



اتقن لغة الفيجوال بيسك 2013 بالأمثلة

Visual Studio 2013

الاستاذ حمزة جميل إبراهيم



كلمة

في البداية لآبد لي من تقديم الشكر والعرفان لكل من وقف معي وساندني وشجعني على اتمام هذا الكتاب المتواضع.

اتممت هذا الكتاب بجهود شخصية وعمل متواصل عسى وان يفلح هذا الكتاب المتواضع برفد اصحاب الاختصاص ببعض المعلومات التي تنفعهم في انشاء برامج تطبيقية معينة.

قد يحتوي هذا الكتاب على بعض الطرق او الشفرات البرمجية الغير واضحة يرجى المراسلة على

.... hmzjl122@gmail.com

والله ولي التوفيق

الاستاذ: حمزة جميل ابراهيم.

العنوان: العراق - النجف الاشرف



hamza.jameel.568

المقدمة

الفصل الأول: المتتارفة

المشروع 1: الحاسبة البسطة

التعامل مع الأدوات التالفة:

نموذج form

صندوق النص Textbox

الدالات labels

ازرار الامر buttons

اسناد القفم الى خاصة أداة

دالة Val

العمليات الرفاضية الاساسفة

ابسط اشكال If

المشروع 2: عبارات الشرط (العملة)

التعامل مع الأدوات التالفة:

النموذج Form

صندوق النص Text Box

الدالات Label

زر الامر Button

شكل الجملة الشرطفة If

شكل جملة الشرط Select Case

التعامل مع المتغفرات

المشروع 3: التعامل مع المتغفرات

التعامل مع المتغفرات Variables

المتغفرات الخاصة باجراء.

المتغفرات الساكنة Static الخاصة باجراء.

المتغفرات العامة لنموذج.

المتغفرات العامة لمشروع.

التعامل مع عدة نماذج.

استخدام الواحدات النمطفة Module.

المشروع 4: عبارات الشرط (سفرة سفاحفة)

التعامل مع الأدوات التالفة:

زر الخيار RadioButton

زر التحقق CheckBox

صندوق المجموعة GroupBox

Select و if الجمل الشرطية

المشروع 5: حلقات التكرار

جمل التكرار

For

Do .. while

مسائل رياضية

المشروع 6: الدوال الحوارية

دوال حوار المستخدمين

دالة الإدخال InputBox

دالة الرسائل MsgBox

حلقة While

المشروع 7: تعدد الواجهات

التعامل مع الأدوات التالية

عمل شريط القوائم باستخدام Menustrip

Multiple InterFace

المشروع 8: توليد أرقام عشوائية

التعامل مع الأدوات التالية

دالة Ran

صندوق القائمة ListBox

تكرار Do..loop While

المشروع 9: نوافذ الحوار

التعامل مع نوافذ الحوار Dialog

نافذة الالون ColorDialog

نافذة الخطوط FontDialog

نافذة حفظ ملف SaveFileDialog

نافذة فتح ملف OpenFileDialog

استخدام دالة My.Computer

المشروع 10: التعامل مع الدوال

اجراء العمليات الرياضية التالية:

مضروب العدد (المفكوك)

مجموع العدد

الأعداد الأولية

المشروع 11: التعامل مع صندوق القائمة

التعامل مع ListBox

التعامل مع MessageBox

المشروع 12: الملفات النصية

التعامل مع ما يأتي:

عنصر TabControl

مكتبة System.IO

الكتابة على الملفات باستخدام StreamWriter

القراءة من الملفات باستخدام StreamReader
المشروع 13: دوال التعامل مع النص

التعامل مع الدوال التالية:

Length
Substring
Replace
Trim, RTrim, LTrim
ToLower
ToUpper
Asc
Chr

المشروع 14: التعامل مع صندوق النص الغني

التعامل مع صندوق النص الغني RichTextBox:
اجراء العمليات التالية:

حفظ نص الصندوق في ملف نصي
فتح ملف نصي داخل الصندوق
البحث عن كلمات داخل الصندوق

المشروع 15: انشاء مفكرة (خطوات متقدمة)

التعامل مع الأدوات التالية:

صندوق النص الغني RichTextBox
شريط القوائم MenuStrip
قائمة زر الفأرة الايمن ContextMenuStrip

المشروع 16: انشاء حاسبة رقمية

التعامل مع الأدوات
التعامل مع الدوال الرياضية
التعامل مع تعليمات الخطأ

Resume
Catch

المشروع 17: التعامل مع أشرطة المسار

التعامل مع شريط المسار TrackBar
دالة RGB

المشروع 18: عرض صور (1)

التعامل مع صندوق الصورة PictureBox
التعامل مع نوافذ الحوار
التعامل مع ListBox

المشروع 19: رسالة بريد الكتروني

التعامل مع شريط التقدم ProgressBar
التعامل مع المؤقت Timer
التعامل مع Panel

التعامل مع Email
المشروع 20: عرض الصور (2)
التعامل مع TreeView
التعامل مع ImageList
التعامل مع SplitContainer
المشروع 21: التعامل مع الملفات (2) (LogIn)
التعامل مع الملفات
التعامل مع دول الحوار
المتحولات العامة
المشروع 22: تشغيل ملفات صوت وفيديو
التعامل مع WindowsMediaPlayer
التعامل مع TrackBar
التعامل مع SplitContainer
المشروع 23: الشبكة DataGridView
التعامل مع الشبكة DataGridView
التعامل مع نافذة OpenFileDialog
الحلقات المتداخلة
الملفات النصية
المشروع 24: الشبكة (التعامل مع Excel)
التعامل مع الشبكة DataGridView
جلب بيانات من Excel
تصدير بيانات الى Excel
المشروع 25: NotifyIcon
التعامل مع NotifyIcon
التعامل مع ContextMenuStrip
التعامل مع MessageBox
المشروع 26: الأحداث (1)
المشروع 27: الأحداث (2)
التعامل مع حدث DragDrop
التعامل مع حدث DragEnter
المشروع 28: التعامل مع قواعد البيانات
التعامل مع DataSet

تعتبر الفيجوال بيسك6 Visual Basic 6 البوابة الرئيسية لدخول عالم البرمجة الواسع والمتشعب. والاختصاصي الذي يتقن البرمجة بلغة الفيجوال بيسك قد لا يجد صعوبات كبيرة في تعلم واحتراف الاصدارات اللاحقة لهذه اللغة كالفيجوال بيسك (VB) 2013، فيمكنه بناء تطبيقات في مختلف المجالات دون مشاكل كما ان أغلب الايعازات والاورامر الموجودة في هذه الاصدار الحديث للغة يشبه الى حد ما الاوامر الموجودة في الفيجوال بيسك6 مع بعض الاختلافات البسيطة بل يمكن القول بان ايعازات هذه اللغة هي نتاج تطوير وتوسعة ايعازات الفيجوال بيسك6. كذلك اذا تطلب الأمر البرمجة بلغات اخرى سيجد من السهولة التأقلم معها.

كذلك الحال بالنسبة لغير الاختصاصي سيجد الفيجوال بيسك VB هي الاداة الطيبة ذات المرونة الكبيرة في بناء التطبيقات التي يحتاجها في عمله لان السهولة والبساطة التي تتميز بها لغة الفيجوال بيسك VB تسمح له بسرعة تعلمها والغوص في عالم البرمجة الشيق والممتع.

كما لا يخفى على الجميع مدى الطلب على مبرمجي الفيجوال بيسك يكفي الدخول الى النت والبحث عن فرصة عمل لتجد عشرات الألاف من الفرص في مختلف انحاء العالم.

ان جُل ما يحتاج اليه الطالب لتعلم هذه اللغة هو برمجة وتنفيذ الكثير من المشاريع ولذا اعتمت في هذا الكتاب على مجموعة من المشاريع التعليمية التي تُغطي أغلب الادوات التي تتوفر في هذه اللغة باتباع ابسط الطرق مع التركيز على الاساسيات التي يحتاجها المبرمج في اغلب الحالات، لذا يجب على الطالب ان يكتشف بنفسه الحالات الخاصة التي يحتاج اليها في مشاريعه.

يتألف هذا الكتاب من قسمين الأول يضم مجموعة من المشاريع التعليمية والتي يمكن تنفيذها مع الاستعانة بالكتاب، والقسم الثاني من الكتاب يضم اساسيات التعليمات البرمجية في اللغة.

يُرافق الكتاب قرص يحوي على جميع مشاريع الكتاب.

الفصل الأول

المشاريع (Projects)

المشروع 1

الحاسبة البسيطة

ستتقن من خلال المشروع

التعامل مع الأدوات التالية:

- نموذج Form
- صندوق النص TextBox
- الدلالات Labels
- ازرار الامر Buttons
- اسناد القيم الى خاصية أداة
- دالة Val
- العمليات الرياضية الاساسية
- ابسط اشكال If

أهداف المشروع

عند تنفيذ المشروع تظهر الواجهة التالية التي تسمح للمستخدم بتنفيذ العمليات الحسابية الاساسية في الفيچوال بيسك على عددين فقط.



مراحل المشروع:

- 1- انشئ مشروعاً جديداً.
- 2- اضبط خصائص النموذج كالتالي:

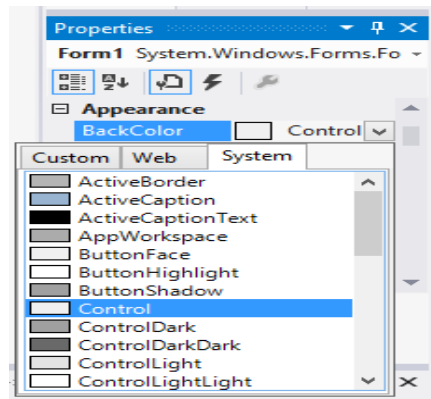
Form1	
name	Frm1
BackColor	الاقتراضي

Text	حاسبة
RightToLeft	Yes
RightToLeftLayout	True
Icon	اختر اي صورة تعجبك (ساتركها كما هي)

الخصائص:

Name: تمثل اسم النموذج ، وعند انشاء اي كائن فان الفيچوال بيسك تضع له اسماً افتراضياً، وتستخدم هذه الخاصية من قبل المبرمج مع الاكواد البرمجية فقط ولا تظهر للمستخدم خلال مرحلة التنفيذ ، ويفضل تغيير هذه الخاصية لتسهيل عمل المبرمج وتستخدم اسماء ذات دلالات تتلائم مع طبيعة عمل المشروع.

BackColor: تحدد لون أرضية النموذج ، ولإختيار اي لون لأرضية(خلفية) النموذج انقر على السهم المقابل للخاصية ستظهر لك ثلاث تبويبات Custom , Web , System من خلال هذه التبويبات يمكنك اختيار اللون المفضل لديك كألوان النظام او الخاصة او الشفافة .



Text: تتضمن عنوان النموذج (يظهر أعلى النموذج).

Icon: تستخدم لوضع ايقونة خاصة بالنموذج. وستكون هذه الايقونة للملف التنفيذي (EXE). عند النقر ستظهر نافذة يتم من خلالها اختيار الصورة المطلوبة.

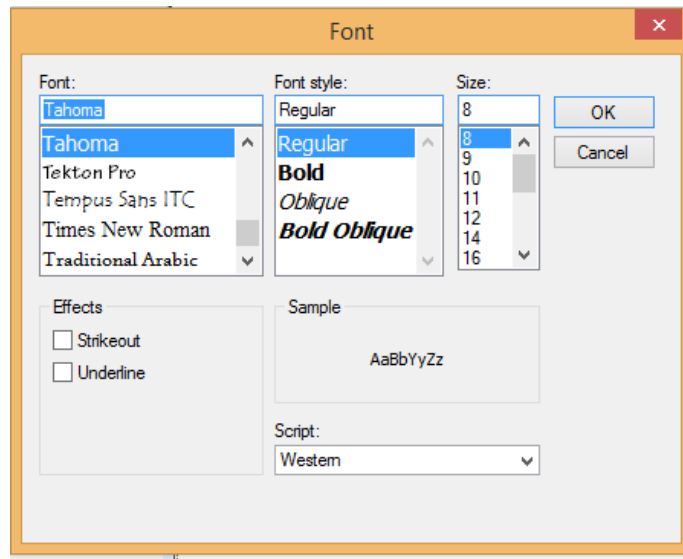
RightToLeft: تغيير اتجاه عنوان النموذج أي جعله من اليمين الى اليسار او بالعكس وتأخذ هذه الخاصية قيم منطقية.

كائنات المشروع:

3- اصف ثلاث دلالات Labels واضبط خصائصها بالشكل التالي:

Label1		Label2		Label3	
Name	الافتراضي	Name	الافتراضي	Name	الافتراضي
BackColor	الافتراضي	BackColor	الافتراضي	BackColor	الافتراضي
Text	العدد الاول	Text	العدد الثاني	Text	الناتج
Font	Tahoma; 10pt	Font	Tahoma; 10pt	Font	Tahoma; 10pt

الخاصية **Font** تمكنك من اختيار نوع ونمط الخط الذي ترغب به . وذلك من خلال النقر على المربع المنقط المقابل لها فتظهر نافذة تضم عدة أنواع من الخطوط بالإضافة الى الانماط الخاصة بكل نوع وحجم الخط. كما في الشكل التالي.



4- اصف ثلاثة صناديق نص TextBox واضبط خصائصها كالتالي:

Text Box 1		Text Box 2		Text Box 3	
Name	الافتراضي	Name	الافتراضي	Name	الافتراضي
TextAlign	Center	TextAlign	Center	TextAlign	Center
Font	Tahoma; 10pt	Font	Tahoma; 10pt	Font	Tahoma; 10pt
ForeColor	Maroon	ForeColor	Maroon	ForeColor	Maroon
Text		Text		Text	

الخاصية **TextAlign**: يقصد بها محاذاة النص في صندوق النص:

(الوسط Center, لليمين Right, للييسار Left)

الخاصية **ForeColor** تحدد لون الخط.

الخاصية **Text** تضم النص الذي يظهر في صندوق النص وعادة الفيچوال بيسك 2013 لا يضع اي قيمة افتراضية لها فتكون فارغة

5- اصف ثمانية أزرار أوامر Buttons واضبطها كالتالي:

Button1		Button9	
Name	الافتراضي	Name	الافتراضي
Text	+	Text	E&xit
Font	Tahoma; 10pt	Font	Tahoma; 10pt

ملاحظة: نضع المحرف & بعد الحرف للتسطير فعند كتابتها في العنوان تظهر هكذا Exit .

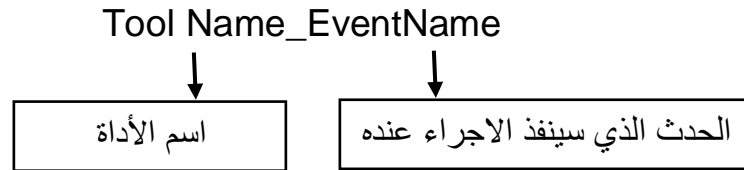
6- التعليمة البرمجية لأزرار الأمر

على كل زر أمر يجب كتابة الأكواد (الشيفرة البرمجية) التي سيتم تنفيذها في مرحلة التنفيذ عندما ينقر المستخدم على زر الأمر.

لبرمجة زر الجمع انقر مرتين على الزر ستظهر لك التعليمة البرمجية التالية

- **Private**: تعني ان الاجراء لا ينفذ الى على النموذج نفسه.
- **Sub**: تحدد بداية الاجراء ويليه اسمها.
- **Button1**: اسم زر الامر
- **Click**: طريقة التنفيذ تتم بالنقر على زر الامر.
- **End Sub**: نهاية الاجراء.

يأخذ اسم الاجراء الشكل التالي:

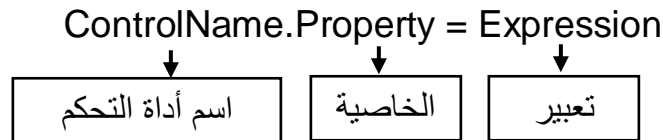


ملاحظة الاجراء يرتبط مع الأداة عن طريق اسمها، فإذا غيرت اسم الأداة بعد كتابة إجراء معين سيضيع الاجراء عنها. لذا بعد تغيير اسم اي أداة تحكم يجب تعديل أسماء جميع الاجراءات المكتوبة عليها.

❖ لتنفيذ عملية الجمع، اكتب الاجراء في تعليمة الاسناد التالية :

`TextBox3.Text = Val(TextBox1.Text) + Val(TextBox2.Text)`

برمجة تعليمة الاسناد = تستخدم لنسب قيمة الى خاصية أداة التحكم وتأخذ الشكل:



- التعليمة التالية تضع العدد 33 في صندوق النص:

`Textbox3.text=33.`

- التعليمة التالية تضع ناتج تعبير حسابي في صندوق النص:

`Textbox3.text=9-3/2+5.`

- التعليمة التالية تضع نص في صندوق النص:

`Textbox.text="Welcome in Visual Studio 2013".`

الدالة Val تستخدم لتحويل النص الى قيمة رقمية.

- مثال `val(50)` فان الناتج سيكون 50. اما `val("kaka")` فان الناتج هو صفراً.
- ❖ في زر الطرح اكتب التالي:

`TextBox3.Text = Val(TextBox1.Text) - Val(TextBox2.Text)`

❖ في زر الضرب اكتب التالي:

```
TextBox3.Text = Val(TextBox1.Text) * Val(TextBox2.Text)
```

❖ في زر القسمة اكتب التالي:

```
If Val(TextBox2.Text) = 0 Then  
    TextBox3.Text = "ERROR"  
Else  
    TextBox3.Text = Val(TextBox1.Text) / Val(TextBox2.Text)  
End If
```

إذا كان ناتج المقسوم عليه يساوي صفر ستظهر كلمة ERROR في صندوق الناتج.

أحد أشكال **التعليمة الشرطية if**:

If Condition then

Statement(s)

Else

Statement(s)

End if

حيث ان الشرط Condition عبارة عن مقارنة ترجع اما صح true او خطأ false. اذا كانت نتيجة الشرط صح وبهذا سننفذ عبارة if . واذا كانت نتيجة الشرط خطأ فانه سننفذ عبارة Else.

في مثال الشرط:

```
Val(textbox2.text)=0
```

اذا كانت قيمة النص في صندوق النص textbox2 صفراً، فان الفيچوال ببسك سينفذ تعليمة if ويكون الناتج في textbox3 هو ERROR كما في الشكل التالي

```
Textbox3.text="ERROR"
```

اما اذا كانت قيمة النص textbox2 لا تساوي صفر فان الفيچوال ببسك سينفذ التعليمة

```
Textbox3.text= val(textbox1.text) / val(textbox2.text)
```

❖ في زر باقي القسمة اكتب التالي:

```
If Val(TextBox2.Text) = 0 Then  
    TextBox3.Text = "ERROR"  
Else  
    TextBox3.Text = Val(TextBox1.Text) mod Val(TextBox2.Text)  
End If
```

❖ في زر الرفع لقوة اكتب التالي:

Textbox3.text=val(textbox1.text)^ val(textbox2.text)

❖ في زر المسح Clear اكتب التالي:

Textbox1.text=""

Textbox2.text=""

Textbox3.text=""

(فان "" تعني ا فراغ المحتويات)

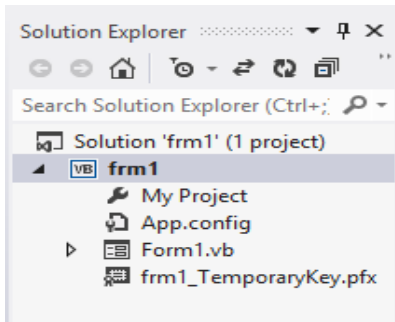
❖ في زر Exit اكتب التالي:

END

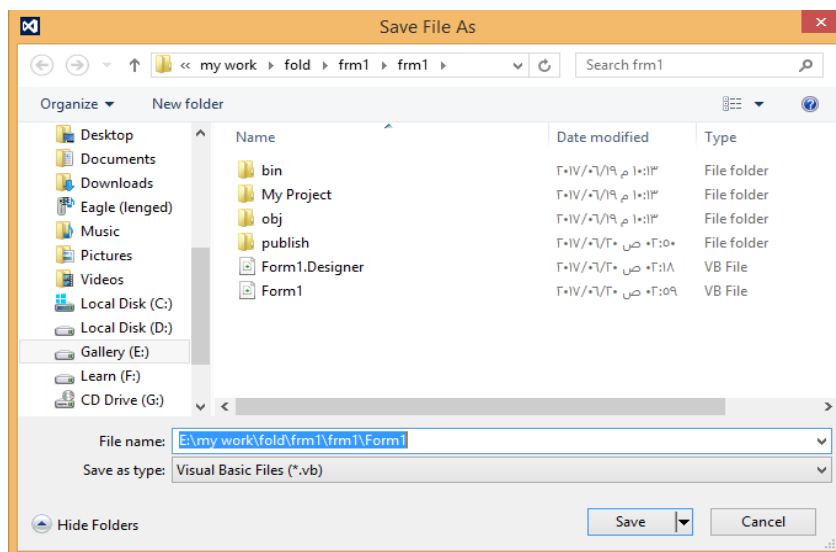
وتعني انتهاء التنفيذ.

7- حفظ المشروع:

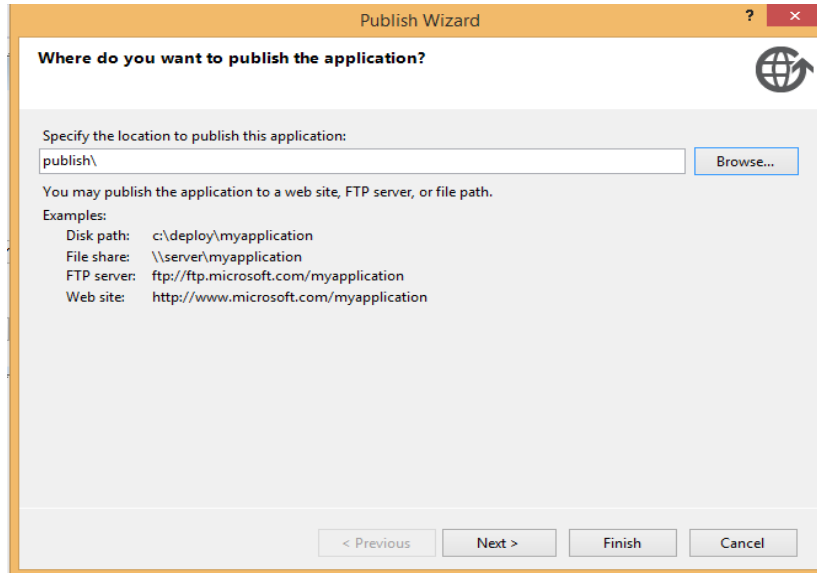
- يولد الفيجوال بيسك عدد من الملفات لكل مشروع. لذا فان الفيجوال بيسك تقوم تلقائياً بإنشاء مجلد لحفظ ملفات المشروع فيه.
- يخزن النموذج بملف تنتهي باللاحقة vb
- لمعاينة مكونات المشروع انقر على Solution Explorer



- حفظ المشروع في الفيجوال بيسك يتم من خلال (save form1.vb) للنموذج أما إذا أردنا حفظ المشروع باسم جديد نختار الأمر (Save AS) ستظهر نافذة تسمح لك بإختيار مكان الحفظ. يُمكن حفظ كل مكونات المشروع من خلال (Save All).



- لعمل نسخة تنفيذية من المشروع، من قائمة BUILD اختر الامر Publish ستظهر لك النافذة التالية:



اضغط على Browse لتحديد مكان الخزن ثم اضغط على Finish لاتمام العملية عندها سيتحول المشروع الى ملف قابل للتنصيب على اي حاسوب وجاهز للعمل.

المشروع 2

عبارات الشرط (العُملة)

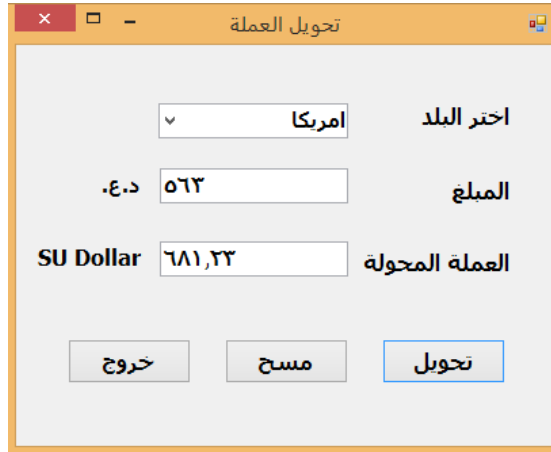
سنتقن من خلال المشروع

التعامل مع أدوات التحكم التالية:

- النموذج Form
- صندوق النص TextBox
- الدلالات Label
- زر الامر Button
- شكل الجملة الشرطية If
- شكل جملة الشرط Select Case
- التعامل مع المتغيرات

اهداف المشروع

بناء واجهة لتحويل مبلغ من المال بالعملة العراقية الى ما يكافئه من العملات لعدد من الدول موجودة ضمن قائمة يتم اختيارها من قبل المستخدم.



مراحل المشروع

- 1- انشئ مشروعاً جديداً.
- 2- اضبط خصائص النموذج كالتالي:

Form	
Name	Frm
Text	تحويل العملة
RightToLeft	Yes
RightToLeftLayout	True

3- اضع عنصرين Text Box واضبطهما كالتالي:

TextBox1		TextBox2	
name	TxtAmount	Name	TxtConverted
Font	Tahoma; 12pt; style=Bold	Font	Tahoma; 12pt; style=Bold

4- اضع خمسة انواع من Label واضبط خاصية Text لكل منهم ك(اختر البلد، المبلغ، العلمية المحولة، د.ع. ، -)

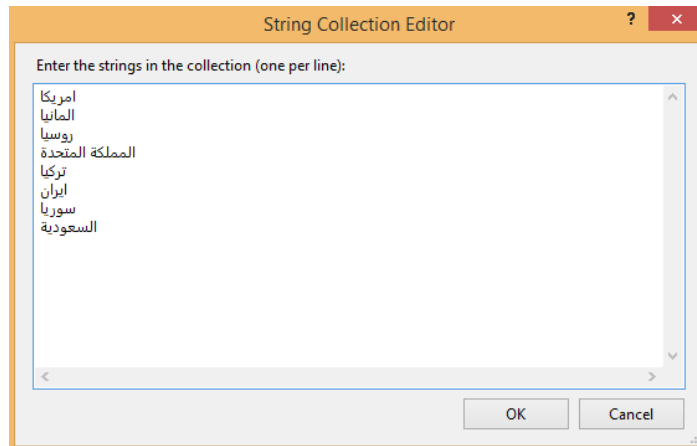
ملاحظة يفضل تسمية الأدوات التي ستتعامل معها برمجيًا باسماء ذات دلالة، كما يلي:

العنصر	التسمية
Form	Frm
textbox	Txt
Label	Lbl
Button	Btn
Combobox	Cbo

5- اضع صندوق ComboBox واضبطها كما يلي:

ComboBox	
Name	CboCountry
Font	Tahoma; 12pt; style=Bold
Items	امريكا المانيا روسيا المملكة المتحدة تركيا ايران سوريا السعودية
Text	اختر واحد...

تستخدم خاصية **Items** لملئ القائمة بالعناصر المطلوبة كما في الشكل التالي:



6- اضع ثلاثة أزرار أمر Button واضبط الخصائص كالتالي:

Button1		Button2		Button3	
Name	BtnConvert	Name	BtnReset	Name	BtnExit
Font	Tahoma; 12pt; style=Bold	Font	Tahoma; 12pt; style=Bold	Font	Tahoma; 12pt; style=Bold
Text	تحويل	Text	مسح	Text	خروج

7- ضع الشيفرات التالية في قسم تصريحات Class Form

```
Dim Dollar_US As Double = 1.21
Dim Euz_German As Double = 2.32
Dim Robiah_russia As Double = 3.123
Dim Pound_UK As Double = 2.134
Dim Lairah_Turkey As Double = 1.543
Dim Real_Iran As Double = 0.356
Dim Lairah_syria As Double = 0.341
Dim Real_saudi As Double = 3.12
```

الشيفرة السابقة تضمنت الاعلان عن متغيرات واسناد قيم خاص بكل منها.

ملاحظة عند تعريف المتغيرات في قسم التصريحات الخاصة بالنموذج يمكننا ذلك من استخدامها في أي إجراء يكتب داخل النموذج وتسمى في هذه الحالة بالمتغيرات العامة للنموذج ولا يمكن استخدامها في الاجراءات الاخرى خارج النموذج. في المشروع التالي ساتطرق الى توضيح الاعلان عن المتغيرات بالتفصيل.

8- برمجة زر "تحويل"

يعتمد هذا الإجراء على اختيار البلد من قبل المستخدم من القائمة المنسدلة ليتم إعطاء النتائج المتعلقة به. شكل العبارة الشرطية if ويستخدم هذا الشكل اذا كان لدينا عدة احتمالات، المطلوب تنفيذ واحد منها.

If Condition1 then

Statement(s)1

Elseif Condition2 then

Statement(s)2

.

.

.

Else

Statement(s)n

End if

يقوم الفيچوال ببسك بإختبار الشرط الأول Condition1 إذا كان False، فإنه ينتقل لإختبار الشرط الثاني Condition2 وهكذا الى أن يصل الى الشرط المتحقق، عندها سيقوم بتنفيذ ال Statement الخاصة بالشرط المتحقق. ثم ينتقل الى End if (أي انه لا يقوم بفحص اي شرط من الشروط التالية للشرط المتحقق و عليه فإنه سيقوم بتنفيذ تعليمات عبارة واحدة فقط).

```
Private Sub BtnConvert_Click(sender As Object, e As EventArgs) Handles
BtnConvert.Click
```

```

If CboCountry.SelectedIndex = 0 Then
    TxtConverted.Text = Math.Round(Dollar_US * TxtAmount.Text, 2)
    LblConverted.Text = "SU Dollar"
ElseIf CboCountry.SelectedIndex = 1 Then
    TxtConverted.Text = Math.Round(Euz_German * TxtAmount.Text, 2)
    LblConverted.Text = "DEM"
ElseIf CboCountry.SelectedIndex = 2 Then
    TxtConverted.Text = Math.Round(Robiah_russia * TxtAmount.Text, 2)
    LblConverted.Text = "RUB"
ElseIf CboCountry.SelectedIndex = 3 Then
    TxtConverted.Text = Math.Round(Pound_UK * TxtAmount.Text, 2)
    LblConverted.Text = "GBP"
ElseIf CboCountry.SelectedIndex = 4 Then
    TxtConverted.Text = Math.Round(Lairah_Turkey * TxtAmount.Text, 2)
    LblConverted.Text = "TYR"
ElseIf CboCountry.SelectedIndex = 5 Then
    TxtConverted.Text = Math.Round(Real_Iran * TxtAmount.Text, 2)
    LblConverted.Text = "IRR"
ElseIf CboCountry.SelectedIndex = 6 Then
    TxtConverted.Text = Math.Round(Lairah_syria * TxtAmount.Text, 2)
    LblConverted.Text = "SYP"
ElseIf CboCountry.SelectedIndex = 7 Then
    TxtConverted.Text = Math.Round(Real_saudi * TxtAmount.Text, 2)
    LblConverted.Text = "SAR"
Else
    ' خطأ رسالة اظهر
    MsgBox("القائمة من محدد بلد اختر", MsgBoxStyle.Critical, "خطأ")
    ' البلدان قائمة الى التركيز نقل
    CboCountry.Focus()
End If

```

دالة **Round** تعمل على تقريب الاعداد العشرية الى اعداد اكبر باقل منازل عشرية وتأخذ الشكل العام:

Math.Round(Value, Digit) As Double

حيث ان

Value: عدد عشري.

Digit: عدد المنازل بعد الفاصلة العشرية.

Double: اي ان العدد الناتج من هذه الدالة ممكن ان يكون عدد عشري بمراتب اقل.

مثال:

Math.Round(2.34656,2)

الناتج: 2.35

في الاجراء أعلاه تم اختبار الخاصية SelectedIndex للـ ComboBox في تحديد البلد المطلوب .أما اذا لم يتم تحدد بلد معين من القائمة فستظهر رسالة خطأ. كما في الشكل المجاور.



دالة **MsgBox** تستخدم لإظهار صندوق الرسالة المتمثل بالشكل أعلاه، ولها عدة أشكال سنتطرق اليها في المشاريع اللاحقة.

- يمكن استبدال جملة الشرط If بـ Select Case.

حيث توفر الفيچوال بيسك عبارة شرط من نوع Select Case بدل If..then ..elseif.. وتقوم هذه العبارة بإختبار شرط واحد ومقارنة نتيجة هذا الشرط مع مجموعة من القيم (ضمن Case) وعند تحقق أي قيمة من هذه التعبيرات فان الفيچوال بيسك سيقوم بتنفيذ Statement الخاصة بهذا Case. والشكل العام لهذه التعليمة كالتالي:

```

Select Case testexpression
    Case Expressionlist1
        Statement(s)
    Case Expressionlist2
        Statement(s)
    Case Expressionlist3
        Statement(s)...
    Case Else
        Statement(s)
End Select

```

• Testexpression: عبارة الاختبار.

• Expressionlist: وهي قائمة تضم قيمة واحدة او اكثر.

• Statement: عبارة عن لا شيء او سطر برمجي او اكثر.

ينفذ البرنامج قيمة Case والاسطر البرمجية التابعة لها واذا لم ينفذ اي واحدة منها فانه سينفذ Case Else.

باستخدام Select Case تصبح الشيفرة البرمجية لزر "تحويل" كالتالي:

```
Select Case CboCountry.SelectedIndex
```

```
Case 0
```

```
    TxtConverted.Text = Math.Round(Dollar_US * TxtAmount.Text, 2)
```

```
    LblConverted.Text = "SU Dollar"
```

```

Case 1
    TxtConverted.Text = Math.Round(Euz_German * TxtAmount.Text, 2)
    LblConverted.Text = "DEM"
Case 2
    TxtConverted.Text = Math.Round(Robiah_russia * TxtAmount.Text, 2)
    LblConverted.Text = "RUB"
Case 3
    TxtConverted.Text = Math.Round(Pound_UK * TxtAmount.Text, 2)
    LblConverted.Text = "GBP"
Case 4
    TxtConverted.Text = Math.Round(Lairah_Turkey * TxtAmount.Text, 2)
    LblConverted.Text = "TYR"
Case 5
    TxtConverted.Text = Math.Round(Real_Iran * TxtAmount.Text, 2)
    LblConverted.Text = "IRR"
Case 6
    TxtConverted.Text = Math.Round(Lairah_syria * TxtAmount.Text, 2)
    LblConverted.Text = "SYP"
Case 7
    TxtConverted.Text = Math.Round(Real_saudi * TxtAmount.Text, 2)
    LblConverted.Text = "SAR"
Case Else
    ' خطأ رسالة اظهر
    MsgBox("القائمة من محدد بلد اختر", MsgBoxStyle.Critical, "خطأ")
    ' البلدان قائمة الى التركيز نقل
    CboCountry.Focus()
End Select

```

ملاحظة في برمجة زر "تحويل" اعتمدنا على خاصية `SelectedIndex` العنصر المختار من القائمة `ComboBox` بدلاً من `Text` وتأخذ هذه الخاصية القيمة صفراً للعنصر الأول والقيمة واحد للعنصر الثاني والقيمة اثنان للعنصر الثالث وهكذا..

9- برمجة زر "مسح":

في هذا الاجراء نعمل على افراغ الأدوات من بياناتها نسحباً لعمليات ادخال اخرى.

```

Private Sub BtnReset_Click(sender As Object, e As EventArgs) Handles
BtnReset.Click
    TxtAmount.Text = ""
    TxtConverted.Text = ""
    CboCountry.Text = "...واحد اختر"
    LblConverted.Text = "-"
End Sub

```

9. برمجة زر "خروج":

```

Private Sub BtnExit_Click(sender As Object, e As EventArgs) Handles BtnExit.Click
    End
End Sub

```

المشروع 3 التعامل مع المتغيرات

ستتقن من خلال المشروع

التعامل مع المتغيرات Variables

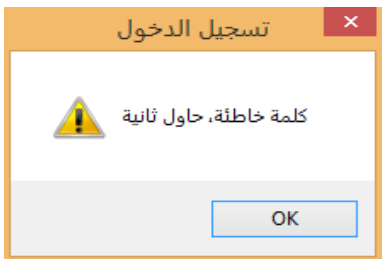
- المتغيرات الخاصة بإجراء.
- المتغيرات الساكنة **Static** الخاصة بإجراء.
- المتغيرات العامة لنموذج.
- المتغيرات العامة لمشروع.

التعامل مع عدة نماذج.

استخدام الوحدات النمطية Module.

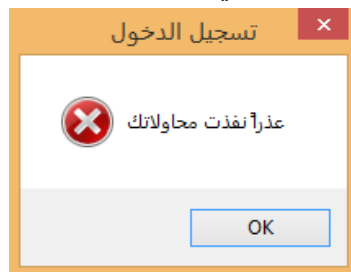
اهداف المشروع

برنامج يواجهه تطلب اسم مستخدم وكلمة سر.

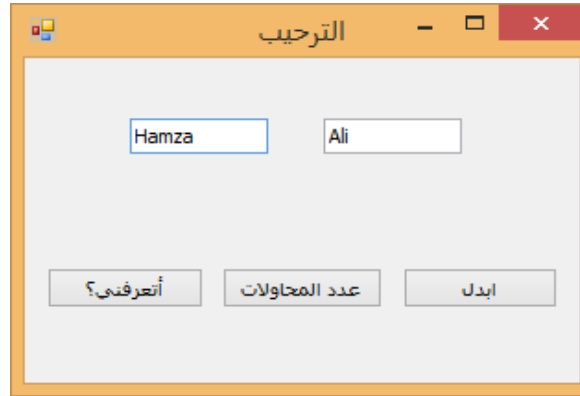


- إذا أدخل المستخدم كلمة السر خاطئة يظهر المربع الحواري التالي:

- يسمح البرنامج بإجراء 4 محاولات خاطئة لكلمة السر المدخلة وفي الخامسة تظهر الرسالة التالية:



- إذا ادخل المستخدم كلمة السر بصورة صحيحة ستظهر النافذة التالية:

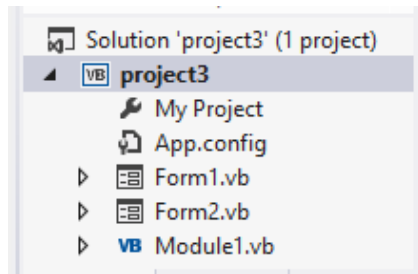


مراحل المشروع:

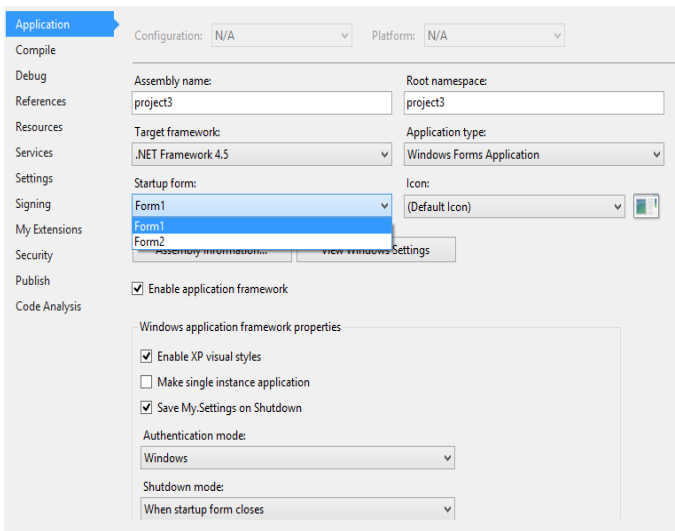
- 1- انشئ مشروعاً جديداً.
- 2- اضع نموذج اخر ووحدة نمطية Module للمشروع:
 - إضافة نموذج: PROJECT → Add Window Form
 - إضافة وحدة نمطية: PROJECT → Add Module

ملاحظات

- للتنقل بين مكونات المشروع استخدم مستكشف المشروع



- لجعل تنفيذ المشروع يبدأ من نموذج معين اختر Project Properties → PROJECT أو الضغط على مفتاحي Alt+F7 واختر Startup form.



- 3- اضبط خصائص النماذج كما يلي:

Form1

Form2

FormBorderStyle	FixedToolWindow	FormBorderStyle	Sizable
BackColor	ابيض	BackColor	ابيض
RightToLeft	true	RightToLeft	true
Text	تسجيل الدخول	Text	الترحيب

الخاصية **FormBorderStyle** تستخدم لتحديد شكل اطار النموذج القيمة الافتراضية له **Sizable** وتعني يمكن للمستخدم تغيير أبعاد النموذج.

والقيمة **FixedToolWindow** تعني حجم النموذج يبقى ثابت ولا يمكن تغييره من قبل المستخدم.

4- اصف أداتين textbox كالتالي:

Textbo1		Textbox2	
Name	txtn	Name	Txtp
PasswordChar	*	PasswordChar	

الخاصية **PasswordChar** لإظهار حرف أو رمز في صندوق النص بدل الأحرف المدخلة (غالباً يكون *).

5- اصف زرّين أمر على النموذج الاول واضبطهم كالتالي:

Button1		Button2	
Name	Btnok	Name	Btncn
BackColor	ابيض	BackColor	ابيض
Text	موافق	Text	الغاء
TextAlign	middleleft	TextAlign	middleleft
Font	Tahoma; 11.25pt; style=Bold	Font	Tahoma; 11.25pt; style=Bold
Image	اختر صورة	Image	اختر صورة
ImageAlign	middleRight	ImageAlign	middleRight

الخاصية **TextAlign** تستخدم لتحديد محاذاة النص داخل زر الامر، والقيمة **middleleft** توسيط باتجاه اليسار.

الخاصية **ImageAlign** لتحديد محاذاة الصورة داخل زر الأمر واخترنا القيمة **middleRight** لجعل الصورة بجانب الكلمة وتجنب وضع الصورة تحت الكلمة.

6- اصف label وضع في الخاصية **text** للاول "الاسم" وللثاني "كلمة المرور".

7- اصف PictureBox واضبط خاصية **Size Mode** بالقيمة **StretchImage**. وضع فيه صورة من اختيارك.

8- اصف صندوقين نص في النموذج الثاني واترك خصائصهما الافتراضية كما هي.

9- اصف ثلاثة أزرار أمر في النموذج الثاني وسمها "ابدل" ، "عدد المحاولات" و "أتعرفني؟".

10- التعليمة البرمجية لزر "ابدل"

للتبديل بين محتوى الصندوقين لأبد من استخدام وسيط (متغير) نضع فيه قيمة الصندوق الاول ثم نضع قيمة الصندوق الثاني في الصندوق الاول ، ثم نضع قيمة الوسيط في الصندوق الثاني.

فالمتغير **Variable** يمثل عنوان محجوز في ذاكرة الحاسوب، يمكن القراءة منه والكتابة عليه، ويتم التصريح عن المتغير في الفيچوال ببسك كما يلي:

Dim variable_name As type

- variable_name يمثل اسم المتغير.

- Type هو النوع البياني للمتغير.

وفترة حياة المتغير تعتمد على طريقة تعريفه إذا كان محلي (يعرف في اجراء معين) عندها تبدأ حياة المتغير مع بداية تنفيذ الاجراء وتنتهي بنهاية تنفيذ الاجراء. أهم الأنواع البيانية للمتغيرات.

Integer	عدد صحيح
Long	عدد صحيح طويل
Single	عدد عشري
Double	عددي عشري مضاعف الدقة
string	سلسلة نصية

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Dim s As String
    s = TextBox1
    TextBox1.Text = TextBox2.Text
    TextBox2.Text = s
End Sub
```

11- برمجة زر "عدد المحاولات"

```
MsgBox(x)
```

12- برمجة زر "أعرفني؟"

```
MsgBox(username)
```

في الوحدة النمطية Module تم تعريف متغير public ليتسنى استخدام المتغير في جميع نماذج المشروع وفي كل الاجراءات التابعة له. كما يلي:

```
Module Module1
    Public username As String
    Public x As Integer
End Module
```

13- الشيفرة البرمجية لزر "موافق"

```
Static i As Integer
i = i + 1
username = Txtn.Text
x = i
If Txtp.Text = "12345" Then
    Me.Hide()
    Form2.Show()
Else
    If i = 5 Then
        MsgBox("الدخول تسجيل", MsgBoxStyle.Critical, "الدخول تسجيل")
        End
    Else
        MsgBox("الدخول تسجيل", MsgBoxStyle.Exclamation, "الدخول تسجيل")
        Txtp.Focus()
    End
End
```

```
SendKeys.Send("{Home}+{End}")
End If
End If
```

14- تعليمة زر "الغاء"

```
End
```

ملاحظات

- 1- في التعليمة **Static** فان المتغير في الإجراء يُحافظ على قيمته عند استدعاء للإجراء.
- 2- يمكن أن نرمر للنموذج الحالي Form او me كلاهما صحيحين وتُستخدم معها الطرق التالية:

Me.hide لاختفاء النموذج

Form2.show لاطهار النموذج

- 3- طريقة focus تُستخدم لوضع التحديد على أداة تحكم. مثل

```
Txtp.Focus()
```

- 4- الطريقة Sendkeys ارسال محارف الى النافذة النشطة. مثل:

```
SendKeys.Send("{Home}+{End}")
```

هذا السطر البرمجي يؤدي الى تظليل النص المكتوب في الصندوق.

المشروع 4

عبارات الشرط (سفرة سياحية)

ستتقن من خلال المشروع

التعامل مع أدوات التحكم التالية:

– زر الخيار RadioButton

– زر التحقق CheckBox

– صندوق المجموعة GroupBox

الجمل الشرطية if و Select

اهداف المشروع:

عند تنفيذ المشروع تظهر واجهة تضم الخدمات التي تقدمها شركة سياسية معينة لشخص ما.

مراحل المشروع

- 1- انشئ مشروعاً جديداً.
- 2- اضبط خصائص النموذج كالتالي:

Form1	
BackColor	ابيض
Name	Form1
Text	سفرات سياحية
StartPosition	CenterScreen

- 3- اضف صندوق قائمة واضبطه بالشكل:

ComboBox	
Name	Cbostr
Items	*** ****
Font	Tahoma;12pt;style=Bold
Text	اختر نوع الخدمة
RightToLeft	True

4- اضع خمسة أدوات GroupBox واضبط خاصية RihtToLeft بالقيمة True والخاصية Text كالتالي: "النقل" ، "الطعام" ، "الاماكن" ، "طريقة الدفع" و "الفترة".

توضيح نستخدم صندوق المجموعة لإحتواء مجموعة من الأدوات المترابطة فيما بينها، حيث يجب إنشاء الصندوق ومن ثم اضافة الأدوات اليه . ولنتأكد من الأدوات تابعة للصندوق عند سحب الصندوق تُسحب الأدوات معه.

عند وضع مجموعة من RadioButton داخل صندوق المجموعة يمكن لزر واحد فقط من المجموعة ان يتحدد. عند النقر عليه بالفأرة

5- اضع زرین RadioButton داخل صندوق المجموعة "النقل" واضبطه بالشكل:

RadioButton1		RadioButton2	
Name	radro	Name	radai
Checked	True	Checked	false
Text	بري	Text	جوي

الخاصية **Checked** لعنصر RadioButton تاخذ القيمة true اذا كان الزر محدد، وتاخذ القيمة False اذا كان الزر غير محدد

6- اضع ثلاثة ازرار CheckBox داخل صندوق المجموعة "الطعام" واضبط الخصائص كالتالي:

CheckBox1		CheckBox2		CheckBox3	
Name	Chbreak	Name	Chlunch	Name	Chdinner
Checked	True	Checked	False	Checked	False
Text	فطور	Text	غداء	Text	عشاء

7- اضع ثلاثة ازرار CheckBox داخل صندوق المجموعة "الاماكن" واضبط الخصائص:

CheckBox1		CheckBox2		CheckBox3	
Name	Chsul	Name	ChDohuk	Name	ChErbil
Checked	True	Checked	False	Checked	False
Text	سليمانية	Text	دهوك	Text	اربيل

الخاصية **Checked** للعنصر CheckBox تاخذ القيمة true اذا كان الزر محدد واذا كان الزر غير محدد تأخذ القيمة False.

8- اضع زرین RadioButton داخل صندوق "طريقة الدفع" واضبط الخصائص:

RadioButton1		RadioButton2	
Name	radcash	Name	radInsall

Checked	True	Checked	false
Text	نقداً	Text	بالتقسيط

9- اضع زرین RadioButton داخل صندوق المجموعة "الإقامة" واضبط الخصائص:

RadioButton1		RadioButton2	
Name	radten	Name	radfive
Checked	True	Checked	false
Text	10 ايام	Text	5 ايام

10- اضع صندوق نص باسم (txtresult) لوضع الناتج فيه، و زر أمر باسم "المبلغ"

11- برمجة زر "المبلغ"

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim n As Single
    Select Case Cbostr.SelectedIndex
        Case 0
            n = 100
        Case 1
            n = 200
        Case 2
            n = 300
        Case Else
            MsgBox("الخدمة نوع اختر", MsgBoxStyle.Critical, "خطأ")
            Exit Sub
    End Select

    If Radro.Checked = True Then n = n + 100
    If Radai.Checked = True Then n = n + 250
    If Chbreak.Checked = True Then n = n + 75
    If Chlunch.Checked = True Then n = n + 120
    If Chdinner.Checked = True Then n = n + 200
    If Chsul.Checked = True Then n = n + 300
    If ChDohuk.Checked = True Then n = n + 400
    If ChErbil.Checked = True Then n = n + 450
    If Radcash.Checked = True Then n = n * 0.3
    If RadInsall.Checked = True Then n = n * 1.1
    If Radten.Checked = True Then n = n + 230
    If Radfive.Checked = True Then n = n * 0.2

    TextBox1.Text = n
    Label2.Text = "DIQ"
End Sub
```

المشروع 5 حلقات التكرار

ستتقن من خلال المشروع

جمل التكرار

For –

Do.. while –

مسائل رياضية

اهداف المشروع

في واجهة المشروع التالية يمكن عمل ما يأتي:

- حساب مضروب عدد معين.
- اختبار العدد أولي أم لا.
- حساب المضاعف المشترك البسيط.
- حساب القاسم المشترك الأعظم.



مراحل المشروع

- 1- انشئ مشروعاً جديداً.
- 2- اصف أدوات التحكم كما في النموذج اعلاه.
- 3- برمجة زر "مضروب العدد".

الحلقة التكرارية For

شكلها العام:

For counter = start **to** end [**step** increment]

Statement(s)

Next counter

• Counter: هو عداد الحلقة التكرارية

- Start: القيمة الاولى للعداد.
- End: القيمة النهائية للعداد.
- Step: مقدار التغيير في قيمة العداد وجميعها قيم رقمية.

ملاحظات:

- 1- يمكن ان يكون step موجب او سالب، اذ كان موجب فان القيمة الابتدائية start تكون أصغر من او تساوي القيمة النهائية End والا فان الحلقة التكرارية لن تنفذ. أما اذا كان سالب فان القيمة الابتدائية start أكبر او تساوي القيمة النهائية End لثنفذ تعليمات التكرار بصورة صحيحة. إذا لم تُحدد مقدار الزيادة او النقصان فان مقدار الزيادة او النقصان يعتبرها الفيچوال ببسك واحد. واذا كان مقدار الزيادة او النقصان اكبر من واحد يجب كتابته بعد step.
- 2- Exit For يمكن استخدام هذه التعليمة البرمجية مع الحلقة التكرارية for لكسر تنفيذها (اي ينتقل التنفيذ الى ما بعد next) وبهذا ينتهي التكرار.

آلية عمل الحلقة التكرارية:

- اسناد القيمة الأولية (start) الى العداد (Counter).
- مقارنة قيمة العداد مع القيمة النهائية (End) اذا كانت اكبر يقفز العداد الى ما بعد next واذا كانت أصغر يستمر بتنفيذ الخطوات اللاحقة ويفحص مقدار الزيادة او النقصان بعد (step).
- ينفذ التعليمات (statement).
- بعدها يزيد مقدار العداد بـ 1 اذا لم يكن step محدد ، والا سيزيد او ينقص المقدار بقدر step
- وهكذا يعاود تنفيذ الخطوات الى أن يصل الى القيمة النهائية.

مضروب العدد: هو حاصل ضرب الاعداد من 1 الى العدد نفسه n اي ($1 * 2 * 3 * \dots * n$).

```
Private Sub Button1_Click_1(sender As Object, e As EventArgs) Handles Button1.Click
    Dim i As Integer, n As Integer
    Dim fact As Double = 1
    n = Val(TextBox1.Text)
    For i = 1 To n
        fact *= i
    Next i
    TextBox3.Text = fact
End Sub
```

4- برمجة "اولي ام لا"

- العدد الأولي: هو العدد الذي لا يقبل القسمة على اي عدد (ما عدا على نفسه والواحد فقط).
نستخدم عداد تكراري يبدأ من 2 وقيمته النهائية اقل من n:
- اذا قبل القسمة على اي عدد ضمن هذه الاعداد يصبح n غير اولي.
 - واذا لم يقبل القسمة على اي عدد منها يكون العدد n اولي.

