

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/261913367>

Visual Basic كتاب لغة

Book · January 2007

CITATIONS

0

READS

1,012

1 author:



Hussein M. Salman

University of Babylon

5 PUBLICATIONS 10 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



blind source separation [View project](#)



Independent Component Analysis [View project](#)

جامعة بابل
كلية الهندسة

البرمجة بلغة

البيساي المرئي

Visual basic

إعداد

المبرمج/حسين محمد سلمان

٢٠٠٧

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
الرَّحْمَنُ ، عِلْمُ الْقُرْآنِ ، خَلْقُ
الْإِنْسَانِ ، عِلْمُهُ الْبَيَانِ

صدق الله العلي العظيم

الرحمن (١-٤)

فهرس المحتويات Contents

٣	المحتويات
٦	الاهداء
٧	شكر وتقدير
٨	مقدمة
١٧-١٠	الفصل الاول/بيئة التطوير Visual Basic
١٠	١-١ مقدمة
١٠	٢-١ الدخول الى اللغة
١١	٣-١ فتح مشروع جديد
١١	٤-١ فتح مشروع قديم
١١	٥-١ اضافة نموذج جديد
١٣	٦-١ شريط القوائم
١٥	٧-١ الشريط القياسي
٢٣-١٨	الفصل الثاني/نوافذ الادوات والخصائص
١٨	١-٢ نافذة تصميم النموذج
١٨	٢-٢ مربع الادوات
٢١	٣-٢ نافذة الخصائص
٣٦-٢٤	الفصل الثالث/بناء مشروع جديد
٢٤	١-٣ مراحل تصميم المشروع
٣٣	٢-٣ مثال توضيحي
٣٤	٣-٣ تسمية الكيانات
٥٠-٣٧	الفصل الرابع/المتغيرات والبيانات
٣٧	١-٤ الثوابت والمتغيرات
٣٧	٢-٤ تعريف الثوابت والمتغيرات
٣٧	٣-٤ اشكال تصريح البيانات
٣٩	٤-٤ انواع البيانات

٤١	٥-٤ ادخال واخراج البيانات
٤١	٦-٤ مثال توضيحي ١
٤٤	٧-٤ مثال توضيحي ٢
٤٦	٨-٤ دالة الادخال InputBox
٤٧	٩-٤ اخراج البيانات
٤٨	١٠-٤ دالة الاخراج MsgBox
٦٩-٥١	الفصل الخامس/اتخاذ القرار وعبارات التكرار
٥١	١-٥ مقدمة
٥١	٢-٥ عوامل المقارنة
٥٢	٣-٥ العوامل المنطقية
٥٢	٤-٥ عبارة IF الشرطية
٥٥	٥-٥ مثال توضيحي
٥٦	٦-٥ الجملة الشرطية Select Case
٥٧	٧-٥ مثال توضيحي
٥٩	٨-٥ عبارة التكرار Do
٦٠	٩-٥ مثال توضيحي
٦٢	١٠-٥ عبارة For التكرارية
٦٣	١١-٥ امثلة توضيحية اضافية
٩١-٧٠	الفصل السادس/المصفوفات
٧٠	١-٦ مقدمة
٧٠	٢-٦ مصفوفات المتغيرات
٧٠	٣-٦ انشاء المصفوفة
٧٠	٤-٦ الصيغة العامة للمصفوفة
٧٢	٥-٦ التعامل مع المصفوفة بداخل شفرة البرنامج
٧٢	٦-٦ امثلة توضيحية حول المصفوفات
٨١	٧-٦ العمليات الحسابية والمصفوفات

٨٨	٨-٦ ضرب المصفوفات الثنائية
٩٧-٩٢	الفصل السابع/الدوال والبرامج الفرعية
٩٢	١-٧ مقدمة
٩٢	٢-٧ الدوال
٩٥	٣-٧ دوال التعامل مع النصوص
٩٦	٤-٧ البرامج الفرعية
١٠٨-٩٨	الفصل الثامن/الملفات وقواعد البيانات
٩٨	١-٨ الملفات
٩٨	٢-٨ الملفات المتتابة
١٠١	٣-٨ الدوال المستخدمة مع الملفات
١٠١	٤-٨ الملفات ذات الدخول العشوائي
١٠٥	٥-٨ قواعد البيانات
١٠٦	٦-٨ طرق التعامل مع قواعد البيانات
١٠٨	٧-٨ الاداة Data
١٠٩	المصادر

إهداء

إلى ينبوع الحنان:

أبي وأمي.....

إلى سندي في الحياة:

إخوتي وأخواتي.....

إلى عطر حياتي:

زوجتي.....

إلى فلذات كبدي:

زهراء ، فاطمة ،.....

حسين

شكر وتقدير

أقدم خالص شكري وتقديري إلى كل من ساعدني وشجعني في اعداد هذا الكتاب واخص بالذكر منهم شقيقي وسندي في الحياة د.جاسم(ابومحمد) وأخي وزميلي العزيزالمبرمج فراس الطريحي وكذلك زملائي في الكلية منهم د.سعد سفاح والسيدة انتظار جابر ؛ أقدم شكري وامتناني إلى كافة الزملاء في قسم تدريب الحاسبات سابقا في كلية الهندسة .
كذلك انحني إجلالا للأستاذ الفاضل الدكتور المهندس ستاريدرسدخان مدير قسم الشؤون العلمية والعلاقات الثقافية في رئاسة جامعة بابل وذلك لتشجيعه اللامحدود لي في إعداد هذا الكتاب.

المقدمة

ان التطور الهائل في استخدام الحاسبات في كافة مجالات الحياة يدعو الى ان يواكب الجميع هذا التطور ، ولما كانت لغات البرمجة هي الشئ الملموس في استخدام هذا الجهاز العملاق في معناه والصغير في حجمه حيث ان هذه اللغات هي الواجهة بين المبرمجين من جهة وبين جهاز الحاسبه من جهة اخرى.

الذي دفعني الى اعداد هذا الكتيب هو الحاجة الملحة الى مصدر مختصر ومفيد للمساعدة في تعلم البرمجة بلغة البرمجة "البيسك المرئي Visual Basic " حيث ان هذه اللغة من اللغات التي يمكن استخدامها في كافة المجالات العلمية والهندسية ؛ كذلك فان النقص الهائل في مصادر هذه اللغة العربية والعراقية بالذات وحاجة الطلبة اليها كل هذا دفعني الى اعداد هذا الكتيب.

انني قمت بتدريس وتدريب الطلبة هذه اللغة لاكثر من ثلاث سنوات متواصلة لطلبة المرحلة الثانية في قسم الهندسة المدنية/كلية الهندسة منذ عام ٢٠٠٢ وذلك مع أكفاء الاساتذة في القسم .

يتالف الكتيب من ثمانية فصول الاول يتضمن شرح مفصل لبيئة هذه اللغة واشروطها الموجودة في الواجهة الرئيسية .

الفصل الثاني تناول نافذة الادوات ونافذة الخصائص حيث تعد هاتين النافذتين من اهم نوافذ هذه اللغة ويجب على جميع مستخدمي هذه اللغة تعلم واستخدام هاتين النافذتين.

الفصل الثالث تضمن كيفية بناء مشروع جديد وكيفية التعامل معه من خلال مراحلها الثلاث مرحلة اعداد الواجهة ومرحلة اعداد الخصائص ومرحلة البرمجة .

الفصل الرابع تم التعرف بشكل مفصل على انواع المتغيرات والبيانات وكيفية تعريفها والتعامل معها.

الفصل الخامس تم شرح العبارات الشرطية وعبارات التكرار وكيفية استخدامها والتعامل معها .

الفصل السادس تناول بشكل مفصل المصفوفات بنوعها احادية الابعاد ومتعددة الابعاد.

الفصل السابع تناول كيفية تعريف وتصريح الدوال والبرامج الفرعية وكيفية التعامل معها وبشكل مفصل.

الفصل الثامن تناول الملفات بشكل مفصل كذلك تناول كيفية التعامل مع قواعد البيانات وبشكل مفصل.

ان جميع ماتناولته هذه الفصول مدعم بالامثلة التوضيحية وفي جميع الحالات المطروقة ، اما عن الازطاء فانا في اعتقادي لا يوجد شيء من كتاب او ملزمة او أي مبحث هو خال من الازطاء وكامل لان العلم في جميع المجالات يتطور ويتقدم ويتغير يوم بعد الاخر كذلك فان كل كتاب او بحث لا يخلو من الازطاء والهفوات الطباعية احيانا والعلمية احيان اخرى .

الفصل الاول/بيئة التطوير Visual Basic

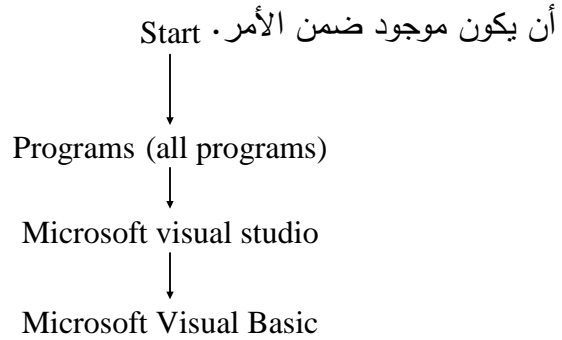
١-١-١- مقدمة

إن لغة البرمجة بلغة Visual Basic وتقرأ بـ (فيجوال بيسك) ومعناها البيسك المرئي حيث إن Visual تعني المرئي و BASIC تعني لغة بيسك الاعتيادية (Basic Applications System for Interchange Codes) ، تكاد تكون هذه اللغة أهم لغات البرمجة في نظام التشغيل Windows وهي من اللغات التي تعتمد على مبدأ برمجة الهدف Object Oriented Programming ومختصرها هو (OOP) وتسمى أيضا بالبرمجة الكيانية .

إن أهمية هذه اللغة تأتي من سهولة الاستخدام وسهولة كتابة البرنامج فيها وكذلك متانة هذه اللغة وتعاملها مع تطبيقات أخرى منها تطبيقات قواعد البيانات وتطبيقات الرسم بالحاسبة وتطبيقات الارتباط بشبكات الاتصالات.

١-٢-١- الدخول إلى اللغة Visual Basic

يتم الدخول إلى هذه اللغة من خلال الملف Microsoft Visual Basic وهذا الملف أما



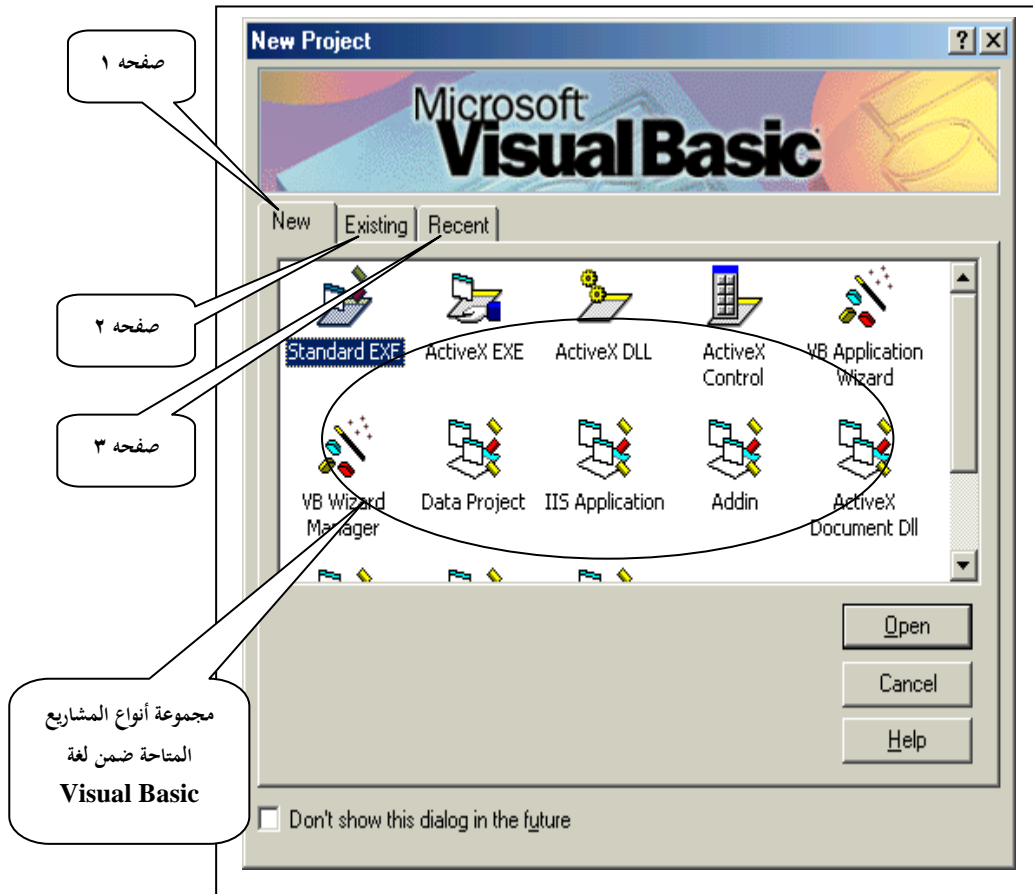
أو عن طريق ايقونة موجودة على سطح المكتب Desktop تحت عنوان Microsoft Visual Basic .

عند الدخول لبيئة هذه اللغة سوف تظهر نافذة فيها مجموعة من الرموز التي تبين كل الأنواع المتاحة لعمل المشروع الذي تريد عمله، إن أهم هذه الأنواع هو النوع standard.exe وهو النوع

القياسي كما موضح في الشكل رقم (١)

تحتوي النافذة المبينة في الشكل رقم (١) على ثلاث صفحات هي :-

١-صفحة ١ واسمها New :حيث تظهر فيها كل أنواع المشاريع المتاحة في لغة Visual Basic، نقوم باختيار النوع الأول فيها وهو قياسي standard exe ونضغط على (open) الأنواع الأخرى تخص استخدامات خاصة يمكن أن تعمل في بيئة Visual Basic) سوف تظهر لنا إطار المشروع (أوما يسمى بمستوعب المشروع). المشروع الذي تم فتحه تحت اسم project1 وبدخله نموذج اسمه form1 ، هذه الأسماء هذه يضيفها معالج اللغة بشكل تلقائي فإذا فتحنا مشروع آخر فانه سوف يحمل اسم هو project2 وبدخله نموذج اسمه form1 .



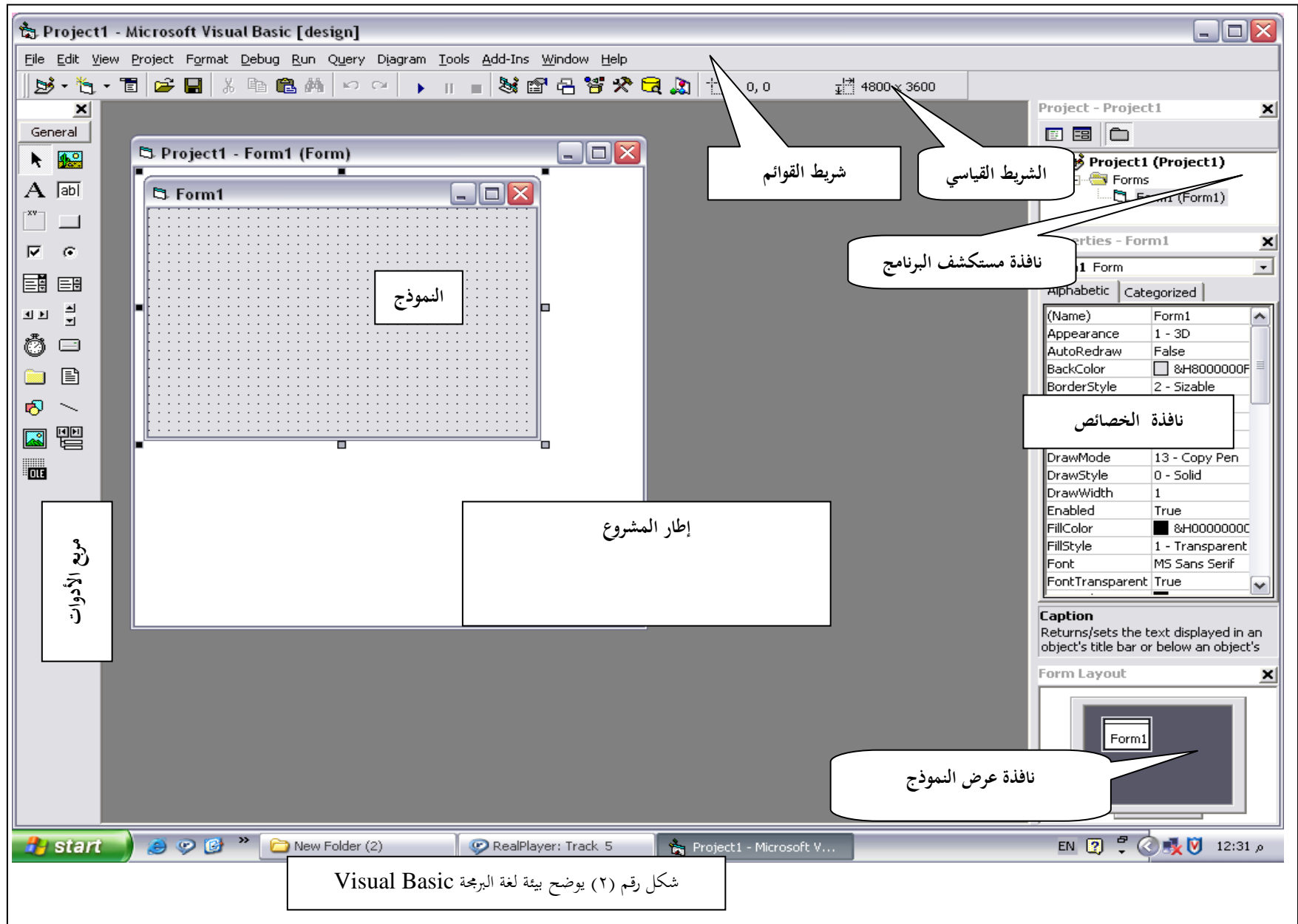
شكل رقم (١) يوضح أنواع المشاريع المتاحة ضمن بيئة لغة Visual Basic

٢- صفحه ٢ واسمها Existing :حيث تحوي على أسماء المشاريع الموجودة والتي تم عملها في وقت سابق.

٣- صفحه ٣ واسمها Recent :حيث تحوي هذه النافذة على آخر مجموعه من المشاريع التي تم فتحها او عملها او تم التعامل معها .

، نضغط على المفتاح open للموافقة على فتح مشروع جديد.

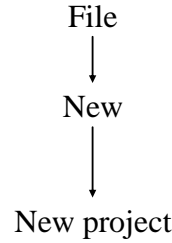
تحتوى بيئة البرمجة فبجوال بيسك على أدوات برمجة لمساعدتك على بناء برامج Visual Basic كما موضحة وبالتفصيل في الشكل رقم(٢).



١-٣- فتح مشروع جديد

يمكن فتح مشروع جديد :

١- من النافذة الموضحة في الشكل رقم (٢) واختيار

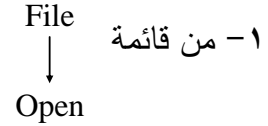


أو

٢- باختيار أول رمز من اليسار في الشريط القياسي (الذي يأتي شرحه بالتفصيل لاحقاً). عند اختيار أي من الاختيارين سوف تظهر لنا النافذة الموضحة في الشكل (١) .

١-٤- فتح مشروع قديم

يمكن فتح مشروع قديم تم عمله (تصحيحه) باستخدام لغة Visual Basic ويمكن فتحه باستخدام طريقتين هما:-



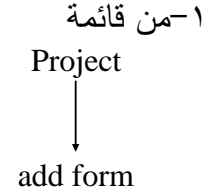
٢- باستخدام الشريط القياسي وذلك بالضغط على الأيقونة التي تحمل شكل مجلد (folder) مفتوح.

سوف تظهر نافذة فيها جميع المشاريع المخزونة في الحاسبة والتي بامتداد (VBP) والموضحة في الشكل رقم (١) الصفحة الثانية Existing. (جميع المشاريع التي تصمم باستخدام (VB6) يكون امتدادها (VBP)) يعد فتح تحديد المشروع المراد إظهارها والضغط على الأمر OPEN.

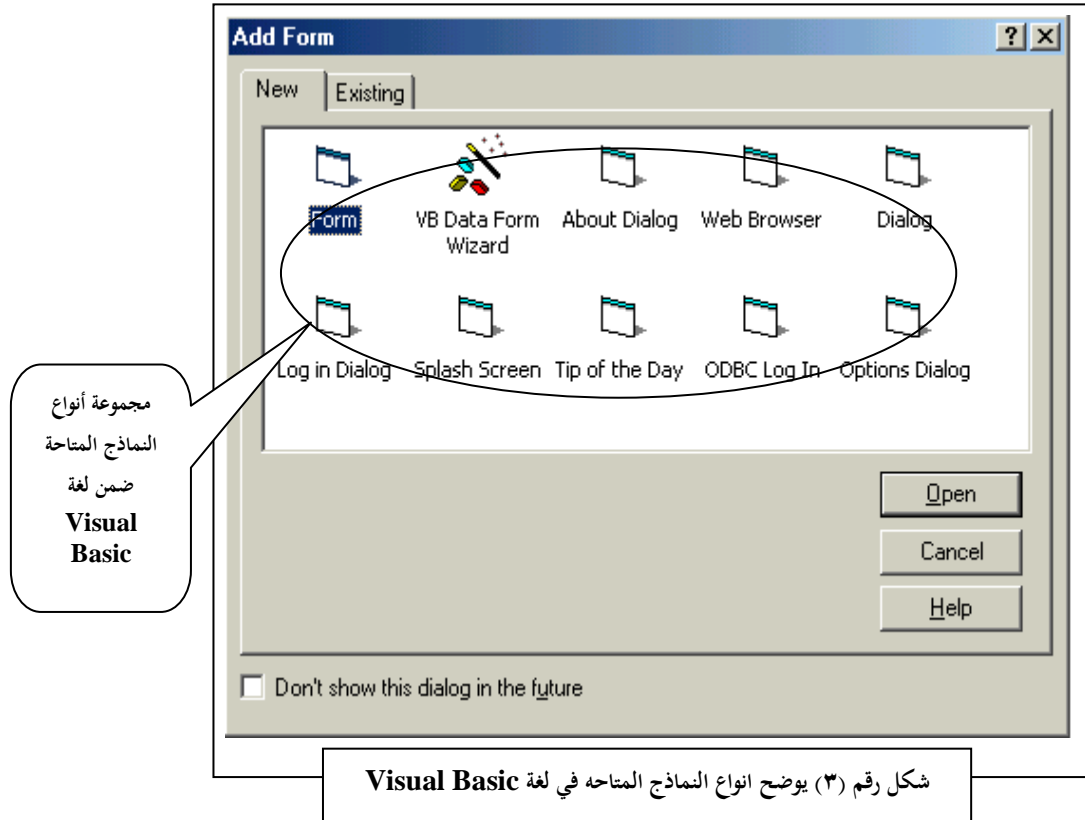
١-٥- إضافة نموذج جديد:

كل مشروع يتم فتحه فإنه يحتوي على نموذج بداخله (كما في الشكل رقم ٢) وهذا شكل تلقائي ولا يمكن إن يكون هنالك مشروع لا يحتوي على نموذج ويمكن أن يحتوي كل مشروع على أكثر من نموذج واحد.

