

القرية الهندسية
ENGINEERING VILLAGE
ELECTRONICS & MECHANICA
للإلكترونيات والميكانيكا

احترف

الفيجوال بيسك

في 10 أيام



Microsoft®

Visual Studio®.NET

كتاب عملي ، تطبيقي ، مبسط



هذا الكتاب حر ومخصص للتوزيع المجاني في
نسخته الإلكترونية ويمكنك أن تنشره بشكل كامل
أو جزء منه بشرط ذكر المصدر.

جميع الحقوق محفوظة لدى مؤسسة القرية
الهندسية ولا يجوز استخدام هذا الكتاب بأي
صورة تجارية دون موافقة خطية من الناشر. ومن
يخالف ذلك يعرض نفسه للمساءلة القانونية

الطبعة الأولى ٢٠١٥

www.ev-center.com

فريق العمل

الإشراف العام
فهد السيابي

إعداد
شيخة المنذري

التدقيق و المراجعة
وائل المغيري

التصميم والإخراج الفني
شهرة المنذري

إهداء

أعزاءنا مبرمجي المستقبل في كل مكان،،

تحية طيبة

يسرنا أن نهدىكم هذا الكتاب ليكون شعلة لدركم وخطوتكم الأولى في عالم برمجة الفيچوال بيسك.

تم إطلاق هذا الكتاب بشكل حر ومجاني تحت رخصة المشاع الإبداعي ضمن مبادرة **#القرية_الهندسية** لإثراء المحتوى العربي بالكتب التطبيقية المبسطة في مجال البرمجة. نسأل الله أن يجعل هذا العمل فاتحة الطريق لتطوير مهارات الكثير من المبرمجين العرب المبدعين.

ولنعلم بجد للتحويل من مجتمع مستهلك للتكنولوجيا إلى مجتمع منتج لها.

تحياتنا الحارة
فريق العمل

المقدمة

تعتبر لغة **الفيجوال بيسك** من إحدى وأقوى اللغات في عالم البرمجيات التي تعمل تحت بيئة مايكروسوفت ويندوز والتي تساهم في إنشاء وتطوير التطبيقات والبرامج الصغيرة والمتوسطة والكبيرة التي تخدم سوق العمل من أجل تسريع العديد من العمليات التجارية. و**الفيجوال بيسك** لغة يسهل التعامل معها بعكس لغات البرمجة الأخرى لذلك يمكن أن يتعامل معها أي شخص من الفئات المختلفة بسهولة

للأسباب التالية :

لللمبتدئين

لأنها تعرفهم بيئة التطوير المتكاملة الخاصة بها، وتفسر لهم كل ما يحتاجونه للبرمجة الفعلية من أوامر برمجية ودوال وتوابع مع شرح النماذج والأدوات المستخدمة فيها، وبالتالي اكتساب الخبرة في إنشاء البرامج والتطبيقات الصغيرة والمتوسطة مثل برمجة الآلة الحاسبة.

للمتوسطين

لأنها تحقق لهم قفزة نوعية إلى مواضيع متقدمة في البرمجة بالاتصال مع نظام قواعد البيانات **DataBases**، وإنشاء تطبيقات متقدمة مثل برامج المحاسبة وبرامج إدارة الموارد البشرية وغيرها.

للمتقدمين

الذين يصلون إلى مرحلة الإحترافية في **الفيجوال بيسك** وبرمجة التطبيقات والبرامج المعقدة بالإضافة إلى إنشاء مواقع ويب.

الفهرس

7

المقدمة

13

الفصل الأول

الدرس الأول

14

أنواع لغات البرمجة (Programming Languages) 1

15

مراحل تطور لغة الفيجوال بيسك (Visual Basic Language) 2

17

مصطلحات برمجية أساسية (Terms) 3

19

أنواع المتغيرات (Types of Variables) 4

20

تعريف المتغيرات والثوابت (Variable and Constants) 5

الدرس الثاني

21

كيفية تثبيت برنامج الفيجوال بيسك 1

24

واجهة تشغيل برنامج الفيجوال بيسك 2

26

القوائم الرئيسية في بيئة تطوير الفيجوال بيسك 3

31

الفصل الثاني

الدرس الأول

أدوات التحكم (Tool Box) 1

خصائص أدوات التحكم (Properties of Tool Box) 2

واجهة المستخدم الرسومية (Form) 3

الدرس الثاني

تطبيقات على كيفية كتابة الأوامر البرمجية (Codes) 1
لأدوات التحكم وخصائصها

51

الفصل الثالث

الدرس الأول

الدالة الشرطية (Conditional Function) 1

الدالة (Select Case) 2

الدرس الثاني

الدالة التكرارية (Repetition or Looping) 1

32

34

36

43

9

52

61

64

75

الفصل الرابع

الدرس الأول

الدالة (Val Function) 1

الدالة (Mod Function) 2

الدرس الثاني

المصفوفات (Array) 1

89

الفصل الخامس

الدرس الأول

تطبيقات على أدوات التحكم والدوال في الفيچوال بيسك 1

الدرس الثاني

تطبيقات على أدوات تحكم مختلفة في الفيچوال بيسك 1

111

الملحقات

126

الخاتمة

127

المصادر

76

77

78

90

97

التمهيد

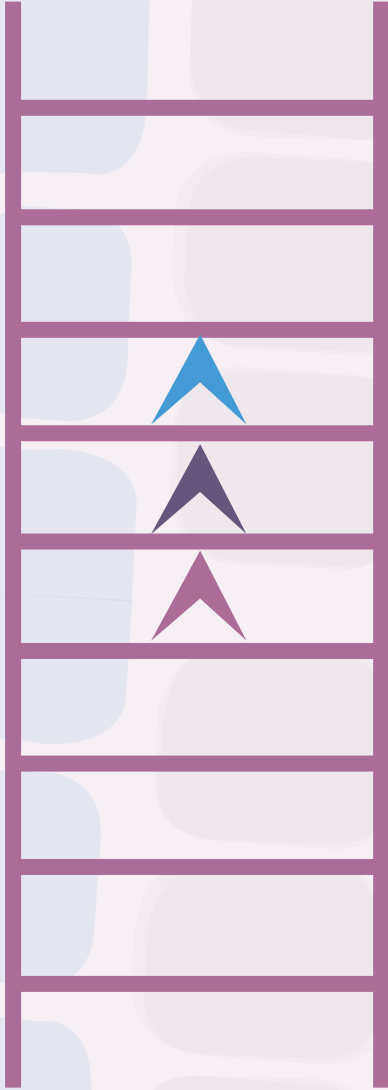
مرحبا بك في **عالم الفيجوال بيسك** . سأأخذك هذا الكتاب في رحلة للتعرف على أساسيات البرمجة بلغة الفيجوال بيسك . خطوة بخطوة من لحظة تثبيت البرنامج وتنصيبه على جهازك لأول مرة وصولا إلى تنفيذ مجموعة من التطبيقات المفيدة بيدك مع مجموعة من التحديات المذهلة . في نهاية الكتاب لصقل مهارتك والتأكد من إكتسابك للمهارات الأساسية . في هذا الكتاب ستتعلم بالممارسة، فهو ليس للقراءة والمطالعة .

أحضر جهاز الحاسوب الخاص بك استعدادا للرحلة.

في البداية سنسترجع معا : **ما هي البرمجة ؟** و **ما هي الفيجوال بيسك ؟**
و **ما هي مراحل تطورها ؟**

إنطلق !

لنتسابق نحو القمة



الفصل الأول

- 1 كيفية تثبيت برنامج الفيجوال بيسك
- 2 واجهة تشغيل برنامج الفيجوال بيسك
- 3 القوائم الرئيسية في بيئة تطوير الفيجوال بيسك

الدرس الثاني

- 1 مراحل تطور لغة الفيجوال بيسك (Visual Basic Language)
- 2 مصطلحات برمجية أساسية
- 3 أنواع المتغيرات (Types of Variables)
- 4 تعريف المتغيرات والثوابت (Variable and Constants)
- 5

الدرس الأول

أنواع لغات البرمجة Programming Languages

1. لغة الآلة (Machine Language) :

هي اللغة التي يفهمها الحاسب الآلي مباشرة و **تتمثل**: برموز قد تكون أصفار أو وحدات مثل **Binary language (0,1)** التي تشكل أوامر برمجية يفهمها الحاسوب وتعتبر من أصعب لغات البرمجة وتستغرق الكثير من الوقت عند برمجتها.

2. لغة المجمع (Assembly or Language) :

هي عبارة عن مترجمات للبرامج حيث يكتب البرنامج بلغة برمجة قريبة من الانجليزية وتقوم هذه المجمات بترجمتها إلى لغة الآلة. وتعتبر هذه اللغة أبسط و أوضح بكثير من لغة الآلة، لكنها بقيت صعبة قليلا مثل لغة الرموز **assembly Language** .

3. لغات عالية المستوى (High Level Language) :

هي لغة سهلة جدا مقارنة باللغات الأخرى ولغة قريبة من لغة البشر لذلك سميت لغة عالية المستوى وهي من أقوى لغات البرمجة ولا تحتاج إلى كتابة كودات أو أوامر برمجية كثيرة مثل بقية اللغات وعندما يتم برمجتها يجب تحويلها إلى لغة الآلة لكي يفهمها الحاسوب بواسطة مترجم أو محول يسمى **converter** حيث يحول لغات البرمجة عالية المستوى إلى لغة الآلة **lk**، ومن أمثلة اللغات العالية المستوى لغة **C++** و لغة الفيجوال بيسك **Visual Basic** .

مراحل تطور لغة الفيچوال بيسك (Visual Basic)

أصل لغة فيچوال بيسك هي لغة بيسك التي ظهرت في
كلية دارتماوث (Dartmouth)



جون كيميني
John Kemeny



توماس كيرتز
Thomas Kurtz

ظهور لغة الفيچوال
بيسك على يد

1963 م

أصبحت لغة الفيچوال بيسك بسرعة فائقة من
أشهر و أسهل لغات البرمجة بل لم تقف عند هذا
فحسب فأصبحت كلغة تعليمية تدرّس في المعاهد
و الكليات والجامعات و بدأ انتشارها

أخذت هذه اللغة في التطور الدائم مع المحافظة
على سهولة استخدامها و بساطتها و كان الإصدار
الأول لهذه اللغة

ظهور الإصدار الثاني من لغة الفيچوال بيسك

ظهور الإصدار الثالث من لغة الفيچوال بيسك

السبعينات

1991 م

1992 م

1993 م

من ذلك الوقت تعاقبت إصدارات كثيرة منها.

وبعد العديد من التطويرات التي تمت على لغة
الفيجوال بيسك

2005 م

تم إصدار الفيجوال بيسك المرئي والذي يشمل على
الفيجوال بيسك دوت نت [Visual Basic.Net](#) والسي
شارب # C والسي بلس بلس ++C و [ASP.Net](#) والعديد
من لغات التطوير

2008 م

تم تطوير الفيجوال البيسك المرئي أكثر فأكثر في هذا
الاصدار

2010 م

وصول الفيجوال البيسك المرئي إلى بيئة التطوير
مايكروسوفت فيجوال إستيديو إكسبريس [Microsoft
Visual Studio Express](#) .

2014 م

الفيجوال بيسك (Visual Basic)

هي أداة تطويرية ولغة من لغات البرمجة عالية المستوى وتعتبر واجهة المستخدم الرسومية التي تستخدم لإنتاج التطبيقات والبرامج المرئية وتطبيقات ويب. وتحتوي على العديد من الإعدادات والأوامر البرمجية الجاهزة التي تسهل علينا كتابة الشفرة البرمجية أو الأمر البرمجي وتصميم البرامج التي تعمل تحت نظام الويندوز وكذلك البرامج التي تعمل على مواقع ويب. وتساعد هذه اللغة على زيادة الإنتاجية عند تصميم البرامج خاصة البرامج المتعلقة بقواعد البيانات وبرامج الانترنت. والملاحظ أنه عندما تعتاد على استخدام بيئة التطوير الخاصة بالفيجوال بيسك فانك ستصبح قادرا على استخدام نفس الأدوات مع السي شارب وكذلك السي بلس بلس وغيرها من أدوات التطوير المتضمنة في [Microsoft Visual Studio](#) .

مصطلحات برمجية أساسية

1 الكود أو الشفرة البرمجية (Code) :

هو عبارة عن أوامر برمجية تعطى للكمبيوتر على شكل نص تكتب بواسطة لغة من لغات البرمجة كلغة الفيجوال بيسك والتي يفسرها مترجم يسمى (Compiler) ليفهمها الكمبيوتر ولكل فورم أو وحدة نمطية أو أداة تحكم كود خاص به في الفيجوال بيسك. وقد يكون الكود مقسما إلى كائنات Objects، ودالات أو توابع Functions، وإجراءات Procedures، و أحداث Events.

2 الكائن (Object) :

قد يكون تعريف لكود أداة تحكم مثل الزر Button ويكون له عدة إجراءات.

3 الإجراء (Procedure) :

هو عملية أو عمليات ينفذها البرنامج عند تحقق حدث معين كالنقر على أداة الزر Button في البرنامج.

4 الحدث (Event) :

هو عبارة عن فعل يقوم به المستخدم للبرنامج المصمم كالنقر على أداة الزر Button أو تغيير نص في مربع نص Text Box أو حتى تحريك الماوس أو الفأرة ويرافقه تنفيذ الإجراء المتعلق به.

الثوابت (Constants) :

5

هي عبارة عن مواقع في الذاكرة تخزن فيها بيانات بشكل مؤقت ولكن لا يمكن تغيير قيمتها أثناء تنفيذ البرنامج. فمثلا يمكن للبرنامج أن يخزن في الثابت Con1 القيمة 1 لكنه لا يستطيع تغييرها وإن حاول تغييرها فإن المترجم يرسل رسالة خاطئة أثناء تنفيذ البرنامج.

المتغيرات (Variables) :

6

المتغيرات هي عبارة عن مواقع في الذاكرة تخزن البيانات بشكل مؤقت ويمكن تغيير قيمتها أثناء تنفيذ البرنامج. فمثلا يمكن للبرنامج أن يخزن في المتغير Var1 القيمة 1 ثم يخزن القيمة 2 والقيمة التي سترجع من المتغير هي آخر قيمة تم تخزينها.



تابع فانت المبرمج الناجح

أنواع المتغيرات (Types of Variables)

حجم الذاكرة	يمثل	نوع المتغير
byte 1	يمثل حروف	String
byte 2	يمثل أعداد صحيحة	Integer
byte 8	يمثل أرقام ذات فواصل أو كسور	Double
byte 1	يمثل قيمتين: صحيحة (True)، خاطئة (False)	Boolean
byte 4	يمثل عدد صحيح طويل	Long
byte 1	يمثل عدد صحيح	Short
byte 1	يمثل حرف واحد فقط	Char
byte 1	يمثل قيم موجبة فقط ما بين (0 إلى 255)	Byte
byte 8	يمثل التاريخ والوقت	Date
Single 4	يمثل عدد غير صحيح بخانة واحدة بعد الفاصلة	byte
-	يمثل الأرقام العشرية	Decimal

الشروط التي يجب إتباعها عند تحديد اسم المتغير:

- 1 أن يبدأ بحرف هجائي (انجليزي) مثل: (a , b , ... or A, B, ...)
- 2 أن لا يحتوي على علامات خاصة مثل: (! , ? , : , ...)
- 3 أن لا يكون هذا الاسم من الكلمات المحفوظة في مكتبة الفيچوال بيسك مثل End , Byte, If, Dim وغيرها .
- 4 يمكن للاسم أن يتضمن أرقام و حروف.

تعريف الثوابت والمتغيرات عند كتابة الكود في الفيجوال بيسك:

قاعدة الثابت:

كلمة التعريف هي: (Const) 1
Const CN as DataType = Data (Value)
Const + (اسم الثابت) as (اسم الثابت) = (قيمة الثابت)
الاسم الثابت أو القيمة الثابتة : CIV
نوع البيانات : Data Type
مثال:
Const C1 as integer = 1

تعريف المتغيرات عند كتابة الكود:

2
المتغير يكون اسم غير معرف به في قاموس الفيجوال بيسك.
كلمة التعريف هي: (Dim)
Const + (اسم المتغير) as (اسم المتغير) = (قيمة المتغير)

القاعدة:

Dim Var1 as DataType = Data
اسم المتغير : Var1
مثال:
Dim Student as string = "Sheikha"
إذا كان المتغير من نوع "string" يكون بين علامتي اقتباس

كيف يمكن تخزين البيانات في المتغير؟

القاعدة:

Var1 = Data "المتغير 1 يساوي البيانات المخزنة"
مثال:
S1 = textbox1.Text "المتغير S1 يخزن البيانات في أداة التحكم التي تدعى مربع نص TextBox".

أمثلة أخرى على تعريف أنواع المتغيرات:

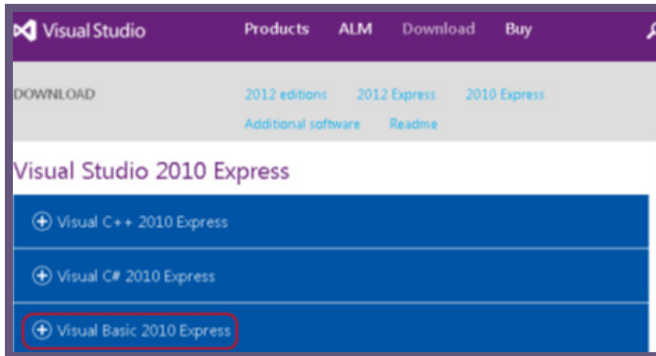
Dim age As Integer = 25
Dim abc As Boolean = True
Dim Pi As Double = 3.14

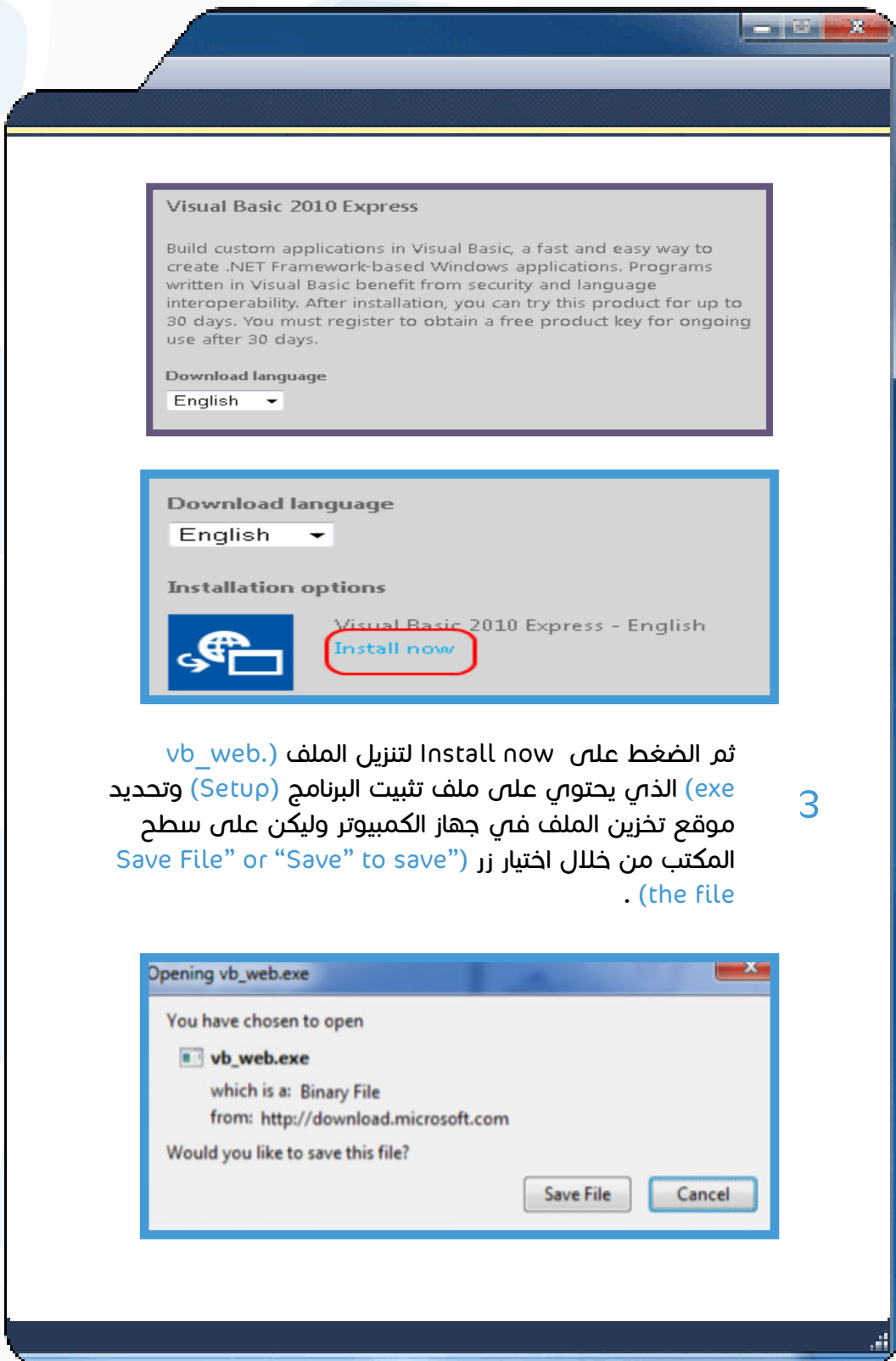
كيفية تثبيت برنامج الفيجوال بيسك على الحاسب الآلي

يتوافق برنامج الفيجوال بيسك ويعمل تحت نظام التشغيل Windows مثل: Windows 8، Windows 7، Windows XP، Windows Vista. وبالتالي لتثبيت برنامج الفيجوال بيسك على جهاز الكمبيوتر لا بد من احتواءه على نظام التشغيل (Windows).

الخطوات المتبعة لتنصيب (تثبيت) البرنامج:

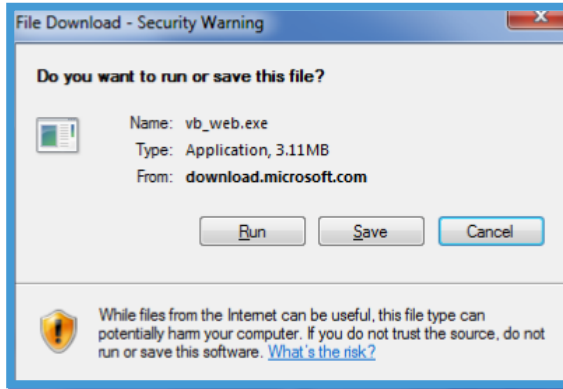
- 1 من خلال البحث في الانترنت يتم الدخول إلى الموقع الرسمي لتحميل برنامج الفيجوال بيسك 2010 من خلال الرابط التالي:
<http://www.microsoft.com/visualstudio/eng/downloads#d-2010-express>
- 2 تظهر مجموعة من خيارات التنزيل المتعلقة بالبرنامج في الموقع، ويتم اختيار برنامج الفيجوال بيسك Visual Basic Express 2010 ، ثم يتم اختيار إحدى لغات التنزيل مثل: English.