الحلقة التكرارية Do While

شكلها العام:

```
Do While (condition)
    Statement(s)
Loop
```

عندما يُنفذ الفيچوال بيسك هذا الكود البرمجي يبدأ أولاً باختبار الشرط `condition` اذا كان `true` يستمر بتنفيذ الخطوات التالية الى ان يصل الى `Loop` ويستمر بالتنفيذ عدد من المرات اعتماداً على الشرط، واذا كان الشرط `False` سيقفز الى ما بعد `Loop` أي انه سيخرج من الحلقة التكرارية نهائياً.

ملاحظات

1. يمكن استخدام (`Exit do`) لكسر الحلقة ونقل التنفيذ الى ما بعد `Loop`، وبهذا سينتهي التكرار.
2. لا يوجد في `Do While` زيادة تلقائية للعداد لذا يُعرف متغير ويُسند له قيمة ويزداد مقدارها داخل الحلقة التكرارية.

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Dim n As Integer, i As Integer
    n = Val(TextBox2.Text)

    RadioButton1.Checked = True
    i = 2
    Do While (i < n)
        If n Mod i = 0 Then
            RadioButton1.Checked = False
            Exit Do
        End If
        i += 1
    Loop
End Sub
```

5- برمجة زر "المضاعف المشترك"
المضاعف المشترك البسيط لعددين (`SCD`): هو اصغر عدد يقبل القسمة على كلا العددين ويكون محصور بين اكبر عدد وحاصل ضرب العددين. مثلاً حاصل ضرب 3 و 4 هو 12
الفكرة البرمجية:

- نفرض ان العددين `n,m`.
- نُوجد أكبر عدد بينهما وليكن `max`.
- نستخدم حلقة تكرارية من `max` الى حاصل ضرب العددين `n*m`.
- في كل دورة نختبر `i` يقبل القسمة على `n` و `m`.
- اذا قبل القسمة `i` على `n` و `m` فهو المضاعف المشترك البسيط عندها نكسر الحلقة بالـ (`Exit for`).

كما يلي:

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    Dim n As Integer, m As Integer, i As Integer
    Dim max As Integer
    n = Val(TextBox5.Text)
    m = Val(TextBox6.Text)

    If n > m Then max = n Else max = m

    For i = max To n * m
        If (i Mod n = 0) And (i Mod m = 0) Then
            Exit For
        End If
    Next
    TextBox9.Text = i
End Sub
```

5- برمجة زر "القاسم المشترك"

القاسم المشترك الأعظم (GCD): هو أكبر عدد يقبل القسمة على العددين بلا باقي ويكون محصور بين اصغر عدد بينهما وبين الـ 1 مثلاً القاسم المشترك بين 4 و 2 هو 2
الفكرة البرمجية:

- نفرض العددين n, m .
 - نوجد اصغر عدد بينهما وليكن \min .
 - نستخدم عداد يبدأ من \min الى 1 في كل مرة نختبر ناتج قسمة العددين n, m على i اذا قيل القسمة بدون باقي فهو القاسم المشترك بينهما.
- كما يلي:

```
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
    Dim n As Integer, m As Integer
    Dim i As Integer, min As Integer
    n = Val(TextBox7.Text)
    m = Val(TextBox8.Text)

    If n < m Then min = n Else min = m
    For i = min To 1 Step -1
        If (n Mod i = 0) And (m Mod i = 0) Then
            Exit For
        End If
    Next
    TextBox4.Text = i
End Sub
```

المشروع 6 الدوال الحوارية

ستتقن من خلال المشروع

دوال حوار المستخدمين

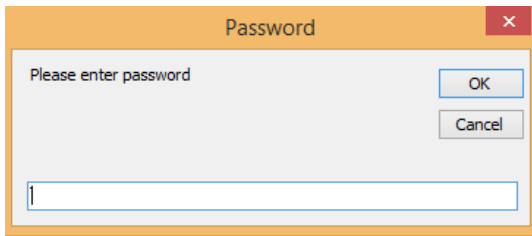
– دالة الإدخال InputBox

– دالة الرسائل MsgBox

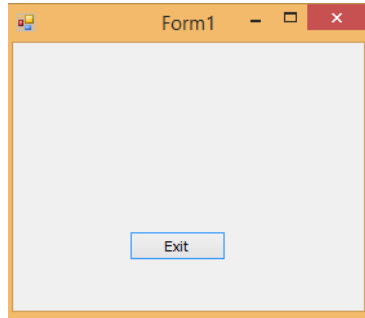
– حلقة While

اهداف المشروع

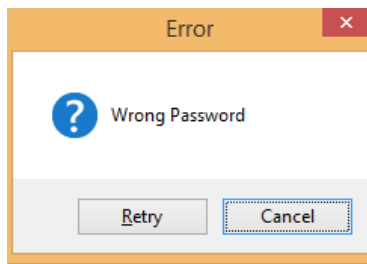
• عند تنفيذ المشروع يظهر صندوق الإدخال يطلب كلمة السر:



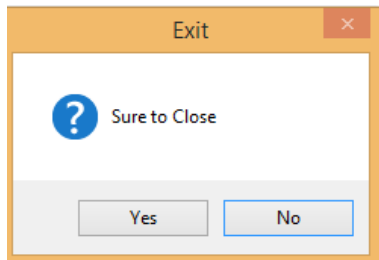
• إذا ادخل المستخدم كلمة السر بصورة صحيحة يظهر النموذج التالي



• إذا أخطأ المستخدم بإدخال كلمة السر تظهر الرسالة التالية:



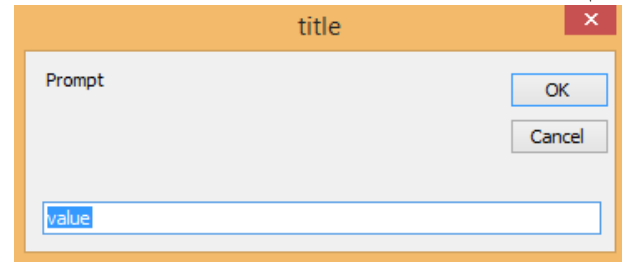
• إذا اراد المستخدم الخروج من البرنامج تظهر رسالة لتأكيد الخروج كالتالي:



دالة **InputDialog** تستخدم لإظهار صندوق إدخال يطلب إدخال بيانات من المستخدم، وتنتظر أن يقوم المستخدم بكتابة نص في الصندوق او ينفق على أحد الزرين "OK" او "Cancel" وتعيد سلسلة نصية تمثل محتوى الصندوق.

شكلها العام:

`InputDialog(prompt, [title], [default], [xpos], [ypos])`



حيث ان:

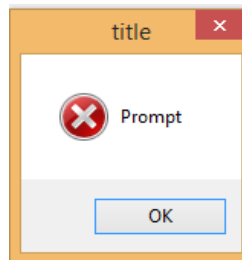
- Prompt: هي الرسالة التي تظهر في الصندوق.
- Title: عنوان الصندوق.
- Default: القيمة الافتراضية التي تظهر في الصندوق.
- Xpos, ypos: احداثيات الصندوق.

عند النقر على زر "OK" او مفتاح "Enter" فان القيمة المرجعة من الدالة هي النص المكتوب داخل الصندوق، اما اذا نقر المستخدم على زر "Cancel" فان الدالة لا ترجع اي قيمة مهما كان النص المكتوب داخل الصندوق.

دالة **MsgBox** تستخدم لإظهار صندوق الرسائل للمستخدم ويمكن ان يحتوي الصندوق على ازرار او ايقونات ذات اشكال مختلفة بالاضافة الى نصوص معينه لبيان حالات معينه والقيمة التي ترجعها الدالة تكون من نوع **Integer** تمثل الزر الذي نقره المستخدم.

شكلها العام:

`MsgBox(Prompt, [buttons], [title])`

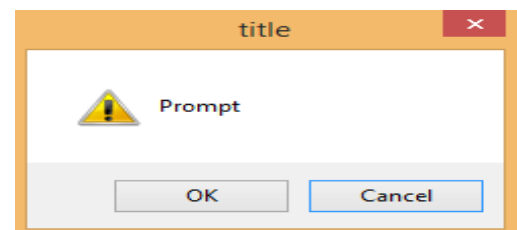


حيث ان:

- Prompt: الرسالة التي تظهر داخل الصندوق.
- Title: عنوان الصندوق.
- Buttons: تحدد الايقونة وازرار الصندوق.

فيمكن استخدام علامة التعجب بالاضافة الى ازرار (Ok, Cancel) على أن يكون الزر الثاني هو الافتراضي، باستخدام التعبير التالي:

```
MsgBox("Prompt", MsgBoxStyle.Exclamation +
MsgBoxStyle.OkCancel + MsgBoxStyle.DefaultButton2,
"title")
```



هنالك عدة أشكال للايقونة التي تظهر داخل الصندوق، كل ايقونة لها تعبير خاص بها كما في الجدول ادناه.

MsgBoxStyle.Question	ايقونة الاستفهام
MsgBoxStyle.Exclamation	ايقونة التعجب
MsgBoxStyle.Information	ايقونة المعلومات
MsgBoxStyle.Critical	ايقونة الخطأ

وكذلك بالنسبة للازرار ايضاً لكل منها تعبير خاص به.

MsgBoxStyle.OkCancel	Ok Cancel
MsgBoxStyle.OkOnly	Ok
MsgBoxStyle.YesNo	Yes No
MsgBoxStyle.YesNoCancel	Yes No Cancel
MsgBoxStyle.AbortRetryIgnore	Abort Retry Ignore
MsgBoxStyle.RetryCancel	Retry Cancel

اما التعابير الخاصة بالازرار الافتراضية كالتالي:

MsgBoxStyle.DefaultButton1	الزر الاول هو الافتراضي
MsgBoxStyle.DefaultButton2	الزر الثاني هو الافتراضي
MsgBoxStyle.DefaultButton3	الزر الثالث هو الافتراضي

2- انقر على النموذج مرتين واكتب التعليمة التالي في حدث **Form_Load** ليتم تنفيذه عند تحميل النموذج من الذاكرة.

عملية الادخال تمر بالاحتمالات التالية:

- اذا أدخل المستخدم كلمة السر بشكل صحيح يتم كسر التكرار.
- اذا لم يدخل المستخدم بيانات في الصندوق وكذلك اذا تجاوز المستخدم ثلاث محاولات يُغلق البرنامج.
- اذا ادخل كلمة السر بصورة خاطئة يظهر صندوق يتضمن اعادة المحاولة او اغلاق البرنامج.

```
Dim x As Integer, i As Integer = 0, r As Integer
While True
    i += 1
    x = Val(InputBox("Please enter password", "Password"))
    If x = "12345" Then Exit While
    If x = Nothing Then End
    If i = 3 Then End
    r = MsgBox("Wrong Password", MsgBoxStyle.Question +
MsgBoxStyle.RetryCancel + MsgBoxStyle.DefaultButton2, "Error")
    If r = vbCancel Then End
End While
```

3- برمجة زر "Exit" للخروج من البرنامج:

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

```
    Dim x As Integer
    x = MsgBox("Sure to Close", MsgBoxStyle.Question + MsgBoxStyle.YesNo +
MsgBoxStyle.DefaultButton2, "Exit")
    If x = vbYes Then End
End Sub
```

المشروع 7 تعدد الواجهات

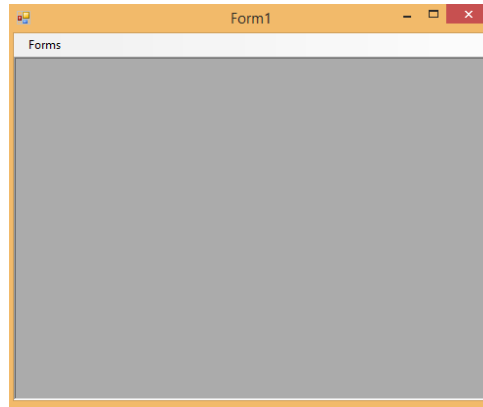
ستتقن من خلال المشروع

التعامل مع الأدوات التالية

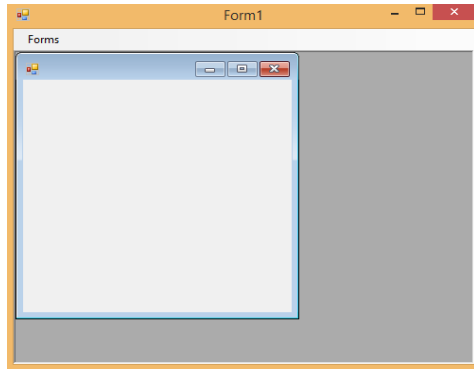
- عمل شريط القوائم باستخدام Menustrip
- Multiple InterFace

اهداف المشروع

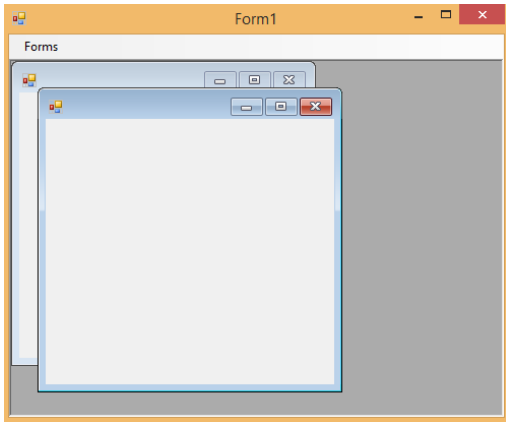
عند التنفيذ تظهر الواجهة التالية



- عندما يختار المستخدم Form2 تفتح داخل النموذج الرئيسي



- وعندما يختار المستخدم Form3 من قائمة Forms يفتح النموذج الثالث داخل النموذج الرئيسي.

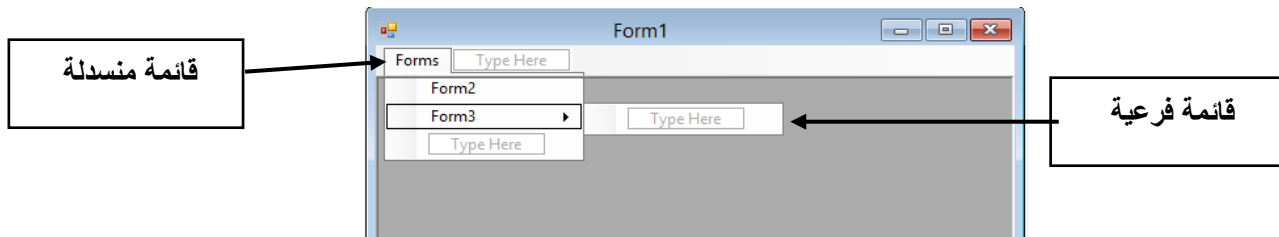


1- انشئ مشروعاً جديداً.

2- اضع ثلاثة نماذج واجعل خاصية `ismdicontainer=true` للنموذج الاول سيتحول لون النموذج الى الرمادي.

3- اضع `menustrip` على النموذج الاول

الاداة `menustrip` بعد النقر بالفأرة مرتين عليها في صندوق الادوات تضاف على سطح النموذج ويكون موقعها اعلى النموذج بمجرد انقر بالماوس على `Type here` داخل هذا الشريط سيظهر المؤشر الوامض قم بكتابة ما يلي:



كما هو معروف فان شريط القوائم يتضمن قوائم منسدلة تضم مجموعة من الأوامر الخاصة بها، ويمكن ان يحتوي الامر الواحد على قائمة فرعية اخرى (تضم امر او عدة اوامر). وهكذا بالنسبة للبقية وتضم هذه الاداة الكثير من الخصائص سنتطرق اليها لاحقاً.

4- برمجة أمر "Form2"

```
Private Sub Form2ToolStripMenuItem_Click_1(sender As Object, e As EventArgs)
Handles Form1ToolStripMenuItem.Click
    Dim f2 As New Form
    f2.MdiParent = Me
    f2.Show()
End Sub
```

ملاحظة تتم كتابة الشيفرة البرمجية الخاص بكل أمر في شريط القوائم من خلال النقر المزدوج عليه بالفأرة.

5- برمجة أمر "Form3"

```
Private Sub Form3ToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
Form2ToolStripMenuItem.Click
    Dim f3 As New Form
    f3.MdiParent = Me
    f3.Show()
End Sub
```

المشروع 8 توليد أرقام عشوائية

ستتقن من خلال المشروع

التعامل مع الأدوات التالية

– دالة Ran

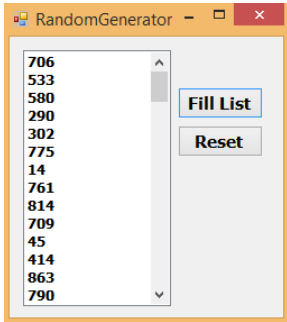
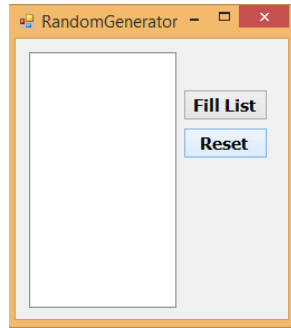
– صندوق القائمة ListBox

– تكرر Do..loop While

اهداف المشروع

عند التنفيذ تظهر الواجهة التالية:

- عند النقر على زر Fill List يتم ملئ القائمة بمئة رقم عشوائي.
- عند النقر على زر Reset يتم افرغ القائمة من عناصرها.



مراحل المشروع

1- انشئ مشروعاً جديداً.

2- اضع الأدوات الظاهرة على النموذج أعلاه.

الخاصية **Items** في الأداة **ListBox** من خلالها يمكن التحكم بعناصر القائمة، ويمكن التعامل معها برمجياً من حيث الإضافة والحذف وكذلك التعديل على عناصر القائمة.

الحلقة Do ..loop while

شكلها العام:

Do

Statement(s)

Loop While Condition

الملاحظ في هذا الشكل من التكرار يبدأ البرنامج بتنفيذ الخطوة الاولى ومن ثم يتحقق من الشرط، فاذا كان الشرط **False** فيكتفي البرنامج بتنفيذ خطوة واحدة فقط، أما اذا كان الشرط **True** فيستمر بتنفيذ الخطوات مادام الشرط متحقق.

3- برمجة زر "Fill List"

دالة Ran تستخدم لتوليد ارقام عشوائية محصورة بين 1 والعدد الذي تضرب به دالة توليد الارقام العشوائية.

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim RandNo, i As Integer
    i = 1
    Do
        ' عشوائي رقم توليد
        RandNo = Rnd() * 1000
        ' القائمة الى الرقم اضافة
        ListBox1.Items.Add(RandNo)
        i += 1
    Loop While i <= 100
End Sub
```

4- برمجة زر "Reset"

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    ' العناصر من القائمة افرغ
    ListBox1.Items.Clear()
End Sub
```

المشروع 9

نوافذ الحوار

ستتقن من خلال المشروع

التعامل مع نوافذ الحوار Dialog

– نافذة الالون ColorDialog

– نافذة الخطوط FontDialog

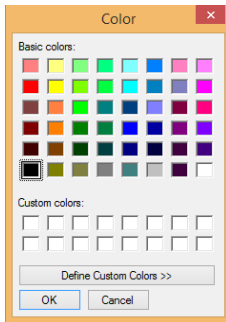
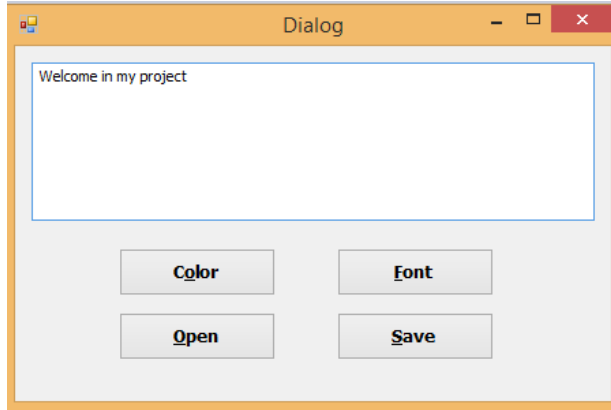
– نافذة حفظ ملف SaveFileDialog

– نافذة فتح ملف OpenFileDialog

– استخدام دالة My.Computer

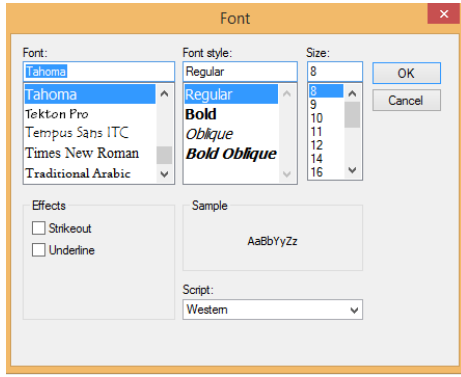
اهداف المشروع

عند التنفيذ تظهر الواجهة التالية التي تتطلب إدخال بيانات الى صندوق النص لكي تنشط الأزرار للاستخدام.

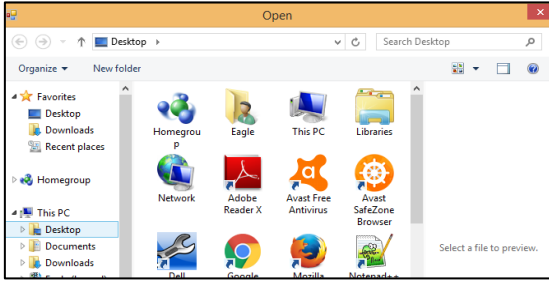


- عند النقر على زر "Color" تظهر نافذة الالون، لإختيار لون معين.

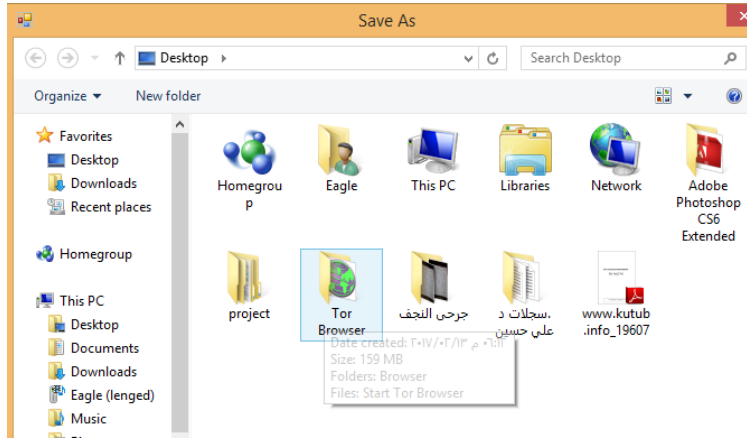
- عند النقر على زر "Font" تظهر نافذة خاصة بالخطوط.



- عند النقر على "Open" تظهر نافذة لفتح الملفات.



- وبالنقر على "Save" تظهر نافذة لحفظ الملف ومن خلالها يتم تحديد صيغة حفظ الملف.



مراحل المشروع

- 1- انشئ مشروعاً جديداً.
- 2- اضع الأدوات الظاهرة على النموذج مع التأكيد على جعل خاصية `MultiLine=True` للـ `TextBox`. الخاصية `MultiLine` لصندوق النص التي تسمح لك بكتابة أكثر من سطر في الصندوق اذا جعلت قيمتها `True`، اما اذا تركتها على القيمة الافتراضية `False` فلا يُمكنك كتابة اكثر من سطر في الصندوق.
- 3- اضع أدوات النوافذ الى مشروعك (`ColorDialog`, `FontDialog`, `OpenFileDialog`, `SaveFileDialog`) عن طريق سحب كل اداة من صندوق الادوات الى النموذج ستظهر هذه الادوات اسفل النموذج.
- 4- في حدث `Form1_Load` اكتب التعليمات التالي:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Button1.Enabled = False
    Button2.Enabled = False
    Button3.Enabled = False
```

```
Button4.Enabled = False
End Sub
```

فعند تحميل البرنامج من ذاكرة الحاسوب ستكون الأزرار غير مفعلة الى ان يقوم المستخدم بادخال نص الى الصندوق عندها تنشط الأزرار.

5- في حدث TextBox1_TextChanged اكتب الاسطر البرمجية التالية:

```
Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles
TextBox1.TextChanged
    Button1.Enabled = True
    Button2.Enabled = True
    Button3.Enabled = True
    Button4.Enabled = True
End Sub
```

6- برمجة زر "Color"

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

    If ColorDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
        TextBox1.ForeColor = ColorDialog1.Color
    End If
End Sub
```

7- برمجة زر "Font"

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    If FontDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
        TextBox1.Font = FontDialog1.Font
    End If
End Sub
```

8- برمجة زر "Open"

الدالة **My.Computer** تستخدم في هذا الجزء من البرنامج لقراءة النصوص من الملف بعد فتحه باستخدام دالة **.OpenFileDialog**.

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    Dim myfile As String
    If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
        OpenFileDialog1.Filter = "textfile|*.txt|wordfile|*.docx"
        myfile = My.Computer.FileSystem.ReadAllText(OpenFileDialog1.FileName)
        TextBox1.Text = myfile
    End If
End Sub
```

9- برمجة زر "Save"

تم استخدام دالة **My.Computer** لكتابة البيانات على الملف المحفوظ باستخدام **.SaveFileDialog**.

```
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
```

```
SaveFileDialog1.Filter = "pdf|*.pdf|word|*.docx"  
If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then  
    My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName,  
    TextBox1.Text, True)  
End If  
End Sub
```

تم وضع عبارة الشرط IF لضمان عمل البرنامج بصورة صحيحة وتجنب الاخطاء اذا لم يُحدد اي ملف من النافذة.

المشروع 10

التعامل مع الدوال

ستتقن من خلال المشروع

اجراء العمليات الرياضية التالية:

- مضروب العدد (المفكوك)
- مجموع العدد
- الأعداد الأولية

اهداف المشروع

عند تنفيذ المشروع تظهر الواجهة التالية:



- في القسم الايمن من الواجهة، عند ادخال عدد صحيح والنقر على زر "مضروب العدد" يظهر الناتج في الصندوق أسفل الزر، وعند ادخال عدد آخر في الصندوق التالي ثم النقر على زر "مجموع العدد" ايضاً سيظهر الناتج في الصندوق اسفل الزر.
- في القسم الأيسر من الواجهة، عند ادخال عدد والنقر على زر "الأعداد الأولية" تظهر جميع الأعداد الأولية التي تقع بين 1 و العدد المُدخل في القائمة أسفل الزر، ويظهر عددها في الصندوق أسفل القائمة.

ملاحظات

- العدد الأولي: هو العدد الذي يقبل القسمة على نفسه والواحد فقط.
- مضروب(مفكوك) العدد: هو حاصل ضرب الأعداد من 1 إلى نفس العدد.
- مجموع العدد: هو حاصل جمع الأعداد من 1 إلى نفس العدد.

مراحل المشروع

- 1- انشئ مشروعاً جديداً.
 - 2- اضع الأدوات الظاهرة على النموذج.
 - 3- اضع وحدة نمطية Module للمشروع، ثم ضع الدوال التالية فيها.
- دالة Fact حيث يُمرر إليها عدد صحيح وتُعيد مضروب (مفكوك) العدد على أن تكون من نوع Double (لان المفكوك يمكن ان يصل الى أرقام ضخمة) مع التأكيد على اسناد القيمة الابتدائية لـ $F=1$ للحصول على نتائج دقيقة.

```
Function fact(n As Integer) As Double
    Dim i As Integer, f As Double = 1
    For i = 1 To n
```

```

    f *= i
Next i
Return f
End Function

```

يمكن ان نستغني عن المتغير f في دالة fact ونستخدم اسم الدالة فقط فيكون شكل الدالة كالتالي:

```

Function fact(n As Integer) As Double
    Dim i As Integer
    Fact=1
    For i = 1 To n
        fact *= i
    Next i
End Function

```

كما يمكن استخدام طريقة الاستدعاء الذاتي بدل الحلقة For لاجاد المفكوك فيكون شكل الدالة كالتالي:

```

Function fact(n As Integer) As Double
    If (n = 0) Then
        fact = 1
    Else
        fact = n * fact(n - 1)
    End If
End Function

```

في الشيفرة البرمجية أعلاه تقوم دالة fact باستدعاء نفسها بعدد قيمة n وعندما تصل قيمة n الى الصفر تتوقف الدالة، ومن هذا يتبين أن n! هو عبارة عن ضرب متكرر لقيم متناقصة $n! = n * (n-1)!$.

- دالة sum ايضاً تُمرر لها عدد صحيح وتُعيد مجموع العدد من نوع Long، وربما يكون هذا العدد كبير لذا يفضل اختيار نوع بياني اكبر حجم. على ان تكون القيمة الابتدائية للدالة صفراً للحصول على نتائج دقيقة.

```

Function sum(n As Integer) As Long
    Dim i As Integer
    sum = 0
    For i = 1 To n
        sum += i
    Next
End Function

```

كذلك يمكن الاستعانة بمبدأ الاستدعاء الذاتي بدلاً من استخدام العداد التكراري كالتالي:

```

Function sum(n As Integer) As Long
    If n = 0 Then
        sum = 0
    Else
        sum = n + sum(n - 1)
    End If
End Function

```

- دالة primenumber حيث تُمرر لها عدد صحيح، فاذا كان العدد أولي تُعيد True وإلا تُعيد False اختبار العدد الأولي n نقسمه على كل الأعداد المحصورة بين 2 و n-1، اذا لم يقبل القسمة على أي عدد منهم فان العدد أولي، أما اذا قبل القسمة على واحد منهم فالعدد غير اولي.

داخل دالة primenumber نفرض في البداية ان العدد اولي ثم نستخدم عداد تكراري لفحص قسمته على الاعداد بين 2 و n-1 فاذا انتهى الدوران ولم يتحقق الشرط داخل If فان العدد اولي، اما اذا وجدنا عدد واحد يقبل القسمة (يتحقق الشرط داخل If) فان العدد غير اولي ويكسر الدوران.

```
Function primenumber(n As Integer) As Boolean
    Dim i As Integer
    primenumber = True
    For i = 2 To n - 1
        If n Mod i = 0 Then
            primenumber = False
            Exit For
        End If
    Next
End Function
```

4- استدعاء دالة fact في زر "مضروب العدد"

```
Private Sub BtnFactor_Click(sender As Object, e As EventArgs) Handles BtnFactor.Click
    TextBox4.Text = fact(Val(TextBox3.Text))
End Sub
```

5- استدعاء دالة sum في زر "مجموع العدد"

```
Private Sub BtnSum_Click(sender As Object, e As EventArgs) Handles BtnSum.Click
    TextBox6.Text = sum(Val(TextBox5.Text))
End Sub
```

6- استدعاء دالة primenumber في زر "الأعداد الأولية"

نريد طباعة الاعداد الاولية بين 1 والعدد المُدخل في صندوق النص، ولهذا يجب ان ندور على الاعداد من 1 الى val(textbox6.text) باستخدام for، وفي كل مرة نجد عدد اولي(باستدعاء دالة primenumber) نضيفه الى القائمة ListBox1 ونُزيد قيمة total بمقدار واحد.

```
Private Sub BtnPrime_Click(sender As Object, e As EventArgs) Handles BtnPrime.Click
    ListBox1.Items.Clear()
    Dim i, total As Integer
    For i = 1 To Val(TextBox1.Text)
        If primenumber(i) = True Then
            ListBox1.Items.Add(i)
            total += 1
        End If
    Next
    TextBox2.Text = total
End Sub
```


المشروع 11

التعامل مع صندوق القائمة

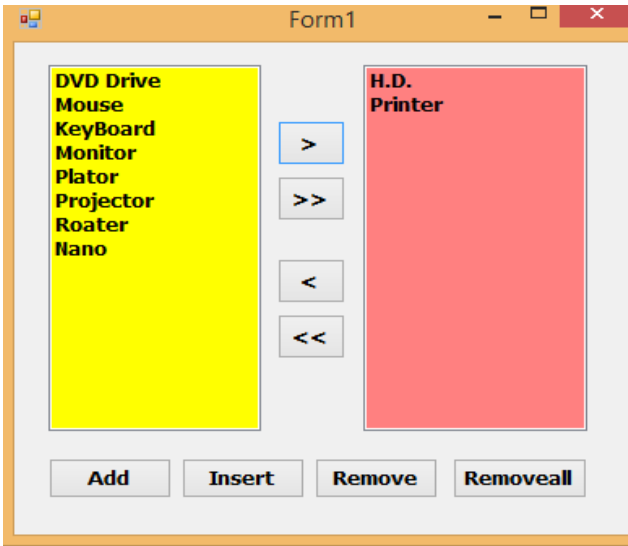
ستتقن من خلال المشروع

التعامل مع ListBox

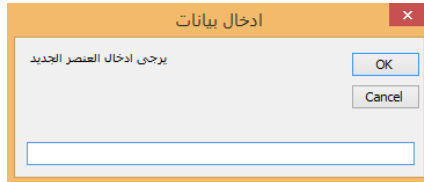
التعامل مع MessageBox

اهداف المشروع

عند تنفيذ المشروع تظهر الواجهة المجاورة التي تسمح للمستخدم بإجراء العمليات ادناه.

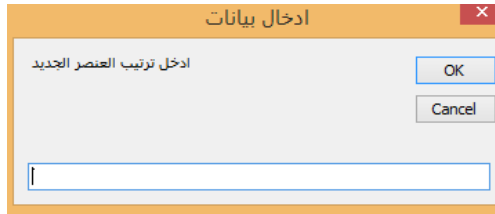
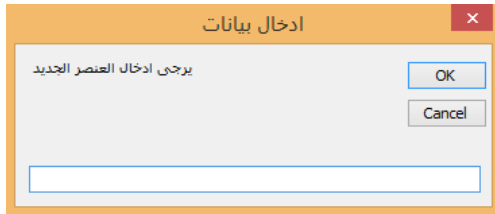


- النقر على زر ">" يؤدي الى حذف العنصر المحدد من القائمة في اليسار ونقله الى القائمة في اليمين.
- النقر على ">>" يؤدي الى نقل جميع العناصر من القائمة اليسرى ووضعها في القائمة اليمنى.
- النقر على "<" يؤدي الى نقل العنصر المحدد من القائمة الثانية الى القائمة الاولى.
- النقر على "<<" يؤدي الى نقل جميع العناصر من القائمة الثانية الى القائمة الاولى.
- النقر على زر "Add" يظهر صندوق حوارى يسمح بإدخال عناصر جديدة الى القائمة الاولى.



يرجى ادخال العنصر الجديد

- النقر على زر "Insert" يظهر صندوق حوارى يطلب اضافة عنصر جديدة وصندوق حوارى آخر يطلب تحديد موقع العنصر الجديد داخل القائمة الاولى.

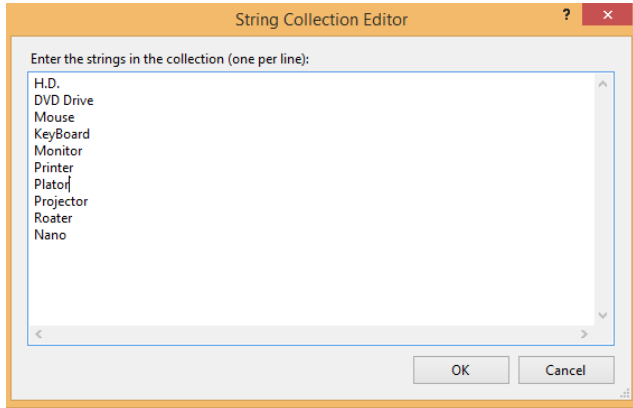


- عند تحديد عنصر من القائمة الاولى والنقر على "Remove" يحذف العنصر منها.
- النقر على "Removeall" يؤدي الى حذف جميع عناصر القائمة الاولى.

مراحل المشروع

1- انشئ مشروعاً جديداً.

2- اصف للنموذج صندوقين قائمة ListBox.



3- املأ القائمة الأولى بالعناصر الظاهرة في الصورة عن طريق خاصية Items واجعل قيمة الخاصية SelectionMode الى Multisimple للقائمتين.

4- اصف ثمانية أزرار أمر بالاسماء والترتيب الظاهر على النموذج.

تسمح الخاصية SelectionMode بألية تحديد العناصر من القائمة:

None	لا تسمح باختيار أي عنصر من القائمة
One	تسمح باختيار عنصر واحد فقط من القائمة
MultiSimple	يمكن تحديد أكثر من عنصر من القائمة عن طريق النقر بالفأرة والضغط على أحد المفاتيح Ctrl أو Shift.
MultiExtended	تُمكن من تحديد أكثر من عنصر من القائمة عن طريق الفأرة والضغط على مفتاح Ctrl وكذلك تسمح بتحديد كل عناصر القائمة من خلال الفأرة ومفتاح Shift.

يمكن التعامل مع عناصر كل قائمة من خلال ما يأتي:

الخاصية **ListBox.Items.Count** تمثل عدد عناصر القائمة.

الخاصية **ListBox.Items(i)** تُعطي العنصر من القائمة والذي يحمل الرقم i، علماً ان ترقيم عناصر القائمة يبدأ من الصفر الى **ListBox.Items.Count-1**.

الخاصية **SelectedItems** تستخدم لمعرفة اذا كان العنصر محدد (فتأخذ القيمة True) واذا كان غير محدد تأخذ القيمة False. الخاصية **ListBox.Items.add(Item)** تُضيف العنصر Item الى القائمة.

الخاصية **ListBox.Items.Remove(ListBox.SelectedItems)** حذف العنصر المحدد من القائمة.

الطريقة **ListBox.Items.Clear** تسمح جميع عناصر القائمة (افراغ القائمة).

الخاصية **ListBox.Items.Insert(Index, Item)** تُضيف عنصر جديد للقائمة Item حسب الموقع المحدد Index.

صندوق الرسالة **MessageBox.Show(text)** يُستخدم لإظهار رسالة داخل صندوق قد تتضمن ايقونة أو عنوان.

5- في Class Form1 اكتب الاجراء التالي والذي يقوم بإلغاء تفعيل ازرار ">>" ، ">" ، "<<" و "Remove" و "Removeall" اذا كانت القائمة الاولى فارغة، وإلا فانه يقوم بتفعيلهم، كذلك فانه يقوم بالغاء تفعيل ازرار "<" و "<<" اذا كانت القائمة الثانية فارغة، وإلا فانه يقوم بتفعيلها.

```
Sub EnableButtons()
    If ListBox1.Items.Count > 0 Then
        Button1.Enabled = True
        Button2.Enabled = True
    End If
End Sub
```

```

        Button7.Enabled = True
        Button8.Enabled = True
    Else
        Button1.Enabled = False
        Button2.Enabled = False
        Button7.Enabled = False
        Button8.Enabled = False
    End If
    If ListBox2.Items.Count > 0 Then
        Button3.Enabled = True
        Button4.Enabled = True
    Else
        Button3.Enabled = False
        Button4.Enabled = False
    End If
End Sub

```

6- اكتب الاجراء التالي على زر ">"

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim i As Integer
    If ListBox1.SelectedItem = Nothing Then
        MsgBox("الاولى القائمة من عنصر اختر", MsgBoxStyle.Critical, "خطأ")
        Exit Sub
    Else
        Do While i <= ListBox1.Items.Count - 1
            If ListBox1.SelectedItem <> Nothing Then
                ListBox2.Items.Add(ListBox1.SelectedItem)
                ListBox1.Items.Remove(ListBox1.SelectedItem.ToString)
            Else
                i += 1
            End If
        Loop
        EnableButtons()
    End If
End Sub

```

7- اكتب الاجراء التالي على ">>"

```

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Dim i As Integer
    While i <= ListBox1.Items.Count - 1
        ListBox2.Items.Add(ListBox1.Items(i))
        i += 1
    End While
    ListBox1.Items.Clear()
    EnableButtons()
End Sub

```

8- اكتب الاجراء التالي على "<"

```

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click

```

```

Dim i As Integer
If ListBox2.SelectedItem = Nothing Then
    MsgBox("القائمة من عنصر اختر", MsgBoxStyle.Critical, "خطأ")
    Exit Sub
Else
    Do While i <= ListBox2.Items.Count - 1
        If ListBox2.SelectedItem <> Nothing Then
            ListBox1.Items.Add(ListBox2.SelectedItem)
            ListBox2.Items.Remove(ListBox2.SelectedItem.ToString)
        Else
            i += 1
        End If
    Loop
    EnableButtons()
End If
End Sub

```

9- اكتب الاجراء التالي على "<<"

```

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    Dim i As Integer
    While i <= ListBox2.Items.Count - 1
        ListBox1.Items.Add(ListBox2.Items(i))
        i += 1
    End While
    ListBox2.Items.Clear()
    EnableButtons()
End Sub

```

10- اكتب الاجراء التالي على "Add"

```

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
    Dim x As String = InputBox("بيانات ادخال", "الجديد العنصر ادخال يرجى")
    ListBox1.Items.Add(x)
    EnableButtons()
End Sub

```

11- اكتب الاجراء التالي على "Insert"

```

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
    Dim namex As String = InputBox("بيانات ادخال", "الجديد العنصر ادخل")
    Dim nux As Integer = InputBox("بيانات ادخال", "الجديد العنصر ترتيب ادخل")
    ListBox1.Items.Insert(nux, namex)
    EnableButtons()
End Sub

```

12- اكتب الاجراء التالي على "Remove"

```

Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
    If ListBox1.SelectedItem <> "" Then
        ListBox1.Items.Remove(ListBox1.SelectedItem)
    Else

```

```
        MessageBox.Show("الاولى القائمة من عنصر اختر")
    End If
    EnableButtons()
End Sub
```

13- اكتب الاجراء التالي على "Removeall"

```
Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
    ListBox1.Items.Clear()
    EnableButtons()
End Sub
```

المشروع 12

التعامل مع الملفات النصية

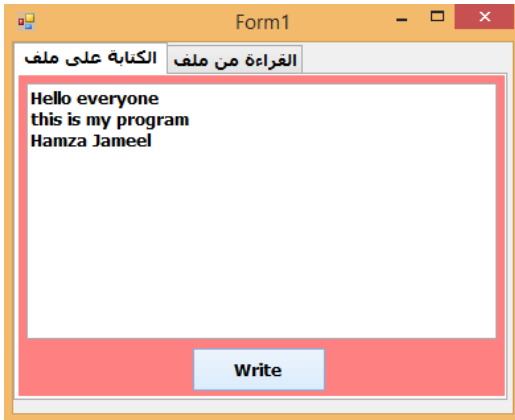
ستتقن من خلال المشروع

التعامل مع ما يأتي:

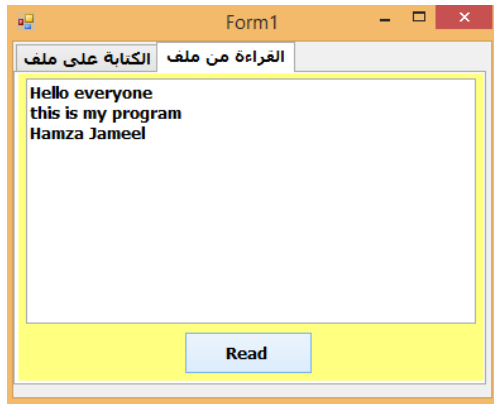
- عنصر TabControl
- مكتبة System.IO
- الكتابة على الملفات باستخدام StreamWriter
- القراءة من الملفات باستخدام StreamReader

اهداف المشروع

عند التنفيذ تظهر واجهة تحتوي على تبويبين:



- الاول باسم "الكتابة على الملف" والذي يتطلب ادخال نص وعند النقر على زر "Write" يكتب النص على ملف محفوظ في مكان محدد برمجياً.



- الثاني باسم "القراءة من ملف" عند النقر عليه تظهر الواجهة الخاصة به وعندما يتم النقر على زر "Read" يُنقل النص من الملف الى صندوق النص الموجود على التبويب.

مراحل المشروع

1- انشئ مشروعاً جديداً.

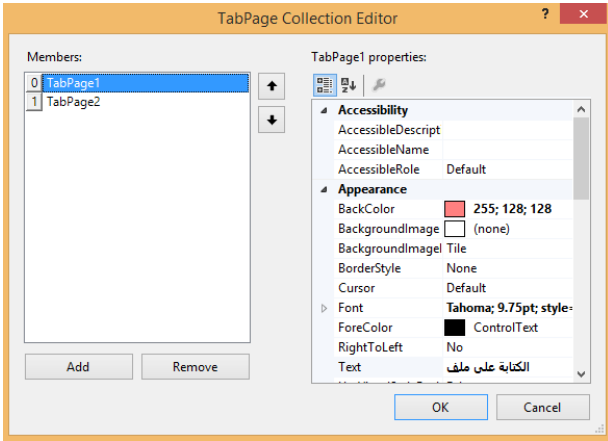
2- اضع أداة TabControl وضع بداخل كل صفحة من التبويب (TextBox و Button) ، يمكن التنقل بين التبويبات من خلال النقر على العنوان الخاص بكل تبويب.

يمكن التحكم بخصائص التبويب من خلال TabPages ستظهر نافذة تحرير التبويب التي تتكون من جزئين:

- الاول أسماء صفحات التبويب وفي الاسفل زري امر "Add" لاضافة تبويب و"Remove" لحذف التبويب

- الجزء الثاني يضم الخصائص الخاصة بكل صفحة

اختر منها Text و BackColor.



في قسم General Declaratio اكتب السطر البرمجي التالي:

Imports System.IO

في هذا الامر يتم استدعاء الفضاء الخاص بعملية الادخال والايخارج.

3- اكتب الاجراء التالي في زر "Write"

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim myfile As New StreamWriter("E:\myfile.txt")
    myfile.WriteLine(TextBox1.Text)
    myfile.Close()
End Sub
```

StreamWriter تستخدم للكتابة في ملف نصي.

4- اكتب الاجراء التالي في زر "Read"

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Dim myfile As New StreamReader("E:\myfile.txt")

    TextBox2.Text = myfile.ReadToEnd()
    myfile.Close()
End Sub
```

StreamReader تستخدم للقراءة من ملف نصي.

المشروع 13

دوال التعامل مع النص

ستتقن من خلال المشروع

التعامل مع الدوال التالية:

- Length
- Substring
- Replace
- Trim, RTrim, LTrim
- ToLower
- ToUpper
- Asc
- Chr

اهداف المشروع

عند التنفيذ تظهر الواجهة التالية:



- في الجزء الأعلى عند كتابة نص بحروف كبيرة في الصندوق الأعلى لليسار يظهر النص بحروف صغيرة في الصندوق الأسفل. وفي الصندوق الأعلى المحاذي لليمين عند كتابة نص بحروف صغيرة يظهر النص بحروف كبيرة في الصندوق الأسفل.
- في القسم الثاني من الواجهة يسمح زر "استبدال النص" باستبدال كل ظهور لنص بنص اخر في صندوق النص "الاصلي" ويضع النص بعد الاستبدال في صندوق النص "النتائج".

مراحل المشروع

1- انشئ مشروعاً جديداً.

2- اضع الأدوات الظاهرة على النموذج أعلاه.

دوال التعامل مع النصوص

دالة Length

تُعيد هذه الدالة عدد أحرف السلسلة النصية.مثال

```
Dim s1 As String = "Hamza Jameel"  
MsgBox(s1.Length)
```

تُعيد 12

قامت بحساب الفراغ بين الكلمات ايضاً

دالة SubString

تعمل هذه الدالة اقتطاع جزء من النص length المطلوب

شكلها العام:

Str.Substring(startIndex,Length)

Startindex: موقع بداية القص ويبدأ من الصفر و Length يمثل عدد الأحرف المشمولة بالاقتطاع. مثال

```
Dim s1 As String = "Hamza Jameel"  
MsgBox(s1.Substring(0, 4))
```

تُعيد " Hamz"

يمكن ان نبدأ اللاقتطاع من اي موقع من النص كـ 1 او 3 حسب رغبة المبرمج.

دالة Trim

تعمل على إزالة الفراغات من يمين ويسار النص. مثال

```
Trim(" Hamza ")
```

تُعيد "Hamza"

دالة RTrim

تعمل هذه الدالة مسح الفراغات من يمين النص. مثال

```
Trim(" Hamza ")
```

تُعيد " Hamza"

دالة LTrim

تعمل هذه الدالة مسح الفراغات من يسار النص. مثال

```
Trim(" Hamza ")
```

تُعيد "Hamza"

دالة ToLower

تحول هذه الدالة أحرف السلسلة النصية الى احرف صغيرة. مثل

```
Dim s1 As String = "HAMZA"  
MsgBox(s1.ToLower)
```

تُعيد "hamza"

ToUpper دالة

تحول أحرف السلسلة النصية الى أحرف كبيرة. مثل

```
Dim s1 As String = "hamza"  
MsgBox(s1.ToUpper)
```

"HAMZA" تُعيد

Asc دالة

تُعيد الرقم المكافئ للحرف المدخل والموجود في جدول ASCII. مثال

```
Asc("A")
```

تُعيد 65.

Chr دالة

تُعيد هذه الدالة الحرف المكافئ للرقم المُدخل. مثال

```
Chr(65)
```

4- الاجراء الخاص بتحويل احرف السلسلة النصية الى احرف صغيرة.

```
Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles  
TextBox1.TextChanged  
Dim str1 As String  
str1 = TextBox1.Text  
TextBox2.Text = str1.ToLower  
End Sub
```

5- الاجراء الخاص بتحويل أحرف السلسلة النصية الى أحرف كبيرة.

```
Private Sub TextBox3_TextChanged(sender As Object, e As EventArgs) Handles  
TextBox3.TextChanged  
Dim str1 As String  
str1 = TextBox3.Text  
TextBox4.Text = str1.ToUpper  
End Sub
```

6- برمجة زر "استبدال النص"

Replace دالة : تستخدم هذه الدالة لابدال نص مكان نص اخر، وتكون حساسة لحالة الاحرف

شكلها العام:

Str.Replace(oldstring, newstring)

Oldstring تمثل النص القديم المراد ابداله، **newstring** تمثل النص الجديد الذي سيُبدل به

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Dim str1 As String = TextBox7.Text
    TextBox8.Text = str1.Replace(TextBox5.Text, TextBox6.Text)
End Sub
```

المشروع 14

التعامل مع صندوق النص الغني

ستتقن من خلال المشروع

التعامل مع صندوق النص الغني RichTextBox:

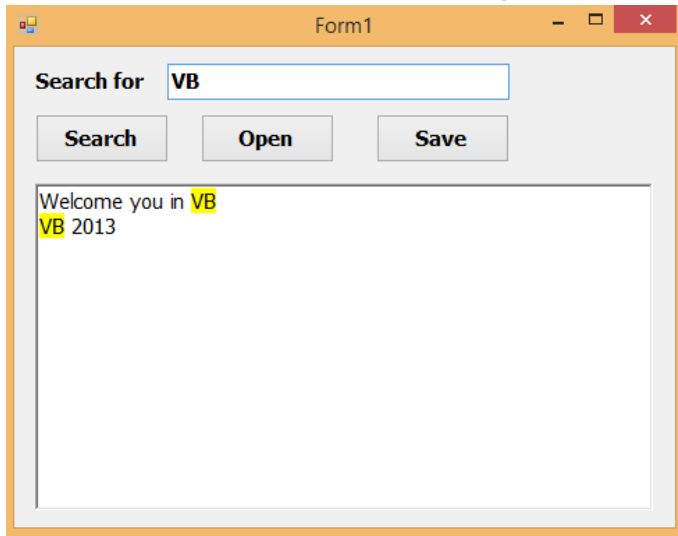
اجراء العمليات التالية:

- حفظ نص الصندوق في ملف نصي
- فتح ملف نصي داخل الصندوق
- البحث عن كلمات داخل الصندوق

اهداف المشروع

عند التنفيذ تظهر الواجهة التالية:

- عند ادخال نص في صندوق النص والنقر على زر "Search" يقوم البرنامج بالبحث داخل صندوق النص الغني عن جميع



النصوص المطابقة له وتظليلها باللون الأصفر، وعند البحث مرة اخرى عن نص آخر يقوم البرنامج بإلغاء تظليل النصوص القديمة وتظليل الأخر وهكذا.

- عند النقر على زر "Open" تظهر نافذة يمكنك من اختيار ملف نصي ليتم فتحه داخل صندوق النص الغني.
- عند النقر على زر "Save" تظهر نافذة تسمح باختيار مكان لحفظ الملف النصي باسم معين.

مراحل المشروع

1- انشئ مشروعاً جديداً.

2- اضع الأدوات الظاهرة على النموذج.

بعض الدوال المستخدمة مع RichTextBox:

دالة TextLength

تُعيد عدد يمثل طول السلسلة النصية داخل صندوق النص الغني.

دالة LastIndexOf(string)

تُعيد قيمة رقمية معتمدة على موقع اخر ظهور للنص المحدد ضمن الكائن(صندوق النص).

دالة Find

تستخدم للبحث عن نص وتعيد قيمة رقمية ويمكنها التحكم وتحديد آلية البحث عن النص في صندوق النص الغني.

دالة SelectionBackColor

تستخدم لوضع لون على النص المُحدد في صندوق النص الغني.

