١

ثم نقوم بزيادة قيمة المتغير N بمقدار ١ عن طريق كتابة التعبير التالي

N +=

لاحظ أننا استخدمنا معاملي التعيين الحسابي += بدلا من كتابته على هذه الصورة N=N+1 الآن أصبحت قيمة المتغير N تساوي ٢

ثم نكتب الكلمة المحجوزة Loop والتي تعيد التنفيذ لبداية الحلقة فيتم اختبار الشرط مرة أخرى وبناء على القيمة الناتجة من الشرط يتم التنفيذ أو يتم الخروج من الحلقة إذا كانت القيمة الناتجة هي False أما السطر التالي

Console.ReadLine()

فالفائدة منه هو إبقاء الشاشة السوداء مفتوحة لحين الضغط على مفتاح Enter بعد تشغيل البرنامج ستجد أنه طبع الأعداد من ١ إلى ١٠ على الشاشة

استخدام Until بدلاً من While في حلقة Do.. Loop

عند استخدام كلمة While مع حلقة Do.. Loop يتم اختبار الشرط قبل الدخول إلى الحلقة بينما يمكن اختبار الشرط بعد تنفيذ الحلقة لمرة واحدة على الأقل باستخدام الكلمة المحجوزة Until وتكتب الحلقة على هذه الصورة:

Do

Code Block

Loop Until Condition

نبدأ بكتابة كلمة Do ثم التعليمات المطلوب تكرار تنفيذها، ثم نقوم بتكرار التنفيذ باستخدام كلمة Loop بناء على قيمة التعبير الشرطي المكتوب بعد كلمة Until، الاختلاف الثاني بين الحلقتين هنا حيث يتم تكرار تنفيذ الحلقة إذا كانت القيمة الناتجة عن الشرط هي False ويتم إنهاء الحلقة إذا كانت قيمة الشرط True


```
Sub Main()
    Dim N As Byte = 1
    Do
        Console.WriteLine(N)
        N += 1
    Loop Until N > 10
    Console.ReadLine()
End Sub
```

في هذه الحلقة سيتم تنفيذ التعليمات مرة واحدة على الأقل قبل اختبار قيمة الشرط تغير التعبير الشرطي فبدلاً من $N \geq 10$ عند استخدام الكلمة While في بداية الحلقة، أصبح $N < 10$ عند استخدام الكلمة Until في نهاية الحلقة

ملاحظات هامة:

- يمكن استخدام كلمة While في نهاية الحلقة لاختبار الشرط بعد تنفيذ التعليمات مرة واحدة على الأقل، ويمكن أيضاً استخدام الكلمة Until في بداية الحلقة لاختبار الشرط قبل الدخول في الحلقة، جرب كتابة الحلقة باستخدام كلمة While في نهاية الحلقة ومرة أخرى باستخدام كلمة Until في بداية الحلقة وحاول أن تستنتج الفرق بينهما.
- يمكنك استخدام عبار Exit Do للخروج من الحلقة في أي وقت إذا تحقق شرط معين.
- يمكن أن تستمر الحلقة التكرارية إلى ما لانهاية إذا لم يتحقق الشرط المطلوب لإنهاء الحلقة، ففي المثال السابق مثلاً إذا لم تقم بزيادة قيمة المتغير N ستدخل الحلقة في عملية تكرارية لا نهائية وستطبع الرقم 1 في كل مرة.

For ... Next

تُستخدم الحلقة التكرارية For ... Next لتكرار تنفيذ مجموعة من التعليمات لعدد محدد من المرات معلوم مسبقاً على عكس حلقة Do والتي لا يُعرف فيها عدد مرات التكرار مسبقاً، وفي حالة معرفة عدد مرات التكرار المطلوبة من المفضل استخدام حلقة For Next بدلاً من حلقات Do والصورة العامة لحلقة For.. Next بهذا الشكل:

```
For Counter=Start To End [Step]
    Code Block
Next Counter
```

تستخدم حلقة For متغير عددي كعداد للحلقة Counter تحدد له قيمه ابتدائية Start وقيمة نهائية End، ثم تحدد قيمة الخطوة أي مقدار الزيادة Step وإذا لم يتم تحديدها تكون قيمة الزيادة الافتراضية هي ١، ثم يتم تكرار الحلقة بكلمة Next يليها اسم متغير العداد

Counter

مثال:

طباعة الأعداد الفردية من ١ إلى ١٠٠

أنشئ مشروع جديد من نوع Console Application وقم بتغيير اسمه إلى 100PrintFrom1To ثم اكتب التعليمات البرمجية التالية:

```
Sub Main()
    Dim N As Byte
    For N = 1 To 100 Step 2
        Console.WriteLine(N)
    Next N
    Console.ReadLine()
End Sub
```

بدانا بالإعلان عن متغير عداد الحلقة وسميناه N من نوع Byte

ملاحظة: يمكن تسميه متغير العداد بأي اسم مناسب، أما اختيار النوع Byte فهو الأنسب في هذه الحالة حيث أننا لن نحتاج إلا إلى تخزين القيم العددية من ١ إلى ١٠٠ في هذا المتغير فليس هناك حاجة لاستخدام متغير أكبر في المدى مثل Integer أو Short

بدأت الحلقة بالكلمة المحجوزة For ثم تحديد القيمة الابتدائية للمتغير ب ١ والقيمة النهائية ب ١٠٠، ثم تم تحديد قيمة الخطوة ب ٢ وذلك حتى يتم طباعة الأعداد الفردية فقط، ثم نكتب جملة طباعة قيمة المتغير على الشاشة ثم كلمة Next والتي تقوم بزيادة قيمة متغير العداد بمقدار القيمة المحددة في Step ثم تعيد تنفيذ الحلقة حتى تصل قيمة العداد إلى قيمة أكبر من القيمة النهائية وهي ١٠٠

ويمكن أيضاً الخروج من الحلقة في أي وقت باستخدام الكلمة Exit For

مثال على حلقات التكرار:

برنامج لطباعة حاصل ضرب عددين حسب الأعداد التي يتم إدخالها باستخدام حلقات التكرار

أنشئ مشروع جديد من نوع Windows Forms Application وقم بتصميم الواجهة بهذا الشكل:



- ◀ قم بتغيير خاصية Text للفورم إلى New Calculator
- ◀ أضف أداة ListBox وزر Button وقم بتغيير خاصية Text له إلى Start وغير خاصية Name إلى btnStart ثم قم بفتح نافذة التعليمات البرمجية في حدث Click للزر btnStart واكتب التعليمات البرمجية التالي:

```
Private Sub btnStart_Click(sender As Object, e As EventArgs) Handles
    Dim Num1, Num2, Result As Single
    Try
        Do
            Num1 = InputBox("Input First Number", "Input Number")
            Num2 = InputBox("Input Second Number", "Input Number")
            ListBox1.Items.Add(Num1 & "*" & Num2 & "=" & Num1 * Num2)
        Loop Until Num1 = Nothing Or Num2 = Nothing
    Catch ex As Exception
        MsgBox("Program Cancelled")
    End Try
End Sub
```

شرح التعليمات البرمجية:

Dim Num1, Num2, Result As Single

- الإعلان عن ثلاثة متغيرات من نوع Single لأننا نريد إيجاد حاصل ضرب أي عدد سواء كان صحيحاً أو به جزء عشري لذا كان الاختيار الأنسب للنوع Single واستخدمنا عبارة Try لتفادي الخطأ وستظهر الفائدة من استخدامها عند تنفيذ البرنامج
- استخدمنا حلقة Do وكلمة Until في نهاية الحلقة وذلك لكي يتم تنفيذ الحلقة مرة واحدة على الأقل حتى يمكننا اختبار الشرط الذي يفترض أن صندوق الإدخال InputBox قد ظهر بالفعل وتم الضغط على زر Cancel لإرجاع قيمة "لا شيء"

Num1 = InputBox("Input First Number", "Input Number")

Nothing

- نقوم بتعيين قيمة للمتغير Num1 عن طريق استدعاء صندوق الإدخال InputBox لإدخال قيمة عددية للرقم الأول
- نستدع صندوق الإدخال مرة أخرى لتعيين قيمة للعدد الثاني وتخزينها في المتغير Num2

```
Num2 = InputBox("Input Second Number", "Input Number")
```

- في السطر التالي نقوم بطباعة عملية الضرب في أداة ListBox، واستخدمنا العامل التسلسلي & لضم مجموعة من القيم مع بعضها البعض لتظهر كتلة واحدة

```
ListBox1.Items.Add(Num1 & "*" & Num2 & "=" & Num1 * Num2)
```

