

العمل مع المجموعات وفضاء الأسماء

العمل مع مجموعة الكائنات

المجموعات **Collections** هي طريقة فعالة للتحكم بالكائنات والبيانات الأخرى في **VB.Net**. الكائنات الموجودة على النموذج يتم تخزينها مع بعضها في نفس الملف ، ويعتبر **VB.Net** تلك الكائنات كأعضاء في مجموعة واحدة تسمى **Controls** "أدوات التحكم" وهي جزء من فضاء الأسماء **system.collection** الذي تزوده المكتبة **.NET Framework**. يتم إنشاء هذه المجموعة تلقائياً عندما نفتح نموذجاً جديداً ، وعندما نضيف كائنات إلى النموذج ستصبح جزءاً من تلك المجموعة تلقائياً .

في الواقع يحتوي **VB.Net** على عدة مجموعات قياسية من الكائنات يمكنك استعمالها عند كتابة برامجك . كل مجموعة في البرنامج لها اسم خاص بها كي تستطيع استعمالها كوحدة مستقلة في شفرة البرنامج .

استخدام كائنات إحدى المجموعات

يمكنك استخدام الكائنات الموجودة في مجموعة أو أعضاء فريدين في المجموعة بتحديد فهرس الكائن في المجموعة يخرن **VB.Net** كائنات المجموعة في عكس الترتيب الذي أنشأها به ، لكي تتمكن من استعمال "ترتيب ولادة" كل كائن لتتمكن من استخدامه فردياً ، أو يمكنك استعمال حلقة للتنقل بين عدة كائنات .

مثلاً لتعريف آخر كائن أنشأته على نموذج حدد الفهرس صفر .

مثل **Controls(0).Text = "Business"**

أما الكائن ما قبل الأخير له الفهرس 1 الخ

مثال :-

تستعمل حلقة **for next** التالية مربع رسالة لإظهار أسماء آخر أربع أدوات أضيفت إلى نموذج

```
Dim i As Integer
For i = 0 To 3
MsgBox (Controls (i) .Name)
Next
```

كتابة الحلقات **for each next**

بالرغم من انه يمكنك استخدام أعضاء المجموعة فردياً إلا أن الطريقة الأكثر فائدة للعمل مع الكائنات الموجودة في مجموعة ما هي بمعالجتها كمجموعة .

السبب الذي يجعل المجموعات تتواجد هو لكي تتمكن من معالجة مجموعات من الكائنات بفعالية مثلاً قد تريد إظهار أو نقل أو فرز أو تغيير اسم أو تغيير حجم مجموعة من الكائنات دفعة واحدة .

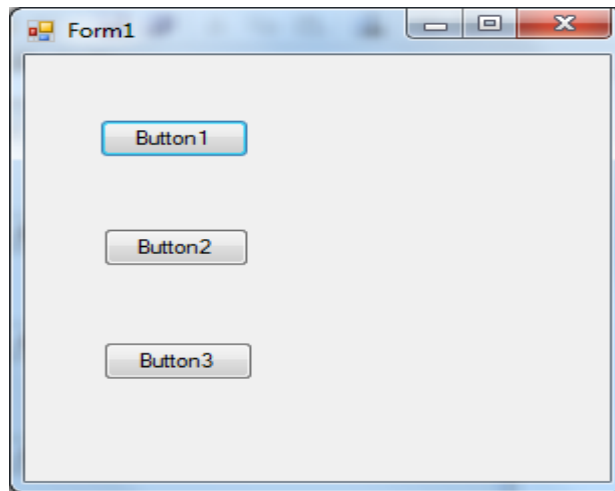
لتنفيذ إحدى تلك المهام يمكنك استعمال حلقة `for each next` للتنقل بين كائنات المجموعة كائناً تلو الآخر .

الصيغة العامة لحلقة `for each next` مع المجموعة

```
Dim i As Control
.....
For Each i In Controls
process object
Next
```

المتغير `i` مصرح عنه من النوع `Control` ويمثل الكائن الحالي في المجموعة `Control` تستعمل الحلقة لمعالجة الكائنات الفردية في المجموعة مثلاً قد تريد تغيير الخاصية `Visible` أو `Enabled` أو `Left` التابعة للكائنات الموجودة في المجموعة أو قد تريد سرد اسم كل كائن في لائحة .
مثال :

أنشئ مشروعاً باسم `my controls collection` وبه ثلاثة أزرار على الجهة اليسرى كالتالي



اضبط خاصية `name` للزر `button3` من نافذة الخصائص إلى `btnMoveObjects` صرح عن المتغير `ctrl` تحت الجملة `Public Class Form1` اعلى النموذج

على النموذج

```
Public Class Form1
'Declare a variable of type Control to
'represent form controls
Dim ctrl As Control
```

انقر الزر **button1** واكتب به الكود التالي :

```
For Each ctrl In Controls
    ctrl.Text = "Click Me!"
Next
```

حيث تقوم هذه الحلقة بتغيير الخاصية **text** لكل كائن في النموذج إلى **Click Me!**

انقر الزر **button2** واكتب به الكود التالي :

```
For Each ctrl In Controls
    ctrl.Left = ctrl.Left + 25
Next
```

تقوم هذه الحلقة بنقل الكائنات الى اليمين 25 بكسلاً

ولنقل الكائنات 25 بكسلاً الى اليسار اطرح 25 بدلاً من الاضافة

ملاحظة:-

كلما تنقر الزر **button2** تتحرك الكائنات إلى اليمين

استعمال الخاصية name في حلقة for each next

إذا أردت معالجة عضو واحد أو أكثر من مجموعة بشكل مختلف عن معالجتك الأعضاء الآخرين، يمكنك استعمال الخاصية

name التي تعرف بشكل فريد كل كائن موجود على النموذج .