ثم الضغط على Install now لتنزيل الملف (vb_web.exe) الذي يحتوي على ملف تثبيت البرنامج (Setup) وتحديد موقع تخزين الملف في جهاز الكمبيوتر وليكن على سطح المكتب من خلال اختيار زر ("Save" or "Save File" to save) (the file).

3



4 ثم الضغط على زر Run لمتابعة وإكمال خطوات تثبيت البرنامج .

5 ثم يتم فتح الملف لتثبيت برنامج الفيجوال بيسك من خلال الضغط على ملف (Setup) لفتح قائمة التثبيت واختيار الزر Next .

6 ثم يتم الموافقة على (License Terms) المتعلقة ببرنامج الفيجوال بيسك من خلال النقر على الخيار: I have read and accept the license terms ، ثم النقر على زر Next.

7 بعد ذلك يتم اختيار الملف الذي يحتوي على بعض خصائص ومواصفات برنامج الفيجوال بيسك أو اختيار الملف الذي يحتوي على كل الخصائص والمواصفات والمرفقات الأخرى للبرنامج من خلال الضغط على الخيار: Microsoft Silver Light .

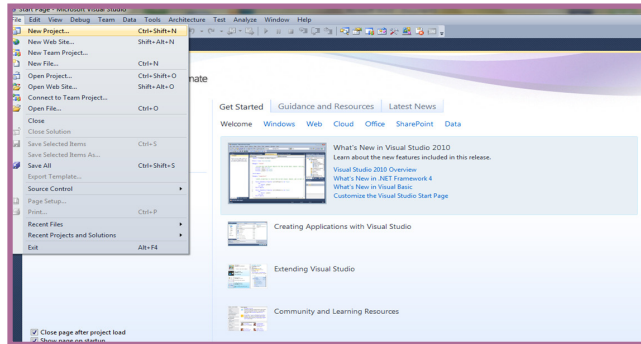
8 ثم الضغط على زر Next، واختيار موقع حفظ ملف تثبيت البرنامج على جهاز الكمبيوتر مثل الملف C ثم النقر على زر Install.

9 الانتظار إلى اكتمال تثبيت جميع الملفات والخيارات المرفقة مع البرنامج ثم النقر على الزر Exit بعد اكتمال التثبيت (Setup Complete) وبالتالي يصبح برنامج الفيجوال بيسك مثبتا على جهاز الكمبيوتر ويتم فتحه مباشرة من القائمة (Start).

واجهة تشغيل برنامج الفيجوال بيسك

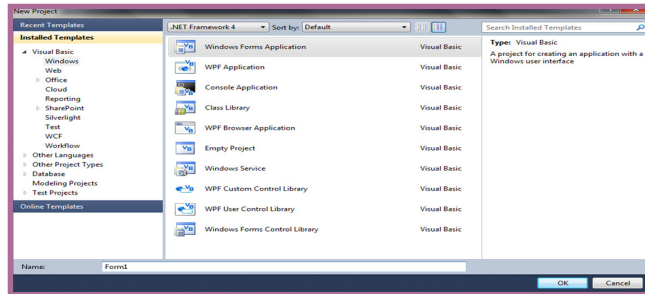
افتح مشروع جديد:

يتم اختيار الخانة **File** من شريط المهام الموجود في أعلى البرنامج والنقر على أول خيار **New Project** وقد يكون هذا الخيار موجود مباشرة عند فتح البرنامج في الجانب الأيسر .



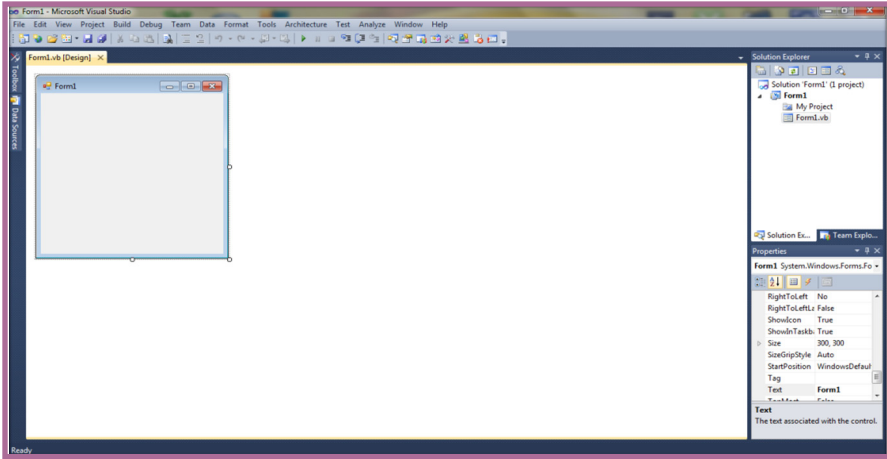
لاختيار نموذج التصميم:

يتم اختيار النموذج **Windows application >> Windows form application** المختص بتصميم البرامج النضية أو الرسومية.



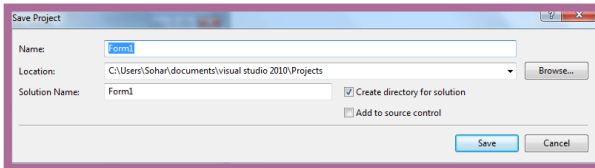
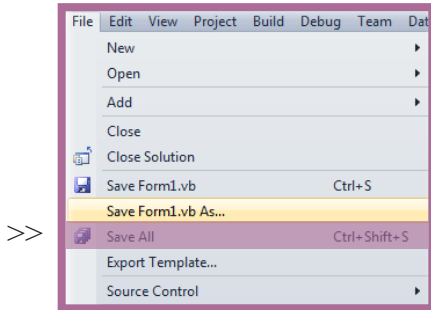
عند اختيار تطبيق Windows Form Application

تفتح صفحة تصميم البرنامج وتتضمن واجهة المستخدم الرسومية (Form) أو الشاشة الرئيسية مع باقي مكونات بيئة تطوير الفيجوال بيسك .



عند الانتهاء من تصميم أي برنامج أو تطبيق

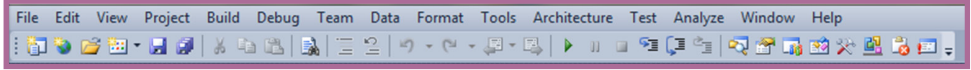
يتم حفظه عن طريق الضغط على
File >> Save All ثم الضغط على Save



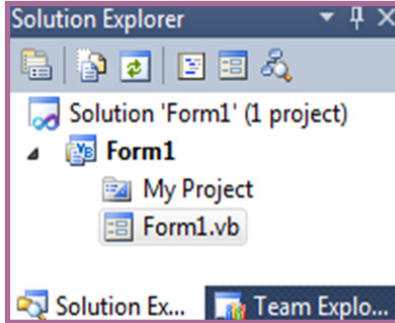
القوائم الرئيسية في بيئة تطوير برنامج الفيجوال بيسك

شريط المهام:

يستخدم هذا الشريط للقيام بجميع خصائص البرنامج الموضحة به ويكون في أعلى صفحة البرنامج.



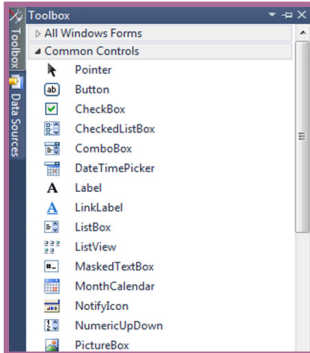
:Solution Explorer



توضح هذه القائمة العناصر و الكائنات المستخدمة في المشروع الخاص بك مثل عدد الفورم التي صممت في المشروع وتوجد هذه القائمة في الجانب الأيمن من صفحة البرنامج .

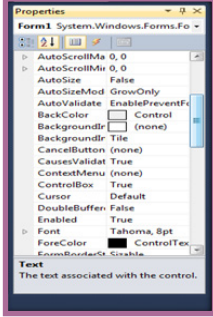
26

شريط أدوات التحكم (ToolBox) :



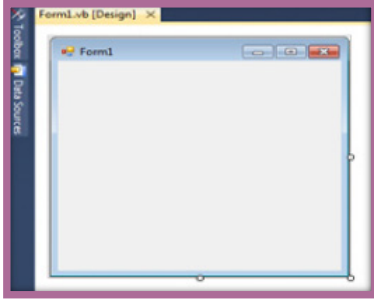
هو الذي يتضمن جميع أدوات التحكم المختلفة التي يتم استخدامها وإدراجها في واجهة المستخدم الرسومية (Form) من أجل تصميم تطبيق معين وتنفيذ الأوامر البرمجية عليها.

نافذة الخصائص (Properties):



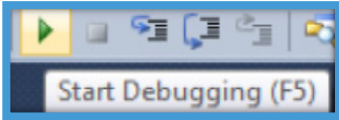
هي قائمة عرض خصائص ومميزات أدوات التحكم والتي تساعد على ضبط خصائص أي أداة تحكم أثناء تصميم البرنامج دون الحاجة إلى كتابة كود أو أمر برمجي. ولعرض قائمة الخصائص يتم النقر على قائمة view ثم على **Properties Window** وقد تكون في الجانب الأيمن في الأسفل من البرنامج.

شاشة التصميم أو واجهة المستخدم الرسومية (Form):



هي النافذة أو الواجهة الرئيسية في البرنامج التي يتم تصميم وتنفيذ البرنامج أو التطبيق من خلالها .

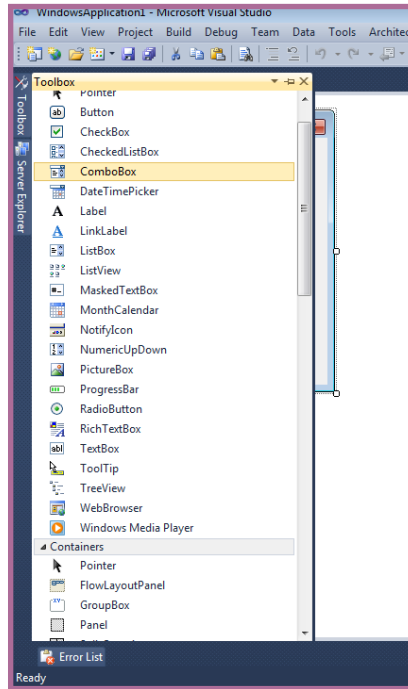
زر (Debug) الموجود في شريط المهام:

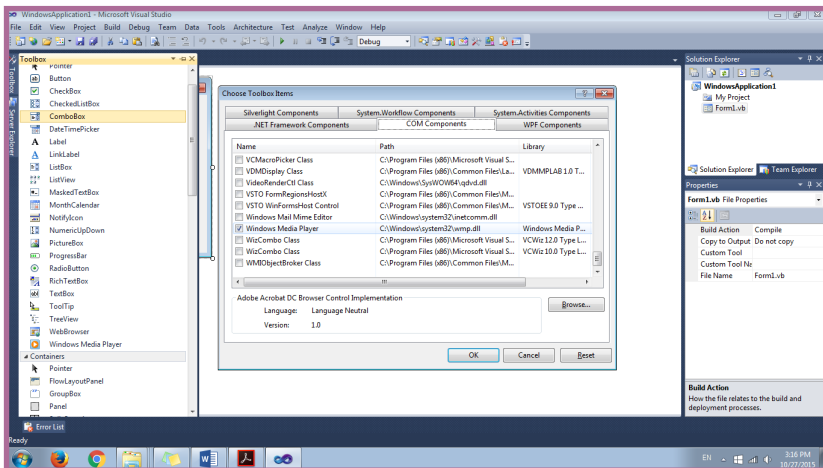
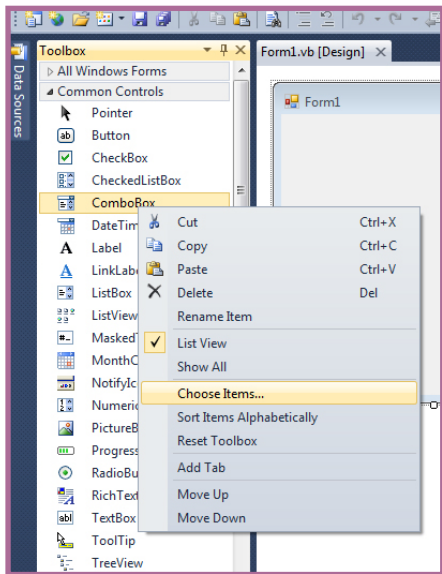


يستخدم لمعاينة كيفية سير عمل البرنامج بعد تصميمه وكتابة الأوامر البرمجية فيه ومراجعة ما إذا كانت فيه أخطاء برمجية أم لا ويمكننا الضغط على F5 كوصلة سريعة لـ **Debug** .

ملاحظة:

عند إختفاء إحدى قوائم البرنامج يتم إظهارها من خلال:
النقر على قائمة (view) ثم اختبار القوائم منها وكذلك يمكن إدراج قوائم
أخرى عن طريق نفس القائمة (view) ثم النقر على (other windows)







إذا لم تحاول أن تفعل شيئاً
أبعد مما قد أتقنته .. فأنت لا
تتقدم أبداً.

من أقوال الفيلسوف
(رونالد اسبورت)

الفصل الثاني

تطبيقات على كيفية كتابة الأوامر البرمجية
(Codes) لأدوات التحكم وخصائصها

1

الدرس الثاني

واجهة المستخدم الرسومية
(Form)

3

خصائص أدوات التحكم
(Properties of ToolBox)

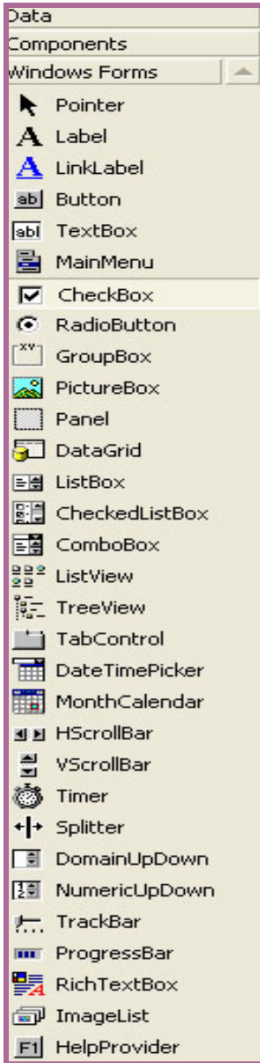
2

أدوات التحكم المستخدمة في
برنامج الفيجوال بيسك (ToolBox)

1

الدرس الأول

أدوات التحكم في برنامج الفيجوال بيسك (Tool Box)



تتكون قائمة الأدوات (Tool Box) في برنامج الفيجوال بيسك من 47 أداة تحكم ومنها:

1 الشاشة المرئية أو النافذة (Form):

عبارة عن إطار أو نافذة أو صفحة تشكل واجهة أي برنامج وتحتوي بداخلها على أدوات التحكم الأخرى مثل الزر، مربع النص، صندوق الاختيار وغيرها.

2 أداة الزر (Button):

هي من أهم أدوات التحكم الأساسية في أي برنامج وعند النقر عليه، يقوم بتنفيذ إجراء أو عملية أو حدث معين.

3 مربع النص (Text Box):

هو أداة مخصصة لكتابة وإدخال أو إخراج النصوص في البرنامج ويستخدمه المستخدم لكتابة الحروف أو الكلمات والجمل أو الأرقام. ويمكن إعطائه أمر برمجي لإظهار أي نص أو رقم.

6 أداة التسمية (Label):

تستخدم لإظهار وكتابة النصوص على النموذج لإظهار نتيجة.

7 خانة الاختيار (RadioButton): هو عنصر تحكم يسمح للمستخدم باختيار خيار واحد فقط من مجموعة معرفة مسبقا من الخيارات وقد تكون له قيمتان True أو False.

8 صندوق أو خانة الاختيار (Check Box): هو عنصر تحكم يسمح للمستخدم باختيار أكثر من خيار في نفس الوقت من مجموعة معرفة مسبقا من الخيارات والتي قد تكون لها قيمتان True أو False .

9 قائمتي السرد أو التحرير (Combo Box & List Box): وتستخدمان لإنشاء قوائم تحتوي على قيم أو معلومات ثم يتم الاختيار منهما.

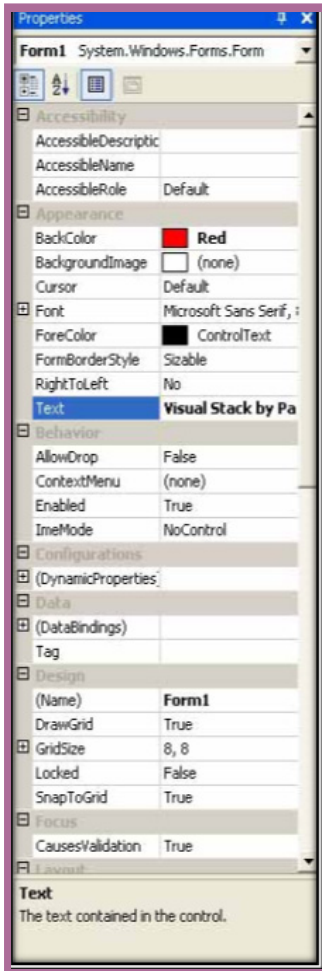
10 صندوق الصورة (Picture Box): يستخدم لإظهار صورة والتحكم في حجمها أو إظهار حجمها الأصلي.

11 الصورة (Image): تستخدم لإظهار صورة ولها خاصيتين: تحدد ما إذا كان يجب تكبير أو تصغير الصورة / أو تستخدم لتحديد الصورة.

12 الإطار أو صندوق المجموعة (Group Box): هو الذي يحيط أو يتضمن عناصر أخرى من أدوات التحكم وقد يفصل بين مجموعة وأخرى من نوعية البيانات الموجودة في الواجهة أو الشاشة.

13 المؤقت (Timer): هو الذي يقوم بتنفيذ الإجراء أو الحدث المتعلق به كل مدة محددة تحدد بثنائية وهي أداة تستخدم للوقت والتاريخ أيضا.

خصائص أدوات التحكم (Properties)



لكل أداة تحكم في برنامج الفيجوال بيسك خصائص معينة مثل تغيير اللون والحجم والخط والخلفية وغيرها من الخصائص التي يمكن التحكم فيها من خلال قائمة الخصائص أو من خلال كتابتها كأوامر برمجية في صفحات الكودات ومن هذه الخصائص (Properties) هي:

الاسم (Name) : هي خاصية أساسية لأي أداة تحكم ويعتمد عليها في تحديد إجراءات متعلقة بها أثناء كتابة الأوامر البرمجية.

النص (Text) : تحدد أو تضبط تسمية عنصر التحكم التي ستظهر في فورم أو شاشة البرنامج.

الخط (Font) : خاصية تستخدم لتغيير نوع خط أي أداة تحكم تستخدم في البرنامج.

لون الخلفية (Back Color) : تحدد أو توضح لون خلفية أداة التحكم.

صورة الخلفية (Back ground Image) : تستخدم هذه الخاصية لتغيير الخلفية كصورة لأي أداة تحكم في البرنامج.

Back ground Image Layout : تستخدم لتغيير حجم الخلفية من خلال الخيارات الموجودة ومنها **Title, Center, Zoom** وغيرها.

رمز (Icon) : هذه الخاصية تستخدم كرمز لأي شاشة أو واجهة أو برنامج أو أيقونة تشير إلى أمر ما.

Enabled : خاصية تستخدم فيما إذا كانت أداة التحكم قابلة للاستخدام أو لا من قبل المستخدم. على سبيل **المثال**: تلاحظ في إحدى البرامج مثلا على أداة زر معين لون رمادي مهما ضغطت عليه لا يمكنك استخدامه وذلك بسبب وجود هذه الخاصية التي تحدد قيمتان: عندما تكون الأداة مفعلة أو قابلة للاستخدام تأخذ القيمة **True** وعندما تكون الأداة غير مفعلة تأخذ القيمة **False** .

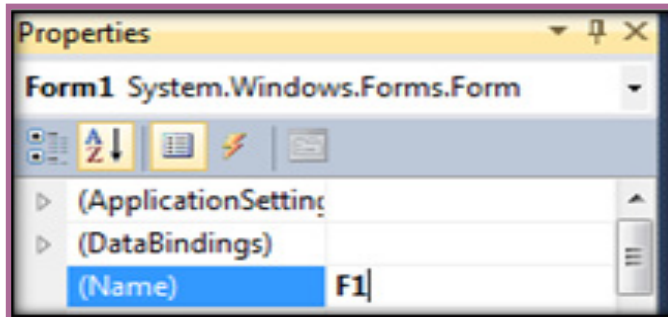
خاصية Visible : تحدد ما إذا كانت أداة التحكم مرئية أم لا. وهذا يعني هل ستظهر الأداة للمستخدم في البرنامج أم لا تظهر وتأخذ هذه الخاصية قيمتان إما مرئية **True** أو غير مرئية **False** .

ما هي واجهة المستخدم الرسومية (Form) ؟

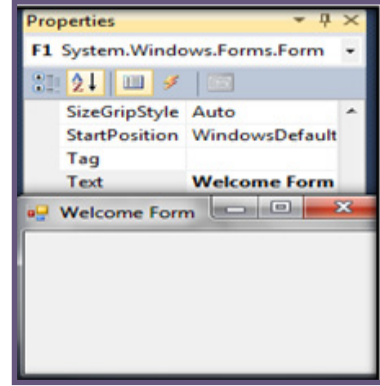
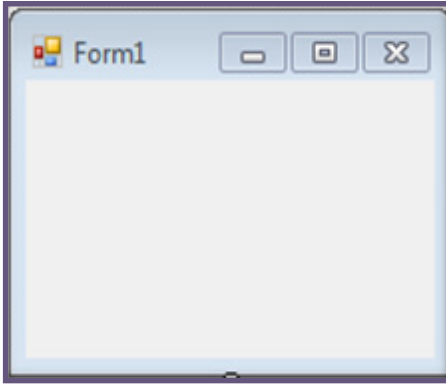
هي شاشة التصميم أو الشاشة المرئية التي تعتبر النافذة أو الواجهة الرئيسية في برنامج الفيجوال بيسك والتي يتم تصميم وتنفيذ البرامج أو التطبيقات من خلالها وتحتوي بداخلها على أدوات التحكم الأخرى في البرنامج مثل الزر (Button)، مربع النص (TextBox)، صندوق الاختيار (CheckBox)، أداة التسمية (Label) وغيرها.