- ثم تأتي مرحلة اختبار الشرط باستخدام كلمة Until

Loop Until Num1 = Nothing Or Num2 = Nothing

والذي يختبر القيمة التي سيتم إرجاعها من صندوق الإدخال في حالة الرقم الأول أو الرقم الثاني إن كانت Nothing تنتهي الحلقة ، ولاحظ أننا استخدمنا في هذا الشرط العامل المنطقي Or والذي يقارن بين تعبيرين منطقيين ويرجع قيمة منطقية وكما ذكرنا سابقاً ففي جميع الحالات سيُرجع قيمة True إلا في حالة واحدة فقط وهي أن تكون قيمة التعبيرين هي False ، والمطلوب في هذا الشرط للخروج من الحلقة هو أن تكون النتيجة True أي عندما تكون قيمة أي من التعبيرين True أو كليهما ومعنى أن تكون قيمة التعبير True هي أن يضغط المستخدم على زر Cancel في صندوق الإدخال فيُرجع القيمة Nothing وبالتالي تصبح نتيجة التعبير True وفي حالة الضغط على زر Cancel لإنهاء الحلقة يحدث الخطأ التالي

```
Conversion from string "" to type 'Single' is not valid.
```

ويحدث هذا الخطأ لأن المتغير المطلوب إسناد القيمة له من نوع Single والقيمة المطلوب إسنادها إليه من صندوق الإدخال هي من نوع String ففي حالة إن كانت هذه القيمة Nothing يحدث هذا الخطأ فتتلقاه Catch وتظهر رسالة بإنهاء البرنامج وتتوقف عملية الضرب

ويمكن تفادي هذا الخطأ بطرق أخرى منها على سبيل المثال تحويل القيمة الموجودة في صندوق الإدخال إلى قيمة رقمية قبل إسنادها إلى المتغير باستخدام الدالة Val كما يلي

```
Num2 = Val(InputBox("Input Second Number" "Input Number"))
```

في هذه الحالة لن يحدث خطأ عندما نضغط على زر Cancel في صندوق الإدخال.



أوراق عمل إثرائية

العام الدراسي : 2022-2023

ورقة عمل إثرائية (1) الموضوع: توليد الأرقام العشوائية

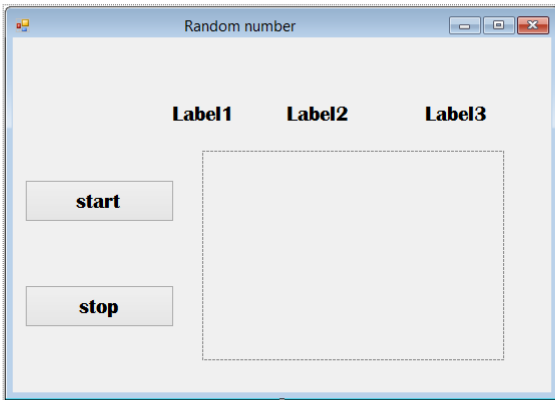


الفكرة الرئيسية:

عند الضغط على زر Start تظهر ارقام عشوائية من صفر إلى تسعة في كل من Label 1 ، Label2 ، و Label3 لحين الضغط على زر Stop، وعند ظهور رقم 7 في أي منهم تظهر الصورة في Picture Box

الخطوة الأولى:

انشاء نموذج باسم Random number



الخطوة الثانية:

إضافة الكائنات

- 1- Button: عدد (2) ، واضبط خاصية text للزر الأول Start، والزر الثاني Stop.
- 2- Label: عدد (3)
- 3- Picture Box : عدد (1) ، مع مراعاة:
 - تغيير الخاصية Visible الى False لعدم ظهور الصورة في بداية التشغيل.
 - تعديل حجم الصورة من SizeMode لتصبح Zoom.

الخطوة الثالثة:

كتابة التعليمات البرمجية

أولاً: النقر المزدوج على زر Start وكتابة التعليمات البرمجية التالية:

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

    PictureBox1.Visible = False
    Label1.Text = CStr(Int(Rnd() * 10))
    Label2.Text = CStr(Int(Rnd() * 10))
    Label3.Text = CStr(Int(Rnd() * 10))

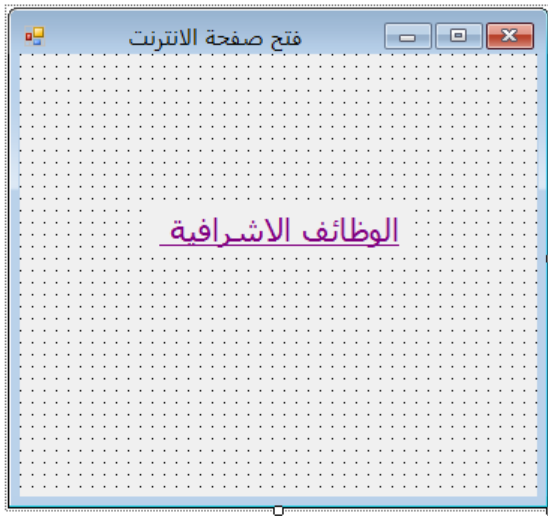
    If (Label1.Text = "7") Or (Label2.Text = "7") Or (Label3.Text = "7")
Then
        PictureBox1.Visible = True
    End If
End Sub
```

ثانياً: النقر المزدوج على زر Stop وكتابة التعليمات البرمجية التالية:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    End
End Sub
End Class
```

ورقة عمل إثرائية (2)

الموضوع: استخدام المكون LinkLabel



الفكرة الرئيسية:

إضافة رابط لصفحة انترنت لنموذج Form.

الخطوة الأولى:

إنشاء نموذج باسم (فتح صفحة الانترنت)

الخطوة الثانية:

إضافة الكائنات:

المكون LinkLabel ، مع تغيير الخاصية LinkVisited الى True لتغيير لون النص في حالة زيارة الموقع على الانترنت.

الخطوة الثالثة:

كتابة التعليمات البرمجية

الضغط المزدوج على المكون LinkLabel ستنقل إلى منطقة التعليمات البرمجية تحت الحدث LinkClicked .
الحالة الأولى: عند الضغط على الرابط ينتقل الى الموقع المكتوب بالتعليمة البرمجية بين (" ")
بغض النظر عن المتصفح المستخدم.

```
System.Diagnostics.Process.Start("http://www.hasobkw.net/pages/courses.aspx")
```

الحالة الثانية: عند الضغط على اللينك ينتقل الى الموقع المكتوب بالتعليمة البرمجية بين (" ")
باتباع المتصفح المكتوب بالتعليمة البرمجية وهنا تم تحديد Inter Explorer

```
System.Diagnostics.Process.Start("IExplore.exe", "http://www.hasobkw.net/pages/courses.aspx")
```

الحالة الثالثة: الطريقة نستطيع استخدامها مع كل البرامج فمثلا إذا كان لدينا ملف اسمه Textfile.txt نريد أن نفتح ببرنامج على القرص C: Word نستخدم التعليمة البرمجية:

```
System.Diagnostics.Process.Start("Winword.exe", "c:\Textfile.txt")
```

ورقة عمل إثرائية (3)

الموضوع: التعامل مع القوائم Menu باستخدام MenuStrip وأشرطة ToolStrip

الفكرة الرئيسية:

إضافة القوائم للبرنامج والتعديل عليها باستخدام المكون MenuStrip

الخطوة الأولى:

إنشاء مشروع جديد باسم MyMenu

الخطوة الثانية:

إضافة الكائنات:

1- إضافة Label عدد (2)

2- إضافة PrintDialog من Toolbox

3- MenuStrip من Toolbox إضافة القوائم التالية:

- Time يظهر الوقت الحالي في Label1.
- Date يظهر التاريخ الحالي في Label2.
- Print يستدعي صندوق الحوار الطباعة.

- تحديد مفاتيح الوصول السريع التي تخصص للاستخدام بحرف محدد مع مفتاح Alt بإضافة العلامة & في الخاصية Text قبل الحرف المراد (مثال &Print:)

الخطوة الثالثة:

كتابة التعليقات البرمجية

```
Private Sub ToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ToolStripMenuItem.Click
    Label1.Text = TimeString
End Sub
Private Sub ToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ToolStripMenuItem.Click
    Label2.Text = DateString
End Sub
Private Sub ToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ToolStripMenuItem.Click
    PrintDialog1.ShowDialog()
End Sub
```

تضم مجموعة الأدوات Dialogs مجموعة من الأدوات يمكن استخدامها كما بالأمثلة التالية:
مثال : لتلوين خلفية Label1 باستخدام الأداة ColorDialog

ColorDialog1.ShowDialog()

Label1.BackColor = ColorDialog1.Color

: مثال فتح ملف rtf في صندوق النصوص الغنية RichTextBox باستخدام الأداة OpenFileDialog

OpenFileDialog1.ShowDialog()

RichTextBox1.LoadFile(OpenFileDialog1.FileName)

: مثال حفظ محتوى نص في صندوق النصوص الغنية RichTextBox1 باستخدام الأداة SaveDialog

SaveFileDialog1.ShowDialog

RichTextBox1.SaveFile(SaveFileDialog1.FileName)

ورقة عمل إثرائية (4)

الموضوع: العمليات الرياضية البسيطة + ، - ، * ، /

الفكرة الرئيسية:

إنشاء آلة حاسبة بسيطة.

الخطوة الأولى:

إنشاء نموذج باسم (العمليات الرياضية البسيطة)

الخطوة الثانية:

إضافة الكائنات:

1- TextBox عدد (2)

2- GroupBox عدد (1)

3- أزرار Radio Button عدد (4)

4- Label عدد (4)

يتم ضبط الخصائص كما بالشكل السابق.

