

الفصل العاشر

الأحداث Events

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

بالانتهاء من هذا الفصل ستكتسب المعارف وتندرب على المهارات التي تجعلك قادراً 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 أيضاً يمكنك إطلاق البوق، ويمكنك تغيير 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 دائماً أن يكون برنامجك طبيعي ومشابه للحياة الواقعية، بمعنى توقع من المستخدم أن يقوم بتصرفات تشبه سلوكه مع الأشياء الحقيقية كالمذياع والهاتف، وخلافها. مثلاً 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 وهي 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 أي خطأ فوراً .

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

ولكى تحدث استجابة لأي 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 المؤقت Timer يقع تلقائياً 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 أبداً 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 معينة 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 التي وقع فوقها الحدث (كزر الأمر). أما العملية التي تتسبب في 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 وينتهي بعبارة End Sub ويظهر داخل نافذة الكود مثل الموجود في شكل ١٠-١ ويربط Visual Basic بين الأداة والحدث بعلامة الشرطة التحتية (_). بالنظر إلى شكل ١٠-١ تجد أن الأداة هنا هي 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 التي تكتبها داخل الإجراء (بين سطر Sub وسطر End Sub) يقوم Visual Basic باستدعائها وتنفيذها عندما يقع الحدث على الأداة، أى أن Visual Basic يقوم بإعداد إجراء خالى لا يحتوى على شئ، ويسمح لك أن تكتب فيه التعليمات التي تريد أن ينفذها عند وقوع الحدث .



شكل ١٠-١ نافذة الكود وكيفية كتابة الإجراء.

سنشرح كيفية كتابة التعليمات التي توجه الإجراء لتنفيذ مهمة معينة عند حديثنا عن البرمجة باستخدام Visual Basic 2008.

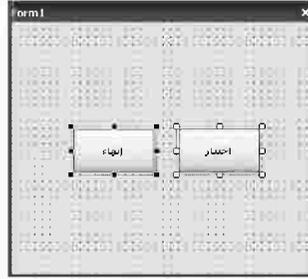


كتابة إجراءات البرمجة

يتم إدخال التعليمات التي تريد أن ينفذها Visual Basic عند وقوع الحدث من خلال إجراء أو Procedure ، وتتم كتابة الإجراء/الإجراءات داخل نافذة تسمى نافذة الكود. ويمكن أن تشمل نافذة الكود على إجراء واحد أو أكثر. تستخدم نافذة الكود لكتابة التعليمات وتعديلها بنفس الطريقة التي 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 يلي سنوضح خطوات استدعاء نافذة الكود وكتابة الإجراءات بداخلها فى مرحلة التصميم عن طريق إعداد برنامج لا يقدم شيئاً، ولكنه يوضح فكرة الاستجابة للأحداث وكيفية استخدام نافذة الكود وكتابة الإجراءات. تابع الخطوات التالية :

١. تأكد أنك داخل أحد المشروعات وإلا قم بإنشاء مشروع نوافذ جديد حيث يظهر النموذج Form1 تلقائياً داخل بيئة التطوير Visual Studio 2008.
٢. من مربع الأدوات انقر أداة زر الأمر  ثم ارسم الأداة على النموذج باستخدام مفهوم السحب والإلقاء (راجع استخدام أدوات التحكم فى الفصل السابع).
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 إلى btnTest (راجع ضبط الخصائص فى الفصل الثامن).
٦. باتباع الخطوتين رقم ٤ ، ٥ غير عنوان زر الأمر الثانى إلى "إنهاء" والاسم إلى btnExit.

باتباع الخطوات السابقة يجب أن يكون النموذج Form1 مثل شكل ١٠-٢ التالى.



شكل ١٠-٢ توقيع الأدوات وضبط خصائصها.

٧. انقر الزر الختبار نقرا مزدوجا، تظهر نافذة الكود كما في شكل ١٠-٣. تستخدم نافذة الكود لكتابة التعليمات التي تتسبب في الاستجابة للحدث وتنفيذ ما تريده عندما تنقر زر الأمر. تشمل نافذة الكود بالإضافة إلى شريط العنوان وأشرطة التمرير وأزرار التكبير والتقليص والاسترجاع الموجودة في نوافذ Windows على خانة رئيسية نوضحها فيما يلي:



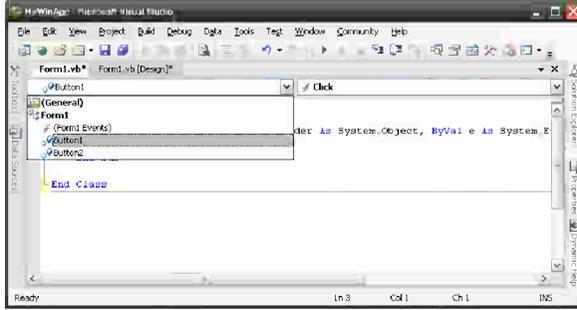
شكل ١٠-٣ كتابة الإجراءات داخل نافذة الكود.

