الباب الثالث
أساسيات عمل التطبيقات

٦. التعامل مع النماذج

٧. استخدام أدوات التحكم

٨. ضبط الخصائص Setting Properties.

٩. التعرف على الأحداث والإجراءات

١٠. تصميم القوائم وأشرطة الأدوات.

١١. استخدام المربعات الحوارية

١٢. إدارة الحلول والمشروعات.

obeikandl.com

obeikandl.com

الفصل السادس
التعامل مع النماذج
Forms

تبدأ أول خطوة في إعداد برنامج **Visual Basic** بتصميم الواجهة. والنموذج **(Form)** واحد من أهم عناصر واجهات البرامج. يستخدم النموذج عادة لتوضع بداخله الكائنات أو العناصر المختلفة **(Controls)** التى يتعامل معها البرنامج مثل النصوص والرسوم أو أزرار الأوامر...الخ.

بانتهاء هذا الفصل ستتعرف على ما يلي:

- المقصود بالنماذج.
- التحكم في النماذج أثناء التصميم .
- التحكم في النماذج أثناء التشغيل.
- إظهار النماذج.
- وراثة النماذج.

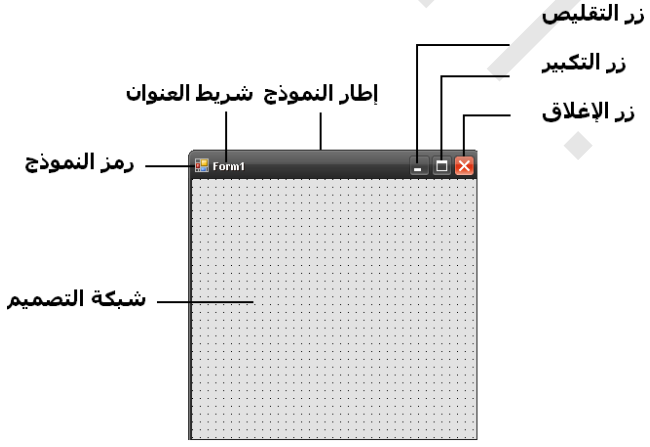
obeikandl.com

يمكن تقسيم برنامج Visual Basic إلى عنصرين رئيسيين. العنصر الأول هو العنصر المرئي، والعنصر الثاني هو العنصر البرمجي. العنصر البرمجي هو كود البرنامج الذي تكتبه لتأدية وظائف البرنامج ولا يراه المستخدم حين تشغيل البرنامج، أما العنصر المرئي أو بمعنى آخر واجهة المستخدم فهي النوافذ والنماذج وعناصر التحكم. وهي العناصر التي تتقبل مدخلات المستخدم وتستجيب لأوامره وخياراته.

ولأن النماذج بدون عناصر تحكم ليست لها فائدة، وعناصر التحكم بدون نموذج يحتويها لا تتحقق الفائدة منها، فان هذا الفصل والفصل التالي يكمل كل منهما الآخر.

النماذج Forms

فى بداية أى مشروع جديد فإن Visual Basic يفتح لك واجهة نافذة لنموذج فارغ (Form) لا يحتوى على أي أداة كما فى شكل٦-١. وعادة توضع داخل النموذج جميع الأدوات التى ستلزمك فى البرنامج الذى تقوم بتصميمه. إذاً النموذج عبارة عن كائن يعمل كحاوية للكائنات الأخرى (كالعناوين ومربعات النص ومربعات الرسم) التي تتكون منها في النهاية واجهة المستخدم. وعادةً يحتوي البرنامج على نموذج واحد على الأقل أو على عدة نماذج. ومع ذلك يمكنك إنشاء برنامج ليس به نماذج على الإطلاق (يقوم بمعالجة أو نسخ ملفات مثلا من سطر الأوامر).



شكل ٦-١نموذج النافذة هو نقطة البداية في واجهة المستخدم.

obeikandl.com

إذا أردت إضافة نموذجا جديدا فى برنامجك فى أى وقت فيجب اختيار Add Windows Form من قائمة Project حيث يظهر المربع الحوارى Add New Item. اختر Windows Form ثم انقر زر Add لإضافة النمـــوذج الجديد.

عندما تبدأ مشروعا نوافذياً جديدا يفتح لك Visual Basic المشـــروع الافتراضـــي والمكون من نموذج واحد فقط. يبين شكل ٦-١شكل النموذج كما يبدو في بيئة التصميم، وهو يحتوي على كل العناصر التي توجد في نوافذ البرامج حال تشغيلها.فهو يحتوي علـــى شريط عنوان وقائمة التحكم وعدة أزرار للتحكم (تكبير،تصغير، إغلاق). لاحـــظ أن زر الإغلاق يبدو دائما أثناء التصميم حتى لو اخترت إخفاءه أثناء التشغيل .

