الفصل السادس
التعرف على الأحداث والإجراءات

بعد تصميم النموذج وتوقيع الأدوات على النماذج وضبط خصائص الكائنات التى تضعها عليها، تأتى مرحلة التفاعل مع مستخدم البرنامج وهي المرحلة التى تجعل هذه الأدوات تتأثر بالأحداث التى تقع عليها. فقد انتهى عصر البرامج التي تعمل من أولها إلى آخرها بتعليمات مسلسلة. وحاز المستخدم على قدرة أكبر من التحكم في وظائف البرنامج ووقت تنفيذها.

بانتهاء هذا الفصل ستتعرف على :

- المقصود بالحدث وكيفية الاستجابة للأحداث.
- كتابة إجراءات الحدث.
- أنواع الأحداث.
- فهم تسلسل الأحداث.

المقصود بالحدث Event

ربما كان مبدأ البرنامج المسير بالأحداث جديداً إلى حد ما. لكن فكرة الاستجابة بالحدث ليست كذلك.

أغلب ما في هذا العالم أشياء تسير بالأحداث. فمثلاً السيارة: بالضغط على مفتاح ما يمكنك إضاءة الأنوار، بالضغط على زر أيضا يمكنك إطلاق البوق، ويمكنك تغير سرعة السيارة بالمقدار الذي تريد بالضغط على دواسة البنزين، بل يمكنك إيقافها تماماً بالضغط على الفرامل.

هذا النوع من التحكم يمكنك من التحكم في كيفية ووقت تنفيذ الحدث وهو نموذج للبرامج المسيرة بالأحداث.

حاول دائما أن يكون برنامجك طبيعي ومشابه للحياة الواقعية، بمعني توقع من المستخدم أن يقوم بتصرفات تشبه سلوكه مع الأشياء الحقيقية كالمذياع والهاتف، وخلافها. مثلا إن كان البرنامج يصدر أصوات توقع من المستخدم أن يبحث عن زر يقوم بالتحكم في الصوت.

وغير تشابه البرامج المسيرة بالأحداث مع الواقع، هناك فائدة أخرى للبرمجة المسيرة بالأحداث وهى إمكانية استخدام الأحداث لتعطى رد فعل فوري للمستخدم. فمثلاً يمكن كتابة إجراء يتحقق من صحة قيمة المدخلات بمجرد انتهاء المستخدم من إدخالها وبالتالي يمكنك استدراك أيخطأ فوراً .

والحدث من وجهة نظر Visual Basicهو كل ما يقوم به المستخدم من حركات للفأرة أو استخدام للوحة المفاتيح، مثلا نعتبر النقر على زر الفأرة حدث، والنقر المزدوج حدث وضغط أى مفتاح من لوحة المفاتيح أيضا حدث.

ولكى تحدث استجابة لأى حدث لابد من وجود أداة ليقع فوقها الحدث،أى أن الحدث نفسه بدون أداة لا يعنى شيئاً، كما أن الأداة نفسها بدون وقوع حدث عليها لا تعنى شيئاً وستبقى مجرد أداة على النموذج .

هناك أحداث لا تصدر مباشرة من المستخدم. فمثلاً المؤقتTimer يقع تلقائيا كل فترة زمنية وليس نتيجة لعمل يقوم به المستخدم.

هل وقع الحدث ؟

هذا السؤال كان على برنامجك أن يسأله، ما لم تكن لغة Visual Basic حاضرة لتزيل عنا عناء إجابة هذا السؤال. لن نحتاج أبدا لشغل أنفسنا بذلك ، فعندما يشعر Windows بحدث ما فإنه يرسل رسالة إلى البرنامج النشط وعندما يستقبل البرنامج الرسالة فإنه يترجمها ثم يفسر أي الأحداث تعبر عنه هذه الرسالة ثم يسلم الرسالة إلى عنصر التحكم المقصود بالرسالة والموجود في واجهة التطبيق.

في Visual Basic كل عنصر تحكم أو نافذة قادرة على التعرف فقط على مجموعة معينه من الأحداث ولكي يتفاعل مع حدث ما يجب كتابة الكود المطلوب تنفيذه في نافذة هذا الحدث. وبالتالي الأحداث التي لا تحتوى على كود يتم تجاهلها .

