# الفصل التاسع

# تابع مفاهيم OOP

## Programming Concepts في هذا الفصل

9

في هذا الدرس نتسكمل شرح مفاهيم OOP وذلك من خلال النقاط التالية :

- إضافة خصائص properties للفصيلة
- ما هى الخاصية Polymorphism
- ما هى NameSpace
- كيفية إنشاء NameSpace
- ما هى مكتبة الفصائل Class Lib
- كيفية إنشاء مكتبة فصائل Class Lib
- properties
- تحديد الخاصية الافتراضية default propority

## إضافة خصائص properties للفصيلة

من الإمكانيات المتاحة عند التعامل مع الفصائل هى إمكانية إضافة خصائص Properties للفصيلة class مثل التى نتعامل معها عند التعامل عند التعامل مع الأدوات tools ويوفر Visual Studio.Net ذلك بسهولة ولتوضيح ذلك تابع الخطوات التالية:

### فى لغة VB.NET

1- قم بانشاء تطبيق جديد من النوع ConsoleApplication كما سبق

2- قم بانشاء فصيلة Person كما سبق وكما فى السطور التالية:

```
Public Class person
Private Pname, Paddress As String
End Class
```

### فى لغة C#

```
public class person
  {
     private string name;
  }
```

فى هذه السطور:

تم إنشاء فصيلة جديدة Class بالاسم person وهذه الفصيلة class تحتوى على المتغيرين Pname,Paddress والمطلوب إنشاء خاصية بالاسم Name للتعامل مع المتغير Pname بحث يتم تسجيل قيمة هذا المتغير وأخذ قيمة هذا المتغير من خلال هذه الخاصية ولتحقيق ذلك قم بكتابة سطر تعريف خاصية جديدة بالاسم Name كما فى السطر التالى:

```
Public Property Name()
```

- فى هذا السطر يتم الإعلان عن خاصية بالاسم Name باستعمال الكلمة المحجوزة Property.
- وبمجرد كتابة هذا السطر يقوم Visual Basic .Net بكتابة باقى السطور المطلوبة لإنشاء الخاصية كما فى السطور التالية:

```
Public Class person
Public Pname, Paddress As String
Public Property Name()
Get
```

```
End Get
Set(ByVal Value)
End Set
End Property
End Class
```

**فى لغة C#**

```
public string Name
    {
        get
        {
            return name;
        }
        set
        {
            name = value;
        }
    }
```

فى هذه السطور:

قام Visual Basic.Net باضافة أجزاء الخاصية Name تلقائيا حيث اضيفت الدالة Get التى تنفذ عندما تريد أخذ قيمة الخاصية أى عند تقوم باستعمال الخاصية وتكتب السطر:

```
R=Obj.Name
```

فى هذه الحالة أنت تطلب أخذ قيمة الخاصية Name لذلك تنفذ الدالة Get وعليك كتابة الاوامر التى تحقق ذلك وهى ببساطة كما فى السطر التالى:

```
Return Pname
```

فى هذا السطر يتم اعادة قيمة المتغير Pname وهو المطلوب فى حالة Get.

وفى حالة تسجيل قيمة المصفوفة تنفذ الدالة Set والتسجيل يتم كما فى الصورة التالية:

```
Obj.Name="omr"
```

فى هذه الحالة تطلب تسجيل قيمة الخاصية Name بالقيمة omr لذلك نكتب فى الدالة Set السطر التالى:

```
Pname=value
```

وهذا معناه ان القيمة omr توضع تلقائيا فى المتغير Value الذى نضع قيمته فى المتغير Pname وبالتالى تصبح سطور الخاصية داخل الفصيلة ؛ كاملة كما فى السطور التالى:

```
Public Class person
    Public Pname, Paddress As String
    Public Property Name()
      Get
        Return Pname
      End Get
      Set(ByVal Value)
        Pname = Value
      End Set
    End Property

  End Class
```

