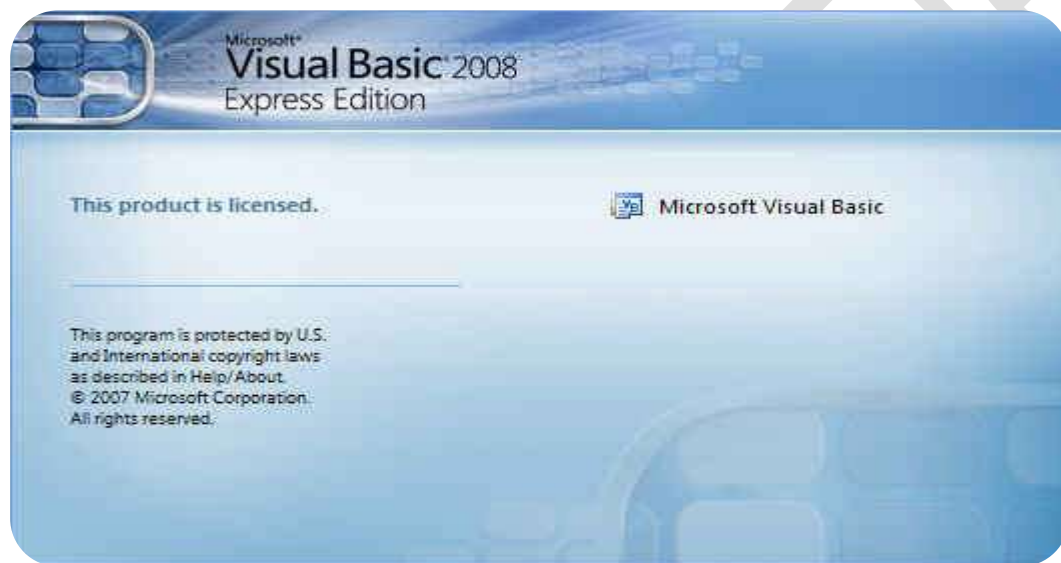


# LECTURE OF VISUAL BASIC 2008



Prepared By

Eng. M. Abou Elela

2010-2011

## بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

أقدم لكم كتابي لشرح برنامج فيجوال بيسك 2008  
من البداية إلى أول طريق الاحتراف خطوة بخطوة  
ويحتوي الكتاب على شرح وافى للبرنامج مع تمارين  
لكل جزء ليتمكنك من الاستفادة منها عمليا ولتكون  
دليلا على صحة المعلومة المشروحة

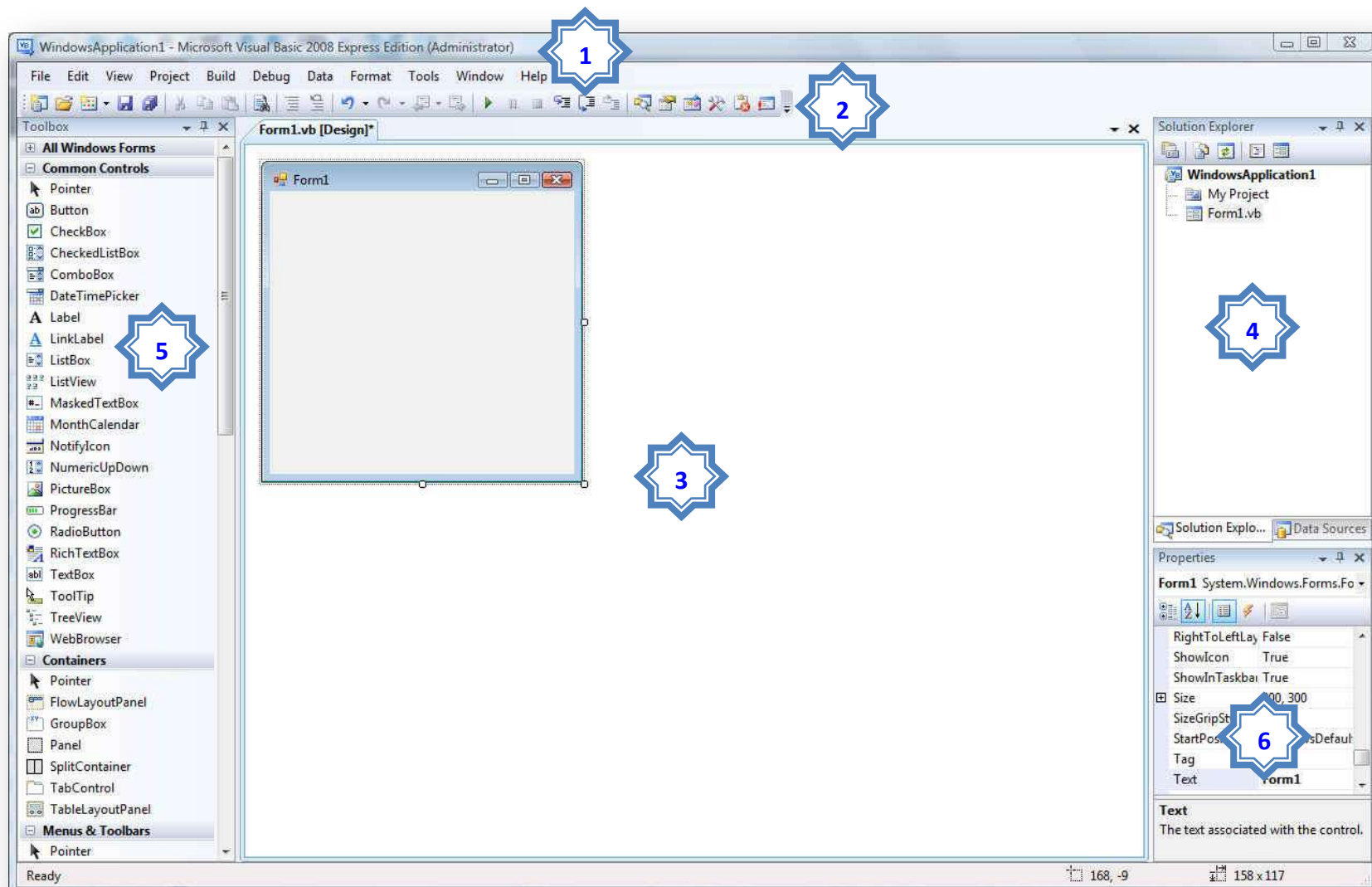
الكتاب مجاني لوجه الله تعالى

ولا أرجو منكم غير صالح الدعاء لي ولوالدي

مع تحياتي

مهندس / محمد أبو العلا

## واجهة البرنامج



هناك العديد من الأدوات التي سوف تراها في فيجوال بيسك 2008 والتي سوف نشرحها تباعا وهي

### شريط القوائم Menu Bar

وهو شريط للتعامل مع قوائم أوامر البرنامج مثله مثل باقي برامج مايكروسوفت أوفس

### شريط الأدوات Tool Bar

هو شريط بة مجموعة من اختصارات الأيقونات ( الرموز) للتعامل مع البرنامج من خلالها بسهولة واختصار

### المصمم Designer

وهي المنطقة التي يظهر فيها الفورم الذي نتعامل معه عند التصميم ومن خلالها يمكن إضافة أو إزالة محتويات الفورم

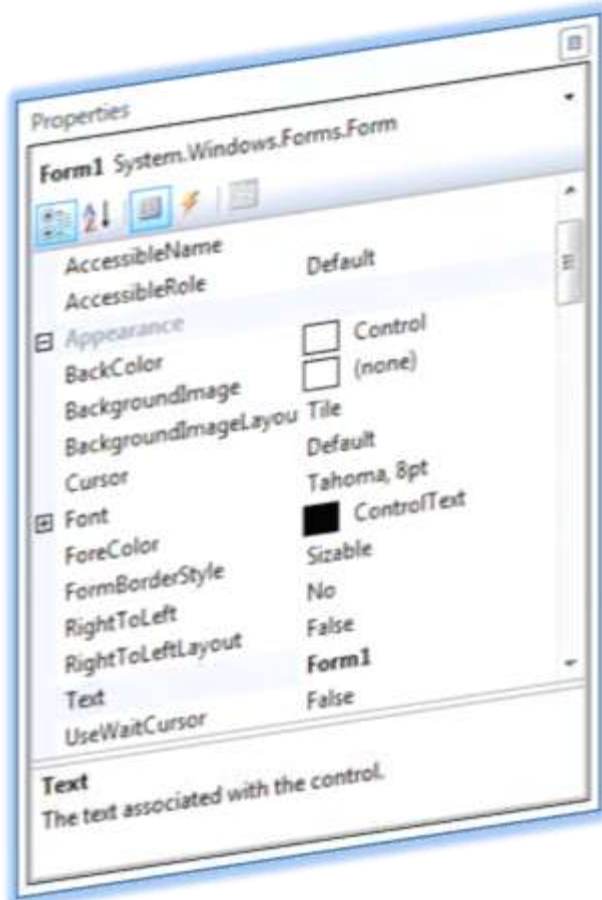
### متصفح المشروع Solution Explorer

وهي المنطقة التي تظهر فيها كل محتويات المشروع

## 5 صندوق الأدوات Toolbox

5

وهو المنطقة التي تظهر بها الأدوات التي يتم التعامل بها وإضافتها إلى الفورم ويوجد بعض الأدوات المشابهة لبرامج الورد والإكسل وقد قامت مايكروسوفت بإخفاء الأدوات النادرة الاستعمال ولكنك يمكنك إظهار الأداة التي تحتاجها من View ثم Toolbars واختيار الأداة التي تريد إظهارها وسيتم شرح هذه الأدوات من خلال التمارين



## 6 نافذة الخصائص Properties

6

تستخدم نافذة الخصائص لتغيير الخصائص للنموذج نفسه ولجميع الكائنات الموجودة عليه من الأزرار والمكونات الأخرى كما تستطيع تغيير هذه الإعدادات من نافذة الأدوات أو من خلال تغيير الكود في ( خانة الكود ) فيمكنك تغيير حجم النموذج أو طريقة إظهاره أو اختفاؤه أو لون الخط أو غيره من الخصائص وسنتعرف عليها كاملا من خلال مشروعنا

## ملحوظة

يتم إرفاق كل مشروع من المشاريع المنفذة في هذا العمل للتوضيح

قبل التعامل مع البرنامج يجب معرفة التالي

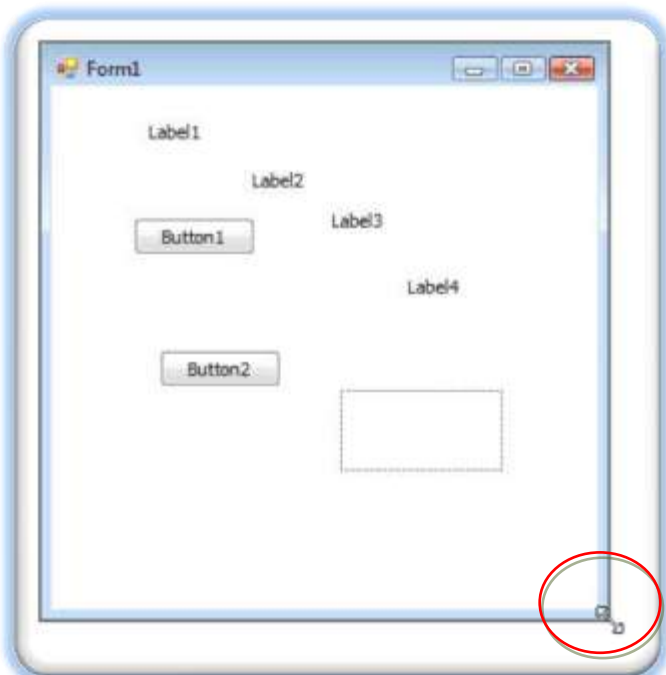
الخطوات	العملية
< Start > All Programs > Microsoft Visual Studio 2008 ثم اختار Microsoft Visual Studio 2008	لفتح الفيچوال بيسك 2008
شريط أدوات الفيچوال بيسك قم باختيار File > open project	فتح مشروع موجود مسبقا من داخل البرنامج
من قائمة Debug اختار start Debugging أو F5	تنفيذ البرنامج ( تشغيله )
View > Toolbox أو Alt+Ctrl+X	إظهار صندوق الأدوات
اختر المادة المراد تغيير خصائصها و حددها على الفورم ثم اذهب إلى نافذة الخصائص واختر الخاصية التي تريد أن تغيرها ثم غيرها	تغيير الخصائص
Project > Add Windows Form	لإضافة نموذج جديد
بواسطة الضغط على Ctrl+Tab ثم انتقل بينها بتكرار الضغط على Tab أو التّنقل بالأسم بين الملفات المفتوحة وأدوات التطوير وتستطيع اختار المادة المعنية بواسطة الماوس مباشرة خلال الضغط على Ctrl+Tab	التنقل بين الملفات المفتوحة
استخدم Alt+F7 للتنقل بينهم أواستخدم Alt+Shift+F7 للتنقل بينهم بطريقة عكسية	التنقل بين أدوات التطوير
تحت قائمة FILE اختار EXIT والحفظ	إغلاق الفيچوال بيسك 2008

وكما تعودنا معا لتسهيل الشرح سوف نقوم بالتطبيق المباشر على التمارين والتعرف أكثر بالبرنامج من خلال تنفيذ هذه التمارين وكيفية التعامل معها وتطبيق ما نريده عليها

عند فتح البرنامج فيجوال بيسك 2008 لأول مرة تظهر لنا الشاشة التالية ومن خلالها يتم التعامل مع مشاريع البرنامج كما يلي



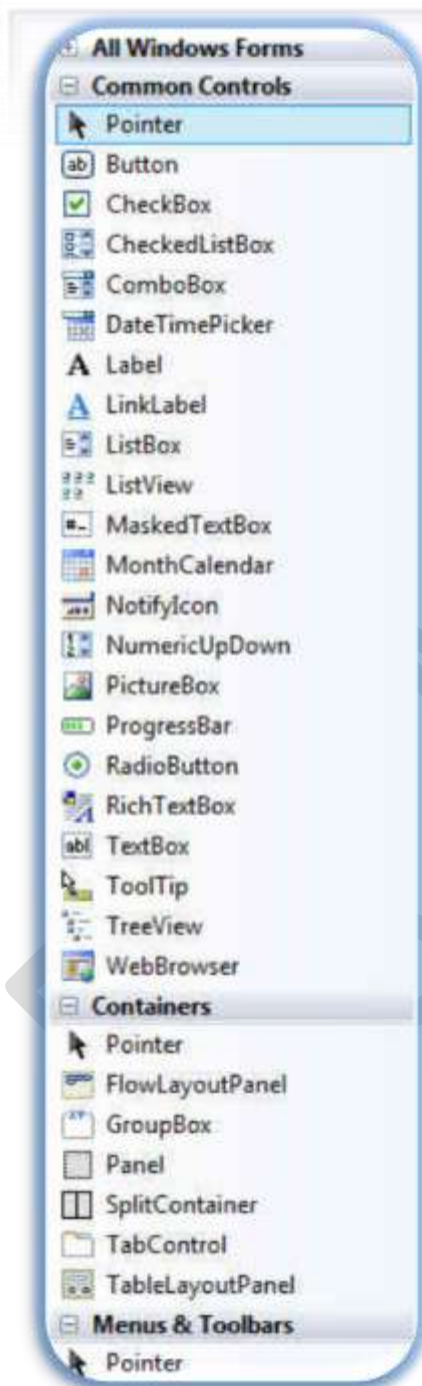
ليظهر لنا أول نافذة للبرنامج بها نموذج **Form** للتعامل معه وتنفيذ المشروع عليه من خلال إضافة ما نريده من خلال صندوق الأدوات وتعديل خصائصه من خلال صندوق الخصائص ليناسب الغرض منه في المشروع



### ❖ النموذج Form

دعنا نتفق من البداية أن سطح العمل في البرنامج هو النموذج كما يمكن إن يحتوى النموذج على أكثر من أداة ونظرا لصغر مساحته الافتراضية من البرنامج يمكننا تكبير المساحة لاحتواء كل الأدوات التي نريدها فيه بالذهاب إلى أسفل يمين الفورم وبالضغط على الماوس يتم التحكم في مساحة سطحه وبذلك نستطيع إن نخرج عليه كل ما نحتاجه من الأدوات

### ❖ صندوق الأدوات Toolbox



وهو مدخلنا إلى التعامل مع الفورم فمن خلاله ومن خلاله فقط يتم إدراج الأدوات إلى سطح الفورم ومن ثم يتم التعديل عليها حسب فكر المبرمج ليصل في النهاية إلى الشكل والمضمون الذي نريده وتحقيق الهدف من البرنامج وسوف يتم التعرف على مكوناته من خلال التمارين التي سوف نقوم بعملها معا بإذن الله

### ❖ الهدف

يجب إن يكون لك هدف و تصور مسبق لشكل البرنامج الذي ستقوم على تنفيذه فلا بد من معرفة مسبقة وتصور لكل ما يحتويه الفورم من أدوات تخدم مشروعك ولا تسمح ح إلى نفسك وتترك تحديد الهدف أو المكونات وأنت تعمل داخل البرنامج فسوف تجد نفسك تقوم بإضافة مكونات أزلتها بدون هدف مما يضيع وقتك ويشتت تفكيرك في النهاية فأفضل البرامج هو أبسطهم طالما يؤدي الغرض منه (ويمكن إضافة أى أداة بالنقر عليها ثم النقر على الفورم لتندرج تلقائيا إلى الفورم ليتم تنسيقها والتعامل معها )

### ❖ الكائنات

وهي كل ما نضيفه إلى الفورم للعمل عليها ويمكننا التعامل مع كل إضافة بعد إدراجها إلى الفورم على حدة وذلك بالتحكم في خصائصها من خلال نافذة الخصائص التي تتغير تلقائيا بمجرد اختيار الإضافة أو الأداة التي يتم التعامل معها على الفورم

وألان لنبدأ مع التطبيق مع مراعاة كل الملاحظات والشروط السابقة عند التنفيذ

## التمرين الأول

المراد عمل تمرين بة زر واحد فقط وعند الضغط عليه تظهر كلمة ( بسم الله الرحمن الرحيم )

### الخطوات

1. إضافة **label** وهي أداة للكتابة بداخلها مثل مربع الكتابة في الأوفس بالضبط
2. إضافة **button** وهي أداة تضيف زر إلى الفورم وبالضغط على هذا الزر يقوم بتنفيذ أمر معين من خلال كتابة الكود بة كما سنتعلم لاحقا

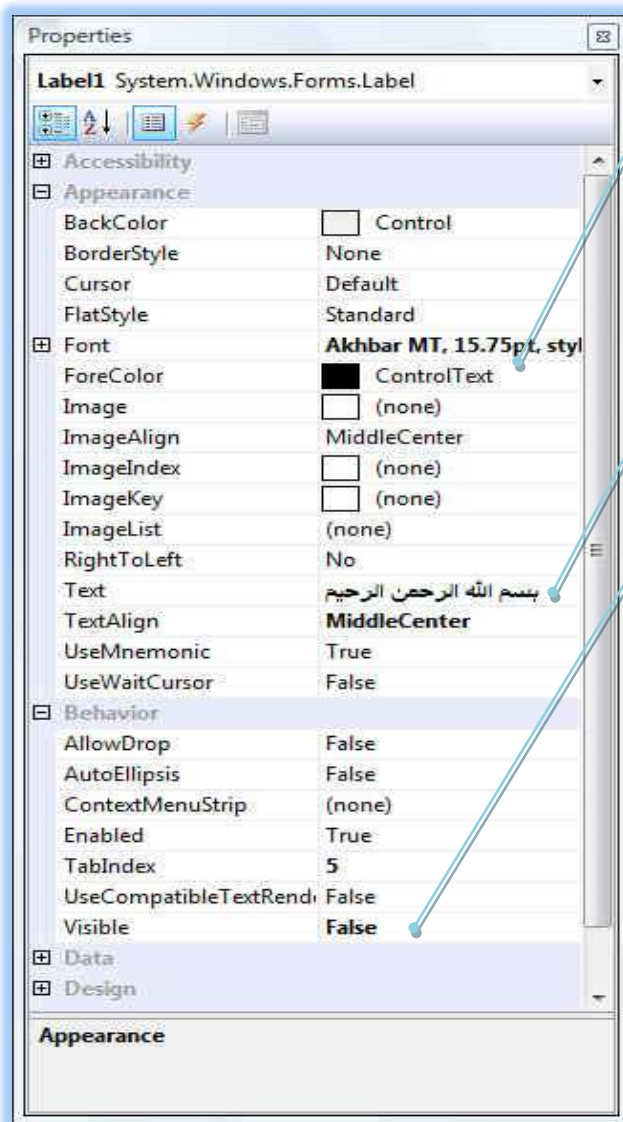
طبعا نحن الآن قد أضفنا ال **label** و ال **button** ولكن بإعدادات البرنامج والآن سوف نقوم بإعداد خصائص كل منهما حسب رغبتنا ولكن قبل إن نقوم بالإعدادات نسال أنفسنا هذا السؤال

أنت وضعت الأداة **label** و **Button** في الفورم هنا ليه ؟

لكي تظهر بها كلمة ( بسم الله الرحمن الرحيم ) عند الضغط على الزر

عند حصولنا على هذه الإجابات نذهب إلى خصائص كل من ال **label** و **button** ونغيرهم كما يلي

### أولا ال Label



هذا الجزء خاص بإعدادات الخط و الألوان لهذا ال Label و محتوياته

يتم تغيير الكلمة الموجودة في ال **label** من هنا حسب الكلمة المراد إظهارها بة

من هنا يتم إخفاء ال **label** بكل ما يحتويه عند بداية التشغيل



شكل الفورم بعد إعداد خصائص Label



### ثانياً الـ Button

يتم تغيير اسم زر من **Button1** إلى (أبدا) وهو الزر الذي تظهر كلمة (بسم الله الرحمن الرحيم) في الـ **Label** عند النقر عليه

ملحوظة لن نكرر الخصائص التي تم التعامل معها فالآن نحن نعلم أنه لتغيير الاسم الظاهر لنا يتم التلاعب بالخاصية **Text**

وألآن نريد أنه يتم إظهار المحتوى في الـ **Label1** في حالة النقر على زر أبدا فلأبد من برمجة هذا الزر الذي نريد من خلاله إظهار المحتوى **label1** والذي هو من الأساس مخفي كما سبق و ووضعتنا الخصائص لة

ويكون كتابة الكود بالزر بالنقر عليه مرتين لتظهر لنا صفحة كتابة الأكواد ويتم كتابة الكود التالي

وهذه طريقة كتابة الأكواد و متبعة لجميع الكائنات المدرجة في النموذج



ثم نقوم بتشغيل البرنامج وذلك باستخدام **F5** ونرى النتيجة

تحميل التمرين الأول



## التمرين الثاني

المراد عمل تمرين يكون به سؤال وإجابته مع أظهار صورة تنتمي إلى الإجابة

مثلا ( ما هي قبلة المسلمين ؟ ) الإجابة هي ( الكعبة ) ..... ( وتظهر صورة للكعبة مصاحبة للإجابة )



طبعا الإجابة والصورة يكونان مختلفيان عند التشغيل ويظهرا عند النقر على زر الإجابة ويتم تجهيزهما لذلك من صندوق الخصائص

الأدوات التي يتم إدراجها من صندوق الأدوات هي

1. عدد 2 label لكتابة السؤال والإجابة بكل منهما
2. عدد 1 button زر لإظهار الإجابة عند الضغط عليه
3. عدد 1 PictureBox لإدراج الصورة به

طبعا بعد كتابة السؤال في label1 وكتابة الإجابة في label2 نقوم بالذهاب إلى خصائص كل منهم وتغيير إعداداتهم كما بالشكل مع ملاحظة أن نخفي الإجابة label2 وليس السؤال label1 عند بداية التشغيل تغيير اللون ونوع الخط أيضا على حسب إرادتك أو كما هو مبين بالشكل

ثم الذهاب إلى صندوق الصورة PictureBox واختيار الصورة المطلوب إظهارها وتغيير خصائصها بحيث تظهر كاملا في مربع الصورة مع ملاحظة أنه يمكن تغيير حجم الصورة حسب التصميم المرغوب به وموضعها في الفورم كما نجعل الصورة أيضا مخفية عند بداية التشغيل كما بالشكل

الباقى إمامنا هو button1 وهو زر الإجابة والمراد برمجته بحيث أنه عند النقر عليه يظهر كل من label2 و PictureBox المختلفان أساسا كل منهم حسب إعدادهما في الفورم فبعد تغيير اسم الزر إلى الإجابة يتم النقر عليه مرتين وكتابة الكود التالي

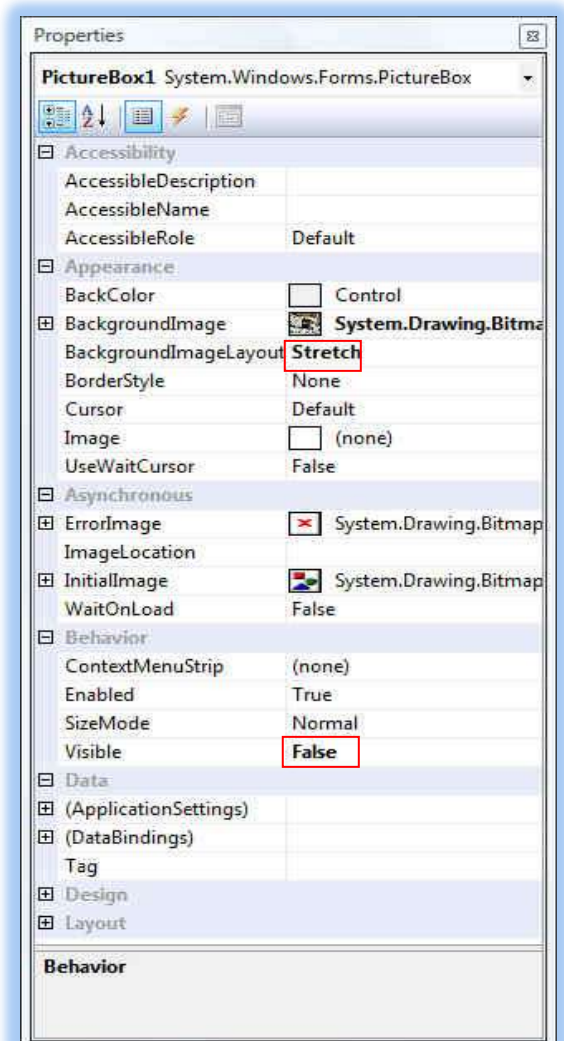
Label2.Visible = (True)

PictureBox1.Visible = (True)

وأنت هنا تطلب منه

جعل خاصية الإظهار لـ label2 محققة أي يكون ظاهر

جعل خاصية الإظهار لـ PictureBox1 محققة أي يكون ظاهر



```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Label2.Visible = (True)
    PictureBox1.Visible = (True)
End Sub
End Class
```

تحميل التمرين الثاني

### التمرين الثالث

جميعنا ندخل المنتديات وهذا الشكل ليس بالغريب علينا فالمطلوب هو عمل نموذج عند إدخال كلمة السر والموافقة على الاتفاقية تظهر لنا رسالة ( كلمة المرور صحيحة ) وفي حالة ترك كلمة المرور فارغة والضغط على زر الدخول تظهر لنا رسالة ( برجاء إدخال كلمة المرور الصحيحة ) مع تحقيق شرط أنه لا بد من كتابة كلمة السر الصحيحة والموافقة أيضا على الاتفاقية باختبار مربع الاتفاقية وغير ذلك لا تظهر أي رسالة

طبعا في مراحل متقدمة يمكن تغيير الرسالة هنا بصندوق الرسائل أو بنموذج آخر مثلا أو صفحة ويب أو أي شيء آخر حسب تصميم المشروع

طبعا لن نتكلم عن الخصائص إلا الجديد منها وتعتبر هذه قاعدة من الآن توفير الوقت والمجهود

العناصر المدرجة من صندوق الأدوات بالنموذج هي

1. عدد 2 label وذلك لكتابة الرسالتين عليهم
2. عدد 1 button وذلك لتنفيذ وتشغيل البرنامج من خلاله
3. عدد 1 textbox وهو لإدخال كلمة المرور
4. عدد 1 checkbox وذلك للموافقة على شروط الاتفاقية من خلاله

الجديد هنا خصائص الـ checkbox صندوق الاختيار وفيها سوف نغير أسماء إلى ( الاتفاقية ) ونغير اتجاهه إلى ناحية اليمين ليظهر المربع على يمين الجملة لأنها مكتوبة باللغة العربية وجعله غير مختار

والباقى إمامنا الآن هو برمجة الزر button وذلك لتنفيذ فكر البرنامج وعمل المرجو منه ليكون الكود كالتالي

```
If (TextBox1.Text = "1612") And (CheckBox1.Checked = True)
Then Label1.Visible = True

If (TextBox1.Text = "")
Then Label2.Visible = True
```

وهو في حالة إن textbox الكتابة الموجودة به هي "1612" وأيضا أن يكون مربع الاختيار في checkbox قيمته محققة أي مختارة إذن تكون خاصية الظهور للـ label1 محققة أي ظاهرة ( الرسالة الأولى )

وفي حالة أن textbox الكتابة الموجودة به هي "فارغة" إذن تكون خاصية الظهور للـ label2 محققة أي ظاهرة ( الرسالة الثانية )

تحميل التمرين الثالث

## التمرين الرابع

المطلوب برنامج يحسب اليوم في السنة مثلا يوم 1 يناير هو معروف أنه يوم 1 في السنة ولكن لو فرضنا انك تريد أن تعلم رقم يوم ميلادك من سنة مولدك ( وليكن مولدك في 16 ديسمبر سنة 1973 ) فما هو رقم يومك من السنة ؟

العناصر المدرجة في النموذج هي

1. عدد 1 **DateTimePicker** من صندوق الأدوات وهي لإدراج التاريخ ميلادك المراد معرفه رقم اليوم فيه من السنة
2. عدد 1 **button** لإظهار النتيجة بالنقر عليه

ملاحظة

عند إدراج **DateTimePicker** إلى النموذج يظهر تاريخ اليوم لتاريخ بنائك المشروع ويمكن تغييره من خواص **DateTimePicker1** من الخاصية **value** من صندوق الخصائص الخاص بها إلى التاريخ المراد إظهاره عند بداية التشغيل

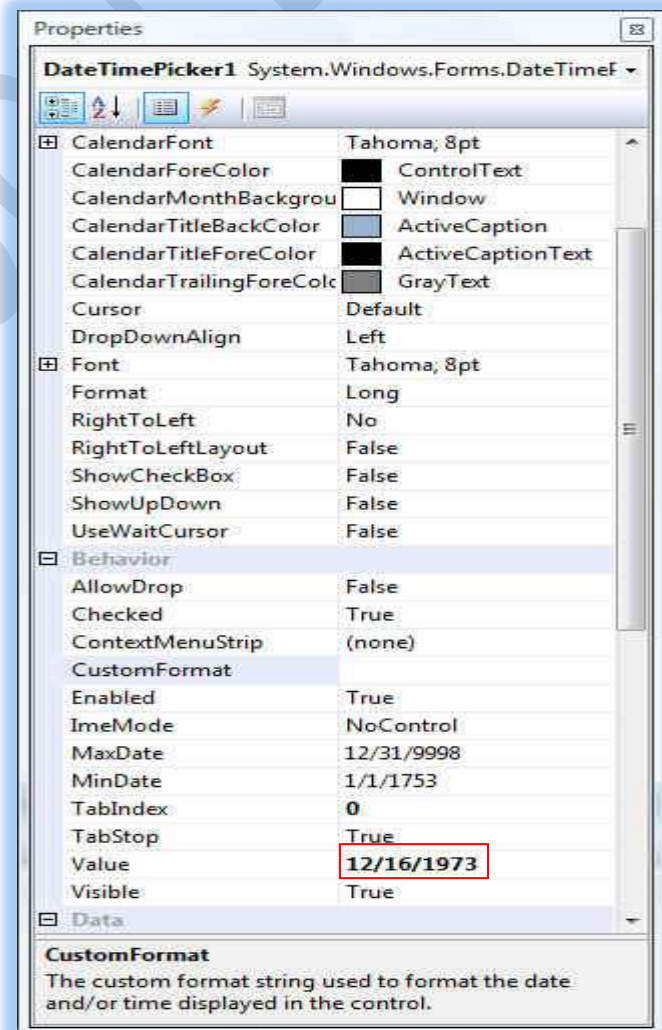
يتم برمجة **button** لتظهر رسالة عند النقر عليه تقوم بحساب رقم اليوم من التاريخ الذي تم إدخاله في **DateTimePicker1** لتكون البرمجة كالتالي

```
MsgBox("&_ أنت مولود في اليوم " & DateTimePicker1.Value.DayOfYear & " من السنة ")
```

وهنا نطلب منة عند النقر على **button** تظهر صندوق رسالة **Msgbox** مكتوب فيه " أنت مولود في يوم " نلاحظ وضع العلامة **&** لإضافة شيء آخر إلى سطر الجملة ثم نجعل الـ **DateTimePicker** يحسب قيمة اليوم المدرج فيه من أيام السنة ثم نزيد كتابة " من السنة " حتى تكون تمييزا لنتائج العملية الحسابية التي سبقتها ليكون الشكل النهائي للرسالة هو

أنت مولود في اليوم 350 من السنة

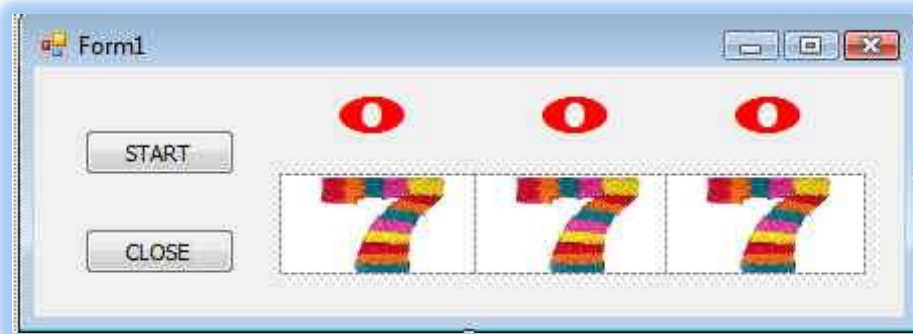
OK



تحميل التمرين الرابع

## التمرين الخامس

فكرة عمل هذا التمرين هو لعبة الأرقام شبيهة بالروليت اللعبة الشهيرة وهي عبارة عن نموذج هذا النموذج بة زر لتوليد أرقام في كل مرة يتم النقر عليه وتكون الأرقام على ثلاث خانات والمفروض إن توليد الأرقام يكون عشوائي وان حدث أن يظهر الرقم 7 في مكان توليد الأرقام يظهر صورته والرابح هو الذي يظهر الرقم 7 في الثلاث خانات فيكون الرقم **777** هو الرقم الرابح



هذا النموذج يحتوي على

1. عدد 3 LABEL لإدراج الأرقام بها
2. عدد 2 BUTTON أحدهما للإغلاق والأخرى لتوليد الأرقام
3. عدد 3 PICTUREBOX لوضع الصور بها



## لماذا وضعت 3 صور؟

لإظهار كل صورة تحت الرقم الخاص بها في حالة ظهور الثلاثة تظهر ثلاث صور حاملة الرقم 777 يمكنك الاستغناء عنها بصورة واحدة فقط



عند الوقوف على صندوق الصورة لإدراج الصورة وبالضغط على العلامة الذي تعلقه تظهر لنا النافذة التالية ومنها يمكن إدراج الصورة والتحكم بوجودها داخل الإطار

يتم برمجة الزر المسمى **START** كما يلي ونلاحظ أن شرحي هنا مقسم إلى ثلاث أجزاء

1 جزء خاص بالصور وفيه يتم إخفاء جميع الصور الموجودة عند النقر على الزر **start**

```
PictureBox1.Visible = False
PictureBox2.Visible = False
PictureBox3.Visible = False
```

1

```
Label1.Text = CStr(Int(Rnd() * 10))
Label2.Text = CStr(Int(Rnd() * 10))
Label3.Text = CStr(Int(Rnd() * 10))
```

2

```
If (Label1.Text = "7") Then PictureBox1.Visible = True
If (Label2.Text = "7") Then PictureBox2.Visible = True
If (Label3.Text = "7") Then PictureBox3.Visible = True
```

3

2 جزء خاص بالأرقام

- وهنا نريده أن يظهر في الـ LABEL أرقام مسلسلة ولذلك قمنا بإدراج الكود **CStr** من () إلى 10
- وأدراج الكود **Int** مفادها إن الأرقام جميعها أرقام صحيحة
- وأدراج الكود **Rnd** مفادها أن إظهار هذه الأرقام يكون عشوائي غير مرتب

3 الجزء الخاص بتحقيق الشرط وإظهار الصور تباعا له وهو أنه في حالة أن يظهر بداخل الـ label رقم 7 تظهر الصورة التابعة له مباشرة

تحميل التمرين الخامس

## مصطلحات فيجوال بيسك 2008

مما سبق نستطيع أن نحدد بعض المصطلحات المتبعة عند البرمجة في بيئة فيجوال بيسك 2008 ومن هذه المصطلحات

### الجملة البرمجية (Program Statements)

الجملة البرمجية هي عبارة عن الجملة المكتوبة في السطور البرمجية (خانة الكود) وتقوم هذه الجملة بعمل ما خلال مرحلة تنفيذ البرنامج لأن **COMPILER** (المترجم إلى لغة الآلة) يقوم بقراءة هذه الجملة وتنفيذها ويختلف طول هذه الجملة بحسب الحاجة فبعضها قد يكون طويلاً والبعض الآخر قد يحتوي على كلمة واحدة لكن جميعها يجب أن تتبع الطرق البرمجية التي يتقبلها المترجم أو **COMPILER** وفي فيجوال بيسك 2008 الجملة البرمجية قد تحتوي على كلمات مثل خصائص، أسماء كائنات، متغيرات، أرقام، رموز، وقيم

`If (TextBox1.Text = "1612") And (CheckBox1.Checked = True) Then Label1.Visible = True`

### الكلمات المحجوزة (Keywords)

هي كلمات محجوزة في بيئة التطوير هذه الكلمات تتعامل مع **COMPILER** بالطريقة التي قد حددت سلفاً من قبل مطوري لغة البرمجة مثل الكلمة **END** وتستخدم لإغلاق البرنامج أو التطبيق وعليه فلا يمكنك أن تقوم بتعريف متغير بنفس الكلمة والكلمات المحجوزة تعتبر جزء من بنية الجملة البرمجية التابعة للفيجوال بيسك معظم الكلمات المحجوزة تظهر باللون الأزرق في محرر الكود

End Sub

### المتغيرات (Variables)

المتغيرات هي عبارة عن حافظات للبيانات تحفظ البيانات بشكل مؤقت ويتم تعريف المتغير بإضافة كلمة **DIM** قبل المتغير وتقوم هذه المتغيرات بحفظ البيانات بشكل مؤقت وعادة ما تكون هذه البيانات عبارة عن أسماء ملفات، أرقام، تواريخ، صور كما سنوضح لاحقاً

### الأدوات (Controls)

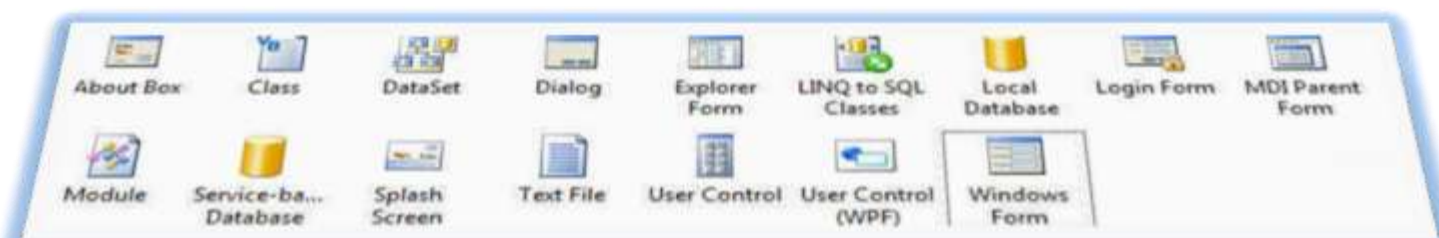
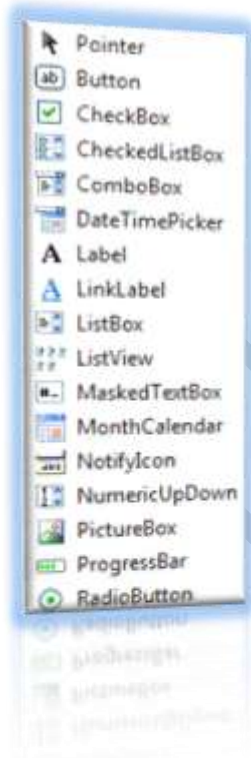
الأدوات هي عبارة عن تلك الأدوات التي تقوم باستخدامها للإضافة كائن إلى الفورم مثل الأزرار، صناديق النص الليبلات وصناديق الصور وغير من الأدوات التي يمكنك إضافتها إلى الفورم

### الكائنات (Objects)

الكائنات هي عبارة عن العناصر التي تقوم بصنعها بواسطة برنامج فيجوال بيسك باستخدام أحد الأدوات الموجودة في صندوق الأدوات **TOOLBOX** مثل الفورم وصندوق الحوار

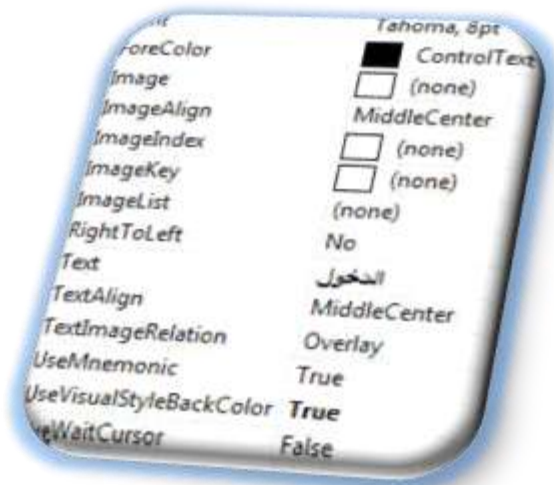
### الكلاس (Class)

الكلاس أو القالب عبارة عن قالب لكائن أو أكثر والذي يحدد فيه ماذا يفعل هذا الكائن وفي فيجوال بيسك 2008 تستطيع استخدام أي من القوالب الموجودة ضمن بيئة التطوير



## مجالات الأسماء (Namespaces)

عبارة عن قوالب **Classes** مرتبة تحت اسم معين مثل **System. Diagnostics** أو **System. Windows** ولكي نستطيع إن نصل إلى هذه المجالات **Classes** لابد أن تكتب **Imports** في أعلى الفورم متبوعاً باسم مجال الأسماء المحدد



## الخصائص (Property)

الخاصية هي عبارة عن قيمة معينة محمولة بواسطة كائن معين. فمثلاً الزر **Button** لديه خاصية النص **TEXT** وهذه الخاصية تجعلك قادر على تغيير اسم الزر وكذلك الخاصية **IMAGE** والتي تحدد مسار الصورة الموجودة على الزر ففي فيجوال بيسك تستطيع من تغيير الخصائص وقت التصميم بسهولة من نافذة الخصائص

## الأحداث (Event Procedure)

الأحداث هي عبارة عن كود معين يتم تنفيذه عندما تتم معالجة كائن ما في البرنامج فمثلاً الزر يحدد ماذا يقوم البرنامج بتنفيذه في حالة النقر عليه

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Label1.Visible = True
```

## الطرق (Methods)

هي عبارة عن أوامر برمجية معينة لتقوم ببعض الأعمال أو تنفذ خدمات معينة لكائن معين في داخل البرنامج

```
PictureBox1.Visible = False
```

يوجد بعض التقارب بين الخصائص **Properties** والطرق **Methods** ويمكن ان نفرق بينهما بواسطة مكانهما في الكود البرمجي



## أدوات لجلب المدخلان من المستخدم

تقدم بيئة فيجوال بيسك العديد من الأدوات لجلب المدخلان من مستخدم البرنامج مثل صناديق النص **Textbox** وتستخدم لإدراج بعض النصوص بها وهناك القوائم **Menus** التي تقوم باستقبال المدخلان بواسطة النقر عليها بالماوس أو باختيارها بواسطة الكيبورد وهناك العديد من الأدوات الأخرى مثل **RadioButton** تستخدم لمعرفة ما إذا كان الشخص ذكراً أو أنثى و الاختيار بين حالتين و الأداة **CheckBox** تستخدم في حالة ما أردنا من المستخدم اختيار أكثر من خيار وهناك الأداة **Listbox** وهي تتيح لك اختيار أكثر من خيار بطريقة التظليل وسنوضح ذلك بإذن الله في التمارين القادمة

## التمرين السادس

المطلوب هو عمل تمرين لبرنامج لاستعراض مكونات الكمبيوتر على أن تكون وجهته كالشكل 1



فكرة عمل البرنامج هو عند الضغط أو اختيار اسم المكون تظهر له صورة توضيحية في نفس النموذج على أن تظهر في البداية المكونات الظاهرة للاستخدام وعند النقر أو اختيار الـ case تظهر محتوياتها وباختيار كل محتوى تظهر صور توضيحية له كما بالشكل 2

الكائنات المدرجة في هذا التمرين هي بسيطة جدا وهي

1. عدد 12 PictureBox لوضع الصور بهم
2. عدد 11 CheckBox ليتم اختيار المكونات منها

الخطوات

اولا بالنسبة للـ PictureBox يتم وضع الصور بهم وتجهيز الخصائص لها بحيث تكون مختفية منذ البداية واختيار مكانها في النموذج ثانيا يكون تجهيز الـ CheckBox وكتابة اسم المكون عليه وتجهيز خصائص اللون والخط له لتناسب مع ذوقك ويتم كتابة الكود له بحيث عند الضغط عليه تظهر صورة المكون المراد اظهاره مع ملاحظة ان كل CheckBox توجد له PictureBox خاصة به

نلاحظ انه عند الضغط على CheckBox المسمى case تظهر لنا CheckBox اخرى وهي للمكونات الداخلية للـ case وباختيارهم تظهر صور توضيحية لكل منهم على حدة

```
PictureBox1.Visible = False
PictureBox2.Visible = True
PictureBox3.Visible = False
PictureBox4.Visible = False
PictureBox5.Visible = False
PictureBox6.Visible = False
PictureBox7.Visible = False
PictureBox8.Visible = False
PictureBox9.Visible = False
PictureBox10.Visible = False
PictureBox11.Visible = False
PictureBox12.Visible = False
```

هذا مثال لكتابة كود زر الاختيار CheckBox وهنا نجد ان جميع الـ PictureBox مختلفة الا الـ PictureBox2 وهو الخاص باظهار الصورة للـ CheckBox1 وهكذا يتم برمجة باقي الـ CheckBox

نلاحظ ايضا ان الـ CheckBox الخاصة بالحالة case لا تظهر في بداية النموذج ولكن فقط في حالة اختيار الـ case لعرض مكوناتها

وهذا كود يترجمه الفورم عند بدايته حتى يقوم باخفاء هذه الـ CheckBox واظهارها فقط عند الحاجة الى اظهارها وهو عند الضغط على CheckBox المسمى بالـ case ويتك برمجة الاظهار في الـ CheckBox الـ case

```
CheckBox5.Visible = False
CheckBox6.Visible = False
CheckBox7.Visible = False
CheckBox8.Visible = False
CheckBox9.Visible = False
CheckBox10.Visible = False
CheckBox11.Visible = False
```

تعلمنا مما سبق كيفية استخدام الأداة **checkbox** والأداة **PictureBox** وكيفية التعامل معها واستخدامها داخل النموذج (**form**) ولكن لاحظنا أنه في المثال السابق قد استخدمنا عدد 12 **PictureBox** و أعطينا كود لكل منهم فبالتالي كان عندنا أكثر من 12 كود خاصة بالصور فقط لأنها كانت كثيرة في النموذج السابق أليس هذا مضيعة للوقت وأيضا قد يرفع لك نسبة الخطأ عند البرمجة من كثرة الـ **PictureBox** وما سنتعلمه الآن هو طريقة احترافية بسيطة لتعامل مع **PictureBox** في حالة ان تكون كثيرة وسنتعلمها على هيئة تمرين كالتالي

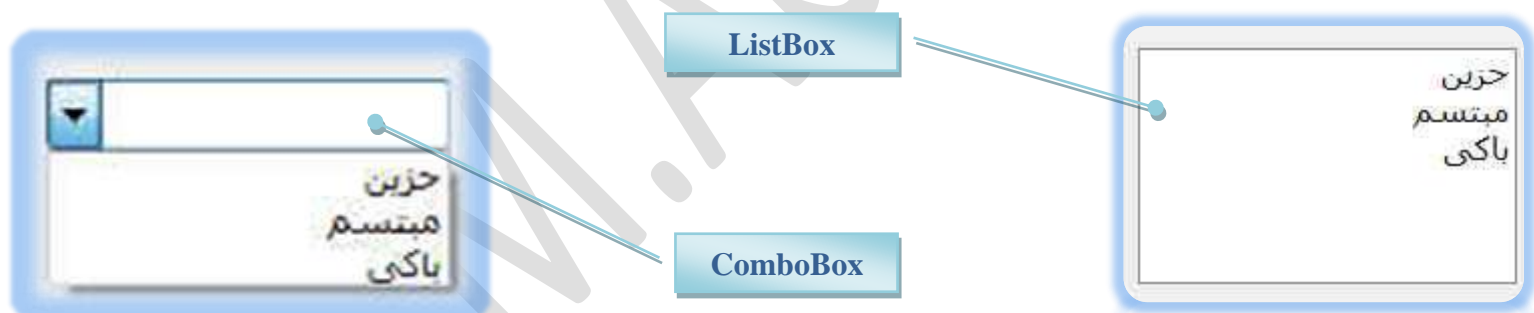
### تحميل التمرين السادس

### التمرين السابع

المطلوب عمل تمرين بة نموذج ويحتوى هذا النموذج على بعض الاختيارات وكل اختيار منهم يؤدي إلى إظهار صورة معينة داخل **PictureBox**

الصور المدرجة في هذا النموذج لابد ان تكون اسمها سهل وبدون مسافات وقد استخدمت في هذا النموذج 3 صور للتوضيح وقد قمت بتسميتهم (**sadface,happyface,cryface**) مما يسهل على كتابة الكود لهم

من أسهل الأدوات التي يتم بها إدراج اختيارات هي **ListBox** و **ComboBox** والتعامل معهم متشابه تماما مع اختلاف بسيط في طريقة عرض كل منهم



لاحظ هذا الفرق جيدا ويمكنك استخدام كل منهم حسب رغبتك وتصميمك

### الكائنات المستخدمة

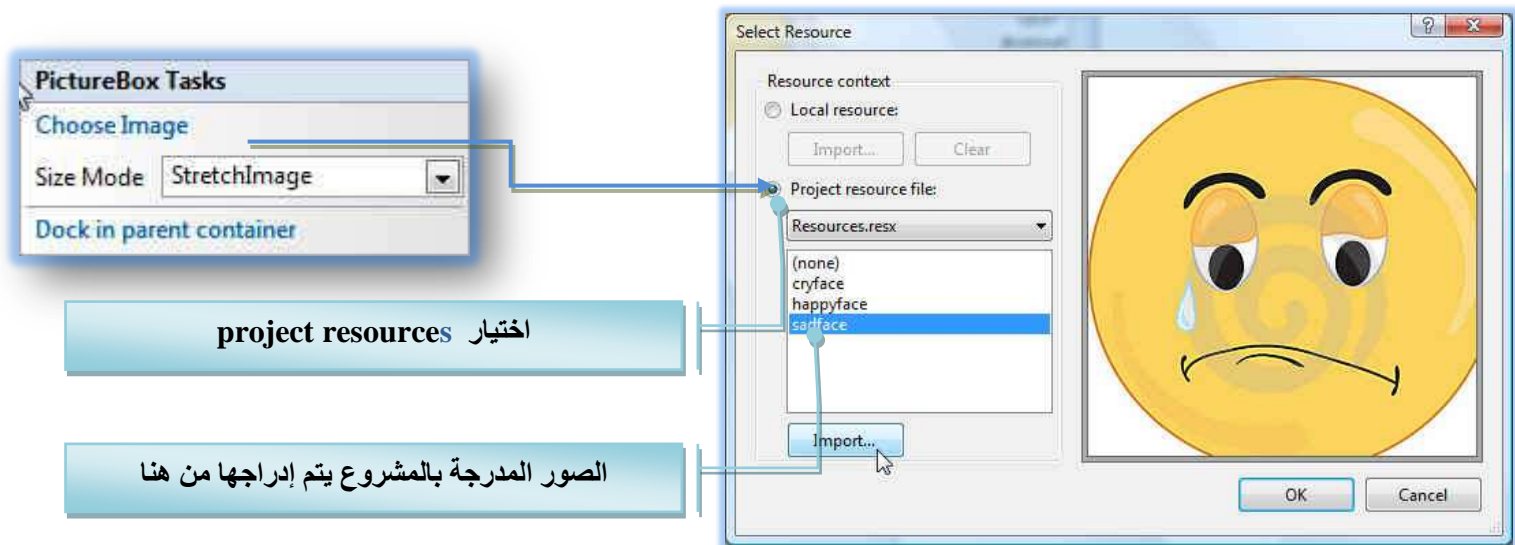
1. عدد 1 **label** لكتابة الجملة الظاهرة بالصور في
2. عدد 1 **ListBox** وهو لإدراج الاختيارات عليه على إن تكون الاختيارات كالتالي ( حزين – مبتسم – باكي )
3. عدد 1 **PictureBox** وذلك لإدراج الثلاث صور التي تعرض أي منها في حالة اختيار أي اختيار من الاختيارات السابقة



أرجو الاهتمام بالشرح التالي وهي طرق كتابة الكود لكل منهم لأننا لن نعود لها مرة أخرى فأرجو الاهتمام والتركيز مع الإلمام بالملاحظات السابقة

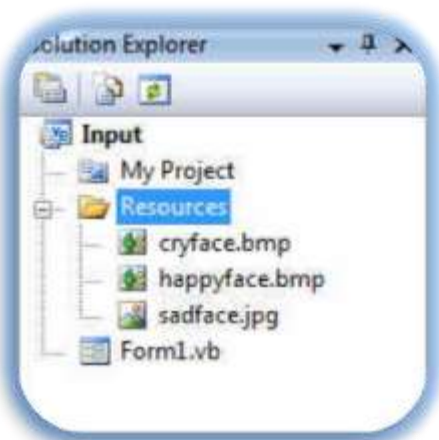


أولا وضع الصور في PictureBox ولن تختلف كثيرا مع الطريقة المتبعة سابقا ولكن سيتم التعامل معها كالتالي



اختيار project resources

الصور المدرجة بالمشروع يتم إدراجها من هنا



بعد الانتهاء من الخطوات السابقة يتم إدراج الصور في الـ solution explorer كما هي موضحة بالشكل

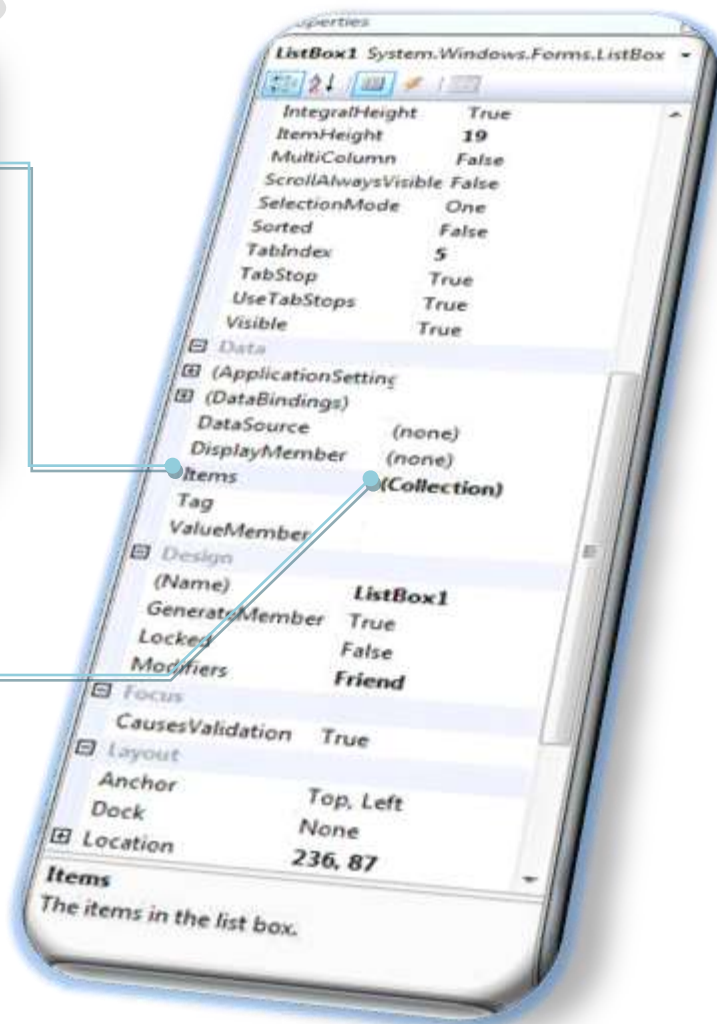
نلاحظ ان امتداد الصورة مهم جدا في حجم البرنامج فكلما كانت مساحات الصور اصغر كان مساحة البرنامج اصغر وظهورها عند الحاجة أسرع نوعا ما

بعد تغيير اسم الـ picturebox إلى LstIndex من خاصية NAME بصندوق الخصائص

ثانيا كتابة الاختيارات ( المدخلات ) في الـ ListBox وهناك أسلوبين متبعين عند كتابة الكود وهما أولا من خلال صندوق الخصائص ويكون كالتالي



من صندوق الخصائص يتم اختيار الخاصية items لتظهر لنا النافذة التالية والتي يتم فيها كل الاختيارات التي تريد إظهارها مهما كان عددها بشرط أن تكون كل اختيار في سطر خاصة



نلاحظ أن التعامل مع **ComboBox** شبيه تماما كمثل التعامل مع **ListBox** ولكن الاختلاف في شكل العرض لكل منهم كما هناك خاصية إضافية للـ **ComboBox** فيمكنك مثلا إظهار قيمة افتراضية في البداية تعتبر كعنوان للاختيارات الموجودة



ويمكن استخدام هذه الخاصية من صندوق الخصائص التابعة للـ **ComboBox** وذلك بكتابتها أمام الخاصية **TEXT** الخاصة بـ

الطريقة الثانية وهي طريقة احترافية ويجب عند استعمالها الإلمام بكود الفيچوال بيسك 2008 ويتم كتابتها عند النقر المزدوج على النموذج ( **FORM** ) لتظهر لنا صفحة كتابة الأكواد لنكتب الكود التالي في شرط **Form1\_Load** ليتم ظهوره مباشرة وتلقائيا في **ListBox** عند تشغيل البرنامج

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    ListBox1.Items.AddRange(New Object() {" SADFACE", " HAPPYFACE", " CRYFACE"})
    ' سيتم كتابة الأكواد كما بالشكل ليتم تلقائيا إظهارها عند تشغيل الفورم في الـ ListBox
End Sub
```

