نادي الشبيبة للعلوم والحاسوب في شهبا

لغة البرمجة المرئية

Visual Basic 6

طريق الإحتراف

أول الكلام

بعد اختراع الإنسان للكمبيوتر ، توجب عليه تصميم اللغات والأنظمة التي تتعامل معه ، فوجدت أولاً أنظمة التشغيل التي تمكنك من تشغيل الكمبيوتر والتعامل معه ومع محتوياته بالطريقة المثلى ، كما أنشأت لغات البرمجة التي تتيح لك فرصة تصميم البرامج المختلفة لمساعدتك على إنجاز أعمالك بسرعة أكبر ودقة أكثر بكثير من إنجازها يدوياً .

إن أول ما وُجد من لغات البرمجة هو لغة الآلة (Machine Language) ، وهي اللغة الأسهل بين اللغات بالنسبة للكمبيوتر ، والأعقد بينها بالنسبة للإنسان ، حيث يتوجب على من يريد التعامل مع هذه اللغة أن يكون ملماً إلماماً واسعاً بطريقة عمل الكمبيوتر وبنيته الداخلية وعناوين الذواكر فيه وما شابه ...

بعدها ظهرت اللغات التي يتعامل معها المستخدم باللغة العادية والتي كانت أسهل بكثير من سابقتها من حيث كتابة الأوامر والتعليمات والتعامل مع البيانات ... ومن هذه اللغات :

Basic – Pascal – Fortran – C – C++ .... وكانت جميعها تعمل ضمن بيئة النظام Dos ، مما يستدعي كتابة الكثير من التعليمات أثناء تصميمك لبرنامج ما فيها ، فمثلاً : إذا أردت إنشاء برنامج صغير تظهر فيه نافذة واحدة وزري أمر فقط ، توجب عليك كتابة العشرات من الأسطر والتعليمات الطويلة والمعقدة لإنشائه ....

هذا الأمر دعى الشركات البرمجية الكبيرة إلى اختراع لغات أكثر تطوراً وسهولة ، فوُجدت فكرة البرمجة المرئية ، حيث أَوجدَت عدة لغات مرئية مثل Visual Basic , Visual C++ , Visual Delphi وكلها لغات قوية تمتعت بالسهولة والمرونة حيث تجد فيها جميع النوافذ والأزرار جاهزة ، فمثلاً إذا أردت الآن إنشاء برنامج صغير تظهر فيه نافذة واحدة وزري أمر ، فلن يستغرق هذا من وقتك الثمين أكثر من دقيقة ، حيث ستجدها جاهزة ، وما عليك أنت سوى اختيارها ورسمها ... ثم كتابة الشيفرات المناسبة لها وبهذا تكون قد وفرت على نفسك الكثير والكثير من الوقت والجهد ، وهذا ما أعطى هذه اللغات شهرة واسعة وانتشار كبير .

يوجد في هذا المرجع ست عشر باباً ، يحوي بعضها على أكثر من برنامج واحد ، فإذا تمكنت من تنفيذها وفهمها كلها ، أصبحت مبرمجاً ملماً بجميع موجودات وخصائص لغة Visual Basic 6 ، وتستطيع تصميم أي برنامج يخطر على بالك .....

كما يوجد أيضاً في آخر المرجع ملحقاً شاملاً عن أهم خصائص هذه اللغة ، يمكنك الاطلاع عليه عند انتهائك من دراسة هذا المرجع ....

- كيفية إنشاء البرامج وتنفيذها ، بدءاً من أسهل البرامج إلى أعقدها . . . . .
- التعامل مع كافة خصائص وأدوات اللغة . . . .
- التعامل مع عناصر التحكم الخارجية . . . .
- إنشاء برامج الوثائق المتعددة ، والنوافذ الأبناء كالبرامج المكتبية وبرامج الرسم . . . .
- التعامل مع الملفات الصوتية . . . .
- إنشاء واستخدام قواعد المعطيات في البرامج . . . .
- التعامل مع الصور المتحركة وتوقيت الصوت والصورة . . . .
- التعامل مع الملفات وكيفية إنشاءها والوصول إليها . . . .
- محاكاة بعض برامج النظام المفيدة كالمفكرة والمستكشف . . . .
- إعداد وحزم البرامج والتجهيز لتسويقها . . . .

مع شرحٍ كامل لمراحل التصميم والبرمجة لكافة البرامج

SIA Programming

مفاهيم أولية	3
كتابة البرنامج الأول	6
تطوير البرنامج	11
النماذج المتعددة والرسائل	13
العداد	21
الواجهة متعددة الوثائق	25
المفكرة	29
أشرطة القوائم والأدوات والحالة	32
الحلقات والوقت الضائع	36
البرنامج الأول : رسم النقاط	36
البرنامج الثاني : رسم النقاط واستخدام الحلقات	38
التحكم بالملفات	40
البرنامج الأول : اختيار الملفات	40
البرنامج الثاني : عارض الصور	42
إنشاء وتحرير الملفات	45
إنشاء الملفات النصية	51
نسخ ونقل البيانات	54
برنامج الملفات الصوتية	57
برنامج الصوت والصورة	60
قواعد المعطيات	64
برنامج المستكشف	69
برنامج مفكرة المواعيد	72
إعداد وتجهيز البرامج	91
ملحق	94

مفاهيم أولية

قبل أن نبدأ بكتابة أي برنامج يتوجب علينا الاطلاع على بعض القضايا الأساسية في هذه اللغة

مراحل كتابة البرنامج . .

تمر كتابة أي برنامج في فيجوال بيسيك في المراحل التالية :

1- مرحلة البرمجة المرئية : وهي التي يتم فيها تصميم النوافذ والنماذج ومحتوياتها من أزرار أمر أو تسميات أو صور ....الخ .

2- مرحلة كتابة الشيفرة : وفيها يتم كتابة الأوامر اللازمة لكل كائن في المشروع .

3- مرحلة الاختبار : وهي لفحص طريقة عمل البرنامج وتفيد في اكتشاف الأخطاء الغير متوقعة أثناء مرحلة كتابة الشيفرة .

4- مرحلة التجميع : وفيها نقوم بتجميع كافة النماذج والبرامج الجزئية الموجودة في المشروع ووضعها في ملف تنفيذي واحد يحمل اللاحقة EXE .

إنشاء مشروع جديد . .

عند الدخول إلى فيجوال بيسيك يظهر مربع حوار باسم ((New project)) ، وفيه ثلاث أبواب ....

ألاحظ أن الباب New هو الباب الفعال : ويحتوي بداخله على عدة خيارات لإنشاء مشروع جديد ، وألاحظ أيضاً أن الخيار الفعال هو Standard.Exe .. لذلك ما عليّ سوى النقر على زر " **open** " , وبالتالي سينشأ مشروع جديد يحمل الاسم Project1 ويحوي بداخله على نموذج واحد أيضاً اسمه Form1 ...

يفضل أولاً حفظ المشروع في دليل العمل ، ويكون الحفظ باختيار الأمر Save Project من قائمة File ، فيظهر مربع حوار الحفظ :

أحتفظ أولاً بالنموذج حيث أعطيه اسماً وليكن MyProgram وأنقر زر " **Save** " وتلقائياً يأخذ الاسم اللاحقة Frm ((اختصاراً لكلمة Form)) ، بعد ذلك يظهر مربع حوار لحفظ المشروع ككل ، فأكتب له اسماً وليكن MyProgram [1] ثم أنقر زر " **Save** " وتلقائياً يأخذ الاسم اللاحقة Vbp وهي اختصار للكلمات التالية : Visual Basic Project وبالتالي أصبح لدي في دليل العمل :

[1] يفضل دوماً أثناء حفظ المشروع أن يكون للمشروع والنموذج نفس الاسم وذلك كي يتسنى لنا فيما بعد معرفة ما هي النماذج التابعة لكل مشروع

** النموذج باسم MyProgram.Frm : وفيه معلومات عن النموذج ومحتوياته التي سننشئها أثناء تصميم المشروع .

** المشروع ككل باسم MyProgram.Vbp ويحوي معلومات عن النماذج المحتواة في المشروع ،إذ يمكن أن يحتوي المشروع على أكثر من نموذج واحد كما سنرى لاحقاً .

نافذة فيجوال بيسيك . .

بعد دخولك إلى فيجوال بيسيك وإنشاءك للمشروع الجديد ، تلاحظ وجود أربع نوافذ أخرى غير نافذة المشروع الأساسية وهذه النوافذ هي :

1-)) نافذة المشروع ...Project Window : ويظهر فيها أسماء محتويات المشروع الحالي كالنماذج والبرامج الجزئية (التي سنراها لاحقاً ...) ويلاحظ فيها ثلاث أزرار أمر هي :

1- Toggle Folders يعرض محتويات المشروع كبنية شجرية (كما في برنامج المستكشف) .

2- View Object يعرض الكائن الحالي على الشاشة (وذلك في حال كان لدي أكثر من كائن في المشروع)

3- View Code يعرض نافذة كتابة الشيفرة للكائن المحدد .

ملاحظة : إذا لم تكن النافذة المذكورة ظاهرة على الشاشة فيمكنك إظهارها بضغط المفتاحين Ctrl+R من لوحة المفاتيح ، أو باختيار الأمر Project Explorer من قائمة View...

2-)) نافذة الخصائص Properties Window : ويعرض فيها خصائص الكائن المحدد ، كاللون والاسم وحجم الخط والطول والعرض والارتفاع ..... ، وألاحظ وجود بابين فيها هما :

أ- Alphabetic : وتستعرض فيها الخصائص مرتبة ترتيباً أبجدياً لاتينياً .

ب- Categorized وتستعرض فيها الخصائص مرتبة ترتيباً وظيفياً .

وبهذا نلاحظ أن الخاصية تؤثر في عمل أو شكل الكائن ، فمثلاً :

خاصية Caption تغير عنوان الكائن ..

خاصية BackColor تغير لون الكائن المحدد ..

خاصية Font تغير شكل وحجم الخط للكائن ...الخ .

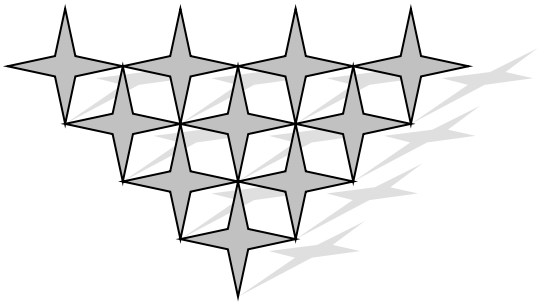
ملاحظة : إذا لم تكن النافذة المذكورة ظاهرة على الشاشة فيمكنك إظهارها بضغط المفتاح F4 من لوحة المفاتيح ، أو باختيار الأمر Properties Window من قائمة View...

3-)) نافذة توضع النموذج Form Layout Window : وفيها نتحكم بمكان عرض النموذج على الشاشة أثناء تنفيذ البرنامج ، ولنقل مكان النموذج أقوم بسحبه إلى المكان المطلوب ...

ملاحظة : إذا لم تكن النافذة المذكورة ظاهرة على الشاشة فيمكنك إظهارها باختيار الأمر Window Form Layout من قائمة View ...

4-)) نافذة صندوق الأدوات ToolBox Window : وتحوي جميع الكائنات والأغراض التي يحتاجها المبرمج لتصميم برنامجه (كأزرار الأمر ومربع النص وصندوق الصور وأزرار الاختيار ....الخ) ، ولإدراج أي من هذه الأغراض ضمن النموذج أنقر على الغرض المطلوب نقرتين متتاليتين فيتم إدراجه في النموذج ، بعدها أتحكم بموقعه على النموذج بسحبه بالماوس إلى المكان المطلوب ، و أيضاً أستطيع تغيير حجمه عن طريق المقابض التي تظهر عليه ...

- 1- إذا لم يكن صندوق الأدوات ظاهراً على الشاشة ، يمكنك إظهاره باختيار أمر ToolBox من قائمة View .
- 2- لمعرفة اسم أي كائن من هذه النافذة : ضع المؤشر عليه (دون نقر) فيظهر صندوق نص صغير يحتوي على الاسم .



كتابة البرنامج الأول

Writing The First Program

بعدما تعرفنا على نافذة فيجوال بيسيك الأساسية وكيفية التعامل معها ، أصبح بإمكاننا الآن إنشاء برنامج بسيط يمكننا من خلاله معرفة طريقة كتابة أي برنامج بمراحله الأربع .

قبل البدء بكتابة البرنامج سنفترض أن برنامج فيجوال بيسيك قد تم إعداده على القرص تحت مجلد فرعي باسم VB6 ، لذلك ولسهولة العمل سنقوم بإنشاء المجلدات الفرعية التالية ضمن المجلد VB6 :

1- **مجلد باسم MyPhotos** ننسخ إليه جميع الصور المنشأة في برنامج الرسام (Bmp.*) والتي سنحتاجها أثناء تصميم البرامج .

2- **مجلد باسم MyIcons** ننسخ إليه جميع ملفات الرموز (Ico.*) التي سنحتاجها أثناء تصميم البرامج .

3- **مجلد باسم MyExercise** وسأستخدمه لتخزين البرامج التي سأقوم بتصميمها ، بحيث أقوم بتقسيم هذا المجلد إلى (**ست عشر**) مجلد فرعي تكون أسماؤها بالشكل :

Exer1 ، Exer2 ، Exer3 ، ..... Exer(n)

بحيث أخزن البرنامج الأول في المجلد Exer1 ، والبرنامج الثاني في المجلد Exer2 ، وهكذا ....

البرنامج المطلوب

مطلوب إنشاء برنامج باسم FirstProg.Exe ، بحيث تظهر عند تشغيل هذا الملف نافذة فيها أربعة أزرار :

1) زر Welcome : بنقره تظهر عبارة / أهلاً بكم مع برنامج فيجوال بيسيك 6 / .
2) زر Date : بنقره يظهر التاريخ الحالي .
3) زر Time : بنقره يظهر الوقت الحالي .
4) زر Time –Date : بنقره يظهر التاريخ والوقت الحاليين .
5) زر Exit بنقره يتم إنهاء البرنامج .

خطوات تنفيذ البرنامج السابق

شغل برنامج فيجوال بيسيك ، وأنشئ مشروعاً جديداً بالطريقة التي رأيناها ....

احفظ النموذج وخزنه باسم FirstProg.Frm في المجلد الفرعي C:\Vb6\MyExercise\Exer1

احفظ المشروع وخزنه باسم FirstProg.Vbp في المجلد الفرعي C:\Vb6\MyExercise\Exer1

❋ تغيير خاصية العنوان للنموذج :

تأكد أن النموذج هو الفعال ، ثم أنقر على الخاصة Caption في نافذة الخصائص ، حول لغة الكتابة إلى اللغة العربية [1] ثم اكتب عبارة // برنامج الوقت والتاريخ //

❋ جعل العنوان يظهر من اليمين لليسار :

لاحظت أن عنوان النموذج قد ظهر في القسم الأيسر من شريط عنوان النموذج ، ولجعله يظهر في القسم الأيمن :
ادخل إلى الخاصية RightToLeft من نافذة الخصائص وحول قيمتها إلى True ..

❋ تغير خاصية الاسم للنموذج :

إن أي كائن في فيجوال بيسيك يكون له اسم داخلي ، وعن طريق هذا الاسم يتمكن المبرمج وبرنامج فيجوال بيسيك من التعامل مع هذا الغرض ، كما أن فيجوال بيسيك لا يمكنه الوصول إلى أي غرض إلا عن طريق اسمه ... لذلك يعد اسم أي كائن مهماً في جميع المراحل .

لتغيير اسم النموذج ادخل إلى الخاصية Name من نافذة الخصائص ، واكتب فيه / FrmFirstProg / ...

لاحظ أننا سبقنا اسم النموذج بالأحرف / Frm / وطبعاً هذا ليس إجبارياً ، ولكن يفضل أن نسبق اسم أي كائن بثلاث أحرف تدل على نوع هذا الكائن (زر أمر – مربع نص – صورة – ....) وذلك كي يسهل علينا الرجوع إليه إذا حدث أي خطأ أثناء كتابة الشيفرة ...

❋ إضافة زر أمر الخروج ((Exit)) :

لإضافة زر أمر : أنقر نقرتين متتاليتين على CommandButton من نافذة صندوق الأدوات ، فيظهر زر الأمر في النموذج ، حركه إلى المكان الذي تريده في النموذج وتحكم بحجمه ، وبعدها غير خصائصه كما يلي :

- غير اسمه (خاصية Name) إلى / CmdExit /
- غير عنوان (خاصية Caption) إلى / E&xit /

: أننا وضعنا الرمز (&) قبل الحرف (X) في عنوان الزر ، وفائدة ذلك أن الزر يصبح فعالاً بضغط المفتاحين Alt + X من لوحة المفاتيح ، وبمعنى آخر (يمكنك الخروج من البرنامج بالنقر على زر Exit أو بضغط المفتاحين Alt + X من لوحة المفاتيح) .

❋ إضافة زر إظهار عبارة الترحيب ((Welcome)) :

بنفس الطريقة السابقة أضف زر أمر جديد وغير اسمه إلى / CmdWelcome / ، و أيضاً غير عنوانه إلى / & Welcome / .

[1] للتحويل إلى اللغة العربية اضغط مفتاحي Alt + Shift على اليمين ، أو اختر الرمز AR من شريط المهام ... أما إذا أردت التحويل من اللغة العربية إلى اللغة اللاتينية اضغط مفتاحي Alt + Shift على اليسار ، أو اختر الرمز EN من شريط المهام ...

❋ إضافة زر إظهار الوقت ((Time)) :

بنفس الطريقة السابقة أضف زر أمر جديد وغير اسمه إلى / CmdTime / ، و أيضاً غير عنوانه إلى / &Time / .

❋ إضافة زر إظهار التاريخ ((Date)) :

أضف زر أمر جديد وغير اسمه إلى / CmdDate / ، و أيضاً غير عنوانه إلى / &Date / .

❋ إضافة زر إظهار الوقت والتاريخ ((Date & Time)) :

أضف زر أمر جديد وغير اسمه إلى / CmdDateTime / ، و أيضاً غير عنوانه إلى / Time – Date / .

❋ إضافة صندوق النصوص TextBox إلى النموذج :

(وهو الصندوق الذي سيظهر به التاريخ أو الوقت أو كليهما معاً عند النقر على أحد الأزرار السابقة)

لإضافة صندوق نصوص : أنقر نقرتين متتاليتين على TextBox من نافذة صندوق الأدوات ، فيظهر صندوق النصوص في النموذج ، حركه إلى المكان الذي تريده في النموذج وتحكم بحجمه ، وبعدها غير خصائصه كما يلي :

- غير اسمه (خاصية Name) إلى / TxtDisplay /
- غير محتواه (خاصية Text) واجعلها فارغة ...
- غير خاصية الخط (Font) ، واجعل الخط بحجم / 14 /
- غير خاصية المحاذاة (Alignment) واجعلها مساوية لـ / 2-Center / ، وهي لجعل الكتابة تظهر في منتصف صندوق النصوص .
- غير خاصية تعدد الأسطر (MultiLine) واجعلها مساوية لـ / True /

إن الخاصية الأخيرة (MultiLine) تجعل إمكانية تعدد الأسطر متاحة ، وبالتالي إذا كان النص المكتوب في صندوق النصوص كبيراً ، فسيتم تجزئته إلى عدة سطور بحيث يتناسب طول كل سطر مع حجم الصندوق . وبفرض أننا جعلنا هذه الخاصية غير متاحة ، أي جعلنا قيمتها = False فحينها سيظهر كامل محتوى الصندوق على سطر واحد مما يؤدي إلى اختفاء قسم منه .

بهذا نكون قد انتهينا من مرحلة البرمجة المرئية ، وعلى الرغم من أننا لم ننتهي بعد من تصميم كل البرنامج ، لكنه بالإمكان تنفيذه لرؤية طريقة عمله :

احفظ البرنامج بالنقر على زر الحفظ Save Project من شريط الأدوات القياسي .

يمكنك تنفيذ البرنامج بعدة طرق :

1- اختيار الأمر Start من القائمة Run .
2- ضغط المفتاح F5 من لوحة المفاتيح .
3- نقر الزر Start من شريط الأدوات القياسي .

والنتيجة سيتم عرض المشروع على الشاشة ، و لإنهاء العرض والعودة إلى مرحلة التصميم ، أنقر على زر End

كتابة الشيفرات لأزرار الأمر

كتابة الشيفرة تعني إعطاء التعليمات الواجب تنفيذها أثناء استخدام الكائن .... فمثلاً :

نحن نريد من البرنامج أن **ينتهي** عند **النقر** على للزر Exit , لذلك يتوجب علينا إعلام فيجوال بيسيك بذلك ، ويكون هذا باستخدام الشيفرة التي سنكتبها كما يلي :

(✻) أنقر نقرتين متتاليتين على الزر Exit ..... ستفتح نافذة كتابة الترميز **Code** ، وتلاحظ في أعلى هذه النافذة قسمي عنوان :

1- **القسم الأيسر** : يظهر فيه اسم الكائن الذي ستتم كتابة الشيفرة له ، وهو في مثالنا (الكائن CmdExit) .

2- **القسم الأيمن** : يظهر فيه اسم الحدث الذي ستُنفذ الشيفرة حال وقوعه ، وهو في مثالنا الحدث Click (والذي يعني/ النقر /) ، وسنتعرف لاحقاً على هذه الأحداث .

(✻) في قسم كتابة الشيفرة من النافذة المذكورة ، أكتب تعليمة إنهاء البرنامج وهي : ***END*** وبالتالي ستصبح الشيفرة لديك كما بالشكل :

```
Private Sub CmdExit_Click( )
End
End Sub
```

لاحظت أن السطرين الأول والأخير مكتوبين بشكل تلقائي ، حيث أنهما يظهران دائماً عند كتابة شيفرة أي كائن ، وهما يُعلِمان فيجوال بيسيك ببدء وانتهاء تنفيذ الشيفرة للكائن المحدد ....

احفظ المشروع و نفذه ، ولاحظ أنه بنقرك على الزر Exit يتم إنهاء العمل والعودة إلى مرحلة التصميم ..

✻ كتابة الشيفرة لحدث النقر على الزر Welcome :

أنقر نقرتين على الزر Welcome فتفتح نافذة كتابة الشيفرة ، اكتب فيها الشيفرة التالية :

```
TxtDisplay.Text = " أهلا بكم في برنامج فيجوال بيسيك 6 "
```

إن الشيفرة السابقة تعني إسناد عبارة " أهلا بكم في برنامج فيجوال بيسيك 6 " إلى خاصية النص Text التابعة لصندوق النصوص TxtDisplay .

✻ كتابة الشيفرة لحدث النقر على الزر Time :

أكتب الشيفرة التالية في حدث النقر على الزر Time :

```
TxtDisplay.Text = Time()
```

يرجع التابع Time() قيمة الوقت الحالي ويسندها إلى الخاصية Text العائدة لصندوق النصوص ....

✻ كتابة الشيفرة لحدث النقر على الزر Date :

أكتب الشيفرة التالية في حدث النقر على الزر Date:

```
TxtDisplay.Text = Date ()
```

يرجع التابع Date () قيمة التاريخ الحالي ويسندها إلى الخاصية Text العائدة لصندوق النصوص ....

❋كتابة الشيفرة لحدث النقر على الزر Date_- Time :

أكتب الشيفرة التالية في حدث النقر على الزر Date - Time :

```
TxtDisplay.Text = Now ()
```

يرجع التابع Now () قيمة الوقت و التاريخ الحاليين ويسندها إلى الخاصية Text العائدة لصندوق النصوص ....

بهذا نكون قد انتهينا من مرحلة كتابة الشيفرة ، وأصبح البرنامج جاهزاً للتنفيذ ، لذلك قم بحفظه ثم تنفيذه ، وجرب جميع الأزرار ، وتأكد من أن البرنامج يعمل بشكل صحيح .

تفيد الخطوة الأخيرة (وهي مرحلة اختبار عمل البرنامج) في معرفة الأخطاء التي مرت معنا دون الانتباه إليها ...

بعد تأكدك من أن البرنامج يعمل بالشكل المطلوب ، أصبح بمقدورك الآن تنفيذ المرحلة الأخيرة من مراحل تصميم المشروع ، وهي مرحلة التجميع وإنشاء ملف تنفيذي ، ويكون ذلك كما يلي :

ادخل إلى قائمة File واختر منها الأمر Make FirstProg.Exe ، سيظهر مربع حوار لتحديد اسم الملف التنفيذي وموقعه ، بإمكانك كتابة اسم وموقع جديد **لكن** يفضل الاحتفاظ به بنفس الاسم المقترح عليك وفي المجلد الفرعي C:\Vb6\MyExercise\Exer1

تجريب الملف التنفيذي :

بعد أن أنشأت ملف تنفيذي للبرنامج ، أصبح بمقدورك تنفيذه دون الدخول إلى فيجوال بيسيك نهائياً ولفعل ذلك اتبع الخطوات التالية :

1- أغلق برنامج فيجوال بيسيك نهائياً (ستظهر لك رسالة تطالبك بحفظ المشروع ، لذلك أنقر على الزر Yes) .
2- أدخل إلى برنامج المستكشف ، ومنه ادخل إلى المجلد C:\Vb6\MyExercise\Exer1 .
3- أنقر نقرتين متتاليتين على الملف FirstProg.Exe .

تلاحظ أن البرنامج بدأ بالعمل على الرغم من أنك لم تشغله من ضمن فيجوال بيسيك ، وهذا يعني أنه يمكنك تنفيذ هذا البرنامج على حاسب آخر لا يحتوي على نسخة فيجوال بيسيك *

---

* على الرغم من أننا نستطيع تنفيذ هذا البرنامج على حاسب آخر لا يحتوي على نسخة فيجوال بيسيك ، إلا أننا نحتاج إلى ملفات أخرى غير الملف التنفيذي الذي أنشأناه ، على سبيل المثال : نحتاج إلى الملف Msvbvm60.dll الذي يجب أن يتواجد في دليل العمل أو في المجلد C:\Windows\System\ ، وعلى كل حال سنتعلم لاحقاً كيفية إنشاء برنامج إعداد لأي برنامج ننشؤه ، حيث يقوم برنامج الإعداد تلقائياً بنسخ الملفات التي يحتاجها للتنفيذ إلى دليل النظام C:\Windows\System\

تطوير البرنامج السابق . .

بعد نجاحنا بإنشاء البرنامج وتنفيذ جميع مراحله الأربعة ، سنقوم الآن بإضافة بعض التعديلات عليه ، بأن نضع صورة كخلفية للنموذج ونغير في لون الكتابة ونضيف بعض الأزرار الجديدة إليه ، وهذه الأزرار هي :

I- زر Year يعطي السنة الحالية .

II- زر Month يعطي الشهر الحالي .

III- زر Hour يعطي الساعة الحالية .

IV- زر Minute يعطي الدقائق الحالية .

V- زر Clean يقوم بمسح محتوى صندوق النصوص .

للعودة إلى البرنامج السابق :

1. شغل برنامج فيجوال بيسيك ، فيظهر مربع حوار New .
2. أنقر على باب Existing وهو لفتح مشروع منشأ مسبقاً .
3. ادخل إلى المجلد C:\Vb6\MyExercise\Exer1 ، واختر اسم المشروع FirstProg.Vbp ، ثم أنقر الزر **Open** .

*** وضع صورة تظهر على خلفية النموذج :**

ادخل إلى خاصية Picture للنموذج وحدد الصورة / C:\VB6\MyPhotos\Prog1.Bmp / .

*** تغيير أيقونة البرنامج :**

ادخل إلى الخاصية Icon للنموذج وحدد الأيقونة / C:\VB6\MyIcons\Prog1.Ico / .

*** تغيير لون صندوق النصوص ولون الكتابة فيه :**

- لتغيير لون خلفية صندوق النصوص : ادخل إلى الخاصية BackColor واختر اللون الأسود .
- لتغيير لون النص الذي يظهر فيه : ادخل إلى الخاصية ForeColor واختر اللون الأصفر .

تلاحظ عند الدخول إلى خاصية اللون ظهور الألوان مبوبة في بابين :

أ- System : ويعرض فيها الألوان المستخدمة في نظام التشغيل (لون شريط العنوان أو الخطوط أو الرسائل ...)

ب- Palette : ويحوي جميع الألوان ، ويمكنك أيضاً اختيار ألوان إضافية ، فإذا أردت ذلك :

أنقر على اللون المطلوب بالزر الأيمن واستمر بالنقر واسحبه إلى المربعات البيضاء الموجودة أسفل هذه النافذة ، ثم حرر الماوس ، فيظهر لك مربع حوار لتحديد اللون المرغوب والتحكم بدرجته ، بعدها أنقر على زر Add Color فيصبح اللون الذي اخترته ضمن الباب Palette .

