MicroSoft

VISUAL BASIC 6.0

فيجوال بيسك 6.0

اعداد المهندس

احمد ابراهيم احمد

بسم الله الرحمن الرحيم

﴿وما اوتيتم من العلم إلا قليلا﴾

المقدمة

الحمد لله رب العالمين والصلاة والسلام على سيدنا محمد وعلى اله وصحبه اجمعين.

من المعلوم لدى جميع مستخدمي الكمبيوتر. انه بعد ان تم اصدار نظام الويندوز وانتهاء عصر الـ MSDOS فان الحاسبات اخذت بالانتشار بصورة اسرع واصبحت شيئا ضروريا في الحياة العصرية لدى معظم الناس، وذلك لانها اصبحت اكثر سهولة لانها اعتمدت الاسلوب الصوري أي اصبح نظام التشغيل على شكل واجهات صورية . حيث اتاحت للمستخدم ان يقوم بعدة عمليات بمجرد النقر على ازرار و الاختيار من القوائم. كذلك يمكن ان تلاحظ الان ان هناك كماً هائلا من البرامج المنتشرة والمتوافقة مع انظمة التشغيل حيث يمكنك ان تجد معظم ما قد يخطر ببالك من البرامج وتحقق غايتك. لكنك في بعض الاحيان تجد بعض البرامج لاتتوفر فيها جميع احتاجاتك . فاذا كنت ممن يهتمون بكيفية صنع هذه البرامج وتصميمها وبرمجتها أي انك تريد ترك كونك مستخدم عادي للبرامج وان تصبح مبرمج ومصنع لمثل هذه البرامج. عليك بتعلم احدى لغات البرمجة الرئيسية كـ Visual C او Visual Basic او FoxPro الخ. تتيح لك هذه اللغات بان تصمم برامجك بنفسك وكذلك ان تصيرها وتسخرها لاستخداماتك الخاصة ، فيمكنك ان تقوم بعمل برامج تساعدك في عملك ، أي يمكن ان تصنع برنامجك الخاص بنفسك، في هذا الكتاب سنقوم بتعلم لغة Visual Basic وهي اللغة الاسهل والاكثر انتشارا ، حيث ساعدت بساطتها في انتشارها بشكل سريع. فكلمة Visual تعني مرئي اي لغة بيسك المرئية . اما كلمة

بيـــــــــــسك فهـــــــــــي اختـــــــــــصار لجملـــــــــــة>>

Beginner’s All – Purposes Symbolic Instruction Code

والتي تعني شفرة التعليمات الرمزية لجميع الأغراض للمبتدئين. وتعد لغة فيجول بيسك من اسهل لغات البرمجة و أكثرها شيوعا. أصل لغة فجوال بيسيك هي لغة بيسيك التي ظهرت في كلية دارتماوث Dartmaouth علي يد جون كيمني John Kemeny و توماس كيرتز Thomas Kurtzعام 1963م و قد أصبحت بسرعة فائقة من أشهر و أسهل لغات البرمجة بل لم تقف عند هذا فحسب فأصبحت كلغة تعليمية تدرٌس في المعاهد و الكليات و الجامعات و بدأ انتشارها في السبعينات. ثم أخذت هذه اللغة في التطور الدائم مع المحافظة على سهولة استخدامها و بساطتها. و كان الإصدار الأول لهذه اللغة عام 1991 م ثم كان الإصدار الثاني منها عام 1992 م. ثم كان الإصـــدار الثالـــث عـــام 1993 ومـــن ذلـــك الوقـــت تعاقبـــت إصـــدارات كثيـــرة منهـــا .

و فيجول بيسك تستخدم مفهوم البرمجة الشيئية Object Oriented Programming Languageوتختصر إلى OOP وفكرة البرمجة الشيئية هي تقسيم المشروع إلى كائنات Objectكل منها له خصائص Properties يحدد شكله وسلوكه وله أحداث تجري عليه Eventsوردود أفعال تؤثر فيه وفي الكائنات التي من حوله تسمي إجراءات Procedure وتفعل كل هذا بالكود Code والكود يحتوي علي أوامر والأوامر هي التي تؤثر علي الكائنات في خصائصها وسلوكها أو خصائص وسلوك من حولها من الكائنات . ألان اعتقد انك عرفت أننا لا يمكننا حصر الأوامر ومعرفة عددها. أما الدوال عبارة عن برامج صغيرة وبالطبع هذه البرامج تحتـــــــــــوي علـــــــــــي أوامـــــــــــر..

أما عن حدودها فلكل شي له حدود والكمال لله وحدة سبحانه تعالي ولكني باعتقادي هو أن حدود اللغة تتوقف علي حدود عقلية المبرمج..

انطلاقا من قول الحبيب محمد (ﷺ) (ان الله يحب اذا عمل احدكم عملاً ان يتقنه).

سياخذك هذا الكتاب الى عالم البرمجة حيث سيبدا معك من الصفر ويعلمك اساسيات البرمجة حيث ستتمرن مع امثلة ودروس عملية خطوة بخطوة ويحاول ان يشرح لك كل خطوة قمت بها ، ويوصلك الى مستوى متوسط في البرمجة. تكون فيه قادرا على بناء برامج خاصة بك .

ومن الله التوفيق.

تعرف على فيجوال بيسك

- **مفاهيم عن البرمجة**

ان جميع لغات البرمجة تحتوي على أوامر وايعازات خاصة بها،يستخدمها المبرمجون في بناء برامجهم. فالمبرمج يقوم بتوظيف هذه الأوامر وتسخيرها والاستفادة منها بأكبر قدر ممكن للوصول إلى المطلوب، حيث تعمل هذه الأوامر والايعازات مجتمعة مع بعضها في تكوين البرنامج، ولكل لغة من لغات البرمجة قدرات خاصة ومزايا معينة تميزها عن غيرها من اللغات، حيث تتفاوت هذه اللغات من ناحية الصعوبة والإمكانيات. اما البرمجة فهي عبارة عن سلسلة من الخطوات تنفذ بشكل تلقائي عند تنفيذ البرنامج ، ان اللغة البرمجية تعطي المبرمج إمكانية تطبيق أفكاره و تحويلها إلى شئ مفيد (أو ضار كالفيروسات) ورؤية نتائجها عمليا.

- **تشغيل برنامج الفيجوال بيسك والتعرف على الواجهة**

ستتعلم في هذا الدرس:

- ✓ تشغيل فيجوال بيسك.
- ✓ التعرف على نوافذ بيئة التطوير المتكاملة.
- ✓ التعرف على قوائم بيئة التطوير المتكاملة.
- ✓ أشرطة الأدوات.

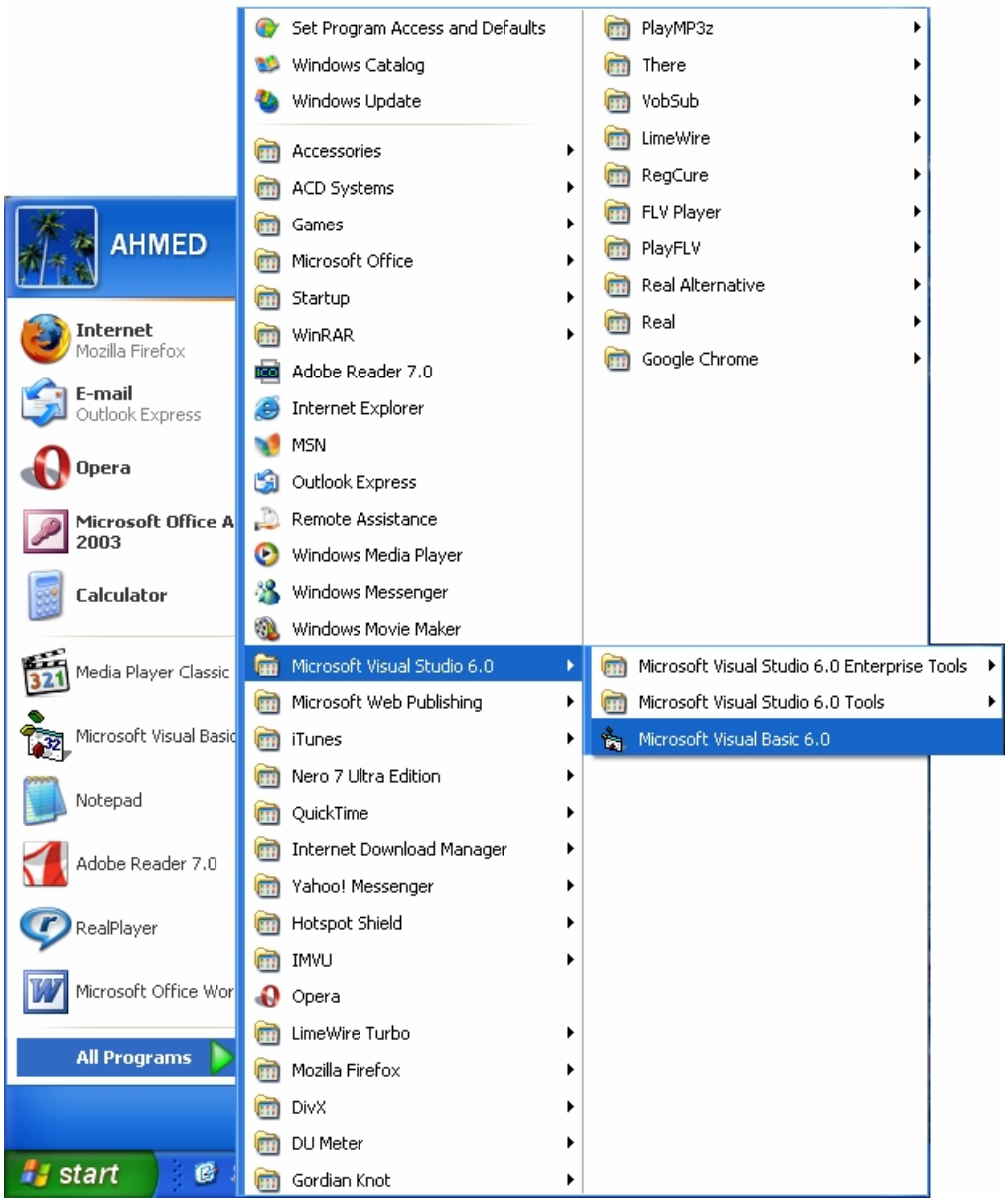
تشغيل فيجوال بيسك:

كأي برنامج أخر فان لغة فيجوال بيسك لكي يتم تشغيلها يجب ان تكون مثبتة في الحاسبة، فبعد تنصيب اللغة على الحاسبة ولتشغيلها يجب الذهاب إلى قائمة :

Start → All Programs → Microsoft Visual Studio 6.0 → Microsoft Visual Basic 6.0

حيث نقوم بالضغط على إيقونة

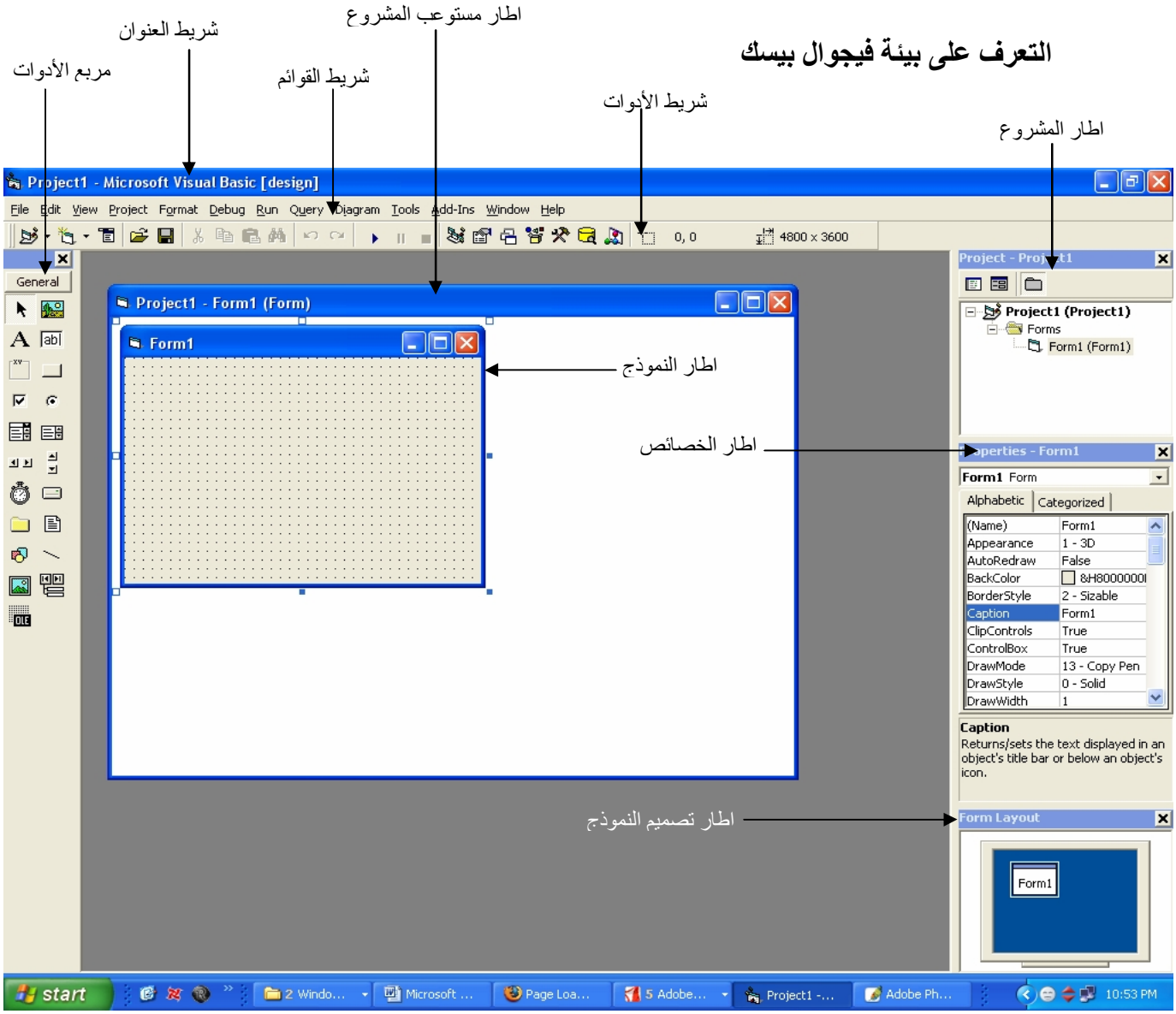
Microsoft Visual Studio 6.0 كما مبين في الشكل (1-1)



الشكل (1-1)

أو عن طريق جعل لها اختصار على سطح المكتب Desktop.

التعرف على بيئة فيجوال بيسك



تحتوي بيئة فيجوال بيسك على أدوات برمجة لمساعدتك على بناء برامج فيجوال بيسك. يزود **شريط القوائم** وصولا إلى معظم الأوامر التي تتحكم ببيئة برمجة فيجوال بيسك. تعمل القوائم والأوامر في فيجوال بيسك مثلما تفعل في كل البرامج المتوافقة مع ويندوز ، وباستطاعتك الوصول إليها بواسطة الماس أو لوحة المفاتيح. تحت شريط القوائم يوجد شريط الأدوات ، وهو عبارة عن مجموعة من الأزرار التي تشكل اختصارات لتنفيذ الأوامر والتحكم ببيئة برمجه فيجوال بيسك . يجب أن تكون معتادا على شريط الأدوات أذا استعملت مايكروسوفت وورد أو مايكروسوفت اكسل من قبل. لتنشيط أحد أزرار شريط الأدوات ، أنقره بواسطة الماوس.

بالإضافة إلى ذلك يوجد **مربع الأدوات** و **أطار مستوعب المشروع** و **أطار النموذج** و **أطار المشروع** و **أطار الخصائص** و **أطار تصميم النموذج**. سنتعلم كيفية استعمال هذه الميزات لتخصيص بيئة البرمجة بهذا الشكل لاحقا ولا يتم الاعتياد على هذه الأدوات الا من خلال استعمالنا لها.

التعرف على نوافذ بيئة فيجوال بيسك

نافذة مشروع جديد New Project

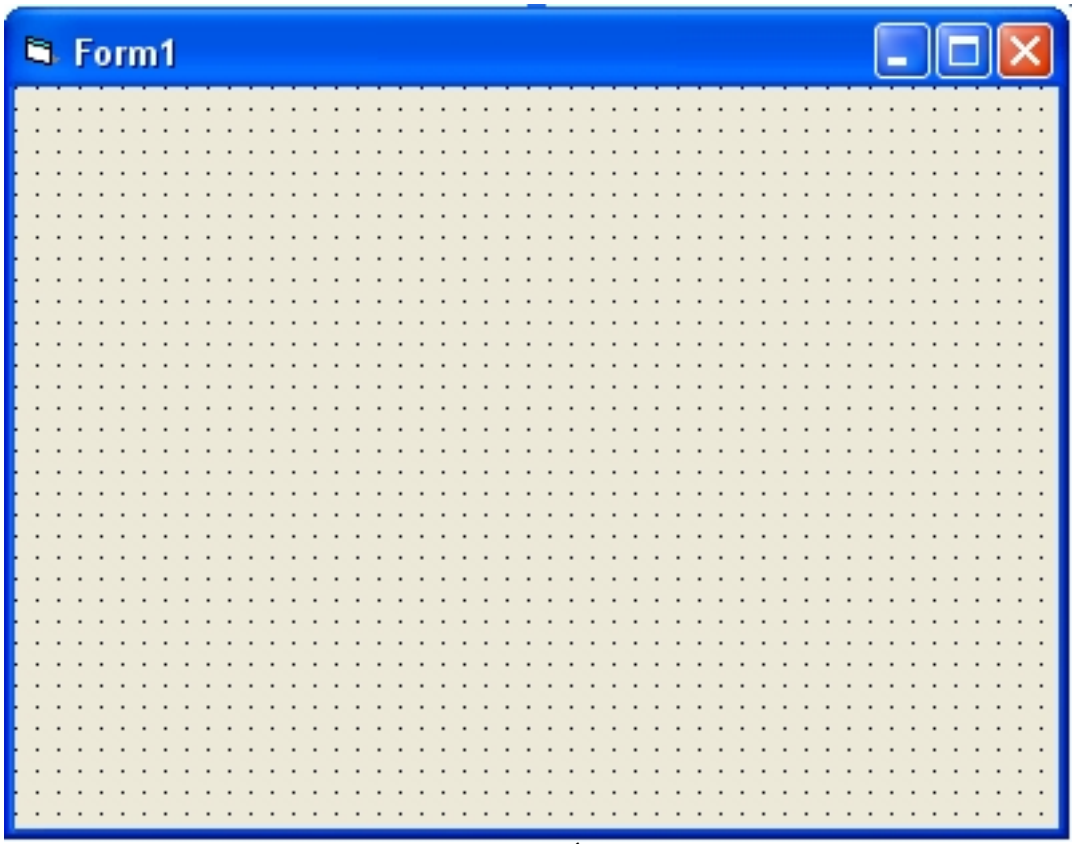
عندما تقوم بتشغيل منصة العمل Visual Basic لأول مرة، فان لهذه النافذة احتمال مؤكد للظهور .فعن طريقها تستطيع الاختيار بين أنواع عدة من المشاريع كـ ActiveX DLL، ActiveX EXE، Standard EXE .... الخ .حدد النوع Standard EXE ثم انقر على الزر Open أو اضغط على المفتاح .[ENTER] فالقاعدة السائدة لجميع كتب Visual Basic التعريفية تبدأ الشرح دائما بالنوع Standard EXE . تستطيع إلغاء فكرة ظهور هذه النافذة عن طريق تحديد الخيار Don’t show this dialog in the future الموجود في أسفلها حتى لا تظهر كلما شغلت البرنامج. انظر الشكل (2-1)

شكل(2-1)

ملاحظة :إذا لم يرق لك هذا التغيير، واردت العودة إلى الوضع السابق، فاختر الأمر Options ... من قائمة Tools ، ومن مربع الحوار الذي يظهر امامك اختر تبويب Environment ، وانقر على الخيار Prompt for project. حيث ستعود هذه الواجهة للظهور كلما شغلت فيجوال بيسك.

نافذة مصمم النماذج Form Designer

هذه النافذة تعتبر سر نجاح Visual Basic ، وهي أشهر نوا فذ Visual Basic منذ الإصدار الأول، مع العنوان الابتدائي لها Form1 عن طريق هذه النافذة تقوم بعملية تصميم **واجهة برنامجك** اما بتعديل خصائصها أو وضع باقة أدوات عليها باستخدام الفأرة Mouse مبينة لك قوة ونجاح فكرة لغات البرمجة المرئية Visual Programming Languages . طريقة وضع الأدوات على نافذة النموذج أشبه ما تكون بعملية رسم مربعات كما في برنامج الرسام Paint . المزيد أيضا، عمليات التحرير كالنسخ واللصق والقص مدعومة على جميع الأدوات التي تضعها. وتسمى أيضا واجهة البرنامج انظر الشكل(3-1)



الشكل(3-1) نافذة مصمم النماذج أو تسمى " نافذة النموذج " للاختصار

نافذة صندوق الأدوات Toolbox "مربع الأدوات"

بعد ان تظهر لنا نافذة النموذج السابقة، فان استحقاق وضع الادوات عليها قد حان .

والأدوات موجودة في نافذة صندوق الأدوات Toolbox شكل (4 -1) حيث تسمى **أدوات التحكم**، التي تعرض لك 20 أداة قياسية مدمجة في جميع برامجك المصممة تحت ، Visual Basic وقد تحتوي على مجموعة أدوات إضافية تدعى ActiveX Controls. يمكنك استعمال

هذه الأدوات لإضافة أعمال فنية وتسميات وأزرار ومربعات سرد وأشرطة تمرير وقوائم وأشكال هندسية إلى واجهة البرنامج، كل أداة تحكم تضيفها إلى النموذج تصبح **كائنا** أو عنصر واجهة قابل للبرمجة في برنامجك. ستكون هذه العناصر مرئية لمستخدم برنامجك عندما يكون البرنامج مشتغلا. مع ان الأدوات عددها 20 أداة إلا انه يوجد 21 زر Buttons ،هذا الزر الإضافي موجود في الركن العلوي الأيسر من الأدوات على شكل مؤشر Pointer وظيفته الأساسية إلغاء عملية طلب رسم أداة . لا تشغل نفسك به كثيرا، فهو يضغط نفسه تلقائيا بمجرد انتهائك من عملية رسم أو وضع الأداة على نافذة النموذج. يحتوي مربع الأدوات أيضا على أدوات تحكم يمكنك استعمالها لإنشاء كائنات تؤدي عمليات خاصة "في الكواليس" في برنامج فيجوال بيسك، تنفذ هذه الكائنات عملا مفيدا لكن المستخدم لايمكن رؤيتها عندما يكون البرنامج مشتغلا، إنها تتضمن كائنات لمعالجة المعلومات في قواعد البيانات، وتعقب مرور الوقت في برنامجك.

لفتح مربع الأدوات انقر زر مربع الأدوات في شريط الأدوات حيث يظهر مربع الأدوات عادة في الجهة اليسرى من الشاشة.

زر مربع الادوات

شكل (4-1) صندوق الأدوات "مربع الأدوات"

شرح مختصر لمربع الأدوات

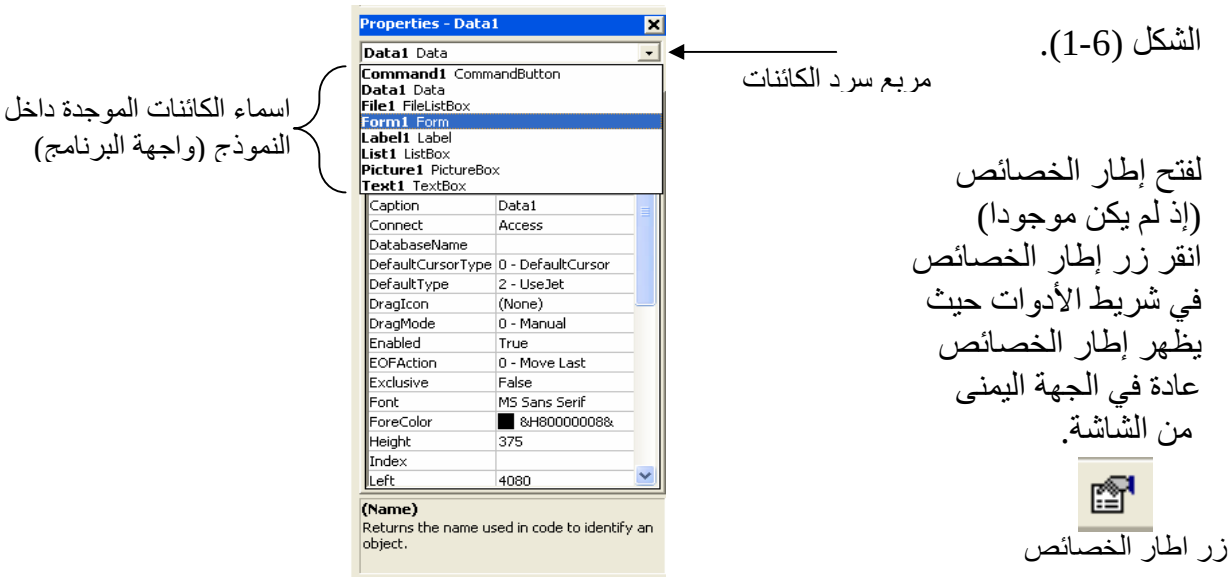
الأداة	اسم الأداة	الوصف
	Pointer مؤشر	وهي تعيد مؤشر الماوس إلى السهم الطبيعي اذا كان المؤشر على إحدى الأدوات أو الأشكال الأخرى غير الوجه الطبيعي.
	Picture Box صندوق صورة	وهي أداة تتيح لك إضافة صورة إلى البرنامج وإدماج الصورة في البرنامج.
	Label تسمية	وهي أداة تتيح لك إضافة عنوان ما إلى البرنامج ، أو نص في أي مكان من النموذج ، وقد تستخدم الـ Label في منح المستخدم بعض المساعدة للدلالة إلى شئ ما . وتعرض نص ثابت على الشاشة أثناء التشغيل لاتستطيع تغيره.
	TextBox مربع نص	وهي أداة تتيح لك أن تترك للمستخدم فرصة إدخال بيانات مثل أسمه مثلا أو مسار البرنامج أو شئ آخر.
	Frame الأطر	وهي أداة تستخدم لصنع إطار وتضمين بعض الأدوات بداخله
	CommandButton زر أمر	وهي أداة زر أمر حيث تتيح لك أن تجعل للمستخدم زر ليضغط عليه عند تنفيذ أمر معين.
	CheckBox مربع اختيار	وهي أداة مربع اختيار ، لتتيح للمستخدم اختيار أكثر من خيار.
	OptionButton زر اختيار	وهي أداة زر اختيار "راديو" ، لتتيح للمستخدم من اختيار خيار واحد من عدة خيارات .
	ComboList القائمة المنسدلة	وهي أداة القائمة المنسدلة حيث تنسدل منها قائمة ليختار منها المستخدم أحدى الخيارات المتاحة أو القيم المتاحة.
	List قائمة	وهي قائمة تشبه إلى حد ما الـ ComboBox ولكن تختلف عنها بانها ليست منسدلة
	HScrollBar شريط الانزلاق العرضي	وهي أداة شريط الانزلاق العرضي ، وتستخدم في انزلاق الصور والكائنات الأكبر من اللازم عرضيا.
	VScrollBar شريط الانزلاق الطولي	وهي أداة تشبه السابقة ولكن شريط انزلاق طولي وليس عرضي.
	Timer المؤقت	وهي أداة المؤقت وظيفتها أنها تقوم بأداء عمل معين أو عدة أعمال معينة بصفة دورية كلما مر زمن معين تحدده أنت، أي تكرار الأمر
	DriveListBox مربع سرد محركات الأقراص	وهي عبارة عن قائمة تسرد فيها أسماء محركات الأقراص للقرص الصلب HardDisk وكذلك السيدي روم CD-ROM .
	DirListBox مربع سرد الدلائل	وهي عبارة عن قائمة تسرد فيها أسماء الدلائل (Folders) داخل محركات الأقراص للقرص الصلب HardDisk وكذلك CD-ROM .
	FileListBox مربع سرد الملفات	وهي عبارة عن قائمة تسرد فيها أسماء الملفات (Files) الموجودة داخل الدلائل (Folders) أو داخل محركات الأقراص للقرص الصلب HardDisk وكذلك السيدي روم CD-ROM . حيث انك تحدد المسار أيضا
	Shape شكل هندسي	وهي أداة تستخدم لرسم أشكال هندسية كالدائرة والمثلث والمربع ... الخ
	Line خط	أداة تستخدم لرسم خط في واجهة البرنامج
	Image صورة	هي أداة لإضافة صورة إلى النموذج وتختلف عن PictureBox بعض الشيء في الخصائص

	Data بيانات	أداة تستخدم في ربط البرنامج بقاعدة بيانات خارجية.
	OLE Object Link Embed	هي مختصر لـ (تضمين وربط الكائنات) وهي أداة تستخدم لربط وتضمين برامج خارجية متوافقة مع ويندوز لتعمل في برنامجك.

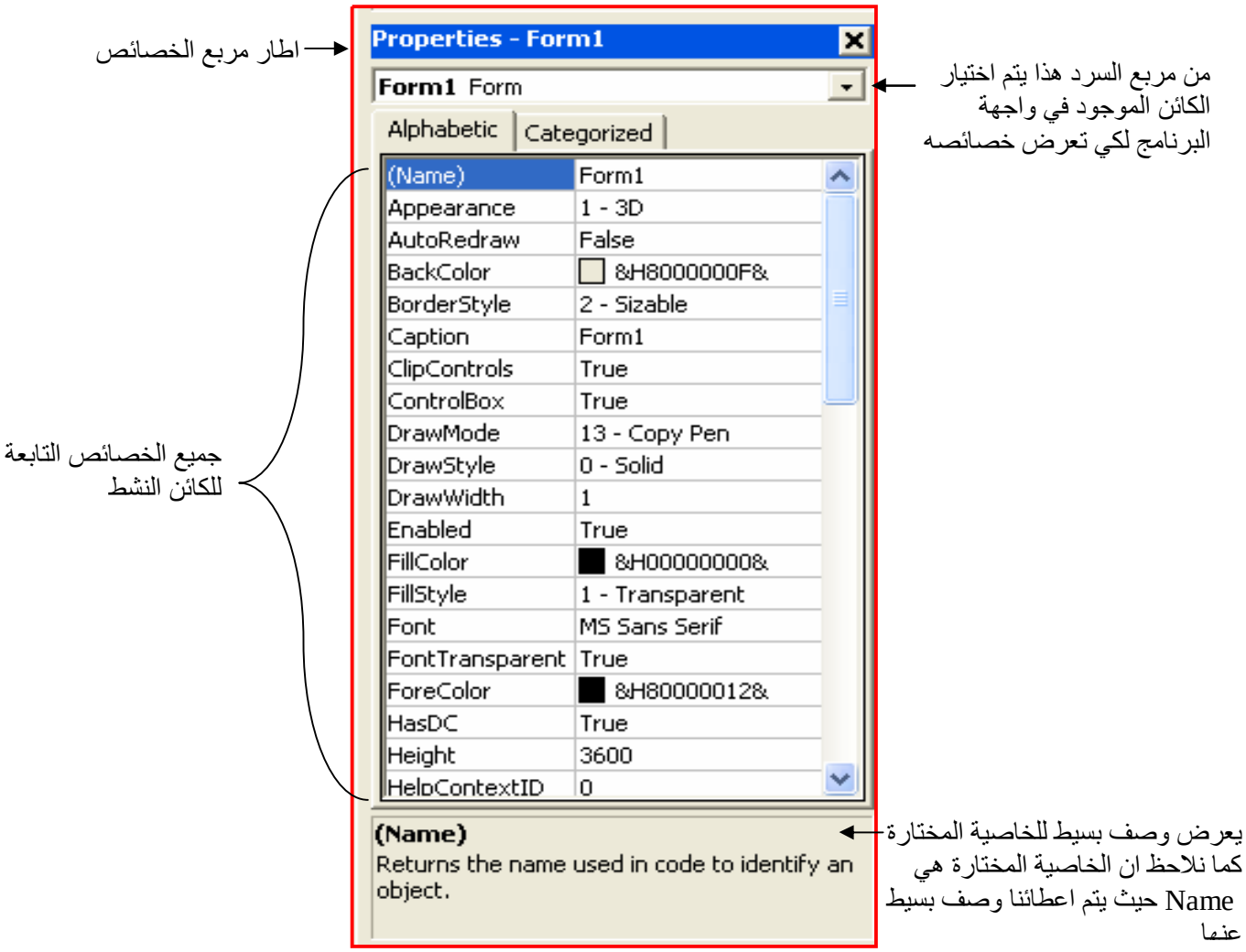
إطار (نافذة) الخصائص:

يتيح لك إطار الخصائص تغيير مميزات، أو **إعدادات الخاصية** ، لعناصر واجهة البرنامج الموجودة في النموذج. إعداد خاصية هو ميزة احد الكائنات في واجهة البرنامج، مثلا يمكنك تغير خط أو لون الكتابة داخل مربع النص وكذلك الحجم وذلك من خلال خاصية الخط التابعة لمربع النص "هذه إحدى خصائصه" فكل كائن يملك عدة من الخواص، حيث توجد عدة خواص مشتركة بين كل الكائنات (أي كل كائن يملك خاصية حيث ان هذه الخاصية موجودة في بقية الكائنات) كالاسم مثلا أي اسم الكائن حيث كل كائن لديه اسم معين لذلك هذه الخاصية تعتبر مشتركة لان جميع الكائنات تملكها. وكذلك توجد خواص مشتركة بين مجموعة من الكائنات فقط كالخط ولونه مثلا كما في الكائنات التي تظهر نصوص أو تسميات على واجهة البرنامج.

وكذلك توجد خصائص منفردة لكل كائن أي حسب استخدامه، يمكن تغيير الخصائص عن طريق إطار الخصائص واختيار الخاصية التي نريدها وهذه الطريقة تعتبر يدوية وتكون في مرحلة التصميم، أو يمكن تغير الخصائص عن طريق إضافة شفيرة برمجية في إطار الشفيرة لتغيير خاصية أو عدة خواص عندما يكون البرنامج مشتغلا. يحتوي إطار الخصائص على مربع سرد الكائنات (ComboBox) يبين كل عناصر (كائنات) واجهة البرنامج الموجودة على النموذج انظر الشكل (5-1). كما يبين إطار الخصائص الإعدادات التي يمكن تغييرها لكل كائن. انظر الشكل (6-1).



شكل (5-1) يبين مربع سرد الكائنات داخل النموذج (واجهة البرنامج)



شكل (6-1) يبين إطار الخصائص وكذلك يظهر الخصائص التابعة للكائنات

- **التفكير في الخصائص**

في فيجوال بيسك ، كل عنصر من واجهة البرنامج بما في ذلك النموذج له مجموعة من الخصائص المعرفة، يمكن ضبط الخصائص أثناء التصميم باستعمال إطار الخصائص، كما يمكن ضبطها في الشفيرة إثناء عمل البرنامج لتؤدي عملا مفيدا. لكنك في البدء قد تجد انه من الصعب استيعاب مفهوم الخصائص ، وقد يساعدك التعبير عنها بتشبيهها بالسيارة. السيارة شيء تستعمله لانتقال من مكان إلى أخر ، وبما انها كائن مادي تشغل حيزا من الفراغ فانها تمتلك عدة خصائص متأصلة بها، كاسم الصانع واللون والموديل والمحرك، كما انها مصممة على نحو ما (سيارة صالون أو للحمل أو معدات ثقيلة أو نقل الركاب أو للاستخدام العسكري) . في فيجوال بيسك هذه الأمور هي **خصائص** الكائن الذي هو السيارة. القالب الذي أعطى هيكل السيارة يسمى عندها **أداة التحكم**. يتم تعريف معظم خصائص

السيارة إثناء صنعها، اما بعض الخصائص الأخرى (كالسرعة والعمر والخيارات كالمرايا) فقد تتغير بعد استعمال السيارة. خلال العمل مع فيجوال بيسك ستجد خصائص من النوعين.

- **شرح سريع لمعظم الخصائص**

الخاصية	الوصف
Name	هي خاصية باسم الأداة البرمجي أي خاصة في البرمجة (تدخل في كتابتة الشفيرة) لذلك يجب ان تكون اسم فريد أي لا يجوز ان يحمل كائنان نفس الاسم "البرمجي" حتى تستطيع الفيجوال بيسك من التمييز بين الأدوات من خلال أسماءها (كما ان لكل شخص رقم هوية فريد خاص به لا يملكه غيره)
Alignment	المحاذاة ويقصد به محاذاة النص داخل الكائن أي جعله يبدأ من جهة اليمين أو في الوسط أو من اليسار حيث يحوي على ثلاث خيارات هي: 0-Left Justify 1-Right Justify , 2-Center
Appearance	المظهر ويقصد بها مظهر (شكل) الكائن إثناء تشغيل البرنامج اما ان يكون مجسم أو ان يكون مسطح لذلك يحتوي على خيارين هما: 0-Flat , 1-3D
Auto size	التحجيم الذاتي ويقصد بها تحجيم الأداة لتحتوي المحتوى الذي في داخل الأداة كالنص أو الصورة مثلا، حيث سوف يتغير حجم الأداة ليصبح بحجم المحتوى فإذا كان المحتوى كبيرا سوف يكبر حجم الأداة أي الإطار الخارجي للأداة والعكس صحيح حيث تحتوي على قيمتين منطقيتين هما: True أي تفعيل التحجيم أو False عدم تفعيل التحجيم الذاتي
Back color	اللون الخلفي وهي خاصية لتغيير اللون الخلفي للأداة أو النموذج حيث عند اختيارها سوف تظهر لديك قائمة بالألوان المستعملة في بيئة فيجوال بيسك ويمكنك اختيار اللون المناسب مع ما يتناسب مع التصميم
Back style	اسلوب الخلفية: ويقصد به اسلوب عرض الخلفية اما ان تكون ملونة حسب اللون المختار في خاصية الـ Backcolor أو ان تكون شفافة أي سوف يلغى تأثير خاصية الـ Backcolor وذلك يعني انها تحتوي على خيارين هما: 0-Transparent ويقصد به الشفافية أو 1-Opaque ويقصد به غير شفاف.
Boarder style	نوع الإطار وهي خاصية لعمل إطار للكائن حيث في بعض الكائنات يحوي فقط على قيمتين هما: 0-None لا يوجد إطار خارجي أو1-Fixed Single يوجد إطار خارجي ثابت للكائن . اما النموذج (واجهة البرنامج) فانه يحتوي على خيارات اكثر ولكنه يحتوي على None أيضا في حالة اذا أردت عدم عمل إطار للواجهة الخارجية للبرنامج، يمكنك رؤية تلك الخيارات من خلال الوقوف على خاصية Boarder Style
Caption	عنوان: ويقصد به العنوان الذي سوف يظهر للمستخدم على تلك الأداة حيث يختلف عن اسم الأداة لان الاسم لايظهر للمستخدم ولكنه يستخدم في كتابة الشفيرة اما العنوان فهو ما تريد ان يقراه المستخدم على ذلك الكائن سواء أكان زر أمر أو تسمية (Label) كمثال على ذلك زر "OK" فكلمة OK هي التسمية التي بقرائها المستخدم على هذا الزر
Drag icon	إيقونة السحب ويقد بهذه الخاصية هي تحديد الإيقونة التي تريد ان يراها المستخدم عند النقر على الكائن ومحاولة جره (سحبه) حيث ستحدد الإيقونة من ملف خارجي

Drag mode	طريقة السحب : في هذه الخاصية يوجد خياران وهي تعمل مع الخاصية Drag icon حيث اما ان تجعل إيقونة السحب تظهر للمستخدم أو ان لاتظهر للمستخدم حتى ولو تم تحديد الإيقونة في الخاصية السابقة فهذه الخاصية هي التي تسمح أو لا تسمح بظهور الإيقونة والخياران هما: 0-Manual هذا لايسمح بالظهور ، و -1 Automatic حيث يسمح للإيقونة بالظهور
Data Field DataFormat DataMember DataSource	خاصية البيانات : حيث توجد عدة خصائص للبيانات حقل البيانات و صيغة البيانات وأعضاء البيانات ومصدر البيانات تستخدم هذه الخواص في حالة ربط الأداة ببيانات خارجية كربط البرنامج مع قاعدة بيانات أكسس
Enabled	التمكين: وهي خاصية تمكين أو عدم تمكين الكائن حيث يوجد بها خيارين هما True و False حيث ان True تمكن المستخدم من التعامل مع الكائن سواء بالكتابة فيه اذا كان مربع نص أو من الضغط عليه اذا كان زر أمر، اما False فهي عدم تمكين المستخدم من الكتابة أو الضغط على الزر.
Font	خط : خاصية الخط ، يمكنك تغيير نوع الخط المستخدم مع تلك الأداة حيث انه يمكنك اختيار أي خط موجود على الحاسبة كما هو الحال مع برنامج الورد أو اكسل وكذلك تستطيع تغيير حجم الخط وإضافة تأثيرات كغامق أو مائل أو تحته خط ، وبعد تغيير الخط سوف ترى انه قد غير نوع الخط المستخدم في تلك الأداة إلى الخط الجديد.
Fore color	لون الخط: هذه الخاصية تمكنك من تغيير لون الخط
Height	الارتفاع: هذه الخاصية تمكن من تحديد ارتفاع الكائن حيث تستطيع تغيرها يدويا أو برمجيا .
Index	فهرس : هو عبارة عن رقم يعطى في حال انه تم تسمية اكثر من كائن من نفس النوع بنفس الاسم البرمجي فان كل كائن يحمل نفس الاسم البرمجي سوف يحمل رقم مفهرس يختلف عن الكائنات الأخرى وهنا لامجال للتكرار، قد قلنا انه لا يمكن ان يحمل كائنان أو اكثر نفس الاسم البرمجي لكن ذلك ممكن بشرط ان يضاف الـ Index إلى اسم الكائن فيصبح اسم الكائن على شكل مصفوفة والـ Index هو سوف يكون اسم كل كائن في تلك المصفوفة فمثلا لو كان لدينا مربع نص واسمه كان txtbox فاذا أدرجنا مربع نص ثاني وثالث واسمينا كل منهما txtbox أيضا فانه يجب علينا إعطاء كل واحد رقم في الخاصية Index تختلف عن رقم الكائن الأخر ، لذلك سوف تكون أسمائهم هي: txtbox(0) , txtbox(1), txtbox(2) وهكذا الحال في انه لو تم عمل نسخ لكائن معين فان الفيجوال سوف تسألك فيما اذا أردت ان تحوله إلى مصفوفة تحكم أي سوف تغير الـ Index فقط وتبقي الاسم كما هو لان النسخ يبقي على الاسم!
Left	اليسار: هذه الخاصية تحدد موقع الكائن من خلال الجهة اليسرى له أي بمعنى أخر الاحداثي السيني للكائن ولكن تاخذ الجهة اليسرى للكائن فلو كان الكائن مربع فانها سوف تحدد موقع الضلع الايسر
Link	ارتباط : وهي خاصية تستخدم مع الكائنات التي ترتبط خارجيا
Mouse icon	ايقونة الفارة: هذه الخاصية تستخدم في تحديد شكل ايقونة الفارة عن المرور من فوق الكائن أي في حالة الوقوف فوق الكائن فان شكل المشيرة سوف يتغير الى الشكل الذي قمنا نحن باختياره وايضا يتم تحديد الايقونة من ملف خارجي، كمثال على ذلك تلاحظ في بعض الازرار عند الوقوف فوقه فان ايقونة الفارة تتغيير الى شكل يد او شكل معين

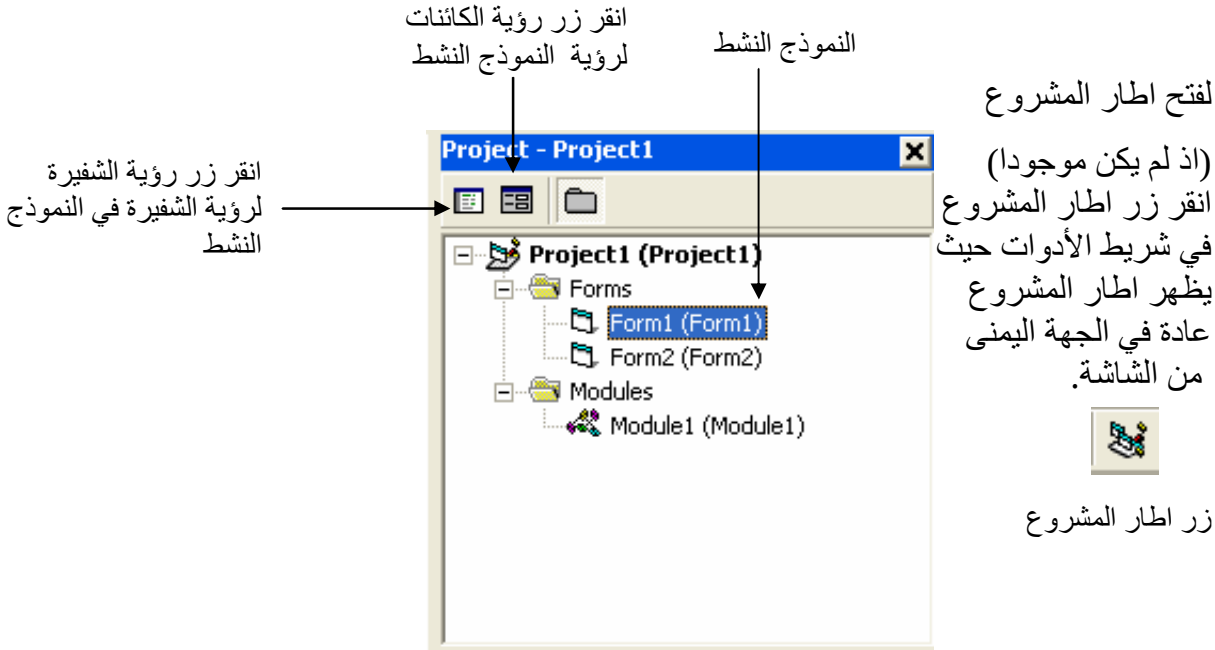
Mouse pointer	مؤشر الفارة: وهي خاصية تستخدم لتحديد شكل الفارة (ايقونة الفارة) حيث تحوي على سبعة عشر نوع ، تبدا من الـ 0-Defualt وهو شكل المشيرة في الحالة الطبيعة الى 15-Size NE SW، هذه الايقونات توجد داخل الفيجوال بيسك تعطينا ايها لتنسيق العمل ، ولكن الخيار السابع عشر هو 99- Customize ، يستخدم هذا الخيار مع الخاصية السابقة عند تحديد شكل ايقونة الفارة من ملف خارجي واردنا اظهارها سوف نقوم باختيار الـ 99- Customize حتى نرى تاثير الخاصية السابقة.
Moveable	قابلية التحريك: تستخدم هذه الخاصية مع النماذج (واجهة البرنامج) حيث تستطيع من خلالها تحديد فيما اذا كان النموذج قابل للتحريك ام لا حيث يحوي على خيارين هما : True يعني هذا ان النموذج قابل للتحريك، او False وتعني انه غير قابل للتحريك . كمثال على ذلك عند الضغط على شريط العنوان لاي نافذة في الوندوز مع استمرارية الضغط على الماوس وسحب النافذة الى مكان معين سوف ترى ان النافذة قد تحركت فهذا يعن انها قابلة للحركة.
Right To Left	اليمين لليسار: تستخدم في حالة كون النصوص باللغة العربية حيث سوف تظهرها من الجهة اليمنى.
Top	اعلى: هذه الخاصية تحدد موقع الكائن من خلال الجهة العليا له أي بمعنى اخر الاحداثي الصادي للكائن ولكن تاخذ الجهة العليا للكائن فلو كان الكائن مربع فانها سوف تحدد موقع الضلع الاعلى
Width	العرض: هذه الخاصية تمكن من تحديد عرض الكائن فيمكنك تحديد ابعاد النموذج او ابعاد مربع النص او عرض زر امر او أي شيء اخر يكون مرئي للمستخدم ضمن واجهة البرنامج و تستخدم مع الخاصية Height لتحديد ابعاد الكائن أي العرض والارتفاع، حيث تستطيع تغيرها يدويا او برمجيا
Tap index	فهرس التاب: وهي خاصية تحدد رقم معين لكل كائن مرئي أي يستطيع المستخدم ان يراه في حال وجوده على النموذج وان يتعامل معه فمثلا الاشكال الهندسية والخط يستطيع المستخدم ان يراها ولكنه لا يستطيع التعامل معها أي لايستطيع النقر عليها او اختيارها في حال تشغيل البرنامج، وكما قلنا فانها تعطي رقم فريد لكل كائن ، حيث تستخدم في حالة الضغط على مفتاح الـ Tab من الكيبورد فان تنشيط الكائنات على النموذج سوف يتغيير وفقا للـ Tabindex أي اقل رقم هو الذي يكون الاول واكبر رقم هو الاخير .
ShowInTaskbar	اظهار في شريط المهام: حيث تستعمل هذه الخاصية مع النماذج (واجهة البرنامج) وهي تعني اظهار البرنامج في شريط المهام عند اشتغال البرنامج او لا، لذلك تحتوي على قيمتين هما: True اظهار و False عدم اظهار
Visible	مرئي: وهي خاصية مهمة وكثيرة الاستعمال حيث هنا تحدد أي من الكائنات التي تريدها ان تكون مرئية للمستخدم وتظهر في واجهة البرنامج واي منها لاتريدها ان تظهر للمستخدم أي تبقى وتؤدي عملا ولكن وراء الكواليس ، او يمكنك ان تظهرها فيما بعد ، يمكنك من ان تغير هذه الخاصية برمجيا، كمثال على ذلك توجد بعض البرامج تطلب اسم دخول و كلمة سرية في البدء حتى تستطيع الولوج الى داخل البرنامج ففي البدء يكون كل النموذج غير مرئي لكن فقط واجهة اسم المرور والرقم السري وزر امر واحد او اثنان يكونان مرئيان ففي حال ادخالهما بشكل صحيح فانه يتم اظهار واجهة البرنامج للمستخدم حيث انه تم تغيير الخاصية Visible لواجهة البرنامج الى اظهار ، لذلك هي تملك قيمتين وهما : True أي جعل الكائن مرئي وهي الحالة الطبيعة لكل كائن عند ادراجه ، وكذلك القيمة False وهي تعني جعل الكائن غير مرئي.

ملاحظة :
جميع الخواص السابقة التي ذكرناها والتي لم نذكرها يمكن تغييرها عن طريق اطار الخصائص اثناء التصميم او يمكن تغيرها برمجيا أي اثناء تشغيل البرنامج بكتابة الشفيرة البرمجة لها للتغيير

اطار (نافذة) المشروع:

يتألف برنامج المشروع من عدة ملفات مجموعة سويا ، عندما يكتمل البرنامج. لمساعدتك على التبديل بين المكونات خلال العمل على مشروع ، أضاف مصممو فيجوال بيسك **أطار المشروع** الى بيئة البرمجة (يسمي بعض الاشخاص هدا الاطار مستكشف المشروع). يسرد أطار المشروع كل الملفات المستعملة في عملية البرمجة ويؤمن وصولا اليها من خلال زرين هما رؤية **الكائنات** ورؤية **الشيفرة**. عندما تضيف ملفات فردية الى المشروع وتحفظها فيه أو تحذفها منه بأستعمال أوامر القائمتين Project و File ستبدوا هذه التغييرات في أطار المشروع .

يسمى ملف المشروع (الذي يحتوي على لائحة بكل الملفات المساعدة في مشروع برمجة) ملف **مشروع فيجوال بيسك** وله ملحق (.vbp) . في أصداري فيجوال بيسك ٥ و ٦ ، يمكن تحميل اكثر من ملف مشروع واحد في أطار المشروع بنفس الوقت ، ويمكنك التنقل بينها عبر نقر أسم المشروع . تحت اسم المشروع ، يعرض اطار المشروع مكونات كل مشروع في بنية شجرية مشابهة لتصميم مستكشف ويندوز. يمكنك توسيع أو طي فروعها ، بما في ذلك النماذج والوحدات النمطية وغيرها من الفئات ، عبر نقر العلامات + و - بجانب المجلدات ، أنظر الشكل (1-7)



شكل (1-7) اطار المشروع

- بعد ان انهينا شرح واجهات بيئة فيجوال بيسك بحمد الله سنتطرق الى شرح القوائم بشكل سريع.

قوائم برنامج لغة فيجوال بيسك

يحتوي فيجوال بيسيك علي ١٣ قائمة، و هم كالتالي:

القائمةFile:

تحتوي هذه القائمة على اوامر اساسية خاصة للمشاريع بشكل عام، كإنشاء مشروع جديد، حفظ محتويات المشروع، طباعة محتويات المشروع وترجمة المشروع وتحويله الى ملف ثنائي exe .

الميزة التي اضيفت ل VB6 هي امكانية فتح اكثر من مشروع في نسخة واحدة من البيئة، وهي ميزة تعرف بالمشاريع المتعددةMultiple Projects.

القائمةEdit:

تحوي هذه القائمة على اوامر التحرير القياسية كالقص، النسخ واللصق . بالاضافة الى اوامر خاصة بقواعد البيانات في حالة وجود قاعدة بيانات في نافذة عرض البيانات . Data View

معظم الاوامر الواردة في اسفل هذه القائمة تستخدمها مع نافذة محرر الاكواد Code Window أطار الشفيرة.

القائمةView:

ذكرت معظم محتوياتها في فقرة" نوافذ بيئة فيجوال بيسك" ، حيث تحتوي على اطار نافذة المشروع و اطار الخصائص وكذلك اطار الخصائص و محتويات اخرى.

القائمةProject:

معظم اوامرها خاصة بمحتويات المشاريع، فهي تمكنك من اضافة عنصر او عناصر من عناصر المشروع كنوافذ النماذج Forms ، ملفات البرمجة Module ، فئاتClasses ....الخ .

المزيد ايضا، يمكنك اضافة ادوات تحكم ActiveX Controls اضافية عن طريق الامر Components "وهي ادوات تحكم تضاف الى مربع الادوات" او تضمين مكتبات ActiveX DLLخارجية عن طريق الامر References.

القائمةFormat:

الاوامر الموجودة في هذه القائمة خاصة بتنسيق الادوات التي تضعها على نافذة النموذج من ناحية موقعها على النافذة، فتوجد اوامر مرنة توفر عليك جهد محاذاة الادوات او توسيطها على النافذة، بالاضافة الى تغيير ترتيب ظهور الادوات أي وضع أداة فوق الكل او أداة خلف الكل.

الامر الاخير. Lock Controls تستخدمه اذا كنت راضيا عن تصميم الادوات وتود منع نفسك من تغيير احجامها او مواقعا عن طريق الخطأ، هذا القفل تستطيع فتحة بكل بساطة باختيار نفس الامر مرة اخرى.

القائمةDebug:

معظم اوامر التنقيح وضعت اسفل هذه القائمة . من هذه الاوامر اختيار طريقة تنفيذ البرنامج، كتنفيذ سطر واحد منه Step Into ، اجراء كامل Step Over ، امر سابق Step Outاو التنفيذ حتى الوصول الى السطر الذي يوجد عليه مؤشر الكتابة RuntoCursor . وبالنسبة لنقاط القطع BreakPointsفهي علامات تظهر مبدئيا باللون الاحمر على سطر معين بحيث تتم عملية الايقاف المؤقت للبرنامج عند الوصول الى هذه العلامات.

القائمةRun:

عن طريق هذه النافذة البسيطة تستطيع تنفيذ البرنامج وتمكنك من اختيار الاوامر الاخرى كالايقاف المؤقت Break او انهاء عملية تنفيذ البرنامج . End بالنسبة للامر Start with Full Compileهو مشابه لامر التنفيذ Start ولن تحتاجه الا في حالات نادرة.

القائمةQuery:

هذه القائمة جديدة على مبرمجي VB5 وهي متوفرة لنسخة المحترفين Professional Editionوالمؤسسات Enterprise Edition للاصدار السادسVB6. اوامر هذه القائمة غير ممكنة حتى تنشئ جملة استعلام SQL باستخدام الأداة Microsoft Query Builder.

القائمةDiagram:

ايضا هذه قائمة جديد ة على مبرمجي VB5 ومتوفرة لنفس النسخ المذكورة في الفقرة السابقة . اوامر هذه القائمة غير ممكنة الا في حالة تعاملك مع قاعدة بيانات QL Serverاو ORACLE.