**فى لغة C#**

```
public class person
  {
    private string name;

    // A read-write instance property:
    public string Name
    {
      get
      {
        return name;
      }
      set
      {
        name = value;
      }
    }
```

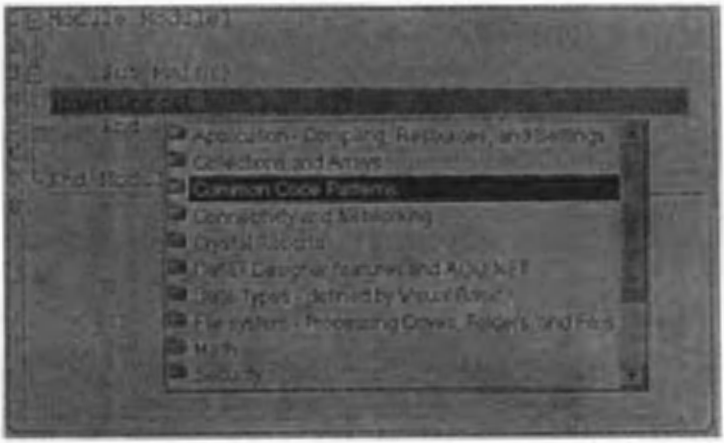
من فضلك قم بمراجعة سطور الفصيلة class وكذلك سطور تعريف خاصية Property بالاسم Name وحاول تكرار ذلك وإنشاء خاصية بالاسم address للتعامل مع المتغر Paddress.

## قطع الاكواد الجاهزة : Snippet

1- اضغط طرف الماوس الأيمن داخل صفحة الأكواد تحصل على قائمة كما فى الشكل (1-9).

الشكل (9-1)

اختر Insert snippet كما فى الشكل السابق تحصل على قائمة بمجموعات الأكواد التى كما يوجد فى الشكل (9-2).

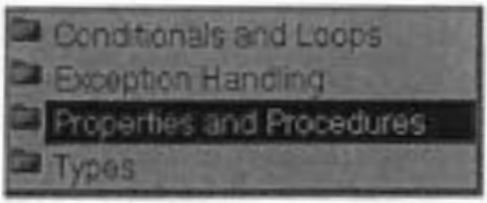


الشكل (9-2)

كما فى الشكل(9-3).

الشكل (9-3)

2- فى هذا الشكل توجد مجموعات الأكواد الأساسية مثل جمل الشرط والتكرار كما يوجد فى الشكل (4-9).




الشكل (4-9)

3- وعند اختيار properties and Procedur تحصل على أنواع كما فى الشكل (5-9).

الشكل (5-9)

اختر تعريف خاصية NewProperty تحصل على الأكواد كما فى السطور.

```
Private newPropertyValue As Integer
   Public Property NewProperty() As Integer
      Get
         Return newPropertyValue
      End Get
      Set(ByVal value As Integer)
         newPropertyValue = value
      End Set
   End Property
```

في هذه السطور:

تم الإعلان عن الخاصية مع تحديد دوال تسجيل Set وإعادة القيمة Get.

## ما هي NameSpace

هي شكل تنظيمي يمكن أن يحتوى على أكثر من فصيلة Class ويمكن للـ NameSpace أن يحتوى على أكثر من NameSpace ويتم فتح هذه المكتبات NameSpace باستعمال كلمة Imports كما يحدث في أول البرنامج حيث تلاحظ وجود السطر.

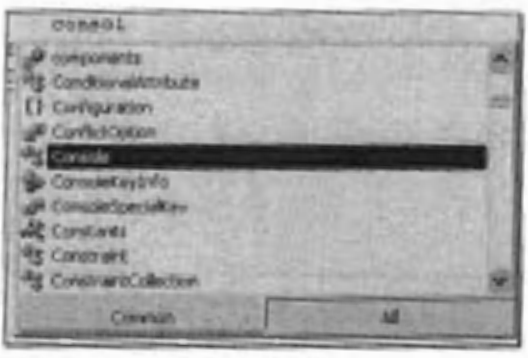
```
Imports System
```

## في لغة C#

```
using System;
```

فهذا السطر أمر الى المترجم بأن يقوم باستيراد ملف الفصائل Classes المسمى system لأننا سوف نستعمل بعض الفصائل المعرفة بداخله وإلا سوف نضطر للاشارة إلى System كلما أردنا التعامل مع أي فصيلة class داخل هذا الـ NameSpace.