الخطوة الثالثة:

كتابة التعليمات البرمجية:

- الإعلان عن المتغير على مستوى النموذج ثم الضغط المزدوج على كل زر Radio Button وكتابة

التعليمات البرمجية التالية:

```
Public Class Form1
    Dim x, y As Double

    Private Sub RadioButton1_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton1.CheckedChanged
        If IsNumeric(TextBox1.Text) And IsNumeric(TextBox2.Text) Then
            x = TextBox1.Text
            y = TextBox2.Text
            Label1.Text = x + y
        Else
            MsgBox("رقمية قيمة ادخل ")
        End If
    End Sub
End Class
```

حل آخر

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged

    Try
        x = TextBox1.Text
        y = TextBox2.Text
        Label1.Text = x + y

    Catch ex As Exception
        MsgBox("رقمية قيمة ادخل ")
    End Try
End Sub
```

مراعاة تغير العملية الحسابية لكل RadioButton2,3,4 (+ أو - أو * أو /)

ورقة عمل إثرائية (5) الموضوع: العمليات الرياضية المتقدمة

الفكرة الرئيسية:

إنشاء آلة حاسبة متقدمة.

الخطوة الأولى:

إنشاء مشروع جديد باسم Adv. Math

الخطوة الثانية:

إضافة الكائنات:

1- TextBox عدد (2)

2- groupBox عدد (1)

3- أزرار Radio Button عدد (4)

4- Label عدد (4)

يتم ضبط الخصائص كما بالشكل السابق.



```
Public Class Form1
    Dim x, y As Double

    Private Sub RadioButton1_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton1.CheckedChanged

        If IsNumeric(TextBox1.Text) And IsNumeric(TextBox2.Text) Then
            x = TextBox1.Text
            y = TextBox2.Text
            Label1.Text = x Mod y
        Else
            MsgBox("رقمية قيمة ادخل ")
        End If
    End Sub
End Sub
```

حل آخر

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged

    Try
        x = TextBox1.Text
        y = TextBox2.Text
        Label1.Text = x Mod y
    Catch ex As Exception
        MsgBox("رقمية قيمة ادخل ")
    End Try
End Sub
```

مراعاة تغير العملية الحسابية لكل RadioButton2,3 (^ أو &)

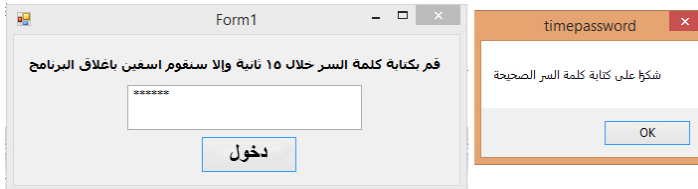
ورقة عمل إثرائية (6) الموضوع: استخدام المؤقت لتحديد فترة زمنية

الفكرة الرئيسية:

حماية برنامج بكلمة سر تدخل خلال فترة زمنية محددة.

الخطوة الأولى:

إنشاء مشروع جديد باسم Timed Password



الخطوة الثانية:

إضافة الكائنات:

- 1- Label عدد (1) ، مع ضبط خاصية Text إلى "قم بكتابة كلمة السر خلال 15 ثانية وإلا سنقوم بإغلاق البرنامج"
- 2- TextBox عدد (1) ، مع ضبط خاصية Password Char التابعة للمكون TextBox على "*"
- 3- مؤقت Timer، مع ضبط خاصية Interval على 15000 وخاصية Enabled على True
- 4- زر Button عدد (1)، مع ضبط خاصية Text على "دخول"
- 5- تغيير الخاصية AcceptButton التابعة للنموذج Form1 إلى Button1

الخطوة الثالثة:

كتابة التعليمات البرمجية

```
Public Class Form1

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

        If TextBox1.Text = "123456" Then
            Timer1.Enabled = False
            MsgBox("الصحيحة السر كلمة كتابة على شكراً")
        Else
            MsgBox("أخرى مرة حاول خاطئة، السر كلمة")
            TextBox1.Text = ""
        End If
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Timer1.Tick
        End
    End Sub
```

ورقة عمل إثرائية (7) الموضوع: تحريك الصورة

الفكرة الرئيسية:

إنشاء أزار لتحريك الصورة إلى أعلى وإلى أسفل
الخطوة الأولى:

إنشاء مشروع جديد باسم My Moving Icon
الخطوة الثانية:

إضافة الكائنات:

1- نموذج Form، مع ضبط خاصية Text إلى "التحريك البسيط للرسوم"
2- زر Button عدد (2)، مع ضبط خاصية Text للزر الأول إلى "تحريك لأعلى"، وللزر الثاني إلى "تحريك للأسفل"

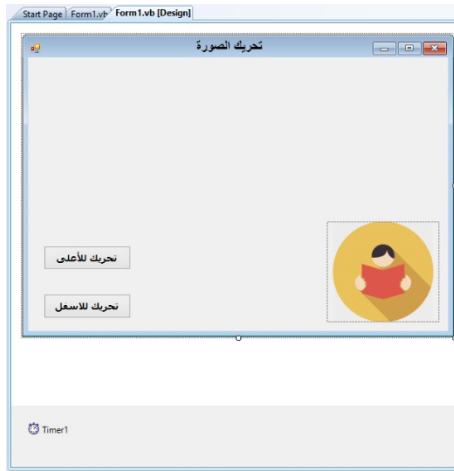
3- Picture Box : عدد (1) ، مع إضافة صورة ونعدل الخاصية SizeMode إلى StretchImage

4- مؤقت Timer، مع ضبط الخاصية Interval للمؤقت إلى 75.
الخطوة الثالثة:

كتابة التعليمات البرمجية

- الإعلان عن متغير GoingUp على مستوى المشروع، مع ضبط المتغير على True .

```
Public Class Form1
Dim GoingUp As Boolean
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
GoingUp = True
Timer1.Enabled = True
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
GoingUp = False
Timer1.Enabled = True
End Sub
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
If GoingUp = True Then
If PictureBox1.Top > 10 Then
PictureBox1.Location = New Point _
(PictureBox1.Location.X - 10, _
PictureBox1.Location.Y - 10)
End If
Else
If PictureBox1.Top < (Me.Size.Height - 75) Then
PictureBox1.Location = New Point _
(PictureBox1.Location.X + 10, _
PictureBox1.Location.Y + 10)
End If
End Sub
End Sub
```



يمكن تحريك الصورة لأعلى باستخدام:
PictureBox1.Top -= 10
يمكن تحريك الصورة لليسار باستخدام:
PictureBox1.Left -= 10

ورقة عمل إثرائية (8) الموضوع: عملية التلوين

الفكرة الرئيسية:

تلوين خلفية البرنامج، بخلط ثلاثة اللون الأحمر واللون الأخضر واللون الأزرق بالدالة **FromArgb**

الخطوة الأولى:

إنشاء مشروع جديد باسم **Coloring**
الخطوة الثانية:

إضافة الكائنات

1- Label عدد (4)

2- زر Button عدد (1)

3- Textbox عدد (3)

يتم ضبط الخصائص كما بالشكل السابق.

الخطوة الثالثة:

كتابة التعليمات البرمجية:

```
Color Public Class Form1
```

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

```
Dim rgb1, rgb2, rgb3 As Integer
```

```
rgb1 = TextBox1.Text
```

```
rgb2 = TextBox2.Text
```

```
rgb3 = TextBox3.Text
```

```
Me.BackColor = Color.FromArgb(rgb1, rgb2, rgb3)
```

```
End Sub
```

```
End Class
```

ملاحظة:

- يمكن التلوين باستخدام اللون مباشرة باستخدام التعليمات البرمجية التالية:
- التعليمة التالية لتغيير لون الخلفية باللون الأزرق
- `TextBox1.BackColor = Color.Blue`
- التعليمة التالية لتغيير لون النص باللون الاحمر
- `TextBox1.ForeColor=Color.Red`

ورقة عمل إثرائية (9) الموضوع: متصفح انترنت

الفكرة الرئيسية:

تصميم متصفح انترنت

الخطوة الأولى:

إنشاء مشروع جديد باسم My_WebBrowser

الخطوة الثانية:

إضافة الكائنات

- 1- Label عدد (1)
- 2- TextBox عدد (1)
- 3- Button عدد (1)
- 4- WebBrowser

يتم ضبط الخصائص كما بالشكل السابق.

