

تطبيقات برمجية (١) الملزمة الأولى

النموذج (Form): هو عبارة عن مجموعة من المعلومات أو الخيارات مرتبطة مع بعضها لتتمد المستخدم بالمعلومات التي يريدها. وأي Form تعتبر class .

Controls : مجموعة من الأدوات الجاهزة تجعل المستخدم يتعامل بسهولة مع المعلومات والخيارات. هناك عدة أنواع من هذه الأدوات فمنها ما يسمح للمستخدم بعرض المعلومات مثل Labels ومنها ما يسمح للمستخدم بعرض المعلومات أو إدخال معلومات مثل textbox و combo box وهناك ما يمكن المستخدم من تنفيذ مجموعة من الأوامر مثل Button.

قواعد التصميم الصحيح للنماذج

١. البساطة و السهولة : يجب أن يكون التصميم سهلا ليتمكن المستخدم من إتمام مهمته في أسرع وقت ممكن ، فمثلا الأدوات المرتبطة ببعضها يجب أن تجمع في مكان واحد ، لذلك استخدام الأداة Tab Order تسهل على المستخدم التعامل مع البرنامج ، أيضا استخدام القيم الافتراضية (Default values) يسهل على المستخدم إدخال البيانات فمثلا إذا كان ٩٠% من المستخدمين يختارون غزة على أنها مدينتهم يمكن جعلها قيمة افتراضية ، عندها لن يحتاج ٩٠% من المستخدمين إدخال اسم المدينة.
٢. أماكن الأدوات : مكان وضع الأداة (Control) يجب أن يعكس مدى أهمية استخدامها ، فمثلا إذا صممت نموذج يدخل بيانات المستخدم وكانت هذه البيانات منها إجباري و منها اختياري يجب أن يكون الإجباري في مكان أعلى من الاختياري ، أيضا لا يجب وضع أدوات على النموذج بشكل كبير يجعل من الصعب على المستخدم التعامل مع النموذج.
٣. التناسق : يفضل أن يكون التصميم لا يثير غضب المستخدم فمثلا لا يعقل أن تطلب من المستخدم إدخال اسمه ثم تنقله إلى نموذج آخر مختلف و تطلب منه نفس الطلب. أيضا التناسق يتم عن طريق اختيار الحجم و اللون و نوع الخط ونوع الأداة.
٤. الأدب : لا بد أ، يكون تصميمك مؤدبا و ذلك مثلا باستخدام كلمات جيدة مثل أهلا وسهلا ، مرحبا بكم ، نأسف لعدم السماح لك بالدخول و هكذا....

كيفية إضافة نموذج جديد إلى مشروع (project)

١. اختر القائمة Project menu ثم انقر على Add Windows Form سيفتح مربع حوار لإنشاء نموذج جديد.
٢. انقر على Windows Form ثم انقر على Open ستجد أن نموذج جديد تم إضافته.

كيفية إضافة نموذج برمجيا أثناء عمل البرنامج :- على فرض أننا أنشأنا نموذج جديد كما وضعنا قبل قليل اسمه DialogForm نكتب الكود التالي.

```
// This example assumes that you have already designed a form
// called DialogForm
DialogForm myForm;
myForm = new DialogForm();
```

خصائص النموذج (form properties)

أمثلة لبغض الخصائص و استخداماتها.

اسم الخاصية	الاستخدام	برمجيا
StartPosition	تستخدم لتحديد الموضع الذي سيظهر فيه النموذج عند التنفيذ و هناك عدة خيارات	
BackColor	لتغيير لون الخلفية للنموذج	<code>MyForm.BackColor = System.Drawing.Color.Red;</code>
ForeColor	لتغيير لون الخط للأدوات التي ستضاف إلى النموذج	
Text	يستخدم لتغيير عنوان النموذج	<code>MyForm.Text = "Myform";</code>
Cursor	تستخدم لتغيير شكل الفأرة عندما تكون موجودة داخل النموذج و هناك عدة خيارات	
BackGroundImage	تستخدم لجعل خلفية النموذج صورة معينة بدلا من أن تكون الخلفية لون	
Opacity	تستخدم لتغيير الشفافية للنموذج عن طريق وضع قيمة من ٠ إلى ١	<code>MyForm.Opacity = .5;</code>

استخدام الدوال الخاصة بالنموذج (form methods)

تستخدم الدوال التالية للتحكم في طريقة ونمط عرض النموذج للمستخدم وهي تنتمي جميعا للصف `System.Windows.Forms.Form`. لكي نستخدم هذه الدوال يجب أولا أن نمتلك كائن من نوع `form` أو استخدام الكلمة `this` إذا أردنا تنفيذ هذه الدوال على النموذج الحالي. الجدول التالي يوضح بعض الأمثلة على هذه الدوال و الغرض منها.

Form.Show	تقوم بعرض النموذج مع إمكانية الانتقال إلى نموذج آخر قبل الانتهاء من العمل مع هذا النموذج.
Form.ShowDialog	تقوم بعرض النموذج مع عدم إمكانية الانتقال إلى نموذج آخر إلا بعد الانتهاء من العمل مع هذا النموذج.
Form.Activate	تستخدم عندما يكون عدة نماذج معروضة على الشاشة و تريد أن تستخدم أحد هذه النماذج أي ليصبح أمام كل النماذج.
Form.Hide	تقوم هذه الدالة بإخفاء النموذج لكن مع بقاء النموذج في الذاكرة ، وعند استدعاء الدالة <code>Show</code> يظهر النموذج مرة أخرى.
Form.Close	تقوم هذه الدالة بإغلاق النموذج مع عدم بقاء النموذج في الذاكرة ، وعند استدعاء الدالة <code>Show</code> لا يمكن أن يظهر النموذج مرة أخرى.

