وزارة التعليم العالي و البحث العلمي
هيئة التعليم التقني
الكلية التقنية / كركوك
قسم هندسة تقنيات الالكترونيك و السيطرة

Microsoft Visual & Algorithm Studio 6.0 Enterprise Edition

الخوارزميات و الفيجوال بيسك 6

إعداد
حسين نورالدين عزت
محمد محمود أحمد

2012-2013

الفهرست

المقدمة	5
الفصل الاول	
الخوارزميات والمخططات الانسيابية وأهميتها في البرمجة	6
مخططات سير العمليات أو(الانسيابية)	7
الرموز المستخدمة في تمثيل المخططات الانسيابية (سير العمليات)	8
أنواع مخططات سير العمليات (المخططات الانسيابية)	9
مفهوم العداد Counter	20
مفهوم المجاميع الإجمالية	22
صيغة التكرار باستعمال الشكل الاصطلاحي (الدوران)	26
تطبيقات عامة	32
الفرز (Sorting)	34
الفصل الثاني (الفيجوال بيسك 6)	
مقدمة	35
لغة Visual Basic	36
شرح واجهة الفيجوال بيسك	37
واجهة البرنامج	39
شرح صندوق الأدوات	41
شرح نافذة الخصائص	42
الأحداث Events	43
مجموعة من الأحداث	44
مصطلحات هامة	46
الثوابت Constants	46
انواع الثوابت	47
الإعلان عن الثوابت	48
المتغيرات VARIABLES	49
مدي استخدام المتغير وعمره	51
طرق تعريف المتغيرات في البرنامج	53
المتغيرات والمساحات التخزينية	54
بنية الفيجوال بيسك	57
استخدام عبارات التحكم في Visual Basic	57
العمليات المنطقية Boolean operations	60
المصفوفات	62
أنواع المصفوفات	63
المصفوفات الاحادية البعد	63
المصفوفات الثنائية البعد	64
المصفوفات متعددة الأبعاد	65
استخدام الـ Loops لمعالجة المصفوفات	66

مصفوفة التحكم (control array)	69
Module	70
الفورم	72
كيفية تغير حجم الفورم	74
التعامل مع النماذج (form)	75
خاصية موقع الفورم	77
تحريك الكائنات move	79
كيفية كتابة الشفرة	79
الالوان في الفيجوال بيسك	82
حفظ المشروع	83
التعامل مع زر الأمر Command Button	84
التعامل مع Label و Text	88
التعامل مع Text box	90
صندوق الصور Picture Box	93
صندوق الصورة Image Box	94
الاداة Check Box	95
الاداة Option	99
الاداة List Box	101
التعامل مع أشرطة التمرير	104
التعامل مع أدوات عرض الملفات	107
التعامل مع أداة التوقيت Timer	109
دوال الوقت و التاريخ	111
الاداة Calendar	113
صندوق الادخال Input box	114
أداة الأشكال(Shape)	115
رسم الاشكال الهندسية في البرنامج بواسطة الكود	116
دوال API	118
التعامل مع الحافظة	119
محرر القوائم	121
الرسائل Msgbox	127
صنايق الحوار Dialog Common	130
الدوال الجاهزة(Bulit-Function)	133
التعامل مع الأخطاء	139
تشغيل الملفات	140
الرياضيات في الفيجوال بيسك	144
العبارة الشرطية IF	150
العبارة Select Case	152
حلقة التكرار Do While	153
حلقة التكرار For Next	154

جمل التكرار المتداخلة (nested for)	155
انشاء قواعد البيانات بـ (Visual Data Manager)	156
تقنيات الوصول الى قواعد البيانات	158
طريقة إضافة وتبديل ال skin	161
عمل آلة حاسبة بسيطة	166
الفيجوال بيسك و دوره في التحكم	168
تجربة	169
المصادر	175

بسم الله الرحمن الرحيم

مقدمة

الحمد لله والصلاة والسلام على سيدنا محمد (صلى الله عليه وسلم)، وعلى آل بيته الطيبين الطاهرين وأصحابه اجمعين .

اللهم لا علم لنا إلا ما علمتنا إنك انت العزيز الحكيم .

ان الهدف من تأليف هذا الكتاب هو ايجاد مرجع شامل ومختصر للخوارزميات وللغة البرمجة المرئية (فيجوال بيسك) بأسلوب سهل ومختصر وبالتالي اتاحة الفرصة لجميع الدارسين للتعامل معها واتقانها .

يشمل هذا الكتاب فصلين تتضمن الفصل الأول الخوارزميات والفصل الثاني الفيجوال بيسك ، وتبحث كلا الفصلين في ثناياها مواضيع مختلفة من مواضيع البرمجة ، وقد تم تناول هذه المواضيع بشكل ميسر مع الرسوم التوضيحية يقرب لذهن القارئ فهم الموضوع واستيعابه بكل يسر .

ان ما يميز هذا الكتاب عن الكتب الاخرى هو طرحه لموضوع التحكم بواسطة الحاسوب على الآلات والمكائن بشكل عملي صوري مع الاشكال والتوضيحات مع التطبيقات العملية يستطيع من خلاله الدارسين تنفيذه على الحاسوب وهذا واضح في نهاية الفصل الثاني .

وفي الختام نسأل الله التوفيق والرضى ، آملين ان يكون هذا الكتاب نافعا ومفيدا ومرجعا مهما لكل المهتمين في هذا المجال ..

وفقنا الله واياكم للعمل الصالح اللهم آمين..

حسين نورالدين عزت / ماجستير علوم ادارة صناعية
محمد محمود أحمد / بكلوريوس هندسة الكترونيك و سيطرة
الكلية التقنية / كركوك
كركوك / العراق

الخوارزميات والمخططات الانسيابية وأهميتها في البرمجة

رغم أن الحاسب الكتروني يتميز بقدرته على انجاز العمليات الحسابية حسب الأوامر والتعليمات المعطاة له بسرعة فائقة وبدقة متناهية وكذلك بإمكانياته الكبيرة في حفظ المعلومات الواسعة والمختلفة التي يعجز الإنسان عن حفظها واستعادتها باستعمال ذاكرته العادية . فهو يعجز عن أن يقوم بشكل ذاتي بحل أية مسألة مهما كانت بسيطة ،أي ان عمله ينحصر في انجاز للمسائل التي تبرمج له بشكل صحيح بتوافق مع الأسس العلمية الصحيحة التي تعتمد عليها هذه الحلول.
وتصميم البرمجيات ينطلق من الخوارزميات *والمخططات الانسيابية حيث يشكلان بداية الطريق لإنشاء أي برنامج ، لذا سيتم توضيح هذه المفاهيم وكيفية التعامل معها .

خطوات حل مسألة باستخدام الحاسب:
يمكن أيجاز مراحل حل مسألة ما باستخدام الحاسب الالكتروني في خمسة مراحل :-

1- تعريف وتحليل المسألة:-
2- وضع الحل التخطيطي (سرد خطوات وبياتي)
3- كتابة الكود البرمجي
4- ترجمة البرنامج المصدري
5- تنفيذ البرنامج وتجربته

أولا: تعريف وتحليل مسألة :
إن تعريف المسألة هوعبارة عن دقة التعبير في تعريف المسألة بحيث يجعلها معروفة و مفهومة بصورة واضحة دون أي غموض لجميع الأشخاص العالمين ضمن مجال الاختصاص الذي تخضع له المسألة. أما تحليلها ووضع الحل فهو لب المسألة وقد نعاني صعوبات في ذلك حيث يمكن أن نستخدم كثير من القوانين والطرق الرياضية المناسبة للحل وقد نضطر لتطويرها لتتناسب مع المسألة المدروسة وفي هذه الخطوة يجب تحديد ما يلي :

1- عناصر المخرجات (النتائج المطلوبة التوصل إليها)وتحديد نوعها (بيانات حرفية أو رقمية أو حرفية رقمية)
2- عناصر المدخلات (المعطيات)وتشمل المتغيرات والمعلمات (parameters) ذات التأثير واللازمة للوصول إلى قيم عناصر المخرجات وكذلك يجب تحديد نوع المدخلات وتوصيفها بدقة .
3- طرق الحل المناسبة وتقييمها بما يتلاءم مع طريقة تنفيذها وفي ضوء ذلك نختار الأفضل .

ثانيا:-وضع الحل التخطيطي (سرد خطوات وبياني) :
نقوم هنا بالتعبير عن الحل أو الحلول التي استنتجت سابقا على شكل خطوات متسلسلة ومترابطة منطقيا ودقيقة الوصف للوصول إلى الحل وهي ما تدعى بالخوارزمية وقي وقد يكون في كثير من الأحيان وضع مخطط تسلسلي يسمى مخطط سير العمليات وذلك باستخدام مجموعة من الأشكال الرمزية الاصطلاحية التي تكون أكثر وضوحا وأنسبفهما .

ثالثا :-كتابة الكود البرمجي :
ليتمكن الحاسب من فهم هذا الحل يجب تحويله إلى لغة يفهمها لأن مخطط سير العمليات ما هو إلا عبارة عن تخطيط تصوري للمبرمج ولكنه غير مفهوم عند الحاسب الالكتروني .

رابعا:- ترجمة البرنامج المصدري :
يتم ذلك بإدخال البرنامج إلى الحاسب وترجمته إلى لغة الآلة(الثنائية) بوساطة برنامج الترجمة أو التجميع (CompilerorAssembler) وذلك في حال عد وجود أخطاء في البرنامج المصدر وسمى البرنامج الجديد المترجم البرنامجالهدفي (Object Program) وتمر عملية الترجمة بالمراحل التالية :

A - مرحلة التحليل المعجمي (Lexical Analysis):
وفيها تتم مطابقة كلمات البرنامج المصدري والعلاقات والأسماء مع تلك الممسوح بها في اللغة واكتشاف الأخطاء فيها ان وجدت .

B - مرحلة التحليل اللغوي والنحوي (Syntax Analysis).

تتم فيها مطابقة تعليمات برنامج المصدر مع القواعد اللغوية للغة المستخدمة ،واكتشاف أي الأخطاء فيها فضلا عن عملية تحويل تعليمات البرنامج المصدري إلى تعليمات وأوامر رمزية بلغة التجميع

C - مرحلة ترجمة البرنامج إلى لغة الآلة : (Machine Language)
وتتم فيها تحويل البرنامج المصدر إلى برنامج بلغة التجميع ومنها نحصل على البرنامج ألهدفي الذي نستطيع تنفيذه.

خامسا: تنفيذ وتجربة البرنامج:
يتم تجربة البرنامج ألهدفي على عينة من المعطيات أو بيانات (Data) للتأكد من صحته منطقيا، فإذا ما ثبت صحة طريقةالحل المقترحة يمكننا تنفيذ البرنامج على المعطيات الحقيقية وطباعة النتائج المطلوبة.

مفهوم الخوارزمية أو اللوغاريتم

كلمة الخوارزمية (Algorithm) مشتقة نسبة إلى العالم العربي المشهور الخوارزمي الذي قام بوضع أسس المسائل بشكل تتابعي . والخوارزمية هي عبارة عن مجموعة من الخطوات المتسلسلة التي تصف بصورة مضبوطة وبدون أي غموض جميع الخطوات الرياضية والمنطقية اللازمة لحل مسألة ما.وقد تطور هذا المفهوم وأصبح يعني طريقة أو خطة شاملة وعامة لحل مسألة ما، وتقوم بوصف كافة الخطوات بشكل مفصل.
ونقول عن طريقة حل مسألة بأنها خوارزمية إذا اتصفت بما يلي :-

1- تحقق إمكانية وصف الطريقة المتبعة في حل المسألة بعدد من الخطوات التي تحوي تعاليم أو أوامر بشكل مفصل وصريح توضح فيها ما يجب عمله في كل خطوة بدون التباس فيه.
2- تحوي عل عدد محدد من الخطوات توصلنا إلى النتائج انطلاقا من المعطيات المتوفرة ، فوجود عدد نهائي من الحلول لطريقة ما ، لا يمكننا من اعتبارها خوارزمية حل .
3- يجب أن تتضمن الخوارزمية جميع الشروط والاعتبارات في حل المسألة مهما اختلفت المعطيات التي تتناولها المسألة المطروحة .

المتحولات (المتغيرات)

المتحولات (Variables) عبارة عن مقادير تأخذ قيما مختلفة وتخزن قيم المتحول في خلية من ذاكرة الحاسب وهذا يعني أننا عندما نسميه متحولا أو متغيرا فإن الحاسب يخصص له خلية من ذاكرته معرفة باسمه وان أية قيمة يأخذها هذا المتحول أو المتغير تخزن في خلية الذاكرة المعنونة باسمه وتزول القيمة السابقة في الخلية عند إسناد قيمة جديدة لها ونستطيع استخدام هذه القيمة بذكر اسم المتحول أو المتغير ويمكننا تسمية هذه المتحولات أو المتغيرات بأي اسم نريده مثلا حرف (X,A,B,....) أو أكثر من حرف أو حتى كلمة والأفضل أن تحمل معنى مثل (VALUE ,SUM)
وتقسم هذه المتحولات أو المتغيرات إلى ثلاثة أنواع :-

1- المتحولات العددية
2- المتحولات الحرفية
3- المتحولات المنطقية

تأخذ المتحولات العددية قيما رقمية ويمكننا جراء العمليات الحسابية من جمع وطرح وغيره عليها أما الحرفية فلا نستطيع إجراء العمليات الحسابية عليها أنما هي اسما أو معطيات تصنيفية وحتى لو أشرنا أليها بأرقام فهي لا تخضع للعمليات الحسابية ، وبالنسبة للمتحولات المنطقية فهي تأخذ قيمتين صح (TRUE) وخطأ (FALSE) ونحتاجها في الاختيارات والاختبارات وتطبق عليها عمليات الجبر البولي (AND ,OR ,NOT) وقد تعرضنا لفكرة عن المتحول كونه سيمر استخدامها في الخوارزميات والمخطط التدفقية وهي لا تخضع لقواعد هنا بعكس ما نجد في متحولات لغات البرمجة التي تخضع لقواعد المبرمجة المتعامل معها .

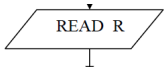
مخططات سير العمليات أو(الانسيابية)

جاءت مخططات سير العمليات (Flowcharts) كضرورة لتسهيل عمل المبرمج عندما تتعقد الخوارزمية خاصة أي تزداد خطواتها والمقارنات فيها فمخططات سير العمليات ما هي إلا تمثيل رسومي أو وصفي لخطوات حل المسألة (لخوارزمية) بشكل أكثر وضوحا وأنسب فهما ،باستخدام مجموعة من رموز الأشكال الاصطلاحية (الهندسية).

أهمية خرائط سير العمليات :-

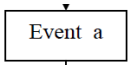
1- تساعد المبرمج على الإلمام الكامل بالمسألة المراد حلها،والسيطرة على كل أجزائها بحيث تساعده على اكتشاف الأخطاء المنطقية (Logical errors) والتي غالبا ما تجهد المبرمج .
2- في المسائل ذات الأوضاع المختلفة، والتي تكثر فيها التفرعات يصعب غلى المبرمج تذكرها ومتابعتها أثناء كتابة المبرمج بدون الاعتماد على خرائط سير العمليات.
3- تساعد المبرمج على تعديل البرامج الموضوعة بمجرد النظر مما يزيد في فاعلية هذه البرامج .
4- يعتبر الاحتفاظ بخرائط سير العمليات لحل مسألة (مسائل) معينة مرجعا هاما يمكن الرجوع إليها عند إجراء تعديلات عليها أو استخدامها لحل مسائل أخرى مشابهة دون التقيد بوجود المبرمج الذي بحل المسألة باعتبار أن الحل / الحلول الأولى قد تمت صياغتها في خطوات واضحة وبسيطة ومفهومة يسهل متابعتها وفهمها من قبل مبرمج آخر .

الرموز المستخدمة في تمثيل المخططات الانسيابية (سير العمليات)؟

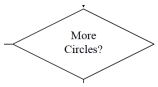


فيما يلي أهم الرموز المستخدمة في تمثيل مخططات سير العمليات مع بيان توضيحها وحسب ما هو معتمد في المعهد الوطني الأمريكي (AINSI) :

الإطار المستطيل المنهى بنصفي دائرة :-



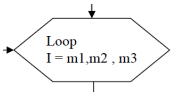
يستخدم هذا ألإطار للدلالة على نقطة بداية المخطط الانسيابي أو نهايتها فنضع فيه إما كلمة البداية أو (START) في أول المخطط وكلمة النهاية أو توقف (STOP/END)عند نهاية المخطط ويجب أن يحوي المخطط عل إطار واحد للبداية ولكنه قد يحوي عل أكثر من إطار توقف في عدة أماكن من المخطط.



1- الإطار المتوازي الأضلاع :-

يستخدم للدلالة على قراءة المعطيات وطباعتها أي لعمليات الإدخال والإخراج (Input /Output) ومن خلاله نعين أسماء وعناوين للخلايا التي تخزن فيها المعطيات عند تنفيذ الخوارزمية وكذلك الخلايا المطلوب طباعتها فمثلا (READ x , y) ويعني أنه عند تنفيذ الأمر يجب قراءة متحولتين (اسم أو عدد) وتخزينهما في خليتين معونتين ب (x, y).

2- الإطار المستطيل :-



يستخدم لتحديد العمليات الحسابية وبيان الخلايا التي تخزن فيها نتائج العمليات الحسابية وهي ما تدعى بالأوامر أو التعاليم الحسابية في المخطط الانسيابي فمثلا (x = y + 10) ويعني إضافة محتوى الخلية المعونةy إلى العدد (10) وتخزين الناتج في الخلية المعونة(x).

3- الإطار المعين : -

يستخدم لوضع التساؤلات والاختبارات في المخطط الانسيابي فعندما نكتب ضمن إطار المعين (x =y) فهذا يعني هل محتوى (x)يساوي محتوى(y) ، ويتفرع من إطار المعين مسارات قد تكون أثنا ن أو ثلاثة حسب حالة الاختبار ففي حالة (x= y) يتفرع مسارين عن إطار المعين لأن جواب السؤال سيكون إما نعم أو لا وثلاثة مسارات في حال كان السؤال(x<=y) فكل مسار يمثل حالة .

ملاحظة :- إن وجود التعبير (x =y) في إطار مستطيل يعني حفظ أوخزن محتوى الخلية المعنونة (y) في الخلية المعنونة (x) بينما وجوده في الإطار المعين يعني هل(x) تساوي (y) .

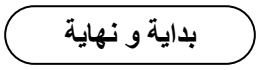
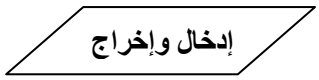
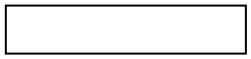
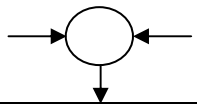
4- الإطار المستطيل المنتهي بنصفي معين :-

يستخدم للتعبير عن حالات التكرار والحلقات مثلا تكرار إعطاء قيمة لعداد(I)من (1) حتى(n) وتكتب بالشكل(I =1, nI)

5- شكل الدائرة: -

تستخدم لبيان وصل منطقة من المخطط الانسيابي مع منطقة ثانية ويوضع داخله رقم أو حرف ونفس الرقم أو الحرف في المنطقة الأخرى المراد القفز إليهاو الشكل (1) يبين الرموز والإشكال الاصطلاحية العامة المستخدمة في رسم المخططات الانسيابية.

الرموز و الوظائف

الشكل الاصطلاحي	معنى الرمز
	يستخدم هذا الرمز للدلالة على بداية البرنامج ونهايته
	يمثل هذا الرمز كل من عمليتي الإدخال (قراءة البيانات تمهيدا لمعالجتها) والإخراج (عرض النتائج على الشاشة، طباعة، ...)
	رمز المعالجة، وقد يحتوي هذا الرمز على عملية حسابية أو عملية تخزين
	رمز اتخاذ قرار، ويستخدم هذا الرمز في خطوات المعالجة التي تتطلب إجراء عملية منطقية كالمقارنة أو عملية اختبار. ونتيجة لعملية الاختبار أو المقارنة، سيتم إتباع أحد المسارين.
	اتجاه سير البرنامج. تستخدم الأسهم لبيان حركة واتجاه خريطة التدفق.
	نقطة توصيل وربط (Connector).

الشكل (1)

ملاحظة :- يمكن كتابة الخوارزمية باللغة العربية أو باللغة الانكليزية .

أنواع مخططات سير العمليات (المخططات الانسيابية)

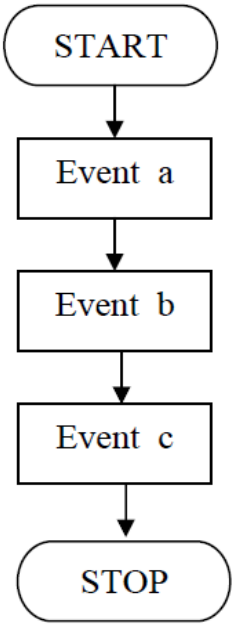
يوجد أربعة أنواع رئيسة للمخططات الانسيابية :-

1- مخططات التتابع البسيط (Simple sequential Flowcharts).
2- مخططات التفرع (Branched Flowcharts).
3- مخططات التكرار البسيط (Loop Flowcharts).
4- مخططات التكرار المتداخلة (Nested –loop- Flowcharts)

ويمكن للبرنامج الواحد أن يشتمل على أكثر من نوع واحد من هذه الأنواع ، وسنتناول فيما يأتي شرح هذه الأنواع بشيء من التفصيل .

1- مخططات التتابع البسيط :-

يخلو هذا النوع من التفرعات (Branches) والتكرارات (Loops)أو أي قرارات (Decisions) والشكل العام لهذا النوع موضحة في الشكل (2) :-



شكل (2)

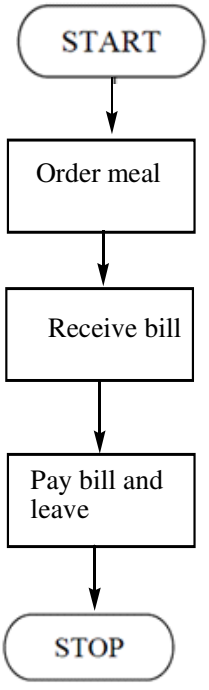
و كلمة (Event) تعني الحدث أو العملية المطلوب تنفيذها

مثال (1):-

اكتب وارسم خوارزمية لتناول وجبة طعام في مطعم ثم أرسم المخطط الانسيابي لهذه المسألة .

الخوارزمية :

1- أبدأ
2- أطلب الوجبة
3- استلم الفاتورة
4- ادفع الفاتورة وغادر
5- توقف



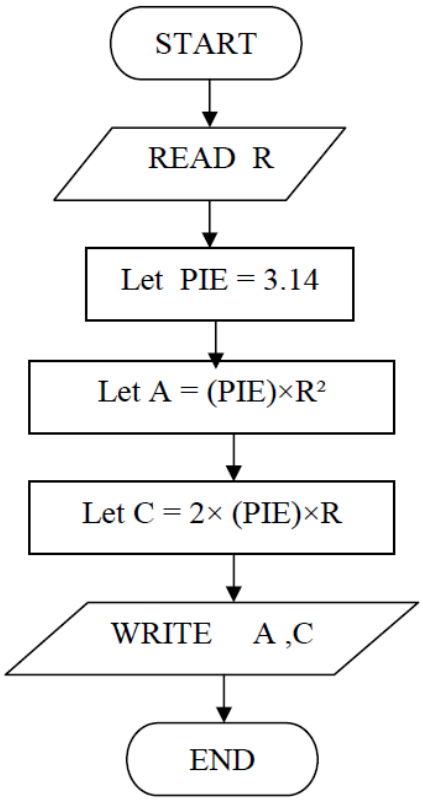
الشكل (3)

مثال (2):-

أكتب وارسم خوارزمية التي تمكننا من حساب محيط ومساحة دائرة نصف قطرها (R) وارسم المخطط الانسيابي لهذه المسألة.

الخوارزمية :

1- أبدأ
2- اجعل قيمة PIE = 3.14
3- احسب المساحة (A) من المعادلة $A=PIE * R^2$
4- أحسب المحيط (C) من المعادلة C=2*PIE*R
5- أطبع قيم كلا من المساحة والمحيط
6- توقف



شكل(4)

مثال (3):-

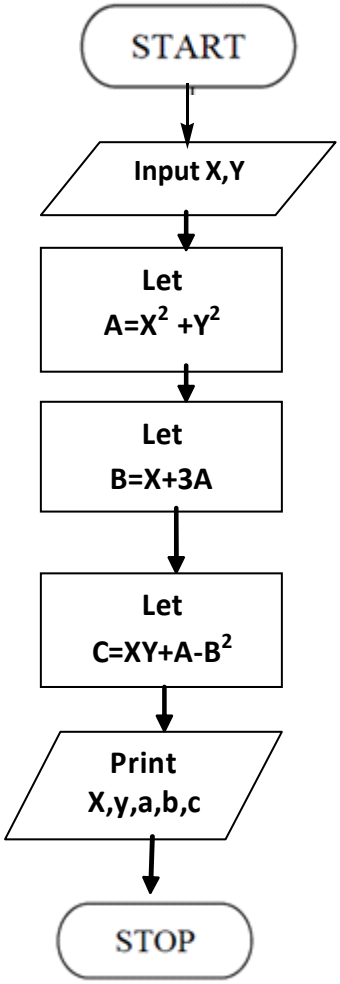
ارسم خريطة سير العمليات لحساب قيمة كل من المتغيرات C,A,B في المعادلة الآتية :

$A=X^2 + Y^2$ .....(1)

B= X + 5A.....(2)

$C =XY +A -B^2$....(3)

حيث أن X,Y قسم معلومة



الشكل (5)

الحل :-

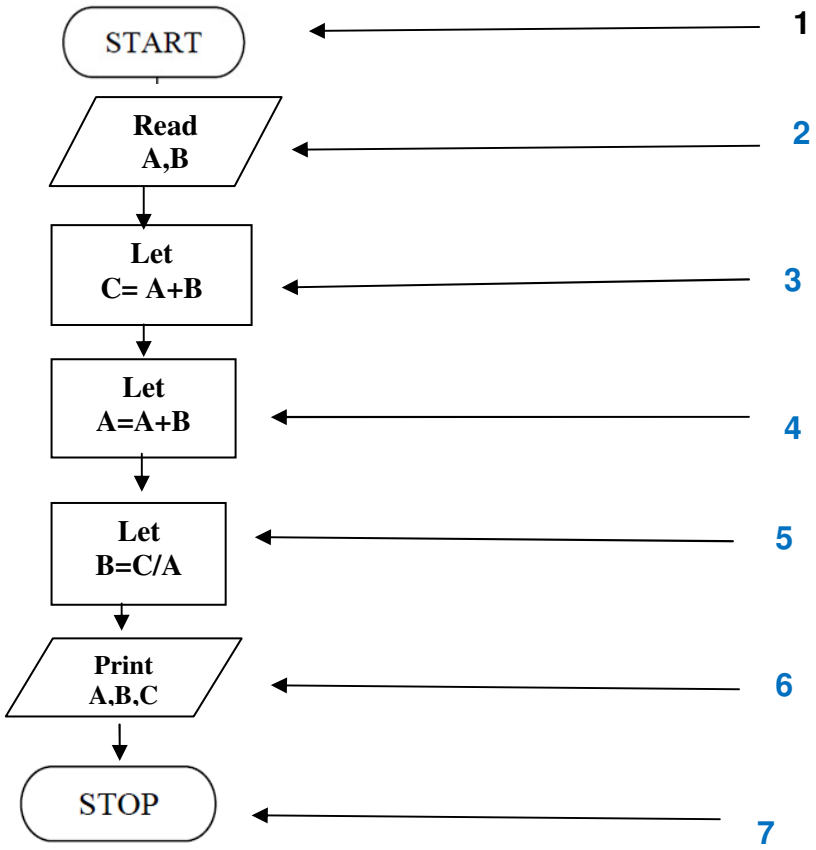
المدخلات في المسألة هي (X,Y) والمطلوب حساب قيمة المتغير (A) في المعادلة (1)، وحساب قيمة المتغير (B)في المعادلة(2) بالاعتماد على قيمة (X) المعلومة لدينا وقيمة المتغير (A) المحسوبة في الخطوة السابقة ،أما قيمة المتغير (C) فيتم حسابها في المعادلة (3) بالاعتماد على قيم كل من المتغيرات B.A,Y,X وكلها معلومة .
وتكون خطوات حل المسألة كما هو مبين في الشكل (5)

1- ابدأ
2- اقرأ كل من قيم كل من (X, Y)
3- احسب قيمة A من المعادلة (1)
4- احسب قيمة B من المعادلة (2)
5- احسب قيمة C من المعادلة (3)
6- أطبع قيم كل من C,B,A,Y,X
7- توقف.

مثال (4) :-

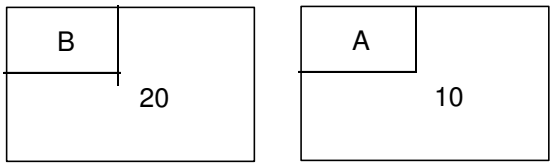
الشكل (6) يمثل مخططا انسيابيا لمجموعة من العمليات الحسابية، أدرس ناتج العمليات بعد تتبع المخطط ، إذا علمت أن :

A=10 , B=20

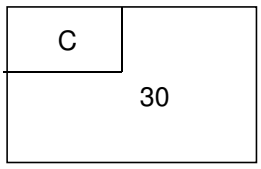


الشكل (6)

عند تتبع المخطط الانسيابي في الشكل (6) نلاحظ ان الخطوة الأولى تمثل قراءة قيم المتغيرات (A,B) وبالتالي تخصص القيم في ذاكرة الحاسب على النحو التالي :

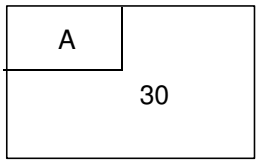


أما الخطوة الثانية فهي معالجة وتتخلص بحساب قيمة المتغير (C) وفقا للمعادلة



C=A+B
= 10 +20
=30

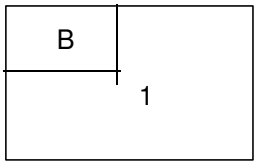
أما الخطوة الثالثة فهي أيضا معالجة وتتخلص بحساب قيمة (A) وفقا للمعادلة



A = A+ B
= 10 +20
= 30

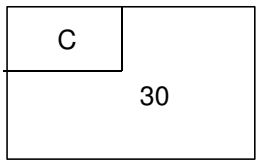
نلاحظ في هذه الخطوة أن قيمة المتغير (A) الجديدة هي قيمة المتغير (A) القديمة مضافا إليها قيمة المتغير (B) وبالتالي فإن قيمة المخزن (A) تصبح تساوي (30) بدلا من (10) وذلك بإلغاء القيمة السابقة واستبدالها بالقيمة الجديدة فقط.

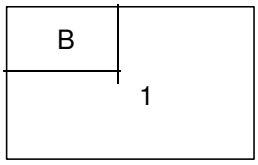
أما الخطوة الرابعة فهي تعني تغير قيمة المتغير (B) لتصبح:

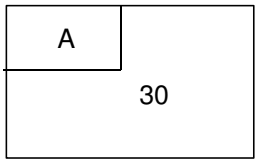


B = C /A
= 30 30
= 1

لا حظ أننا أخذنا آخر قيمة للمتغير A لحساب قيمة B ن وبالتالي فإن القيمة النهائية للمتغيرات (C,A,B) تكون على النحو التالي :







أما الخطوة الخامسة فهي طباعة قيم المتغيرات (C,B,A)

A= 30 B=1 C=30

أما الخطوة السادسة فهي توقف .

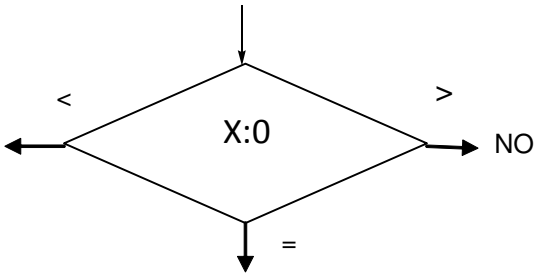
ويمكن تلخيص خطوات تنفيذ المخطط الانسيابي في المثال أعلاه بالجدول التالي:

رقم الخطوة	A	B	C	المخرجات
1	10	20		
2	10	20	30	
3	30	20	30	
4	30	1	30	
5				30 1 30
6	STOP			

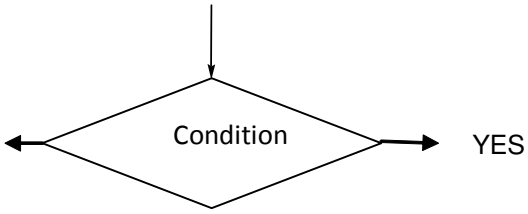
ونلاحظ أن كل خطوة تمثل وضع الحاسب في تلك اللحظة.

2- مخططات التفرع:-

هذا النوع من المخططات الانسيابية يتضمن اتخاذ قرار أ و مفاضلة بين اختيارين أو أكثر ، وهناك أسلوبان في تنفيذ القرار

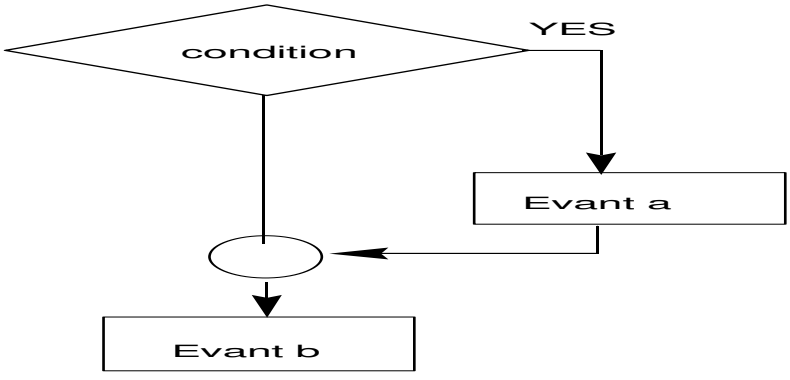


(قرار ذو ثلاثة تفرعات)



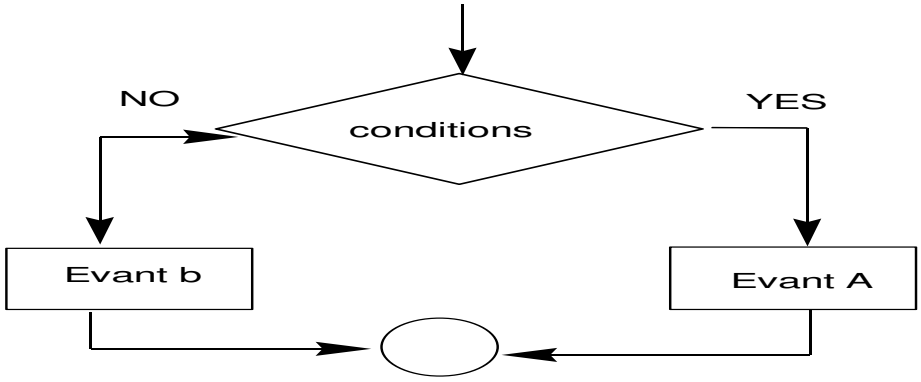
(قرار ذو تفرعين)

وبشكل عام فإن مخططات التفرع يمكن أن تأخذ إحدى الصورتين الآتيتين :



شكل (7)

اذا كان جواب الشرط (yes)فان الحدث التالي في التنفيذ هو (a) ثم يتبعه الحدث (b) أما إذا كان (No) فان الحدث التالي في التنفيذ هو (b)



شكل (8)

اذا كان جواب الشرط (Yes)فان الحدث التالي في التنفيذ هو الحدث (a) أم إذا كان الجواب (No) فان الحدث التالي في التنفيذ هو (b)

مثال (5):-

أكتب وارسم المخطط الانسيابي لإيجاد قيمة اقتران المطلقة المعرفة حسب القاعدة التالية :

$$F(x) = |x| = \begin{cases} x & ,x \geq 0 \\ -x & ,x < 0 \end{cases}$$

1. ابدأ
2. اقرأ قيمة المتغير(x)
3. إذا كانت x أكبر أو تساوي صفرا أذهب إلى الخطوة (4) وإلا فاذهب إلى الخطوة (5)
4. احسب قيمة الاقتران F(x) من العلاقة F(x) =x ثم اذهب الى الخطوة (5) .
5. احسب قيمة الاقتران من العاقة F(x) = -x
6. أطبعقيم كل من (x) وF(x)
7. توقف

مثال(6):-

ارسم خريطة سير العمليات لحساب قيمة الاقتران f(x,y) المعرف حسب القاعدة :

$$F(X,Y) = \begin{cases} X + Y & X > Y \\ X^2 + 5 & X = Y \\ Y + 3X & X < Y \end{cases}$$

إذا علمت أن قيم كل من (X) و (Y) معلومة

الحل :

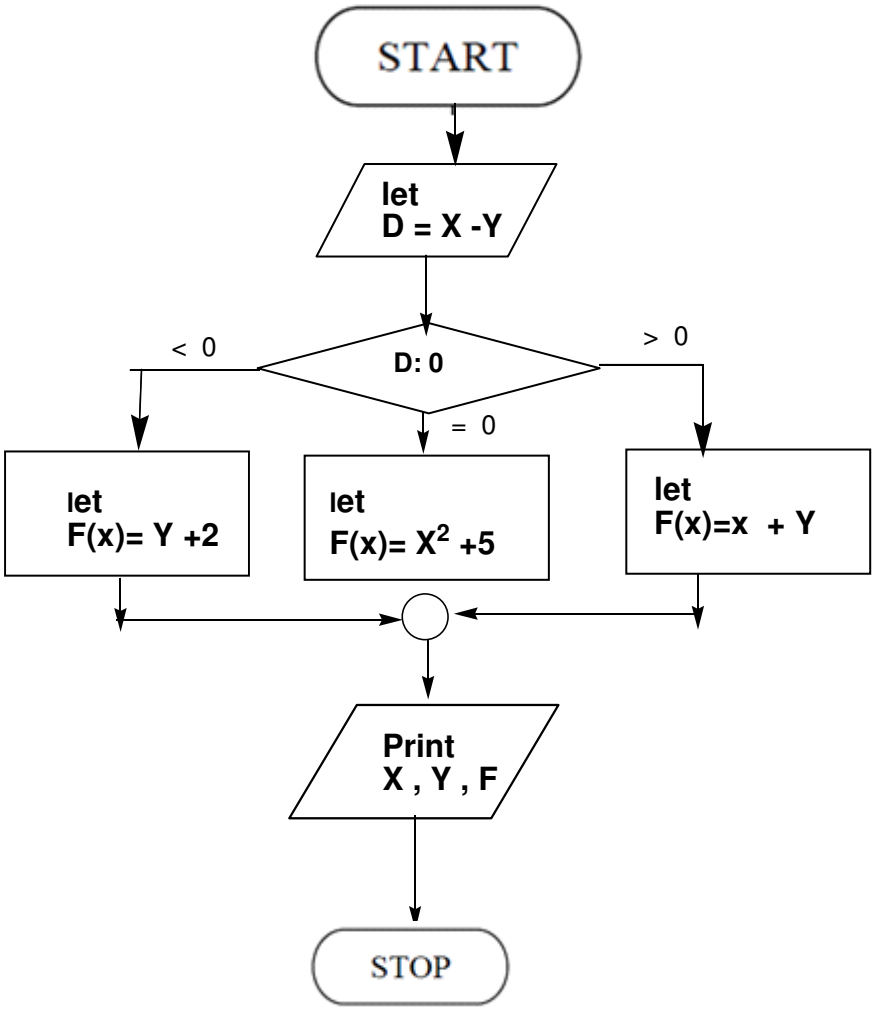
يمكن اعادة تعريف الاقتران على النحو التالي :

$$F(X,Y) = \begin{cases} X + Y & X - Y > 0 \\ X^2 + 5 & X - Y = 0 \\ Y + 3X & X - Y < 0 \end{cases}$$

وبالتالي تكون خطوات الحل :

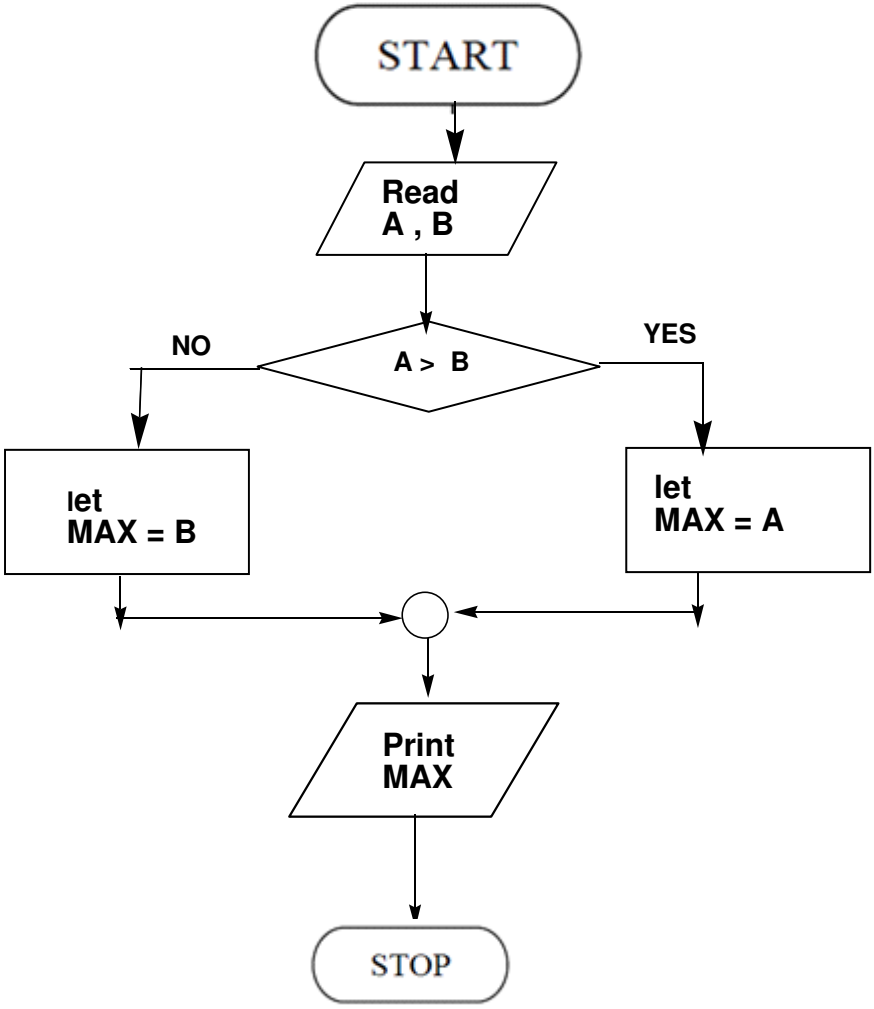
1- أبدأ
2- ادخل قيم كل من (X , Y)
3- اجعل قيمة (D= X –Y)
4- اذا كانت قيمة(D) أكبر من صفر فاذهب إلى الخطوة (5)
واذا كانت قيمة (D) تساوي صفرا فاذهب الى الخطوة (6)
أما اذا كانت (D) أقل من صفر فاذهب الى الخطوة (7)
5- احسب قيمة الاقتران F(X , Y) من المعادلة (1) ثم اذهب الى الخطوة (8)
6- احسب قيمة الاقتران F(X , Y) من المعادلة (2) ثم اذهب الى الخطوة (8)
7- احسب قيمة الاقتران F(X , Y) من المعادلة (3) ثم اذهب الى الخطوة (8)

8- اطبع قيم كل من(F , X)
9- توقف



شكل (9)

مثال (7):-
أرسم خريطة سير العمليات لاختيار القيمة العظمى لقيمتين معلومتين
الحل :-
المدخلات :A , B
المخرجات : MAX
العلاقات : Max = MAX[A,B]
وعليه تكون خطوات الحل كالآتي :-
1- أبدأ
2- اقرأ قيم كل من (A,B)
3- اذا كانت (A>B) اذهب إلى خطوة (4) وإلا فاذهب الى الخطوة (5)
4- اجعل (MAX =A() ثم اذهب الى الخطوة (6)
5- اجعل (MAX =B)
6- اطبع (MAX)
7- توقف



شكل (10)

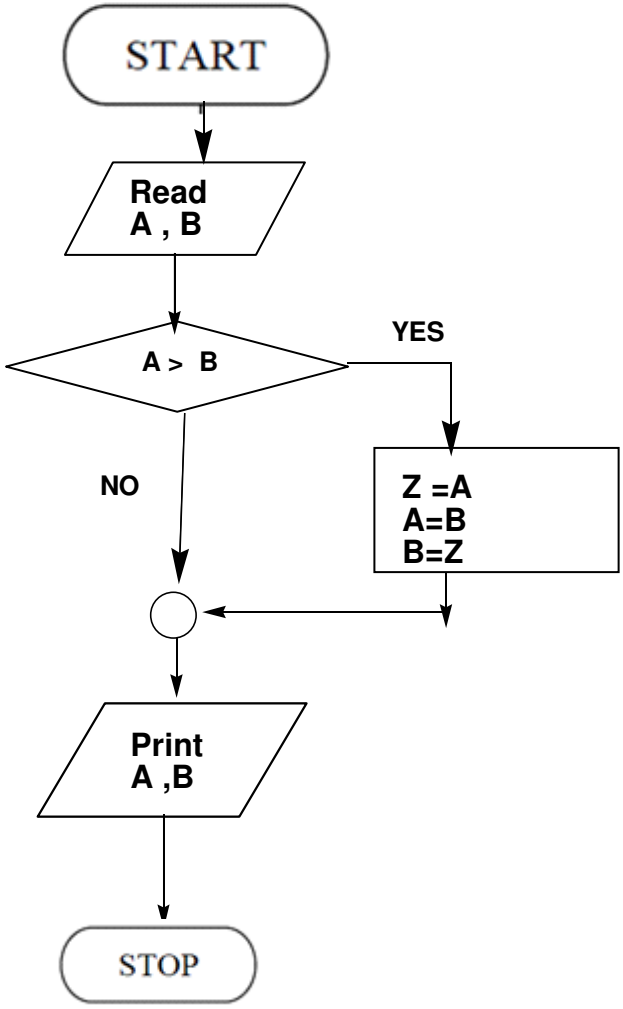
مثال(8) :-

اكتب خريطة سير العمليات لإعادة ترتيب قيميتين ترتيبا تصاعديا

الحل :-

تكون خطوات الحل كالآتي :

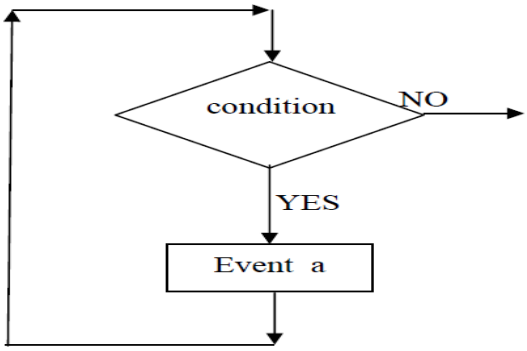
1- ابدأ
2- اقرأ القيميتين (A,B)
3- اذا كانت (A , B) اذهب إلى الخطوة (4) وإلا فاذهب إلى خطوة(5)
4- استبدل قيميتي (A ,B)
5-اطبع (A , B)
6-توقف



شكل (11)

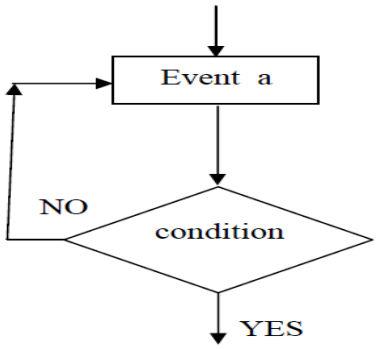
3- خرائط التكرار البسيط :Loop Flowchart-

هذا النوع من الخرائط نحتاج اليها لإعادة تنفيذ مجموعة من الجمل في البرنامج عددا محدودا أو غير محدود من المرات والشكل العام لمثل هذه الخرائط كما يلي :



يتكرر تنفيذ الحدث a عدداً من المرات طالما كان جواب الشرط نعم

DO WHILE



يتكرر تنفيذ الحدث a إلى أن يصبح جواب الشرط نعم

DO UNTIL

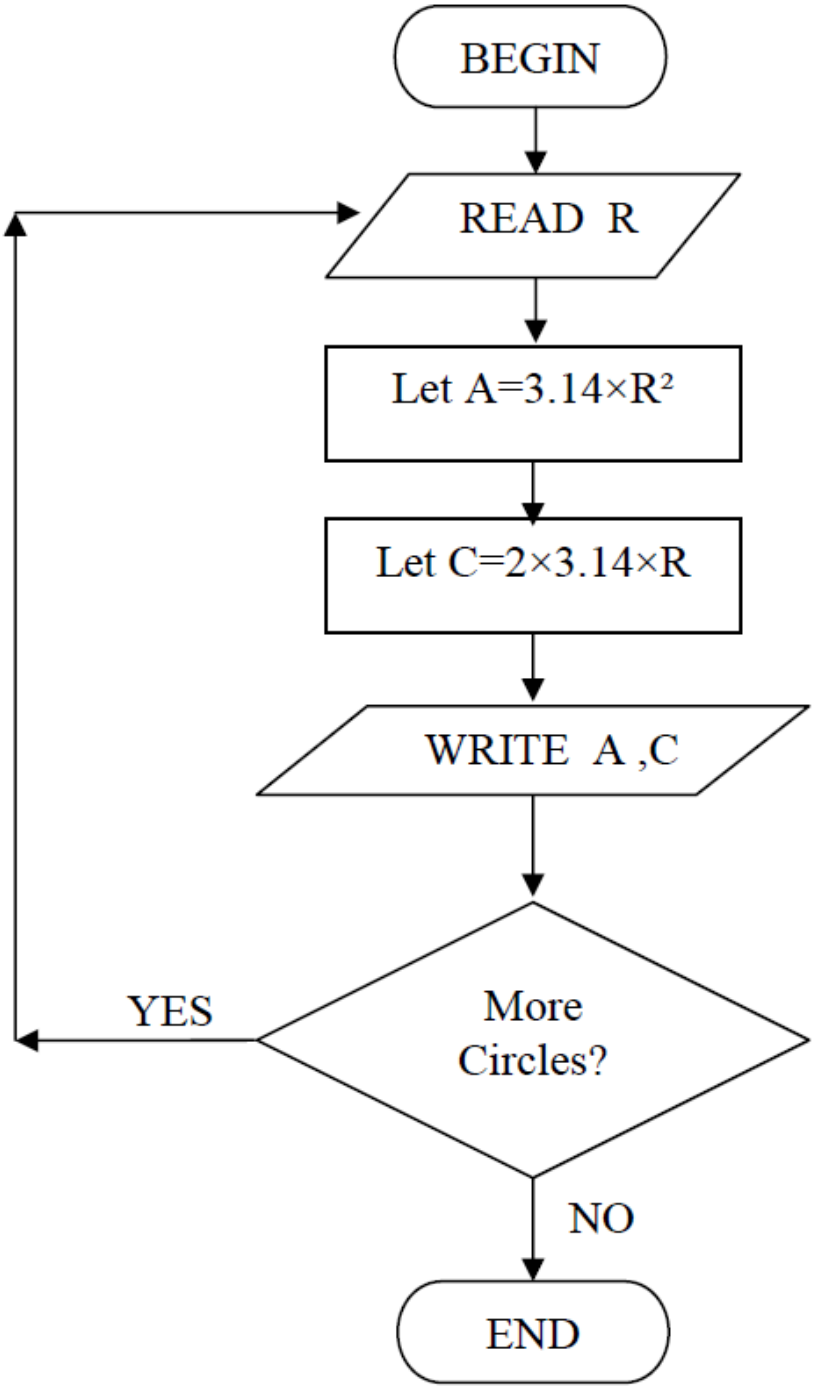
مثال (9):-

ارسم خريطة سير العمليات لإيجاد مساحة ومحيط مجموعة من الدوائر أنصاف أقطارها معلومة :

الحل :-

تكون خطوات الحل كما يلي :

1- أبدأ
2- اقرأ نصف قطر الدائرة (R)
3- أوجد مساحة الدائرة (A) ومحيط الدائرة (C)
4- أطبع قيم كل من C , A, R
5- هل هناك مزيد من الدوائر ؟
فان كان الجواب نعم فعد إلى الخطوة (2)
وإن كان ل فآهب إلى الخطوة (6)
6- توقف.



شكل (12)

مفهوم العداد Counter

في كثير من الأحيان نحتاج في برامج الحاسب الالكتروني إلى العد(counting) فقد نريد مثلا أن نعد عدد الطلاب والطالبات ضمن الشعبة ، وقد تكون هذه العملية سهلة للإنسان لأنها أصبحت ضمن قدراته العقلية التي يكتسبها من الطفولة إلا آت الحاسب يحتاج إلى تصميم خوارزمية العد (Counting Algorithm) تتضمن خطوات معينة إذا اتبعها استطاع أن يعد .

ويمكن تحدي الخطوات التي يتبعها الحاسب حتى يتمكن من العد في الخطوات الأساسية الآتية :

1- اجعل العداد مساويا للصفر
2- اجعل القيمة المضافة الجديدة للعداد تساوي القيمة العددية له زائد واحد ، أي أن :
قيمة العداد (الجديدة) = قيمة العداد (القديمة) + 1
3- كرر الخطوات ابتداء من الخطوة 2

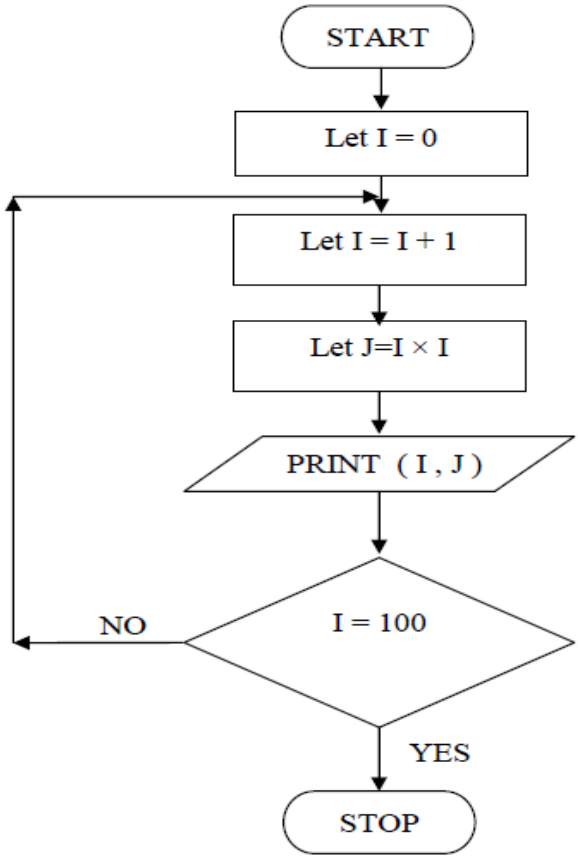
مثال (10):-

ارسم الخوارزمية التي يتبعها الحاسب لطباعة الأعداد الطبيعية من 1 إلى 100 ومربعاتها

الحل :-

1- ابدأ
2- اجعل(I=0)
3- اجعل (I = I + 1)
4- اجعل ($J = I^2$)
5- أطبع(J,I)
6- إذا كانت(I= 100) أذهب إلى الخطوة (7) وإلا فاذهب إلى الخطوة (3)
7- توقف

1- START
2- Let I = 0
3- Let I = I + 1
4- Let J = I×I
5- PRINT (I , J)
6- If I = 100 Goto (7) Else Goto (3)
7- STOP



شكل (13)

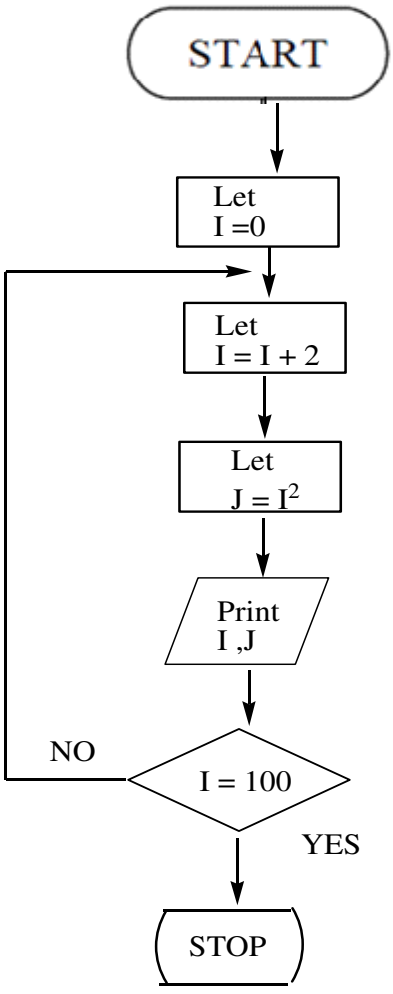
مثال(11) :-

ارسم واكتب خوارزمية التي يتبعها الحاسب الالكتروني لطباعة الإعداد الزوجية من (1) إلى (100)ومربعاتها .

الحل :-

1- ابدأ
2- اجعل (I=0)
3- اجعل(I = I + 2)
4- اجعل($J = I^2$)
5- أطبع (J,I)
6- إذا كانت(I= 100) أذهب إلى الخطوة (7) وإلا فاذهب إلى الخطوة (3)
7- توقف

1- START
2- Let I = 0
3- Let I = I + 2
4- Let J = I×I
5- PRINT (I , J)
6- If I = 100 Goto (7) Else Goto (3)
7-STOP



شكل (14)

ملاحظة:- لاحظ أن القيمة الجديدة للعداد = القيمة القديمة +2 أي أن الزيادة الدورية في كل تكرار تساوي 2.

مفهوم المجاميع الإجمالية

نحتاج في برامج الحاسب في كثير من الأحيان إلى جمع مجموعة كبيرة من الأعداد التي تمثل ظاهرة معينة ، فمثلا عندما نريد أن نحسب معدل علامات طالب وكذلك الأمر في هذه الحالة يجب علينا ان نرشد الحاسب للقيام بعملية الجمع ويمكننا ذلك باستخدام متغيرين اثنين أحدهما المتغير الذي نجمعه والآخر هو الجمع الاجمالي (المجمع) ويم ذلك وفق الخطوات التالية :

1- اجعل المجمع مساويا للصفر.
2- أدخل قيمة واحدة للمتغير
3- اجعل القيمة الجديدة للمجمع تساوي القيمة القديمة له زائد القيمة المدخلة للمتغير أي أن :
قيمة المجمع الجديدة = قيمة المجمع القديمة + آخر قيمة مدخلة للمتغير.
4- كرر الخطوات ابتداء من الخطوة الثانية

مثال(12) :-

ارسم خريطة سير العمليات لإيجاد الوسط الحسابي لأعمار طلاب شعبتك .

