الفصل الخامس والعشرون
مقدمة إلى لغة Access VBA

رغم أنه بإمكانك تطوير نظم كاملة لقواعد البيانات بدون كتابة أمر واحد وبدون استخدام لغة Access VBA ،إلا أن هناك أسباباً تدعوك لاستخدام Access VBA .

بانتهاء هذا الفصل ستتعرف على :

- التفرقة بين الماكرو ولغة SQL ولغة Access VBA .
- تقرير متى تستخدم Access VBA .
- كتابة برنامج بسيط باستخدام Access VBA وتنفيذه .

في معظم التطبيقات البسيطة لن تحتاج إلى لغة برمجة ولا لمعرفة لغة VBA، إلا أن التطبيقات المعقدة تحتاج إلى كتابة برامج أو إجراءات صغيرة. لأداء بعض الوظائف التى يصعب على الماكرو تنفيذها. وفي هذا الفصل سنلقي نظرة عابرة على البرامج أو الإجراءات أو الوحدات النمطية ، نتعرف منها على فكرة عمل الوحدة النمطية من خلال برنامج بسيط جداً يوضح خطوات التعامل مع الوحدة النمطية.

لغة Visual Basic للتطبيقات Access VBA

VBA اختصار لعبارة Visual Basic For Applications ويقصد بها لغة البرمجة التي يستخدمها مبرمجو ومطورو Access لكتابة البرامج أو التطبيقات المطلوبة لتطوير تطبيقاتهم. وقد ترجمتها شركة مايكروسوفت هكذا "Visual Basic للتطبيقات". سنستخدم فيما يلي عبارة VBA أو Access VBA للإشارة إلى لغة Visual Basic For Applications التي تستخدمها Access 2007 .

وكان "بيل جيتس" صاحب شركة مايكروسوفت قد طالب منذ عام ١٩٨٩ بتطوير لغة قياسية واحدة يمكن استخدامها في البرمجة مع جميع البرامج التي تنتجها الشركة مثل مجموعة برامج Microsoft Office أو Visual Basic، وقد تم ذلك بالفعل في عام ١٩٩٥. لذلك فإن لغة VBA لغة عامة تستخدمها جميع البرامج التي تعمل تحت بيئة Windows مثل Excel أو Word أو Project. ومع ذلك فإن كل برنامج من هذه البرامج أضاف للغة VBA سمات خاصة به. وهذا هو السبب أننا نقول Access VBA وليس VBA فقط للإشارة إلى اللغة التي تستخدمها Access.

مدخل إلى Access VBA

تعتبر VBA هى لغة البرمجة الحقيقية وليست لغة ماكرو ، ويمكنك إنشاء مجموعة من الماكرو القوية من خلال استخدام دوال VBA والإجراءات الفرعية، ورغم أنه بإمكانك تطوير نظم كاملة لقواعد البيانات بدون كتابة أمر واحد وبدون استخدام لغة Access

VBA، إذ يكفى استخدام الدوال والماكرو لتنفيذ مهام وإجراءات كثيرة، رغم كل ذلك فهناك أسباب تدعوك لاستخدام Access VBA منها:

- إنشاء وظائف خاصة User Defined Functions لأداء المهام التى يتكرر استخدامها.
- استخدام تعبيرات تشتمل على دوارات Loops.
- فتح أكثر من قاعدة بيانات فى النظام.
- متابعة الأخطاء التى تحدث وتقرير ما يجب عمله إذا حدثت.
- فى التطبيقات الكبيرة التى تتطلب التعامل مع Application Program (API) Interface وهو عبارة عن مكتبة من الوظائف موجودة فى نظام Windows.

تشبه لغة Access VBA لغة Visual Basic ولذلك فإذا كانت لك خبرة بلغة Visual Basic فسيسهل عليك تعلم لغة Access VBA.