`ListBox1.Items.AddRange(New Object() {" SADFACE", " HAPPYFACE", " CRYFACE"})`

الكود خاص بإضافة عناصر إلى  
**ListBox1**

إضافة جديدة إلى الـ **ListBox1**

أسماء الصور المدرجة  
بالمشروع

`ComboBox1.Items.AddRange(New Object() {" SADFACE", " HAPPYFACE", " CRYFACE"})`

نلاحظ التشابه بين الكودين في السطرين السابقين مع اختلاف بسيط في تغيير اسم الاداة المستخدمة

ملاحظة

وجود بعض السطور بالألوان عند كتابتنا في صفحة الأكواد وهي للتمييز بين الأكواد فمثلا اللون

الأصفر وهو خاص بإدراج أسماء الكائنات التي يتم إدراجها من صندوق الأدوات

الأزرق وهو يظهر للأسماء المحجوزة للبرنامج أي يظهر للأوامر التي يتم استخدامها في الفيچوال بيسك 2008

الأحمر وهو تظهر به الجمل التي تظهر في المشروع ونلاحظ أنها دائما موجودة بين العلامتين " "

الأخضر وهو يظهر عند كتابة جمل شرح للتذكير فقط ولا دخل لها في تنفيذ الأكواد وتكون دائما بداية السطر لها علامة ' '

والآن نأتي إلى مرحلة كتابة الكود في الـ **ListBox1** والغرض منها هو إظهار كل صورة مما سبق وتم إدراجهم في المشروع وارتباطها باسم الاختيار المدرج سابقا في الـ **ListBox1** ويتم ذلك بالنقر على الـ **ListBox1** مرتين لتظهر لنا صفحة الأكواد ويتم كتابة الكود كالتالي

```
Private Sub ListBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ListBox1.SelectedIndexChanged
    Select Case ListBox1.SelectedIndex

        Case 0
            LstIndex.Image = Global.WindowsApplication7.My.Resources.sadface
            LstIndex.Visible = True

        Case 1
            LstIndex.Image = Global.WindowsApplication7.My.Resources.happyface
            LstIndex.Visible = True

        Case 2
            LstIndex.Image = Global.WindowsApplication7.My.Resources.cryface
            LstIndex.Visible = True

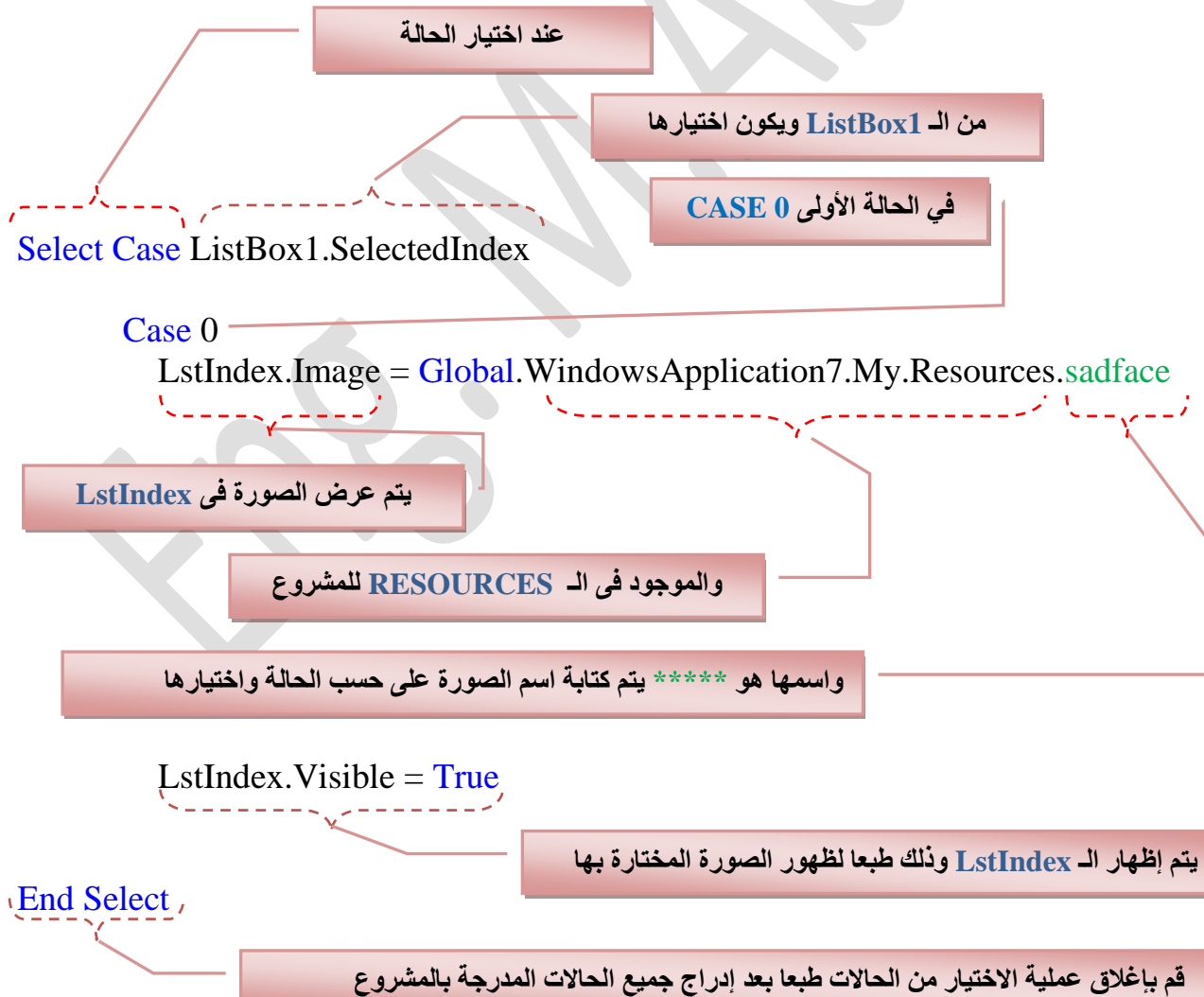
    End Select
End Sub
```



يتم ملاحظة ان حالات الاختيار مرقمة وتبدأ بالترقيم 0

- الحالة حزين CASE 0
- الحالة مبتسم CASE 1
- الحالة باكي CASE 2

ويتم كتابة كود الاختيار فيما بينهم كالتالي سوف يتم الشرح على حالة واحدة لتشابههم في الكود



يتم التكرار للحالتين التاليين مع ملاحظة تغيير اسم الصورة لكل حالة

تحميل التمرين السابع

والآن بعد تنفيذنا لهذا المشاريع السابقة أصبحت لنا خبرة لا بأس بها في التعامل مع خصائص الكائنات ومنها

- ☒ PictureBox
- ☒ CheckBox
- ☒ ListBox
- ☒ ComboBox
- ☒ Form
- ☒ Button
- ☒ Label

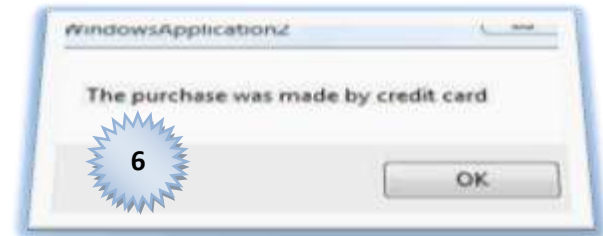
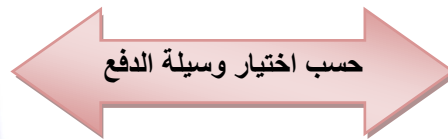
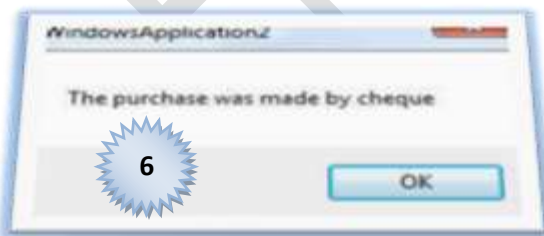
ولذلك سوف نقوم بعمل تمرين كبير يحتوى على معظم هذه الأدوات وتنسيقها معا والتعامل مع خصائصها لإظهارها بالشكل المطلوب ولأني هنا بصدد شرح وليس ابتكار أو التعامل مع محترفين سأقوم بالشرح بالطريقة التقليدية وهناك بدائل قد ذكرتها من قبل للبرمجة السريعة وتوفير سطور عديدة من اكواد البرمجة ولكني سأتبع الطريقة البدائية و سوف أكثر من استخدام الأكواد حتى يتثنى لنا التدريب عليها والتعامل مع الفيچوال بيسك 2008 بأقل مجهود وابطس المعلومات الأكواد



كلنا شاهدنا بعض المواقع التي تتيح لك الشراء عبر الانترنت ونحن هنا بصدد عمل نموذج لواجهة شراء كمبيوتر وملحقاته على أن يتم أدرج

1. أسماء شركات للكمبيوتر ويتم الاختيار نوع الجهاز من بينهم مع وجود صورة تحمل لوجو الشركة تظهر عند الاختيار
2. موديلات أجهزة مختلفة لكل شركة تظهر صورته عند اختياره هنا اخترت ثلاث موديلات
3. اختيار طريقة الدفع سواء شيك أو من خلال كرت كارت
4. أدرج بعض صور لبعض الإضافات التي يمكن شرائها كملحقات للجهاز الكمبيوتر الذي قمت باختياره سابقا
5. وطبعاً يوجد زر لتنفيذ عملية الشراء بناء على الاختيارات السابقة

6. إظهار رسالة عند الشراء تبلغك بان عملية الشراء تمت بنجاح ويوضح بها وسيلة الدفع المختارة من خلالك



مع ملاحظة أنه لم تظهر رسالة الشراء أياً في حالة اختيار الجهاز وطريقة الدفع

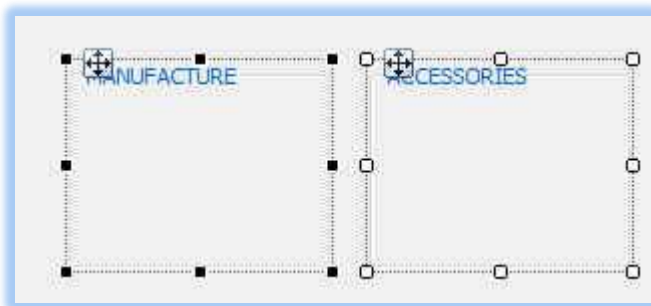
## التمرين الثامن

ملاحظة يمكنك الاختيار بين الطريقة المتبعة في التمرين السابع لإدراج الصور في الـ **PICTUREBOX** أو يمكنك استخدام الطريقة الأقل احترافا كما تعلمنا في بداياتنا وهذه هي الطريقة التي اتبعها هنا لأنني بصدد شرح مبسط للجميع نعم هي طويلة ومملة ولكن الغرض منها إكسابك المهارات على التعامل مع الأكواد وكرر أني لم اكرر خطوات بديهية أو معروفة من البداية توفيراً للوقت والمجهود ودعوة مني لك لليقظة وفهم الأمور بطبيعتها

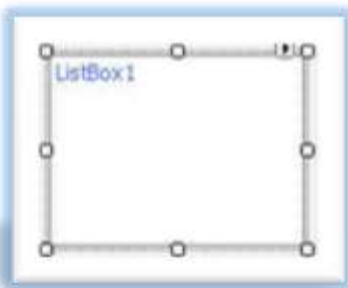
الكائنات المدرجة وهي

1. عدد 1 **LABEL** وهو لإدراج العنوان علياً ويتم تنسيقه كما يلي

BUY YOUR COMPUYER



2. عدد 2 **GroupBox** وهما لنقسم بها **Checkbox** إلى مجموعتين هما مجموعة الـ **Manufacture** ومجموعة الـ **Accessories**



3. عدد 3 **Checkbox** وهما يتم من خلالهم التعامل مع **PictureBox** ويتم تقسيمهم إلى مجموعتين في **GroupBox** مجموعة مكونة من 3 **CheckBox** وهي مجموعة **Manufacture** ومجموعة الـ 6 **CheckBox** وهي مجموعة **Accessories**

4. عدد 1 **ListBox** وهو لإدراج موديلات الأجهزة بداخلة

5. عدد 1 **ComboBox** ويتم فيه إدراج طريقة الدفع وهناك اختيار بين الدفع عن طريق شيك أو الدفع عن طريق كرت



6. عدد 1 **BUTTON** وهو لتنفيذ عملية الشراء من خلاله وهو زر عادي جداً كما بالشكل



7. عدد 20 **PictureBox** ويتم تصنيف هذه الصور إلى 3 أجزاء وجعلهم مختلفين مع بداية التشغيل

A. مجموعة صور للشركة المصنعة والتي يتم التعامل معها من خلال عدد **Checkbox** من (1 إلى 3) لأننا اخترنا ثلاث لوجوهات لثلاث شركات فقط المصنعة للكمبيوتر ويكون الصور في **PictureBox** من (1 إلى 3) بها بالتنسيق التالي





B. مجموعة صور الأجهزة والتي يتم اختيارها من خلال **Listbox** وعددهم 9 صور على أساس أن كل شركة مساهمة بنوع كمبيوتر من الأنواع الآتية جهاز كمبيوتر مكتبي و لابتوب و ميني لابتوب ويكون الصور **PictureBox** من ( 4 إلى 12 ) كالتالي



C. مجموعة صور طريقة الدفع وهما صورتين فقط في **PictureBox** ( من 3 إلى 14 ) ويكون شكلهم كالتالي



D. مجموعة صور للإضافات وهما في **PictureBox** (من 15 إلى 20 ) يوضع فيها صورة الأجهزة المضافة إلى الجهاز الأصلي كوسائل مساعدة له ويكون تنسيقهم كالتالي

لاحظ انه لو كنا استخدمنا الطريقة السابقة في التمرين رقم 7 والخاصة بالـ **PictureBox** سوف نقوم بعمل عدد 9 **PictureBox** فقط بدلا من 20 **PictureBox** وسوف نوفر أيضا في كتابة الأكواد لها لكني اخترت الطريقة المطولة هذه لتكرار الأوامر عليكم والتمرن عليها

يتم ترتيب كل ما سبق في نموذج واحد للحصول على نموذج كالتالي

مجموعة **PictureBox** من 3 إلى 12

مجموعة **PictureBox** من 1 إلى 3

مجموعة **Checkbox** من 1 إلى 3 داخل **GroupBox**

مجموعة **Checkbox** من 4 إلى 9 داخل **GroupBox**

مجموعة **PictureBox** من 13 إلى 14

الـ **label** لكتابة الجملة به حسب التنسيق

الـ **Listbox** الذي سيتم اختيار موديلات الأجهزة منه

الـ **ComboBox** الذي سيتم إدراج وسيلة الدفع به

مجموعة **PictureBox** من 15 إلى 20

الزر **button** لتنفيذ عملية الشراء

## كتابة الأكواد للتمرين

☒ أولاً

الـ **Label** لا يوجد به اكواد فقط التنسيق كما بالشكل وكتابة الجملة الظاهرة إمامكم ( **BUY YOUR COMPUTER** )

☒ ثانياً

الـ **PictureBox** جمعها لا يوجد بها اكواد فقط الصور بالطريقة الموضحة وكلها بنفس إعداد الخصائص ويتم وضع كل مجموعة صور حسب التوضيح السابق فوق بعضهم وفي مكانهم الموضح بالنموذج وذلك على أساس إن يتم ظهورهم في نفس المكان عند اختيارهم ويتم تعديل الخاصية **visible** ليتم إخفائهم جميعاً عند بدا التشغيل **F5**

☒ ثالثاً

الـ **Checkbox** مجموعة **Manufacture** وهي يوجد بها عدد **3 Checkbox** ( من 1 الى 3 ) وهي مرتبطة بمجموعة الصور ( من 1 إلى 3 ) وهي صور للعلامات المميزة للشركات ويكون كتابة الكود في **CheckBox1** والمسمى **ACER** كالتالي

هذه الأكواد تنفذ في حالة تغيير حالة الاختيار في **CheckBox1**

طبعا يتم في بداية الأمر تجهيز **CheckBox1** وجعلها غير مختارة عند التشغيل ويتم ذلك من صندوق الخصائص

```
Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged
```

هذا الكود لإخفاء العلامة ✓ من **CheckBox2** و **CheckBox3**

```
CheckBox2.Checked = False
CheckBox3.Checked = False
```

نلاحظ أن الـ **CheckBox** الثلاثة تتعامل مع عدد 12 صورة أي **PictureBox** ولذلك يجب اظهار الصورة الخاصة بالـ **CheckBox1** وإخفاء الـ 11 صورة الخاصة بالأجهزة الأخرى

```
PictureBox1.Visible = True
PictureBox2.Visible = False
PictureBox3.Visible = False
PictureBox4.Visible = False
PictureBox5.Visible = False
PictureBox6.Visible = False
PictureBox7.Visible = False
PictureBox8.Visible = False
PictureBox9.Visible = False
PictureBox10.Visible = False
PictureBox11.Visible = False
PictureBox12.Visible = False
```

هنا يتم وضع الشرط انة في حالة عدم اختيار **CheckBox1** تختفي الصورة الخاصة به والموجودة في **PictureBox1**

```
If CheckBox1.Checked = False Then PictureBox1.Visible = False
End Sub
```

نلاحظ أن الأكواد السابقة سوف تتكرر لكل من **CheckBox2** و **CheckBox3** وذلك لتشابههم جميعاً ولكن الاختلاف سيكون في **PictureBox** أيهم يكون ظاهر وأيهم مختفي حسب الصورة المراد إظهارها وإخفاء باقي الصور لاحظ الفرق في التمرين المرفق

لاحظ هنا عدد الأكواد هنا كبير لأننا اختارنا أن نتعامل مع كل صورة على حدة **PictureBox** ولكن لو اتبعنا الطريقة في التمرين رقم 7 سنوفر أكثر من 11 كود في المرة الواحدة بإجمالي 33 كود للـ **CheckBox** من 1 إلى 3 فقط

☒ رابعا

الـ **ListBox** وهو خاص باختيار نوع أو موديل الكمبيوتر الذي تم اختيار الشركة المنتجة له سابقا وهي خاصة بالتعامل مع الصور **PictureBox** ( من 4 إلى 12 ) بعد كتابة المدخلات بة بإحدى الطرق السابقة وإظهارها كما بالشكل تأتي مرحلة كتابة الأكواد له وتكون كمل يلي



Case 0  
Case 1  
Case 3

من الطبيعي أن تتغير صور الأجهزة وذلك باختلاف الماركة ( الشركة المصنعة ) ونوع الجهاز نفسه **case0,case1,case2** حسب كل شركة وهذا سيكون شرطنا في كتابة الكود هنا

```
Private Sub ListBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ListBox1.SelectedIndexChanged
```

```
    If CheckBox1.Checked = True Then
        Select Case ListBox1.SelectedIndex
            Case 0
                PictureBox4.Visible = True
                PictureBox5.Visible = False
                PictureBox6.Visible = False
                PictureBox7.Visible = False
                PictureBox8.Visible = False
                PictureBox9.Visible = False
                PictureBox10.Visible = False
                PictureBox11.Visible = False
                PictureBox12.Visible = False
        End Select
    End If
```

```
    If CheckBox1.Checked = True Then
        Select Case ListBox1.SelectedIndex
            Case 1
                PictureBox5.Visible = True
                PictureBox4.Visible = False
                PictureBox6.Visible = False
                PictureBox7.Visible = False
                PictureBox8.Visible = False
                PictureBox9.Visible = False
                PictureBox10.Visible = False
                PictureBox11.Visible = False
                PictureBox12.Visible = False
        End Select
    End If
```

```
    If CheckBox1.Checked = True Then
        Select Case ListBox1.SelectedIndex
            Case 2
                PictureBox6.Visible = True
                PictureBox4.Visible = False
                PictureBox5.Visible = False
                PictureBox7.Visible = False
                PictureBox8.Visible = False
                PictureBox9.Visible = False
                PictureBox10.Visible = False
                PictureBox11.Visible = False
                PictureBox12.Visible = False
        End Select
    End Select
```

في حالة اختيار **CheckBox1** مثلا وهو الخاص بشركة **ACER** يتم اختيار الحالة الاولى **CASE0** وهي مثلا **PC COMPUTER** وفي حالة اختيارها يتحقق ان تظهر الصورة في **PictureBox4** وهي الخاصة بعرض صورة **PC COMPUTER** والخاص بشركة **DELL** التي تم اختيارها في الشرط الاول

في حالة اختيار **CheckBox1** مثلا وهو الخاص بشركة **ACER** يتم اختيار الحالة الثانية **CASE1** وهي مثلا **LAPTOP COMPUTER** وفي حالة اختيارها يتحقق ان تظهر الصورة في **PictureBox5** وهي الخاصة بعرض صورة **LAPTOP COMPUTER** والخاص بشركة **DELL** التي تم اختيارها في الشرط

في حالة اختيار **CheckBox1** مثلا وهو الخاص بشركة **ACER** يتم اختيار الحالة الثالثة **CASE2** وهي مثلا **MINI NOTEBOOK** وفي حالة اختيارها يتحقق ان تظهر الصورة في **PictureBox6** وهي الخاصة بعرض صورة **MINI NOTEBOOK** والخاص بشركة **DELL** التي تم اختيارها في الشرط الاول وإخفاء باقي **PictureBox** الخاصة بالأجهزة

ملاحظة انه سيتم تكرير الخطوات وذلك في حالة اختيار **CheckBox2** و **CheckBox3** الخاص بالشركتين الاخرين **HP** و **DELL** مع اختلاف الصور الخاصة بكل منهم لاحظ الفروق في التمرين المرفق ولا تتعدى الفروق غير اظهار او اخفاء صورة معينة من خلال التحكم في **PictureBox12** الخاص بها واظهاره واخفاء الباقي وأيضا لاحظوا كمية الأكواد المستخدمة في كل حالة وكيف كان من الممكن اختصارها لو استخدمنا الطريقة السابقة في التمرين رقم 7



## خامسا ☒

الـ **CheckBox** الخاصة بمجموعة **ACCESSORIES** وهي يوجد بها عدد 6 **CheckBox** ( من 4 الى 9 ) وهي خاصة بالتحكم في إظهار **PictureBox** ( من 15 الى 20 ) كما سبقه ووضحنا



```
Private Sub CheckBox4_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox4.CheckedChanged
```

```
    PictureBox15.Visible = True
    If CheckBox4.Checked = False Then PictureBox15.Visible = False
End Sub
```

وهنا يتم وضع شرط أنه في حالة عدم اختيار **CheckBox4** أي إزالة علامة الاختيار مئة فطبعا بالتالي تختفي الصورة الموجودة في **PictureBox15**

وهنا في حالة اختيار **CheckBox4** يتم اظهار الصورة **PictureBox15** والمفروض ان بها صورة **KEYBOARD** حسب برمجة **CheckBox4**

وطبعا سوف تتكرر العملية لكل من **CheckBox5** و **CheckBox6** و **CheckBox7** و **CheckBox8** و **CheckBox9** باختلاف إظهار **PictureBox** الخاص لكل منهم والتحكم في شرط الاختيار كل حسب حالته

## سادسا ☒

الـ **ComboBox** وهو خاص بالتعامل مع الصور الخاصة بـ **PictureBox** ( من 13 إلى 14 ) بالإظهار أو الإخفاء حسب الحالة المختارة وذلك بعد كتابة كود الحالة بإحدى الطرق السابقة له



Case 0  
Case 1

```
Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
    Select Case ComboBox1.SelectedIndex
        Case 0
            PictureBox13.Visible = True
            PictureBox14.Visible = False
        Case 1
            PictureBox14.Visible = True
            PictureBox13.Visible = False
    End Select
End Sub
```

وهي عند اختيار الحالة الثانية **case 1** في **ComboBox1** يتم اظهار الصورة الموجودة في **PictureBox14** الخاصة بطريقة الدفع **sheque** وإخفاء الصورة الموجودة في **PictureBox13** الخاصة بالطريقة الأخرى

وهي عند اختيار الحالة الأولى **case 0** في **ComboBox1** يتم اظهار الصورة الموجودة في **PictureBox13** الخاصة بطريقة الدفع **credit card** وإخفاء الصورة الموجودة في **PictureBox14** الخاصة بالطريقة الأخرى

سابقا ☒

الـ button وهو زر إعطاء أمر الشراء الكمبيوتر والأجهزة الإضافية في حالة وجودها في حالة الاختيار



نلاحظ أن الزر لا يمكن تفعيله الا في حالتين ولا بد من توافرهم معا

**الحالة الأولى**

لا بد من وجود جهاز تشترية اي لا بد من اختيار إحدى الشركات للتعامل معها

**الحالة الثانية**

لا بد من اختيار إحدى وسائل الدفع المتوفرة

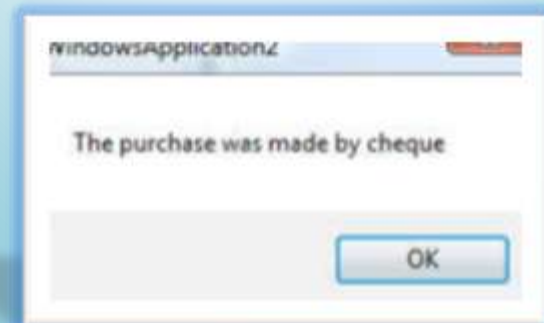
في حالة عم تحقق الشرطين معا سوف تتوقف العملية ولا ينفذ الزر الكود المطلوب منه تنفيذه

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

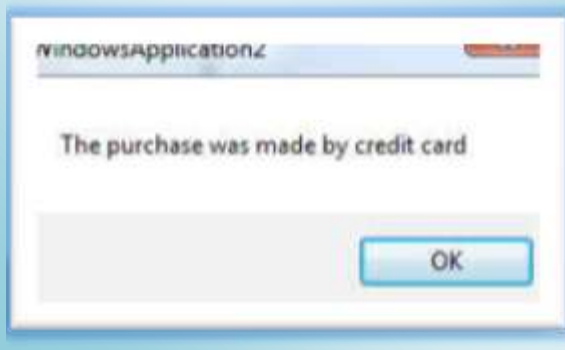
    If CheckBox1.Checked Or CheckBox2.Checked Or CheckBox3.Checked Then
        Select Case ComboBox1.SelectedIndex
            Case 0
                MsgBox(" The purchase was made by credit card ")
            Case 1
                MsgBox(" The purchase was made by cheque ")
        End Select
    End If

End Sub
```

شرطنا هنا أنه في حالة اختيار كل من **CheckBox1** او **CheckBox2** او **CheckBox3** وايضا لو تم اختيار الحالة **case 1** من **ComboBox1** والخاصة باختيار وسيلة الدفع **cheque** تظهر لنا الرسالة التالية عند الضغط على الزر **buy now**

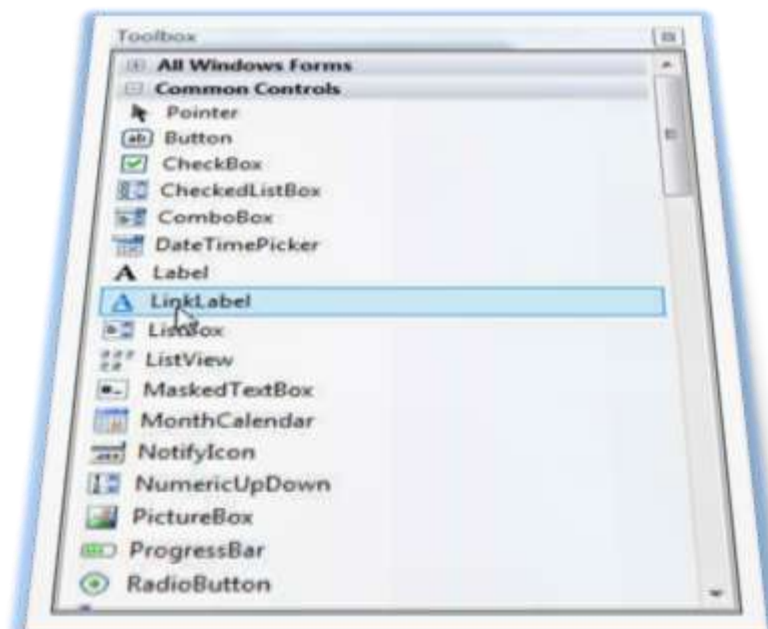


شرطنا هنا أنه في حالة اختيار كل من **CheckBox1** او **CheckBox2** او **CheckBox3** وايضا لو تم اختيار الحالة **case 0** من **ComboBox1** والخاصة باختيار وسيلة الدفع **credit card** تظهر لنا الرسالة التالية عند الضغط على الزر **buy now**



هكذا نكون انتهينا من التمرين و لك أن تقارن بين الطريقتين في استخدام **PictureBox** واختيار الانسب

تحميل التمرين الثامن



## التمرين التاسع

سنتعرف معا في هذا التمرين على كيفية إدراج رابط لموقع على صفحة النموذج لدينا وذلك من خلال استخدام الأداة **linklabel** وذلك من خلال صندوق الأدوات وهي الأداة المتوفرة لإضافة أي رابط له علاقة بالانترنت على النموذج

والمطلوب هو عمل نموذج به عدة روابط لمواقع على الانترنت مع ملاحظة

إن بعض أنظمة التشغيل تحتوي على أكثر من متصفح غير المتصفح الافتراضي **Internet Explorer** مثل متصفح **Mozilla Firefox** مثلا أو غير ذلك في حالة وجوده فقط على الجهاز فيمكننا توجيه الكود إلى استخدام المتصفح المفضل لدينا عند فتح الموقع على الانترنت وذلك بإضافة اسم المتصفح كما يلي قبل الموقع المراد فتحه من خلاله مثل **"Firefox.exe"** في حالة استخدامه أو **"iexplore.exe"** في حالة استخدامه أو ترك عنوان الموقع كما فقط ليفتح مع المتصفح الافتراضي بدون توجيه الكود

بعض إضافة الـ **linklabel** وتجهيز خصائصها لتظهر كما بالصور يتم إدراج الكود لكل منها ( الصور **PictureBox** هنا لمجرد التوضيح وليس لها أي دخل في الكود وكذلك حال الـ **GroupBox** وهما للتنسيق فقط على إن يتم كتابة الأكواد فقط في **linklabel** )



```
Private Sub LinkLabel2_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel2.LinkClicked
    System.Diagnostics.Process.Start("firefox.exe", "https://www.google.com")
End Sub
```

```
Private Sub LinkLabel11_LinkClicked_1(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel11.LinkClicked
    System.Diagnostics.Process.Start("iexplore.exe", "https://www.google.com")
End Sub
```

```
Private Sub LinkLabel4_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel4.LinkClicked
    System.Diagnostics.Process.Start("https://www.opendrive.com/files/9517664_i07P4_415e/Access2007%20part2.pdf")
End Sub
```

```
Private Sub LinkLabel3_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel3.LinkClicked
    System.Diagnostics.Process.Start("https://www.opendrive.com/files/9507601_K45tb_fc73/Access2007%20part1.pdf")
End Sub
```

تحميل التمرين التاسع

## القوائم والأدوات ونوافذ الحوار

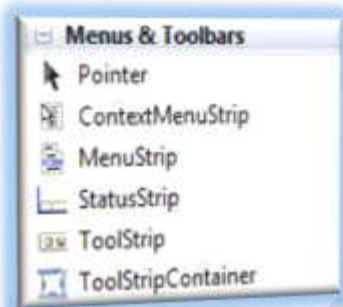
سنبدأ في هذا الفصل بتعلم القوائم وأشرطة الأدوات وصناديق الحوار ولنتعرف عليهم أكثر نفتح أي برنامج يتعامل مع بيئة ويندوز وسوف نأخذ في مثالنا هذا برنامج الفيچوال بيسك 2008 نفسه



تختلف شرائط القوائم من برنامج إلى آخر حسب نوع البرنامج وطبيعة شرائط القوائم والأدوات التي يستخدمها للتعامل مع محتواة ولكن معظم البرامج تتشابه في القوائم

( FILE,EDIT,VIEW,HELP )

في الأسماء فقط ولكن المضمون كل برنامج على حسب استخدامه



سنتعلم معا في هذا الجزء التعامل مع الأداة **MenuStrip** وتستخدم لعمل شريط القوائم والأداة **ToolStrip** وتستخدم في عمل شريط الأدوات ويتم الوصول إليهم طبعا من صندوق أدوات البرنامج وهما من الأدوات التي تضيف مظهر جمالي واحترافي للبرنامج التي تقوم عليه

إضافة القوائم باستخدام الأداة **MenuStrip**

يمكنك إضافة القوائم إلى برنامجك وكذلك يمكنك التعديل على استخدام الأداة **MenuStrip** كما يمكنك إضافة اللمسات الخاصة على القوائم مثل مفاتيح الاختصار وعلامات تأشير فمع هذه الأداة تستطيع إضافة القوائم الأساسية لبرنامجك بضغطة زر واحدة بدون تعقيد طبعا بعد إضافة القوائم لبرنامجك لابد من إضافة الكود لهذه القوائم لان بيئة التطوير تساعدك فقط في تصميم القوائم

كما يمكننا إضافة مفاتيح الوصول وهي تلك المفاتيح أو الحروف التي تضغطها بالإضافة إلى زر الكي بورد **Alt** للوصول السريع إلى قائمة ما فمثلا لفتح قائمة **File** يتم الضغط على **Alt+F** من لوحة المفاتيح ويتم ذلك بان ندرج العلامة **&** قبل الحرف **F** الذي تريد ان يكون هو مفتاح الوصول له ويتم تلقائيا إدراج سطر تحت الحرف المعين في بعض أنظمة التشغيل حسب الإعدادات الإقليمية للويندوز فمثلا

القائمة	مفتاح الوصول	طريقة كتابته في النموذج أو صندوق الخصائص
File	Alt+F	&File
Edit	Alt+D	E&dit
View	Alt+W	Vie&w

## قواعد عامة متعارف عليها لإضافة القوائم إلى برنامجك ومنها

1. في حالة القوائم باللغة الانجليزية ابدأ كل بند من القوائم بحرف **Capital** واحرص إن يكون كل بند عبارة عن كلمة واحدة أو اثنتين كحد أقصى
2. اجعل قوائمك سهلة ومفهومة واختر كلمات سهلة للتعبير عن وظائفها لا تجعل المستخدم يحترق في وظيفة بند من بنود القائمة فمثلا لإغلاق البرامج استخدم الكلمة "إغلاق" أو **Close** حسب لغة البرنامج
3. أضف مفاتيح للوصول لكل بند في قوائمك قدر المستطاع ويفضل أن يكون مفتاح الوصول هو الحرف الأول في البند لو أمكن
4. يمكنك تمييز البند الذي يحتوي على علامة وصول بوضع ... مثلا إمام أسمة للدلالة على أنه مفتاح وصول أو يتم كتابة مفتاح الوصول من خلال إضافته في صندوق الأدوات لهذا البند مثلا (**Ctrl+F**)
5. قم بجعل كافة البنود المتشابهة تحت قائمة واحدة، فمثلا البنود الخاصة بالفتح والحفظ والطباعة تحت قائمة واحدة
6. إذا كان لديك بند من بنود القائمة يستخدم طريقة الفتح والغلق قم بإضافة زر تأشير  **On/Off** بجانب البند فإذا كان المؤشر موجود يكون معناه مفتوح والعكس بالعكس

للعلم القواعد أعلاه ليست ملزمة لأي مصمم عند برمجة القوائم ولكن المبرمج المتطلع إلى الاحتراف الذي يريد تصميم البرامج بمعاييرية أو وفق المعايير القياسية وعليه إتباع القواعد أعلاه وغيرها من معايير القوائم وانظروا بأنفسكم إلى البرامج العالمية المشهورة ستجد إن مصمميها اهتموا بهذه المعايير

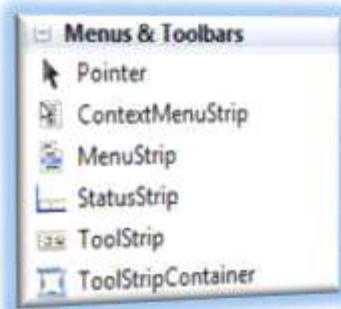
وللتجربة قم بفتح نموذج عمل جديد وباختيار الأداة **MenuStrip** من صندوق الأدوات وإدراجها داخل النموذج والعمل عليها مع إدراج مفاتيح الوصول السريع لها و مراعاة ما سبق في الجدول السابق للقوائم

يظهر المكون في صفحة البرنامج تلقائياً

هنا يتم كتابة أسماء البنود المدرجة داخل القائمة مثال على البنود المدرجة داخل قائمة **File**  
**Open, Save, Save as, close, Exit**

هنا يتم كتابة القائمة الرئيسية مثال على ذلك قائمة **File. Edit. View**

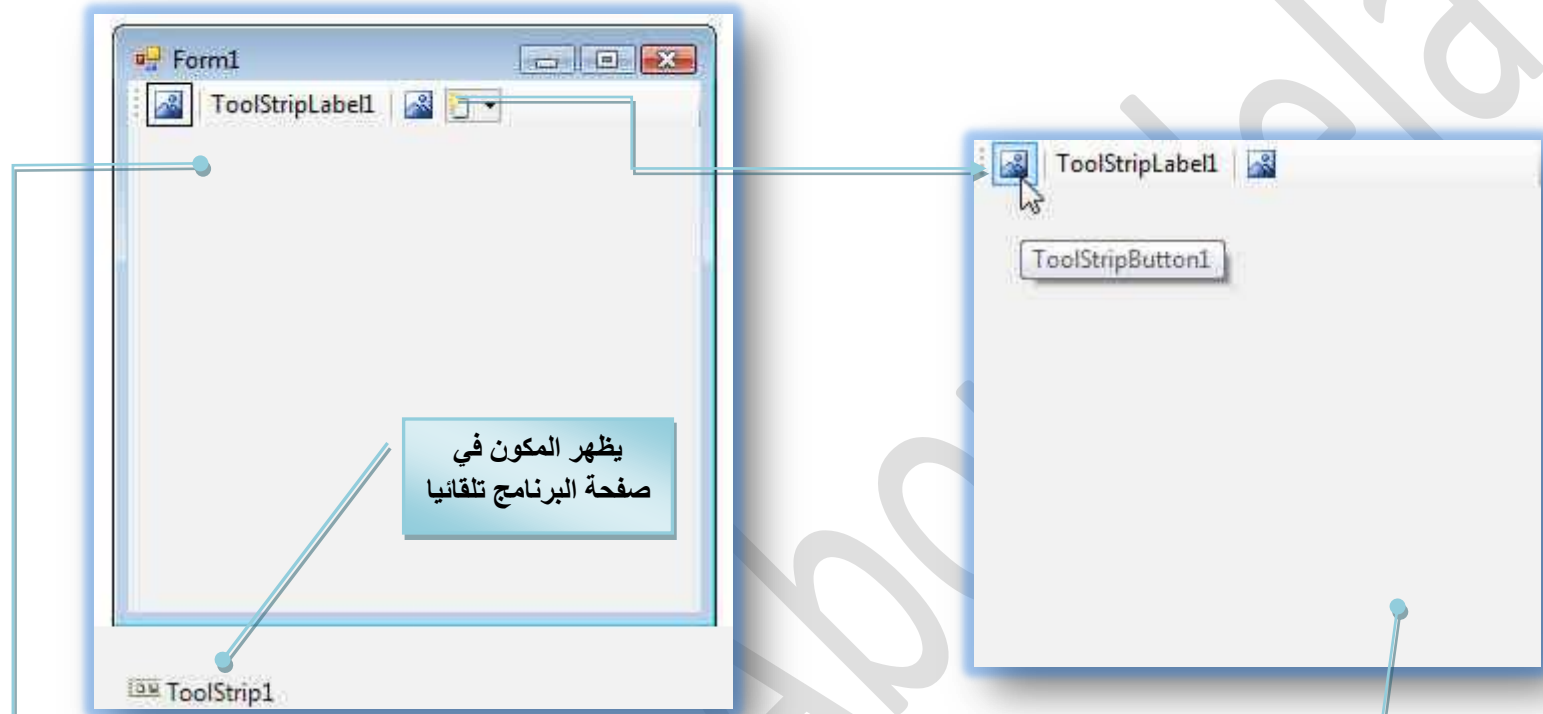
عند التنفيذ **F5** يكون النموذج كما هو موضح بالضغط على قائمة **File** تظهر لنا القائمة المنسدلة منها وبها البنود التي تم إضافتها إليها



### إضافة أشرطة الأدوات Toolbars باستخدام ToolStrip

بإمكانك إضافة شريط للأدوات في برنامجك لتقوم بنفس العمل التي تقوم به القوائم بحيث يختار المستخدم شريط الأدوات أو القوائم حسب رغبته في التعامل مع البرنامج ويعتبر هذا نوع من التسهيل على المستخدم بحيث يستخدم الماوس أو الكي بورد ولديه أكثر من بديل للتعامل في واجهة المستخدم و توجد بها العديد من الإضافات والميزات حيث يمكنك إضافة فواصل و صناديق تأشير وأماكن للنصوص وذلك من خلال إضافة الأداة ToolStrip من خلال صندوق الأدوات الخاص بالبرنامج

وللتجربة قم بفتح نموذج عمل جديد وباختيار الأداة ToolStrip من صندوق الأدوات وإدراجها داخل النموذج والعمل عليها

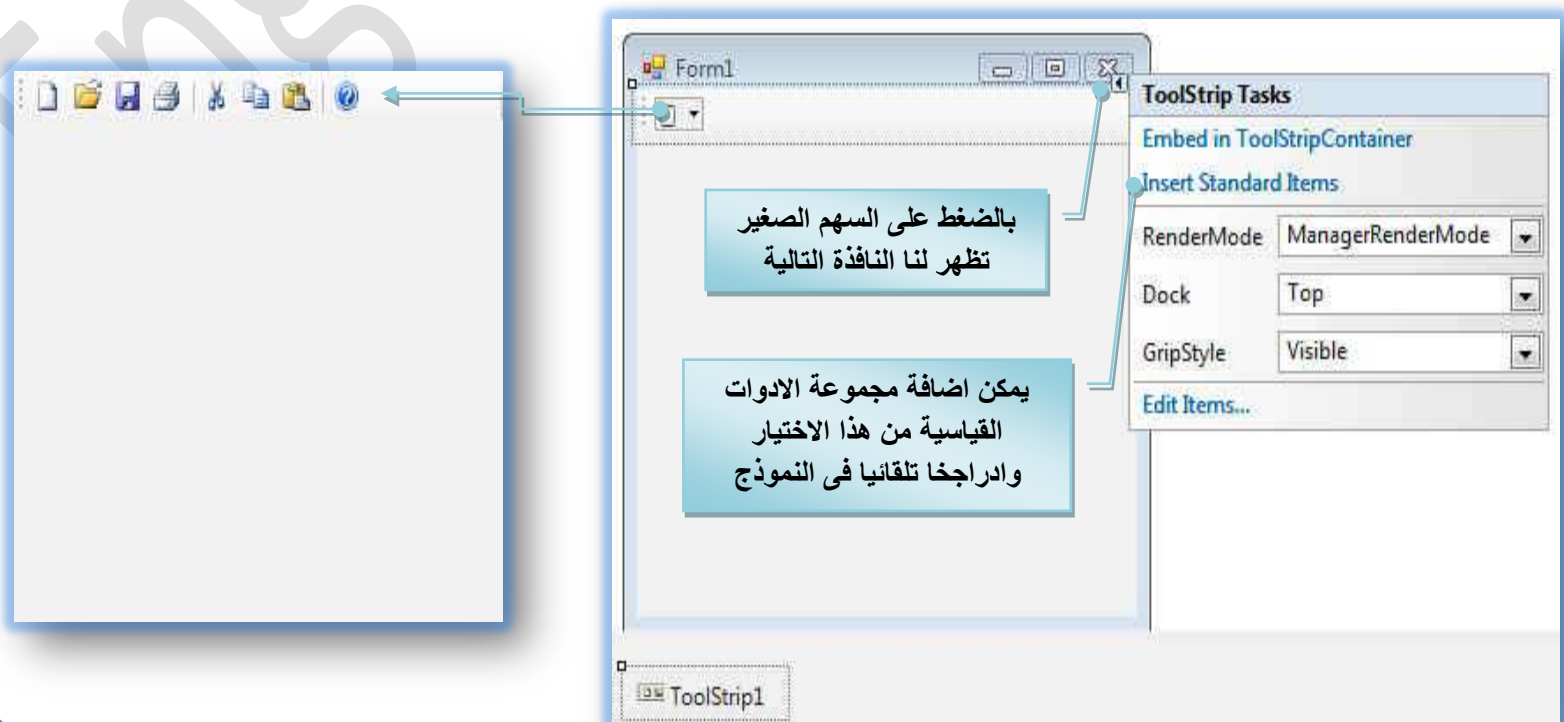


يظهر المكون في  
صفحة البرنامج تلقائياً

هنا يتم اختيار الأداة من عدة اختيارات منها زر ونص  
وفاصل وغيرها حسب التصميم

عند التنفيذ F5 يكون النموذج كما هو موضح  
بالضغط على الزر أو النص في شريط الأدوات تتم  
تنفيذ العملية طبعاً بعد إضافة الكود لها عند التصميم

ويمكن أيضاً إضافة مجموعة الأدوات القياسية بهذه الطريقة



بالضغط على السهم الصغير  
تظهر لنا النافذة التالية

يمكن إضافة مجموعة الأدوات  
القياسية من هذا الاختيار  
وإدراجها تلقائياً في النموذج

## التمرين العاشر

المطلوب هو عمل برنامج لإظهار الوقت والتاريخ ويتكون البرنامج من شريط الأدوات القائمة **Information** والذي تظهر منها بندين هما **Time, Date** وأيضا يحتوى على شريط الأدوات الذي توجد به أداتين هما **Time, Date** ممثلين في الأيقونتين الموضحين بالشكل مع العلم أنه في حالة اختيار الوقت أو التاريخ في أى من الطريقتين تظهر لنا صورة مرتبطة بالوقت والتاريخ مع إظهار الوقت والتاريخ في كل حالة



## الأدوات المستخدمة

1. **ToolStrip** وهي تستخدم لعمل شريط الأدوات والمكون من الأيقونتين كما بالشكل راجع الطريقة مما سبق
2. **MenuStrip** وهو لعمل شريط القوائم والمكون من البندين **Time** و **Date** وراعى إدخال مفاتيح الوصول كما شرحنا من قبل لكل منهم
3. **PictureBox** وعددهم 2 هم لإدراج الصورتين الخاصتين بالوقت والتاريخ لكل حالة
4. **Textbox** وهو ليتم إظهار التاريخ أو الوقت به عند الاختيار فيما بينهم

بعد الانتهاء من عمل القوائم والأدوات والوصول بهم إلى هذا الشكل الموضح سابقا من خلال تغيير الخواص المطلوبة لكل منهم في صندوق الخواص لكل أداة على حدة تأتي مرحلة كتابة الأكواد وهي تكون كالتالي

طبعنا نحن متفقين أن الأمر الذي ينفذه البند **Time** من شريط القوائم هو نفس الأمل التي تنفذه صورة **Time** والموجودة بشريط الأدوات فيكون بذلك الكود واحد لكل منهم بمعنى أنه نفس الكود للثنتين وكذلك بالنسبة إلى **Date** في الحالتين

كود إظهار الوقت في **Textbox**

فيكون كتابة الكود في البند **Time** كالتالي

```
Private Sub TimeToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton3.Click
    TextBox1.Text = TimeString
    PictureBox1.Visible = True
    PictureBox2.Visible = False
End Sub
```

إخفاء الصورة المرتبطة بالتاريخ والموجودة في **picturebox2**

إظهار الصورة المرتبطة بالوقت والموجودة في **picturebox1**

طبعا نفس الكود السابق والمكتوب في البند **Time** من شريط القوائم سوف يكتب كما هو شريط الأداة للأداة **Time** لان المفروض أنهم يقومون بنفس الوظيفة ولن نكررها هنا راجع التمرين

وننتقل لكتابة الكود في الأداة **Date** من شريط الأدوات المدرج كالتالي

كود إظهار التاريخ في textbox