دالة IndexOf

تعيد قيمة رقمية معتمده على اول ظهور للنص المحدد ضمن الكائن، كما ان عملية البحث تبدأ بموقع النص المحدد.

3- اكتب الاجراء التالي في class Form1

وتتضمن إلغاء تفعيل ازرار "البحث"، "الحفظ" و "صندوق النص" اذا كان صندوق النص الغني فارغ.

```
Sub Enabletools()  
If RichTextBox1.TextLength = Nothing Then  
    TextBox1.Enabled = False  
    Button2.Enabled = False  
    Button3.Enabled = False  
Else  
    TextBox1.Enabled = True  
    Button2.Enabled = True  
    Button3.Enabled = True  
End If  
End Sub
```

4- اكتب الاجراء التالي في زر "Search"

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click  
    Dim index As Integer = 0  
    Dim temp As String = RichTextBox1.Text  
    RichTextBox1.Text = ""  
    RichTextBox1.Text = temp  
    While index < RichTextBox1.Text.LastIndexOf(TextBox1.Text)  
        RichTextBox1.Find(TextBox1.Text, index, RichTextBox1.TextLength,  
RichTextBoxFinds.None)  
        RichTextBox1.SelectionBackColor = Color.Yellow  
        index = RichTextBox1.Text.IndexOf(TextBox1.Text, index) + 1  
    End While  
    Enabletools()  
End Sub
```

5- اكتب الاجراء التالي في زر "Open"

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click  
    OpenFileDialog1.Filter = "textfile|*.txt"  
    If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then  
        RichTextBox1.Text =  
My.Computer.FileSystem.ReadAllText(OpenFileDialog1.FileName)  
    End If  
    Enabletools()  
End Sub
```

6- اكتب الاجراء التالي في زر "Save"

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    SaveFileDialog1.Filter = "textfile|*.txt"
    If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
        My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName,
RichTextBox1.Text, True)
    End If
End Sub
```

7- اكتب التالي في Form1_Load

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

8- اكتب التالي في RichTextBox1_TextChanged

```
Private Sub RichTextBox1_TextChanged(sender As Object, e As EventArgs) Handles
RichTextBox1.TextChanged
    Enabletools()
End Sub
```

المشروع 15

إنشاء مفكرة (خطوات متقدمة)

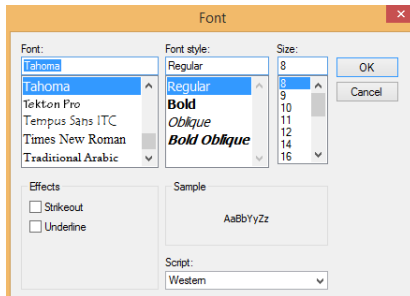
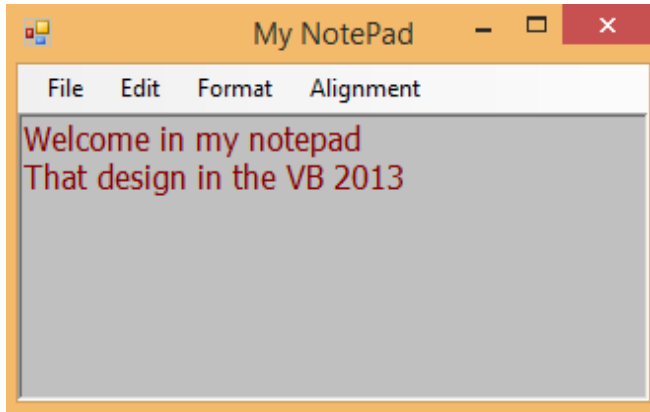
ستتقن من خلال المشروع

التعامل مع الأدوات التالية:

- صندوق النص الغني RichTextBox
- شريط القوائم MenuStrip
- قائمة زر الفأرة الايمن ContextMenuStrip

اهداف المشروع

عند التنفيذ تظهر الواجهة التالية التي تضم صندوق نص غني للكتابة وشريط قوائم لإجراء عمليات مختلفة على النص. يمكن للمستخدم الكتابة وإجراء مجموعة من التنسيقات على النص

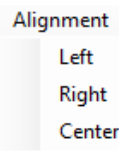


- لتغيير نوع ولون وحجم الخط اختر Format/Font فتظهر نافذة تسمح للمستخدم باختيار التنسيق المناسب.

- لتغيير لون الخط اختر Format/ForeColor فتظهر نافذة الالوان القياسية التي تسمح للمستخدم باختيار اللون المطلوب.

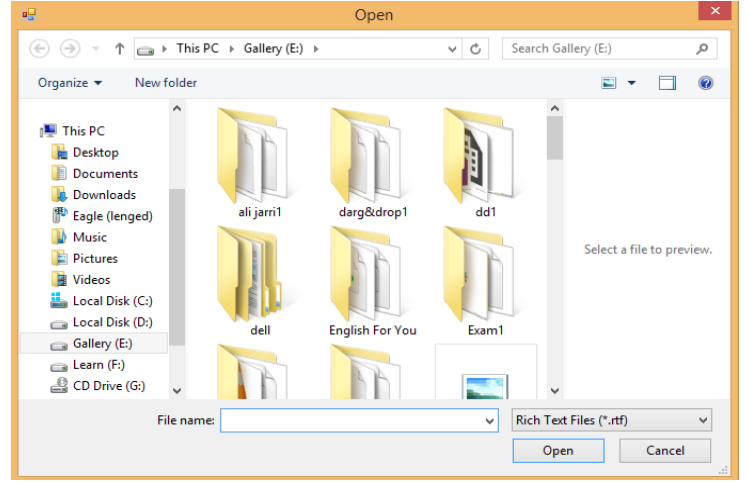
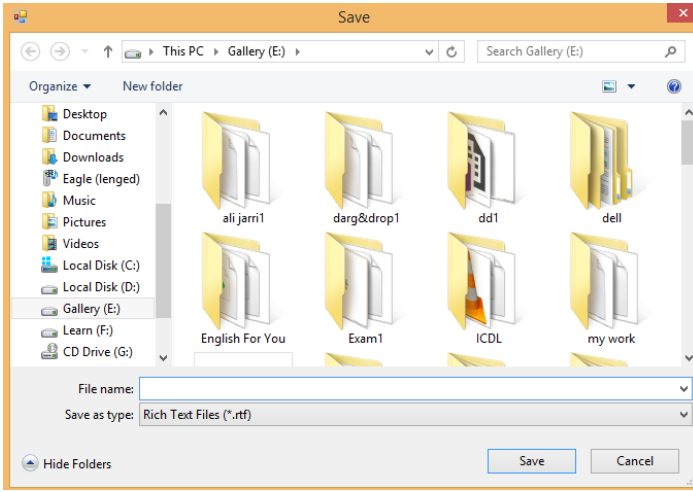


- كذلك يمكن تغيير لون خلفية صندوق النص الغني من خلال Format/BackColor حيث تظهر نافذة الالوان القياسية ايضاً.

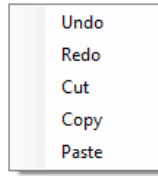


- كما يمكن تغيير محاذاة النص. حيث يتم اختيار نوع المحاذاة من خلال قائمة Alignment

- حفظ البيانات، يتم من خلال الامر File/Save حيث تظهر نافذة الحفظ القياسية لاختيار مكان الحفظ. فتح ملف من خلال الأمر File/Open ايضاً تظهر نافذة تسمح باختيار الملف المطلوب، لإنشاء ملف جديد من خلال الأمر File/New.



- كما ان عمليات تحرير النص كالتقدم، التراجع، القص، النسخ ، اللصق وتحديد كامل النص في الصندوق بالاضافة الى حذف النص تتم من خلال قائمة Edit.
- عند النقر بزر الفأرة الأيمن داخل صندوق النص الغني تظهر قائمة منبثقة تضم عدة أوامر.

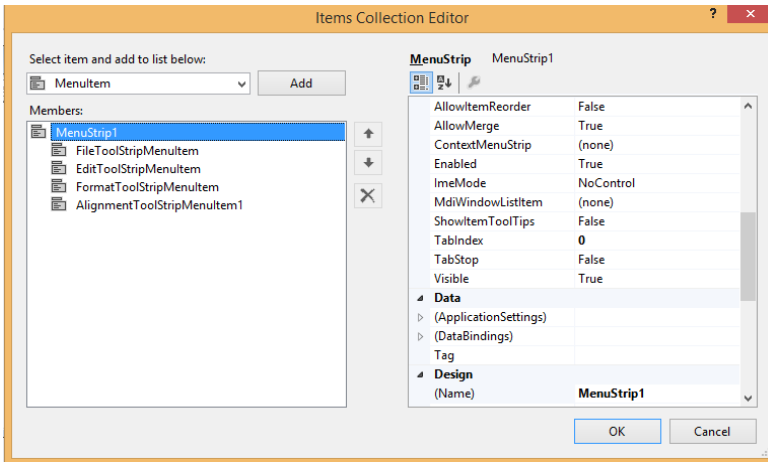


مراحل المشروع

1- انشئ مشروعاً جديداً.

2- اصف الأدوات التالية الى النموذج.

- **MenuStrip**: تُستخدم هذه الأداة لإنشاء شريط قوائم للبرنامج بالاضافة الى امكانية إضافة شريط القوائم القياسي (هو شريط معد مسبقاً يحتوي على مجموعة من القوائم وكل قائمة تضم عدة أوامر) عن طريق النقر بالفأرة على السهم الخاص بالأداة واختيار الأمر **Insert Standard Items** قد تظهر صورة صغيرة بجانب بعض الأوامر الخاصة بهذا الشريط.



كما يمكن حذف او اضافة قائمة او امر من خلال خاصية **Items** ستظهر نافذة تتكون من جزئين وكل جزء يحتوي على خيارات يمكن من خلالها تغيير ما يلزم من محتويات الشريط.

يحتوي الجزء الأيسر من النافذة على الأزرار التالية:

- زر **Add** يستخدم لإضافة قائمة جديدة للشريط .
- السهمين الجانبيين لتغيير تسلسل قوائم الشريط حيث يعمل السهم باتجاه الأعلى لرفع القائمة والسهم باتجاه الأسفل لانزال القائمة.
- يستخدم زر X الجانبية لحذف القائمة المطلوبة.

يحتوي الجزء الأيمن من النافذة على الخصائص المتعلقة بكل قائمة موجودة ضمن الشريط ومنها يمكن اختيار الاسم البرمجي الخاص بكل قائمة **Name**، واختيار **Text** وتمثل الاسم الذي يظهر عند التنفيذ، أو اضافة صورة صغيرة لأمر معين من خلال خاصية **Image**. بالاضافة الى خاصية **ShortcutKeys** وتسمح بالوصول الى القائمة من خلال الاختصار المرافق لها. حيث توجد مجموعة من الاختصارات يمكن اختيار الاختصار المناسب لكل أمر ضمن القوائم.

- **ContextMenuStrip**: تستخدم هذه الأداة لظهور القائمة المنبثقة لزر الفأرة الأيمن بعد اختيارها املاًها بالأوامر المطلوبة.

▪ **RichTextBox**: ثم انقر على السهم الموجود في أعلى الأداة واختر Dock in Parent Container لجعل العنصر يملئ النموذج. ثم انقر على صندوق النص واجعل خاصية **ContextMenuStrip=ContextMenuStrip1** القائمة المنبثقة داخل صندوق النص عند النقر بالزر الايمن للفأرة.

ملاحظة لظهار القائمة المنبثقة خلال فترة تصميم البرنامج انقر على اسم الأداة واستخدم الفأرة ولوحة المفاتيح لمثلها بالأوامر.

3- في Class Form اكتب الدالة التالية:

التي تستخدم لفحص محتوى الصندوق

```
Private Function checkrich() As Boolean
    If RichTextBox1.TextLength = Nothing Then
        checkrich = True
    Else
        checkrich = False
    End If
End Function
```

واكتب ايضاً الاجراء التالي الذي يقوم بحفظ محتوى صندوق النص في ملف.

Catch في حالة حدوث اي خطأ ينتقل الى ما بعد Exception

```
Private Sub savetext()
    Try
        Dim savedlog As SaveFileDialog = New SaveFileDialog
        savedlog.Title = "Save"
        savedlog.Filter = "Rich Text Files (*.rtf)|*.rtf"
        If savedlog.ShowDialog() = System.Windows.Forms.DialogResult.OK Then
            RichTextBox1.SaveFile(savedlog.FileName,
RichTextBoxStreamType.RichText)
        End If
    Catch ex As Exception : End Try
End Sub
```

4- اكتب الإجراء التالي في أمر "File/New"

في هذا الاجراء يتم التحقق من محتوى الصندوق قبل البدء بحذف محتواه.

```
Private Sub NewToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
NewToolStripMenuItem.Click
    If checkrich() = True Then
        RichTextBox1.Clear()
    Else
        savetext()
        RichTextBox1.Text = ""
    End If
End Sub
```

5- اكتب الإجراء التالي في "File/Open"

```
Private Sub OpenToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
OpenToolStripMenuItem.Click
    Try
        Dim opendlog As OpenFileDialog = New OpenFileDialog
```

```

    opendlog.Title = "Open"
    opendlog.Filter = "Rich Text Files (*.rtf)|*.rtf"
    If opendlog.ShowDialog() = System.Windows.Forms.DialogResult.OK Then
        RichTextBox1.LoadFile(opendlog.FileName)
    End If
Catch ex As Exception : End Try
End Sub

```

6- اكتب الاجراء التالي في أمر File/Save حيث سنكتفي باستدعاء اجراء الحفظ فقط.

```

Private Sub SaveToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
SaveToolStripMenuItem.Click
    savetext()
End Sub

```

7- اكتب الاجراء التالي في أمر File/Exit

```

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

```

8- اكتب الاجراء التالي في أمر Edit/Undo

```

Private Sub UndoToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
UndoToolStripMenuItem.Click
    RichTextBox1.Undo()
End Sub

```

9- برمجة الأمر Edit/Redo

```

Private Sub RedoToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
RedoToolStripMenuItem.Click
    RichTextBox1.Redo()
End Sub

```

10- اكتب الاجراء التالي في أمر Edit/Cut

```

Private Sub CutToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
CutToolStripMenuItem.Click
    RichTextBox1.Cut()
End Sub

```

11- اكتب الاجراء التالي في أمر Edit/Copy

```

Private Sub CopyToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
CopyToolStripMenuItem.Click
    RichTextBox1.Copy()
End Sub

```

12- اكتب الاجراء التالي في أمر Edit/Paste

```
Private Sub PasteToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
PasteToolStripMenuItem.Click
    RichTextBox1.Paste()
End Sub
```

13- اكتب الاجراء التالي في Edit/Selectall

```
Private Sub SelectAllToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles SelectAllToolStripMenuItem.Click
    RichTextBox1.SelectAll()
End Sub
```

14- اكتب الاجراء التالي في Edit/Clear

```
Private Sub ClearToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ClearToolStripMenuItem.Click
    RichTextBox1.Clear()
End Sub
```

15- اكتب الاجراء التالي في Format/Font

```
Private Sub CustomizeToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles CustomizeToolStripMenuItem.Click
    Try
        Dim fontdlg As FontDialog = New FontDialog
        fontdlg.Font = RichTextBox1.Font
        If fontdlg.ShowDialog = System.Windows.Forms.DialogResult.OK Then
            RichTextBox1.Font = fontdlg.Font
        End If
    Catch ex As Exception : End Try
End Sub
```

16- اكتب الاجراء التالي في Format/ForeColor

```
Private Sub OptionsToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles OptionsToolStripMenuItem.Click
    Try
        Dim colordlg As ColorDialog = New ColorDialog
        colordlg.Color = RichTextBox1.ForeColor
        If colordlg.ShowDialog = System.Windows.Forms.DialogResult.OK Then
            RichTextBox1.ForeColor = colordlg.Color
        End If
    Catch ex As Exception : End Try
End Sub
```

17- اكتب الاجراء التالي في Format/BackColor

```
Private Sub BackColorToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles BackColorToolStripMenuItem.Click
    Try
        Dim colordlg As ColorDialog = New ColorDialog
        colordlg.Color = RichTextBox1.BackColor
        If colordlg.ShowDialog = System.Windows.Forms.DialogResult.OK Then
            RichTextBox1.BackColor = colordlg.Color
        End If
    Catch ex As Exception : End Try
End Sub
```

```
End If
Catch ex As Exception : End Try
End Sub
```

18- اكتب الإجراء التالي في Alignment/Left

```
Private Sub LeftToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
LeftToolStripMenuItem.Click
    RichTextBox1.SelectionAlignment = HorizontalAlignment.Left
End Sub
```

19- اكتب الإجراء التالي في Alignment/Right

```
Private Sub rightToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
CenterToolStripMenuItem.Click
    RichTextBox1.SelectionAlignment = HorizontalAlignment.Right
End Sub
```

20- اكتب الإجراء التالي في Alignment/Center

```
Private Sub centerToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles RightToolStripMenuItem.Click
    RichTextBox1.SelectionAlignment = HorizontalAlignment.Center
End Sub
```

تنبيه فيما يخص التعليمات البرمجية الخاصة بالأوامر الموجود في القائمة المنبثقة انقر على كل أمر ثم ضع التعليمات التي تتناسب مع وظيفته.

المشروع 16

انشاء حاسبة رقمية

ستتقن من خلال المشروع

- التعامل مع أدوات التحكم
- التعامل مع الدوال الرياضية
- التعامل مع تعليمات الخطأ

Resume –

Catch –

اهداف المشروع

بناء آلة حاسبة تمكن من اجراء العمليات الحسابية:

- العمليات الحسابية الاساسية +, -, *, /,
- الجزر التربيعي Sqrt.
- عمليات Sin, Cos, Tan
- مربع العدد x^2 ، مكعب العدد x^3 والنسبة المئوية %.

مراحل المشروع

1- انشئ مشروعاً جديداً.

2- اصف صندوق نص وعشرة أزرار أمر واضبط اسمائها بالارقام (0,1,2,..9) واكتب التعليمات التالية في كل زر امر :

TextBox1.AppendText (0), TextBox1.AppendText (1),... TextBox1.AppendText(9)

تعليمة **On Error Resume Next** تعني عند حدوث اي خطأ في تنفيذ تعليمة ينتقل التنفيذ الى التعليمة التالية.

تعليمة **Convert** تستخدم لتحويل المتغير الى انواع بيانية مختلفة ك(string, double, ...).

دالة InStr

تُعيد هذه الدالة رقم يحدد أول ظهور لسلسلة نصية string2 في سلسلة نصية اخرى، واذا لم تجد السلسلة النصية string2 في string1 ستكون القيمة المُرجعة صفر. مثل

InStr("Tomsim", "sim")

تُعيد 4 اي ان السلسلة "sim" موجود في السلسلة (" Tomsim") اعتباراً من الحرف رقم 4.

شكلها العام:

InStr(string1,string2,[Compare])

- String1 هو النص الذي نبحث فيه.
- String2 هو النص الذي نبحث عنه.
- Compare هي طريقة المقارنة وتكون اختيارية.

3- في قسم التصريحات العامة للنموذج اكتب التعليمات التالية:

```
Dim x, y As Double
Dim Op As Char
```

4- اضع خمسة أزرار أمر وأضبط اسم كل منها بعملية حسابية +، -، *، /، %.

ثم اكتب داخل كل زر أمر التعليمة الخاص بالعملية الرياضية التي يمثلها.

■ عملية الجمع:

```
x = Val(TextBox1.Text)
Op = "+"
TextBox1.Text = ""
```

■ عملية الطرح:

```
x = Val(TextBox1.Text)
Op = "-"
TextBox1.Text = ""
```

■ عملية الضرب:

```
x = Val(TextBox1.Text)
Op = "*"
TextBox1.Text = ""
```

■ عملية القسمة:

```
x = Val(TextBox1.Text)
Op = "/"
TextBox1.Text = ""
```

■ عملية النسبة المئوية:

```
x = Val(TextBox1.Text)
Op = "%"
TextBox1.Text = ""
```

5- اضع زر أمر آخر باسم "+/-" حيث يستخدم لتغيير اشارة العدد من السالب للموجب والعكس صحيح.

واكتب الإجراء التالي:

```
Private Sub Button26_Click(sender As Object, e As EventArgs) Handles Button26.Click
    Dim no1 As Double
    no1 = Convert.ToDouble(Val(TextBox1.Text))
    If no1 = 0 Then
        Exit Sub
    End If
End Sub
```

```

Else
    Dim no2 As Double
    no2 = no1 - no1 - no1
    TextBox1.Text = Convert.ToString(no2)
End If
End Sub

```

6- اضع زر أمر اخر باسم "Bk" ويستخدم لحذف مراتب من الرقم المدخل:
واكتب الاجراء التالي.

```

Private Sub Button20_Click(sender As Object, e As EventArgs) Handles
Button20.Click
    If TextBox1.TextLength > 0 Then
        TextBox1.Text = TextBox1.Text.Remove(TextBox1.TextLength - 1)
    End If
End Sub

```

7- اضع زر أمر اخر باسم "Sin" يستخدم لايجاد جيب زاوية معينة.
واكتب الاجراء التالي:

```

Private Sub Button19_Click(sender As Object, e As EventArgs) Handles
Button19.Click
    Dim sn As Double
    sn = Convert.ToDouble(Val(TextBox1.Text))
    TextBox1.Text = Convert.ToString(Math.Sin(sn))
End Sub

```

8- اضع زر أمر اخر باسم "Cos" يستخدم لايجاد جيب تمام الزاوية.
واكتب الاجراء التالي:

```

Private Sub Button18_Click(sender As Object, e As EventArgs) Handles
Button18.Click
    Dim cosn As Double
    cosn = Convert.ToDouble(Val(TextBox1.Text))
    TextBox1.Text = Convert.ToString(Math.Cos(cosn))
End Sub

```

9- اضع زر باسم "Tan" حيث يستخدم لايجاد ظل تمام الزاوية.
واكتب الاجراء التالي:

```

Private Sub Button17_Click(sender As Object, e As EventArgs) Handles
Button17.Click
    Dim tn As Double
    tn = Convert.ToDouble(Val(TextBox1.Text))
    TextBox1.Text = Convert.ToString(Math.Tan(tn))
End Sub

```

10- اضع زر أمر باسم "1/x" يستخدم لايجاد مقلوب العدد.

```
Private Sub Button21_Click(sender As Object, e As EventArgs) Handles
Button21.Click
    On Error Resume Next
    TextBox1.Text = 1 / TextBox1.Text
End Sub
```

من المشاكل التي يمكن ان تواجه المستخدم هي ادخال قيمة صفر او ترك صندوق النص فارغ وينقر على هذا الزر عندها سيحدث خطأ في البرنامج، ولتلافي مثل هكذا اخطاء استخدمنا تعليمة **On Error** في الاجراء السابق.

11- اصف زر أمر باسم "C" يستخدم لحذف بيانات صندوق النص.

واكتب الاجراء التالي:

```
Private Sub Button27_Click(sender As Object, e As EventArgs) Handles
Button27.Click
    TextBox1.Text = ""
End Sub
```

12- اصف زر أمر باسم "." يستخدم لوضع الفاصلة للأعداد العشرية، لايمكن اضافة الفاصلة في بداية العدد كـ "34." كما لا يمكن وضع اكثر من فاصلة في العدد الواحد كـ "2.5.4"، ولهذا يجب التحقق من احتواء صندوق النص على اعداد قبل اضافتها وكذلك يجب التحقق من عدم وجودها مسبقاً في العدد.

ثم اكتب الاجراء التالي:

```
Private Sub Button11_Click(sender As Object, e As EventArgs) Handles
Button11.Click
    If (TextBox1.Text = Nothing) Or (InStr(TextBox1.Text, ".") > 0) Then
        Exit Sub
    Else
        TextBox1.AppendText(".")
    End If
End Sub
```

13- اصف زر أمر باسم "=" يستخدم ليجاد الناتج.

ثم اكتب الاجراء التالي :

```
Private Sub Button10_Click(sender As Object, e As EventArgs) Handles
Button10.Click
    y = Val(TextBox1.Text)
    Select Case Op
        Case "+"
            TextBox1.Text = x + y
        Case "-"
            TextBox1.Text = x - y
        Case "*"
            TextBox1.Text = x * y
        Case "/"
            If y = 0 Then
                MsgBox("Divisoin by Zero", MsgBoxStyle.Critical, "ERROR")
            Exit Sub
    End Select
```



```

        End If
        TextBox1.Text = x / y
    Case "%"
        TextBox1.Text = x * y / 100
    End Select
End Sub

```

14- اضع زر اخر باسم "x²" حيث يستخدم لاجاد مربع العدد.

واكتب الاجراء التالي:

```

Private Sub Button25_Click(sender As Object, e As EventArgs) Handles
Button25.Click
    Dim po2 As Double
    po2 = Convert.ToDouble(Val(TextBox1.Text))
    TextBox1.Text = Convert.ToString(Math.Pow(po2, 2))
End Sub

```

15- اضع زر اخر باسم "x³" ويستخدم لاجاد معكب العدد.

واكتب الاجراء التالي:

```

Private Sub Button24_Click(sender As Object, e As EventArgs) Handles
Button24.Click
    Dim po3 As Double
    po3 = Convert.ToDouble(Val(TextBox1.Text))
    TextBox1.Text = Convert.ToString(Math.Pow(po3, 3))
End Sub

```

16- اضع زر اخر باسم "Sqrt" ويستخدم لاجاد الجذر التربيعي للعدد.

واكتب الاجراء التالي:

```

Private Sub Button23_Click(sender As Object, e As EventArgs) Handles
Button23.Click
    Dim sqr As Double
    sqr = Convert.ToDouble(Val(TextBox1.Text))
    TextBox1.Text = Convert.ToString(Math.Sqrt(sqr))
End Sub

```

المشروع 17

التعامل مع أشرطة المسار

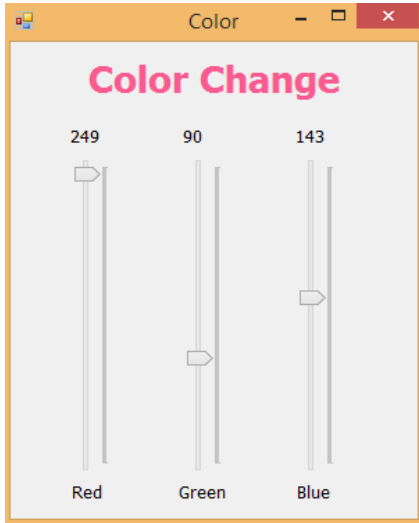
ستتقن من خلال المشروع

التعامل مع شريط المسار TrackBar

دالة RGB

اهداف المشروع

عند التنفيذ تظهر الواجهة التالية:



- يُمكن للمستخدم تغيير لون النص (Color Change) من خلال تغيير درجة اللون الأحمر عن طريق شريط المسار العمودي الأول، ودرجة اللون الأخضر عن طريق شريط المسار العمودي الثاني، ودرجة اللون الأزرق عن طريق شريط المسار الثالث.

مراحل المشروع

1- انشئ مشروعاً جديداً.

2- اصف للنموذج ثلاث أشرطة مسار واضبط الخصائص كالتالي:

Minimum	0
Maximum	255
SmallChange	1
LargeCharge	10
Orientation	Vertical

تصميم شريط المسار TrackBar

يمكن تحويل اتجاه الشريط من افقي الى عمودي وبالعكس من خلال الخاصية **Orientation** حيث ان القيمة **Horizontal** لجعل اتجاه الشريط افقي، والقيمة **Vertical** لجعل اتجاهه عمودي.

اهم خصائص شريط المسار

Minimum	الحد الأدنى لقيمة شريط المسار
Maximum	الحد الأعلى لقيمة شريط المسار

Value	القيمة الحالية لشريط المسار
SmallChange	مقدار التغير في القيمة Value عند النقر على قطعة الشريط
LargeChange	مقدار التغير في القيمة Value عند النقر على أعلى أو أسفل الشريط

اهم أحداث الشريط

Scroll تحريك القطعة المتحركة في شريط المسار.

دالة RGB

تعيد هذه الدالة رقماً صحيحاً يمثل اللون وتأخذ الشكل:

FromArgb(red, green, blue)

حيث ان Red، Green، Blue هي ارقام محصورة بين 0 و 255 وتمثل درجة الأحمر، الأخضر والأزرق.

FromArgb(0, 0, 0) اللون الاسود

FromArgb(255, 255, 255) اللون الابيض

3- اضع ستة Label ، ثم سم ثلاث منها Red، Green، Blue وضعها اسفل الاشرطة، ثم ضع الثلاثة الأخرى أعلى الاشرطة واترك اسمائها الافتراضية كما هي لاننا سنتعامل معها برمجياً.

4- ضع التعليمات البرمجية التالية في قسم التصريحات للـ Form

```
Dim red As Integer
Dim green As Integer
Dim blue As Integer
```

5- اكتب الاجراء التالي في قسم التصريحات للـ Form

```
Private Sub changecolor()
    Dim mycolor As Color = Color.FromArgb(red, green, blue)
    Me.Label4.ForeColor = mycolor
    Me.Refresh()
End Sub
```

6- انقر على الشريط الاول واكتب الاجراء التالي:

```
Private Sub TrackBar1_Scroll(sender As Object, e As EventArgs) Handles
TrackBar1.Scroll
    red = TrackBar1.Value
    Label1.Text = TrackBar1.Value
    changecolor()
End Sub
```

7- انقر على الشريط الثاني واكتب الاجراء التالي:

```
Private Sub TrackBar2_Scroll(sender As Object, e As EventArgs) Handles
TrackBar2.Scroll
    green = TrackBar2.Value
    Label2.Text = TrackBar2.Value
    changecolor()
End Sub
```

End Sub

8- انقر على الشريط الثالث واكتب الاجراء التالي:

```
Private Sub TrackBar3_Scroll(sender As Object, e As EventArgs) Handles
TrackBar3.Scroll
    blue = TrackBar3.Value
    Label3.Text = TrackBar3.Value
    changecolor()
End Sub
```

9- استدعي الاجراء داخل حدث تحميل النموذج:

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

المشروع 18

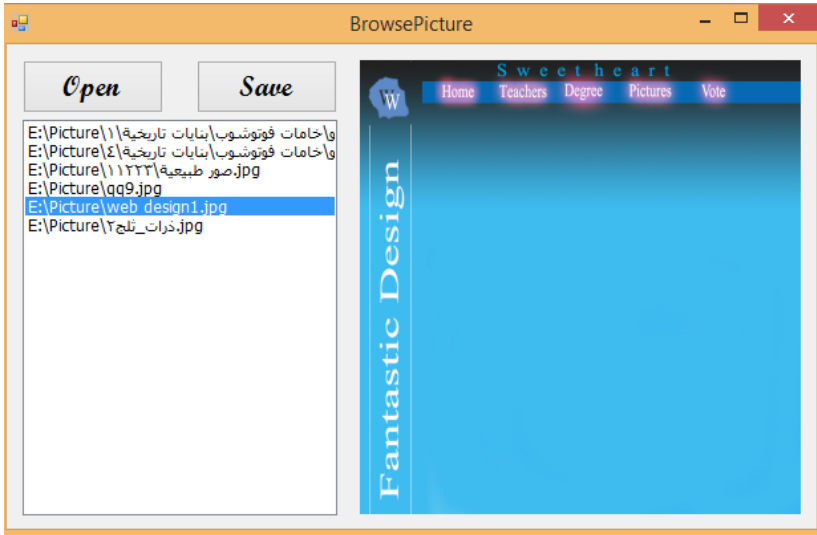
عرض صور (1)

ستتقن من خلال المشروع

- التعامل مع صندوق الصورة PictureBox
- التعامل مع نوافذ الحوار
- التعامل مع ListBox

اهداف المشروع

عند التنفيذ يظهر النموذج التالي:



- عندما ينقر المستخدم على زر "Open" تظهر نافذة لاختيار صورة وفتحها في صندوق الصورة. ووضع اسم الملف ومساره داخل صندوق القائمة.
- عند النقر على مسار الملف الموجود في صندوق القائمة تُفتح الصورة في صندوق الصورة.
- عند النقر على زر "Save" تظهر نافذة لخصر الصورة باسم معين وبمكان من اختيار المستخدم.

مراحل المشروع

- 1- انشئ مشروعاً جديداً.
- 2- اصف للنموذج زررين امر واضبط الاسم "Open" و "Save" وصندوق قائمة بالإضافة الى صندوق صورة واجعل خاصية **SizeMode** للقيمة **StretchImage**.
- 3- اصف للنموذج **OpenFileDialog** و **SaveFileDialog** قيم خاصية **SizeMode** هي:

Normal	ابقاء الصورة على حجمها الطبيعي دون تغييره
StretchImage	جعل حجم الصورة مساوي لحجم صندوق الصورة على النموذج
AutoSize	جعل حجم صندوق الصورة مساوياً لحجم الصورة
CenterImage	ابقاء حجم صندوق الصورة ثابت واطهار مركز الصورة اذا كانت اكبر من حجم الصندوق
Zoom	تغيير أبعاد صندوق الصورة حسب حجم الصورة مع الاحتفاظ بابعاد العنصر دون تجاوزها داخل النموذج

4- اكتب الاجراء التالي في زر "Open"

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    OpenFileDialog1.Filter = "JpegImage|*.Jpg|BmpImage|*.Bmp|PngImage|*.Png"
    OpenFileDialog1.Title = "OpenPicture"
    OpenFileDialog1.InitialDirectory = Application.StartupPath
    If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
        PictureBox1.Load(OpenFileDialog1.FileName)
        ListBox1.Items.Add(OpenFileDialog1.FileName)
    End If
End Sub
```

5- اكتب الاجراء التالي في "Save"

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    SaveFileDialog1.Filter = "JpegImage|*.Jpg|BmpImage|*.Bmp|PngImage|*.Png"
    SaveFileDialog1.Title = "SavePicture"
    SaveFileDialog1.InitialDirectory = Application.StartupPath
    If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
        PictureBox1.Image.Save(SaveFileDialog1.FileName)
    End If
End Sub
```

6- انقر على صندوق القائمة واكتب التالي:

```
Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles ListBox1.SelectedIndexChanged
    PictureBox1.Load(ListBox1.SelectedItem)
End Sub
```

المشروع 19

رسالة بريد الكتروني

ستتقن من خلال المشروع

التعامل مع شريط التقدم ProgressBar

التعامل مع المؤقت Timer

التعامل مع Panel

التعامل مع Email

اهداف المشروع

- عند تنفيذ المشروع يظهر النموذج الأول ويتضمن شريط تحميل اخضر ما ان يصل الشريط الأخضر الى نهايته فيختفي النموذج الاول.

ويظهر النموذج الثاني والذي يتطلب إدخال بيانات خاصة برسالة بريد الكتروني (كـ بريد الشخص المستلم، عنوان الرسالة ونص الرسالة) بعد إتمام عملية ادخال البيانات في النموذج يتم النقر على زر Send لارسال الرسالة.

- كما يمكن للمستخدم تحريك النموذج بالنقر على الشريط الوردي في أعلى النموذج وسحبه الى اي اتجاه يرغب به.

مراحل المشروع

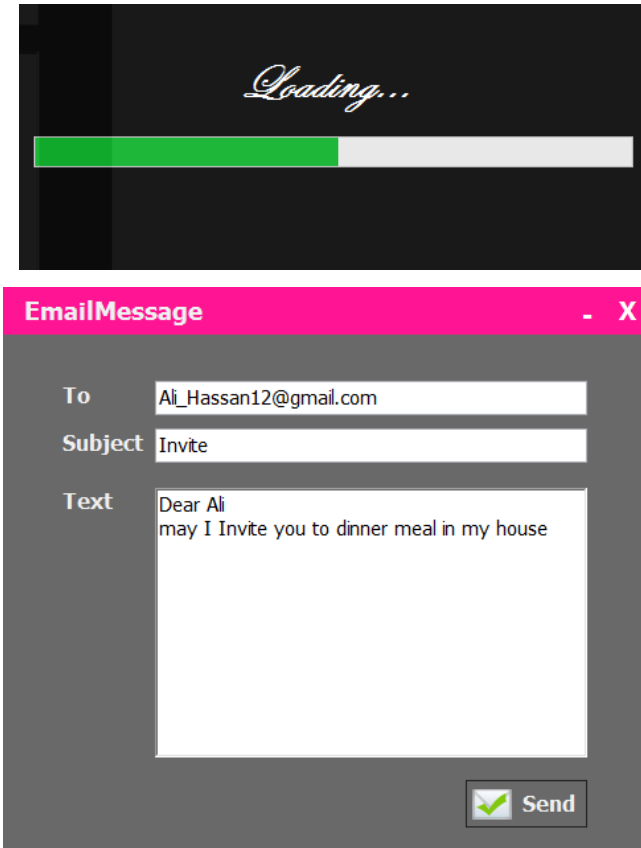
1- انشئ مشروعاً جديداً.

2- اصف للنموذج الاول Label بعنوان "Loading..." وشريط تقدم ProgressBar واضبط خصائص الشريط كالتالي:

Minimum	0
Maximum	100
Value	0

أهم خصائص الشريط هي:

Minimum	الحد الأدنى لقيمة شرط التقدم
Maximum	الحد الأعلى لقيمة شريط التقدم



Step	مقدار الزيادة في القيمة الحالية Value لشريط التقدم عند كل استدعاء
------	---

اضبط خصائص النموذج الأول كالتالي:

BackColor	Balck
FormBorderStyle	None
StartPosition	CenterScreen
Opacity	90%

3- اصف نموذج ثاني واجعل خصائصه كخصائص النموذج الأول واجعل لون الخلفية DimGray.

4- اصف عنصرين TextBox و RichTextBox و Label و زر أمر.

5- اصف Panel للنموذج الثاني واضبط خاصية Dock=Top و BackColor=DeepPink

ثم اصف Label باسم "X"، "-" و "EmailMessage" وضعهم على Panel.

6- اصف مؤقتاً Timer واضبط مجاله الزمني Interval على 50 ميلي ثانية.

المؤقت Timer يستخدم لتنفيذ اجراء معين كل فترة زمنية محددة.

الخاصية Interval هي المجال الزمني للمؤقت، أي الزمن الفاصل بين كل تنفيذ لإجراء المؤقت، حيث يقاس بالملي ثانية (اي عند وضع القيمة 1000 في المجال الزمني فهذا يعادل ثانية واحدة) اذا كان المجال صفراً فلن يعمل المؤقت.

7- اكتب الاجراء التالي داخل حدث تحميل النموذج الأول.

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Timer1.Start()
    Timer1.Interval = 50
End Sub
```

8- اكتب الاجراء التالي داخل حدث المؤقت.

التعليمة ProgressBar1.Increment(1) تعني زيادة Value لشريط التقدم بمقدار واحد في كل Tick للمؤقت.

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    ' زيادة قيمة شريط التقدم مع كل تكة للمؤقت
    ProgressBar1.Increment(1)
    '
    If ProgressBar1.Value = ProgressBar1.Maximum Then
        Me.Hide()
        Form2.Show()
        Timer1.Stop()
    End If
End Sub
```

التعليقات البرمجية للنموذج الثاني

9- اكتب التعليمة التالية في General

```
Imports System.Net.Mail
```



```
Dim drag As Boolean
Dim mousex As Integer
Dim mousey As Integer
```

الحدث MouseDown: يحصل هذا الحدث عندما ينقر المستخدم على أحد أزرار الفأرة.

ثم اكتب التعليمات البرمجية التالية في اجراء MouseDown للـ Panel

اسناد قيم للمتغيرات

Mousex الأحداث السيني للفأرة يساوي حاصل الفرق بين الاحداثي السيني لموقع مؤشر نوافذ النظام والمسافة بين الحافة اليسرى للنموذج مقاسه بالبكسل.

Mousey الأحداث الصادي للفأرة يساوي حاصل الفرق بين الاحداثي الصادي لموقع مؤشر نوافذ النظام والمسافة بين الحافة العليا للنموذج وتقاس بالبكسل ايضاً.

* الاجراء

```
Private Sub Panel1_MouseDown(sender As Object, e As MouseEventArgs) Handles
Panel1.MouseDown
    drag = True
    mousex = Windows.Forms.Cursor.Position.X - Me.Left
    mousey = Windows.Forms.Cursor.Position.Y - Me.Top
End Sub
```

10- اكتب الاجراء التالي في حدث MouseMove للـ Panel.

حدث MouseMove: يحصل هذا الحدث عندما يقوم المستخدم بتحريك الفأرة.

حافة النموذج العليا تساوي حاصل الفرق بين الاحداثي الصادي لنوافذ النظام والاحداثي الصادي للفأرة.

حافة النموذج اليسرى تساوي حاصل الفرق بين الاحداثي السيني لنوافذ النظام والاحداثي السيني للفأرة.

```
Private Sub Panel1_MouseMove(sender As Object, e As MouseEventArgs) Handles
Panel1.MouseMove
    If drag Then
        Me.Top = Windows.Forms.Cursor.Position.Y - mousey
        Me.Left = Windows.Forms.Cursor.Position.X - mousex
    End If
End Sub
```

11- اكتب التعليمة التالية في حدث MouseUp للـ Panel

حدث MouseUp: يحصل هذا الحدث عندما يحرر المستخدم زر الفأرة.

تعليمه تعطيل الاجراء السابق

```
Private Sub Panel1_MouseUp(sender As Object, e As MouseEventArgs) Handles
Panel1.MouseUp
    drag = False
End Sub
```

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

```
    If TextBox1.Text = "" Or TextBox2.Text = "" Or RichTextBox1.Text = "" Then
        Exit Sub
    End If
```

```
    ' متغير يمثل رسالة البريد الالكتروني
```

```
    Dim mymailmessage As New MailMessage
```

```
    Try
```

```
        ' الطرف الذي قام بارسال الرسالة
```

```
        mymailmessage.From = New MailAddress("hamzaj@gmail.com")
```

```
        ' الطرف الذي سيستلم الرسالة
```

```
        mymailmessage.To.Add(TextBox1.Text)
```

```
        ' عنوان الرسالة
```

```
        mymailmessage.Subject = (TextBox2.Text)
```

```
        ' يمثل نص الرسالة
```

```
        mymailmessage.Body = RichTextBox1.Text
```

```
        ' متغير من نوع بروتوكول يسمح بارسال رسالة بريدية
```

```
        Dim smtp As New SmtpClient("smtp.gmail.com")
```

```
        ' منفذ يستخدمه البروتوكول للنقل
```

```
        smtp.Port = 587
```

```
        ' امكانية تشفير الاتصال
```

```
        smtp.EnableSsl = True
```

```
        ' التصديق على تأهيل المرسل
```

```
        smtp.Credentials = New System.Net.NetworkCredential("hamzaj@gmail.com",
"j115160712")
```

```
        ' ارسال الرسالة عبر البروتوكول
```

```
        smtp.Send(mymailmessage)
```

```
        RichTextBox1.Text = ""
```

```
    Catch ex As Exception : End Try
```

```
End Sub
```

13- اكتب التعليمة التالية في "X"

```
Private Sub Label1_Click(sender As Object, e As EventArgs) Handles Label1.Click
```

```
    End
```

```
End Sub
```

14- اكتب التعليمة التالية في "-"

```
Private Sub Label6_Click(sender As Object, e As EventArgs) Handles Label6.Click
```

```
    WindowState = FormWindowState.Minimized
```

```
End Sub
```

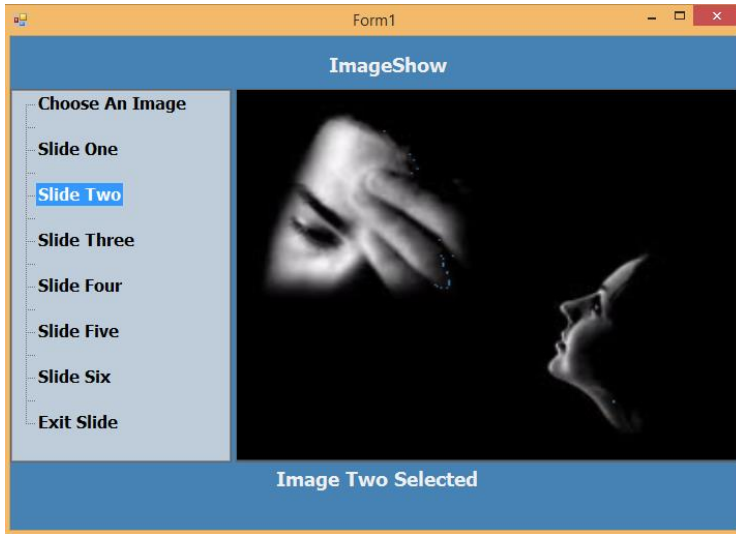
المشروع 20 عرض الصور (2)

ستتقن من خلال المشروع

- التعامل مع TreeView
- التعامل مع ImageList
- التعامل مع SplitContainer

اهداف المشروع

عند تنفيذ المشروع تظهر الواجهة التالية



- عند النقر على اي عقدة في القسم الأيسر من النافذة سيتم عرض صورتها في القسم الأيمن.

مراحل المشروع

1- انشئ مشروعاً جديداً.

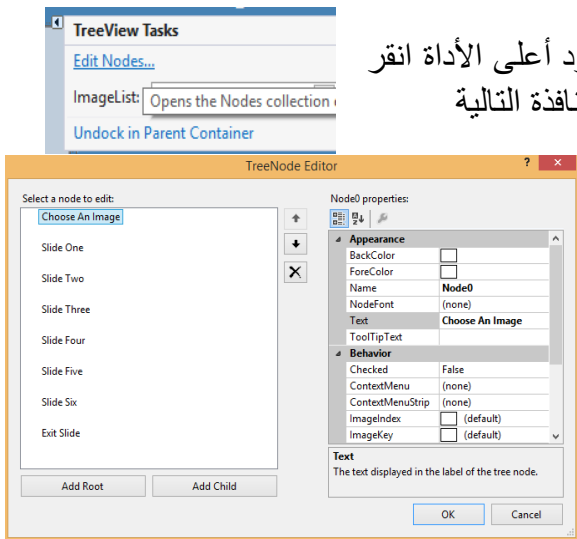
2- اضع Split Container للنموذج وغير ابعاده كما تظهر على النموذج.

3- اضع TreeView في القسم الأيسر من النموذج، ثم انقر على السهم الموجود أعلى الأداة انقر على الخيار الثالث في الصورة المجاورة بعدها انقر على Edit Node ستظهر النافذة التالية

وتتكون من جزئين.

الجزء الايسر ويضم ما يأتي:

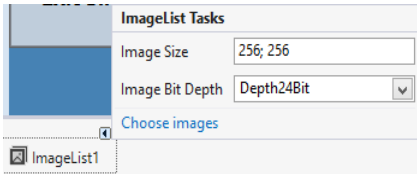
- قائمة تحتوي على عقد الأداة.
- زر Add Root يستخدم هذا الزر لإضافة عقدة جديدة.
- زر Add Child يستخدم لإضافة عقدة فرعية للعقدة الرئيسية.
- زر الحذف: يستخدم لحذف العقد المطلوبة.
- أزرار الاتجاهات لتحويل العقدة الرئيسية الى فرعية وبالعكس.



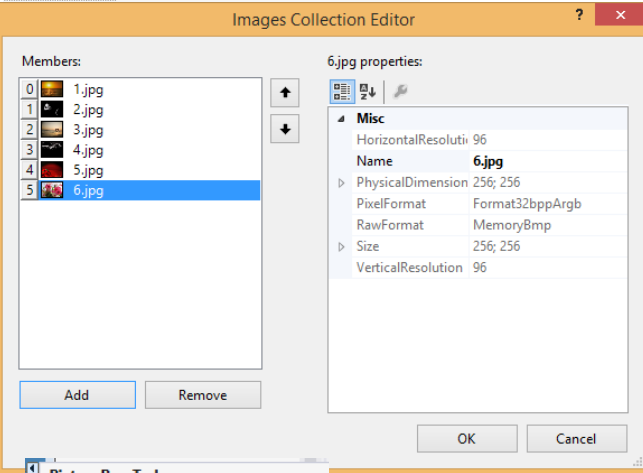
الجزء الأيمن ويضم خصائص كل عقدة في الشجرة حيث يمكن تغيير اسم العقد من خلال Text واختيار اسم برمجي لكل عقدة والتحكم بتغيير مظهر وسلوك العقد.

بعد اتمام التغييرات المطلوبة على العُقد انقر ok.

4- اضع أداة ImageList للنموذج وانقر على السهم الخاص بها واضبط الخصائص كما في الصورة، ثم اختر Chooseimages ستظهر النافذة ادناه وتتكون من جزئين الجزء الأيسر يحتوي على:



قائمة تضم الصور المضافة للأداة وبجانب كل صورة رقم يمثل تسلسلها في الأداة.



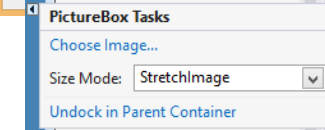
• زر Add لإضافة صورة للأداة.

• زر Remove لحذف الصورة المطلوبة من الأداة.

• أزرار الاتجاهات لتغيير تسلسل الصورة للأعلى أو الأسفل.

القسم الأيمن يحوي المعلومات الخاصة بكل صورة.

بعد اكمال اضافة الصورة انقر OK.



5- اضع PictureBox للنموذج وانقر على السهم الموجود أعلى العنصر واضبط خاصية SizeMode على StretchImage.

6- انقر على TreeView واكتب الاجراء التالي.

```
Private Sub TreeView1_AfterSelect(sender As Object, e As TreeViewEventArgs) Handles TreeView1.AfterSelect
```

```
    If e.Node.Name = "Node2" Then
        PictureBox1.Image = ImageList1.Images(0)
        Label2.Text = "Image One Selected"
    End If
    If e.Node.Name = "Node4" Then
        PictureBox1.Image = ImageList1.Images(1)
        Label2.Text = "Image Two Selected"
    End If
    If e.Node.Name = "Node6" Then
        PictureBox1.Image = ImageList1.Images(2)
        Label2.Text = "Image Three Selected"
    End If
    If e.Node.Name = "Node8" Then
        PictureBox1.Image = ImageList1.Images(3)
        Label2.Text = "Image Four Selected"
    End If
    If e.Node.Name = "Node10" Then
        PictureBox1.Image = ImageList1.Images(4)
        Label2.Text = "Image Five Selected"
    End If
    If e.Node.Name = "Node12" Then
        PictureBox1.Image = ImageList1.Images(5)
        Label2.Text = "Image Six Selected"
    End If
    If e.Node.Name = "Node14" Then
        PictureBox1.Image = Nothing
    End If
```

المشروع 21

التعامل مع الملفات (2)

(LogIn)

ستتقن من خلال المشروع

- التعامل مع الملفات
- التعامل مع دوال الحوار
- المتحولات العامة

اهداف المشروع

عند تنفيذ المشروع تظهر نافذة تطلب ادخال UserName و Password :

- اذا لم يُدخل المستخدم Username و Password ونقر على زر Login

- سيظهر صندوق رسالة يطلب اضافة شخص جديد عند النقر على OK من الصندوق الحواري سيتحول زر Login الى Register.

- بعد ادخال Username و Password ثم النقر على زر Register.
- سيظهر صندوق رسالة يؤكد نجاح عملية الإدخال.

- اما اذا ادخل المستخدم Username و Password بصورة صحيحة ونقر على زر LogIn سيظهر صندوق حوار يؤكد صحة كلمة السر. بالنقر على زر Ok في صندوق الحوار يختفي النموذج الاول ويظهر النموذج الثاني.
- لتغيير كلمة السر، يجب ان يُدخَل المستخدم كلمة السر القديمة بشكلها الصحيح في صندوق النص الأول، ثم يضع كلمة السر الجديدة في صندوق النص الثاني ويُعيد كتابتها في صندوق النص الثالث ولضمان نجاح عملية تغيير كلمة السر يجب ان تكون الكلمتان في الصندوق الثاني والثالث متطابقتين ثم ينقر على زر ChangePassWord.

التعامل مع الملفات التسلسلية برمجياً

الملفات التسلسلية(Sequential): تُخزن البيانات في الملف التسلسلي على شكل سلسلة نصية، يُفتح الملف التسلسلي إما للقراءة او للكتابة أي لا يمكن القراءة والكتابة بأن واحد، كما يتم الوصول للبيانات بشكل متسلسل.

- فتح ملف تسلسلي للكتابة فيه.

حيث يتم انشاء الملف ان لم يكن موجود، واذا كان موجود يتم فتحه للكتابة.

FileOpen(Filenumber ,Pathname ,OpenMode.Append)

حيث:

- FileNumber هو رقم بين (1 و 511) نُعطيه للملف عند فتحه، ثم نتعامل مع الملف عن طريق هذا الرقم.
- Pathname هو اسم الملف ويمكن ان يحوي على مساره الكامل.

- تعليمة الكتابة في الملف بعد فتحه للكتابة.

WriteLine(Filenumber, Outputlist)

- Outputlist هو مجموعة التعابير التي تُريد كتابتها في الملف.

- تعليمة الطباعة في ملف.

PrintLine(Filenumber , Outputlist)

- تعليمة غلق ملف.

FileClose(Filenumber)

(تكتب او تطبع البيانات في الملف عادة باستخدام Write أو Print وبالتعليمة Append)

مثل:

```
FileOpen(1,myfile,OpenMode.Append)
```

```
WriteLine(1,"This is my Note")
```

FileClose(1)

عند فتح الملف باستخدام NotePad سنجد العبارة ضمن محتوى الملف.

- تعليمية فتح ملف للقراءة منه.

FileOpen(Filenumber, Pathname ,**OpenMode.Input**)

- تعليمية القراءة بيانات من ملف.

LineInput(Filenumber)

خلال هذه التعليمية سيتم قراءة سطر واحد من ملف.

- اختبار الوصول الى نهاية الملف.

EOF(Filenumber)

حيث تُعيد هذه الدالة True اذا وصل مؤشر القراءة الى نهاية الملف، وإلا فإنها تُعيد False.

في المثال التالي يتم قراءة محتويات الملف حتى النهاية واضافتها الى ListBox

FileOpen(1,myfile,**OpenMode.Input**)

Do While Not EOF(1)

X=LineInput(1)

ListBox1.Items.add(x)

Loop

FileClose(1)

مراحل المشروع

1- انشئ مشروعاً جديداً.

2- اضبط خصائص النموذج الأول كما في الجدول ادناه

BackColor	DimGray
FormBorderStyle	None
StartPosition	CenterScreen

3- اضف Panel للنموذج الأول واضبط خصائصه كالتالي:

BackColor	64؛ 64؛ 64
Dock	Top

4- اضف ثلاثة Label للنموذج واضبط الخصائص كالتالي:

Label1		Label2		Label3	
BackColor	64; 64; 64	BackColor	64; 64; 64	BackColor	64; 64; 64

Font	Times New Roman; 15.75pt; style=Bold	Font	Times New Roman; 15.75pt; style=Bold	Font	Times New Roman; 15.75pt; style=Bold
ForeColor	ButtonFace	ForeColor	ButtonFace	ForeColor	ButtonFace
Text	Login	Text	-	Text	X

5- اضع اربعة TextBox واضبط الخصائص كالتالي:

TextBox1		TextBox2		TextBox3		TextBox4	
Name	Txttempuser	Name	Txttemp	Name	Txtusername	Name	Txtpassword
BackColor	DimGray	BackColor	DimGray	BackColor	DimGray	BackColor	DimGray
ForeColor	DarkGray	ForeColor	DarkGray	ForeColor	DarkGray	ForeColor	DarkGray
Text	UserName	Text	PassWord	Text		Text	

ثم ضع TextBox1 على TextBox3 و TextBox2 على TextBox4.

6- اضع ثلاثة PictureBox واضبطها كما تظهر على النموذج السابق. ثم اضع وحدة نمطية Module للبرنامج.

7- اضع زرین أمر واجعل Text="Login" والثاني Text="Register" ثم ضع الاول فوق الثاني.

8- اضع العناصر كما تظهر على النموذج الثاني.

9- اكتب الاجراءات التالية الخاصة بالـ Panel والتي تسمح بنقل النموذج حسب رغبة المستخدم في قسم تصريحات النموذج.