قائمة الكائنات

تستخدم هذه القائمة لعرض الكائنات الموجودة على النموذج بما فيها الأداة النشطة نفسها. وتعرض الكائنات بالأسماء التي اخترتها لكل منها باستخدام خاصية الاسم **Name** فإذا لم تكن غيرت الاسم التلقائي الذي يخصصه **Visual Basic** للأدوات، فستظهر الأدوات بهذا الاسم التلقائي.

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 ١٠-٤). وبالطبع فإن محتويات هذه القائمة تختلف حسب الكائنات الموجودة بالنموذج.



شكل ١٠-٤ إظهار أسماء الكائنات.

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



٢. من القائمة المنسدلة الموجودة فى شكل ١٠-٤ انقر الكائن **BtnExit** ، يتغير اسم الكائن فى قائمة الكائنات وفى اسم الإجراء إلى الكائن الذى اخترته. وهكذا يمكنك اختيار أى كائن موجود بالنموذج من قائمة الكائنات وكتابة أو تعديل الإجراء الخاص به.

٣. انقر زر السهم مرةً أخرى ثم اختر الأداة **BtnTest** .

قائمة الأحداث

تستخدم هذه القائمة لعرض الأحداث الممكنة للأداة النشطة. وبالطبع فإن محتويات هذه القائمة تختلف حسب الأحداث التى يمكن أن تقع للأداة النشطة. وتستخدم قائمة الأحداث أيضا فى تغيير الحدث الذى ترغب فى الاستجابة له، انقر السهم الموجود على يمين القائمة. تظهر قائمة بالأحداث الممكنة. إذا غيرت حدث ، سيتغير اسم الحدث الموجود فى القائمة وفى 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 والتي تستخدم لكتابة الإجراء. بمجرد فتح نافذة الكود لكتابة حدث الزر، يظهر في مربع الإجراء الكود التالي:

```
Private Sub btnTest_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnTest.Click
```

End Sub

فأول سطرين في الواقع عبارة عن سطر واحد يبدأ بكلمة **Private** ككلمة وصول ثم بعد ذلك كلمة **Sub** متبوعةً باسم الإجراء الذي يتكون عادةً من اسم الأداة مضافاً إليها نوع الحدث وبينهما العلامة **_** وبعد ذلك تأتي معاملات الإجراء ولسنا في حاجة إليها الآن في هذا المستوى من الدراسة وأخيراً كلمة **Handles** متبوعةً باسم الأداة والحدث مفصولتان بالنقطة **."** أما السطر الأخير فيحتوي على العبارة **End Sub** التي تعلن عن انتهاء الإجراء ومن ثمّ يمكنك كتابة تعليمات الإجراء بين السطرين. يمكنك الكتابة بطريقة مشابهة للكتابة في 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 الآتي:

١. لفتح نافذة الكود، انقر الزر من النموذج نقرأ مزدوجاً بحيث يظهر الإجراء داخل نافذة الكود مثل شكل ١٠-٣ السابق.
٢. اكتب الأمر التالي بين سطري الإجراء:

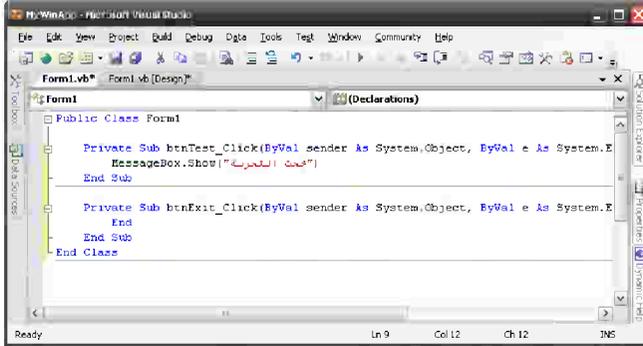
```
MessageBox.Show("نجحت التجربة")
```

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

٤. اكتب الأمر التالي:

End

تصبح محتويات النافذة بهذا الشكل (انظر شكل ١٠-٥).



شكل ١٠-٥ نافذة الكود بعد كتابة إجراء التحديث.