```
Private Sub ToolStripButton4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton4.Click
    TextBox1.Text = DateTime.Now.ToString("dd/MM/yyyy")
    PictureBox2.Visible = True
    PictureBox1.Visible = False
End Sub
```

إخفاء الصورة المرتبطة بالوقت والموجودة في picturebox1

إظهار الصورة المرتبطة بالتاريخ والموجودة في picturebox2

طبعا نفس الكود السابق والمكتوب في الأداة **Time** في شريط الأدوات سوف يكتب كما هو للبند **Time** في شريط القوائم لان المفروض أنهم يقومون بنفس الوظيفة ولن نكررها هنا راجع التمرين

الكود المستخدم لإدراج الوقت هو **TimeString**  
الكود المستخدم لإدراج التاريخ هو **DateString**

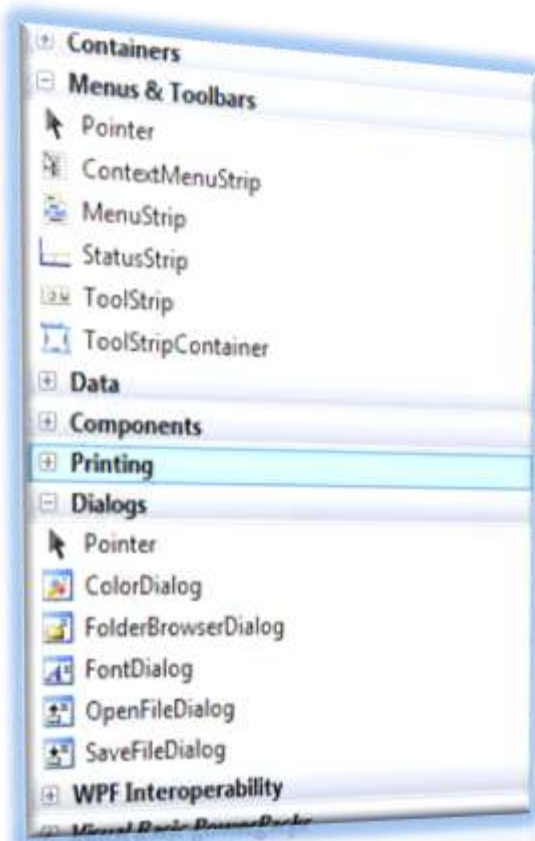
### تحميل التمرين العاشر

### نوافذ الحوار Using Dialog Box Controls

يحتوى الفيچوال بيسك 2008 على ثمان نوافذ حوار **Dialog Box Controls** جاهزة للاستخدام فهي معده لك مسبقاً لذلك فلا تحتاج إلى إعدادها من جديد وتستخدم للمهام المشهورة والمتكررة مثل نافذة فتح ملف أو غلقه أو طباعته و لن تحتاج إلا أن تقوم بإدخال كود للأحداث المتوفرة ضمن المكون ( نافذة الحوار ) أما بالنسبة للتصميم فقد تم تصميمها مسبقاً بحسب المعايير الموجودة مع نظام التشغيل الويندوز وهذه النوافذ هي

الهدف منه	اسم المكون
للحصول على ملف أو امتداد معين من الملفات من قرص معين من مجلد معين في الكمبيوتر للملفات الموجودة مسبقاً	OpenFileDialog
تحدد اسم القرص واسم المجلد وكذلك اسم الملف للملف الجديد	SaveFileDialog
تسمح للمستخدم من اختيار نوع الخط وطريقة عرضه	FontDialog
تسمح للمستخدم اختيار لون محدد من مجموعه ألوان.	ColorDialog
تسمح للمستخدم من التنقل بين المجلدات واختيار مجلد معين	FolderBrowserDialog
تسمح للمستخدم بتغيير خيارات الطباعة.	PrintDialog
تسمح للمستخدم بمعاينة المواد التي يريد طباعتها قبل الطباعة كما يفعل برنامج الورد	PrintPreviewDialog
تسمح للمستخدم بتغيير خيارات الصفحة بتغيير الحدود للصفحة وكذلك حجم الورق وغيرها من الإعدادات	PageSetupDialog





و الآن بعد أن عرفنا الفرق بين شريط القوائم وشريط الأدوات ينبغي لنا معرفة نوافذ الحوار وهي مجموعة من النوافذ التي تظهر لنا عند إجراء أمر معين وتساعدنا على الوصول إلى الهدف المرجو منها فمثلا عند اختيار بند **save** أو **open** في أي برنامج سبق وتعاملنا معه من قبل تظهر لنا نافذة في كلا من الحالتين ففي الأولى لحفظ الملف وتختار منها مكان حفظه وفي الحالة الثانية فتح ملف ومنها تختار موقع الملف لاستدعائه وفتحة والتعامل معه وهذه هي نوافذ الحوار فهي ليست بالجديدة علينا ولكن الجديد هنا أننا سوف نقوم بإدراجها داخل برنامجنا في **فيجوال بيسك** وهي تعتبر خطوة احترافية نحو البرمجة في **الفيجوال بيسك 2008** لأن من خلالها نستطيع أن نوجه برنامجنا إلى الهدف منه عن طريق إظهار نوافذ حوارية لتسهيل العملية على المستخدم للبرنامج وإظهار البرنامج بمظهر جمالي واحترافي في الوقت نفسه وقد تعلمنا مما سبق أن لكل كائن مدرج في النموذج أداة يتم استخدامها من صندوق الأدوات عند الحاجة إلى إدراجها بالعمل القائمين عليه وتظهر أسفل النموذج تلقائياً عند استدعائها للعمل فيه



الأداة الخاصة بالتعامل مع شريط القوائم

الأداة الخاصة بالتعامل مع نافذة الحوار فتح

الأداة الخاصة بالتعامل مع شريط الأدوات

الأداة الخاصة بالتعامل مع نافذة الحوار الألوان

سنحاول معا عمل تمرين يجمع كل ما سبق من أدوات فمثلا نريد أن نصمم تمرين بالشكل التالي مكون من

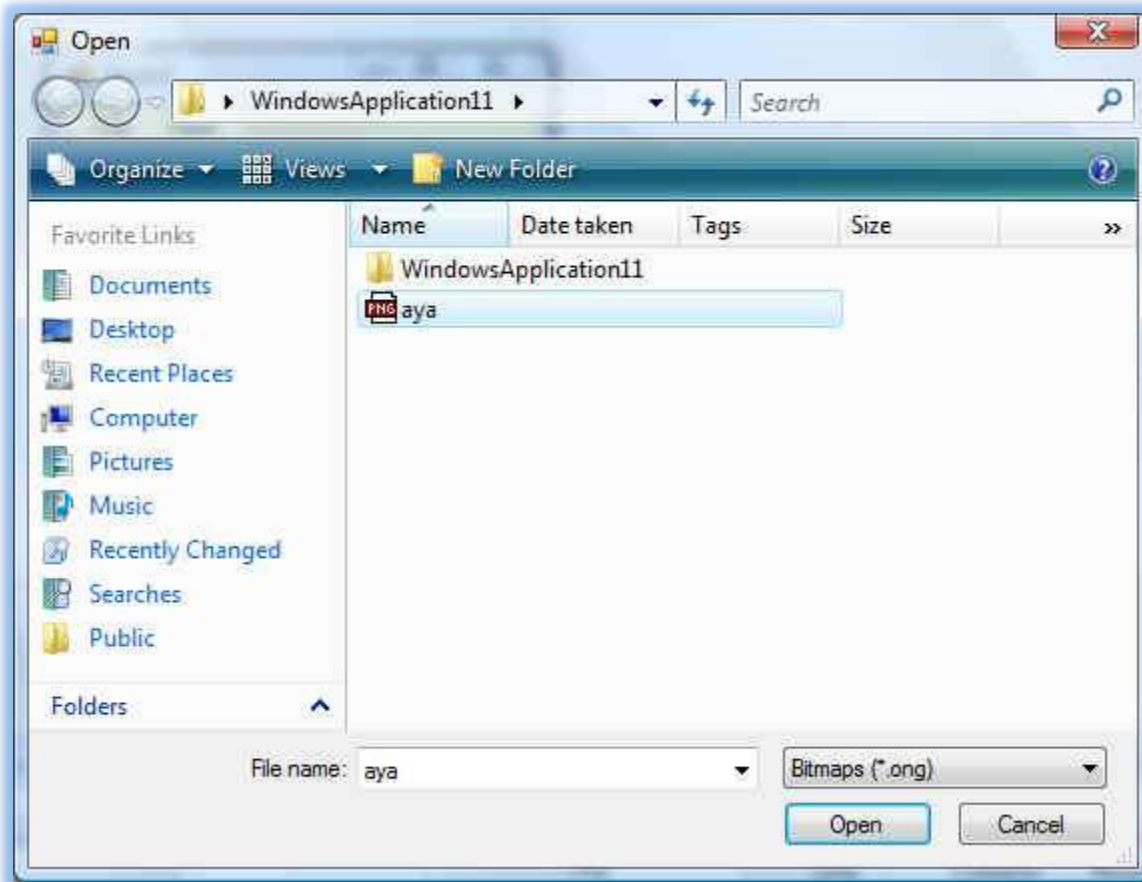
1. label يتم كتابة ( بس الله الرحمن الرحيم ) به

2. PictureBox ليتم وضع صورة الاية به

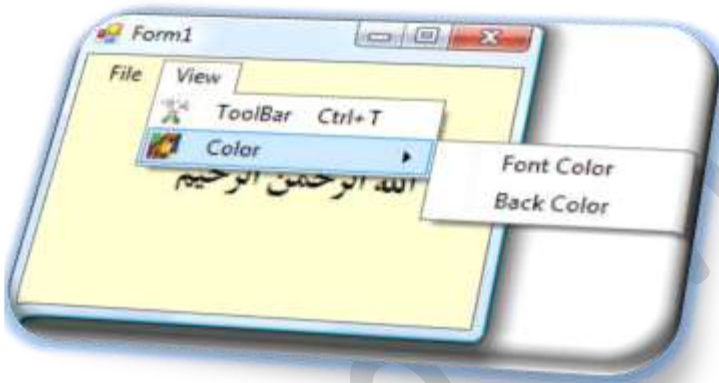
3. شريط قوائم به ( File, View )  
A. القائمة File يظهر قائمة بها



i. **Open** وهو لإظهار النافذة الحوارية التالية والمختصة بفتح ملف ما سوف يتم تحديده عند كتابة الكود

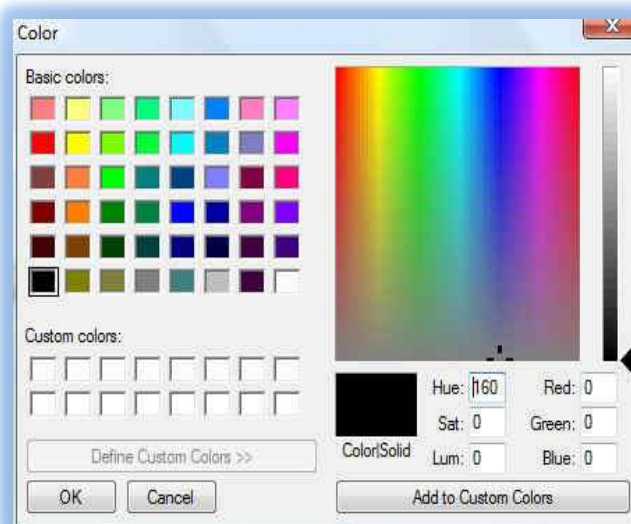


ii. **Exit** وهو للخروج فقط وإغلاق البرنامج



B. من القائمة **View** يظهر قائمة بها بند  
 i. أدوات **ToolBar** والى يقوم بإظهار شريط الأدوات أو إخفائه عند اختياره  
 ii. ألوان **Color** وتظهر لنا قائمة أخرى منسدلة منها يوجد بها  
 (a) **Font color** وهي لتغيير لون الخط  
 (b) **Back color** وهي لتغيير الخلفية

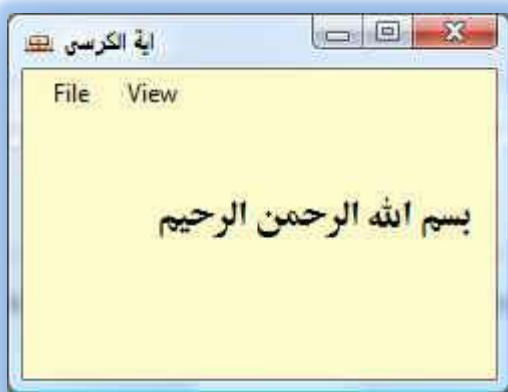
والذي يتم ظهور النافذة التالية والخاصة بتغيير ألوان كل من الخط والخلفية كلا على حدة





4. شريط أدوات و بة ثلاث أدوات هما **فتح** و**ألوان** و**مسح**

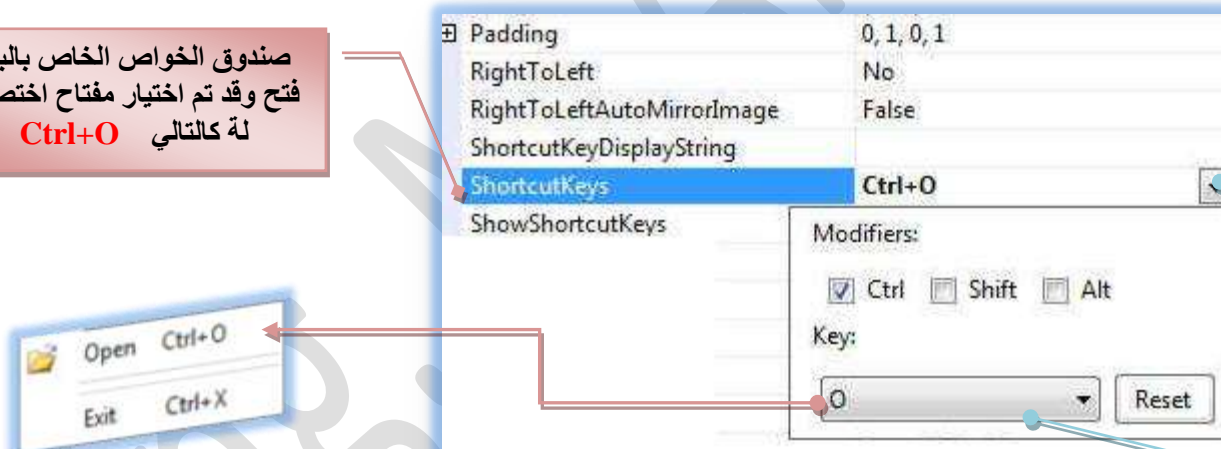
- ❖ **فتح** يقوم بنفس وظيفة بند **فتح** من القائمة **File**
- ❖ **ألوان** يقوم بنفس عمل البند **Font color** والمنسدل من البند **Color** في قائمة **View**
- ❖ **مسح** يقوم بإخفاء صورة السورة عند اختياره



على إن يظهر لنا البرنامج عند التشغيل بهذا الشكل والتنسيق وطبعا يتم ذلك من خلال تغيير خواص كل عنصر و إظهاره بالتنسيق المطلوب حسب التصميم

- ❑ تغيير اسم النموذج وإدراج أيقونة مميزة له
- ❑ تغيير لون الخلفية للنموذج وشريط الأدوات
- ❑ إخفاء شريط الأدوات من النموذج عند التشغيل
- ❑ إخفاء سورة الآية ( الصورة ) من القائمة لحين استدعائها
- ❑ جعل النموذج نفسه قابل للتمدد حيث يحتوى الصورة كاملة عند استدعائها وفي حالة إلغاء الصور يعود لحجمه الأول
- ❑ إضافة اختصارات إلى بنود البرنامج وقد تعلمنا طريقة مما سبق والاني نتعلم معا طريقة جديدة لإضافة اختصار إلى اي بند من خلال صندوق الخواص الخاصة بة لتظهر أمام أسمة فمثلا من خلال الخاصية **shortcutkeys** يمكننا اختيار اختصار لأي بند

صندوق الخواص الخاص بالبند فتح وقد تم اختيار مفتاح اختصار لة كالتالي **Ctrl+O**



ثم يتم من هنا اختيار المفتاح من بين قائمة تحتوي على العديد من الاختيارات في هذه الحالة تم اختيار المفتاح **O**

بالوقوف هنا إمام الخاصية تظهر لنا النافذة لنختار منها اي زر يتم استخدامه مع حرف الاختصار وهناك اختيارات بين **Ctrl or Shift or Alt**



- ❑ إضافة صورة بجانب اسم البند فمثلا نريد أن نضيف صورة دلالية للبند **Color** والموجود بالقائمة **View** أو الأداة **Color** ويكون كالتالي

عند التصميم وبالوقوف على كلمة **Color** في شريط الأدوات وبالنقر على زر الفارة الأيمن تظهر لنا القائمة التالية والتي من خلالها يتم التحكم في العديد من خواص البند القائمين عليه

## قائمة خواص البند

جعل الأداة مختارة منذ البداية أو لا

من هنا يتم إضافة الصورة بجانب الاسم

من هنا يتم التحكم في إظهار أو إخفاء مفتاح الاختصار بجانب الاسم

إظهار صندوق الخواص لهذه الأداة للتعامل معها وتنسيقها

التحكم في نوع الأداة وتغييرها

أوامر التحكم في الأداة من قص ولصق ومسح

وهناك اختلاف بسيط بين البند في شريط القوائم والأداة من شريط الأدوات عند التعامل معها بالطريقة السابقة تتمثل في التالي

## قائمة خواص الأداة

جعل الأداة مختارة منذ البداية أو لا

من هنا يتم إضافة الصورة

التحكم في شكل إظهار الأداة ويمكن إظهار اسم الأداة أو صورتها أو الاثنين معا

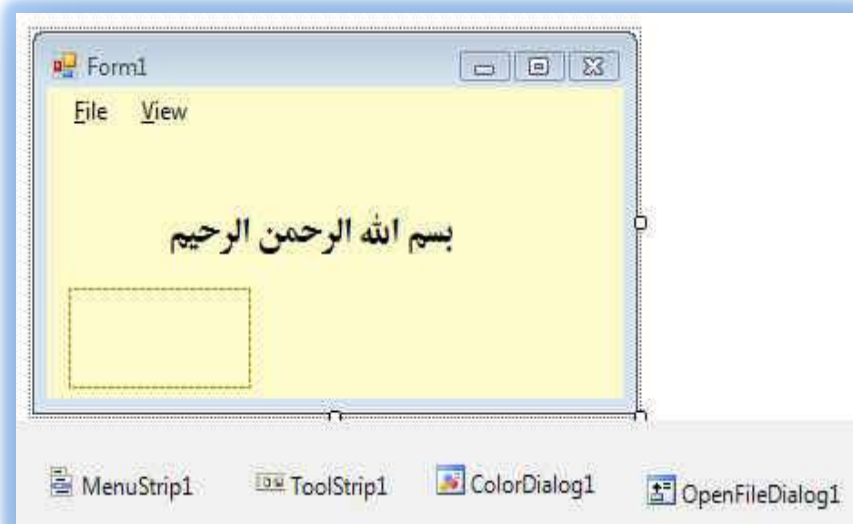
إظهار صندوق الخواص لهذه الأداة للتعامل معها وتنسيقها

إعداد محاذاة الأداة بالنسبة إلى شريط الأدوات

أوامر التحكم في الأداة من قص ولصق ومسح

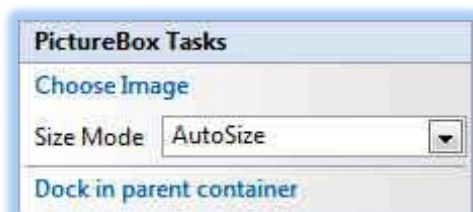
## التنسيق وكتابة الأكواد

بعد التجهيزات والإعدادات السابقة نصل بالنموذج إلى هذا الشكل عند التصميم وتأتي الآن مرحلة إدخال الأكواد لكل كائن موجود بالنموذج على حدة وسوف نستطرد معا في هذا التمرين في كتابة الأكواد



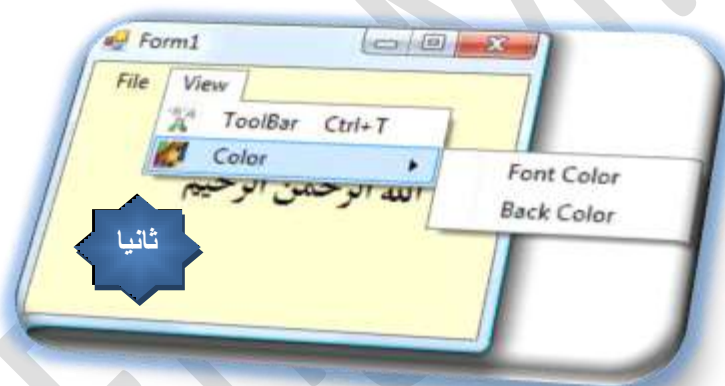
1. **Form1** لا توجد بة اكواد ولكن يتم تنسيقه حسب الشكل المطلوب وتتحكم بخاصية **Autosizemode** وجعلها **GrowAndShrink** وهي خاصية تجعله قابل للتمدد والانكماش تبعا لمحتواة

2. **Label1** لا يوجد بة اى اكواد فقط التنسيق وكتابة ( بسم اله الرحمن الرحيم )



3. **Picturebox1** لا يوجد بة اى اكواد فقط يتم تنسيقه لإظهار الصورة بة كاملة بطريقة تلقائية حسب حجم الصورة

4. **MenuStrip1** وهي الأداة الخاصة بشريط القوائم كما سبق وتعلمنا وتكون قوائمها وتنسيقها كما سبق وذكرنا في بداية التمرين ليكون الشكل كالتالي



مما سبق استطعنا أن نصل بالنموذج إلى هذه المرحلة وهذه القوائم بنفس إعداداتها وشكلها وتنسيقها كما سبق وذكرنا السبيل إلى ذلك سوف نقوم الآن بكتابة الأكواد فقط وليس الأكواد الأساسية للبرنامج

أولاً

قائمة File

A. كتابة الكود في البند **Open** بالوقوف والنقر مرتين لإدخال الكود التالي

```

OpenFileDialog1.Filter = "(*.png)|*.png"
If OpenFileDialog1.ShowDialog() = DialogResult.OK Then
    PictureBox1.Image = System.Drawing.Image.FromFile _
        (OpenFileDialog1.FileName)
End If

```

لو تم فتح نافذة الحوار (**OpenFileDialog1**) قم بعمل عملية تصفية لها (**Filter**) لكي تظهر فقط الملفات التي لها الامتداد **png** (وهو امتداد الصورة المستخدمة في المشروع ويمكن تغييرها وتغيير الامتداد حسب رغبة المصمم) وعند اختيار الصورة إذا قم بإظهار الملف المختار في **PictureBox1**

B. كتابة الكود في البند **Exit** بالوقوف علياً والنقر مرتين لإدخال الكود التالي

```

Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ExitToolStripMenuItem.Click
    Close()
End Sub

```

فقط يتم كتابة الكود  
**Close**

ثانياً

قائمة View

A. كتابة الكود في البند **Toolbar** وهو يستخدم لإظهار وإخفاء شريط الأدوات عند الضغط علياً طبعاً سوف يتم إخفاء شريط الأدوات عند بداية التشغيل كما سبق ووضحنا

```

Private Sub ToolbarToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ToolbarToolStripMenuItem.Click
    If ToolbarToolStripMenuItem.Checked = True Then ToolStrip1.Visible = True
    If ToolbarToolStripMenuItem.Checked = False Then ToolStrip1.Visible = False
End Sub

```

لو كان اختيار الزر **ToolStripMenuItem** محقق  
يكون شريط الأدوات **ToolStrip1** ظاهر في النموذج

لو كان اختيار الزر **ToolStripMenuItem** غير محقق  
يكون شريط الأدوات **ToolStrip1** مخفي في النموذج

**B.** كتابة الكود في البند **Color** وهو بالضغط عليه تظهر لنا بندين آخرين وهما الذي سوف يكون يهما الأكواد المطلوبة

**i.** البند **Font Color** وهو لتغيير لون الخط ويكون كتابة الكود به كالتالي

```
Private Sub FontColorToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles FontColorToolStripMenuItem.Click
    ColorDialog1.ShowDialog()
    Label1.ForeColor = ColorDialog1.Color
End Sub
```

وهنا نعطي أمر بالكود أن اللون المستخدم من نافذة الألوان التي ظهرت من قبل يتم تنفيذه على الجملة الموجود في **Label1** وهذا التنفيذ يكون مختص بلون الخط وهو **ForeColor**

يتم إظهار نافذة الحوار **ColorDialog1**

**ii.** البند **Back Color** وهو لتغيير لون الخلفية ويكون كتابة الكود به كالتالي

```
Private Sub BackColorToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles BackColorToolStripMenuItem.Click
    ColorDialog1.ShowDialog()
    Label1.BackColor = ColorDialog1.Color
End Sub
```

وهنا نعطي أمر بالكود أن اللون المستخدم من نافذة الألوان التي ظهرت من قبل يتم تنفيذه على الجملة الموجود في **Label1** وهذا التنفيذ يكون مختص بلون الخلفية وهو **BackColor**

يتم إظهار نافذة الحوار **ColorDialog1**

باتمام هذه المرحلة نكون قد انتهينا من كتابة الأكواد في شريط القوائم وبتنفيذ البرنامج **F5** والعمل على القائمة نرى أنها تؤدي العمل المرغوب منها

**5.** **ToolStrip1** وهي الأداة الخاصة بشريط الأدوات كما سبق وتعلمنا وتكون أدواتها وتنسيقها كما سبق وذكرنا في بداية التمرين ليكون الشكل كالتالي



نلاحظ أنها تتكون من ثلاث أدوات وهما ( **فتح وألوان ومسح** ) ويتم تنسيقهم كما اتفقنا عليه من قبل نلاحظ إن الهدف من الأداة ( **فتح وألوان** ) هما نفس الأهداف في البندين ( **Open, Font Color** ) ولذلك هناك طريقتين مختلفتين لكتابة الأكواد في هذين الأدوات

الطريقة الأولى  
هي نسخ الأكواد السابقة والخاصة بالبند ( **Open , Font Color** ) في كل من الأداة ( **فتح وألوان** ) بشريط الأدوات

ولكن لو أردنا كتابة الكود باحترافية أكثر وبأقل عدد من سطور الأكواد يتم اتباع الطريقة التالية

الطريقة الثانية  
وهي بإدراج هذا الكود في كلا من الحالتين السابقين ( نلاحظ أن الكود مختلف في الحالتين ولكن الهدف أو المضمون واحد )

### A. كتابة الكود في الزر فتح **Open**

```
Private Sub OpenToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripMenuItem.Click
    OpenToolStripButton_Click(sender, e)
End Sub
```

وهذا الكود يعتبر كود توجيهي إلى هذا الزر حيث إننا نوجه تعليمات الزر هذا إلى البحث عن الأداة **OpenToolStripButton\_Click** في حالة النقر عليها ومعرفة الكود الذي قامت بتنفيذه واستدعائه لتنفيذ نفس الكود بنفس الشروط لهذا الزر في شريط الأدوات

### B. كتابة الكود في الزر **Color**

```
Private Sub ToolStripButton2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton2.Click
    FontColorToolStripMenuItem_Click(sender, e)
End Sub
```

وهذا الكود يعتبر كود توجيهي إلى هذا الزر حيث إننا نوجه تعليمات الزر هذا إلى البحث عن الأداة **FontColorToolStripMenuItem\_Click** في حالة النقر عليها ومعرفة الكود الذي قامت بتنفيذه واستدعائه لتنفيذ نفس الكود بنفس الشروط لهذا الزر في شريط الأدوات

### C. كتابة الكود في الزر **Hide**

```
Private Sub ToolStripButton1_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton1.Click
    PictureBox1.Visible = False
End Sub
```

عند الضغط أو اختيار هذه الأداة يتم إخفاء **PictureBox1** وبذلك نقوم بإخفاء الصورة التي تم إدراجها فيه من قبل

وبذلك نكون قد انتهينا من إدراج الأكواد إلى البرنامج وعند إجراء عملية التنفيذ **F5** وتشغيل البرنامج سوف نرى ما قمنا بتنفيذه حقيقة إمامنا كل ما أرجوه منكم هو الاهتمام بالأكواد الجديدة بجانب صندوق الخصائص لكل أداة من الأدوات التي تم إضافتها في المشروع وستمكن من ذلك من خلال مقارنتهم ببعض من خلال التمرين المرفق

تحميل التمرين الحادي عشر



مما سبق نستطيع أن نلخص بعض الاستخدامات في فيجوال بيسك 2008 إلى التالي

الخطوات	من أجل
أضف <b>ToolStrip</b> إلى الفورم ثم قم بعمل <b>Right-Click</b> على الأزرار لتنظيم خواصهم وإضافة أزرار أخرى ثم <b>Double-Click</b> على أي بند لكتابة الكود الخاص به	إضافة شريط أدوات
أضف واحدة من الثمانية المكونات التابعة لنوافذ الحوار والموجودة في شريط الأدوات في <b>Dialogs and Printing Toolbars</b> إلى الفورم وقم بتغيير خصائصها من نافذة الخصائص	إضافة نافذة حوار
أضف المكون <b>OpenFileDialog</b> إلى النموذج والاستخدام نافذة الحوار لا بد من استخدام الطريقة <b>ShowDialogMethod</b> والخاصية <b>FileName</b> تحتوي على اسم الملف الذي تم اختياره لفتحة	إضافة نافذة حوار لفتح نوع من الملفات
أضف المكون <b>ColorDialog</b> إلى النموذج ولإظهار صندوق الألوان لا بد من استخدام الطريقة <b>ShowDialogMethod</b> والخاصية <b>Color</b> تحتوي على اسم الملف الذي تم اختياره لفتحة	استخدام نافذة حوار لفتح صندوق الألوان
أضف المكون <b>MenuStrip</b> إلى النموذج ثم اذهب إلى كلمة <b>Type Here</b> والموجودة اعلي النموذج وأضف القائمة التي تريدها وأضف لها قوائم فرعية حسب تصميمك	لإنشاء بند في القوائم
<b>Double-Click</b> على البند المراد وأضف الحرف & قبل الحرف الذي تريد تخصيصه وجعله مفتاح وصول له	إضافة مفتاح وصول للبند
حدد البند ثم اذهب إلى الخصائص واضبط الخاصية <b>ShortcutKeys</b> على الاختصار الذي تريد	إضافة اختصار للبند
بواسطة الماوس بطريقة السحب والإلقاء	تغيير ترتيب عناصر القوائم

### بسم الله الرحمن الرحيم

من الشرح السابق **لفيجوال بيسك 2008** تعلمنا كيف نبني برامج وكيفية تنظيم واجهة البرنامج للمستخدمين وكذلك كيفية التعديل على البرنامج كما تعلمنا كذلك كيف نتعامل مع بيئة التطوير وبإذن الله تعالى سوف نعرف معا الكثير في المرحلة القادمة عن مراحل الأكواد في **فيجوال بيسك 2008** وكيفية تعامل المعالج مع الأوامر البرمجية كما سوف نتعرف على كيفية استخدام الجمل الشرطية والمؤقتات والمصفوفات وجمل الدوران **Loops** والتعامل مع الجمل النصية **Debug** وكذلك كيف نتعامل مع أخطاء التشغيل وسنتعرف أكثر على تنظيم المكونات في واجهة المستخدم وبرمجة قواعد البيانات ( **Access 2007** ) وكذلك على برمجة مواقع الانترنت

## الجمل البرمجية Program Statement

هي عبارة عن ترابط منسق من الكلمات والخصائص والمكونات والمتغيرات والأرقام والمعاملات الخاصة والقيم الأخرى التي ترتب بشكل منطقي لتصنع أمر برمجي معين مفهوم لدى المترجم للغة الآلة **Compiler** قد تكون الجملة البرمجية عبارة عن كلمة واحدة فقط مثل كلمة **End** والتي تقوم بإغلاق البرنامج كما سبق وعرفنا أو قد تكون الجملة البرمجية عبارة عن مجموعة من المكونات مثل

**Label1.text = TimeString**

وهي جملة برمجية كاملة وفيها قد أسندنا الخاصية **Text** إلى التابعة للمكون **Label1** إلى الطريقة **TimeString** والخاصة بإظهار الوقت الحالي كما سبق وتعلمنا من التمارين السابقة والجملة البرمجية السابقة ( الكود ) والذي يترجمها الـ **Compiler** كالتالي ( قم بإظهار الوقت الحالي في الخاصية **Text** والخاصة بالمكون **Label1** )

ولابد لكي يقوم الـ **Compiler** بفهم الكود أو الجملة البرمجية وترجمتها للغة البرنامج بالطريقة الصحيحة أن تكون مكتوبة حسب مجموعة من الخطوات التي هو قادر على التعامل معها وداخلها في تكوين البرنامج **فيجوال بيسك 2008** ولكي نتعلم بناء جملة برمجية صحيحة لابد من الاطلاع على القوانين أو أساسيات بناء الجمل البرمجية وكيفية معالجة البيانات ضمن البرنامج لكتابة جملة برمجية صحيحة ولغة **فيجوال بيسك 2008** هي لغة برمجية سهلة تسهل على المبرمجين العديد من الصعاب لذلك فبناء برنامج بها سهل جداً ويكون قريب من اللغة العامية في بعض الأحيان كل هذا من أجل التسهيل على المبرمجين وجعلهم يفرغون عقولهم للأفكار الجديدة والتطويرية فبدلاً من كتابة صفتين من الكود لإنشاء فورم مثلاً تتم العملية فقط بواسطة السحب والإلقاء بواسطة الماوس وهناك العديد من الوسائل التي تبسط لنا البرمجة **بالفيجوال بيسك 2008** وفي نفس الوقت بيئة التطوير تساعدك في تحديد الأخطاء وتقديم الحلول الممكنة أو المقترحات الممكنة للمبرمج كما ستتعرف فيما بعد على الكائنات والدوال والكلمات والطرق والخصائص الموجودة مسبقاً في بيئة **.Net** الدوت نت سنتعلم كيف نستفيد منها لتطوير وتصميم برامج عملاق بإذن الله تعالى

## المتغيرات variables

المتغير هو مكان مؤقت لحفظ البيانات في البرنامج و تستطيع استخدام متغير واحد أو أكثر في برنامجك وقد تكون هذه المتغيرات كلمات أو أرقام أو تواريخ أو خصائص و باستخدام المتغيرات تستطيع تسمية كل نوع من أنواع البيانات باسم سهل التذكر ذو معنى مفيد ليساعد على تسهيل عملية البرمجة وتقوم المتغيرات بحفظ البيانات التي يدخلها المستخدم أو يتم جلبها من النظام أو من الشبكة أو غيرها من المصادر وقت عمل البرنامج **Run-Time** وقد تكون المتغيرات عبارة عن بيانات تمت معالجتها ببرنامجنا وقت عمل البرنامج فنستطيع أن نستعرض البيانات المخزنة في المتغيرات على الفورم أو تخزينها في قاعدة البيانات ( خزنها بشكل دائم ) لان المتغيرات تخزنها بشكل مؤقت فقط لحين إغلاق البرنامج أو للوقت الذي نحدده نحن كما إن استخدام المتغيرات في بيئة التطوير يلزمنا بتخطيط لمعرفة ما هي المتغيرات التي نحتاجها لان حجز المتغيرات في برنامجنا مثل حجز كرسي في قاعة المحاضرة فلا نقوم بحجز الكرسي إلا إذا كنا محتاجين له فعلاً

## تعريف المتغيرات بواسطة الكلمة Dim

لنعرف متغير في فيجوال بيسك لا بد من استخدام الكلمة **Dim** وهي اختصار لكلمة **Dimension** واتفق برمجياً على إن هذه الكلمة تأمر الكمبيوتر بحجز مكان في الذاكرة للمتغير هذا وتسمح للكمبيوتر بمعرفة نوع البيانات التي سيتعامل معها ونستطيع كتابة المتغيرات في أي منطقة في الكود عند الحاجة إلى ذلك بشرط واحد وهو تعريف هذا المتغير قبل استخدامه والمتغيرات لها أنواع عديدة مثل الأعداد والتواريخ والنصوص ولا بد من تحديد نوعية المتغير لكي نستطيع أن نحجز له مكان في الذاكرة فسعة التخزين لكل متغير مختلفة عن الآخر وبعد تعريف المتغير يمكنك إسناد البيانات إليه حسب نوعه وذلك بإضافة العلامة ( = ) بعد اسم المتغير

ويجب ملاحظة أنه عند تحديد أسماء للمتغيرات لابد من التنبيه لبعض النقاط الهامة من أجل سهوله التعامل مع المتغيرات في التطبيقات العملاقة والتي تحوي العديد من المتغيرات

1. يجب أن يبدأ اسم المتغير بحرف أو ( \_ ) علامة سطريه لان المتغيرات في **فيجوال بيسك 2008** تتكون من حروف وعلامات سطريه وأرقام فقط
2. الأفضل أن تكون المتغيرات قصيرة ومفهومة ويفضل أن لا يتجاوز عدد الأحرف فيها عن 33 حرف
3. لابد أن تكون أسماء المتغيرات معبرة عن استخداماتها وان لزم ذلك دمج كلمتين أو أكثر
4. استخدم خليطاً من الحروف والأرقام والعلامات السطريه ( \_ ) في تعريفك للمتغيرات ويفضل جعل الحرف الأول **Capital** والبقية **small**
5. لا تستخدم الكلمات المحجوزة في فيجوال بيسك مثل ( **Dim, If** ) أو أسماء الخصائص أو أسماء الكائنات وإلا سيقابلك خطأ ما وقت تشغيل البرنامج

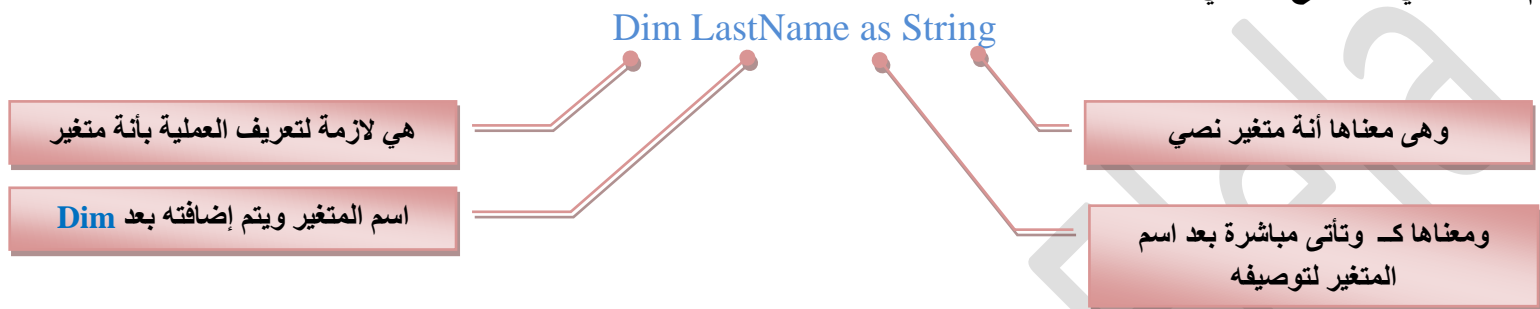
6. لجودة أكثر في برنامجك يفضل بداية اسم كل متغير بثلاثة حروف تعبر عن نوعية بيانات المتغير فمثلا يمكنك تعريف متغير بالاسم مثل **strName**
7. تستطيع تسمية المتغير باستخدام حروف اللغة العربية ولكن لعدم معرفة مضاعفات مثل هذه العملية في المستقبل فيفضل أن يكون المتغير باللغة الانجليزية فقط

ومثال على ذلك

أولا نقوم بتعريف متغير وليكن متغير نصي و أسمة

## LastName

ثانيا يتم تعريفه في البرنامج كالتالي



وبعد تعريف المتغير كما سبق يمكننا إدراج البيانات له حسب رغبتك وطبعاً بناء على نوع المتغير فالمتغير السابق هو نوعه نصي وتكون بياناته ( كلمات و جمل مثل الأسماء والأماكن والرموز الخاصة و **الأرقام** أو اي بيانات نصية ) وطبعاً يتم إضافة البيانات بعد كتابة العلامة (=) بعد أسمة ويتم أصافة البيانات بيمين العلامتين " " على إن يكون الكود كالتالي

**LastName = " البيانات النصية المدرجة "**

سوف يتعامل معها على أنها  
نصوص لأننا قد قمنا بتوصيف  
المتغير من البداية على أنه  
متغير نصي **string**

لقد اتفقنا إن هذا النوع من  
المتغيرات هو متغير نصي  
فماذا لو قمنا بإضافة أرقام  
كنوع بيانات مدرجة فيه ؟

ثالثاً يتم إسناده إلى اي كائن موجود بالنموذج نريده أن يوصف بهذا المتغير مثلاً

**Label1.Text = LastName**

هنا تم إسناده إلى الكائن **Label1** لكي يظهر المتغير **LastName** في الخاصية **Text** الخاصة بـ **Label1**

ويتم تجميع كل ما سبق في الكود التالي وكتابته مباشرة في الـ **Form** في حالة **Form1\_Load** وذلك بعد إضافة **label** فقط إلى النموذج

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Dim LastName As String
        LastName = " بسم الله الرحمن الرحيم "
        Label1.Text = LastName
    End Sub
End Class
```

تحميل التمرين الثاني عشر

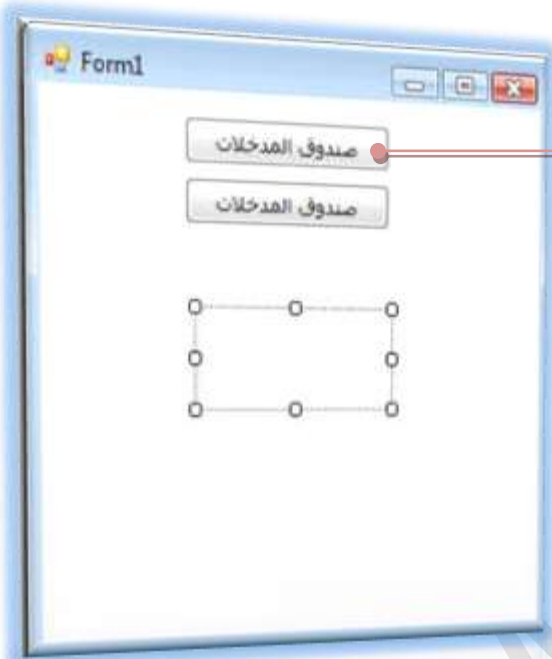
في التمرين السابق قد قمنا بتعريف المتغير **Lastname** وذلك باستخدام المتغير **Dim** ولكننا نستطيع تعريف أكثر من متغير في سطر كودي واحد وهذه ميزة ليست موجودة من قبل في الفيچوال بيسك بهذه السهولة فيمكننا إضافة مثلاً أكثر من **200** متغير على إن يكون الكود مثلاً

```
Dim LastName, Prompt, Fullnsme As String
```

( هنا أضفت ثلاثة متغيرات فقط بشرط أن يتم إسناد كل منهم إلى كائن موجود بالنموذج أو تعريفه وسوف نتعرف أكثر على هذه المتغيرات من خلال التمارين فيما بعد )

### استخدام المتغيرات لحفظ المدخلات

معظم الأمثلة التي تعاملنا معها سابقاً كانت لحفظ المدخلات في صندوق نص **Textbox** وذلك من خلال الخاصية **Text** لكن في بعض الأحيان نريد أن نحفظ المدخلات في مكان آخر وليس في خاصية من خواص الكائن المدرج بالنموذج كما سبق وتعلمنا ولذلك نستخدم المتغيرات **variables** ويعتبر صندوق المدخلات **InputBox** هو إحدى الطرق المستخدمة لجلب المدخلات من المستخدم ولذلك سنقوم من خلاله بإظهار صندوق المدخلات للمستخدم ثم حفظ النص الذي يدخله المستخدم في متغير ومثال على ذلك التمرين التالي والمطلوب فيه



عمل نموذج كما بالشكل التالي عند الضغط على زر صندوق المدخلات نفتح لنا نافذة صندوق المدخلات والتي من خلالها نتمكن من إضافة البيانات إلى النموذج في الكائن الذي تم إسناده إليه



ويكون عمل هذا التمرين بإدراج كل من

1. عدد 2 Button
2. عدد 1 Label

نلاحظ هنا أني قد قمت بعمل الأزرار بنفس الاسم وهذا لهدف سوف ندرکه فيما بعد

بعد عمل الإعدادات الخاصة بالـ **Label** وهو الكائن المدرج بالنموذج والذي سوف تنسب إليه البيانات المدخلة من خلال صندوق المدخلات تأتي مرحلة كتابة الأكواد في كل من الزر الأول ( صندوق المدخلات ) وتكون كتابة الكود كالتالي

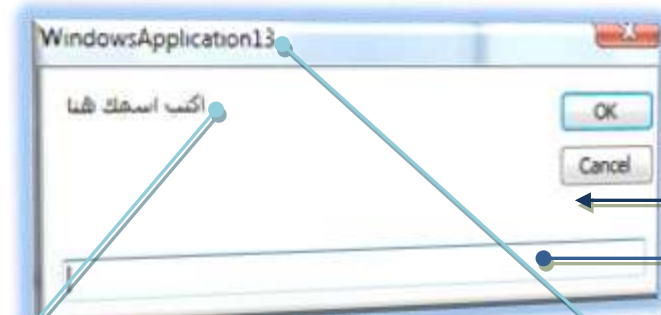
```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim Prompt, FullName As String
    Prompt = "اكتب اسمك هنا"
    FullName = InputBox(Prompt)
    Label1.Text = FullName
End Sub
```

تم إضافة متغيرين وهما **Prompt , FullName** وتعريفهم أنهم **string** أي متغيرات نصية

تم تعريف المتغير **Prompt** بأنة نص ويظهر بالشكل التالي

تم تعريف المتغير **FullName** بأنة صندوق مدخلات وبة الكلمة المسنودة إلى المتغير **Prompt**

هنا تم إسناد النص الذي يتم إدخاله في صندوق المدخلات من خلال المتغير **FullName** إلى الكائن **label** بالنموذج واطهر القيمة بة



مما سبق وبالنقر على الزر الأول ( صندوق المدخلات ) عند تشغيل النموذج F5 يتم إظهار صندوق المدخلات التالي مباشرة وذلك ليتم من خلاله إدخال البيانات إلى النموذج وإظهارها في Label1

قم بإدخال الاسم مثلا وانقر ok ليظهر الاسم تلقائيا بالنموذج كالتالي



هنا يظهر النص الذي تم إسناده إلى المتغير  
**Prompt**

اسم المشروع القائمين عليه

نلاحظ انه يوجد زر ( صندوق المدخلات ) أخر لم نقم بتوظيفه  
وكتابة الكود به وهو يتم كتابة الكود به كما بالشكل التالي

شكل الكود الأساسي هو

**FullName = InputBox(Prompt, Title)**

```
Private Sub Button2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
Dim Prompt, FullName As String
Prompt = "اكتب اسمك هنا"
FullName = InputBox(Prompt, "بيانات شخصية")
Label1.Text = FullName
End Sub
```

تم إضافة متغيرين وهما  
**Prompt, FullName**  
وتعريفهم أنهم **string** أي  
متغيرات نصية

تم تعريف المتغير **Prompt**  
بأنة نص ويظهر بالشكل التالي

تم تعريف المتغير **FullName** بأنة صندوق المدخلات  
وبه الكلمة المسنودة إلى المتغير **Prompt** ونص  
مرفق ك **Title** لصندوق المدخلات

هنا تم إسناد النص الذي يتم إدخاله في صندوق المدخلات من  
خلال المتغير **FullName** إلى الكائن **label** بالنموذج  
وأظهر القيمة به



نفس النتيجة السابقة ولكن هنا يظهر عنوان لصندوق  
المدخلات وهو كما وضحنا عند كتابة الكود

تحميل التمرين الثالث عشر

**صندوق الرسائل MsgBox** مما سبق ندرک أنه يمكننا التلاعب بصندوق الرسائل **Msgbox** ليتم إظهاره كما نريد في النموذج وإعداده بالطريقة المثلى لمشروعنا وسوف نقوم بتعلم بعض من هذه الإعدادات لصندوق الرسائل معا على إن نتفق أن هناك قاعدة رئيسية لكتابة الكود عند إظهار صندوق الرسائل وهي كالتالي

**MsgBox(Prompt,MsgBoxStyle,Title)**

**MsgBox** ☒

وهو كود استدعاء صندوق الرسائل في فيجوال بيسك 2008

**Prompt** ☒

وهي الرسالة المكتوبة كنص **String** داخل صندوق الرسائل والتي توجد دائما بين علامتين " "

**MsgBoxStyle** ☒

وهو خاص بشكل صندوق الرسائل والغرض منه وهناك انواع كثيرة من صناديق الرسائل تفي الغرض منها ومن هذه الانواع

- ❖ **OkOnly**
- ❖ **OkCancel**
- ❖ **RetryCancel**
- ❖ **Question**
- ❖ **YesNoCancel**
- ❖ **MsgBoxHelp**

وتحتوى صناديق الرسائل هذه على ازرار ويتم تخصيصها تبعا لحاجتها في المشروع

**Title** ☒

وهي الرسالة المكتوبة كنص **String** كعنوان لصندوق الرسائل وتوجد دائما بين علامتين " "

وسنأخذ مثال على ذلك وهو صندوق الرسالة التالي والخاص بـ **OkOnly** ليكون كتابة الكود لة كالتالي

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
MsgBox("كتاب فيجوال بيسك 2008", MsgBoxStyle.OkOnly, "إعداد مهندس محمد ابو العلا")
End Sub
```

**MsgBox**

**Prompt**

**Title**

**MsgBoxStyle**



وعند التشغيل **F5** وبالنقر على الزر تظهر لنا الرسالة التالية راجع التمارين الأخرى ولاحظ الفرق بينهم

تحميل التمرين الرابع عشر

وبناء على ما سبق سوف نقوم بعمل تمرين آخر ويعتبر تمرين جامع بين التمرين الثالث عشر والرابع عشر حيث إننا سوف نغير في شكل المشروع بإظهار صندوق رسالة آخر فيه وذلك بسطر كودي واحد فقط دعونا نرى ماذا يحدث لو أضفنا سطر الكود التالي إلى التمرين الثالث عشر كالتالي

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Dim Prompt, FullName As String
    Prompt = "اكتب اسمك هنا"
    FullName = InputBox(Prompt, "بيانات شخصية")
    MsgBox(FullName, , "أضافة البيانات الى صندوق المدخلات")
    Label1.Text = FullName
End Sub
```

الكود الذي تم إضافته إلى التمرين السابق



وعند الضغط على زر المدخلات ويظهر لنا صندوق المدخلات وبالنقر على زر **ok** للإدخال يتم ظهور رسالة أخرى تأكيدية بإضافة التغيير إلى صندوق المدخلات حسب **الدالة** المضافة إلى المشروع ويكون شكله كالتالي

لاحظ إن **الدالة** الذي تم إضافتها هي لصندوق الرسائل و أصل **الدالة** له كما سبق وذكرنا كالتالي

**MsgBox(Prompt,MsgBoxStyle,Title)**  
 ( "أضافة البيانات الى صندوق المدخلات" , , FullName )

نلاحظ أننا تركنا مكان **MsgBoxStyle** فارغ وبالتالي ظهر في صندوق الرسالة الزر **ok** ويمكننا التحكم فيه كما سبق وشرحنا ومن هنا تعرفنا على مفهوم جديد لسطر الكود وهو

#### الدالة

والدالة هي مجموعة من الجمل البرمجية التي تقوم بعمل برمجي محدد و منظم وله معنى وترجع نتيجة هذا العمل إلى البرنامج ويمكن إسنادها إلى متغير معين أو يمكن إسنادها إلى دالة أخرى أو خاصية معينة وهذا ما لاحظناه من قبل في كل التمارين السابقة إننا نقوم بكتابة الكود على هيئة مجموعة من الجمل يفهمها البرنامج

تحميل التمرين الخامس عشر

## البرمجية التراكيب Structures

توجد أنواع أخرى للمتغيرات و تسمى تراكيب Structures يتم تعريفها من قبل المبرمج نفسه وتستخدم هذه الطريقة إذا كان لديك مجموعة من البيانات المترابطة فيما بينها ولكن كل نوع من أنواع هذه البيانات مختلف عن الآخر ولكن مع وجود رابط بين البيانات جميعها

فمثلا إذا كان لدينا بيانات طلاب بمدرسة مثل ( الاسم - تاريخ الميلاد - تاريخ القيد بالمدرسة ) وتستخدم هذه البيانات أكثر من مرة فيمكننا تعريف هذه البيانات بشكل جماعي عن طريق التراكيب Structures ولكي نفهم طريقة تعريفهم علينا أن نعرف التالي

الاسم المختار له من طرف المبرمج	نوع المتغير	المتغير
StuName	String	( نص ) الاسم
StuBirthDate	Date	( تاريخ ) تاريخ الميلاد
StuFileDate	Date	( تاريخ ) تاريخ القيد

ويتم كتابة تعريفات Structures للمتغيرات السابقة كالتالي

```
Dim StuName as String
Dim StuBirthDate as Date
Dim StuFileDate as Date
```



تعريف اسم الطالب  
تعريف تاريخ ميلاد الطالب  
تعريف تاريخ القيد بالمدرسة

لاحظ أن لدينا مجموعة من البيانات ( أكثر من نوع من أنواع البيانات ) ولكن هناك رابط ما بين هذه البيانات وهي إن لها علاقة بالطلاب لذلك استخدمنا التراكيب Structures ويكون طريقة كتابة الكود كالتالي

```
Public Class Form1
    Structure Student
        Dim StuName As String
        Dim StuBirthDate As Date
        Dim StuFileDate As Date
    End Structure
End Class
```

لاحظ مكان كتابة الكود في منطقة  
**public Class**  
والخاصة بالنموذج

ثم إذا اردنا تعريف اسم الطالب جديد ضمن الكود بعد تعريف التركيب أعلاه نكتب الكود كالتالي

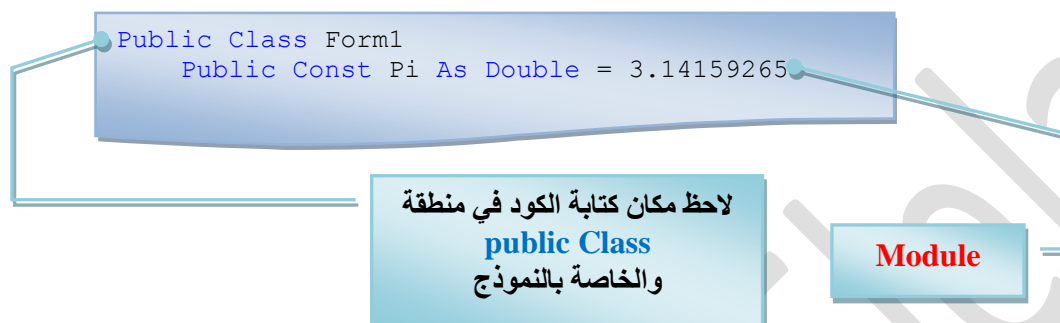
```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim ProductManager As Student
    ProductManager.StuName = "mohamed abou elela"
End Sub
```

هنا يتم إضافة طالب جديد إلى Structures يتم إضافته في النموذج  
ويتم تعريف نوع البيانات المدرجة له StuName أي أنها اسم الطالب الجديد



## الثوابت Constants

ويمكن تعريف الثوابت بأنها المتغيرات التي لا تتغير قيمتها. فمثلاً قيمة المتغير **Pi** ( $\pi$ ) أو ما يسمى بـ "باي" فيعتبر من الثوابت لأن قيمته معروفة ومعلومة وهي **3.14159265** فبدل من حفظه كمتغير في البرنامج يمكن حفظه كثابت **Constant** (أي قيمة لا تتغير) وتستخدم الثوابت بشكل كبير في العمليات الحسابية وحل المعادلات الرياضية ويتم كتابة الكود لها كما بالشكل التالي



إذا أردنا استخدام الثوابت **Constants** على طول الفورم فنقوم بتعريف الثابت في أعلى منطقة الفورم التابعة للفورم أما إذا أردنا استخدام الثابت في حاله واحده فقط فنقوم بتعريف الثابت في داخل الإجراء الذي نريد أن نستخدم فيه الثابت وفي حالة كان لدينا أكثر من فورم ونريد استخدام الثابت في كل فورم نقوم بتعريف الثابت في قالب برمجي يسمى **Module** مسبقاً بالكلمة **Public** كما بالشكل

## المعاملات الرياضية Operators

ونقصد بها العلامات الرياضية والتي يمكن استخدامها داخل الفيچوال بيسك 2008

المعامل	الوصف	مثال
+	الجمع	2+3 = 5
-	الطرح	5-2= 3
*	الضرب	2*3 = 6
/	القسمة مع إظهار الكسور	3/2 = 1.5
\	القسمة بدون إظهار الكسور	3/2 = 1
^	الأس	3^2 = 9
&	لدمج أكثر من كلمة معا	3&2 = 32 5 &4 = 54
Mod	باقي ناتج القسمة	18 Mod 4 = 2 9 Mod 3 = 0 16 Mod 5 = 1

سوف نقوم الآن بعمل تمرين تطبيقي على كيفية استخدام المعاملات الرياضية التقليدية والمتقدمة كما ذكرناها من قبل داخل بيئة الفيجوال بيسك 2008 وسوف نقوم بعمل تمرين كما بالشكل وهو يتكون من

1. عدد 2 **GroupBox**
2. عدد 3 **Textbox**
3. عدد 4 **Label**
4. عدد 8 **Radio Button**

ويتم ترتيبهم وتنسيقهم من خلال صندوق الخصائص الخاص لكل منهما حسب الشكل المطلوب أو حسب رغبتك من حيث تصميمك وقبل كتابة الكود اسأل نفسك هذا السؤال

ماذا تريد أن تفعل الآن؟

عند كتابة اي قيمة في المتغيرين الأول والثاني واختيار العملية الحسابية يتم إظهار الناتج في حقل النتيجة

نلاحظ هنا أن لدينا عدد اثنين متغيرين وهما الذي يجري عليهم العمليات الحسابية وليكن اسمهم المتغيرين  $(X, Y)$  وهما مشتركين في جميع العمليات الحسابية التي تتم على هذا النموذج ولذلك أرى ومن الطبيعي أن يتم تعريف المتغيرين  $(X, Y)$  كمتغيرات من النوع الحسابي **Double** لكي نستطيع تطبيق العمليات الحسابية عليهم ويكون الكود لهما كالتالي

**Dim x, y As Double**

وهناك طريقتين لتعريف المتغيرين بالنموذج وهما كتابة الكود السابق قبل كل عملية كالتالي

```
Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton3.CheckedChanged
    Dim x, y As Double
    x = TextBox1.Text
    y = TextBox2.Text
    TextBox3.Text = x * y
End Sub
```

1

كتابة الكود بهذه الطريقة في كل عملية من العمليات الحسابية

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

2

يتم كتابة الكود في منطقة **Public Class** مرة واحدة فقط ويكون كود العملية الحسابية كالتالي

```
Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton3.CheckedChanged
    x = TextBox1.Text
    y = TextBox2.Text
    TextBox3.Text = x * y
End Sub
```

2

العملية الحسابية هنا الضرب \* نلاحظ تغيير المعامل للعملية الحسابية لكل عملية على حدة كما في الجدول السابق

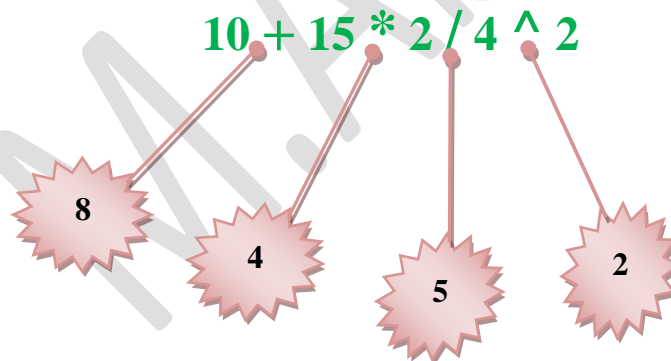
تحميل التمرين السادس عشر

## ترتيب العمليات الحسابية في بيئة التطوير

عند استخدام المعادلات الرياضية فمن المنطقي أنها تحتوي على معاملات حسابية (  $^$ ,  $*$ ,  $/$ ,  $-$ ,  $+$  ) وفي الفيچوال بيسك 2008 يجب أن ندرك تماماً كيفية تعامل بيئة التطوير مع تسلسل العمليات الحسابية في هذه المعادلات الرياضية ولمعرفة الخطوات التي تتم يقوم بها الفيچوال بيسك لمعرفة النتائج لابد أن ندرك تسلسل الخطوات الحسابية حسب الجدول التالي

الشرح	المعامل	ترتيب العملية
يقوم البرنامج بحساب ما بين الأقواس	( )	1
يقوم البرنامج بحساب الأس	^	2
يقوم البرنامج بحساب الأرقام السالبة	-	3
يقوم البرنامج بحساب الضرب	*	4
يقوم البرنامج بحساب القسمة	/	5
يقوم البرنامج بعملية القسمة بدون احتساب كسور في النتيجة	\	6
يقوم البرنامج بحساب باقي القسمة	Mod	7
يقوم البرنامج بحساب عملية الجمع	+	8
يقوم البرنامج بحساب عملية الطرح	-	9

ولتبسيط الجدول السابق دعنا نتعرف على ناتج العملية الحسابية التالية



بعد أن قمنا بتحديد ترتيب كل عملية في المعادلة السابقة من الجدول السابق نجد أن تسلسل العمليات على المعاملات الموجودة بالتمرين هي ( عملية الأس ) ثم ( عملية الضرب ) ثم ( عملية القسمة ) ثم ( عملية الجمع ) فيكون حل التمرين كالتالي

$$10 + 15 * 2 / 4 ^ 2$$

1.  $10+15*2/16$  ( عملية الأس )
2.  $10+30/16$  ( عملية الضرب )
3.  $10+1.875$  ( عملية القسمة )
4.  $11.845$  ( عملية الجمع )