كيف تتم الاستجابة للحدث؟

قلنا أن الاستجابة للحدث تتطلب وقوع الحدث نفسه (كالنقر مثلا)، والأداة التى وقع فوقها الحدث (كزر الأمر). أما العملية التى تتسبب فى وقوع الحدث فتسمى إجراء أو Procedure. ويخصص Visual Basic إجراء أو Procedure لكل حدث على كل كائن موجود على النموذج.

يبدأ الإجراء عادة بكلمة Sub وينتهى بعبارة End Sub ويظهر داخل نافذة الكود مثل الموجود فى شكل ٦-١ويربط Visual Basic بين الأداة والحدث بعلامة الشرطة التحتية (_) . بالنظرإلى شكل ٦-١تجد أن الأداة هنا هى Button1 ومعناها زر أمر، والحدث هو Click()أى نقر زر الفأرة والربط بينهما بعلامة _ ويتم كتابة الإجراء نفسه وهو عبارة عن تعليمات أو Codes تقوم بكتابتها كما هو واضح من الشكل بين بداية الإجراء ونهايته. وهذه التعليمات التى تكتبها داخل الإجراء (بين سطر Sub وسطرEnd Sub) يقوم Visual Basicباستدعائها وتنفيذها عندما يقع الحدث على الأداة، أى أن Visual

Basicيقوم بإعداد إجراء خالى لا يحتوى على شئ، ويسمح لك أن تكتب فيه التعليمات التى تريد أن ينفذها عند وقوع الحدث .

شكل ٦-١ نافذة الكود وكيفية كتابة الإجراء.

سنشرح كيفية كتابة التعليمات التى توجه الإجراء لتنفيذ مهمة معينة عند حديثنا عن البرمجة باستخدام Visual Basic 2012.

كتابة إجراءات الحدث

يتم إدخال التعليمات التى تريد أن ينفذها Visual Basic عند وقوع الحدث من خلال إجراء أو Procedure ، وتتم كتابة الإجراء/الإجراءات داخل نافذة تسمى نافذة الكود. ويمكن أن تشتمل نافذة الكود على إجراء واحد أو أكثر. تستخدم نافذة الكود لكتابة التعليمات وتعديلها بنفس الطريقة التى تكتب بها داخل أى محرر للكلمات أو منسق للنصوص ويتم استدعائها بمجرد النقر المزدوج على أى كائن موجود على النموذج (الكائن يمكن أن يكون نموذج أو أداة) فى مرحلة التصميم. وفيما يلى سنوضح خطوات استدعاء نافذة الكود وكتابة الإجراءات بداخلها فى مرحلة التصميم عن طريق إعداد برنامج لا يقدم شيئا، ولكنه يوضح فكرة الاستجابة للأحداث وكيفية استخدام نافذة الكود وكتابة الإجراءات. تابع الخطوات التالية :

١. تأكد أنك داخل أحد المشروعات وإلا قم بإنشاء مشروع نوافذى جديد حيث يظهر النموذج Form1 تلقائيا داخل بيئة التطوير Visual Studio 2012.

٢. من مربع الأدوات انقر أداة زر الأمر Button ثم ارسم الأداة على النموذج باستخدام مفهوم السحب والإلقاء (راجع استخدام أدوات التحكم فى الفصل الرابع).

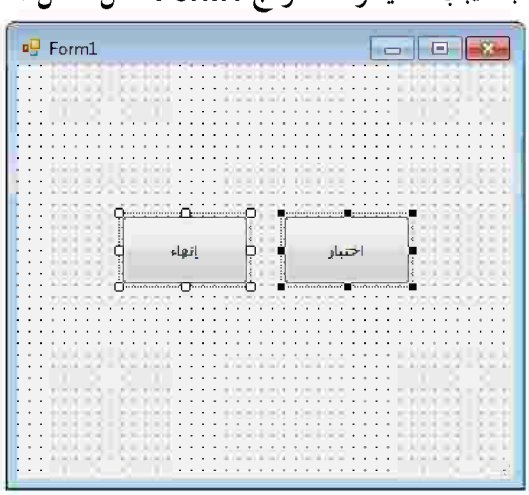
٣. كرر الخطوة السابقة لتوقيع زر أمر آخر على النموذج على يسار الزر السابق.