❋ إنشاء أزرار الأمر :

أنشأ خمسة أزرار أمر في النموذج واجعل عناوينها على التوالي :
&Year , &Month , &Hour , Mi&nute , &Clean

وغير أسماءها أيضاً على التوالي إلى :
CmdYear , CmdMonth , CmdHour , CmdMinute , CmdClean

بقي عليك بذلك كتابة الشيفرات المناسبة لحدث النقر على هذه الأزرار ، لذلك ادخل إلى حدث النقر لكل زر على حدا ، واكتب لكل منها الشيفرة المناسبة .... وإليك هذه التوابع التي ستحتاج إلى قسماً منها :

تابع Year(Date) يرجع قيمة السنة من التاريخ الحالي .
تابع Month(Date) يرجع قيمة الشهر من التاريخ الحالي .
تابع Day(Date) يرجع قيمة اليوم من التاريخ الحالي .
تابع Hour(Time) يرجع قيمة الساعة من الوقت الحالي .
تابع Minute(Time) يرجع قيمة الدقائق من الوقت الحالي .
تابع Second(Time) يرجع قيمة الثواني من الوقت الحالي .

❋ كتابة شيفرة حدث النقر على زر Clean :

اكتب الشيفرة التالية في حدث النقر على زر Clean

```
TxtDisplay.Text = " "
```

بعد انتهائك من كتابة الشيفرة نفذ البرنامج وتأكد من صحة عمله ، احتفظ به ، واصنع منه ملف تنفيذي بنفس الطريقة المشروحة سابقاً واحتفظ بالملف التنفيذي في المسار :

C:\Vb6\MyExercise\Exer1\ NewFirstProg.Exe

برنامج النماذج المتعددة والرسائل

The Multi Forms & Messages Program

سنتعلم في هذا البرنامج كيفية إنشاء اكثر من نموذج مع إظهار رسائل تُعلِم المستخدم بشيءٍ ما ....

فكرة البرنامج :

عند تشغيل هذا البرنامج ، تظهر الواجهة الأساسية وفيها أربع أزرار أمر :

1- زر About 1 : بنقره تظهر رسالة تشير إلى مصمم هذا البرنامج ...

2- زر About 2 : بنقره يظهر نموذج آخر فيه صورة ومجموعة عبارات ...

3- زر Dialog box : يظهر صندوق حوار لإدخال عبارة ما ، ثم تظهر هذه العبارة على واجهة البرنامج الأساسية ...

4- زر Dialog form : يظهر نموذج حوار لإدخال عبارة ما ، ثم تظهر هذه العبارة على واجهة البرنامج الأساسية ...

5- زر Exit : لإنهاء البرنامج ...

التنفيذ المرئي للبرنامج :

أنشئ مشروعاً جديداً واحفظه في المجلد Exer2 بحيث :

- يكون اسم النموذج C:\Vb6\MyExercise\Exer2\MultiForm.Frm
- يكون اسم المشروع C:\Vb6\MyExercise\Exer2\MultiForm.Vbp

: لقد أسهبنا في طريقة إنشاء وكتابة الشيفرات للكائنات التي مرت معنا في البرنامج الأول ، ولكن سنعتمد بدءاً من هذا البرنامج وحتى نهاية المرجع على أسلوب الجداول في التصميم ، وسيتضح لك ذلك أثناء إنشاء هذا البرنامج :

نفذ النموذج لديك مستعيناً بهذا الجدول :

Object	Properties	Set
Form	Name	FrmMultiForm
	Caption	النماذج المتعددة والرسائل
	RightToLeft	True
	Picture	C:\VB6\MyPhotos\MultiForm.Bmp
	Icon	C:\VB6\MyIcons\MultiForm.Ico
CommandButton	Name	CmdAbout1
	Caption	About1
CommandButton	Name	CmdAbout2
	Caption	About2
CommandButton	Name	CmdDialogBox
	Caption	Dialog Box
CommandButton	Name	CmdDialogForm
	Caption	Dialog Form
CommandButton	Name	CmdExit
	Caption	E&xit

*** كتابة الشيفرة للزر Exit :**

اكتب الشيفرة التالية في حدث النقر على زر Exit

```
Private Sub CmdExit_Click( )
End
End Sub
```

*** شيفرة الزر About1 :**

اكتب الشيفرة التالية في حدث النقر على زر About 1 :

MsgBox " 6 تم تصميم هذا البرنامج في لغة فيجوال بيسيك ", vbInformation, "حول البرنامج"

إن التابع MsgBox يظهر رسالة مؤلفة من ثلاث وسطاء :

1- الأول : هو نص الرسالة ، وهي في مثالنا رسالة : // تم تصميم هذا البرنامج في لغة فيجوال بيسيك 6 // .

2- الثاني : هو صورة أو إشارة صغيرة تظهر على الرسالة ، وهي في مثالنا إشارة // vbInformation // أي إشارة المعلومات .

3- الثالث : هو عنوان الرسالة التي ستظهر ، وهو في مثالنا عبارة // حول البرنامج // . (*)

قد يمر معك أحياناً شيفرات طويلة نسبياً كالشيفرة السابقة ، لذلك باستطاعتك أن تجعلها تظهر على عدة أسطر وذلك بأن تجزئها باستخدام الشرطات السفلية ، فمثلاً : نستطيع كتابة شيفرة الزر About 1 بطريقة أخرى كما يلي :

```
MsgBox " 6 تم تصميم هذا البرنامج في لغة فيجوال بيسيك ", _
       vbInformation, _
       "حول البرنامج"
```

لاحظت : أن الرسالة التي ظهرت بالنقر على الزر السابق كانت محدودة الخيارات ، لكنك قد ترغب أحياناً بأن تكون هذه الرسالة تحمل صورة أو عبارات أكثر منها ، لذلك سنقوم الآن بإنشاء هذه الرسالة ، لكن يتوجب علينا أولاً إنشاء نموذج جديد ..

إنشاء نموذج جديد

ادخل إلى قائمة Project واختر منها أمر Add Form ، فيظهر مربع حوار ، أنقر منه زر **Open** .....

احفظ النموذج الجديد بنفس مجلد العمل الحالي وباسم About2.Frm ، وغير خصائصه مستعيناً بالجدول التالي :

Object	Properties	Set
Form	Name	FrmAbout2
	Caption	About this program
	BorderStyle	1- Fixed Single
	Picture	C:\VB6\MyPhotos\About2.bmp
CommandButton	Name	CmdOk
	Caption	&Ok
	Default	True

(*) يمكنك مراجعة الجدولين / 3 – 4 / آخر المرجع لمعرفة مزايا التابع MsgBox

إن الخاصية BorderStyle تتحكم بطريقة عرض النموذج و إمكانية تغيير حجمه أم لا ، ولها خمس حالات :

الحالة	الوصف
0-None	يظهر النموذج دون شريط عنوان نهائياً
1-Fixed Single	يصبح النموذج غير قابل لتغيير حجمه ويمكنك أن تضيف إليه زر التصغير
2-Sizeable	يمكن تغيير حجمه ويظهر فيه زري التكبير والتصغير (1)
3- Fixed Dialog	يصبح النموذج غير قابل لتغيير حجمه ولا يمكن أن تضيف إليه زر التصغير
4-Fixed Tool Window	لا يظهر فيه سوى زر الإغلاق فقط ولا يظهر مربع التحكم
5- Sizeable Tool Window	نفس الخيار السابق لكن يمكن فيه تغيير الحجم

❋ خاصية Default = True تجعل النقر على هذا الزر مكافئاً للضغط على مفتاح Enter

❋ شيفرة الزر About2 :

اكتب الشيفرة التالية في حدث النقر على زر About 2

```
FrmAbout2.Show 1
```

إن الشيفرة السابقة تظهر النموذج About2 على الشاشة ، أما وجود الرقم (1) فهذا يجعل النموذج فعالاً وحيداً ، أي لا يمكن للمستخدم أن ينقر في أي مكان إلا بعد إغلاقه للنموذج الحالي ..... فما هي فائدة ذلك ؟ ..

إن النموذج عندما يُفتح يستهلك قسماً من الذاكرة ، فلو أتحنا الخيار للمستخدم بأن ينقر على الزر About 2 عدة مرات فسيؤدي ذلك إلى فتح النموذج الأخير أكثر من مرة وبالتالي نكون قد استهلكنا قسم كبير من الذاكرة دون سبب ، لذلك جعلنا هذا النموذج نموذجاً فعالاً وحيداً ، وذلك بأن أضفنا الرقم (1) إلى نهاية الإجراء Show

❋ شيفرة الزر Ok في النموذج About 2 :

اكتب الشيفرة التالية في حدث النقر على زر CmdOk

```
FrmAbout2.Hide
```

إن الشيفرة السابقة تخفي النموذج About2 (لا تقوم بإغلاق النموذج بل تخفيه فقط ، أي يبقى محملاً في الذاكرة ...)

(1) إن هذا الشكل هو الوحيد من بين الأشكال الخمسة السابقة التي يظهر فيها النموذج بالشكل الطبيعي ، أما الأشكال السابقة تجعل النموذج يظهر على شكل مربعات الحوار التي تعرفها ...

❊ شيفرة الزر Dialog box في النموذج الأساسي

اكتب الشيفرة التالية في حدث النقر على زر Dialog box

```
Dim Inp_Text
Inp_Text = InputBox ( " أكتب هنا " , " صندوق إدخال " , " أدخل أي نص " )
Print " You Write.... " + Inp_Text
```

قمنا أولاً بتعريف متحول ، سنضع فيه قيمة النص الذي سنكتبه ، ويكون التعريف باستخدام الكلمة المحجوزة Dim ، بعدها استخدمنا تابع InputBox حيث يرجع هذا التابع قيمة ما يتم وضعها في المتحول Inp_Text ، ثم استخدمنا أمر الطباعة Print الذي سيطبع على نافذة النموذج القيمة التي أدخلناها مسبوقة بعبارة .....You Write

إنشاء صندوق إدخال مخصص :

صندوق الإدخال الذي أنشأناه منذ قليل كان محدوداً ولا يحوي إلا على العبارة التي طلبنا كتابتها ، لذلك سننشئ نموذجاً جديداً بحيث نجعله كصندوق إدخال ...

أضف نموذج جديد واحفظه باسم Input .Frm وصممه مستعيناً بالجدول التالي :

Object	Properties	Set
Form	Name	FrmInput
	BorderStyle	1-Fixed Single
	Caption	Input any value
	Picture	C:\VB6\MyPhotos\Input.Bmp
CommandButton	Name	CmdOk
	Caption	Ok
CommandButton	Name	CmdCancel
	Caption	Cancel
	Cancel	True (1)
TextBox	Name	TxtInput
	Text	أكتب هنا

❊ كتابة شيفرة الزر Dialog Form

اكتب الشيفرة التالية في حدث النقر على زر Dialog Form

```
FrmInput.show 1
FrmMultiForm.Cls (2)
Print " You Write ...." + FrmInput.TxtInput.Text
```

لاحظ أنه في السطر الأخير من الشيفرة السابقة كتبنا عبارة FrmInput.TxtInput.Text وهي للوصول إلى الخاصية Text العائدة للكائن TxtInput العائد بدوره للنموذج FrmInput ، أي أننا نستطيع الوصول إلى أي خاصية بذكر النموذج والكائن التابعة له.....

بإمكانك الاحتفاظ بالمشروع وتنفيذه ، ولكنك ستلاحظ أنه لا يعمل بشكل كامل عندما تظهر النموذج الجديد وتنقر على زري Ok أو Cancel إذ أننا لم ننتهي بعد من البرنامج بشكل كامل ....

(1) هذه الخاصية تجعل الزر يستجيب للضغط على مفتاح Esc

(2) تعني مسح جميع العبارات المكتوبة على النموذج FrmMultiForm

*** كتابة شيفرتي الزرين Ok و Cancel :**

من المعروف أن أي مربع حوار يحوي عادة على زري أمر هما :

1- **Ok** (**موافق**) : يعني النقر عليه قبول جميع التغيرات المدخلة وإغلاق مربع الحوار

2- **Cancel** (**إلغاء الأمر**) : يعني النقر عليه رفض جميع التغيرات المدخلة وإغلاق مربع الحوار ، أي استعادة الحالة التي كانت قبل الدخول لمربع الحوار هذا ....

سنحتاج لكتابة الشيفرتين المطلوبتين إلى استخدام مفهوم المتحول العام ، ويكفي الآن أن نعلم أن المتحول العام هو المتحول الذي يمكن الوصول إليه من أي برنامج جزئي أو نموذج فرعي موجود في المشروع ، وكي نستطيع التصريح عن متحول عام علينا إنشاء برنامج جزئي مستقل (Module) ، وتكون خطوات الإنشاء باتباع الخطوات التالية :

1- ادخل إلى قائمة Project واختر الأمر Add Module

2- يظهر مربع حوار أنقر منه زر Open ...

احفظ البرنامج الجزئي الجديد باسم MultiForm.Bas

أكتب في قسم التصريحات العامة الأمر التالي :

```
Option Explicit
Global gInp_Text
```

بذلك نكون قد عرفنا متحول عام باسم gInp_Text وسيستطيع أي برنامج جزئي أو نموذج فرعي موجود في المشروع من التعامل معه ...

$ في الحدث Form_Load للنموذج FrmMultiForm أكتب الشيفرة التالية :

```
gInp_Text = " اكتب هنا "
```

تُحمل الشيفرة السابقة عند تحميل النموذج ، أي عند بدء تشغيل البرنامج

*** كتابة شيفرة النقر على الزر Ok :**

اكتب الشيفرة التالية في حدث النقر على زر Ok :

```
FrmInput.Hide
gInp_Text = Txtinput.Text
```

*** كتابة شيفرة النقر على الزر Cancel :**

اكتب الشيفرة التالية في حدث النقر على زر Cancel :

```
FrmInput.Hide
```

*** تعديل شيفرة النقر على الزر Dialog Form :**

عدل شيفرة الزر Dialog Form لتبدو بعد انتهاءك منها كما يلي :

```
FrmInput.TxtInput.Text = gInp_Text
Frminput.Show 1
Frmmymulti.Cls
Print " You Write  :  " + gInp_Text
```

*** تحديد كائن بدء التشغيل :**

لقد أصبح في المشروع ثلاث نماذج وبرنامج جزئي ، لذلك يتوجب علينا إعلام فيجوال بيسيك ما هو الكائن الذي يجب تحميله أولاً ... لذلك ادخل إلى القائمة Project واختر أمر Project Properties ... سيظهر مربع حوار ، تأكد من تفعيل باب General ثم أنقر على خانة Startup Object واختر منها اسم النموذج المطلوب تشغيله أول تحميل المشروع والذي هو النموذج FrmMultiForm ....

احتفظ بالمشروع ثم نفذه وتأكد من صحة عمله ... ثم أجري له نسخة تنفيذية واحفظ الملف التنفيذي باسم Multi.Exe في مجلد العمل C:\Vb6\MyExercise\Exer2

كان بالإمكان أن ننفذ الخطوة الأخيرة من المشروع ((وهي كتابة شيفرتي النقر على زري Ok و Cancel)) بطريقة أسهل بكثير ، ودون الخوض في كل ما ذكرناه ، ولكننا تعمدنا ذلك بقصد العمل أكثر في البرنامج مما أكسبنا خبرة أكثر ... و أيضاً التعرف على مفاهيم جديدة في فيجوال بيسيك كالمتحول العام والبرامج الجزئية (Modules) ....

** حاول بنفسك أن تصمم هذه الفكرة بالطريقة الأسهل .... فقط أضف إلى النموذج الأساسي كائن جديد وهو Label ثم اجعل زر Ok يسند القيمة التي تدخلها في نموذج الإدخال (FrmInput) إلى هذا الكائن ، واجعل زر الإلغاء Cancel يخفي النموذج فقط دون أن يسند أي قيمة .....

تطوير البرنامج السابق

سنضيف للبرنامج السابق زر أمر جديد ، بحيث يظهر عند نقره نموذج آخر يتحرك فيه قسم من الصورة باستمرار ، ليفت نظر المستخدم لإدخال قيمة ما ...

أضف إلى النموذج الأساسي الكائنان التاليان حسب الجدول :

Object	Properties	Set
CommandButton	Name	CmdNewDialog
	Caption	New Dialog
Label	Name	Label1
	Caption	اجعله فارغاً

❋ أضف نموذج جديد واحفظه باسم NewShow.Frm واجعل خصائصه مطابقة للجدول التالي :

Object	Properties	Set
Form	Name	FrmNewShow
	BorderStyle	1-Fixed Single
	Caption	New Show...
CommndButton	Name	CmdOk
	Caption	Ok
CommndButton	Name	CmdCancel
	Caption	Cancel
	Cancel	True
Timer	Name	Timer1
	Interval	300
TextBox	Name	TxtInput
	Caption	اجعله فارغاً

❋ اكتب الشيفرة التالية في حدث النقر على زر New Dialog :

```
FrmNewShow.Show 1
```

❋ اكتب الشيفرة التالية في حدث النقر على زر Ok في النموذج FrmNewShow :

```
FrmMultiForm.Label1. Caption = TxtInput.Text
FrmNewShow . Hide
```

❋ اكتب الشيفرة التالية في حدث النقر على زر Cancel في النموذج FrmNewShow :

```
FrmNewShow . Hide
```

جرب البرنامج بعد حفظه ... أنقر على زر New Dialog فتلاحظ ظهور النموذج الجديد

اكتب عبارة ما ثم أنقر Ok فتلاحظ ظهورها على واجهة البرنامج الأساسية ، أما إذا نقرت زر Cancel فتلاحظ عدم ظهورها ويتم الاحتفاظ بالعبارة السابقة ...

** إضافة الصور المتحركة للبرنامج :

أولاً : عليك إنشاء عشر صور في برنامج الرسام وتخزينها في نفس دليل العمل ، وذلك كما يلي :

أرسم أول صورة واكتب فيها عبارة **Write Here** .... احفظها باسم Ne10.Bmp

احذف حرف الـ **e** بالممحاة ثم اختر أمر / حفظ باسم/ من قائمة / ملف / واحفظها باسم Ne9.Bmp

احذف حرف الـ **r** بالممحاة ثم اختر أمر / حفظ باسم/ من قائمة / ملف / واحفظها باسم Ne8.Bmp .... وهكذا..

استمر بهذه الخطوات حتى تحصل على عشر صور مخزنة في دليل العمل الحالي ....

*** كتابة شيفرة الكائن Timer :**

اكتب الشيفرة التالية للكائن Timer1 :

```
Private Sub Timer1_Timer()
Dim BmpPath, BmpFileName
Static PhotoNum
  BmpPath = App.Path
  If Right (BmpPath, 1) <> "\" Then
     BmpPath = App.Path + "\"
  Else
     BmpPath = App.Path
  End If
     PhotoNum = PhotoNum + 1
  If PhotoNum > 10 Then
     PhotoNum = 1
  End If
  BmpFileName = BmpPath + "NE" + Format(PhotoNum) + ".bmp"
  FrmNewShow.Picture = LoadPicture(BmpFileName)
End Sub
```

* عرّفنا أولاً متحولين محليين باستخدام Dim ، حيث ستضع في المتحول الأول مسار الصورة النقطية ، وفي الثاني سنضع الاسم الكامل للصورة ((المسار والاسم واللاحقة)) ، ثم عرفنا متحول ساكن باستخدام عبارة Static وسوف تتحول قيمته من (1-10) ليتم عرض الصور .... **لكن** ما الفرق بين التحول الساكن والمحلي ؟؟ ....

المتحول المحلي يفقد قيمته عند انتهاء تنفيذ الإجراء .. .. بينما يتمتع المتحول الساكن بقدرته على الاحتفاظ بقيمته حتى بعد نهاية تنفيذ الإجراء ...

* التابع App.Path يعيد سلسلة محرفية تدل على مسار الدليل الذي ينفذ فيه البرنامج....

* التابع LoadPicture يعيد الصورة المشار إلى اسمها ومسارها باستخدام الوسيط الذي يلحق به

نفذ البرنامج ولاحظ أن الصورة تتبدل كل / 300 / ميلي ثانية ، وذلك حسب الخاصية Interval

اصنع ملف تنفيذي لبرنامجك واحفظه في نفس الدليل الحالي باسم NewMulti.Exe **ولاحظ** أنه إذا رغبت فيما بعد بنقل الملف التنفيذي إلى مجلد آخر ، فعليك أن تنقل الصور المنشأة أيضاً إلى نفس الدليل .....

The Counter Program

برنامج العداد

سنتعلم في هذا البرنامج كائن هام في اللغة ، يسمى المؤقت (Timer) ، و أيضاً سنشاهد كيفية تصميم القوائم بالإضافة لبعض العناصر الجديدة ....

فكرة البرنامج : عند تشغيل هذا البرنامج تظهر النافذة الموضحة بالشكل :

حاول أولاً نقر زر البدء دون إدخال أي قيمة ولاحظ ظهور رسالة تنبيه بوجوب إدخال قيمتين ....

أدخل أولاً قيمتين دنيا وقصوى ، ثم أنقر زر **بدء** فتلاحظ ظهور الأرقام في المربع الأبيض ....

غير قيمة شريط التمرير ولاحظ تغير سرعة العد وفقاً لقيمة شريط التمرير التي حددتها ....

❋ أنشئ مشروعاً جديداً واحفظه في المجلد C:\Vb6\MyExercise\Exer3 كما يلي :

النموذج باسم C:\Vb6\MyExercise\Exer3\Counter.Frm

المشروع باسم C:\Vb6\MyExercise\Exer3\Counter.Vbp

صمم النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmCounter
	BorderStyle	1-Fixed Single
	Caption	برنامج العداد
	Icon	C:\VB6\MyIcons\Counter.ico
TextBox	Name	TxtDisplay
	Aligment	2-Center
	Enable	False (1)
	Font Name	Time New Roman
	Font Size	72
VscrollBar	Name	VsBar
	Max	20 " تحديد القيمة القصوى لشريط التمرير "
	Min	1 " تحديد القيمة الدنيا لشريط التمرير "
	Value	5 " تحديد قيمة افتراضية يعمل بها عند بدء التشغيل "
CommandButton	Name	CmdExit
	Caption	خروج

(1) جعلنا خاصية التأهيل غير متاحة كيلا يستطيع المستخدم من الكتابة في مربع النصوص أثناء التنفيذ ...

Object	Properties	Set
		تابع ...
CommandButton	Name	CmdSrart
	Caption	بدء
CommandButton	Name	CmdMin
	Caption	إدخال قيمة دنيا
CommandButton	Name	CmdMax
	Caption	إدخال قيمة قصوى
Label	Name	LblFast
	Caption	سريع
Label	Name	LblSlow
	Caption	بطيء
Timer	Name	Timer1
	Interval	200

❋ تصميم القائمة :

لتصميم شريط قوائم لأي برنامج : ادخل إلى قائمة Tools واختر أمر Menu Editor سيظهر لك مربع حوار ، أحدد فيه القوائم باستخدام خانتي Caption و Name بالإضافة للأسهم الأربعة .....

صمم القائمة File مستعيناً بالجدول التالي :

Caption	Name
ملف	MnuFile
بدء...	MnuStart
خروج...	MnuExit

❋ اكتب شيفرة الخروج كما مر معك سابقاً ....

❋ اكتب الشيفرة التالية في حدث النقر على زر إدخال القيمة الدنيا CmdMin :

```
Min = InputBox ( "إدخال قيمة " , "أدخل قيمة دنيا" )
```

❋ اكتب الشيفرة التالية في حدث النقر على زر إدخال القيمة القصوى CmdMax :

```
Max = InputBox ( "إدخال قيمة " , "أدخل قيمة قصوى" )
```

❋ في القسم العام (General) للنموذج FrmCounter عرف أربع متحولات وذلك كما يلي :

```
Dim Min, Max, View, Num As Integer
```

- المتحولان Min , Max سنسند إليهما القيمتين الدنيا والقصوى اللتان يدخلهما المستخدم ...
- المتحول View سيأخذ إحدى القيمتين (صفر/واحد) وهما للإيعاز ببدء أو إنهاء العرض ....
- المتحول Num تتحول قيمته بين القيمتين الدنيا والقصوى .....

❋ في قسم تحميل النموذج Form_Load() أسند لجميع المتحولات السابقة قيمة الصفر كما يلي :

```
Min = 0
Max = 0
View = 0
```

```
Num = 0
```

❋ اكتب الشيفرة التالية في حدث النقر على زر البدء CmdStart :

```
If Min = 0 And Max = 0 Then
  MsgBox " عليك أولا إدخال قيمتين دنيا وقصوى "
Else
  View = 1
  Num = Min
End If
```

تستخدم حلقة If لاختبار شرط ما ، فإذا تحقق تقوم بتنفيذ الشيفرة التي بداخلها ، وإذا لا .. فتنفذ الشيفرة الموجودة في القسم Else منها ..... ، ووظيفتها في الشيفرة السابقة هي اختبار المتحولين (Min , Max) فإذا كانا مساويان للصفر ، فهذا يعني أن المستخدم لم يقم بعد بإدخال قيمة ، فتظهر رسالة تنبيه ولا ينفذ البرنامج .... ، أما إذا كان المستخدم قد أدخل قيمة ، حينها تنفذ شيفرة Else والتي تسند للمتحول View القيمة / واحد/ وبالتالي تعلم المؤقت ببدء التنفيذ كما سنرى في الشيفرة القادمة .... وأيضاً تسند القيمة الدنيا للمتحول Num الذي سيبدأ العد به ...

❋ اكتب الشيفرة التالية في الحدث **Timer1** :

```
If View = 1 Then
  TxtDisplay.Text = Num
  Num = Num + 1
  If Num = Max + 1 Then
    Num = Min
  End If
End If
```

إن حلقة If الثانية من الشيفرة السابقة تضمن استمرار العد بعد وصول إلى القيمة القصوى ، أما سبب تحديدنا شرط الاختبار بالقيمة (Max + 1) فهو أن زيادة القيمة Num تتم بعد عملية الإسناد لمربع النصوص .....

❋ اكتب الشيفرة التالية لحدث تغير شريط التمرير VsBar :

```
Private Sub VsBar_Change()
    Timer1.Interval = VsBar.Value * 100
End Sub
```

تربط الشيفرة السابقة سرعة تغير قيمة المؤقت بقيمة شريط التمرير التي يغيرها المستخدم .... **فمثلاً** : إذا حرك المستخدم مربع شريط التمرير حتى أصبحت قيمته تساوي / 7 / ، عندها سيسند إلى المؤقت القيمة700 = Interval ... وبالتالي سيتم التغيير كل / 700/ ميلي ثانية .

❋ اكتب الشيفرة التالية في حدث النقر على أمر البدء من القائمة / ملف / ..

```
Private Sub MnuStart_Click()
CmdStart_Click
End Sub
```

تعني الشيفرة السابقة : تنفيذ الشيفرة المكتوبة في حدث النقر على الزر CmdStart وذلك عندما يقوم المستخدم بنقر الأمر / **بدء** / من قائمة / **ملف** / ، وتستخدم مثل هذه الطريقة لتفادي كتابة الشيفرة نفسها أكثر من مرة ...

❋ في حدث النقر على أمر الخروج ((MnuExit)) من قائمة / **ملف** / ، أكتب تعليمة الإنهاء **End** ، ويمكنك أيضاً كتابة الأمر (CmdExit_Click) عوضاً عنها ....

* جعل النموذج يظهر في منتصف الشاشة أثناء التنفيذ :

أنقر في النافذة Form Layout ثم أنقر على صورة النموذج فيها بالزر الأيمن ، واختر منها الخيار Startup Position فتظهر قائمة أخرى اختر منها Center Screen وبهذا سيتم عرض النموذج في منتصف الشاشة عند التنفيذ ....

لاحظت في القائمة السابقة ظهور أربع خيارات للعرض ، ولكل منها وظيفة مبينة في الجدول التالي :

الحالة	الوصف
0-Manual	يدوي : يقوم المستخدم بسحب النموذج إلى المكان المطلوب
1- Center Owner	في منتصف النموذج الذي يتبع له النموذج الحالي (ونستخدمها عند وجود أكثر من نموذج في المشروع)
2- Center Screen	منتصف الشاشة
3- Windows Default	الحالة الافتراضية التي يظهر بها عند أول تشغيل برنامج فيجوال بيسيك

بعد انتهائك من البرنامج واختبار وظائفه والتأكد من صحة العمل .... احفظه ، ثم أجري له نسخة تنفيذية في المسار التالي :

C:\Vb6\MyExercise\Exer3\Counter.exe

الواجهة متعددة الوثائق

Multiple Documents Interface (MDI)

إن جميع البرامج الشهيرة كبرامج .... - Word – Excel –Corel هي برامج متعددة الوثائق ، أي أنها عبارة عن نموذج أب وحيد يضم بداخله عدد من النماذج الفرعية والتي تسمى بالنماذج الأبناء .....

