

الحمد لله رب العالمين والصلاة والسلام على أشرف المرسلين سيدنا محمد صلى الله عليه وسلم

تعتبر لغة الفيجول بيسيك دوت نت من أكثر اللغات كفاءة وشهرة فى عالم البرمجة

فلغة الفيجول بيسيك دوت نت أحد لغات إصدارة الدوت نت الجديدة لمايكروسوفت

فهذه اللغة تتمتع بمميزات عديدة وتسهيلات كثيرة فهي تختلف إختلافا ملحوظا عن

فيجول بيسيك ٦ ليس فى كل شىء بل فى العديد منها

ولذلك حرصت على وضع بعض الدروس البسيطة إلى حد ما مما أعرفه عن اللغة

ومما وصلت إليه ولذلك أتمنى أن كل متصفح ينال أكبر معرفة والإستفادة من هذه

الدروس والله ولى التوفيق ولكن أريد من كل قارئ من التأكد من المعلومة

بالإختبار والسؤال عنها للوصول إلى خير النتائج إن شاء الله

وأتمنى منكم خير الدعاء



الاسم : محمدى عبد الغال محمد

بكالوريوس حاسبات ومعلومات قسم نظم المعلومات جامعة المنصورة – مصر (دفعة ٢٠٠٦)

هذه الدروس الحقيقة عملتها وأنا فى دخولى ثالثة كلية فى حين معلوماتى البسيطة وهذا من فضل

ربى وحاليا جمعتها فى كتاب لتكون عوننا منجزا وبسيطا جدا جدا

لكل مبتدأ يمكنه قراءة هذا الكتاب مع معرفته بخلفية برمجية ولو ضعيفة جدا .

وقد طرحت هذه الدروس من قبل على موقعى المتواضع من مدة طويلة جدا وهذا الموقع عملته

بهتمل لتمرين يدى عليه وكل هذا من فضل ربي .

والحمد لله أعرف مواضع كثيرة من أهمها سى شارب وأعرف التعامل بقواعد البيانات معها والكثير والكثير ولكنى قلت قواعد البيانات لأهميته عند البعض ولكنى ضعيف فى بعض النقاط .
الموقع هو بسيط ولكن يمكن تطلعه عليه وقرأة مقالاته وهو :

www.powerearth.4t.com

powerearth4t@Yahoo.com

هذه الدروس للمبتدئين تذاكر فى (٥) أيام وللمحترفين تذاكر فى (٣) أيام .

ممكن أعمل دروس ملمة ومنجزة وبسيطة وكثيرة فى السى شارب ولكن لو فى تشجيع .
ولو أى أخ عربى أراد أى شىء فى أى حاجة هو محتاجها ممكن أساعده بمعلوماتى المتواضعة أو أوفر له المصدر المتاح لحاجته .

وعلى فكرة أنا من النوع اللى بينجز فى أى حاجة ممكن أذاكرها وهذا واضح فى الدروس .

نسأل الله أن يتقبل منا أعمالنا ويوفقنا إلى ما يحب ويرضى .

أى حاجة تقف أمامك "صلى على النبى" وضع هذه الجملة فى رأسك فى كل حياتك

"ومن يتق الله يجعل له مخرجا ويرزقه من حيث لا يحتسب"

"لا يؤمن أحدكم حتى يحب لأخيه ما يحب لنفسه"

الآية والحديث أمشى عليهم مدى حياتك وستفتح لك أبواب الخير وذلك من خلال تجربتى .

وأعرف مافيش حاجة صعبة حتى لو لقيت حد قوى فى شىء ما العملية ليست صعبة والمهم أنك

تعرف وتقرأ هو مش أحسن منك ولكن أجتهد والله ولى التوفيق .

أدعوا لأخوانكم فى (أفغانستان-الشيشان-العراق-فلسطين-البوسنة)

أدعوا على أعداء الإسلام (أمريكا-بريطانيا-روسيا الملعونة-إسرائيل)

اللهم أهدى ولاة أمورنا إلى ماتحب وترضى .

اللهم صلى وسلم على سيدنا محمد

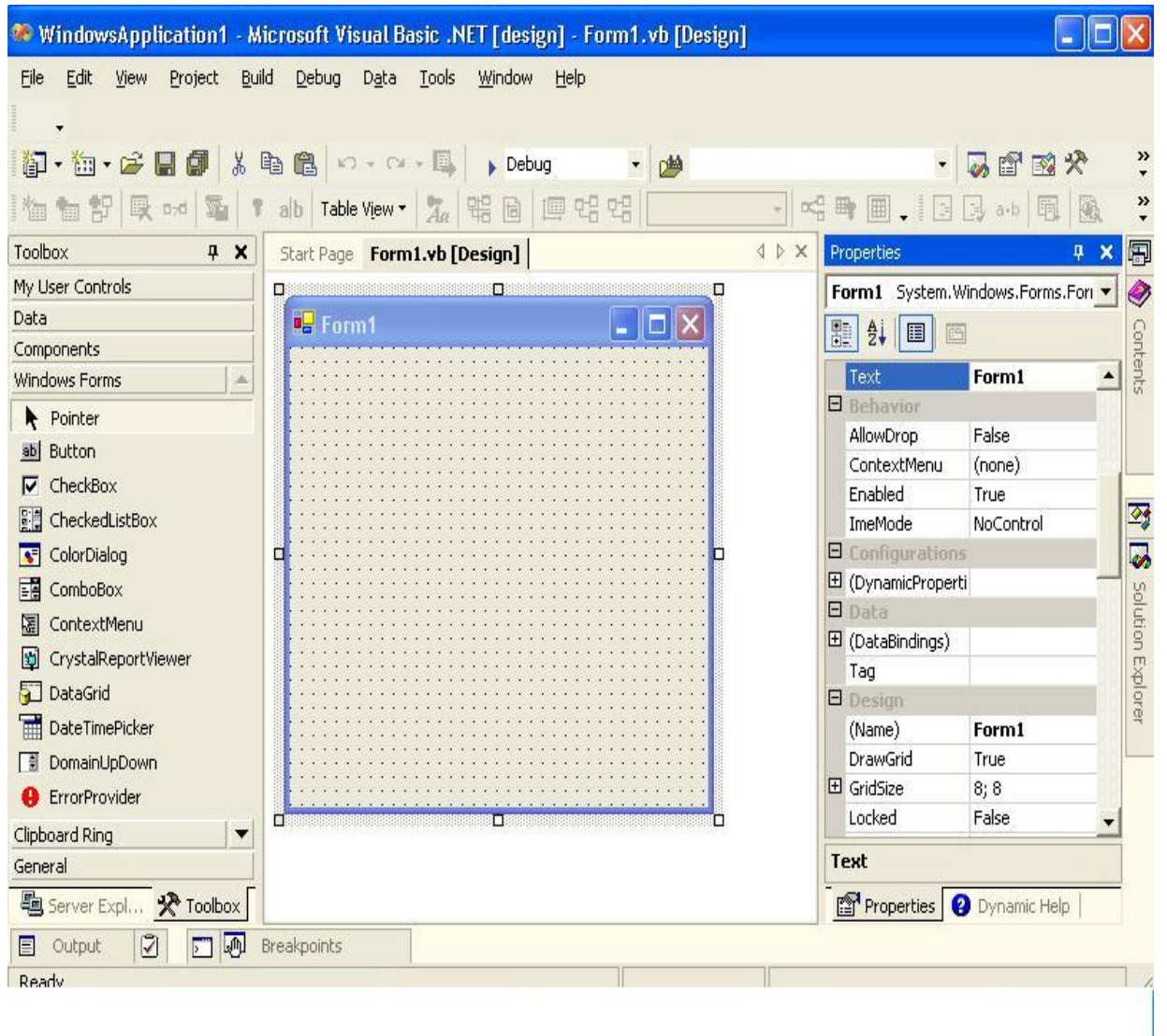
واجهة البرنامج IDE

أولا لابد من التعرف على البيئة المتعامل معها وهي واجهة البرنامج وتسمى

IDE(Integrated Development Environment)

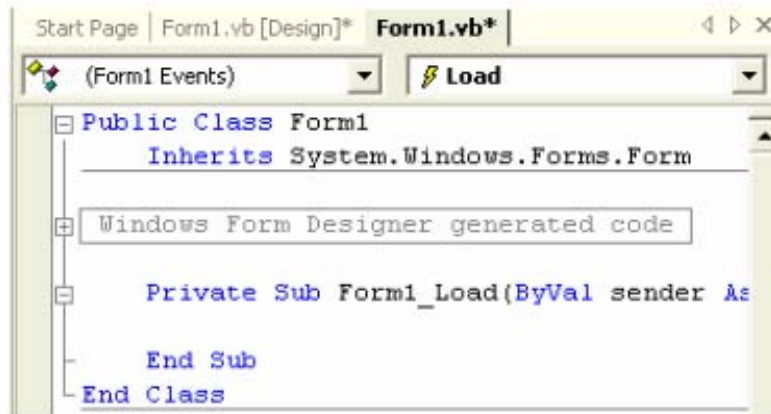
وهي تشمل الأدوات المستخدمة في عملية البرمجة وخصائص هذه الأدوات والمكان الازم لكتابة الكود

Solution وهي النافذة التي تحتوى على مكونات البرنامج مثل الفورم وغيرها في العمل الحالى والقوائم و**Explorer**



مكان كتابة الكود وهو بالضغط مرتين على الأداة المستخدمة

وبالضغط مرتين على الفورم Double Click



ويوجد قائمة منسدلة لوضع الحدث الازم للإداة مثل عند الضغط مرتين أو الضغط مرة أو عند تحريك الماوس وغيرها

وتسمى Events



أما هذه القائمة المنسدلة فهي تحتوى على الأدوات الموجودة على الفورم ويمكن تحديدها لكتابة الكود



أما هذا الجزء فهو لإختيار مكان الكود أو سطح التصميم

فهذا الجزء يدل على إختيار مكان التصميم *Form1.vb[Design]

