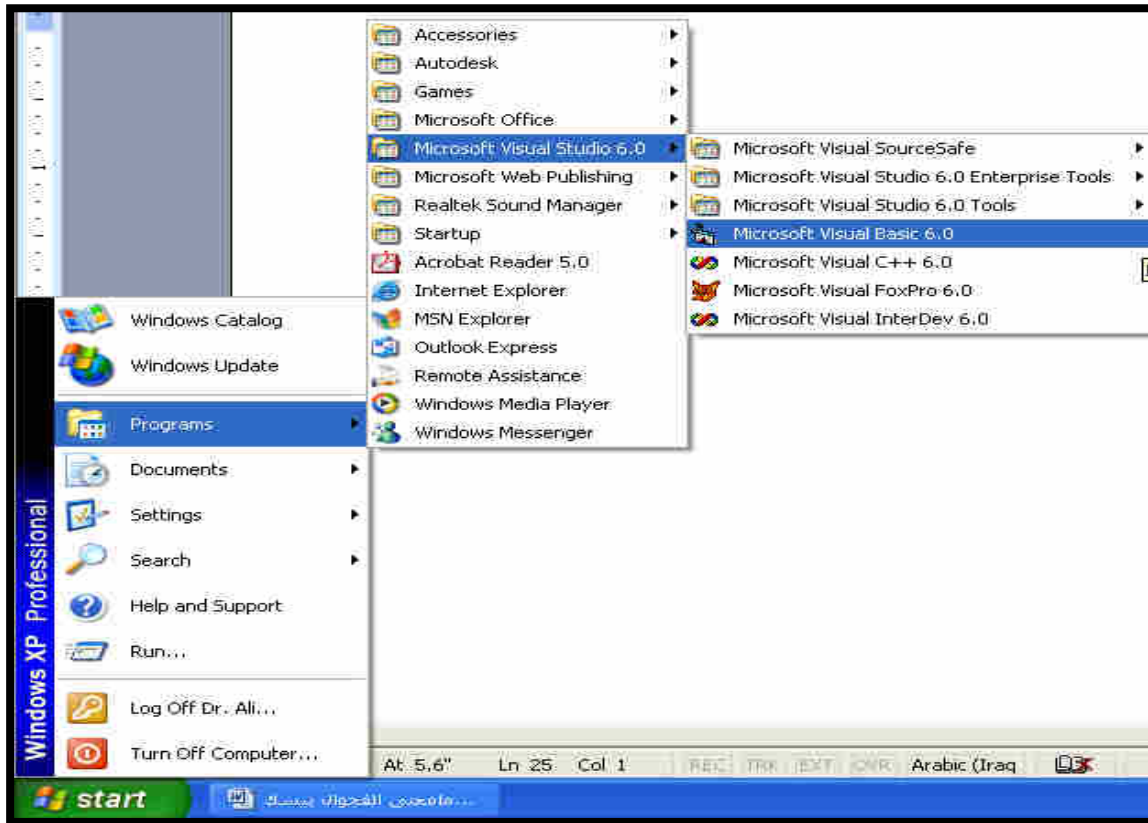


مامعنى الفجوال بيسك ؟

الفجوال بيسك هي لغة برمجة من اهم لغات الحاسبات وكان اسمها أولا بيسك ، ثم طورت لتصبح فجوال بيسك و هي لغة مرئية سهلة التعلم و بسيطة كل البسط و قوية و مشهورة أيضا .
يستخدم VB مفهوم "البرمجة الشيئية" (Object Oriented Programming . OOP) والتي تتكون فيها البرامج من عدة اشياء او "مكونات" Objects كل مكون له مواصفات و "خصائص" Properties تحدد شكله وسلوكه وله "ردود افعال" Actions يتاح له من خلالها التفاعل مع "الاحداث" Events او "المستخدم" User تمكن المبرمج على ان يجعل كل Object يتفاعل بالطريقة التي يريدها ويتم ذلك من خلال "شفرة البرنامج" Code والذي هو عبارة عن مجموعة اوامر تعرف الObject كيف يتصرف.