البرنامج الذي سننشئه الآن مشابه لهذه البرامج ... حيث تظهر واجهة أساسية تحوي بداخلها ثلاث نماذج أبناء ، ويمكنك عن طريق القوائم التحكم بطريقة عرضها (تجانب أفقي أو عمودي أو تتالي ...)

أنشئ مشروعاً جديداً ... وقم بحذف النموذج الموجود ، لأننا لسنا بحاجته حيث سنضيف نموذج أب وهو من نوع آخر غير النوع الذي نتعامل معه عادةً ....

- لحذف النموذج الحالي (Form1) : اختر الأمر Remove Form1 من قائمة Project ...
- لإضافة النموذج الأب ، اختر الأمر Add Mdi Form من قائمة Project يظهر مربع حوار ، أنقر منه زر **Open** .. [1]

احفظ المشروع بحيث :

النموذج باسم C:\Vb6\MyExercise\Exer4\Mdi.Frm ...

المشروع باسم C:\Vb6\MyExercise\Exer4\Mdi.Vbp ...

❋ أضف نموذج جديد واحفظه باسم Child1.Frm في نفس دليل العمل ، ولتحويل النموذج الجديد إلى نموذج ابن :

ادخل إلى خصائصه وغير الخاصية MdiChild إلى True ....

❋ أضف نموذجين جديدين واحفظهما على التوالي باسم Child2.Frm و Child3.Frm وحولهما أيضاً إلى نماذج أبناء بنفس الطريقة السابقة ....

❋ صمم النموذج الأول Child1 وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmChild1
	Caption	الابن الأول
	RightToLeft	True
CommandButton	Name	CmdClose
	Caption	إغلاق
PictureBox	Name	Picture1
	AutoSize	True تجعل حجم صندوق الصورة يتغير وفقا لحجم الصورة
	Picture	C:\VB6\MyPhotos\Child1.Bmp

[1] لا يمكن أن يحتوي البرنامج ذات الوثائق المتعددة إلا على نموذج أب واحد فقط ، بينما يمكن أن يحوي عدة نماذج أبناء ... ولتتأكد من ذلك حاول أن تضيف نموذج أب آخر من قائمة Project .... وستلاحظ أنك لن تستطيع ذلك ... !!

صمم قائمة للنموذج كما يلي :

Caption	Name
ملف	MnuFile
إغلاق النموذج ...	MnuClose

❋ اكتب الشيفرة التالية في حدث النقر على زر الإغلاق للنموذج :

```
Unload Frmchild1
```

❋ اكتب الشيفرة السابقة نفسها في حدث النقر على أمر القائمة / إغلاق النموذج MnuClose / ...

إن الشيفرة السابقة تقوم بإلغاء تحميل النموذج من الذاكرة ، أي تغلقه نهائياً .....

❋ صمم النموذج الثاني Child2 وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmChild2
	Caption	الابن الثاني
	RightToLeft	True
CommandButton	Name	CmdClose
	Caption	إغلاق
PictureBox	Name	Picture1
	AutoSize	True
	Picture	C:\VB6\MyPhotos\Child2.Bmp

❋ اكتب شيفرة حدث النقر على زر الإغلاق كما في النموذج السابق ....

❋ صمم النموذج الثاني Child2 وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmChild3
	Caption	الابن الثالث
	RightToLeft	True
CommandButton	Name	CmdClose
	Caption	إغلاق
PictureBox	Name	Picture1
	AutoSize	True
	Picture	C:\VB6\MyPhotos\Child3.Bmp

❋ اكتب شيفرة حدث النقر على زر الإغلاق كما في النموذج السابق ....

تصميم النموذج الأب :

❋ عد إلى النموذج الأب ونفذه كما الجدول التالي :

Object	Properties	Set
MdiForm	Name	FrmMdi
	Caption	الوثائق المتعددة
	RightToLeft	True

❋ صمم قوائم النموذج كما بالجدول التالي :

Caption	Name
ملف	MnuFile
خروج...	MnuExit
عرض	MnuView
الابن الأول...	MnuViewChild1
الابن الثاني...	MnuViewChild2
الابن الثالث...	MnuViewChild3
-...	MnuLine تعطي خط فاصل
عرض الكل...	MnuViewAll
إخفاء الكل...	MnuHideAll

كتابة شيفرات أوامر قائمة عرض

❋ اكتب الشيفرة التالية في حدث النقر على أمر / الابن الأول – MnuViewChild1 / من قائمة **عرض**

```
FrmChild1.Show
```

❋ اكتب الشيفرة التالية في حدث النقر على أمر / الابن الثاني – MnuViewChild2 / من قائمة **عرض**

```
FrmChild2.Show
```

❋ اكتب الشيفرة التالية في حدث النقر على أمر / الابن الثالث – MnuViewChild3 / من قائمة **عرض**

```
FrmChild3.Show
```

❋ اكتب الشيفرة التالية في حدث النقر على أمر / عرض الكل – MnuViewAll / من قائمة **عرض**

```
FrmChild1.Show
FrmChild2.Show
FrmChild3.Show
```

❋ اكتب الشيفرة التالية في حدث النقر على أمر / إخفاء الكل – MnuHideAll / من قائمة **عرض**

```
FrmChild1.Hide
FrmChild2.Hide
FrmChild3.Hide
```

تجريب البرنامج

لبدء تجريب البرنامج عليك أولاً الدخول إلى قائمة Project واختيار الأمر Project Properties فيظهر مربع حوار ، حدد في خانة Startup Object النموذج الواجب تشغيله أول بدء التنفيذ وهو النموذج FrmMdi .... بعدها نفذ البرنامج **ولاحظ** : أنه عندما يكون نموذج الابن الأول هو الفعال ، تختفي القائمتان الخاصتان بالنموذج الأب ، وتظهر القائمة الخاصة بالنموذج الابن الأول ، وسبب ذلك أن الابن الأول يحوي على قائمة خاصة فيه ، أما إذا كان أحد النموذجين (الابن الثاني أو الثالث) هو الفعال ، فتظهر القائمتان الخاصتان بالنموذج الأب ....

* الآن سنقوم بإنشاء قائمة // **إطار** // للنموذج الأب ، ولكن لكي نجعل هذه القائمة تظهر عند تفعيل نموذج الابن الأول ، سنقوم بإضافتها أيضاً له ...

* ادخل إلى قوائم النموذج الأب وأضف له قائمة // **إطار** // ، واجعل اسمها (Name) = MnuWindow ... تلاحظ وجود خانة Window List في مربع الحوار السابق ، لذلك قم بوضع إشارة تحقق أمام هذه الخانة [1]

* قم بإضافة نفس القائمة إلى النموذج الابن الأول وبنفس الخواص (تأكد من تفعيل خانة Window List أيضاً ..) .

أضف قائمة (ترتيب) للنموذج الأب وفق الجدول التالي :

Caption	Name
ترتيب	MnuArrange
تتالي ...	MnuCascade
تجانب أفقي ...	MnuTile
تجانب عمودي ...	MnuVertical
ترتيب أيقونات ...	MnuIcon

* أضف نفس القائمة السابقة للنموذج الابن الأول ....

```
*كتابة شيفرات أوامر قائمة ( ترتيب ) . . .
```

[[بعد أن تكتب هذه الشيفرات للنموذج الأب ، أكتبها نفسها لنموذج الابن الأول]]

اعتمد في كتابتك للشيفرات على الجدول التالي :

عنوان الأمر	اسمه	الشيفرة
تتالي	MnuCascade	FrmMdi.Arrange vbCascade
تجانب أفقي	MnuTile	FrmMdi.Arrange vbTileHorizontal
تجانب عمودي	MnuVertical	FrmMdi.Arrange vbTileVertical
ترتيب أيقونات	MnuIcon	FrmMdi.Arrange vbArrangeIcons

معتمداً على نفسك والجدول السابق .... اكتب شيفرات قائمة ترتيب في نموذج الابن الأول ...

* احفظ المشروع ونفذه وتأكد من عمله بشكل صحيح ، ثم أنشئ نسخة تنفيذية منه ، واحفظ الملف التنفيذي الجديد في الدليل الحالي لعملك ..

C:\Vb6\MyExercise\Exer4\Mdi.Exe

[1] إن هذه الخاصية تستخدم فقط في برامج النماذج متعددة الوثائق ، وعند التنفيذ يظهر فيها عناوين النماذج الأبناء الموجودة في النموذج الأب ومن خلال هذه القائمة تستطيع التنقل بين النماذج الأبناء المفتوحة ...

برنامج المفكرة

The Notepad Program

إن البرنامج الذي سننشئه يشبه تقريباً برنامج المفكرة الذي يأتي مع نظام التشغيل Windows ، حيث سيقوم البرنامج بإنشاء مستندات نصية جديدة عندما نطلب منه ذلك ، وسيقوم أيضاً بإظهار عناوين مختلفة لكل مستند جديد ننشئه ، حيث تظهر عناوين المستندات بالأسماء : Doc1 , Doc2 , Doc3....Etc .

أنشئ مشروعاً جديداً واحفظه في المجلد Exer5 بحيث :

النموذج باسم C:\Vb6\MyExercise\Exer5\Document.Frm

المشروع باسم C:\Vb6\MyExercise\Exer5\Notepad.Vbp

صمم النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmDoc
	Caption	Doc1
	MdiChild	True
TextBox	Name	Txtdoc
	MultiLine	True
	ScrollBar	3-Both
	Text	اجعله فارغاً

❋ اختر الأمر Add MdiForm من قائمة Project .... ثم احفظ النموذج الجديد باسم Notepad.Frm وهذا النموذج سيصبح النموذج الأب للنموذج السابق ... صممه كما الجدول التالي :

Object	Properties	Set
MdiForm	Name	FrmNotepad
	Caption	المفكرة

❋ نفذ البرنامج ولاحظ أن مربع النص الذي أنشأناه لا يظهر كاملاً ضمن النموذج ، ولجعلها كذلك سأضيف الشيفرة التالية إلى الإجراء Form_Resize () في النموذج FrmDoc :

```
Me.TxtDoc.Height = Me.ScaleHeight
Me.TxtDoc.Width = Me.ScaleWidth
```

إن الإجراء Form_Resize () ينفذ عندما يقوم المستخدم بتغيير حجم النموذج .... أما عبارة **ME** فهي كلمة محجوزة تستبدل أثناء تنفيذ البرنامج باسم النموذج الذي تنفذ داخله الشيفرة ....

❋ نفذ البرنامج ولاحظ أنه كلما غيرت حجم النموذج تغير حجم مربع النص ليكون مطابقاً دوماً لحجم النموذج .

❋ أضف القائمة التالية للنموذج FrmNotepad :

Caption	Name
File	MnuFile
....New	MnuNew
....Exit	MnuExit

* اكتب الشيفرة التالية في حدث النقر على أمر New من قائمة File :

```
Dim NewChild As New FrmDoc
NewChild.Show
```

يتم في الشيفرة السابقة أولاً تعريف متحول جديد وهذا المتحول من نوع النموذج الذي قمنا نحن بإنشاءه ، وبالتالي سيأخذ النموذج الجديد جميع مواصفات النموذج FrmDoc ، بعدها تقوم الشيفرة بإظهار النموذج الجديد ...

* لجعل عناوين النماذج الجديدة تتغير عند كل نموذج جديد ، عدل شيفرة القائمة New كما يلي :

```
Dim NewChild As New FrmDoc
Static DocNum
If DocNum = 0 Then
   DocNum = 1
End If
DocNum = DocNum + 1
NewChild.Show
NewChild.Caption = "Doc " + CStr (DocNum)
```

إن التابع CStr يحول القيمة الرقمية التي يشير إليها المتحول إلى قيمة نصية ....

* **إنشاء شريط أدوات للبرنامج :** قم بإضافة زر أمر جديد للنموذج الأب ، ستلاحظ ظهور رسالة خطأ .... مفادها أنه لا يمكنك إضافة أي كائن جديد إلى النموذج ذات الوثائق المتعددة إلا إذا قمت بإضافتها ضمن شريط أدوات ... لذلك سنضيف أولاً شريط أدوات كما يلي :

أضف صندوق صور إلى النموذج الأب ، ولاحظ ظهوره في أعلى النموذج ....

* أضف زر أمر جديد للخروج وذلك بأن ترسم هذا الزر بالماوس ضمن صندوق الصور الذي أنشأته ، بعدها اكتب فيه شيفرة الخروج ...

* **إنشاء زر أمر من صنعك :** سنقوم الآن بصنع زر أمر خاص ، ولكن قبل البدء عليك إنشاء صورتين في برنامج الرسام وتخزينهما في دليل العمل الحالي كما بالشكل :

New2.bmp

New1.bmp

بعدها أضف للنموذج ثلاث كائنات صور (Image)

(**لا تنسى أن ترسم الصور بالماوس وليس نقراً مزدوجاً**) وفق الجدول التالي :

Object	Properties	Set
Image	Name	Img1
	Picture	C:\VB6\MyPhotos\New1.Bmp
	Visible	False
Image	Name	Img2
	Picture	C:\VB6\MyPhotos\New2.Bmp
	Visible	False
Image	Name	Img3
	Picture	C:\VB6\MyPhotos\New3.Bmp
	Stretch	True
	Visible	True

إن خاصية Visible تتحكم بعرض أو إخفاء الكائن أثناء تنفيذ البرنامج ....

❋ اكتب الشيفرة التالية في الإجراء Mouse Down التابع للكائن Img3 :

```
Img3.Picture = Img2.Picture
```

❋ اكتب الشيفرة التالية في الإجراء Mouse Up التابع للكائن Img3 :

```
Img3.Picture = Img1.Picture
```

❋ اكتب الشيفرة التالية في حدث النقر على الصورة Img3 :

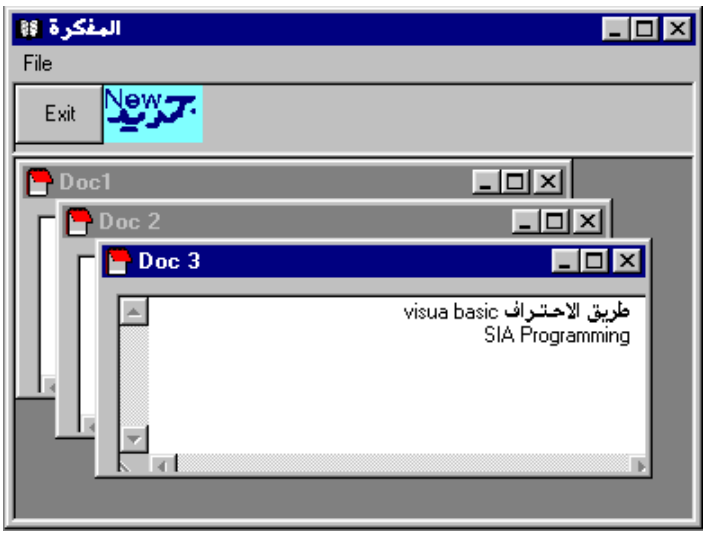
```
MnuNew_Click
```

بذلك تكون قد انتهيت من البرنامج ، احفظه واصنع منه نسخة تنفيذية واحفظها بالدليل الحالي :

C:\Vb6\MyExercise\Exer5\MyNotepad.Exe

الذي أنشأته قريب من برنامج المفكرة الأساسي في النظام Windows ، إلا أنك لا تستطيع الاحتفاظ بالملف

أو تنسيقه أو الفتح أو .... وسيصبح بمقدورك إنجاز هذه العمليات لاحقاً عندما نتعرف وإياك على إجراءات الحفظ والنسخ والنقل والفتح و .. و ..



أشرطة القوائم والأدوات والحالة

The Menus - Tools & Statute Bars

سنتعلم هنا كيفية إنشاء أشرطة الأدوات والحالة بالإضافة لأشرطة القوائم ....

عند تشغيل البرنامج تظهر نافذة فيها أربع أزرار أدوات ، وهي لتلوين النموذج والخروج منه ، وعندما تضع المؤشر على أي زر تلاحظ ظهور معلومات عنه في شريط الحالة ...

التنفيذ : أنشئ مشروعاً جديداً واحفظه في المجلد Exer6 بحيث :

النموذج باسم C:\Vb6\MyExercise\Exer6\MenuTool.Frm

المشروع باسم C:\Vb6\MyExercise\Exer6\MenuTool .Vbp

صمم النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmMenu
	Caption	أشرطة الأدوات والحالة
	BorderStyle	1-Fixed Single
	RightToLeft	True
PictureBox	Name	PicTool
	Align	1-Align Top
PictureBox	Name	PicStatute
	Align	2-Align Bottom
Image " ارسمها في مربع الصورة العلوي "	Name	ImgRed
	Picture	C:\VB6\MyPhotos\Red.Bmp
	Stretch	True
Image " ارسمها في مربع الصورة العلوي "	Name	ImgBlue
	Picture	C:\VB6\MyPhotos\Blue.Bmp
	Stretch	True
Image " ارسمها في مربع الصورة العلوي "	Name	ImgGreen
	Picture	C:\VB6\MyPhotos\Green.Bmp
	Stretch	True
Image " ارسمها في مربع الصورة العلوي "	Name	ImgExit
	Picture	C:\VB6\MyPhotos\Exit.Bmp
	Stretch	True
Image	Name	ImgExitDn
	Picture	C:\VB6\MyPhotos\ExitDn.Bmp
	Visible	False
Image	Name	ImgExitUp
	Picture	C:\VB6\MyPhotos\Exit.Bmp
	Visible	False
Label " ارسمه في مربع الصورة السفلي "	Name	LblRed
	Caption	أنقر هنا لاختيار اللون الأحمر
	Visible	False
Label " ارسمه في مربع الصورة السفلي "	Name	LblBlue
	Caption	أنقر هنا لاختيار اللون الأزرق
	Visible	False
		تابع.....

Object	Properties	Set
Label " ارسمه في مربع الصورة السفلي "	Name	LblGreen
	Caption	أنقر هنا لاختيار اللون الأخضر
	Visible	False
Label " ارسمه في مربع الصورة السفلي "	Name	LblExit
	Caption	أنقر هنا للخروج
	Visible	False
CheckBox	Name	ChkColorMenu
	Caption	عرض قائمة الألوان
	Value	1-Checked

* لقد جعلنا خاصية Visible للصورتين ImgExitDn و ImgExitUp تحمل القيمة False أي لن يظهرا عند التنفيذ ، ولكن الفائدة منهما هي جعل زر الخروج يتغير عند النقر عليه بحيث يبدو مضغوطاً ... ، وكنا نستطيع تطبيق هذه العملية على باقي الأزرار بأن نضيف صورتين جدد لكل صورة ، لكن للاختصار لم نقم بذلك (**بإمكانك التجريب**) .

* صمم قوائم النموذج كما الجدول التالي :

Caption	Name
ملف	MnuFile
خروج....	MnuExit
ألوان	MnuColor
أحمر....	MnuRed
أزرق....	MnuBlue
أخضر....	MnuGreen
عرض	MnuView
شريط الأدوات....	MnuTool " اجعل صندوق الاختبار Checked لها فعالاً "
شريط الحالة....	MnuStatute " اجعل صندوق الاختبار Checked لها فعالاً "

بهذا نكون انتهينا من البرمجة المرئية ، بقي لدينا كتابة الشيفرات ، ولأن الكائنات التي لدينا كثيرة ، سنعتمد في كتابة الشيفرات على الجداول .....

* **اعتمد** على الجدول التالي واكتب جميع الشيفرات الواردة فيه في حدث النقر Click :

Object	Code	Mark
ImgRed	FrmMenu.BackColor = QBColor(12)	الرقم 12 يعطي اللون الأحمر
	MnuRed.Enabled = False	يصبح الأمر "أحمر " من القائمة غير متاح
	MnuBlue.Enabled = True	
	MnuGreen.Enabled = True	
ImgBlue	FrmMenu.BackColor = QBColor(1)	
	MnuRed.Enabled = True	
	MnuBlue.Enabled = False	
	MnuGreen.Enabled = True	
ImgGreen	FrmMenu.BackColor = QBColor(2)	
	MnuRed.Enabled = True	
	MnuBlue.Enabled = True	
	MnuGreen.Enabled = False	
ImgExit	End	

Object	Code	Mark
MnuRed	ImgRed_click	أي تنفيذ نفس الشيفرة الواردة في نقر الصورة
MnuBlue	ImgBlue_click	
MnuGreen	ImgGreen_click	
MnuExit	End	

ChkColorMenu	If ChkColorMenu.Value = 0 Then
	MnuColor.Visible = False
	Else
	MnuColor.Visible = True
	End If
		التحكم بعرض أو إخفاء قائمة ‘‘ألوان"

MnuTool	If MnuTool.Checked = True Then
	PicTool.Visible = False
	MnuTool.Checked = False
	Else
	PicTool.Visible = True
	MnuTool.Checked = True
	End If
		إذا كان ظاهراً يخفيه ، وإذا مخفياً فيظهره

MnuStatute	If MnuStatute.Checked = True Then
	PicStatute.Visible = False
	MnuStatute.Checked = False
	Else
	PicStatute.Visible = True
	MnuStatute.Checked = True
	End If	

* **اعتمـــــد** علـــى الجـــدول التـــالي واكتـــب جميـــع الشـــيفرات الـــواردة فيـــه في حـــدث تمريـــر المـــاوس فـــوق الكائن Mouse_Move :

Object	Code	Mark
ImgRed	LblRed.Visible = True LblExit.Visible = False LblGreen.Visible = False LblBlue.Visible = False	إظهار النص الخاص باللون الأحمر فقط وإخفاء الباقي
ImgBlue	LblBlue.Visible = True LblRed.Visible = False LblGreen.Visible = False LblExit.Visible = False	
ImgGreen	LblGreen.Visible = True LblExit.Visible = False LblRed.Visible = False LblBlue.Visible = False	
ImgExit	LblExit.Visible = True LblRed.Visible = False LblBlue.Visible = False LblGreen.Visible = False	

بقي مرحلة قصيرة ، وهي جعل مظهر زر الخروج يتغير عند النقر عليه ، لذلك .... **تـــابع** .

❋ اكتب الشيفرة التالية في حدث (**استمرار النقر** [1]) على صورة ImgExit (الحدث Mouse _Down)

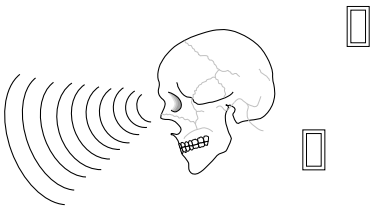
```
ImgExit.Picture = ImgExitDn.Picture
```

❋ اكتب الشيفرة التالية في حدث تحرير الماوس عن الصورة (الحدث Mouse Up)

```
ImgExit.Picture = ImgExitUp.Picture
```

جرب البرنامج وتأكد من عمله بشكل صحيح ، ثم اجري له نسخة تنفيذية ،واحفظها في المجلد Exer6

C:\Vb6\MyExercise\Exer6\MenuTool.Exe



1- عند اختيارك للون معين ، فإنه يصبح غير متاح في القوائم ..

2- تستطيع التحكم بعرض و إخفاء قائمة ألوان عن طريق مربع الاختيار " عرض قائمة الألوان " ...

3- ظهور تلميح عن الزر الذي تضع الماوس فوقه (دون نقر) ، في شريط الحالة ...

4- إمكانية التحكم بعرض شريطي الأدوات والحالة من قائمة "عرض ".

[1] وهو النقر على الصورة دون تحرير زر الماوس ....

7 الحلقات والأحداث المعلقة

Loops & Pending Event‘s

الحدث المعلق هو الأمر الموجود في وحدة معالجة الحاسب (CPU) ، والذي لم ينفذ بعد ... فمثلاً : إذا طلبت من الحاسب أن يطبع على نموذج ما الأعداد من (1 – 10000) ، وأثناء تنفيذه للطباعة أعطيته أمر بإصدار صوت معين فتلاحظ أنه لن يصدر الصوت إلا بعد انتهائه من تنفيذ أمر الطباعة ... وبالتالي نسمي أمر إصدار الصوت بالحدث المعلق ، ولكن بإمكانك أن تجعله يستجيب لأي أمر تطلبه مهما كانت الوظيفة التي يقوم بها كبيرة ، وستلاحظ فائدة ذلك عند تصميمك لهذا البرنامج ، كما ستتعلم كيفية استخدام حلقات التكرار في برامجك .....

يقوم برنامجنا الأول برسم عدد لا نهائي من النقاط وبألوان مختلفة ... وعند نقرك على زر التوقف ، ينتهي رسم النقاط ، وهنا تكمن فكرة تنفيذ الأحداث المعلقة ...

أنشئ مشروعاً جديداً واحفظه في المجلد Exer7 بحيث :

النموذج باسم C:\Vb6\MyExercise\Exer7\Loop.Frm

المشروع باسم C:\Vb6\MyExercise\Exer7\Loop.Vbp

❋ نفذ النموذج وفق الجدول :

Object	Properties	Set
Form	Name	FrmLoop
	Caption	الحلقات
	BorderStyle	1-Fixed Single
CommandButton	Name	CmdStart
	Caption	بدء
CommandButton	Name	CmdStop
	Caption	إيقاف
CommandButton	Name	CmdExit
	Caption	خروج

❋ أولاً عرف المتحول Pause في قسم التصريح العام للنموذج كما يلي :

```
Option Explicit
Dim Pause
```

المتحول السابق سيأخذ قيمتين فقط :

1- إذا أخذ قيمة (الواحد) : سيوعز للبرنامج بإنهاء عملية الرسم ..

2- إذا أخذ قيمة الصفر: سيوعز للبرنامج ببدء عملية الرسم ..

❋ اكتب شيفرة الخروج في حدث النقر على زر الخروج CmdExit

❋ اكتب الشيفرة التالية في حدث النقر على زر الإيقاف CmdStop :

```
Pause =1
```

❋ اكتب الشيفرة التالية في حدث النقر على زر البدء CmdStart :

```
Private Sub CmdStart_Click()
1-)     Dim X, Y, Counter
2-)     CurrentX = ScaleWidth / 2
3-)     CurrentY = ScaleHeight / 2
4-)     FrmLoop.DrawWidth = 15
5-)     Pause = 0
6-)     For Counter = 1 To 1000
7-)         X = Rnd * ScaleWidth
8-)         Y = Rnd * ScaleHeight
9-)         PSet (X, Y), QBColor(Rnd * 15)
10-)    If Counter = 999 Then
11-)        Counter = 1
12-)    End If
13-)    If Pause = 1 Then
14-)        Exit For
15-)    End If
16-)    DoEvents
17-)Next Counter
End Sub
```

شرح الشيفرة السابقة

رقم التعليمة	وظيفتها
2 و 3	تحديد إحداثيات النقطة بأن يكون سينها هو نصف عرض النموذج ، وعينها هو نصف ارتفاعه ....
4	تحديد عرض النقطة التي سيتم رسمها بـ 15
6	بداية حلقة FOR بالتنفيذ وهي تستمر حتى التعليمة 17 ، وهي تنفذ الشيفرة الموجودة ضمنها (1000) مرة ...
7	جعل قيمة سينات النقطة تتحول بقيم عشوائية بين الصفر وعرض النموذج ، إذ أن التابع RND يعطي قيم عشوائية بين الصفر والواحد ، وبما أننا ضربناه بالقيمة ScaleWidth ، فستتراوح قيمته بين الصفر وعرض النموذج ....
9	تابع PSet يرسم النقطة في النموذج وفق الإحداثيات المحددة وبالألوان المحددة في الوسيط الثاني له ، و أيضاً وفق التابع RND ستتحول قيم الألوان بقيم عشوائية بين الصفر و 15 ....
12–11–10	حلقة IF لجعل التنفيذ لا نهائي ...
15–14–13	تختبر قيمة المتحول Pause فإذا كان مساوياً للواحد يتم الخروج من حلقة IF ، ويأخذ المتحول Pause قيمة الواحد عند نقر زر الإيقاف كما كتبنا في شيفرته ((راجع شيفرة الزر CmdStop))
16	تعليمة DoEvents هي التي تسمح بتنفيذ الأحداث المعلقة ، فبعد أن يرسم الحاسب كل نقطة يصل إلى هذه التعليمة ، فيذهب بموجبها إلى وحدة المعالجة CPU فإذا وجد فيها أي أمر ينفذه ثم يتابع تنفيذ حلقة FOR ...

