

## الفصل الثالث التعامل مع النماذج Forms

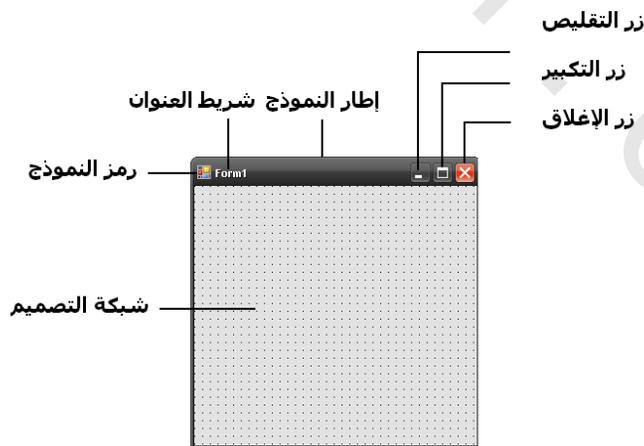
تبدأ أول خطوة في إعداد برنامج **Visual Basic** بتصميم الواجهة. والنموذج (**Form**) واحد من أهم عناصر واجهات البرامج. يستخدم النموذج عادة لتوضيح بداخله الكائنات أو العناصر المختلفة (**Controls**) التي يتعامل معها البرنامج مثل النصوص والرسوم أو أزرار الأوامر... الخ.  
بانتهاء هذا الفصل ستعرف على ما يلي:

- ◆ المقصود بالنماذج.
- ◆ التحكم في النماذج أثناء التصميم.
- ◆ التحكم في النماذج أثناء التشغيل.
- ◆ إظهار النماذج.
- ◆ وراثه النماذج.

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 تحقق الفائدة منها، فإن هذا الفصل والفصل التالي يكمل كل منهما الآخر.

## النماذج Forms

في بداية أى مشروع جديد فإن Visual Basic يفتح لك واجهة نافذة لنموذج فارغ (Form) لا يحتوي على أي أداة كما في شكل ٣-١. وعادة توضع داخل النموذج جميع الأدوات التي ستلزمك في البرنامج الذي 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 مثلاً من سطر الأوامر).



شكل ٣-١ نموذج النافذة هو نقطة البداية في واجهة المستخدم.

إذا أردت إضافة نموذجاً جديداً في برنامجك في أي وقت فيجب اختيار **Add Windows Form** من قائمة **Project** حيث يظهر المربع الحوارى **Add New Item**. اختر **Windows Form** ثم انقر زر **Add** لإضافة النموذج الجديد.



عندما تبدأ مشروعاً نوفاً جديداً يفتح لك **Visual Basic** المشروع الافتراضي والمكون من نموذج واحد فقط. يبين شكل ٣-١ شكل النموذج كما يبدو في بيئة التصميم، وهو يحتوي على كل العناصر التي توجد في نوافذ البرامج حال تشغيلها. فهو يحتوي على شريط عنوان وقائمة التحكم وعدة أزرار للتحكم (تكبير، تصغير، إغلاق). لاحظ أن زر الإغلاق يبدو دائماً أثناء التصميم حتى لو اخترت إخفاءه أثناء التشغيل.

العنوان الافتراضي الذي يخصصه **Visual Basic** للنموذج في بادئ الأمر هو **Form1** حيث أنها أول واجهة في البرنامج وعند إضافة واجهة جديدة فإن العنوان سيتغير إلى **Form2** وهكذا يضيف البرنامج رقماً تسلسلياً لرقم آخر واجهة في كل مرة تفتح واجهة جديدة. إذا أردت أن توقف هذا التسلسل اختر **New Project** من القائمة **File** لفتح مشروع جديد.

تظهر أرضية النموذج أثناء التصميم على هيئة شبكة نقطية **Grid** كي تسمح لك بمحاذاة العناصر على النموذج. يمكنك تغيير صفات الشبكة من خلال مربع الخصائص، حيث يمكنك إخفاء الشبكة كلياً أو إظهارها مع عدم التقيد بها عند إدراج العناصر (حيث أنه 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

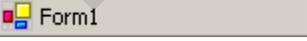
يمكنك التحكم في النموذج أثناء تصميمه بأن تسحبه من مقابض التحجيم أو بأن تحدد قيم الخاصيتين **Height** "ارتفاع" و **Width** "عرض" من مربع الخصائص، كما يمكنك كذلك التحكم في مكان وحجم النموذج أثناء تشغيل البرنامج (أي بالكود).

