

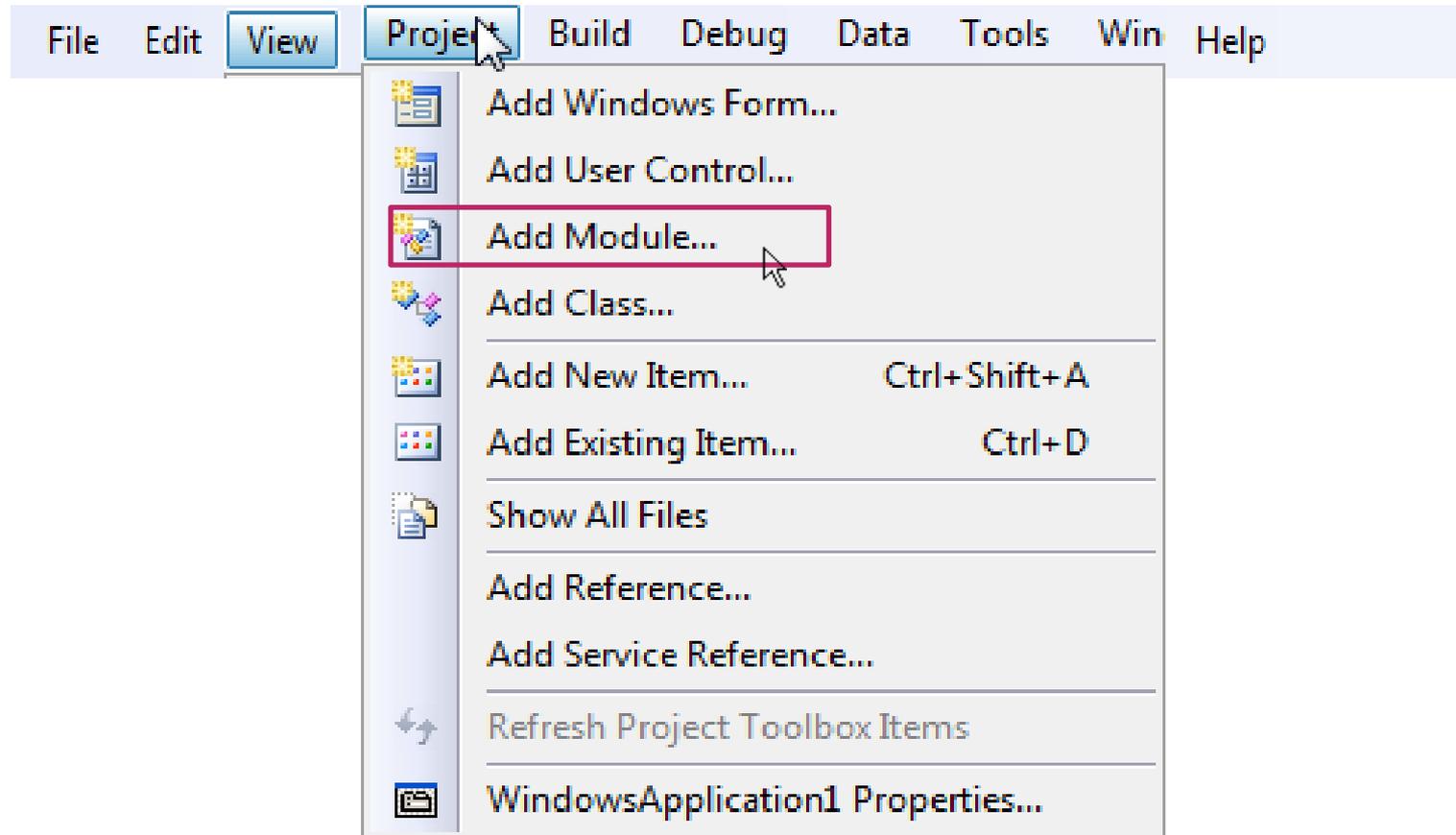
الاجراءات والدوال

ملفات البرمجة (وحدات نمطية) Module

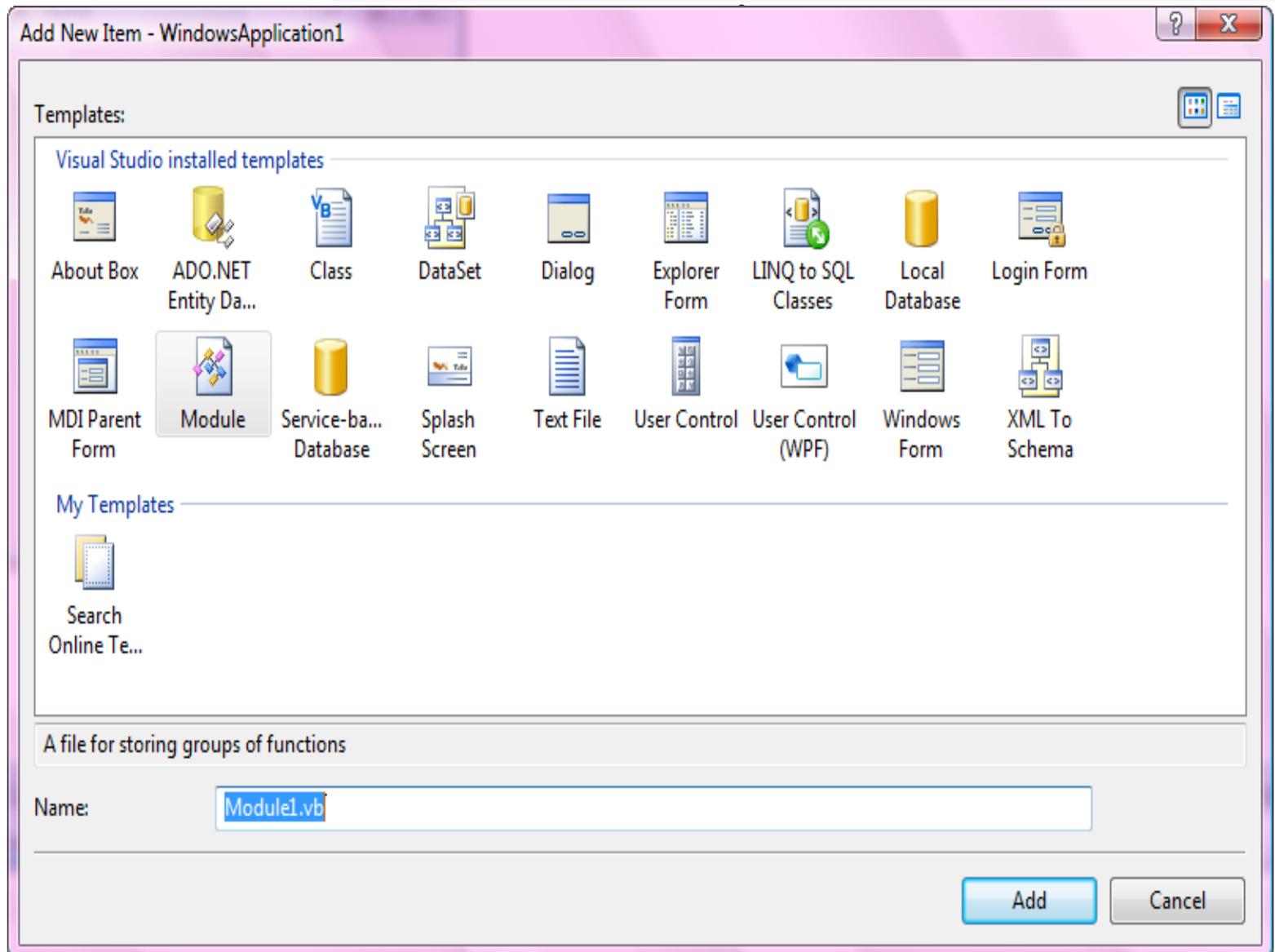
- ❖ هي ملفات مستقلة بها أوامر البرمجة يمكن استدعاؤها من أماكن أخرى داخل البرنامج لتنفيذها .
- ❖ يتم انشاؤها عندما يتم استدعاء الاوامر من قبل العديد من النماذج .
- ❖ إذا تم استدعاؤها من نموذج واحد فمن الأفضل وضعها في ملف النموذج نفسه **Form Module** (**D-click** على النموذج) .
- ❖ كل مجموعة أوامر توضع في اجراء .
- ❖ نستخلص أن ملف البرمجة هو تجميع لعدة إجراءات معاً كل اجراء يقوم بمهمة ما .