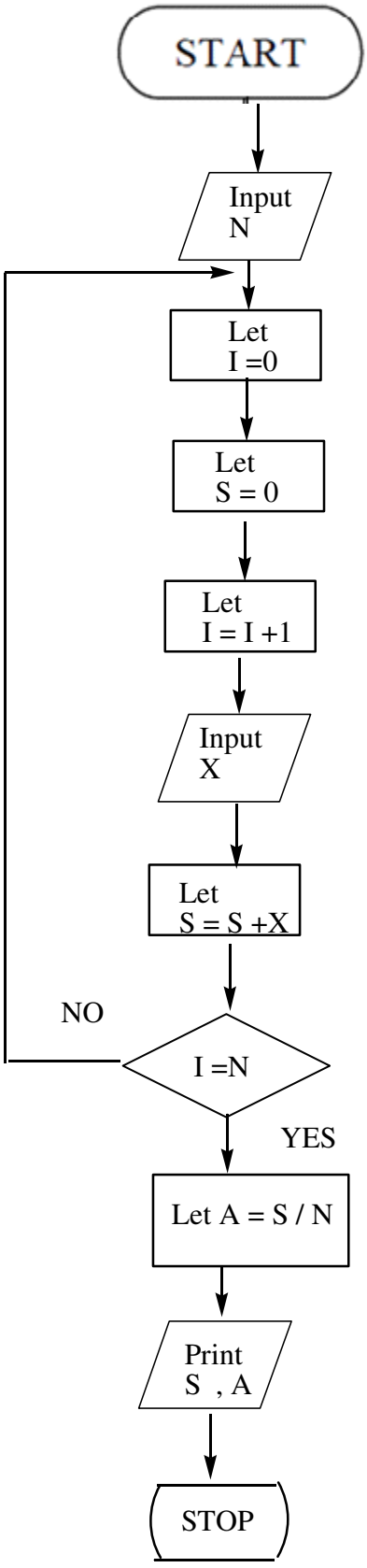
الحل :-

المتغيرات :

N: يمثل عدد الطلاب
I: عداد
X: عمر الطلاب
S: مجموع اعمار الطلاب
A: معدل اعمار الطلاب

وتكون خطوات الحل كما هو مبين في الشكل (15) كما يأتي :

1- أبدأ
2- ادخل اجمالي عدد الطلاب (N)
3- اجعل (I=0)
4- اجعل(S=0)
5- ادخل (I=I+1)
6- ادخل (X)
7- اجعل (S=S+X)
8- إذا كانت (I=N)اذهب إلى الخطوة (9) وإلا عد إلى الخطوة (5)
9- اجعل(A=S/N)
10-اطبع (S,A)
11-توقف


START
Input
N
Let
I =0
Let
S = 0
Let
I = I +1
Input
X
Let
S = S +X
NO
I =N
YES
Let A = S / N
Print
S , A
STOP

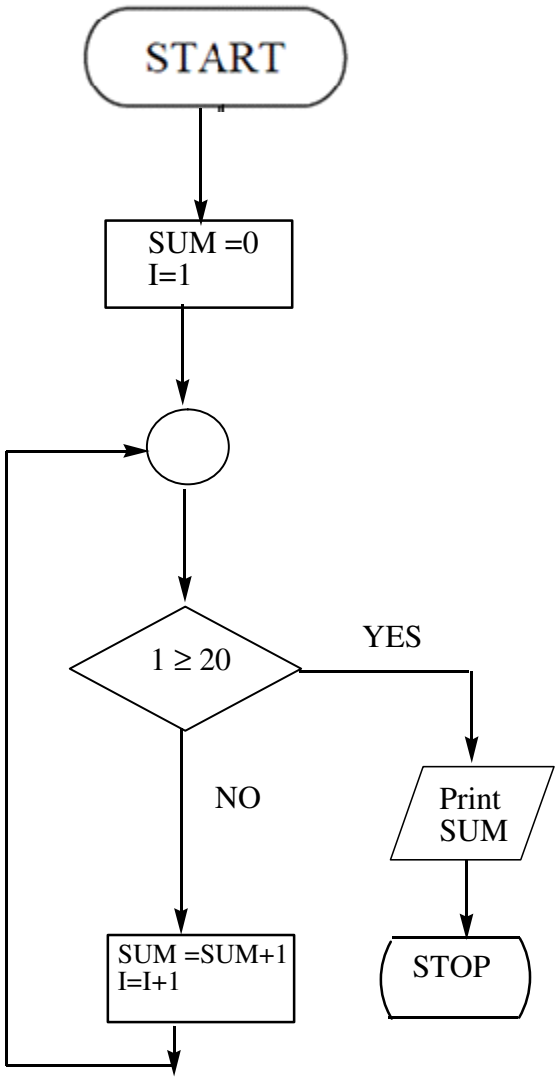
شكل (15)

مثال (13):-

ارسم خريطة سير العمليات لا يجاد مجموع الاعداد من(1)إلى(20)

الحل : -

1- ابدأ
2- ضع (SUM =0)
3- إذاكان جواب الشرط (1:≥ 20) (YES) اذهب الى الخطوة (5)
4- اجعل : (SUM =SUM +1)
(I= I + 1)
ثم اذهب الى الخطوة (3)
5- اكتب المجموع SUM
6- توقف



شكل (16)

نلاحظ أن قيمة العداد تبدأ بقيمة ابتدائية مقدارها(1) ثم يتبع ذلك سؤال هل قيمة العداد تزيد القيمة العظمى للعداد (20)فإذا كان الجواب (NO) كرر تنفيذ مجموعة جمل التكرار وهي :

(SUM = SUM +1)

ثم اضف الزيادة الدورية لقيمة العداد واذا كان الجواب (YES) تابع تنفيذ التعليمات الواقعة خارج جمل التكرار .

مثال(14) :-

ارسم خريطة سير العمليات لحساب قيمة الاقتران

$(F(X)=X^2+1)$لمجموعة من قيم (X) الموجبة .

الحل : -

خطوات الحل تكون كالآتي :

1- ابدأ
2- اقرأ قيمة(X)
3- اذا كانت قيمة(X>0)إذهب إلى الخطوة (4) وإلا إذهب إلى الخطوة (6) مباشرة
4- احسب قيمة الاقتران F(X) من القاعدة :

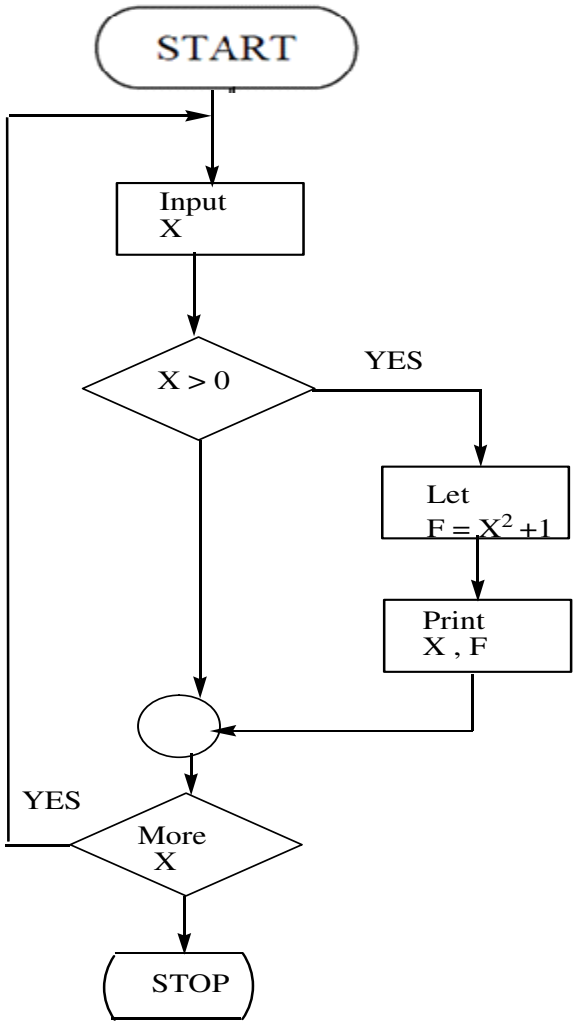
$F(X)=X^2+1$

5- اطبع قيم كل من (X , Y)
6- هل هناك قيمة أخرى ل(x) ؟

اذا كان الجواب نعم عد إلى الخطوة (2)

واذا كان الجواب لا اذهب إلى الخطوة (7)

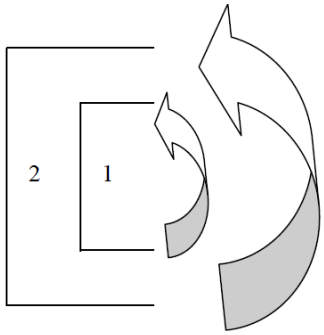
7- توقف



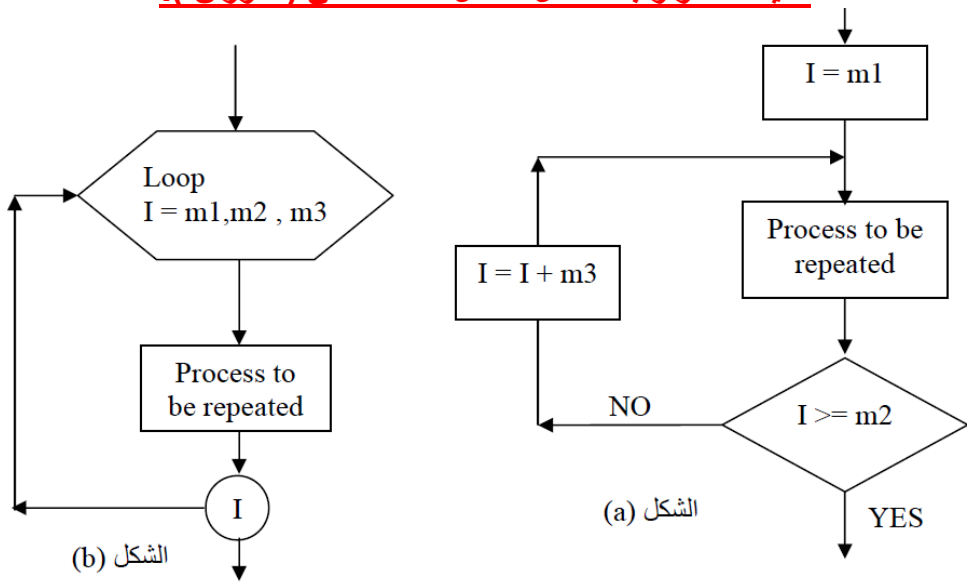
شكل (17)

4- خرائط التكرار المتداخلة (Nested –loop flowcharts)

تكون التكرارات متداخلة تماما بحيث لا تتقاطع فإذا كان لدينا تكرارين من هذا النوع يسمى التكرار (1) تكرار داخلي بينما التكرار (2) تكرار خارجي ويتم التنسيق في عمل هذين التكرارين بحيث تكون أولوية التنفيذ للتكرار الداخلي كما هو الشكل التالي :



صيغة التكرار باستعمال الشكل الاصطلاحي (الدوران):



شكل (18)

نلاحظ في الشكل (a) أنه لتحقيق التكرار نحتاج لما يلي :

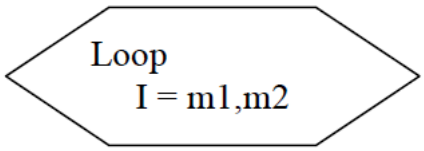
1- العداد (I) متغير التكرار.
2- القيمة الأولية للعداد وتساوي m1.
3- القيمة النهائية للعداد وتساوي m2.
4- الزيادة الدورية (الزيادة عند نهاية كل تكرار) وتساوي m3.

وتكون آلية عمل هذه العناصر كما يحددها المبرمج بما يلي :

1- اجعل العداد (I) يبدأ بقيمة أولية مقدارها m1.
2- أتم الاجراءات المطلوب اعادته .
3- إذا كانت قيمة العداد (I) وصلت إلى القيمة النهائية m2اذهب إلى الخطوة التالية في البرنامج وإلا فاذهب إلى الخطوة (4) .
4- زد العداد(I) بمقدار الزيادة الدورية m3.
5- عد إلى الخطوة (2).

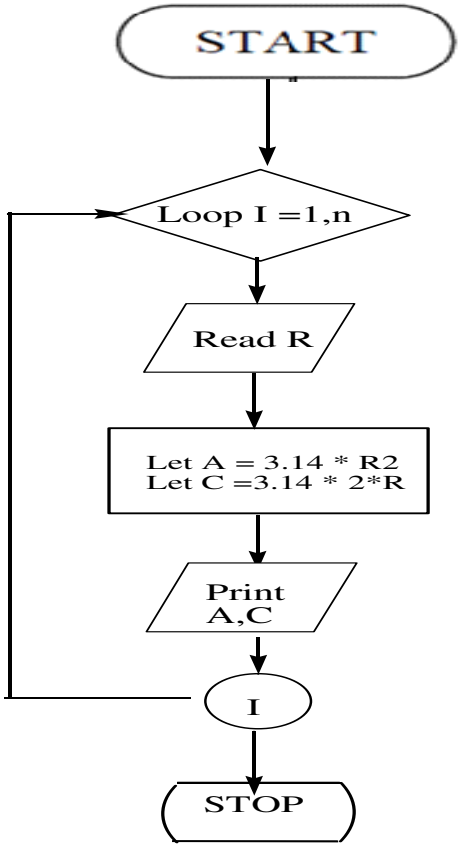
يمكننا استبدال الخطوات (1-3-4-5) في الشكل (a) بخطوة واحدة مبينة في الشكل (b) حيث ينفذها الحاسب بشكل آلي مما يؤدي إلى تسهيل عملية البرمجة واختصار عدد التعليمات وتجنب الأخطاء .

نشير إلى أن قيمة m3 تساوي (1) دائما مالم تعط قيمة اخرى غير ذلك وفي حال عدم ذكر قيمة m3تكون قيمتها مساوية (1) ضمنا وتمثل كما يلي



مثال(15) :-

أعد رسم المخطط التدفقي لإيجاد مساحة ومحيط n من الدوائر باستخدام جملة الدوران المتداخلة .



شكل (19)

مثال (16):-

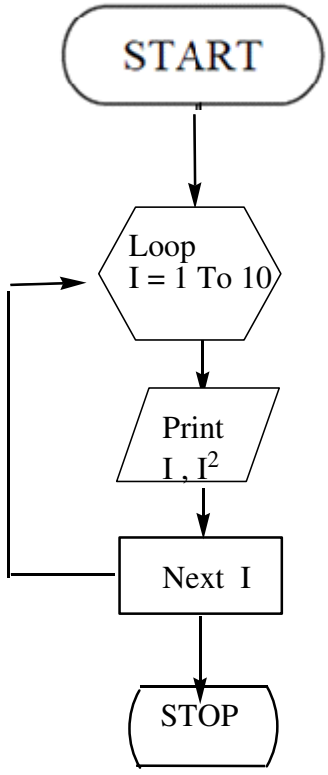
ارسم خوارزمية لطباعة الاعداد من 1 إلى 10 ومربعاتها :

الحل : -

القيمة الابتدائية =1

القيمة النهائية = 10

الزيادة الدورية =1



شكل (20)

مثال (17):-

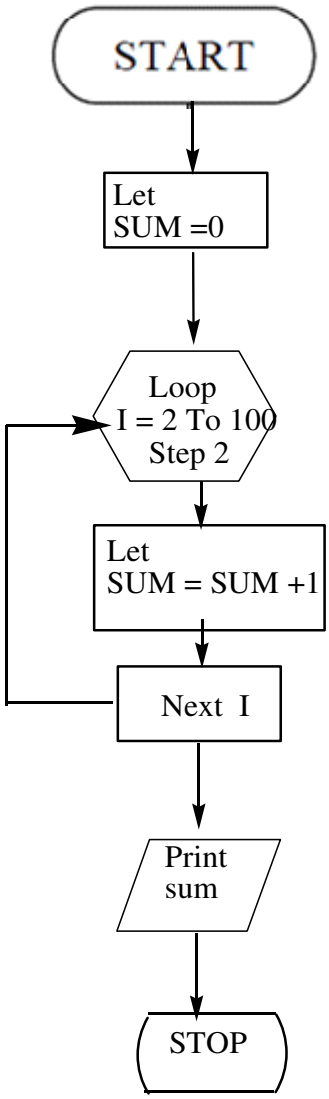
أرسم خريطة سير العمليات لإيجاد مجموع الأعداد الزوجية من(2)إلى(100).

الحل :-

القيمة الابتدائية 2

القيمة النهائية 100

الزيادة الدورية 2



شكل (21)

مثال(18) :-

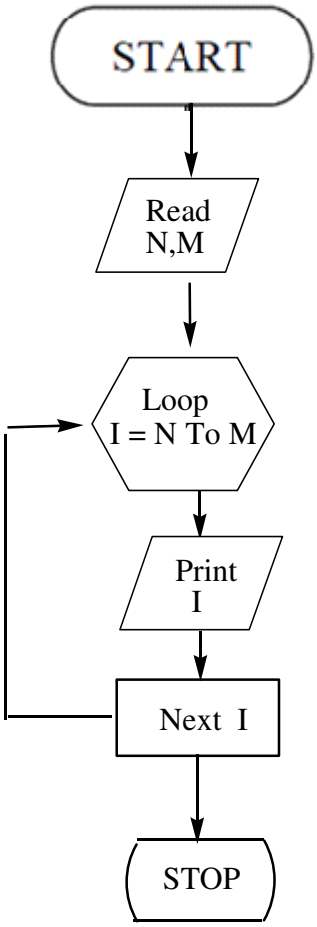
ارسم المخطط التدفقي لقراءة عددين وطباعة الاعداد الواقعة بينهما على الترتيب.

الحل :-

القيمة الابتدائية = قيمة العدد الاول

القيمة النهائية = قيمة العدد الثاني

الزيادة الدورية = 1



شكل (22)

مثال (19):-

ارسم خريطة سير العمليات لإيجاد قيمة الاقتران

$F(x) = ax^2 + bx + 5$

لجميع قيم (x) من 1 إلى 15 وبزيادة دورية مقدارها 0.5.

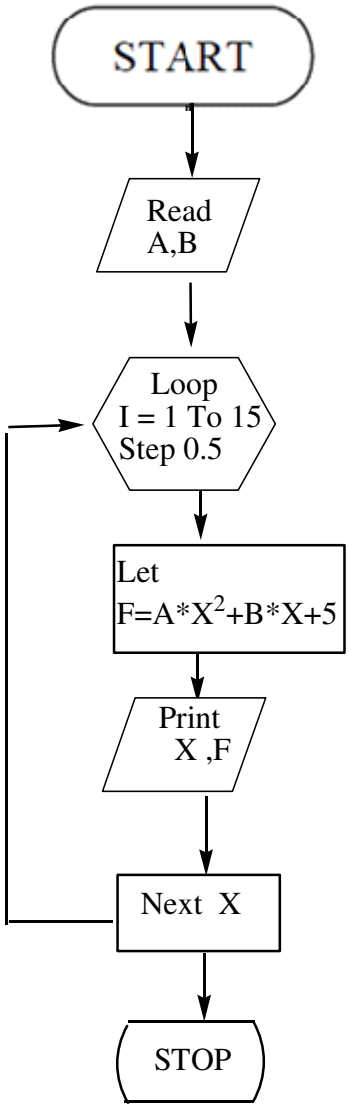
الحل :-

متغير جملة التكرار هو x

القيمة الابتدائية تساوي 1

القيمة النهائية تساوي 15

الزيادة الدورية تساوي 0.5



شكل (23)

مثال(20) :-

ارسم خريطة سير العمليات للإيجاد مجموع حدود السلسلة التالية :-

1+1/2+1/3+1/4+...+1/N

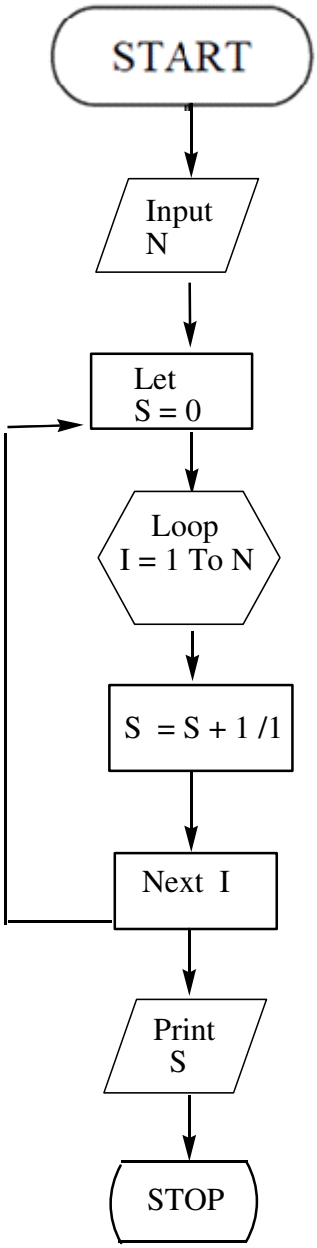
الحل :-

متغير جملة التكرار (I=)

القيمة الابتدائية ل (I = 1)

القيمة النهائية ل (I =N)

الزيادة الدورية (= 1)



شكل (24)

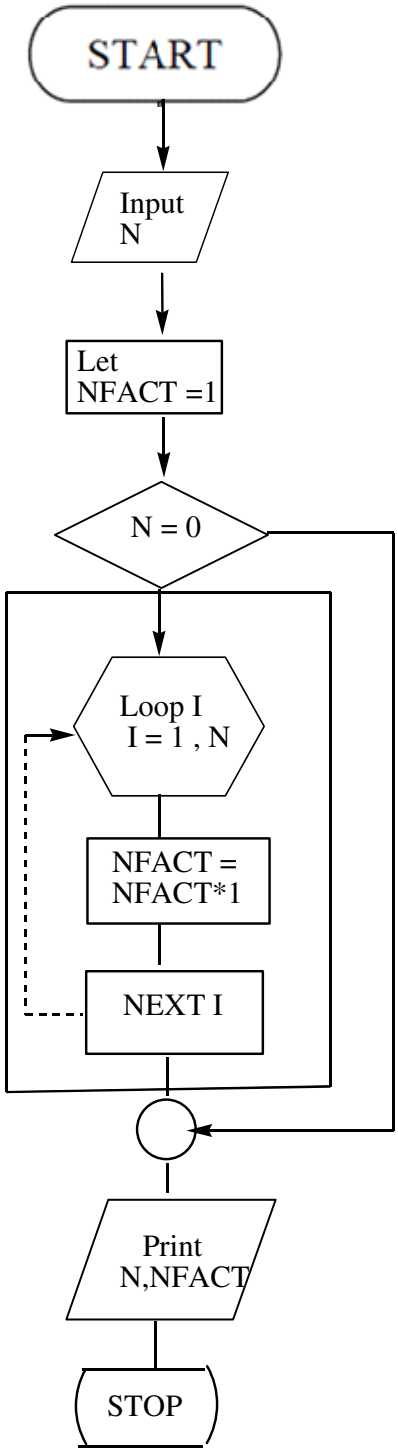
مثال(21) :-

ارسم خريطة سير العمليات لإيجاد **N!**

الحل :-

$$N! = \begin{cases} 1 & ; N = 0 \\ N((N-1)(N-2)...3*2*1 & ; N > 0 \end{cases}$$

فخطوات الحل كما هو مبينة في الشكل التالي :-



شكل (25)

تطبيقات عامة

البحث عن قيمة معينة (SEARCHING)

مثال(22) :-

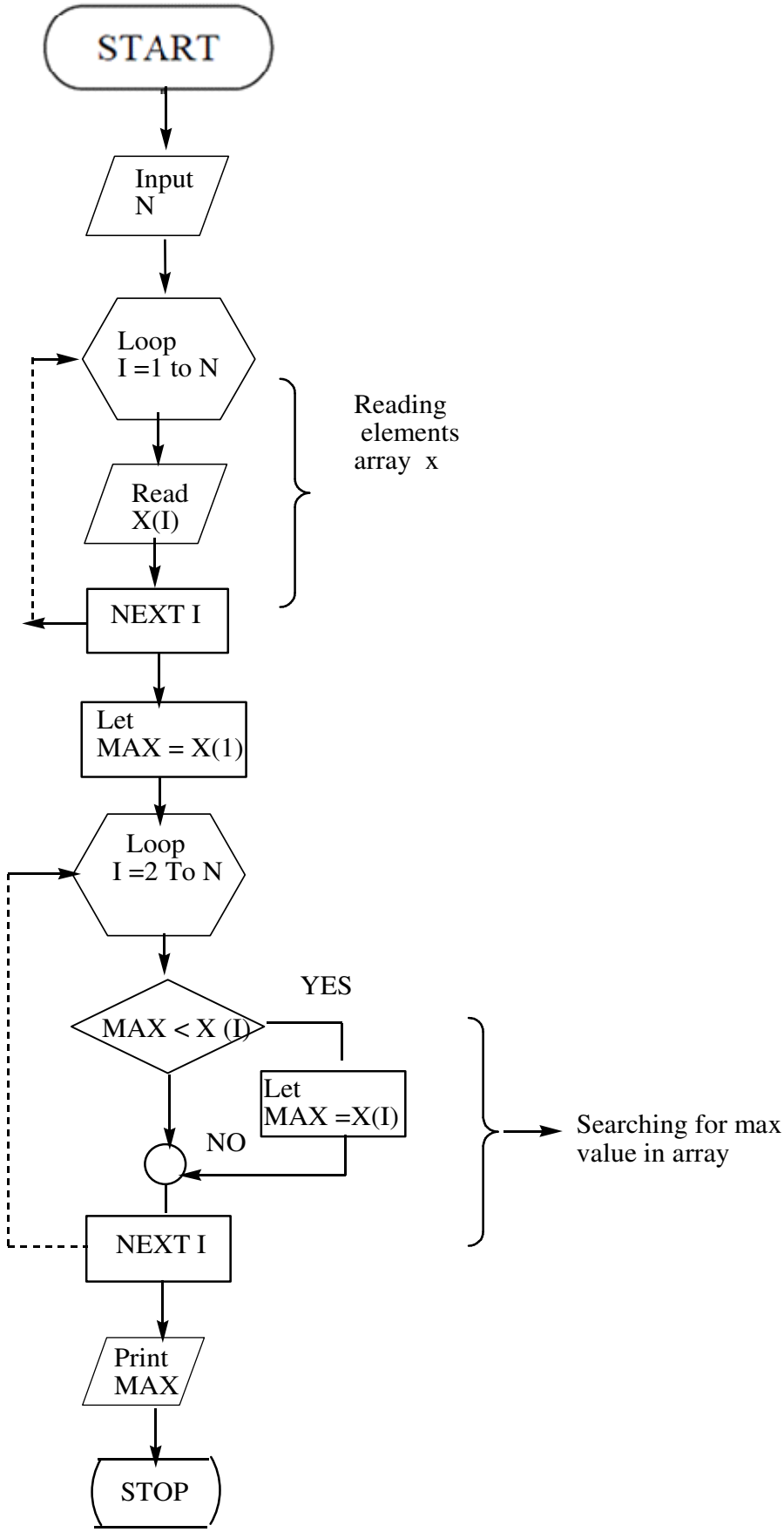
ارسم خريطة سير العمليات لإيجاد القيمة العظمى لعدد معين من القيم :

الحل : -

نفترض ان القيم هي : $X_1 , X_2 , \ldots, X_n$

المطلوب إيجاد قيمة MAX بحيث :-

$$MAX = \max\{ xi\}$$
$$I = 1 ,n$$


START
Input
N
Loop
I =1 to N
Read
X(I)
NEXT I
Reading
elements
array x
Let
MAX = X(1)
Loop
I =2 To N
YES
MAX < X (I)
Let
MAX =X(I)
NO
NEXT I
Searching for max
value in array
Print
MAX
STOP

شكل (26)

الفرز (Sorting)

مثال (23):-

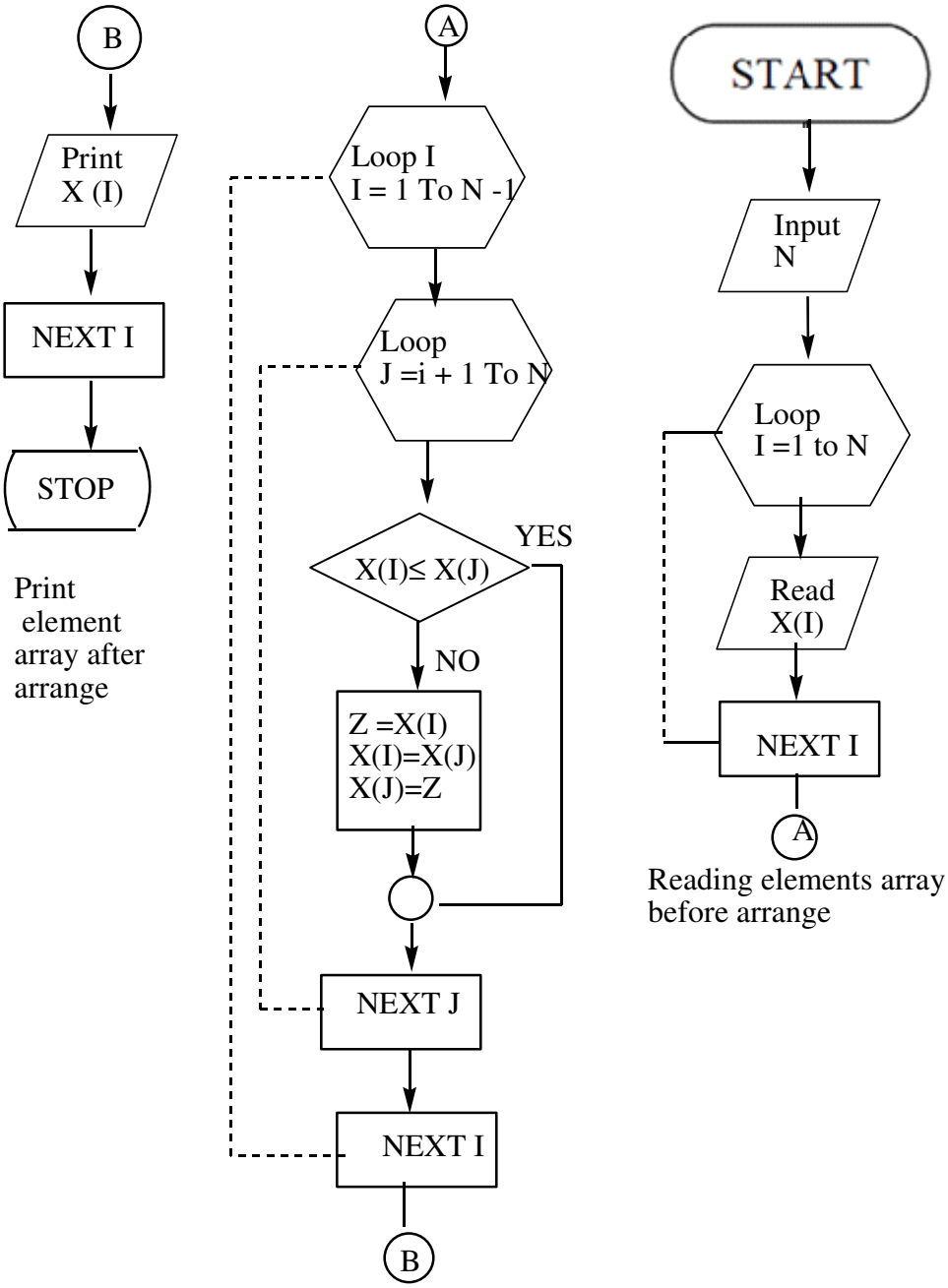
ارسم المخطط الانسيابي لترتيب عناصر المصفوفة أحادية البعد ترتيبا تصاعديا .

الحل :-

عناصر المصفوفة (X)هي :(X_1 , X_2 , X_n)

المطلوب ترتيب القيم ترتيبا تصاعديا بحيث :-

$(X_1 < X_2 < X_3 < ....<X_n)$



شكل (27)

الفصل الثاني (الفيجوال بيسك 6)

مقدمة

برنامج الكمبيوتر (Program) :- عبارة عن سلسلة من الأوامر أو التعليمات (Instructions) المخزنة في ملف أو أكثر ليتم تنفيذها عند الحاجة، وأغلب البرامج تأخذ ما يسمى بالمدخلات Input وتقوم بإعطاء المخرجات Output.

البرمجة (Programming) :- هي عملية كتابة وتأليف البرامج.

المبرمج (Programmer) :- هو الفني أو المهندس المختص الذي يتولى كتابة وصيانة البرامج.

تقنيات البرمجة

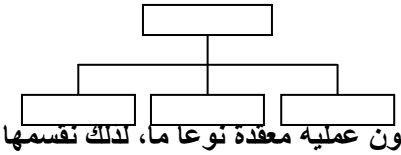
تطورت الأساليب والتقنيات المستخدمة في كتابة البرامج عبر السنين، ويرجع ذلك إلى تغير متطلبات البرمجة وأهميتها وأهدافها.

البرمجة الإجرائية Procedural Programming

البرنامج الإجرائي عبارة عن لائحة من الأوامر تنفذ خطوة خطوة، وهي الطريقة التي يعمل بها المعالج والكمبيوتر بشكل عام لأن الكمبيوتر ينفذ ولا يفكر، يستخدم هذا الأسلوب لكتابة البرامج الصغيرة والمهمة جدا والتي تحتاج إلى ميزات عالية ووصول إلى أجزاء حساسة من الحاسوب مثل إدارة الإقلاع Booting ومن هذه اللغات البرمجية لغة التجميع Assembly.

البرمجة البنيوية Structured programming

جوهر البرمجة البنيوية فالمهمة المعقدة تقسّم إلى مجموعة من المهمات الأصغر إلى أن تصبح بسيطة و مستقلة كي تنجز بسهولة.



على سبيل المثال إن احتساب معدل أجور العمال في شركة ما قد يكون عمليه معقدة نوعا ما، لذلك نقسمها إلى أجزاء صغيرة :-

1. احتساب أجر كل عامل
2. تعداد العمال
3. جمع الأجور
4. تقسيم المجموع على عدد العمال.

يتم حل كل من هذه المشاكل الصغيرة عن طريق إجراء مستقل (Procedure) أو دالة (Function) ثم يتم توليف هذه الإجراءات والدوال للحصول على البرنامج النهائي.

البرمجة المرئية Visual Programming

البرمجة المرئية هي عبارة عن أسلوب جديد للبرمجة، تستخدم فيه برامج مساعدة لتصميم واجهة الاستخدام (الأزرار، النصوص، القوائم، ...) وربطها بالشفرة البرمجية، وتكوّن هذه البرامج المساعدة ببيئة التطوير المدمجة Integrated Development Environment.

الآن ومع استخدام الواجهات الرسومية، تظهر أمام المستخدم عشرات الأزرار والخيارات والقوائم وغيرها، ولا يمكن للبرنامج أن يتوقع ما الذي سيحدث في الخطوة التالية، لذا فإن البرنامج يقسم إلى عدة إجراءات أو وظائف ينفذ كل منها عندما يقع حدث معين : نقرة على الزر الأيمن أو الأيسر للفأرة، ضغط أحد المفاتيح، الاتصال بالإنترنت، ... الخ. ويسمى الإجراء الذي يعمل عند حدوث الحدث بالإجراء الحدثي.

ومن أشهر بيئات التطوير الرسومية نذكر Visual Basic، Visual C++، Visual Java ++، Delphi وغيرها.

البرمجة المرئية هي إذا طريقة برمجة تستخدم لغة أو بيئة برمجة يتم خلالها اختيار المكونات البرمجيّة كالأزرار والصور والأدوات الأخرى بطريقة مرئية عن طريق السحب والإسقاط أو باستخدام القوائم. ويعتمد تشغيل البرامج المرئية على الحدث، على عكس البرامج الإجرائية التي تعتمد على التعليمات النصية التي تؤدي سلسلة من الأعمال وفقا للمدخلات المتوقعة من المستخدم.

لغة Visual Basic

لقد قامت شركة مايكروسوفت بتصميم فيجوال بيسك بناءً على لغة البرمجة بيسك . هذه اللغة (بيسك) مصممة خصيصاً للمبتدئين في البرمجة نظراً لسهولتها على عكس اللغات التي كانت متوفرة حينذاك مثل الكوبول و فورتران و لغة التجميع Assembly
رغم أن لغة بيسك مصممة للمبتدئين إلا أن برامجها تحتاج لبعض الدراسة لفهمها و هي
تعتمد على النصوص بشكل تام لذا فإن البرامج التي تكتب بلغة بيسك تنتج ناتجاً نصياً أيضاً و لا تظهر في نوافذ كما هو حال برامج الويندوز الحالية .

لغة فيجول بيسك 6 غنية عن التعريف، هي لغة برمجة وبيئة تطوير ومعالج، تعتبر من اللغات الشيئية الكائنية المنحنى
تعتبر لغة فيجول بيسك 6 اخر اصدارات البيسك المرئي استكمالا للفيجول بيسك 1وصولاً للفيجول بيسك 5
وبعدها بدءت لغة اصدارات الفيجول ستوديو دوت نت في سنة 2003 وصولاً الى الاصدار الاخير سنة 2010
على الرغم من قدم لغة vb 6 سنة 1998 الا انها اللغة الاكثر شعبية في العالم لسهولته الشديدة مقارنة من السي بلس وغيرها من اللغات ،، الجميل في الفيجول بيسك شاشتها مرئية وبالامكان التعديل على البرنامج بسهولة على عكس الاسمبلي شاشتة تشبه نظام الدوس

1 - أصبح بإمكان المبرمج أن يصمم برامج ذات واجهات جميلة حيث توفر لغة فيجوال بيسك للمبرمج (Objects) وأدوات تستخدم فى نظام التشغيل نفسه كصناديق النصوص والصور إلخ ...
2 - دعم شركة ميكروسوفت لهذة اللغة حيث أضافتها ضمن مجموعة برامج أوفيس من أجل تطوير قواعد البيانات التى ينشأهل برنامج على سبيل المثال MS-Access .
3 - تم تطوير عدد كبير من (OCX ACTIVEX) من خلال العديد من الشركات والمحترفين وطرحت فى الأسواق لتستخدم فيما بعد فى دعم برامج فيجوال بيسك.
4 - دعم جديد من ميكروسوفت للغة فيجوال بيسك فى الإصدار الخامس حيث أضافت فى الإصدار الخامس من فيجوال بيسيك إمكانية بناء عناصر التحكم (OCX ACTIVEX) وبذلك أصبحت لغة فيجوال بيسيك تضاهى بقية اللغات القوية فى إمكانياتها .
5 - أضافت ميكروسوفت إمكانية إستدعاء الكثير من الوظائف الموجودة فى نظام التشغيل وإمكانية الإستفادة من بعض البرامج التى تعمل فى بيئة ويندوز وتسخيرها لصالح البرنامج المنشأ بلغة الفيجوال بيسيك .
6 - مع كل هذا التطور بقيت لغة فيجوال بيسك حتى الإصدار الخامس لها ضعيفة فى معالجة قواعد البيانات وفى إنشاء تطبيقات تدعم الإنترنت مما دفع شركات أخرى كشركة الى طرح عدة إصدارات من لغتها البرمجية (Borland)
(Delphi)دلفى والتى تتمتع بمحرك قواعد بيانات قوى ودخلت ميكروسوفت منافسة مع شركة بورلاند جعلت هذة المنافسة شركة ميكروسوفت تطور فيجوال بيسك لتفوز فى هذة المنافسة وأصدرت ميكروسوفت الإصدار السادس الذى يتميز عن سابقه بدعمه القوى لقواعد البيانات ولا زالت المنافسة مستمرة فظهرت إصدارات أخرى من دلفى وإصدار جديد من الفيجوال بيسيك . Visual Basic.net

وهذة المنافسة فى صالح المبرمجين الآن فيجب أن يستفيدو منها حيث تطرح هذة الشركات أحدث ما توصلو إليه من علوم البرمجة وأحياناً بلا مقابل لتدعم كفائتهافى تلك المنافسة.

شرح واجهة الفيجوال بيسك

هذه الواجهه سوف تظهر عند بدء تشغيل الفيجوال بيسك وفي الأسفل شرح لها.

يمكنك إزالتها من بدء التشغيل عن طريق تفعيل علامة الإختيار Dont't Show this dialog in the future

علامات التبيويب في أعلى الواجهة:

New : إختيار مشروع جديد .
Existing : فتح ملفات مخزنة لديك.
Recent : قائمة بآخر الملفات التي قمت بفتحها حسب التاريخ .

New :يضم مايلي:-

Standard EXE: وهو يقوم بإنشاء برنامج تنفيذي بعد طلبك لذلك من (Make EXE) في قائمة (File) .

Activex EXE: وهو أيضاً يقوم بإنشاء ملفاً له الامتداد (DLL) وهو ملف ذو برامج فرعيه مساعدة وهو لا يعمل بنفسه بل معexe.

Activex DLL: وهو ينشئ ملفات لها امتداد EXE تعمل مع برامج إخرى لكي يعمل في شكل (OLE) .

Activex control: وهو يساعد على إنشاء ملف من نوع (ocx) و هي مهمة وتساعد على إحتواء واجهة مستخدم أو برامج فرعية.

Activex Document DLL: وهو يساعد على إنشاء ملف ذو امتداد (DLL) يقوم بتشغيل برامج على موقع في الانترنت.

Activex Document exe: يقوم بإنشاء ملفات تظهر على الانترنت.

Add In: وهو خاص بالفيجوال بيسك حيث يمكنك إضافة واجهة مع ما ذكر سابقاً خاصة بك.

VB Application Wizard : وهو معالج تلقائي في هذه اللغة يساعدك على إنشاء نوافذ عدة دون كتابة سطر واحد من النوع exe .

IIS Aplication : نقصد بـ (IIS) إختصار لكلمة (Internet Information Server) و يقوم بالتحضير لتصميم (web class) وهو ما يسميه المستخدم صفحات(HTML) .

Data project : تضع مشروعاً لقواعد بيانات وتهيئة في بيئة متكاملة من أدوات الربط وصنع التقارير دون الحاجة لفتح برامج أخرى.

Vb wizard manager : و تبدو مشابهة لمعالجات (Microsoft) يمكنك إستخدامه.

DHTML Application : تسهل عليك صنع صفحات (HTML) ديناميكية للإنترنت دون تعلم لغة Java .

Vb Enterprise Edition controls : وهي تساعد على وضع جميع الأدوات المستخدمة بكثرة لقواعد البيانات و الاتصال ومن ثم تطبيقها وهي للمحترفين.

يحتوى الفيجوال بيسيك علي 13 قائمة ، و هي كالتالي

File : تحتوي علي اوامر اساسية، مثل فتح و حفظ مشروح، و فتح مشروع جديد، و تحويل البرنامج الي امتدادExe.

Edit : تحتوي علي اوامر التحرير العادية بالاضافة الي اوامر اخري.

View: تعمل محتويات هذه القائمة علي اظهار بعض الاشياء مثل صفحة كود و فورم و صندوق الادوات...الخ .

Project: تحتوي هذه القائمة علي اوامر خاصة بمحتويات المشروع مثل اضافة Module او ClassModule او غيرها.

Format: بهذه القائمة يمكنك تنسيق برنامجك، مثلا وضع زر في منتصف الشاشة تماما او محاذاته الي اليمين او اليسار او غيره، و بها ايضا الامر LockControls الذي يمكنك من عدم السماح بتحريك او تغير حجم كائن، و يمكنك ازالته بالضغط عليه مرة اخري.

Debug : تحتوي هذه القائمة علي معظم اوامر التشغيل، و منها اختيار طريقة تنفيذ البرنامج، مثل تنفيذ سطر واحد منه StepInto ، او تنفيذ كله StepOver ، او الامر السابق StepOut ، او التنفيذ حتي السطر الذي يوجد عليه مؤشر الكتابة RunToCursor ، و BreakPoints التي تظهر باللون الاحمر عند;كتابه الكود، فعند اختيارها يتم توقف مؤقت للبرنامج اثناء تشغيله.

Run : من خلال هذه القائمة يمكنك تشغيل او ايقاف البرنامج.

Query : تحتاج هذه القائمة الي جمل استعلامSQL.

Diagram : تحتاج هذه القائمة الي SQL اوORACLE.

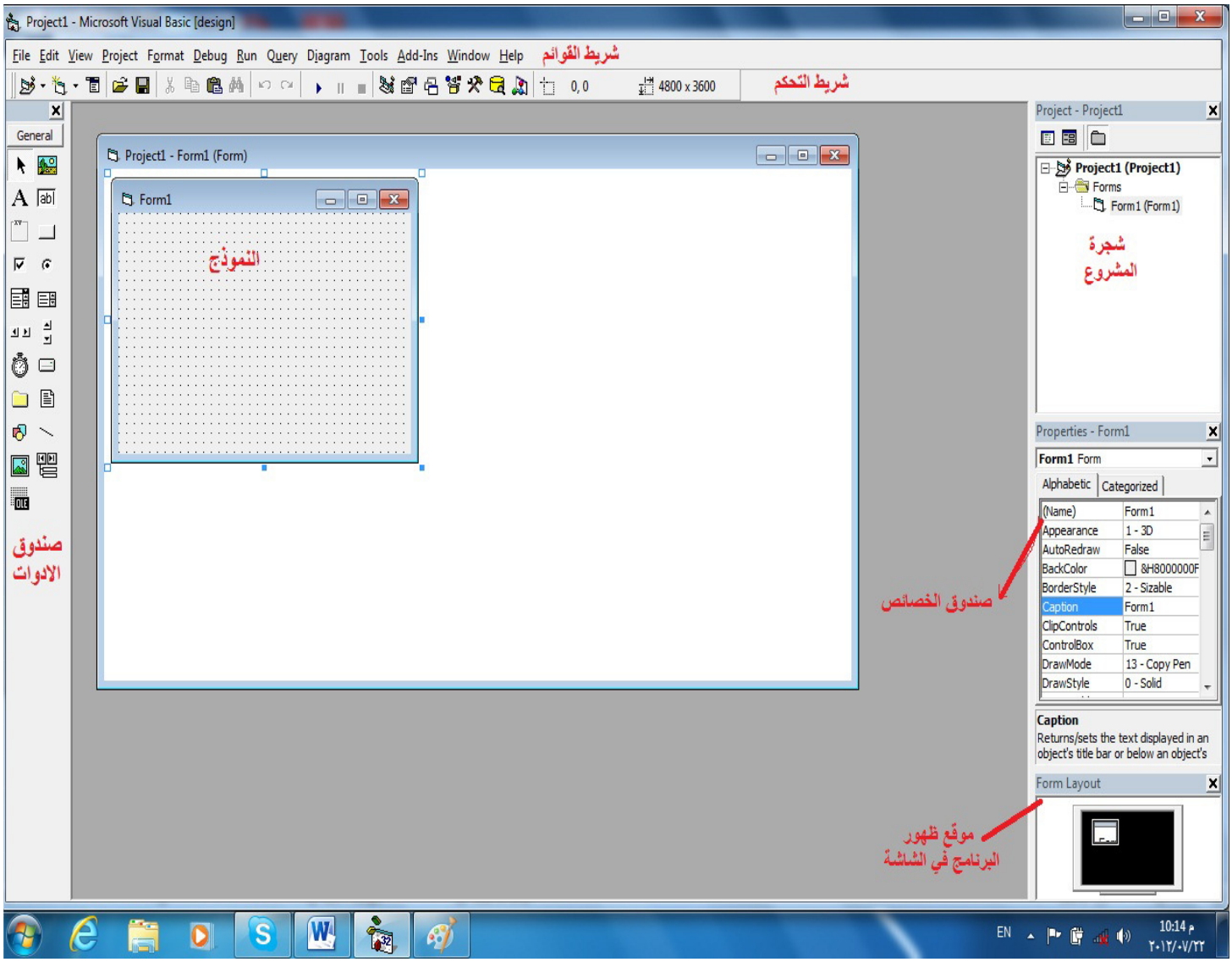
Tools : تحتوي هذه القائمة علي اوامر كثيرة مثل AddProcedur الذي يمكنك من كتابة اجراء جديد و لكن يجب ان تكون في صفحة الكود عند اختيار هذا الامر، اما ProcedureAtributes فهو يمكنك تخصيص بعض الاشياء للاجراء الذي انشأته مثل تغيير ID و غيرها.

Add-Insert : تحتوي هذه القائمة على برامج مستقلة توفر خدمات للبرنامج .

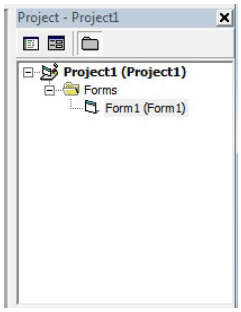
Window : تنظم هذه القائمة صفحة الكود و الفورم ليكونا ظاهرين مع بعضهما.

Help : تحتوي هذه القائمة علي تعليمات خاصة ولا يمكنك فتحها الا اذا تم تحميل مكتبة MSDN و بها ايضا الصفحة الخاصة بنبذة عن الفجوال.

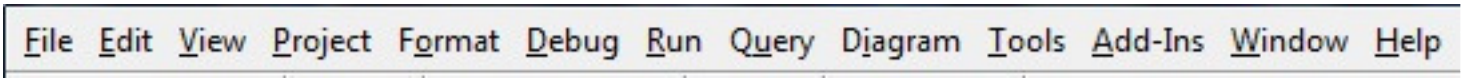
واجهة البرنامج



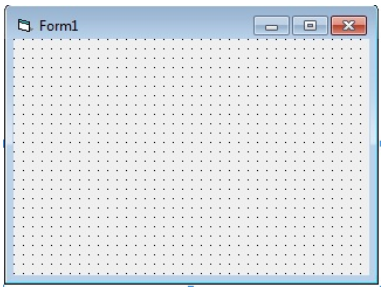
1 - شجرة المشروع (TREE PROJECT) : ويقوم بعرض جميع المشاريع و النماذج للمشروع الحالي.



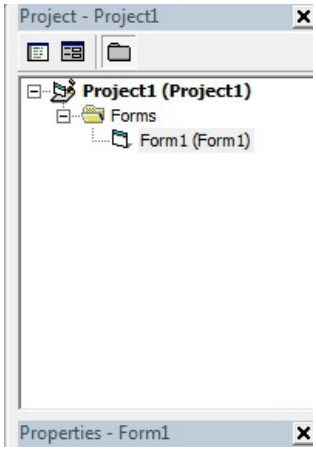
2 - شريط القوائم (Menu bar): يحتوي على 13 قائمة تحتوي كل قائمة منها على مجموعة اختيارات مختلفة. فمثلاً تستخدم قائمة file للتعامل مع ملفات المشاريع المختلفة بينما تستخدم قائمة Edit في كتابة البرامج وتصميم النماذج وتستخدم قائمة view للتحكم في عرض أو إخفاء عناصر بيئة التصميم، وتستخدم القائمة Run لتنفيذ أو إيقاف البرنامج وتستخدم القائمة Debug لاكتشاف وتصحيح الأخطاء.



3 - نافذة البرنامج (Form Window) : يظهر النموذج في منتصف الشاشة تقريبًا ويستخدم لتصميم واجهات البرنامج حيث توضع عليه جميع الأدوات المستخدمة. ويحتوي البرنامج على واحد أو أكثر من هذه النماذج



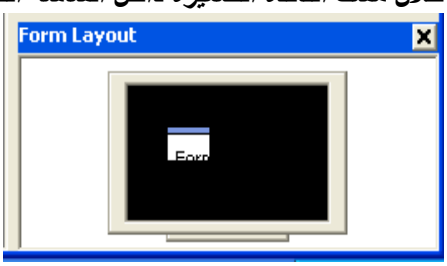
4 - نافذة الخصائص (Properties Window): تضم نافذة المشروع مجموعة الملفات والنماذج الخاصة بالمشروع ويمكن رؤية أي عنصر إما بالنقر المزدوج عليه أو بتنشيطه ثم نقر زر View Code أو زر View Form.



5 - نافذة الأدوات (TOOL BOX) : يشتمل على مجموعة أدوات التحكم التي تسهل التعامل مع Visual Basic وتستخدم في إعداد البرامج. يمكن نقل مربع الأدوات إلى أي مكان داخل الشاشة.

6 - شريط الأدوات (Toolbar) : يحتوى شريط الأدوات على مجموعة من الأزرار يقوم كل منها مقام وظيفة يمكن استدعاءها مباشرة بمجرد النقر عليها بدلاً من فتح القوائم واختيار الأوامر.

7-نافذة (form Layout) : ووظيفة النافذة انه يبين لك موقع النافذة بعد تشغيل البرنامج اي في اي اتجاه من الشاشة وتستطيع تحريك النافذة من خلال النافذة نفسها وذلك بتحريكه من خلال مسك النافذة الصغيرة داخل الشاشة الظاهرة امامك وتغيير مكان البرنامج من الشاشة .



شرح صندوق الأدوات

شكل الأداة	إسمها	شرحها
	Picture Box	صندوق عرض الصور
	Text Box	أداة النصوص والكتابة
	Label	أداة عرض النصوص
	Frame	إطارات داخل البرنامج
	Check Box	زر إختيار
	Combo Box	قائمة منسدلة
	Command Button	زر أمر
	Data Control	ربط قاعدة بيانات
	Drive Box	أداة عرض السواقات
	Dir Box	أداة عرض المجلدات
	File Box	أداة عرض الملفات
	Image	أداة الصور
	List Box	أدة القائمة
	Line	خط
	Ole Object	عنصر تحكمOle
	Option Box	زر خيارات
	Vertical Scroll Bar	شريط التمرير العمودي
	Horizontal Scroll Bar	أشرطة التمرير الأفقي
	Shape	أداة الأشكال
	Timer	أداة التوقيت

شرح نافذة الخصائص

في هذا الجدول خصائص عناصر التحكم القياسية

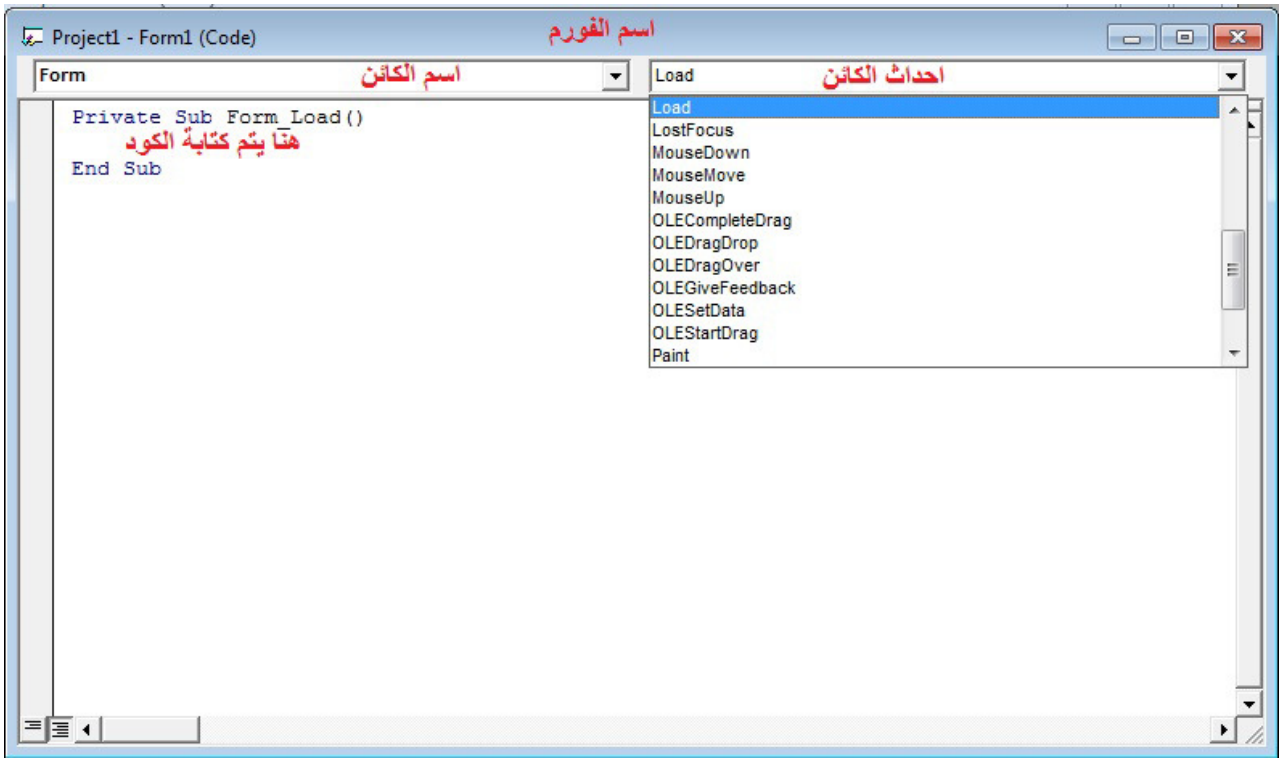
الخاصية	الوظيفة
Name	تحديد اسم للاداة
Backcolor	لون الخلفية
Borderstyle	نوع وسمك اطار النافذة
Caption	عنوان الاداة
Text	عنوان الاداة
Controlbox	مايظهر في اعلى كل نافذة في اقصى اليمن للتحكم بالنافذة
Enabled	لتمكين الاداة
Font	الخط
Forecolor	اللون الامامي
Height	الارتفاع
Icon	الرمز
Left	موقع الاداة من اليسار
Mdichild	تستخدم في البرامج متعددة النوافذ
Mousepointer	مؤشر الفارة
Picture	الصورة
Righttoleft	مفيدة جدا في البرامج العربية لتحويل القوائم من اليسار الى اليمين
Top	موقع الركن العلوي من الاداة
Visible	اخفاء الاداة
Width	العرض
Windowstate	حالة النافذة من حيث الحجم
Backstyle	جعل الاداة شفافة
Borderstyle	حذف الاطار المحيط بالاداة
Databasename	اسم قاعدة البيانات
Recordset	اذا استخدمنا البحث في قاعدة البيانات Seek يجب ان توضع في حالة table بالامر
Recordsource	تحديد الجدول من قاعدة البيانات
Datafield	ربط اداة النص بحقل معين من جدول معين
Datasource	ربط اداة النص بجدول معين من قاعدة البيانات
Value	القيمة
Max	الحد الاعلى
Min	الحد الادنى
Smallchange	التغيير الصغير
Largechange	التغيير الكبير

Interval	الفترة الزمنية المراد تنفيذ حدث المؤقت بعد مرورها
Tooltiptext	التعليق الذي يظهر فوق الاداة عند ابقاء الفارة علية لمدة قصيرة
Tabstop	توقيف التنقل بزرtabعلى زر معين
Tabindex	ترتيب التنقل بين الادوات

الأحداث Events

إذا ارت ان تنقر بفأرة (الماوس) على شيء فهذا يسمى حدث و الفجوال بيسك 6 لا يقدم هذا الحدث فقط بل جملة من الأحداث المختلفة لكل الأدوات ، و سنأخذ المهم منها :-

قم بالضغط المزدوج على الزر (CommandButton1) ستظهر لك هذه الصورة هنا يتم تغيير الأحداث للأدوات.



اضغط على السهم الذي يغير الأحداث ستظهر لك الأحداث المتاحة لهذا الكائن .
و تستطيع نسخ و لصق و الحذف أي شفرة (code) بدون خوف مما يجعل إمكانية التحرير للشفرة واسعة .

و تختلف أحداث الكائنات على حسب الاحتياجات ، اختر الأحداث المناسبة لتنفيذ عملك و سنأخذ معلومات عن الأحداث .

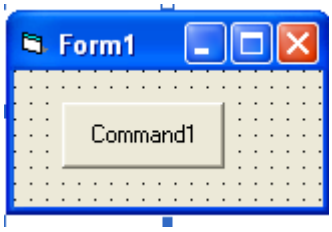
مجموعة من الأحداث

-:Click-1

و هو حدث الضغط بالفأرة مرة واحدة .

مثال :-

اضف اداة command الى الفورم ؟



كتابة الكود

```
Private Sub Command1_Click()
Command1.Caption = "click"
End Sub
```

شرح الكود

عند النقر بزر الماوس سوف يتغير اسم الزر الى click .

-:Dblclick-2

و هو حدث الضغط المزدوج بالفأرة.

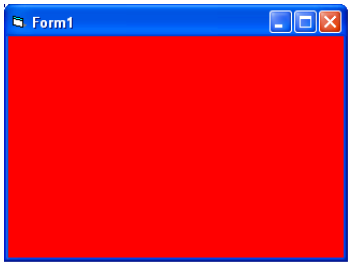
مثال :-

اكتب الكود التالي في اداة الفورم ؟

```
Private Sub Form_DblClick()
Form1.BackColor = vbRed
End Sub
```

شرح الكود

عند النقر المزدوج بالماوس تتغير خلفية الفورم الى اللون الاحمر.



-:MouseMove-3

و هو حدث يظهر عندما تحرك الفأرة .

مثال :-

اضف اداة command الى الفورم واكتب الكود التالي في الحدث المذكور ؟

```
Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Command1.Caption = "mouse move"
End Sub
```

شرح الكود

عند تحريك موشر الماوس فوق الزر سوف تتغير اسم الزر الى mouse move .

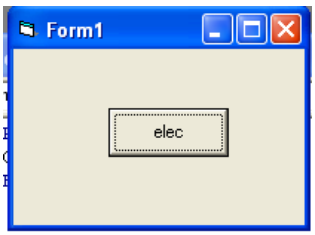
-:MouseDown-4

و هو حدث يظهر عندما تستمر بضغط الفأرة على الكائن.

مثال :-
اضف اداة command الى الفورم واكتب الكود التالي في الحدث المذكور ؟

```
Private Sub Command1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
Command1.Caption = "elec"
End Sub
```

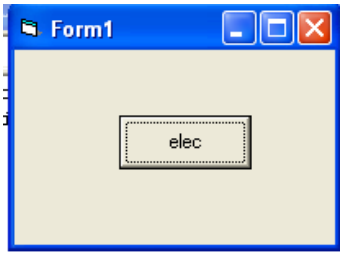
شرح الكود
عند الضغط بالماوس على الزر سوف تتغير الاسم الى elec.



MouseUp-5:-
و هو الحدث الذي يظهر عندما تسحب يدك من الضغط من الفأرة على الكائن.
مثال :-
نفس المثال السابق نقوم بتغيير الحدث فقط .

KeyDown-6:-
و هو حدث يظهر بعدما تضغط على لوحة المفاتيح.
مثال :-
اضف اداة command الى الفورم واكتب الكود التالي في الحدث المذكور ؟

```
Private Sub Command1_KeyUp(KeyCode As Integer, Shift As Integer)
Command1.Caption = "elec"
End Sub
```



KeyUp-7:-
و هو حدث يظهر بعدما تسحب يدك من الضغط على لوحة المفاتيح.
مثال :-
نفس المثال السابق نقوم بتغيير الحدث فقط .

