القسم السادس

الفصل الخامس عشر

انشاء مكتبات
Class Lib,Custom Control

Programming Concepts فى هذا الفصل

15

فى هذا الفصل نتناول موضوع من الموضوعات المتقدمة وهو انشاء مكتبات فصائل لتطبيقات النوافذ Windows Applications والتى تسمى Class Library وكذلك انشاء أدوات Tools جديدة وذلك من خلال النقاط التالية :-

- ما هى المكتبة class Lib
- انشاء تطبيق مكتبة Class Lib بسيط
- انشاء تطبيق Windows App يستخدم المكتبة Class Lib1
- انشاء تطبيق مكتبة Class Lib2 متقدم
- انشاء تطبيق Windows App يستخدم المكتبة Class Lib2
- ما هى Custom controls
- انشاء تطبيق أداة Custom Control بسيطة MyTimer
- انشاء تطبيق يستعمل الاداة الجديدة MyTimer
- انشاء تطبيق أداة Custom Control متقدمة LoginDialog
- انشاء تطبيق يستعمل الاداة الجديدة LoginDialog

ما هى المكتبة class Lib

من المفاهيم المتقدمة هو إمكانية إنشاء مكتبة فصائل class lib تحتوى على مجموعة من الفصائل classes كثيرة الاستعمال بالنسبة لك أو بالنسبة لفريق العمل ووضعها فى مكتبة تترجم لل ملف بالامتداد dll (Dynamic Link Lib) ثم استعمال هذه المكتبة من قبل أى برنامج جديد لك أو لفريق البرمجة وهذا الأسلوب يوفر الكثير حيث يوفر اعادة كتابة هذه الفصائل classes أكثر من مرة ، كما يوفر إعادة ترجمة هذه الفصائل أكثر من مرة وتصحيح الأخطاء حيث تقوم بترجمتها مرة واحدة وتصحيح الاخطاء مرة واحدة ثم استعمالها كما تشاء كما أنها توفر صورة غير نصية من المكتبة DLL بحيث يمكن استعمالها دون الاطلاع على نص الأوامر أو التعديل فيها

انشاء تطبيق مكتبة Class Lib بسيط

لتوضيح هذه الفكرة سوف نقوم بانشاء مكتبة فصائل class lib بسيطة تحتوى على فصيلة واحدة class بها دالة Method واحدة للتجربة فقط على نقوم باعداد مكتبة فصائل class Lib متقدمة فيما بعد ولتوضيح ذلك تابع الخطوات التالية:

1-قم باعداد تطبيق جديد من النوع ClassLibrary كما فى الشكل (1-15)

شكل (1-15)

2- اترك الاسم الافتراضى Classlibrary1 كما هو ثم اضغط الزر Ok تحصل على ملفات تطبيق المكتبة وبها الملف class1.vb الذى يحتوى فصيلة class بالاسم class1مفتوحة للكتابة.

3- اكتب تعريف دالة Method بسيطة لعرض رسالة تفيد انها تستدعى من مكتبة الفصائل ClassLib1 كما فى السطور الموضحة فى الشكل (2-15).

فى لغة VB.NET

```
Public Class Class1
    Public Sub Hello()
        MsgBox("Hello from Classlibrary1")
    End Sub
End Class
```

شكل (2-15)

فى لغة C#

```
1  using System;
2
3  namespace ClassLibrary2
4  {
5      /// <summary>
6      /// Summary description for Class1.
7      /// </summary>
8      public class Class1
9      {
10         public Class1()
11         {
12             //
13             // TODO: Add constructor logic here
14             //
15         }
16         public string Hello()
17         {
18     return "Hello";
19         }
20
21     }
22 }
23
```

فى هذه السطور :

- فى السطر رقم 1 تم تعريف فصيلة class بالاسم Class1.
- فى السطر رقم 2 تم تعريف برنامج فرعى من النوع Sub (Method) كدالة عضوة للفصيلة Class1.
- فى السطر رقم 3 تم استدعاء الدالة ()Msgbox وبين أقواسها رسالة تفيد انها تعرض من مكتبة الفصائل ClassLib.

4- قم بترجمة تطبيق المكتبة ClassLibrary1 وذلك باختيار ClassLibrary1 Build من الاختيار Build من القائمة الرئيسية تحصل على رسالة تفيد ترجمة التطبيق بدون أخطاء (Build: 1 succeeded, 0 failed, 0 skipped).

قبل الانتقال الى الخطوة الثانية وهى استعمال هذه المكتبة تأكد من ترجمة وإنشاء ملف ClassLibrary1.DLL وذلك بفتح المجلد MyDocuments من سطح المكتب DeskTop ثم فتح المجلد Visual Studio Projects تجد مجلد بالاسم ClassLibrary1 افتحه تجد مجلد بالاسم bin افتحه تحصل على الملف ClassLibrary1.dll كما فى الشكل (3-15).

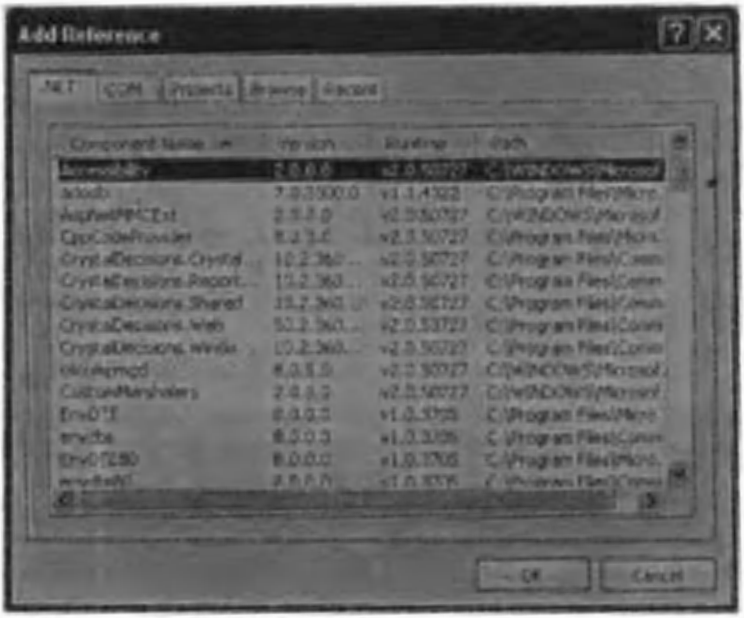
شكل (3-15)

انشاء تطبيق Windows App يستخدم المكتبة ClassLibrary1

لانشاء تطبيق يستعمل المكتبة التى تم إنشائها ClassLibrary1 اتبع الخطوات التالية:

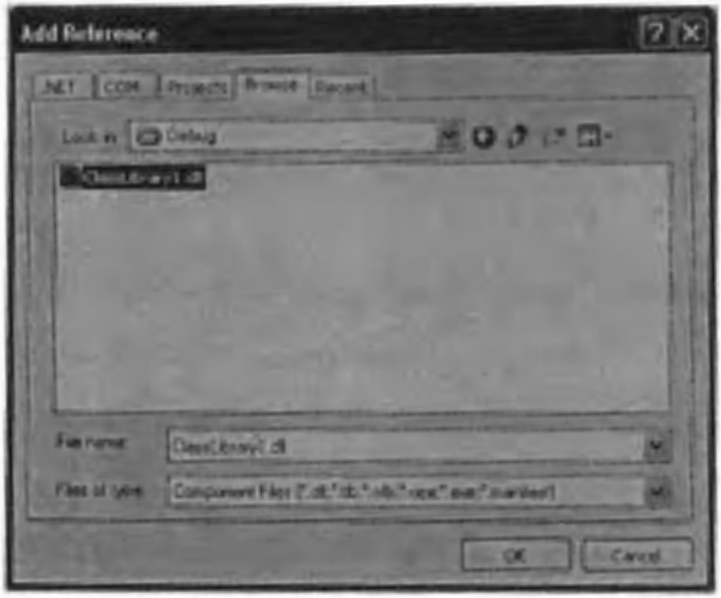
1- قم بانشاء تطبيق جديد من النوع Windows Appliaction كما سبق.

2- اختر Project من القائمة الرئيسية ومنها اختر Add Reference يظهر مربع حوار Add Reference للاشارة الى ملف المكتبة كما فى الشكل (4-15).



شكل (15-4)

3- اضغط الصفحة Projects (Tab) ثم اضغط الزر Browse واختر الملف ClassLibrary1 كما في الشكل (15-5)



شكل (15-5)

4- اضغط الزر Open فيتم اضافة ملف المكتبة الى مجموعة العناصر المختارة Selected Components أسفل مربع الحوار ،اضغط الزر Ok تعود الى بيئة التصميم.