القائمةTools:

تحتوي على اوامر مختلفة التصانيف كمحرر القوائم Menu Editor ومسهل كتابة الاجراءات Add Procedureوغيرها ...اذا ادرت تخصيص بيئة فيجوال بيسك فالامر Options يمكنك من الوصول الى صندوق الحوار Options الذي يوفر لك عشرات الخيارات والخاصة بتغيير اعدادات بيئة فيجوال بيسك.

القائمةAdd-Ins:

الاوامر الموجودة في هذه القائمة عبارة عن برامج مستقلة تسمى الاضافات Add-Ins هدفها توفير خدمات اضافية لبيئة فيجوال بيسك تزيد من مرونتها . تطوير هذا النوع من البرامج خارج نطاق الكتاب.

القائمةWindow:

تحتوي هذه القائمة على طريقة عرض النوافذ التي تعمل عليها حيث يمكنك الاختيار بين النوافذ ويمكنك تحديد أي نافذة ستكون فعالة أي نافذة الشفيرة او نافذة النموذج او يمكنك عرضهما طوليا او عرضيا معا.

القائمةHelp:

بالنسبة لمستخدمي الاصدار السادس من Visual Basic ، فلن يتمكنوا من الوصول الى التعليمات الفورية الا في حالة انزال نسخة من مكتبة Microsoft DeveloperNetwork والمألوفة بالاختصارMSDN.

مصطلحات هامة

هناك بعض المصطلحات التي يجب معرفتها عند العمل مع فيجوال بيسك

- **أدوات التحكم** : وهي أدوات تحكم تم تصميمها لاستخدامها في فيجوال بيسك بأكثر من طريقة.
- **حدث Event** : هو فعل ما يقوم به الجهاز أو المستخدم مثل ضغط زر من لوحة المفاتيح أو نقر بالفارة
- **الوظائف Methods** : وهي وظائف محددة يمكن للكائن ان يقوم بها .
- **كائن Object** : هو كائن (أداة) خاص له مميزاته الخاصة من خصائص ووظائف وأحداث يمكن ان ينجزها.
- **إجراء Procedure** : هو إجراء مقطع او أكثر من التعليمات في البرنامج (في جزء الشفيرة) وتكتب عادة لغرض معين ، وغالبا ما تكون متصلة بحدث لذا تسمى الإجراءات الحدثية Events Procedure.
- **الخصائص Properties**: وهي خصائص الكائن مثل لونه وموقعة في واجهة البرنامج ونوع الخط ... الخ.

انهاء برنامج فيجوال بيسك

هناك عدة طرق لانهاء البرنامج

1. عن طريق الذهاب الى القائمة File ثم نختار Exit.
2. عن طريق غلقها من مربع التحكم "مربع اغلاق" الموجود في شريط العنوان .
3. عن طريق لوحة المفاتيح بالضغط على المفتاحين Alt + F4 معا.

عندما تقوم باغلاق البرنامج فانه سوف يسالك فيما اذا كنت تريد ان تحفظ التغيرات التي عملتها في البرنامج فاذا لم تكن تريد حفظ التغيرات فقم بالضغط على زر No واذا اردت الحفظ فاختر زر Yes حيث سوف يتم سوالك عن المكان الذي تريد فيه حفظ برنامجك

ملاحظة:

ستقوم في هذا الكتاب بتنفيذ تمارين منوعة حيث سنحتاج بعض هذه التمارين في امثلة اخرى (أي سنقوم بتطويرها في تمارين اخرى) لذلك عليك بخزنها لانك ستحتاجها في المستقبل لاكمال الدروس المقررة في هذا الكتاب.
افضل طريقة لعمل ذلك هي:
انشاء مجلد في القرص "D:\" حيث سنسميه VBLessons ونقوم بعمل مجلد جديد لكل برنامج نقوم بعمله ويكون هذا المجلد الجديد داخل المجلد VBLessons ونقوم باعطاء المجلد الجديد اسم يدل على البرنامج الذي نقوم بتصميمه وذلك لتسهيل عملية تذكر و فتح البرامج التي صممناها. أي عمل ارشفة للبرامج.

- **كتابة اول برنامج لك**

ستتعلم في هذا الدرس كيفية:

- ✓ انشاء واجهة برنامج جديد.
- ✓ ظبط خصائص كل كائن في واجهة البرنامج.
- ✓ كتابة شفيرة في البرنامج.
- ✓ حفظ البرنامج وتشغيله.
- ✓ بناء ملف تنفيذي.

كما تعلمت في الدرس الاول ، تحتوي بيئة مايكروسوفت فيجوال بيسك على عدة ادوات فعالة لمساعدتك في تشغيل برامجك وادارتها. ويحتوي فيجوال بيسك ايضا على كل شيء تحتاج اليه لبناء برامج خاصة بك متوافقة مع ويندوز من الصفر.
ستتعلم في هذا الدرس كيفية انشاء واجهة بسيطة لكن جذابة بواسطة ادوات التحكم المتوفرة في مربع الادوات. ثم سستتعلم كيفية تخصيص عمل تلك الادوات بواسطة مميزات خاصة تدعى خصائص. ثم سترى كيفية تحديد الطريقة التي تريد ان يشتغل بها البرنامج بواسطة شفيرة برمجية نصية.واخيرا سوف تتعلم كيفية حفظ برنامجك الجديد وتشغيله وكيفية تصريفه كملف تنفيذي.

اول برنامج لك في فيجوال بيسك Welcome Pro:

البرنامج الذي سوف تنشئه متوافق مع ويندوز اسمه Welcome Pro وهو برنامج الترحيب، واجهة البرنامج بسيطة ويمكن انشائها في دقائق مع فيجوال بيسك

شكل (2-1)

سيبدو برنامجك على هذا النحو بعد الانتهاء من تصميمه

خطوات البرمجة

تحتوي واجهة البرنامج Welcome Pro على ستة ازرار امر وعلى اطار للعنوان "تسمية" وكذلك على مربع نص. يتم وضع هذه العناصر في البرنامج عن طريق وضع ثمانية كائنات على النموذج ثم تغيير عدة خصائص لكل كائن منها. بعد تصميم واجهة البرنامج اضفنا شفيرة للست ازرار لمعالجة عمليات النقر عليهم . لانشاء أي برنامج في فيجوال بيسك هناك ثلاث خطوات رئيسية : انشاء واجهة البرنامج وضبط الخصائص وكتابة الشفيرة.

انشاء واجهة البرنامج

ستبدأ ببناء Welcome Pro بإنشاء مشروع جديد ثم استعمال أدوات التحكم الموجودة في مربع الأدوات لإنشاء الواجهة.

انشاء الواجهة

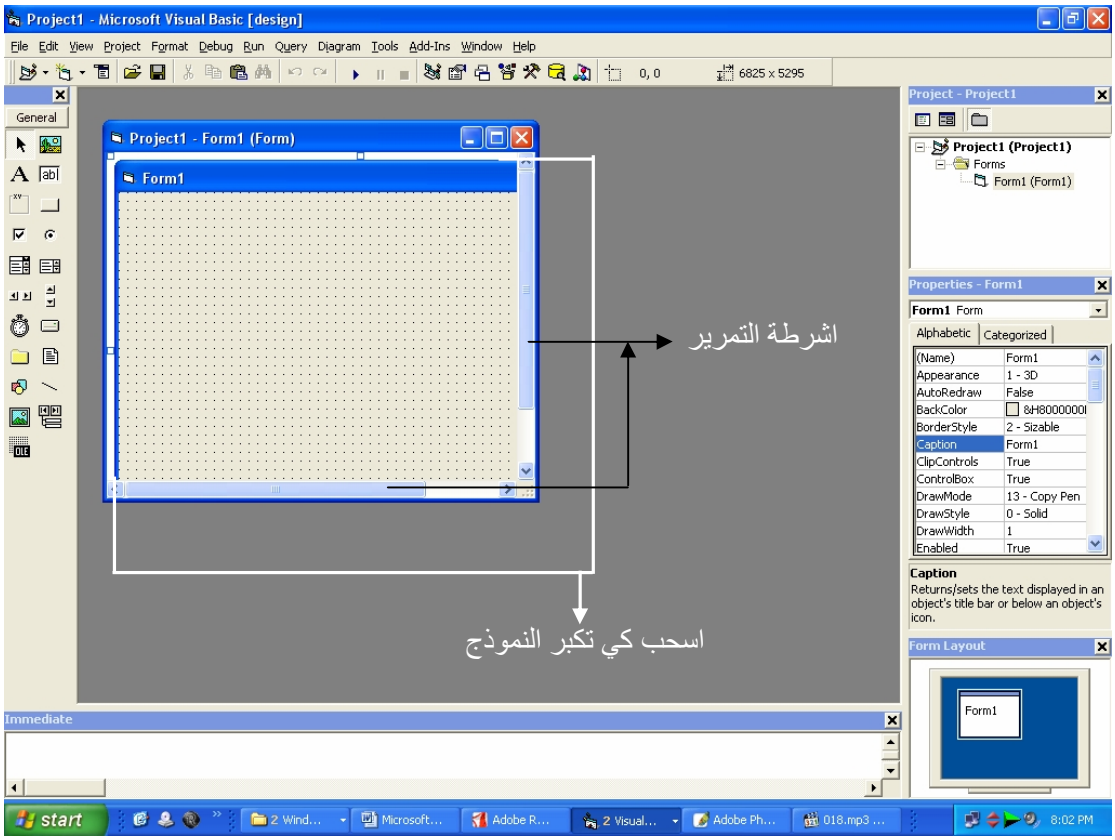
1. قم بتشغيل البرنامج عن طريق Start ثم الخطوات المتبعة في التشغيل
2. ستظهر لك الواجهة التي تسالك عن ماذا تريد ان تعمل كما مبين في الشكل (2-2)

شكل (2-2)

3. اختر New ثم Standard EXE ثم انقر فوق زر Open
4. اما في حال كنت قد شغلت البرنامج واردت ان تنشئ مشروع جديد فاختر الامر File → New Project
5. انقر No اذا سئلت عما اذا كنت تريد حفظ التغييرات على المشروع السابق في حال كنت قد فتحت مشروع سابق.
6. ثم انقر OK لانشاء برنامج فيجوال بيسك 32 بت قياسي.

يهيئ فيجوال بيسك الوضع لمشروع برمجة جديد ويعرض نموذجا خاليا كي تستعمله لانشاء واجهة البرنامج ، ستكبر النموذج ثم ستنشئ الازرار في الواجهة.

7. ضع مؤشر الماوس فوق الزاوية السفلى اليمنى لاطار النموذج (وليس لاطار مستوعب المشروع) الى ان يتغيير المؤشر الى مؤشر تحجيم ثم كبر حجم النموذج ليسع الكائنات الموجودة في برنامجك . بينما تكبر حجم النموذج ستظهر اشرطة تمرير في اطار المشروع كما مبين في الشكل (2-3).



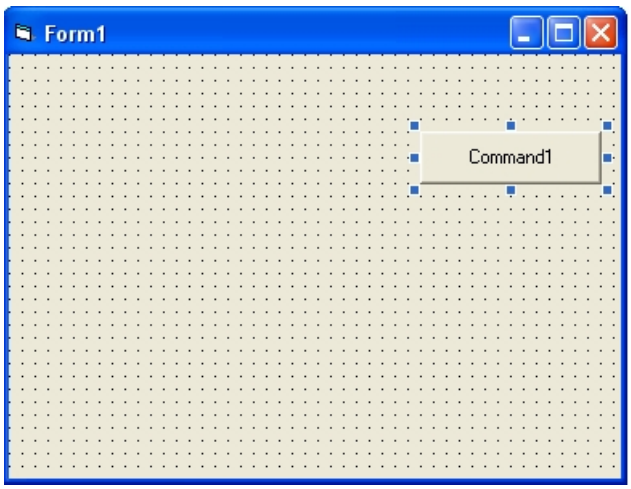
شكل (2-3)

- قم ايضا بتغيير اطار مستوعب المشروع بنفس الطريقة السابقة حتى تستطيع رؤية واجهة التصميم كاملة

8. انقر اداة التحكم زر الامر في مربع الادوات ثم ضع مؤشر الماوس فوق النموذج. تصبح اداة التحكم زر الامر منتقاة، ويتغير مؤشر الماوس الى شكل متصالب عند وضعه فوق النموذج. تم تصميم الشكل المتصالب لمساعدتك على اعطاء زر الامر شكل المستطيل . عندما تضغط زر الماوس الايسر باستمرار وتسحب، ياخذ زر الامر شكلا ويستقر على الشبكة المكونة من تقاطع النقاط على النموذج. حاول انشاء اول زر امر لك الان.

زر أمر

9. انقل مؤشر الماوس الى الجهة اليمنى ، انقر زر الماوس الايسر باستمرار واسحب الى الاسفل واليسار . توقف عن السحب وافلت زر الماوس عندما يصبح زر الامر مشابها للزر المبين في شكل (2-4).



شكل (2-4)

يظهر على النموذج زر امر مزود بمقابض انتقاء. يسمى هذا الزر Command1 وهو اول زر امر في البرنامج (تذكر اسم هذا الزر لانك ستشاهده لاحقا عند كتابة شفيرة البرنامج).

يمكنك نقل ازرار الاوامر بسحبها بواسطة الماوس وتغيير حجمها باستعمال مقابض الانتقاء عندما يكون فيجوال بيسك في وضع التصميم. لكن عندما يكون البرنامج مشتغلا، لن يتمكن المستخدم من نقل عناصر الواجهة الا اذا غيرت خاصية معينة في البرنامج للسماح بذلك.

اضافة زر امر ثاني وثالث و .. و... و سادس

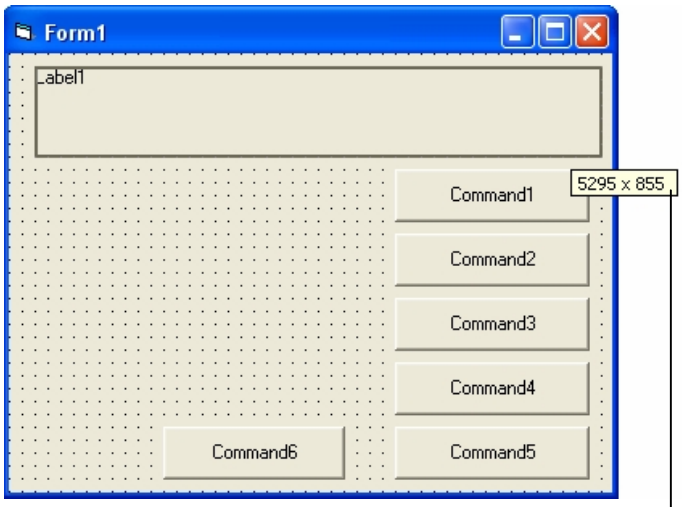
يمكنك حذف أي كائن بانتقائه على النموذج ثم ضغط المفتاح Delete

1. انقر اداة التحكم زر الامر في مربع الادوات.
2. انشئ زر امر ثاني تحت الزر الاول على النموذج (اجعل حجمه كحجم الزر الاول بداعي الانسجام).
3. انشئ ست ازرار امر كما مبين في شكل (2-5).
4. انقل او غير حجم الزر حسب الضرورة بعد وضعه . اذا ارتكبت خطا، يمكنك حذف زر الامر والبدء من جديد.

اضافة تسمية العنوان " Label"

تسمية

1. انقر اداة التحكم تسمية Label في مربع الادوات ثم ضع مؤشر الماوس فوق النموذج . يتم انتقاء اداة التحكم تسمية ويتغير مؤشر الماوس الى شكل متصالب عندما يصبح فوق النموذج.
2. انشى مربعا على طول النموذج كما مبين في الشكل (2-5)



شكل (2-5)

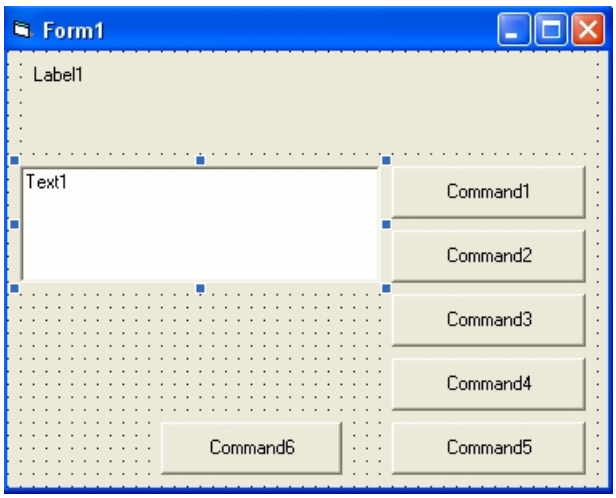
❖ يسمى هذا المربع الذي يحتوي على قياسات افقية وعمودية ، مربع تحجيم.

تعطى الارقام الابعاد الافقية والعمودية، على التوالي، للكائن الذي تنشئه.

الارقام هي في وحدة قياس تسمى تويب (twip) وهي تساوي 1/20 من النقطة (النقطة تساوي 1/72 من البوصة وبالتالي التويب يساوي1/1440 من البوصة) النقطة هي الـ Pixel.

اضافة مربع نص

1. انقر اداة التحكم مربع نص TextBox في مربع الادوات.
2. انقر واسحب على الجهة اليسرى للنموذج حتى يصبح شكل النموذج كما مبين في الشكل (2-6).



شكل (2-6)

حاول ان تجعل نموذجك يبدو بهذا الشكل.

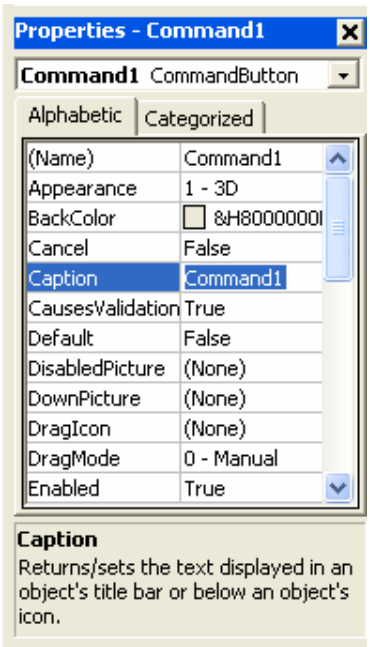
ضبط الخصائص

كما اكتشفت في الدرس الاول . انه يمكنك تغيير الخصائص بانتقاء كائنات على النموذج وتغيير اعداداتها في اطار الخصائص. ستبدا بضبط الخصائص في هذا البرنامج عبر تغيير اعداد الخاصية Caption لازرار الامر.

اعداد خصائص زر الامر

1. انقر زر الامر الاول (Command1) على النموذج. ستلاحظ إحاطت زر الامر بمقابض انتقاء
2. انقر شريط عنوان اطار الخصائص نقرا مزدوجا. ستلاحظ انه سوف يفصل عن بقية النوافذ ويكبر حجمه ولارجاعة الى وضعه الطبيعي قم بالنقر على الشريط نقرا مزدوجا مرة أخره ، ستلاحظ رجوع نافذة الخصائص بين النوافذ.

يعرض اطار الخصائص اعدادات (خصائص) زر الامر الاول . انظر شكل (2-7)



شكل (2-7)

3. انقر نقرا مزدوجا الخاصية Caption في العمود الايسر لاطار الخصائص.

يظهر اعداد الخاصية الحالي (" command1 ") منتقى في اطار الخصائص.

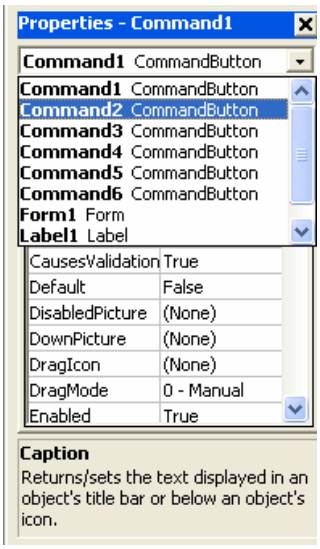
4. اكتب "التاريخ" ثم اضغط المفتاح Enter.

سوف تتغير الخاصية Caption الى "تاريخ " بدل "command1" في اطار الخصائص وفي النموذج ، غير الان الخاصية Caption لبقية الازرار الى:

الخاصية	تغير الى
Caption للزر الاول Command1	"التاريخ"
Caption للزر الثاني Command2	"الوقت"
Caption للزر الثالث Command3	"رسالة الترحيب"
Caption للزر الرابع Command4	"مسح"
Caption للزر الخامس Command5	"خروج"
Caption للزر السادس Command6	"تغيير العنوان"

5. افتح مربع سرد الكائنات في اعلى اطار الخصائص.

تظهر كل كائنات واجهة البرنامج في مربع السرد هذا. كما في شكل (2-8)



شكل (2-8)

6. انقر Command2 وهو زر الامر الثاني في مربع السرد

تظهر اعدادات خصائص زر الامر الثاني في اطار الخصائص.

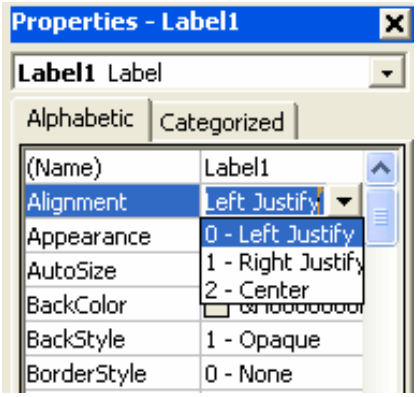
ملاحظة:

ان استعمال مربع سرد الكائنات هو طريقة مريحة للتنقل بين الكائنات الموجودة في برنامجك. بامكانك ايضا التنقل بين الكائنات بنقر أي كائن على النموذج بشكل فردي.

ضبط خصائص تسمية العنوان

1. انقر تسمية العنوان الموجودة على النموذج حيث ستظهر خصائص التسمية في اطار الخصائص
2. انقر الخاصية Alignment ثم انقر سهم مربع السرد المنسدل الذي يظهر على اليمين.

تظهر قائمة بخيارات محاذاة في مربع السرد انظر شكل (2-A).



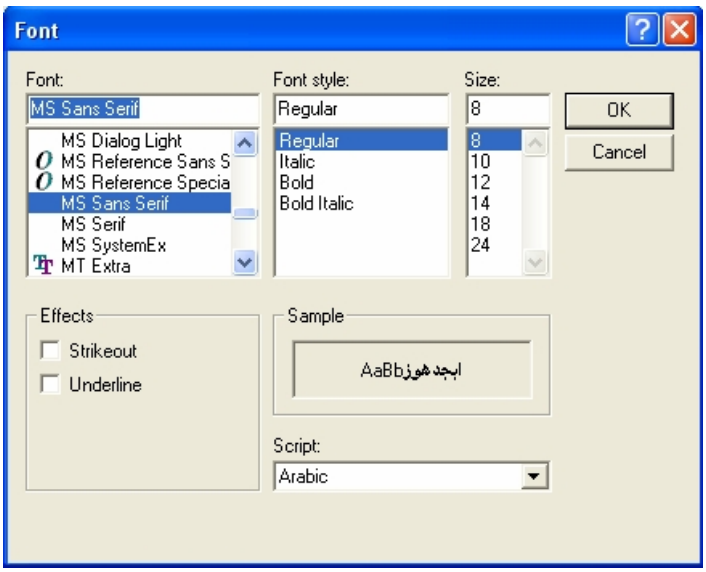
شكل (A -2)

3. انقر الخيار 2-Center.

تتغير الخاصية Alignment "محاذاة" للتسمية المنتقاة الى 2-center أي سوف تتوسط الكتابة داخل التسمية.

4. انقر نقرا مزدوجا الخاصية Font في اطار الخصائص.

يظهر مربع الحوار Font كما مبين في الشكل (2-9)



شكل (2-9)

5. غير الخط الى Time New Roman ونمط الخط الى Bold وحجم الخط الى 24 ثم انقر OK.

تظهر التسمية بالخط والنمط والحجم الذي حددته. سنغيير الان التسمية التوضيحية Caption لتسمية العنوان الى "Welcome Pro"

6. انقر نقرا مزدوجا الخاصية Forecolor في اطار الخصائص.

سيظهر امامك مربع سرد يتظمن علامتي تبويب System و Palette وهي خيارات

لتغيير لون الكائن، تعرض علامة التبويب system الالوان الحالية المستخدمة لعناصر الواجهة في كمبيوترك. تستخدم هذه الخاصية لتغيير لون الخط للكائن.

7. انقر علامة التبويب Palette ثم مربع اللون الرصاصي الفاتح.
سيتغير لون النص في مربع التسمية الى رصاصي فاتح.
8. انقر نقرا مزدوجا الخاصية Backcolor حيث ستظهر نافذة كما في الخاصية السابقة. قم بانتقاء اللون الرصاصي القاتم ، حيث سيتغير لون الخلفية لمربع التسمية الى رصاصي قاتم.

ضبط خصائص مربع النص

1. انقر كائن مربع النص على النموذج
2. انقر الخاصية Alignment واختار 1-Right Justify أي محاذاة الى اليمين
3. ثم انتق الخاصية Font وغير نوع الخط الى Time New Roman كما وضحنا سابقا لكن الحجم اجعله 18 نقطة.
4. انتق الخاصية RightToLeft واختر True
5. انتق الخاصية MultiLine واختر True وهي تعني السماح لمربع النص ان يظهر عدة اسطر وليس سطر واحد فقط.
6. انتق الخاصية Text وقم بالضغط على مفتاح Delete لحذف كلمة "Text" من مربع النص التي تظهر بداخله.

ملخص الخصائص للبرنامج Welcome Pro

الكائن	الخاصية	الاعداد
Command1	Caption	"التاريخ"
Command2	Caption	"الوقت"
Command3	Caption	"رسالة الترحيب"
Command4	Caption	"مسح"
Command5	Caption	"خروج"
Command6	Caption	"تغيير العنوان"
Label1	Alignment	2-center
	Font	24 نقطة , Bold , Time New Roman
	Forecolor	رصاصي فاتح
	Backcolor	رصاصي قاتم
	Caption	"Welcome Pro"

TextBox1	Alignment	1-Right Justify
	Font	18 نقطة , Bold , Time New Roman
	RightToLeft	True
	Text	فارغ
	MutiLine	True

حيث سيكون شكل البرنامج النهائي كما في شكل (1-2) لكن من دون جملة الترحيب بداخل مربع النص لانك لم تشغل البرنامج بعد.

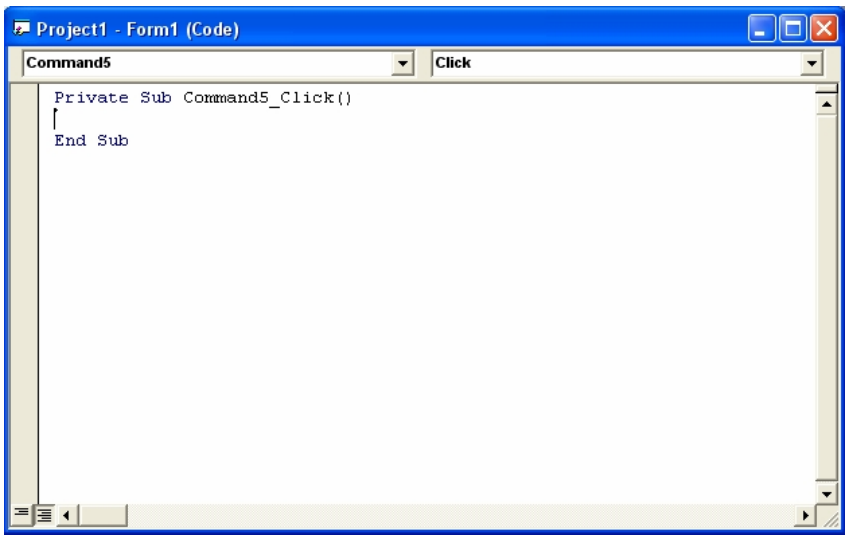
كتابة شفيرة البرنامج

الان اصبحت جاهزا لكتابة شفيرة البرنامج Welcome Pro ، يمكنك الان ان تشاهد البرنامج أي ان تشغله ومن دون كتابة شفيرة البرنامج . ولكن سوف تكتب شفيرة البرنامج في الخطوات القادمة في اطار شفيرة البرنامج.

استعمال اطار الشفيرة

1. انقر زر الامر "خروج" نقرا مزدوجا

سيظهر اطار الشفيرة كما في شكل (2-10)



شكل (2-10)

يوجد داخل اطار الشفيرة الذي امامك جمل برمجية تشير الى بداية ونهاية هذا الروتين الفرعي او الاجراء الحدثي لفيجوال بيسك. الاجراء الحدثي هو كتلة شفيرة مقترنة بكائن معين في واجهة البرنامج.

```
Private Sub Command5_Click()
End Sub
```

الحدث هنا هو نقرة ماوس Click حيث سينفذ البرنامج الشفيرة المكتوبة في هذا الاجراء والتي تكون محصورة بين هاتين الجملتين

Private Sub Command5_Click()

هنا تكتب الشفيرة البرمجية للحدث "نقرة ماوس" ففي حال قام المستخدم بالنقر على هذا الزر فان الشفيرة المكتوبة هنا سوف تنفذ

End Sub

سوف نقوم بكتابة جملة برمجية تقوم بانهاء البرنامج والخروج منه وهي End لذلك يجب كتابتها بين الجملتين السابقتين.

سنكتب الان شفيرة زر "التاريخ"

افتح مربع سرد الكائنات المنسدل في اطار الشفيرة

تظهر كائنات واجهة البرنامج Welcome pro في مربع السرد كما في الشكل (2-11)

شكل (2-11)

انقر Command1 في مربع السرد

يظهر اجراء مقترن بالزر command1 فوق الاجراء الاول.

رغم انك غيرة التسمية التوضيحية لهذا الزر الى "التاريخ" الا ان اسمه بقي command1 ، يمكن ان يكون اسم عنصر الواجهة مختلفا عن تسميته التوضيحية وذلك ليتلائم مع حاجة المبرمج.

اكتب الجمل البرمجية التالية بين الجملتين Private Sub و End Sub ، اضغط Enter بعد كل سطر وتاكد من ان الجمل مطابقة تماما لتلك الموجودة هنا، اذا ارتكبت خطا فان فجوال بيسك سوف يظهر الجملة الخطا باللون الاحمر لذلك عليك تصحيحها او كتابتها من جديد.

```
Private Sub Command1_Click()
Text1.Text = Date
End Sub
Private Sub Command2_Click()
Text1.Text = Time
End Sub
Private Sub Command3_Click()
Text1.Text = "مرحبا بكم في اول برنامج لكم في فيجوال بيسك"
End Sub
Private Sub Command4_Click()
Text1.Text = ""
End Sub
Private Sub Command5_Click()
End
End Sub
Private Sub Command6_Click()
Label1.Caption = "اكتب العنوان الذي تريده هنا "
End Sub
```

بعد الانتهاء من كتابة الشفيرة يجب ان تبدو كما في الشكل (2-12)

شكل (2-12)

نلقي نظرة على الاجراء Command1_Click

يتم تنفيذ الاجراء Command1_Click عندما يقوم المستخدم بالنقر على زر "التاريخ" حيث سوف تنفذ الشفيرة المكتوبة بداخله وهي شفيرة استدعاء التاريخ حيث سوف توضع قيمة التاريخ في مربع النص لكي نستطيع رؤيتها.

نلقي نظرة على الاجراء Command6_Click

يتم تنفيذ الاجراء Command6_Click عندما يقوم المستخدم بالنقر على زر "تغيير العنوان" حيث سوف تنفذ الشفيرة المكتوبة بداخله وهي شفيرة تغيير الخاصية Caption لتسمية العنوان أي لـ (Label1) حيث سوف توضع الجملة التي سوف نكتبها في الخاصية caption لـ label1 حيث سوف تتغير من Welcome Pro الى الجملة التي سوف نكتبها.

حفظ البرنامج

نذهب الى File ثم نختار Save Project As حيث سوف يظهر لنا مربع حوار يطلب منا ادخال اسم النموذج وكذلك مكان الحفظ

قم بتسمية نموذجك باسم معين يتناسب مع برنامجك فمثلا سمه Welcome ثم حدد مكان لخزنه فيه ، يستحسن عمل مجلد جديد لكن مشروع جديد وذلك لكي تعزل ملفات كل مشروع وحدها ولا تختلط لانه عند حفظ المشروع "النموذج" سوف تحفظ معه ملفات اخرى تابعة للغة والمشروع .

ويمكنك حفظ المشروع عن طريق الذهاب الى شريط الادوات واختيار رمز الـ Floppy حيث سيتم ايضا حفظ المشروع.

تشغيل البرنامج

بعد ان قمت بحفظ مشروعك يجب ان ترى فيما اذا كان برنامجك يعمل ام لا ما عليك سوع الضغط على زر تشغيل من شريط الادوات لتشغيل البرنامج او عن طريق الضغط على المفتاح F5 من لوحة المفاتيح او عن طريق القائمة Run.

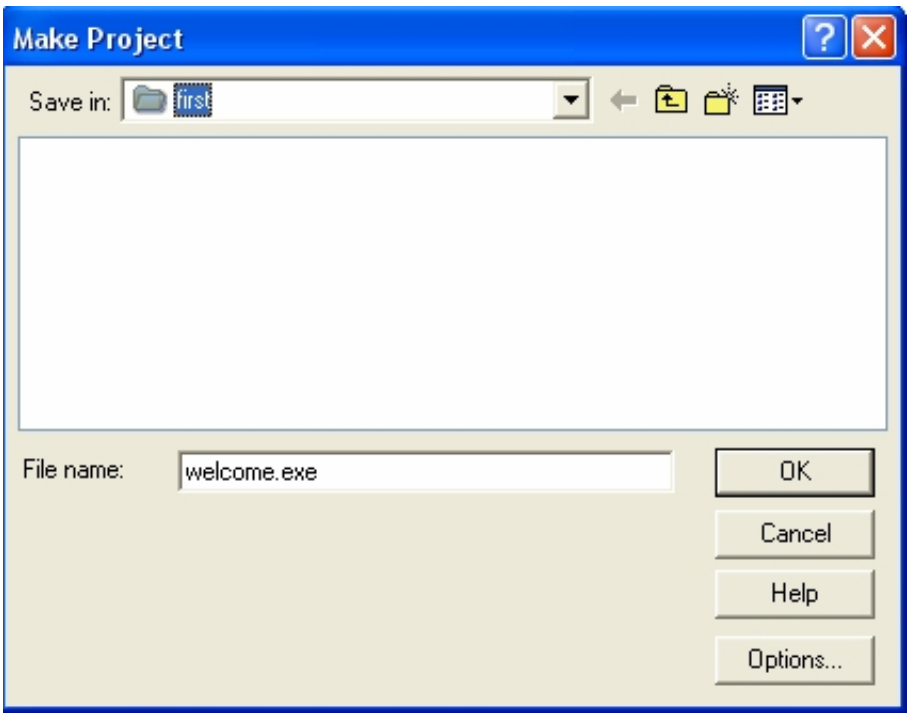
انقر زر "التاريخ" سوف ترى انه سيتم وضع تاريخ اليوم داخل مربع النص كذلك بالنسبة للـ " الوقت" سترى انه تم وضع الوقت الحالي "أي لحظة النقر" داخل مربع النص.

وكذلك الحال مع بقية الازرار يمكن تجربة ذلك من خلال النقر عليهم.

انشاء ملف تنفيذي

بعد ان تاكد ان برنامجك يعمل بشكل صحيح وارد ان تجعله برنامج ينفذ من تلقاء نفسه عند فتحة كاي برنامج اخر أي لا يتم تشغيله بواسطة لغة فيجوال بيسك ، يجب عليك ان تجعله مستقل أي ان يصبح ملف تنفيذي

وذلك عن طريق الذهاب الى قائمة File ثم Make Welcome.exe يضيف فيجوال بيسك اسم البرنامج الى القائمة تلقائيا لذلك نرى انه سماه Welcome.exe لاننا طلبنا منه ذلك. بعد ذلك سيظهر مربع الحوار Make Project كما في الشكل (2-13)



شكل (2-13)

اذا اردت قم بتسميته باسم معين او تركه على الاسم الافتراضي الذي تعطيه فيجوال بيسك، بعد ذلك اضغط على زر Ok وبعد ان قمت بتحديد مساره ، قم بغلق فيجوال بيسك واذهب الى مكان خزن الملف التنفيذي الـ exe ثم قم بتشغيله ستلاحظ انه اصبح برنامج مستقل يعمل بمجرد فتحه.

- **العمل مع ادوات التحكم**

ستتعلم في هذا الدرس كيفية:

- ✓ استعمال كائنات نظام الملفات وكائن صورة لاستعراض الاعمال الفنية.
- ✓ استعمال كائن OLE لتشغيل برامج متوافقة مع ويندوز في كمبيوترك.
- ✓ تثبيت ادوات تحكم اكتف اكس ActiveX.

كما تعلمت في الدرسين 1 و2 ، فان ادوات التحكم في فيجوال بيسك هي الادوات الرسومية التي تستعملها لبناء واجهة برامج فيجوال بيسك. تتواجد ادوات التحكم في مربع الادوات في بيئة البرمجة، ويتم استعمالها لانشاء الكائنات على النموذج عبر سلسلة بسيطة من نقرات الماوس وحركات السحب. ستتعلم في هذا الدرس كيفية استعراض محركات الاقراص والمجلدات في الكمبيوتر ، وتشغيل برامج متوافقة مع ويندوز ، ستساعدك التمارين في هذا الدرس على تصميم برامج فيجوال بيسك خاصة بك وستعلمك اكثر عن الكائنات والخصائص والشفيرة البرمجية. كما ستتعلم كيفية اضافة ادوات تحكم اكتف اكس الى مربع الادوات لتزيد من فاعلية فيجوال بيسك.

استعمال كائنات نظام الملفات

يزود فيجوال بيسك ثلاثة كائنات مفيدة تساعد على الوصول الى نظام الملفات . انها **مربعات سرد محركات الاقراص** التي تتيح لك رؤية محركات الاقراص الصالحة في الكمبيوتر، و **مربعات سرد الدلائل** التي تتيح لك التنقل بين المجلدات الموجودة في محرك اقراص معين، و**مربعات سرد الملفات** التي تتيح لك انتقاء ملف معين في مجلد. في التمرين التالي ستستعمل هذه الكائنات الثلاثة لبناء برنامج Photo Browser يحدد موقع الملفات التي تحتوي على اعمال فنية في الكمبيوتر ويعرضها.

❖ **ملاحظة**

ستستعمل في هذا الدرس ايضا **كائن صورة** . يستطيع كائن الصورة عرض انواع مختلفة من التنسيقات الرسومية : كالصور النقطية ذات الامتداد bmp. والملفات ذات الملحق wmf. والرموز (الايقونات ico.) وملفات الموشر cur. والتنسيقين jpg. و gif. .

البرنامج Photo Browser

يستعمل البرنامج Photo Browser كائنات نظام الملفات الثلاثة وكائن صورة وعدة اسطر شفيرة برمجية لانشاء برنامج يستعرض الصور . بعد الانتهاء من عمل البرنامج يمكنك استعماله في عملك اليومي لاستعراض الصور المخزونة في كمبيوترك. او في قرص مرن او في وحدة خزن مؤقتة.

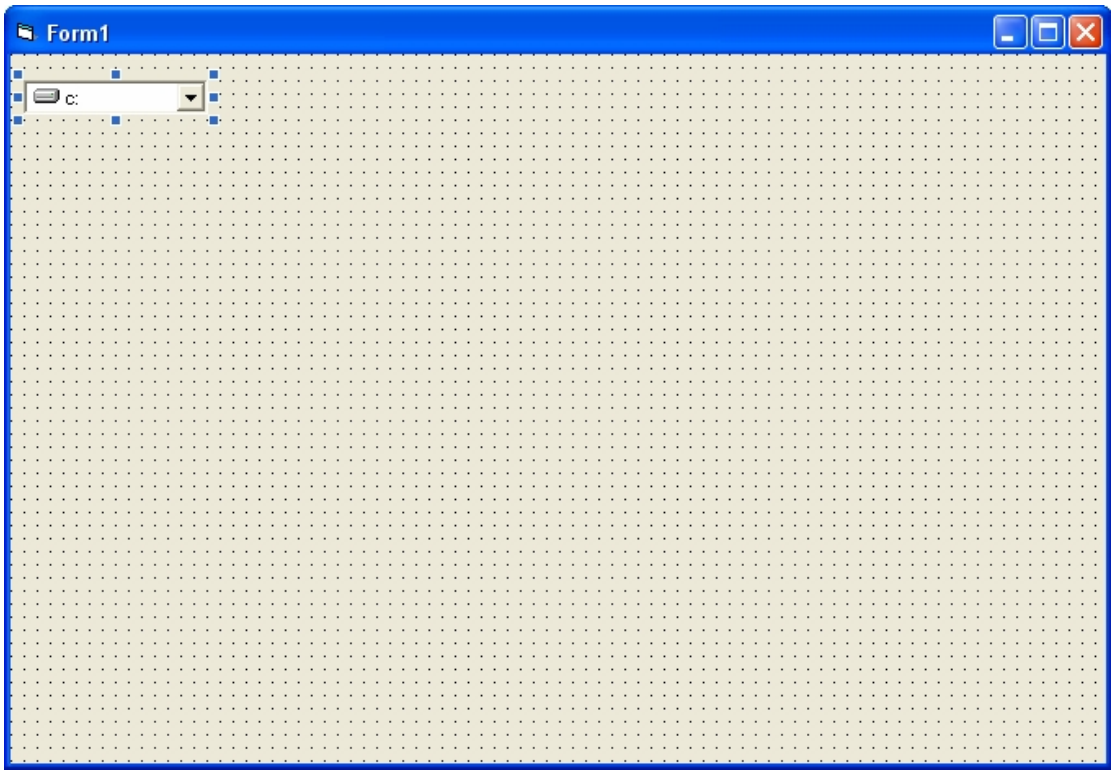
بناء البرنامج Photo Browser

1. اختر الامر New Project < File ثم انقر OK لانشاء ملف تنفيذي قياسي.
2. كبر حجم النموذج لكي يصبح بامكانه استيعاب كائنات نظام الملفات والرسوم التي ستعرضها فيه.

   ستظهر اشرطة تمرير حول النموذج لكي يصبح بامكانك رؤية أي جزء مختف اثناء تطوير برنامجك.
3. انقر اداة التحكم مربع سرد محركات الاقراص DriveListBox في مربع الادوات.

مربع سرد محركات الاقراص

4. انقل مؤشر الماوس الى زاوية النموذج اليسرى العليا ثم ارسم مربع سرد محركات الاقراص كما في الشكل (1-3)



شكل (1-3)

يضع فيجوال بيسك محرك الاقراص الحالي وتسميته في الكائن عندما تنشئه. تظهر هذه المعلومات لمساعدة المستخدم على معرفة محرك الاقراص المنتقى حاليا عندما يستعمل البرنامج، كما يتيح لك مشاهدة المعلومات خلال التصميم ليكون بامكانك تحجيم الكائن كما يجب.

مربع سرد الدلائل

5. انقر اداة التحكم مربع سرد الدلائل (الفولدرات) DirListBox في مربع الادوات واضف مربع سرد الدلائل تحت مربع سرد محركات الاقراص في النموذج ، افسح المجال لظهور اربعة او خمسة مجلدات في مربع السرد.

يؤمن مربع سرد الدلائل وصولا الى المجلدات الموجودة في الكمبيوتر، تظهر المجلدات كما ستبدو عليه عند تشغيل البرنامج. قد تتسرع وتنقر المجلدات الان، لكن بما ان المربع غير نشط لن يفيدك ذلك، فالمجلدات تظهر فقط لتمكنك من تحجيم الكائن بشكل صحيح.

مربع سرد الملفات

6. انقر اداة التحكم مربع سرد الملفات FileListBox في مربع الادوات واضف مربع سرد الملفات تحت مربع سرد الدلائل في النموذج ، افسح المجال لظهور اربعة او خمسة ملفات في مربع السرد.

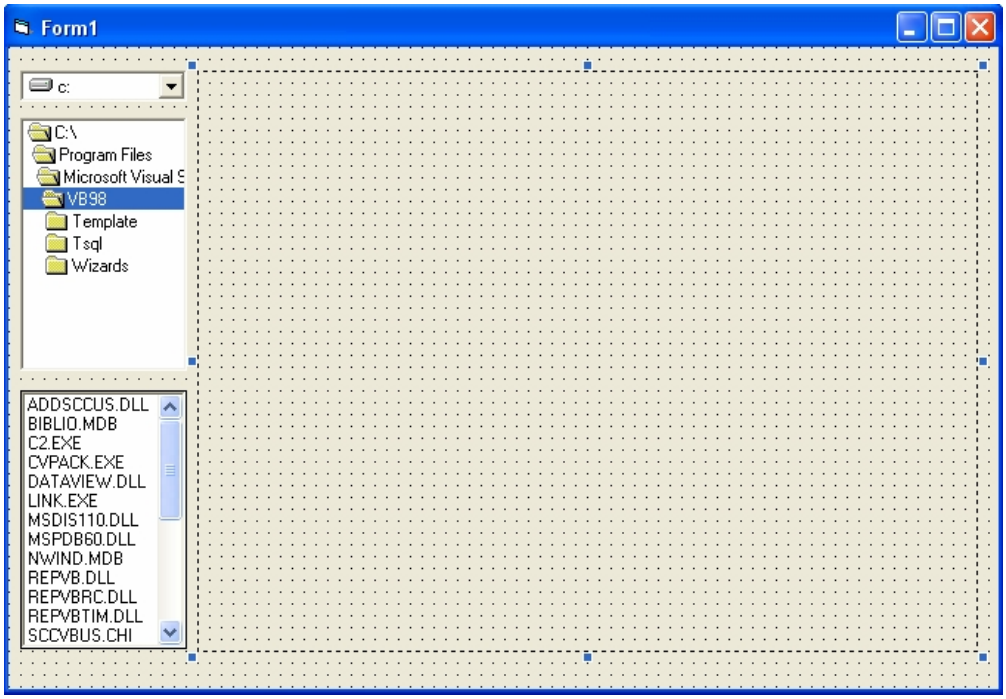
يتيح مربع سرد الملفات للمستخدم امكانية انتقاء ملف موجود في الكمبيوتر، عندما ينتقي المستخدم ملفا، يضع فيجوال بيسك اسم الملف في الخاصية FileName لكائن مربع سرد الملفات، وبطريقة مشابهة تتلقى الخاصية Drive لكائن مربع سرد محركات الاقراص والخاصية Path لكائن مربع سرد الدلائل اسمي محرك الاقراص والمجلد

اللذين ينتقيهما المستخدم في مربعات سرد محركات الاقراص والدلائل، ستستعمل هذه الخصائص الثلاثة معا في البرنامج Photo Browser .

7. انقر اداة التحكم صورة في مربع الادوات واضف مربع صورة كبير على النموذج الى يمين مربعات سرد الاقراص و الدلائل والملفات .

صورة

بعد اضافة هذا المربع يجب ان تبدو الشاشة كما في الشكل (2-3)



شكل (2-3)

8. اضبط الان الخصائص التالية باستعمال اطار الخصائص:

الكائن	الخاصية	الاعداد
File1	Pattern	*.bmp; *.jpg; *.ico; *.gif
Image1	Stretch	True
Image1	BoarderStyle	1-Fixed Single

الخاصية Pattern لمربع سرد الملفات مهمة جدا في هذه الحالة . انها تسرد التنسيقات الصورية الصالحة التي يمكن ان يعرضها فيجوال بيسك في البرنامج داخل مربع الصورة. اذا تركت الخاصية فارغة فان مربع سرد الملفات يعرض كل انواع الملفات داخل المجلد أي يعرض *.* فاذا قام المستخدم باختيار ملف ذي صيغة لا يدعمها مربع الصورة (مثلا *.Tiff) فان ذلك سيؤدي الى حدوث عطل مما يؤدي الى توقف البرنامج، من الافضل ازالة مثل هذه المشاكل قبل ان تبرز اذا امكن ذلك، حيث هنا نحن نريد فقط ان نعرض الملفات الصورية لذلك حددناها في الخاصية Pattern ،

سنقوم الان باضافة بضعة اسطر شفيرة الى الاجراءات المقترنة بكائنات نظام الملفات، تسمى هذه الاجراءات **اجراءات حدثية .**

9. انقر كائن مربع سرد محركات الاقراص على النموذج نقرا مزدوجا ثم اكتب الجملة البرمجية بين الجملتين Private Sub و End Sub في الاجراء الحدثي Drive1_Change اكتب

Dir1.path = Drive1.Drive أي يجب ان تبدو الشفيرة كما مبين

```
Private Sub Drive1_Change()
Dir1.Path = Drive1.Drive
End Sub
```

تقوم هذه الجملة بتحديث الخاصية Path لمربع سرد الدلائل عندما ينتقي المستخدم محرك اقراص من مربع سرد محركات الاقراص. تربط الجملة الكائنين ببعضهما بحيث يسرد مربع سرد الدلائل المجلدات الموجودة في محرك الاقراص المنتقى.

10. اغلق اطار الشفيرة. ثم انقر نقرا مزدوجا مربع سرد الدلائل على النموذج واضف الجملة البرمجية التالية الى الاجراء الحدثي Dir1_Change:

```
File1.path = Dir1.Path
```

تربط هذه الجملة مربع سرد الملفات بمربع سرد الدلائل لكي يتم عرض الملفات التي بداخل المجلد المنتقى في مربع سرد الملفات على شرط ان تطابق الخاصية Pattern أي سوف يعرض مربع سرد الملفات فقط الملفات الصورية.

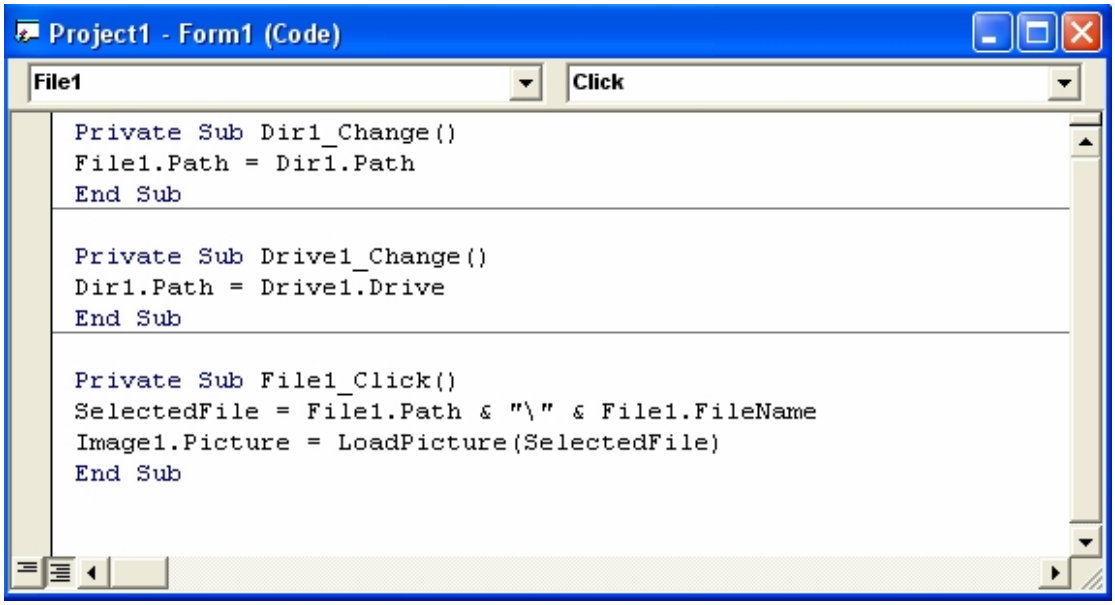
11. اغلق اطار الشفيرة. ثم انقر نقرا مزدوجا مربع سرد الملفات على النموذج واضف الجملة البرمجية التالية الى الاجراء الحدثي File1_Click :

```
SelectedFile = File1.Path & "\" & File1.Filename
Image1.Picture = loadPicture (SelectedFile)
```

هذان السطران هما اساس البرنامج . يستعمل السطر الاول العامل & لدمج الخاصية Path والرمز \ والخاصية Filename التابعة لمربع السرد File1 ويخزن اسم المسار الناتج في المتغير SelectedFile . المتغير هو مكان مؤقت لتخزين المعلومات في البرنامج . في هذه الحالة يحتفظ المتغير SelectedFile بكل اسم ملف قام المستخدم بانتقائه (بما فيه اسم محرك الاقراص والمجلد) . تستعمل الجملة الثانية في الاجراء ألحدثي المتغير

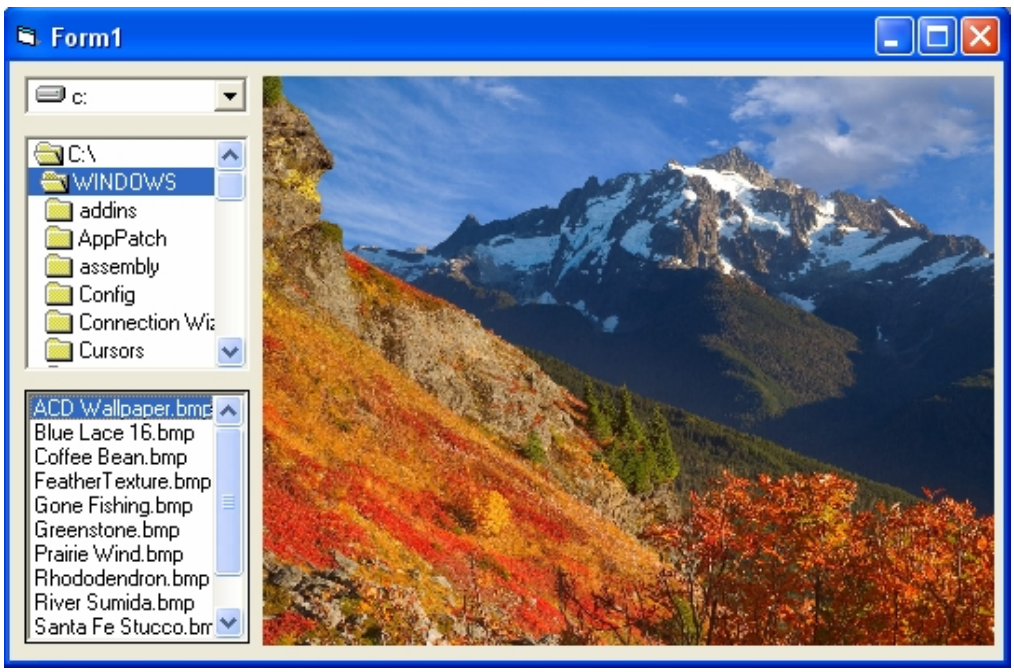
SelectedFile عند تحميلها الملف في مربع الصورة Image1 على النموذج باستعمال الوظيفة LoadPicture والخاصية Picture .

يجب ان يبدو اطار الشفيرة كما مبين في الشكل (3-3)



شكل (3-3)

12.قم بتشغيل البرنامج الان وقم بانتقاء مجلد يحوي على ملفات صورية لرؤية البرنامج كيف يعمل. انظر الشكل (3-4) البرنامج وهو يعمل .



شكل (3-4)

13. انقر زر حفظ المشروع ، قم بانشاء مجلد جديد داخل المجلد الذي كوناه مسبقا VBLessons وسميه Photo ثم احفظ المشروع باسم Photo Browser وذلك لاننا سوف نستخدمه في درس لاحق.

استعمال كائن OLE لتشغيل التطبيقات

احدى اهم ميزات فيجوال بيسك هي قدرته على العمل مع البرامج الاخرى المتوافقة مع ويندوز . باستعمال كائن OLE يمكنك تشغيل تطبيقات ويندوز من برنامجك اثناء تشغيله ومعالجة انواع مختلفة من المعلومات ، كما يمكنك تشغيل مكونات فردية من تطبيقات اخرى (مثل مدقق الاملاء في وورد) باستعمال تقنية خاصة تسمى الأتمتة (Automation).