أمثلة على استخدام الدوال:-

```
// This example assumes that you have created a Form class called
// DialogForm
DialogForm myForm = new DialogForm();
// Shows the form regularly
myForm.Show();
// Shows the form modally
myForm.ShowDialog();
myForm.Hide();
myForm.Show();
myForm.Close();
```

Form Lifetime Events

هناك العديد من events التي تظهر أو تنفذ أثناء دورة حياة النموذج (form) و التي يمكن للمبرمج استخدامها لتنفيذ بعض المهام ، سنذكر بعض هذه events ومتى تنفذ في الجدول التالي:

ينفذ عندما نقوم بعرض النموذج لأول مرة على الشاشة.	Load
ينفذ عندما يكون عدة نماذج معروضة على الشاشة وأصبح النموذج في الأمام (Activated) أو في الخلف (Deactivate)	Activated/Deactivate
ينفذ عندما تتغير خاصية (Visible) أي يظهر أو يختفي النموذج عن طريق الدالة Hide والدالة Show	VisibleChanged
ينفذ قبل إغلاق النموذج مباشرة	Closing
ينفذ بعد إغلاق النموذج مباشرة	Closed

أمثلة على استخدام events:-

```
Form myForm = new Form();
myForm.Show(); // The Load event fires here
myForm.Hide(); // Form is now invisible
myForm.Show(); // The Load event doesn't fire again
myForm.Close(); // Closes and disposes the form
myForm.Show(); // Throws an exception because myForm
// is no longer available
```

Buttons Handler

سنتعلم في هذه المحاضرة كيف نتعامل مع الأداة Button وهي أكثر الأدوات استخداما حيث لا يخلوا برنامج سواء كان صغيرا أو كبيرا من استخدامها.

هناك عدة أنواع من الدوال التي تتعامل مع التغيير الذي يحدث على الزر (Button) سنتعرف على أكثرها استخداما من خلال الأمثلة ولكن سنذكرها أولا :

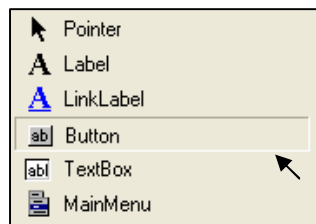
- ✓ النوع الأول يستدعى عند الضغط على الزر (Button).
- ✓ النوع الثاني يستدعى عندما تقع الفأرة على الزر (Button).
- ✓ النوع الثالث يستدعى عندما تخرج الفأرة عن الزر (Button).

ولكن النوع الأول هو أكثرها استخداما . سنوضح عدة أمثلة عن النوع الأول . ثم سنوضح مثالا يشملها كلها.

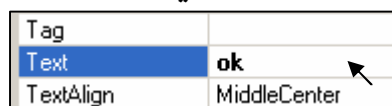
مثال ١ : سنقوم في هذا المثال بكتابة برنامجا يقوم بإظهار رسالة ترحيب عند الضغط على الزر (Button) .

الخطوات :

- ١- افتح مشروعا جديدا وملف من نوع Windows Application
- ٢- اسحب من قائمة Toolbox أداة Button ثم ضعها على Form



٣- غير خاصية text الموجودة في خصائص button إلى ok



٤- انقر نقرا مزدوجا على الزر (Button) ستلاحظ فتح دالة .

٥- أكتب داخل الدالة الكود التالي:

```
MessageBox.Show("Hello");
```

٦- نفذ البرنامج .

الملاحظات :

ستلاحظ أنك عند الضغط على الزر ستخرج رسالة مكتوب عليها Hello وذلك لان السطر الذي كتبناه سيعرض صندوق رسالة و سيكتب فيه الرسالة التي أدخلناها له في الدالة show .



قبل الضغط على ok



بعد الضغط على ok

مثال ٢ : سنقوم في هذا المثال بكتابة برنامجا يقوم بتغيير لون form عند الضغط على الزر (Button). ومع كل ضغطة يعطي لون جديد.

الخطوات :

١- افتح مشروعا جديدا وملف من نوع Windows Application

٢- اسحب من قائمة Toolbox أداة Button ثم ضعها على Form



٣- غير خاصية text الموجودة في خصائص button إلى Color

٤- انقر نقرا مزدوجا على الزر (Button) ستلاحظ فتح دالة يمكنك تغيير اسم هذه الدالة إلى أي اسم معبر مثل (Chang_Color)

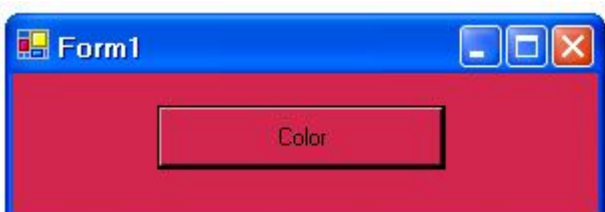
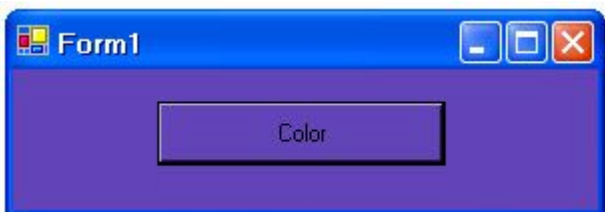
٥- أكتب دالة الدالة الكود التالي :

```
private void Chang_Color(object sender, System.EventArgs e)
{
    Random r=new Random();
    int i=r.Next(255);
    int j=r.Next(255);
    int k=r.Next(255);
    Color c=Color.FromArgb(i,j,k);
    this.BackColor=c;
}
```

٦- نفذ البرنامج .

الملاحظات :

ستلاحظ أنك عند الضغط على الزر سيتغير لون form وعند كل ضغطة جديدة يظهر لون جديد. السبب في ذلك هو الكائن العشوائي (Random) حيث قمنا في السطر الأول بتعريف كائن من نوع Random ثم أنشأنا ثلاث متغيرات عشوائية مداها من ١ إلى ٢٥٥. ثم صممنا لونا عن طريق الدالة FormArgb. هذه الدالة تأخذ قيما للألوان الأساسية وتصنع لونا بناءا على هذه القيم. وأخيرا قمنا بجعل هذا اللون لون خلفية form عن طريق المتغير BackColor. فم كل ضغطة ستتغير القيم العشوائية و سيظهر لون جديد.