5- وقع زر أمر Button وغير عنوانه الى Hello واضغطه مرتين واكتب به سطور أوامر استدعاء الدالة Hello المعرفة فى الفصيلة class1 المعرفة فى مكتبة الفصائل ClassLibrary1 التى قمنا بإعدادها وقم بعمل Reference إليها وتتضح هذه السطور فى الشكل (6-15).

```
Dim obj1 As New ClassLibrary1.Class1()
obj1.Hello()
```

شكل (6-15)

فى لغة C#

```
106         private void button1_Click(object sender, System.EventArgs e)
107         {
108             ClassLibrary2.Class1 obj1=new ClassLibrary2.Class1 ();
109             textBox1.Text =obj1.Hello ();
110
111         }
```

فى هذه السطور

- فى السطر رقم 59 تم الإعلان عن المتغير Obj1 كهدف (object) من الفصيلة Class1 المعرفة داخل المكتبة ClassLibrary1.
- فى السطر رقم 60 تم استدعاء الدالة ()Hello العضوة فى الفصيلة Class1.

وبالتالى عند تنفيذ البرنامج والضغط على الزر Hello تنفذ الدالة Hello وتحصل على الرسالة كما فى الشكل (7-15).

شكل (7-15)

انشاء تطبيق مكتبة Class Lib2 متقدم

فى الفقرات السابقة قمنا بإعداد مكتبة Class Library بسيطة وكذلك تطبيق يستعمل هذه المكتبة وذلك لتوضيح كيفية انشاء مكتبة فصائل Class Library وكيفية استعمالها ولكن بالطبع لن نقوم بإعداد مكتبة لا تحتوى إلا على Class بسيط بهذا الشكل وكذلك لاتحتوى إلا على دالة تقوم بعرض رسالة ولكن الطبيعى إننا نقوم باعداد مكتبة class Library ذات قيمة ،ولتوضيح ذلك نقوم باعداد مكتبة Class Library تحتوى على فصيلتين Classes الأولى MathOperations وتوفر دوال لتحقيق العمليات الحسابية الأساسية وهى الجمع والطرح والضرب والقسمة من خلال مجموعة الدوال الأعضاء Member Methods التى تحقق هذه العمليات والثانية AdvcOperation التى توفر بعض العمليات مثل عملية الأس ()Power .

ولتحقيق ذلك تابع الخطوات التالية:

1 -قم باعداد تطبيق جديد من النوع ClassLibrary كما فى سبق.

2- اترك الاسم الافتراضى ClassLibrary2 كما هو ثم اضغط الزر Ok تحصل على ملفات تطبيق المكتبة وبها الملف class1.vb الذى يحتوى فصيلة class بالاسم class1 مفتوحة للكتابة.

3- قم بتغيير اسم الفصيلة class1 الى MathOperations ثم اكتب تعريف الدوال Methods التى تحقق العمليات البسيطة كما فى السطور الموضحة فى الشكل (8-15).

```
Public Class MathOperation
Dim No1 As Double
Dim No2 As Double
Dim Res As Double

Public Function MySum(ByVal v_No1 As Double, ByVal v_No2 As Double)
No1 = v_No1
No2 = v_No2
Res = No1 + No2
Return Res
End Function
Public Function MySub(ByVal v_No1 As Double, ByVal v_No2 As Double)
No1 = v_No1
No2 = v_No2
Res = No1 - No2
Return Res
End Function
Public Function MyMul(ByVal v_No1 As Double, ByVal v_No2 As Double)
No1 = v_No1
No2 = v_No2
Res = No1 * No2
Return Res
End Function

Public Function MyDiv(ByVal v_No1 As Double, ByVal v_No2 As Double)
No1 = v_No1
No2 = v_No2
Res = No1 / No2
Return Res
End Function

End Class
```

الشكل(8-15)

فى لغة #C

```
    public class MathOperations
    {
        double No1,No2,No3;

        public MathOperations()
        {
            // TODO: Add constructor logic here
        }
    public  double MySum(double v_No1,double v_No2)
        {
        No1=v_No1;
        No2=v_No2;
        No3=No1+No2;
return No3;
        }
public double MySub(double v_No1,double v_No2)
        {
            No1=v_No1;
            No2=v_No2;
            No3=No1-No2;
            return No3;
        }

public double MyMul(double v_No1,double v_No2)
        {
            No1=v_No1;
            No2=v_No2;
            No3=No1*No2;
            return No3;
        }
public double MyDiv(double v_No1,double v_No2)
        {
            No1=v_No1;
            No2=v_No2;
            No3=No1/No2;
            return No3;
        }
        }
```

فى هذه السطور:

- فى السطر رقم 1 تم تعريف فصيلة class بالاسم MathOperations.
- فى السطور رقم 2و3و4 تم الاعلان عن ثلاثة متغيرات من النوع Doublelhx .No1,No2,No3
- من السطر رقم 5 الى السطر رقم 28 تم انشاء 4 دوال Methods أعضاء فى الفصيلة لتحقيق العمليات الحسابية (/,*,-,+).
- فى السطر رقم 5 تم انشاء الدالة ()MySum التى تقوم باستقبال معاملين (Paramters) فى المتغيرين v_No1,v_No2.

- فى السطرين رقم 6و7 تقوم الدالة بوضع قيم المعاملات فى متغيرى الفصيلة No1,No2.
- فى السطر رقم 8 ثم تقوم بجمع القيمتين ووضع النتيجة فى المتغير Res.
- فى السطر رقم 9 يتم إعادة النتيجة باستعمال الأمر Retuen مع المتغير Res.
- وبالمثل يتم إعداد الدوال MySub(),MyDiv(),MyMul().

4- قم بتعريف فصيلة Class جديدة بعد إنتهاء الفصيلة الأولى (End Class) وذلك بالاسم AdvcOperation وقم بكتابة دوال هذه الفصيلة كما هو موضح فى السطور التالية كما بالشكل (9-15).

```
Public Class AdvcOperation
Dim b As Integer
Dim p As Integer
Dim res As Integer

Public Function MyPower(ByVal v_b As Integer, ByVal v_p As Integer)
Dim i As Integer
res = 1
b = v_b
p = v_p
For i = 1 To p
res = res * b
Next
Return res
End Function

Public Function MyPower(ByVal v_factNo As Integer)
Dim i As Integer
res = 1
For i = 1 To v_factNo
res = res * i
Next
Return res
End Function
End Class
```

شكل (9-15)

فى لغة C#

```
47      public class AdvcOperation
48      {
49  public int b,p,res;
50          public int MyPower(int v_b,int v_p)
51          {
52
53              b=v_b;
54              p=v_p;
55              int i;
56              for (i=1 ;i<=p ;i++)
57          res=res*b;
58              return res;
59          }
60          public int MyFact(int v_factNo)
61          {
62              int i;
63              res=1;
64              for (i=1 ;i<=v_factNo ;i++)
65                  res=res*i;
66              return res;
67          }
68      }
```

فى هذه السطور:

- فى السطر رقم 31 تم تعريف فصيلة class بالاسم AdvcOperation.
- فى السطور رقم 32و33و34 تم الاعلان عن ثلاثة متغيرات من النوع Double.
- فى السطر رقم 35 إلى السطر رقم 42 تم انشاء دالة Methods بالاسم MyPower() كدالة عضوة فى الفصيلة لحساب قيمة رقم مرفوع لأس (Power).
- فى السطر رقم 37 تم استعمال For Loop لتكرار ضرب الاساس b فى القيمة Res التى تأخذ أول مرة القيمة 1 ثم تأخذ نتيجة الضرب فاذا كان المطلوب ايجاد قيمة الرقم 5 مرفوع للاس 4 يتم ضرب الرقم 5 أول مرة فى 1 ثم فى 5 ثم فى 25 وهكذا

حتى تنتهي قيمة الاس ثم تقوم بوضع النتيجة في المتغير Res واعادتها عند استدعاء الدالة.

- بالمثل يتم انشاء دالة أخرى بالاسم ()MulFact يمكنك مراجعة سطورها من السطر رقم 43 الى السطر رقم 50.

5- قم بترجمة تطبيق المكتبة ClassLibrary2 وذلك باختيار Build ClassLibrary2 من الاختيار Build من القائمة الرئيسية لتحصل على رسالة تفيد ترجمة التطبيق بدون أخطاء (Build: 1 succeeded, 0 failed, 0 skipped).

قبل الانتقال الى الخطوة الثانية وهي استعمال هذه المكتبة تأكد من ترجمة وإنشاء ملف ClassLibrary2.DLL وذلك بفتح المجلد MyDocuments من سطح المكتب DeskTop ثم فتح المجلد Visual Studio Projects تجد مجلد بالاسم ClassLibrary2 افتحه تجد مجلد بالاسم bin افتحه تحصل على الملف ClassLibrary1.dll كما سبق.