برنامجك ولاحظ أنه يمكنك إيقافه بمجرد النقر على زر / إيقاف / ... ماكان سيحدث لو أنك لم تضع عبارة DoEvents في حلقة For ؟

بفرض أنك لم تضعها ، فستلاحظ أن البرنامج سيعمل ... ولكنه لن يمكنك إيقافه مهما نقرت على زر الإيقاف ....

البرنامج وأصنع منه نسخة تنفيذية بعد تجريبه ، واحفظ الملف التنفيذي في المجلد Exer7

C:\Vb6\MyExercise\Exer7\Loop.Exe

فكرة البرنامج الثاني مشابهة تقريباً للأول ، لكن يوجد فيه شريط تمرير للتحكم بحجم النقاط ، وأيضاً بوجود شكل آخر للحلقات ، ورسالة تأكيد عند الخروج ....

التنفيذ : أنشئ مشروعاً جديداً واحفظه في نفس المجلد Exer7 بحيث :

النموذج باسم C:\Vb6\MyExercise\Exer7\BestLoop.Frm

المشروع باسم C:\Vb6\MyExercise\Exer7\BestLoop.Vbp

نفذ النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmBestLoop
	Caption	Best Loop
	BorderStyle	1-Fixed Single
	ControlBox	False
CommandButton	Name	CmdStart
	Caption	Start
CommandButton	Name	CmdStop
	Caption	Stop
CommandButton	Name	CmdExit
	Caption	Exit
HscrollBar	Name	HsBar
	Max	75
	Min	1
	Value	10

❋ أولاً عرف المتحول Pause في قسم التصريح العام للنموذج كما يلي :

```
Option Explicit
Dim Pause
```

❋ اكتب الشيفرة التالية في حدث النقر على زر الإيقاف CmdStop :

```
Pause =1
```

❋ اكتب الشيفرة التالية في حدث النقر على زر البدء CmdStart :

```
Dim X, Y, Counter
    CurrentX = ScaleWidth / 2
    CurrentY = ScaleHeight / 2
    FrmBestLoop.DrawWidth = HsBar.Value
    Pause = 0
    Do While DoEvents()
        X = Rnd * ScaleWidth
        Y = Rnd * ScaleHeight
        PSet (X, Y), QBColor(Rnd * 15)
     If Counter = 999 Then
        Counter = 1
     End If
     If Pause = 1 Then
        Exit Do
     End If
 Loop
```

أن الشيفرة السابقة مشابهة تقريباً للتي كتبناها في البرنامج الأول ، مع بعض الاختلافات وهي :

- استعملنا هنا حلقة Do While وهي تعني ((**طالما الشرط محقق**)) ، ويتم إنهاء هذه الحلقة بعبارة Loop ، وأيضاً استخدمنا عبارة DoEvents() ، وهي تختلف عن العبارة التي استخدمناها في البرنامج السابق ، بأنها هنا تعبر عن تابع ، وهذا التابع يرجع قيمة (**واحد**) طالما أن النموذج مفتوحاً ، وقيمة (**صفر**) عندما يغلق النموذج .. و***باختصار*** : فإن عبارة Do while السابقة تعني (" **نفذ محتويات الحلقة طالما النموذج مفتوحاً ، مع مراعاة تنفيذ الأحداث المعلقة** ") .
- حددنا عرض نقطة الرسم بالعبارة ((FrmBestLoop.DrawWidth = HsBar.Value)) ، وذلك كي نضمن تغير حجم نقطة الرسم عندما يقوم المستخدم بتغيير قيمة شريط التمرير ...

** البرنامج ولاحظ ما يلي :

أنقر زر Start ولاحظ بدء رسم النقاط بالحجم /10/ ...

غير قيمة شريط التمرير ، ... لا بد أنك لاحظت أن النقاط بقيت بحجمها ولم تتغير ...

انقر على زر Start مجدداً ، ولاحظ تغير حجم النقاط إلى الحجم الجديد ... ولجعل حجم النقاط يتغير مباشرة عند تغير قيمة شريط التمرير ، اكتب الشيفرة التالية في حدث تغيير قيمة الشريط :

```
Private Sub HsBar_Change()
Cls
FrmBestLoop.DrawWidth = HsBar.Value
End Sub
```

* إظهار رسالة تأكيد عند الخروج من البرنامج بنقر زر **Exit** :

عدل شيفرة نقر الزر Exit كما يلي :

```
Dim Test
Test = MsgBox("هل حقاً تريد الخروج", vbYesNo + vbMsgBoxRtlReading + vbMsgBoxRight )
If Test = 6 Then
   End
End If
```

* إن تابع MsgBox في الشيفرة السابقة أخذ وسيطين :

الأول : هو نص الرسالة المطلوب ظهورها ....

الثاني : هو تكوين لثلاث وسطاء :

1. vbYesNo : يظهر في مربع الرسالة زري أمر فقط هما (نعم – لا) .
2. vbMsgBoxRtlReading : يجعل اتجاه أزرار الأمر مطابق للغة العربية . (بإمكانك الاستعاضة عنها بكتابة الرقم 1048576 مكانها)
3. vbMsgBoxRight : يجعل الكتابة في المربع تظهر من اليمين لليسار . (استعض عنها بالرقم 524288)

* لاحظت أخيراً وجود شيفرة الاختبار (إذا كانت قيمة المتحول = 6 ، فيتم الخروج) ، ويعود ذلك إلى أن تابع MsgBox يعيد قيمة بحسب الزر الذي نقره المستخدم ، ونقر الزر / نعم / يؤدي لإعادة القيمة / 6 / ...

البرنامج وأصنع منه نسخة تنفيذية بعد تجريبه ، واحفظ الملف التنفيذي في المجلد Exer7

C:\Vb6\MyExercise\Exer7\BestLoop.Exe

التحكم بالملفات

File Control

سنتعلم الآن كيفية التحكم بالملفات ، والوصول إلى عمليات الفتح والحفظ والاختيار ....

عندما تشغل البرنامج تظهر واجهة البرنامج الأساسية ، وباستخدام القوائم يمكنك اختيار ملف ما من أي مجلد موجود على إحدى السواقات الموجودة لديك ...

التنفيذ : أنشئ مشروعاً جديداً واحفظه في المجلد Exer8 بحيث :

النموذج باسم C:\Vb6\MyExercise\Exer8\FileSelect.Frm

المشروع باسم C:\Vb6\MyExercise\Exer8\ FileSelect Vbp

: سنتعامل في هذا البرنامج مع كائن جديد وهو Common Dialog وهذا الكائن غير موجود في صندوق الأدوات لذلك سأقوم بإضافته كما يلي :

أنقر باليمين على صندوق الأدوات واختر الأمر Components ، فيظهر مربع حوار أنقر منه على Browse فيظهر مربع حوار آخر لتحديد ملف Ocx لذلك أنقر على الملف Comdlg32.ocx ثم أنقر **فتح** ثم **موافق** ...[2]

النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmFileSelect
	Caption	التحكم بالملفات
	BorderStyle	1-Fixed single
	Picture	C:\VB6\MyPhotos\FileSelect
CommomDialog	Name	CommDial
	CancelError	True [1]

صمم قوائم النموذج وفق الجدول التالي :

Caption	Name
ملف	MnuFile
اختيار ملف	MnuSelect
خروج....	MnuExit

* اكتب شيفرة الخروج للقائمة ((خروج – MnuExit))

---

[1] هذه الخاصية تجعل النقر على زر // إلغاء الأمر // يؤدي لحدوث خطأ ، وسترى فائدة ذلك أثناء شيفرة القائمة MnuSelect

[2] يوجد هذا الملف افتراضياً في مجلد النظام C:\Windows\System

❋ اكتب الشيفرة التالية في حدث النقر على القائمة MnuSelect :

```
On Error GoTo WhenError
CommDial.Filter = "All Files (*.*)  |*.*|  " +   "Text Files (*.txt)  |*.txt|  " + _
                  "Doc Files (*.doc) |*.doc|  "
CommDial.FilterIndex = 1
CommDial.ShowOpen
MsgBox ": لقد اخترت الملف     " + CommDial.filename, 524288 + 1048576, "اختيار"
Exit Sub
WhenError :
MsgBox "لم تختر شيء  "  , 524288 + 1048576, "خطأ"
Exit Sub
```

في الشيفرة السابقة : نعلم فيجوال بيسيك أولاً أنه عند حدوث الخطأ يجب ينتقل إلى العنوان WhenError ، والخطأ سيحدث إذا لم يقم المستخدم باختيار أي ملف ...

بعدها نحدد أنواع الملفات التي نريد ظهورها ..

بعدها جعلنا القيمة الافتراضية للأنواع التي ستظهر هي All Files أي كل الملفات (وهو النوع الأول)

بعدها استدعينا مربع حوار الفتح وذلك باستخدام المنهج ShowOpen

(❋) هناك أيضاً مناهج أخرى تستخدم مع كائن Common Dialog هي :

المنهج	وظيفته	المنهج	وظيفته
ShowOpen	يظهر مربع حوار الفتح	ShowSave	يظهر مربع حوار الحفظ
ShowColor	يظهر مربع حوار اختيار لون	ShowPrinter	يظهر مربع حوار تجهيز الطابعة
ShowFont	يظهر مربع حوار اختيار خط	ShowHelp	يظهر مربع حوار المساعدة

❋ احفظ المشروع وجربه وتأكد من عمله ثم اصنع منه نسخة تنفيذية في المجلد Exer8

C:\Vb6\MyExercise\Exer8\FileSelect.Exe

لقد استخدمنا في البرنامج السابق مربع حوار جاهز من نظام Windows وهو مربع حوار الفتح ، أما الآن فسنقوم بتصميم مربعات حوار خاصة بنا وذلك باستخدامنا لكائنات جديدة من صندوق الأدوات هي :

- DriveListBox : يستخدم هذا الكائن لإظهار أسماء السواقات الموجودة واختيار واحدة منها .
- DirListBox : يستخدم هذا الكائن لإظهار أسماء المجلدات الموجودة والدخول إلى مجلد معين .
- FileListBox : يستخدم هذا الكائن لإظهار أسماء الملفات في مجلد ما ، واختيار ملف منها .

برنامجنا الجديد بإظهار مربع حوار يمكنك من خلاله مشاهدة ملفات الصور الموجودة .

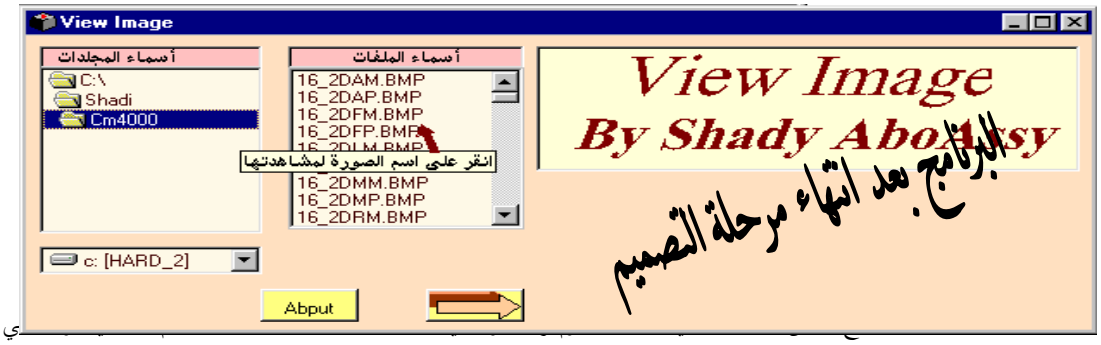
التنفيذ : أنشئ مشروعاً جديداً واحفظه في المجلد Exer8 بحيث :

النموذج باسم C:\Vb6\MyExercise\Exer6\ViewImage.Frm

المشروع باسم C:\Vb6\MyExercise\Exer6\ ViewImage .Vbp

* نفذ النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmViewImage
	Caption	View Image
DirListBox	Name	DirName
	ToolTipText	اختر من هنا اسم المجلد الذي يحوي الصور
DriveListBox	Name	DriveName
FileListName	Name	FileName
	ToolTipText	أنقر على أي صورة لمشاهدتها
PictureBox	Name	PicView
	AutoSize	True
Label	Name	LblDir
	BorderStyle	1-Fixed Single
	Caption	أسماء المجلدات
Label	Name	LblDrive
	BorderStyle	1-Fixed Single
	Caption	اسم السواقة
Label	Name	LblFile
	BorderStyle	1-Fixed Single
	Caption	أسماء الملفات
CommandButton	Name	CmdExit
	Picture	C:\VB6\MyPhotos\ExitPic.Bmp
	Style	1-Graphical
CommandButton	Name	CmdAbout
	Picture	C:\VB6\MyPhotos\AboutPic.Bmp
	Style	1-Graphical



❋ اكتب الشيفرة التالية في حدث تحميل النموذج (Form_ Load)

```
FileName.Pattern = "*.bmp;*.jpg;*.wmf;*.Gif;*.Ico"
```

تحدد الشيفرة السابقة أنواع الملفات الواجب احتوائها في لائحة الملفات ، ويجب أن تكون فقط ملفات صور ...

❋ اكتب الشيفرة التالية في حدث تغيير اسم السواقة (DriveName _Change)

```
1)    On Error GoTo ErrorDrive
2)       DirName.Path = DriveName.Drive
3)       Exit Sub
4)    ErrorDrive :
5)       MsgBox "انتبه يا صاح فالسواقة غير جاهزة", vbMsgBoxRight, "تأكد من السواقة "
6)       DriveName = DirName.Path
7)       Exit Sub
```

ماذا تعني الشيفرة السابقة ؟

السطر الأول : نعلم فيجوال بيسيك بأنه في حال حدوث خطأ ، عليه الانتقال إلى العنوان ErrorDrive (الموجود في السطر / **4** /) ، وحينها لا يقوم بتنفيذ الشيفرتين (2 – 3) ، بل ينفذ الشيفرات (5 – 6 – 7) وهي الموجودة في عنوان الخطأ (ErrorDrive) .

السطر الثاني : إسناد اسم السواقة إلى مسار الأدلة ، **وبمعنى آخر** : ظهور أسماء الأدلة الموجودة في السواقة المختارة .

السطر السادس : إعادة اسم السواقة الذي كان قبل حدوث الخطأ .

باختصار : في الشيفرة السابقة : عندما تغير اسم السواقة ، يظهر في (مربع أسماء الأدلة) جميع الأدلة الموجودة في السواقة التي اخترتها ، وإذا اخترت سواقة غير جاهزة ، فتظهر لك رسالة تنبهك بأن السواقة غير جاهزة ، ثم تظهر اسم السواقة التي كانت قبل انتقاء السواقة غير الجاهزة .

❋ اكتب الشيفرة التالية في حدث تغيير اسم المجلد (DirName_Change) :

```
FileName.Path = DirName.Path
```

تظهر في صندوق (أسماء الملفات) ، جميع ملفات الصور الموجودة في المجلد المختار من صندوق الأدلة .

❋ اكتب الشيفرة التالية في حدث النقر على أسماء الملفات (FileName _ Click) :

```
Dim DirPath, ImgName, ImgFull
DirPath = DirName.Path
ImgName = FileName.FileName
If Right(DirPath, 1) <> "\" Then
DirPath = DirPath + "\"
End If
ImgFull = DirPath + ImgName
PicView.Picture = LoadPicture(ImgFull)
```

عرفنا ثلاث متحولات : الأول سنضع فيه مسار المجلد المختار ، والثاني نضع فيه اسم الصورة التي نقرنا عليها ، والثالث سنضع فيه اسم الصورة ومسارها كاملاً . بعدها أسندنا القيم إلى متحولين ، ثم استخدمنا تابع Right [1] حيث سيقوم باختبار آخر محرف من مسار مجلد الصور ، فإذا كان لا يساوي علامة الفصل بين المسارات (\ ) فيضيفها إليه ...

وأخيراً استخدمنا تابع LoadPicture وهذا التابع يحمل الصورة الموجود اسمها ومسارها في الوسيط المسند إليه ، والذي هو مثالنا (ImgFull) .

❋ اكتب الشيفرة التالية في حدث النقر على زر About :

```
MsgBox "برنامج عرض الصور من إعداد وتصميم" & Chr(13) + "شادي أبو عاصي " _
, vbInformation + vbMsgBoxRight, "حول البرنامج"
```

إن تابع Chr(13) تظهر كل رسالة على سطر ... ، أما عبارة vbInformation فتؤدي لظهور إشارة المعلومات ، في مربع الرسالة .

تأكد من عمل برنامجك بالشكل المطلوب واحتفظ به ثم أجري له نسخة تنفيذية في المجلد Exer8

C:\Vb6\MyExercise\Exer8\ViewImage.Exe

[1] التابع Right يأخذ وسيطين : الأول يحدد سلسلة المحارف المطلوب اختبارها ، والثاني يحدد عدد المحارف من اليمين ...

إنشاء وتحرير الملفات

Creating & Editing Files

سنتعلم في هذا الفصل كيفية إنشاء ملفات جديدة تخدم حاجياتنا البرمجية ، حيث سنقوم بإنشاء برنامج لأرشفة أسماء موظفي شركة ما ، وعند إتقانك لهذا البرنامج ، ستتمكن من إنشاء أي برنامج أرشفة جديد تريده كبرامج تنظيم عيادة الطبيب أو مكتب المحامي أو طلاب المدرسة أو .....

تشغيل البرنامج تظهر الواجهة الأساسية له ، ويمكنك من خلالها إدخال بيانات جديدة أو التعديل في بيانات موجودة مسبقاً أو إلغاء البيانات التي لا تحتاجها ....

التنفيذ : أنشئ مشروعاً جديداً واحفظه في المجلد Exer9 بحيث :

النموذج باسم C:\Vb6\MyExercise\Exer9\Company.Frm

المشروع باسم C:\Vb6\MyExercise\Exer9\CompanyVbp

صمم النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmCompany
	Caption	موظفي الشركة
	Picture	C:\VB6\MyPhotos\Company.Bmp
TextBox	Name	TxtName
	Text	اجعله فارغ
TextBox	Name	TxtPosition
	Text	اجعله فارغ
TextBox	Name	TxtSalary
	Text	اجعله فارغ
TextBox	Name	TxtMark
	Text	اجعله فارغ
	MultiLine	True
Label	Name	LblName
	Alignment	2-Center
	BorderStyle	1-Fixed Single
	Caption	الاسم
Label	Name	LblPosition
	Caption	المنصب
Label	Name	LblSalary

اجعل خاصتي Alignment و Border Style للتسميات الثلاث الآتية مطابقة للتسمية السابقة

	Caption	الراتب
Label	Name	LblMark
	Caption	ملاحظات
Label	Name	LblNum " سيحتوي على رقم الموظف المتسلسل "
	Caption	اجعله فارغ

❋ صمم قوائم النموذج وفق الجدول التـــالي :

Caption	Name	ShortCut (هذه الخاصية لتحديد مفتاح اختصار)
ملف	MnuFile	
خروج....	MnuExit	Ctrl +X
بيانات	MnuData	
إدخال....	MnuInsert	Shift +Ins
بحث....	MnuFind	Shift +F3
حذف....	MnuDelete	Shift +Del
السجل التالي....	MnuNext	F6
السجل السابق....	MnuPrev	F7

لكـي نسـتطيع التعامـل مـع البيـانات وتخزينهـا وتحريرهـا ، علينـا أولاً إنشـاء سـجل ، والسـجل هـو مجموعـة مـن عـدة حقـول تؤلف كلها بيانات أو معلومات عن شيء واحد ، وهذه الحقول تحوي بداخلها على معلومات مختلفة كالاسم والراتب والعمر والعنوان ..... ، ثم نقوم بتخزين مجموعة السجلات في ملف ما على القرص ، ومن خلال الملف يقوم المستخدم بالتعامل مع البيانات الموجودة من حذف أو تحرير أو إضافة ....

لإنشاء سجل علينا أولاً التصريح عنه كنمط ، ثم نعرف عدة متحولات كي تتعامل مع هذا النمط ، ولكي يكون هذا النمط عاماً في النموذج علينا التصريح عنه ضمن برنامج جزئي (Module) ، لذلك أضف برنامج جزئي جديد واحفظه باسم **Company .Bas** ثم اكتب في قسم التصريحات العامة الشيفرة التالية :

```
Option Explicit
Type RecEmploy
        Name As String * 20
        Position As String * 20
        Salary  As String * 5
        Mark As String * 200
End Type
```

لقد قمنا في الشيفرة السابقة بتعريف نوع معطيات جديد أسميناه (RecEmploy) ، وجعلناه يتألف من أربع حقول نصية (String) وبأطوال مختلفة ، أما عبارتي (Type) و (End Type) فهما اللتان تستخدمان لتعريف السجل .

اكتب الشيفرة التالية في قسم التصريح العام للنموذج

```
Option Explicit
        Dim Employ As RecEmploy
        Dim FileNum As Integer
        Dim RecLenght As Long
        Dim RecCurrent As Long
        Dim RecLast As Long
```

عرفنا في الشيفرة السابقة خمس متحولات :

الأول متحول من نوع (RecEmploy) أي أنه سجل تابع للنمط الذي عرفناه في البرنامج الجزئي ، وبالتالي فسيحوي المتحول الذي عرفناه على أربع حقول هي (الاسم – المنصب – الراتب – ملاحظات) ، والمتحولات الباقية ستحتوي على رقم الملف الذي سنخزن السجلات فيه ، وطول السجل ، والسجل الحالي ، والسجل الأخير .

قبل البدء بكتابة شيفرات التعامل مع سجلات الملف عليك معرفة المفاهيم التالية **والتمعن** فيها جيداً :

- ♣ إن أي ملف يفتح في ويندوز يكون له رقم معين ، ويتم معرفة هذا الرقم باستخدام تابع FreeFile .
- ♣ لمعرفة طول أي سجل نستخدم تابع Len .
- ♣ لمعرفة طول أي ملف نستخدم تابع FileLen .
- ♣ يستخدم تابع Trim لإعادة محتوى سلسلة نصية دون الفراغات التي فيها ، فبفرض أنك عرفت متحول لسلسلة نصية بطول /20/ محرف ، وأدخلت فيها اسم من عشر أحرف ، فسيكون هناك عشر فراغات لا تريد ظهورها ، عندها تستخدم تابع Trim .
- ♣ لتخزين بيانات في خانة من خانات سجل ما ، ينبغي كتابة اسم السجل واسم الخلية مفصولة عنه بنقطة ، كأن تكتب : **RecEmploy.Name = " Omar"** .
- ♣ عندما يكون لدينا عدة شيفرات نريد تنفيذها اكثر من مرة في البرنامج ، نقوم بوضعها ضمن إجراء ، ثم عندما نريد تنفيذها نكتب فقط اسم الإجراء ، وبذلك نتفادى تكرار كتابة الشيفرات نفسها عدة مرات ، ولإضافة إجراء **إلى النموذج** : ادخل إلى نافذة الشيفرات ، واختر أمر Add Procedure من قائمة Tools فيظهر لك مربع حوار اكتب فيه اسم الإجراء ثم أنقر **OK** ثم اكتب الشيفرات ضمن الإجراء الذي ظهر في نافذة شيفرات النموذج .[1]
- ♣ لقراءة محتويات سجل ما وإظهارها ، استخدم تابع Get مع المعاملات التالية له :

```
Get  #FileNum, RecNum, Variable
```

حيث أن (FileNum) و (RecNum) هما رقم الملف ورقم السجل ، أما (Variable) فهو المتجول الذي سيتم وضع محتويات السجل فيه .

- ♣ لتخزين محتوى سجل في ملف ، استخدم تابع Put مع المعاملات التالية له :

```
Put  #FileNum, RecNum, Variable
```

حيث أن (FileNum) و (RecNum) هما رقم الملف ورقم السجل ، أما (Variable) فهو المتجول أو السجل الذي يحوي على بيانات السجل .

- ♣ لفتح ملف موجود مسبقاً ، أو إنشاء ملف جديد : نستخدم تابع Open مع المعاملات التالية له :

```
Open "FileName" For Random As Number Len = LenghtRecord
```

الكلمات الموضوع تحتها خط هي قيم متحولات تحدد اسم ومسار الملف ورقمه وطول كل سجل فيه ، أما باقي الكلمات فهي كلمات محجوزة في فيجوال بيسيك . وعلى كل حال سيتضح لك ذلك اكثر عند كتابة الشيفرات التالية :

بعدما سبق : أصبح بمقدورك الآن كتابة الشيفرات المناسبة :

* تأكد من تفعيل نافذة شيفرات النموذج وأضف إجراءاً جديداً باسم ViewRec ، واكتب فيه الشيفرة التالية :

---

[1] إن فكرة الإجراء في فيجوال بيسيك تشبه تقريباً فكرة الملفات الدفعية في Dos أو الماكروات في البرامج المكتبية ....

```
Public Sub ViewRec()
Get #FileNum, RecCurrent, Employ
TxtName.Text = Trim(Employ.Name)
Txtposition.Text = Trim(Employ.Position)
TxtSalary.Text = Trim(Employ.Salary)
TxtMark.Text = Trim(Employ.Mark)
LblNum.Caption = RecCurrent
End Sub
```

لاحظ في الشيفرة السابقة أننا قمنا بقراءة محتويات السجل الحالي (RecCurrent) ، وأسندنا محتوياته إلى المتحول (Employ) ومن ثم أسندنا محتويات خلايا السجل (Employ) إلى مربعات النصوص في النموذج ، ثم أظهرنا رقم السجل الحالي (RecCurrent) في مربع التسمية (LblNum)

* اكتب الشيفرة التالية في قسم تحميل النموذج (Form _Load) :

```
RecLenght = Len(Employ)
FileNum = FreeFile
Open "C: \Company.Sia" For Random As FileNum Len = RecLenght
RecCurrent = 1
RecLast = FileLen("C: \Company.Sia") / RecLenght
If RecLast = 0 Then
   RecLast = RecLast + 1
End If
ViewRec
```

لاحظ أنك قد حصلت على قيمة السجل الأخير (RecLast) بتقسيم طول الملف على طول السجل الواحد ، وفي أول عملية تشغيل للبرنامج ستكون قيمته تساوي الصفر لأن الملف لم يحتوي بعد على أي سجلات ، لذلك قمنا بإسناد القيمة / 1 / له باستخدام عبارة IF ، وأخيراً استدعينا الإجراء ViewRec والذي سيقوم بإظهار محتوى السجل .

* أنشئ إجراءاً جديداً في النموذج باسم SaveRec حيث سيقوم هذا الإجراء بحفظ البيانات التي سندخلها ، واكتب فيه الشيفرة التالية :

```
Public Sub SaveRec()
Employ.Name = TxtName.Text
Employ.Position = TxtPosition.Text
Employ.Salary = TxtSalary.Text
Employ.Mark = TxtMark.Text
Put #FileNum, RecCurrent, Employ
End Sub
```

أسندنا أولاً قيم البيانات التي أدخلناها إلى السجل Employ ثم خزناه في الملف باستخدام تابع (Put) .

* اكتب الشيفرة التالية في حدث النقر قائمة (إدخال MnuInsert) :

```
SaveRec
TxtName.Text = ""
TxtPosition.Text = ""
TxtSalary.Text = ""
TxtMark.Text = ""
RecLast = RecLast + 1
RecCurrent = RecLast
LblNum.Caption = RecCurrent
TxtName.SetFocus
```

دقق في الشيفرة السابقة ولاحظ سهولتها ، واعلم أن خاصة SetFocus تجعل الكائن فعالاً ...

* اكتب الشيفرة التالية في حدث النقر على قائمة (السجل التالي MnuNext) :

```
If RecCurrent = RecLast Then
    MsgBox "لا يوجد تالي فأنت في آخر الملف"
Else
    SaveRec
    RecCurrent = RecCurrent + 1
   ViewRec
End If
```

عندما تصل إلى نهاية الملف تظهر لك رسالة تحذير

❋ اكتب الشيفرة التالية في حدث النقر على قائمة (السجل السابق MnuPrev) :

```
If RecCurrent = 1 Then
    MsgBox "لا يوجد سابق فأنت في بداية الملف"
Else
    SaveRec
    RecCurrent = RecCurrent - 1
   ViewRec
EndIf
```

❋ اكتب الشيفرة التالية في حدث النقر على قائمة (الخروج MnuExit) :

```
SaveRec
Close FileNum
End
```

قمنا أولاً بحفظ السجل الحالي ثم أغلقنا الملف باستخدامنا لتابع Close وهذا التابع يغلق الملف ذات الرقم المحدد بعده وهو في مثالنا FileNum

❋ اكتب الشيفرة التالية في حدث إلغاء تحميل النموذج (Form _unload) :

```
SaveRec
Close FileNum
```

تضمن الشيفرة السابقة تخزين آخر سجل أدخلته في حال أنك خرجت من البرنامج بالنقر على زر الإغلاق **X** وليس بالنقر على قائمة الخروج ...