## التحكم في الحجم أثناء التصميم

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

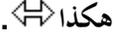
### التحجيم الأقصوالأدنى

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

1. وجه المؤشر إلى الركن الأيمن العلوى من النافذة ثم انقر زر التكبير ، تغطي النافذة واجهة الشاشة كلها.
2. انقر زر استعادة الحجم ، تعود النافذة إلى حجمها السابق قبل التكبير.
3. انقر زر التقليص ، تظهر نافذة الواجهة على شكل رمز بهذا الشكل  Form1 في أسفل الشاشة.
4. 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 هي عليه أو تصغيرها عما هي 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 هي عليه اتبع الخطوات التالية:

١. حرك مؤشر الفأرة إلى الحافة اليسرى للنافذة، يتحول المؤشر إلى سهم ذو رأسين هكذا .
٢. اضغط زر الفأرة الأيسر واستمر ضاغطا أثناء سحب السهم ذو الرأسين في اتجاه اليسار للنافذة. تظهر حافة مظلمة تحت السهم لتوضح حدود النافذة من الناحية اليسرى.
٣. عندما تشعر أن الحافة المظلمة في 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.

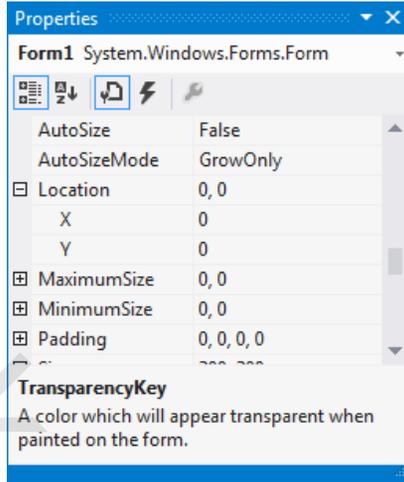


#### ضبط مكان النموذج بطريقة السحب

٤. يمكنك نقل أي نموذج إلى أي مكان داخل الشاشة. لنقل نافذة النموذج إلى أسفل الشاشة اتبع الآتي:
٥. وجه المؤشر إلى شريط العنوان. سيتحول المؤشر إلى سهم هكذا .
٦. اسحب شريط العنوان لأسفل، وعندما تشعر أن البرواز المظلل أصبح في 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 ٣-٢). 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.



شكل ٣-٢ ضبط مكان النموذج من خلال مربع الخصائص.

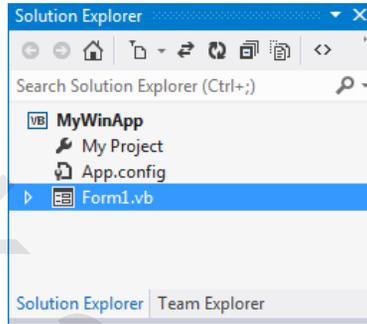
النماذج الوليدة في التطبيقات متعددة الوثائق يحدد موقعها بالنسبة للنافذة الأم كما في شكل ٣-٣. وقد تأخذ الخصائص X و Y قيمًا سالبة وفي هذه الحالة تكون العناصر مخفية جزئياً أو كلياً.



شكل ٣-٣ النماذج الوليدة داخل النافذة الأم.

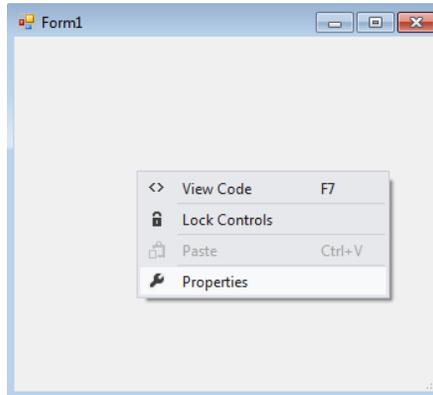
## فتح نموذج آخر

يظهر في بيئة تطوير **Visual Studio 2012** داخل نافذة مستكشف الحل أسماء النماذج المستخدمة في هذا المشروع كما في شكل ٣-٤ .



شكل ٣-٤ أسماء النماذج داخل نافذة المستكشف.