إنشاء تطبيق Windows App يستخدم المكتبة ClassLibrary2

لانشاء تطبيق يستعمل المكتبة التي تم انشائها ClassLibrary2 اتبع الخطوات التالية:

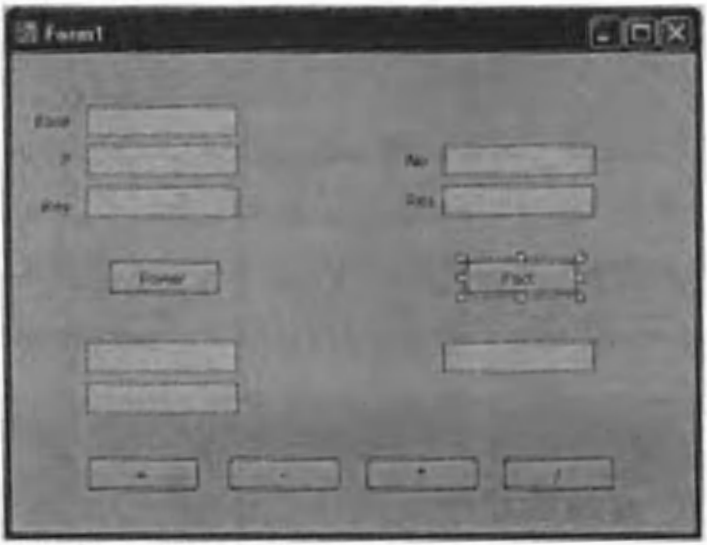
1- قم بإنشاء تطبيق جديد من النوع Windows Appliaction كما سبق.

2- اختر Project من القائمة الرئيسية ومنها اختر Add Reference يظهر مربع حوار Add Reference للاشارة الى ملف المكتبة كما سبق.

3- اضغط الصفحة (Tab) Projects ثم اضغط الزر Browse واختر الملف ClassLibrary2 كما سبق.

4- اضغط الزر Open فيتم اضافة ملف المكتبة الى مجموعة العناصر المختارة Selected Components اسفل مربع الحوار ، اضغط الزر Ok تعود الى بيئة التصميم.

5- قم بتصميم النموذج Form1 ليصبح كما في الشكل (10-15).



شكل (15-10)

6- اضغط مرتين الزر Power واكتب به سطور أوامر استدعاء الدالة MyPower() المعرفة فى الفصيلة AdvcOperation المعرفة فى مكتبة الفصائل ClassLibrary2 التى قمنا بإعدادها وقم بعمل Reference اليها وتتضح هذه السطور فى الشكل (15-11).

```
Dim AdvcOpObj As New ClassLibrary2.AdvcOperation()
Dim r As Integer
r = AdvcOpObj.MyPower(Val(TextBox3.Text), Val(TextBox4.Text))
TextBox5.Text = r
```

شكل (15-11)

فى لغة C#

```
164        private void button1_Click(object sender, System.EventArgs e)
165        {
166            int r;
167            ClassLibrary2.AdvcOperation AdvcOpObj =new ClassLibrary2.AdvcOperation ();
168            r=AdvcOpObj.MyPower  (Int32.Parse (textBox3.Int32.Parse (textBox4));
169            textBox5.Text =r+"";
170        }
```

فى هذه السطور :

- فى السطر رقم 236 تم تعريف متغير هدف (Object) بالاسم AdvcOpObj من الفصيلة AdvcOperation المعرفة بالمكتبة ClassLibrary2.

- فى السطر رقم 238 تم استدعاء الدالة (MyPower() مع متغير الهدف مع ارسال المعاملات ووضع النتيجة فى المتغير r.
- فى السطر رقم 239 تم وضع قيمة المتغير r فى مربع النص TextBox3.

7- اضغط مرتين الزر Fact واكتب به سطور أوامر استدعاء الدالة (MyFact() المعرفة فى الفصيلة AdvcOperation المعرفة فى مكتبة الفصائل ClassLibrary2 التى قمنا بإعدادها وقم بعمل Reference إليها وتتضح هذه السطور فى الشكل (12-15).

```
242    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
243        Dim AdvcOpObj As New ClassLibrary2.AdvcOperation()
244        Dim r As Integer
245        r = MathOpObj.MyFact(Val(TextBox4.Text))
246        TextBox3.Text = r
247    End Sub
```

شكل (12-15)

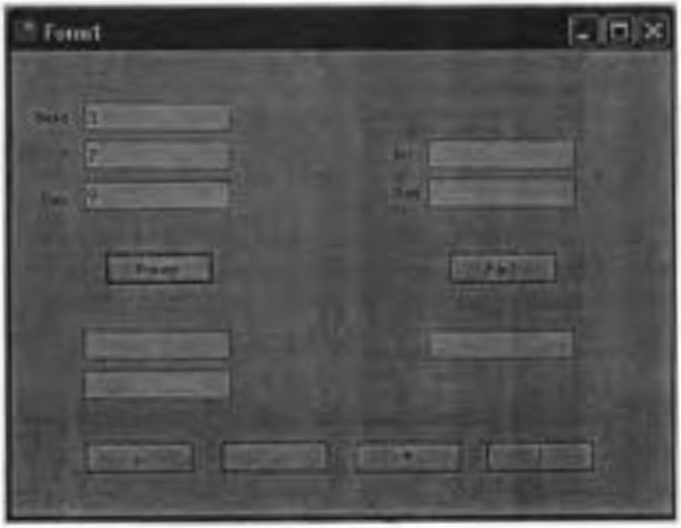
فى لغة C#

```
156        private void button1_Click(object sender, System.EventArgs e)
157        {
158            int r;
159    ClassLibrary2.AdvcOperation AdvcOpObj =new ClassLibrary2.AdvcOperation ();
160    r=AdvcOpObj.MyFact (Int32.Parse (textBox4));
161    textBox3.Text =r+"";
162        }
```

فى هذه السطور:

- فى السطر رقم 242 تم تعريف متغير هدف (Object) بالاسم AdvcOpObj من الفصيلة AdvcOperation المعرفة بالمكتبة ClassLibrary2.
- فى السطر رقم 245 تم استدعاء الدالة (MyFact() مع متغير الهدف مع ارسال المعاملات ووضع النتيجة فى المتغير r.
- فى السطر رقم 246 تم وضع قيمة المتغير r فى مربع النص TextBox3.

8- نفذ البرنامج عند هذه المرحلة وجرب كتابة رقم كأساس ورقم كأس ثم اضغط الزر Power ولاحظ النتيجة وجرب كتابة رقم فى خانة No ثم اضغط الزر Fact ولاحظ النتيجة كما فى الشكل (13-15).



شكل (13-15)

9- قم بكتابة سطور أوامر الأزرار /,*,-,+ كما هو موضح فى السطور التالية كما بالشكل (14-15).

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Dim mathop As New ClassLibrary2.MathOperation()
        Dim r As Double
        r = mathop.MySum(Val(TextBox6.Text), Val(TextBox7.Text))
        TextBox8.Text = r
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        Dim mathop As New ClassLibrary2.MathOperation()
        Dim r As Double
        r = mathop.MySub(Val(TextBox6.Text), Val(TextBox7.Text))
        TextBox8.Text = r
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
        Dim mathop As New ClassLibrary2.MathOperation()
        Dim r As Double
        r = mathop.MyMul(Val(TextBox6.Text), Val(TextBox7.Text))
        TextBox8.Text = r
    End Sub
```

```
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
    Dim mathop As New ClassLibrary2.MathOperation()
    Dim r As Double
    r = mathop.MyDiv(Val(TextBox6.Text), Val(TextBox7.Text))
    TextBox8.Text = r
End Sub
```

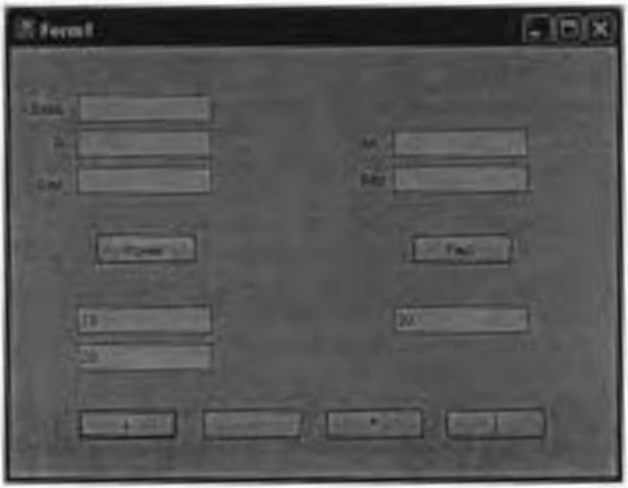
شكل (14-15)

فى لغة C#