KeyPress-8:-
و هو حدث يظهر عندما تضغط على أزرار معينة على لوحة المفاتيح
مثال :-
نفس المثال السابق نقوم بتغيير الحدث فقط .

وهناك احداث اخرى كثيرة تختلف وظائفها حسب الحدث ولقد تطرقنا الى اهمها في الشرح .

ملاحظة :* للسهولة تستطيع نسخ و لصق و حذف أي شفرة من مكان و ضعها في مكان آخر.
* الشفرة يتم ترجمتها بالتسلسل سطرا سطراً أي ما كتبته أولا يتم ترجمته ثم الذي يليه .

مصطلحات هامة

ستقابلنا من آن لآخر بعض المصطلحات ومن أهمها :-

* أدوات التحكم (Controls): كائنات برمجية (objects) تم تصميمها مرة واحدة وتستخدم مرات عديدة، وهي تمثل القطع المكونة لواجهة البرامج المصممة بـ VB من أمثلة أدوات التحكم نذكر مربع النص (Textbox) وزر الأمر (Command Button) .

* الحدث (Event) : هو فعل يقوم به المستخدم أو نظام التشغيل أو البرنامج نفسه. من أمثلة الأحداث نذكر ضغط أحد المفاتيح أو نقر احد أزرار الفأرة.

* كائن (Object) : الكائن البرمجي أحد العناصر الأساسية للبرنامج، يحتوى على خصائص (Properties) تحدد مميزاته ، ووظائف (methods) تحدد مهامه التي يمكن إنجازها ومجموعة من الأحداث التي يمكن أن يشعر بها. من أمثلة الكائنات المستخدمة في VB النماذج (forms) وأدوات التحكم (Controls).

* الخصائص (Properties) : هي مزايا الكائن مثل حجمه وموقعه على الشاشة ولونه ونوع الخط المستخدم في الكتابة عليه. أي أن الخصائص تحدد مظهر الكائن على وجه الإجمال.

* الوظائف (methods) : هي أفعال محددة يمكن للكائن أن يقوم بها، مثلاً النافذة (form) لها وظيفة تسمي)hide تقوم بإخفائها. أي أن الخصائص تحدد سلوك الكائن على وجه الإجمال.

*Parameters: يمثل القيمة الجديدة للخاصية .

* الاجراءات (Procedures) : هي مقاطع من التعليمات (والتي نطلق عليها الكود) تكتب لغرض معين وغالبًا ما تكون مقترنة بحدث فتسمي عندئذ بالاجراءات الحدثية .

*النماذج (Forms):عبارة عن الشاشة التي يتم استخدامها لأجراء عملية التصميم وعليه تضاف الكائنات مثل صناديق النص والصور والعنوان .

*الوحدة النمطية القياسية :- ويعني بها نافذة module وسيتم التعرف اليها اكثر من خلال الشرح .

مكونات لغة البرمجة فيجوال بيسك

1 - الوظائف. (Functions)
2 - الطرق أو الوسائل (Methods)
3 - الأحداث (Events)
4 - مفاتيح المهام (Keywords by Task)
5 - الخصائص (Properties)
6 - التعبيرات (Statements)
7 - الكائنات (Objects)

الثوابت Constants

هي قيمة يتم حجزها في الذاكرة وتبقى ثابتة على مدى البرنامج ، تنقسم الثوابت في لغة الفيجوال بسيك إلى ثلاثة اقسام رئيسية وهي(نصية او رقمية او منطقية) ويمكننا تعريف ثابت في البرنامج من خلال الكلمة المحجوزة :

```
Cons  name  as  type = value
```

ولاحظ هنا عند كتابة الثابت لا يمكنك نسب قيمة له إلا في نفس السطر الذي قمت بكتابة الثابت فيه .

الطرق التي من خلالها تستطيع تعريف ثوابت لبرنامج

1 - الطريقة التقليدية تعريف ثابت ونسب قيمة له مثل :

```
Cons name as string = "hussain "
```

2 - كتابة الثابت بدون تحديد نوعه مثل :-

```
Cons name = "hassan "
```

3 - كتابة اكثر من ثابت مع تحديد القيمة لها في سطر واحد مثل :-

```
Cons name as type = value
Cons name as string = "eng " , age as integer = 22
```

4 - كتابة اكثر من ثابت بدون تحديد نوع الثابت مثل :

```
Cons name = "elec " , age = 22
```

فوائد الثوابت في البرنامج

A - تسهيل البرمجة عندما تريد التعويض بقيمة ثابتة لعدد غير معين من المتغيرات.
B - انها تتعامل مع المصفوفات بكل يسر وسهولة ويمكنها الوصول إلى مصفوفة معينة .

الاخطاء المتوقعة حدوثها عند تعاملك مع الثوابت

1 -تغيير قيمة الثابت في البرنامج او اثناء التشغيل مثال:-

```
Cons age  as integer = 20
Age = 25
```

2 - تعريف الثابت في سطر ونسب القيمة له في سطر اخر كما قلنا سابقاً .

```
Cons age  as integer
age = 15
```

مدي الثوابت

1- ثوابت عامة : إذا أردت أن يكون الثابت عاما أي يمكن استخدامه من أي مكان في البرنامج فيجب أن تعلن عنه في الوحدة النمطية بشرط أن يسبق الإعلان عنه كلمة (Public) .

مثال :-

```
Public Const MyName As String = " electronic "
```

2- ثوابت على مستوي الوحدة النمطية : لكي تستخدم الثابت في وحدة نمطية فقط يجب أن تعلن عنه في قسم الإعلانات في هذه الوحدة النمطية تسبقه كلمه (Private) .

مثال :-

```
Private Const MyAge As Integer = 25
```

3- ثوابت علي مستوي الأجراء :لكي تستخدم الثابت مؤقتا داخل إجراء معين أعلن عن الثابت داخل هذا الإجراء بنفس الطريقة السابقة .
و نكتشف من الكلام السابق أن الثوابت مشابه إلي حد كبير للمتغيرات في طريقة الإعلان عنها بأمر الإعلان (Const) أو حتى الإعلان عن نوع الثابت لزيادة سرعة التعامل معه وأيضا مجال رؤية الثابت وعمرة.

انواع الثوابت

1 - ثابت عددي وهذه القيمة تقسم إلى نوعين :
A - ثابت عددي صحيح INTEGER .
B - ثابت عددي حقيقي REAL وله شكلان :
a - شكل عشري : مثل 1.15
b- شكل أسي : مثل 3 ^3

2 - ثابت حرفي STRING :
وهو تابع لمحارف مكتوبة بين علامتي التنصيص " " لا يستخدم الثابت المحرفي في العمليات الحسابية وإنما يعامله الكمبيوتر كنص مثلاً لو أدخلنا التعليمة Print 2*3 فإنه سوف يظهر ناتج ضرب العددين أي 5 .

أما لوأدخلنا التعليمة Print "2*3" فإنه سوف يظهر الرقمان كما هما (2*3).

ملاحظة هامة جداً:- عند تعاملنا مع الثوابت يجب علينا ادخال القيمة للثابت في نفس جملة تعريف الثابت اي في نفس السطر .

الإعلان عن الثوابت

1-إذا تم الإعلان داخل إجراء يعتبر الثابت المعلن عنه محليا (Const a=5)

2-وإذا تم الإعلان في القسم العام للنافذة يصبح عاما لجميع إجراءات تلك النافذة (Const a=5)

3-وإذا تم الإعلان في القسم العام للملف يصبح عاما لجميع إجراءات ذلك الملف (Const a=5)

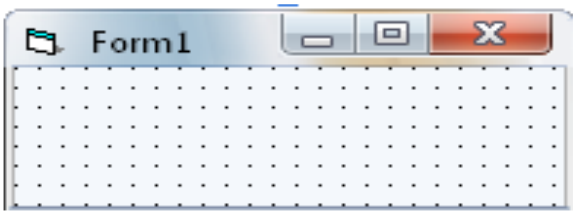
4-أما إذا تم الإعلان في القسم العام للملف - كما سبق - لكن سبقت كلمة (Const) بكلمة (Public) فسيصبح الثابت شاملا لجميع ملفات البرنامج ونوافذه

يتم الإعلان عن الثابت الشامل هكذا (Public Const a=5)

مثال :- لتعريف قيمة x في البرنامج ؟

تعريف على مستوى الاجراءات	Const x =3.14
تعريف على مستوى الاجراءات و الفورم	Private const x =3.14
تعريف على مستوى الفورم	Public const x =3.14

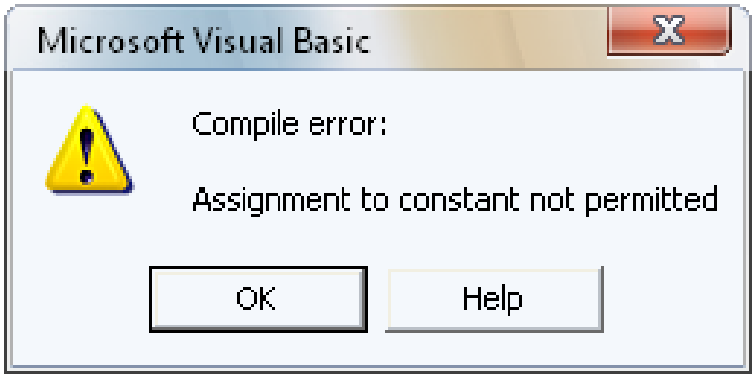
مثال :- صمم الشكل التالي للفورم ؟



الحل :-

```
Private Const y = 10
Private Sub Form_Load()
Const x = 3.14
Print x, y
End Sub
```

ملاحظة :- عند قيامك بتغيير قيمة الثابت في الكود مثل x =x+1 سوف يظهر لك النافذة التالية ينبهك بوجود خطأ في الكود و السبب انه لا تستطيع تغيير قيمة الثابت لانك اعلنت عنه كقيمة ثابتة غير قابلة للتغيير .



المتغير هو مكان في الذاكرة يتم تخصيصه لك لتقوم بوضع عنوان له ويخزن المعلومة التي تريدها بداخلة كما يمكنك تغير هذه المعلومة بمعلومة أخري في نفس المكان ويظل عنوانه ثابت بعكس الثابت .

هناك ستة طرق للاعلان عن المتغيرات

1- أمر الإعلان Dim : يستخدم لتعريف متغير ديناميكي (Dynamic Variables) ضمن الأجراء ويكون مجال رؤية هذا المتغير داخل الأجراء فقط وعمرة من عمر الأجراء أي أن عندما ينتهي الأجراء ينتهي معه مفعول المتغير المحلي اي داخل الفورم فقط ويصبح لا قيمة له أو صفر ولهذا سمي أمر الإعلان هذا بديناميكية لأنه يوفر في الذاكرة بشكل جيد .

Ex) w as Integer

2- أمر الإعلان Redim: تستخدم لتعريف مصفوفة ديناميكية (Dynamic Array) غير معرفة لعدد العناصر أي يمكنك تغير أبعادها أثناء عمل البرنامج ويعمل أمر الإعلان (Redim) عندما ترغب في تحديد حجم المصفوفة وتظهر الفائدة من أمر الإعلان (Redim) في استغلال الجزء المطلوب فقط من الذاكرة دون زيادة . وأمر الإعلان هذا يجعلنا ندخل إلي عالم آخر وهو عالم .

ملاحظة :- أن أمر الإعلان (Redim) لا يستخدم ألا داخل أجراء فقط أي لا يستخدم في قسم الإعلانات مثل الأمر (Dim) .

3- أمر الإعلان Static : يستخدم لتعريف متغير ستاتيكي (Static Variables) ضمن الأجراء ويكون مجال رؤية هذا المتغير داخل الأجراء فقط وعمرة من عمر الوحدة التي بداخلها الأجراء سواء كانت الوحدة هذه (Form) أو (Module) أو (Class) أي أن عندما ينتهي الأجراء يظل المتغير ساكن في الذاكرة وكذلك قيمته موجودة ولكنك لا يمكنك الوصول إلية ألا من داخل الأجراء التابع له.

ملاحظة :- حاول أن لا تكثر من استخدام أمر الإعلان (Static) ألا في أضيق الحدود لأنه يستغل الذاكرة طوال فترة عمل البرنامج .

Ex) Static jhz As Integer

4- أمر الإعلان Private : يستخدم لتعريف متغير عام (General Variables) ولكنه يكون على مستوي الوحدة سواء كانت (Form)أو ملف برمجة (BAS) أو فئة (Class) وبالطبع عمره يكون من عمر ال.

Ex) Private zainab As String

5- أمر الإعلان Public: يستخدم لتعريف متغير عام (General Variables) ولكنه يكون علي مستوي المشروع ككل ويمكن الوصول إليه من جميع الوحدات الموجودة في المشروع وعمرة من عمر البرنامج ككل ويظل في الذاكرة حتى ينتهي البرنامج.

Ex) Public elec AS Integer

6- أمر الإعلان Global: أمر الإعلان أو الكلمة المحجوزة (Global) كانت تستخدم في الإصدارات القديمة للفيجول بيسك ومازالت حتى الآن تستخدم وهي تؤدي نفس وظيفة أمر الإعلان (Public) ولكنك لن تستطيع التصريح عنها ألا في الوحدات النمطية فقط.

الإعلان باستخدام الوظيفة AS :
وهنا تفيد الوظيفة AS في تميز نوع المتغير الذي يأتي بعد الوظيفة AS مع أحد الأوامر Redim. Dim. Static. Global Private Public حيث يتم كتابة الأمر ثم اسم المتغير ثم كتابة الوظيفة AS ثم كتابة نوع المتغير.

مثال :- لتعريف متغير من نوع (String)

Dim hassan AS String

في قسم التصريحات (General)

تعريف متحول عددي
Dim no1 As Integer

تعريف مصفوفة عددية سعتها عناصرها (10)
Dim list(10) As Integer
تعريف مصفوفة مؤلفة من (10) عناصرعرض ، و(10) عناصر إرتفاع ، أي(100) عنصر
Dim matrix(10, 10) As Integer

تعريف متغير نصي
Dim name1 As String

تعريف متغير نصي بطول (50) حرف
Dim name1 As String * 50

تعريف مصفوفة نصية تتكون من (10)عناصر
Dim list1(10) As String

تعريف مصفوفة نصية تتألف من (10) عناصر بطول (50) حرف

Dim list1(10) As String * 50

تعريف متغير للقيم(نعم) او(لا)
Dim a As Boolean

ملاحظة :- أوامر الإعلان السابق ذكرها هي تعتبر ضمن الكلمات المحجوزة مثل الجملة (Option Explicit) أو أمر (Print) والكلمات المحجوزة هي الكلمات التي يحتفظ بها الفيجول بيسك لنفسه ولا يمكنك استخدامها كاسم للمتغير أو للثابت .

الإعلان باستخدام أمر تعريف دالة

في هذه الطريقة يتم استخدام أحد الأوامر التالية:

(CCur , CLng , CDbl , CInt, CStr , CSng, Cvar)

للأنواع الآتية علي التوالي .

(Currency , Long , Double , Integer , String , Single , Variant)

وعند كتابة أي حرف بعد أي من هذه الأوامر تتحول كل المتغيرات التي تبدأ بهذه الأحرف إلي نفس النوع المعلن عنه في الأوامر المستخدم .

مثال :- نستخدم الأمر (CInt) للإعلان عن جميع المتغيرات الموجودة في البرامج والتي تبدأ بحرف (A) علي إنها من النوع (Integer) .

CInt A

مثال :- الإعلان عن المتغيرات الموجودة في البرامج و التي تبدأ بأحد الأحرف التالية (B أو C أو D) على أنها من النوع (String)

Cstr B-D

لاحظ أنه لا ينبغي كتابة أكثر من حرفين متصلين بدون العلامة (-) بعد الأمر .
للإعلان عن أي متغير يبدأ بالحرف (A) أو بأحد الحروف من (D) إلي (F) أو من (X) إلي (Z) على أنه من النوع(Double)

مثال :-

CDbl A, D-F , X-Z

Lifetime and Scope of Variable مدى استخدام المتغير وعمره

ويقصد بمدى استخدام المتغير (Scope Of Variable) والإجراءات و الوحدات النمطية التي ستتأثر به أي الأماكن التي أن يستخدم فيها هذا المتغير داخل البرنامج أما عمر المتغير (Lifetime Of Variable) فيقصد به المدة التي سيبقي المتغير خلالها محتفظا بقيمته الحالية داخل الذاكرة دون أن يفقدها وتنقسم المتغيرات من حيث مدة بقائها في الذاكرة ومداها إلي متغيرات عامة ومتغيرات علي مستوي الوحدة النمطية ومتغيرات علي مستوي الأجراء وفيما يلي نوضح كل نوع من هذه الأنواع الثلاثة والأمر الذي يستخدم للإعلان عنه .

1 - المتغيرات العامة .

هي المتغيرات التي يمكنك استخدامها من أي مكان داخل البرنامج أو التطبيق وتبقى في ذاكرة الحاسب طوال فترة عمل البرنامج فإذا انتهى البرنامج تحذف من الذاكرة ولذلك يجب أن يعلن عن المتغير العام من خلال الوحدة النمطية لكي تتعرف عليه جميع الإجراءات الموجودة في جميع الوحدات النمطية بالبرنامج أو التطبيق.

يستخدم الأمر (Public)للإعلان عن المتغيرات العامة في المثال التالي يتم الإعلان عن متغير عام لكي تستخدمه جميع الإجراءات في جميع الوحدات النمطية من نوع (Integer)واسمه (mm)

مثال :-

```
Public mm AS Integer
```

2- المتغيرات على مستوي الوحدة النمطية .

بإمكانك الإعلان عن متغير وتقيده على مستوى وحدة نمطية في هذه الحالة لن تستطيع استخدام المتغير إلا من خلال الوحدة النمطية التي أعلنت عنه فيها ولن تستطيع استخدامه خارجها ، فترة عمل هذا النوع من المتغيرات هي أيضا فترة عمل البرنامج أي الفرق بينها وبين المتغيرات العامة هو في المدى الذي تستخدم فيه فقط.

للإعلان عن متغير من هذا النوع استخدم الأمر (Private) بدلا من الأمر (Public)في المثال التالي يتم الإعلان عن متغير من نوع (String)واسمه (zh) لكي يستخدم فقط مع الوحدة النمطية التي يوجد بها.

```
Private zh  As String
```

وهذا الأمر يمكن إدخاله من خلال الأجراء لإدخال الأمر علي مستوي الوحدة النمطية استخدمه بنفس الطريقة التي تستخدمها للإعلان عن المتغير العام مع فارق واحد وهو استخدام أمر (Private) بدلا من أمر(Public)

3- متغيرات علي مستوي الأجراء .

يقتصر مدى هذه المتغيرات علي الأجراء الموجود به فقط ولا يمكن استخدامه في أي مكان غيرة وهي بهذا تعتبر اقل المتغيرات مدى من حيث عمرها فهي تبقي موجودة بالذاكرة حتى بعد أن ينتهي الأجراء الذي أعلن فيه عنها وبهذا يتضح أن الفرق بين هذه المتغيرات و المتغيرات العامة أو المتغيرات علي مستوي الوحدة النمطية في مداها فقط حيث لا يتعدى مداها الأجراء الذي أعلن عنها فيه . يستخدم لهذا الغرض الأمر (Static) ويتضح ذلك من المثالين التالين في المثال الأول يظهر المتغير (S2) محتفظا بقيمته فترة تنفيذ الأجراء (Load Form) وبمجرد الخروج من الأجراء ستكون قيمته تساوي صفر بينما في حالة استخدام المثال الثاني فإن المتغير سيحتفظ بقيمته بعد تنفيذ نفس الأجراء السابق.

مثال :-

```
Private Sub Form_Load (Cancel As Integer)
Dim S2 As Integer
S2 = 5
End Sub
```

ملاحظة :- يوفر عليك مدي المتغيرات وعمرها استهلاك مساحة من الذاكرة بدون داع فمثلا إذا كنت تريد استخدام متغير في أكثر من وحدة نمطية فيجب أن تعلن عنه كمتغير عام بالأمر (Public) وإذا كنت تحتاج للمتغير في وحدة نمطية واحدة فقط استخدم الأمر(Private) للإعلان عنه أما إذا كنت تحتاج للمتغير مؤقتا في هذا الأجراء فقط استخدم أمر (Static) ليبقي مداه داخل الأجراء فقط.

معنى ذلك أن هذه المستويات بأختصار هى على النحو التالى :-

في القسم العام لملفات البرمجة Dim عام ، Private عام ، Public شامل ، Global شامل

في القسم العام للنوافذ Dim عام ، Private عام ، Public عام

داخل إجراءات الملفات Dim ساكن

داخل إجراءات النوافذ Dim محلي ، Static ساكن

المصطلحات :-

1- شامل : عام لجميع نوافذ البرنامج وملفاته .

2- عام : عام لجميع إجراءات النافذة أو الملف الذي تم الإعلان داخلها (الشامل أعم) .

3- المحلي والساكن : كلا منهما خاص بالإجراء الذي تم الإعلان داخله .

الفرق بين المحلي والساكن

1- المتغير المحلي يفقد قيمته عند الخروج النهائي من الإجراء .

2- الساكن يحتفظ بقيمته طوال مدة تشغيل البرنامج .

الشروط اللازمة لكتابة اسماء المتغيرات في البرنامج :

1- عدم استخدام الكلمات المحجوزة في البرنامج مثل :

(if dim while do)

2- عدم البدء برقم في اسم المتغير مثل:-

dim 2m as string

3 - يمنع استخدام الرموز في اسماء المتغيرات مثل :-

dim $ as string

4 - يمنع استخدام النقطة او الفاصلة في اسم المتغير مثل :-

dim a . as string
dim a , as string

تنقسم المتغيرات إلى اربعة اقسام رئيسية وهى كالتالى :-

1 - متغيرات عددية .
2 - متغيرات نصية .
3 - متغيرات منطقية.
4 - متغير عام ويحمل كل هذه القيم .

طرق تعريف المتغيرات في البرنامج

1- لكتابة نوع المتغير هو بوضع علامة كما موضح في في الجدول:-

نوع النتغير	الرمز
Integer	%
Long	&
Single	!
Double	#
Currency	@
String	$

مثال :-

```
Dim Input1 $
Static Av %
Private Total !
```

2- وهي الطريقة الاعتيادية وذلك بكتابة اسم المتغير كاملة .

مثال :-

```
Dim input as String
Static AV as Integer
Private total as Single
```

طرق تعريف المتغيرات في البرنامج باستخدام الكلمة المحجوزة (Dim)

1 - الطريقة الافتراضية لتعريف المتغيرات :-

```
Dim name as string
```

2 - طريقة تعريف متغير منفصل مع نسب القيمة له في نفس السطر :-

```
Dim name as string = "ali "
```

3 - طريقة تعريف اكثر من متغير في سطر واحد ويكونوا لنفس النوع (اي كلاهما من نوع نصي او كلاهما من نوع رقمي) :-

مثال:-

```
Dim f _ name as string , 1_ name as string
```

في هذا المتغير عرفنا متغيريين من نوع نص اسم المتغير الاول (F_name) والثاني(L_name)

4 - تعريف متغيرات مختلفة في نفس السطر (اي مختلفة في النوع اي متغير نصي ومتغير رقمي ومتغير منطقي) مثل :-

```
Dim name as string , age as integer  , elec as boolean
```

عرفنا (3) متغيرات من انواع مختلفة منها النصي ومنها الرقمي ومنها المنطقي.

لاحظ اننا كتبنا المتغيرات جميعها بكلمة واحد فقط للتعريف وهي Dim

5 - تعريف متغيرات من غير نوع في سطر واحد ونسب القيم لها مثل :-

```
Dim name as string = " hassan " , age as integer = 12 , elec as boolean = true
```

6 - تعريف متغير من دون ذكر الكلمة Dim مثل :-

```
Name = " ali "
```

7- باستخدام الكلمة المحجوزة static :

```
Static  name as string
```

سيبقى البرنامج محتفظاً بها داخل البرنامج من لحظة تشغيله إلى لحظة ايقافه ، المعلومات المستخدمة لاضافة المتغيرات باستخدام الامر (Dim) لاتختلف عن طرق اضافة المتغيرات في الاستاتيك (static)

مع التنبيه ان الكلمة الاولى تكون اخف على الذاكرة من الكلمة الثانية لان الاولى تحتفظ بالقيمة داخل اجراء منفصل بها والثانية تحتفظ بها طيلة وقت البرنامج.

المتغيرات والمساحات التخزينية التي تأخذها من الذاكرة توضحها الجداول التالي :-

أنواع البيانات التي يمكن أن تخزن أرقام عشرية :

نوع البيان	الحجم الذى يشغله فى الذاكرة بالبايت
single	4 Byte
double	8 Byte
decimal	16 Byte

أنواع البيانات التي يمكن أن تخزن أرقام صحيحة :

نوع البيان	الحجم الذى يشغله فى الذاكرة بالبايت
Byte	1 Byte
Short	2 Byte
Integer	4 Byte
Long	8 Byte
decimal	16 Byte

أنواع البيانات الحرفية:

نوع البيان	الحجم الذى يشغله فى الذاكرة بالبايت
char	2 Byte
string	10+(2*عدد الحروف)

أنواع البيانات الأخرى:

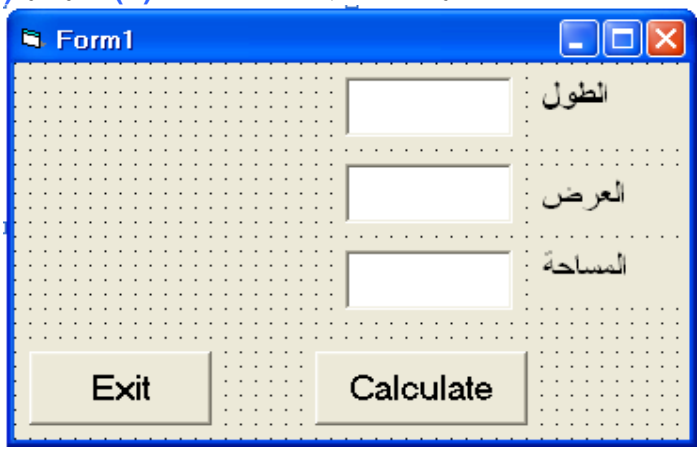
نوع البيان	الحجم الذى يشغله فى الذاكرة بالبايت
date	8 Byte
boolean	2 Byte

الجدول التالي يوضح وصف و طرق تعريف المتغيرات

النوع	الوصف	طريقة التعريف	مثال على المتغير و قيمته
Integer	عدد صحيح اعتيادي يتراوح من 32768-الى 32767	Dim x as integer	X= 15
Long	عدد صحيح طويل اكبر من 32767او اصغر من -32768	Dim b as long	B = 335674855
single	عدد عشري ذو دقة اعتيادية	Dim y as single	Y= 10.50
Double	عدد عشري ذو دقة مضاعفة	Dim AA as double	AA= 10.1234567
String	متغير حرفي (للحروف فقط)	Dim name as string	Name1 = "ali"
Date	تاريخ	Dim DT as Date	DT = #8-1-1982#
Boolean	متغير منطقي	Dim Y as boolean	Y = true
Variant	كل انواع المعطيات المتغيرة	Dim V as variant	V = 82 V = "college"
Currency	قيم مبالغ الدولارات والإشارة $	Dim C as curtency	O = 55.4
Byte	نوع البيانات الرقمية الموجبة الصحيحة فقط	Dim DG as byte	my Byte = 255

مثال :-

قم بتصميم الواجهة ثم كتابة الشفرة لحساب مساحة قطعة أرض مستطيلة الشكل أبعادها (L) كطول و (W) كعرض .



اولا :-

اضف الادوات(text)عدد (3)

(label) عدد 3 وسمهم (الطول ، العرض ، المساحة)

واداة (command) عدد (2)(وسمهم (exit ,calculate

مرحلة كتابة الكود

Text3.text = val (Text1.text * Text2.text)

(Val) : تستخدم لتحويل القيمة الحرفية الى قيمة عددية .

(Cstr) : تستخدم لتحويل القيمة العددية الى قيمة حرفية .

شرح الكود

يقوم بضرب محتوى ال (Text1 و Text2) ويضع الناتج في Text3 .

ونلاحظ بان القيم متغير دائما وهذا مايسمى بـ (المتغير) لان المعلومة يتغير باستمرار .

مثال :- صمم الشكل التالي عند تشغيل البرنامج ؟

الحل :-

```
Private Sub Command1_Click()
i = i + 1
Print i
End Sub
```

مثال :- صمم الشكل التالي عند تشغيل البرنامج ؟

الحل :-

```
Private Sub Command1_Click()
Static i As Integer
i = i + 1
Print i
End Sub
```

استخدام عبارة (Option explicit statement)

تستعمل جملة Option explicit لكي يعرض الفيجوال بيسك رسالة خطأ كلما وجد متغير لم يتم تصريحه علنا اي تعريفه في الكود اي يجب الاعلان عن جميع المتغيرات قبل استخدامها من خلال الكلمة dim .

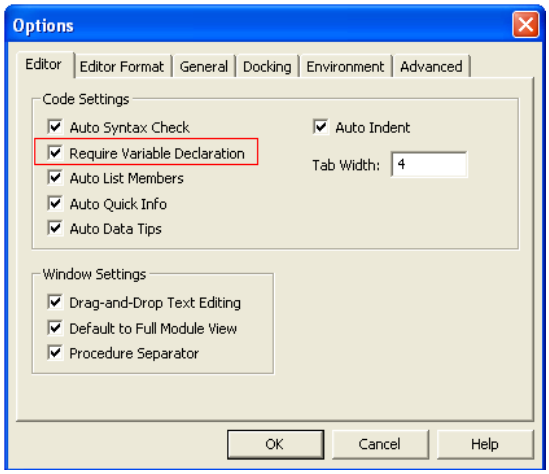
توضيح اكثر:
في التعريف التالي للمتغير

```
Dim eng
Eng = "hassan"
```

في حالة عدم استخدام جملة Option explicit فان البرنامج سيعتبر الكلمة (eng) الموجودة في السطرين متغيران وهذا يعني عدم حفظ المتغير في الموقع المحدد وحجز موقعين لنفس المتغير وهذا لا يوفر مساحة في الذاكرة كذلك فان تمرير المتغيرات لن يتم كما كنت تقصد ، اما من خلال جملة Option explicit فيستعرض المترجم compiler على المتغير eng على انه متغير غير معرف سابقا وهنا سوف ينبهك بوجود خطأ .
للحصول على عبارة Option explicit في المشروع نتبع الاتي :-

tools►option►editor

ونختار مربع الاختيار require variable declaration



عند تنفيذ الخطوات السابقة وفتح نافذة الكود نلاحظ ظهور عبارة Option explicit في منطقة الكود البرمجي العام general .

بنية الفيجوال بيسك

تتكون لغة الفيجوال بيسك من مجموعة من التعليمات مكتوبة في أسطر وتعليمة الفيجوال بيسك تتكون من :

1- الكلمات المحجوزة للفيجوال للبيسك والخاصة بتعليمة معينة : مثلاً (End و If وFor)

2- المحارف والمتحولات والثوابت :

أولا_ المحارف

تستخدم لغة البيسك ثلاثة أنواع من المحارف هي:

A - الأحرف الأنكليزية الصغيرة والكبيرة

B - الأرقام من 0 الى 9

C - محارف خاصة متعددة الأنواع ومنها كلتالي :-

استخدام عبارات التحكم في Visual Basic

1- العمليات الحسابية Arithmetic operations

تستخدم لغة Visual Basic العمليات الحسابية التالية:

المعامل	المعنى	مثال
+	الجمع	A = 5 B= 6.5 C = A + B ==> 11.5
-	الطرح	D = A – B ==> -1.5
*	الضرب	E = A * B ==> 32.5
/	القسمة	F = 10 / 3 ==> 3.33
\	الناتج الصحيح من القسمة (Integer division)	G = 10 \ 3 ==> 3
Mod	باقي القسمة (Modulo arithmetic)	Mod 10 ==> 33H = 1
^	الأس	I = 2^3 ==> 8

مثال :- حول التعابير الجبرية التالية إلى تعابير بلغة بيسك :

التعبير الحسابي بلغة بيسك	التعبير بالصيغة الرياضية
3 * A – A * B	3A – AB
A * X ^ 2 + B * X + C	$AX^2 + BX + C$
A / (5 * B)	$\frac{A2}{5B}$
5 * B / A	$\frac{5B}{A}$
(A + B) / (3 * D)	$\frac{A + B}{3D}$
A ^ 2 / B ^ 4 – C ^ 5 / (X - Y)	$\frac{A^2}{B^2} - \frac{C^2}{X - Y}$

أولويات تنفيذ المعاملات

المعامل	المعنى	الترتيب
( )	الأقواس	1
^	الأس	2
-	الإشارة السالبة (Negation) مثال:- 5 * - 3 = -15	3
* ، /	الضرب والقسمة	4
\	الناتج الصحيح من القسمة	5
Mod	باقي القسمة (Modulo arithmetic)	6
+، -	الجمع و الطرح	7
=, < >, <, >, <=, >=	العوامل العلاقية (المقارنة)	8
Not	عكس	9
And	و	10
Or	أو	11

ملاحظة: عند تساوي الأولوية تكون الأفضلية للمعامل الموجود على اليسار.

ملاحظة : - ركز على اولوية تنفيذ العمليات المبينة باللون الاحمر .

مثال:- مانتيجة تنفيذ العمليات الحسابية في التعبير الحسابي التالي ؟

EX) X = 3 * (8 - 6) + 4 ^ 2
SO)
X = 3 * 2 + 4 ^ 2
X = 3 * 2 + 16
X = 6 + 16
X = 22

مثال :- مانتيجة تنفيذ العمليات الحسابية في التعبير الحسابي التالي ؟

EX) X = 8 – 3 * 2 + 4 / 2 ^ 2
SO)
X = 8 – 3 * 2 + 4 / 4
X = 8 – 6 + 4 / 4
X = 8 – 6 + 1
X = 2 + 1
X = 3

مثال:- مانتيجة تنفيذ العمليات الحسابية في التعبير الحسابي التالي ؟

EX) X = (6 - 1) * 2 – 7 + 4
SO)
X = 5 * 2 – 7 + 4
X = 10 – 7 + 4
X = 3 + 4
X = 7

مثال:- مانتيجة تنفيذ العمليات الحسابية في التعبير الحسابي التالي ؟

EX) X = (3 + 2) * 2 ^ 2
SO)
X = 5 * 2 ^ 2
X = 5 * 4
X = 20

أمثلة:- أنجز العمليات التالية ؟

EX)4 + 3 * 2 = 10
EX) (4 + 3)*2 = 14
EX)4+8/2*3^2 = 40
EX)100 \ 30 = 3
EX)100 Mod 30 = 10
EX) (8 < > 3) Or (5 + 3 >= 8) = True

EX)Not((8 Mod 2) = 0) = False
EX) (8 < > 3) And (5 + 3 > = 8) = True
EX)2 * 5 ^ 2 + 3 * 7 Mod 5 = 5
EX)Not (3 < > 13 Mod 10) Or (4 >= 32 \ 8) = True

ملاحظه هامه جدا: دائما عدد خطوات الحل يساوي عدد العمليات الحسابية وهي خطوه تنبئ عن مدى صحة الحل .

2- عمليات المقارنة Comparison operations

تستخدم لغة Visual Basic عمليات المقارنة التالية:-

المعامل	المعنى	مثال
=	يساوي	3 = 3 ==> True
<>	لا يساوي	3 <> 3 ==> True
>	أكبر من	3 > 3 ==> False
<	أقل من	3 < 2 ==> False
>=	أكبر من أو يساوي	3 >= 4 ==> False
<=	أقل من أو يساوي	3 <= 3 ==> True

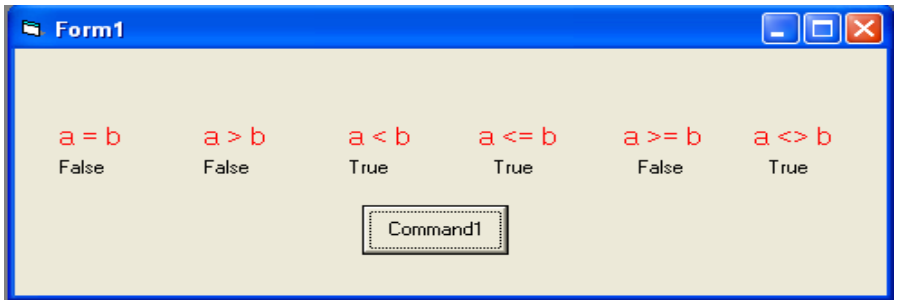
مثال:- مانتيجة عمليات المقارنة التالية إذا علمت أن (A = 1) , (B = 5)

مثال	نتيجة المقارنة
A = B	False
B = A + 4	True
A < > 1	False
A < B	True
A < 0	False
A < = 3	True
B < = 3	False
A > B	False
A > 0	True
B > = 5	True
B > = 6	False

مثال :- اضف الاداة label عدد 6 واداة command عدد 1 .
الحل :-

```
Private Sub Command1_Click()
Dim a As Integer
Dim b As Integer
Dim a1 As Boolean
Dim b1 As Boolean
Dim c1 As Boolean
Dim e1 As Boolean
Dim f1 As Boolean
a = 54
b = 80
Label1.Caption = a = b
Label2.Caption = a > b
Label3.Caption = a < b
Label4.Caption = a <= b
Label5.Caption = a >= b
Label6.Caption = a <> b
End Sub
```

اضغط f5 لتشغيل البرنامج .



3- العمليات المنطقية Boolean operations

تستخدم لغة Visual Basic عديد العوامل المنطقية أهمها:

المعامل	المعنى	مثال
Not	لا (عكس)	Not (3 = 3) ==> False
And	و	(3>2) And (5<6) ==> True
Or	أو	(3<2) Or (5<6) ==> True

كيفية استخدام المعاملات المنطقية الثلاث المذكورة TABLE

A	B	A AND B	A OR B	NOT A
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

العامل Not :-

العامل (Not) يطبق عملية نفي منطقي على المعامل الممرر إليه، لذلك فهو يعيد نفيا منطقيا لمعامله فإن كان المعامل يشير إلى (true) فإن (Not) سيعيد (false) والعكس صحيح .

مثال :- المثال التالي سيعطي (x) القيمة (false) و (y) القيمة (true)
الحل :-

```
Dim x , y As Boolean
x =  Not  23 > 14     false
y =  Not  23 > 67     true
```

العامل And :-

يقوم بعملية ربط منطقي بين جملتين منطقيتين إذا كانت الجملتان تشيران إلى (True) فإن (And) سيعيد (true) وإن كان على الأقل أحد الجملتين خاطئا سيعبد (false).

مثال :- المثال التالي سيعين قيمة (a) إلى (true) و (b) على (false)
الحل :-

```
Dim a, b As Boolean
a =  23 >14  And 11 > 8       true
b =  14 >23 And 11 > 8        false
```

العامل Or :-

يقوم بعملية فصل منطقي بين جملتين فإن كان أحد أو كلا الجملتين يشير إلى (true) فإن (Or) ستعيد (True) أما إن كانت كلا الجملتين تشيران إلى (False) فإن (Or) ستعيد (False).

مثال :-

```
Dim c, d As Boolean
c = 23 >14  Or  8  >11      true
d = 23 >67  Or  8  >11      false
```

العامل Xor :-

يقوم بعملية استبعاد منطقي على جملتين، فإن كان أحدهما صحيحاً والآخر خاطئا يعيد (true) أما إن كان كلاهما صحيحا أم خاطئا يعيد (False) .

مثال :-

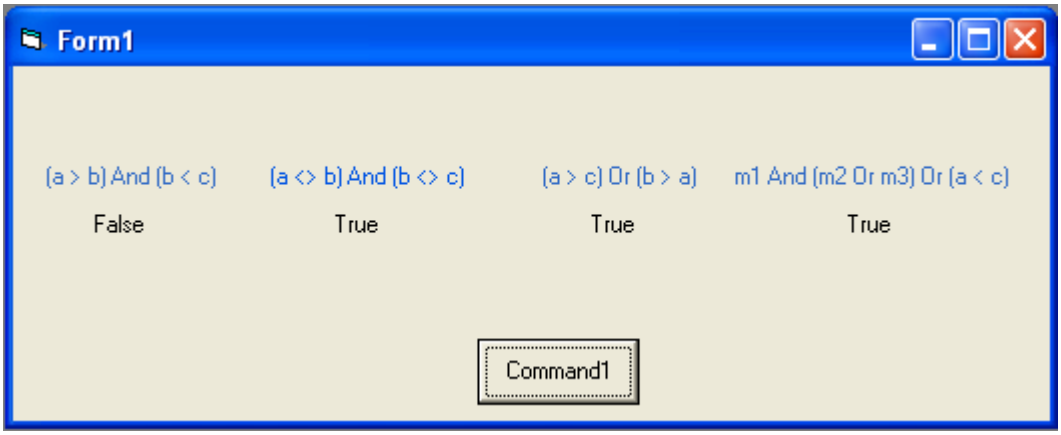
```
Dim e, f, g As Boolean
e =  23 >67    Xor  11 > 8    true
f  =  23 >14   Xor  11  >8    false
g = 14 >23   Xor    8 >11     false
```

مثال عام عن دمج عمليات المقارنة و العمليات المنطقية :- اضف الاداة label عدد 4 واداة command عدد 1 .

الحل :-

```
Private Sub Command1_Click()
Dim a As Integer
Dim b As Integer
Dim c As Integer
Dim d As Integer
Dim m1 As Boolean
Dim m2 As Boolean
Dim m3 As Boolean
Dim m4 As Boolean
a = 50
b = 70
c = 80
d = 90
Label1.Caption = (a > b) And (b < c)
Label2.Caption = (a <> b) And (b <> c)
Label3.Caption = (a > c) Or (b > a)
Label4.Caption = m1 And (m2 Or m3) Or (a < c)
End Sub
```

اضغط f5 لتشغيل البرنامج .



علامات الترقيم

وهي الفاصلة(،) والفاصلة المنقوطة(;) والنقطة(.) والنقطتان فوق بعضهما(:) وإشارة الإستفهام(؟) وإشارة التعجب(!) والفراغ( ) والأقواس () وعلامات التنصيص (') ورموز أخرى مثل ($) وإشارة النسبة (%) وإشارة العطف (&) ورمز الإعداد (#) والشخطة (_) و علامة آت (@)والخط المائل(/)

توضيح اكثر

احدى طرق كتابة نوع المتغير هو بوضع علامة كما موضح في في الجدول:-

الرمز	نوع التتغير
%	Integer
&	Long
!	Single
#	Double
@	Currency
$	String

مثال :- لتعريف نوع المتغير تستخدم الرموز المحجوزة لكل نوع من المتغيرات .

الحل :-

```
Dim Input1  $
Static Av  %
 Private Total  !
```

المصفوفات

عبارة عن متغيرات مفهرسة Indexed Variables تحتوي على بيانات عديدة من نفس النوع Data Type .
كل مصفوفة لها اسم واحد يمكن استخدامه للرجوع إلى أي عنصر فيها وذلك باقتران هذا الاسم بدليل يمثل مكان العنصر فيها ، ويمكن انشاء مصفوفة لإحتواء أي نوع من أنواع البيانات مثل ، النصوص والأعداد الحقيقية و الصحيحة وغيرها .

ماهى المصفوفة: تعرفنا من قبل على المتغيرات وعرفنا ان المتغير يعتبر وعاء يحتوى على قيمة والان نتعرف على المصفوفات والتى يمكن تعريفها على انها وعاء يحتوى على العديد من المتغيرات اى ان الفرق بينها وبين المتغير انها يمكن ان تحتوى على العديد من القيم بينما يحتوى المتغير على قيمة واحدة فقط.

إليك توضيح أكثر ، إذا أردت تصميم برنامج لشئون الموظفين بدون استخدام المصفوفات وبفرض أن عدد الموظفين (100) موظف فان الحل باستخدام المتغيرات العادية يتطلب الإعلان عن (100)متغير ثم تكرار الأوامر التي تتعامل مع الموظفين المائة ولا شك أن هذه مسألة شاقة ومطولة.
أما الحل الامثل في هذه الحالة فهو استخدام مصفوفة تتكون من (100) عنصر و الصيغة التي تحقق هذا الغرض لهذه المصفوفة كما يلي:

```
Private Names(99)  As String
```

الخصائص الأساسية للمصفوفة في الفيجيوال بيسك

* اسم المصفوفة يمثل عنوان Address في الذاكرة ولا يمكن تغييره أثناء تنفيذ البرنامج .
* يمكن الإعلان عن مصفوفة لأي نوع من أنواع البيانات بما في ذلك الأنواع المعرفة من قبل المستخدم User-defined type والـ Object Variables .
* كل وحدة بيانات منفردة في المصفوفة تسمى عنصر Element .
* جميع العناصر تكون من نفس النوع إلا في حالة الإعلان عن المصفوفة كـ Variant Data Type .
* جميع العناصر تكون مخزنة على التتابع في ذاكرة الحاسوب ودليل أول عنصر هو الصفر كـ Default ، ويمكن جعله 1 باستخدام جملة Option Base .
* يحدد موقع العنصر بواسطة رقم الصف ورقم العمود الذي يقع فيه .
* لكل مصفوفة حد أعلى Upper bound ، وحد أدنى Lower bound ؛ وعناصر المصفوفة تكون محصورة بين هذين الحدين .
* من الممكن أن تكون المصفوفة ذات بعد واحد أو متعددة الأبعاد .

أنواع المصفوفات

1- المصفوفات احادية البعد .
2- المصفوفات الثنائية البعد .
3- المصفوفات متعددة الابعاد .

المصفوفات الاحادية البعد

يقصد بكلمة بُعد هو عدد الدلائل (Indexes) المستخدمة للرجوع إلى عنصر في المصفوفة, تستخدم المصفوفات متعددة الأبعاد لتخزين بيانات مرتبطة ببعضها البعض.

تتكون المصفوفة الاحادية البعد من عمود واحد وعدة صفوف حسب العناصر المراد اضافتها للمصفوفة ويرمز الرقم الموجود بين الاقواس عند تعريف المصفوفة الاحادية الى عدد صفوف العناصر .

الإعلان عن المصفوفات الاحادية

هناك ثلاث طرق للإعلان عن هذا النوع تعتمد على النطاق scope الذي سوف تغطيه المصفوفة :-

* للإعلان عن مصفوفة عامة Puplic Array تستخدم جملة Puplic في قسم الإعلان Declaration Section للوحدة Module.
* للإعلان عن Module-Level Array تستخدم جملة Private في قسم الإعلان الخاص بـ Procedure.
* للإعلان عن Local Array تستخدم جملة Dim في الـ Procedure.
* تحديد الحدين الأعلى والأدنى للمصفوفة Upper bound & Lower bound:
* عند الإعلان عن مصفوفة، يكتب الحد الأعلى بعد الاسم وبين الأقواس.
* لا يمكن أن يزيد الحد الأعلى عن نطاق نوع المتغير Long Data Type.
* الحد الأدنى الإفتراضي Default هو الصفر.
* لتحديد الحد الأدنى، ينبغي كتابته صراحة باستخدام كلمة To.

الصيغة العامة لتعريف المصفوفة الاحادية :

```
Dim Arrayname (N) As type
```

وكذالك يمكن تعريف المصفوفة الاحادية على النحو التالي :

```
Dim Arrayname (F to L) As type
```

Dim : هي جمله تعريف المتغيرات
Arrayname : اسم المصفوفة
N : عدد عناصر المصفوفة من (0 الى N)
Type : تحديد نوع عناصر المصفوفة
F :بداية عدد العناصر
L : نهاية عدد العناصر

مثال :-
إعلان عن مصفوفة Student يمكن وضعه في جزء الأعلان الخاص بالـ Module أو داخل الـ Procedure:

```
Dim Student (1000) As Integer
```

وبذلك يكون عدد عناصر هذه المصفوفة 1001 عنصر، ولجعلها مصفوفة عامة، نستبدل Dim بـ Public كالتالي:-

```
Public Student (1000) As Integer
```

ولتحديد الحد الأدنى لهذه المصفوفة بـ1 وبالتالي يصبح عدد عناصرها = 1000 عنصر فقط، نكتب To كالتالي:-

```
Public Student (1 To 1000) As Integer
```

مثال:-
الإعلان عن مصفوفة بها 51 عنصر مرقمة من 0 إلى 50 من دون تحديد نوع بيانات عناصرها، سيعتبر النوع Variants مباشرة :-

```
Dim DayArray (50)
```

مثال:-
الإعلان عن المتغير BirthDate على أنه مصفوفة من التواريخ المفهرسة من 1 حتى 10:-

```
Dim BirthDate (1 To 10) As Date
```

مثال :- تعريف مصفوفة مكونة من خمسة ارقام :-
الحل :-

```
Dim eng (1 to 5) As integer
```

سوف يبدء الترقيم لفهرس eng امصفوفة من الرقم 1 الى الرقم 5

المصفوفة	قيمته
Eng (1)	3
Eng (3)	4
Eng (3)	55
Eng (4)	52
Eng (5)	99

شرح المصفوفة :-
كل عنصر من عناصر المصفوفة يقابل الرقم الموجود مابداخلها .

المصفوفات الثنائية البعد

عند تعريف المصفوفة ثنائية البعد فلا بد من تحديد كلا من عدد الاعمدة و عدد الصفوف وفي حالة المصفوفة الثنائية فان عدد الاعمدة يساوي (2) .

الصيغة العامة لتعريف المصفوفة الثنائية :

```
Dim arrayname (I,J) as type
```

I : عدد الصفوف .
J : عدد الاعمدة .

مثال :- اكتب الكود اللازم لتعريف مصفوفة مكونة من اسماء ومعدلات خمس طلاب ثم صباغة العناصر على الفورم ؟

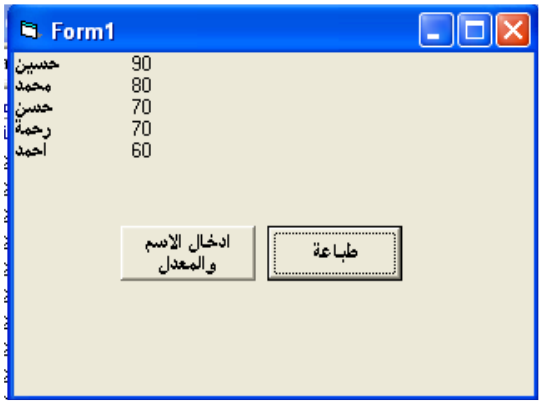
```
Dim eng(4, 1) As Variant

Private Sub Command1_Click()
eng(0, 0) = InputBox (" ادخل الاسم الاول ")
eng(0, 1) = InputBox("ادخل معدل الطالب الاول")
eng(1, 0) = InputBox(" ادخل الاسم الثاني ")
eng(1, 1) = InputBox("ادخل معدل الطالب الثاني")
```

```
eng(2, 0) = InputBox(" ادخل الاسم الثالث ")
eng(2, 1) = InputBox("ادخل معدل الطالب الثالث")
eng(3, 0) = InputBox(" ادخل الاسم الرابع ")
eng(3, 1) = InputBox("ادخل معدل الطالب الرابع")
eng(4, 0) = InputBox(" ادخل الاسم الخامس ")
eng(4, 1) = InputBox("ادخل معدل الطالب الخامس")
End Sub

Private Sub Command2_Click()
Print eng(0, 0),
Print eng(0, 1)
Print eng(1, 0),
Print eng(1, 1)
Print eng(2, 0),
Print eng(2, 1)
Print eng(3, 0),
Print eng(3, 1)
Print eng(4, 0),
Print eng(4, 1)
End Sub
```

اضغط F5 لتشغيل البرنامج .



لاحظ ان عدد الصفوف يساوي 5 وعدد الاعمدة يساوي 2 وفي حالة زيادة عدد الاعمدة اكثر من 2 تعتبر المصفوفة مصفوفة متعددة الابعاد .

المصفوفات متعددة الأبعاد

هي تلك المصفوفة التي يزيد عدد الاعمدة لها عن اثنين .

مثال :- نفس المثال السابق بحيث يكون عدد الصفوف يساوي (5) وعدد الاعمدة يساوي (5) فيكون شكل المصفوفة بعد التنفيذ .

توضيح على ذلك:-

تخزين كل نقطة Pixel موجودة على شاشة الحاسوب نحتاج لتخزين إحداثياتها x & y وهذا ممكن باستخدام مصفوفة ذات بعدين.

يمكن معرفة أبعاد مصفوفة بالنظر إلى تعريفها؛ حيث يتم الإعلان عن مصفوفة متعددة الأبعاد بذكر قيمة لأبعادها بعد اسمها مباشرة وداخل الأقواس، يتم فصل قيمة كل بعد بالفاصلة، سواء كانت ذات بعدين أو ثلاثة أبعاد .

```
Static A(3, 4) As Double
Static A(1 To 10, 1 To 10) As Integer
Dim B (3, 1 To 10, 1 To 15)
```

ملاحظة:-

* عدد عناصر المصفوفة متعددة الأبعاد يمكن معرفته بضرب قيم الأبعاد ببعضها.
* عند إضافة أبعاد المصفوفة فإن مساحة التخزين المطلوبة سوف تزيد زيادة كبيرة ولذلك ينبغي الاحتراس وتفادي استخدام النوع Variant قدر الإمكان لما يتطلبه من مساحة تخزينية كبيرة .

استخدام الـ Loops لمعالجة المصفوفات

يمكن بكفاءة وسهولة معالجة المصفوفات باستخدام الـ Loops وخصوصاً المصفوفات متعددة الأبعاد حيث يتم معالجتها باستخدام Loops متداخلة.

مثال:-

الجمل التالية سوف تعطي قيمة لكل عنصر من عناصر المصفوفة A هذه القيمة تكون مرتبطة بموقع العنصر داخل المصفوفة كما سنرى:

```
Dim I As Integer, J As Integer
Static A(1 To 10, 1 To 10) As Integer
For I=1 To 10
For J=1 To 10
A(I, J)= I*10*J
Next J
Next I
```

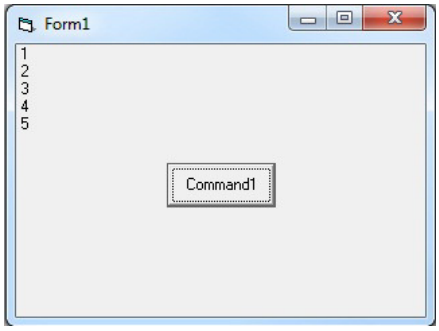
استخدام جمل الدوران مع المصفوفات

استخدام ال (loops)لمعالجة المصفوفات :-

يمكن بكفاءة وسهولة معالجة المصفوفات باستخدام ال (loops)وخصوصا المصفوفات متعددة الابعاد حيث يتم معالجتها باستخدام (loops) متداخلة .

مثال :- نفذ البرنامج التالي:

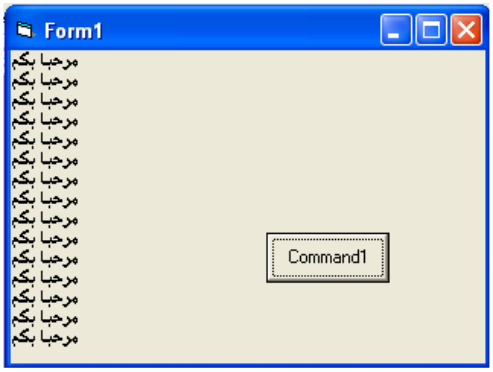
```
Private Sub Command1_Click()
For M = 1 To 5
Print M
Next
End Sub
```



مثال :- نفذ البرنامج التالي:

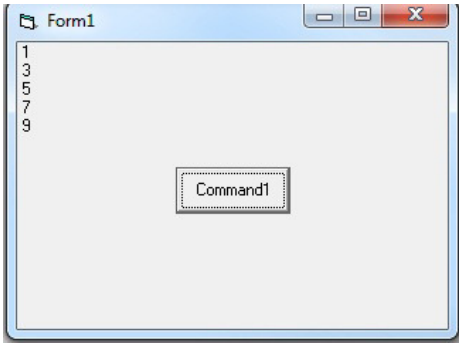
```
Private Sub Command1_Click()
For M = 1 To 5
```

```
Print "مرحبا بكم"
Next
End Sub
```



مثال :- هنا نستخدم الخطوة STEP اي قم بلعد من 1 الى 10 بخطوتين

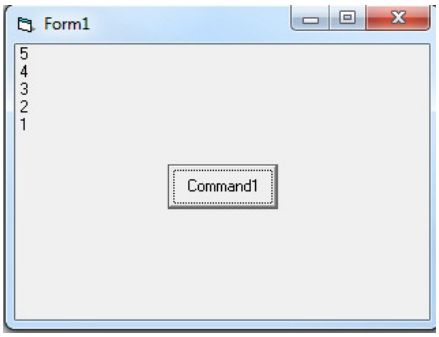
```
Dim M As Integer
Private Sub Command1_Click()
For M = 1 To 10 Step 2
Print M
Next
End Sub
```



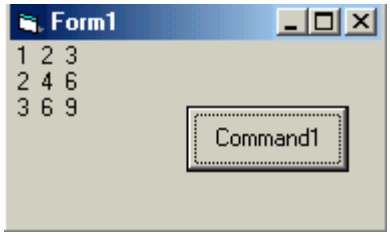
ملاحظة :- من خلال الكلمة step يمكن زيادة الدوران بالعدد المحدد ، اذا لم نستخدم الكلمة step بذلك نزيد الدوران بمقدار واحد .

مثال :- نفذ البرنامج التالي:-

```
Private Sub Command1_Click()
For M = 5 To 1 Step -1
Print M
Next
End Sub
```



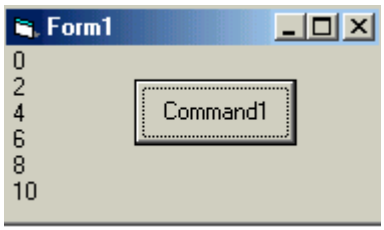
مثال :- اكتب الكود اللازم لشكل الفورم ؟



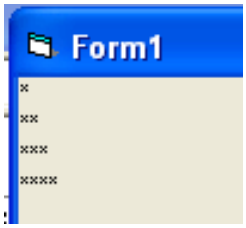
```
Private Sub Command1_Click()
Dim a(1 To 3, 1 To 3) As Integer
Dim i As Integer, j As Integer
For i = 1 To 3
For j = 1 To 3
a(i, j) = i * j
Print a(i, j); "";
Next j
Print
Next I
End Sub
```

مثال :- ماهو حاصل الضرب للبرنامج التالي ؟

```
Option Explicit
Dim a(5) As Integer
Dim I As Integer
Private Sub Command1_Click()
For I = 0 To 5
 a(I) = I * 2
 Print a(I)
Next
End Sub
```



مثال :- اكتب جمل الدوران لطباعة شكل النجمة كما مبين من الشكل ؟



الحل :-

```
Dim i  As Integer
Dim J As Integer
Private Sub Command1_Click()
For i = 1 To 4
 For J = 1 To i
Print "*";
Next j
Print
Next i
End Sub
```

مصفوفة التحكم (control array)

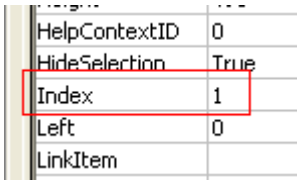
هي مصفوفة من احدى ادوات التحكم تظهر من خلال رسم الاداة على النموذج ومن ثم نسخها و لصقها على نفس النموذج من جديد حيث تظهر عبارة للتاكد من ان المبرمج يرغب بتعريف مصفوفة تحكم ام لا فاذا تم اختيار yes يتم نسخ اداة التحكم .

خطوات تعريف مصفوفة التحكم من نوع text

1- اضف مربع نص الى النموذج فيظهر باسم text1
2- انسخ مربع النص
3- الصق مربع النص على نفس النموذج
4- ستظهر لك نافذة عند اختيار yes سوف يقوم بلصق مربع النص على نفس النموذج بنفس الاسم text1 ولكن بفهرس مختلف index على النحو التالي.

Text1(0)
Text1 (1)

للتاكد من رقم الفهرس انظر الى خصائص مربع النص .

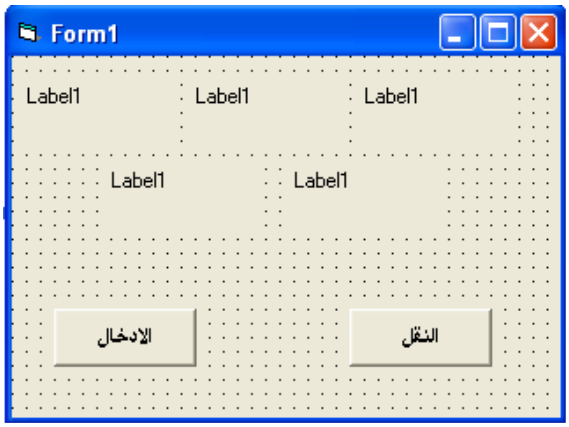


مثال :-

لتعريف مصفوفة مكونة من 5 اسماء ومن ثم ادخال الاسماء ونقلها الى مصفوفة تحكم من النوع label كما في الشكل التالي ؟

الحل :-

اضف اداة label الى النموذج وانسخ الاداة كما ذكرنا سابقا اربعة مرات مكزنا بذلك مصفوفة تحكم تبدء من label (0) الى label (4) وتظهر جميعها على الفورة كالتالي .