Project → **Add Module** → **Add New Item** مربع حوار

ملفات البرمجة Module



إنشاء ملفات البرمجة Code Module

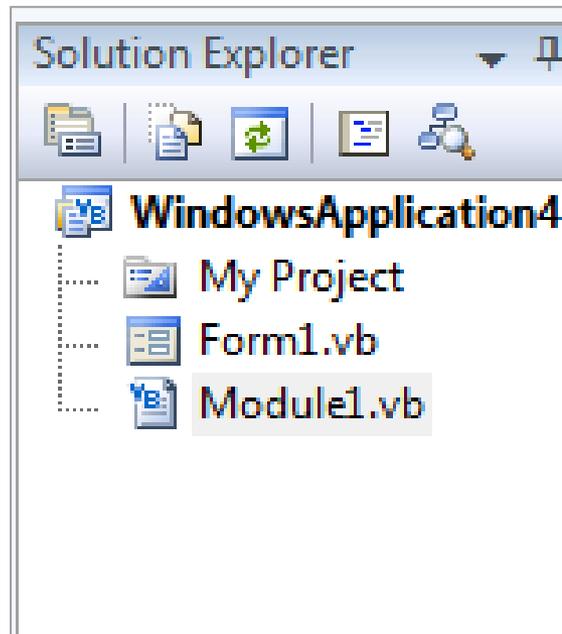
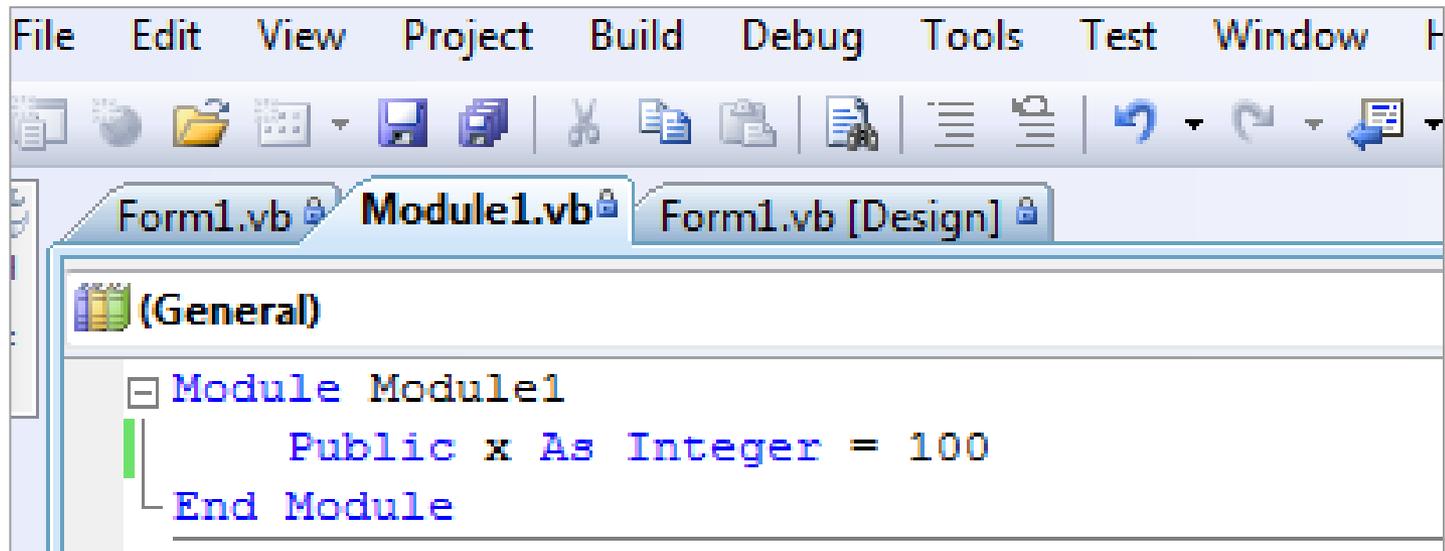


