



Les bases des interfaces graphiques

C++

الواجهات الرسومية في
السي بلس بلس

hichemfel@gmail.com

« عندما يولد الانسان يولد معه سجل انجازاته تكون كل صفحاته بيضاء, والانسان له القدرة على ان يتعلم ويعمل وحينها فقط يمكنه ملئ سجل انجازاته بإنجازات صالحة أو انجازات سيئة كما يمكنه أن يتركه خال من الانجازات , فحياة كل انسان تقدر بما يضعه هذا الانسان في سجل انجازاته , والأجيال لا تتذكر الا كل انجاز صالح من أسلافها»

هشام فلواط

لغة السي بلس بلس من أهم وأشهر لغات البرمجة المعروفة لدى المبرمجين حول العالم وذلك للمزايا والقدرات التي تتمتع بها هذه الأخيرة ومن ابرز هذه المزايا الواجهات الرسومية . لكن المشكلة هي قلة المراجع التي تخصص دروس خاصة للواجهات خاصة المراجع العربية . ومن خلال هذا الدرس سأحاول دعم المكتبة العربية وتقديم خدمة لمبرمجي لغة السي بلس بلس نعم هي ليست خدمة كبيرة أو مميزة ولكن أعتقد أنها قد تنفع شخصا ما في عالم البرمجة .

ما نحتاجه هو لغة السي بلس بلس (فيجيوال سي بلس بلس) نسخة 2008 أو

2010 أو 2012

زائد الإرادة وحب البرمجة .

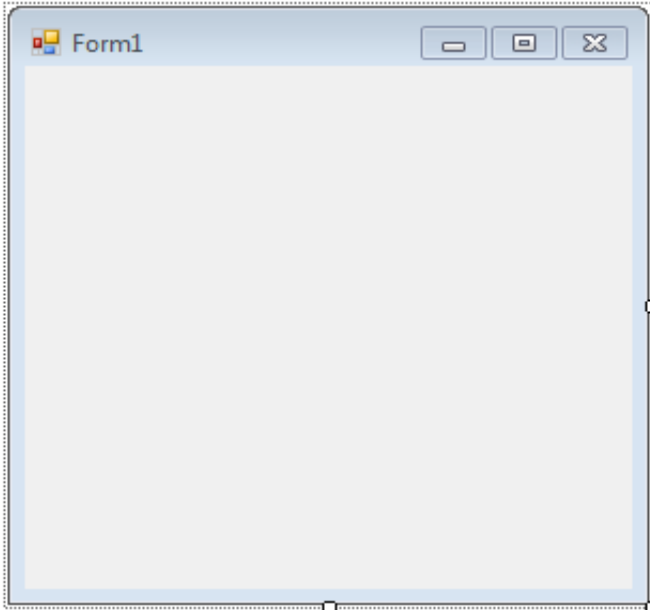


البرنامج الأول

افتح برنامج فيجوال سي بلس بلس أي نسخة تريد ماعدا نسخة 2012 فإنها تختلف عن نسخة 2008 و 2010 سنشرحها لاحقا المهم افتح برنامج جديد كالمعتاد واختر

Windows Forms Application

سيظهر لنا علي الشاشة الفورم الذي سنقوم بوضع الأدوات عليه مثل الأزرار والصور وغيرها

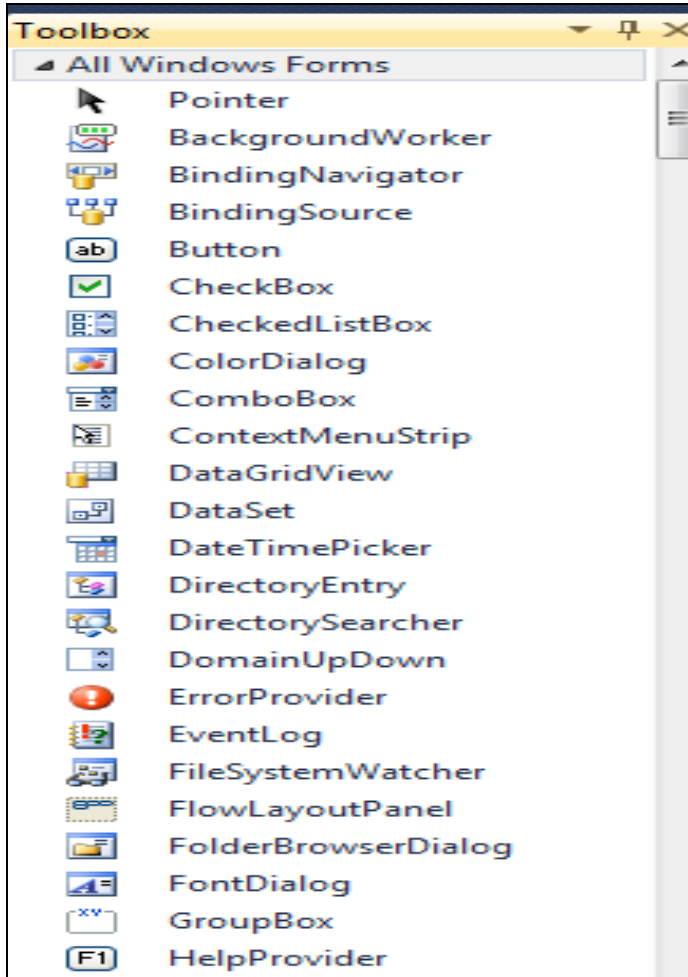


هناك شيان مهمان في برامج الواجهات الرسومية هما شريط الأدوات وشريط الخصائص .
شريط الأدوات يحتوي علي الأدوات التي نستعملها في رسم الفورم مثل أداة الزر.
شريط الخصائص فيه يمكن تحديد خصائص الأدوات التي استعملناها مثل الاسم واللون والطول والعرض .
لذلك اذا كان غير ظاهران في برنامجك أظهرهما من شريط المهام التعليمية

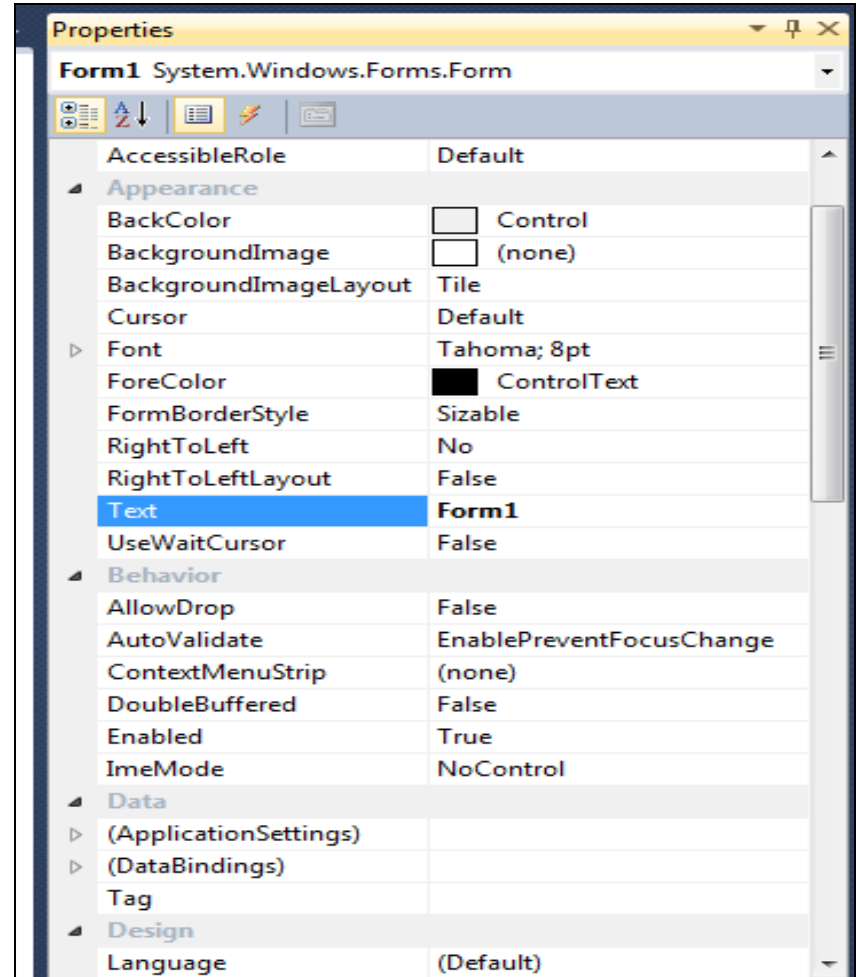
View -> Toolbox

View -> Other Windows -> Properties

البرنامج الأول



شريط الأدوات



شريط الخصائص

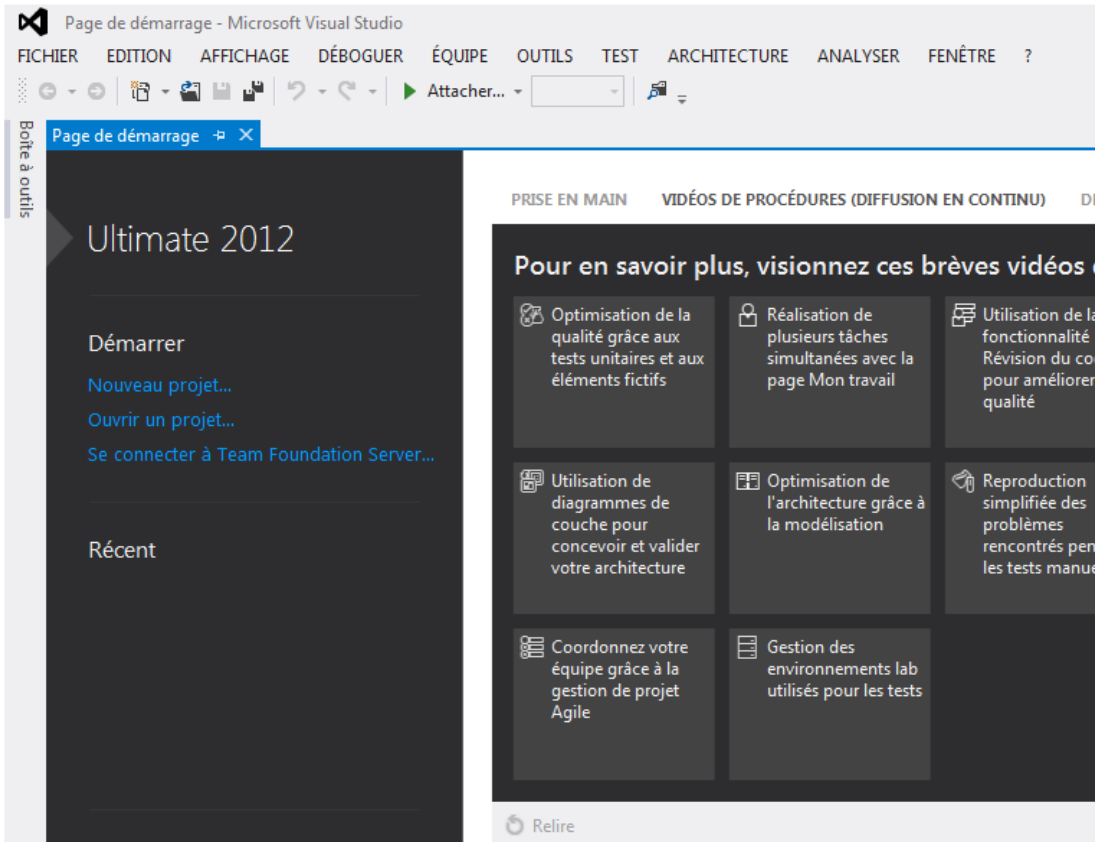
الوجهات الرسومية في الفيجوال سي بلس بلس 2012

FICHER -> Nouveau -> Projet

بعد فتح الفيجوال تظهر الصورة 1 , ثم نتبع الاختيارات التالية

CLR -> Projet vide CLR

عندها تظهر الصورة 2 نتبع الاختيارات التالية



نختار الاسم المناسب ثم مكان الحفظ
الذي نريد حفظ البرنامج فيه وفي
النهاية نقر على الزر

OK

ثم نضيف فورم جديد من القائمة

PROJET -> Ajouter un
nouvel élément

عندها تظهر الصورة 3 فنختار

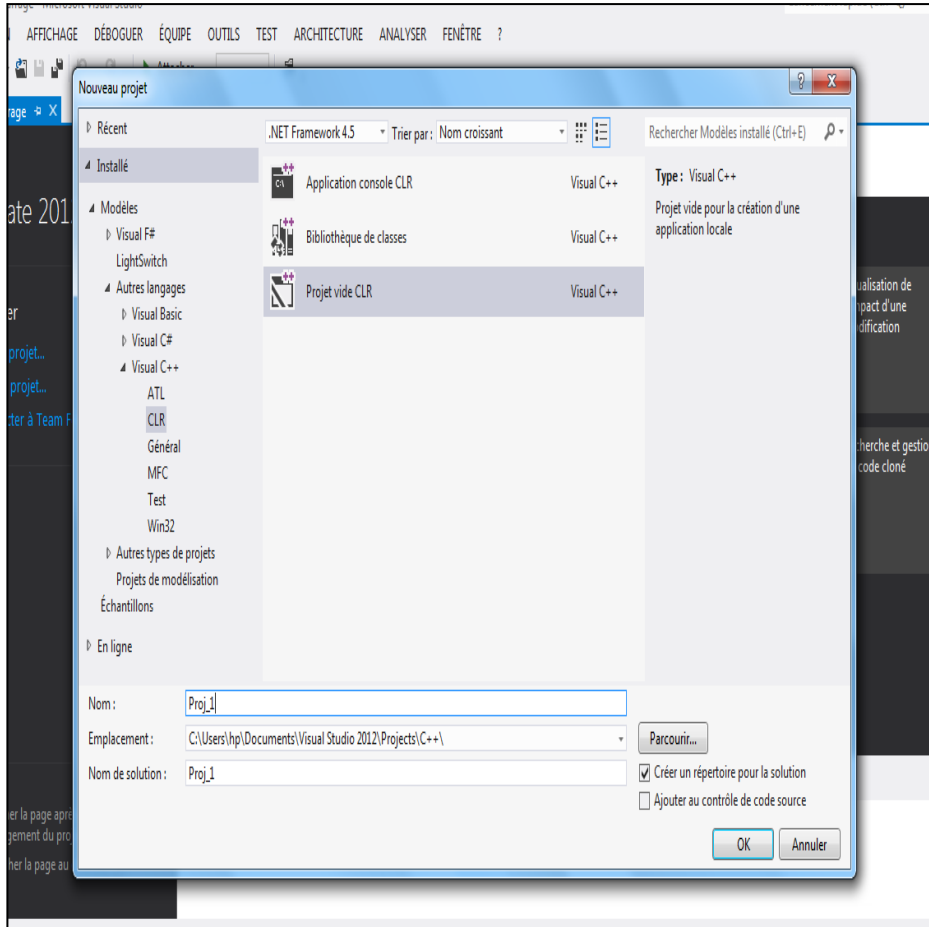
UI -> Windows Form

نختار اسم الفورم وبعدها نقر على الزر

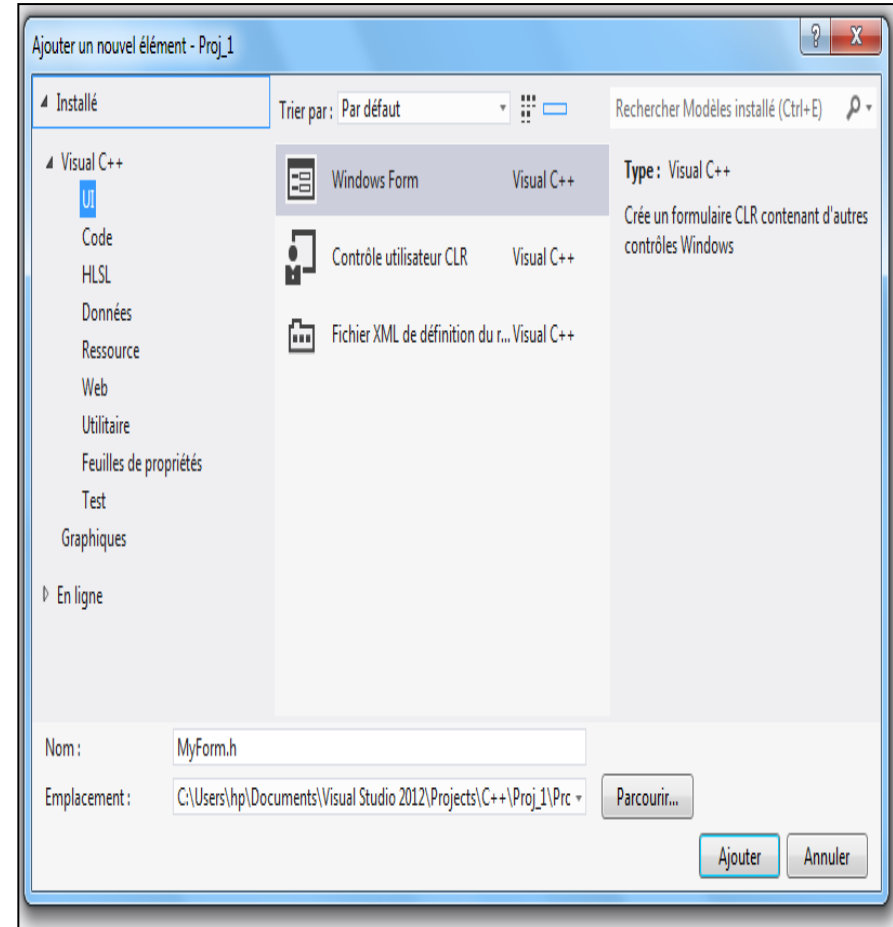
Ajouter

صورة 1

البرنامج الأول



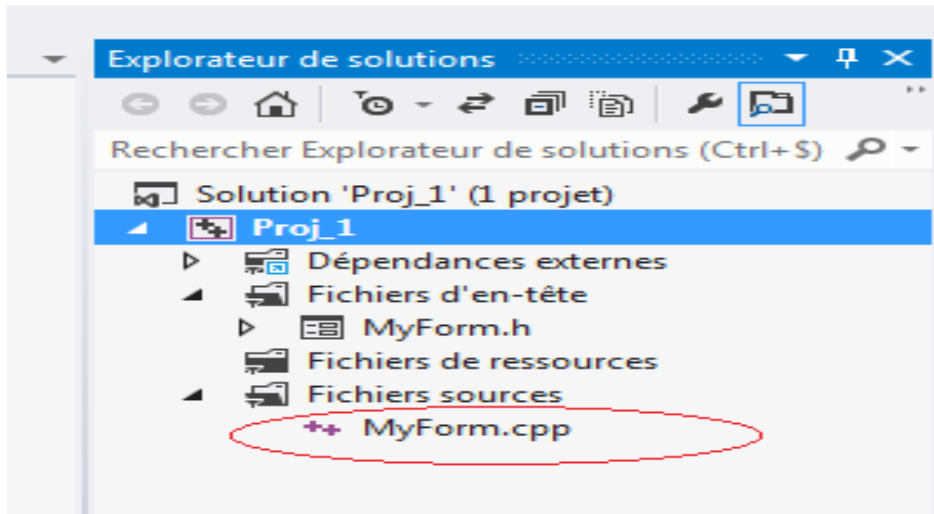
صورة 2



صورة 3

حتى لا تصادف أي مشاكل مع الأدوات الغير مرئية نضيف الي الملف `.cpp` للفورم الكود التالي

البرنامج الأول



الكلمات الملونة بالأحمر تقابل

MyForm

تقابل اسم الفورم الذي اخترته

Proj_1

تقابل اسم المشروع الذي اخترته

.cpp

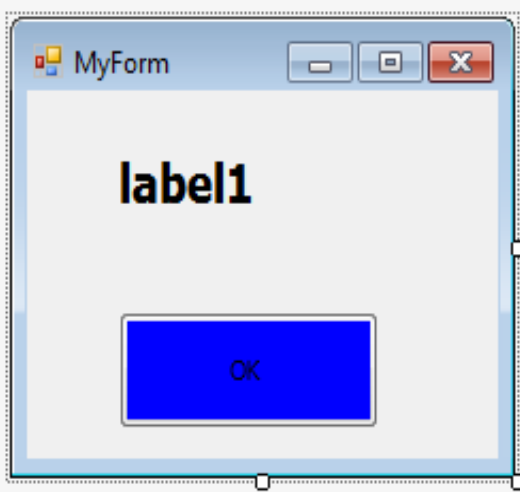
كود الملف

```
#include "MyForm.h"
using namespace Proj_1;
[STAThreadAttribute]
int main(array<System::String ^> ^args)
{
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    Application::Run(gcnew MyForm());
    return 0;
}
```


البرنامج الأول

الآن من شريط الأدوات الخاص بالأدوات البرمجية أضيف أداة الزر إلي الفورم وذلك بسحبها عن طريق الفأرة إلى داخل الفورم
(Button)

من شريط الخصائص نغير بعض خصائص الزر وذلك بعد تحديد الزر .