الهدف من هذا الفصل هو إعطاءك فكرة عن لغة VBA للحصول على تفاصيل عن كيفية البرمجة باستخدام VBA أو للتعرف على الدوال والإجراءات بالتفصيل، ننصحك بمراجعة كتابنا "البرمجة المتقدمة باستخدام Access 2010"

الوحدات النمطية والدوال والإجراءات الجزئية

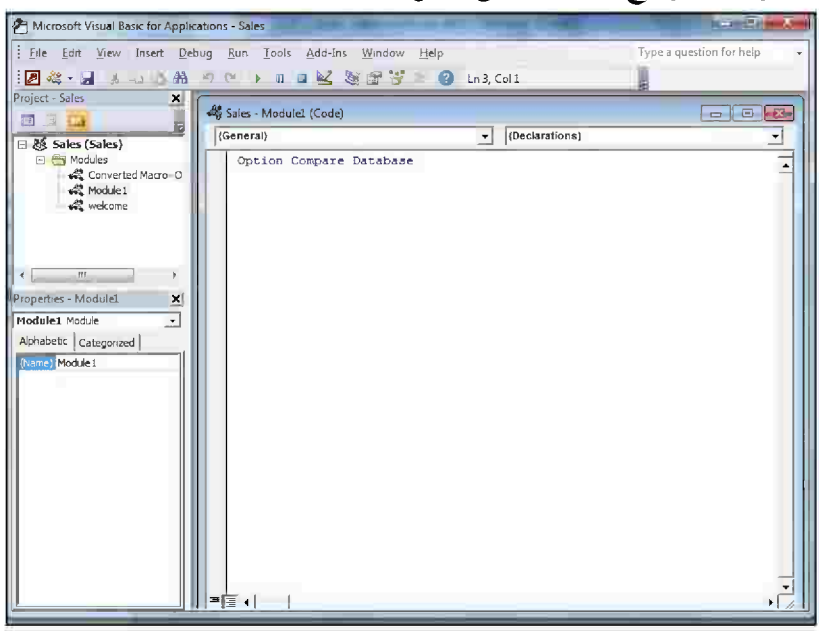
تعرفنا فى الفصل السابق على كيفية إنشاء الماكرو، وسنتعرف الآن على الوحدة النمطية وترجمتها Module كما وردت فى البرنامج وهى الكلمة المألوفة للمبرمجين الذين يتعاملون مع لغات البرمجة. وتسمى أيضاً فى وسط المبرمجين برنامج.

ولذلك يمكننا تعريف الوحدة النمطية على أنها برنامج يشتمل على مجموعة من التعريفات والإجراءات والعبارات المخزنة مع بعضها البعض كوحدة واحدة فى Access VBA.

وتعرض الوحدات النمطية سواءً الجديدة أو الموجودة من قبل فى إطار طريقة عرض التصميم فقط داخل نافذة الوحدة النمطية.

اذن الوحدات النمطية عبارة عن حاوية نضع فيها كود VBA ، توفر Access أربع أنواع للوحدات النمطية نوجزها فيما يلى:

- ***Access Modules*** *الوحدات النمطية لـ* ***Access***: يمكنك إنشاء وحدات نمطية فى Access تشتمل على كود VBA بنفس الطريقة التى تقوم فيها بإنشاء أى كائن جديد من كائنات قاعدة البيانات، نشط التبويب Create "إنشاء" ومن مجموعة Macro&Code "وحدات ماكرو ورمز" انقر Module "وحدة نمطية" تظهر نافذة برنامج VBA كما فى شكل ٢٥-١.



شكل ٢٥-١ نافذة محرر VBA

- ***Form Modules*** *الوحدات النمطية للنماذج*: تشتمل الوحدات النمطية للنماذج على كود لمعالجة الإجراءات الموجودة فى أدوات تحكم النموذج. وبشكل أساسى عندما تقوم بإضافة كود لأى كائن فى النموذج فإنك تقوم بإنشاء تصنيف لهذا الكائن فى قاعدة البيانات. يمكنك فتح الوحدات النمطية للنموذج من خلال نقر التبويب

Design "تصميم" ومن مجموعة Tools "أدوات" انقر زر View Code "عرض التعليمات البرمجية".