```

Dim drag As Boolean
Dim mousex As Integer
Dim mousey As Integer

Private Sub Panel1_MouseDown(sender As Object, e As MouseEventArgs) Handles Panel1.MouseDown
    drag = True
    mousex = Windows.Forms.Cursor.Position.X - Me.Left
    mousey = Windows.Forms.Cursor.Position.Y - Me.Top
End Sub

Private Sub Panel1_MouseMove(sender As Object, e As MouseEventArgs) Handles Panel1.MouseMove
    If drag Then
        Me.Top = Windows.Forms.Cursor.Position.Y - mousey
        Me.Left = Windows.Forms.Cursor.Position.X - mousex
    End If
End Sub

Private Sub Panel1_MouseUp(sender As Object, e As MouseEventArgs) Handles Panel1.MouseUp

```



```

drag = False
End Sub

```

10- اكتب الاجراء التالي على زر تحميل النموذج.

```

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ' اخفاء زر Register
    btnregister.Visible = False
    ' اخفاء صندوق نص اسم المستخدم
    Txtusername.Visible = False
    ' اخفاء صندوق كلمة السر
    Txbpassword.Visible = False

End Sub

```

11- برمجة زر الامر "Login"

```

Private Sub btnlogin_Click(sender As Object, e As EventArgs) Handles btnlogin.Click
    Dim checker, s As String
    If Txtusername.Text = "" And Txbpassword.Text = "" Then
        checker = MsgBox("Do you want add new person?", MsgBoxStyle.Information +
MsgBorderStyle.YesNo, "Query")

        If checker = vbYes Then
            btnlogin.Visible = False
            Txttemp.Visible = False
            Txttempuser.Visible = False
            Txtusername.Visible = True
            Txbpassword.Visible = True
            Txtusername.Focus()
            btnregister.Visible = True
        End If
    ElseIf Txbpassword.Text = "" Then
        MsgBox("Check Username or Password")
    Else
        ' وضع الملف بنفس مسار البرنامج
        myfile = Application.StartupPath + "\main.txt"
        ' فتح الملف للقراءة
        FileOpen(1, myfile, OpenMode.Input)
        ' قراءة بيانات من الملف
        Do While Not EOF(1)
            ' وضع سطر في متغير
            s = LineInput(1)
            ' تدقيق كلمة السر المدخلة مع البيانات المقروءة من الملف
            If Txbpassword.Text = s Then
                MsgBox("Correct PassWord", , "")
                Me.Hide()
                ' وضع كلمة السر في متغير عام
                pass1 = s
                Form2.Show()
            Exit Do
        Exit Sub
    End If

```

```

        End If
    Loop
    'غلق الملف'
    FileClose(1)

```

```

End If
End Sub

```

12- اكتب الاجراء التالي في حدث النقر لصندوق النص الأول بمعنى ان التعليمات البرمجية ستُنفذ عند النقر على الصندوق بالفأرة. في هذا الاجراء عند النقر على TextBox1 سيختفي ويظهر TextBox3 لكتابة اسم المستخدم فيه.

```

Private Sub Txttempuser_Click(sender As Object, e As EventArgs) Handles
Txttempuser.Click
    Txttempuser.Visible = False
    Txtusername.Visible = True
    Txtusername.Focus()
End Sub

```

13- اكتب الاجراء التالي في حدث النقر لصندوق النص الثاني

في هذا الاجراء عند النقر على TextBox2 سيختفي ويظهر TextBox4 لكتابة كلمة السر فيه.

```

Private Sub Txttemp_Click(sender As Object, e As EventArgs) Handles Txttemp.Click
    Txttemp.Visible = False
    Txtpassword.Visible = True
    Txtpassword.Focus()
End Sub

```

15- برمجة زر "Register"

```

Private Sub btnregister_Click(sender As Object, e As EventArgs) Handles
btnregister.Click
    Dim x As String
    If Txtusername.Text <> "" And Txtpassword.Text <> "" Then
        myfile = Application.StartupPath + "\main.txt"
        'فتح الملف للكتابة'
        FileOpen(1, myfile, OpenMode.Append)
        'كتابة كلمة السر في الملف'
        WriteLine(1, Txtpassword.Text)
        'غلق الملف'
        FileClose(1)
        Txtusername.Text = ""
        Txtpassword.Text = ""
        x = MsgBox("add data successfully!", MsgBoxStyle.YesNo +
MsgBoxStyle.Information, "Adding")

        If x = vbYes Then
            btnlogin.Visible = True
            btnregister.Visible = False
            Txtusername.Focus()
        End If
    Else

```

```

        MsgBox("Enter UserName or Password.", MsgBoxStyle.Exclamation, "ERROR")
    End If
End Sub

```

"X" Label برمجة -16

```

Private Sub Label2_Click(sender As Object, e As EventArgs) Handles Label2.Click
    End
End Sub

```

"-" Label برمجة -17

```

Private Sub Label4_Click(sender As Object, e As EventArgs) Handles Label4.Click
    WindowState = FormWindowState.Minimized
End Sub

```

"ChangePassWord" زر برمجة -18

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    'مقارنة كلمة السر القديمة مع المُدخلة في صندوق النص الاول'
    If pass1 <> Txtoldpass.Text Then
        MsgBox("Incorrect old password", , "Error")
    'مقارنة كلمة السر الجديدة في صندوق النص الثاني والثالث'
    ElseIf Txtnewpass.Text <> Txtconpass.Text Then
        MsgBox("new pasword don't matched", , "Error")
    Else
        'وضع اسم ومسار الملف في المتغير'
        myfile = Application.StartupPath + "\main.txt"
        'فتح الملف للكتابة'
        FileOpen(1, myfile, OpenMode.Append)
        'كتابة كلمة السر الجديدة في الملف'
        WriteLine(1, Txtnewpass.Text)
        FileClose(1)
    End If
End Sub

```

"Cancel" زر برمجة -19

```

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

```

-20 ضع التعليمة التالية في الوحدة النمطية

```

Module Module1
    Public pass1, myfile As String
End Module

```

المشروع 22

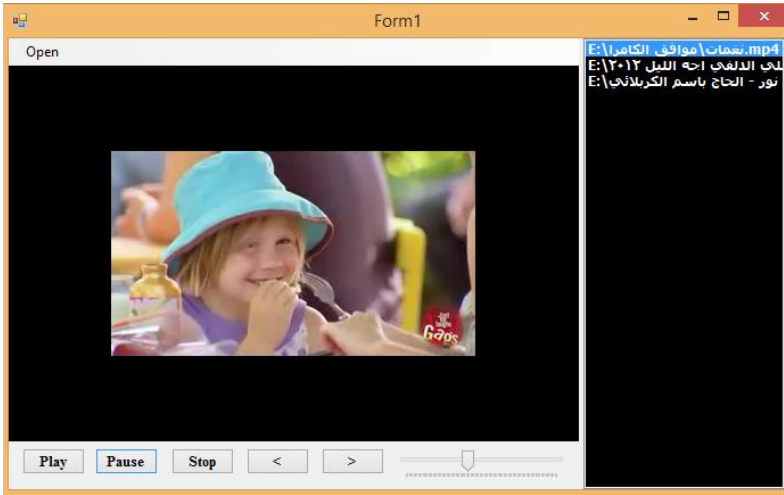
تشغيل ملفات صوت وفيديو

ستتقن من خلال المشروع

- التعامل مع WindowsMediaPlayer
- التعامل مع OpenFileDialog
- التعامل مع TrackBar
- التعامل مع SplitContainer

اهداف المشروع

عند تنفيذ المشروع تظهر النافذة التالية:



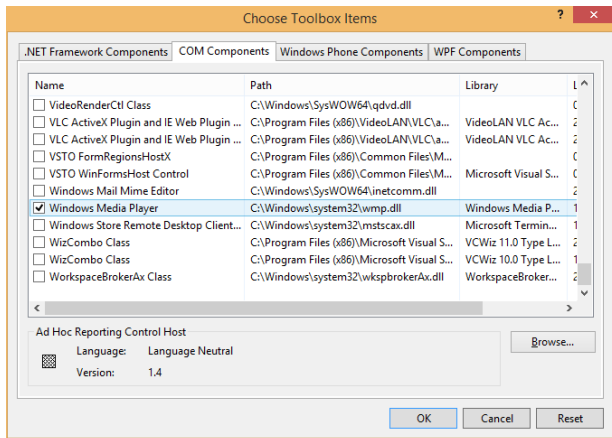
- عند النقر على Open تظهر نافذة يتم من خلالها اختيار الملف المطلوب تشغيله.
- كل ملف يقوم المستخدم باختياره يضاف الى القائمة بالجانب الايمن من النافذة، ويتم تشغيله بالجانب الأيسر منها.
- الأزرار الموجودة أسفل الفيديو تسمح للتحكم بالفيديو كتشغيل، إيقاف، تسريع وإبطاء الفيديو كما يمكن التحكم بمستوى صوت الفيديو.

مراحل المشروع

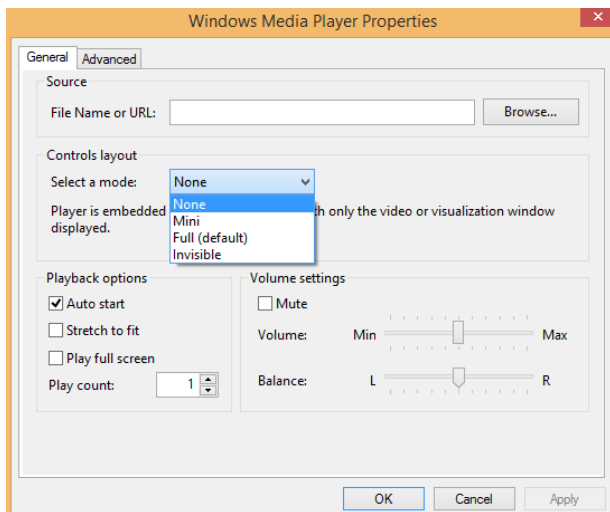
1- انشئ مشروعاً جديداً.

2- اصف الأدوات الظاهرة على النموذج.

ملاحظة عنصر WindowsMediaPlayer غير موجود ضمن شريط الأدوات Toolbox ولاضافته لها انقر بزر الفأرة الايمن



على شريط الادوات واختر Choose Items.. من القائمة المنبثقة، ستظهر نافذة تحتوي على عدة تبويبات اختر تبويب Com Component اختر الاداة بالنقر داخل المربع الجانبي لها ،لتظهر علامة صح داخله ثم انقر OK.



اسحب العنصر من شريط الادوات الى النافذة ثم انقر بزر الفأرة الايمن واختر Properties من القائمة المختصرة للاداة. ومن نافذة الخصائص اجعل خيار Select A mode = None لاختفاء شريط التحكم الخاص بالاداة.

3- برمجة "Open"

```
Private Sub OpenToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
OpenToolStripMenuItem.Click
    OpenFileDialog1.Filter = "AllFile|*.*"
    OpenFileDialog1.ShowDialog()
    وضع الملف في الاداة لتشغيله
    AxWindowsMediaPlayer1.URL = OpenFileDialog1.FileName
    اضافة الملف الى القائمة
    ListBox1.Items.Add(OpenFileDialog1.FileName)
End Sub
```

4- برمجة زر "Play"

```
Private Sub btnplay_Click(sender As Object, e As EventArgs) Handles btnplay.Click
    AxWindowsMediaPlayer1.Ctlcontrols.play()
End Sub
```

5- برمجة زر "Pause"

```
Private Sub btnpause_Click(sender As Object, e As EventArgs) Handles
btnpause.Click
    AxWindowsMediaPlayer1.Ctlcontrols.pause()
End Sub
```

6- برمجة زر "Stop"

```
Private Sub btnstop_Click(sender As Object, e As EventArgs) Handles btnstop.Click
    AxWindowsMediaPlayer1.Ctlcontrols.stop()
End Sub
```

7- برمجة زر "<"

```
Private Sub btnback_Click(sender As Object, e As EventArgs) Handles btnback.Click
    AxWindowsMediaPlayer1.Ctlcontrols.fastReverse()
End Sub
```

8- برمجة زر ">"

```
Private Sub btnforward_Click(sender As Object, e As EventArgs) Handles  
btnforward.Click  
    AxWindowsMediaPlayer1.Ctlcontrols.fastForward()  
End Sub
```

9- انقر على صندوق القائمة واكتب الكود التالي، الخاص بتشغيل اي ملف يتم اختياره منها.

```
Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs)  
Handles ListBox1.SelectedIndexChanged  
    AxWindowsMediaPlayer1.URL = ListBox1.SelectedItem  
End Sub
```

10- ربط شريط المسار بحجم الصوت الخاص بالـ WindowsMeidaPlayer من خلال الاجراء التالي:

```
Private Sub TrackBar1_Scroll(sender As Object, e As EventArgs) Handles TrackBar1.Scroll  
    AxWindowsMediaPlayer1.settings.volume = TrackBar1.Value  
End Sub
```

المشروع 23

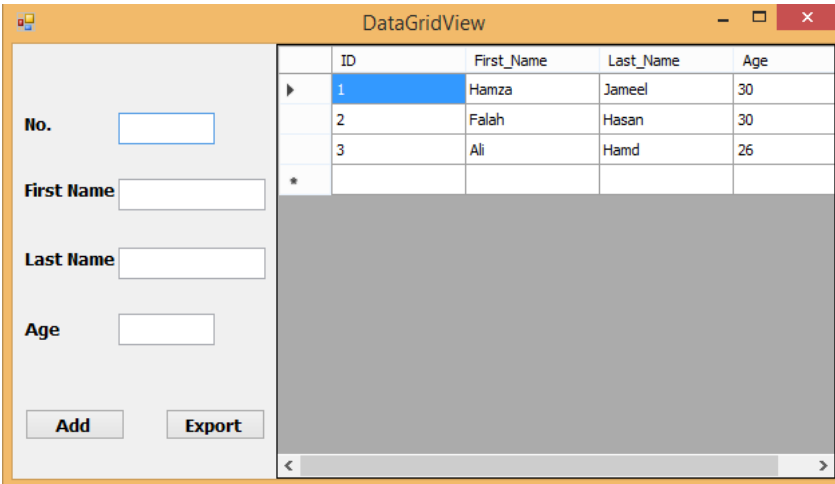
الشبكة DataGridView

ستتقن من خلال المشروع

- التعامل مع الشبكة DataGridView
- الحلقات المتداخلة
- الملفات النصية

اهداف المشروع

عند تنفيذ المشروع يظهر النموذج التالي:



يتم إدخال بيانات الى الشبكة من خلال مجموعة صناديق النص الموجودة في القسم الأيسر من النافذة ويستخدم زر "Add" لإضافة البيانات الجديدة الى الشبكة بينما يستخدم زر "Export" لحفظ بيانات الشبكة في ملف نصي بنفس مسار البرنامج.

تصميم الشبكة DataGridView

تعتبر الشبكة أحد أهم أدوات التحكم الأساسية في الفيجوال بيسك وهي كثيرة الاستعمال في التطبيقات التي تتطلب اظهار جداول. حيث يمكن ان تكون هذه الجداول إدخالات او ناتج معالجة او محتويات جداول من قاعدة البيانات.

كما ان طريقة اضافتها للنموذج لا تختلف عن اضافة بقية الادوات.

اهم خصائص الشبكة

يمثل عدد أعمدة الشبكة	Columns
يمثل عدد صفوف الشبكة	Rows
تحديد اكثر من صف او عمود	MultiSelect
تسمية عناوين عمود رقم n في الشبكة	Columns(n).HeaderText
تسمية العمود رقم n في الشبكة	Columns(n).Name
لون خلفية الشبكة	BackgroundColor
تحديد عرض العمود رقم n	Columns(n).Width
تحديد ارتفاع العمود رقم n	Rows(n).Height
وضع قيمة في عنوان الخلية التابعة للصف n	Rows(n).HeaderCell.Value

تحديث بيانات الشبكة بعد إجراء عمليات الحذف أو الإضافة أو التعديل	Refresh
عدد الصف الذي وقع عليه الاختيار من الشبكة	SelectedRows.Count
محتوى الصف رقم n	SelectedRows(n).Index
حذف صف من الشبكة	Rows.RemoveAt
جعل الشبكة للقراءة (اي لا يمكن اضافة بيانات للخلايا من خلال النقر عليها)	ReadOnly
عدد صفوف الشبكة	RowCount
عدد اعمدة الشبكة	ColumnCount
يستخدم لتغيير لون صفوف الشبكة	RowsDefaultCellStyle.BackColor

أهم احداث الشبكة

النقر على خلية	CellClick
الضغط على حرف من الكيبورد	KeyPress
انتقال التحديد الى الخلية	CellEnter
الخروج من خلية	CellLeave

تسمية رؤس اعمدة الشبكة

هنالك عدة طرق لتسمية رؤس الأعمدة عن طريق الكود في الفيجوال بيسك وابتسط هذه الطرق هي:

```
DataGridView1.ColumnCount = 3
DataGridView1.Columns(0).Name = "No"
DataGridView1.Columns(1).Name = "Year"
DataGridView1.Columns(2).Name = "Team"
```

```
DataGridView1.Rows.Add("1", "2017", "Chelsea")
DataGridView1.Rows.Add("2", "2017", "Tottenham")
```

	No	Year	Team
▶	1	2017	Chelsea
	2	2017	Tottenham
*			

حيث ان ترقيم الاعمدة والصفوف يبدأ من الصفر.

فيكون الناتج منا في الشكل المجاور.

يمكن التعامل مع الخلية واعطائها قيمة داخل شبكة من خلال الكود التالي:

```
DataGridView1.Rows(2).Cells(2).Value = "Liverpool"
```

	No	Year	Team
▶	1	2017	Chelsea
	2	2017	Tottenham
*			Liverpool

فيكون الناتج كما يظهر في الشكل المجاور.

مراحل المشروع

1- انشئ مشروعاً جديداً.

2- اصف الأدوات الظاهرة على النموذج.

3- ضع التعليمة التالية في قسم التصريحات العامة للمشروع.

Imports System.IO

4- اكتب التعليمة التالية في Class Form1

```
Dim table As New DataTable("table")
```

يتضمن الكود اعلاه تعريف متغير من نوع DataTable والذي يمثل جدول وهمي في ذاكرة الحاسوب.

والجدول عبارة عن خلايا موزعة بشكل افقي(صفوف) وعمودي(أعمدة).

5- اكتب الاجراء التالي في حدث تحميل النموذج.

حيث يتضمن انشاء تسميات لأعمدة الشبكة مع تحديد النوع البياني الخاص بها.

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load  
    'Add Column to table
```

```
    table.Columns.Add("ID", Type.GetType("System.Int32"))  
    table.Columns.Add("First_Name", Type.GetType("System.String"))  
    table.Columns.Add("Last_Name", Type.GetType("System.String"))  
    table.Columns.Add("Age", Type.GetType("System.Int32"))
```

مصدر بيانات الشبكة الجدول اعلاه،

```
    DataGridView1.DataSource = table
```

تحديث بيانات الشبكة،

```
    DataGridView1.Refresh()
```

جعل خلايا الشبكة قابلة للقراءة فقط بحيث لا يمكن الكتابة بالنقر عليها،

```
    DataGridView1.ReadOnly = True
```

```
End Sub
```

6- برمجة زر "Add" والمتضمن اضافة بيانات المُدخلة في صناديق النص إلى الشبكة.

```
Private Sub btnadd_Click(sender As Object, e As EventArgs) Handles btnadd.Click  
    table.Rows.Add(txtno.Text, txtfname.Text, txtlname.Text, txtage.Text)
```

```
    DataGridView1.Refresh()
```

```
    txtno.Text = ""
```

```
    txtfname.Text = ""
```

```
    txtlname.Text = ""
```

```
    txtage.Text = ""
```

```
    txtno.Focus()
```

```
End Sub
```

7- برمجة زر "Export" لتصدير بيانات الشبكة إلى ملف نصي.

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

```
    Dim myfile As String
```

وضع ملف بنفس مسار البرنامج،

```
    myfile = Application.StartupPath + "\file.txt"
```

فتح الملف للكتابة،

```
    Dim writefile As StreamWriter = New StreamWriter(myfile)
```

استخدام حلقات تكرارية للمرور على اعمدة و صفوف الشبكة،

```

For i As Integer = 0 To DataGridView1.Rows.Count - 2
  For j As Integer = 0 To DataGridView1.Columns.Count - 1
    طباعة قيمة خلايا الشبكة في الملف النصي
    writefile.Write(vbTab &
DataGridView1.Rows(i).Cells(j).Value.ToString() & vbTab & "|")
  Next
  writefile.WriteLine("")
  writefile.WriteLine("_____")
Next
غلق الملف النصي
writefile.Close()
اظهار رسالة لتأكيد تصدير البيانات
MessageBox.Show("Exported data done")
End Sub

```

المشروع 24 الشبكة (التعامل مع Excel)

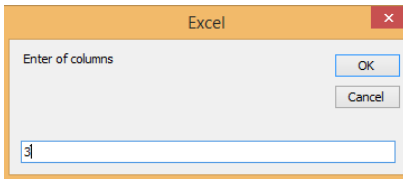
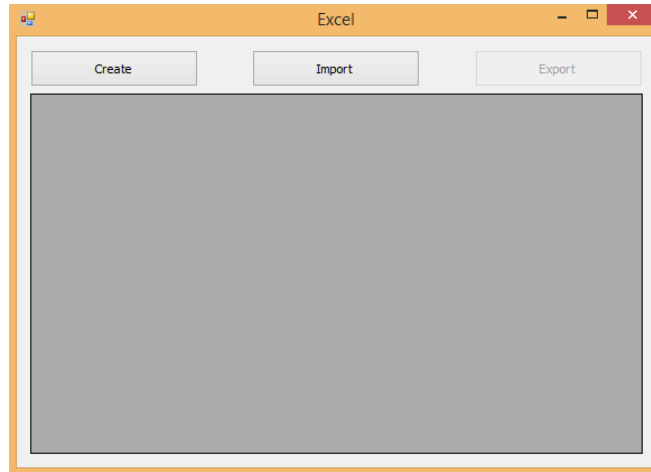
ستتقن من خلال المشروع

- التعامل مع الشبكة DataGridView
- التعامل مع OpenFileDialog
- التعامل مع SaveFileDialog

اهداف المشروع

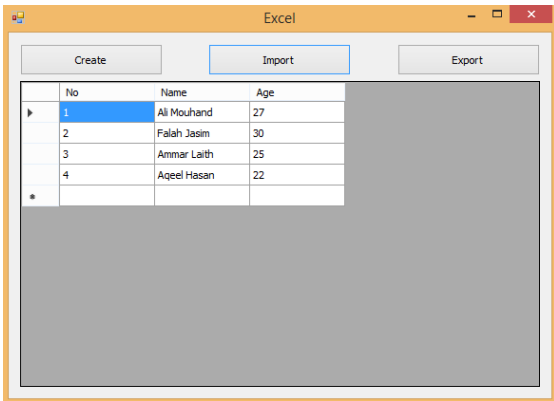
عند تنفيذ المشروع تظهر الواجهة التالية:

تحتوي على زرین مفعلين وزر ثالث غير مفعول لكل منهم عمل يؤديه كما يلي:



- عند النقر على زر "Create" يظهر صندوق حوار يطلب ادخال رقم يمثل عدد أعمدة الشبكة، ثم تظهر صناديق حوار يتم من خلال إدخال أسماء لرؤوس أعمدة الشبكة.

- عند النقر على زر "Import" تظهر نافذة تسمح لك بتحديد مكان وجود ملف Excel لفتحه.



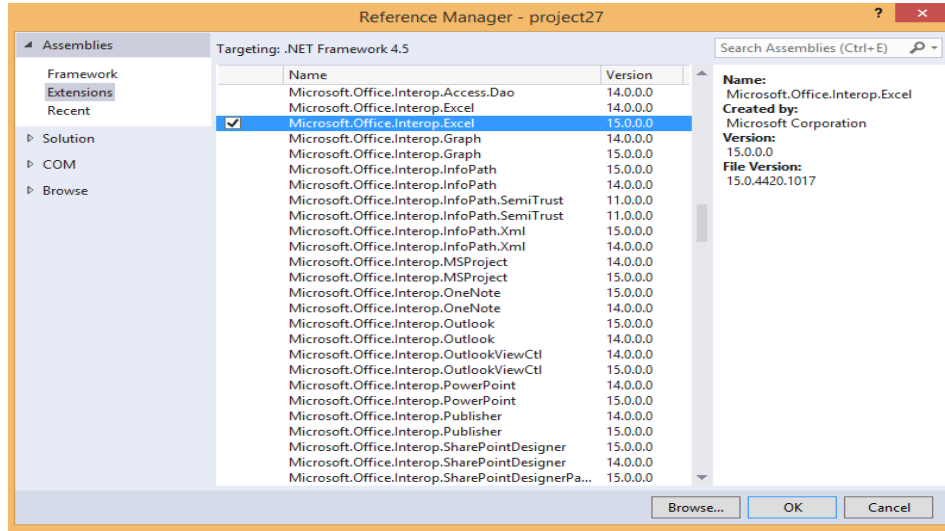
- يمكن التعديل على البيانات المعروضة في الشبكة وإضافة وحذف منها كما انه عند النقر على اي من الزرين السابقين يتم تفعيل زر Export.
- عند النقر على زر "Export" تظهر نافذة لتحديد مكان حفظ الملف باسم من اختيار المستخدم وبنوعين 2003 أو 2007.

مراحل المشروع

1- انشئ مشروعاً جديداً.

2- اصف الأدوات الظاهرة على النموذج.

3- للتعامل مع تطبيق الاكسل يجب استخدام دالة الاكسل من خلال Project → Add Reference.. ستظهر الواجهة التالية نختار Extensions ثم اختر الاداة Microsoft.Office.Interop.Excel.



برمجة الاتصال بتطبيق الاكسل

في الفيچوال بيسك يمكن كتابة اجراءات تقوم بارسال تعليمات الى تطبيق وتتحكم بالكائنات التي يمكن ان ينشئها هذا التطبيق، مثلاً يمكن كتابة اجراء يقوم بفتح صفحة Excel وارسال بيانات اليها.

قبل ان تستخدم تقنية الاتصال بالتطبيقات، عليك ان تدرس بعناية كائنات التطبيق الذي تريد ان تستخدمه وتتعلم خصائصه والاحداث التي تريد التحكم بها.

هناك خطوات يمكن من خلالها التحكم في كائن التطبيق كالتالي:

- انشاء متغير من النوع كائن لكل كائن التطبيق.
- انشاء تنفيذ instance من كائن التطبيق وجعل المتغير يؤشر عليه.
- التحكم بالكائن من خلال ضبط خصائصه او استدعاء طرقه.
- إغلاق (تحرير) الكائن عند الانتهاء.

برمجة New لإنشاء كائن

يمكن استخدام New عند التصريح عن متغير كائن لإنشاء تنفيذ instance. كما يمر في الخطوات التالية من هذا المشروع.

برمجة تحرير كائن

بعد استدعاء الكائن فان اجراءك يكون هو المسؤول عن فتح واغلاق الكائنات، اي عندما تنتهي من العمل مع كائن عليك إغلاق الكائن والخروج من التطبيق. يتم تحرير الكائن من خلال التعليمة التالية:

Objvar=Nothing

فان هذه التعليمة تقوم بقطع الاتصال الا انها في معظم التطبيقات لا تقوم بإغلاق التطبيق. فلإغلاق التطبيق عليك استخدام Close او Exit حسب نوع التطبيق.

اما اذا كنت تعمل على تطبيق مثل Excel فانه عليك اولاً غلق التطبيق من خلال Quit ومن ثم تحرير المتحول، كما يلي:

apExcel.Quit

apExcel=nothing

كما لا يخفى على الجميع ان Excel يتكون من مصنف (الـ ملف الذي يتم انشاءه بتطبيق Excel) ويدعى Workbook وكل مصنف يتكون من ورقة عمل Worksheet التي بدورها تتكون من خلايا Cells مرتبة على شكل افقي وعمودي ويتم التعامل مع هذه الكائنات من خلال التصريح عن متغيرات من نوع كائنات يتم فتحها وبعد اتمام العمل عليها لا بد من غلقها. كما سيمر لاحقاً

4- في قسم التصريحات العامة اكتب التعليمة التالية:

```
Imports Microsoft.Office.Interop.Excel
```

تكمين اهمية هذه المكتبة عند الاتصال بالاكسل وتصدير البيانات اليه.

5- ان عملية الاتصال بالاكسل وجلب البيانات منه تتطلب عدة امور:

- OleDbConnection حيث يقوم هذا الكائن بالاتصال بمصدر البيانات.
- DataSet جدول وهمي لـ تخزين البيانات فيه.
- DataAdapter لجلب البيانات من مصدرها.

تتوفر الكائنات السابقة في التعليمة التالية:

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

لذا يجب وضع التعليمة السابقة في قسم التصريحات العامة للبرنامج

صرح عن الأدوات التالية في قسم التصريحات الخاصة Class Form

```
' الملف لفتح متغير عن الاعلان '
Dim ofd As New OpenFileDialog
' الملف لحفظ متغير عن الاعلان '
Dim sfd As New SaveFileDialog
```

في حدث تحميل النموذج ضع التعليمة التالية:

لاننا لا نرغب بتصدير بيانات فارغة الى مصنف الاكسل ولهذا نعمل على الغاء تفعيل زر Export.

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ' زر تفعيل الغاء '
    BtnExport.Enabled = False
End Sub
```

6- ضع الإجراء التالي في زر "Create".

يعمل هذا الإجراء على افراغ بيانات الشبكة وكذلك يسمح بتحديد عدد أعمدتها وتسمية الأعمدة خلال التنفيذ ليتسنى إدخال بيانات جديدة.

```
Private Sub BtnCreate_Click(sender As Object, e As EventArgs) Handles
BtnCreate.Click
    DataGridView1.DataSource = Nothing
    ' الشبكة اعمدة عدد ادخال '
    Dim x As Integer = InputBox("Enter number of columns")
    ' الشبكة اعمدة عدد تحديد '
    DataGridView1.ColumnCount = x
    ' الشبكة لاعمدة عناوين اختيار '
    For i As Integer = 0 To x - 1
        DataGridView1.Columns(i).HeaderText = InputBox("Enter name of column " & i)
```

```

Next
' زر تفعيل Export
BtnExport.Enabled = True
End Sub

```

7- في زر "Import" ضع الاجراء التالي:

يتيح لك هذا الاجراء اختيار ملف الاكسل وعرضه في DataGridView.

```

Private Sub BtnImport_Click(sender As Object, e As EventArgs) Handles
BtnImport.Click
' فتحها المطلوب الملفات نوع تحديد
ofd.Filter = "Excel 2003|*.xls|Excel 2007|*.xlsx"
If ofd.ShowDialog = System.Windows.Forms.DialogResult.OK Then
' البيانات من الشبكة افراغ
DataGridView1.DataSource = Nothing
' الشبكة اعمدة حذف
DataGridView1.Columns.Clear()
' متغير في الملف ومسار اسم وضع
Dim path As String = ofd.FileName
' الملف لفتح الاكسل ببرنامج الاتصال
Dim con As New
System.Data.OleDb.OleDbConnection("Provider=Microsoft.Ace.OLEDB.12.0;Data Source=" &
path & ";Extended Properties=Excel 12.0;")
' الجدول لخزن متغير عن الاعلان
Dim ds As New DataSet
' الجدول من بيانات سحب
Dim da As New OleDbDataAdapter("Select * From [Sheet1$]", con)
' الوهمي الجدول اداة في البيانات وضع
da.Fill(ds, "[Sheet1$]")
' الشبكة في الجدول بيانات عرض
DataGridView1.DataSource = ds.Tables("[Sheet1$]")
' زر تفعيل Export
BtnExport.Visible = True
Else
Exit Sub
End If
End Sub

```

8- برمجة زر "Export"

في هذا الإجراء يتم انشاء مصنف يحتوي على ورقة عمل واحدة ثم العمل على خزن بيانات الشبكة فيها.

```

Private Sub BtnExport_Click_1(sender As Object, e As EventArgs) Handles
BtnExport.Click
' en-US لغة الى جهازك لغة تحويل
System.Threading.Thread.CurrentThread.CurrentCulture =
System.Globalization.CultureInfo.CreateSpecificCulture("en-US")

' تطبيق متغير عن الاعلان
Dim ExcelApp As New Application
' مصنف متغير عن الاعلان
Dim ExcelWorkbook As Workbook

```

```

' عمل ورقة متغير عن الاعلان
Dim ExcelWorksheet As Worksheet

Dim misval As Object = System.Reflection.Missing.Value
' ملف انشاء
ExcelWorkbook = ExcelApp.Workbooks.Add(misval)
' العمل ورقة مع التعامل
ExcelWorksheet = ExcelWorkbook.Sheets("Sheet1")

For titlecol As Integer = 0 To DataGridView1.ColumnCount - 1
    ' العمل ورقة في الشبكة اعمدة عناوين وضع
    ExcelWorksheet.Cells(1, titlecol + 1) =
DataGridView1.Columns(titlecol).HeaderText
Next

For i As Integer = 0 To DataGridView1.RowCount - 2
    For j As Integer = 0 To DataGridView1.ColumnCount - 1
        ' الاكسل في عمل ورقة الى الشبكة بيانات نقل
        ExcelWorksheet.Cells(i + 2, j + 1) =
DataGridView1.Rows(i).Cells(j).Value.ToString
    Next
Next
sfd.Filter = "Excel 2003|*.xls|Excel File|*.xlsx"
If sfd.ShowDialog = System.Windows.Forms.DialogResult.OK Then
    ' عمل ورقة في ملف حفظ
    ExcelWorksheet.SaveAs(sfd.FileName)
End If
ExcelWorkbook.Close()
ExcelApp.Quit()
' القادم التنفيذ في المشاكل لتجنب القيم تحرير
System.Runtime.InteropServices.Marshal.ReleaseComObject(ExcelApp)
System.Runtime.InteropServices.Marshal.ReleaseComObject(ExcelWorkbook)
System.Runtime.InteropServices.Marshal.ReleaseComObject(ExcelWorksheet)
' القيم افراغ
ExcelApp = Nothing
ExcelWorkbook = Nothing
ExcelWorksheet = Nothing
' الحفظ بعد المف فتح
If MsgBox("Do you want open file?", MsgBoxStyle.YesNo) = MsgBoxResult.Yes Then
    Process.Start(sfd.FileName)
End If
End Sub

```

المشروع 25

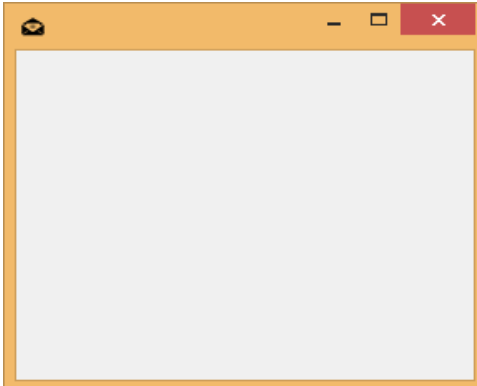
NotifyIcon

ستتقن من خلال المشروع

- التعامل مع NotifyIcon
- التعامل مع ContextMenuStrip
- التعامل مع MessageBox

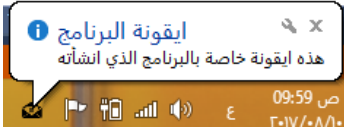
اهداف المشروع

عند تنفيذ المشروع تظهر الواجهة التالية:

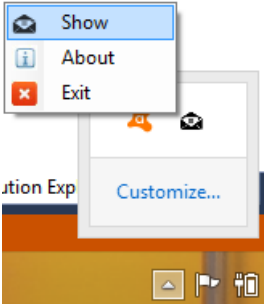


- عند النقر على زر Minimize من شريط العنوان سيختفي النموذج من سطح المكتب.

- ليظهر مربع بجانب ساعة النظام في ناحية الإعلام من شريط المهام كما في الشكل التالي.



- عند النقر على السهم الموجود في القسم الأيمن من شريط المهام ستظهر أيقونة البرنامج مع ايقونات البرامج الأخرى داخل مربع عند النقر على ايقونة البرنامج بزر الفأرة الأيمن ستظهر قائمة مختصرة تحتوي على ثلاث أوامر بجانب كل أمر صورة فالامر Show لعرض النموذج على سطح المكتب، About لعرض معلومات عن مبرمج البرنامج والأمر Exit لإنهاء البرنامج. كما في الشكل المجاور.



الحدث Form1_Resize

يحصل الحدث Resize على النموذج عند كل تغيير في حجم النموذج، ويأخذ الاجراء التالي:

```
Private Sub Form1_Resize(sender As Object, e As EventArgs)
```

الحدث NotifyIcon1_MouseDoubleClick

يحصل هذا الحدث عند النقر على ايقونة البرنامج الموجودة في ناحية الإعلام من شريط المهام مرتين متتاليتين، ويأخذ الحدث الإجراء التالي:

Private Sub NotifyIcon1_MouseDoubleClick(sender As Object, e As MouseEventArgs)

صندوق الرسالة MessageBox

يستخدم لإظهار رسالة للمستخدم تخبره بشيء خاص بعمل البرنامج كالخروج من برنامج او معلومات تخص عمليات الحذف او الإضافة او التحديث.

الشكل العام لصندوق النص هو:

MessageBox(Prompt, Title, Buttons, Icon)

حيث ان:

- **Prompt**: هي الرسالة التي ستظهر داخل الصندوق.
- **Title**: يمثل عنوان الصندوق.
- **Buttons**: الأزرار التي يمكن وضعها داخل الصندوق.
- **Icon**: الأيقونة التي ستظهر داخل الصندوق والتي تشبه الى حد كبير الأيقونات في MsgBox.

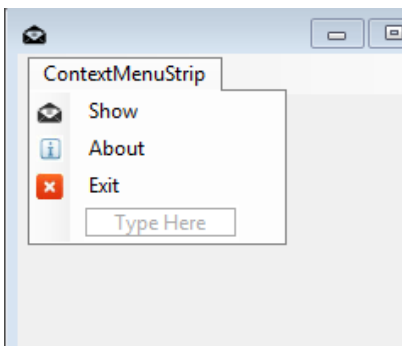
مراحل المشروع

1- انشئ مشروعاً جديداً.

2- اصف أداة NotifyIcon الى النموذج واضبط خصائصها كالتالي:

NotifyIcon	
BalloonTipIcon	Info
BalloonTipText	ايقونة الخاصة بالبرنامج الذي انشأته
BalloonTipTitle	ايقونة البرنامج
ContextMenuStrip	ContextMenuStrip1
Icon	اختر ايقونة للاداة
Text	برنامجي

3- اصف الاداة ContextMenuStrip الى النموذج ثم ضع في القائمة ثلاث عناصر وسمها كما تظهر في الشكل المجاور. واختر الايقونة الملائمة لإسم كل عنصر.



4- اكتب التعليمات التالية على اجراء From_Resize:

في هذا الإجراء عند النقر على زر Minimize الخاصة بنموذج البرنامج يختفي النموذج من سطح المكتب وتظهر ايقونة البرنامج في شريط المهام ويظهر صندوق بعنوان ومحتوى خاص بالبرنامج لفترة زمنية قصيرة ثم يختفي.

```
Private Sub form1_Resize(sender As Object, e As EventArgs) Handles MyBase.Resize
    النموذج تصغير زر على النقر عند'
    If Me.WindowState = FormWindowState.Minimized Then
        النموذج إخفاء'
        Me.Hide()
    End If
End Sub
```

```

النظام ساعة بجوار البرنامج ايقونة إظهار'
NotifyIcon1.Visible = True
' الأداة بالون اظهار مدة '
NotifyIcon1.ShowBalloonTip(5)
' الأداة بالون داخل الأيقونة'
NotifyIcon1.BalloonTipIcon = ToolTipIcon.Info
' الأداة بالون في سيظهر الذي النص '
NotifyIcon1.BalloonTipText = "انشأته الذي بالبرنامج الخاصة ايقونة"
' البالون في النص عنوان '
NotifyIcon1.BalloonTipTitle = "البرنامج ايقونة"

```

```

End If
End Sub

```

5- برمجة حدث NotifyIcon_MouseDoubleClick

في هذا الإجراء عند النقر على ايقونة البرنامج نقرتين متتاليتين بزر الفأرة الايمن تختفي ايقونة البرنامج من شريط المهام ويظهر نموذج البرنامج على سطح المكتب بحجمه الاعتيادي.

```

Private Sub NotifyIcon1_MouseDoubleClick(sender As Object, e As MouseEventArgs)
Handles NotifyIcon1.MouseDoubleClick
' النموذج إظهار'
Me.Show()
' المهام شريط من الأيقونة إخفاء
NotifyIcon1.Visible = False
' الاعتيادية بالحالة المكتب سطح على النموذج عرض
Me.WindowState = FormWindowState.Normal
End Sub

```

6- برمجة العنصر Show من القائمة المختصرة لزر الفأرة الأيمن:

```

Private Sub ShowToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ShowToolStripMenuItem.Click
' النموذج إظهار'
Me.Show()
' المهام شريط من الأيقونة إخفاء
NotifyIcon1.Visible = False
' الاعتيادية بالحالة المكتب سطح على النموذج عرض
Me.WindowState = FormWindowState.Normal
End Sub

```

7- برمجة العنصر About من قائمة الزر الأيمن للفأرة:

```

Private Sub AboutToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
AboutToolStripMenuItem.Click
' المشروع ميرمج عن رسالة إظهار'
MessageBox.Show("البرنامج عن", "جميل حمزة اعداد من برنامج",
MessageBoxButtons.OK,
MessageBoxIcon.Information)
End Sub

```

8- برمجة عنصر Exit من قائمة الزر الأيمن:

```
Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles  
ExitToolStripMenuItem.Click  
    'التطبيق انهاء'  
    Application.Exit()  
End Sub
```

المشروع 26

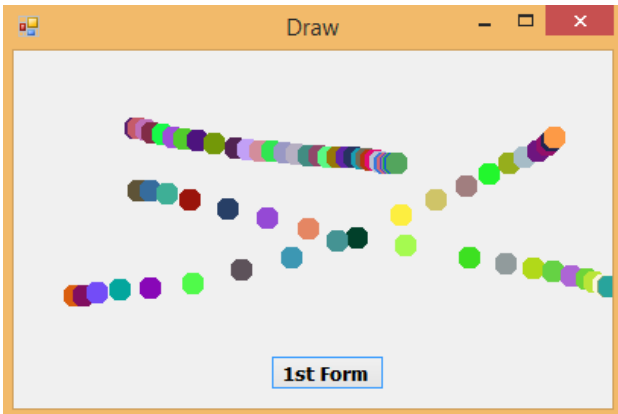
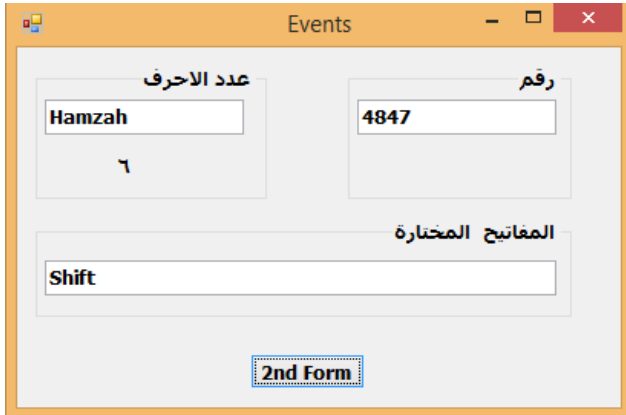
الاحداث (1)

ستتقن من خلال المشروع

- التعامل مع الاحداث
- Change تغيير محتوى صندوق النص
- KeyPress النقر على مفتاح
- KeyDown النقر على مفتاح
- MouseDown النقر على زر الفأرة
- MouseUp تحرير زر الفأرة المضغوط
- MouseMove تحريك مؤشر الفأرة مع استمرار الضغط على احد ازرارها

اهداف المشروع

عند تنفيذ المشروع تظهر الواجهة التالية:



- عندما يقوم المستخدم بكتابة او حذف حرفاً في صندوق النص الاول سيظهر عدد الاحرف اسفل الصندوق.
- في صندوق النص الثاني لا يمكن للمستخدم بدخال ارقام.
- عندما ينقر المستخدم على مفتاح Shift او Ctrl سيظهر اسم المفتاح المضغوط في صندوق النص الثالث.
- عندما ينقر المستخدم على زر 2nd Form سيظهر النموذج الثاني الذي يتضمن رسم دوائر ملونة باستخدام مؤشر الماوس وعند النقر على زر 1st Form تتم العودة الى النموذج الأول.

مراحل المشروع

- 1- انشئ مشروع جديد.
 - 2- اصف العناصر الظاهرة على النموذجين.
 - 3- برمجة صندوق النص الاول.
- الحدث **Change** يحصل عندما يقوم المستخدم بتغيير محتوى صندوق النص

اكتب الاجراء التالي في حدث **Change** لصندوق النص **TextBox1**

```
Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles
TextBox1.TextChanged
    ' النصية السلسلة طول ايجاد
    Label1.Text = Len(TextBox1.Text)
End Sub
```

4- برمجة صندوق النص الثاني TextBox2

لجعل صندوق النص لا يقبل الا ارقام بالاضافة الى الفاصلة العشرية، يجب علينا فحص المفتاح الذي تم النقر عليه، اذا كان رقم نسمح له بالظهور في صندوق النص ولتحقيق هذا الامر سنستخدم حدث **KeyPress** يحصل هذا الحدث عندما ينقر المستخدم على مفتاح من لوحة المفاتيح.

حيث تم وضع نظام باسم ANSI وهو مختصر يمثل الاحرف الاولى من اسم المعهد الوطني الامريكي للمعايير ويختص بوضع معايير للغات البرمجة والانظمة.

حيث وضع هذا المعهد جدولاً لترميز 256 حرفاً من لوحة المفاتيح ويسمى هذا الجدول باسم (ASCII) ويمثل American Standerd Code For Information Interchange.

حيث ان المحارف (0 - 127) تمثل المحارف والرموز للوحة المفاتيح US والمحارف (128 - 256) تمثل المحارف الخاصة باللغة العربية للوحة المفاتيح. كما سيمر في الجدول لاحقاً.

اكتب الاجراء التالي في حدث **KeyPress** للصندوق الثاني.

```
Private Sub TextBox2_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox2.KeyPress
    ' عليه الضغط تم الذي المفتاح
    Select Case e.KeyChar
        Case Is = ChrW(8) ' Back Space

        Case Is = ChrW(46) ' العشرية الفاصلة

        Case ChrW(48) To ChrW(57) ' 0 الى 9 الارقام

        Case Else
            e.KeyChar = ChrW(0) ' الادخال إلغاء
    End Select
End Sub
```

5- برمجة صندوق النص الثالث TextBox3.

لتمييز المفاتيح الوظيفية لكائن معين يجب استخدام حدث **KeyDown** او **KeyUp** على الكائن.

الحدث **KeyDown** يحصل عندما يقوم المستخدم بالضغط على مفتاح معين من لوحة المفاتيح.

الحدث **KeyUp** يحصل عندما يقوم المستخدم بتحرير مفتاح مضغوط من لوحة المفاتيح.

اكتب الاجراء التالي في حدث **KeyDown** لصندوق النص TextBox3

```
Private Sub TextBox3_KeyDown(sender As Object, e As KeyEventArgs) Handles
TextBox3.KeyDown
    ' Shift مفتاح على ضغط
    If e.KeyCode = Keys.ShiftKey Then
        TextBox3.Text = "Shift"
    ' Ctrl مفتاح ضغط
```

```

ElseIf e.KeyCode = Keys.ControlKey Then
    TextBox3.Text = "Ctrl"
End If
End Sub

```

6- اكتب الاجراء التالي في زر "2nd Form"

حيث يتضمن هذا الاجراء اخفاء النموذج الاول و اظهار النموذج الثاني.

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    ' النموذج اخفاء
    Me.Hide()
    ' الثاني النموذج اظهار
    Form2.Show()
End Sub

```

7- رسم الدوائر على سطح النموذج الثاني

في قسم التصريحات العامة للنموذج ضع المتغيرات التالية:

```

' منطقي متغير
Dim doit As Boolean
' رسومي متغير
Dim gr As Graphics

```

اكتب التعليمة التالية في حدث Load للنموذج

```

Private Sub Form2_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ' الرسومي بالمتغير التحكم
    gr = Me.CreateGraphics
End Sub

```

الحدث **MouseDown** يحصل عندما يقوم المستخدم بضغط اح ازرار الفأرة

اكتب التعليمة التالية في حدث MouseDown للنموذج

```

Private Sub Form2_MouseDown(sender As Object, e As MouseEventArgs) Handles
Me.MouseDown
    doit = True
End Sub

```

الحدث **MouseMove** يحصل هذا الحدث عندما يقوم المستخدم بتحريك مؤشر الفأرة.

اكتب التعليمات التالية في حدث MouseMove للنموذج

```

Private Sub Form2_MouseMove(sender As Object, e As MouseEventArgs) Handles
Me.MouseMove
    ' المنطقي المتغير قيمة اختيار
    If doit Then
        ' الرسم فرشاة لون تحديد
        Dim br As New SolidBrush(Color.FromArgb(Rnd() * 255, Rnd() * 255, Rnd() *
255))
        ' الهندسي الشكل رسم
        gr.FillEllipse(br, New Rectangle(e.X - 5, e.Y - 5, 15, 15))
    End If
End Sub

```

الحدث **MouseUp** يحصل هذا الحدث عندما يقوم المستخدم بتحرير زر الفأرة المضغوط.

اكتب التعليمات التالية في حدث **MouseUp** للنموذج.