٥. اضغط مفتاح F5 لتشغيل البرنامج (أو انقر الرمز  من شريط الأدوات)

يبدأ البرنامج في العمل وتصبح أزرار الأوامر جاهزة للعمل الآن.

٦. انقر الزر , تظهر عبارة "نجحت التجربة" داخل مربع رسالة. يمكنك

أن تنقر عدة مرات وفي كل مرة يقع الحدث (وهو هنا النقر على زر الأمر) حيث

يقوم Visual Basic بتنفيذ الإجراء الخاص بهذا الحدث (وهو هنا

BtnTest_click()). أما إذا لم تكتب إجراء (تعليمات) للأداة الموجودة على

النموذج فإن هذه الأداة لن تستجيب لأي حدث وستصبح مجرد ديكور على

النموذج .

٧. انقر زر , 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 2008 ، وكذلك

عرفنا كيف نكتب كود لإجراء حدثي. ولكن لكتابة إجراءات حديثة جيدة لا بد من فهم

أعمق لطبيعة الأحداث المتسلسلة والتي تحدث معاً، ففي حين يفتح ذلك أمامك باب

أوسع لاختيار الحدث الأكثر تناسباً ومنطقيةً لإثارة الكود وتشغيله، إلا أنه في نفس الوقت يفتح أبواباً 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 تبعاً لفعل معين
- أن تفهم 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 في 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، مثلاً 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 للتعامل مع هذا النوع ما يلي:

الخطوة الأولى: لا تقم بمعالجة أي حدث لا يهمك، سيقبل ذلك من احتمالات المشاكل.
الخطوة الثانية: أن تفهم تماماً 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، وفي أغلب الأحوال يهمننا معالجة النوع الأول فقط والتي تنتج من أوامر المستخدم للبرنامج، لأن إثارة الأحداث هو طريق المستخدم للتحكم في البرنامج.

الأحداث التي يثيرها المستخدم

وهي التي تنتج عن فعل المستخدم مثل ضغط مفتاح أو نقر زر الفأرة ويمكن أن يثير المستخدم أحداثاً أخرى بطريق غير مباشر. فمثلاً عند النقر على مربع نص لتعديله فإن الحدث **Click** يثار ولكن ما قد لا تلاحظه هو أن هناك أحداثاً أخرى تثار معه منها الحدث **GotFocus** وهذا الحدث سببه انتقال التركيز إلى عنصر التحكم، أيضاً يثار حدث آخر وهو **LostFocus** يكون ذلك بالنسبة لعنصر التحكم الذي غادرناه إلى مربع النص.

وكما ستري لاحقاً 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 أحداثاً في البرنامج.

الأحداث الشائعة

كما أن هناك خصائص شائعة موجودة في معظم الأدوات، هناك أيضاً أحداث شائعة تستجيب لها معظم الأدوات. في حين توجد أحداثاً خاصة لا تشعر بها إلا أدوات معينة. يوضح جدول ١٠-١ التالي باختصار أهم الأحداث الشائعة ووقت حدوثها.

لتأكد أن أداة معينة تستجيب لحدث أم لا، انقر نقراً مزدوجاً فوق هذه الأداة لإظهار نافذة الكود ثم ابحث عن الحدث في مربع الأحداث.



جدول ١٠-١ الأحداث الشائعة المستخدمة مع معظم أدوات التحكم

الحدث	معناه	وقت حدوثه
<i>أولاً: أحداث ناتجة عن حركات الفأرة</i>		
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

الحدث	معناه	وقت حدوثه
		المزدوج فوق النموذج أو الأداة
MouseMove	حركة الفأرة	يقع حينما يحرك المستخدم الفأرة فوق الواجهة أو الأداة
MouseDown	زر الفأرة لأسفل	يقع أثناء عملية النقر أى عندما ينقر المستخدم زر الفأرة وقبل أن يحرره
MouseUp	زر الفأرة لأعلى	يقع بعد عملية النقر
<i>ثانياً: أحداث السحب والإلقاء</i>		
DragOver	سحب فوق	يقع حينما يمر المستخدم فوق أداة أخرى أثناء سحب الأداة
DragDrop	إلقاء	يقع عندما تستقر الأداة في المكان المطلوب في نهاية عملية السحب
<i>ثالثاً: أحداث ناتجة عن لوحة المفاتيح</i>		
KeyPress	ضغط حرف	يقع عندما يضغط المستخدم أحد الحروف الأبجدية من لوحة المفاتيح
KeyDown	المفتاح لأسفل	يقع أثناء ضغط المفتاح من لوحة المفاتيح وقبل تحريره
KeyUp	المفتاح لأعلى	يقع بعد ضغط المفتاح