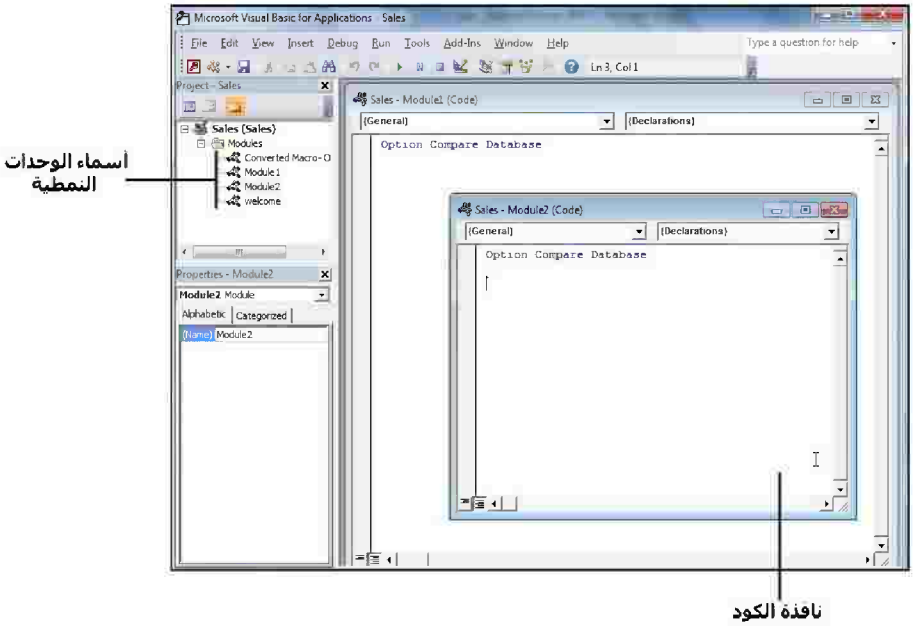
- ***Report Modules*** *وحدات نمطية للتقارير*: وهى تحتوى على كود لمعالجة الأحداث الموجودة فى أدوات التقرير الموجودة فى رأس وتذييل النموذج أو فى باقى أجزاؤه، ويمكنك فتح وحدات النمطية للنموذج من خلال نقر زر View Code "عرض التعليمات البرمجية" الموجود فى التبويب Design "تصميم".
- ***Class Modules*** *وحدات تصنيف*: هى وحدات غير مرتبطة بالنموذج أو التقرير وتسمح لك بإعطاء تعريف مخصص للكائنات وخصائصها وطرقها.

فتح نافذة الوحدة النمطية

لفتح نافذة الوحدة النمطية تابع الخطوات التالية:

١. افتح قاعدة بيانات Access من عندك.

٢. من نافذة قاعدة البيانات نشط التبويب Create "إنشاء" ومن مجموعة Macro&Code "وحدات ماكرو ورمز" انقر زر Module "وحدة نمطية"، تظهر نافذة جديدة لوحدة نمطية (انظر شكل ٢٥-٢).



شكل ٢٥-٢ بيئة التطوير المتكاملة الجديدة المستخدمة لكتابة كود VBA .

ويلاحظ من الشكل أن بيئة التطوير المتكاملة تشبه بيئة تطوير Visual Basic إلى حد كبير.

شرح نافذة الوحدة النمطية

تختلف نافذة الوحدة النمطية عن بقية النوافذ التى تعودت عليها فى Access، ولذلك سنتوقف قليلا لشرح هذه النافذة .

نافذة الوحدة النمطية عبارة عن محرر نصوص تم تصميمه خصيصا لتسهيل عملية إدخال Access VBA وتعديله وتعرض فيه العبارات والوظائف والماكرو باللغة الإنجليزية من اليسار إلى اليمين ، ومع ذلك يمكن تسمية الوحدة النمطية باسم عربى أو مختلط يشتمل شكل ٢٥-٣ على وحدة نمطية داخل نافذة الوحدة النمطية.