نغير الخصائص التالية

Text	OK
BackColor	Blue
Name	but_aff

نضيف إلي الفورم أداة أخرى وهي label غير خاصية الخط Font
ضع أي قيمة تريد لحجم الخط . في الأخير نحصل علي الشكل المقابل .

بواسطة الفأرة ننقر مرتين علي الزر الموجود علي الفورم حتى نتمكن من كتابة الكود الذي نريد . نكتب البرنامج التالي

`label1->Text = " C++";`

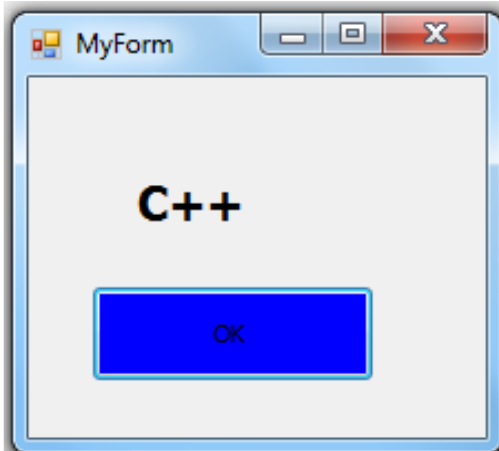
```
95 | #pragma endregion
96 | private: System::Void but_aff_Click(System::Object^ sender, System::EventArgs^ e) {
97 |     label1->Text = "C++";
98 | }
99 | };
100 | }
```

البرنامج الأول

▶ Déboqueur Windows Ic

ننقر علي F5 او على السهم الأخضر الصغير المبين على الصورة .

حتى نتمكن من مشاهدة النتيجة . بالنقر على الزر الموجود على الفورم نحصل على نتيجة البرنامج .



كان المثال السابق بسيط ولكن هو كبداية جيدة للدخول إلي الواجهات الروسية , ادا سبق لك وان عملت علي الواجهات في برامج أخرى مثل جافا أو سي شارب أو دلفي لاشك وانك لاحظت انه يكتب اسم الأداة البرمجية ثم نقطة وبعدها الخاصية أو الميثود الخاص بالأداة مثل

```
label1.Text = " C++";// EXp : C#
```

لكن في لغة السي بلس بلس بدل النقطة نكتب -> مثل

```
label1->Text = " C++";//EXP : C++
```

كما لاحظت فالفروقات بسيطة وهذا جيد لمن كان على اطلاع بالواجهات .

ادا لم تفهم المثال السابق فحاول مجددا حتى لا تصادف مشاكل في طريقنا .

ادا لم تكن هناك أية مشاكل , سأحاول شرح بعض الأدوات الأساسية في لغة فيجوال السي بلس بلس

البرنامج الأول

```
1 #pragma once
2
3 #include " "
4
5 namespace test {
6     using namespace System;
7     using namespace System::ComponentModel;
8     using namespace System::Collections;
9     using namespace System::Windows::Forms;
10    using namespace System::Data;
11    using namespace System::Drawing;
12    /// <summary>
13    /// Summary for Form1
14    /// </summary>
15    public ref class Form1 : public System::Windows::Forms::Form
16    {
17    public:
18        Form1(void)
19        {
20            InitializeComponent();
21            //
22            //TODO: Add the constructor code here
23            //
24        }
25
26    protected:
27        /// <summary>
28        /// Clean up any resources being used.
29        /// </summary>
30        ~Form1()
31        {
32            if (components)
```

الصورة المقابلة تمثل مكان كتابة الأكواد الخاص بأدوات الفورم عند النقر عليها أو باستعمال الأحداث هذا الملف نحصل عليه عن طريق النقر بالزر الأيمن للفأرة على الفورم واختيار مشاهدة الكود أو من خلال النقر المزدوج على أداة موضوعة على الفورم

بملاحظة السطر 15 على الصورة نشاهد أن الفورم عبارة عن فئة مشتقة من فئة أخرى وهي **System: Windows::Forms::Form** ادا الفورم عبارة عن فئة نشاهد في السطر 18 المشكل وفي السطر 30 المهتم , هذا الكلام لا يهم من لا يعرف البرمجة الكائنية التوجه ولكن يسهل الأمر كثيرا على من يعرفها لو أخذنا المثال السابق يمكن اعتبار الزر عبارة عن خاصية تملك نوعا معيناً مثل التصريح بالمتغيرات العادية **int var** شاهد صورة أ واعتبار حدث النقر على الزر عبارة عن ميثود لفئة الفورم شاهد صورة ب في السطر 3 نلاحظ أنه يمكننا تضمين المكتبات التي نحتاج إليها مثل **math.h** وأيضا ملفات **.h** للفئات التي ننشئها نحن أثناء عملنا .

صورة أ

```
        {
            delete components;
        }
    }

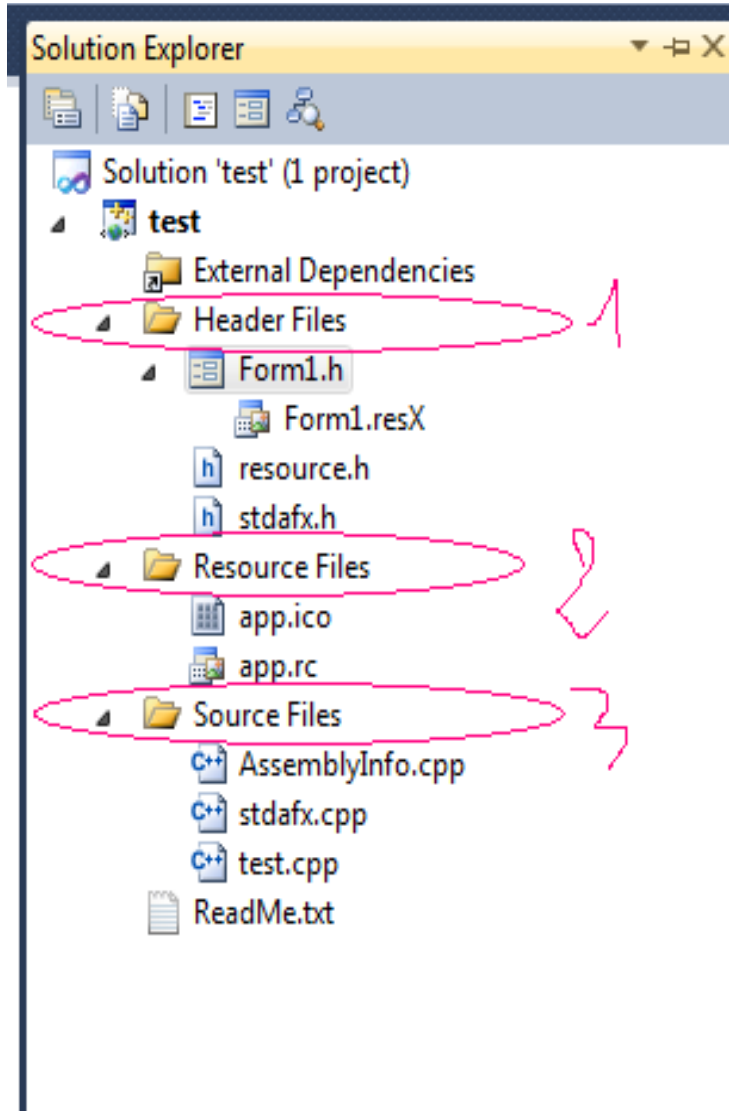
private: System::Windows::Forms::Button^  but_aff;|
private: System::Windows::Forms::Label^  label1;

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>

89 |
90 |     }
91 | #pragma endregion
92 | private: System::Void but_aff_Click(System::Object^ sender, System::EventArgs^ e) {
93 |     label1->Text = "C++";
94 |     }
95 | };
96 | }
97 |
```

صورة ب

Solution Explorer



الرقم 1

هذا الملف يحتوي على الملفات `.h` الخاصة بالبرنامج أو التي ننشئها بأنفسنا أثناء البرمجة الكائنية التوجه .

الرقم 2

هذا الملف يحتوي على الملفات المكملة للبرنامج مثل الصور والأيقونات لتجميل البرنامج .

الرقم 3

هذا الملف يحتوي على الملفات `.cpp` الخاصة بالبرنامج أو التي ننشئها بأنفسنا أثناء البرمجة الكائنية التوجه مثلا .

ملاحظة : يمكن من خلاله فتح ملفات موجودة أو حذفها وأيضا إضافة ملفات أخرى .

شرح بعض الأدوات الأساسية

الفورم

جدول 1

BackColor	تغيير أو اختيار لون للخلفية.
ForeColor	تغيير أو اختيار لون الخط المستعمل داخل الأداة .
Text	اختيار عنوان (يظهر علي الأداة) وللفورم أيضا عنوان.
FormBorderStyle	اختيار نوع وحدود الأداة .
Font	اختيار نوع خط الكتابة للعنوان أو حقل الكتابة .
Name	إعطاء اسم للأداة .
Cursor	شكل سهم الفأرة عند وضعة على الأداة .
BackgroundImage	اختيار صورة للخلفية .
BackgroundImageLayout	طريقة عرض صورة الخلفية .
Icon	اختيار أيقونة للفورم .
MaximizeBox	إخفاء زر التكبير من الفورم .
MinimizeBox	إخفاء زر التصغير من الفورم .
StartPosition	اختيار مكان عرض الفورم علي الشاشة .
WindowState	شكل الفورم عند تنفيذ البرنامج .
MaximumSize , MinimumSize	اختيار القيمة الأعلى والأدنى لطول وعرض الفورم الممكنة .

شرح بعض الأدوات الأساسية

Button

الزر

للزر أيضا خصائص مشتركة مع الفورم لذلك لادعي لتكرار الجدول 1 .
يمكن اعتبار أن الخصائص المشروحة في الجدول 1 مشتركة مع معظم
الأدوات فعندما أكتب جدول 1 أقصد به تلك الخصائص زائد الخصائص
الخاصة بكل أداة .

الزر لديه خصائص جدول 1 وبالنقر عليه مرتين بواسطة الفأرة يأخذنا إلى مكان كتابة الكود الذي ينفذ عند
النقر عليه بواسطة الفأرة .

Enabled	عندما تكون هذه الخاصية خاطئة لايمكن النقر علي الأداة أثناء التنفيذ
Visible	أظهار أو إخفاء الأداة أثناء التنفيذ .
Modifiers	اختيار مجال الرؤيا عام أو خاصالخ....

جدول 2

FlatStyle	شكل حدود الزر هل هي ثلاثية الأبعاد أم غير ذلك .
-----------	---

شرح بعض الأدوات الأساسية

textBox

هذه الأداة عبارة عن حقل للكتابة تستعمل في إدخال المعلومات إلي البرنامج مثل الحقول التي يطلب منا ملؤها أثناء التسجيل في موقع معين .

الخصائص : جدول 1 + جدول 2

BorderStyle	اختيار شكل الحدود .
TextAlign	اختيار موقع النص .
Multiline	تشغيل هذه الخاصية يجعلها متعددة الأسطر .
ScrollBars	اختيار اتجاه شريط التمرير .
AcceptsReturn	قبول خاصية الرجوع إلى السطر .
AcceptsTab	(Tab) قبول الفراغات .
ReadOnly	الخاصية صحيحة يعني أن الحقل للقراءة فقط .

شرح بعض الأدوات الأساسية

label

عبارة عن حقل لإظهار النتائج أو عرض كتابات معينة, مثل اذا أردنا كتابة الرقم السري في الحقل المخصص لذلك فإننا سنجد كتابة تشير إلى ذلك مثلا " أدخل رمز العبور من فضلك" هذه الكتابة من الجيد أن تكون داخل هذه الأداة .

الخصائص هي نفس خصائص **textBox** ماعدا خاصية تعدد الأسطر .

comboBox

حقل لعرض أو إدخال معلومات, تشبه أداة اختيار البلد عند التسجيل في موقع معين .

الخصائص : جدول 1 + جدول 2

DropDownStyle

اختيار شكل الظهور وكذلك إمكانية الكتابة أو لا .

Items

خاصية تسمح بكتابة عناوين الاختيار في الأداة .

DropDownStyle → DropDownList

تحقيق هذه الخاصية يمنع التغير في النص الذي يظهر على الأداة .

شرح بعض الأدوات الأساسية

هذه بعض الدوال المستعملة بكثرة للتعامل مع **comboBox**

comb->Items->Count	إعادة عدد العناصر في comb
comb->Items[i]	إعادة العنصر ذو الرتبة i
comb->Items->Add(T obj)	إضافة عنصر إلى آخر comb
comb->Items->AddRange(T []obj)	إضافة مصفوفة إلى آخر comb
comb->Items->Insert(i , T obj)	إضافة عنصر في الوضعية i
comb->Items->RemoveAt(i)	حذف العنصر ذو الرتبة i
comb->Items->Clear()	حذف جميع العناصر
comb->Items->IndexOf(T obj)	إعادة رتبة العنصر obj
comb->SelectedIndex	إعادة رتبة العنصر المحدد .
comb->SelectedItem	إعادة العنصر المحدد .
comb->Text	إعادة النص الظاهر على comb

أثناء البرمجة لا تنسى التحويلات لان نوعية العناصر المعادة من الدوال قد تختلف في نوعتها مثل دالة التحويل إلي سلسلة حرفية **ToString()**

شرح بعض الأدوات الأساسية

هذه الأداة عبارة عن حقل لعرض المعلومات بشكل أكبر .

listBox

الخصائص : جدول 1 + جدول 2

SelectionMode

هذه الخاصية تمكننا من اختيار طريقة تحديد العناصر الموجودة فيها هل التحديد بشكل فردي أم يمكن تحديد العناصر دفعة واحدة .

Items

خاصية تسمح بإضافة عناصر جديدة إلى الأداة .

ملاحظة : تمتلك هذه الأداة مجموعة من الدوال الجاهزة . وطريقة استعمالها هي نفس الطريقة التي رأيناها مع **comboBox** أي أن الدوال تحمل نفس الاسم , مثال إضافة عنصر جديد تكون بالشكل

```
listBox1->Items->Add("Hichem");
```

شرح بعض الأدوات الأساسية

listView

أداة لعرض المعلومات بشكل مرتب يشبه الجدول .

الخصائص : جدول 1 + جدول 2

View	اختيار شكل العرض .
Gridlines	إذا كانت هذه الخاصية محققة يعني ظهور خطوط تفصل بين الأسطر.
Columns	إضافة أعمدة جديدة إلي الأداة طبعا يمكن اختيار عنوان للعمود وأشياء أخرى.
Items	إضافة عناصر جديدة إلى عمود .
CheckBoxes	تحقيق الخاصية يظهر لنا خانات الاختيار لاختيار الأسطر التي نريد .
FullRowSelect	تحقيق الخاصية يعني إمكانية تحديد سطر معين .
MultiSelect	تحقيق الخاصية يعني إمكانية تحديد عدة سطر.
HotTracking	تحقيق الخاصية يعني تغير لون الأسطر عند تمرير الفأرة عليها .
largeImageList StateImageList SmallImageList	هذه الخصائص تسمح بإضافة صور إلي الأداة على شكل أيقونات .

imageList

الخواص الأخيرة المتعلقة بالصور لا تتعامل إلا مع أداة واحدة , سنشرحها لاحقا اسمها

شرح بعض الأدوات الأساسية

بعض العمليات الأساسية :

إضافة نص إلي عمود (صالحة في حالة تعاملنا مع العمود الأول فقط) :

```
listView1->Items->Add(String^ s);
```

التعامل مع عدة أعمدة : (3 أعمدة مثلا)

```
ListViewItem ^item1 = gcnew ListViewItem("",1);  
item1->SubItems->Add("1");  
item1->SubItems->Add("2");  
item1->SubItems->Add("3");  
  
listView1->Items->Add(item1);
```

البرامتر الثاني في الدالة الموجودة في السطر الأول أقصد الرقم 1 هو ترتيب الصورة في أداة الصور التي نريد إضافتها على شكل أيقونة

```
ListViewItem ^item4 = gcnew ListViewItem("",1);  
item4->SubItems->Add(textBox1->Text);  
item4->SubItems->Add(textBox2->Text);  
item4->SubItems->Add(textBox3->Text);  
  
listView1->Items->Add(item4);
```

هذه الأداة تحتوي على الكثير والكثير من الدوال لا سيعنى شرحها الآن اذا أردت التعمق أكثر عليك أن تبحث عنها أو اكتشفها من الشرح الذي يظهر عند كتابة (->) بعد اسم أداة أو فئة أو مساحة أسماء .

مثال تطبيقي 1

افتح مشروع جديد من نوع Windows Forms Application



غير خاصية **Text** للفورم إلى **EX1**

أضف إلى الفورم 3 من **label** و**غير خاصية Text**

فيها إلى ما تشاهده على الصورة .

أضف 2 من **textBox**

غير الخصائص التالية :

textBox1 **Name** → **txtb_name**

textBox2 **Name** → **txtb_ovr**

BackgroundImage

اختر الصورة التي تريد .

BackgroundImageLayout

Center

FlatStyle

Flat

Name

but_save

ForeColor

Control

أضف 1 من **Button**

غير خصائص الزر التالية
كما في الجدول المقابل :

مثال تطبيقي 1

أضف 1 من ComboBox

غير الخصائص التالية كما في الجدول المقابل :

Name	comb_sp
ForeColor	Blue
DropDownStyle	DropDownList
Items	إعلام ألي رياضيات فيزياء كيمياء فنون

أضف 1 من ListBox

غير الخصائص التالية كما في الجدول المقابل :

Name	listBox_v
ForeColor	Blue
RightToLeft	Yes

ملاحظة : الخاصية الأخيرة حتى نتمكن من الكتابة من اليمين إلى اليسار للكتابة باللغة العربية لمن يريد .