٤. انقر أداة زر الأمر الأول Button1 لتنشيطها ثم اضغط مفتاح F4 ، يظهر مربع الخصائص وبه خصائص هذه الأداة.

٥. من مربع الخصائص غير العنوان Text من العنوان التلقائى Button1 إلى "اختبار"، والاسم Name من Button1 إلىbtn Test (راجع ضبط الخصائص فى الفصل الخامس).

٦. باتباع الخطوتين رقم ٤ ، ٥ غير عنوان زر الأمر الثاني إلى "إنهاء" والاسم إلى btn Exit.

باتباع الخطوات السابقة يجب أن يكون النموذج Form1 مثل شكل ٦-٢ التالى.



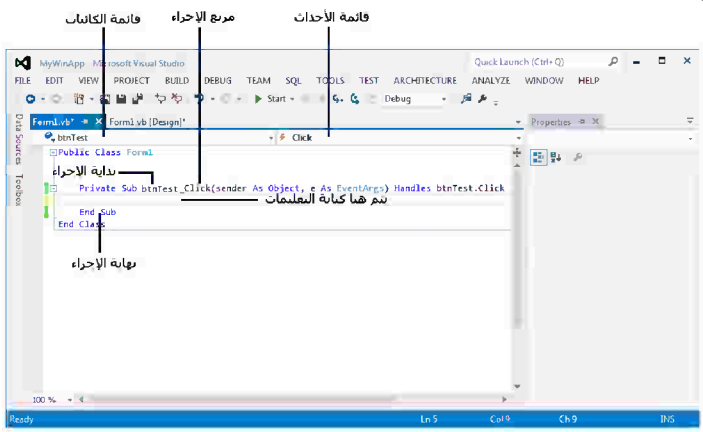
شكل ٦-٢ توقيع الأدوات وضبط خصائصها.

٧. انقر الزر

نقرا مزدوجا،تظهر نافذة الكود كما فى شكل ٦-٣.

تستخدم نافذة الكود لكتابة التعليماتالتى تتسبب فى الاستجابة للحدث وتنفيذ ما تريده عندما تنقر زر الأمر .

تشتمل نافذة الكود بالإضافة إلى شريط العنوان وأشرطة التمرير وأزرار التكبير والتقليص والاسترجاع الموجودة فى نوافذ **Windows** على خانات رئيسية نوضحها فيما يلى:



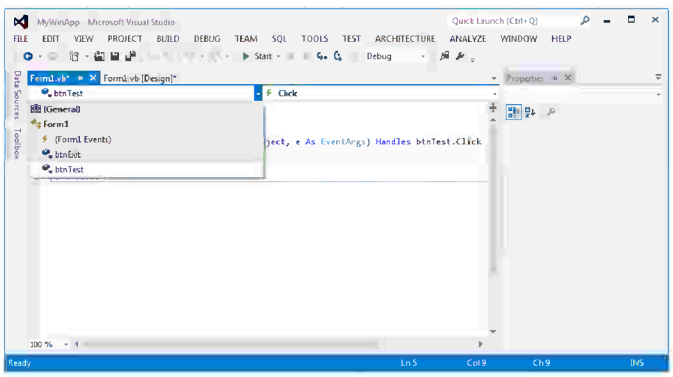
شكل ٦-٣ كتابة الإجراءات داخل نافذة الكود.

قائمة الكائنات

تستخدم هذهالقائمة لعرض الكائنات الموجودة على النموذج بما فيها الأداة النشطة نفسها. وتعرض الكائنات بالأسماء التى اخترتها لكل منها باستخدام خاصية الاسم **Name** فإذا لم تكن غيرت الاسم التلقائى الذى يخصصه **Visual Basic** للأدوات، فستظهر الأدوات بهذا الاسم التلقائى.