يمكن إضافة نموذج جديد إلى المشروع الحالي (الموجود حالياً على الشاشة) وذلك بواسطة طريقتين هما:-



٢- باستخدام الشريط القياسي وذلك بالضغط على الأيقونة الثانية من اليسار في الشريط القياسي. في كلا من الطريقتين سوف تظهر لنا نافذة فيها جميع النماذج المتاحة في لغة Visual Basic كما في الشكل رقم (٣)



يتضمن الشكل اعلاه صفحتين الاولى واسمها New والتي تحتوي على انواع النماذج المتاحة في لغة Visual Basic ، وصفحه اخرى اسمها Existing والتي تحتوي على النماذج التي تم انشاؤها في وقت سابق. نقوم بتاثير النوع الأول (والنوع المراد إنشاؤه) ثم ونضغط OPEN

١-٦- شريط القوائم:-

يزود شريط القوائم وصولا إلى معظم الأوامر التي تتحكم ببيئة المبرمج فيجوال بيسك حيث تعمل الاوامر والقوائم مثلما تعمل في كل البرامج المتوافقة مع نظام التشغيل windows و يمكن الوصول إليها بواسطة الفأرة أو لوحة المفاتيح. عند فتح مشروع جديد فسوف تظهر نافذة تحتوي بطبيعة الحال على مجموعة في النوافذ والأشرطة. قد تظهر الواجهة ولا تحتوي على أي نافذة أو شريط إلا شريط القوائم الذي يظهر بشكل تلقائي مع أي تطبيق ضمن نظام التشغيل WINDOWS، يمكن إظهار أي نافذة أو شريط من خلال قائمة VIEW، وفيما يلي شرح لمحتويات شريط القوائم كل أمر في شريط القوائم هو عبارة عن قائمة أوامر وإيعازات:-

١-٦-١- القائمة FILE:

تحتوي هذه القائمة على مجموعة من الأوامر الخاصة بالتعامل مع المشاريع مثل NEW لإنشاء مشروع جديد و OPEN لفتح مشروع سابق، وحفظ المشاريع عن طريق SAVE أو SAVE AS، وحذف المشروع REMOVE وخلق نسخة تنفيذية من التطبيق MAKE PROJECT.EXE بالإضافة إلى أمر الطباعة PRINT والخروج Exit .

١-٦-٢- القائمة EDIT:

تحتوي هذه القائمة على أوامر متشابهة لما تحتويه قائمة EDIT في مختلف التطبيقات، فهي تحتوي أوامر القص والنسخ واللصق والحذف للنصوص ولعناصر التحكم وأوامر البحث find والاستبدال replace .

١-٦-٣- القائمة VIEW:

تهتم أوامر هذه القائمة بالتحكم بظهور مختلف نوافذ VB، إذ يمكن من خلال هذه القائمة إخفاء وإظهار عدة نوافذ من بيئة VB .

١-٦-٤- القائمة project:

تحتوي هذه القائمة مجموعة من الأوامر الإضافية مكونات جديدة إلى المشروع مثل نماذج جديدة ووحدات نمطية (MODULE) وملفات أخرى.

١-٦-٥- القائمة FORMAT :

تستخدم هذه القائمة لتحديد هيئة وحجم عناصر التحكم الموجودة على النموذج والتباعد فيما بينها.

١-٦-٦- القائمة DEBUG :

تحتوي هذه القائمة على مجموعة أوامر خاصة باختبار التطبيق، وتقوم بإظهار النتائج ضمن نوافذ مخصصة في بيئة VB . كما يمكن من خلال هذه القائمة تنفيذ التعليمات البرمجة خطوه خطوه، وبهذا يمكن مراقبة أداء المشروع.

١-٦-٧- القائمة RUN:

من خلال هذه القائمة يمكن تنفيذ التطبيق ضمن بيئة VB ، يتم بدء التنفيذ باستخدام الأمر START أو F 5 ويستخدم الأمر break لإيقاف التطبيق توقفاً مؤقتاً في المرحلة الحالية (من خلال التنفيذ) ، والأمر RESTART لإعادة التنفيذ من البداية إما الأمر CONTINUE فيستخدم لاستئناف التنفيذ من النقطة التي تم فيها التوقف ، الأمر END هو لإنهاء التنفيذ .

١-٦-٨- القائمة QUERY :

تكون الأوامر من هذه القائمة متاحة فقط عندما نقوم بإنشاء استعلام من خلال الأداة

MICROSOFT QUERY BUILDER

١-٦-٩- القائمة DIAGRAM:

تكون أوامر القائمة متاحة عندما يتفاعل التطبيق مع SQL SERVER أو ORACLE لإنشاء أو لتعديل مخطط قاعدة البيانات.

١-٦-١٠- القائمة TOOLS :

تحتوي هذه القائمة على أدوات مختلفة ومتنوعة مثل الأمر OPTION الذي يفتح نافذة حواريه لتعديل خصائص بيئة التطوير.

١-٦-١١- القائمة ADD-In:

تحتوي على أوامر متعلقة بتقنيات وكائنات خارجية يتم دمجها مع بيئة التطوير. كما تحتوي الأمر Visual Data Manager الذي يتم من خلال الوصول إلى برنامج مدير البيانات . Visdata

١-٦-١٢ - القائمة WINDOW :

يتم من خلال هذه القائمة إدارة نوافذ بيئة التطوير VB.

١-٦-١٣ - القائمة HELP:

يتم من خلال هذا الأمر استدعاء نظام المساعدة الفوري للغة Visual Basic وهو (MSDN).

١-٧-١ - الشريط القياسي للغة VB:

هذا الشريط إذ لم يكن موجودا فيمكن اضراره من خلال القائمة View/toolbars. هذا الشريط يحتوي على جميع الايعازات المهمة في لغة VB حيث يمكن التعرف على أسماء هذه الايكونات من خلال نص المساعدة السريع الذي يظهر عند وضع مؤشر الفأرة على الرمز وبدون الضغط عليه وهي كما يلي بدا من اليسار:- (هذه الأوامر عبارة عن رموز (ICONS)).

١-٧-١-١ - مشروع جديد: استخدام هذه الايقونة يؤدي الى خلق مشروع جديد بالاضافة للمشروع الحالي.

١-٧-٢ - إضافة نموذج: تؤدي هذه الايقونة الى اضافة نموذج جديد الى المشروع الحالي.

١-٧-٣ - تحرير قائمة menu:- هي أداة رسومية تدير البرامج في القوائم. تتيح هذه الايقونة للمستخدم اضافة قوائم جديدة وتعديل القوائم الموجودة وإعادة ترتيبها وحذف القوائم القديمة. كما يتيح لك اضافة مؤثرات خاصة على القوائم، كمفاتيح الوصول وعلامات الاختيار واختصارات لوحة المفاتيح.

١-٧-٤ - فتح مشروع موجود في الحاسبه ومصمم في وقت سابق.

١-٧-٥ - خزن المشروع والنموذج .

١-٧-٦ - قطع:- يتم تحديد جزء من شفرة المشروع او أي كيان وقطعه (أي إزالته من النموذج)

٧-٧-١ - نسخ:- يتم تحديد أي كيان ونسخه (هنا الكيان المستنسخ يحتفظ بشيء خصائص الكيان الاصلى).

٧-٧-٨ - لصق:- يتم لصق الجزء او الكيان الذى تم نسخه او قطعه.

٧-٧-٩ - بحث :- البحث عن كيان او كلمه .

٧-٧-١٠ - التراجع :- التراجع عن خطوه معينه الى الخطوه السابقه لها .

٧-٧-١١ - التقدم:- يستخدم هنا الامر بعد استخدام التراجع ، أي التقدم الى الخطوه التاليه بعد ان تراجعنا عنها .

٧-٧-١٢ - التنفيذ :- حت حالات هذا الامر يمكن تنفيذ المشروع او اعاده تنفيذ

RESTART

٧-٧-١٣ - التوقف Break :- تستخدم هذه الايقونه لايقاف التنفيذ ولكن ليس التوقف بشكل نهائى أي انه يمكن استئناف التنفيذ من النقطة التى تم التوقف فيها.

٧-٧-١٤ - الانتهاء END :- هنا يتم إنهاء التنفيذ بشكل قطعي أي إذا أردنا التنفيذ مرة أخرى فانه سوف يتم التنفيذ من البداية وليس من النقطة التى تم الانتهاء عندها.

٧-٧-١٥ - مستكشف المشروع PROJECT EXPLORER:- عند الضغط على هذه الايقونه فانه سوف تظهر نافذة صغيره على يمين الشاشة تحتوى على المشاريع الحالية التي تم بنائها في اللغة وكذلك النماذج الموجودة في كل مشروع حيث يمكن إظهار أي نموذج وذلك بالضغط عليه مرتين بمفتاح الفارة الأيسر.

٧-٧-١٦ - نافذة الخصائص Properties window :- هذه الايقونه تستخدم لإظهار نافذة خصائص تظهر عادة في يمين الشاشة تحتوى على خصائص الكيان الحالي وسوف نأتى على شرح هذه النافذة بشكل مفصل لاحقا.

٧-٧-١٧ - نافذة توضع النماذج FORM LAYOUT WINDOW:- هذه الأيقونة سوف تظهر نافذة في أسفل يمين الشاشة. حيث يمكن باستخدام هذه النافذة تحديد موضع ظهور النماذج على شاشة مستخدم عند تشغيل التطبيق.

٧-٧-١٨ - مستعرض الكيانات OBJECT BROWER :- تقوم هذه الايقونة بإظهار نافذة مستعرض الكيانات (الكائنات)، حيث من خلال هذه النافذة يمكن للمبرمج ان يستعرض

جميع الكيانات التي يستخدمها في عملية التطوير فيتعرف على خصائص وطرق واحداث كل كائن.

١-٧-١٩ - أدوات التحكم **control tools**:- بالنقر على هذه الايقونه سوف تظهر نافذة على يسار الشاشة يطلق عليها بنافذة الادوات (او مربع الادوات) تحتوى على عناصر تحكم أساسية (intrinsic controls) التي يمكن وضعها على النموذج فى مرحلة التصميم. هذه الادوات سوف ناتى على شرحها بالتفصيل لاحقا .

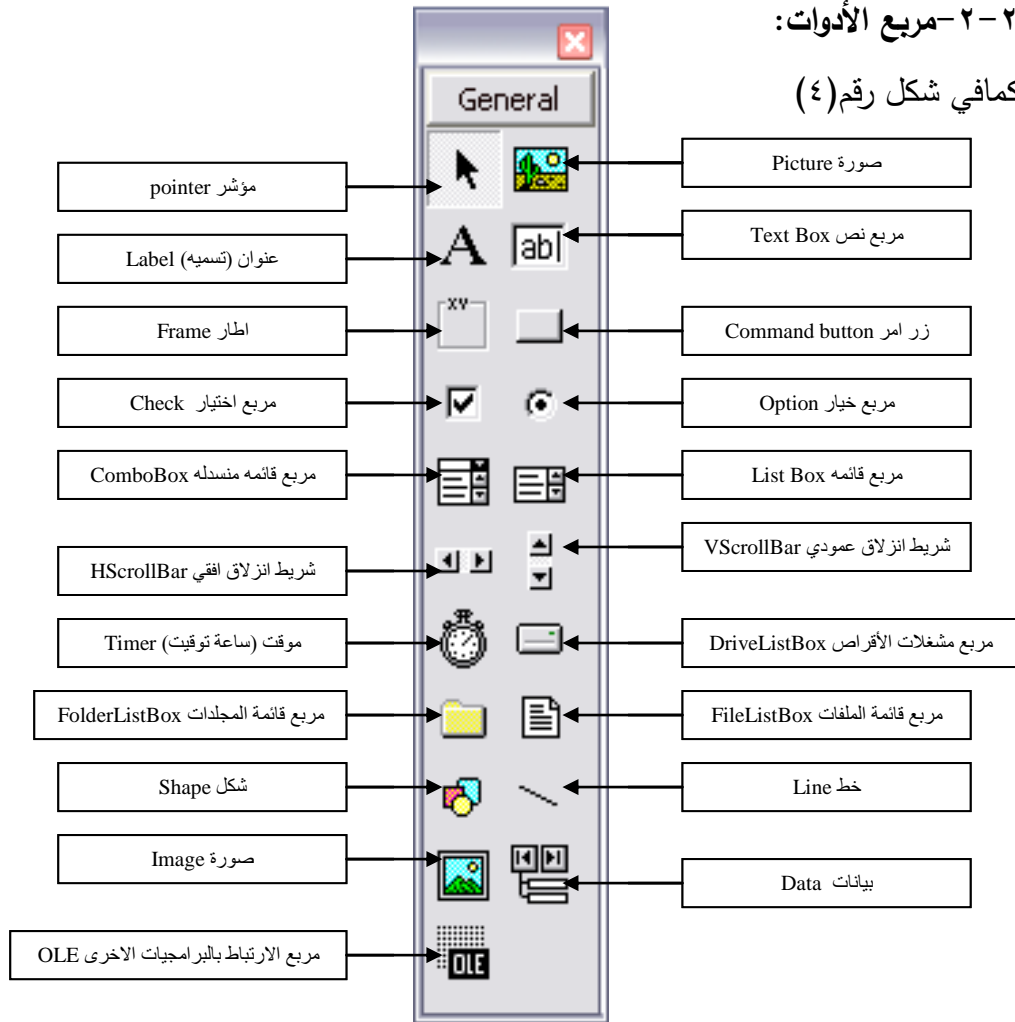
الفصل الثاني/نوافذ الادوات والخصائص

١-٢- نافذة تصميم النموذج

نلاحظ أن النموذج موجود على إطاره الخارجي مقابض وهي أشبه بمربعات صغيرة ذات لون مميز موجودة هذه المقابض في الزوايا الأربع وفي أواسط الأضلاع أي أن عددها ثمانية مقابض وهذا يعني أن الكيان الفعال حالياً هو النموذج بحيث أن مربع الخصائص يحتوي على خصائص الكيان الفعال حالياً وهو النموذج وفائدتها في تصغير وتكبير الشاشة .

٢-٢- مربع الأدوات:

كما في شكل رقم (٤)



شكل رقم (٤) يوضح صندوق الادوات

شكل الأداة	اسمها	معناها واستخدامها
	مؤشر Pointer	لا تعتبر هذه اداة من ادوات التحكم التي توضع على النموذج ولكنها تستخدم للتأشير على الكيانات الموجودة في النموذج
	مربع صورة Picture	تستخدم هذه الاداة لرسم مربع صورة على النموذج يمكن ان نضع فيه أي تشكيلة من صورة او مجموعة صور او صورة مع شكل هندسي أي انها تسمح لنا بتشكيل صورته حسب رغبتنا.
	عنوان (تسمية) Label	يستخدم لوضع عنوان label على نموذج يمكن من خلاله توضيح عنوان الكيان يكون بجانبه مثلاً.
	زر أمر Command Button	تستخدم هذه الاداة لوضع مفاتيح على النموذج عند ضغطها سوف يتم تنفيذ اجراء معين
	اطار frame	تستخدم هذه الاداة لوضع اطار ممكن ان يجمع بداخله عدة كيانات وادوات اخرى .
	مربع نص text box	يستخدم لادخال واخراج نص معين ويعتبر افضل اداة للتعامل مع النصوص .
	مفتاح خيار option button	خلال هذه الاداة يمكن وضع خيار واحد على النموذج واكر من خيار وذلك كاسلوب لتنفيذ خيار واحد من مجموعة خيارات .
	مربع تدقيق Check Box	تستخدم لوضع مربع اختيار واحد او اكثر على نموذج والذي يمكن من خلاله تصعيد مربع اختيار واحد اكثر.

<p>يمكننا هذا الكيان من التعامل مع القوائم حيث انه يتم ادخال واسترجاع أي عنصر من والى تلك القائمة .</p>	<p>مربع قائمة List box</p>	
<p>هذه الاداة شبيهه الى حد كبير الاداة التي تسبقها (مربع قائمة) سوى فرق قليل هو ان هذه الاداة لاتظهر بشكل مستطيل وفيه قائمة عناصر ولكن تظهر بشكل مستطيل صغير يحتوي علىعنصر واحد (اخر عنصر تم استخدامه) اما اذا اردنا استخدام عنصر غيره هنالك خيارين هو اما ان نكتب العنصر المرغوب به او نضغط السهم الموجود في اليمين فتظهر قائمة مستدلة للأسفل ونختار العنصر المراد منها.</p>	<p>مربع القائمة المنسدله Combobox</p>	
<p>هذه الاداة تستخدم لوضع شريط انزلاق عمودي لاي كيان نريد وضع الشريط له</p>	<p>شريط انزلاق عمودي Vscrollbar</p>	
<p>نفس الاداة السابقة لكن يكون شكلها افقي .</p>	<p>شريط انزلاق افقي Hscrollbar</p>	
<p>تستخد هذه الاداة لوضع مربع قائمة منسدلة يحتوي على اسماء مشغلات الاقراص حيث انه سوف يأخذ اسماء مشغلات الاقراص المتاحة في الحاسبة</p>	<p>مربع قائمة مشغلات Drive list box</p>	
<p>تستخدم هذه الاداة لوضع مؤقت على النموذج يستخدم لتوقيت العديد من العمليات والاحداث الممكن اجراؤها في المشروع.</p>	<p>المؤقت Timer</p>	
<p>يستخدم لوضع مربع قائمة يحتوي على قائمة باسماء ملفات</p>	<p>مربع قائمة ملف</p>	

بيانيه بامتداد معين وموجوده في مجلد تم اختياره بواسطة الاداة (مربع قائمة مجلدات).	File list box	
يستخدم لوضع مربع قائمة يحتوى على قائمة باسماء المجلدات (الادله) فى مشغل اقراص معين .	مربع قائمة مجلدات Dir list box	
يستخدم لرسم خط على النموذج	خط Line	
يستخدم لرسم شكل هندسي على النموذج .	شكل shape	
تستخدم هذه الاداة للارتباط بقواعد البيانات والتعامل مع بياناتها كعرض البيانات او تحديثها او إضافة بيانات جديدة .	بيانات data	
تستخدم هذه الاداة لوضع صورة معينة على النموذج وهذه الصورة يتم سحبها من ملف في الحاسبة .	صورة Image	
هذه الاداة تستخدم للارتباط بانظمة التطبيقات الاخرى الموجودة في الحاسبة والتي تعمل ضمن بيئة نظام التشغيل Windows .	OLE (Object Linking Embedding)	

٢-٣- نافذة الخصائص :-

تحتوي هذه النافذة على مجموعة من الخصائص التي تحدد مواصفات عنصر التحكم. تحدد كل خاصية ميزة معينة للعنصر كالتطول والعرض ولون الخلفية حيث تعرض هذه النافذة خصائص الكيان المحدد حاليا وعند انتقال التحديد الى عنصر اخر (عند اختيار عنصرا خر ، أي جعله بداخل مقابض وذلك بالضغط عليه بمفتاح الفارة الايسر مرة واحدة)، كما في شكل رقم (٥).

ستظهر تلقائياً خصائص هذا العنصر في القائمة. خصائص كل كيان تختلف عن خصائص الكيانات الأخرى ولو ان بعض الكيانات تتشابه في بعض الخصائص. في نافذة الخصائص يكون على اليسار اسم الخاصية وعلى اليمين اعداد تلك الخاصية حيث إن جميع الخصائص لجميع الكيانات يكون اعدادها تلقائياً وذلك بواسطة معالج اللغة ،ويمكن تغيير اعداد أي خاصية حسب رغبة المبرمج وحسب الخيارات المتاحة لتلك الخاصية ويكون التغيير في مرحلة تصميم المشروع . تحتوى نافذة الخصائص صندوق قائمة يحوى قائمة باسماء الكيانات الموجودة في المشروع (كل أداة تحكم موجودة على النموذج تعتبر كيان) .يمكن اختيار أي عنصر لاستعراض خصائصه مباشرة . إما في أسفل النافذة فيوجد اطار DESCRIPTION أي الوصف، يحتوي على معلومات توضيحية عن وظيفة الخاصية المحددة. يمكن اخفاء او اظهار DESCRIPTION بالنقر الزر الأيمن للفارة على نافذة الخصائص وتحديد او الغاء القائمة DESCRIPTION DESCRIPTION يمكن استعراض الخصائص اما وفق الترتيب الابجدي للخصائص من قائمة ALPHABETIC او وفق فئات الخصائص من القائمة CATEGORIZED ، حيث تقسم الخصائص الى عدة فئات هي :-

1- APPEARANCE (الظهور) :- تحتوى هذه الفئة مجموعة

الخصائص المتعلقة بطرف الاظهار والشكل العام.

2- BEHAVIOR (السلوك) :- تصنف قائمة الخصائص

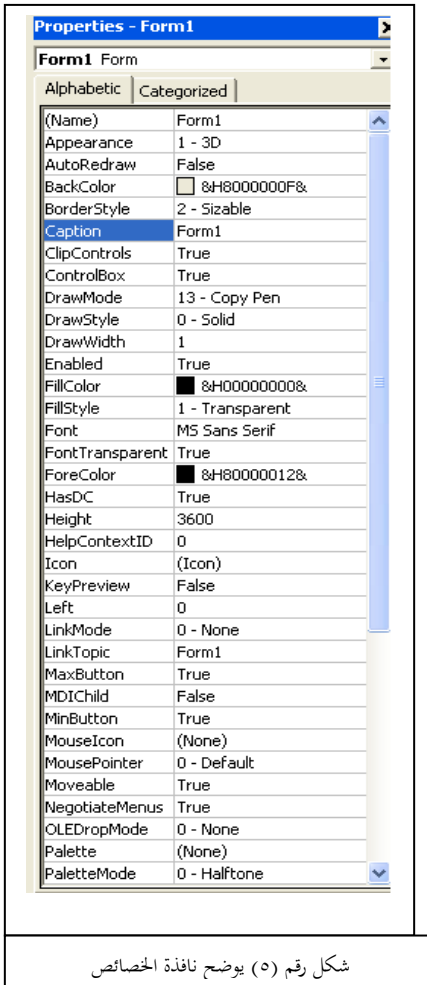
الموجودة ضمن هذه الفئة طريقة عمل الكيان بالنسبة لباقي العناصر في التطبيق .

3- DDE (DYNAMIC DATA EXCHANGE) :-

ويقصد بها التبادل الديناميكي وهو الوسيلة الاتصال بين النماذج في التطبيق، هذه الفئة خاصة بالنماذج فقط، إذ لا تمتلك باقي كيانات التطبيق الخصائص الموجودة في هذه اللغة.

FONT الخط :- تحتوى هذه الفئة على الخصائص المتعلقة بنمط

الخط الذي سيظهر في الكيان مثل النوع والحجم والتأثيرات المتعلقة بالخط.



شكل رقم (٥) يوضح نافذة الخصائص

MISC متفرقات: - ويقصد بها مواصفات مختلفة، إذ تحتوي هذه الفئة على مجموعة من الخصائص ذات تأثيرات متنوعة، اغلب الخصائص الموجودة في هذه الفئة كثيرة الاستخدام أثناء تطوير التطبيقات.

POSITION الموضع: - تقوم الخصائص الواردة في هذه الفئة بتحديد حجم الكائن من ارتفاع وعرض إحداثيات على النموذج.

SCALE المقياس: - تحدد هذه الخصائص مقياس الإحداثيات المستخدمة في النموذج.

الفصل الثالث/بناء مشروع جديد

٣-١- مراحل تصميم المشروع:-

أي مشروع في لغة VB لا يكتمل الا بثلاث مراحل مترابطة مع بعضها البعض وهي:-
أولاً: مرحلة التصميم :- في هذه المرحلة يتم تصميم المشروع بشكل مظهرى أي انه فى هذه المرحلة يتم وضع الادوات المراد وضعها فى النموذج ، وذلك بالاستعانة بصندوق الادوات (TOOL BOX) .

كيفية وضع الأدوات على النموذج :-

نضغط على الادوات المراد وضعها على النموذج بمفتاح الفارة الايسر مرة واحده فقط ننقل مؤشر الفارة الى النموذج فتلاحظ ان مؤشر الفارة قد تغير شكله ، عادة يتغير شكل مؤشر الفارة الى + .

نضغط على مفتاح الفارة الايسر وتسحب الفارة باتجاه قطري (عند السحب تبقى ضاغطين على مفتاح الفارة الايسر) . بحيث يكون شكل الاداة مربع او مستطيل .

ملاحظة: - بعد اضافة أي عنصر الى النموذج يمكن تحريكه وتغيير موقعه وحجمه وذلك بالضغط عليه بمفتاح الفارة الايسر حيث يصبح بداخل مقايض عند وضع المؤشر على احد المقايض سوف يتغير شكل مؤشر الفارة الى سهم ذو رأسين فاذا اردنا تغيير حجم مربع الاداة باتجاه معين نضغط على المقبض الذى بذلك الاتجاه ثم نسحب بذلك الاتجاه ، وهو اما اتجاه افقى او عمودى او اتجاه قطرى .