ولتوضيح ذلك ارجع إلى بيئة التطوير وداخل محرر سطور الأوامر اكتب .System تلاحظ ظهور الفصائل classes الموجودة به كما في الشكل (6-9).




الشكل (6-9)

وكما أشرنا يمكن للـ NameSpace أن يحتوى على Classes ويمكن أن يحتوى على NameSpaces.

### كيفية انشاء NameSpace

ويتم انشاء NameSpace بسهولة بمجرد كتابة كلمة NameSpace ثم الاسم وفى النهاية يتم انهائها بالامر End NameSpace تابع الخطوات التالية:

1- قم بإنشاء تطبيق جديد من النوع ConsoleApplication.

2- اكتب فى نهاية سطور الأوامر وبعد الأمر End Module سطور تعريف NameSpace بالاسم OurNameSpace كما فى السطور التالية:

```
1. Namespace OurNameSpace
2. Public Class class1
3. Public var1, var2 As String
4. End Class
5. Public Class class2
6. Public var11, var22 As String
7. End Class
8. End Namespace
```

**فى لغة C#**

```
namespace OurNameSpace
{
    public class class1
    {
        public string var1, var2;
    }

    public class class2
    {
        public string var11, var21;
    }
```

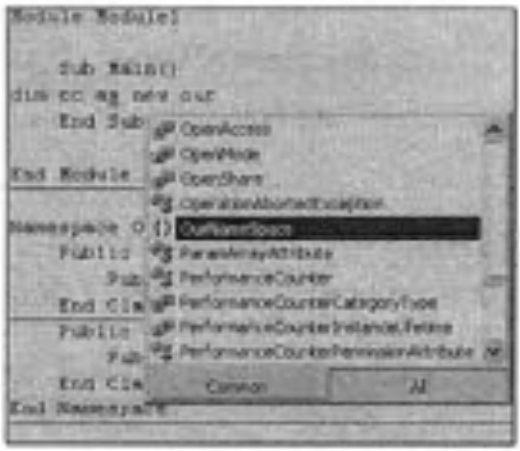
فى هذه السطور :

- فى السطر رقم 1 تم استعمال الأمر Namespace لإنشاء Namespace جديد بالاسم **OurNameSpace**.
- فى السطر رقم 2 تم الإعلان عن فصيلة class جديدة بالاسم class1 .

- فى السطر رقم 5 تم الإعلان عن فصيلة class جديدة بالاسم class2.
- فى السطر رقم 8 تم إنهاء الـ Namespace باستعمال الأمر end Namespace.

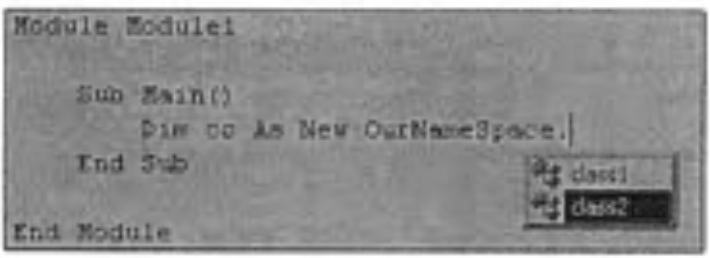
## استعمال الـ Namespace

بعد إنشاء Namespace وإنشاء فصائل classes بداخله المطلوب كيفية استعمال الفصائل classes المعرفة داخل الـ Namespace ويتم ذلك بتعريف هدف Object من الفصيلة class المعرفة داخل Namespace ولكن بعد نسبتها إلى الـ Namespace ولتوضيح عد الى بيئة التطوير وداخل الدالة الرئيسية للبرنامج Sub Main() اكتب تعريف هدف Object تلاحظ ظهر اسم الـ Namespace كما فى الشكل (7-9) .