❖ الاعلان عن متغير x فى ال module :

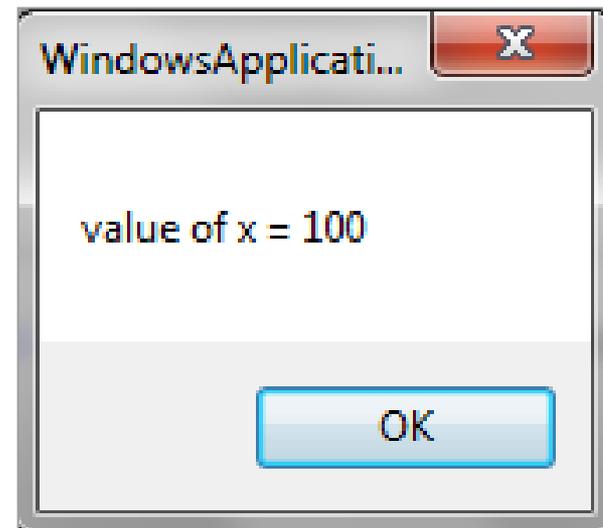
Public x as integer = 100

• طباعة قيمة x فى الحدث form_load





```
Private Sub Form1_Load(ByVal sender As System.  
    MsgBox("value of x = " & x)  
End Sub
```



تعريف الإجراء

- ❖ مجموعة من الأوامر المرتبطة معا لاداء مهمة معينة .
- ❖ قد يكون قصير لا يزيد عن سطر واحد أو يمتد لعدة صفحات ، تستدعي في أي مكان بكتابة اسم الاجراء .

أنواع الإجراءات

- ❖ إجراءات تعود بقيمة (دالة **Function**)
- ❖ إجراءات لا تعود بقيمة (روتين فرعي **Subroutine**)

إستدعاء الإجراء

- ❖ باستخدام كلمة **Call** لإستدعاء الإجراء الفرعي أو نستدعيه بدونها.
- ❖ الدالة لا يمكن استدعاؤها بكلمة **Call** .

الإجراءات التي لا تعود بقيمة

Public /Private / . . . Sub **procedurename** ([parameters])

-
-
-

End Sub

حيث

- ❖ Public / Private تحدد مدى الإجراء .
- ❖ Sub للإعلان عن إجراء لا يعيد قيمة .
- ❖ Subname اسم الإجراء وله نفس شروط تسمية المتغير .
- ❖ () يمرر أو لا يمرر من خلالها القيم (الوسائط Parameters) .
- ❖ المعاملات parameters اختيارية حسب حاجة البرنامج

فائدة إنشاء الإجراءات

- ❖ تقليل الأخطاء : في كل مرة تكرر الأوامر يتم تكرار الأخطاء
- ❖ الثبات وقابلية الصيانه : عند تكرار الأوامر في أماكن كثيرة من الممكن نسيان أحد الأماكن وتصحيح الخطأ بها .
- ❖ الوقت والجهد .
- ❖ تساعد علي تنظيم الاوامر في module.

مثال

❖ أكتبي إجراء (SUM) لجمع عددين حيث يتم تمرير عددين
لحساب مجموع العددين وطباعته ...

```
Public sub SUM (byref X as integer , byref Y as  
integer )
```

```
    Dim Z as integer
```

```
    Z=X+Y
```

```
    msgbox (Z)
```

```
End sub
```

❖ إعادة للمثال السابق دون تمرير معاملات

```
Public sub SUM ( )  
    Dim z , x , y as integer  
    x=200      // x=t1.text  
    y=400     // y=t2.text  
    Z=X+Y  
    msgbox (Z)  
End sub
```

تعريف إجراءات تعيد قيمة (الدالة)