اداة OLE

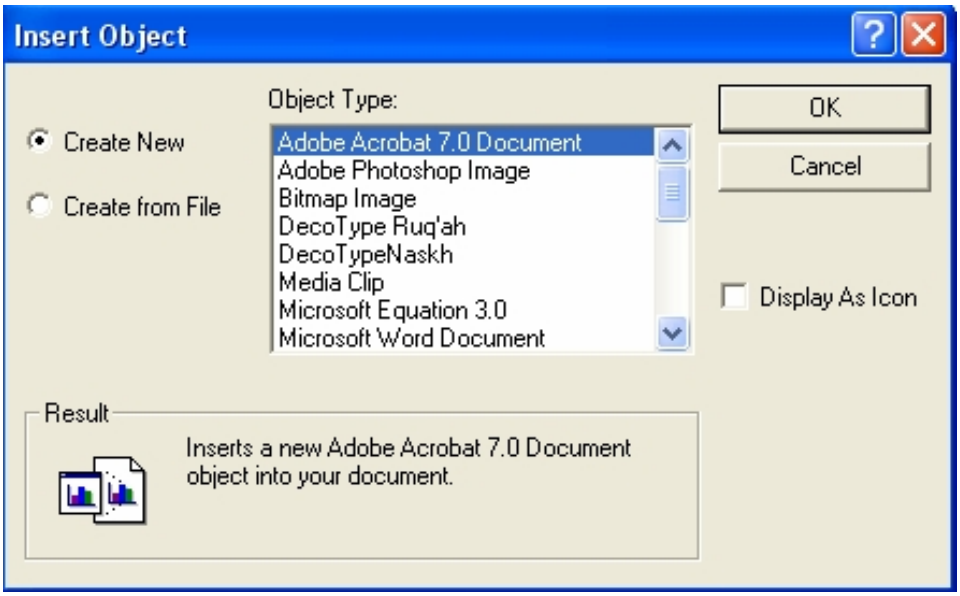
ستتعلم في التمرين التالي كيف يعمل كائن OLE وكيف يمكن استعماله (من دون شفيرة) لانشاء تطبيق يدعى Bid Application ، يشغل مستندات وورد باد وبرنامج الرسام (Paint) . لكي يتمكن المستخدم من استخدام برامج ترتبط بالبرنامج الذي صممته .

انشاء البرنامج

1. اختر الامر New Project < File ثم OK لانشاء ملف تنفيذي قياسي.
2. انشى تسمية تحمل العنوان Bid Application في اعلى الزاوية اليسرى للنموذج انشئ تسمية ثانية تحمل العنوان " تشغيل البرامج وورد باد والرسام من داخل البرنامج" اترك مساحة اضافية في التسمية الاولى اذ سيزيد حجم العنوان عند ضبط الخصائص لاحقا.
3. تحت التسمية الثانية انشئ تسميتان متساوية التباعد على النموذج وتحمل العناوين "برنامج الكتابة وورد باد" و "برنامج الرسم Paint" على التوالي.
   ستستعمل هذه التسميات لمعرفة كائنات OLE المستعملة لتشغيل التطبيقات وورد باد والرسام تباعا . اضف الان كائنات OLE الى النموذج.
4. انقر اداة التحكم OLE في مربع الادوات.
5. تحت التسمية "برنامج الكتابة وورد باد" انشئ مستطيلا بحجم علبة الكبريت تقريبا باستعمال اداة التحكم OLE . حاول ان يبدو برنامجك كما في شكل (5-3)

شكل (3-5)

عندما تفلت زر الماوس ، يعرض فيجوال بيسك مربع الحوار Insert Object الذي يحتوي على لائحة بكل كائنات التطبيقات التي يمكنك استعمالها في برنامجك (قد يستغرق ذلك لحظة او لحظتين لان فيجوال بيسك يحتاج الى تجميع المعلومات من سجل كمبيوترك). تختلف الائحة من كمبيوتر لاخر. ا نظرشكل (3-6)



شكل (3-6) مربع الحوار Insert Object

6. مرر الى اسفل لائحة الكائنات ثم انقر الكائن WordPad Document اذا كان البرنامج موجود في حاسبتك " حيث ان الوورد بات" يكون من ضمن الوندوز اذ لم يكن هذا البرنامج متوفرا، انتق معالج نصوص اخر متوافق مع ويندوز او تطبيقا مشابها في مربع الحوار.

7. انقر مربع الاختيار Display As Icon في مربع الحوار Insert Object لكي يظهر التطبيق كرمز في برنامجك.

اذ لم تنقر مربع الاختيار هذا ، يظهر كائن التطبيق (مستند عادة) ظمن اطار في برنامجك.

8. انقر زر OK لاغلاق مربع الحوار Insert Object ولتشغيل البرنامج وورد باد.

يشتغل وورد باد ويعرض مستندا فارغا ضمن اطار. سيصبح هذا المستند قالبا (Template) في البرنامج Bid Application . يجب ان يحوي على أي معلومات قد تفيد المستخدم خلال استعماله البرنامج.كمعلومات عن شركة البناء والاسماء والعناوين والمواد... الخ.

9. اكتف الان بكتابة Bid Application في الوورد باد ثم استعمل الامر Insert > Date and Time لاضافة التاريخ والوقت الى القالب.

يظهر النص في مستند وورد باد بالشكل الذي سيكون عليه عند تشغيل البرنامج.

10. اختر الامر File > Exit في برنامج وورد باد.

اذا سئلت فيما اذا كنت ترغب في تعديل المستند المصدر انقر Yes . سترى هذا النوع من الاسئلة عند اغلاق كائنات تطبيقات معينة.

بعد الانتهاء من كائن OLE الاول يجب ان يبدو النموذج كما في الشكل (5-3). غير حجم كائن OLE او التسمية اذا كان أي نص مخفيا.

11. كرر الخطوات من 4 الى 10 لاضافة كائن صورة نقطية حيث سيكون اسمها في مربع الحوار Insert Object هو "BitMap Image" تحت التسمية " برنامج الرسم Paint ".

تاكد من نقر الاختيار Display As Icon في مربع الحوار Insert Object واضف بعض معلومات القالب مثلا اضف صورة او شعار الشركة او أي رسم اخر عندما يفتح برنامج الرسام. ثم اغلق برنامج الرسم Paint واكمل بقية الخطوات كما في الخطوات السابقة.

12. ضع زر امر في اسفل النموذج بعد ذلك انقر نقرا مزدوجا الكائن واكتب الجملة End في الاجراء الحدثي Command1_Click

الجملة End تنهي البرنامج عندما ينتهي المستخدم من استعماله.

13. اضبط الخصائص التالية للكائنات الموجودة على النموذج باستعمال اطار الخصائص.

الكائن	الخاصية	الاعداد
Command1	Caption	"Exit"
Label1	Font	الخط Time New Roman ، نمط الخط Bold، حجم الخط 18
Label2	Alignment	1-Right Justify
Label3	Alignment	2-Center
Label4	Alignment	2-Center
OLE1	BackStyle	0-Transpernt
	BoarderStyle	0-None
OLE2	BackStyle	0-Transpernt
	BoarderStyle	0-None
OLE3	BackStyle	0-Transpernt
	BoarderStyle	0-None

14.اختر الامر File > Save Project As واحفظ النموذج على القرص بالاسم MyOleBid .

بعد الانتهاء يجب ان يبدو النموذج MyOleBid كما في الشكل (3-7):

شكل (3-7)

15.حاول ان تقوم بتشغيل البرنامج لترى كيفية عمل الكائن OLE

لتشغيل احد هذين البرنامجين قم بالضغط على ايقونتة نقرا مزدوجا لكي يعمل.

خطوة متقدمة : تثبيت ادوات تحكم اكتيف اكس Active X

يمكنك زيادة وظائفية فيجوال بيسك بتثبيت ادوات تحكم اكتيف اكس المرفقة مع فيجوال بيسك او التي تنشئها بنفسك او التي تشتريها من باعة الادوات البرمجية. للمحافظة على موارد الكمبيوتر ومساحة سطح المكتب الخالية ، يعرض فيجوال بيسك فقط المجموعة الاساسية لادوات تحكم الواجهات في مربع الواجهات في مربع الادوات عندما تفتح مشروعا جديدا. لكن يمكنك تخصيص مربع الادوات لكل مشروع على حدة باستعمال الامر Components < Project . تمتلك ادوات تحكم اكتيف اكس، التي تستفيد من تقنية اكتيف اكس 32 بت (مواصفات مايكرو سوفت للكائنات القابلة للبرمجة والمستعملة في البرامج وانظمة التشغيل وادوات الانترنيت) ملحق اسم الملف ocx. تتم اضافتها الى نظام التشغيل تلقائيا كلما قمت بتثبيت برنامج جديد ("يعلم" فيجوال بيسك بوجود ادوات تحكم اكتيف اكس جديدة ببحثه عن تلك القادمة مع برامج محددة في سجل النظام ويندوز).

تثبيت ادوات تحكم اكتيف اكس (مربع الحوار الشائع)

يتضمن كل اصدار من فيجوال بيسك ادوات تحكم اكتيف اكس اضافية يمكنك استعمالها في مشاريعك(اذا كنت تمتلك النسخة Professional او Enterprise من فيجوال بيسك ، ستكون لديك مجموعة مهمة من ادوات اكتيف اكس لتختار منها).

على سبيل المثال هناك اداة تحكم اكتيف اكس مفيدة ، هي CommonDialog ، موجودة في الملف Comdlg32.ocx لانشاء مربعات حوار قياسية مثل Open و Save As الخ.

نفذ هذه الخطوات لتثبيت ادوات تحكم اكتيف اكس:

1. اختر الامر New Project < File ثم انقر OK لانشاء ملف تنفيذي قياسي.
2. اختر الامر Components < Project ثم انقر علامة التبويب Control .

او عن طريق الضغط على Ctrl + T من لوحة المفاتيح.

يظهر مربع الحوار Components.

تعرض علامة التبويب Control لائحة ابجدية بادوات تحكم اكتيف اكس المتوفرة في كمبيوترك والتي يمكنك اضافتها الى مربع ادوات مشروعك . لاعطائك مرونة في طريقة انشاء البرامج. يتضمن كل مشروع مربع ادوات خاص به يحتوي على ادوات تحكم فيجوال بيسك الافتراضية واي ادوات تحكم اكتيف اكس تنتقيها.

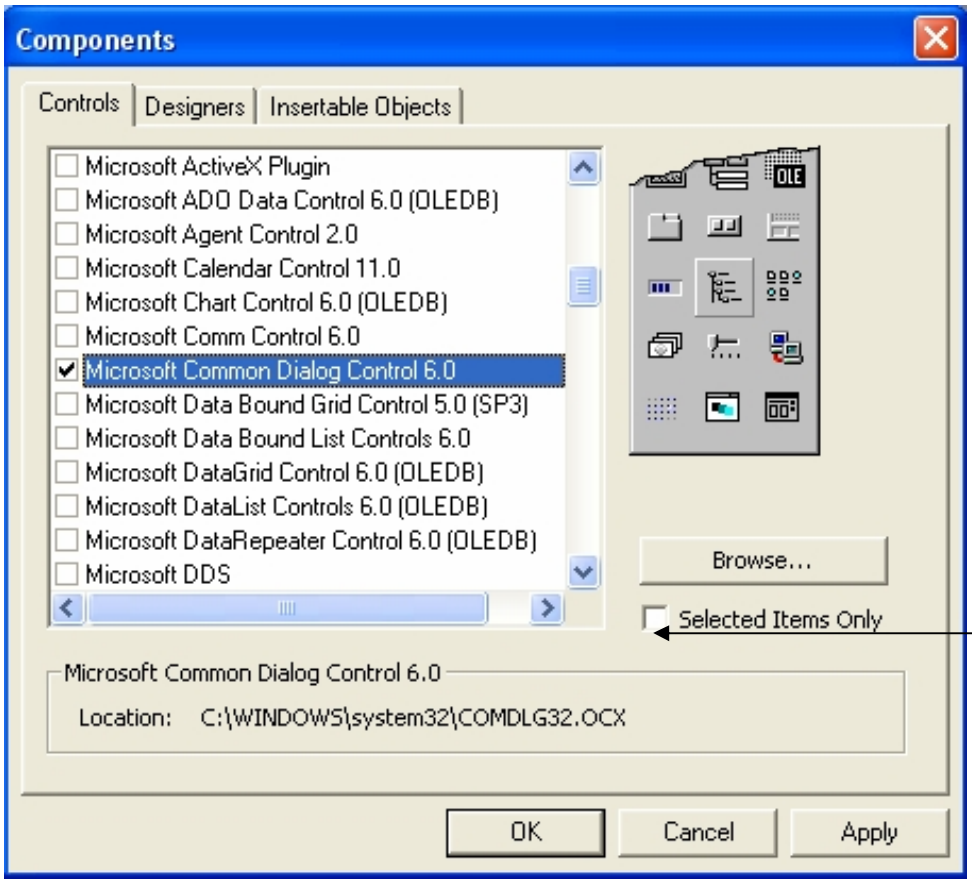
وبالتالي ، أي ادوات تحكم تضيفها الى المشروع الان ستظهر في مربع ادوات هذا المشروع فقط. في الخطوات التالية ستضيف اداة التحكم CommonDialog الى مربع ادواتك.

❖ **تلميح**

يحتوي مربع الحوار **Components** ايضا على علامة تبويب Insertable Object (كائنات قابلة للادراج) يمكنك استعمالها لاضافة كائنات برامج الى مربعات ادوات مشروعك. **الكائن القابل للادراج** هو مكون برمجي يزوده برنامج اخر متوافق مع ويندوز، كمستند وورد او ورقة عمل اكسل. ستجد تلك الادوات مفيدة كادوات تحكم اكتيف اكس تماما.

3. انقر مربع الاختيار المجاور لاداة التحكم المسماة Microsoft Common Dialog Control 6.0 .

يتم انتقاء اداة التحكم اكتيف اكس هذه ويظهر مكان الملف ocx. في اسفل مربع الحوار. كما مبين في شكل (3-8)



مكان الملف ocx.

شكل (3-8) مربع حوار Components

4. انقر OK لاضافة ادوات تحكم اكتيف اكس المنتقاة الى مربع ادوات هذا المشروع.

يعرض مربع الادوات اداة التحكم الجديدة كما مبين في الشكل (3-9).

شكل (3-9)

تعمل اداة تحكم اكتيف اكس هذه تماما كبقية ادوات تحكم فيجوال بيسك في مربع الادوات. في الواقع ، اذا كنت لا تعرف انها ادوات تحكم اكتيف اكس، فمن الصعب تمييزها عن ادوات التحكم الافتراضية. يتم انتقاء ادوات تحكم اكتيف اكس بنقرها وهي تستعمل لانشاء كائنات على النموذج تماما مثلما تستعمل ادوات التحكم الاخرى. تمتلك ادوات تحكم اكتيف اكس اعدادات خصائص قابلة للتعديل ايضا ويمكن استعمالها في الشفيرة كبقية ادوات التحكم الاخرى التي استعملتها في الامثلة السابقة.

مرجع سريع للدرس الثالث:

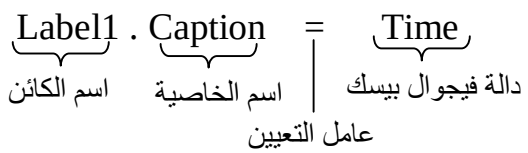
تحميل رسم خلال التشغيل	استدع الدالة LoadPicture وعين النتيجة الى الخاصية Picture التابعة لكائن صورة او كائن مربع رسم. التركيب النحوي لهذه الجملة هو: Object.Picture = LoadPicture (SFile) حيث Object هو اسم الكائن و Sfile هو متغير يحتوي على اسم ملف الرسم مثلا. SelectedFile = "C:\Truck.bmp" Image1.Picture = LoadPicture (SelectedFile)
تشغيل برامج متوافقة مع ويندوز	استعمل كائن OLE لرسم مربع للبرنامج على النموذج ثم انتق كائن البرنامج في مربع الحوار Insert Object لشمله في برنامجك.
تثبيت ادوات تحكم اكتف اكس	اختر الامر Components < Project ثم انقر علامة التبويب Control. انتق ادوات تحكم اكتيف اكس التي تريد اضافتها الى مربع ادوات المشروع ثم انقر الزر OK.

هيكلية جمل فيجوال بيسك البرمجية

كما تعلمت من الدروس السابقة ، يسمى السطر من الشفيرة في برنامج فيجوال بيسك **جملة برمجية** . **الجملة البرمجية** هي أي تركيبة كلمات فيجوال بيسك الدليلية ومن الخصائص والدلالات والعوامل والرموز التي تنشى معا تعليمة صالحة يتعرف عليها مصرف فيجوال بيسك . يمكن ان تكون الجملة البرمجية باكملها مجرد كلمة دليلية بسيطة، مثلا :

Beep

التي تصدر نوتة من مكبر الصوت في كمبيوترك، او قد تكون تركيبة من عناصر ، كالجملة التالية التي تعين الوقت الحالي الى الخاصية Caption التابعة لاحدى التسميات



تسمى قواعد البناء التي يجب استعمالها عند بناء جملة برمجية **التركيب النحوي** يشارك فيجوال بيسك العديد من قواعد تركيبه النحوي مع اصدارات سابقة للغة البرمجية بيسك (Basic) ومع مصرفات لغات اخرى . الخدعة لكتابة جمل برمجية جيدة هي تعلم التركيب النحوي لاكثر عناصر اللغة فائدة ثم استعمال تلك العناصر بشكل صحيح لمعالجة البيانات في برنامجك . لحسن الحظ ان فيجوال بيسك يتولى معظم الاعمال الصعبة عنك ، لذا يكون الوقت الذي تصرفه في كتابة الشفيرة البرمجية قصير نسبيا، وسيكون بالامكان استعمال النتائج مجددا في برامج مستقبلية.

المتغيرات والثوابت

المتغيرات والثوابت هي اساس أي لغة برمجية . إن استيعاب انواع المتغيرات من المسائل الضرورية التي تمكنك من اختيار الانواع المناسبة للمتغيرات سواء لارسالها الى الدوال او لإجراء العمليات الحسابية عليها . بودي التحدث عن مبدئي قابلية الرؤية وعمر الحياة قبل الخوض في تفاصيل المتغيرات.

قابلية الرؤية وعمر الحياة

قابلية الرؤية وعمر الحياة من احد المبادئ الضرورية في جميع لغات البرمجة، و

Visual Basicيعتبر لغة برمجية حقيقة تدعم هذان المبدئان. قابلية الرؤية– Visibility او المدى - Scope للمتغير ، تمثل قدرة البرنامج على الوصول الى المتغير واستخدامه، فالمتغير X الموجود في الكود التالي لا يمكن الوصول اليه خارج الاجراءMySub1:

```
Sub MySub1 ()
Dim X As Integer
X = 20
End Sub

Sub MySub2 ()
Print X  ' لايمثل هذا X السابق
End Sub
```

اما عمر الحياة LifeTime للمتغير، فهي تمثل الفترة التي يظل فيها المتغير محتفظا بقيمته، فالمتغير X الموجود في الكود السابق، سينتهي ويزال تلقائيا من ذا كرة Visual Basicولكي تفهم الاسلوب الذي يتبعه Sub1 . بمجرد الخروج من الاجراء لتطبيق مبدأ قابلية الرؤية وعمر المتغيرات، عليك معرفة انواع المتغيرات من منظور الرؤية وعمر الحياة:

المتغيرات المحلية الديناميكية:

المتغيرات المحلية الديناميكية Dynamic Local Variables هي متغيرات تولد مع السطر الذي تعلن عنها فيه داخل الاجراء وتموت بعد نهاية الاجراء مباشرة ويتم تحرير المساحة التي حجزتها هذه المتغيرات في الذاكرة، وبالنسبة لقابلية الرؤية فلن تستطيع الوصول الى هذه المتغيرات الا في نفس الاجراء الذي صرح فيه المتغير .تستخدم الكلمة المحجوزة Dim لتصريح المتغير مع كتابة اسمه ونوعه:

```
Dim sName As String
Dim iAge As Integer
```

اذا كانت الكلمة المحجوزة Option Explicit موجودة في اعلى منطقة الاعلانات العامة لنافذة النموذج او ملف البرمجة BAS ، فعليك اللالتزام بالتصريح كما في الصيغة السابقة، وان لم تكن الكلمة المحجوزة Option Explicit مسطورة فيمكنك تعريف المتغير مباشرة دون الالتزام بعملية التصريح باسناد قيمة ابتدائية له:

```
sName =  " Ahmed"
```

```
iAge = 99
```

صحيح ان الكود السابق يوفر عليك عناء تصريح المتغير لا انه غير محبذ بشكل كبير لدى المبرمجين الجادين، قد يعرض هذا المثال احد الاسباب:

```
sCompanyName = " Microsoft"
Print sConpanyName
```

الناتج من عملية الطباعة Print في الكود السابق لن يكون كما هو متوقع"Microsoft " ، فالمتغير المستخدم في السطر الثاني هو sConpanyName وليس sCompanyName وهذا الخطأ كفيل في نمو الشوائب البرمجية Bugs في برامجك. سبب آخر قد يجعلك تحبذ الالتزام بعملية التصريح وهو ان جميع المتغيرات تكون من النوع Variant ان لم يتم تصريح نوع غير ذلك، والنوع Variant هو ابطأ انواع المتغيرات كما سيأتي لاحقا.

في مثالنا السابق؛ يؤدي فرض الاعلان عن المتغيرات Option Explicit إلى الاعلان عن خطأ و توقف البرنامج . وفي جميع الحالات فإن الخطأ في كتابة اسم المتغير أو اسناد قيمة إلى متغيرات لم يتم الاعلان عنها مسبقاً سيتتسبب في الإعلان عن خطأ، وسيتوقف البرنامج ايضاً.

ملاحظة :توفر لك بيئة فيجوال بيسك خيار يلزمك بعملية التصريح أي بكتابة الكلمة المحجوزة Option Explicitفي جميع وحدات برامجك كنوافذ النماذج، ملفات البرمجة .... الخ .لتفعيل الاختيار، حدد الاختيار Require Variable Declaration من خانة التبويبEditor في صندوق الحوارOptions.

اخيرا، القيمة الابتدائية للمتغير العددي المصرح هي 0 ، والحرفي يكون قيمة حرفية خالية" " ، اما الكائنات فهي لا شئ Nothing.

المتغيرات المحلية الستاتيكية:

قابلية الرؤية للمتغيرات المحلية الستاتيكية Static Local Variables هي مثل قابلية الرؤية للمتغيرات المحلية الديناميكية أي لن تتمكن من الوصول اليها الا من داخل الاجراء المصرح عنها فيه، وبالنسبة لعمر حياة المتغير الاستاتيكي فهو يبقى محتفظا بقيمته حتى نهاية البرنامج اذا كان في ملف برمجة BAS او حتى يموت الكائن التابع له . لتصريح متغير ستاتيكي استخدم الكلمة المحجوزة Static عوضا عنDim:

```
Static x As Boolean
```

تستطيع جعل جميع المتغيرات التابعة للاجراء ستاتيكية بوضع نفس الكلمة

المحجوزة عند بداية الاجراء:

```
Static Sub Counter ()
جميع المتغيرات التالية ستاتيكية'
Dim iCounter As Integer
Dim iCounter2 As Integer
…
End Sub
```

لا تحاول تطبيق الكود السابق كثيرا، فالمتغيرات الستاتيكية ابطأ من المتغيرات الديناميكية الى جانب قيامها بحجز مواقع هذه المتغيرات في الذاكرة طوال فترة عمل البرنامج، فلا تحاول استخدامها الا عند الحاجة كالرغبة في تنفيذ اجراء معين لمرة واحدة مثلا او الاحتفاظ بقيمة المتغير في عداد:

```
Sub PrintData ()
Static x As Boolean
If x Then
Exit Sub
Else
x = True
End If
…
End Sub

Sub Counter ()
Static i As Integer
i = i + 1
End Sub
```

اخيرا، الكلمة المحجوزة Static لا تطبق الا على المتغيرات المحلية فلا تحاول استخدامها على متغيرات عامة او على مستوى الوحدة فهي بطبيعتها ستاتيكية.

المتغيرات على مستوى الوحدة:

نقصد بالوحدة هي الوحدة البرمجية المتمثلة في ملف Module او ملف برمجة BAS او نافذة نموذج Form او فئة .... Class الخ من الوحدات المكونة للمشروع . يمكنك تصريح متغير على مستوى الوحدة في منطقة الاعلانات العامة للوحدة أي خارج الاجراءات.

قابلية الرؤية لهذا النوع من المتغيرات يكون عام لجميع اكواد الوحدة في حالة استخدام الكلمة المحجوزة Dim اوPrivate:

```
Dim sName As String
Dim iAge As Integer
Sub SetData ( )
sName = "Ahmed"
iAge = 99
End Sub
Sub PrintData ( )
Print sName
Print iAge
End Sub
```

اما اذا كنت تريد تعريف متغيرات عامة قابلة للوصول من جميع انحاء المشروع ،

فالكلمة المحجوزة Public تفي بالغرض:

```
في ملف برمجةBAS ‘
Public No_Of_User As Integer
في نافذة نموذجForm1 ‘
Public sCurrentUser As String
في نافذة النموذجForm2 ‘
Private Sub Form_Load()
If  No_Of_User <= 0 Then
Exit Sub
Else
Me.Caption = Form1.sCurrentUser
End If
End Sub
```

ملاحظة :بالنسبة للكلمة المحجوزة Global فهي مازالت موجودة لضمان التوافقية مع الاصدارات القديمة ل Visual Basic ، وهي تؤدي نفس غرض الكلمة المحجوزة Public ، ولكنك لن تستطيع استخدامها الا في ملفات البرمجة BAS فقط.

اما عمر الحياة لهذا النوع من المتغيرات فيكون مرافق لعمر حياة الكائن التابع له والمصرح فيه –كعمر حياة المتغيرات الستاتيكية، وبالنسبة للمتغيرات العامة المصرحة في ملفات البرمجة BAS، فستظل محتفظة بقيمتها حتى نهاية تنفيذ البرنامج.

المتغيرات

نستطيع ان نعرف المتغيرات بمنظورين، بالمنظور الرياضي يعرف المتغير على انه مجهول س يحتوي على قيمة معينة، اما بالمنظور البرمجي– وهو الاهم – يعرف المتغير على انه قيمة تحفظ في ذاكرة الجهاز ويمكن ان تتغير قيمته خلال تنفيذ البرنامج اكثر من مرة . وتختلف المساحة المحجوزة لحفظ هذه القيمة باختلاف نوع المتغير، فمتغيرمن النوع Byte لا يستهلك

سوى بايت واحد من ذاكرة الحاسب، في حين أن متغير من نوع String قد يحجز مساحة تصل الى 2 جيجابايت. وفيما يلي عرض لجميع انواع المتغيرات المدعومة من قبلVisual Basic:

المتغيرات من النوعByte:

يستطيع هذا النوع الاحتفاظ باي قيمة صحيحة ضمن المجال العددي (0 و 255) وهو اصغر انواع المتغيرات اذ لا يحجز سوى 1 بايت . بداية المتغيرات من نوع Byte كانت منذ الاصدار VB4 وكانت معظم استخداماتها في نسخة 16bit من الاصدار VB4 ، اذ كانت المصفوفة من النوع Byte تستخدم كثيرا عند الاتصال باجراءات API التي تتعامل مع الحروف، اما مع الاصدارات الاحدث فلن تتمكن من الاستفادة وتطبيق الطرق القديمة على المتغيرات من النوع Byte ، لان الترميز المتبعUNICODE يستهلك مساحة 2 بايت للحرف الواحد وليس 1 بايت كترميز ASCII , باختصار لا تضع في ذهنك أي قضايا حرفية Stringsعند استخدامك للمتغيرات من النوعByte خاصة عند الغوص في اعماق اجراءات API، فيمكن قصر استخدامك لها على الاعداد الصغيرة او البيانات الثنائية مع المتغيرات من نوعByte.

المتغيرات من النوعInteger:

اسند أي قيمة عددية صحيحة في المجال (32,767 - 32,768) للمتغيرات من النوع Integer فهي تحجز مساحة 2 بايت .وعند الحديث عن اجراءات API الخاصة بالحروف، فالمصفوفة من النوع Integer هي الانسب للترميز . UNICODE بعيدا عن اجراءات API الحرفية، تفيدك المتغيرات من هذا النوع عند التعامل مع الاعداد الصحيحة، الا انني احبذ استخدام المتغيرات من النوع Long لقدرتها على احتواء قيم اكبر بكثير من المتغيرات من النوع Integer، كما انها النوع القياسي لاغلب اجراءات . API اما في حالة المصفوفات الكبيرة، فانني افضل استخدام المتغيرات من النوع Integer لتوفير % 50 من مساحة الذاكرة.

المتغيرات من النوعLong:

المتغيرات من نوع Long تستطيع حمل قيم عددية صحيحة في المجال (2,147,483,648- ، 2,147,483,647) فهي تحجز مساحة قدرها 4 بايت للمتغير الواحد، وكما ذكرت في الفقرة السابقة اني احبذ استخدامها عوضا عن المتغيرات من النوع Integer ، فهي تحمل قيم كبيرة جدا مقللة الخوف من ظهور خطأ وقت التنفيذ Overflow ، فلو كتبت كود يقرأ حجم ملف معين وكنت من المدمنين للنوع Integer، فستصاب بخيبة امل كبيرة عندما تتعامل مع الملفات التي تزيد احجامها عن: 32,767

```
Dim iFileSize As Integer
```

سيظهر خطأ اذا زاد حجم الملف عن 32,676 بايت‘

iFileSize = FileLen (“C:\MyFile.DAT”)

المتغيرات من النوعBoolean:

المتغيرات من النوع Boolean هي نفس المتغيرات من النوع Integer ولكن القيم التي تمكنك من اسنادها اليها تكون اما 0 False او 1 True ، حجم المتغيرات من النوع Boolean مثل حجم المتغيرات من النواع Integer أي 2 بايت، الا انها لا تستخدم سوى 1 بت متجاهلة ال 15 بت الاخرى . صحيح ان الحجم 2 بايت يعتبر زيادة غير مستخدمة، الا ان المتغيرات من النوع Booleanتسهل عليك عملية قراءة وفهم الاكواد.

المتغيرات من النوعSingle:

مجال القيم التي يمكن للمتغيرات من النوع Single احتوائها هو الاعداد الموجبة من (1.401298e-45 الى 3.402823e38) او الاعداد السالبة من (3.402823e38- الى 1.401298e-45-) وتستهلك مساحة 4 بايت.

ربما يفضل معظم مبرمجي Visual Basic النوع Single على النوع Double لاعتقادهم ان الأول اسرع في التنفيذ من الثاني، هذا الاعتقاد صحيح في النوع القديم من المعالجات والتي لا تحتوي على مساعد رياضي ،Math Coprocessor اما اغلب المعالجات الجديد تحتوي على المساعد الرياضي وهو خاص بالعمليات الحسابية للاعداد ذات الفاصلة العائمة Floating Pointمما يجعل السرعة متقاربة جدا بين النوعين Single و Double ، لذلك ينصح باستخدام النو ع Double عوضا عن النوع Single حتى تتقي شر الخطأ OverFlow ودقة اعلى للاعداد لكبر مجال القيم الممكنة بها . من ناحية اخرى، قد تكون المتغيرات من النوع Single اسرع بكثير من المتغيرات من النوع Double عند التعامل مع الخصائص او الطرق التي تحتك مع الاحداثيات بشكل ملحوظ كـ CurrentX ، Circle، Line، ScaleWidth،ScaleHeight ....الخ فهذه الاحداثيات تستخدم النوع Single ، واستخدام النوع Double معها ابطأ لان Visual Basicيضطر الى تحويل متغيرات النوع السابق الىSingle.

المتغيرات من النوعDouble:

مجال القيم التي يمكن للمتغيرات من النوع Double احتوائها هو الاعداد الموجبة من (4.9406564581247e-324 الى 1.79769313486232e308)او الاعداد السالبة من (4.9406564581247e-324- الى 1.79769313486232e308-)وتستهلك مساحة 8 بايت. معظم دوال Visual Basic الخاصة بالاعداد تعود بقيمة من النوع Double لذلك هو

النوع المفضل دائما، الا ان عيبه الوحيد هو في المساحة الكبير التي يحتجزها، وقد يظهر هذا العيب جليا في المصفوفات الكبيرة من النوعDouble.

المتغيرات من النوعCurrency:

يمكن للمتغيرات من النوع Currency الاحتفاظ بقيم عشرية للفاصلة الثابتة FixedPoint شريطة ان تكون محصورة في داخل المجال (922,337,203,685,477.5808 - ، 922,337,203,685,477.5808) وحجمها 8 بايت ايضا . يوفر هذا النوع من المتغيرات عناء التقريب باستخدام دوال التقريب كـ Round ، .... Fixالخ والتي تستخدمها بكثرة مع المتغيرات من النوع Double و Single مما يبطئ العمليات الحسابية، مع ذلك الاستخدام المجرد للمتغيرات من النوع Currency ابطأ خمس او اربع مرات من المتغيرات Double و Singleفلا تستخدمها بكثرة في حالة تطبيق آلاف العمليات الحسابية عليها.

المتغيرات من النوعDate:

هذا النوع من المتغيرات يحمل قيم تاريخية تبدأ من التاريخ 1 ينا ير 100 الى31 ديسمبر 9999 ويشمل نفس المتغير وقت يبدأ من الساعة 00:00:00 ص حتى الساعة 23:59:59 م وتستهلك مساحة 8 بايت، وفي حقيقة الامر المتغيرات من النوع Date هي نفس المتغيرات من النوع Double، فالجزء العشري يمثل وقت معين والجزء الصحيح يمثل تاريخ معين، فالقيمة 37257.5تمثل الساعة الثانية عشر ظهرا من يوم 1 يناير عام . 2002 السبب الذي جعلني اذكر تفاصيل المتغيرات من هذا النوع هو اعطائك افكار مرنة تمكنك من اجراء عمليات كثيرة على قيم التاريخ وهذه بعض الامثلة:

```
Dim dDateVar As Date
dDateVar = Now
اطبع التاريخ فقط‘
Print Int(dDateVar)
اطبع الوقت فقط‘
Print CDate(dDateVar - Int(dDateVar))
اضف اسبوع واحد‘
Print dDateVar + 7
احذف 30 يوم‘
Print dDateVar - 30
احذف 6 ساعات‘
Print dDateVar - 0.25
```

ملاحظة :ان لم تكن دقيق في كتابة الاعداد المناسبة– وخصوصا العشر ية، فان نتائج العمليات السابقة لن تكون متوقعة.

المتغيرات من النوعString:

لماذا لغة ال BASIC سهلة؟ والجواب بسبب المتغيرات الحرفية من نوع ! String اذا كنت من مبرمجي C فانسى كل شئ متعلق بقضية حجز المساحة في الذاكرة سواء كان ديناميكيا او ستاتيكيا باستخدام المصفوفات، او التحقق من طول النص وغيرها من الامور التي تتطلب 3 او 6سطور لاسناد قيمة الى متغير حرفي، ف Visual Basicهو المتكفل بهذه الامور تلقائيا بمجرد تصريح متغير من النوعString او اسناد قيم حرفية له.

منذ الاصدار– VB4 نسخة عيار 32 بت - اصبحت المتغيرات الحرفية Strings تعتمد ترميز UNICODEوليس . ASCII بصفة عامة، يوجد نوعان من انواع المتغيرات الحرفية يوفرهما Visual Basicلك هما المتغيرات ثابتة الطولFixed-length والمتغيرة الطول-Variable. Length

المتغيرات ثابتة الطول هي متغيرات حرفية عدد حروفها محدد في اثناء تصريحها ولا يمكن ان يتغير:

```
Dim FixedStr As String * 12
sFixedStr =
```

فالعدد الاقصى من الحروف التي يمكن للمتغير FixedStr ان يحمله هو 12 مما يؤدي الى استهلاك مساحة قدرها 24 بايت– لا تنسى ان UNICODE يستهلك2 بايت للحرف الواحد .

من عيوب المتغيرات ثابتة الطول هو عدم توافقها مع تقينة COM ومعظم دوال مكتبات VB و VBAالداخلية واجراءات API لا تدعم هذا النوع من المتغيرات، وحتى لو كان عدد حروف القيمة المسندة اقل من عدد الحروف المصرحة، فان المسافات " " ستحل محل الخانات الفارغة، ولا يمكن لهذا النوع من المتغيرات ان تكون مرئية على مستوى الوحدة من النوع Public ، كما لا يمكنه حمل عدد من الحروف اكبر من 64 كيلوبايت، الا ان الميزة التي تظهر بها عند اسناد القيم الحرفية لهذه المتغيرات فنتائجها تكون دائما اسرع من المتغيرات من النوع المتغيرة الطول، وذلك لان Visual Basic لا يقوم باي عمليات احتجاز في الذاكرة والتحقق من المساحة المتوفرة .... الخ.

ملاحظة COM :او برمجة الكائنات المكونة Component Object Model من التقنيات المبنية على OLE والتي تمكن تطبيقات Windows من الاتصال وتبادل البيانات فيما بينها.

بالنسبة للمتغيرات المتغيرة الطول Variable-Length فهي باختصار تغطي على جميع عيوب النوع السابق ،الا انها تحتجز مساحة تعادل ضعف عدد الحروف + 10 بايتات إضافية تحوي

معلومات عن المتغير الحرفي كحجمه وغيرها من التفاصيل التي يخفيها Visual Basic عنك، والعدد الاقصى من الحروف التي يمكن حفظها في هذا النوع يصل إلى 2 جيجا بايت.

المتغيرات من النوعObject:

معظم المتغيرات التي تمثل كائنات سواء صرحت بالنوع Object او بنوع فئات هي متغيرات من النوعObject:

```
Dim X As Object
Dim Y As Form
Dim Z As Text
```

المتغيرات من النوعVariant:

ظهرت المتغيرات من النوع Variant في الاصدار VB3 وتعدلت بنيته التحتية منذ الاصدار VB4حتى تتوافق مع تقنية COM ، ويستطيع حمل جميع انواع البيانات السابق ذكرها مثل : Long، Date، .... Stringالخ.

الحجم الذي يستهلكه المتغير Variant هو 16 بايت.

الميزة التي تتميز بها المتغيرات من نوع Variant ليس فقط في امكانية اشتمالها على انواع مختلفة من البيانات بل واجراء العمليات الحسابية او المنطقية عليها، حيث يقوم Visual Basic باختبار نوع المتغيرات ومن ثم اجراء العملية الحسابية او المنطقية المناسبة لها:

```
Dim X As Variant
Dim Y As Variant
Dim Z As Variant
X = 2000 ‘ Integerقيمة من النوع
Y = CLng(2000) ‘ Longقيمة من النوع
Z = X + Y ‘ Longقيمة من النوع
X = CDbl(2.5) ‘ Doubleقيمة من النوع
Z = X + Y ‘ Doubleقيمة من النوع
```

لا تحاول الاعتماد على الطرق السابقة بشكل استثنائي، فقد تعطيك نتائج غير متوقعة، فمثلا استخدام معامل الجمع + مع متغيرين من النوع Variant يؤدي الى جمعهما اذا كانت قيم عددية، اما الحرفية فتتم عملية الدمج بينهما كاستخدام معامل الدمج& ، واذا كان احد المتغيرين حرفي والاخر عددي فسيقوم VisualBasic بمحاولة تحويل القيمة الحرفية الى عددية، وان لم يستطع فرسالة الخطأ Type Mismatch سيكون لها نصيب في الظهور:

```
Dim X As Variant
Dim Y As Variant
Dim Z As Variant
X = 20
Y = "20"
Z = X + Y ‘ Z = 40
X = "20"
Z = X + Y ‘ Z = “2020”
Print Z
X = 20
Y = "abcd"
Z = X + Y ' خطأ رسالة
```

المتغيرات من النوع Variant ابطأ انواع المتغيرات، فلا تحاول الاعتماد عليها الا عند الضرورة القصوى .

اذا لم تسند أي قيمة للمتغير Variant فان القيمة الابتدائية له هي Empty والتي تستطيع اختبارها بالدالةIsEmpty:

```
Dim X As Variant
Print IsEmpty(X) ‘ True
X = "20"
Print IsEmpty(X) ‘ False
X = Empty
Print IsEmpty(X) ‘ True
```

والمتغيرات من النوع Variant يمكن لها ان تحتوي كائنات Objects ، لكن لا تنسى استخدام الكلمة المحجوزة Set عند اسناد قيمة كائن الى متغير، واذا اردت الاستعلام عن نوع المتغير، فالدالة IsObject تفي بالغرض المطلوب:

```
Dim X As Variant
Set X = Text1
Print IsObject(X)  ‘ True
Text1.Text = X.Text ‘ "النص"
```

إصطلاحات تسمية المتغيرات

قد تكون تسمية المتغيرات صعبة احيانا لانك تحتاج الى اسماء قصيرة يسهل تذكرها . لتجنب الالتباس ، استعمل الاصطلاحات التالية عند تسمية المتغيرات.

- ابدا اسم كل متغير بحرف . هذه احدى متطلبات فيجوال بيسك. يجب ان تكون اسماء متغيرات فيجوال بيسك اقصر من 256 حرفا ولايمكنها ان تحتوي على نقاط؟!
- اختر اسماء واضحة، مع دمج كلمتين او اكثر في اسم المتغير اذا كان لذلك داع. مثلا SalesTaxRate اوضح بكثير من Rate او Tax .
- اذا شئت استعمل تركيبة من احرف صغيرة وكبيرة ومن ارقام. احد الاصطلاحات المقبولة هو استعمال حرف كبير بداية كل كلمة ، مثلا، DateOfBirth.
- لا تستعمل كلمات فيجوال بيسك الدليلية او اسماء الكائنات او الخصائص كاسماء متغيرات.
- (اختياري) ابدا كل اسم متغير باختصار يتالف من حرفين او ثلاثة احرف يتناسب مع نوع البيانات التي سيتم تخزينها في المتغير. مثلا استعمل الاسم strName لاظهار ان الاسم Name سيحتوي على سلسلة نصية.

الثوابت

ابسط انواع الثوابت هي الثوابت العددية والتي يمكنك كتابتها مباشرة بالنظام العشري Decimalاو باضافة البادئة &H للنظام الستعشري Hexadecimal او البادئة &O للنظام الثماني:

جميع الاعداد التالية تساوي‘ 15

```
Print 15
Print &HF
Print &O17
```

من الضروري ان انبه هنا بان جميع الاعداد المستخدمة في النظام الستعشري Hexadecimal كـ 0 ،1 ، F ، E، 2 ....والنظام الثماني Octal والتي تكتبها في اكوادك تعتبر في نظر Visual Basicاعداد من النوع Integer مالم تضيف الرمز & بعد نهاية العدد فسيكون من النوع

Long، قد تكون جملتي السابقة ليست ذات اهمية كبيرة عند معظم المبرمجين المبتدئين، لذلك علي ان اشد انتباههم بهذا المثال:

```
ستعشري‘
&HF000 = 61440
Print &HF000            ' - 4096
Print &HF000&           ' 61440
ثماني‘
&O170000 = 61440
Print &O170000          ' - 4096
Print &O170000&         ' 61440
```

بعد الثوابت العددية تأتي الثوابت الحرفية Strings ، والتي يشترط كتابتها بين علامتي التنصيص المزدوجة" " ، ولاستخدام علامة التنصيص " في نفس الثابت الحرفي.

فكرة الثوابت المسماة شبيهه بفكرة المتغيرات، ويكمن الفرق بينهما في أن قيم الثوابت لايمكنك تعديلها وقت التنفيذ لانها قيم ليست موجودة بالذاكرة كقيم المتغيرات، وانما يتم استبدال هذه الاسماء بقيمتها الفعلية في الكود اثناء عملية الترجمة Compiling ، فالثوابت تحفظ مباشرة في الملف التنفيذي EXE للبرنامج.

تستطيع تعريف ثابت جديد باستخدام العبارةConst:

```
Const PI = 3.14
Print PI
```

كما يفضل تعريف نوع الثابت لزيادة سرعة التعامل معه:

```
Const PI As Double = 3.14
Const sName = "College"  As String
Const sVALUE As Long = &H32FE&
```

ارجو ان تلتزم بالقيمة المناسبة عند تحديد نوع الثابت، فلا تسند قيمة عشرية لثابت صحيح كالنوع Integer مثلا، لان قيمة الثابت ستتغير ان لم تظهر رسالة الخطأType Mismatch:

```
Const PI As Integer = 3.14     ' 3  ستكون القيمة
Const PI As Integer  =  “abc"      ستظهر رسالة خطأ
```

اخيرا، قابلية الرؤية الافتراضية للثوابت تكون Private على مستوى الاجراء المحلي، او على مستوى نافذة النموذج او الفئة اذا صرح عنها في منطقة الاعلانات العامة، او على مستوى المشروع اذا صرح عنها في ملفات البرمجة . BAS مع تضمين الكلمة المحجوزةPublic:

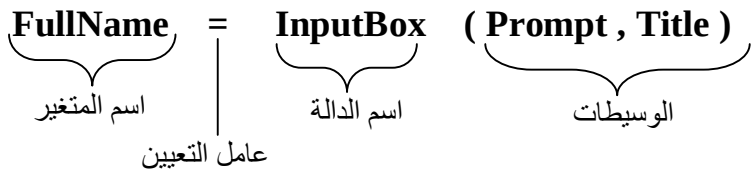
Public Const PI As Double = 3.14

دالتا الادخال والاخرج

تستخدم الدالتين InpuBox و MsgBox للحصول على معلومات من المستخدم واظهارها للمستخدم بواسطة مربعات حوار.

الدالة InputBox

InputBox هي كلمة دليلة خاصة في فيجوال بيسك تعرف بدالة، والدالة هي جملة تؤدي عملا ذا فائدة (كحث المستخدم على ادخال معلومات او احتساب معادلة) ثم تعيد نتيجة الى البرنامج. يمكن تعين القيمة التي اعادتها الدالة الى متغير ما ، او يمكن تعينها الى خاصية او جملة او دالة اخرى. تستعمل دالات فيجوال بيسك في اغلب الاحيان وسيطة واحدة او اكثر لتعريف النشاطات التي تقوم بها ، عندما تستعمل الدالة اكثر من وسيطة واحدة ، يتم الفصل بينها بفواصل وتوضع كلها بين قوسين. تبين الجملة التالية كيفية استدعاء دالة لها وسيطتان.



الدالة MsgBox

MsgBox هي ايضا كلمة دليلة وتعرف بدالة الرسالة، حيث تظهر للمستخدم رسالة تخبره بامر ما كرسالة ترحيب او رسالة انذار او رسالة سؤال لاستفسار منه عن شيء حيث تستخدم الاخيرة في تعيين الجواب الى متغير ،

صيغة كتابة الـ MsgBox

MsgBox (Message , NumberOfButtons, Title)

حيث Message هو النص الذي سيتم عرضه على الشاشة. و NumberOfButtons هو رقم نمط الزر (من 1 الى 5) و Title هو النص المعروض في شريط عنوان مربع الحوار .

في حال اختيار الـ NumberOfButtons وجعله كوسلية حوار مع المستخدم أي تجعل المستخدم يختار احد الامرين فعليك (يستحسن ، وليس الزاما) تعينها الى متغير وذلك لمعرفة ما هو اختيار المستخدم. حيث سنقارن المتغير ونعرف ذلك. كما مبين

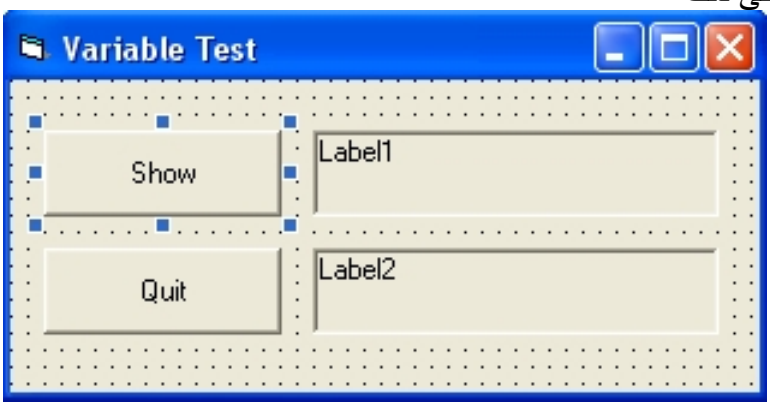
User_Choice = MsgBox (Message , NumberOfButtons, Title)

استعمال المتغيرات في البرنامج

يمكن ان تحافظ المتغيرات على نفس القيمة في كل البرنامج او يمكن ان تتغير القيمة عدة مرات وفقا لحاجتك. يبين التمرين التالي كيف يمكن ان يحتوي متغير يدعى LastName على نص ورقم وكيف يمكن تعينه لخصائص احد الكائنات.

تغيير قيمة متغير

1. شغل فيجوال بيسك
2. اختر الامر New Project < File ثم OK لانشاء ملف تنفيذي قياسي.
3. قم باضافة زري امر (CommandButton) الى النموذج
4. قم باضافة تسميتين (Label) الى النموذج ايضا.
5. حاول ان تجعل نموذجك كما في الشكل (4-1) " واذا اردت ان ترتبه حسب ذوقك فلا ضير في ذلك"



شكل (4-1)

6. اضبط خصائص النموذج وكائناته حسب الجدول التالي:

الكائن	الخاصية	القيمة
Form1	Caption	Variable Test
Command1	Caption	Show
Command2	Caption	Quit
Label1	BoarderStyle	1-Fixed Single
Label2	BoarderStyle	1-Fixed Single

7. انقر زر الامر Quit نقرا مزدوجا

يظهر الاجراء الحدثي Command2_Click في اطار الشفيرة ، اكتب End للخروج من البرنامج. انظر شكل (4-2)

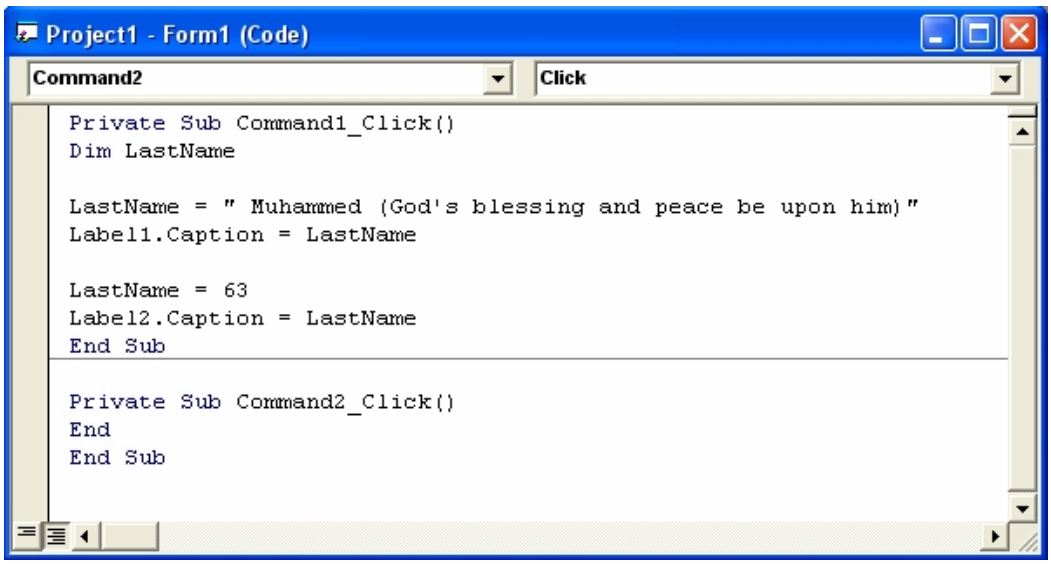
8. انقر زر الامر Show نقرا مزدوجا.

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة.

9. اكتب الجمل التالية لتصريح المتغير LastName واستعماله.

```
Dim LastName
LastName = "Muhammed (God's blessing and peace be upon him) "
Label1.Caption = LastName
LastName = 63
Label2.Caption = LastName
```

الجمل البرمجية مرتبة في ثلاث مجموعات . الجملة الاولى تصرح المتغير LastName باستعمال الجملة Dim . بما ان نوع المتغير غير محدد، سيعتبر انه من النوع متنوع (Variant) " أي انه متغير يمكنه ان يحوي على نص او ارقام". يُعين السطران الثاني والثالث الاسم "Muhammed(God's blessing and peace be upon him" الى المتغير LastName ثم يبينان هذا الاسم في التسمية الاولى على النموذج. يبين هذا المثال احد الاستعمالات الاكثر شيوعا للمتغيرات في برنامج ما – نقل المعلومات الى خاصية. يُعين السطر الرابع الرقم 63 الى المتغير LastName (بمعنى اخر، سيغير محتويات المتغير) . تزيل هذه العملية السلسلة النصية من المتغير وتستبدلها برقم. لم يوضع الرقم بين علامتي اقتباس. تتطلب السلاسل النصية وضعها بين علامات اقتباس اما الارقام فلا (لو انك وضعت علامات اقتباس حول الرقم لاصبح سلسلة نصية ولم يعد من الممكن استعماله في الصيغ الرياضية).يجب ان تبدو الشاشة كما في شكل (2-4).



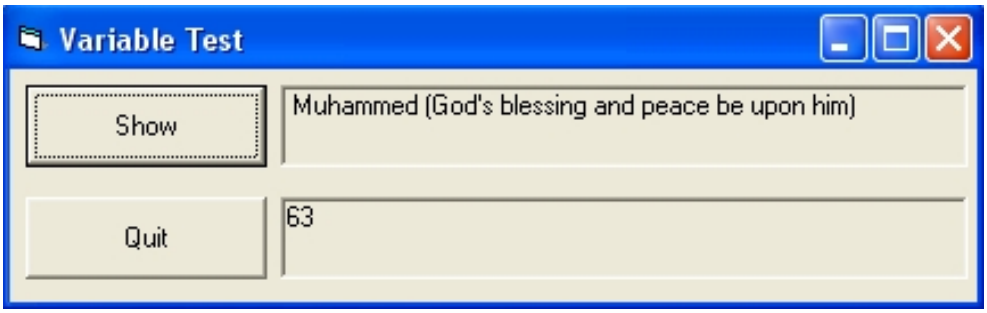
شكل (2-4)

9. انقر زر التشغيل في شريط الادوات، لتشغيل البرنامج.

يشتغل البرنامج في بيئة البرمجة.

10. انقر زر Show

يصرح البرنامج المتغير ويعين قيمتين اليه مرتين، وينسخ كل واحد منها الى التسمية المناسبة على النموذج. يعطي البرنامج الاخراج التالي . شكل (3-4)



شكل (3-4)

استعمال متغير لتخزين الادخال

احد الاستعمالات العملية للمتغير هو حفظ المعلومات التي يدخلها المستخدم . رغم انه يمكنك في اغلب الاحيان استعمال كائن ما (كمربع سرد الملفات او مربع نص) لاستخراج تلك المعلومات، فقد ترغب احيانا بالتعامل مع المستخدم مباشرة وحفظ الادخال في متغير وليس في خاصية. احدى الطرق لفعل هذا هي باستعمال الدالة InputBox لاظهار مربع حوار على الشاشة ثم تخزين النص الذي يكتبه المستخدم في احد المتغيرات. ستجرب هذا الاسلوب في المثال التالي.

الحصول على ادخال باستعمال InputBox

1- شغل فيجوال بيسك
2- اختر الامر New Project < File ثم OK لانشاء ملف تنفيذي قياسي.
3. قم باضافة زري امر (Command) الى النموذج
4. قم باضافة تسمية (Label) الى النموذج ايضا.
5. حاول ان تجعل نموذجك يبدو كما في الشكل (4-4) " واذا اردت ان ترتبه حسب ذوقك فلا ضير في ذلك ايضا"

شكل (4-4)

6. اضبط خصائص النموذج وكائناته حسب الجدول التالي:

الكائن	الخاصية	القيمة
Form1	Caption	InputBox Tester
Command1	Caption	InputBox
Command2	Caption	Quit
Label1	BoarderStyle	1-Fixed Single

7. انقر زر الامر Quit نقرا مزدوجا

يظهر الاجراء الحدثي Command2_Click في اطار الشفيرة ، اكتب End للخروج من البرنامج. انظر شكل (4-5)

8. انقر زر الامر InputBox نقرا مزدوجا.

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة.