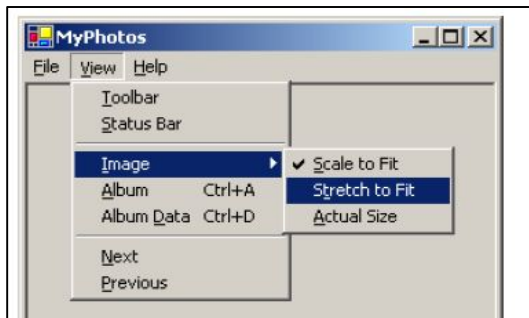


القوائم (Menus)

القوائم: هي إحدى الطرق المناسبة لتجميع الأوامر المتشابهة أو المرتبطة ببعضها البعض في مكان واحد. أنواع القوائم:

النوع الأول : main menu

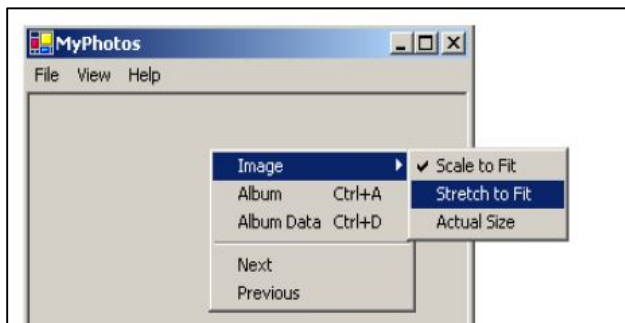
وهي عبارة عن قوائم تظهر عادة مرتبة أفقياً في أعلى النوافذ، وعند تفعيلها إما بالفأرة أو ب لوحة المفاتيح فإنها تظهر قوائم أسفلها. ومن الأمثلة الشائعة عليها قائمة الملف وقائمة العرض وقائمة المساعدة. ويظهر في الشكل (١-١) المقابل أحد الأمثلة عليها.



شكل (١-١)

النوع الثاني : context menu

وهي عبارة عن قوائم تظهر في وضع معين أو في موقف خاص، وتكون عبارة عن أوامر مرتبطة بشيء ما تظهر عند إعطاء أمر خاص مثل الضغط على الزر الأيمن للفأرة. الشكل (٢-١) يوضح ظهور قائمة تحتوي على عدة أوامر تتعلق بالصورة التي ستعرض وذلك عند الضغط على الزر الأيمن للفأرة.



شكل (٢-١)

- الآن سنقوم بدراسة كلا النوعين وسنتعلم كيف يمكننا
- تعريف أنواع القوائم المختلفة.
 - إنشاء قوائم وتعديلها.
 - التعامل مع menu events
 - التعامل مع عدة قوائم مع بعضها البعض.

أولاً: القوائم الرئيسية (main menus)

سنقوم بتوضيح كيفية إنشاء وتعديل القوائم الرئيسية من خلال هذا المثال . لنفرض أننا نريد إنشاء برنامج يقوم بعرض الصور ، هذا البرنامج سيتكون من عدة مهام أولاً تحميل الصورة (load) ثانياً إغلاق البرنامج (Exit) وغالباً ما يكون هذان الأمران في قائمة ملف (File).

خطوات الحل :

أولاً: فتح ملف جديد من نوع (widows application)

ثانياً: إضافة قائمة رئيسية .

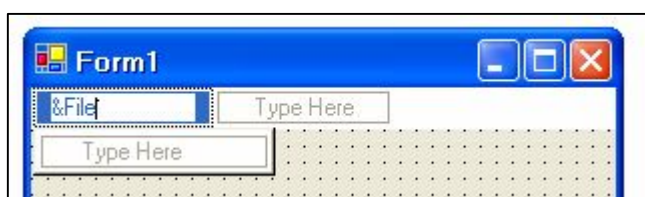
و ذلك عن طريق سحب أداة (MainMenu) من قائمة الأدوات (Toolbox) ووضعها على النافذة (form) الجديدة . كما هو موضح في الشكل (٣-١)



شكل (٣-١)

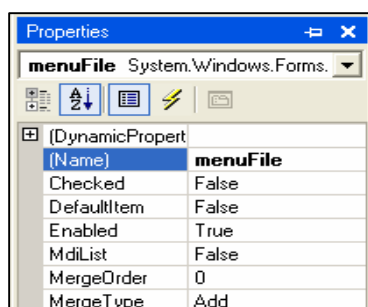
ثالثاً: إنشاء قائمة ملف (File)

✓ بعد إضافة (MainMenu) سيظهر أسفل (form) متغير اسمه تلقائياً (mainMenu1) قم بالضغط عليه ستظهر في الأعلى قائمة مكتوب عليها (Type Here) . قم بالضغط عليها واكتب (&File) وهذا هو اسم القائمة . إضافة إشارة & قبل حرف F تعني أن هذا الحرف سيصبح مفتاح النداء (access key) على قائمة (File) أي أننا يمكن استدعاء هذه القائمة بدون استخدام الفارة عن طريق الضغط على (Alt + F) . فمثلاً إذا أردنا جعل الحرف (e) هو حرف النداء علينا كتابة (Fil&e).



شكل (٤-١)

✓ الآن نريد تغيير اسم القائمة (Name property) وذلك عن طريق الضغط على (MainMenu) ثم زر F4 سنلاحظ ظهور قائمة تسمى (properties window) . اضغط على خاصية (Name) ثم اكتب الاسم الجديد وليكن (menuFile).



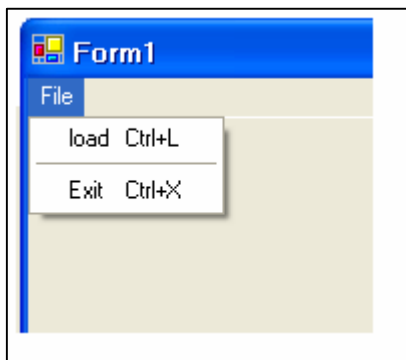
- ✓ لإضافة (menu item) باسم (load) علينا الضغط أسفل حسمه (HHE) ثم حسمه (&load) . لتغيير (Name property) إلى (menu load) علينا إتباع نفس الخطوات السابقة.

- ✓ لوضع فاصل (separator) بعد قائمة (load) يمكننا وضع إشارة (-) بدلا من كتابة اسم. الشكل المقابل يوضح شكل القائمة بعد إضافة فاصل.