للتعرف على قائمة الكائنات وكيفية استعمالها تابع الخطوات التالية:

١. وجه المؤشر إلى السهم الموجود على يمين قائمة الكائنات ثم انقر زر الفأرة. تظهر قائمة منسدلة تشتمل على أسماء جميع الكائنات الموجودة بالنموذج (انظر شكل ٦-٤). وبالطبع فإن محتويات هذه القائمة تختلف حسب الكائنات الموجودة بالنموذج.



شكل ٦–٤ إظهار أسماء الكائنات.

إذا كانت الكائنات الموجودة بالنموذج أكثر من حجم القائمة، فسيظهر لها شريط تمرير رأسي يمكنك استخدامه في استعراض أسماء بقية الكائنات.

٢. من القائمة المنسدلة الموجودة في شكل ٦–٤ انقر الكائن **btn Exit**، يتغير اسم الكائن في قائمة الكائنات وفي اسم الإجراء إلى الكائن الذي اخترته. وهكذا يمكنك اختيار أي كائن موجود بالنموذج من قائمة الكائنات وكتابة أو تعديل الإجراء الخاص به.

٣. انقر زر السهم مرةً أخرى ثم اختر الأداة **btn Test**.

قائمة الأحداث

تستخدم هذه القائمة لعرض الأحداث الممكنة للأداة النشطة. وبالطبع فان محتويات هذه القائمة تختلف حسب الأحداث التي يمكن أن تقع للأداة النشطة. وتستخدم قائمة الأحداث أيضا في تغيير الحدث الذي ترغب في الاستجابة له، انقر السهم الموجود على يمين القائمة. تظهر قائمة بالأحداث الممكنة. إذا غيرت حدث ، سيتغير اسم الحدث الموجود في القائمة وفي اسم الإجراء.

إذا كانت الأحداث الممكنة أكثر من حجم القائمة، فسيظهر لها شريط تمرير رأسي يمكنك استخدامه في استعراض بقية الأحداث.

مربع الإجراء

هو المساحة المتاحة داخل نافذة الكود والتى تستخدم لكتابة الإجراء. بمجرد فتح نافذة الكود لكتابة حدث الزر، يظهر فى مربع الإجراء الكود التالى:

```
Private Sub btn Test _Click(sender As Object, e As Event Args )
Handles btn Test .Click

End Sub
```

فأول سطرين فى الواقع عبارة عن سطر واحد يبدأ بكلمة **Private** ككلمة وصول ثم بعد ذلك كلمة **Sub** متبوعةً باسم الإجراء الذى يتكون عادةً من اسم الأداة مضافاً إليها نوع الحدث وبينهما العلامة "_" وبعد ذلك تأتى معاملات الإجراء ولسنا فى حاجة إليها الآن فى هذا المستوى من الدراسة وأخيراً كلمة **Handles** متبوعةً باسم الأداة والحدث مفصولتان بالنقطة ".". أما السطر الأخير فيحتوى على العبارة **End Sub** التى تعلن عن انتهاء الإجراء ومن ثمَّ يمكنك كتابة تعليمات الإجراء بين السطرين. يمكنك الكتابة بطريقة مشابهة للكتابة فى برامج تحرير أو تنسيق النصوص العادية. ويمكن كتابة أكثر من إجراء واحد داخل مربع الإجراءات.

لكتابة التعليمات الخاصة بطباعة عبارة معينة عندما تنقر زر الأمر "اختبار" أو الخروج من البرنامج عندما تنقر زر الأمر "إنهاء" اتبع الآتى :

١. لفتح نافذة الكود، انقر الزر

من النموذج نقرا مزدوجابحيث يظهر الإجراء داخل نافذة الكود مثل شكل ٦-٣ السابق.