انقر الزر **button3** واكتب به الكود التالي :

btnMoveObjects لهذا الزر تغيرت إلى **name** الخاصية

```
For Each ctrl In Controls
    If ctrl.Name <> "btnMoveObjects" Then
        ctrl.Left = ctrl.Left + 25
End If
Next
```

ملاحظة:-

كلما نقرت الزر **button3** يتحرك الزران العلويان على النموذج لكن الزر الثالث يبقى في نفس المكان .

إنشاء مجموعات خاصة بك

يتيح لك **VB.Net** إنشاء مجموعاتك الخاصة لتعقب البيانات في البرنامج وللتلاعب بها بطريقة نظامية .

على الرغم من أن المجموعات تنشأ في اغلب الأحيان لتخزين كائنات ، يمكنك استعمالها أيضاً لتخزين قيم رقمية

أو نصية أثناء التشغيل .

تصريح مجموعات جديدة

يتم تصريح المجموعات الجديدة مثلما يتم التصريح عن المتغيرات في البرنامج ، والمكان الذي تصرّحها فيه يحدّد مداها ، أو الامتداد الذي تستمر فيه قيمها بالتواجد .

ولان المجموعات مفيدة جداً فإننا نصرح عنها عادة في أعلى النموذج أو في وحدة نمطية .

الصيغة العامة للتصريح عن مجموعة جيدة

Dim collectionname As New Collection

حيث collectionname هو إسم مجموعتك . إذا وضعت تصريح المجموعة في وحدة نمطية ، استعمل الكلمة الأساسية Public بدلاً من الكلمة الأساسية Dim .

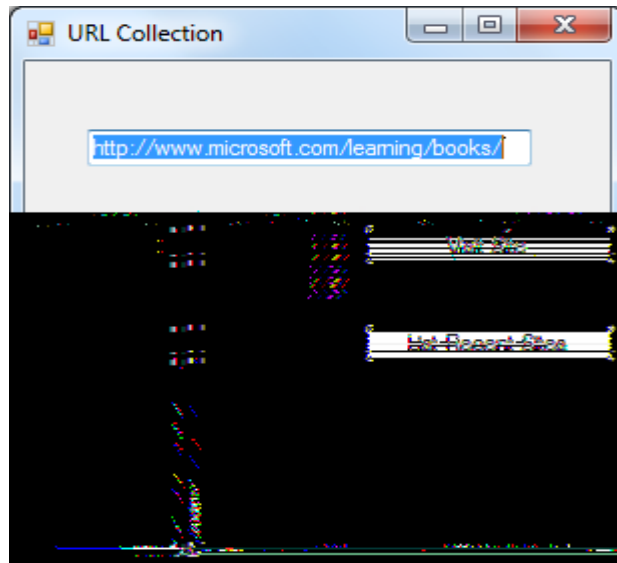
بعد أن تنشئ مجموعة ، يمكنك إضافة أعضاء إليها باستعمال الطريقة add ، ويمكنك فحص الأعضاء الفرديين باستعمال حلقة for each next

مثال : تعقب عناوين الانترنت باستعمال مجموعة جديدة

1. أنشي مشروعاً جيداً باسم My URL Collection .
2. أدرج على النموذج مربع نص وزري أمر .
3. اضبط الخصائص التالية للنموذج وكائناته .

الإعداد	الخاصية	الكائن
http://www.microsoft.com/learning/books	Text	TextBox1
Visit Site	Text	Button1
List Recent Sites	Text	Button2
URL Collection	Text	Form1

سيظهر النموذج بعد ضبط خصائصه كما يلي :



انتقل إلى محرر الشفرة ثم اكتب الكود البرمجي التالي

1. التصريح عن مجموعة جديدة باسم `URLsVisited` تستخدم على مستوى النموذج

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

2. انقر على الزر `Visit Site` ثم اكتب الشفرة التالية

```
URLsVisited.Add(TextBox1.Text)
System.Diagnostics.Process.Start(TextBox1.Text)
```

تقوم الشفرة اعلاه بالاتي

الطريقة `add` تقوم بتعبئة المجموعة بأعضاء

يفترض البرنامج أن هناك عنوان انترنت صالح قد تم وضعه في الكائن `TextBox1` كسلسلة إلى المجموعة

`URLsVisited` بعدها يتم استدعاء الطريقة `System.Diagnostics.Process.Start`

مع استخدام العنوان الموجود في مربع النص كعامل لها حيث تقوم الطريقة `Start` بفتح العنوان باستخدام مستعرض الويب الافتراضي في الكمبيوتر .

3. انقر على الزر `List Recent Sites` ثم اكتب الشفرة التالية

```
Dim URLName As String = "", AllURLs As String = ""
For Each URLName In URLsVisited
AllURLs = AllURLs & URLName & vbCrLf
Next URLName
```

```
MsgBox(AllURLs, MsgBoxStyle.Information, "Web sites visited")
```

يقوم الكود أعلاه بعرض كافة محتويات المجموعة التي تمت زيارتها