```
private void button3_Click(object sender, System.EventArgs e)
{
ClassLibrary3.MathOperations mo=new ClassLibrary3.MathOperations ();
    double r;
    r=mo.MySum (double.Parse (textBox6),double.Parse (textBox7));
    textBox8.text=r;
}
private void button4_Click(object sender, System.EventArgs e)
{
ClassLibrary3.MathOperations mo=new ClassLibrary3.MathOperations ();
    double r;
    r=mo.MySub (double.Parse (textBox6),double.Parse (textBox7));
    textBox8.text=r;
}
private void button5_Click(object sender, System.EventArgs e)
{
ClassLibrary3.MathOperations mo=new ClassLibrary3.MathOperations ();
    double r;
    r=mo.MyMul (double.Parse (textBox6),double.Parse (textBox7));
    textBox8.text=r;
}
private void button6_Click(object sender, System.EventArgs e)
{
ClassLibrary6.MathOperations mo=new ClassLibrary3.MathOperations ();
    double r;
    r=mo.MyDiv (double.Parse (textBox6),double.Parse (textBox7));
    textBox8.text=r;
}
```

حاول فهم شرح هذه السطور من شرح السطور السابقة.

10- نفذ البرنامج وجرب الأوامر الجديدة تحصل على النتيجة كما فى الشكل (15-15).



شكل (15-15)

ما هى Custom controls

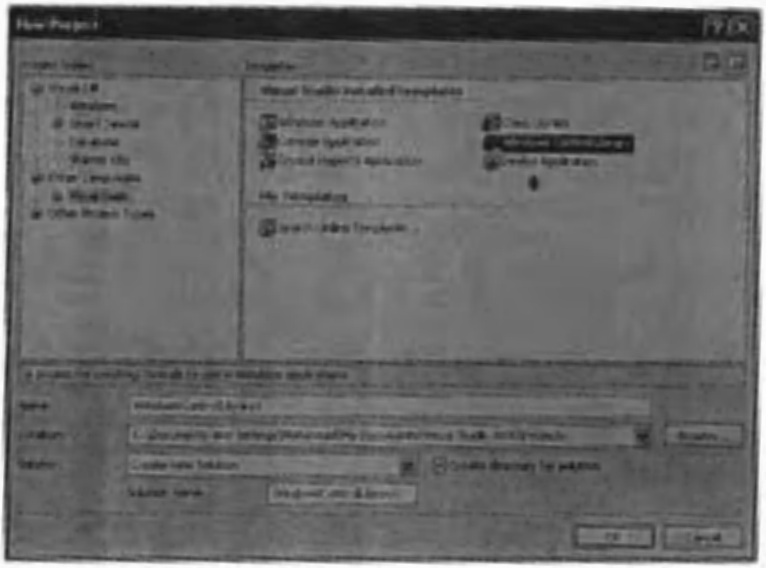
هى فكرة إنشاء أدوات Tools (ويطلق عليها Controls) مثل الأدوات الموجودة بمربع الأدوات ToolBox ولكن تكون خاصة بنا وتحقق عمليات ووظائف غير الأدوات المعرفة باللغة ، والأدوات المعرفة باللغة تسمى Built In Controls لإنها مبنية باللغة وليس لها دخل بها إلا الاستعمال.

بينما الأدوات التى نحن بصددها يقوم مستخدم اللغة (المبرمج) بإنشائها لذلك تسمى User defind Controls أو Custom controls أو WindowsControlLibrary ويمكنك أن تجد الآلف الكثيرة من هذه الأدوات التى يقوم المبرمجين بإعدادها ونشرها على شبكة الإنترنت إذا بحثت عن الامتداد OCX وهى مفيدة جداً حيث إنها عبارة عن برامج كاملة فى صورة أدوات بسيطة تضاف إلى شريط الأدوات ، ولتوضيح أهميتها تابع معنا الفقرات التالية:

انشاء تطبيق أداة Custom Control بسيطة MyTimer

الخطوة الأولى هى شرح بالخطوات كيفية انشاء أداة Control بسيطة تقوم بعرض الوقت والتاريخ الحالى تلقائيا ولتحقيق ذلك تابع الخطوات التالية:

1- قم بانشاء تطبيق جديد من النوع WindowsControlLibrary وغير اسمه الى MyTimer كما فى الشكل (16-15).



شكل (15-16)

2-اضغط الزر Ok تدخل الى بيئة التصميم وتحصل على نموذج Form صغير وبسيط يشابه النموذج Form الخاص بتطبيق Windows Appliaction كما في الشكل (15-17).

شكل (15-17)

3- قم بتوقيع أداة Label وأداة Timer واضبط خاصية Enabled لأداة الـ Timer1 على القيمة True والخاصية Interval على القيمة 1000.

4- اضغط أداة الـ Timer بالماوس مرتين واكتب بها أوامر عرض الوقت والتاريخ كما هو موضح في الشكل (18-15).

```
Public Class MyTimer
    Inherits System.Windows.Forms.UserControl

 Windows Form Designer generated code

    Private Sub Timer1_Tick(ByVal sender As System.Object
        Label1.Text = Now
    End Sub
End Class
```

شكل (18-15)

في لغة #C

```
        private void timer1_Tick(object sender, System.EventArgs e)
        {
            label1.Text=(Int32.Parse(label1.Text)+1)+"";

        }
    }
```

مع ملاحظة تغيير اسم الفصيلة إلى MyTimer.

5- قم بترجمة تطبيق الأداة وذلك باختيار Build MyTimer من الاختيار Build من القائمة الرئيسية وتأكد من عدم وجود أخطاء.

6- اذهب إلى مجلد التطبيق MyTimer وتأكد من وجود الملف MyTimer.dll داخل الفهرس bin داخل الفهرس MyTimer داخل الفهرس Visual Studio Projects داخل الفهرس My Documents كما في الشكل (19-15).

شكل (15-19)

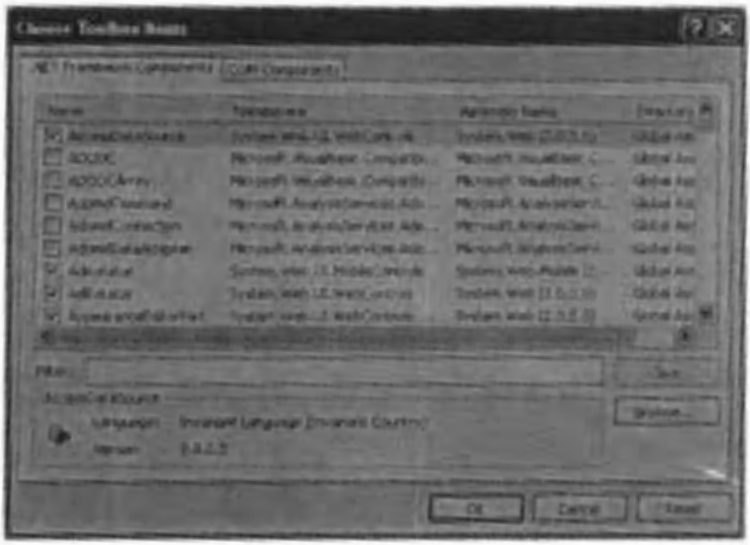
بهذا تم إعداد الأداة ويبقى استعمالها.

انشاء تطبيق يستعمل الاداة الجديدة MyTimer

هذه هى الخطوة الثانية فى التعامل مع الأدوات Windows Control Library وهى إعداد التطبيق الذى يستعمل هذه الأداة ويتم ذلك كما فى الخطوات التالية:

1- قم بإنشاء تطبيق من النوع Windows Application كما سبق.

2- اضغط بالطرف الأيمن للماوس فوق مربع الأدوات ToolBox تحصل على قائمة اختيارات Menu اختر منها Custom ToolBox يظهر مربع حوار Customize ToolBox كما بالشكل (20-15).



شكل (15-20)

3- اضغط الصفحة (tab) .NET Framework Components. ثم اضغط الزر Browse لتحميل ملف الأداة MyTimer.Dll الذى انشأناه فى الفقرة السابقة.

4- يظهر مربع حوار فتح ملف حدد مجلد الملف MyTimer وحدد الملف MyTimer.DLL كما فى الشكل (21-15).

شكل (21-15)

5- اضغط الزر Open يتم اضافة الأداة الى مجموعة الأدوات المختارة والعودة إلى مربع حوار Cusomize ToolBox.

6- اضغط الزر Ok تعود إلى بيئة التصميم وقد تم إضافة الأداة الجديدة MyTimer فى أخر مربع الأدوات ToolBox.

7- قم بسحب وتوقيع الأداة الجديدة MyTimer على النموذج Form لحصل على الأداة التى صممناها وتراها تعمل تلقائيا فى بيئة التصميم وتعرض الوقت والتاريخ كما فى الشكل (22-15).

شكل (22-15)

8- نفذ البرنامج تلاحظ أن الأداة تعمل كما كانت تعمل فى بيئة التصميم بهذا تم إنشاء الأداة مرة واحدة ثم استعمالها أكثر من مرة.

فى لغة Java

المكتبات Packages

هى فكرة ممتازة وقوية فى لغات البرمجة.

وهى فكرة بناء مكتبات من الفصائل classes ، حيث يتم وضع عدد ضخم من الفصائل classes ولذلك فإننا نضع كل مجموعة من الفصائل classes المتشابهة مع بعضها البعض في حزمة واحدة.

- إذن المكتبات Packages هى عبارة عن مجلدات (Directories) تنظم الفصائل classes بحيث أن كل مجموعة فصائل classes لها وظائف مشتركة فى لغة الجافا Java يتم وضعها فى حزمة Package معينة.
- وقد تعاملنا فيما مضى مع المكتبات Packages الخاصة بلغة الجافا Java مثل:

```
import javax.swing.*;
```

وفيها المكتبات package المسماه (javax.swing) هي حزمة تحتوي على الفصائل classes الخاصة بأدوات الـ swing وبالمثل باقي المكتبات packages التي تعاملنا معها من قبل.

وهذا الأسلوب لا تخلو منه لغة برمجة وبدونه تصبح جميع الفصائل مع بعضها بدون تنظيم ولا تقسيم ، ولكن عمليا يتم وضع جميع الفصائل التى تؤدى وظائف متشابهة معا ، مثل فصائل الرسم Graphics والأدوات Components كما أشرنا .

ولكن هذه المكتبات تأتى مبنية فى لغة الجافا فكيف نستطيع بناء حزم خاصة بنا ولماذا؟ نريد أن ننشئ المكتبات Packages الخاصة بنا للأسباب الآتية:

- عندما تنشئ نظاماً معقداً فيفضل وضع مجموعة الوظائف الخاصة بجزء معين من النظام فى حزمة package خاصه بها فمثلاً قد يكون النظام يتعامل مع شبكات أو رسومات أو قاعدة بيانات أو واجهة مستخدم ولذلك يفضل إنشاء حزمة package لكل وظيفة على حدة بأن تضع حزمة package للشبكات و حزمة package للرسومات وهكذا.
- من الممكن إعادة استخدام المكتبات Packages التى تنشأها فى برامج أخرى فإذا كانت هناك حزمة Package أنشأتها توفر وظيفة عامة فستطيع إعادة استخدامها مباشرة.

مثال آخر

```
import java.util.*;
```

فى هذا المثال يتم استيراد (الاشارة الى) جميع فصائل classes (*) المكتبة util الموجودة فى الفهرس java وبالتالى يمكن استعمال الفصائل classes الموجودة بها كما فى السطر.

```
Date today = new Date();
```

فى هذا السطر يتم تعريف متغير هدف بالاسم today من نوع الفصيلة Date وبالتالي اذا لم يتم استيراد المكتبة التى تحتوى على الفصيلة Date لا يتعرف المترجم عليها ويعطى خطأ.

ويمكن استيراد (الإشارة إلى) فصيلة محددة class (اسم الفصيلة) موجودة بالمكتبة كما فى السطر.

```
import java.util.Date;
```

فى هذا السطر تم تحديد استيراد الفصيلة Date من المكتبة java.util.

مع ملاحظة أن الإشارة إلى جميع فصائل المكتبة لا يزيد من حجم البرنامج.

ملحوظة:

لايمكن استيراد أو الإشارة إلى جميع مكتبات فهرس محدد فمثلا من غير المقبول كتابة السطر .

```
 import java.*
or
 import java.*.*
```

للإشارة إلى جميع مكتبات الفهرس java.

ملحوظة:

يجب مراعاة عدم الاشارة الى مكتبات Packages تحتوى على فصائل classes بنفس الاسم والا سيحدث تداخل Conflict بينهما كما فى السطور التالية:

```
import java.util.*;
import java.sql.*;
```

المشكلة فى هذه السطور هو احتواء المكتبتين على فصيلة بالاسم Date مما يحدث تداخل.

ولو افترضنا انك تريد استعمال كلا الفصيلتين Date من كلا المكتبتين يتم الاشارة اليهما صراحة كما فى السطور التالية:

```
java.util.Date deadline = new java.util.Date();
java.sql.Date today = new java.sql.Date(...);
```

إنشاء المكتبة Creating Packages

لنفترض الآن أننا سننشىء فصيلة class بها دالة اسمها printData() مهمتها طباعة رسالة على الشاشة. ونريد أن نستخدم هذه الدالة فى فصيلة class أخرى.

الخطوات

1-قم بنشاء تطبيق جديد واكتب فيه السطور التالية :

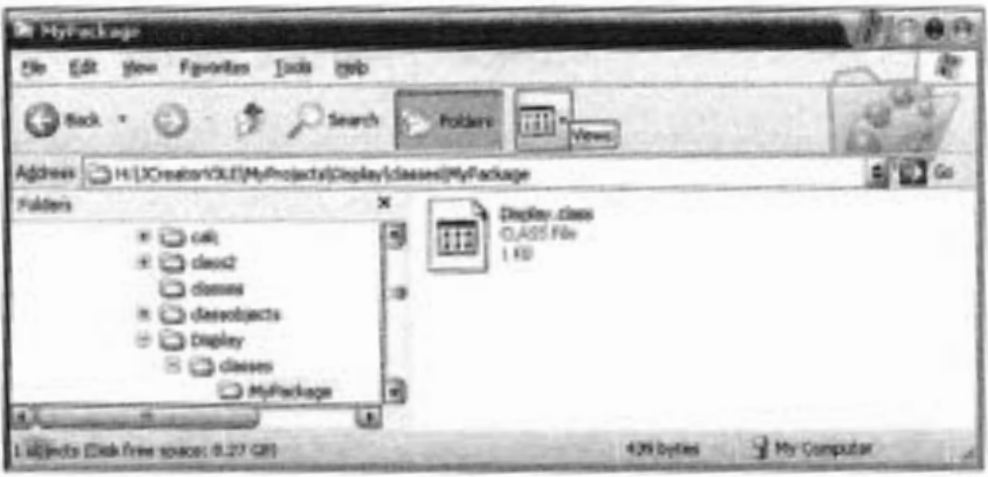
```
1  package MyPackage;
2  public class Display {
3
4      public void printData ()
5      {
6  System.out.println("This is class Display");
7
8      }
9  }
10
```

فى هذه السطور:

فى السطر رقم 1 يتم انشاء مجلداً Directory اسمه MyPackage. هذا المجلد سينشأ فى نفس مسار الملف Display.Java أى D:\Pack .

1. لكى تستطيع استخدام الفصيلة class المسماه Display واستدعاءها من حزمة package أخرى لابد أن يكون تعريفها public لكى تستطيع استخدامها.
2. فى السطر 5 يتم تعريف الدالة PrintData() التى تطبع رسالة على الشاشة.

وبعد كتابة البرنامج بدون أخطاء وترجمته من داخل البرنامج JCreator يتم انشاء المكتبة MyPackage كما فى الشكل (23-15).



شكل (32-15)

فى هذا الشكل تلاحظ انشاء فهرس بالاسم MyPacjage وبداخله الفصيلة Display.class كما أشرنا مع ملاحظة أهمية تحديد المسارات.

ولعمل ترجمة Compile لهذا البرنامج من خلال شاشة الـ Dos افتح شاشة الـ Dos حول المسار إلى المسار الفرعى D:\Pack وهو مسار الملف Display.Java.

قم بترجمة البرنامج لينشئ المكتبة package التى سميناها MyPackage ويضع الفصيلة class الناتجه فى هذا المجلد D:\Pack\MyPackage ولتنفيذ ذلك نكتب الأمر التالى.