من الخصائص التي يمكن أن يتم التحكم فيها وتغييرها في واجهة المستخدم الرسومية:

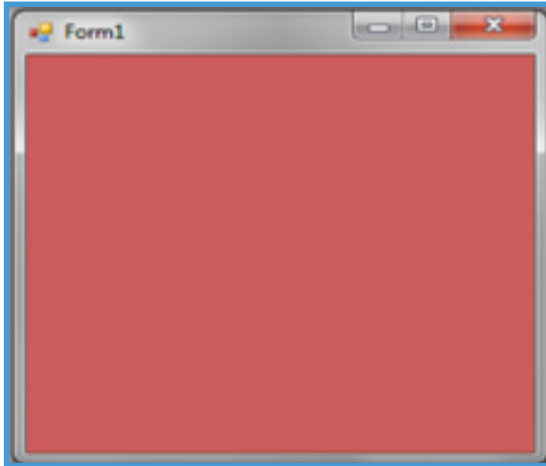
1 الاسم (Name) : هي خاصية أساسية لأي أداة تحكم لتغيير اسمها الأصلي إلى اسم مختصر والذي يعتمد عليه عند مناداة الأداة وتنفيذ أمر برمجي عليها في صفحة الكود من أجل تحديد إجراءات متعلقة بها.



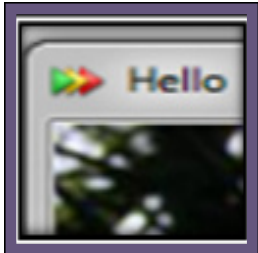
2 النص (Text) : هذه الخاصية تحدد أو تضبط تسمية أي عنصر مثل الاسم الظاهر أعلى شاشة البرنامج (Form).



3 لون الخلفية (Back Color) : تستخدم هذه الخاصية لتغيير لون خلفية شاشة التصميم.



4 صورة الخلفية (Back Ground Image Layout & Back Ground Image) : تغيير خلفية الشاشة (كصورة) بالإضافة إلى تغيير وضعية الصورة مثل (Title, Stretch, Zoom, Center).



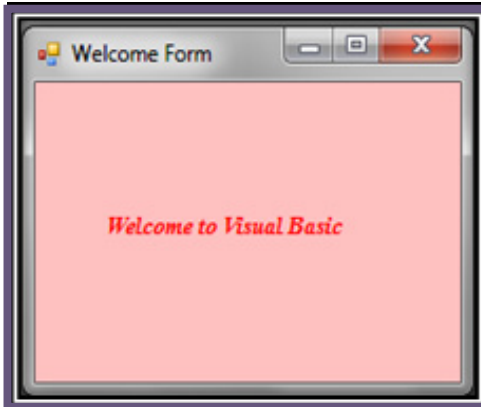
5 الرمز (Icon) : تستخدم هذه الخاصية لتحديد رمز أو شعار معين لشاشة التصميم يحدده المستخدم.

ومن أدوات التحكم التي تتضمنها شاشة التصميم وخصائصها:

أداة التسمية (Label) : هي إحدى أدوات التحكم التي تستخدم لإظهار تسمية عنصر معين أو كلمة أو لإظهار نتيجة. ويتم إدراجها في شاشة التصميم كما هو موضح بالشكل أدناه :



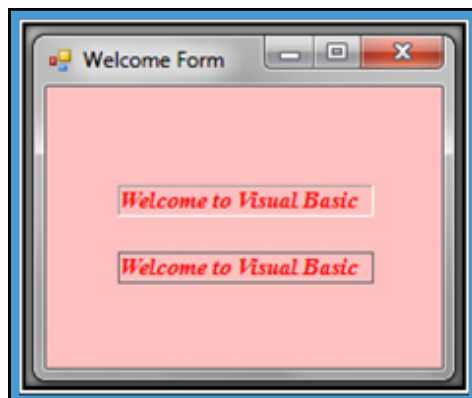
ويمكن تغيير بعض خصائص هذه الأداة من خلال قائمة الخصائص في البرنامج:



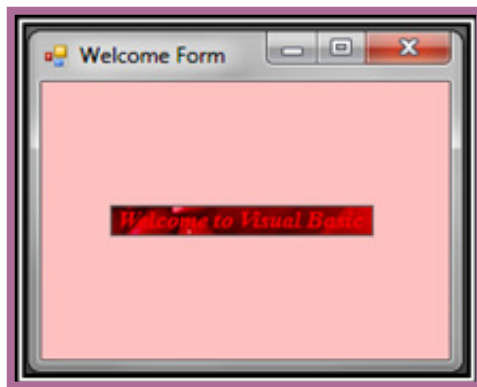
الخط (Font) : تستخدم هذه الخاصية في تغيير حجم وشكل الخط أو النص المكتوب داخل أداة التسمية Label.

لون الخط (Fore Color) : تستخدم لتغيير لون الخط.

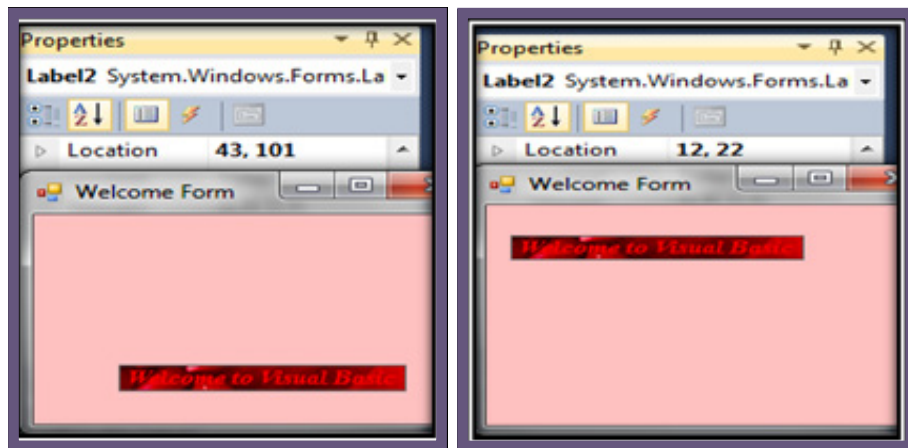
شكل الاطار (Border Style) : تستخدم هذه الخاصية لتغيير شكل الإطار المحيط بأداة التسمية ويمكن أن يكون بعدة أشكال مثل (Fixed 3D, Fixed Single, None).



الصورة ووضعيتها (Image & Image Align) : تستخدم هذه الخاصية من أجل إضافة صورة داخل أداة التسمية بالإضافة إلى تغيير وضعية الصورة داخل هذه الأداة أو تغيير اتجاهها.

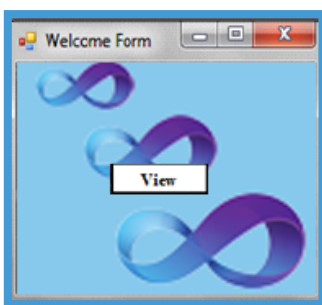
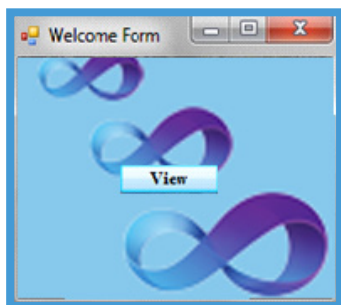


موقع الأداة (Location) : خاصية موقع الأداة التي تحدد بعد الأداة عن يسار أو يمين أو أعلى إطار الفورم أو الشاشة التي تحتويها.



الزر (Button) : هو من أهم أدوات التحكم الأساسية في أي برنامج، وأكثر أداة يعتمد عليها في الانتقال إلى صفحة الكودات لكتابة الأوامر البرمجية فيه وبالتالي عند النقر عليه يقوم بتنفيذ إجراء أو عملية معينة أثناء تشغيل البرنامج أو التطبيق. ويمكن تغيير خصائص أداة الزر من خلال قائمة الخصائص في البرنامج. ومن الأمثلة على هذه الخصائص:

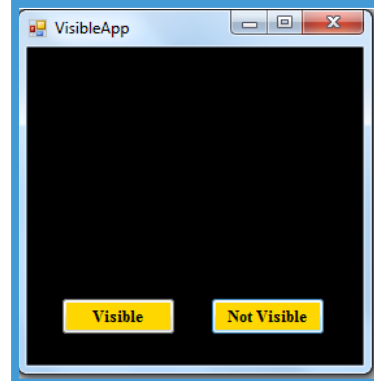
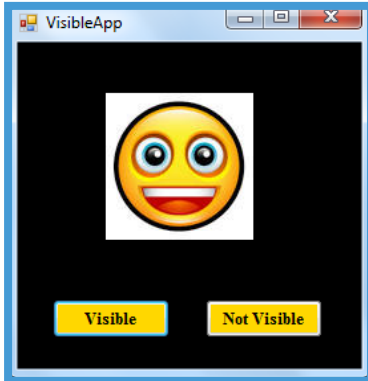
أشكال الحدود (Flat Style) : تستخدم هذه الخاصية لتغيير شكل وإطار أو حدود أداة الزر وتشمل عدة أنواع في قائمة الخصائص مثل (Flat, Popup, Standard, System).



قابلية الاستخدام (Enabled): هي خاصية تستخدم فيما إذا كانت أداة التحكم قابلة للاستخدام أو لا من قبل المستخدم. على سبيل المثال: تلاحظ في إحدى البرامج مثلًا على أداة زر معين لون رمادي مهما نقرت عليه لا يمكنك استخدامه وذلك بسبب وجود هذه الخاصية التي تحدد قيمتان: عندما تكون الأداة مفعلة أو قابلة للاستخدام تأخذ القيمة True وعندما تكون الأداة غير مفعلة تأخذ القيمة False .



الخاصية المرئية (Visible): تحدد هذه الخاصية ما إذا كانت أداة التحكم مرئية أم لا. وهذا يعني هل ستظهر الأداة للمستخدم في البرنامج أم لن تظهر وتأخذ هذه الخاصية قيمتان إما مرئية True أو غير مرئية False . ويمكن استخدام هذه الخاصية عن طريق قائمة الخصائص في البرنامج أو كتابتها كأمر برمجي. مثال ذلك: إضافة صندوق الصورة Picture Box إلى الشاشة مع وضع صورة وتحديد خاصية المرئي Visible وغير المرئي Not Visible للصورة:



كيفية كتابة الأوامر البرمجية (Codes) لأدوات التحكم وخصائصها

الدرس الثاني

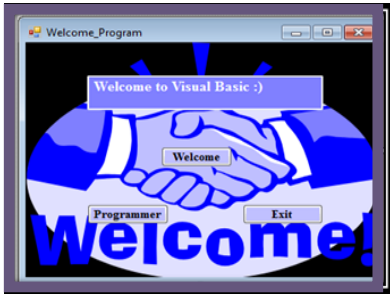
تطبيق (1):

إعطاء أمر برمجي لأداة الزر Button عند النقر عليه بإظهار الجملة التالية في مربع النص: `TextBox 1 . Text = "Hi Visual Basic "`



لكتابه الكود التالي يتم الانتقال إلى صفحة الكود في البرنامج من خلال النقر المزدوج على أداة الزر (Button) :
`Text Box 1.Text = " Hi Visual Basic (: "`

تطبيق (2):



برنامج ترحيب : يستخدم هذا التطبيق كواجهة ترحيب لبعض البرامج الصغيرة حيث إنه عند تنفيذ وتشغيل البرنامج والنقر على زر Welcome يظهر النص الترحيبي داخل مربع النص Text Box بينما يستخدم الزر Programmer للإظهار اسم مبرمج هذا التطبيق و الزر Exit للخروج من البرنامج.

يتم كتابة الكود التالي عند النقر المزدوج على كل من الأزرار، Programmer.
: Welcome. Exit

Programmer Button :

```
Private Sub Button1_Click ( By Val sender As System. Object, By  
Val e As System. EventArgs ) Handles Button 1 .Click  
Text Box 1 . Text = "Done By : Sheikha Al Mandhari "  
End Sub
```

Welcome Button :

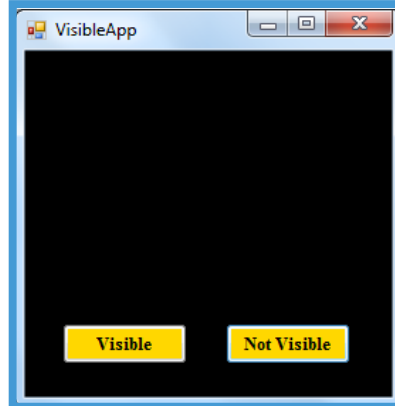
```
Private Sub Button2_Click ( By Val sender As System. Object, By  
Val e As System. EventArgs) Handles Button 2 .Click  
Text Box 1 . Text = " Welcome to Visual Basic (: "  
End Sub
```

Exit Button :

```
Private Sub Button3_Click ( By Val sender As System. Ob-  
ject, By Val e As System .EventArgs) Handles Button 3 .Click  
End sub
```

تطبيق (3):

لتفعيل خاصية **Visible** من خلال كتابتها كأمر برمجي في هذا التطبيق يتم إضافة صندوق الصورة **Picture Box** إلى الشاشة مع وضع صورة وتحديد خاصية المرئي **Visible** وغير المرئي **Not Visible** للصورة :



يتم كتابة الكود التالي عند النقر المزدوج على كل من أدواتي الزر **Visible** و **Not Visible** :

غير مرئي
(Not Visible)



```
Button2_Click "Not Visib"  
PictureBox1.Visible = False
```

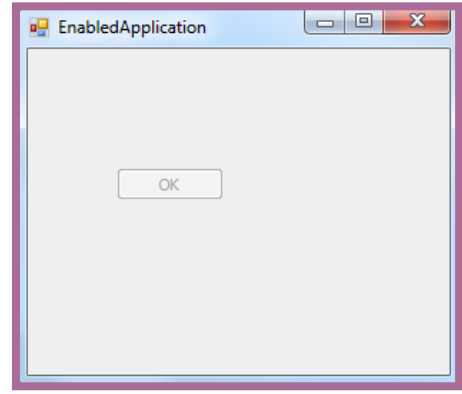
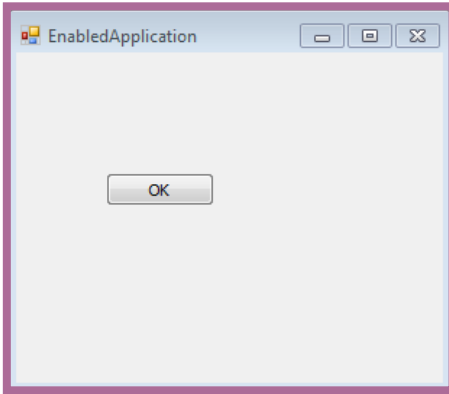
مرئي
(visible)



```
Button1_Click "Visible"  
PictureBox1.Visible = True
```

تطبيق (4):

إضافة زر Button إلى الشاشة وتغيير اسمه إلى Ok وتنفيذ الخاصية Enabled



لتنفيذ خاصية Enabled لأداة الزر OK : يتم كتابة الكود التالي عند النقر
المزدوج عليه :

Button 1_Click

```
Button1.Enabled = False
```

“أداة الزر غير قابلة للاستخدام”

Button 1_Click

```
Button1.Enabled = True
```

“أداة الزر قابلة للاستخدام”

تطبيق (5):

الأوامر البرمجية المتحكم في التنقل بين أدوات التحكم باستخدام مفاتيح لوح المفاتيح (Key Board) في الكمبيوتر وهي: (Up Key, Down Key, Left Key, Right Key)

```
Private Sub _KeyDown (ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles Button1.KeyDown, Button2.KeyDown, TextBox1.KeyDown, MyBase.KeyDown
If e.KeyValue = Keys.Down Then
```

```
    TextBox1.Text = "Down Key pressed"
End If
```

```
If e.KeyValue = Keys.Left Then
```

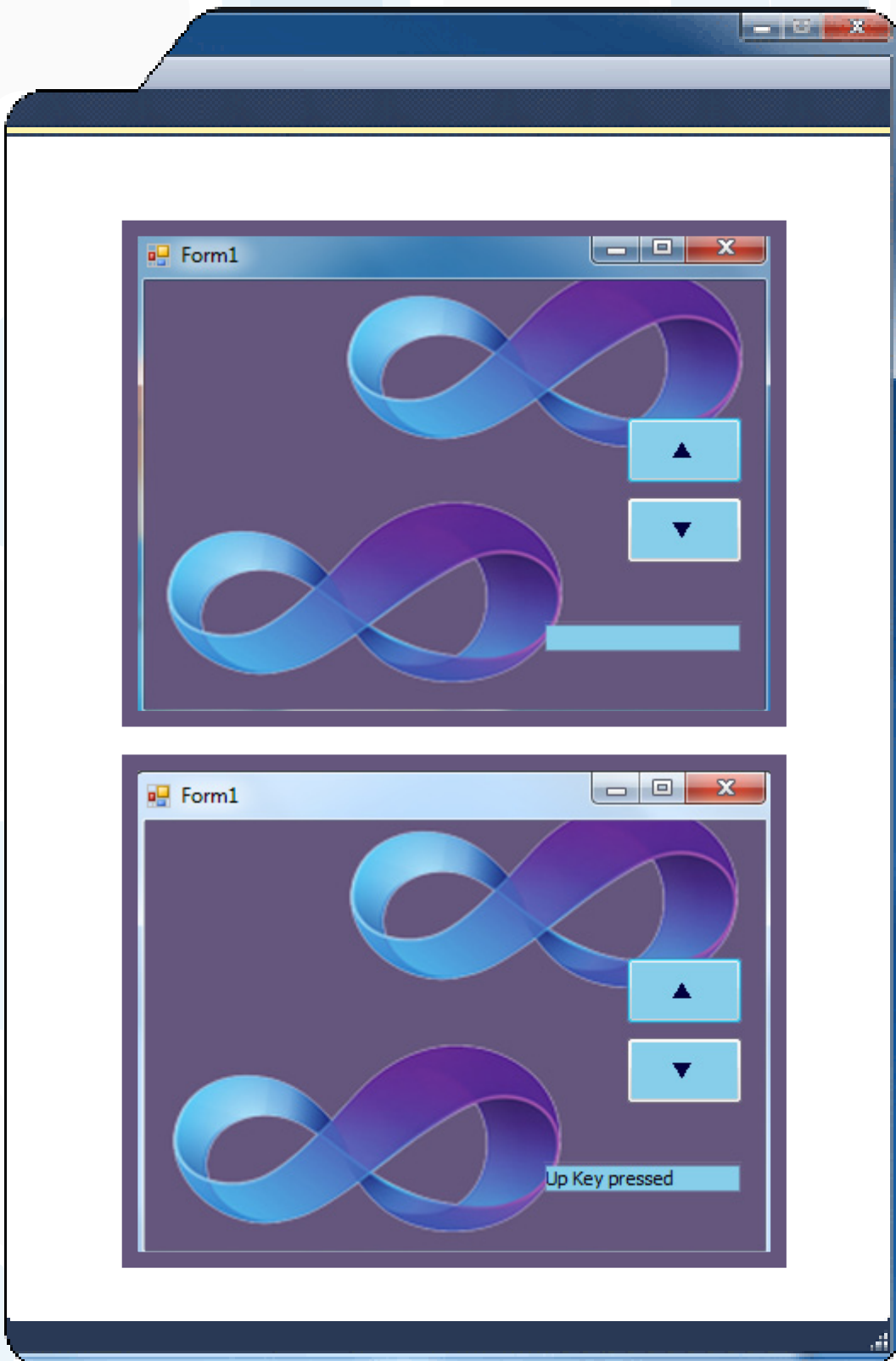
```
    Label1.Text = "Left Key pressed"
End If
End Sub
```

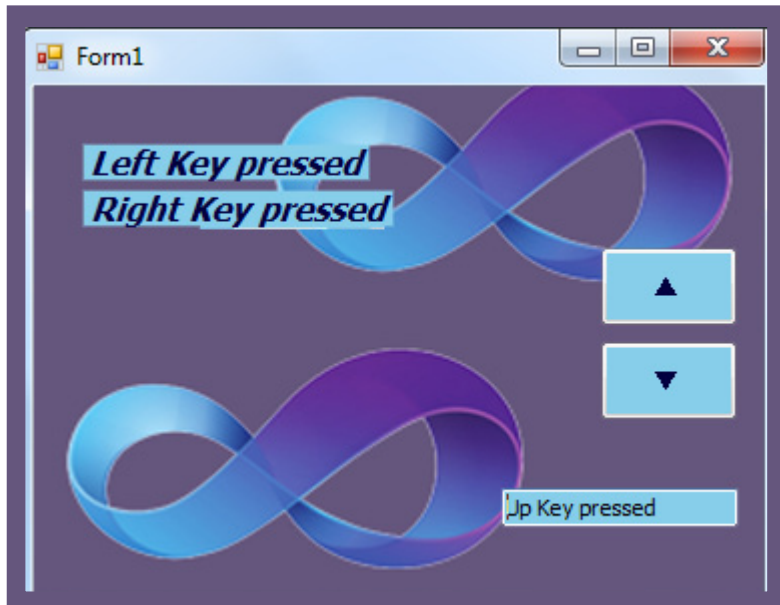
```
Private Sub _KeyUp (ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles Button1.KeyUp, Button2.KeyUp, TextBox1.KeyUp, MyBase.KeyUp
If e.KeyValue = Keys.Up Then
```

```
    TextBox1.Text = "Up Key pressed"
End If
```

```
If e.KeyValue = Keys.Right Then
```

```
    Label2.Text = "Right Key pressed"
End If
End Sub
```







لقد خلقنا لنمارس الاصرار
هكذا نستطيع ان نعرف من
نحن.

توبائس وولف
كاتب أمريكي

الفصل الثالث

الدالة التكرارية
(Repetition or Looping)

1

الدرس الثاني

الدالة
(Select Case)

2

الدالة الشرطية
(Conditional Function)

1

الدرس الأول

الدرس الأول

الدوال أو التوابع (Functions) في برنامج الفيجوال بيسك

الدالة (Function) :

هي عبارة عن أسماء محجوزة (reserved words) ، ومعروفة من قبل برنامج الفيجوال بيسك لتقوم بتنفيذ إجراء معين أو مجموعة من الجمل البرمجية أو القواعد البرمجية التي تقوم بعمل برمجي معين وله معنى عند كتابة الكود الخاص بها.