أما هذا الجزء يدل على إختيار مكان الكود *Form1.vb



ولعمل إختبار للبرنامج إضغط مفتاح F5

أو إضغط على مفتاح Run من Toolbar

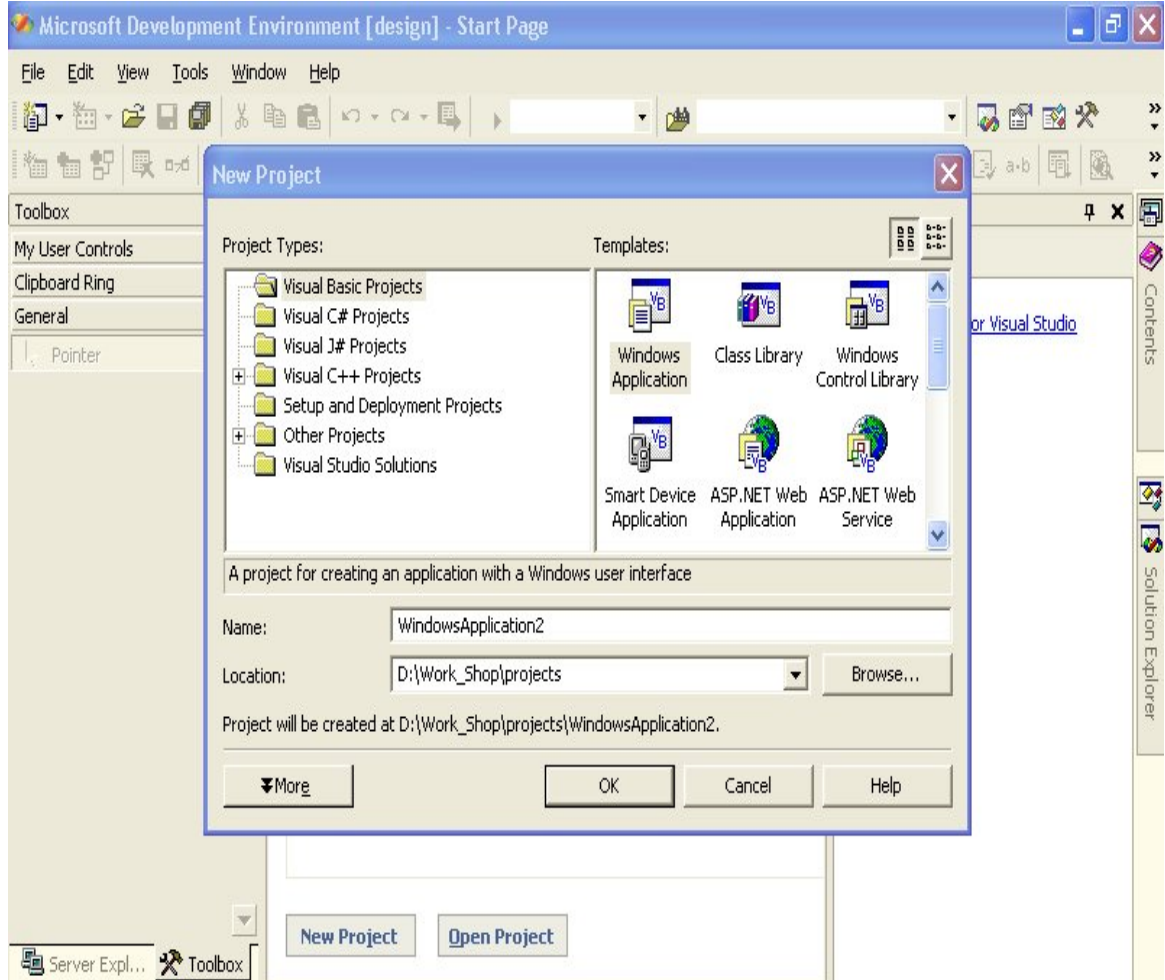


ويوجد قائمة تسمى List Members

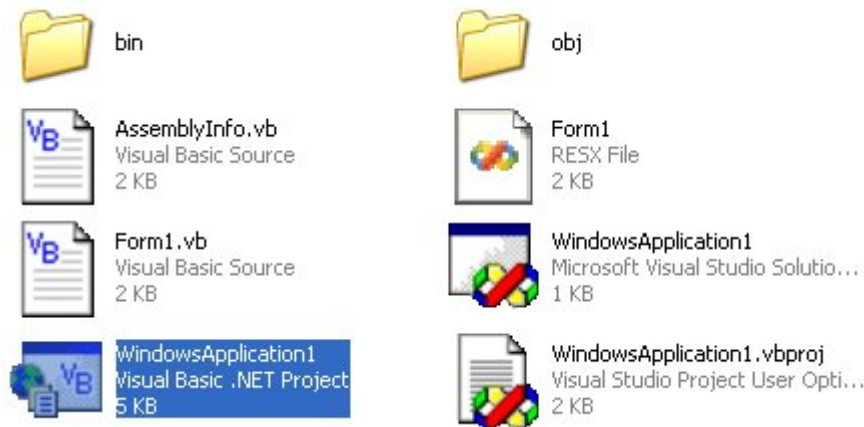
وهي تحتوى على مجموعة من Methods&Properties&Events&Argument List

مثل TextBox1. فبعد هذه النقطة تظهر List تحتوى على المجموعة السابقة

أما من ناحية الحفظ ففي أول فتح للبرنامج يظهر مربع لإختيار اللغة وإسمها ومسار الحفظ



وإذا أردت التعديل في البرنامج بعد الحفظ نذهب إلى مكان الحفظ ونضغط على هذه الأيقونة المظلمة



ولرؤية التطبيق التنفيذي exe نفتح فولدر bin

تطبيقات إبتدائية Console Applications

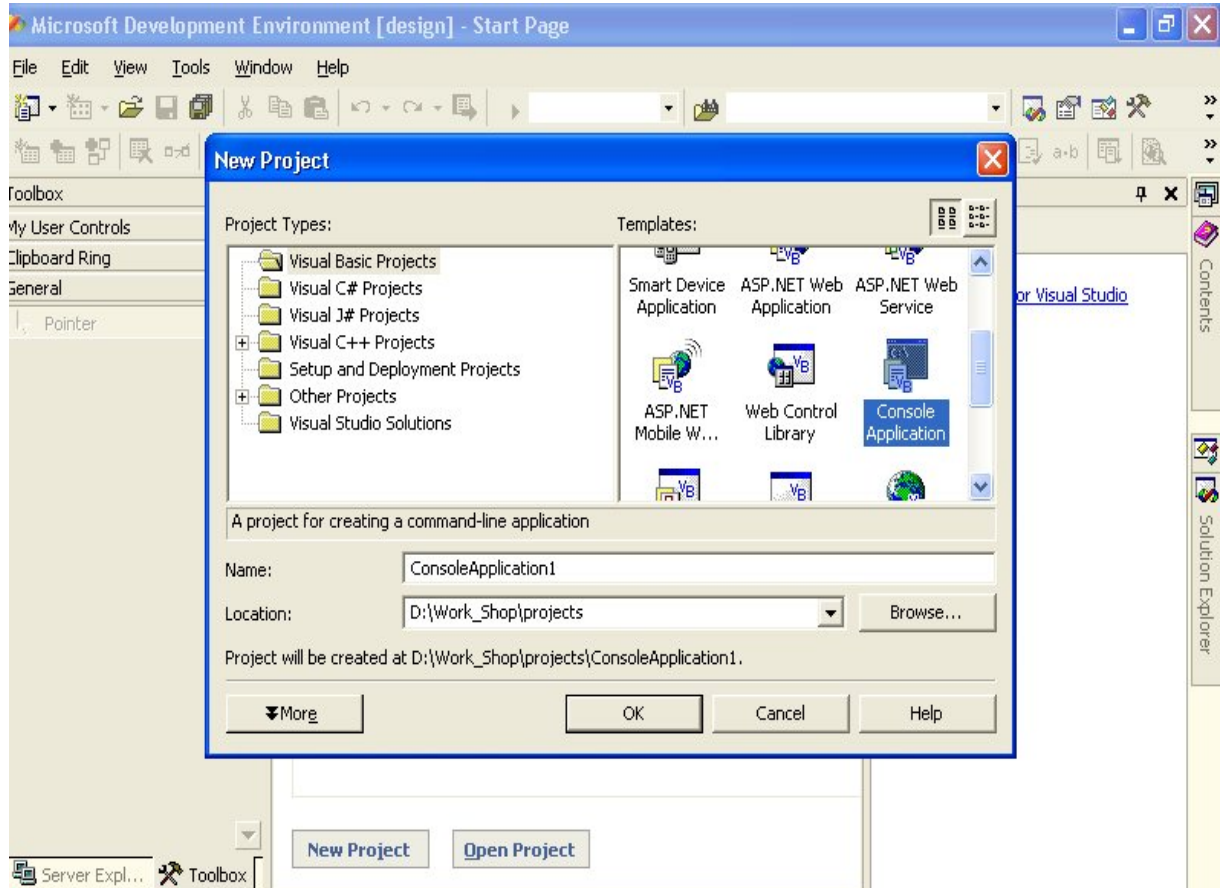
هذا النوع من التطبيقات يسمى بتطبيقات Console Applications