مثال تطبيقي 1

كتابة الكود : قبل ذلك أود أن أتطرق إلى شيء وهو التحويلات بين الأنواع المعروفة سأذكر بعضها.

`int var = System::Int32::Parse(String^);` `String^` → `int`

`Double var = System::Double::Parse(String^);` `String^` → `Double`

توجد طريقة أخرى أسهل وهي : `System::Convert::To` ثم اختيار النوع الذي نريد مثل :

```
int a = System::Convert::ToInt32(textBox1->Text);
```

MessageBox هذه الفئة تمكننا من عرض رسائل أثناء تنفيذ البرنامج وتكمن فائدتها من خلال عرض رسائل الأخطاء التي تحذرنا أو لتنبه المستخدم أو لتأكيد الخيارات .

```
System::Windows::Forms::MessageBox::Show(msg);
```

تستقبل براميثر من النوع سلسلة نصية , سيعرض لنا البرنامج الرسالة مع زر للتأكيد . وهذا ليس النوع الوحيد فهناك دوال بنفس الشكل أعيد تعريفها لتستقبل براميثرات أخرى كعنوان الرسالة و الأيقونة التي تريد إظهارها ونوع الأزرار وعددها .



يمكن أن يكون عبارة عن دالة تعيد قيمة من النوع

```
System::Windows::Forms::DialogResult
```


مثال تطبيقي 1

```
22 public ref class Form1 : public System::Windows::Forms::Form
23 {
24 private:
25     int nb_stud;
26 public:
27     Form1(void)
28     {
29         InitializeComponent();
30         nb_stud=0;
31     }
32 protected:
33     /// <summary>
34     /// Nettoyage des ressources utilisées.
35     /// </summary>
36     ~Form1()
37     {
38         if (components)
39         {
40             delete components;
41         }
42     }
43 private: System::Windows::Forms::Label^ label1;
44 protected:
```

كما تشاهد علي الصورة قمنا بالتصريح بمتغير من النوع عدد صحيح في السطر 25 وظيفته معرفة عدد الأفراد الدين تم تسجيلهم , وفي السطر 30 داخل مشكل الفورم قمنا بإعطاء الصفر كقيمة ابتدائية له , وضعناه داخل مشكل الفورم لأنه هو أول ما ينفذ من البرنامج .

الآن اذهب إلي الفورم وانقر نقرا مزدوجا بالزر الأيسر للفأرة على الزر هذا سيأخذنا إلى مكان كتابة الكود البرمجي . قم بكتابته ونفذ البرنامج

الكود في الصفحة الآتية .

مثال تطبيقي 1

```
try
{
    nb_stud=nb_stud+1;
    String^ name = txtb_name->Text;
    Double ov = System::Double::Parse(txtb_ovr->Text);
    String^ spec = comb_sp->Text;

    if(name != ""&&spec!=""&&ov>0)
    {
        String^ s = nb_stud.ToString() + "\t" + name + "\t" + ov.ToString() + "\t" + spec;
        listBox_v->Items->Add(s);
        listBox_v->Items->Add("-----");
    }
    else{
        System::Windows::Forms::MessageBox::Show(" Error.");
    }
}
catch(...) {
    System::Windows::Forms::MessageBox::Show("Exception.");
}
```

مثال تطبيقي 2

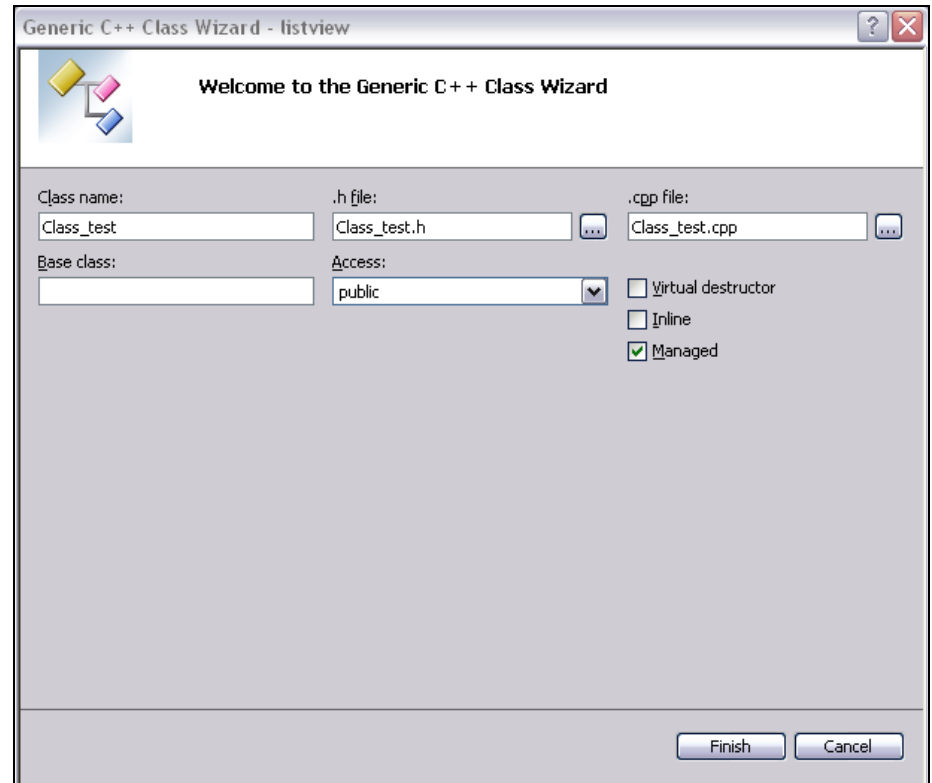
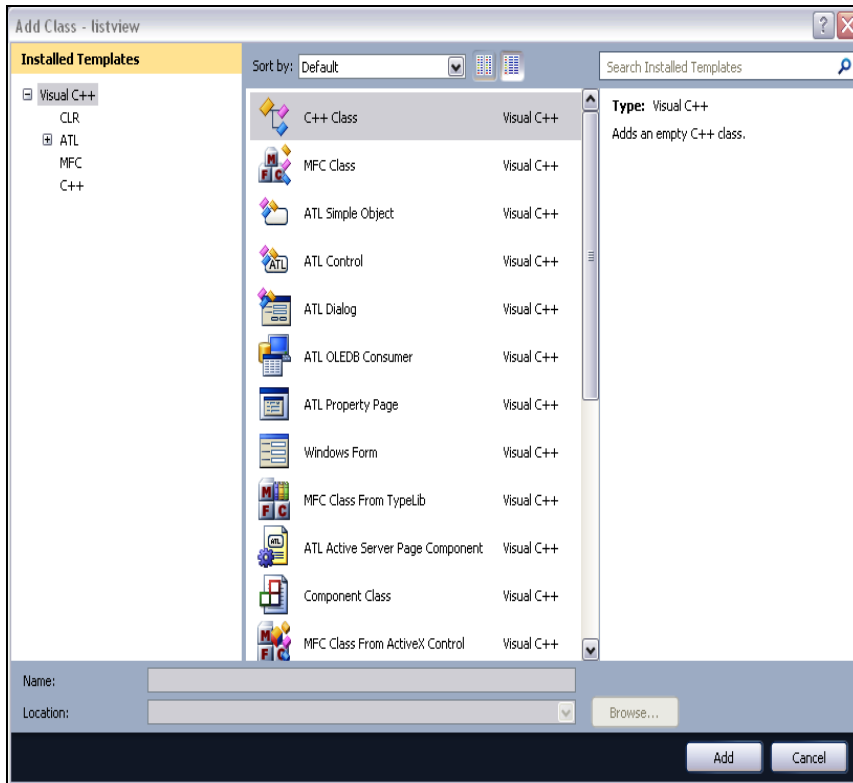
سنعيد نفس المثال السابق لكن هذه المرة سنستعمل البرمجة كائنية التوجه مع ListView

حتى نتمكن من استعمال البرمجة الكائنية التوجه يجب نضيف كلاس إلي البرنامج والطريقة هي

C++ class

ثم نختار

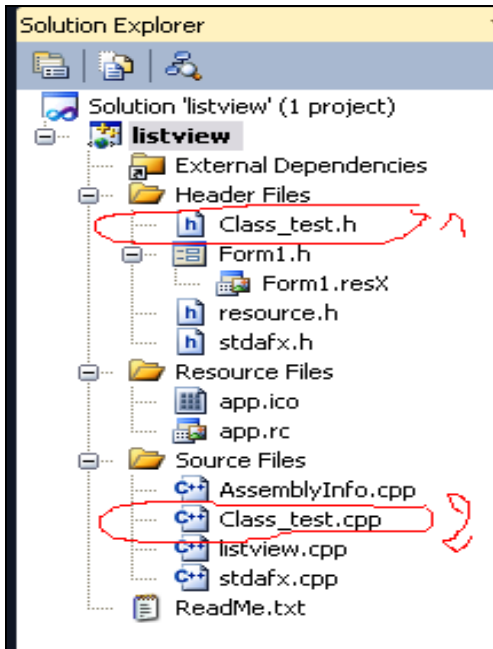
Project -> Add class



Class_test

نختار اسم الكلاس وأنا اخترت

مثال تطبيقي 2



بعد اضافة الكلاس سيظهر في **Solution Explorer** **.h** and **.cpp**

لمزيد من التفصيل في البرمجة الكائنية التوجه عليك ببذل الجهد لدراستها إن لم تكن قد اطلعت عليها من قبل .

نعد إلى الواجهة الرسومية نحذف **listBox** ونضع بدلها **listview**

أضيف أيضا الأداة **imageList**

غير الخصائص التالية للأداة **listview1**

Name	Listview_v
ForeColor	Blue
View	Details
Gridlines	True
HotTracking	True
SmallImageList	imageList1
FullRowSelect	True

مثال تطبيقي 2

من الخاصية Columns أضيف أربعة أعمدة وغير خاصية text لكل عمود على الترتيب

(الأول لا تكتب شيء) الاسم المعدل التخصص وخاصة عرض كل عمود Width إلى 100

بالنسبة للأداة `imageList1` من الخاصية Images اختر الصور التي تحب ظهورها على

شكل أيقونات لأسطر الأداة لعلمكم كل صورة تمتلك ترتيب معين في جدول داخل أداة الصور مثلا الصورة الأولى تحمل الترتيب 0 .

The screenshot shows a Windows form titled 'Form1' with the following elements:

- Three text boxes labeled 'الاسم', 'معدل', and 'التخصص'.
- A dropdown menu for 'التخصص'.
- An image of a person with a green plus sign.
- A table with columns 'الاسم', 'المعدل', and 'التخصص'.

الاسم	المعدل	التخصص

كتابة الكود

في الصفحة الموالية ستجد كود الكلاس .

أنوه فقط أننا استعملنا النوع `String^` بدل من

`std::string`

وهذا حتى نتفادي التحويلات التي قد تصعب من عملنا .

الحجز في الذاكرة للأنواع التي تمتلك العلامة `^`

`= gcnew`

مثال تطبيقي 2

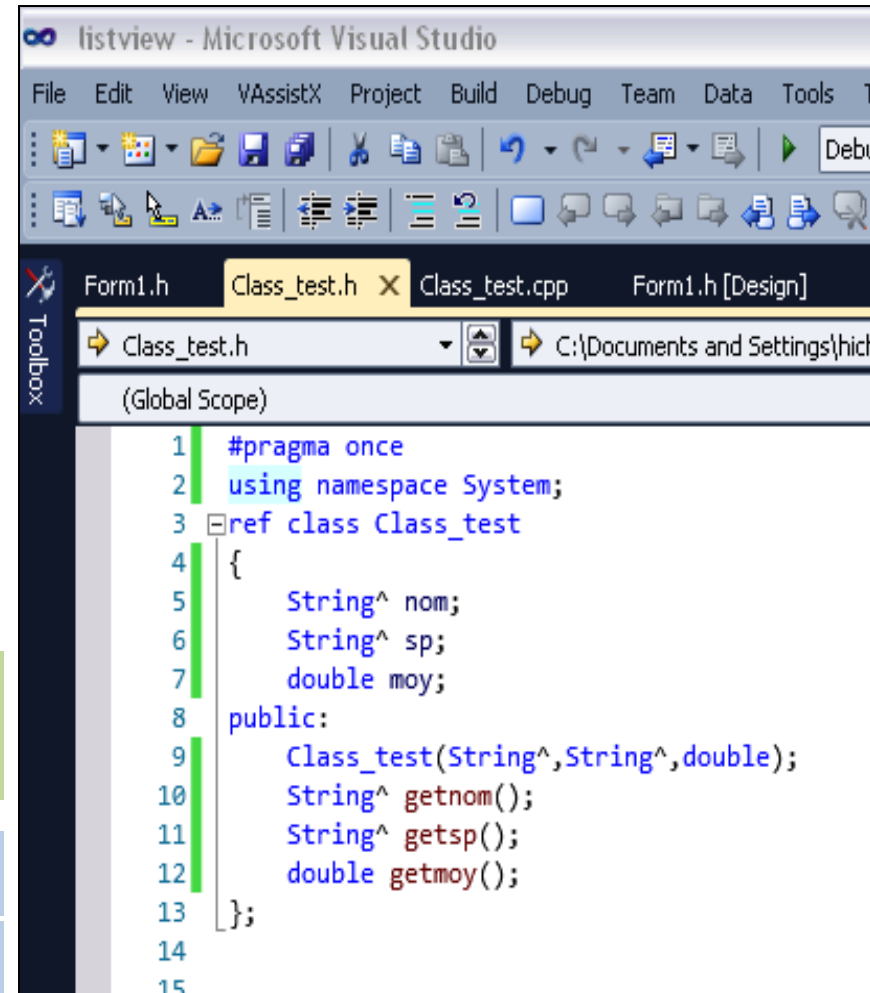
```
#pragma once
using namespace System;
ref class Class_test
{
    String^ nom;
    String^ sp;
    double moy;

public:
    Class_test(String^,String^,double);
    String^ getnom();
    String^ getsp();
    double getmoy();
};
```

تعمدت وضع الصورة حتى يتضح الأمر بشكل جيد أنسخ الكود إلى الملف **.h**

استعملنا مساحة الأسماء `using namespace System;`

من أجل استعمال النوع `String^`



The screenshot shows the Microsoft Visual Studio IDE with the 'listview - Microsoft Visual Studio' window open. The 'Class_test.h' file is selected in the Solution Explorer. The code in the editor is as follows:

```
1 #pragma once
2 using namespace System;
3 ref class Class_test
4 {
5     String^ nom;
6     String^ sp;
7     double moy;
8 public:
9     Class_test(String^,String^,double);
10    String^ getnom();
11    String^ getsp();
12    double getmoy();
13 };
14
15
```

عند التعامل مع الواجهات الرسومية الكلمة المفتاحية `ref` ضرورية في بداية الكلاس لا تنسى .

مثال تطبيقي 2

```
#include "StdAfx.h"
#include "Class_test.h"

Class_test::Class_test(String^ nom ,String^ sp,double moy)
{
    this->nom=nom;
    this->sp=sp;
    this->moy=moy;
}

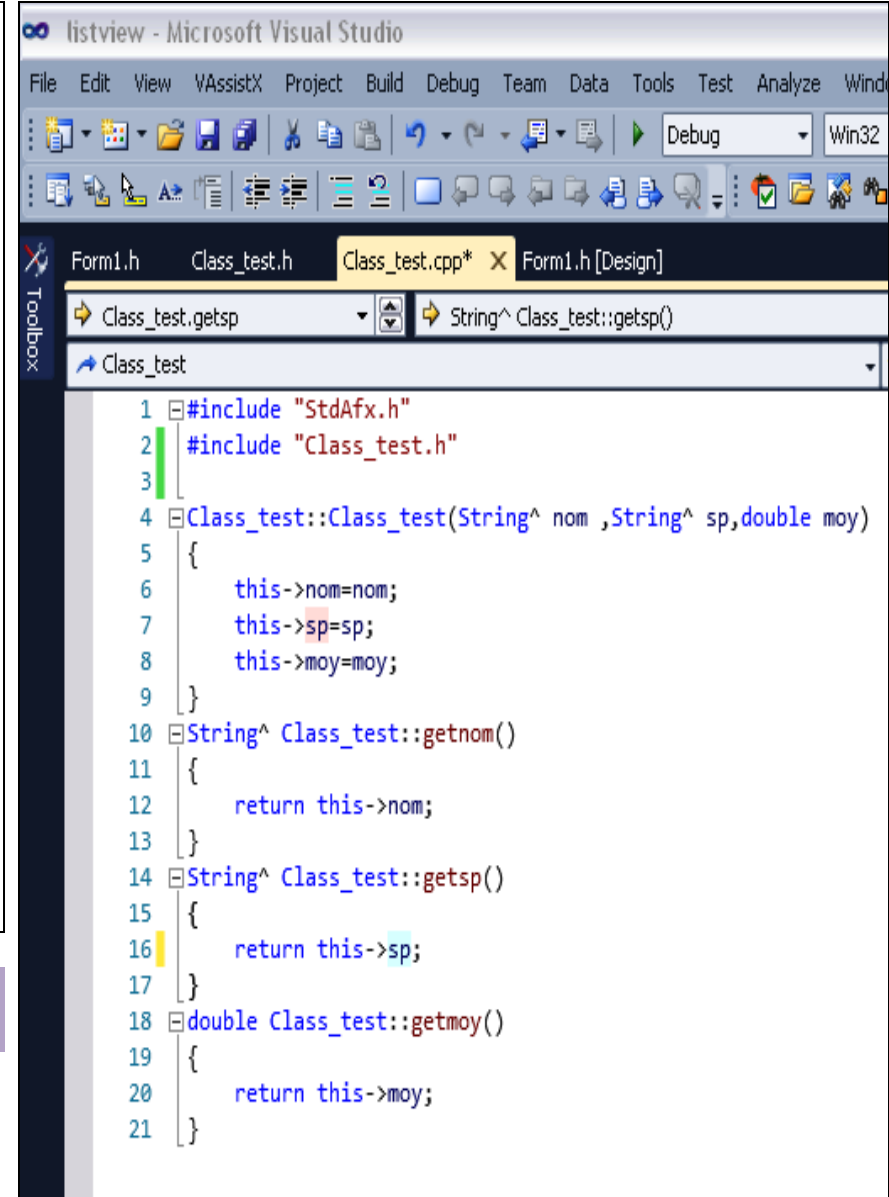
String^ Class_test::getnom()
{
    return this->nom;
}

String^ Class_test::getsp()
{
    return this->sp;
}

double Class_test::getmoy()
{
    return this->moy;
}
```

.cpp

انسخ الكود إلي الملف



```
listview - Microsoft Visual Studio
File Edit View VAssistX Project Build Debug Team Data Tools Test Analyze Wind
Debug Win32
Form1.h Class_test.h Class_test.cpp* X Form1.h [Design]
Class_test.getsp String^ Class_test::getsp()
Class_test
1 #include "StdAfx.h"
2 #include "Class_test.h"
3
4 Class_test::Class_test(String^ nom ,String^ sp,double moy)
5 {
6     this->nom=nom;
7     this->sp=sp;
8     this->moy=moy;
9 }
10 String^ Class_test::getnom()
11 {
12     return this->nom;
13 }
14 String^ Class_test::getsp()
15 {
16     return this->sp;
17 }
18 double Class_test::getmoy()
19 {
20     return this->moy;
21 }
```

مثال تطبيقي 2

```
15 public ref class Form1 : public System::Windows::Forms::Form
16 {
17     int ind_img;
18 public:
19     Form1(void)
20     {
21         InitializeComponent();
22         ind_img=0;
23         //
24         //TODO: Add the constructor code here
25         //
26     }
27 }
```

```
1 #pragma once
2 #include "Class_test.h"
```

قم بالتصريح بمتغير من نوع عدد صحيح كما في السطر 17 وأعطيه قيمة ابتدائية كما في السطر 22 سنستعمل هذا المتغير لتحديد ترتيب الصور التي سنظهرها على شكل أيقونات **لا تنسى السطر 2**.

```
String^ nom = txtb_name->Text;
String^ sp=comb_sp->Text;
double moy=System::Double::Parse(txtb_ovr->Text);
Class_test^ cls= gcnew Class_test(nom,sp,moy);
ListViewItem ^item1 = gcnew
ListViewItem("",ind_img);
item1->SubItems->Add(cls->getnom());
item1->SubItems->Add(cls->getsp());
item1->SubItems->Add(cls->getmoy()+"");
listView1->Items->Add(item1);

ind_img++;
if(ind_img==3)
    ind_img=0;
```

بعد النقر المزدوج على الزر أنسخ الكود المقابل إلي داخله .
في السطر الرابع عرفنا متغير من نوع الكلاس الذي عرفناه سابقا ثم قمنا بحجز الذاكرة له مرفقا بالبرامترات لأن الكلاس الذي عرفناه يحتوي علي مشكل .