شكل ٢٥-٣ شرح مبسط لنافذة الوحدة النمطية

دقق النظر فى شكل ٢٥-٣ ستجد أن النافذة تشتمل على :

- *شريط القوائم* **Menu bar**: ويشتمل على مجموعة من الأوامر المستخدمة لأداء مهام معينة خاصة بلغة **Visual Basic** .
- *شريط الأدوات* **Tool bar**: يشتمل على أزرار لتشغيل إجراءات البرنامج أو توقيفها أو تشغيلها أو الحصول على تعليمات مساعدة. وجه المؤشر إلى كل زر على حده ثم اترك المؤشر لمدة ثوانى لتتعرف على وظيفة الأزرار الموجودة بشريط الأدوات.
- *نافذة* **Code** *"الكود"* : والتى يتم من خلالها كتابة الكود المطلوب تنفيذه وهى النافذة الأساسية الموجودة فى بيئة التطوير المتكاملة.
- *نافذة* **Project** *"المشروع"*: وتحتوى على الوحدات النمطية الموجودة بقاعدة البيانات الحالية.

- *نافذة* ***Properties*** "الخصائص": وتحتوى على خصائص الوحدة النمطية الحالية.

كتابة برنامج بسيط

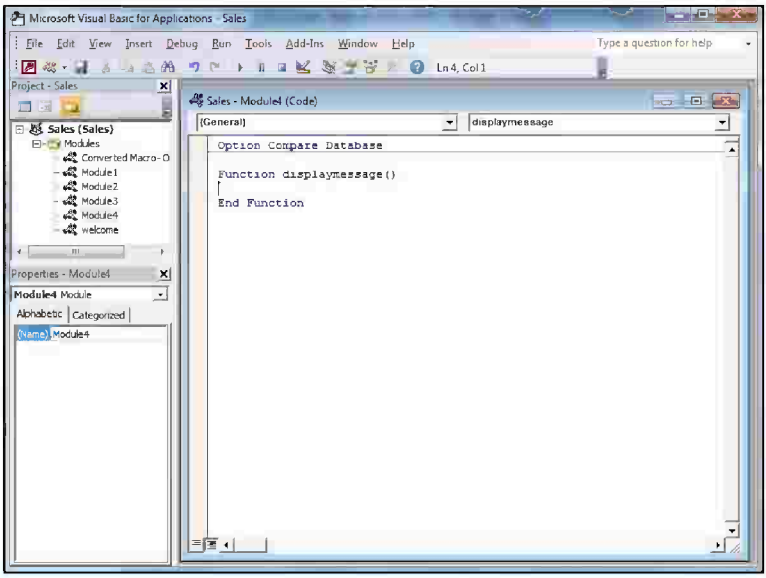
فيما يلى سنوضح خطوات كتابة مثال بسيط يشتمل على دالة بسيطة لتوضيح فكرة كتابة الوحدة النمطية وتشغيلها.

١. من موقع نقطة الإدراج داخل نافذة Code اكتب :

```
Function Displaymessage ( )
```

٢. اضغط مفتاح الإدخال.

بمجرد ضغط مفتاح الإدخال ستفهم Access أنك تريد تعريف دالة Displaymessage ولذلك سينتقل المؤشر تلقائياً إلى منطقة كتابة الدالة، وستكتب لك أيضاً سطر End Function فى الوقت الذى ستسمح لك فيه بمتابعة إكمال الدالة الجديدة (انظر شكل ٢٥-٤) .



شكل ٢٥-٤ كتابة دالة داخل الوحدة النمطية .

٣. يظهر الآن فوق النافذة اسم الدالة الجديدة.

٤. انقر السهم الموجود على يمين اسم الدالة، تظهر قائمة منسدلة تشتمل على خيارين هما **Declarations** و **Displaymessage**. معنى هذا أن الوحدة النمطية (أو البرنامج) يشتمل الآن على هذين القسمين.

لا تشغل بالك الآن بمحتويات قسم **Declarations** وهو القسم الذى يظهر تلقائياً بمجرد فتح نافذة الوحدة النمطية. يكفى أن تعرف أن هذا الإجراء يظهر دائماً في بداية كل برنامج، وأنه هو الإجراء الوحيد الذى تسميه **Access** نيابةً عنك أما بقية الإجراءات فتقوم أنت بتسميتها داخل البرنامج.

٥. انقر **Declarations**، تنتقل إلى قسم **Declarations** داخل الوحدة النمطية (البرنامج).

٦. انقر السهم مرة ثانية ثم اختر **Displaymessage** . سترجع مرة أخرى إلى قسم الدالة .