جرب نفس التسلسل في الجدول السابق على نفس التمرين في الحالة التالية

$$(10 + 15) * 2 / 4 ^ 2$$

1.  $30*2/4^2$  (حساب ما بين الأقواس)
2.  $30*2/16$  ( عملية الأس )
3.  $60/16$  ( عملية الضرب )
4.  $3.75$  ( عملية القسمة )

النتائج في الحالة الأولى (11.845) والنتائج في الحالة الثانية (3.75) وذلك راجع لاختلاف تسلسل المعاملات الحسابية رغم أنها نفس الإجراءات الحسابية لكن ترتيب تسلسلها يختلف

التعامل مع الطرق **Methods** ضمن بيئة التطوير

من خلال العمل في بيئة التطوير قد تحتاج العديد من الأوامر لصنع معادلات مختلفة أو لتصميم برنامج يحسب مجموعة من القيم أو المتغيرات فالطرق المقصودة هنا هي تلك المعادلات المخزونة مسبقاً ضمن بيئة التطوير ويمكن الحصول على هذه الطرق أو الدوال من الكلاس **System.Math** وهناك طرق أو دوال عديدة تابعة للكلاس **System.Math** ويمكن استخدام أيًا من هذه الطرق ولاكن لابد من استيراد الكلاس **System.Math** أولاً ضمن مجالات الأسماء في أعلى منطقة الكود قبل كتابة أي كود آخر حتى قبل كتابة ( **Public Class Form1** ) في صفحة كتابة الكود على ان يكون طريقة استيراد الكلاس **System.Math** عن طريق كتابة الكود التالي

**Imports System.Math**

ويكون شكلها في منطقة كتابة الكود كالتالي

```
Imports System.Math
    ' يتم استيراد الكلاس اولا بكتابة الكود السابق '

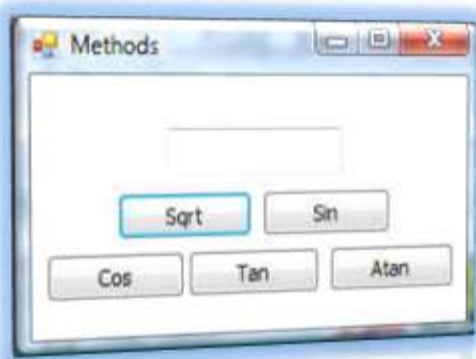
Public Class Form1
    ' بعد ذلك يتم كتابة باقى الكود للبرنامج وبعد '
```

وهذه بعض الدوال أو الطرق المتوفرة ضمن الكلاس **System.Math**

**n** ترمز إلى المتغير ويمكن تغييرها باى رقم داخل المعادلة

الغرض منها	الطريقة
ترجع لنا قيمة <b>n</b> بالموجب لو كانت قيمتها بالسالب وعدم تغييرها لو كانت بالموجب	<b>Abs(n)</b>
معرفة ظل الزاوية <b>n</b> بوحدة قياس الراديان	<b>Atan(n)</b>
معرفة جتا الزاوية <b>n</b> بوحدة قياس الراديان	<b>Cos(n)</b>
معرفة جيب الزاوية <b>n</b> بوحدة قياس الراديان	<b>Sin(n)</b>
معرفة ظل الزاوية <b>n</b> بوحدة قياس الراديان	<b>Tan(n)</b>
لمعرفة الجذر التربيعى لـ <b>n</b>	<b>Sqrt(n)</b>
ترجع لنا القيمة -1 لو كانت <b>n</b> اصغر من الصفر ترجع لنا القيمة 0 لو كانت <b>n</b> يساوى الصفر ترجع لنا القيمة +1 اذا كانت <b>n</b> اكبر من الصفر	<b>Sign(n)</b>

وسوف نقوم الآن بعمل تمرين بسيط لحساب الجذر التربيعى لرقم ما **n** وذلك باستخدام طرق **System.Math**



```
Imports System.Math
Public Class Form1
```

والمطلوب هو عمل برنامج لحساب بعض العمليات الحسابية لرقم معين ويتم ذلك من خلال عمل النموذج التالي وإدراج بة كل من

1. عدد 1 Textbox لإدخال الرقم الذي سوف تجرى عليه العملية الحسابية **n**
2. عدد 5 Button لإجراء العمليات الحسابية من خلالهم تبعا للكود بهم

بعد إعداد النموذج كما بالشكل من خلال تغيير خواصه وخواص كل من الكائنات المدرجة بة حسب الشكل أو التنسيق الذي تراه مناسباً نأتي إلى مرحلة كتابة الكود وتكون كالتالي

نحن نعلم إننا في هذا التمرين سوف نتعامل مع الطرق **Methods** ولذلك لابد من استيراد الكلاس **System.Math** وذلك عن طريق كتابة الكود كما ذكرنا من قبل وطبعاً يتم كتابته ضمن مجالات الأسماء في أعلى منطقة الكود قبل كتابة أي كود آخر

بدون هذا الكود لن نستطيع استخدام أي من الطرق السابق ذكرها

ثم يتم كتابة الأكواد في الأزرار المنفذة له كل كود على حسب نوع الزر فمثلاً زر **Sqrt** وهو زر الحصول على الجذر التربيعي للقيمة المدخلة في **Textbox1** ويكون الكود فيها كالتالي

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    MsgBox(Sqrt(TextBox1.Text), , "الجذر التربيعي")
End Sub
```

صندوق الرسالة التي تظهر عند الضغط على الزر **Sqrt**

هنا تظهر الـ **Title** وهو النص الذي يظهر كعنوان لصندوق الرسائل

هنا تظهر الـ **prompt** والتي تم إسنادها إلى إظهار الجذر التربيعي **Sqrt** للقيمة المدخلة في **TextBox1**

مكان الأزرار التي تظهر في صندوق الرسالة كما تعلمنا من قبل

وبتنفيذ نفس الخطوات السابقة على كل من الأزرار (**Sin** و **Cos** و **Tan** و **Atan**) وذلك بكتابة نفس الكود مع تغيير الطريقة **Sqrt** إلى الطريقة المتبعة بكل زر مما سبق حسب وظيفته انظر التمرين

حاول تجربة الخطوات السابقة بدون كتابة كود استيراد الكلاس **System.Math**

والموجود ضمن مجالات الأسماء في أعلى منطقة الكود

```
Imports System.Math
```

ولاحظ الفرق

تحميل التمرين السابع عشر

## البرمجة بالأحداث Event-Driven Programming

البرمجة بالأحداث هي البرمجة التي تعتمد بشكل أساسي على الأحداث **Events** التابعة للمكونات المتوفرة في بيئة التطوير **Objects** فبدلاً من كتابة كود من صفحتين أو ثلاث لمراقبة المستخدم حتى يضغط على الزر **Button** فقط نكتب الأمر الذي نريد تنفيذه في الحدث **Click** التابع للزر **Button** فالبرمجة بالأحداث توفر علينا الكثير من الجهد والوقت بسبب وجود العديد من الأحداث المخزونة مسبقاً في بيئة الفيجوال بيسك 2008 وقد استخدمنا في التمارين السابقة حدث واحد فقط من هذه الأحداث وهو يكتب تلقائياً عند النقر المزدوج لأي كائن مدرج بالنموذج ويكتب كود الحدث له تلقائياً كما يلي

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

المكون Button1

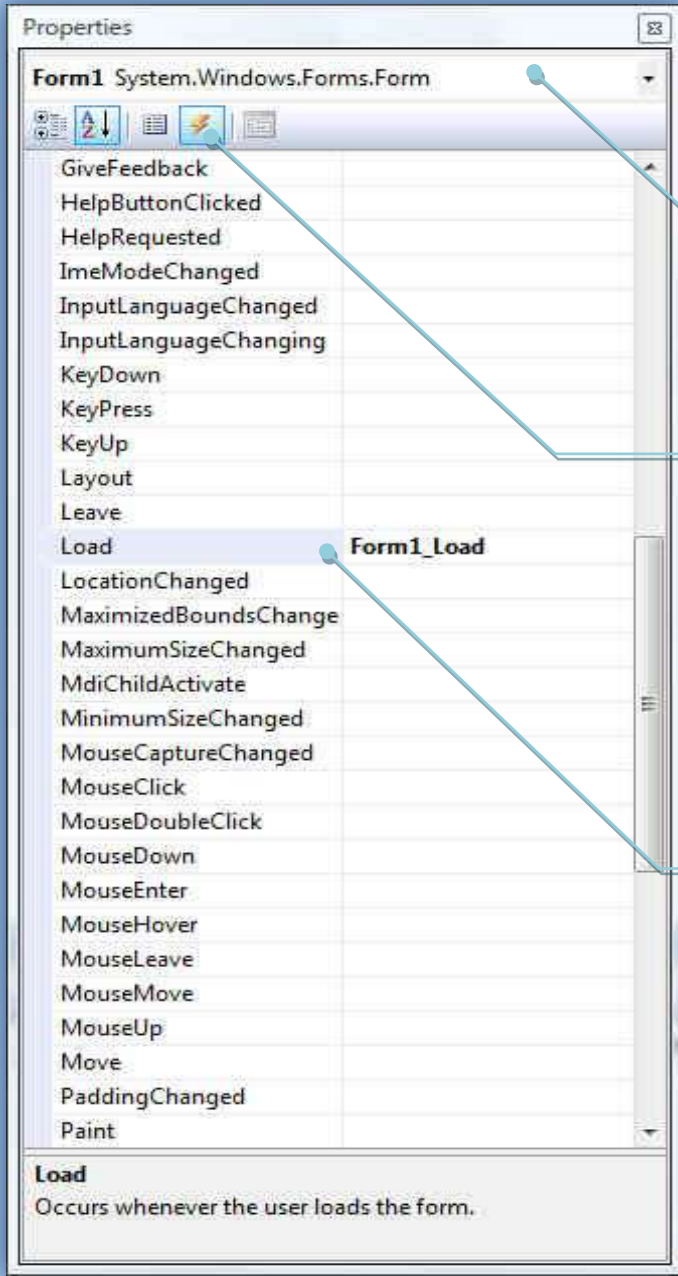
الحدث Click

طبعاً الكود السابق قد مر علينا كثيراً فيما سبق من خلال التمارين التي قمنا بتنفيذها معاً وهناك العديد من الأحداث المدعومة من الفيجوال بيسك 2008 والخاصة لكل مكون من مكونات بيئة التطوير **Objects** ويمكن معرفة هذه الأحداث عن طريق إدراج كائن إلى النموذج ثم فتح صفحة كتابة الأكواد واختيار المكون تظهر لنا كل الأحداث التي يمكننا استخدامها في التعامل مع

1 عند فتح القائمة **General** تظهر لنا كل الكائنات المدرجة بالنموذج للتعامل معها مهما كان عددها وباختيار أي كائن منهم

2 يتم إظهار الحدث المتعامل معه تلقائياً في القائمة **Declarations**

3 وفتح القائمة **Declarations** تظهر لنا جميع الأحداث التي يمكن استخدامها مع هذا الكائن وهناك الكثير منها



وهناك طريقة اخرى للتعامل مع الاحداث وتغيرها وذلك من خلال صندوق الخصائص الخاص بكل كائن مدرج بالنموذج كالتالي

الكائن المدرج بالنموذج والمتعامل مع خصائصه

من هنا يتم ظهور الأحداث التي يمكن التعامل بها مع الكائن

الحدث المختار والذي يتم حدوثه عند التعامل مع الكائن وهنا الحدث هو **Load** أي أنه سوف يتم تنفيذ الأكواد الموجودة في هذا الكائن بمجرد تشغيله **F5**

## جمل القرارات Decision Structures

### الجمل الشرطية If .....Then

كثيرا ما نحتاج في البرمجة إلى جمل فيها قرارات معينة فمثلا يمكننا كتابة جملة البرمجة باستخدام أداة **الشرط IF** في البرمجة فنقول فيها إذا كان **A** أكبر من **B** فننفذ الأمر التالي ونكتب أمر معين آخر وتسمى هذه الجمل بالجمل الشرطية وقد تم شرحها من قبل راجع التمارين المرفقة بالكتاب التمرين الثالث وسوف نتعرف عليها الآن بطريقة أخرى حيث سوف ندخلها في تعاملاتنا مع المعاملات الحسابية ومثال على ذلك هو الشرط التالي

### Score > 50

وهنا نجد أن هناك احتمالين إما أن يكون فعلا النتيجة أكبر من 50 وتكون الجملة الشرطية صحيحة ( محققة الشروط) أو أن يكون أقل من ذلك أو يساويه وفي هذه الحالة تكون الجملة الشرطية غير صحيحة ( غير محققة الشروط ) وفي كل من الحالتين سوف يتم توجيه البرنامج لتنفيذ أمر ما وهناك بعض المعاملات ( الرموز ) والتي يمكن معرفتها

تقرأ في سطر الأكواد من اليسار إلى اليمين

يساوى	=
أكبر من	>
أصغر من	<
أكبر من أو أصغر من ≠	<>
أصغر من أو يساوى ≥	<=
أكبر من أو يساوى ≤	>=

تحميل التمرين الثامن عشر

جملة **IF** الشرطية التي تعالج أكثر من شرط **If.....ElseIf** لاحظنا في التمرين السابق أننا استخدمنا جملة **IF** التي تعالج شرط واحد فقط وعند تحقيق هذا الشرط تظهر رسالة وفي حالة عدم تحقيقه تظهر رسالة بذلك ولكن هناك بعض النتائج التي يمكن أن يكون لها أكثر من ناتج مثلاً فيجب علينا في هذه الحالة استخدام الدالة **IF** التي تعالج أكثر من شرط في هذه الحالة وتخصيص كل شرط من شروطها بقيمة ناتج جديد و مثال على ذلك هو التمرين التالي



والمطلوب هو عمل نموذج كما بالشكل والهدف منه هو عند إدخال قيمة الناتج تظهر لنا رسالة تعرفنا حالة الطالب من نجاح أو رسوب حسب المجموع أو النتيجة التي تم إدخالها إلى النموذج المكون من

1. عدد **Label 1** وهو لكتابة النص الظاهر في النموذج بة
2. عدد **Textbox 1** وهو يتم استقبال المدخلات من المستخدم وهو درجة المجموع أو النتيجة
3. عدد **Button 1** وهو زر تنفيذ الشرط ويتم كتابة الكود بة كالتالي



تظهر هذه الرسائل عند إدخال القيم في **Textbox** على أن يتم اظهر كل رسالة " **ناجح** " عند تحقق الشرط وهو أن تكون القيمة المدرجة **اكبر من 50** وظهور رسالة " **راسب** " عند تحقق الشرط آخر وهو أن تكون القيمة المدرجة **اقل من 50** وفي حالة **عدم إدخال اى قيمة** في **Textbox** تظهر رسالة " **تأكد من المدخلات** "

ويكون كتابة الكود في الزر **Button** عند الحدث **Click** كالتالي

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    If TextBox1.Text = "" Then
        MsgBox("تأكد من المدخلات", MsgBoxStyle.Information, "النتيجة")
    ElseIf TextBox1.Text >= 50 Then
        MsgBox("ناجح", , "النتيجة")
    ElseIf TextBox1.Text < 50 Then
        MsgBox("راسب", , "النتيجة")
    End If
End Sub
```

لاحظ هنا أنى قد تلاعبت في كتابة الكود لصندوق الرسائل حتى تظهر لنا صورة **Information** عند ظهور الرسالة كما تعلمنا من قبل خلال برمجة صندوق الرسائل **Msgbox**

تحميل التمرين التاسع عشر



استخدام المعاملات المنطقية في الجمل الشرطية خلال كتابة الجمل الشرطية قد نحتاج إلى إضافة بعض الشروط في الجملة الواحدة أو قد نحتاج إلى تعقيد الشرط البرمجي ( وهو وضع أكثر من شرط لتحقيق الهدف ) و في هذه الحالة لابد من استخدام المعاملات المنطقية في الجمل الشرطية ومن المعاملات المستخدمة مع الجمل الشرطية

الهدف منه	معناه	المعامل المنطقي
يستخدم لإضافة شرط جديد إلى الشرط الأول في الجملة الشرطية و لابد من تحقيقهما معا لتحقيق الجملة الشرطية If	و	And
يستخدم لإضافة شرط بديل للشرط الأول في الجملة الشرطية وفي حالة تحقق اى منهما تتحقق القاعدة	أو	Or
يستخدم في حالة عدم تحقق الشرط المكتوب فإن الجملة الشرطية صحيحة	ليس	Not
يستخدم في حالة وجود شرطين وتحقق احدهما فقط فهو كافي لجعل الجملة الشرطية صحيحة	إذا كان أحد	Xor

نلاحظ من الجدول السابق أن هناك أكثر من معامل منطقي يستخدم مع الجملة الشرطية **If** وقد مر علينا بعضها من قبل من خلال التمارين السابقة ولكن لنتأكد أكثر من فهمنا للجمل الشرطية وأهميتها في عمليات البرمجة وكتابة الأكواد سوف نقوم معا بعمل التمرين شامل يحتوى على جميع المعاملات المنطقية لإظهار الفرق بينهم وذلك وهو يتكون من التالي

من الآن سوف أتركك لكي تتعرف على محتويات النموذج من خلال صورة التمرين أو المرفق مع الكتاب فاعتقد أنك لا تحتاج إلى هذه الطريقة البدائية بعد كمية التمارين التي نفذت سابقا فقط سوف أقوم بذكر اى كائن جديد يتم استخدامه لأول مرة في النموذج

هذه الأداة اسمها MaskedTextBox وإدراجها من خلال صندوق الأدوات وهي تستخدم لإجبار المستخدم على إدخال قيمة معينة بالطريقة التي تراها مناسبة في الإدخال ويتم تسبقها بعد إدراجها بالنموذج كالتالي

من هنا يتم اختيار طريقة إدخال البيانات

من هنا يتم إدخال بعض التعديلات على طريقة الإدخال المختارة

على إن يتم كتابة الأكواد كالتالي في كل زر من الموجودين بالنموذج كل حسب حالته وهذا ليس بالجديد علينا فقط عملنا على تمارين مشابهة فيما سبق وأكثر تعقيدا

```
If MaskedTextBox1.Text = "16/12/1972" And CheckBox1.Checked Then
    MsgBox("تاريخ الميلاد صحيح", , "If And")
End If
```

هنا يتم وضع شرط اول بان التاريخ يكون 1972/12/16 وشرط ثانى وهو ان وضع علامة اختيار فى CheckBox1 وفى حالة تحقق الشرطين معا تتحقق الجملة الشرطية

```
If MaskedTextBox1.Text = "16/12/1972" Then
    MsgBox("تاريخ الميلاد صحيح", , "If")
End If
```

هنا يتم وضع الشرط بان التاريخ يكون 1972/12/16 وفى حالة تحقق الجملة الشرطية



```
If MaskedTextBox1.Text = "16/12/1972" Or MaskedTextBox1.Text = "16/12/1973" Then
    MsgBox("تاريخ الميلاد صحيح", , "If Or")
End If
```

هنا يتم وضع الشرط بان التاريخ هو 1972/12/16 او 1973 /12/16 فان كان احدهما تتحقق الجملة الشرطية

```
If MaskedTextBox1.Text = "16/12/1972" Xor MaskedTextBox1.Text = "16/12/1973" Then
    MsgBox("تاريخ الميلاد صحيح", , "If Xor")
End If
```

End If

هنا يتم وضع الشرط بان التاريخ هو يكون احدى التاريخين 1972/12/16 او 1973 /12/16 فان كان احدهما تتحقق الجملة الشرطية

تحميل التمرين العشرون

استخدام **OrElse** و **AndAlso** في الجملة الشرطية **If**

المعامل المنطقي **AndAlso** هو نفس المعامل **And** ولكن مع إضافة صغيرة جداً وهي التحقق من الشرط الأول قبل الانتقال إلى الشرط الثاني فإذا كان الشرط الأول خاطئاً لا يقوم بالانتقال إلى الشرط الثاني بل يقوم مباشرة بالإبلاغ بأن نتيجة الجملة خاطئة مما يزيد من سرعة التأكد من الجمل البرمجة الشرطية بدون التحقق من جانبي الشرط

المعامل المنطقي **OrElse** هو يقوم بالتحقق من الشرط الأول فإذا كان الشرط الأول صحيحاً لا يقوم بالانتقال إلى الشرط الثاني بل يقوم مباشرة بالإبلاغ بأن نتيجة الجملة صحيحة مما يزيد من سرعة التأكد من الجمل البرمجة الشرطية

استخدام الأداة الشرطية **Select Case**

الأداة الشرطية **Select Case** والتي تعني "اختر الحالة" إذا كان لدينا متغير واحد وله ثلاث أو أكثر من ثلاث حالات أو قيم لأنها تعطي للكواد البرمجي سهولة أكثر في القراءة والمراجعة في هذه الحالة وكما تعلمنا من قبل فإن الأداة الشرطية تبدأ بالكواد **Select Case** وتنتهي بالكواد **End Select** ويتم بينهم كتابة الحالات ويمكننا أيضاً الاستعانة بالأداة **Case Else** ومعناها "أي حالة أخرى غير مذكورة" كما يمكننا كذلك استخدام المعاملات المنطقية (**>**، **<**، **<=**، **>=**) في الجمل البرمجية الشرطية **Select Case** ولكن بعد إضافة **Is** بعد كلمة **Case**

التمرين التالي هو تمرين على الأداة الشرطية **Select Case** وهو تمرين بسيط لأننا قد أخذنا فكرة عن الأداة الشرطية **Select Case** من قبل خلال التمارين السابقة والمطلوب منك هو عمل نموذج كما بالشكل التالي والهدف منه هو معرفة نوع برجك الميلادي من تاريخ مولدك فعند اختيار تاريخ ميلادك من **ComboBox** يقوم بكتابة اسم البرج الخاص بهذا التاريخ في **TextBox** وتظهر صورة رمزية لهذا البرج في **PictureBox** (للعلم التمرين رقم 7 نفس الفكرة مع بعض الاختلافات البسيطة) وسوف تكون كتابة الكود في الزر **Button** كالتالي



عند اختيار التاريخ من **ComboBox**

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Select Case ComboBox1.SelectedIndex
        Case 0
            TextBox1.Text = "برج الحمل"
            PictureBox1.Visible = True
            PictureBox1.Image = Global.WindowsApplication1.My.Resources._1
    End Select
```

تم الإشارة إلى الحالة الأولى فقط ومسح باقي الحالات للتشابه فيما بينهم راجع الكود كاملاً في التمرين المرافق

يتم إظهار الصورة 1 في صندوق الصور **PictureBox1** والمدرجة بطريقة الـ **Resources**

يتم إظهار صندوق الصور **PictureBox1** لآنة مخفي عند التشغيل **F5**

عند اختيار الحالة الأولى **Case 0**

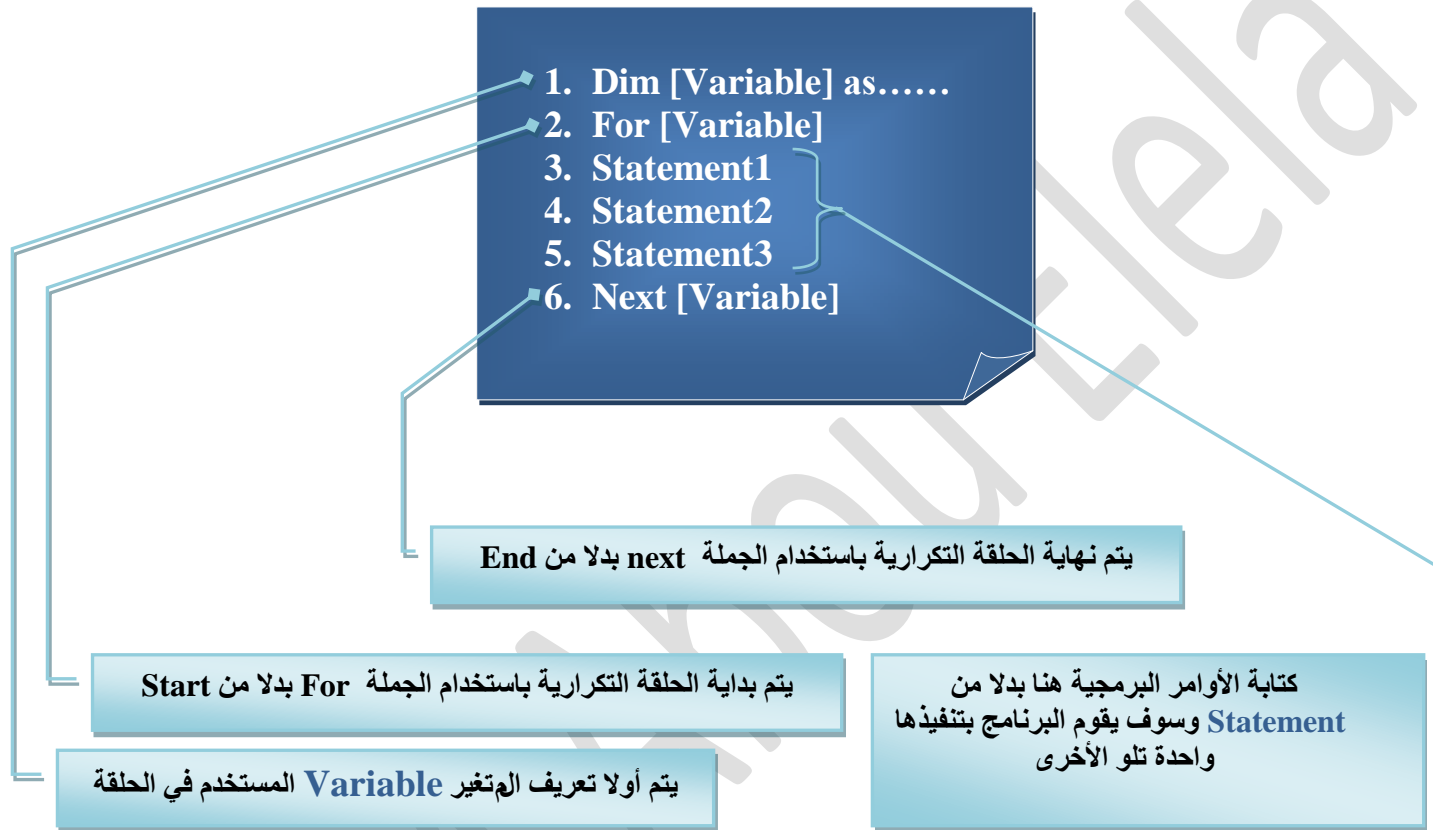
```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    PictureBox1.Visible = False
End Sub
```

إخفاء **PictureBox1** عند بداية التشغيل **F5** يتم كتابة الكود في الـ **Form1**

تحميل التمرين الحادي والعشرون

الحلقات التكرارية والمؤقتات **Loops and Timers**أولا الحلقات التكرارية **For...Next**

إن استخدام الحلقات التكرارية **For...Next** مهم في الحالات التي تلجأ فيها إلى كتابة عدد كبير من الأكواد فبدلاً من تكرار الكود عدة مرات سيتم استخدام حلقة تكرارية **For...Next** لاختصار الكود وهناك من يفضل عدم استخدام الحالة التكرارية ويقوم بنفسه بكتابة الكود أكثر من مرة وهنا أريد أن أسئلة سؤالا؟ ماذا لو تريد تكرار هذا الكود مثلاً 100 مرة هل ستقوم كتابته 100 مرة للوصول إلى هدفك والحلقات التكرارية **For...Next** ليست مجرد تكرار أرقام أو جمل فقط ولكنها تستخدم مثلاً في تكرار عرض صور معينة على الشاشة أو تكرار عرض جملة معينة مرات عديدة والشكل العام لجملة البرمجة للحلقة التكرارية **For...Next** هي



وسيوضح المثال التالي الفكرة أكثر فالتمرين العملي خير برهان على الشرح هذا وقد استخدمنا في التمرين التالي الحلقة التكرارية **For...Next** وذلك لتكرار جملة الشهادة ( لا إله إلا الله سيدنا محمد رسول الله ) وذلك 100 مرة فبعد أن نصل بالنموذج إلى هذا الشكل وتعديل خواص الـ **Textbox** من صندوق الخصائص الخاص بها لكي تستطيع أن تستوعب المحتوى وإظهار شريط التمرير الراسي تستطيع أن تراجع التغيير بمقارنة صندوق الأدوات بالتمرين المرفق ويتم كتابة الكود بحيث عند النقر على الزر **Button** تظهر في **Textbox** الشهادة وعددها 100 مرة وفي حالة التكرار تظهر 100 مرة أخرى وهكذا ليكون الشكل التالي



ويكون كتابة الكود في الزر **Button** كالتالي

2

يتم إسناد المتغير **View** بطريقة العرض إلى القيمة **Chr (9)** وهي خاصة بطريقة العرض للكلمات داخل **Textbox** ويمكننا التعديل فيها بتغيير الرقم 9 إلى رقم آخر ويمكن الدمج بين تنسيقين أو أكثر عن طريق العلامة **&** ويكون كالتالي **Chr(13) & Chr(10)** قم بتغيير الأرقام بالتمرين ولاحظ الفرق العملي

1

أولا يتم تعريف المتغيرات  
**A.** يتم تعريف المتغير **n** على أنه متغير عددي ( عدد صحيح ) يمثل عدد مرات التكرار داخل الحلقة  
**B.** يتم تعريف المتغير **View** وهو أسلوب عرض الكلمة داخل **Textbox** على أنه متغير نصي

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    Dim n As Integer
    Dim view As String
    view = Chr(9)
    For n = 1 To 100
        TextBox1.Text = TextBox1.Text & " لا اله الا الله سيدنا محمد رسول الله " & n & view
    Next n
End Sub
```

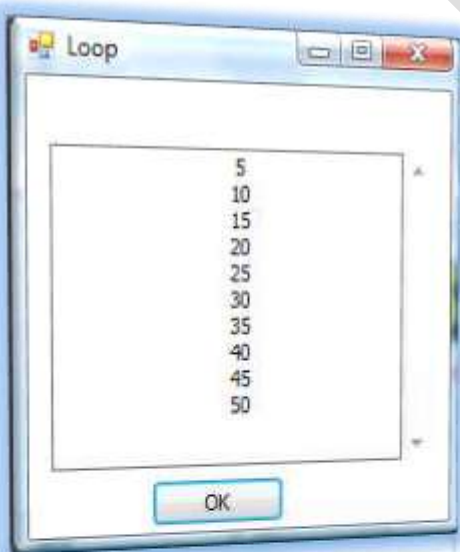
4

هنا يتم تعريف بالجملة المراد إظهارها في **Textbox** ونجد سطر الأمر مكون من ثلاث أجزاء تربطهم العلامة **&**  
**A.** الجملة النصية في **Textbox** وتكون عادة بين ""  
**B.** **n** وهو رقم تكرار الكلمة  
**C.** **View** وهو طريقة عرض الكلمة كما سبق ووضحنا

3

هنا نبدأ كتابة الكود للحلقة التكرارية **For...Next** وتكون بدايتها بكتابة **For** وهنا يتم تعريف البرنامج بعدد تكرار الجملة أو الرقم أو أي كان مدرج في الخطوة التالية له

تحميل التمرين الثاني والعشرون



و في نفس التمرين السابق يمكن تعديل الحلقة التكرارية **For...Next** في الكود الخاص بها للوصول إلى النتيجة التالية وذلك بتغيير الكود إلى التالي

تعريف **n** عدد صحيح **Integer**  
ويمكن تعريف **n** كعدد عشري **Single** في حالة استخدامه

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim n As Integer
    Dim view As String
    view = Chr(13) & Chr(10)
    For n = 5 To 50 Step 5
        TextBox1.Text = TextBox1.Text & " " & n & view
    Next n
End Sub
```

قمنا بتعديل الحلقة التكرارية لكي تقوم بتكرار الأرقام من الرقم 5 إلى الرقم 100 على أن تكون المسافة بينهم هي 5 كما تم إزالة الرسالة من **Textbox** وتديل تنسيق العرض إلى **Chr(13) & Chr(10)** لملاحظة الفرق

تحميل التمرين الثالث والعشرون

وكما ذكرنا من قبل إننا نستطيع استخدام حلقات التكرار في العديد من المهام التي نحتاجها في برامجنا فالمسألة ليست تكرار أرقام و نصوص فقط ولكن يمكننا استخدامها في تكرار عرض مجموعة صور معينة على الشاشة مهما كان عددها بدون الحاجة إلى أي طريقة سابقة من طرق إظهار الصور فسنعلم الآن إظهار مجموعة من الصور الموجودة على جهازك من خلال حلقات التكرار والتي سوف نتعلمها معا من خلال التمرين التالي وذلك باستخدام طريقتين لكتابة الكود فبعد الوصول بالنموذج إلى هذا الشكل والتنسيق وذلك بإضافة **PictureBox** وعدد 2 زر **Button** كل منهما لتنفيذ كود معين

أولا إظهار الصور من خلال صندوق الرسائل



وذلك بكتابة الكود في الزر الأول **Button1** ويكون كالتالي

أولا يتم تعريف المتغير **n** على أنه متغير عددي ( عدد صحيح ) يمثل عدد مرات التكرار داخل الحلقة وهي 9 صور

```
Dim N As Integer
For N = 1 To 9
    PictureBox1.Image = System.Drawing.Image.FromFile _
        ("C:\Users\mohamed\Desktop\WindowsApplication24\fac\FACE0" & N & ".JPG")
    MsgBox ("PRESS OK TO SHOW NEXT FACE ", , "SHOW FACES")
Next
```

هنا يتم تحديد صندوق الصور **PictureBox1** وكتابة الكود به على ان يظهر الصور الموجودة بالمسار المذكور ( يتم تعديل المسار عند تشغيل التمرين على جهازك او نسخ الملف على سطح المكتب عندك وتغيير كلمة **mohamed** باسم جهازك ) ثم نقوم بتحديد نوع امتداد الصور المطلوب عرضها في **PictureBox1** وهنا الصور من النوع **jpg**

ثم نقوم بتنسيق شكل صندوق الرسائل **Msgbox** الذي يظهر بعد كل صورة للانتقال إلى الأخرى كما تعلمنا من قبل

عند الانتهاء من تكرار العملية لعدد 9 مرات لأننا عرضنا 9 صور يقوم البرنامج بالانتهاء ولكن سوف نقوم الآن بعمل نفس التمرين بطريقة احترافية أكثر وهي إظهار الصور بدون أي رسائل وعند الانتهاء من عرض الـ 9 صور لا ينتهي البرنامج من العرض بل يكرر العرض مرات ومرات لحين إغلاقنا نحن البرنامج أو عدم الضغط على زر العرض **Button2** الذي سنقوم بكتابة الكود به بالطريقة التالية

لاحظ المسار الموجود بالتمرين وهو منقول من جهازي ولا بد من تعديل المسار إلى مكان الصور المطلوب عرضها حتى يعمل البرنامج بشكل جيد

"C:\Users\mohamed\Desktop\WindowsApplication24\fac\FACE0"



وهنا لن نحتاج إلى برمجة صندوق الرسائل فالصور المعروضة تظهر تلقائياً بمجرد النقر على الزر **START SHOW** في صندوق عرض الصور **PictureBox** وذلك يتم عن طريق

أولاً نقوم بتعريف المتغير **S** كمتغير عام على طول الفورم بمعنى إننا لا بد أن نقوم بتعريفه تحت **Public Class** مباشرة كما سبق وتعلمنا

```
Public Class Form1
    Dim S As Integer = 1
```

ثم يتم كتابة الكود في **Button2** كالتالي

مسار الصور من جهازك وقد نبهنا عنة من قبل ارجع للتوجيهات السابقة

```
PictureBox1.Image = System.Drawing.Image.FromFile _
    ("C:\Users\mohamed\Desktop\WindowsApplication24\fac\FACE0" & S & ".JPG")
S = S + 1
If S = 10 Then
    S = 1
End If
End Sub
```

1

إذا يتم عرض الصورة 1 من جديد وهكذا

2

لو أصبح الناتج 10 يعني يتم عرض الـ 9 صور لأننا بدأنا بالصورة 0 لاحظ اسم الصورة بالمسار

3

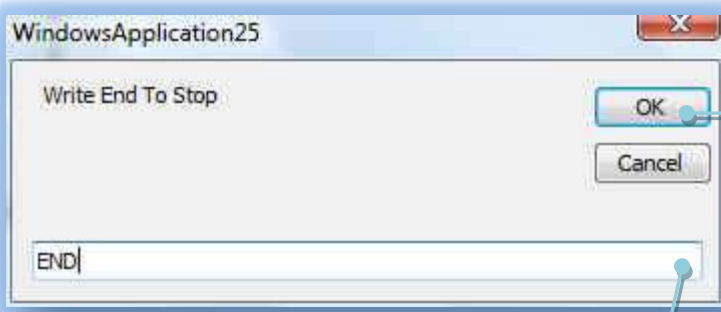
بمعنى أن يتم عرض أول صورة ويتم إضافة 1 فتعرض الثانية ثم إضافة 1 آخر فتعرض الثالثة وهكذا

تحميل التمرين الرابع والعشرون

إنهاء الحلقات التكرارية بدون تكملة التكرار

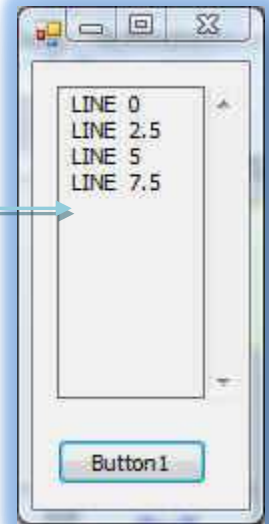


إذا أردنا إنهاء الحلقات التكرارية عند تحقق شرط معين علينا تحديد ذلك الشرط في الجمل التكرارية **For...Next** ثم كتابة **Exit For** وسيقوم البرنامج مباشرة بالتوقف عن التكرار والانتقال و تنفيذ السطر البرمجي الذي يلي الحلقة التكرارية فمثلا لو أردنا توقف البرنامج عن التكرار عند قيام المستخدم بإدخال كلمة **END** فقط في صندوق المدخلات للبرنامج وغير ذلك يقوكم باستكمال عملية التكرار لحين الانتهاء منها ومثال على ذلك التمرين التالي وهو شبيه بالتمرين الثاني والعشرون والسابق شرحه لكن فقط أضفنا عليه بعض التعديلات لكي يتناسب مع شرط البرمجة الجديد وهو عند الضغط على الزر **Button1** يقوم البرنامج بإظهار الرسالة التالية



وعندما نقوم بكتابة أي كلمة في صندوق الرسالة يقوم البرنامج بتنفيذ التكرار المطلوب مرة وهو تكرر الأرقام من الرقم 0 إلى الرقم 15 وذلك بخطوة مقدارها 2.5 وسوف يقوكم بالتكرار كلما نقرنا على **OK** أو كتابة أي نص في صندوق الرسائل مهما كان ولن يتوقف إلا في حالة كتابة الكلمة **END** فقط

كتابة **END** لوقف التكرار ولاحظ أن يتم كتابتها بالحروف  
Capital



وليكون كود البرنامج الجديد كالتالي وسوف أقوم بشرح التغييرات فقط التي أضفناها على الكود السابق وعليك الرجوع إلى التمرين الثاني والعشرون لمراجعة شرح الأكواد السابقة

1. يتم تعريف المتغير **n** على أنه متغير عددي غير صحيح (كسر) يمثل عدد مرات التكرار داخل الحلقة
2. يتم إضافة تعريف المتغير **name** وهو الخاص بطريقة عرض صندوق المدخلات **InputBox**

```
Dim n As Single
Dim view, name As String
view = Chr(13) & Chr(10)
For n = 0 To 15 Step 2.5
    name = InputBox(" Write End To Stop ")
    TextBox1.Text = TextBox1.Text & " LINE " & n & view
    If name = "END" Then Exit For
Next n
```

لو تمت كتابة الكلمة **END** يتوقف التكرار

يتم إسناد المتغير **name** إلى عرض صندوق المدخلات **InputBox**

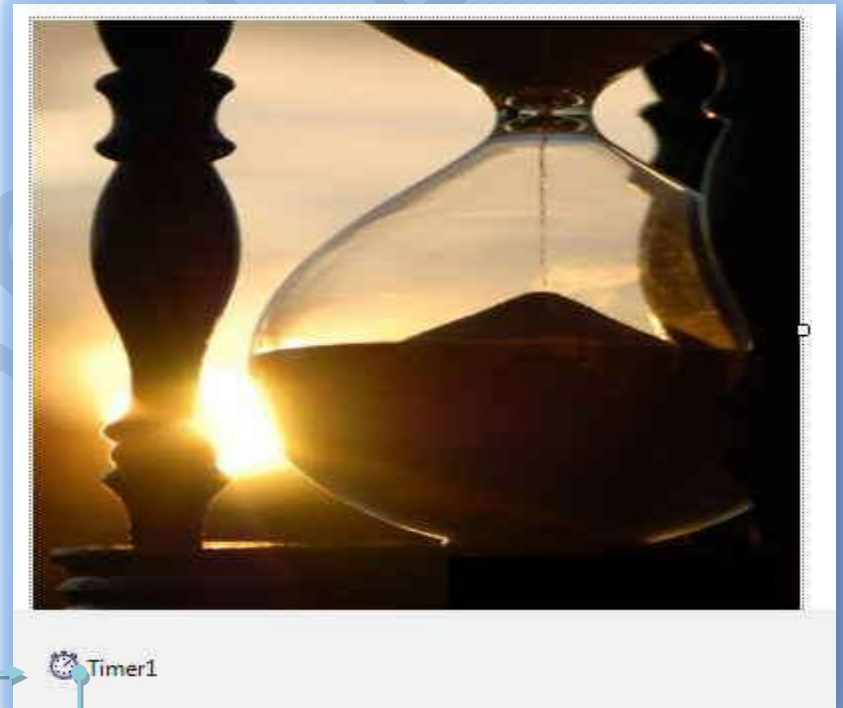
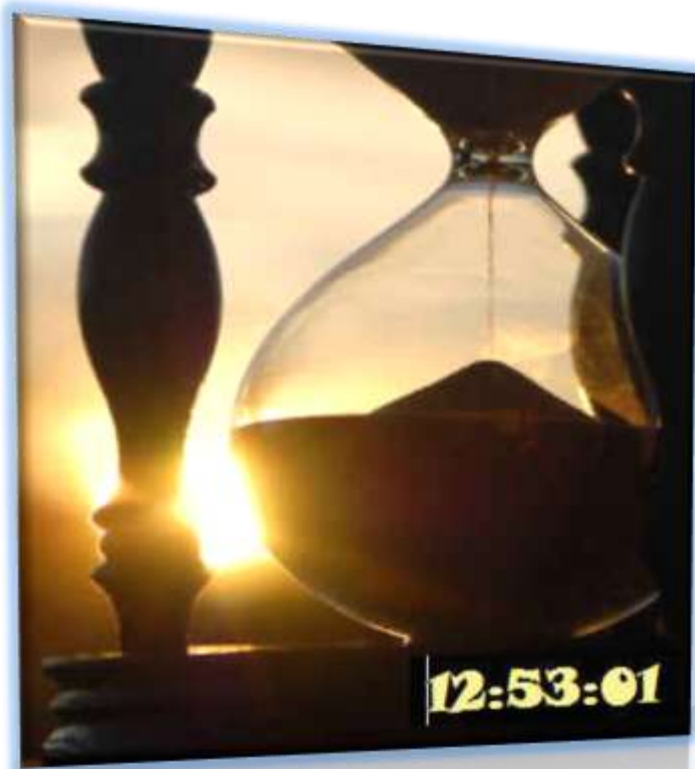
تحميل التمرين الخامس والعشرون



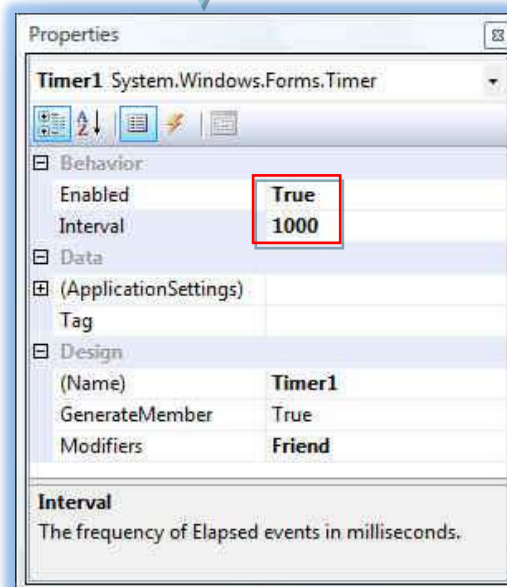
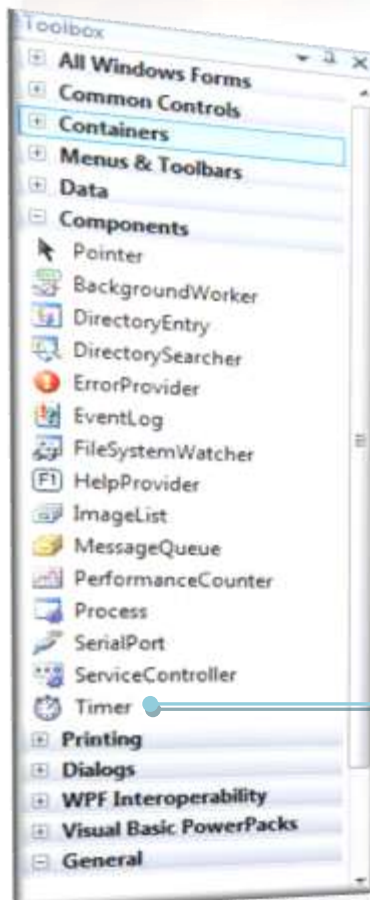
## المؤقت Timer

المؤقت أداة مهمة ومميزة يمكننا الاعتماد عليها في الكثير من المهمات البرمجية فالمؤقت هو عبارة عن ساعة إيقاف مخفية تستطيع التعرف على ساعة النظام والاستفادة من ذلك في برامجك كما تستطيع الاستفادة من أداة المؤقت في الكثير من الأمور بتكرار أمر برمجي معين بين كل وقت وآخرى أو بحساب ساعات العمل على برنامج معين أو بحساب الوقت منذ تشغيل البرنامج أو استخدام المؤقت لغلق البرنامج أو اظهر رسالة أو لعمل ساعة أو منبه للوقت و العديد من المهام البرمجية التي لها علاقة بالوقت

وسوف نقوم في التمرين التالي باستخدام المؤقت **Timer** في عمل تمرين عبارة عن ساعة حقيقية تظهر الوقت الفعلي من جهازك ولعمل ذلك يجب أولا الوصول بشكل النموذج إلى الشكل التالي وهنا نلاحظ أننا سوف نهتم بالشكل الجمالي للنموذج وذلك بجعل خلفيته صورة وإخفاء شريط الأزرار من الأعلى ليكون النموذج بهذا الشكل والساعة هنا مدرجة داخل **Textbox** تم تنسيقه أيضا للوصول إلى هذا الشكل عند عرض الساعة بة وذلك من خلال صندوق الخصائص الخاص بكل منهما والتعامل معه ليس بالجديد علينا هنا ولكن الجديد هو استخدام الأداة **Timer** والتي من شأنها إدراج الوقت في النموذج ليظهر في **Textbox** ويتم ذلك من خلال صندوق الأدوات وباختيار الأداة **Timer** وإدراجها داخل النموذج تظهر لنا بالشكل التالي



ويتم تعديل خواص الأداة **Timer1** كما يلي من صندوق الخواص لها



وبالنقر مرتين على الأداة **Timer1** لإظهار صفحة الكود الخاص بها ويتم كتابة الكود التالي بها

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Timer1.Tick
    TextBox1.Text = TimeString
End Sub
```

هنا يتم توجيه الأداة **Timer1** إلى إظهار الوقت الحالي في **Textbox** وذلك بكتابة الكود **TimeString** والخاص بإظهار الوقت الحالي

تحميل التمرين السادس والعشرون

استخدام المؤقت لتحديد فترة زمنية

كما وضحنا سابقاً بأننا نستطيع استخدام المؤقت **Timer** لكل ما له علاقة بالوقت أو التوقيت فيمكنك حفظ ما يقوم المستخدم بطباعته كل فترة زمنية معينة أو تحديد مدة زمنية معينة لإدخال كلمة السر أو إظهار رسالة ترحيب لمدة معلومة عند تشغيل البرنامج

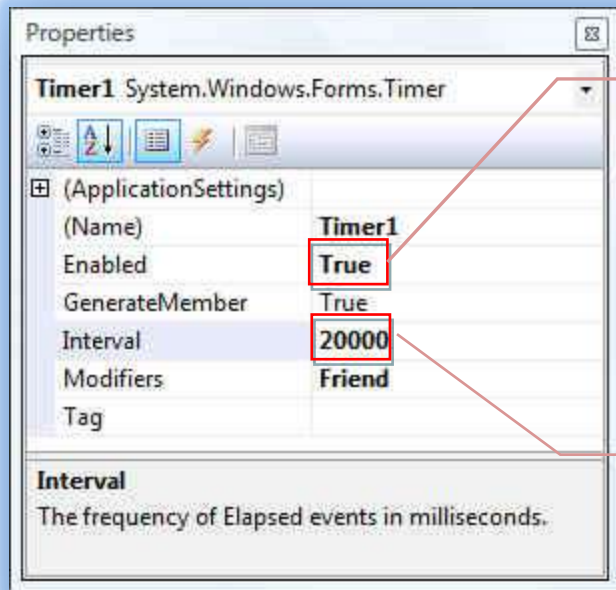


لنأخذ الآن مثال عن برنامج يقوم بطلب كلمة سر للدخول إليه ولا بد على المستخدم أن يقوم بإدخال كلمة السر الصحيحة (1612) خلال 20 ثانية وفي هذه الحالة يعطى البرنامج رسالة تفيد بصحة كلمة السر وفي حالة كتابة كلمة سر خطأ يعطيه أيضاً رسالة تفيد بان كلمة السر خاطئة وإذا تأخر المستخدم عن إدخال كلمة السر خلال 20 ثانية وهو الوقت الذي تم اختياره من المصمم للانتظار المستخدم لإدخال كلمة السر للمرور إلى البرنامج فسيغلق البرنامج تلقائياً بعد إعطاء رسالة بالإغلاق

كلمة السر الصحيحة هي  
1612



التمرين شبيه بالتمرين الثالث في الكتاب ولكن الفرق هنا أننا قمنا باستخدام الأداة **Timer** للتحكم في زمن الرسائل وفترة انتظار البرنامج ويتم إدخال الأداة **Timer** إلى البرنامج كما سبق وتعلمنا ويتم أعدادها كالتالي من خلال صندوق خواصها



يتم جعل الخاصية **Enabled** وذلك لكي نتيج للـ  
**Timer** بالعمل عند بداية التشغيل **F5**

هنا يتم تحديد الوقت الذي سوف ينتظره البرنامج  
للإغلاق وهو **20 ثانية** وتكتب **20000**

ويتم كتابة الكود التالي بالنقر مرتين على الأداة **Timer**

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    MsgBox(" سيتم إغلاق البرنامج الان ", , " Timed Password ")
End
End Sub
```

يقوم البرنامج بالإغلاق بعد النقر على زر  
موافق بالرسالة السابقة

تظهر هذه الرسالة بعد **20 ثانية** كما تم  
الإعداد لها مسبقا وتكون كالتالي



يقوم البرنامج بتوقيف عمل الأداة **Timer** في  
حالة إدخال كلمة السر الصحيحة

ويتم كتابة الكود التالي في زر الدخول **Button1** ويكون

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    If TextBox1.Text = "1612" Then
        Timer1.Enabled = False
        MsgBox(" كلمة المرور صحيحة انتظر تحميل البرنامج ", , " Timed Password ")
    Else
        MsgBox(" رجاء إدخال كلمة المرور الصحيحة ", , " Timed Password ")
        TextBox1.Text = ""
    End If
End Sub
```

تظهر هذه الرسالة في حالة كتابة كلمة  
سر خاطئة



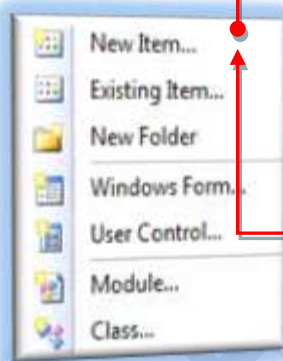
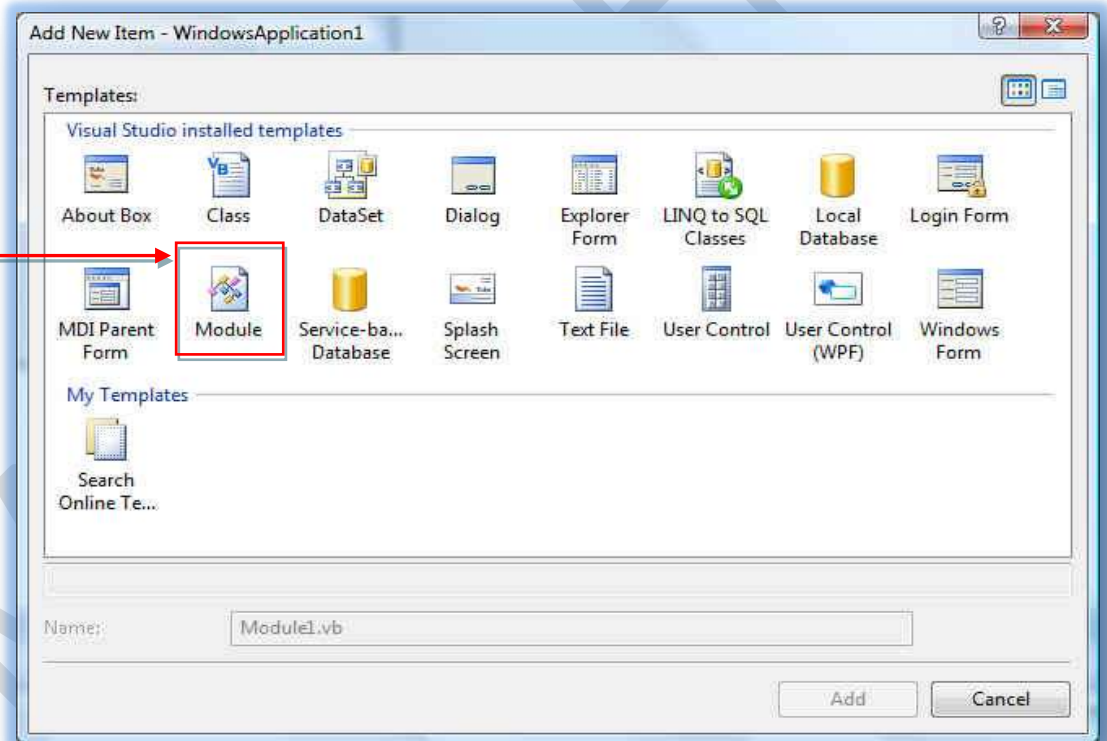
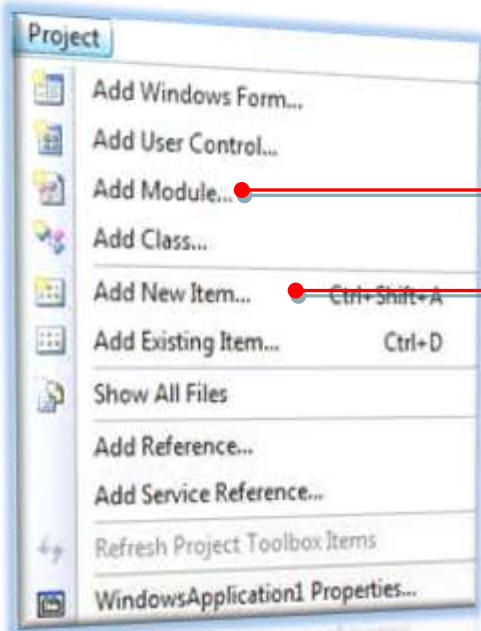
تحميل التمرين السابع والعشرون