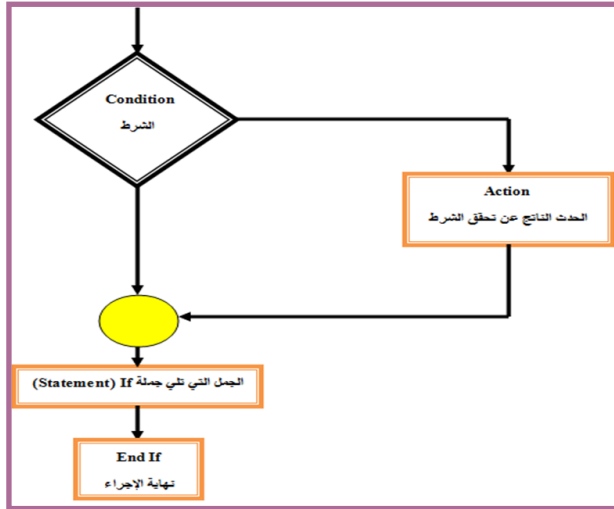
ومن أنواعها :

الدالة الشرطية (Conditional Function) :

هي عبارة عن جملة تحكم **Control Statement** برمجية لا يمكن تنفيذها إلا عند تحقق شرط معين أو على حسب التحقق من صحة الشرط المحدد. وتشمل هذه الدالة عدة أنواع **مثل (If-End If, If-Else-End If, If-Elseif-End If, Select Case)**.

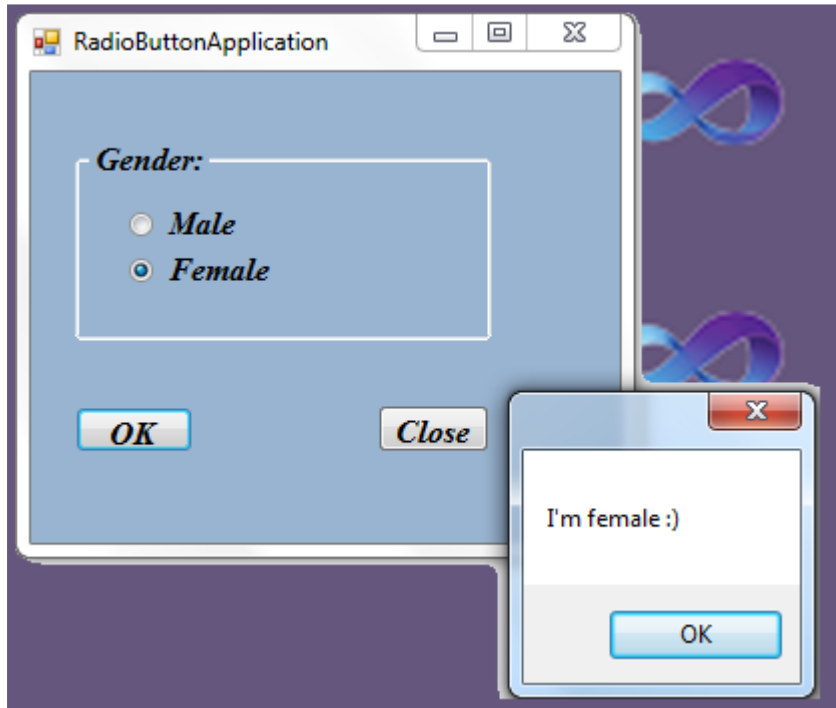
الصيغة العامة أو القاعدة المتبعة للدالة الشرطية :

```
If condition (الشرط الواجب تحققه) then
Statement (التعليمات أو الإجراءات التي تنفذ في حالة تحقق الشرط)
.....
.....
End If (نهاية تنفيذ الإجراء)
```



تنفيذ الدالة الشرطية (If Statement) من خلال الأوامر البرمجية :

تتضمن التطبيق التالي خانتي اختيار بداخل صندوق المجموعة Group- Box والذي يحدد الجنس Gender وله خيارين إما خانة الاختيار Male أو خانة الاختيار Female وعند تنفيذ الدالة الشرطية IF Statement لكلا الخيارين وتشغيل التطبيق يتم النقر على إحدى الخيارين ثم النقر على الزر OK لتنفيذ الإجراء وهو إظهار رسالة MessageBox تأكيد على تحقق الشرط عند النقر على إحدى الخيارين ويستخدم الزر Exit للخروج من التطبيق .



يتم كتابة الكود التالي عند النقر المزدوج على أداة الزر OK:

```
If RadioButton1.Checked = True Then  
    MessageBox.Show("I'm male:")  
End If  
If RadioButton2.Checked = True Then  
    MessageBox.Show("I'm female:")  
End If
```

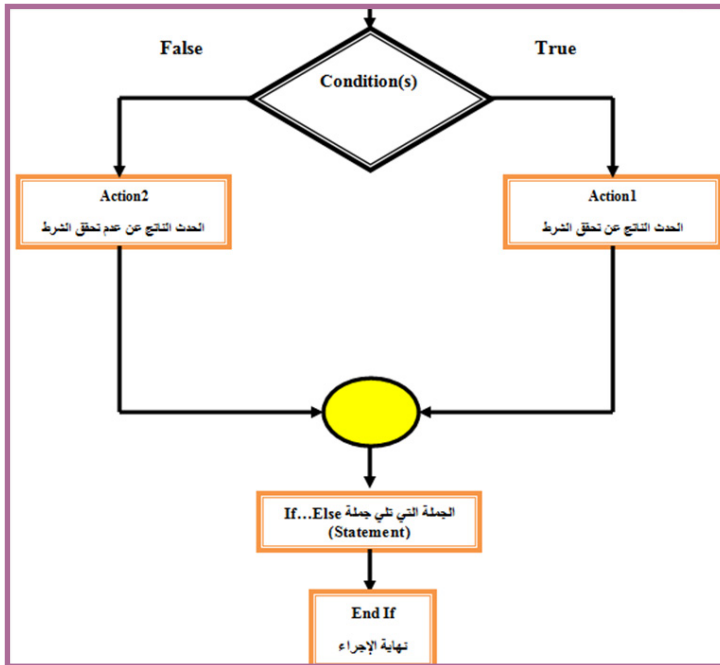
وعند النقر على الزر Exit يتم كتابة الكود التالي لإغلاق البرنامج أو التطبيق: End

النوع الثاني من أنواع الدالة الشرطية If Statement وهو : If-Else-End If

الصيغة العامة له كالآتي:

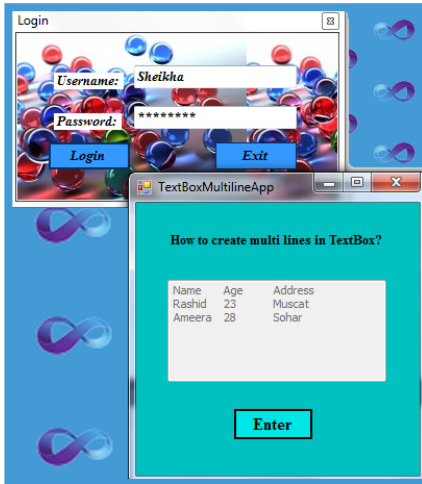
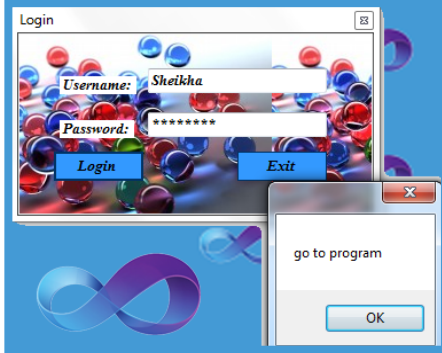
”الشرط الواجب تحققه“ **If (condition) Then**
”في حالة تحقق الشرط يتم تنفيذ هذا الإجراء“ **Statement1**
”تستخدم للفصل بين إجراءين أو أمرين برمجيين“ **Else**
”في حالة عدم تحقق الشرط يتم تنفيذ الإجراء الثاني“ **Statement2**
”نهاية تنفيذ الإجراء“ **End If**

يستخدم هذا النوع من الدالة الشرطية **Conditional Function** في حالة تحقق شرط معين وإن لم يتحقق ذلك الشرط فإن البرنامج يقوم بتنفيذ إجراء أو بيان **Statement** آخر.

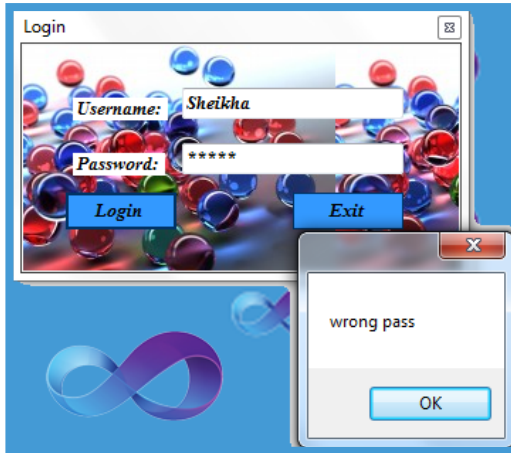


تطبيق :

برمجة صفحة رمز المستخدم (Login Page) :



تحليل الشاشة: تتكون من متغيرين اسم المستخدم **User name** و كلمة المرور/الرقم السري **Password** وعند تنفيذ وتشغيل البرنامج أو الصفحة يتم إدخال البيانات المحددة في صفحة الكود لكل من المتغيرين بطريقة صحيحة ويقوم البرنامج بإرسال رسالة ترحيب ثم الانتقال إلى الصفحة الثانية **List Box Application** وهي البرنامج وفي حالة إدخال البيانات بشكل خاطئ يقوم البرنامج بإرسال رسالة تنبيه بإدخال اسم المستخدم أو كلمة المرور خطأ.



يتم كتابة الكود التالي
عند النقر المزدوج على
أداة الرز Login و Exit :

Login Button :

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

```
Dim a As String "تعريف المتغير"
```

```
Dim b As Integer
```

```
a = "Sheikha" كل متغير " في كل متغير "
```

```
b = 12345
```

```
"كتابة الدالة الشرطية (If Statement) للمتغيرين a و b للتحقق من صحة كلا الشرطين."
```

```
If TextBox1.Text = a And TextBox2.Text = b Then
```

```
MessageBox.Show("Welcome to Application One")
```

```
Application1.Show() "تنفيذ الإجراء في حالة تحقق الشرطين"
```

```
Else
```

```
"تنفيذ الإجراء في حالة عدم تحقق صحة إحدى أو كلا الشرطين"
```

```
MessageBox.Show("Wrong Username and Password!")
```

```
End If
```

```
End Sub
```

Exit Button :

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

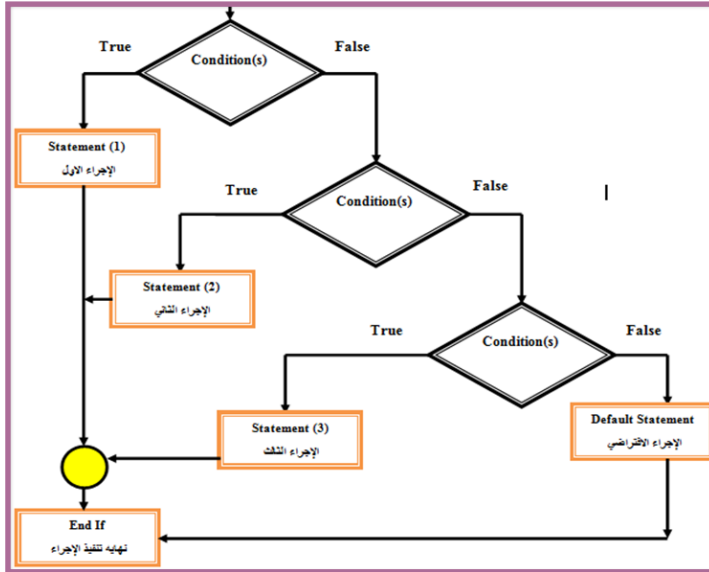
تحدي :

قم بتنفيذ أمر برمجي في حالة إدخال اسم المستخدم **Username** أو كلمة المرور **Password** خطأ ثلاث مرات، يطلب البرنامج إعادة كتابة اسم مستخدم أو كلمة مرور جديدة من خلال إظهار رسالة تنبيه ثم يتم الانتقال إلى شاشة أخرى، بحيث تظهر بعد شاشة الدخول للبرنامج أو التطبيق .

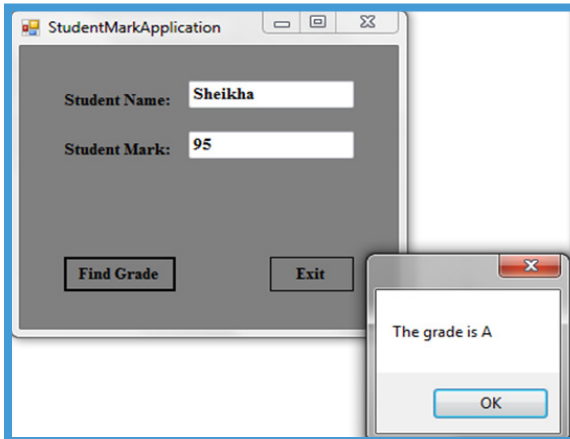
النوع الثالث: الدالة الشرطية Elseif Statement:

(Elseif) هو نوع آخر من أنواع الدالة الشرطية Conditional Function وتستخدم لتنفيذ أمر برمجي في حالة التحقق من عدة شروط فإن لم يتحقق الشرط الأول يتم اختبار الثاني وإن لم يتحقق الشرط الثاني يتم اختبار الشرط الثالث وهكذا ... حتى يتحقق الشرط وإن لم يتحقق أي من الشروط فإن البرنامج يقوم بتنفيذ الإجراء أو العملية في Statement الواقع بعد Else. القاعدة أو الصيغة العامة المتبعة :

```
If (condition) Then  
Statement 1  
Else If (condition) Then  
Statement 2  
Else If (condition) Then  
Statement 3  
...  
Else  
Statement  
End If
```



مثال تطبيقي على (Elseif) :



برنامج يقوم بإظهار رسالة تحتوي على معدل الطالب على حسب النتيجة المدخلة في مربع النص Student Mark حيث يشترط لكل نتيجة معدل معين. فمثلا : إذا كانت نتيجة الطالب ما بين 100 و90 فإنه سيحصل على المعدل أ أو Grade A، وإذا كانت نتيجة الطالب ما بين 90 و80 فإنه سيحصل على المعدل ب أو Grade B، وهكذا... وهذا يعني للحصول على معدل معين لابد من تحقق صحة شرط النتيجة المدخلة في البرنامج .

يتم كتابة الكود التالي عند النقر المزدوج على أداتي الزر Find
Grade و Exit:

Find Grade Button :

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim StudentName As String
    Dim StudentMark As Integer
    StudentName = TextBox1.Text
    StudentMark = Val(TextBox2.Text)
```

```
    If StudentMark >= 90 Then
        MessageBox.Show("The grade is A")
    Else If StudentMark >= 80 Then
        MessageBox.Show("The grade is B")
    Else If StudentMark >= 60 Then
        MessageBox.Show("The grade is C")
    Else If StudentMark >= 50 Then
        MessageBox.Show("The grade is D")
    Else
        MessageBox.Show("The grade is F")
    End If
End Sub
```

Exit Button :

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

الدالة (Select Case)

تعني هذه الدالة "اختر الحالة" وتستخدم عندما يكون للمتغير عدة حالات أو قيم أو احتمالات. مثلا يقوم المستخدم بإدخال رقم أصغر من 5، فيقوم البرنامج بطباعة رسالة بحسب الرقم المدخل عوضا عن تكرار استخدام الدالة If و Else.

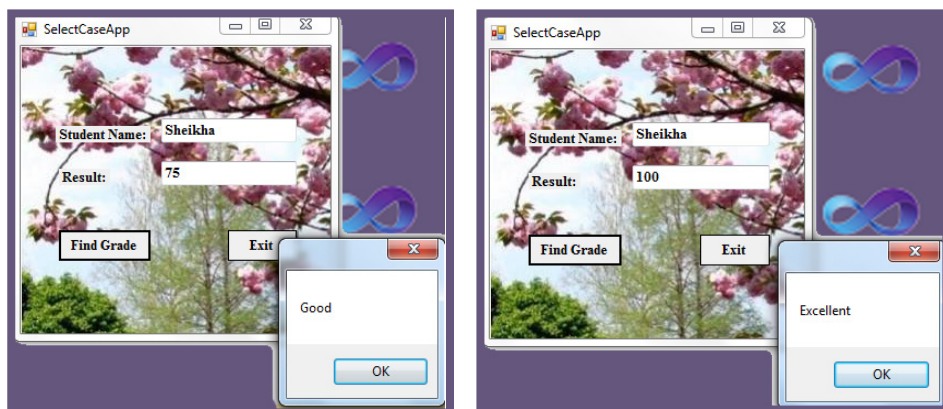
القاعدة المتبعة :

```
Select Case expression "المتغير الذي سيتم اختباره"  
Case value1 "قيمة المتغير في الحالة الأولى"  
Statements (s)  
Case value2  
Statements (s)  
Case value3  
Statements (s)  
Case Else  
Statements (s)  
End Select
```

مثال تطبيقي على (Select Case) :

برنامج معرفة مستوى الطالب

تحليل التطبيق: برنامج يقوم بإظهار رسالة تحتوي على مستوى الطالب على حسب النتيجة المدخلة في مربع النص Result حيث يشترط لكل نتيجة مستوى أو حالة معينة. **فمثلاً:** في الحالة الأولى إذا كانت نتيجة الطالب أكبر من أو يساوي 90 فإنه سيحصل على المستوى Excellent، وفي الحالة الثانية إذا كانت نتيجة الطالب أكبر من أو يساوي 80 فإنه سيحصل على المستوى Very Good، وهكذا... وهذا يعني للحصول على مستوى معين توجد أكثر من حالة ونتيجة ولا بد من تحقق صحة شرط النتيجة المدخلة في البرنامج.



يتم كتابة الكود التالي عند النقر المزدوج على أداتي الزر Find و Grade :Exit

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

```
Dim SN As String
```

```
Dim SM As Integer
```

```
SN = TextBox1.Text
```

```
SM = Val (TextBox2.Text)
```

```
Select Case SM
```

Case Is >= 90

MessageBox.Show("Excellent")

Case Is >= 80

MessageBox.Show("Very Good")

Case Is >= 70

MessageBox.Show("Good")

Case Is >= 60

MessageBox.Show("Acceptable")

Case Else

MessageBox.Show("failed")

End Select

End Sub

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

End

End Sub

ملاحظة:

طريقة أخرى لتفعيل الدالة (Select Case) هي إضافة (To) بين قيمتين. مثال: (Case 0 To 5)

الدالة التكرارية (Repetition)
(or Looping Function)

الدالة التكرارية (Looping) هي عملية التكرار والاستمرار في تنفيذ مجموعة من الجمل البرمجية طالما بقي الشرط محققا أو حتى يتحقق الشرط أو لعدد يحدده المستخدم من المرات. ومن أنواعها : **While-End** ، **While** ، **Do-Loop** ، **For** .

النوع الأول: While-End While :

تستخدم هذه الدالة للاستمرار في تنفيذ كود معين (أمر برمجي) أو إعادة تنفيذه خلال حلقة تكرارية طالما بقي الشرط المتعلق به محققا أو حتى يتحقق الشرط وفي حالة عدم تحقق الشرط يخرج من الحلقة التكرارية.

القاعدة المتبعة:

” شرط تكرار الحلقة ” While (Condition)

Statement1 } “التعليمات أو الإجراءات التي تنفذ بعد تحقق الشرط”

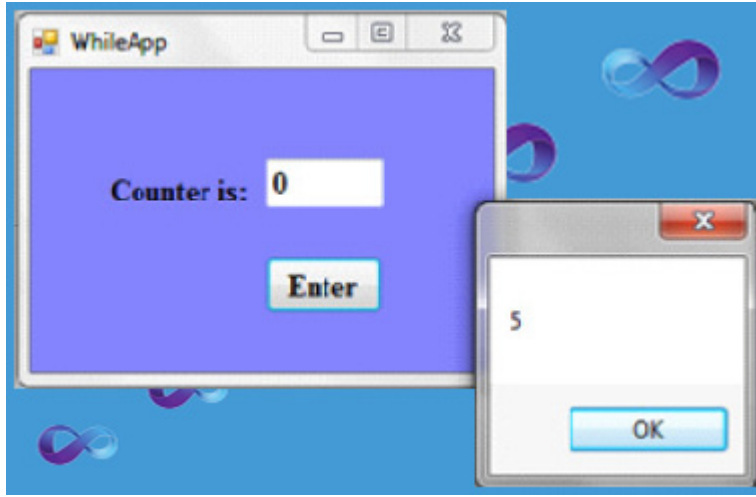
Statement2 }

Statement(s) }

End While "إنهاء التكرار"

مثال تطبيقي على (While Function) :

تنفيذ برنامج يقوم بإظهار رسالة خمس مرات للمستخدم ويكتب فيها ترتيب التكرار. وشرط تكرار هذه الحلقة أن تكون قيمة ال **counter** أصغر من 6 وعندما تصبح قيمته 6 سيخرج البرنامج من الحلقة ويتوقف عن التكرار. وعند تنفيذ الحلقة في كل مرة يتم طباعة رسالة تحتوي على قيمة ال **counter**.



يتم كتابة الكود التالي عند النقر المزدوج على أداة الزر Enter :

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles Button1.Click  
Dim counter As Integer  
counter = TextBox1.Text  
counter = 0  
While (counter < 6)  
MessageBox.Show(counter)  
counter = counter + 1  
End While  
End Sub
```

النوع الثاني: الدالة التكرارية For...Next :

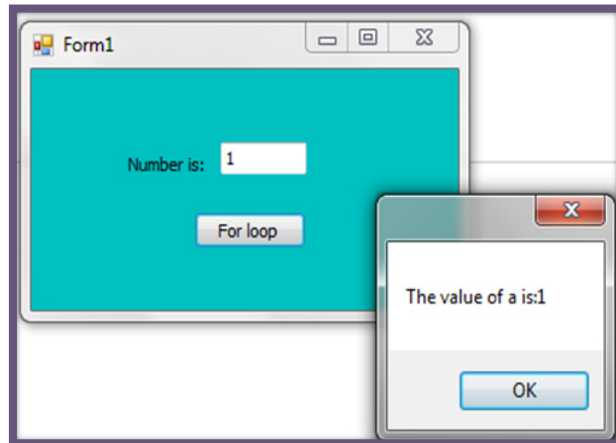
الدالة (For Loop) تقوم بتنفيذ تكرار الكود أو الجملة البرمجية المحصورة بين العدد أو القيمة الأولى (الابتدائية) من أعداد التكرار والقيمة النهائية التي تتوقف عندها حلقة التكرار.