في كل مرة أرجع متغير ترتيب الصور إلي 0 عند وصوله الي 3 لأنني اخترت 3 صور فقط في أداة الصور .

مثال تطبيقي 2

	الاسم	التخصص	المعدل
	merieme	فزياء	18,77
	hichem	اعلام ألي	16,75
	oussama	اعلام ألي	15,23
	imene	اعلام ألي	17,42
	kheled	فنون	11,59
	saleh	كيميا	12,07
	ahlem	رياضيات	15,33
	ahlem	رياضيات	15,33

النتيجة

مثال بسيط لكن بإمكانك عمل أشياء أكثر أهمية
موظفا ما تعلمناه معا الى الآن

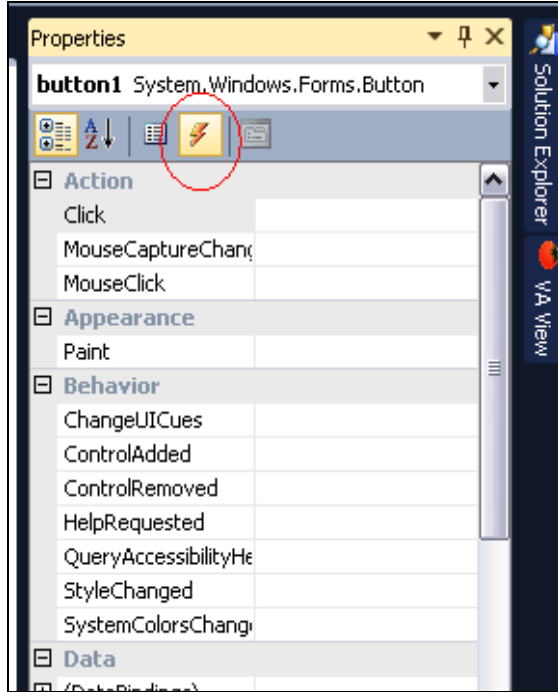
```
String^ nom = txtb_name->Text;
String^ sp = comb_sp->Text;
double moy = System::Double::Parse(txtb_ovr->Text);

ListViewItem ^item1 = gcnw ListViewItem("", ind_img);
item1->SubItems->Add(nom);
item1->SubItems->Add(sp);
item1->SubItems->Add(moy+"");
listView1->Items->Add(item1);
ind_img++;
if(ind_img == 3)
    ind_img=0;
```

إذا كنت من الذين لا يتقنون البرمجة الكائنية
التوجه إليك الكود الذي تكتبه بداخل الزر .

الأحداث

بالنقر على الأيقونة التي عليها دائرة كما في الصورة من قائمة الخصائص ننتقل عندها الى قائمة الأحداث الخاصة بالأداة .



الأحداث هي كل الأمور التي تطرأ على الأداة مثل النقر عليها بواسطة الفأرة أو النقر المزدوج أو تمرير الفأرة عليها أو تغير خاصية ما من خواص الأداة فادا تحقق حدث ما وأردنا استعماله في برنامجنا ما علينا سوى كتابة الكود المقابل له , فمثلا النقر المزدوج بالفأرة علي الزر أثناء التصميم ثم كتابة الكود المتعلق بهذا الزر ما هو إلا حدث النقر على الزر كان باستطاعتنا أن نتوجه إلي قائمة الأحداث الخاصة بالزر ونختار منها حدث النقر **Click** ثم ننقر نقرا مزدوجا بالفأرة عليه سيفتح لنا البرنامج مكان كتابة الكود الخاص بهذا الزر كما كان يحدث في السابق .

بعض أحداث الفورم

Load

أثناء تشكل الفورم .

FormClosing

أثناء غلق الفورم .

FormClosed

بعد غلق الفورم .

افتح برنامج جديد من نوع الواجهات الرسومية ثم اذهب الى أحداث الفورم

مثال 1

Load

```
System::Windows::Forms::MessageBox::Show("Evt Load", "Load");
```

FormClosing

```
System::Windows::Forms::MessageBox::Show("Evt FormClosing", "FormClosing");
```

```
System::Windows::Forms::DialogResult réponse =  
System::Windows::Forms::MessageBox::Show("Voulez-vous vraiment quitter l'application", "Closing",  
System::Windows::Forms::MessageBoxButtons::YesNo,  
System::Windows::Forms::MessageBoxIcon::Question);
```

```
if (réponse == System::Windows::Forms::DialogResult::No)  
    e->Cancel = true;
```

FormClosed

```
System::Windows::Forms::MessageBox::Show("Evt FormClosed", "FormClosed");
```

كما تلاحظ في كل حدث معين كتبنا الكود المقابل له وهو عبارة عن إظهار رسائل, قد تستفيد منه في برامجك مثل الحدث الأول قد تستفيد منه مثلا في التأكد من صحة الاتصال مع قاعدة البيانات أو قراءة التاريخ وعمل شيء يتناسب مع ذلك اليوم هذه عبارة عن أمثلة بسيطة وهناك الكثير .

الأحداث

أحداث الفأرة هي الأحداث المتعلقة بالفأرة كتمرير الفأرة على الأداة أو وضع الفأرة على الأداة... الخ..

MouseDown	عند النقر بالزر الأيسر للفأرة على الأداة .
MouseEnter	إذا كان موضع الفأرة على الأداة .
MouseLeave	إذا سحبنا الفأرة من على الأداة .
MouseMove	إذا كانت الفأرة تتحرك على الأداة .
MouseUp	عند إفلات الزر الأيسر للفأرة من الضغط .

label

أضيف للمثال 1, 2 من أداة

مثال 2

label1

Text احداثيات الفأرة على الفورم ,

Label2

Text

MouseMove

ثم اذهب الى احداث الفورم واكتب الكود التالي في الحدث

```
label2->Text = "( "+e->X +" , "+ e->Y +" )";
```

أضيف للمثال 2 الأدوات التالية وغير الخصائص المطلوبة

مثال 3

Button

Text	Delete All
BackColor	Gray

ListBox

اكتب الكود التالي بداخل الزر طبعا بعد النقر المزدوج عليه

```
listBox1->Items->Clear();
```

ثم الى احداث الزر التالية واكتب لكل حدث الكود المقابل له ,

MouseMove

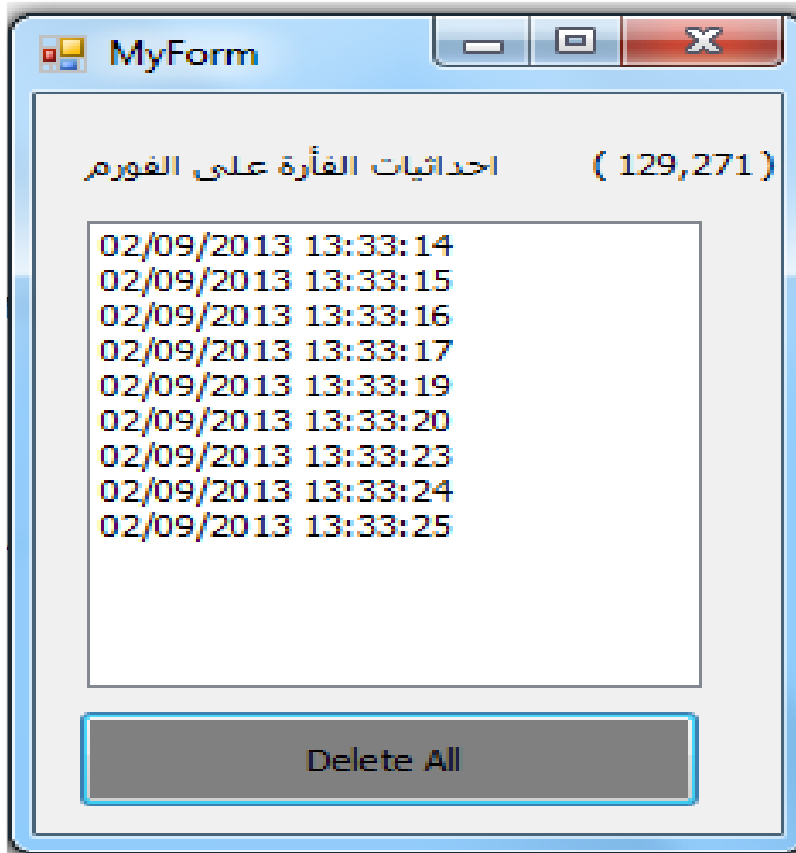
```
button1->BackColor=Color::Blue;
```

MouseLeave

```
button1->BackColor=Color::Gray;
```

MouseEnter

```
listBox1->Items->Add(DateTime::Now+"");
```



النتيجة هي الصورة الظاهرة امامكم

أولا نظهر في كل مرة احداثيات الفأرة على الفورم .

ثانيا كلما سحبنا الفأرة الى الأداة نسجل تاريخ ووقت دخول الفأرة.

ثالثا كلما كانت الفأرة على الزر يتغير لونه الى الأزرق .

شرح بعض الأدوات الأساسية

RadioButton

عبارة عن أزرار للاختيار مثل الذي نجده في المواقع لاختيار الجنس ذكر أو أنثى ولعلك لاحظت أنه يتم اختيار واحد فقط فلو أنك اخترت ذكر فان اختيار أنثى يلغى والعكس ولجعل برنامجنا متعدد الاختيارات الأحذية نستعين بالأداة **GroupBox**.

الخصائص تشبه التي مرت معنا سابقا في مضمونها وربما الإضافة هي الخاصية التي تجعل الاختيار صحيح افتراضيا وهي **Checked**

يمكن التحقق من الاختيار بالشكل `if(radioButton->Checked)`

CheckBox

أيضا تستعمل للاختيار لكن تتميز عن سابقتها كونها تسمح بالاختيار المتعدد .

مثال 1

افتح برنامج جديد للواجهات الرسومية وضع الأدوات التالية مع خصائصها

GroupBox1

Text

لون الفورم

بداخلها ضع 3 من أداة RadioButton

GroupBox2

Text

عنوان الفورم

نفس الشيء أيضا .

شرح بعض الأدوات الأساسية

RadioButton1

Text AliceBlue

RadioButton2

Text Orchid

RadioButton3

Text. GreenYellow

RadioButton4

Text C++

RadioButton5

Text C#

RadioButton6

Text Python

ستلاحظ هذه النتيجة بعد كتابة الأكواد الموالية .

ضع أيضا الأدوات التالية

Panel

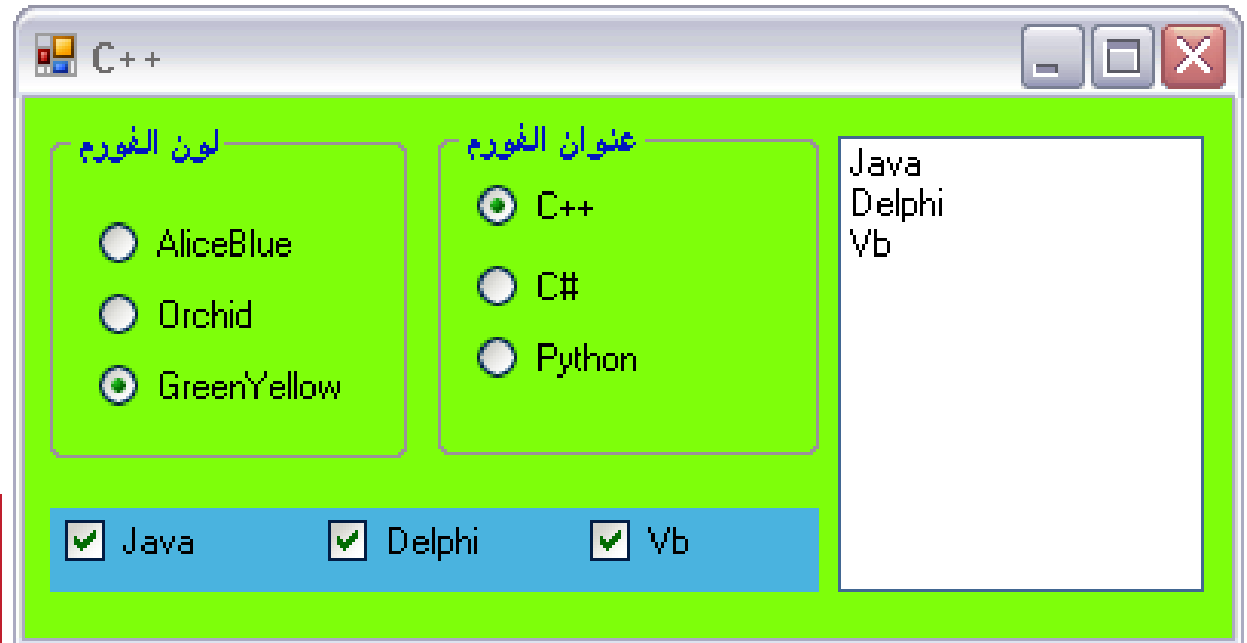
BackColor

اختر لون تفضله

بداخلها ضع 3 من أداة CheckBox

غير خاصية النص لها كما في الصورة (Text)

ListBox



شرح بعض الأدوات الأساسية

من قائمة الأحداث أكتب لكل أداة الكود المقابل لها في الحدث

RadioButton1

```
Form1::BackColor=Color::AliceBlue;
```

RadioButton2

```
Form1::BackColor=Color::Orchid;
```

RadioButton3

```
Form1::BackColor=Color::GreenYellow;
```

RadioButton4

```
Form1::Text=radioButton4->Text;
```

RadioButton5

```
Form1::Text=radioButton5->Text;
```

RadioButton6

```
Form1::Text=radioButton6->Text;
```

CheckedChanged

CheckBox1

```
listBox1->Items->Add(checkBox1->Text);
```

CheckBox2

```
listBox1->Items->Add(checkBox2->Text);
```

CheckBox3

```
listBox1->Items->Add(checkBox3->Text);
```

شرح بعض الأدوات الأساسية

عبارة عن أشرطة للتمرير أو قد يستعملنا للحصول على قيم عن طريق المؤشر والفرق بينهما بسيط الأول أفقي والثاني عمودي .

HScrollBar

VScrollBar

نفس الخصائص ولكن تغير القيمة يكون بالنقر وليس بالسحب .

NumericUpDown

Value	القيمة التي عليها المؤشر
Minimum	القيمة الأدنى التي يصل اليها المؤشر
Maximum	القيمة الأقصى التي يصل اليها المؤشر
SmallChange	القيمة الصغرى التي يتم بها التغير
LargeChange	القيمة الكبرى التي يتم بها التغير

الاداة	الخاصية	القيمة
Label1	Text	hScrollBar :
Label2	Text	?
Label3	Text	vScrollBar :
Label4	Text	?
Label5	Text	NumericUpDown :
Label6	Text	?

مثال 1

افتح مشروع جديد وضع الادوات التي في الجدول وغير الخصائص المطلوبة . يمكن أن تستعين بالصورة في الصفحة الموالية .

شرح بعض الأدوات الأساسية

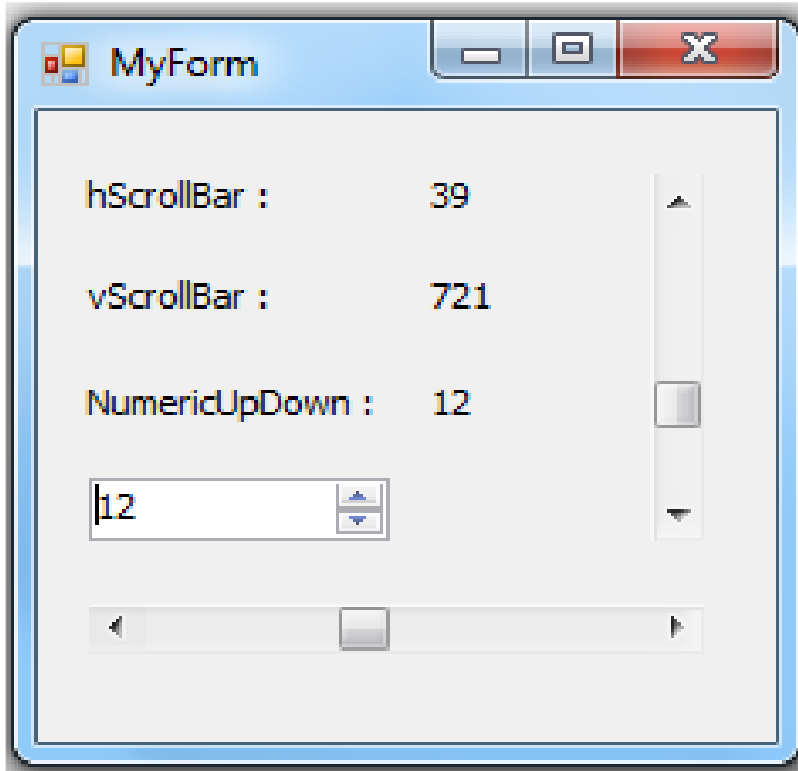
ضع أيضا الأدوات التالية

HScrollBar

VScrollBar

NumericUpDown

غير الخصائص التي في الجدول للأدوات الثلاثة وبنفس الترتيب



Value	50	0	0
Minimum	0	0	0
Maximum	100	1000	50
SmallChange	1	1	1
LargeChange	10	10	10

شرح بعض الأدوات الأساسية

بعد النقر المزدوج بواسطة الفأرة وبالزر الأيسر لها على كل أداة من الأدوات الثلاث نكتب الكود المقابل لكل أداة .