Scope **Function** **function-name** (**parameters**) as **data tybe**

·
·
·

End function

الفرق بين تعريف الدالة والروتين الفرعي

- ❖ استخدام (**Function**) بدلاً من (**Sub**) .
- ❖ تحديد نوع القيمة المرجعة من الدالة بعد القوسين .
- ❖ End function
- ❖ للرجوع بالقيمة نستخدم return

مثال

❖ دالة لحساب طول سلسلة حرفية ممررة اليها ...

```
Public function computelength(byval strtext as string) as integer
```

```
    return strtext.length
```

```
End function
```

التحليل

- ❖ بعد القوسين تم ذكر (**as integer**) وهي القيمة التي ستعود بها .
- ❖ **Return** تقوم بمهمتين :
 - * إنهاء عمل الدالة ولا يتم تنفيذ أي جملة أسفلها.
 - * إعادة القيمة الموجودة بعدها للأمر الذي استدعى الدالة .
- ❖ **End function** لإنهاء الدالة .

إستدعاء الإجراءات

إستدعاء إجراء لا يعود بقيمة

- ❖ بذكر إسم الإجراء مباشرة أو مسبقاً بكلمة (**Call**) .
- ❖ مثال :

Sum (10,20)

إستدعاء إجراء يعود بقيمة

- ❖ بذكر إسم الدالة مسبقاً بإسم المتغير الذي يستقبل القيمة الراجعة.
- ❖ مثال :

Dim X as integer

X= computelength ("Dammam university")

❖ إستدعاء الدالة يوضع على يمين العلامة = أو داخل تعبير
If (computelength ("Dammam university") =20) then

تمرير الوسائط Parameters

تستخدم الوسائط للسماح للأمر الذي يستدعي الإجراء بتمرير بعض البيانات ليتم استخدامها في القيام بمهمة معينة .

تعريف الوسائط Parameters

- ❖ إسم الوسيط .
- ❖ كلمة **as** .
- ❖ نوع البيان .

```
Public sub Myproc ( strc as string )
```

تعريف عدة وسائط لإجراء واحد

```
Public sub Myproc ( strc as string , x as integer )
```

❖ عند الاستدعاء لابد من تمرير قيم بنفس عدد ونوع الوسيط وبنفس الترتيب المستخدم في الإعلان عن الإجراء .

تمرير قيم صريحة للإجراء

```
Myproc ("this is a string", 11)
```

تمرير متغيرات للإجراء

```
Dim s as string=t1.text  
Dim y as integer=100  
Myproc (s, y)
```

❖ في الإجراء (**Myproc**) تم تمرير الوسائط بالقيم **Byval** وليس بالإشارة **Byref**.

الفرق بين تمرير الوسائط بالقيم ، بالإشارة

التمرير بالإشارة	التمرير بالقيمة
يتلقى الإجراء البيانات فإذا قام الإجراء بالتعديل فإنه يؤثر على المتغير الأصلي	يتلقى الإجراء نسخة من البيانات فإذا قام الإجراء بتعديل هذه النسخة فإن ذلك لن يؤثر على المتغير الأصلي

إذا لم يتم كتابة نوع التمرير فإن Visual 2005 يعتبره تمرير بقيمة

مثال

```
Private sub zz (byref x as integer)
    x=x+10
    debug.write(x)
End sub
```

الخروج من الإجراء

❖ يتم تنفيذ الأوامر داخل الإجراء من البداية إلى النهاية وعند الوصول إلى جملة (**end sub**) أو (**End function**) فإن التحكم يخرج من الإجراء ويعود إلى العبارة التي استدعت الإجراء .

❖ لمغادرة الإجراء قبل الوصول لنهايته والعودة لجملة الاستدعاء يتم استخدام العبارة (**Exit sub**) أو (**Exit function**) .

تجنب الإجراءات المتعاقبة (إستدعاء لا نهائي)

```
Public sub Dosomething ( )  
    call Dosomethingelse ( )
```

```
End sub
```

```
Public sub Dosomethingelse ( )  
    call Dosomething ( )
```

```
End sub
```

عند إستدعاء أحد هذه الإجراءات فإن البرنامج يدخل في حلقة لا نهائية.