القاعدة المتبعة:

```
For [variable] = Start To End  
Statement1  
Statement2  
...  
Next [Variable]
```

مثال تطبيقي على الدالة التكرارية For...Next :

تحليل التطبيق: تصميم برنامج يقوم بتنفيذ الحلقة التكرارية للمتغير (a) بين القيمة الأولى (0) وتزايد إلى أن تصل إلى القيمة النهائية (10) التي تتوقف عندها الحلقة التكرارية مع إظهار أعداد التكرار في كلا من مربع النص TextBox1 ورسالة MessageBox في نفس الوقت.



يتم كتابة الكود التالي عند النقر المزدوج على أداة الزر For : Loop

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim a As Integer = 0
    "بداية الحلقة التكرارية من العدد 0 إلى
    العدد النهائي 10 الذي يتوقف عنده التكرار"
    For a = 0 To 10
        TextBox1.Text = a
        MessageBox.Show("The value of a is:" & a)
    Next a
End Sub
```

النوع الثالث: الدالة التكرارية Do...Loop :

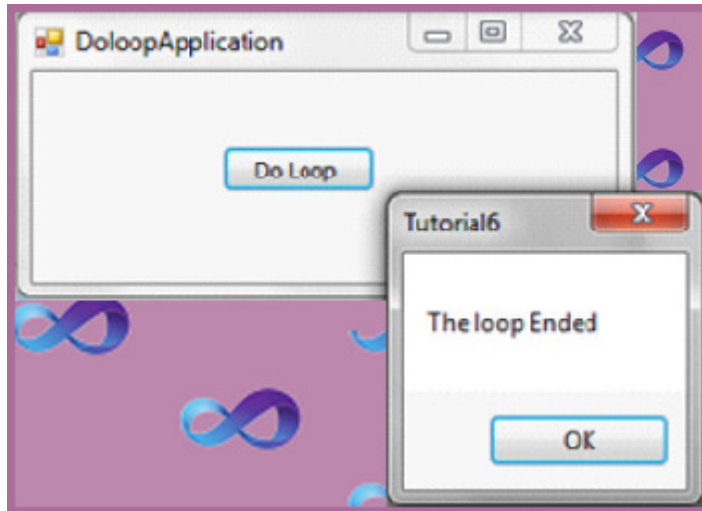
تستخدم هذه الدالة في الاستمرار في تكرار تنفيذ الكود المرفق لها وبالتالي على المبرمج وضع الشرط الذي يؤدي إلى إيقافها ضمن الكود المرفق باستخدام Exit Do التي تخرج من الحلقة التكرارية وليس من الإجراء كله. ولهذه الدالة أنواع منها : Do-While-Loop, Do-Until-Loop .

القاعدة المتبعة:

```
Do
Statement(s)
Exit Do
Statement(s)
Loop
```

مثال تطبيقي:

تحليل التطبيق: يقوم هذا التطبيق بتكرار كود معين عند تشغيله حيث ندخل في حلقة تكرارية بإظهار قيمة العداد (counter) في رسالة Message-Box حتى تصبح قيمته (3) وبعدها يخرج من الحلقة التكرارية فقط بسبب وجود الأمر البرمجي Exit Do ثم يتم تنفيذ الإجراء الذي يلي الحلقة التكرارية وهو ظهور رسالة MessageBox (The loop Ended).



يتم كتابة الكود التالي عند النقر المزدوج على الزر Do Loop :

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

```
Dim counter As Integer
```

```
counter = 0
```

```
Do
```

```
If counter = 3 Then
```

```
Exit Do "يستخدم هذا الأمر البرمجي للخروج من الحلقة التكرارية فقط"
```

```
MsgBox("The loop Ended")
End If
MsgBox(counter)
counter = counter + 1
Loop
MsgBox("The loop Ended")
End Sub
```

النوع الرابع: الدالة التكرارية Do-While-Loop

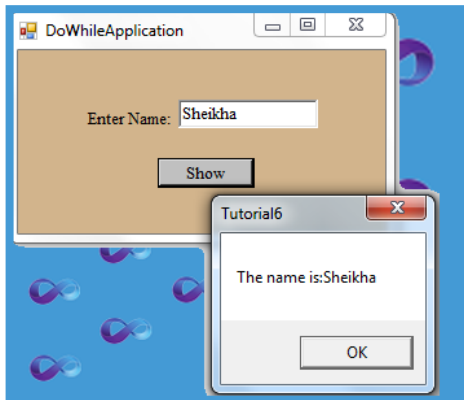
في هذه الدالة يقوم البرنامج بتكرار الكود المرفق له طالما بقي الشرط محققا على يمين While وعند عدم تحققه فإن البرنامج يخرج من الحلقة التكرارية أي بعكس عمل الدالة Do-Until-Loop.

القاعدة المتبعة:

Do While “شرط تكرار الحلقة”
Statement1
Statement2
Statement(s)
Loop “إنهاء عملية التكرار”

مثال تطبيقي:

يجمع بين الدالة التكرارية
Exit-Do و Do-While-loop :



تحليل البرنامج:

تنفيذ برنامج يقوم بإظهار رسالة `MessageBox` 5 مرات بشكل تكراري ويوجد بها الاسم المدخل عند إدخال أي اسم في مربع النص `TextBox1` وإيقاف التكرار أو الحلقة التكرارية بإظهار رسالة أخرى تحتوي على كلمة "End" من خلال الأمر البرمجي `Exit Do`:

يتم كتابة الكود التالي عند النقر المزدوج على الزر `Show`:

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

```
Dim i As Integer = 1 "هذا المتغير يستخدم في عداد التكرار"
```

```
Dim a As String "المتغير الذي يتم إدخاله في مربع النص كاسم"
```

```
a = TextBox1.Text
```

```
Do While i <= 5
```

```
    "شروط تكرار الحلقة وهو إظهار رسالة بالاسم المدخل 5 مرات فقط"
```

```
    MessageBox.Show("The name is:" & a)
```

```
    If a = "End" Then
```

```
        Exit Do "إنهاء الحلقة التكرارية بإظهار رسالة End"
```

```
    End If
```

```
    TextBox1.Text = a
```

```
    "إكمال الإجراء عند إدخال اسم في مربع النص وإظهار 5 رسائل بالتزايد"
```

```
    i = i + 1
```

```
Loop "الخروج من الإجراء كله"
```

```
End Sub
```

النوع الخامس: الدالة التكرارية Do-Until-loop

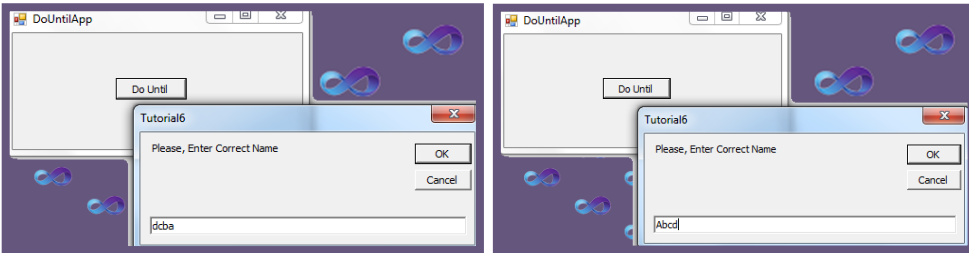
في هذه الدالة يقوم البرنامج بالاستمرار في تكرار الكود المرفق لها حتى يتحقق الشرط أو الاستمرار في التكرار في حالة عدم تحقق الشرط.

القاعدة المتبعة:

```
Do Until (condition)
Statement1
Statement 2
...
Loop
```

مثال تطبيقي:

تحليل التطبيق: يقوم هذا التطبيق بالاستمرار في تكرار ظهور مربع الإدخال Input Box لكتابة الاسم الصحيح فيه في حالة عدم تحقق الشرط بكتابة كلمة خاطئة أو التكرار حتى يتحقق الشرط وذلك بإدخال الكلمة الصحيحة في صندوق الإدخال وهي "Abcd".

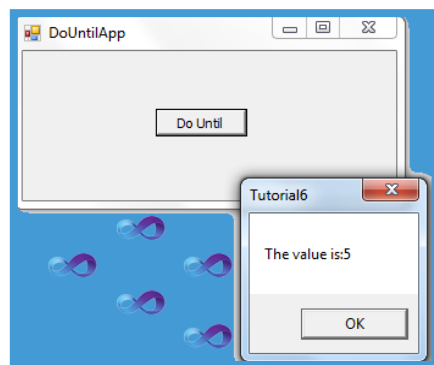
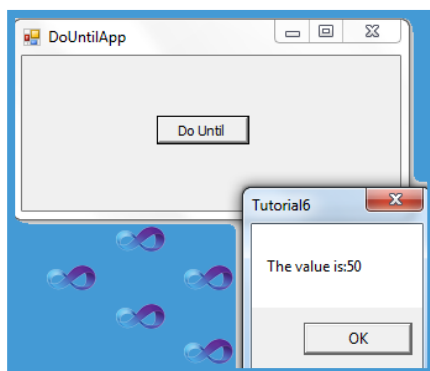


بعد تصميم التطبيق، يتم كتابة الكود التالي من خلال النقر المزدوج على الزر Do Until:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Dim x As String = "Abcd"
Do
x = InputBox("Please, Enter Correct Name")
Loop Until x = "Abcd"
End Sub
```

تطبيق: مقارنة بين الدالة التكرارية Do-Until- loop و Do-While-loop

تحليل التطبيق: ينفذ هذا التطبيق حلقة تكرارية لأمر برمجي وهو ظهور تزايد قيمة العدد (Number) كل مرة 5 مرات حتى تصبح قيمة العدد (= Number 50) وبعدها يتوقف عن الحلقة التكرارية.



يتم كتابة الكود التالي عند النقر المزدوج على الزر **Do Until**:

سنلاحظ في الدالة **Do-Until-Loop** إن البرنامج يقوم بالاستمرار في تكرار الكود المرفق لها في حالة عدم تحقق الشرط أو حتى يتحقق الشرط المطلوب وهو بأن قيمة العدد تساوي 50.

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

```
Dim Number As Integer
```

```
Do Until (Number = 50)
```

```
Number = Number + 5
```

```
MsgBox("The value is:" & Number)
```

```
Loop
```

```
End Sub
```


أما في الدالة **Do-While-Loop** يقوم البرنامج بتكرار الكود المرفق لها طالما بقي الشرط محققا على يمين **While** وعند عدم تحققه فإن البرنامج يخرج من الحلقة التكرارية أي بعكس عمل الدالة **Do-Until-Loop**. وهذا ما لاحظناه عند تطبيق نفس الأمر البرمجي وهو بأن قيمة العدد (**Number = 50**) لم يتحقق شرط الدالة التكرارية **Do-While-Loop** حيث خرج مباشرة من الحلقة التكرارية وتم تنفيذ الحلقة التكرارية بتغيير شرط تكرار الحلقة وهو بأن قيمة العدد (**Number <= 50**) كما هو موضح في الكود أو الأمر البرمجي التالي:

تنفيذ الأمر البرمجي:

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

```
Do While Number <= 50
```

```
MsgBox("The value is:" & Number)
```

```
Number = Number + 5
```

```
Loop
```

```
End Sub
```



ضع دائما صورتك التي تريد أن تكون
عليها في عقلك ومخيلتك , وستتجه
تدريجيا نحوها .. اذا لم تهزم نفسك ,
ستهزمك نفسك .. سلم النجاة لا يعاني
من الإزدحام في أعلاه

نابليون هيل
كاتب أمريكي

الفصل الرابع

المصفوفات (Array) 1

الدرس الثاني

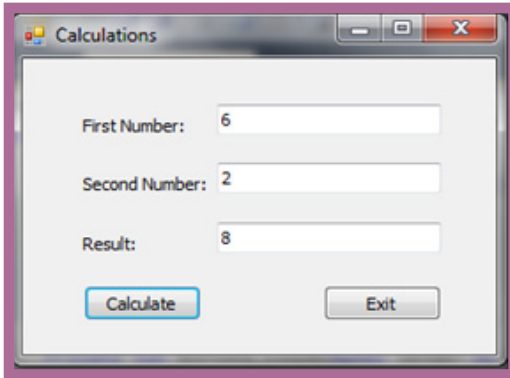
الدالة (Mod Function) 2

الدالة (Val Function) 1

الدرس الأول

الدالة (Val Function)

هي الدالة التي تقوم بتحويل القيمة النصية (String) إلى قيمة عددية (Integer) ومثال ذلك: `Val (TextBox1.Text)`



مثال تطبيقي :

تحليل برنامج: يقوم البرنامج التالي بتنفيذ عملية حسابية وهي "الجمع" عند إدخال رقمين في مربع النص `TextBox1,2` والنقر على زر `Calculate` لإظهار الناتج في مربع النص `Result`.

لكتابة الكود أو الأمر البرمجي يتم النقر المزدوج على زر `Calculate` فيما يلي:
`Dim a, b As Integer`

"تعريف متغيرين ألا وهما الرقم الأول والرقم الثاني في البرنامج"

"بعد ذلك يتم تحديد مكان أو أداة التحكم لكل متغير في البرنامج مع كتابة خاصية (Function Val) لإجراء العملية الحسابية"

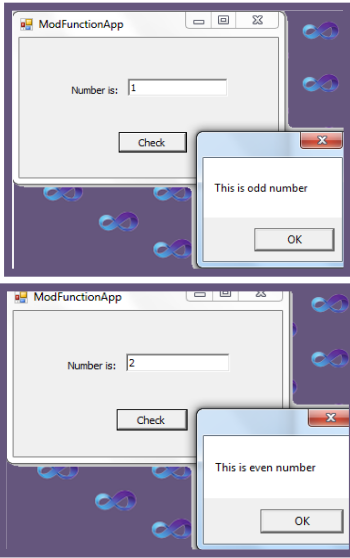
`a = Val(TextBox1.Text)`

`b = Val(TextBox2.Text)`

`TextBox3.Text = a + b`

"تنفيذ العملية الحسابية بين المتغيرين لإظهار الناتج في مربع النص الثالث"

الدالة (Mod Function)



تستخدم هذه الدالة لتحديد ما إذا كان الرقم المدخل في التطبيق زوجيا Even أم فرديا Odd.

مثال تطبيقي :

تحليل البرنامج: في هذا البرنامج يطلب من المستخدم إدخال أي رقم في مربع النص Text-Box1 وعند النقر على زر Check يقوم البرنامج بإرسال رسالة تحدد ما إذا كان الرقم المدخل زوجيا Even أم فرديا Odd وذلك من خلال استخدام الدالة Mod Function.

يتم كتابة الكود التالي عند النقر المزدوج على الزر Check:

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

```
Dim a As Integer "المتغير (العدد) الذي يتم إدخاله في مربع النص"
```

```
a = TextBox1.Text
```

```
If a Mod 2 = 0 Then
```

```
"لو كان باقي قسمة العدد على 2 يساوي 0 فهذا يعني إن العدد زوجي"
```

```
MessageBox.Show("This is even number")
```

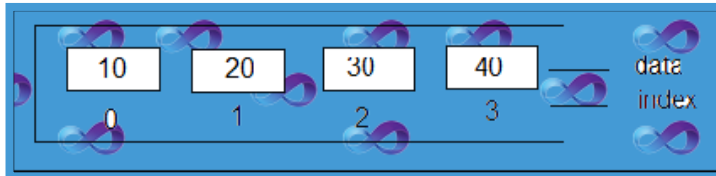
```
Else
```

```
MessageBox.Show("This is odd number")
```

```
End If End Sub
```

المصفوفات (Array)

المصفوفة : هي نوع من أنواع الدالة التسلسلية **Serial Function** وهي عبارة عن متغيرات مفهرسة (**Indexed Variables**) تحتوي على بيانات عديدة من نفس النوع **Data Type** وكل مصفوفة لها اسم واحد يمكن استخدامه للرجوع إلى أي عنصر (**Element**) فيها وذلك باقتران هذا الاسم بدليل يمثل مكان العنصر فيها، ويمكن إنشاء مصفوفة لاحتوائها على أي نوع من أنواع البيانات مثل : النصوص والأعداد الحقيقية والصحيحة وغيرها. وأهميتها تكمن في صناعة أوامر برمجية (**Codes**) قصيرة وبسيطة ذات قوة كبيرة لأنه يمكن بناء **Loops** تتعامل بكفاءة مع المصفوفات مهما كان عدد عناصرها (**Elements**).



شكل المصفوفة المخزنة في ذاكرة الحاسوب

كيفية تنفيذها كأمر برمجي في برنامج الفيجوال بيسك:

```
Dim (Array Name) As DataType ( ) = New DataType (No. of Index ,
{Elements of Array})
```

"تعريف المصفوفة مع نوع البيانات الموجودة فيها وفهرستها"
يتم تعريف العناصر في المصفوفة بطريقتين:

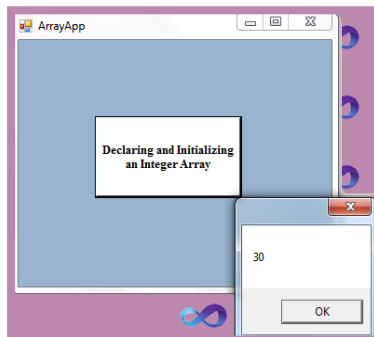
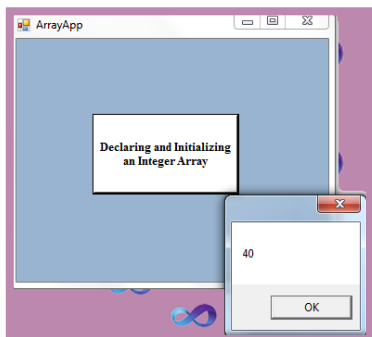
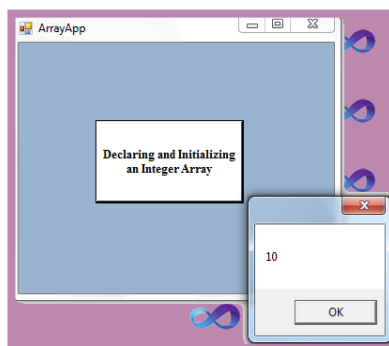
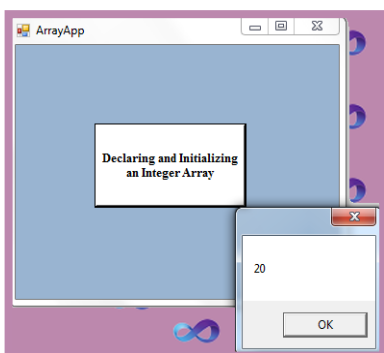
- 1 {Elements of Array}
- 2 ArrayName(No. of Index) = Element

"إدخال المصفوفة في دالة تكرارية" **Loop Function**

النوع الأول : المصفوفة من نوع بيانات أعداد صحيحة (Integer)

مثال تطبيقي :

تحليل التطبيق: من خلال هذا التطبيق يتم تعريف أربعة عناصر (Elements) في مصفوفة من نوع بيانات أعداد صحيحة (Integer) حيث يتم تعيين كل عنصر داخل فهرس المصفوفة (Index of Array) الذي يبدأ دائما من الصفر 0 إلى 3 ويمكن استرداد هذه العناصر من المصفوفة من خلال تنفيذ الدالة التكرارية For-Next.