```
javac -d . Display.java
```

الاختيار d- ينشىء المجلدات (directories) التى نريد أن ننشئها لاحظ وجود مسافة

بعد d> ثم ضع نقطة ثم اترك مسافة ثم اكتب Display.java. النقطة هنا ضرورية لأنها تشير إلى المسار الحالى وهو D:\Pack لأنك تريد أن تنشىء MyPackage فى هذا المسار.

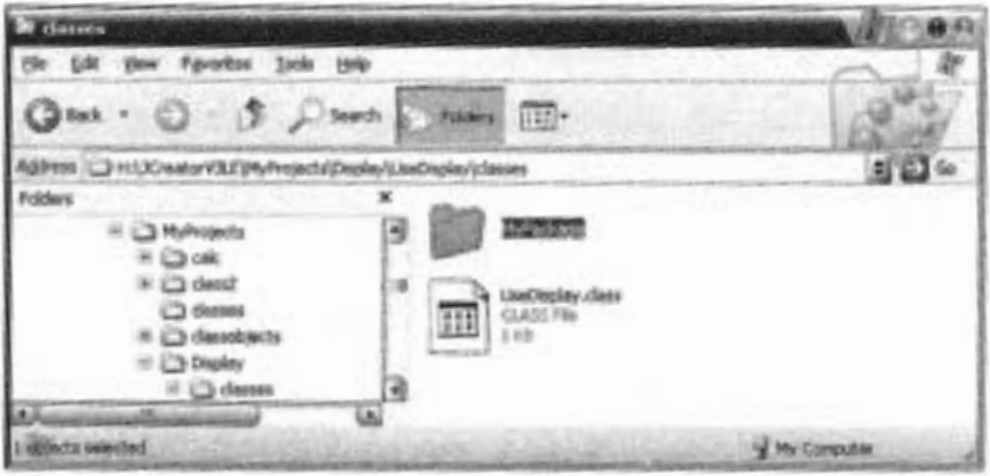
3. فى المسار D:\Pack ستلاحظ أنه تم إنشاء المجلد MyPackage قم بفتح المجلد ستلاحظ وجود Display.class فى هذا المجلد.
4. قم بالرجوع إلى المسار D:\Pack
5. فى المسار D:\Pack قم بإنشاء ملفاً جديداً باسم UseDisplay.java والذي يستخدم المكتبة package التي أنشأناها في البرنامج السابق. لاحظ أن هذا الملف لابد أن يكون فى نفس مسار المجلد MyPackage.

استعمال المكتبات

6. قم بانشاء تطبيق جديد لاستعمال المكتبة MyPackage التى تم إنشائها فى الخطوة السابقة وذلك كما فى السطور التالية:

```
1  import MyPackage.Display;
2  public class UseDisplay {
3
4      public static void main(String[] args) {
5          Display d = new Display ();
6              d.printData ();
7
8      }
9  }
```

مع ملاحظة مسار المكتبة MyPackeg ، ولتنفيذ هذا البرنامج تم وضع المكتبة فى نفس مسار البرنامج كما فى الشكل (24-15).



شكل (24-15)

في سطور البرنامج:

- في الملف UseDisplay.java بدأ البرنامج بتضمين الفصيلة class المسماة Display الموجوده في المجلد MyPackage.
- في السطر 5 يتم إنشاء هدف object من نوع الفصيلة class المسماة Display. لاحظ أن الفصيلة Display تعريفها موجود في المكتبة MyPackage.
- في السطر 6 يتم استدعاء الدالة ()printData التي هي أحد دوال الفصيلة class المسماة Display.
- ترجم Compile هذا الملف بالطريقة العادية.

```
D:\Pack>javac UseDisplay.java
```

- قم بتنفيذ البرنامج عن طريق D:\Pack>java UseDisplay.

لاحظ طباعة الرسالة عن طريق الدالة ()printData.

أو قم بتنفيذ البرنامج من خلال أحد البرامج المخصصة لذلك كما أشرنا وكما تم من قبل .

النوع Static Imports

من المزايا التي اضيفت ابتداء من الإصدار JDK 5.0, إمكانية اضافة الكلمة static بعد الأمر import مما يوفر في استدعاء الدوال وذلك كما في السطور التالية:

```
1. import static java.lang.System.*;
2. class x
3. {
4. public staic void main String args[p])
5. {
6. out.println("Goodbye, World!"); // i.e., System.out
7. exit(0); // i.e., System.exit
8. }
```

فى هذه السطور:

فى السطر رقم 1 تم إضافة الكلمة static بعد كلمة import مما سهل استعمال دوال الفصيلة بدون الاشارة الى المكتبة كما فى السطر رقم6.

وتتضح فائدة ذلك عند التعامل مع بعض الدوال التى لا تحتاج أسماء الفصائل والمكتبات كما فى السطور التالية:

```
sqrt(pow(x, 2) + pow(y, 2))
```

وذلك بعد بعد استعمال الـ static import للفصيلة Math

وبالتالى يكون ذلك اوضح وأبسط من السطر التالى:

```
Math.sqrt(Math.pow(x, 2) + Math.pow(y, 2))
```

وكذلك بالنسبة للمتغيرات المعرفة (الثوابت) كما فى السطر.

```
if (d.get(DAY_OF_WEEK) == MONDAY)
```

بدلا من

```
If (d.get(Calendar.DAY_OF_WEEK) == Calendar.MONDAY
```

فى Oracle

ما هى نظرية REUSE إعادة الاستخدام

هي أحد أهم خصائص البرمجة والتطوير SOFTWARE ENGINEERING وهى تعنى انشاء العناصر objects مرة واحدة واستخدامها أكثر من مرة مما يوفر

1- وقت التطوير.

2- توحيد قواعد التطوير.

3- توحيد شكل التطبيق.

4- سرعة التطوير.

ويتم تحقيق هذه النظرية فى جميع عناصر البرمجة مثل:

1- العناصر OBJECTS

وذلك بإنشاء:

1-1- مكتبة عناصر (أدوات) OBJECT LIBRARIES.

2-1-مجموعة عناصر OBJECT GROUP.

2-الخصائص PROPERTIES

وذلك بإنشاء :

1-2-مجموعة الخصائص المرئية VISUAL ATTRIBUATES.

2-2-مجموعة الخصائص PROPERTY CLASS.

3- أكواد PL/SQL

وذلك بإنشاء :

1-3- البرامج الفرعية والدوال PROCEDURES & FUNCTIONS.

2-3- مكتبات حزم الدوال والبرامج الفرعية PACKAGES.

3-3- مكتبات الدوال والبرامج الفرعية PL/SQL LIBRARIES.

وفيما يلى خطوات تحقيق هذه الطرق.

1- بالنسبة لأوامر البرمجة PL/SQL

ما هي حزم الدوال والبرامج الفرعية **PACKAGE**

هي أحد الطرق المتاحة لتجميع أوامر PL/SQL في لغة ORACLE تحت اسم يعبر عن الغرض من هذه الأوامر وقد تناولناها في فصول الـ PL/SQL.

دائما يسعى التطوير في طرق البرمجة إلى النظام واختصار أوامر البرامج لتسهيل إعادة استخدامها.

وكانت أول هذه الطرق البرامج الفرعية PROCEDURES والدوال FUNCTIONS التي تناولناها في أول هذا الكتاب الأول فبدلا من كتابة أوامر PL/SQL كل مرة في أكثر من موضع وعند الاحتياج إليها يعاد كتابتها ظهرت فكرة الدوال والبرامج الفرعية التي تقوم بتجميع مجموعة الأوامر وإنشاء دالة FUNCTION أو PROCEDURE تؤدي غرض معين وعند الاحتياج لهذا الغرض يتم استدعاء الدالة FUNCTION وليس هناك الحاجة لإعادة كتابة سطورها مرة أخرى وهذه هي أول خطوة حيث يتم الإنشاء مرة واحدة والاستدعاء أكثر من مرة.

وجاءت فكرة ال PACKAGE لتزيد الأمر تنظيما فبدلا من إنشاء كثير من الدوال FUNCTION والبرامج الفرعية PROCEDURES المتناثرة وغير المنتظمة يتم إنشاء PACKAGE (حزمة) لغرض معين وليكن الرسائل MESSAGES مثلا ثم تجميع (إنشاء) مجموعة من دوال (FUNCTIONS) و PROCEDURES الخاصة بإرسال الرسائل MESSAGES داخل هذه الحزمة PACKAGE.

وبهذا الأسلوب يصبح لديك مكتبة من الحزم PACKAGES وليس الدوال فمثلا :

- حزمة الرسائل MESSAGE PACKAGE .
- حزمة مراجعة البيانات VALIDATION PACKAGE .
- حزمة التعامل مع قوائم الاختيارات MENU PACKAGE .

وبهذا أصبح الأمر أكثر نظاما .

مزايا استعمال PACKAGES :

مما تناولناه في الفقرة السابقة نجد التعامل مع PACKAGE يتميز بما يلي :

1- تجميع الدوال FUNCTIONS و البرامج الفرعية PROCEDURES التي تخص موضوع واحد في حزمة واحدة PACKAGE.

2- يتم استعمال الدوال والبرامج الفرعية بفتح PACKAGE الخاصة بهم فقط ولا تحتاج لفتح كل دالة أو برنامج فرعي علي حدة .

3- يتم الإشارة إلى اسم الحزمة PACKAGE عند استدعاء دالة FUNCTION أو PROCEDURES داخلها بالشكل التالي :