ثانياً: مرحلة إعداد خصائص كل كيان :- بما ان كل كيان له مجموعة خصائص تم اعدادها بشكل تلقائي من قبل مصممى اللغة، بعض الخصائص نرغب بشكل ضرورى جدا تغييرها الى ما يلائم رغباتنا وحسب الخيارات والمجالات المتاحة ، نلاحظ ان جميع الخصائص لجميع الكيانات المتاحة هي مرتبه بشكل ابجدى (حسب ابجدية الحروف الانكليزيه) ، عدا الخاصيه name حيث ان هذه الخاصيه تحدد اسم الكيان وليس عنوانه والفرق بين الاثنين هو ان اسم الكيان هو الاسم الذى يتم التعامل معه فى الشفرات وهذا الاسم لا يظهر على الكيان فى مرحلتى التصميم والتنفيذ ، اما العنوان فانه مجرد نص (او نص وصفى) يظهر فوق الاداة فى مرحلة التصميم ومرحلة التنفيذ ويمكن تغييره الى أي نص اخر من خلال الخاصية caption

للكيانات label وcommand نلاحظ ان هناك ثلاث اشكال لاعدادات الخصائص القسم الاول منها يمتد الضغط عليه ((على الخاصيه)) فان شكل مؤشر الفارة يتحول الى مؤشر نصوص وهو خط مستقيم عمودي على سبيل المثال خاصية caption وخاصية next وغيرها من الخصائص . القسم الثاني من الخصائص عند الضغط عليه فانه يظهر في يمينه راس سهم متجه الى الاسفل عند الضغط عليه سوف تتسدل قائمة فيها مجموعة خيارات وهذا يعنى ان اعداد هذه الخاصية ينحصر في تلك القائمة أي يجب ان نختار احد خيارات تلك القائمة، على سبيل المثال خاصية الرؤية visible فان فيها اعدادين هما true ، false القسم الثالث من الخصائص عند الضغط عليها سوف يظهر في يمينها مربع صغير يحتوى على ثلاث نقاط (...) وهذا يعنى انه عند النقر على (...) سوف تفتح لنا نافذة خاصية font فان فيها خصائص أخرى يتم تحديدها وهي نوع الخط وهيئة الخط، وكذلك خاصية picture فانه يتم تحديدا امتداد الملف الذي يحوى على صورته هذا ما يخص اعداد الخصائص في مرحلة اعداد الخصائص، هنالك اعداد للخصائص في مرحلة البرمجة (مرحلة كتابة الشفرات)، حيث ان هنالك صيغة عامة لأعداد الخصائص:-

Object. Property=value

فيما يلي جداول يبين بعض الخصائص واعداداتها المتاحة ووصف لكل اعداد. هذه الخصائص هي الخصائص التي يكون اعدادها يحوي على قائمة من الخيارات بغض النظر عن نوع الكيان:-

الوصف	الاعداد	الخاصية
- يظهر هذا الاعداد النموذج بشكل مسطح	0-flat	Appearance
- يظهر النموذج ثلاثي البعد	1-3d	
- بلا حدود تظهر فيها النافذة بدون شريط عنوان أو حدود جانبية.	0-none	Border style

<p>- حد مفرد ثابت تظهر فيها النافذة بحد مفرد ولا يمكن للمستخدم ان يغير من ابعاد النافذة بعد التنفيذ ويظهر في هذا الإعداد شريط العنوان وزري التكبير والتصغير.</p>	<p>1-fixedsingle</p>	
<p>-قابل للتغيير وهي الحالة الافتراضية لاعدادات النماذج يظهر فيها شريط العنوان وأزرار التحكم وصندوق التحكم في شريط العنوان ويمكن للمستخدم ان يقوم بتغيير ابعاد النافذه خلال مرحلة التنفيذ.</p>	<p>2-sizeable</p>	
<p>-صندوق حوار ثابت بسلك النموذج في هذا الاعداد سلوك صناديق الحوار أي لا يمكن تغييرحجم النموذج ولا يحتوي على زري التكبير والتصغير ولا على صندوق التحكم وهذا الخيار مفيد لاجل تصميم صناديق حوار نموذجية.</p>	<p>3-ixeddialog</p>	
<p>الاعلاق فقط بواسطة (x) ويكون هذا النموذج ثابت ولا تظهر ايقونة النموذج في شريط المهام وهذا الخيار مفيد لتصميم نوافذ ادوات التحكم.</p>	<p>4-fixd tool window</p>	
<p>-نافذة ادوات متغيرة : يسلك النموذج في هذا الخيار نفس سلوك الخيار السابق غير انه متغير الحجم.</p>	<p>5-sizedble tool window</p>	
<p>مفعلة :- يتم اظهار ازرار التحكم وصندوق التحكم في شريط عنوان النموذج وهذا الاعداد هو الحالة الافتراضية .</p>	<p>True</p>	<p>Controlbox</p>

معطلة :اخفاء ازرار التحكم وصندوق التحكم في شريط عنوان النموذج وبالتالي عدم قدرة المستخدم على تصغير او تكبير النموذج الى الحد الادنى او الاقصى.	False	
مفعلة : يتم تفعيل زر التكبير وظهوره على شريط العنوان وهذا الاعداد هو الاعداد الافتراضي	True	Maxbutton
معطلة : يبدو زر التكبير على شريط العنوان باهتا ويستطيع المستخدم تصغير النموذج الى حده الادنى .	False	
مفعلة:الحالة الافتراضية حيث يتم تفعيل زر التصغير في شريط العنوان.	True	Minbutton
معطلة: يظهر زر التصغير باهتاً في شريط العنوان ويستطيع المستخدم تكبيرالنموذج الى حده الادنى.	False	
مفعلة: عند هذا الاعداد يكون اسم النموذج في الجهة اليمنى من شريط العنوان وكذلك القوائم المستدلة يكون اتجاهها من اليمين الى اليسار والطباعة النموذجية تكون من اليمين الى اليسار.	True	RightToLeft
معطلة: عكس العملية السابقة،وهذا الاعداد هو الاعداد الافتراضي .	False	
مفعلة: يكون الكائن عند هذا الاعداد مرئياً وهذا هو الاعداد الافتراضي.	True	Visible

معطلة: يكون الكائن غير مرئي عند التنفيذ.	False	
يدوي يتم فيها تحديد كيفية ظهور النموذج في نافذة توضح النماذج (يدوياً) (from layout window) او من الخاصتان (top و left).	0-manual	Startupposition
مركز البرنامج الذي ينتمي النموذج اليه عند هذه القيمة يظهر النموذج في منتصف المساحة المحجوزة على الشاشة النافذة الرئيسية للبرنامج.	1-centrowner	
مركز الشاشة : يتم وضع النموذج في منتصف الشاشة بغض النظر عن المساحة المحجوزة للبرنامج.	2-centerscreen	
نافذة افتراضية : يظهر فيها النموذج بالاعدادات الافتراضية من قبل برنامج VB6	3-windows default	
حالة النافذة الطبيعية : وهى الحالة الافتراضية . تظهر فيها نافذة النموذج بشكلها الطبيعي موضوعة على الشاشة حسب الابعاد المحددة للنموذج .	• -Normal	Windows sates
حالة الحد الأدنى : تظهر النافذة بعد تشغيل البرنامج فى حالة (التكبير) لتملا كل الشاشة .	1-Minimized	
محاذاة الى اليسار يظهر النص في يسار صندوق النصوص .	0-Left justify	Alignment

توسيط:إلى اليمين ، يظهر النص في يمين صندوق النصوص .	1-Right justify	
توسيط : يظهر النص فى وسط صندوق النصوص .	2-Center	
مفعلة : وهى قيمة افتراضية ، حيث تشارك الأداة في البرنامج بخصائصها والاجراءت	True	Enabled
معطلة : تكون الأداة غير فعالة في البرنامج ويتم إيقاف تأثير كافة خصائصها الأخرى في عملية البرمجة.	False	
قياسي : هذه القيمة هي الحالية الافتراضية حيث تظهر خانة الاختيار على شكل تحقق.	0-standard	Style check
رسومي : تجعل هذه القيمة خانة الاختيار على شكل زر امر .	1- graphical	
تظهر خانة الاختيار فارغة من إشارة التحقق.	0-unchecked	Value
تكون خانة الخيار محتوية على إشارة التحقق بلون غامق	1- checked	
تحتوي خانة الاختيار على إشارة التحقق بلون باهت.	2- grayed	
بدون انحياز وهي الحالة الافتراضية.	0-none	Align
انحياز للأعلى: يظهر صندوق الصور في أعلى النموذج وتستخدم هذه القيمة عند الحاجة لإنشاء شريط أدوات بالاستعانة بأداة صندوق الصور.	1-align top	

انحياز للأسفل: يظهر صندوق الصور في أسفل النموذج وتستخدم هذه القيمة عند الحاجة لإنشاء شريط حالة	2-align bottom	
انحياز لليساار: يظهر في أقصى يسار النموذج.	3-align left	
انحياز لليمين: يظهر في أقصى يمين النموذج	4-align right	
خلفية شفافة: بحيث لا يظهر عند هذا الإعداد لون الخلفية لهذه الأداة.	0-transparent	Back style
خلفية كامده: يظهر عند هذه القيمة لون الخلفية أو الأرضية للأداة.	1-opaque	
شفاف: وبالتالي تظهر الأداة بشكل بدون إطار	0-transparent	Border style
خالص: يظهر الإطار في الأداة متصل بدون انقطاع	1-solid	
شرطات	2-dash	
نقاط	3-dot	
شرطة ونقطة	4-dash-dot	
شرطة ونقطتين	5-dash-dot-dot	
خالص داخلي: تكون عند هذه القيمة ثخانة الإطار من الداخلة	6-inside solid	
تعبئة خالصة: عند هذه القيمة تمتلئ الأداة بلون التعبئة fill color.	0-solid	Fill style

تعبئة شفافة: عند هذه القيمة يظهر فقط لون الإطار للأداة	1-transparent	
تعبئة بخطوط أفقية: تكون التعبئة عبارة عن خطوط أفقية يحدد لونها بلون التعبئة	2-horizontal line	
تعبئة بخطوط عمودية	3-verticalline	
تعبئة بخطوط مائلة صاعدة	4-upwarddiagonal	
تعبئة بخطوط مائلة صاعدة	5-downward diagonal	
تعبئة بخطوط متقطعة	6-cross	
تعبئة بخطوط متقاطعة مائلة.	7-diagonalcross	
مستطيل	0-rectangle	Shape
مربع	1-square	
بيضوي	2-oval	
دائري	3-circle	
مستطيل مدور الحواف	4-rounded rectangle	
مربع مدور الحواف	5-rounded square	
عدم السحب: أي ان هذا الكيان لايمكن سحبه بمؤشر الفارة	0- Manual	Drag mode
ممكن السحب: أي انه يمكن سحبه	1- Automatic	

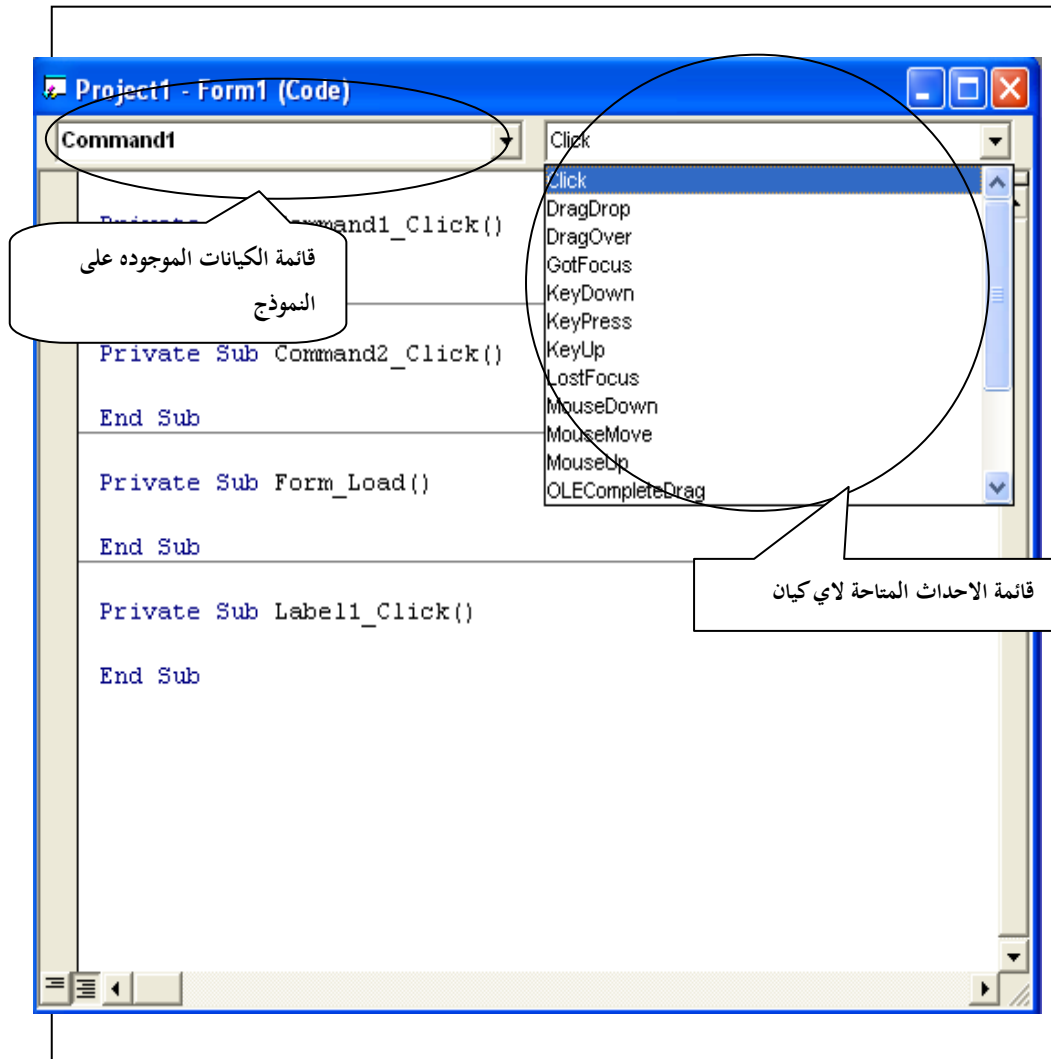
ثالثا //مرحلة البرمجة :-

تشكل هذه المرحلة اهم مرحلة في مراحل بناء المشروع حيث بدون هذه المرحلة فان المشروع لا يمكن ان ينفذ ، في هذه المرحلة يتم بناء وكتابة شفرات المشروع والتي على اساسها يتم تنفيذ او توجيه معالج اللغة بكيفية التنفيذ المشروع .

هذه المرحلة يتم فيها كتابة شفرات المشروع من خلال نافذة CODE WINDOW والتي يمكن الوصول اليها بعدة طرق هي اما بالضغط مرتين بمفتاح الفارة الايسر على الكيان المراد كتابة شفرته و من خلال قائمة

Code Window ← View

كما موضح بالشكل رقم (٦)



شكل رقم (٦) يوضح نافذة الشفرات (نافذة البرنامج)

تحتوي نافذة الشفرات فى اعلاها على صندوقي قائمة الذي على اليسار يحتوى على قائمة باسماء الكيانات التى تم وضعها فى المشروع بضمنها الكيان form وكذلك general أى الجزء الخاص بالشفرات العامة،

حيث يتم اختبار اسم الكيان وكتابة شفرته ، اما صندوق القائمة الذى فى يمين النافذة فانه يحتوى على مجموعة احداث ، والمقصود بالحدث Event هو الاجراء الذى يتم اتخاذه لتنفيذ شفرة الكيان المحدد وعلى الاكثر يكون الحدث هو click أى يتم تنفيذ هذا الاجراء (او شفرة هذا الكيان) وذلك بالضغط على مفتاح الفارة الايسر .بشكل عام فان الجميع ما سوف نأتى على شرحه هو ضمن هذه المرحلة وهى مرحلة البناء البرمجى للمشروع .

عندما نختار أى كيان لكتابة شفرته فان نافذة البرمجه تظهر بالشكل التالي :-

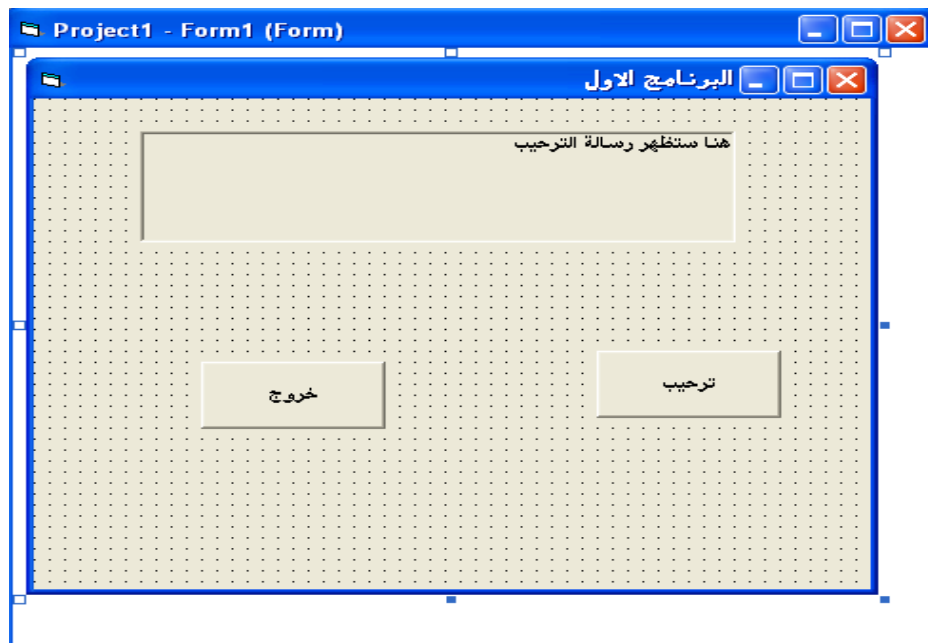
```
Private sub      اسم الكيان _click( )
| ← مؤشر ألكتابه
END SUB
```

هذا التصريح يظهر بشكل تلقائي مع أى كيان .

٣-٢- مثال توضيحي:-

المثال التالي نوضح فيه مراحل تصميم المشروع،في هذا المثال سنقوم بتصميم مشروع فيه مفتاحي امر الاول لعرض جملة الترحيب والثاني للخروج ورسالة الترحيب ستعرض باستخدام كيان تسميه (Label)

اولا:مرحلة التصميم:



ثانيا:مرحلة اعداد الخصائص

الاعداد	الخاصيه	الكيان
البرنامج الاول	Caption	Form1
True	RightToLeft	
ترحيب	Caption	Command1
خروج	Caption	Command2
	Caption	Label1
1-fixd single	BorderStyle	

ثالثا:مرحلة البرمجة

تعد هذه المرحلة من اهم مراحل تصميم المشروع حيث من خلالها يتم كيفية توجيه عمل معالج اللغة عند التنفيذ-سياتي شرح هذه المرحلة بالتفصيل- في هذا المشروع نحتاج فقط كتابة شفرات للكيانيين (command1 و command2) ،عندما نريد كتابة شفرات كيان معين ننقر على ذلك الكيان بالنقر المزدوج فتظهر لنا نافذة كتابة الشفرات والتي تم مناقشتها سلفا، في الامر الذي اسمه "خروج" سنكتب

End

وهذا يعني اننا عند التنفيذ عندما ننقر على هذا المفتاح سوف يتم انتهاء التنفيذ ، اما في الامر الذي اسمه "ترحيب" سنكتب العبارة التاليه

Label1.caption=" مرحبا بكم في لغة Visual Basic"

عند التنفيذ ستظهر هذه العبارة في مربع التسمية Label1

٣-٣-تسمية الكيانات:-

كما تعلم ويجب ان نعلم ان جميع الكيانات في Visual Basic تمتلك خاصيه فريده هي خاصية الاسم NAME وهي اول خاصية في مربع الخصائص PROPERTIES WINDIOW ،هذه الخاصيه (خاصية الاسم) لا تظهر في واجهة النموذج ولكنها تظهر في الشفرات ويمكن التعامل مع الكيان بدلالة هذه الخاصية.اغلب اذ لم يكن جميع مبرمجوا Visual Basic متفقون على استعمال ثلاثة احرف بادئه في اسماء الكيانات ، لان ذلك يساعد على

معرفة اية اداة تحكم انشاءت الكيان . يمكن استعمال اسلوب التسميه هذا للكيانات التي تنشئها لتجعل شفرة البرنامج اكثر وضوحا ولتسهل فهمها على الآخرين من مطوري VB .
بالاضافة لذلك تؤدي مصطلحات التسمية الى سرد الكيانات بترتيب ابجدي في مربع سرد الكيانات في نافذة الشفرات .
يبين الجدول التالي مصطلحات تسمية الكيانات ويتضمن اسم الكيان بالعربية والانكليزية ومصطلح التسمية (الاحرف البادئة) ومثال عن كيفية استعمال كل كيان :-

مثال	الأحرف البادئة	اسم الكيان بالإنكليزية	اسم الكيان بالعربية
CboEnglish	cbo	ComboBox	مربع كتابة وسرد
ChkReadOnly	chk	CheckBox	مربع قائمة تدقيق
CmdCancel	cmd	CommandButton	مفتاح (زر) امر
DatBiblio	dat	Data	بيانات
DirSource	dir	DirectoryBox	مربع سرد الدلائل
DrvTargst	drv	DriveHistBox	مربع سرد محركات الاقراص
FilSource	fil	FileListBox	مربع سرد الملفات
FraLanguage	fra	Frame	إطار
FrmPrintForm	frm	Form	نموذج
HsbVolume	hsb	HorizontalScrollBar	شريط انزلاق افقي
ImgEmptyBarrel	img	Image	صوره
LblInstructions	lbl	Label	تسمية (عنوان)
LinUnderLine	lin	Line	خط
LstPeripherals	lst	ListBox	مربع سرد قوائم

MnuFileOpen	mnv	Menu	قائمة
OleObject	ole	OLE	کیان ارتباط بتطبيقات أخرى
OptFrench	opt	OptionButton	زر خيار
PicSmokeCloud	pic	PictureBox	مربع رسم
ShpWireScreen	shp	Shape	شكل
TxtGetName	txt	TextBox	مربع نص
TmrAlarm	tmr	Timer	عداد (موقت)
VsbTemperature	vsb	VerticalScrollbar	شريط انزلاق عمودي

الفصل الرابع/المتغيرات والبيانات

٤-١- الثوابت والمتغيرات :-

ان اول شي يجب الخوض فيه-ضمن المجال البرمجي-هو مايسمى بالمتغيرات ، حيث ان اغلب لغات البرمجة- اذا لم يكن جميعها-تمتلك القابلية على خزن المعلومات بشكل وقتي ، هذه المعلومات هي البيانات التي نحتاجها عند تنفيذ برامجنا فى أي تطبيق لغة VB تظهر متغيرات او ثوابت كتقنية لحزن تلك المعلومات .

إن المتغير هو ببساطة خزين ، او موقع محدد فى الذاكرة حيث نستطيع الوصول الى ذلك الموقع التخزيني وتخزن او تسترجع المعلومات فى أي وقت و كيفما نريد .

٤-٢- تعريف الثوابت والمتغيرات :-

بشكل عام هنالك نوعين من تصريح الثوابت والمتغيرات الاول يكون عام أي اننا نقوم بتصريح الثوابت والمتغيرات بشكل عام أي فى الجزء GENERAL (قسم التصريحات العام) وخارج شفرة أي كيان ، وتكون صيغة هذا التصريح هي باستخدام العبارة OPTION EXPLICIT ثم فى السطر التالى او الاسطر التالية نصرح بالثوابت والمتغيرات، هذا النوع يفيدنا عندما يكون لدينا ثوابت ومتغيرات نحتاجها فى عدد كبير من الكيانات. اما النوع الثانى فيدعى بالتصريح الضمنى أي نقوم بتصريح متغيرات كل كيان بداخل شفرته (برنامج) فقط .