```
Private Sub Form2_MouseUp(sender As Object, e As MouseEventArgs) Handles Me.MouseUp
    ' المنطقي لمتغير قيمة انساد
    doit = False
End Sub
```

8- اكتب الاجراء التالي في زر "1st Form":

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    ' النموذج اخفاء
    Me.Hide()
    ' الثاني النموذج اظهر
    Form1.Show()
End Sub
```

جدول لاهم ترميزات المحارف (0 - 127) الخاصة US

Code	char	Code	char	Code	char	Code	char
1	_	57	9	81	Q	105	i
8	BackSpace	58	:	82	R	106	j
30	-	59	;	83	S	107	k
36	\$	60	<	84	T	108	l
37	%	61	=	85	U	109	m
38	&	62	>	86	V	110	n
39	'	63	?	87	W	111	o
40	(64	@	88	X	112	p
41)	65	A	89	Y	113	q
42	*	66	B	90	Z	114	r
43	+	67	C	91	[115	s
44	,	68	D	92	\	116	t
45	-	69	E	93]	117	u
46	.	70	F	94	^	118	v
47	/	71	G	95	_	119	w
48	0	72	H	96	'	120	x
49	1	73	I	97	a	121	y
50	2	74	J	98	b	122	z
51	3	75	K	99	c	123	{

52	4	76	L	100	d	124	
53	5	77	M	101	e	125	}
54	6	78	N	102	f	126	~
55	7	79	O	103	g	127	_
56	8	80	P	104	h		

المشروع 27

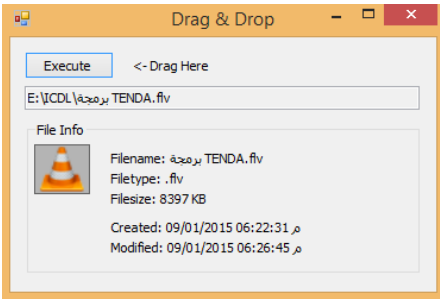
الأحداث (2)

ستتقن من خلال المشروع

التعامل مع حدث DragDrop
التعامل مع حدث DragEnter

اهداف المشروع

عند تشغيل المشروع تظهر الواجهة التالية:



التي تتضمن معلومات عن الملف الذي تم سحبه وافلاته فوق زر "Execute". ومن هذه المعلومات هي مكان حفظ الملف، اسم الملف، نوع الملف، حجم الملف مقاساً بالكيلو بايت (KB)، تاريخ انشاء الملف وتاريخ التعديل. كما تحوي النافذة على صورة صغيرة لأيقونة الملف وتختلف صورة الايقونة باختلاف انواع الملفات.

عند النقر على زر "Execute" يتم تشغيل الملف باستخدام البرنامج الخاص به. فلو كان الملف صورة يتم عرضها من خلال برنامج عرض الصور واذا كان الملف نصي يتم فتحه ببرنامج NotePad أو اي برنامج اخر خاص بعرض الملفات النصية وهكذا مع بقية الملفات.

مراحل المشروع

1- انشئ مشروعاً جديداً.

2- اضف الأدوات الظاهرة على النموذج. اجعل خاصية AllowDrag=True للأداة Button لجعلها تستقبل الافلات عليها. والخاصية SizeMode=StretchImage للعنصر PictureBox لضبط مكان ظهور الصورة داخل الاداة. في قسم التصريحات العامة للنموذج ضع التعليمة التالية.

```
Imports System.IO
```

تحتوي System.IO على انواع تسمح بالقراءة والكتابة بالملفات بالاضافة الى انواع الدعم الأساسي للملف والدليل.

اكتب التعليمة التالية في Class Form

```
Private Myfile As FileInfo
```

تزودك تعليمة FileInfo بخصائص وطرف انشاء، نسخ، حذف ونقل الملفات.

ثم ضع الاجراء التالي والمتضمن عرض خصائص الملف في قسم Claas Form

```
Private Sub showfileinfo()  
    ' الملف خصائص عرض  
    TextBox1.Text = Myfile.FullName  
    lblfilename.Text = "Filename: " & Myfile.Name  
    lblfiletype.Text = "Filetype: " & Myfile.Extension  
    ' بايت بالكيلو مقاس الملف حجم
```

```

lblfilesize.Text = "Filesize: " & Math.Round(Myfile.Length / 1024) & " KB"
lblcreated.Text = "Created: " & Myfile.CreationTime
lblmodified.Text = "Modified: " & Myfile.LastWriteTime
' الصورة صندوق في الملف ايقونة وضع '
PictureBox1.Image = Icon.ExtractAssociatedIcon(Myfile.FullName).ToBitmap
End Sub

```

3- برمجة زر الأمر:

يستقبل هذا الزر البيانات المسحوبة والتي سيتم افلاتها فوقه . ونستخدم في هذه الحالة حدثي **DragDrop** و **DargEnter**.
الحدث DragDrop يحدث عندما يقوم المستخدم بسحب البيانات وافلاتها فوق الأداة.

ويأخذ هذا الحدث الشكل التالي:

```

Private Sub Object_DragDrop(sender As Object, e As DragEventArgs) Handles
Object.DragDrop
حيث ان Object: يمثل الكائن الذي يستقبل الحدث.

```

الحدث DragEnter يحدث عندما يكون مؤشر الفأرة فوق الأداة محملاً بالبيانات. ويأخذ الشكل التالي:

```

Private Sub Object_DragEnter(sender As Object, e As DragEventArgs) Handles
Object.DragEnter
Object يمثل الكائن المستقبل للحدث.

```

اكتب على حدث **DargDrop** لزر الأمر

```

Private Sub Button1_DragDrop(sender As Object, e As DragEventArgs) Handles
Button1.DragDrop
' افلاته المطلوب الملف مسار اخذ '
Dim file As String = e.Data.GetData(DataFormats.FileDrop)(0)
' الملف معلومات و مسار اخذ '
Myfile = New FileInfo(file)
' مقبول غير الملف كان اذا العملية الغاء '
If String.IsNullOrEmpty(Myfile.Extension) Then Exit Sub
' الملف خصائص عرض اجراء استدعاء '
showfileinfo()
End Sub

```

اكتب التعليمات التالية في حدث **DragEnter** لزر الأمر

```

Private Sub Button1_DragEnter(sender As Object, e As DragEventArgs) Handles
Button1.DragEnter
' سحبه المطلوب الملف عرض '
If e.Data.GetDataPresent(DataFormats.FileDrop) Then
e.Effect = DragDropEffects.Copy
End If
End Sub

```

في حدث **Click** لزر الأمر ضع التعليمات التالية:

في هذا الحدث سيقوم البرنامج بفتح او تشغيل الملف المسحوب والمفلت فوق زر الأمر.

```

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

```

' البرنامج خلال من الملف فتح او تنفيذ '

Try

Process.Start(TextBox1.Text)

Catch ex As Exception

' خطأ رسالة '

MsgBox(ex.ToString)

End Try

End Sub

المشروع 28

التعامل مع قواعد البيانات

ستتقن من خلال المشروع

التعامل مع DataSet

اهداف المشروع

عند تنفيذ البرنامج تظهر الواجهة التالية والتي تحوي على ثلاث أجزاء، الجزء الاول: يضم الشبكة التي تقوم بعرض جميع حقول جدول قاعدة البيانات.

Code	FirstName	LastName	Gender	DateOfBrith	Address	Phone	E_mail
23332	Ali	Jameel	Male	12/05/1990	Kufa	390439048	Alijml@gmail.com
23290	Huda	Ammar	Female	10/10/1998	Baghdad	8978333	Hudamr@gmail.c...
489574	Sura	Jasim	Female	07/10/1996	Karkuk	485784	difhd@yahoo.com
89561	Muna	Satjad	Female	01/10/1996	Baghdad	4985489	Muna@gmial.com
85748	Khalid	Modhar	Male	21/05/1990	Diyala	485749587	Khar@gmail.com
84957948	Humam	Modhar	Male	12/05/1991	pofqfio	945849899	dklfr@gmail.com

البيانات الأساسية

Code: 23332 Date Of Brith: 1990/05/12

First Name: Ali Address: Kufa

Last Name: Jameel Phone: 390439048

Gender: Male Male Female E mail: Alijml@gmail.com

أزرار البرنامج

<< < New Save Update Delete Exit > >>

الجزء الثاني: يحوي صناديق النص والدلالات الخاصة ببيها.

الجزء الثالث: يحوي على الأزرار الخاصة بالعمليات المختلفة على البيانات.

• عندما يقوم المستخدم بالنقر

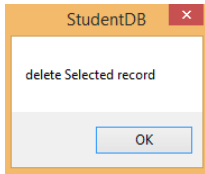
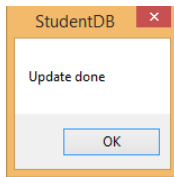
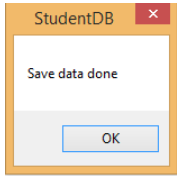
على "New" يتم افراغ صناديق النص للمستخدم باضافة بيانات جديدة.

• عند النقر على "Save" يتم حفظ البيانات الجديدة في قاعدة البيانات وتأكيداً لذلك تظهر الرسالة كما في الشكل المجاور.

• ولحفظ تحديث البيانات يتم النقر على زر "Update".

• اختر السجل المطلوب ثم انقر على زر "Delete" ليتم حذف البيانات وتأكيداً لعملية الحذف تظهر الرسالة المجاورة.

• يمكن التنقل بين سجلات قاعدة البيانات من خلال النقر على الازرار الخاصة فمثلاً يؤدي النقر على زر "<<" للانتقال الى السجل الأول، والزر ">" للانتقال الى السجل السابق ، للانتقال الى السجل الاخير يتم بالنقر على زر ">>"، وللوصول الى السجل اللاحق يتم بالنقر على زر ">".

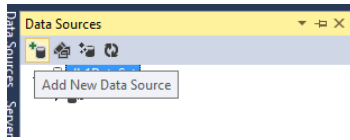


1- انشئ قاعدة بيانات باستخدام برنامج مايكروسوفت أكسس (Microsoft Access 2013) أو أي برنامج آخر من برامج التعامل مع قواعد البيانات على أن تتضمن هذه القاعدة الحقول الظاهرة في الشكل المجاور.

Field Name	Data Type
Code	Number
FirstName	Short Text
LastName	Short Text
Gender	Short Text
DateOfBrith	Date/Time
Address	Short Text
Phone	Short Text
E_mail	Short Text

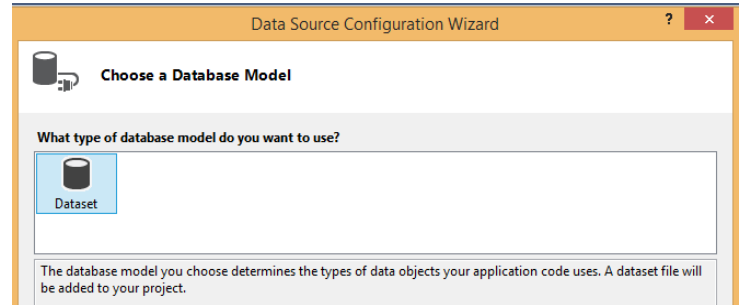
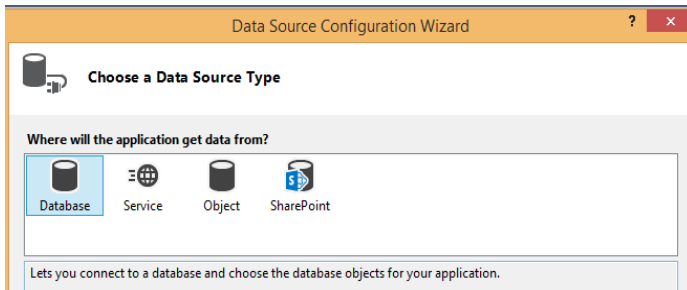
2- انشئ مشروعاً جديداً.

3- اظهر صندوق Data Sources من خلال Data Sources → Other Windows → View



4- من صندوق Data Sources انقر على الاداة كما في الشكل المجاور

ستظهر نافذة تسمح لك بتحديد مصدر قاعدة البيانات اختر منها Database ثم انقر على Next ستظهر نافذة اخرى تسمح لك بتحديد اداة الاتصال بقاعدة البيانات اختر منها الأداة Dataset.



ثم انقر Next ستظهر نافذة اخرى يمكنك من تحديد الأتصال بقاعدة البيانات منها اختر New Connection ستظهر نافذة الاتصال انقر على زر Change لتحديد مصدر البيانات ستظهر نافذة باسم "Change Data Source" تحتوي على عدة مصادر للبيانات عليك اختيار (Microsoft Access Database File (OLEDB)) لأننا سنتصل بقاعدة بيانات تم انشاؤها باستخدام برنامج Microsoft Access.

عليك تحديد مكان حفظ قاعدة البيانات على جهازك من خلال النقر على زر Browse، ستظهر نافذة تسمح لك باختيار قاعدة

البيانات. بعد تحديد

مصدر البيانات

واختيار قاعدة

البيانات يُفضل

التحقق من صحة

الاتصال بقاعدة

البيانات من خلال

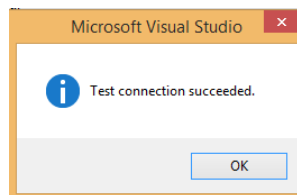
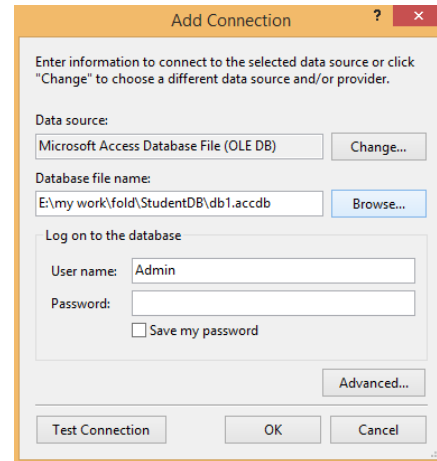
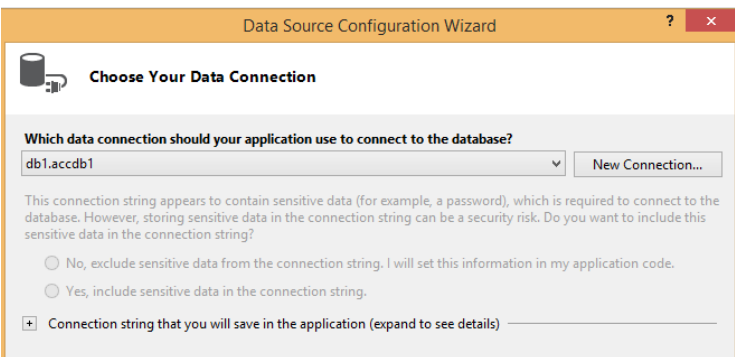
النقر على زر

Test

من Connection

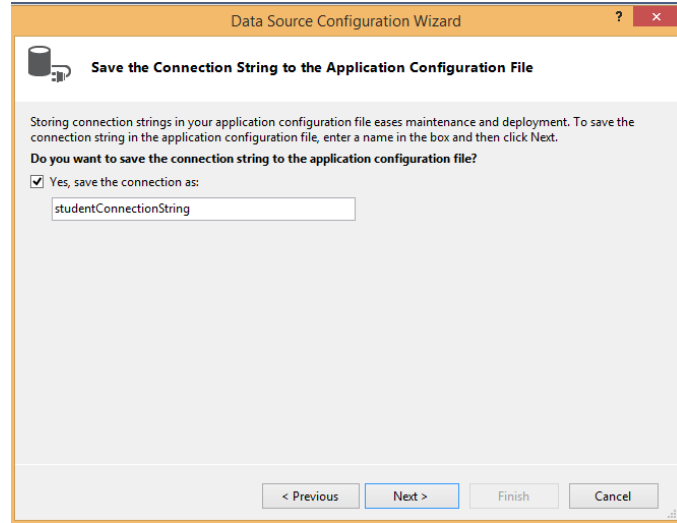
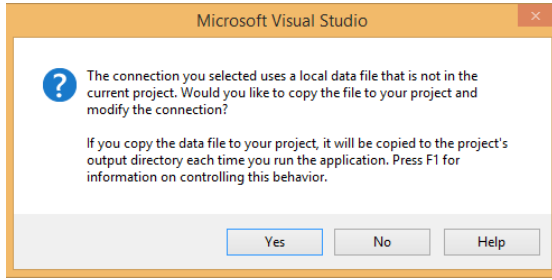
نافذة "Add

Connection"

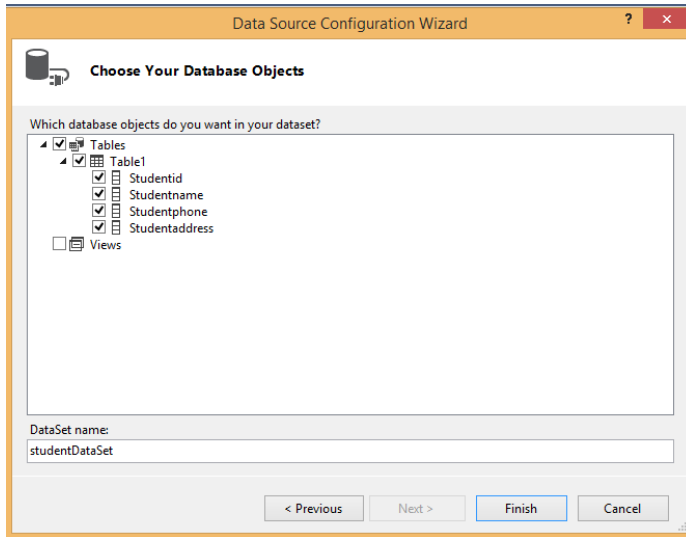


سيظهر مربع حوار يحوي على نص (Test connection succeeded) اذا كان الاتصال بقاعدة البيانات صحيح.

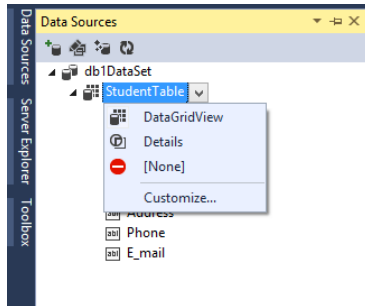
انقر على Ok ثم استمر بالنقر على Next سيظهر مربع حوار يتيح لك عمل نسخة من قاعدة البيانات داخل مشروعك اختر No ومن النافذة التالية ضع اسم للاتصال او اترك الاسم الافتراضي كما هو انظر الاشكال التالية:



بالنقر على Next ستظهر نافذة اخرى تسمح لك باختيار الجدول وكذلك اختيار الحقول المطلوبة منه عليك النقر داخل المربع المجاور لاسم الجدول لاختيار جميع حقول الجدول. كما يمكنك وضع اسم للـ Dataset او الإبقاء على الاسم الافتراضي لها. كما في الشكل المجاور.



لاتمام عملية الاتصال بقاعدة البيانات انقر Finish.



5- من صندوق Data Source انقر بزر الفأرة الأيمن على اسم الجدول واختر الامر DataGridView ثم اضغط على اسم الجدول واسحبه الى النموذج ثم افلته لإضافة الشبكة الى سطح النموذج، انقر بزر الفأرة الأيمن مرة اخرى على اسم الجدول واختر الامر Details ثم اضغط على اسم الجدول واسحبه الى النموذج ثم افلته لإضافة صناديق النص والدلالات الخاصة بقاعدة البيانات.

سيضاف تلقائياً في أعلى النموذج شريط أدوات خاص بالتعامل مع قاعدة البيانات احذفه ثم رتب الأدوات على سطح النموذج.

ملاحظة: عند سحب الأدوات الى سطح النموذج سٌضاف تلقائياً مجموعة من الاكواد الخاصة بها. احذفها لأننا سنتعامل مع الأمر برمجياً ولا نهتم لما يضيفه البرنامج.

6- اضع صندوق مجموعة باسم "البيانات الاساسية" وضع فيه صناديق النص والدلالات. ثم اضع زري راديو الاول باسم "Male" والثاني باسم "Female" و اضع صندوق مجموعة ثاني باسم "ازرار البرنامج".

7- اضع 9 أزرار أمر بالتسميات الظاهرة على النموذج وضعها داخل صندوق المجموعة الذي يحمل اسم "أزرار البرنامج".

8- في قسم التصريحات العامة للنموذج ضع الاجراء التالي:

يقوم هذا الاجراء بتفعيل أزرار الراديو حسب جنس الطالب الموجود في حقل "Gender" لقاعدة البيانات.

```
Private Sub change()
```

```
    ' Male يساوي الجنس صندوق نص يكون عندما
```

```
    If GenderTextBox.Text = RadMale.Text Then
```

```
        ' Male بال الحاص راديو زر تفعيل
```

```
        RadMale.Checked = True
```

```
        ' Female بال الخاص راديو زر تفعيل الغاء
```

```
        RadFemale.Checked = False
```

```
        ' Female يساوي الجنس صندوق نص كان اذا
```

```
    ElseIf GenderTextBox.Text = RadFemale.Text Then
```

```
        ' Female بال الخاص راديو زر تفعيل
```

```
        RadFemale.Checked = True
```

```
        ' Male لل راديو زر الغاء
```

```
        RadMale.Checked = False
```

```
    End If
```

9- اكتب التعليمات التالية في زر "New"

```
Private Sub BtnNew_Click(sender As Object, e As EventArgs) Handles BtnNew.Click
```

```
    ' جديد سجل اضافة
```

```
    StudentTableBindingSource.AddNew()
```

```
    ' code لصندوق الكتابة مؤشر نقل
```

```
    CodeTextBox.Focus()
```

```
    ' Male راديو زر تفعيل
```

```
    RadMale.Checked = True
```

```
    ' الجنس صندوق في الراديو زر نص وضع
```

```
    GenderTextBox.Text = RadMale.Text
```

```
End Sub
```

10- اكتب التعليمات التالية في زر "Save"

```
Private Sub BtnSave_Click(sender As Object, e As EventArgs) Handles BtnSave.Click
```

```
    ' الحفظ اتمام
```

```
    StudentTableBindingSource.EndEdit()
```

```
    ' dataset بيانات تحديث
```

```
    TableAdapterManager.UpdateAll(Db1DataSet)
```

```
    ' الحذف عملية تأكيد رسالة
```

```
    MsgBox("Save data done")
```

```
End Sub
```

11- اكتب التعليمات التالية في زر "Update"

```
Private Sub BtnUpdate_Click(sender As Object, e As EventArgs) Handles
```

```
BtnUpdate.Click
```

```
    ' الحفظ عملية اتمام
```

```

StudentTableBindingSource.EndEdit()
' بيانات تحديث dataset
TableAdapterManager.UpdateAll(Db1DataSet)
' البيانات تحديث تأكيد رسالة
MsgBox("Update done")
End Sub

```

12- اكتب التعليمات التالية في زر "Delete"

```

Private Sub BtnDelete_Click(sender As Object, e As EventArgs) Handles
BtnDelete.Click
' الحالي السجل حذف
StudentTableBindingSource.RemoveCurrent()
' السجل حذف عملية تأكيد رسالة
MsgBox("Selected record deleted")
End Sub

```

13- اكتب التعليمات التالية في زر "Exit"

```

Private Sub BtnExit_Click(sender As Object, e As EventArgs) Handles BtnExit.Click
' البرنامج من الخروج
Application.Exit()
End Sub

```

14- اكتب الكود التالي في زر "<<"

```

Private Sub BtnFirst_Click(sender As Object, e As EventArgs) Handles
BtnFirst.Click
' الاول للسجل الانتقال
StudentTableBindingSource.MoveFirst()
' الجنس صندوق محتوى تغيير اجراء استدعاء
change()
End Sub

```

15- اكتب التعليمات التالية في زر "<"

```

Private Sub BtnPre_Click(sender As Object, e As EventArgs) Handles BtnPre.Click
' السابق للسجل الانتقال
StudentTableBindingSource.MovePrevious()
' الجنس صندوق محتوى تغيير اجراء استدعاء
change()
End Sub

```

16- اكتب التعليمات التالية في زر ">>"

```

Private Sub BtnLast_Click(sender As Object, e As EventArgs) Handles BtnLast.Click
' الاخير للسجل الانتقال
StudentTableBindingSource.MoveLast()
' الجنس صندوق محتوى تغيير اجراء استدعاء
change()
End Sub

```

17- ضع التعليمات التالية في زر ">"


```
Private Sub BtnNext_Click(sender As Object, e As EventArgs) Handles BtnNext.Click
    ' التالي للسجل الانتقال
    StudentTableBindingSource.MoveNext()
    ' الجنس صندوق محتوى تغيير اجراء استدعاء
    change()
End Sub
```

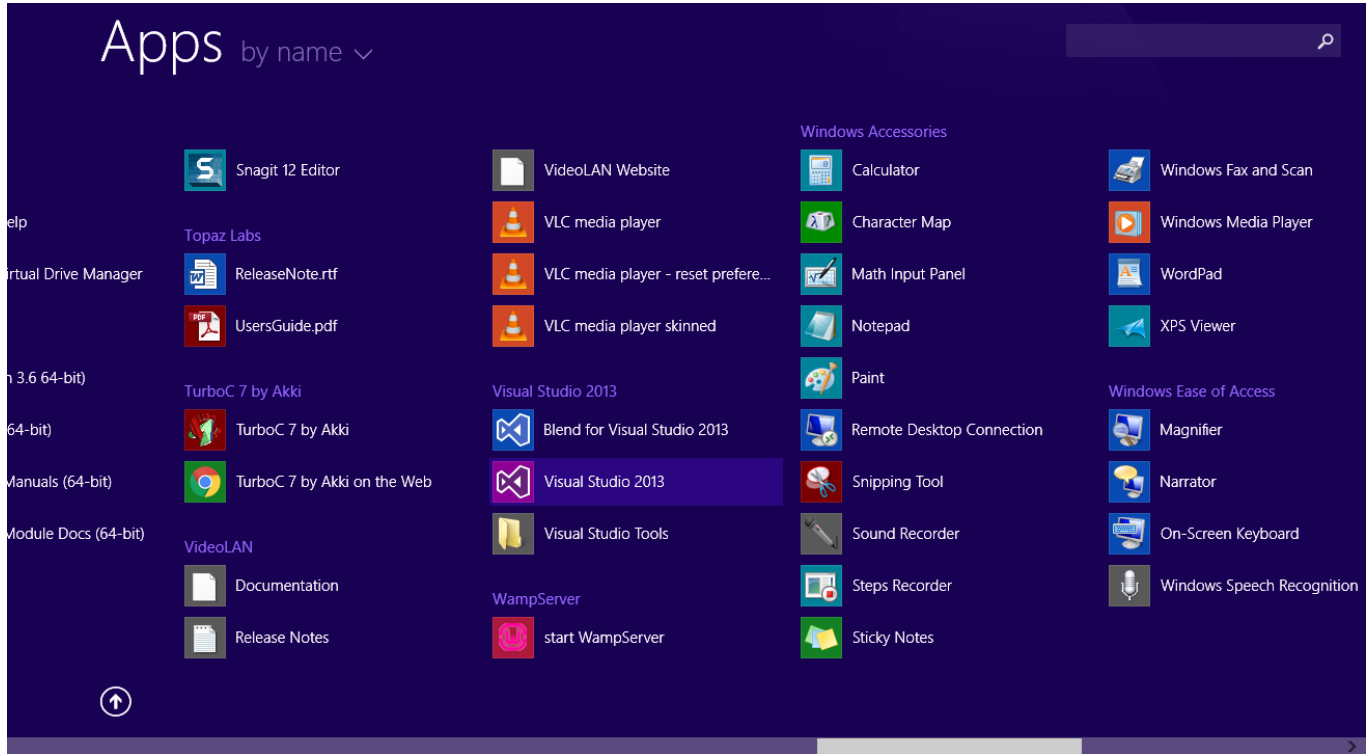
الفصل الثاني

نوافذ VB

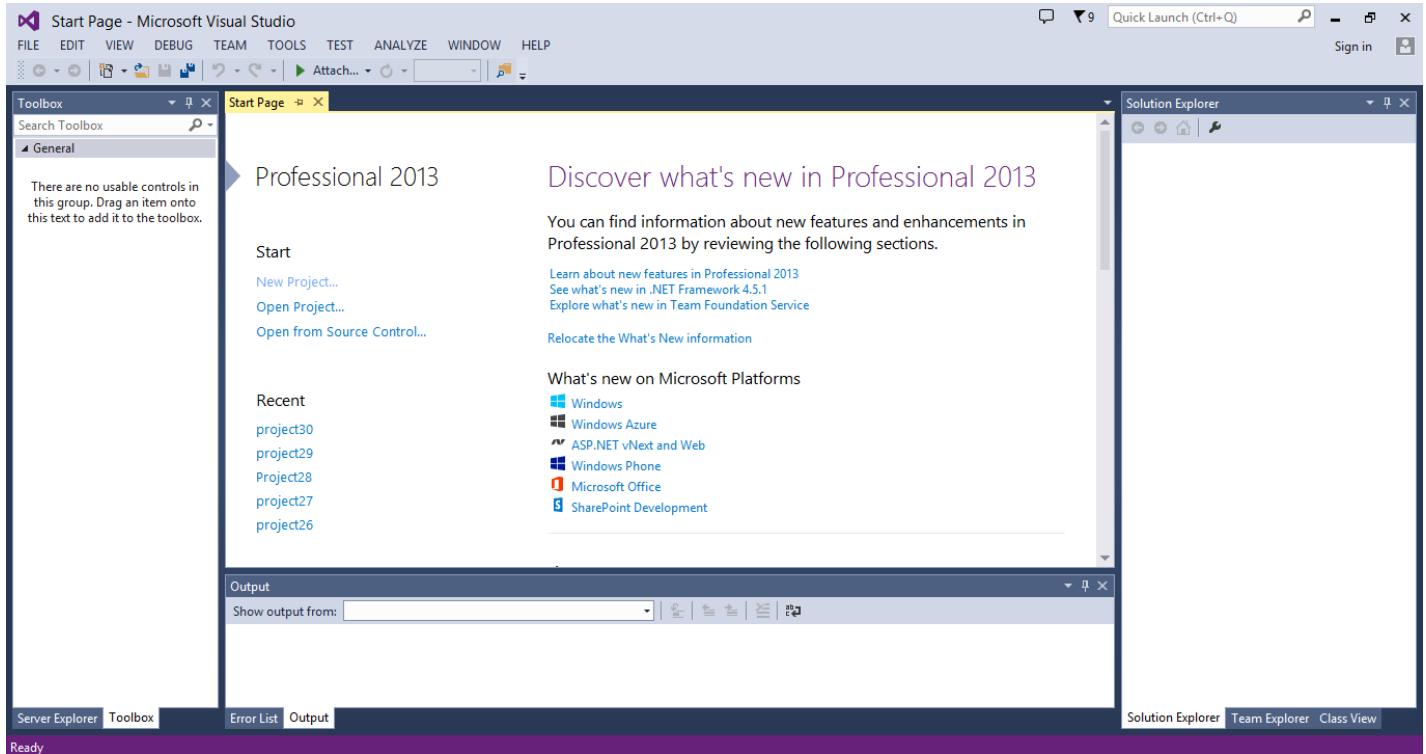
محتويات الفصل

- تشغيل الفيچوال بيسك VB
- انشاء مشروع جديد
- مكونات الواجهة الرئيسية للفيچوال بيسك
- شريط العنوان Title Bar
- شريط القوائم Menus Bar
- شريط الأدوات Tools Bar
- نافذة تصميم النموذج Form Designer
- متصفح المشروع Solution Explorer
- صندوق الأدوات Tool Box
- نافذة البرمجة Code Windows
- نافذ الخصائص Properties Windows

بعد تنصيب الفيجوال بيسك على حاسوبك الشخصي فم بتشغيل اللغة من خلال النقر على الأيقونة التي تحمل اسمها. كما في الشكل التالي.

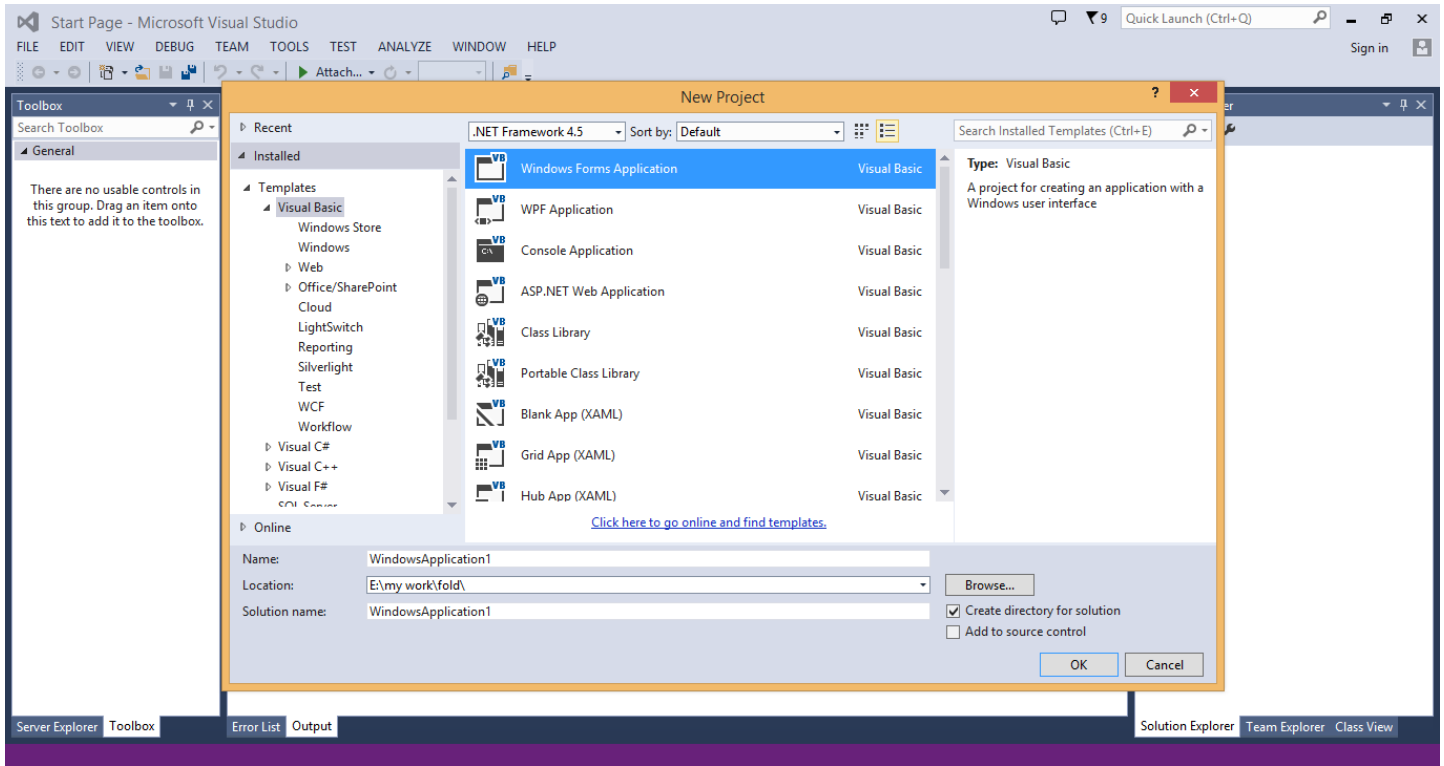


بعدها ستظهر لك الواجهة الرئيسية للغة الفيجوال بيسك كما في الشكل التالي.

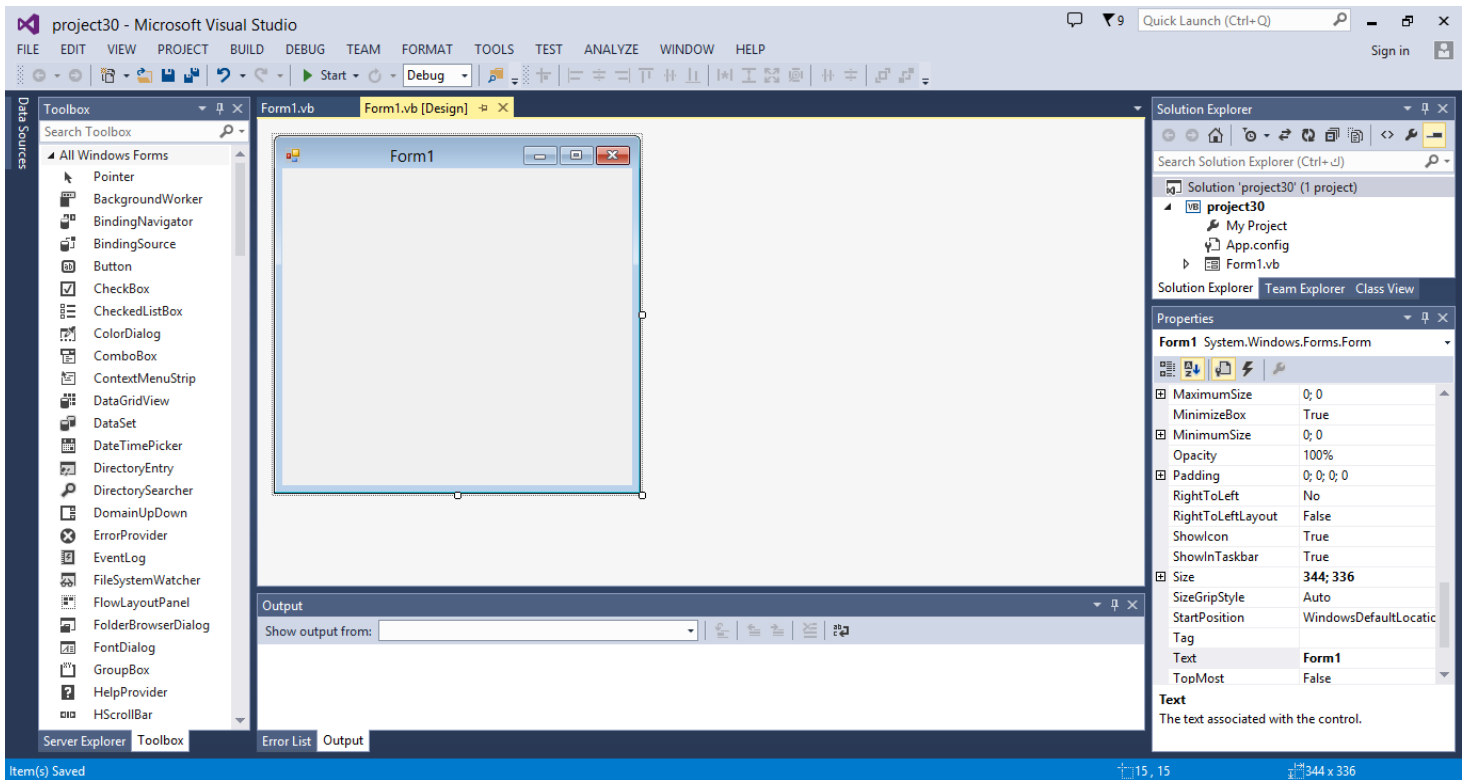


انشاء مشروع جديد

لإنشاء مشروع جديد من النافذة أعلاه انقر على **New Project...**. ستظهر لك النافذة التالية قم بإختيار **Windows Forms Application** ثم حدد اسم لمشروعك (يفضل أن يكون الاسم مقارب لعمل المشروع) ثم حدد مكان حفظ المشروع وانقر على **OK**.



بظهور أول نموذج في نافذة الفيجوال بيسك هذا يعني أنك أكملت الخطوة الأولى في طريق عمل تطبيق باستخدام الفيجوال بيسك 2013، يمثل النموذج الارضية التي توضع عليها كل أدوات التحكم كما يمكن التحكم بخصائصها وخصائص الأدوات من خلال النوافذ الجانبية التي توفرها اللغة.



من النظرة الأولى على النافذة أعلاه للفيجوال بيسك نلاحظ انها مزدحمة بأدوات التحكم والنوافذ أن هذا الازدحام قد يسبب بعض الإرباك في باديء الأمر، يمكن تجاوزه من خلال فهم ودراسة كل أدوات التحكم والنوافذ، حيث سنجد ان الأمر أسهل مما نتصور وان استخدام هذه الأدوات يوفر علينا الكثير من الوقت والجهد في انشاء التطبيقات حيث سنجد الكثير من الحلول الجاهزة للمشاكل التي تواجهنا والتي تتطلب سطور برمجية كثيرة.

مكونات الواجهة الرئيسية للفيجوال بيسك

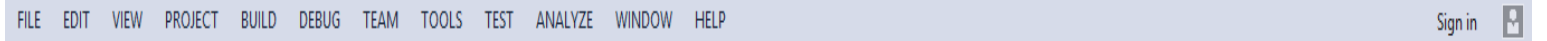
شريط العنوان Title Bar

يضم هذا الشريط اسم المشروع واسم اللغة بالإضافة الى أزرار التحكم بالنافذة، كما يحتوي على مربع البحث ورمز التبليغات.



شريط القوائم Menu Bar

يحتوي هذا الشريط على مجموعة من القوائم وكل قائمة تضم مجموعة من الأوامر وكل أمر يؤدي وظيفة معينة. اهم القوائم في هذا الشريط هي:



1- قائمة File: تحتوي هذه القائمة على مجموعة من الأوامر الخاصة بالتعامل مع المشاريع مثل New لإنشاء مشروع جديد، Open لفتح مشروع سابق، Add لإضافة مشروع جديد او ملف موجود مسبقاً، وحفظ المشاريع من خلال Save أو Save As أو All، بالإضافة الى امر الطباعة Print والخروج Exit.

2- قائمة Edit: تحتوي قائمة Edit في الفيجوال بيسك على أوامر مشابهة لما تحتويه قائمة Edit في التطبيقات الاخرى، حيث تحتوي على أوامر القص والنسخ واللصق وحذف النصوص وأدوات التحكم بالإضافة الى أوامر البحث Find والإستبدال Replace.

3- قائمة View: الهدف الأساسي من الأوامر الموجودة في هذه القائمة هو التحكم بظهور مختلف النوافذ في الفيجوال بيسك، كما يمكن من خلال هذه القائمة حذف عدة نوافذ من واجهة الفيجوال بيسك مما يسمح بتوسيع مساحة العمل كما يمكن استدعاء هذه النوافذ عند الحاجة اليها.

4- قائمة Project: تحتوي هذه القائمة على الأوامر الخاصة باضافة مكونات جديدة الى المشروع ك النماذج والوحدات النمطية وملفات اخرى بالإضافة الى عرض نافذة خصائص المشروع.

5- قائمة Build: تحتوي هذه القائمة على مجموعة من الأوامر الخاصة ببناء وتنظيف المشروع بالإضافة الى نشر المشروع وادارة مكونات المشروع.

6- قائمة Debug: تحتوي هذه القائمة على مجموعة من الأوامر الخاصة باختبار التطبيق، تقوم بإظهار النتائج ضمن نوافذ مخصصة في الفيجوال بيسك. كما يُمكن من خلال هذه القائمة تنفيذ التعليمات البرمجية خطوة خطوة، وهذا يساعد في مراقبة اداء عمل المشروع.

7- قائمة Tools: تحتوي هذه القائمة على أدوات متنوعة ومختلفة مثل الأمر Option الذي يفتح نافذة حوارية لتعديل خصائص بيئة التطوير .

8- قائمة Window: يتم من خلال هذه القائمة ادارة نوافذ واجهة الفيجوال بيسك.

9- قائمة Help: يُمكن من خلال هذه القائمة استدعاء نظام المساعدة الفوري للـ VB.

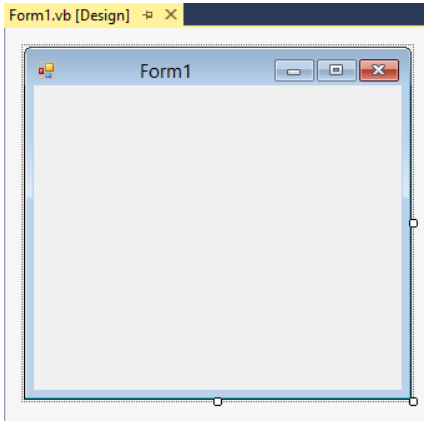
شريط الأدوات Tools Bar

يحتوي هذا الشريط على مجموعة من الصور المصغرة تمثل الأوامر الأكثر استخداماً في شريط القوائم. مثل حفظ وفتح المشاريع وإيقاف التطبيقات والبحث في الملفات وكذلك اوامر وضع اشارة التعليقات او ازالتها.



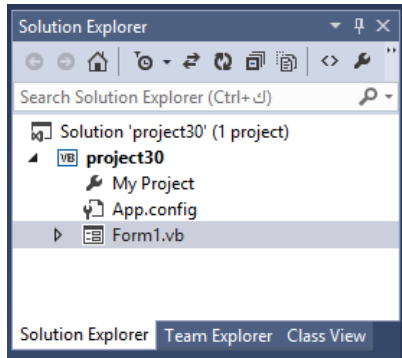
نافذة تصميم النموذج Form Designer

تحتوي هذه النافذة على النموذج الذي نقوم بتصميمه ووضع الأدوات عليه. يمكن أن يتكون المشروع من عدد كبير من النماذج، كما يمكن فتح نوافذ تصميم أكثر من نموذج بنفس الوقت.



متصفح المشروع Solution Explorer

تُظهر هذه النافذة محتويات التطبيق من نماذج ووحدات نمطية وغيرها من كائنات المشروع. تحوي هذه النافذة على شريط أدوات صغير يقع في الأعلى أهمها:



◆ **Properties**: تُظهر نافذة خصائص المشروع التي تحتوي على كثير من الخيارات.

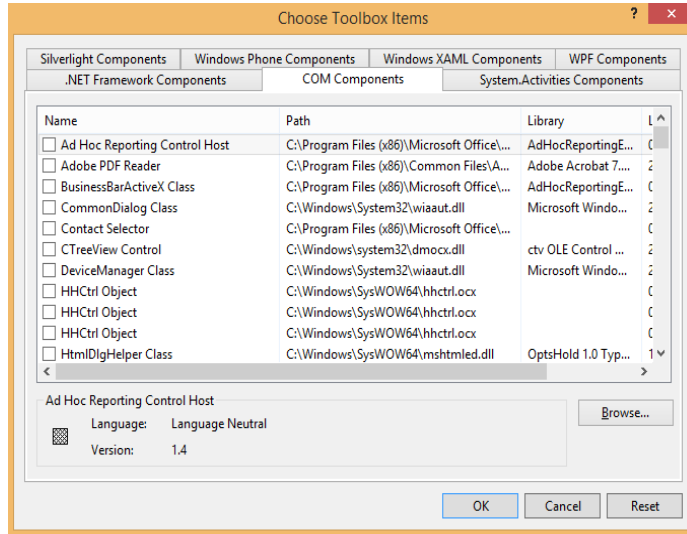
◆ **View Code**: تُظهر نافذة البرمجة للكائن المحدد.

◆ **Show All Files**: عرض كل ملفات المشروع.

صندوق الأدوات Tool Box

يحتوي هذا الصندوق على أدوات التحكم الأساسية التي يُمكن وضعها على النموذج في مرحلة التصميم، كما يُمكن إضافة أدوات تحكم أخرى والتي تعتبر نادرة الاستخدام من خلال النقر بزر الفأرة الأيمن على صندوق الأدوات ونختار الأمر Choose Items من القائمة المختصرة.

- List View
- Show All
- Choose Items...
- Sort Items Alphabetically
- Reset Toolbox



صندوق الأدوات، قام

لمصممي لغة الفيجوال بيسك بإنشاء قوائم عديدة ووضعوا فيها الأدوات التي تحمل نفس الغرض أو تتقارب في عملها. كما يمكن للمستخدم إنشاء قوائم جديدة وفق تسميات يرغب بها مع امكانية وضع مجموعة من أدوات التحكم فيها، في الحالة الافتراضية توجد عشرة قوائم تبدأ بـ All Windows Forms وتنتهي بـ WPF Interoperability. لإضافة قائمة جديدة الى صندوق

◆ لتنظيم توزيع أدوات التحكم في

Reset Toolbox

Add Tab

Delete Tab

Rename Tab

Move Up

Move Down

الأدوات، ننقر بزر الفأرة الأيمن فوق صندوق الأدوات ونختار الأمر (Add Tab) من القائمة المختصرة ثم نضع اسم لها ونضغط على مفتاح Enter فيقوم الفيجوال بيسك بإضافة قائمة فارغة الى صندوق الأدوات لا تحوي أي عنصر. يمكن نقل أداة واحد أو عدة أدوات من أدوات القوائم الأخرى الى القائمة الجديدة عن طريق السحب والإفلات. كما يمكن إضافة أدوات تحكم إضافية الى القائمة الجديدة.

◆ يمكن حذف القائمة التي تم إنشاؤها، عن طريق النقر بزر الفأرة الأيمن على اسم القائمة في صندوق الأدوات ثم إختيار الأمر (Delete Tab) من القائمة المختصرة.

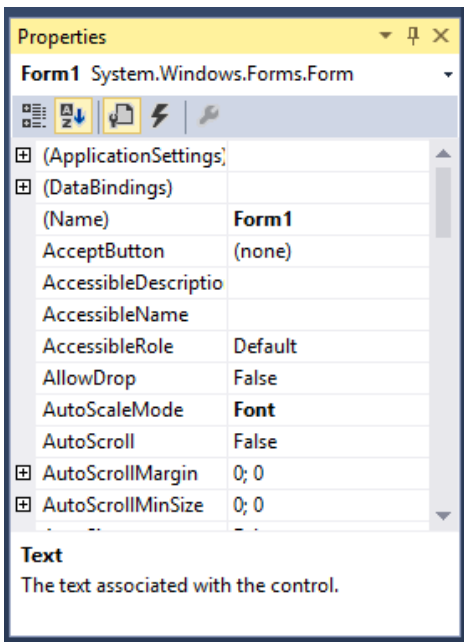
◆ يمكن إعادة تسمية القائمة الجديدة بإختيار الأمر (Rename Tab).

ملاحظة: لا يمكن حذف أو تغيير أسماء القوائم الأساسية في صندوق الأدوات.

نافذة البرمجة Code Windows

تستخدم هذه النافذة لكتابة التعليمات البرمجية التي تحدد سلوك واسلوب عمل النماذج وأدوات التحكم. يمكن في الفيجوال بيسك فتح نافذة برمجة لكل نموذج.

يُمكن فتح نافذة البرمجة بالنقر المزدوج على اي أداة تحكم في النموذج، أما بالنقر المزدوج على سطح النموذج. كما يُمكن فتح نافذة البرمجة من خلال النقر على الأيقونة View Code الموجودة في نافذة متصفح المشروع أو بالضغط على مفتاح F7.



sender As Object, e As EventArgs) Handles MyBase.Load

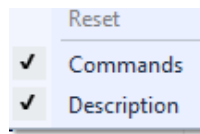
نافذ الخصائص Properties Windows

تحتوي هذه النافذة على مجموعة من الخصائص التي تُحدد مواصفات النموذج او الأدوات الموجودة على سطح النموذج، كل خاصية تقوم بتحديد صفة معينة للأداة كتغيير الحجم أو إخفاء وإظهار الأداة، عند الانتقال الى أداة اخرى ستظهر خصائص هذه الأداة في النافذة تلقائياً، يُمكن فتح نافذة خصائص أي أداة من خلال النقر على أداة التحكم بزر الفأرة الأيمن واختيار الأمر Properties أو من خلال الضغط على مفتاح F4.

تستخدم نافذة الخصائص بشكل كبير خلال عملية التطوير للتعامل مع الأدوات وضبط خصائصها. حيث تختلف الخصائص حسب نوع الأداة، إلا ان هنالك الكثير من الخصائص التي تكون مشتركة مع جميع أدوات التحكم.

كما يوجد في أعلى نافذة الخصائص صندوق قائمة يضم أسماء أدوات التحكم الموجودة على سطح النموذج، يُمكن اختيار اي أداة من القائمة لإستعراض خصائصها.

ويوجد أسفل نافذة الخصائص إطار التوصيف Description يحتوي على معلومات توضيحية عن وظيفة الخاصية المحددة، يُمكن إخفاء أو إظهار إطار التوصيف من خلال النقر بزر الفأرة الايمن على نافذة الخصائص وتحديد أو إلغاء تحديد الامر Description.



تحتوي نافذة الخصائص أسفل صندوق القائمة على أداة الأحداث مجموعة من الفئات الخاصة بالأحداث المتعلقة بالكائن المحدد.

يُمكن عرض الخصائص إما حسب الترتيب الأبجدي أو على شكل فئات حيث تُقسم الخصائص الى عدة فئات تختلف حسب نوع الكائن المحدد. أما بالنسبة لأهم فئات النموذج فتكون كالتالي:

- ◆ **Accessibility إمكانية الوصول:** تحوي هذه الفئة على مجموعة من الخصائص المتعلقة بطرق الوصول الى الكائن المحدد.
- ◆ **Appearance المظهر:** تحوي هذه الفئة على مجموعة من الخصائص المتعلقة بطرق الإظهار والشكل العام للكائن مثل اللون الخلفية ووضع صورة للخلفية وشكل المؤشر.
- ◆ **Font الخط:** تحتوي هذه الفئة على الخصائص المتعلقة بنمط الخط الذي سيظهر على الكائن ك(النوع والحجم والتأثيرات المتعلقة بالخط) بالإضافة الى شكل حدود النموذج واتجاه رسم الكائنات على النموذج بالإضافة الى النص الذي يظهر على النموذج.
- ◆ **Behavior السلوك:** تصف قائمة الخصائص الموجودة ضمن هذه الفئة طريقة عمل الكائن بالنسبة الى باقي الأدوات في التطبيق.
- ◆ **Data البيانات:** تحتوي هذه الفئة على عدة خصائص تحدد عمل ميزات مكونات ملف التطبيق والأغلفة البيانية لعنصر التحكم.