```
PACKAGE . FUNCTION ( )
```

وهنا يعطى شيء من التنظيم .

4- إنشاء واستعمال PACKAGE يمكنك من إخفاء نصوص أوامر PL/SQL الخاصة بالدوال و البرامج الفرعية حيث يمكنك ترجمتها واستعمالها دون الرجوع لنص البرامج .

هذه بعض المزايا وكما ذكرنا استعمال PACKAGE يعتبر أول الطرق بعد إعادة كتابة الأوامر أكثر من مرة . ولكن مازال هناك الكثير من هذه الطرق مثل :

أجزاء حزمة الدوال PACKAGE PARTS :

قبل الاستمرار وشرح كيفية إنشاء PACKAGE نود أن نوضح أولا أجزائها ،وهي تتكون من الأجزاء التالية :

1. قسم التعريفات PACKAGE SPECIFICATION:

في هذا الجزء يتم الإعلان عن المتغيرات العامة . وكذلك الإعلان عن الدوال FUNCTIONS والبرامج الفرعية PROCEDURES ، والمقصود بالإعلان عنها أي كتابة تعريف أي اسم الدالة ومعاملاتها وأنواعها وكذلك القيمة المرتجعة RETURN VALVE بالنسبة للدوال ولا يتم كتابة سطور أوامر PL/SQL .

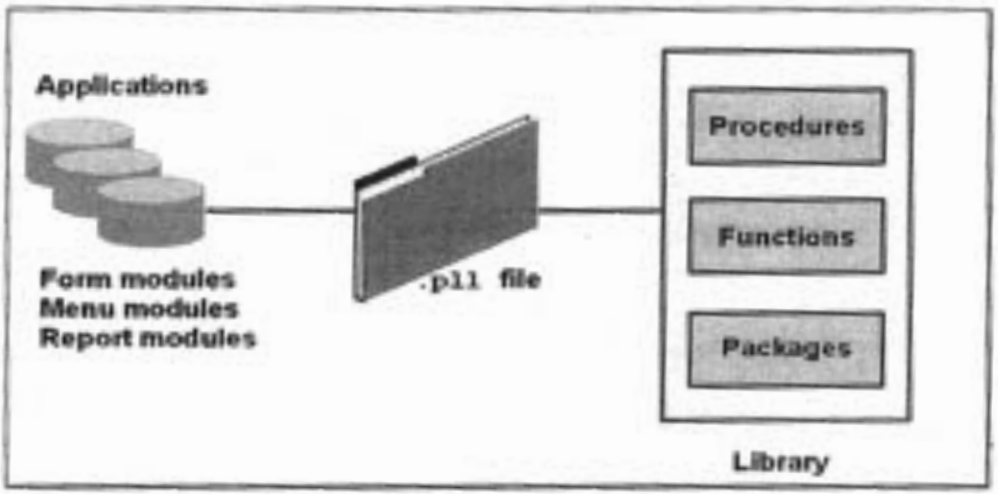
2. قسم الأوامر PACKAGE BODY:

يسمي هذا القسم بجسم الحزمة PACKAGE BODY لأنة يحتوى علي سطور أوامر PL/SQL التي تتكون منها الدوال والبرامج الفرعية فالقسم الأول PACKAGE SPECIFICATION ما هو إلا إعلان وتعريف لأسماء الدوال ولكن هذا القسم هو الذي يحتوي على سطور الأوامر التي تحقق أعمال الدوال وبدونها تصبح الدوال بالتالي حزمة البرامج PACKAGE

ليس لها قيمة فتخيل لو أنك كتبت عنوان دالة تقوم بحساب الضريبة المستحقة ولكنك لم تقم بكتابة الأوامر التي تقوم بحساب هذه الضريبة فهذا ليس له معنى ،وهذا ما يحدث بالضبط بالنسبة لعنوان الحزمة PACKAGE SPECIFICATION وجسم الحزمة PACKAGE BODY وإذا أردت المزيد يمكنك الرجوع الى فصول كتاب الـ PL/SQL.

تركيب المكتبة

ويظهر ذلك كما في الشكل (25-15).



شكل (25-15)

خطوات انشاء PL/SQL LIB

1- قم بعرض النافذة Object Navigator وحدد الاختيار PL/SQL Lib.

2- اضغط الزر + لانشاء مكتبة جديدة ثم قم بتغيير الاسم.

3- قم بكتابة دوال Functions وإجراءات Procedures.

إنشاء عناصر لإعادة الاستخدام DESIGN FOR REUSE

ذكرنا أنه من النقاط المتطورة في عالم التطوير والبرمجة هو إنشاء (إعداد) المكتبات بأشكالها المختلفة من مكتبات دوال PLL ومكتبات عناصر OBJECT LIBRARIES والكثير من الأشكال وهي الخطوة المتقدمة التي يسعى إليها أي شخص مطور أو مكتب أو شركة تطوير وتوجد أكثر.

من صورة لذلك منها ما يلي :-

1- القوالب . TEMPLATES

- قوالب النماذج FORMS TEMPLATES.
- قوالب التقارير REPORTS TEMPLATES.
- قوالب الرسومات GRAPHICS TEMPLATES.

2- الاستفسارات الخارجية EXTERNAL QUERIES.

3- عناصر من النوع VISUAL ATTRIBUTES.

4- عناصر PROPERTY CLASS.

5- عناصر SUB CLASSING &COPYING &READER.

6- مكتبة العناصر OBJECT LIBRARIES &SMART CLASS.

7- مجموعة العناصر OBJECT GROUPS.

8- المعاملات PARAMETERS.

9- قوائم المعاملات PARAMETERS LISTS.

مكتبة العناصر OBJECT LIBIRAY

بالإضافة لفكرة حزم الدوال والبرامج الفرعية PACKAGERS التي شرحناها توجد فكرة أخري تسهل علي المطور DEVELOPER إعادة البناء كل مرة في البداية وهي مكتبة العناصر أو OBJECT LIBRARIES وفكرتها إنشاء مكتبة من العناصر (OBJECTS) الخاصة بالتطوير مع إعطائها الخصائص المطلوبة ثم إعادة استعمالها وقت الحاجة .

ويتم ذلك بتجهيز العناصر وإعطائها الخصائص المطلوبة ثم إضافتها للمكتبة .

ولتوضيح هذا الموضوع نشرح النقاط التالية :

- الخطوة الأولى : إنشاء مكتبة عناصر CREATE OBJECT LIBRARIES.
- الخطوة الثانية : إنشاء عناصر وإعطائها خصائص .
- الخطوة الثالثة : إضافة عناصر لمكتبة العناصر وجعلها من النوع SMART CLASS.
- الخطوة الرابعة : إنشاء نموذج جديد واستعمال مكتبة العناصر .

صديقي :

هذه الفقرة مهمة جدا وهي التعامل مع مكتبة العناصر OBJECT LIBRARIES .

فكما تعلم أن مرحلة اعداد وتنسيق شكل النموذج وتحديد خصائص العناصر ITEMS مهمة وتستغرق الكثير من الوقت بالاضافة لمشكلة تحديد كل مطور الخصائص حسب ما يرى ولكن تخيل ما تستطيع فعلة باستعمال مكتبة العناصر .

تستطيع :

1- إعداد صور مختلفة من جميع العناصر المستعملة في النماذج للاستعمالات المختلفة ، فمثلا إعداد أكثر من عنصر TEXT ITEM:

الأول : هو مربع النص التقليدي المستعمل في الادخال وتعطية اسما وليكن GEN-TEXT مع تحديد الخصائص التي تناسب هذا الاستعمال مثل العرض والطول (HEIGHT,WIDTH) والألوان و الفونتات وغيرة .

الثاني : مربع نص يميز كلمة السر وتحدد له الخصائص التي تميزه وتجعل له اسم أخر وليكن PASSW-TEXT.وثالث ورابع حسب الاستخدامات التي تراها .

2- صور مختلفة من زر الأمر BUTTON أحدهم تقليدي والأخر ICONIC والأخر خاص باستدعاء FORM والأخر خاص باستدعاءLOVوأخر خاص باستدعاء تقرير REPORT .

3- صور مختلفة لجميع العناصر بما فيهم LOV وتحديد خصائصها من طول وعرض وألوان وفونتات.تخيل انشاء مثل هذه المكتبة يعتبر عمل قوي وممتاز.

والسؤال : ماذا تستفيد من انشاء مثل هذه المكتبة أو شئت قل مكتبات ؟

1- تستطيع اعداد النموذج FORM بسرعة تقارب 1/ 10 من السرعة التي تقوم بها باعداد نموذج من البداية وتحديد مواصفاتة كل مرة .

2- تضمن تطابق جميع النماذج في الخصائص أي تصبح الخصائص STANDARD .

3- تضمن عدم قيام كل مطور بتحديد الخصائص كما يشاء .