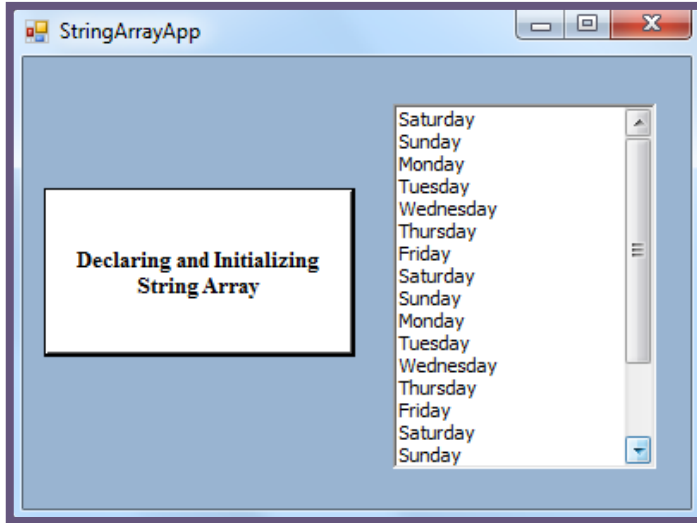
بعد تصميم التطبيق أعلاه، يتم كتابة الكود التالي من خلال النقر
المزدوج على الزر **Declaring and Initializing an Integer Array**

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim array As Integer() = New Integer(3) { }
    "{10, 20, 30, 40}"
    array(0) = 10
    array(1) = 20
    array(2) = 30
    array(3) = 40
    For i As Integer = 0 To array.Length - 1
        MessageBox.Show(array(i))
    Next
End Sub
```

النوع الثاني: المصفوفة من نوع بيانات حروف (String)

مثال تطبيقي :

تحليل التطبيق: من خلال هذا التطبيق يتم تعريف سبعة عناصر (Elements) في مصفوفة من نوع بيانات حروف (String) وهي أسماء أيام الأسبوع حيث يتم تعيين كل اسم داخل فهرس المصفوفة (week(6)) (Index of Array) الذي يبدأ دائما من الصفر 0 إلى 6 ويمكن استرداد هذه الأسماء من المصفوفة من خلال تنفيذ الدالة التكرارية For-Next وتحديد طول مصفوفة أيام الأسبوع وعند تشغيل التطبيق يتم إظهار أسماء أيام الأسبوع في القائمة (ListBox1) واستردادها من المصفوفة (Week(6)) عند النقر على الزر.



بعد تصميم التطبيق أعلاه، يتم كتابة الكود التالي عند النقر المزدوج على الزر
: Declaring and Initializing String Array

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

```
Dim i As Integer
```

```
Dim week(6) As String
```

```
week(0) = "Saturday"
```

```
week(1) = "Sunday"
```

```
week(2) = "Monday"
```

```
week(3) = "Tuesday"
```

```
week(4) = "Wednesday"
```

```
week(5) = "Thursday"
```

```
week(6) = "Friday"
```

```
For i = 0 To week.Length - 1
```

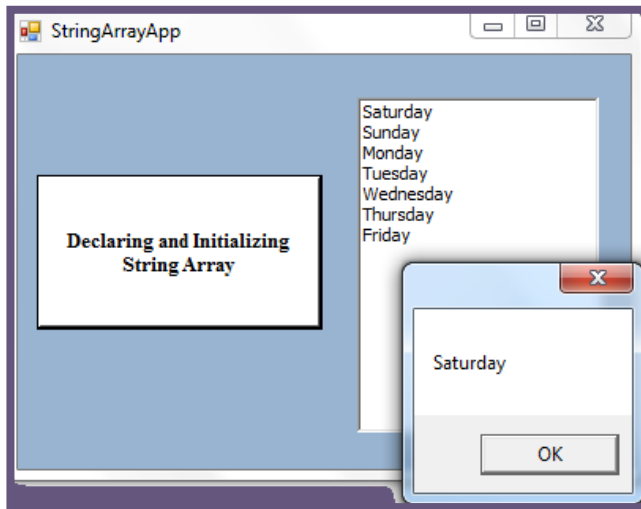
```
ListBox1.Items.Add(week(i))
```

```
Next
```

```
End Sub
```

ملاحظة: ولتحديد ظهور أو اختيار اسم واحد من عناصر المصفوفة (Week(6)) في التطبيق يتم تحديد اسم العنصر المطلوب وفهرسته من خلال كتابة الأمر البرمجي التالي:

```
Dim weekName As String = week(0)
If week(0) = weekName Then
MessageBox.Show(week(0))
End If
```

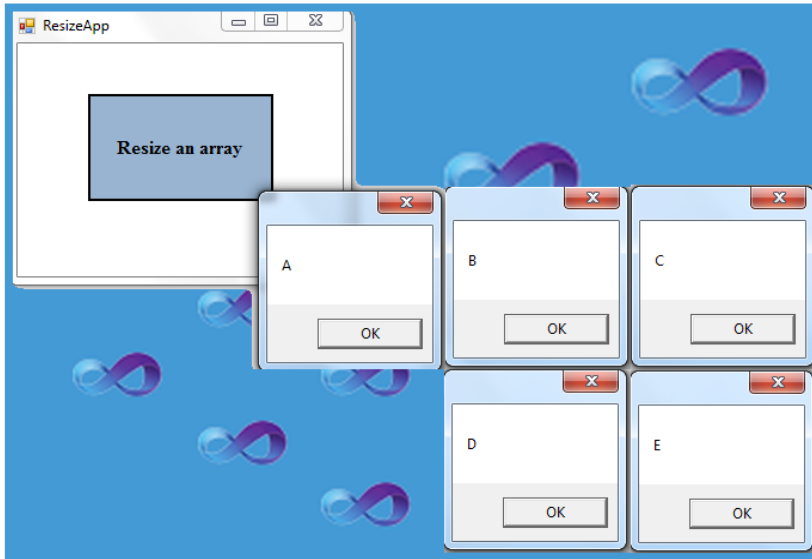


كيفية تغيير حجم المصفوفة (Array.Resize)

يمكن تغيير حجم أي مصفوفة، وتصبح مصفوفة ذات حجم أكبر أو أصغر من العناصر المخزنة بها من خلال الأمر البرمجي (Array.Resize). وهذا يعني تغيير عدد من عناصر المصفوفة أحادية البعد إلى حجم جديد آخر.

مثال تطبيقي :

تحليل التطبيق: خلال هذا التطبيق يتم تغيير حجم المصفوفة (Abcde) من الحجم الأصلي المكون من خمس عناصر مفهرسة من 0 إلى 4 وهي A- B- C- D- E والنقر على الزر **Resize an array** إلى الحجم الجديد المكون من عنصرين من 0 إلى 1 وهي A-B التي تظهر في رسالة **MessageBox** بعد إظهار الحجم الأصلي للمصفوفة عند تشغيل البرنامج.



بعد تصميم التطبيق أعلاه، يتم كتابة الكود من خلال النقر المزدوج على الزر
: `Resize an array`

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

```
Dim Abcde As Char() = New Char(4) {}
```

```
Abcde(0) = "A"
```

```
Abcde(1) = "B"
```

```
Abcde(2) = "C"
```

```
Abcde(3) = "D"
```

```
Abcde(4) = "E"
```

```
For i As Integer = 0 To Abcde.Length - 1 "المصفوفة الأصلية"
```

```
MessageBox.Show(Abcde(i))
```

```
Next
```

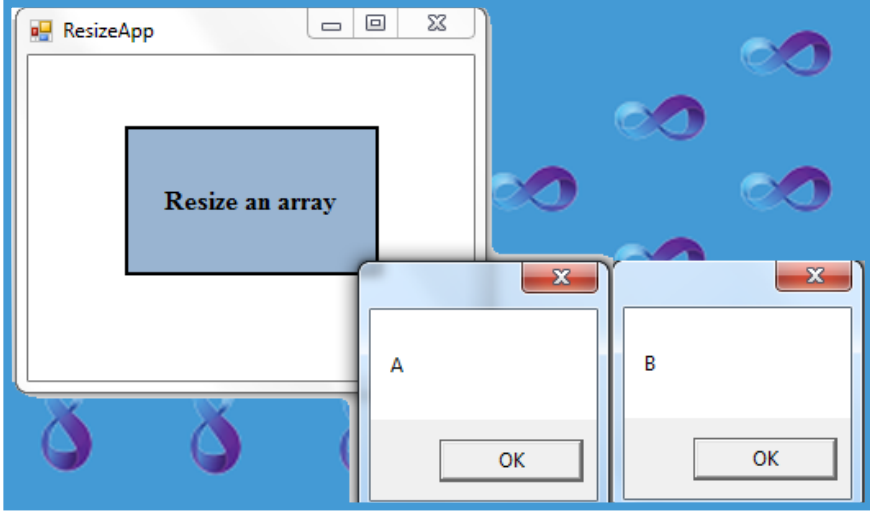
```
Array.Resize(Abcde, 2) "المصفوفة بعد تغيير حجمها"
```

```
For i As Integer = 0 To Abcde.Length - 1
```

```
MessageBox.Show(Abcde(i))
```

```
Next
```

```
End Sub
```



"عناصر المصفوفة بعد تغيير حجمها"

المصفوفة الحيوية (Dynamic Array) <

في بعض الأحيان قد لا يمكننا معرفة كيف يمكن زيادة حجم المصفوفة وربما لا تكون لدينا القدرة على تغيير حجم المصفوفة في وقت تشغيل البرنامج. و الحل هو **المصفوفات الحيوية (Dynamic Array)** حيث لها القدرة على تغيير حجمها في أي وقت وهي من بين الميزات الأكثر مرونة ومريحة في Visual Basic، والتي تساعدك على إدارة الذاكرة بكفاءة. على سبيل المثال، يمكنك استخدام مصفوفة كبيرة لفترة قصيرة ومن ثم تحرير الذاكرة في النظام عندما لا نحتاج إلى استخدامها في المصفوفة. والبدل هو أن يتم تعريف أو إعلان (Declare) مصفوفة ذات حجم أكبر ومن ثم تجاهل عناصر المصفوفة التي لا نحتاج إليها.



ويمكننا التعريف أو الإعلان عن مصفوفة حيوية من خلال إعطاءها قائمة أبعاد فارغة (An Empty Dimension List). مثال ذلك: `Dim DynArray ()`.

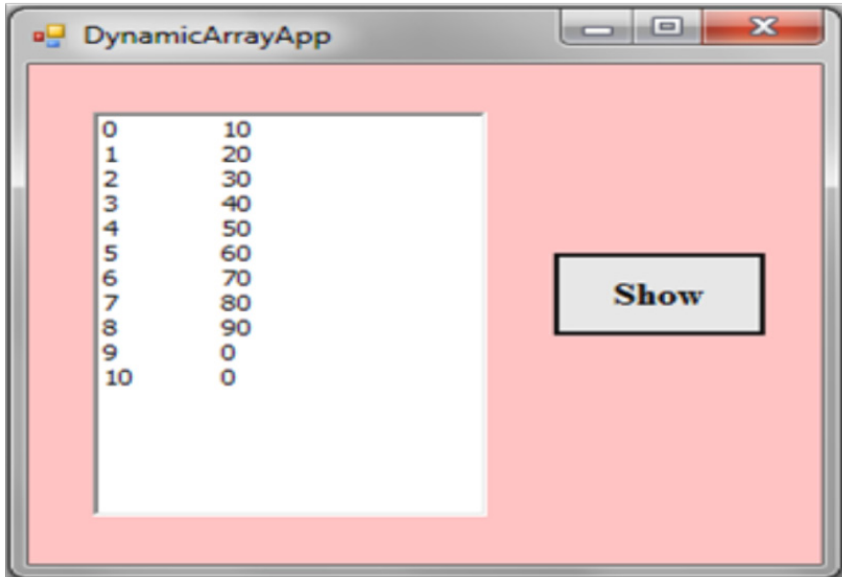
ثم تخصيص العدد الفعلي من العناصر (Exact number of elements) في المصفوفة من خلال الأمر البرمجي التالي مثلا:
`ReDim DynArray (X + 1)`.

حيث (X) توضح عدد العناصر في المصفوفة الفارغة أو الأصلية (X+1) هي عند إضافة عناصر أخرى لتغيير حجمها.

حيث تستخدم `ReDim` في الإجراء `Statement` حيث تكون قابلة لتنفيذ الإجراء (`an executable statement`) أثناء وقت تشغيل البرنامج. وهي تدعم نفس بناء الجملة المستخدمة للمصفوفة الثابتة (`Fixed Array`) ويمكنها تغيير عدد العناصر في المصفوفة.

مثال تطبيقي :

تحليل التطبيق: يقوم هذا التطبيق بتنفيذ المصفوفة الحيوية `Dynamic Array` ذات قائمة أبعاد فارغة `marray ()` لتغيير حجم المصفوفة من `marray(2)` إلى `marray(10)` ذات حجم أكبر حيث يتم تخصيص عدد العناصر الفعلية لكل مصفوفة وتجاهل عناصر أخرى من المصفوفة التي لا نحتاج إليها في المصفوفة الأكبر التي تم تغيير حجمها من خلال تنفيذ الأمر البرمجي `ReDim Reserve` الذي يستخدم للحفاظ على العناصر الموجودة داخل المصفوفة عند تغيير حجمها. وعند تشغيل البرنامج تظهر جميع العناصر في المصفوفة قبل وبعد تغيير حجمها في قائمة `ListBox1` عند النقر على زر `Show`.



بعد تصميم التطبيق أعلاه، لتنفيذه يتم كتابة الكود عند
النقر المزدوج على الزر Show

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

```
Dim marray() As Integer "تحديد مصفوفة حيوية ذات قائمة أبعاد فارغة"  
ReDim marray(2) "تخصيص عدد من عناصر المصفوفة قبل تغيير حجمها"  
marray(0) = 10  
marray(1) = 20  
marray(2) = 30
```

```
ReDim Preserve )  
"تخصيص عناصر أخرى للمصفوفة عند تغيير حجمها"  
marray(3) = 40  
marray(4) = 50  
marray(5) = 60  
marray(6) = 70  
marray(7) = 80  
marray(8) = 90
```

```
For i = 0 To 10 "دخول عناصر المصفوفة في حلقة تكرارية محددة"  
ListBox1.Items.Add(i & vbTab & marray(i))  
" إظهار جميع عناصر المصفوفة في" ListBox1  
Next i  
End Sub
```



النجاح قمة لا يرتقي سلمها إلا
أصحاب الهمم العالية لأن همتهم
تقودهم إلى المواصلة وإن تعثرت
خطاهم

الفصل الخامس

تطبيقات على أدوات تحكم
مختلفة في الفيچوال بيسك

1

الدرس الثاني

89

تطبيقات على أدوات التحكم
والدوال في الفيچوال بيسك

1

الدرس الأول

تطبيقات على أدوات التحكم والدوال في الفيجوال بيسك

التطبيق الأول:

صندوق أو خانة الاختيار (Check Box): هو عنصر أو أداة تحكم يسمح للمستخدم باختيار أكثر من خيار في نفس الوقت من مجموعة معرفة مسبقاً من الخيارات والتي قد تكون له قيمتان True أو False .

تحليل التطبيق:



يتضمن هذا التطبيق ثلاث خانات اختيار من نوع **CheckBox** بداخل صندوق المجموعة والذي يحدد الهوايات -Hobbies أو خانة الاختيار **Traveling** أو خانة الاختيار **Reading Stories** وعند تنفيذ الدالة الشرطية **IF State** لكل خيار وتشغيل التطبيق يتم النقر على أي خيار أو أكثر من خيار ثم النقر على الزر **Choose** لتنفيذ الإجراء وهو إظهار رسالة **MessageBox** تأكيد على تحقق الشرط عند النقر على أي من الخيارات الثلاثة ويستخدم الزر **Exit** للخروج من التطبيق.

يتم كتابة الكود التالي عند النقر المزدوج على أداة الزر **Choose** :

```
If CheckBox1.Checked = True Then  
    MessageBox.Show("I like traveling(:")  
End If  
If CheckBox2.Checked = True Then  
    MessageBox.Show("I like shopping(:")  
End If  
If CheckBox3.Checked = True Then  
    MessageBox.Show("I like Reading Stories(:")  
End If
```

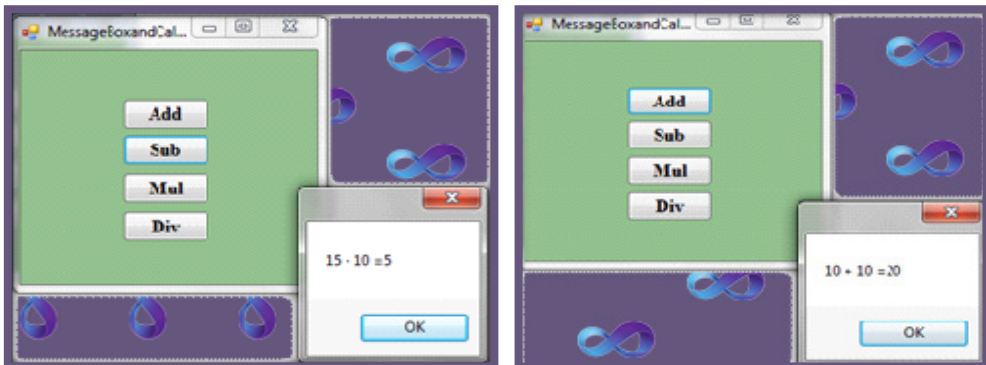
ويتم كتابة الكود التالي عند النقر المزدوج على أداة الزر **Exit** للخروج من التطبيق: **End**

التطبيق الثاني :

تنفيذ العمليات الحسابية عن طريق الأمر البرمجي (MessageBox)

تحليل التطبيق :

تصميم وتنفيذ برنامج يقوم بإظهار رسالة MessageBox لعددتين مع إجراء العمليات الحسابية عليهما لإظهار الناتج من خلال تحديد أربعة أزرار Button للعمليات الحسابية وهي (Add, Sub, Mul, Div) وكتابة أو تنفيذ الأوامر البرمجية فيهم :



يتم كتابة الكود التالي عند النقر المزدوج على كلا من الأزرار
: Add, Sub, Mul, Div

Add Button :

```
Private Sub Button1_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button1.Click  
MessageBox.Show(" 10 + 10 " & "=" & 10 + 10)  
End Sub
```

Sub Button :

```
Private Sub Button2_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button2.Click  
MessageBox.Show(" 15 - 10 " & "=" & 15 - 10)  
End Sub
```

Mul Button :

```
Private Sub Button3_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button3.Click  
MessageBox.Show(" 10 * 10 " & "=" & 10 * 10)  
End Sub
```

Div Button :

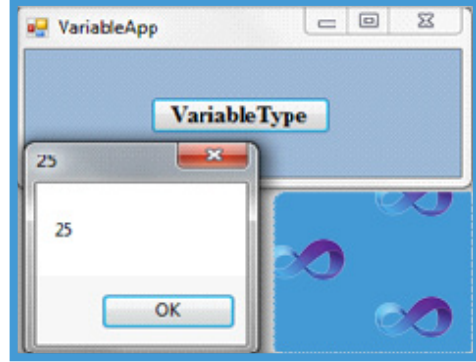
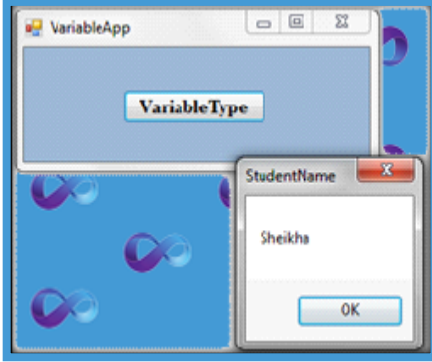
```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles Button4.Click  
MessageBox.Show(" 15 / 5 " & "=" & 15 / 5)  
End Sub
```

التطبيق الثالث :

: أنواع المتغيرات (Variables Types)

تحليل التطبيق:

يقوم هذا التطبيق بتنفيذ إجراء بسيط وهو إظهار رسالتين `MessageBox` الأولى تحتوي على متغير من نوع `String` والثانية تحتوي على متغير ثنائي من نوع `Integer` عند تشغيل البرنامج أو التطبيق والنقر على زر `Variable Type`.



لتنفيذ التطبيق يتم النقر المزدوج على أداة الزر `VariableType` :

```
Dim StudentName As String = "Sheikha" "تعريف المتغير"  
Dim StudentAge As Integer = 25  
MessageBox.Show(StudentName, "StudentName ")  
"إظهار رسالة بيانات المتغير"  
MessageBox.Show(StudentAge, StudentAge)
```

التطبيق الرابع :

من خصائص الدالة `Select Case`: عند استخدام إحدى المعاملات المنطقية مثل (<, >, =, <=, >=, <>...) يتم إضافة (Is) بعد كل حالة (Case).

تحليل التطبيق: في هذا البرنامج أو التطبيق يوجد للمتغير أكثر من حالة حيث أن في الحالة الأولى إذا كان العمر المدخل 12 فستظهر رسالة بالمرحلة التي يكون فيها الشخص خلال هذا العمر وهو بأنه قاصر على حسب القانون وفي الحالة الثانية إذا كان العمر المدخل 15 فإنه مكلف بحسب القانون المدني وهكذا... وهذا يعني إنه في عمر معين (المتغير في البرنامج) توجد حالة خاصة له ويتم تحديده مسبقا من خلال صفحة الكود.



يتم كتابة الكود التالي عند النقر المزدوج على أداة الزر **Enter Age**:

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

```
Dim age As Integer
```

```
age = TextBox1.Text
```

```
Select Case age
```

```
Case 12
```

```
MessageBox.Show ("أنت قاصر بحسب القانون")
```

```
Case 15
```

```
MessageBox.Show ("أنت مكلف بحسب القانون المدني")
```

```
Case 18
```

```
MessageBox.Show ("يحق لك الآن إمتلاك الشركات والإدارة")
```

```
Case 40
```

```
MessageBox.Show ("دخلت مرحلة الوقار")
```

```
Case 65
```

```
MessageBox.Show ("بلغت عمر التقاعد")
```

```
Case Else
```

```
MessageBox.Show ("معلومات أخطري")
```

```
End Select
```

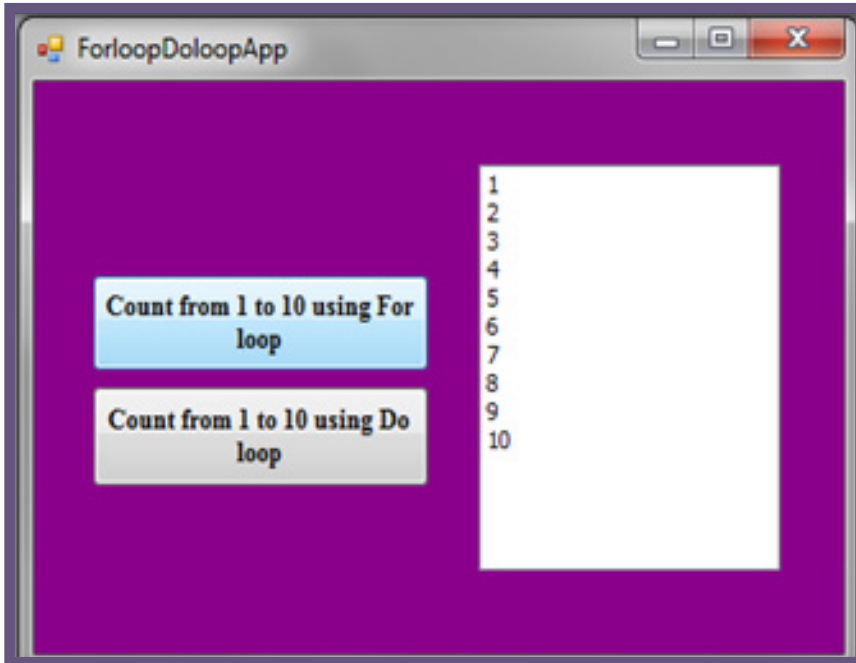
```
End Sub
```

التطبيق الخامس:

مقارنة بين الدالة (For-Next- loop & Do loop)

تحليل التطبيق:

تطبيق مكون من أداتي زر كل واحد منهما يقوم بتنفيذ إجراء أو عملية التكرار للأمر برمجي وهو العد من 1 إلى 10 باستخدام الدالة التكرارية For...Next للتكرار من القيمة الابتدائية (1) وينتهي التكرار عند القيمة النهائية (10) عند تشغيل التطبيق والنقر على الزر الأول. أما بالنسبة للزر الثاني يقوم بتنفيذ حلقة تكرارية وشرط تكرار الحلقة هو عد الأعداد الأصغر من أو يساوي 10 من خلال تنفيذ أمر برمجي باستخدام الدالة Do-While-Loop وإظهار نتيجة الحلقة التكرارية لكل من أداتي الزر في القائمة ListBox1 عند تشغيل البرنامج.



يتم كتابة الكود التالي عند النقر المزدوج على أداتي الزر التاليين:

Count from 1 to 10 using For loop :

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

```
Dim i As Integer
```

```
ListBox1.Items.Clear()
```

”تنفيذ الحلقة التكرارية من القيمة الابتدائية (1) إلى أن تنتهي عند القيمة

```
For i = 1 To 10 (10) النهائية
```

```
ListBox1.Items.Add(i)
```

```
Next i
```

```
End Sub
```

Count from 1 to 10 using Do loop :

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