❋ اكتب الشيفرة التالية في حدث النقر على قائمة (بحث MnuFind) :

```
Dim NameFind As String
Dim RecNum As Integer
Dim Found As Boolean
Dim EmpFind As RecEmploy
Found = False
NameFind = InputBox("أدخل اسم للبحث عنه" , " بحث " )
For RecNum = 1 To RecLast
    Get #FileNum, RecNum, EmpFind
    If NameFind = Trim(EmpFind.Name) Then
        Found = True
        Exit For
    End If
Next
If Found = True Then
   SaveRec
   RecCurrent = RecNum
   ViewRec
Else
   MsgBox "لم يتم العثور على الاسم الذي أدخلته"
End If
```

النمط Boolean يأخذ فيه المتحول قيمتين فقط إما False أو True

عمليـة الحـذف : عنـد النقـر علـى زر الحـذف يجـب ظهـور رسـالة تأكيـد ، فـإذا نقـر منهـا المسـتخدم علـى زر ((**لا**)) يجـب أن يعود إلى البرنامج ، أما إذا نقر زر ((**نعم**)) فتبدأ عملية الحذف التي تتلخص خطواتها بما يلي :

1. إنشاء ملف جديد على القرص .
2. نسخ جميع السجلات من الملف الحالي إلى الملف الجديد ، عدا السجل المطلوب حذفه .
3. حذف الملف القديم (Company.Sia) ، و إعادة تسمية الملف الجديد وجعله باسم (Company.Sia) .

* لتنفيذ الخطوات السابقة اكتب الشيفرة التالية في حدث النقر على قائمة (حذف MnuDelete) :

```
Dim HelpFileNum, HelpRecNum, RecNum As Integer
Dim HelpEmploy As RecEmploy
If MsgBox("هل تريد حقاً حذف سجل هذا الموظف ؟", vbYesNo) = vbNo Then
   Exit Sub
End If
HelpFileNum = FreeFile
Open "C:\HelpComp.Sia" For Random As HelpFileNum Len = RecLenght
RecNum = 1
HelpRecNum = 1
Do While RecNum < RecLast + 1
    If RecNum <> RecCurrent Then
        Get #FileNum, RecNum, HelpEmploy
        Put #HelpFileNum, HelpRecNum, HelpEmploy
        HelpRecNum = HelpRecNum + 1
    End If
    RecNum = RecNum + 1
Loop
Close FileNum
Close HelpFileNum
Kill "C:\Company.Sia"
Name "C:\HelpComp.Sia" As "C:\Company.Sia"
Open "C:\Company.Sia" For Random As FileNum Len = RecLenght
RecLast = RecLast - 1
If RecCurrent > RecLast Then
    RecCurrent = RecLast
End If
ViewRec
```

بذلك تكون قد انتهيت من تنفيذ البرنامج نهائياً .... قم بتجريبه والتأكد من صحة عمله ، واحتفظ به ، ثم حوله إلى ملف تنفيذي في الدليل Exer9

C:\Vb6\MyExercise\Exer9\Company.Exe

قد تبدو لك شيفرات هذا البرنامج طويلة ، لكن يمكنك استخدامها في أي برنامج قادم تنشئه ، فأي برنامج تريد تصميمه ويحتـوي علـى أمـر الحـذف مـثلاً ، تسـتطيع نسـخ شـيفرة الحـذف مـن هـذا الـبرنامج إليـه مـع تغيـير بسـيط في أسمـاء المتحولات فقط .

إنشاء الملفات النصية

Creating The Text Files

سنتعلم في هذا البرنامج كيفية إنشاء ملفات نصية وحفظها ، أو فتح ملفات موجودة مسبقاً ....

عند تشغيل البرنامج تظهر نافذة تمكنك قوائمها من إنشاء ملف نصي جديد وحفظه باسم معين في أي مجلد من مجلدات القرص ، وأيضاً يمكنك فتح أي ملف نصي موجود ... وبرنامجنا هذا يشبه إلى حد كبير برنامج المفكرة الذي يأتي مع نظام التشغيل Windows .

التنفيذ : أنشئ مشروعاً جديداً واحفظه في المجلد Exer10 بحيث :

النموذج باسم C:\Vb6\MyExercise\Exer6\TextFile.Frm

المشروع باسم C:\Vb6\MyExercise\Exer6\ TextFile Vbp

نفذ النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmTextFile
	Caption	إنشاء الملفات النصية
TextBox	Name	TxtFile
	MultiLine	True
	ScrollBars	3-Both
	Text	اجعله فارغاً
CommonDialog	Name	CommDial
	CancelError	True
	Filters	(الملفات النصية)|*.Txt|

صمم قوائم النموذج وفق الجدول التالي :

Caption	Name
ملف	MnuFile
جديد....	MnuNew
فتح....	MnuOpen
حفظ....	MnuSave
خروج....	MnuExit

قبل.. البدء بكتابة الشيفرات .....**استرق النظر بما يلي :**

سنتعرف الآن على طرق جديدة لإنشاء الملفات أو الكتابة فيها أو القراءة منها ، بحيث سنستخدم التعاليم التالية :

A- لإنشاء ملف والكتابة فيه ، اكتب الشيفرتين التاليتين :

```
Open FileName For Output As FileNum
Write #FileNum , Text
```

حيث FileName هو اسم الملف المطلوب إنشاءه ، و FileNum هو رقمه التسلسلي ، و Text هو النص المطلوب كتابته في الملف .

B- لقراءة محتوى ملف موجود اكتب الشيفرتين التاليتين :

```
Open FileName For Input As FileNum
Variable = Input$ (Number, FileNum)
```

حيث أن تابع Input$ يأخذ وسيطين : الأول يحدد عدد المحارف المطلوب قراءتها من الملف ، والثاني يحدد رقم الملف المطلوب القراءة منه ، أما Variable فهو متحول نصي تُسند إليه المحارف التي ستتم قراءتها .

C– لمعرفة ما إذا كان ملف موجود على القرص أم لا ، استخدم تابع Dir كما يلي :

Dir(PathName)

حيث أن PathName هو مسار واسم الملف ، وهذا التابع يعيد فقط اسم الملف ، لذلك إذا أعاد التابع قيمة محرفية فارغة ، فهذا يعني أن الملف غير موجود .

كتابة الشيفرات :

❋ اكتب الشيفرة التالية في قسم التصريح العام للنموذج (General) :

```
Option Explicit
Dim FileNum As Integer
```

❋ اكتب الشيفرة التالية مرتين :في حدث **تحميل النموذج (Form _Load)** ، وفي (**تغيير الحجم Form _Resize**)

```
TxtFile.Height = FrmTextFile.ScaleHeight
TxtFile.Width = FrmTextFile.ScaleWidth
```

❋ اكتب الشيفرة التالية في حدث إلغاء تحميل النموذج (Form _Unload) :

```
IF MsgBox("هل حقاً تريد الخروج ؟", vbYesNo + 524288 +  1048576, "تأكيد خروج ") = vbNo Then
            Cancel = True
End If
```

عندما يأخذ المتحول Cancel قيمة True لا يتم إغلاق النموذج .

❋ اكتب الشيفرة التالية في حدث النقر على قائمة (خروج MnuExit) :

```
IF MsgBox("هل حقاً تريد الخروج ؟", vbYesNo + 524288 + 1048576, "تأكيد خروج ") = vbYes Then
      End
End If
```

❋ أنشئ إجراءً جديداً في ترميز النموذج ، واسميه Title ، ثم اكتب فيه الشيفرة التالية :

```
Public Sub Title()
FrmTextFile.Caption = " انشاء الملفات النصية "  +  "  ("  + CommDial.filename  + ")"
End Sub
```

سنستخدم الإجراء السابق لتحديث عنوان النموذج عندما نحتفظ بالملف ، بحيث يظهر اسم الملف في عنوان النافذة .

❋ أنشئ إجراءً جديداً في ترميز النموذج ، واسميه SaveFile ، ثم اكتب فيه الشيفرة التالية :

```
Open CommDial.filename For Output As FileNum
    Write  # FileNum, TxtFile.Text
    Close FileNum
```

سنستخدم الإجراء السابق لكتابة البيانات التي أدخلناها في الملف النصي ، أي الاحتفاظ بالملف .

❋ اكتب الشيفرة التالية في حدث النقر على قائمة (**حفظ – MnuSave**) :

```
On Error GoTo SaveError
CommDial.ShowSave
FileNum = FreeFile
BeginSave:
```

```
If Dir(CommDial.filename) <> "" Then
    If MsgBox("هذا الملف موجود مسبقاً ، هل تريد استبداله", vbYesNo) = vbYes Then
        SaveFile
        Title
    Else
        CommDial.ShowSave
        GoTo BeginSave
    End If
Else
    SaveFile
    Title
    Exit Sub
End If
Close FileNum
Exit Sub
SaveError:
Exit Sub
```

لاحظ وجود عنوانين في الشيفرة السابقة هما : *BeginSave* و *SaveError* ، فعندما يظهر مربع حوار الحفظ سيقوم المستخدم بنقر إحدى زريه (**حفظ** أو **إلغاء الأمر**) ، فإذا نقر على زر **حفظ** ستنفذ الشيفرات الموجودة في العنوان *BeginSave* ، أما إذا نقر على زر **إلغاء الأمر** فستنفذ الشيفرة الموجودة في عنوان *SaveError* وبالتالي سيختفي مربع حوار الحفظ .

لاحظ أيضاً أن الشيفرة تناقش احتمال وجود ملف بنفس الاسم الذي أدخله المستخدم ، حيث تظهر رسالة تنبيه لذلك ، ويمكن المستخدم حينها أن يستبدل الملف الموجود أو أن يدخل اسماً مختلفاً .

❋ اكتب الشيفرة التالية في حدث النقر على قائمة (**فتح – MnuOpen**) :

```
Dim NumChar
On Error GoTo OpenError
  FileNum = FreeFile
  CommDial.ShowOpen
  Open CommDial.filename For Input As FileNum
  NumChar = FileLen(CommDial.filename)
  TxtFile.Text = Input$(NumChar, FileNum)
  FrmTextFile.Caption = "إنشاء الملفات النصية      " + CommDial.FileTitle
  Close FileNum
  Exit Sub
OpenError:
  Exit Sub
```

عند حدوث الخطأ تنفذ شيفرة العنوان **OpenError**: وتقوم بإلغاء مربع حوار الفتح ، أما الخطأ فسيحدث إذا نقر المستخدم على زر **إلغاء الأمر**

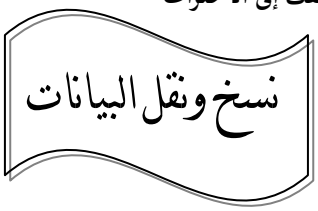
❋ اكتب الشيفرة التالية في حدث النقر على قائمة (**جديد – MnuNew**) :

```
TxtFile.Text = ""
FrmTextFile.Caption = "انشاء الملفات النصية"
```

تقوم الشيفرة السابقة بمسح محتويات مربع النصوص ، ثم تحديث قيمة عنوان النافذة .

برنامجك جاهز ، اختبره وتأكد من عمله بالشكل المطلوب ثم احتفظ به وحوله إلى نسخة تنفيذية في المجلد Exer10 باسم TextFile.Exe

Copy & Move Data



كثيراً ما يحتاج المستخدم إلى نسخ بيانات معينة من برنامج لآخر ، كأن ينسخ نصوص مكتوبة في برنامج المفكرة مثلاً أو أن ينسخ صور مرسومة في برنامج الرسام إلى صورة أخرى ، **وبرنامجنا الجديد** سيتيح لنا إمكانية التعامل مع البيانات من حيث نسخها أو نقلها من برنامج لآخر .

:

عند تشغيل البرنامج تظهر الواجهة الأساسية وفيها مربعي صورة ، ومربعي نصوص ، ويمكنك من خلالها نسخ أو نقل الصور بين مربعي الصور ، أو نسخ ونقل النصوص بين مربعي النصوص ، كما يمكنك أيضاً نسخ بيانات من برامج أخرى كبرنامج الرسام أو وورد أو المفكرة .....

* أنشئ مشروعاً جديداً في المجلد Exer11 واحفظه باسم :

النموذج باسم C:\Vb6\MyExercise\Exer11\CopyMove.Frm

المشروع باسم C:\Vb6\MyExercise\Exer11\CopyMove.Vbp

صمم النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmCopymove
	Caption	النسخ والنقل
	RightToLeft	True
PictureBox	Name	Pic1
	BorderStyle	0-None
PictureBox	Name	Pic2
	BorderStyle	0-None
TextBox	Name	Text1
	Text	اجعله فارغاً
TextBox	Name	Text1
	Text	اجعله فارغاً

صمم قوائم النموذج وفق الجدول التالي :

Caption	Name	Short Cut
ملف	MnuFile	
خروج....	MnuExit	
تحرير	MnuEdit	
قص....	MnuCut	Ctrl +X
نسخ....	MnuCopy	Ctrl +C
لصق....	MnuPaste	Ctrl +V

قبل البدء بكتابة الشيفرات سنتعرف على المفاهيم التالية :

1) لكي نتمكن من إجراء عمليات النسخ واللصق ، علينا الوصول إلى الحافظة ، ويكون ذلك باستخدام عبارة Clipboard ، وبما أن نوع البيانات قد يكون صور أو نصوص ، فعلينا التمييز بنوعها للغة فيجول بيسيك وذلك باستخدام التوابع التالية :

SetText – SetData – GetText – GetData

2) لمعرفة نوع العنصر الفعال من نافذة تحوي على عدة أنواع عناصر ، نستخدم الكلمة المحجوزة TypeOf ونستخدم معها عبارة Screen.ActiveControl

❋ اكتب الشيفرة التالية في حدث النقر على قائمة (نسخ MnuCopy) :

```
Clipboard.Clear
If TypeOf Screen.ActiveControl Is TextBox Then
      Clipboard.SetText      Screen.ActiveControl.SelText
ElseIf TypeOf Screen.ActiveControl Is PictureBox Then
      Clipboard.SetData      Screen.ActiveControl.Picture
End If
```

اعلم أن عبارة SelText تعني النص المختار أو المحدد من مربع النصوص ....

❋ اكتب الشيفرة التالية في حدث النقر على قائمة (قص MnuCut) : [1]

```
Clipboard.Clear
If TypeOf Screen.ActiveControl Is TextBox Then
      Clipboard.SetText      Screen.ActiveControl.SelText
      Screen.ActiveControl.SelText = " "
ElseIf TypeOf Screen.ActiveControl Is PictureBox Then
      Clipboard.SetData      Screen.ActiveControl.Picture
      Screen.ActiveControl.Picture = LoadPicture()
End If
```

أن عملية إسناد بيانات إلى الحافظة لا تستخدم إشارة المساواة (=) ...

❋ اكتب الشيفرة التالية في حدث النقر على قائمة (لصق MnuPaste) :

```
If TypeOf Screen.ActiveControl Is TextBox Then
      Screen.ActiveControl.SelText = Clipboard.GetText()
ElseIf TypeOf Screen.ActiveControl Is PictureBox Then
      Screen.ActiveControl.Picture = Clipboard.GetData()
End If
```

البرنامج وجربه ولاحظ أنه بالإمكان نسخ ونقل البيانات من برامج أخرى إلى هذا البرنامج ، فمثلاً جرب أن تفتح برنامج الرسام وارسم فيه صورة ، ثم انسخها ، وعد إلى برنامجك واختر أمر اللصق بعد تفعيل إحدى مربعي الصورة ، ولاحظ أن الصورة قد ظهرت فيه ...

[1] تعلم أن عملية القص تشمل مرحلتين هما : نسخ البيانات المحددة إلى الحافظة ، ومسح هذه البيانات من مصدرها ....

تحسين البرنامج :

لاحظ أن مربعي الصورة في البرنامج عند التنفيذ غير واضحين بشكل مناسب ، وأن المستخدم قد لا يعلم إن كان قد تفعل مربع الصورة أم لا ..... لذلك سنقوم بجعل مربع الصورة يتغير شكله قليلاً عند الانتقال إليه ، أي عندما يصبح فعالاً ...

❋ اكتب الشيفرة التالية في حدث تفعيل مربع الصورة الأول ، ((**ويتم التفعيل بالنقر عليها بالماوس**)) :

```
Private Sub Pic1_GotFocus()
    Pic1.BorderStyle = 1
End Sub
```

❋ اكتب الشيفرة التالية في حدث إلغاء تفعيل مربع الصورة الأول ، ((**ويتم التفعيل بالنقر على كائن آخر غيره**)) :

```
Private Sub Pic1_LostFocus()
    Pic1.BorderStyle = 0
End Sub
```

❋ اكتب الشيفرة التالية في حدث تفعيل مربع الصورة الثاني :

```
Private Sub Pic2_GotFocus()
    Pic2.BorderStyle = 1
End Sub
```

❋ اكتب الشيفرة التالية في حدث إلغاء تفعيل مربع الصورة الثاني :

```
Private Sub Pic2_LostFocus()
    Pic2.BorderStyle = 0
End Sub
```

اصنع ملفاً تنفيذياً للبرنامج واحفظه في المجلد ***C:\Vb6\MyExercise\Exer11\CopyMove.Exe***

The Wave Files Program

برنامج الملفات الصوتية

برنامجنا الجديد يقوم بفتح ملفات الأصوات الموجودة في مجلد ما على القرص ويقوم بقراءتها وإظهار الصوت من كرت الصوت الموجود بالجهاز ((**ملفات الصوت هي الملفات التي تحمل اللاحقة WAV**)) .

❋ أنشئ مشروعاً جديداً في المجلد Exer12 واحفظه باسم :

النموذج باسم C:\Vb6\MyExercise\Exer12\PlayWave.Frm

المشروع باسم C:\Vb6\MyExercise\Exer12\PlayWave.Vbp

صمم النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmPlayWave
	Caption	Play Wave
CommandButton	Name	CmdExit
	Caption	Exit
CommandButton	Name	CmdOpen
	Caption	Open
Shape	Name	Shape1
	FillStyle	0-Solid
	Shape	2- Oval
Tegomm (1)	Name	Tegomm
	BevelWidth	5 تحدد قيمة تحدب شكل الأزرار
	DeviceType	WaveAudio (2)
	UpdateInterval	50
CommonDialog (3)	Name	CommDial
	CancelError	True
	Filter	(Wave Files) |*.Wav|
HscrollBar	Name	HSBar
	Max	100
	Min	0

:

❋ اكتب شيفرة إنهاء البرنامج في حدث النقر على زر الخروج CmdExit

(1) لإضافة هذا الكائن عليك إضافة الملف TegoMM32.Ocx إلى صندوق الأدوات ...

(2) إن كائن TegoMM يتعامل مع أكثر من نوع ملفات كالفيديو أو ملفات الميديا .... لذلك حددنا له نوع الملفات التي سيتعامل معها بأنها ملفات من نوع WaveAudio

(3) لإضافة هذا الكائن عليك إضافة الملف ComDlg32.Ocx إلى صندوق الأدوات ...

❋ اكتب الشيفرة التالية في حدث النقر على زر الفتح (CmdOpen) :

```
On Error GoTo OpenError
   CommDial.ShowOpen
   Tegomm.FileName = CommDial.FileName
   Tegomm.Command = "Open"
   FrmPlayWave.Caption = "Play Wave"  +   " (" + CommDial.FileTitle + " )"
   Exit Sub
OpenError:
    Exit Sub
```

إن الشيفرة Tegomm.Command = "Open" تعني تجهيز عملية قراءة الملف المفتوح ....

❋ اكتب الشيفرة التالية في الحدث Status Update لعنصر TegoMM :

هذا الحدث سينفذ كل / 50 / ميللي ثانية ، وذلك لأننا وضعنا هذه القيمة في الخاصية UpdateInterval

```
Private Sub Tegomm_StatusUpdate()
    Static ForWidth, Size
    If Tegomm.Mode <> 526 Then
       Shape1.Visible = False
       Exit Sub
    End If
    HSBar.Value = ((Tegomm.Position) * 100 / (Tegomm.Length))
    Shape1.Visible = True
    If Size = 0 Then
       ForWidth = ForWidth + 50
       If ForWidth >= 1200 Then
          Size = 1
       End If
    End If
    If Size = 1 Then
      ForWidth = ForWidth - 50
      If ForWidth < = 0 Then
         Size = 0
      End If
    End If
    Shape1.FillColor = QBColor(Int(Rnd * 15))
    Shape1.Width = ForWidth
End Sub
```

إن الخاصية **Mode** التابعة لكائن الـ **Tego** تأخذ القيمة / **526**/ عندما يكون الملف في حالة القراءة ، وفيما عدا ذلك تتغير قيمتها إلى قيم أخرى ، أي تصبح غير مساوية للقيمة **526**

(هكذا صنعتها شركة الـ **TegoSoft** ....[$])

إن الشيفرة HSBar.Value = ((Tegomm.Position) * 100 / (Tegomm.Length)) تضمن سير مؤشر شريط التدرج مع قراءة الملف الصوتي بالشكل المناسب ....

[$] لمزيد من المعلومات راجع الجدول 5 آخر المرجع .....

إن مهمة عبارات **IF** الأخيرة والتي تناقش قيم المتحولين **Size** و **ForWidth** هي تغيير حجم كائن الـ **Shape** ، والمتحول **Size** أخذ قيمتين هما (**الصفر** أو **الواحد**) ، فإذا كان صفراً يزيد حجم الكائن ، وإذا كان واحد فينقص الحجم .

جرب البرنامج :

انقر على الزر OPEN واختر أي ملف صوتي موجود لديك ، ولاحظ أن اسم الملف قد ظهر في عنوان النموذج .

انقر على زر تشغيل الصوت ، سيبدأ البرنامج بقراءة الملف الصوتي ، **وعند الانتهاء منه :**

انقر مجدداً على زر القراءة ولاحظ أن البرنامج لم يعيد قراءة الملف الصوتي من جديد ....

لاحظ أن مؤشر شريط التدرج قد بقي في آخر الشريط ، وعلى الرغم من أن ذلك لا يؤثر على عمل البرنامج ، لكن لتحسين عمله و إصلاح الفكرتين الأخيرتين ، سنكتب الشيفرة التالية في الحدث Done والذي ينفذ عند انتهاء البرنامج من قراءة الملف الصوتي ...

```
Private Sub Tegomm_Done()
    Tegomm.Command = "Prev"
    HSBar.Value = 0
End Sub
```

الشيفرة السابقة تفيد بتهيئة عملية إعادة القراءة ، و إعادة مؤشر الشريط لبدايته ....

بذلك أصبح برنامجك جاهز ولا يشكو من شيء

احفظه واصنع منه نسخة تنفيذية ، واحفظ الملف التنفيذي الجديد في المجلد Exer12 :

C:\Vb6\MyExercise\Exer12\PlayWave.Exe

برنامج الصوت والصورة

Sound & Image Program

سنتعلم في هذا البرنامج كيفية إنشاء برامج توافق بين الصوت والصورة ، وتنسق بين ظهور الصوت وظهور الصور المناسبة له بنفس التوقيت ....

سيتضمن برنامجنا الجديد مقطع من أغنية للسيدة فيروز ، ومع ظهور الأغنية تظهر صور متحركة لعازف كمان ، يمكنك التحكم بإخفائه أو إظهاره ، وتظهر أيضاً كلمات الأغنية بتوافق مع الصوت .

❋ أنشئ مشروعاً جديداً في المجلد Exer13 واحفظه باسم :

النموذج باسم C:\Vb6\MyExercise\Exer13\Feiruz.Frm

المشروع باسم C:\Vb6\MyExercise\Exer13\Feiruz.Vbp

ملاحظات قبل البدء بالتصميم :

1–)) يجب إنشاء اثنا عشر صورة ، وتخزينها في نفس مجلد العمل (**Exer13**) ، بحيث تأخذ هذه الصور الأسماء **Kaman1.Jpg – Kaman2.Jpg – Kaman3.Jpg.....Kaman12.Jpg** وهي صور لعازف كمان وبوضعيات مختلفة لكل صورة .

2–)) يجب إنشاء ست صور ، وتخزينها في نفس مجلد العمل (**Exer13**) ، بحيث تحوي كل صورة على قسم من مقطع الأغنية ، وتأخذ الأسماء **Spk0.Jpg – Spk1.Jpg.....Spk5.Jpg**

3–)) إنشاء صورة للسيدة فيروز في نفس المجلد باسم **Feiruz.Jpg** ((**إذا أمكن** ...))

4–)) يجب إضافة الملف **Mci32.Ocx** إلى صندوق الأدوات لأن النموذج سيتضمن كائن **MMControl** .

بعد الانتهاء من تنفيذ الملاحظات السابقة قم بتصميم النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmFeiruz
	Caption	برنامج الصوت والصورة
	RightToLeft	True
Image	Name	ImgKaman
	Picture	C:\Vb6\MyExercise\Exer13\Kaman1.Jpg
	Stretch	True
Image	Name	ImgSpeak
	Picture	C:\Vb6\MyExercise\Exer13\Frieuz.Jpg
	Stretch	True

Object	Properties	Set
Image	Name	ImgFinal
	Picture	C:\Vb6\MyExercise\Exer13\Kaman1.Jpg
	Stretch	True
	Visible	False
Image	Name	ImgFinalFeiruz
	Picture	C:\Vb6\MyExercise\Exer13\Feiruz.Jpg
	Stretch	True
	Visible	False
Image	Name	ImgSong
	Index	0
	Picture	C:\Vb6\MyExercise\Exer13\Spk0.Jpg
	Stretch	True
	Visible	False

تابع ← ← ←

ملاحظة أنشئ خمس صور أخرى بنفس مواصفات الصورة الأخيرة التي أنشأتها عدا خاصية Index فيجب أن تكون ذات قيم متتابعة من (0) حتى (5) ، واجعلها كلها باسم ImgSong وبهذه الحالة ستظهر لك رسالة تسأل إذا كنت تريد إنشاء نسق ، لذلك انقر منها على زر **نعم** ، بعدها أسند للصور الجديدة التي أنشأتها باقي الصور Spk1- Spk2 –Spk3..... ، بعدها أكمل تصميم النموذج

CommandButton	Name	CmdStart
	Caption	استماع
CommandButton	Name	CmdStop
	Caption	إيقاف
CommandButton	Name	CmdExit
	Caption	خروج
Timer	Name	Timer1
	Interval	200
MMControl	Name	MctrlSong
	DeviceType	WaveAudio
	UpdateInterval	100
	Visible	False
CheckBox	Name	ChkKaman
	Caption	عرض عازف الكمان
	Value	1-Checked
CheckBox	Name	ChkRepeat
	Caption	إعادة آلية
	Value	1-Checked

كتابة الشيفرات

❋ اكتب الشيفرة التالية في قسم التصريح العام للنموذج (General) :

```
Option Explicit
Dim View As Boolean
```

سنستخدم هذا المتحول للإيعاز ببدء أو إيقاف تحريك عازف الكمان .

❋ اكتب الشيفرة التالية في حدث النقر على صندوق التحقق (ChkKaman) :

```
If ChKKaman.Value = 0 Then
    ImgKaman.Visible = False
Else
    ImgKaman.Visible = True
End If
```

* اكتب شيفرة الخروج في حدث النقر على زر الخروج (CmdExit)

* اكتب الشيفرة التالية في حدث تحميل النموذج (Form _Load) :

```
Dim PathSong
PathSong = App.Path
If Right(PathSong, 1) <> "\" Then
   PathSong = PathSong + "\"
End If
MCtrlSong.FileName = PathSong + "Feiruz.Wav"
MCtrlSong.Command = "Open"
CmdStop.Enabled = False
```

تقوم الشيفرة السابقة بتجهيز الكائن MMControl لقراءة الملف Feiruz.Wav ثم تعطيل زر الإيقاف لأنه لا معنى من نقره طالما أنه لم يتم إصدار الصوت بعد .