كتابة الكود :-

```
Dim eng (4) as string
Dim I,j as intrger

Private Sub Command1_Click()
For I = 0 to 4
Eng (i) = inputbox(" ادخل الاسم")
Next I
End Sub

Private Sub Command2_Click()
For j = 0 to 4
Label (j) = eng (j)
Next j
End Sub
```

لتشغيل البرنامج اضغط F5.

Module

ماهو المديول (Module)
الموديول هى شاشه لكتابه الأكواد العامه والتصريحات

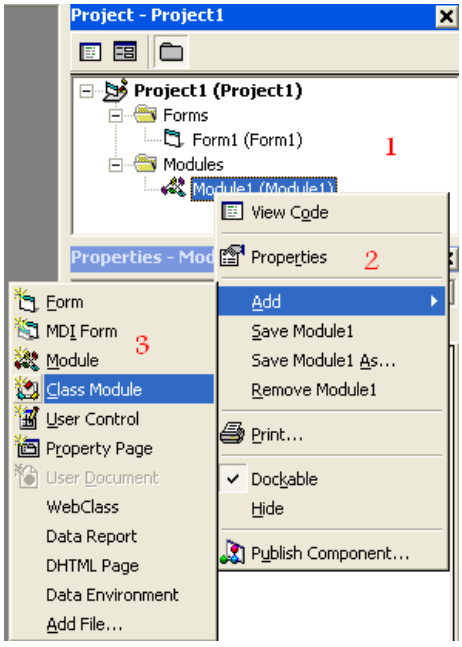
* لماذا نستخدم المديول (Module) ؟
نستخدم الموديول عندما نحتاج لتعريف متغير او دالة او اجراء في جميع المشروع و لا يكون مقتصرا على فورم واحد فقط .
* ما الفرق بين تعريف متغير في المديول (Module) او في الفورم ؟
عندما اريد تعريف المتغير (X)في الفورم اتعامل معه داخل الفورم فقط و عندما اعرفه في الموديول استطيع استخدامه في كل المشروع باستخدام (Public) قبل المتغير يكون متاح للاستخدام في كل (Forms) بخلاف (Private)
وكذلك الحال بالنسبة للدوال (Functions) و الاجراءات(Procedures)
و الصيغة العامة للتعريف

توضيح اكثر :-
باختصار شديد ال moudle يستخدم في التخفيف على المستخد من كتابة الأكواد الكثيرة فمثلاً ماذا لو كنت تتعامل مع قاعدة بيانات وفي المشروع أكثر من 50 شاشة (forms) هل ستقوم في كل واحدة من هذه الشاشات بتعريف سلسلة الاتصال بقاعدة البيانات بالطبع لا.
وهنا تكمن فائدة وخاصية وميزة ال moudle عن غيره في كتابة أكواد تستطيع التعامل معها في كل المشروع وفي كل شاشاته.

لادراج (Module) فى مشروعك لدينا طريقتين

1- من خلال القائمة project ►add moudle

2- من خلال النقر بزر الفارة الايمن على نافذة المشروع project explorer واختيار add ►class moudule



يمكن تعريف ثلاثة انواع من الاجراءات ذات الهدف العام فى ال module

1- الاجراءات function
الصيغة العامة للاجراء

```
Public function function _ name (arguments) [as type]
function statment (VB statement / s)
end function
```

A– function: كلمة ثابتة تحدد استخدامنا لاجراء العام بهذا النوع .
B– function name : اسم الدالة التي تريد انشاءها .
C– arguments : متغيرات وسيطية يتم استعمالها في الدالة .
D– as tybe : لتحديد نوع النتيجة التي تعيدها الدالة مثلا النوع الافتراضي هو متنوع variant .

مثال :- اضف اداة ال command و text الى النموذج .
نعرف متغير في النطاق العام public variable كما يلي :

```
Public eng as string
```

في النموذج نعطي المتغير قيمة دون الحاجة لاعادة تعريفه مرة اخرى كما يلي :

```
Private Sub Command1_Click()
Eng = "hussain"
Text1.text = eng
End sub
```

شرح الكود
لاحظ اننا قمنا بتعريف المتغير في ال modul ولم نعرفه في النموذج لان عند تعريف متغير في ال modul سوف يكون عام على النماذج .

2- الاجراءات الفرعية sub :
الصيغة العامة للاجراء :-

```
Sub procedure_name(arguments)
Procedure statements
End sub
```

يتميز الاجراء الفرعي بانه لا يعيد قيمة مقترنة باسمه ويستخدم للحصول على ادخال من المستخدم او لعرض المعلومات او طباعتها او لمعالجتها بعض الخصائص المتعلقة بشرط ، كما يمكن استعمالها لمعالجة واعادة عدة متغيرات عند مناداتها وتعيد عدة قيم بعكس الاقتران الذي يعيد قيمة واحدة .

الصيغة العامة :-

```
Sub procedudre_name (arguments)
procedure ststements
```

مثال :-
من خلال نافذة ال project explorer انقر على المساحة الفارغة كليك يمين ثم

add << new moudle

واضف الكود التالي :-

```
Sub ENG()
MsgBox("هندسة الالكترونيك و السيطرة")
End Sub
```

وقم باضافة اداة command واضف الية الكود :-

```
Private Sub Command1_Click()
 Call ENG
End Sub
```

شرح الكود :-
لقد قمنا بتعريف متغير في ال module وهي كتابة كود صندوق الرسالة كما هو مبين ، واضفنا اداة الزر وقمنا بكتابة كود المناداة في الزر وعند الضغط عليه سوف يقوم بتنفيذ ماموجود في module ، وبهذا لو كان في البرنامج 40 فورم نستطيع مناداته من اي فورم كان .

ملاحظة :- وحدة الاجراءات (Module) هي ملف بالامتداد Bas تظهر تحت نماذج المشروع (بعد أضافتها من قبل المستخدم) في نافذة متصفح المشروع وتعرف أجراءات الوحدة عادة بالامر Public بدل Private لكي تكون متاحة لكل نماذج المشروع. وتصبح أداة View Object غير فعالة عند أختيار الوحدة لآن الوحدة لا يمكن عرضها مثل النوذج وانما نعرض الشفرة الخاصة بها من خلال أداة View code.

3- الدوال الجاهزة(Bulit-Function) :- سناتي الى شرحها في الصفحة (133) .

طرق الادخال والإخراج في لغة الفيجوال بيسك

1- الادخال INPUT :

نستخدم نوعين من طرق ادخال البيانات

A- Textbox
B- Inputbox

2-الاخراج output:

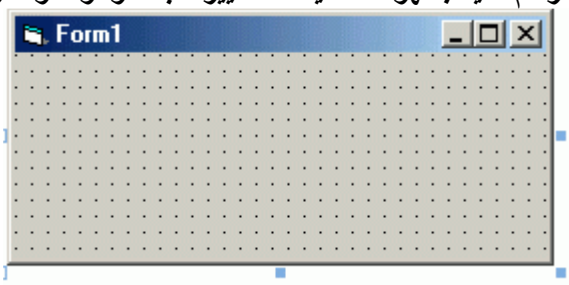
يوجد عدة طرق لاخراج البيانات

A- Textbox
B- Label
C- message box
D- print

الفورم

شكل الفورم والتعامل معه

النموذج (form1) و هو عبارة عن نافذة أو هو عبارة عن نموذج فارغ و هو الذي نكوّن و نضع عليه الرسوم و الصور و الأزرار و عناصر الإدخال و الإخراج و غير ذلك و الذي سيكون واجهة برنامجك الذي ستصنعه.
وهو حقيقة يشبه لوحة رسم لأنك تستطيع الرسم عليه بسهولة كما يمكنك تغيير حجمه و لونه و أطرافه ، كما تستطيع تغيير عنوانه ايضا .



شكل الفورم

هذا الجدول يبين شرح خصائص الفورم

اسم الخاصية	معناها
Name	اسم الفورم
Appearance	الظهور
BackColor	لون الخلفية
BorderStyle	نمط الإطار
CauseValidation	التصحيح
Caption	العنوان
ControlBox	إظهار وإخفاء الأزرار العلوية
Enabled	الفاعلية
Font	نوع الخط

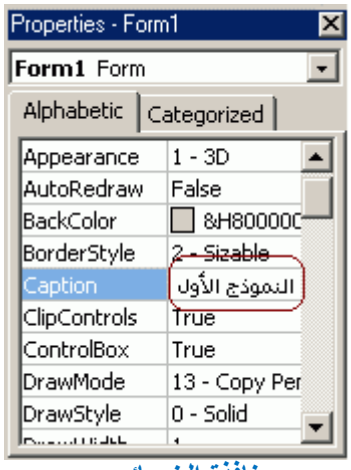
ForColor	لون الخط
Left	البعد عن الحافة اليسرى
Height	الإرتفاع
Icon	أيقونة الفورم
MaxButton	تفعيل وإلغاء زر التكبير
MDIChild	يعمل داخل فورم النافذة الأم
MinButton	تفعيل وإلغاء زر التصغير
RightToLeft	الإتجاه من اليمين لليسار
TabIndex	ترتيب Tab
MousePointer	مؤشر الماوس
MouseIcon	الأيقونة التي تشمل مؤشر الماوس
MoveAble	قابل للحركة
Picture	صورة الخلفية
StartUpPosition	موضع الفورم على الشاشة
TabStop	إيقاف ال Tab
ToolTipText	ماسيظهر عند وضع الماوس فوقها
Tag	Tag
Caption	النص
Top	البعد عن حافة النافذة العليا
Visible	مرئي أو مخفي
Width	العرض
WindowsStat	حجم الفورم عن التشغيل

ملاحظة :- الفورم السابق هو نتيجة إختيارنا في البداية لـ(standar.exe) هذا يعني أن هذا الفورم هو الفورم الإفتراضي .

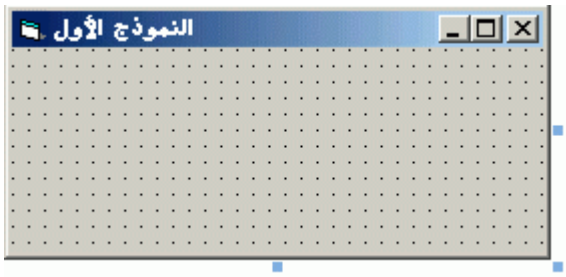
كيف نغير عنوان الفورم من كلمة (form1) إلى العنوان الذي نريده

انقر فوق الفورم نقرة واحدة (لتحدده أو تختاره) ، ثم اذهب لمربع الخصائص أو نافذة الخصائص (properties window) موقعه على يمين الشاشة إذا كان غير ظاهر على واجهة البرنامج إذهب إلى شريط الأدوات و انقر على الزر
او من القائمة view ►properties windowo
او الضغط على مفتاح F4

عندها تظهر نافذة الخصائص اختر caption و اكتب : " النموذج الأول " .



نافذة الخصائص

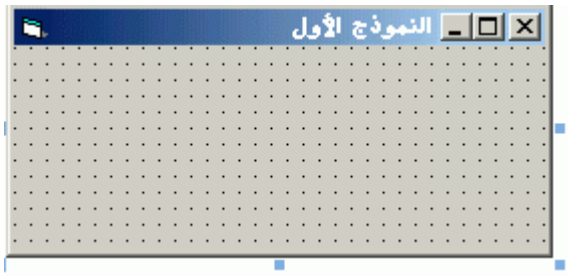


تغير عنوان الفورم انظر فوق

ملاحظة :- نافذة الخصائص تخصص جميع العناصر الموجودة على الفورم كالأزرار و عناصر الادخال والإخراج إلخ.

كيفية وضع العنوان على اليمين

في نافذة الخصائص السابقة ابحث عن rigth to left يظهر سهم منسدل للأسفل انقرعليه وستلاحظ أن لهذه الخاصية قيمتان fals و true حيث fals هي الإفتراضية غيرها إلى true .



لقد تغير عنوان الفورم من اليسار إلى اليمين

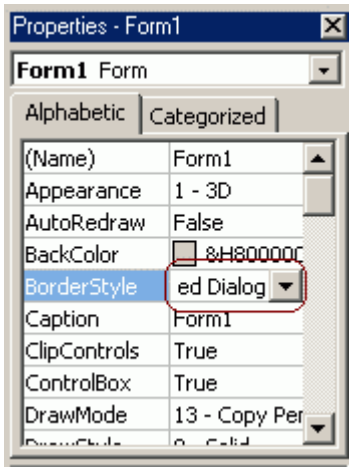
الآن تغيرت و أصبحت من اليمين إلى اليسار أليس كذلك .

كيفية تغير حجم الفورم

تستطيع تغيير حجم الفورم بطريقة سهلة ... أنت ترى في الاعلى على الفورم السابق مربعات صغيرة زرقاء اللون أو في أركان الفورم ضع مؤشر الفارة فوق أحد تلك المربعات سيتغير شكل المؤشر إلى مؤشر ذو رأسين تستطيع من خلال ذلك تغيير حجم الفورم بالسحب و الإفلات.
و هناك طريقة أخرى لتغيير حجم الفورم ...إذهب إلى نافذة الخصائص انقر فوق السهم المتجه نحو الأسفل في عمود الإنزلاق الموجود على يمين نافذة الخصائص و ابحث عن خاصية العرض Width و خاصية الطول Heigth و غير إلى المقاس الذي تريده بالتحديد
.

كيفية تغير إطار الفورم برز إغلاق فقط أو بدون أزرار من فوق؟

في مربع الخصائص سوف تجد الخاصية Border Style أنقر فوق السهم للأسفل لعرض خيارات وتغيير شكل الفورم أنظر إلى الصورة التالية :



التعامل مع النماذج (form)

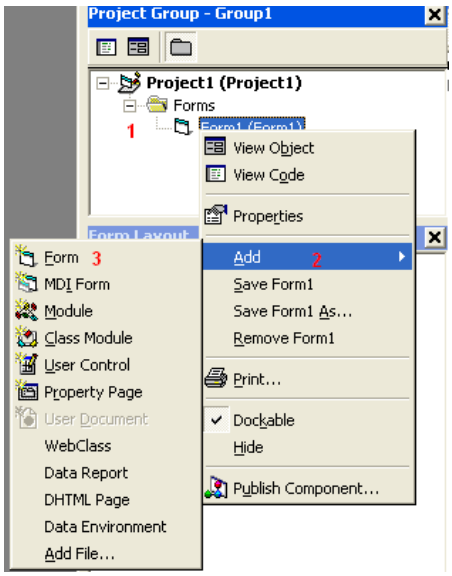
كيفية اضافة عدة فورمات للبرنامج

لو كان لديك عدة فورمات في برنامجك كيف ستظهر وتغلق النماذج .

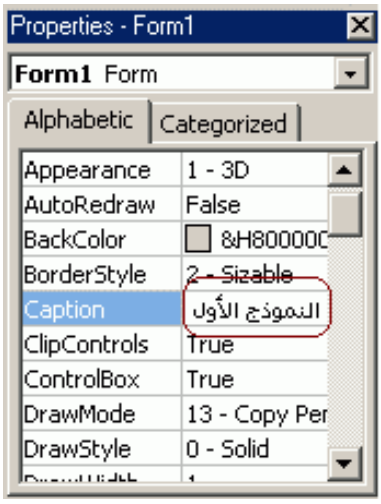
1- عن طريق project ► Add form

2- عن طريق الضغط كلك يمين على view ► project explorar

3- او file ►Add project



1 - بعد فتح الفيجوال بيسك وتجهيز الفورم الأول سوف يكون إسمه بالأساس (Form1)غير خاصية (Caption) أي العنوان إلى "النموذج الأول" .



أضف إلى الفورم زرين أمر وغير خاصية (Caption) لكل منهما

ثم أضف فورم ثاني(Form2) وغير (Caption)إلى "النموذج الثاني" عن طريق احدى الطرق المذكورة .

أضف إلى الفورم الثاني زر أمر "عودة للنموذج الأول"

في الفورم الأول "النموذج الأول" أنقر فوق الزر "إظهار النموذج الثاني " نقرتان لفتح المكان المخصص لكتابة الشيفرة
أكتب الشيفرة التالية :

```
Unload Me
Form2.Show
```

السطر الأول : يعني إغلاق الفورم الحالي أي "النموذج الأول"
السطر الثاني : يعني إظهار الفورم الثاني الذي إسمه(Form2)

وفي الفورم الثاني أنقر فوق الزر "عودة للنموذج الأول " نقرتان وأكتب نفس الكود السابق ولكن عليك تبديل (Form2) إلى (Form1)
إذاً نستنتج أن شفرة إغلاق فورم لوحده هي (Me Unload) أو Form1.Hide إستبدل (Form1) بإسم الفورم الذي تريد إخفائه

الأن عد إلى الفورم الأول وأنقر نقرتان فوق الزر " إغلاق البرنامج " وأكتب التعليمة التالية (End)

وهي تقوم بإغلاق كافة البرنامج أي كافة النماذج الموجودة
إذا تعليمة إغلاق نموذج واحد هي Me Unload وتعليمة إغلاق كافة البرنامج هي End

هذا بالنسبة للإظهار والإغلاق هناك تصغير الفورم عن طريق النقر فوق زر .
أضف إلى أحد الفورمات السابقة زر أمر قم بستميته "تصغير"

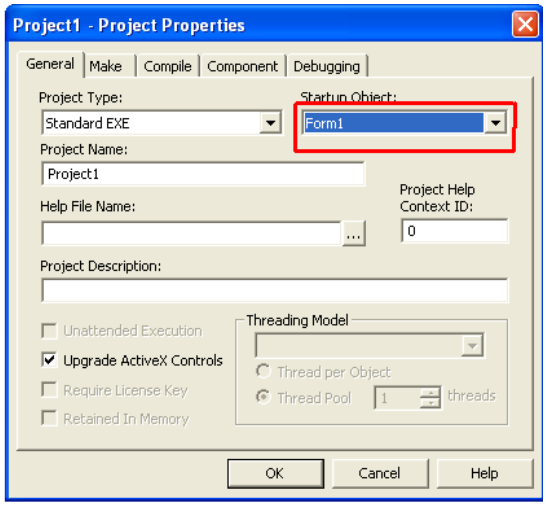
أكتب الكود التالي:

```
Form1.WindowState = 1
```

لاحظ أن (WindowsState)هي خاصية من خصائص الفورم ولها ثلاث خيارات
0 - Normal الوضع العادي
1 - Minimized أصغر حجم
2- Maximized أكبر حجم
في التعليمة السابقة لو وضعت رقم (2) فإن النموذج سوف يكبر إلى ملىء الشاشة

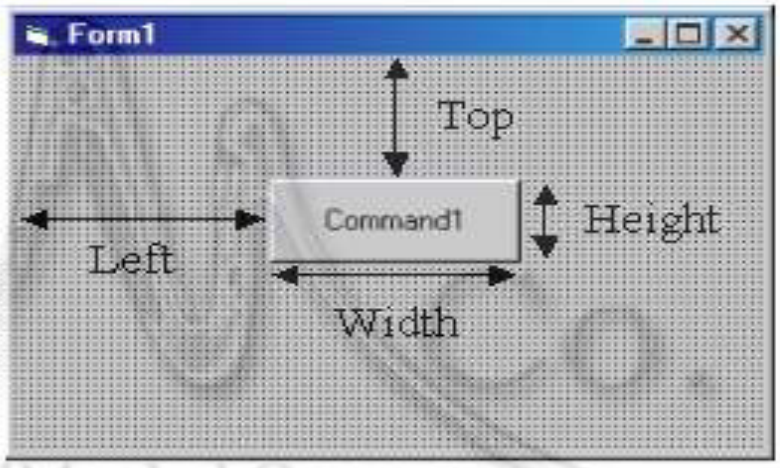
ملاحظة :-
عند اضافة فورمين للنموذج مثلا form1 & form2 فعند تشغيل البرنامج سيكون التحميل للفورم الاول دائما form1 واذا اردنا تغيير ترتيب تشغيل الفورم مثلا عند تشغيل البرنامج يتم تحميل الفورم الثاني form2 بدل الفورم الاول وهذا خطورة مهمة جدا في تغيير ترتيب التشغيل للفورم ويتم من خلال القائمة :

Project ► Project properties
سيظهر لنا نافذة ومن القائمة general ► (startrun object) نختار اسم الفورم الذي نريد تحميله اولا .



خاصية موقع الفورم

ملاحظة :- في الفيجوال بيسك يكون احداثيات ال X و Y نحو الاسفل ونقطة الاصل 0,0 يكون كما مبين من الشكل :-



خصائص الأبعاد التابعة للأداوت

اسم الخاصية	تتحكم في ...
Top	الطرف العلوي للأداة
Left	الطرف الأيسر للأداة
Height	ارتفاع الأداة
Width	عرض الأداة

لتغيير موقع الفورم نتبع الاتي :-

```
Form1.left=100
Form1.top=200
Form1.height=300
```

```
Form1.width=400
```

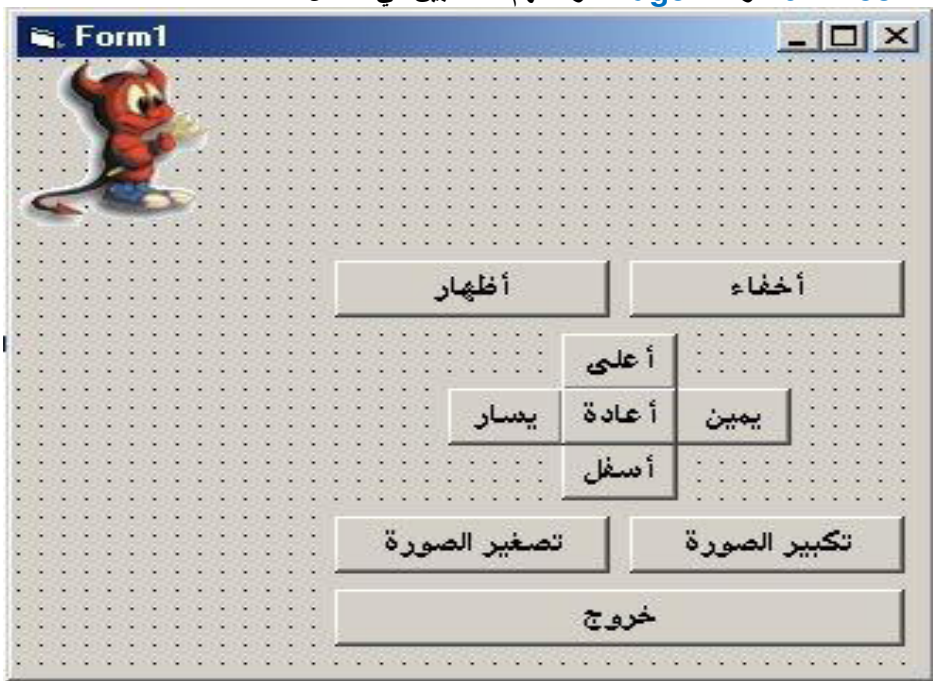
ويوجد طريقة سهلة نتجنب بها بطئ التنفيذ وكتابة اربعة اسطر من الاوامر وذلك بكتابة الامر move

```
Form1.Move 100, 200, 300, 400
```

ملاحظة :- *الحد الادنى للنوذج الافقي (left = 12000) ، والحد الادنى للنوذج العمودي (top = 9000).
*(تويب) هي وحدة قياس المسافة على النموذج .

مثال :-

نقوم باضافة اداة command عدد 10 واداة image وسمهم كما مبين في الشكل :-



نقوم بكتابة الكود الخاص لكل زر

```
Private Sub Command1_Click()
Image1.Visible = False
'اخفاء
End Sub

Private Sub Command2_Click()
Image1.Visible = True
'أظهار
End Sub

Private Sub Command3_Click()
Image1.Top = Image1.Top – 50
'أعلى
End Sub

Private Sub Command4_Click()
Image1.Left = Image1.Left + 50
'يمين
End Sub

Private Sub Command5_Click()
Image1.Top = 0
Image1.Left = 0
'أعادة
End Sub

Private Sub Command6_Click()
```

```
Image1.Left = Image1.Left – 50
'يسار
End Sub

Private Sub Command7_Click()
Image1.Top = Image1.Top + 50
'أسفل
End Sub

Private Sub Command8_Click()
Image1.Width = Image1.Width + 100
Image1.Height = Image1.Height + 100
'تكبير الصورة
End Sub

Private Sub Command9_Click()
Image1.Width = Image1.Width - 100
Image1.Height = Image1.Height – 100
'تصغير الصورة
End Sub

Private Sub Command10_Click()
End
'خروج
End S
```

تحريك الكائنات move

نقل الكائنات في نظام الاحداثيات للفورم .
تتيح لنا هذه الطريقة نقل الكائنات في نظام الاحداثيات x,y على الفورم .

والقاعدة العامة للطريقة move هي :

```
Object name . move          left , top
```

left , top : احداثيات ال x,y او

Object name : اسم الكائن الذي تريد نقله

move : نقل او تحريك

مثال :- تحريك الصورة بالاحداثيات (left = 2000) و (top = 2000) ؟

```
Picture1 . move   2000,2000
```

كيفية كتابة الشفرة :

يوجد طريقتين لكتابة الشفرة في الفيجوال بيسك :-

1 - الطريقة :- القاعدة العامة لكتابة الشفرة :

مثلا (LIST , COMBO box ,....)

```
Object . method  value
```

مثال : -

```
List1 . Additem "Iraq"
```

2 - الخاصية:- القاعدة العامة لاي شفرة بالكود في البرنامج :

مثلا (TEXT , LIBEL ,.....)

```
Object . property = value
```

مثال :-

```
libel1 . text = "welcome"
```

توضيح اكثر :

إذا أردت تنفيذ أي تغيير في خصائص أي أداة أثناء تشغيل العرض فيلزمك أن تكتب للبرنامج شفرة بذلك أي تعليمية برمجية أي (Code) و جميع الخصائص يتم تغيير خصائصها بهذه الطريقة :

اسم الأداة . اسم الخاصية = القيمة

مثال:-

إذا أردت تغيير الكتابة على الزر إلى (تم الضغط على الزر) مثلا

```
CommandButton1.Caption = " تم الضغط على الزر "
```

اسم الأداة — أساسية — خاصية التسمية — أساسية — قيمة نصية

و بالمثل : إذا أردت تغير حجم الخط إلى حجم (20) لمربع النص تكتب هذه الشفرة :

خاصية سمك (حجم) الخط

```
TextBox1. FontSize = 20
```

اسم الأداة — أساسية — خاصية الخط — أساسية — قيمة حجم الخط

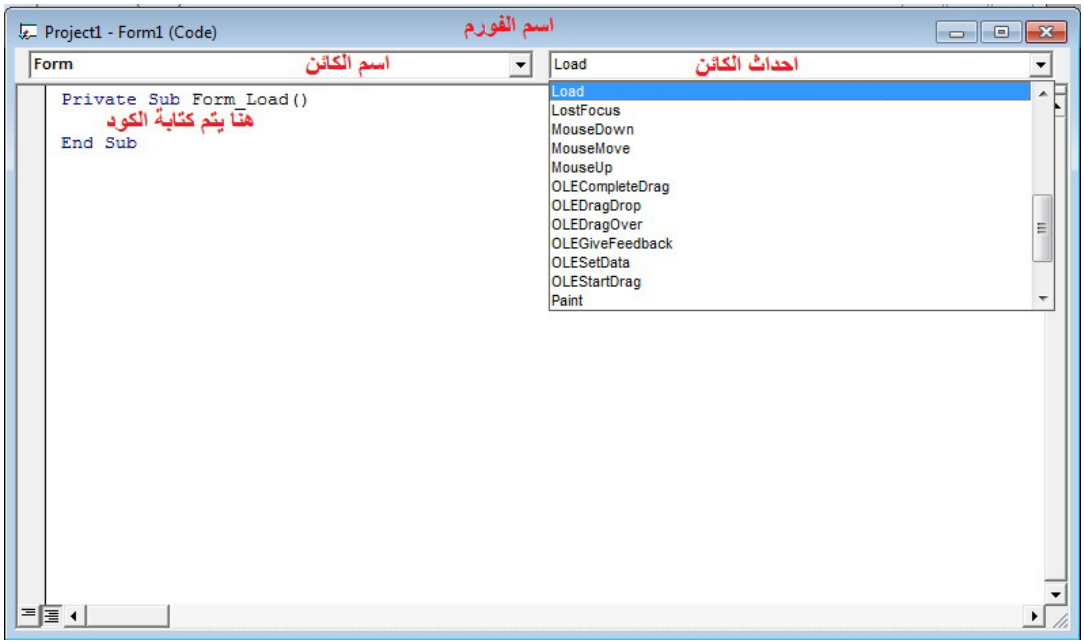
مثلا تريد أن تخفي أداة تستعمل الخاصية Visible كالتالي :

```
TextBox1.Visible = False
```

اسم الأداة — أساسية — خاصية الإخفاء و الإظهار — أساسية — قيمة false تعني إخفاء

شرح الكود ؟

إذا ضغطت على الزر (Command) سيقوم بإخفاء مربع النص (TextBox1) لان الخاصية visible يقوم باخفاء اي كائن يستعمل معه .



كتابة الكود :-

```
Private Sub CommandButton1_Click()
TextBox1.Visible = False
End Sub
```

و تستطيع أن تلغي عمل شفرة ما بدون مسحها بوضع قبلها العلامة (') مثل :

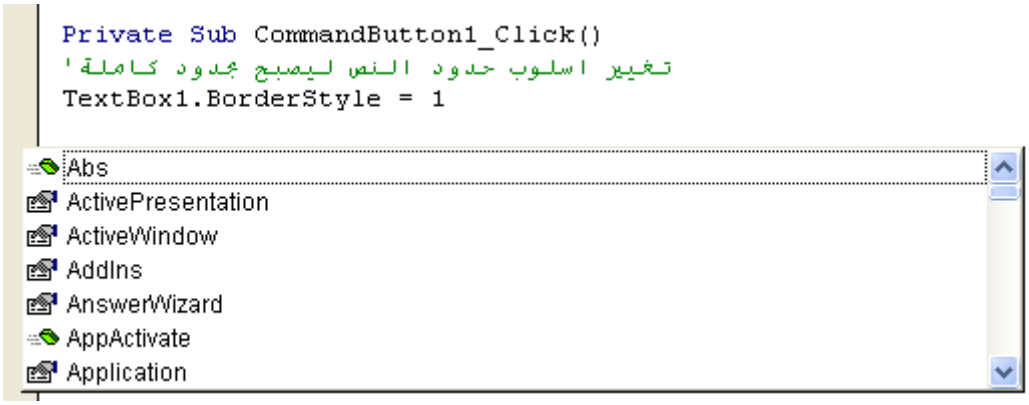
```
Private Sub CommandButton1_Click()
'TextBox1.Visible = False
TextBox1.BorderStyle = 0
End Sub
```

هذه الشفرة لن تنفذ بسبب العلامة (') في بدايتها

و حتى تستطيع كتابة تعليق على الشفرة بنفس الطريقة .

```
Private Sub CommandButton1_Click()
'تغيير أسلوب حدود النص ليصبح بحدود كاملة
TextBox1.BorderStyle = 1
End Sub
```

قد تواجه صعوبة في حفظ الشفرات أنصحك بعدم حفظها بل اجعل البرنامج هو من يساعدك فقم بما يلي:
إذا أردت أن يساعدك البرنامج فقط قم بالضغط على زر (Ctrl) من لوحة المفاتيح مع المسافة سوف يظهر لك جميع الشفرات المتاحة مثل التالي :
اذهب لمحرر الشفرات و المكان الذي تريد وضع فيه الشفرة ثم اضغط على المسافة و(Ctrl) .
ستظهر لك مثل ما بالصورة التالية :-



اكتب الحروف التي تريدها أو اسم الكائن و سيساعدك البرنامج في اختيار اسم الأداة و حتى الخصائص .

نصيحة : إذا أخطأت في حرف واحد أو حتى رقم لن يتم تشغيل الشفرة فتأكد دائما من الحروف و استعمل مساعد الشفرة دوما .

الالوان في الفيجوال بيسك

يوجد(4) طرق للتعامل مع الالوان وهي :

1 - عن طريق الدوال الجاهزة منها دوال(VB) و هي .

VbRed أحمر
VbYellow أصفر
VbGreen أخضر
VbBlue أزرق
VbBlack أسود
VbWhite أبيض
VbGray رمادي
vbMagenta زهري مائل للبنفسجي

مثال :-

```
Form1.BackColor = vbGreen
Form1.BackColor = vbBlue
```

2 - و هناك دوال أخرى تسمى بدوال Qb و هي اختصار ل Quick Basic و تسند إليها أرقام من 0 إلى 16 كالتالي QbColor(1)

مثال :-

```
Form1.BackColor = QBColor (0)     اسود
Form1.ForeColor = QBColor (15)    ابيض
```

وهذا جدول يبين رقم كل لون في البرنامج

الرمز	اللون	الرمز	اللون
0	Black أسود	8	Grey رمادي
1	Blue أزرق	9	Light Blue أزرق فاتح
2	Green أخضر	10	Light Green أخضر فاتح
3	Cyan أزرق سماوي	11	Light Cyan سماوي فاتح
4	Red أحمر	12	Orange برتقالي
5	Pink زهري	13	Light Pink زهري فاتح
6	Brown بني	14	Yellow أصفر
7	White أبيض	15	Light White أبيض ساطع

3 - الطريقة الثالثة من خلال نظام (RGB) أي red green blue

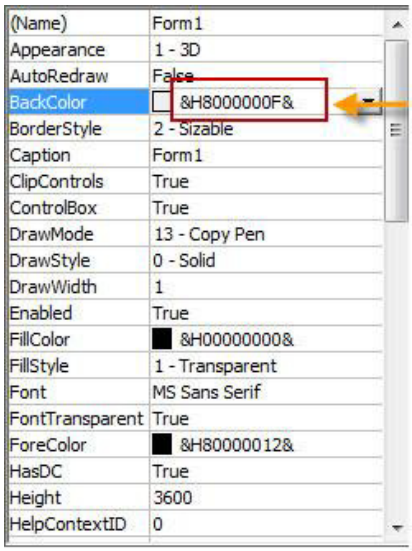
حيث تحدد كمية اللون الأحمر و الأخضر و الأزرق و هذه بصراحة للمحترفين بالألوان.

مثال :-

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

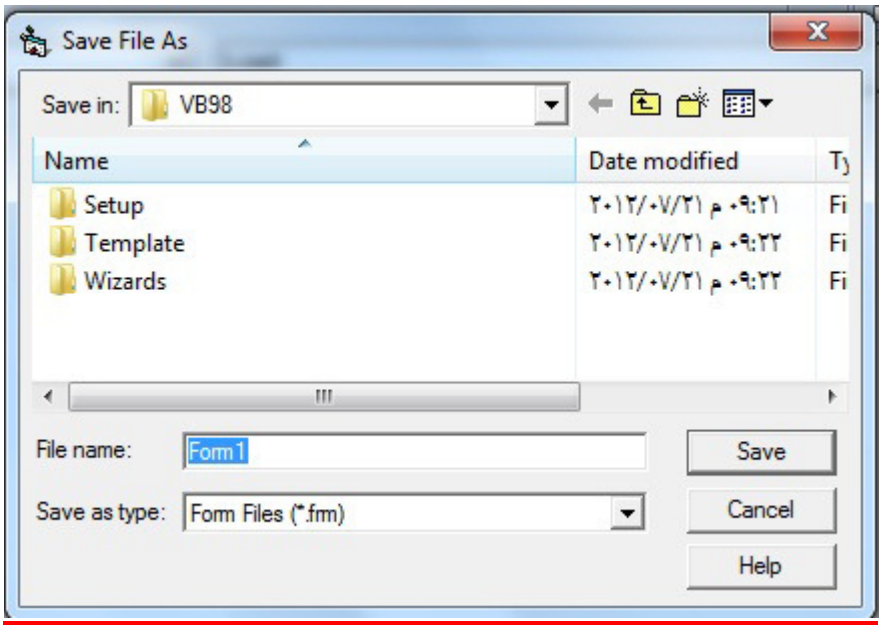
حيث أعلى قيمة تكون (255)

4 - تحديد اسم اللون من خلال خصائص الاداة .



ملاحظة :- في بعض الادوات يتطلب تغيير خاصية (style) الى (0- graphical) .

حفظ المشروع



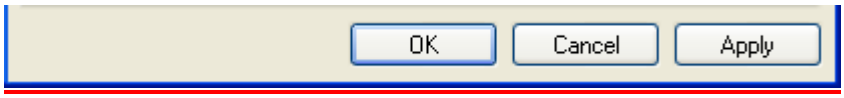
بعد أن عملت برنامجاً على الفيجوال بيسك فإنك سوف تحفظه وطريقة الحفظ هي :-

1- عند إغلاق الفيجوال بيسك فسوف يتم سؤالك إذا كنت تريد الحفظ . تختار (Yes) فيظهر لك صندوق حوار ويكون قد أعطى لمشروعك إسم (Project1)
لنفرض أنك لم تغير الإسم أنقر فوق الزر (Save) فيتم حفظ المشروع .

2- ويمكنك الحفظ من قائمة (File) ثم تختار(Save) وتحفظ المشروع
بعد ان حفظنا المشروع (البرنامج) باسم)project1(اذهب قائمة (File) ثم اختر منها make project1.exe
طبعا عند حفظ البرنامج البرنامج فجوال بيسيك تلقائيا يضيف العبارة make project1.exe
ليسهل عليك العمل و هذا اكيد يختلف فعندك مثلا لا يظهر project1.exe make لكن يظهر اسم البرنامج الذي حفظته
الان اذهب للمسار الذي كنت حفظت فيه البرنامج تراه قد عمل ايقونة لبرنامجك اوتوماتيكيا و عند النقر على الايقونة يفتح برنامجك الرائع .

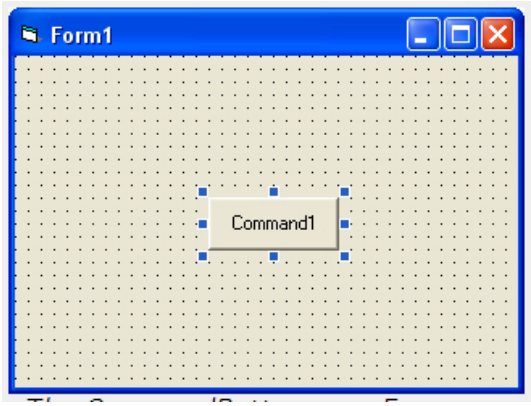
التعامل مع زر الأمر Command Button

وشكلها في الكومبيوتر كما مبين :



زر الأمر هو من الأدوات المهمة في الفيجوال بيسك وإذا عملت أي برنامج فسوف تحتاج بالتأكيد إلى وضع زر أمر في برنامجك فهو الأداة الأساسية للأحداث .

لكتابة أي حدث في الزر أنقر فوقه نقرتين ثم أكتب الشيفرة التي تريد
لتغيير النص الذي سيظهر في الزر طبعاً تعرفها في خاصية (Caption) أكتب التسمية.
ويمكنك أيضاً إضافة صورة إلى الزر ويمكنك تغير اللون . ولاكن عليك أن تبدل خاصية الستايل (Style) إلى (Graphical)
ويمكن أن يستخدم الزر ليعمل أكثر من حدث في نفس الوقت في (Click و MouseDown و MouseUp)

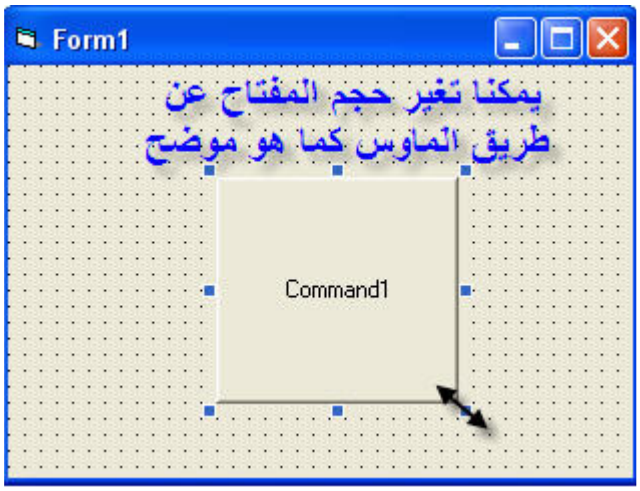


الخواص الأساسية للزر

اسم الخاصية	معناها
Name	اسم الأداة
Appearance	الظهور
BackColor	لون الخلفية
Caption	النص
Default	زر إفتراضي عند ضغط مفتاح الإدخال
DisablePicture	الصورة عندما يكون غير متاح
DownPicture	الصورة عند الضغط لأسفل
Enabled	الفاعلية
Font	نوع الخط
ForColor	لون الخط
MaskColore	قناع اللون
Style	النمط
Left	البعد عن الحافة اليسرى
MousePointer	مؤشر الماوس
MouseIcon	الأيقونة التي تشمل مؤشر الماوس
RightToLeft	الإتجاه من اليمين لليسار
TabIndex	ترتيب Tab
TabStop	إيقاف ال Tab

ToolTipText	ماسيظهر عند وضع الماوس فوقها
Tag	Tag
Picture	الصورة في الوضع العادي
Top	البعد عن حافة النافذة العليا
Visible	مرئي او مخفي
Width	العرض

ويمكنا ان نغير حجم الزر على حسب رغبتك .



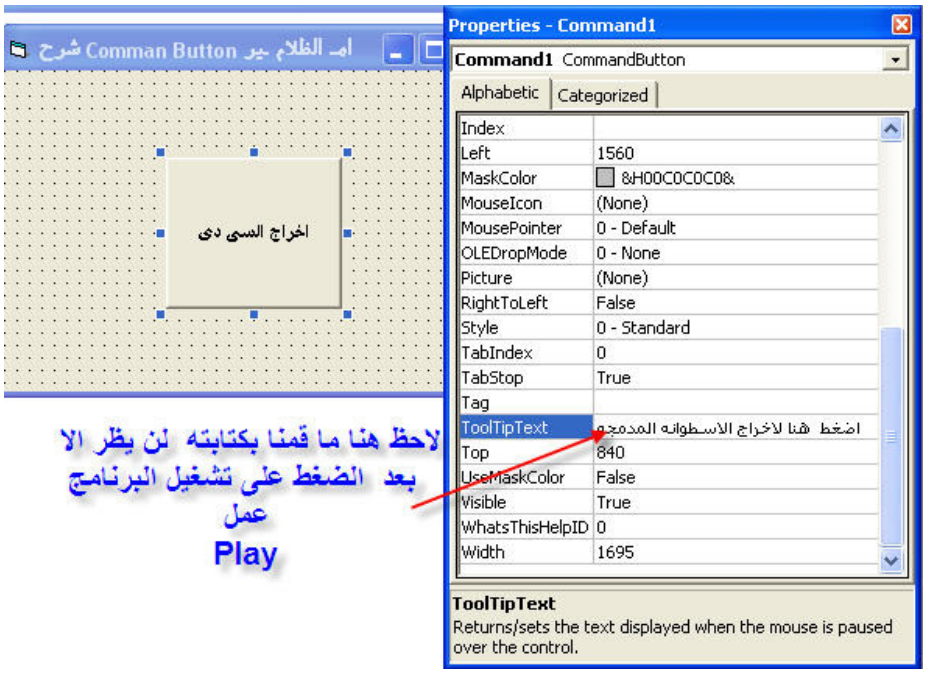
أى اداة من أدوات الفجوال بيسك 6 يتم وضعها على النموذج بطريقتين أما النقر المزدوج على الاداة فتجدها تلقائيا فى وسط النموذج أو عن طريق سحب الاداة من صندوق الادوات ووضعها على النموذج .

الخصائص المميزة لزر الامر فهى على النحو التالى:

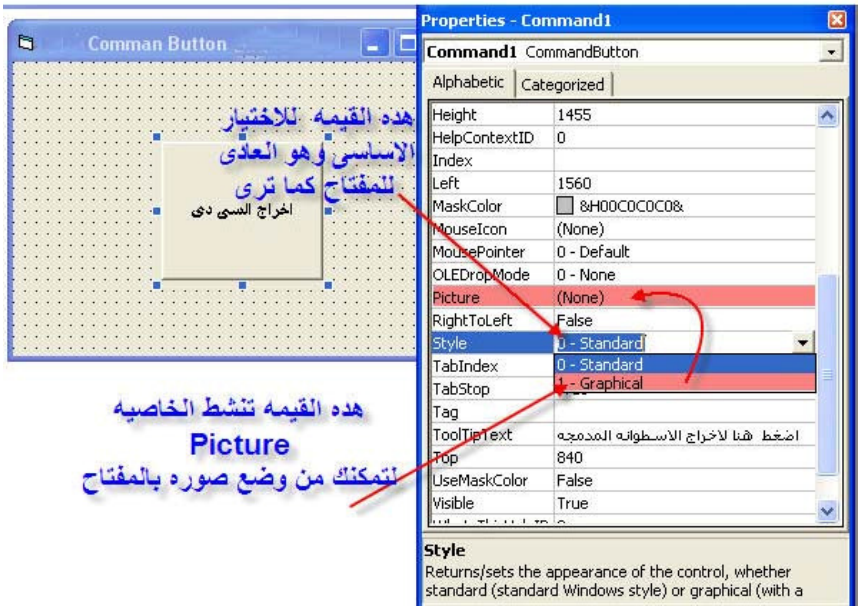
1-الزر الافتراضى Default :-
وهو يمكن المستخدم من تنفيذ الامر المرتبط بالزر اما بالنقر عليه بزر الفارة او بالضغط Enter من لوحة المفاتيح
وذلك اذا غيرت القيمة الى True والوضع الافتراضى له هوfalse

2-خاصية الالغاء Cancel :-
وهو يمكن المستخدم من تنفيذ الامر المرتبط بالزر اما بالنقر عليه بزر الفارة او بالضغط على زر الهروب Esc من لوحة المفاتيح
وذلك اذا غيرت الخاصية الى True والوضع الافتراضى لهو هوFalse

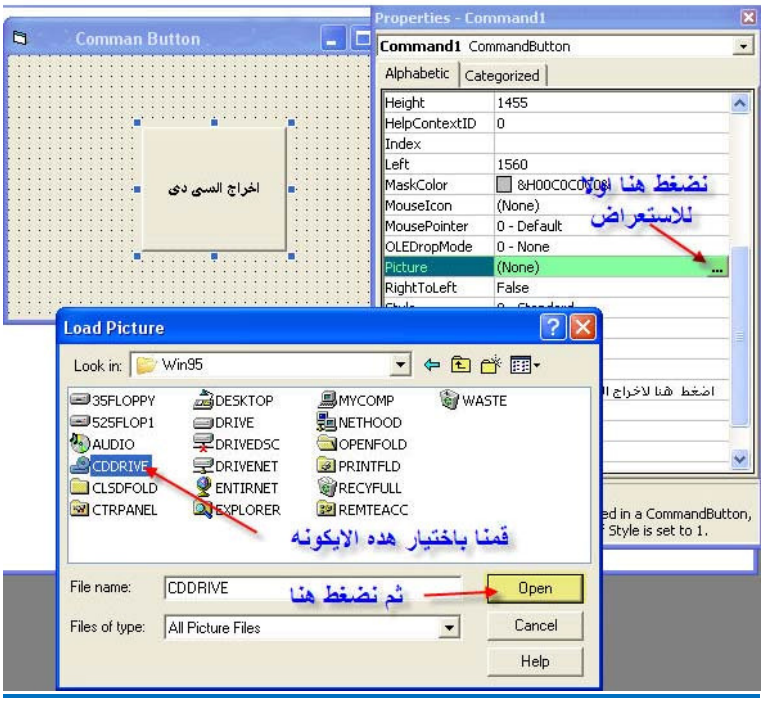
3-خاصية Tip Tool Text :-
حيث ما يتم كتابة امام هذه الخاصية يظهر اسفل زر الامر بمجرد الوقوف عليه لتوضيح معلومة معينة او توضيح وظيفته

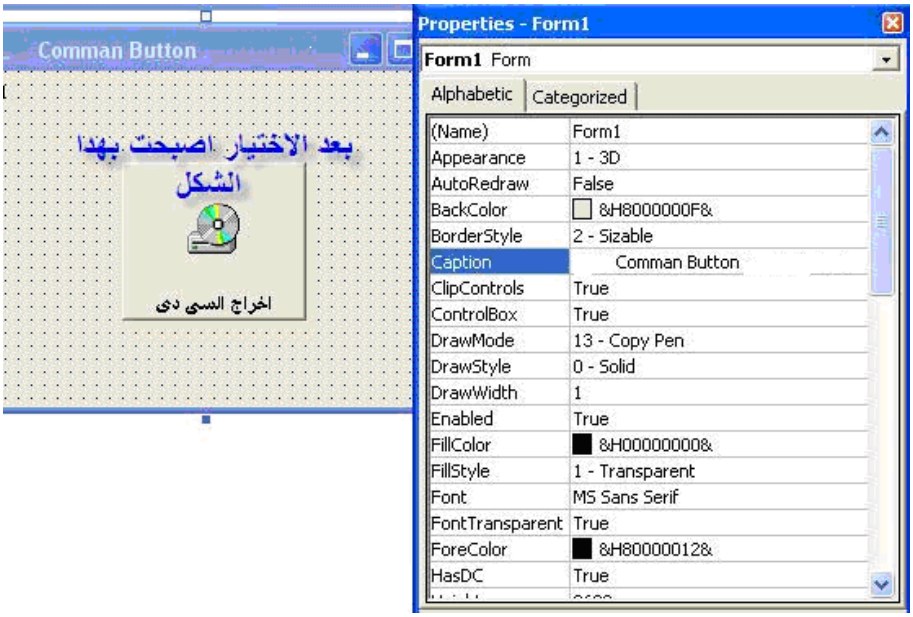


4 - خاصية Style :



5-خاصيه Picture :





مثال :-

عند الضغط على مفتاح (Command1) سوف ينفذ عدة اوامر مرة واحدة وهو(تغيير لون الفورم، تغيير عنوان الفورم ، تغيير عرض الزر ، تغيير عنوان الزر ، تغيير موقع الزر) .

قم بوضع (Command1)على نافذة البرنامج
انقر نقرا مزدوجا على (Command1)
أدخل الكود التالي:-

```
Private Sub Command1_Click()
    Form1.BackColor = 0
    Form1.Caption = "eng"
    Command1.Width = 2500
    Command1.Caption = " ele "
    Command1.Top = 10
End Sub
```

*** وبعد ذلك اضغط (F5) لتشغيل البرنامج.**

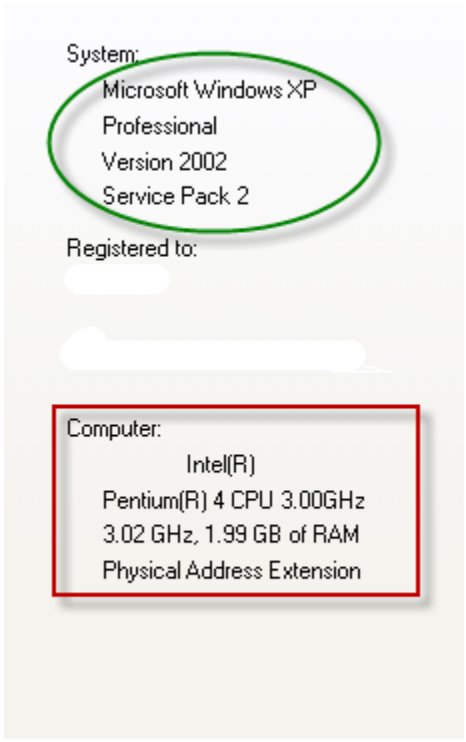
التعامل مع Label و Text

أولا التعامل مع الأداة Label

مثل عرض مواصفات الكومبيوتر من خلال

My Computer >>>> properties

فتظهر النافذة التالية :



تستخدم الأداة(Label)لإظهار النصوص على النموذج دون أن تتمكن من الكتابة فيها .

لتغير النص الذي سيظهر بداخلها طبعاً عن طريق (Caption) .

وهذا الجدول يبين الخصائص الأساسية(Label)

اسم الخاصية	معناها
Name	اسم الأداة
Aignment	محاذاة
Appearance	الظهور
AutoSize	حجم تلقائي حسب النص
BackColor	لون الخلفية
BorderStyle	نمط الإطار
Caption	النص
Enabled	الفاعلية
Font	نوع الخط
ForColor	لون الخط
Left	البعد عن الحافة اليسرى
MousePointer	مؤشر الماوس
MouseIcon	الأيقونة التي تشمل مؤشر الماوس
RightToLeft	الإتجاه من اليمين لليسار
TabIndex	ترتيب Tab
TabStop	إيقاف الTab
ToolTipText	ماسيظهر عند وضع الماوس فوقها
Tag	Tag
Caption	النص
Top	البعد عن حافة النافذة العليا
Visible	مرئي أو مخفي
Width	العرض

كتابة كود عدة خصائص ل(text)بواسطة(with)

مثال :- اضف اداتي text , command واكتب الاكواد التالية في اداة command ؟

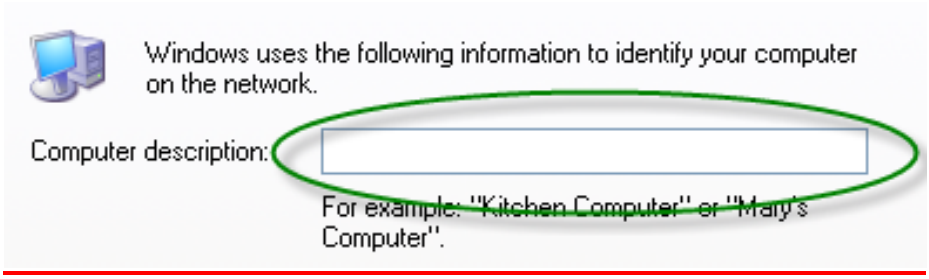
```
Private Sub Command1_Click()
With Text1
.Text = " فيجوال بيسك 6 "
.Font.Bold = True
.Font.Name = "rod"
.Font.Size = 40
.BackColor = vbblak
.ForeColor = vbWhite
End With
End Sub
```

التعامل مع TextBox

مثل عرض وتغيير اسم الكومبيوتر

My Computer >>>> properties

فتظهر النافذة التالية :-



أداة(TextBox) تستخدم في إدخال النصوص إلي البرنامج كما يمكن إستخدامها في إخراج النصوص.
وهي لاتختلف كثيراً عن(Label) سوى أنها يمكن الكتابة فيها أما(Label) فلا يمكن الكتابة فيها .

وهذا جدول الخصائص الأساسية ل Text Box

اسم الخاصية	معناها
Name	اسم الأداة
Aignment	محاذاة
Appearance	الظهور
AutoSize	حجم تلقائي حسب النص
BackColor	لون الخلفية
BorderStyle	نمط الإطار
CauseValidation	التصحيح
Text	النص
Locked	مقفول لا يمكن الكتابة بداخلها
MaxLength	الحد الأقصى
Enabled	الفاعلية
Font	نوع الخط
ForColor	لون الخط
Left	البعد عن الحافة اليسرى
MousePointer	مؤشر الماوس
MouseIcon	الأيقونة التي تشمل مؤشر الماوس
PasswordChar	حرف كلمة المرور
MultiLine	هل تستخدم لعدة أسطر أم لا
ScrollBars	أشرطة التمرير
RightToLeft	الإتجاه من اليمين لليسار
TabIndex	ترتيب Tab
TabStop	إيقاف ال Tab
ToolTipText	ماسيظهر عند وضع الماوس فوقها
Tag	Tag
Caption	النص

Top	البعد عن حافة النافذة العليا
Visible	مرئي أو مخفي
Width	العرض

إن التعامل مع هاتين الأداتين ليس صعباً ، عليك فهم صندوق الخصائص فتعرف كيف تتعامل مع كافة الأدوات

إذا أردت أن تربط بين جملتين استعمل الرمز(&) و ممكن أن تستعمل بدلها (+) و هي تتم هكذا :-

Text1.Text = "حسن" & " محمد"

و النتيجة ستكون (حسن محمد) مثل نتيجة الجملة التالية :-

Text1.Text = "حسن محمد"

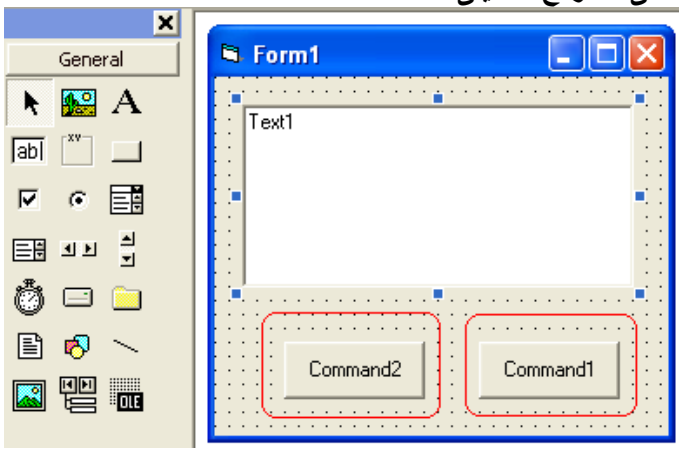
آو حتى تربط بين جملة و رقم .

مثال :-

صمم برنامج يحتوي على زرين امر وصندوق نص وأجعل المشروع يكتب رسالة ترحيب عند نقر الزر الاول بينما يؤدي النقر على الزر الثاني الى مسح رسالة الترحيب

الحل :-

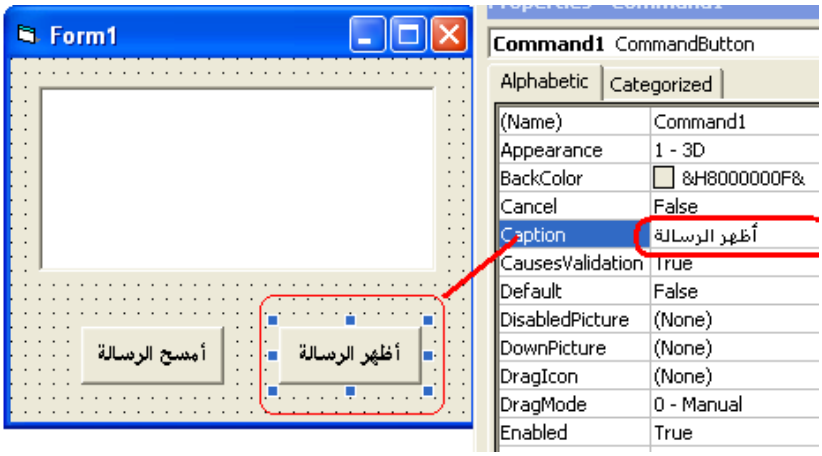
نضع زر أمر عدد (2) وصندوق نص على النموذج كما يلى :-

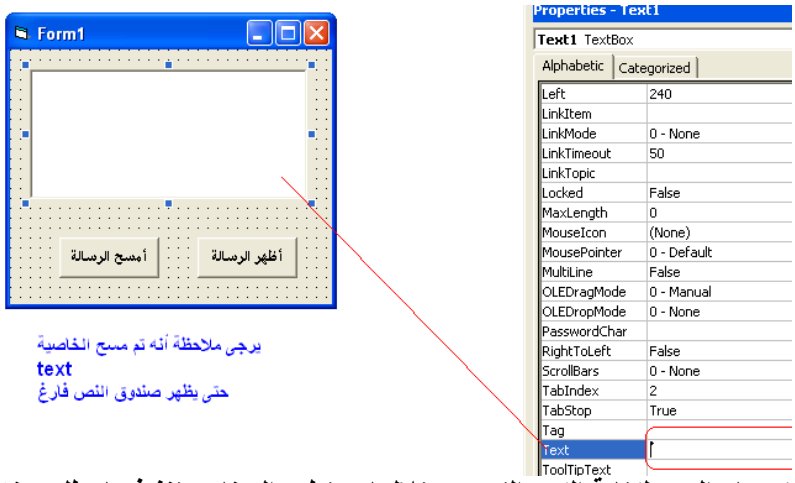


نمسح الخاصية (Text)لصندوق النص و نغير خاصية (caption) لزر الامر على النحو التالى:

الزر الاول = أظهر الرسالة

الزر الاخر= أمسح الرسالة

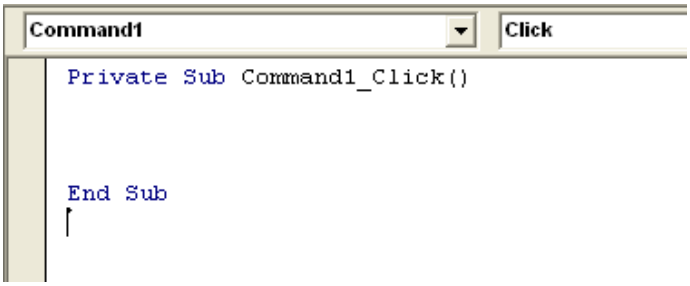




بعد ما قمنا بتصميم البرنامج الان جاء الدور لكتابة الكود التى من خلالها يستطيع البرنامج تنفيذ ما يطلب منه .