```
Dim a As Integer
```

```
a = 1
```

```
ListBox1.Items.Clear()
```

```
Do While a <= 10
```

”شروط تكرار الحلقة هي عد الأعداد الأصغر من

أو يساوي (10) وبعدها تتوقف الحلقة التكرارية“

```
ListBox1.Items.Add (a)
```

```
a = a + 1
```

```
Loop
```

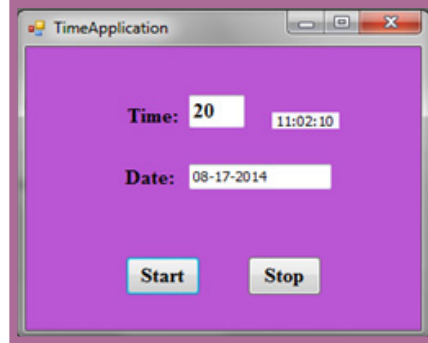
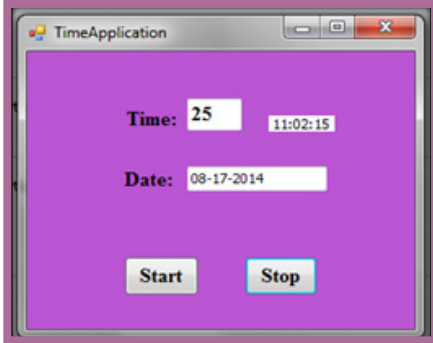
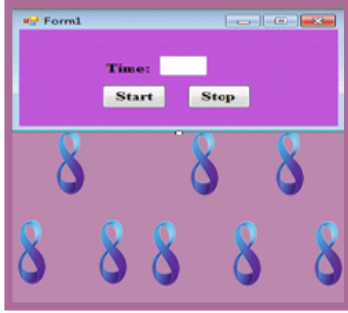
```
End Sub
```


تطبيقات على أدوات تحكم
مختلفة في الفيجوال بيسك

التطبيق الأول:

المؤقت (Timer)

هو إحدى أدوات التحكم الموجودة في Tool-Box في برنامج الفيجوال بيسك وهو الذي يقوم بتنفيذ الإجراء الخاص به وتوقيته كل مدة محددة تحدد بثانية. ومن خلال التطبيق التالي سيتم معرفة كيفية برمجة المؤقت في برنامج الفيجوال بيسك:



(وقت تشغيل المؤقت Timer ووقت إيقافه)

بعد تصميم هذا التطبيق، يتم كتابة الكود التالي عند النقر المزدوج على كلا من `Timer1` وأداتي الزر `Start` و `Stop`:

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

"الكود الذي يكتب في المؤقت هو تحديد بداية الوقت وتزايد كل

ثانية في مربع النص (Time)"

```
Dim a as integer
```

```
a = Val ( )
```

```
TextBox1.Text = Val (TextBox1.Text + 1)
```

```
Label3.Text = TimeString " لإظهار الوقت "
```

```
TextBox2.Text = DateString " لإظهار التاريخ "
```

```
End Sub
```

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

" كود تشغيل و تنفيذ الإجراء الخاص بالمؤقت كل ثانية" ()

```
End Sub
```

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

```
Timer1.Stop ()
```

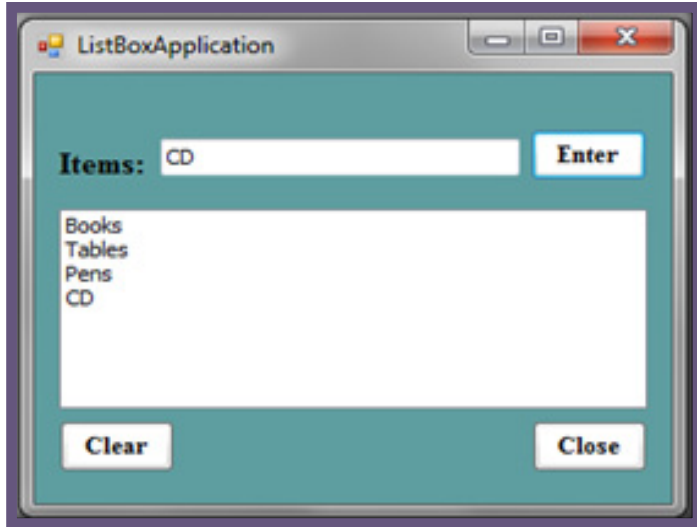
"كود إيقاف تنفيذ الإجراء الخاص بالمؤقت كل ثانية"

```
End Sub
```

التطبيق الثاني:

القائمة (ListBox)

هي أداة من أدوات التحكم في البرنامج وتستخدم للاختيار بين عدة اختيارات و تصلح عندما يكون عدد الاختيارات كبيرا وذلك من خلال سرد أو إدراج أو كتابة مجموعة من العناصر (الاختيارات) بداخلها.



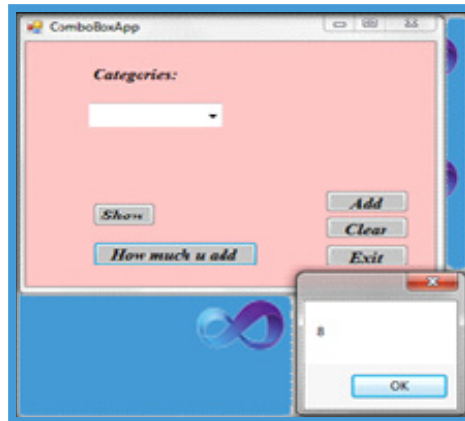
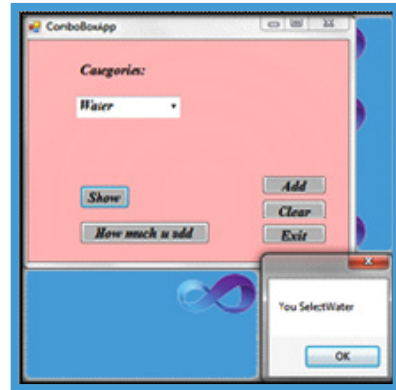
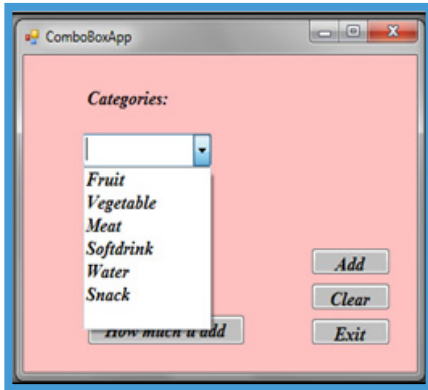
بعد أن يتم تصميم التطبيق أعلاه، يتم كتابة الكود التالي عند النقر المزدوج على الأزرار **Enter** و **Clear** و **Close** :

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    ListBox1.Items.Add(TextBox1.Text) "لإضافة عناصر في القائمة"
    Or ListBox1.Items.Add ("new item")
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    TextBox1.Clear() "لحذف أي عنصر من مربع النص"
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
End
End Sub
```

التطبيق الثالث:

مربع السرد أو التحرير (ComboBox)

مربع السرد أو التحرير **ComboBox**: هو إحدى أدوات التحكم الموجودة في البرنامج وهو الذي يجمع بين القائمة **ListBox** ومربع النص **Textbox** ويحتوي على العديد من الاختيارات أو العناصر بداخله حيث يستخدم للاختيار بين متعدد و يصلح عندما يكون عدد الاختيارات كبيرا.



بعد تصميم التطبيق أعلاه، يتم كتابة الكود التالي عند النقر
المزدوج على الأزرار **Add**، **Clear**، **Show**، **How much u add**،
Exit :

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
ComboBox1.Items.Add("Fruit")
ComboBox1.Items.Add("Vegetable")
ComboBox1.Items.Add("Meat")
ComboBox1.Items.Add("Softdrink")
ComboBox1.Items.Add("Water")
ComboBox1.Items.Add("Snack")
ComboBox1.Items.Add(ComboBox1.Text)
End Sub
```

Clear Button: "لحذف العناصر كلها"

```
Private Sub Button2_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
ComboBox1.Items.Clear ()
End Sub
```

Show Button: "لمعرفة أي عنصر تم اختياره من مربع السرد"

```
Private Sub Button4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button4.Click
MessageBox.Show("You Select" + ComboBox1.Text")
End Sub
```

How much u add Button : "لمعرفة عدد العناصر الموجودة في مربع السرد"

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button5.Click
MessageBox.Show(ComboBox1.Items.Add(ComboBox1.Text))
End Sub
```

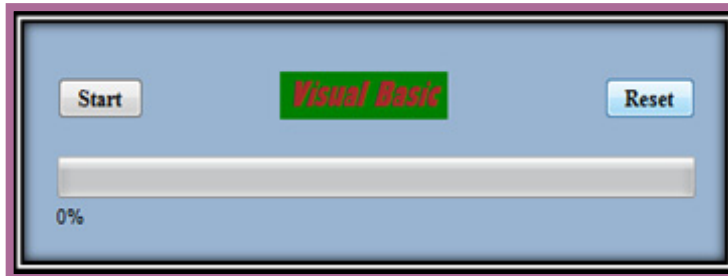
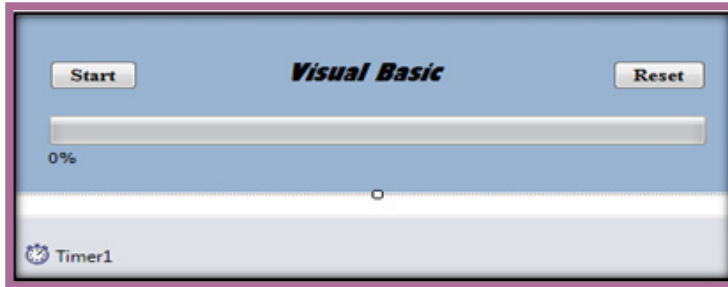
Exit Button: "للخروج من التطبيق"

```
Private Sub Button3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button3.Click
End
End Sub
```

التطبيق الرابع :

توضيح آلية عمل الشريط المتحرك (Progress Bar) عند تحميل أي برنامج

الشريط المتحرك **Progress Bar**: يمثل عنصر تحكم في البرنامج وهو عبارة عن شريط التقدم يتم استخدامه لتوفير ملاحظات مرئية للمستخدمين حول وضع بعض المهام في التطبيق المبرمج بحيث تظهر حركة الشريط الذي يملأ من اليسار إلى اليمين حتى يتم الانتهاء من تلك المهام أو الإجراءات وبالتالي يكتمل الشريط ويتوقف عن الحركة. ومثال ذلك يظهر هذا الشريط في البرامج أثناء عملية تحميل هذه البرامج أو إعادة تحميلها.



تحليل التطبيق:

لبداية حركة **Progress Bar** يتم النقر على الزر **Start** مع ظهور نسبة التحميل المئوية أسفل الشريط وعند إكمال التحميل تظهر كلمة **Hello** في أداة التسمية **Label1** مع تغير لون خلفية أداة التسمية ولون الخط في نفس الوقت. وعند إعادة التحميل من خلال الشريط **Progress Bar** يتم النقر على الزر **Reset** وترجع كلمة **Visual Basic** بظهور في **Label1** مع تغير لون خلفية أداة التسمية ولون الخط وتصبح نسبة التحميل **0%**.

ملاحظة: يضاف المؤقت **Timer** في هذا البرنامج حيث يتم تحديد مدة زمنية معينة مع الشريط **Progress Bar** أثناء عملية التحميل حتى اكتمال المهمة أو الإجراء.

يتم كتابة الكود التالي عند النقر المزدوج على المؤقت **Timer1** وأداتي الزر **Start** و **Reset**:

Start Button :

```
Private Sub Button1_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button1.Click  
()Timer1.Start "تشغيل المؤقت"  
End Sub
```

Reset Button :

```
Private Sub Button2_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button2.Click  
Timer1.Stop () "إيقاف المؤقت"  
ProgressBar1.Value = 0 "انتهاء التحميل في الشريط"  
"إظهار هذه الكلمة عند إعادة التحميل"  
Label1.Text = "Visual Basic"  
Label1.BackColor = Color.Green "تغيير لون خلفية الكلمة"  
Label1.ForeColor = Color.Brown "تغيير لون الخط"  
Label2.Text = "0%" "النسبة المئوية عند الإعادة"  
End Sub
```

ويكتب الأمر البرمجي التالي عند النقر المزدوج على المؤقت
:Timer

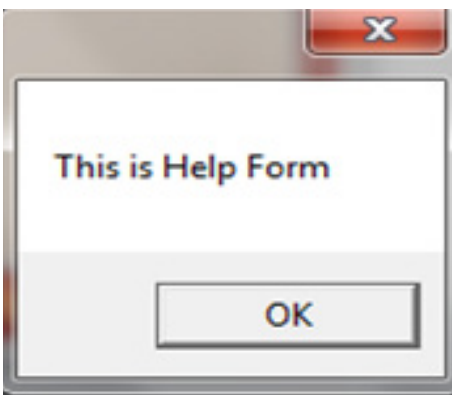
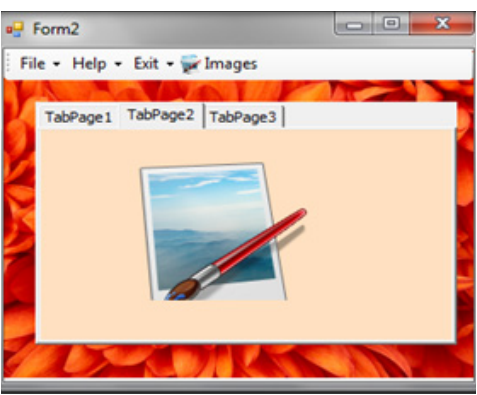
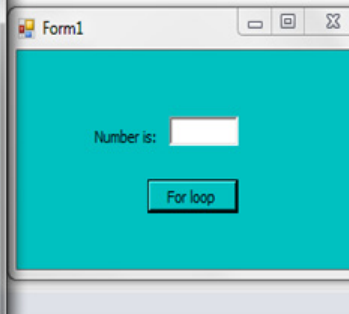
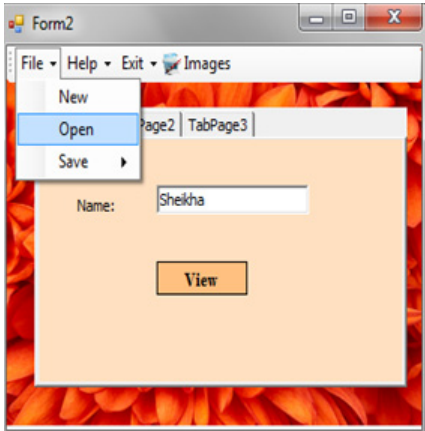
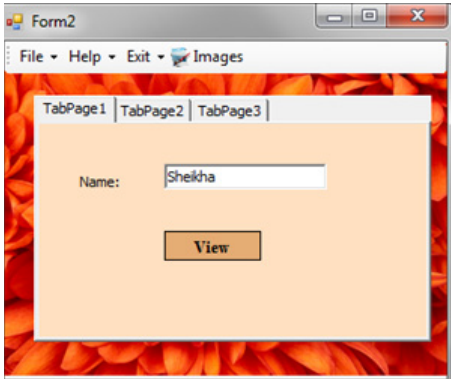
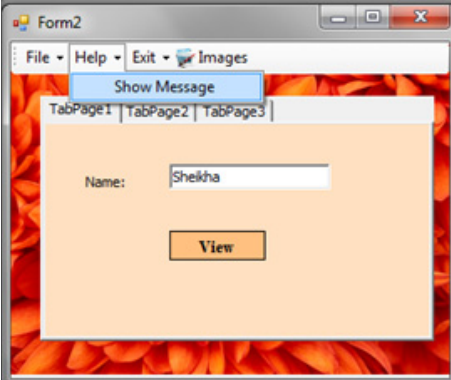
```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    ProgressBar1.Increment1 "حركة الشريط وتزايد بمقدار نقطة"
    If ProgressBar1.Value = 100 "عند اكتمال حركة الشريط أو اكتمال التحميل"
    Then
        Label1.Text = "Hello " "تظهر في أداة التسمية كلمة أخرى وهي Hello"
        Label1.BackColor = Color.Gold "تغيير لون خلفية الكلمة"
        Label1.ForeColor = Color.Gray "تغيير لون الخط"
    End If
    Label2.Text = ProgressBar1. "تزايد النسبة المئوية أثناء عملية التحميل"
    Value & ("%")
End Sub
```

التطبيق الخامس :

شريط القوائم Tool Strip و أداة التنقل بين الصفحات Tab Control

شريط القوائم Tool Strip: هو شريط المهام الذي يوجد في أعلى صفحة أي برنامج أو تطبيق وهو عنصر تحكم في برنامج الفيجوال بيسك ويتضمن بداخله مجموعة من الخانات التي تعتبر بمثابة خيارات للبرنامج حيث أن من خلالها يمكن حفظ ملف أو اختيار صورة أو حذف معلومات أو الانتقال إلى صفحات أخرى من التطبيق أو الخروج من التطبيق وغيرها من الخيارات من خلال تحديد خيار لكل خانة وكتابة أمر برمجي متعلق بها.

أداة التنقل بين صفحات البرنامج Tab Control: عبارة عن عنصر تحكم في برنامج الفيجوال بيسك ويستخدم في شاشات (Forms) أي تطبيق للانتقال من صفحة إلى أخرى في نفس الشاشة مثلا ويمكن أن يتضمن أدوات تحكم أخرى في البرنامج مع تحديد مهام أو إجراءات خاصة بها من خلال الأوامر البرمجية ولها فائدة بأنها تقلل أو تقلص حجم التطبيق المصمم أو المبرمج لعدم استهلاك الكثير من الشاشات (Forms) في نفس التطبيق.



يتم كتابة الكود التالي عند النقر المزدوج على كل من أدوات التحكم التالية لإظهار إجراءات التطبيق عند تشغيله :

OpenToolStripMenuItem (File-Open:

```
Private Sub Open Tool Strip Menu Item_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Open Tool Strip MenuItem.Click
```

```
Form1.Show () "لإظهار شاشة أخرى"
```

```
End Sub
```

CloseProgramToolStripMenuItem (Exit-Close Program) :

```
Private Sub Close Program Tool Strip Menu Item_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Close Program Tool Strip MenuItem.Click
```

```
End "لإغلاق التطبيق"
```

```
End Sub
```

Enter Button :

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

```
TextBox1.Text = "Sheikha" "للإظهار هذا الاسم عند النقر على الزر"
```

```
End Sub
```

ShowMessageToolStripMenuItem (Help-Show Message) :

```
Private Sub Show Message Tool Strip Menu Item_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ShowMessage Tool Strip MenuItem.Click
```

```
MessageBox.Show("This is Help Form ")
```

"إظهار رسالة في التطبيق عند النقر على الخانة (Help) ثم (Show Message)

```
End Sub
```

Tool Strip Label4 (Images) :

```
Private Sub ToolStripLabel4_Click(ByVal sender As System.Object,
```

```
ByVal e As System.EventArgs) Handles Tool Strip Label4.Click
```

```
PictureBox1.Image = My.Resources.Picture1
```

"إظهار صورة في الصفحة الثانية لأداة التنقل بين الصفحات من خلال النقر على

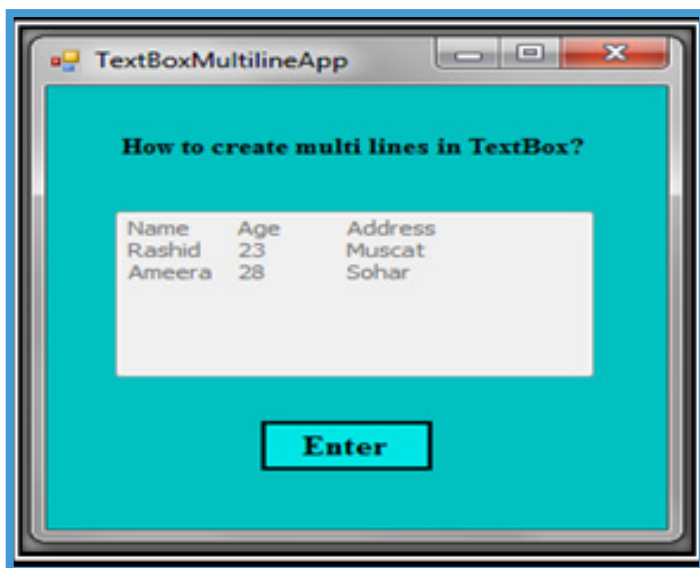
خانة (Images)"

```
End Sub
```

التطبيق السادس :

استخدام وكتابة Multi Lines كأمر

برمجي في مربع النص TextBox



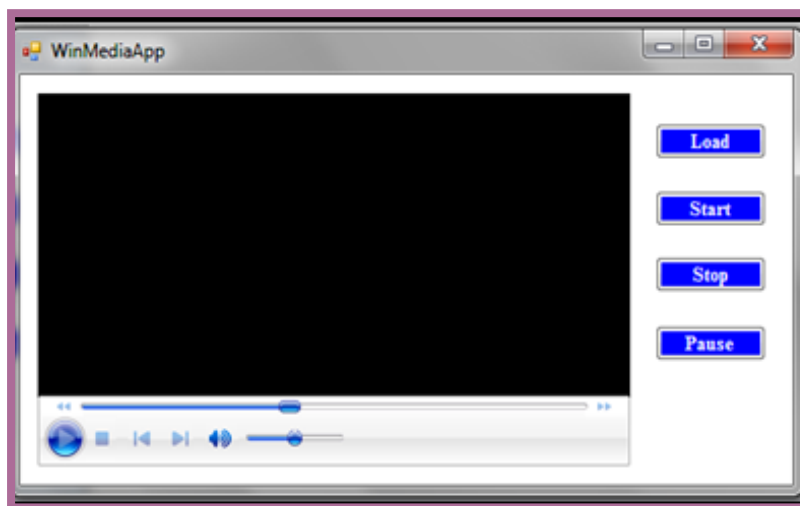
لتنفيذ التطبيق السابق ، يتم كتابة الكود التالي عند النقر المزدوج على الزر Enter :

```
TextBox1.Text = ("Name" & vbTab & "Age" & vbTab & "Address")  
TextBox1.Text &= (vbNewLine & "Rashid" & vbTab & "23" & vbTab &  
"Muscat")  
TextBox1.Text &= (vbNewLine & "Ameera" & vbTab & "28" & vbTab  
& "Sohar")
```

ملاحظة: تستخدم vbTab للفصل بمسافة بين الكلمة الأولى والثانية في السطر (Line) وتستخدم vbNewLine للانتقال إلى السطر الثاني (Line 2) في مربع النص نفسه TextBox.