شكل (٦-١)

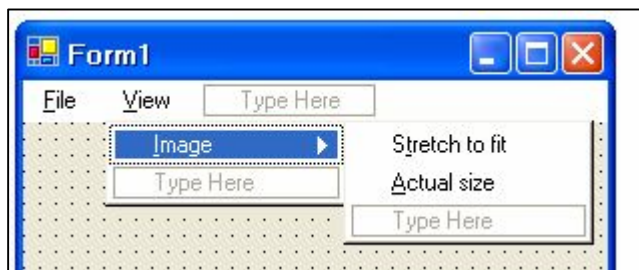
- ✓ أيضا يلزمنا تحديد المفاتيح التي ستستدعي قائمة (load) و ذلك من خلال خاصية (shortcut). اضغط على الخاصية ثم اختر (Ctrl + L).
- ✓ الآن يمكننا إضافة (menu item) باسم (E&xit) بإتباع الخطوات السابقة وتغيير الاسم إلى (menuExit). سيصبح البرنامج كما هو موضح في الشكل (٧-١).



شكل (٧-١)

رابعا: إنشاء قائمة العرض (view)

- ✓ بإتباع الخطوات السابقة يمكننا إضافة قائمة (view) و وضع (menu item) التالية (Image) ثم تعريف (menu item) فرعية داخل Image هي (Stretch to fit) و (Actual size) لتصبح القائمة كما هو موضح في الشكل (٨-١) المقابل مع مراعاة تغيير أسماء العناصر كما هو موضح في الجدول التالي



شكل (٨-١)

Name	MenuItem
menuView	view
menuImage	image
menuStretch	Stretch to fit
menuActual	Actual Size

خامسا: إضافة (Handler) للعنصر (load)
إضافة (Handler) مهم جدا لجعل القوائم الرئيسية (MainMenu) و العناصر الفرعية (menu item) التابعة لها فعالة ، أي يمكنها تنفيذ مهام عند حدوث (event)، لذلك برنامجنا حتى الآن لا يمكنه تنفيذ شيء أي عند الضغط على عنصر (load) لن يحدث أي شيء لذلك لا بد من إضافة (Handler).

✓ لإضافة (Handler) للعنصر (load) يجب الضغط أولا على قائمة (File) ثم الضغط مرتين على قائمة (load) سنلاحظ إنشاء دالة (function) هذه الدالة سوف تستدعي عند الضغط على عنصر (load) . لذلك يمكننا كتابة المهام التي نريد أن ننفذها عند الضغط على (load) داخل هذه الدالة لذلك يسمى هذا النوع (Click Event).
✓ لكي يمكننا جعل برنامجنا يعرض صور لا بد أولا من إضافة (PictureBox) من قائمة (ToolBox) وذلك لتعرض الصور عليها ، أيضا سنغير اسمه إلى (pbxPhoto) ، وذلك كما تم توضيحه سابقا.
✓ الآن يمكننا إضافة المهمة التي نريد و هي عرض (File Dialog) لنبحث عن طريقه عن صور و عند اختيار إحدى الصور يتم عرضها . هذه المهام سيقوم بها البرنامج التالي عند كتابته داخل الدالة التي تسمى (menuLoad_Click)

```
private void menuLoad_Click(object sender, System.EventArgs e)
{
    OpenFileDialog dlg = new OpenFileDialog();
    dlg.Title = "Load Photo";
    dlg.Filter = "jpg files (*.jpg)"
        + "|*.jpg|All files (*.*)|*.*";
    if (dlg.ShowDialog() == DialogResult.OK)
    {
        try
        {
            pbxPhoto.Image = new Bitmap(dlg.OpenFile());
        }
        catch (Exception ex)
        {
            MessageBox.Show("Unable to load file: "
                + ex.Message);
        }
    }
    dlg.Dispose();
}
```

سادسا: إضافة (Handler) للعنصر (Exit)
يمكن إضافة (Handler) بطريقة أخرى وذلك بإتباع الخطوات التالية
✓ اعرض (properties window) للعنصر (exit)
✓ اضغط على زر (Events)

✓ اضغط مرتين على (Click item) سوف يتم فتح نفس الدالة السابقة ولكن اسمها
(menuExit_Click).

✓ اكتب داخل الدالة جملة لإغلاق البرنامج (this.close())

ثانياً: التعامل مع عدة عناصر عن طريق handler واحد (Shared Handler)

سنقوم في هذا القسم بالتعامل مع العنصر (image) سيكون عمله كالتالي عند الضغط على (image) ثم على (Stretch to fit) سنلاحظ أن الصورة توسعت لتملأ picture Box تماماً أما إذا اخترنا (Actual Size) الصورة ستكون في حجمها الطبيعي فقط . سنقوم بذلك عن طريق خاصية (SizeMode) لصندوق الصورة (PictureBox) خاصية (SizeMode) تحتوي على الخيارين السابقين.

خطوات الحل :

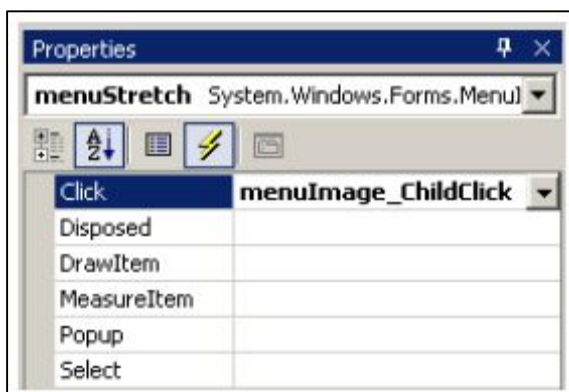
أولاً: تعريف مصفوفة من نوع (PictureBoxSizeMode) اسمها (modeMenuArray)
وذلك قبل تعريف (constructor)

```
private PictureBoxSizeMode[] modeMenuArray =
{
    PictureBoxSizeMode.StretchImage,
    PictureBoxSizeMode.Normal
};
```

ثانياً: تعريف متغير من نوع (integer) اسمه (_selectedImageMode) وذلك بعد تعريف المصفوفة السابقة

```
private int _selectedImageMode = 0;
```

ثالثاً: إضافة handler للعنصر (Stretch to fit) اسمه (menuImage_ChildClick)
✓ اعرض (properties window) للعنصر (Stretch to fit)
✓ اضغط على زر (Events)
✓ اضغط على الفراغ على يمين الخاصية (Click)
✓ ادخل اسم (menuImage_ChildClick)
✓ اضغط (Enter)



شكل (٩-١)

رابعاً: إضافة handler للعنصر (Actual Size) اسمه (menuItem_ChildClick) أي أننا سنجعل handler واحد لكل العنصرين اسمه (menuItem_ChildClick) وذلك بإتباع نفس الخطوات لكن هذه المرة لن ندخل الاسم ولكن سنجده جاهز مخزن وما علينا إلا اختياره بالنقر عليه علي يمين الخاصية (click).