HScrollBar1

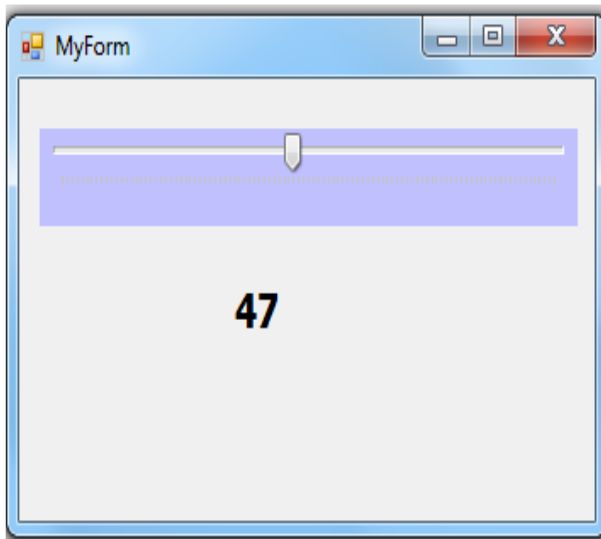
```
label2->Text = hScrollBar1->Value.ToString();
```

VScrollBar1

```
label4->Text = vScrollBar1->Value.ToString();
```

NumericUpDown1

```
label6->Text = numericUpDown1->Value.ToString();
```



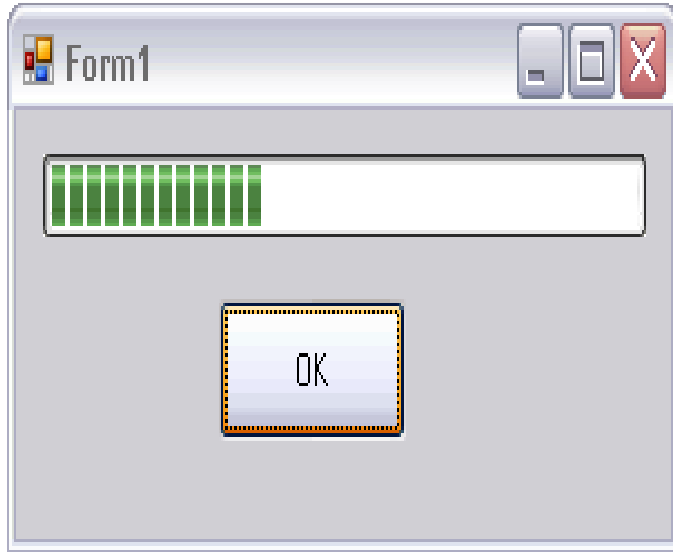
ملاحظة : الأداة التي في الصورة عملها مثل الأدوات الثلاث السابقة ولها نفس خصائصهم .

أعطيتها لون مختلف حتى تظهر علي الصورة واسمها

TrackBar

شرح بعض الأدوات الأساسية

ProgressBar يشبه الشريط الذي يعطينا نسبة التحميل . ويستعمل لإعطاء نسبة انجاز عمل معين .



ProgressBar

مثال افتح مشروع جديد ثم أضيف الأدوات

و أداة الزر غير خاصية النص للزر إلى OK

Minimum

القيمة الابتدائية .

Maximum

القيمة النهائية القصوى .

Style

اختيار شكل الإشارة هل هي خطوط متقطعة أو مستمرة .

Value

القيمة التي يشير اليها المؤشر .

Minimum = 0

Maximum = 1000

غير الخاصيتين التاليتين

```
10 using namespace System::Data;
11 using namespace System::Drawing;
12 using namespace System::Threading;
13 /// <summary>
14 /// Description résumée de Form1
```

أكتب الكود الخاص بالزر

```
for(int i=0;i<1001;i++)
{
    progressBar1->Value = i;
    System::Threading::Thread::Sleep(10);
}
```

أضيف مساحة الأسماء المعينة في الصورة السطر 12

`using namespace System::Threading;`

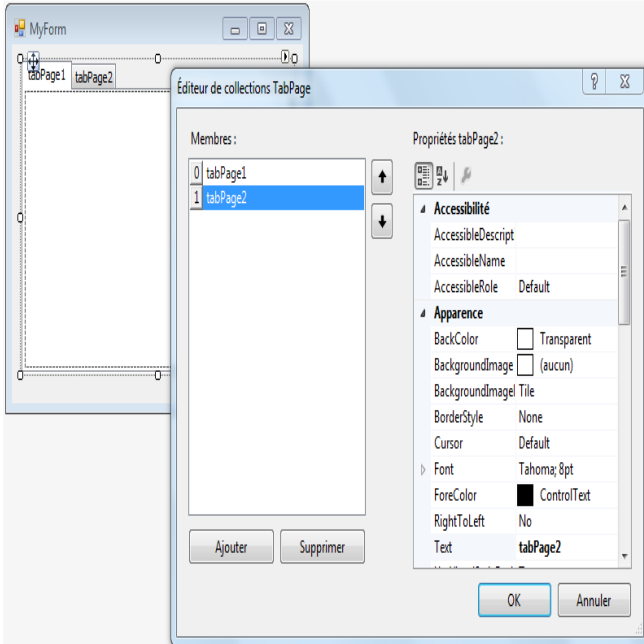
10 تمثل زمن الانتظار

شرح بعض الأدوات الأساسية

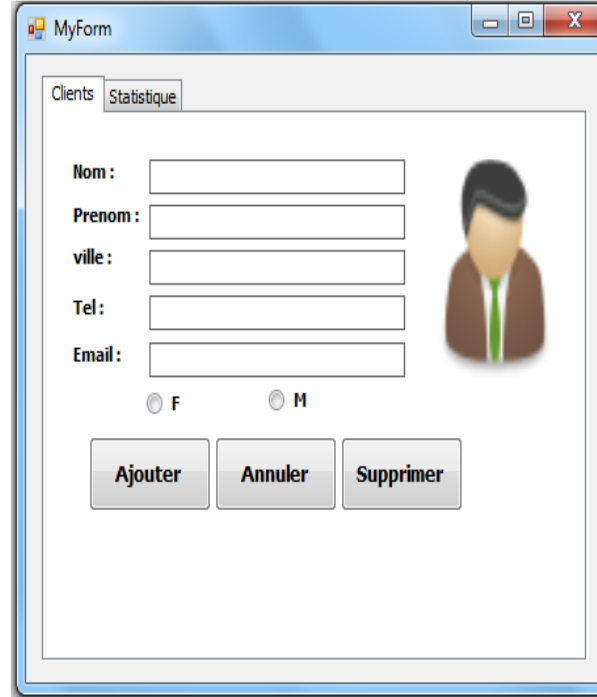
عبارة عن حاوية للأدوات تمكننا من فتح صفحات متعددة في نفس الفورم **TabControl**

الخصائص أغلبها مرت معنا لذلك لا داعي لان نعيدها وأهم خاصية هي (صورة 1) **TabPage**

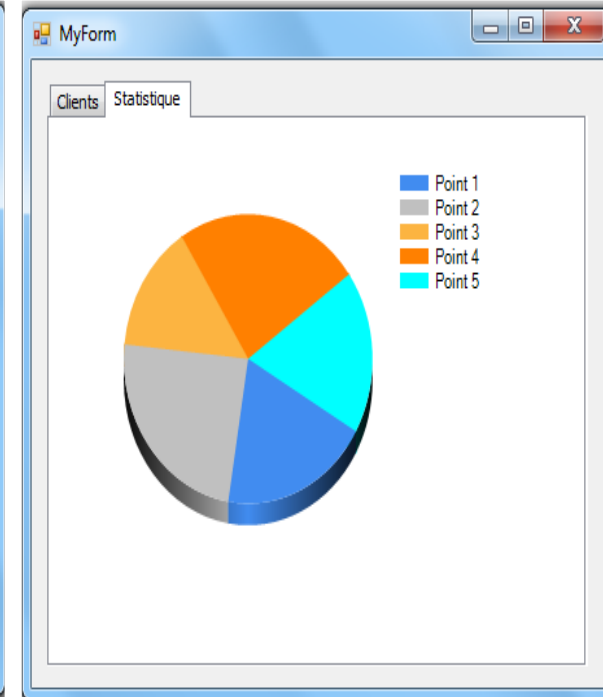
من خلال هذه الخاصية يمكننا اضافة صفحات جديدة أو حذفها , أيضا يمكننا اعطاء الخصائص الخاصة بكل صفحة اللون والخط والصورةويمكننا تصميم كل صفحة على حدى بواسطة الادوات المعروفة



صورة 1



صفحة 1



صفحة 2

شرح بعض الأدوات الأساسية

هذه الاداة تسمح لنا بتمثيل الأشجار مثل شجرة العائلة , المهم الأشجار معروفة في الخوارزميات سواء الثنائية او المتعددة .

TreeView

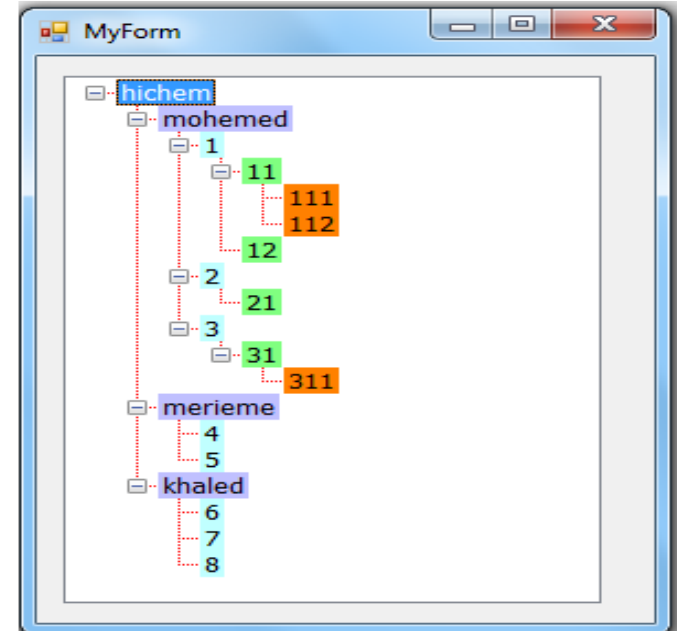
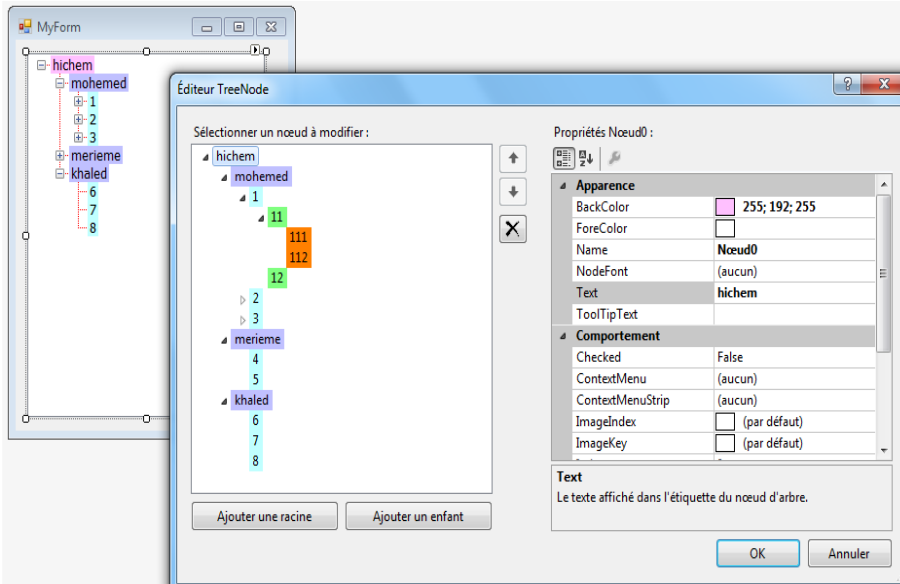
Nodes أهم خاصية تسمح بإضافة عناصر جديدة أو أبناء للعناصر الموجودة .

lineColor اختيار لون الخط الذي يربط بين العناصر .

HotTracking تلون العناصر أثناء تمرير الفأرة عليها.

ItemHeight اختيار المسافة التي تفصل بين العناصر .

CheckBoxs اظهار صناديق الاختيار .



مثال 1

Nodes صورة عند فتح الخاصية

شرح بعض الأدوات الأساسية

افتح مشروع جديد واطيف اليه الادوات التالية

مثال 2

Button1

Text

View

Button2

Text

Clear

TreeView1

ForeColor

Black

LineColor

Red

HotTracking

True

كتابة الكود

Button2

```
treeView1->Nodes->Clear();
```

Button1

شرح بعض الأدوات الأساسية

```
array<TreeNode ^> ^ Wilaya = { gcnew TreeNode("Jijel"),
    gcnew TreeNode("Adrar"),
    gcnew TreeNode("Blida"),
    gcnew TreeNode("Tizi Ouzou") };
TreeNode ^ Algerie = gcnew TreeNode("Algerie", Wilaya);

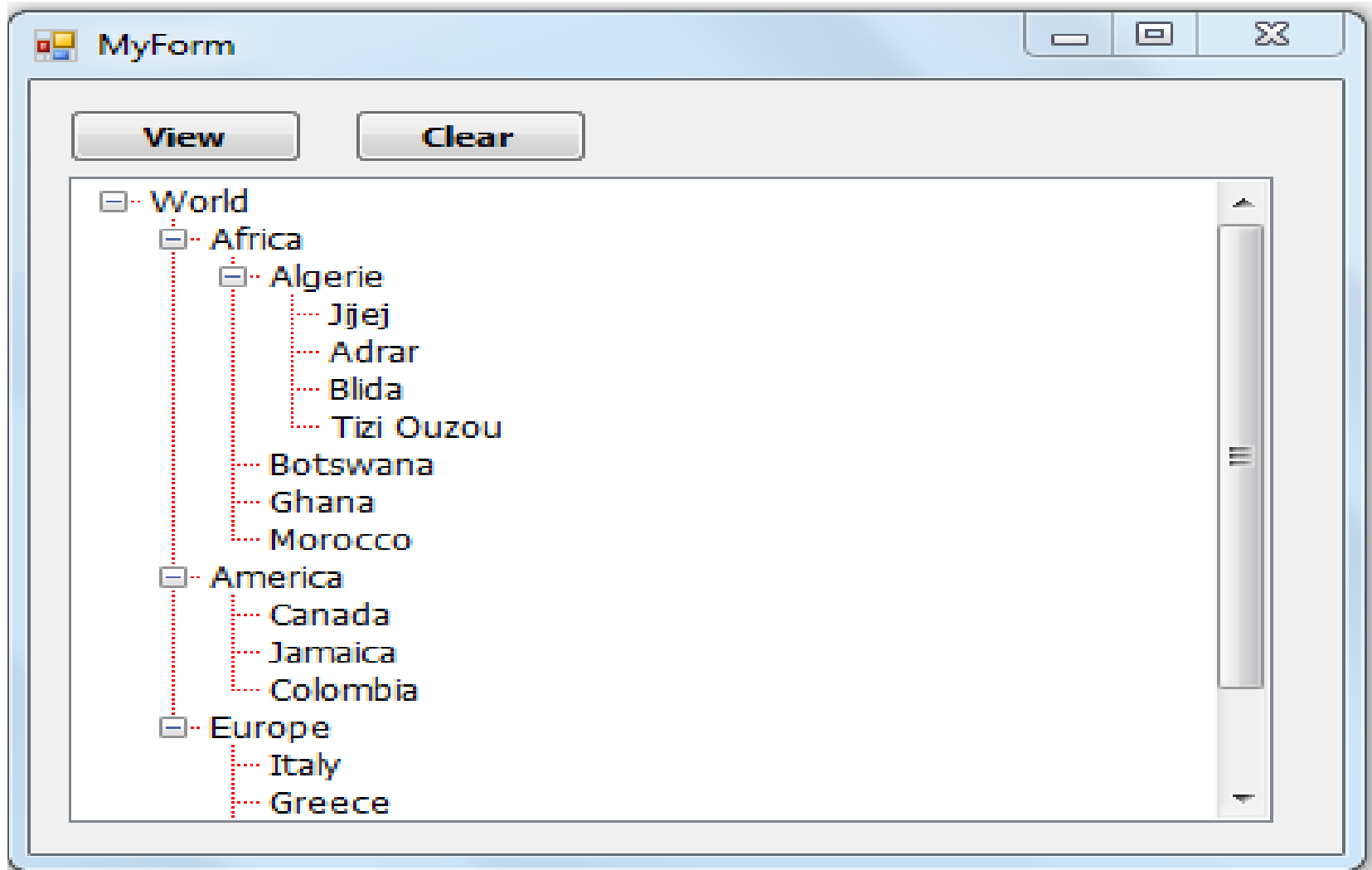
array<TreeNode ^> ^ nodAfricans = { Algerie,
    gcnew TreeNode("Botswana"),
    gcnew TreeNode("Ghana"),
    gcnew TreeNode("Morocco")};

array<TreeNode ^> ^ nodAmericans = { gcnew TreeNode("Canada"),
    gcnew TreeNode("Jamaica"),
    gcnew TreeNode("Colombia")};

array<TreeNode ^> ^ nodEuropeans = { gcnew TreeNode("Italy"),
    gcnew TreeNode("Greece"),
    gcnew TreeNode("Spain"),
    gcnew TreeNode("England") };

TreeNode ^ nodAfrica = gcnew TreeNode("Africa", nodAfricans);
TreeNode ^ nodAmerica = gcnew TreeNode("America", nodAmericans);
TreeNode ^ nodEurope = gcnew TreeNode("Europe", nodEuropeans);

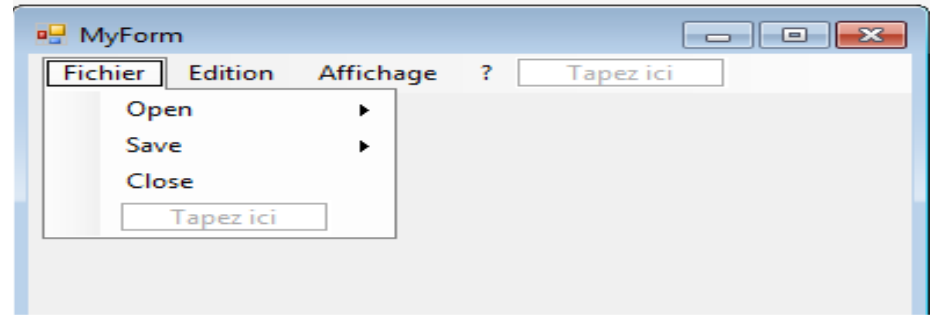
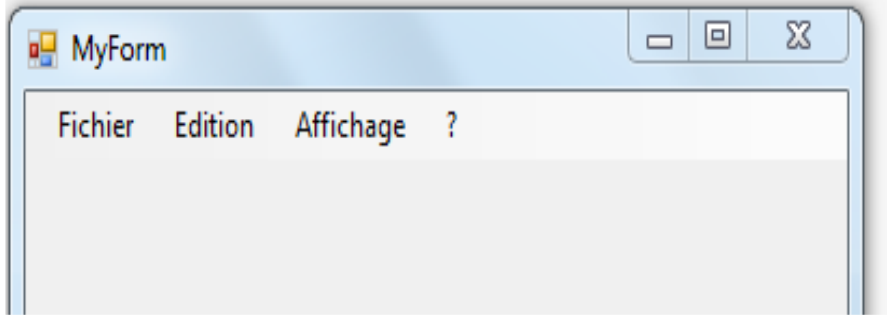
array<TreeNode ^> ^ nodContinents = { nodAfrica, nodAmerica, nodEurope };
TreeNode ^ nodWorld = gcnew TreeNode("World", nodContinents);
treeView1->Nodes->Add(nodWorld);
```