- ◆ **Design التصميم:** تحدد خصائص هذه الفئة الأسم البرمجي للنموذج واللغة بالاضافة الى توليد شفرات خاصة بالكائن والتحكم بتحريك او تغيير حجم الكائن.
- ◆ **Misc متفرقات:** ويقصد بها مواصفات مختلفة، حيث تحتوي هذه الفئة على مجموعة من الخصائص ذات تأثيرات متنوعة. أغلب الخصائص الموجودة في هذه الفئة كثيرة الاستخدام خلال تطوير التطبيق.

الفصل الثالث

عناصر التحكم الأساسية

محتويات الفصل

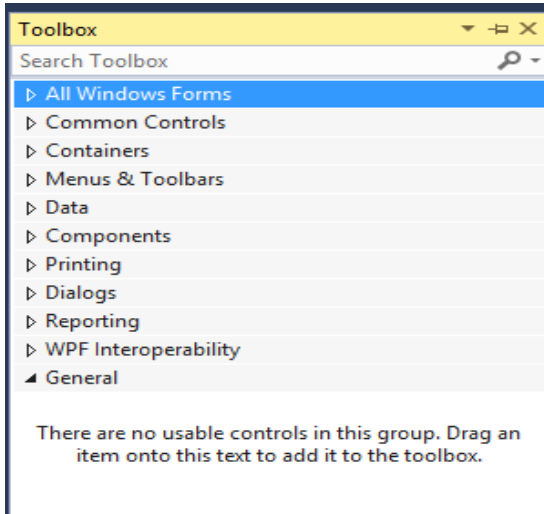
- عناصر التحكم الأساسية
- صندوق أدوات التحكم
- النموذج Form
- انشاء مشروع جديد
- إضافة أدوات التحكم الى نموذج
- تسمية أدوات التحكم
- المؤشر Pointer
- زر الأمر Button
- صندوق التحقق CheckBox
- قائمة صناديق التحقق CheckedListBox
- صندوق القائمة المنسدلة ComboBox
- انتقاء الوقت والتاريخ DateTimePicker
- الدلالة Label
- دلالة الارتباط LinkLabel
- صندوق القائمة LsitBox
- قائمة العرض ListView
- صندوق النص المقنع MaskedTextBox
- الروزنامة MonthCalender
- رمز الإعلام NotifiIcon
- الزيادة والنقصان العددي NumericUpDown
- صندوق الصورة PictureBox
- شريط التقدم ProgressBar
- زر الراديو RadioButton
- صندوق النص الغني RichTextBox
- صندوق النص TextBox
- أداة التلميح ToolTip
- أداة الشجرة TreeView
- متصفح النت WebBrowser
- لوح التخطيط الإنسيابي FlowLayoutPanel
- صندوق المجموعة GorupBox
- اللوح Panel

SplitContainer	فصل المحتويات	+
TabControl	التبويب	+
TableLayoutPanel	لوح التخطيط المسطح	+
ContextMenuStrip	أداة القائمة المختصرة	+
MenuStrip	أداة شريط القوائم	+
StatusBar	شريط الحالة	+
ToolStripContainer	وعاء شريط الأدوات	+
ErrorProvider	مزود الأخطاء	+
ImageList	أداة قائمة الصور	+
PrintDocument	أداة طباعة المستند	+
PrintPreviewDialog	أداة معاينة قبل الطباعة	+
PrintDialog	مربع حوار الطباعة	+
ColorDialog	مربع حوار الألوان	+
FontDialog	مربع حوار الخط	+
OpenFileDialog	أداة فتح الملفات	+
SaveFileDialog	أداة حفظ الملفات	+
Chart	المخططات	+

نعتمد في الفيجوال بيسك بشكل أساسي على أدوات التحكم التي نضعها على نماذج المشروع، تتضمن الفيجوال بيسك عدداً من أدوات التحكم الأساسية والاضافية وفي الاحيان عندما يكون المشروع معقداً نوعاً ما او يحتاج الى اسلوب احترافي يمكن تحميل أدوات التحكم من الانترنت او الشراء من شركات خاصة تقوم بتطوير وانشاء أدوات تحكم جديدة. ومن ثم اضافة هذه الأدوات الى المشروع والاستفادة منها ومن خصائصها.

صندوق ادوات التحكم

ادوات يستخدمها المبرمج لتصميم النموذج وتقوم هذه الأدوات بالوظائف المطلوبة من البرنامج، ولكل اداة خصائص واحداث يتم برمجتها لتقوم بغرض معين ولكل اداة غرض تؤديه يختلف عن البقية، ولكل اداة نافذة خصائص يتم من خلالها التحكم بمميزاتها وكذلك يمكن برمجة خصائص الادوات باستخدام الشفرات البرمجية.

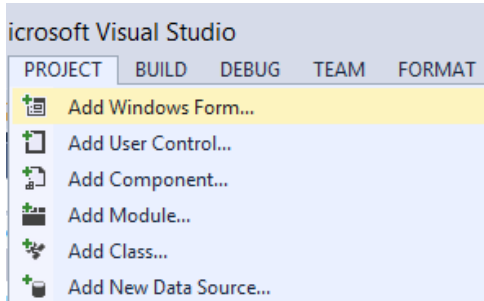


تقسم صناديق أدوات التحكم الى مجاميع وكل مجموعة تضم عدة ادوات.

يتوفر في لغة الفيجوال بيسك 2013 مربع بحث يقع في أعلى الواجهة يُمكنك كتابة اسم اي اداة فيه ليتم عرض مكان وجودها مما يسهل عليك ايجادها و اضافتها للمشروع او اظهارها ضمن واجهة اللغة.

النموذج Form:

هي واجهة التطبيق وهي بمثابة الوعاء الذي توضع عليه كل الكائنات الموجودة في صندوق ادوات الفيجوال بيسك ويظهر تلقائياً عند انشاء مشروع جديد ويمكن ان يحتوي التطبيق على نموذج واحد او عدة نماذج.



لا تظهر النماذج ضمن صندوق أدوات الفيجوال بيسك بل يمكن اضافتها من خلال شريط القوائم **Project** → **AddWindowsForm**

أهم خصائص النموذج هي:

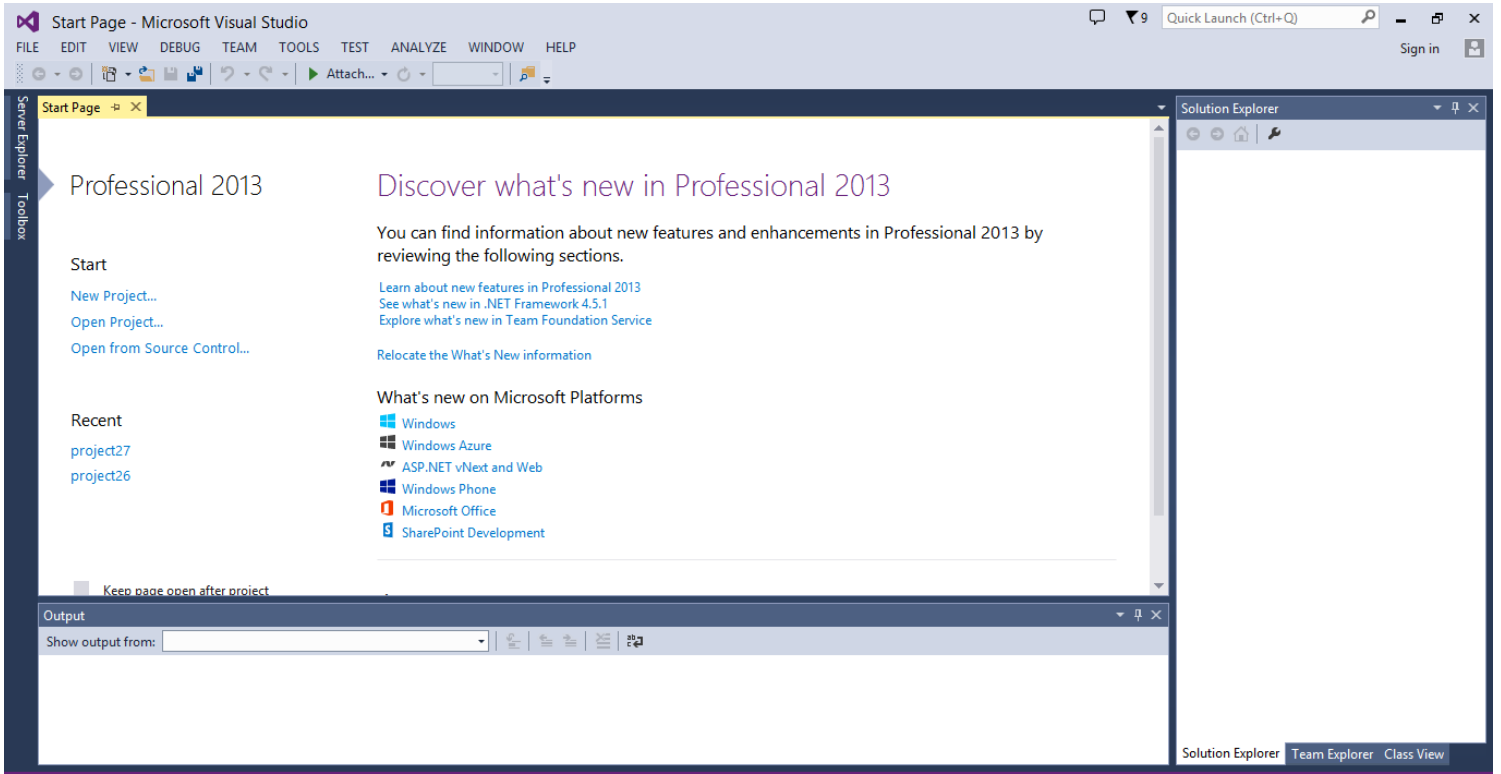
- ◆ **Name**: هو اسم النموذج الذي نتعامل معه برمجياً
- ◆ **BackColor**: لتغيير لون خلفية النموذج وهناك ثلاثة انواع من الالوان:
- **System**: تتأثر الخلفية بسميز الويندوز.
- **Web**: هي ألوان الفيجوال بيسك ولا توجد في فيجوال بيسك 6.
- **Custom**: لتخصيص درجة لون معين.
- ◆ **BackgroundImage**: وضع صورة كخلفية للنموذج.
- ◆ **BackgroundImageLayout**: وتأخذ القيم التالية:
- **None**: عرض الصورة من أول النموذج الى نهايته.
- **Tile**: عرض الصورة بحجمها الطبيعي اذا كانت اقل من حجم النموذج يتم تكرار الصورة.
- **Stretch**: عرض الصورة على كامل النموذج.
- **Center**: عرض الصورة بحجمها الطبيعي في منتصف النموذج.
- **Zoom**: وضع الصورة على كامل النموذج وبنسب متفاوتة في الطول والعرض.

- ◆ **Cursor**: تغيير مؤشر الفأرة على النموذج.
- ◆ **Font**: تغيير حجم ونمط ونوع الخط.
- ◆ **ForeColor**: تغيير لون خط النموذج.
- ◆ **FormBorderStyle**: تحديد الاطار الخارجي للنموذج ويأخذ القيم التالية:
 - **None**: حذف شريط عنوان النموذج.
 - **FixedSingle**: منع المستخدم من توسيع او تضيق اطار النموذج.
 - **Fixed3D**: جعل النموذج تاخذ شكل ثلاثي الابعاد.
 - **FixedDialog**: تحجم النموذج.
 - **Sizable**: جعل اطار النموذج حر بحيث يمكن للمستخدم من توسعته او تضيقه.
 - **FixedToolWindows**: تحجم النموذج مع حذف مفاتيح التحكم بالنموذج والابقاء على مفتاح الإغلاق فقط.
 - **SizableToolWindows**: التحكم بتوسعة وتضيق النموذج وحذف مفاتيح التحكم بها والابقاء على مفتاح الاغلاق فقط.
- ◆ **RightToLeft**: تحدد محاذاة الكائنات على سطح النموذج وتأخذ القيم التالية:
 - **Yes**: المحاذاة من اليمين الى اليسار.
 - **No**: المحاذاة من اليسار الى اليمين.
- ◆ **Text**: وضع عنوان للنموذج.
- ◆ **Size**: لتغيير حجم النموذج.
- ◆ **AllowDrop**: لتمكين السحب والافلات للنموذج.
- ◆ **ContextMenuStrip**: لانشاء قائمة زر ايمن للفأرة باستخدام الاداة **ContextMenuStrip**.
- ◆ **Enable**: تستخدم لتمكين او تعطيل النموذج.
- ◆ **Locked**: لمنع التعديل على النموذج وقت التصميم.
- ◆ **Icon**: لاختيار ايقونة للنموذج.
- ◆ **ShowIcon**: تأخذ القيم التالية:
 - **True**: اظهار ايقونة النموذج في شريط عنوانها.
 - **False**: اخفاء ايقونة النموذج من شريط عنوانها.
- ◆ **ShowInTaskbar**: تاخذ القيم التالية:
 - **True**: اظهار النموذج في شريط مهام نظام Windows اثناء التنفيذ.
 - **False**: اخفاء النموذج من شريط المهام.
- ◆ **StartPosition**: تُحدد موقع ظهور النموذج.
- ◆ **MaximizeBox**: لإظهار او إخفاء مفتاح التكبير.
- ◆ **MinimizeBox**: لإظهار او إخفاء مفتاح التصغير.
- ◆ **Opacity**: للتحكم في درجة شفافية النموذج.
- ◆ **TopMost**: لاطهار النموذج أعلى النوافذ المفتوحة.
- ◆ **WindowState**: وضع ظهور النموذج.
 - **Normal**: الوضع الطبيعي للنموذج على الشاشة.
 - **Minimized**: يظهر النموذج في شريط المهام وليس على الشاشة.
 - **Maximized**: جعل النموذج يظهر ويحتل كامل الشاشة.
- ◆ **IsMdiContainer**: تصبح النموذج حاوية للنماذج الاخرى بداخلها.

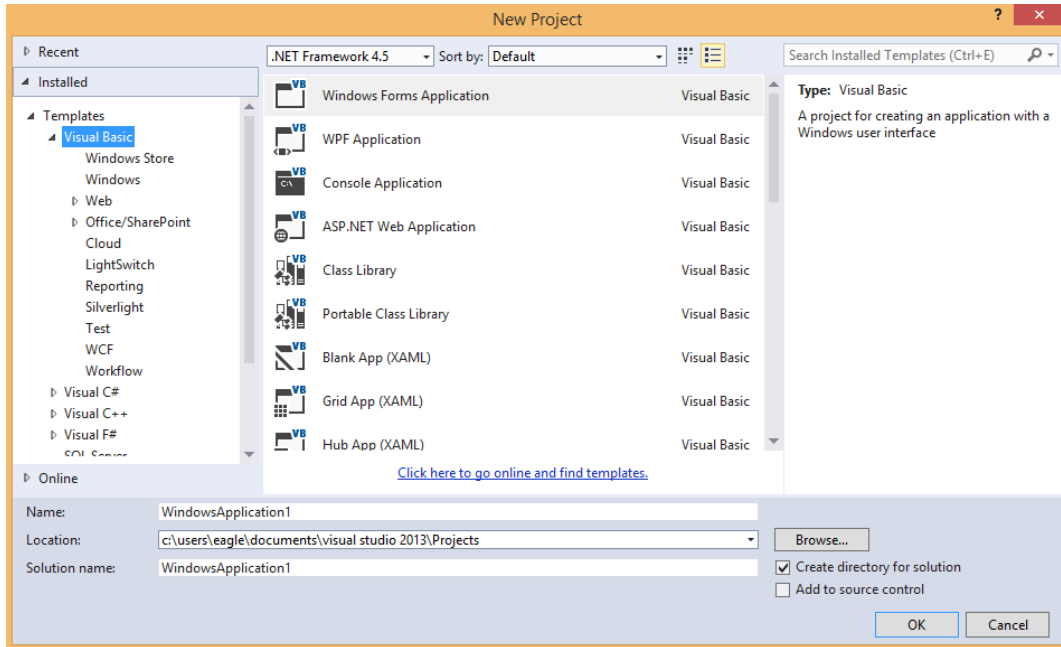
كما ان أغلب هذه الصفات هي مشتركة بين النموذج والأدوات الموجودة في شريط الأدوات.

انشاء مشروع جديد

بعد تشغيل الفيجوال بيسك تظهر نافذة نختار منها الامر ... New Project... لإنشاء مشروع جديد

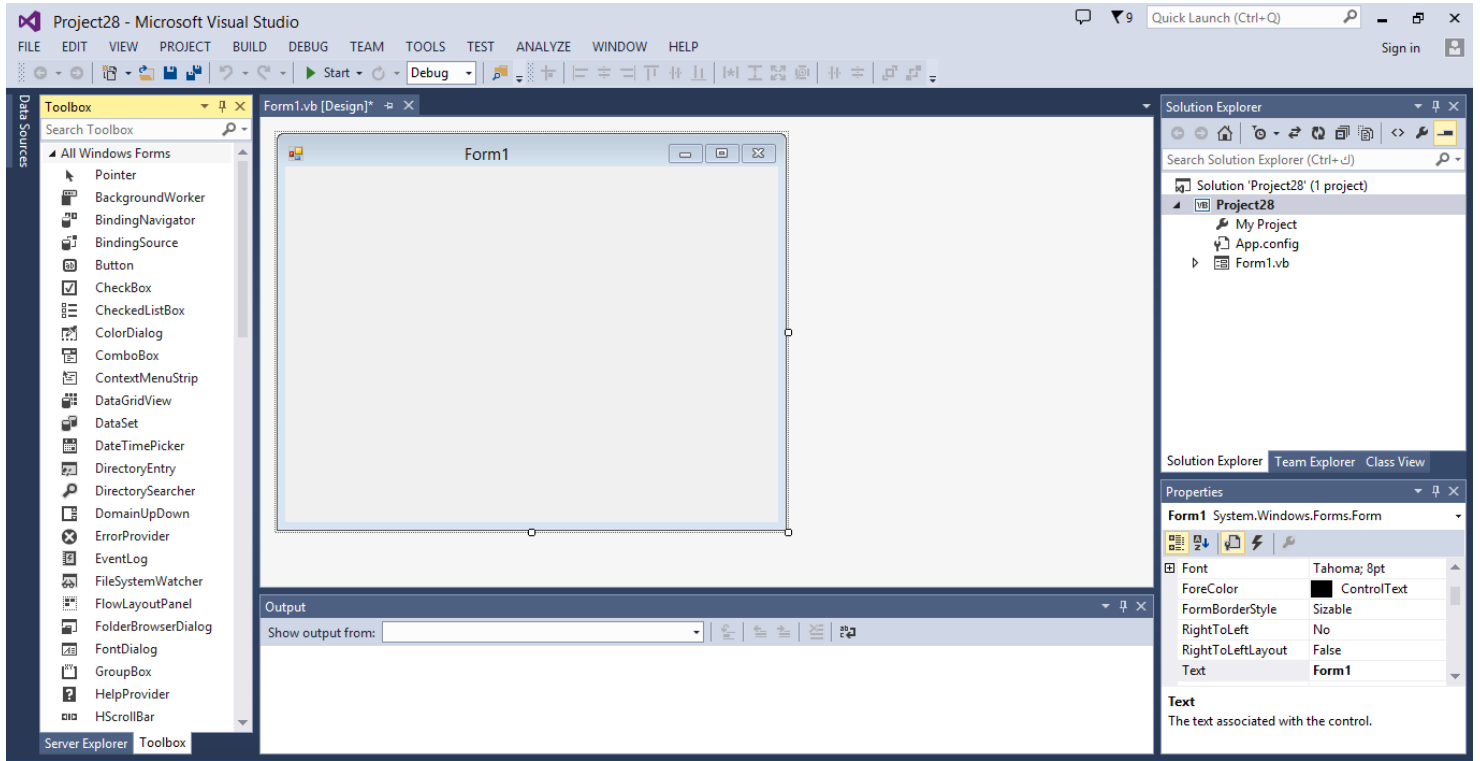


ستظهر النافذة التالية يتم من خلالها تحديد اسم المشروع ومكان خزنه في الحاسوب.



ملاحظة: تمتاز الفيجوال بيسك بإمكانية تجريب التطبيق وتنفيذه دون الخروج من واجهة اللغة مع التركيز على الاختلاف بين مرحلة تصميم المشروع ومرحلة تنفيذه.

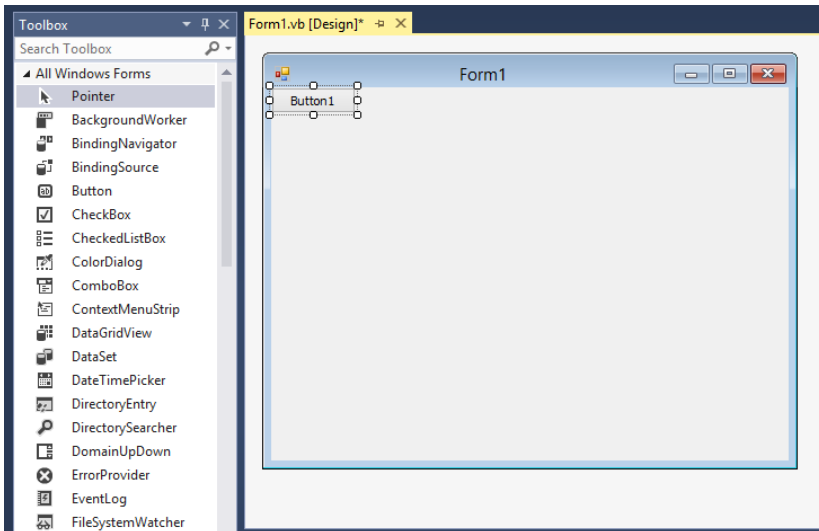
مرحلة التصميم Design: في هذه المرحلة يتم اضافة أدوات التحكم الى النماذج واسناد الخصائص لها وكتابة الاكواد البرمجية. مرحلة التنفيذ Run: في هذه المرحلة يتم اظهار الأدوات التي أضيفت مرحلة التصميم على الشاشة كما ستظهر للمستخدم ولا يمكن في هذه المرحلة التغيير او التعديل على التصميم.



اضافة أداة تحكم الى نموذج:

يتم اضافة الاداة الى النموذج بأحد الطريقتين:

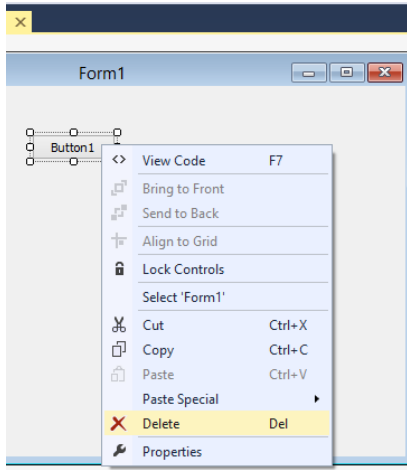
■ النقر المزدوج على الأداة التي نريد اضافتها والموجودة في صندوق الادوات، سيقوم الفيجوال بيسك بوضع الأداة في أعلى يسار النموذج.



■ نختار الأداة التي نريد اضافتها الى النموذج من صندوق الادوات بالنقر مرة واحدة على ايقونة الأداة ثم ننقر فوق النموذج في المكان الذي نريد وضع الأداة فيه ونسحب بالفأرة الى أن تأخذ الأداة حجمها الطبيعي.

بعد اضافة الأداة يمكن تغيير موقعها وحجمها وتحريكها باستخدام الفأرة.

- لحذف اداة تحكم من سطح النموذج نحدد الأداة ثم نضغط على المفتاح Delete من لوحة المفاتيح، او ننقر بزر الفأرة الايمن على الأداة التي نريد حذفها ثم نختار الأمر Delete من القائمة المختصرة التي ستظهر.



تسمية أدوات التحكم

عند اضافة أي أداة الى سطح النموذج يقوم الفيجوال ببيسك باعطائها اسم افتراضي، مثلاً لو اضعنا عنصري TextBox فان الاول سيحمل اسم TextBox1 والثاني TextBox2، يُفضّل تجنب الإتماد على الاسماء الافتراضية التي يضعها الفيجوال ببسك ، وتعتبر هذه الطرق مربكة للعمل وتؤدي الى حدوث اخطاء في المشاريع الضخمة التي تستخدم عدد كبير من أدوات التحكم، ويُفضّل اختيار اسم للأداة تكون بادئة الاسم تدل على نوع الأداة ولاحقة تدل على الهدف من الأداة فاذا أردنا اضافة صندوق نص يحوي على اسم الطالب فالاسم سيكون (TxtStudName) حيث ان البادئة Txt تدل على نوع الأداة وهو TextBox واللاحقة StudName تدل على ان صندوق النص مخصص لوضع اسم الطالب فيه.

ويُفضّل ان يكون اسم أداة التحكم صغير حتى يسهل التعامل معه برمجياً.

من فوائد طريقة التسمية السابقة حيث يمكن التعديل على البرنامج بأقل وقت وممكن وكذلك سهولة تذكر الاسماء لان التسميات العشوائية تجعل الأمور أكثر تعقيداً، وكذلك يمكن الاستفادة من طريقة التسمية السابقة عند العمل ضمن فريق مؤلف من عدة مبرمجين.

فعند اضافة أداة وتسميتها بالطريقة الصحيحة سيُسهل على باقي المبرمجين معرفة الأداة الجديدة وسهولة التعامل معها وكذلك عند انضمام شخص جديد الى فريق العمل لن يحتاج الى فترة طويلة للتأقلم مع اسلوب عمل الطاقم.

وفي يلي جدول يُبين البادئة المقترحة لأغلب أدوات التحكم:

الكائن	البادئة
Button	Btn
CheckBox	chk
CheckListBox	Chklst
ColorDialog	ClrDlg
ComboBox	Cbo
DataGridView	grd
DataSet	DtSt
FontDialog	FntDlg
GroupBox	Grp
ImageList	Imglst
Label	Lbl

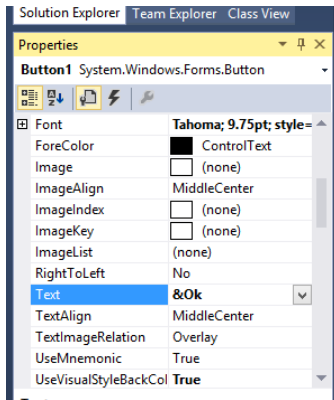
LinkLabel	LnkLbl
ListBox	Lst
ListView	Lstv
OpenFileDialog	OpnFDlg
Panel	Pnl
PictureBox	Pic
ProgressBar	Prg
Textbox	Txt
Timer	Tmr
TreeView	TrV

المؤشر Pointer

هي الأداة الوحيدة في صندوق الأدوات التي لا تُضاف ولا تُرسم على النموذج. وعندما يكون المؤشر محدداً نستطيع تغيير حجم وتحريك أدوات التحكم الموجودة على النموذج، كما يُمكن تحديد عدة أدوات تحكم بنفس الوقت وضبط الخصائص المشتركة لهم.

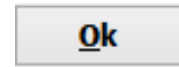
زر الأمر Button

يستخدم زر الامر بشكل أساسي لتنفيذ تعليمات برمجية عند النقر عليه ومن أهم خصائص زر الأمر:

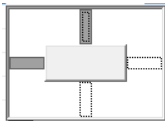


◆ **Text:** وتمثل العنوان الذي سيظهر على زر الأمر، يمكن تحديد حرف نشط في العنوان اي يُسطر تحت الحرف ويُمكن للمستخدم في مرحلة التنفيذ استخدام مفتاح +ALT+الحرف المسطر لتنفيذ التعليمات داخل زر الأمر بدل النقر عليه بالفأرة.

وللتسطير تحت الحرف ضع الرمز (&) قبل الحرف في مثالنا هذا كتبنا &OK:



سيكون شكل زر الأمر



◆ **Anchor:** ولهذه الخاصية اربع اتجاهات وعند الضغط على كل اتجاه يتم تثبيت المسافة من ذلك الاتجاه عند التكبير والتصغير.

◆ **AutoSize:** التحكم بحجم زر الامر تلقائياً وتأخذ هذه الخاصية القيم التالية:

– **True:** تفعيل الخاصية حيث يزداد حجم الأداة بزيادة حجم الخط ويقل حجمها عند نقصان حجم الخط.

– **False:** الغاء تفعيل تغيير حجم الأداة تلقائياً.

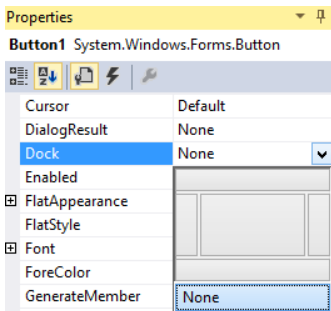
◆ **Dock:** وضع زر الامر في احد جوانب النموذج ولا يتحرك من مكانه عند تكبير او تصغير النموذج.

◆ **Image:** وضع صورة على زر الامر ليس كخلفية بل على جزء منه.

◆ **ImageAlign:** تحديد مكان الصورة.

◆ **Location:** تحديد موقع زر الأمر على النموذج عن طريق وضع قيم في الاحداثي X والاحداثي Y.

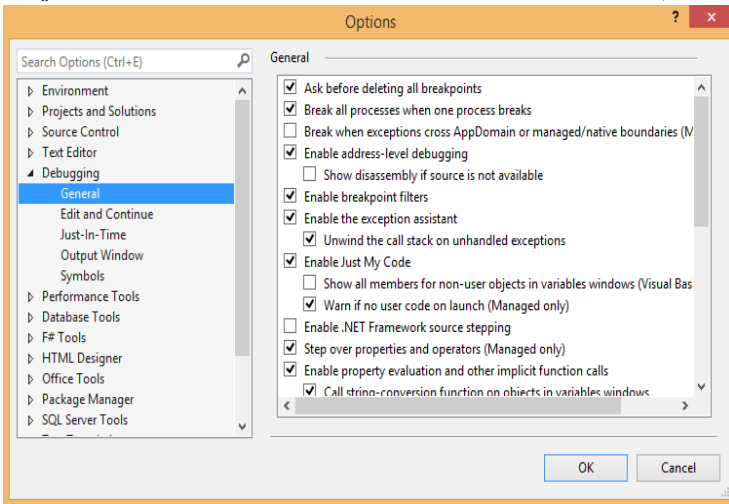
◆ **TabIndex:** ترتيب زر الامر في الامر Tab.



- ◆ **Margin**: تحديد المسافة بين زر الأمر واي عنصر تحكم اخر.
- ◆ **FlatStyle**: اختيار الشكل المطلوب استخدامه لزر الامر.
- ◆ **UseMnemonic**: لجعل اول حرف من زر الامر يعمل كإختصار مفاتيح.
- ◆ **Visible**: تاخذ القيم التالية:
 - **True**: لظهار زر الامر على سطح النموذج خلال التنفيذ.
 - **False**: اخفاء زر الأمر من سطح النموذج.
- ◆ **TextImageRelation**: تحدد مكان الصورة مع النص وتاخذ القيم التالية:
 - **Overlay**: لجعل الصورة والنص بنفس المستوى.
 - **ImageAboveText**: لجعل الصورة فوق النص.
 - **TextAboveImage**: لجعل النص فوق الصورة.
 - **ImageBeforeText**: لجعل الصورة قبل النص.
 - **TextBeforeImage**: لجعل النص قبل الصورة.
- ◆ **MaximumSize**: وضع اكبر حجم لزر الأمر.
- ◆ **MinimumSize**: وضع اصغر حجم لزر الأمر.

صندوق التحقق CheckBox

يستخدم لعرض مجموعة من الخيارات امام المستخدم، ويمكن للمستخدم ان يختار واحدة منها او اكثر حسب الضرورة كما في النافذة التالية:

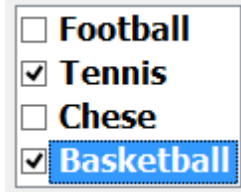


اهم خصائص صندوق التحقق هي:

- ◆ **CheckAlign**: يستخدم لتحديد مكان مربع الاختيار داخل الأداة.
- ◆ **Checked**: تأخذ هذه الخاصية القيم التالية:
 - **True**: اذا كانت الأداة محددة.
 - **False**: اذا كانت الأداة غير محددة.
- ◆ **CheckedState**: وتأخذ هذه الخاصية القيم التالية:
 - **Checked**: لوضع علامة صح داخل مربع الخاص بالصندوق.
 - **Unchecked**: لإلغاء علامة الصح من مربع الصندوق.
 - **Indeterminate**: لوضع علامة على شكل مربع بدل علامة الصح داخل مربع الصندوق.
- ◆ **ThreeState**: تتحكم بالحالة الثالثة للصندوق والمتمثلة بـ **Indeterminate** وتأخذ هذه الخاصية القيم التالية:
 - **True**: لتنشيط الحالة الثالثة للصندوق.
 - **False**: لإلغاء تنشيط الحالة الثالثة للصندوق.

قائمة صناديق التحقق **CheckedListBox**

كالأداة السابقة مع توفر ميزة اضافة مجموعة من صناديق التحقق في أداة واحدة وتكمن اهميتها في تحديد هوايات او رغبات لاشخاص معينين.

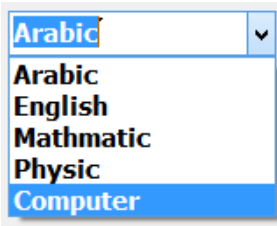


اهم خصائص الأداة هي:

- ◆ **Items**: وتخص العناصر التي سيتم وضعها داخل هذه الاداة حيث ان كل كلمة او عدة كلمات يتم وضعها في سطر واحد تعتبر عنصر واحد داخل القائمة.
- ◆ **CheckOnClick**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تحديد الصندوق عند النقر عليه بالفأرة مرة واحدة.
 - **False**: لا يتم تحديد الصندوق عند النقر عليه بالفأرة مرة واحدة.
- ◆ **HorizontalScrollBar**: عمل شريط تمرير لاطهار جميع العناصر الموجودة ضمن القائمة.
- ◆ **MultiColumn**: تأخذ هذه الخاصية القيم التالية:
 - **True**: عرض العناصر في اكثر من عمود داخل الأداة.
 - **False**: عرض العناصر في عمود واحد داخل الأداة.
- ◆ **Sorted**: تأخذ هذه الخاصية القيم التالية:
 - **True**: فرز عناصر القائمة أبجدياً.
 - **False**: ابقاء عناصر القائمة دون فرز.

صندوق القائمة المنسدلة **ComboBox**

هي اداة مرنة وفعالة تجمع بين صندوق النص وصندوق القائمة، حيث يمكن للمستخدم ان يكتب فيها او يختار عنصر من عناصرها. كما في النافذة المجاورة.



اهم خصائص صندوق القائمة المنسدلة هي:

- ◆ **Items**: تستخدم لاضافة عنصر او مجموعة عناصر للقائمة.
- ◆ **MaxDropDownItems**: الحد الأعلى لعدد العناصر التي ستظهر داخل القائمة.
- ◆ **DropDownStyle**: تستخدم لتغيير شكل الاداة وتأخذ القيم التالية:
 - **DropDownList**: تمكن من تحرير النص في خانة النص.
 - **Simple**: لا تمكن من تحرير النص داخل خانة النص.
 - **DropDown**: تمكن من تحرير النص مع إظهار العناصر بشكل مبدئي.
- ◆ **DropDownHeight**: تستخدم لوضع ارتفاع القائمة.
- ◆ **DropDownWeight**: تستخدم لوضع عرض القائمة.
- ◆ **Sorted**: تأخذ هذه الخاصية القيم التالية:
 - **True**: فرز عناصر القائمة المنسدلة.
 - **False**: إبقاء عناصر القائمة بلا فرز.

- ◆ **AutoCompleteMode**: النص الذي على أساسه يُقترح جزء من عناصر القائمة وتأخذ القيم التالية:
 - **None**: لن تظهر ميزة اكمال النص تلقائياً.
 - **Suggest**: تظهر اقتراح لعملية اكمال النص تلقائياً.
 - **Append**: تظهر تكملة النص تلقائياً.
 - **SuggestAppend**: تظهر اقتراح وتكملة النص تلقائياً.
- ◆ **AutoCompleteSource**: مصدر الاكمال التلقائي للبيانات.

يمكن التعامل مع الاداة برمجياً من خلال الطرق التالية:

- ◆ **Add**: اضافة عنصر جديد للقائمة.
- ◆ **Clear**: افرغ القائمة من العناصر.
- ◆ **Remove**: حذف عنصر من القائمة.
- ◆ **RemoveAt**: حذف عنصر يحمل ترتيب معين.
- ◆ **Count**: يظهر عدد جميع عناصر القائمة.
- ◆ **IndexOf**: لمعرفة ترتيب عنصر داخل القائمة.

ComboBox1.Items.Add (TextBox1.text)

ComboBox1.Items.Clear

ComboBox1.Items.Remove(ComboBox1.SelectedItem)

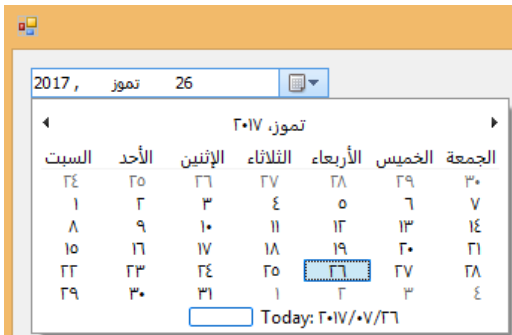
ComboBox1.Items.RemoveAt(ComboBox1.SelectedIndex)

Label1.Text = ComboBox1.Items.Count

MsgBox(ComboBox1.Items.IndexOf(TextBox1.Text))

انتقاء الوقت والتاريخ DateTimePicker

تستخدم هذه الاداة لإختيار التاريخ بدل كتابته يدوياً وتساعد في تجنب الاخطاء الناتجة عن تنسيق التاريخ.



اهم خصائص هذا الكائن هي:

- ◆ **Format**: تستخدم في تحديد تنسيق التاريخ المعروف في داخل هذه الاداة ومن انواعه:
 - **Long**: كتابة التاريخ بدل وضع الارقام مكانه.
 - **Short**: يظهر التاريخ على شكل ارقام.

- **Time**: تظهر الوقت فقط.
- **Custom**: تظهر ما يتم تخصيصه في CustomFormat.
- ◆ **CustomFormat**: تحديد تنسيق التاريخ وتأخذ عدة أشكال (MM/dd/yyyy)، (dd-MM-yyyy)، (hh/mm/ss).
- ◆ **MinDate**: تحديد أدنى تاريخ.
- ◆ **MaxDate**: تحديد أقصى تاريخ.
- ◆ **ShowUpDown**: تستخدم لتغيير التاريخ عن طريق النقر على السهم الجانبي (السهم للأعلى زيادة ، السهم للأسفل نقصان).
- ◆ **Value**: قيمة التاريخ المطلوب عرضها.
- ◆ **ShowCheckBox**: تظهر مربع التحقق بجوار التاريخ الظاهر وهي ترتبط بخاصية Checked.
- ◆ **Checked**: ترتبط بالخاصية السابقة وتظهر علامة الصح داخل المربع.

ملاحظة **MinDate** و **MaxDate** يمكن لهذه الخاصيتين اختيار وقت لإنهاء البرنامج وكما هو معلوم لا يوجد برنامج لا يعتمد على التاريخ. فتكمن أهميتهما في حماية البرامج.

التعامل مع الأداة برمجياً

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If DateTimePicker1.Checked = True Then
        TextBox1.Text = DateTimePicker1.Value
    Else
        TextBox1.Text = ""
    End If
End Sub
```

ارجاع رقم اليوم من الاسبوع

```
Label1.Text = DateTimePicker1.Value.DayOfWeek
```

او ارجاع رقم اليوم من السنة

```
Label1.Text = DateTimePicker1.Value.DayOfYear
```

الدلالة Label

تستخدم لظهور نص أمام المستخدم دون القدرة على تغييره أو تعديله، حيث تستخدم كعنوان لأداة تحكم أخرى مثل TextBox و ComboBox في المثال التالي نستخدم اثنين من الأداة Label الاول للدلالة على ان صندوق النص خاص باسم الطالب، والثاني للدلالة على ان صندوق القائمة يحوي الصفوف الدراسية.

اهم خصائص الدلالة هي:

- ◆ **Text**: تحوي النص الذي تظهره التسمية.
- ◆ **Visible**: وتأخذ هذه الخاصية احدى القيم:
 - **True**: لإظهار الأداة على سطح النموذج خلال التنفيذ.
 - **False**: لإخفاء الأداة من سطح النموذج عند التنفيذ.

دلالة الارتباط LinkLabel

تشبه الى كبير اداة Label مع توفير الكثير من المميزات الاخرى كتصفح مواقع الويب او الانتقال الى نموذج اخر.

www.google.com
www.yahoo.com
www.YouTube.com

اهم خصائص هذا الكائن هي:

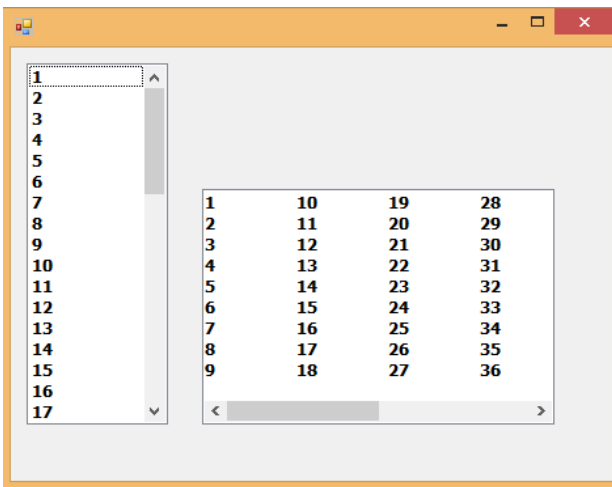
- ◆ **ActiveLinkColor**: تحديد اللون عند تنشيط الرابط عند الضغط عليه فقط.
- ◆ **DisabledLinkColor**: تحديد اللون المستخدم عند وقف الرابط.
- ◆ **LinkArea**: تحديد نطاق النص المستخدم الرابط من Start و Length.
- ◆ **LinkBehavior**: تحديد قيمة سلوك الرابط ولها عدة قيم.
- ◆ **LinkColor**: تحديد لون الرابط الافتراضي عند ظهوره.
- ◆ **LinkVisited**: تُحدد تنشيط او إلغاء لون الرابط عند زيارته.
- ◆ **VisitedLinkColor**: تحدد لون الرابط الذي تمت زيارته سابقاً.

التعامل مع الاداة برمجياً

```
Private Sub LinkLabel2_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs)
Handles LinkLabel2.LinkClicked
    System.Diagnostics.Process.Start("www.google.com")
End Sub
```

صندوق القائمة ListBox

تقوم هذه الأداة بعرض قائمة من العناصر (Items) أمام المستخدم، مع امكانية فرز هذه العناصر او الاضافة عليها او تعديلها وكذلك حذفها بالاضافة الى امكانية عرض هذه العناصر في عمود واحد او اكثر كما في الشكل المجاور:



اهم خصائص صندوق القائمة هي:

- ◆ **Items**: اضافة عناصر الى القائمة.
- ◆ **MultiColumn**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: عرض عناصر القائمة في أكثر من عمود.
 - **False**: عرض عناصر القائمة في عمود واحد.
- ◆ **ColumnWidth**: تحديد عرض العمود (تحديد المسافة بين الاعمدة داخل القائمة).
- ◆ **Stored**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: ترتيب عناصر القائمة أبجدياً.
 - **False**: ترك عناصر القائمة بلا ترتيب.

◆ **HorizontalScrollbar**: اظهار شريط التمرير الافقي للقائمة.

◆ **ScrollAlwaysVisible**: اظهار اشطرة التمرير دائماً.

◆ **SelectionMode**: تحديد عدد العناصر داخل القائمة.

التعامل مع صندوق القائمة برمجياً

لاضافة عنصر جديد

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    For i As Integer = 1 To 50
        ListBox1.Items.Add(i)
    Next
End Sub
```

حذف عنصر محدد من قائمة

```
Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles ListBox1.SelectedIndexChanged
    ListBox1.Items.Remove(ListBox1.SelectedItem)
End Sub
```

```
Private Sub btndlt_Click(sender As Object, e As EventArgs) Handles btndlt.Click
    If ListBox2.SelectedIndex = -1 Then
        MsgBox("القائمة من عنصر اختر")
    Else
        ListBox2.Items.RemoveAt(ListBox2.SelectedIndex)
    End If
End Sub
```

قائمة العرض ListView

تستخدم لعرض مجموعة من العناصر بطرق مختلفة:

اهم خصائص هذه الاداة هي:

◆ **Alignment**: تحدد محاذاة الأداة.

◆ **AllowColumnReorder**: تتيح للمستخدم امكانية تغير عمود من عدمه.

◆ **CheckBoxes**: لعرض ايقونه صندوق التحقق قبل العنصر.

◆ **Columns**: تستخدم لاضافة او حذف عمود وتستخدم في حالة Details=View

◆ **Items**: اضافة او حذف العناصر.

◆ **Groups**: اضافة او حذف مجموعة عناصر.

◆ **FullRowSelect**: اختيار جميع عناصر القائمة.

◆ **LabelEdit**: تسمح بتعديل عناوين العناصر.

◆ **View**: تسمح بتحديد الشكل الذي تعرض به العناصر.

◆ **Small&LargeImageList**: التحكم بطريقة عرض الصور داخل الاداة.

صندوق النص المقنع MaskedTextBox

هي اداة تشبه صندوق النص في الشكل الى انها تختلف عنه بالعمل وتستخدم لعمل قناع معين لإجبار المستخدم على استخدام نوع محدد من البيانات.

اهم خصائص صندوق النص المقنع هي:

- ◆ **AsciiOnly**: تحدد القيم المدخلة بحروف.
- ◆ **BeepOnError**: اصدار صوت عند الخطأ
- ◆ **Mask**: تحدد شكل القناع برمز ك (#، %) كدلالة على البيانات.
- ◆ **PasswordChar**: لاطهار رمز داخل الصندوق بدل البيانات المستخدمة للمحافظة على سرية البيانات وعدم كشفها من الاخرين اثناء الادخال ك كلمات المرور.
- ◆ **PromptChar**: شكل الحرف المستخدم في تنسيق الاداة ويمكن تغييره باي حرف اخر.
- ◆ **Text**: تمثل القيمة المدخلة للاداة.
- ◆ **HidePromptOnLeave**: لاختفاء التنسيق عند الخروج من الاداة.
- ◆ **TextMaskFormat**: اظهار النص بتنسيق خاص في ال Mask.

الروزنامة MonthCalendar

تستخدم هذه الاداة لعرض التاريخ بصورة كاملة على النموذج كاسماء ايام الاسبوع واررقام ايام الاشهر بالتفصيل مع الاشارة الى تاريخ اليوم.

تموز، ٢٠١٧						
الجمعة	الخميس	الأربعاء	الثلاثاء	الاثنين	الأحد	السبت
٣٠	٢٩	٢٨	٢٧	٢٦	٢٥	٢٤
٧	٦	٥	٤	٣	٢	١
١٤	١٣	١٢	١١	١٠	٩	٨
٢١	٢٠	١٩	١٨	١٧	١٦	١٥
٢٨	٢٧	٢٦	٢٥	٢٤	٢٣	٢٢
٤	٣	٢	١	٣١	٣٠	٢٩
Today: ٢٠١٧/٠٧/٢٨						

اهم خصائص هذه الاداة هي:

- ◆ **AnnuallyBoldedDates**: تسمح بتحديد مجموعة من التواريخ السنوية وجعلها غامقة.
- ◆ **BoldedDates**: تسمح لك بتحديد التواريخ بلون غامق.
- ◆ **MonthlyBoldedDates**: جعل تاريخ الشهر غامق.
- ◆ **FirstDayOfWeek**: يحدد اول يوم من ايام الاسبوع.
- ◆ **MinDate**: الحد الادنى من التاريخ المسموح به.
- ◆ **MaxDate**: الحد الاقصى من التاريخ المسموح به.
- ◆ **ShowToDay**: تأخذ هذه الخاصية القيم التالية:
 - True: عرض تاريخ اليوم أسفل الاداة كما في الصورة أعلاه.
 - False: إخفاء تاريخ اليوم من أسفل الاداة.
- ◆ **ShowToDayCircle**: وتأخذ هذه الاداة القيم التالية:
 - True: وضع علامة المربع على تاريخ اليوم كما في الصورة أعلاه.
 - False: إزالة العلامة عن تاريخ اليوم.

تموز، ٢٠١٧						
الأحد	السبت	الجمعة	الخميس	الأربعاء	الثلاثاء	الاثنين
٢	١	٣٠	٢٩	٢٨	٢٧	٢٦
٩	٨	٧	٦	٥	٤	٣
١٦	١٥	١٤	١٣	١٢	١١	١٠
٢٣	٢٢	٢١	٢٠	١٩	١٨	١٧
٣٠	٢٩	٢٨	٢٧	٢٦	٢٥	٢٤
٦	٥	٤	٣	٢	١	٣١

Today: ٢٠١٧/٠٧/٢٨

◆ **ShowWeekNumbers**: عرض أرقام الأسبوع على يسار كل صف من الأيام الأسبوع كما في الصورة المجاورة.

◆ **MaxSelectionCount**: تحدد عدد الأيام التي يمكن اختيارها في الأداة.

◆ **ScrollChange**: عدد الأشهر التي يتم التنقل إليها عند النقر على زر التالي /السابق لعرضها.

◆ **SelectionRange**: يمكن من خلالها تحديد نطاق التواريخ المختارة في الأداة وتأخذ القيم التالية:

– **Start**: تاريخ البدء.

– **End**: تاريخ الانتهاء.

◆ **CalendarDimensions**: تحدد أبعاد الأداة (عدد

صفوف وأعمدة الأشهر المعروضة في الأداة). وتأخذ القيم التالية:

– **Width**: تحدد عدد الأشهر التي يمكن عرضها أفقياً داخل الأداة.

– **Height**: تحدد عدد الشهر التي يمكن عرضها عمودياً في الأداة.

تموز، ٢٠١٧							آب، ٢٠١٧						
الأحد	السبت	الجمعة	الخميس	الأربعاء	الثلاثاء	الاثنين	الأحد	السبت	الجمعة	الخميس	الأربعاء	الثلاثاء	الاثنين
٦	٥	٤	٣	٢	١	٣١	١٣	١٢	١١	١٠	٩	٨	٧
١٣	١٢	١١	١٠	٩	٨	٧	١٤	١٣	١٢	١١	١٠	٩	٨
٢٠	١٩	١٨	١٧	١٦	١٥	١٤	٢١	٢٠	١٩	١٨	١٧	١٦	١٥
٢٧	٢٦	٢٥	٢٤	٢٣	٢٢	٢١	٢٨	٢٧	٢٦	٢٥	٢٤	٢٣	٢٢
٣١	٣٠	٢٩	٢٨	٢٧	٢٦	٢٥	٣٠	٢٩	٢٨	٢٧	٢٦	٢٥	٢٤

Today: ٢٠١٧/٠٧/٢٨

رمز الإعلام NotifyIcon

تُستخدم هذه الأداة لوضع ايقونة البرنامج بجوار الساعة في شريط المهام لنظام التشغيل كرمز صغير تظهر عند النقر على السهم الصغير مع باقي الايقونات المخفية داخل علبة النظام في الجانب الأيمن من شريط المهام.

اهم خصائص رمز الإعلام هي:

◆ **BalloonTipIcon**: تحديد نوع الايقونة التي ستظهر في Tip عندما نقف على ايقونة البرنامج في شريط المهام وتضم عدة قيم.

– **None**: لا تُظهر أي ايقونة.

– **Info**: تُظهر ايقونة المعلومات في بالون الاداة.

– **Warning**: تُظهر ايقونة تحذير في بالون الاداة.

– **Error**: تُظهر ايقونة خطأ في البالون.

◆ **BalloonTipText**: تحديد النص الذي سيظهر في Tip.

◆ **BalloonTipTitle**: تحديد العنوان الذي سيظهر في الرسالة.

◆ **Icon**: تحدد الايقونة التي ستعرض في علبة النظام.

◆ **ContextMenuStrip**: تستخدم هذه الخاصية لعمل قائمة مختصرة لزر الفأرة الايمن.

التعامل مع الاداة برمجياً

NotifyIcon1.BalloonTipIcon = ToolTipIcon.Info

NotifyIcon1.BalloonTipText = "Set your text"

NotifyIcon1.BalloonTipTitle = "Set title"

NotifyIcon1.Visible = True

الزيادة والنقصان العددي NumericUpDown

تُظهر هذه الاداة بداخلها رقم يمكن من خلال الاسهم الجانيه لها زيادة قيمته او نقصانها، كما يمكن ان تستخدم لإدخال بيانات رقمية.