خامساً: انقر مرتين على (menuItem_ChildClick) واكتب ما يلي :

```
private void menuItem_ChildClick(object sender, System.EventArgs e)
{
    if (sender is MenuItem)
    {
        MenuItem mi = (MenuItem)sender;
        _selectedImageMode = mi.Index;
        pbxPhoto.SizeMode = modeMenuArray[mi.Index];
        pbxPhoto.Invalidate();
    }
}
```

هذا البرنامج سيقوم أولاً بالتحقق من أن الكائن المرسل (sender) من نوع (menuItem) وذلك لتفادي أي أخطاء ثم سيقوم بتحويل الكائن (sender) إلى كائن من نوع (menuItem) ثم سيضع في المتغير الذي قمنا بتعريفه (_selectedImageMode) سيضع فيه قيمة (Index) وهي التي تحدد أي العنصرين تم اختياره فمثلاً رقم واحد يعني أن المستخدم اختار (Stretch to fit) أما الرقم اثنان يعني أن المستخدم اختار (Actual Size) وباستخدام هذا المتغير سنغير خاصية (SizeMode) لصندوق الصورة وذلك عن طريق المصفوفة التي قمنا بتعريفها ، أخيراً سنقوم بإعطاء أمر لعرض الصورة بخواصها الجديدة وذلك عن طريق الدالة (Invalidate).

ثالثاً: التعامل مع (Popup Events)

عادة يحتاج المستخدم لعلامة توضح له أي الخيارات مطبق الآن ليتمكن من اختيار الخيار المناسب ، في برنامجنا حتى الآن لا توجد هذه الميزة فمثلاً عند اختيارنا (Actual Size) لا يوجد ما يدل على أن الصورة مطبق عليها هذا الخيار ، لكن إذا ما أردنا تطبيق ذلك فلا بد من استخدام (popup events).

(popup events) : هو عبارة عن دالة تطبق ليس عند الضغط على العنصر كما في (click event) ولكنها تطبق قبل عرض (submenu) التابعة لها. فمثلاً عند وضع الفأرة على العنصر (image) تنفذ هذه الدالة ثم تعرض القائمة التي تحتوي على (Actual Size) و (Stretch to fit) . هذا يعني انه يمكننا استخدامها لتغيير مظهر (submenu) الخاصة ب image لتصبح أكثر وضوحاً للمستخدم.

الخطوات:

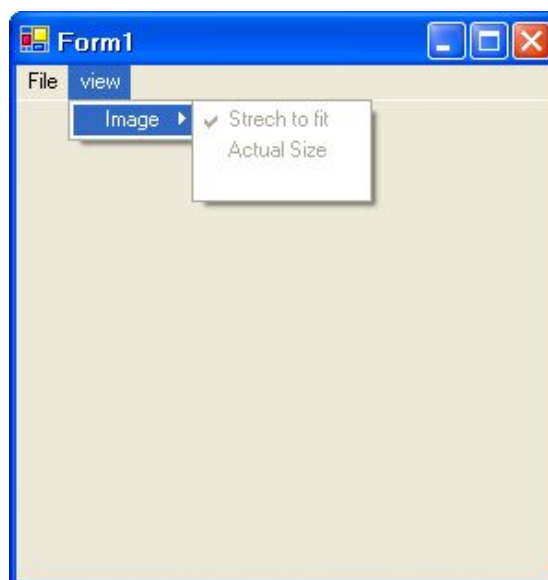
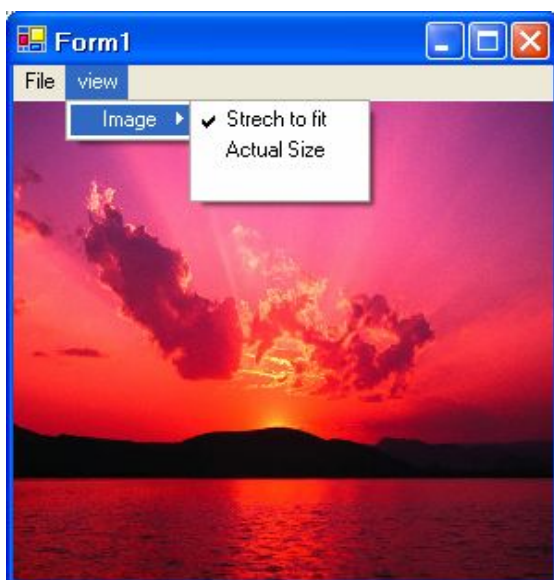
أولاً : إضافة (popup event) للعنصر (image) وذلك بالنقر مرتين على كلمة Popup

الموجودة في (Events) في (properties window) الخاصة بالعنصر image . سنلاحظ ظهور دالة اسمها (MenuImage_Popup).

ثانيا : أكتب البرنامج التالي داخل الدالة التي تم عرضها.

```
private void menuImage_Popup(object sender, System.EventArgs e)
{
    if (sender is MenuItem)
    {
        bool bImageLoaded
            = (pbxPhoto.Image != null);
        foreach (MenuItem mi in
            ((MenuItem) sender).MenuItems)
        {
            mi.Enabled = bImageLoaded;
            mi.Checked =
                (this._selectedImageMode == mi.Index);
        }
    }
}
```

سيقوم البرنامج التالي قبل عرض submenu التابعة للعنصر image بفحص كل خيار موجود في submenu فإذا وجدته هو المطبق الآن سيقوم بوضع علامة صح أمامه كما يوضح الشكل التالي وذلك عن طريق خاصية (checked) . ونلاحظ أيضا انه في حال عدم تحميل صورة لن تكن القائمة فعالة مما يجعل البرنامج أكثر مرونة .