٧. من موقع نقطة الإدراج اكتب:

Msgbox "مرحباً بك مع كمبيوساينس"

٨. هذه الدالة تطلب إظهار رسالة " مرحباً بك مع كمبيوساينس"

٩. افتح قائمة **File** ثم اختر **Save** وعندما يظهر مربع حوار **Save** انقر **Yes** ومن مربع **Save As** الذى يظهر بعد ذلك اكتب **Welcome** لحفظ الوحدة النمطية بهذا الاسم. ثم اغلق نافذة الوحدة النمطية .

١٠. تغلق نافذة الوحدة النمطية وترجع إلى نافذة قاعدة البيانات. ستجد اسم الوحدة النمطية تحت وحدات نمطية .

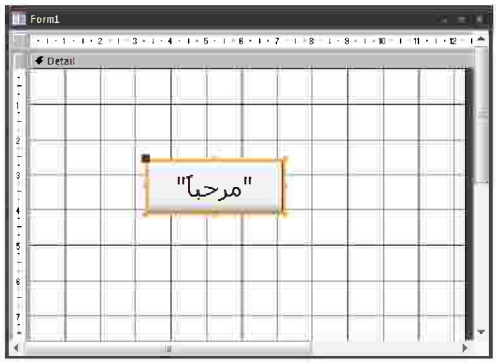
تشغيل البرنامج (الوحدة النمطية)

البرنامج البسيط الذى كتبناه عبارة عن دالة صغيرة لإظهار رسالة، لتنفيذ هذا البرنامج يجب استدعائه من داخل نموذج وذلك باتباع الخطوات التالية :

١. من نافذة قاعدة البيانات نشط التبويب Create "إنشاء" ومن مجموعة Forms "نماذج" انقر زر Form Design "تصميم النموذج".
تظهر نافذة تصميم النموذج . ويظهر التبويب Design "تصميم النموذج" هو المختار.

٢. من مجموعة Controls "عناصر التحكم" انقر زر Button "زر"

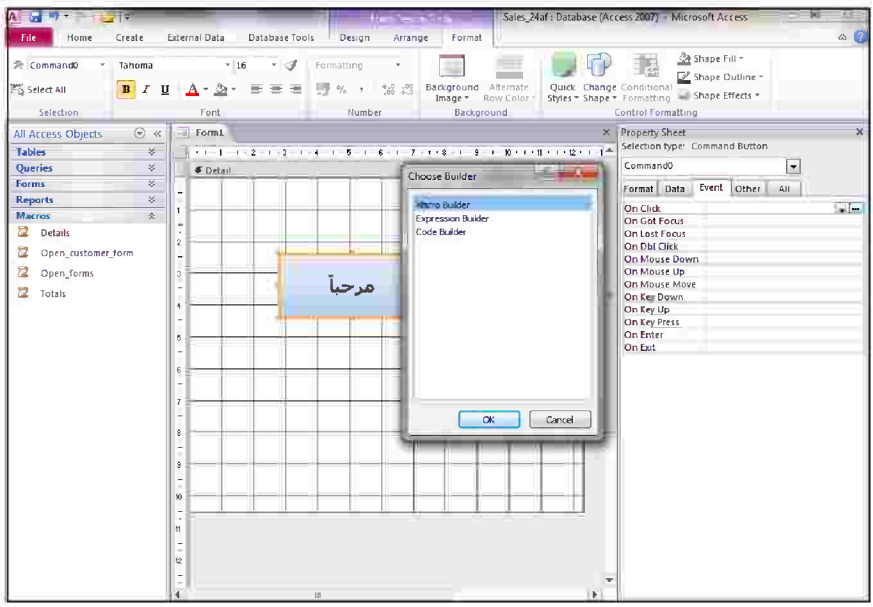
ثم ارسم زر أمر في منطقة Detail "تفاصيل" داخل النموذج بحجم مناسب، إذا ظهر مربع معالج زر الأمر انقر زر Cancel "إلغاء الأمر". يظهر زر الأمر داخل النموذج بعنوان مقترح، غيَّر العنوان المقترح إلى "مرحباً" (انظر شكل ٢٥-٥) .