اولا :- عند الضغط على زر" أظهر الرسالة" ظهور رسالة فى صندوق النص نصها"مرحبا بك في القسم" .

اضغط (Double Click) على زر الامر" أظهر الرسالة" فيتم فتح نافذة الكود تحت الاداة كما فى الصورة التالية: -



نكتب الكود التالى في الزر :-

```
Private Sub Command1_Click()
Text1.Text = " مرحبا بك في القسم "
End Sub
```

ثانيا :- عند الضغط على زر" أمسح الرسالة" يتم مسح مربع النص .

```
Private Sub Command1_Click()
Text1.Text = "  "
End Sub
```

للقيام بتشغيل البرنامج نقوم بالضغط على زر
start كما فى الصورة التالية :-

صندوق الصور PictureBox

الاسم	الوظيفة
Aignment	محاذاة
Appearance	نوع الإطار
AutoSize	حجم تلقائي حسب الصورة
ToolTipText	ماسيظهر إذا وضع الماوس فوقها
Picture	الصورة
BackColor	الون الخلفيه
Visible	مخفي أم مرئي

يستخدم هذا الصندوق لعرض الصور بحجمها الطبيعي وله إستخدامات أخرى كثيرة فهو يستخدم كإطار لأدوات تحكم أخرى وهو أيضاً يستخدم لعرض الفيديو ويستخدم للرسم والإنشاء ويمكن تحويله إلى أداة عرض نصوص وزر امر ، سوف نتحدث عن ذلك لاحقاً.

لوضع صورة : من مربع الخصائص الخاص ب (Picture) سوف تجد الخاصية (Picture) أنقر فوق الزر لتحديد صورة .

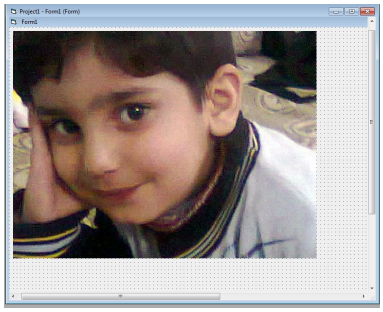
أو عن طريق الكود التالي في حدث (Load) للفورم :-

```
Picture1 . Picture = LoadPicture (هنا ضع عنوان الصورة)
```

يمتاز صندوق الصور (Picture) عن صندوق عرض الصور (Image) بمايلي:

- يعرض الصورة بحجمها الطبيعي ولو صغرت إطار الصندوق فلن تصغر الصورة. بعكس (Image)
- يمكن تغيير لون الخلفية ونمط الإطار أما (Image) لا
- يستخدم في أغراض كثيرة أما (Image) لعرض الصور فقط .

هذا شكل الأداة على الفورم



الخصائص الأساسية ل(Picture Box)

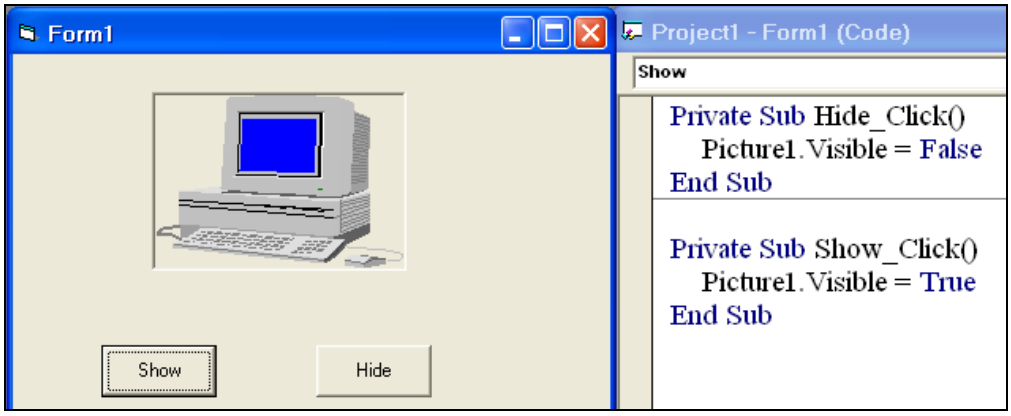
اسم الخاصية	معناها
Name	اسم الأداة
Aignment	محاذاة
Appearance	نوع الإطار
AutoSize	حجم تلقائي حسب الصورة
BackColor	لون الخلفية
BorderStyle	نمط الإطار
Caption	النص
Enabled	الفاعلية
Font	نوع الخط
ForColor	لون الخط
Height	الإرتفاع
Index	ترتيب مفتاح Tab

Left	البعد عن الحافة اليسرى
MousePointer	مؤشر الماوس
MouseIcon	الأيقونة التي تشمل مؤشر الماوس
Picture	الصورة
RightToLeft	الإتجاه من اليمين لليسار
TabIndex	ترتيب Tab
TabStop	إيقاف ال Tab
ToolTipText	ماسيظهر عند وضع الماوس فوقها
Tag	Tag
Caption	النص
Top	البعد عن حافة النافذة العليا
Visible	مرئي او مخفي
Width	العرض

مثال :-

أكتب برنامجا يقوم بإظهار و إخفاء صورة الكمبيوتر المخزنة في الملف:

ضع اداة (picture) واداة (command) عدد (2)



ثم اضغط مفتاح (f5) لتشغيل البرنامج .

صندوق الصورة Image Box

يستخدم فقط لعرض الصور وتكبيرها وتصغيرها حسب ماتريد .

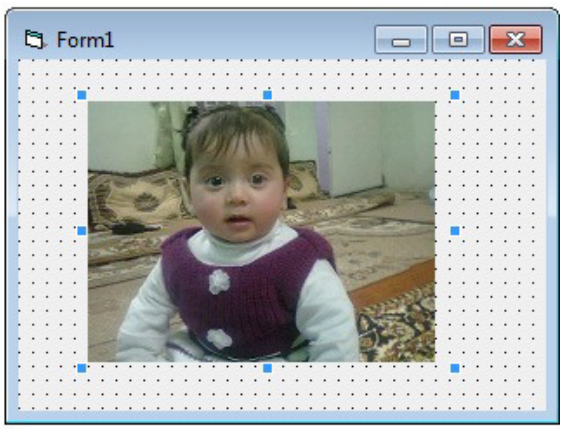
لوضع صورة : من مربع الخصائص الخاص ب (Image) سوف تجد الخاصية (Picture) أنقر فوق الزر لتحديد صورة .

أو عن طريق الكود التالي في حدث (Load) للفورم :

Image1 . Picture = LoadPicture (هنا ضع عنوان الصورة)

لتغير حجم الصورة هناك خاصية تحت خاصية (Picture) إسمها (Stretch)عليك تبديلها إلى (True)

هذا شكل ال (Image) على الفورم لاحظ حجم الصورة الطبيعي في أعلى الصفحة.

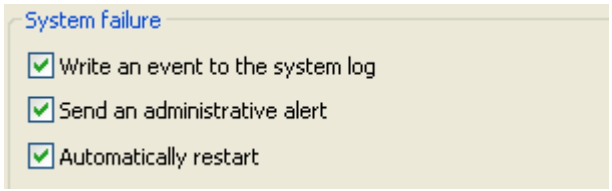


الخصائص الأساسية ل (Image)

اسم الخاصية	معناها
Name	اسم الأداة
Aignment	محاذاة
BorderStyle	نوع الإطار
Enabled	الفاعلية
Font	نوع الخط
ForColor	لون الخط
Height	الإرتفاع
Index	ترتيب مفتاح Tab
Left	البعد عن الحافة اليسرى
MousePointer	مؤشر الماوس
MouseIcon	الأيقونة التي تشمل مؤشر الماوس
Picture	الصورة
Stretch	السماح بتغيير حجم الصورة
TabIndex	ترتيب Tab
TabStop	إيقاف ال Tab
ToolTipText	ماسيظهر عند وضع الماوس فوقها
Top	البعد عن حافة النافذة العليا
Visible	مرئي او مخفي
Width	العرض

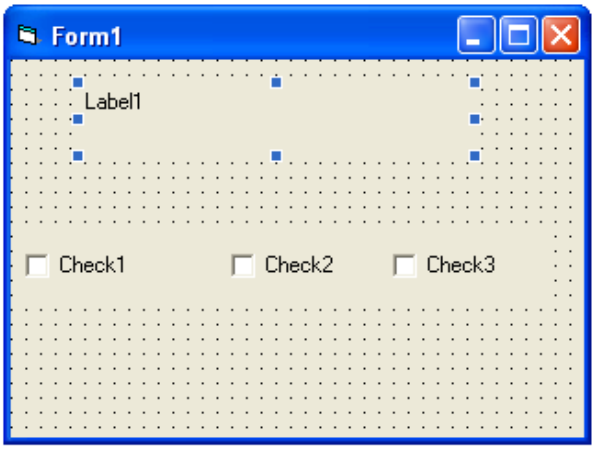
الاداة Check Box

اذا فرضنا انه يوجد علي الفورم (3) منها يمكننا اختيار واحد فقط او اثنين او الثلاثه معا ويمكننا ايضا ان لا نختار منهم شيئا .

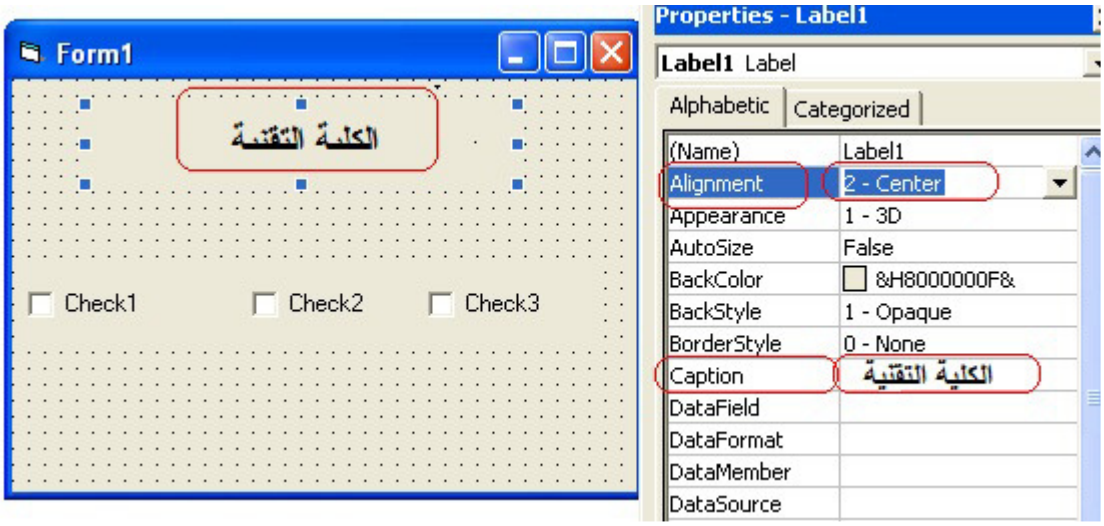


مثال :-

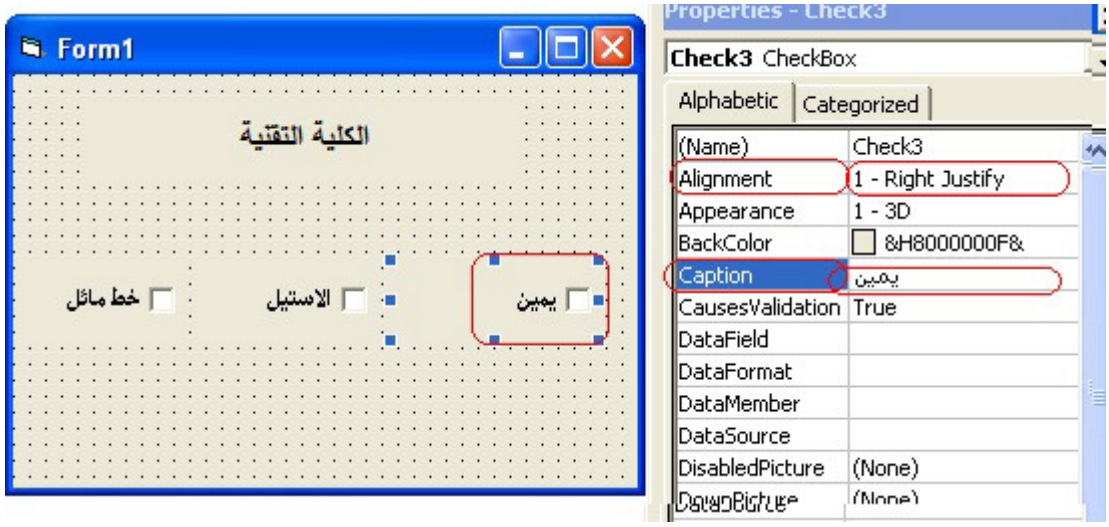
قم بتصميم برنامج وضع عليه (label) عدد (1) و اداة (check) عدد(3) خانة أختيار كما فى الصورة التالية :-



قم بضبط خاصية (caption)لأداة العنوان (label)الى "الكلية التقنية " وأجعل خاصية (Alignment) الى (center) كما فى الصورة التالية :



قم بتغير خاصية (caption)لمربعات الاختيار الثلاثة الى "يمين"،"استايل"،"خط مائل" وقم أيضا بضبط (Alignment) الى(Right)) حتى تكون مربعات الاختيار على اليمين كما فى الصورة التالية :



كتابة الكود :

نقوم بالضغط دبل كليك على مربع الاختيار رقم (1) و الذى قمنا بتمسية بـــ"خط مائل" ثم نكتب الشفرة أو الكود التالى :

```
Private Sub Check1_Click()
label1.FontItalic = True
End Sub
```

شرح الكود :
عند أختيار مربع الاختيار رقم (1) بالضغط عليه أجعل خاصية (FontItalic) للاداة (label1) تساوى (True) أى أعمل أداة العنوان بالخط المائل .
نقوم بالضغط دبل كليك على مربع الاختيار رقم (2) و الذى قمنا بتمسية بـــ"استايل "ثم نكتب الشفرة أو الكود التالى

```
Private Sub Check1_Click()
label1.BorderStyle = 1
End Sub
```

شرح الكود :
عند أختيار مربع الاختيار رقم (2) بالضغط عليه أجعل خاصية (BorderStyle) للاداة (label1) تساوى(1)
نقوم بالضغط دبل كليك على مربع الاختيار رقم (3) و الذى قمنا بتمسية بـــ"يمين " ثم نكتب الشفرة أو الكود التالى : -

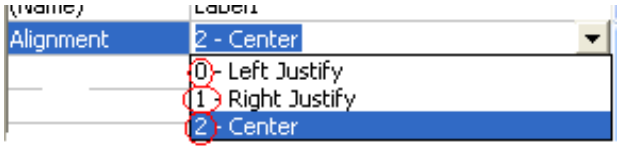
```
Private Sub Check1_Click()
label1.Alignment = 1
End Sub
```

شرح الكود :

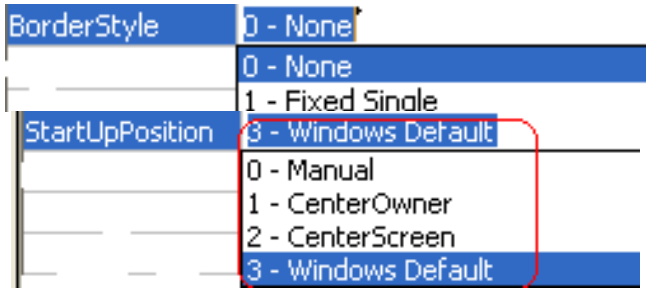
عند أختيار مربع الاختيار رقم (3) أجعل خاصية (Alignment) للاداة (label1) تساوى(1) أى أجعل الخط تجاة اليمين .

وقد تتسائل عن الارقام التى تم كتابتها فى الكود من أين أتينا بها ؟

والجواب هو أن لمعظم خصائص الادوات أرقام يمكنك التعبير عنها فى صحفة الكود بدلا من كتابة الخاصية نفسها فمثلا الخاصية (Alignment)تأخذ القيمة (0)لعرض النص تجاة الشمال (1), لتجاة اليمين (2) , لجعل النص فى المنتصف فيتم التعويض بهذه الارقام فى صفحة الكود بدلا (left ,right ,center) ويمكن ملاحظة أى قيمة رقمية لاى خاصية من صندوق الادوات كما فى الصورة التالية:

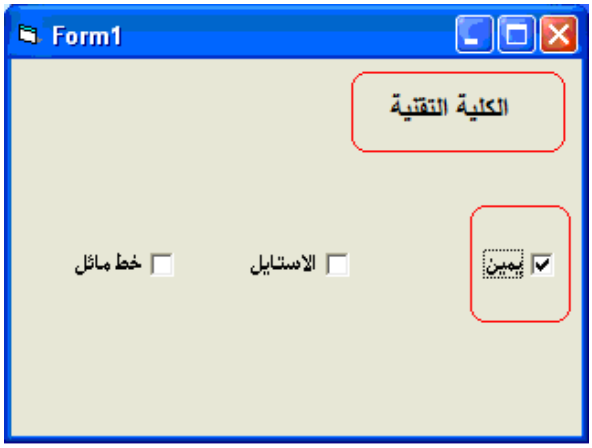


يرجى ملاحظة الارقام التى توجد قبل الاسم

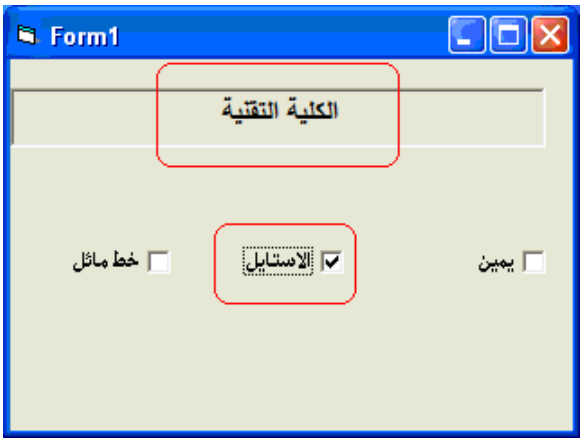


وهكذا فأن الارقام التى أتيت بها فى شفر الكود ما هى الا خصائص الاداة الان نقوم بتشغيل البرنامج لنلاحظ ماذا سيحدث أنظر الى الصور التالية :

وعندما تختار خانة الاختيار"يمين" ستجد النص جهة اليمين



وعندما نختار خانة الاختيار "استايل" ستجد أن الاستيل تغير الى ثلاثى الابعاد

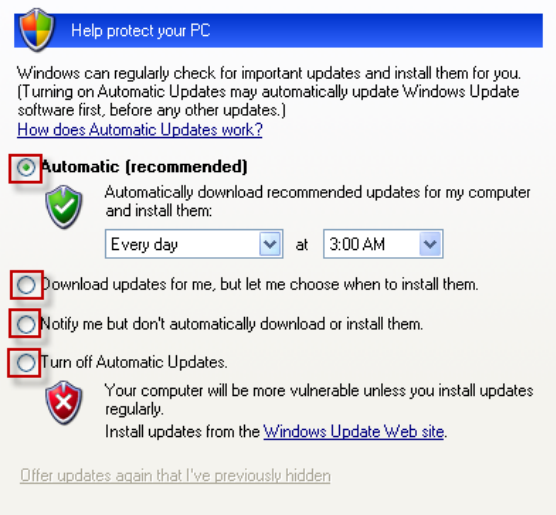


وعندما تختار خانة الاختيار "خط مائل" ستجد أن الخط اصبح مائل

بل نستطيع أن نختار جميع الخيارات المتاحة كما فى الصورة التالية:

Option الاداة

وشكله في الكومبيوتر كالاتي :

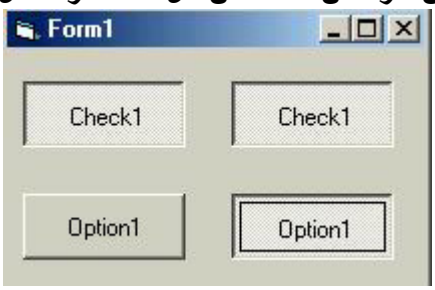


ولنفرض انه يوجد منها (3) واخترنا الاول ثم نأتي لنختار الثاني سيلغي اختيار الاول ويقوم بأختيار الثاني اي ان الاختيار من واحده منها يلغي عمل الاخر

اهم خصائصه :(Caption , Font, Enabled, BackColor, Forecolor, value and visible)

إن النص الذي بجانب كل أداة هو تابع لها يمكنك تغييره عن طريق خاصية Caption وعند النقر عليه تتغير قيمة الأداة

وعليك أن تعرف أن هنا نمطان لكل أداة أنظر الشكل التالي :



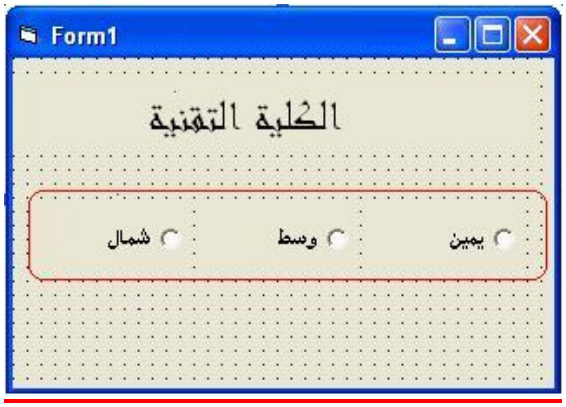
تم تغيير النمط عن طريق خاصية (Style) إلى (Graphical)
ولاحظ كيف تمكنا من تحديد خانتي الإختيار بينما لم نحدد سو زر خيار واحد.
لتغيير القيمة الإفتراضية لكل أداة من خاصية (Value) في مربع الخصائص

وللاداتين نفس الخصائص وهذا جدول للخصائص الأساسية:

اسم الخاصية	الوظيفة
Name	اسم الأداة
Aignment	محاذاة
Appearance	الظهور
BackColor	لون الخلفية
Caption	النص

DisabledPicture	الصورة عند عدم الفاعلية
DownPicture	الصورة عند النقر للأسفل
Enabled	الفاعلية
Font	نوع الخط
ForColor	لون الخط
Left	البعد عن الحافة اليسرى
MousePointer	مؤشر الماوس
MouseIcon	الأيقونة التي تشمل مؤشر الماوس
RightToLeft	الإتجاه من اليمين لليسار
TabIndex	ترتيب Tab
TabStop	إيقاف ال Tab
ToolTipText	ما سيظهر عند وضع الماوس فوقها
Tag	Tag
Picture	الصورة
Style	النمط
Value	القيمة
Top	البعد عن حافة النافذة العليا
Visible	مرئي مخفي او
Width	العرض

مثال :- قم بتصميم برنامج مكون من (3) زر خيار و أداة نص ثم أضبط خصائصهم كما فى الشكل التالى:-



نقوم بكتابة الكود التالى تحت زر الخيار "يمين"

```
Private Sub Option1_Click()
Label1.Aligment = 1
End Sub
```

نقوم بكتابة الكود التالى تحت زر الخيار "وسط"

```
Private Sub Option1_Click()
Label1.Aligment = 2
End Sub
```

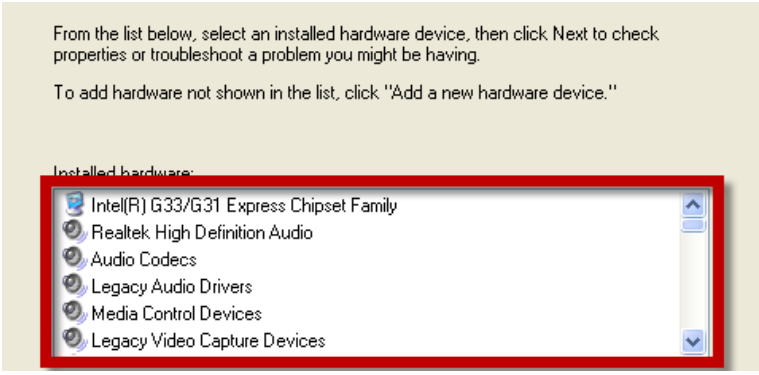
نقوم بكتابة الكود التالى تحت زر الخيار "شمال"

```
Private Sub Option1_Click()
Label1.Aligment = 0
End Sub
```

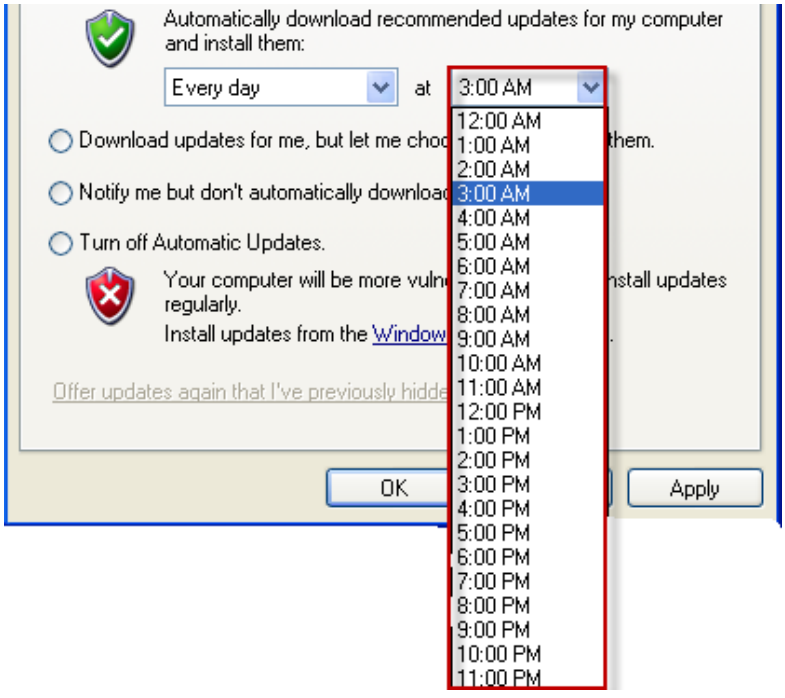
قم بتشغيل البرنامج F5

ستلاحظ أنك لو أخترت اليمين فلن تستطيع أخيار كلا من الشمال أو الوسط و لوأنك أخترت الشمال ستجد أنك غير قادر على أختيار كلا من الوسط و اليمين اذن أنت تجبر المستخدم لا ختيار خيار واحد فقط وليس خيار متعدد كما فى chek box .

List Box الاداة



Combo Box الاداة



ملاحظة : يستخدم في الاداتين القاعدة العامة الثانية لكتابة الكود (Object . method value)

لايختلف ال (combo) عن ال (list) كثيراً سوى أن ال (combo) لايظهر القائمة إلى عند النقر عليه ويستخدم الكومبو أيضاً للكتابة بداخله.

طريقة الإضافة إلى الأداتين هي نفس الطريقة:

1- عن طريق الخاصية (List) يمكنك كتابة العناصر .

2- إذا كنت تريد إضافة عناصر عند تحميل البرنامج عليك كتابة هذا الكود في حدث(Load) للفورم:

```
List1.AddItem "النص هنا"
.
.
.
Combo1.AddItem "النص هنا"
```

3- أما إذا كنت قد وضعت أداة الكتابة(TextBox) فأكتب الكود التالي:

```
List1.AddItem Text1.Text
Combo1.AddItem Text1.Text
```

إذا لإضافة نص عليك وضعه بين علامتي الإقتباس " "

ولإضافة النص من صندوق كتابة ضع إسم ال (text) دون علامات الإقتباس

ويمكنك تعديل (ListBox) للسماح بتحديدات متعددة أنظر الشكل في الأسفل وذلك عن طريق تعديل الخصائص (Style و MultiSelect)

ويمكن ربط ال (combo) و ال(list) بقاعدة بيانات جاهزة ويتم عرض السجلات بداخلهما .

هذا جدول الخصائص الأساسية لـ List Box

اسم الخاصية	معناها
Name	اسم الأداة
Appearance	الظهور
columns	عرض التحديد
Back Color	لون الخلفية
Enabled	الفاعلية
Font	نوع الخط
ForColor	لون الخط
Left	البعد عن الحافة اليسرى
MultiSelect	السماح بتحديدات متعددة
Style	يحوي على خانات إختيار
MousePointer	مؤشر الماوس
MouseIcon	الأيقونة التي تشمل مؤشر الماوس
RightToLeft	الإتجاه من اليمين لليسار
TabIndex	ترتيب Tab
TabStop	إيقاف ال Tab
ToolTipText	ماسيظهر عند وضع الماوس فوقها

Tag	Tag
Caption	النص
Top	البعد عن حافة النافذة العليا
Visible	مرئي او مخفي
Width	العرض

مثال :-

قم باضافة الاداة list box واداة command عدد 4 و واداة text وسمهم كما هو مبين من الشكل :

كتابة الكود :

في الاداة form وفي حدث load اضف الكود التالي :

```
Private Sub Form_Load()
List1.AddItem "مصر"
List1.AddItem "العراق"
List1.AddItem "فلسطين"
End Sub
```

زر مسح المحتويات

```
Private Sub Command1_Click()
List1.RemoveItem (List1.ListIndex)
End Sub
```

زر الاضافة

```
Private Sub Command1_Click()
List1.RemoveItem (List1.ListIndex)
End Sub
```

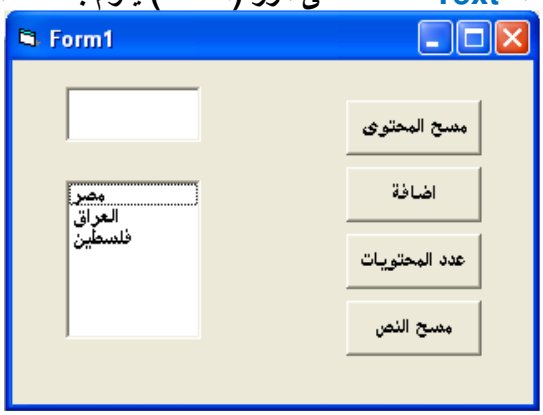
زر مسح النص

```
Private Sub Command4_Click()
Text1.Text = ""
End Sub
```

زر الاضافة

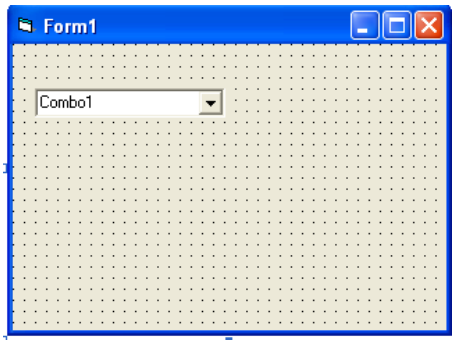
```
Private Sub Command2_Click()
List1.AddItem (Text1.Text)
End Sub
```

نقوم بتشغيل البرنامج ، وعند الكتابة في اداة الـ Text نضغط على الزر (اضافة) يقوم باضافة الاسم الجديد الى القائمة .



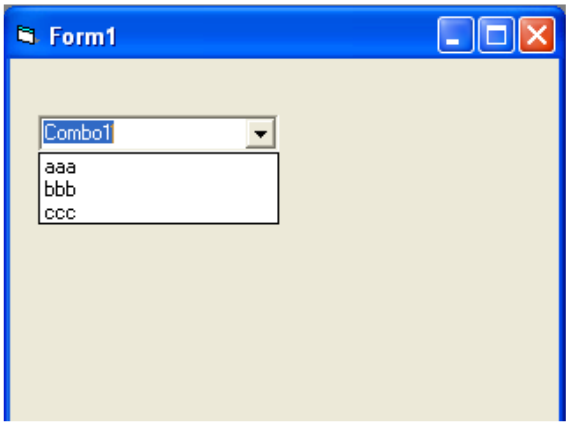
مثال :-

نقوم باضافة combobox الى المشروع وفي اداة الـ form نضيف الكود التالي :-




```
Private Sub Form_Load()
Combo1.AddItem ("aaa")
Combo1.AddItem ("bbb")
Combo1.AddItem ("ccc")
End Sub
```

نقوم بتشغيل البرنامج .



ونستطيع اضافة اي اسم الى Combo بنفس الكود السابق ولا تنسى علامة الاقتباس (" ")

التعامل مع أشرطة التمرير

ان انظمة التشغيل المختلفة واي برنامج كان لا تخلو من اشرطة التمرير الافقية او العمودية وهي من الادوات المهمة المساعدة للنظام او البرامج المتنوعة ولها استعمالات مختلفة ايضا سنذكرها في الامثلة القادمة ، و تستخدم ايضا لتغيير القيمة زيادة أو نقصان سوف نأخذ مثالاً لهذه الأداة .

هذه الأداة نوعان :

شريط التمرير الأفقي : (HScroll)

شريط التمرير العمودي : (VScroll)

ملاحظة:شريط التمرير الأفقي والعمودي لهما نفس الوظيفة والخصائص والأحداث ولكن الفرق في أتجاه الشريط فقط لا غير

هذا الجدول يبين الخصائص الأساسية لأشرطة التمرير

اسم الخاصية	معناها
Name	اسم الأداة
Enabled	الفاعلية
LargeChange	قيمة التبديل الكبرى
Max	الحد الأقصى
Min	الحد الأدنى
SmallChange	قيمة التبديل الصغرى
Value	القيمة الإفتراضية
Left	البعد عن الحافة اليسرى
MousePointer	مؤشر الماوس
MouseIcon	الأيقونة التي تشمل مؤشر الماوس
RightToLeft	الإتجاه من اليمين لليسار
TabIndex	ترتيب Tab
TabStop	إيقاف ال Tab
ToolTipText	ما سيظهر عند وضع الماوس فوقها
Tag	Tag
Top	البعد عن حافة النافذة العليا
Visible	مرئي او مخفي
Width	العرض

مثال :-

بعد تجهيز الفورم أضف إليه ثلاث أشرطة تمرير إما أفقي أو عمودي . لنفرض أنك وضعت أشرطة أفقية(HScroll) .
أسمائها على التوالي: (HScroll1 HScroll2 HScroll3)
ثم أضف إلى الفورم صندوق الصورة (Picture1)
ولاتنسى الأدوات (Label) والتي تعني النصوص التي بجانب كل شريط تمرير

أنقر فوق الفورم نقرتين وأضف الكود التالي في حدث (Load) :

```
Private Sub Form_Load()
HScroll1.Min = 0
HScroll1.Max = 250
HScroll1.SmallChange = 30
HScroll1.LargeChange = 50

HScroll2.Min = 0
HScroll2.Max = 250
HScroll2.SmallChange = 30
HScroll2.LargeChange = 50

HScroll3.Min = 0
HScroll3.Max = 250
HScroll3.SmallChange = 30
HScroll3.LargeChange = 50

End Sub
```

شرح الكود :

وتعني تحديد القيمة الصغرى والكبرى لأشرطة التمرير وتحديد سرعة التمرير .

أضف الكود التالي إلى شريط التمرير الأول:

```
Private Sub HScroll1_Change()
Picture1.BackColor = RGB(HScroll1.Value, HScroll2.Value, HScroll3.Value)
End Sub
```

الكود التالي للشريط الثاني:

```
Private Sub HScroll2_Change()
Picture1.BackColor = RGB(HScroll1.Value, HScroll2.Value, HScroll3.Value)
End Sub
```

الكود التالي للشريط الثالث :

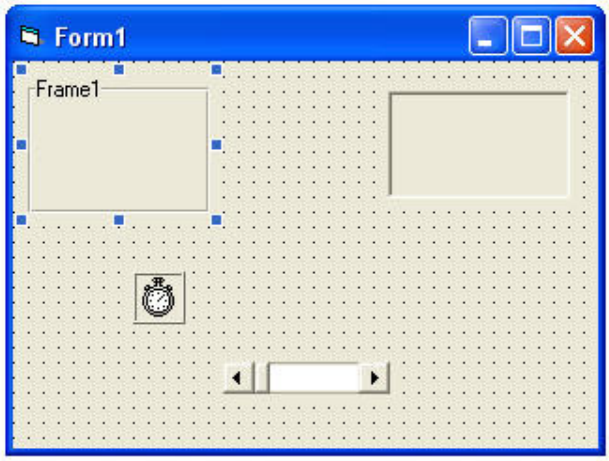
```
Private Sub HScroll3_Change()
Picture1.BackColor = RGB(HScroll1.Value, HScroll2.Value, HScroll3.Value)
End Sub
```

بعد تشغيل البرنامج جرب تبديل أشرطة التمرير ولاحظ الحدث

مثال :-

تغير لون الفورم او اى كائن على الفورم .

ننشى مشروع جديد ونضيف اليه ((scroll bar) افقى او راسى ، picturebox ، frame ، timer)



نكتب خصائص(scrollbar) فى(form) فى الحدث(load):

```
HScroll1.LargeChange = 5
HScroll1.Min = 1
HScroll1.Max = 15
HScroll1.Value = 1
```

اقل قيمة و اعظم قيمة لاداة (scrollbar)ومن الضروري تحديده وانت مخير في تحديد القيم .

اضغط (Double Click) على(scrollbar) ونكتب الكود التالي :

```
Frame1.BackColor = QBColor(Rnd * 15)
Form1.BackColor = QBColor(Rnd * 15)
Picture1.BackColor = QBColor(Rnd * 15)
```

شرح الكود

هو لتغير لون الفورم على حسب الالوان وتكون عشوائيه ونضغط دبل كلك على التايمر ونكتب الكود التالي :

```
Picture1.Top = Rnd * Form1.Height
Picture1.Left = Rnd * Form1.Width
Frame1.Top = Rnd * Form1.Height
Frame1.Left = Rnd * Form1.Width
```

ووظيفة التايمر هو لتغيير مكان ال(frame) و (picturebox) بشكل عشوائي اثناء العمل ويجب تغيير (interval) للتايمر الى قيمة مناسبة وانت مخير فيها .

التعامل مع أدوات عرض الملفات

أدوات عرض الملفات هي ثلاث :

صندوق عرض محركات الأقراص Driver List Box

استعمالها : الأداة تقوم بإظهار شتى الأقراص المركبة على الحاسوب .

صندوق عرض المجلدات Dir List Box

استعمالها : الأداة تقوم بإظهار المجلدات الموجودة في قرص ما أو في مجلد آخر .

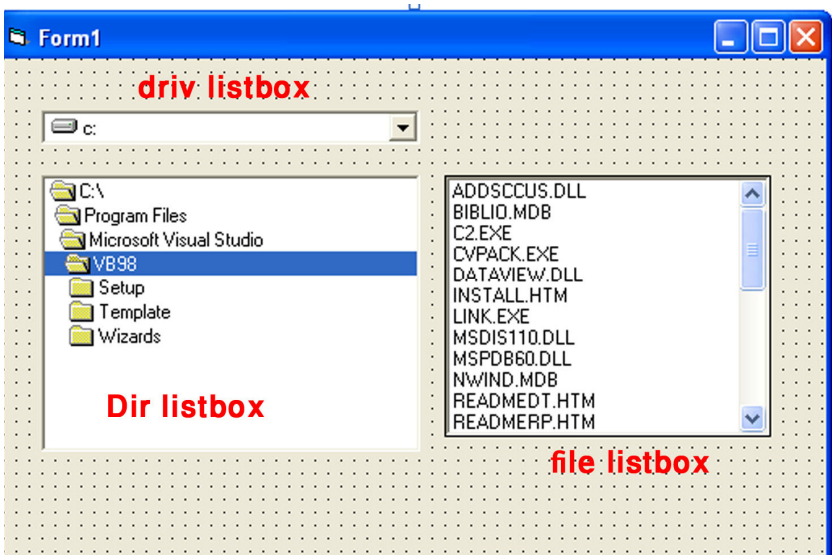
صندوق عرض الملفات File List Box

استعمالها : الأداة تقوم بإظهار الملفات الموجودة في مجلد ما أو قرص ما .

وهذه الأدوات يجب أن تكون مترابطة ببعضها .

مثال :-

بعد تجهيزك للفورم أضف إليه الأدوات السابقة . وأضف أداة (Label) كالشكل التالي :-



اضغط (Double Click) فوق صندوق محركات الأقراص (Drive1) وأكتب الكود التالي :

```
On Error Resume Next
Dir1.path = Drive1.drive
```

شرح الكود

السطر الأول يعني عند الخطأ لاتتابع مثلاً لو حدد محرك القرص اليزري ولم يكن هناك قرص فسوف يعطيك رسالة خطأ. أما هذا السطر يمنع الخطأ ومتابعة العمل أيضاً

في السطر الثاني عرض المجلدات داخل محرك الأقراص الذي تم تحديده

أنقر فوق صندوق المجلدات (Dir1) وأكتب :

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

```
Label 1 = Dir1
```

شرح الكود :
السطر الأول: عرض الملفات التي داخل هذا المجلد (في صندوق الملفات (File1)
السطر الثاني: عرض المسار الكامل في أداة النصوص Label4

اضغط (Double Click) فوق صندوق الملفات (File1) وأكتب الكود التالية :

```
Label1 = Dir1 + "\" + File1
```

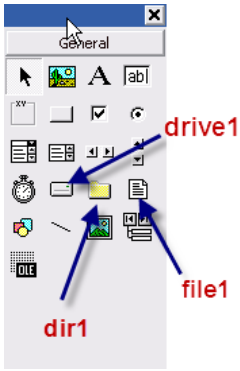
ويعني عرض المسار الكامل للملف في (Label1)

جرب البرنامج ولاحظ الحدث ، هذه الأدوات مهمة في أي برنامج فهي تستخدم لفتح ملف مثلاً.

ملاحظة:- يمكنك تحديد نوع الملفات التي تريده أن يظهر في صندوق عرض الملفات صور(Bmp) ملفات صوت (Wav) وذلك عن طريق خاصية(Pattern) مثل عرض صورة بامتداد jpg نكتب (*.jpg;*).

مثال : -
عمل برنامج عارض الصور .

نقوم باضافة الادوات التالية (image ، drive ، dir ، file)



ضع هذه الأدوات كما بالشكل التالى :-

اجعل خاصية (stretch) لأداة (image) (true) لكى تظهر الصورة كاملة بأداة ال (image)
الان ناتي لكتابة الأكواد .

اضغط (Double Click) على اداة (dir1) ثم نقوم بكتابة هذا الكود فى(dir1_change)

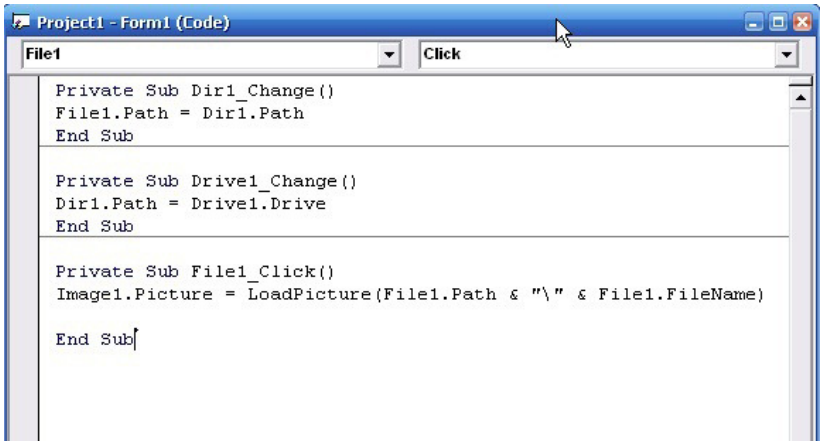
```
File1.Path = Dir1.Path
```

اضغط (Double Click) على اداة (drive1)ثم نكتب هذا الكود فى(drive1_ change)

```
Dir1.Path = Drive1.Drive
```

وهذا اخر كود اضغط (Double Click) على اداة (file1)ثم نقوم بكتابته فى (File1_Click) لتحميل الصور فى اداة (image)

```
On Error Resume Next
Image1.Picture = LoadPicture(File1.Path & "\" & File1.FileName)
```



ثم اضغط على (f5)لتشغيل البرنامج لتقوم بعرض الصور اختر (drive) ثم (folder) ثم الصورة .

التعامل مع أداة التوقيت Timer

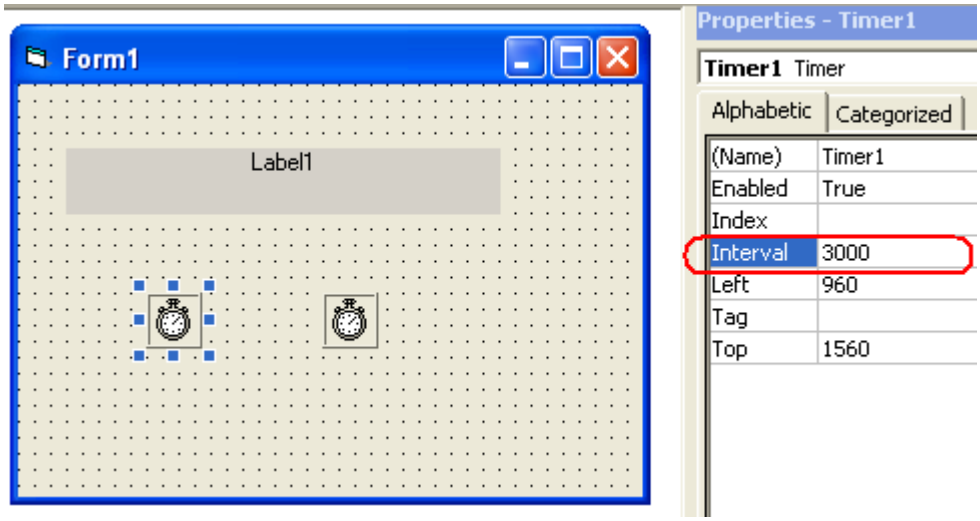
وهي اداة تستخدم لعمل حدث معين بعد مرور فترة معينة او لتكرار الحدث طول فترة زمنية معينة . واهم خاصية من خواص تلك الاداة هي خاصية (interval) وهي التي تحدد الفترة الزمنية لتنفيذ الحدث او الكود وتحدد الفترة الزمنية بالملي ثانية (ms) فمثلا لو اردت ان يعمل كل ثانية عليك كتابة (1000) للخاصية (interval) القيمة الكبرى للتوقيت هي (65535) فانها تساوي تقريبا دقيقة واحدة و (5)ثواني ونصف الثانية .

ملاحظة : يجب وضع قيمة ل (Interval) حسب المشروع والا لن يحدث اي تغيير للمشروع ، وان كل (1000)في التايمر يساوي ثانية واحدة.

لتعطيل التايمر هناك خاصية (Enabled) تتيح تعطيله او تشغيله .

مثال :-
قم بعمل نموذج وضع علية تايمر عدد(2) و(label)عدد (1) ونفترض أننا نريد من البرنامج أن يكتب "قسم الالكترونيك و السيطرة " بعد مرور (3) ثوانى و يغير لون الخط بعد مرور (5) ثوانى ؟

أولا نقوم بتصميم البرنامج كما فى الشكل التالى :-



ولاحظ أننا جعلنا الخاصية (Interval)تساوى (3000) للتايمر الاول و أيضا نجعل تلك الخاصية للتايمر الثانى تساوى (5000)أى أننا نقول للبرنامج نفذ الكود الذى سيكتب تحت التايمر الاول بعد مرور (3) ثوانى و أيضا نفذ الكود الذى سيكتب تحت التايمر الثانى بعد مرور(5) ثوانى .

اضغط (Double Click) على (Timer1) ثم اكتب الكود التالي:

```
Label1.Caption = " قسم الالكترونيك و السيطرة"
```

أما (Timer2) اضغط (Double Click) ثم اكتب الكود التالى:

```
Label1.BackColor = vbred
```

بعد ذلك قم بتشغيل البرنامج ولاحظ ماذا سيحدث ستجد أن بعد مرور (3) ثوانى أن البرنامج قد كتب " قسم الالكترونيك و السيطرة " وبعد مرور (5) ثوانى من التشغيل ستجد أن خلفية النص تغير الى اللون الاحمر.

اضغط على (f5)

ملاحظة :- لا تحاول الاكثار من الاكواد الطويلة بداخل الحدث (timer) للاداة (timer) فذالك يسبب استهلاك كبير للمعالج مما يؤدي الى ابطاء تنفيذ برامج و البرامج الاخرى في نظام التشغيل .

دوال الوقت و التاريخ

دوال الوقت والتاريخ: "Date and Time Functions "

الخاصية	العمل
Now	اظهار التاريخ والوقت الآن
Date	إدراج تاريخ
Day	أدارج رقم اليوم خلال الشهر 1- 31
Day of year	أدارج رقم اليوم خلال السنة 1- 366
Hour	أدارج الساعة 0-23
Minute	أدراج الدقيقة 1-59
Second	أدارج الثانية 0-59
Month	أدراج اسم الشهر 1- 12

1- الدالة: "Time"
تستخدم هذه الدالة لعرض الوقت الحالي .

2- الدالة: "Now"
تستخدم هذه الدالة لعرض تاريخ اليوم و الوقت الحالي .

3- الدالة: "Year"
تستخدم هذه الدالة لعرض السنة من تاريخ معين .

4- الدالة: "Month"
تستخدم هذه الدالة لعرض رقما يقع ما بين 1 إلى 12 يمثل الشهر من تاريخ معين .

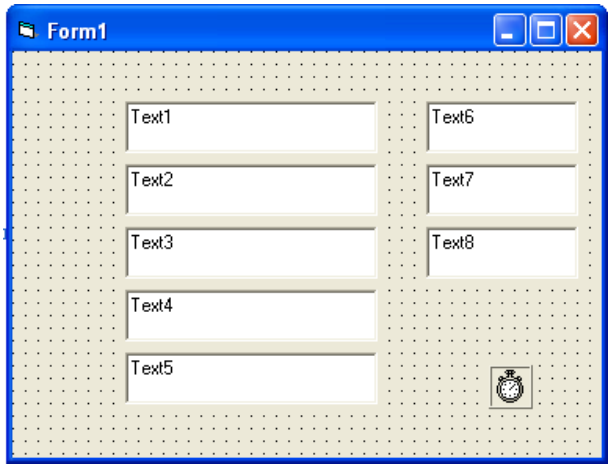
5-الدالة: "Day"
تستخدم هذه الدالة لعرض رقما يقع ما بين 1 إلى 31 يمثل اليوم من تاريخ معين .

6- الدالة: "Hour"
تعود هذه الدالة بقيمة رقمية تعبر عن الساعة .

7- الدالة: "Minute"
تعود هذه الدالة بقيمة رقمية تعبر عن الدقائق .

8- الدالة: "Second"
تعود هذه الدالة بقيمة رقمية تعبر عن الثواني .

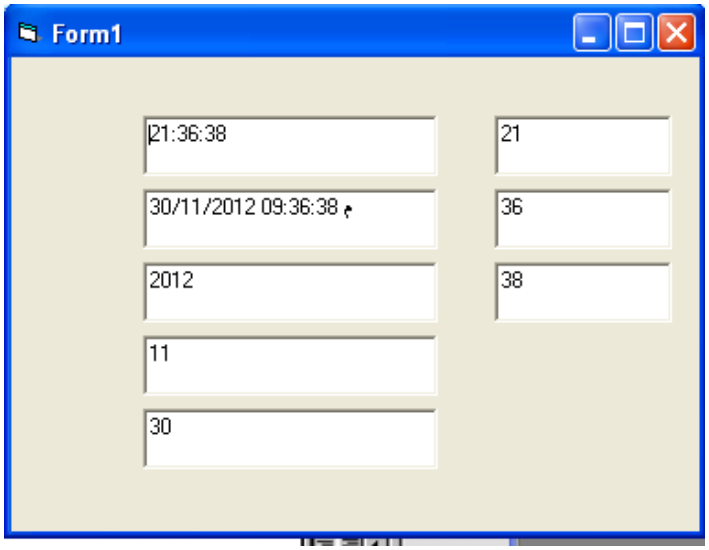
مثال :-
قم بإنشاء نموذج (Form) وقم بوضع (8)أداوت نص (Textbox)و أداة (Timer)ليصبح شكل الورم بهذا الشكل الموضح في الصورة التالية:



غير قيمة الخاصية (Interval)للأداة (Timer)إلى (1000) حيث أن هذه القيمة تعني تنفيذ الكود المكتوب تحت أداة الـ(Timer)كل ثانية واحدة , ثم افتح شاشة الكود الخاصة بالأداة (Timer)في الحدث (Timer)ثم قم بكتابة الكود الموضح بالصورة التالية:

```
Private Sub Timer1_Timer()
Text1.Text = Time()$
Text2.Text = Now()
Text3.Text = Year(Now(()
Text4.Text = Month(Now(()
Text5.Text = Day(Now(()
Text6.Text = Hour(Now(()
Text7.Text = Minute(Now(()
Text8.Text = Second(Now(()
End Sub
```

قم بتشغيل البرنامج F5 .



شرح الكود:

في السطر الأول من الكود نجعل قيمة صندوق النص (Text1)تساوى الوقت الحالي بجهاز الحاسب .

في السطر الثاني من الكود نجعل قيمة صندوق النص (Text2)تساوى تاريخ اليوم و الوقت الحالي بجهاز الحاسب عن طريق الدالة (Now) .

في السطر الثالث من الكود نريد عرض السنة من التاريخ الحالي بجهاز الحاسب في صندوق النص (Text3) عن طريق الدالة (Year).
في السطر الرابع من الكود نريد عرض الشهر من التاريخ الحالي بجهاز الحاسب في صندوق النص (Text4) عن طريق الدالة (Month).

في السطر الخامس من الكود نريد عرض اليوم من التاريخ الحالي بجهاز الحاسب في صندوق النص (Text5) عن طريق الدالة (Day).

في السطر السادس من الكود نريد عرض الساعة فقط من الوقت الحالي بجهاز الحاسب في صندوق النص (Text6) عن طريق الدالة (Hour).

في السطر السابع من الكود نريد عرض الدقائق فقط من الوقت الحالي بجهاز الحاسب في صندوق النص (Text7) عن طريق الدالة (Minute).

في السطر الثامن من الكود نريد عرض الثواني فقط من الوقت الحالي بجهاز الحاسب في صندوق النص (Text8) عن طريق الدالة (second).

طريقة اخرى

يجب ان توضع بين علامتين(# #) ، اكتب الكود في الاداة command

```
MyDate = #1/20/2001#
Print MyDate

MyDate = #2:30:00 AM#
Print MyDate

MyDate = #1/20/2001 2:30:00 AM#
Print MyDate
```

التاريخ الهجري و الميلادي

```
VBA.Calendar = vbCalHijri      هجري
VBA.Calendar = vbCalGreg       ميلادي
```

ومن ثم كتابة أمر التاريخ بصورة طبيعية كما يلي :

```
Text1.Text = DateTime.Date
```

الاداة Calendar

لاضافتها قم بالضغط على Ctrl + T ومن ثم اختيار Microsoft Calendar Control 10.0 أو أي اصدار آخر .
ومن ثم قم برسمها على الفورم ، وستجد بها العديد من الخيارات وأهمها (هذه الخيارات أغلبها في Custem) .
Value : لتحديد التاريخ المطلوب ومنه يمكن ضبطها بوضع الأمر التالي :

```
Calendar1.Value = DateTime.Calendar
```

* First Day : لتحديد اليوم الأول من الأسبوع .

* Day Length - Month Length : لتحديد طريقة عرض اليوم والشهر .

* Grid Call Efect : لاختيار طريقة من ثلاث طرق للعرض باختلاف التحديد .

ستجد أيضاً العديد من الخيارات لاظهارها واخفائها تحت بند Show .
ومن ثم في Font ستجد ثلاثة أنواع : خط (اليوم من الأسبوع - الأرقام الخاصة بأيام الشهر - العنوان) .
وفي Color ستجد لون (الخلفية - اليوم من الأسبوع - الأيام من الشهر - شيء ما لا أعرفه - العنوان) .
أما عن الأوامر التي يمكن استخدامها فمنها:
NextDay - NextMonth - NextWeek – NextYear (اليوم - الأسبوع - الشهر - السنة) التالية .
وعكسها : PreviousWeek – PreviousYear - PreviousDay - PreviousMonth

صندوق الادخال Input box

يستخدم لعرض نافذة مستقلة وكتابة المعلومات فية .

القاعدة العامة للدالة:

eng = inputbox ("prompt ", "title" ,"الاسم الافتراضي", موقعه في x, موقعه في y)

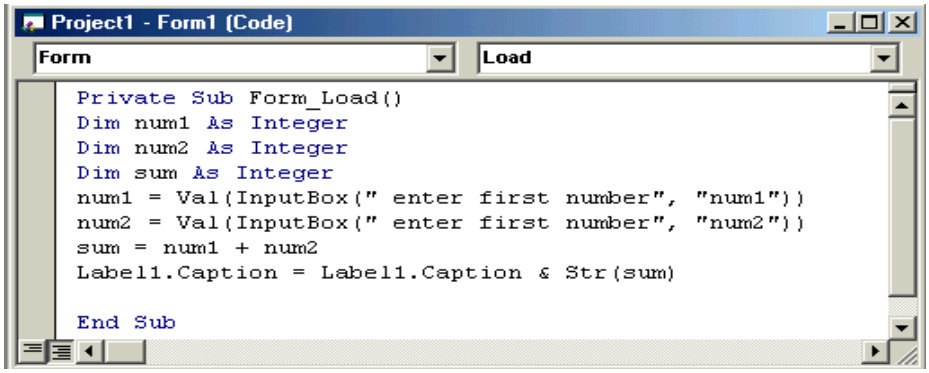
كتابة الموقع اختياري في حالة لم تستعملهما فان InPutBox يظهر في وسط الشاشة، والاسم الافتراضي اختياري ايضا اذا تركته فارغا فان مربع النص يظهر فارغا اثناء التنفيد. .

مثال :-

صمم برنامج باستخدام الدالة inputbox لجمع رقمين.

الحل :-

اكتب الكود المبين في ال (form) في حدث ال (load) .



اضغط (F5) لتشغيل البرنامج

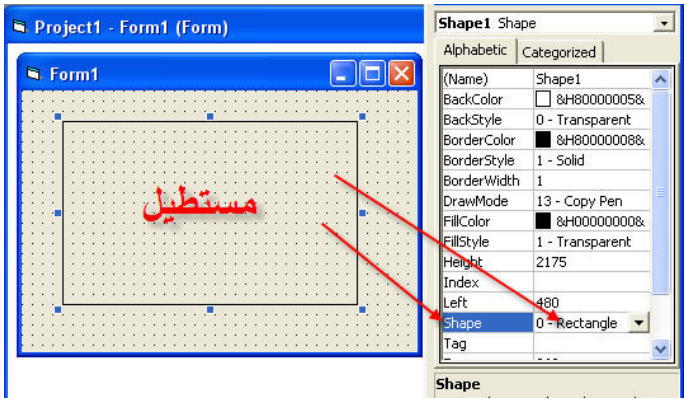
عند كتابة سطر الاوامر

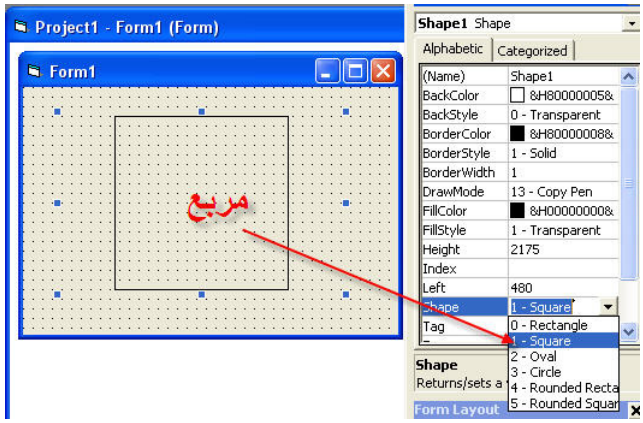
تظهر النافذة السابقة لادخال الرقم الاول او الثاني .

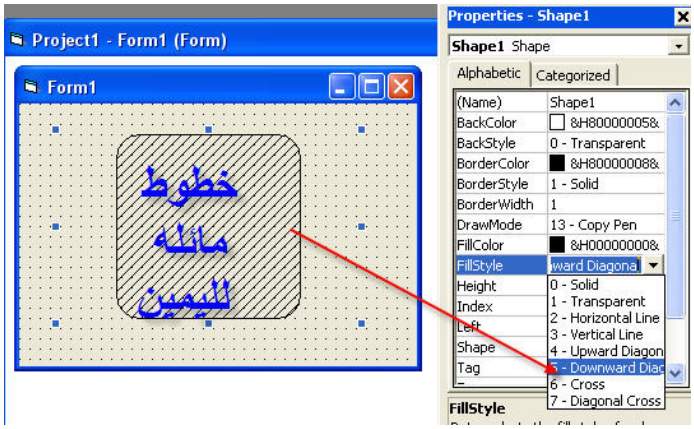
ثم تظهر النافذة السابقة لعرض النتيجة بنافذة مستقلة ايضا .