### الوحدات البرمجية منطقية Modules

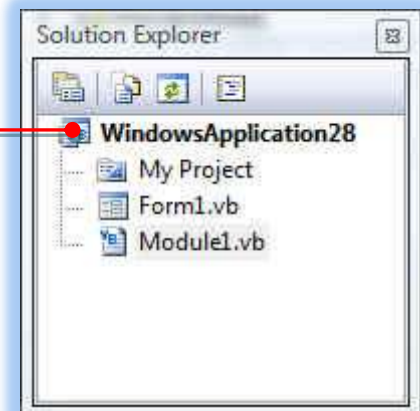
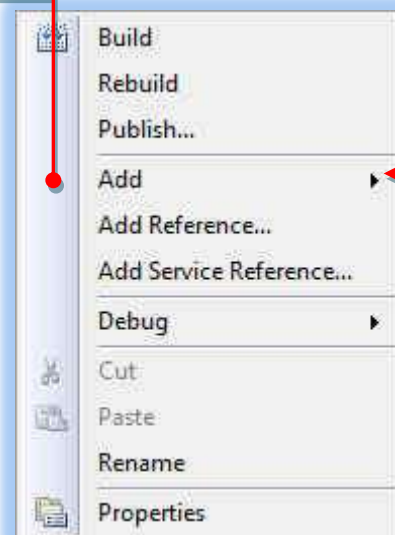
وهي عبارة عن وحدات برمجية في البرنامج تحتوي على متغيرات عامة ودوال برمجية وإجراءات وأحداث نستطيع استخدامها على طول البرنامج ويتم التعامل معها عند برمجة البرامج الكبيرة والمشاريع المتوسطة والعلاقة والتي تحتوي على العديد من النوافذ والإجراءات والتي تستخدم فيها العديد من المتغيرات وبشكل متكرر وفي بيئة التطوير عندما نقوم بتعريف متغير ما فإن هذا المتغير يمكن استخدامه داخلياً فقط بداخل الحدث الذي تم تعريف المتغير فيه أو بداخل الإجراء البرمجي المحدد وإذا توسعت المسألة أكثر وعرفنا المتغير في أعلى منطقة الكود التابعة للفورم فيمكننا استخدام المتغير في داخل الأكواد الموجودة في هذا الفورم فقط وإذا احتجنا لهذا المتغير في فورم آخر فيجب علينا تعريفه مرة أخرى وهذا لا يناسب المشاريع الكبيرة لذلك كان لابد لنا من استخدام الوحدات البرمجية المعروفة **Modules** والتي تساعدنا في اختصار الوقت وتوفير علينا تكرار تعريف المتغير في أكثر من فورم على مستوى المشروع أو التطبيق. فالـ **Modules** هي وحدة برمجية يمكننا تعريف المتغيرات والإجراءات والأحداث بداخلها واستخدامها في أي منطقة في المشروع ككل

### إنشاء وحدة برمجية Creating a Module

هناك عدة طرق في الفيجوال بيسك 2008 لإنشاء وحدة برمجية **Module** فيمكن إدراجها من خلال فتح قائمة **Project** من شريط قوائم البرنامج ومن خلالها يتم اختيار **add Class** لتظهر لنا النافذة التالية ومختار بها **Module** تلقائياً أو يتم اختيار **Add New Item** لتظهر لنا نفس الرسالة ولكننا نقوم باختيار الـ **Module** في هذه الحالة منها



أو من خلال صندوق **Solution Explorer** ويتم ذلك بالوقوف على اسم المشروع والنقر بزر الفأرة الأيمن لتظهر لنا قائمة نختار منها **Add** لتظهر لنا قائمة فرعية ومنها نختار **New Item** لتظهر نافذة نضيف منها الـ **Module** بعد اختياره وإضافته إلى المشروع ليتم العمل عليه



```
Module Module1
```

```
Public View As String
```

ثم يتم كتابة اكواد المشروع

```
End Module
```

والآن بعد إضافة وحدة برمجية **Module** يقوم البرنامج بفتح صفحة الأكواد تلقائياً علماً بأن الآن أي متغير ستقوم بتعريفه يكون متغير عام للمشروع كله ويمكنك استخدامه في أي فورم على طول المشروع وتعريف المتغيرات العامة بداخل الوحدات البرمجية **Module** سهل للغاية فقط قم بكتابة كلمة **Public** ثم قم بإضافة تعريف المتغير كما تعلمنا من قبل لو فرضنا أن هناك متغير أسمة **View** وهو متغير عام من النوع **Short** يتم تعريفه في الـ **Module** كما بالشكل

ومثال على ذلك سنقوم مع بعمل تمرين هو شبيه بالتمرين الخامس من الكتاب ولكن الاختلاف هنا أنه يوجد بة متغير **Wins** ونقوم بإضافة **Label** تظهر فيه جملة **عدد مرات الربح** وإسنادها إلى المتغير الجديد **Wins** لكي يظهر عدد مرات الربح والمرتبطة بظهور الرقم **7** نقوم بعمل نموذج وتنسيقه كما بالشكل كما نقوم بإضافة **Module** إلى المشروع ولكي نقوم بتعريف المتغيرات بة فرضاً أن المتغير المضاف **Wins** متغير عام يستخدم في جميع نماذج المشروع على أن يتم أولاً تعريف المتغير **Wins** كتغير عام داخل الـ **Module** كالتالي



وهنا نطلب من البرنامج أن يقوم بتعريف المتغير **Wins** كمتغير لجميع نماذج المشروع المستخدمة

```
Module Module1
Public wins As String
End Module
```

ثم نقوم بكتابة الكود التالي في الفورم لكي ينفذ عند بدء التشغيل **F5**

```
Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
PictureBox1.Visible = False
End Sub
```

وهنا نطلب من البرنامج أنه عند بداية تشغيل المشروع يتم إخفاء المكون **PictureBox1**

ثم نقوم بكتابة الكود التالي في **Button2** والمسمى **Close**

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

وهنا يتم كتابة كود الإغلاق **End**

ونأتى إلى آخر مرحلة بكتابة الكود في **Button1** والمسمى **Start**

راجع التمرين الثالث لمراجعة شرح الأكواد وسوف أقوم بإضافة الشرح للتعديلات التي أضيفت للمشروع

هنا نعطي كود بآنة عند ظهور الأرقام العشوائية من 1 إلى 10 يقوم بإخفاء الصورة في **PictureBox1** فيها عدا الرقم 7

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Label1.Text = CStr(Int(Rnd() * 10))
    If (Label1.Text = "1" Or "2" Or "3" Or "4" Or "5" Or "6" Or "8" Or "9" Or "10") Then
        PictureBox1.Visible = False
    If (Label1.Text = "7") Then PictureBox1.Visible = True
    If (Label1.Text = "7") Then
        Label2.Visible = True
        Beep()
        wins = wins + 1
        Label2.Text = ("الربح مرات عدد" & wins)
    End If
End Sub
```

لو تم ظهور الرقم 7 في **Label1**  
 1. يتم ظهور الصورة في **PictureBox1**  
 2. يتم إظهار الجملة في **Label2**  
 3. يقوم بإصدار صوت تنبيه **Beep**

وفي حالة ظهور الرقم 7 مرة أخرى يقوم بجمع 1 إلى قيمة المتغير **Win** السابقة ويقوم بإظهار الناتج في الرسالة **Label2** بجانب الجملة السابقة

### تحميل التمرين الثامن والعشرون

مما سبق نجد أن إمامنا ثلاث طرق لتعريف المتغيرات

1. تعريف المتغير داخل المكون نفسه وذلك في حالة استخدامه في اكواد هذا المكون فقط ويكون الكود كالتالي

(نوع) **As** (اسم المتغير) **Dim**

2. تعرف المتغير أول صفحة الأكواد في المنطقة **Public Class** وذلك في حالة استخدام هذا المتغير في معظم اكواد المكونات المدرجة بهذا النموذج ويكون الكود كالتالي

(نوع) **As** (اسم المتغير) **Dim**

3. تعريف المتغير في الوحدة النمطية **Module** وذلك في حالة استخدامه في جميع نماذج المشروع القائمين عليه ويكون الكود الخاص به كالتالي

(نوع) **As** (اسم المتغير) **Public**

نلاحظ أنه من المهم جدا تحديد نوع المتغير عند تعريفه في كل مما سبق لان الناتج يعتمد اعتمادا كليا على نوع المتغير من خلال تعريفنا له والمثال التالي خير برهان على أن تحديد نوع المتغير يؤثر تأثيرا مباشرا على النتيجة النهائية للمخرجات وهو عبارة عن برنامج حسابي يتم فيه حساب القيمة الإجمالية لأي رقم ( مبلغ مثلا ) تقوم بإدخاله داخل **Textbox** بعد إضافة قيمة ربح قدرها **11%** إلى الرقم الذي تم إدخاله ( المبلغ الأصلي )

### تحميل التمرين التاسع والعشرون

استخدام المصفوفات للتعامل مع البيانات الرقمية والنصية  
تعرف المصفوفات بأنها الطريقة الأمثل للتعامل مع كمية كبيرة من البيانات المتشابهة فنحن نعلم بأن البرامج تعتمد بشكل أساسي على  
البيانات ( المدخلات ) ويقوم البرنامج بالتعامل مع هذه البيانات وتحليلها بحسب البرمجة المسبقة لهمن خلال المصمم وعند التعامل مع  
البيانات في فيجوال بيسك 2008 تعلمنا كيف نقوم باستخدام المتغيرات لحفظ هذه البيانات في متغيرات ثم إسناد قيمة هذه المتغيرات إلى  
خاصية معينة في صندوق نص أو غيره لكن عندما يكون لدينا العديد من البيانات المتشابهة فلا بد لنا من استخدام المصفوفات للتعامل مع  
هذه البيانات وللعلم توجد المصفوفات في معظم لغات البرمجة للتعامل مع البيانات فإذا كان لدينا مصفوفة فيها ثلاثة أعمدة وثلاثة صفوف  
فستطيع استخدامها للتعامل مع تسعة من البيانات المتشابهة

#### إنشاء المصفوفات

لا بد هنا أن نعلم بأنه عند كتابة المصفوفة في مكان ما فإن هذا المكان يحدد لنا نطاق استعمال المصفوفة فمثلا عند كتابة المصفوفة في  
داخل إجراء أو حدث معين لأي مكون مدرج بالنموذج فإننا نستخدم المصفوفة في داخل ذلك الإجراء فقط، أما إذا قمنا بكتابة المصفوفة في  
بداية الفورم ( أعلى منطقة الكود في الفورم ) فإن المصفوفة يمكن استخدامها على طول الكود في الفورم وفي حالة كتابة المصفوفة بداخل  
وحدة برمجية Module فنستطيع استخدام المصفوفة في أي مكان في المشروع وهذا ليس بالجديد علينا فقد عرفناه من خلال التعامل مع  
المتغيرات من قبل وعند إنشاء أي مصفوفة فانت باختصار تجيب عن هذه أو توضح المعلومات التالية عند تعريف المصفوفة

#### اسم المصفوفة

اسم المصفوفة هو الاسم الذي ستستخدمه لاستدعاء مصفوفتك في أي مكان في الكود واسم المصفوفة لا بد أن يخضع لنفس الشروط  
اللازمة لاسم المتغير راجع شروط كتابة اسم للمتغير مما سبق

#### نوع البيانات

لا بد لنا من تحديد نوعية البيانات التي سوف يتم تسجيلها في المصفوفة وفي أغلب الحالات تكون هذه البيانات من نوع واحد  
( كلها نصية أو كلها وقتية أو كلها رقمية ) ونادراً ما تكون أكثر من نوع ولا بد لك من تحديد نوعية البيانات داخل المصفوفة لتحصل على  
التعامل الأمثل للبيانات من قبل بيئة التطوير أما إذا لم تعرف نوعية البيانات التي سوف تستخدمها بداخل المصفوفة أو كان هناك أكثر من  
نوع من أنواع البيانات فلا بد لك إذا من استخدام النوع Object للتعبير عن البيانات

#### أبعاد المصفوفة

لا بد لك من تحديد مقاسات أو أبعاد المصفوفة فهناك مصفوفات ذات بعد واحد (مثل قائمة من البيانات ) وهناك ذات بعدين (مثل جدول من  
البيانات ) وهناك مصفوفات بثلاثة أبعاد للمصفوفات مثل ( التي تعالج مجموعة كبيرة ومعقدة ومتداخلة من البيانات )

#### عناصر المصفوفة

عناصر المصفوفة أو عدد عناصر المصفوفة هو عدد العناصر التي ستتعامل معها بداخل المصفوفة وهذه العناصر تطابق عدد العناصر في  
القائمة التي ترتبها المصفوفة ولا بد أن يبدأ الترتيب من الرقم **صفر 0** وفي الفيجوال بيسك 2008 إذا كان عدد العناصر **تسعة 9** فإن بيئة  
التطوير ترتبهم في قائمة من **0 إلى 8** حيث يأخذ العنصر الأول الترتيب صفر والعنصر الثاني الترتيب واحد وهكذا

#### أنواع المصفوفات

#### A. المصفوفة الثابتة Fixed-Size array

وهي المصفوفة التي تحتوي على عدد ثابت ومحدد من العناصر و يكون كود تعريف المصفوفات الثابتة كالتالي

**Dim** ArrayName (Dim1Index, Dim2Index, ...) **As** DataType

الكلمة التي تستخدم لتعريف المصفوفة  
وتختلف تبعاً لمكان استخدامها في  
نموذج أو في Module كما سبق  
وذكرنا

نوعية بيانات المصفوفة  
وتحدد نوعها على حسب  
نوع المدخلات كما سبق  
وذكرنا

اسم المصفوفة التي نريد تعريفها  
نرجو مراجعة شروط اختيار اسم  
المتغيرات لأنهم متشابهين

عدد عناصر المصفوفة ويتم التعامل  
معها كما سبق على أن يكون الكون  
لمصفوفة من البعد التاسع ( 8 )

وبتطبيق القاعدة السابقة على مصفوفة ذات بعد واحد وعدد عناصرها **10** وأسم المصفوفة هو **Student** ومن النوع **النصي**

لو يتم تعريفها في النموذج أو في أى إجراء لمكون مدرج يكون كالتالي

**Dim Student (0 To 9)As String**

**Or**

**Dim Student (9)As String**

**Puplic Student (0 To 9)As String**

**Or**

**Puplic Student ( 9)As String**

لو يتم تعريفها في الوحدة النمطية **Module** يكون كالتالي

لأننا اتفقنا أن المصفوفة يتم  
حسابها من أول العنصر **0**  
وليس **1**

لماذا كتبنا من ( 0 To 9 )  
فقط و عدد العناصر هي **10**

ماذا لو أردنا تطبيق القاعدة السابقة على مصفوفة ذات **بعدين** فمثلا نريد أن ننشئ مصفوفة لجدول مباراة تنس يكون اسم المصفوفة هو **Score** ومن النوع **الرقمي** ويكون عدد عناصر **البعد الأول 10** وهو عدد الجولات في المباراة وعدد عناصر **البعد الثاني 2** وهما اللاعبين المتنافسين في المباراة

لو يتم تعريفها في النموذج أو في أى إجراء لمكون مدرج يكون كالتالي

**Dim Score (0 To 1 , 0 To 9)As Short**

**Or**

**Dim Score (1 , 9)As Short**

**Puplic Score (0 To 1 , 0 To 9)As Short**

**Or**

**Puplic Score (1 , 9)As Short**

لو يتم تعريفها في الوحدة النمطية **Module** يكون كالتالي

ويقوم الكمبيوتر فور تشغيل البرنامج الذي يحتوى على مصفوفة بحجز مكان له في وحدة الذاكرة كالتالي

الحالة الثانية									الحالة الأولى		
9	8	7	6	5	4	3	2	1	0	0	
										1	
										2	
										3	
										4	
										5	
										6	
										7	
										8	
										9	

( صفيين و 9 أعمدة )

( عمود واحد و 9 صفوف )

ولكن كيف سيتم التعامل مع هذه المصفوفات وإدخال المعلومات إليها



## التعامل مع عناصر المصفوفات

للتعامل مع عناصر المجموعات سواء أردنا الحصول على قيمة معينة تابعة لعنصر معين في المصفوفة أو تعبئة عنصر معين في المصفوفة بقيمة معينة فلا بد لنا من تحديد ترتيب العنصر في المصفوفة ويكون ذلك شبيه بطريقة الإحداثيات في الرسم البياني كما تعلمنا في سنوات الدراسة كتابة قيمته ليتم تعبئتها مثلا

الحالة الأولى	
0	
1	
2	
3	
4	Mohamed
5	
6	
7	
8	
9	

( عمود واحد و 9 صفوف )

المصفوفة الموجودة بالحالة الأولى والتابعة للطلاب Student نكتب الكود التالي لإضافة اسم الطالب الرابع على أن يكون أسمة **Mohamed**

Student(4) = "Mohamed"

المصفوفة الموجودة بالحالة الثانية والتابعة للنتيجة Score نكتب الكود التالي لإضافة نتيجة لصالح اللاعب الأول الذي حصل على 7 نقاط خلال الجولة الخامسة

الحالة الثانية									
9	8	7	6	5	4	3	2	1	0
					7				0
									1

( صفين و 9 أعمدة )

score(0,4) = "7"

بداية ونهاية المصفوفة **Lbound and Ubound** نستطيع تحديد تسلسل أعلى قيمة في المصفوفة وتسلسل أقل قيمة في المصفوفة بواسطة الدوال **Lbound and Ubound** حيث أن الدالة **Lbound** ترمز لتسلسل أقل قيمة في المصفوفة وكما هو معروف بأن المصفوفات في فيجوال بيسك 2008 تبدأ من التسلسل صفر فإن الدالة **Lbound** سترجع لنا القيمة صفر و الدالة **Ubound** فترجع لنا تسلسل أعلى قيمة في المصفوفة فإذا كانت المصفوفة ذات سبعة ( 7 ) عناصر فإن الدالة **Ubound** سترجع القيمة السادسة ( 6 ) لنا لان التسلسل يبدأ من الرقم صفر ( 0 ) كما ذكرنا سابقاً

تصميم مصفوفة لخرن درجة الحرارة لأيام الأسبوع سنقوم الآن بتصميم مصفوفة ثابتة ذات بعد واحد تقوم بخرن درجة الحرارة لأيام الأسبوع ( 7 ) وسنقوم بتعبئتها باستخدام صندوق الإدخال **InputBox** والتي تظهر لنا لإدخال درجات الحرارة لأيام الأسبوع عند الضغط على **Button1** والتعامل معها باستخدام الدالة **For Next Loop** التي عرفناها فيما سبق فنحن نستخدم الدالة **For Next Loop** للتعامل مع كل بند من بنود المصفوفة على حده إذا استلزم الأمر ونقوم بعرض عناصر المصفوفة بداخل **Label1** مدرج بداخل الفورم ويظهر عند النقر على الزر **Button2** وفي نفس الوقت سنقوم بحساب متوسط حرارة الأسبوع

ومما سبق نستنتج أن النموذج يتكون من

1. عدد **Button 1** وذلك لإدخال درجات الحرارة من خلال صندوق الإدخال **InputBox** لكل يوم على حدة
2. عدد **Button1** وذلك لإظهار **Label1** و التي تظهر بة درجات الحرارة التي قمنا بإدخالها من قبل لكل يوم على حدة وأيضا يظهر فيه متوسط درجات الحرارة للأسبوع
3. عدد **Label 1** وهو الذي تظهر بة المعلومات المطلوبة مسبقا ويتم إخفائه منذ بداية التشغيل للنموذج **F5**

اعتقد أني قد وصلت بكم الآن إلى مرحلة استطيع فيها أن اترك لك دائما حرية التنسيق للمكونات المدرجة في النموذج على ان تخدم الأهداف المرجوة منها ويمكنك معرفة التنسيق الذي لم اذكره من خلال صناديق الخصائص لكل كائن مدرج في النموذج على حدة



هذا هو تصميمي للنموذج وهو غير مقيد لك فحاول أن تترك لنفسك مجال من الابتكار لأن المشكلة ليست فقط نسخ خطوات اكواد ولكن حاول أن تبتكر أو تضيف أي شيء خاص بك بالنموذج أو البرنامج حتى لو جملة نصية من خلال ما تعلمته سابقا

في بداية الأمر نحن متفقين على استخدام المصفوفات في إدخال المعلومات إلى البرنامج وهي درجات الحرارة اليومية كما أن المصفوفة التي نعمل عليها هي مصفوفة ثابتة ذات بعد واحد فقط وتمثل عدد أيام الأسبوع السبعة

إذا مصفوفتنا في هذا التمرين هي تمثل نفس شكل المصفوفة التي درسناها سابقا في الحالة الأولى والمكونة من **عمود واحد فقط** ولكن هنا سوف تتكون من **7 صفوف** وهي **أيام الأسبوع** ابتداء من الرقم **0** كما عرفنا من قبل وسوف تكون مصفوفتنا في هذا التمرين اسمها **Temp** ومن النوع الرقمي **Integer**

و أيضا متفقين أن المصفوفة المستخدمة سوف يتم التعامل معها على طول النموذج للمشروع فلو كان المشروع

يتكون من أكثر من نموذج فوجب علينا في هذه الحالة إنشاء **Module** وكتابة التعريف بالبطريقة المتبعة في هذه الحالة كما عرفنا سابقا ولكن مشروعنا هنا على نموذج واحد فقط ولذلك يكفي أن نقوم بتعريف المصفوفة في النموذج الحالي فقط ولكن في بداية الفورم كما يلي

مصفوفة اسمها **Temp** تتكون من عمود واحد وسبع صفوف من **0** إلى **6** ويتم تعريفها على إنها رقمية **Integer**

```
Public Class Form1
    Dim Temp(0 To 6) As Integer
```

وبعد إدخال كل من المكونات السابق ذكرها يتم التعامل أولا مع **Button1** والمسمى ( **إدخال درجات الحرارة** ) ويتم كتابة الكود بة كالتالي

يتم تعريف المتغيرات الخاصة بالرسائل داخل صندوق الإدخال كما تعلمنا سابقا

يتم إخفاء **Label1** وذلك في حالة ظهوره عند إدخال درجات الحرارة الجديدة

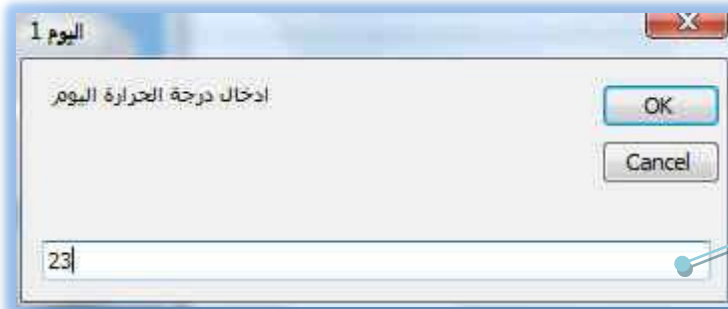
الرسالة التي تظهر داخل صندوق الإدخال **InputBox**

```
Dim Prompt, Title As String
Dim n As Short
Label1.Visible = False
Prompt = "إدخال درجة الحرارة لليوم"
For n = 0 To UBound(Temp)
    Title = "اليوم " & (n + 1)
    Temp(n) = InputBox(Prompt, Title)
Next
```

الرسالة التي تظهر في شريط صندوق المدخلات **InputBox**

هنا يتم تعريف الدالة للمصفوفة بأنها تبدأ من **0** إلى أعلى قيمة داخل المصفوفة وهي **6** حسب تعريفها السابق

هنا يتم إدخال درجات الحرارة في المصفوفة **Temp** عن طريق صندوق المدخلات **InputBox** وظهوره حسب التنسيق المذكور



وهكذا نكون انتهينا من كتابة الكود في زر (إدخال درجات الحرارة) وعند تشغيل البرنامج **F5** والنقر على تظهر الرسالة التالية

هنا يتم إدخال درجات الحرارة إلى المصفوفة لكل يوم من أيام الأسبوع

وننتقل إلى المرحلة الأخرى في كتابة الأكواد وهي كتابة الكود الخاص بالـ **Button2** والمسمى (إظهار درجات الحرارة) ويكون كتابة الكود كالتالي

يتم تعريف المتغيرات الخاصة بالرسائل داخل صندوق الإدخال كما تعلمنا سابقا

يتم إظهار **Label1** والذي تم إخفائه منذ البداية من خلال خصائصه لظهور المدخلات

```

Dim n As Short
Dim Result As String
Dim Total As Single = 0
Label1.Visible = True
Result = " اعلى درجة حرارة خلال الاسبوع هي " & vbCrLf & vbCrLf
For n = 0 To UBound(Temp)
    Result = Result & " درجة الحرارة اليوم هي " & (n + 1) & vbCrLf & _
    Temp(n) & vbCrLf
    Total = Total + Temp(n)
Next
Result = Result & vbCrLf & _
" متوسط درجات الحرارة هو " & Format(Total / 7, "0.00")
Label1.Text = Result
End Sub

```

هنا يتم تعريف الدالة للمصفوفة بأنها تبدأ من 0 إلى أعلى قيمة داخل المصفوفة وهي 6 حسب تعريفها السابق ويتم إسناد المتغير الذي تم تعريفه سابقا بـ **Result** إلى كل من

1. إظهار النتيجة السابقة **Result**
2. إظهار النص المذكور " "
3. إضافة 1 إلى المتغير **n** وهنا ليظهر رقم تسلسلي لأيام الأسبوع
4. إضافة التنسيق **vbTab** وهي تقوم بعمل الزر **Tab** من لوحة المفاتيح
5. إضافة قيمة درجة الحرارة من المصفوفة **Temp** كما سبق وأدخلناها
6. إضافة التنسيق **vbCrLf** وهي للبدائية من سطر جديد
7. ثم إغلاق الدالة **Next** بـ **Next**

إظهار المتغير **Result** في المكون **label1**

يتم إسناد المتغير **Result** إلى كل من

1. درجة الحرارة المدخلة من خلال المتغير **Result**
2. إضافة التنسيق **vbCrLf** وهي للبدائية من سطر جديد
3. إظهار النص المذكور " "
4. إجراء العملية الحسابية للحصول على المتوسط لدرجات الحرارة وذلك بجمع كل درجات الحرارة المدخلة وقسمتها على عدد أيام الأسبوع وإظهار الناتج إلى رقمين عشريين

يتم إسناد المتغير الذي تم تعريفه سابقا بـ **Result** إلى كل من

1. النص المذكور " "
2. إضافة التنسيق **vbCrLf** وهي للبدائية من سطر جديد
3. إضافة التنسيق **vbTab** وهي تقوم بعمل الزر **Tab** من لوحة المفاتيح

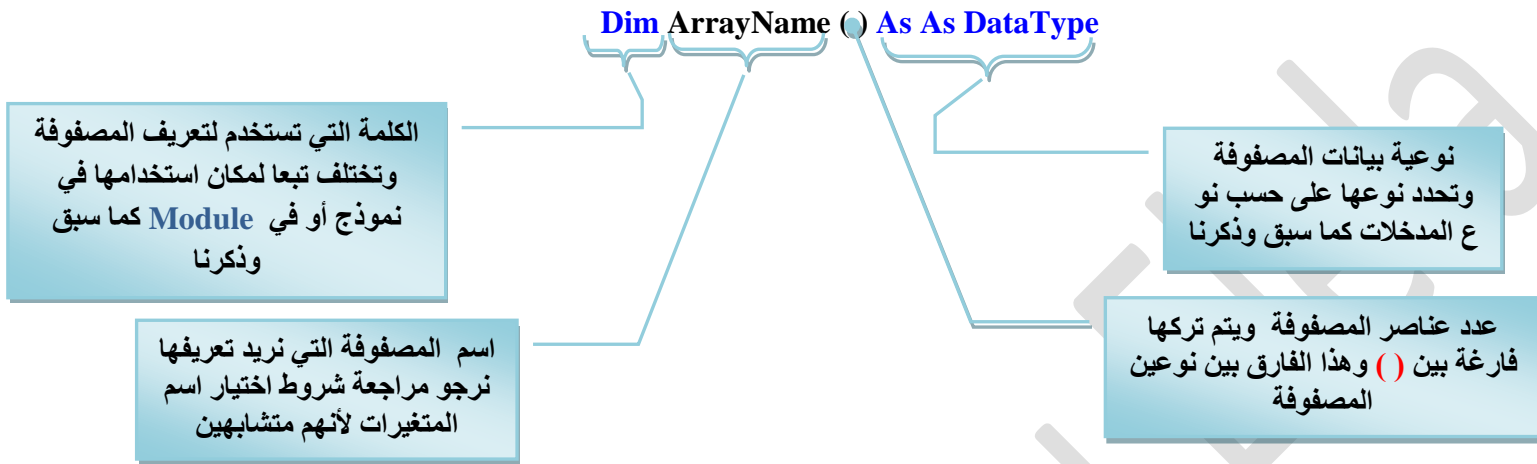
يتم استخدام العلامة & (بالشكل) للدمج بين المتغيرات أو نصوص أو تنسيق داخل الأكواد

وعند التشغيل **F5** بالنقر على الزر (إظهار درجات الحرارة) يتم إظهار **Label1** وبة كل المعلومات المدرجة داخل المصفوفة وأيضا فيمه متوسط درجات الحرارة كما بالشكل السابق

تحميل التمرين الثلاثون

### B. المصفوفة المتغيرة ( التفاعلية ) Fixed-Dynamic array

هي المصفوفة التي تحتوي على عدد متغير من العناصر كذلك التي يتغير عدد عناصرها في وقت تشغيل البرنامج بسبب البيانات التي تتعامل معها تسمى بالمصفوفات المتغيرة أو المصفوفات التفاعلية **Dynamic array** وهذا النوع من المصفوفات لا يحتاج لتحديد عدد عناصر المصفوفة في مرحلة الكود وإنما في مرحلة تشغيل البرنامج **Runtime** و يكون كود تعريف المصفوفات المتغيرة كالقاعدة التالية



نلاحظ أنه قد تم تعريف المصفوفة المتغيرة هنا بدون ذكر عدد عناصرها في مرحلة الكود ولكن بعد كود تعريف المصفوفة نقوم بكتابة كود آخر يسمح بكتابة عدد عناصر المصفوفة في مرحلة تشغيل البرنامج **Runtime** وبذلك يتم السماح للمستخدم بكتابة عدد عناصر المصفوفة باستخدام الكود بدون كتابة عدد العناصر في مرحلة الكود من البداية وإنما إسناد العدد إلى متغير معين ويقوم مستخدم البرنامج بتعبئة هذا المتغير ولتوضيح الفكرة سنقوم الآن بتصميم مصفوفة ثابتة ذات **بعد واحد** تقوم بخزن درجة الحرارة **لعدد غير معلوم من الأيام** وسنقوم بتعبئتها باستخدام صندوق الإدخال **InputBox** والتي تظهر لنا لإدخال درجات الحرارة لهذه الأيام عند الضغط على **Button1** والتعامل معها باستخدام الدالة **For Next Loop** التي عرفناها فيما سبق فنحن نستخدم الدالة **For Next Loop** للتعامل مع كل بند من بنود المصفوفة على حده إذا استلزم الأمر ونقوم بعرض عناصر المصفوفة بداخل **Label1** مدرج بداخل الفورم ويظهر عند النقر على الزر **Button2** وفي نفس الوقت سنقوم بحساب متوسط حرارة عدد الأيام المدخلة

التمرن هو نفس التمرين السابق ولكن الفرق هنا أننا لم نحدد عدد الأيام في المصفوفة بأنفسنا ولكن تركنا المستخدم هو الذي يقوم بذلك من خلال تعديل الأكواد لإظهار صندوق إدخال آخر **InputBox** في بداية تشغيل البرنامج لكي يحدد فيه عدد الأيام وبذلك يقوم بتحديد عناصر المصفوفة حسب رغبة المستخدم ثم يقوم البرنامج بتنفيذ نفس الأكواد السابق ذكرها في التمرين السابق (الثلاثون) ويكون التعديل كالتالي

يتم تعريف المصفوفة المتغيرة حسب قاعدتها وتعريف متغير جديد **Days** ويكون للفورم ككل

مصفوفة اسمها **Temp** تتكون من عمود واحد وسبع صفوف من **0 إلى 6** ويتم تعريفها على إنها رقمية **Integer**

```
Public Class Form1
    Dim Temp() As Integer
    Dim Days As Short
```

وبعد إدخال كل من المكونات السابق ذكرها يتم التعامل أولاً مع **Button1** والمسمى ( **إدخال درجات الحرارة** ) ويتم إضافة الأكواد التالية

يتم إسناد صندوق المدخلات الجديد إلى المتغير **Days** الذي تم إضافته وتنسيقه

```
Dim Prompt, Title As String
Dim n As Short
Label1.Visible = False
Prompt = "إدخال درجة الحرارة لليوم"
Days = InputBox("؟ كم عدد الأيام", "قم بإنشاء المصفوفة")
If Days > 0 Then ReDim Temp(Days - 1)
For n = 0 To UBound(Temp)
    Title = "اليوم " & (n + 1)
    Temp(n) = InputBox(Prompt, Title)
Next
```

يتم إضافة القاعدة **If** حيث أنه إذا كان المتغير **Days** أكبر من الصفر أي له قيمة معلومة يتم تغيير تعريف المصفوفة **Temp** إلى القيمة التي تم إدخالها

طبعا يتم طرح 1 من القيمة المدخلة لان المصفوفة تبدأ من الصفر كما سبق وشرحنا

وبإضافة هذا الكود تظهر لنا الرسالة التالية في عند تشغيل البرنامج والنقر على الزر ( **إدخال درجات الحرارة** ) لكي يقوم المستخدم بإعادة تعريف عناصر المصفوفة من خلاله وهو بإدخال عدد الأيام التي سوف يتم التعامل معها

هنا يتم إدخال درجات عدد عناصر المصفوفة والممثلة في عدد الأيام

ثم يقوم البرنامج بعد ذلك باستكمال الخطوات التي تم إعدادها من قبل حتى النهاية كما سبق

هنا يتم إدخال درجات الحرارة إلى المصفوفة لكل يوم من أيام الأسبوع

وننتقل إلى المرحلة التالية وهي إدخال التعديلات على الكود الخاص بالـ **Button2** والمسمى (إظهار درجات الحرارة) ويكون تعديل الكود كالتالي

```
Dim n As Short
Dim Result As String
Dim Total As Single = 0
Label1.Visible = True
Result = " اعلى درجة حرارة خلال الاسبوع هي " & vbCrLf & vbTab
For n = 0 To UBound(Temp)
    Result = Result & " درجة الحرارة اليوم هي " & (n + 1) & vbTab & _
    Temp(n) & vbCrLf
    Total = Total + Temp(n)
Next
Result = Result & vbCrLf & _
" متوسط درجات الحرارة هو " & Format(Total / Days, "0.00")
Label1.Text = Result
End Sub
```

إجراء العملية الحسابية للحصول على المتوسط لدرجات الحرارة وذلك بجمع كل درجات الحرارة المدخلة وقسمتها على عدد الأيام التي تم تحديدها من خلال المستخدم في المتغير **Days** وإظهار الناتج إلى رقم عشري واحد



وعند التشغيل **F5** وتحديد عدد الأيام إدخال درجات الحرارة كما سبق من خلال الزر (إدخال درجات الحرارة) يتم النقر على الزر (إظهار درجات الحرارة) يتم إظهار **Label1** وبة كل المعلومات المدرجة داخل المصفوفة وأيضا فيمه متوسط درجات الحرارة كما بالشكل

تحميل التمرين الواحد و الثلاثون



تعلمنا مما سبق كيف نصمم المصفوفات وكيف نتعامل هذه المصفوفة مع البيانات وكيف تقوم بتخزينها و في هذه الخطوة سنتعلم أكثر عن الطرق المتوفرة ضمن بيئة التطوير والتي تساعدنا على التعامل الأمثل مع البيانات الموجودة بداخل المصفوفة حيث توفر لنا بيئة التطوير العديد من المميزات التي يمكن أن نستخدمها في التعامل مع البيانات الموجودة بداخل المصفوفة فيمكننا البحث بداخل البيانات و ترتيب البيانات أو عكس الترتيب وتنفيذ مهام أخرى

سوف نعمل تمرين يقوم بتوليد الأرقام العشوائية من خلال **مصفوفة** من **النوع الثابت** اسمها **Rand** ومن النوع الرقمي **Integer** وعدد عناصرها هو **200 عنصر** وإضافة مجموعة من الأرقام العشوائية و زر ترتيب هذه الأرقام تصاعديا و زر عكس الترتيب التصاعدي إلى ترتيب تنازلي و زر إغلاق كما سنتعلم في تمريننا هذا كيفية إضافة شريط التطوير من خلال صندوق الأدوات ويعرض لك هذا الشريط نسبة ما قد تم إنجازه من العملية ونسبة المتبقي بواسطة إظهار مستطيلات خضراء في المسار الأبيض وعندما يمتلئ الفراغ الأبيض بالمستطيلات فهذا يعني اكتمال العملية يسمى هذا الشريط **Progress bar** ونلاحظ هذا الشريط كثيرا عند تنزيل البرامج وهناك خاصيتين هامتين لهذا الشريط وهما **Maximum** وتعني القيمة العليا و **Minimum** القيمة الدنيا ويتم ملئ الشريط بالاعتماد على هاتين القيمتين ويمكننا التحكم بهما في مرحلة الكود أو من خلال صندوق خصائص شريط التطوير كما سنرى نحاول عمل نموذج بة المكونات التي بالشكل ثم ننتقل لمرحلة كتابة الأكواد وتكون كالتالي

1. تعريف المصفوفة **Rand** كما تعلمنا من قبل وتكون كالتالي

```
Public Class Form1
    Dim Rand(0 To 199) As Integer
```

2. كتابة كود خصائص شريط التطوير **Progress bar** في النموذج **form1**

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ProgressBar1.Minimum = 0
    ProgressBar1.Maximum = UBound(Rand)
End Sub
```

القيمة الدنيا للـ **Progress bar** وتكون صفر ( 0 )

القيمة العليا للـ **Progress bar** وتكون منسوبة إلى اعلي قيمة بالمصفوفة

3. كتابة الكود في زر توليد الأرقام العشوائية و سوف نلاحظ من كتابتنا الأكواد هنا أنها قد مرت علينا جميعا من قبل في تمارين سابقة مثل التمرين الخامس والثلاثون

يتم تعريف المتغير **n** على أنه متغير عددي

يتم تفريع **TextBox1** من محتوياته

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim n As Integer
    TextBox1.Text = ""
    For n = 0 To UBound(Rand)
        Rand(n) = Int(Rnd() * 500)
        TextBox1.Text = TextBox1.Text & Rand(n) & vbCrLf
        ProgressBar1.Value = n
    Next n
End Sub
```

هنا يتم انساب قيمة شريط التطوير **ProgressBar1** إلى القيمة **n** حتى يكتمل مع اكتمال عدد عناصر المصفوفة

هنا يتم إدراج كود التوليد العشوائي كما تعلمنا في التمرين الخامس ولكن الفرق هنا أنه يولد الكود لعدد **500** رقم

يتم استخدام الدالة **For...Next** للمصفوفة كما سبق وشرحنا في التمرين الثلاثون

الرجاء الاستعانة بالتمرين الخامس والتمرين الثلاثون للشرح الوافي للأكواد إذا لم تستطع تحصيلها هنا

4. كتابة الكود في زر ترتيب الأرقام تصاعدي و سوف تكون أكواد كالتالي

يتم تعريف المتغير **n** على أنه متغير عددي

يتم تنسيق **TextBox1** لإظهار الأرقام حسب التنسيق

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Dim n As Integer
    TextBox1.Text = "ترتيب الأرقام تصاعدي" & vbCrLf
    Array.Sort(Rand)
    For n = 0 To UBound(Rand)
        TextBox1.Text = TextBox1.Text & Rand(n) & vbCrLf
        ProgressBar1.Value = n
    Next n
End Sub
```

هنا يتم انساب قيمة شريط التطوير **ProgressBar1** إلى القيمة **n** حتى يكتمل مع اكتمال عدد عناصر المصفوفة

يتم تنسيق **TextBox1** لإظهار الأرقام حسب التنسيق

يتم استخدام الدالة **For...Next** للمصفوفة كما سبق وشرحنا في التمرين الثلاثون

هنا يتم إدراج كود الترتيب التصاعدي للأرقام العشوائية الناتجة من عملية توليد الأرقام السابقة



## 5. كتابة الكود في زر ترتيب الأرقام تنازلي و سوف تكون أكوادة كالتالي

يتم تعريف المتغير **n** على أنه متغير عددي

يتم تنسيق **TextBox1** لإظهار الأرقام حسب التنسيق

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
    Dim n As Integer
    TextBox1.Text = "ترتيب الارقام تنازلي" & vbCrLf
    Array.Reverse(Rand)
    For n = 0 To UBound(Rand)
        TextBox1.Text = TextBox1.Text & Rand(n) & vbCrLf
        ProgressBar1.Value = n
    Next n
End Sub
```

هنا يتم انساب قيمة شريط التطوير **ProgressBar1** إلى القيمة **n** حتى يكتمل مع اكمال عدد عناصر المصفوفة

يتم تنسيق **TextBox1** لإظهار الأرقام حسب التنسيق

يتم استخدام الدالة **For...Next** للمصفوفة كما سبق وشرحنا في التمرين الثلاثون

هنا يتم إدراج كود عكس الترتيب السابق سواء تم تنفيذه بعد التوليد العشوائي أو بعد الترتيب التصاعدي سوف يقوم بعكس ترتيب الأرقام في كل من الحالتين

ترتيب الأرقام التصاعدي

```
0
7
7
9
11
14
15
22
22
22
26
27
28
30
36
40
40
40
44
```

ترتيب الأرقام تنازلي

```
499
498
496
493
490
489
489
480
480
474
474
469
464
463
463
461
461
459
458
```

تحميل التمرين الثاني و الثلاثون

## التعامل مع المجموعات Collections

المجموعات **Collections** هي عبارة عن مجموعة من الكائنات **Objects** التي توجد في تطبيقاتنا وبشكل أوضح هي عناصر التحكم الموجودة على الفورم مثل صناديق النص والأزرار وغيرها فبيئة التطوير تقوم بحفظ جميع الكائنات على الفورم مع الكود في ملف واحد ولكننا لم نعرف بأن بيئة التطوير تتعامل مع هذه الكائنات على أنها أعضاء في مجموعته واحدة وتسمى هذه المجموعة **Controls Collection** والتي تعتبر جزء من مجال الأسماء **System. Collections** يتم إنشاء المجموعات **System. Collections** أوتوماتيكياً بعد أن تقوم بإضافة فورم (نموذج) لبرنامجك وعندما تقوم بإضافة كائنات إلى هذا الفورم تكون هذه الكائنات تلقائياً جزءاً من المجموعات **System. Collections** وتقوم بيئة التطوير بالتعامل مع هذه المجموعات بنفس الطريقة التي تتعامل بها مع المصفوفات حيث أول عنصر فيها يبدأ من الصفر وهكذا وعندما نتعرف على المجموعات، نستطيع أن نتعامل مع بيئة التطوير ومع الكائنات بداخل الفورم وكذلك نعرف كيفية إضافة كائنات معينة (أزرار أو صناديق نص) إلى الفورم بواسطة الكود وبدون استخدام الطريقة البدائية (صندوق الأدوات) وتستطيع استعراض الكائنات التي استخدمتها في برنامجك بواسطة هذه المجموعات وبدعم من بيئة التطوير

## فهرسة الكائنات في المجموعات

نستطيع فهرسة الكائنات في المجموعات **Collection** أو فهرسة عناصر هذه المجموعات بمعرفة رقم كل كائن في فهرس المجموعة وللعلم يقوم الفيجوال بيسك 2008 بعمل فهرس عكسي لكل كائن على الفورم فأخر كائن تمت إضافته إلى الفورم يعتبر العنصر رقم صفر (0) في مصفوفة مجموعة الكائنات وعليه بمعرفة تسلسل إضافة الكائنات للفورم نستطيع فهرسة تلك الكائنات ويمكننا بناء على ذلك كتابة أكواد لتغيير خواص هذه الكائنات

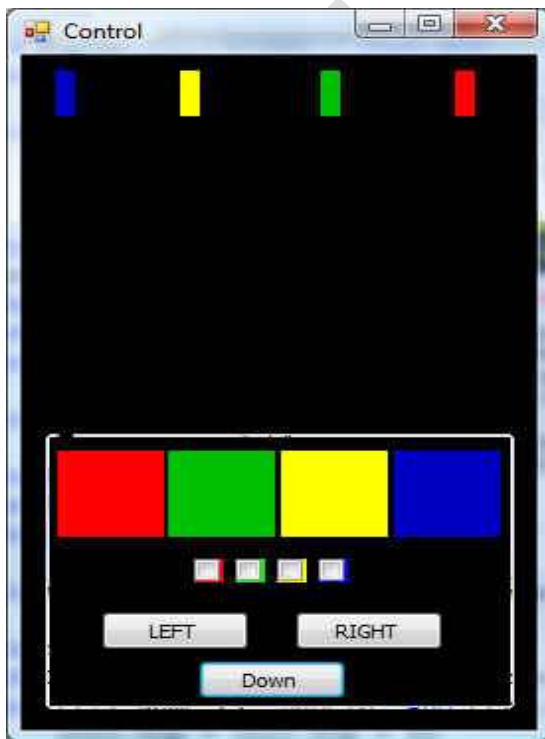
فمثلاً لو أردنا تغيير الخاصية **Text** التابعة للكائن صفر (0) وهو تبعاً لفهرسة المجموعة يكون آخر مكون تمت إضافته إلى النموذج يكون الكود كالتالي

```
Controls(0).Text = "Mohamed"
```

ولكن لو تم إضافة كائن آخر بالنموذج بعد هذا الكائن يقوم الفيجوال بيسك بتعديل فهرسة المجموعة بحيث أن يتم تغيير رقم هذا المكون إلى (1) ويتم إعطاء التسلسل (0) إلى آخر مكون وهكذا... ولفهم أكثر لطريقة الفهرسة العكسية للمجموعات في الفيجوال بيسك 2008 حمل التمرين التالي وقم بتشغيله ثم قم بإضافة **Label2** جديد فيه وقم بتشغيله مرة أخرى لتلاحظ الفرق بنفسك فقد تغير اسم **Label1** والموجود بالنموذج إلى **Label1** وتغير اسم **Label2** والذي قمت أنت بإدخاله إلى **"Mohamed Abou Elela"** وذلك بناء على الكود الموجود في **Form1** والخاص بالمكون رقم صفر (0) وهو آخر مكون تم إدراجه بالنموذج

استخدام الحلقات التكرارية **For Each...Next** للتعامل مع المجموعات

نستطيع التعامل مع الكائنات بداخل المجموعات كل على حده ولكن من الأفضل التعامل معها جميعاً باستخدام الحلقات التكرارية لأننا قد نحتاج إلى تغيير أسماء الكائنات أو إلى تحريك الكائنات على الفورم أو إلى ترتيب أو تغيير الأبعاد بشكل دفعه واحدة لكل الكائنات بالفورم وتنفيذ مثل هذه الأوامر نستخدم حلقة تكرارية خاصة وهي **For Each...Next** للتعامل مع كل الكائنات بداخل المجموعة مرة واحدة والحلقة التكرارية **For Each...Next** مثل الحلقة المعروفة **For .....Next** وباستخدامها نستطيع تعديل خواص الكائنات الموجودة ضمن المجموعة مثل إظهار أو إخفاء الكائنات وكذلك تفعيل أو إلغاء أو تحريك الكائنات أو إظهار قائمة بأسماء الكائنات وغيرها



وسوف نقوم بإعطاء مثال بسيط على كيفية التعامل مع المجموعات بتغيير خواصها أو بتحريكها أو تحريك أحدها في أي اتجاه في النموذج وسوف نقوم بعمل نموذجية مجموعة من أزرار وسوف نقوم بكتابة الأكواد في كل منهما راجع الأكواد من خلال التمرين (a,b) تجدها واضحة وبسيطة لا تحتاج إلى شرح فقط حاول تفسيرها معتمداً على ما سبق وقمنا بشرحه وبناء على المثالين (a,b) حاول تطبيق ما تعلمته على التمرين الثالث والثلاثون بالتحكم في الأكواد للوصول إلى الهدف المرجو منها استخدم **GroupBox** لتحديد المجموعات وهذا يقلل من استخدامك الأكواد فيمكن ضم مجموعة من الكائنات في صندوق مجموعة واحد **GroupBox** وتنفيذ الكود على صندوق المجموعة نفسه كما تم في هذا التمرين ولك حرية الابتكار

تحميل التمرين الثالث والثلاثون

إنشاء المجموعات الخاصة بك **Your Own Collections**

عرفنا أن الفيچوال بيسك 2008 يقوم بإنشاء المجموعات **Collections** تلقائياً عند فتح الفورم وإضافة الكائنات لها و تستطيع أنت أن تنشئ المجموعات الخاصة بك فيمكنك إنشاء المجموعات التي تقوم بمتابعة البيانات في البرنامج والتعامل معها بشكل أوتوماتيكي وبالرغم من إن المجموعات تقوم بحفظ الكائنات تستطيع جعل مجموعتك تحفظ الكائنات وكذلك القيم النصية والرقمية عند تنفيذ البرنامج ولاحظ ان طبيعة هذه المجموعات هي نفس طبيعة المصفوفات التي تعرفنا عليها في الفصل السابق

## تعريف مجموعة جديدة

يتم تعريف المجموعات الجديدة كأنها متغيرات جديدة في البرنامج ويتم تحديد قدرات المجموعة التي قمت بتعريفها بمعرفة المكان الذي قمت بتعريف المجموعة فيه فبإمكاننا أن نقوم بتعريف هذه المجموعات في بداية الكود للفورم بعد **Public Class Form** أو في أي مكان آخر و لكن من الأفضل في بداية الفورم على أن تكون القاعدة العامة لها هي كالتالي



بعد تعريف المجموعة نستخدم الطريقة **Add** لإضافة عناصر جديدة إلى المجموعة ويمكننا التعامل مع عناصر المجموعة بواسطة الحلقة التكرارية **For Each...Next**



وسوف ندرس معاً مثلاً على طريقة التعامل مع المجموعات التي نقوم بإنشائها و في هذا المثال سنصمم تطبيق صغير يقوم بفتح وصلات لمواقع الإنترنت من خلال المتصفح الافتراضي في جهاز الكمبيوتر لديك و سنستخدم المجموعة التي سنقوم بتعريفها لحفظ عناوين تلك المواقع التي قمنا بزيارتها على أن يتم إدخال الموقع المراد زيارته في **Textbox** يتم إضافته بالطريقة **Add** إلى المجموعة المختارة فيتم عمل مشروع جديد وبعد تصميمه يتم كتابة الأكواد كالتالي

1. نقوم بتعريف مجموعتنا الجديدة في أعلى منطقة الكود في الفورم تحت **Public Class** كالتالي

```
Public Class Form1
    Dim Myweb As New Collection ()
```

اسم المجموعة الجديدة هو **Myweb** وتعرف على أنها مجموعة جديدة **Collection()**

2. ثم نقوم بكتابة الكود التالي في **Button1** والمسمى **Visit** والخاص بتنفيذ أمر تصفح الموقع المكتوب في **Textbox**

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    myweb.Add(TextBox1.Text)
    System.Diagnostics.Process.Start(TextBox1.Text)
End Sub
```

يتم إضافة اسم الموقع الموجود في **TextBox1** الى المجموعة **myweb**

هنا يتم كتابة الكود الخاص بفتح الموقع الموجود في **TextBox1** وتسجيله

3. ثم نقوم بكتابة الكود التالي في **Button2** والمسمى **History** وهو الذي يظهر لنا المواقع التي تم تصفحها من خلال رسالة **Msgbox**

هنا يتم تعريف المتغيرين **webname** و **Allsite** كمتغيرات نصية بالمجموعة

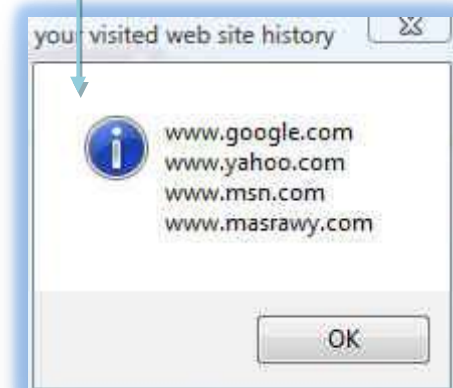
```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Dim webName As String = "", Allsite As String = ""
    For Each webName In myweb
        Allsite = Allsite & webName & vbCrLf
    Next webName
    MsgBox(Allsite, MsgBoxStyle.Information, "your visited web site history")
End Sub
```

يتم إسناد المواقع إلى المتغير **webname** والذي بدوره يسند كل هذه المواقع إلى المتغير **Allsite**

هنا يتم تنسيق صندوق الرسائل **Msgbox** والتي يظهر بها المواقع التي تم زيارتها

استخدام الحلقة **For Each ..... Next** وذلك لكل موقع **webname** في المجموعة **myweb**

تحميل التمرين الرابع و الثلاثون



استعراض الملفات النصية والتعامل معها من أسهل الطرق لفتح الملفات النصية هي فتح الملف النصي بداخل صندوق النص وإذا كان الملف النصي كبيراً نضيف أشرطة تمرير **Scrollbars** لصندوق النص حتى يستطيع المستخدم قراءة الملف النصي كاملاً و للتعامل مع الملفات النصية يجب أن نعرف أربع دوال تسهل لنا عملية التعامل مع تلك الملفات هذه الدوال هي

وظيفة	الدالة
تقرأ سطر من سطور الملف النصي	LineInput
تبحث عن نهاية الملف النصي	EOF
تقوم بإغلاق الملف النصي	FileClose
وتقوم هذه الدالة بفتح الملف النصي	FileOpen

فتح الملفات النصية الملفات النصية هي الملفات التي تحتوي على مجموعة من النصوص والأسطر والكلمات ويكون امتداد هذا النوع من الملفات هو كالتالي ( **txt, ini, inf, log, doc, docx** ) وتحتوي هذه الملفات على نصوصه مرتبة بشكل مميز نستطيع قراءتها من خلال صناديق النص **Textbox** ولفتح الملفات النصية نستخدم نافذة الحوار **OpenFileDialog** التي تفتح لنا الملفات ثم نقوم بوضع فلتر لتحديد نوع الملف النصي على حسب امتدادا الملف المطلوب عرضة فإذا كان تحديد الفلتر (الامتداد) **txt** لتقوم النافذة بإظهار الملفات النصية فقط، ثم نختار الملف المحدد ونوافق عليه فتقوم النافذة **OpenFileDialog** بحفظ مسار الملف لنا نستطيع استخدام هذا المسار لفتح الملف

#### الدالة FileOpen

بعد معرفة مسار الملف النصي بواسطة الخطوة السابقة نقوم باستخدام الدالة **FileOpen** لفتح الملف والطريقة العامة لهذه الدالة

**FileOpen ( filename, pathname, mode )**

الدالة المستخدمة لفتح الملف

وهو رقم الملف ويستخدم من 1 إلى 255 ويتم كتابة فقط عدد الملفات التي يتم التعامل معها