شكل (١٠-١)

رابعاً: التعامل مع (Context Menu)

هي عبارة عن قائمة تظهر عند الضغط على الزر الأيمن من الفأرة وتكون دائماً مخصصة لعنصر معين. وعادة ما تستخدم لعرض الخيارات الخاصة بعنصر معين والتي نريدها سريعاً، فمثلاً في مثالنا نريد أن ننشأ (context menu) خاصة بصندوق الصورة (pictureBox) حيث تعرض الخيارات الموجودة في قائمة (View)، وطبعاً لأن (context menu) خاصة بالصورة فقط عند عدم تحميل صورة يجب أن تكون غير فعالة.

الخطوات:

أولاً: إضافة (context menu) وذلك عن طريق سحب أداة (contextMenu) من قائمة الأدوات (Toolbox) ووضعها على النافذة (form). ثم سنقوم بتغيير اسمها إلى (ctxtMenuView).

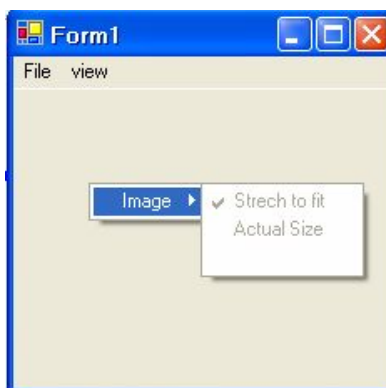
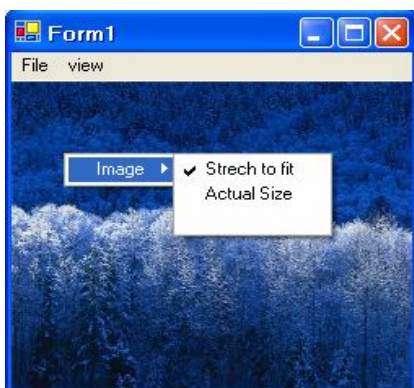
ثانياً: جعل (context menu) خاصة فقط بصندوق الصورة (pictureBox) وذلك عن طريق عرض (properties window) الخاصة بصندوق الصورة (pictureBox) ثم الضغط على الخاصية (contextMenu) ثم النقر على السهم واختيار اسم (ctxtMenuView) وهو الاسم الذي سميناه به (context menu) التي انشأنا.

ثالثاً: إضافة عناصر (context menu). يمكننا فعل ذلك كما فعلنا مع (mainMenu) لكن هناك طريقة أسهل، فنحن نريد نسخ محتويات القائمة (View) فقط. لذلك يمكننا إنشاء دالة تقوم بنسخ محتويات القائمة (View) إلى (contextmenu) واستدعاء هذه الدالة في (constructor). الدالة التالية تقوم بهذا العمل عن طريق دالة تسمى (CloneMenu)

```
private void DefineContextMenu()
{
    // Copy the View menu into ctxtMenuView
    foreach (MenuItem mi in menuView.MenuItems)
    {
        ctxtMenuView.MenuItems.Add(mi.Index, mi.CloneMenu());
    }
}
```

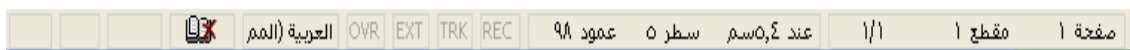
DefineContextMenu(); // this code must be inside the constructor

ونلاحظ في الشكل مظهر (context menu) عند وجود صورة وعند عدم وجودها



Status bars

Status Bar : هو مكان أو صندوق تعرض فيه معلومات سريعة تلزم المستخدم بشكل سريع و متكرر وتكون هذه المعلومات خاصة بالمهمة الحالية أو مكان مؤشر الكتابة . فعلى سبيل المثال في صفحة Microsoft word تستخدم status Bar لعرض المعلومات التالية (رقم الصفحة – مكان المؤشر – رقم السطر – العدد الكلي للصفحات) . شكل (١-٢) يوضح status bar خاص لبرنامج word.



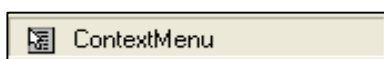
شكل (١-٢)

يتكون Status Bar من عدة مساحات صغيرة تسمى panels ، قد يحتوي Status Bar على عدد كبير من panels أو قد لا يحتوي على أي منها. ففي الشكل (١-٢) يحتوي Status Bar على ١٣ panel .

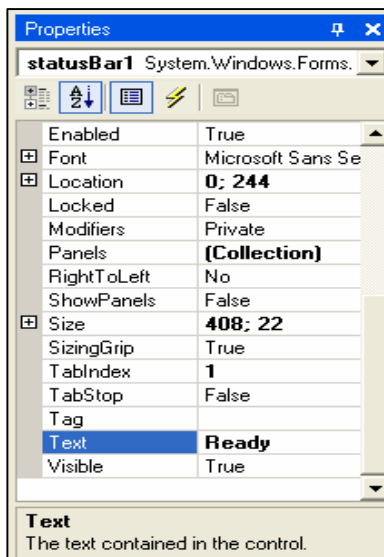
كيفية الاستخدام: سنشرح كيفية الاستخدام من خلال المثال الذي قمنا باستخدامه في موضوع القوائم وهو برنامج عرض الصور. في هذا الدرس سنقوم بإضافة status bar يحتوي على ثلاث panels سيكون وظيفة الأولى عرض اسم الملف الذي تم فتحه ، و سيكون وظيفة الثانية عرض حجم الصورة المعروضة ، أما الثالثة سيكون وظيفتها عرض النسبة المئوية للصورة المعروضة بالنسبة لحجم form . سنبدأ بالخطوات على فرض أن برنامج عرض الصور جاهز.