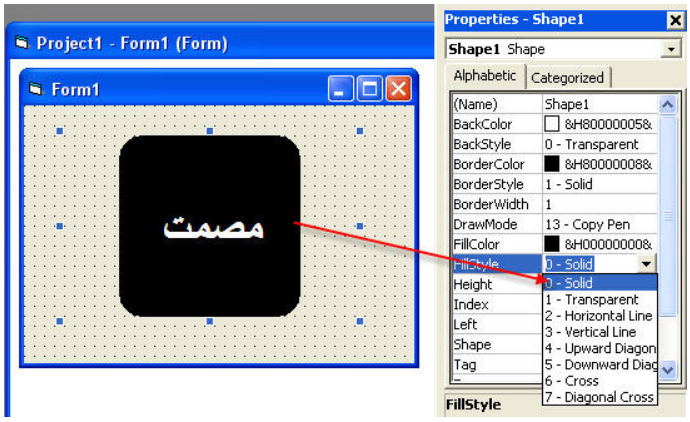
أداة الأشكال(Shape)

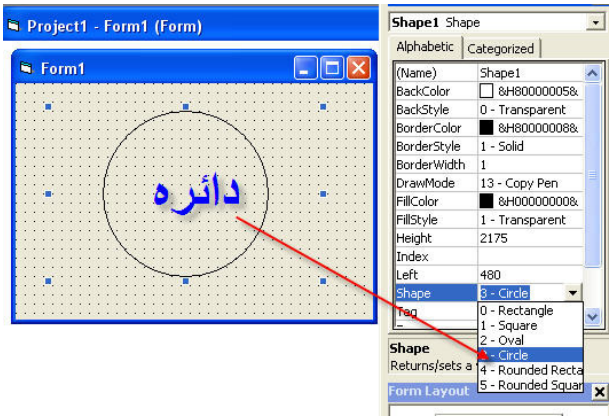
تستخدم لرسم شكل هندسي على الفورم للتجميل فقط وتأخذ أشكال هندسية عديدة منها:

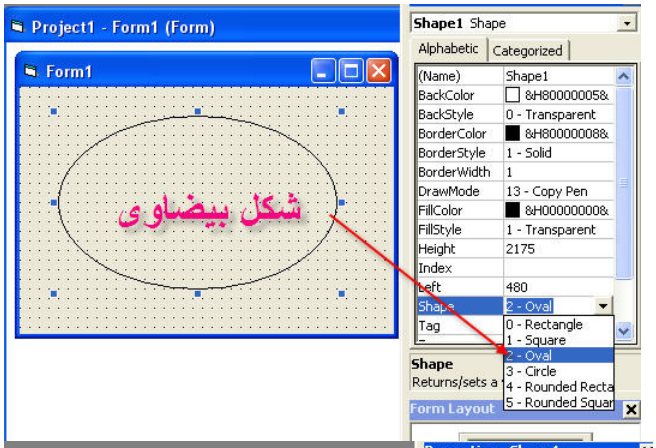


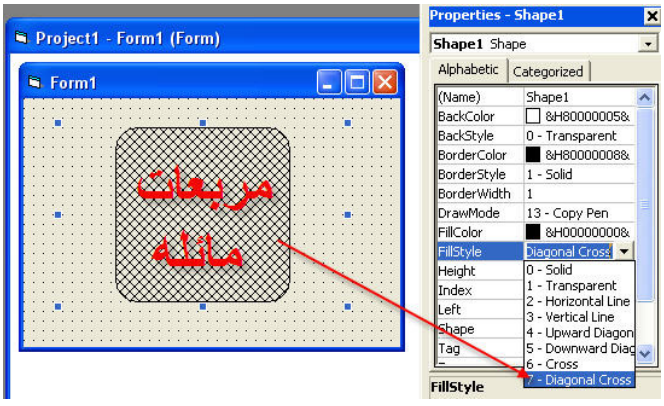


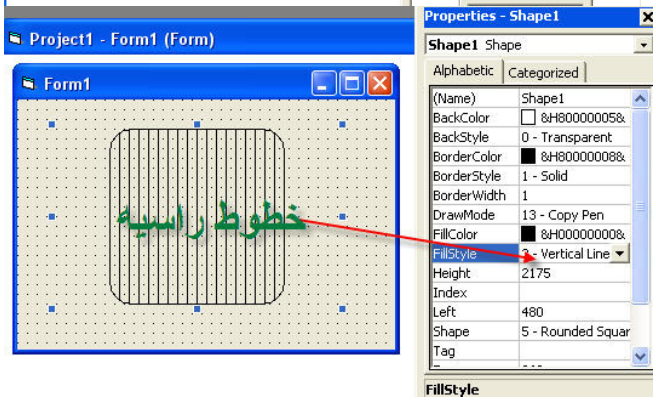












رسم الخط (Line) تستخدم لرسم خط على الفورم وهى أيضا للتجميل فقط .

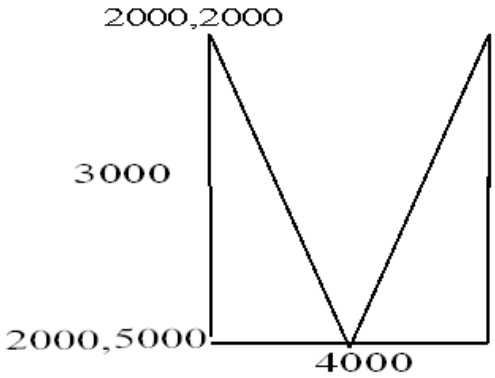
رسم الاشكال الهندسية في البرنامج بواسطة الكود

ر سم مستقيم	Line (x1,y1)-(x2,y2), color ,bf
رسم الدائرة	Circle(X,Y),Radius,Color
رسم المثلث	Line (x1,y1)- (x2,y2) ,color نكرر الكود (3)مرات مع معلومية الاحداثيات
رسم المربع	Line (x1,y1)-(x2,y2),color نكرر الكود (4)مرات مع معلومية الاحداثيات
رسم نقطة	pset(X,Y) ,Color
حركة الفورم	Move X1,Y1
مسح ما مرسوم ومطبوع على الفورم	(Cls) او (Me . Cls) او (form1 . Cls)

ملاحظة :- في الفيجوال بيسك يكون احداثيات ال X وY نحو الاسفل كما مبين من الشكل :-

مثال : -

ارسم الشكل التالي باستخدام الاحداثيات المعلومة ؟

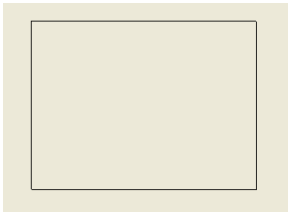


الحل :-

```
Private Sub Command1_Click()
Line (2000, 5000)-(6000, 5000)
Line (6000, 5000)-(6000, 2000)
Line (6000, 2000)-(4000, 5000)
Line (4000, 5000)-(2000, 2000)
Line (2000, 2000)-(2000, 5000)
End Sub
```

مثال :-

ارسم المربع التالي باستخدام الاحداثيات ؟



الحل :-

نكتب الكود التالي في زر (command) .

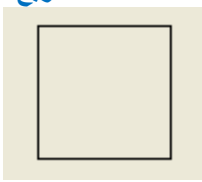
```
Private Sub Command1_Click()
Line (2000, 3000)-(6000, 3000)
Line (2000, 3000)-(2000, 6000)
Line (6000, 6000)-(6000, 3000)
Line (6000, 6000)-(2000, 6000)
End Sub
```

ملاحظة :- لرسم مربع او مستطيل فارغ او ممتلئ نستعمل سطر واحد من الكود.

مثال :-

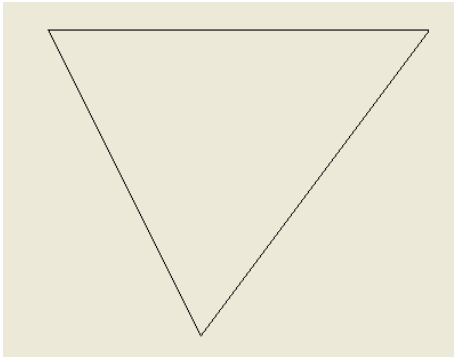
```
Line (1000, 1000)-(2000, 2000), red, BF مربع ممتلئ
```

```
Line (1000, 1000)-(2000, 2000), red, B  مربع فارغ
```



مثال :-

ارسم المثلث التالي باستخدام الاحداثيات ؟

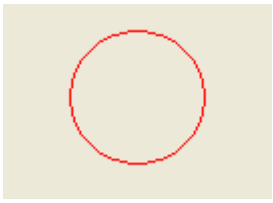


الحل :-

نكتب الكود التالي في زر (command)

```
Private Sub Command1_Click()
Line (2000, 3000)-(7000, 3000)
Line (2000, 3000)-(4000, 7000)
Line (4000, 7000)-(7000, 3000)
End Sub
```

مثال :-
ارسم الدائرة التالية بمعلومية النقطة و نصف القطر .



الحل :-

```
Private Sub Command1_Click()
Circle (2000, 3000), 500, vbRed
End Sub
```

دوال API

ما المقصود بدوال API

كلمة API هي اختصار ل Application Programing Interface ، واجهة برمجة التطبيقات .
تكمن ماهية دوال API في مجموعة من الدوال Functions و الإجراءات Subs الموزعة على مجموعة من ملفات النظام ذات الإمتداد DLL واختصارها (dynamic link library)أي مكتبات الربط الديناميكي .

إستخدام دوال API لا يقتصر على لغة برمجة معينة ،لكن تختلف طرق التعامل معها من لغة لأخرى .
يعتبر دعم لغة Visual Basic لدوال API شيئا مهما في الحد ذاته حيث يسد عجز هذه اللغة البرمجية في مجموعة من الموافق و يزيد بالتالي قوتها وهذا نظرا لإحتواء ملفات DLL الخاصة بالنظام مجموعة من الأوامر الجاهزة و الدقيقة و الغنية التي تغني المبرمج عن كتابة أسطر طويلة من الشيفرات البرمجية وهو قادر على اختصارها فقط في سطرين في بعض الحالات .

و أما الميزة الأساسية لدوال API هي كونها تغني المبرمج عن تكرار نفس الأوامر و الإجراءات في عدة برامج و هو قادر على استعمال ملف صغير يحتوي تلك الأوامر في عدة برامج وغالبا ما يسمى هذا الملف وحدة نمطية أو Modul و يكون بامتداد *.bas .
مهما تحدثنا عن هذه الدوال فهي تبقى قوية،سريعة، و ساحرة في الكثير من الأحيان .

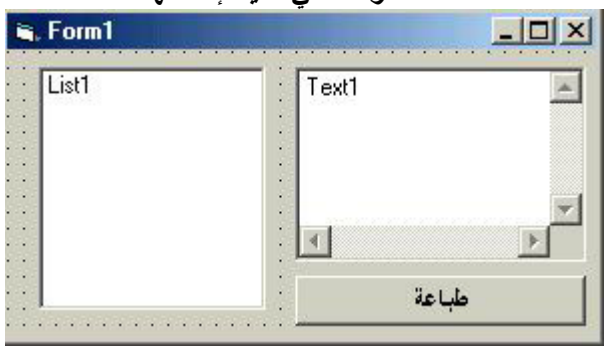
الطباعة باستخدام Print Object

إن الطباعة من الأساسيات في برامجك فأي برنامج يحتاج إلى الطباعة سوف نقدم مثالاً بسيطاً لطباعة نص دون إستخدام توابع (API) .

مثال :-
أولاً يجب أن يكون لديك طابعة إفتراضية مثبتة ولو لم يكن لديك طابعة أضف طابعة وهمية للتجريب عن طريق
لوحة التحكم -> الطابعات -> إضافة طابعة

ثانياً قم باضافة الادوات التالية :
1 - أداة النصوص Text1
2 - الصندوق List1
3 - زر أمر Command1

هذه الأدوات التي عليك إضافتها



كتابة الكود :

1- اضغط (Double Click) فوق الفورم وأكتب الشيفرة التالية في الحدث Load للفورم .

```
Dim a As Integer
For a = 0 To Printers.Count -1
List1.AddItem Printers(a).DeviceName
Next
End Sub
```

شرح الكود :

هذا الكود يقوم بوضع قائمة بأسماء الطابعات المثبتة لديك في الصندوق (List1) لتتمكن من إختيار الطابعة التي تريد الطباعة عليها.

2- اضغط (Double Click) فوق (List1) وأكتب الكود التالي :-

```
Set Printer = Printers(List1.ListIndex)
```

هذا الكود يقوم بتحديد الطابعة من المجموعة المثبتة لديك

3- الأن نبدأ بعملية الطباعة :-

أكتب النص الذي تريد طباعته في Text1

أضف الكود التالي إلى زر Command1:

```
Printer.FontSize = 40
Printer.Print Text1.text
Printer.EndDoc
```

في السطر الأول تحديد حجم الخط

في السطر الثاني إرسال النص إلى الطابعة

في السطر الثالث بدء الطباعة الفعلية

هذه عملية الطباعة البسيطة بإستخدام(Print Object) باستخدام توابع(API) .

التعامل مع الحافظة

سوف تعرف كيف تنسخ النص وتلصق ماهو منسوخ من أي برنامج أخر وكذلك الصور.

فلو إلتقطت صورة عن طريق مفتاح(Print Screen SysRa) فإنك تستطيع لصق الصورة في برنامجك وحفظها .

طريقة العمل ليست صعبة بعد تجهيز الفورم أضف العناصر التالية:

أداة النصوص (Text1) (5)أزرار (Command) غير (Caption) كما في الصورة و أداتين عرض الصور (Picture1) و (Picture2)

كتابة الكود :

في الزر مسح الحافظة أضف السطر التالي :

```
Clipboard.Clear
```

في الزر نسخ النص أضف السطر التالي وهو يقوم بنسخ المعلومات الى الحافظة:

```
Clipboard.SetText Text1.Text
```

في الزر لصق النص أضف السطر التالي:

```
Text1.Text = Clipboard.GetText
```

هذه الأسطر السابقة تقوم بنسخ كافة النصوص ولصقه

أما إذا أردت نسخ جزء محدد فإستبدل (Text1.text) بالكود (Text1.SelText)

نأتي الأن إلى نسخ ولصق الصور:
نفس الطريقة السابقة ولكن إستبدل (Text1.text) بإسم أداة الصور (Picture1)

في الزر نسخ الصورة أضف السطر التالي:

```
Clipboard.SetData Picture1.Picture
```

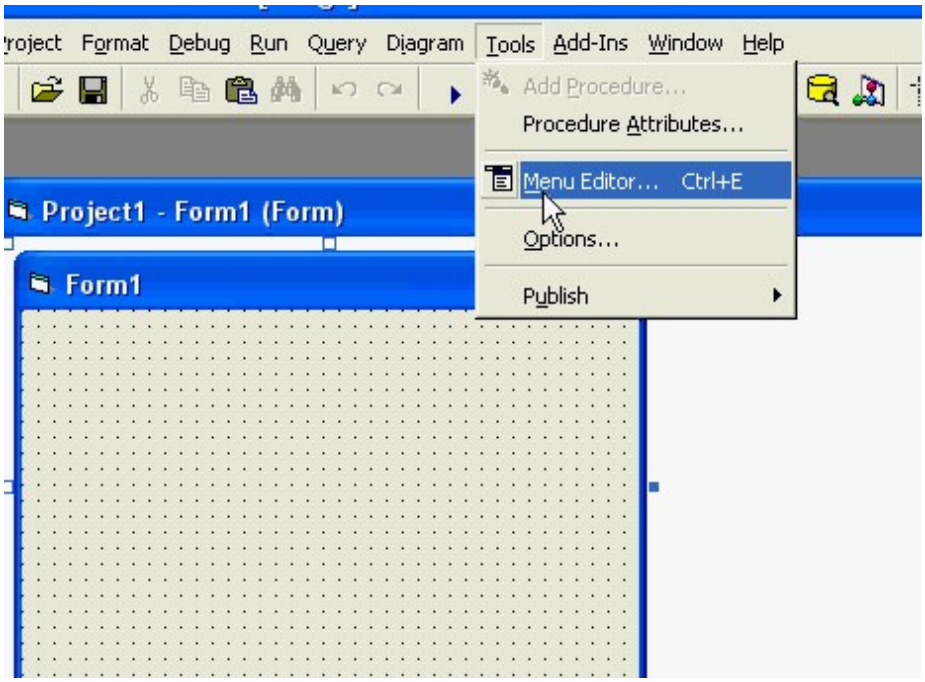
في الزر لصق الصورة أضف الكود التالي :

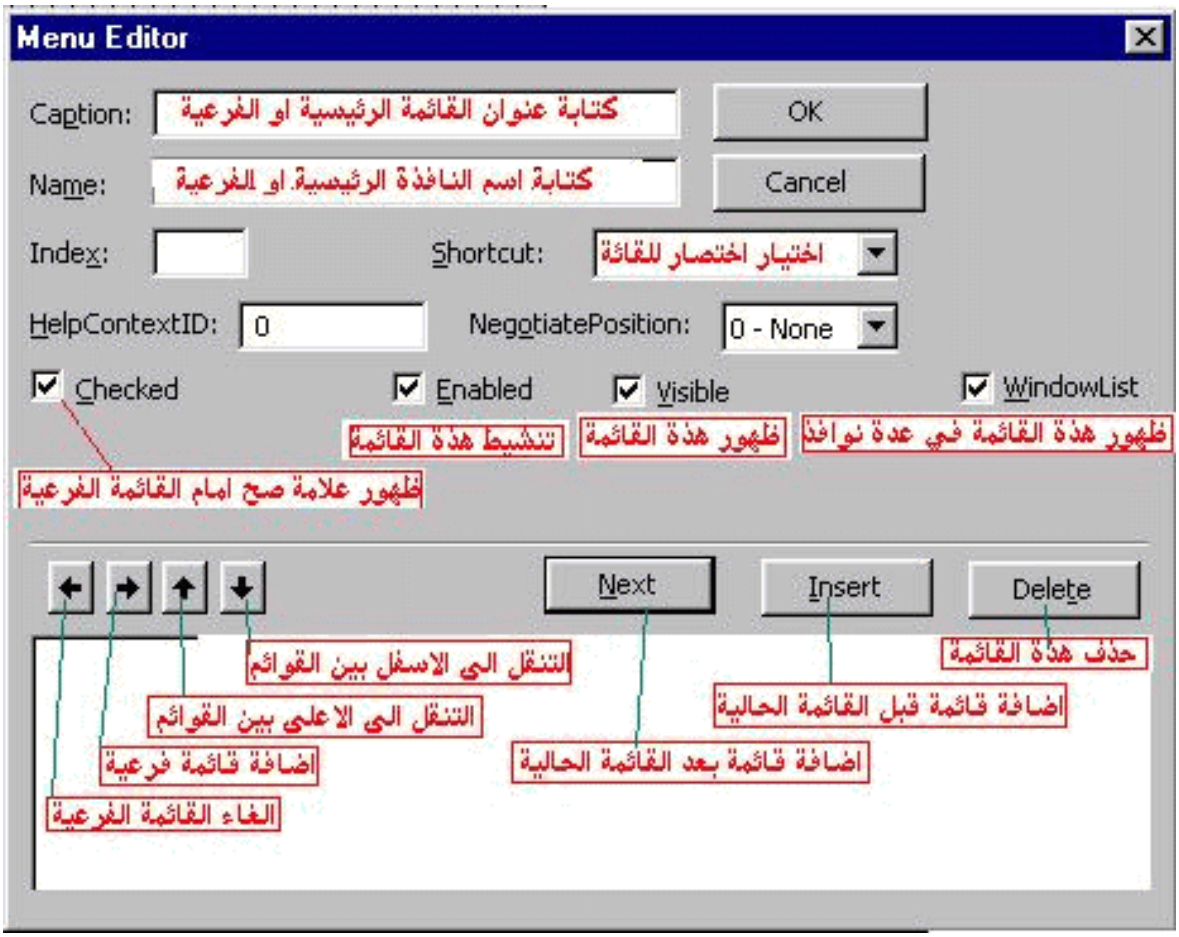
```
Picture2.Picture = Clipboard.GetData
```

محرر القوائم

إن أبسط برنامج تريد صنعه سوف يحتاج بالتأكيد إلى قوائم منسدلة مثلاً : (ملف . تحرير . تعليمات .....)
وهذا الدرس يساعدك على كيفية إنشاء القوائم :

لفتح محرر القوائم أنقر فوق الزر في شريط الأدوات أو من قائمة(Tools)إختر(Menu Edior)فيظهر لك محرر القوائم





الاساسيات المهمة

1- (caption) منها بتكتب اسم القائمة الذي تريدها مثلا (file) اومن الممكن ان تكتب بالعربي ملف مثلا .

2- (name)وفيها تكتب اي رمز حرفي بالانجليزي المهم كل قائمة لها اسم خاص بها او حرف .

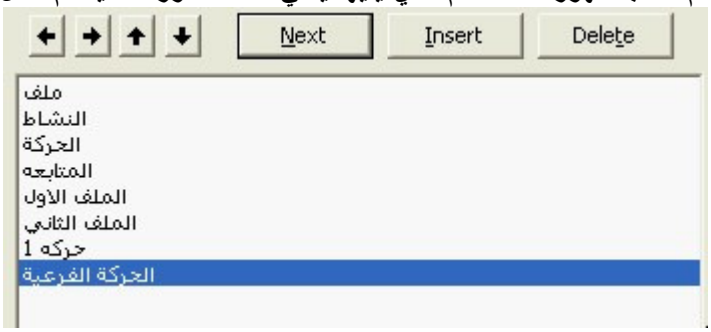
3- (short cut) ومنها تختار اسم اختصاري من الكي بورد للقائمة بشرط ان لا تكون قائمة رئيسية مثل /file / view ولكن المنسدل فقط

4- (next) ومنها تستطيع انشاء قائمة جديدة منها .

5- (delete) ومنها تقوم بحذف القائمة الغير مطلوبه .

6- وهي الاهم : عندك اربع اسهم في اتجاهات مختلفة اول سهمان اتجاههم فوق وتحت

و فائدتهم ترتيب القائمة حسب الاسبقية يعني مثلا انت تعمل قوائم كالتالي (ملف - النشاط - الحركة - المتابعه - الملف الاول - الملف الثاني – حركة1 - الحركه الفرعية)
وبعد كتابتهم نريد ترتيبهم عندما نضغط على القائمة(ملف)يظهر منها (الملف الاول) ومن الملف الاول (الملف الثاني) ونريد عمل قائمة بجانب (ملف) تكون القائم(النشاط) ونكتب بجانب القائمة (النشاط) قائمة اخرى وهي (الحركه) ومن القائمه الحركة(حركة 1 - الحركة الفرعية) . فأنك ستقوم بترتيب القوائم حسب ظهور القائمة ثم الذي يليها يعني مثلا الصورة الاتية تم عمل القوائم



ونريد ظهورها مثل ما نكتب يبقى الترتيب سيكون كالاتي : الذي نريده الاول وبعدها يتفرع منه الثاني كالتالي ولاحظ الفرق بين الترتيب الاتي والصورة السابقة:

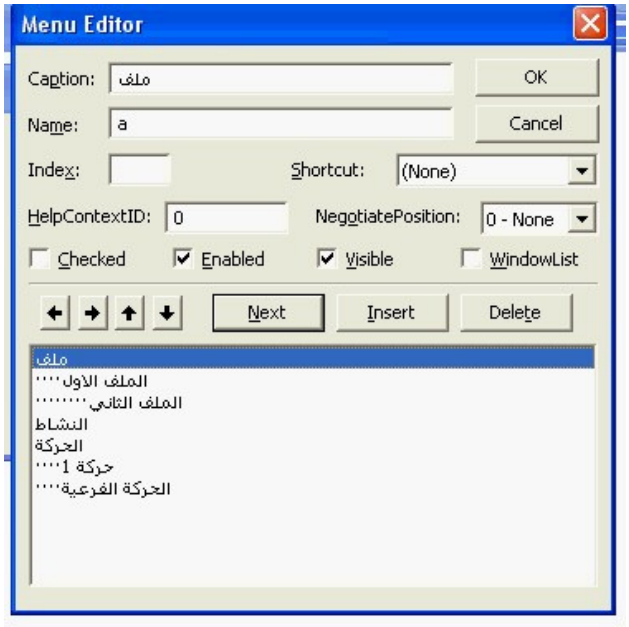
1- الملف
2- الملف الاول
3- الملف الثاني
4- النشاط
5- الحركة
6- حركة1
7- الحركة الفرعيه

وهذه الترتيب بأستخدام السهمان فوق وتحت مع مرعات الوقوف بالماوس على القائمة المراد تحريكها
وجرب هذا ثم اضغط ok وبعدها شغل البرنامج ستلاحظ بان كل القوائم بجنب بعض ولاكن بقي السهمان الاخريين

للسهمان الاخران وهما مهمين جدا لقدعملنا في المثال السابق الاتي :

1-الملف
2- الملف الاول
3- الملف الثاني
4- النشاط
5- الحركة
6- حركة1
7- الحركة الفرعيه

عندما نضغط على ملف يتفرع الملف الاول ثم على الملف الاول ونضغط عليها بالماوس وبعدها ننتقل الى السهم اليمين وضغط عليه مره واحده من القائمة ملف وننتقل الى الملف الثاني ونضغط على السهم اليمين مرتين يعني متفرع من الملف الفرعي الخارج من ملف وهكذا وسيكون شكلهم كالاتي :



من قائمة الحركة يظهر تحتهم حركة (1) وتحتها الحركة الفرعية
وجرب تشغيل البرنامج وعندما نريد عمل اسم مختصر للقائمة الملف الاول من (short cut) تختار اسم لها ولكن لا تستطيع عمل اسم مختصر للقائمة الرئيسية مثل ملف .

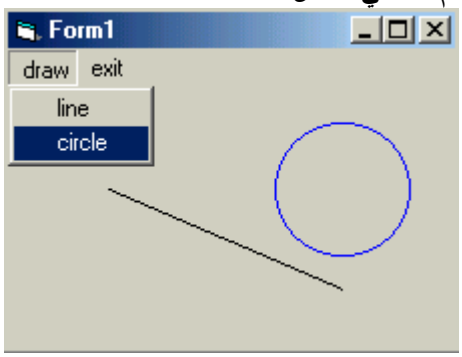
وجرب تشغيل البرنامج سيكون شكلها كالتالي :

ربما سوف تتسائل كيف تظهر القائمة عند النقر بالزر الأيمن للماوس هذا بسيط :
فقط في مكان الحدث عند النقر بالزر الأيمن فوق الفورم مثلاً عل فرض أن الفورم إسمه (Form1)
إختر الحدث (Mouse Down) لل (Form1) وأكتب الكود التالي :

```
Then If Button = 2
menu1 Form1.PopupMenu
End If
```

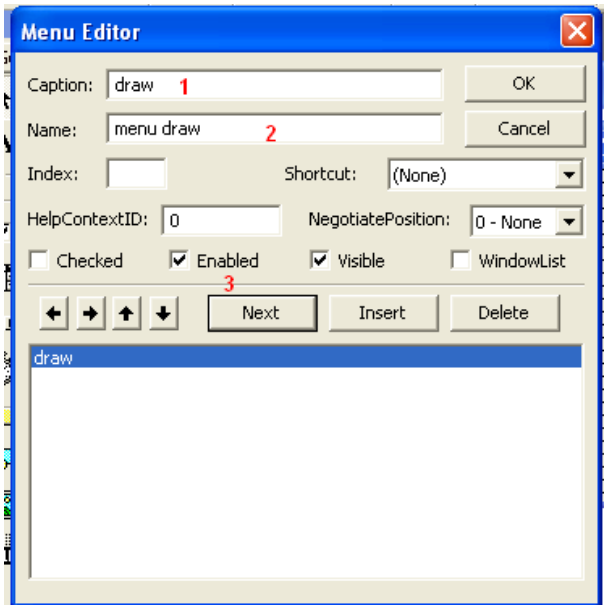
مثال :-

صمم برنامج لرسم مستقيم ودائرة بواسطة القوائم كما في الشكل ؟



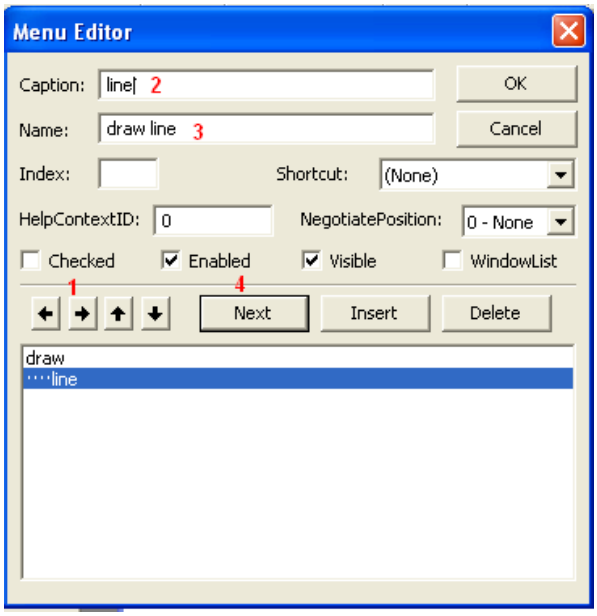
الحل :- نتتبع الاتي :

Tools► menu editor

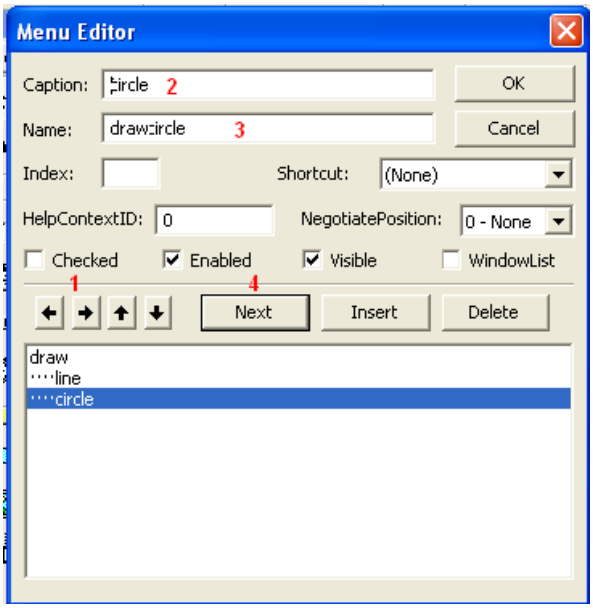


عند الضغط على الزر next تفتح قائمة جديدة ونتبع التالي :

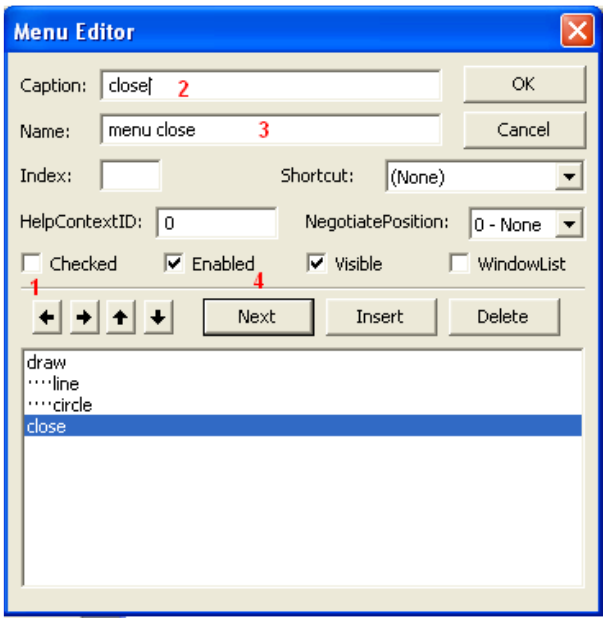
القائمة الفرعية (الخط line):



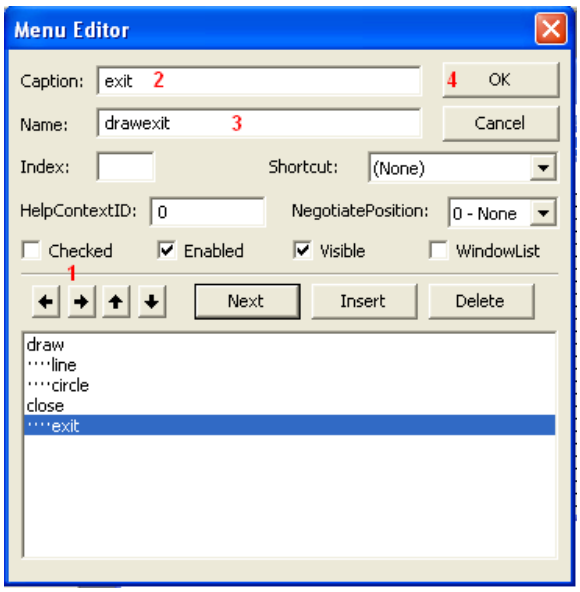
القائمة الفرعية (الدائرة circle) :



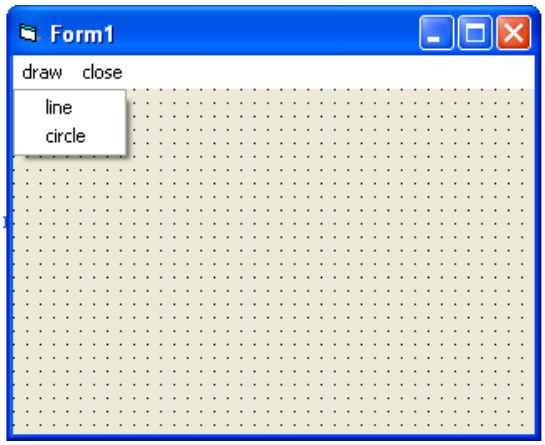
القائمة الرئيسية (الاغلاق close) :-



القائمة الفرعية للاغلاق (الخروج) :-



ويكون الشكل النهائي للقائمة كالتالي :-



الان جاء دور كتابة الاكود للقوائم :-
نحدد القائمة الفرعية line ونكتب فية الكود التالي :-

```
Private Sub drawline_Click()
Line (750, 750)-(2500, 2500), vbBlack
End Sub
```

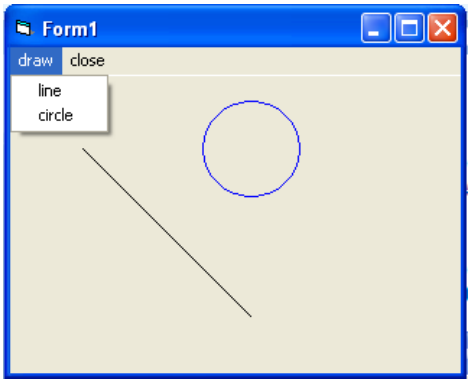
نحدد القائمة الفرعية circle ونكتب فية الكود التالي :-

```
Private Sub drawcircle_Click()
Circle (2500, 750), 500, vbBlue
End Sub
```

القائمة الفرعية Exit

```
Private Sub drawexit_Click()
End
End Sub
```

الشكل النهائي :-



الرسائل Msgbox

الشكل العام

MsgBox "Message", Symbol, "Title"

Message :عبارة عن سلسلة حرفية (string)تمثل الرسالة التي نريدها تظهر في صندوق الرسائل
Title :عبارة عن سلسلة حرفية (string)تمثل عنوان الصندوق الذي يظهر في شريط العنوان للصندوق
Symbol :عبارة عن رمز يمثل قيمة صحيحة (integer)آو ثابتا حرفيا (constant) وتمثل القيمة الصحيحة آو الثابت الحرفي

هذا الجدول يعطيك أمثلة عن الرسائل

الشكل العام	التنفيذ
Msgbox "رسالتك هنا"	
MsgBox " حدث خطأ ", vbCritical, "خطأ"	
MsgBox "رسالتك هنا", vbExclamation, "العنوان"	
MsgBox "رسالتك هنا", vbInformation, "العنوان"	

MsgBox "رسالتك هنا ", vbOKCancel, "العنوان"

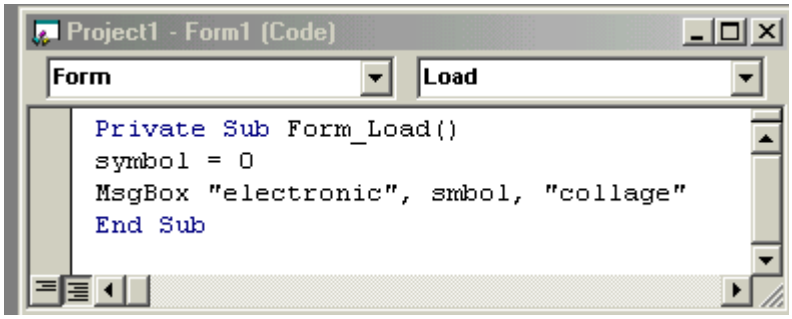
MsgBox "رسالتك هنا ", vbYesNo + vbQuestion, "العنوان"

MsgBox "رسالتك هنا ", vbSystemModal, "العنوان"

MsgBox "رسالتك هنا ", vbAbortRetryIgnore, "العنوان"

مثال :- اكتب الكود اللازم لاظهار النافذة التالية ؟

الحل :- نكتب الكود التالي في الفورم .

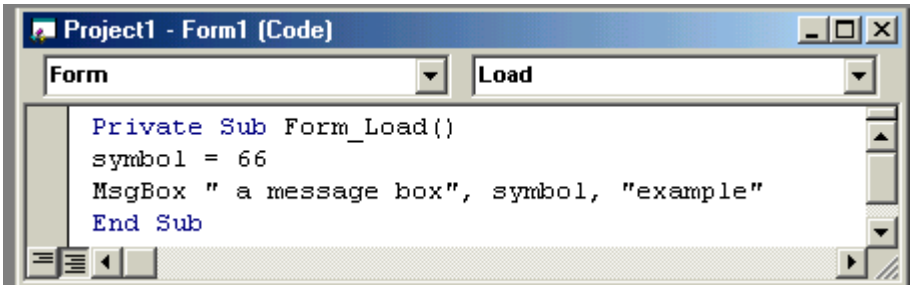


شرح الكود :

ان قيمة الزر ok هي صفر.

مثال :- اكتب الكود اللازم لاظهار النافذة التالية ؟

الحل :- نكتب الكود التالي في الفورم .



ملاحظة:- تستطيع جمع ارقم عدة ادوات لعرضها في الرسالة .

شرح الكود :
ان قيمة علامة المعلومات هي 64 وقيمة الازرار الثلاثة هي 2 و المجموع يساوي 66 .

ملاحظة :- في المثالين السابقين قد عرفنا المتغير symbol ثم كتبنا الكلمة في الكود دون ذكر قيم الرموز.

الجدول التالي يبين الأيقونات وقيمها والثوابت الحرفية التي تمثلها

الاستخدام	شكل الايقونة	القيمة	الثابت
عند حدوث خطأ		16	vbcritical
عند طلب معلومات اضافية		32	vbquestion
عند حدوث خطأ والطلب من المستخدم التدخل		48	vbexclamation
لاخبار معلومة معينة		64	vbinformation

الجدول التالي يبين أزرار الأوامر وقيمها والثوابت الحرفية التي تمثلها

الثابت	القيمة	أنواع الـــCommands
Vbokonly	0	Ok
Vbokcancel	1	Ok, Cancel
vbAbortRetryIgnor	2	Abort, Retry, Ignore
vbYesNoCancel	3	Yes, No, Cancel
vbYesNo	4	Yes, No
vbRetryCancel	5	Retry, Cancel

كيفية وضع عدة أسطر تحت بعضها في الرسالة:-

```
Dim a As String
a = "الأول السطر" & vbCrLf
a = a & "السطر الثاني" & vbCrLf
a = a & "السطر الثالث"
MsgBox a, vbMsgBoxRight + vbInformation, "العنوان"
```

ويمكنك أن تضيف عدد السطور التي تريد بنفس الطريقة
السطر الأخير بدون "&vbCrlf"
لاحظ المكتوب بالاحمر وهو نوع الرسالة وهو يعني إتجاه الرسالة من اليمين لليسار + تلميحات وهذا هام في الرسالة المكتوبة باللغة العربية أي إنه يمكنك إضافة إشارة + بعد النوع الأول لإختيار نوع آخر مثلاً سؤال ("نعم ولا"+ "خطأ") هذه طريقة كتابتها:

```
MsgBox "الرسالة", vbCritical + vbYesNo, "عنوان"
```

وهذه طريقة أخرى

```
Dim a As String
a = "السطر الأول" & vbCrLf & "السطر الثاني" & vbCrLf & "السطر الثالث"
MsgBox a, vbMsgBoxRight + vbInformation , "العنوان"
```

صنايق الحوار Dialog Common

سوف نتحدث عن كيفية استخدام صناديق الحوار CommonDialog ربما لم تسمع بهذه الأداة سابقاً ولاكن سوف تعرفها:

عندما تفتح أي برنامج فإنك سوف تحفظ عملك في النهاية ماذا يظهر لك عندما تختار "حفظ" ؟
وإذ فتحت أي برنامج وتريد فتح ملف فسوف يظهر لك أيضا صندوق الحوار الخاص بالفتح وهو نفس صندوق الحفظ.
هذا الصندوق يسمى (Common Dialog) و هذه الأداة تستخدم في معظم البرامج المبرمجة بواسطة الفيجوال بيسك.

هناك طريقتان لإستخدام هذه الأداة :

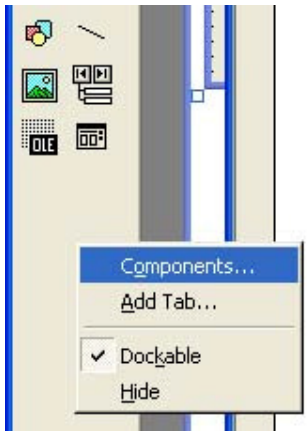
1 - بإستخدام أداة جاهزة : Microsoft Common Dialog Control

2 - بإستخدام إجراءات API : وهذه عملية صعبة لا نتطرق اليها حاليا .

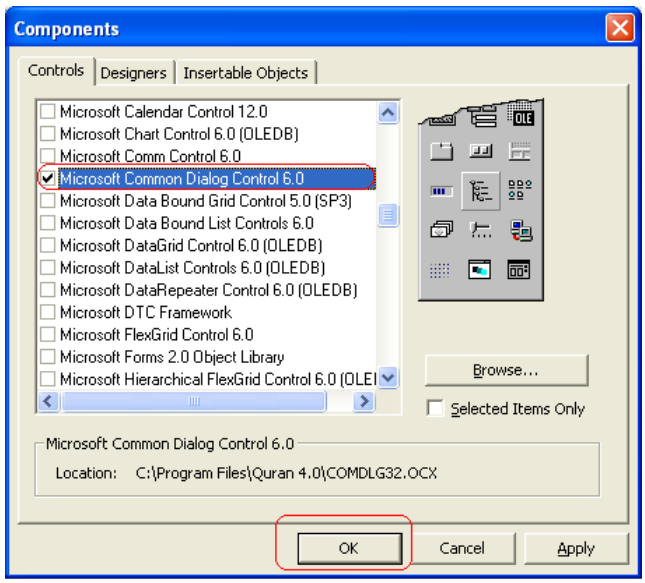
لذلك سنستخدم الأداة : Microsoft Common Dialog Control .

وكيفية اضافة الاداة الى الفورم كالتالي :-

نضغط بزر الماوس الايمن فوق صندوق الأدوات إختر Components



ثم تجد الأداة **Microsoft Common Dialog Control 6.0** حددها وأنقر **Apply**



اهم خصائصها :

1- خاصية الفعل (Action)

وهى التى تحدد نوع مربع الحوار الذى سيظهر للمستخدم وكما ذكرنا سابقا دليل كل رقم من الارقام من (1) الى (6)

2- خاصية المرشح (Filter)

وهى تحدد انواع الملفات التى تريد ان تعرضها فى مربع الحوار فاذا ارت ان تعرض الملفات النصية مثلا تكتب الكود التالى ويمكن تغيير اسم الاداة من خصائصها .

3- خاصية الدليل المبدئى(IntDir)

وهى خاصية اختيارية وهى تستخدم لتحديد الدليل الذى سيعرضه مربع الحوار عند ظهوره كما فى المثال التالى :

مثال :-

ضع الادوات التالية على الفورم (CommonDialog1) ، (Picture) , (Command).

الحل :- ضع الكود التالي في (Command) :

```
CommonDialog1.DialogTitle = "فتح ملف"
CommonDialog1.Filter = "Gif صور|*.gif|JPG صور|*.jpg|Bmp صور|*.bmp|الملفات كافة|*.*"
CommonDialog1.ShowOpen
If CommonDialog1.FileName = "" Then Exit Sub
Picture1.Picture = LoadPicture(CommonDialog1.FileName)
```

شرح الكود:

السطر الأول : تحديد العنوان الذي سوف يظهر في أعلى الصندوق

السطر الثاني : تحديد الملفات التي سوف يعرضها الصندوق

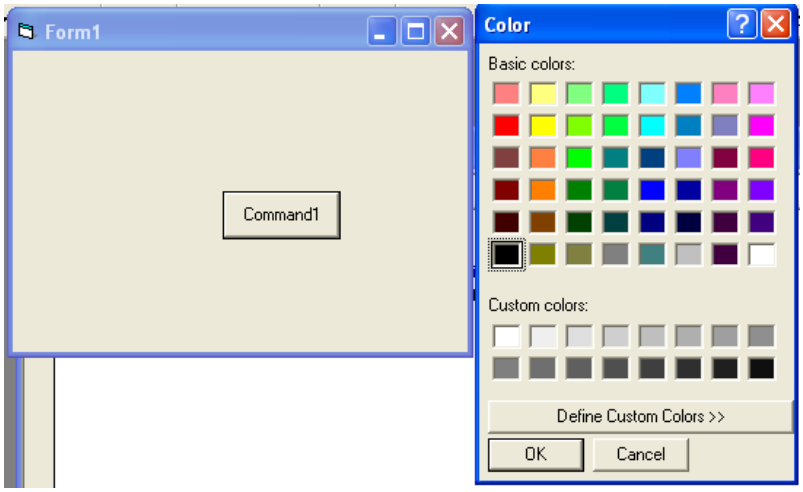
السطر الثالث : إظهار الصندوق

السطر الرابع : إذا لم يتم تحديد ملف ألغي الحدث
السطر الخامس: عرض الصورة التي قمت بتحديدها في الأداة (Plcture1)
كان هذا مثال بسيط لتمسك رأس الخيط مازل هناك العديد من العمليات بواسطة (CommonDialog) منها تحديد الخط والتلوين و خصائص الطباعة و الحفظ .

مثال :-
نفس الأدوات السابقة أضف زر (Command2) وأكتب الكود التالي في (CommonDialog) .
الحل :-

```
CommonDialog1.ShowColor
If CommonDialog1.Color = 0 Then Exit Sub
Picture1.BackColor = CommonDialog1.Color
```

بعد تشغيل البرنامج وعند الضغط على الزر سيظهر نافذة الالوان ونختار لون يكون لون الخلفية للنافذة الحالية :-



شرح الكود :
السطر الأول : أمر بإظهار صندوق اللوان في الأداة CommonDialog
السطر الثاني : شرط باستخدام الجملة if وهو في حال عدم اختيار لون من صندوق الألوان في CommonDialog قم بإنهاء الحدث ولذا استخدمنا exit sub
السطر الثالث : هنا أمر بتلوين خلفية الفورم 1 باللون الذي تم اختياره في صندوق الألوان CommonDialog

مثال :-
على عملية فتح وحفظ مستند نص .

قم باضافة الادوات التالية :
زرين أمر (command)وسمهم (فتح و حفظ)
أداة عرض النص (Label1)
أداة الكتابة (Text1) ولاتنسى تبديل الخصائص (MultiLine) و (ScrollBars)
والأداة (CommonDialog) .

لاحظ الشكل التالي :

كتابة الكود:
اضغط (Double Click) على زر الفتح وأكتب الكود التالي:

```
Dim file As String
CommonDialog1.Filter = "النصوص ملفات|*.txt"
CommonDialog1.ShowOpen
Open CommonDialog1.FileName For Input As #1
Input #1, file
Close
Text1.Text = file
```

شرح الكود :
السطر الأول : التصريح عن متحول نصي
السطر الثاني : تحديد الملفات التي سيعرضها CommonDialog
السطر الثالث : إظهار اللكمون دايلوك
السطر الرابع : والخامس والسادس فتح المستند كمتحول
السطر السابع : عرض مسار الملف في (Label1)
السطر الأخير : عرض النص في أداة الكتابة(Text)
نأتي إلى الحفظ :
ضع الكود التالي في زر الحفظ:

```
Dim a As String
a = Text1
CommonDialog1.Filter = "Text File|*.txt"
CommonDialog1.FileName = a
CommonDialog1.ShowSave
```

شرح الكود :
السطر الأول : التصريح عن متحول نصي (a)
السطر الثاني : إسناد النص المكتوب إلى المتحول (a)
السطر الثالث: تحديد قيمة الكمون دايلوك وهو المتحول(a)
السطر الرابع : عرض مربع الحفظ .
وبذلك نكون قد أعطينا فكرة عن إستعمال CommonDialog، طبعاً هناك العديد من الأحداث لهذه الأداة .

الدوال الجاهزة(Bulit-Function)

الدوال هي أسماء محجوزة ومعرفة من قبل الفيجوال بيسك لتقوم بعمل معين مثل المصفوفات والقيم المطلقة وغيرها.
أو بمعني ابسط هي عبارة عن برنامج صغير مكتوب مسبقا و محفوظ داخل لغة الفيجول بيسك يمكن استدعاءه من خلال برنامجك والاستفادة منه ، إضافة إلى ذلك يمكنك أنت أن تكتب دالة وتستدعيها أكثر من مرة داخل برنامجك . والدوال في الفيجول بيسك تنقسم إلى :

1- الدوال الرياضيةMathematics Functions .
2- دوال سلاسل البياناتString Functions .
3- دوال الوقت و التاريخDate and Time Functions
4- دوال التحقق من أنواع البياناتData Type Inspection Functions .
5- دوال المدخلات و المخرجاتProgram Output and User Input Functions .
6- دوال ماليةFinancial Functions .
7- دوال التحويلاتConversion Function .
8- دوال التعامل مع الفهارس.
9- دوال التعامل مع الملفات.
10- دوال المصفوفات.
11- و دوال متنوعة أخري.
12- دوال معرفة من قبل المبرمج(UDF) User Defined Functions .

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

في ما يلي جدول بالدوال الخاصة بلغة فيجوال بيسك حيث تكون بشكل برامج فرعية جاهزة

الدالة	الوظيفة	مثال
Abs	Absolute Value	Abs (-3) =3
Asc	ASCII code of chaeacter	Asc(“A”) = 65
Chr	Character of ASCII code	Che(65) =A
Date	Current Date as string	Date = 19/11/2007
Format	Date of Number converted to a text string	
Left	Selected left end of a text string	Left (“abcd”,2,3)= bc
Len	Number of character in a string	Len(“Ahmed”)=5
Log	Logarithm of value	Log(1)=0
Mid	Selected portion of a text string	
Now	Current time and date	Now =19/12/2007 10:25:28م
Right	Select right end of a text string	Right(“abcde”,3,2)= bc
Round	Approximate to nearest integer	Round (3.45) =3
Rnd	Random number in range (0-1)	Rnd = 0.7055475
Sqr	Square root of a number	Sqr(9) =3
Str	Number converted to text string	Str (99) =99
Time	Current time and date	Time = 10:25:43م
Val	Numeric value of a gavin string	Val(99) =99

1- دالة المقارنة string compare

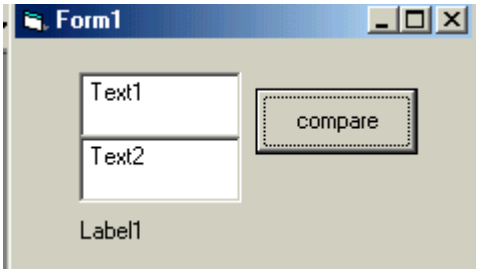
الشكل العام لهذه الدالة هي :-

Str Comp(String 1 , String 2)

مثال :- قارن بين “ زيد “ و “ حسن “

الحل :-

```
Private Sub Command1_Click()
Dim res As Integer
res = StrComp(Text1.Text, Text2.Text)
If res = -1 Then
Label1.Caption = " first  < second "
ElseIf res = 1 Then
Label1.Caption = "first  > second"
Else
Label1.Caption = "first  = second "
End If
```



2 - الدالة العكسية Reverse:

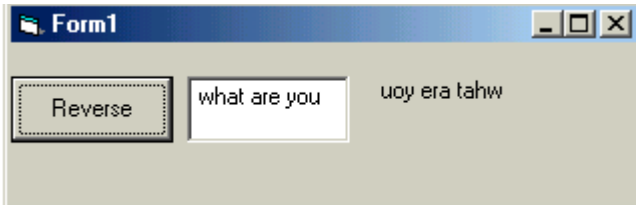
الشكل العام لهذه الدالة هي :

Str Reverse (String)

مثال:- جد عكس التصريح “ what are you ”

الحل :-

```
Private Sub Command1_Click()
Label1.Caption = StrReverse(Text1.Text)
End Sub
```



3- دالة الاستبدال Replace:-

الشكل العام لهذه الدالة هي: Replace (String ,S1,S2)

مثال :-

استبدل كلمة (hussen مع ali)

الحل :-

```
Private Sub Command1_Click()
Label1.Caption = Replace(Text1.Text, "hussen", "Ali")
End Sub
```

4- الدالة الاحرف الكبيرة والاحرف الصغيرة Ucase, Lcase :

الشكل العام لهذه الدالة: Ucase $(Str.Text) Lcase $(Str.Text)

مثال :-

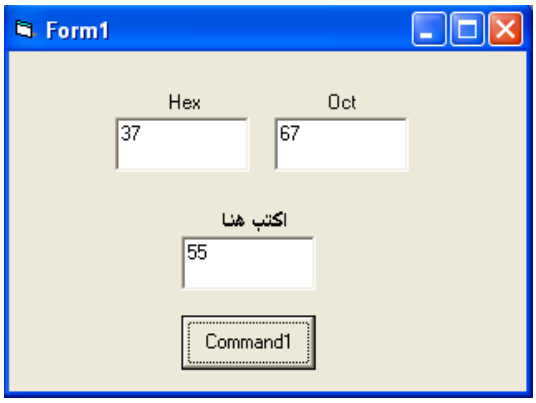
```
Private Sub Command1_Click()
Text2.Text = UCase$(Text1.Text)
Text3.Text = LCase$(Text1.Text)
End Sub
```

5- دالة التحويل من نظام لاخر Convert:

الشكل العام لهذه الدالة هي : النظام السداسي عشر Hex$(text1.text)
النظام الثماني Oct$(text2.text)

مثال :-

```
Private Sub Command1_Click()
Text2.Text = Hex$(Text1.Text)
Text3.Text = Oct$(Text1.Text)
End Sub
```



6- دالة Val :- و هي دالة مشهورة جداً تقوم بتحويل القيمة النصية إلى قيمة عددية ، لو كنت تريد مثلا أن تكتب في نص و ثم تدع البرنامج يضرب الرقم في (10) هذا هو المطلوب :

```
Label1.Caption = Val(Text1.Text) * 10
```

و تحتاج لإضافة أداتي (Text) ، (Label) و زر (Command) يتم تنفيذ بداخله الشفرة و نختار حدث مناسب ألا و هو الضغط على الزر)Click(.

7- الدالة Rnd : - و هي دالة جميلة تقوم بتوليد أرقام عشوائية بين 0 و 1 في كل مرة مثلاً إذا وضعتها في زر و الصيغة العامة

```
Rnd (Number)
```

مثال :-

```
Print Rnd
```

و ستجد أنها تقوم بتوليد أرقام عشوائية بين 0 و 1

ملاحظة :- أذا أردنا أن نولد أرقاماً عشوائية بين 5 و 15 ، فإننا نقوم بكتابة الأمر.

```
Private Sub Command1_Click()
Text1.Text = 5 + (15 - 5) * Rnd
End Sub
```

أما أذا أردت أن يبدأ التوليد من الصفرالى الرقم 5 فيكون الأمر كالآتي:

```
Private Sub Command1_Click()
Text1.Text = 5 * Rnd
End Sub
```

8- الدالة Int :- تقوم بجعل الرقم العشري عدد صحيح مثال :

```
Label1.Caption = Int(1.95667899)
```

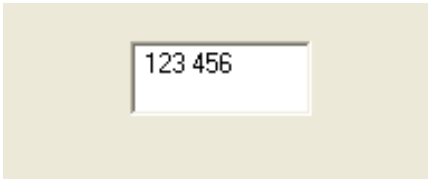
ستكون نتيجة التنفيذ هو الرقم (1) .

9- الدالة Str:- تقوم بتحويل الرقم إلى نص في الكود وهي عكس الدالة VAL.

مثال :-

```
Private Sub Command1_Click()
Text1.Text = Str(123) & Str(456)
End Sub
```

الحل :-



10- الدالة Round:- تقوم بتقريب العدد للمنزلة المطلوبة

```
Label1= Round(2.56789, 1)
```

و هذا معناه إننا نقرب الرقم (2.56842) إلى أقرب منزلة عشرية لانا و ضعنا (1) بعد الرقم و سيعطينا البرنامج الجواب و هو (2.6) و بدون الرقم نقرب لأقرب عدد طبيعي .

مثال :-

```
Label1= Round(2.56789)
 Anewer = 3
```

11- دالة len : - و تستخدم لمعرفة عدد أحرف جملة معينة أو حتى ارقام مثل التالي :

```
Label2.Caption = Len("حسن")
Anewer = 3
```

و ستكون النتيجة هي الرقم (3) لان عدد الأحرف (3) أرقام

12- دالة Trim :- و تفيد في انك إذا كانت لديك جملة نصية أو رقمية و احتمال كتب المستخدم في بدايتها مسافة أو حتى في نهايتها فبإمكانك محو المسافة من الجملة بهذه الدالة مثل :-

```
Label1.Caption = Trim(Text1.Text)
```

13- دالة LTrim :- تمحو المسافات من الجهة اليسرى left للنص:-
اضف الكود التالي الى اداة command .

```
Label1.Caption = LTrim(Text1.Text)
```

14-دالة RTrim :- تمحو المسافات من الجهة اليمنى right للنص:-
اضف الكود التالي الى اداة command .

```
Label1.Caption = RTrim(Text1.Text)
```

15- دالة Left :- تقوم بعرض عدد معين من الأرقام أو الأحرف حسب الطلب مثل المثال التالي تقوم بقص رقم واحد من بداية النص أو الرقم :

```
Label1.Caption = Left(Text1.Text, 1)
```

16- دالة Right :- تقوم بعرض عدد معين من الأرقام و الأحرف حسب الطلب مثل المثال التالي تقوم بقص رقم واحد من نهاية النص أو الرقم :

```
Label1.Caption = Right(Text1.Text, 1)
```

17- دالة Mod :- تقوم بعرض باقي القسمة مثل باقي قسمة (4 على 3 هو 1 فناتج العبارة سيكون 1 :

```
Label1.Caption = 4 Mod 3
```

18- دالة chr :- تقوم بقراءة الرقم في شفرة ascii ويعطينا الحرف المقابل له .
الصيغة العامة .

```
Print chr(65)
```

ويعطينا الناتج وهو الحرف A

19- دالة التحويل من النظام العشري الى النظام الثنائي :-

```
Private Sub Command1_Click()
MsgBox Dec2Bin(Text1.Text)
End Sub
Function Dec2Bin(ByVal n As Long) As String
Do Until n = 0
If (n Mod 2) Then Dec2Bin = "1" & Dec2Bin Else Dec2Bin = "0" & Dec2Bin
n = n \ 2
Loop
End Function
```

20- الدالة Sin :- و هي تقوم بحساب جيب الزاوية المعطى للبرنامج بالنظام الدائري و الصيغة العامة لها Sin(Number) .

21- الدالة Cos :- و هي تقوم بحساب جيب التمام الزاوية المعطى للبرنامج بالنظام الدائري و الصيغة العامة لها Cos(Number) .

22- الدالة Tan:- و هي تقوم بحساب الظل الزاوية المعطى للبرنامج بالنظام الدائري و الصيغة العامة لها Tan(Number) .