الشكل(7-9)

- اختر الـ Namespace الخاصة بنا والتى أنشأناها بالاسم OurNameSpace ثم اكتب . تلاحظ ظهور فصائلها كما فى الشكل (8-9) .




الشكل(8-9)

**فى لغة C#**

```
19          person p=new person();
20          p.Name ="Omr";
21      OurNameSpace cc=new OurNameSpace.
22                                  class1   class OurNameSpace.class1
23                                  class2
24  }
```

فى هذا الشكل تلاحظ ظهور فصائل classes الـ Namespace التى أنشاناها ويمكنك التعامل معها بباقى الخطوات كما سبق.

## إنشاء مكتبة الفصائل creating class library

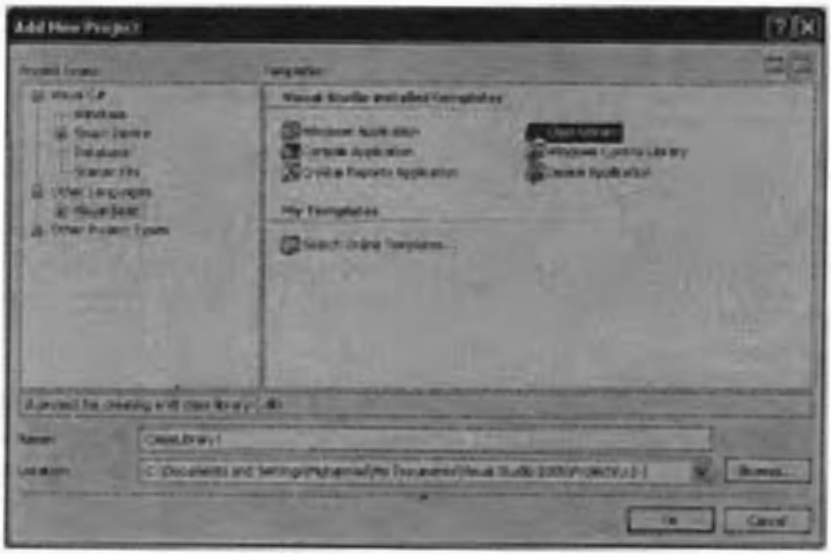
بعد شرح بعض المفاهيم المهمة لمفهوم OOP تعال معنا نشرح عملياً كيفية إنشاء مكتبة الفصائل class Library هى احد انواع التطبيقات التى يوفرها Visual Basic.Net حيث يسمح بانشاء تطبيق عبارة عن ملف بالامتداد DLL(Dynamic Link Lib) يحتوى على مجموعة من الفصائل Classes التى يمكن استعمالها من داخل اى برنامج بعد الإشارة الى ملف المكتبة.

**ملحوظة:**
الملفات ذات الامتداد DLL عبارة عن ملفات مكتبات يتم إنشائها منفصلة ليصلح ربطها بأكثر من برنامج مما يوفر عدم إعادة كتابة سطور محتوياتها أكثر من مرة بل تكتب مرة واحدة ثم تستعمل أكثر من مرة.
وسميت Dynamic لعدم ربطها ربطا نهائيا بالبرنامج بل تربط مؤقتا فقط أثناء التشغيل.

## إنشاء مكتبة فصائل Class library

تابع معنا الخطوات التالية :

1- شغل vb.net ثم انقر (اختر) new project ثم اختر visual basic projects ومن الجانب الأيمن اختر class library ثم اكتب اسماً my class ثم انقر ok كما فى الشكل (9-9) .