٤-٣- أشكال تصريح البيانات:

١-الشكل الاول يكون باستخدام العبارة DIM وهى اختصار لـ DEMENSION وتعنى بعد

فى الذاكرة (حجم) ويستخدم هذا الشكل فى التصريح العام والتصريح الضمنى وهذا الشكل هو

احدى الحالتين :-

DIM اسم المتغير

النوع البياني AS اسم المتغير DIM

٢- الشكل الثانى هو باستخدام العبارة BUPLIC او GLOBAL وهاتين العبارتين

تستخدمان فى

التصريح العام للبيانات :

اسم المتغير

BUPUBLIC

BUPUBLIC النوع البياني AS اسم المتغير

GLOBAL النوع البياني AS اسم المتغير

اسم المتغير

GLOBAL

٣- الشكل الثالث هو باستخدام العبارات PRIVATE او LOCAL وهاتين العبارتين تستخدمان

في التصريح الضمني للبيانات وشكلهما هو :-

اسم المتغير

PRIVATE

PRIVATE النوع البياني AS اسم المتغير

اسم المتغير

LOCAL

LOCAL النوع البياني AS اسم المتغير

هذه أشكال التصريحات أعلاه هي لأنواع البيانات القابلة للتحويل ونعني بها إن المتغير يكون ذات قيمة بدائية (ابتدائية/أولية) غير ثابتة أي انه عند تنفيذ البرنامج يأخذ المتغير قيمة معينة ويتم عليها الأجراء الملائم وعند الانتهاء من البرنامج وتنفيذه مرة أخرى فانه سوف يأخذ قيمة ابتدائية جديدة كأن تكون ٠ أو ١ أو أي قيمة أخرى يتم إعطائها له في البرنامج .

في حين إن هنالك صيغة أخرى من التصريحات وهي صيغة التصريحات المستقرة STATIC وهي يمكن ان تستخدم في كلا نوعي التصريحات (العام والضمني)، وهذه الصيغة هي :-

اسم المتغير STATIC

النوع البياني AS اسم المتغير STATIC

في هذه الصيغة من التصريحات (أي STATIC) يكون المتغير ثابت أي انه يحتفظ بقيمة آخر تنفيذ سابق له كقيمة أولية إذا أصبحت قيمته مثلاً هي (١) وتم تنفيذ البرنامج مرة أخرى فان قيمة المتغير الابتدائي في هذا التنفيذ سوف تكون (1) علماً ان قيمته الابتدائية في التنفيذ الأول للبرنامج هي ليست هذه القيمة ، قد تكون مثلاً (٠) او (٢) او أي قيمة أخرى غير (١) .

التصريح بالثوابت:-

الثوابت:- هي المتغيرات التي لها قيمة ثابتة قبل دخولها للبرنامج واستخدامها في عباراته وهذه القيمة تعطى للمتغير عند التصريح به، أما عن شكل التصريح به هو

قيمة = النوع البياني AS اسم الثابت DIM CONST

(القيمة هنا هي حسب النوع البياني)

٤-٤-٤- أنواع البيانات:-

فيما يلي جدول يبين أنواع البيانات المعرفة في لغة Visual Basic وحجم الذاكرة المتاح لكل متغير يأخذ ذلك النوع ويبين الجدول كذلك مدى العناصر التي يمكن التعامل معها ضمن هذا النوع :-

نوع البيانات	الحجم	المدى	مثال عن الاستعمال
INTEGER (عدد صحيح)	2BYTE	من ٣٢٧٦٨ - إلى ٣٢٧٦٧	DIM BIRD % BIRD %= 37 او Dim bird as integer BIRD = 37

DIM LON& LON & = 30000 او DIM LON AS LONG LON =30000	من ٢١٤٧٤٨٣٦٤٨ - ٧ إلى ٢١٤٧٤٨٣٦٤	٤ BYTE	LONG (عدد صحيح طويل)
DIM PRICE PRICE ! = 899.9 او DIM PRICE AS SINGLE PRICE = 899.9	من 3.402823e38 -الى 3.402823e38	8 BYTE	SINGLE (نقطة عائمة فردية الدقة)
DIM PI# PI# = 3-141592 او DIM PI AS DOUBLE DIM PI =3.141592	- من 1.797693134862D308 الى 1.797693134862D308	٨ BYTE	DOUBLE) نقطة عائمة مزدوجة الدقة)
DIM DINAR @ DINAR @ =15 او DIM DINAR AS CURRENCY DINAR =15	- من 922337203685477.5808 الى 922337203685477.5808	8BYTE	CURRENCY (عمله)
DIM ST \$ ST\$ ="DOLAR" او DIM ST AS STRING ST = " DOLAR"	من 0 الى 65535 حرفا	بايت لكل حرف	STRING (سلسلة رمزيه)
DIM FLAG AS		2BYTE	BOOLEAN

BOOLEAN FLAG = TRUE			(منطقي)
DIM D AS DATE D = # 3-1-63 #		8BYTE	DATE تاريخ
DIM TOTAL TOTAL = 29.31	كل مجالات انواع البيانات	16BYTE للارقام ، 22BYTE 1BYTE+ لكل حرف (للسلاسل)	variant متنوع

٤-٥- ادخال واخراج البيانات:- من الضروري جدا فهم عمليات ادخال واخراج البيانات من والى معالج اللغة ،ففى جميع لغات البرمجة يوجد جمل ادخال البيانات واخراج النتائج . هنا فى لغة Visual Basic هنالك عدة طرق لادخال واخراج البيانات وكما يلى :-

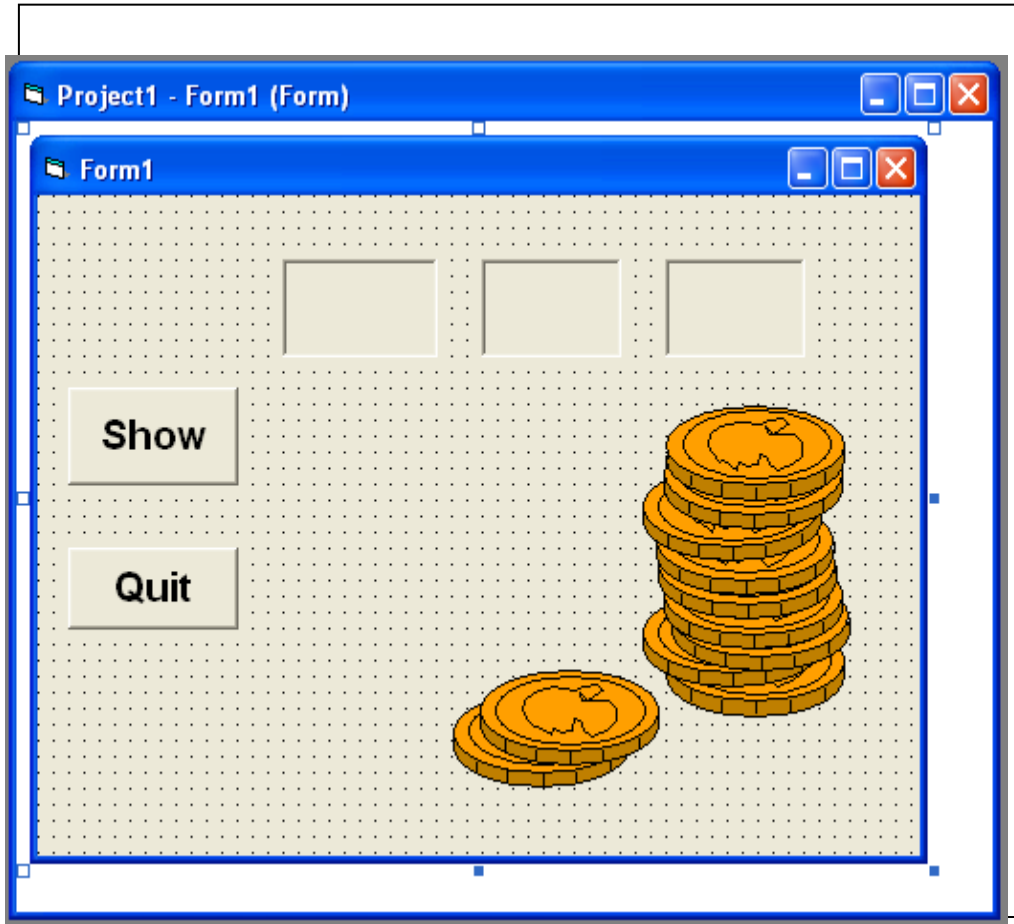
يمكن ادخال البيانات الى معالج اللغة اما عند تنفيذ المشروع او عند كتابة شفرة البرنامج ، فعند كتابة يتم باعطاء المتغيرات المستخدمة فى الشفرة قيم ثابتة وهذه القيم لا تتغير الا اذا قمنا بتغييرها فى الشفرة ،

إما عن إخراج هذه القيم فيتم باستخدام الخاصية CAPTION او TEXT التابعة لكيان الاخراج،

٤-٦- مثال توصيحي ١:-

فى المثال التالى يوضح كيفية التعامل مع بيانات تم ادخالها فى شفرة البرنامج وكيفية إظهار النتائج.

مرحلة إعداد واجهة النموذج:-



مرحلة إعداد الخصائص:-

مرحلة إعداد خصائص الكيانات الموجودة على النموذج:-

SHOW	CAPTION	COMMAND 1
BOLD , ARIAL وبحجم ١٦	FONT	
QUIT	CAPTION	COMMAND 2
BOLD , ARIAL وبحجم ١٦	FONT	
	CAPTION	LABEL1 LABEL2 LABEL3
١ FIXED SINGLE	BORDER STYLE	
٢ -CENTER	ALIGNMENT	

BOLD , TIMES NEW ROMAN بحجم ١٤	FONT	
"C:\program files\microsoft visual studio\common\graphics\metafile\ business\coins.wmf"	PICTURE	IMAGE1
TRUE	STRETCH	
FALSE	VISIBLE	

مرحلة كتابة شفرة البرنامج:- إن المشروع لا يمكن تنفيذه ما لم نكتب له شفرة معينه من خلالها يستطيع معالج اللغة معاملة كيانات المشروع. لكتابة شفرة أي كيان نضغط عليه بمفتاح الفارة الأيسر مرتين بشكل متتالي . نضغط على مفتاح الأمر COMMAND2 والذي أسميناه بـ QUIT: ونكتب في شفرته الجملة END وهذه تعنى بالنسبة لمعالج اللغة بإنهاء التنفيذ. ثم نضغط على مفتاح الأمر COMMAND1 والذي أسميناه بـ SHOW ونكتب فيه الشفرة التالية:-

```
Label1.caption=int(rnd*10)
Label2.caption=int(rnd*10)
Label3.caption=int(rnd*10)
If (label1.cption=5) or (label2.caption=5) or (label3.caption=5) then
  Image1.visible=true
Endif
```

نلاحظ ان في هذه الشفرة تم ادخال بيانات إلى البرنامج من خلال خاصية CAPTION التابعة للكيان نسميه LABEL1 وفي شفرة البرنامج، حيث ان هذه الأرقام هي إعدادات صحيحة وعشوائية. بالنسبة للشفرة اعلاه،

في السطر الأول نقوم بإعداد الخاصية VISIBLE (خاصية الاظهاروالاخفاء بالأعداد FALSE أي اخفاء الصورة الموجودة في الكيان IMAGE1 ، في الأسطر الثاني والثالث والرابع نقوم بإسناد قيمة عشوائية نستنتجها من الدالة RND وهى اختصار لـ)

RANDOMIZE) أي عشوائية ، هذه القيمة العشوائية نأخذ فقط القيم الصحيحة
 INTEGER بواسطة الدالة INT التي تستخلص القيم الصحيحة فقط هذه القيمة الصحيحة
 العشوائية نقوم بإسنادها ((بواسطة عامل الإسناد =)) إلى الخاصية CAPTION التابعة إلى
 الكيانات Label1 , Label2 , Label3 حيث انه في كل مرة نضغط على امر الكيان
 SHOW (في التنفيذ) فانه يتم وضع ثلاث قيم صحيحة عشوائية جديدة إلى الكيانات أعلاه
 . في السطر الرابع من الشفرة السابقه تأتي عبارة IF هنا قمنا بمقارنة ما موجود في
 خاصية CAPTION للكيانات Label1 مع الرقم (٥) باستخدام المعامل المنطقي OR
 حيث انه باستخدام هذا العامل فانه يكفي على الأقل قيمة واحدة مساوية لـ (٥) فانه سوف يقوم
 بتنفيذ العبارة التي تلي كلمة THEN وهي عملية جعل قابلية رؤية الصورة في كيان الصور
 IMAGE1 هو ممكن رؤيتها .

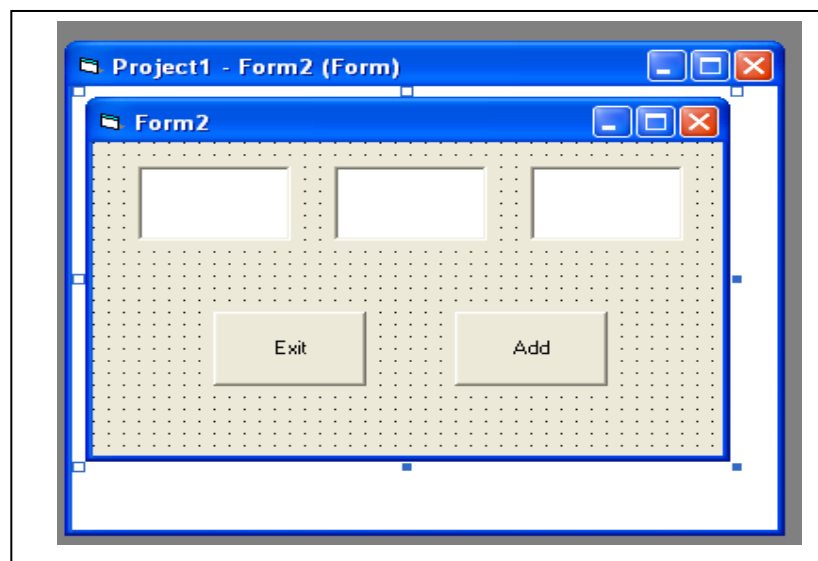
٤-٧-مثال توضيحي ٢:-

المثال اعلاه وضح لنا كيفية ان قيم البيانات تم ادخالها خلال شفرة البرنامج كذلك يمكن ادخال
 البيانات في تنفيذ البرامج بواسطة مربع النص (Textbox.) او دالة الادخال (inputbox) ،

المثال التالي يوضح كيفية ادخال البيانات وإخراج النتائج من خلال مربع نص.

المثال يوضح كيفية اجراء عملية حسابية معينة بين عددين ووضع النتيجة في مربع نص ثالث

-:



إعداد خصائص الكيانات: -

الاعداد	الخاصية	الكيان
ADD	CAPTION	COMAND1
EXIT	CAPTION	COMAND2
	TEXT	TEXT ¹ و TEXT ² و TEXT ³

كتابة شفرة الكيانات:-

نكتب شفرة الأمر COMAND2 والذي اسمه EXIT : نكتب في شفرته الجملة END ثم
نكتب شفرة الأمر COMAND1 والذي اسمه ADD والشفرة هي :-

```
Dim a as integer ,b as integer
```

```
A=val(text1.text)
```

```
B=val(text2.text)
```

```
Text3.text=a+b
```

في السطر الاول تم تعريف (تصريح) متغيرين هما A , B على انهما عددين صحيحين
INTEGER ذات النقطة الثابتة ويمكن تعريفهما على انهما عددين حقيقيين ذات نقطة عائمة
بالتصريح SINGLE .

في السطرين الثاني والثالث في الشفرة تم استخدام الدالة val وهى value ان هذه الدالة
تقوم بمعاملة ما موجود في مربع النص على انه قيمة عدد numerical value ، أي يتم
تحويل ما موجود في مربع نص من قيمة النصية textual value الى قيمته العددية
NUMERICAL VALAU ويتم إسناد هذه القيم الى متغيرين A و B . في السطر
الأخير تتم عملية الجمع بين العددين A و B نضع النتائج في مربع النص TEXT 3 ، وهذه
أيضا تعتبر طريقة إخراج البيانات .

تنفيذ المشروع :- عند تنفيذ المشروع نضع مؤشر الفارة في مربع النص الاول وندخل رقم معين ثم نضع مؤشر الفارة في مربع النص الثاني وندخل رقم معين ثم نضغط مفتاح الأمر ADD فيتم إخراج الناتج في مربع النص الثالث.

كذلك يمكن استخدام دالة الإدخال INPUT BOX لإدخال البيانات حيث إنها تسهل العمل بالنسبة لواجهة المشروع أيضا وذلك بحذف مربع النص TEXT3 ومربع النص TEXT2 إما بالنسبة لشفرة الكيان مفتاح الأمر COMMAND1 والذي اسمه add نقوم بتغيير السطرين الثاني والثالث الى الشفرة التالية:-

```
A=INPUTBOX (" INTER FIRST NUNBER ")  
B = IN PUT BOX ( " INTER SECOND NUMBER "
```

٤-٨- دالة الإدخال **INPUT BOX** : - وهى عبارة عن مربع حوار يتم من خلاله التحوار مع المستخدم user لإدخال بياناته إما صيغة هذه التالية هي :-

```
InputBox(Prompt,[Title],[Default],[Xpos],[Ypos],[HelpFile],[context])  
as string
```

حيث ان :-

١- PROMPT (إجباري) :- (معناه توجيه) وهى الرسالة المطلوب عرضها ويجب ان لا تزيد على (١٢٤) رمز وتوضع بين علامتي اقتباس " " ويمكن وضعها ضمن متغير حرفي وتعويض هذا المتغير مكان الجزء PROMPT فى الدالة اعلاه .

٢- (TITLE) (اختياري) :- عنوان صندوق الحوار ويجب وضعه بين علامتي اقتباس " " .

٣- (DEFAULT) (اختياري) :- القيمة التى تحتل المفتاح المراد ظهوره وهى على الاكثر مفتاحي OK او CANCEL تظهر بشكل تلقائى .

٤- (XPOS) (اختياري) :- وهو عبارة عن رقم يمثل الاحداثي السيني لموقع مربع الحوار لهذه الدالة على الشاشة (ويمثل احداثي النقطة الاولى من اعلى اليسار من مربع الحوار) .

- ٥- (YPOS) (اختياري) :- يمثل احداثي الصادي لموقع مربع الحوار .
- ٦- (HELP FILE) (اختياري) :- اسم الملف المساعدة للبرنامج ويوضع في نفس مجلد البرنامج .
- ٧- (CONTEXT) (اختياري) :- ومعناه سياق ، ويمثل سياق رقم الصفحة التي تحتوي على معلومات عن صندوق الحوار الحالي في ملف المساعدة .
- في مربع الحوار الخاص بهذه الدالة يظهر مربع (او مستطيل) يشبه مربع النص textbox موجود في هذا المربع مؤشر ويشبه مؤشر الكتابة في نظام word فنقوم بإدخال البيانات ثم نضغط المفتاح enter في لوحة المفاتيح ، ان البيانات التي نقوم بإدخالها هي حسب النوع البياني للمتغير الذي تسند إليه هذه الدالة.
- إما شكل الدالة هو :

العنوان TITLE	
التوجيه Prompt	cancel
مربع ادخال البيانات	ok

٤-٩- إخراج البيانات :

هنالك عدة طرق لإخراج البيانات منها باستخدام مربع تسمية label او باستخدام مربع نص textbox او باستخدام الدالة msgbox وهي باختصار لـ message box أي مربع الرسائل ومن خلاله يتم عرض رسائل خطأ او عرض رسائل توجيه وإرشاد.

ملاحظة :

نقصد هنا بإخراج البيانات (المعلومات) وهي تلك المعلومات والنتائج التي تكون نوعها البياني هو عددي او رمزي والتي تعرف بمتغيرات . إما بالنسبة لعرض الكيانات او إخفائها (حيث إنها

تعتبر مخرجات برامج معينة (فذلك يتم من خلال التحكم بتصيب خاصية الرؤية visible لذلك الكيان الذي نريد رؤيته او عدم رؤيته :-

(لإظهار الكيان) Visible =true . اسم الكيان

(لإخفاء الكيان) Visible =false . اسم الكيان

٤-١٠-١ دالة الاخراج MsgBox :

وهي مختصر (مربع الرسائل Message Box)

تعرض الداله MsgBox الرسائل في صناديق حوار، وصيغة هذه الدالة :-

Msgbox(Prompt,[Buttons],[Title],[helpfile],[context])

حيث ان :-

١- Prompt (اجباري) : الرسالة المطلوب عرضها ،ويجب الايزيد عدد احرف هذه الرساله على ٢٥٦ رمز، وتوضع بين علامتي اقتباس " " ويمكن وضعها ضمن متغير حرفي وتعويض هذا المتغير مكان الجزء prompt .

٢- [Buttons] :هذا الجزء مسؤول عن الازرار التي ستعرض في صندوق الحوار ،كذلك تحديد الايقونات المراد عرضها في مربع الحوار ،ايضا ممكن اختيار شكل ونمط مربع الحوار،حيث لاختيار نوع الازرار المراد عرضها والايقونات وشكل الصندوق معا نضع علامة الجمع + بينها ، وممكن استخدام قيم او ثوابت حرفية مكافئة او كلاهما، كما سيأتي في الجدول التالي.

٣- [Title] :عنوان صندوق الحوار ويجب وضعه بين علامتي اقتباس " " .

٤-[HelpFile] :اسم ملف المساعدة-ان وجد-للبرنامج ويوضع في نفس مجلد البرنامج.

٥-[Context] :امتداد الصفحة التي تحتوي على معلومات عن صندوق الحوار الحالي في ملف المساعدة.

الجدول التالي يبين قيم الازرار (Buttons) والايقونات والانماط والثوابت الحرفيه المكافئه لها وهي نفسها يمكن استخدامها في الجزء Buttons في الدالة MsgBox .

الوصف	الثوابت الحرفيه المكافئة	القيمة
يعرض زر Ok	Vbokonly	0
يعرض زري Ok و Cancel	Vbokcancel	1

يعرض الازرار (إجهاض،اعادة المحاولة،تجاهل) Abort ,Retry ,Ignore	Vbabortretryignore	2
يعرض الازرار (نعم،لا،الغاء الامر) Yes ,No ,Cancel	Vbyesnocancel	3
يعرض زري نعم ولا Yes ,No	Vbyesno	4
يعرض زري اعادة المحاولة والغاء الامر Retry ,Cancel	Vbretrycancel	5
يعرض ايقونة الرسالة الحرجة	Vbcritical	16
يعرض ايقونة علامة السؤال	Vbquestion	32
يعرض ايقونة علامة التعجب	Vbexclamation	48
يعرض ايقونة علامة المعلومات	Vbinformation	64
يجعل الزر الاول في صندوق الحوار افتراضيا	Vbdefaultbutton1	٠
يجعل الزر الثاني في الصندوق افتراضيا	Vbdefaultbutton2	٢٥٦
يجعل الزر الثالث في الصندوق افتراضيا	vbdefaultbutton3	٥١٢
يجعل الزر الرابع في الصندوق افتراضيا	Vbdefaultbutton4	٧٦٨
يجعل نمط الصندوق تطبيقي	Vbapplicationmode	٠
يجعل نمط الصندوق نظامي	Vbssystemmode	٤٠٩٦
يظهر زر المساعدة في الصندوق	vbmsgboxhelpbutton	١٦٣٨٤
يجعل الصندوق فوق كافة النوافذ	Vbmsgboxsetforegro und	٦٥٥٣٦
يجعل عنوان الصندوق في يمين شريط العنوان	Vbmsgboxright	٥٢٤٢٨٨
يجعل اتجاه الصندوق من اليمين لليسا ر بحيث تظهر ازرار الاوامر والايقونات متوافقه مع صناديق الحوار العربي	vbmsgboxrtlreading	١٠٤٨٥٧٦

بالإضافة الى الدالة MSGBOX فان هنالك طريقة أخرى لإخراج البيانات وهي باستخدام عبارة الطباعة PRINT وصيغتها (تعبير معين PRINT) حيث ان العبارة PRINT تطبع المخرجات على سطح النموذج. الجدول التالي يبين القيم التي ترجعها الداله MsgBox عند الضغط على أي زر فيها وهذه القيم تفيدنا في امكانية اتخاذ أي اجراء نريده

الوصف (الزر الذي يتم ضغطه)	القيمة المرجعة
OK	١
CANCEL	٢
ABORT	٣
RETRY	٤
IGNORE	٥
YES	٦
NO	٧

أما شكل الدالة (MSGBOX) هو على شكل مربع حوار وهو بالشكل التالي :-

TITLE العنوان

الأيقونة التي
تظهر

PROMPT التوجيه

BUTTON المفاتيح (الأزرار)

الفصل الخامس/اتخاذ القرار وعبارات التكرار

٥-١- مقدمة :-

ان جميع لغات البرمجة (ألا ما ندر منها) تتعامل مع الشفرات (جمل البرمجة) تعاملاً تسلسلياً أي إنها تقوم بتنفيذ عباراتها البرمجية بشكل تسلسلي sequential ، وهذا الشيء سهل جداً في البرمجة أما إذا كان لدينا عبارتين ونريد عرض احدهما بالاعتماد على قيمة معينة. هنا يجب فقط باستخدام العبارات (الجمل) الشرطية حيث ان ألكره الأساسية لبناء هذه العبارات هو إنها تقوم باختيار قيمة معينة او مجموعة قيم أخرى. وعلى أساس هذا الاختيار نقوم بتنفيذ عبارة معينة او مجموعة عبارات ، او عدم تنفيذ تلك العبارات . الجمل الشرطية المتاحة في لغة VB بعضها شرطية فقط مثل عبارتي IF و SELECT CASE وبعضها شرطية تكرارية مثل عبارات WHILE -DO ، DO-WHILE ، DO- LOOP وبعضها تكرارية فقط (ولو إنها لا تعتبر عبارة شرطية أحياناً) مثال عبارة FOR.