شرح بعض الأدوات الأساسية

أداة تسمح بعمل شريط القوائم للفورم

MenuStrip



TextDirection

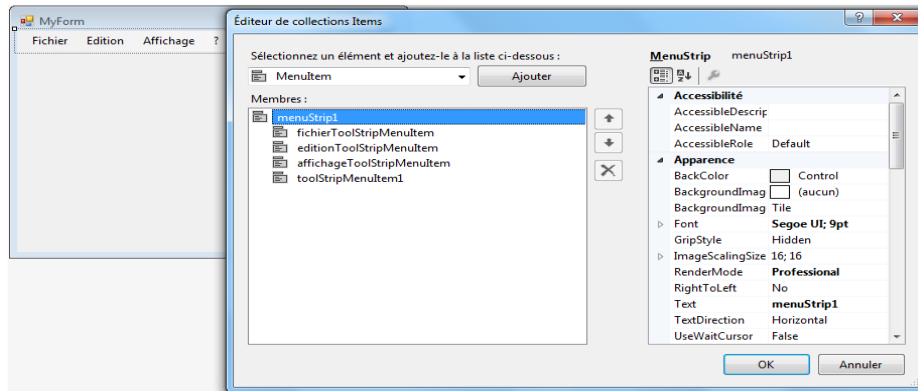
اختيار اتجاه الكتابة

LayoutStyle

اختيار شكل الظهور

items

أهم خاصية بواسطتها يمكن اضافة قوائم جديدة واختيار خصائص كل قائمة على حدى من لون ونوعية الخط والايقونة وصورة الخلفية واتجاه الكتابة وغيرها



items

هذه الصورة تظهر عند اختيار الخاصية

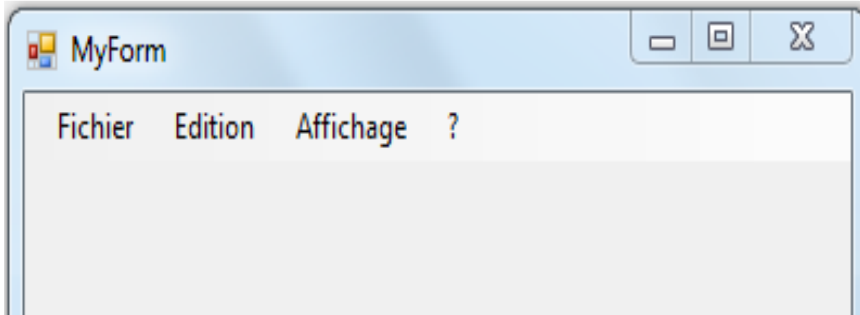
عند تحديد شريط القوائم يظهر أعلى يمين الشريط مثلث صغير بالنقر عليه تظهر قائمة يمكن اختيار بعض الخصائص منها وايضا يمكن اضافة قوائم مألوفة .

شرح بعض الأدوات الأساسية

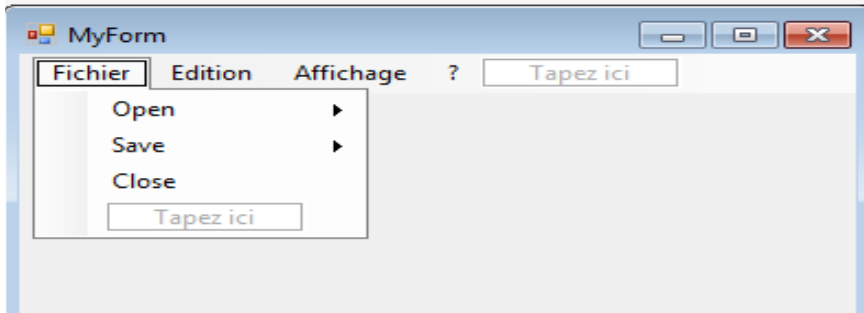
MenuStrip

افتح مشروع جديد وأضف إليه أداة عمل شريط القوائم

مثال 1



أضيف القوائم كما في الصورة المقابلة وذلك بتحديد الشريط وهذا سيتيح لك إضافة قوائم جديدة وأيضا إضافة قوائم فرعية ويمكننا مواصلة تسلسل القوائم الي ما نريد .
اختر القائمة الأولى وأضيف إليها 3 قوائم فرعية أخرى



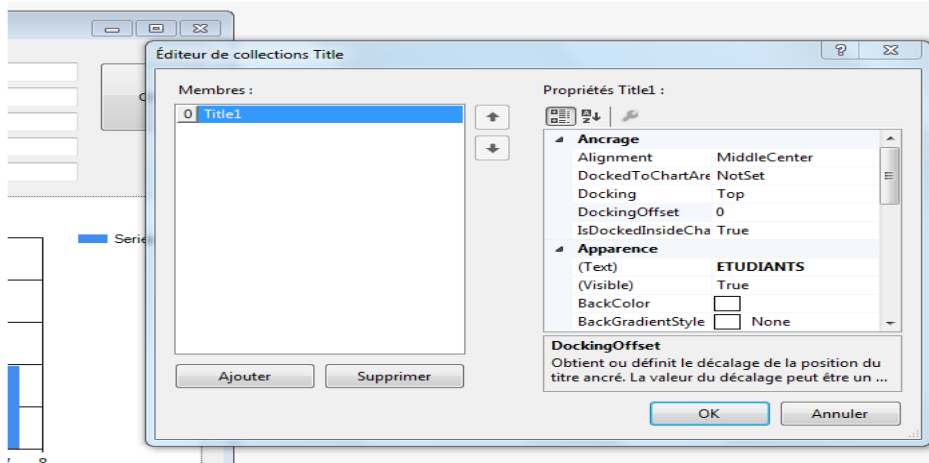
لكتابة الكود المتعلق بالقوائم الفرعية يكفي ان نختار القائمة التي نريد ثم ننقر عليها نقرا مزدوجا بواسطة الفأر حينها نكتب الكود في المكان المخصص لذلك .

اختر القائمة الفرعية Close و اكتب الكود التالي بدخلها وهو كود لغلق البرنامج .

```
Application::Exit();
```

شرح بعض الأدوات الأساسية

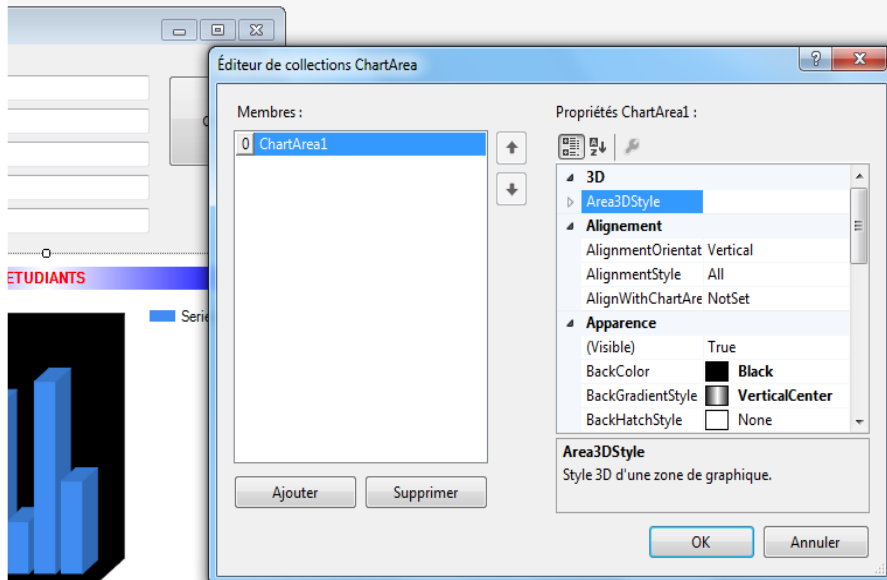
أداة تستعمل لعمل المخططات الاحصائية . **Chart**



Titles

كتابة عنوان للمخطط (Text)

الصورة 1 تظهر عند اختيار هذه الخاصية ستجد حينها الكثير من الخواص المتعلقة بالخط و الالوان وبعض الأشياء التجميلية عليك اكتشافها .



chartAreas

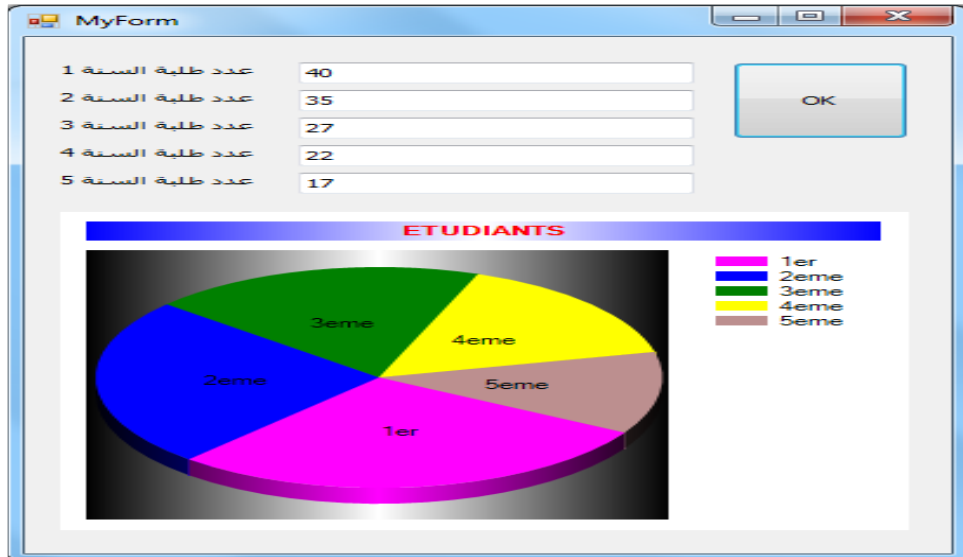
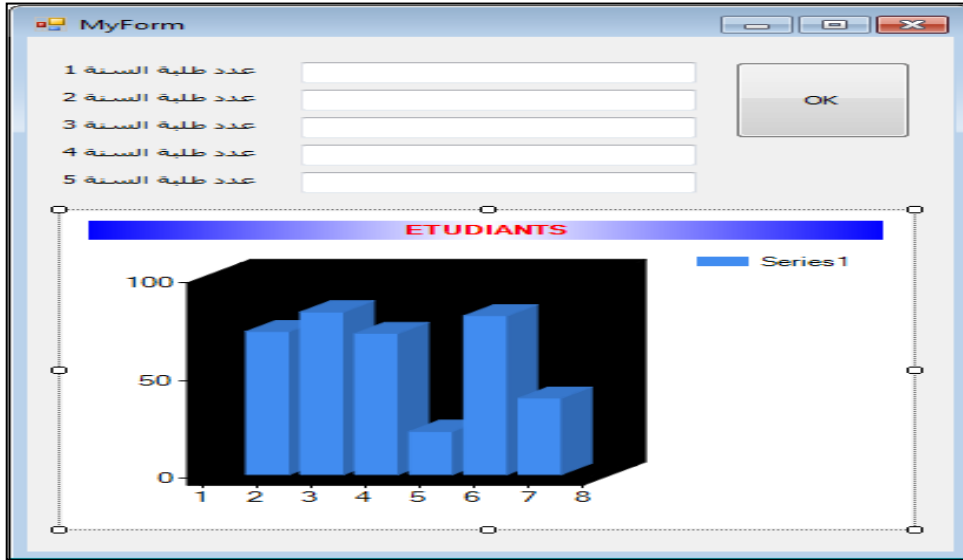
Area3DStyle تجد في هذه الخاصية امكانية تفعل الشكل الثلاثي الأبعاد و يمكن أيضا تغيير زوايا المخطط .

الصورة 2 تظهر عند اختيار هذه الخاصية ستجد حينها الكثير من الخواص المتعلقة بالألوان وبعض الأشياء التجميلية عليك اكتشافها .

شرح بعض الأدوات الأساسية

مثال 1

صمم الفورم المقابل طبعا مع اضافة أداة عمل المخططات الاحصائية وغير الخصائص المطلوبة كما تريد وأقصد العنوان واللون والشكل .



النتيجة بعد ادخال المعطيات والنقر على الزر .

شرح بعض الأدوات الأساسية

```
try{
int etud1 = System::Int32::Parse(textBox1->Text);
int etud2 = System::Int32::Parse(textBox2->Text);
int etud3 = System::Int32::Parse(textBox3->Text);
int etud4 = System::Int32::Parse(textBox4->Text);
int etud5 = System::Int32::Parse(textBox5->Text);

chart1->Series->Clear();
System::Windows::Forms::DataVisualization::Charting::Series^ s=gcnew System::Windows::Forms::
DataVisualization::Charting::Series();
s->Points->AddY(etud1);
s->Points[0]->Label="1er";
s->Points[0]->Color=Color::Magenta;
s->Points->AddY(etud2);
s->Points[1]->Label="2eme";
s->Points[1]->Color=Color::Blue;
s->Points->AddY(etud3);
s->Points[2]->Label="3eme";
s->Points[2]->Color=Color::Green;
s->Points->AddY(etud4);
s->Points[3]->Label="4eme";
s->Points[3]->Color=Color::Yellow;
s->Points->AddY(etud5);
s->Points[4]->Label="5eme";
s->Points[4]->Color=Color::RosyBrown;
s->ChartType=SeriesChartType::Pie;
chart1->Series->Add(s);
}
catch(...){
}
```

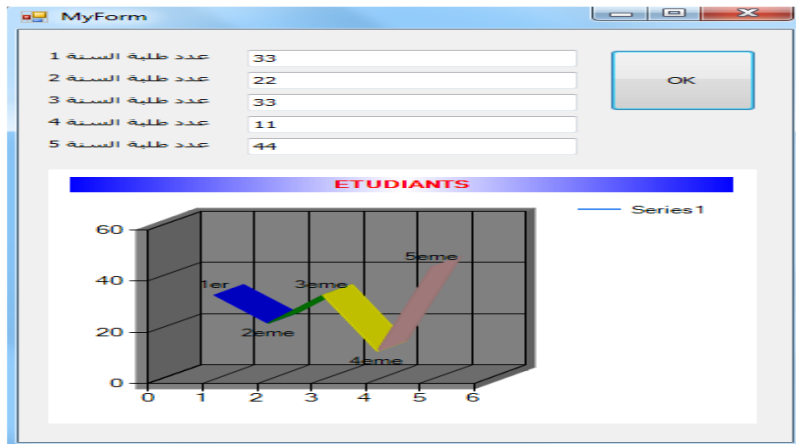
شرح بعض الأدوات الأساسية

```
s->ChartType=SeriesChartType::Pie;
```

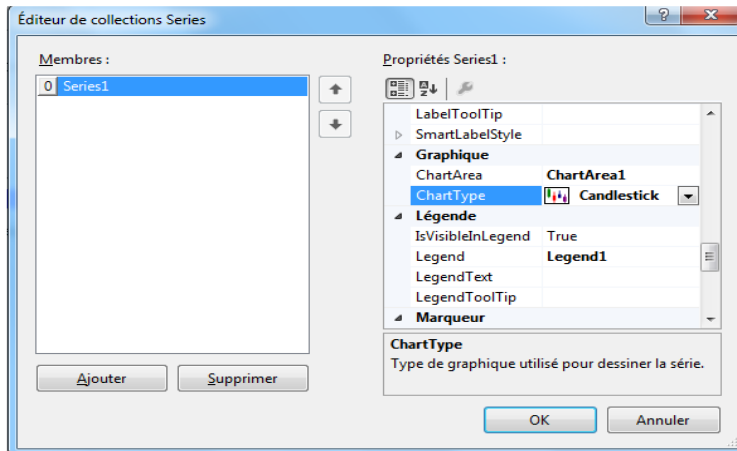
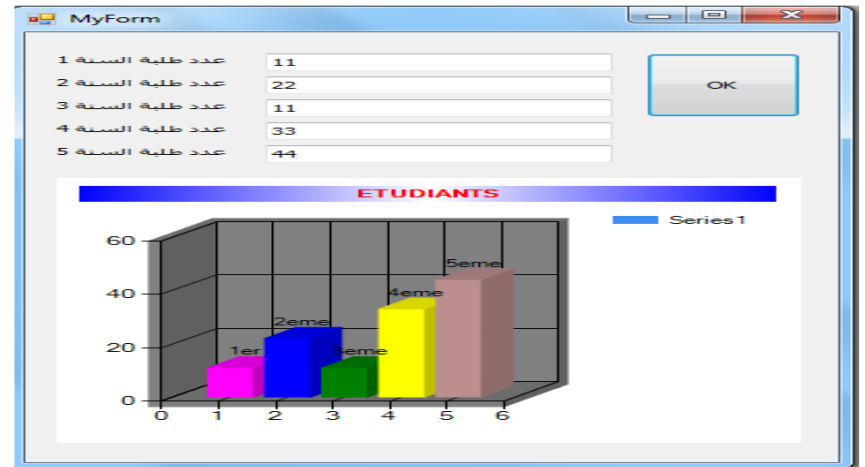
هذه التعليمة تمكننا من اختيار نوع المخطط هل هو على شكل دائرة أو أعمدة .

في الكود السابق غير هذه التعليمة الي :

```
s->ChartType=SeriesChartType::Line;
```



```
s->ChartType=SeriesChartType::Column;
```



اذا اردت المزيد من الأشكال اختر أداة المخططات ثم من

شريط الخصائص اختر : **Series -> ChartType**

تجيد الكثير من المخططات اختر اسم المخطط الذي تريد ثم

انقله الي التعليمة السابقة

```
s->ChartType=SeriesChartType:: ;
```


شرح بعض الأدوات الأساسية

أداة تسمح لنا بالتعامل مع التاريخ والوقت.

DateTimePicker

Format

اختيار ظهور التاريخ أو الوقت والصيغة أيضا .

ShowUpDown

اختيار التاريخ بشكل يشبه الاختيار اليدوي .

MaxDate

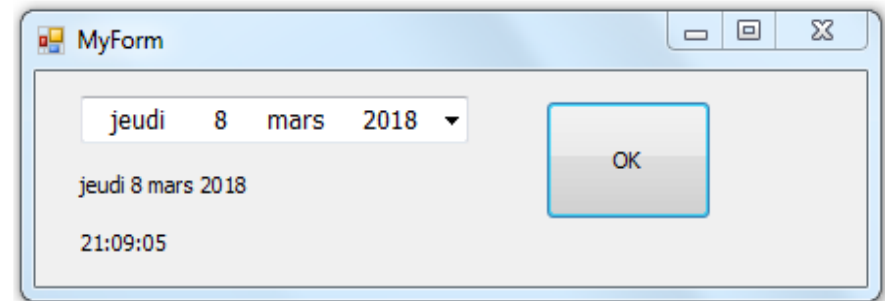
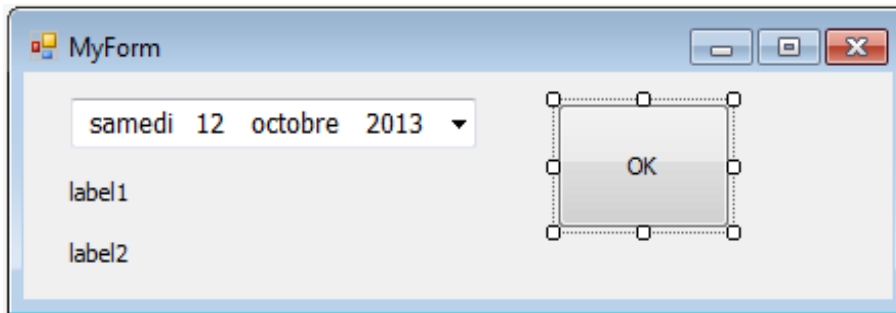
تحديد أكبر تاريخ ممكن الوصول اليه .