الخطوة الأولى : إضافة Status Bar

- ١ - فتح نافذة Toolbox
- ٢ - اختيار أداة StatusBar وقم بسحبها ثم وضعها على form



٣ - عدل خاصية تسمى (text) إلى Ready



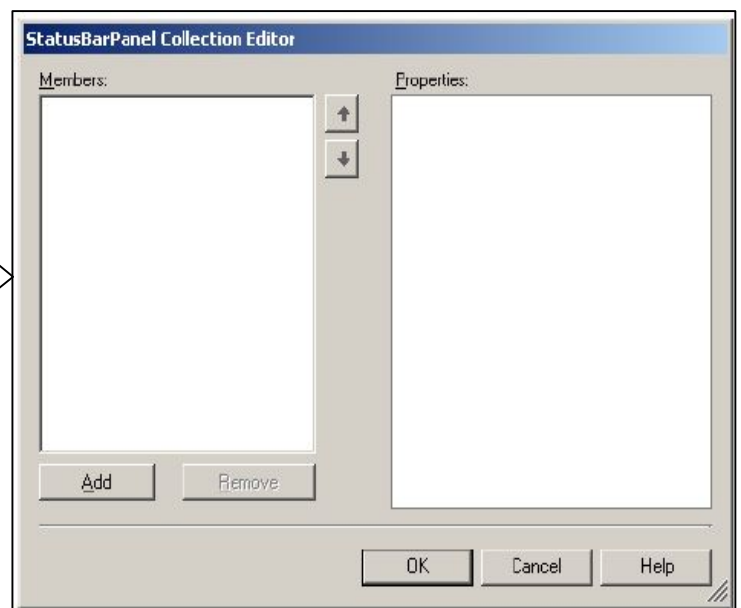
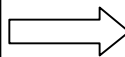
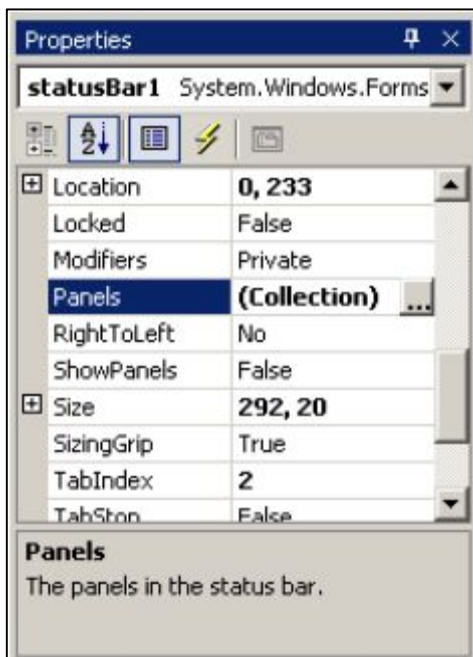
الخطوة الثانية : تعديل خاصية (text) لتناسب فتح صورة

- ١- افتح الدالة menuLoad_click
- ٢- أضف رسالة (loading) قبل تحميل الصورة .و رسالة (loaded) بعد تحميل الصورة
كما هو موضح في الأسفل.
- ٣- أضف رسالة (unable to load) عند عدم فتح الصورة.

```
try
{
    statusBar1.Text
        = "Loading " + dlg.FileName;
    pbxPhoto.Image
        = new Bitmap(dlg.OpenFile());
    statusBar1.Text
        = "Loaded " + dlg.FileName;
}
catch (Exception ex)
{
    statusBar1.Text
        = "Unable to load " + dlg.FileName;
    MessageBox.Show(
        "Unable to load file: "
        + ex.Message);
}
```

الخطوة الثالثة : إضافة panels و تغيير خصائصها

- ١- اعرض properties window الخاصة بأداة statusBar1
- ٢- اختر الخاصية Collections ثم انقر على المربع الصغير على يمينها لعرض صفحة التعامل مع panels



- ٣- أضف panel الأولى و ذلك بالضغط على (Add) و غير خصائصها كما يوضح الجدول التالي :

Settings	
Property	Value
(Name)	sbpnlFileName
AutoSize	Spring
BorderStyle	None
ToolTipText	Image File Name

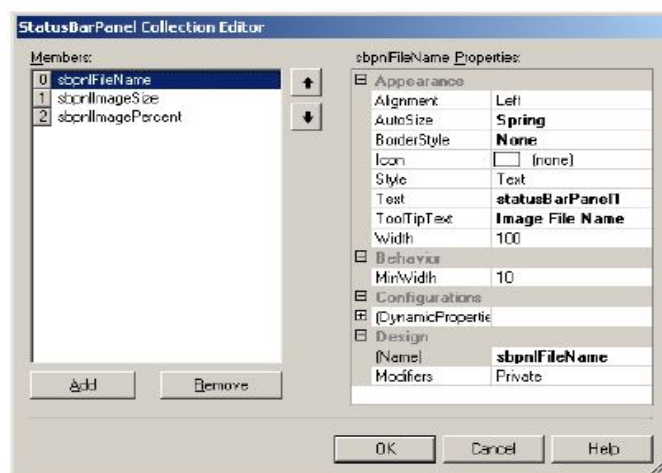
٤- أضف panel الثانية و ذلك بالضغط على (Add) مرة أخرى وغير خصائصها كما يوضح الجدول التالي :

Settings	
Property	Value
(Name)	sbpnlImageSize
AutoSize	Contents
ToolTipText	Image Size

٥- أضف panel الثالثة و ذلك بالضغط على (Add) مرة أخرى وغير خصائصها كما يوضح الجدول التالي :

Settings	
Property	Value
(Name)	sbpnlImage-Percent
Style	OwnerDraw
ToolTipText	Percent of Image Shown
Width	75

بعد هذه الخطوات سيكون شكل الصفحة كما هو موضح في الشكل



ملاحظة ١ : بالنسبة لخاصية (AutoSize) هناك خيارين هما :

- ✓ (spring) يعني أن المحتويات ومساحة panel ستتمدد مع تمدد form.
- ✓ (Contents) يعني أن مساحة panel ستتمدد حتى تتناسب حجم المحتويات فقط ولن تتمدد أكثر من ذلك حتى لو غيرنا حجم form.