لإظهار أي نموذج غير النموذج المعروف في بيئة التصميم، انقر نقرا مزدوجا فوق اسم النموذج. ستظهر نافذة النموذج بحجمها المناسب. انقر الواجهة بالزر الأيمن ، تظهر قائمة مختصرة (موضعية) تحتوي على أوامر التعامل مع النموذج مثل الموجودة في شكل ٣-٥، ويتم استدعاء أحد أوامر القائمة بمجرد النقر عليه فمثلاً نقر الأمر **Properties** في هذا المثال يظهر مربع الخصائص الخاص بالواجهة، بدلا من النقر على الواجهة لتنشيطها ثم الضغط على المفتاح **F4** .



شكل ٣-٥ إظهار القائمة المختصرة (الموضعية).

## إظهار النموذج

لن تحتاج لمسألة إظهار النموذج إذا كنت ستنشئ برنامجاً يتكون من نموذج واحد فقط حيث يسمى هذا النموذج بنموذج البدء **Startup Form** ، ويقوم **Visual Basic** تلقائياً 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** بشرائط القوائم ثم اختر **Project Properties** من القائمة المنسدلة الناتجة (حيث **Project** عبارة عن اسم المشروع الحالي). ويمكن أيضاً أن يبدأ البرنامج من إجراء فرعي يسمى **Main** في أحد الوحدات البرمجية **Modules** بدلاً من النماذج.



عندما يحتوي برنامجك على عدة نماذج فلا بد 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 نافذة، 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:

**frmName.Close()**

## تحسين أداء البرنامج

إذا احتوى برنامجك على نوافذ تعرض وتحفى 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()** في حالة انتهاء استخدامك للنوافذ.

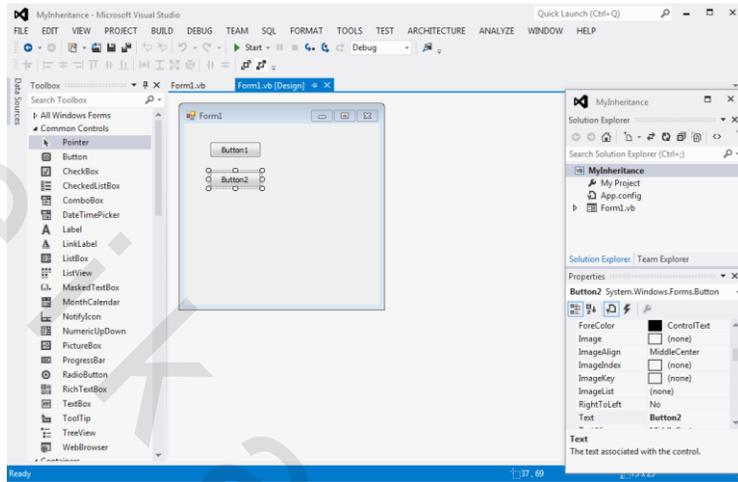
## وراثة النماذج

تعتبر وراثة الكائنات من المبادئ الأساسية للغات البرمجة التي تدعم البرمجة المسيرة بالكائنات **Object Oriented Programming** أو ما يعرف اختصاراً **OOP**. ففي حالة النماذج على سبيل المثال، يمكنك إنشاء نموذج رئيسي يحتوي على السمات العامة التي ترغب في إضافتها لعدد آخر من النماذج ثم وراثة النموذج الرئيسي في كل نموذج من هذه النماذج ومن ثم تضمين ما بالنموذج الرئيسي من عناصر وسمات مع القدرة على تغييرها أو الإضافة إليها. ويمكنك داخل **Visual Basic 2012** توريث النماذج باستخدام المربع الحوارى **Inheritance Picker**.

للتعرف على كيفية استخدام المربع الحوارى **Inheritance Picker** في 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**. لأداء ذلك، انقر رمز زر الأمر **Button** بمربع الأدوات بالجانب الأيسر من بيئة التطوير ثم اسحب الزر وقم بتحريره في المكان المناسب داخل النموذج (انظر شكل

٦-٣.



شكل ٦-٣ إضافة زرى الأمر إلى النموذج.