٥-٢- عوامل المقارنة :-

قبل الخوض في العبارات الشرطية يجب ان نعرف ماهية أشكال وأنواع أدوات وعوامل المقارنة التي تستخدم لاختيار قيمتين او متغيرين والجدول التالي يبين عوامل المقارنة التي يمكن استعمالها في التعبيرات الشرطية والتي يتيحها معالج Visual Basic.

عامل المقارنة	معناه	مثال
=	يساوي	a=b
<>	لا يساوي	a<>b
>	اكبر من	a>b
<	اصغر من	a=	اكبر من او يساوي	a>=b

$a \leq b$	اصغر من او يساوي	\leq
------------	------------------	--------

٥-٣-العوامل المنطقية:-

في بعض الأحيان نحتاج لاختيار أكثر من تعبير شرطي واحد في القسمين IF...THEN و ELSEIF . فإذا أردنا شمل أكثر من تعبير شرطي في عبارة شرطية واحدة (وخاصة عبارة IF)، نستخدم احد العوامل المنطقية لربط الشروط الإضافية سوية والجدول التالي يبين العوامل المنطقية المتاحة في لغة VB ومعنى كل أداة :-

الأداة (العامل المنطقي)	معناه
AND	إذا كان التعبيران شرطيان كلاهما صح (أي كلاهما متحقق) تكون نتيجة التعبير العام صح مما يؤدي الى تنفيذ العبارات التي تلي الكلمة THEN
OR	إذا كان احد التعبيرين الشرطيين صح، تكون النتيجة صح
NOT	إذا كان التعبير الشرطي خطأ ، تكون النتيجة صح . وإذا كان التعبير الشرطي صح تكون النتيجة خطأ (أي تقوم بقلب نتيجة التعبير)، وهذه الأداة تستخدم مع تعبير شرطي واحد
XOR	إذا كان كلا التعبيرين نتيجتهما متوافقة (أي إذا كان كلاهما خطأ أو كلاهم صح) تكون نتيجة التعبير الشرطي العام خطأ ، وإذا اختلفت نتيجة التعبيرين الشرطيين تكون نتيجة التعبير العام صح .

٥-٤-عبارة IF الشرطية:-

تعتبر عبارة IF أهم عبارة شرطية فبالرغم من إمكانية استخدام غيرها من الجمل الشرطية والتكرارية الأخرى ولكن لايمكن ان نستغني عنها بشكل نهائي من حيث انه يمكن الاستغناء عن الجمل الأخرى . هنالك عدة أشكال وهيئات عامة هي:-

IF ...THEN...
IF...THEN...ELSE...

IF ...THEN...ELSEIF ...ELSE...

IIF (expression , truepart , falsepart)

يمكن كتابة هذه العبارة على سطر واحد:

IF Condition THEN Statement

حيث ان Statement هي عبارة من عبارات لغة VB او يمكن كتابتها بالشكل :-

IF Condition THEN Statement1: Statement2: Statement3

إما في حالة الكتابة على أكثر من سطر ، يجب إضافة ENFIF :-

IF condition THEN

statement1

statement2

.....

.....

ENDIF

يكون الشرط condition عادة عبارة عن تعبير منطقي (مقارنة مثلاً) يرجع إما صح true

او خطأ False يمكن أيضا ان يكون الشرط أي تعبير رقمي ، سيعتبر معالج VB ان الشرط

محقق TRUE إذا كانت نتيجة التعبير قيمة لا تساوي الصفر وألا فانه سيعتبر الشرط غير

محقق FALSE . سيقوم VB في حال تحقق الشرط بتنفيذ التعليمات الموافقة للشرط.

لاحظ ان الشفرتين التاليتين متكافئتين :-

IF X>Y THEN X=Y: Y=Z

IF X<Y THEN

X=Y

Y=Z

ENDIF

الشكل التالي لعبارة IF هو :-

IFTHEN... ELSE....

في هذه الحالة لدينا شرط معين إذا تحقق نريد ان ننفذ تعليمات معينة وإلا (أي إذا لم يتحقق

الشرط) ننفذ تعليمات أخرى . يمكن ان نكتب العبارة على سطر واحد فقط:-

IF condition THEN statement ELSE statement

IF condition THEN statement1:statement2 ELSE statement 1:statement2

كما يمكن ان نكتب العبارة على عدة اسطر :-

```
IF condition THEN
Statement -if - true
ELSE
statements-if -false
End if
```

الشكل :

IIF (expression, truepart,falsepart)

تقوم هذه الدالة باختيار التعبير expression ، فإذا وجدته TRUE فأنها تقوم بإرجاع الجزء true part إما إذا وجدته FALSE فأنها سوف تقوم بإرجاع الجزء false true في المثال التالي ، يمكن كتابة شفرته في الكيان form (كيان النموذج حيث انه يعتبر إذا كانت قيمة المتغير NUMBER اكبر من ١٠٠٠ فان الدالة IFF ستعيد السلسلة الرمزية LARGE إما إذا كانت قيمة NUMBER اصغر او يساوي ١٠٠٠ فأنها سوف تعيد السلسلة الرمزية SMALL والشفرة هي :-

```
DIM NUMBER AS INTEGER
DIM CHECKL AS STRING
NUMBER = INPUTBOX ('INTER THE NUMBER')
CHECK=IIF(NUMBER>1000,"LARGE","SMALL")
PRINT CHECK
```

الشكل :

IF ... THEN ... ELSEIF ...ELSE...

يستخدم هذا الشكل في حال وجود عدة كتل من البيانات ، كتله واحدة فقط من بينها سوف تنفذ وصيغتها :-

```
IF condition THEN
    [statement block-1]
[else if condition2 then
    statement block-2]]
.....
.....
```

.....
[else
 (statement block-n)]
end if

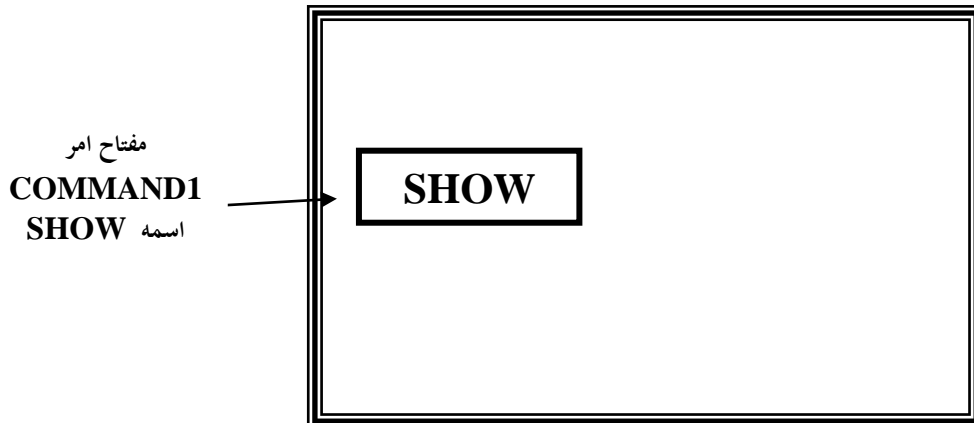
يقوم معالج VB باختيار أول شرط 1 CONDITION فإذا وجده غير محقق (FALSE) فإنه ينتقل لاختيار الشرط الثاني 2 CONDITION ، وهكذا دواليك إذا كُنت هنالك شروط أخرى . حتى يجد شرطاً محققاً ، عندها ستنفذ كتلة العبارات الموافقة لهذا الشرط ثم ينتقل للتعليمات التي تلي end if (أي انه لا يفحص أي شرط من الشروط التالية للشرط الذي تحقق وبالتالي فان كتلة عبارات واحدة على الأكثر فقط ستنفذ).

يمكن ان تحوي التعليمة الشرطية على ELSE ، عندها وفي حال عدم تحقق أي شرط من الشروط الموضوعه ، سيقوم VB بتنفيذ كتلة العبارات الموافقة ل ELSE .

٥-٥-٥-٥-٥ مثال توضيحي:-

المشروع التالي هو مثال جيد ليوضح استخدام الشكل الأخير من أشكال عبارة IF الشرطية وكذلك استخدام الدالتين INPUTBOX و MSGBOX واستخدام احد العوامل المنطقية وهو AND . المشروع عبارة عن ادخال اسم المستخدم وكلمة المرور الخاصة به فإذا كانت AHMAD او ALI الأسماء وكلمات مرور الخاصة به هي SCIENCE و ENGINEERING على التوالي فإنه في حالة AHMAD سوف يعرض صورة معينة (صورة حاسبة) ورسالة معينة إما في حالة ALI فإنه سوف يعرض صورة أخرى ورسالة أخرى إما إذ لم يكن الاسم AHMAD ولا ALI فإنه سوف يعرض رسالة خطأ معينة . المشروع يتكون من نموذج وفيه مفتاح امر واحد :-

واجهة النموذج :-



إعداد خصائص الكيانات

الكيان	الخاصية	الاعداد
COMMAND1	CAPTION	SHOW
	FONT	BOLD,ARIAL بحجم ١٦

كتابة شفرة المشروع

نقوم بكتابة شفرة الكيان COMMAND1 فقط والذي اسمه SHOW :-

```
USERNAME =INPUTBOX ("ENTER YOUR FIRST NAME ")
PASSWORD =INPUTBOX ("ENTER YOUR PASSWORD")
IF (USERNAME ="AHMAD"AND (PASSWORD="SCIENCE"))THEN
MSGBOX ("WELLCOME AHMAD !READY TO START")
FORM1.PICTURE=LOADPICTURE("C:\program files\microsoft
visual studio\common\graphics\metafile\ computer\computer.wmf")
ELSEI (USERNAME ="ALI")AND (PASSWORD=" ENGINEERING")
MSGBOX (WELLCOME ALI !READY TO DISPLAY")
FORM1.PICTURE=LOADPICTURE("C:\program files\microsoft
visual studio\common\graphics\metafile\ office\book1.wmf" ")
ELSE
MSGBOX ("SORRY,IDONT KNOW YOU")
END
ENDIF
```

٥-٦- الجمللة الشرطية SELECT CASE :-

تقدم لنا لغة Visual Basic عبارة SELECT CASE كخيار آخر عوضاً عن العبارة IF...THEN...ELSEIF...ELSE...

من الكتل. تعطي العبارة CASE SELECT إمكانيات مشابهة لعبارة IF...THEN...ELSEIF...ELSE... لانها تجعل التعليمات البرمجية أكثر وضوحاً وأسرع في التنفيذ.

تقوم هذه العبارة بفحص تعبير واحد Test Expression في بداية الجملة ومن ثم مقارنة نتيجة التعبير من مجموعة تعابير (كل تعبير مرتبط مع CASE) وعند تطابق قيمة التعبير TEXTEXPRETION مع احد التعابير Expression List فأنها ستنفذ كتلة العبارات المرتبطة مع CASE المتوافقة .

كما في الصيغة العامة التالية لعبارة SELECT CASE :-

```
SELECT CASE test expression
    [case expression1
      [statementblock-1]]
    [case expression list2
      [statementblock-2]]
    [case else
      [statement block-n]]
END SELECT
```

- تأخذ expression list احد الإشكال التالية:-

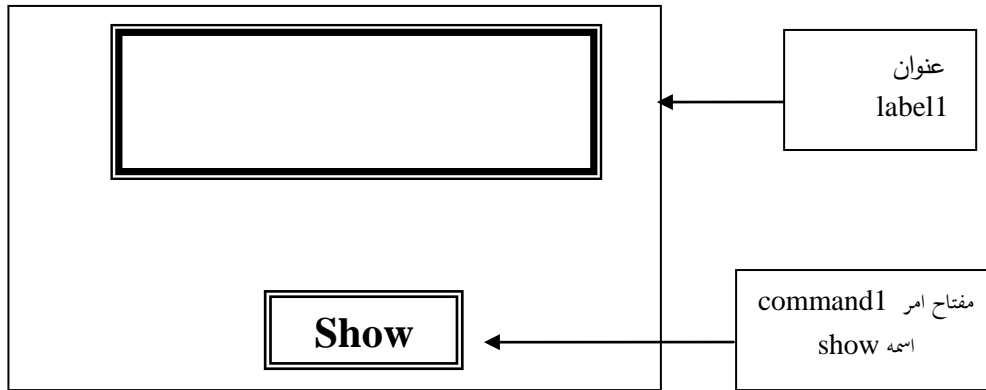
- قيمة واحدة expr
 - عدة قيم expr1,expr2,.....
 - مدى معين (مجال) من القيم expr to expr
 - مقارنة is comp expr < المتغير <
- (حيث comp هو معامل مقارنة: <, >, =, <=, >=, <>)

- كل Statementblock (كتلة عبارات) عبارة عن جملة واحدة او أكثر.

٥-٧- مثال توضيحي:-

في المشروع التالي نلاحظ مثال جيد لاستخدام عبارة select case حيث سوف يتعرف على عمر شخص معين وإذا كان ضمن مدى معين فسوف يخرج رسالة معينة:-

إعداد الواجهة :-



إعداد خصائص الكيانات

الاعداد	الخاصية	الكيان
" SHOW"	Caption	Command1
	Caption	Label1

كتابة شفرات COMMAND1 :-

```
AGE = INPUT BOX (" ENTER YOUR AGE")
SELECT CASE AGE
CASE IS < 13
    LABEL 1 .CAPTION ="YOUR CHILD"
CASE 13 TO 19
    LABEL 1.CAPTION ="YOU SECONDORY STDENT"
CASE 21
    LABEL 1.CAPTION ="YOU IN THE UNIVERSITY"
CASE IS >100
    LABEL 1.CAPTION ="LOOKING GOOD"
CASE ELSE
    LABEL 1.CAPTION ="THAT ANINC AGE TOBE "
END SELECT
```

٥-٨-عبارة التكرار Do :-

عبارات التكرار هي عبارة واحدة وهي عبارة DO...LOOP والتي تقع في أربعة أشكال ، في جميع هذه الأشكال يكون هنالك شرط معين يحدد استمرار التكرار او عدم استمرار التكرار .
كما في عبارة IF فإن الشرط هو إما تعبير منطقي يرجع TRUE او FALSE او تعبير يرجع قيمة رقمية (تعتبر قيمة التعبير التي لا تساوي صفر هي TRUE، وإذا كانت تساوي صفر فأنها FALSE).

والان نأتي لدراسة أشكال هذه العبارة علماً إننا سنتناول مع كل شكل من هذه الأشكال أمثلة وهذه الأمثلة مكتوبة في شفرة كيان النموذج:-

١- في الشكل التالي لـ DO... LOOP فان تكرار تنفيذ كتلة العبارات STATEMENT سيستمر طالما كان الشرط CONDITION غير متحقق:-

DO UNTIL condition

[STATEMENT]

[EXIT DO]

[STATEMENT]

LOOP

عندما يصل معالج اللغة الى هذه العبارة فإنه سيبدأ أولاً باختبار CONDITION فإذا وجده متحققاً TRUE فإنه سوف يقفز فوراً لما بعد سطر LOOP (أي انه سوف لن ينفذ عبارات الحلقة ، إما إذا وجده FALSE فإنه ستنفذ العبارات STATEMENT ثم يعود للسطر DO UNTIL ليختبر ثانياً السطر ويعاد التنفيذ إذا كان الشرط FALSE او يخرج من العبارة إذا كان الشرط TRUE وهكذا....

وبالتالي فان عبارات الحلقة ستنفذ عدداً من المرات طالما بقي الشرط FALSE ولن تنفذ ولا مرة إذا كان الشرط منذ البداية TRUE .

العبارة EXIT DO تؤدي الى الخروج الفوري من عبارة التكرار (أي عند تنفيذ هذه العبارة لا ينفذ معالج لغة VB عبارات الحلقة التي تليها ولا يعاود الدورات بل يقفز مباشرة خارج عبارة الدواره (أي الى الجملة التي تلي كلمة LOOP) . يمكن لهذه العبارة (عبارة EXIT DO) ان تظهر أكثر من مرة في الحلقة .

٥-٩-٩-٥ مثال توضيحي :-

لنفرض إننا نريد حساب مجموع الاعداد من ١ الى ١٠ :-

```
DIM X AS INTEGER ,I AS INTEGER
I=1
X=0
DO UNTIL I>10
    X=X+I
    I=I+1
LOOP
```

الحلقة DO UNTIL السابقة ستنفذ ١٠ مرات ، في كل مرة تزداد قيمة المتغير I ب ١ . في الحلقة الأولى ستزيد X ب ١ ، وفي الحلقة الثانية ستزيد X ب ٢ وهكذا في آخر تنفيذ الدائرة (الحلقة) تكون قيمة I مساوية ل ١٠ فتزيد X ب ١٠ ثم تصبح I مساوية ل ١١ فيتحقق الشرط UNTIL ونخرج من الدائرة.

٢- الشكل الثاني للحلقة التكرارية الشرطية DO ... LOOP ينفذ العبارات في أول دورة ثم يختبر الشرط ، أي ان العبارات ستنفذ على الأقل مرة واحدة ، يتوقف التكرار حين يصبح الشرط متحققاً :-

```
DO [STATEMENTS]
    [EXIT LOOP]
    [STATEMENTS]
LOOP UNTIL condition
```

المثال التالي هو مثال جيد لتوضيح عبارة DO UNTIL وه يقوم بقراءة درجة الحرارة بالفهرنهايت ويحولها الى السيليزي :-

```
PROMPT = " ENTER A FAHRENHEIT TEMPERATURE "
DO

    FTEMP =IN PUT BOX ( prompt, " Fahrenheit to Celsius")
    IF F TEMP <> " " THEN
        Celsius = INT ((F TEMP +40 ) * 5/9 -40)
    MSGBOX (Celsius) ,"TEMPERATURE TO Celsius"
```



```

ENDIF
LOOP UNTIL FTEMP = " "
END

```

٣- الشكل الثالث وهو DO WHILE في هذا الشكل من الدورات الشرطية التكرارية ، تنفذ العبارات فقط إذا كان الشرط متحققاً ويتوقف التكرار عندما يصبح الشرط غير متحقق
 -: FALSE

```

DO WHILE condition
  [STATEMENT(S)]
  [EXIT DO]
  [STATEMENT(S)]
LOOP

```

يمكن كتابة مثال جمع الاعداد من ١ الى ١٠ باستخدام هذا الشكل

```

DIM X AS INTEGER , I AS INTEGER
I=1
X=0
DO WHILE I<= 10
  X=X+1
  I=I+1
LOOP

```

٤- الشكل الرابع والأخير من أشكال الدوارة DO -LOOP هي الدوارة DO-LOOP في هذا الشكل تنفذ العبارات مرة واحدة على الأقل ويتوقف التكرار عندما يصبح الشرط غير متحقق :-

```

DO
  [STATEMENT(S)]
  [EXIT DO]
  [STATEMENT(S)]
LOOP WHILE condition(s)

```

يمكن اخذ مثال درجات الحرارة كمثال جيد لفهم هذا الشكل :-

```

DO
  F=INPUTBOX ("ENTER AFAHRENHEIT TEMPRATURE ")
  IF F <> " " THEN
    a=INT (( F+40)*5/9-40)
    MSGBOX (a),t," TEMPRATURE TO CELSUS "
  ENDIF

```

```
LOOP WHILE F<>" "  
END
```

١٠-٥-١- عبارة التكرار FOR :-

هذه الجملة تستخدم فقط لتكرار تنفيذ مجموعة من العبارات (كتلة من العبارات) لعدد من المرات ، حيث عند الوصول الى الحد الأعلى للتكرار سوف يخرج من الدورة وفيما عداه لا يخرج ، ولا يعتمد الخروج على شرط منطقي او عملية حسابية معينة . لغة VB تمتلك عبارة واحدة فقط هي عبارة تكرار وهي عبارة FOR - NEXT :-

تستخدم عبارة DO السابقة عادة عندما لا نعرف مسبقاً عدد مرات تكرار تنفيذ عبارات الحلقة . إما إذا كنا نعرف مسبقاً عدد مرات تكرار تنفيذ مجموعه من العبارات فإنه من الأفضل ان نستخدم عبارة FOR التكرارية ، حيث انه على خلاف DO فإن عبارة FOR تستخدم متغير يسمى عداد الدورة (الحلقة) يزداد او ينقص في كل دورة من دورات تكرار الحلقة . الصيغة العامة لعبارة FOR :-

```
FOR counter =start (initial value ) TO end [step]  
[STATEMENT(S)]  
[EXIT FOR]  
[STATEMENT(S)]  
NEXT [COUNTER]
```

حيث ان :-

COUNTER هو متغير نوعه INTEGER ولا نحتاج لتعريفه (عداد الحلقة

(

START(initial value) القيمة البدائية لإعداد الحلقة.

END القيمة النهائية لعداد الحلقة.

STEP مقدار الزيادة او النقصان قيم عداد الحلقة بعد كل تنفيذ لعبارات

الحلقة .

جميع هذه الخيارات هي قيم رقمية .

ملاحظه :-

يمكن ان تكون STEP موجبة او سالبه . إذا كَأنت STEP موجبه فَأَن START يجب ان تكون اصغر او تساوي قيمة END كي تنفذ عبارات الحلقة إما إذا كَأنت قيمة STEP سالبه فَأَن START يجب ان تكون اكبر او تساوي قيمة END إذا لم تحدد الخطوة STEP فَأَن معالج اللغه يعتبرها بشكل تلقائي زيادة بمقدار ١ .

يقوم معالج لغة VB بالخطوات التالية لتنفيذ عبارة FOR :-

١- يسند القيمة START الى عداد الحلقة COUNTER .

٢- يفحص فيما إذا كَأنت قيمة العداد اكبر من END في حال كون STEP موجب ، فإذا كَأنت اكبر فأنه سيقفز فيما بعد NEXT (أي لا ينفذ عبارات الحلقة) . (في حالة ان STEP سالبة فأنه يفحص فيما إذا كَأنت قيمة العداد اصغر من END) .

٣- وألا فأنه سينفذ تعليمات الحلقة .

٤- بعدها سيزيد مقدار العداد ب ١ إذا لم تكن STEP محددة وإلا فأنه سيزيده (او ينقصه) بمقدار STEP .

٥- سيعود تكرار الخطوات من ٢ الى ٤ .