اهم خصائص هذه الاداة هي:

- ◆ **Increment**: زيادة قيمة العدد بمقدار معين.
- ◆ **DecimalPlaces**: تحديد عدد المراتب العشرية للعدد.
- ◆ **HexaDecimal**: عرض القيم بطريقة نظام HexaDecimal.
- ◆ **InterceptArrowKeys**: تحدد عمل الازرار بزيادة او نقصان قيمة العدد من خلال النقر على الاسهم الجانيه.
- ◆ **Maximum**: وهي اقصى قيمة لمقدار الزيادة
- ◆ **Minimum**: هي اقل قيمة لمقدار النقصان.
- ◆ **UpDownAlign**: تحديد موقع الاسهم اما على جانب اليمين او اليسار.
- ◆ **ThousandsSeparator**: اظهار فارزة المراتب كل ثلاث قيم داخل العدد.
- ◆ **Value**: القيمة الحالية للاداة التي تظهر عند التنفيذ.

التعامل مع الاداة برمجياً

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    NumericUpDown1.ReadOnly = True
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If NumericUpDown1.Value = 30 Then
        NumericUpDown1.Value = 0
    Else
        NumericUpDown1.Value += 1
    End If
End Sub
```

صندوق الصورة PictureBox

تستخدم هذه الاداة لعرض اي صورة يتم اختيارها من الجهاز، كما يمكنها عرض انواع مختلفة من الصور (Bmp, ico, Png,) .(jpg)

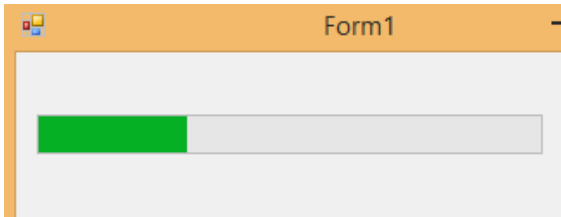
اهم خصائص صندوق الصورة هي:

- ◆ **Image**: تستخدم لتحديد الصورة المطلوب عرضها.
- ◆ **ImageLocation**: يحوي مسار الصورة المطلوب تحميلها من القرص او الـ Web.
- ◆ **InitialImage**: وضع صورة تظهر عند انتظار تحميل صورة اخرى.
- ◆ **ErrorImage**: وضع صورة عند فشل تحميل صورة اخرى.
- ◆ **SizeMode**: تحديد كيفية تعامل الاداة مع الصورة والتحكم بحجمها.

```
ofd.ShowDialog()
PictureBox1.SizeMode = PictureBoxSizeMode.Zoom
PictureBox1.Load(ofd.FileName)
PictureBox1.Visible = True
```

شريط التقدم ProgressBar

هنالك بعض الاجراءات التي يتطلب تنفيذها المزيد من الوقت وخلال فترة تنفيذها قد يشعر المستخدم بالملل، لذا تُستخدم هذه الأداة من أجل إضافة ميزة عرض تظهر للمستخدم ما تم تنفيذه من هذا الاجراء والنسبة المتبقية منه.



اهم خصائص شريط التقدم هي:

- ◆ **Maximum**: تُحدد القيمة العظمى لشريط التقدم.
- ◆ **Minimum**: تُحدد القيمة الدنيا لشريط التقدم.
- ◆ **Value**: القيمة الابتدائية لشريط التقدم وغالباً تكون صفراً.
- ◆ **Step**: مقدار الزيادة في قيمة شريط التقدم.
- ◆ **Style**: تُحدد شكل شريط التقدم.

التعامل مع الاداة برمجياً

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    If ProgressBar1.Value = 100 Then
        Timer1.Enabled = False
        Exit Sub
    End If
    ProgressBar1.Increment(1)
End Sub
```

زر الراديو RadioButton

تُستخدم هذه الأداة في التطبيقات بشكل مجموعة داخل إطار تتألف من زرین أو أكثر، ويُمكن للمستخدم إختيار خيار واحد فقط من الخيارات المعروضة أمامه، أي لا يمكنه تحديد أكثر من أداة واحدة حيث يتم تحديدها وإلغاء تحديدها من خلال النقر عليها بالفأرة.

اهم خصائص زر الراديو هي:

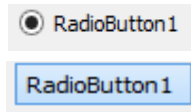
- ◆ **Checked**: تأخذ هذه الخاصية أحد القيم التالية:
 - **True**: اذا كانت الأداة محددة.
 - **False**: اذا كانت الأداة غير محددة.

◆ **CheckedAlign**: تحديد موقع صندوق التحقق على أحد جوانب الأداة.

◆ **Appearance**: تحديد مظهر الأداة وتأخذ القيم التالية:

– **Normal**: جعل الأداة تظهر بشكلها المعروف كما في الصورة المجاورة.

– **Button**: جعل الأداة تأخذ شكل كزر أمر.

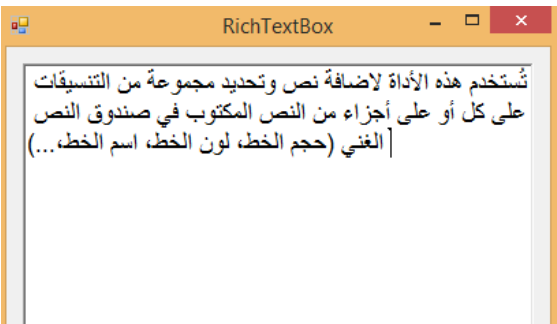


التعامل مع زر الراديو برمجياً

```
Private Sub RadioButton1_CheckedChanged(sender As Object, e As EventArgs) Handles
RadioButton1.CheckedChanged
    Dim x As String
    If RadioButton1.Checked = True Then
        x = MessageBox.Show("الخروج تأكيد", "البرنامج من الخروج تريد هل",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question)
        If x = vbYes Then
            Application.Exit()
        ElseIf x = vbNo Then
            RadioButton1.Checked = False
            Exit Sub
        End If
    End If
End Sub
```

صندوق النص الغني RichTextBox

تُستخدم هذه الأداة لإضافة نص وتحديد مجموعة من التنسيقات على كل أو جزء من النص المكتوب في صندوق النص الغني كـ (حجم الخط، لون الخط، اسم الخط،...) كما يُمكن من خلال هذه الأداة التعامل مع الملفات بسهولة كـ(فتح وحفظ) النصوص.



اهم خصائص صندوق النص الغني هي:

◆ **Text**: وضع نص داخل الأداة.

◆ **AutoWordSelection**: وتأخذ هذه الخاصية القيم التالية:

– **True**: تفعيل اختيار الكلمة تلقائياً.

– **False**: إلغاء اختيار الكلمة تلقائياً.

◆ **ShowSelectionMargin**: وتأخذ هذه الخاصية القيم التالية:

– **True**: رؤية الهوامش المختارة.

– **False**: إلغاء رؤية الهوامش.

◆ **MaxLength**: الحد الأقصى لعدد الأحرف التي يمكن إدخالها في الصندوق.

- ◆ **Margin**: تحديد المسافة بين هذه الأداة و هامش أداة اخرى.
- ◆ **HideSelection**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: اخفاء الجزء المحدد.
 - **False**: إلغاء خاصية الاخفاء.
- ◆ **DetecUrls**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: جعل عنوان الموقع الموجود في الصندوق يعمل كرابط.
 - **False**: إلغاء عمل عنوان الموقع كرابط.

التعامل مع صندوق النص الغني برمجياً

- ◆ **Find**: البحث داخل الصندوق.
- ◆ **LoadFile**: فتح ملف.
- ◆ **SaveFile**: حفظ ملف.
- ◆ **SelectedText**: اختيار النص لاجراء عمل معين.
- ◆ **SelectedRTF**: جعل التنسيق بصيغة RTF.
- ◆ **SelectionStart**: معرفة بداية النص المحدد(المظلل).
- ◆ **SelectionLength**: معرفة طول النص المحدد(المظلل).
- ◆ **SelectionFont**: وضع تنسيق للنص المحدد.
- ◆ **SelectionColor**: وضع لون للنص المحدد.
- ◆ **SelectionBackColor**: وضع لون لخلفية النص المحدد.
- ◆ **SelectionAlignment**: وضع محاذاة للنص المحدد.
- ◆ **SelectionBullets**: تنقيط النص المحدد.

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    fd.ShowDialog()
    RichTextBox1.SelectionFont = fd.Font
End Sub
```

صندوق النص TextBox

تسمح هذه الأداة للمستخدم بإدخال البيانات الى البرامج بطريقة سهلة وطبيعية، وتعتبر هذه الأداة من الأدوات الأكثر استخداماً في الفيجوال بيسك ، حيث تمتلك مجموعة كبيرة جداً من الأحداث والخصائص والعمليات.



اهم خصائص صندوق النص هي:

- ◆ **Text**: تضم النص الخاص بالأداة.
- ◆ **TextAlign**: تُحدد محاذاة النص داخل الصندوق وتأخذ القيم التالية:
 - **Left**: محاذاة النص للييسار.
 - **Right**: محاذاة النص لليمين.
 - **Center**: وضع النص في الوسط.

- ◆ **MultiLine**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح بتعدد الأسطر داخل الصندوق.
 - **False**: إلغاء تعدد الأسطر في الصندوق.
- ◆ **MaxLength**: الحد الأقصى لعدد الأحرف التي يمكن إدخالها الى صندوق النص.
- ◆ **CharacterCasing**: تحدد هذه الخاصية حالة الاحرف في صندوق النص وتأخذ القيم التالية:
 - **Normal**: تكون حالة الأحرف مطابقة لما يكتبه المستخدم.
 - **Upper**: جعل النص المكتوب في الصندوق يظهر بحروف كبيرة.
 - **Lower**: جعل النص المكتوب في الصندوق يظهر بحروف صغيرة.
- ◆ **AllowDrop**: وتأخذ هذه الدالة القيم التالية:
 - **True**: السماح للمستخدم بسحب النص الى الصندوق.
 - **False**: لا يمكن سحب النص الى الصندوق.
- ◆ **PasswordChar**: تُحدد الرمز الذي سيظهر بدل الاحرف المكتوبة في النص تفيد هذه الخاصية بالحفاظ على سرية كلمات المرور.
- ◆ **ReadOnly**: جعل نص الأداة للقراءة فقط، أي لا يمكن للمستخدم الكتابة فيه.
- ◆ **ScrollBars**: وتأخذ هذه الخاصية القيم التالية:
 - **None**: جعل الصندوق بلا أشرطة تمرير.
 - **Horizontal**: إظهار شريط التمرير الأفقي فقط.
 - **Vertical**: إظهار شريط التمرير العمودي فقط.
 - **Both**: إظهار شرطي التمرير الأفقي والعمودي معاً.
- ◆ **WordWrap**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تفعيل خاصية إتفاف النص في الصندوق (تجعل المؤشر يبدأ بسطر جديد تلقائياً بعد وصوله الى نهاية السطر الحالي).
 - **False**: إلغاء خاصية إتفاف النص للصندوق.

التعامل مع صندوق النص برمجياً

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If TextBox1.TextLength <> Nothing Then
        TextBox1.ForeColor = Color.Blue
        TextBox1.WordWrap = True
    End If
End Sub
```

أداة التلميح ToolTip

تستخدم هذه الأداة لعرض تلميح بسيط عند تمرير مؤشر الفأرة على أحد الأدوات دون النقر عليها، ويمكن للتلميح أن يظهر على شكل مربع يحتوي على معلومات متعلقة بالأداة ومن ثم يختفي بعد فترة زمنية قصيرة.



اهم خصائص أداة التلميح هي:

- ◆ **Active**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: ظهور التلميح عند تمرير مؤشر الفأرة على الأداة المعنية.
 - **False**: لا يظهر التلميح بتمرير مؤشر الفأرة على الأداة المطلوبة.
 - ◆ **BackColor**: لون خلفية مربع التلميح.
 - ◆ **ForeColor**: لون خط مربع التلميح.
 - ◆ **ToolTipIcon**: تحدد ايقونة مربع التلميح وتأخذ القيم التالية:
 - **None**: لا تظهر اي ايقونة في مربع التلميح.
 - **Info**: تظهر ايقونة علامة المعلومات داخل مربع التلميح.
 - **Warning**: تظهر ايقونة على شكل علامة تحذير في مربع التلميح.
 - **Error**: تظهر ايقونة علامة خطأ في مربع التلميح.
 - ◆ **ToolTipTitle**: العنوان الذي يظهر في مربع التلميح.
 - ◆ **AutomaticDelay**: وقت (الفترة الزمنية) لظهور التلميح ويقاس بالمللي ثانية.
 - ◆ **InitialDelay**: تُحدد الفترة الزمنية التي يجب إبقاء مؤشر الفأرة فوق الأداة ليظهر مربع التلميح.
 - ◆ **IsBalloon**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: يحول شكل التلميح الى بالون.
 - **False**: الإبقاء على شكله العادي.
 - ◆ **ReshowDelay**: تُحدد الوقت الفترة الزمنية لإعادة عرض التلميح.
 - ◆ **ShowAlways**: تأخذ هذه الخاصية القيم التالية:
 - **True**: إظهار التلميح بصورة دائمة.
 - **False**: ظهور التلميح عند تمرير مؤشر الفأرة على الأداة المطلوبة.
- التعامل مع أداة التلميح برمجياً

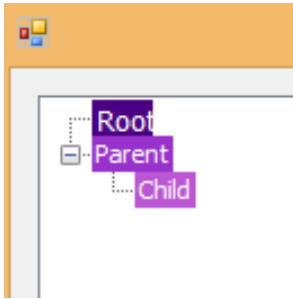
```

Private Sub TextBox1_MouseHover(sender As Object, e As EventArgs) Handles
TextBox1.MouseHover
    ToolTip1.ToolTipTitle = "نص صندوق"
    ToolTip1.ToolTipIcon = ToolTipIcon.Info
    ToolTip1.IsBalloon = True
    ToolTip1.BackColor = Color.Blue
    ToolTip1.ForeColor = Color.Azure
    ToolTip1.SetToolTip(Me.TextBox1, " نصية بيانات ادخال يجب ")
End Sub

```

أداة الشجرة TreeView

تستخدم هذه الأداة لعرض مجموعة من العناصر على شكل شجرة ذات فروع، وتقسم فروع هذه الشجرة الى فروع رئيسية تسمى Root والفروع الثانية تسمى Parent والفروع بالمستوى الثالث تسمى Child.

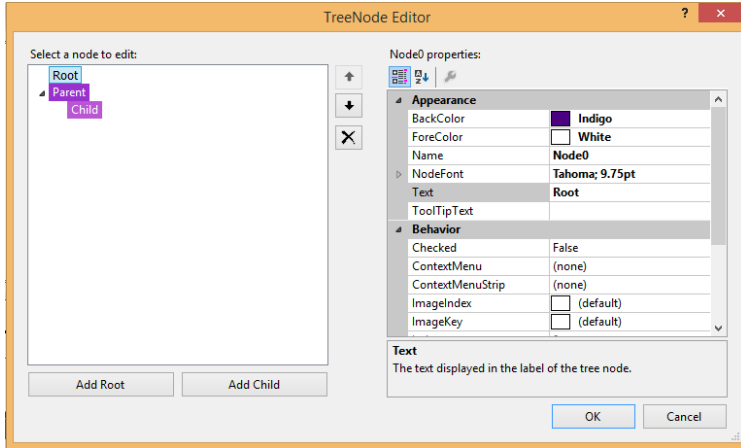


اهم خصائص أداة الشجرة هي:

- ◆ **Node**: إدخال عناصر للشجرة على اختلاف مستوياتها.
- ◆ **CheckBoxes**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: إظهار صندوق التحقق بجانب Node.
 - **False**: إظهار العقد بلا صناديق تحقق.
- ◆ **HotTracking**: وتأخذ الخاصية القيم التالية:
 - **True**: جعل اسماء العقد تظهر بشكل مماثل للإرتباط التشعبي.
 - **False**: جعل اسماء العقد تظهر بشكلها الاعتيادي.
- ◆ **LabelText**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح للمستخدم بتحرير اسم Node.
 - **False**: لا تسمح بتحرير اسم Node.
- ◆ **LineColor**: تُحدد لون الخطوط التي تربط عقد الشجرة.
- ◆ **PathSeperator**: العلامة التي تستخدم لمسار عقدة الشجرة.
- ◆ **RightToLeftLayout**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: جعل تخطيط الشجرة محاذياً من اليمين الى اليسار.
 - **False**: إبقاء تخطيط الشجرة محاذياً من اليسار الى اليمين.
- ◆ **ShowLines**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: إظهار الخطوط بين عقد الشجرة.
 - **False**: إخفاء الخطوط الموجودة بين عقد الشجرة.
- ◆ **ShowNodeToolTips**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: إظهار التلميحات على العقد.
 - **False**: أخفاء التلميحات من العقد.
- ◆ **ShowPlusMinus**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: إظهار أزرار علامة +/- بجانب عقد Parent.
 - **False**: إخفاء أزرار علامة +/- الموجودة بجانب عقد Parent.
- ◆ **ShowRootLines**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: عرض الخطوط بين عقد Root.
 - **False**: إخفاء الخطوط الموجودة بين عقد Root.
- ◆ **StateImageList**: تستخدم هذه الخاصية لربط أداة Tree بأداة ImageList لعرض الصور الموجودة فيها.

تحرير العقد Edit Nodes

عند اضافة أداة الشجرة الى سطح النموذج، تظهر الاداة بلا اي عقدة ولهذا يجب على المبرمج ان يقوم باضافة عقد مختلفة المتسويات للشجرة وان يقوم بتحرير هذه العقد واختيار الاسماء الخاصة بها والتي تتلائم مع التطبيق الذي يُريد انشاءه بالاضافة الى تنسيقها وتحديد الالوان المطلوبة.



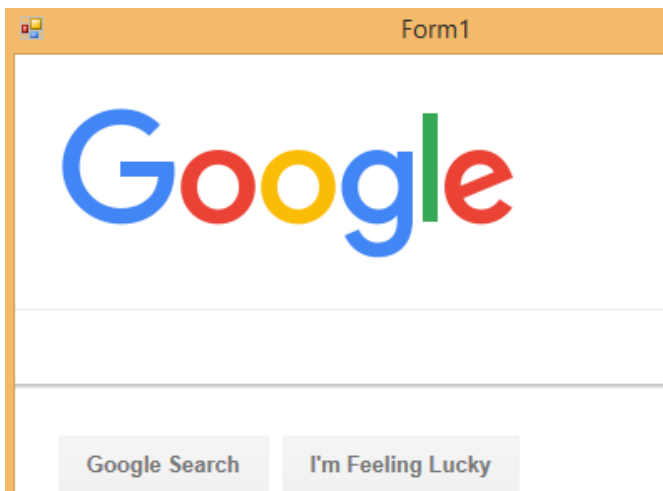
يتم تحرير عقد الشجرة أما بالنقر على السهم أعلى الأداة واختيار الامر Edit Nodes أو اختياره من خلال Nodes في نافذة الخصائص. ستظهر نافذة تحرير عقد الشجرة كما في الشكل المجاور.

التعامل مع الاداة برمجياً

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If TreeView1.SelectedNode Is Nothing Then
        TreeView1.Nodes.Add(TextBox1.Text)
    Else
        TreeView1.SelectedNode.Nodes.Add(TextBox1.Text)
    End If
End Sub
```

متصفح الويب WebBrowser

تستخدم هذه الأداة لعرض وتصفح مواقع الانترنت.



اهم خصائص متصفح الويب هي:

- ◆ **AllowNavigation**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح للأداة باستعراض صفحة اخرى بعد تحميل الصفحة الأولى.
 - **False**: لا تسمح بذلك.
- ◆ **Url**: تحديد الموقع المطلوب تصفحه.

التعامل مع الاداة برمجياً

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

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    WebBrowser1.GoBack()
```

End Sub

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click  
    WebBrowser1.GoForward()  
End Sub
```

End Sub

```
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click  
    WebBrowser1.GoHome()  
End Sub
```

End Sub

لوح التخطيط الإنسيابي FlowLayoutPanel

تستخدم هذه الأداة كحاوية لبقية الأدوات وتعمل على ترتيبها تلقائياً وبهذا توفر على المبرمج الوقت الذي يستغرقه في ترتيب الأدوات بداخلها.

اهم خصائص لوح التخطيط هي:

- ◆ **FlowDirection**: تحديد اتجاه العناصر داخل الأداة.
- ◆ **WrapContent**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح للأدوات الموجودة داخل اللوح بالإنحناء وعدم اجتياز حدوده.
 - **False**: لا يمكن للأدوات الإنحناء عند وصولها الى حدود اللوح.

صندوق المجموعة GroupBox

عبارة عن صندوق توضع فيه مجموعة من الأدوات تحمل النوع نفسه. مثلاً توضع فيه أدوات RadioButton او CheckBox ليتم اختيار الأدوات التي تحقق الغرض المطلوب في البرنامج.

اهم خصائص صندوق المجموعة هي:

- ◆ **Text**: التسمية الخاصة بالمجموعة.
- ◆ **AutoSizeMode**: وتأخذ هذه الخاصية القيم التالية:
 - **GrowAndShrink**: يتوسع ويتقلص ليناسب حجم محتوياته.
 - **GrowOnly**: يتوسع فقط بما يتناسب مع حجم محتوياته.

اللوح Panel

تستخدم هذه الأداة كحاوية لبقية الأدوات التي تقوم بمهام معينة.

اهم خصائص اللوح هي:

- ◆ **AutoSizeMode**: وتأخذ هذه الخاصية القيم التالية:
 - **GrowAndShrink**: يتوسع ويتقلص ليناسب حجم محتوياته.

فصل المحتويات SplitContainer

تستخدم هذه الاداة ايضا كحاوية للأدوات، بالإضافة الى انها تقوم بتقسيم سطح النموذج الى قسمين من نوع Panel يمكن اضافة الأدوات إليها للقيام بمهام معينة.

اهم خصائص فصل المحتويات هي:

- ◆ **FixedPanel**: تعمل على تثبيت حجم الاداة.
- ◆ **IsSplitterFixed**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: وتعني ان الفاصل بين اللوحين ثابت ولا يمكن تغييره.
 - **False**: امكانية تحريك الفاصل بين اللوحين.
- ◆ **Orientation**: وتأخذ هذه الخاصية القيم التالية:
 - **Vertical**: جعل الفاصل عمودي.
 - **Horizotal**: جعل الفاصل افقي.
- ◆ **Panel1/2Collapsed**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: قفل اللوح 1 / 2.
 - **False**: فتح اللوح 1 / 2.
- ◆ **SplitterDistance**: تحديد مسافة الفاصل من يسار او أعلى الحافة مقاسة بوحدة البكسل.
- ◆ **SplitterIncrement**: تحدد عدد بكسلات مقدار الزيادة عند تحريك الفاصل.
- ◆ **SplitterWidth**: تُحدد سُمك الفاصل.

التبويب TabControl

تتكون هذه الأداة من تبويب واحد او عدة تبويبات، كل تبويب يمكن أن يحتوي على مجموعة أدوات تؤدي مهمة معينة من التطبيق.

اهم خصائص التبويب هي:

- ◆ **RightToLeftLayout**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: ضبط اتجاه التبويبات من اليمين الى اليسار.
 - **False**: ضبط اتجاه التبويبات من اليسار الى اليمين.
- ◆ **Text**: اختيار اسم التبويب.
- ◆ **HotTrack**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تغيير شكل التبويبات عند تمرير مؤشر الفأرة فوقها.
 - **False**: لا يتأثر شكل التبويبات بمرور المؤشر.
- ◆ **Appearance**: وتأخذ هذه الخاصية القيم التالية:

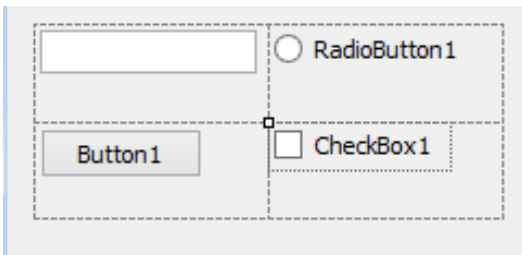
- **Normal**: أظهار التبويبات بشكلها الطبيعي.
- **Buttons**: جعل مظهر التبويبات على شكل أزرار أمر.
- **FlatButton**: جعل مظهر التبويبات على شكل أزرار مسطحة.

التعامل مع التبويب برمجياً

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If TabControl1.SelectedTab Is Nothing Then
        TabControl1.TabPages.Add(TextBox1.Text)
    Else
        TabControl1.TabPages.Remove(TabControl1.SelectedTab)
    End If
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    TabControl1.Appearance = TabAppearance.Buttons
    TabControl1.Alignment = TabAlignment.Bottom
End Sub
```

لوح التخطيط المسطح TableLayoutPanel

هذه الأداة عبارة عن جدول يتكون من مجموعة من الخلايا الموزعة بشكل افقي وعمودي، كل خلية يمكن ان تضم أداة واحدة فقط.



اهم خصائص لوح التخطيط هي:

- ◆ **ColumnCount**: يُحدد عدد الأعمدة في الجدول.
- ◆ **Columns**: اضافة وحذف أعمدة وتُحدد شكل العمود في الجدول.
- ◆ **RowCount**: يُحدد عدد الصفوف في الجدول.
- ◆ **Row**: اضافة وحذف صفوف وتُحدد شكل الصفوف في الجدول.
- ◆ **CellStyle**: يُحدد مظهر حدود الخلية في الجدول.

أداة القائمة المختصرة ContextMenuStrip

تُستخدم هذه الأداة لإظهار قائمة مختصرة عند النقر بمفتاح الفأرة الأيمن على النموذج أو على أي أداة من أدوات التحكم الموجودة على سطح النموذج خلال فترة التنفيذ.

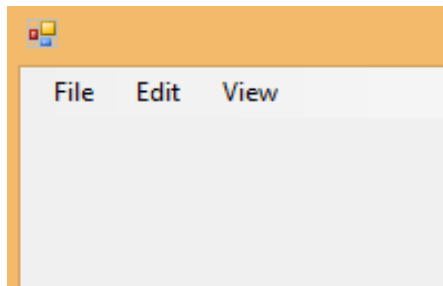
اهم خصائص أداة القائمة المختصرة هي:

- ◆ **DropShadowEnable**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: وضع ظل للقائمة.
 - **False**: ازالة ظل القائمة.
- ◆ **Items**: اضافة عناصر للقائمة.
- ◆ **LayoutStyle**: تُحدد شكل تخطيط بنود الأداة.

- ◆ **ShowImageMargin**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تحديد هامش لعرض صور ضمن أوامر القائمة.
 - **False**: لا تترك هامش للصور.
- ◆ **ShowCheckMargin**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: عرض هامش التحقق.
 - **False**: إلغاء هامش التحقق.
- ◆ **ShowItemsToolTips**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: عرض تلميحات على عناصر القائمة.
 - **False**: لا تعرض تلميحات العناصر.
- ◆ **Checked**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: إظهار صندوق تحقق بجوار عناصر القائمة.
 - **False**: لا يظهر صندوق التحقق بجوار العناصر.
- ◆ **Text**: وضع نص متعلق بالأداة.
- ◆ **CheckOnClick**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: وضع علامة صح في صندوق التحقق الذي بجوار العناصر بنقرة واحدة.
 - **False**: إبقاء علامة الصح ظاهرة في الصندوق.
- ◆ **DoubleClickEnable**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تفعيل العناصر بالنقر المزدوج بالفأرة.
 - **False**: لا تُفعل العناصر.
- ◆ **DropDown**: تحديد الأداة التي تحتوي على القائمة المنسدلة.
- ◆ **DropDownItems**: وضع عناصر داخل القائم المنسدلة.
- ◆ **ShortcutKeyDisplayString**: النص الذي يعرض كحرف مختصر.
- ◆ **ShortcutKeys**: اختيار مفتاح اختصار لعناصر القائمة.

أداة شريط القوائم MenuStrip

تعتبر القوائم من الأدوات الأساسية في بناء التطبيقات باستخدام الفيجوال بيسك ، خصوصاً عندما يكون لدينا عدد كبير من الأوامر. ويمكن اعتبار كل عنصر من عناصر القائمة كزر أمر نكتب عليه إجراء يُنفذ عند النقر عليه.



يمكن انشاء شريط القوائم باستخدام أداة MenuStrip.

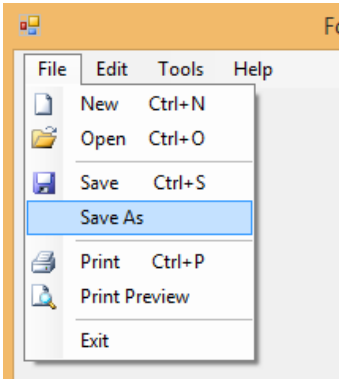
اهم خصائص أداة MenuStrip هي:

- ◆ **AllowItemsReorder**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح بإعادة ترتيب العناصر عند الضغط على مفتاح ALT.
 - **False**: لا تسمح بترتيب العناصر.

- ◆ **Items**: مجموع العناصر التي تظهر في الشريط.
- ◆ **LayoutStyle**: تُحدد اتجاه عناصر الشريط.
- ◆ **ShowItemToolTips**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: عرض التلميح على العنصر.
 - **False**: لا يُعرض التلميح.
- ◆ **Stretch**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح بتوسع اداة الشريط الى نهاية الحاوية.
 - **False**: لا تسمح بالتوسع.
- ◆ **ImageScalingSize**: تُحدد حجم صور عناصر القائمة.
- ◆ **MdiWindowLsitItem**: تُحدد العنصر من الشريط الذي سيعرض كقائمة MdiWindow (واجهة متعددة المستندات).
- ◆ **RenderMode**: اختيار شكل رسومي للاداة.
- ◆ **TextDirection**: تحدد اتجاه رسم النص على الاداة.
- ◆ **CheckState**: تبين حالة عنصر صندوق التحقق.
- ◆ **DisplayStyle**: شكل عرض العنصر وتأخذ القيم التالية:
 - **None**: إخفاء العنصر.
 - **Text**: عرض نص العنصر فقط.
 - **Image**: عرض صورة العنصر فقط.
 - **ImageAndText**: عرض صورة ونص العنصر.
- ◆ **DoubleClickEnable**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تنشيط Toolstrip من خلال النقر المزدوج بالفأرة.
 - **False**: لا تُنشَط Toolstrip بالنقر المزدوج للماوس.
- ◆ **ImageScaling**: ضبط حجم الصورة المستخدمة مع العنصر.

ملاحظة :

يحتوي الفيچوال بيسك 2013 على شريط قوائم قياسي جاهز للاستخدام، نقوم بالنقر على السهم أعلى أداة MenuStrip ونختار الامر Insert Standard Items سيظهر الشريط أعلى النموذج كما في الشكل التالي:



شريط الحالة StatusStrip

هو شريط يظهر أسفل النموذج يستخدم لعرض معلومات معينة في التطبيق، يضم هذا الشريط عدد من الأدوات كما تظهر في الصورة التالية:

لاضافة عنصر من عناصر الأداة انقر على السهم الموجود داخل المربع ستظهر قائمة اختر العنصر المطلوب.

اهم خصائص شريط الحالة هي:

◆ **GripStyle**: وتأخذ هذه الدالة القيم التالية:

– **Hidden**: إخفاء القبضة من ToolStrip.

– **Visible**: إظهار القبضة على ToolStrip.

◆ **Items**: مجموع العناصر التي تظهر على ToolStrip.

◆ **LayoutStyle**: تُحدد توجيه التخطيط للـ ToolStrip.

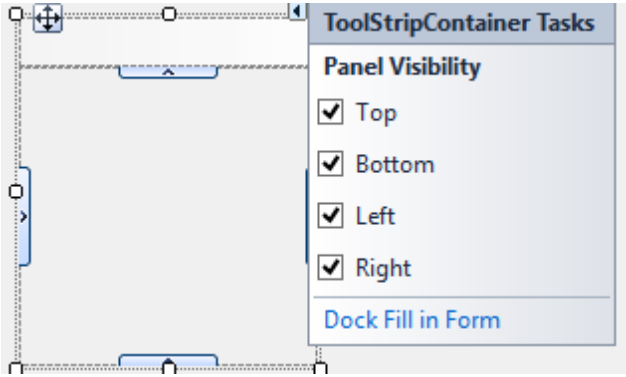
التعامل مع الأداة برمجياً

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ToolStripStatusLabel1.Text = Today
    ToolStripStatusLabel2.Text = TimeOfDay
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    If ToolStripProgressBar1.Value <= ToolStripProgressBar1.Maximum - 1 Then
        ToolStripProgressBar1.Value += 1
    Else
        Timer1.Enabled = False
    End If
End Sub
```

وعاء شريط الأدوات ToolStripContainer

تسمح هذه الأداة بإستخدام Panel في الاتجاهات الأربعة والتي يمكن استخدامها لإضافة عناصر تحكم كـ شرطي عنوان، شريط ادوات، شريط الحالة وغيرها) تعمل كل أداة من من هذه الادوات بشكل مستقل لتؤدي وظائف معينة.



مزود الخطأ ErrorProvider

تسمح هذه الاداة بتحديد مكان الخطأ داخل البرنامج مما يسهل على المستخدم تجنب تكرار الأخطاء.

اهم خصائص مزود الاخطاء هي:

- ◆ **BlinkRate**: تحدد الفترة الزمنية لظهور ايقونة الخطأ بجانب الكائن مقاسه بالملي ثانية.
- ◆ **BlinkStyle**: وتأخذ هذه الخاصية القيم التالية:
- **BlinkDifferentError**: يحدث الوميض عدة مرات ويتوقف.
- **AlwaysBlink**: يستمر الوميض بلا توقف.
- **NeverBlink**: لا يحدث الوميض نهائياً.
- ◆ **Icon**: شكل ايقونة الخطأ.

التعامل مع الاداة برمجياً

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If Val(TextBox1.Text) < 1 Then
        ErrorProvider1.SetError(TextBox1, "Enter value greater than 0")
    Else
        ErrorProvider1.Clear()
    End If
End Sub
```

أداة قائمة الصور ImageList

تستخدم هذه الاداة لضم مجموعة من الصور وعمل اتصال لها مع الكائنات الأخرى.

اهم خصائص قائمة الصور هي:

- ◆ **ColorDepth**: تحدد عدد الالوان المستخدمة في إعداد الصور.
- ◆ **Images**: الصور المخزونة في الأداة .Imagelist.
- ◆ **ImageSize**: تحدد حجم الصور في الأداة.

أداة طباعة المستند PrintDocument

تستخدم هذه الأداة لتحديد البيانات المطلوب طباعتها.

اهم خصائص طباعة المستند هي:

- ◆ **DocumentName**: اسم المستند المعروض للمستخدم.

التعامل مع الأداة برمجياً

```
Private Sub Button1_Click_1(sender As Object, e As EventArgs) Handles Button1.Click
    PrintDocument1.Print()
End Sub
```

```
Private Sub PrintDocument1_PrintPage(sender As Object, e As Printing.PrintPageEventArgs)
Handles PrintDocument1.PrintPage
    Dim fnt As New Font("Arial", 18, FontStyle.Bold)
    e.Graphics.DrawString(RichTextBox1.Text, fnt, Brushes.Black, 100, 100)
End Sub
```

أداة معاينة قبل الطباعة PrintPreviewDialog

تستخدم هذه لعرض محتويات المستند بشكل Dialog قبل طباعتها. حيث تسمح للمستخدم

اهم خصائص الأداة هي:

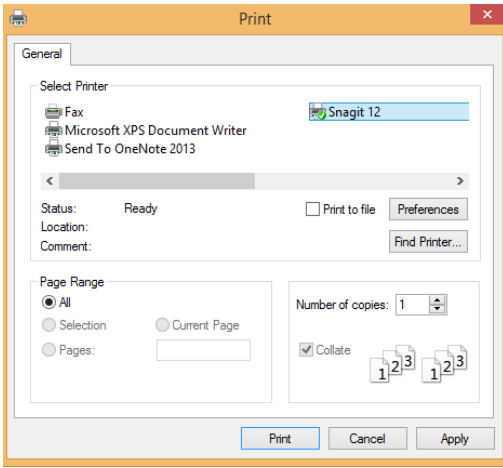
- ◆ **Document**: معاينة مستند الطباعة.

التعامل مع الاداة برمجياً

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    If RichTextBox1.Text = "" Then
        MsgBox("Write text")
    Else
        PrintPreviewDialog1.ShowDialog()
    End If
End Sub
```

مربع حوار الطباعة PrintDialog

تستخدم هذه الأداة لعرض نافذة الطباعة.



التعامل مع الاداة برمجياً

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    PrintDialog1.ShowDialog()
End Sub
```

مربع حوار الألوان ColorDialog

تستخدم هذه الأداة لعرض مجموعة من الألوان في نافذة أمام المستخدم ليقوم بإختيار لون واحد منها.

اهم خصائص الأداة هي:

- ◆ **AllowFullOpen**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح باستخدام زر الألوان المخصصة من مربع الحوار.
 - **False**: لا تسمح باستخدام زر الألوان المخصصة من مربع الحوار.
- ◆ **AnyColor**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح لصندوق العرض بعرض كل الألوان المتاحة في مجموعة من الألوان الأساسية.
 - **False**: لا تسمح بعرض بها.
- ◆ **FullOpen**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح بعرض الألوان المخصصة عند فتح مربع الحوار.
 - **False**: لا تسمح بعرضها عند الفتح.
- ◆ **SolidColorOnly**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تحدد المستخدمين بإختيار الألوان الصلبة فقط من مربع الحوار.
 - **False**: لا تحدد

التعامل مع الاداة برمجياً

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    ColorDialog1.ShowDialog()
    RichTextBox1.ForeColor = ColorDialog1.Color
End Sub
```

مربع حوار الخط FontDialog

تستخدم هذه الأداة لعرض مجموعة من انواع، حجوم وأنماط الخطوط في نافذة أمام المستخدم ليقوم بإختيار ما يناسبه منها.

اهم خصائص الأداة هي:

- ◆ **ShowApply**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح بعرض زر تطبيق في مربع الحوار.
 - **False**: لا تعرض الزر داخل مربع الحوار.
- ◆ **ShowColor**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح بعرض خيار الألوان داخل مربع الحوار.
 - **False**: لا يُعرض خيار الألوان في المربع.
- ◆ **ShowEffects**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح بعرض تسطير، توسيط وخيارات لون الخط.
 - **False**: لا تسمح بعرضهم.

التعامل مع الأداة برمجياً

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    FontDialog1.ShowColor = True
    FontDialog1.ShowApply = True
    FontDialog1.ShowEffects = True
    FontDialog1.ShowDialog()
    TextBox1.Font = FontDialog1.Font
End Sub
```

أداة فتح الملفات OpenFileDialog

تستخدم هذه الاداة لعرض مجموعة الملفات من خلال توفير نافذة تفتح للمستخدم للوصول الى مكان وجود ملفاته واختيار الملف المطلوب ليتم فتحه باستخدام الأداة المناسبة لنوع الملف.

اهم خصائص الأداة هي:

- ◆ **AddExtension**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: تسمح باضافة امتدادات تلقائياً الى اسماء الملفات.
 - **False**: لا تضع امتداد للملف.
- ◆ **CheckFileExists**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: عرض تحذير عندما يقوم المستخدم بتحديد اسم ملف غير موجود.
 - **False**: لا يعرض التحذير.
- ◆ **CheckPathExists**: وتأخذ هذه الخاصية القيم التالية:
 - **True**: التحقق من وجود المسار المحدد قبل العودة من Dialog.

– **False**: لا يتحقق من المسار.

◆ **DefaultExt**: امتداد الملف الافتراضي، عندما يكتب المستخدم اسم الملف يتم اضافة الامتداد الى نهاية الملف المحدد.

◆ **DereferenceLinks**: وتأخذ هذه الخاصية القيم التالية:

– **True**: تتحكم بمختصرات الوصول للمحتويات قبل الرجوع من Dialog.

– **False**: لا تسمح بالمختصرات.

◆ **FileName**: عرض اول ملف في مربع الحوار او اخر ملف تم اختياره من قبل المستخدم.

◆ **Filter**: تحدد نوع الملفات المطلوبة مثلاً (WordFile|*.docx|ImageFile|*.Bmp...).

◆ **InitialDirectory**: تعيين دليل ملفات ابتدائي لمربع الحوار.

◆ **MultiSelect**: وتأخذ هذه الخاصية القيم التالية:

– **True**: تسمح بتحديد أكثر من ملف في Dialog.

– **False**: لا تسمح بذلك.

◆ **Title**: النص الذي يظهر كعنوان لمربع الحوار.

التعامل مع الأداة برمجياً

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    OpenFileDialog1.Filter = "FileImage|*.jpg|BmpImage|*.bmp"
    OpenFileDialog1.InitialDirectory = Application.StartupPath
    OpenFileDialog1.ShowDialog()
    PictureBox1.Load(OpenFileDialog1.FileName)
End Sub
```

أداة حفظ الملفات SaveFileDialog

تسمح هذه الأداة للمستخدم بحفظ ملفات محددة بالمكان الذي يرغب به وبالامتداد المطلوب.

اهم خصائص هذه الأداة هي:

◆ **CreatePrompt**: وتأخذ هذه الخاصية القيم التالية:

– **True**: تتطلب اخذ اذن من المستخدم لانشاء ملف جديد اذا حدد المستخدم ملف غير موجود.

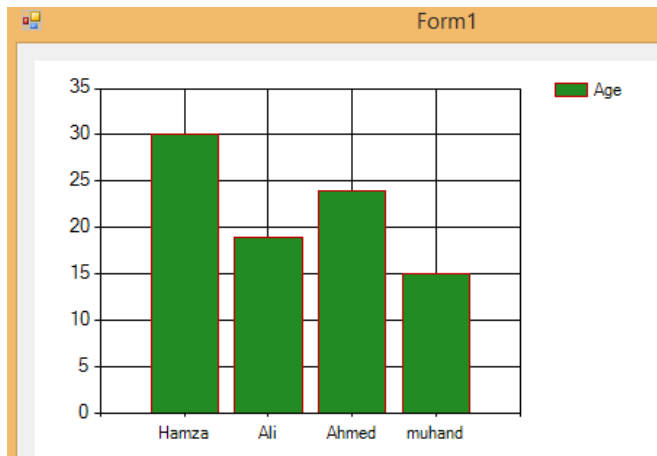
– **False**: لا تُنشأ ملف.

التعامل مع الأداة برمجياً

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    SaveFileDialog1.Filter = "jpgimage|*.jpg|pngimage|*.png"
    SaveFileDialog1.InitialDirectory = Application.StartupPath
    SaveFileDialog1.ShowDialog()
    PictureBox1.Image.Save(SaveFileDialog1.FileName)
End Sub
```

المخططات Chart

تستخدم هذه الاداة لرسم اشكال بيانية على سطح النموذج لتوضيح نسب معينة أو أرقام احصائية تخص دراسة أو موضوع معين.



اهم خصائص المخططات هي:

- ◆ **Name**: اسم السلسلة البيانية.
- ◆ **BorderColor**: وضع لون لحدود السلسلة.
- ◆ **BorderDashStyle**: وضع شكل لحدود السلسلة.
- ◆ **BorderWidth**: تحديد سُمك حدود السلسلة.
- ◆ **Color**: لون السلسلة البيانية.
- ◆ **ChartType**: تحدد نوع المخطط المستخدم في رسم السلسلة.
- ◆ **Points**: عدد نقاط السلسلة البيانية.
- ◆ **Label**: التسمية الخاصة بنقاط السلسلة.

التعامل مع الأداة برمجياً

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Me.Chart1.Series("Age").Points.AddXY("Hamza", 30)
    Me.Chart1.Series("Age").Points.AddXY("Ali", 19)
    Me.Chart1.Series("Age").Points.AddXY("Ahmed", 24)
    Me.Chart1.Series("Age").Points.AddXY("muhand", 15)
End Sub
```

الفصل الرابع

العلامات الأساسية للفيجوال بيسك

محتويات الفصل

Variables المتغيرات

الشكل العام لتعريف متغير

شروط اختيار اسم المتغير

انواع البيانات العددية

String Variables المتغيرات النصية

Character Variables المتغيرات الحرفية

Date Variables متغيرات التاريخ

خيارات التدقيق، التصريح والاستنتاج

Variant المتغير من نوع

Object المتغير من نوع

تحويل الأنواع البيانية للمتغيرات

تعلمية الإسناد لمتغير

Constants الثوابت

مجال تعريف المتغير وحياة المتغير

Static Variable المتغيرات الساكنة

العمليات الحسابية

العمليات المنطقية

البنى الشرطية

- If... Then
- If... Then... Else
- If... Then... ElseIf... Else
- If... Then...ElseIf...ElseIf...Else
- IIf(Expression , True , False)
- SelectCase

Loop جمل التكرار

- For ... Next
- While... Loop

- Do While... Loop
- Do Until... Loop

العبارات المتداخلة

Properties الخصائص

Events الأحداث

Class الفئة

صفحة برمجة النموذج

تسمية إجراءات الأحداث

Procedures البرامج الفرعية

Arrays المصفوفات

Multi-Dimensional Arrays المصفوفات المتعددة الأبعاد

Dynamic Arrays المصفوفات الديناميكية

دوال فحص البيانات

Structures التراكيب

المتغيرات Variables

المتغير هو عبارة عن عنوان موقع معين محجوز في ذاكرة الحاسوب يُمكن القراءة منه والكتابة فيه، تختلف حجم المساحة المحجوزة للمتغير باختلاف النوع البياني الخاص به. حيث تُستخدم المتغيرات في البرامج لتخزين القيم بشكل مؤقت.

يسمح الفيجوال بيسك باستخدام المتغيرات دون الإعلان عنها، كما في الفيجوال بيسك 6. وعند الاعلان عن متغير وأسناد قيمة له يقوم الفيجوال بيسك بتحديد النوع البياني للمتغير من خلال القيمة المسنده له.

القواعد البرمجية السليمة تُحتم علينا ان نُعرف كل متغير نستخدمه ونُحدد نوعه البياني.

الشكل العام لتعريف متغير:

Dim variablename [As type]

حيث ان:

◆ Variablename: يمثل اسم المتغير.

◆ Type: يمثل النوع البياني للمتغير. (String, Double, Integer).

عندما ننشأ متغير باسم a يقبل اعداد صحيحة داخل إجراء النقر لزر أمر.

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim num As Integer
    num += 1
    MsgBox(num)
End Sub
```

يمكن تعريف أكثر من متغير باستخدام Dim كما يلي:

Dim a, b As Integer

ففي هذه الحالة سيُعتبر الفيجوال بيسك المتغيرين من نوع Integer

كذلك الحالة بالنسبة للاعلان التالي:

Dim a = 0

حيث اعتبر الفيجوال بيسك المتغير a من نوع Integer معتمداً على القيمة المسنده للمتغير.

بالعودة الى المتغير num في الإجراء السابق سيُوجد عند تنفيذ الإجراء ولا يمكن التعامل معه من اجراء اخر وكذلك قيمته ستضيع عند الخروج من الإجراء.

فعند تنفيذ اجراء النقر سيؤدي الى ظهور صندوق رسالة يحتوي على القيمة 1.

عندما نُعلن عن متغير ولا نُعطي له نوع بياني محدد فان VS سيُعتبر المتغير من نوع Object الذي يقبل ارقاماً وتواريخاً ونصوصاً. ولا يتطلب إجراء اي عملية تحويل للبيانات عندما ننسبها الى متغير من نوع Object حيث ان VS سيقوم تلقائياً بعملية التحويل. ولهذا يجب علينا عند الإعلان عن متغير ان نحدد نوعه البياني.

شروط اختيار اسم المتغير

عند اختيار اسم للمتغير يراعى عدة شروط لتجنب حدوث الاخطاء:

- 1- ان لا يكون اسم المتغير من الكلمات المحجوزة في اللغة كـ (End , for , goto, case).
- 2- ان يبدأ الاسم بحرف واحد او اكثر او حرف متبوع بارقام ورموز مثل (a, username, num1, ...).
- 3- لا يبدأ اسم المتغير بارقام او رموز مثل (1a, %num, ...).
- 4- ان لا يحتوي على علامات ترقيم او فراغات.

انواع البيانات العددية

النوع البياني	حجم التخزين (المساحة المحجوزة في الذاكرة)	المجال
Byte	1 byte	0 to 255
SByte	1 byte	-128 to 127
Boolean	2 bytes	True Or False
Short	2 bytes	-32,768 to 32,767
UShort	2 bytes	0 to 65,535
Integer	4 bytes	-2,147,483,648 to 2,147,483,647
UInteger	4 bytes	0 to 4,294,967,295 بدون اشارة
Single	4 bytes	-3.402823E38 to -1.4011298E45 قيم سالبة 1.401298E45 to 3.402823E38 قيم موجبة
Long	8 bytes	-9,223,372,036,854,755,808 to 9,223,372,036,854,755,807
ULong	8 bytes	0 to 18,446,744,073,709,551,615
Double	8 bytes	-4.94065645841247E-324 to -1.79769313486232E308 الاعداد السالبة 1.79769313486232E308 to 4.94065645841247E-324 الاعداد الموجبة
Decimal	16 bytes	الأعداد الصحيحة والكسرية (ذات فاصلة مدرجة بواسطة عامل تحليل من 0 الى 28)

ملاحظة: 1Byte=8 bit فكل حرف او رقم يُخزن في الذاكرة يمثل بـ(8) نبضات، البت هو اصغر وحدة خزن في الحاسوب.

المتغيرات النصية String Variables

تخزن المتغيرات النصية النصوص فقط . يتم الإعلان عن المتغيرات النصية كما يلي:

```
Dim str1 As String
```

حيث نستطيع اسناد اي قيمة نصية للمتغير str1 وكذلك يمكن تخزين 2 غيغابايت من النصوص في المتغير. أي ما يقارب 2 بليون حرف.

```
str1="Welcome World"
```

كذلك يمكن اسناد قيم رقمية بهئية نصية الى المتغير من خلال التعبير التالي:

```
Dim num As Integer = 3200
```

```
Dim str1 As String = "25,000"
```

مع التنبيه الى ان المتغير str1 في مثل هذه الحالة يُمكن استخدامه في العمليات الحسابية وكذلك المتغير num يمكن استخدامه في العمليات النصية، حيث ان الـ VS سينجر التحويلات الضرورية طالما ان الخيار Option strict معطل (Off).

المتغيرات الحرفية Character Variables

المتغيرات الحرفية تخزن حرف مفرد بحجم (2 bytes) يتم التصريح عن المتغيرات الحرفية كما يلي:

```
Dim str1 As Char = "A"
```

في المثال اعلاه تم اسناد الحرف A للمتغير str1.

عندما يتم الاعلان عن متغير من نوع Char واسناد سلسلة نصية او اكثر من حرف للمتغير str1 ففي هذه الحالة يتم اخذ الحرف الأول فقط للمتغير الحرفي. كما في الاجراء التالي:

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

```
    Dim str1 As Char = "ABC"
```

```
    MsgBox(str1)
```

```
End Sub
```

في الاجراء السابق تم الاعلان عن متغير حرفي str1 وتم اسناد "ABC" له ولكن عند تنفيذ الإجراء فان قيمة المتغير ستكتفي باخذ الحرف الاول فقط ويتم اهمال بقية الحروف. اي ستظهر الرسالة داخل الصندوق الحرف A.

عندما يكون خيار Option strict معطل (Off) عندها يمكن تحويل الحرف الى القيم العددية الموافقة لها حسب نظام الانسي ANSI وتعني (المعهد القياسي الدولي الأمريكي American National Standard Institute).

كما في الإجراء التالي:

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

```
    Dim str1 As Char = "A"
```

```
    MsgBox(Convert.ToInt32(str1))
```

```
End Sub
```

نلاحظ في الإجراء السابق تم الاعلان عن متغير اسمه str1 من نوع Char واسندت له القيمة A وهي قيمة حرفية، داخل صندوق النص تم استخدام Convert لتحويل النوع البياني للمتغير من حرفي الى رقمي. اي ستظهر القيمة 65 داخل الصندوق.

متغيرات التاريخ Date Variables

قيم التاريخ والوقت تخزن بشكل داخلي في تنسيق خاص، يمكن وصفها على سبيل التقريب بانها قيم مزدوجة، الجزء الصحيح منها يمثل التاريخ والجزء الكسري يمثل الوقت. يمكن التصريح عن متغير من نوع وقت كما يلي:

```
Dim mydate As Date
```

يمكن تنسيق التاريخ من خلال تحديد صيغة التاريخ من تبويب Format من خيار Region الموجود في Control Panel. حيث يكون تنسيق التاريخ في الولايات المتحدة mm/DD/YY اما في باقي البلدان فان التنسيق سيكون dd/mm/yy. لاحظ المثال التالي:

```
Dim mydate As Date = #5/3/2017#
```

يجب وضع التاريخ بين علامتي (#) كما يجب مراعاة تنسيق التاريخ بحيث يكون مطابق لتنسيق تاريخ الجهاز لديك لتجنب حدوث الاخطاء فاذا غيرنا ترتيب التاريخ في القيمة المسندة للمتغير كما يلي:

```
Dim mydate As Date = #2017/6/7#
```

في هذه الحالة لا يُنفذ البرنامج لان تنسيق التاريخ في القيمة المسندة للمتغير غير صحيح.