٢. اكتب الأمر التالي بينسطرى الإجراء:

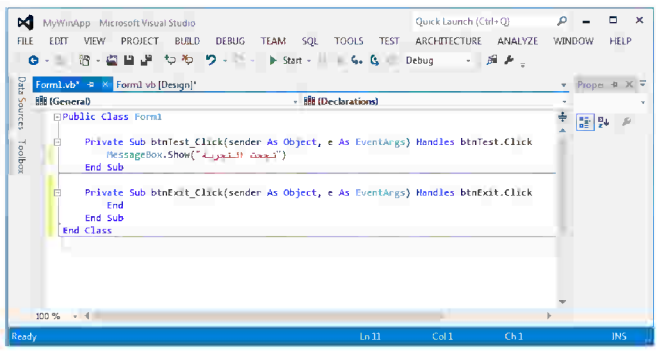
```
Message Box .Show ("نجحت التجربة")
```

٣. من قائمة الكائنات انقر زر السهم ومن القائمة المنسدلة انقر الكائن **Btn Exit** يتغير اسم الكائن فى قائمة الكائنات إلى **Btn Exit** ثم اختر **Click** من قائمة الأحداث.

٤. اكتب الأمر التالي:

```
End
```

تصبح محتويات النافذة بهذا الشكل (انظر شكل ٦-٥).



شكل ٦-٥ نافذة الكود بعد كتابة إجراء الحدثين.

٥. اضغط مفتاح **F5** لتشغيل البرنامج (أو انقر الزر ▶ من شريط الأدوات) يبدأ البرنامج فى العمل وتصبح أزرار الأوامر جاهزة للعمل الآن.

٦. انقر الزر [اختبار]، تظهر عبارة " نجحت التجربة" داخل مربع رسالة.يمكنك أن تنقر عدة مرات وفى كل مرة يقع الحدث (وهو هنا النقر على زر الأمر) حيث يقوم **Visual Basic** بتنفيذ الإجراء الخاص بهذا الحدث (وهو هنا **btn Test _click()**). أماإذا لم تكتب إجراء (تعليمات) للأداة الموجودة على النموذج فان هذه الأداة لن تستجيب لأى حدث وستصبح مجرد ديكور على النموذج .

٧. انقر زر [انهاء]،يقوم **Visual Basic** بإنهاء البرنامج استجابةً لأمر **End** الموجود بالإجراء الخاص بهذا الزر وترجع مرةً أخرى إلى واجهة التصميم.

فهم تسلسل الأحداث

حتى الآن تعرفنا على الأحداث وكيف يتعامل معها**Visual Basic 2012** ، وكذلك عرفنا كيف نكتب كود لإجراء حدثي. ولكن لكتابة إجراءات حدثية جيدة لابد من فهم أعمق لطبيعة الأحداث المتسلسلة والتي تحدث معاً، ففي حين يفتح ذلك أمامك باب أوسع لاختيار الحدث الأكثر تناسباً ومنطقيةًلإثارة الكود وتشغيله، إلا أنه في نفس الوقت يفتح أبوابا من المشاكل غير المتوقعة، فعلى عكس ما تتوقع قد يسلك البرنامج سلوك غريب، فضلاً عن احتمال دخول البرنامج في دوارة لا نهائية ويسبب عطل للنظام، من أسباب ذلك

على سبيل المثال أن تستخدم الحدث Change لتغيير محتويات نص المربع الذي أطلقه حيث يؤدي ذلك إلى إعادة إطلاق الحدث مرة أخرى وهكذا دواليك.

هناك خطوط عامة يجب اتباعها لتجنب هذا النوع من المشاكل وهي كالتالي:

- أن تتبين بداية أن عدة أحداث ستقع معا
- أن تحدد أي من الأحداث يقع تبعا لفعل معين
- أن تتفهم تسلسل وتعاقب الأحداث
- أن تختبر رد فعل البرنامج تجاه تصرفات عديدة وغير نمطية

الأحداث المتسلسلة (ذات السبب الواحد)

هذا النوع من الأحداث أحد مشاكل البرمجة المسيرة بالأحداث، كما أنه فى نفس الوقت أحد مصادر قوتها. ينبع مصدر صعوبة هذه البرمجة أيضاً من أن أي حالة يكون عليها البرنامج يمكن الوصول إليها من خلال عدد لا نهائي من الأحداث المتعاقبة، مثلا لو أن التركيز في أحد مربعات النصوص، هل كانت قبل ذلك في مربع آخر أم على زر أمر، عند تبديل حالة مربع اختيار هل تم ذلك بالفأرة أم بمفتاح المسافة على لوحة المفاتيح.. وهكذا مما قد يخلق عدد لا نهائي من الحالات المطلوب معالجتها.