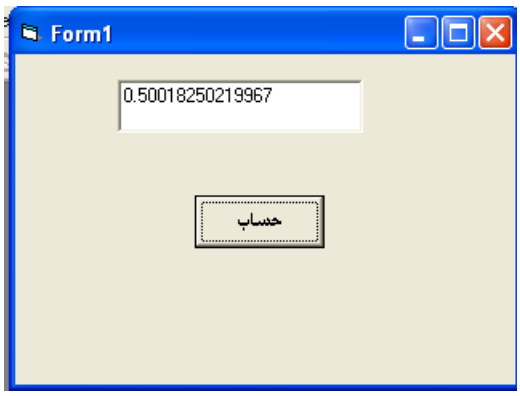
الآن إذا أردت أن تجرب الدالة Sin لإيجاد الجيب 30 و الذي هو 0.5 ستجد أن الفيجوال بيسك سيعطيك أرقام كبيرة و غريبة و السبب هو أن الفيجوال بيسك 6.0 يستخدم النظام الدائري Radian لكي يجد الدوال المثلثية و نحن بالغالب نستعمل نظام الدرجات Degree لذلك سنضطر إلى التحويل من الدائري للدرجات.

للتحويل بين النظامين سنستعمل قيمة بايPi و للتحويل من Radianإلى Degree يجب علينا ضرب الزاوية ب (باي pi) مقسومة على (Pi/180) ثم نجد ما نريده من الدوال

مثال :-
لإيجاد جيب الزاوية 30 اضف اداة text و command واكتب الكود التالي في اداة الزر .
الحل :-

```
Text1.Text = Sin(Val(Text1.Text) * pi / 180)

'Answer is 0.5
```



و لكن عليك أن تعرف الثابت pi في القسم العام .

```
Const pi = 22 / 7
```

التعامل مع الأخطاء

إذا كنت تريد عمل أي برنامج بسيط فلابد أن يحصل في بعض الأحداث الأخطاء وهذاالخطأ سوف يتسبب في إغلاق البرنامج .
للتصريح عن الخطأ وتجاهله :
الكود القياسي لتجاهل الأخطاء هي :

```
On Error Resume Next
```

تكتب قبل الحدث وتقوم بإلغاء الحدث عند وقوع خطأ اي تنتقل الى الخطوة الثانية عند حدوث اي خطا في البرنامج .
وهناك طريقة أخرى وهي عمل حدث كرسالة مثلاً عند وقوع الخطأ
أولا عليك التصريح عن الخطأ كمايلي :

```
On Error Goto R
```

(R) هي اسم الخطأ يمكنك تغييره لما تريد
بعد ذلك أكتب الحدث مثلاً (Form2.Show) اذا لم يكن الفورم (2) موجوداً سوف يتم اغلاق البرنامج
فعليك أن تكتب التعليمة بالشكل التالي

```
On Error GoTo R
Form2.Show
Exit Sub
R:
MsgBox "في البرنامج حدث خطأ", vbCritical, "غير موجود"
```

شرح الكود
في السطر الأول التصريح عن الخطأ
في السطر الثاني الحدث الأساسي
في السطرالرابع والخامس يتم تحديد فعل المتحول
يمكنك تحديد أي حدث للمتحول بدلاً من الرسالة أو يمكنك أن تكتب السطر التالي :

```
MsgBox Err.Description
```

أي أن الرسالة ستكون من أصل الخطأ

مثال :-

اصنع (Form) أضف اليه أداة (Drive1) "محركات الأقراص" وأداة (Dir1) "صندوق المجلدات "
أنقر فوق : (Drive1) وأضف الأسطر التالية

```
On Error GoTo error1
Dir1.Path = Drive1.Drive
Exit Sub
error1:
MsgBox "الرجاء إدخال قرص في محرك الأقراص هذا", vbCritical, "خطأ"
```

شرح الكود

السطر الأول التصريح عن الخطأ
السطر الثاني الحدث الأساسي "عرض المجلدات داخل محرك الأقراص المحدد"
الأسطر الأخيرة عمل المتحول وهو عرض رسالة خطأ
ويمكنك الإستغناء عن الرسالة السابقة بالرسالة الأساسية للخطأ بالتعليمة التالية :

```
MsgBox Err.Description
```

هذا مجرد شرح بسيط عن الأخطاء لكي يكون برنامجك نظيفاً خالياً من أي خطأ (Error)

تشغيل الملفات

هناك عدة طرق لتشغيل الملفات في الويندوز من داخل الفيجوال بيسك وسوف نتطرق الى اهم الطرق لتشغيل الملفات التنفيذية والملفات الأخرى.

* تشغيل الملفات التنفيذية الذي تنتهي بالامتداد (EXE) .

1 - بإستخدام الدالة (Shell) :-
هذه الدالة تستخدمها الفيجوال بيسك لتشغيل الملفات التنفيذية .

صيغتها العامة كما يلي.

```
Shell "File Name Path", WindowsStyle
```

File Name Path : مسار الملف التنفيذي المراد تشغيله.

WindowsStyle : طريقة عرض نافذة البرامج المراد تشغيله . وهذا الجدول يبين الثوابت لهذا الوسيط:

الثابت	قيمته	شرحه
vbHide	0	تكون النافذة مخفية
vbNormalFocus	1	تكون النافذة في حجمها الطبيعي مع إعطاء التركيز لها
vbMinimizedFocus	2	تكون النافذة في وضع التصغير على شريط المهام وإعطائها التركيز
vbMaximizedFocuse	3	تكون النافذة في وضع التكبير مع إعطائها التركيز
vbNormalNoFocus	4	تكون النافذة في وضع التصغير دون إعطائها التركيز
vbMinimizedNoFocus	5	تكون النافذة في وضع التصغير دون إعطائها التركيز

مثال : - لتشغيل المفكرة في الوضع العادي ؟

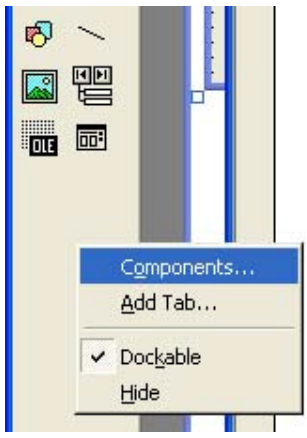
```
Shell "notepad.exe", vbNormalFocus
```

لاحظ انه لم نكتب المسار الكامل للمفكرة لأنها داخل مجلد النظام .

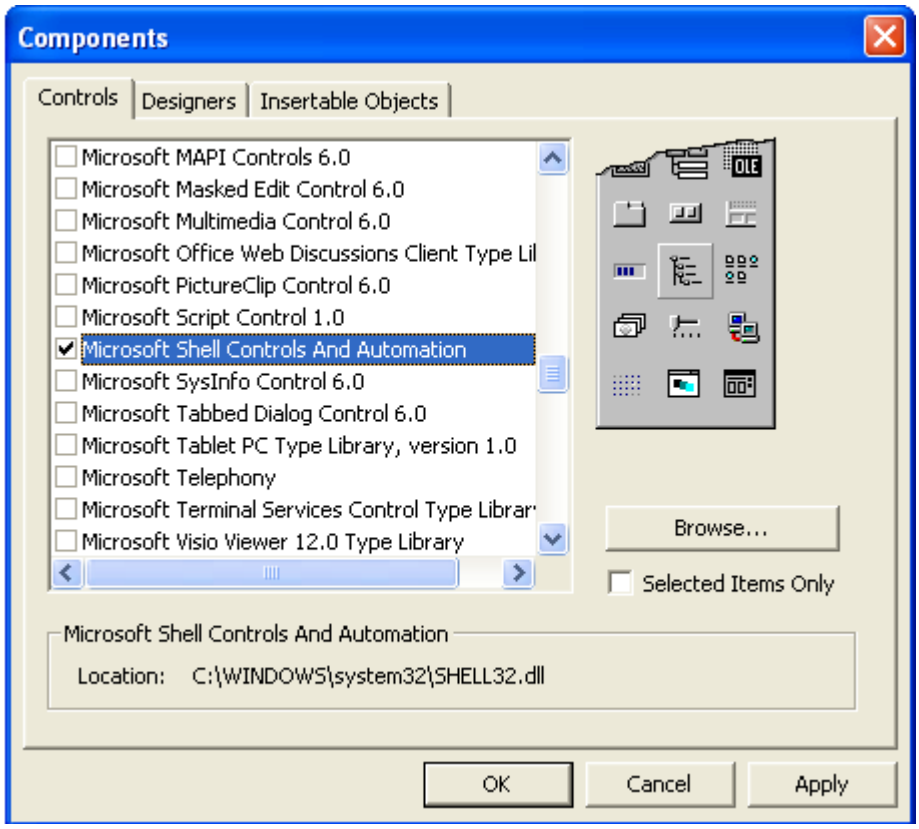
مكتبات DLL وما هي اهميتها

إننا نسمع عن ملفات ذات الصيغة DLL وهي اختصار (dynamic link library) ولكن هل نعلم ماذا تقدمة هذه الملفات الى برامجنا ليس من الممكن العمل على مشروع كبير واكمالة من دون أن نستخدم أي من هذه الملفات ضمن المشروع وسوف نقوم بشرح بعض اوامر مكتبة من مكتبات DLL العديدة التي يتيحه لك الفجول بيسك والتعامل معها لتسهيل المهام والعقبات العديدة التي تحول دون أكمال مشروعك .

كيفية اضافة اداة shell الى البرنامج كالتالي .
كليك يمين فوق صندوق الأدوات إختر Components



ثم تجد الأداة Microsoft Shell Controls And Automation حددها وأنقر Apply



هذه المكتبة عبارة عن ملف DLL موجود في مجلد system32 وتقوم هذه المكتبة بتشغيل الكثير من اليات التحكم الخاصة بويندوز كما يوجد بها جميع ايقونات النظام وتحتوى ايضاً على جميع مربعات رسائل النظام ألان قم بوضع علامة صح في الخانة الفارغة أمام اسم المكتبة ثم قم بالضغط علىOK بعد ذلك قم بالدخول الى شاشة الكود وفي خانة التصريحات العامة قم بالتصريح عن متغير من نوع Shell بهذه الطريقة

```
Dim ENG As New Shell
```

وألان أصبح لدينا الحق في طلب أي تطبيق من تطبيقات التحكم التابعة للويندوز باستخدام أوامر هذه المكتبة ولكن ما تلك التطبيقات التي من الممكن إن تتيحها لنا هذه المكتبة ، في الكثير من البرامج نلاحظ وجود مربعات استكشاف للمجلدات كالذي بالصورة :

سوف نقوم ألان بعمل واحد مماثل له بالضبط

مثال :-
قم بإضافة CommandButton الى الفورم وغير الخاصية Caption الى (اختيار مجلد)
ثم قم بإضافة TextBox عدد 2 الى الفورم واجعل الادوات على الفورم بهذة الطريقة :

والان قم بوضع هذا الكود في Command

```
Dim ShFB As Folder
On Error Resume Next

Set ShFB = SH.BrowseForFolder(hWnd, "please choose a folder and click OK!", 1)
With ShFB.Items.Item

Text1.text = .Path
Text2.text = "Name: " & .Name & vbCrLf & "Type: " & .Type & vbCrLf _ &
"Last Modified: " & .ModifyDate & vbCrLf & "Parent : " & .Parent & vbCrLf"
End With
```

ملاحظة : - الـ (Vbcrlf) تقوم باضافة سطر جديد

الان قم بتشغيل البرنامج وقم بالضغط على الزر اختيار مجلد

ومن قائمة المجلدات اختر مجلد واضغط موافق سوف تلاحظ ارجاع مسار المجلد الذي اخترتة الى text1.text
وارجاع التفاصيل الخاصة بالمجلد الى text2.text والان اصبح باستطاعتك عمل مستكشف في برنامجك يقوم بالكثير من المهام التي توليها الية .

والان ماذا ايضا تستطيع هذة المكتبة تقديمة :

كل المهام هنا :

قائمة التشغيل

SH.FileRun

المساعدة

SH.Help

البحث

SH.FindFiles

البحث عن جهاز

```
SH.FindComputer
```

خصائص شريط المهام

```
SH.TrayProperties
```

خصائص الوقت والتاريخ

```
SH.SetTime
```

مربع حوار ايقاف التشغيل

```
SH.ShutdownWindows
```

الذهاب للانترنت

```
SH.Open  "هنا اكتب عنوان موقع الانترنت الذي ترغب بفتحة"
```

فتح المستكشف

```
SH.Explore "Path"
```

فتح قرص او مجلد

```
SH.Open "Path"
```

ايضا تستطيع فتح أي عنصر في لوحة التحكم والاوامر كاملة هنا

```
SH.ControlPanelItem "ALSNDMGR.CPL"
SH.ControlPanelItem "appwiz.cpl"
SH.ControlPanelItem "bthprops.cpl"
SH.ControlPanelItem "desk.cpl"
SH.ControlPanelItem "firewall.cpl"
SH.ControlPanelItem "hdwwiz.cpl"
SH.ControlPanelItem "inetcpl.cpl"
SH.ControlPanelItem "intl.cpl"
SH.ControlPanelItem "irprops.cpl"
SH.ControlPanelItem "main.cpl"
SH.ControlPanelItem "mmsys.cpl"
SH.ControlPanelItem "ncpa.cpl"
SH.ControlPanelItem "netsetup.cpl"
SH.ControlPanelItem "nusrmgr.cpl"
SH.ControlPanelItem "nwc.cpl"
SH.ControlPanelItem "odbccp32.cpl"
SH.ControlPanelItem "powercfg.cpl"
SH.ControlPanelItem "sysdm.cpl"
SH.ControlPanelItem "telephon.cpl"
SH.ControlPanelItem "timedate.cpl"
SH.ControlPanelItem "wscui.cpl"
SH.ControlPanelItem "wuaucpl.cpl"
SH.ControlPanelItem "cmicnfg.cpl"
SH.ControlPanelItem "alsndmgr.cpl"
SH.ControlPanelItem "ALSNDMGR.CPL"
SH.ControlPanelItem "ImageDrive.cpl"
SH.ControlPanelItem "CMICNFG.CPL"
SH.ControlPanelItem "sapi.cpl"
SH.ControlPanelItem "CMICNFG.CPL"
SH.ControlPanelItem "ALSNDMGR.CPL"
SH.ControlPanelItem "ODBCCP32.CPL"
```

2- باستخدام اجراءات واجهة برمجة التطبيقات API , الدالة WinExec

الوظيفة :-
تشبه الدالة السابقة تماما التصريح للدالة :

مثال :-

```
Private Sub Command1_Click()
eng = WinExec("Notepad.exe c:Test.txt", vbMaximizedFocus)
End Sub
```

ولكن يجب التصريح عن الكود التالي في (module)

```
Declare Function WinExec& Lib "kernel32" (ByVal _
lpCmdLine As String, ByVal nCmdShow As Long)
```

شرح الكود :

عند الضغط على الزر يتم فتح النص text الموجود في الدرايف c بنافذة كبيرة مع التركيز حسب الجدول .

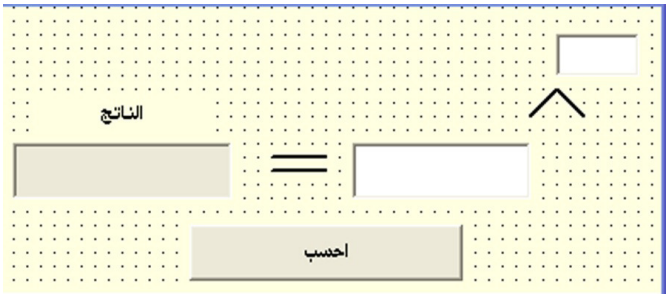
ويوجد دوال اخرى وطرق اخرى لفتح البرامج الاخرى في الفيجوال بيسك .

الرياضيات في الفيجوال بيسك

هو عبارة عن تصميم برامج لحل معادلات الاشكال الهندسية حسب قانون كل شكل مع تعريف المتغيرات لكل برنامج .

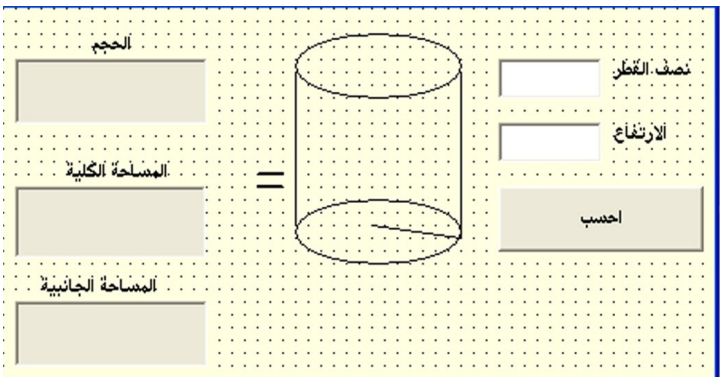
امثلة على حل المعادلات في الفيجوال بيسك

1 – صمم الشكل الاتي (للرفع الى القوى (الاس) .



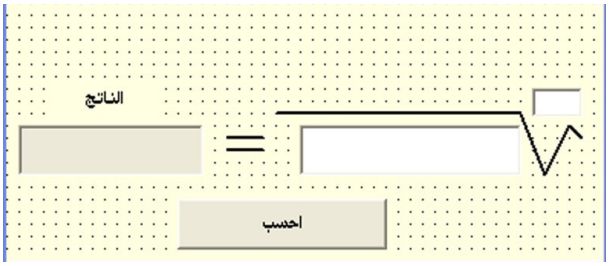

```
Dim x As Double
Dim y As Double
x = Val(Text1.Text)
' الرقم
y = Val(Text2.Text)
' الاس
Label1.Caption = x ^ y
' الناتج
```

2 – صمم الشكل الاتي لايجاد (حجم الاسطوانة) .



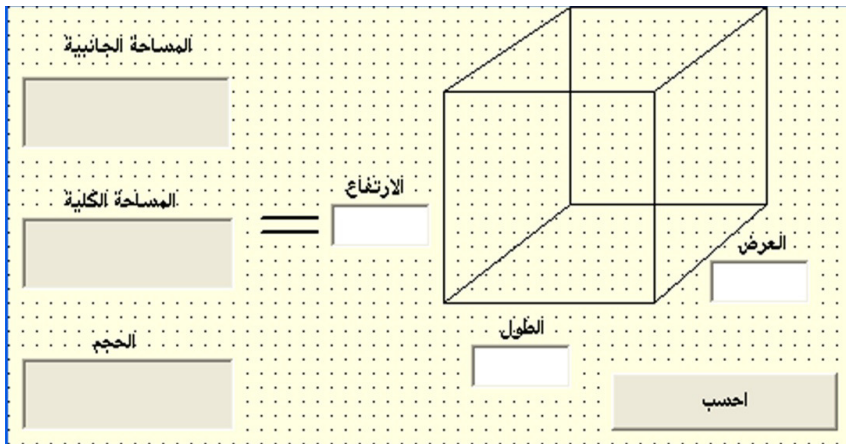

```
Dim x As Double
Dim y As Double
x = Val(Text1.Text)
' نصف القطر
y = Val(Text2.Text)
' الارتفاع
Label3.Caption = (x ^ 2) * (3.14) * y
' الحجم
Label5.Caption = (2 * x * 3.14 * y)
' المساحة الكلية
Label7.Caption = ((2 * x * 3.14 * y) + (2 * ((x ^ 2) * (3.14) * y))
' المساحة الجانبية
```

3 – صمم الشكل الاتي لايجاد (الجذور) .



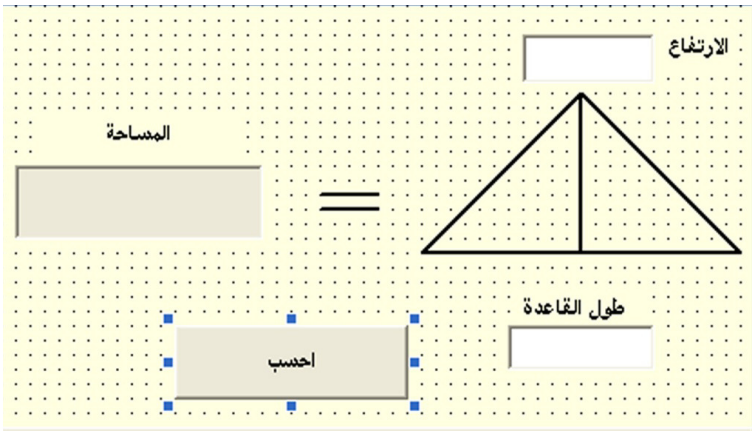

```
Dim x As Double
Dim y As Double
x = Val(Text1.Text)
' تحت الجذر
y = Val(Text2.Text)
' القوى
Label1.Caption = y ^ (1 / x)
' الناتج
```

4 – صمم الشكل الاتي لايجاد (حجم و مساحة متوازي المستطيلات و المربع) .



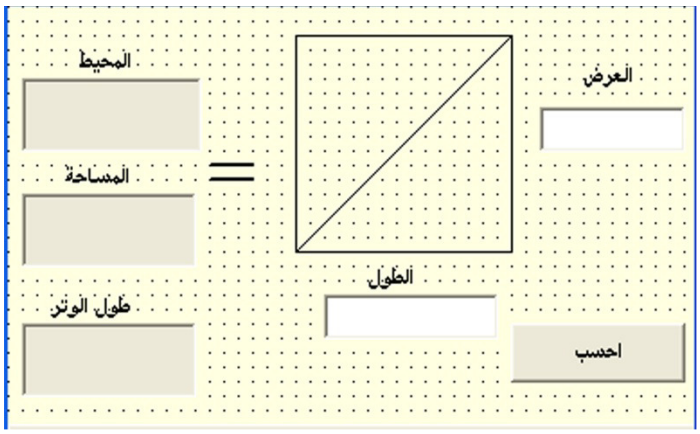
```
Dim x As Double
Dim c As Double
Dim v As Double
x = Val(Text1.Text)
' العرض
c = Val(Text2.Text)
' الطول
v = Val(Text3.Text)
' الارتفاع
Label4.Caption = (2 * x * v) + (2 * x * c) + (2 * c * v)
' المساحة الجانبية
Label5.Caption = (x * v * c)
' المساحة الكلية
Label8.Caption = (2 * x * c) + (2 * c * v)
' الحجم
```

6 – صمم الشكل الاتي لايجاد (مساحة المثلث) .



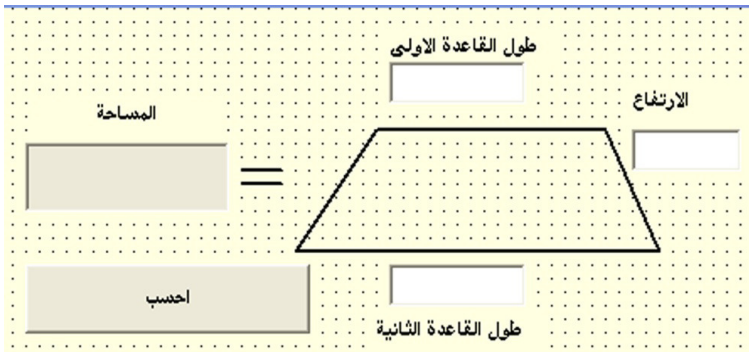

```
Dim x As Double
Dim y As Double
x = Val(Text1.Text)
' الارتفاع
y = Val(Text2.Text)
' ول القاعدة
Label3.Caption = (1 / 2 * x * y)
' المساحة
```

7 – صمم الشكل الاتي لايجاد (مساحة المستطيل او المربع) .



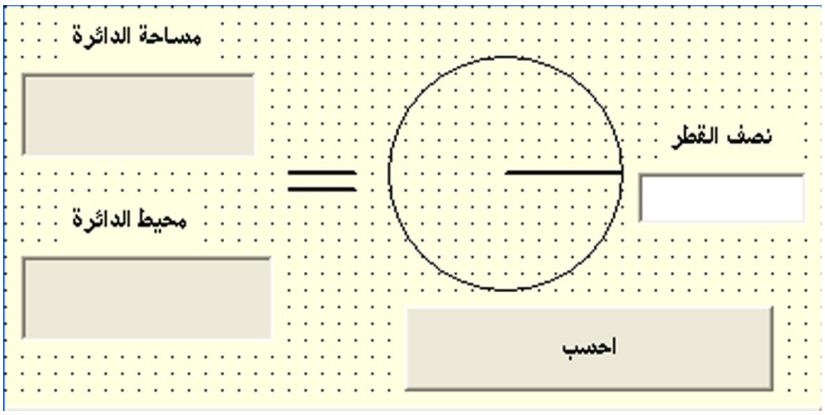
```
Dim x As Double
Dim y As Double
x = Val(Text1.Text)
' الطول
y = Val(Text2.Text)
' العرض
Label3.Caption = (x * y)
' المحيط
Label5.Caption = Sqr((x ^ 2) + (y ^ 2))
' المساحة
Label7.Caption = ((x + y) * 2))
' ول الوتر
```

8 – صمم الشكل الاتي لايجاد (حساب مساحة شبه منحرف) .



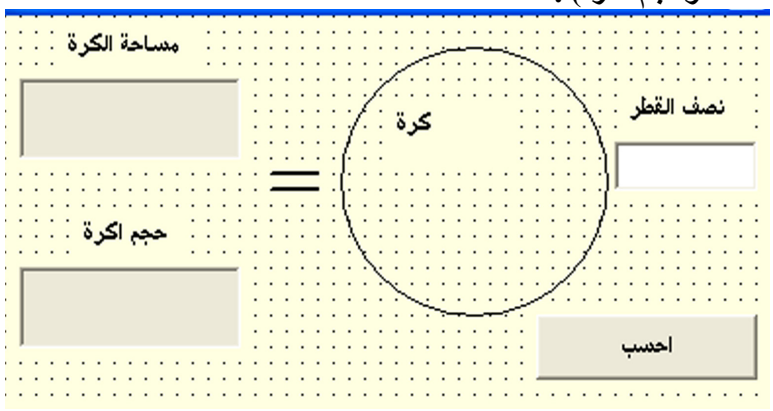

```
Dim x As Double
Dim d As Double
Dim z As Double
x = Val(Text1.Text)
' ول القاعدة الاولى
d = Val(Text2.Text)
' ول القاعدة الثانية
z = Val(Text3.Text)
' الارتفاع
Label4.Caption = (1 / 2 * (x + d) * z)
' الناتج
```

9 – صمم الشكل الاتي لايجاد (مساحة و محيط الدائرة) .



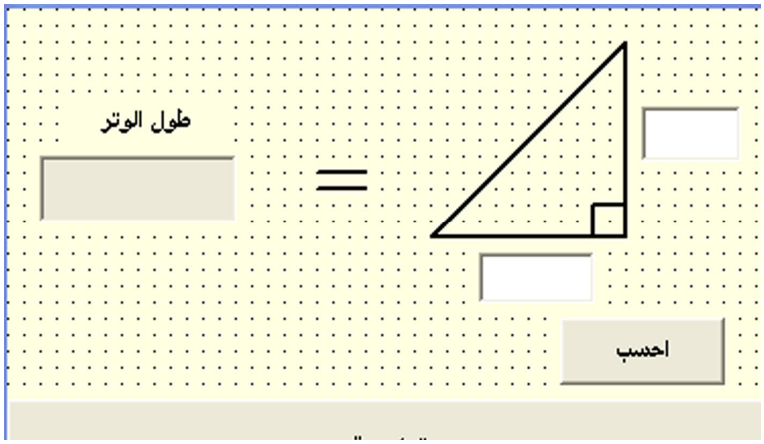

```
Dim x As Double
x = Val(Text1.Text)
' نصف القطر
Label2.Caption = (x ^ 2) * (3.14)
' مساحة الدائرة
Label3.Caption = 2 * x * 3.14
' محيط الدائرة
```

10 - صمم الشكل الاتي لايجاد (مساحة وحجم الكرة) .



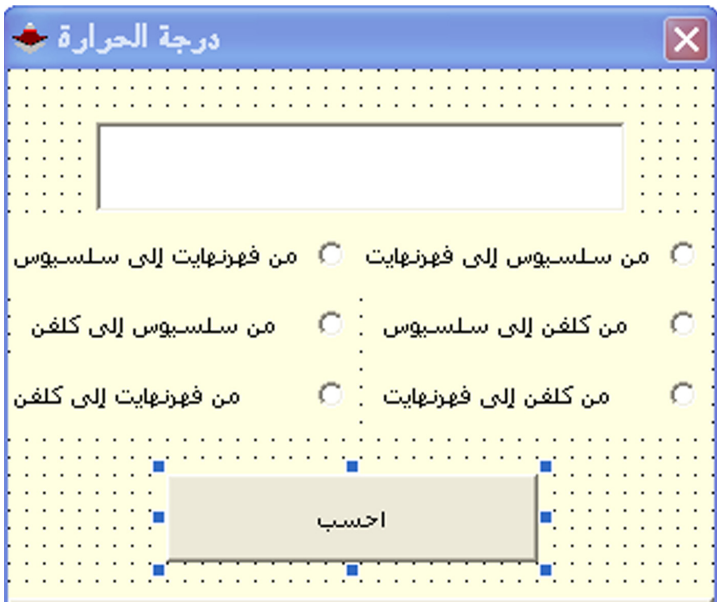

```
Dim x As Double
x = Val(Text1.Text)
' نصف القطر
Label3.Caption = (4 * 3.14 * (x ^ 2))
' مساحة الكرة
Label4.Caption = ((4 / 3) * (3.14) * (x ^ 3))
' حجم الكرة
```

11 – صمم الشكل الاتي لتطبيق (نظرية فيثاغوس) .




```
Dim x As Double
Dim y As Double
Dim z As Double
x = Val(Text1.Text)
' الطول
y = Val(Text2.Text)
' الارتفاع
Label1.Caption = Sqr((x ^ 2) + (y ^ 2))
' الناتج
```

12 – صمم الشكل الاتي (لتحويل درجات الحرارة) .




```
Dim x As Double
Dim z As Double
x = Val(Text1.Text)
' الدرجة
If Option1.Value = True Then
Text1.Text = ((9 * x) / 5) + 32
End If
' من سلسيوس الى فهرنهايت
If Option2.Value = True Then
Text1.Text = ((x - 32) * 5) / 9
End If
' من فهرنهايت الى سلسيوس
If Option3.Value = True Then
Text1.Text = (x + 273)
End If
' من سلسيوس الى كلفن
If Option4.Value = True Then
Text1.Text = (x – 273)
End If
' من كلفن الى سلسيوس
If Option5.Value = True Then
Text1.Text = ((9 * (x - 273)) / 5) + 32
End If
' من كلفن الى فهرنهايت
If Option6.Value = True Then
Text1.Text = (((x - 32) * 5) / 9) + 273
End If
' من فهرنهايت الى كلفن
```

تسمح الدالة IF بتحقيق مجموعة من الشروط يعني اذا تحقق الشرط المعين افعل كذا وكذا ، واذا لم يتحقق افعل كذا وكذا.
مثلا إذا سجل عضو دخوله للموقع وهو ليس موجود في قاعدة البينات سوف يقوم بطباعة خطأ وإذا كان العضو موجود في قاعدة البينات سوف يقبل تسجيل دخول الى الموقع .

هناك ثلاث طرق لكتابة الدالة الشرطية IF

1 - العبارة: IF - Then

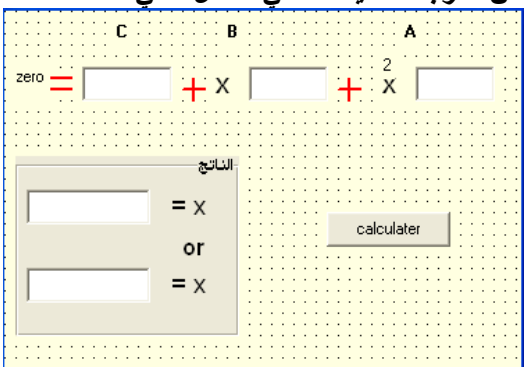
تستخدم عبارة الشرط (If) في شروط أي أنه إذا تحقق الشرط عند ذلك إعمل الحدث وإلا أي إذا لم يتحقق ألغي الحدث أو إعمل حدث أخر.
التعليمة البسيطة لعبارة (If) هي كالتالي:

IF	Condition	Then	Statement
إذا	الشرط	عندئذ	تعليمة

مثال :- اكتب برنامج لادخال درجة طالب ثم طباعة كلمة (pass) اذا كان الطالب ناجح
في هذه التعليمة إذا تحقق الشرط تنفذ كل اتعليمات التي بعد كلمة (Then) وإذا لم ينفذ الشرط تنفذ التعليمات الواردة على السطر التالي لعبارة الشرط.

```
Dim x as integer
Private sub command1_click()
X= cint(text1.text)
If x>= 50 then text2.text= "pass"
End sub
```

مثال:- نقوم الان بتصميم برنامج لحل معادلة من الدرجة الثانية كما في الشكل الاتي ؟



الحل :-
نكتب الكود في الزر (calculater)

```
Private Sub Command1_Click()
Dim a As Double
Dim b As Double
Dim c As Double
Dim r As Double
Dim d As Double
Dim e As Double
a = Val (Text1.Text)
b = Val (Text2.Text)
c = Val (Text3.Text)
r = Val (Text4.Text)
d = Val (Text5.Text)
e = (b ^ 2 - 4 * a * c)
If e < 0 Then MsgBox "المعادلة لا تحل"
If e < 0 Then Exit Sub
```

```
If e >= (b ^ 2 - 4 * a * c) Then Text4.Text = (-b - Sqr(b ^ 2 - 4 * a * c)) / (2 * a)
If e >= (b ^ 2 - 4 * a * c) Then Text5.Text = (-b + Sqr(b ^ 2 - 4 * a * c)) / (2 * a)
```

ملاحظة :- القانون المستخدم هنا هو قانون الدستور .

مثال :-
لتغيير لون الخط باستخدام الاداة timer
الحل :-

```
Private Sub Timer1_Timer()
Static I As Integer
 Label1.ForeColor = QBClor(i)
 I = i+1
 If  I = 16 Then
 I =0
 End If
 End Sub
```

2 - العبارة IF – Then - Else

الأن نقوم بإضافة else تعني بأنه اذا كان الشرط خطأ قم بطباعة شيء اخر .
عندما يتحقق الشرط نفذ تعليمة ما وإذا لم يتحقق فهناك تعليمة أخرى سوف تنفذ في حال عدم تحقق الشرط.
الشكل العام للتعليمة:

IF	Condition	Then	Statement 1	Else	Statement 2
إذا	الشرط	عندئذ	تعليمة	وإلا	تعليمة عدم تحقق الشرط

مثال:-
اكتب برنامج لادخال درجة طالب ثم طباعة كلمة متفوق اذا كان المعدل 100، و كلمة ناجح اذا كان درجته اكبر او يساوي (50) و طباعة كلمة راسب خلاف ذلك .
الحل :-

```
Private Sub Command1_Click()
If Text2 < 50 Then
Text1.Text = "راسب"
Else
Text1.Text = "ناجح"
End If

If Text2 > 100 Then
Text1.Text = "المعدل من 100"
End If

If Text2.Text = 100 Then
Text1.Text = "متفوق"
End If
End Sub
```

3- إذا كانت التعليمة في عدة أسطر فيجب دائماً أن توجد تعليمة End If في نهاية عبارة IF .
مثال :-

```
If Text1.text="eng"
Then
Msgbox'تم تسجيل دخولك'
Unload me
Else
Msgbox"الكلمة التي قمت بكتابتهاغير صحيحة "
End If
```

شرح الكود
اذا كتبت في ال (text) كلمة (eng) سوف يتم تسجيلك بنجاح واذا لم يطابق كلمة (eng) سوف يظهر رسالة يقول (الكلمة التي قمت بكتابتهاغير صحيحة) .

العبارة Select Case

جملة (Select) تشبه تقريباً جملة (If) إلا أنها تستعمل حالات متعددة وفي كل حالة تكتب تعليمة مختلفة.
الشكل العام للعبارة :

```
Select Case (العنصر)
Case (الحالة)
التعليمة
End select
```

ونستخدمها عندما نريد ان ننفذ مجموعة من الجمل والتي تعتمد على قيمة متغير واحد وتنفذ بالشكل

يمكنك إضافة الكثير من الحالات ولاكن قبل (End Select)

مثال :-
أضف فورم وأضف الأدوات التالية:
Text1
Label1
Command1

اضغط (Double Click) فوق الزر وأكتب الشيفرة التالية:

```
Private Sub Command1_Click()
Dim a As Integer
a = Text1.Text
Select Case a
Case 0 To 50
Label1 = "fild"
Case 50 To 80
Label1 = "successful"
Case 80 To 100
Label1 = "excellent"
End Select
Exit Sub
MsgBox "الرجاء إدخال قيمة عددية  ", vbCritical, "خطأ في الكتابة "
End Sub
```

شرح الكود
السطر الأول: التصريح عن متحول عددي صحيح (a)
السطر الثاني: إسناد النص في (Text1) إلى المتحول (a)
السطر الثالث: إلى(End Select) بداية التعليمة ()Select . أي حدد حالات المتحول (a)في كل حالة عرض تعليمة مختلفة
السطور التي بعد (End Select) هي عمل الخطأ أي عرض رسالة عند الخطأ . حاول كتابة نص عادي فسوف تظهر لك الرسالة.
لا حظ أنه أنهينا العبارة ب(End Select) مثل(End If)
ويمكنك إستعمال العبارة الشرطية (IF) ولاكن عليك كتابة الجملة كاملة في كل حالة من الحالات السابقة.

حلقة التكرار Do While

تستعمل جملة التكرار (Do) لعمل حدث ما بعدد معين .
يجب أن تصرح عن متحول قبلها لكي تعمل الجملة :

```
Dim a As Integer
```

الشكل العام

```
Do
A = a+1
التعليمة
Loob Until (a Value)
```

شرح الكود
السطر الأول صرحنا عن متحول عددي
السطر الثاني بداية جملة التكرار
السطر الثالث زيادة قيمة المتحول(a) في كل حلقة تكرار
السطر الرابع تكتب التعليمة التي تريدها
السطر الخامس هو عدد مرات التكرار وهو سوف يكون قيمة المتحول a مثلاً : (6>a) أو (5=a)
لذلك سوف يتم عمل التعليمة التي كتبتها (5) مرات ولهذا السبب قلنا عليك التصريح عن متحول لأنه دون ذلك سوف تكرر التعليمة إلى مالانهاية.

مثال:-
اكتب برنامج لطباعة كلمة (hello) خمسة مرات مع تسلسل كل كلمة مستخدماً (do while)
الحل :-

```
Dim i as integer
Private Sub Command1_Click ()
i = 1
Do while i <= 5
Print "hello"; i
i = i + 1
Loop
End Sub
```

مثال :-
اكتب برنامج لجمع الارقام من (1) الى (10) باستخدام Do While ثم اطبع الناتج؟
الحل :-

```
Private Sub Command1_Click()
Dim x As Integer
Dim y As Integer
x = 0
y = 0
Do While x <= 10
x = x + y
y = y + 1
Loop
Label1.caption = x
Label2.caption = y
End sub
```

وسيكون الناتج في Label1 الرقم 15 ، وفي Label2 الرقم 6

حلقة التكرار For Next

حلقة التكرار(For) مشابهة تقريباً للحلقة السابقة (Do)

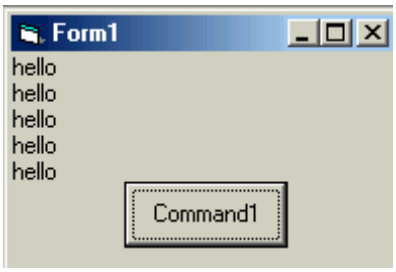
تستعمل لتكرار عمل معين بعدد محدد.
الش0كل العام للتعليمة:

For a= القيمة الأولى To القيمة الثانية
الحدث (التعليمة)
Next

ملاحظة : - (a) مجرد وسيط يمكنك تغيره .

مثال:-
اكتب برنامج لطباعة كلمة (hello) خمسة مرات ؟
الحل :-

```
Dim i as integer
Private Sub Command1_Click ()
For i = 1 To 5
Print "hello"
Next i
End SuB
```



مثال :-
اضف اداة COMMAND و , LABEL واضف الكود الاتي الى الزر ؟
الحل :-

```
Private Sub Command1_Click()
Dim INI As Integer
Dim FIN As Integer
Dim X, Y As Integer
INI = 0
FIN = 10
For Y = 0 To 10 Step 1
X = X + Y
Next Y
Label1.Caption = X
End Sub
```

وسيكون الناتج 55

جمل التكرار المتداخلة (nested for)

يمكن ان تكون الجمل التابعة لجمل التكرار الاولى هي جمل تكرار اخرى، في هذه الحالة يجب انهاء الحلقات الداخلية اولاً ثم الحلقات الخارجية بالتتابع كما يلي.

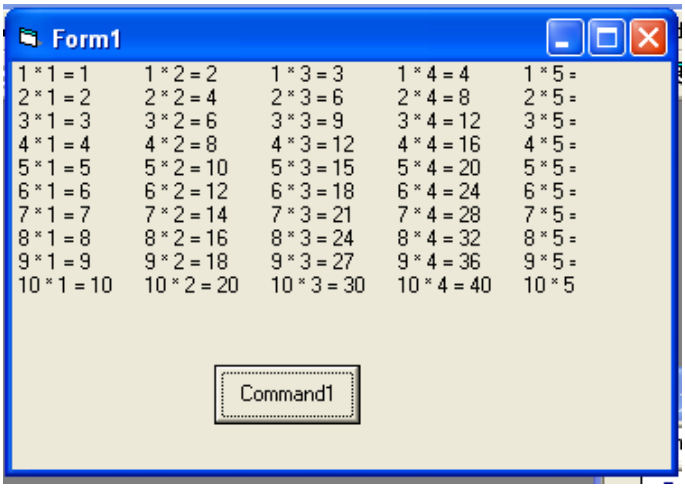
```
For i = 1 To n
For j = 1 To m
Statements
Next j
Next i
```

مثال:-

اكتب برنامج لطباعة جدول الضرب ؟

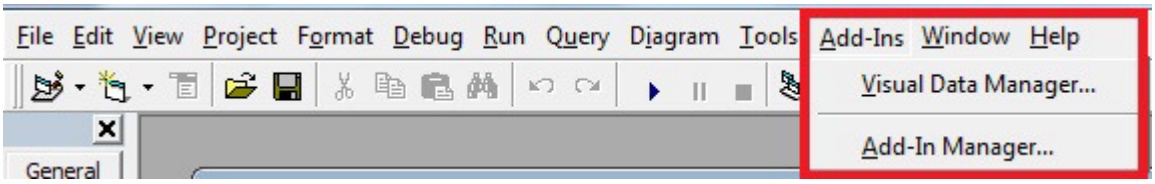
الحل :-

```
Dim I, j As Integer
Private Sub command1_click()
For I = 1 To 10
For j = 1 To 10
p = I * j
Print I; "*"; j; "="; p,
Next j
Print
Next I
End Sub
```

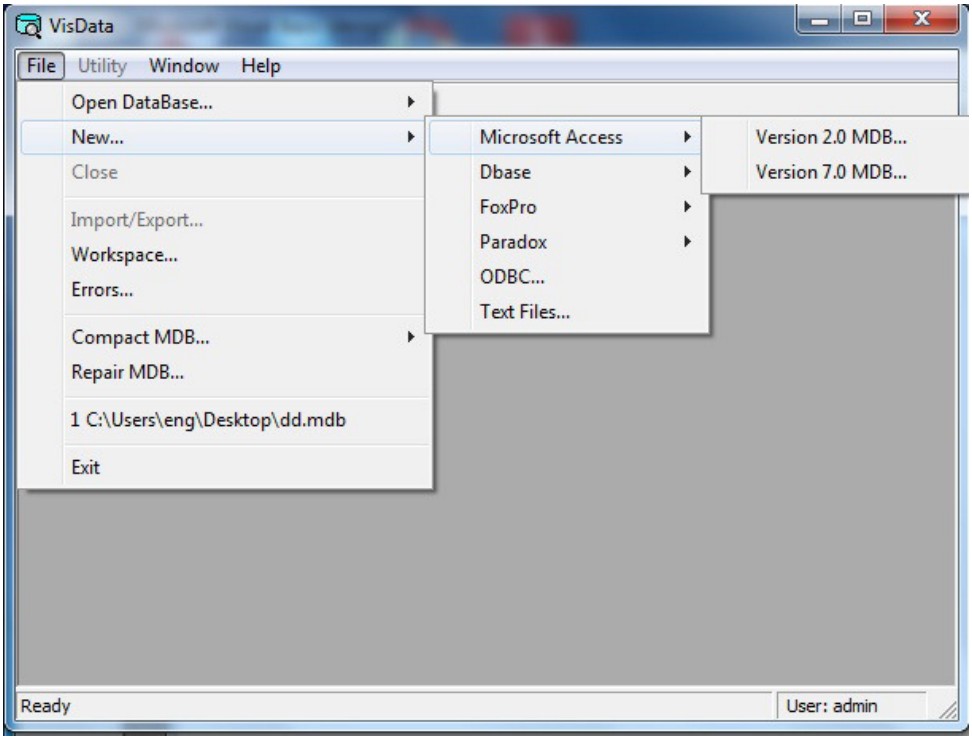


كما هو ممكن انشاء قاعدة البيانات داخل برنامج الـ (Access) نفسه فإنه من الممكن ايضاً انشاء أي نوع من انواع قواعد البيانات داخل بيئة خاصة داخل (VB)تسمى (Visual Data Manager)، و ذلك كما يلي :

1 - الخطوة الأولى فتح الفيجوال بيسك كالمعتاد و من ثم الذهاب إلى قائمة **Add-ins ► visual data manager**

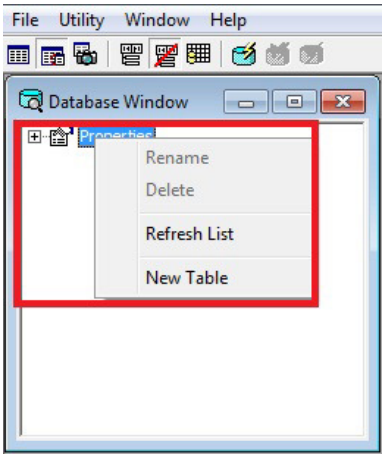


2 - بعد ذلك ستفتح لك نافذة البرنامج اذهب إلى **File ► new ► Microsoft access ► version 7.0 mdb**

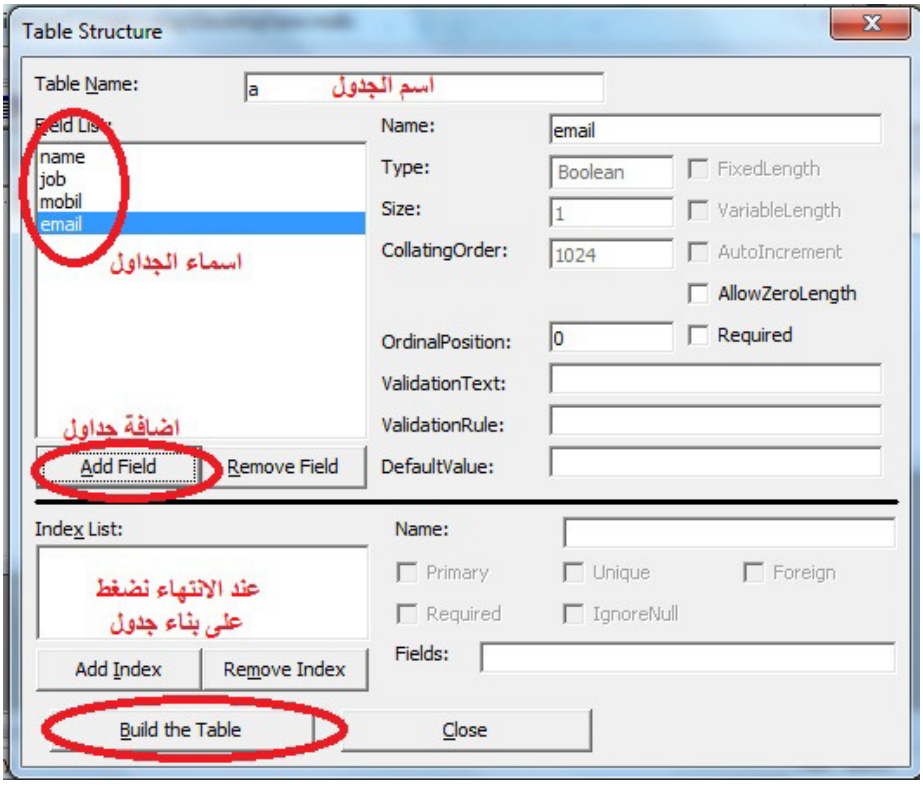


سيظهر لك مربع حفظ لتحدد الموقع الذي تريد أن تحفظ فيه قاعدة بياناتك بعدها اختر حفظ .

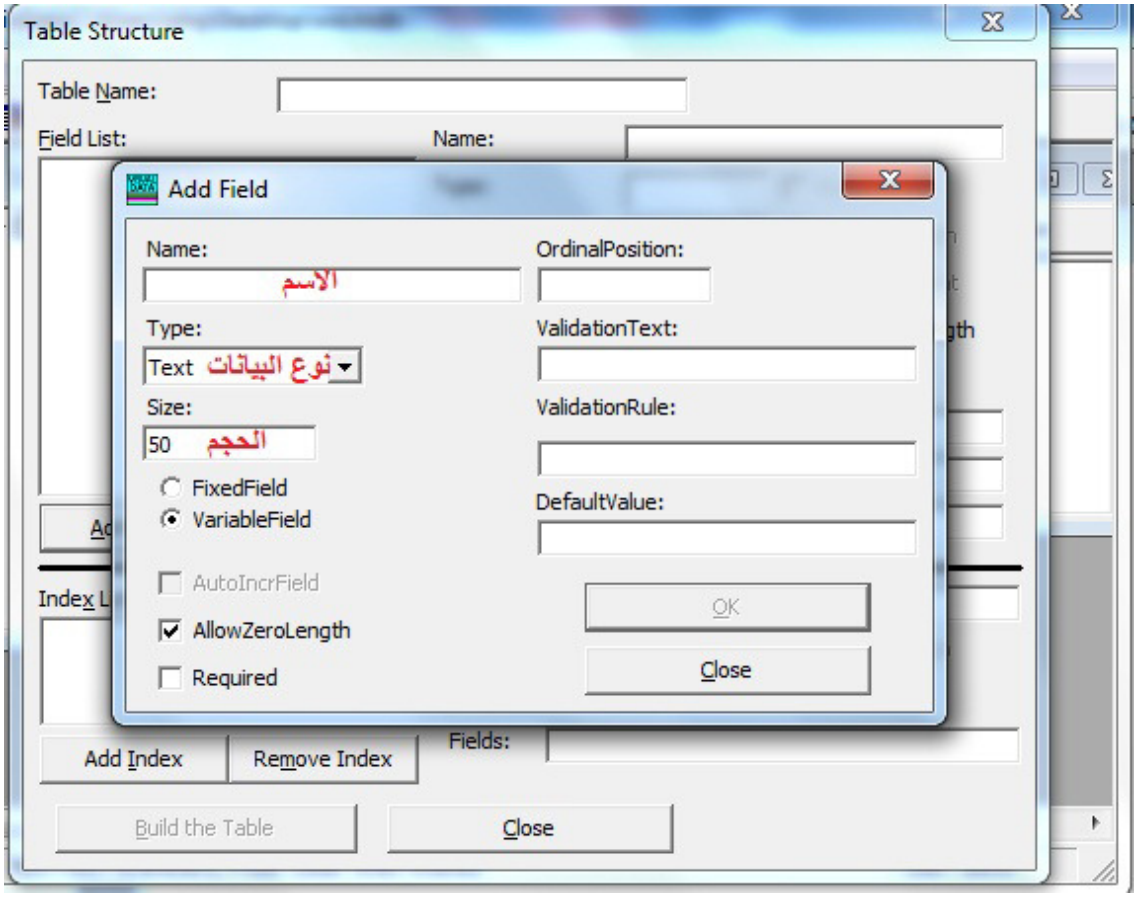
3 - بعد ذلك ستظهر لك قاعدة البيانات ، حدد الخصائص (Properties) بالزر الأيمن ثم أختر (New Table) من اجل انشاء جدول .



4 - بعد ذلك ستفتح لك نافذة تكتب فيها أسم الجدول و ثم أضف حقول الجدول الحقول كما هو مبين .

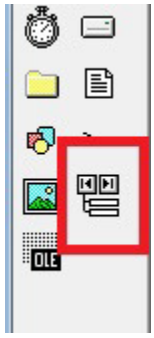


5 - يتفتح لك نافذة إضافة الحقول ، اكتب أسم الحقل في المكان المخصص و حدد نوع بيانات الحقل أي هل هي رقمية أو حرفية ... الخ ، ثم كرر العملية حتى تنهي جميع الحقول التي تريدها .



تقنية *Data Access Objects* (DAO) : توفر هذه التقنية مجموعة من الكائنات (Objects) للتخاطب مع قاعدة البيانات، وإرسال الأوامر لها. وهي مصممة بشكل رئيس للتعامل مع قواعد بيانات أكسس (الى الاصدارة 97) ، و يمكنك استخدامها في التعامل مع قواعد بيانات أخرى من خلال المحرك Jet الذي توفره.

اداة التحكم في البيانات Data Control : و هي تقنية سهلة للتعامل مع قاعدة البيانات و هي احدى ادوات التحكم الموجودة في صندوق الأدوات القياسي ، و هي تمكننا من الاتصال بمجموعة من السجلات (Recordset) في قاعدة بيانات (Jet) للتعامل معها و كذلك ربطها بعدة ادوات في فيجوال بيسك مثل اداة (Textbox).



تقنية Remote Data Objects (RDO) : طوّرت هذه التقنية بشكل رئيس لتوفر طبقة مُبَسِطة لتقنية (ODBC) لمبرمجي لغة فيجوال بيسك.

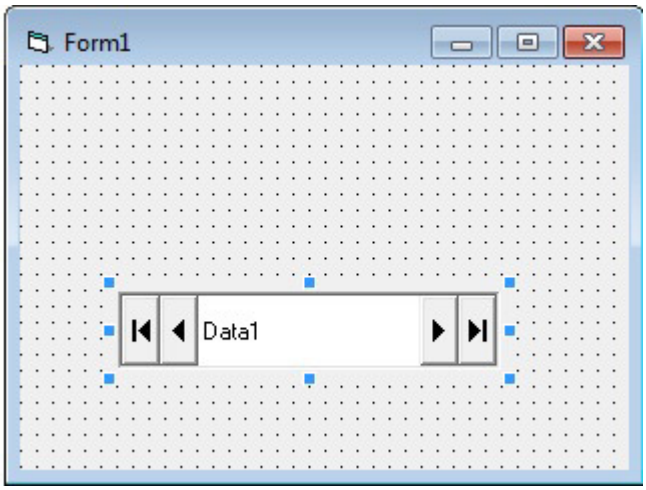
تقنية ActiveX Data Objects (ADO) : واستكمالاً لمسيرة التقدم والتطوير طرحت شركة مايكروسوفت بعدها نظام (ADO)، والتي قامت بتسهيل عملية التعامل مع قواعد البيانات بشكل كبير، وأصبحت الشكل النهائي للوصول لقاعدة البيانات ليس في لغة (VB) و حسب ، بل و في العديد من لغات البرمجة .

سنستخدم التقنية الثانية اداة التحكم في البيانات (Data Control)

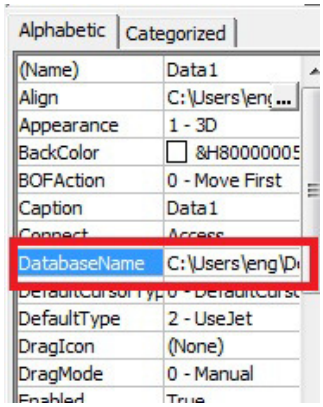
قبل عملية الربط يجب علينا اولاً تصميم قاعدة البيانات ، و يجب ان تكون قاعدة البيانات من نوع (Access 97) على اقصى تقدير .

خطوات الربط ستكون على النحو الآتي :-

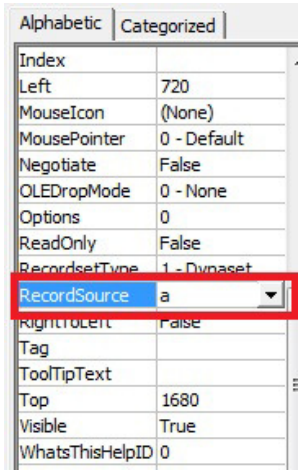
1- يتم انزال الاداة(Data Control) من صندوق الادوات الى الواجهة (Form)، و سيعطى لها الاسم (Data1) .



2- من خلال خصائص الاداة (Data Control) يتم تحديد قاعدة البيانات المراد الاتصال بها عن طريق الخاصية (DatabaseName)

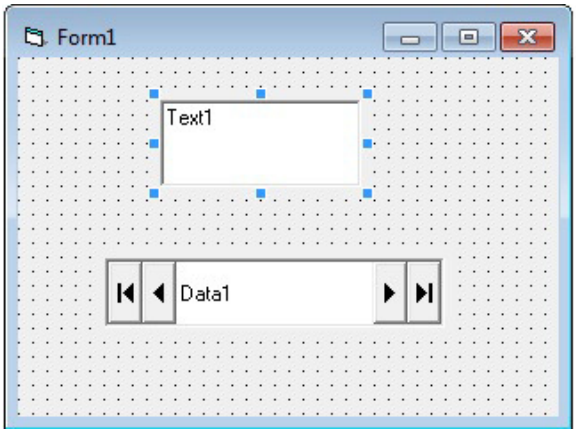


3- بعد ذلك يتم تحديد الجدول المراد الاتصال به عن طريق الخاصية (RecordSourse) و التي معناها السجلات التي اود استرجاعها من قاعدة البيانات .

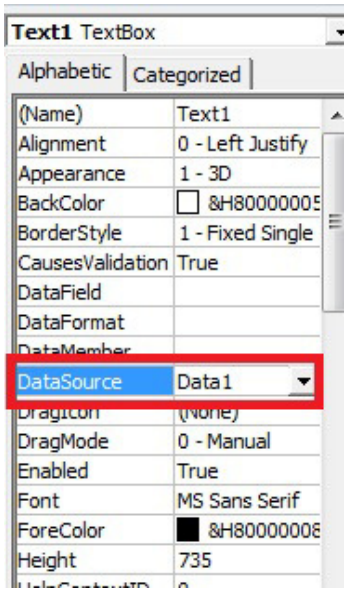


ربط مربعات النص (Textboxes) بحقول قاعدة البيانات

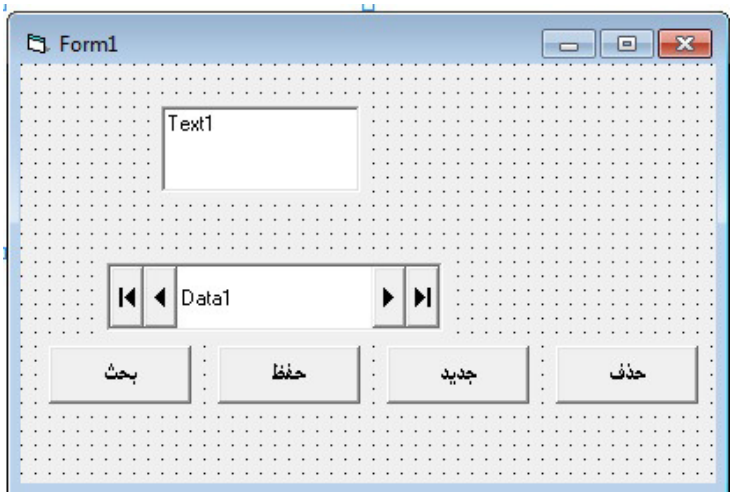
يتم ربط قاعدة البيانات بمربعات النص(Textbox) عن طريق الاداة (Data Control) من اجل عرض حقول البيانات فيها ، و يتم ذلك بإتباع الخطوات التالية :



1 - اختر الخاصية(DataSource)من خصائص مربع النص المراد عرض حقل من البيانات من خلاله و ضع فيه اسم الـ (Data Control) التي تصلنا بقاعدة البيانات (و التي سيكون اسمها في هذا السياق Data1) .



نقوم باضافة الازرار الثلاثة ونكتب الكود الخاص لكل زر (بحث ، حفظ ، جديد) كما في الشكل .



العمليات الرئيسية على سجلات قاعدة البيانات

1- انشاء سجل جديد في قاعدة البيانات

```
Data1.recordset.Addnew
```

شرح الكود:

Data1 :اسم هو ال (Data Control) التي توصلنا بقاعدة البيانات المطلوبة .

RecordSet : يقصد به مجموعة السجلات التابعة للجدول المراد .

إضافة سجل اليه) و التي ترتبط بها الأداة (Data) .

2- حفظ البيانات في قاعدة البيانات.

```
Data1.recordset.update
```

و هي تجرى بعد عملية الاضافة او التعديل(فقط)من اجل حفظ السجل الجديد داخل قاعدة البيانات .