لاحظ الاجراء التالي:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim mydate As Date
    mydate = Today
    MsgBox(mydate)
End Sub
```

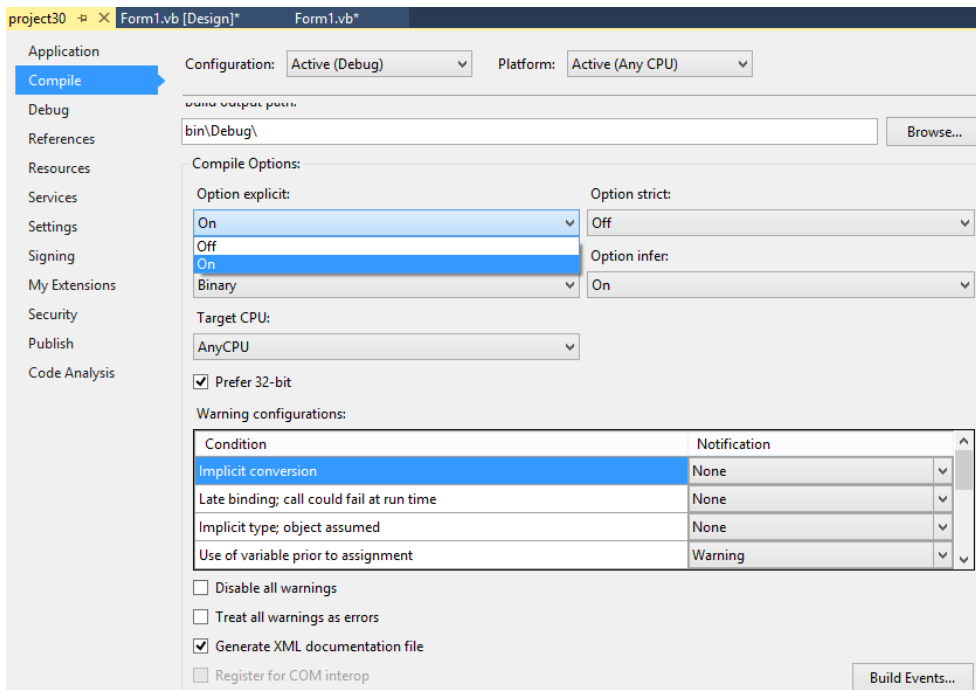
في الاجراء السابق تم الاعلان عن متغير mydate من نوع Date في الخطوة التالية تم اسناد قيمة للمتغير متمثلة بدالة Today حيث تُعيد هذه الدالة تاريخ اليوم بالاعتماد على تاريخ الجهاز. وهو ما سيظهر في نافذة صندوق الرسالة.

خيارات التدقيق، التصريح والاستنتاج

يُتيح لك مترجم الفيچوال ببسك ثلاث خيارات للتعامل مع المتغيرات:

- ◆ خيار التصريح Option Explicit: للدلالة على التصريح عن جميع المتغيرات المستخدمة في البرنامج.
- ◆ خيار التدقيق Option Strict: للدلالة على ان جميع المتغيرات ستكون محددة النوع.
- ◆ خيار الاستنتاج Option Infer: لتمكين المترجم من تحديد نوع المتغير من خلال القيمة المسندة اليه.

لهذه الخيارات الثلاثة تأثير كبير على الطريقة التي تصرح وتستخدم فيها المتغيرات. وستفهم عمل هذه الخيارات من خلال استكشاف هذه الإعدادات. وسوف تفهم أيضا بشكل أفضل كيف يعامل المترجم المتغيرات من الافضل ان تعمل على تفعيل (On) جميع هذه الخيارات. اذا لم تعرف المتغيرات المستخدمة في برنامجك فان المحرر سيضع تحتها خط متعرج. تدل على وجود خطأ وعند تحريك المؤشر فوق الاسم الذي تحته خط سيظهر وصف للخطأ في أداة ToolTip. ولهذا يجب وضع عبارة **Option Explicit Off** في بداية الملف. اي قبل جميع الاحداث حيث سيكون تأثير هذه العبارة على الاجراءات التي تقع على نفس النموذج ولا تؤثر على باقي الاجراءات والاحداث في بقية نماذج المشروع، ويمكن تفعيل هذا الخيار مع بقية خيارات على لكامل المشروع من خلال نافذة خصائص المشروع **Project → Properties** ستظهر نافذة خصائص المشروع اختر تبويب **Compile** كما في الشكل التالي:



يمكن التعامل مع هذه الخيارات برمجياً من خلال الكود بدل التعامل معها من خلال نافذة خصائص المشروع وكما يلي:

Option Explicit Off

Public Class Form1

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

mydate = Today

MsgBox(mydate)

End Sub

End Class

في الاجراء السابق تم تعطيل خيار Option Explicit وبهذا يمكن استخدام المتغيرات دون الاعلان عنها او تحديد نوعها البياني. وبهذا سيعتبر الـ VS من نوع Variant

المتغير من نوع Variant

عندما لا تُحدد بشكل صريح نوع المتغير، فان الفيچوال بيسك سيعطيه نوع البيانات Variant الذي يقبل أرقاماً وتواريخ ونصوصاً. وبهذا لا يتم اجراء اي عملية تحويل للبيانات عندما تُنسب المتغيرات الى النوع Variant لان الفيچوال بيسك سيقوم تلقائياً بعملية التحويل. اذا كنا نعرف مسبقاً النوع البياني الذي سُنسب للتغير فمن الافضل ان نحدد النوع البياني عند تعريف المتغير.

المتغير من نوع Object

يمكن القول بان المتغيرات في الفيچوال بيسك هي أكثر من كونها اسماء او مكان لحفظ القيم. فهي كيانات ذكية ولا تخزن البيانات فقط وانما تعالج قيمها. ربما سمعت بان المتغيرات في الفيچوال بيسك هي كائنات. انظر الى التصريح التالي:

Dim expr As Date

وبهذا يُمكن اسناد قيمة تاريخ للمتغير expr كما يلي:

expr = #2/3/2013#

يمكن وضع تعليمة برمجية تعود بتاريخ جديد من أربع سنوات للأمام كما يلي:

expr.AddYears(4)

حيث يتم اسناد القيمة الجديدة للتاريخ الى متغير اخر من نفس النوع كما في الاجراء التالي:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim expr, newdate As Date
    expr = #2/3/2013#
    newdate = expr.AddYears(4)
    MsgBox(newdate)
End Sub
```

يمكن اضافة عدد من السنوات الى متغير التاريخ وكذلك يمكن اضافة أشهر وأيام، بالاضافة الى الطرق فان النوع تاريخ يعرض خاصيات مثل الأشهر والأيام والتي تعود بشهر التاريخ وعدد الأيام على الترتيب، تماماً مثل الخصائص والطرق للأدوات التي نضعها على النموذج. كل هذا يتم باستخدام دوال اعدت مسبقاً للتعامل مع المتغير من فئة مماثلة للمتغير نفسه.

يُمكنك بناء دوال تحوي متغيرات من نفس نوع المتغير لاستخراج الأشهر من التاريخ او اضافة عدد من الأيام لتكتشف بان الحرف او الرقم او علامة الترقيم وغيرها من الدوال التي نستخدمها في البرنامج تم بناءها من نفس المتغيرات، ولهذا عليك التفكير بالمتغير على انه حافظه للقيم وعندما نريد ان نعالج هذه القيم نكتب اسم المتغير متبوعاً بنقطة لرؤية قائمة بالدوال التي اعدت مسبقاً للتعامل معه.

تستطيع لغات البرمجة معاملة المتغيرات البسيطة بفعالية اكثر من الكائنات، فالعدد الصحيح Integer ياخذ 2 بايت في الذاكرة، كما ان المترجم سيولد كود فعال جدا لمعالجة المتغير الصحيح (اضافة قيمة رقمية، مقارنته بعدد صحيح اخر، ...). يتضح مما سبق بان المتغيرات هي كائنات وهذه ليست مفاجئة ولكنه امر غريب بالنسبة للمبرمجين قليلي الخبرة في البرمجة الكائنية التوجه (OOP) ولهذا يمكن اعتبار أداة صندوق النص TextBox كائن وتوفر الخاصية Text التي تسمح بقراءة او وضع نص في الأداة، وعليه فان اي اسم يُكتب ويتبع بنقطة واسم واخر يدل على كائن والاسم الاخر هو الخاصية او الطريقة للكائن.

تحويل الأنواع البيانية للمتغيرات

في كثير من الاحيان نحتاج الى تحويل المتغيرات من نوع لآخر. الجدول التالي يضم الأنواع البيانية وطرق تحويلها.

النوع البياني	طريقة التحويل
Byte	ToByte
SByte	ToSByte
Boolean	ToBoolean
Char	ToChar
String	ToString
Date	ToDateTime
Decimal	ToDecimal
Double	ToDouble
Short	ToInt16
Integer	ToInt32
UInteger	ToUInt32
Long	ToInt64
ULong	ToUInt64
Single	ToSingle

الإجراء التالي يوضح كيفية تحويل عدد صحيح الى عشري:

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

```

Dim zz As Integer = 20303
Dim xx As Double
xx = Convert.ToDouble(zz / 3)
MsgBox(xx)
End Sub

```

تعلمية الإسناد لمتغير

من خلال هذه التعلمية يتم وضع قيمة معينة في المتغير. وتأخذ هذه التعلمية الشكل التالي:

VariableName = Expression

حيث ان:

- **VariableName**: يمثل اسم المتغير.
- **Expression**: يمثل تعبير يُرجع قيمة من نفس نوع بيانات المتغير.

امثلة:

```

Dim x As Integer = 50      ' 50 بقيمة متغير عن الاعلان
Dim s As String = "Ali"   ' نصي متغير عن الاعلان
x = x * 2                  ' العددي المتغير قيمة مضاعفة
s = s + " Hassan"         ' Ali Hassan قيمة اصبحت

```

الثوابت Constants

في بعض الاحيان نحتاج الى التعامل مع قيم ثابتة لا تتغير في البرامج. كالقيمة 3.14 مثلاً والتي نستخدمها في الحسابات المثلثية. يُمكن تعريف الثوابت باستخدام التعلمية:

Const Constname = Expression

اليك الاجراء التالي:

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim x As Single
    Dim r As Integer = InputBox("بيانات ادخال", "الدائرة قطر نصف قيمة ادخل")
    Const pi = 3.14
    x = 2 * pi * r
    MsgBox(x)
End Sub

```

في الاجراء السابق نلاحظ ان المتغيران x, r يمكن ان تتغير قيمهم مع كل عملية تنفيذ حسب القيم المدخلة. اما بالنسبة لقيمة المتغير Pi فهي ثابتة ولا تتغير عند ك عملية تنفيذ.

مجال تعريف المتغير وحياة المتغير

عند تعريف متغير داخل اجراء معين، فان التعلميات الموجودة ضمن هذا الاجراء فقط يمكنها ان تقرأ او تُعدل قيمة هذا المتغير، ويوصف هذا المتغير بانه متغير محلي Local (اي خاص بهذا الاجراء فقط). ويمكن ان تستخدم عدة اجراءات نفس اسم المتغير لانه

لا توجد اي علاقة بين المتغيرات المحلية لاجراء والمتغيرات المحلية لاجراء آخر. كذلك يُمكن ان نجعل هذا المتغير يستخدم في جميع اجراءات النموذج أو الوحدات النمطية أو في جميع إجراءات المشروع من خلال ما يلي:

- Dim داخل الإجراء: يعني ان المتغير محلي Local خاص بالاجراء فقط ويمكن أن تخزن فيه قيم بسيطة.
- Dim في قسم التصريحات العامة لصفحة برمجة النموذج لنموذج معين: يعني أن المتغير يمكن ان يستخدم في جميع إجراءات نفس النموذج وبهذا يعتبر المتغير متغير نموذج.
- Public في قسم التصريحات العامة لنموذج: يعتبر المتغير عام ويمكن أن يستخدم في جميع إجراءات المشروع وبهذا يجب ان يسبق المتغير باسم النموذج عند استخدامه في نموذج آخر (Formname.Variablename).
- Public في قسم التصريحات العامة لوحدة نمطية: متغير عام يُمكن استخدامه في جميع إجراءات المشروع حيث يكون مجال تعريفه اوسع من كل الانواع السابقة.

ملاحظات

1. لجعل عملنا أكثر تنظيم يفضل وضع جميع المتغيرات العامة في وحدة نمطية مخصصة لتسهيل العودة اليها. وتجنب تعريف نفس المتغير العام في وحدتين نمطيتين مختلفتين لتجنب الاخطاء المحتملة الحدوث اثناء ترجمة البرنامج (Compiler).
2. يمكن استخدام نفس اسم المتغير في اكثر من مجال، يمكن ان يكون لدينا متغير عام باسم Sum ومتغير محلي باسم Sum أيضاً. داخل الاجراء تستخدم المتغير المحلي وخارج الاجراء نستخدم المتغير العام لان المتغير يعود للتعريف الاقرب.

المتغيرات الساكنة Static Variable

لكل متغير يستخدم في المشروع مدة حياة، فالمتغيرات العامة Global تُحفظ طيلة فترة فتح المشروع. أما المتغيرات المحلية Local لاجراء فتوجد خلال فترة تنفيذ الاجراء، وعند الانتهاء من تنفيذ الإجراء تضيع قيم هذه المتغيرات وتُحرر مواقع الذاكرة المحجوزة لها خلال تنفيذ الإجراء وعند تنفيذ الاجراء مرة اخرى يعاود الفيچوال بيسك حجز اماكن لها في الذاكرة.

يمكن ان نجعل الفيچوال بيسك يحافظ على قيمة المتغير من عملية استدعاء لآخر للإجراء من خلال تعريف المتغير باستخدام التعليمة Static التي تماثل تعليمة Dim في الاستخدام.

الاجراء التالي يقوم بعدد مرات النقر على زر أمر عند تنفيذ برنامج معين

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Static combine As Integer
    combine += 1
    MsgBox(combine)
End Sub
```

المتغير الساكن (combine) في البرنامج السابق هو متغير محلي Local، ولو استخدمنا في البرنامج السابق التعليمة Dim بدل من Static سيقوم باظهار 1 دائماً.

كذلك سنحصل على نفس النتيجة عند تعريف المتغير combine كمتغير عام. وبهذا سنمنح باقي الاجراءات فرصة الوصول لهذا المتغير وتعديله.

يمكن جعل جميع متغيرات اجراء معين ساكنة من خلال وضع الكلمة Static في بداية الاجراء Sub وكذلك الحال بالنسبة للدوال Function.

العمليات الحسابية

تتوفر في لغة الفيچوال بيسك العمليات الحسابية التالية:

الجمع (+):

$$6+7=13$$

“Hello”+” dad”=”Hello dad”

الطرح (-):

$$5-3=2$$

الضرب (*):

$$5*7=35$$

القسمة (/):

$$8 / 2 = 4$$

القسم الصحيح من القسمة (\):

$$7 \setminus 4 = 1$$

باقي القسمة (mod):

$$7 \bmod 3 = 1$$

رفع القوة (^):

$$7 ^ 3 = 343$$

العمليات المنطقية

تُستخدم هذه العمليات مع المتغيرات المنطقية أو التعبيرات المنطقية كالشروط كما سنرى لاحقاً.

(OR):

False OR False = False

False OR True = True

True OR False = True

True OR True = True

(AND):

False AND False =False

False AND True =False

True AND False = False

True AND True = True

(NOT):

NOT False = True

NOT True =False

البنى الشرطية

تُستخدم بنى الشرط لتحديد فيما إذا كانت مجموعة من التعليمات ستُنفذ أم لا، وذلك حسب قيمة تعبير معين. هنالك عدة أشكال لبنى الشرط:

- If... Then
- If... Then... Else
- If... Then... ElseIf... Else
- If... Then...ElseIf...ElseIf...Else
- IIf(Expression , True , False)
- SelectCase

- If... Then

تُستخدم هذه الجملة الشرطية لتنفيذ تعليمة أو مجموعة تعليمات وفق شرط معين. يُمكن كتابة هذه الجملة الشرطية على سطر كما يلي:

If condition Then statement(s)

حيث ان:

- **Condition**: يمثل الشرط
- **Statement(s)**: يمثل تعليمة برمجية أو أكثر.

يُمكن كتابة جملة الشرط بشكل اخر كالتالي:

If condition Then

statement(s)

End If

يُمكن ان يكون الشرط عبارة عن تعبير منطقي (مقارنة مثلاً) يُرجع إما صواب True أو خطأ False. يُمكن الشرط أن يكون تعبير رقمي يُعتبر الشرط محقق True إذا كانت نتيجة التعبير لا تساوي صفراً وإلا فان الشرط غير محقق False.

يقوم الفيچوال ببيسك بتنفيذ التعليمات المرافقة للشرط المتحقق. كما في الأمثلة التالية:

If n>m Then m=n

و

If n>m Then

M=n

End If

- If... Then...Else

في هذه الحالة اذا تحقق الشرط سُنفذ تعليمات معينة واذا لم يتحقق الشرط سُنفذ تعليمات أخرى.

If condition **Then** statement(s) **Else** statement(s)

وكذلك يمكن كتابتها بالشكل التالي:

If condition **Then**

statement(s)

Else

statement(s)

End If

If... Then...ElseIf...Else –

أو الشكل التالي:

If... Then...ElseIf...ElseIf...Else –

نلجأ الى استخدام احد الشكلين اذا كان لدينا عدة كتل من التعليمات البرمجية، وكتلة واحد من بينها هي التي سننفذ.

يمكن كتابة هذه الاشكال على اسطر متعددة بدل السطر الواحد. كما يلي:

If condition **Then**

Statement(s)

ElseIf condition **Then**

Statement(s)

Else

Statement(s)

End If

أو

If condition **Then**

Statement(s)

ElseIf condition **Then**

Statement(s)

.
. .
. .

Else

Statement(s)

End If

يقوم الفيچوال ببيسك باختبار شرط If اذا كان غير محقق False سينتقل لاختبار شرط ElseIf وهكذا يستمر الى ان يتحقق الشرط عندها سيُنفذ كتلة البيانات (Block) التابعة للشرط المتحقق ثم ينتقل للتعليمات التي تلي End If (اي انه لا يفحص اي شرط من الشروط التي تلي الشرط المتحقق وبهذا فان كتلة واحدة من التعليمات ستُنفذ).

يمكن وضع Else سواء استخدمنا ElseIf ام لا .

برنامج لمعرفة نتيجة طالب:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim z As Integer = InputBox("قيمة ادخل")
    Dim ss As String
    If z < 50 Then
        ss = "Fail"
    ElseIf z < 60 Then
        ss = "Accept"
    ElseIf z < 70 Then
        ss = "Med"
    ElseIf z < 80 Then
        ss = "VeryGood"
    Else
        ss = "excellent"
    End If
    MsgBox(ss)
End Sub
```

يقوم البرنامج السابق بفحص القيمة المُدخلة z في اكثر من مرة.

والانسب في مثل هذه الحالة استخدام Select Case.

IIf(Expression , True , False) –

تقوم هذه التعليمة الشرطية بفحص التعبير **Expression** ، فاذا كان الشرط متحقق فانها تُعيد True، اما اذا كان الشرط غير متحقق فانها تُعيد False.

في البرنامج التالي اذا كانت قيمة x أكبر من صفر فان الدالة IIf ستُعيد "positive" واذا كانت قيمة x أصغر من صفر فانها تُعيد "Negative".

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim z As Integer = InputBox("قيمة ادخل")
    Dim ss As String
    ss = IIf(z > 0, "positive", "Negative")
    MsgBox(ss)
End Sub
```

Select Case –

يُقدم الفيچوال ببسك تعليمة Select Case خيار آخر بدلاً من التعليمة If... Then...ElseIf...ElseIf...Else لاختيار كتلة من التعليمات (Block) من بين مجموعة من الكتل. حيث تُقدم تعليمة Select Case امكانيات مشابهة لتعليمة If... Then...ElseIf...ElseIf...Else إلا انها تجعل التعليمات البرمجية أكثر وضوحاً وأسرع في التنفيذ.

تقوم هذه التعليمة بفحص تعبير واحد testexpression في بداية التعليمة ومن ثم مقارنة نتيجة التعبير مع مجموعة من التعابير (لكل تعبير Case) وعندما تتطابق testexpression مع احد التعابير expressionlist فانها ستنفذ كتلة التعليمات (Block) المرتبطة بالـ Case المتحققة.

Select Case testexpression

Case expressionlist1

Statement(s)1

Case expressionlist2

Statement(s)2

Case Else

Statement(s)n

End Select

تأخذ expressionlist أحد الأشكال التالية:

- قيم expression
- مجال القيم To expression
- مقارنة Is Compare Expression

(حيث ان Compare هو معامل مقارنة = , <= , >= , < , >)

- كل statement(s) عبارة عن لاشيء أو تعليمة أو أكثر.
- اذا تطابقت عدة قيم Case قيمة التعبير، فان كتلة التعليمات الموافقة لـ Case الأولى هي فقط التي ستنفذ.
- اذا لم تطابق قيمة التعبير لاي من قيمة Case، عندها سيُنفذ الفيچوال ببسك التعليمات البرمجية الموجودة في Case Else البرنامج التالي لاختبار اي عدد اذا كان موجب او سالب.

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim z As Integer = InputBox("قيمة ادخل")
    Dim ss As String
    Select Case z
        Case Is > 0
            ss = "positive"
        Case Is < 0
            ss = "Negative"
        Case Else
            ss = "no sign"
    End Select
    MsgBox(ss)
End Sub
```

في البرنامج اعلاه تقوم Select Case بحساب قيمة تعبير الاختبار مرة واحدة، بعكس تعليمة If...ElseIf...Esle يتم حساب قيم مختلفة في كل ElseIf ويمكن استبدال If...ElseIf...Esle بـ Select Case يحدث هذا في بعض الحالات وليس في جميع الحالات.

جمل التكرار Loop

تستخدم جمل التكرار لتنفيذ مجموعة من التعليمات بشكل متكرر.

جمل التكرار في الفيچوال بيسك هي:

- For ... Next
- While... Loop
- Do While... Loop
- Do Until... Loop

For ... Next -

تستخدم هذه التعليمة لتنفيذ تعليمة معينة عدد من المرات بالاعتماد على القيمة النهائية للعداد الخاص بها، مع هذه الجملة التكرارية يكون الاجراء اكثر فاعلية بخلاف Do. حيث تستخدم For متغير يسمى عداد الحلقة يزداد او ينقص في كل مرة من مرات تكرار التعليمات. شكلها العام:

For counter = initial_vlaue to end_value [step]

Statement(s)

Next

حيث ان:

- Counter: عداد الحلقة التكرارية.
- initial_vlaue : القيمة الابتدائية لعداد الحلقة التكرارية.
- end_value: القيمة النهائية لعداد الحلقة التكرارية.
- Step: مقدار التغير في قيمة العداد (زيادة أو نقصان) بعد كل تنفيذ.

ملاحظات:

- يمكن ان تكون step موجبة أو سالبة، اذا كانت موجبة يجب ان تكون قيمة initial_value أصغر من أو تساوي end_value، وإلا فان التعليمات البرمجية للحلقة لن تُنفذ. اما اذا كانت سالبة يجب ان تكون قيمة initial_value أكبر من أو تساوي end_value ، وإلا فان التعليمات البرمجية للحلقة التكرارية لن تُنفذ.
- عادة اذا كان التغير في قيمة عداد الحلقة هو الزيادة بمقدار واحد، فلا توجد ضرورة لكتابة step لان الفيچوال بيسك سيعتبرها تلقائياً زيادة بمقدار واحد.
- يمكن استخدام التعليمة Exit For لايقاف تكرار الحلقة عند تحقق شرط معين.

خطوات تنفيذ الحلقة For

- 1- نضع لعداد الحلقة counter قيمة ابتدائية.
- 2- تدقيق قيمة العداد اذا كانت اصغر من end_value فتكون step موجبة، واذا كانت أكبر من end_value يقفز التنفيذ الى ما بعد next ، اذا كانت step سالبة يدقق اذا كانت قيمة العداد اصغر من end_value).
- 3- تنفيذ تعليمات الحلقة.

4- يزيد قيمة العداد بمقدار واحد اذا لم يكن step محدد، او يزيده او ينقصه بمقدار step.

5- سكرر تنفيذ الخطوات من 2 الى 4.

برنامج لطباعة الاعداد من 1 الى 5

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    For i As Integer = 1 To 5
        MsgBox(i)
    Next
End Sub
```

While... Loop -

تستخدم تعليمات التكرار (Loop) لتكرار تنفيذ كتلة (Block) من التعليمات عدداً غير محدد من المرات، وهناك عدة اشكال لهذه التعليمات ولجميعها يكون هنالك شرط يحدد استمرار التكرار أم لا. كما هو الحال مع تعليمة If، والشرط إما يكون تعبير منطقي يُرجع True أو False أو تعبير يُرجع قيمة رقمية (اذا كانت لا تساوي صفرأ True وإلا False).

في الشكل التالي للـ While حيث يتم فحص الشرط قبل تنفيذ كتلة التعليمات (Block) اذا كانت الشرط متحقق يستمر بالتنفيذ، ويتوقف التنفيذ اذا كان الشرط غير متحقق.

Initial_value

While condition

Statement(s)

End While

يبدأ الفيچوال ببسك باختبار الشرط condition اذا كان متحقق True يستمر بتنفيذ التعليمات داخل الحلقة التكرارية While، واذا لم يكن متحقق False سيقفز الى السطر الذي يلي End While، ويعود مرة أخرى الى ليتحقق من صحة الشرط ليُنفذ التعليمات وهكذا الى ان يصل الى مرحلة يصبح فيها الشرط False (غير متحقق). فيخرج من الجملة التكرارية.

مثال يوضح كيفية استخدام هذه الحلقة:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim i As Integer = 1
    While i < 5
        MsgBox(i)
        i += 1
    End While
End Sub
```

ملاحظة: يمكن استخدام التعليمة Exit While لكسر تكرار حلقة الـ While عند تحقق شرط معين.

Do While... Loop -

تُنفذ التعليمات البرمجية اذا كان الشرط condition متحقق، ويتوقف التكرار عندما يصبح الشرط غير متحقق. كما في الشكل التالي:

Initial_value

Do While condition

Statement(s)

Loop

يمكن كتابة المثال السابق بالشكل التالي:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim i As Integer = 1
    Do While i < 5
        MsgBox(i)
        i += 1
    Loop
End Sub
```

هنالك شكل اخر للجملة التكرارية Do While كما يلي:

Initial_value

Do

Statement(s)

Loop While condition

يقوم الفيچوال ببسك بتنفيذ تعليمة واحدة ومن ثم يختبر شرط condition اذا كان الشرط متحقق True يعود الى العبارة التي تلي Do ليستمر بالتنفيذ واذا كان الشرط غير متحقق False سيقفز التنفيذ الى ما بعد While.

البرنامج التالي يوضح عملها:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim i As Integer = 1
    Do
        MsgBox(i)
        i += 1
    Loop While i < 5
End Sub
```

Do Until... Loop –

شكلها العام:

Do Until condition

Statement(s)

Loop

يتم تنفيذ التعليمات الخاصة بالـ Do Until عندما يكون الشرط condition غير متحقق False، اما اذا كان الشرط متحقق True سيقفز تنفيذ البرنامج الى ما بعد Loop.

وبهذا نستدل على ان Do Until تعمل عندما يكون الشرط غير متحقق والعكس صحيح.

برنامج يوضح عملها:

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

```

Dim i As Integer = 1
Do Until i > 5
    MsgBox(i)
    i += 1
Loop
End Sub

```

في البرنامج السابق تم الاعلان عن متغير واسناد قيمة ابتدائية له، ثم يقوم البرنامج بالتحقق من الشرط condition اذا كان الشرط True ينتقل التنفيذ الى ما بعد loop، اما اذا كان الشرط غير متحقق هنالك شكل اخر للـ Do Until.

Do

Statement(s)

Loop Until condition

في الشكل أعلاه يتم تنفيذ التعليمات اولاً ثم يختبر الشرط اي سنُنفذ على الأقل مرة واحدة ، يتوقف التكرار عندما يصبح الشرط محققاً: برنامج جمع الاعداد المحصورة بين 1 و 5:

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim sum = 0, i As Integer = 1
    Do
        sum += i
        i += 1
    Loop Until i > 5
    MsgBox(sum)
End Sub

```

ملاحظة: يمكن استخدام عبارة Exit Do لكسر التكرار عند تحقق شرط معين.

العبارات المتداخلة

تتطلب الحاجة في بعض البرامج الى استخدام عبارة شرطية مع جملة تكرارية لاداء غرض معين او يتطلب الامر استخدام اكثر من جملة تكرار في نفس البرنامج كما سيمر علينا في البرامج التالية:

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim x = 0, i As Integer = 1
    Do While i < 10
        If i Mod 2 = 0 Then
            x += i
        End If
        i += 1
    Loop
    MsgBox(x)
End Sub

```

في البرنامج اعلاه يتم جمع الاعداد الزوجية المحصورة بين 1 و 10، يتم المرور على الاعداد من خلال الحلقة التكرارية Do While، في جملة الشرط If يتم التحقق من العدد اذا كان زوجي ليتم جمعه. وفي النهاية يتم عرض النتيجة في صندوق الرسائل.

في البرنامج التالي سنجد ان جملة الشرط Else... Else If داخل جملة شرط If... Else اخرى للتحكم في اظهار واخفاء صندوق النص على النموذج، اذا كان اسم المستخدم المدخل في المتغير "username" هو "Admin" سيظهر صندوق النص على النموذج، عندما يكون الاسم المدخل "customer" يتم اخفاء الصندوق من سطح المكتب.

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim username As String = TextBox1.Text
    If username = "Admin" Then
        TextBox2.Visible = True
    Else
        If username = "customer" Then
            TextBox2.Visible = False
        Else
            MsgBox("Error, try Again")
        End If
    End If
End Sub
```

الخصائص Properties

هي صفات تخص الكائنات يتم تحديدها (بعد تكوين الكائن) اما من الخيارات ضمن نافذة الصفات خلال فترة التصميم او من خلال الشفرة البرمجية مثلاً من صفات صندوق النص هي حجم الصندوق وقيمة النص ولون الخط وغيرها، عند كتابة الكود التالي:

```
TextBox1.text="Welcome"
```

حيث ان TextBox1 هو اسم الكائن (يمثل صندوق النص) و Text هي احدى الصفات الخاصة بصندوق النص و Welcome هي القيمة التي ستظهر داخل الصندوق والتي تخزن في صفة Text وبهذا فان صفات الكائنات تكون كما يلي:

Object.properties=Value

حيث ان:

Object: يمثل الكائن.

Properties: صفة الكائن.

Value: تمثل قيمة الصفة.

الأحداث Events

هي الافعال الخارجية التي تطبق على الكائنات والتي تنتج عنها ردود أفعال وتتحقق عند تنفيذ الشفرة ومن اكثر الاحداث استخداماً هو حدث النقر بالفأرة (Click)، ويتعرف النموذج والادوات على الاحداث تلقائياً وتتجاوب معها لكون هذه الاحداث مُعرفة مسبقاً. مثلاً عند النقر بالفأرة على كائن زر الأمر يؤدي الى مسح النص الموجود داخل صندوق النص. كما يلي:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    TextBox1.Text = ""
End Sub
```

حيث ان Button1 زر امر والحدث الذي يطبق عليه هو النقر والذي سينتج عن عملية النقر Click إعطاء قيمة فارغة لصفة Text الخاصة بكائن صندوق النص.

الفئة Class

هي جزء من أجزاء البرنامج الذي تقوم بإنشائه والبرنامج هو مجموعة classes وكل Class يحتوي على خصائص الجزء الذي يمثله وقد تكون هذه الخاصية شكل أو سلوك أو وظيفة ومجموعة الفئات Class هي التي تعطي غي النهاية المشروع الذي سيصبح برنامجاً فيما بعد وليس من الضروري أن يكون Class شئ نراه في صورة مادية مثل Toolbox فمن الممكن ان يكون سطر من الاوامر والفئات تخزن في جزء يسمى Microsoft Intermediate Language وتختصر MSIL وهي لغة بسيطة يمكن لاي لغة من لغات الفيجوال بيسك والتعامل معها بلا اي وسيط. وعندما تعمل على الفيجوال بيسك يمكنك استدعاء أحد الفئات Classes الموجودة في Primary Microsoft Dotnet Framework Components وبعد ان تنتهي من العمل فان تحويل نواتج العمل مرة اخرة الى اللغة الوسيطة (MSIL) عن طريق Common Language Runtime وتختصر CLR.

صفحة برمجة النموذج

تُخزن إجراءات أحداث النموذج والادوات الموضوعية على سطحه في صفحة برمجة النموذج، وتعتبر جزء من النموذج وتُحفظ معه في نفس الملف.

تضم صفحة برمجة النموذج على قسم التصريحات العامة (أعلى الصفحة) ويحوي هذا القسم على:

- 1- التعليمات Option Explicit On/Off لجعل تعريف المتغيرات واجب أم لا.
- 2- تعريف المتغيرات Variables والثوابت Constant وجعلها عامة على مستوى النموذج (اي يُمكن استخدامها في جميع إجراءات النموذج).
- 3- تعريف البرامج الفرعية وجعلها عامة على مستوى النموذج حيث يُمكن استخدامها في جميع إجراءات النموذج. وعادةً لا ترتبط هذه البرامج بأداة تحكم او حدث. البرامج الفرعية إما تكون إجراء Sub (لا تُرجع قيمة) أو دالة Function (تُرجع قيمة).

تسمية إجراءات الأحداث

يقوم الفيجوال بيسك 2013 بإنشاء قالب إجراء الحدث ويقوم بإعطاء اسم للإجراء عندما نطلب بناء إجراء حدث على أداة تحكم. وبهذا لا يستوجب علينا القيام بتسمية إجراءات الأحداث.

يقوم الفيجوال بيسك بوضع اسماء لاجراءات الاحداث كما يلي:

- 1- إجراءات أحداث النموذج حيث يتألف الاسم من كلمة Form ثم الإشارة () متبوع باسم الحدث Eventname.

```
Private Sub Form1_eventname(sender As Object, e As EventArgs) Handles MyBase.eventname  
Statement(s)  
End Sub
```

- 2- إجراءات أدوات التحكم على سطح النموذج فان اسم الاجراء يتألف من اسم اداة التحكم Toolname ثم الاشارة () ثم اسم الحدث eventname.

```
Private Sub toolname_eventname(sender As Object, e As EventArgs) Handles toolname.eventname  
statement(s)  
End Sub
```

الكلمتان Sub,End Sub تُحددان بداية ونهاية الاجراء، الكلمة التي تتبع Sub تمثل اسم الاجراء

statement(s) تمثل التعليمات البرمجية المطلوب تنفيذها عند وقوع الحدث.

Private تدل على ان الاجراء خاص يمكن استدعاؤه من اجراءات على نفس النموذج.

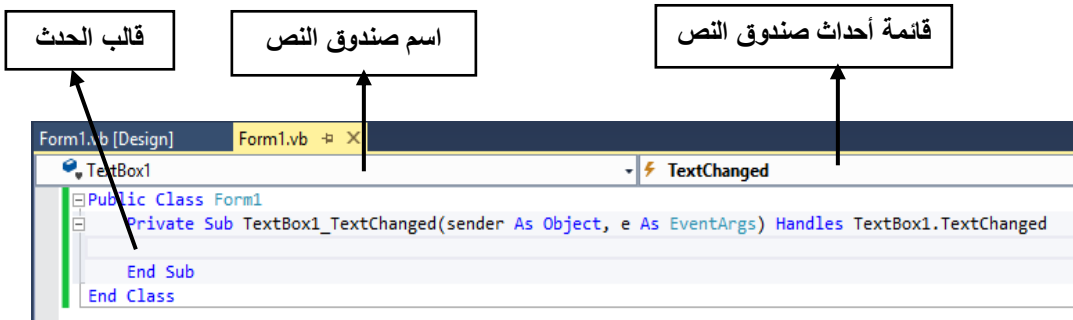
ملاحظة: يجب تغيير اسم الكائنات (أدوات التحكم) قبل كتابة إجراءات الأحداث على أدوات التحكم. عند تغيير اسم أداة تحكم بعد كتابة إجراء حدث معين يخصص تلك الأداة عليك أن تقوم بتغيير أسماء الإجراءات المكتوبة وإلا فلن يتمكن الفيجوال بيسك من مطابقة أداة التحكم مع الإجراءات الموافقة المكتوبة.

لكتابة إجراء حدث معين على أداة تحكم ننقر نقرأ مزدوجاً على أداة التحكم فيقوم الفيجوال بيسك بنقلنا الى صفحة برمجة النموذج ويُنشئ قالباً لإجراء الحدث الافتراضي لأداة التحكم ثم نقوم بكتابة الشفرة البرمجية المطلوبة داخل هذا الاجراء.

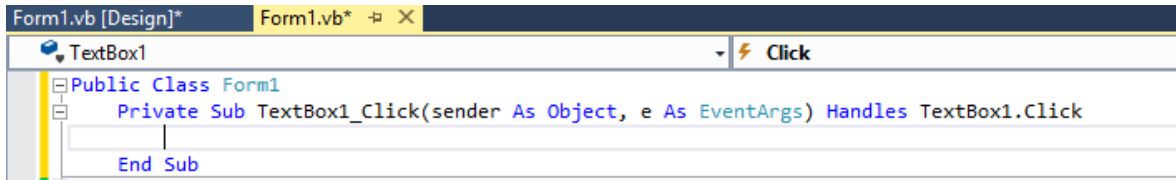
يمكن اختيار حدث جديد من قائمة الأحداث لأداة التحكم فيقوم الفيجوال بيسك بإنشاء قالب جديد للحدث المختار.

مثلاً نريد كتابة إجراء لحدث النقر Click لصندوق النص TextBox.

- عند النقر المزدوج على صندوق النص يقوم الفيجوال بيسك بفتح صفحة برمجة النموذج ويُنشئ قالباً للحدث الافتراضي لصندوق النص TextBox هو TextChanged.



- من قائمة الأحداث نختار حدث النقر Click فينشئ الفيجوال بيسك قالباً للحدث، نكتب ضمن القالب التعليمات البرمجية.



البرامج الفرعية Procedures

هي عبارة عن شفرات برمجية يتم كتابتها لاداء غرض معين. يتم تنفيذ هذه الشفرات عند استدعاؤها بشكل صريح. للبرامج الفرعية الكثير من الفوائد تتمثل بتجنب تكرار كتابة كود معين في أكثر من مرة.

تُستخدم البرامج الفرعية اذا كان هنالك عدة اجراءات تقوم بنفس التعليمات فنضع هذه التعليمات في برنامج فرعي واحد ونجعل جميع الاجراءات الاخرى تستدعي هذا البرنامج وبهذا يتم القضاء على تكرار التعليمات وتسهيل التعديل عليها لاحقاً.

تُكتب البرامج الفرعية إما في صفحة برمجة النموذج أو في الوحدات النمطية. اذا كُتب البرنامج الفرعي في وحدة نمطية يتم استدعاؤه في جميع اجزاء المشروع (اي في جميع الاجراءات الموجودة في نماذج المشروع).

أما اذا كتبنا البرنامج الفرعي في صفحة برمجة النموذج فيمكن استدعاؤه في اجراءات الخاصة بنفس النموذج بذكر اسم البرنامج الفرعي فقط وكذلك يُمكن استدعاؤه في إجراءات النماذج الاخرى لنفس المشروع من خلال كتابة اسم النموذج متبوعاً باسم البرنامج الفرعي (Formname.Procedurename).

تتكون البرامج الفرعية من إجراءات Sub أو دوال Function.

- الإجراءات Sub: لا تُرجع قيمة، ويتم استدعاؤها بذكر اسمها فقط.

- الدوال Function: تُرجع قيمة، وتستخدم في التعبيرات. وتكون القيمة المُرجعة (رقمية أو نصية أو منطقية أو غيرها).

الإجراء Sub

يمكن التصريح عن الاجراء داخل Class النموذج او في الوحدة النمطية كما في الشكل التالي:

[Public | Private | Friend] Sub name([argument])

Statement(s)

Exit Sub

Statement(s)

End Sub

حيث ان:

- **Public**: جعل الاجراء عام، اي يُمكن استدعاؤه في جميع اجزاء المشروع (وهي الحالة الافتراضية اذا لم نكتب شيء).
- **Private**: جعل الاجراء خاص، اي لا يُستدعى الا في مكان تعريفه (صفحة برمجة النموذج أو الوحدة النمطية).
- **Friend**: يُمكنك من الوصول الى جميع أنواع والاعضاء التي تم وضع علامة الصديق عليها.
- **Name**: يُمثل اسم الاجراء.
- **Statement(s)**: تمثل التعليمات البرمجية التي ستنفذ عند استدعاء الاجراء.
- **Exit Sub**: تعليمة كسر الإجراء.
- **Argument**: تمثل المتغيرات المدخلة للإجراء ويفصل بينهما (,) اذا كان هنالك أكثر من متغير. وتعرف المتغيرات في الإجراء كما يلي:

[ByRef | ByVal] variablename [As Type]

حيث ان:

- **ByVal**: طريقة التمرير للمتغير هي تمرير القيمة Value.
- **ByRef**: طريقة التمرير للمتغير هي تمرير العنوان Reference. وهي الطريقة الافتراضية اذا لم نكتب شيء.
- **Type**: تمثل النوع البياني للمتغيرات المدخلة للإجراء كـ (Integer, Long, Double, String, Currency, ...) او Variant أو كائن Object إذا ل تُحدد النوع البياني.

استدعاء الإجراء:

[Call] SubName [argument]

Call: يُمكن الاستغناء عن استخدامها، واذا استخدمت يجب وضع متغيرات الادخال ضمن قوسين ().

Subname: اسم الاجراء.

Argument: متغيرات الادخال الخاصة بالاجراء.

اليك الاجراء التالي الذي يقوم بطباعة رسالة حسب القيمة المُدخلة للاجراء في المتغير x.

```
Sub showmsg(ByVal x As String)
    MsgBox(x)
End Sub
```

يكون استدعاء الإجراء كما يلي:

```
Call showmsg(n)
```

الدالة Function:

يُصرح عادة عن الدالة في Class الخاصة بالنموذج او في الوحدة النمطية Module وتأخذ الشكل التالي:

```
[ Public | Private | Friend ] Function name ([argument]) As Type
```

```
Statement(s)
```

```
Exit Function
```

```
Statement(s)
```

```
End Function
```

تعمل متغيرات الادخال للدالة عمل متغيرات ادخال الإجراء. لكن يُمكن تمييز الدالة عن الإجراء بما يلي:

- تُنسب القيم المُرجعة الى اسم الدالة وبهذا يُمكن استخدامها في التعبيرات.
- يجب وضع أقواس بعد اسم الدالة عند استدعائها.
- للدالة نوع بياني يُحدد نوع القيمة المُرجعة.

مثلاً يمكن ان نستخدم دالة لسحاب نصف قطر الدائرة.

```
Function radcir(pi As Single, r As Integer) As Long
    radcir = 2 * pi * r
End Function
```

يتم استدعاء الدالة بالطريقة التالية:

```
Dim z As Long
z = radcir(x, y)
```

بالنظر لما سبق يُمكن ان نستنتج ما يلي:

- البرامج الفرعية عبارة عن إجراءات Sub أو دوال Function.
- خاصة Private يُمكن استدعاؤها في المكان الذي أنشئت فيه وفي كل إجراءات صفحة برمجة النموذج على نفس النموذج.
- عامة Public وبهذا يُمكن استدعاؤها في جميع أجزاء المشروع. كذلك اذا لم تُحدد تُعتبر عامة. وتعرف في هذه الحالة في وحدة نمطية Module.
- لا يُمكن تعريف برنامجين عامين بنفس الاسم حتى اذا كانا في وحدتين نمطيتين مختلفتين.
- يتم استدعاء البرنامج العام Public المعرف في صفحة برمجة النموذج من نموذج آخر عن طريق ذكر اسم النموذج متبوع باسم البرنامج.

```
Formname.Procedurename
```

- أما اذا كان البرنامج معرف في وحدة نمطية Module فيتم استدعاءه عن طريق اسمه فقط.

المصفوفات Arrays

هي عبارة عن سلسلة من المتغيرات تحمل جميعها نفس الاسم والنوع البياني ولكل وقع فيها فهرس Index متسلسل. لكل مصفوفة حد أدنى وحد أعلى وتكون القيم بين هذين الحدين متجاورة. تقوم الفيچوال ببسك بحجز مواقع في الذاكرة لجميع عناصر المصفوفة. ولهذا يجب اعطاء طول للمصفوفة يتناسب مع ما نحتاجه.

تحمل جميع عناصر المصفوفة نفس النوع البياني. وعندما تكون المصفوفة من النوع Variant فيكون لكل عنصر من عناصر المصفوفة نوع مختلف من البيانات (عدد صحيح، نص، تاريخ) فيمكن تعريف مصفوفة من كل الانواع البيانية.

عند تعريف المصفوفة نتبع اسمها بالحد الأعلى لها بين قوسين. على ان يكون الحد الأعلى لها عدداً صحيحاً. لاحظ التعاريف التالية للمصفوفات.

الاعلان عن مصفوفة تضم 10 عناصر تبدأ من 0 الى 10, Dim x(10) as integer

مصفوفة تضم 20 عنصر (0 الى 20) , Dim a(20)as single

التعريف الاول سيخلق مصفوفة تتكون من 10 عناصر (اعداد صحيحة) ويبدأ ترقيم مواقع عناصر المصفوفة من 0 الى 10
التعريف الثاني سيخلق مصفوفة تضم 20 عنصر (عدد عشري) حيث يبدأ ترقيم مواقع عناصر المصفوفة من 0 الى 20 وبهذا نلاحظ ان الحد الادنى الافتراضي للمصفوفة هو صفر.

تستخدم الحلقات التكرارية في أغلب الأحيان كطريقة فعّالة لمعالجة المصفوفات. مثلاً ، الحلقة التكرارية التالية تعطي قيم اولية الى عناصر المصفوفة اكبر بمقدار واحد من قيمة موقع كل عنصر.

```
Dim x(5) As Integer
For i As Integer = 0 To 5
    x(i) = i + 1
Next
```

بالنظر لما سبق يمكن ان نلاحظ بان الحد الأعلى الموضوع بين قوسين للمصفوفة هو رقم صحيح واحد لذلك يُسمى هذا النوع من المصفوفات بالمصفوفات الأحادية البعد **One-Dimensional Arrays** وتكون عبارة عن مصفوفة صف أو عمود.

المصفوفات المتعددة الأبعاد Multi-Dimensional Arrays

تتألف هذه المصفوفات من صفوف و أعمدة ويتم الاعلان عنها بالطريقة التالية:

A(5,5) As Integer

حيث يكون الحد الأدنى الافتراضي للصفوف هو صفر. وكذلك الحال بالنسبة للأعمدة ايضاً يكون صفراً.

تُعالج المصفوفات المتعددة الأبعاد بفعالية باستخدام الحلقات التكرارية For. في المثال التالي تقوم الحلقات التكرارية باسناد قيم لكل عنصر في المصفوفة a حسب موقع كل عنصر.

```
Dim a(5, 5) As Integer
For i As Integer = 0 To 5
    For j As Integer = 0 To 5
        a(i, j) = i + j + 5
    Next
Next
```

المصفوفات الديناميكية Dynamic Arrays

أحياناً، قد لا يكون لدينا علم مسبق بطول المصفوفة التي نحتاجها، فبالإمكان تغيير طول المصفوفة أثناء التنفيذ. يمكن تغيير حجم المصفوفة الديناميكية أثناء التنفيذ. وتعتبر المصفوفات الديناميكية من بين أكثر الميزات في الفيجوال بيسك مرونة وفعالية، حيث يمكن ان نستخدم مصفوفة ضخمة لفترة زمنية قصيرة ومن ثم نقوم بتحرير مواقع الذاكرة المحجوزة لها عندما لا نعود بحاجة إليها.

كما ان تعريف مصفوفة بحجم كبير سيؤدي الى حجز مكان كبير في الذاكرة بلا فائدة وتؤدي الى إبطاء سرعة البرنامج.

لتعريف مصفوفة ديناميكية نتبع ما يلي:

1- يتم تعريف المصفوفة باستخدام التعليمة Dim أو Static مع أبعاد فارغة.

Dim a() As Integer

2- نُعيد حجز عدد حقيقي من العناصر عند الحاجة باستخدام التعليمة ReDim. وتعتبر تعليمة حجز تنفيذية، تجعل البرنامج يقوم بحجز مواقع في الذاكرة عند كل تنفيذ للتعليمة، كما يمكن لتعليمة ReDim أن تُغير عدد العناصر الكلي للمصفوفة أو الحد الأدنى أو الحد الأعلى.

في المثال التالي ، يتم خلق مصفوفة ديناميكية a من خلال تعريفها في الوحدة النمطية:

Dim a() As Integer

بعد ذلك يتم حجز حجم محدد للمصفوفة باستخدام دالة أو اجراء:

Sub | Function calcvalue() As Integer

:

ReDim a(5,6)

:

End Sub | Function

دوال فحص البيانات

يمكن تجنب بعض أخطاء التنفيذ عند استخدام النوع Variant من خلال فحص الانواع البيانية للتعبير قبل القيام بالعمليات الحسابية، الجدول التالي يوضح دوال فحص البيانات:

الدالة	الوصف
IsArray(varname)	تعيد True اذا كان المتغير مصفوفة
IsDate(Expression)	تعيد True اذا كان من الممكن تحويل التعبير الى تاريخ وإلا تعيد False. من الممكن ان يكون التعبير تاريخ.
IsError(Expression)	تعيد True اذا كان التعبير exception type
IsDBNull(Expression)	تعيد True اذا كان التعبير لا يحتوي على بيانات

يتكون التركيب من سلسلة من المتغيرات المختلفة بالتسمية والنوع البياني، يتم الاعلان عن التركيب داخل Class النموذج وكما يلي:

Private | Public Structure name

Statement(s)

End Structure

بعد الاعلان عن التركيب ووضع المتغيرات اللازمة داخله يتم تعريف متغير اخر من نوع تركيب ليتم استدعاءه داخل الاجراءات بنفس مكان تعريف التركيب.

المثال التالي يتضمن الاعلان عن تركيب يضم بيانات موظف في شركة معينة.

```
Private Structure cutomers
    Public Fname As String
    Public Lname As String
    Public age As Integer
    Public salary As Double
End Structure
Private employee As cutomers
```

يتيح لنا المتغير employee التعامل مع عناصر التركيب ويمكن من خلاله اعطاء قيم لها او طباعة قيمها وكذلك التعديل عليها، علماً ان البيانات المدخلة تخص شخص واحد فقط كما في الامثلة التالية:

اعطاء قيم لعناصر التركيب

```
employee.fname = "Hamza"
```

```
employee.lname="Jameel"
```

طباعة عنصر معين من عناصر التركيب.

```
MsgBox(employee.fname)
```

من الامثلة السابقة يتضح لنا طريقة التعامل مع كل عنصر في التركيب من خلال ذكر اسم التركيب متبوع باسم العنصر (structurename.elementname).

اذا كان لدينا مجموعة اشخاص ففي هذه الحالة يتم الاعلان عن مصفوفة بحجم معين من نوع تركيب وكل عنصر في هذه المصفوفة يضم البيانات المعلنة في التركيب ك (fname, lname, age, salary) كما يلي:

```
Private Structure cutomers
    Public fname As String
    Public lname As String
    Public age As Integer
    Public salary As Double
End Structure
Private employee(5) As cutomers
```

الشكل التالي يوضح كيفية خزن بيانات التركيب لعناصر المصفوفة في مواقع الذاكرة.

مواقع الذاكرة	100	101	102...	105
عناصر المصفوفة	A(0)	A(1)	A(2)...	A(5)

البيانات التي يحتويها كل عنصر	Fname	Fname	Fname	Fname
	Lnaem	Lnaem	Lnaem	Lnaem
	Age	Age	Age	Age
	salary	salary	salary	salary

ويتم اعطاء القيم لها من خلال ذكر موقع مصفوفة التركيب متبوع باسم عنصر التركيب (Structurename(Index).Elementname) كما يلي:

```
employee(0).fname = "Hamza"
employee(0).lname = "Jameel"
```

كذلك الطباعة ستكون بالشكل التالي:

```
MsgBox(employee(0).fname)
MsgBox(employee(0).lname)
```

وكذلك الحال بالنسبة لتعديل بيانات اي عنصر داخل التركيب يتم بنفس الطرق السابقة.

ملاحظة: الطريقة الاكفأ في التعامل مع مصفوفة التركيب يتم باستخدام حلقة تكرارية For.

في المثال التالي يتم اعطاء قيم لمصفوفة التركيب من خلال حلقة For.

```
For i As Integer = 0 To 5
    employee(i).fname = InputBox("Enter First name " & i, "Input")
    employee(i).lname = InputBox("Enter last name " & i, "Input")
    employee(i).age = InputBox("Enter age " & i, "Input")
    employee(i).salary = InputBox("Enter salary " & i, "Input")
Next
```