MinDate

تحديد أصغر تاريخ ممكن .

Value

التاريخ الذي تشير اليه الأداة .



```
// label1->Text=dateTimePicker1->Value+"";  
DateTime ^date = gnew DateTime();  
date = dateTimePicker1->Value;  
label1->Text = date->ToLongDateString();  
label2->Text = date->ToLongTimeString();
```

شرح بعض الأدوات الأساسية

أداة تسمح لنا بالتعامل مع التاريخ وتستعمل عادة للاظهار فقط . **MonthCalendar**

بعض الخصائص التي تميزها عن الأداة السابقة .

FirstDayOfWeek

اختيار اليوم الذي يكون كبداية للأسبوع .

MaxSelectionCount

اختيار عدد الأيام التي يمكن اختيارها دفعة واحدة .

ShowToday

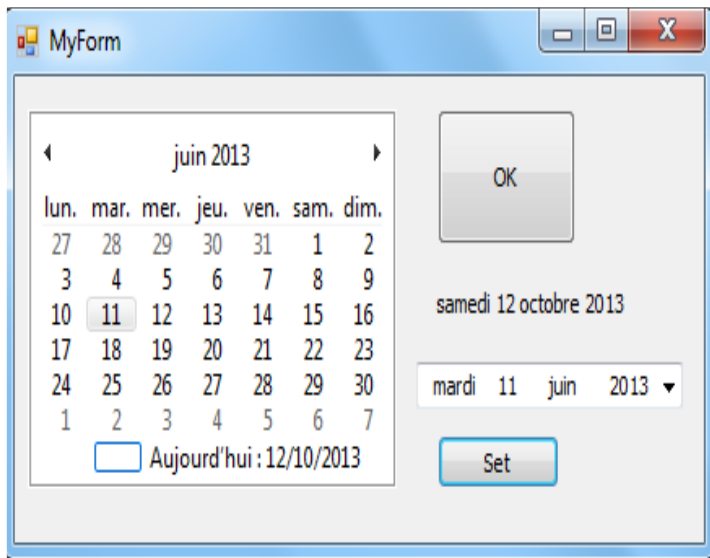
اظهار تاريخ اليوم .

ShowTodayCircle

تلوين وتميز تاريخ اليوم .

ShowWeekNumbers

اظهار أرقام الأسابيع .



OK

```
DateTime ^date = gcnw DateTime();  
date = monthCalendar1->TodayDate;  
label1->Text = date->ToLongDateString();
```

Set

```
monthCalendar1->SetDate(dateTimePicker1->Value);
```

شرح بعض الأدوات الأساسية

الأدوات الغير مرئية

هي التي نضعها على الفورم ولا نشاهدها أثناء التنفيذ ولكن نشاهد فعلها .

Timer أداة تقوم بإنجاز عمل محدد كل فترة زمنية محددة , هذه الفترة يحددها المبرمج .

بواسطة هذه الخاصية يمكن تشغيل وتوقيف الأداة .

Enabled المدة الزمنية التي تنتظرها الأداة حتى تعيد تنفيذ العمل المطلوب تحدد هذه المدة بالميلي ثانية .

Interval الحدث الذي نكتب الكود بداخله كما يمكن الوصول اليه عن طريق النقر المزدوج على الاداة .

Tick



مثال 1 افتح مشروع جديد .

Timer أضيف الاداة

شكل الفورم بنفس الصورة التي تشاهدها .

شرح بعض الأدوات الأساسية

Button1(Marche)

```
timer1->Enabled = true;
```

Button2(Arret)

```
timer1->Enabled = false;
```

الكود الخاص بالأزرار :

Enabled

True

Interval

1000 تنفيذ العمل كل 1 ثانية .

Timer : خصائص الأداة :

من خلال الحدث **Tick** أو بالنقر المزدوج على الاداة **Timer** نكتب الكود التالي :



```
DateTime ^d = gnew DateTime();  
d = DateTime::Now;  
  
label2->Text = d->ToLongTimeString();  
label4->Text = d->ToShortTimeString();  
  
label6->Text = d->ToLongDateString();  
label8->Text = d->ToShortDateString();
```

Now تعني التاريخ والوقت في هذه اللحظة .

شرح بعض الأدوات الأساسية

أداة تستعمل لفتح علبة حوار , حتى تتمكن من فتح ملفات أو صور أو غير ذلك .

OpenFileDialog

أداة تستعمل لفتح علبة حوار لحفظ الملفات أو الصور التي قام البرنامج بمعالجتها

SaveFileDialog

أهم الخصائص للأداتين وهي مشتركة بينهما في الجدول التالي :

String^ FileName

خاصية تحمل رابط الملف أو الصورة أو أي شيء يراد فتحه أو حفظه .

String^ Filter

خاصية لتحديد أنواع الملفات التي تظهر عند فتح علبة الحوار .

int FilterIndex

اختيار رقم الملف الذي نقترحه للظهور عند فتح علبة الحوار .

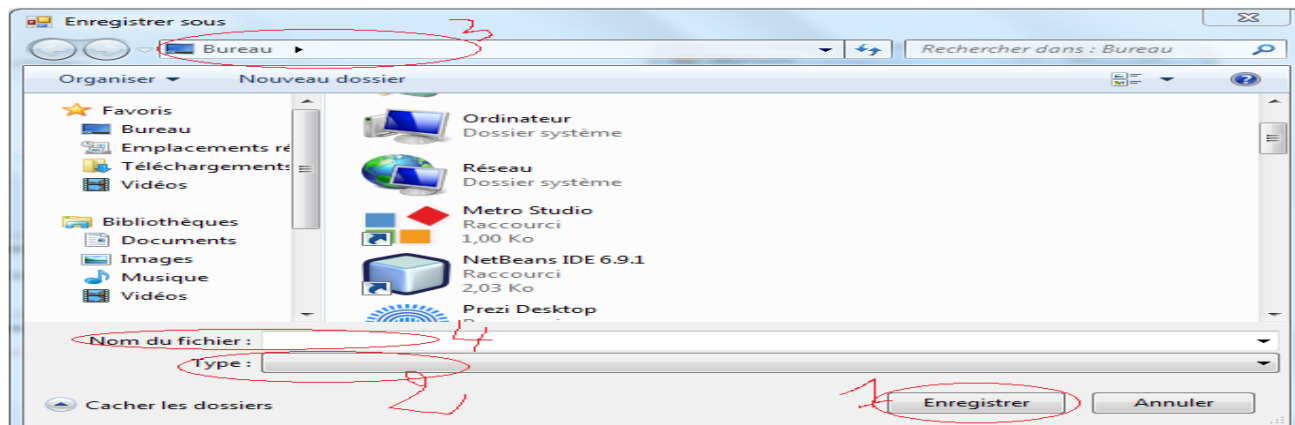
String^ InitialDirectory

الملف الذي نقترحه علي المستخدم ليتم فيه الحفظ .

DialogResult.ShowDialog()

دالة لفتح علبة الحوار وترجع قيمة من النوع

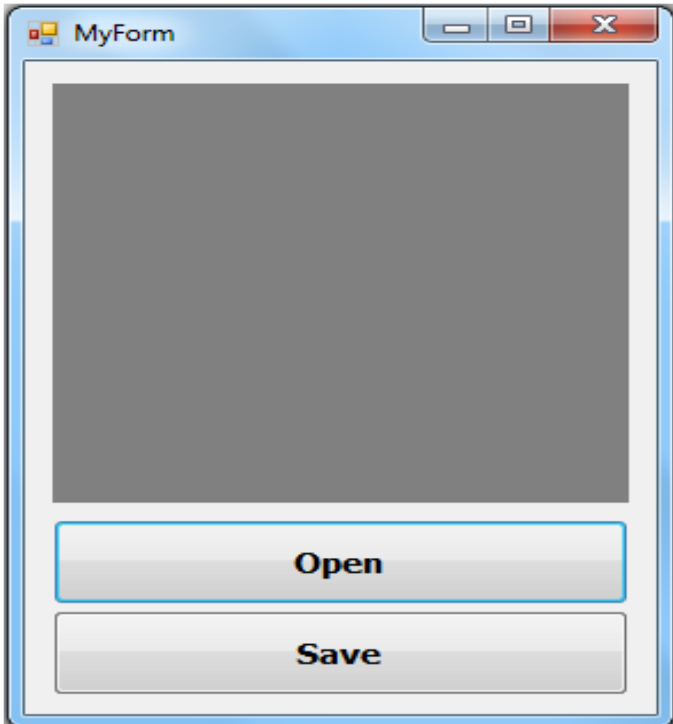
DialogResult



شرح بعض الأدوات الأساسية

افتح مشروع جديد وأضف أداتي الحوار السابقتين أداة الفتح و أداة الحفظ الى الفورم , ثم أضف الأدوات كما في الصورة .

مثال 1



PictureBox

الأداة الرمادية هي

غير الخصائص للأدوات كما في الجدول :

Button1(Open)

Text

Open

Button2(Save)

Text

Save

PictureBox1

SizeMode

StretchImage

هذا البرنامج يقوم بفتح صورة من اختيار المستخدم ثم حفظها في مكان وتحت اسم من اختيار المستخدم .

شرح بعض الأدوات الأساسية

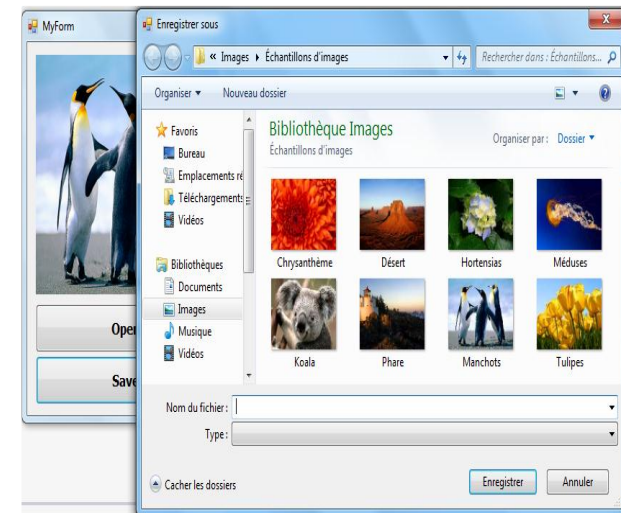
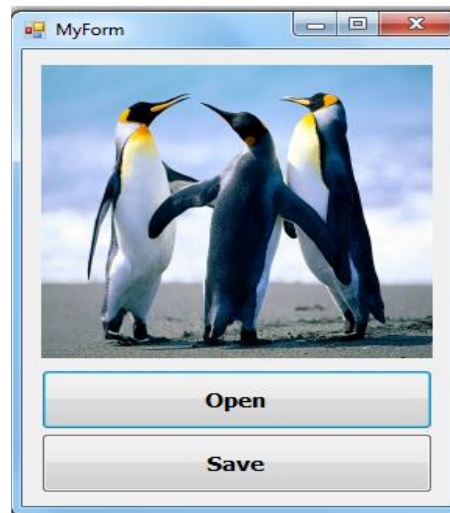
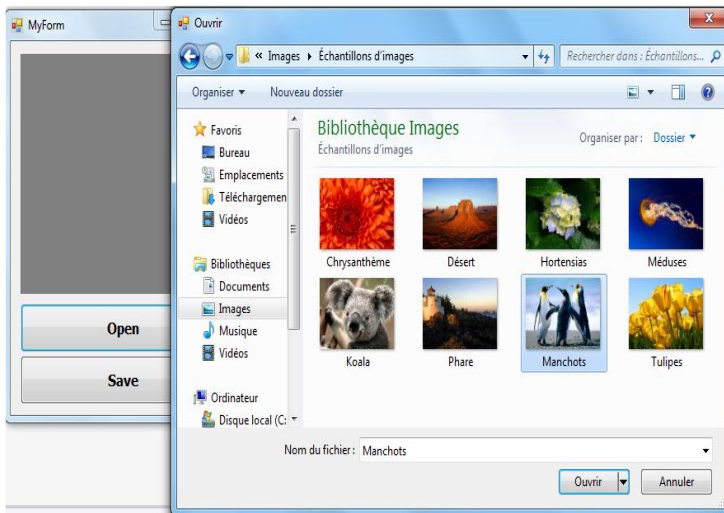
اكتب كود الزرين كما في الجدول التالي :

Button1(Open)

```
if (openFileDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK)
{
    pictureBox1->Image = System::Drawing::Image::FromFile(openFileDialog1->FileName,true);
}
```

Button2(Save)

```
if (saveFileDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK)
{
    pictureBox1->Image->Save(saveFileDialog1->FileName+".JPEG");
}
```



شرح بعض الأدوات الأساسية

أداة تستعمل لفتح علبة حوار , حتى تتمكن من اختيار نوع الخط الذي نريد.

fontDialog

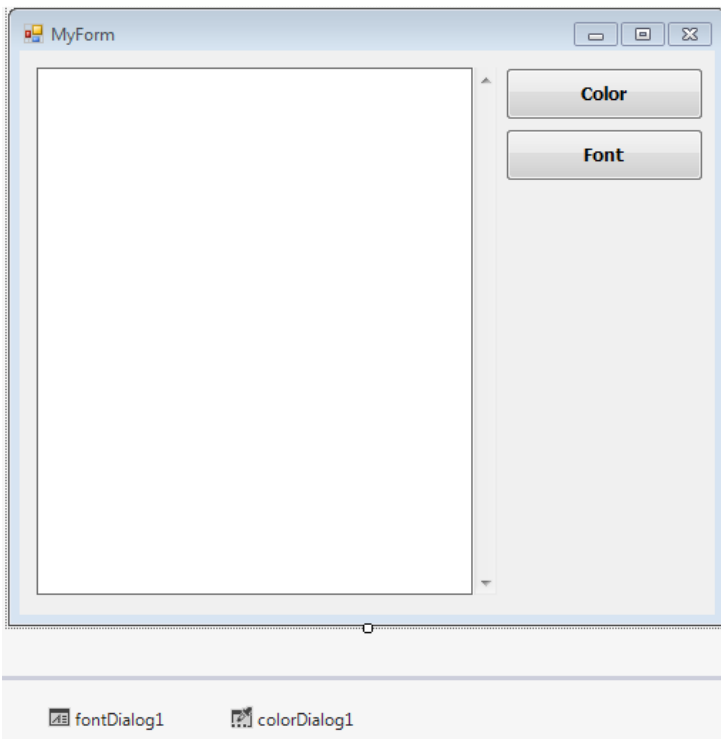
أداة تستعمل لفتح علبة حوار لاختيار اللون الذي نريد .

colorDialog

الخصائص مثل أداتي الحوار السابقتين .

افتح مشروع جديد وأضيف أداتي الحوار السابقتين أداة اختيار الخط و أداة اختيار اللون الى الفورم , ثم أضيف الأدوات كما في الصورة .

مثال 1



Button1(Color)

Text	Color
------	-------

Button2(Font)

Text	Font
------	------

TextBox1

BorderStyle	FixedSingle
-------------	-------------

ScrollBars	Vertical
------------	----------

Multiline	True
-----------	------

شرح بعض الأدوات الأساسية

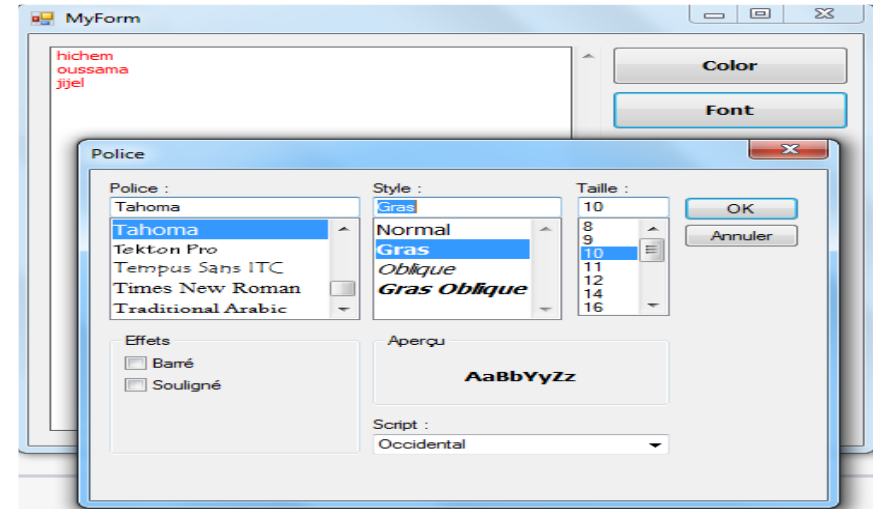
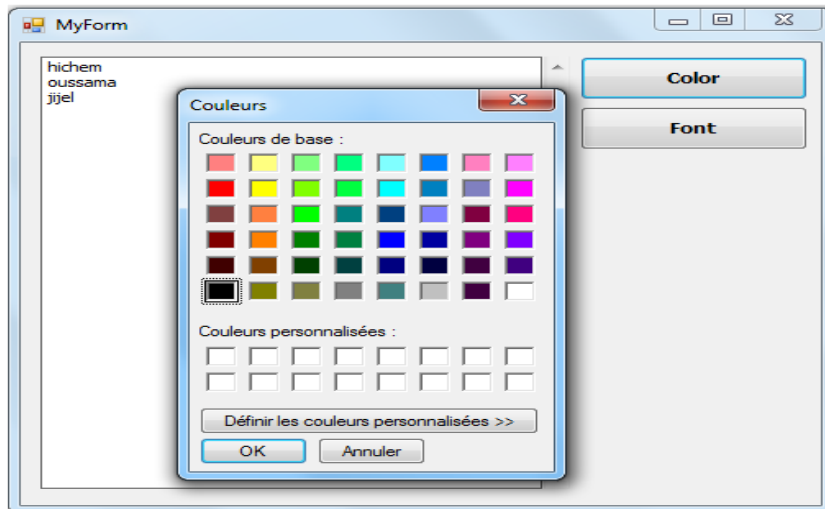
اكتب كود الزرين كما في الجدول التالي :

Button1(Color)

```
if (colorDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK)
{
    textBox1->ForeColor = colorDialog1->Color;
}
```

Button2(Font)

```
if (fontDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK)
{
    textBox1->Font = fontDialog1->Font;
}
```