فهى تطبيقات ترى كما لو كانت تنفذ فى بيئة الدوس

فأنا أرى أنها أحد التطبيقات الإبتدائية لكل مبرمج مبتدىء فهى تتميز بالبساطة

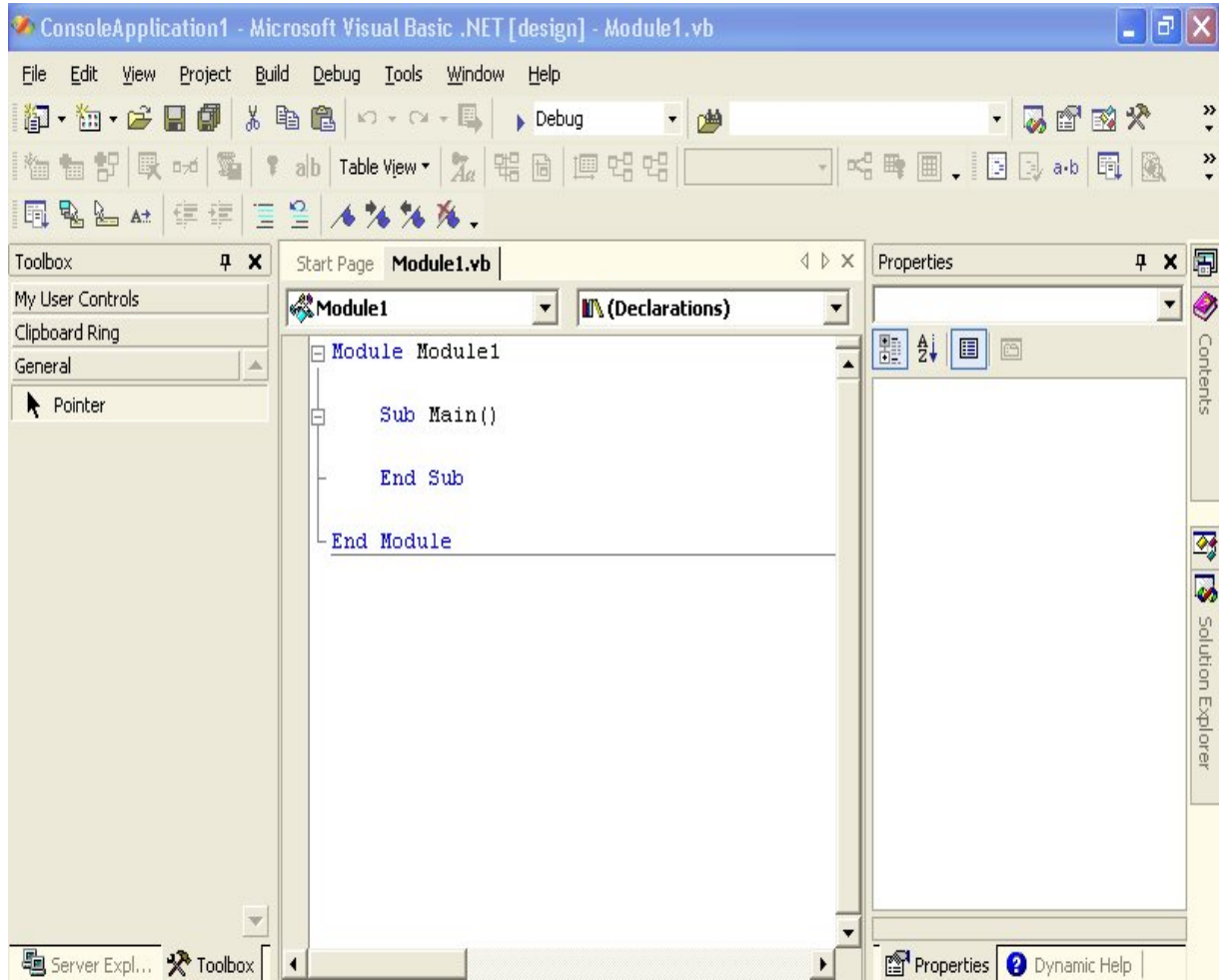
ولكنى لاأطيل فيها لأنها ليست محور هدفنا

فهدفى هو التركيز على Windows Applications



فهي لا تحتوي على أدوات كما قلنا تنفذ فهي مثل Dos Window

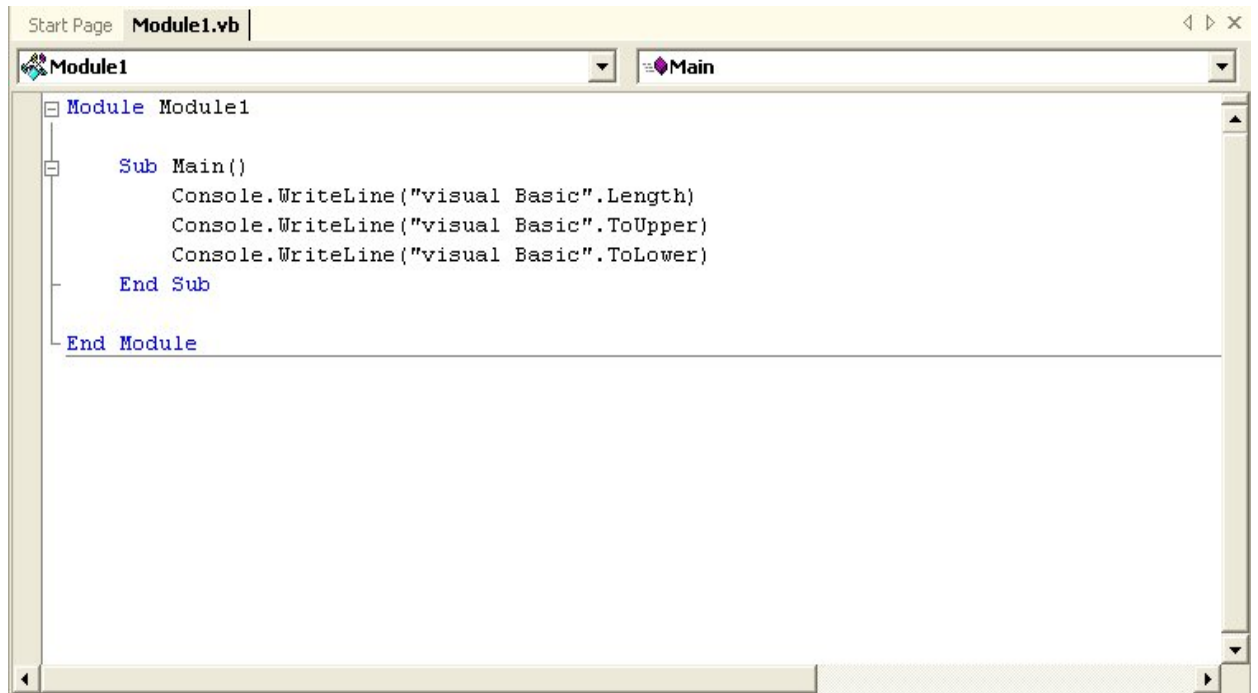
ومكان كتابة الكود كما هو موضح



ففي هذا المثال الأول لهذا النوع من التطبيقات تقوم الجملة الأولى بتحديد طول الجملة

والجملة الثانية تقوم بجعل الجملة بالحروف الكبيرة

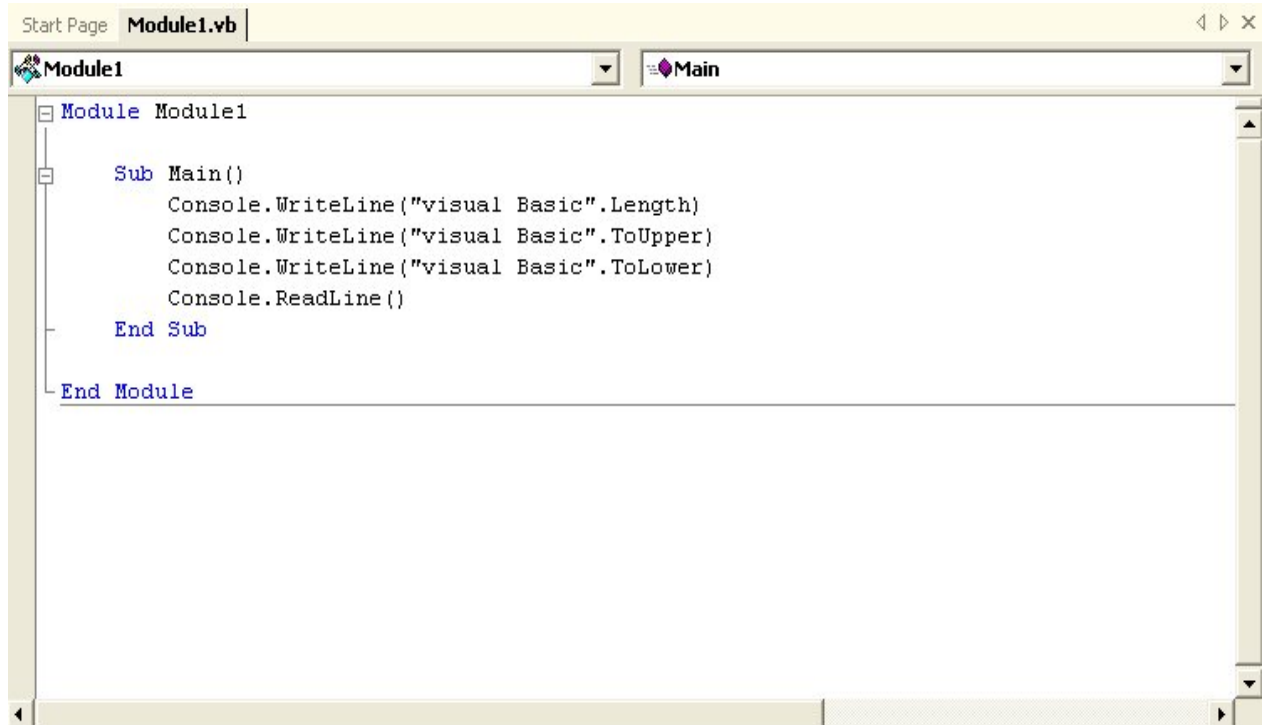
والجملة الثالثة تقوم بجعل الجملة بالحروف الصغيرة



```
Start Page Module1.vb
Module1
Main
Module Module1
    Sub Main()
        Console.WriteLine("visual Basic".Length)
        Console.WriteLine("visual Basic".ToUpper)
        Console.WriteLine("visual Basic".ToLower)
    End Sub
End Module
```

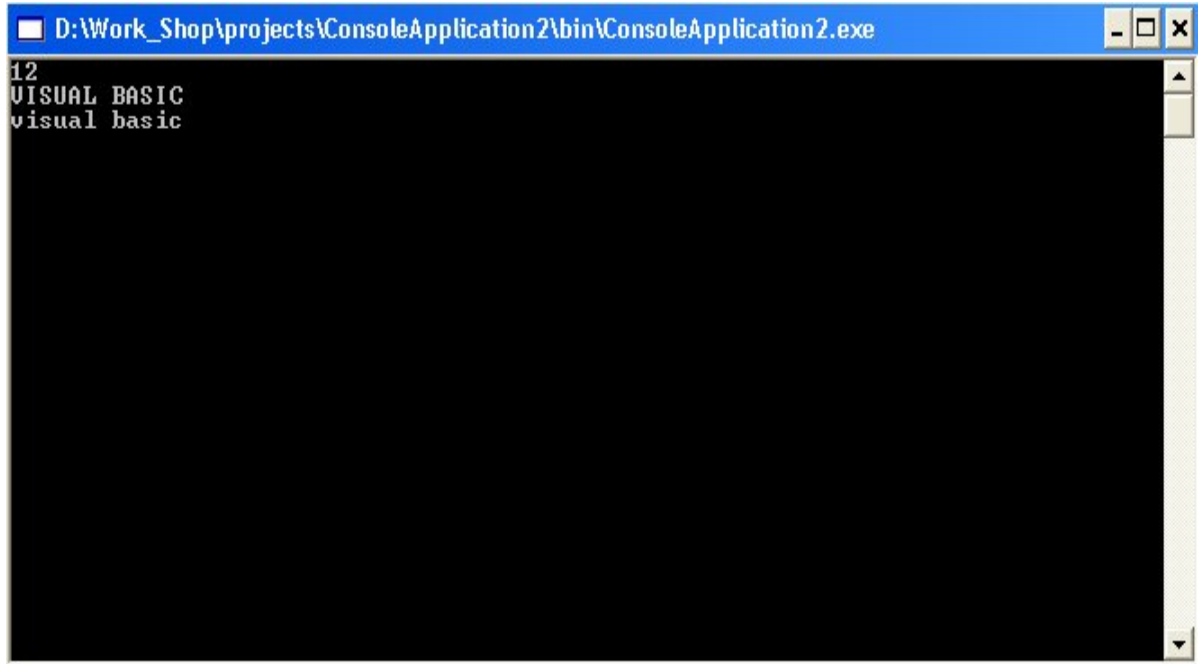
ولمعرفة Output نضغط F5

ولكن النتائج تظهر بسرعة ولذلك نكتب في الآخر `console.readline()`



```
Start Page Module1.vb
Module1
Main
Module Module1
    Sub Main()
        Console.WriteLine("visual Basic".Length)
        Console.WriteLine("visual Basic".ToUpper)
        Console.WriteLine("visual Basic".ToLower)
        Console.ReadLine()
    End Sub
End Module
```

وتكون Output



```
D:\Work_Shop\projects\ConsoleApplication2\bin\ConsoleApplication2.exe
12
VISUAL BASIC
visual basic
```

فهذا هو المثال الأول والأخير من هذا النوع من التطبيقات

المتغيرات Variables

فى بداية أى برنامج لابد من تعريف المتغيرات المستخدمة أولا فى بداية البرنامج الفرعى المستخدمة

فتستخدم Variables لتخزين البيانات ويمكن أن تكون هذه البيانات فى صورة

Text & Numbers & Times & Dates & Other Data Types

فالمتغير لابد من تعريفه أولا وهذه العملية تسمى **Declaration**

ولابد أيضا من تحديد نوع البيانات المخزنة فيه وتسمى **Data Type**

والمتغير لابد إلا يحتوى على مسافات أو رموز ولايزيد عن ٢٥٥ حرف ويمكن أن يحتوى على _

ولايجب تكرار أسم المتغير داخل البرنامج الفرعى الواحد

وتعريف المتغير يكون فى الصيغة الآتية

Dim VariableName As Data Type

فإذا أردنا تعريف متغير من النوع **Integer**

Dim X AS integer

Dim X,Y AS integer

ويمكن تعيين قيمة ابتدائية لهذا المتغير

Dim X AS integer = 5

فإذا أردنا تعريف متغير من النوع **String**

Dim X AS String

Dim X,Y AS String

ويمكن تعريف متغيرين من نوعين مختلفين

Dim X AS String = 5, Y AS Double = 3.4

وتنقسم أنواع البيانات Data Type إلى

String - Boolean - Date - Object - Number

ويستخدم النوع String

لتخزين الكتابات من النوع Text

Dim Mido AS String

Mido="محمدي"

ويستخدم النوع Boolean للاختبار

وهذا النوع أما True False

True = -1 , 1

False = 0

Dim Success AS Boolean

ويستخدم النوع Date لتخزين التواريخ

Dim xdate AS Date

xdate=#10/10/1985#

xdate=#10:10:00 AM#

ويصلح النوع Object لجميع أنواع البيانات Data Type

وهو أكثرهم مرونة

Dim Mido

Dim Mido AS Object

ويستخدم النوع Number إلى نوعين

Integer - Decimal

والنوع Integer ينقسم إلى أربع أنواع ويتضمن الأعداد الصحيحة الموجبة والسالبة و الصفر