الخطوة الثالثة:

كتابة التعليمات البرمجية بالضغط المزدوج على الزر Go:

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        WebBrowser1.Navigate(TextBox1.Text)
    End Sub

End Class
```

حيث أن:

تستخدم لفتح الصفحة المحددة بين القوسين كعامل

Navigate

ورقة عمل إثرائية (10) الموضوع: البحث في النص

الفكرة الرئيسية:

البحث عن نص معين داخل فقرة.

الخطوة الأولى:

إنشاء مشروع جديد بإسم Search

الخطوة الثانية:

إضافة الكائنات

1. Label عدد (2)

2. TextBox عدد (2) ، مع مراعاة كتابة أي فقرة في TextBox الأول،

وكتابة الكلمة المراد البحث عنها في TextBox الثاني.

3. زر أمر Button عدد (1)

يتم ضبط الخصائص كما بالشكل السابق.

الخطوة الثالثة:

كتابة التعليمات البرمجية

```
Public Class Form1

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

        Dim i As Integer
        i = InStr(TextBox2.Text, TextBox1.Text, CompareMethod.Binary)
        If i > 0 Then
            TextBox2.Select(i - 1, Len(TextBox1.Text))
            TextBox2.Focus()
            TextBox2.SelectedText = ""
        Else
            MsgBox("لم يتم العثور على شيء")
        End If

    End Sub
End Class
```

ورقة عمل إثرائية (11)

الموضوع: إضافة أشرطة الأدوات Toolbars باستخدام ToolStrip

الفكرة الرئيسية:

إضافة شريط أدوات قياسي – شريط قوائم قياسي مع كتابة التعليمات البرمجية للأدوات و الأوامر

الخطوة الأولى:

إنشاء مشروع جديد باسم ToolStrip

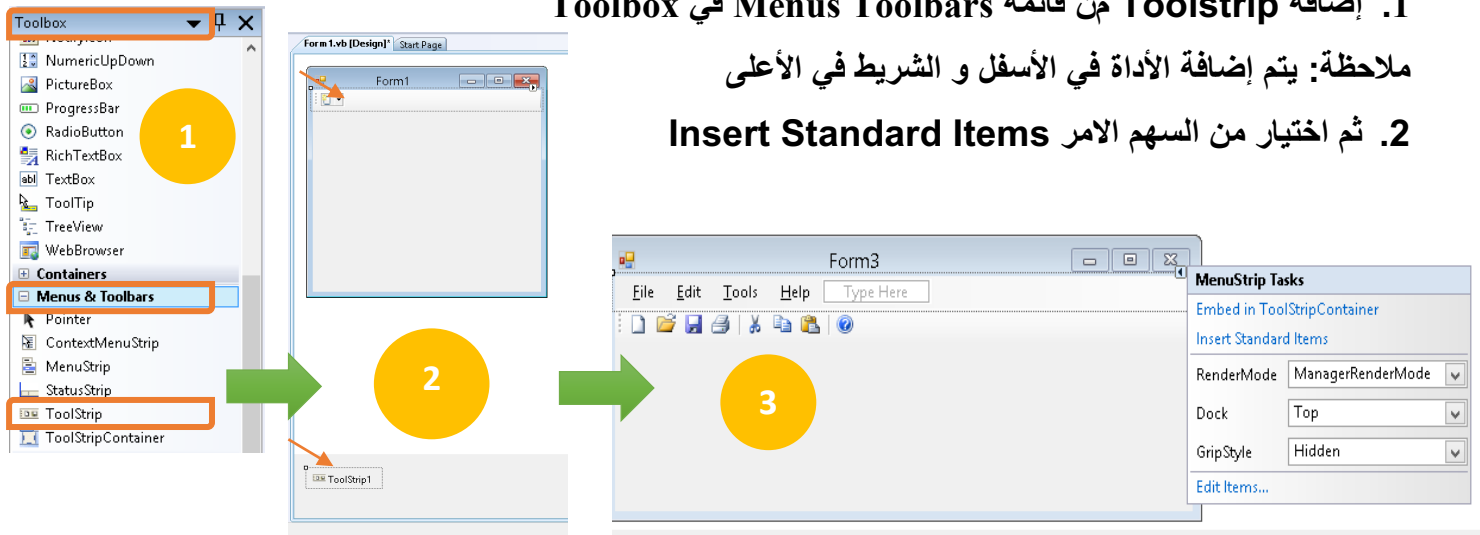
الخطوة الثانية:

إضافة الكائنات:

1. إضافة ToolStrip من قائمة Menus Toolbars في Toolbox

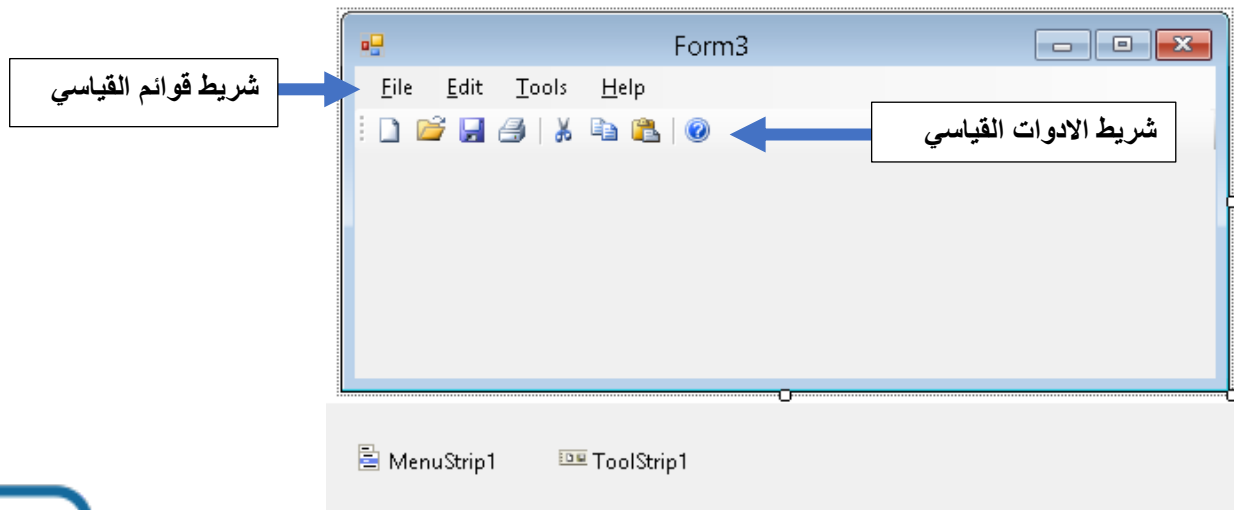
ملاحظة: يتم إضافة الأداة في الأسفل و الشريط في الأعلى

2. ثم اختيار من السهم الامر Insert Standard Items



3. سوف يتم إضافة الأدوات الافتراضية في الشريط و يمكن التطبيق أيضا على Menu Strip لإدراج شريط قوائم

قياسي و كتابة التعليمات البرمجية للأدوات و الأوامر كما تم ذكره سابقا كما هو موضح في الشكل التالي :



ورقة عمل إثرائية (12) الموضوع: انشاء القوائم المختصرة pop menu :

الفكرة الرئيسية:

إضافة قائمة مختصرة ببعض الأوامر عند الضغط على بالزر الأيمن على العنصر

الخطوة الأولى:

إنشاء مشروع جديد باسم Context Menu Strip

الخطوة الثانية:

إضافة الكائنات:

1. إضافة Rich Textbox من Toolbox

2. إضافة اداة Context Menu Strip من

menus&Toolbars

ملاحظة: يتم إضافة الأداة في الأسفل و الشريط في الأعلى

3. ثم تضاف الأوامر المطلوبة مثال (Copy-

(Past-Delete

الخطوة الثالثة:

1. من خصائص العنصر (Rich Textbox

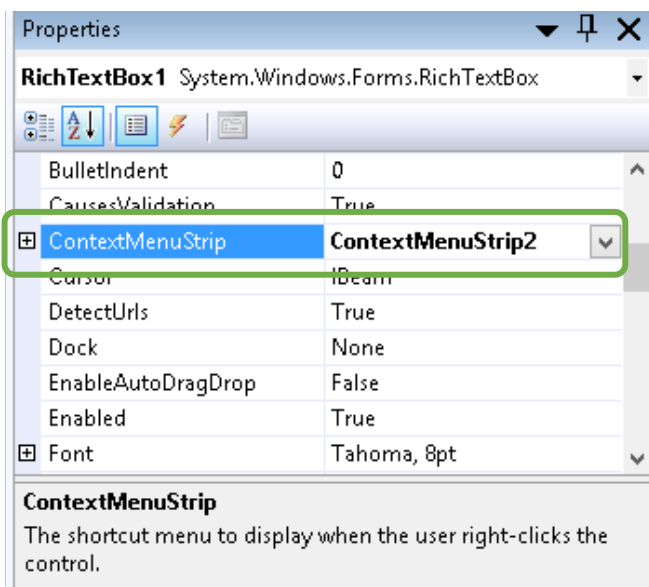
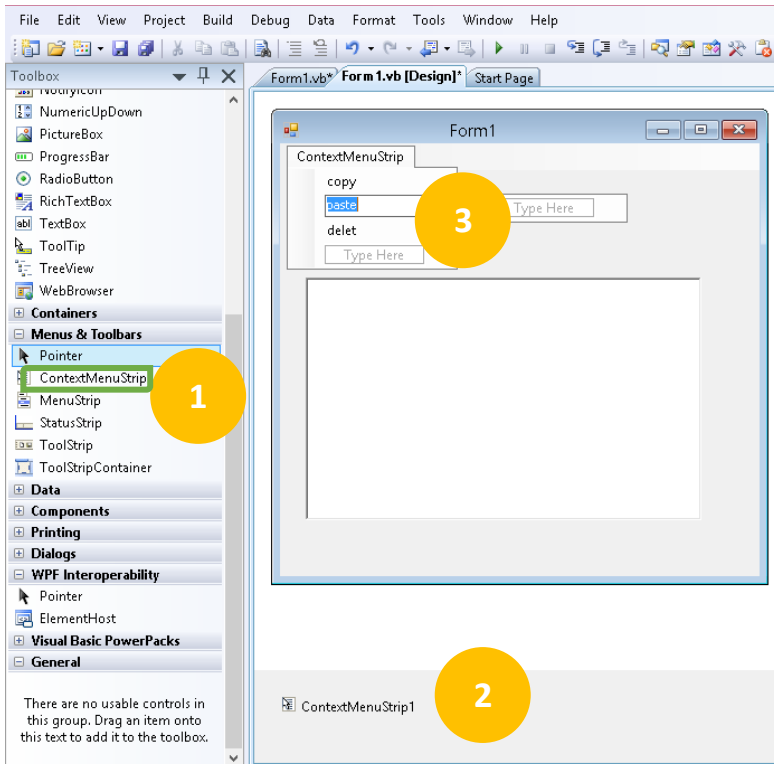
الذي تم تحديده للضغط بالزر الأيمن تغيير خصائصه

2. من خصائص Rich text يتم

تحديد Context Menu Strip للربط بين

الزر الأيمن و الكائن

كما هو في الشكل التالي :



ورقة عمل إثرائية (13) الموضوع: انشاء progress Bar

الفكرة الرئيسية:

إضافة progress Bar عند تحميل الصفحة يظهر النسبة المئوية للتحميل والتزايد بمعدل واحد.

الخطوة الأولى:

إنشاء مشروع جديد باسم progress Bar

الخطوة الثانية:

إضافة الكائنات:

1. إضافة أداة Button – label
2. إضافة Timer
3. إضافة progress Bar

الخطوة الثالثة:

1. تعديل خصائص Timer :

Enabled : True

Interval : 150 (مقدار التزايد يمكن تغييره على حسب السرعة)

2. كتابة التعليمات البرمجية :