٣. انقر الزر الأول نقرأ مزدوجاً ثم قم بإدخال الكود التالي:

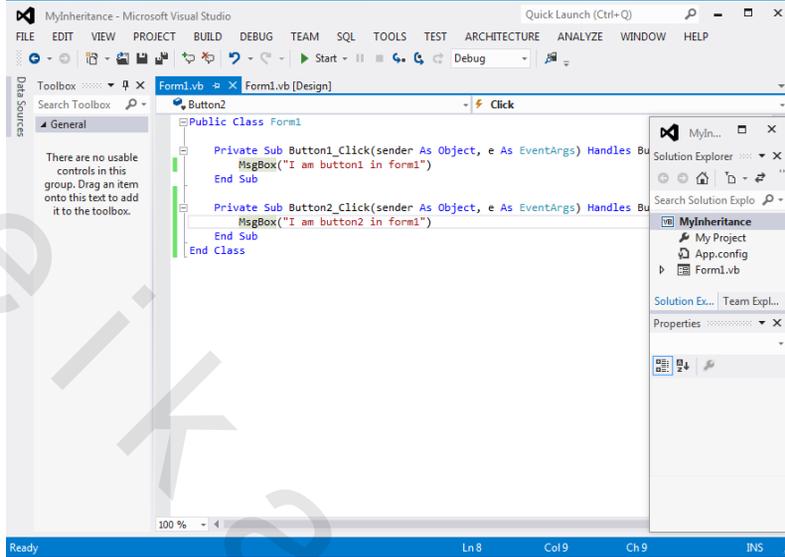
**MsgBox("I am button1 in form1")**

٤. قم بتكرار الخطوة السابقة مع الزر الثاني مع إدخال الكود التالي (انظر شكل ٣-٣)

(٧):

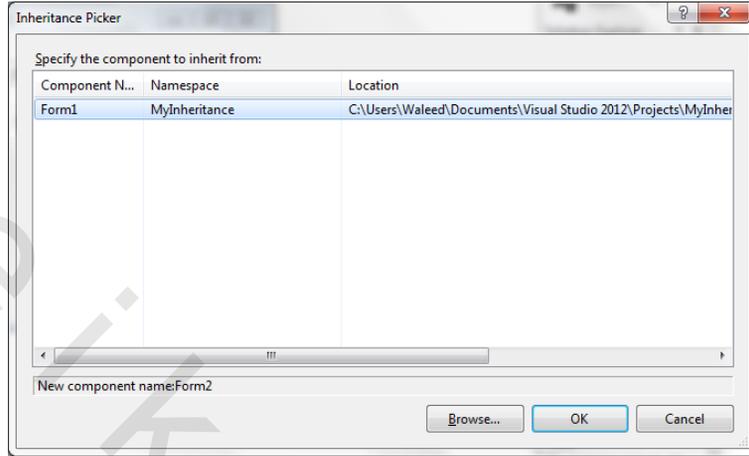
**MsgBox("I am button2 in form1")**

## الفصل الثالث: التعامل مع النماذج



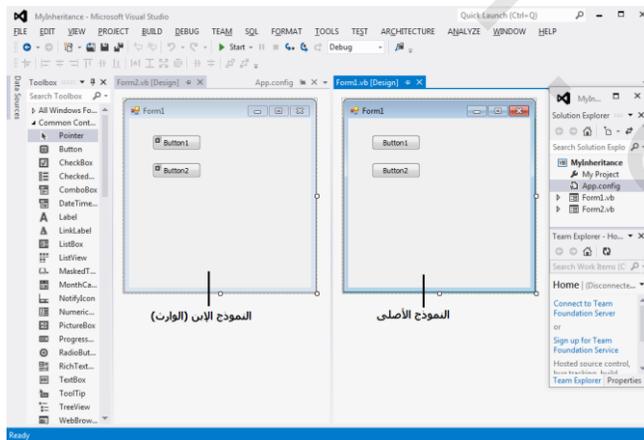
شكل ٣-٧ إضافة الكود إلى زرى الأمر.

٥. قم بحفظ المشروع ثم افتح قائمة **Build** من شريط القوائم واختر **BuildMyInheritance** من القائمة المنسدلة الناتجة لبناء مشروعك الجديد.
٦. قم بتشغيل المشروع ثم انقر أي 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 **Windows Forms** من الجانب الأيسر بالمربع الحوارى ثم اختر **Inherited Form** من المربع الموجود 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 ٣-٨). لاحظ أننا لدينا نموذجاً واحداً.



شكل ٣-٨ 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 ليبدل على وراثه هذا الزر من نموذج آخر (انظر شكل ٣-٩).



شكل ٣-٩ النموذجان داخل بيئة التطوير.

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

سنتعرف على المزيد عن وراثة الكائنات بالفصل الخامس عشر من هذا الجزء إن شاء الله.