Byte - Short (Int 16) - Integer (Int 32) - Long (Int 64)

أما النوع Decimal فهو لتخزين الأرقام العشرية Floating Point Numbers

وتنقسم إلى نوعان

Single - Double

والنوع Double أكثر دقة من النوع Single

وتستخدم Single في الحسابات العلمية

أنواع المتغيرات

Local Variables - Global Variables

أولا المتغيرات المحلية

وهو أن المتغير المعرف في البرنامج الفرعي يكون معرف له فقط أى لا يمكن إستخدامه في برنامج فرعي واحد

```
1 Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
2     Dim x As String
3     x = "محمدى"
4     MsgBox(x)
5 End Sub
```

ثانيا المتغيرات العامة

فهي تأخذ شكلين

أما تعريفها في النموذج الحالى وينتهى عمر Life Time بمجرد إنتهاء النموذج المعرف فيه المتغير

```

Public Class Form1
    Inherits System.Windows.Forms.Form
    Dim x As String
    Windows Form Designer generated code
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        x = "محمدى"
        MsgBox(x)
    End Sub
End Class

```

أما لتعريف المتغير فى المشروع بأكمله فتعرف **Public Shared**

```

Public Class Form1
    Inherits System.Windows.Forms.Form
    Public Shared x As String

```

وتستخدم فى النماذج الأخرى فى هذه الصورة

Form1.x.(Method)

الثوابت Constants

الثوابت هي متغيرات تعرف من اجل تسهيل عملية البرمجة

Const X AS Integer = 5

ويمكن تعريفها مثل المتغيرات

فإستخدامها كثابت عام فى المشروع كله

Public Const X AS Integer = 5

أحد البرامج الفرعية وإذا كتابنا فى

$x = 2$

تظهر رسالة خطأ لأنها معرفة من قبل على إنها ثابت

العمليات الرياضية Math Operations

تستخدم * فى عملية الضرب

تستخدم / فى عملية القسمة

تستخدم \ فى عملية باقى القسمة

تستخدم - فى عملية الطرح

تستخدم + فى عملية الجمع

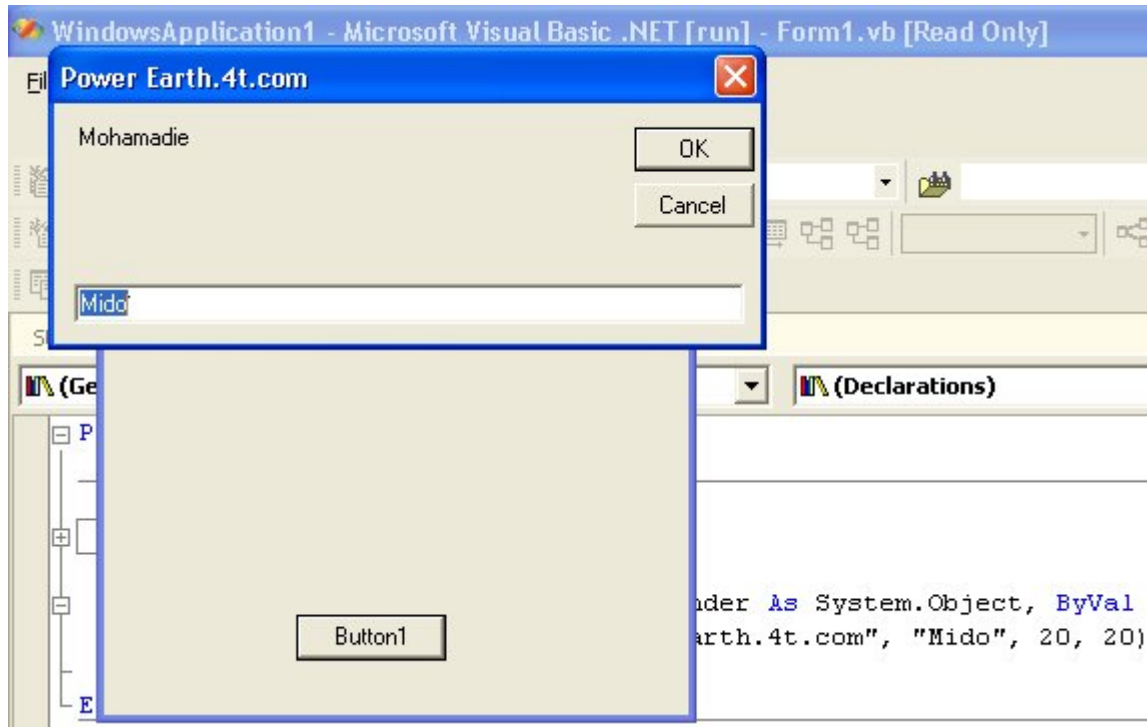
تستخدم ^ فى عملية رفع الأس

وتوجد أولوية لهذه العمليات من حيث التنفيذ

inputbox(Prompt,[title],[Default],[Xpos],[Ypos])

inputbox("Mohamadie","Power Earth.4t.com","Mido",20,20)

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    InputBox("Mohamadie", "Power Earth.4t.com", "Mido", 20, 20)
End Sub
```



ويمكن وضعها في متغير

Dim m As String

m = Inputbox("Mohamadie","Power Earth.4t.com","Mido",20,20)

الجمل الشرطية Conditions

تستخدم الجمل الشرطية للأختبار

أولا

If . . . Then

وتكون على الشكل التالي

If Condition Then Statement 1:Statement 2:Statement 3:.....

If X = "Mohamadie" Then End:Msgbox("Power Earth.4t.com")

فيتم أولا إختبار الشرط إذا تحقق يقوم بتنفيذ الجمل

وإذا لم تتحقق يذهب إلى السطر التالي

ثانيا

If Condition Then

Statement 1

Statement 2

Statement 3

Statement n

End If

If Year = 1985 Then

Name = "Mohamadie"

Faculty = "CS & IS"

Country ="Egypt"

End If

فيتم إختبار الشرط إذا حققه

يتم تنفيذ الجمل التالية وإذا لم يحققه ينتقل إلى السطر الذي بعد End If

ثالثا

If Condition Then

StatementBlock 1

StatementBlock 2

StatementBlock 3

StatementBlock n

Else

StatementBlock n+1

End If

If A = "Tanta" Then

Name = "Mohamadie"

Else

Msgbox("Not Included")

End If

Name = "Mohamadie" يتم اختبار الشرط إذا تحقق يكون

Else وإذا لم يتحقق تنفذ الجملة بعد

رابعاً

If Condition Then

Statement 1

ElseIf Condition Then

Statement 2

Else

Statement

End If

Statement 1 إذا تحقق الشرط الأول ينفذ

Statement 2 وإذا لم ينفذ ينتقل إلى الشرط الذي يليه وإذا تحقق ينفذ

وهكذا

Else وإذا لم يتحقق أي شرط يتم الذهاب إلى جملة

خامساً

الجملة الشرطية

Select Case TestValue

Case Value 1

Statement 1

Case Value 2

Statement 2

End Select

Dim X AS Single

X = Textbox1.text

Select Case x

Case Is < 50

Msgbox("كل سنة وأنت طيب")

Case 50:64

Msgbox("مقبول")

Case Is = 65.1

Msgbox("نتيجة محمدى فى السنتين الأولتين من الكلية-جيد")

End Select

X < 50 فإذا كانت

تنفذ الجملة التالية لهذا الشرط

وإذا لم تنفذ تنتقل إلى الشرط الذى يليه

وهكذا

الحلقات التكرارية

Loops

تستخدم الحلقات التكرارية لتكرار تنفيذ بعض العمليات

أولا

For ... Next

For Counter = Start To End [Step Increment]

Statements

Next [Counter]

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim i As Integer
    For i = 0 To 10
        MsgBox("Mohamadie")
    Next
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim i As Integer
    For i = 0 To 10 Step 2
        MsgBox("Mohamadie")
    Next
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim i As Integer
    For i = 10 To 2 Step -1
        MsgBox("Mohamadie")
    Next
End Sub
```

ثانيا

Do . . . Loop

Do While Condition

Statements

Loop

إذا تحقق الشرط يتم تنفيذ تكرار العمليات داخل Loop

وإذا لم يتحقق ينتقل إلى السطر الذي بعد Loop

Do . . . Loop

Do Until Condition

Statements

Loop

إذا لم يتحقق الشرط يتم تنفيذ تكرار العمليات داخل Loop

وإذا لم يتحقق ينتقل إلى السطر الذي بعد Loop

خامسا

While . . . End While

While Condition

Statements

End While

While = طالما

طالما الشرط True نفذ الجمل

وإذا كان False أنتقل بعد End While

المصفوفات

Arrays

تستخدم المصفوفات فى عمليات Store In Collections

ومن أقرب الأمثلة للمصفوفات فى التطبيق Hash Table

وتعرف كالتالى

مثلا

Dim Names(5) As Integer

وبذلك تم تعريف مصفوفة عددها ٦ عناصر

Names(0)

Names(1)

Names(2)

Names(3)

Names(5)

**Dim Names(5) As Integer = { محمدى ، مصر ، طنطا ، كلية الحاسبات والمعلومات ، }
{جامعة المنصورة ، الفرقة الثالثة**

ويمكن كتابتها على الشكل التالى

Names(0) = محمدى

Names(5) = الفرقة الثالثة

ويمكن معرفة الحد الأدنى للمصفوفة وهو في جميع الأحوال صفر

Msgbox(LBound(Names))

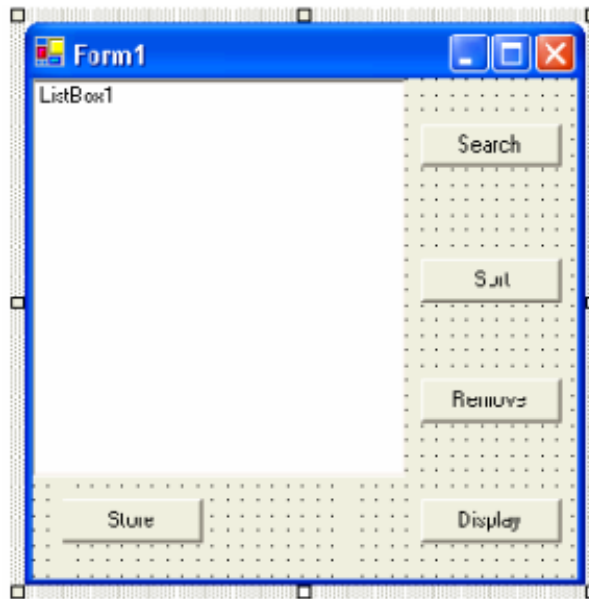
ويمكن معرفة الحد الأعلى للمصفوفة

Msgbox(UBound(Names))

ويمكن معرفة طول المصفوفة أي عدد عناصرها

Msgbox(Names.length)

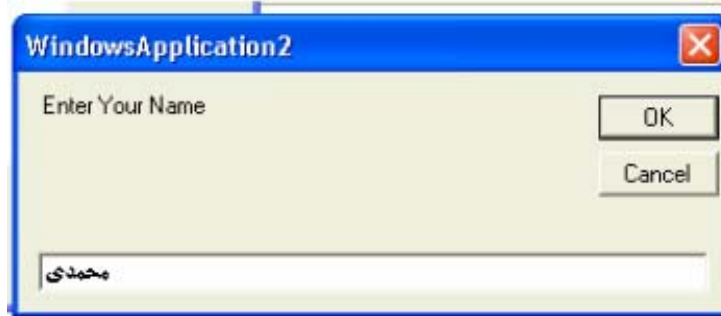
مثال توضيحي



لتخزين المصفوفة

```
Public Class Form1
    Inherits System.Windows.Forms.Form
    Dim names() As String
    Dim i As Integer = 0
    Windows Form Designer generated code

    Private Sub Button1_Click(ByVal sender As
        Dim name As String
        name = InputBox("Enter Your Name")
        While name <> "Mohamadie"
            ReDim Preserve names(i)
            i = i + 1
            name = InputBox("Enter Your Name")
        End While
    End Sub
End Class
```