الشكل(9-9)

بهذا تم إنشاء مكتبة فصائل class Lib بالاسم ClassLibrary1 وتم تعريف ذلك في ملف بالامتداد VB.

2-قم بتفحص الأوامر الناتجة تلاحظ إنشاء فصيلة بالاسم class1 ليس بها أي شيء ويمكن تعريف أكثر من فصيلة CLASS بهذا الملف ويمكن تجميع أكثر من ملف من هذه الملفات (ملفات الفصائل ) في مجموعة تسمى Assembly .

ويمكن إنشاء فصيلة class بهذا الملف في أي وقت بكتابة ما يلي :

```
class  class_name
.....
end class
```

3- غير اسم الفصيلة class باستبدال السطر class class1 بالسطر class student

4- غير اسم الفصيلة في solution وذلك باختياره بالطرف الأيمن وتغير اسمه إلى student.

بهذا تم إنشاء فصيلة class مع ملاحظة إمكانية إنشاء أي فصيلة في أي وقت داخل هذا الملف بكتابة كلمة class وبعدها اسم الفصيلة في نهاية السطور تغلق الفصيلة بالعبارة end class كما سبق.

5- قم بتعريف متغيرات var وخصائص properties الفصيلة student كما في السطور التالية:

```
Public Class student
    Dim student_no As Integer
    Dim student_name As String
    Public Property stno()
        Get
            Return student_no
        End Get
        Set(ByVal Value)
            student_no = Value
        End Set
    End Property
    Public Property stname()
        Get
            Return student_name
        End Get
        Set(ByVal Value)
            student_name = Value
        End Set
    End Property
End Class
```

**فى لغة C#**

```
public class person
{
public string Pname, Paddress;
public void  SetData( string name_v  ,   string address_v  )
{
Pname = name_v;
Paddress = address_v;
}
public string  getName()
{
return Pname;
}
public string  getAddress()
{
return Paddress;
}
}

public class student: person
{
int degree;
public void  setdegree(int  degree_v)
{
degree = degree_v;
}
```

```
public int  retdegree()
{
return degree;
}

}
```

راجع هذه السطور تلاحظ تعريف متغيرين وتعريف خصائص التعامل مع المتغيرين كما سبق.

6- من القائمة الرئيسية اختر Build ثم Build ClassLibrary1 تلاحظ قيام Visual Basic.Net بترجمة ملف المكتبة والمفترض أننا نحصل على ملف بالاسم Classlibrary1.dll وللتأكد من ذلك من فضلك اذهب الى فهرس هذا المثال حسب مكان حفظك للامثلة تلاحظ وجود الملف داخل الفهرس bin ويمكنك مشاهدة ذلك بمحاولة حفظ أحد ملفات التطبيق كما بالشكل (10-9) .

الشكل (10-9)

## إستعمال مكتبة الفصائل Using class library

بعد إنشاء مكتبة الفصائل ClassLibrary1 وإنشاء فصيلة class بالاسم student داخل هذه المكتبة وبعد القيام بترجمة ملف المكتبة الى ClassLibray1.DLL المطلوب معرفة كيفية استعمال هذه المكتبة، ولتوضيح ذلك تابع الخطوات التالية:

1- قم بإنشاء تطبيق جديد من النوع ConsoleApplication كما سبق.

2- من القائمة الرئيسية أختر Project ثم Add Refrence للإشارة إلى ملف المكتبة classLibrary1.Dll كما فى الشكل (11-9).

الشكل(11-9)

3- يظهر مربع حوار Add Refrence به ثلاث صفحات (tabs) اختر الصفحة Projects ثم اضغط الزر Browse واختر ملف المكتبة ClassLibrary1.DLL كما فى الشكل (12-9) .

الشكل(12-9)

4- اختر الملف ClassLibrary1.dll ثم اضغط الزر Open تعود إلى بيئة التطوير وقد تم الإشارة إلى ملف المكتبة ويمكن استعمال فصائلها.