9. اكتب الجمل التالية لتصريح متغيرين واستدعاء الدالة InputBox.

```
Dim Prompt, FullName
Prompt = " Please Enter Your Name."
FullName = InputBox ( Prompt)
Label1.Caption = FullName
```

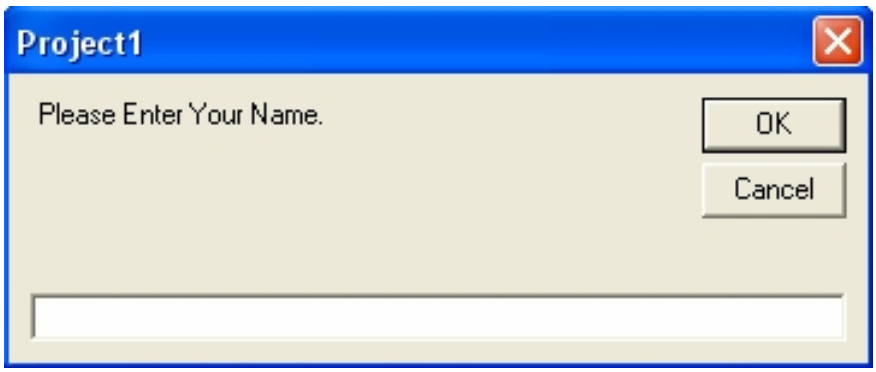
انت الان تصرح متغيرين FullName و Prompt باستعمال الجملة Dim يعين السطر الثاني في الاجراء الحدثي مجموعة من الاحرف، او سلسلة نصية، الى المتغير Prompt . ستستعمل هذه الرسالة كوسيطة نصية للدالة InputBox (**الوسيطة** هي قيمة او تعبير يتم تمريره الى اجراء فرعي او دالة). يستدعي السطر التالي الدالة InputBox ويعين نتيجة الاستدعاء (السلسلة النصية التي يكتبها المستخدم) الى المتغير FullName . InputBox هي دالة خاصة في فيجوال بيسك تعرض مربع

حوار على الشاشة وتطلب من المستخدم كتابة شيء ما. بالاضافة الى سلسلة الطلب، تدعم الدالة InputBox وسيطات اخرى قد ترغب في استخدامها بين حين واخر.

بعد ان تعيد InputBox سلسلة نصية الى البرنامج ، تضع الجملة الرابعة اسم المستخدم في الخاصية Caption للكائن Label1 مما يعرضه على النموذج.

10. انقر زر التشغيل في شريط الادوات لتشغيل البرنامج.
11. انقر زر InputBox

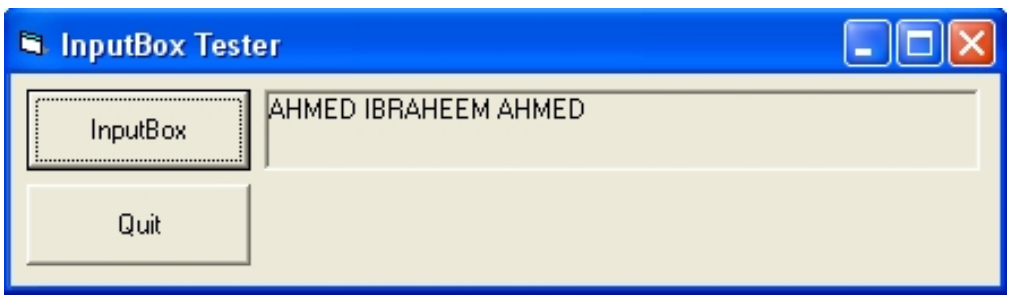
ينفذ فيجوال بيسك الاجراء الحدثي Command1_Click ويظهر مربع الحوار InputBox على الشاشة كما في شكل (4-5)



شكل (4-5)

12.اكتب اسمك بالكامل ثم انقر OK.

تعيد الدالة InputBox اسمك الى البرنامج وتضعه في المتغير FullName. عندها يستعمل البرنامج المتغير لعرض اسمك على النموذج، كما مبين في الشكل (4-6).



شكل (4-6)

استعمل الدالة InputBox في برنامجك كلما اردت حث المستخدم على ادخال معلومات . يمكنك استعمالها مع ادوات تحكم الادخال الاخرى لتنظيم تدفق البيانات من البرنامج واليه.

استعمال متغير للاخراج

يمكنك عرض محتويات المتغير من خلال تعيينه الى خاصية (الخاصية Caption لكائن تسمية مثلا) او بتمريره كوسيطة الى مربع حوار. احدى دالات مربعات الحوار المفيدة لعرض الاخراج هي MsgBox . على غرار InputBox ، تاخذ هذه الدالة وسيطة واحدة او اكثر كادخال لها ويمكن تعيين نتائج استدعاء الدالة الى متغير ما.

تستعمل الدالة MsgBox سلاسل نصية لاظهار الاخراج في مربع حوار. انها تدعم عددا من الوسيطات الاختيارية.

عرض رسالة باستعمال الدالة MsgBox

سنقوم بتحديث البرنامج السابق

1. اذا كنت لا ترى اطار الشفيرة، انقر نقرا مزدوجا الزر InputBox على النموذج. يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة (هذه هي الشفيرة البرمجية التي كتبتها في التمرين السابق).
2. استعمل الماوس لانتقاء الجملة التالية في الاجراء الحدثي (السطر الاخير).

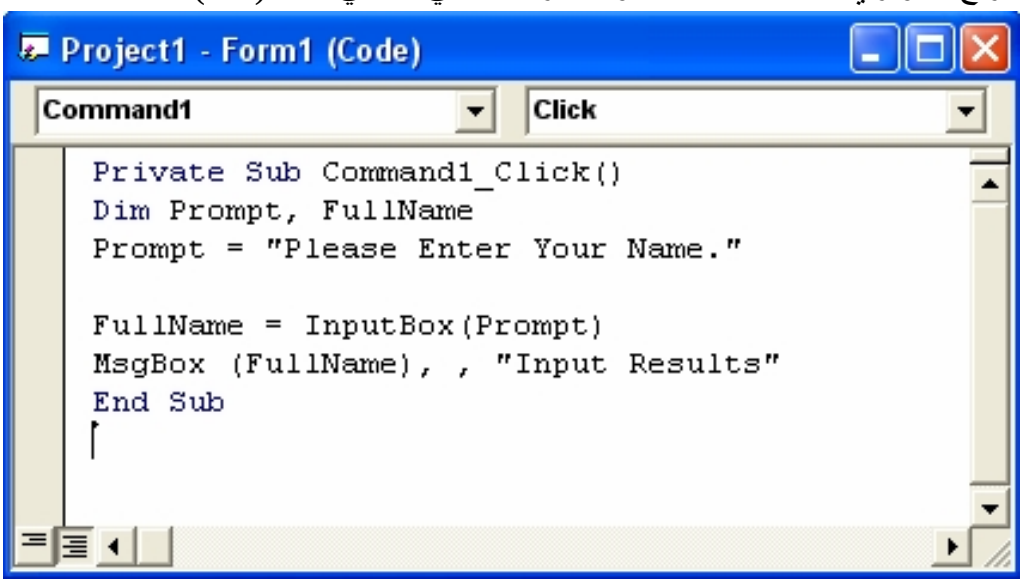
```
Label1.Caption = FullName
```

هذه هي الجملة التي تعرض محتويات المتغير FullName في كائن التسمية.

3. اضغط المفتاح Delete لحذف السطر. تختفي الجملة من اطار الشفيرة.
4. اكتب السطر التالي في الاجراء الحدثي كبديل:

```
MsgBox ( FullName) ,   ,  "Input  Results"
```

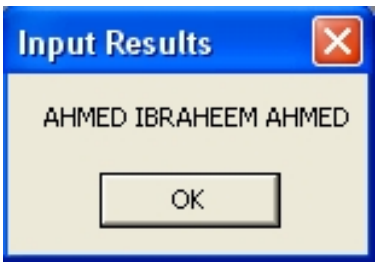
ستستدعي هذه الجملة الجديدة الدالة Msgbox ، وتعرض محتويات المتغير FullName في مربع الحوار ، وتضع الكلمات Input Results في شريط العنوان للمربع الحواري للدالة. يجب ان يبدو الاجراء الحدثي كما في شكل (7-4)



شكل(7-4)

5. انقر زر التشغيل في شريط الادوات.
6. انقر الزر InputBox واكتب اسمك في مربع الحوار ثم انقر OK.

يتم تخزين الادخال في المتغير FullName ثم يظهر في مربع رسالة. يجب ان تبدو الشاشة كما في شكل (4-8) " لكن بدون اسمي، بل باسمك".



شكل (4-8)

7. انقر OK لاغلاق مربع الرسالة ثم انقر Quit لانهاء البرنامج.

المصفوفات

يمكنك Visual Basic من انشاء والتعامل مع المصفوفات Arrays سواء كانت احادية البعد او متعددة الابعاد– قد تصل الى 60 بعدا:

```
Dim OneDim (99) As Intger    ' 100عنصر
Dim TwoDim (4, 9) As Integer     " ثنائية الابعاد
Dim ThreeDim (2, 2, 2) As Integer   "  ثلاثية الابعاد

Dim OneDArray(0 To 10) As String
Dim TwoDArray(0 To 10, 0 To 10) As Long
Dim OneDArray(15 To 22) As String
```

تستطيع البدء في عملية اسناد القيم بمجرد تصريح المصفوفة مع العلم ان فهرس المصفوفة Array Indexيبدأ من صفر مالم تستخدم الكلمة المحجوزة Option Base 1 في منطقة الاعلانات العامة للوحدة البرمجية فانه سيبدأ بواحد:

```
OneDim (0) = 100
OneDim (1) = 200
TwoDim (0, 0) = (100, OneDim (0) + OneDim (1))
```

ملاحظة :رغم ان بدء ترقيم فهرس المصفوفة يمكن ان يبدأ بواحد، الا انني لا احبذ لمبرمجي Visual Basicفعل ذلك، فعند نقل الاكواد بين المشاريع المختلفة او الوحدات البرمجية المختلفة قد لا يتم تفعيل الكلمة المحجوزة Option Base 1 مما يترتب عنه ظهور عشرات الاخطاء البرمجية.

المصفوفات السابقة TwoDim ، OneDim و ThreeDim هي مصفوفات ستاتيكية أي ثابتة الحجم لا تتغير في وقت التنفيذ، لذلك فالمرونة الحقيقة ستكون مع المصفوفات الديناميكية Dynamic Arraysالتي تتيح لك التحكم في حجم المصفوفات كلما دعت الحاجة، وتصريحها يكون بدون ذكر حجمها:

```
Dim DynamicArray ( ) As String
```

قبل ان تبدأ في عملية اسناد القيم، عليك استخدام الكلمة المحجوزة ReDim اولا مع ذكر الحجم:

```
ReDim DynamicArray (2)
DynamicArray (0) = " Ahmed "
DynamicArray (1) = " Muhammed"
DynamicArray (2) = " Ali "
```

لو اردت زيادة او تقليص حجم المصفوفة، استخدم ReDim مرة اخرى وعليك معرفة ان جميع محتويات المصفوفة سوف تلغى:

```
ReDim DynamicArray (4)
DynamicArray (3) = " Hassan"
DynamicArray (4) = " Omer"
Print DynamicArray (4)    '   Omer سوف تطبع
Print DynamicArray (2)    '   لاتطبع شيء
```

واذا رغبت بتغيير حجم المصفوفة دون المخاطرة بفقد البيانات الموجودة فيها ، فالكلمة المحجوزة Preserve جاهزة للاستخدام:

```
ReDim Preserve DynamicArray (4)
DynamicArray (3) = " Noor"
DynamicArray (4) = "Sarah"
Print DynamicArray (4)    '      Sarah تطبع
Print DynamicArray (2)    '      Ali  تطبع
```

الحديث عن Preserve يقودني لاخبارك انك لن تستطيع تغيير ابعاد المصفوفة، فالمصفوفات الديناميكية التالية:

```
Dim OneDim () As Integer
Dim TwoDim () As Integer
```

```
ReDim OneDim (4)
ReDim TwoDim (2, 2)
```

لن تستطيع تغيير ابعادها باستخدام Preserve:

مستحيل‘

```
ReDim Preserve OneDim (3, 3)
ReDim Preserve TwoDim (1)
```

ولكن هذا ممكن‘

```
ReDim OneDim (3, 3)
ReDim TwoDim (1)
```

من المزايا التي اضيفت الى الاصدار VB6 للمصفوفات الديناميكية هي امكانية نسخ قيم مصفوفة كاملة الى اخرى في سطر واحد شريطة ان تكونا من نفس النوع، فبامكانك كتابة شيئا مثل:

```
Dim MyArray (20) As Integer
Dim YourArray ( ) As Integer
MyArray (0) = 10
MyArray (1) = 20
…
YourArray ( ) = MyArray ( )
Print YourArray (0)       ' = 10
```

نقطة اخيرة حول المصفوفات الديناميكة وهي امكانية تدميرها باستخدام العبارة Erase:

```
Erase OneDim
```

النوع:Variant :

المتغيرات من النوع Variant يمكن لها ان تمثل مصفوفات اما عن طريق اسناد مصفوفة لها:

```
Dim Cities(2) As String
Dim vCities As Variant
Cities(0) = "بغداد"
Cities(1) = "موصل"
Cities(2) = "بصرة"
vCities = Cities
Print vCities(1)  ' تطبع الموصل
```

او باستخدام الدالة Array:

```
Dim vCities As Variant
vCities = Array ( "بغداد" , "موصل", "بصرة" )
Print vCities(0)  ' تطبع بغداد
```

التحكم في سير البرنامج

90%من الاجراءات لن تكون ذات قيمة معنوية كبيرة مالم تستخدم عبارات التفرع كـ If و Selectاو الحلقات التكرارية كـ For … Next او Do … Loop لتتحكم في سير البرنامج، الفقرات التالية تشرح عبارات التفرع Branch Statements وعبارات التكرار Looping Statements.

استعمال بنيات القرار

ستتعلم في هذا الجزء كيفية التفرع شرطيا الى اقسام معينة من شفيرة البرنامج وفقا لادخال المستخدم . كما ستتعلم كيفية تقييم خاصية او متغير واحد او اكثر باستعمال التعابير الشرطية ومن ثم ستنفذ جملة برمجية واحدة او اكثر وفقا للنتائج.

استعمال التعابير الشرطية

احد اهم الادوات لمعالجة المعلومات في أي جزء حدثي هو التعبير الشرطي. **التعبير الشرطي** هو جزء من جملة برمجية تطرح سؤال صح او خطا عن خاصية او متغير او جزء من البيانات في شفيرة البرنامج. مثلا، يقييم التعبير الشرطي التالي

```
Price < 100
```

الى صح اذا كان المتغير Price يحتوي على قيمة اقل من 100 والى خطا اذا كان يحتوي على قيمة اكبر من 100 او تساويها. يمكنك استعمال عوامل المقارنة التالية في التعبير الشرطي.

عامل المقارنة	معناه
=	يساوي
< >	لايساوي
>	اكبر من

<	اصغر من
>=	اكبر من او يساوي
<=	اصغر من او يساوي

التفرع باستخدامIF

جملة If لا يستغني عنها اي مبرمج، ليس في Visual Basic وحسب وانما في جميع لغات البرمجة . ومما لا شك فيه تعتبر If من اكثر العبارات استخداما في البرنامج، وهي تنجز اما في سطر واحد او– المفضل -عدة سطور:

في سطر واحد'

```
If X > 0 Then Y = 0
If X > 0 Then Y = 0 Else Y = X
If X > 0 Then X = 0 : Y = 0 Else Y = X
```

في عدة سطور'

```
If X > 0 Then
Y = 0
End If
If M > 0 Then
T = 1
Else
T = -1
End If
If M > 0 Then
T = 1
ElseIf M < 0 Then
T = -1
Else
T = 0
End If
```

اختصار الجمل الشرطية:

اذا تحقق الشرط او اصبحت نتيجة التعبير الشرطي الذي يلي جملة If هو ،True فان اكواد التي تلي عبارة If … Then سيتم تنفيذها:

```
If Value = True Then
End If
If x <> 0 Then
End If
```

تستطيع اختصار الجمل الشرطية، فلو لاحظ ت في جملة If الاولى اننا اختبرنا القيمة المنطقية Valueاذا ما كانت صح او خطأ . المبرمجون المتمرسون يفضلون كتابتها بهذا الشكل:

```
If Value Then
End If
```

الكود السابق صحيح والسبب في ذلك، ان Visual Basic يعتبر اي قيمة غير الصفر هي True اما الصفر فهي False في حالة اختبار الشروط مع جملة . If قد تختصر جملة الشرط الثانية ايضا كما في هذا الكود:

```
If x Then
End If
```

صحيح ان النتيجة مماثلة، لكن حاول تجنب الاختصار في حالة استخدام قيم غير منطقية حتى لا تظهر لك نتائج غير متوقعة . لان ا لقيم المناسبة للاختصارات هي القيم المنطقية فقط وفيما عدى ذلك قد يسبب لك الكثير من الشوائب . راقب الكود التالي:

```
x = 3
y = 4
If x <> 0 And y <> 0 Then      الطريقة الصحيحة
If x And y Then      مشكلة في هذا الاختصار
```

لقد استخدمت اسلوب الاختصار ويبدو ان جملتى الشرط متماثلتين، لكن هل تصدق ان الشرطين السابقين مختلفان تماما ! اذا اردت معرفة السبب قم بالرجوع الى قيمة المتغيرين x و y بنظام الاعداد الثنائي 0011 و 0100 وستعرف السبب ، والذي نستنتج منه باختصار على ان معاملات الربط كـ And و Or تقوم بمقارنة القيم الثنائية للعدد بغض النظر عن نوعه.

ملاحظة :

يمكنك استخدام الامر GoTo مع IF

```
If X > 0 then GoTo  AA:
….
….
End if
….
…
AA:
```

استعمال بنيات القرار

ستتعلم في هذا الدرس كيفية:

- ✓ كتابة تعابير شرطية.
- ✓ استعمال الجملة If … Then للتفرع الى مجموعة من الجمل البرمجية استنادا الى شرط.
- ✓ استعمال زر خيار ومربع اختيار ومربع سرد لمعالجة ادخال المستخدم.
- ✓ استعمال الجملة Select … Case لانتقاء خيار من بين عدة خيارات في الشفيرة البرمجية.

لقد استعملت في الدروس الاخيرة عددا من ادوات فيجوال بيسك لمعالجة ادخال المستخدم. لقد استعملت القوائم والكائنات ومربعات الحوار لعرض الخيارات على المستخدم ، وعالجت الادخال باستعمال الخصائص والمتغيرات. ستتعلم في هذا الدرس كيفية التفرع شرطيا الى اقسام معينة من شفيرة البرنامج وفقا لادخال المستخدم. كما ستتعلم كيفية تقييم خاصية او متغير واحد او اكثر باستعمال التعابير الشرطية ومن ثم ستنفذ جملة برمجية واحدة او اكثر وفقا للنتائج.

البرمجة المسيرة بالاحداث

لقد عرضت البرامج التي كتبتها حتى الان في هذا الكتاب قوائم وكائنات ومربعات حوار على الشاشة، وقد شجعت المستخدمين على التحكم بعناصر الشاشة وفقا للترتيب الذي يناسبهم. لقد وضعت البرامج المستخدم في موضع المسؤلية وانتظرت اجابة منه ، ثم عالجت الادخال بالتوقع. في اوساط البرمجة تعرف هذه المنهجية بالبرمجة المسيرة بالاحداث او الحدثية. تقوم هذه المنهجية على بناء البرنامج من خلال انشاء مجموعة من الكائنات "الذكية" التي تعرف كيف تستجيب عندما يتفاعل المستخدم معها. ثم تعالج الادخال من خلال الاجراءات الحدثية المقترنة بها.

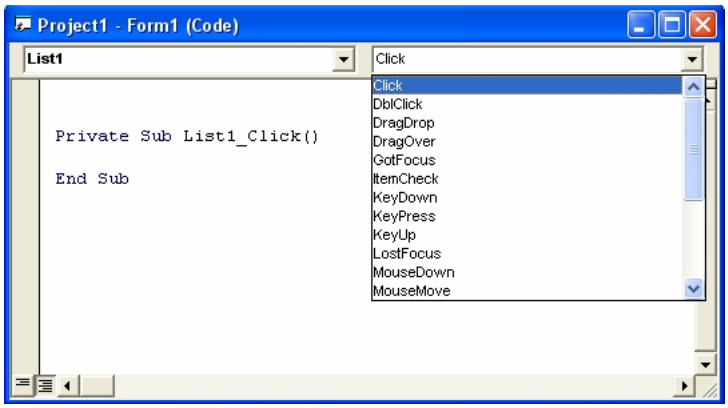
يمكن ان ياتي الادخال من الكمبيوتر نفسه ايضا. فقد يتم مثلا اعلام برنامج عند وصول رسالة بريد الكتروني او بعد مضي فترة زمنية معينة. الكمبيوتر هو الذي يطلق هذه الاحداث وليس المستخدم. بغض النظر عن كيفية اطلاق الحدث، يتجاوب فيجولا بيسك باستدعاء الاجراء الحدثي المقترن بالكائن الذي يناسب ذلك الحدث . مع انك تعاملت بشكل رئيسي مع الحدثين Click و Change ، الا ان كائنات فيجوال بيسك يمكنها ان تستجيب لانواع مختلفة من الاحداث.

تعني طبيعة فيجوال بيسك المسيرة بالاحداث ان معظم البرمجة التي تتم في برنامجك يتم انجازها بواسطة الاجراءات الحدثية. تُعالج كتل الشفيرة المكتوبة في الاجراءات الحدثية الادخال وتحتسب قيما جديدة وتعرض الاخراج، كما انها تتولى مهاما اخرى، ستتعلم في هذا الدرس كيفية استعمال بنيات القرار لمقارنة المتغيرات والخصائص والقيم، وستتعلم كيفية تنفيذ جملة برمجية واحدة او اكثر وفقا للنتائج. في الدرس التالي ، ستستعمل الحلقات لتنفيذ مجموعة من الجمل مرارا وتكرارا لحين تحقق الشرط. سويا، ستساعدك بنيات التحكم بتدفق العمل الفعالة هذه على بناء اجراءات تتجاوب مع كل الحالات تقريبا.

الاحداث التي تدعمها كائنات فيجوال بيسك

لكل كائن في فيجوال بيسك مجموعة من الاحداث المعرفة مسبقا التي يستجيب لها. تلك الاحداث مسرودة في مربع سرد الاجراءات في اطار الشفيرة. يمكنك كتابة اجراء حدثي لاي واحد من هذه الاحداث، واذا حصل احد تلك الاحداث اثناء تشغيل البرنامج، ينفذ فيجوال بيسك الاجراء الحدثي المقترن به. مثلا يدعم كائن مربع السرد "List " الاحداث التالية: DblClickو Click و DragDrop و DragOver و GotFocus و ItemCheck و

KeyDown و KeyPress و KeyUp و LostFocus و MouseDown و MouseMove و MouseUp و OLECompleteDrag و OLEDragDrop و OLEDragOver و OLEGiveFeedback و OLESetData و OLEStartDrag و Scroll و Validate . مع انك لا تحتاج الى برمجة اكثر من حدث او حدثين في برامجك، الا انه من الجيد معرفة توفر كل هذه الخيارات عند انشاء عناصر واجهة البرنامج. يبين الشكل (1-5) سردا جزئيا لاحداث كائن مربع السرد "List" في اطار الشفيرة.



شكل (1-5)

بنيات القرار If … Then

تستطيع التعابير الشرطية ان تتحكم بالترتيب الذي يتم فيه تنفيذ الجمل عند استعمالها ضمن كتلة من الجمل تعرف بـ **بنية قرار** . تتيح لك بنية القرار If … Then تقييم شرط في البرنامج واتخاذ خط عمل معين وفقا للنتيجة. تكتب بنية القرار If … Then في ابسط اشكالها على سطر واحد:

If *condition* Then *statement*

حيث condition هو التعبير الشرطي و statement هي جملة برمجية فيجوال بيسك صالحة ، مثلا ، الجملة

If Score > = 20 Then Label1.Caption = "You Win!"

هي بنية قرار If … Then تستعمل التعبير الشرطي

Score > = 20

لتحديد ما اذا كان على البرنامج ان يضبط الخاصية Caption للكائن Label1 عند "You Win!" ام لا. اذا احتوى المتغير Score على قيمة اكبر او تساوي 20، يضبط فيجوال بيسك الخاصية Caption والا يتخطى جملة الضبط وينفذ السطر التالي من الاجراء

الحدثي. ينتج عن هذا النوع من المقارنات دائما قيمة صح او خطا. لاينتج ابدا عن التعبير الشرطي قيمة "ربما".

اختيار عدد من الشروط في بنية القرار If … Then

يدعم فيجوال بيسك ايضا بنية قرار If … Then تتيح لك شمل عدة تعابير شرطية. يمكن ان تبلغ كتلة الجمل هذه عدة اسطر وتحتوي على الكلمات الدليلية ElseIf و Else و End If.

If *condition1* Then

Statements executed if condition1 is true

نفذ الجمل (الجملة) البرمجية اذا كان الشرط 1 صحيح.

ElseIf *condition2* Then

Statements executed if condition2 is true

نفذ الجمل (الجملة) البرمجية اذا كان الشرط 2 صحيح.

يمكنك اضافة (ElseIf) هنا مع شرط ثالث .

Else

Statements executed if none of the conditions is true

نفذ الجمل (الجملة) البرمجية اذا لم يتحقق أي شرط من الشروط السابقة.

End If

في هذه البنية يتم تقييم condition1 اولا. اذا كان هذا التعبير الشرطي صح ، يتم تنفيذ كتلة الجمل التي تليه، جملة تلو الاخرى (يمكنك وضع جملة برمجية واحدة او اكثر). واذا لم يكن الشرط الاول صح، يتم تقييم التعبير الشرطي الثاني condition2 . اذا كان الشرط الثاني صح، يتم تنفيذ الكتلة الثانية من الجمل. (يمكنك اضافة المزيد من الشروط والجمل ElseIf اذا كانت هناك شروط اكثر يجب تقيمها). اخيرا اذا لم يكن أي من التعابير الشرطية صح ، يتم تنفيذ الجمل التي تلي الكلمة الدليلية Else. اخيرا يتم انهاء البنية باكملها بالكلمات الدليلية End If.

تبين الشفيرة البرمجية التالية كيف يمكن استعمال البنية If … Then المتعددة الاسطر لتحديد مقدار الضريبة المتوجبة في ضريبة تصاعدية افتراضية.

```
If  AdjustedIncome <= 24650 Then   ' 15% tax   %15 ضريبة
    TaxDue = AdjustedIncome * 0.15
ElseIf  AdjustedIncome <= 59750 Then  ' 28% Tax  %28 ضريبة
     TaxDue = 3697 + ( (AdjustedIncome – 24650 ) * 0.28 )
ElseIf  AdjustedIncome <= 124650 Then  ' 31% Tax  %31 ضريبة
     TaxDue = 13525 + ( (AdjustedIncome – 59750 ) * 0.31 )
ElseIf  AdjustedIncome <= 271050 Then  ' 36% Tax  %36 ضريبة
     TaxDue = 33644 + ( (AdjustedIncome – 124650 ) * 0.36 )
Else
     TaxDue = 86348 + ( (AdjustedIncome – 271050 ) * 0.396 )
End If
```

مهم

يعتبر ترتيب التعابير الشرطية في العبارتين If … Then و ElseIf امرا في غاية الاهمية.ماذا لو عكست ترتيب التعابير الشرطية في مثال احتساب الضريبة المبين اعلاه فذكرت المعدلات من الاكبر الى الاصغر ؟ سيوضع مسددو الضرائب من فئات النسب المؤية 15 و 28 و 31 بالمئة في الفئة 36 بالمئة لان دخلهم جميعا سيكون اقل من 271050 او يساويه (يتوقف فيجوال بيسك عند اول تعبير شرطي يكون صح، حتى لو كانت التعابير الاخرى صح ايضا) . بما ان كل التعابير الشرطية في هذا المثال تفحص نفس المتغير، يجب سردها بترتيب تصاعدي لتوزيع مسددي الضرائب في الاماكن المناسبة.

تنبيه: عندما تستعمل اكثر من تعبير شرطي واحد عليك الانتباه الى ترتيب تلك التعابير.

ستستعمل في التمرين التالي بنية قرار If … Then لتتحقق من هوية المستخدمين اثناء تسجيل دخولهم الى البرنامج.

التحقق من هوية المستخدمين باستعمال If … Then

1. شغل فيجوال بيسك وافتح مشروعا exe. قياسيا جديدا.
2. استعمل اداة التحكم زر امر لانشاء زر امر في الزاوية اليسرى العليا للنموذج.
3. اضبط الخاصية Caption لزر الامر عند "Log in"

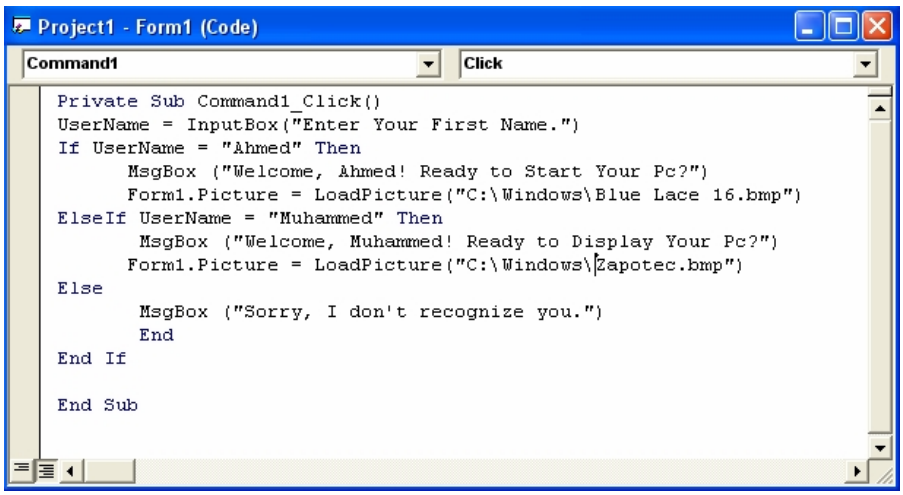
4. انقر الزر Log In نقرا مزدوجا.

يظهر الاجراء الحدثي Command1_click في اطار الشفيرة.

5. اكتب الجمل البرمجية التالية في الاجراء:

```
UserName = InputBox ("Enter Your First Name." )
If UserName = "Ahmed" Then
    MsgBox ( "Welcome, Ahmed! Ready to Start Your Pc?")
    Form1.Picture = LoadPicture ( "C:\Windows\Blue Lace 16.bmp")
ElseIf UserName = "Muhammed" Then
     MsgBox ( "Welcome, Muhammed! Ready to Display Your Pc?")
    Form1.Picture = LoadPicture ( "C:\Windows\ Zapotec.bmp")
Else
     MsgBox ( "Sorry, I don't recognize you.")
     End   ' انهاء البرنامج لعدم معرفة الشخص
End If
```

بعد الانتهاء يجب ان تبدو الشاشة كما في شكل (2-5):



شكل (2-5)

6. انقر زر التشغيل في شريط الادوات.

يظهر البرنامج في بيئة البرمجة ، يظهر نموذج فارغ على الشاشة مع الزر Login في الزاوية اليسرى العليا.

7. انقر الزر Log In.

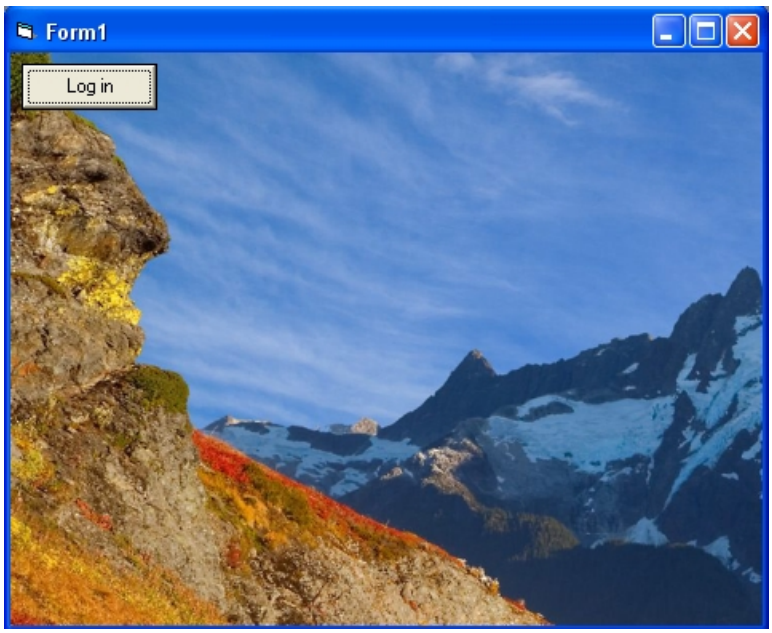
تعرض الدالة InputBox المكتوبة في الاجراء الحدثي Command1_Click مربع حوار يطلب منك ان تكتب اسمك الاول.

8. اكتب Muhammed ثم اضغط المفتاح Enter.

تقارن بنية القرار If … Then الاسم الذي كتبته بالنص "Ahmed" في التعبير الشرطي الاول ، اذا كتبت Ahmed ، سيقيم التعبير الى صح وتعرض الجملة If…. Then رسالة ترحيب باستعمال الدالة MsgBox. واذا لم يكن صح ستقارن الكلمة التي كتبتها مع التعبير الشرطي الثاني ، فاذا كتبت Muhammed ، سيقيم التعبير الى صح وتعرض الجملة If…. Then رسالة ترحيب باستعمال الدالة MsgBox.

9. انقر OK في مربع الرسالة.

ينغلق مربع الرسالة ويتم تحميل ملف صورة من الموقع المحدد، على النموذج كما مبين في الشكل (3-5)



شكل(3-5)

10. انقر الزر Login واكتب Amar ثم انقر OK.

يتم تنفيذ الجملة Else في بنية القرار وتظهر الرسالة التالية في كائن MsgBox.

11. انقر OK لاغلاق مربع الرسالة.

ينغلق مربع الرسالة والبرنامج. لقد منع مستخدم غير مرخص له من استعمال البرنامج.

استعمال العوامل المنطقية في التعابير الشرطية

يتيح لك فيجوال بيسك اختبار اكثر من تعبير شرطي واحد في القسمين If..Then و ElseIf اذا اردت شمل اكثر من معيار انتقاء واحد في بنية القرار. يتم ربط الشروط الاضافية سوية باستعمال واحد او اكثر من العوامل المنطقية التالية:

العامل المنطقي	معناه
And	اذا كان التعبيران الشرطيان صح ، تكون النتيجة صح.
OR	اذا كان احد التعبيرين الشرطيين صح، تكون النتيجة صح.
Not	اذا كان التعبير الشرطي خطا، تكون النتيجة صح. واذا كان التعبير الشرطي صح، تكون النتيجة خطا.
Xor	اذا كان احد التعبيرين الشرطيين فقط صح، تكون النتيجة صح، واذا كان كلاهما صح او خطا، فالنتيجة خطا.

ملاحظة:

عندما يقيم برنامجك تعبيرا معقدا يتضمن انواعا مختلفة من العوامل، فانه يقيم العوامل الرياضية اولا ثم عوامل المقارنة ثم العوامل المنطقية.

اضافة حماية كلمة المرور باستعمال العامل AND.

12. انقر الزر Log In نقرا مزدوجا لفتح الاجراء الحدثي Command1_click في اطار الشفيرة.
13. اكتب الجملة التالية بين الجملة InputBox و الجملة If … Then في الاجراء (بين السطرين الاول والثاني):

```
Pass = InputBox ( "Enter Your Password." )
```

14. عدل الجملة If … Then الى الشكل التالي:

If UserName = "Ahmed" And Pass = "1984" Then

تتضمن الجملة الان العامل المنطقي And الذي يتحقق من اسم المستخدم ومن كلمة المرور قبل قبول Ahmed في البرنامج.

15.عدل الجملة ElseIf الى الشكل التالي:

ElseIf UserName = "Muhammed" And Pass = "1980" Then

يضيف العامل المنطقي And اختبارا لكلمة المرور "1980" التي تخص Muhammed.

16.انقر زر التشغيل في شريط الادوات .

17.انقر الزر Log in واكتب Ahmed ثم انقر OK.

يطالبك البرنامج بكلمة المرور.

18.اكتب 1984 وانقر OK.

يقيم التعبير الشرطي And على انه صح ويتم قبول Ahmed في البرنامج.

19.انقر زر OK لاغلاق مربع الرسالة.

20.انقر زر الانهاء في شريط الادوات لانهاء البرنامج.

فكرة

اذا كنت تكتب اصدارا كامل المزايا لبرنامج المرور بواسطة كلمة سر واسم مستخدم، يجب ان تفكر في استعمال كائن مربع نص لتلقي ادخال كلمة المرور في البرنامج. تدعم كائنات مربعات النص الخاصية PasswordChar التي تتيح لك اظهار حرف نائب ، كنجمة (*) مثلا، بينما يكتب المستخدم ، والخاصية MaxLength التي تتيح لك تحديد العدد الاقصى للاحرف التي يمكن ادخالها.

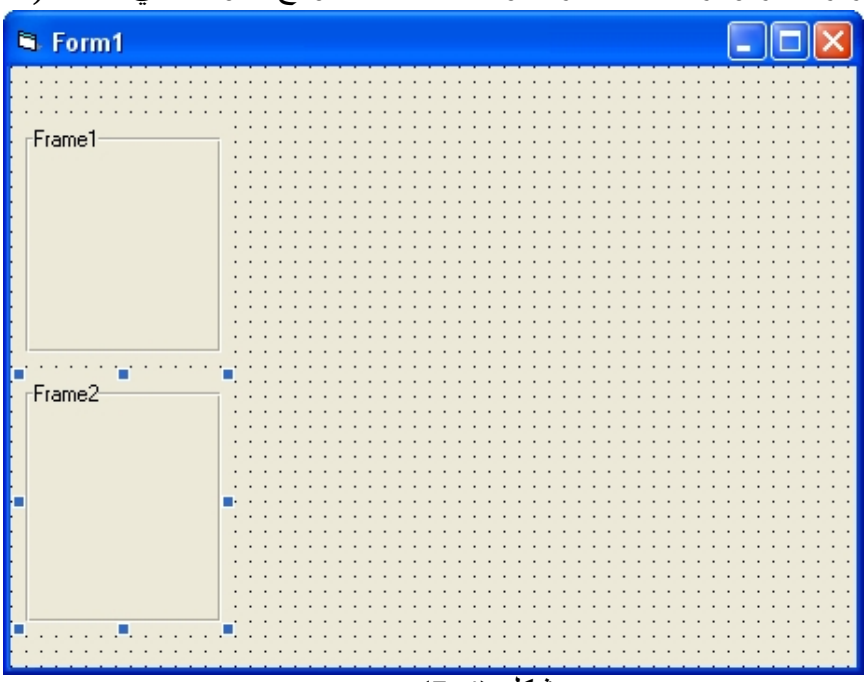
برنامج Shopping

يقوم هذه البرنامج بدور بيئة طلبيات شراء حاسبات حيث انك سوف تحدد نوع الحاسبة والمواصفات ثم طريقة الدفع ثم المبلغ الكلي. خلال عمل البرنامج Shopping راقب كيفية عمل عناصر زر الخيار ومربع الاختيار والقائمة وقائمة منسدلة في البرنامج.

بناء البرنامج Shopping

1. اختر الامر File ثم New Project
   اضغط على OK لانشاء ملف تنفيذي قياسي.
2. قم بادراج الاداة اطر "Frame" من مربع الادوات على النموذج مرتين.
   حاول ان تجعل الواحدة تحت الاخرى وان تحتوي على مساحة لاننا سنضع بداخلها ازرار خيار ومربعات اختيار حاول ان تجعل النموذج يبدو كما في الشكل (4-5)

أطر



شكل (4-5)

تسمية

مربع نص

زر امر

3. قم بادراج تسميتين على النموذج واحدة للعنوان والاخرى للمبلغ الكلي ضع تسمية العنوان فوق الكل وضع تسمية المبلغ الكلي في الاسفل من جهة اليمين
4. قم بادراج مربع نص وضعه فوق تسمية المبلغ الكلي.
5. قم بادراج زر امر وضعه في وسط النموذج من الاسفل
6. قم بادراج ثلاث ازرار اختيار من مربع الادوات وضعها داخل الاطر الاول
7. قم بادراج ثلاث مربعات اختيار وضعها داخل الاطر الثاني.

زر اختيار

مربع اختيار

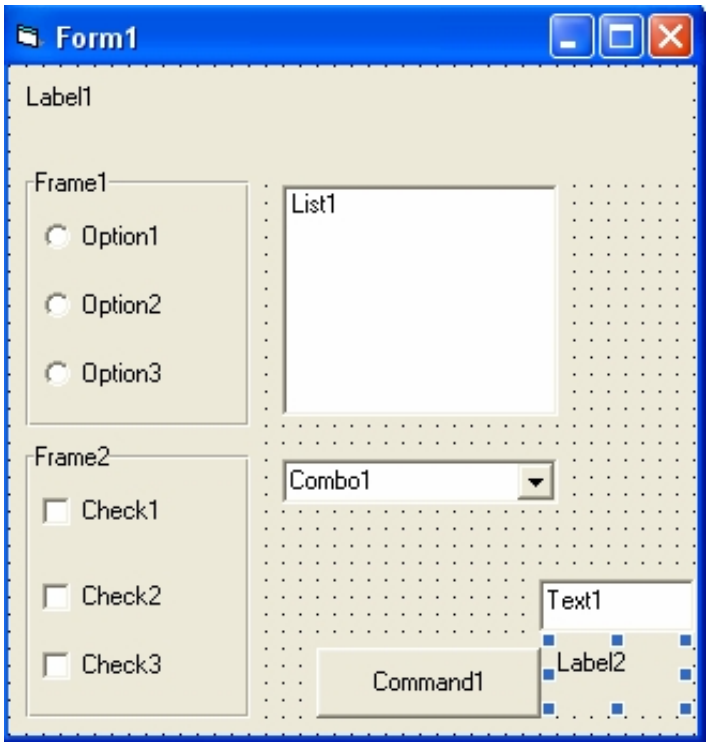
قائمة

8. قم بادراج قائمة من مربع الادوات وضعها على النموذج في الوسط من الاعلى واجعل مساحتها كبيرة

قائمة منسدلة

9. قم بادراج قائمة منسدلة وضعها اسفل القائمة.

حاول ان تجعل برنامجك يبدو كشكل (5-5)



شكل (5-5)

ضبط خصائص الكائنات

1. قم بضبط خصائص الكائنات التي ادرجتها في النموذج كما مبين في الجدول

الكائن	الخاصية	القيمة
Label1	Caption	"Shopping Program"
	Font	Size=24 نقطة
	Alignment	2-Center
Label2	Caption	"المبلغ الكلي"
	BoarderStyle	1-Fixed Single
Command1	Caption	"Exit"
Frame1	Caption	"نوع الحاسبة"
Frame2	Caption	"البرامج المرافقة"
Combo Box	Text	"طريقة الدفع"
	RightToLeft	True

Option1	Caption	"PC (Desktop)"
Option2	Caption	"Macintosh"
Option3	Caption	"Laptop"
CheckBox1	Caption	"Windows XP (10$)"
CheckBox2	Caption	"Microsoft Office 2003 (5$)"
CheckBox3	Caption	"Collection Games (15$)"
Text1	Text	""
	Locked	True

1. قم بإدراج زر أمر ثاني وضعه فوق زر الأمر الأول واضبط خصائصه كالتالي.

الكائن	الخاصية	القيمة
Command2	Caption	"OK"

2. يجب ان يكون برنامجك كما في الشكل (5-6)

شكل (5-6)

كتابة شفيرة البرنامج:

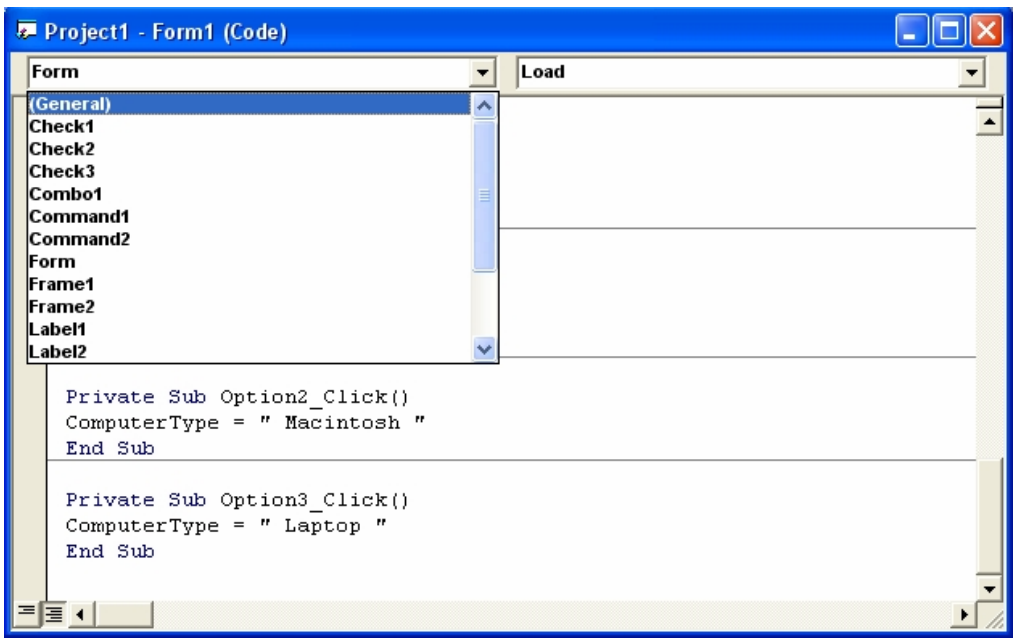
سنقوم الان بكتابة الشفيرة البرمجية لـ Shopping ، سنقوم بعرض ثلاثة انواع من الحاسبات حيث سيقوم المستخدم باختيار حاسبة واحدة من نوع الحاسبات ولكنه يستطيع اختيار كل البرامج من البرامج المرافقة حيث سوف يظهر المبلغ الكلي في مربع النص وفي حالة انه تم البيع سنقوم بالضغط على زر OK حيث سوف نقوم بخزن المعلومات عن الحاسبة المباعة في القائمة.

1. انقر نقرا مزدوجا على النموذج لفتح اطار الشفيرة حيث سيظهر الاجراء الحدثي Form_Load اكتب الجمل البرمجية بين الجملتين Private Sub و End Sub

```
Private Sub Form_Load()
Combo1.AddItem "نقد"
Combo1.AddItem "أجل"
End Sub
```

نقوم في هذا الاجراء(وهو اجراء يتم تنفيذه عند تحميل البرنامج لاننا في الحدث Load) باضافة عناصر الى القائمة المنسدلة ، حيث تستخدم الخاصية AddItem لاضافة عناصر الى القائمة المنسدلة ، هنا سوف نقوم اضافة "نقد" و "أجل" في طريقة الدفع

2. انقر القائمة المنسدلة في اطار الشفيرة لجزء الكائنات واختر General كما مبين في الشكل (7-5)



شكل (7-5)

3. قم بتعريف متغير واسمه ComputerType كمتغير نصي ، اكتب الجملة البرمجية التالية في الـ General:

```
Dim ComputerType As String
```

4. قم باغلاق اطار الشفيرة البرمجية ، ثم انقر نقرا مزدوجا الكائن زر الاختيار "Option1" واكتب الجملة البرمجية بين الجملتين Private Sub و End Sub

سنقوم بوضع نص معين داخل المتغير ComputerType حيث سنستدعيه فيما بعد في

خطوات البرمجة هنا وضعنا النص " PC " وسيتغير نوع النص في كل زر اختيار لان هذا المتغيير يحدد نوع الحاسبة التي سيختارها المستخدم.

```
Private Sub Option1_Click()
ComputerType = " PC"
End Sub
```

5. قم بالنقر نقرا مزدوجا على زر الاختيار الثاني Option2 واكتب الجملة البرمجية التالية بين الجملتين Private Sub و End Sub

```
Private Sub Option2_Click()
ComputerType = " Macintosh "
End Sub
```

6. قم بالنقر نقرا مزدوجا على زر الاختيار الثالث Option3 واكتب الجملة البرمجية التالية بين الجملتين Private Sub و End Sub

```
Private Sub Option3_Click()
ComputerType = " Laptop "
End Sub
```

7. قم بالنقر نقرا مزدوجا على مربع الاختيار الاول Check1 واكتب الجملة البرمجية التالية بين الجملتين Private Sub و End Sub

```
Private Sub Check1_Click()
If Check1.Value = 1 Then
Text1.Text = Val(Text1.Text) + 10
Else
Text1.Text = Val(Text1.Text) - 10
End If
End Sub
```

سنقوم هنا بقراءة حالة مربع الاختيار فاذا كان المستخدم قد اختار Windows Xp فاننا سوف نقوم باضافة 10 دولارات على السعر حيث سنرى ان السعر سوف يظهر في مربع النص ولمعرفة فيما اذا كان المستخدم قد اختار الامر ام لا، فنستطيع معرفة ذلك من خلال قيمة مربع الاختيار فاذا كانت تساوي 1 فذلك يعني انه قد تم اختياره واذا لم يساوي 1 فيعني انه لم يختره او ان المستخدم قد ازال الاختيار فلذلك سوف نقوم بطرح 10 دولارات من المبلغ وايضا نضع الناتج في مربع النص من خلال الجملة

```
Text1.text = Val(Text1.text) -10
```

اما الايعاز Val فهو يستخرج القيمة العددية بداخل مربع النص وهو مختصر للكلمة Value

8. قم بالنقر نقرا مزدوجا على مربع الاختيار الثاني Check2 واكتب الجملة البرمجية التالية بين الجملتين Private Sub و End Sub

```
Private Sub Check2_Click()
If Check2.Value = 1 Then
Text1.Text = Val(Text1.Text) + 5
Else
Text1.Text = Val(Text1.Text) - 5
End If
End Sub
```

9. قم بالنقر نقرا مزدوجا على مربع الاختيار الثالث Check3 واكتب الجملة البرمجية التالية بين الجملتين Private Sub و End Sub

```
Private Sub Check3_Click()
If Check3.Value = 1 Then
Text1.Text = Val(Text1.Text) + 15
Else
Text1.Text = Val(Text1.Text) - 15
End If
End Sub
```

10. انقر ألان زر الأمر Exit نقرا مزدوجا واكتب الجملة البرمجية التالية بين الجملتين Private Sub و End Sub

```
Private Sub Command1_Click()
End
End Sub
```

11. انقر ألان زر الأمر OK نقرا مزدوجا واكتب الجملة البرمجية التالية بين الجملتين Private Sub و End Sub

```
Private Sub Command2_Click()
If Option1.Value = True Then
Text1.Text = Val(Text1.Text) + 400
Option1.Value = False
ElseIf Option2.Value = True Then
Text1.Text = Val(Text1.Text) + 350
Option2.Value = False
ElseIf Option3.Value = True Then
```

```
Text1.Text = Val(Text1.Text) + 600
Option3.Value = False
End If
List1.AddItem ComputerType & "   " & Val(Text1) & "$   " & _
Combo1.List(Combo1.ListIndex)
Check1.Value = 0
Check2.Value = 0
Check3.Value = 0
Text1.Text = ""
End Sub
```

سنقوم هنا بتنفيذ الشفيرة البرمجية الحالية في حالة النقر على زر الامر OK ، حيث في البداية سوف نبحث عن أي زر اختيار قد اختاره المستخدم ولمعرفة الزر الذي قد تم اختياره من قبل المستخدم نختبر الخاصية "Value" لكل زر اختيار فاذا كان يساوي "True" فهذا يعني انه قد تم اختياره لذلك سوف يتحقق الشرط فاذا قد اختار المستخدم زر الاختيار الاول وهو يعني حاسبة من نوع "PC Desktop " فاننا سوف نقوم باضافة سعرها الى مربع النص ثم بعد ذلك نقوم باعطاء الخاصية Value القيمة "False" وذلك نلغي الاختيار السابق حتى يبدا المستخدم بحساب قيمة ثانية.

بعد ذلك سوف نقوم باضافة نوع الحاسبة والمبلغ الكلي مع العلامة $ مع نوعية الدفع في القائمة من خلال الايعاز AddItem حيث ان المتغير ComputerType يحمل نوع الحاسبة والايعاز Val(text1) ياخذ القيمة من داخل مربع النص ويقوم بدمجها مع نوع الحاسبة ثم ندمجها مع الرمز $ من خلال اداة الدمج (&) واخيرا نقوم بدمجهم مع نوعية الدفع ونقوم باضافتهم الى القائمة.

بعد ذلك نقوم بتحرير مربعات الاختيار (أي تصبح كانها لم تضغط) وذلك عند اعطائها القيمة 0 واذا اعطيناها القيمة 1 فذلك يعني انها مختارة. واخيرا نقوم بافراغ مربع النص من أي قيمة لذلك نعطيه القيمة "" للدلالة على انه فارغ **Text1.Text = ""**

تشغيل البرنامج:

1. قم بتشغيل البرنامج عن طريق الضغط على مفتاح F5 من لوحة المفاتيح.
2. بعد تشغيل البرنامج اختر حاسبة معينة ثم اختر البرامج التي تريدها ان تاتي مع الحاسبة و نوعية الدفع ثم قم بالضغط على زر Ok حيث سوف ترى اضافة اسم الحاسبة ونوعية الدفع و السعر في القائمة .
3. احفظ البرنامج باسم معين ثم اغلقه

التفرع باستخدام Select

بامكانك تطبيق مبدأ التفرع باستخدام عبارةSelect:

```
Select Case iDay
Case 1
sDay = "السبت"
Case 2
sDay = "الاحد"
…
Case 7
sDay = "الجمعة"
Case Else
sDay = "يوم غير معرف"
End Select
```

تكمن قوة عبارة Case في امكانية تطبيق المعاملات المنطقية او تحديد مجال للقيم:

```
Select Case iAge
Case Is <= 0
sDesc = "لم يولد"
Case 1 To 11
sDesc = "طفل"
Case 15 To 20
sDesc = "مراهق"
Case 21 To 50
sDesc = "رجل"
Case Is >= 51
sDesc = " عجوز"
End Select
```

المزيد ايضا، يمكنك تحديد مجموعة قيم:

```
Select Case sLetter
Case “A” To “Z”, “a” To “z”
sLetter = "حرف ابجدي"
Case “0” To “9”
sLetter = "رقم"
Case "." , ":" ,  ";"
sLetter = "رمز"
Case Else
sLetter = "غير معرف"
End Select
```

ليس هذا فقط، بل يمكنك تحديد مجموعة جمل شرطية:

Select Case True

Case x > 0, Y < 0

تعادل،

If (X > 0) Or (Y < 0)

End Select

بنية القرار Select Case

يتيح لك فيجوال بيسك التحكم بتنفيذ الجمل في برامجك باستعمال بنيات القرار Select Case. تشبه البنية Select Case البنية If … Then …ElseIf لكنها اكثر فاعلية عندما يعتمد التفرغ على متغير رئيسي واحد، او **حالة اختبار.** يمكنك ايضا استعمال البنية Select Case لتسهيل قراءة البرنامج ولجعله اكثر فاعلية. تركيبها النحوي هو:

Select Case *Variable*

Case *Value1*

Program statement executed if value1 matches variable

ستنفذ جمل البرنامج اذا طابقت Value1 المتغير

Case *Value2*

Program statement executed if value2 matches variable

ستنفذ جمل البرنامج اذا طابقت Value2 المتغير

Case *Value3*

Program statement executed if value3 matches variable

ستنفذ جمل البرنامج اذا طابقت Value3 المتغير

.

.

.

End Select

تبدا البنية Select Case بالكلمتين الدليلتين Select Case وتنتهي بالكلمتين End Select . عليك استبدال Variable بالمتغير او الخاصية او التعبير الذي سيكون في القيمة الدليلية او

حالة الاختبار في البنية. واستبدال Value1 و Value2 و Value3 بالارقام او بالسلاسل او بالقيم الاخرى التي لها صلة بحالة الاختبار التي تتم دراستها. اذا طابقت احدى القيم المتغير ، يتم تنفيذ الجمل الموجودة تحت تعبيرها Case ويتابع فيجوال بيسك تنفيذ شفيرة البرنامج بعد الجملة End Select . يمكنك ادراج أي عدد تريده من تعابير Case في البنية Select Case كما يمكنك اضافة اكثر من قيمة واحدة في كل تعبير Case . اذا ذكرت عدة قيم بعد حالة ما، افصل بينهما بفواصل.

تدعم البنية Select Case ايضا استعمال تعبير Case Else الذي يمكنك استعماله في حالة عدم مطابقة أي من الحالات السابقة.