* اكتب الشيفرة التالية في حدث النقر على زر الاستماع (CmdStart) :

```
View = True
MCtrlSong.Command = "play"
CmdStart.Enabled = False
CmdStop.Enabled = True
```

* اكتب الشيفرة التالية في حدث النقر على الإيقاف (CmdStop) :

```
View = False
MCtrlSong.Command = "Stop"
CmdStart.Enabled = True
CmdStop.Enabled = False
```

* اكتب الشيفرة التالية في حدث عدم إصدار الصوت (MctrlSong _Done) :

```
View = False
CmdStop.Enabled = False
CmdStart.Enabled = True
If MCtrlSong.Position = MCtrlSong.Length Then
   MCtrlSong.Command = "Prev"
   ImgKaman.Picture = ImgFinal.Picture
      If ChkRepeat.Value = 1 Then
        CmdStart_Click
        ImgSpeak.Visible = True
      Else
        ImgSpeak.Picture = ImgFinalFeiruz.Picture
      End If
End If
```

لاحظ أن الشيفرة السابقة تختبر إذا ما كان قد انتهى الملف الصوتي ، فإذا كان ذلك تختبر أيضاً قيمة صندوق التحقق أو الاختبار (ChkRepeat) ، فإذا وجدته محققاً تعيد العرض وإصدار الصوت من جديد ....

* اكتب الشيفرة التالية في المؤقت (Timer1) :

```
Dim PathName, FileName
Static Num As Integer
If View = True Then
   PathName = App.Path
```

```
   If Right(PathName, 1) <> "\" Then
      PathName = PathName + "\"
   End If
   Num = Num + 1
   If Num = 13 Then
     Num = 1
   End If
      FileName = PathName + "kaman" + Format(Num) + ".jpg"
      ImgKaman.Picture = LoadPicture(FileName)
End if
```

تابع الشيفرة

* اكتب الشيفرة التالية في حدث تعديل خصائص كائن MMControl (MctrlSong _Update) :

```
If MCtrlSong.Position > 1 And MCtrlSong.Position < 2786 Then
        ImgSpeak.Picture = ImgSong(0).Picture
End If
If MCtrlSong.Position > 2786 And MCtrlSong.Position < 7293 Then
        ImgSpeak.Picture = ImgSong(1).Picture
End If
If MCtrlSong.Position > 7293 And MCtrlSong.Position < 10088 Then
        ImgSpeak.Picture = ImgSong(2).Picture
End If
If MCtrlSong.Position >= 10088 And MCtrlSong.Position < 14469 Then
        ImgSpeak.Picture = ImgSong(3).Picture
End If
If MCtrlSong.Position >= 14469 And MCtrlSong.Position < 18747 Then
        ImgSpeak.Picture = ImgSong(4).Picture
End If
If MCtrlSong.Position >= 18747 Then
        ImgSpeak.Picture = ImgSong(5).Picture
End If
```

لاحظ أن الشيفرة السابقة كلها عبارة عن شيفرات اختبار للموقع الذي أصبحت فيه عملية القراءة ، إذ أن الأغنية التي لدينا يبلغ طولها (18747) ميللي ثانية، فعندما ينتهي المقطع الأول يكون الموقع قد تجاوز الـ (7293) ميللي ثانية ، وعندما ينتهي المقطع الثاني يكون الموقع قد تجاوز الـ (10088) ميللي ثانية وهكذا ..... أما بالنسبة لمواقع مقاطع الأغنية وقيمها فيمكنك معرفتها بسهولة وبعدة طرق .

البرنامج وتأكد من صحة عمل كافة العناصر التي فيه كأزرار الأمر وصناديق الاختبار ، وإذا وجدته يعمل بشكل صحيح ، احتفظ به واصنع منه نسخة تنفيذية في المجلد **Exer13**

C:\Vb6\MyExercise\Exer13\FeiruZ.Exe

إذا قمت بنقل الملف التنفيذي إلى مجلد آخر ، فيجب نقل جميع ملفات الصور معه ،

وكذلك ملف الصوت Feiruz.Wav ..

قواعد المعطيات

Data Bases

هي عبارة عن كائنات تنشأ ضمن برامج خاصة بذلك ، وتتألف كل قاعدة معطيات من عدة عناصر تشكل بمجموعها طريقة سهلة للتعامل مع البيانات التي يدخلها المستخدم ، كبيانات موظفي شركة أو دليل هاتف أو محل تجاري .... ، وأهم عنصر في قاعدة المعطيات هو الجدول ، والجدول كما هو معروف : عبارة عن مجموعة أعمدة وصفوف تشكل بتقاطعها خلايا تحوي بداخلها على معلومات لشيءٍ ما .

عدة برامج لإنشاء قواعد المعطيات ، أكثرها استخداماً هو برنامج **MsAccess** ....

يقوم البرنامج الجديد الذي سننشئه بتنظيم بضاعة محل أدوات كهربائية ، لذلك سنقوم بإنشاء قاعدة معطيات باستخدام برنامج **Access** ، وسنفترض أن أرشيف المحل يتألف من جدول فيه خمسة حقول هي :

الجهاز – نوعه – قياسه – العدد – سعر المبيع

إنشاء قاعدة البيانات المطلوبة :

1) شغّل برنامج **Microsoft Access** .
2) تظهر نافذة افتتاحية ، انقر منها على خيار (**قاعدة بيانات فارغة**) ، ثم انقر زر (**موافق**) .
3) يظهر مربع حوار لتحديد اسم وموقع القاعدة : حدد الاسم بـ **Store.Mdb** واحفظها في المجلد C:\Vb6\MyExercise\Exer14 ثم انقر زر (**إنشاء**) .
4) تظهر قاعدة البيانات وهي تحوي على عدة عناصر ، سنستخدم منها الجداول ، لذلك انقر مباشرة على زر (**جديد**) فتظهر نافذة انقر منها على خيار (**عرض تصميمي**) ، ثم انقر زر (**موافق**) .
5) تظهر نافذة التصميم وهي تتألف من ثلاثة أعمدة : الأول لتحديد اسم الحقل ، والثاني لتحديد نوع البيانات التي فيه (نص – رقم – مذكرة ...الخ) .
6) اكتب أسماء الحقول الخمسة (الجهاز – نوع الجهاز – قياس الجهاز – العدد – سعر المبيع) ، واجعل أنواعها كلها رقمية ، عدا حقلي الجهاز والنوع فاجعلها نصية .
7) بعد الانتهاء ، اختر الأمر (**طريقة عرض صفحة البيانات**) من قائمة (**عرض**) ، فتظهر لك رسالة تطالب بحفظ الجدول انقر منها على زر (**نعم**) ، ثم تظهر نافذة أخرى لتحديد اسمه ، حدد الاسم بـ **Store** ، ثم انقر (**موافق**) ، ثم تظهر رسالة لإنشاء مفتاح أساسي ، انقر منها على زر (**لا**) .

8) بذلك أصبح الجدول في حالة إدخال البيانات ، لذلك قم بإدخال مجموعة من البيانات للتجربة ، وبعد الانتهاء أغلق برنامج Access .

شغل برنامج فيجول بيسيك للبدء بالبرنامج

* أنشئ مشروعاً جديداً في المجلد Exer14 واحفظه باسم :

النموذج باسم C:\Vb6\MyExercise\Exer14\Store.Frm

المشروع باسم C:\Vb6\MyExercise\Exer14\Store.Vbp

صمم النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmStore
	Caption	محل الأدوات الكهربائية
	RightToLeft	True
Data	Name	DataStore
	Caption	الأدوات الكهربائية
	DataBaseName	C:\Vb6\MyExercise\Exer14\Store.Mdb
Label	Name	LblName
	Alignment	2-Center
	BorderStyle	1-Fixed Single
	Caption	اسم الجهاز
Label	Name	LblType
	Caption	نوع الجهاز
Label	Name	LblSize
	Caption	قياس الجهاز
Label	Name	LblTotal
	Caption	العدد
Label	Name	LblSale
	Caption	سعر المبيع
TextBox	Name	TxtName
	DataSource	DataStore
	DataField	الجهاز
	Text	اجعله فارغاً
TextBox	Name	TxtType
	DataSource	DataStore
	DataField	نوع الجهاز
	Text	اجعله فارغاً
TextBox	Name	TxtSize
	DataSource	DataStore
	DataField	قياس الجهاز
	Text	اجعله فارغاً
TextBox	Name	TxtTotal
	DataSource	DataStore
	DataField	العدد

اجعل خاصتي المحاذاة ونمط الحدود لكافة التسميات Label متشابهة

Object	Properties	Set
	Text	اجعله فارغاً
TextBox	Name	TxtSale
	DataSource	DataStore
	DataField	سعر المبيع
	Text	اجعله فارغاً
CommandButton	Name	CmdExit
	Caption	خروج
CommandButton	Name	CmdAppend
	Caption	إضافة سجل
CommandButton	Name	CmdDelete
	Caption	حذف
CommandButton	Name	CmdFind
	Caption	بحث

اكتب شيفرة الخروج في حدث النقر على زر الخروج (CmdExit) .

احتفظ بالبرنامج وجربه .... لاحظ أن محتويات صناديق النصوص قد أخذت قيمها من قاعدة البيانات التي أنشأتها ، ويمكن التنقل بين محتويات القاعدة بنقر أزرار التنقل التابعة للكائن Data

❋ اكتب الشيفرة التالية في حدث النقر على إضافة سجل (CmdAppend) :

```
DataStore.Recordset.AddNew
```

التابع AddNew يتبع إلى خصائص الكائن Data وهو يقوم بإضافة سجل جديد إلى القاعدة .

❋ اكتب الشيفرة التالية في حدث النقر على زر حذف (CmdDelete) :

```
If MsgBox("هل أنت متأكد", _
        vbYesNo + 524288 + 1048576, "تأكيد") = vbYes Then
   DataStore.Recordset.Delete
   DataStore.Recordset.MoveNext
```

تقوم الشيفرة بإظهار رسالة تأكيد حذف ، وفي حال الموافقة تحذف السجل الحالي ، ثم تنتقل إلى السجل التالي .

عند نقر زر البحث يجب أن تظهر نافذة يدخل من خلالها المستخدم اسم ونوع الجهاز المطلوب البحث عنه ، فيتم الانتقال إلى السجل المطابق .

أنشأ نموذجاً جديداً ، واحفظه باسم Find.Frm في نفس دليل العمل ، وصممه وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmFind
	Caption	إدخال معايير البحث
	RightToLift	True
TextBox	Name	TxtFindName
	Text	اجعله فارغاً

TextBox	Name	TxtFindType
	Text	اجعله فارغاً
CommandButton	Name	CmdOk
	Caption	موافق
CommandButton	Name	CmdCancel
	Caption	إلغاء

سنحتاج إلى متحولين يستخدمان من قبل النموذجين معاً ، ولذلك علينا إنشاء برنامج مستقل Module .

أنشأ برنامج مستقل للمشروع واحفظه باسم Find.Bas في نفس دليل العمل ، ثم اكتب فيه الشيفرة التالية في قسم التصريح العام له :

```
Option Explicit
Global FindName, FindType
```

نكون بذلك قد عرفنا متحولين عامين يمكن استخدامهما من قبل النموذجين معاً .

* عد إلى النموذج الأساسي FrmStore وعرف في قسم التصريح العام له متحول باسم RecNum حيث سنخزن فيه رقم السجل الذي نحن فيه ، وذلك كما يلي :

```
Option Explicit
Dim RecNum As Integer
```

* أضف إجراء فرعي للنموذج باسم FindRecNum واكتب فيه الشيفرة التالية :

```
Public Sub FindRecNum()
Do While DataStore.Recordset.BOF = False
  DataStore.Recordset.MovePrevious
  RecNum = RecNum + 1
Loop
End Sub
```

يقوم الإجراء السابق بالانتقال لبداية السجلات ، ويضيف واحد إلى المتحول RecNum في كل خطوة انتقال وبالتالي سيحمل المتحول RecNum رقم السجل الذي كنا فيه ، أما التابع **BOF** فهو يعني الوصول لبداية الملف .

* أضف إجراء فرعي آخر للنموذج باسم GotoPrevRec واكتب فيه الشيفرة التالية :

```
Public Sub GotoPrevRec()
Dim I As Integer
   DataStore.Recordset.MoveFirst
   For I = 1 To RecNum - 1
       DataStore.Recordset.MoveNext
   Next I
End Sub
```

يقوم الإجراء السابق بالعودة إلى السجل الذي كنا فيه بعد أن يوجد رقمه .

* اكتب الشيفرة التالية في حدث النقر على زر البحث (CmdFind) :

```
Dim Found As Boolean
FrmFind.Show 1
RecNum = 0
Found = False
FindRecNum
Do While DataStore.Recordset.EOF = False
  If TxtName.Text = FindName And TxtType.Text = FindType Then
     Found = True
     Exit Do
```

```
  End If
  DataStore.Recordset.MoveNext
Loop
If Found = False Then
   MsgBox "لم يتم العثور على جهاز مطابق", 524288 + 1048576, "غير موجود"
   GotoPrevRec
End If
```

لاحظ أن الشيفرة السابقة تقوم بإظهار نموذج البحث الذي أنشأناه ، ثم توجد رقم السجل عن طريق استدعاء الإجراء FindRecNum ، بعدها تبدأ بالبحث عن السجل المطلوب ، وذلك بمطابقة المتحولين FindName و FindType مع كافة سجلات القاعدة وذلك عن طريق حلقة Do While ، أما تابع **EOF** فهو يدل على نهاية سجلات القاعدة . وفي حال لم يتم إيجاد سجل مطابق ، تظهر رسالة بذلك وتتم العودة للسجل الذي كنا فيه وذلك باستدعاء الإجراء GotoPrevRec

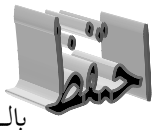
* اكتب الشيفرة التالية في حدث النقر على زر موافق (CmdOk) العائد للنموذج FrmFind :

```
Private Sub CmdOk_Click()
FindName = TxtFindName.Text
FindType = TxtFindType.Text
FrmFind.Hide
End Sub
```

تقوم الشيفرة السابقة بإسناد القيم التي أدخلها المستخدم إلى المتحولين FindName و FindType اللذين استخدما في زر البحث للنموذج الأساسي .

* اكتب الشيفرة التالية في حدث النقر على زر إلغاء (CmdCancel) العائد للنموذج FrmFind :

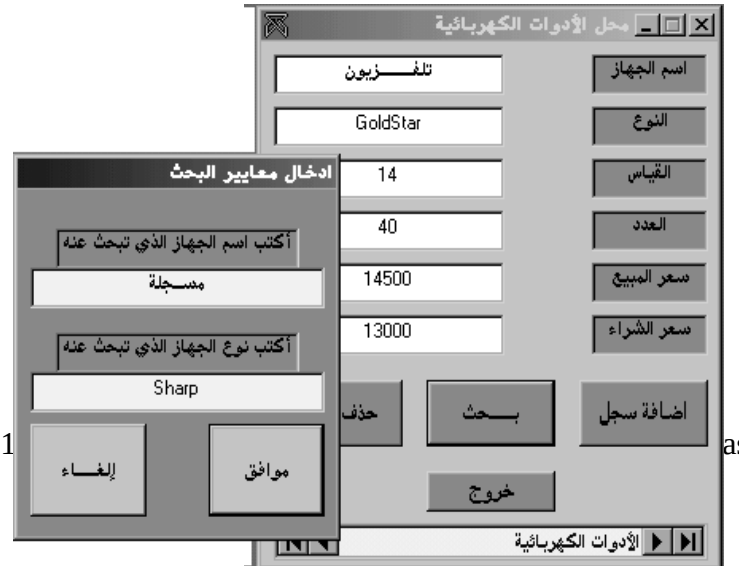
```
Private Sub CmdCancel_Click()
FrmFind.Hide
End Sub
```



بالبرنامج وجربه وتأكد من عمله بالشكل المطلوب ، ثم اصنع منه نسخة تنفيذية في مجلد العمل الحالي

C:\Vb6\MyExercise\Exer14\Store.Exe

أن قاعدة البيانات Store.Mdb يجب أن تتواجد دوماً في نفس مجلد العمل ...



The Explorer Program برنامج المستكشف

سنقوم في هذا البرنامج بإنشاء برنامج متشابه إلى حد ما مع برنامج المستكشف الذي يأتي مع نظام Windows

❋ أنشئ مشروعاً جديداً في المجلد Exer15 واحفظه باسم :

النموذج باسم C:\Vb6\MyExercise\Exer15\Explorer.Frm

المشروع باسم C:\Vb6\MyExercise\Exer15\Explore.Vbp

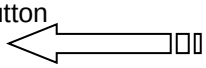
❋ سنحتاج في هذا البرنامج إلى كائنين جديدين هما : Explorer Tree و Explorer List لذلك قم بإضافة الملف Explorer. Ocx ولاحظ أن الكائنين أضيفا إلى صندوق الأدوات ....

صمم النموذج وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmExplorer
	Caption	المستكشف
	RightToLeft	True
Frame	Name	FrameTree
	BackColor	أبيض
	Caption	اجعله فارغاً
Explorer Tree	Name	ExpTree1
Explorer List	Name	ExpList1
Label	Name	LblPath
	Alignment	2-Center
	Caption	المسار الذي اخترته هو :
TextBox	Name	TxtPath
	Alignment	2-Center
	Text	اجعله فارغاً
Frame	Name	FrameOption
	Caption	طريقة عرض المحتويات
OptionButton	Name	OptLarge (1)

(1) قم برسم أزرار الخيارات Option Button الأربعة داخل كائن الـ Frame باستخدام طريقة السحب والإفلات وليس باستخدام النقر المزدوج ....

Object	Properties	Set
	Caption	رموز كبيرة
OptionButton	Name	OptSmall
	Caption	رموز صغيرة
OptionButton	Name	OptList
	Caption	قائمة
OptionButton	Name	OptWide
	Caption	تفاصيل
CommandButton	Name	CmdExit
	Caption	خروج



❋ اكتب شيفرة الخروج في حدث النقر على زر الخروج (CmdExit)

❋ اكتب الشيفرة التالية في الحدث TreeDataChanged التابع للكائن ExpTree1

```
Private Sub ExpTree1_TreeDataChanged()
   ExpList1.TreeDatas = ExpTree1.TreeDatas
End Sub
```

تنفذ الشيفرة السابقة عندما يقوم المستخدم بتغيير اسم القرص أو المجلد في شجرة المستكشف ، وتضمن هذه الشيفرة ظهور محتويات المجلد المشار إليه في قائمة المستكشف (ExpList1)

❋ اكتب الشيفرة التالية في الحدث GetFileName التابع للكائن ExpList1

```
Private Sub ExpList1_GetFileName()
  TxtPath.Text = ExpList1.FileName
End Sub
```

تنفذ الشيفرة السابقة عندما يقوم المستخدم بالنقر على أسماء أحد الملفات الواردة في قائمة المستكشف ...

❋ اكتب الشيفرة التالية في حدث النقر على زر الخيار (OptLarge) :

```
If OptLarge.Value = True Then
    ExpList1.View = 0
End If
```

❋ اكتب الشيفرة التالية في حدث النقر على زر الخيار (OptList) :

```
If OptList.Value = True Then
  ExpList1.View = 2
End If
```

❋ اكتب الشيفرة التالية في حدث النقر على زر الخيار (OptSmall) :

```
If OptSmall.Value = True Then
  ExpList1.View = 1
```

```
End If
```

* اكتب الشيفرة التالية في حدث النقر على زر الخيار (OptWide) :

```
If OptWide.Value = True Then
  ExpList1.View = 3
End If
```

إن خاصية View العائدة للكائن ExpList هي التي تحدد طريقة عرض المحتويات في قائمة المستكشف ، وهي تأخذ قيم محصورة بين (3 - 0) ، وكل رقم يدل على طريقة معينة للعرض ، كما يلـي :

قيمة الخاصة View	طريقة العرض
0	رموز كبيرة
1	رموز صغيرة
2	قائمة
3	تفاصيل

(**حول**) إلى البرنامج :

سنتعلم في هذه الفقرة كيفية الاستفادة من نماذج جاهزة تأتي مع لغة فيجول بيسيك ...

اختر أمر Add Form لإضافة نموذج جديد للبرنامج ، ومن مربع الحوار الذي ظهر اختر About Dialog ...

لقد تم إضافة نموذج جديد للمشروع ، وعليك الآن تغيير عناوين المسميات التي فيه بما يتلاءم مع طبيعة البرنامج لديك ، وبذلك نكون قد وفرنا عناء إنشاء نموذج يدوياً ...

لربط النموذج الجديد مع البرنامج ، أضف زر أمر جديد أسميه ((حول)) ، ثم اكتب فيه الشيفرة التالية :

```
Private Sub CmdAbout_Click()
    frmAbout.Show 1
End Sub
```

وجود زر System Info... في النموذج الجديد ، وهذا الزر يعرض معلومات عن النظام ومكونات الحاسب ، وتحوي شيفرة هذا الزر على توابع وتصاريح معقدة موجودة في مكتبات النظام الأساسية (DLL) ....

البرنامج وتأكد من صحة عمله ، ثم احتفظ به واصنع منه نسخة تنفيذية في دليل العمل ..

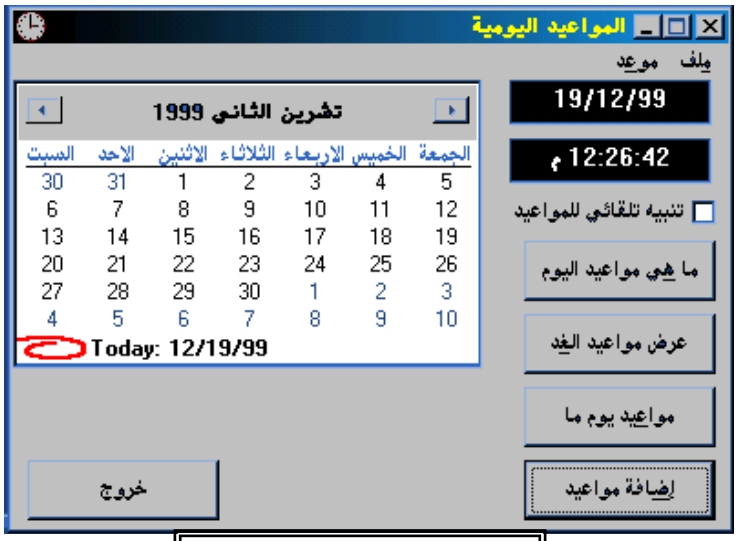
C:\Vb6\MyExercise\Exer15\Explorer.Exe

برنامج مفكرة المواعيد

The Agenda Program

برنامجنا الأخير هذا ، يعد من البرامج العملية والمفيدة جداً ، وهو برنامج شامل تقريباً لمعظم الأفكار والتعليمات التي مرت معنا خلال المرجع .... ولفهم فكرته وطريقة عمله ، سنلقي نظرة سريعة عليه ....

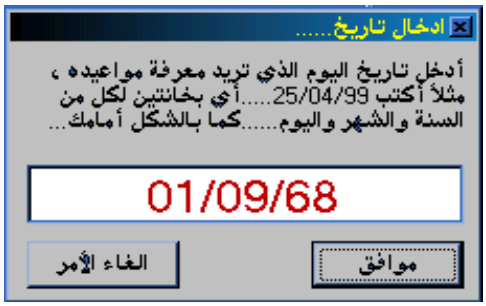
بتشغيل البرنامج تظهر الواجهة الأساسية له ، والتي تشمل على عدة اختيارات .....



الواجهة الأساسية للبرنامج

* بنقر زر [**ما هي مواعيد اليوم**] : سيظهر نموذج جديد يعرض لك مواعيد هذا اليوم ، ويمكنك التنقل بين المواعيد بنقر زر [**الموعد التالي**] ، وعندما يتم عرض جميع المواعيد ، تظهر لك رسالة تخبرك بذلك .....

* بنقـــــــــر زر [**عـــــــــــرض مواعيـــــــــــد الغـــــــــــد**] : يظهـــــــــــر لـــــــــك نفـــــــــس النموذج السابق ، ولكنه يحوي على مواعيد يوم الغد ، وأيضاً يمكنك التنقل بينها بنقر زر [**الموعد التالي**] ....



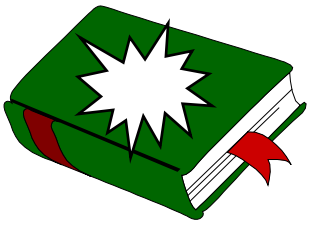
* بنقر زر [**مواعيد يوم ما**] : سيظهر مربع إدخال لإدخال تاريخ اليوم الذي تريد معرفة مواعيده ، وبعد تحديد التاريخ انقر زر موافق فيعرض لك مواعيد اليوم المحدد ، وفي حال عدم وجود مواعيد لذلك اليوم ، تظهر رسالة بذلك .....

* بنقر زر [**إضافة مواعيد**] : سيظهر مربع إدخال لإدخال الموعد الجديد ، وفيه يتم تحديد تاريخ الموعد باليوم والساعة وشرح عن طبيعته ..... وبعد تحديد هذه البيانات ، انقر زر [**إضافة**] ، فيتم إضافة هذا الموعد إلى سجل المواعيد السابقة ....

الأهم في هذا البرنامج هي وجود خانة اختيارية للتنبيه التلقائي للمواعيد ..... فإذا فعّلت هذه الميزة ستظهر لك رسالة تنبهك :

أنه في حال تشغيل هذه الميزة فسيتم تصغير البرنامج للحد الأدنى ، وسيقوم البرنامج بتنبيهك تلقائياً لمواعيدك لهذا اليوم قبل حينها بمدة معينة تحددها أنت .... فإذا نقرت زر نعم من هذه الرسالة سيظهر لك مربع إدخال تحدد فيه عدد الدقائق السابقة للتنبيه ... فمثلاً : إذا حددت مدة قدرها // **30** // فسيتم تنبيهك على مواعيدك قبل نصف ساعة من حينها ، وطبعاً يتم التنبيه طالما أنك لم تغلق البرنامج ، ومهما كانت الأعمال الأخرى التي تقوم بها ..

في جميع مراحل عمل البرنامج ، يتم اختبار شكل التاريخ والوقت الذي يقوم المستخدم بإدخاله ، فإذا لم يكن مطابقاً للشكل الصحيح ، فيتم تنبيه المستخدم بذلك .....



:

قبل البدء بالتنفيذ يجب إضافة الملفين التاليين إلى البرنامج :

MsComCTL.Ocx – MsComCT2.Ocx

* صمم النموذج الأساسي وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmAgenda
	Caption	المواعيد اليومية
	RightToLeft	True
MonthView	Name	MonthView1
	Enable	False
	ShowToDay	True
Label	Name	LblCurrentDate
	Alignment	2-Center
Label	Name	LblCurrentTime
	Alignment	2-Center
CommandButton	Name	CmdShow
	Caption	ما هي مواعيد اليوم
CommandButton	Name	CmdTomorrow
	Caption	عرض مواعيد الغد
CommandButton	Name	CmdWhatDate
	Caption	مواعيد يوم ما
CommandButton	Name	CmdAppend
	Caption	إضافة مواعيد
CommandButton	Name	CmdExit
	Caption	خروج
CheckBox	Name	ChkAlarm
	Alignment	1-Right Justify
	Caption	تنبيه تلقائي للمواعيد
Timer	Name	TmrAlarm
	Interval	1000

* **صمم قوائم** النموذج وفق الجدول التالي :

Caption	Name	Short Cut
ملف	MnuFile	
خروج....	MnuExit	Ctrl +X
موعد	MnuAppoint	
إضافة مواعيد....	MnuAppend	F12
عرض مواعيد الغد....	MnuTomorrow	Ctrl +T
ما هي مواعيد اليوم....	MnuToDay	Ctrl +D
مواعيد يوم ما....	MnuAnyDay	Ctrl +A

نتمكن من حفظ المواعيد لدينا بشكل دائم ، سنحتاج إلى وجود ملف يقوم بذلك ، لأنه وكما نعلم أن التخزين في ذاكرة الجهاز تعتبر عملية تخزين مؤقتة ، كونها تُفرغ بعد الخروج من البرنامج ، أما عملية حفظ البيانات بالملف فتعتبر عملية تخزين دائمة ، كونها لا تزول إلا إذا قام المستخدم بحذف الملف الحاوي على البيانات ..... وفي عملنا هذا نحتاج إلى ملف يحوي عدد من السجلات ، بحيث يمثل كل سجل موعد معين ، وكل سجل سيتألف من ثلاث حقول هي :

تاريخ الموعد (Appointment Date)

وقت المواعد (Appointment Time)

طبيعة الموعد أو شرح عنه (Appointment Description)

المشروع سيحوي على عدة نماذج ، وجميع هذه النماذج ستتعامل مع الملف المذكور ، فسنقوم بتعريف الملف والمتحولات اللازمة للبرنامج ضمن برنامج جزئي (**Module**) ..... لذلك : أضف برنامج جزئي ، واحفظه باسم **Agenda.Bas** ..... **وقبل** البدء بتعريف المتحولات سنتعرف على الأفكار التالية :

1-) تتيح جميع اللغات البرمجية كائن أو مفهوم برمجي يسمى **المصفوفات** ، والمصفوفة هي عبارة عن تجمع لعدد من البيانات أو السجلات تساعد كثيراً في تسريع إنجاز الأعمال ..... فمثلاً :

إذا كان لدي قائمة باسم مائة موظف ، و أريد تعريفها في برنامج ، فسيستغرق ذلك وقت كبير وعمليات شاقة ، إذ أنني في هذه الحالة سأحتاج لتعريف مائة متحول ..... لذلك أقوم بوضع هذه الأسماء كلها في مصفوفة واحدة بحيث تشمل المائة اسم دفعة واحدة .....