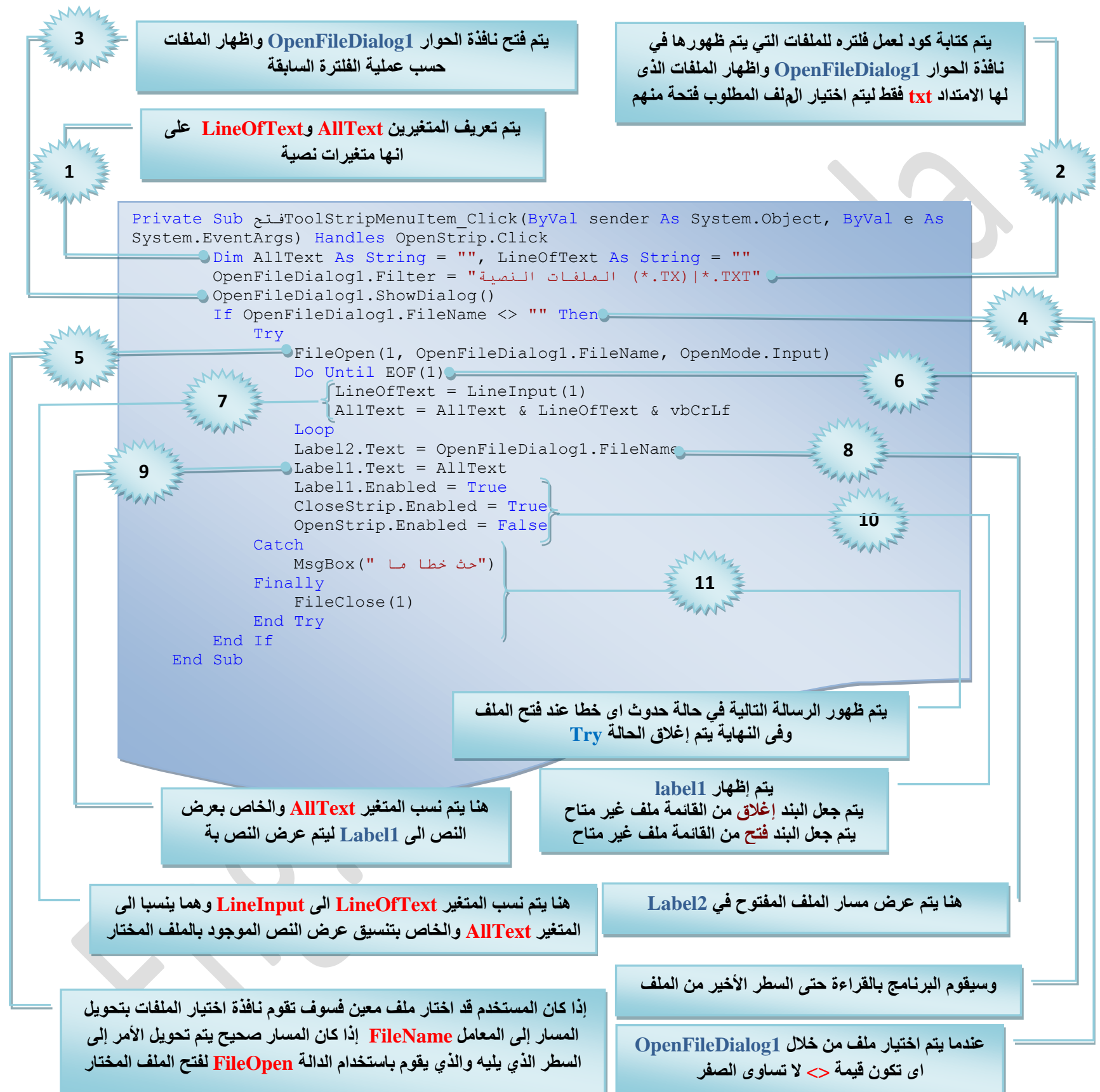
هو مسار الملف الذي اخترناه له وسيظهر لنا في نافذة حوار **OpenFileDialog1**

وهو وضع استخدام الملف وهو إما وضع مدخلان **OpenMode.Input** أو وضع مخرجات **OpenMode.output**

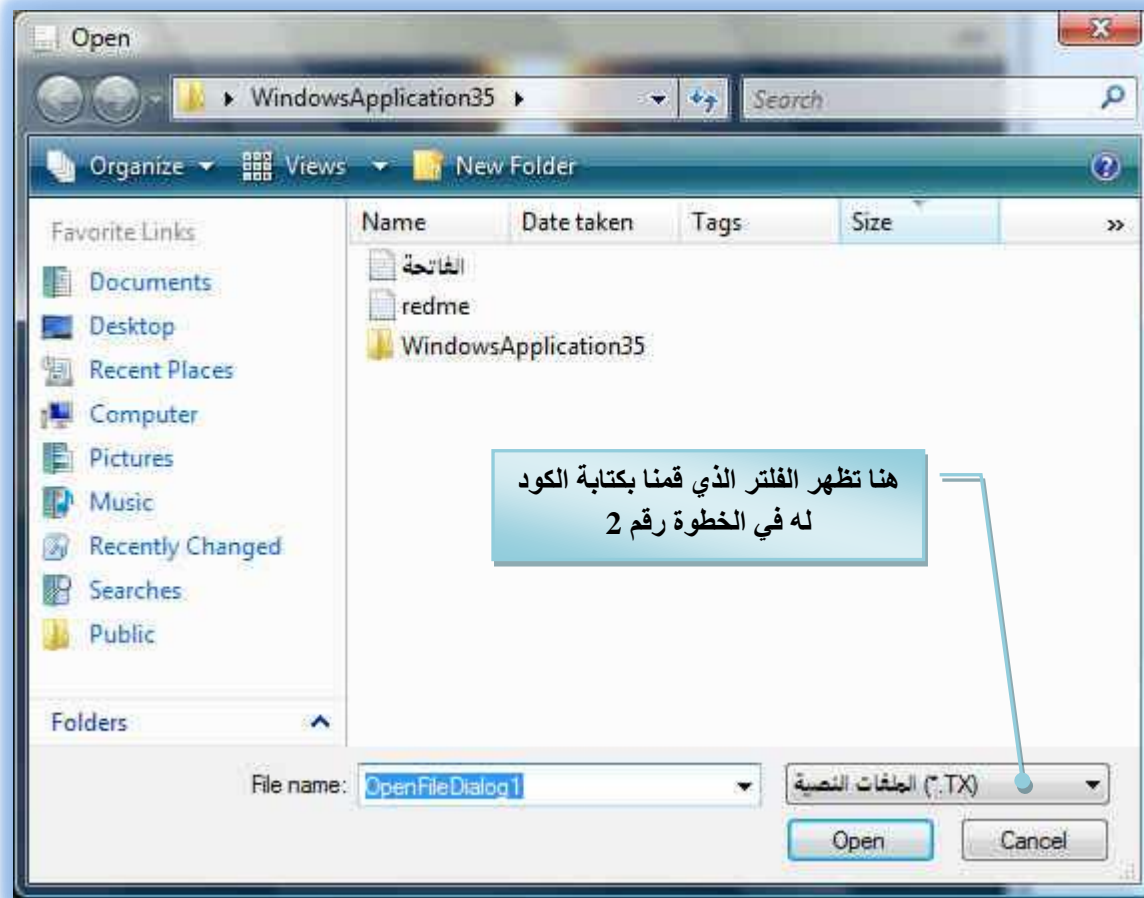
ولعمل تطبيق على الكود السابق سوف نقوم معا بعمل مشروع يقوم بالتعامل مع الملفات النصية ذات الامتداد **txt** وسوف نقوم في هذا التمرين بعمل برنامج يمكننا من خلاله فتح اي ملف له الامتداد **txt** وموجود على الكمبيوتر لدينا فبعد تصميم النموذج كما بالشكل أو حسب اختيارك مع ملاحظة إن التمرين يحتوي على **MenuStrip1** لعمل شريط قوائم من خلالها و **OpenFileDialog1** لاستخدامها لإظهار نافذة الحوار كما تعلمنا سابقاً وأيضاً يتم إدراج عدد 2 **Label** احدهما يظهر بس مسار الملف على الكمبيوتر والأخر يظهر به النص الموجود داخل الملف المختار قراءته و يمكنك تغييره بـ **Textbox** في حالة عرض نصوص كبيرة وتحتاج إلى شريط تمرير أفقي لقراءة النص كله بعد التنسيق وعمل القوائم اللازمة نقوم بكتابة الأكواد كالتالي



1. كتابة الكود الخاص بالبند فتح وهو عند النقر عليه يتم فتح نافذة الحوار **OpenFileDialog1** ليتم اختيار الملف الذي له الامتداد **Txt** فقط من خلاله وعرضه في **label1** أو **TextBox1** حسب تصميمك وإظهار مساره في **Label2** كما بالشكل



بعد الانتهاء من كتابة الأكواد بالطريقة السابقة يتم كتابة الكود للبند فتح من القائمة ملف يتم إظهار نافذة **OpenFileDialog1** لفتح اختيار ملف الـ **Txt** المفروض عرضه نصه في **Label1**



2. كتابة الكود الخاص بالبند إغلاق وهو عند النقر عليه يتم مسح النص من **label1** وجعل البند **فتح** نشط ويمكن استخدامه لفتح ملف **txt** مرة أخرى ويكون الكود فيه كالتالي

يقوم بمسح النص الموجود في **Label1**

```
Private Sub إغلاقToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CloseStrip.Click
    Label1.Text = ""
    Label2.Text = "لفتح الملفات النصية استخدم الانر فتح"
    CloseStrip.Enabled = False
    OpenStrip.Enabled = True
End Sub
```

يتم جعل البند إغلاق غير نشط  
يتم تنشيط البند فتح من القائمة ملف

يقوم بإظهار الرسالة في **Label2**

3. كتابة الكود الخاص بالبند خروج وهو عند النقر عليه يتم إغلاق البرنامج والخروج

```
Private Sub خروجToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ExitStrip.Click
    End
End Sub
```

تحميل التمرين الخامس و الثلاثون

إنشاء ملف نصي جديد وحفظه على جهازك

قد نحتاج لإنشاء ملف نصي وحفظه على جهازك للعديد من الأسباب منها حفظ إعدادات برنامجنا أو إعداد تقارير معينة أما طريقة حفظ النصوص في ملف نصي فتكون كالتالي

1. أخذ المدخلات النصية من المستخدم وإسنادها إلى متغيرات معينة
2. السماح للمستخدم من تحديد مكان حفظ الملف النصي باستخدام نافذة حوار **SaveFileDialog**
3. استخدام المسار الذي يحدده المستخدم لحفظ البيانات النصية فيه لفتح الملف
4. استخدام الدالة **PrintLine** لحفظ البيانات النصية إلى الملف المفتوح
5. بعد إكمال الحفظ نقوم بإغلاق الملف المفتوح بواسطة الدالة **File.Close**

سوف نقوم بعمل تمريننا هذا على أطلال المشروع السابق فقد استبدلنا **Label1** بـ **RichTextbox1** وذلك حتى نستطيع أن نتعامل مع



الملفات النصية الكبيرة والتي تحتوي على عدد من الجمل والكلمات الكثيرة بدون مشاكل كما قمنا باستبدال شريط القوائم في المشروع السابق بشريط الأدوات وتفعيل الأزرار الأساسية لشريط الأدوات مثل ( جديد وفتح وحفظ وقص ولص وخروج وأيضا مساعدة ) والذي سوف نتعرف على الأكواد الخاصة بهم جميعا في هذا المشروع ولكن الجديد في هذا المشروع أننا قد استعنا بأداة جديدة **SaveFileDialog** وهي الخاصة بفتح نافذة حفظ الملفات على جهازك فهي شبيهة بالأداة **OpenFileDialog** التي تم استخدامها سابقا لفتح الملفات ولكن هذه الأداة تستخدم لحفظها مع ملاحظة أنه قد تم استخدام الكود السابق في التمرين الخامس والثلاثون والخاص للبند فتح هناك للأداة فتح هنا والكود الخاص بالبند إغلاق هناك لأداة جديد هنا والكود الخاص بالبند بخروج هنالك إلى الأداة خروج هنا وسوف نتعرف معا في التمرين التالي على كتابة الكود لباقي الأدوات في شريط الأدوات الجديد ولكن بعد عمل النموذج التالي وتنسيقه كما تراه مناسباً للمشروع وقم بإضافة شريط الأدوات له وكتابة الأكواد في الأداة ( فتح و خروج و جديد ) كما سبق من التمرين السابق من البنود ( فتح و خروج و إغلاق ) على التوالي والآن نأتي لمراحل كتابة الكود في شريط الأدوات للنموذج الجديد كالتالي

1. كتابة الكود في الأداة **Help** مساعدة

```
Private Sub Help_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles Help.Click
    MsgBox("معلومات حول البرنامج", , "إعداد وتصميم مهندس محمد أبو العلا")
End Sub
```

وستخدم الأداة **Help** وذلك لإظهار معلومة حول البرنامج نفسه أو مصممة وأعتقد ان الجميع يعرفها من خلال تعامله مع البرامج الشهيرة والكود هنا عبارة عن كود لصندوق الرسائل سهل جدا وقد تعاملنا معه من قبل



2. كتابة الكود للأداة (Copy, cut, paste) وهو كود متشابه في جميع الحالات لأنه يطبق على النص الموجود داخل Textbox سواء كان الإجراء هو (قص أو لص أو نسخ) ويكون الكود كالتالي

```
Private Sub CutToolStripButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cut.Click
    TextBox1.Cut()
End Sub
```

يتم استخدام الكود  
 TextBox1.Copy() في حالة النسخ  
 TextBox1.Paste() في حالة اللصق  
 TextBox1.Cut() في حالة القص

3. ونأتي في هذه المرحلة إلى كتابة الكود في الأداة حفظ (Save) ويكون كالتالي

2 يتم فتح نافذة الحوار SaveFileDialog1 وتسجيل اسم الملف لحفظه حسب عملية الفلتر السابقة

يتم كتابة كود لعمل فلتره لنوع الملف الذي سوف يتم حفظه من خلال نافذة الحوار SaveFileDialog1 وذلك لحفظ الملف بالامتداد txt فقط

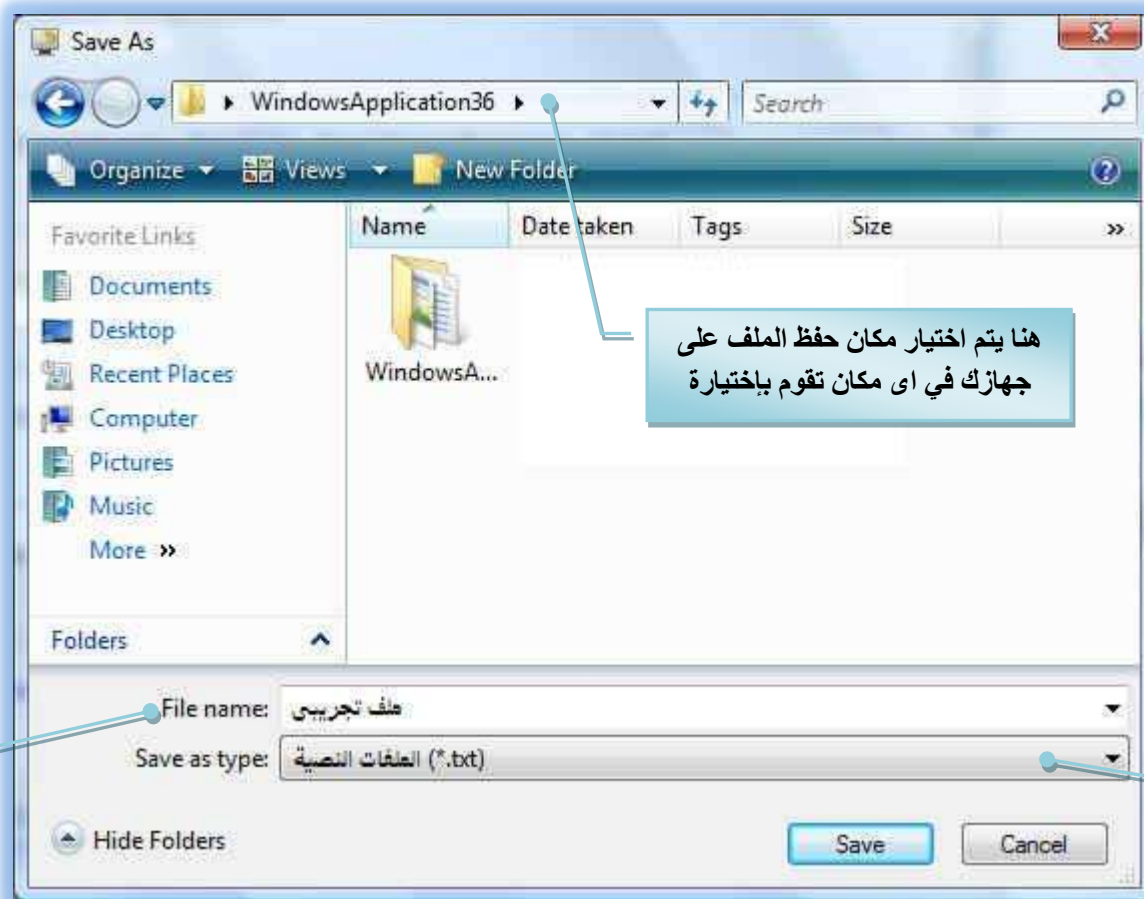
```
Private Sub Save_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Save.Click
    SaveFileDialog1.Filter = "النمىة الملفات (*.txt)|*.txt"
    SaveFileDialog1.ShowDialog()
    If SaveFileDialog1.FileName <> "" Then
        FileOpen(1, SaveFileDialog1.FileName, OpenMode.Output)
        PrintLine(1, TextBox1.Text)
        FileClose(1)
    End If
End Sub
```

3 عندما يتم اختيار ملف من خلال SaveFileDialog1 أى تكون قيمة <> لا تساوى الصفر

يقوم البرنامج بحفظ النص الموجود في صندوق النص بداخل الملف النصي باستخدام الدالة PrintLine

4 وعند اختيار المستخدم لمسار معين كما في الكود يقوم البرنامج بفتح الملف النصي الذي أختاره المستخدم (أو يقوم بإنشائه إذا لم يكن موجوداً)

وعند كتابة الكود السابق وتشغيل البرنامج F5 نقوم بكتابة أى نص في مكان كتابة النصوص وهو TextBox1 وبعد الانتهاء منة يتم حفظه بالنقر على الأداة حفظ (save) لتظهر لنا النافذة الحوارية التالية



هنا نقوم لكتابة اسم الملف المراد حفظه وقد اخترت حفظة باسم ( ملف تجريبي )

هنا يتم ظهور اسم الامتداد Txt الذي سوف يحفظ بيه الملف حسب إعداداتنا أثناء كتابة كود الفلتر في الأداة حفظ

ويمكننا فتحه مرة أخرى والتعامل معه بالإضافة أو المسح وهكذا كأى برنامج يتعامل مع النصوص Txt موجود على الكمبيوتر لدينا

### تحميل التمرين السادس و الثلاثون

#### كيف نستخدم النماذج

يعطينا الفيجوال بيسك الخيار في التعامل مع النماذج فبإمكاننا عرض جميع النماذج في وقت واحد وبإمكاننا عرض كل نموذج في وقت الحاجة إليه ونستطيع عرض أكثر من نموذج كما يمكن أن نتحكم في المستخدم بحيث يستخدم نموذج معين أولاً دون غيره ثم نسمح له باستخدام نموذج معين أو بقية النماذج و إذا كان لدينا نموذجين و أردنا فتح نموذج معين للمستخدم و لا نريد أن نسمح له باستخدام النموذج الآخر فيمكننا استخدام ShowDialog أما إذا أردنا السماح للمستخدم باستخدام النموذجين في نفس الوقت فنستخدم Show

لاحظنا من خلال التمرين السابق إننا استخدمنا ميزة جديدة وهي الأداة Help وهنا قد أضفنا من خلالها رسالة بسيطة تظهر عند النقر عليها وهي للتوضيح فقط ولكن في الأساس وفي المشروعات الكبيرة تجد هذه الأداة قد تحتوي على نموذج آخر يكون به معلومات كثيرة وبعض البرامج يوجد بها شرح للبرنامج نفسه ويظهر لنا عند النقر على هذه الأداة البسيطة فيمكن من خلالها فتح نموذج آخر يتضمن كل ما نريد أن نقولك للمستخدم عن هذا البرنامج من اتفاقية استخدام لإرشادات تعامل معه كما سوف نقوم بتطبيقه الآن من خلال اختيارنا لأي مشروع من المشاريع السابقة وسوف أقوم هنا باختيار المشروع الثالث والثلاثون وسوف أقوم بإضافة هذه الأداة عليه وسوف أقوم بتوجيهها لفتح نموذج آخر تظهر به المعلومات التي نريد إضافتها إلى التمرين

نلاحظ إن كل تعاملاتنا السابقة كانت مع نموذج واحد فقط وألآن سوف نقوم بالتعامل مع أكثر من نموذج ولهذا ينبغي علينا لتوفير الأكواد أن نقوم باستخدام الـ Module وذلك بدون نقاش ( اعتبروها قاعدة ) لكي نقوم بكتابة تعريفات المتغيرات المستخدمة في المشروع بها كما تعلمنا من قبل



1. في النموذج **Control** بعد إضافة الأداة **Help** نقوم بكتابة الكود التالي بها

```
Private Sub HelpToolStripButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles HelpToolStripButton.Click
    My.Forms.Form2.ShowDialog()
End Sub
```

هنا يتم استخدام الكود **ShowDialog** لظهور النموذج الاخر وذلك لاننا لا نريد ان يقوم المستخدم باستخدام النموذجين في نفس الوقت

عند التعامل مع النموذج على هذا الأساس يجب أن نكتب في الكود **ShowDialog** وهو يجعلنا نتعامل مع النموذج الثاني على أنه (صندوق حوار) ولهذا فسيظهر النموذج على أساس أنه نافذة حوار وبسبب ذلك لن يستطيع المستخدم الرجوع إلى النموذج الأولي إلا بعد أن يغلق النموذج الحواري أولا بالموافقة علي شروطه وهذه هي كل التعديلات التي نقوم بإدخالها على النموذج الأول **Form1** والمسمى **Control**

2. يتم إضافة النموذج الثاني **Form2** ونقوم بتغيير خاصية الاسم له إلى **Help** وإدخال زر **Button** و **Textbox** به ويتم كتابة الأكواد به كالتالي

A. كتابة كود الزر **Button** والمسمى بـ **Ok** ويكون كالتالي

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Me.DialogResult = DialogResult.OK
End Sub
```

نبهنا سابقا بان البرنامج يقوم بالتعامل مع النموذج على أنه صندوق رسالة وهنا يتم الموافقة على الرسالة بالزر **Ok** ويمكن طبعا التعديل عليه كما تعلمنا سابقا

B. كتابة الكود التالي في النموذج **Form2** نفسه وهو الكود الخاص بفتح ملف ( التعليمات أو الاتفاقية أو المساعدة ) اي كان المعلومة المراد إظهارها في النموذج **Help** فبعد فتح النموذج سيقوم هذا النموذج بفتح ملف نصي في التعليمات ليقرأه المستخدم، وسنستخدم الفئة **StreamReader** ولذلك لا بد من إضافة مجال الأسماء الخاص به أولا في أول الكود

```
Imports System.IO
Public Class Form2
```

يتم تعريف المتغيرات **StreamReader** و **StreamToDisplay**

استيراد مجال الأسماء **system.IO** سيسهل لنا التعامل مع الفئة **StreamReader**

```
Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim StreamToDisplay As StreamReader
    StreamToDisplay = _
        New
        StreamReader("C:\Users\mohamed\Desktop\WindowsApplication33\help.txt")
    TextBox1.Text = StreamToDisplay.ReadToEnd()
    StreamToDisplay.Close()
    TextBox1.Select(0, 0)
End Sub
```

هنا يتم إزالة اي تحديد في النص في حالة وجوده

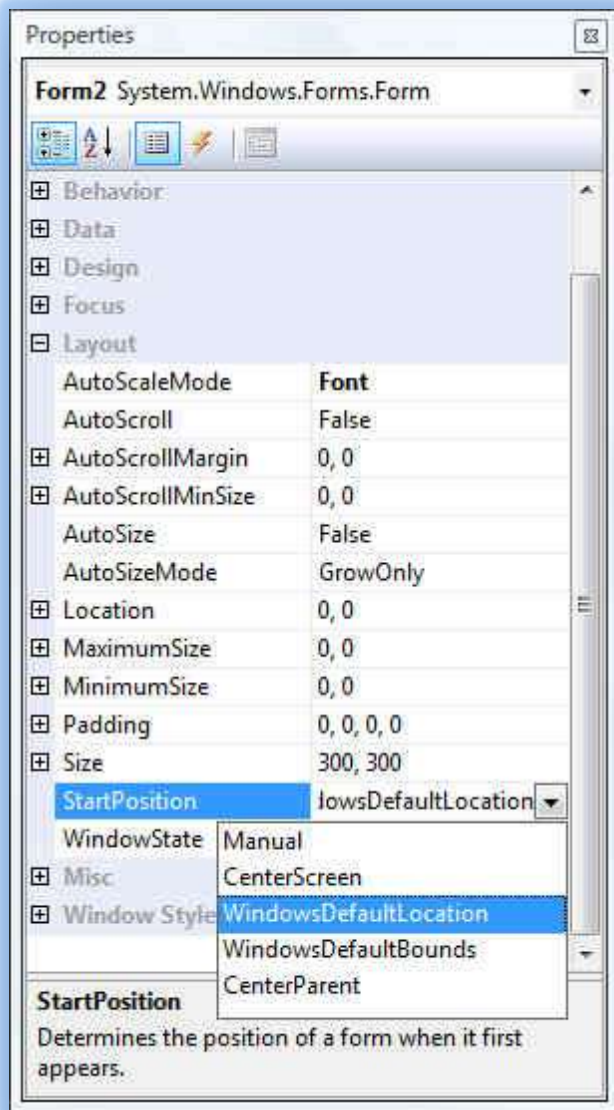
هنا يتم إسناد المسار السابق في المتغير **StreamReader** لكي يتم قراءته في **Textbox1** حتى نهايته

هنا كود إغلاق الملف النصي بعد نقل كل محتوياته إلى **Textbox**

هنا يتم كتابة المسار للملف الذي سوف يظهر في المتغير **StreamReader**

تحميل التمرين السابع و الثلاثون

نقوم الآن بتنفيذ البرنامج **F5** ونأكد من وجود الملف النصي **Help** في مساره الصحيح ولضمان ذلك قم بضغط التمرين المرفق على سطح المكتب وقم بتغيير اسم **Mohamed** من كود المسار باسم جهازك أنت

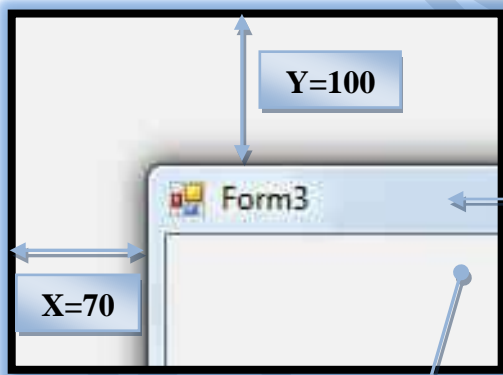


### تحديد موقع النماذج على سطح المكتب

بعد تصميم البرنامج الذي قد يحتوي على أكثر من نموذج يمكننا التحكم في مكان إظهار كل نموذج فقد نحتاج لأن يظهر النموذج الأول في وسط الشاشة ونموذج آخر في أعلى الشاشة والنموذج الثالث في وسط النموذج الثاني وهكذا أو قد نحتاج لتغيير مكان ظهور نموذج معين إذا اختار المستخدم خيار معين (لاحظنا ذلك في بعض التمارين السابقة) ونحن نستطيع تغيير مكان النموذج على الشاشة بطريقتين

1. بواسطة صندوق الخصائص لكل نموذج ومنة نستطيع تحديد مكان النموذج باستخدام الخاصية **StartPosition** وذلك باختيارنا أحد هذه الخيارات

الخاصية	الهدف منها
CenterScreen	يقوم بوضع النموذج في وسط الشاشة وهذا الخيار مفضل عند كثير من المبرمجين
WindowsDefaultLocation	هذا الخيار الطبيعي المعتمد في حالة لم تقم بتحديد أحد الخيارات ويقوم بوضع النموذج في المكان الذي يراه نظام التشغيل مناسباً وعادة ما يكون النموذج في أعلى يسار الشاشة
WindowsDefaultBounds	هذا الخيار يقوم أولاً بتغيير طول وعرض النموذج ثم يقوم بوضعه في المكان الذي يقترحه نظام الويندوز وعادة ما يكون أعلى يسار الشاشة
CenterParent	فيعتبر مناسب للتطبيقات الكبيرة التي تحتوي على أكثر من نموذج ويقوم بوضع النموذج الأول في وسط شاشة ويكون هو النموذج الأب ويقوم بوضع النموذج الثاني في وسط شاشة النموذج الأب ويسمى بالنموذج الابن
Manual	وعند تحديد هذا الخيار لابد أن نذهب إلى الخاصية <b>Location</b> ونقوم بكتابة مكان ظهور النموذج بالأرقام هكذا



يتم قياس الأبعاد من حدود الشاشة والقياس هنا بالكسل

Location	70, 100
X	70
Y	100

2. بواسطة كتابة الأكواد وذلك باستخدام خاصية **DesktopBounds** وتسمى **DesktopBounds** ويكون كتابة في الزر المخصص لإظهار النموذج **Form2** بعد إضافته بالطرق السابقة الكود كالتالي

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim Formsize As New Rectangle(200, 100, 300, 250)
    form2.DesktopBounds = Formsize
    form2.Show()
End Sub
```

كود عرض النموذج **Form2**

يتم تعريف متغير جديد أسمة **Formsize** وية يتم كتابة إحداثيات وأبعاد النموذج الجديد

يتم إسناد الخاصية **DesktopBounds** الى المتغير **Formsize**

تعلمنا فيما سبق كيفية إضافة نموذج جديد ثم تعديل خواصه من خلال استخدام صندوق الخواص التابع له ولكن لو أردنا استخدام طريقة أكثر احترافية وهي غير ملزمة للجميع لكي نتمكن من خلالها بإدخال نموذج جديد إلى المشروع من خلال استخدام الأكواد و أيضا التحكم في جميع خواص النموذج الجديد وذلك من خلال الأكواد وهذا مثال على قدرتنا على إدخال نموذج جديد والتحكم في خواصه بكتابة الأكواد ويكون كالتالي

يستخدم الكود التالي بتحكم في الخاصية **BackColor** لون النموذج

هنا نقوم بإدخال متغير أسمة **Form2** ويعامل كنموذج

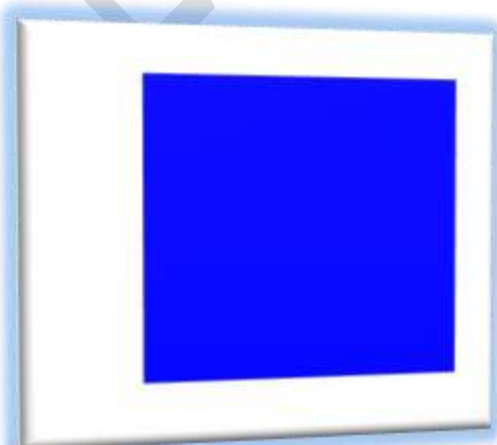
يستخدم الكود التالي بتحكم في الخاصية **Text** أسم النموذج

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim form2 As New Form
    form2.Text = "النموذج الثاني"
    form2.BackColor = Color.Blue
    form2.FormBorderStyle = Windows.Forms.FormBorderStyle.None
    form2.StartPosition = FormStartPosition.Manual
    Dim Form2Rect As New Rectangle(200, 100, 300, 250)
    form2.DesktopBounds = Form2Rect
    form2.ShowDialog()
End Sub
```

يستخدم الكود التالي بتحكم في الخاصية **StartPosition** موقع النموذج

يستخدم الكود التالي بتحكم في الخاصية **Manual** السابق اختيارها لموقع إظهار النموذج كما سبق وشرحنا الكود

يستخدم الكود التالي بتحكم في الخاصية **BorderStyle** إطار النموذج



بعد كتابة الأكواد كالسابق في الزر **Button1** في النموذج الأول وتشغيل البرنامج **F5** يقوم بإظهار النموذج **Form2** كالتالي ويكون أبعاده بالنسبة لحدود الشاشة كما تم وإدخالها في الأكواد و أبعاده أيضا كما تم وكتابتها في الأكواد والتقدير بالبكسل

تحميل التمرين الثامن و الثلاثون

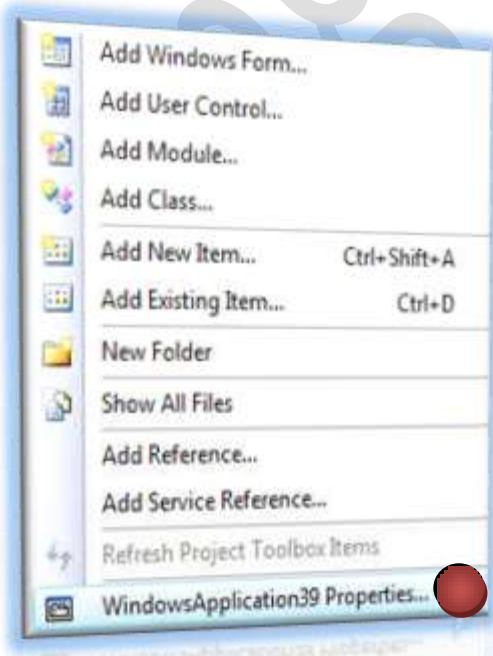
من خلال استخدامنا الأكواد السابقة وطريقة التحكم في المكونات تم إضافة نموذج إلى المشروع عن طريق الأكواد وليس ذلك فقط فقد قمنا بتحديد العديد من الخصائص التابعة له بواسطة الكود و تلك العملية حفزتنا لنقوم بالعديد من المهام بواسطة الأكواد فمثلا نستطيع إضافة الكائنات مثل ( أزرار، صناديق نص، وغيرها ) بواسطة الكود كما سريفيدنا هذا كثيرا في برامجنا الكبيرة إذا تم ربط تلك العملية بقواعد البيانات فسيكون برنامجنا متقدما ومثاليا جدا ونحن نستطيع إضافة الكائنات بواسطة الكود بتعريف الكائنات أولا ثم إضافتها للنماذج راجع الأكواد التالية

```
Public Class Form1

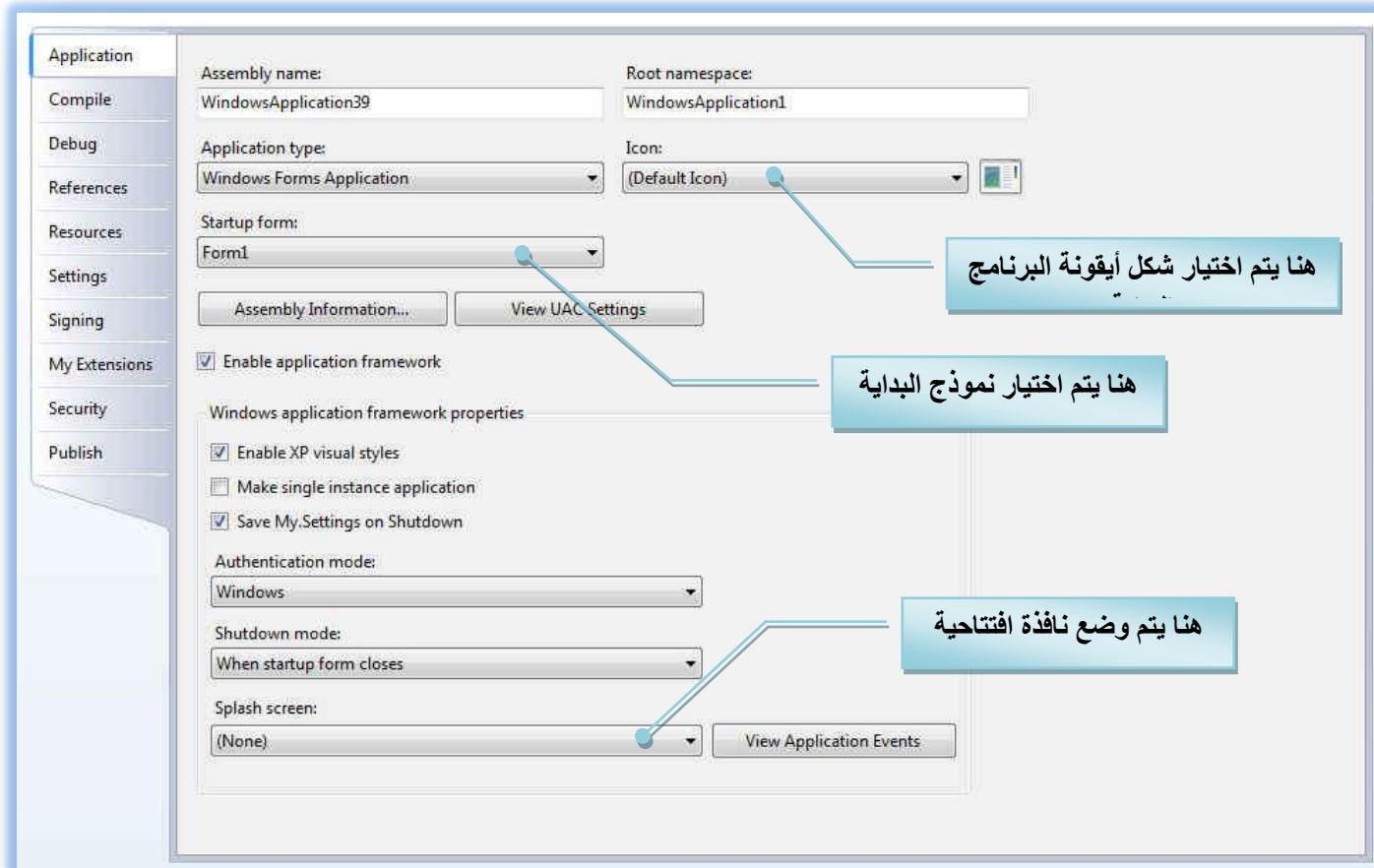
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
        احظ الفرق باستخدام الكود التالي وبين الكود السابق الخاص باضافة زر هنا نقوم بالتحكم في
        الزر في نفس النموذج وبذلك باستخدام الاداة me للتطبيق على نفس النموذج
        Me.Button1.Location = New Point(50, 50)
        Me.Button1.Text = "open form2"
    End Sub

    Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button1.Click
        Dim form2 As New Form
        Dim button1 As New Button
        form2.Text = "form2"
        form2.StartPosition = FormStartPosition.CenterScreen
        form2.MaximizeBox = True
        ' form2.MinimizeBox = True
        نستخدمها في حالة ان نريد تصغير الفورم في شريط المهام عند البداية
        form2.WindowState = FormWindowState.Maximized
        form2.Controls.Add(button1)
        الكود التالي هو لاضافة زر الى الفورم الثاني
        button1.Location = New Point(200, 300)
        button1.Text = "close"
        form2.ShowDialog()
    End Sub
End Class
```

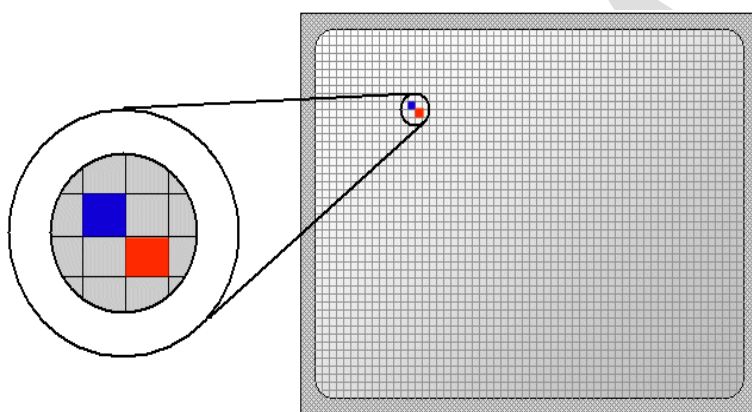
تحميل التمرين التاسع و الثلاثون



لاحظنا فيما سبق إن المشروع الواحد يمكن أن يحتوي على أكثر من نموذج ولكن لابد أن يكون لأي مشروع ناجح نموذج أساسي يعتبر هو الواجهة التي يمكن المستخدم من خلالها التعامل مع مشروعك ويمكن اختيار اي من نماذج المشروع ليكون هو واجهة المشروع ونموذج البداية بالطريقة التالية لتغيير النموذج الذي يظهر في بداية تشغيل التطبيق نذهب إلى القائمة **Project** من شريط قوائم البرنامج فيجوال بيسك 2008 ونقوم باختيار **( My Project Name Properties )** لتظهر لنا هذه النافذة التالية والتي نختار منها نموذج بداية التشغيل **F5**



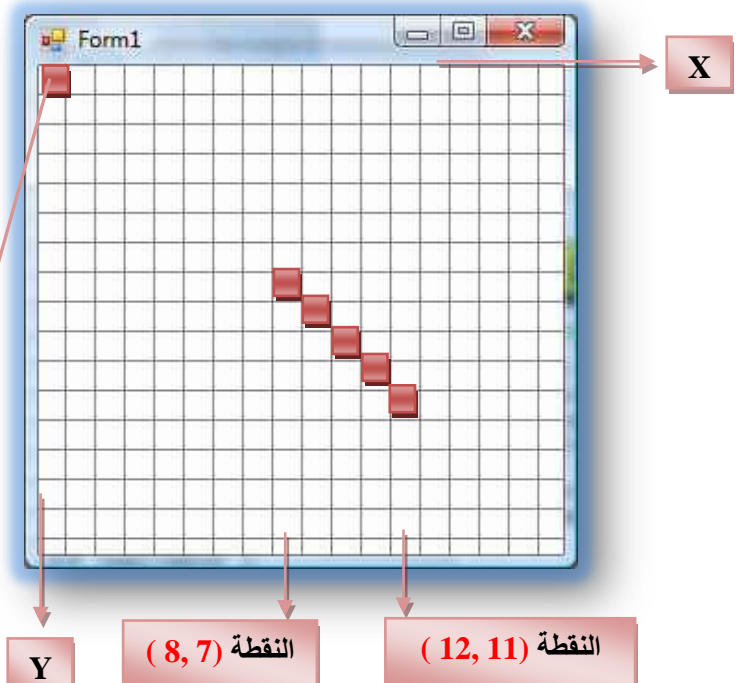
إضافة تأثيرات رسومية و متحركة



يتم التعامل مع الرسومات باستخدام مجال الأسماء **System. Drawing** فقد تعلمنا سابقًا كيف نقوم بإضافة الصور وغيرها من الرسومات إلى النماذج بواسطة **الفيجوال بيسك 2008** فقمنا بالتعامل مع صناديق الصور وغيرها والآن سنقوم بالتعامل مع ما يسمى بدوال الـ **GDI** الموجودة ضمن مجال الأسماء **System. Drawing** التي تأتي ضمن إطارات العمل الدوت نت **.NET** والتي نستطيع من خلالها التعامل مع الصور وكذلك رسم الأشكال الثنائية الأبعاد بداخل نظام الويندوز وينبغي علينا قبل دراسة الرسومات فهم طبيعة النموذج **Form layout** فكل نموذج له تصميمه الخاص حيث يبدأ هذا التصميم من نقطة معينة في أعلى يسار الشاشة وهذا التصميم يكون على شكل صفوف وأعمدة تتكون من نقاط صغيرة جدًا وهذا ما يسمى بالبيكسل **Pixel** هذه النقاط ترسم على المحورين الأفقي والعمودي

بينما يمثل المحور العمودي الرمز **Y**

عندما نحدد نقطة معينة على النموذج فهذه النقطة هي عبارة عن نقطة تلاقي المحور الأفقي مع المحور العمودي **(X,Y)** وعليه فأعلى نقطة في يسار النموذج تساوي القيمة **(0,0)** عند وجود نقطة واحدة فتعتبر نقطة فقط، لكن عند وجود مجموعة نقاط فهذا يعني أنه لدينا خط أو دائرة أو مستطيل أو غيرها من الأشكال الرسومية وعند تحديد نقطة البداية ونقطة النهاية يتم معرفة طول واتجاه الخط أو الشكل الرسومي المطلوب





يحتوي مجال الأسماء **System.Drawing** على العديد من الفئات التي تساعدنا على التعامل مع الرسومات في برنامجنا و سنتعرف الآن إلى **System.Drawing.Graphics** التي تعنى برسم الأشكال على النماذج و يمكنك معرفة بقية الفئات بالرجوع إلى التعليمات المرفقة بالفيجوال بيسك 2008 فللجدول التالي يحتوي على بعض الطرق **Methods** المتوفرة ضمن الفئة **Graphics**

الوصف	الطريقة	الشكل
لرسم خط يوصل بين نقطتين	<b>DrawLine</b> <b>FillLine</b>	خط مستقيم
لرسم مستطيل أو مربع يوصل بين أربع نقاط	<b>DrawRectangle</b> <b>FillRectangle</b>	مستطيل
لرسم قوس أو خط منحنى جزء من دائرة	<b>DrawArc</b> <b>FillArc</b>	قوس
لرسم شكل دائري أو ببيضاوي محدود بواسطة مستطيل	<b>DrawEllipse</b> <b>FillEllipse</b>	دائرة
لرسم مضلع وهو شكل يحتوي على العديد من الأضلاع والزوايا و تخزن قيمها بداخل مصفوفة	<b>DrawPolygon</b> <b>FillPolygon</b>	مضلع
هذا المنحنى نقوم بتحديد النقاط التي يمر فيها وتخزينها في مصفوفة بخلاف القوس في <b>DrawArc</b> حيث يتم تحديد نقطتين فقط	<b>DrawCurve</b> <b>FillCurve</b>	منحنى

تستخدم جميع الطرق السابقة لإضافة رسومات فارغة داخل النموذج ولكن في حالة أن نريد أن هذه الرسومات تكون ملونة تضيف البادئة **Fill** في بداية الطريقة بعد حذف **Draw** منها

فعند استخدام **System.Drawing.Graphics** لرسم الأشكال على النماذج لابد من استخدام ( كائن وسيط ) قلم **Pen** أو فرشاة **Brush** للرسم فعند رسم خط مستقيم أو شكل فارغ يمكننا استخدام القلم أما إذا أردنا رسم الأشكال المليئة بالألوان فلا بد من استخدام الفرشاة ولكن نستطيع أن نقوم بإتباع الخطوات التالية

1. نقوم بتعريف كائن الرسومات ويتم تعريف الأدوات فنستخدم القلم (الأحمر) أو نستخدم الفرشاة (الصفراء)

```
Dim GraphicsFun As Graphics
Dim BrushColor As New SolidBrush(Color.Yellow)
Dim PenColor As New Pen(Color.Red)
```

2. نقوم بتحديد مهمة الكائن وهي إنشاء الرسوم

```
GraphicsFun = Me.CreateGraphics
```

3. هنا يتم تحديد نوع الأداة التي تم تعريفها من قبل ونوع الرسم الذي سوف تقوم برسمه معرفا بإحداثيات كما تم وعرفنا من قبل

```
GraphicsFun.FillRectangle(BrushColor, 150, 10, 250, 100)
GraphicsFun.FillEllipse(BrushColor, 50, 150, 100, 80)
```

سوف نقوم الآن بعمل تمرين يقوم برسم مستقيم لونه احمر وإحداثياته هي نقطة البداية ( 20,30 ) و نقطة النهاية ( 100,80 ) عند النقر على زر **Button** معين فيكون كتابة الكود في الزر كالتالي

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim GraphicsFun As Graphics
    'يتم تعريف كائن الرسومات
    Dim PenColor As New Pen(Color.Red)
    'يتم تحديد نوع الاداة المستخدمة واللون المستخدم بها
    GraphicsFun = Me.CreateGraphics
    'يتم تحديد وظيفة كائن الرسومات
    GraphicsFun.DrawLine(PenColor, 20, 30, 100, 80)
    'هنا يتم تحديد نوع الشكل المراد رسمة واحداثياته في النموذج
End Sub
```

راجع التمرين للحصول على اكواد وطرق رسم أكثر

تحميل التمرين الأربعة



تصغير وتكبير كائن ما خلال مرحلة تنفيذ البرنامج

هل فكرت يوماً في تقريب صورة معينة من داخل برنامج عدة مرات لكي تراها بوضوح يمكننا ذلك من خلال كتابة كود في الفيچوال بيسك 2008 يقوم بزيادة ارتفاع وعرض صندوق الصورة بمقدار معين ولأن مساحة الصورة بداخل صندوق الصورة تكبر بشكل مطاطي بسبب الخاصية **SizeMode** فستكبر الصورة مع صندوق الصورة في نفس الوقت بنفس المقدار وكلما استمرينا في الضغط على الصورة فإنها ستكبر أكثر فأكثر بنفس القيمة وسنرى ذلك في التمرين التالي فبعد إضافة صندوق صورة إلى النموذج وتنسيق النموذج كما بالشكل يتم كتابة الكود التالي داخل صندوق الصورة كالاتي

هنا يتم تغيير ارتفاع صندوق الصور **PictureBox1** عند النقر على قيمة 15 نقطة ( بكسل )

```
Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click
    PictureBox1.Height = PictureBox1.Height + 15
    PictureBox1.Width = PictureBox1.Width + 15
End Sub
```

هنا يتم تغيير عرض صندوق الصور **PictureBox1** عند النقر على قيمة 15 نقطة ( بكسل )

وكما تعلمنا من قبل يمكن تغيير العرض والطول لـ **PictureBox1** بالزيادة والنقصان ( تكبير / تصغير ) وذلك بالتحكم في الاشارة (+/-)

تحميل التمرين واحد و أربعة

## إضافة فئة Class جديدة إلى مشروعك

الفئة **Class** هي عبارة عن وعاء حاضن لكائن برمجي أو أكثر يقوم محرر الكود بتلوينها باللون الأزرق في فيجوال بيسك 2008 و بعد تعريف الفئة وإضافة الكائن البرمجي إليها سيتضمن هذا الكائن خصائص وأحداث وطرق **Methods** مثل الكائنات التي نضيفها للنموذج ولإضافة فئة جديدة لبرنامجنا من مستكشف المشروع **Solution Explorer** نختار **Add** ثم **New Item** ثم نختار **Class** ثم نقوم بتعريف الفئة باستخدام الأكواد

وسنأخذ تمرين تطبيقي لإنشاء وتعريف فئة تحت اسم **Student** تقوم هذه الفئة بأخذ الاسم الأول والأخير مع تاريخ الميلاد للطالب وتقوم بحفظ البيانات في خصائص الفئة وسوف نقوم بإضافة طرق **Methods** لحساب عمر الطالب بمجرد معرفة تاريخ ميلاده كما سنتعلم عبر هذا التطبيق كيف نصمم فئة خاصة بنا وكذلك كيف نستفيد من الإجراءات التابعة لهذه الفئات في مرحلة الكود وبعد الانتهاء من تصميم النموذج كالشكل المطلوب نقوم بإضافة **Class** إلى المشروع ونقوم بتسميته **Student.vb** ليفتح سيقوم الفيجوال بيسك بإضافة الفئة إلى ملفات المشروع ثم سيقوم بفتح صفحة الكود الخاصة بالفئة وسنقوم الآن بكتابة الأكواد الخاصة بالفئة **Student** كالتالي

1. تعريف متغيرات الفئة **Student** وهما المتغير الخاصين بأدراج الأسماء إلى المشروع كالتالي

```
Public Class student
    Private Name1 As String
    Private Name2 As String
```

هنا نقوم بتعريف المتغيرات الخاصة بدراج لاسم الاول والثاني للطالب على انها متغيرات خاصة بالفئة (**student**) فقط

## 2. إنشاء الخصائص للمتغيرين السابقين وذلك بكتابة الكود التالي فقط

```
Public Property FirstName() As String
```

ثم الضغط على **Enter** ليقوم البرنامج باستكمال باقي الأكواد كالتالي

```
Public Property FirstName() As String
    Get
    End Get
    Set(ByVal value As String)
    End Set
End Property
```

والتي تعني ماذا سوف يرى المبرمج عند استخدام الخاصية **FirstName**

يحدد هنا ماذا يحدث إذا قام المبرمج بتغيير قيمة الخاصية **FirstName**

انتهاء كود الخاصية

ثم يتم استكمال الكود بإدخال التالي

```
Public Property FirstName() As String
    Get
        Return Name1
    End Get
    Set(ByVal value As String)
        Name1 = value
    End Set
End Property
```

يتم إضافة الخاصيتين **Return Name1** و **Name1 = value** حسب الكود

ويتم كتابة نفس الكود للخاصية **LastName** كالسابق

3. إنشاء الطريقة **Method** والخاصة لحساب العمر من خلال إدخال تاريخ ميلاد الطالب

```
Public Function Age(ByVal Birthday As Date) As Integer
    Return Int(Now.Subtract(Birthday).Days / 365.25)
End Function
```

إنشاء الطريقة **Method** في الفئة **Class** لابد من تعريف دالة **Function** بداخل الفئة ثم يتم إدخال المعاملات (المعادلة الحسابية والمتغيرات الخاصة بها)

4. ثم يتم كتابة الكود التالي في الزر **Button1** والموجود بنموذج المشروع كالتالي

يتم تعريف المتغيرات **newstudent** وانسابة الى الكلاس **student** و **birthday**

```
private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim newstudent As New student
    Dim birthday As Date
    newstudent.FirstName = TextBox1.Text
    newstudent.LastName = TextBox2.Text
    birthday = DateTimePicker1.Value.Date
    MsgBox("الطالب " & newstudent.FirstName & " " & newstudent.LastName & " هو عمرة " & newstudent.Age(birthday) & " سنة")
End Sub
```

يتم إسناد المتغير **birthday** الى **DateTimePicker1**

هنا يتم تنسيق **MsgBox** التي تظهر عند النقر على الـ **Button1**

يتم إسناد المتغير **newstudent** الى **FirstName** و **LastName**



وعند تشغيل **F5** البرنامج قم بكتابة الاسم الأول والأخير للطالب وإدخال تاريخ ميلاده وبالنقر على الزر تظهر لنا الرسالة التالية كما بالشكل

تحميل التمرين الثاني و أربعون

## قواعد البيانات وبرمجة إنترنت

في التمرين السابق استخدمنا نموذج بسيط وغير مرتبط بقواعد البيانات **Database** لذلك فهو يقوم بعرض سجل واحد فقط وسوف نتعامل الآن مع الجزء الخاص بقواعد البيانات لكي تستطيع الإلمام بهذا الجزء من الكتاب لابد من تعلمك أولا قواعد البيانات فلا فائدة من معرفة ما يلي بدون أن تكون على علم على الأقل بمصطلحات قواعد البيانات ويمكنك الاستعانة بكتاب أكسس 2007 والخاص بالشرح الوافي لقواعد البيانات من الروابط التالية كما إننا سوف نقوم بالاستعانة بالتمارين الموجودة بة لاستكمال شرحنا في الجزء الخاص بربط قواعد البيانات بنماذج الفيچوال بيسك 2008

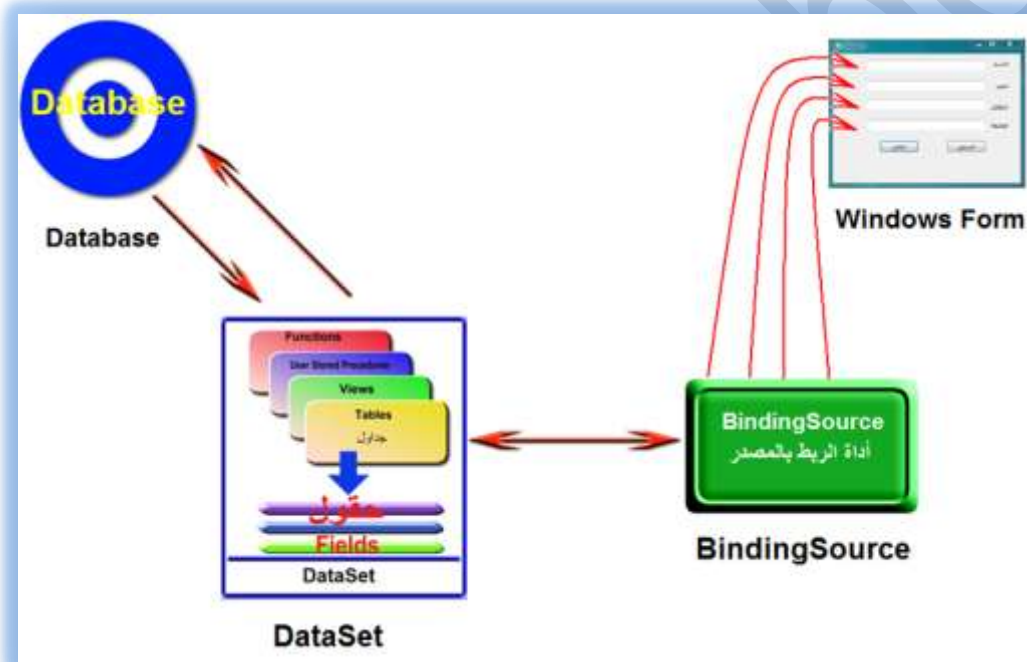
Microsoft Access 2007 Part 1  
Download

Microsoft Access 2007 Part 2  
Download

والآن سوف نتعلم معا كيف نتعامل مع البيانات المخزنة في قواعد البيانات أو في مواقع الإنترنت في البداية نستعلم كيفية التعامل مع قواعد البيانات باستخدام تقنية **ADO.NET** وهي تقنية هامة للتعامل مع قواعد البيانات وسوف نتعلم كيفية عرض البيانات وتعديلها والبحث فيها باستخدام بعض الأدوات وكذلك الكود في الفيچوال بيسك 2008 الذي تم فيه تحديث العديد من الأساليب من أجل التعامل الأمثل مع قواعد البيانات فلن نتعامل مع قواعد البيانات على أنها سجلات أو بيانات فقط ولكن هناك العديد من التقنيات للتعامل مع البيانات بكل احترافية فتقنية **ADO.NET** تستخدم أكثر من مكون للتعامل مع قواعد البيانات وهم