العنوان الافتراضى الذى يخصصه Visual Basic للنموذج فى بادئ الأمر هو Form1 حيث أنها أول واجهة فى البرنامج وعند إضافة واجهة جديدة فإن العنـــوان ســـيتغير إلى Form2 وهكذا يضيف البرنامج رقماً تسلسلياً لرقم آخر واجهة فى كل مرة تفتح واجهة جديدة. إذا أردت أن توقف هذا التسلسل اختر New Project من القائمة File لفـــتح مشروع جديد.

تظهر أرضية النموذج أثناء التصميم على هيئة شبكة نقطية Grid كى تسمح لك بمحاذاة العناصر على النموذج. يمكنك تغيير صفات الشبكة من خلال مربع الخصـــائص، حيـــث يمكنك إخفاء الشبكة كليا أو إظهارها مع عدم التقيد بها عند إدراج العناصر (حيث أنـــه عند تمكينها يتم ضبط الركن الأيسر العلوي للكائن على أقرب نقاط الشبكة إليه) أو تغيير المسافة بين نقاط الشبكة (أقرب أو أبعد حسب الحاجة).

التحكم في النماذج

يمكنك التحكم في النموذج أثناء تصميمه بأن تسحبه من مقابض التحجيم أو بـــأن تحدد قيم الخاصيتين Height "ارتفاع" و Width "عرض" من مربع الخصـــائص، كمـــا يمكنك كذلك التحكم في مكان وحجم النموذج أثناء تشغيل البرنامج (أي بالكود).

openkandil.com

التحكم في الحجم أثناء التصميم

لأن برنامج Visual Basic واحد من البرامج التى تعمل تحت بيئة Windows فانه يتمتع بالمزايا التى تتمتع بها برامج Windows ومنها تحجيم نوافذ برامج Visual Basic أو إحدى النوافذ التى تظهر أثناء التعامل معه أو نقلها من مكانها إلى مكان آخر. ونوضح ذلك فيما يلى:

التحجيم الأقصى والأدنى

المقصود بتحجيم النماذج تكبيرها بحيث تشمل كل المساحة المتوفرة وهو ما يسمى بالتحجيم الأقصى (Maximizing) أو تقليصها بحيث تظهر على شكل رمز فى أسفل الشاشة وهو ما يسمى بالتحجيم الأدنى (Minimizing) ، أو إعادتها إلى حجمها السابق قبل تكبيرها أو تقليصها (Restoring) ، وفيما يلى نوضح خطوات التحجيم الأقصى للواجهة ثم إعادتها لحجمها السابق (قم أولاً بتشغيل البرنامج).

١. وجه المؤشر إلى الركن الأيمن العلوى من النافذة ثم انقر زر التكبير تغطى النافذة واجهة الشاشة كلها.
٢. انقر زر استعادة الحجم ، تعود النافذة إلى حجمها السابق قبل التكبير.
٣. انقر زر التقليص ، تظهر نافذة الواجهة على شكل رمز بهذا الشكل Form1 فى أسفل الشاشة.
٤. انقر رمز الواجهة نقرا مزدوجا بعد تقليصها. ترجع إلى مكانها وحجمها قبل التحجيم الأدنى.

تغيير حجم نافذة النموذج

نقصد بتغيير حجم نافذة النموذج تكبيرها عما هى عليه أو تصغيرها عما هى عليه، وهذا الوضع بخلاف التحجيم الأقصى والاستعادة الذى شرحناه فى البند السابق، فهو يسمح باختيار الحجم الذى يناسبك وليس بالتحجيم الأقصى أو الأدنى فقط. لتغيير حجم نافذة النموذج بحيث تصبح أكبر مما هى عليه اتبع الخطوات التالية:

obeikandl.com

١. حرك مؤشر الفأرة إلى الحافة اليسرى للنافذة، يتحول الـمؤشر إلى سهم ذو رأسين هكذا ⇔.

٢. اضغط زر الفأرة الأيسر واستمر ضاغطا أثناء سحب السهم ذو الرأسين فى اتجاه اليسار للنافذة. تظهر حافة مظللة تحت السهم لتوضح حدود النافذة من الناحية اليسرى.

٣. عندما تشعر أن الحافة المظللة فى المكان المطلوب ارفع إصبعك من على زر الفأرة، يتم تكبير حجم النافذة يدويا.

توجيه المؤشر إلى ركن النافذة بدلا من حافتها يتيح لك تغيير الحجم أفقيا ورأسيا.