لعرض المصفوفة

```
Private Sub Display_Click(ByVal sender As System
    Dim x As Integer
    For x = 0 To UBound(names)
        ListBox1.Items.Add(names(x) + vbNewLine)
    Next
End Sub
```



لترتيب المصفوفة

```
Private Sub Sort_Click(ByVal sender As System
    Array.Sort(names)
End Sub
End Class
```

للبحث في المصفوفة

```
Private Sub Search_Click(ByVal sender
    Dim h As String
    Dim f As Integer
    h = InputBox("Enter Your Search")
    f = Array.BinarySearch(names, h)
    If f < 0 Then
        MsgBox (Now)
        MsgBox (Today)
        MsgBox ("Not Found")
    Else
        MsgBox (f)
    End If
End Sub
```

للأزالة في المصفوفة

```
Private Sub Removr_Click(ByVal sender As System.
    ListBox1.Items.Remove (ListBox1.SelectedItem)
End Sub
```

الأجراءات Procedures

تستخدم الأجراءات لتقليل حجم الكود ولك يتجميع Procedure يمكن إستدعائه عند الحاجة إليه جزء من الكود فى صورة

والصيغة العامة للأجراء هى

[Privat|Public] Sub ProcedureName(Arg 1 , Arg 2 , ...)

وتكتب فى أى برنامج فرعى ويمكن إستدعائها

وصيغة الإستدعاء

Call ProcedureName(Arg 1 , Arg 2 , ...)

```
Private Sub Button1_Click
    MsgBox("Mohamadie")
End Sub
Private Sub mido()
    MsgBox(Today)
End Sub

Private Sub Button2_Click
    Call mido()
End Sub
```

Private ليكون الأجراء مستخدم فى النموذج المتواجد فيه أو الوحدة البرمجية البرنامج الفرعى تستخدم كلمة

تستخدم كلمة **Public** ليكون الأجراء مستخدم فى المشروع كله

وتستخدم فى النماذج الأخرى على هذه الصورة

Call Form1.mido()

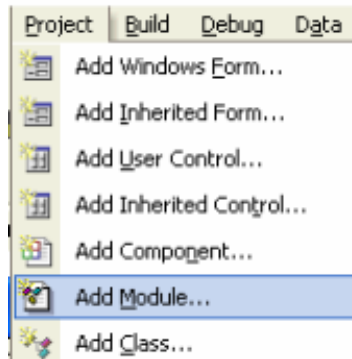
```
Public Sub CalcSum(ByVal Num1 As Integer, ByVal Num2  
    MsgBox(Num1 + Num2)  
End Sub
```

الوحدات البرمجية

Modules

الوحدات البرمجية هي مكان **Procedures - Functions** وإيضا يمكن تعريف المتغيرات العامة يمكن كتابة

لتسهيل عملية كتابة الكود



وهي تأخذ الأمتداد **.vb**.

```
Start Page | Form1.vb [Design] | Module1.vb* |
\ (General)
Module Module1
  Public Sub CalcSum(ByVal Num1 As
    MsgBox (Num1 + Num2)
  End Sub
End Module
```

ويمكن إستدعائها بجملة **Call** في أى برنامج فرعى

الوظائف Functions

الوظائف مثل الإجراءات ولكن الوظيفة تقوم بإرجاع ناتج لهذه الوظيفة
وتكون صيغتها كالتالي

[Privat|Public] Sub FunctionName(Arg 1 , Arg 2 , ...) As DataType

**Public Function TotalSum(Num1 AS Integer,Num2 AS Integer) AS
integer**

TotalSum = Num1+Num2

End Function

Dim T As Integer

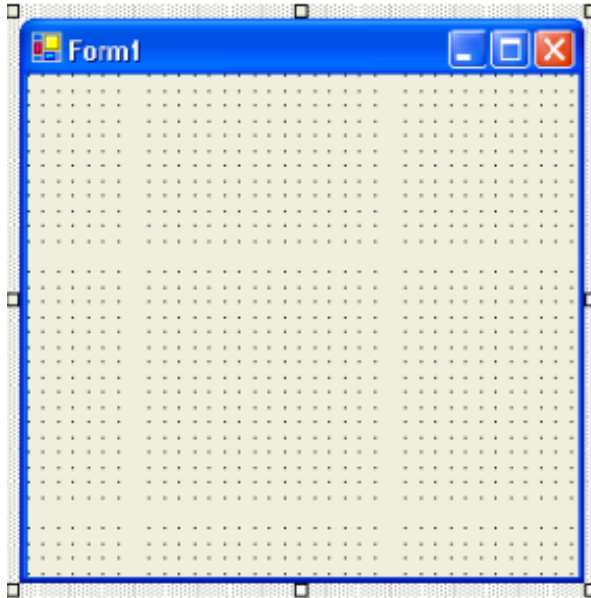
T = TotalSum(10+10)

التعامل مع النماذج

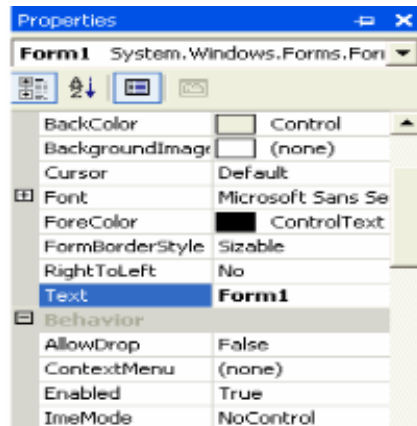
Forms

النموذج هو الذي من خلاله يكمن Application

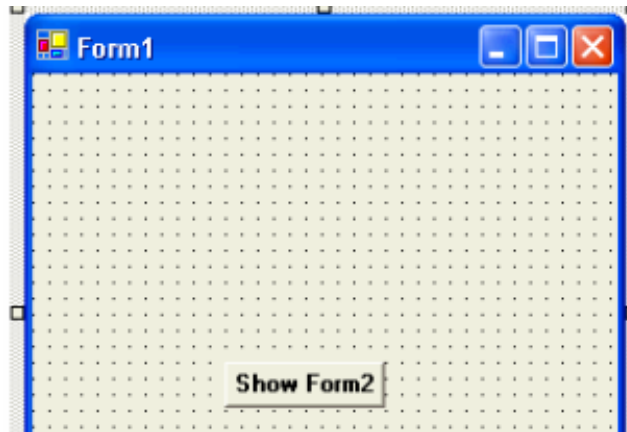
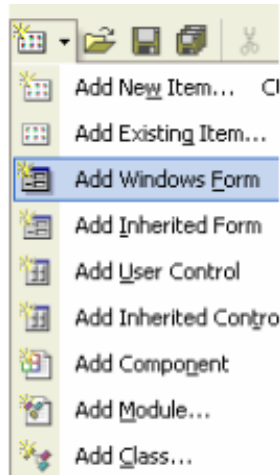
Form1



ويكون لهذا النموذج عدة خصائص يمكن التحكم فيها من خلال نافذة الخصائص Properties



أما بالنسبة لفتح الفورم أو النماذج من بعضها

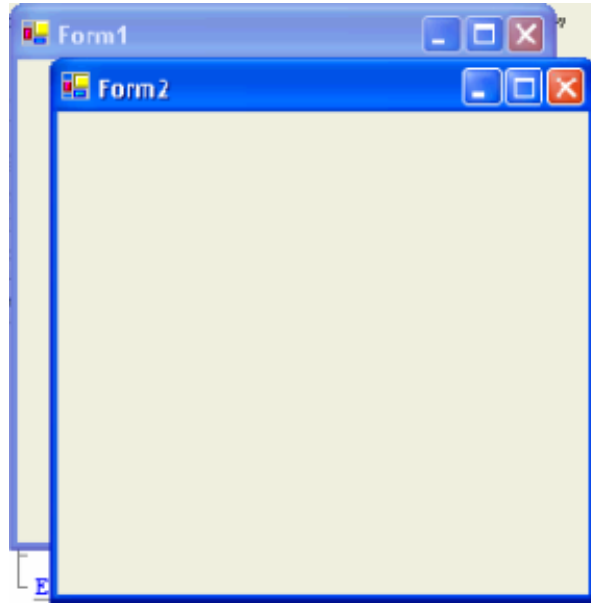


فالظهور النموذج الثاني نكتب هذا الكود

```
Public Class Form1
    Inherits System.Windows.Forms.Form
    Dim frm As New Form2
    Windows Form Designer generated code

    Private Sub Button1_Click(ByVal sender
        frm.Show()
    End Sub
End Class
```

فتكون النتيجة كالآتي



ويمكن جعل النموذج الأول False في حين ظهور النموذج الثاني

ويمكن إخفاء النموذج الأول باستخدام الأسلوب `frm.Hide()`

```
Public Class Form1
    Inherits System.Windows.Forms.Form
    Dim frm As New Form2
    Windows Form Designer generated code

    Private Sub Button1_Click(ByVal sender As Object, ByVal e As EventArgs) Handles Button1.Click
        frm.ShowDialog()
    End Sub
End Class
```

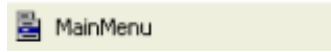
ويمكن تعريف النموذج Public ليستخدم في المشروع كله

إنشاء القوائم Menus

القوائم من العناصر المهمة **Interface** لذلك تساعد القوائم في تسهيل الخدمات عبر المستخدم في شكل البرنامج

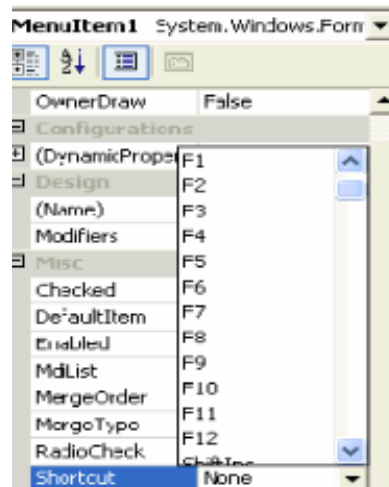
ولعمل القوائم

توفر لنا لغة **VB.NET** أداة لعمل القوائم



والقوائم لها عدة خصائص من خلال نافذة الخصائص

ويمكن إستخدام مفاتيح للإختصار من خلال نافذة الخصائص



ويمكن إضافة الكود بالضغط مرتين على أي جزء من أجزاء القائمة

القوائم المختصرة Shortcut Menus

القوائم المختصرة وهي ممكن نعبّر عنها **Right Click**

المختصرة القوائم **Interface** لذلك تساعد القوائم المختصرة في تسهيل الخدمات عبر المستخدم من العناصر المهمة في شكل

ولعمل القوائم المختصرة

توفر لنا لغة **VB.NET** أداة لعمل القوائم المختصرة

ContextMenu



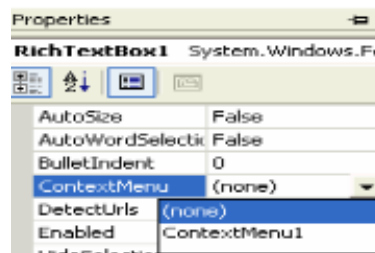
ويمكن إضافة الكود بالضغط مرتين على أي جزء من **ContextMenu**