حقيقة تستفيد الكثير .

عموما : جرب إنشاء مكتبة أو مكتبات وجرب استعمالها وسوف تشعر بالفرق .

كما أنه سوف يفيدك كثيرا بإضافة مكتبة عناصر قيمة إلى أسطوانة الكتاب لتوفر عليك الكثير .

كما يمكن نسخ العناصر من مكتبة إلى مكتبة وكذلك حذف عناصر من مكتبة ويمكن نسخ العنصر إلى بيئة التطوير والتغيير فيه ثم نسخه بعد التعديل للمكتبة .

بالطبع كنا نود شرح خطوات تحقيق ذلك ولكنه يطول لذلك يمكنك الرجوع الى كتاب الطريق الى احتراف اوراكل .

قوالب النماذج Template Form

نتابع موضوع من الموضوعات المتقدمة ولا استغناء عنه في عالم التطوير والبرمجة وهو إعادة استخدام العناصر REUSE الذى بدأناه من قبل ونتابعه بكيفية إنشاء واستعمال قوالب النماذج TEMPLATE FORM ويعتبر هذا الموضوع هو أساس بناء التطبيقات لأي مطور أو شركة اعداد تطبيقات وفي هذا الفصل نتناول النقاط التالية :

- ما هي فكرة قالب نموذج TEMPLATE FORM ؟
- ما هي فوائد استخدام القالب ؟
- ما هي طريقة استخدم القالب ؟
- ما هي مكونات القالب ؟
- كيفية إنشاء نموذج قالب ؟
- كيفية بناء نموذج جديد علي قالب ؟

ما هي فكرة إنشاء قالب نموذج TEMPLATE FORM

هي فكرة إنشاء نموذج قياسى يوفر هذا النموذج كل مواصفات النموذج الذي نحتاجة في جميع نماذج التطبيق من دوال وأوامر ثم استعمال هذا النموذج كقالب TEMPLATE لبناء النماذج الاخري وبالتالي تأخذ النماذج الجديدة خصائص ودوال وإمكانيات النموذج الاصلي TEMPLATE مما يوفر الكثير من وقت التطوير وسوف يتضح ذلك من الفقرات

التالية وسوف يتضح أكثر عند إعداد تطبيق حقيقي باستخدام قالب النموذج TEMPLATE كما في هذا الكتاب.

فوائد استعمال قالب النموذج TEMPLATE FORM

يوفر ذلك الكثير من الفوائد منها:

1. مشاركة النماذج لنفس العناصر

عند قيامك باعداد تطبيق تلاحظ انك تستعمل بعض العناصر في جميع النماذج مثل الرسائل ALERTS وكذلك قائمة القيمة LOV والكثير من العناصر وبالتالي تضطر بدون استعمال القالب TEMPLATE لانشائها اكثر من مرة ولكن باستعمال القالب تجد أنك تنشئها فقط في القالب TEMPLATE وتستعملها في باقي النماذج المبنية علي القالب.

2. مشاركة الاوامر

من الفوائد التي يوفرها استعمال القالب هو إنشاء وحدات البرمجة أي الدوال والبرامج الفرعية ووضعها في مكتبة أوامر PLL LIB وعمل ATTACH لها في النموذج القالب وبالتالي عند بناء نموذج جديد علي القالب يمكنك الاستفادة من هذة الاوامر وذلك مثل دوال التحقق من البيانات VALIDATION ودوال التعامل مع الأخطاء والرسائل وكذلك الأوامر المسؤلة عن الخصائص العامة للنموذج مثل MAXMIZE وغيره

3. الالتزام بقواعد عامة (مقاييس)

يؤدي اعداد قالب TEMPLATE واستعمالة لبناء نموذج آخر إلى توحيد قواعد التطوير مثل شكل النماذج وكذلك اشكال العناصر فبالطبع انشاء مكتبة عناصر OBJECT LIB وتحول عناصرها إلى SMART CLASSES كما سبق وعند إنشاء نموذج جديد نبني علي القالب تاخذ عناصر النموذج الجديد نفس خصائص القالب TEMPLATE عن طريق SMART CLASSES.

4. قصر وقت التطوير

بالطبع بناء نموج قالب TEMPLATE FORM يحتوي علي الخصائص العامة ومكتبة الدوال PLL ومكتبة عناصر OLB بها SMART CLASSES يؤدي إلى زيادة سرعة إنشاء نماذج

جديد مبنية علي هذا القالب عكس قيامك بإنشاء نموذج جديد كل مرة وتحديد خصائص العناصر وكذلك كتابة الأوامر والدوال.

5. تنظيم طريقة التطوير

تؤدي عملية إنشاء القالب إلى تنظيم فريق المطورين فعند البدء في مشروع جديد (NEW PROJECT) يتم تقسيم العمل بحيث يقوم مجموعة من المبرمجين المحترفين باعداد قوالب النظام TEMPLATE ويقوم مجموعة من المبرمجين المبتدءين باستعمال القوالب في اعداد النماذج لتطبيق وبالتالي لا يتم تكرار الوقت والجهد بالاضافة لذلك لاتحتاج إلى عدد كبير من المبرمجين المحترفين.

عيوب استعمال القوالب TEMPLATE

1. عدم تطور مهارات بعض المطورين

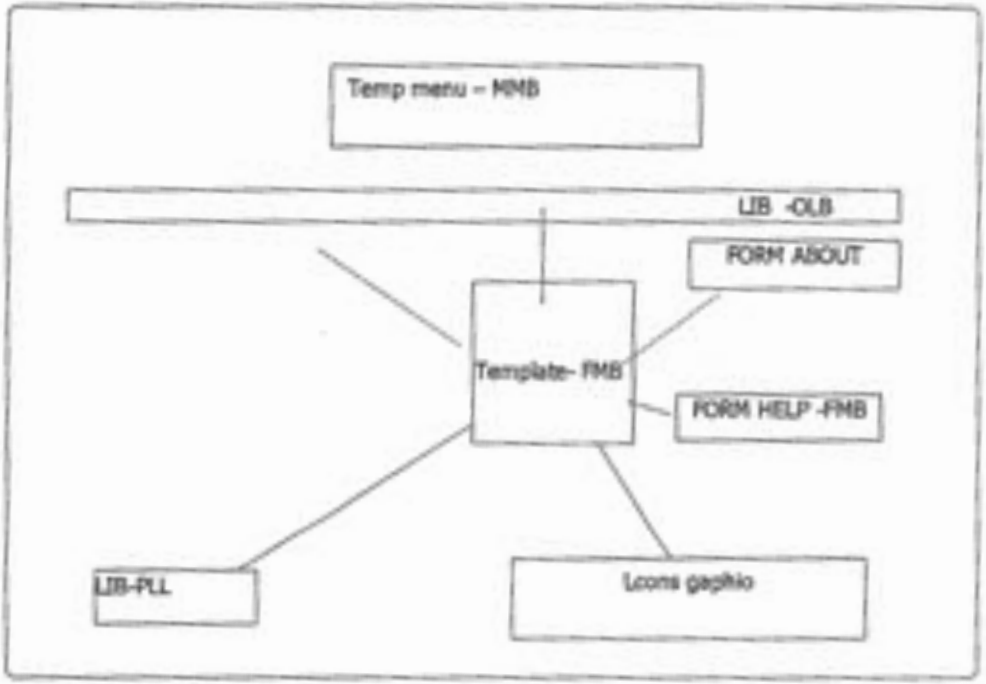
يؤدي استعمال نموذج قالب TEMPLATE FORM من قبل البرمجين المبتدئين لإنشاء نموذج وعدم كتابة الأوامر المطلوبة لهذة القوالب إلى ضعف المستوي لأن القالب يوفر معظم المطلوب وهم يقومون بعمل نماذج تشابه القالب بدون بذل مجهود كبير وبالتالي يضعف المستوي وأكثر من ذلك عند وجود مشكلة تحول إلى المبرمجين المحترفين وبالتالي يظل المبرمجين المبتدئين بعيد عن عملية تطوير المهارات والخل في ذلك محاولة مشاركتهم في عملية التطوير ومراجعة أعمالهم.

مكونات نموذج القالب TEMPLATE FORM

لتوضيح المكونات والعناصر التي يمكن أن يحتوي عليها TEMPLATE انظر إلى الشكل (15-26).

في هذا الفصل عرضنا مقدمة عن قالب النموذج TEMLATE FORM و تناولنا بعض هذة الاجزاء وكيفية إنشائها مثل مكتبة الكائنات EBJECT LIB وتحويل عناصرها إلى SMART CLASSES وكذلك حزم الدوال والبرمجة الفرعية PACKAGES وكذلك مجموعة الخصائص المرئية VISUAL ATTRIBUTE ماذال هناك عناصر وجوانب أخري لقالب

النموذج TEMPLATE FORM نريد تناولها ولكن نتركها لموضوع اخر نتناقش فية عملية التطوير بصفة عامة ولكن نريد أن نشرح هذا من وجهة النظر العملية كيفية استعمال FORM BUILDER لإنشاء نموذج ثم استعمالة كقالب TEMPLATE FORM وإنشاء نموذج جديد مبني عليه.



شكل (26-15)

ويمكنك مشاهدة كيفية تطبيق هذه القواعد فى كتاب الطريق إلى احتراف **Oracle**.