استعمال عوامل المقارنة مع البنية Select Case

يتيح لك فيجوال بيسك استعمال عوامل المقارنة لادراج نطاق من قيم الاختبار في البنية Select Case . عوامل المقارنة التي يمكن استعمالها هي = ، <> ، < ، > ، =< ، => . لاستعمال عوامل المقارنة عليك كتابة الكلمة الدليلية Is او To في التعبير لتعريف المقارنة التي تقوم بها . ترشد الكلمة الدليلية Is المصرف (Compiler) ليقارن متغير الاختبار بالتعبير الذي يليها. تعرف الكلمة الدليلية To نطاقا من القيم . تستعمل البنية التالية الكلمتين الدليليتين Is و To والعديد من عوامل المقارنة لاختيار المتغير Age ولعرض احدى خمس رسائل:

```
Select Case  Age
Case Is < 13
        Label1.Caption = " Enjoy Your Youth"
Case 13 To 19
        Label1.Caption = " Enjoy Your teens"
Case 20 to 49
        Label1.Caption = " Enjoy Your Time"
Case 50 To 90
        Label1.Caption = " good age"
Case Else
        Label1.Caption = " That a nice age to be"
End Select
```

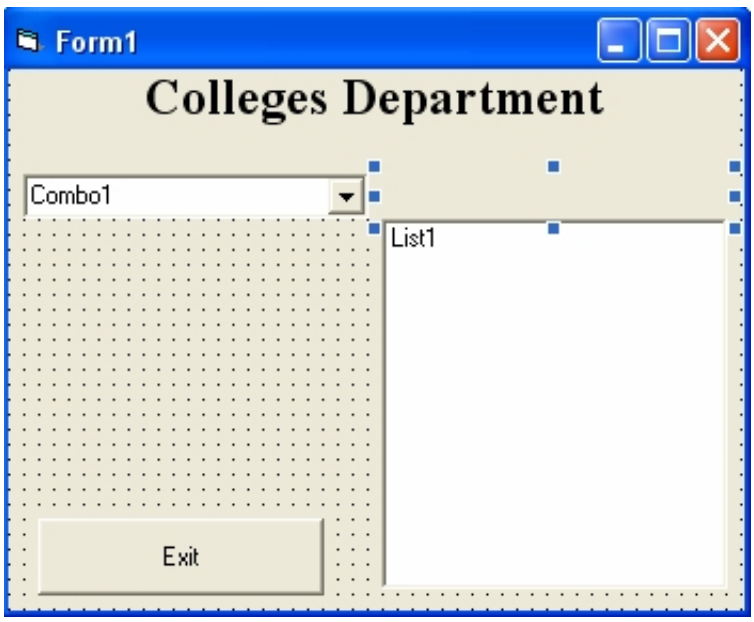
ان بنية القرار Select Case هي اكثر وضوحا من البنية If … Then واكثر فاعلية عند اتخاذ ثلاثة قرارات متفرعة او اكثر بناء على متغير واحد او على خاصية واحدة . لكن عند اجراء مقارنتين او اقل ، او عند العمل على عدة قيم مختلفة، قد تفضل استعمال البنية If … Then .

استعمال البنية Select Case لمعالجة القائمة المنسدلة

1. اختر الامر New Project < File وانشيء تطبيقا قياسيا جديدا.
   يظهر نموذج فارغ في بيئة البرمجة.
2. انقر اداة التسمية (Label) في مربع الادوات ثم انشيء مربعا كبيرا في النصف العلوي للنموذج لاظهار عنوان للبرنامج.
3. انقر اداة القائمة (List) في مربع الادوات ثم انشئ مربع سرد تحت تسمية العنوان.
4. انشئ تسمية صغيرة فوق كائن مربع السرد.
5. انقر اداة زر الامر في مربع الادوات ثم انشئ زر امر صغير في اسفل النموذج.
6. انقر زر اطار الخصائص في شريط الادوات ثم اضبط خصائص الكائنات التالية على النموذج:

الكائن	القيمة	الاعداد
Label1	Caption	Colleges Department
	Font	Time New Roman ، Bold ، 16 نقطة
Label2	Caption	"" (هذه العلامة تعني فارغ)
Command1	Caption	Exit

7. بعد الانتهاء من ضبط الخصائص ، يجب ان يبدو النموذج شبيها بالشكل (8-5)



شكل (8-5)

ستكتب الان شفيرة البرنامج لتمهيد القائمة المنسدلة.

8. انقر النموذج نقرا مزدوجا.

يظهر الاجراء الحدثي Form_Load في اطار الشفيرة.

9. اكتب الشفيرة البرمجية التالية لاعطاء قيمة بدء للقائمة المنسدلة.

```
Combo1.AddItem "College Of medicine"
Combo1.AddItem "College Of Engineering"
Combo1.AddItem "College Of Science"
Combo1.AddItem "College Of Art"
Combo1.AddItem "College Of Education"
```

10. تستعمل هذه الاسطر الطريقة AddItem الخاصة بكائن القائمة المنسدلة لاضافة الادخالات الى القائمة المنسدلة الموجودة على نموذجك.

11. افتح مربع سرد الكائنات ثم انقر على الكائن Combo1.

يظهر الاجراء الحدثي Combo1_Change في اطار الشفيرة ، افتح مربع سرد الاجراءات الحدثية ثم اختر الاجراء الحدثي Combo1_Click

يظهر الاجراء الحدثي Combo1_Click في اطار الشفيرة

12. اكتب الاسطر التالية لمعالجة انتقاء القائمة المنسدلة الذي قام به المستخدم:

```
Label2.Caption = Combo1.Text
Select Case Combo1.ListIndex

Case 0
    List1.Clear
    List1.AddItem "Anatomy"
    List1.AddItem "Gentics"
    List1.AddItem "Dentisrty"
    List1.AddItem "Pharmacy"
Case 1
    List1.Clear
    List1.AddItem "Computer Eng."
    List1.AddItem "Electrical Eng."
    List1.AddItem "Civil Eng."
    List1.AddItem "Mechanics Eng."
Case 2
    List1.Clear
    List1.AddItem "Computer Sci."
    List1.AddItem "Zoology"
    List1.AddItem "Mathematics "
    List1.AddItem "Chemistry"
Case 3
```

```
    List1.Clear
    List1.AddItem "linguistics"
    List1.AddItem "archaeology"
    List1.AddItem "grammar"
    List1.AddItem "History"
Case 4
    List1.Clear
    List1.AddItem "Biology"
    List1.AddItem "Holy Quran"
    List1.AddItem "Mathematics"
    List1.AddItem "Chemistry"
End Select
```

ينسخ السطر الاول اسم بند القائمة المنسدلة المنتقى الى التسمية التوضيحية الخاصة بالتسمية الثانية Label2 على النموذج. اهم خاصية مستعملة في الجملة هي Combo1.Text التي تحتوي على نص البند المنتقى في القائمة المنسدلة. الجمل المتبقية هي جزء من بنية القرار Select Case . تستعمل البنية Combo1.ListIndex كمتغير حالة اختبار وتقارنه بعدة قيم. تحتوي الخاصية ListIndex دائما على رقم البند المنتقى في القائمة المنسدلة. البند العلوي أي "College Of medicine" هو 0 ، والثاني هو 1 والذي يليه هو 2 وهكذا. من خلال استعمال ListIndex تتمكن البنية Select Case بسهولة من معرفة ما اختاره المستخدم وتضيف الاقسام الخاصة بالكلية المنتقاة الى القائمة.

13.افتح مربع سرد الكائنات ثم انقر الكائن Command1 .

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة.

14.اكتب End في الاجراء الحدثي ثم اغلق اطار الشفيرة.

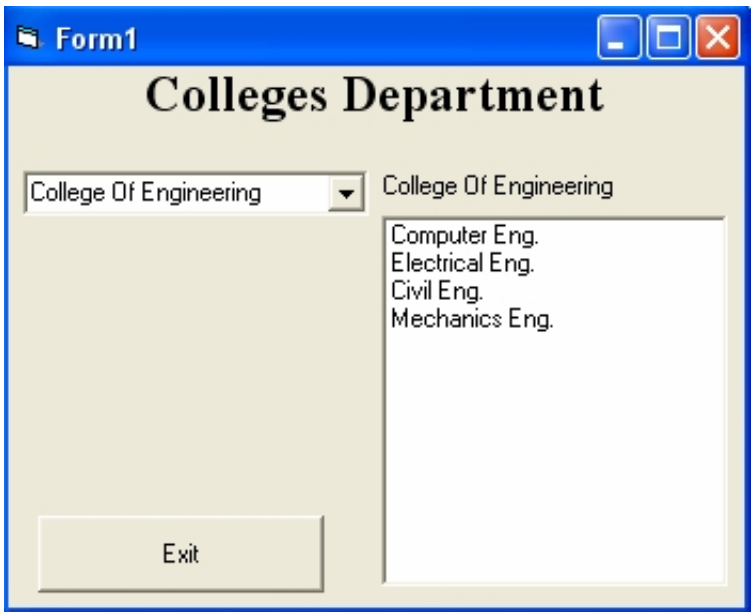
15.احفظ النموذج بالاسم Colleges Depart .

16.انقر زر التشغيل لتشغيل البرنامج.

17.اختر أي كلية من الكليات الموجودة في القائمة المنسدلة

يعرض البرنامج الاقسام التابعة لكل كلية موجودة في القائمة المنسدلة .

يعرض الشكل (9-5) اقسام كلية الهندسة.



شكل (5-5)

18. انقر الزر Exit لانهاء البرنامج.

يتوقف ابرنامج وتظهر بيئة البرمجة من جديد.

لقد انتهيت من العمل عل بنيات القرار Select Case في هذا التمرين.

الحلقات التكرارية

الحلقات التكرارية من الامور المهمة في كتابة البرامج حيث سترى انها ذات استعمالات واسعة ويمكن ان لايخلو أي برنامج من حلقة تكرارية . تستخدم الحلقة التكرارية لتكرار امر معين عدة مرات وذلك بتعين متغير كعداد ثم تتكرر العملية الى ان يصل العداد الى القيمة التي حددناها لتكرار الامر . تختصر الحلقات التكرارية عشرات الجمل البرمجية في عدة اسطر . ويمكن ان تستمر الحلقات التكرارية في تنفيذ جمل معينة حتى يتحقق شرط معين وفي حال لم يحقق الشرط تستمر الى ما لانهاية او لحين غلق البرنامج.

حدد القيمة الابتدائية والقيمة النهائية للحلقةFor … Next:

```
Dim i As Integer
For i = 2 To 4
Print i          سيتكرر تنفيذ الامر ثلاث مرات
Next
```

وعليك معرفة ان مقدار الزيادة سيضاف الى متغير الحلقة حتى بعد نهايتها:

```
Dim i As Integer
For i = 1 To 3
Print i
Next
```

قيمة المتغير بعد نهاية الحلقة 4 وليس3 ، Print i تستطيع التحكم في مقدار الزيادة او النقصان باستخدامStep:

```
Dim i As Integer
For i = 10 To 0 Step -1
Print i
Next
```

يمكنك تطبيق فكرة الحلقات المتداخلة Nested Loops كهذا الكود الذي يطبع جدول الضرب:

```
Dim A As Integer
Dim b As Integer
For A = 1 To 5
For b = 1 To 5
Print A, "x", b, "=", A * b
Next b
Next A
```

بامكانك انهاء الحلقة في أي وقت تريد باستخدام العبارةExit For:

```
Dim i As Integer
For i = 0 To 100
If MsgBox ( "هل تريد انهاء الحلقة؟" ,vbYesNo )  =  vbYes Then
Exit For
End If
…
Next
```

اما بالنسبة للحلقة Do … Loop فهي اكثر مرونة من الحلقة For … Next ، لانك لا تحدد عدد معين من التكرار وانما جملة شرطية باستخدام While اوUntil:

```
x = 10
Do While x >= 0
Print x
x = x - 1
Loop
```

```
x = 10
Do Until x <= 0
Print x
x = x - 1
Loop
```

ستتم عملية تنفيذ الحلقة مادامت الجملة الشريط صحيحة True في حال استخدام الكلمة المحجوزة While او False في حال استخدام الكلمة المحجوزة Until. واذا اردت تنفيذ الحلقة التكرارية مرة واحد على الاقل، ضع جملة الشرط في اسفل الحلقة:

```
x = -1
Do
Print x
x = x - 1
Loop While x >= 0
```

```
x = 1
Do
Print x
x = x - 1
Loop Until x >= 0
```

بامكانك وضع جملة الشرط في داخل الحلقة ايضا باستخدام عبارة If او ،Select لكن لا تنسى انهاء الحلقة بالعبارةExit Do:

```
x = 10
Do
If x <= 0 Then
Exit Do
End If
Print x
x = x - 1
Loop
```

التحويل بين For … Next وDo … Loop:

تستطيع تحويل حلقة For … Next الى حلقة Do … Loop والعكس صحيح، لكن عليك الانتباه الى ان القيم التي تحددها في بداية الحلقة For … Next تمثل عدد التكرار حتى وان تغيرت، فبالرغم من ان الحلقتين التاليتين متشابهتين:

```
A = 5
For i = 1 To A
…
Next
```

```
i = 1
Do
…
i = i + 1
Loop Until i > A
```

الا ان الاختلاف سيظهر في حال ما اذا تم تغيير قيمة المتغير A ، فالحلقة For … Next سيتم تنفيذه دائما 5 مرات حتى وان تغيرت قيمة المتغير A في داخل الحلقة، بينما تغيير القيمة يؤثر بشكل كبير على عدد مرات تكرار الحلقة Do … Loop.

.....

اما حلقة For Each فهي تطبق على كائنات المجموعات Collections:

```
Dim ctrl As Control
```

محاذاة جميع الادوات في النموذج الى اليسار'

```
For Each ctrl In Controls
ctrl.Left = 0
Next
```

استعمال الحلقات وعدادات الوقت

ستتعلم في هذا الدرس كيفية:

- ✓ استعمال حلقات For … Next لتنفيذ جمل لعدد محدد من المرات.
- ✓ استعمال الطريق Print لاظهار اخراج على النموذج.
- ✓ استعمال حلقة Do لقراءة محتويات قائمة واجراء عمليات عليها.
- ✓ استعمال عداد وقت لتكرار حلقة لمدة معينة من الزمن
- ✓ انشاء ساعة رقمية ومنبه مواعيد خاصين بك.

ستتعلم في هذا الدرس كيفية تنفيذ كتلة جمل مرارا وتكرارا وذلك عن طريق استعمال حلقة. ستستعمل حلقة For…Next لتنفيذ جملة عددا معينا من المرات ، ثم ستستعمل حلقة Do لتنفيذ جمل الى ان يقيم التعبير الشرطي في الحلقة الى صح. كما ستتعلم كيفية استعمال الطريقة Print لاظهار نصوص وارقام على النموذج ، وستتعلم كيفية استعمال عداد وقت لتنفيذ الشفيرة عند فواصل زمنية محددة في البرنامج.

عرض المعلومات باستعمال حلقة For … Next

1. شغل برنامج فيجوال بيسك وافتح مشروعا جديدا.
2. زد طول النموذج باستعمال مؤشر التحجيم لانجاز بعض المساحة الاضافية لاظهار الاخراج.
3. استعمل اداة التحكم زر امر لانشاء زر امر في الجهة اليمنى للنموذج.
4. افتح اطار الخصائص ثم اضبط الخاصية Caption لزر الامر عند Loop.
5. افتح مربع سرد الكائنات في اعلى اطار الخصائص وانقر اسم الكائن Form1 . تظهر خصائص النموذج في اطار الخصائص .
6. غير الخاصية Font الى Time New Roman.

تتحكم الخاصية Font بكيفية ظهور النص على النموذج. يمكنك استعمال أي خط متوفر في الكمبيوتر خطوط الـ TrueType هي الافضل لانه يمكنك عرضها بعدة احجام ولان شكلها على الشاشة هو نفسه عند الطباعة.

7. غير الخاصية AutoRedraw الى True.

اذا تم اخفاء نموذجك ، ستعيد الخاصية AutoRedraw اظهار أي نص تم عرضه بواسطة الطريقة Print.

8. انقر الزر Loop على النموذج نقرا مزدوجا.

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة.

9. اكتب الجمل البرمجية التالية في الاجراء :

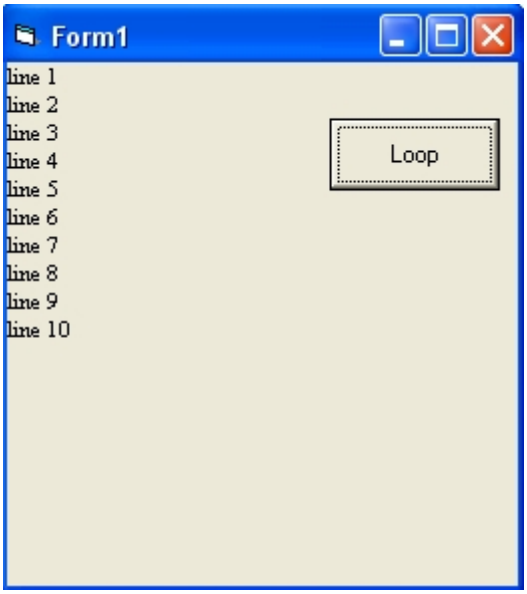
```
For i = 1 to 10
    Print "Line" ; i
Next i
```

تستعمل الحلقة For … Next هنا الطريقة Print لعرض الكلمة Line يليها عداد الحلقة 10 مرات على النموذج. توجه الفاصلة المنقوطة (;) في الجملة Print فيجوال بيسك الى عرض المتغير العداد بجانب السلسلة "Line" من دون ترك مسافة اضافية بينهما (لكنك رغم ذلك ستشاهد مسافة بين "Line" وقيمة العداد عندما يشتغل البرنامج. عند طباعة قيم رقمية ، تحجز الطريقة Print مساحة لعلامة الطرح، حتى لو لم تكن هناك حاجة لها).

10. انقر زر التشغيل في شريط الادوات.

11.انقر الزر Loop.

تطبع الحلقة For … Next عشرة اسطر على النموذج كما هو مبين في شكل (1-6)



شكل (1-6)

12.انقر الزر Loop مرة ثانية.

تطبع الحلقة For … Next عشرة اسطر اخرى على النموذج (او بقدر ما يتسع له النموذج) .

13.انقر زر الانهاء في شريط الادوات.

14.انقر زر حفظ المشروع ، واحفظه باسم MyLoop.

تغيير خاصية في الحلقة For … Next

يتيح لك فيجوال بيسك تغيير الخصائص وتحديث المتغيرات المهمة في الحلقة . في التمرين التالي ستعدل البرنامج السابق (MyLoop) بحيث يغير الخاصية FontSize في الحلقة For…Next ، تحدد هذه الخاصية حجم الاحرف على النموذج، ويمكنك استعمالها كبديل لتغيير الحجم بواسطة الخاصية Font.

تغيير الخاصية FontSize

1. انقر الزر Loop نقرا مزدوجا.

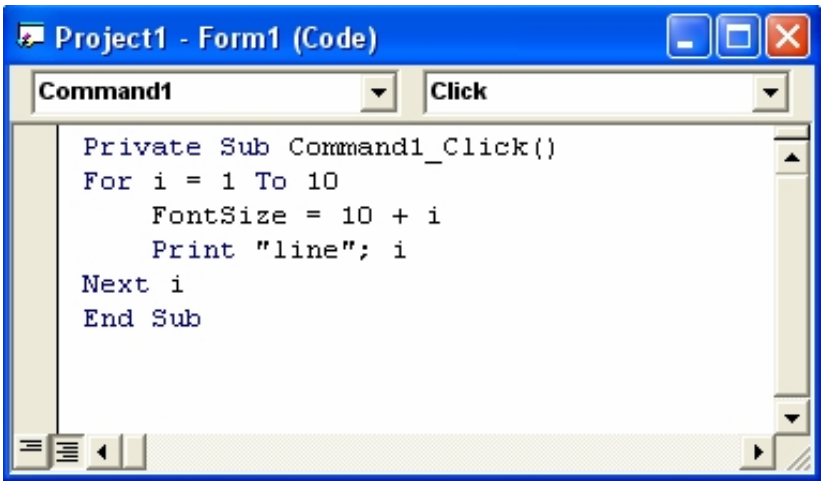
تظهر الحلقة For … Next في اطار الشفيرة.

2. اكتب الجملة التالية تحت الجملة For مباشرة:

```
FontSize = 10 + i
```

تضبط هذه الجملة الخاصية FontSize للنموذج عند 10 نقاط اكثر من قيمة عداد الحلقة. في المرة الاولى خلال الحلقة سيضبط حجم الخط عند 11 نقطة وفي المرة الثانية عند 12 نقطة وهكذا حتى الحلقة الاخيرة عندما يتم تكبير الخط الى 20 نقطة.

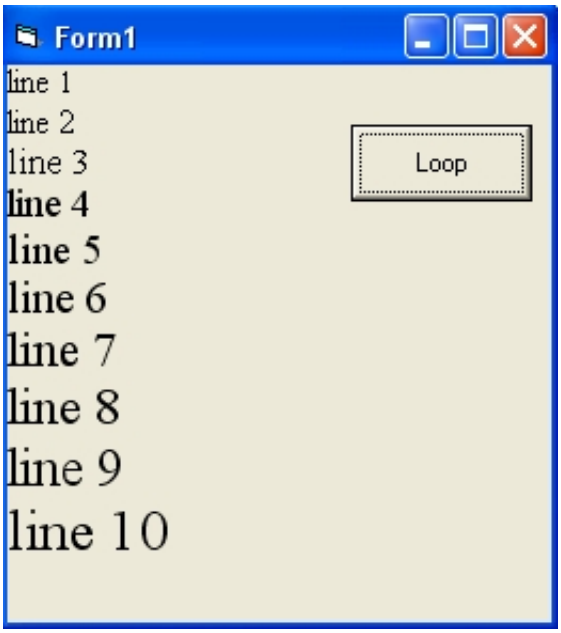
بعد الانتهاء يجب ان تبدو الحلقة كما في شكل (6-2).



شكل (6-2)

3. انقر زر التشغيل في شريط الادوات.
4. انقر الزر Loop.

تعرض الحلقة For … Next الاخراج التالي على النموذج. كما في شكل (6-3)



شكل (6-3)

كل مرة تم فيها زيادة عداد الحلقة ، ادى ذلك الى تكبير حجم الاحرف على النموذج.

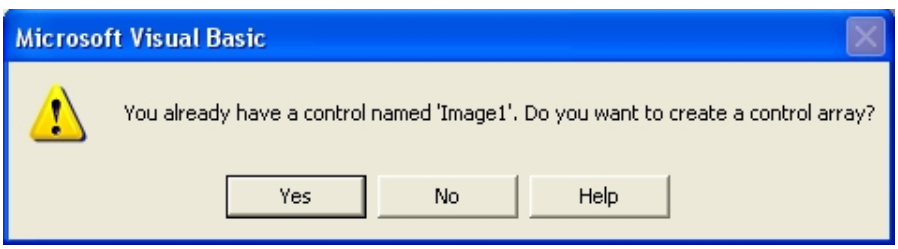
5. انقر زر انهاء لايقاف البرنامج.

مهم

توفر الحلقة For … Next مساحة كبيرة في البرنامج، في المثال السابق عالجت حلقة For…Next التي طولها اربع اسطر فقط ما يوازي ما يعالجه عشرون جملة برمجية.

فتح الملفات باستعمال حلقة For … Next

1. اختر الامر New Project < File ثم انقر OK.
2. انقر اداة التحكم صورة من مربع الادوات ثم انشئ مربع صورة صغير بالقرب من الزاوية اليسرى العليا للنموذج.
3. اختر الامر Copy < Edit. او عن طريق الضغط على مفتاحي Ctrl + C . توضع نسخة عن مربع الصورة في حافظة ويندوز . ستستعمل هذه النسخة لانشاء ثلاثة مربعات صور اضافية على النموذج.
4. اختر الامر Edit > ثم Paste او عن طريق الضغط على مفتاحي Ctrl + V. يعرض فيجوال بيسك رسالة تسالك عما اذا كنت ترغب بانشاء مصفوفة تحكم (Control Array) في البرنامج. **مصفوفة التحكم** هي مجموعة كائنات متماثلة في واجهة البرنامج . يحمل كل كائن في المجموعة نفس الاسم (لكن يحمل كل كائن رقم فهرس (Index) فريد خاص به) لذا يمكن انتقاء المجموعة بكاملها وتعريفها دفعة واحدة. لكن يمكن استعمال كل كائن في المصفوفة على حدة مما يعطيك تحكما كاملا بكل بند من البنود الموجودة في واجهة البرنامج. سوف تكون الرسالة كما في شكل (6-4)



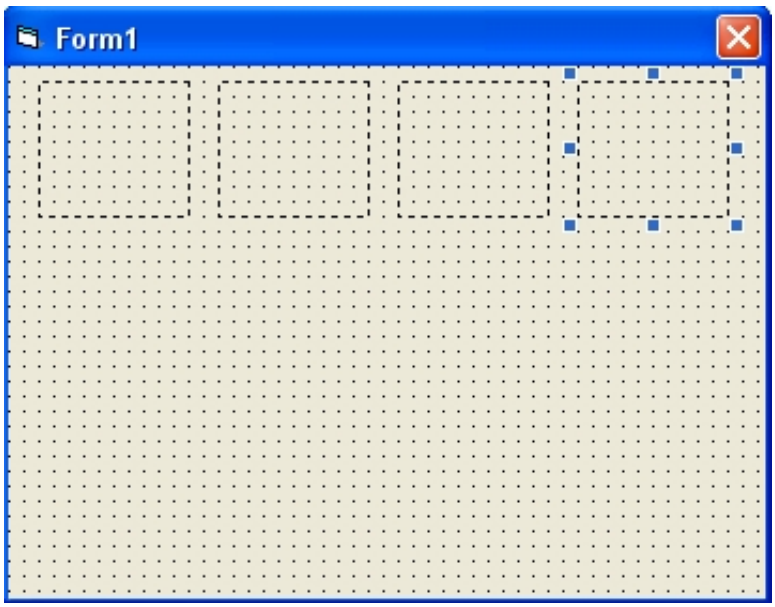
شكل (6-4)

5. انقر Yes لانشاء مصفوفة تحكم.

ينشئ فيجوال بيسك **مصفوفة تحكم** مؤلفة من مربعات صور ويلصق مربع الصورة الثاني في زاوية النموذج اليسرى العليا. يتم انتقاء مربع الصورة الجديد.

6. اسحب مربع الصورة الثاني الى يمين مربع الصورة الاول.
7. اختر الامر Paste او Ctrl+V ، ثم اسحب مربع الصورة الثالث الى يمين مربع الصورة الثاني.
8. اختر الامر Paste او Ctrl+ V ، مرة اخرى ثم اسحب مربع الصورة الرابع الى يمين مربع الصورة الثالث.

بعد الانتهاء من اللصق يجب ان تبدو الشاشة كما في شكل (5-6).



شكل (5-6)

9. انقر اداة التحكم زر امر في مربع الادوات ثم انشئ زر امر في اسفل النموذج.

ستضبط الان خصائص الكائنات على النموذج. ستضبط اولا خصائص مربعات الصور في مصفوفة التحكم باعتبارها مجموعة واحدة.

10. انقر مربع الصورة الاول واضغط على المفتاح Shift باستمرار ثم انقر مربع اصورة الثاني والثالث والرابع.

تظهر جميع مربعات الصور في مصفوفة التحكم منتقاة كلها على النموذج.

11. افتح اطار الخصائص ثم اضبط الخصائص التالية (بعد الانتهاء من ضبط خصائص مربعات الصورة انقر زر الامر لضبط خاصيته).

الكائن	الخاصية	الإعداد
مصفوفة التحكم Image1	BoarderStyle	1-Fixed Single
	Stretch	True
Command1	Caption	"Display Images"

12.انقر نقرا مزدوجا الزر Display Images على النموذج لعرض الاجراء الحدثي الخاص بكائن زر الامر.

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة.

13. سنقوم الان بجلب اربعة صور من MyDocument او من أي مكان يحوي على صور ثم نقوم بتغيير اسمائهم الى اسم معين لكن متسلسل ، ساقوم بتسميتهم Natral1 للصورة الاولى و Natral2 للثانية و Natral3 و Natral4 للثالثة والرابعة، هنا تعمدت ان تكون جميع الصور تحمل نفس الاسم ولكن تختلف في الرقم وذلك لكي استطيع ان اثبت جزء النص من الاسم "Natral" واغير الرقم في كل مرة . ساقوم بنسخ هذه الصور الى مجلد قد انشئته مسبقا اسمه NatrPicture وقد وضعته في (C:\) لذا ساقوم باستدعاء الصور منه.

تنويه:

يفضل ان تكون جميع الصور بنفس الابعاد وذلك لكي تظهر بنسق جميل، **ولكن يجب ان تكون من نفس النوع أي تحمل نفس الامتداد.**

14.كبر اطار الشفيرة ثم اكتب الحلقة For… Next التالية:

```
For i = 1 to 4
Image(i – 1).Picture = LoadPicture ("C:\NatrPicture\Natral" & i & ".jpg")
Next i
```

الحلقة تستعمل الدالة LoadPictrue لتحميل اربعة ملفات صورية من المجلد NatrPicture في قرصك الثابت . القسم الخدعة من الحلقة هو الجملة

```
Image(i – 1).Picture = LoadPicture ("C:\NatrPicture\Natral" & i & ".jpg")
```

التي تحمل الملفات من القرص الثابت . القسم الاول من الجملة

```
Image1( i – 1 ).Picture
```

يستعمل الخاصية Picture لكل مربع صورة في مصفوفة التحكم . يمكنك استعمال البنود في مصفوفة التحكم من خلال فهارسها (Index) ، لذا تستطيع ذكر مربعات الصور بشكل فردي في هذا المثال كالتالي : Image1(0) و Image1(1) و Image1(2) و Image1(3) . الرقم

الموجود بين القوسين هو قيمة الـ Index في المصفوفة، في هذا المثال يتم احتساب قيمة الفهرس (Index) الصحيح من خلال طرح 1 من المتغير i "العداد" . يتم انشاء اسم الملف باستعمال المتغير العداد i وعامل الضم & الذي تعلمت عنه في دروس سابقة.

الشفيرة البرمجية

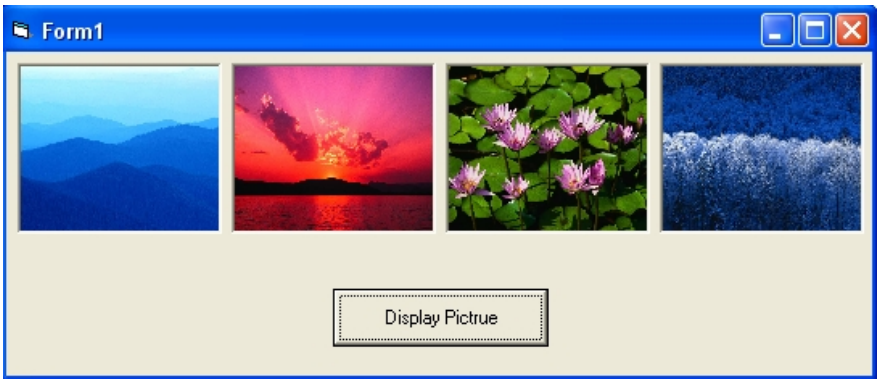
```
LoadPicture ( "C:\NatrPicture\Natral" &  i & ".jpg")
```

تدمج اسم مسار واسم ملف والامتداد ".jpg" لانشاء اربع اسماء ملفات صورية صالحة في القرص الثابت. في هذا المثال، يتم تحميل Natral1.jpg و Natral2.jpg و Natral3.jpg و Natral4.jpg في مربعات الصور.

15.انقر زر حفظ المشروع .

16.انقر زر التشغيل واضغط على زر Display Picture

تُحمل الحلقة For … Next الصور من القرص الثابت الى مربعات الصورة. سيعرض البرنامج الاخراج التالي كما مبين في شكل (6-6) ،" ستعرض الصور التي تختارها انت"



شكل (6-6)

الجملة Exit For

تتيح لك الجملة Exit For الخروج من الحلقة For … Next قبل ان ينتهي تنفيذ كل تكراراتها . هكذا سيصبح بامكانك الاستجابة الى حدث معين حصل قبل ان تتم الحلقة العدد المحدد من المرات. مثلا الحلقة For … Next التالية:

```
For i = 1 To 5
    InpName = InputBox ("Enter your name ")
    If InpName = "Ahmed" then
    MsgBox " Welcome " & InpName
```

```
        Exit For
        End If
        MsgBox "Sorry , I can't recognize you!" & InpName
    Next i
```

تحث المستخدم على كتابة اسم معين (هنا نريد كلمة Ahmed) فاذا قام المستخدم بادخال كلمة Ahmed فان الشرط سوف يتحقق وتعرض رسالة الترحيب ثم تستعمل الجملة Exit For للخروج من الحلقة For … Next ، اما اذ لم يدخل كلمة Ahmed فسوف لن يتحقق الشرط وتعرض رسالة عدم تمييز الشخص وهكذا.

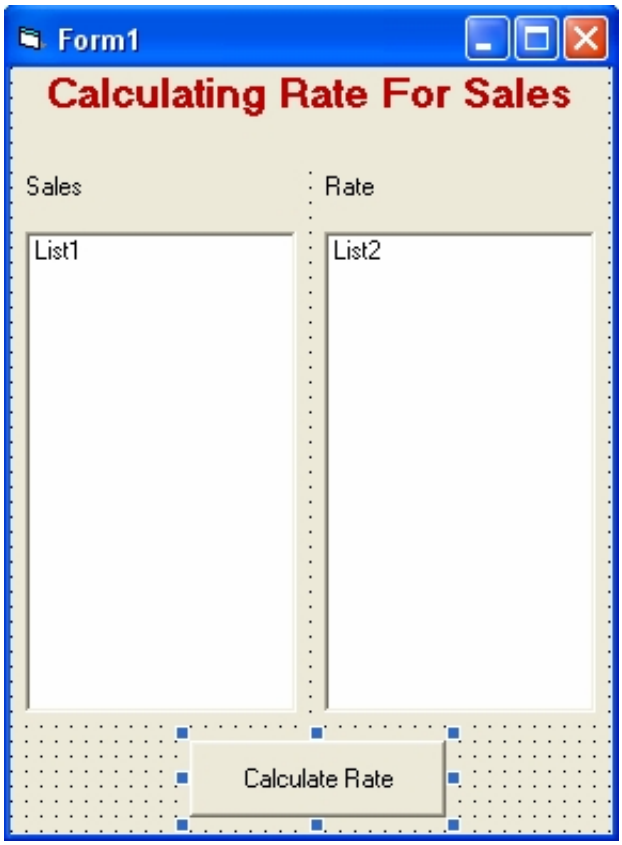
تستعمل Exit For عادة مع الجمل IF.

قراءة عناصر قائمة واجراء بعض العمليات عليها باستخدام حلقة Do

1. اختر الامر New Project < File ثم OK.
   يعرض فيجوال بيسك نموذجا جديدا في بيئة البرمجة.
2. انقر اداة التحكم القائمة List من مربع الادوات ثم قم بانشاء قائمة في الجهة اليسرى من النموذج.
3. كرر العملية مرة اخرى لكن ضع القائمة الثانية الى يمين القائمة الاولى
4. انقر اداة التحكم زر الامر Command من مربع الادوات وضعه اسفل النموذج في الوسط.
5. انقر اداة التحكم تسمية Label من مربع الادوات وقم بانشاء تسمية للعنوان وضعها في اعلى النموذج وعلى طول النموذج.
6. قم بوضع تسميتين اخرتين فوق كل قائمة .
7. افتح اطار الخصائص ثم قم بضبط الخصائص للكائنات كما مبين في الجدول.

الكائن	الخاصية	القيمة
Command1	Caption	Calculate Rate
Label1	Caption	Calculating Rate For Sales
	Alignment	2-Center
	ForeColor	"Red"
Label2	Caption	Sales
Label3	Caption	Rate

يجب ان يبدو نموذجك كما مبين في شكل (7-6).



شكل (7-6)

8. انقر النموذج نقرا مزدوجا.

يظهر الاجراء الحدثي Form_Load في اطار الشفيرة . سنقوم هنا باضافة اسعار المبيعات في القائمة List1 .

9. اكتب الجمل البرمجية التالية :

List1.AddItem 10
List1.AddItem 15
List1.AddItem 12
List1.AddItem 16
List1.AddItem 30
List1.AddItem 5
List1.AddItem 45
List1.AddItem 100
List1.AddItem 65
List1.AddItem 80
List1.AddItem 25
List1.AddItem 78
List1.AddItem 88
List1.AddItem 32

List1.AddItem 85

بامكانك ادخال اسعار حسب الرغبة .

تقوم الجملة البرمجية List1.AddItem باضافة عنصر الى القائمة رقم 1 حيث يستخدم الايعاز AddItem لاضافة عناصر الى القائمة . هنا قمنا بكتابة الجملة البرمجية 15 مرة وذلك لاضافة 15 عنر في القائمة. يمكن وضع الجملة List1.AddItem داخل زر امر وعند الضغط عليه نقوم باخذ قيمة العنصر من داخل مربع نص TextBox او عن طريق InputBox ثم نقوم باضافته الى القائمة كما مبين

```
List1.AddItem  Text1.text
```

او

```
Itm = InputBox ( "اضافة الى القائمة" )
List1.AddItem   Itm
```

10.انقر زر الامر Calculate Rate نقرا مزدوجا.

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة .

11.اكتب الجمل البرمجية التالية :

```
List2.Clear
Msg = " Please Enter The Tax Value. It Will compute as % Percent"
tax = InputBox ( Msg, "Rate Value")
If  Val( tax ) <> 0 Then
x = List1.ListCount
i = 0
Do While x > 0
Itm = List1.List(i) * Val( tax ) /  100
List2.AddItem   Itm
i = i + 1
x = x - 1
Loop
End If
```

تهتم هذه الاسطر البرمجية بحساب الضريبة ، يقوم السطر الاول بمسح محتويات القائمة الثانية وذلك في حالة ان لم تكتب List2.Clear ، فاذا قمت بالضغط على زر الامر Calculate Rate اكثر من مرة فان القائمة سوف لم تمسح المحتويات السابقة بل سوف تضيف المعلومات الجديدة فوقها لذلك استخدما خاصية تنظيف محتويات القائمة قبل كل شئ.

يعين السطر الثاني سلسلة نصية الى المتغير Msg ، الذي يستعمل بعد ذلك لعرض رسالة تعليمات في مربع الادخال InputBox ، يعين مربع الادخال InputBox قيمة الضريبة التي ادخلها المستخدم (التي تحسب % أي اذا ادخل 5 فسوف تحسب 5% وهكذا) الى المتغير tax تستخدم الجملة الشرطية If في مقارنة قيمة المتغير tax فاذا كانت 0 فان الجملة الشرطية سوف لن تتحقق وينتهي الاجراء اما اذا كانت لا تساوي 0 ، فسوف تقوم الجملة List1.ListCount بتعين عدد عناصر القائمة List1 الى المتغير X ، تستخدم الخاصية ListCount لمعرفة عدد عناصر القائمة. نقوم بعد ذلك بتعيين القيمة 0 الى المتغير i حيث سوف نستخدمه كفهرس (Index) للوصول الى عناصر القائمة الاولى،

تستخدم الجملة Do الشرط While أي معناه استمر في تكرار العمليات المحصورة بين Do و Loop مادام المتغير X اكبر من 0 ، فاذا اصبح صفرا او اقل من صفر فسوف تتوقف الجملة Do عن التنفيذ،

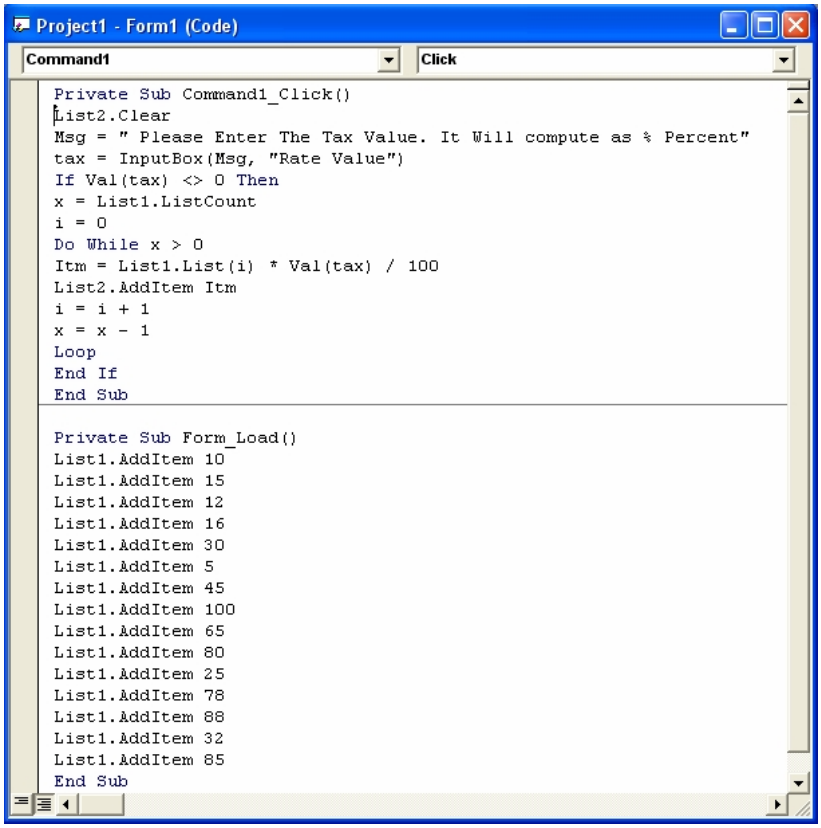
تعمل الجملة List1.List(i) * Val(tax) / 100 بسحب اول عنصر من القائمة الاولى List1 ثم تضربها بقيمة الضريبة التي قام المستخدم بادخالها ثم تقسم الناتج على 100 وذلك لان الضريبة سنحسبها % ، ثم بعد ذلك نقوم بتعيين الناتج الى المتغير Itm ، تستخدم الخاصية List(i) لقراءة العنصر ذو الموقع i في القائمة ،حيث تعتبر العناصر في القائمة كعناصر المصفوفة أي كل عنصر يحتوي على رقم فهرس (Index) ، يحدد موقع (عنوان) العنصر في القائمة (المصفوفة) لذلك نستخدم الخاصية List( ) لقراءة العنصر بعد كتابة رقم العنصر بين القوسين.

بعد ان قمنا باحتساب قيمة الضريبة ووضعها في المتغير Itm ، نقوم الان باضافة المتغير Itm في القائمة الثانية List2 باستخدام الخاصية AddItem ،

بعد ذلك نقوم بزيادة المتغير i بمقدار 1 حتى نقوم بقراءة قيمة العنصر الثاني في القائمة وذلك لان اول عنصر في القائمة يكون الفهرس (Index) له يساوي 0 والثاني =1 وهكذا ... أي اخر عنصر يكون الـ (Index) له يساوي ListCount – 1 ،

بعد ذلك نقوم بانقاص المتغير X بمقدار 1 وذلك لاننا نستعمله في تحقيق شرط الجملة Do فعندما يصل الصفر سوف تتوقف الجملة Do عن التكرار. اما الايعاز Loop فيستخدم للتكرار أي

سوف يعيد تنفيذ الجمل المحصورة بين Do و Loop . يجب ان تبدو الشفيرة البرمجية كما في شكل (6-8)



شكل (6-8)

12. انقر زر التشغيل في شريط الادوات.

يشتغل البرنامج وترى ان القائمة الاولى قد وضع فيها القيم التي ادخلتها .

13. انقر زر الامر Calculate Rate سيطلب منك البرنامج ادخال رقم ، قم بادخال أي رقم وانقر زر OK سترى ان البرنامج قد قام باحتساب قيمة الضريبة لكل عنصر في القائمة الاولى ووضعها في القائمة الثانية.

14. انقر زر ايقاف لايقاف البرنامج ، احفظ البرنامج باسم معين.

استعمال الكلمة الدليلية Until في الحلقات Do

ان الحلقات Do التي عملت عليها لحد الان استعملت الكلمة الدليلية While لتنفيذ مجموعة من الجمل طالما ان شرط الحلقة بقي صح، يتيح لك فيجوال بيسك ايضا استعمال الكلمة الدليلية Until في الحلقات Do ليتكرر تنفيذها الى ان يصبح الشرط صح. يمكن استعمال الكلمة الدليلية Until في اعلى الحلقة Do او في اسفلها لاختبار الشرط ، تماما

كما هو الحال مع الكلمة الدليلية While. مثلا، تستعمل الحلقة Do التالية الكلمة الدليلية Until لتتكرر الى ان يكتب المستخدم الكلمة "Done" في مربع ادخال:

```
Do
    InpName = InputBox ( "Enter Your name or Done.")
    If  InpName <> "Done"  Then Print InpName
Loop  Until InpName = "Done"
```

كما ترى ، الحلقة التي تستعمل الكلمة الدليلية Until تشبه كثيرا الحلقة التي تستعمل While ، باستثناء ان شرط الاختبار يحتوي عادة على العامل المعاكس = (يساوي) بدلا من <> (لايساوي) في هذه الحالة.

تجنب الحلقات اللامتناهية

نظرا لطبيعة الحلقات Do، من المهم جدا ان تصمم شرطها بحيث يكون لها نقطة خروج حقيقية . اذا لم يتم تقييم اختبار الحلقة الى خطا ابدا، سيتم تنفيذ الحلقة الى مالانهاية ولن يستجيب البرنامج الى الادخال بعد ذلك . اليك المثال التالي:

```
Do
    Number = InputBox ( "Enter a number to square. Type -1 to quit")
    Number = Number ^ 2
    Print Number
Loop While Number >=0
```

في هذه الحلقة ، يكتب المستخدم رقما تلو الاخر ويقوم البرنامج بتربيعه وطبعه على النموذج. لسوء الحظ، عندما يرغب المستخدم بانهاء البرنامج، لن يتمكن من ذلك لان شرط الخروج المشار اليه لا يعمل . اذ عندما يكتب المستخدم 1- يربعه البرنامج وتصبح قيمة المتغير Number تساوي 1 لذلك سوف لن يتحقق شرط الخروج. (يمكن اصلاح هذه المشكلة من خلال وضع شرط خروج اخر) . يجب الانتباه من الحلقات اللامتناهية عند كتابة الحلقات Do . لحسن الحظ ، من السهل اكتشاف هذه الحلقات اذا كنت تختبر برنامجك بشكل جيد.

استعمال كائنات عدادات الوقت

يتح لك فيجوال بيسك تنفيذ مجموعة من الجمل **لمدة معينة من الوقت** وذلك باستعمال كائن عداد وقت الـ (Timer) . **عداد الوقت** هو اداة توقيت غير مرئية تتيح لك استعمال ساعة الكمبيوتر من ضمن برامجك. يمكنك استعماله للعد تنازليا من وقت معين او للتسبب بحصول تاخير في البرنامج او لتكرار عمل ما عند فترات زمنية محددة.

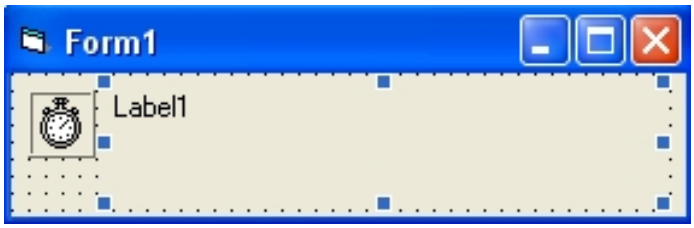
تصل دقة العداد الى 1 بالالف من الثانية . ومع ان العدادات لا تكون مرئية خلال التشغيل ، الا ان كل عداد يكون مقترنا باجراء حدثي يشتغل كلما انقضت **الفترة الزمنية** المحددة له. يتم تحديد الفترة الزمنية للعداد باستعمال الخاصية Interval، اما تنشيط العداد فيتم من خلال ضبط خاصيته Enable عند True . عند تنشيط العداد سيشتغل دون توقف (وينفذ اجراءه الحدثي عند الفترة الزمنية المحددة) الى ان ينهي المستخدم البرنامج او يتم تعطيل العداد.

انشاء ساعة رقمية باستعمال كائن عداد وقت Timer

اكثر الاستعمالات العملية للعداد هي الساعة الرقمية . ستنشئ في التمرين التالي ساعة رقمية بسيطة تتعقب الوقت الحالي وصولا حتى الثانية. ستضبط في هذا المثال الخاصية Interval للعداد عند 1000 لتجعل فيجوال بيسك يحدث وقت الساعة كل 1000 ملي ثانية ، أي مرة كل ثانية .

انشاء البرنامج Digital Clock

1. اختر الامر New Project < File ثم OK.
2. غير حجم النموذج الى اطار صغير .
   لاتريد ان تحتل الساعة مساحة كبيرة.
3. انقر اداة التحكم عداد الوقت Timer في مربع الادوات.
4. انشئ عداد وقت صغير في الناحية اليسرى من النموذج.
   عندما تنتهي من انشاء العداد، يعيد فيجوال بيسك تحجيمه الى حجمه القياسي .
5. انقر اداة التحكم تسمية Label في مربع الادوات.
6. انشئ تسمية تملا معظم مساحة النموذج.
   ستستعمل التسمية لاظهار الوقت. يجب ان يبدو النموذج كما في شكل (9-6)



شكل (6-9)

7. افتح اطار الخصائص واضبط الخصائص التالية في البرنامج.

الكائن	الخاصية	الاعداد
Label1	Caption	فارغ
	Font	Time New Roman، Bold، 24 نقطة
	Alignment	2-Center
Timer1	Interval	1000
	Enable	True
Form1	Caption	"Digital Clock"

8. انقر كائن العداد Timer1 على النموذج نقرا مزدوجا.

يظهر الاجراء الحدثي Timer1_Timer في اطار الشفيرة.

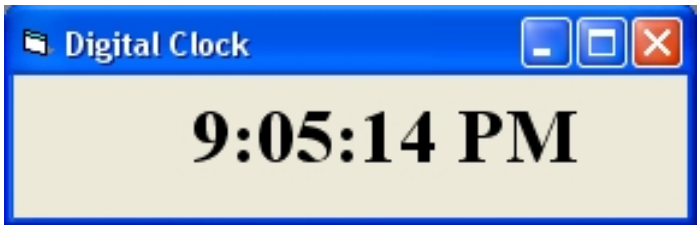
9. اكتب الجملة التالية:

```
Label1.Caption = Time
```

تحصل هذه الجملة على الوقت الحالي من ساعة الكمبيوتر وتعينه الى الخاصية Caption للكائن Label1 . يتطلب هذا البرنامج جملة واحدة فقط لانك ضبطت خاصية عداد الوقت Interval في اطار الخصائص عند 1000. يتولى عداد الوقت الباقي

10. اغلق اطار الشفيرة وانقر زر التشغيل في شريط الادوات لتشغيل الساعة.

تظهر الساعة كما مبين في شكل (6-10).



شكل (6-10)

11.راقب الساعة لبعض الوقت.

يُحدث فيجوال بيسك الوقت كل ثانية.

12.احفظ النموذج والمشروع بالاسم MyDigitalClock.

خطوة متقدمة: استعمال كائن عداد وقت لضبط حدود زمنية

يمكنك استعمال عداد الوقت ايضا بحيث ينتظر انقضاء مدة محددة من الوقت ويقوم بعدها بالسماح بتنفيذ عمل ما او بمنعه.

يبين لك التمرين التالي كيف يمكنك استعمال هذه الفكرة لضبط حدود زمنية قصوى لكتابة كلمة مرور (كلمة المرور لهذا البرنامج هي "visualbasic"). يستعمل البرنامج عداد وقت لانهاء نفسه اذا لم يتم ادخال كلمة مرور صالحة خلال 15 ثانية (عادة يكون هذا البرنامج جزءا من برنامج اكبر). يمكنك استعمال تقنية عداد الوقت هذه ايضا لعرض رسالة ترحيب او رسالة حقوق النشر على الشاشة. او لتكرار حدث كل فترة زمنية معينة كحفظ الملف على القرص كل عشر دقائق مثلا.

ضبط حدود زمنية قصوى لكلمة المرور

1. اختر الامر New Project < File ثم OK.
2. غير حجم النموذج الى اطار مستطيل صغير بحجم مربع ادخال.
3. انقر اداة التحكم مربع نص TextBox في مربع الادوات.
4. انشئ مربع نص مستطيل في وسط النموذج.
5. انقر اداة التحكم تسمية Label في مربع الادوات ثم انشئ تسمية طويلة فوق مربع النص.
6. انقر اداة التحكم زر امر Command في مربع الادوات ثم انشئ زر امر تحت مربع النص.
7. انقر اداة التحكم عداد الوقت في مربع الادوات.
8. انشئ عداد وقت في الزاوية اليسرى السفلى للنموذج (او في أي مكان تحب).
9. اضبط الخصائص التالية للبرنامج.

الكائن	الخاصية	الاعداد
Text1	Text	فارغ
	PasswordChar	*
Label1	Caption	Enter Your Password Within 15 seconds
Command1	Caption	Try Password
Timer1	Interval	15000
	Enable	True
Form1	Caption	Password

ستعرض الخاصية PasswordChar نجوما (*) في مربع النص عندما يكتب المستخدم كلمة المرور . ان ضبط الخاصية Interval عند 15000 يعطي المستخدم مهلة 15 ثانية ليكتب كلمة المرور وينقر الزر Try Password . يؤدي ضبط الخاصية Enable عند True (الاعداد الافتراضي) الى تشغيل عداد الوقت عندما يبدا البرنامج (يمكنك ايضا تعطيل هذه الخاصية ثم تنشيطها في الاجراء الحدثي اذا كنت لاتحتاج الى عداد الوقت الا لاحقا في البرنامج).

يجب ان يبدو النموذج كما في شكل (11-6)

شكل (11-6)

10.انقر عداد الوقت Timer1 في النموذج نقرا مزدوجا .

يظهر الاجراء Timer1_Timer في اطار الشفيرة. اكتب الجمل التالية:

```
MsgBox ( " Sorry your time is up.")
End
```

تعرض الجملة الاولى رسالة تشير الى انتهاء الوقت المحدد لادخال كلمة المرور ، وتنهي الجملة الثانية البرنامج. ينفذ فيجوال بيسك هذا الاجراء الحدثي اذا بلغت الفترة الزمنية لعداد الوقت 15 ثانية ولم يتم ادخال كلمة مرور صالحة.

11.انقر الكائن Command1 في مربع السرد المنسدل في اطار الشفيرة ثم اكتب الجمل البرمجية التالية في الاجراء الحدثي Command1_Click:

```
If  Text1.Text  =  "visualbasic" Then
     Timer1.Enable = False
      MsgBox ("Welcome to the system!")
      End
Else
      MsgBox ( "Sorry , Friend , I don't Know you.")
End If
```

تتاكد هذه الشفيرة مما اذا كانت كلمة المرور المكتوبة في مربع النص هي visualbasic ام لا . اذا كانت كذلك ، يتم تعطيل عداد الوقت وتظهر رسالة ترحيب وينتهي البرنامج (بامكانك جعل البرنامج يبقى مشتغلا وينفذ خطوات اخرى) . اذا لم تكن كلمة المرور مطابقة، يتم اعلام المستخدم بذلك من خلال رسالة ويمنح المستخدم فرصة اخرى لكتابة كلمة المرور، ولكن سيكون لديه 15 ثانية فقط ليفعل ذلك !

12.اغلق اطار الشفيرة ثم انقر زر التشغيل لتشغيل البرنامج.

يشتغل البرنامج وتبدا ساعة الـ 15 ثانية بالعمل.

13.اكتب كلمة " Project " او أي كلمة لكن ليس الكلمة الصحيحة في مربع النص ثم انقر Try Password.

يظهر مربع الحوار التالي على الشاشة ليبلغك ان كلمة المرور كانت خطاء.

14.انقر زر OK وانتظر الى ان تنتهي فترة طلب الدخول

يعرض البرنامج الرسالة التالية :

15. انقر زر حفظ المشروع

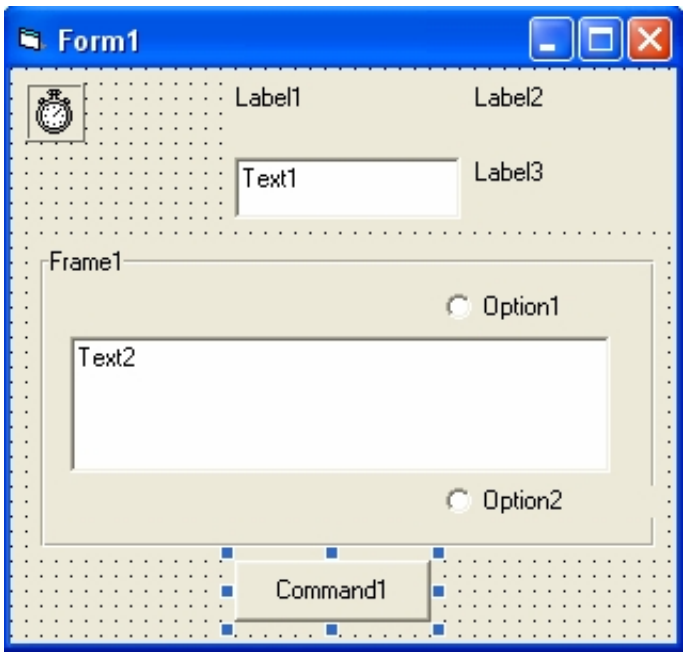
عمل منبه باستخدام عداد الوقت Timer

يبين لك البرنامج التالي كيفية الاستفادة من عداد الوقت بحيث يمكنك استعماله كمنبه حيث ستقوم بضبط وقت التنبيه وكذلك ستحدد كيفية تنفيذ التنبيه اما بعرض رسالة او عن طريق تشغيل برنامج تطبيقي موجود في ويندوز.