ملاحظة ٢ : بالنسبة لخاصية (Style) هناك خيارين أيضا هما :

- ✓ (text) أي أن المحتوى سيكون نص عادي يتغير عن طريق خاصية (text)
- ✓ (OwnerDraw) أي أن المحتوى سيتغير عن طريق (DrawItem event)

الخطوة الرابعة : عرض النص والقيم على panels

- ١- افتح الدالة menuLoad_click واجعل الخاصية (ShowPanels) تساوي (false) و ذلك لأنه قبل عرض الصورة لا تعمل panels.
- ٢- داخل الدالة اكتب الأسطر التالية التي تقوم بعرض اسم الملف و حجم الصورة.
- ٣- اجعل الخاصية (ShowPanels) تساوي (true) وذلك ليتم عرض panels لأنها يجب أن تعرض بعد عرض الصورة فقط. سيصبح شكل الدالة menuLoad_click كما هو موضح في الأسفل :

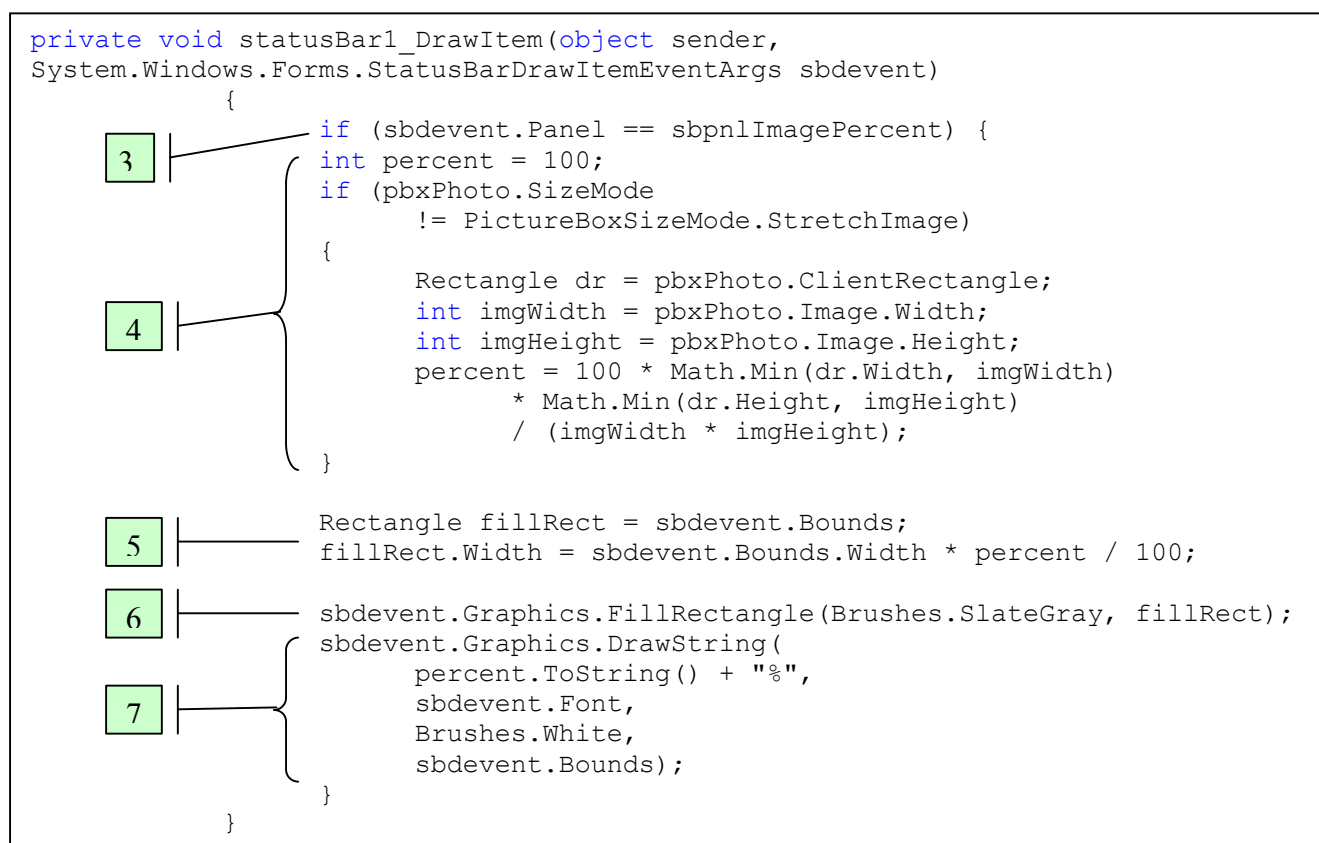
```
private void menuItem2_Click(object sender, System.EventArgs e)
{
    OpenFileDialog dlg = new OpenFileDialog();
    dlg.Title = "Load Photo";
    dlg.Filter = "jpg files (*.jpg)"
        + "|*.jpg|All files (*.*)|*.*";
    if (dlg.ShowDialog() == DialogResult.OK)
    {
        try
        {
            1 | this.statusBar1.ShowPanels = false;
            statusBar1.Text= "Loading " + dlg.FileName;
            pbxPhoto.Image = new Bitmap(dlg.OpenFile());
            2 | statusBar1.Text= "Loaded " + dlg.FileName;
            {
                this.sbpnlFileName.Text = dlg.FileName;
                this.sbpnlImageSize.Text = string.Format("{0:#} x {1:#}",
                    this.pbxPhoto.Width, this.pbxPhoto.Height);
            }
            3 | this.statusBar1.ShowPanels = true;
        }
        catch (Exception ex)
        {
            statusBar1.Text = "Unable to load " + dlg.FileName;
            MessageBox.Show("Unable to load file: " + ex.Message);
        }
    }
    dlg.Dispose();
}
```

الخطوة الخامسة: عرض النسبة المئوية للصور على panel باستخدام DrawItem

- ١- اعرض properties window الخاصة بأداة statusBar1
- ٢- انقر مرتين على الخاصية DrawItem نلاحظ فتح دالة تسمى

(statusBar1_DrawItem)

- ٣- تأكد من أن الكائن الذي سنتعامل معه من نوع (sbpnlImagePercent)
- ٤- أحسب النسبة المئوية للجزء المعروض من الصورة
- ٥- أحسب المساحة التي سنستخدمها من panel وذلك حسب النسبة المئوية السابقة
- ٦- أرسم شكل المستطيل (rectangle) وأعرضه على panel
- ٧- اعرض النسبة المئوية على panel



- ٨- الآن سنقوم بإعادة رسم (panel) فقط عندما يتغير عرض الصورة، طبعاً ذلك يحدث عند تغيير (SizeMode) وذلك في نهاية الدالة التي تسمى (menuImage_ChildClick).

statusBar1.Invalidate();