مثال عن استخدام عبارة FOR :

مثال جمع الاعداد من ١ الى ١٠ باستخدام FOR

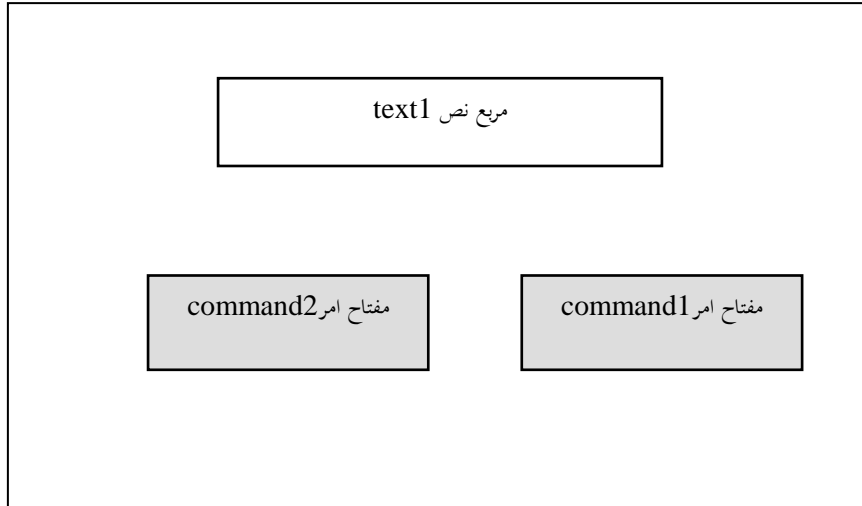
```
DIM X AS INTEGER
FOR I= 1 TO 10
  X=X+I
NEXT I
```

٥-١١- أمثلة توضيحية إضافية :

٥-١١-١- مثال ١

في هذا المثال سوف نستخدم مفتاح امر ومربع نص ودالة ادخال input box سوف نقارن اسم نقوم بإدخاله من خلال دالة الادخال فإذا كان مطابق لاسم معين وليكن مثلاً "ALI" فإنه سوف يطبع العبارة "OK YOUR NAME IS ALI" والاسوف يطبع الاسم فقط :-

إعداد الواجهة



إعداد الخصائص

الاعداد	الخاصية	الكيان
Input	Caption	Command1
Exit	Caption	Command2

كتابة الشفرات :-

شفرة الأمر 1 command هي:

```

DIM NAME AS STRING
Name = INPUTBOX (" INTER YOUR NAME ", "NAMEBOX")
IF NAME ="ALI" THEN
    TEXT1.TEXT="OK,YOUR NAME IS ALI "
ELSE
    TEXT1.TEXT=NAME
ENDIF

```

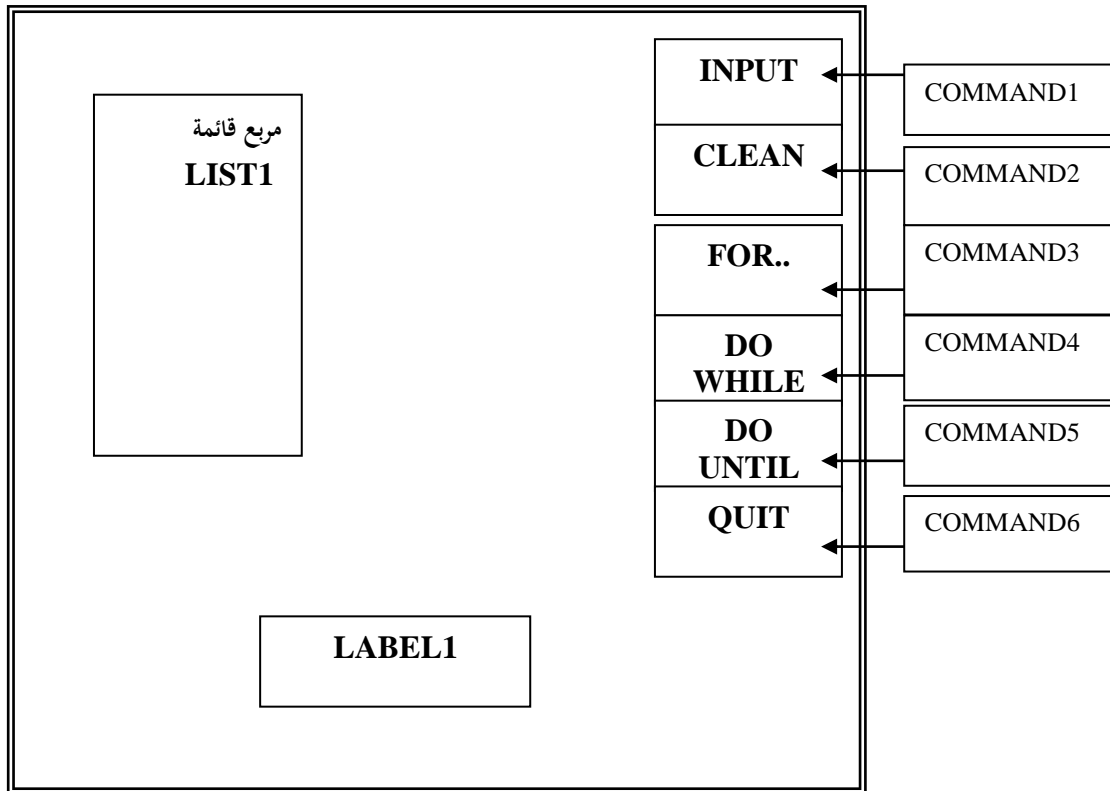
شفرة الأمر 2 COMMAND وهي :-

END

٥-١١-٢-مثال ٢

في هذا المثال سوف نستخدم واجهة لإيجاد جدول ضرب أي عدد وطباعة النتائج في مربع قائمة
وبستخدام عدة أساليب (او عبارات) وهي عبارة FOR وعبارة DO UNIT وعبارة DO
WHILE سوف نستخدم عدة مفاتيح ونستخدم مربع قائمة واحد ونستخدم مربع تسمية
LABEL وكما يلي :-

إعداد واجهة :-



إعداد خصائص كيانات الواجهة:-

الاعداد	الخاصية	الكيان
"INPUT "	CAPTION	COMMAND1
"CLEAN"	CAPTION	COMMAND2
"FOR... "	CAPTION	COMMAND3
"DO WHILE.."	CAPTION	COMMAND4
"DO UNTIL"	CAPTION	COMMAND5
"QUIT"	CAPTION	COMMAND6
"X-MULTI"	CAPTION	LABEL1

كتابة شفرة المشروع :-

أولاً وقبل كل شيء نقوم بالتصريح بمتغير ما وليكن X في الجزء GENERAL وكم يلي :

```
OPTION EXPLICIT
DIM X AS INTEGER
```

ثم نقوم بكتابة شفرات مفاتيح الأوامر

الأمر COMMAND1) عند استخدام هذا الأمر نقوم بإدخال العدد المراد جدول الضرب له)
X=INPUT BOX ("ENTER ANY NUMBER","MULTI-TABLE")

الأمر COMMAND2 :وهو يستخدم لتنظيف مربع القائمة LIST1 من اي شيء موجود في داخله:-

```
LIST1.CLEAR
```

الأمر COMMAND3 :هذا الأمر يستخدم لإيجاد جدول الضرب للعدد X الذي تم إدخاله بواسطة مفتاح الأمر INPUT وذلك باستخدام عبارة FOR ويتم طباعة الناتج جدول الضرب في القائمة LIST1 :-

```
DIM I AS INTEGER
```

```

LIST 1 .CLEAR
FOR I=1 TO 10
  LIST1.ADDITEM I*X
NEXT I

```

شفرة الأمر COMMAND5: وهو يستخدم لإيجاد جدول الضرب للعدد X باستخدام عبارة
 -: DO WHILE –LOOP

```

DIM I AS INTEGER
LIST1.CLEAR
I=1
Do while (I<=10)
  LIST1.ADDITEM I*X
  I=I+1
LOOP

```

شفرة الأمر COMMAND6: وهو يستخدم لإيجاد جدول الضرب للعدد X باستخدام عبارة
 -:DO UNTIL –LOOP

```

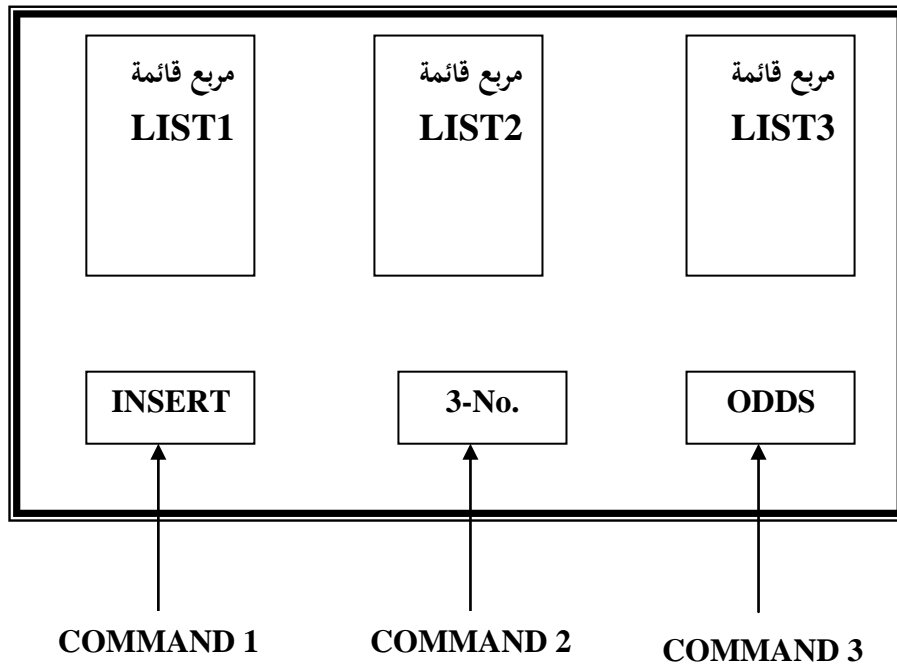
DIM I AS INTEGER
LIST1.CLEAR
I=1
DO UNTIL (I>10)
  LIST1.ADDITEM I *X
  I=I+1
LOOP

```

٥-١١-٣-مثال ٣

يوضح هذا المثال كيفية التعامل مع مربعات القوائم وكيفية التعامل مع عناصر تلك القوائم. في هذا المثال سوف نستخدم مربع قائمة نضع فيه مجموعة من الاعداد (عند التنفيذ) ومربع القائمة آخر نسحب مجموعة من الاعداد التي تقبل القسمة على ٣ بدون باقي من مربع القائمة الاول ونضعها في مربع القائمة الثاني وذلك من خلال مفتاح امر معين وتستخدم مربع قائمة ثالثة نضع فيه الاعداد الفردية من خلال مفتاح آخر معين يتم سحب هذه الاعداد من مربع القائمة الاول وكما يلي :-

إعداد واجهة المشروع



إعداد خصائص الكيانات: -

الإعداد	الخاصية	الكيان
" INSERT "	CAPTION	COMAND1
" ٣ - NO . "	CAPTION	COMAND2
" ODDS "	CAPTION	COMAND3

كتابة شفرات المشروع: -

١- الكيان COMMAND1 :-


```
DIM I AS INTEGER
LIST1.CLEAR
FOR I = 1 TO 50
  LIST1.ADDITEM I
NEXT I
```

- : COMMAND2 ٢-شفرة الكيان

```
DIM I AS INTEGER
LIST2.CLEAR
FOR I = 1 TO 50
  IF VAL(LIST1.LIST(I)) MOD 3=0 THEN
    LIST2.ADDITEM LIST1.LIST(I)
  END IF
NEXT I
```

--: COMMAND3 ٣-شفرة الكيان

```
DIM I AS INTEGER
LIST3.CLEAR
FOR I = 1 TO 50
  IF VAL(LIST1.LIST(I)) MOD 2=1 THEN
    LIST2.ADDITEM LIST1.LIST(I)
  END IF
NEXT I
```

الفصل السادس/المصفوفات

٦-١-١- مقدمة:-

ما هي المصفوفة : المصفوفة (ومعناها MATRIX أو ARRAY أو CLASS) هي عبارة عن هيكل بياني موجود في جميع لغات البرمجة ويتم تمثيله في الحاسبة بمجموعة من مواقع الذاكرة المتجاورة (يجب إن تكون متجاورة) ويكون لجميع عناصر المصفوفة نفس النوع البياني لكن قيمها تختلف من عنصر إلى آخر .

تقسم المصفوفات من حيث تعاملها مع الذاكرة إلى :-

- مصفوفات المتغيرات .
- مصفوفات أدوات التحكم .
- المصفوفات الديناميكية .

إما من حيث إبعاد المصفوفة فأنها تقسم إلى :-

- المصفوفات المتجهة (أو المصفوفات الأحادية) .
- المصفوفات متعددة الأبعاد (ثنائية ، ثلاثية ، الخ) .

٦-٢-٢- مصفوفات المتغيرات :-

إن مصفوفات المتغيرات مفيدة لأنها تساعد على تعقب كميات كبيرة من البيانات بطرق كأنت لتكون غير ممكنة لو استعملت المتغيرات التقليدية. كذلك تتيح مصفوفات المتغيرات على استخدام مجموعة كاملة من القيم باستعمال اسم واحد ثم معالجة القيم بشكل فردي أو جماعي عبر استعمال حلقة FOR – NEXT أو إحدى عبارات do...

٦-٣-٣- إنشاء المصفوفة

يتم إنشاء تصريح المصفوفات في البرنامج مثلما يتم تصريح المتغيرات لكن المكان الذي تصرح فيه المصفوفة يحدد مكان استعمالها أو ما يسمى (مجالها) DOMAIN في البرنامج. إذا تم تصريح المصفوفة محلياً ، يمكن استعمالها في الأجراء الذي صرحته فيه فقط . إما إذا تم تصريحها عموماً في وحده نمطية قياسية فيمكن استعمالها في أي مكان في البرنامج.

٦-٤-٤- الصيغة العامة للمصفوفة:-

إن التركيب النحوي الأساسي لتصريح المصفوفة ثابتة الحجم هو:-

PUBLIC arrayname (DIM1elements,DIM2elements,.....)AS datatype

حيث إن :

- PUBLIC : هي الكلمة الدلالية التي تنشئ مصفوفة عامة ويمكن إن تكون STATIC إذا كان تصريح المصفوفة خاصة .
- arrayname: هو اسم المصفوفة ، وهو اسم متغير يمكن إن يكون أي اسم ويفضل إن يكون ذا معنا يوجي إلى صلة معينة بالمصفوفة.
- DIM1elements : هو عدد عناصر في البعد الأول في المصفوفة.
- DIM2elements: هو عدد العناصر في البعد الثاني في المصفوفة(يمكن إنشاء إبعاد إضافية في المصفوفة) فالمصفوفة التي تحتوي على بعد واحد فأنها تسمى مصفوفة أحادية وإذا كانت تحتوي على بعدين فأنها تسمى مصفوفة ثنائية وهكذا . إما عن عدد عناصر المصفوفة فإنه حاصل ضرب عدد العناصر إبعاد المصفوفة، فإذا كان البعد الأول هو ٥ والبعد الثاني هو ٣ فأن عدد عناصر المصفوفة (٣*٥=١٥) إما إذا كانت المصفوفة ثلاثية وإبعادها هي (٢، ٣، ٥) فأن عدد عناصر المصفوفة هو (٢*٣*٥=٣٠) .

ملاحظة أحيانا وفي حالة المصفوفات الأحادية لا يتم تحديد عدد عناصر المصفوفة وهذا ما يدعى بالمصفوفة الديناميكية.

- datatype : هي الكلمة الدلالية التي تحدد نوع البيانات التي ستخزن في المصفوفة أي انه إذا كان نوع البيانات INTEGER فأن جميع البيانات يجب إن تكون إعداد صحيحة وإذا كان STRING فأنها يجب إن تكون جميعها سلاسل رمزية إلا في حالة النوع البياني variant فإنه يقبل أي نوع بياني .

ملاحظة

بخصوص فهرسة عناصر المصفوفة فإنه يبدأ بشكل تلقائي من الصفر إي انه إذا كان عدد عناصر المصفوفة هو ١٠ فإنه سوف يتم فهرسة العناصر من ٠ إلى ٩ وهي عشرة عناصر . إما إذا أردنا إن تبدأ الفهرسة من ١ فأننا يجب إن نصرح في البداية بهذا الأساس وذلك بوضع العبارة OptionBase1 في الجزء general في نافذة ال code بالضبط كما عملنا مع العبارة . option explicit

٦-٥- التعامل مع المصفوفات بداخل شفرة البرنامج

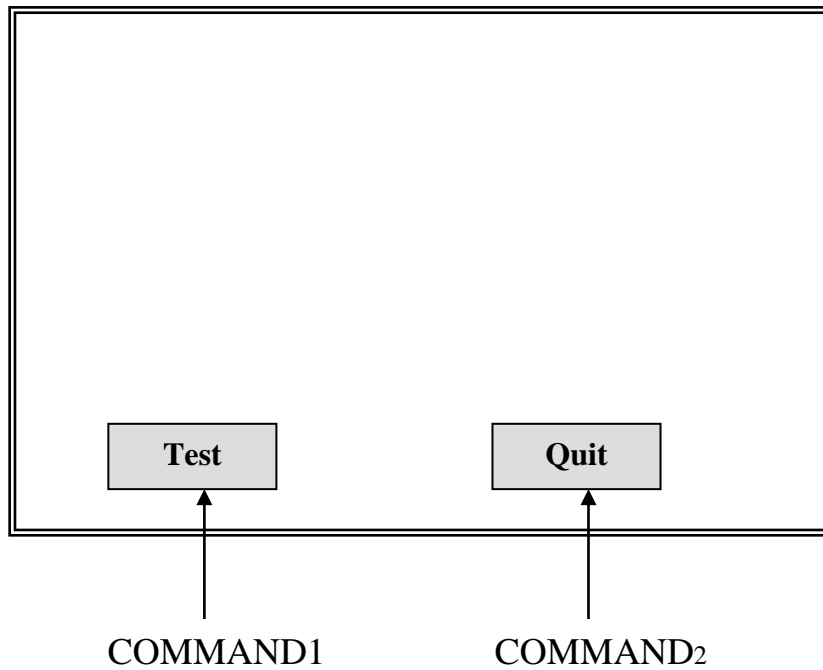
بما إن المصفوفة هي مجموعة من البيانات من نفس النوع لذلك من الضروري جداً عندما نتعامل معها إن نستخدم إحدى دورات التكرار (العبارات التكرارية) وان أفضل العبارات التكرارية الملائمة للتعامل مع المصفوفات هي عبارة for...next حيث إنها عبارة تكرارية غير مشروطة كذلك فأن فيها عداد قيمة الأولية والنهائية معلومتان وهذا هو حال المصفوفة حيث إن القيمة الأولية للعداد تمثل بداية الفهرسة للمصفوفة (الحد الأدنى للمصفوفة) والقيمة النهائية للعداد تمثل نهاية الفهرسة للمصفوفة (الحد الأخير لها) .
وسوف يتضح فيما يأتي ماتم شرحه بخصوص المصفوفات الاحادية والثنائية وذلك في كل من المشاريع الاتية :-

٦-٦- امثلة توضيحية حول المصفوفات:

٦-٦-١- المشروع الأول

في هذا المشروع سوف يتم التعامل مع مصفوفة أحادية تحتوي على مجموعة قيم تمثل مجموعة من الدرجات وسوف يتم اختبار أي من هذه الدرجات هي العليا وأي منها هي الأقل:-

١- تصميم واجهة المشروع



إعداد خصائص الكيان

الإعداد	الخاصية	الكيان
Test	Caption	Command1
Quit	Caption	Command2

إعداد وكتابة شفرات المشروع

١- شفرة الكيان command2 الذي اسمه Quit هي:

end

٢- شفرة الكيان command1 الذي اسمه Test :

```

Dim i as integer ,x as integer ,y as integer
Dim title as string
Static degree(5) as integer
For i=0 to 4
    title="degree"&i
    degree(i) =inputbox ("enter the degree ",title)
Next i
X=degree(0)
For i=1 to 4
    If x>=degree(i) then
        MsgBox (" the highest degree is "& x)
    Else
        Y=x
        X=degree(0)
        degree(i)=y
    End if
Next i
For i= 1 to 4
    If x<degree(i) then
        MsgBox("the lowest degree is "& x)
    Else

```

```
Y=x
X=degree(i)
degree(i)=y
End if
Next i
```

٦-٦-٢-المشروع الثاني

سنتعلم في هذا المشروع طريقة أخرى لقراءة وطباعة عناصر المصفوفة حيث سنستخدم مربعات نص بقدر عدد عناصر المصفوفة وهي في الحقيقة مربع نص واحد ولكن نقوم بتكراره بطريقة النسخ واللصق ،وهذه الطريقة (النسخ واللصق) مفيدة في حالة مصفوفات عناصر التحكم مثلا في حالة مصفوفة من الصور image او مصفوفة من الأشكال shape .

في مثالنا هذا سنقوم بقراءة (ادخال) مصفوفة احادية مكونة من ١٠ اعداد ويتم ترتيبها مره تنازليا ومره تصاعديا ويتم قراءة المصفوفة وطباعتها من خلال نفس مربعات النص.

اعداد واجهة المشروع:-

واجهة المشروع عند تنفيذ الترتيب التصاعدي

خلال اعداد واجهة المشروع وخاصة عندما نريد انشاء مصفوفة عناصر التحكم (في هذا المثال مصفوفة مربعات النص) علينا مراعاة الامور التالية:-

- ننشئ مربع نص على النموذج ونسميه مثلا mtxt ونضبط خصائصه حسبما نريد.
- نضغط click ايمن على مربع النص ونختار الامر نسخ copy ثم نضغط click ايمن على مكان فارغ على سطح النموذج ونختار الامر لصق paste ثم نختار yes نلاحظ تم انشاء مربع نص جديد يحمل نفس اسم مربع النص الاول.
- خلال الخطوة السابقة عندما اخترنا الامر لصق ظهر لنا مربع حوار يحوي مفتاحين هما yes و No ويحتوي على الرسالة الاتية:

you already have a control named 'mtxt'. Do you want to create a (control array?
الرسالة اعلاه تعني (انك تملك عنصر تحكم .هل تريد انشاء مصفوفة عناصر تحكم)

1. اذا اخترنا No سينشئ مربع نص لاعلاقة له بمربع النص الاول.
2. اذا اخترنا Yes سينشئ مربع نص جديد اسمه mtxt والخاصية index له مساوية لـ 1 .

- نلاحظ ان الخاصية index لمربع النص الاول قبل عملية النسخ واللصق كانت فارغة ولكن اصبحت 0 بعد العملية وهذا يعتبر بمثابة عداد لمربعات النص (لمصفوفة عناصر التحكم) .

اعداد خصائص كيانات المشروع:-

الاعداد	الخاصية	الكيان
ترتيب تصاعدي	Caption	Command1
ترتيب تنازلي	Caption	Command2
خروج	Caption	Command3

كتابة شفرات الكيانات:-

1. شفرة الكيان "خروج" : End
2. شفرة الكيان "ترتيب تصاعدي" :

Dim a(10) As Integer

```

Dim i As Integer, j As Integer
Dim x As Single
For i = 0 To 9
    a(i) = Val(mtxt(i))
Next i
For i = 0 To 9
    For j = i + 1 To 9
        If a(j) < a(i) Then
            x = a(j)
            a(j) = a(i)
            a(i) = x
        End If
    Next j
Next i
For i = 0 To 9
    mtxt(i) = a(i)
Next i

```

٣.شفرة الكيان "ترتيب تنازلي" :

```

Dim a(10) As Integer
Dim i As Integer, j As Integer
Dim x As Single
For i = 0 To 9
    a(i) = Val(mtxt(i))
Next i
For i = 0 To 9
    For j = i + 1 To 9
        If a(j) > a(i) Then
            x = a(j)
            a(j) = a(i)
            a(i) = x
        End If
    Next j
Next i
For i = 0 To 9
    mtxt(i) = a(i)
Next i

```

٦-٦-٣-المشروع الثالث

في هذا المشروع سنتعلم كيفية قراءة عناصر المصفوفة (احادية او ثنائية) من خلال الاداة inputbox وكيفية طباعة عناصر المصفوفة على النموذج باستخدام العبارة . Print

في هذا المشروع سنقوم بقراءة مصفوفة مربعة (3x3) ثم ابدال عناصر الصف الاول مع عناصر الصف الاخير وطباعة المصفوفة على النموذج قبل وبعد عملية الابدال.

اعداد واجهة المشروع:-



اعداد خصائص الكيانات:-

الكيان	الخاصية	الإعداد
Command1	Caption	Read/replace
Command2	Caption	exit

كتابة شفرات المشروع:-

- شفرة الامر end : exit

- شفرة الامر read/replace

```
Dim a(3, 3) As Integer
Dim i As Integer, j As Integer
Dim x As Single
For i = 0 To 2
  For j = 0 To 2
    a(i, j) = InputBox("inter any number")
  Next j
Next i
For i = 0 To 2
  For j = 0 To 2
    Print a(i, j) & " ";
  Next j
  Print
Next i
For j = 0 To 2
  x = a(0, j)
  a(0, j) = a(2, j)
  a(2, j) = x
Next j
Print: Print
For i = 0 To 2
  For j = 0 To 2
    Print a(i, j) & " ";
  Next j
Print
Next i
```

في المشروع الأتي سنستخدم طريقة تعتبر (من وجهة نظر المبرمج) من افضل الطرق للتعامل مع المصفوفات في بيئة VB حيث انه في المشروع الاول تم استخدام الدالة MsgBox لاطهار النتائج وهذا لايسمح لنا بمعاينة جميع عناصر المصفوفة في آن واحد ، في المشروع الثاني

استخدمنا مربعات نص وهذه الحالة غير ايجابية حيث لايمكن استخدامها في حالة كون عناصر المصفوفة ذات عددكبير جدا (مثلا مصفوفة تتكون من ١٠٠ عنصر) ، وهذا يتطلب منا انشاء مربعات نص كثيرة جدا ، كذلك غير فعالة في حالة تحديد عناصر المصفوفة خلال تنفيذ البرنامج، اما في المشروع الثالث فقد تم استخدام سطح النموذج لطباعة عناصر المصفوفة وهذه الحالة غير فعالة ايضا حيث انها سوف تشغل مساحة غير معروفة مسبقا من سطح النموذج وهذا ما يؤثر سلبا على الكيانات الاخرى.