انشاء المنبه باستعمال عداد الوقت Timer

1. اختر الامر New Project < File ثم OK.
2. انقر اداة التحكم تسميةLabel في مربع الادوات ، ثم انشئ تسمية في وسط النموذج في الاعلى.
3. انشئ تسمية اخرى في الجهة اليمنى للنموذج في الاعلى.
4. انشئ تسمية ثالثة تحت التسمية الثانية على النموذج.
5. انقر اداة مربع نص TextBox من مربع الادوات وانشئ مربع نص تحت التسمية الاولى أي يجب ان يكون في وسط النموذج.
6. انقر اداة التحكم عداد الوقت Timer وضعه على النموذج.
7. انقر اداة التحكم الاطر Frame من مربع الادوات وانشئ اطار كبير بحيث يتسع لمربع نص وزري اختيار، بالاسفل من الادوات السابقة.
8. انقر اداة زر اختيار OptionButton في مربع الادوات وانشئ زر اختيار داخل الاطار الذي صنعته بالاداة Frame .
9. انشئ زر اختيار اخر OptionButton وضعه داخل اداة الاطر Frame ايضا.
10. انقر اداة مربع النص TextBox وانشئ مربع نص داخل اداة الاطر ايضا. (ستستخدم لكاتبة رسالة التنبيه)
11. انقر اداة التحكم زر امر CommandButton في مربع الادوات وانشئ زر امر في اسفل النموذج بالوسط.

يجب ان يبدو نموذجك كما في الشكل (12-6).



شكل (12-6)

12.اضبط الخصائص التالية للبرنامج.

الكائن	الخاصية	الاعداد
Command1	Caption	"تشغيل"
Label1	Caption	فارغ
	Alignment	2-Center
	BoarderStyle	1-Fixed Single
Label2	Caption	"الوقت الحالي"
	Alignment	1-Right Justify
Label3	Caption	"وقت التنبيه"
	Alignment	1-Right Justify
Text1	Text	فارغ
	Alignment	2-Center
Text2	Text	فارغ
	Alignment	1-Right Justify
	RightToLeft	True
	MultiLine	True
Timer1	Interval	1000
	Enable	True
Frame1	Caption	"التنبيه بـ"
	RightToLeft	True
Option1	Caption	"رسالة نصية"
	Alignment	1-Right Justify
	RightToLeft	True
Option2	Caption	"تشغيل برنامج"
	Alignment	1-Right Justify

	RightToLeft	True
Form1	Caption	"المنبه"
	RightToLeft	True

تستخدم الخاصية RightToLeft في حالة كون البرنامج الذي تصممه باللغة العربية لذلك اجعل قيمتها True .

يجب ان يبدو نموذجك كما مبين في شكل (13-6) بعد ان قمت بضبط الخاصائص.

شكل (13-6)

13.انقر النموذج نقرا مزدوجا.

يظهر الاجراء الحدثي Form1_Load في اطار الشفيرة.

اكتب الجمل البرمجية التالية:

```
Text1.Text = Time
Option1.Value = True
```

الجملة البرمجية الاولى تعين قيمة الوقت الى مربع النص Text1 اثناء تحميل النموذج. حيث تستخدم الدالة Time لسحب الوقت الحالي من ساعة الكمبيوتر.

اما الجملة الثانية فتجعل زر لاختيار الاول أي Option1 هو المنتقى في بدء التشغيل وذلك بسبب جعل الخاصية Value له تساوي True أي في الحالة الطبيعية فان المنبه سوف ينبه برسالة لاننا جعلنا زر الاختيار الاول منتقى عن طريق الجملة البرمجية الثانية.

14. انقر الكائن Command1 في مربع السرد المنسدل في اطار الشفيرة ثم اكتب الجملة التالية :

```
Timer_Start = 1
```

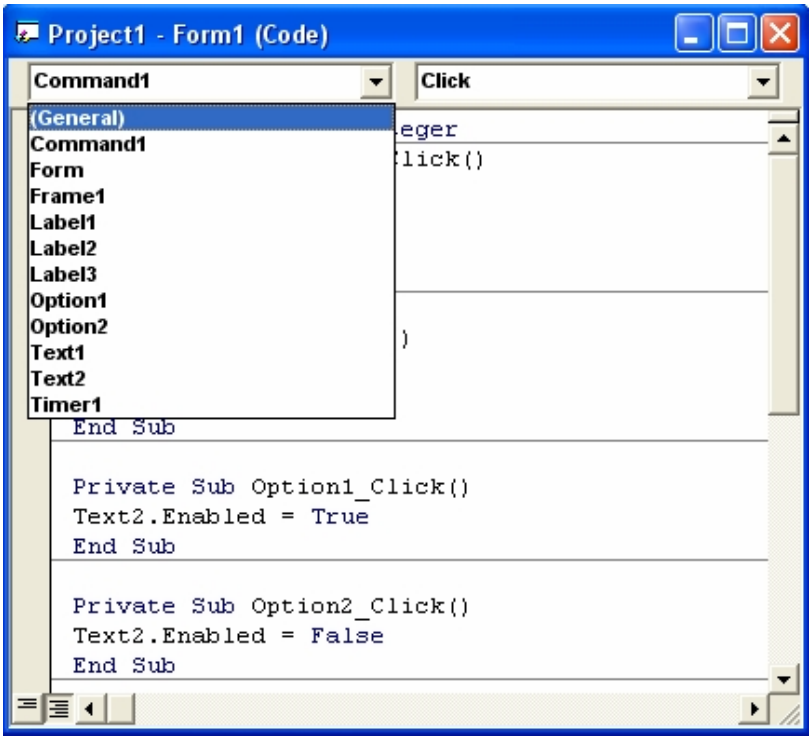
نقوم في هذه الجملة بتعيين قيمة 1 للمتغير Timer_Start وذلك لاعطاء امر باننا نريد تشغيل المنبه أي في حال انه قمنا بتعيين وقت التنبيه في مربع النص ولكننا لم نضغط على الزر "تشغيل" فان المنبه سوف لن يعمل حتى لو وصل وقت التنبيه المطلوب ، يتم تشغيل المنبه فقط بعد ان نقوم بتحديد وقت التنبيه بعد ان نضغط على زر "تشغيل" وذلك لكي يتم عمل المنبه. فعند الضغط على زر "تشغيل " سيعين فيجوال بيسك قيمة 1 للمتغير Timer_Start . حيث سنستعمله في ما بعد للتحقق من ان المستخدم قد ضغط على زر "تشغيل"

15. في الاجراء السابق قمنا بتحديد قيمة 1 للمتغير Timer_Start لكن تلك القيمة تعتبر 1 فقط داخل ذلك الاجراء الحدثي أي في الاجراء Command1_Click حيث سيموت المتغير Timer_Start في حالة خروجنا من ذلك الاجراء أي ان عمر المتغير مرتبط بعمر الاجراء الحدثي ، ولكننا نحتاج المتغير Timer_Start في اجراء حدثي اخر (في الاجراء Timer1_Timer) ولكنه سيصبح 0 في حال استدعائه في اجراء اخر وذلك يعني ان المستخدم لم يضغط على زر تشغيل !؟ . مما يعني عدم اشتغال المنبه!. لحل مثل هذه المشكلة أي جعل الكائن يحافظ على قيمته في كل الاجرءات الحدثية أي على مستوى النموذج ولا ترتبط نهاية عمره بنهاية الاجراء الذي هو فيه ، بل ترتبط بنهاية النموذج او البرنامج او تنتهي اذا قام المستخدم بانهائه. نقوم بتعريف المتغير في المنطقة العامة للنموذج وهي منطقة الـ (General) ، في حال تعريف متغير في هذه المنطقة فان المتغير سوف يعتبر عموميا لكل الاجراءات الحديثية التابعة لذلك النموذج.

16. انقر الكائن General في مربع السرد المنسدل في اطار الشفيرة ثم اكتب الجملة التالية:

```
Dim Timer_Start  As Integer
```

كما مبين في شكل (14-6)



شكل (14-6)

قمنا بتعريف المتغير Timer_Start كعدد صحيح.

17.انقر زر الاختيار الثاني "تشغيل برنامج" Option2 نقرا مزدوجا.

يظهر الاجراء الحدثي Option2_Click في اطار الشفيرة .

اكتب الجملة البرمجية التالية:

Text2.Enabled = False

قمنا هنا بتعطيل الخاصية Enable للكائن مربع النص الثاني(الذي يستخدم لكتابة رسالة التنبيه التي سوف تظهر للمستخدم في حال كون التنبية كان بواسطة رسالة).

وذلك لانه عند اختيار الاختيار الثاني أي التنبيه **بتشغيل برنامج** فانه لاحاجة من ان تجعل المستخدم يكتب رسالة في مربع النص لانه سوف ينبه بواسطة تشغيل برنامج.

لذلك قمنا بضبط الخاصية Enable عند False أي لاتستطيع الكتابة في مربع النص الثاني.

18.انقر زر الاختيار الاول "رسالة نصية" Option1 نقرا مزدوجا.

يظهر الاجراء الحدثي Option1_Click في اطار الشفيرة.

اكتب الجملة البرمجية التالية:

Text2.Enabled = True

قمنا هنا بارجاع الخاصية تمكين "Enable" للكائن مربع النص الثاني لان المستخدم اختار طريقة التنبيه بواسطة **رسالة نصية** لذلك هو بحاجة لان يكتب نص الرسالة في مربع النص .

19.انقر عداد الوقت Timer1 في النموذج نقرا مزدوجا .
يظهر الاجراء Timer1_Timer في اطار الشفيرة. اكتب الجمل التالية:

```
Label1.Caption = Time

If  Timer_Start = 1 Then

If  Time = Text1.Text Then

If Option1.Value = True Then
MsgBox Text2.Text

Else
Shell "C:\Windows\System32\Calc.exe"
End If
End If
End If
```

بعد ان جعلت قيمة الـ Interval للعداد تساوي 1000 أي ان العداد سوف يكرر العمليات البرمجية المكتوبة داخله في اطار الشفيرة كل ثانية ، يعين السطر الاول الوقت الى الخاصية Caption للكائن تسمية Label1 كل ثانية ، أي ستبدو كساعة رقمية ،
اما السطر الثاني فانه يقارن المتغير Timer_Start فاذا كان 1 فذلك يعني ان المستخدم قد شغل المنبه واذا لم يكن 1 فانه سوف ينهي البرنامج أي يذهب الى اخر End If في الاجراء الحدثي ، في حال كون المتغر يحمل القيمة 1 فان الجملة الشرطية سوف تتحقق وسيقوم البرنامج بتنفيذ الجمل المحصورة بين الجملة If و End If ، السطر الثالث يحتوي على جملة شرطية ايضا وهي مقارنة الوقت الحالي مع وقت التنبيه ، فالوقت الحالي يتم معرفته عن طريق الدالة Time التي تعيد الوقت الحالي من ساعة الكمبيوتر ، ووقت التنبيه يتم تعينه من قبل المستخدم وتوضع قيمته في مربع النص الاول Text1 ، لذلك تعمل الجملة الشرطية على استخدام عامل المقارنة = للمقارنة بين الوقت الحالي Time و وقت التنبه ، فاذا تساويا فذلك يعني انه وصل الى الوقت المطلوب ويجب ان ينبه المستخدم اما عن طريق رسالة نصية او عن طريق تشغيل برنامج،

حيث توجد جملة شرطية ثالثة وهي لاختبار نوع التنبية أي تنبيه بواسطة رسالة ام بواسطة برنامج. فاذا كانت قيمة الخاصية Value للكائن Option1 تساوي True فذلك يعني ان المستخدم قد اختار التنبيه بواسطة رسالة ، حيث يتم عرض الرسالة على المستخدم وذلك عن طريق الدالة MsgBox التي تعرض محتويات مربع النص الثاني الخاص بالرسائل.

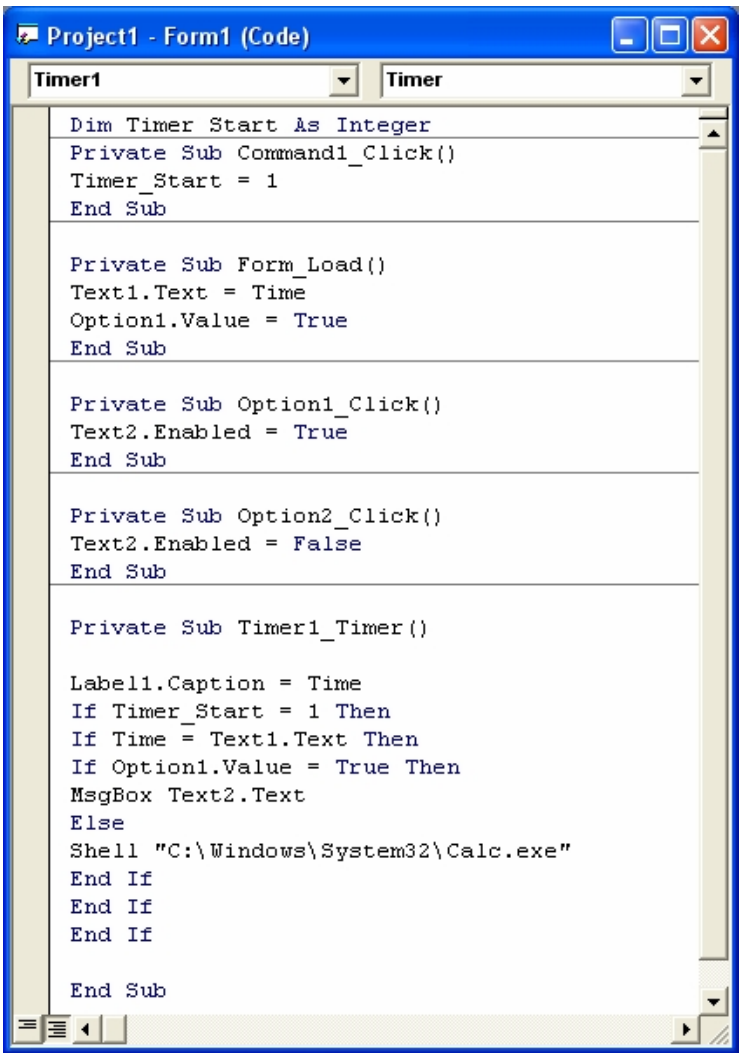
اما في حال لم تكن قيمة الخاصية Value تساوي True فذلك يعني ان المستخدم قد اختار التنبيه بواسطة تشغيل برنامج.

يتم تشغيل البرامج التطبيقية عن طريق استخدام الايعاز Shell حيث يستخدم هذا الايعاز لتشغيل برامج من داخل فيجوال بيسك، حيث تكون صيغة الايعاز Shell هي:

Shell "عنوان البرنامج بالكامل مع اسم البرنامج وامتداده"

كما مبين فوق فاننا نقوم بتشغيل حاسبة وندوز .

يجب ان يبدو اطار الشفيرة كما مبين في الشكل (6-15)



شكل (6-15)

20.اغلق اطار الشفيرة .

21.انقر زر التشغيل

يشتغل البرنامج وتبدا الساعة بالعمل حيث سوف ترى في وقت التنبيه ان تم وضع الوقت لحظة اشتغال البرنامج. قم بادخال وقت للتنبيه وليكن قريبا من الوقت الحالي للساعة (حتى لاتنتظر طويلا لترى عمل البرنامج). بعد ذلك اختر نوع التنبيه ثم انقر على الزر "تشغيل" وانتظر الى ان يتساوى الوقت الحالي مع وقت التنبيه حيث سترى رسالة التنبيه او برنامج الحاسبة سوف يشتغل. كما مبين في شكل (16-6)

شكل (16-6) التنبيه بواسطة رسالة نصية

حيث ستعرض الرسالة النصية في وقت التنبيه كما مبين في الشكل التالي:

22.قم بتعيين وقت تنبيه اخر واختر التنبيه بواسطة تشغيل برنامج وانقر على زر تشغيل وانتظر حتى يحين وقت التنبيه حيث سترى ان حاسبة ويندوز ستعمل عندما يحين وقت التنبيه.

23.انقر زر ايقاف البرنامج لايقاف النموذج عن العمل

24.احفظ المشروع والنموذج.

مكتبات VB وVBA

يوفر لك Visual Basic مئات الاجراءات والكائنات المضمنة في مكتبات VB وVBA والتي لا غنى عنها في برامجك الجدية، صحيح انك تستطيع محاكاة معظم هذه الدوال بكتابة اكواد لانجازها، الا ان استخدام الاجراءت والدوال المشمولة في مكتبات Visual Basic يعتبر افضل بكثير من انجازها بنفسك من منظور تحسين الكفاءة Optimization ، فهذه الدوال صممها مبرمجون محترفون بلغات اخرى مما يجعل تنفيذها اسرع بكثير من اكوادك المكتوبة ب Visual Basic . يأخذك هذا الفصل في جولة مع العشرات من هذه الدوال والاجراءت والتي سأتطرق اليها باختصار.

التعامل مع الاعداد

يوفر لك Visual Basic عشرات المعاملات Operators والدوال الخاصة بالاعداد باختلاف انواعها Long ، Integer، .... Byteالخ، بالاضافة الى دوال رياضية كدوال المثلثات او الدوال الاسية.

المعاملات الرياضية

يوفر لك Visual Basic المعاملات الاربع الرئيسية (+ ، – ، * ، /) وفي حالة تطبيقها على انواع مختلفة من القيم، فان القيم الابسط ستتحول مؤقتا الى الاعقد Integer الى Long و Singleالى Double ، بالنسبة لمعامل القسمة / فهو يقوم بتحويل جميع القيم المتمثلة في الحدين الايمن والايسر الى النوع Double ، لذلك ينصح باستخدام معامل القسمة الصحيحة \ مع المتغيرات الصحيحة Integer ،Byte و Long فهو اسرع اربع مرات من المعامل:/

```
Dim X As Long, Y As Long, Z As Long
Z = X / Y
Z  = X \ Y  ' هذا اسرع
```

كذلك معامل الاس ^ فهو يحول جميع القيم الى النوع Double ، وفي احيان كثيرة لن تحتاج الا للمتغيرات الصحيحة، لذلك ينصح باستخدام معامل الضرب عوضا عن الاس:

```
Dim X As Long, Y As Long,
Y = X ^ 3
Y = X * X * X
```

اما معامل باقي القسمة MOD فيقوم بتحويل القيم الى Long مما لا يعطي دقة في التعامل مع انواع الفاصلة العائمة الاخرى كـ Double و Single ، تستطيع تطوير دالة اخرى تعود بباقي القسمة للاعداد غيرLong:

```
Function ModEx (dN As Double, dD As Double) As Double
ModEx = dN – Int(dN / dD) * dD
End Function
```

بالنسبة للمعاملات المنطقية، فيوفر Visual Basic ست معاملات منطقية هي (= ، > ، < ، >= ، <= ، <>)بالنسبة لمعامل المساواة = فهو ليس كمعامل اسناد القيم الى المتغيرات، فمعامل المساواة = هو المعامل الذي يطبقهVisual Basic في داخل الجمل الشرطية او حتى اذا سبقه معامل اسناد آخر، فالكود التالي:

```
Dim X As Integer, Y As Integer
X = Y = 10
```

يبين لنا ان المعامل = الثاني الموجود في السطر الثاني هو معامل مقارنة المساواة وليس اسناد القيم.

ملاحظة :تفرق معظم لغات البرمجة الاخرى بين معامل المساواة ومعامل اسناد القيم، فنجد في لغة ال C معامل المساواة هو = = ومعامل اسناد القيم هو= .

اما معاملات الربط XOR ، AND و NOT فهي مدعومة ايضا لربط الجمل المنطقية ويمكنك استخدامها للاعداد حيث تؤثر على البتات Bits التي تمثل قيمة العدد بالنظام الثنائيBinary.

اولوية العوامل

يتيح لك فيجوال بيسك وضع قدر ما تشاء من العوامل الرياضية في صيغة واحدة طالما كان هناك عامل واحد بين كل متغير رقمي وتعبير رقمي. مثلا ، هذه صيغة مقبولة في فيجوال بيسك:

```
Total = 10 + 15 * 2 / 4 ^ 2
```

تعالج هذه الصيغة عدة قيم وتعين النتيجة الى متغير اسمه Total. لكن كيف يتم تقييم تعبير كهذا في فيجوال بيسك؟ بمعنى اخر، ما هي العوامل الرياضية التي يقيمها فيجوال بيسك اولا عند حل الصيغة؟ ربما لم تلاحظ ذلك ، لكن ترتيب التقييم له تاثير كبير في هذا المثال. يحل فيجوال بيسك هذه المعضلة باعتماده ترتيبا معينا لاولويات العوامل الرياضية. لائحة القواعد تلك تخبر فيجوال بيسك عن العوامل التي يجب ان يستعملها اولا عند تقييمه تعبيرا

يحتوي على اكثر من عامل واحد . يسرد الجدول التالي كل العوامل حسب الترتيب الذي سيتم تقييمها فيه (يتم تقييم العوامل الموجودة في المستوى نفسه في هذا الجدول من اليسار الى اليمين مثلما تبدو في التعبير).

العامل (العوامل)	ترتيب الاولوية
( )	يتم تقييم القيم الموجودة بين الاقواس اولا دائما
^	ياتي الأس (رفع عدد الى قوة) ثانيا
-	ياتي انشاء عدد سالب ثالثا (الاشارة)
/ *	ياتي الضرب والقسمة رابعا
\	تاتي القسمة العددية الصحيح خامسا
Mod	تاتي باقي القسمة سادسا
+ -	ياتي الجمع والطرح اخيرا

انطلاقا من ترتيب الاولويات المبين هنا سيعالج فيجوال بيسك التعبير التالي

Total = 10 + 15 * 2 / 4 ^ 2

باستعمال الخطوات التالية:

Total = 10 + 15 * 2 / 4 ^ 2

Total = 10 + 15 * 2 / 16

Total = 10 + 30 / 16

Total = 10 + 1.875

Total = 11.875

اسبقية المعاملات:

من المفيد ان اذكر هنا، ان من الاخطاء الشائعة التي يقع فيها اغلب المبرمجين هو نسيان ان اسبقية المعامل And اعلى من المعامل Or ، فمقارنة المعامل And تتم قبل مقارنة المعامل Or حتى ولو كان المعامل Or هو السابق اي في الجهة اليسرى قبل المعامل And ،ففي هذا المثال:

Print True Or False And False

للوهلة الاولى يعتقد المبرمج ان النتيجة هي False بينما النتيجة الحقيقية هي True وذلك، لان المعامل الشرطي And يتم اختبار ه قبل المعامل الشرطيOr. ولتجنب ذلك، استخدم الاقواس:

Print (True Or False) And False

المزيد ايضا، اسبقية المعامل Not اعلى من اسبقية المعامل And اي يتم تنفيذه دائما قبل معامل Andفالعبارة:

Print Not True And False

ستقوم بتنفيذ المعامل Not على كلمة True الاولى فقط حتى تكون False And False وبعد ذلك يأتي دور المعامل And وكما هو واضح فنتيجة التعبير هيFalse.

اما اذا اردت تنفيذ المعامل And ومن ثم عكس النتيجة فتستطيع ان تستخدم الاقواس والتي لها الأسبقية الأولى على جميع المعاملات مثل:

Print Not (True And False)

وفي هذه الحالة، سيكون الناتج النهائي هوTrue.

ما هي الدوال

الدوال هي أسماء محجوزة ومعرفة من قبل فيجوال بيسك لتقوم بعمل معين مثل المصفوفات والقيم المطلقة وغيرها..

أو بمعني ابسط واعم هي عبارة عن برنامج صغير مكتوب مسبقا و محفوظ داخل لغة الفيجول بيسك يمكن استدعاءه من خلال برنامجك والاستفادة منه ، إضافة إلى ذلك يمكنك أنت أن تكتب دالة وتستدعيها أكثر من مرة داخل برنامجك . والدوال في الفيجول بيسك تنقسم إلى:

1. الدوال الرياضيةMathematics Functions .
2. دوال سلاسل البياناتString Functions .
3. دوال الوقت و التاريخ Date and Time Functions
4. دوال التحقق من أنواع البيانات Data Type Inspection Functions
5. دوال المدخلات و المخرجات Program Output and User Input Functions
6. دوال ماليةFinancial Functions
7. و دوال متنوعة أخرى
8. دوال معرفة من قبل المبرمج(UDF) User Defined Functions .

ونلاحظ أن نوع الدوال من رقم (1) حتى رقم (7) أنها دوال داخلية ضمن اللغة، أي تأتي مُعرفة في لغة الفيجول بيسك و مبرمجة مسبقاً لكي تستخدمها مباشرةً. أما النوع الأخير فيتم تعريفه بواسطة المبرمج.

الدوال الرياضية

من الدوال الرياضية التي يوفرها لك Visual Basic دالة القيمة المطلقة Abs ودالة الجذر التربيعي Sqr والدالة الاسية Exp ، اما الدالة Sgn فهي تعود بالقيم 1 اذا كان العدد المرسل لها موجب، والقيمة 1 - اذا كان العدد المرسل لها سالب، وصفر اذا كان العدد المرسل صفر.

بالنسبة لدالة اللوغارتم Log فهي تعود باللوغارتم الطبيعي للعدد، اما للاعداد الاخرى، فتستطيع تطوير هذه الدالة:

```
Function LogEx (dN As Double, dBase As Double ) As Double
LogEx = Log ( dN ) / Log ( dBase )
End Function
```

كما انه يمكنك تطوير دالة الجذر التربيعي Sqr ايضا، حيث يمكن ان تعود بالجذر النوني للعدد:

```
Function NthSqr (iNum As Integer, iRoot As Integer) As Double
NthSqr = iNum ^ (1 / iRoot)
End Function
```

مثال الحصول على الجذر التكعيبي' لعدد' 8

```
Print NthSqr(8,3) ' 2 ستعطي الناتج
```

اخيرا الدوال المثلثية Tan ، Cos، Sinو Atn التي تعود بالقيمة المناسبة استنادا الى الزاوية المرسلة لها بالراديان (Rad) ، اما بالنسبة للدوال المثلثية الاخرى ،Sec , Cosec، .... الخ فيمكنك اشتقاقها بتطبيقات معادلاتها المعروفة،

هذه دالة الـ Sec

```
Function Sec (X As Double) As Double
Sec(X) = 1 / Cos(X)
End Function
```

وهذه دالة الـ Cosec

```
Function Cosec( X as Double) As Double
Cosec(X) = 1 / Sin(X)
End Function
```

تنسيق الاعداد

من اقوى دوال التنسيق هي دالة Format التي توفر لك خيارات لا نهائية لتنسيق الاعداد، الحروف، الوقت والتاريخ ايضا، ساسرد لك في هذه الفقرة طرق تنسيق الاعداد فقط.

الصيغة المبسطة للدالة Format تتطلب العبارة- او القيمة -و طريقة التنسيق:

Format (طريقة التنسيق, القيمة)

يوجد نوعان من طرق التنسيق . النوع الاول هو التنسيقات القياسية والثاني هو التنسيقات الخاصة . التنسيقات القياسية عبارة عن قيم نحدد نوع تنسيق الارقام كـ General Number لتنسيق الرقم بشكل عام او Currency لتنسيق الرقم على شكل عملة وغيرها من القيم التي تجدها في مكتبةMSDN:

```
Print Format(1234567, "General Number")   >>>  1234567
Print Format(1234567, "Currency”)         >>>>> $ 1,234,567.00
Print Format(1234567, "Standard”)         >>>>>  1,234,567
```

اما التنسيقات الخاصة فهي تنسيقات تحددها بنفسك . والتي تستخدم علامات كـ #،% ، .... 0 الخ، تجدها ايضا في مكتبةMSDN:

```
Print Format(1234.567, "#,##.00")  >>>> 1, 234.57

Print Format(0.234, "#.#%")      >>>>  23.4%

Print Format(20, "00#.00")       >>>>>  020.0
```

دوال اخرى

- من دوال حذف الفواصل الدالتين Int و Fix ، الاولى تحذف الفاصلة وتحول العدد الى عدد صحيح اقل من او يساوي العدد المرسل بينما الثانية تحذف الفاصلة فقط:

```
Print Int(1.2)        ' 1
Print Int(-1.2)       ' -2
Print Fix(1.2)        ' 1
Print Fix(-1.2)       ' -1
```

- اما دالة التقريب Round فقد ظهرت في الاصدار VB6 التي تمكنك من تحديد عدد الارقام العشرية:

```
Print Round(2.12567, 2)      ' 2.13
```

- وعند الحديث عن الاعداد الستعشرية Hexadecimal والثمانية Octal فان الدالتين Hex و Oct تحول اعداد النظام العشري الى الانظمة السابقة:

```
Print Hex$(100)    ' 64
Print Oct$(100)    ' 144
```

- اما الدالة Rnd فهي تعود بقيمة عشوائية اصغر من 1 واكبر من او تساوي صفر، تستطيع تخصيص مجال معين من الاعداد باستخدام هذه المعادلة:

فللحصول على اعداد عشوائية في المجال(5 - ، 2)اكتب شيئا مثل:

```
Print Int(8 * Rnd + -5)
```

- دوال تحويل القيم الى اعداد لعل اشهرها Val ، لكن المفضل استخدام دوال التحويل التي تمكنك من تحد يد نوع القيمة كـ CInt للاعداد Integer وCLng للاعداد Long، و CDbl للاعداد .... Double الخ.
- الدالة : Exp تستخدم هذه الدالة في تحديد القيمة (e) وهي قاعدة اللوغاريتم الطبيعي مرفوعة بقوة الرقم الذي تتضمنه حيث (e) تساوي تقريبا 2.7182818 وتأخذ الصورة العامة التالية:

```
MyNumber=Exp (رقم)
```

- الدالة : CInt وهي تشبه الدالة Round مع اختلاف بسيط وهو لا يمكنك تحديد عدد الأرقام التي ستقرب إليه بعد العلامة العشرية، وتأخذ الصورة التالية:

```
MyNumber= CInt(4.52696)
```

حيث 4.52696 الرقم المراد تقريبه وسيكون الناتج عدد صحيح مقرب 5.

كذلك الحال بالنسبة للدالتان CLng حيث تستخدم لتقريب الاعداد من نوع Long و Cdbl حيث تستخدم الاخرى لتقريب الاعداد من نوع Double.

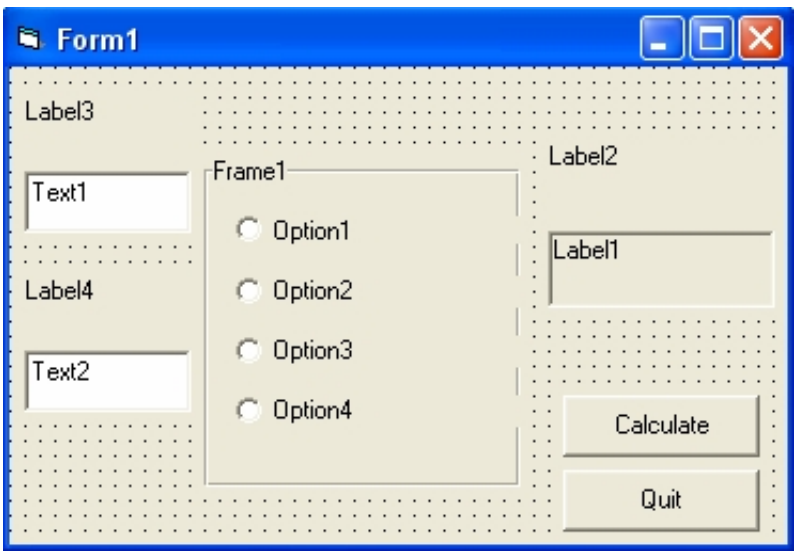
- الدالة : CSng وتحول هذه الدالة نوع بيانات متغيرها إلي عدد أحادي الدقة Single .
- الدالة : CVar وتعمل هذه الدالة علي تحويل بيانات متغيرها إلي النوع Variant .

الرياضيات الاساسية : + و – و / و *

ان عوامل الجمع والطرح والضرب والقسمة واضحة ويمكن استعمالها في أي صيغة تحتوي على ارقام او على متغيرات رقمية. يوضح التمرين التالي كيف يمكنك استعمالها في برنامج.

العمل مع العوامل الاساسية

1. اختر الامر New Project < File ثم OK لانشاء ملف تنفيذي قياسي.
2. قم باضافة زري امر الى النموذج.
3. اضف اربع تسميات الى النموذج.
4. اضف مربعي نص.
5. اضف اداة الاطر "Frame" الى النموذج.
6. اضف اربع ازرار اختيار الى النموذج "ضعها داخل اداة الاطر"
7. حاول ان تجعل نموذجك يبدو كما في الشكل (1-5).



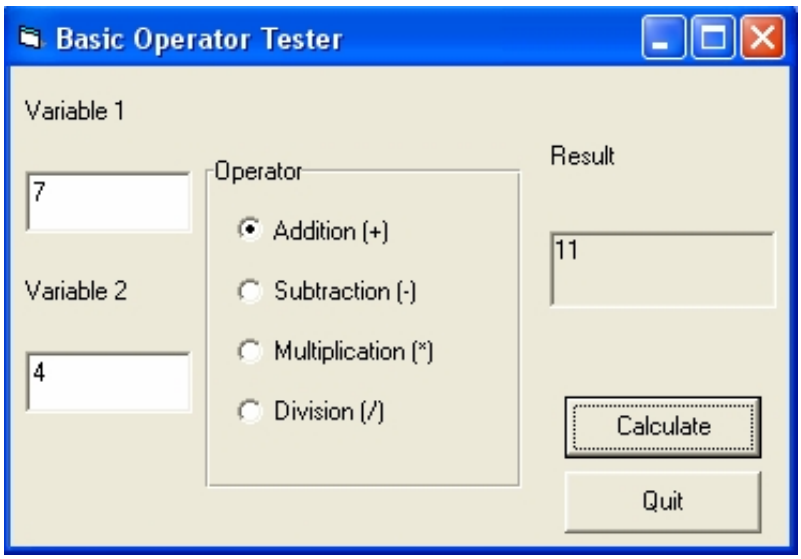
شكل (9-4)

8. اضبط خصائص النموذج وكائناته كما مبين في الجدول التالي.

الكائن	الخاصية	القيمة
Form1	Caption	Basic Operator Tester
Option1	Caption	Addition (+)
Option2	Caption	Subtraction (-)
Option3	Caption	Multiplication (*)

Division (/)	Caption	Option4
Operator	Caption	Frame1
Calculate	Caption	Command1
Quit	Caption	Command2
""	Caption	Label1
1-Fixed Single	BoarderStyle	
Result	Caption	Label2
Variable 1	Caption	Label3
Variable 2	Caption	Label4
""	Text	TextBox1
""	Text	TextBox2

9. يجب ان يبدو برنامجك كما في الشكل (4-10).



شكل (4-10)

10.انقر زر الامر Quit نقرا مزدوجا واكتب End في الاجراء الحدثي Command2_Click.

11.انقر زر الامر Calculate نقرا مزدوجا .

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة.

12.اكتب الجمل التالية لتصريح متغيرين FirstNo و SecondNo واستعمالهما.

يجب ان يبدو اطار الشفيرة كما في شكل (4-11)

```
Dim FirstNo, SecondNo
FirstNo = Val(Text1.Text)
SecondNo = Val(Text2.Text)
```

اذا كان اول زر اختيار قد اختير فاجمع العددين

```
If Option1.Value = True Then
Label1.Caption = FirstNo + SecondNo
End If
```

اذا كان ثاني زر اختيار قد اختير فاطرح العددين

```
If Option2.Value = True Then
Label1.Caption = FirstNo - SecondNo
End If
```

اذا كان ثالث زر اختيار قد اختير فاضرب العددين

```
If Option3.Value = True Then
Label1.Caption = FirstNo * SecondNo
End If
```

اذا كان رابع زر اختيار قد اختير فاطرح العددين

```
If Option4.Value = True Then
Label1.Caption = FirstNo / SecondNo
End If
```

تصرح الجملة الاولى في الاجراء متغيرين للاستعمال العام من النوع Variant . يستعمل هذان المتغيران لتخزين القيم المكتوبة في مربعي النص. وستكون عندهما المرونة الكافية لمعالجة أي نوع من البيانات الرقمية تريد استعماله. تُحمل الجملتان التاليتان البيانات من مربعات النص الى المتغيرات وتحول السلاسل النصية الى ارقام باستعمال الدالة Val:

```
FirstNo = Val(Text1.Text)
SecondNo = Val(Text2.Text)
```

الدالة Val هي روتين خاص يحول وسيطة نصية الى قيمة رقمية . هذا التحويل ضروري لصحة سير عملية الجمع في هذا البرنامج. ان نوع البيانات الافتراضي الذي يعيده مربع النص هو نص . هذه ليست بمشكلة بالنسبة لثلاثة عوامل . تعمل العوامل (- و * و /) مع الارقام فقط، لذا عندما ينتقي المستخدم احد هذه العوامل في البرنامج، يحول فيجوال بيسك تلقائيا القيم المعادة الى المتغيرين FirstNo و SecondNo الى ارقام.

يعمل العامل + مع السلاسل النصية والارقام على حد سواء. بما ان نوع البيانات الافتراضي الذي يعيده مربع النص هو نص، سيعتبر فيجوال بيسك القيمتين FirstNo وSecondNo انهما نص عندما تستعمل العامل + . عندها سيدمج فيجوال بيسك ، او يضم القيمتين بدل جمعهما مع بعضهما حسابيا (مثلا ، "100" + "20" سيساوي "10020").

مهم

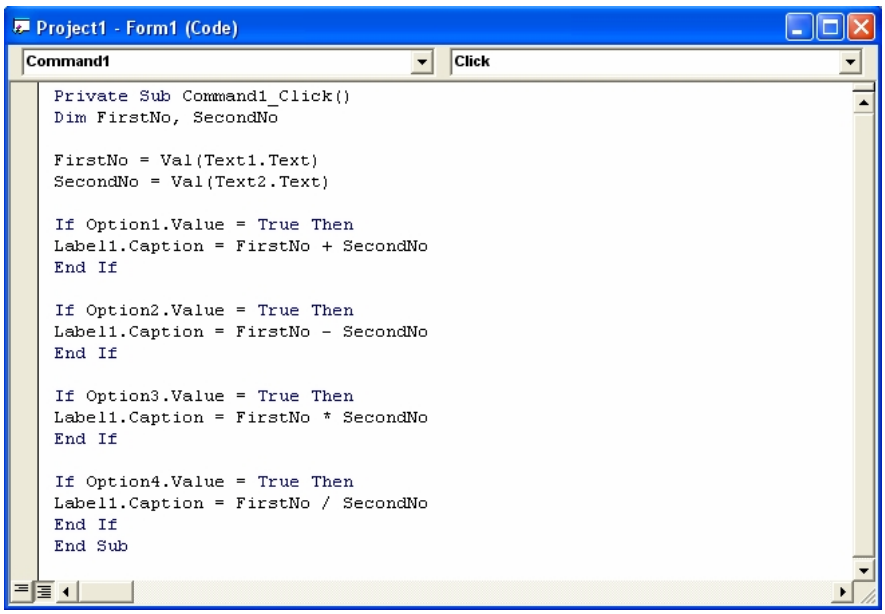
من المهم ان تفحص كل عملية حسابية في البرنامج لتتاكد من ان البرنامج يعمل كما يجب، اذ لايكفي فحص جزء واحد من البرنامج.

13. مرر نزولا في اطار الشفيرة وعاين الصيغ الاربع التي تستعمل العوامل الرياضية الاساسية.

تستعمل الصيغة الاولى في الاجراء عامل الجمع (+) وبنية القرار If … Then.

```
' if First Button Clicked add numbers
If Option1.Value = True Then
Label1.Caption = FirstNo + SecondNo
End If
```

اذا كانت الخاصية Value لزر الخيار الاول تساوي True (اذ تم نقر الزر)، يتم عندها جمع المتغيرين بواسطة العامل + وتُعين النتيجة الى التسمية. تملك الصيغ الثلاث المتبقية المنطق نفسه اذ يستعمل كل منها بنية القرار If … Then والخاصية Caption للكائن Label1 . ان بنيات القرار مثل If … Then مفيدة جدا في معرفة أي خيار انتقاه المستخدم عندما تتوفر امامه عدة خيارات.



شكل (11-4)

14.شغل البرنامج ، ثم قم باختبار البرنامج لترى نتيجة البرمجة.

15.اغلق اطار الشفيرة واحفظ البرنامج.

استعمال العوامل المتقدمة : \ و Mod و ^ و &

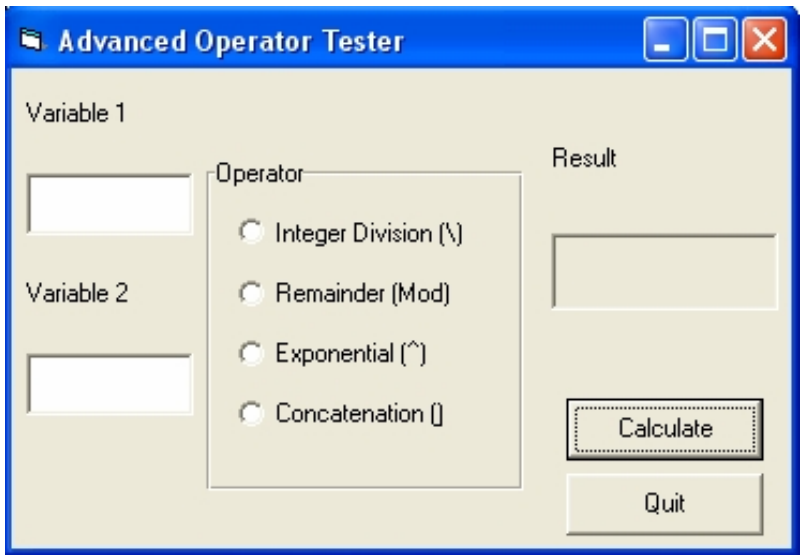
بالاضافة الى العوامل الرياضية الاساسية الاربعة ، يتضمن فيجوال بيسك اربعة عوامل متقدمة تؤدي قسمة العدد الصحيح (\) وباقي القسمة (Mod) و الاس (^) وضم السلاسل (&) . هذه العوامل مفيدة في الصيغ الرياضية ذات الاستعمال الخاص وفي برامج معالجة النصوص. يبين لك البرنامج التالي (هو تعديل طفيف على البرنامج السابق) كيف يمكنك استعمال كل واحد من هذه العوامل.

العمل مع العوامل المتقدمة

1. انشىء برنامج كما في الخطوات السابقة للتمرين السابق.
2. ستقوم بتغيير طفيف في الخواص كما مبين

الكائن	الخاصية	القيمة
Form1	Caption	Advanced Operator Tester
Option1	Caption	Integer Division(\)
Option2	Caption	Remainder (Mod)
Option3	Caption	Exponential (^)
Option4	Caption	Concatenation (&)

3. يجب ان يبدو نموذج كما في شكل (12-4)



شكل (12-4)

4. انقر الزر Calculate نقرا مزدوجا .

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة.

5. اكتب الجمل التالية لتصريح متغيرين FirstNo و SecondNo واستعمالهما.

يجب ان يبدو اطار الشفيرة كما في شكل(13-4)

```
Dim FirstNo, SecondNo
FirstNo = Text1.Text
SecondNo = Text2.Text
```

اذا كان اول زر اختيار قد اختير فاقسم العددين قسمة عددية صحيحة

```
If Option1.Value = True Then
Label1.Caption = FirstNo \ SecondNo
End If
```

اذا كان ثاني زر اختيار قد اختير فاعد باقي قسمة العددين

```
If Option2.Value = True Then
Label1.Caption = FirstNo Mod SecondNo
End If
```

اذا كان ثالث زر اختيار قد اختير فارفع العدد الاول للاس العدد الثاني

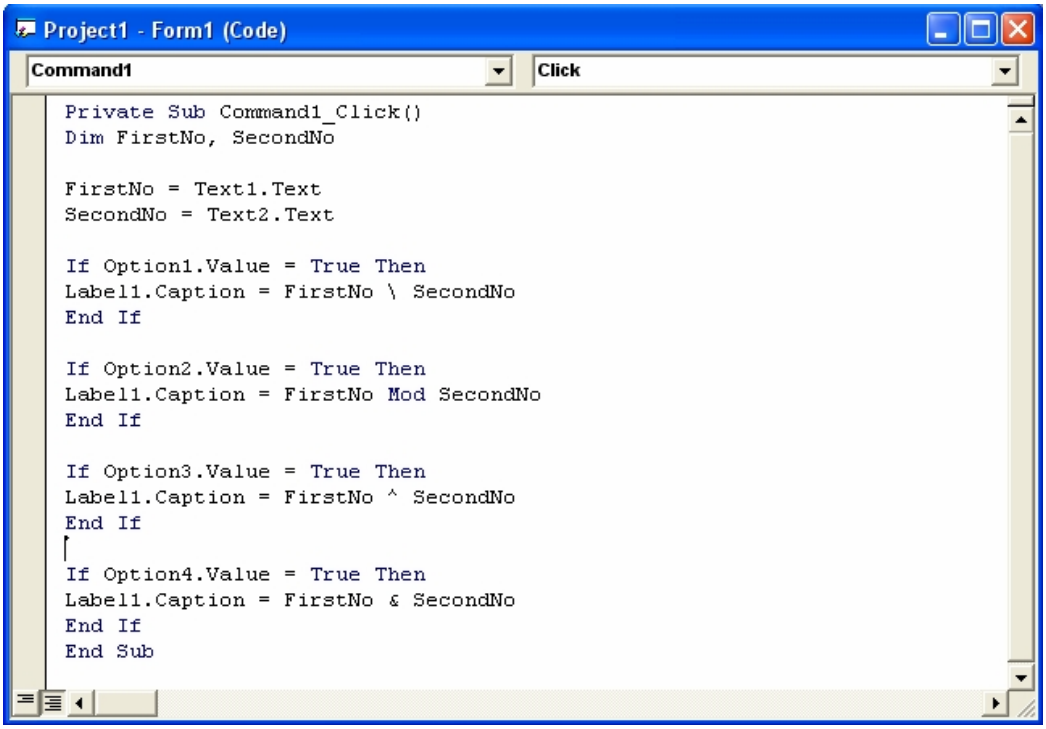
```
If Option3.Value = True Then
Label1.Caption = FirstNo ^ SecondNo
End If
```

اذا كان رابع زر اختيار قد اختيرَ فادمج العددين

```
If Option4.Value = True Then
Label1.Caption = FirstNo & SecondNo
End If
```

الاجراء Command1_click يشبه الاجراء Command1_Click في البرنامج السابق. تصرح الشفيرة متغيرين من النوع Variant وتعطيهما البيانات من مربعات النصوص وتحتسب قيمة الصيغة المنتقاة بواسطة بنية القرار If … Then.

لكن هناك فارق واحد مهم: هذا الاجراء الحدثي لا يستعمل الدالة Val لتحويل البيانات الى قمية رقمية عندما يقراها من مربعات النص. هذا التحويل ليس ضروريا للعوامل المتقدمة لانها ، على عكس العامل + ، تعمل مع نوع واحد من البيانات : تعمل \ و Mod و ^ مع ارقام فقط وتعمل & مع نصوص فقط، نظرا لعدم وجود غموض في النوع يستطيع المتغير المتنوع تحويل السلاسل المعادة من مربعات النص الى ارقام للعمليات الحسابية التي تتطلب ارقاما.



شكل (4-13)

6. انقر زر التشغيل في شريط الادوات.

يعرض البرنامج مربعي نص تكتب فيهما قيما رقمية، ومجموعة ازرار خيارات للعوامل ، ومربعاً لعرض النتائج وزري امر.

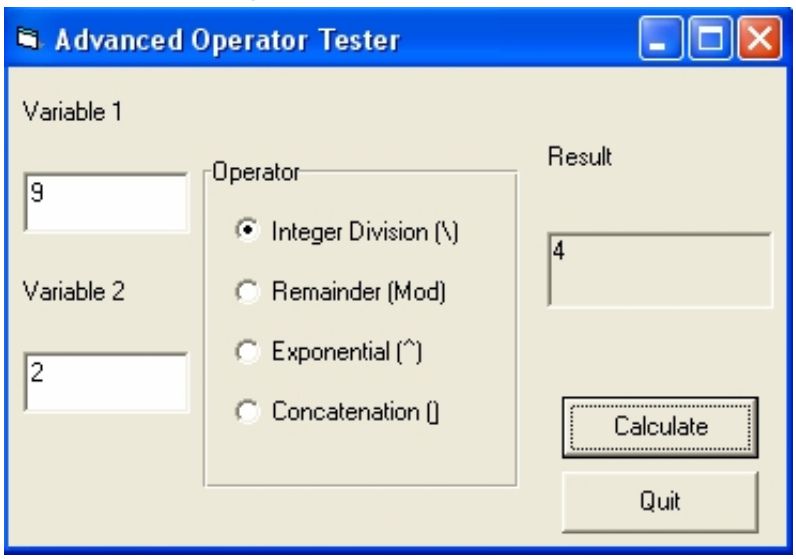
7. اكتب 9 في مربع النص Variable 1 ثم اضغط المفتاح <Tab

8. اكتب 2 في مربع النص Variable 2.

يمكنك الان تطبيق ايا من العوامل المتقدمة على القيم الموجودة في مربعي النص.

9. انقر زر الخيار Integer Division ثم الزر Calculate.

يتم تطبيق العامل على القيمتين ويظهر الرقم 4 في المربع Result، كما في شكل (4-14)



شكل(4-14)

تعطي القسمة العددية الصحيحة فقط العدد الكامل الناتج عن عملية القسمة. مع ان قسمة 9 على 2 تعطي 4.5 ، الا ان القسمة العددية الصحيحة تعطي فقط الجزء الصحيح الكامل (الرقم 4). قد تجد هذا الامر مفيدا اذا كنت تعمل على كميات ليس من السهل قسمتها الى اجزاء كسرية.

10.انقر زر الخيار Remainder ثم الزر Calculate.

يظهر الرقم 1 في مربع Result. يعيد عامل باقي القسمة الباقي (القسم المتبقي الذي لا يمكن قسمته) بعد قسمة العددين. بما ان قسمة 9 على 2 تعطي 4 ويكون الباقي 1 (2*4 +1 =9)، فان النتيجة التي يعطيها العامل Mod هي 1. يتيح لك العامل Mod تتبع "ما تبقى" من العمليات الحسابية.

11.انقر زر خيار Exponentiation ثم الزر Calculate.

يظهر الرقم 81 في المربع Result . يرفع عامل الاس (^) العدد 9 للاس 2 . بما ان 9^2 يساوي 81، فان النتيجة التي يعطيها العامل ^ هي 81 . في فيجوال بيسك ، يكتب 9^2 بهذا الشكل 9^2.

12.انقر زر الخيار Concatenation ثم الزر Calculate.

تظهر السلسلة "92" في المربع Result. عامل ضم السلاسل (&) يدمج السلسلتين المحددتين في الصيغة . لا تكون النتيجة ("92" في هذه الحالة) عددا ، بل تركيبة من الحرف "9" والحرف "2" . يمكن تنفيذ ضم السلاسل فقط على المتغيرات النصية والسلاسل التي تحيط بها علامات اقتباس والمتغيرات التي من النوع Variant . بما ان المتغيرات المستعملة في هذا البرنامج هي من النوع Variant ، يتم تحويلها تلقائيا الى نصوص لاتمام العملية.

13.انقر زر Quit لانهاء البرنامج.

التعامل مع الحروف

من منا لا يستخدم الحروف؟ توفر لك مكتبات VB و VBA عشرات الدوال المختصة في التعامل مع المتغيرات والثوابت الحرفية . Strings اعرض عليك في هذه الفكرة معظم هذه الدوال بالاضافة الى تطبيق فكرة البحث والاستبدالFind and Replace ولكني سأبدأ بالمعاملات الحرفية.

المعاملات الحرفية

الرمز & يمثل معامل الدمج Combine Operator للقيم الحرفية:

```
Dim sName As String
sName =" Allah"
sMyName = sName & " Is The Greatest "
Print sMyName   >>>> Allah Is The Greatest
```

اما معامل الجمع "+" فأنا لا احبذ استخدامه كثيرا، فاذا كان نوع القيم حرفية فسيتحول الى معامل الجمع"&" ،واذا كانت احدى القيم عددية والثانية حرفية قابلة للتحويل الى عددية فسيكون معامل جمع، اما اذا كانت احدى القيم عددية والاخرى حرفية لايمكن تحويلها الى عددية، فان رسالة الخطأ Type Mismatch لها نصيب من الظهور:

```
Print "20" + “30”     ‘ “2030”
Print "20" + 30        ‘ 50
Print "X" + 100   ' رسالة خطأ
```

بالنسبة للمعاملات المنطقية < ،<> ، .... = الخ فيمكن تطبيقها على القيم الحرفية ايضا، حيث تكون قيمة الحروف هي المقابل لها في جدول ASCII او UNICODE:

```
Print "Allah" > "ALLAH" ‘ True
Print "Allah " < "ALLAH " ‘ False
Print "Allah " = "ALLAH " ‘ False
```

ملاحظة :تستطيع ان تجعل Visual Basic يتجاهل مقارنة شكل الحروف الكبيرة والصغيرة عند استخدام معامل المساواة شريطة كتابة الكلمة المحجوزة Option Compare Text في منطقة الاعلانات العامة لكل وحدة برمجية.

البحث والاستبدال

تستطيع البحث عن كلمة او حروف معينة داخل قيمة حرفية عن طريق الدالةInStr التي تعود بموقع ذلك الحرف او بداية الكلمة:

```
Dim Position As Long
Dim StartPoint As Long
StartPoint = 1
Position = InStr (StartPoint, Text1.Text, " الكلمة التي تريد البحث عنها")
If Position > 0 Then
StartPoint = position + 1    ' هذه الجملة تعمل كما في حالة البحث عن التالي في البرامج الاخرى
Text1.SelStart = Position - 1
Text1.SelLength =  " عدد حروف الكلمة التي تبحث عليها"
Text1.SetFocus
End If
```

بالنسبة لعملية استبدال النصوص، فلن تجد اسرع من الدالة Replace التي تمكنك من استبدال حروف معينة بحروف اخرى . هنا سنستبدل جميع كلمات" محمد " الى "محمد صلى الله عليه وسلم "الموجودة في أداة النص:

```
Text1.Text = Replace(Text1.Text , "محمد" , "محمد صلى الله عليه وسلم" )
```

دوال اخرى

من الدوال الحرفية الاخرى دوال استخلاص الحروف اليسرى Left$ ، الحروف اليمنى Right$ و الحروف الوسطىMid:

```
Dim sMyString As String
sMyString = “ABCDEFGHIJKLMNOPQRSTUVWXYZ”
Print Left$ ( sMyString, 5) ‘ ABCDE
Print Right$ ( sMyString, 5) ‘ VWXYZ
Print Mid$ ( sMyString, 20, 5) ‘ TUVWX
Print Mid$ ( sMyString, 20) ‘ TUVWXYZ
```

وعند الحديث عن حروف لغتنا العربية، فمن المعروف ان الدالتين Right وLeft تعطيان نتائج عكسية مع الحروف العربية . فالاستدعاء Right للجملة" مرحبا" سبدأ من الالف فالباء فالحاء الخ .. مما يسبب التشويش على المبرمج العربي.