5- قم بتعريف هدف Object من الفصيلة student داخل الدالة الرئيسية للبرنامج Sub Main() كما في الشكل (13-9).

```
Sub Main()
    Dim a As New ClassLibrary1.
                                student

End Sub
```

الشكل(13-9)

## في لغة C#

```
14        static void Main(string[] args)
15        {
16            //
17            // TODO: Add code to start application here
18            //
19            ClassLibrary1.
20        }                    person
21    }                        student
```

6- قم باستكمال سطور استعمال الفصيلة student وذلك كما في السطور التالية:

```
Sub Main()
    Dim a As New ClassLibrary1.student()
    Dim v1 As Integer
    Dim v2 As String
    a.stname = "OMR"
    a.stno = 125
    v1 = a.stno
    v2 = a.stname
    Console.WriteLine(v1)
    Console.WriteLine(v2)

End Sub
```

**فى لغة C#**

```
14        static void Main(string[] args)
15        {
16
17            ClassLibrary1.person p1=new ClassLibrary1.person ();
18            p1.SetData ("Omr","Cairo");
19            Console.WriteLine (p1.getName());
20            Console.WriteLine (p1.getAddress());
21
22        }
```

7- قم بتنفيذ البرنامج تحصل على القيم التى تم تحديدها وتم طباعتها باستعمال الدالة WriteIn() وبهذا تكتمل خطوات إنشاء واستعمال مكتبة الفصائل Class Library.

### إضافة خصائص creating proprities

نعود إلى الفصيلة السابقة ونرى كيفية إضافة خصائص proprities ويتم ذلك بتعريف متغيرات و تحديد الدوال التي تتعامل معها كما يلي :

قم بتعريف متغير داخل سطور الفصيلة كما يلي :

```
dim  m first name as string
```

يعد تعريف المتغير الذي يمثل الخاصية وهو m first name .

قم بإضافة الدالة ( )get لقراءة قيمة الخاصية (المتغير) وكذلك إضافة الدالة( )set لتسجيل قيمة الخاصية (المتغير ) كما يلي:

```
1: public property first name as  string
2: Get
3: return m first name
4: end Get
5: set(by val   value as string)
6: m first name = value
7: end set
8: end property
```

في هذه السطور :

### اختيار الفصيلة test the class

بعد إضافة فصيلة بالاسم student وتعريف خاصية first name وتم إنشائها إنشاء

دالة لتسجيلها تابع معنا كيفية إنشاء تطبيق لاختبار (استعمال )هذه الفصيلة كما يأتي :

1- اختر قائمة file ثم new ثم project اختر windows apps وأعطيه الاسم test class وتأكد من اختيار أداة الاختيار add to solution وهذا ما يقابل group في vb6 .

2- يتم إضافة التطبيق الجديد إلى التطبيق القديم (class) ويظهرا معاً في نافذة soluation explor.

- انقر تطبيق لاختبار test class بالطرف الأيمن واختر set as start up project من القائمة التي تظهر .
- من نافذة soluation explor لنقر refrance بالطرف الأيمن في برنامج test class يظهر اسم الفصيلة السابق اختارها .
- عد إلى تطبيق test class وأخف زر أمر buttons علية ثم انقرها مرتين للدخول إلى صفحة الأوامر و اكتب السطور التالية:

```
1- Dim NC AS new  my class . student( )
2- NCO FIRST NAME ="OMR"
3- MSG (NC . first name )
4- end sub
```

في هذه السطور:

- في السطر رقم 1 يتم تعريف المتغير NC كهدف جديد للفصيلة Student المعرفة في المكتبة my class.
- في السطر رقم 2 تم تسجيل الخاصية first name للهدف الجديد NC المأخوذ من الفصيلة STUDENT وذلك لتسجيل العبارة الحرفية OMR .
- في السطر رقم 3 يتم عرض قيمة الخاصية first name بعد تسجيلها لاحظ الكتابة المكتوبة الموجود بها الفصيلة student هي my class وبدون كتابة اسم المكتبة تظهر رسالة خطأ تعني عدم التعرف على الفصيلة student ويمكن التعويض عن اسم المكتبة باستعمال الأمر import كما يلي :

```
1- import my class
2- dim NC as new student
3- NC .first name ="OMR"
4- MSGBOX (NC .first name)
```