والقوائم المختصرة لها عدة خصائص من خلال نافذة الخصائص

ويجب تحديد الجزء الذي يوجد فيه القائمة المختصرة وذلك من خلال هذه الخاصية

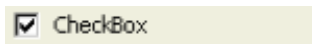
مثلا إذا أردنا وضع القائمة المختصرة في الأداة **RichTextBox**

نذهب إلى خصائص هذه الأداة المراد وضع القائمة المختصرة فيها والصورة الآتية توضح



CheckBox Examples

تستخدم CheckBox في مواضع كثيرة في كثير من البرامج



مثال توضيحي

```
Private Sub Button1_Click(ByVal sender As System
    If CheckBox1.Checked = True Then
        Application.Exit()
    End If
End Sub
```

مثال آخر توضيحي

If CheckBox1.Checked = True Then

**TextBox1.Font = New System.Drawing.Font(TextBox1.Font,
FontStyle.Bold)**

End If

مثال آخر توضيحي

```
Private Sub Button1_Click(ByVal sender As System.C
    If CheckBox1.Checked = True Then
        Me.FormBorderStyle = FormBorderStyle.None
    End If
```

RadioButton Examples

تستخدم **RadioButton** في مواضع كثيرة في كثير من البرامج

RadioButton

مثال توضيحي

```
Private Sub Button1_Click(ByVal sender As  
    If RadioButton1.Checked = True Then  
        End  
    End If  
End Sub
```

مثال آخر توضيحي

If RadioButton1.Checked = True Then

**TextBox1.Font = New System.Drawing.Font(TextBox1.Font,
 FontStyle.Bold)**

End If

مثال آخر توضيحي

```
Private Sub Button1_Click(ByVal sender As System.'  
    If RadioButton1.Checked = True Then  
        Me.FormBorderStyle = FormBorderStyle.None  
    End If  
End Sub
```

ListBox Examples

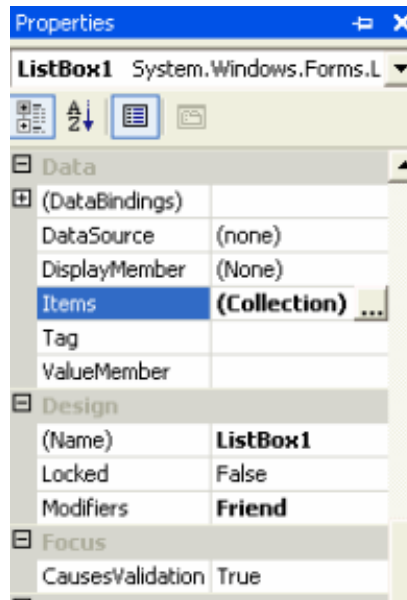
تستخدم **ListBox** لتجميع عدد من **Items**



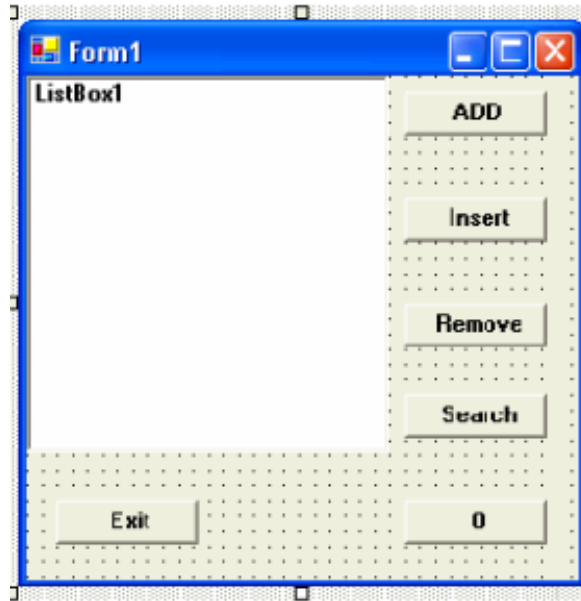
ولها عدة خصائص يمكن التعرف عليها من خلال نافذة الخصائص

ومن أهمها خاصية **Items**

لتخزين البيانات بها



مثال توضيحي



إضافة عناصر جديدة

```
Private Sub Button1_Click(ByVal sender
    Dim m As String
    m = InputBox("Enter Your Item")
    ListBox1.Items.Add(m)
End Sub
```

إدخال عناصر جديدة

ListBox1.Items.Insert(Index, "")

```
Private Sub Button2_Click(ByVal sender
    Dim n As String
    n = InputBox("Enter Your Item")
    ListBox1.Items.Insert(2, n)
End Sub
```

لحذف عناصر

```
Private Sub Button3_Click(ByVal sender
    Dim r As String
    r = InputBox("Enter Your Item")
    ListBox1.Items.Remove(r)
End Sub
```

للبحث عن عناصر

```
Private Sub Button4_Click(ByVal sender
    Dim s As String, b As Integer
    s = InputBox("Enter Your Item")
    b = ListBox1.FindString(s)
    MsgBox(b)
End Sub
```


إزالة كل العناصر

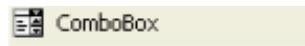
```
Private Sub Button5_Click(  
    ListBox1.Items.Clear()  
End Sub
```

للخروج من البرنامج

```
Private Sub Button6_Click  
    End  
End Sub
```

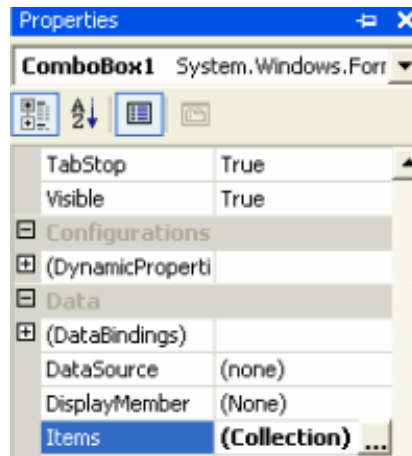
ComboBox Examples

تستخدم ComboBox في مواضع كثيرة في كثير من البرامج



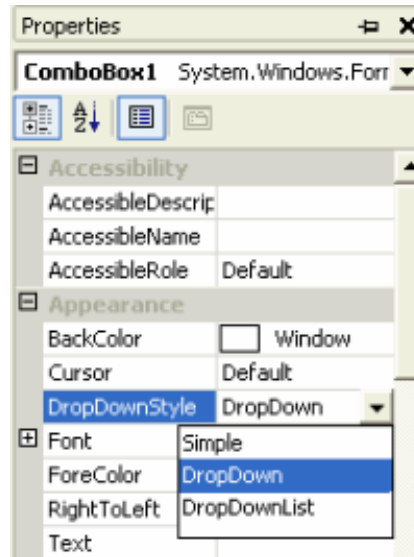
لها عدة خصائص من خلال نافذة الخصائص

وأهمها خاصية وضع Items داخل القائمة المنسدلة



وهي تنفذ مثل ListBox1 فهي تأخذ نفس الأساليب في عملية البرمجة

أما هذه الخاصية تتحكم في شكل القائمة المنسدلة



مثال توضیحی

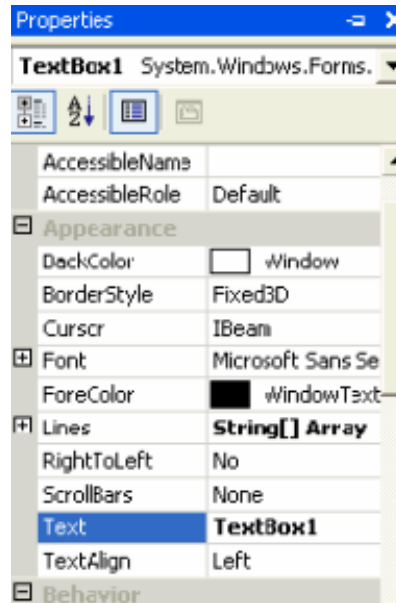
```
Private Sub Button1_Click(ByVal sender As  
    If ComboBox1.SelectedItem = "3" Then  
        MsgBox("الأسلام هو الحل")  
    End If  
End Sub
```

TextBox Examples

من العناصر الهامة جدا في عملية إدخال البيانات
TextBox من الأدوات المهمة في تصميم البرامج

lab1 TextBox

لها عدة خصائص من خلال نافذة الخصائص



وهي تستخدم لكتابة بيانات مدخلة يمكن عمل Access عليها

مثال توضيحي

```
Private Sub Button1_Click(ByVal sender  
    If TextBox1.Text = "Mido" Then  
        MsgBox("الاسلام هو الحل")  
    End If  
End Sub
```

وفي هذا المثال يمكننا معرفة مكان كلمة معينة داخل TextBox

```
Private Sub Button1_Click(ByVal sender
    Dim m As String
    Dim i As Integer
    m = InputBox("Your Search")
    i = InStr(TextBox1.Text, m)
    MsgBox(i)
End Sub
```

وفي هذا المثال يمكننا تنظيف TextBox

```
Private Sub Button1_Click
    TextBox1.Text = ""
End Sub
```

جمع عددين

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

RichTextBox Examples

أولا لابد من التعرف على RTF Language

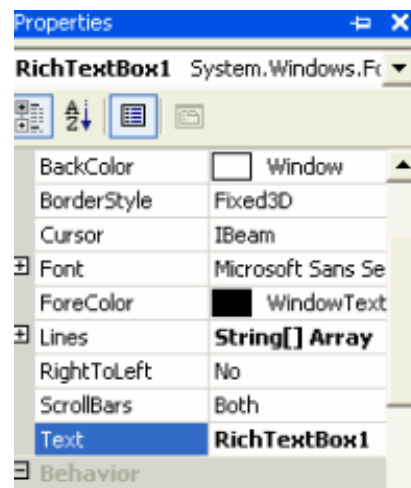
تستخدم هذه الأوامر RTF فى برامج التى يوجد بها Text

مثل Notepad-Wordpad-HTML

وتوجد هذه الأداة فى هذا الموضوع من شريط الأدوات



لها عدة خصائص من خلال نافذة الخصائص



وهى تستخدم لكتابة بيانات مدخلة يمكن عمل Access عليها

مثال توضيحي لعمل بحث داخل RichTextBox

```

Dim m As String
Dim i As Integer
m = InputBox("Your Search", "Find", "Mohamadie")
i = RichTextBox1.Find(m)
If i < 0 Then
    MsgBox("Not Found")
Else
    MsgBox(i)
End If

```

ولعمل Align أى تحديد المكان للكتابة بداخله بمعنى اليمين واليسار والوسط

```
RichTextBox1.Selection.Alignment = HorizontalAlignment.
```

ولعمل SelectAll

```

Private Sub MenuItem16_Click
    RichTextBox1.SelectAll()
End Sub

```

ولعمل Delete

```

Private Sub MenuItem6_Click(ByVal
    RichTextBox1.SelectedText = ""
End Sub

```

ولعمل Cut

```

Private Sub MenuItem3_Click
    RichTextBox1.Cut()
End Sub

```

ولعمل Copy

```

Private Sub MenuItem4_Click
    RichTextBox1.Copy()
End Sub

```

ولعمل Paste

```

Private Sub MenuItem5_Click
    RichTextBox1.Paste()
End Sub

```

ولعمل Undo

```

Private Sub MenuItem1_Click(ByVal sender As System.
    If RichTextBox1.CanUndo Then RichTextBox1.Undo()
End Sub

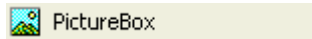
```