بدء تشغيل البرنامج

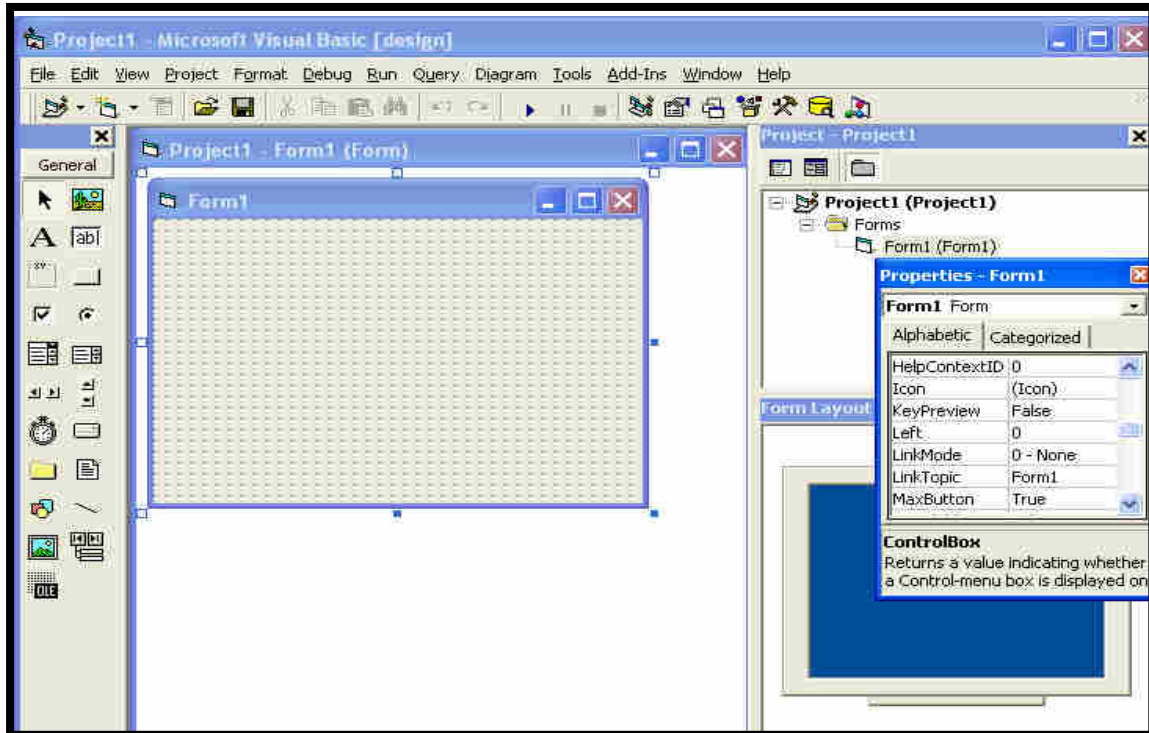
1. انقر المفتاح Start .
2. اذهب الى Programs ومن ثم Microsoft Visual Studio6.0 ثم Microsoft Visual Basic6.0 .
كما في الشكل التالي .



بعد ذلك سيفتح البرنامج وتظهر لنا شاشة المشروع الجديد New Project وبها عدد من الاوضاع التي يختار من بينها المبرمج الصيغة التي يريد لبرنامجها ان يظهر بها ، ومن بعدها نختار الاختيار الاول Standard.exe ومن ثم Open كما في الشكل التالي . هذا التطبيق satandard.exe هو الافتراضي أي لصنع برنامج إفتراضي و قياسي تنفيذي و هذا الإختيار هو الغالب عادة في تطبيقاتنا القادمة.

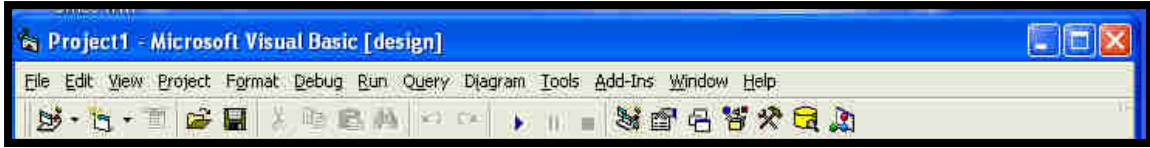


ومن ثم تظهر لنا صفحة البرنامج الرئيسية ، حيث نجد قوائم البرنامج التي لا تختلف كثيرا عن اي برنامج من برامج مايكروسوفت المعروفة وسنتعرف على كل جزء من اجزاءها بالتفصيل .
لاحظ كيف تبدو نافذة VB وكما في الشكل التالي :



كما نرى فان برنامج VB يحتوي عدة نوافذ وهي :

١. النافذة الرئيسية :



تستخدم هذه النافذة لاداء الوظائف الرئيسية في VB مثل حفظ المشروعات او عمل الملفات قابلة للتنفيذ ، وهي تقسم الى ثلاثة اقسام هي :

- شريط العنوان :



سنلاحظ وجود العنوان Microsoft Visual Basic [Design]

والكلمة التي بين القوسين تعكس الطور الذي يعمل به VB حاليا وهي [Design] اي اثناء تصميم البرنامج او [Run] اثناء تشغيله او [Break] اثناء توقيف البرنامج مؤقتا لفحصه .