في هذه السطور:

- في السطر رقم 1 استيراد المكتبة my class باستعمال الأمر import .
- في السطر رقم 2 تم تعريف المتغير NC من نوع الفصيلة student وفي هذه الحالة لم نحتاج لكتابة اسم المكتبة my class وباقي السطور كما سبق .

### إنشاء خصائص للقراءة فقط أو للكتابة فقط read-only and write-only

في بعض الأحيان نحتاج للإنشاء الخاصية للقراءة فقط بحيث يمكن معرفة قيمة الخاصية ولا نسجلها و أحياناً نحتاج للعكس .

ولإنهاء خاصية read-only يتم عبارة read-only أمام الخاصية وكذلك يتم حذف دالة set التي تكتب في الخاصية ويتم ذلك فيما يلي:

```
1: public read only property first name ( ) as string
2:  Get
3: return m first name
4: end get
5: end property
```

في هذه السطور:

- في السطر رقم 1 تم وضع العبارة read-only للإشارة أن هذه الخاصية للقراءة فقط .
- ومن السطر رقم 2 إلى السطر رقم 4 تم إنشاء دالة Get التي تقرأ قيمة الخاصية وتم حذف دالة set وبالتالي أصبحت الخاصية للقراءه ولا يسمح بكتابة قيمه بها .
- بالمثل يمكن عمل العكس وذلك بإنشاء خاصية للكتابة وليس للقراءة وذلك بكتابة دالة set و إلغاء دالة Get وكذلك كتابة العبارة write-only.

### إضافة دالة بناء للفصيلة add constructor

كما شرحنا من قبل أن دالة البناء constructor يتم إنشائها في الفصيلة لتحقيق العمليات الأولية فمثلاً في الفصيلة student التي أنشأناها نريد إرسال كود الطالب عند تعريف الهدف (متغير) من الفصيلة يتم ذلك بإنشاء دالة CONSTRUCTOR وفي لغةVBيتم إنشاء دالة البناء بإنشاء دالة بالاسم NEW( ) كما في السطور التالية:

```
Public sub new (optional by val sid as intege =a)
Stno =sid
End sub
```

في هذه السطور يتم إنشاء دالة البناء بالاسم ( )sub new وفيها يتم تسجيل قيمة المتغير stno المفترض تعريفة بالفصيلة وذلك بقيمة المتغير sid وبالتالي عند التعامل مع الفصيلة وتعريف متغير (هدف) يتم ذلك كما يلي :

```
dim NS as new student (1)
```

في هذه الحالة تم تعريف المتغير NS مع إرسال القيمة 1 كمعامل وبالتالي تأخذه دالة البناء وتسجله في المتغير STNO لمجرد تعريف المتغير .

## إضافة دوال أعضاء للفصيلة Adding methods

كما شرحنا أن الفصيلة تتكون من بيانات (خصائص) ودوال أعضاء methods التي تحدد عمليات الفصيلة ، ولدوال الأعضاء مثل أي دالة يتم إنشائها داخل الفصيلة لتحقيق عمليه معينه مع الفصيلة .

ويتضح ذلك من السطور التالية:

- قم بكتابة الدالة التالية داخل سطور الفصيلة .

```
public function add student ( ) as BOO learn
  add new student to data basic
return true
end function
```

- بمجرد كتابة هذه الدالة داخل سطور الفصيلة تصبح دالة عضوه في الفصيلة وبالتالي استعمال واستدعائها تسجيل بيانات طالب ويتم استدعائها بالصور التالية:

```
dim NS AS NEW STUDENT( )
NS .Add student   ( )
```