يتم تعريف المصفوفة بطريقة مشابهة لتعريف باقي المتحولات ، وبالشكل العام التالي :

Dim ArrayName (Number) As Type

حيث :

ArrayName : هي اسم المصفوفة .

Number : هو عدد العناصر التي ستحتويها المصفوفة .

Type : هو النمط المعرف لعناصرها (صحيح – حقيقي – محرفي .....) .

ففي مثالنا السابق مثلاً نحتاج لتعريف مصفوفة من مائة عنصر ، لذلك نعرفها كما يلي :

Dim Name (100) As String

2-) كي تتمكن جميع نماذج المشروع من التعامل مع متحول ما ، يتم تعريفه في البرنامج الجزئي باستخدام الكلمة المحجوزة Global ، بينما لاحظنا أن باقي عمليات تعريف المتحولات ، تتم باستخدام الكلمة المحجوزة Dim ....

3-) عندما نقوم بتعريف متحول من نوع محرفي (String) فيتم حجز أماكن طويلة له بالذاكرة (تصل إلى **64512** حرف) ، لذلك نقوم بتعريف المتحول وفقاً لحاجتنا له في البرنامج ، فمثلاً إذا أردنا تعريف متحول لتخزين رقم هاتف فيه ، فيكفي أن نعرفه بطول (20) حرف ..... ويكون ذلك كما يلي :

Dim Phone As String * 20

أدخل إلى ترميز البرنامج الجزئي (Agenda.Bas) وعرف فيه المتحولات التالية :

```
Option Explicit
Type Agenda
    Date As String * 20
    Time As String * 10
    Discr As String * 1000
End Type
Global WhatDate As String
Global ProgPath As String
Global AppoinTime  (20) As String * 10
Global AppoinDiscr  (20) As String * 1000
Global MyRecAgenda As Agenda
Global FileNum As Integer
Global RecLength As Integer
Global RecLast As Integer
Global FileName As String
```

سنتعرف على هذه المتحولات تباعاً عند استخدامنا لباقي نماذج المشروع

سنقوم الآن بتجهيز عملية إنشاء أو فتح الملف الحاوي على المواعيد ، وكون الملف عام لكافة النماذج ، فيجب تعريفه ضمن البرنامج الجزئي حصراً ، لذلك أضف إجراء إلى البرنامج باستخدام أمر (Add Procedure من قائمة Tools) ، واسميه / **Main** / ثم اكتب فيه الشيفرة التالية :

```
Public Sub Main()
ProgPath = App.Path
If Right(ProgPath, 1) <> "\" Then
    ProgPath = ProgPath + "\"
End If
FileName = ProgPath + "MyAgendaFile.Sia"
FrmAgenda.Show
Do While DoEvents
    FrmAgenda.LblCurrentTime.Caption = Time
    FrmAgenda.LblCurrentDate.Caption = Date
Loop
End Sub
```

نلاحظ أن الشيفرة السابقة تفتح الملف (MyAgendaFile.Sia) في دليل العمل الحالي ، وأيضاً تضع التاريخ والوقت في اللافتتين (LblCurrentTime – LblCurrentDate) ، طالما أن البرنامج مفتوحاً .

لجعل الإجراء السابق (Main) هو الذي ينفذ حال تشغيل البرنامج ، اختر أمر Agenda Properties من قائمة Project ... وفي خانة Startup Object حدد اسم الإجراء Sub Main .....

بإمكانك الاحتفاظ بالمشروع وتجريبه .... فتلاحظ أنه يعمل بشكل صحيح ، إذ قام بإنشاء الملف المطلوب في دليل العمل الحالي ....

إضافة سجلات المواعيد إلى الملف :

تتطلب عملية إضافة المواعيد إلى الملف ، إنشاء نموذج جديد يمكنني من ذلك .... لذلك أضف نموذجاً جديداً ، واحفظه باسم **Append .Frm** **وطبعاً** سيشمل كل موعد على ثلاث حقول (تاريخ الموعد – وقته – طبيعته) .

صمم النموذج / Append / وفقاً للجدول التالي :

Object	Properties	Set
Form	Name	FrmAppend
	BorderStyle	1-Fixed Single
	Caption	إضافة مواعيد جديدة
	RightToLeft	True
Label	Name	LblDate
	Caption	تاريخ الموعد
Label	Name	LblTime
	Caption	وقت الموعد
Label	Name	LblDiscr
	Caption	طبيعة الموعد
TextBox	Name	TxtDate
	Alignment	2-Center
	MaxLength	8
TextBox	Name	TxtTime
	Alignment	2-Center
	MaxLength	10
TextBox	Name	TxtDiscr
	Alignment	1-Right Justify
	MultiLine	True
CommandButton	Name	CmdAppend
	Caption	إضافة
CommandButton	Name	CmdCancel
	Caption	إغلاق

❋ صمم قوائم النموذج وفق الجدول التالي :

Caption	Name	Short Cut
ملف	MnuFile	
إضافة....	MnuAppend	F12
إغلاق....	MnuClose	Ctrl +W

❋ اكتب الشيفرة التالية في حدث النقر على زر الإضافة (**CmdAppend**) :

```
Private Sub CmdAppend_Click()
RecLength = Len(MyRecAgenda)
FileNum = FreeFile
Open FileName For Random As FileNum Len = RecLength
RecLast = FileLen(FileName) / RecLength
MyRecAgenda.Date = TxtDate.Text
MyRecAgenda.Time = TxtTime.Text
MyRecAgenda.Discr = TxtDiscr.Text
Put #FileNum, RecLast + 1, MyRecAgenda
Close FileNum
End Sub
```

تلاحظ أن الشيفرة السابقة تقوم بتخزين قيم صناديق النصوص (التاريخ – الوقت – طبيعة الموعد) في سجل ، ثم تخزن هذا السجل في الملف بعد معرفة طول السجل الواحد ، وقد مر معنا هذا النوع من الملفات (Random) ، وتعلمنا كيفية التعامل معه في البرنامج التاسع (**Company**) .

❋ اكتب الشيفرة التالية في حدث النقر على زر الإغلاق (Cancel) :

```
Private Sub CmdClose_Click()
  Unload Me
End Sub
```

بقي هناك شيفرة واحدة لإظهار النموذج السابق (Append) ، لذلك عد إلى النموذج الأساسي (Agenda) ، واكتب الشيفرة التالية في حدث النقر على زر إضافة مواعيد (CmdAppend) :

```
Private Sub CmdAppend_Click()
  FrmAppend.Show 1
End Sub
```

❋ معرفة مواعيد اليوم :

سنقوم الآن بكتابة الشيفرات اللازمة لمعرفة مواعيد اليوم ، وتتم هذه العملية بأن يقوم البرنامج بقراءة كافة سجلات الملف ، ثم مطابقة السجلات التي يتطابق تاريخها مع تاريخ اليوم ، ثم عرضها عن طريق نموذج جديد خاص لهذا الغرض ، لذلك يتوجب علينا أولاً إضافة نموذج جديد .....

أضف نموذجاً جديداً ، واحفظه باسم ToDay.Frm ، ثم صممه وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmToDay
	BorderStyle	1-Fixed Single
	Caption	مواعيد اليوم
	RightToLeft	True
Label	Name	LblTime
	Caption	وقت الموعد
Label	Name	LblDiscr
	Caption	طبيعة الموعد
TextBox	Name	TxtTimeToDay
	Alignment	2-Center
TextBox	Name	TxtDiscrToDay
	Alignment	2-Center
CommandButton	Name	CmdNext
	Caption	الموعد التالي
CommandButton	Name	CmdClose
	Caption	إغلاق

❋ صمم قوائم النموذج وفق الجدول التالي :

Caption	Name	Short Cut
ملف	MnuFile	
الموعد التالي....	MnuNext	F6
إغلاق....	MnuClose	Ctrl +W

قبل البدء بكتابة شيفرات معرفة مواعيد اليوم ، سنتعرف على فكرة التوابع (**Functions**)

تكلمنا فيما سبق عن الإجراءات ، وقلنا أن فكرتها تشبه فكرة الماكروات في البرامج المكتبية ، فهي تفيد في عدم تكرار الخطوات المتشابهة أكثر من مرة في البرنامج الواحد ، ويتم استدعاء الإجراء بكتابة اسمه فقط ، وفكرة التوابع تشبه إلى حد كبير فكرة الإجراءات ، فهي تفيد أيضاً في تفادي إعادة كتابة شيفرات طويلة ومتشابهة أكثر من مرة ، وتختلف التوابع عن الإجراءات بأن الأولى تعيد قيمة معينة ، بينما لا تقوم الإجراءات بذلك ، ولفهم التوابع بشكل أكبر سنورد مثال بسيط عن كيفية استخدامها :

بفرض أريد إنشاء برنامج يقوم المستخدم من خلاله بإدخال ثلاثة أعداد ، فيقوم البرنامج بحساب العدد الأكبر بينها ، إن فكرة هذا البرنامج سهلة للغاية ، حيث يتم اختبار العدد الأكبر باستخدام حلقات الاختبار (**IF**) ، وكون المستخدم سيقوم بإدخال أكثر من مجموعة أعداد ، فمن الصعب كتابة شيفرات الاختبار هذه أكثر من مرة ، لذلك أضعها في تابع معين ، فيقوم التابع باختبارها وإرجاع الأكبر بينها .

* إضافة التابع تكون باستخدام الأمر (Add Procedure) من قائمة (Tools) ، ومن مربع الحوار الذي يظهر أختار خيار (Function) ، وأحدد له اسماً ، وسأسميه في مثالنا بـ Max ، وبعد أن يظهر التابع في النموذج أغير ترويسته لتتوافق مع الغرض المطلوب ، فأكتب :

```
Public Function Max (Num1, Num2, Num3 As Integer, IsMax As Integer)
```

ثم أكتب فيه شيفرات الاختبار للأعداد الثلاثة (Num1, Num2, Num3) وأسند الأكبر بينهم للمتحول IsMax ، ولاستدعاء هذا التابع من جسم البرنامج ، أدخل قيم الأعداد الثلاثة ، ثم أكتب :

```
Call Max (Num1, Num2, Num3,IsMax)
Print IsMax
```

فيستدعي البرنامج التابع ويحسب الأكبر بينهم ، ثم تطبع عبارة Print IsMax قيمته على النموذج ....

وبذلك كلما أردت حساب الأكبر بين قيم جديد ، أقوم باستدعاء التابع فقط . **{{ انتهى المثال }}**

* سنقوم في برنامجنا بتعريف متحول منطقي (Boolean) لاختبار وجود مواعيد لهذا اليوم أم لا ، وأيضاً سنستخدم تابع ، نرسل له تاريخ يوم ما ، ، فإذا وجد مواعيد ، قام بتخزينها في المصفوفة ، وأسند قيمة الصح (True) للمتحول المنطقي ، وإذا لا فيسند قيمة الخطأ (False) للمتحول المنطقي .

* كتابة الشيفرات اللازمة لمعرفة المواعيد :

أضف تعريف متحول منطقي باسم Found إلى قسم التصريح العام في النموذج الأساسي (Agenda) كما يلي :

```
Option Explicit
Dim Found As Boolean
```

* أضف تابع جديد لترميز النموذج باسم WhatAppoinDay ثم عدل ترويسته ، وأكتب فيه الشيفرات التالية :

```
Public Function WhatAppoinDay(WhatDate As String)
Dim Counter As Integer
Dim AppoinCount As Integer
Dim RecToSearsh As Agenda
EmptyThisArray
RecLength = Len(MyRecAgenda)
FileNum = FreeFile
AppoinCount = 0
Open FileName For Random As FileNum Len = RecLength
RecLast = FileLen(FileName) / RecLength
For Counter = 1 To RecLast
   Get #FileNum, Counter, RecToSearsh
   If Trim(RecToSearsh.Date) = WhatDate Then
      AppoinTime(AppoinCount) = Trim(RecToSearsh.Time)
      AppoinDiscr(AppoinCount) = Trim(RecToSearsh.Discr)
      AppoinCount = AppoinCount + 1
   End If
Next Counter
Close FileNum
For Counter = 0 To 20
   If Trim(AppoinTime(Counter)) <> "" Then
      Found = True
      Exit For
   End If
Next Counter
End Function
```

* **تلاحظ** في شيفرة التابع وجود إجراء (EmptyThisArray) ، ويقوم هذا الإجراء بتفريغ المصفوفتين اللتان سنخزن فيهما {{ أوقات مواعيد اليوم(AppoinTime) ، وشرح عنها (AppoinDiscr) }} وسنكتب شيفرة هذا الإجراء بعد قليل [1].

* **تلاحظ** أن التابع يقوم بفتح الملف ، واختبار كافة سجلاته ، فإذا وجد سجل مطابق تاريخه للتاريخ الذي أرسلناه له ، قام بتخزين هذا السجل في المصفوفة ، وإذا لا تجاوزه للسجل التالي ، ثم تقوم حلقة / **FOR** / التالية باختبار المصفوفة التي تم تخزين السجلات فيها (AppoinTime) ، فإذا وجدت فيها سجل واحد على الأقل لا يساوي الفراغ ، فهذا يعني وجود مواعيد للتاريخ المحدد ، فتسند قيمة الصح للمتحول **FOUND** وتنهي الحلقة ، وسنلاحظ بعد قليل كيف سنختبر هذا المتحول المنطقي.....

* **إضافة** إجراء تفريغ المصفوفتين للنموذج :

أضف إجراءاً جديداً للنموذج ، واسميه باسم EmptyThisArray ، ثم اكتب فيه الشيفرة التالية :

[1] تعتبر عملية تفريغ المصفوفة من العمليات الهامة أثناء العمل ، لأننا عندما نقوم بالتصريح عن متحول أو مصفوفة في الذاكرة ، فإن الحاسب يسند قيم عشوائية لها ، وبالتالي قد تظهر لنا سلاسل محرفية لم نقم نحن بإدخالها ، مما يؤدي إلى أخطاء في عمل البرنامج ، أما عملية التفريغ فتقوم بإسناد فراغات لكامل محتوى المصفوفة .

```
Public Sub EmptyThisArray()
Dim AA As Integer
For AA = 0 To 20
    AppoinTime(AA) = String$(10, " ")
    AppoinDiscr(AA) = String$(1000, " ")
Next AA
End Sub
```

إن تابع String$ يسند عدد معين من المحارف إلى متحول محرفي ، بعد أن نحدد له اسم المتحول، وعدد المحارف، وما هو المحرف ، فمثلاً الشيفرة التالية:

SS = String$(15,"N")

تعني إسناد خمس عشرة حرف /N/ إلى المتحول المحرفي SS

❋ اكتب الشيفرة التالية في حدث النقر على زر مواعيد اليوم (CmdShow) :

```
Private Sub CmdShow_Click()
Call WhatAppoinDay(Date)
If Found = True Then
  FrmToDay.Show 1
Else
  MsgBox "لا توجد مواعيد لهذا اليوم.....", vbOKOnly  + 524288 + 1048576
End If
End Sub
```

❋ **تلاحظ** أن الشيفرة السابقة تستدعي التابع بعد أن ترسل له قيمة تاريخ اليوم (DATE) ، ثم تختبر قيمة المتحول المنطقي Found فإذا كان صحيحاً يظهر النموذج FrmToDay وإذا لا تظهر رسالة تخبرك بذلك .....

❋ لم ننتهي بعد من إنجاز هذه العملية ، فنحن حتى الآن قمنا بمعرفة مواعيد اليوم وتخزينها بالمصفوفة ، ولإظهارها سننتقل إلى النموذج FrmToDay لإكمال كتابة الشيفرات اللازمة .....

❋ انتقل إلى النموذج المذكور واكتب في قسم التصريحات العامة له تعريف المتحولات التالية :

```
Option Explicit
Dim NewArrayTime(20) As String * 10
Dim NewArrayDiscr(20) As String * 1000
Dim NewArrCount As Integer
Dim NextAppoint As Integer
```

❋ أضف إجراء جديد للنموذج باسم EmptyArray وسنستخدمه لتفريغ المصفوفتين الجدد ، واكتب فيه الشيفرة :

```
Public Sub EmptyArray()
Dim A As Integer
For A = 0 To 20
    NewArrayTime(A) = String$(10, " ")
    NewArrayDiscr(A) = String$(1000, " ")
Next A
End Sub
```

❋ سنقوم الآن بإضافة إجراء جديد يختبر إذا ما كانت إحدى سجلات المصفوفتين الأساسيتين (AppoinTime - AppoinDiscr) فارغة ، ثم ينقل السجلات الغير فارغة فقط منهما ، إلى المصفوفتين الجدد اللتين عرفناهما في النموذج الحالي ....

❋ **أضف** إجراءاً جديداً للنموذج باسم TestArrayForTime ، واكتب فيه الشيفرة التالية :

```
Public Sub TestArrayForTime()
Dim ArrCount As Integer
NewArrCount = 0
For ArrCount = 0 To 20
    If Trim(AppoinTime(ArrCount)) <> "" Then
          NewArrayTime(NewArrCount) = AppoinTime(ArrCount)
          NewArrayDiscr(NewArrCount) = AppoinDiscr(ArrCount)
          NewArrCount = NewArrCount + 1
    End If
Next ArrCount
End Sub
```

❋ بعد أن أصبح الإجراءان جاهزان ، اكتب الشيفرة التالية في قسم تحميل النموذج الحالي (FrmToDay_ Load) :

```
Private Sub Form_Load()
EmptyArray
TestArrayForTime
NextAppoint = 1
FrmToDay.Caption = FrmToDay.Caption & "    " & Str(Date)
TxtTimeToDay.Text = NewArrayTime(0)
TxtDiscrToDay.Text = NewArrayDiscr(0)
End Sub
```

سنستخدم المتحول NextAppoint لمشاهدة باقي مواعيد اليوم ، عند نقر زر / الموعد التالي /

❋ اكتب الشيفرة التالية في حدث النقر على زر الموعد التالي :

```
Private Sub CmdNext_Click()
TxtTimeToDay.Text = NewArrayTime(NextAppoint)
TxtDiscrToDay.Text = NewArrayDiscr(NextAppoint)
NextAppoint = NextAppoint + 1
If NextAppoint > NewArrCount Then
   MsgBox "لا توجد مواعيد اضافية" + Chr(13) + "يمكنك مشاهدتها مرة أخرى بعد نقر زر موافق", 0 + 524288 + 1048576
   NextAppoint = 0
End If
End Sub
```

تلاحظ أن الشيفرة السابقة تسند قيمة أول سجل في المصفوفة إلى صناديق النصوص ، وبنقرة أخرى للزر ، تزداد قيمة العداد واحد وتسند قيمة السجل التالي لصناديق النصوص ،وهكذا .... وعندما تنتهي السجلات ، تظهر رسالة بذلك ، وتعاد قيمة العداد للصفر لإتاحة إمكانية مشاهدتها من جديد .....

❋ اكتب الشيفرة التالية في حدث النقر على زر الإغلاق للنموذج الحالي :

```
Private Sub CmdClose_Click()
  Unload Me
End Sub
```

نكون انتهينا من كتابة الشيفرات اللازمة لعمل الزر / **ما هي مواعيد اليوم** / ، ويمكنك حفظ البرنامج وتجريبه للتأكد من صحة عمله .....

عد إلى النموذج الأساسي (AGENDA)

* اكتب الشيفرة التالية في حدث النقر على زر (عرض مواعيد الغد) :

```
Private Sub CmdTomorrow_Click()
Found = False
WhatDate = Date + 1
Call WhatAppoinDay(WhatDate)
If Found = True Then
    FrmToDay.Caption = "مواعيد الغد"
    FrmToDay.Show 1
Else
    MsgBox "لا توجد مواعيد ليوم الغد.....", vbOKOnly + 524288 + 1048576
End If
End Sub
```

* تقوم الشيفرة السابقة بإرسال قيمة تاريخ الغد إلى التابع ، وفي حال وجود مواعيد له ، يظهر نموذج الإظهار ، وإذا لا ، تظهر رسالة تخبرك بذلك .

تنفيذ عملية البحث عن مواعيد يوم ما يدخله المستخدم :

سنحتاج الآن إلى نموذج جديد يدخل من خلاله المستخدم تاريخ يوم ما للبحث عن وجود مواعيد فيه ، لذلك أضف نموذجاً جديداً للمشروع ، وصممه وفق الجدول التالي :

Object	Properties	Set
Form	Name	FrmInputDate
	BorderStyle	1-Fixed Single
	Caption	إدخال تاريخ
Label	Name	Label1
	Caption	أدخل تاريخ اليوم الذي تريد معرفة مواعيده ، مثلاً أكتب 99/04/25.....أي بخانتين لكل من السنة والشهر واليوم.....كما بالشكل أمامك...
TextBox	Name	TxtDate
	Alignment	2-Center
	MaxLenght	8
CommandButton	Name	CmdOk
	Caption	موافق
CommandButton	Name	CmdCancel
	Caption	إلغاء الأمر

* اكتب الشيفرة التالية في حدث النقر على زر موافق للنموذج الجديد :

```
Private Sub CmdOk_Click()
WhatDate = TxtDate.Text
Unload Me
End Sub
```

* اكتب الشيفرة التالية في حدث النقر زر على إلغاء الأمر للنموذج السابق :

```
Private Sub CmdCancel_Click()
WhatDate = "NoInput"
Unload Me
End Sub
```

إذا نقر المستخدم زر إلغاء الأمر ، تسند الشيفرة السابقة قيمة "NoInput" للمتحول WhatDate وسنناقش هذه الحالة لاحقاً ....

عد إلى النموذج الأساسي (AGENDA)

* اكتب الشيفرة التالية في حدث النقر على زر (مواعيد يوم ما) :

```
Private Sub CmdWhatDate_Click()
Found = False
FrmInputDate.Show 1
If WhatDate = "NoInput" Then
  Exit Sub
End If
Call WhatAppoinDay(WhatDate)
If Found = True Then
  FrmToDay.Caption = "مواعيد ..." & WhatDate
  FrmToDay.Show 1
Else
  MsgBox "لا توجد مواعيد ليوم....." + WhatDate, vbOKOnly + 524288 + 1048576
End If
End Sub
```

تختبر الشيفرة قيمة المتحول WhatDate فإذا كانت تساوي "NoInput" فهذا يعني أن المستخدم نقر على زر إلغاء الأمر ، فيتم الخروج من الإجراء ، و إلا فيتم استدعاء التابع وفقاً للتاريخ الذي أدخله المستخدم .....

* إعداد عملية التنبيه التلقائي للمواعيد :

قلنا أن البرنامج يحوي على خيار / **تنبيه تلقائي للمواعيد** / ، ففي حال اختاره المستخدم ، سيتم تصغير البرنامج للحد الأدنى وتنبيه المستخدم على مواعيده قبل مدة يحددها هو ، ولإنجاز هذه العملية سنستخدم المؤقت الذي يقوم بفحص السجلات كل مدة معينة ، ولا ننسى أنه في حال تكبير نافذة البرنامج يجب أن يؤدي إلى إيقاف عملية التنبيه التلقائي ، أي أن هذه الميزة يجب ألا تعمل إلا إذا كان البرنامج مصغراً للحد الأدنى ...

لإنجاز العملية السابقة سنحتاج إلى متحولين جدد ، أحدهما للإيعاز ببدء أو توقف عمل المؤقت ، والآخر لتحديد المدة التي يرغب المستخدم بالتنبيه على الموعد قبلها ....

* أضف تعريف المتحولين المذكورين إلى قسم التصريحات العامة للنموذج الأساسي (AGENDA) :

```
Dim RunTimer As Boolean
Dim BeforeTime As Integer
```

سنحتاج أيضاً إلى إجراء يقوم من خلاله المستخدم بتحديد المدة السابقة للتنبيه .

* أضف إجراء جديد للنموذج باسم ChoiceTime واكتب فيه الشيفرة التالية :

```
Public Sub ChoiceTime()
On Error GoTo InputError
BeforeTime = InputBox("حدد الوقت الذي تريد من البرنامج تنبيهك على مواعيدك قبله" _
("تحديد الوقت بالدقائق", "20" , " مثلا اذا حددت 20 فسيتم تنبيهك قبل ثلث ساعة" + Chr(13) +
Call WhatAppoinDay(Date)
FrmAgenda.WindowState = 1
Exit Sub
InputError:
   ChkAlarm.Value = 0
   FrmAgenda.WindowState = 0
   Exit Sub
End Sub
```

تنفذ شيفرة اللافتة InputError إذا نقر المستخدم على زر Cancel من صندوق الإدخال الذي ظهر ، وفي هذه الحالة يجب إعادة وضع النافذة إلى حالتها العادية ، و إلغاء تفعيل خانة التنبيه التلقائي ....

* اكتب الشيفرة التالية في حدث النقر على صندوق الاختيار (تنبيه تلقائي للمواعيد)

```
Private Sub ChkAlarm_Click()
If ChkAlarm.Value = 1 Then
  If MsgBox("اذا اخترت هذا الخيار فسيقوم البرنامج بتنبيهك تلقائياً" _
      + Chr(13) + " _  للمواعيد قبل حين موعدها بمدة تختارها أنت"
      + Chr(13) + " _ " فإذا أردت أن يكون هذا الخيار فعالاً سيتم تصغير "
      + Chr(13) + "_ "البرنامج للحد الأدنى وعندما تقوم بتكبيره ستلغى ميزة التنبيه التلقائي"
      + Chr(13) + " _  "ويمكنك إعادة تفعيلها من جديد"
      + Chr(13) + "_ "اذا أردت تشغيل هذه الميزة فانقر زر نعم ، ثم حدد مباشرة المدة"
      + Chr(13) + " _ ".....من الرسالة التي ستظهر لك بعد نقرك زر نعم"
      + Chr(13) + "_ "هل تريد تشغيل ميزة التنبيه التلقائي.....؟"
      , vbYesNo + 524288 + 1048576, "اقرأ بعناية") = vbYes Then
         ChoiceTime 'إجراء لتحديد الوقت
        RunTimer = True
   Else:  ChkAlarm.Value = 0
   End If
End If
End Sub
```

إذا نقرت ميزت التنبيه التلقائي وفعلتها ، فستقوم الشيفرة السابقة بإظهار رسالة تنبيه بأن البرنامج سيصغر للحد الأدنى، ومع الرسالة زري / نعم – لا / ، فإذا اخترت نعم يستدعى الإجراء ChoiceTime الذي يمكنك من إدخال الوقت السابق للتنبيه ، ثم يأخذ المتحول RunTimer القيمة True مما يعني الإيعاز ببدء عمل المؤقت كما ستلاحظ في شيفرة المؤقت التي سنكتبها حالاً ....

* اكتب الشيفرة التالية في المؤقت TmrAlarm :

```
Private Sub TmrAlarm_Timer()
Dim TmrCount As Integer
Dim Answer As Integer
If RunTimer = True Then
  If FrmAgenda.WindowState <> 1 Then
    RunTimer = False
    ChkAlarm.Value = 0
  End If
  For TmrCount = 0 To 20
    If Trim(AppoinTime(TmrCount)) <> "" Then
          If Minute(Trim(AppoinTime(TmrCount))) = Minute(Time) + BeforeTime Then
                MsgBox "لديك موعد بعد  " +  Str(BeforeTime) + " دقيقة..." + Chr(13) + _
                "انقر زر موافق لمشاهدته.....  ", vbOKOnly + 524288 + 1048576, "انتبه"
                FrmAgenda.WindowState = 0
                RunTimer = False
                CmdShow_Click
          End If
    End If
  Next TmrCount
End If
End Sub
```

قد تبدو الشيفرة السابقة متشعبة قليلاً ، لكن فكرتها بسيطة .... إذ أنها تختبر كل ثانية قيمة المتحول RunTimer ، وحالة النموذج (إن كانت مصغرة للحد الأدنى) ، وفي حال إلغاء تفعيل أحد الحالتين ، يتوقف عمل المؤقت ، وتكبر نافذة النموذج إلى حجمها الطبيعي ..... ، وطالما أن المؤقت في حالة العمل فإنه يختبر سجلات المصفوفة ، والتي تحوي على مواعيد اليوم ، فإذا وجد أن توقيت أحد المواعيد أكبر من الوقت الحالي بمدة مقدارها قيمة (BeforeTime) ، ظهرت رسالة تنبهك بالموعد ، وألغيت ميزة التنبيه التلقائي ، وظهر نموذج عرض المواعيد للاطلاع على الموعد .