ضبط مكان النموذج بطريقة السحب

٤. يمكنك نقل أى نموذج إلى أى مكان داخل الشاشة. لنقل نافذة النموذج إلى أسفل الشاشة اتبع الآتى:

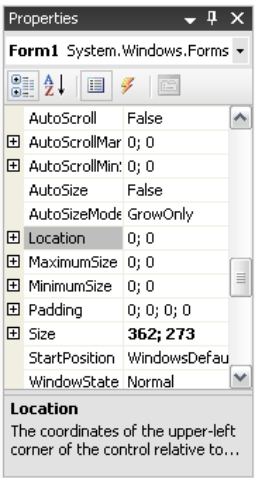
٥. وجه المؤشر إلى شريط العنوان. سيتحول المؤشر إلى سهم هكذا ▌↖.

٦. اسحب شريط العنوان لأسفل، وعندما تشعر أن البرواز المظلل أصبح فى المكان المطلوب ارفع إصبعك من على الزر. تنتقل نافذة الواجهة إلى أسفل الشاشة.

ضبط مكان النموذج باستخدام مربع الخصائص

يمكنك التحكم في مكان النموذج باستخدام خاصتي **X** و**Y** داخل الخاصية **Location** (انظر شكل ٦-٢). حيث تحدد الخاصية **X** لأي كائن بُعد حافته اليسرى عن حافة حاويته اليسرى، بينما الخاصية **Y** تحدد بُعد حافته العلوية عن حافة حاويته العليا. في حالة النافذة العادية (أي غير الموجودة في واجهة متعدد الوثائق (**MDI**) تكون الحاوية هي الشاشة بأكملها، وفي حالة عناصر التحكم تكون الحاوية هي النافذة، وبعض العناصر تستخدم أيضا كحاويات مثل مربع الرسم **PictureBox** والإطار **Frame**.

obeikandl.com



شكل ٦-٢ ضبط مكان النموذج من خلال مربع الخصائص.

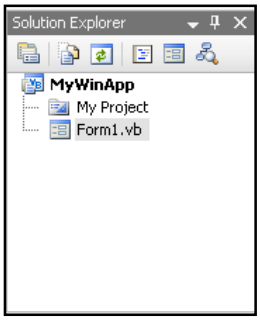
النماذج الوليدة في التطبيقات متعددة الوثائق يحدد موقعها بالنسبة للنافـــذة الأم كمـــا في شكل ٦-٣ . وقد تأخذ الخصائص X و Y قيما سالبة وفي هذه الحالة تكون العناصر مختفية جزئيا أو كليا.

شكل ٦-٣ النماذج الوليدة داخل النافذة الأم.

obeikandl.com

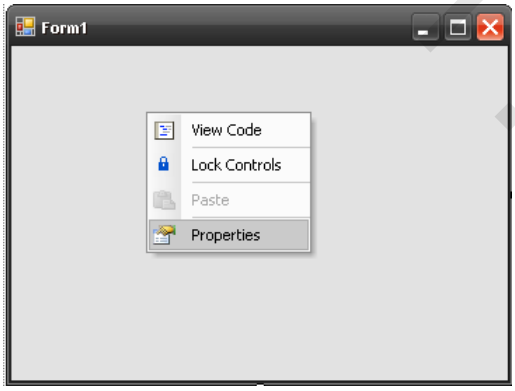
فتح نموذج آخر

يظهر في بيئة تطوير **Visual Studio 2008** داخل نافذة مستكشف الحل أسماء النماذج المستخدمة في هذا المشروع كما في شكل ٦-٤.



شكل ٦-٤ أسماء النماذج داخل نافذة المستكشف.

لإظهار أي نموذج غير النموذج المعروض في بيئة التصميم، انقر نقرا مزدوجا فوق اسم النموذج. ستظهر نافذة النموذج بحجمها المناسب. انقر الواجهة بالزر الأيمن ، تظهر قائمة مختصرة (موضعية) تحتوى على أوامر التعامل مع النموذج مثل الموجودة فى شكل ٦-٥، ويتم استدعاء أحد أوامر القائمة بمجرد النقر عليه فمثلاً نقر الأمر **Properties** فى هذا المثال يظهر مربع الخصائص الخاص بالواجهة، بدلا من النقر على الواجهة لتنشيطها ثم الضغط على المفتاح **F4** .