3- حذف سجل من قاعدة البانات

```
Data1.recordset.delete
Data1.refresh
```

حيث ان هذا الامر يقوم بحذف السجل الحالي ، و الخاصية (Refresh) تقوم تحديث بيانات قاعدة البيانات بعد عملية الحذف.

اوامر التنقل بين السجلات

الانتقال الى السجل الاول	Data1. Recordset . MoveFirst
الانتقال الى السجل الأخير	Data1.Recordset.MoveLast
الانتقال الى السجل التالي	Data1.Recordset.MoveNext
الانتقال الى السجل السابق	Data1.Recordset.MovePrevious
المرور على السجلات بحسب قيمة (#no)	Data1.Recordset.Move (#no)

اوامر البحث بين السجلات

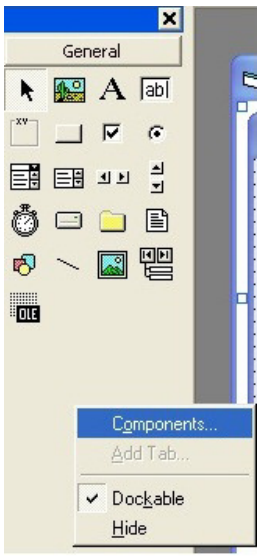
اول سجل يطابق المطلوب من البحث	Data1.Recordset.FindFirst
اخر سجل يطابق المطلوب من البحث	Data1.Recordset.FindLast
السجل التالي الذي يطابق المطلوب من البحث	Data1.Recordset.FindNext
السجل السابق الذي يطابق المطلوب من البحث	Data1.Recordset.FindPrevious

طريقة إضافة وتبديل ال skin

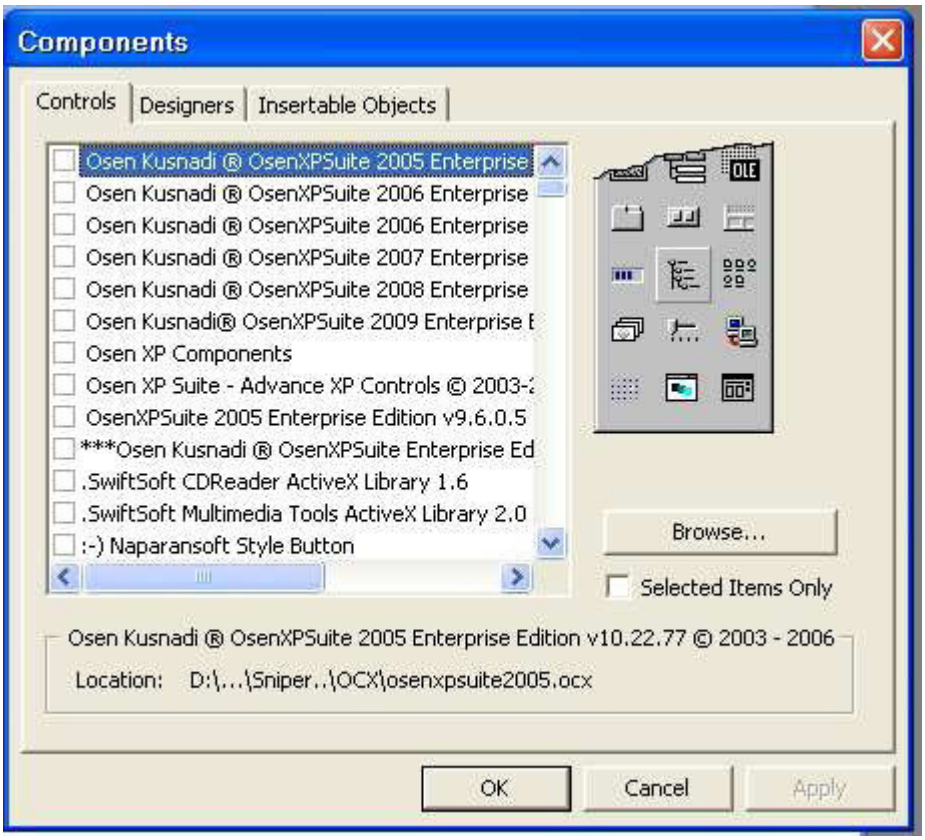
يستخدم اداة ال (skin) لاعطاء مشروعك رونقا و جمالا بدل الشكل القديم للبرنامج

1 - طريقة اضافة ال (skin):-

اضغط (click) بزر الفارة الايمن على ال (tool box) كما في الصورة التالية وقم باختيار (components) .

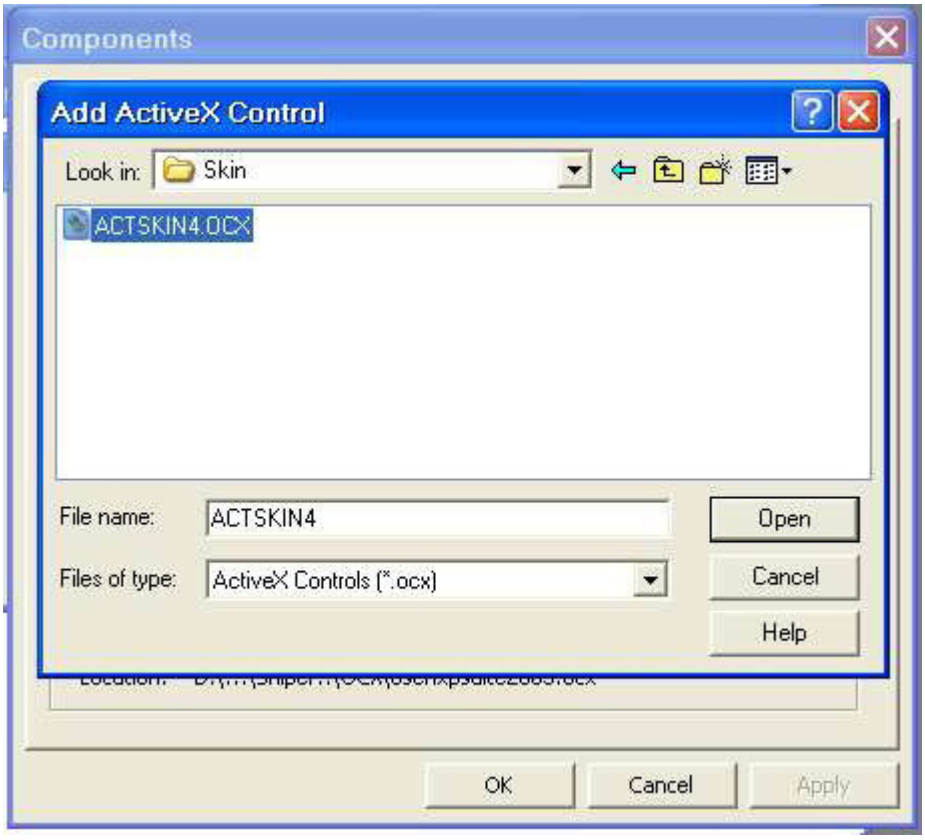


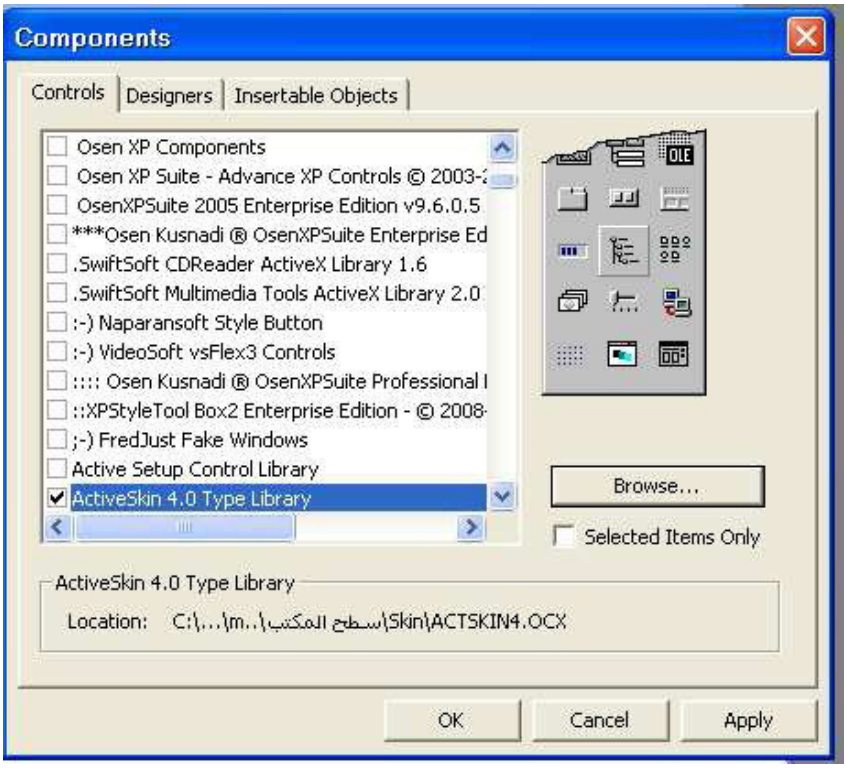
ستظهر شاشة كما في الصورة



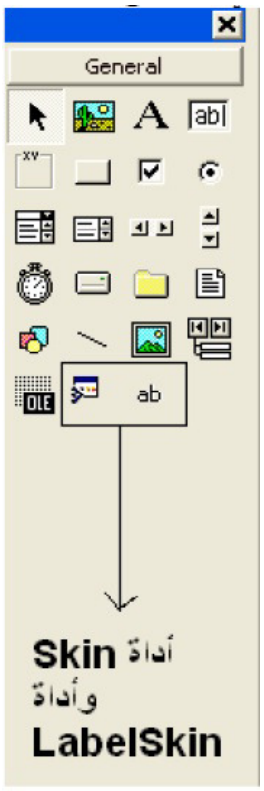
اضغط على (Browse) لاضافة اداة ال (skin) كما في الصورة التالية :-

ملاحظة :- اداة (ACTSKIN4 . OCX) هو اداة يتم تحميله من الانترنيت الى الحاسبة وادراجه الى البرنامج من خلال الخطوات التالية .

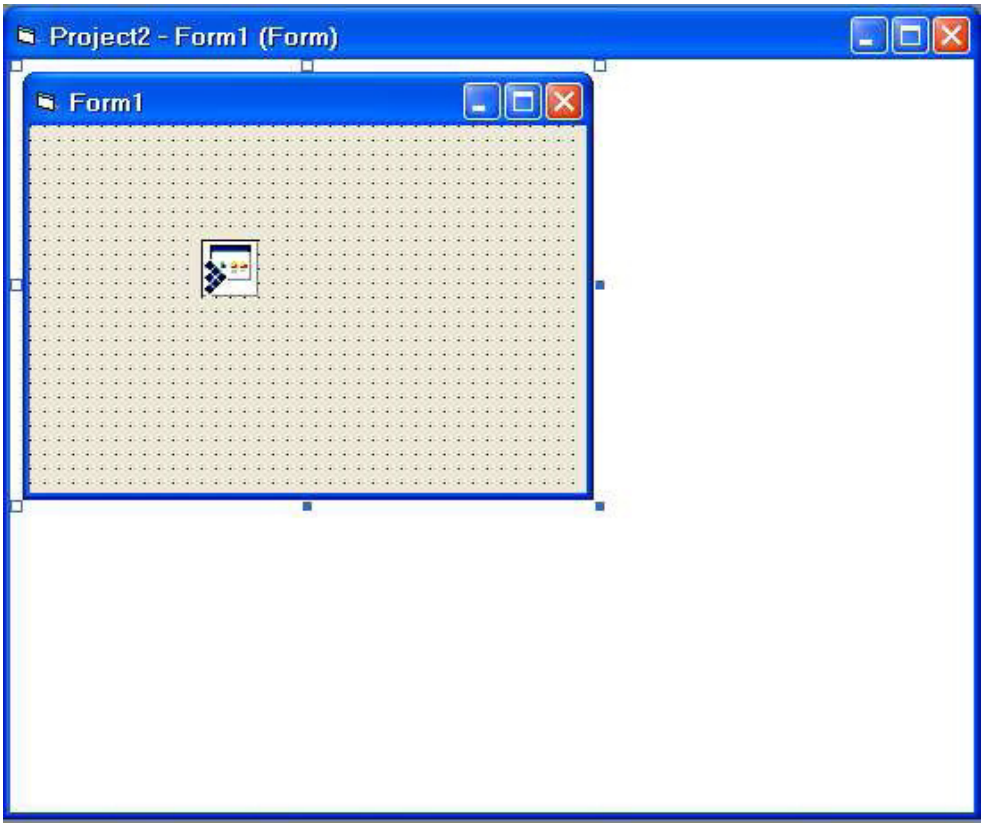




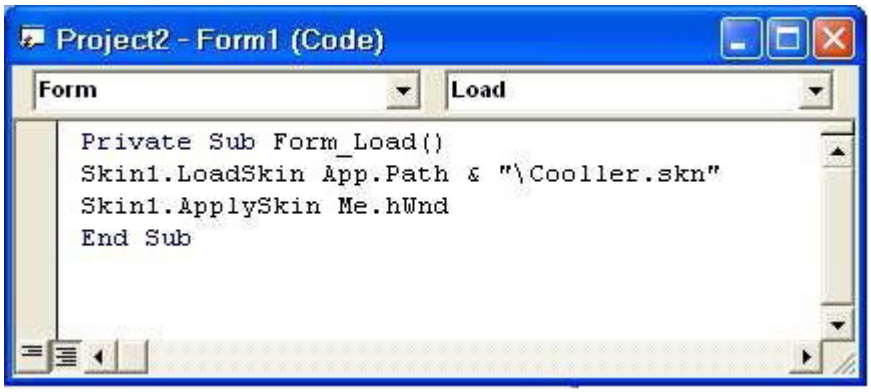
ستظهر اداة ال (SKIN) واداة ال (label skin) في صندوق ال(toolbox) كما في الصورة التالية :



قم باضافة الاداة الى مشروعك كما في الصورة التالية



افتح نافذة كتابة الكود كما في الصورة التالية و اكتب الكود كما موضح



ملاحظة مهمة : - (Cooller.skn)هو اسم ال (skin) الذي تريد اضافته الى المشروع .

شرح الكود

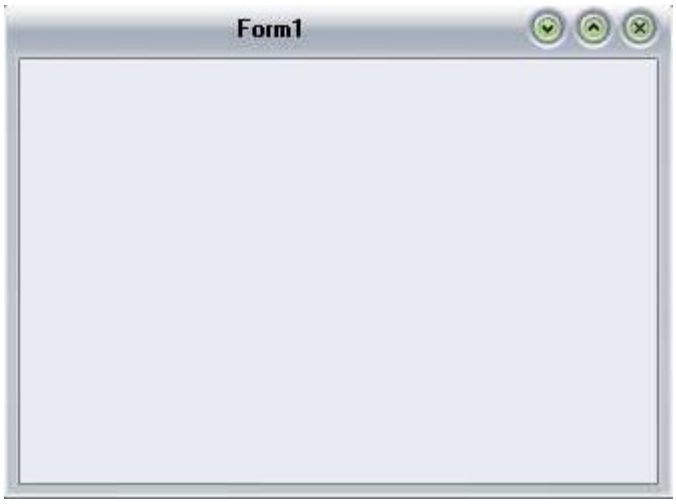
Skin1.LoadSkin App.Path & "\Cooller.skn"

هذا الكود لتحديد مسار ال (skin)

Skin1.ApplySkin Me.hWnd

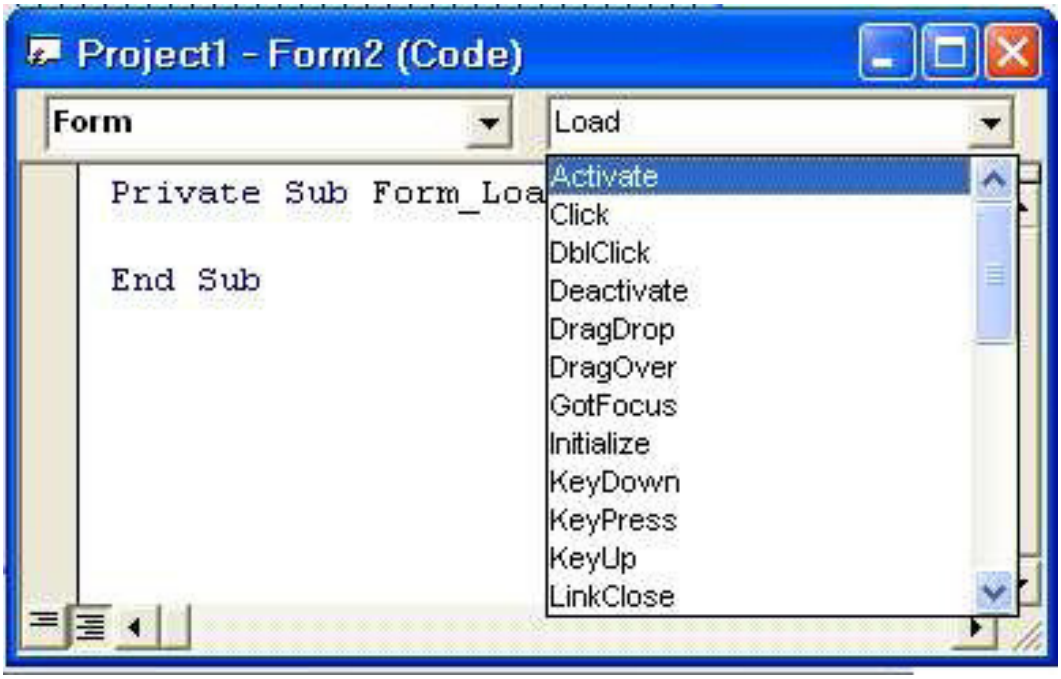
هذا الكود لتشغيل اداة ال skin

قم بتشغيل البرنامج سيظهر لك الفورم كما في الصورة التالية :

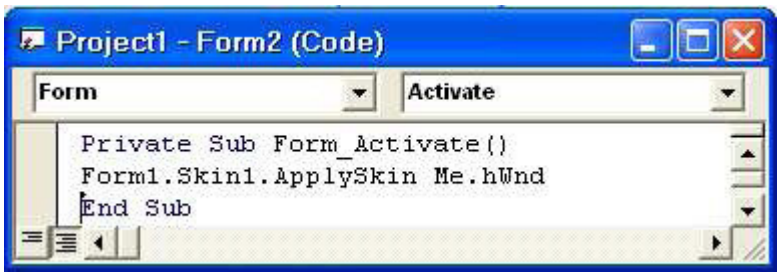


طريقة تطبيق ال (skin) على جميع نوافذ المشروع من خلال اداة ال (skin) الموجودة في الفورة الاول .

نقوم بفتح فورم جديد كما في الصورة التالية ونفتح نافذة الكود للفورم و نختار حدث (Activate) كما في موضح في الشكل

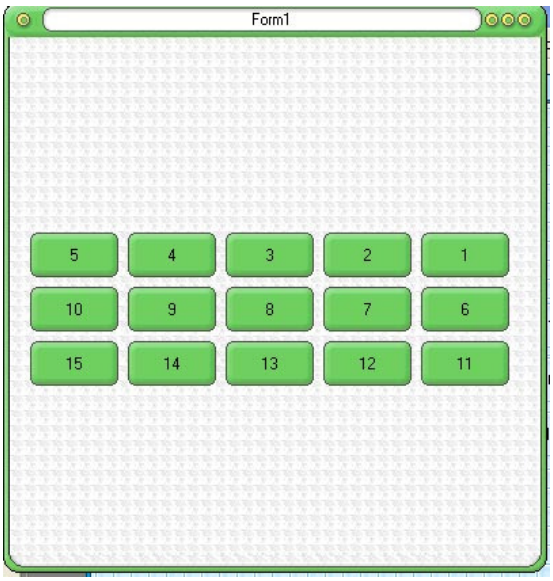


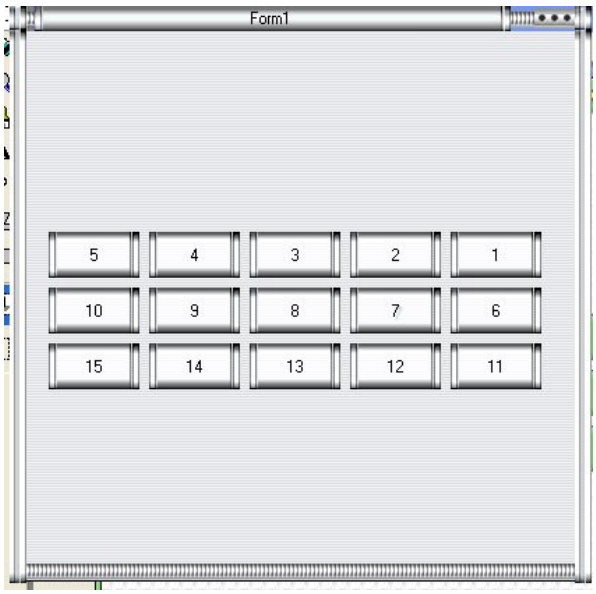
نقوم بكتابة الكود اللازم لتفعيل ال (skin) على الفورم الثاني كما موضح في الصورة



ملاحظة:- اداة ال Actskin4 . ocx وجميع الثيمات التي يتم وضعها في المشروع يجب ان تكون في حافظة واحدة (Folder) لكي يعمل بشكل صحيح .

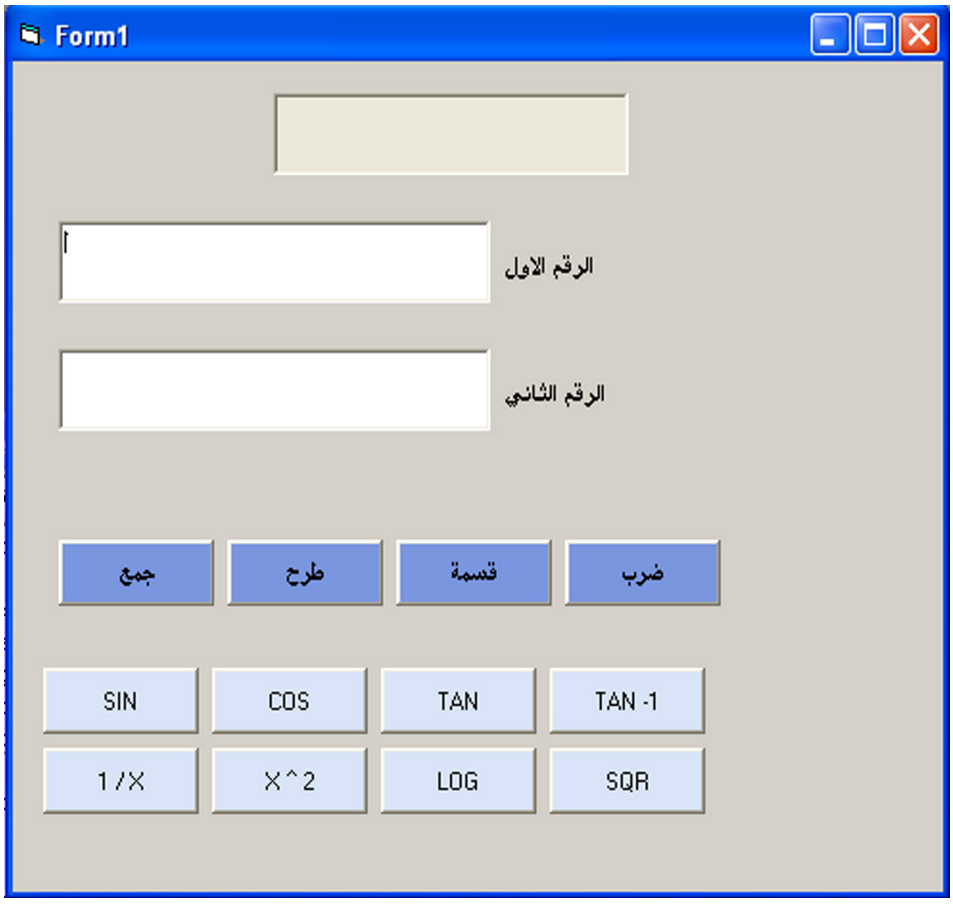
ويوجد اشكال جميلة من الثيمات ليعطي مشروعك منظرا جميلا كما مبين :





عمل آلة حاسبة بسيطة

الشكل العام



يجب تعريف الثابت في قسم التصاريح العامة بالشكل التالي :

```
Const pi = 22 / 7
```

اضف (3)حقول (text)
اضف (3) (label)
اضف (12) زر امر وسمهم كما موضح في الصورة اعلاه :

كتابة الكود :

زر الجمع (+)

```
Label1 . Caption = Val(Text1.Text) + (Text2.Text)
```

زر الطرح (-)

```
Label1. Caption = Val(Text1.Text) - (Text2.Text)
```

زر القسمة (/)

```
Label1. Caption = Val(Text1.Text) / (Text2.Text)
```

زر الضرب (*)

```
Label1. Caption = Val(Text1.Text) * (Text2.Text)
```

زر ال (sin)

```
Label1. Caption = sin (val(Text1.Text) * pi / 180)
```

زر ال (cos)

```
Label1. Caption = cos(val (Text1.Text) * pi / 180)
```

زر ال (tan)

```
 Label1. Caption = tan(val (Text1.Text) * pi / 180)
```

زر ال (tan -1)

```
Label1. Caption = 1/ tan(val (Text1.Text) * pi / 180)
```

زر ال (log)

```
If Val(Text1.Text) >= 0 Then
Label1.Caption = Log(Val(Text1.Text)
Else
MsgBox ""
End If
```

زر ال (1/x)

```
Label1. Caption = 1/val (Text1.Text)
```

زر ال (X ^2)

```
Label1 . Caption = val (Text1.Text) ^ 2
```

زر ال(sqr)

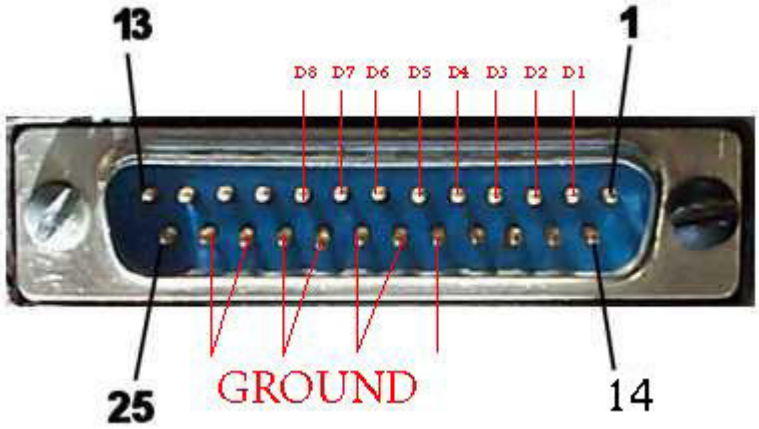
```
If Val(Text1.Text) >= 0 Then
Libel1 . Caption = Sqr(Val(Text1.Text))
Else
MsgBox " "
End If
```

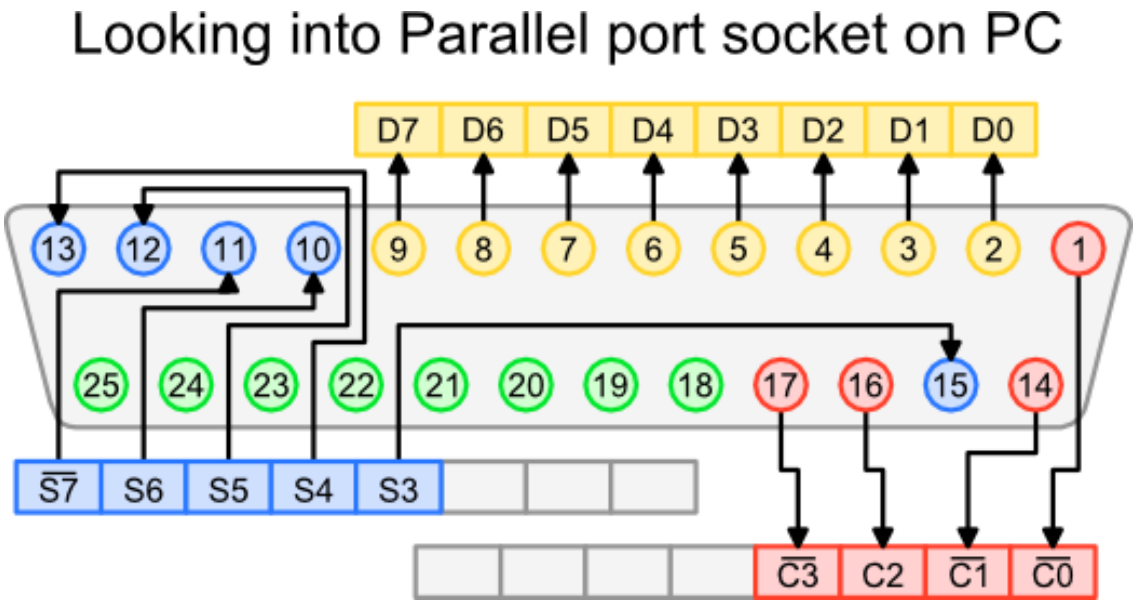
ويوجد انواع عديدة لعمل الحاسبة منها الاعتيادية و العلمية وعدة طرق لكتابة الكود وهذه كانت اسهلها .

الفيجوال بيسك و دوره في التحكم

منفذ التوازي ذو 25 طرف Parallel port

باختصار ان منفذ الطابعة مكون من خمسة وعشرين (Pin أو Hole) حسب شكل المنفذ (بمعنى غاطس أو بارز) تقوم بارسال (0) في حالة (Close) و(1) في حالة - (Open) حيث أن (1) هي قيمة تقترب من (volte 5)في أغلب الأحيان





الرقم 1 + 14،16،17 تسمى بالـ Control Pins
الأرقام من 2 إلإلى 9 تسمى بالـ Data Pins
الأرقام من 10 إلى 13 + 15 تسمى بال. Statues Pins
الأرقام من 18 إلى 25 تسمى بالـ Ground Pins

Control Pins :

عدد 8 من المنافذ ولكن 4 منها غير ظاهرة تستخدم للادخال والاخراج في نفس الوقت.

Data Pins :

هذه هي ال Pins هي الأشهر والأكثر استخداماً ، وتستخدم دائماً لعرض البيانات بعد توصيلها بجهاز مناسب ، وعددها 8 ، وفي الطابعات كانت تستخدم هذه المنافذ لنقل البيانات إلى الطباعة تمهيداً لعملية طباعتها.

Statues Pins :

تستخدم هذه المنافذ لادخال البيانات ، وعددها الفعلي هو 8 وليس خمسة ولكن هناك ثلاث منها غير ظاهرة ، وفي الطابعات كانت تستخدم لادخال بيانات من الطابعة إلى الكومبيوتر مثل رسائل اشعار انتهاء ورق الطابعة.

Ground Pins:

كأي دائرة كهربائية ، لا بد من وجود قطب موجب وقطب سالب لتعمل الدائرة ، ال Ground هو ما يشكل القطب السالب ، وفي العادة لا يتم استخدام سوى واحد فقط من ال pins 8 لجميع ال Pins الخاصة بالادخال والإخراج.

ارسال البيانات عبر منفذ الطابعة

لارسال البيانات عبر منفذ الطابعة لا بد من تحديد رقم المنفذ اولاً وعادة ما يكون H387 ومن ثم ارسال قيمة ما بين 0 إلى 255 لتحديد ال Pinsالتي ستكون عليها القيمة 1 بالشكل التالي مثلاً:

لجعل جميع Pins مفتوحة = 11111111 بالعد الثنائي binary وهو ما يساوي 255

لجعل Pins الأول والثاني والرابع والسادس مفتوحة = 00101011 بالعد الثنائي binarty وهو ما يساوي 1+2+16+64 = 83 .

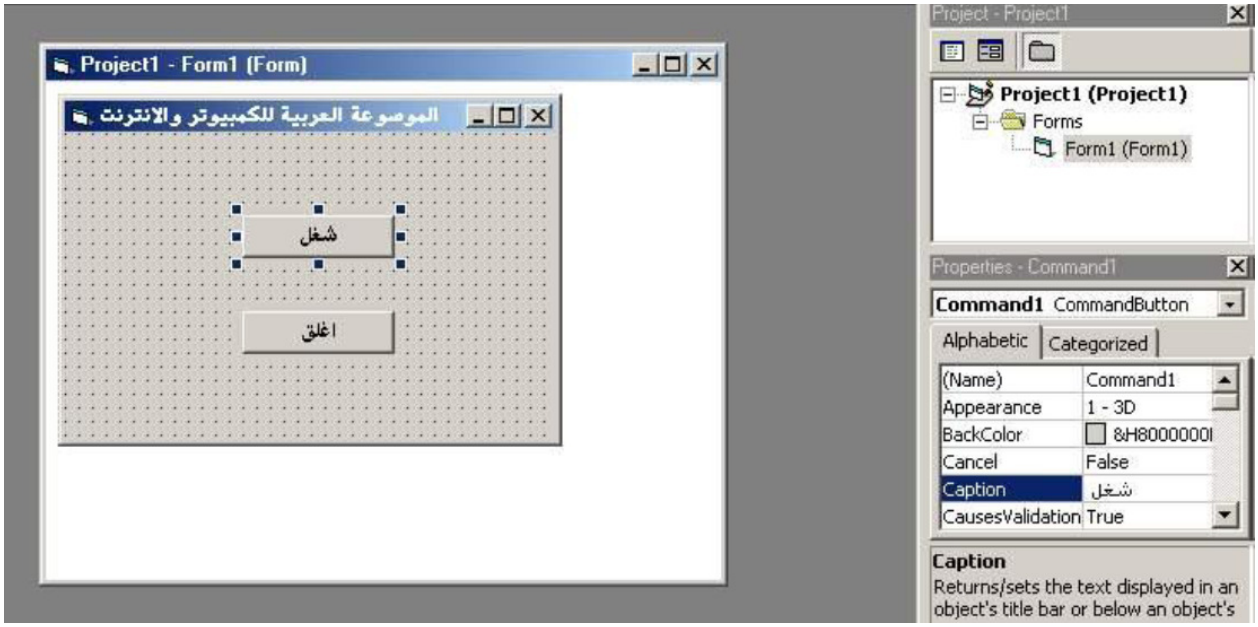
تجربة

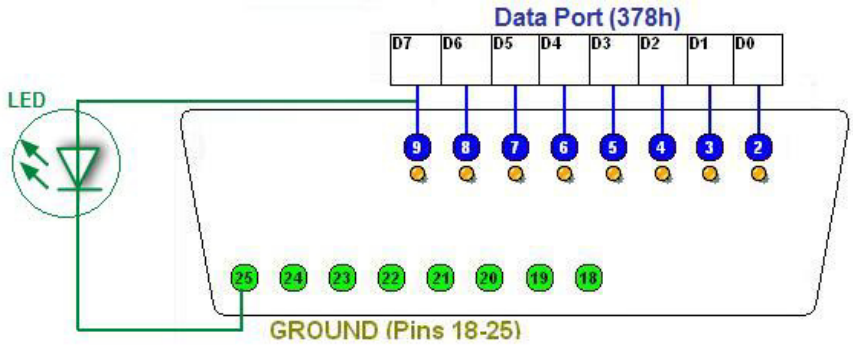
قم بشراء واحد من ال LED's من أي محل للالكترونيات ، وباختصار فال Leds هي عبارة عن انوار كهربائية بسيطة تنير عندما يمر فيها فرق جهد (5) فولت .

خطوتنا الثانية هي معرفة أن لل LED توصيلة للسالب وأخرى للموجب ، قم بتوصيل السالب بواحد من ال Grounds في منفذ الطابعة لديك وقم بتوصيل الموجب في أحد ال Data Pins وليكن الأول لديك ، ومن ثم قم بعمل برنامج سريع يقوم بفتح واغلاق هذا ال Pin .

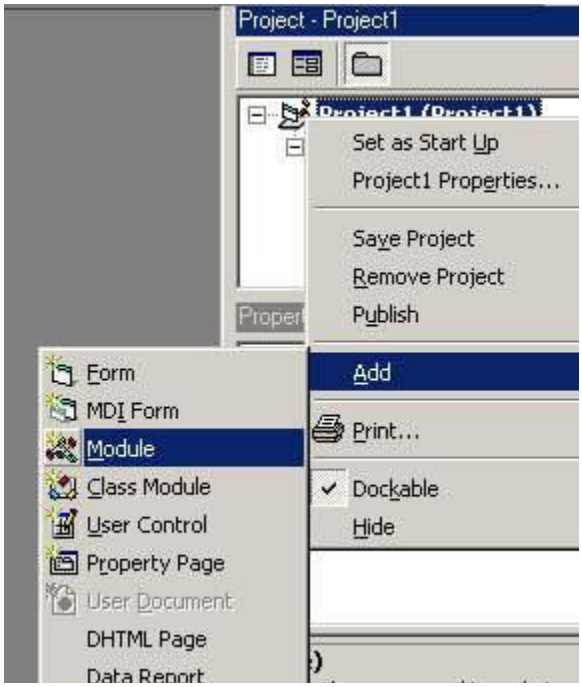
قم بفتح مشروع جديد

اضف الى الفورم الذي تراه امامك زرين امر (command buttons) وغير ال caption كما هو موضح في الصورة:

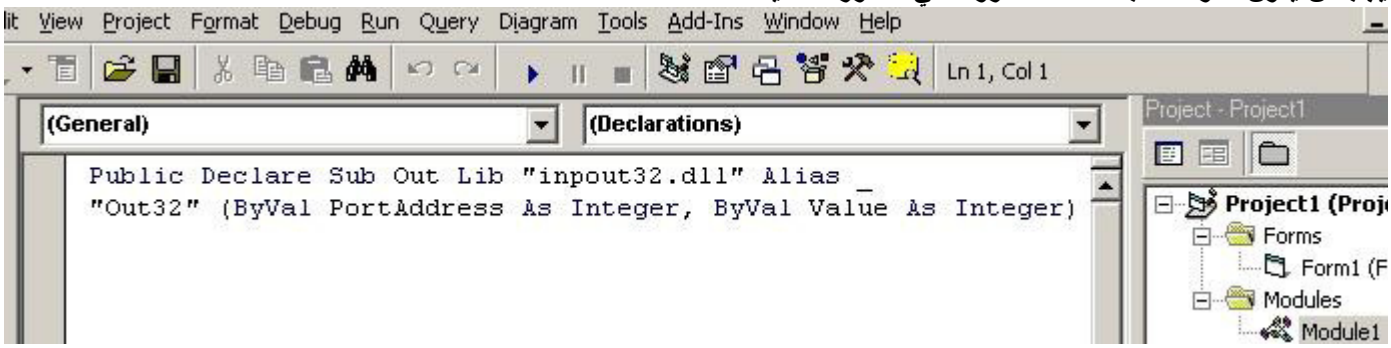




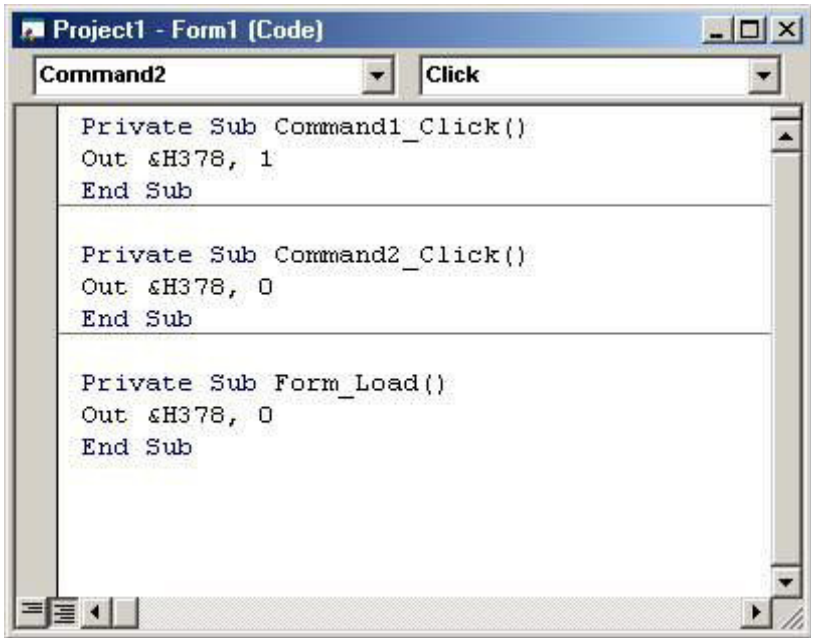
الان يجب اضافة (module) و هو بالضغط بزر الماوس الايمن على كلمة (project) في القائمة الموجودة على اليمين كما هو موضح في الصورة:



الان نكتب التالي في داخل ال (module)
يجب ان يكون الكود مشابه تماماً لما ترونه في الصورة التالية :-



يفضل نسخ الجملة في الاعلى حتى لا تظهر مشاكل تعود لعدم كتابته بالطريقة الصحيحة .
عموما الكود الخاص بالتحكم سيكون كالتالي كما هو موضح في الصورة:



```
Public Declare Function Inp Lib "inpout32.dll"
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll"
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

ومن ثم في زر الأمر(فتح):

```
Out "&H378", 1
```

ومن ثم في زر الأمر (اغلاق) :

```
Out "&H378", 0
```

حيث يتم فتح واغلاق كامل ال (Pins) في كل خطوة.

ملاحظة:- البورت او الحاسبة باكملة يتعامل مع الارقام الثنائية فقط اي (1,0) وان H378 هو اسم المنفذ في الحاسبة .

كيف نخرج تيار كهربائي من بن اخر اليك الحل
اذا اردنا تشغيل البن الثاني نكتب رقم (1)
اذا اردنا تشغيل البن الثالث نكتب رقم (2)
اذا اردنا تشغيل البن الرابع نكتب رقم (4)
اذا اردنا تشغيل البن الخامس نكتب رقم (8)
اذا اردنا تشغيل البن السادس نكتب رقم (16)
اذا اردنا تشغيل البن السابع نكتب رقم (32)
اذا اردنا تشغيل البن الثامن نكتب رقم (64)
اذا اردنا تشغيل البن التاسع نكتب رقم (128)
اذا اردنا تشغيل البن الاول + الثاني نكتب رقم (3)
اذا اردنا تشغيل البن الاول + الثالث نكتب رقم (5)
اذا اردنا تشغيل البن الاول + الثاني + السادس نكتب رقم (33)
النتيجة انه اذا اردنا تحديد بن واحد ناخذ رقم البن ونكتب بدلاً
منه رقم (1) والبنات التي قبله بارقام الاصفار
مثال:-
(1 1 1 1 1 1 1 1) ارقام البنات من (1) الى (8)
(0 0 0 0 1)
النتيجة تشغيل البن رقم (5) ونكتب رقمه في الكود (16)
طبعاً لا ننسى اننا حولنا الرقم الذي حصلنا عليه وهو (10000) من العد الثنائي الى العد العشري
اذا اردنا تحديد اكثر من بن واحد هذا مثال عليه
مثال :-
(1 1 1 1 1 1 1 1) ارقام البنات من (1) الى (8)
(0 0 1 0 1)
هذا المثال لتخريج تيار كهربائي من البن الثالث والخامس ونكتب قيمته بالكود (20)

BIT	D0	D1	D2	D3	D4	D5	D6	D7
VALUE	1	2	4	8	16	32	64	128

ماعلاقة الارقام الثنائية بالفولت و المنفذ

الصورة التالية مثلا، تبين ان الابر(D1, D3 and D4) في وضع عمل ، او بمعنى اخر قيمتهما (5)فولت و البقية قيمتهم صفر فولت.

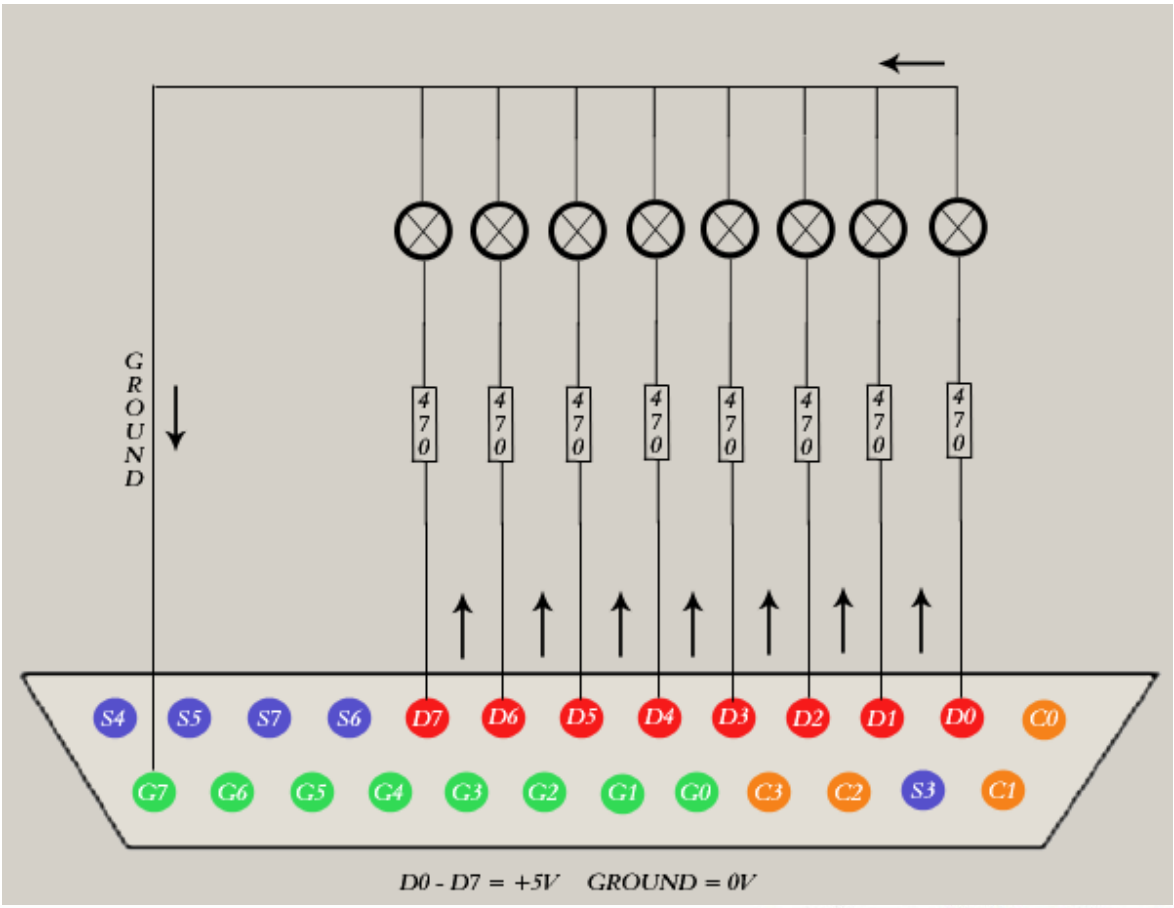
الان، يمكننا تحويل او تفسير الصورة الى الرقم التالي (00001101) الاصفار الاربعة نهملها لان لا قيمة لها ، فيصبح الرقم (1101) و بتحويله الي الرقم العشري فينتج لدينا الرقم (13) .

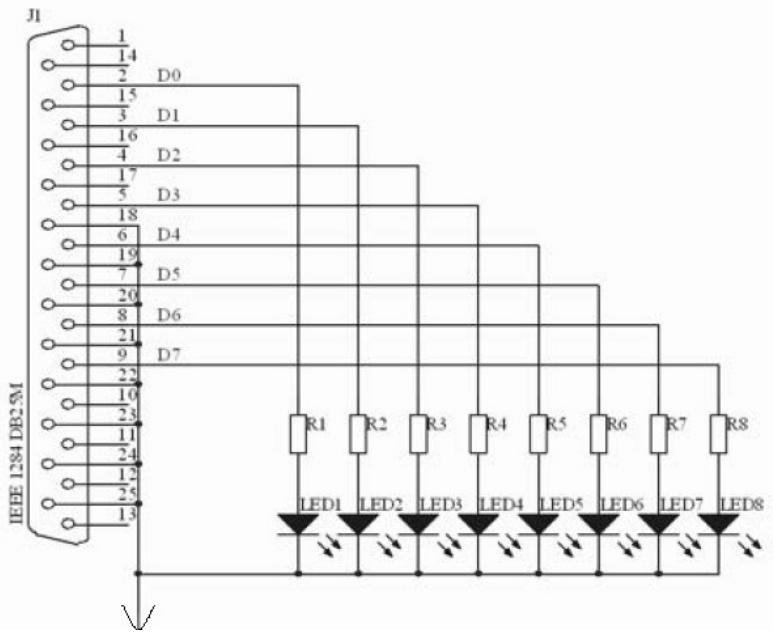
تحذير :

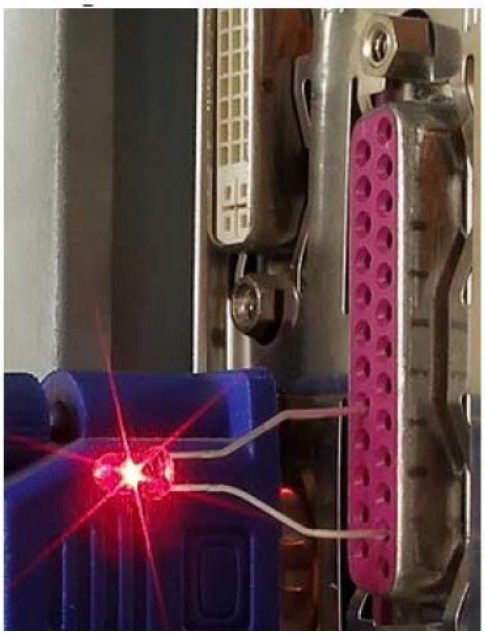
الرجاء التعامل بحذر مع القطع الكهربائية و التأكد الف مرة من انها ليست موصلة بالكهرباء حين العمل عليها و التأكد الف مرة ايضا بانك تعلم ماذا تفعل.

منفذ الطابعة قد يتلف او قد يؤدي الى اضرار عند سوء الاستخدام ، يرجى توصيل المنفذ بكيبل و تكون التجارب على الكيبل لا على المنفذ الملتصق باللوحة الام، مع اخذ الحيطة والحذر و شروط السلامة في كل خطوة تعملها .الكهرباء قد تكون صديقة لنا في كثير من الاحيان ،، لكنها قاتلة في كثير من الاحيان ايضا.

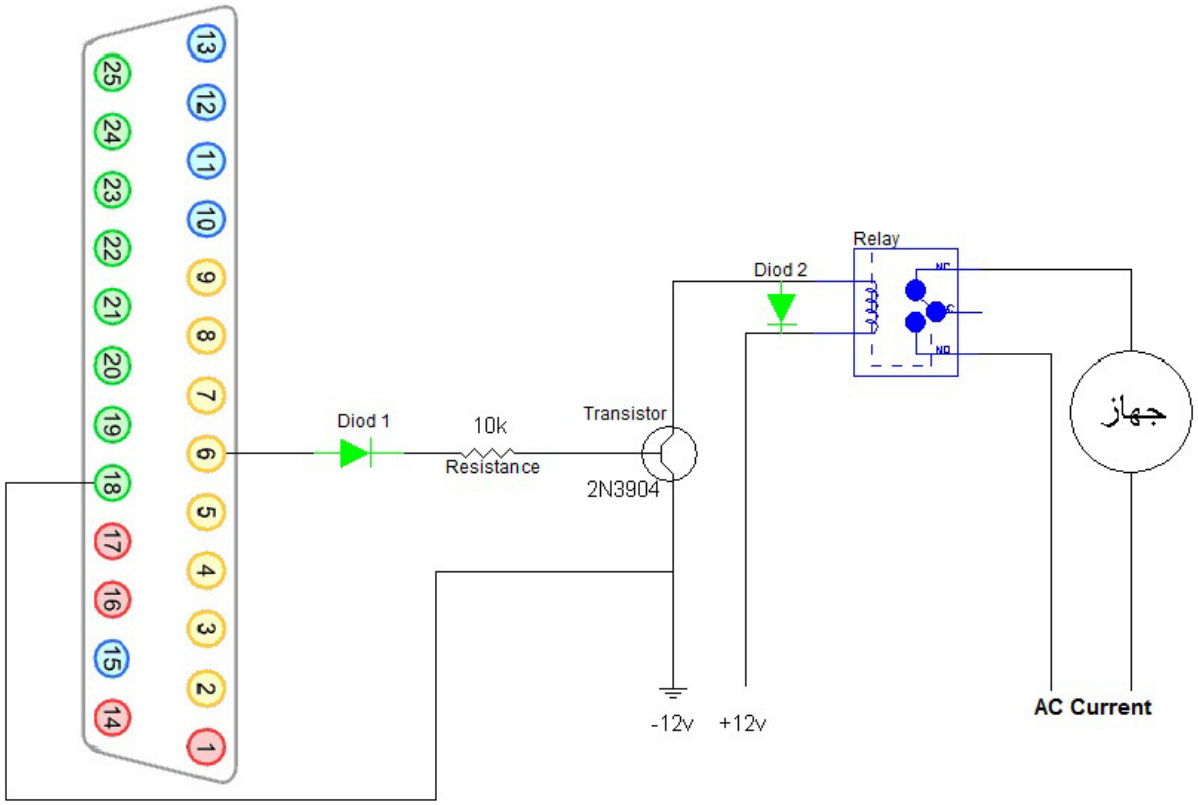
تنبيه : في حالة كونك تعمل على Windows XP فسوف تحتاج لاستخدام هذه المكتبة في الغالب بدلاً من الافتراضية inpout32.dll وهي موجودة في الانترنيت و ننسخها الى C:\Windows\System32.



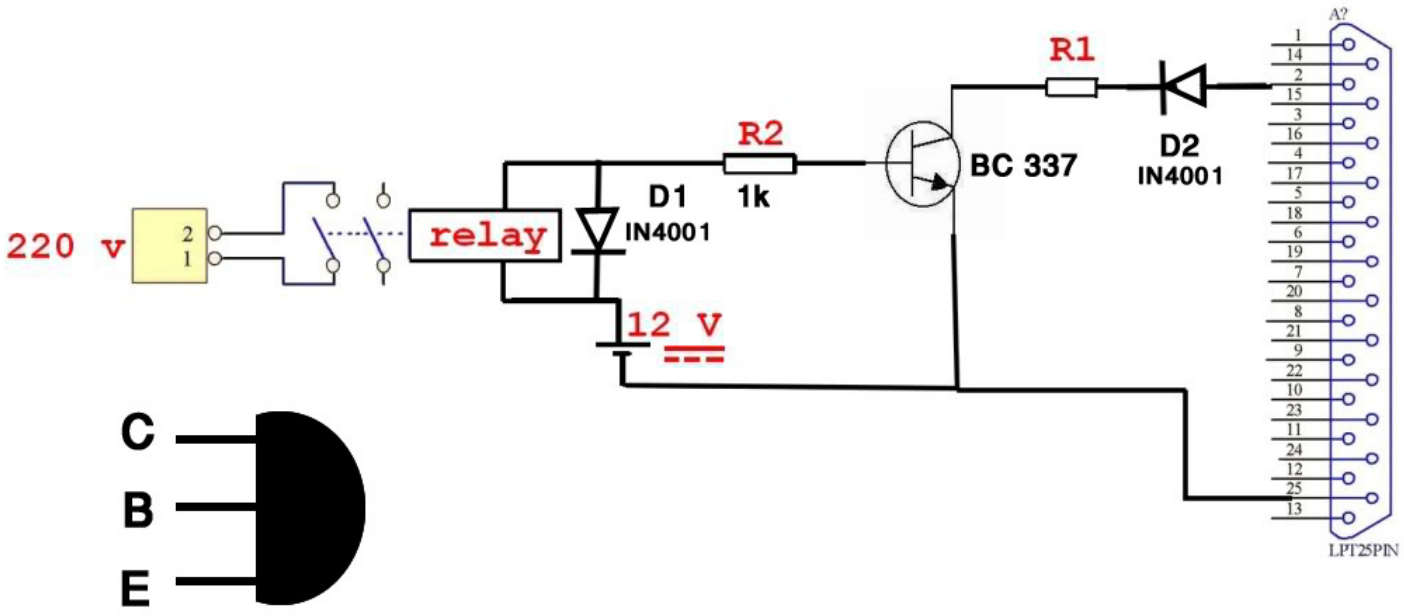




هذه طريقة اعتيادية للتحكم في مصباح صغير و الطريقة الاخرى هي بواسطة الريليه والتحكم بواسطته على اجهزة كهربائية مختلفة بفولتيةعالية ولاكن يجب التاكد جيدا قبل تشغيل الدارة تاكيدا جيدا لنتفادى الاخطاء وعطب الوحة الام (mather bord) .



واذا لم يتم العثور على الترانسيستور فهذا دارة ثانية



(Diod1) يستخدم لحماية (LPT) من التيارات العكسية .

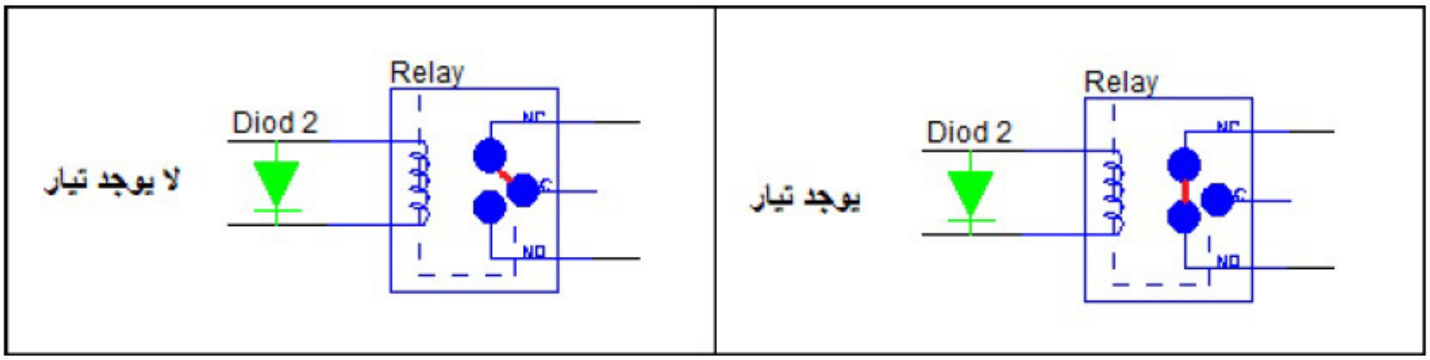
(R) حماية الترانسيستور .

(TR) عند استقبال (5V) من الكومبيوتر يقوم بتمرير المصدر الثاني للكهرباء وهو هنا (12V) وذلك لتشغيل ملف ال (relay) ونقوم باستخدام المصدر الثاني للكهرباء بناءا على قيمة الملف في ال relay فمن الممكن ان يكون (6V, 9V, 12V , 24V) .

(Diod 2) لحماية الترانسستر من التيارات العكسية ، كما انه يحمي الجهاز عند ضعف الامبير في مصدر الطاقة الثانية 12V في حال كان بطارية .

(RELAY) عند استقبال التيار الكهربائي (12V) يقوم بجذب الذراع الداخلي به فيتصل طرفيه ببعض ليسمح بمرور التيار ، مثل الصورة التالية .

(انظر للخط الاحمر وهو التوصيل)



وطريقة اخرى للتحكم هي باستخدام (IC)الدوائر المتكاملة ولاكن لن نتطرق اليها .

المصادر

1-Rod stephens ,visual basic,Programmens refernce wiley publishing

2- ahmad ,abdul mutalib , visual basic 2006 ,ministry of higher education , university of technology , iraq .

3-الزغبي (محمد بلال) الحلبي (يحيى صبري) فيجوال بيسك ،2006 الطبعة الثانية دار وائل للنشر ،عمان،الاردن .

4- الداود (عصام) ،بساطة (عبد الله) visual basic ،2009 ،الطبعة الاولى ،دار وائل للنشر و التوزيع ،عمان، الاردن .

5- النابلسي ، قدري ، البرمجة المرئية فيجوال بيسك ،2009، الطبعة الثانية ،دار وائل للنشر ، عمان ، الاردن .

6- العاني ،سعد عبد العزيز ، تطبيقات في الفيجوال بيسك ، 2005 ،دار الجامد للنشر و التوزيع ، عمان ، الاردن .

7- الفيومي ، محمد احمد ، اسس الحاسبات الالكترونية و معالجة البيانات ، الطبعة الثانية ،دار وائل ، اربد ،جامعة اليرموك ،الاردن ، 1984 .

8- الفيومي ، محمد احمد ، مقدمة في علم الحاسبات و البرمجة بلغة بيسك ، دار الفرقان للنشر ، اربد ،جامعة اليرموك ، الاردن .

مواقع الانترنيت

http://www.vb4arab.com
http://www.vba4a.com
http://www.a1vbcode.com
http://www.star28.com
http://www.khayma.com