٦-٤-٤-المشروع الرابع:-

في هذا المشروع سنتعلم كيفية التعامل من خلال الأداة Inputbox لقراءة عناصرها واستخدام مربع نص واحد فقط لإظهار نتائج المعالجة كطباعة جميع عناصر المصفوفة او جزء منها ،سنقوم في هذا المشروع بقراءة مصفوفة مربعه (٣x٣) باستخدام الاداة inputbox ثم نقوم بابدال عناصر العمود الاول مع عناصر العمود الاخير من المصفوفة ثم نقوم بطباعة المصفوفة بعد عملية التبديل في مربع النص.

اعداد واجهة المشروع:-



اعداد خصائص الكيانات:-

الكيان	الخاصية	الإعداد
--------	---------	---------

Read/replace	Caption	Command1
Exit	Caption	Command2
txt1	Name	Text1
True	Multiline	

كتابة شفرات المشروع:-

• شفرة الامر exit :

• شفرة الامر read/replace

```

Dim a(3, 3) As Integer
Dim i As Integer, j As Integer
Dim x As Single
For i = 0 To 2
  For j = 0 To 2
    a(i, j) = InputBox("inter any number")
  Next j
Next i
For i = 0 To 2
  For j = 0 To 2
    txt1.Text = txt1.Text & a(i, j) & Space(2)
  Next j
  txt1.Text = txt1.Text & vbNewLine
Next i
For i = 0 To 2
  x = a(i, 0)
  a(i, 0) = a(i, 2)
  a(i, 2) = x
Next i
txt1.Text = txt1.Text & vbNewLine
  For i = 0 To 2
  For j = 0 To 2
    txt1.Text = txt1.Text & a(i, j) & Space(2)
  Next j
  txt1.Text = txt1.Text & vbNewLine
Next i

```

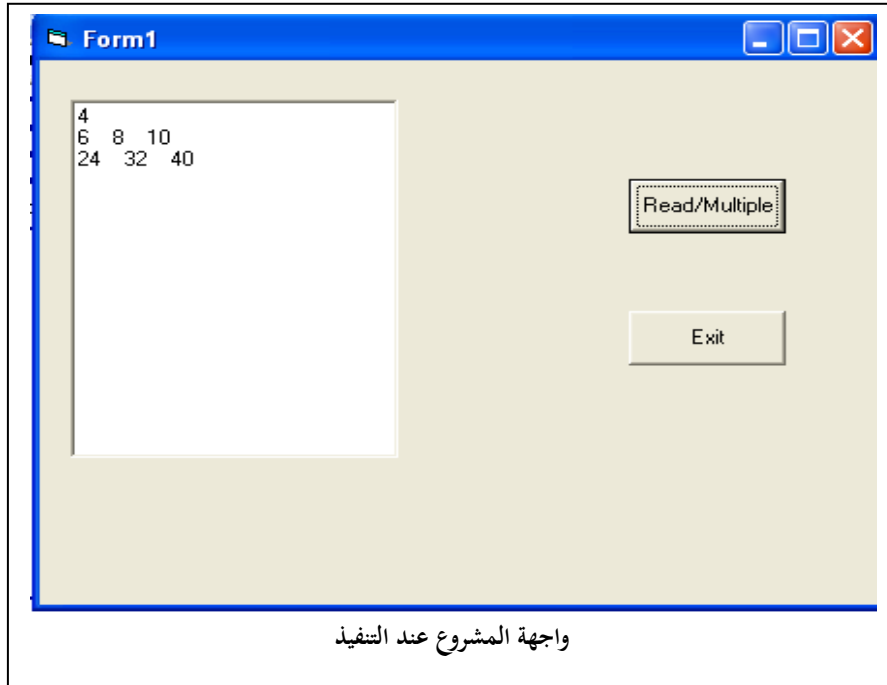
٦-٧-العمليات الحسابية والمصفوفات:-

سنتناول هنا كيفية اجراء العمليات الحسابية(جمع،طرح،ضرب،قسمة) بين عنصر ومصفوفة وبين مصفوفة ومصفوفة أخرى .

٦-٧-١-العمليات الحسابية بين عنصر ومصفوفة:-

إن إجراء أي عملية حسابية (جمع، طرح، ضرب، قسمة) بين أي عنصر ومصفوفة تتم بكل سهولة (في كلا حالتي المصفوفات الاحادية والثنائية) حيث يتم اجراء تلك العملية بين ذلك العنصر وبين جميع عناصر المصفوفة بحيث تنتج لنا مصفوفة ابعادها نفس ابعاد عناصر المصفوفة، وكما موضح في الامثلة التالية:

مثال ١: نأخذ هنا مثال حول ضرب عنصر وليكن X في مصفوفة احادية ولتكن A وكما يلي:
اعداد واجهة المشروع:-



اعداد خصائص الكيانات:-

الكيان	الخاصية	الإعداد
--------	---------	---------

Read/Multiple	Caption	Command1
Exit	Caption	Command2
txt1	Name	Text1
True	Multiline	

كتابة شفرات المشروع:-

• شفرة الامر end : exit

• شفرة الامر read/multiple

```
Dim a(3) As Integer
Dim i As Integer
Dim x As Single
x = InputBox("enter any single number")
For i = 0 To 2
    a(i) = InputBox("inter any number")
    Next i
    txt1.Text = x & vbNewLine
For i = 0 To 2
    txt1.Text = txt1.Text & a(i) & Space(4)
Next i
txt1.Text = txt1.Text & vbNewLine
For i = 0 To 2
    txt1.Text = txt1.Text & (a(i) * x) & Space(4)
Next i
```

خلال شفرة الامر Read/Multiple قمنا بالاتي:

تعريف العنصر X المراد ضربه بالمصفوفة، ثم تعريف مصفوفة احادية (a) بعدها قمنا بقراءة العنصر ثم قراءة المصفوفة بعبارة For وكلا القراءتين تمت بطريقة الدالة Inputbox بعدها تم طباعة العنصر في مربع النص ثم طبعنا المصفوفة في سطر جديد في مربع النص وذلك باستخدام الطريقة vbnewline وتعني تحريك المؤشر الى سطر جديد ، اما الدالة space(4)

فتم استخدامها لحجز فراغات بقدر اربع خانات ويمكن زيادة او تقليل هذا العدد من الفراغات حسب الرغبة ، لكي تظهر فراغات بين عناصر المصفوفة .
بعدها قمنا باجراء عملية الضرب بين العنصر وجميع عناصر المصفوفة وطباعة عناصر المصفوفة الناتجة في سطر جديد بداخل مربع النص.
مثال ٢:- في هذا المثال سوف نقوم باجراء عملية الضرب بين عنصر ومصفوفة مربعه، وكما موضح ادناه:

اعداد واجهة المشروع:-



The screenshot shows a Windows form titled "Form1" with a blue title bar. The form contains a text area on the left displaying a 7x3 grid of numbers: 7, 1 2 3, 4 5 6, 7 8 9, 7 14 21, 28 35 42, 49 56 63. To the right of the text area are two buttons: "Read/Multiple" and "Exit".

واجهة المشروع عند التنفيذ

اعداد خصائص الكيانات:-

الإعداد	الخاصية	الكيان
Read/Multiple	Caption	Command1
Exit	Caption	Command2
txt1	Name	Text1
True	Multiline	

كتابة شفرات المشروع:-

شفرة الامر end : exit

شفرة الامر read/multiple

```

Dim a(3, 3) As Integer
Dim i As Integer, j As Integer
Dim x As Single
x = InputBox("enter any single number")
For i = 0 To 2
  For j = 0 To 2
    a(i, j) = InputBox("inter any number")
  Next j
  Next i
  txt1.Text = x & vbNewLine
  For i = 0 To 2
  For j = 0 To 2
    txt1.Text = txt1.Text & a(i, j) & Space(4)
  Next j
  txt1.Text = txt1.Text & vbNewLine
  Next i
For i = 0 To 2
  For j = 0 To 2
    txt1.Text = txt1.Text & (a(i, j) * x) & Space(4)
  Next j
  txt1.Text = txt1.Text & vbNewLine

```


Next i

٦-٧-٢- العمليات الحسابية بين مصفوفة ومصفوفة:-

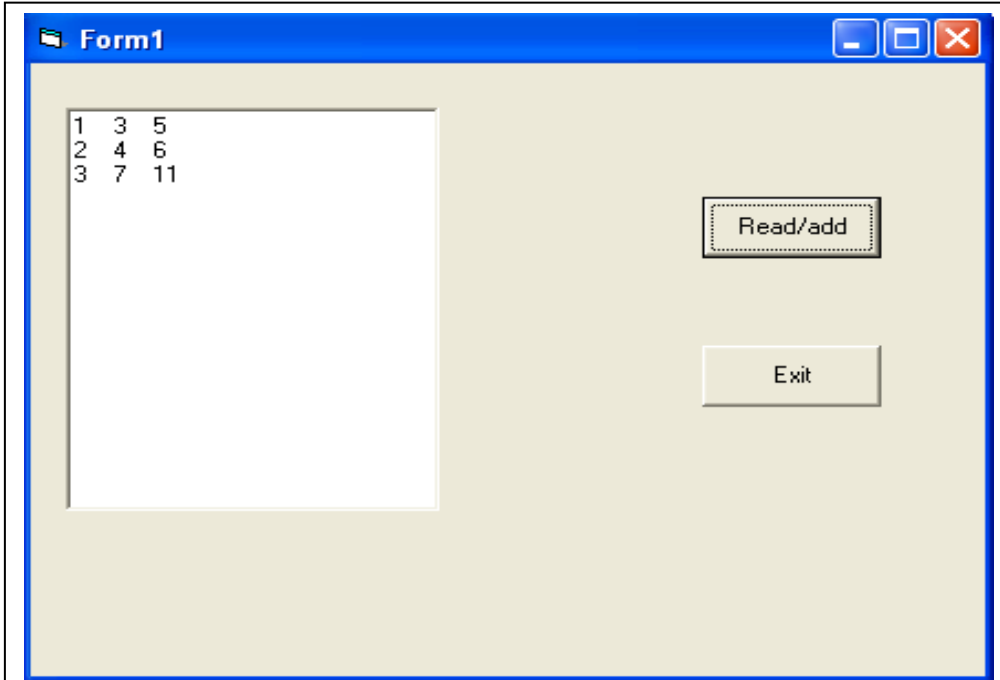
عند اجراء أي عملية حسابية من العمليا الاساسية الاربع بين أي مصفوفتين فانه يجب ان تكون كلا المصفوفتان متساويتان في الابعاد ماعدا عملية ضرب المصفوفات الثنائية وكما سياتي شرحه لاحقا في موضوع لاحق.

مثال ١:- (اجراء عملية الجمع بين مصفوفتين احاديتين)

سوف نقوم بتعريف كلا المصفوفتين a و b وذات بعد واحد ومصفوفة ثالثة c (تمثل المصفوفة الناتجة) بنفس البعد. ثم نقوم بطباعة المصفوفتين قبل الجمع وطباعة المصفوفة الناتجة.

اخذنا في هذا المثال اجراء عملية الجمع ويمكن اخذ أي عملية حسابية اخرى.

اعداد واجهة المشروع:-



The screenshot shows a Windows application window titled "Form1". Inside the window, there is a text area containing a 3x3 matrix of numbers:

1	3	5
2	4	6
3	7	11

Below the text area, there are two buttons: "Read/add" and "Exit".

واجهة المشروع عند التنفيذ

اعداد خصائص الكيانات:-

الإعداد	الخاصية	الكيان
Read/add	Caption	Command1
Exit	Caption	Command2
txtadd	Name	Text1
True	Multiline	

كتابة شفرات المشروع:-

شفرة الامر end : exit

شفرة الامر Read/add

```
Dim a(3) As Integer, b(3) As Integer, c(3) As Integer
```

```
Dim i As Integer
```

```
For i = 0 To 2
```

```
    a(i) = InputBox("inter any number", "array A")
```

```
    txtadd.Text = txtadd.Text & a(i) & Space(4)
```

```
    Next i
```

```
    txtadd.Text = txtadd.Text & vbNewLine
```

```
For i = 0 To 2
```

```
    b(i) = InputBox("inter any number", "array B")
```

```
    txtadd.Text = txtadd.Text & b(i) & Space(4)
```

```
    Next i
```

```
    txtadd.Text = txtadd.Text & vbNewLine
```

```
For i = 0 To 2
```

```
    txtadd.Text = txtadd.Text & (a(i) + b(i)) & Space(4)
```

```
Next i
```

مثال ٢:- (اجراء عملية الطرح بين مصفوفتين مربعيتين)

اعداد واجهة المشروع:-

واجه المشروع عند التنفيذ

اعداد خصائص الكيانات:-

الإعداد	الخاصية	الكيان
Read/sub	Caption	Command1
Exit	Caption	Command2
txtsub	Name	Text1
True	Multiline	

كتابة شفرات المشروع:-

شفرة الامر end : exit

شفرة الامر read/sub

Dim a(3, 3) As Integer, b(3, 3) As Integer

```

Dim i As Integer, j As Integer
For i = 0 To 2
  For j = 0 To 2
    a(i, j) = InputBox("inter any number", "array A")
    txtsub.Text = txtsub.Text & a(i, j) & Space(4)
  Next j
  txtsub.Text = txtsub.Text & vbNewLine
  Next i
  txtsub.Text = txtsub.Text & vbNewLine
  For i = 0 To 2
  For j = 0 To 2
    b(i, j) = InputBox("inter any number", "array B")
    txtsub.Text = txtsub.Text & b(i, j) & Space(4)
  Next j
  txtsub.Text = txtsub.Text & vbNewLine
  Next i
  txtsub.Text = txtsub.Text & vbNewLine
  For i = 0 To 2
  For j = 0 To 2
    txtsub.Text = txtsub.Text & (a(i, j) - b(i, j)) & Space(4)
  Next j
  txtsub.Text = txtsub.Text & vbNewLine
  Next i

```

٦-٨- ضرب المصفوفات الثنائية

ان جميع العمليات الحسابية الاساسية الاربع يمكن إجراؤها بسهولة ماعدا عملية ضرب المصفوفات فانه يشترط ان يكون عدد اعمدة المصفوفة الاولى مساو لعدد صفوف المصفوفة الثانية . يكون ابعاد المصفوفة الناتجة هي عدد الصفوف مساو لعدد صفوف المصفوفة الاولى وعدد الاعمدة هو مساو لعدد اعمدة المصفوفة الثانية .

فاذا كانت $a(3,2)$ و $b(2,2)$ فان المصفوفة الناتجة هي $c(3,2)$ اما عملية الضرب فتتم حسب المعادلة الرياضية التالية =:

$$c(i, j) = \sum a(i, k) * b(k, j)$$

اي انه كل عنصر من عناصر المصفوفة الناتجة هو حاصل مجموع ضرب عناصر صف هذا العنصر بالمصفوفة الاولى مع عناصر العمود المقابل له في المصفوفة الثانية فاذا اخذنا عناصر المصفوفتين a,b السابقتين فان عناصر المصفوفة c الناتجة تكون بالشكل التالي:-

(بحيث يبدأ المصفوفات بـ((0,0))

$$c(0,0)=[a(0,0)*b(0,0)]+[a(0,1)*b(1,0)]$$

$$c(0,1)=[a(0,0)*b(0,1)]+[a(0,1)*b(1,1)]$$

$$c(1,0)=[a(1,0)*b(0,0)]+[a(1,1)*b(1,0)]$$

$$c(1,1)=[a(1,0)*b(0,1)]+[a(1,1)*b(1,1)]$$

$$c(2,0)=[a(2,0)*b(0,0)]+[a(2,1)*b(0,1)]$$

$$c(2,1)=[a(2,1)*b(1,0)]+[a(2,1)*b(1,1)]$$

المثال التالي هو حول ضرب مصفوفتين مربعيتين:

اعداد واجهة المشروع:-

The screenshot shows a Windows form window titled 'Form1'. On the left side, there is a grid of numbers arranged in three rows and two columns. The first row contains '1 3', the second row contains '5 7', and the third row contains '9 3'. Below these, there is a blank space, followed by another two rows and two columns of numbers: the first row contains '2 4', the second row contains '6 8', the third row contains '20 28', the fourth row contains '52 76', and the fifth row contains '36 60'. To the right of the grid, there are two buttons: 'Read/multiple' and 'Exit'.

واجهة المشروع عند التنفيذ

اعداد خصائص الكيانات:-

الإعداد	الخاصية	الكيان
Read/multiple	Caption	Command1
Exit	Caption	Command2
txtmult	Name	Text1
True	Multiline	

كتابة شفرات المشروع:-

شفرة الامر end : exit

شفرة الامر read/multiple

```

Dim a(3, 2) As Integer, b(2, 2) As Integer
Dim c(3, 2) As Integer
Dim i As Integer, j As Integer, k As Integer
For i = 0 To 2
  For j = 0 To 1
    a(i, j) = InputBox("inter any number", "array A")
    txtmult.Text = txtmult.Text & a(i, j) & Space(4)
  Next j
  txtmult.Text = txtmult.Text & vbNewLine
Next i
txtmult.Text = txtmult.Text & vbNewLine
For i = 0 To 1
  For j = 0 To 1
    b(i, j) = InputBox("inter any number", "array B")
    txtmult.Text = txtmult.Text & b(i, j) & Space(4)
  Next j
  txtmult.Text = txtmult.Text & vbNewLine
Next i
txtmult.Text = txtmult.Text & vbNewLine
For i = 0 To 2
  For j = 0 To 1
    For k = 0 To 1
      c(i, j) = c(i, j) + (a(i, k) * b(k, j))
    
```

```
Next k
txtmult.Text = txtmult.Text & c(i, j) & Space(4)
Next j
txtmult.Text = txtmult.Text & vbNewLine
Next i
```

الفصل السابع/الدوال والبرامج الفرعية Functions & Procedures

٧-١-١- مقدمة:

بشكل عام، الدوال والبرامج الفرعية عبارة عن برامج-او اجزاء برامج-ثانوية يتم استخدامها لاداء غرض معين ومن فوائدها:

-تقليل وتلافي التكرار في بناء البرامج مرة اخرى.

-تقليل الوقت المطلوب لبناء البرامج والمشاريع.

-التقليل من الذاكرة المطلوبة لشفرات وبيانات المشروع.

هنالك نوعين من البرامج الفرعية من حيث مكان البناء والتصريح هما:

❖ الوحدات النمطية Module وهي تمثل احد الكيانات الاساسية التي تقدمها لغة VB للمبرمجين حيث يتم إنشاؤها من خلال قائمة `project\add new module` (كما تم شرحه في الفصل الاول ضمن شرح شريط القوائم) في هذه الوحدة النمطية يتم تعريف المتغيرات والكيانات التي يتم استخدامها فقط ويتم كتابة البرامج المراد اجراءها كاملة.

❖ البرامج الفرعية الخاصة بالمشروع المراد عمله وهي تشتمل على الدوال والبرامج الفرعية، ويتم تعريفها في الجزء General الموجود في كل نموذج(في نافذة الشفرات Code Window).

الفرق بين هذين النوعين هو ان الوحدات النمطية Module يمكن استخدامها في أي كيان باي نموذج موجود ضمن المشروع اما النوع الاخر فانه يمكن استخدامه فقط ضمن الكيانات التابعة لنموذج واحد فقط.

٧-٢- الدالة Function :-

الشكل العام لتعريف أي دالة هو :-

```
[public| private] function name [(argument list)] :data type
[statement(s)]
[statement(s)]
End Function
```

حيث ان:

[public| private] تعني ان هذه الدالة هي ذات تصريح خاص او عام فاذا كُأنت ذا تصريح خاص نكتب فقط private واذا كُأنت ذا تصريح عام نكتب public،وهي عبارة اختيارية يمكن عدم استخدامها.

function name : اسم الدالة وهو جزء يجب استخدامه.

[(argument list)]: تعني قائمة المعاملات الخاصة بالدالة ويتم وضعها بين اقواس وهي اختيارية حيث في حالة عدم وجود معاملات لا يتم كتابة هذا الجزء.

Data type: تعني النوع البياني الخاص بالدالة حصرا.

ويمكن استخدام الصيغه اعلاه في وحدة نمطية مستقلة او في نفس نافذة الشفرات التابعه لنفس النموذج وذلك في الجزء General .

عند استدعاء أي داله يتم كتابة اسمها ثم اقواس اذا كُأنت تحتوي على معاملات فيتم وضع اسماء المعاملات بين الاقواس واذا كُأنت لاتحتوي على معاملات لا يتم وضع اقواس.

الفرق بين الدالة والبرنامج الفرعي هو ان الدالة تقوم بارجاع قيمة ذات نوع بياني واحد اما البرنامج الفرعي فان يقوم بارجاع قيمة واحدة او عدة قيم من انواع بيانية مختلفة .