شكل ٦-٥ إظهار القائمة المختصرة (الموضعية).

obeikandi.com

إظهار النموذج

لن تحتاج لمسألة إظهار النموذج إذا كنت ستنشئ برنامجاً يتكون من نموذج واحد فقط حيث يسمى هذا النموذج بنافذة البدء **Startup Form** ، ويقوم **Visual Basic** تلقائيا بتحميل النموذج إلى الذاكرة ثم إظهاره على الشاشة، وسوف يستمر البرنامج في العمل ما لم تغلق هذه النافذة بالطرق المعروفة.

يمكنك تغيير نافذة البدء من مربع الحوار ***Project*** **Properties** إلى أي نافذة. لإظهار هذا المربع، افتح قائمة **Project** بشريط القوائم ثم اختر ***Project*** **Properties** من القائمة المنسدلة الناتجة (حيث **Project** عبارة عن اسم المشروع الحالى). ويمكن أيضا أن يبدأ البرنامج من إجراء فرعي يسمى **Main** في أحد الوحدات البرمجية **Modules** بدلاً من النماذج.

عندما يحتوي برنامجك على عدة نماذج فلابد أن تعرف كيف ومتى تظهر النوافذ المختلفة ويتم التحكم في ذلك من خلال الوظائف **Show** و**ShowDialog** و **Hide** و**Close** الخاصة بالنافذة.

لإظهار النافذة الغير منبثقة أى التى لا تعتمد على غيرها، تستعمل الوظيفة **Show** التى تقوم بتحميل النافذة إلى الذاكرة أولاً ثم إظهارها على الشاشة بعد ذلك وصيغتها هى:

***frmName*.Show()**

حيث تعبر **frmName** على اسم النموذج. ولإظهار النافذة المنبثقة أى التى يجب الاستجابة لها قبل غيرها (مثل المربع الحوارى **Save As**)، تستعمل الوظيفة **ShowDialog()** التى تقوم بتحميل النافذة إلى الذاكرة أولاً ثم إظهارها على الشاشة بعد ذلك وصيغتها هى:

***frmName*.ShowDialog()**

إن لم تعد بحاجة إلى نافذةٍ، فيمكن إخفائها من الشاشة على أن تظل بالذاكرة وذلك باستخدام الوظيفة **Hide()** وصيغتها هى:

***frmName*.Hide()**

أما إذا أردت إغلاق النافذة نهائياً، فيمكنك استخدام الوظيفة **Close()** وصيغتها هى:

openkanal.com

***frmName*.Close()**

تحسين أداء البرنامج

إذا احتوى برنامجك على نوافذ تعرض وتخفى مرارا أثناء تشغيل البرنامج فإن عملية التحميل في كل مرة ستستغرق الكثير من الوقت خاصة عند احتواء هذه النافذة على عناصر تحكم كثيرة، أحد الطرق لعلاج ذلك أن تقوم بتحميل هذه النوافذ مرة واحدة في بداية تشغيل البرنامج ثم تستخدم ()Show و ()Hide لإظهار وإخفاء النافذة عند الحاجة، سيسرع ذلك من الأداء إلا أنه إذا تم تحميل الكثير من النوافذ فسيؤدي ذلك إلى نتيجة عكسية بإبطاء النظام ككل. أيضا ينبغي إفراغ الذاكرة من هذه النوافذ بالأمر()Close فى حالة انتهاء استخدامك للنوافذ.

وراثة النماذج

تعتبر وراثة الكائنات من المبادئ الأساسية للغات البرمجة التى تدعم البرمجة المسيرة بالكائنات **Object Oriented Programming** أو ما يعرف اختصاراً **OOP**. ففى حالة النماذج على سبيل المثال، يمكنك إنشاء نموذج رئيسى يحتوى على السمات العامة التى ترغب فى إضافتها لعدد آخر من النماذج ثم وراثة النموذج الرئيسى فى كل نموذج من هذه النماذج ومن ثمَّ تضمين ما بالنموذج الرئيسى من عناصر وسمات مع القدرة على تغييرها أو الإضافة إليها.

ويمكنك داخل **Visual Basic 2008** توريث النماذج باستخدام المربع الحوارى **Inheritance Picker**.

للتعرف على كيفية استخدام المربع الحوارى **Inheritance Picker** فى وراثة النماذج، تابع معنا الخطوات الآتية:

١. قم بإنشاء مشروع نوافذى جديد كما تعلمت من قبل باسم مناسب وليكن **MyInheritance**.
٢. قم بإضافة زرى أمر للنموذج **Form1**، الأول **Button1** والثانى **Button2**.
٣. انقر الزر الأول نقراً مزدوجاً ثم قم بإدخال الكود التالى:

obeikandl.com

```
MsgBox("I am button1 in form1")
```

٤. قم بتكرار الخطوة السابقة مع الزر الثانى مع إدخال الكود التالى:

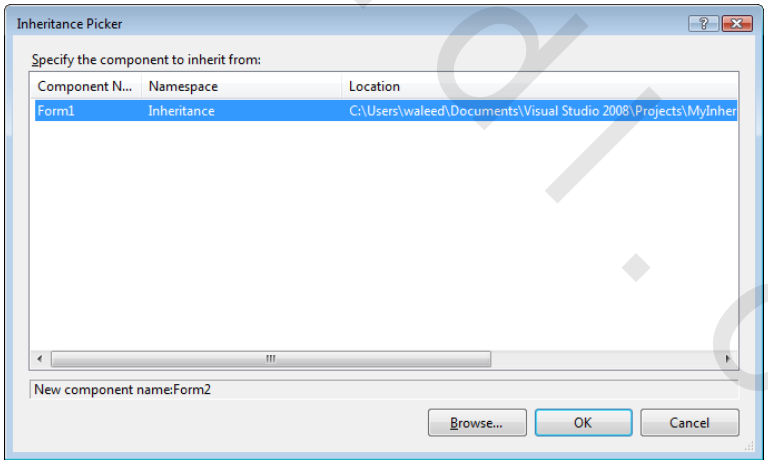
```
MsgBox("I am button2 in form1")
```

٥. قم بحفظ المشروع ثم افتح قائمة **Build** من شريط القوائم واختر **Build** من القائمة المنسدلة الناتجة لبناء مشروعك الجديد.

٦. قم بتشغيل المشروع ثم انقر أيٍ من الزرين، تحصل على مربع رسالة يخبرك بنقر الزر.

٧. افتح قائمة **Project** من شريط القوائم ثم اختر **Add New Item** من القائمة المنسدلة الناتجة، يظهر المربع الحوارى **Add New Item**.

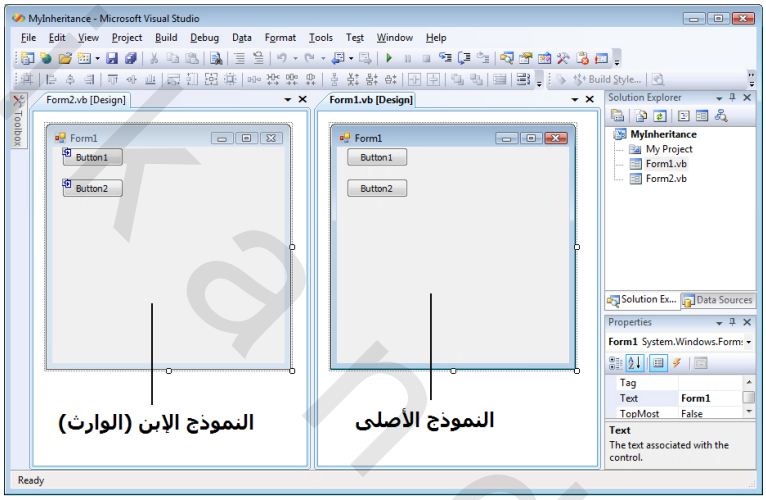
٨. اختر **Inherited Form** من المربع **Templates** بالجانب الأيمن وابق على الاسم الافتراضى للنموذج الموروث الجديد كما هو **Form2** ثم انقر زر **Add**. يتم إغلاق المربع الحوارى **Add New Item** ويظهر المربع الحوارى **Inheritance Picker** الذى يحتوى على جميع النماذج الموجودة بالمشروع (انظر شكل ٦-٦). لاحظ أننا لدينا نموذجاً واحداً.



شكل ٦-٦ المربع الحوارى **Inheritance Picker**.

٩. يظهر النموذج الوحيد **Form1** مختاراً. انقر زر **Ok** لإغلاق المربع الحوارى

Inheritance Picker وإنشاء النموذج الجديد Form2 الذى يرث بدوره النموذج Form1 ومن ثمَّ يحتوى الأول على جميع خصائص ووظائف وكائنات النموذج الأخير، ودلالة ذلك ظهور نفس الزرين بالنموذج الجديد، كما يظهر مؤشر بالركن الأيسر العلوى من كل زر ليدل على وراثة هذا الزر من نموذج آخر (انظر شكل ٦-٧).



شكل ٦-٧ النموذجان داخل بيئة التطوير.

١٠. قم بتشغيل المشروع مع جعل النموذج الثانى هو نموذج بدء التشغيل ثم انقر الزر Button1 أو Button2 تحصل على نفس الرسائل التى حصلت عليه عند نقر الزرين المصاحبين بالنموذج الأول.

سنتعرف على المزيد عن وراثة الكائنات بالجزء الثانى من هذه السلسلة إن شاء الله.