الطريقة السليمة للتعاملمع هذا النوع ما يلي:

الخطوة الأولى: لا تقم بمعالجة أي حدث لا يهمك، سيقلل ذلك من احتمالات المشاكل.

الخطوة الثانية: أن تفهم تماما تسلسل الأحداث، بالطبع لن نستطيع مهما كان حجم الكتاب أن نغطي كافة الأحداث وعلاقاتها التي تصل إلى المئات.

أنواع الأحداث

يوجد نوعان رئيسيان من الأحداث: أحداث يسببها المستخدم و الأخرى يسببها النظام، وفى اغلب الأحوال يهمنا معالجة النوع الأول فقط والتي تنتج من أوامر المستخدم للبرنامج، لأن إثارة الأحداث هو طريق المستخدم للتحكم في البرنامج.

الأحداث التي يثيرها المستخدم

وهى التي تنتج عن فعل المستخدم مثل ضغط مفتاح أو نقر زر الفأرة ويمكن أن يثير المستخدم أحداثا أخرى بطريق غير مباشر. فمثلاً عند النقر على مربع نص لتعديله فإن الحدث **Click** يثار ولكن ما قد لا تلاحظه هو أن هناك أحداث أخرى تثار معه منها الحدث **Got Focus** وهذا الحدث سببه انتقال التركيز إلى عنصر التحكم، أيضا يثار حدث آخر وهو **Lost Focus** يكون ذلك بالنسبة لعنصر التحكم الذي غادرناه إلى مربع النص. وكما سترى لاحقا فإن عدة أحداث يمكن أن تقع نتيجة فعل واحد من المستخدم وعليك فهم ترتيب وقوعها ليمكنك معالجتها بكفاءة. سنتعرف فيما يلي على بعض أفعال المستخدم التي تثير أحداثا في البرنامج.

الأحداث الشائعة

كما أن هناك خصائص شائعة موجودة فى معظم الأدوات، هناك أيضاً أحداث شائعة تستجيب لها معظم الأدوات. فى حين توجد أحداثا خاصة لا تشعر بها إلا أدوات معينة . يوضح جدول ٦-١التالى باختصار أهم الأحداث الشائعة ووقت حدوثها.

لتتأكد أن أداة معينة تستجيب لحدث أم لا، انقر نقرا مزدوجا فوق هذه الأداة لإظهار نافذة الكود ثم ابحث عن الحدث فى مربع الأحداث.

جدول ٦-١ الأحداث الشائعة المستخدمة مع معظم أدوات التحكم

الحدث	معناه	وقت حدوثه
أولاً: أحداث ناتجة عن حركات الفأرة		
Click	النقر	يقع حينما يقوم المستخدم بالنقر فوق الأداة أو الواجهة، أو إذا ضغط المستخدم مسطرة المسافات من لوحة المفاتيح بعد تنشيط زر الأمر
DoubleClick	النقر المزدوج	يقع حينما يقوم المستخدم بالنقر المزدوج فوق النموذج أو الأداة

الحدث	معناه	وقت حدوثه
Mouse Move	حركة الفأرة	يقع حينما يحرك المستخدم الفأرة فوق الواجهة أو الأداة
Mouse Down	زر الفأرة لأسفل	يقع أثناء عملية النقر أى عندما ينقر المستخدم زر الفأرة وقبل أن يحرره
Mouse Up	زر الفأرة لأعلى	يقع بعد عملية النقر
ثانياً: أحداث السحب والإلقاء		
Drag Over	سحب فوق	يقع حينما يمر المستخدم فوق أداة أخرى أثناء سحب الأداة
Drag Drop	إلقاء	يقع عندما تستقر الأداة فى المكان المطلوب فى نهاية عملية السحب
ثالثاً: أحداث ناتجة عن لوحة المفاتيح		
Key Press	ضغط حرف	يقع عندما يضغط المستخدم أحد الحروف الأبجدية من لوحة المفاتيح
Key Down	المفتاح لأسفل	يقع أثناء ضغط المفتاح من لوحة المفاتيح وقبل تحريره
Key Up	المفتاح لأعلى	يقع بعد ضغط المفتاح