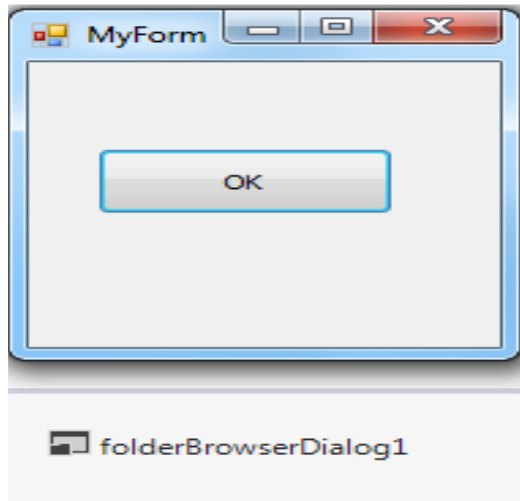
شرح بعض الأدوات الأساسية

FolderBrowserDialog

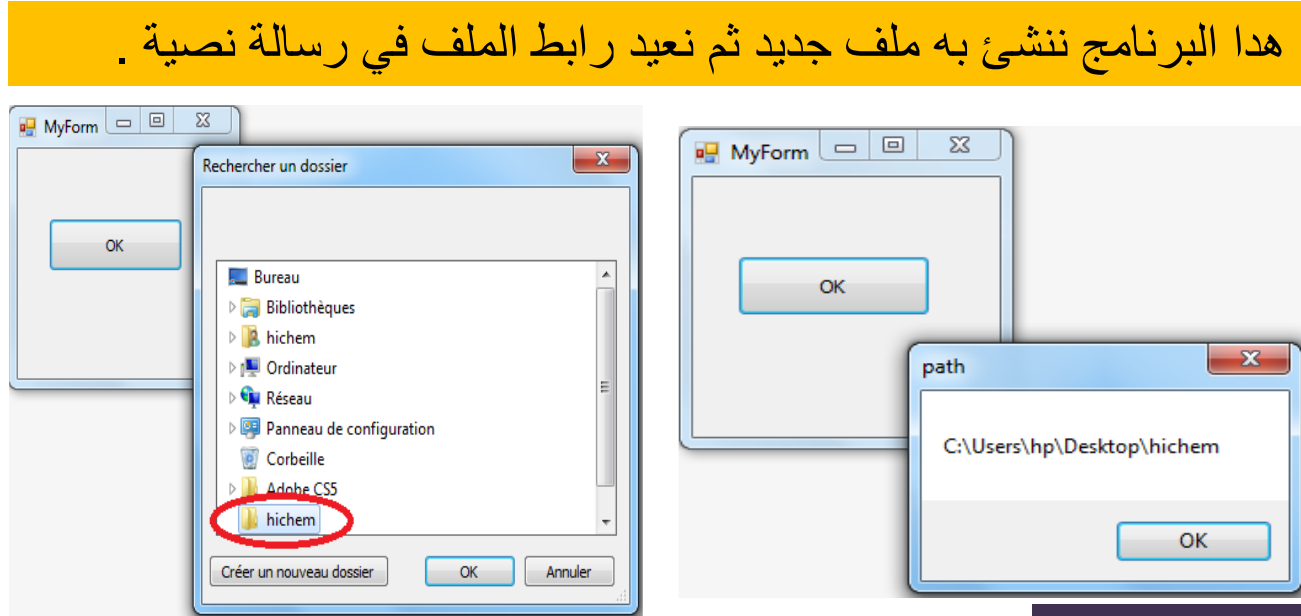
أداة تستعمل لفتح علبة حوار , حتى تتمكن من اختيار ملف او انشاء ملف جديد.

مثال 1

افتح مشروع جديد وأضيف أداة الحوار السابقة الى الفورم , ثم أضيف زر كما في الصورة 1.



صورة 1



كود الزر :

```
if (folderBrowserDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK)
{
    String^ path = folderBrowserDialog1->SelectedPath;
    System::Windows::Forms::MessageBox::Show(path, "path");
}
```

شرح بعض الأدوات الأساسية

ToolTip

أداة تستعمل لإظهار ملاحظة بسيطة للمستخدم عن وظيفة أداة معينة وتظهر الملاحظة في مستطيل أصفر بمجرد مرور الفأرة على الأداة .

بعد اضافة الأداة الي الفورم تظهر لكل أداة علي الفورم خاصية جديدة نكتب فيها الملاحظة التي نريد وهي

ToolTip sur toolTip1

AutoPopDelay

تحديد المدة التي تبقى فيها الملاحظة ظاهرة على الأداة (بالميلي ثانية) .

InitialDelay

تحديد المدة التي تمكثها الفأرة على الاداة حتي تظهر الملاحظة (بالميلي ثانية) .

isBalloon

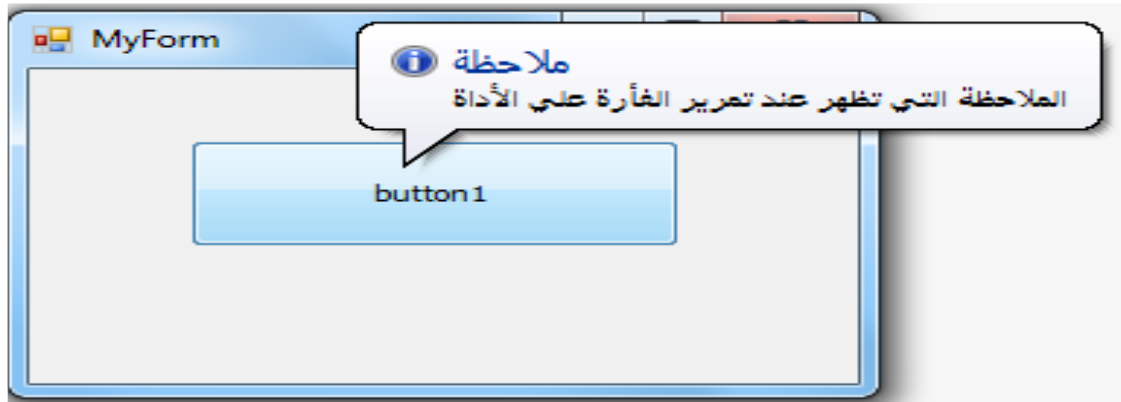
لظهور الملاحظة على شكل فقاعة .

ToolTipIcon

اختيار أيقونة للملاحظة التي تظهر .

ToolTipTitl

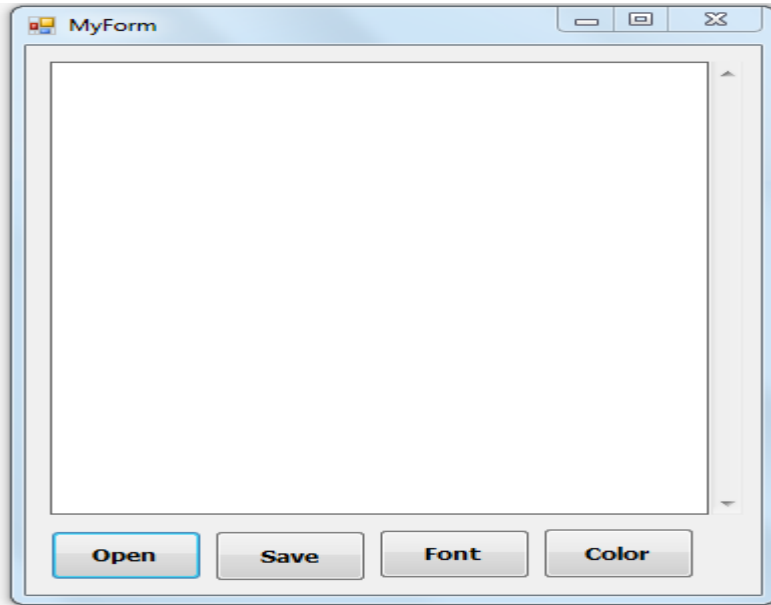
كتابة عنوان للملاحظة التي تظهر .



أمثلة برمجية

مثال 1 نتطرق في هذا المثال الي طريقة التعامل مع الملفات النصية .

افتح مشروع جديد وشكل الفورم المقابل علي الصورة .



أضيف الأدوات التالية أيضا :

OpenFileDialog

SaveFileDialog

fontDialog

colorDialog

Button1(Open)

Text Open

Button2(Save)

Text Save

Button3(Font)

Text Font

Button4(Color)

Text Color

TextBox1

BorderStyle FixedSingle

ScrollBars Vertical

Multiline True

Button1(Open)

```
if (openFileDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK)
{
    StreamReader^ sr = File::OpenText(openFileDialog1->FileName);
    try
    {
        String^ s = "";
        while ( s = sr->ReadLine() )
        {
            textBox1->Text=textBox1->Text+s+Environment::NewLine;
        }
    }
    catch(...){
    }
    finally
    {
        if ( sr )
            delete (IDisposable^)(sr);
    }
}
```

Button2(Save)

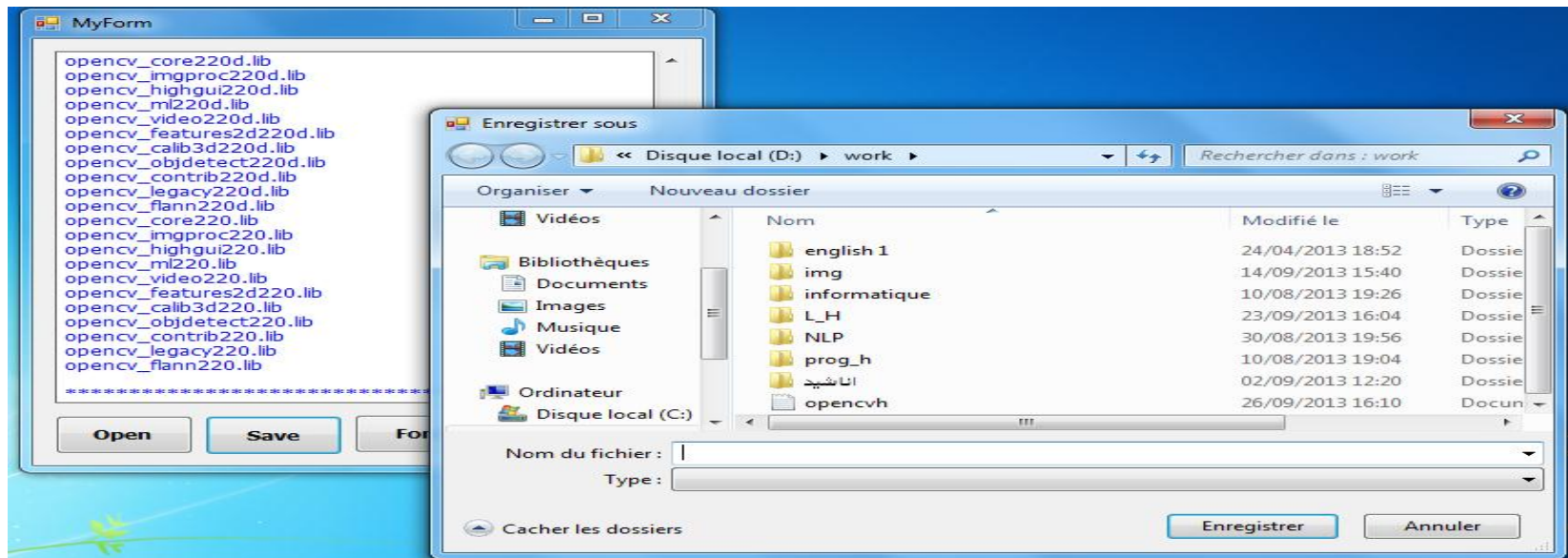
```
if (saveFileDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK)
{
String^ path =saveFileDialog1->FileName+".txt";
if ( !File::Exists( path ) )
{
StreamWriter^ sw = File::CreateText( path );
try
{
sw->WriteLine( textBox1->Text );
}
catch(...){
}
finally
{
if ( sw )
delete (IDisposable^)sw;
}
}
}
```

Button3(Font)

```
if (fontDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK)
{
    textBox1->Font = fontDialog1->Font;
}
```

Button4(Color)

```
if (colorDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK)
{
    textBox1->ForeColor = colorDialog1->Color;
}
```



أمثلة برمجية

مثال 2 نعد المثال 1 بطريقة البرمجة الكائنية التوجه .

```
#include "Myfile.h"
using namespace System::IO;
void Myfile::file_write(String^ path,String^ cont)
{
if ( !File::Exists( path ) )
{
StreamWriter^ sw = File::CreateText( path );
try
{
sw->WriteLine( cont );
}
catch(...){
}
finally
{
if ( sw )
delete (IDisposable^)sw;
} }
void Myfile::file_read(String^ path,System::Windows::Forms::TextBox^ txtbox)
{
StreamReader^ sr = File::OpenText(path);
try
{
String^ s = "";
while ( s = sr->ReadLine() )
{
txtbox->Text=txtbox->Text+s+Environment::NewLine;
}
}
catch(...){
}
finally
{
if ( sr )
delete (IDisposable^)(sr);
}
}
```

أضيف الى المشروع كلاس .

```
#pragma once
```

```
using namespace System;
```

```
ref class Myfile
```

```
{
```

```
public:
```

```
void file_write(String^,String^);
```

```
void file_read (String^,
```

```
System::Windows::Forms::TextBox^);
```

```
};
```

.h

اسم الكلاس :

Myfile

.cpp

أمثلة برمجية

```
1 #pragma once
2 #include "Myfile.h"
3 namespace Projet1 {
4
5     using namespace System;
6     using namespace System::ComponentModel;
7     using namespace System::Collections;
8     using namespace System::Windows::Forms;
9     using namespace System::Data;
10    using namespace System::Drawing;
11    using namespace System::IO;
12    /// <summary>
13    /// Description résumée de MyForm
14    /// </summary>
15    public ref class MyForm : public System::Windows::Forms::Form
16    {
17        Myfile file;
18    public:
19        MyForm(void) { ... }
20
21    protected:
22        ///
23    };
24 }
```

لا تنسى السطر 2
والسطر 17 .

زر اختيار الخط و زر اختيار اللون نبقى نفس الكود .

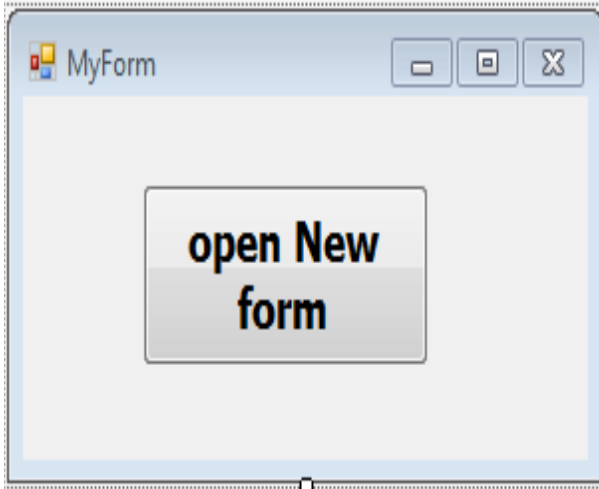
Button1(Open)

```
if (openFileDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK)
{
    file.file_read(openFileDialog1->FileName, textBox1);
}
```

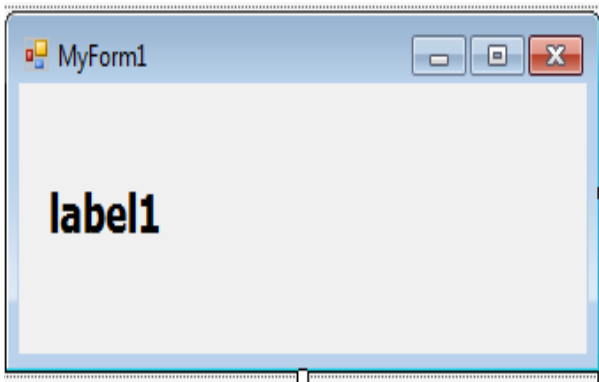
Button2(Save)

```
if (saveFileDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK)
{
    String^ path =saveFileDialog1->FileName+".txt";
    file.file_write(path, textBox1->Text);
}
```

مثال 3 نتعلم طريقة استدعاء فورم جديد.



صورة 1



صورة 2

افتح مشروع جديد وصمم الفورم المقابل .

أضف الي المشروع فورم جديد بنفس الطريقة , صمم الفورم مثل صورة 2

عد الي الفورم الأول وقم بتضمين الفورم الثاني (صورة 3) (سطر 2).

في الفورم الأول أكتب الكود التالي في حدث النقر على الزر :

```
MyForm1^ form =gcnew MyForm1();  
form->text = DateTime::Now.ToString();  
form->ShowDialog();
```

```
Proj1::MyForm  
1 #pragma once  
2 #include "MyForm1.h"  
3 namespace Proj1 {  
4  
5     using namespace System  
6     using namespace System  
7     using namespace System  
8     using namespace System  
9     using namespace System
```

صورة 3

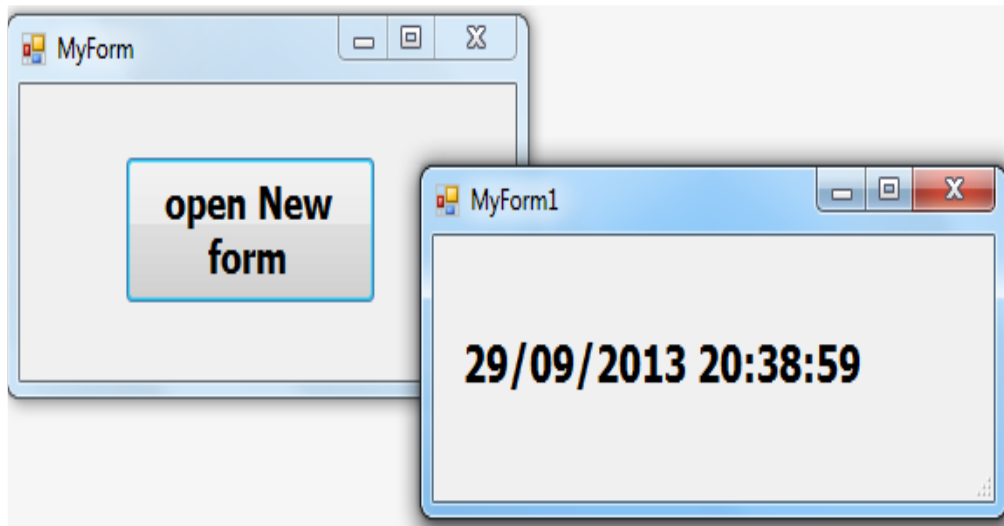
أمثلة برمجية

```
15 public ref class MyForm1 : public Form
16 {
17     public:
18         String^ text;
19     public:
20         MyForm1(void)
21         {
22             InitializeComponent();
23             //
24             //TODO: ajoutez ici le code
```

عد الآن الى الفورم الثاني وقم بالتصريح بمتغير من نوع سلسلة حرفية كما في الصورة (سطر 18).

في الفورم الثاني ايضا قم بنقر مزدوج عليه و اكتب الكود التالي :

```
label1->Text = text;
```



النتيجة :

المراجع

..... أخيرا

أتمنى أن يقدم هذا العمل الاضافة الايجابية لكل من يقرأه
كما أرجو أن يكون محفزا قويا لتعلم الواجهات الرسومية في سي بلس بلس
وأطلب من القراء الكرام أن لا ينسوننا من صالح دعائهم

المراجع :

Beginning

Ivor Horton's

Ivor Horton

Visual C++ 2010

APPRENTISSAGE DU LANGAGE C# 2008

et du Framework .NET 3.5

Hichem Felouat

hichemfel@gmail.com