```
Public Class Form1
```

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

```
Me.Hide()  
Form2.Show()
```

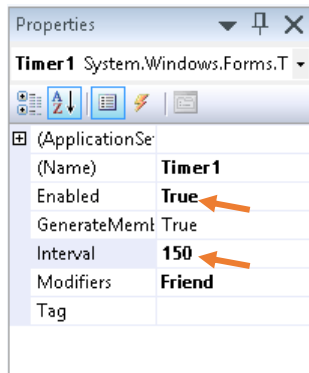
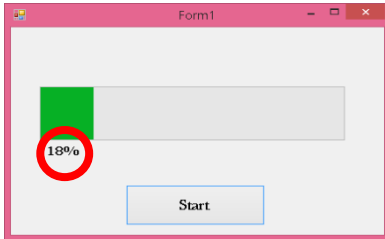
```
End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
```

```
ProgressBar1.Increment(1)  
If ProgressBar1.Value = 100 Then  
Label1.text = "done"  
End If  
Label1.Text = ProgressBar1.Value & "%"
```

```
End Sub
```

```
End Class
```



ورقة عمل إثرائية (14)

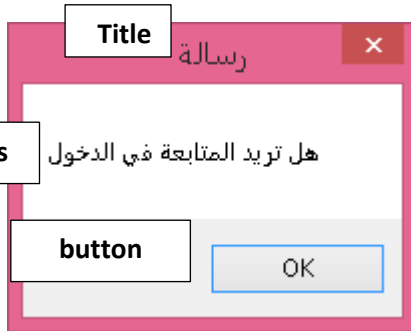
الموضوع: صندوق الحوار Msg Box

الفكرة الرئيسية:

إضافة صندوق الحوار **Msg Box** و التحكم بعدد الأزرار
التعليمة البرمجية: **Msgbox("Messaged",code,"Title")**

موضح بالجدول التالي

++ للتحكم في عدد الأزرار و شكلها عن طريق التحكم في الأرقام الموجودة في خانة رقم الكود بدالة **Msg Box**



الرقم	اختيار النمط	اشكال الأزرار
٠	MsgBoxStyle.OkOnly	
١	MsgBoxStyle.OkCancel	
٢	MsgBoxStyle.AbortRetryIgnore	
٣	MsgBoxStyle.YesNoCancel	
٤	MsgBoxStyle.YesNo	
٥	MsgBoxStyle.RetryCancel	

الرقم	النمط	العلامة
٠	MsgBoxStyle.SystemModal	بدون علامة
١٦	MsgBoxStyle.Critical	⛔
٣٢	MsgBoxStyle.Question	❓
٤٨	MsgBoxStyle.Exclamation	⚠
٦٤	MsgBoxStyle.Information	ℹ

التحكم في الزر الافتراضي الذي يظهر بصندوق الحوار

الرقم	النمط	الزر الافتراضي
٠	MsgBoxStyle.DefaultButton1	الزر الاول
٢٥٦	MsgBoxStyle.DefaultButton2	الزر الثاني
٥١٢	MsgBoxStyle.DefaultButton3	الزر الثالث



الخطوة الأولى:

إنشاء مشروع جديد باسم **Msg Box**

الخطوة الثانية:

كتابة التعليمات البرمجية :

5: زرین Retry و Cancel

32 : ؟

256 : الزر الافتراضي

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
MsgBox ("رسالة", 5 + 32 + 256, "بالدخول المتابعة تريد هل")
```

ورقة عمل إثرائية (15) الموضوع: الترجمة الصوتية للنصوص

الفكرة الرئيسية:

إضافة صندوق حوار يقبل النصوص وترجمتها صوتيا .

الخطوة الأولى:

إنشاء مشروع جديد باسم Text Voice

الخطوة الثانية:

إضافة الكائنات:

1. إضافة أداة Rich Text box
2. إضافة أداة Button

الخطوة الثالثة:

كتابة التعليمات البرمجية :

```
Public Class Form2

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

        Dim voice

        voice = CreateObject("sapi.spvoice")
        voice.speak(RichTextBox1.Text)

    End Sub
End Class
```

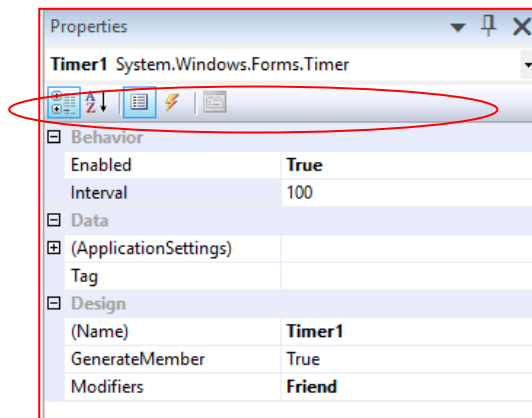
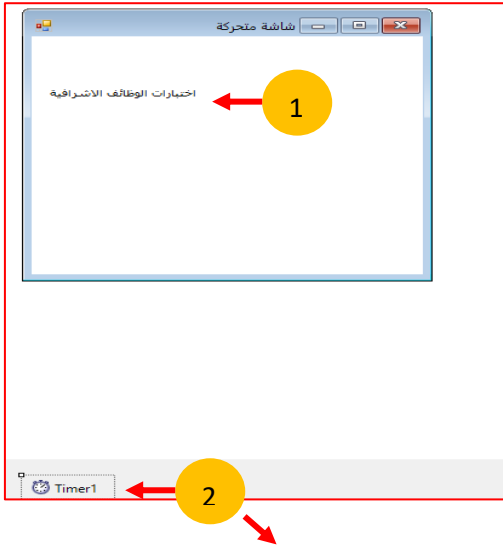
ورقة عمل إشرافية (16) الموضوع : نص متحرك

الفكرة الرئيسية

تحريك نص بشكل مستمر بشكل أفقي عند حدث تحميل النموذج .
الخطوة الأولى :

انشاء نموذج جديد .
الخطوة الثانية :

1. اضافة كائن نص label وتغيير خاصية text إلى " اختبارات الوظائف الاشرافية "
2. اضافة كائن timer للنموذج وتغيير خاصية enabled إلى true .