كما لاحظت من خلال الشيفرات السابقة ، أن تابع WhatAppoinDay الذي يقوم بتخزين مواعيد اليوم بمصفوفة لا ينفذ إلا إذا طلبته بنقر أحد الأزرار ، ولكنك إذا نقرت ميزة التنبيه التلقائي قد لا يكون هذا التابع قد أُستدعي ، وحينها لن يعمل البرنامج ، لذلك سنقوم بتشغيل التابع عند تشغيل البرنامج مباشرةً .....

* اكتب الشيفرة التالية في قسم تحميل النموذج (Form_ Load) :

```
Private Sub Form_Load()
BeforeTime = 20
RunTimer = False
Call WhatAppoinDay(Date)
End Sub
```

يتم أولاً إعطاء قيمة افتراضية للمدة السابقة للتنبيه وهي / 20 / دقيقة ، ويتم إيقاف تشغيل المؤقت ، واستدعاء تابع معرفة المواعيد اليومية لتكون جاهزة في الذاكرة .....

* اكتب الشيفرة التالية في حدث النقر على زر الخروج (CmdExit) :

```
Private Sub CmdExit_Click()
If MsgBox ("هل حقا تريد الخروج من البرنامج....؟", vbYesNo + 524288 + 1048576, "تأكد") = vbYes Then
  End
End If
End Sub
```

* اكتب الشيفرة التالية في حدث إلغاء تحميل النموذج (Form _Unload) :

```
Private Sub Form_Unload(Cancel As Integer)
If MsgBox("هل حقا تريد الخروج من البرنامج....؟", vbYesNo + 524288 + 1048576, "تأكد") = vbNo Then
   Cancel = 1
End If
End Sub
```

تلاحظ أن الشيفرتين السابقتين تظهران رسالة تأكيد خروج إذا نقر المستخدم على زر الإغلاق ✕ أو في حال النقر على زر الخروج (**EXIT**)

أصبح البرنامج جاهزاً للعمل بعد أن انتهت كامل شيفراته ، احتفظ به ، وجربه من كافة نواحيه كالإضافة والتنبيه وعرض المواعيد .....الخ

سنقوم الآن بتحسين عمل البرنامج ، بأن نضع شروط اختبار الصحة للتاريخ والوقت ..... فمثلاً جرب العملية التالية :

شغل البرنامج وانقر زر { **إضافة مواعيد** } .

اكتب في خانة { **تاريخ الموعد** } تاريخ بشكل أو قيم خاطئة : مثلاً (**35/18/99**) ، أو اكتب فيه حروف بين الأرقام، وانقر زر { **إضافة** } ، فتلاحظ أن البرنامج أضاف السجل إلى الملف على الرغم من أن التاريخ خاطئ .....

تفادياً للعملية السابقة ، سنقوم الآن بتصميم توابع تختبر شكل الوقت والتاريخ إذا كان صحيحاً أم لا ، وكون البرنامج يتعامل مع التاريخ والوقت في كل نماذجه ، سنضع هذين التابعين في البرنامج العام (**AGENDA.BAS**) .

قبل إضافة التابعين سنلقي نظرة على تابع MID :

يقوم تابع Mid باقتطاع قسم من سلسلة محرفية بعد تحديد الموقع والسلسلة وعدد المحارف ، ويأخذ الشكل العام التالي :

MyVariable = MID (MyString , Position , Num)

حيث : MyString : هو السلسلة المحرفية المراد الاقتطاع منها .

Position : هو الموقع الذي سيتم منه الاقتطاع .

Num : هو عدد المحارف المراد اقتطاعها .

MyVariable : هو المتحول المحرفي الذي ستُسند إليه المحارف التي اقتطعناها .

* أضف تابع جديد إلى البرنامج العام (**AGENDA.BAS**) باسم TestDateIfPerfect ثم غير ترويسته واكتب فيه الشيفرة اللازمة لذلك كما يلي :

```
Public Function TestDateIfPerfect (AnyDate As String, Perfect As Boolean)
Dim Slash1 As String * 1
Dim Slash2 As String * 1
Dim Day As String * 2
Dim Month As String * 2
Dim Year As String * 2
On Error GoTo DateError
Slash1 = Mid (AnyDate, 3, 1)
Slash2 = Mid (AnyDate, 6, 1)
Day = Mid (AnyDate, 1, 2)
Month = Mid (AnyDate, 4, 2)
Year = Mid (AnyDate, 7, 2)
If Slash1 <> "/" Or Slash2 <> "/" Then
  Perfect = False
  Exit Function
End If
If Day > 31 Or Month > 12 Or Year > 99 Then
  Perfect = False
  Exit Function
End If
Exit Function
DateError:
  Perfect = False
  Exit Function
End Function
```

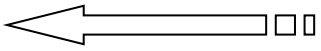
لا ننسى أن التاريخ يجب أن يأخذ شكلاً عاماً واحداً ، بحيث ندخله بشكل خانتين لكل من اليوم والشهر والسنة ، مثلاً : **{{ 25/04/76 }}** ، وفي حال تم الإدخال غير ذلك ، يعتبر الشكل خاطئاً ، وتقوم شيفرة التابع السابق باختبار جميع أجزاء شكل التاريخ ، فإذا وجدت أياً منها خاطئ ، أسندت القيمة False إلى المتحول Perfect وأنهت التابع ، وسنقوم من جسم البرنامج باختبار قيمة المتحول Perfect .

لاحظ أن محتوى اللافتة DateError سينفذ إذا أدخلت محارف بدلاً عن الأرقام في التاريخ .

سنضيف الآن تابع جديد لاختبار شكل الوقت :

* أضف تابع جديد إلى البرنامج العام (**AGENDA.BAS**) باسم TestTimeIfPerfect ثم غير ترويسته واكتب فيه الشيفرة اللازمة لذلك كما يلي :

```
Public Function TestTimeIfPerfect(AnyTime As String,Perfect As Boolean)
Dim Point1 As String * 1
Dim Point2 As String * 1
Dim Hour As String * 2
Dim Minute As String * 2
Dim Second As String * 2
Dim Noon As String * 1
On Error GoTo TimeError
Point1 = Mid(AnyTime, 3, 1)
Point2 = Mid(AnyTime, 6, 1)
Hour = Mid(AnyTime, 1, 2)
Minute = Mid(AnyTime, 4, 2)
Second = Mid(AnyTime, 7, 2)
```



```
Noon = Mid(AnyTime, 10, 1)
If Point1 <> ":" Or Point2 <> ":" Then
  Perfect = False
  Exit Function
End If
If Hour > 12 Or Minute > 59 Or Second > 59 Then
  Perfect = False
  Exit Function
End If
If Noon <> "ص"  And  Noon <> "م" Then
  Perfect = False
  Exit Function
End If
Exit Function
TimeError:
  Perfect = False
  Exit Function
End Function
```

لا ننسى أن الوقت يجب أن يأخذ شكلاً عاماً واحداً ، بحيث ندخله بشكل خانتين لكل من الساعة والدقيقة والثانية ، ورمز خاص للدلالة على الصباح (ص) ، والمساء (م) ، مثلاً :

{{ 10:05:52 ص }} ، وفي حال تم الإدخال غير ذلك ، يعتبر الشكل خاطئاً ، وتقوم شيفرة التابع السابق باختبار جميع أجزاء شكل الوقت ، فإذا وجدت أياً منها خاطئ ، أسندت القيمة False إلى المتحول Perfect وأنهت التابع ، وسنقوم من جسم البرنامج باختبار قيمة المتحول Perfect .

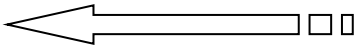
لاحظ أن محتوى اللافتة TimeError سينفذ إذا أدخلت محارف بدلاً عن الأرقام في الوقت .

سنقوم الآن بإضافة الشيفرات اللازمة إلى البرنامج ، لاختبار شكلي التاريخ والوقت .

ادخل إلى ترميز نموذج إضافة المواعيد (FrmAppend) ، وادخل منه إلى ترميز حدث النقر على زر الإضافة CmdAppend وأضف إليه الشيفرات التالية في بدايته :

```
Dim PerfectTime As Boolean
Dim PerfectDate As Boolean
PerfectDate = True
PerfectTime = True
Call TestDateIfPerfect (TxtDate.Text, PerfectDate)
If PerfectDate = False Then
  MsgBox "نمط التاريخ الذي أدخلته خاطئ ...لا يمكن الاضافة", vbOKOnly + 524288 + 1048576, "انتبه"
  Exit Sub
End If
Call TestTimeIfPerfect (TxtTime.Text, PerfectTime)
If PerfectTime = False Then
  MsgBox "نمط الوقت الذي أدخلته خاطئ ...لا يمكن الاضافة", vbOKOnly + 524288 + 1048576, "انتبه"
  Exit Sub
End If
```

تلاحظ أن الشيفرة السابقة ترسل أولاً قيمة التاريخ الذي أدخله المستخدم إلى تابع اختبار التاريخ ، فإذا كان خاطئاً ، تظهر رسالة بذلك ويتم الخروج من الإجراء ، وبالتالي لا تتم عملية الإدخال



ثم تقوم الشيفرة بإرسال قيمة الوقت الذي أدخله المستخدم إلى تابع اختبار الوقت ، فإذا كان خاطئاً ، تظهر رسالة بذلك ويتم الخروج من الإجراء ، وبالتالي لا تتم عملية الإدخال .

وفي حال كان التاريخ والوقت صحيحين ، فلا تنفذ عبارات IF السابقة وينتقل التنفيذ إلى الشيفرات التالية لها ، والتي – كما مر معنا – تعمل على إضافة السجل إلى الملف .

ادخل إلى ترميز نموذج إدخال تاريخ (FrmInputDate) ، وادخل منه إلى ترميز حدث النقر على زر موافق CmdOk وأضف إليه الشيفرات التالية في بدايته :

```
Dim IsPerfect As Boolean
IsPerfect = True
Call TestDateIfPerfect (TxtDate.Text, IsPerfect)
If IsPerfect = False Then
    MsgBox "نمط التاريخ الذي أدخلته خاطئ...", vbOKOnly + 524288 + 1048576, "انتبه"
Exit Sub
End If
```

بقي فقط كتابة شيفرات القوائم ، وهي عملية سهلة إذ أنه لكل قائمة في شريط القوائم ، يوجد زر أمر مشابه لها ، وبالتالي يمكن كتابة اسم هذا الزر في ترميز القائمة ، فمثلاً :

في حدث النقر على قائمة // **إضافة مواعيد** // ، أكتب الشيفرة التي تدل على حدث النقر على زر أمر // **إضافة مواعيد** // ، **أي** اكتب الشيفرة التالية في حدث النقر على قائمة // **إضافة مواعيد** // (**MnuAppend**) :

```
CmdAppend_Click
```

وهكذا الأمر بالنسبة لباقي أوامر القوائم .....

البرنامج جاهزاً للعمل ، ولا يشكو من شيء ، إلا إذا اقترحت أنت بعض التعديلات عليه . . . . احتفظ به ، واصنع منه نسخة تنفيذية في دليل العمل الحالي . . .

C:\Vb6\MyExercise\Exer16\Agenda.Exe

إعداد وتجهيز البرامج

Packing & Setup The Programs

ذكرنا في البرنامج الأول من هذا المرجع أن صنع ملف تنفيذي للبرنامج ، يمكّن أي مستخدم آخر من تشغيل البرنامج وعلى حاسب آخر دون ضرورة وجود نسخة فيجول بيسيك على جهازه ، أي أن البرنامج يمكنه العمل على أي حاسب دون وجود نسخة فيجول بيسيك عليه ، ولكن على الرغم من ذلك فإن الملف التنفيذي لا يعمل إلا إذا توفرت له ملفات أخرى ، كملفات مكاتب الربط الديناميكية DLL وملفات التحكم ActiveX وهي الملفات التي تحمل اللاحقة OCX والتي أضفنا بعضها أثناء صنعنا للبرامج .... لذلك إذا أردت لبرنامجك أن يعمل على حاسب آخر لا يحوي نسخة فيجول بيسيك ، يجب عليك أن ترفق هذه الملفات مع الملف التنفيذي ، ولكنك قد لا تعلم ما هي الملفات اللازمة فعلاً للتشغيل ، ويصعب على المبرمج أحياناً التكهن بملفات الربط الديناميكية DLL اللازمة لعمل البرنامج ..... لذلك أرفقت شركة مايكروسوفت عملية إعداد للبرامج ، تقوم بضغط كافة الملفات اللازمة لتشغيل البرنامج ووضعها في ملف واحد يحمل اللاحقة CAB (اختصاراً لكلمة Cabinet والتي تعني " **خزانة**") ، وتقوم هذه العملية أيضاً بصنع ملف إعداد تنفيذي للبرنامج باسم Setup.Exe .... ولفهم عملية الإعداد بشكل أكبر ، سنقوم بإعداد برنامجنا الأخير (مفكرة المواعيد – Agenda) وتجهيزه للعمل على حاسب آخر لا يحوي نسخة فيجول بيسيك .

تجهيز برنامج المفكرة Agenda

انقر على زر (**ابدأ**) واختر البرامج ، ثم اختر Microsoft Visual Studio 6.0 ، ومنها اختر أدوات اللغة (Microsoft Visual Studio 6.0 Tools) ، ثم اختر عملية إعداد البرامج وحزمها Packing & Deployment Wizard

1–) سيظهر مربع حوار بعنوان **Packing And Deployment Wizard** وهو لتحديد اسم المشروع الذي سنقوم بتجهيزه ... انقر على زر Browse ، فيظهر مربع حوار الفتح ، حدد اسم المشروع المطلوب تجهيزه ، وهو في مثالنا هذا Agenda.Vbp وانقر Open .

2–) بعد أن تمت العودة إلى مربع الحوار الأساسي ، انقر على زر Package [1]

3–) تظهر رسالة مفادها : إن البرنامج Agenda.exe يتعامل مع ملفي تحكم (Ocx) ، وأن هذين الملفين قد يكونا محدَّثين ، فهل ترغب بإعادة صنع ملف تنفيذي لضمان عملهما بشكل جيد ؟ ،
ولأننا نضمن أن الملف التنفيذي يشمل على النسختين المحدثتين ، فننقر على زر (**لا**) .

[1] تلاحظ وجود زر Deploy في مربع الحوار ، وهذا الزر لتجهيز البرنامج على شبكة الانترنيت ....!

4–) يظهر مربع حوار جديد بعنوان Packing & Deployment Wizard Package Type وهو لتحديد نوع التجهيز الذي تريده .

5–) اختر من مربع الحوار السابق الخيار Standard Setup Package ثم انقر زر التالي Next .[1]

6-) يظهر مربع حوار بعنوان Packing & Deployment Wizard Package Folder ، لتحديد اسم المجلد الذي تريد وضع ملفات الإعداد فيه[2] ، ويقترح هو عليك مجلد باسم Package ضمن مجلد العمل الأساسي ، انقر زر التالي Next وإذا لم يكن هذا المجلد موجود ، تظهر رسالة تسألك فيما إذا كنت تريد إنشاءه ، انقر منها زر **نعم** .

7–) يظهر مربع حوار Packing & Deployment Wizard Included Files وهو لعرض الملفات التي سيتم ضغطها في ملف الخزانة Cab .[3] انقر زر التالي Next .

8–) يظهر مربع حوار بعنوان Packing & Deployment Wizard Cab Options وهو لتحديد خيار الضغط الذي تريده ، فإذا اخترت الخيار الأول سيتم الضغط في ملف واحد فقط ، وأختار هذا الخيار في حال أريد نسخ الملف المضغوط على أقراص ليزرية ، أما الخيار الثاني فهو لضغط الملفات في عدة ملفات Cab ، تحدد أنت فيها حجم كل ملف ، وأختار هذا الخيار في حال أريد نسخ الملف المضغوط على أقراص مرنة .

اختر Single Cab ثم انقر زر التالي Next .

9–) يظهر مربع حوار بعنوان Packing & Deployment Wizard Installation Title وهو لتحديد عنوان البرنامج ، اتركه كما هو وانقر زر التالي Next .

10–) يظهر مربع حوار بعنوان Packing & Deployment Wizard Start Menu Items وهو لتحديد اسم المجلد البرمجي الذي تريد وضعه فيه في قائمة البرامج ، ويمكنك من خلال الأزرار الأربعة تغيير اسم المجلد البرمجي أو إضافة مجموعة جديدة.... انقر زر التالي Next .

11–) يظهر مربع حوار بعنوان Packing & Deployment Wizard Install Locations وهو لتحديد وجهة نسخ الملف التنفيذي وملفات التحكم Ocx ، ويفضل تركها على نفس الخيارات المقترحة ، انقر زر التالي Next .

12–) يظهر مربع حوار بعنوان Packing & Deployment Wizard Shared Files وهو لتحديد الملفات المشتركة مع برامج أخرى ، ففي حال كانت أحد ملفات برنامجك مشتركة مع برامج أخرى ، وقام المستخدم بإزالة برنامجك ، فسيؤدي هذا إلى عدم عمل هذه البرامج الأخرى ، وفي هذه الحالة يجب عليك تحديد ما هي الملفات المشتركة ، ولأن برنامجنا لا يشترك مع برنامج آخر بأي ملف ، انقر فقط زر التالي Next دون تحديد أي ملف .

---

[1] تلاحظ وجود نوع آخر للتجهيز وهو Dependency وهذا النوع ينشئ ملفاً جديداً بلاحقة DEP يحتوي على كافة الملفات اللازمة لتشغيل البرنامج ، ولا يصنع ملفاً تنفيذياً ، لذلك قمنا باختيار النوع الأول .....

[2] إذا أردت إنشاء مجلد جديد فانقر على زر New Folder وحدد اسمه ، كما تلاحظ أيضاً وجود زر ...Network وهذا الزر لتحديد اسم مجلد على شبكة اتصال كالانترنيت ....

[3] تلاحظ وجود زر Add في مربع الحوار المذكور ، وهذا الزر لإضافة ملفات أخرى وضغطها ، ونستخدم هذا الزر إذا كان البرنامج يشمل أيضاً على ملفات أخرى كالملفات النصية أو الرسومية ، وكون برنامجنا لا يتعامل مع هكذا ملفات ، فلم نستخدمه ....

13-) يظهر مربع حوار بعنوان Packing & Deployment Wizard Finished! وفيه يمكنك تخزين كل الخيارات التي اخترتها من مربعات الحوار السابقة ، وحفظها باسم في خانة Script Name ، وتفيدك هذه العملية في حال إعدادك لنفس البرنامج مرة أخرى ، انقر زر Finish فتبدأ عملية الإعداد ، وتضغط كافة الملفات اللازمة لتشغيل البرنامج في ملف واحد باسم Agenda.Cab ، وعندما ينتهي يظهر تقرير يخبرك فيه عن المجلد الذي تم الحفظ فيه والأعمال التي قام بها ، ويمكنك حفظ هذا التقرير بنقر زر Save Report .

انقر على زر Close فتعود إلى مربع الحوار الأول ، انقر منه أيضاً زر Close ، وتكون بذلك قد أنهيت عملية الإعداد والتجهيز لبرنامجك ، وأصبح جاهزاً للتسويق .

لترى نتيجة عملك : أدخل إلى المجلد Exer16 فتلاحظ وجود مجلد جديد فيه هو Package وهو الذي يحوي على الملفات اللازمة لإعداد البرنامج بشكل صحيح وهي ثلاث ملفات :

- Agenda.Cab : وهو الملف المضغوط ويحوي بداخله على كافة ملفات الربط الديناميكية DLL وملفات التحكم OCX والملفات التنفيذية EXE .
- Setup.Exe : وهو الملف التنفيذي الذي يتم بتشغيله إعداد البرنامج على الجهاز بشكل صحيح .
- Setup.lst : ويحوي معلومات عن وجهة كل ملف على الجهاز ، أثناء نسخة للملفات إلى الحاسب .

وعلى جميع الأحوال فنحن لا نتعامل أثناء الإعداد إلا مع الملف Setup.Exe

تلاحظ وجود مجلد آخر هو Support وهذا المجلد يحوي على نسخة احتياطية للملفات التي قام الحاسب بضغطها ، لذلك إذا كنت متأكد أن عملية التجهيز قد تمت بشكل صحيح ، يمكنك حذف المجلد Support نهائياً .

لإعداد البرنامج على جهاز آخر

انسخ الملفات الثلاثة المذكورة إلى الجهاز الآخر ، وشغّل فقط الملف التنفيذي Setup.Exe ، فيتم نسخ البرنامج إلى الجهاز الجديد ، وأيضاً يتم إنشاء اختصار له في قائمة البرامج ، واسم خاص في ((**إضافة / إزالة البرامج**)) التي تصل إليها عن طريق لوحة التحكم ، والتي يستخدمها المستخدم إذا أراد إزالة البرنامج من جهازه نهائياً .

Attached ملـــحق

سنورد فيما يلي بعض الجداول لأهم الخصائص .....

جدول 1 :

الخصائص التي تظهر في نافذة الخصائص (**Properties Window**) :

الخاصة (**Property**)	الوصف (**Description**)
Name	**تحديد اسم الكائن**
Alignment	**نوعية محاذاة الكتابة (يمين – يسار – توسيط)**
Appearance	**تحديد مظهر الكائن (ثلاثي الأبعاد – عادي) ، جربها مع مربع نص**
Back Color	**لون الخلفية**
Border Width	**عرض خط الرسم ، جربها مع كائن الـ Shape**
Border Style	**نمط الكائن ، جربها مع النموذج (قابل للتغيير – أحادي ...)**
Caption	**تحديد عنوان الكائن ، أو الاسم الأمامي**
Control Box	**التحكم بإظهار أو إخفاء صندوق التحكم**
Cancel	**جعل زر الأمر يستجيب للضغط على مفتاح الهروب Cancel**
Draw Mode	**تحديد شكل خط الرسم ، جربها مع كائن الـ Shape**
Default	**جعل زر الأمر يستجيب للضغط على مفتاح الإدخال Enter**
Down Picture	**إظهار صورة عند استمرار النقر على الكائن ، جربها مع زر الأمر**
Drag Icon	**اختيار شكل للأيقونة عند سحب الكائن**
Data Source	**اسم كائن الـ Data المصدر لحقول الخلية، جربه مع صندوق نصوص أو تسمية (Label) ، شرط أن يكون لديك كائن Data مرتبط مع قاعدة معطيات**
Data Field	**تحديد الحقل الذي سيأخذ الكائن قيمته منه ، جربه مع صندوق نصوص أو تسمية (Label)**
Data Base Name	**اسم قاعدة البيانات التي سيرتبط بها كائن الـ Data**
Enable	**جعل الكائن متاح للاستعمال أو لا ، أثناء تنفيذ البرنامج**
Font	**تحديد مظهر الخط الأمامي للكائن**
Fore Color	**لون الخط للكائن**
Height	**ارتفاع الكائن**
Index	**رقم الدلالة للكائن ، وذلك في حال وجود عدة كائنات بنفس الاسم**
Left	**بعد الكائن عن حد النموذج الأيسر**

الخاصة (Property)	الوصف (Description)
Max Button	إظهار أو إخفاء زر التكبير والتصغير للنموذج
MDI Form	جعل النموذج نموذجاً ابناً أو لا ، ويستعمل في حالة النماذج الأبناء
Min Button	إظهار أو إخفاء زر التصغير للحد الأدنى للنموذج
Mouse Pointer	تحديد شكل لمؤشر الماوس عند وضعه فوق الكائن
Movable	جعل النموذج قابل للتحريك أو لا
Negotiate Menus	السماح بإظهار القائمة المنبثقة للنموذج أو لا
Picture	تحديد صورة تظهر كخلفية للكائن
Right To Left	السماح باتجاه الكتابة من اليمين إلى اليسار
Scale Height	تحديد ارتفاع النموذج
Scale Width	تحديد عرض النموذج
Scale Mode	تحديد نوع القياس للأطوال في النموذج (Pixel – Point - ...)
Style	تحديد نمط الكائن ، جربه مع زر الأمر لتحويله إلى رسومي ....
Show In Task Bar	إظهار اسم النموذج على شريط المهام
Start Up Position	موضع النموذج على الشاشة عند بدء تنفيذ البرنامج (انظر الجدول 2)
Text	محتوى صندوق النصوص
Top	ابتعاد الكائن عن حد النموذج الأعلى
Tool Tip Text	كتابة عبارة تظهر عند وضع الماوس على الكائن لثانية واحدة دون نقر
Visible	جعل الكائن مرئياً أو مخفياً
What This Button	إظهار زر إشارة الاستفهام في حال كان النموذج كمربع حوار
Window State	حالة النموذج عند بدء تنفيذ البرنامج (كبير – عادي – حد أدنى)

جدول 2

من نافذة توضع النموذج (Form Layout Window) يمكننا تحديد أين سيظهر النموذج عند بدء تشغيل البرنامج ، بالسحب يدوياً ، أو بإظهار القائمة المنبثقة له ، فتظهر أربع خيارات له ، موضحة بالجدول التالي :

الخيار	الوصف
0-Manual	تحديد بشكل يدوي من النافذة ، أو بتغيير خاصتي (Top – Left) للنموذج
1-Center Owner	منتصف المساحة المخصصة للبرنامج ، وذلك في حال كان المشروع يحوي اكثر من النموذج
2-Center Screen	منتصف الشاشة
3-Windows Default	الموضع الافتراضي الذي يقترحه عليك فيجول بيسيك ، وهو يكون عادة في أعلى يسار الشاشة

جدول 3

ذكرنا أن تابع MsgBox يظهر رسالة وأزرار أمر بعد أن تحدد له أنت ما هي الأزرار التي سيظهرها ، فمثلاً :

MsgBox "هل تريد الخروج",VbYesNo

تظهر رسالة مع زري أمر (نعم – لا) ، ولكن يمكن الاستعاضة عن كلمة VbYesNo برقم / **4** / ، أي يمكنك كتابة الشيفرة السابقة بطريقة أخرى هي :

MsgBox "هل تريد الخروج",4

ولكل نوع من الأشكال التي يظهرها هذا التابع ، يوجد رقم خاص به ، والجدول التالي يوضح كافة القيم والأرقام الممكنة :

الرقم	العبارة المكافئة	الوصف
0	VbOkOnly	يعرض زر موافق فقط
1	VbOkCancel	يعرض زري موافق و إلغاء الأمر
2	VbAbortRetryIgnore	يعرض أزرار (إحباط – إعادة المحاولة – تجاهل)
3	VbYesNoCancel	يعرض أزرار (نعم – لا – إلغاء الأمر)
4	VbYesNo	يعرض زري (نعم – لا)
5	VbRetryCancel	يعرض زري (إعادة المحاولة – إلغاء الأمر)
32	VbQuestion	يعرض إشارة السؤال
64	VbInformation	يعرض إشارة المعلومات
524288	VbMsgBoxRight	يجعل عنوان صندوق الرسالة على اليمين
1048576	VbMsgBoxRtlReading	يجعل صندوق الرسالة متوافق مع العربية

جدول 4

عندما تظهر رسالة التابع MsgBox ، فإن التابع يعيد قيمة بحسب الزر الذي نقره المستخدم ، ويمكن إسناد هذه القيمة إلى متحول ما لمعرفة ما هو الزر الذي قام المستخدم بنقره ، والجدول التالي يبين هذه القيم والأزرار المقابلة لها :

القيمة	الزر الذي نقره المستخدم
1	موافق
2	إلغاء الأمر
3	إحباط
4	إعادة المحاولة
5	تجاهل
6	نعم
7	لا

جدول 5

ذكرنا في برنامج الملفات الصوتية (البرنامج الثاني عشر) ، أن كائن الـ TegoMM الخاص يفتح الملفات الصوتية أو ملفات الفيديو ..... تأخذ خاصية الـ Mode فيه قيماً مختلفة ، وذلك حسب الحالة التي تكون فيها (قراءة – إيقاف – تسجيل .....) ، والجدول التالي يبين هذه القيم والحالات الموافقة لها :

القيمة	الحالة
524	يكون الملف غير مفتوح بعد
525	يكون الملف في حالة إيقاف Stop
526	يكون الملف في حالة التشغيل
527	يكون الملف في حالة التسجيل
528	يكون الملف في حالة تخطي إلى موقع جديد
529	يكون الملف في حالة إيقاف مؤقت Pause
530	يكون الملف جاهزاً للتشغيل