الفكرة بكل بساطة لجعل هاتين الدالتين تعملان بشكل صحيح مع الحروف العربية، هي عن طريق تطوير دالتين عربيتين ArRight وArLeft:

```
Function ArLeft(sString As String, lLength As Long) As String
ArLeft = Right$(sString, lLength)
End Function
Function ArRight(sString As String, lLength As Long) As String
ArRight = Left$(sString, lLength)
End Function
```

حيث ان مبد العمل هو عكس الدالتين أي استخدام دالة اليمين بدل دالة اليسار واستخدام دالة اليسار بدل دالة اليمين .

```
Private Sub Form_Click()
مثال للاستدعاء'
Print ArRight ("أحمد ابراهيم" , 4)  ' احمد
Print ArLeft ("أحمد ابراهيم" , 7)  ' ابراهيم
End Sub
```

اعود للدالة Mid مرة اخرى، فعليك معرفة ان Mid هي عبارة Statement ايضا، فيمكنك كتابة شيئا مثل:

```
Dim sMyString As String
sMyString = "abcde"
Mid (sMyString, 2, 3) = "BCD"
Print sMyString          ‘ “aBCDe”
```

وبدلا من معرفة ما اذا كان المتغير الحرفي خاليا باستخدام علامات التنصيص، استخدم الدالة Lenالتي تعود بعدد حروف القيمة المرسلة فهي اسرع مرتين من الطريقة الاولى:

```
If sMyString = "" Then …
If Len(sMyString) = 0 Then…..  هذه اسرع
```

اما لحذف المسافات اليمنى استخدم الدالة RTrim ، المسافات اليسرىLTrim والمسافات اليمنى واليسرى Trim:

```
sMyString = "  12345  "
Print RTrim$ (sMyString)      '   "  12345"
Print LTrim$ (sMyString)      ' "12345  ”
Print Trim$ (sMyString)       ' "12345"
```

الدالة Asc تعود بالعدد المقابل للحرف في جدول ASCII والدالة Chr تعود بالحرف المقابل للعدد– أي العكس:

```
Print Asc ( ب)  >>> 200
Print Chr(200)  >>> ب
```

من الدوال الاخرى دالتي UCase و LCase اللتان تقومان بتكبير الحروف الانجليزية وتصغيرها .استخدام هاتان الدالتان مسألة ضرورية خاصة عند مقارنة القيم الحرفية، فلا تنسى ان " Ahmed " لا تساوي "ahmed "

قد لا يكتب المستخدم حروف كبيرة If Text1.Text = "AHMED" Then

```
If UCase ( Text1.Text ) = "AHMED" Then …
If LCase ( Text1.Text ) = "ahmed" Then …
```

الدالة UCase تقوم بتكبير جميع الحروف والدالة LCase تقوم بتصغير جميع الحروف كما رأينا في المثال السابق، اما اذا كنت تريد تكبير الحرف الاول من كل كلمة، فارسل الثابت vbProperCase الى الدالة StrConv:

```
sMyString = "I like Visual Basic"
Print StrConv(sMyString, vbProperCase) ‘ “I Like Visual Basic”
Print StrConv(sMyString, vbLowerCase) ‘ “i like visual basic”
Print StrConv(sMyString, vbUpperCase) ‘ “I LIKE VISUAL BASIC”
```

التعامل مع الاحرف والنصوص

ستتعلم في هذا الدرس كيفية:

- ❖ البحث في النصوص والبحث عن التالي.
- ❖ استبدال نص معين بنص اخر.

ستتعلم في هذا الدرس كيفية التعامل مع الاحرف والنصوص وكيفية البحث عن حرف او كلمة داخل نص وايضا كيفية البحث عن التالي أي اذا توجد كلمة اخرى او اكثر من كلمة في النص تشبه الكلمة التي نبحث عنها ، أي كيفية الوصل الى كل كلمة . كذلك ستتعلم في هذا الدرس كيفية استبدال حرف (او كلمة) موجود داخل النص بحرف (او كلمة) اخر .

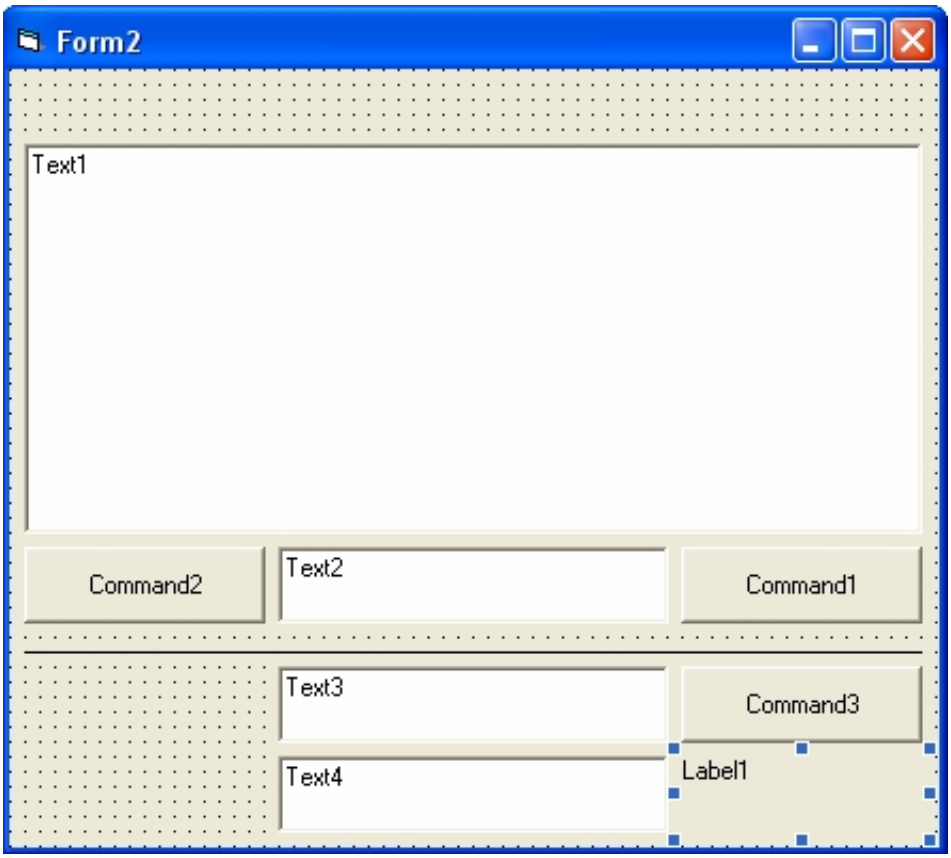
العمل مع النصوص : البرنامج Text Handler

يبين التمرين التالي كيفية التعامل مع النصوص وكيفية البحث والاستبدال فيها .

انشاء البرنامج Text Handler

1. اختر الامر New Project < File ثم OK.
2. انقر اداة التحكم مربع نص TextBox في مربع الادوات وانشئ مربع نص يحتل جزءا كبيرا من مساحة النموذج.
3. انقر اداة التحكم زر امر Command في مربع الادوات .
4. انشئ زر امر اسفل مربع النص من الجهة اليمنى للنموذج.
5. انشئ زر امر ثان في الجهة المقابلة للزر الاول.
6. انقر اداة التحكم مربع نص TextBox في مربع الادوات وانشئ مربع نص بين زري الامر.
7. انقر اداة التحكم خط Line لانشاء خط على النموذج ، انشئه على طول النموذج اسفل زري الامر ومربع النص.
8. انقر اداة التحكم زر امر Command في مربع الادوات ، انشئ زر امر ثالث اسفل زر الامر الاول أي سيكون في الجهة اليمنى للنموذج.
9. انقر اداة التحكم مربع النص ، وانشئ مربع نص ثالث بجانب زر الامر الثالث.
10. انشئ مربع نص رابع اخر اسفل مربع النص الثالث.
11. انقر اداة التحكم تسمية Label في مربع الادوات .
12. انشئ تسمية اسفل زر الامر الثالث.

يجب ان يبدو نموذجك كما في شكل (1-8).

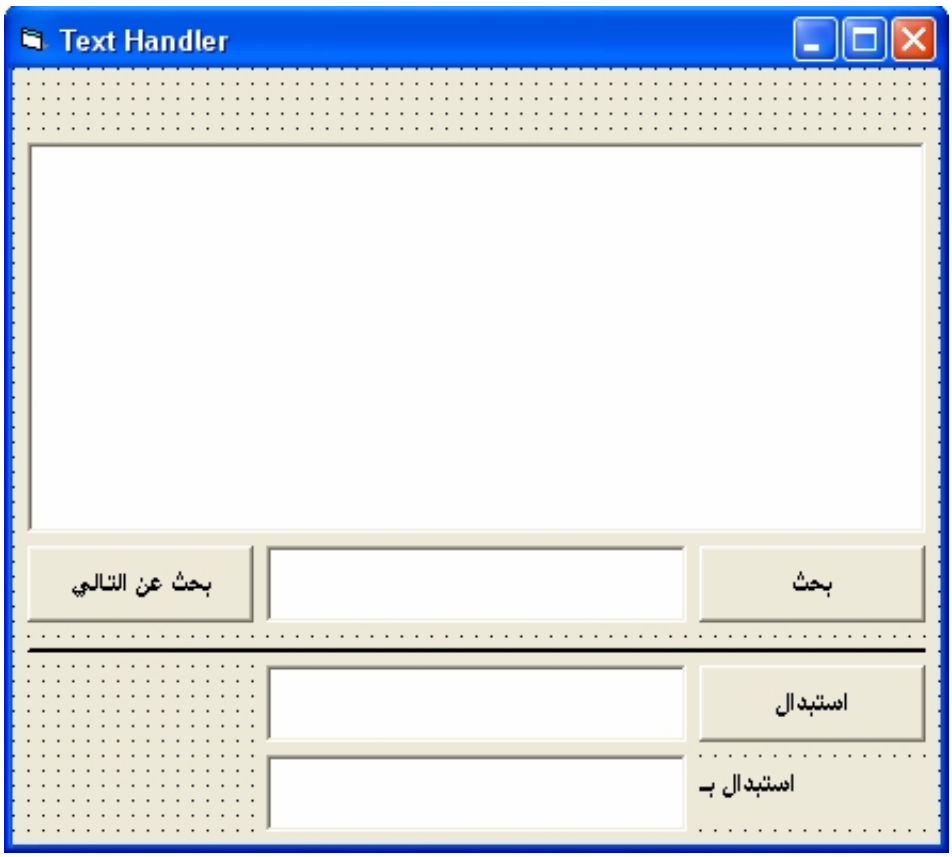


شكل (8-1)

13. اضبط خصائص الكائنات كما مبين في الجدول التالي:

الكائن	الخاصية	الاعداد
Form1	Caption	Text Handler
Text1	Text	فارغ
	MultiLine	True
Text2	Text	فارغ
Text3	Text	فارغ
Text4	Text	فارغ
Command1	Caption	"بحث"
Command2	Caption	"بحث عن التالي"
	Enable	True
Command3	Caption	"استبدال"

14.بعد ان قمت بضبط الخصائص يجب ان يبدو نموذجك كما في شكل (8-2).



شكل (2-8)

15. انقر زر الامر "بحث" نقرا مزدوجا.

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة:

16.اكتب الجمل البرمجية التالية:

```
Dim Position As Long

StartPoint = 1
Position = InStr(StartPoint, Text1.Text, Text2.Text)
If Position > 0 Then
Command2.Enabled = True
StartPoint = Position + 1
Text1.SelStart = Position - 1
Text1.SelLength = Len(Text2.Text)
Text1.SetFocus
Else
Command2.Enabled = False
MsgBox "انتهت عملية البحث ولم يتم العثور على المطلوب", , "بحث"

End If
```

الجملة الاولى تقوم بتعريف متغير Position من النوع Long . اما المتغير StartPoint فسنستخدم هذا المتغير لتحديد بداية البحث في النص أي من اين سيبدا فيجوال بيسك بالبحث، هنا سنقوم بالبحث من اول حرف في النص لذلك قمنا بتعيين الرقم 1 للمتغير StartPoint ، كذلك سنحتاج المتغير StartPoint في عملية البحث عن التالي لذلك سنحتاج ان نعرفه كمتغير عام على مستوى الوحدة أي سنقوم بتعريفه في المنطقة General .

يستخدم السطر الثالث الدالة InStr لتعين موقع الحرف (الكلمة) الى المتغير Position تستخدم الدالة InStr المتغير StartPoint وذلك لتحديد بداية البحث من اين ستكون ثم تستخدم النص الذي نريد البحث فيه وهنا نحن نريد البحث في مربع النص الاول الذي نستخدمه للكتابة لذلك كتبنا Text1.Text ، كذلك تحتاج هذه الدالة الكلمة (الحرف) التي نريد البحث عنها ونحن هنا سنكتب الكلمة التي نريد البحث عنها في مربع النص الثاني، لذلك كتبنا Text2.Text. ستقوم الدالة InStr بارجاع موقع الحرف او الكلمة الى المتغير Position لذلك في حال ايجاد الحرف او الكلمة سيكون المتغير Position يحمل رقم موقع الحرف (الكلمة) . اما في حالة عدم ايجاد الحرف (الكلمة) فان الدالة InStr ستعين الرقم 0 للمتغير Position لانها لم تجد الكلمة المطلوبة.

تقارن الجملة الشرطية If Position > 0 Then المتغير Position فاذا كان اكبر من 0 فذلك يعني انه قد وجد الحرف (الكلمة) المطلوبة. واذا كان 0 فذلك يعني انه لم يجد الكلمة. في حال تحقق الشرط نقوم بتمكين زر امر "بحث عن التالي" وذلك قد تكون هناك اكثر من كلمة تشبه الكلمة التي نبحث عنها.

بعد ذلك نقوم بتعين قيمة المتغير Position ونجمع معه 1 ونعينه الى المتغير StartPoint وذلك لاننا سنستخدم المتغير StartPoint في البحث عن التالي.

تعمل الاسطر الثلاثة الاخرى على تحديد الكلمة التي تم ايجادها ، تستخدم الخاصية Text1.SelStart لتحديد بداية النص الذي سنقوم بتحديده ، اما الخاصية Text1.SelLength تستخدم لتحديد عدد الاحرف التي سنقوم بتحديدها. اما الخاصية Text1.SetFocus فتستخدم لاعطاء التركيز للنص الذي تم تحديده أي للكلمة التي تم ايجادها.

اما في حالة لم يتحقق الشرط فان لمربع الرسالة نصيب في الظهور، حيث سنقوم بتعطيل تمكين زر البحث عن التالي وكذلك باظهار رسالة للمستخدم تخبره بعدم ايجاد الكلمة المطلوبة.

17.انقر الاجراء General في مربع السرد المنسدل في اطار الشفيرة لتصريح المتغير StartPoint كمتغير عام على مستوى النموذج، واكتب الجملة التالية:

```
Dim StartPoint As Long
```

18.انقر الكائن Command2 في مربع السرد المنسدل في اطار الشفيرة(سنبرمج زر البحث عن التالي) ثم اكتب الجمل التالية في الاجراء الحدثي Command2_Click:

```
Dim Position As Long
Position = InStr(StartPoint, Text1.Text, Text2.Text)
If Position > 0 Then
StartPoint = Position + 1
Text1.SelStart = Position - 1
Text1.SelLength = Len(Text2.Text)
Text1.SetFocus
Else
Command2.Enabled = False
MsgBox "انتهت عملية البحث ولم يتم العثور على المطلوب", , "بحث"
End If
```

لا اضن اني بحاجة الى ان اعيد شرح هذه الجمل لاني قد شرحتها في الخطوة السابقة ولكن ان امعنت النظر في الشفيرتين لوجد ان هناك اختلاف واحد وهو انني لم اقم بتعين الرقم 1 للمتغير StartPoint في هذا الاجراء وهذا هو سر البحث عن التالي حيث لو اني قمت بتعين القيمة 1 للمتغير StartPoint فان الدالة InStr ستقوم بالبحث مرة اخرى من اول حرف في النص وستتوقف عند اول كلمة تطابق الكلمة المطلوبة ، لذلك لن نستطيع الوصول الى بقية الكلمات ، لكن هنا قمت باستخدام قيمة المتغير StartPoint التي حصلت عليها من الشفيرة الاولى التي تكون قيمتها تساوي رقم موقع الكلمة المطلوبة زائد واحد أي ستبدا بالبحث مباشرة من بعد الكلمة التي تم ايجادها. وبقية الجمل كما تم شرحها.

19.انقر الكائن Command3 في مربع السرد المنسدل في اطار الشفيرة(سنبرمج زر الاستبدال) ثم اكتب الجملة التالية في الاجراء الحدثي Command3_Click:

Text1.Text = Replace(Text1.Text, Text3.Text, Text4.Text)

تستخدم الدالة Replace لاستبدال حروف او كلمات في نص بحروف او كلمات اخرى حيث تتطلب الدالة Replace في البداية النص الذي نريد استبدال الكلمات فيه ، هنا وضعنا Text1.Text أي سنقوم باستبدال بعض الكلمات الموجودة في مربع النص الاول وكذلك تطلب الدالة ايضا الكلمة التي سيتم استبدالها ، والتي سنقوم بكتابتها في مربع النص الثاني لذلك وضعنا Text3.Text ، واخيرا تحتاج هذه الدالة الى الكلمة التي سنقوم بوضعها مكان الكلمة التي نريد استبدالها لذلك وضعنا Text4.Text (أي مربع النص الذي اسمه "استبدال بـ ").

بعد ان تقوم الدالة بعملية الاستبدال سنقوم بتعين ناتجها الى مربع النص الاول وذلك لكي نرى الاستبدال انه قد تم بالفعل.

20. اغلق اطار الشفيرة ، ثم قم بتشغيل البرنامج من شريط الادوات.

يشتغل البرنامج Text Handler

21. اكتب نص معين في مربع النص الاول .

22. اكتب حرف او كلمة (ولتكن موجودة في النص) للبحث عنها في مربع النص الثاني.

23. انقر زر بحث سترى ان البرنامج سيقوم بتحديد الكلمة التي تم ايجادها وانه تم تمكين زر البحث عن التالي تحسبا لان توجد كلمة اخرى. انظر الشكل (3-8)

شكل(3-8)

24.اكتب أي حرف او كلمة موجودة في النص داخل مربع النص الثالث واكتب الكلمة او الحرف التي تريد استبدال الكلمة بها (اكتبها في حقل استبدال بـ). انقر زر استبدال سترى ان البرنامج سيقوم باستبدال الكلمة القديمة بالكلمة الجديدة . انظر الشكل (4-8) قمت باستبدال كل a بـ A .

شكل(4-8)

25.انقر زر ايقاف البرنامج

26.احفظ البرنامج باسم TextHandler وذلك لاستخدامه فيما بعد.

27.اغلق البرنامج.

- الدالة : SavePicture تعمل هذه الدالة على حفظ الصورة إلى ملف خارجي بأي امتداد تريده ومن أي أداة يمكنها احتواء صورة بداخلها وتأخذ الشكل التالي:

```
SavePicture Picture1.Picture,  "D:\aa.bmp"
```

- الدالة : LoadPicture تعمل هذه الدالة على (تحميل) الصورة من مكان تحدده أنت وتأخذ الصورة التالية:

```
Picture1.Picture = LoadPicture("D:\aa.bmp")
```

- التصريح : Randomize يعمل هذا التصريح مع الدالة Rnd ونستفيد منه عدم التكرارية و الحصول على عشوائية افضل وذلك لأنه يعتمد على ساعة النظام لديك . ويأخذ الصورة التالية.

```
Randomize
MsgBox Int((10 * Rnd) + 1)
```

وهنا نعمل علي توليد أرقام عشوائية من واحد إلى عشرة.

الالوان :

- **الألوان باستخدام: QBColor**

تضم هذه الطريقة خمسة عشر لوناً تبدأ من الصفر وحتى 15 ولكل رقم لون معين ، ويمكن الحصول على الألوان من هذه القائمة بالطريقة التالية:

```
Form1.BackColor = QBColor(Number)
```

حيث Number هو رقم بين 0 و 15.

- **الألوان باستخدام: RGB**

تضم هذه الطريقة الالوان الاساسية الثلاثة وهي الاحمر والاخضر والازرق حيث كل لون يمثل ببايت واحد أي الحد الاعلى لكل لون هو 255 لذلك نستطيع مزج هذه الالوان والحصول على لون معين من مزج قيم معينة.

```
Form1.BackColor = RGB(255 , 255 , 0)
```

يعطي هذا اللون الاصفر

- **الالوان باستخدام : اللوان فيجوال بيسك**

تضم هذه الطريقة الالوان المعرفة من قبل فيجوال بيسك مسبقا أي يمكنك كتابة اسم اللون مباشرة ، كما مبين:

```
Form1.BackColor = VBRed
```

سيعطي هذا اللون الاحمر وللازرق اكتب VBBlue وهكذا .

التعامل مع الوقت والتاريخ

لايقتصر دعم Visual Basic على قيم الوقت والتاريخ في امكانية تصريح متغير من النوع Date، بل يحتوي على عشرات الدوال الخاصة بعرض، تعديل، حساب وتنسيق قيم الوقت والتاريخ.

دوال الوقت والتاريخ

القيم من النوع Date هي قيم تمثل وقت معين او تاريخ معين او كلاهما معا سواء كانت في متغير من النوع Date او ثابت بين العلامتين # و#

```
Dim MyDate As Date
MyDate = #11/14/2008#
Print MyDate
MyDate = #3:10:00 PM#
Print MyDate
MyDate = #11/14/2008  3:10:00 PM #
Print MyDate
```

وقبل ان ابدأ في الحديث عن دوال الوقت والتاريخ، اود ان اعرفك على الخاصية Calendar التابعة للكائن VBA والتي تمكنك من تغيير نظام التاريخ التابع لبرنامجك الى ميلادي او هجري:

```
Dim MyDate As Date
VBA.Calendar = vbCalHijri '  هجري
Print MyDate
VBA.Calendar = vbCalGreg '  ميلادي
Print MyDate

MyBirth = #6/14/1984#
VBA.Calendar = vbCalHijri
MsgBox MyBirth  ' التاريخ هنا سوف يعرض بالهجري
```

والان اعرض لك دوال الوقت والتاريخ مبتدئا بالدالتين Date و Time اللتان تعودان بتاريخ اليوم والوقت الحالي:

```
Print Date
Print Time
```

اما اذا تعاملت مع Date و Time كعبارات Statement ، فهي ستغير قيمة الوقت والتاريخ في النظام:

```
Date = #1/20/2008#
Time = #12:00:00 AM#
```

الدالة Now تعود بقيمة تشمل تاريخ اليوم والوقت الحالي:

```
Print Now
```

اما الدالة Timer فهي تعود بعدد الثواني من منتصف الليل حتى وقت استدعائها أي هي تعمل كعداد، قد تستفيد منها وتطور اجراء انتظار مؤقت قبل تنفيذ كود معين:

```
Sub Wait ( iSeconds As Integer)
Dim sStartTime As Single
sStartTime = Timer
Do
DoEvents
Loop Until Timer – sStartTime >= iSeconds
End Sub
```

اما بالنسبة للدالتين DateValue و TimeValue فهما تعودان بقيمة التاريخ او الوقت الموجود في القيمة المرسلة اليهما:

```
Print DateValue(Now + 2)
Print TimeValue(Now)
```

بامكانك استخدام الدوال (Minute ، Hour، Year، Month، Day و Second) التي تعود بقيمة اليوم، الشهر، السنة، الساعة، الدقيقة والثانية الموجودة في القيمة المرسلة اليهم:

```
Print Month (Date)
Print Hour (#1:20:00 AM#)
```

كما اضاف الاصدار VB6 الدالة MonthName التي تعود باسم الشهر المقابل للرقم المرسل لها:

```
VBA.Calendar = vbCalHijri
Print MonthName(1)    ' محرم
VBA.Calendar = vbCalGreg
 Print MonthName(1)    ' يناير
```

التعامل مع الملفات والمجلدات

لم يضف الاصدار VB6 أي دوال او عبارات جديدة للتعامل مع الملفات والمجلدات، فمعظم ما سأسطره في الفقرات التالية توفر منذ الاصدارات القديمة ل . VisualBasic وسأبدأ بعرض دوال وعبارات تمكنك من التعامل مع الملفات كتعديل خصائصها، اسمائها، الاستعلام عن احجامها .... الخ.

التعامل مع الملفات

يمكنك الامر Name ... As من اعادة تسمية الملف او نقله، والامر FileCopy من نسخ الملف، بينما الامر Kill يحذف الملف:

```
اعادة تسمية ملف،
Name "D:\MyFile.EXT" As "D:\MyFile.DAT"
نقل ملف،
Name "C:\MyFile.EXT" As "D:\MyFile.EXT"
نسخ ملف،
FileCopy "C:\MyFile.EXT" As "D:\MyFile.EXT"
نسخ وتغيير اسم ملف،
FileCopy "C:\MyFile.EXT" As "C:\MyFile2.EXT"
حذف ملف،
Kill "D:\MyFile.EXT"
حذف مجموعة ملفات،
Kill "*.TMP"
```

- الدالة : MkDir تعمل هذه الدالة على إنشاء مجلد جديد في نفس الدليل الذي تحدده ويأخذ الصورة التالية:

```
MkDir "D:\Help"
```

- الدالة : RmDir وهذه الدالة هي عكس الدالة MkDir حيث تعمل على حذف المجلد وتأخذ الصورة التالية:

```
RmDir "D:\Help"
```

تعود الدالة FileLen بقيمة تمثل حجم الملف، والدالة FileDateTime بوقت وتاريخ انشاء الملف:

```
Print FileLen ( sFile )
Print FileDateTime ( sFile )
```

الميزة التي تعجبني في الدالة FileLen هي قابليتها على العودة بحجم الملفات حتى وان كانت مفتوحة، والعيب الذي لا يعجبني في نفس الدالة هو ان القيمة التي تعود بها هي حجم الملف قبل الفتح- أي لن تعود بقيمة صحيحة في حال قيام البرنامج بتغيير حجم الملف.

حيث تعطي حجم الملف بالبايت يمكنك استخدام هذه الدالة لتحويل حجم الملف من البايت الى الميكا بايت

```
Public Function BytesToMegabytes (Bytes As Double) As Double
  Dim dblAns As Double
  dblAns = (Bytes / 1024) / 1024
  BytesToMegabytes = Format(dblAns, "###,###,##0.00")

End Function


Public Function MegaToGegabytes(mega As Double) As Double
  Dim dblAns As Double
  dblAns = (mega / 1024)
  MegaToGegabytes = Format(dblAns, "###,###,##0.00")

End Function
```

كائنات اخرى

الى جانب الدوال والاجراءات السابقة، توفر لك مكتبات VB و VBA مجموعة لا غنى عنها من الكائنات المستخدمة في برامجك الجدية.

كائن البرنامجApp

كائن البرنامج App يمثل البرنامج الحالي الذي يتم تنفيذه . يحتوي على مجموعة من الخصائص والطرق التي سأتطرق الى بعضها هنا.

الخاصية EXEName تعود باسم ملف البرنامج التنفيذي EXE ، والخاصية Path تعود بالمسار الكامل الذي نفذ البرنامج منه:

```
Shell App.Path & “\” & App.EXEName & “.EXE”
```

من الضروري التحقق من الرمز "\" قبل استخدام الخاصية Path ، ففي الكود السابق اضفنا هذا الرمز بعد الخاصية Path وذلك لان مسار البرنامج لن يضاف اليه هذا الرمز، ولكن تظهر المشكلة في الكود السابق اذا تمت عملية تنفيذ البرنامج من الدليل الجذري Root Directory للقرص، انظر الى الرمز "\" في قيمة الخاصية Pathاذا نفذ البرنامج من دليل جذري او فرعي:

```
App.Path  = "C:\"        ' الرمز \ مضاف
App.Path = "C:\Windows"    ' الرمز \ غير مضاف
```

وحتى تتجنب المشكلة السابقة، طور هذه الدالة وحاول الاعتماد عليه عوضا عن الخاصية Path المجردة:

```
Function PathEx() As String
If Right(App.Path, 1) = "\" Then
PathEx = App.Path
Else
PathEx = App.Path & "\"
End If
End Function
Shell PathEx & App.EXEName & “.EXE”
```

كائن الحافظة Clipboard

تتميز معظم تطبيقات Windows بامكانية الاتصال وتبادل البيانات فيما بينها، صحيح ان تبادل البيانات عن طريق الحافظة Clipboard محدود الامكانيات، الا انه اسلوب مازال متبع في معظم تطبيقات Windows . يمكنك Visual Basic من نسخ ولصق البيانات من والى الحافظة عن طريق الكائنClipboard. نبدأ أولاً بنسخ النص الى الحافظة باستخدام الطريقةSetText

```
Clipboard.Clear
Clipboard.SetText Text1.Text
```

ملاحظة :ينصح دائما باستخدام الطريقة Clear لمسح محتويات الحافظة قبل نسخ البيانات لها، وذلك لانه في حالات معينة لن تتمكن من نسخ بيانات جديدة الى الحافظة مالم يتم مسح محتوياتها.

```
Clipboard.Clear
Clipboard.SetText  Text1.Text
```

اما لنسخ الصور، فالطريقة SetData هي المستخدمة:

```
Clipboard.Clear
Clipboard.SetData Picture1.Picture
```

ولاجراء عملية لصق النصوص، فالطريقة GetText هي المستخدمة:

```
Text1.SelText = Clipboard.GetText
Text1.Text = Clipboard.GetText
```

ولصق الصور استخدم الطريقة GetData التي تشترط هيئة الصورة:

```
Set Picture1.Picture = Clipboard.GetData
```

كائن الطابعة Printer

الطباعة من المزايا الضرورية التي لابد من توفيرها في برامجك الجدية . بعيدا عن اجراءات APIالمعقدة، يوفر لك Visual Basic كائن الطابعة Printer Object والذي من اسمه يعرف غرضه.

من الخصائص الموجودة في مجموعة الطابعات Printers خاصية اسم الطابعة DeviceNameاو اسم المشغل DeriverName ، رقم المنفذ Port المركب عليه الطابعة ....الخ:

استخدم الكائن Printer لتعديل خصائص الطابعة الحالية او البدء في عملية الطباعة . من هذه الخصائص ، خاصية ColorMode التي تعرف عن طريقها ما اذا كانت الطابعة داعمة للالوان

ام لا . والخاصية PrinterQuality والتي تعود بنوع الكثافة النقطية وجودة الطباعة .
الخصائص الاخرى تجدها بشكل مفصل في مكتبةMSDN.
اما عملية الطباعة الفعلية فتتم باستخدام طرق كائن الطابعة وهي نفس الطرق الموجودة في كائن النموذج Cirlce ، Line، .... Print.
بعد ان تنتهي من ارسال البيانات الى الطباعة، استخدم الطريقة EndDoc لبدء عملية الطباعة الفعلية:

```
Printer.RightToLeft =True
Printer.FontSize = 40
Printer.Print "ALLAH"
ابدأ عملية الطباعة‘
Printer.EndDoc
```

الطريقة KillDoc تقوم بانهاء عملية الطباعة، والطريقة NewPage واضح من اسمها انها تخرج صفحة جديدة.

اكتشاف الاخطاء

فكرة عامة

تصنف الاخطاء في أي لغة برمجة الى صنفين على اساس وقت حدوثها اما في وقت التصميم او وقت التنفيذ . هذه الاخطاء تسبب انهيار برنامجك وانهاء تنفيذه. بالاضافة الى ذلك، يوجد نوع من الاخطاء التي لا تظهر لك بشكل مباشر تعرف بالشوائب . Bugs لنلقي نظرة على هذه الاصناف:

اخطاء وقت التصميم Design Time Errors:

وتعرف ايضا بالاخطاء النحوية Syntax Errors وهي اسهل انواع الاخطاء اكتشافا واصلاحا . وقت حدوث هذه الاخطاء يكون في مرحلة التصميم او الترجمة للبرنامج. سببها الرئيسي في طريقة كتابة العبارات البرمجية الخاطئة . فمثلا قد تكتب اسم دالة ليست موجودة، او تنشئ حلقة Forبدون اقفالها باستخدامNext.

توفر لك بيئة التطوير المتكاملة لـ Visual Basic تقنية في قمة الروعة هدفها قنص هذه الاخطاء تلقائيا بمجرد الوقوع فيها وذلك بعد الضغط على المفتاحENTER
مثلا، قم بكتابة X = = 4 واضغط مفتاح [ENTER] ستلاحظ ظهور رسالة توضح لك الخطأ وقد قلب لون السطر بالاحمر . تعرف هذه التقنية بالتدقيق النحوي التلقائي Auto Syntax Checkوالتي تستطيع الغا ئها عن طريق الاختيارAuto Syntax Checkالموجود في خانة التبويب Editor في نافذة الخيارات . Options لا اعتقد انك ستلغيها يوما من الايام ؟!

اخطاء وقت التنفيذ Run Time Errors:

وقت ظهور هذه الاخطاء مختلف . فلن تظهر الرسالة المزعجة السابقة وقت كتابة الكود وا نما في وقت التنفيذ . عندما يصل المفسر عند سطر صحيح نحويا لكنه خاطئ منطقيا ستظهر رسالة خطأ بعنوان Run Time Error ويظهر تحديد لمكان السطر الذي وقع فيه الخطأ .مثلا اكتب هذا الكود:

```
Dim X As Byte
X = 256
```

من الواضح ان الصيغة النحوية لهذا الكود صحيحة لكن من الناحية المنطقية خطأ.
جرب تنفيذ البرنامج وستلاحظ ظهور رسالة خطأ OverFlow وذلك لان القيمة القصوى التي يمكن ان يحملها أي متغير من نوع Byte هي . 255 طبعا اخطاء وقت التنفيذ كثيرة جدا جدا، فانت عندما تصمم البرنامج تتوقع ان كل الاحتمالات الخارجية كما هي في حالة تصميم البرنامج .مثلا لو وجد في احد سطور برنامج امر يقوم بمسح ملف معين وكتبت هذا الكود:

```
Kill “FileName.EXT ”
```

قد عمل معك بالشكل المطلوب، لكن افترض ان الملف لم يكن موجود؟ فان رسالة الخطأ سيكون لها نصيب من عمر تنفيذ البرنامج . فلو كنت ذكيا ستتأكد من وجود الملف باستخدام دالة Dir ومن ثم حذفه:

```
If Dir$(“FileName.EXT”) Then Kill “FileName.EXT ”
```

الشوائبBugs:

قد يكون الكود سليم من ناحية نحوية ولا توجد به أي اخطاء في وقت التنفيذ لكن به شوائب . لا يوجد برنامج الا وبه شوائب . الشوائب هي اخطاء في سلوك تنفيذ البرنامج لكنها لا تسبب في

ايقافه، وهي صعبة الايجاد والاكتشاف . لذلك، تجد غالبا في البرامج التجارية الكبيرة صدور نسخ تجريبية Beta توزع على اشخاص وشركات معينة الهدف منها تجربة البرنامج والتحقق من واكتشاف الشوائب الموجودة فيه . من اكبر الاخطاء الذي يقع فيها المبرمج هي محاولة اكتشاف الشوائب بنفسه ، لأنك لن تستطيع اكتشاف الشوائب الا عن طريق غيرك ، ففي حالة تجربة برامجك الشخصية فانك اكثر من يعرف طريقة التعامل معها، لكن في حالة وجود نسخة من البرنامج عند شخص اخر فالوضع يختلف، وتبدأ الشوائب بالظهور لديه.

Visual Basicبه شوائب .هناك الكثير من الشوائب التي تكتشف شهريا وتصدر شركة Microsoftتقارير عنها تجدها بشكل دوري في مكتبة MSDN ، بعضها تم اصلاحه و البعض الاخر لا .المقصد من هذا الكلام، انه مهما كان مستواك في البرمجة لابد من وجود شوائب في برنامجك.

يوجد الكثير من الكتب التي تناقش مسألة الشوائب البرمجية وكيفية تفاديها- اقصد الاقلال منها -الا انها مواضيع خارج نطاق الكتاب.

الكائنErr

عودا الى موضوع اخطاء وقت التشغيل وبالتحديد في مثال الملف ،FileName.EXT بدلا من كتابة عشرات الاسطر للتأكد من قابلية حذف الملف، استخدم كائن الخطأ Err.

قبل تطبيق هذا الكائن ، عليك معرفة أ ن كل خطأ من اخطاء وقت التشغيل له رقم خاص يميزة عن غيره من الاخطاء وكذلك وصف نصي مختصر للخطأ ، وعند حدوث الخطأ يتم وضع هذه البيانات- الخاصة بالخطأ - في الكائن . Err عند رغبتك في الاستمرار في عملية تنفيذ البرنامج حتى عند وقوع الخطأ لابد من كتابة التعليمة On Error Resume Next عند بداية كل اجراء تتوقع حدوث خطأ فيه حتى يستمر في تنفيذ سطور البرنامج راقب هذا المثال:

```
On Error Resume Next
Kill “FileName.EXT”
If Err Then
MsgBox Err.Description
Err.Clear
End If
```

هنا سنقوم بمحاولة حذف الملف ، ان لم يستطع البرنامج فعل ذلك فان الكائنErr سيحتوي على خصائص تتعلق بذلك الخطأ و سنظهر رسالة توضح وصف الخطأ . من المهم ا لتأكد من تنظيف الكائن Err عن طريق استدعاء الطريقة Clear حتى نخبر البرنامج اننا انتهينا من قنص الخطأ وانه لا يوجد خطأ اخر.

اما اذا كانت اكواد الاجراء طويلة ولا تود ان تكتب الجملة الشرطيةIf Err Then مرات متعددة، فيفضل استخدام :On Error Goto X والتي تؤدي الى الانتقال الى سطر معين في حال حدوث أي خطأ في تنفيذ اكواد الاجراء:

```
Sub LongSub ()
On Error Goto XZ:
…
…
XZ:
MsgBox Err.Description
Err.Clear
End Sub
```

العمل مع النماذج والطابعات ومعالجات الاخطاء

ستتعلم في هذا الدرس كيفية:

- اضافة نماذج جديدة الى البرنامج.
- ارسال الاخراج الى الطابعة.
- معالجة اخطاء وقت التشغيل باستعمال معالجات الاخطاء.

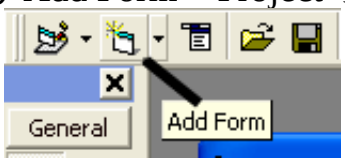
ستتعلم في هذا الدرس كيفية اضافة مزيد من النماذج الى الواجهة لمعالجة الادخال او الاخراج او الرسائل الخاصة. ستتعلم ايضا كيفية ارسال اخراج البرنامج الى الطابعة وكيفية استعمال معالجات الاخطاء لمعالجة النتائج غير المتوقعة.

اضافة نماذج جديدة الى البرنامج

لقد استعمل كل برنامج كتبته حتى الان نموذجا واحدا للادخال والاخراج في العديد من الحالات، يكون النموذج الواحد كافيا للتواصل مع المستخدم. لكن اذا اردت تقديم معلومات اكثر الى (او الحصول على معلومات اكثر من) المستخدم، يتيح لك فيجوال بيسك اضافة نموذج واحد او اكثر الى برنامجك. يعتبر كل نموذج جديد على انه كائن ويملك كائنات وخصائص واجراءات حدثية خاصة به ويملك **اسم فريد**. يسمى النموذج الاول في البرنامج Form1 والنماذج التالية Form2 و Form3 و .... الخ. يسرد الجدول التالي عدة استعمالات عملية للنماذج الاضافية في برنامجك.

النموذج او النماذج	الشرح
الشاشة الافتتاحية	شاشة تعرض رسالة ترحيب او اعمالا فنية او معلومات عن حقوق النشر عندما يبدا البرنامج بالاشتغال
ارشادات البرنامج	شاشة تعرض معلومات وتلميحات عن كيفية عمل البرنامج.
مربعات الحوار	مربعات حوار مخصصة تقبل الادخال وتعرض الاخراج في البرنامج.
محتويات المستند ورسومه الجاهزة	شاشة تعرض محتويات ملف او اكثر مستعمل في البرنامج.

النماذج الفارغة والمسبقة التصميم

يمكنك انشاء نموذج جديد باختيار الامر Add Form < Project او عن طريق النقر على ايقونة AddForm في شريط الادوات 

يظهر مربع حوار يسالك عن نوع النموذج الذي تريد انشاءه (كل اصدار لفيجوال بيسك يملك تشكيلة مختلفة من النماذج المصممة مسبقا). يمكنك اختيار انشاء نموذج جديد فارغ او نموذج مكتمل جزئيا مصمم لمهمة ما.

كيفية استعمال النماذج

يقدم لك فيجوال بيسك مرونة كبيرة عند استعمال النماذج. يمكنك جعل كل النماذج في البرنامج مرئية في الوقت نفسه. او يمكنك تحميلها والغاء تحميلها عند الحاجة اليها . اذا عرضت اكثر من نموذج واحد في الوقت نفسه ، يمكنك السماح للمستخدم بالتبديل بينهم، او يمكنك التحكم بترتيب استعمالها. يسمى النموذج الذي يجب استعماله عند عرضه على

الشاشة بـ **نموذج مشروط** (Modal) يبقى التركيز في النموذج الى ان ينقر المستخدم OK او Cancel او يلغيه بطريقة اخرى). اما النموذج الذي يستطيع المستخدم التبديل منه فيسمى **نموذجا غير مشروط** (NonModal) . تستعمل معظم البرامج المتوافقة مع ويندوز نماذج غير مشروطة عندما تعرض المعلومات كونها تعطي المستخدم حرية اكبر على صعيد الاستعمال، لذا فالنوع غير المشروط هو الافتراضي عندما تنشئ نموذجا جديدا. يمكنك ايضا ضبط أي خاصية على النموذج بشكل مستقل ، بما في ذلك التسمية التوضيحية للنموذج الـ (Caption) وحجمه ونمط حدوده الامامية والخلفية ونوع خطه وصورته الخلفية.

جمل النموذج في الشفيرة البرمجية

بعد انشاء نموذج جديد في بيئة البرمجة ، يمكنك تحميله الى الذاكرة واستعماله من خلال جمل معينة في الاجراءات الحدثية . التركيب النحوي للجملة التي تستعملها لتحميل نموذج جديد هو

```
Load  formname
```

حيث الـ *Formname* هو اسم النموذج الذي تريد تحميله . مثلا، الجملة

```
Load  Form2
```

ستحمل النموذج الثاني في البرنامج الى الذاكرة . بعد تحميل النموذج ، يمكنك استعماله في أي اجراء حدثي ويمكنك الوصول الى أي خاصية او طريقة تريد استعمالها معه.
مثلا لضبط الخاصية Caption للنموذج الثاني في البرنامج عند " New Form" يمكنك كتابة الجملة البرمجية في أي اجراء حدثي :

```
Form2.Caption = "New Form"
```

عندما تصبح جاهزا لاظهار نموذج محمل ، استدعه باستعمال الطريقة Show مع تحديد رغبتك بجعله مشروطا او غير مشروط . التركيب النحوي للطريقة Show هو:

```
Formname.Show  mode
```

حيث الـ *Formname* هو اسم النموذج و *mode* هو 0 لغير المشروط (الافتراضي) او 1 للمشروط. مثلا ، لاظهار النموذج Form2 كنموذج غير مشروط (النمط الافتراضي) ، يمكنك استعمال الطريقة Show من دون تحديد النمط:

```
Form2.Show
```

ولاظهاره كنموذج مشروط اكتب هذه الجملة:

```
Form2.Show 1
```

ملاحظة

اذا استعملت الطريقة Show قبل استعمال الجملة Load، سيحمل فيجوال بيسك النموذج ويعرضه تلقائيا. يزود فيجوال بيسك الجملة Load بشكل مستقل لكي يتمكن المبرمج من تحميل النماذج مسبقا الى الذاكرة ولكي تعمل الطريقة Show بسرعة ولا يلاحظ المستخدم أي تباطؤ في الاداء. من الجيد تحميل النماذج مسبقا خاصة اذا كانت تحتوي على عدة كائنات او اعمال فنية .

اخفاء النماذج والغاء تحميلها

يمكنك اخفاء النماذج باستعمال الطريقة Hide كما يمكنك الغاء تحميلها باستعمال الجملة Unload. وهاتان الكلمتان الدليليتان هما نقيضا الكلمتين Show و Load ، على التوالي. يؤدي اخفاء النموذج الى جعله غير مرئي لكنه يبقيه في الذاكرة ليستعمل لاحقا في البرنامج (ان اخفاء النموذج مطابق لجعله غير مرئي باستعمال الخاصية Visible). يؤدي الغاء تحميل النموذج الى ازالته من الذاكرة وتحرير الذاكرة التي كان يتم استعمالها لتخزين كائنات النموذج ورسومه، لكنه لايحرر المساحة التي تستعملها اجراءات النموذج الحدثية . فهذه الاجراءات تبقى في الذاكرة دائما . يمكنك استعمال الكلمتين الدليليتين Hide و Unload لاخفاء النموذج Form2 والغاء تحميله كالتالي:

```
Form2.Hide
Unload Form2
```

مهم

عند الغاء تحميل نموذج ، ستخسر قيم وخصائص وقت التشغيل العائدة له. اذا اعدت تحميل النموذج، فانه سيحتوي على القيم الاصلية المضبوطة في الشفيرة.

ملاحظة :

يمكنك ايضا اخفاء النموذج واظهاره باستعمال الخاصية Visible

في الحالة الطبيعية تكون الخاصية Visible تساوي True أي يكون النموذج (الكائن) مرئي للمستخدم، مثلا،

تستخدم لاظهار النموذج اذا كان غير مرئي وليس غير محمل Form1.Visible = True

اما لاخفاء النموذج لكن من دون الغاء تحميله أي يبقى محمل في الذاكرة ولكنه غير مرئي نستخدم:

```
Form1.Visible = False.
```

تصغير النماذج

يمكنك تصغير النموذج (وضعه على شريط المهام Taskbar) او تكبيره (توسيعه ليملأ الشاشة باكملها) باستعمال الخاصية WindowState. مثلا ستصغر الجملة التالية النموذج Form1 في البرنامج:

```
Form1.WindowState = 1
```

ولتكبيره استعمل الجملة التالية:

```
Form1.WindowState = 2
```

ولإعادته الى شكله العادي الافتراضي، استعمل الجملة:

```
Form1.WindowState = 0
```

تحديد أي نموذج سيعمل عند بدء التشغيل

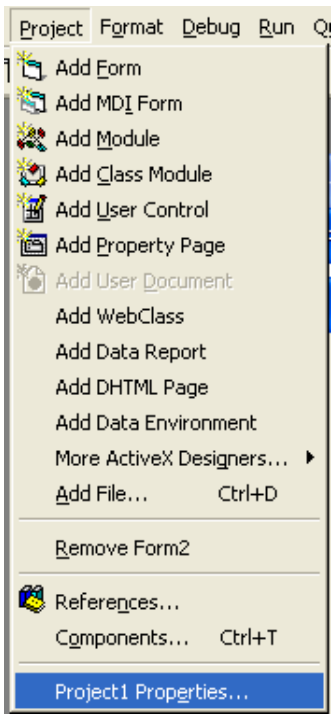
عند اضافة نموذج ثان الى البرنامج او ثالث فان النموذج الذي سوف يحمل ويظهر عند بدء تشغيل البرنامج هو النموذج الاول Form1 لانه يعتبر النموذج الافتراضي للبرنامج ، ماذا لو انك اردت ان تجعل النموذج الثاني او الثالث يبدء بالعمل عند بدء التشغيل وليس النموذج الاول (الافتراضي) يمكنك ذلك عن طريق استعمال الطريقة Load و Show و Unload و Hide حيث ستضع الجمل التالية في الاجراء الحدثي Form1_load للنموذج الاول

```
Load Form2
Form2.Show
Form1.Hide
Unload Form1
```

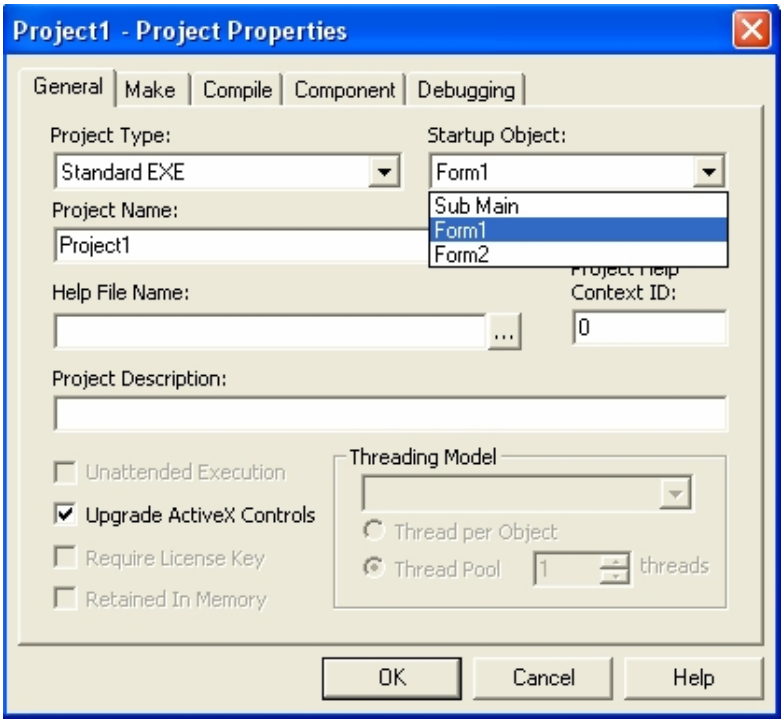
ستقوم بتحميل النموذج الثاني في بدء تشغيل النموذج الاول واظهاره ثم ستقوم باخفاء النموذج الاول والغاء تحمله ايضا في بدء التشغيل. حيث سوف يظهر النموذج الثاني عند تشغيل البرنامج لانك كتبت الجمل السابقة.

ولكن هنا تم تحميل النموذج الاول ومن ثم تم تحميل النموذج الثاني عن طريق النموذج الاول ، ماذا لو اردت ان يتم تحميل النموذج الثاني دون ان يتم تحميل النموذج الاول ومن ثم الغاء تحميله كما بينا سابقا؟ يمكنك ذلك وذلك بالذهاب الى القائمة Project ثم اختيار الامر Properties ثم نقر علامة التبويب General واختيار النموذج المناسب من مربع السرد

Startup Object . اختر النموذج الذي تريده ان يبدا في بدء التشغيل ثم انقر OK. كما مبين في الشكلين (7-1) و (7-2).



شكل(7-1)

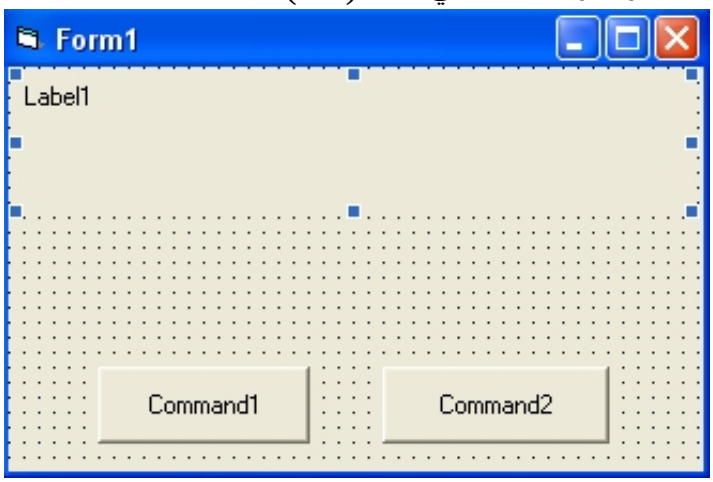


شكل(7-2)

العمل مع عدة نماذج : البرنامج About Me & Help

1. اختر الامر New Project < File ثم OK.
2. انقر اداة التحكم زر امر Command في مربع الادوات .
3. انشئ زر امر في اسفل النموذج.
4. انشئ زر امر ثان بجانب الزر الاول.
5. انقر اداة التحكم تسمية Label في مربع الادوات .
6. انشئ تسمية في اعلى النموذج وعلى طول النموذج. ستستخدم للعنوان.

يجب ان يبدو نموذجك كما في شكل (7-3).



شكل(7-3)

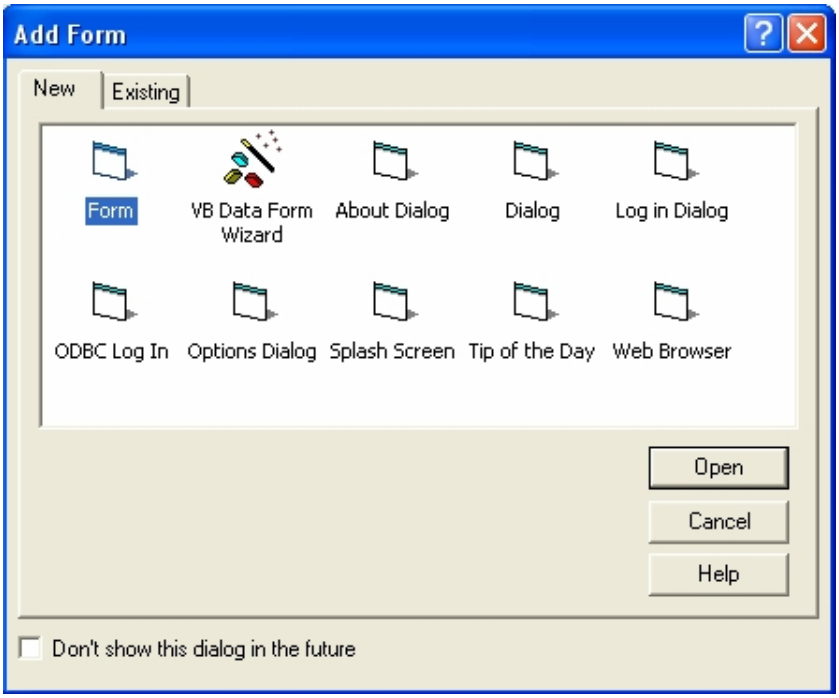
7. اضبط خصائص الكائنات كما مبين في الجدول التالي.

الكائن	الخاصية	الاعداد
Commad1	Caption	About
Command2	Caption	Help
Form1	Caption	Multi Forms
Label1	Caption	This Program Use Multi Forms and show you how to switch between it " "
	Font	الحجم 18 ، Bold
	ForeColor	احمر

8. بعد ان تضبط الخصائص السابقة يجب ان يبدو نموذجك كما في شكل (4-7).

شكل (4-7)

9. اختر الامر Add Form < Project ، ستقوم باضافة نموذج جديد الى مشروعك ، سيعرض فيجوال بيسك مربع حوار يسالك عن نوع النموذج الذي تريد اضافته (أي نموذج فارغ او نموذج مسبق التصميم كما في شكل (5-7)). ستختار اول نموذج أي "Form" ثم اضغط على Open . سيقوم فيجوال بيسك باضافة نموذج جديد الى مشروعك وسيكون اسمه Form2 . حيث سترى انه يشبه النموذج الاول عند فتح البرنامج واختيار مشروع جديد. ستستخدم هذا النموذج الجديد لاظهار نبذة مختصرة عن البرنامج و عن اصدار البرنامج وعن المصمم ومعلومات اخرى انت تريد ان يعرفها المستخدم عنك كعنوان البريد الالكتروني لمصمم البرنامج .



شكل (7-5)

10.بعد ان اضاف فيجوال بيسك النموذج الجديد قم :

11.انقر اداة التحكم صورة Image في مربع الادوات.

12.انشئ كائن صورة على النموذج الثاني في الزاوية العليا اليمنى للنموذج الثاني وليكن حجمها صغير (تستخدم الصورة لعرض شعار البرنامج او الشركة).

13.انقر اداة التحكم تسمية Label في مربع الادوات.

14.انشئ تسمية في اعلى النموذج الثاني في الوسط.

15.انشئ تسمية ثانية تحت التسمية الاولى ولتكن اطول من التسمية الاولى .

16.انشئ تسمية ثالثة تحت التسمية الثانية واترك مسافة قليلة بين التسميتين .

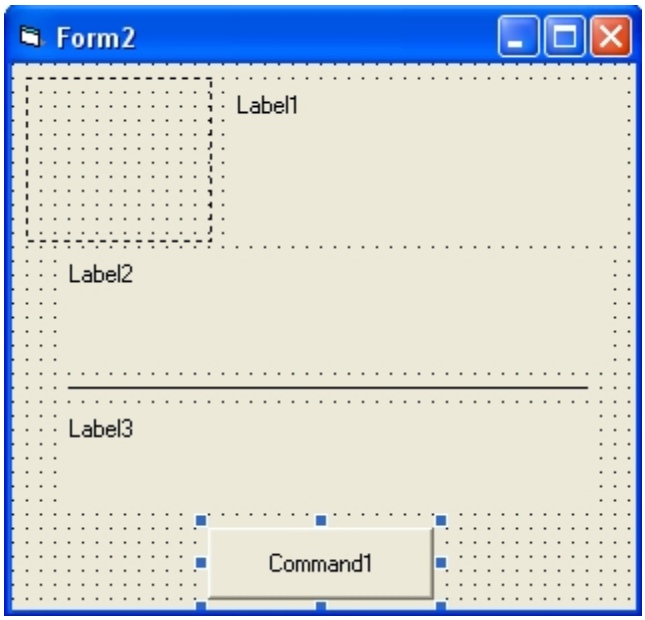
17.انقر اداة التحكم خط Line في مربع الادوات.