شكل ٢٥-٥ إضافة زر أمر لنموذج لتشغيل برنامج "مرحبا" .

٣. ابق على زر الأمر محدداً ثم من التبويب Design "تصميم" ومن مجموعة Tools "أدوات" انقر زر Properties sheet "لوحة الخصائص". تظهر لوحة الخصائص أمامك. من لوحة Properties "الخصائص" نشط التبويب Event "حدث".

٤. أمام خاصية On Click "عند النقر" انقر الزر ...، تظهر نافذة Choose Builder "اختيار المنشئ" (انظر شكل ٢٥-٦) . اختر Code Builder "منشئ التعليمات البرمجية" ثم انقر زر Ok "موافق".

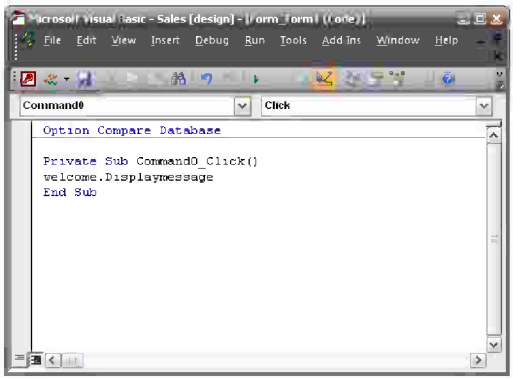


شكل ٢٥-٦ تحديد خصائص زر الأمر .

٥. تظهر بيئة التطوير المتكاملة وتجد نافذة جديدة لكتابة الكود المخصص للزر. من نقطة الإدراج اكتب الجملة الآتية :

```
Welcome.DisplayMessage
```

ثم اغلق بيئة التطوير المتكاملة (انظر شكل ٢٥-٧) .

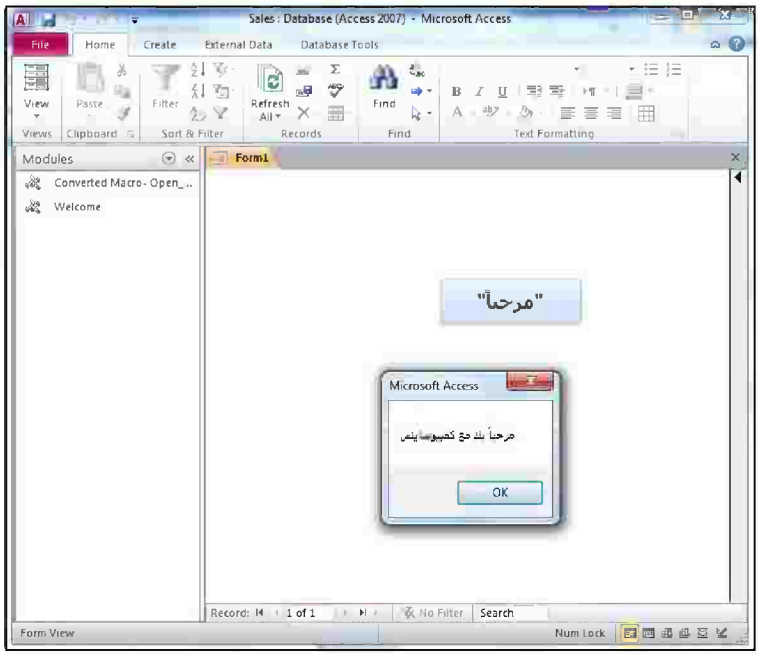


شكل ٢٥-٧ كتابة الكود داخل بيئة التطوير المتكاملة .

إذا كتبت الخاصية السابقة أمام خانة عند النقر ستفهم Access أن المطلوب هو استدعاء ماكرو وليس دالة داخل برنامج.

٦. بدل إلى طريقة عرض النموذج .

٧. انقر زر الأمر، تظهر الرسالة المطلوبة من البرنامج (انظر شكل ٢٥-٨).



شكل ٢٥-٨ مربع الرسالة نتيجة لتنفيذ الإجراء

٨. انقر زر Ok "موافق" لإغلاق مربع الرسالة ثم احفظ النموذج باسم نموذج رسالة ترجع إلى نافذة قاعدة البيانات.