## 1. قاعدة بيانات Database

ويتم إنشائها من خلال برنامج الأكسس راجع كتاب أكسس 2007



تحميل قاعدة بيانات جاهزة للتطبيق

## 2. موصل Connection

وهو يحتوي على معلومات عن قاعدة البيانات ويحتوي على معلومات تستخدمها بقية المكونات للربط مع قاعدة البيانات إذا كانت قاعدة البيانات تحتوي على كلمة سر فهذا الموصل يحتوي على كلمة السر لقاعدة البيانات

## 3. عارض Dataset

وهو عبارة عن عارض لما تحتويه الجدول بداخل قاعدة البيانات

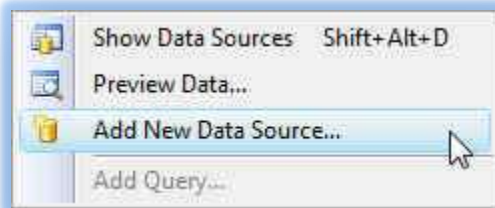
## أداة الربط Binding Source

أنها تقوم بتثبيت الاتصال الذي تم إنشاؤه وربط المعلومات بالفورم الذي ستقوم من خلاله بعرض البيانات

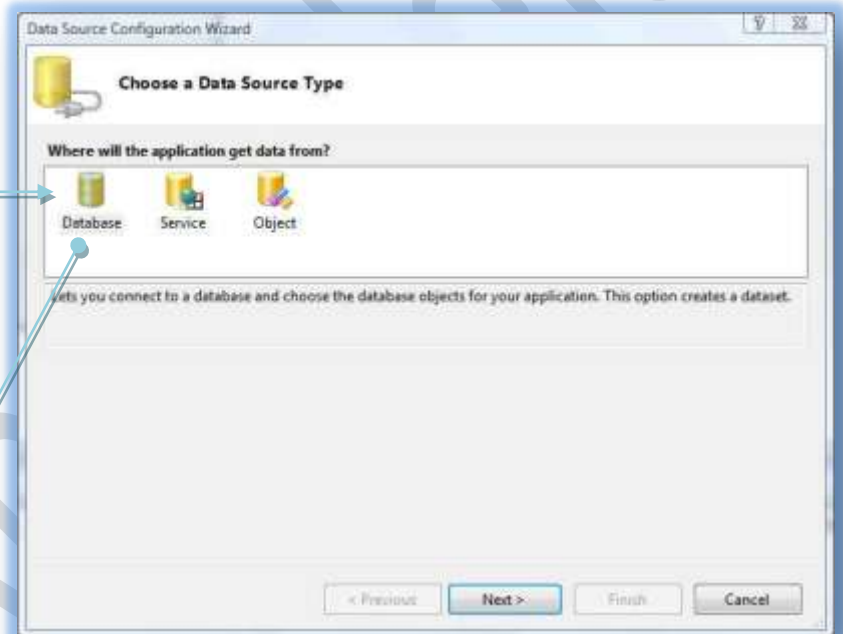
وبعد عملية الربط مع قاعدة البيانات تقوم محرك الربط مع قاعدة البيانات بإنشاء ملف **XML** ليسهل عملية الربط بين قاعدة البيانات وبين المكونات **Dataset** و **table adapter** و **data navigator** وكذلك تسهيل عملية التنقل بين البيانات أو تعديلها

## التعامل مع قاعدة بيانات مايكروسوفت أكسس Microsoft Access

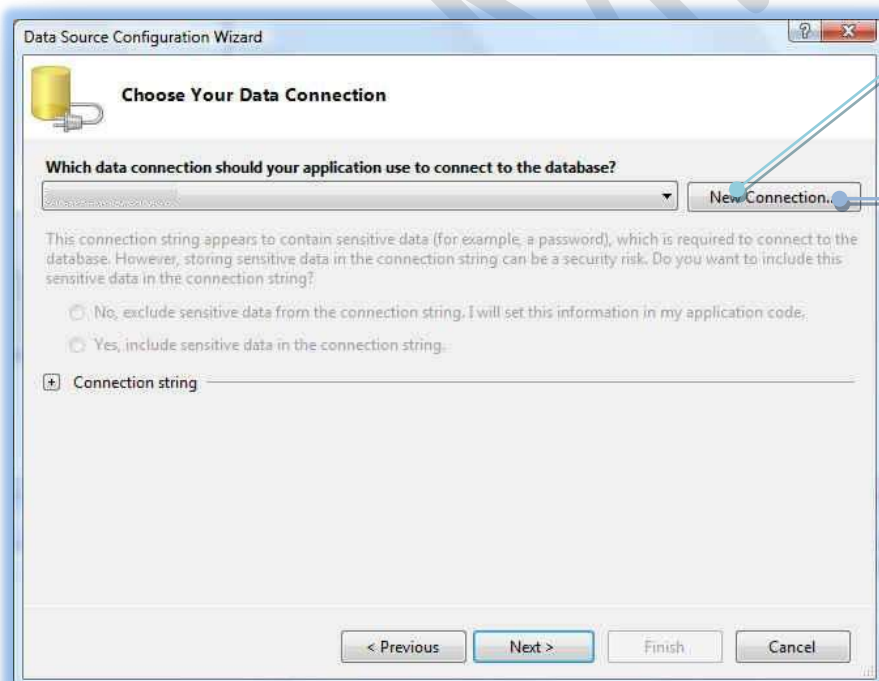
مرفق مع التمرين قاعدة بيانات اسمها (access 2007) وتم تنفيذها من خلال مايكروسوفت أكسس Microsoft Access 2007 وهي قاعدة البيانات العملية والتي تم عملها من خلال التمارين المرفقة بالجزء الثاني من كتاب Microsoft Access 2007 والتي تقوم على حساب درجات عدد من الطلاب لعدد أربع مواد دراسية وإظهار نتيجة كل طالب وتقديره وسوف يتم العمل عليها فبعد إنشاء نموذج بالطريقة المعتادة لدينا من قبل يتم الربط بين الفيچوال بيسك 2008 وقاعدة البيانات المعنية (access 2007) بذلك من خلال الخطوات التالية



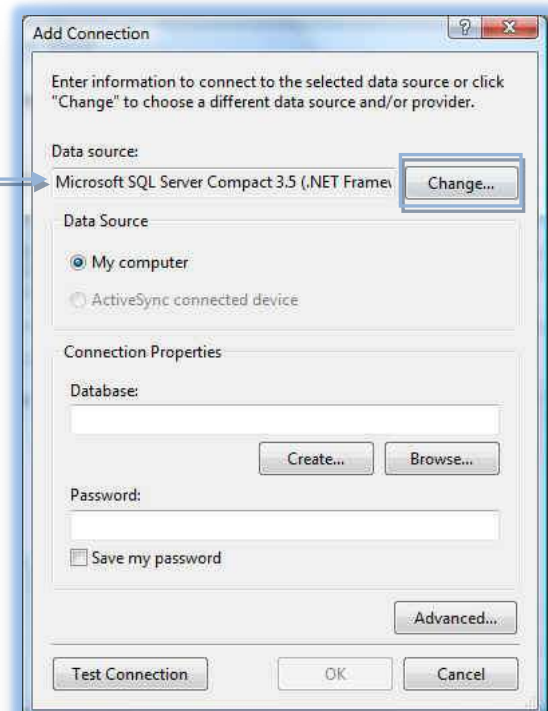
من خلال القائمة Data في شريط القوائم للبرنامج نختار add new data source لتظهر لنا النافذة التالية

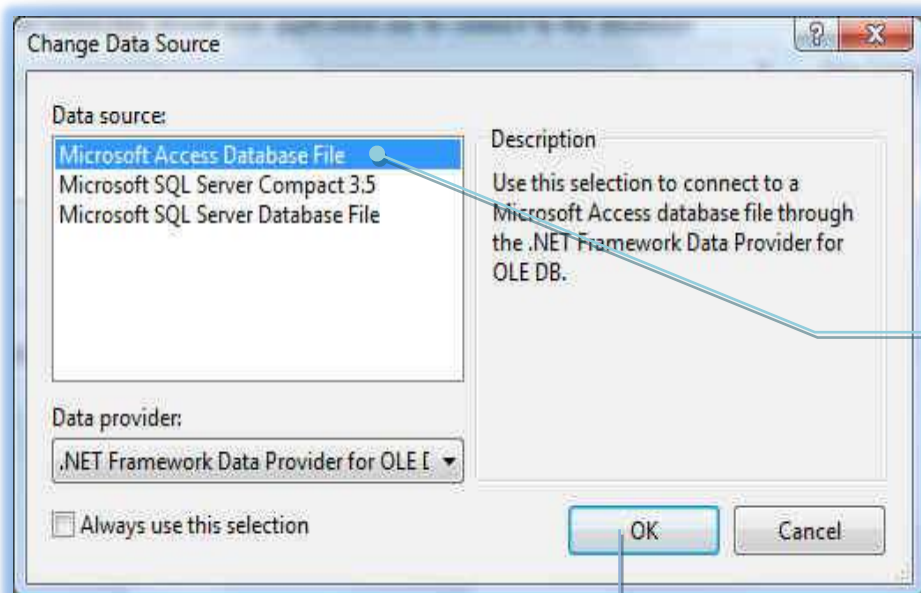


نقوم من خلالها باختيار Database ثم النقر على موافق لتظهر لنا الشاشة التالية

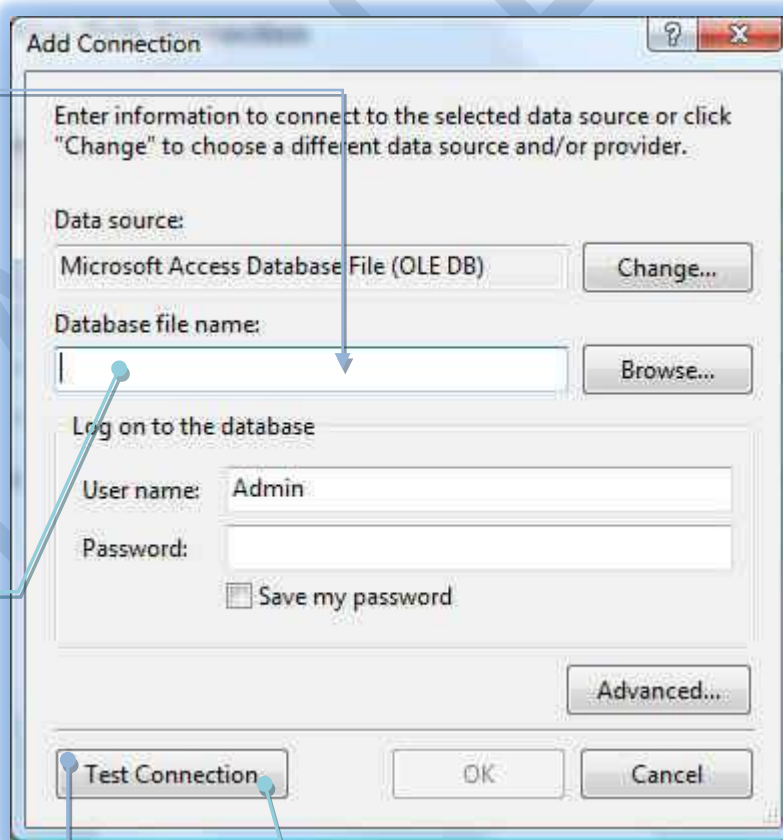


هنا تكون لنا النافذة معلومات عن الاتصال مع قاعدة البيانات وذلك بأخذ كل المعلومات عن قاعدة البيانات وطريقة الربط ومسار قاعدة البيانات وكذلك اسم المستخدم وكلمة المرور لقاعدة البيانات ويتم النقر على Change لتحديد نوع قاعدة البيانات



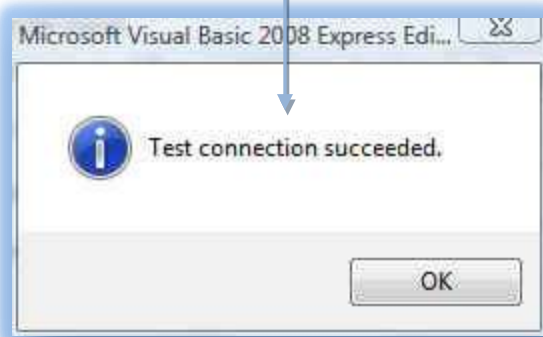


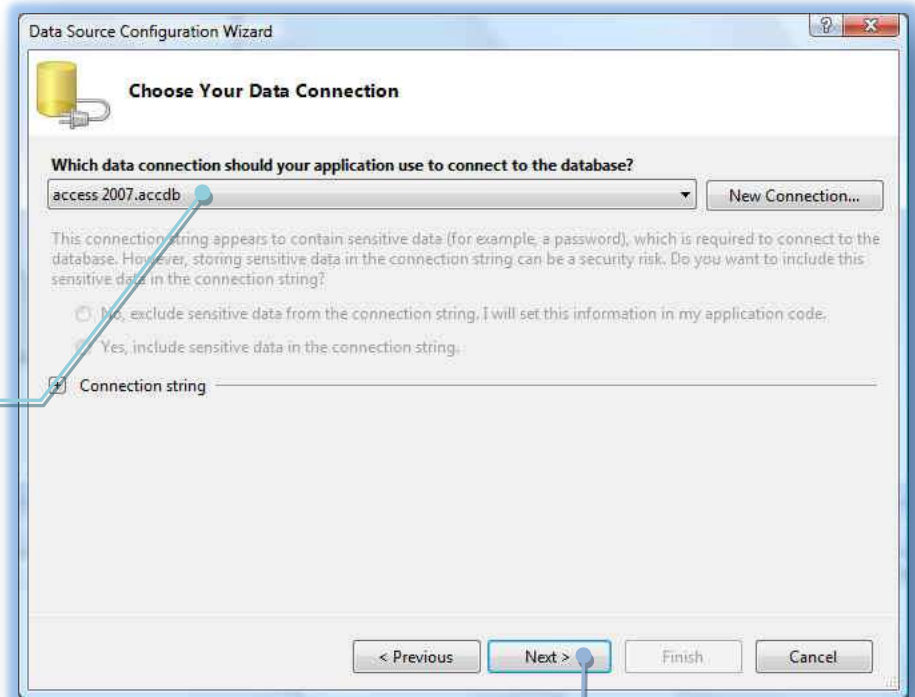
نختار منها نوع قاعدة البيانات التي تم التعامل معها نختار منها  
**Microsoft Access Database File**



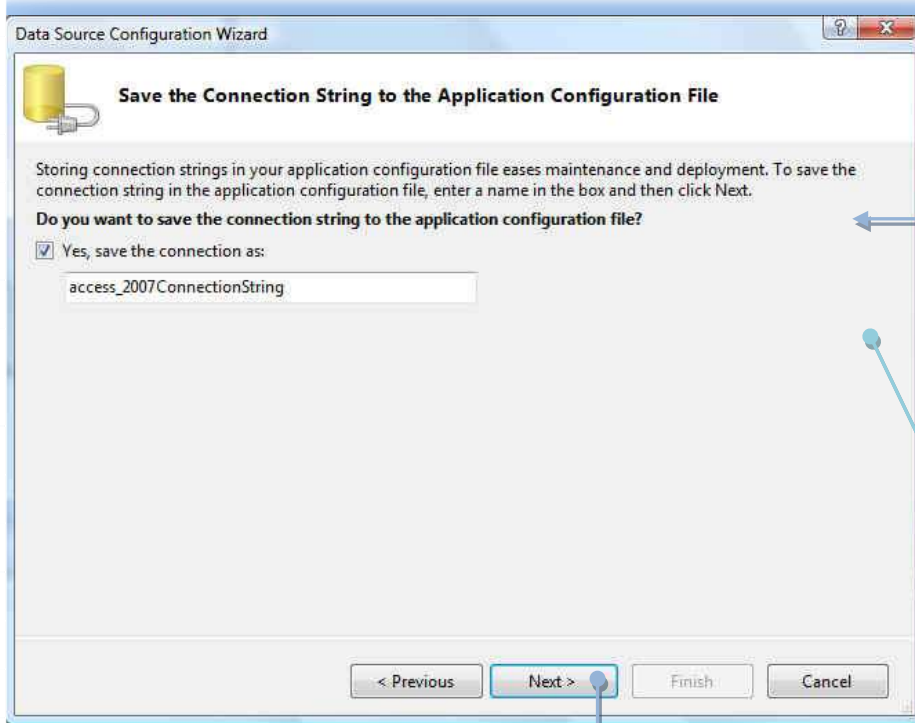
هنا يتم اختيار ملف الأكسس  
**access 2007** الموجود بالمرفقات

هنا يتم اختبار مسار الملف المختار  
من قبل وصحته

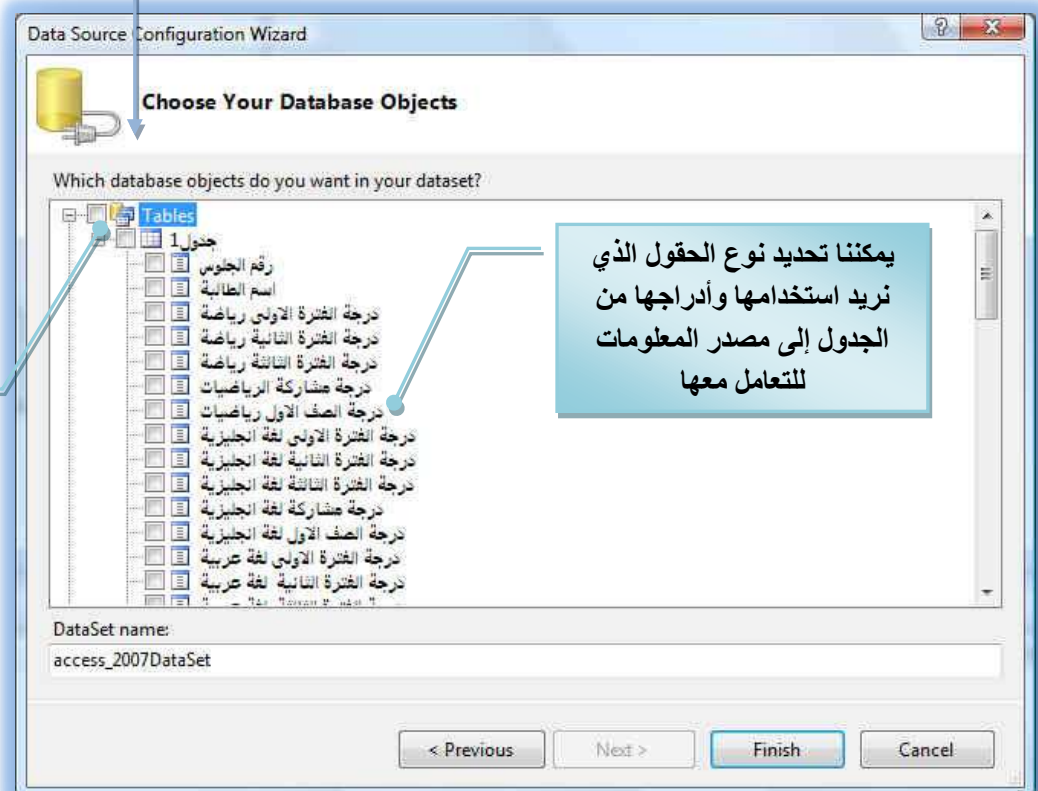




بالموافقة سيقوم الفيچوال بيسك بإرجاعنا لمعالج الاتصال مع قاعدة البيانات مرة ثانية



إذا كنا نريد نقل قاعدة البيانات إلى مجلد البرنامج access2007 داخل البرنامج نختار Yes

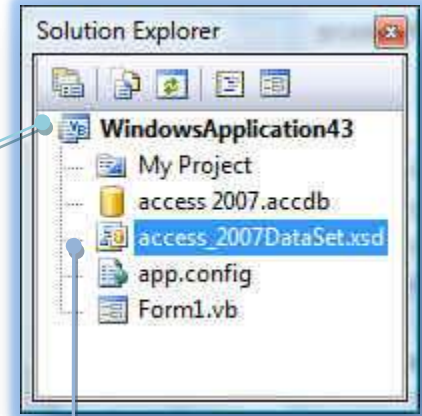


يمكننا تحديد نوع الحقول الذي نريد استخدامها وأدراجها من الجدول إلى مصدر المعلومات للتعامل معها

إذا كانت قاعدة البيانات كبيرة جداً يمكننا اختيار الجزء الذي يخصنا فقط منها ويتم هنا اختيار الملفات التي يتم عرضها وربطها مع الفيچوال بيسك و نختار الجدول كله في حالتنا هنا



وبالنقر على **Finish** يتم إدراج المكونات ويمكن استكشافها من خلال صندوق **solution Explorer**



نلاحظ هنا ظهور ملف أسمة **access\_2007DataSet.xsd** وهو الملف الذي تم تكوينه خلال عملية الاتصال مع قاعدة البيانات وهو من النوع **XML** وبالنقر عليه سنرى عرض لما يحتويه هذا الملف وذلك بعرض سجلات قاعدة البيانات يسمى هذا مصمم مجموعة البيانات

رقم الجلوس	اسم الطالبة	درجة الفترة الاولى رياضة	درجة الفترة الثانية رياضة	درجة الفترة الثالثة رياضة	درجة مشاركة الرياضيات	درجة الصف الاول رياضيات	... درجة الفترة الاولى لغة انج	... درجة الفترة الثانية لغة انج	... درجة الفترة الثالثة لغة انج	درجة مشاركة لغة انجليزية	... درجة الصف الاول لغة انجل

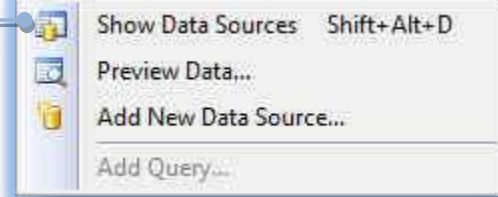
شاشة مصدر البيانات **Data Sources Window**

توجد شاشة مصادر البيانات **Data Sources** مدمجة مع مستكشف المشروع **Solution Explorer** أعلى يمين بيئة التطوير وإذا لم تكن ظاهرة أذهب إلى النموذج **Form1** في مرحلة التصميم ثم من القائمة **Data** من شريط القوائم نختار **Show Data Sources** لتظهر لنا كما بالشكل

مصمم مجموعة البيانات وهذا المصمم يحتوي على الكائنات التي تقوم بالتوصيل بين برنامجك وقاعدة البيانات

**Refresh** وهي لإظهار أي تغيير جديد داخل الجداول

إضافة أو حذف حقول لمجموعة البيانات



أسم مجموعة البيانات

لإضافة مجموعة بيانات **Dataset** جديدة

لتعديل مجموعة البيانات المختارة

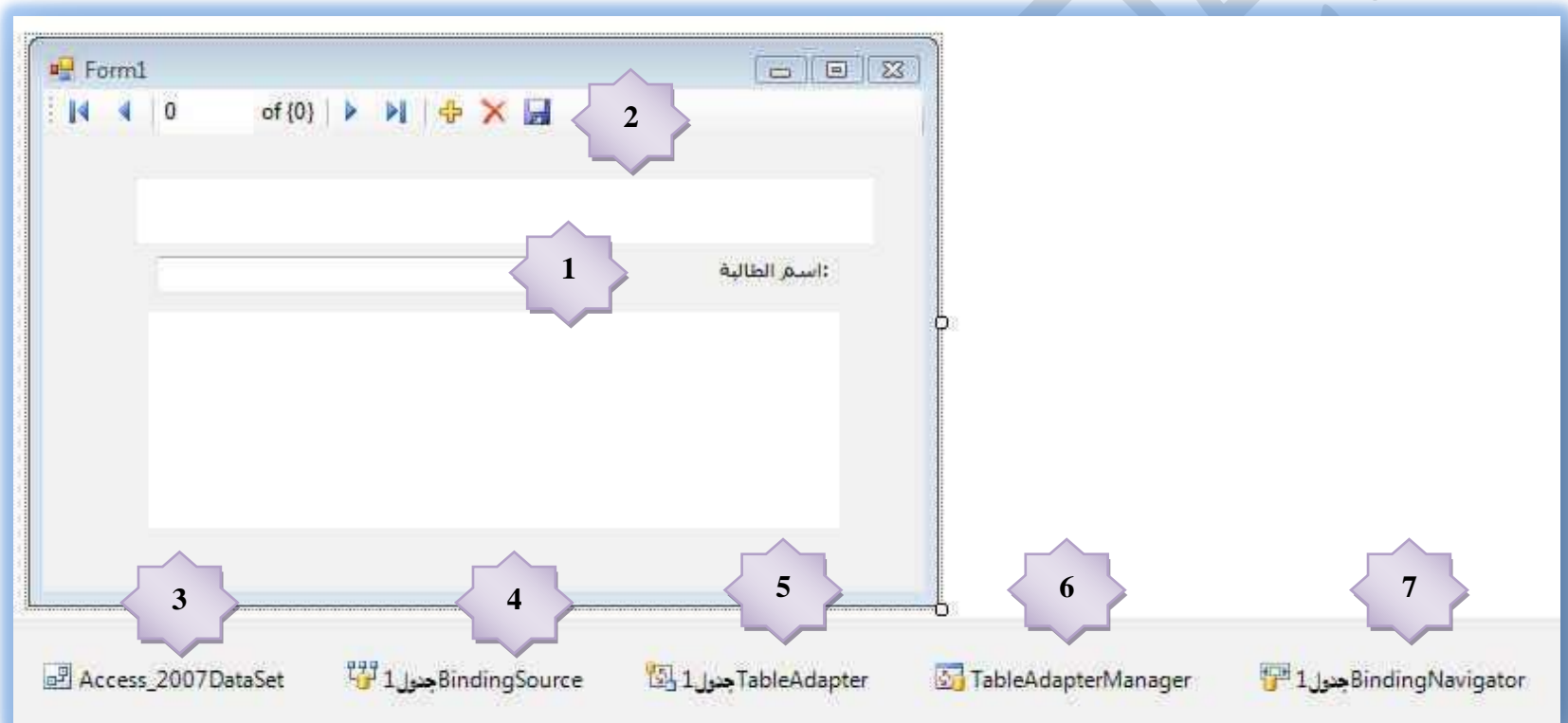
الجدول	رقم الجلوس	اسم الطالبة	درجة الفترة الاولى رياضة	درجة الفترة الثانية رياضة	درجة الفترة الثالثة رياضة	درجة مشاركة الرياضيات	درجة الصف الاول رياضيات	... درجة الفترة الاولى لغة انجليزية	... درجة الفترة الثانية لغة انجليزية	... درجة الفترة الثالثة لغة انجليزية	درجة مشاركة لغة انجليزية	... درجة الصف الاول لغة انجليزية	... درجة الفترة الاولى لغة عربية	... درجة الفترة الثانية لغة عربية	... درجة الفترة الثالثة لغة عربية	درجة مشاركة لغة عربية	... درجة الفترة الاولى مواد اجتماعية	... درجة الفترة الثانية مواد اجتماعية	... درجة الفترة الثالثة مواد اجتماعية	... درجة مشاركة مواد اجتماعية	
الجدول	abl	abl	abl	abl	abl	abl	abl	abl	abl	abl	abl	abl	abl	abl	abl	abl	abl	abl	abl	abl	abl

أسم الجدول الذي اخترناه من قاعدة البيانات

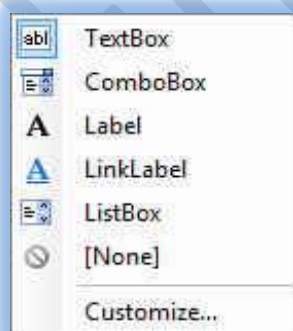
أسماء الحقول الذي تم اختيارها من الجدول ويمكن التحكم في الحقول الذي نريد إرفاقها إلى مصدر البيانات من خلال الخطوات السابقة **وفي حالتنا هنا اخترت جميع الحقول لاني اخترت الجدول كله من الخطوات السابقة**

تعتبر شاشة مصادر البيانات **Data Source** ميزة رائعة في بيئة التطوير للفيجوال بيسك 2008 حيث إنها تختصر الوقت علينا في التعامل مع قواعد البيانات بأنها تقوم بعرض البيانات الموجودة في مجموعة البيانات **Dataset** في تطبيقنا بل وتساعدنا على ربط البيانات في مجموعة البيانات **Dataset** مع عناصر التحكم في برنامجنا فتربط البيانات مع الأزرار و صناديق النص والمؤقتات وغيرها ويجب علينا أن نتذكر الآن بأن مجموعة البيانات **Dataset** ليست هي قاعدة البيانات ( وإنما هي عبارة عن وسيط بين برنامجنا وبين قاعدة البيانات وكل مجموعة من البيانات **Dataset** هي عبارة عن جزء من الجداول والحقول ولا تمثل كل قاعدة البيانات حيث تعرض فقط الجداول والحقول التي اخترناها خلال مرحلة الربط مع قاعدة البيانات كما وضحنا سابقا ) و يتم عرض مجموعة البيانات **Dataset** بشكل شجري **Tree** في شاشة مصادر البيانات **Data Source Window** فكل فرع يحتوي على الكائن الذي اخترته خلال عملية التصميم وكل مرة نقوم بتصميم مجموعة بيانات **Dataset** تقوم شاشة مصادر البيانات بإضافة فرع في الشجرة لتسهيل علينا عملية ربط مجموعة البيانات **Dataset** مع عناصر التحكم على النموذج وأسهل طريقة لإضافة المعلومات الموجودة ضمن مجموعة البيانات **Dataset** إلى تطبيقك هو استخدام الماوس في إضافة المكونات من شاشة مصادر البيانات **Data Source Window** إلى سطح النموذج

والآن سوف نقوم بإضافة أول حقل وليكن حقل ( اسم الطالبة ) إلى النموذج **Form1** وذلك بسحب من نافذة **Data Source** وإفلاته داخل سطح النموذج **Form1** ليظهر لنا النموذج كما بالشكل

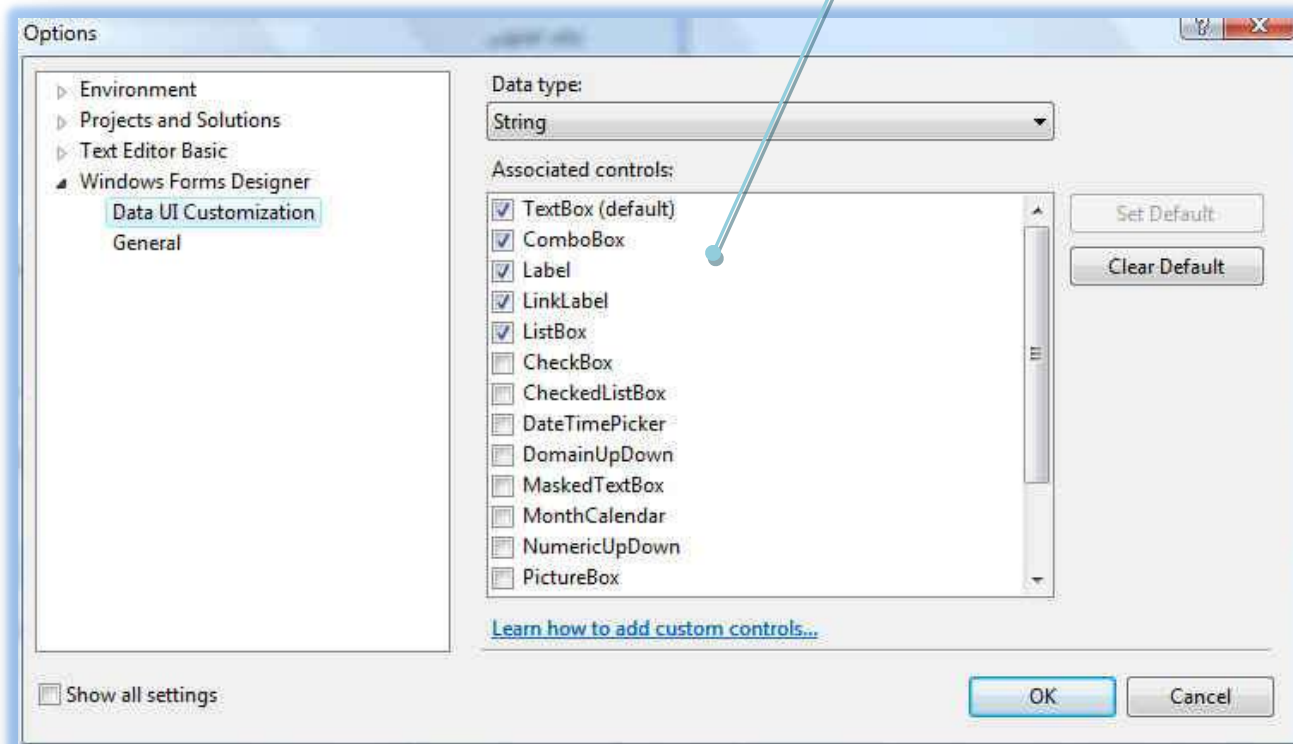


شاهد التغييرات التي حدثت على النموذج بإضافة المكونات الجديدة بالأسفل وكذلك بعض الكائنات على النموذج والتي تسهل علينا عملية الربط مع الحقل المدرج إلى النموذج وقد قام الفيجوال بيسك 2008 بإضافة المكونات التالية تلقائياً بدون أي تدخل منا

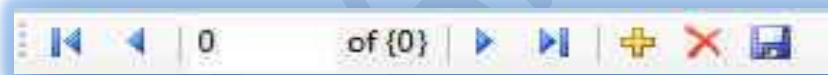


1. أضف صندوق نصي ليقوم بعرض الحقل ( اسم الطالبة ) ويمكننا بالوقوف على الحقل وفتح قائمته اختيار طريقة عرض هذا الحقل على النموذج ويكون ذلك بعدة طرق يتم اختيار الأنسب منها لعرضه كما وتقوم بيئة التطوير بعرض الطرق أو المكونات الأكثر استخداماً فيمكننا عرض النص في الحقل المذكور في صندوق نص **Textbox** أو **ComboBox** أو **Label** أو قائمة **ListBox** ويمكننا تخصيص المكونات التي تظهر بإضافة مكونات جديدة أو بإلغاء مكونات بالضغط على **Customize** تظهر لنا النافذة التالية والتي يمكن من خلالها تخصيص الحقل المدرج في النموذج إلى أي كائن تريد أن يظهر به

عند تخصيص الحقول عند تصميم قاعدة البيانات أثناء تصميم الجدول فيتم عند إدراجها تلقائياً إلى النموذج اختيار المكون الأنسب لعرضها فمثلاً لو خصصنا حقل على أنه تاريخ مثلاً نقوم ببيئة التطوير في عرض هذا الحقل داخل المكون **DateTimePicker** تلقائياً



2. أضافه **مستعرض** في أعلى النموذج لسهولة التحكم والتنقل بين السجلات



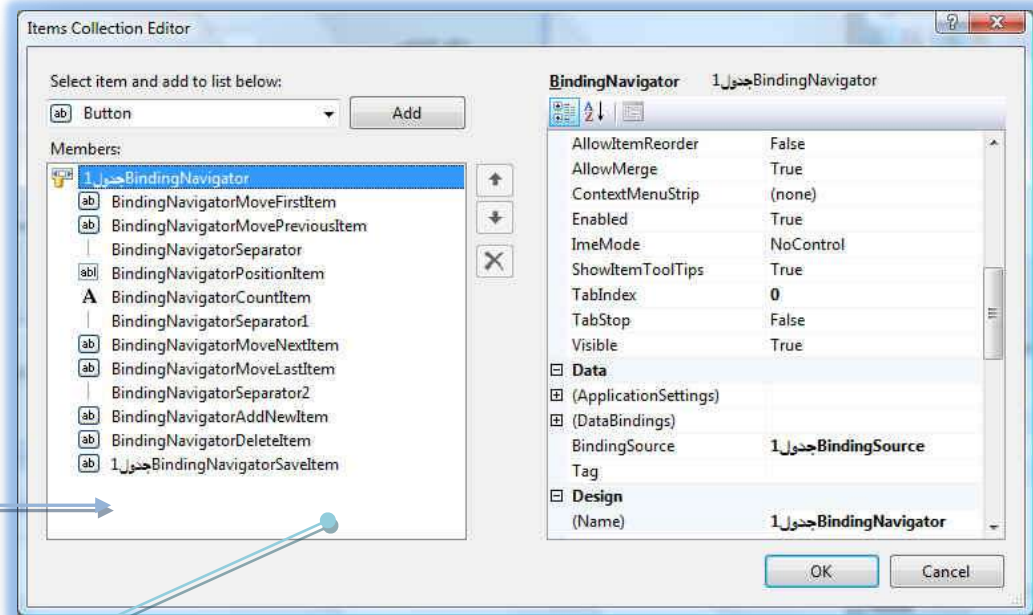
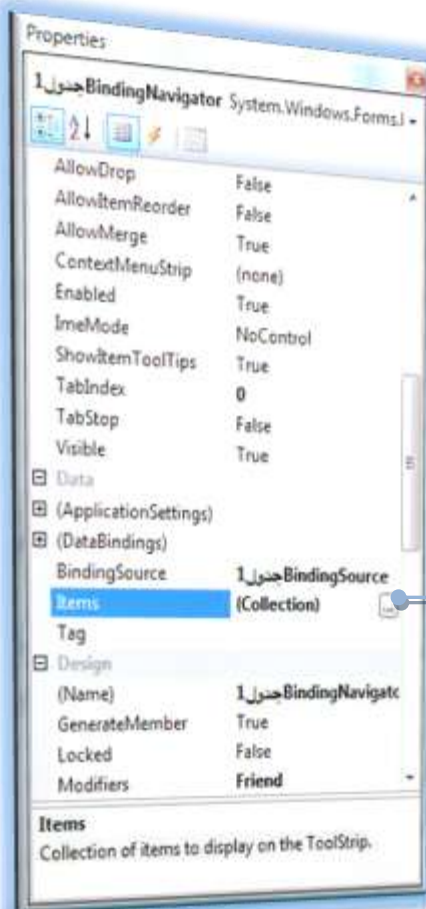
3. أضافه **Access\_2007Dataset** مجموعة البيانات **Dataset** التي صممناها سابقاً والتي تعرض بعض الجداول وبعض الحقول من قاعدة البيانات ( **access\_2007.accdb** ) وهي المستخدمة في التمرين

4. أضافه **BindingSource** جدول1 عبارة عن مكون وسيط يقوم بالوصل بين الجدول (جدول1) والكائنات ( صندوق النص أو غيره ) التي تمثله على النموذج.

5. أضافه **TableAdapter** جدول1 عبارة عن مكون وسيط يقوم بالربط بين مجموعة البيانات **Access\_2007Dataset** وقاعدة البيانات ( **access\_2007.accdb** )

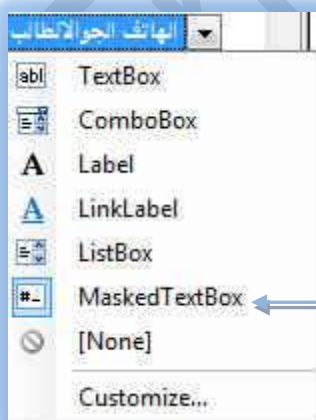
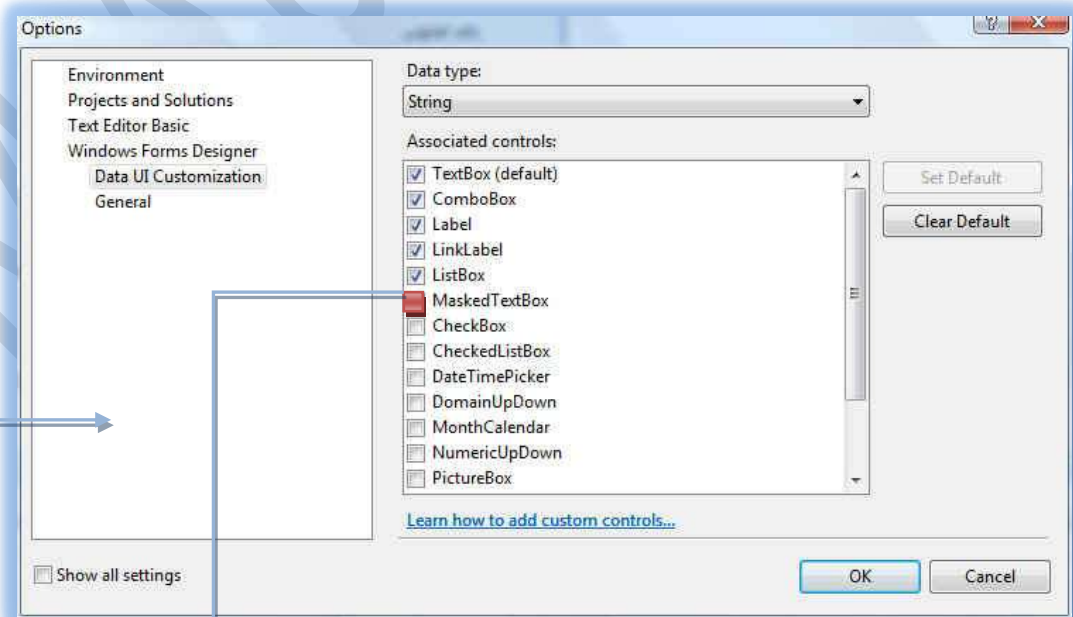
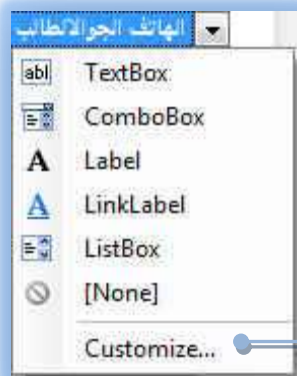
6. أضافه **TableAdapterManager** وهو مثل سابقة لتنظيم الربط بين مجموعة البيانات **Access\_2007Dataset** وقاعدة البيانات ( **access\_2007.accdb** )

إضافة **BindingNavigator** جدول1 وهو المكون الذي يظهر المستعرض 2 وهو عبارة وسيط يسهل لنا عملية استعراض البيانات على النموذج بشكل يمكننا الانتقال من السجل الحالي إلى الذي يليه أو إلى السجل السابق ويمكننا هذا المستعرض من حذف أو إضافة سجل جديد لقاعدة البيانات كما أننا نستطيع حذف أو تعديل أحد أزرار المستعرض فإذا أردنا عدم السماح للمستخدم بحذف السجلات بسهولة نحذف الزر الذي يقوم بالحذف خلال مرحلة التصميم ويتم ذلك من خلال صندوق خصائص **BindingNavigator** جدول1 ويتم الذهاب إلى الخاصية **Items** ونضغط على الزر الخاص بها لتظهر لنا النافذة التالية



ومن هذه النافذة نستطيع عدم تفعيل المكون الذي لا نحتاج إليه في المستعرض كما نستطيع تعديل الخصائص للمكونات بالشكل الذي يناسبنا

نفرض مثلا إننا نريد إضافة ( رقم الهاتف الجوال للطالب ) إلى النموذج من خلال الجدول 1 ولكن نريد إن ندخله إلى النموذج في المكون **Masked Textbox** وهو غير موجود في قائمة الاختيارات فيتم أدراجة من خلال الطريقة السابقة باختياره ثم الموافقة وبالرجوع إلى **رقم الهاتف الجوال للطالب** وفتح قائمته ستظهر كما بالشكل و بها الإضافات وهي **Masked Textbox**



ثم يتم تنسيق **Masked Textbox** حسب رغبتنا كما سبق وذكرنا من قبل ليظهر في النموذج كما بالشكل التالي

وبعد إدخال بعض الحقول من **الجدول 1** بنفس الطرق السابقة إلى النموذج **Form1** نقوم بتشغيل البرنامج **F5** لنرى لنموذج التالي

الانتقال لأخر سجل

الانتقال للسجل التالي

إضافة سجل

حذف سجل

حفظ سجل

الانتقال للسجل السابق

الانتقال لأول سجل

رقم السجل الحالي

شكل حقل الهاتف بعد تنسيق Masked Textbox أو كما سبق وشرحنا

عند تصميم جدول قاعدة البيانات تم اختيار الحقل كتاريخ ولذلك تم إدراجة تلقائيا بهذا الشكل

بعد تنفيذ البرنامج نقوم الآن بإضافة وإلغاء سجلات والذهاب إلى آخر سجل وإلى أول سجل وكما وضحنا من قبل سيقوم البرنامج بحذف السجل أو بإضافته أو تعديله ولكن في مجموعة البيانات **Dataset** فقط ولن ينقل هذه التعديلات إلى قاعدة البيانات الرئيسية في الملف **access\_2007** إذا قمنا بإعادة تشغيل البرنامج مرة ثانية سنلاحظ إن عدد السجلات تغيرت وللتأكد من ذلك نقوم الآن بإغلاق التطبيق نهائيا ثم نقوم بإعادة تشغيل البرنامج مرة أخرى لنشاهد السجلات كما هي بدون تعديل ومن هذا نستنتج بأن التعديلات التي تتم خلال عملية تشغيل البرنامج تحفظ في الذاكرة المؤقتة ولا تحفظ في قاعدة البيانات الأساسية **access\_2007** ونستطيع جعل البرنامج يحفظ التعديلات في قاعدة البيانات الرئيسية **access\_2007** بواسطة الكود التالي والموجود في **Public Class Form1** تحت الحدث **Click** للزر **Save** الموجود بالمستعرض الذي تم إضافته من قبل **الفيجوال بيسك 2008** تلقائيا كما سبق وعرّفنا وقاعدته العامة هي

`Me.(Table Name)TableAdapter.UpdateAll(Me.(Database Name)DataSet.(Table Name))`

اسم الجدول الذي يتم حفظ البيانات فيه  
وهو نفس الجدول للحقل المستخدم

اسم قاعدة البيانات المستخدمة في المشروع

اسم الجدول الذي يتم حفظ البيانات فيه  
وهو نفس الجدول للحقل المستخدم

`Me.(Table Name)TableAdapter.Update(Me.(Database Name)DataSet.(Table Name))`

1. في حالة استخدام **UpdateAll** يتم السماح لـ **TableAdapterManager** بالتعامل مع أكثر من قاعدة بيانات من برنامجك وحفظها ويمكنك التأكد من ذلك بإغلاق البرنامج ثم فتحه مرة أخرى لتشهد البيانات التي قمت بتغييرها قد تغيرت
2. في حالة استخدام **Update** يتم السماح لـ **TableAdapterManager** بالتعامل مع قاعدة بيانات من برنامجك ويتم حفظ البيانات حتى في الجدول

```
Public Class Form1
Private Sub جدول1BindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles جدول1BindingNavigatorSaveItem.Click
Me.Validate()
Me.جدول1BindingSource.EndEdit()
Me.TableAdapterManager.UpdateAll(Me.Access_2007DataSet)
End Sub
End Class
```

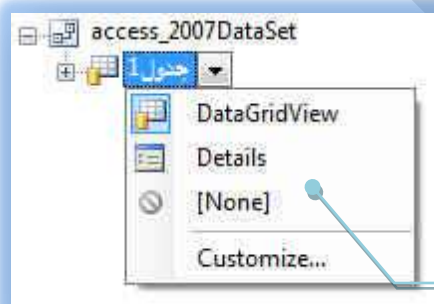
وهو في حالتنا هنا يكون كالكود التالي

تحميل التمرين الثالث والأربعون

والتمرين التالي ستجربة التطبيق العملي على كل من الخطوات السابقة ومرفق معه قاعدة بيانات كبيرة للتعامل معها خلال التصميم

استعراض البيانات على شكل جدول بواسطة الكائن **DataGridView**

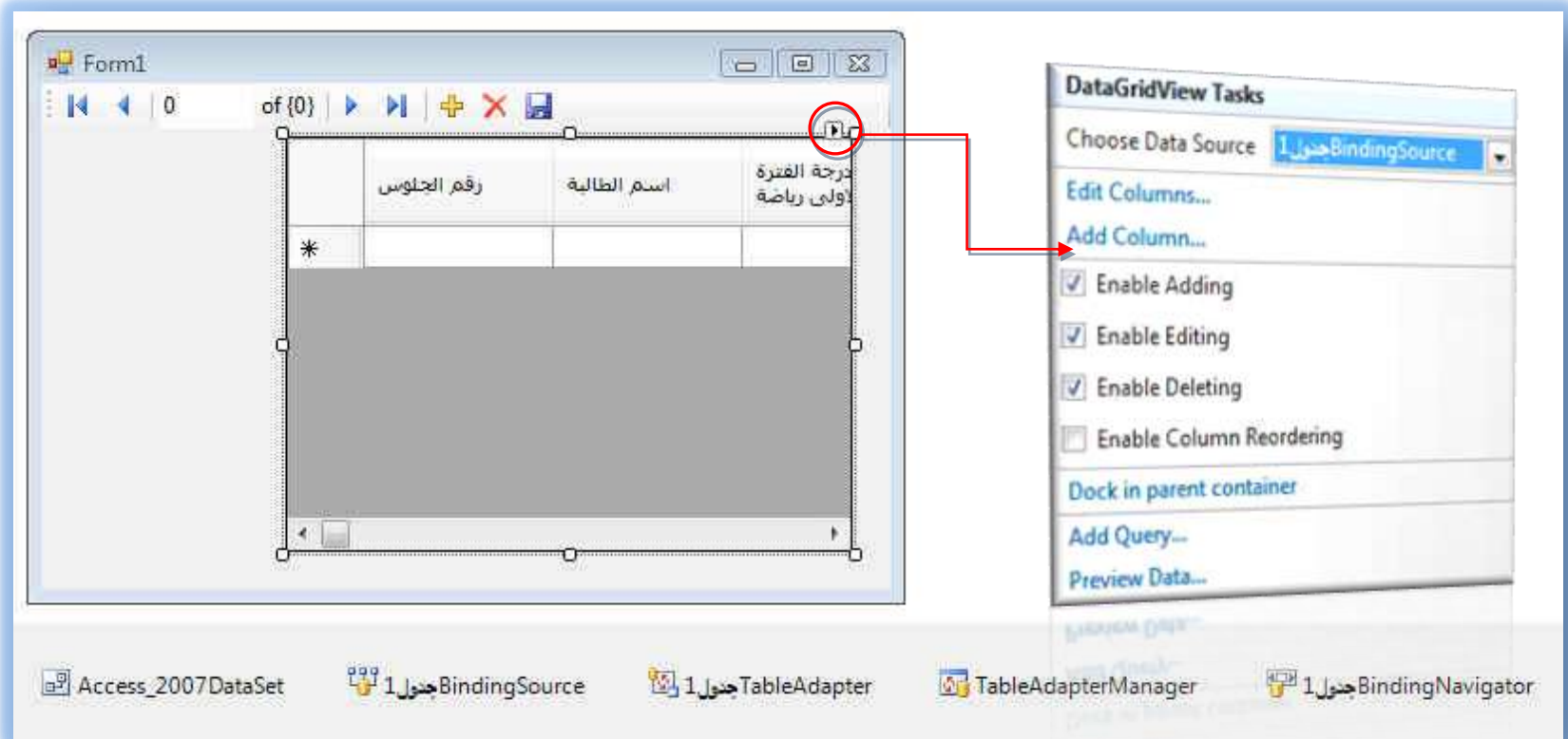
الكائن **DataGridView** هو عبارة عن كائن تتم إضافته إلى النموذج ليقوم بعرض جدول كامل من قاعدة البيانات وقد يحتوي هذا الجدول على أكثر من سجل وتكون طريقة العرض هي طريقة الجدول بحيث يعرض الحقول في أعمدة والسجلات في أسطر كما هو الحال في جداول ملفات الأكسل فهي تتشابه مع هذا الكائن إلى حد كبير في طريقة العرض وفي **الفيجوال بيسك 2008** يمكن عرض البيانات على **DataGridView** مباشرة باستخدام مجموعة البيانات **Dataset** ويتم ربط البيانات مع الكائن بواسطة الخاصية **BindingSource** ويتم ذلك بعد عملية ربط البيانات بواسطة النافذة **Data Source Configuration Wizard** واستخدام نافذة مصادر البيانات التي تعلمنا كيف نستخدمهم من قبل وبعد عملية الربط مع البيانات يقوم **الفيجوال بيسك 2008** بتعبئة الجدول **DataGridView** مباشرة بعد تحميل النموذج



وستعلم ذلك معا فبعد ربط قاعدة البيانات **Access2007** إلى المشروع والقيام بإدراج الجدول كاملا وإتمام الخطوات كما تعلمنا ( راجع الخطوات السابقة ) وبعد الانتهاء من ربط قاعدة البيانات نقوم بالذهاب إلى **Data Source** وبالوقوف السهم الموجود بجانب اسم الجدول المراد إدراجه كاملا إلى المشروع لتظهر لنا القائمة التالية

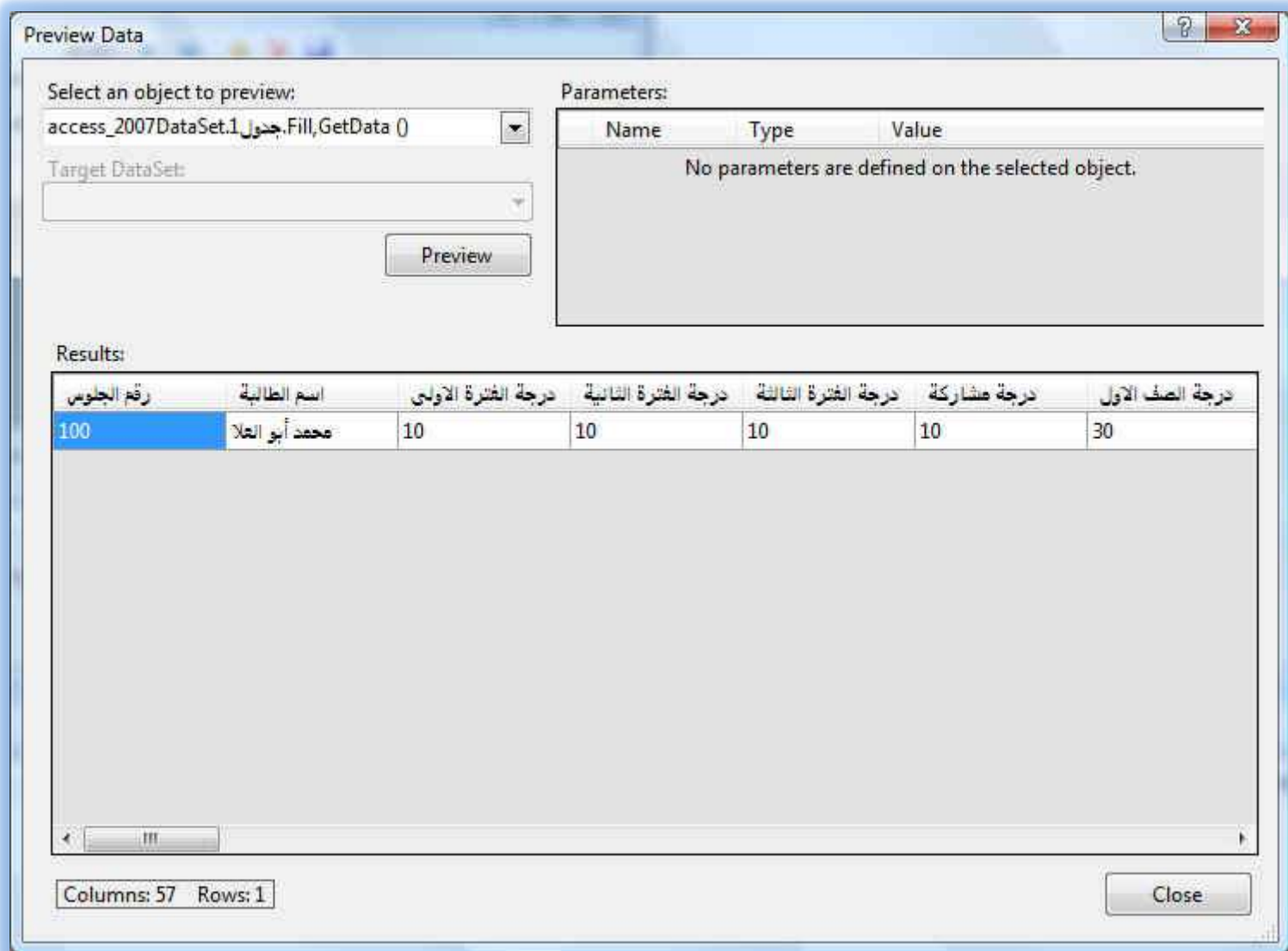
1. **DataGridView** لعرض الجدول بداخل الكائن **DataGridView** على شكل جدول يحتوي على صفوف وأعمدة
2. **Details** ويستخدم لعرض الجدول على شكل مفصل على النموذج حيث يقوم بعرض صندوق نصي لكل حقل وبجانبه ليبل تعريف بصندوق النص
3. **None** يمنع ربط هذا الجدول مع أية كائن، إذا اخترنا هذا الخيار فلن يمكننا ربط الجدول مع الكائنات
4. **Customize** يمكننا من اختيار كائن جديد غير موجود في الخيارات ولا بد أن يتقبل هذا الكائن عرض محتويات جدول بأكمله