18.انشئ خط في المسافة التي تركتها بين التسميتين.

19.انقر اداة التحكم زر امر Command في مربع الادوات.

20.انشئ زر امر في اسفل النموذج في الوسط .

يجب ان يبدو نموذجك الثاني كما في شكل(7-6)



شكل (6-7)

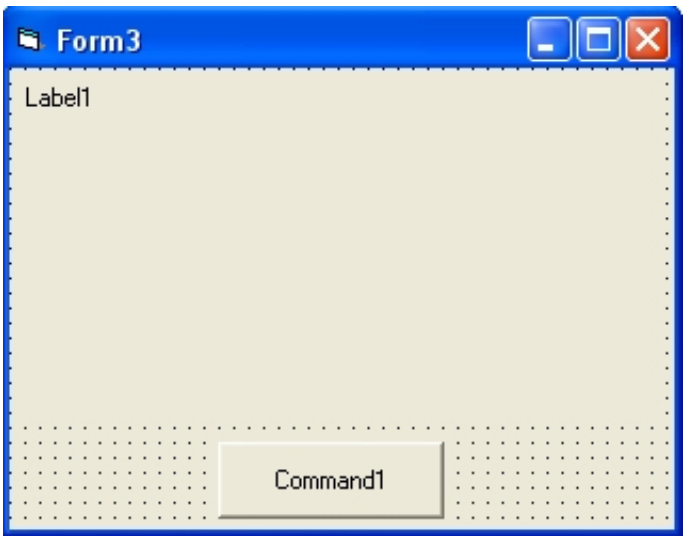
21.اضبط خصائص الكائنات كما مبين في الجدول التالي:

الكائن	الخاصية	الاعداد
Form1	Caption	"About My Program"
	BoarderStyle	1-Fixed Single
Image1	Picture	"اختر صورة موجودة في كمبيوترك، في البرنامج التالي قمت باختيار الصورة من MyDocuments > MyPicture > "Sunset.jpg
	Stretch	True
Command1	Caption	OK
Label1	Caption	"Multi Forms Program . Version 1.0.0"
Label2	Caption	This Product licensed to: Ahmed copyright (c) 20007 - 2008
	BoarderStyle	1-Fixed Single
Label3	Caption	To Contact us (me) Your e-mail@yourweb.com

22.بعد ان تضبط الخصائص السابقة يجب ان يبدو نموذجك كما في شكل (7-7)

شكل (7-7)

23. اختر الامر Add Form <Project ، قم باضافة نموذج ثالث الى المشروع حيث سيضيف فيجوال بيسك نموذج ثالث باسم Form3 ، ستستخدم هذا النموذج الجديد لاظهار معلومات عن البرنامج ، معلومات مساعدة أي تخبر المستخدم كيف يمكنه استعمال برنامجك كما هو الحال في المساعدة (Help) في جميع البرامج، لكن بما ان البرنامج هنا صغير ولا يحتاج الى مساعدة سنكتب أي شئ ولكن لو انك قمت بعمل مشروع كبير واردت ان تضع فيه Help لمساعدة المستخدم على كيفية استعمال البرنامج قم بتحسين هذا النموذج واضافة معلومات مفيدة اليه.
24. بعد ان اضاف فيجوال بيسك النموذج الثالث قم بما يلي:
25. انقر اداة التحكم تسمية Label في مربع الادوات.
26. انشئ تسمية كبيرة تغطي معظم مساحة النموذج (ستستخدم لكتابة المساعدة فيها) او يمكنك الاستعاضة عن التسمية بادراج مربع نص بدل التسمية (قد تجد فيه مرونة اكبر من ناحية التعامل مع النصوص).
27. انقر زر امر Command من مربع الادوات.
28. انشئ زر امر في اسفل النموذج في الوسط.
29. بعد الانتهاء من وضع الكائنات على النموذج ، يجب ان يبدو كما في شكل(8-7)

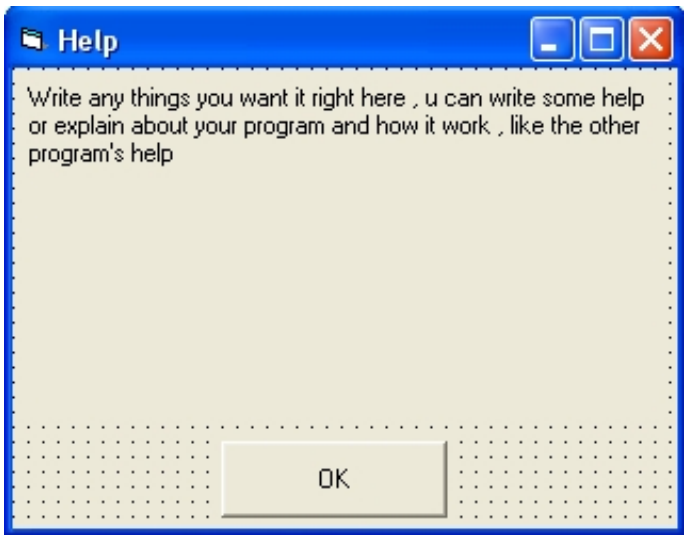


شكل (7-8)

30.اضبط خصائص الكائنات كما مبين في الجدول التالي:

الكائن	الخاصية	الاعداد
Form1	Caption	Help
Command1	Caption	OK
Label1	Caption	Write any things you want it right here , u can write some help or explain about your program and how it work , like the other program's help

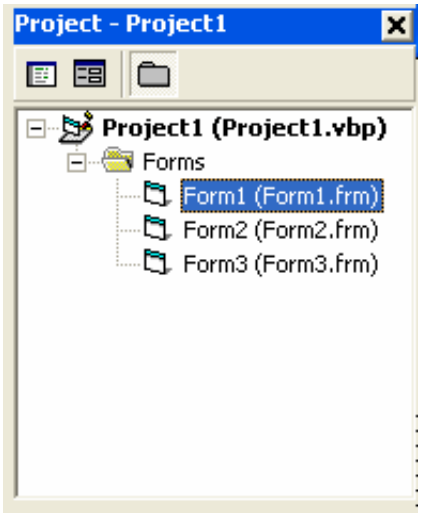
31.بعد ان تضبط الخصائص السابقة يجب ان يبدو نموذجك كما في شكل (7-9)



شكل (7-9)

32.بعد ان قمت بانشاء النماذج الثلاثة ووضع الكائنات عليها ثم ضبط خصائص الكائنات . ستقوم الان بكتابة الشفيرة البرمجية لكل نموذج ولكل او بعض كائناته.

33.اذهب الى **اطار المشروع** الذي يكون في الجهة اليمنى العليا لبرنامج فيجوال بيسك حيث سترى انه يسرد جميع النماذج الموجودة ضمن مشروعك كما مبين في شكل (7-10)



شكل (7-10)

يمكنك اخفاء النماذج في اطار المشروع وذلك بالنقر على الرمز ⊟ حيث ستتجمع النماذج وتصبح داخل المجلد الاصفر Forms الذي يحوي على نماذج المشروع ، حيث سيصبح الرمز على شكل + .

34.قم بالنقر على النموذج الاول Form1 (Form1.frm) في اطار المشروع نقرا مزدوجا ستلاحظ ظهور النموذج Form1 في اطار مستوعب المشروع.

35.انقر زر الامر About نقرا مزدوجا.

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة .

36.اكتب الجمل البرمجية التالية:

```
Load Form2
Form2.Show
```

يعمل السطر الاول على تحميل النموذج الثاني Form2 أي النموذج الذي يظهر معلومات عن المصمم والبرنامج. اما السطر الثاني فيقوم باظهار النموذج الثاني أي يجعله مرئي للمستخدم ويحمله في الذاكرة اذ لم يكن محمل.

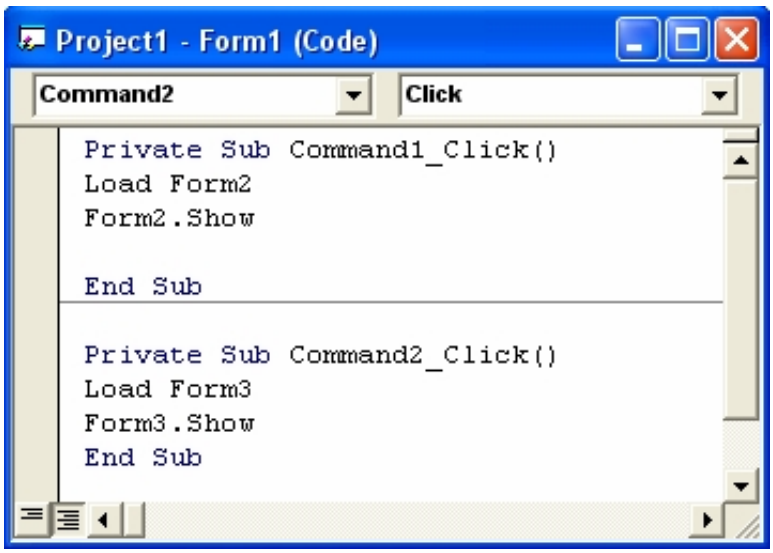
37.انقر الكائن Command2 في مربع السرد المنسدل في اطار الشفيرة ثم اكتب الجمل التالية في الاجراء الحدثي Command2_Click:

Load Form3

Form3.Show

يعمل السطر الاول على تحميل النموذج الثالث Form3 ويعمل السطر الثاني على اظهاره.

يجب ان يبدو اطار الشفيرة كما مبين في الشكل (11-7)



شكل (11-7)

38.قم بالنقر على النموذج الثاني Form2 (Form2.frm) في اطار المشروع نقرا مزدوجا ستلاحظ ظهور النموذج Form2 في اطار مستوعب المشروع.

39.انقر زر الامر OK نقرا مزدوجا.

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة .

40.اكتب الجمل البرمجية التالية:

Form2.Hide

Unload Form2

يعمل السطر الاول على اخفاء النموذج الثاني، يجب ان تجعل زر يقوم بارجاعك الى النموذج الاصلي (أي النموذج الاول في هذا المثال) بعد ان ينتهي المستخدم من قراءة المعلومات المتعلقة بالنموذج الحالي. ويعمل السطر الثاني على الغاء تحميل النموذج الثاني.

41.قم بالنقر على النموذج الثالث Form3 (Form3.frm) في اطار المشروع نقرا مزدوجا ستلاحظ ظهور النموذج Form3 في اطار مستوعب المشروع.

42.انقر زر الامر OK نقرا مزدوجا.

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة .

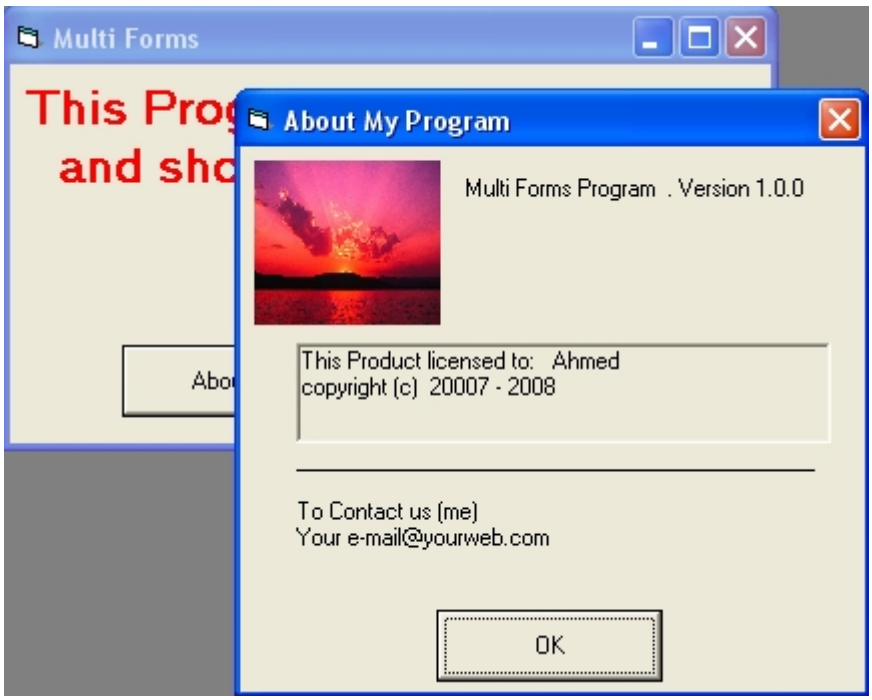
43.اكتب الجمل البرمجية التالية:

```
Form3.Hide
Unload Form3
```

44.يعمل السطر الاول على اخفاء النموذج الثالث، ويعمل السطر الثاني على الغاء تحميل النموذج الثالث.

45.بعد الانتهاء من برمجة كائنات النماذج قم الان بتشغيل البرنامج

46.انقر زر About سترى انه سيقوم بعرض النموذج الثاني ويجعله فوق النموذج الاول كما في الشكل(12-7).



شكل (12-7)

47.انقر زر الامر OK ستلاحظ ان النموذج سوف يختفي ،ويبقى فقط النموذج الاول.

48.انقر زر الامر Help ستلاحظ ظهور النموذج الثالث.

49.انقر زر ايقاف لايقاف االبرنامج.

50.احفظ النموذج باسم معين.

ارسال اخراج البرنامج الى الطابعة

يتيح لك فيجوال بيسك ارسال الاخراج الى طابعة مثبتة باستعمال الطريقة Print. لقد تعلمت في دروس سابقة عن هذه الطريقة عندما استعملتها لعرض نص على النموذج. لارسال الاخراج الى طابعة موصولة بالكمبيوتر ، استعمل الطريقة Print مع كائن الطابعة. مثلا، يرسل السطر التالي السلسلة "God Is The Greatest" الى الطابعة الافتراضية في ويندوز:

```
Printer.Print  "God Is The Greatest"
```

يمكنك ايضا استعمال كائن الطابعة قبل تنفيذ الطباعة لتعديل بعض خصائص الخط. مثلا، تطبع الشفيرة التالية السلسلة "God Is The Greatest" بالحجم 14 نقطة:

```
Printer.FontSize = 14
Printer.Print  "God Is The Greatest"
```

يملك كائن الطابعة عشرات الخصائص والطرق التي يمكنك استعمالها للتحكم بمختلف نواحي الطباعة. ومعظم تلك الخصائص والطرق مشابهة للكلمات الدليلية التي استعملتها سابقا مع النماذج والكائنات التي تنشئها باستعمال ادوات التحكم في مربع الادوات. لكن خصائص الطابعة تختلف في ناحية مهمة عن خصائص النماذج والكائنات التي تنشئها باستعمال ادوات التحكم في مربع الادوات: لا يمكنك ضبط خصائص الطابعة باستعمال اطار الخصائص . يجب ضبط كل خاصية من خلال الشفيرة البرمجية اثناء تشغيل البرنامج.

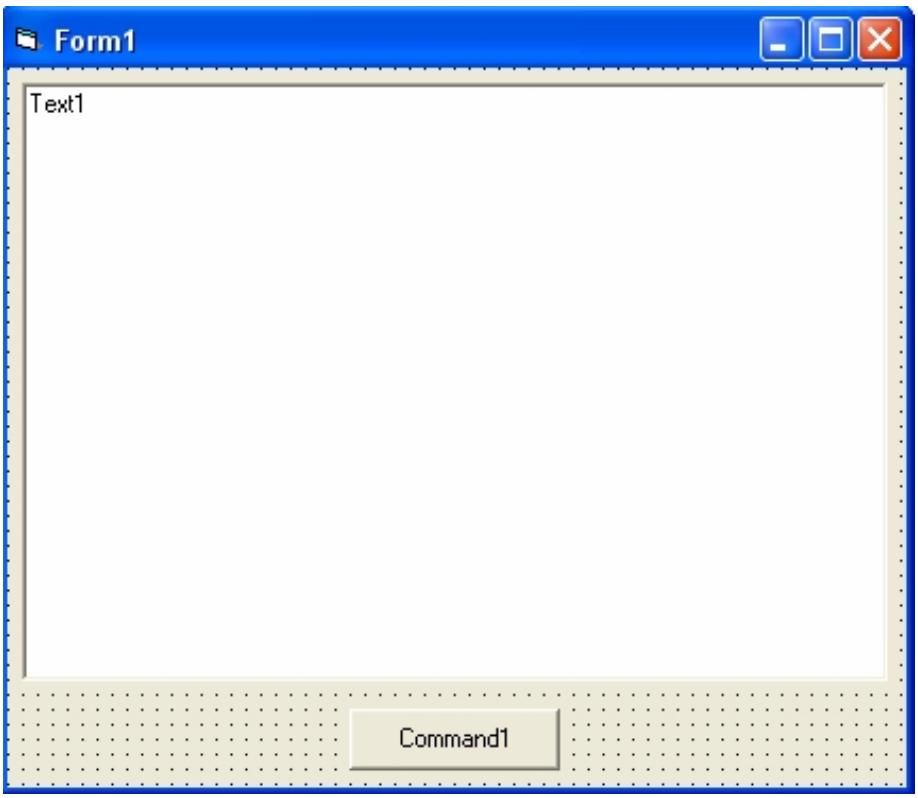
يسرد الجدول التالي بعضا من خصائص الطابعة وطرقها المهمة.

الخاصية	الوصف
FontName	ضبط اسم خط النص.
FontSize	ضبط حجم خط النص.
FontBold	القيمة True تضبط نمط الخط عند اسود عريض.
FontItalic	القيمة True تضبط نمط الخط عند مائل.
Page	تحتوي على رقم الصفحة التي تتم طباعتها.
الطريقة	**الوصف**
Print	تطبع النص المحدد على الطابعة.
NewPage	تبدا صفحة جديدة في عملية الطباعة.
EndDoc	تشير الى انتهاء عملية الطباعة.
KillDoc	تلغي عملية الطباعة الحالية.

عمل برنامج لطباعة النصوص MyPrinter

1. اختر الامر New Project < File ثم OK.
2. انقر اداة التحكم زر امر Command في مربع الادوات .
3. انشئ زر امر في اسفل النموذج.
4. انقر اداة التحكم مربع نص TextBox في مربع الادوات .
5. انشئ مربع نص يملا معظم مساحة النموذج.

يجب ان يبدو نموذجك كما في شكل (13-7).



شكل(13-7)

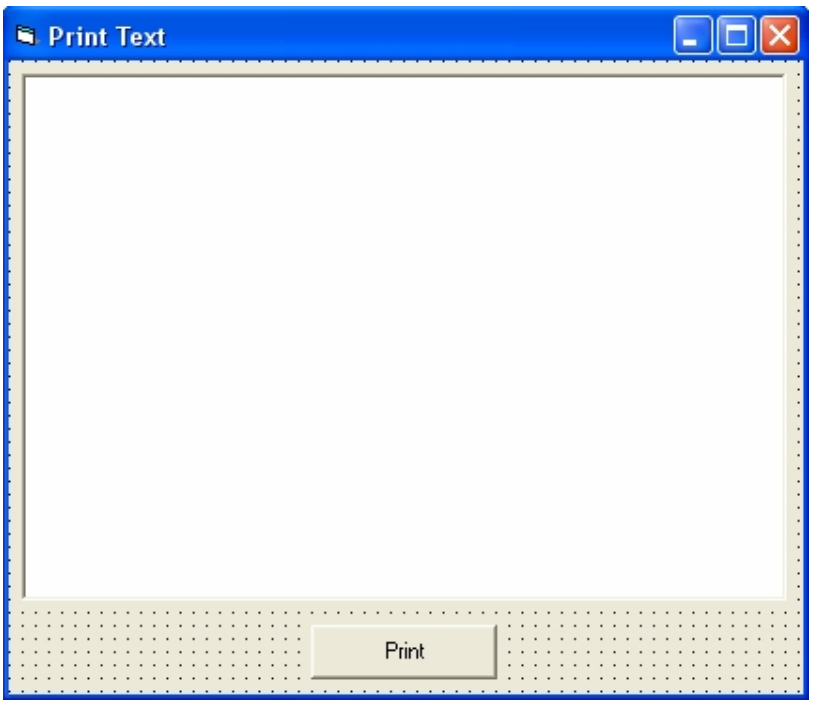
6. اضبط خصائص الكائنات كما مبين في الجدول التالي:

الكائن	الخاصية	الاعداد
Command1	Caption	Print
Text1	Text	فارغ
	MultiLine	True
Form1	Caption	Print Text

تستخدم الخاصية MultiLine في مربع النص للسماح لمربع النص ان يحتوي عدة اسطر ، ففي حال كانت الخاصية MultiLine تساوي False فانك سوف لن

تستطيع كتابة سطر ثاني او ثالث في مربع النص بل تسسطيع فقط ان تكتب سطر واحد

7. بعد ان قمت بضبط الخصائص يجب ان يبدو نموذجك كما في الشكل (14-7)



شكل (14-7)

8. انقر زر الامر Print نقرا مزدوجا.

يظهر الاجراء الحدثي Command1_Click في اطار الشفيرة.

9. اكتب جمل الطباعة التالية في الاجراء الحدثي:

```
Printer.FontName = "Arial"
Printer.FontSize = 16
Printer.FontBold = True
Printer.Print Text1.Text
Printer.EndDoc
```

اليك تفسير بعض النقاط المهمة في هذه الشفيرة

- الخاصية FontName تضبط خط الطابعة عند Arial ، وهو خط موجود في كل انظمة التشغيل ويندوز.
- الخاصية FontSize تضبط حجم الخط عند 18 نقطة.
- ضبط الخاصية عند FontBold عند True يجعل الخط اسود عريض.
- يستعمل السطر الرابع الطريقة Print لطباعة النص الموجود داخل مربع النص.

- ينهي السطر الاخير عملية الطباعة ويرسلها الى الطابعة.

10.انقر زر تشغيل البرنامج في شريط الادوات.
يشتغل البرنامج في بيئة البرمجة.
11.اكتب أي شئ في مربع النص.
12.بعد الانتهاء من الكتابة اضغط زر الامر Print على النموذج لطباعة النص على الورق.
يرسل فيجوال بيسك مستندك الى الطابعة الافتراضية.

مهم

اذا لم تكن الطابعة جاهزة للطباعة ، قد يعيد ويندوز خطا الى فيجوال بيسك لايستطيع برنامجك معالجته، مما قد يؤدي الى ظهور خطا تشغيل او الى توقف البرنامج. ستتعلم كيفية معالجة اخطاء التشغيل الناتجة عن محركات الاقراص والطابعات والاجهزة الاخرى لاحقا في هذا الدرس.

طباعة نموذج باكمله باستعمال الطريقة PrintForm

كبديل لطباعة الاسطر منفردة بواسطة الطريقة Print، يمكنك ارسال كل محتويات نموذج واحد او اكثر الى الطابعة باستعمال الطريقة PrintForm . يتيح لك هذا الاسلوب ترتيب النصوص والرسوم وعناصر الواجهة مثلما تريدها على النموذج ومن ثم ارسال النموذج باكمله الى الطابعة. يمكنك استعمال الكلمة الدليلية PrintForm فقط لطباعة النموذج الحالي، او يمكنك كتابة اسم نموذج لطباعة نموذج معين. مثلا، لطباعة محتويات النموذج الثاني في البرنامج، يمكنك كتابة الجملة التالية:

```
Form2.PrintForm
```

في أي اجراء حدثي في البرنامج.

استعمال PrintForm لطباعة النصوص والرسوم

1. اختر الامر New Project < File ثم OK.
2. انقر اداة التحكم زر امر Command في مربع الادوات، وانشئ زر امر في الزاوية السفلى اليمنى للنموذج.

3. انقر اداة التحكم تسمية Label في مربع الادوات ،انشئ تسمية في وسط النموذج.
4. اضبط خصائص الكائنات التالية في البرنامج:

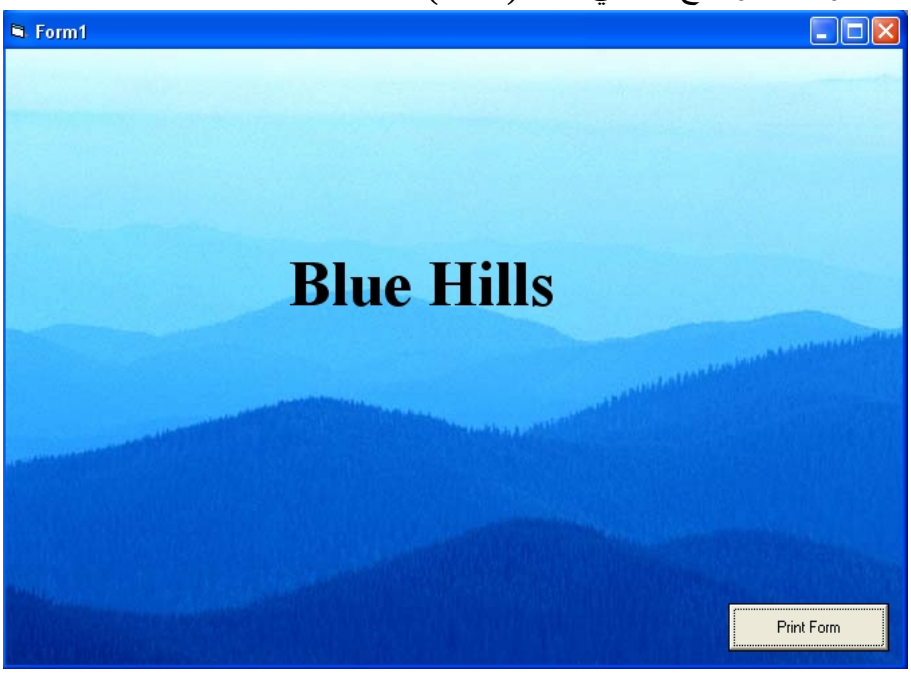
الكائن	الخاصية	الاعداد
Label1	Caption	"Blue Hills"
	BackStyle	0-Transparent
	Font	Time New Roman، Bold ، 36 نقطة
Command1	Caption	"Print Form"
Form1	Picture	"C:\Documents and Settings\All Users\Documents\My Pictures\Sample Pictures\Blue Hills.jpg"

5. انقر الزر Print Form نقرا مزدوجا لفتح اجرائه الحدثي.
6. اكتب الجملة البرمجية التالية:

```
Form1.PrintForm
```

7. انقر زر تشغيل البرنامج .

يعرض البرنامج كما في شكل (7-15)



شكل (7-15)

8. انقر الزر Print Form لطباعة محتويات النموذج.

يرسل فيجوال بيسك كل محتويات النموذج (التسمية والصورة الخلفية وزر الامر) الى الطابعة .

9. انقر زر حفظ ، لحفظ النموذج.

مهم

تطبع الطريقة PrintForm فقط الكائنات المرئية حاليا على النموذج. لازالة الكائنات الغير مرغوب بها من النسخة المطبوعة (كالزر Print Form في المثال السابق)، اضبط خاصيتها Visible عند False قبل استدعاء الطريقة PrintForm . ثم اضف جملة لجعل الكائنات مرئية مجددا بعد ان تكون قد ارسلت النموذج للطباعة.

كما مبين في الجمل الاتية

```
Command1.Visible = False
Form1.PrintForm
Command1.Visible = True
```

التعامل مع الاخطاء باستعمال معالجات الاخطاء

هل صادفت خطا تشغيل في فيجوال بيسك حتى الان ؟ ان خطا التشغيل، او توقف البرنامج، هو حدث غير متوقع يطرا اثناء تشغيل البرنامج لا يستطيع فيجوال بيسك ان يتجاهله . قد يكون اول خطا اعترضك في هذا الدرس هو اثناء محاولتك الطباعة (قد تكون الطابعة خالية من الورق او غير مشتغلة). يحصل خطا التشغيل كلما كان فيجوال بيسك ينفذ جملة لايمكنه اتمامها لسبب من الاسباب "كما هو مطلوب" . ليس لان فيجوال بيسك غير قوي كفاية لمعالجة الخلل، بل لان المصرف "Compiler " لم يتم ابلاغه ما عليه فعله عند حدوث خطا ما.

لحسن الحظ يتيح لك فيجوال بيسك كتابة روتينيات خاصة تسمى **معالجات اخطاء** للتعامل مع اخطاء التشغيل . تُطلع معالجات الاخطاء البرنامج على كيفية المتابعة في حال لم تشتغل احدى الجمل . توضع معالجات الاخطاء في نفس الاجراء الحدثي الذي توجد فيه الجمل موضوع الشك فتصطاد المشاكل عند حصولها باستعمال كائن خاص لمعالجة الاخطاء يدعى Err. يملك الكائن Err خاصية تدعى Number تُعرف عن الخطاء وتتيح للبرنامج التعامل معه. مثلا، اذا تسبب محرك الاقراص (CDROM) بخطا ما، قد يعرض معالج الاخطاء رسالة مخصصة ثم يعطل عمليات القرص الى ان يصلح المستخدم المشكلة.

متى يجب استعمال معالجات الاخطاء

يمكنك استعمال معالجات الاخطاء في أي وضع قد يحدث فيه شئ غير متوقع اثناء تشغيل البرنامج. عادة، يتم استعمال معالجات الاخطاء لمعالجة الاحداث الخارجية التي تؤثر على سير البرنامج – مثلا، احداث سببها عطل في محرك اقراص الشبكة او ترك باب محرك

الاقراص مفتوحا او عدم تشغيل الطابعة. يسرد الجدول التالي المشاكل المحتملة التي يمكن ان تصطادها معالجات الاخطاء.

المشاكل	الوصف
مشاكل الشبكة	تعطل محركات اقراص او موارد الشبكة فجاة
مشاكل الاقراص المرنة CDROM	اقراص غير مهياة او مهياة بشكل غير صحيح، او باب محرك الاقراص مفتوح، او قطاعات معطوبة في الاقراص.
مشاكل الطابعة	طابعات غير مشتغلة او خالية من الورق او غير متوفرة لاسباب اخرى.
اخطاء الفائض	اكثار من معلومات الطباعة او تحميل متغير اكثر من قيمته.
اخطاء نفاذ الذاكرة	نقص في مساحة التطبيق او الموارد في ويندوز.
مشاكل الحافظة	مشاكل في نقل البيانات او في حافظة ويندوز.
اخطاء منطقية	اخطاء في بناء الجمل او في المنطق لم يكشفها المصرف ولا الاختبارات السابقة (كخطا املائي في اسم ملف).

إعداد الفخ: الجملة On Error

الجملة البرمجية المستعملة لكشف خطا التشغيل هي On Error . يتم وضع هذه الجملة في الاجراء الحدثي قبل الجملة التي ترتاب منها . تضبط الجملة On Error ، او تضع، فخا من خلال ابلاغ فيجوال بيسك اين عليه الذهاب عند حدوث خطا.

التركيب النحوي للجملة On Error هو:

On Error GoTo *Address*

حيث *Address* هو اسم معالج الاخطاء .

توضع معالجات الاخطاء عادة عند نهاية الاجراء الحدثي بعد الجملة On Error. لكل معالج اخطاء تسميته الخاصة تليها نقطتان (:) للتعريف عنه مثلا :ErrorHandler او :CDROM_Error . عادة يتالف معالج الاخطاء من قسمين . يستعمل القسم الاول الخاصية Err.Number في بنية قرار (If … Then او Select Case مثلا) ثم يعرض رسالة او يضبط خاصية بناء على الخطا. القسم الثاني هو جملة Resume تعيد التحكم الى البرنامج ليتابع عمله.

الجملة Resume

يمكنك استعمال الكلمة الدليلية Resume . بمفردها كما يمكنك استعمال Resume Next او Resume مع تسمية تريد ان تتفرع اليها ، وفقا لقسم البرنامج الذي تريد المتابعة معه. تعيد الكلمة Resume التحكم الى الجملة التي تسببت بالخطاء(املا في ان يكون قد تم تصليح سبب الخطا) . ان استعمال الكلمة الدليلية Resume هو وسيلة جيدة اذا كنت تطلب من

المستخدم تصحيح الخطا، مثلا، اغلاق باب محرك الاقراص او ادخال القرص اذ لم يكن هناك قرص.

تعيد الكلمتان الدليليتان Resume Next التحكم الى الجملة التي تلي الجملة التي تسببت بالخطا. استعمل Resume Next اذا كنت تريد تخطي المشكلة ومتابعة العمل.

كما يمكنك كتابة تسمية تريد التفرع اليها بعد الكلمة Resume . يتيح لك هذا الاسلوب الانتقال بحرية الى أي مكان في الاجراء الحدثي. المكان النموذجي للتفرع اليه هو السطر الاخير في الاجراء.

معالجات اخطاء محرك الاقراص

يبين لك المثال التالي كيف يمكنك معالجة اخطاء لحل الاخطاء الناتجة عن محركات الاقراص المرنة . ستضيف معالج الاخطاء الى برنامج سيحاول تحميل صور من محرك الاقراص. يمكنك استعمال الاسلوب نفسه لاضافة ميزة معالجة الاخطاء الى أي برنامج فيجوال بيسك – فقط غير ارقام الاخطاء والرسائل التي تريد عرضها.

تلميح

يستعمل البرنامج التالي رقم خطا (من الخاصية Err.Number) لتشخيص وجود خطا تشغيل .

انشاء معالج اخطاء محرك الاقراص

سنقوم هنا بفتح البرنامج Photo Browser الذي قمت بانشاءه في الدرس الثالث، وهو برنامج يستعرض الصور الموجودة في القرص الثابت واقراص المحركات المرنة.

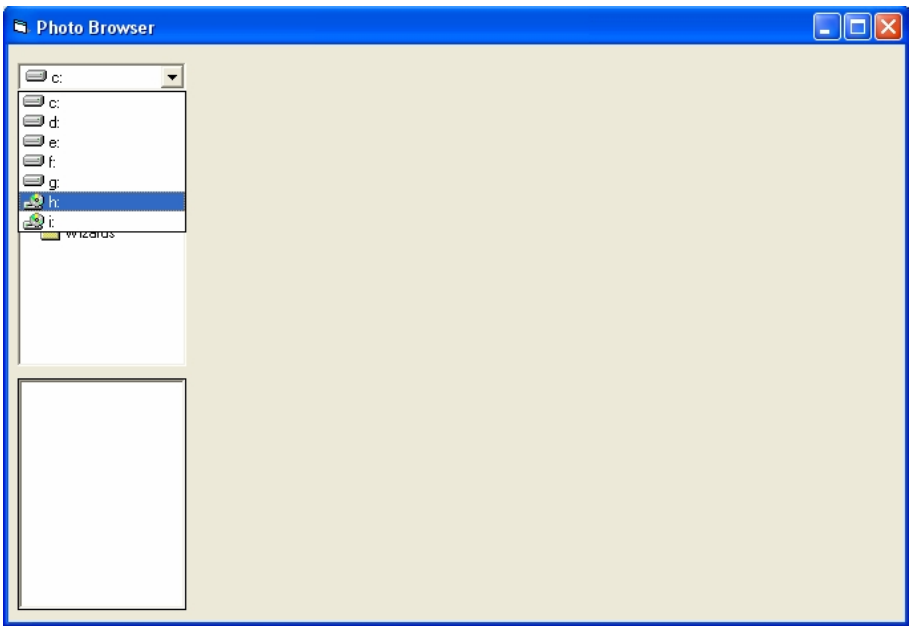
1. اذا لم تكن قد حفظت البرنامج Photo Browser فعليك انشاءه من جديد ، ولعمل ذلك اذهب الى صفحة 37 ثم اتبع الخطوات لكي تنشئ البرنامج.
2. قم بفتح البرنامج Photo Browser الموجود في المجلد

"D:\VBLessons\ Photo" سبق وان خزنته بهذا المكان واذا لم تكن قد خزنته في هذا العنوان قم بفتحه من المكان الذي قد خزنته به.

يظهر النموذج Photo Browser في اطار المشروع.

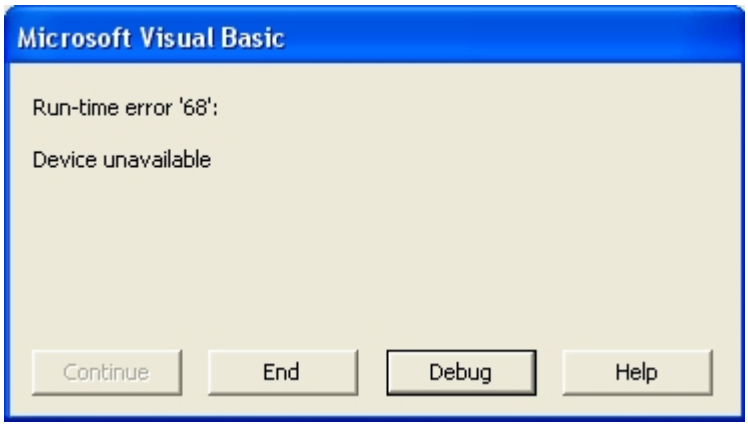
3. اذا لم يكن النموذج مرئيا ، انقر النموذج Form1 في اطار المشروع نقرا مزدوجا.
4. قم بتشغيل البرنامج من زر الادوات.

5. سيشتغل البرنامج ، اذهب الى مربع سرد محركات الاقراص Driver واختر محرك الاقراص أي الـ CDROM وليكن فارغ أي لا يحوي على قرص داخله. كما مبين في شكل (7-16).



شكل (7-16)

6. بعد ان قمت باختيار القرص المرن سترى ان البرنامج سيتوقف وتظهر رسالة خطاء ، تخبرك برقم الخطاء وكذلك بنوع الخطا. كما مبين في الشكل (7-17).

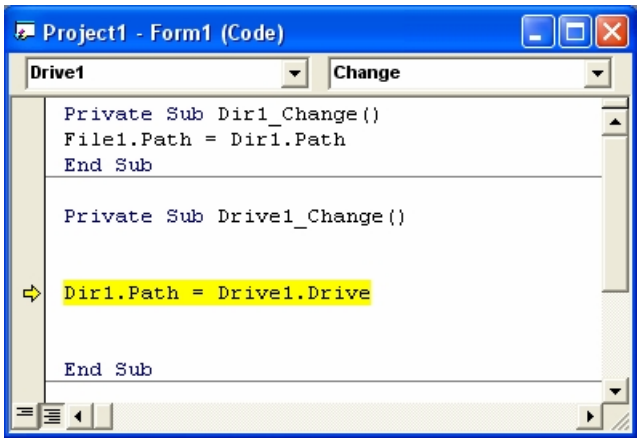


شكل (7-17)

7. انقر زر Debug لرؤية مكان الخطا أي الجملة البرمجية التي تسببت بالخطا . واذا نقرت على زر End فانه سيقوم بانهاء البرنامج. في حال نقر الزر Debug فان الجملة التي تسببت بالخطا هي

```
Dir1.Path = Drive1.Drive
```

حيث سترى انها تحاط بمربع اصفر (أي تحاط الجملة التي تسببت بالخطا باللون الاصفر). كما مبين في الشكل (7-18).



شكل (18-7)

8. لاصلاح الخطا يجب اما ادخال قرص في الـ CDROM ولكن هذا ليس حل لان البرنامج سيتوقف كلما كان الـCDROM لا يحتوي على قرص . لذلك يجب علينا اصلاح المشكلة برمجيا وذلك باخبار البرنامج ماذا يفعل في مثل هذه الحالة (في حالة عدم وجود قرص) سنقوم بكتابة جمل برمجية لمعالجة الخطا.
9. انقر زر ايقاف البرنامج من شريط الادوات ، لايقاف البرنامج عن العمل.
10. انقر الان مربع سرد محركات الاقراص Drive1 نقرا مزدوجا.

يظهر الاجراء الحدثي Drive1_Change في اطار شفيرة البرنامج:

11. اكتب الجملة البرمجية التالية فوق الجملة Dir1.Path = Driver1.Drive كما مبين:

```
On Error GoTo ErrFixer:
```

تنشط هذه الجملة معالج الاخطاء وتبلغ فيجوال بيسك اين عليه الذهاب اذا حصل خطا تشغيل . ستضيف الان معالج الاخطاء ErrFixer الى اسفل الاجراء الحدثي .

12. انتقل الان الى سطر واحد تحت الجملة Dir1.Path=Drive1.Drive واكتب الجمل الاتية:

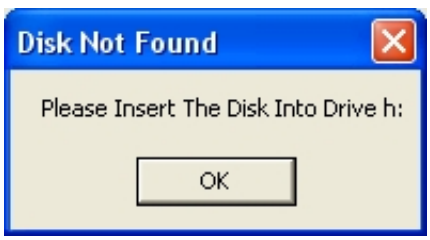
```
Exit Sub

ErrFixer:
    If Err.Number = 68 Then  ' No disk in the CDROM
        MsgBox ("Please Insert The Disk Into Drive " & Drive1.Drive), ,
"Disk Not Found"
     Resume
     End If
```

يختبر تعبير الجملة If … Then الشرطي الخاصية Err.Number ليرى ان كانت تحتوي على الرقم 68 (الرقم الذي يعاد دائما كلما كان محرك الاقراص فارغ "غير متوفر"). اذا كان قد حصل خطا في القرص ، يُعطى المستخدم فرصة لاصلاحه (اما باغلاق باب محرك الاقراص او بادخال قرص اذ لم يكن هناك قرص). ثم متابعة عملية التحميل (محاولة تنفيذ الجملة Dir1.Path=Drive1.Drive مرة ثانية بواسطة الكلمة Resume).
يعرض معالج الاخطاء رسالة للمستخدم ويحول دون توقف البرنامج بشكل نهائي.

اذا لم يعترض البرنامج أي مشاكل في القرص، يتابع البرنامج الى ان تنهي الجملة Exit Sub الاجراء الحدثي. الجملة Exit Sub هي جملة ذات استعمال عام، يمكنك استعمالها لانهاء أي اجراء في فيجوال بيسك قبل تنفيذ الجملة End Sub . في هذه الحالة ، تمنع الجملة Exit Sub اشتغال معالج الاخطاء بعد ان يكون البرنامج قد فتح محرك الاقراص بنجاح.

13. اختر الامر File> Save Photo Browser As واحفظ النموذج بالاسم Final Photo Browser .
14. اختر الامر File > Save Project As واحفظ المشروع بالاسم Final Photo Browser.
15. افتح باب محرك الاقراص واتركه مفتوحا .
16. انقر زر التشغيل لتشغيل البرنامج.
17. اذهب الى مربع سرد محركات الاقراص Driver واختر محرك الاقراص أي الـ CDROM.

يصدر فيجوال بيسك خطا تشغيل ويعرض معالج الاخطاء رسالة الخطا التالية:



18. ادخل أي قرص في محرك الاقراص ثم اغلق الباب .
19. انقر OK لاغلاق معالج الاخطاء واعادة محاولة التحميل.
20. انقر زر الاغلاق في البرنامج لانهاء البرنامج.

ملاحظة :

اذا عجزت عن الخروج من حلقة خطا ما، اضغط Ctrl + Break من لوحة المفاتيح.

خطوة متقدمة : تحديد مدة زمنية لاعادة المحاولة

هناك طريقة اخرى يمكنك اعتمادها في معالج الاخطاء هي تكرار العملية بضع مرات ثم تخطي المشكلة اذا لم يتم حلها. على سبيل المثال في المثال السابق عندما تم عرض رسالة الخطا ، ولم يقم المستخدم بادخال أي قرص في محرك الاقراص ، عند ذلك ستبقى المشكلة قائمة أي سيبقى الخطا قائم وايضا سيحاول معالج الاخطاء معالجتها لذلك يقوم بعرض الرسالة مرة ثانية وثالثة ورابعة الى ان يتم ادخال قرص في محرك الاقراص او ان تقوم بانهاء البرنامج.

لحل مثل هذه المشكلة يستعمل معالج الاخطاء التالي متغيرا عداداً اسمه Tries_No ، لتعقب عدد المرات التي عرضت فيها رسالة الخطا ثم يتخطى البرنامج جملة Dir1.Path=... اذا فشلت مرتين:

```
Tries_No = 0
On Error GoTo ErrFixer:
Dir1.Path = Drive1.Drive

Exit Sub

ErrFixer:
  If Err.Number = 68 Then  'No disk in the CDROM
    MsgBox ("Please Insert The Disk Into Drive " & Drive1.Drive), ,
"Disk Not Found"
    Tries_No = Tries_No + 1
     If Tries_No >= 2 Then
        Resume Next
     Else
        Resume
     End If
  End If
```

انه اسلوب مفيد اذا كان الخطا الذي تعالجه هو مشكلة يمكن ان يصححها المستخدم من حين الى اخر. من المهم ان تتذكر ان الجملة Resume تعيد محاولة الجملة التي تسببت بالخطا وان الجملة Resume Next تتخطى الجملة الخطا وتنتقل الى السطر الذي يليها في الاجراء

الحدثي. عندما تستعمل الجملة Resume Next ، تاكد من ان الجملة التالية هي التي تريد تنفيذها .

أي بمعنى اخر اذا استعملت الجملة Resume Next فانك سوف تهمل الخطا وتكمل البرنامج بعد الجملة التي تسببت بالخطا.

أي يمكن كتابة الجملة البرمجية التالية فقط بدل جمل الخطا السابقة :

```
On Error Resume Next:
Dir1.Path = Drive1.Drive
```

حيث عند حدوث خطا فان البرنامج سوف يقوم بتجاهل جملة الخطا أي يتجاهل جملة الـ Dri1.Path = Drive1.Drive وينفذ الجملة التي تليها .

الجملة التي تلي جملة Dri1.Path = Drive1.Drive هي End Sub. أي سوف يقوم بإنهاء الاجراء الحدثي.

تمارين عملية

ستتعلم في هذه التمارين كيفية :

- **عمل حاسبة كحاسبة ويندوز.**
- **استكمال عمل معالج النصوص Text Handler بشكل اكبر.**
- **استكمال عمل البرنامج Photo Browser بشكل اكبر وافضل.**

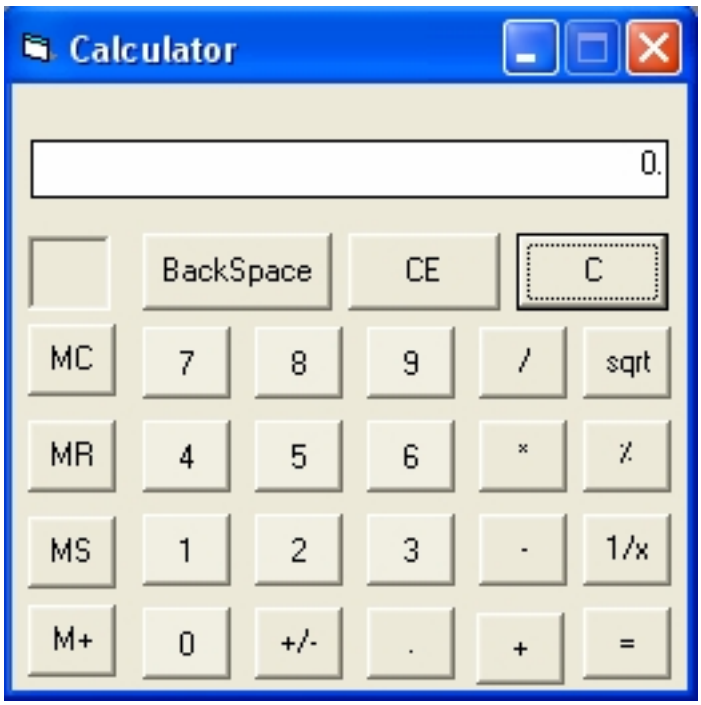
ستقوم في هذه التمارين بتحسين مهارتك بشكل كبير حيث ستقوم باستكمال عمل بعض البرامج التي قمت بعملها سابقا لكن ستقوم الان باضافة خواص اخرى واستكمال عملها بحيث تستطيع استخدامها والاستفادة منها . كما ستتعلم كيفية انهاء البرنامج وجعله متكامل أي جعله جاهز لاستخدام من قبل أي شخص.

النقطة المهمة في هذه التمارين هي اني فقط ساعطيك شكل البرنامج وشرح حول الخواص التي يستطيع عملها، أي سوف لان اقوم بشرح الشفيرة البرمجية لاي برنامج حيث ستقوم انت بعمل ذلك .

التمرين الاول

عمل حاسبة شبيهة بحاسبة ويندوز.

اليك شكل الحاسبة بعد ان تقوم بعملها، ولا اضن اني بحاجة لشرح ما تقوم به الحاسبة من عمل!؟.



التمرين الثاني

استكمال عمل معالج النصوص Text Handler بشكل اكبر.

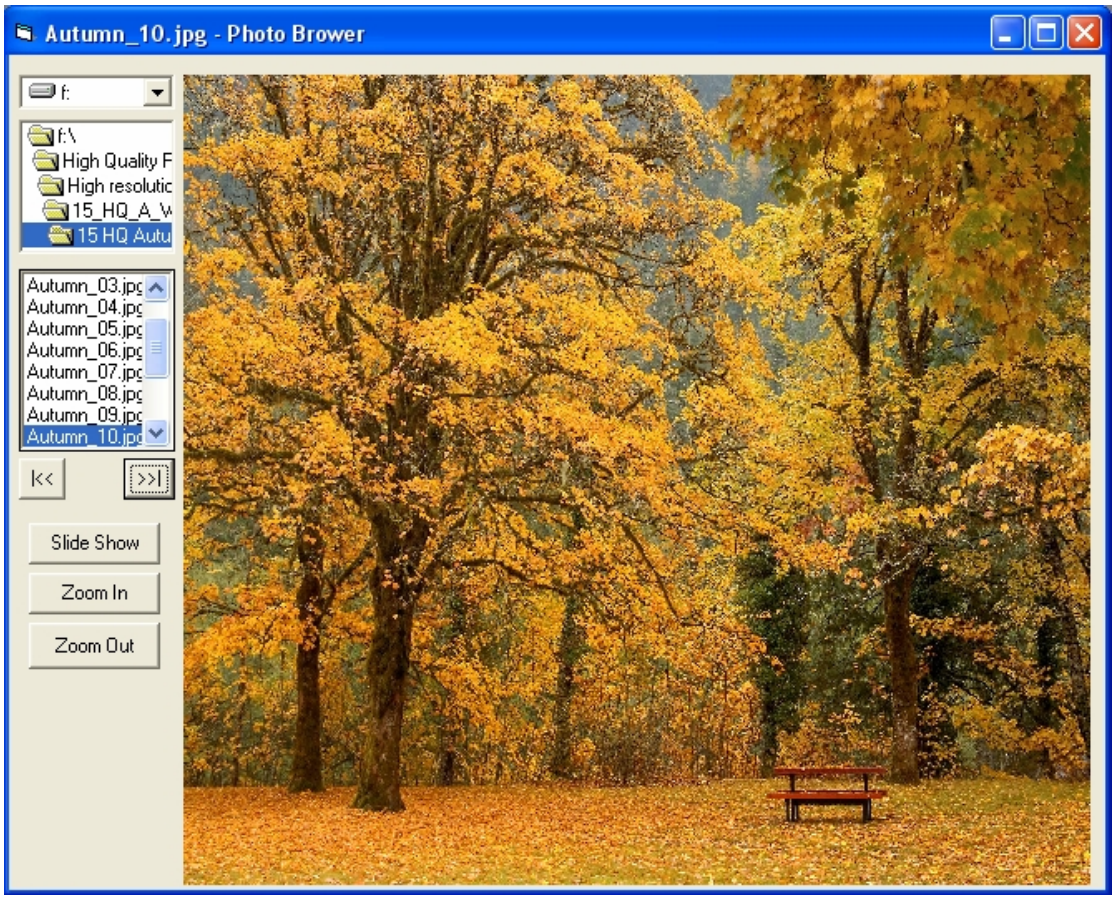
اليك شكل معالج النصوص الذي سوف تقوم بعمله

يمكنك ان تلاحظ انه يحتوى على عدة خواص حيث ستقوم بتطبيق دوال استخراج النصوص Left و Right و Mid و كذلك دوال حذف المسافات Trim كذلك سوف تستخدم كائن حافظة ويندوز ClipBoard حيث ستعمل ازرار التحرير الثلاثة Copy و Cut و Paste كذلك ستقوم بتنسيق النص باستخدام ادوات التنسيق Bold و Italic الخ كما ستقوم بعمل قائمة منسدلة لحجم الخط كما في برامج تحرير النصوص.

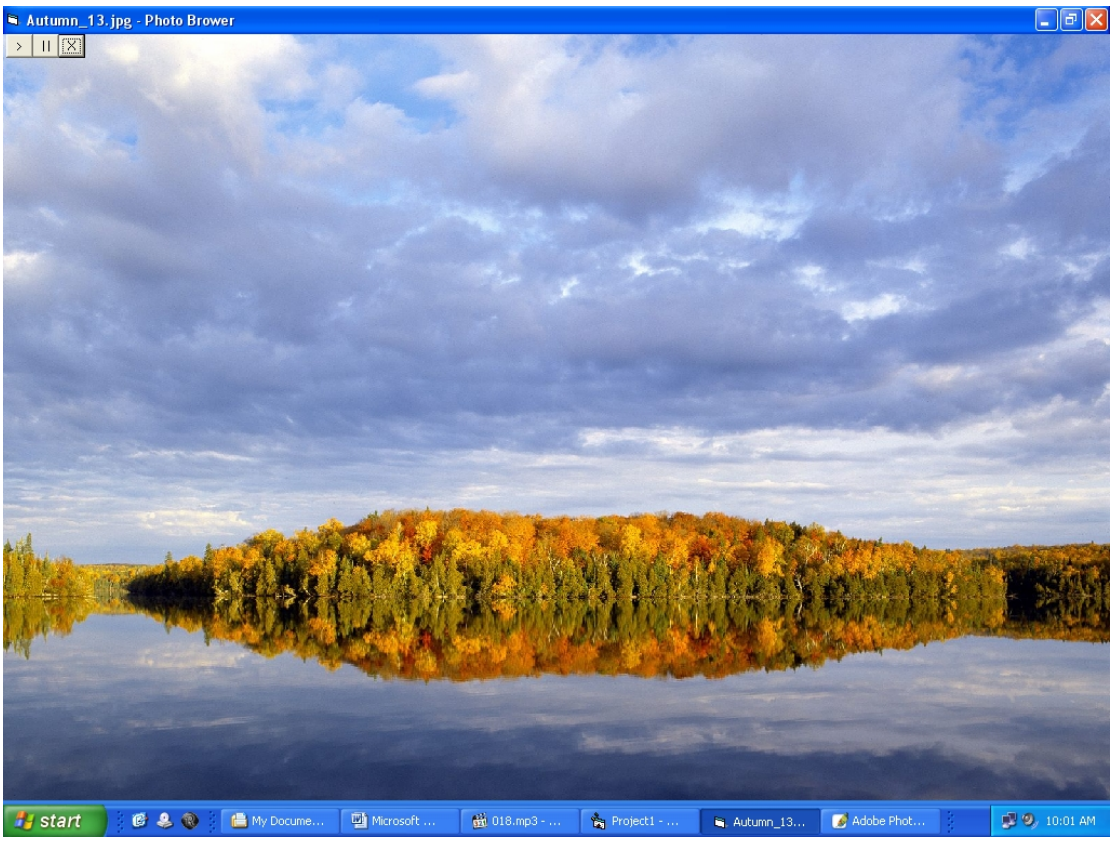
التمرين الثالث

استكمال عمل البرنامج Photo Browser بشكل اكبر وافضل.

اليك شكل برنامج متصفح الصور Photo Browser الذي سوف تقوم بعمله:



يمكنك ان تلاحظ انه يحتوى على عدة خواص كاالسابق والتالي وكذلك Zoom In و Zoom Out (تكبير وتصغير الصورة) ايضا يحوي على الامر عرض كشرائح (Slide Show) حيث تعمل هذا الامر على اخفاء جميع كائنات النوذج عدا مربع الصورة ويقوم بتكبير النموذج ليملا الشاشة ويعرض الصور حسب فترة زمنية معينة. انظر الشكل التالي:



يمكنك ان تلاحظ ان النموذج يعرض اسم الصورة في شريط العنوان (في كلا الحالتين أي في الشكل السابق والشكل التالي ايضا) كما يحتوي الشكل الثاني على ثلاث ازرار وهي اغلاق و ايقاف مؤقت و تشغيل، حيث يستخدم زر اغلاق للرجوع الى النموذج الرئيسي أي يعرض بقية الكائنات اما زر ايقاف مؤقت فانه يوقف الصورة مؤقتا حتى يتم الضغط على زر تشغيل.