الخطوة الثالثة :

كتابة التعليمات البرمجية للكائن timer1

```
Private Sub Timer1_Tick(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
    If Label1.Left < Me.Width Then
        Label1.Left += 6
    Else
        Label1.Left = -Label1.Width
    End If
End Sub
```

ورقة عمل إشرائية (17) الموضوع : نص وامض

الفكرة الرئيسية

تغيير لون نص بشكل مستمر من الأسود إلى الأحمر ومن الأحمر إلى الأسود عند حدث تحميل النموذج

الخطوة الأولى :

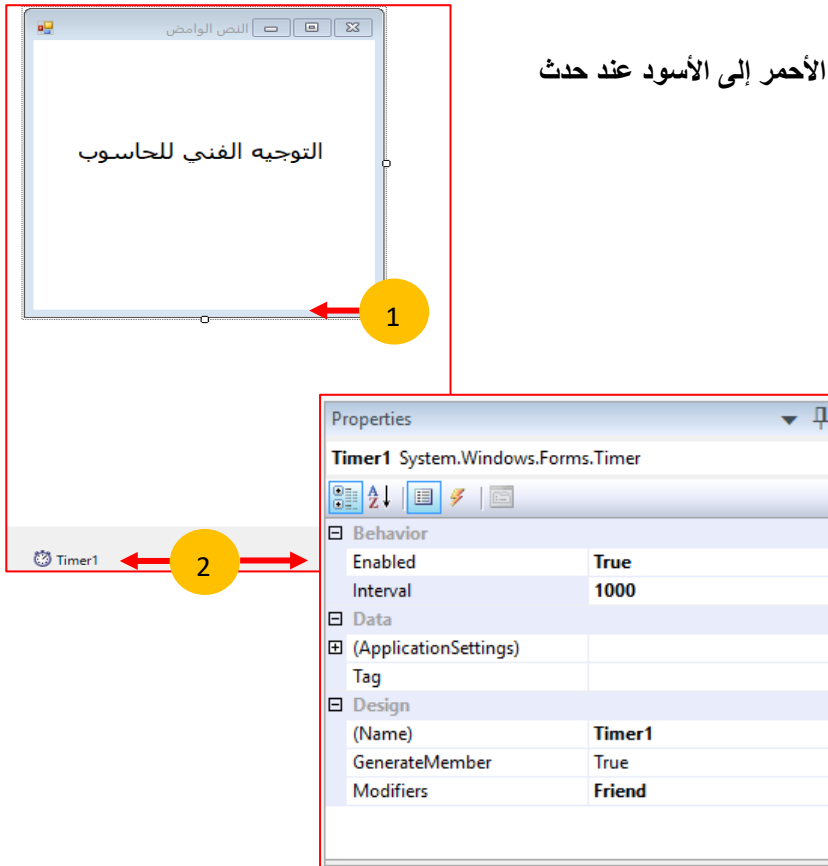
انشاء نموذج جديد .

الخطوة الثانية :

1. اضافة كائن نص label وتغيير خاصية text " إلى لتوجيه الفني للحاسوب " .
2. اضافة كائن timer للنموذج وتغيير خاصية enabled إلى true
3. وتغيير خاصية interval إلى 1000 .

الخطوة الثالثة :

كتابة التعليمات البرمجية للكائن timer1

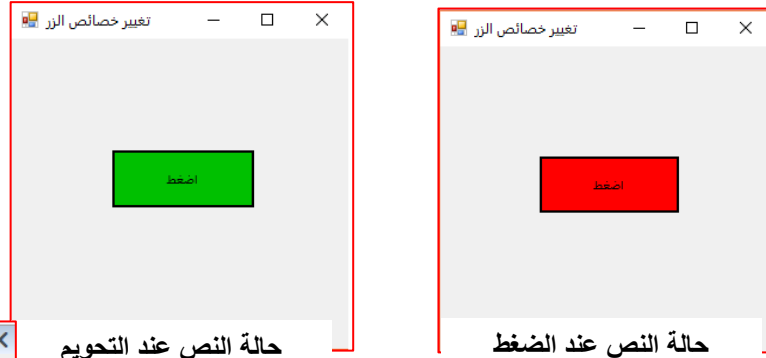


```
Private Sub Timer1_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    If Label1.ForeColor = Color.Black Then
        Label1.ForeColor = Color.Red
    Else
        Label1.ForeColor = Color.Black
    End If
End Sub
```

ورقة عمل إثرائية (18) الموضوع : زر متغير اللون

الفكرة الرئيسية

انشاء زر يتغير لونه عند التحويم ويتغير لونه عند الضغط عليه .

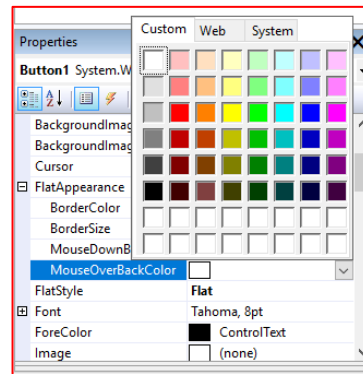
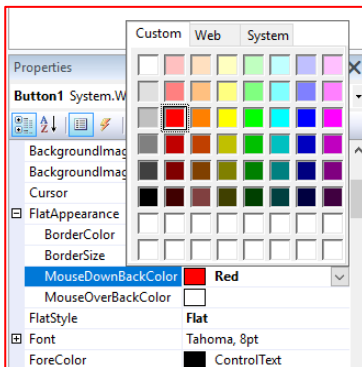
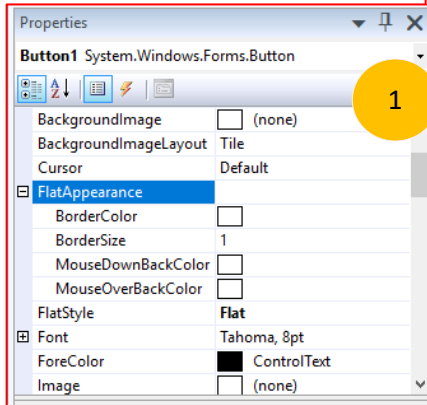


الخطوة الأولى :

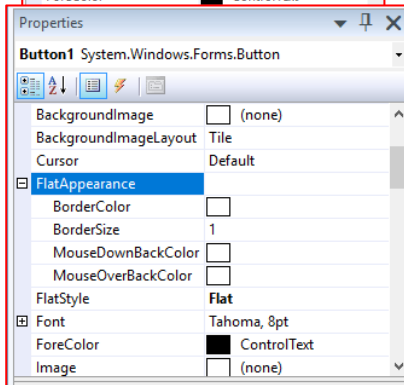
انشاء نموذج جديد .

الخطوة الثانية :

تغيير خاصية mouse over back color : flat appearance
اللون الأخضر .
mouse down back color , إلى اللون الأحمر .



- تغيير الخاصية flat style إلى flat .



ورقة عمل إثرائية (19)

الموضوع: استخدام الأداة DateTimePicker لحساب فارق الأيام بين تاريخين

الفكرة الرئيسية:

لاظهار التاريخ والوقت

الخطوة الأولى:

انشاء نموذج جديد

الخطوة الثانية:

إضافة الكائنات وضبط خصائصها:

1. نضيف label عدد 3 ونضبط خاصية text Label1 و Label2 أما Label3 تكون فارغة

2. نضيف DateTimePicker عدد 2 ونضبط

خاصية Format و CustomFormat .
3. نضيف Bottun واحد ونضبط خاصية text لتكون عدد الأيام.

الخطوة الثالثة:

كتابة التعليمات البرمجية

أولاً: النقر المزدوج على زر عدد الأيام وكتابة التعليمات البرمجية التالية:

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

```
Dim firstDate, secondData As DateTime
```

```
firstDate = DateTimePicker1.Value
```

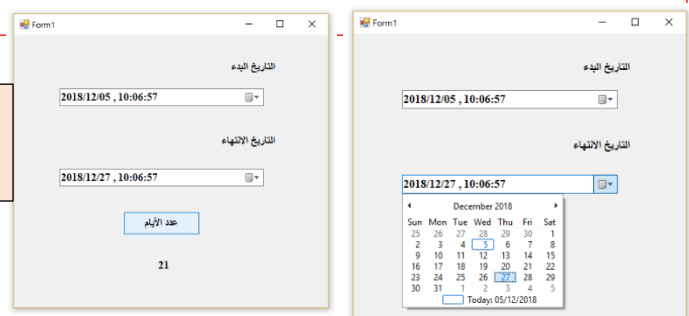
```
secondData = DateTimePicker2.Value
```

```
Label3.Text = secondData.Subtract(firstDate).Days
```

```
End Sub
```

يمكن حساب الفرق بالسنوات بين التاريخين باستخدام :

```
secondData.Year() - firstDate.Year()
```



ورقة عمل إثرائية (20) الموضوع: عرض التاريخ في شريط الحالة

الفكرة الرئيسية:

استخدام شريط الحالة في عرض التاريخ الحالي

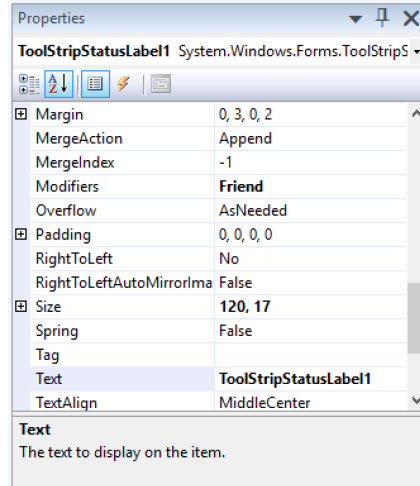
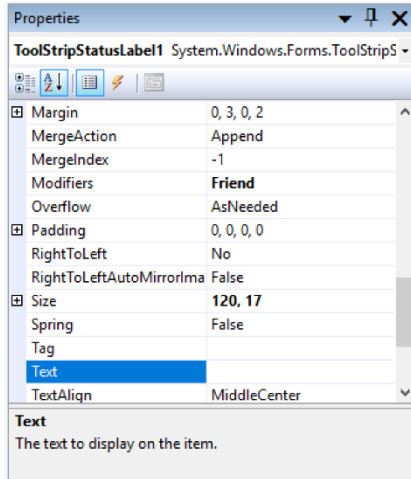
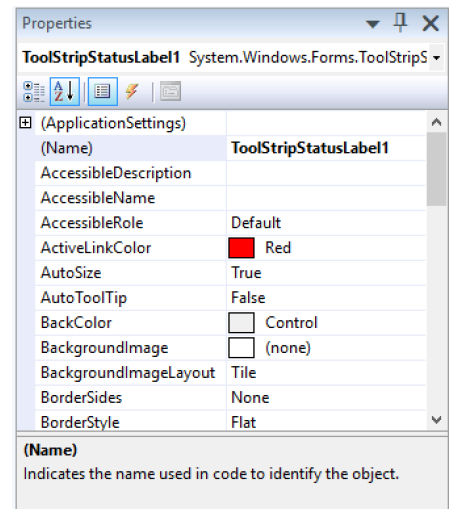
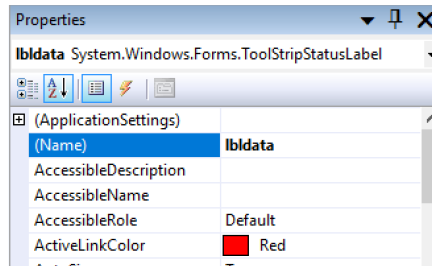
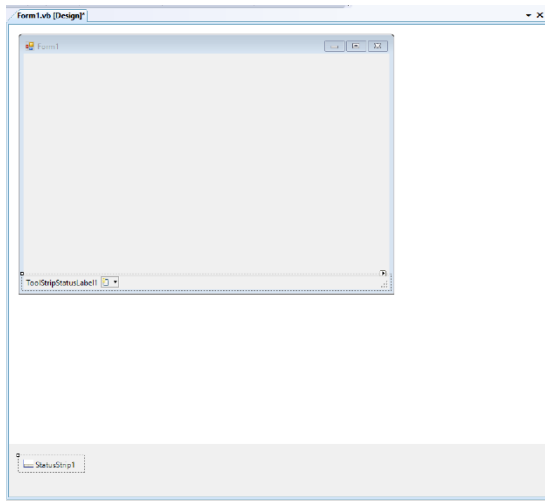
الخطوة الأولى:

انشاء نموذج جديد

الخطوة الثانية:

إضافة الكائنات وضبط خصائصها:

1. نضيف statusStrip1 ونضيف ToolStripStatusLabel1 ونغير خاصية الاسم لتكون lbldate وخاصية text تكون فارغة .



الخطوة الثالثة:

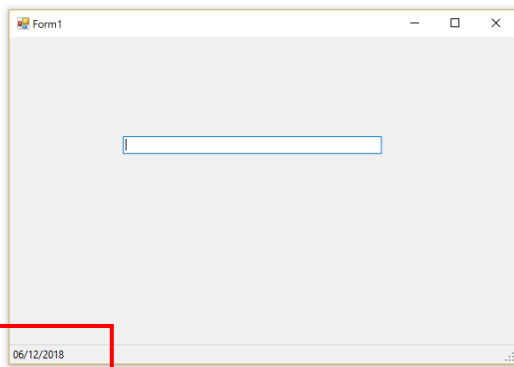
كتابة التعليمات البرمجية

أولاً: النقر المزدوج على النموذج وكتابة التعليمات البرمجية التالية:

```
Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
```

```
lbldate.Text = Now.ToShortDateString
```

```
End Sub
```



ورقة عمل إثرائية (21) الموضوع: عرض إجمالي عدد حروف النص في شريط الحالة

الفكرة الرئيسية:

عرض إجمالي عدد حروف النص المكتوب في مربع النص في شريط الحالة

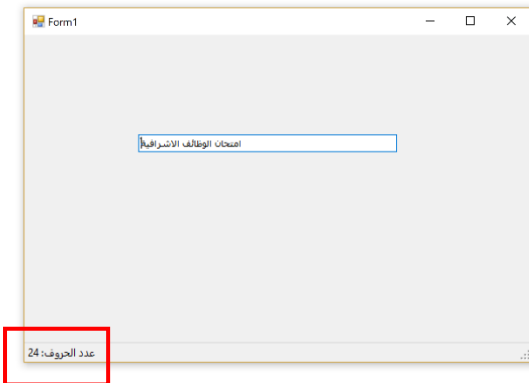
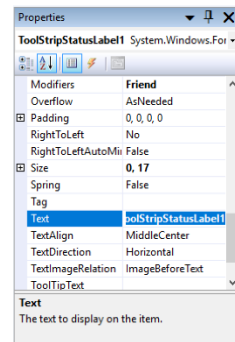
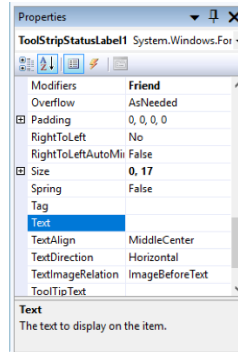
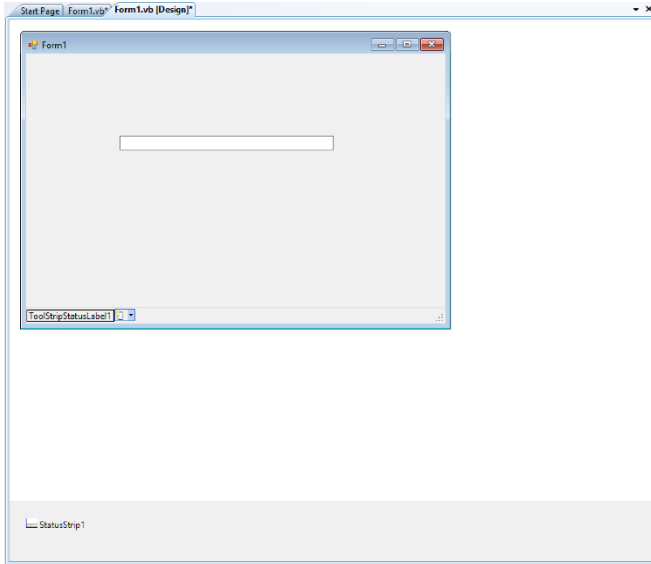
الخطوة الأولى:

إنشاء نموذج جديد

الخطوة الثانية:

إضافة الكائنات وضبط خصائصها:

1. نضيف statusStrip1 ونضيف داخله ToolStripStatusLabel1 ونغير خاصية text لتكون فارغة .



2. نضيف أداة Textbox1 .

الخطوة الثالثة:

كتابة التعليمات البرمجية
أولاً: النقر المزدوج على الأداة Textbox1 وكتابة التعليمات البرمجية التالية:

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox1.TextChanged

    ToolStripStatusLabel1.Text = " : الحروف عدد " &
    Len(TextBox1.Text)

End Sub
```

ورقة عمل إثرائية (22)

الموضوع: حساب كم تبلغ من العمر

الفكرة الرئيسية:

ادخال اليوم والشهر والسنة ليوم ميلادك ثم ادخال أي تاريخ مراد احتساب العمر، في حال كانت سنة الميلاد أكبر من سنة المراد احتساب العمر لها ظهر رسالة بتوضيح ان سنة الميلاد أكبر من السنة المراد احتساب العمر لها وفي حال عدم ادخال البيانات عند الضغط على زر احسب تظهر مسح بإدخال القيم في جميع الحقول ، كذلك عند الضغط على زر جديد يقوم البرنامج بمسح الأرقام المدخلة ، ويجب أن تكون المدخلات قيم رقمية فقط

الخطوة الأولى:

إضافة الأدوات (6 Textbox) – (3 Button) (9 label)

الخطوة الثانية : كتابة التعليمة البرمجية في Button2 جديد

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    Label17.Text = ""
    Label18.Text = ""
    Label19.Text = ""
    TextBox4.Focus()
End Sub
```

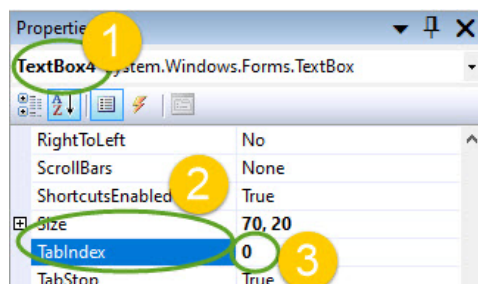
حل لآخر

```
For Each contor In Me.Controls
    If TypeName(contor) = "TextBox" Then
        contor.clear()
    End If
Next
```

كتابة التعليمة البرمجية عند تحميل النموذج

```
Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    TextBox4.Focus()
End Sub
```

أو من خصائص Textbox



- كتابة التعليمية البرمجية في Button1 (أبدأ)

```

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

Dim d1, d2 As Integer
    Dim m1, m2 As Integer
    Dim y1, y2 As Integer

    If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = ""
Or TextBox4.Text = "" Or TextBox5.Text = "" Or TextBox6.Text = "" Then
        MsgBox("الحقول جميع في البيانات أدخل")
        Exit Sub
    End If

    d1 = Val(TextBox1.Text)
    d2 = Val(TextBox4.Text)
    m1 = Val(TextBox2.Text)
    m2 = Val(TextBox5.Text)
    y1 = Val(TextBox3.Text)
    y2 = Val(TextBox6.Text)

    If d1 < d2 Then
        d1 = d1 + 30
        m1 = m2 - 1
    End If

    If m1 < m2 Then
        m1 = m1 + 12
        y1 = y1 - 1
    End If

    If y1 < y2 Then
        MsgBox("العمر احتساب المراد السنة تاريخ من أكبر الميلاد سنة تاريخ")
        Exit Sub
    End If

    Label7.Text = d1 - d2
    Label8.Text = m1 - m2
    Label9.Text = y1 - y2

End Sub