وبالنقر على **DataGridView** ونقوم بإضافتها إلى النموذج، سيقوم **الفيجوال بيسك 2008** بإضافة المكونات التالية **Access\_2007Dataset** و **BindingSource** جدول1 و **TableAdapter** جدول1 و **TableAdapterManager** و **BindingNavigator** جدول1 ( والذي تعاملنا معهم من قبل في التمرن السابق ) وأيضا يتم إضافة الكائن **DataGridView** جدول1 إلى النموذج ليكون النموذج كما بالشكل التالي



ستلاحظ عدم وجود بيانات على الـ **DataGrid** في هذه اللحظة ولكن سيقوم البرنامج بتحميل البيانات بعد تشغيله **F5** وبالذهاب إلى السهم الصغير أعلى يمين الـ **DataGrid** وبالنقر عليه لتظهر لنا النافذة التي نستطيع من خلالها التعامل مع الـ **DataGrid** وسنتعرف على محتواها معا

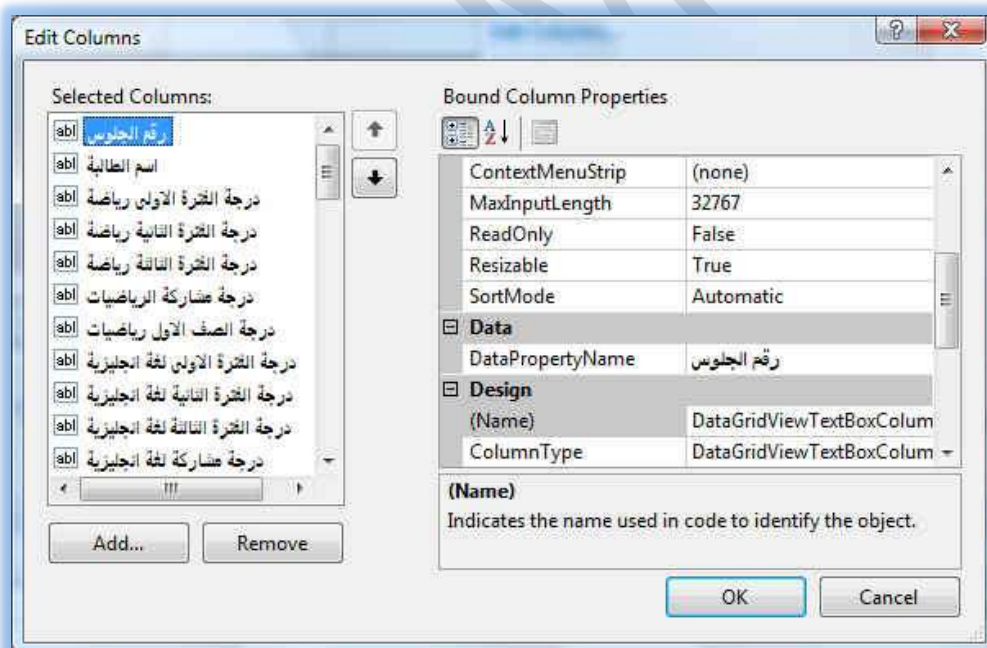




### التعامل مع الأعمدة (الحقول) Colum

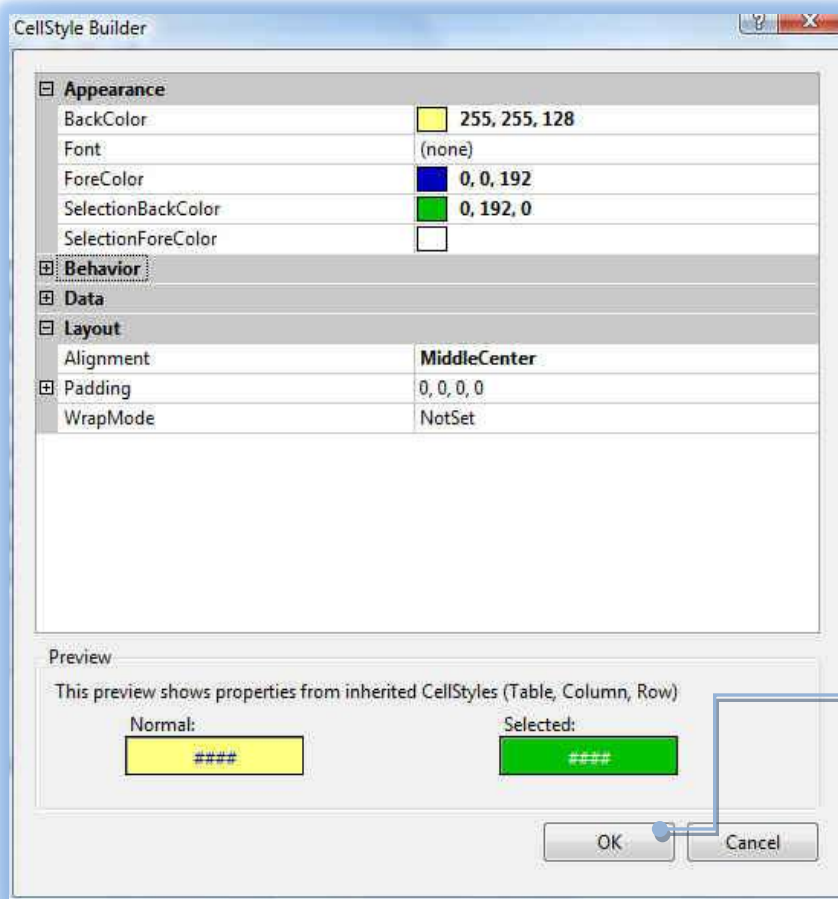
من خلال القائمة السابقة يتم التعامل مع الأعمدة وذلك من خلال النقر على الخيار التالي **Edit Columns** لتظهر لنا هذه النافذة والتي تمكننا من التعامل مع الأعمدة (الحقول) بالإضافة أو الحذف ويمكن التعامل مع خواص كل عمود على حدة وذلك من خلال اختياره من الجزء الأيسر واختيار الخواص المراد تطبيقها على من الجزء الأيمن من النافذة وبالموافقة علىية يتم تطبيقها على الـ **DataGrid**

ويمكننا أيضا من خلالها تغيير شكل خلايا الجدول سواء من خلال تغيير مساحة الخلايا أو لون الخلفية في الخلايا وغيرها من الخصائص فمن خلال النافذة نلاحظ أن العمود الذي نقوم بالتعديل عليه هو العمود المظلل على يسار النافذة وهو هنا (رقم الجلوس) وإذا أردنا تعديل عمود آخر نختاره من يسار النافذة وهكذا



وهناك خواص عديدة للتحكم في الأعمدة فيمكنك جعل البيانات في عمود معين للقراءة فقط أو يمكنك السماح للمستخدم بتغيير عرض الخلية وطولها أو لا ومن هذه الخواص ما يمكننا من تغيير ألوان الخلايا في المشروع

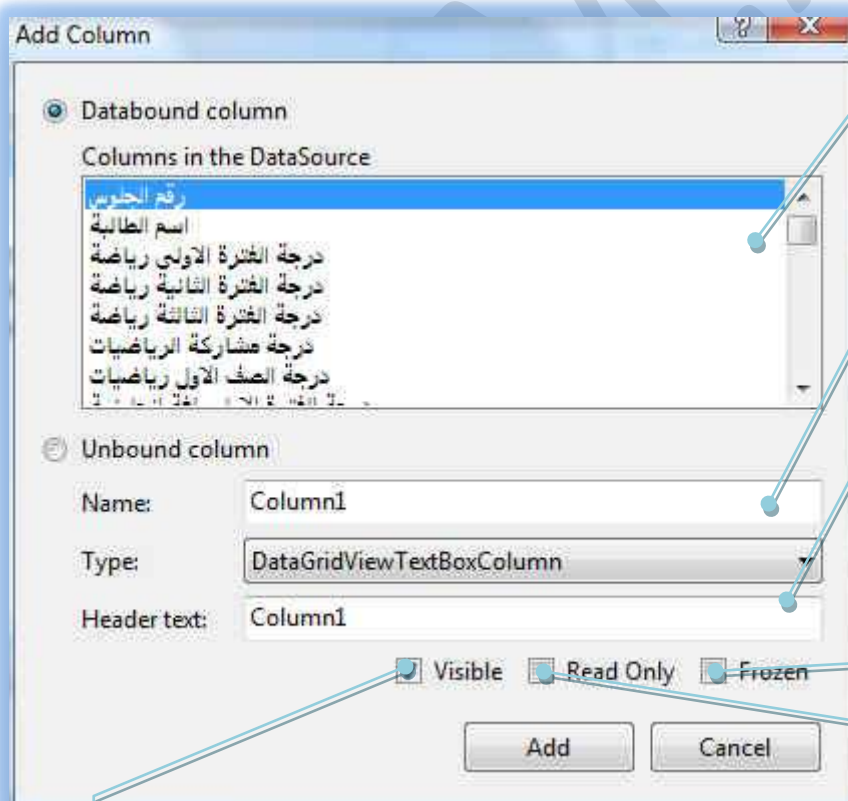




فمثلاً لو أردنا تغيير لون الصفوف في الـ **DataGrid** نقوم بالذهاب إلى الخاصية `DataGridViewCellStyle{ }` وبالنقر على المربع الخاص بها تظهر لنا النافذة التالية وهذه الخاصية تتحكم في لون الصفوف عند التنقل من صف إلى آخر وبتغيير هذه الخاصية يقوم بتغيير لون الصفوف إلى لونين أحدهما بالأصفر والكتابة به بالأزرق وهو لون الصف في الحالة العادية والآخر بالأخضر والكتابة بداخلة بالأبيض وهو لون الصف في حالة الوقوف عليه واختياره ويمكنك تغيير الألوان حسب تنسيقك للـ **DataGrid** ويفيد هذا التغيير في حالة إذا كانت البيانات كثيرة حيث تسهل عملية قراءة البيانات ليكون الشكل كالتالي



وقد نريد أن نحذف بعض الحقول ويتم ذلك من خلال الوقوف على الحقل المراد حذفه والنقر على الزر **Remove** ليتم حذفه فوراً من **DataGrid** وأيضا يمكننا إضافة حقول غير موجود أو سبق وحذفناها عن الحاجة إليها للظهور في الـ **DataGrid** بالنقر على الزر **Add** لتظهر لنا النافذة التالية ليتم من خلالها التحكم في الحقول المدرجة وأيضا التحكم في خواص إدراجها داخل الـ **DataGrid**



أسماء الحقول التي يمكن إضافتها إلى **DataGrid**

اسم الحقل الذي تقوم بإضافته إلى **DataGrid**

عنوان الحقل التي يظهر في **DataGrid**

التحكم في أن يكون الحقل جامد أو ثابت في مكانة وتستخدم في حالة وجود عدد كبير من الحقول في الـ **DataGrid**

التحكم في ظهور أو إخفاء الحقل في **DataGrid**

التحكم في أن يكون الحقل للقراءة فقط أو للتعديل أيضا

تحميل التمرين الرابع والأربعون

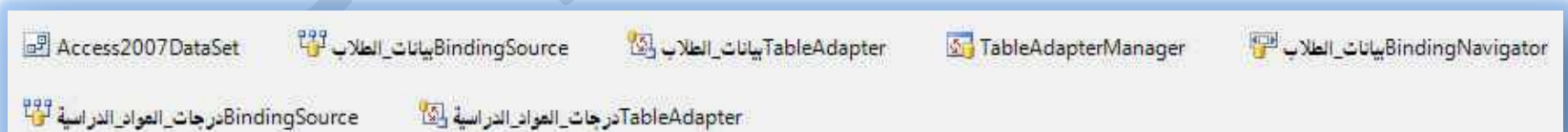
التعامل قواعد البيانات الكبيرة بإضافة **DataGrid** ثانية وكذلك مستعرض إضافي

عند التعامل مع قواعد البيانات الكبيرة التي تحتوي على أكثر من جدول أو جدول كبير جدا نريد تقسيم بياناته قد نحتاج إلى إضافة **DataGrid** أخرى للنموذج وإضافة مستعرض جديد للـ **DataGrid** وإضافة الـ **DataGrid** من مصادر البيانات بسيطة كما تعلمنا من قبل ولكن الإضافة الجديدة هنا هي إضافة مستعرض جديد وتعديل الخاصية **BindingSource** ويتم ذلك كالتالي

1. عندما يكون لدينا جدول كبير ونريد تقسيم بياناته إلى أكثر من **DataGrid** في النموذج نقوم بإدخال الـ **DataGrid** الأول بنفس الطريقة السابقة ويتم تعديله من خلال اختيار مجموعة الحقول التي يتم عرضها ثم القيام بإدخال الـ **DataGrid** الثاني وأيضا تعديل مجموعة الحقول التي تريد أن تظهر به وتطبيق التنسيق السابقة عليه يمكن أن يظهر لنا بهذا الشكل ويتم التحكم في الاثنين **DataGrid** من خلال مستعرض واحد فعند التنقل بين الصفوف في الـ **DataGrid** الأول يتم التنقل تلقائيا في الـ **DataGrid** لنفس الصف



2. في حالة استخدامنا أكثر من جدول تظهر لنا جميع الجداول التي تم إدراجها في المشروع من قاعدة البيانات لتظهر لنا في **Data Source** كالتالي وبعد إضافة الـ **DataGrid** للجدول الأول (بيانات الطلاب) إلى النموذج **Form1** يتم إضافة **DataGrid** أخرى للجدول الثاني (درجات المواد الدراسية) بنفس الطريقة ونلاحظ عند إضافتها إلى النموذج بقيام فيجوال بيسك 2008 بإضافة مكونات جديدة إلى المشروع بجانب المكونات الخمسة التي تم إضافتها من قبل من خلال كل تعاملتنا السابقة لاحظ المكونات الجديدة في الجدول التالي



في حالة إدخال (بيانات الطلاب والمواد الدراسية)

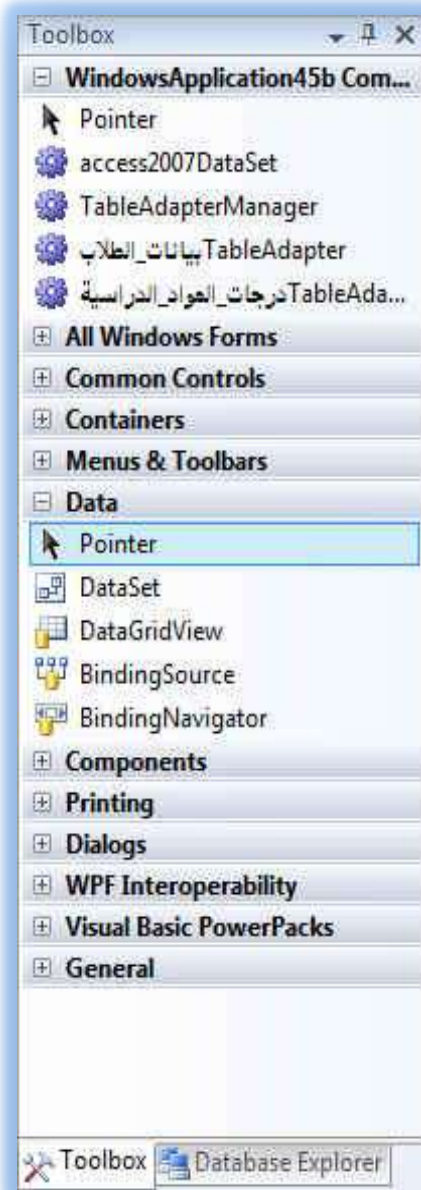
Access\_2007Dataset  
 BindingSource\_بيانات\_الطلاب  
 TableAdapter\_بيانات\_الطلاب  
 TableAdapterManager  
 BindingNavigator\_بيانات\_الطلاب  
 BindingSource\_درجات\_المواد\_الدراسية  
 TableAdapter\_درجات\_المواد\_الدراسية

في حالة إدخال جدول (بيانات الطلاب فقط)

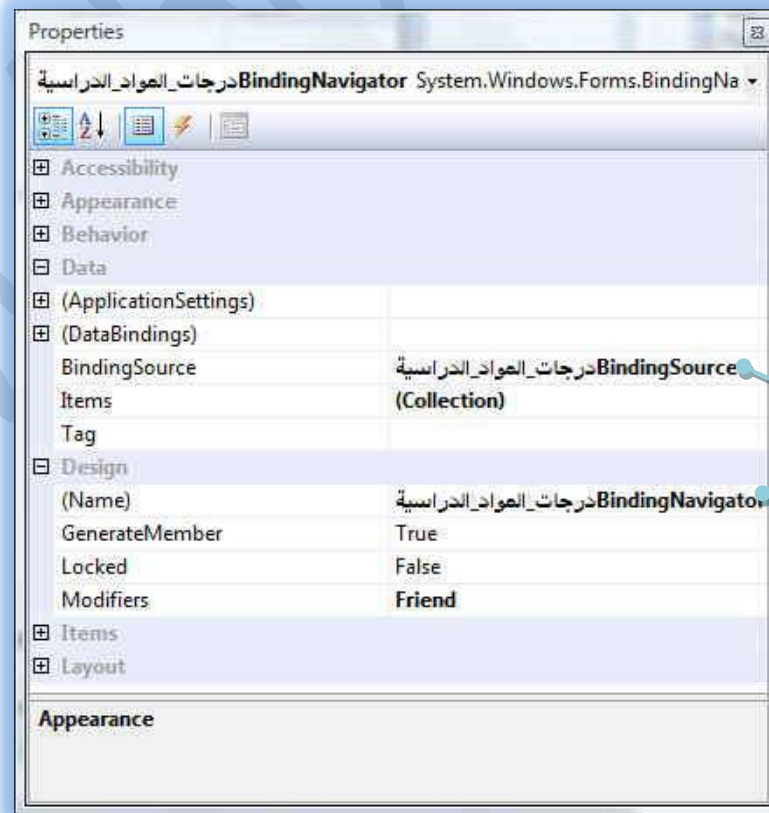
Access\_2007Dataset  
 BindingSource\_بيانات\_الطلاب  
 TableAdapter\_بيانات\_الطلاب  
 TableAdapterManager  
 BindingNavigator\_بيانات\_الطلاب

هذين المكونين هما عبارة عن وسطاء لنقل البيانات بين جدول درجات المواد الدراسية وبين قاعدة البيانات

نلاحظ أنه عند أضافة الجدولين في الـ **DataGrid** وتنسيقهم معا كما تعلمنا من قبل يتم ظهورهم كما بالشكل التالي



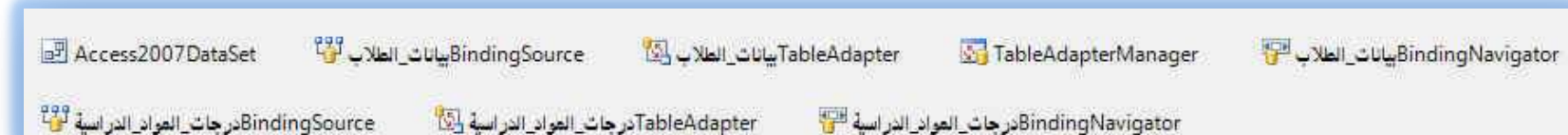
نلاحظ أنه يوجد شريط مستعرض واحد للجدولين ولكن ماذا لو أردنا شريط استعراض لكل جدول منهم على حدة ويكون ذلك بإتباع الآتي من صندوق الأدوات يتم إدراج الأداة **BindingNavigator** إلى النموذج لتقوم هذه الأداة بإضافة مستعرض جديد إلى النموذج خاص بـ **DataGrid** الثانية ولكن لم يتم الربط بها حتى الآن ولكي نقوم بالربط بين الأداة **BindingNavigator** والكائن **DataGrid** من خلال صندوق الخواص الخاص بالأداة **BindingNavigator** كالتالي



يتم من هنا تغيير اسم الأداة إلى **DataGridView** درجات\_المواد\_الدراسية

هنا يتم ربط الأداة **DataGridView** درجات\_المواد\_الدراسية مع **DataGrid** الخاصة بجدول درجات المواد الدراسية

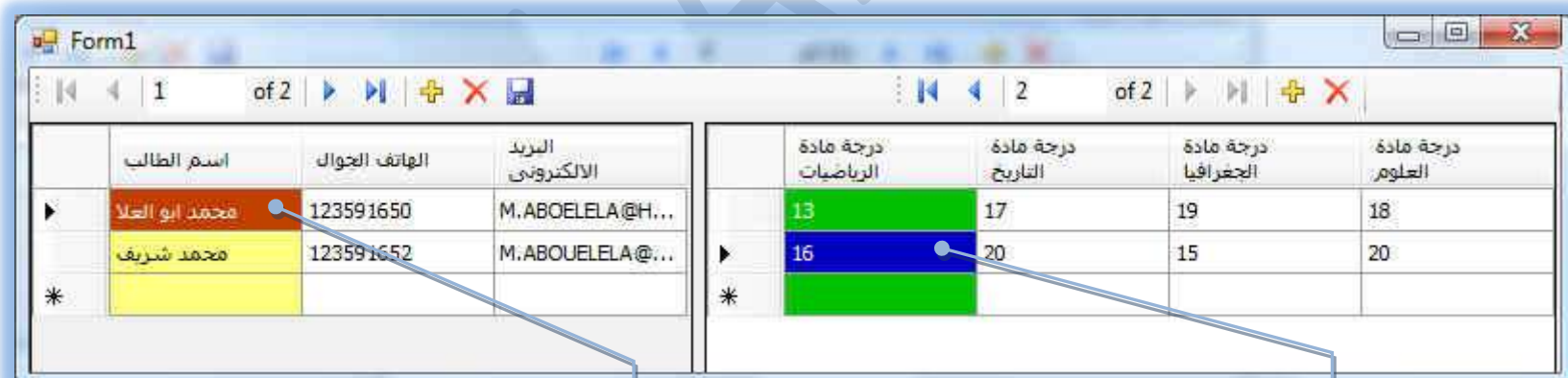
لتصبح المكونات بهذا الشكل ويتم الربط بين الأداة الجديدة للمستعرض **DataGridView** بالـ **DataGrid** الخاصة بالجدول درجات المواد الدراسية ليصبح شكل الأدوات المدرجة بالمشروع كالتالي



والآن سيقوم البرنامج بعد أدرجة مستعرض جديد خاص بالجدول الثاني أسفل المستعرض الخاص بالجدول الأول ومن خلال كل منهم نستطيع أن نقوم بتنسيقهما ووضعهما في وضعهما المناسب من خلال السهم الصغير الموجود لكل مستعرض (يعامل المستعرض معاملة شريط الأدوات كما سبق ودرسنا ) وقد نستطيع أن نصل بالتنسيق إلى الشكل التالي



ومع تنسيق الجداول والحقول بالطرق السابقة وإدخال البيانات إلى المشروع نقوم بتنفيذ المشروع **F5** وبتجربة التنقل بين البيانات باستخدام المستعرضين معا نلاحظ أن المستعرضين يعمل كل منهما بشكل مستقل عن الآخر وهذا يكون مفيد إذا كان لدينا جدولين يحتويان على بيانات كثيرة ونريد مقارنتهما يدويا وبحفظ البيانات يكون الشكل كالتالي



هنا نتعامل مع السجل الثاني

هنا نتعامل مع السجل الثاني

وستجد في التمرين التالي المثال التطبيقي على الحالتين السابقتين للتعامل مع الـ **DataGrid**

الرجاء الإلمام بطريقة عمل قواعد البيانات الخاصة بكم والتعامل معها من خلال تعلم برنامج مايكروسوفت أكسس 2007 ويمكنك تحميل الشرح من الرابط التالي

تحميل التمرين الخامس والأربعون

Microsoft Access 2007 Part 1  
Download

Microsoft Access 2007 Part 2  
Download



تحميل الملف من هنا

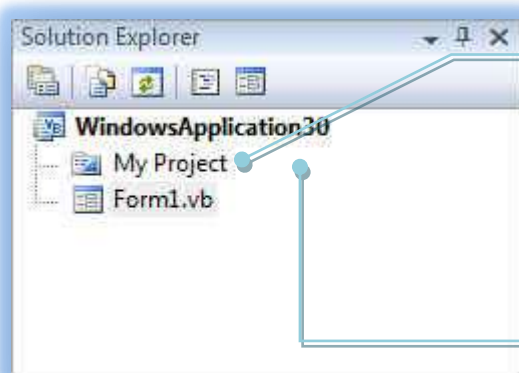
بعد الانتهاء من التمارين السابقة نلاحظ أنه لا يمكنك التعامل مع المشاريع إلا من خلال وجود برنامج فيجوال بيسك 2008 على جهازك وهو موجود بالفعل لأنك أنت مستخدمة والذي تقوم بإنشاء المشاريع عليه ولكن ماذا لو أردت إن يعمل برنامجك على جهاز آخر كجهاز مستخدم عادي لا يتعامل مع الفيجوال بيسك في هذه الحالة لابد من تحويل برنامجك إلى برنامج له **SETUP** ليقوم المستخدم بتنصيبه في جهازه ليقوم بالتعامل معه كحال كل برامج الويندوز ويتم ذلك كالتالي

بقوم باختيار أي مشروع من المشاريع التي سبق التعامل معها ونقوم بالعمل على تحويله إلى برنامج له **SETUP** لنتمكن من تشغيله على أي جهاز آخر وأنا هنا قد اخترت المشروع في التمرين السابع والعشرون للتعامل معه وتنفيذ الخطوات عليه ويتم اتباع الطرق التالية

تختلف الطريقة في الفيجوال بيسك 2008 عنة في النسخ القديمة فيجب أولاً تحميل الملف **MDAC28** وقفل برنامج الفيجوال بيسك 2008 ونقم بوضعه في المسار التالي على جهازك

« Local Disk (C:) » Program Files » Microsoft SDKs » Windows » v6.0A » Bootstrapper » Packages »

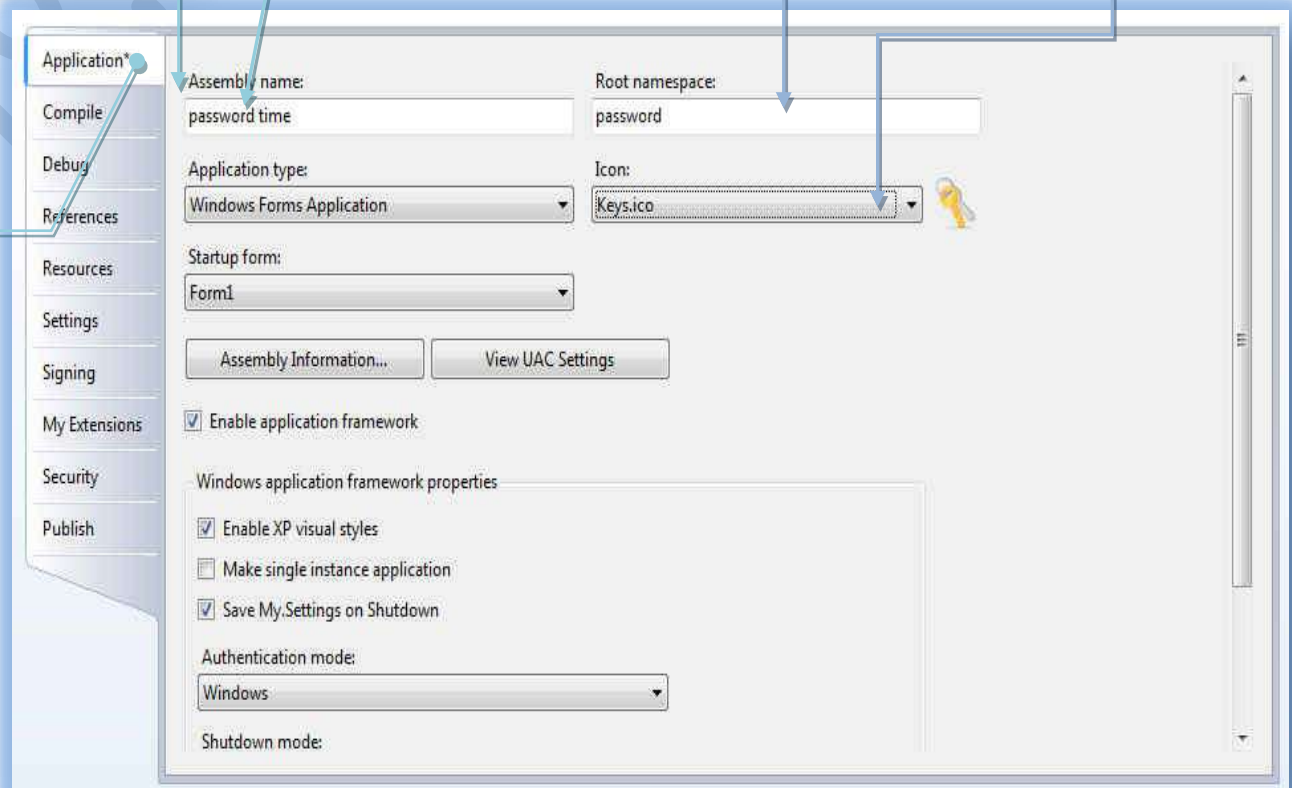
ثم نقوم بفتح برنامج الفيجوال بيسك 2008 وفتح المشروع السابع والعشرون أو أي مشروع تريد التعامل معه لجعله برنامج له الخاصية **SETUP** ونتبع الخطوات التالية



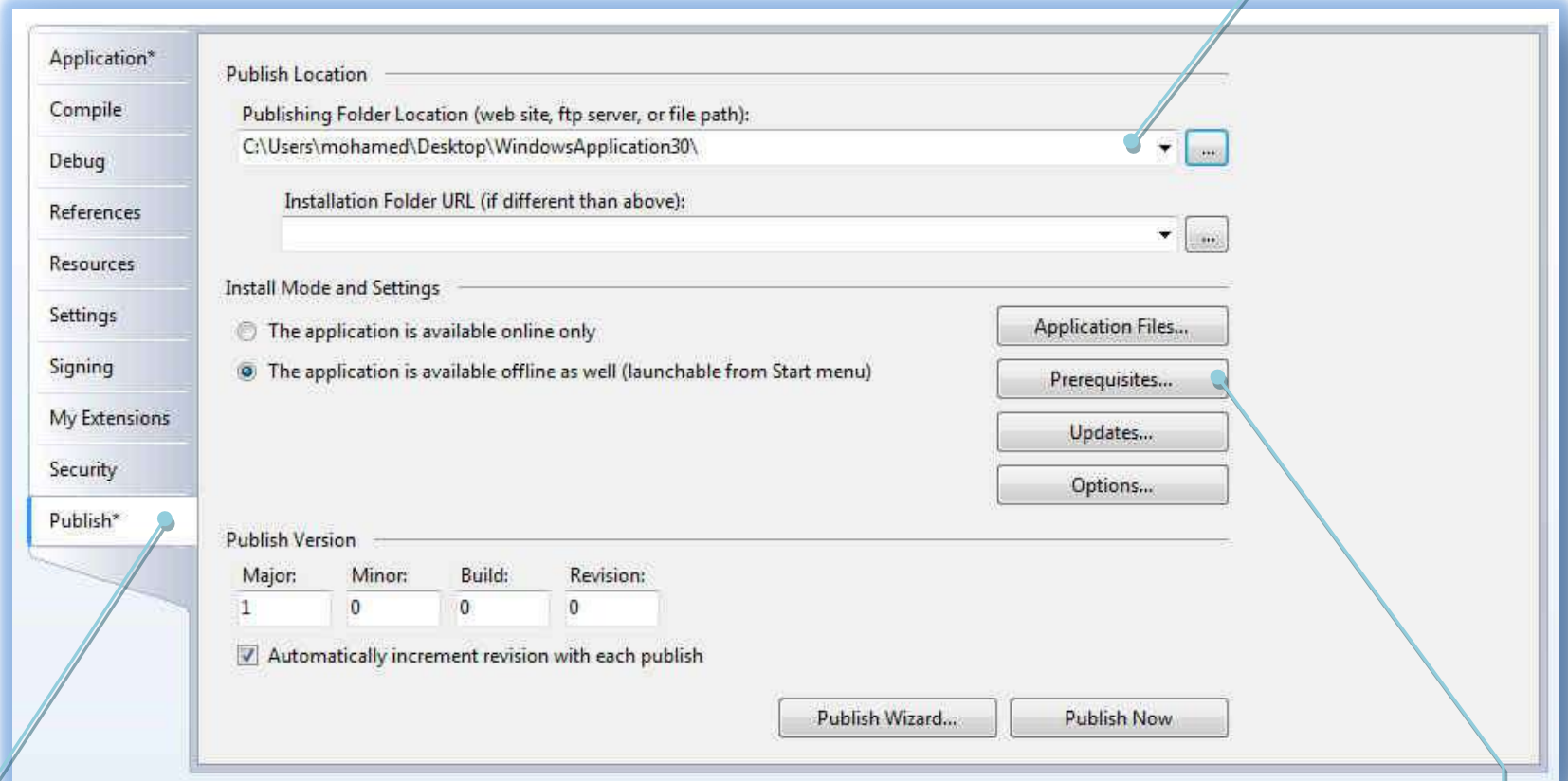
نقوم بالنقر مرتين على **My Project** في Solution Explorer

هنا يتم اختيار اسم الملف بعد التنصيب إلى الجهاز والذي يظهر في شريط القوائم ونختار أيضاً شكل إيقونته من هنا

التبويب Application



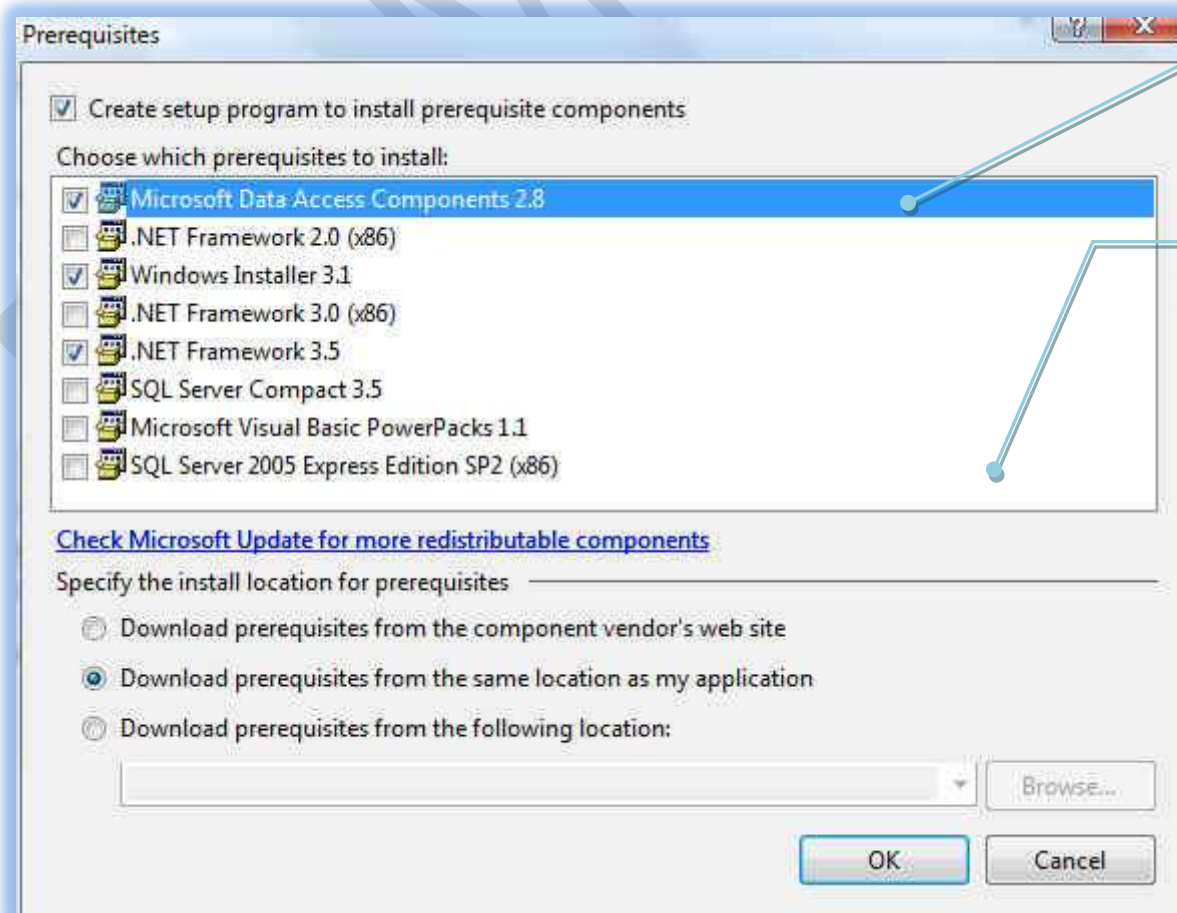
يتم اختيار مسار المشروع الذي تريد تحويله إلى برنامج **SETUP** من جهازك



من التبويب Publish

يتم النقر على الزر Prerequisites لتظهر لنا الشاشة التالية

إن لم تظهر لك هذه الإضافة فهذا دليل على أنك لم تقم بوضع الملف **MDAC28** السابق ذكره في مساره المحدد



نقوم باختيار باقي الإضافات كما هي تماما ثم النقر على الزر **ok**

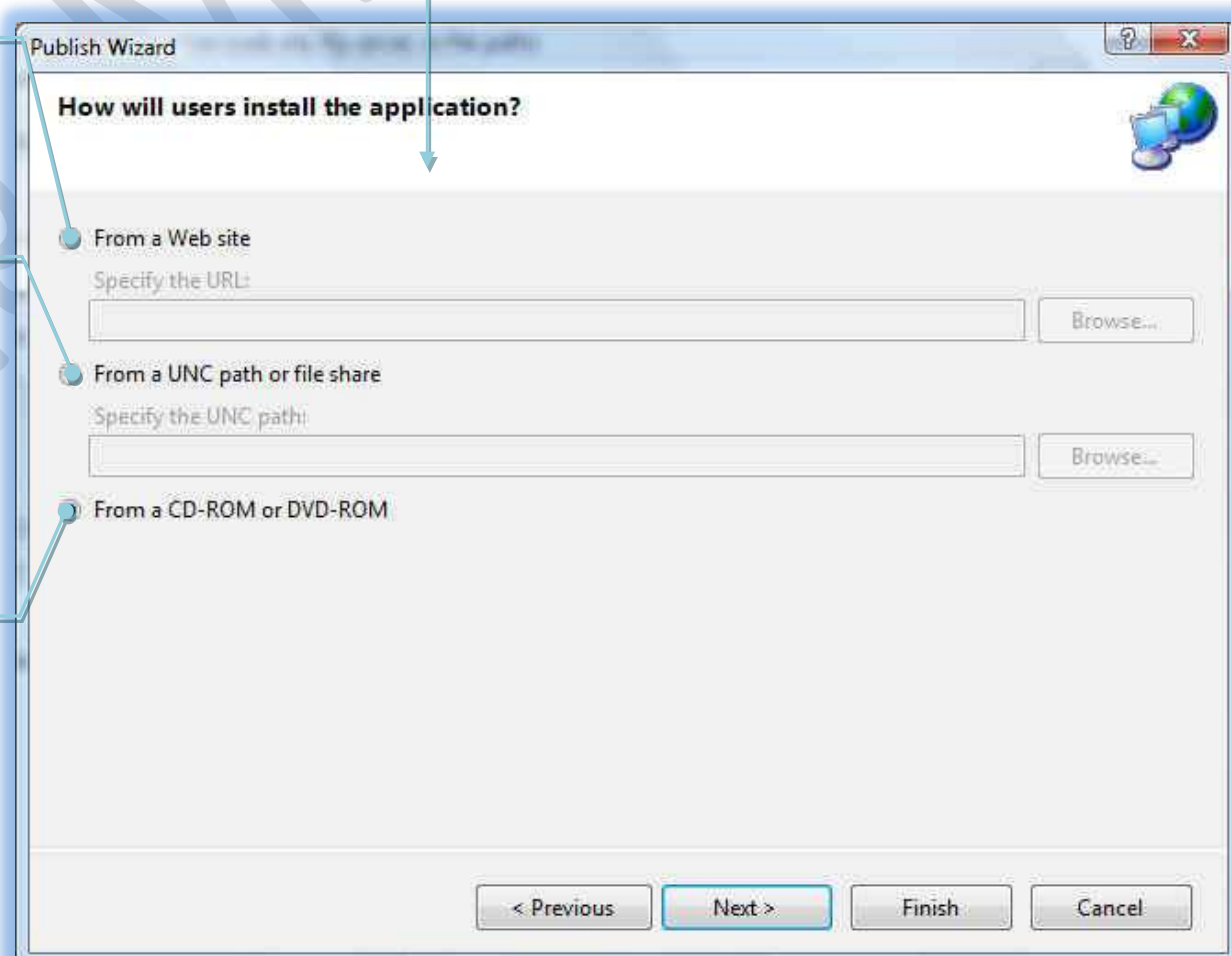
بعد النقر على الزر موافق نلاحظ اختفاء الرسالة السابقة والعودة إلى النافذة الأولى لنقوم باختيار **Publish Wizard** لتظهر لنا النافذة التالية

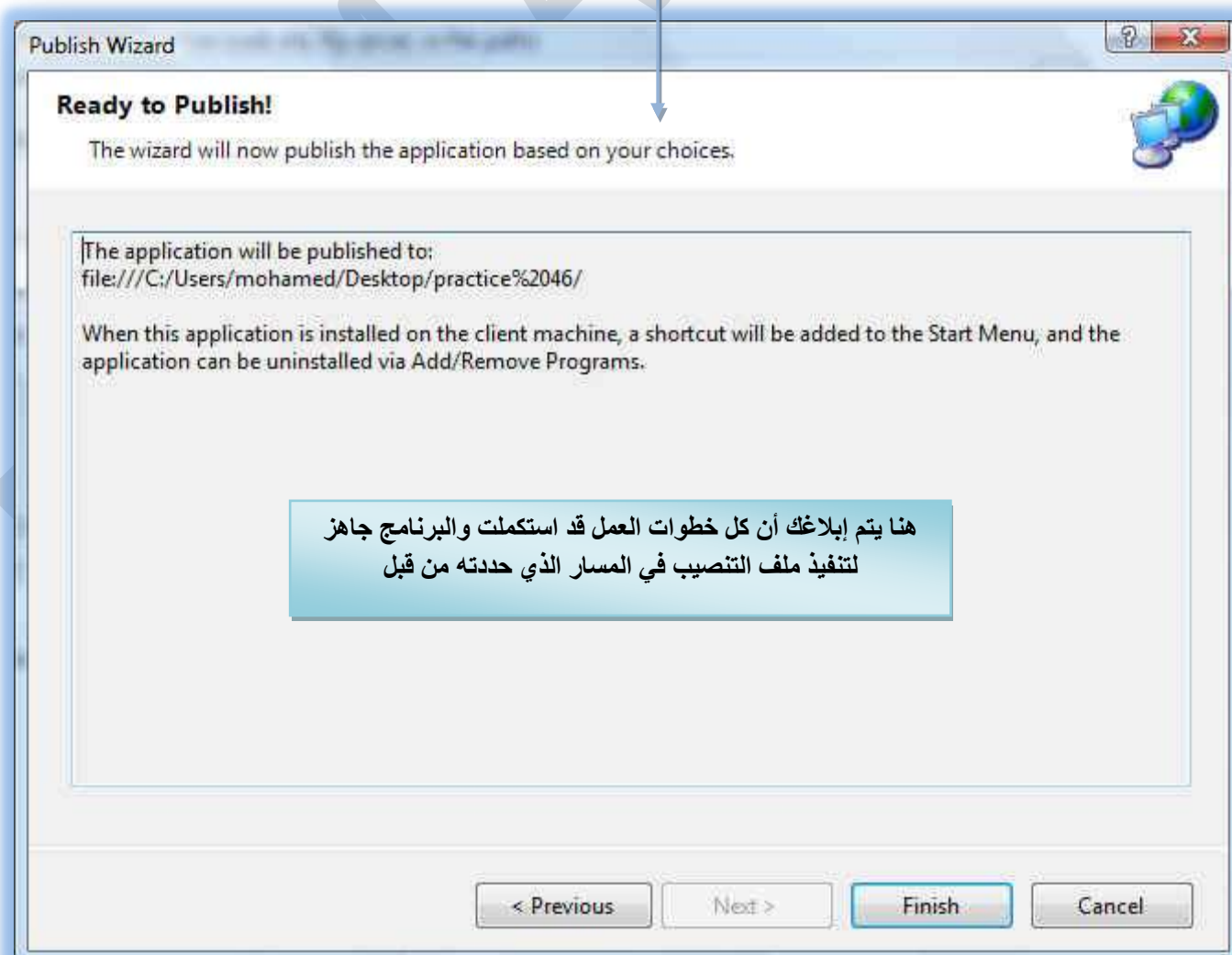
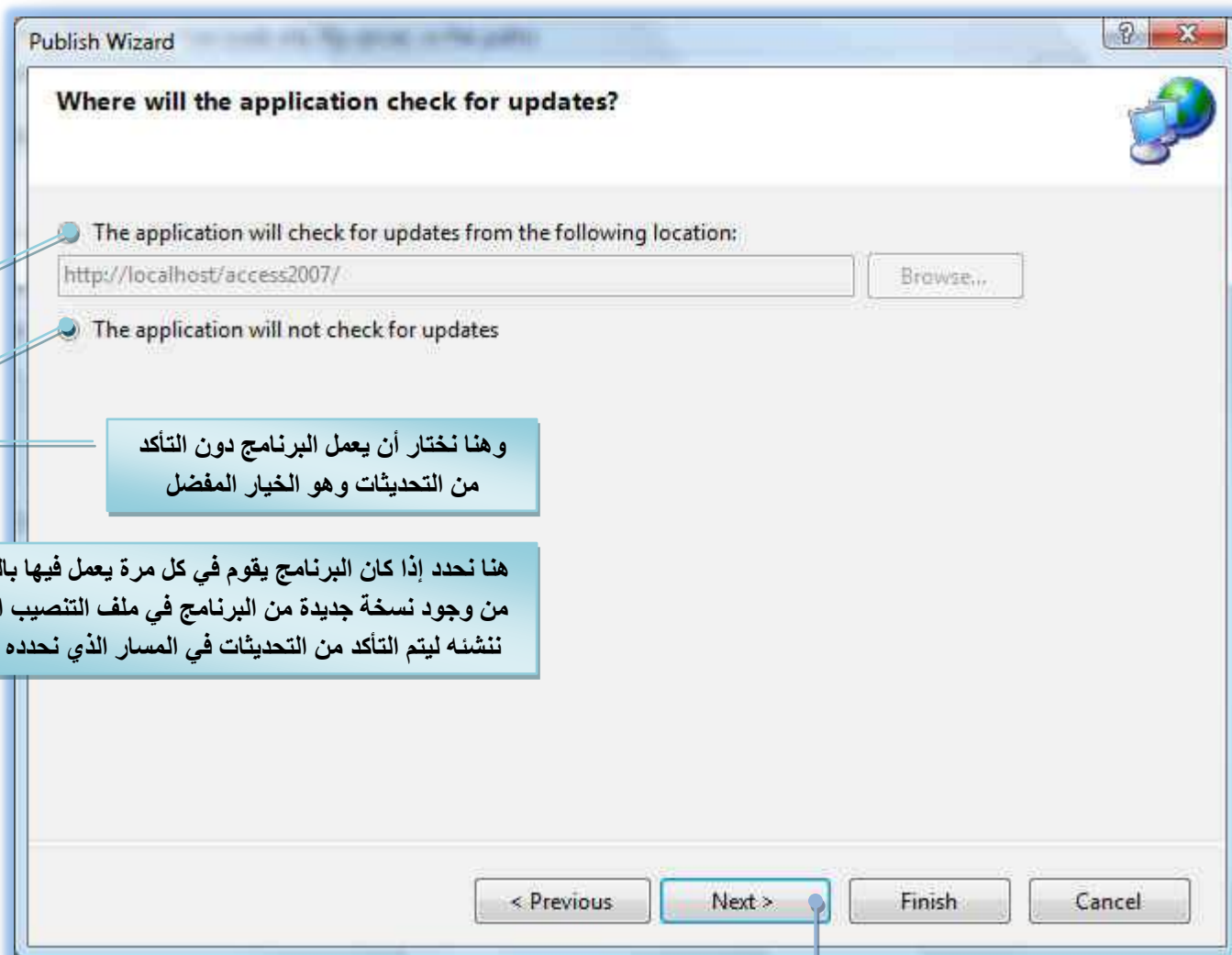


الخيار الأول إذا أردت أن يكون مصدر التنصيب من الإنترنت

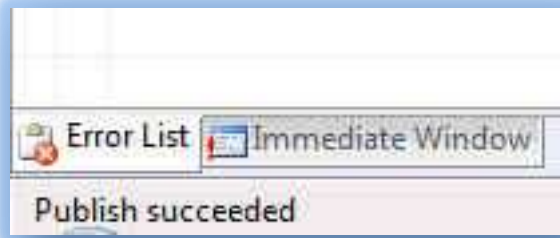
الخيار الثاني إذا كان مصدر التحديث موجود على شبكة محلية

الخيار الثالث للتنصيب المباشر من السي دي أو من القرص الصلب

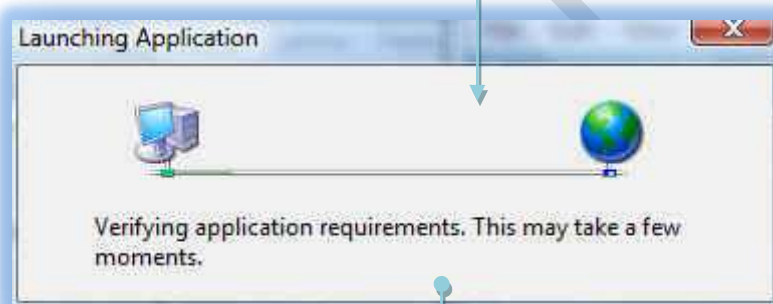
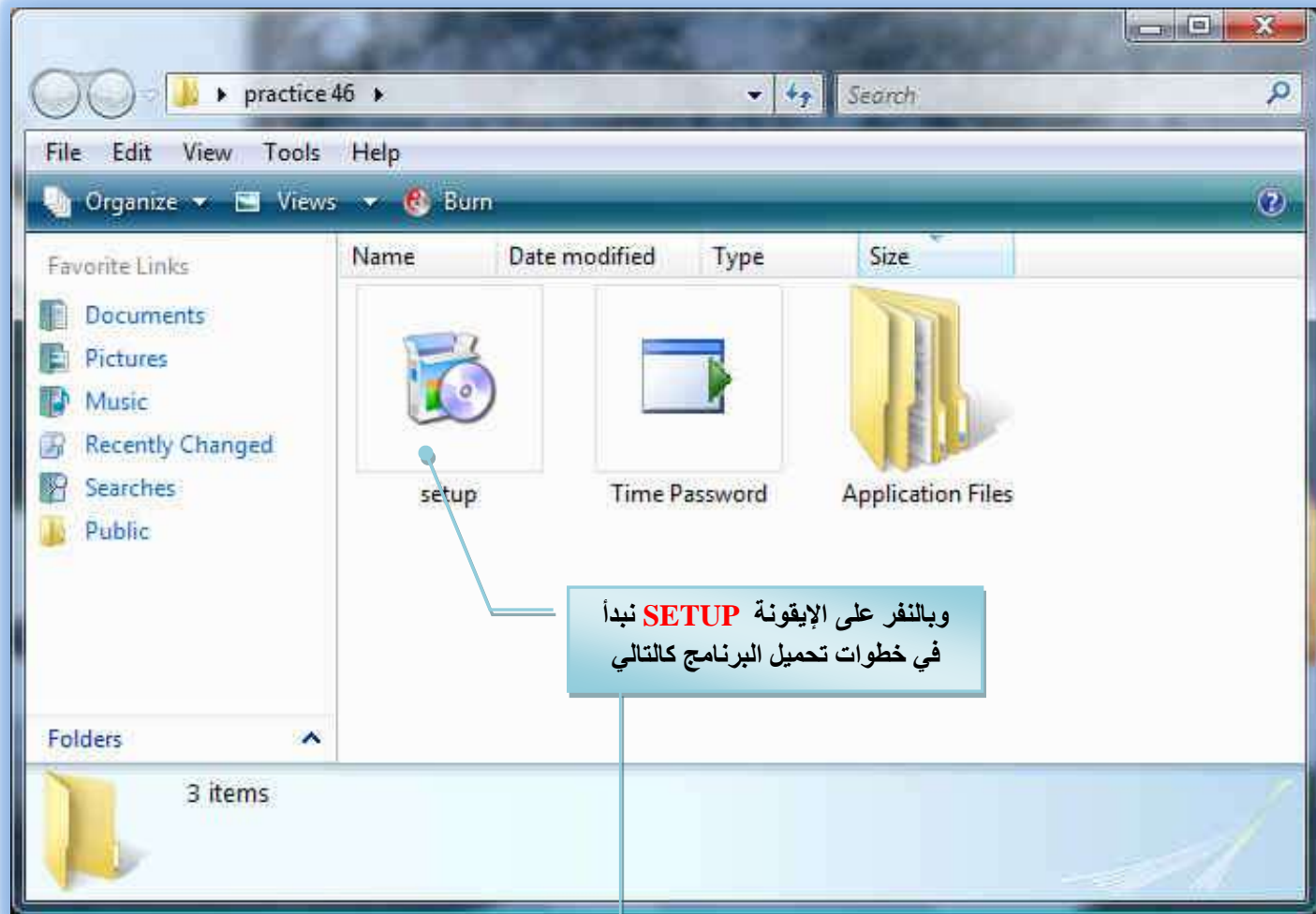








وبالنقر على **Finish** لانتهاؤ الخطوات ومتابعة شريط معلومات البرنامج أسفل يسار شاشة برنامج فيجوال بيسك 2008 لتجد هذه الرسالة وقد ظهر فيه بعد الانتهاء من تنفيذ الخطوات وليقوم بفتح نافذة ملف البرنامج لتشاهد ملفات الإعداد الخاص ببرنامجك من خلالها كالتالي



بعد الانتهاء يقوم البرنامج بالتشغيل تلقائياً ويتم إضافته في القائمة START كبرنامج جديد ضمن برامجك ويكون أسمة وأيقونته كما تم الإعداد لها من قبل



تحميل التمرين السادس والأربعون



ليتم تنفيذ البرنامج كالتالي و بالوصول إلى هذه النتيجة نكون قد انتهينا من شرح برنامج فيجوال بيسك 2008 من بداية تصميم المشروع إلى هذه المرحلة

كلمة السر هي كما أعددناها من قبل في التمرين وهي  
1612

انتمى

لتحميل جميع تمارين الكتاب في رابط واحد فقط

من هنا

أو

من هنا

أرجو وان أكون قد وفقت في تبسيط البرنامج و شرحه للاستفادة منه في حياتنا العملية

ولا أسألكم خير صالح الدعاء

لأي استفسار أرجو مراسلتي على البريد الإلكتروني التالي

M,aboelela@hotmail.com

للحصول على المزيد من مؤلفاتي أرجو تحميل ملف الكتب من الرابط التالي

من هنا

والى اللقاء إن شاء الله تعالى قريباً في شرح برنامج