ولعمل Redo

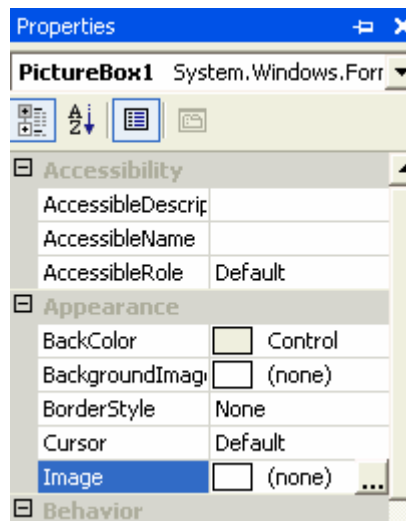
```
Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem2.Click
    If RichTextBox1.CanRedo Then RichTextBox1.Redo()
End Sub
```


PictureBox Examples

تستخدم PictureBox لوضع صور في مكان ما في البرنامج أو في برامج خاصة بالصور



ولها عدة خصائص من خلال نافذة الخصائص



مثال توضيحي

لعمل Rotate

```
PictureBox1.Image.RotateFlip(RotateFlipType.Rotate270FlipXY)
```

```
PictureBox1.Refresh()
```

لعمل Zoom Out

```
PictureBox1.Width = PictureBox1.Width * 0.25  
PictureBox1.Height = PictureBox1.Height * 0.25
```

لعمل Zoom In

```
PictureBox1.Width = PictureBox1.Width * 1.25  
PictureBox1.Height = PictureBox1.Height * 1.25
```

لعمل Copy

```
Private Sub Button1_Click(ByVal sender As Syst  
    Clipboard.SetDataObject (PictureBox1.Image)  
End Sub
```

لعمل Paste

**If Clipboard.GetDataObject.GetDataPresent(DataFormats.Bitmap)
Then**

**PictureBox1.Image =
 Clipboard.GetDataObject.GetData(DataFormats.Bitmap)**

End If

لعمل Clear

```
Private Sub Button1_Click(ByVal sender As System.  
    PictureBox1.CreateGraphics.Clear (Color.Black)  
End Sub
```

Common Dialog Controls Examples

FontDialog Common Dialog لمعالجة الخط ويسمى تأخذ صور عديدة فمنها FontDialog



ومن يتعامل مع الألوان ColorDialog



ومن يتعامل مع فتح الملفات OpenFileDialog



ومن يتعامل مع حفظ الملفات SaveFileDialog



ومن يتعامل مع طباعة الملفات PrintDialog



وكل أداة لها عدة خصائص من خلال نافذة الخصائص

أولا مثال على FontDialog

```
FontDialog1.Font = RichTextBox1.Font
If FontDialog1.ShowDialog = DialogResult.OK Then
    RichTextBox1.Font = FontDialog1.Font
End If
```

مثال على ColorDialog

```
If ColorDialog1.ShowDialog = DialogResult.OK Then
    Me.BackColor = ColorDialog1.Color
End If
```

مثال على OpenFileDialog

OpenFileDialog1.Filter = "Images|*.bmp;*.gif"

If OpenFileDialog1.ShowDialog = DialogResult.OK Then

PictureBox1.Image = Image.FromFile(OpenFileDialog1.FileName)

End If

مثال على SaveFileDialog

SaveFileDialog1.Filter = "Images|*.bmp;*.gif"

If SaveFileDialog1.ShowDialog = DialogResult.OK Then

PictureBox1.Image = Image.FromFile(SaveFileDialog1.FileName)

End If

الرسم Graphics

أولا لابد من التعرف على مكتبة الجرافيك وهي GDI OR GDI+

Graphic Design Interface

هي مجموعة من Classes تمكنك لعمل الجرافيك والنصوص والصور

بإختصار هي محرك الجرافيك في Windows

أما GDI+ فهي الأصدار الجديد في الدوت نت

مثال لرسم خط

لابد من تعريف Pen و Two EndPoints

وملاحظة للرسم في الفورم لابد أن يكون الحدث Event هو Paint

```
Private Sub Form1_Paint(ByVal sender As Object, E
    Dim m As Pen = New Pen(Color.Red, 2)
    Dim point1 As Point = New Point(10, 10)
    Dim point2 As Point = New Point(120, 180)
    Me.CreateGraphics.DrawLine(m, point1, point2)
End Sub
```

مثال لرسم مستطيل

Dim P As Pen = New Pen(Color.Gray)

Dim width As Integer = 50

Dim height As Integer = 100

Me.CreateGraphics.DrawRectangle(p, 5, 5, width, height)

فيكون الشكل العام للمستطيل

Me.CreateGraphics.DrawRectangle(Pen, X, Y, width, height)

لرسم شكل بيضاوي Ellipse

يكون الشكل العام

Me.CreateGraphics.DrawEllipse(Pen, X, Y, width, height)

وبذلك نرى أن الرسم يتطلب Pen و Points

أما بالنسبة للتلوين نستخدم Brush

وتكون صيغتها كالتالي

Dim sBrush As SolidBrush

sBrush = New SolidBrush(brushcolor)

TreeView Examples

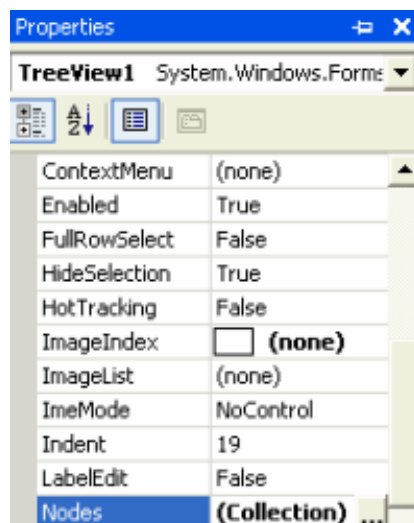
تستخدم TreeView Control فى مواضع كثيرة فى بعض البرامج

فهى تستخدم لتخزين المعلومات بطريقة هرمية



ولها عدة خصائص من خلال نافذة الخصائص

وأهم خاصية فى نافذة الخصائص هى إضافة Nodes و Root



إضافة Node

```
Private Sub Button1_Click(By  
    TreeView1.Nodes.Add("m")  
End Sub
```

لحذف Node

Nodes.Remove(index)

OR

TreeView1.SelectedNode.Remove

OR

TreeView1.Nodes(0).Nodes(2).Remove

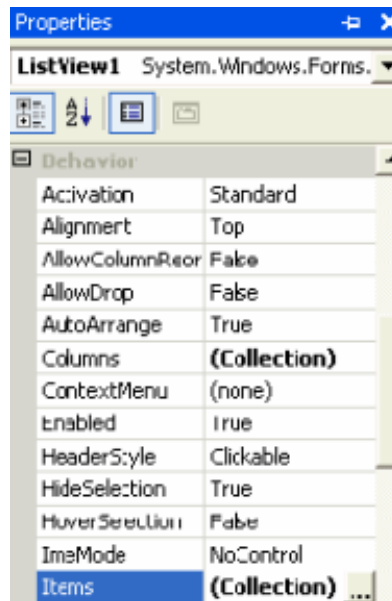
List View Examples

تستخدم **List View Control** في مواضع كثيرة في بعض البرامج

فهي تستخدم لتخزين المعلومات بصور مختلفة



ولها عدة خصائص من خلال نافذة الخصائص



فبعد الأضافة

```
List View1.Items.Add(Caption,index)
```

```
List View1.Items(index1).SubItems(index2)
```

```
List View1.CoulmnHeaders.Add("Mohamadie")
```

وتقريباً برمجتها نفس **TreeView Control**

ومن أهم الخصائص **TreeView** هذه الخاصية



وهي تتحكم في شكل **TreeView**

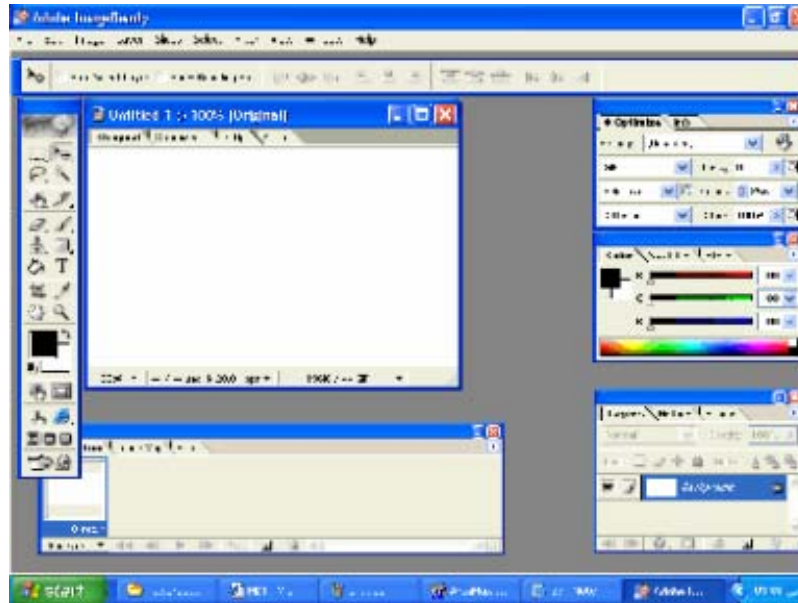
MDI

واجهة البرنامج ذو النماذج المتعددة

Multiple Document Interface (MDI)

فمن أقرب الأمثلة لهذا النوع مثل

Adobe ImageReady - Excel - PaintShop Pro - Macromedia Flash Mx . . .

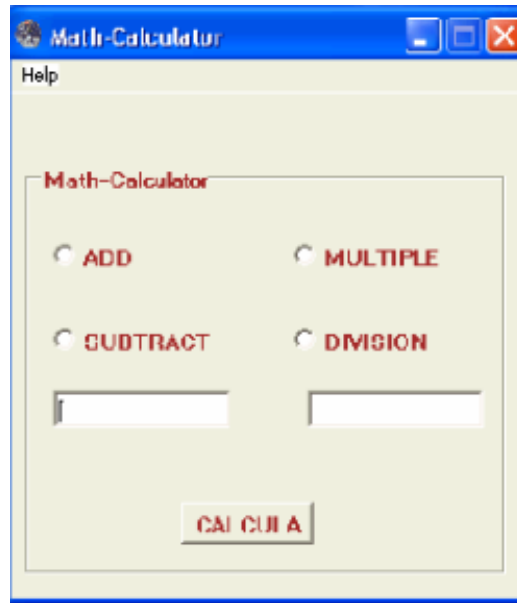


والعكس في SDI

Single Document Interface (MDI)

فمن أقرب الأمثلة لهذا النوع مثل

Calaculator . . .