التطبيق السابع :

تطبيق برنامج Windows Media Player
في برنامج الفيجوال بيسك



بعد تصميم التطبيق السابق ، يتم كتابة الأوامر البرمجية في كل من الأزرار التالية:

Load Button :

"يستخدم هذا الأمر البرمجي في إدراج ملفات الفيديو أو الصوت الموجودة في الجهاز الخاص بك مثلاً"

```
If (OpenFileDialog1.ShowDialog = DialogResult.OK) Then  
AxWindowsMediaPlayer1.URL = OpenFileDialog1.FileName  
"هذا الكود يتبع رابط الفيديو الذي تم استدعائه"
```

Start or Play Button :

يستخدم هذا الكود لتشغيل الفيديو "AxWindowsMediaPlayer1.Ctl-
controls.play()

Stop Button :

AxWindowsMediaPlayer1.Ctl-
controls.stop()

Pause Button :

"إيقاف مؤقت للفيديو "AxWindowsMediaPlayer1.Ctlcontrols.
pause()



من الصعب هزيمة شخص
لم يهزمه اليأس من داخله.

الملحقات

111

التحديات

التحدي الأول

برنامج لإظهار البيانات في شاشة المستخدم
الرسومية (Form):

- أدوات التحكم (Tool Book) المستخدمة للشاشة (Form) :
- 5 أداة تسمية (Label)
 - 4 مربع نص (Text Book)
 - 2 أداة زر (Button)



الخصائص (Properties) :

اختيار خلفية للشاشة Form :
Background Image

الخط المستخدم Font :
Time New Roman

شكل الخط Font Style :
Bold

الخط Size :
12

لون أداة التسمية Student Details :
White

لون خلفية أدواتي الزر Button و أدوات التسمية Label :
Silver

مراجعة ما يلي عند كتابة الكود :

عند تشغيل البرنامج والنقر على زر View يجب إظهار البيانات في مربعات النص
TextBox بالإضافة إلى إظهار Student Details في أداة التسمية Label1.
إعطاء أمر لأداة الزر Exit بإغلاق البرنامج.

التحدي الثاني

برنامج لحساب ناتج أي عملية حسابية (+, -, *, /):

يتكون هذا البرنامج من أدوات التحكم التالية: أداة تسمية label(3) ، مربع نص textbox(3) ، أداة زر button(2) .

ككرة البرنامج: يستخدم مربعي نص 1 textbox و text- box 2 لإدخال رقمين مع كتابة ال "Val" Function في صفحة الكود لإظهار ناتج العملية الحسابية في مربع النص الثالث 3 textbox الذي يختص بخانة "الناتج". واستخدام أداة الزر Calculate لإظهار الناتج وأداة الزر Close للإغلاق صفحة البرنامج.

A screenshot of a Windows-style application window titled "Calculations". The window has a standard title bar with minimize, maximize, and close buttons. The main content area contains three text input fields. The first is labeled "First Number:" and contains the value "6". The second is labeled "Second Number:" and contains the value "2". The third is labeled "Result:" and contains the value "8". Below the input fields are two buttons: "Calculate" and "Exit".

First Number:	6
Second Number:	2
Result:	8

Buttons: Calculate, Exit

A screenshot of a Windows-style application window titled "Calculations". The window has a standard title bar with minimize, maximize, and close buttons. The main content area contains three text input fields. The first is labeled "First Number:" and contains the value "1". The second is labeled "Second Number:" and contains the value "2". The third is labeled "Result:" and contains the value "-1". Below the input fields are two buttons: "Calculate" and "Exit".

First Number:	1
Second Number:	2
Result:	-1

Buttons: Calculate, Exit

التحدي الثالث

برمجة الآلة الحاسبة Calculator Form :

تصميم الشاشتين Calculator_Form1 و Calculator_Form2 مع كتابة الكود الخاص بكل منهما و مراعاة النقاط التالية :

في الشاشة Calculator_Form1 : عند إدخال ثلاثة متغيرات أو أرقام في TextBox1,2,3 والنقر على كل من العمليات الحسابية الموجودة في Group Box (Math Functions) يجب إظهار النتيجة Result في مربع النص TextBox4 عند النقر على الزر Find. ويستخدم الزر Exit للخروج من البرنامج.

Calculator_Form1

First Number: 2

Second Number: 2

Third Number: 2

Result: 6

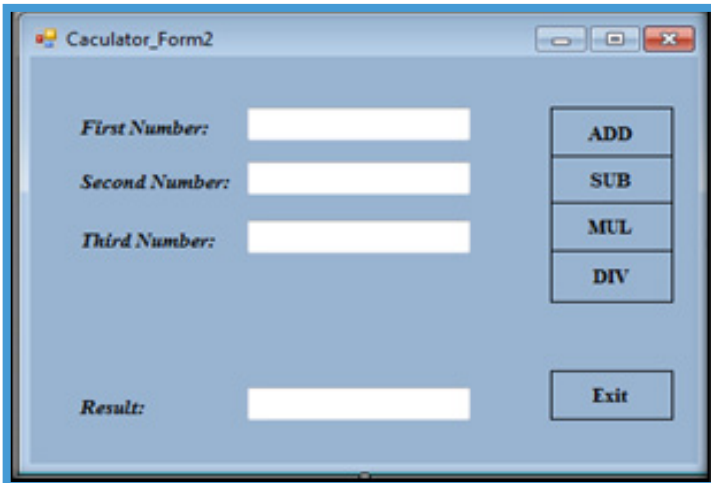
Math Functions:

ADD SUB MUL DIV

Find Exit

في الشاشة Calculator_Form2 :

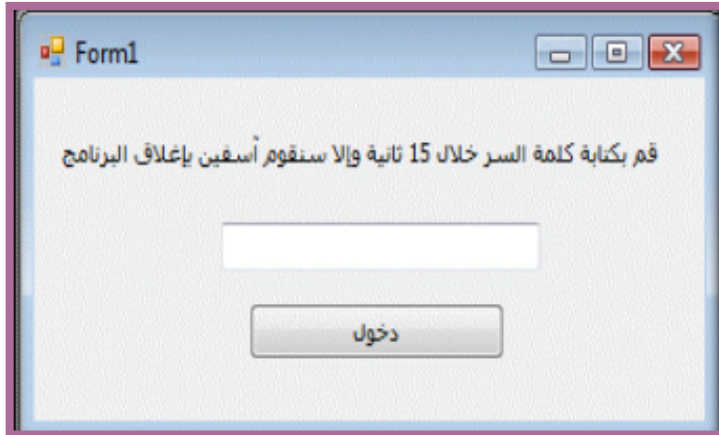
عند إدخال ثلاثة أرقام في TextBox1,2,3 والنقر على كل من العمليات الحسابية الموضحة من خلال استخدام الزر (Add, Sub, Mul & Div Button) يجب إظهار النتيجة Result في مربع النص TextBox4 ويستخدم الزر Exit للخروج من البرنامج.



التحدي الرابع

برمجة المؤقت لتحديد مدة زمنية معينة لإدخال كلمة السر:

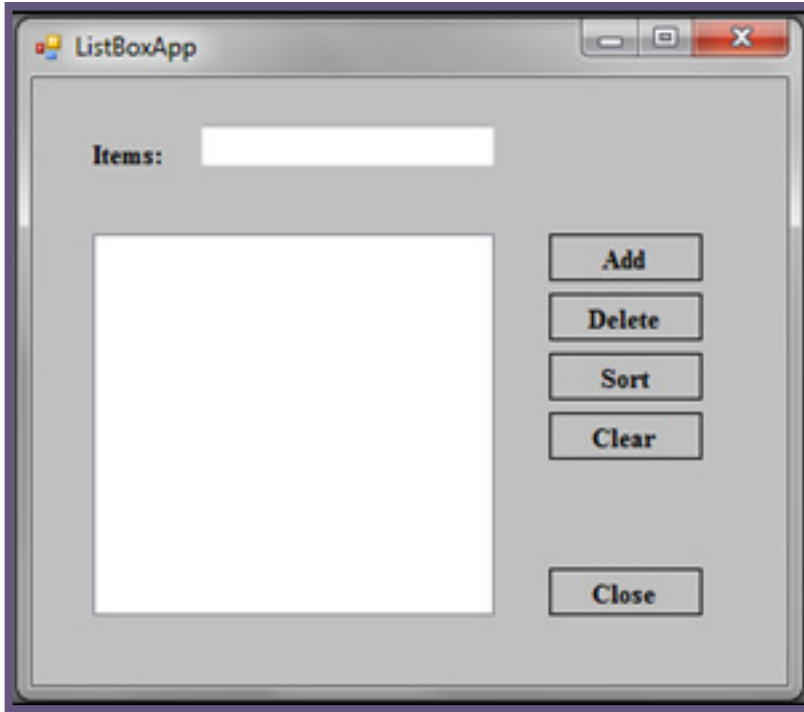
يمكننا استخدام المؤقت **Timer** لكل ما له علاقة بالوقت أو التوقيت، فيمكننا تحديد مدة زمنية معينة لإدخال كلمة السر عند تشغيل البرنامج. قم بتنفيذ برنامج يقوم بطلب كلمة سر ولا بد على المستخدم أن يقوم بإدخال كلمة السر الصحيحة خلال 10 ثواني. فمثلا تكن كلمة السر الصحيحة 123456، وإذا تأخر المستخدم عن إدخال كلمة السر خلال 10 ثواني فسيغلق البرنامج.



The screenshot shows a Windows application window titled "Form1". Inside the window, there is a text label in Arabic that reads "قم بكتابة كلمة السر خلال 15 ثانية وإلا سنقوم أسفين بإغلاق البرنامج". Below the text is a single-line text input field. At the bottom center of the form is a button labeled "دخول" (Login).

التحدي الخامس

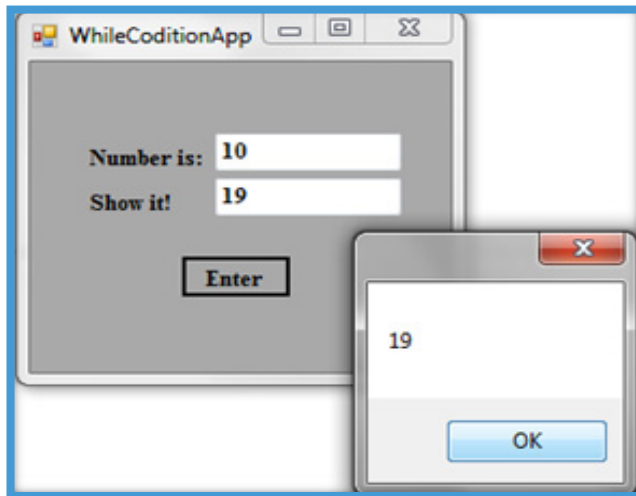
تصميم الفورم ListBoxApp مع كتابة إجراء أو كود لكل من الأزرار الموجودة:



التحدي السادس

تصميم الفورم WhileConditionApp كما هو موضح بالشكل التالي :

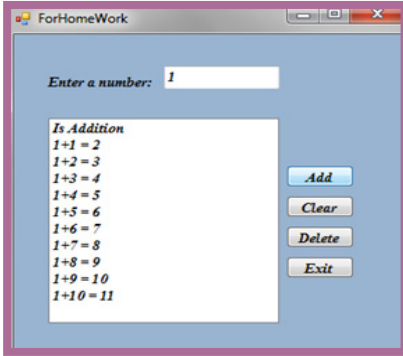
وتنفيذ الدالة التكرارية Repetition Function باستخدام While-End While التي توضح تنفيذ برنامج يقوم بإظهار رسالة 10 مرات بالأعداد المحصورة بين 10 و 20 مع إظهار كل عدد في مربع النص TextBox2 في نفس الوقت. وتكون القيمة الأولى للمتغير (a) تساوي 10 وشرط تكرار الحلقة هو أن تكون قيمة (a) أصغر من 20 .



التحدي السابع

برنامج باستخدام Loop For ...Next :

تصميم برنامج يقوم بتنفيذ عملية حسابية (+, -, *, /) لعددتين مع استخدام For ...Next وتنفيذ النقاط التالية عند كتابة الكود في الزر Add :



تبدأ الحلقة التكرارية بالعدد (1) وهو أول عدد من أعداد التكرار.

وتنتهي الحلقة التكرارية بالعدد (10) وهو الذي يتوقف عنده التكرار. وتمثل الأعداد من 1 إلى 10 (العدد الثاني في العملية الحسابية).

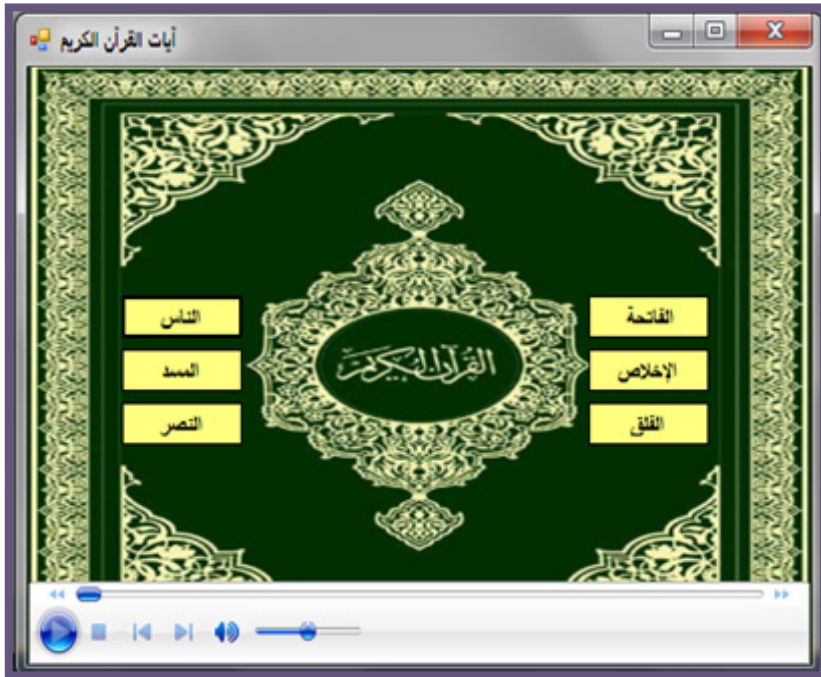
عند إدخال أي عدد صحيح في مربع النص TextBox1 (وهو العدد الأول في العملية الحسابية) والنقر على أداة الزر Add ستظهر العملية الحسابية للعددتين وحاصل جمعهما في القائمة ListBox1.

تنفيذ الإجراء المختص بكل من الأزرار التالية: Clear, Delete, Exit.

التحدي الثامن

برنامج الاستماع لسور القرآن الكريم :

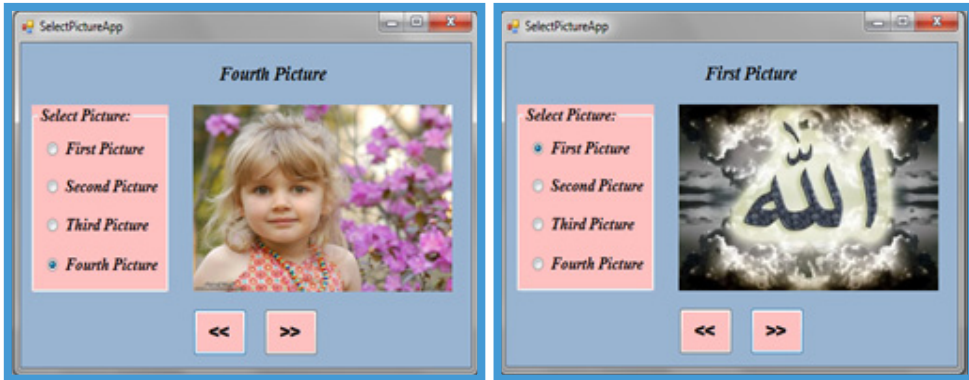
يتم برمجته من خلال استخدام Windows Media Player في برنامج الفيچوال بيسك وكتابة الأوامر البرمجية المتعلقة به.



التحدي التاسع

تصميم الفورم `SelectPictureApp` التي تتضمن أدوات التحكم التالية:
أداة تسمية (Label)
أداة زر (Button)
صندوق الصورة (Picture Box)
إطار المجموعة (Group Box)
أداة الاختيار (Radio Button)

ثم كتابة برنامج يتضمن النقاط التالية :



كتابة أوامر برمجية لكل أداة اختيار في البرنامج لإظهار صورة عند الضغط على كل من أدوات الاختيار مع إظهار اسم كل صورة في أداة التسمية Label1 عند تشغيل البرنامج.

تعريف متغير عام Public Variable من نوع بيانات (Integer) لتحديد رقم لكل صورة متعلقة بأداة الاختيار المحددة.

في أداة الزر (<<) يتم كتابة أمر برمجي لإظهار كل صورة واحدة تلو الأخرى من الأعلى top إلى الأسفل down من خلال تنفيذ الدالة الشرطية (If Statement) وتحديد حالة لكل صورة في صندوق الصورة Picture Box1 عند الضغط على أدوات الاختيار الأربعة من خلال تنفيذ الدالة (Select Case) لتغيير إظهار الصور عند تشغيل البرنامج من خلال الضغط على أداة الاختيار المتعلقة بكل صورة.

في أداة الزر (>>) يتم كتابة أمر برمجي لإظهار كل صورة واحدة تلو الأخرى من الأسفل down إلى الأعلى top من خلال استخدام نفس الدوال Functions التي سيتم تنفيذها في أداة الزر (<<).

مبدع

متألق



لقد وصلت للنهاية
بإمتهار وفكر الله

الخاتمة

وفي الختام نتمنى أن يكون هذا الكتاب قد نال إعجابكم واستحسانكم. ونأمل أن نكون من خلاله قد تمكنا من إثارة شغلة إهتمامك **بالفيجوال بيسك** ليكون بداية الطريق لك نحو التعمق في هذا البرنامج الجميل ذو التطبيقات الكثيرة والمتعددة . وحال هذه اللغة من حال سائر لغات البرمجة ، تقوى بالممارسة والتجربة المستمرة وتندثر وتنسى في حالة ركنها على الرف. ف **البرمجة ليست مجموعة من المعلومات المسلية وإنما أداة للتطوير الخلاق**. نفع الله بكم ووجهكم لما فيه الخير.

المصادر

الكتب (Books) :

- 1 David I. Schneider, An Introduction To Programming Using Visual Basic 2005 . (2006). Publishing by: Person Education Inc, Upper Saddle River, NJ07458. Sixth Edition
- 2 P.J.Deitel & H.M.Deitel, Visual Basic 2008 (How to Program). (2009). Publishing by: Person Education Inc, Upper Saddle River, New Jersey 07458. Ninth Edition

مواقع الانترنت (Internet) :

1 <http://www.abahe.co.uk>

2 <http://www.kutub.info>

الكتب الإلكترونية / المدونات البرمجية :

اسم الكتاب: فيجوال بيسك 2008 خطوة خطوة

المؤلف: Michael Halvorson

صفحة الكتاب من: <http://www.amazon.com/Microsoft-Visual-Basic-Step/dp/0735625497>

ترجمة: مروان المفلحي ، المدونة البرمجية: <http://marwanvb.blogspot.com>

الموقع: <http://Vb4arab.com/vb>

اسم الكتاب: Microsoft Visual Basic 2010 Express

مدونة الجوهريزم: <http://algharism.blogspot.com>

اسم الكتاب: Learn The Basics In Visual Basic 6.0

تأليف وإعداد: دريد عبدالله

تحميل الكتاب من موقع: www.Kutub.info

كتب ونشر بواسطة: Prelware Association

موقع:

فيديوهات تعليمية لبعض تطبيقات برنامج الفيجوال بيسك عن طريق

. موقع :

فيديو [/http://algharism.blogspot.com/search/label](http://algharism.blogspot.com/search/label)

عن القرية

نحن في القرية الهندسية نحلم بجيل من الأطفال والشباب يجد متعته في العلم وإكتساب المعرفة. **نحلم** بمستقبل مشرق ينافس فيه أبناءنا عباقرة العالم ويتفوقون عليهم بجد وشغف ولهذا نعمل جاهدين على تقديم العلوم بطريقة مسلية تنافس البدائل المتوفرة من ألعاب الفيديو ومشاهدة التلفاز. العالم من حولنا يتطور بسرعة رهيبه في مختلف مجالات التكنولوجيا الرقمية والإلكترونية ومن واجبنا أن نسعى لمواكبة التطور والحدائق ليس بالقييل والقال وإنما بالعلم والعمل. **ومعا** يمكننا أن نصنع التغيير !

فهد السيابي
الرئيس التنفيذي

نرجو أن نكون قد استطعنا افادتكم وامناعكم في عالم برمجة الفيچوال بيسك ، ولمزيد من المعلومات وللتحقق من إجاباتكم زوروا موقعنا على الإنترنت www.ev-center.com كما يسعدنا قراءة آرائكم وتعليقاتكم في حساباتنا على مواقع التواصل الإجتماعي عبر الوسم #مبرمجو_المستقبل

 @EngineeringOman

 Engineering Village

 @EngineeringVillage

 EngineeringVillageEV

مؤلفاتنا الأخرى



احترف الأوتوكاد
في 10 أيام



احترف الأردوينو
في 10 أيام



احترف سكراتش
في 10 أيام



احترف الرازبيري باي
في 10 أيام

* كل شخص عليه أن يتعلم برمجة الحاسوب .. لأنه يعلمك كيف تفكر ،، ” ستيف جوبز“

* بيل جيتس | مؤسس ميكروسفت | تعلم البرمجة بعمر 13 عام

* مارك زوكربيرج | مؤسس الفيس بوك | بدأ باستخدام الحاسوب في الصف السادس

* جاك زورسي | مؤسس شبكة تويتر | أهداه والداه حاسوب وهو بعمر 9 سنوات

لم يكونوا يملكون العبقرية أو الذكاء الخارق وإنما بحثوا عن المرح والمتعة في أماكن الابتكار .

البرمجة هي لغة العصر لخدمة الإنسانية ومساعدة الناس ، هي طريقة التفكير والتنوير لعالم اليوم ويجب أن تبدأ بها منذ صغرك ! سواء أردت أن تغير العالم أو أن تملك الكثير من المال .

في العشر السنوات القادمة سيحتاج العالم إلى 10 ملايين مبرمج لخدمة البشرية ، ولا يوجد في العالم سوى 400 ألف مبرمج حاليا . العالم يحتاجنا ، فهيا بنا نبرمج .



لنمرح بتعلم مهارات البرمجة مع الفيچول بيسك

#مبرمجو_المستقبل