```

ورقة عمل إثرائية (23)

الموضوع: التحكم في خصائص TextBox

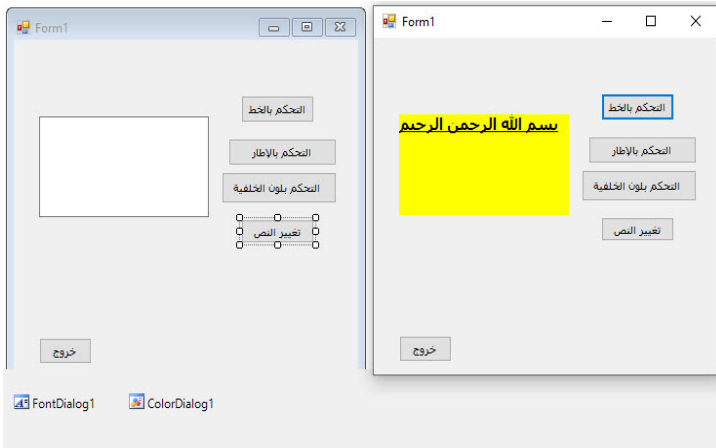
الفكرة الرئيسية:

التحكم في خصائص Textbox من خلال البرمجة
تغيير تنسيقات الخط ، إطار ولون خلفية الأداة TextBox ،

مسح النص ، TextBox1 تعديل في خاصية

ReadOnly = True

حتى لا يتغير النص إلا من خلال زر Button4 زر تغيير النص



```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        FontDialog1.ShowColor = True
        FontDialog1.ShowDialog()
        TextBox1.Font = FontDialog1.Font
        TextBox1.ForeColor = FontDialog1.Color
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        If TextBox1.BorderStyle = BorderStyle.None Then
            TextBox1.BorderStyle = BorderStyle.Fixed3D
        Else
            TextBox1.BorderStyle = BorderStyle.None
        End If
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        ColorDialog1.ShowDialog()
        TextBox1.BackColor = ColorDialog1.Color
    End Sub
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        Dim m As String
        m = InputBox("الجدید النص أدخل")
        TextBox1.Text = m
    End Sub
    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
        Me.Hide()
        Form2.Show()
    End Sub
End Class
```

ورقة عمل إثرائية (24) الموضوع: تحويل العملات

الفكرة الرئيسية:

عرض واجهة لتحويل المبلغ المكافئ بالدينار لقيمة مبلغ بعملة أخرى مع ظهور شكل العملة

الخطوة الأولى:

انشاء نموذج جديد وإضافة الأدوات Label عدد 3 – Textbox عدد 2 – combobox – picturebox – button

الخطوة الثانية:

كتابة التعليمات البرمجية
أولاً: النقر المزدوج على زر تحويل وكتابة التعليمات البرمجية التالية:

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

    If Not IsNumeric(TextBox1.Text) Then
        MsgBox("رقمية قيمة ادخل")
        TextBox1.Clear()
        TextBox2.Clear()
        Exit Sub
    End If

    Dim Rate As Double

    Select Case ComboBox1.SelectedIndex
        Case 0
            Rate = 0.3
        Case 1
            Rate = 0.316
        Case Is = 2
            Rate = 0.4
        Case Else
            MsgBox("العملة تحيدي الرجاء ")
            Exit Sub
    End Select

    PictureBox1.Image = ImageList1.Images(ComboBox1.SelectedIndex)
    TextBox2.Text = Val(TextBox2.Text) * Rate

End Sub
End Class
```

ورقة عمل إثرائية (25) الموضوع: إعادة تعيين جميع المدخلات على الشاشة

الفكرة الرئيسية:

نقوم بالبحث عن جميع الأدوات داخل النموذج من النوع صندوق نص (TextBox) وإعادة تعيين قيمة الخاصية نص (text) لتساوي ""

الخطوة الأولى:

إنشاء نموذج جديد وإضافة 6 أدوات عنوان (Label) و 8 أدوات صندوق نص (TextBox) و زر (Button) وتعديل الخصائص ليتناسب مع الصورة

الخطوة الثانية:

النقر المزدوج على زر "مستخدم جديد" وكتابة التعليمات البرمجية التالية (بدون النص التوضيحي):

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

من خلال مصفوفة الأدوات داخل النموذج

```
For Each myControl As Control In Me.Controls
```

هل الأداة من النوع صندوق نص

```
If TypeName(myControl) = "TextBox" Then
```

نعم، قم بإعادة تعيين الخاصية نص

```
myControl.Text = ""
```

```
End If
```

```
Next
```

ورقة عمل إثرائية (26) الموضوع: توليد أحرف أبجدية عشوائية

الفكرة الرئيسية:

عند الضغط على زر **start** يتم توليد أحرف عشوائية من **A → Z** في **Label1** و عند ظهور حرف **B** تظهر الصورة في **PictureBox1**.

الخطوة الأولى:

انشاء نموذج باسم **Random Alpha**

الخطوة الثانية:

إضافة الكائنات

Button: عدد (1) ، واضبط خاصية **text** للزر الأول **Start**

Picture Box : عدد (1) ، مع مراعاة:

- تغيير الخاصية **Visible** الى **False** لعدم ظهور الصورة في بداية التشغيل.
- تعديل حجم الصورة من **SizeMode** لتصبح **Zoom**.

الخطوة الثالثة: كتابة التعليمات البرمجية

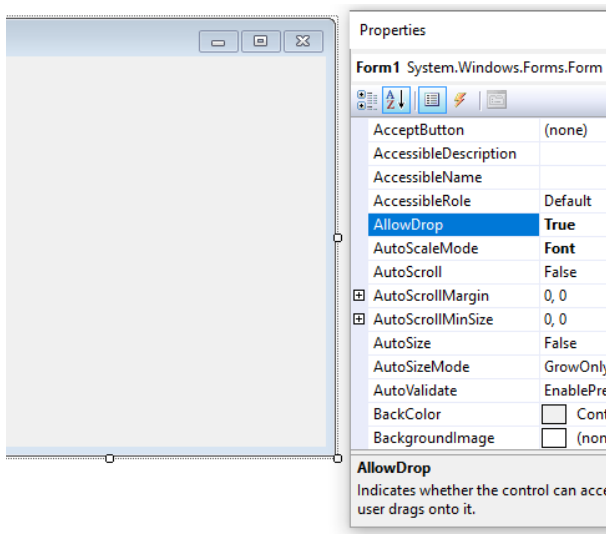
أولاً: النقر المزدوج على زر **Start** وكتابة التعليمات البرمجية التالية

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles Button1.Click
        Dim Rndm As New Random
        Dim chars As String = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        Label1.Text = chars(Rndm.Next(chars.Length)).ToString
        If Label1.Text = "B" Then
            pictureBox1.Visible = True
        Else
            pictureBox1.Visible = False
        End If
    End Sub
End Class
```


ورقة عمل إثرائية (27)

الموضوع: عرض صورة في خلفية النموذج بسحبها وإفلاتها فيه



الفكرة الرئيسية:

عرض صورة كخلفية للنموذج من أي ملف بسحبها وإفلاتها داخل النموذج

الخطوة الأولى:

إنشاء نموذج جديد، عدل الخاصية (AllowDrop) في النموذج لتصبح (True)

الخطوة الثانية:

نقوم بكتابة التعليمات البرمجية اللازم لتحديد نوع عملية الإفلات في الحدث (DragEnter) للنموذج (بدون النص التوضيحي):

```
Private Sub Form1_DragEnter(ByVal sender As Object, ByVal e As
System.Windows.Forms.DragEventArgs) Handles Me.DragEnter
    تحديد نوع عملية الافلات
    e.Effect = DragDropEffects.Copy
End Sub
```

الخطوة الثالثة: نقوم بكتابة التعليمات البرمجية اللازم لعرض الصورة عند إفلات الملف في النموذج من خلال الحدث (DragDrop) (بدون النص التوضيحي):

```
Private Sub Form1_DragDrop(ByVal sender As Object, ByVal e As
System.Windows.Forms.DragEventArgs) Handles Me.DragDrop
    نقوم باختبار ما إذا كان الملف قابل للعرض
    Try
        نقوم باختبار العنصر الأول في حال إذا تم إفلات أكثر من ملف
        Dim myFile As String = e.Data.GetData(DataFormats.FileDrop)(0)
        نقوم بتغيير صورة الخلفية بناءً على الملف الذي تم إفلاته
        Me.BackgroundImage = Image.FromFile(myFile)
    Catch ex As Exception
        في حال ما إذا كان الملف غير قابل للعرض، نظهر رسالة
        MsgBox ("للعرض قابل غير الملف")
    End Try
End Sub
```