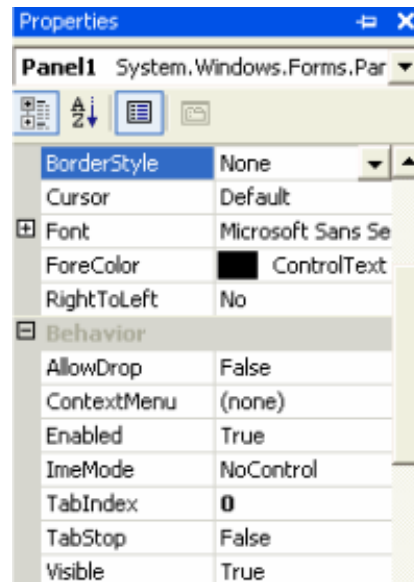


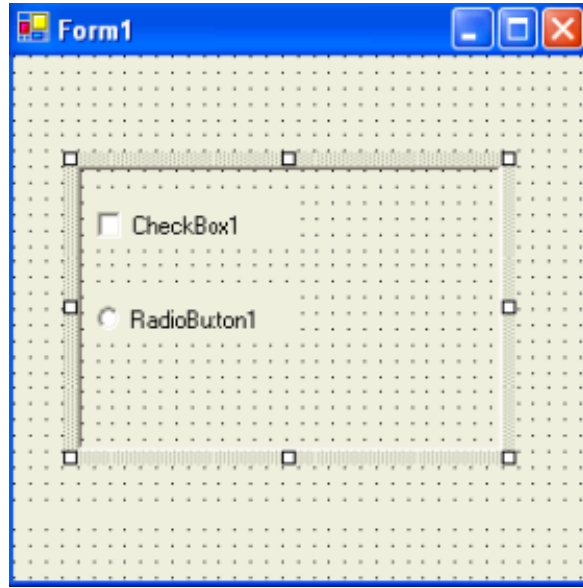
Panel

تستخدم Panel لتحتوى على بعض Tools



ولها عدة خصائص للتحكم فى الشكل وغيرها





وتشبه هذه الأداة **GroupBox**

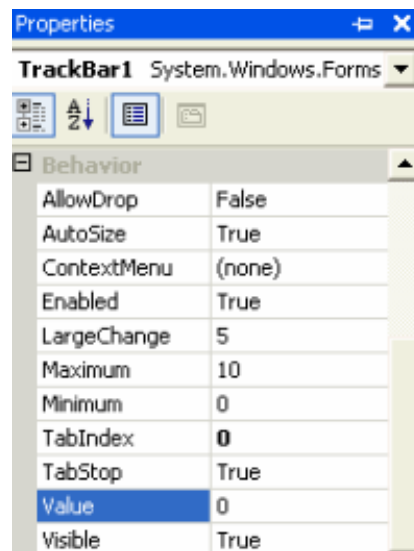


TrackBar

نرى هذه الأداة في الكثير من البرامج



ولها عدة خصائص



ويمكن التحكم في الشكل رأسى أو أفقى من هذه الخاصية



مثال توضيحي

```
Private Sub TrackBar1_Scroll(ByVal  
    Label1.Text = TrackBar1.Value  
End Sub
```

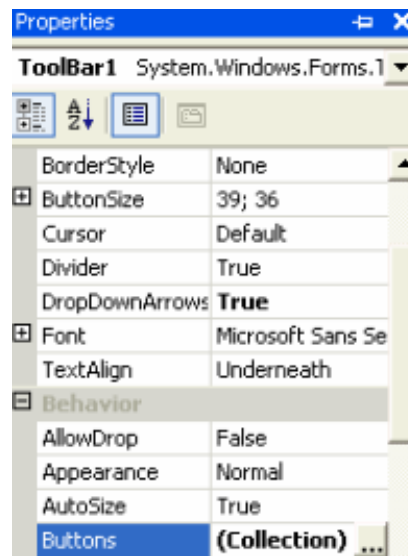
ملحوظة يجب أن تكون خاصية Value محصورة بين خاصيتي Maximum & Minimum

ToolBar

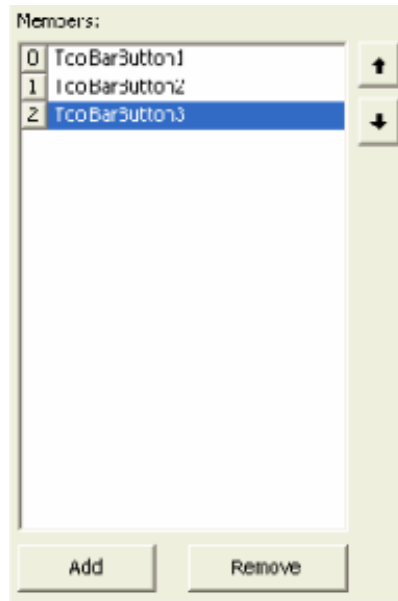
تستخدم لتخزين بعض Buttons في أعلى الفورم



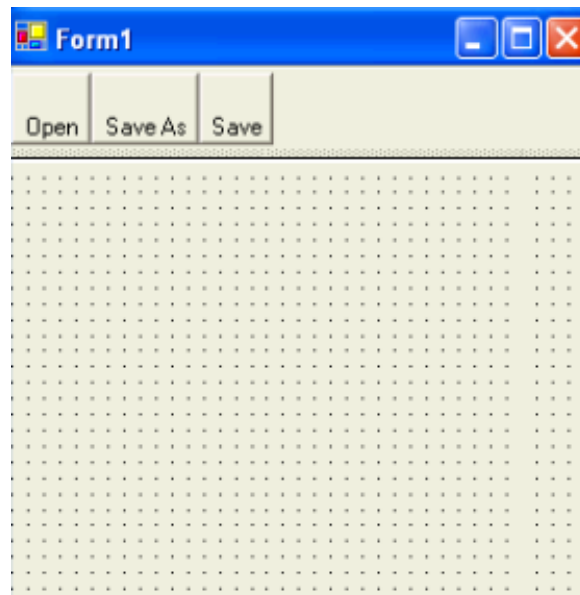
ولها عدة خصائص



ومن خاصية Buttons يمكن تخزين مجموعة من Buttons

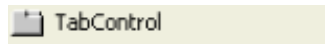


ويكون شكلها النهائي على هذه الصورة

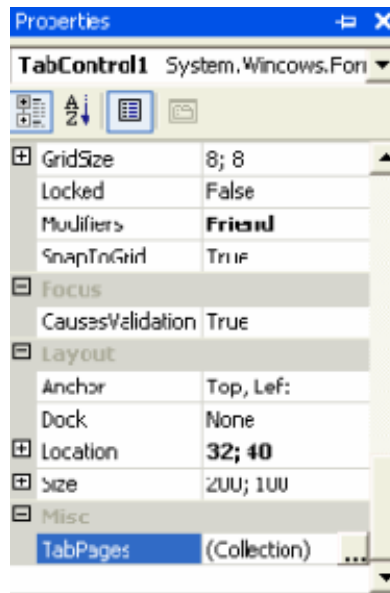


TabControl

تستخدم هذه الأداة لتسمح للمستخدم بأن يقوم بعملية access على أكثر من محتوى screens



ولها عدة خصائص



مثال توضيحي على الشكل النهائي لهذه الأداة

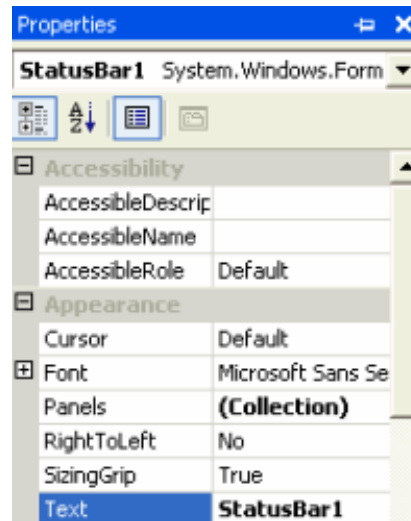


StatusBar

تستخدم في كثير من البرامج لتعمل مثل Help Information في البرنامج

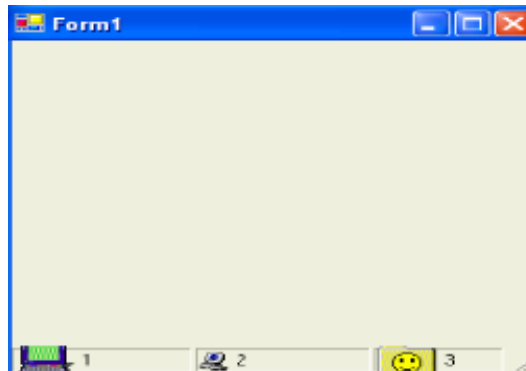


ولها عدة خصائص



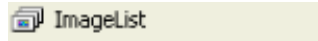
ويجب التأكيد على هذه الخاصية تكون True

ShowPanels True

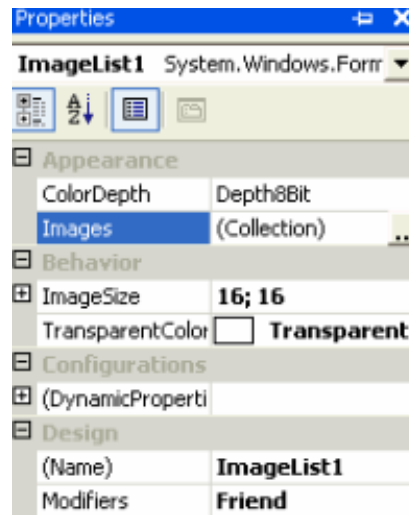


ImageList

تستخدم في كثير من البرامج وخاصة برامج الصور فهي أداة تستخدم لتخزين مجموعة متنوعة من الصور

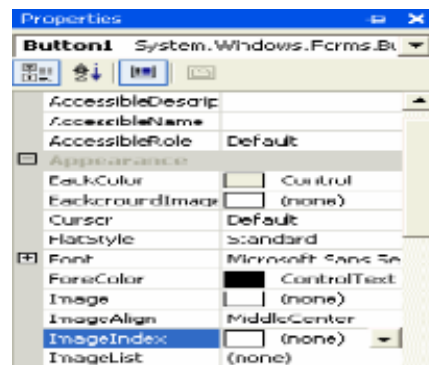


ولها عدة خصائص



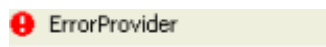
ومن خاصية Images يمكن تخزين مجموعة من الصور

ومن خاصيتي ImageList, ImageIndex يمكن تحديد الصورة لكي تكون خلفية لأي أداة

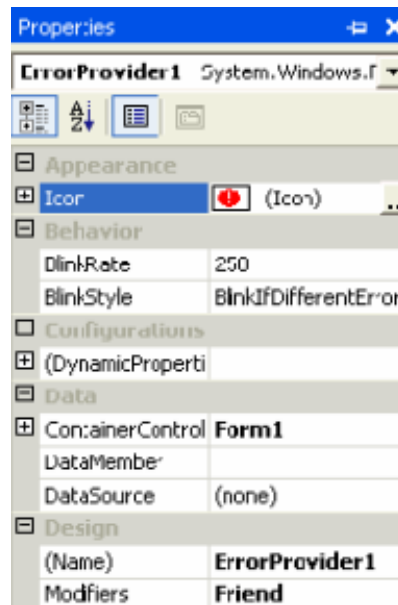


ErrorProvider

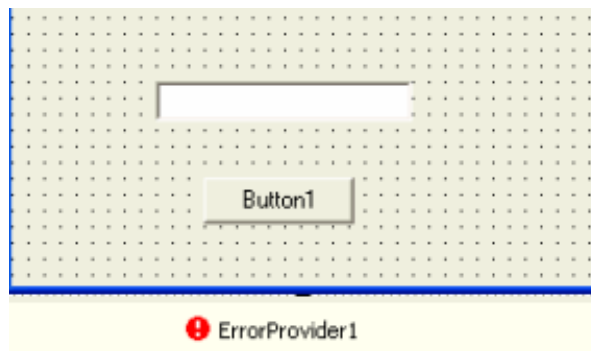
تستخدم هذه الأداة لعمل إنذار alert للمستخدم User



ولها عدة خصائص



مثال توضيحي



```
Private Sub Button1_Click(ByVal sender As System.Object)
    If TextBox1.Text = "محمدى" Then
        ErrorProvider1.SetError(TextBox1, "You Won")
    Else
        ErrorProvider1.SetError(TextBox1, "")
    End If
End Sub
```



الحمد لله

كلام سهل وبسيط ولكنه بداية مربحة

أتمنى خير الدعاء