مثال حول استخدام الدول Function :-

نقوم اولا بفتح وحدة نمطية Module ونكتب فيها الشفرة التالية :

```
Function numeric(n as integer) as string
```

```
  Select case n
```

```
    Case 1:numeric="first"
```

```
    Case 2:numeric="second"
```

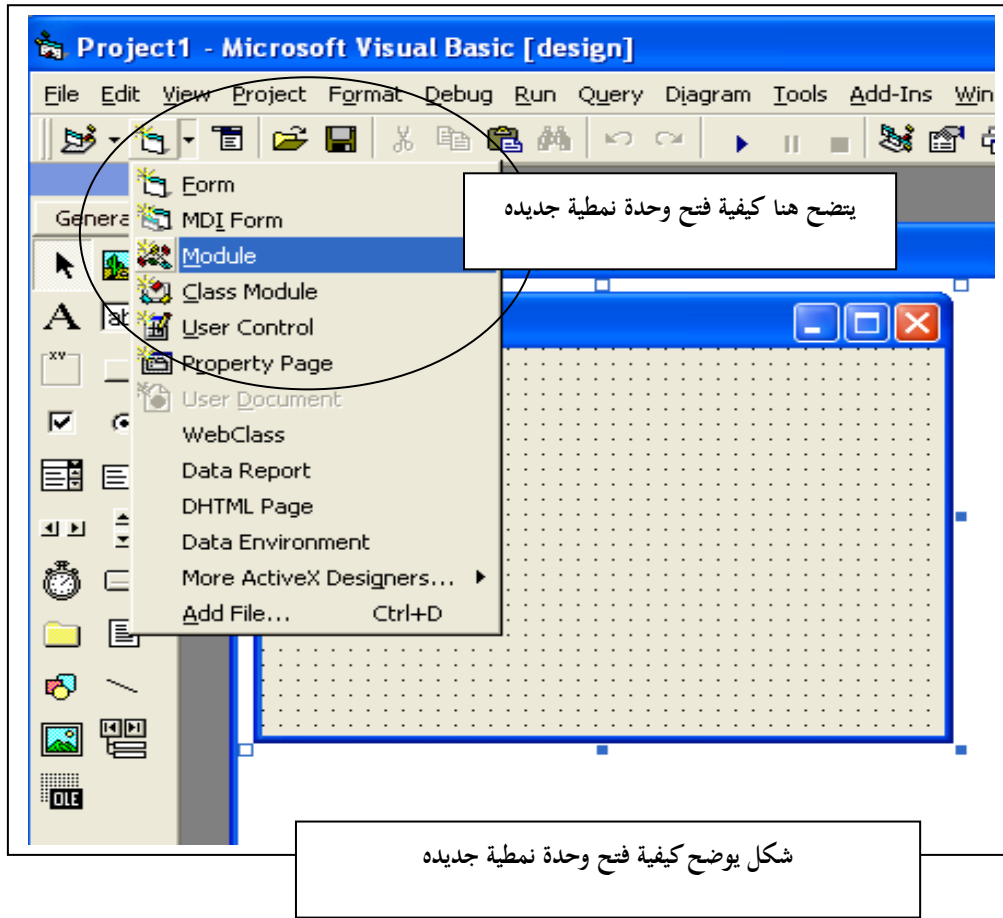
```
    Case 3:numeric="third"
```

```
    Case 4:numeric="fourth"
```

```
    Case 5:numeric="fifth"
```

```
  End select
```

```
End function
```



ثم نضع على النموذج مفتاح امر ومربعي نص ونكتب في مفتاح الامر الشفرة التالية

```
Dim a as integer  
A=val(text1.text)  
Text2.text=numeric(a)
```

شكل المشروع عند التنفيذ

في هذا المشروع نقوم اولا بكتابة رقم (من ١ إلى ٥) في مربع النص الاول ثم نضغط click ايسر على مفتاح الامر حيث سوف يقوم البرنامج بطباعة هذا الرقم كتابة في مربع النص الثاني

٧-٣- دوال التعامل مع النصوص :-

في لغة VB هنالك مجموعة من الدوال الجاهزة التي تسهل العمل والتعامل مع النصوص وهذه الدوال هي :

الدالة	استخدامها	مثال
Len(s)	تفيد طول السلسلة S	Len("ali") فأنها تعيد ٣
Left (S,N)	تعيد n رموز السلسلة	Left("ali",2) فأنها تعيد ali

	S بدأ من يسار السلسلة s	
Right (S,n)	تعيد n من رموز السلسلة s بدا من يمين السلسلة s	"I" Right ("ali",1) فأنها تعيد "I"
Mid (s,n,m)	تعيد m من رموز السلسلة s بدء من الرمز n	Mid ("mahamed",3,5) فأنها تعيد "mahamed"
Ltrim(s)	تمسح الفراغات من يسار السلسلة S إذا كان هنالك فراغات	Ltrim (" ali ") فأنها تعيد ali
Rtrim(s)	تمسح الفراغات من يمين السلسلة s	Rtrim (" ali ") فأنها تعيد ali
Trim(s)	تمسح الفراغات من يمين ويسار السلسلة s	trim(" ali ") فأنها تعيد ali
Ucase(s)	تحول أحرف السلسلة s إلى أحرف كبيرة	Ucase (" ali ") فأنها تعيد ALI
Lcase (s)	تحول أحرف السلسلة s إلى أحرف صغيرة	Lcase (" ali ") فأنها تعيد ali
ASC(S)	تعيد رقم الحرف S الموجود في جدول رموز ASCII	ASC (" ali ") فأنها تعيد ٦٥
CHR(N)	تعيد الحرف الذي رقمه N في جدول رموز ASCII	CHR(65) فأنها تعيد "A"

٧-٤- البرامج الفرعية Procedures :-

الصيغه العامة لتعريف البرامج الفرعية هي:-

[public| private] sub name [(argument list)]
[statement(s)]

[statement(s)]

End sub

ويتم استدعاء البرنامج الفرعي اما بكتابة اسمه فقط اذا كان لا يحتوي على على معاملات او بكتابة الدالة call ثم اسم البرنامج الفرعي ووضع قوسين هلاليين بعد اسم البرنامج ؛ او بكتابة اسم البرنامج الفرعي فقط ثم وضع المعاملات بين اقواس. ويمكن استخدام الصيغه اعلاه في وحدة نمطية مستقلة او في نفس نافذة الشفرات التابعه لنفس النموذج وذلك في الجزء General

.

الفصل الثامن/الملفات وقواعد البيانات Files and DataBase

٨-١- الملفات Files:-

الملف: هو عبارة عن سلسلة من الحروف والارقام المتتابعه ذات حجم معين ويتم تخزينه على وحدة تخزين خارجية Secondary Memory .

تستطيع لغة Visual Basic التعامل مع ثلاث انواع من الملفات هي :

١. الملفات المتتابعة Sequential Files .

٢. الملفات العشوائية Random Access Files .

٣. الملفات الثنائية Binary Files .

وفيما يلي شرح مفصل لكيفية التعامل مع هذه الأنواع الثلاث:

٨-٢- الملفات المتتابعة Sequential Files :-

يتكون هذا الملف من سلسلة من الحروف والارقام مقسمة كالسطور وسميت بالمتتابعه لان عملية قراءتها والبحث فيها عن أي رمز او حرف او جملة او كلمه او فقرة تكون بالتتابع فمثلا عندنا نريد ان نقرأ الرمز السادس الموجود في السطر الخامس فان المعالج يمر على جميع الرموز التي تسبق هذا الرمز . وهكذا ،مما يؤدي الى ضياع وقت في عملية البحث. وتتميز هذه الملفات بانها يمكن قراءة محتوياتها باستخدام أي معالج للنصوص.

■ العمليات التي يمكن إجرائها على الملفات المتتابعه:

١. انشاء ملف جديد.

٢. فتح ملف موجود مسبقا.

٣. القراءة من ملف.

٤. الكتابة في ملف جديد.

٥. الاضافة الى ملف موجود مسبقا.

٦. غلق ملف.

وفيما يلي شرح موجز لهذه العمليات:

١. انشاء ملف جديد:

لا يوجد امر خاص بانشاء ملف جديد ولكن يمكن اعتبار عملية فتح أي ملف للقراءة منه اوللكتابة فية هي بمثابة انشاء ملف جديد.

٢. فتح ملف:

الصيغة العامة لامر فتح الملف في لغة VB هي:

طول السجل=len رقم اختياري للملف #for input اسم الملف Open

حيث ان:

- اسم الملف: هو الاسم المسموح فيه داخل نظام الـ(DOS) و/او امتداده أي يتكون من ثمانية احرف كحد اقصى ثم يتبع بـ(.) ثم امتداد ذلك الملف وهو اما يكون dat او txt.
- For input: ويقصد بها ان هذا الملف مفتوح لغرض القراءة منه فقط ولكي يفت للكتابة فيه نستبدل هذه الجملة بـ output for ولكي نفتح الملف للاضافة اليه نستبدلها بـ for append .
- رقم اختياري للملف: وهو رقم ينسب لهذا الملف ويتم التعامل بهذا الرقم مع الملف بدلا من اسمه الحقيقي.
- طول السجل: ويقصد به بالخرين الوقتي Buffer، والمقصود به هنا هو عدد الرموز القصوى التي يمكن التعامل بها مع الملف في العملية الواحدة-قراءة اوكتابة-فلو حددنا هذا المتغير بـ ٨٠ -على سبيل المثال-فاننا سنتعامل مع الملف بوحدات طولها ٨٠ رمز في المرة الواحدة .

ملاحظة:

- ١- بما أن اسم الملف هو نص إذا يجب وضعه بين علامتي تنصيص مزدوجه(علامتي اقتباس) " " .
- ٢- اذا كان الملف المراد التعامل معه موجود ضمن المجلد الحالي الذي نعمل بداخله (المجلد الذي يحوي المشروع المنشأ) ،فاننا فقط نكتب اسم الملف وهيئته ،اما اذا كان الملف المراد التعامل معه خارج المجلد الموجود فيه المشروع فاننا يجب ان نكتب امتداده بالتام مثلا : "C:\My Document\1.txt" .

٣- القراءة من الملف:

الصيغة العامة لامر القراءة من الملف هي:

متغيرالقراءة , رقم الملف الاختياري#Line input

حيث ان :رقم الملف الاختياري:تم شرحه سلفا

متغير القراءة:ويقصد به متغير من النوع string حيث يقوم معالج اللغة بقراءة سطر

كامل من

الملف ويضعه بداخل هذا المتغير .

مثال:في هذا المثال نقوم بقراءة متغير رمزي اسمه linenead ثم نقوم بفتحملف اسمه mywork وهو موجود بنفس المجلد نفتحه للقراءة ونسميه ب(١) ونحدد طول الـ buffer بصفر ثم نقرأ الملف سطر بعد اخر باستخدام الدوارة Do...loop ويتم عرض محتويات الملف باستخدام الدالة msgbox التي تعرض سطر كل مرة :

```
Dim linenead as string
Open "mywork" for input as #1 len=0
Do until eof(1)
    Line input #1,linenead
    MsgBox linenead
Loop
Close #1
```

٤- الكتابة في الملف:-

الصيغة العامة لامر الكتابة في الملف هي :

متغير الكتابة, رقم الملف الاختياري#Print

حيث ان : رقم الملف الاختياري:وهو ما تم التصريح به عند فتح الملف للطباعة .

متغير الكتابة:وهو اسم المتغير المراد كتابة محتوياته داخل الملف.

المثال التالي يوضح عملية الكتابة في الملف:

```
Dim firstname as string
Dim secondname as string
Open "mywork" for output as #1 len=100
Print #1 "this is my first time to write a file"
```



```
Firstname="Ahmed"  
Secondname="Mohammed"  
Print #1, firstname, secondname  
Close #1
```

ملاحظات

- ❖ عند فتح الملف للكتابة فان معالج لغة VB يقوم بمسح جميع محتويات ذلك الملف وتهيئته للكتابة فيه.
 - ❖ إذا أردنا الكتابة في الملف بدون فقدان في محتوياته السابقه فيجب ان نفتح الملف للاضافة.
 - ❖ لا يمكن إجراء عمليتين على الملف في وقت واحد ، فالملف المفتوح للكتابة او للاضافة لايمكن القراءة منه الا اذا تم غلقة ثم فتحة مرة اخرى للقراءة ، والعكس بالعكس.
- ٥- غلق الملف:-

عند فتح أي ملف لاجراء أي عملية عليه فيجب غلقه قبل انهاء البرنامج ، الصيغه العامة لامر غلق الملف هي:

رقم الملف الاختياري#Close

٨-٣- الدوال المستخدمه مع الملفات:

١) EOF (وهي اختصار لـ End Of File) وتستخدم لاختبار وصول المؤشر الى نهاية الملف او لا وتستخدم بالصيغه التاليه:

If EOF(رقم الملف الاختياري) then

Msgbox "End Of File"

End if

٢) Filelen() :وهي دالة تعيد حجم الملف بالاحرف وتستخدم بالصيغة:

An=filelen(رقم الملف الاختياري)

حيث ان An متغير رقمي

٨-٤- الملفات ذات الدخول العشوائي Random Access Files :-

ويقصد بها تلك الملفات التي يتم ترتيب محتوياتها بشكل عشوائي وليس تسلسلي بحيث عند الحاجة لقراءة مجموعه من البيانات منها فإننا لا نضطر للمرور على جميع البيانات السابقة للبيانات المطلوبة وانما نصل اليها من خلال رمز يشير للخلية التي تحوي تلك البيانات .

■ العمليات التي يمكن القيام بها على الملفات ذات الدخول العشوائي :-

١. تعريف تركيب السجل.

٢. فتح الملف .

٣. القراءة من الملف.

٤. تعديل سجل داخل الملف.

٥. الاضافة للملف.

٦. غلق الملف.

١-تعريف تركيب السجل:

السجل:وهو هيكل بياني يتضمن عدة انواع من البيانات ؛ولمعرفة فكرة السجل بشكل أدق نأخذ مثلا بيانات هوية الطالب ،فهوية الطالب تتضمن رقم الطالب(تعريف الطالب ID) واسم الطالب وعنوانه وتاريخ ميلاده .

نلاحظ ان تعريف الطالب هو نوع رقمي integer واسم الطالب وعنوانه هي سلاسل رمزية string وتاريخ ميلاده هو من نوع التاريخ الذي هو سجل فرعي اخر .

الصيغة العامة لتعريف تركيب السجل هي:

اسم نوع السجل Type

نوعه as اسم الحقل الاول

نوعه as اسم الحقل الثاني

.

.

نوعه as اسم الحقل الاخير

End type

المثال التالي يوضح كيفية تعريف نوع سجل يحتوي على بيانات طالب وهي رقمه(تعريفه ID) واسمه وعنوانه وتاريخ ميلاده (تاريخ الميلاد هو سجل اخر يتم تعريفه قبل تعريف سجل البيانات الرئيسي ويتضمن حقول اليوم والشهر والسنة) ؛كذلك سنتعلم كيفية تعريف متغير كنوع سجل او مصفوفة من السجلات array of record وكيفية قراءة هذه المصفوفه والمثال هو:

Type dat

```

Day as integer
Month as integer
Year as integer
End type
Type student
  Id as integer
  Name as string*30
  Address as string*50
  Brithdat as dat
End type
Dim my_student as student
Dim class_student(10) as student
Dim I as integer
For i=0 to 9
  Class_student(i).id=inputbox("inter the ID of the student")
  Class_student(i).name=inputbox("inter the name of the student")
  Class_student(i).address=inputbox("inter the address of the student")
  Class_student(i).brithdat.day=inputbox("inter the number of day of the student")
  Class_student(i).brithdat.month=inputbox("inter the number of month of the student")
  Class_student(i).brithdat.year=inputbox("inter the number of yearbrithday of the student")
Next i

```

٢-فتح الملف:

الصيغه العامة لامر فتح الملف هي :

Open اسم الملف الخارجي for random as# len= رقم الملف الاختياري اسم الملف الخارجي
السجل

حيث ان :

اسم الملف الخارجي: هو امتداد واسم الملف المراد التعامل معه.

رقم الملف الاختياري: أي رقم يتم تسمية الملف به ليتم التعامل مع الملف من خلال هذا

الاسم(الرقم)

طول السجل: هو طول السجل ويمكن حسابه باستخدام الدالة len()

Example

A=len(student)
Open "C:\DAD\aa" for random as 2 len=a

٣- القراءة من الملف:

الصيغة العامة لامر القراءة من الملف هي:

متغير القراءة , رقم السجل , رقم الملف الاختياري Get

حيث ان:

رقم الملف الاختياري : هو الرقم الاختياري الذي تم افتراضه عند فتح الملف.

رقم السجل: هو رقم السجل (تسلسل السجل) الذي نرغب بقراءة محتوياته.

متغير القراءة: اسم المتغير الذي نريد قراءة السجل بداخله وهو الذي قمنا بتعريفه مسبقا بنفس

نوع السجل.

Example:

Dim stud as student
Get #1,11,stud

٤- تعديل محتويات سجل:

لتعديل محتويات أي سجل يجب اولاً ان نقرأ السجل ثم نعدل محتوياته او محتوى أي حقل من

حقوله ثم نعيد كتابته ونستخدم في الكتابه الامر put :

متغير الكتابة , رقم السجل , رقم الملف الاختياري Put

حيث ان:

رقم الملف الاختياري : هو الرقم الاختياري الذي تم افتراضه عند فتح الملف.

رقم السجل: هو رقم السجل (تسلسل السجل) الذي نرغب بتعديل محتوياته او محتوى احد

حقوله.

متغير الكتابة: اسم المتغير الذي نكتب محتوياته داخل السجل.

Example:

Dim stud as student
Stud.id=13862
Stud.name="ali"

Put #1,11,stud

٥-الإضافة للملف:

المقصود به اضافة سجل جديد الى نهاية الملف وهنا نستخدم الامر put ونعطي للسجل الجديد رقم يساوي عدد السجلات الموجودة + ١ ؛وبذلك فانه يضيف السجل الجديد في اخر الملف. نحتاج هنا لمعرفة عدد السجلات الموجودة في الملف حيث نستخدم القانون التالي:

عدد السجلات في الملف = طول الملف/طول السجل

حيث ان : طول الملف نحصل عليه باستخدام الدالة filelen() وطول السجل باستخدام الدالة len().

مثال: لاضافة سجل جديد للملف الذي رقمه الاختياري ١ (على افتراض ان الملف مفتوح) نبدأ بوضع المحتويات داخل المتغير stud ثم نحسب عدد السجلات ثم نضيف هذا السجل في اخر الملف:

```
Dim recordlen as integer
Dim stud as student
Stud.id=98501
Stud.name="mohammed"
Stud.address="Babylon city ,Hilla"
Stud.brithdat.day=4
Stud.brithdat.month=11
Stud.brithdat.year=1976
Recordlen=filelen(1)/len(student)
Put #1,recordlen+1,stud
```

٦-غلق الملف:

عند فتح أي ملف فيجب غلقه قبل انتهاء البرنامج، الصيغة العامة لامر غلق الملف هي:

Close#الرقم الاختياري

٨-٥-قواعد البيانات DataBases :-

بعد ان نكون قد فهمنا وبشكل جيد ماذا تعني قواعد البيانات وماذا تعني البيانات وماذا تعني كل من السجلات والجداول والحقول؛ بعد فهمنا لهذا كله ،فان معالج لغة VB يقدم لنا عدة طرق للتعامل مع قواعد البيانات هي:

○ الاداة OLE .

- الاداة Data – الاداة المهمة في التعامل مع قواعد البيانات.
 - الاداة FixdGrid – احدى ادوات الـ ActiveX .
 - القائمة Diagram :حيث ان جميع اوامر هذه القائمة تتفعل عندمل يتفاعل التطبيق مع SQL Server او مع النظام Oracle لانشاء او تعديل مخطط قاعدة البيانات.
 - القائمة Add-In :حيث ان اوامر هذه القائمة متعلقة بتقنيات وكائنات خارجية يتم دمجها مع بيئة التطوير .كما تحتوي على الامر Visual Data Manager الذي يتم من خلاله الوصول الى مدير البيانات Visdata .
- ٨-٦- طرق التعامل مع قواعد البيانات:-

هنا سوف نهتم بالتعامل مع قاعدة البيانات بطريقتين وكلاهما في الاخير يلتقيان باستخدام الاداة Data (نتعامل هنا فقط مع قواعد البيانات Access).

الطريقة الاولى:

نقوم بالدخول الى نظام ادارة قواعد البيانات Microsoft Access وانشاء قاعدة بيانات جديدة ،انشاء على الاقل جدول واحد في هذه القاعدة (يتم تحديد عدد الجداول حسب قاعدة البيانات التي نريد إنشاؤها) عند بناء الجدول يتم تحديد اسماء الحقول والانواع البيانية لكل حقل . ثم عند اكمال اسماء الحقول وانواعها البيانية نقوم بتسمية الجدول واغلاقه واغلاق قاعدة البيانات.

الطريقة الثانية:

انشاء قاعدة بيانات من داخل بيئة التطوير VB6 وذلك من خلال القائمة Add-In ثم اختيار الامر Vsual Data Manager سوف تفتح لنا نافذة في مجموعة قوائم ، نختار new من القائمة file ثم نختار Access سوف تفتح لنا نافذة اخرى عنوانها Database Window نضغط بمفتاح الفأرة الايمن بداخلها فتظهر لنا حافظة من اوامرها new table أي اضافة جدول جديد ، عند اختيار هذا الامر سوف تفتح لنا نافذة يتم من خلالها ادخال اسم للجدول وكذلك تحتوي على امر Add Field والمراد منه اضافة حقول للجدول الذي ادخل اسمه ،عند الضغط على هذا المفتاح سوف تفتح لنا نافذة تحتوي على عدة حقول يهمننا منها هو اسم الحقل ونوعه البياني ثم نضغط Ok واذا اردنا اضافة حقول اخرى للجدول نضغط على Add Field مرة اخرى ؛وهكذا بالنسبة لجميع الحقول المراد اضافتها للجدول ،ثم بعد اكمال اعداد جميع حقول الجدول نضغط على المفتاح Build The Table حيث ان هذا المفتاح سوف يقوم ببناء قاعدة

البيانات والجدول في ملف قواعد البيانات Access واذا اردنا اضافة جداول اخرى نضغط بمفتاح الفارة الايمن بداخل نافذة قاعدة البيانات ونختار الامر new table وهكذا بالنسبة لاضافة جداول اخرى لقاعدة البيانات .

يمكن ادخال بيانات لاي جدول في قاعدة البيانات وذلك بالنقر بمفتاح الفارة الايمن على اسم الجدول ثم نضغط على المفتاح Add فتظهر لنا نافذة ذلك الجدول تحتوي على اسماء حقول الجدول ، وبعد ادخال البيانات نضغط المفتاح Update لاضافة هذه البيانات الى قاعدة البيانات ثم نغلق جميع النوافذ؛ وبذلك انتهينا من الطريقة الثانية-بكل تأكيد ان الطريقة الاولى اسهل بكثير من الطريقة الثانية.

الى هنا ،تم اعداد قاعدة بيانات لكن لم يتم التعامل معها من خلال بيئة التطوير VB مباشرة ؛نقوم هنا باعداد واجهة المشروع واعداد خصائص الكيانات بالشكل التالي:

نقوم بوضع الاداة Data على النموذج ونقوم باعداد خصائصها اولاً ويجب ان يكون اولاً اعداد الخاصية Connect حيث من خلال هذه الخاصية يتم تحديد نوعية قاعدة البيانات التي سيتم الارتباط بها والتعامل معها،عند الضغط على هذه الخاصية يظهر لنا نوعية قواعد البيانات المتاحة لنا ،نختار Microsoft Access ثم نقوم باعداد الخاصية DatabaseName لتحديد اسم وامتداد قاعدة البيانات التي تم انشاؤها مسبقاً ثم نقوم باعداد الخاصية RecordSource وهنا تظهر لنا الجداول الموجودة في قاعدة البيانات التي تم تحديدها ضمن الخاصية DatabaseName .

الى هنا تم اهم خصائص الاداة Data وهذا يعني انه تم الارتباط بقاعدة بيانات (ان كل اداة Data واحدة يمكن من خلالها التعامل مع جدول واحد فقط) ؛نحتاج ايضا الى مربعات نص بقدر عدد الجدول الذي تم تحديده ضمن الخاصية RecordSource وبذلك يكون لكل حقل في الجدول مربع نص يتم من خلاله ادخال بيانات ذلك الحقل وكذلك عرض محتويات ذلك الحقل.

بالنسبة لمربعات النص فالخاصية text تبقى فارغة والخاصية DataSource يتم اعدادها باسم اداة البيانات Data ثم يتم اعداد الخاصية DataField حيث سوف تظهر قائمة تحتوي على اسماء حقول الجدول الذي تم تحديده ضمن الخاصية RecordSource في الاداة Data ، ثم نختار اسم حقل معين حيث سيكون مربع هذا خاص بذلك الحقل.

يتم التعامل مع الاداة Data من خلال الطرق التالية:

الطريقة	معناها
Recordset.addnew	اضافة سجل جديد
Updaterecord	حفظ السجل على اخر ادخال
Recordset.delete	حذف السجل الحالي
Recordset.movefirst	الانتقال لاول سجل(جدول) في قاعدة البيانات
Recordset.movelast	الانتقال لآخر سجل(جدول) في قاعدة البيانات
Recordset.moveprevious	الانتقال للسجل(جدول)السابق في قاعدة البيانات
Recordset.movenext	الانتقال للسجل(جدول)اللاحق في قاعدة البيانات
Recordset.bof	اختبار الوصول الى بداية السجلات
Recordset.eof	اختبار الوصول الى اخر السجلات
Recordset.findfirst-criteria	البحث عن اول سجل يحقق الشرط المحدد criteria
Recordset.findnext-criteria	البحث عن السجل التالي الذي يحقق الشرط المحدد criteria
Recordset.lastmodified	الوقوف على اخر تعديل للسجل

المصادر References

- ✓ مايكل هالفرسون ، ١٩٩٩ ، Visual Basic 6.0 خطوة خطوة، الطبعة العربية الاولى ، ترجمة :مركز التعريب والترجمة الدار العربية للعلوم،بيروت،لبنان.
- ✓ د.باسل الخطيب ، ٢٠٠١، تعلم فيجوال بيسك عن طريق الامثلة،دار الرضا للنشر ،سوريا،دمشق، الطبعة الاولى.
- ✓ م.عبد اللطيف عمرو -م.هاني درويش، ١٩٩٦، طريق المبرمج العربي الى احتراف البيسك المرئي،خوارزم للنشر والتوزيع،مصر،الاسكندرية.
- ✓ د.عبد المطلب ابراهيم احمد ، ٢٠٠٤ ، Visual Basic ،الجامعة التكنولوجية ، بغداد.
- ✓ ايفان كالاھان ، ٢٠٠٢ ، كيف تنشئ تطبيقات قواعد البيانات في Microsoft Access 2002 Visual Basic خطوة خطوة ، الطبعة العربية الاولى ، ترجمة:مركز التعريب والترجمة ،الدار العربية للعلوم ، بيروت ،لبنان.