• شريط القوام :

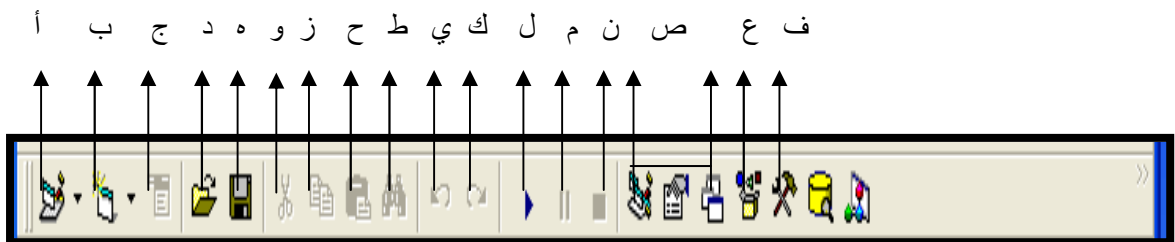


شريط القوائم هو الشريط الابيض الذي يظهر اسفل شريط العنوان في نافذة VB الرئيسية ويحتوي هذا الشريط على القوائم الرئيسية في " VB " وتحتوي كل قائمة على مجموعة من الأوامر المتعلقة بموضوع معين فمثلا: قائمة File تحتوي على الأوامر الخاصة بالمشروعات والملفات مثل حفظ أو تحميل وإضافة ملف أو حذف. وقائمة Edit تحتوي على الأوامر الخاصة بالتحريك مثل القص واللصق والنسخ واوامر اخري. وقائمة View تعمل محتويات هذه القائمة علي اظهار بعض الاشياء مثل صفحة كود و فورم و صندوق الادوات...الخ.

وقائمة Run تحتوي على الاوامر الخاصة بتشغيل البرنامج مثل تنفيذه أو إيقافه بصورة مؤقتة أو انهائه وسوف نشرح كل أمر من هذه الاوامر عند استخدامه .

• شريط الادوات :

يحتوي شريط الادوات على مجموعة الاوامر التي تستخدمها دائما في " VB " هذه الاوامر موجودة في شريط القوائم ولكنها وضعت هنا تسهيلا عليك .

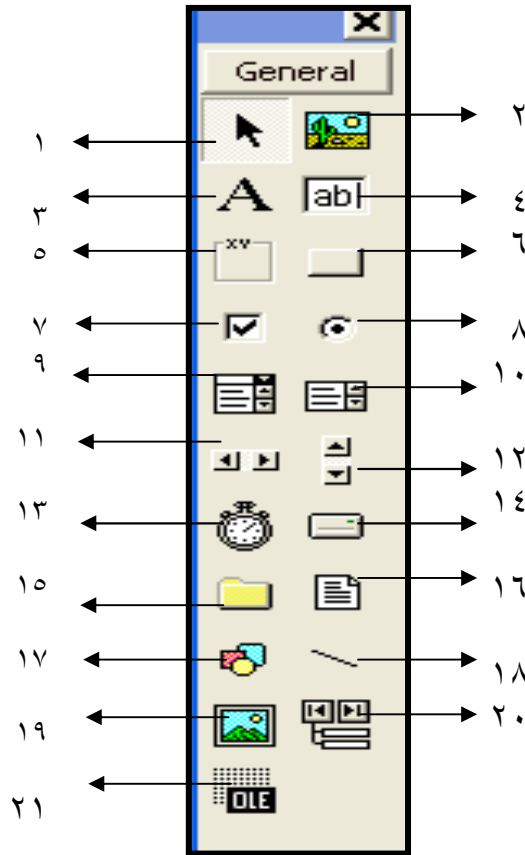


فمثلا بدل من ان تبحث عن امر فتح ملف من قائمة File يمكنك ان تنقر على الرمز مباشرة حيث الايقونة المشار اليها بالحرف "أ" لاضافة مشروع برنامج ، يليه "ب" لاضافة نافذة جديدة ، ثم "ج" لادراج وتحرير القوائم "ملف، تحرير، عرض"، يأتي بعد ذلك أدوات معروفة لاكثرنا ف"د" لفتح مشرع محفوظ ،"هـ" لحفظ المشروع

الحالي، و"للقص، ز" للنسخ، ح" لللصق، ط" للبحث، واخيراً ك،ي" للتراجع للخلف والأمام، يلي هذه الايقونة ثلاثة أخرى خاصة بالتحكم، ف"ل" لتشغيل البرنامج، م" للتوقف المؤقت، ثم ن" لايقاف البرنامج بشكل نهائي اما الايقونات المشار اليها بالحرف "ص" وعددها ثلاثة فهي خاصة بعرض نوافذ خصائص المشروع، والايقونة "ع" لاستعراض المشروع، واخيراً "ف" للتحكم بالادوات.

نافذة الادوات :

يعتبر شريط الأدوات من الكائنات أو النوافذ الرئيسية ويحتوي على كل الأدوات التي يمكن أن تضيفها للفورم وللتعريف بهذه الادوات فقد أشرنا أمام كل منها برقم.



الرقم	الأداة	العمل
1	Pointer	تعيد مؤشر الماوس إلى السهم الطبيعي إذا كان المؤشر على احدى الأدوات أو الأشكال الأخرى غير الوجه الطبيعي
2	PictureBox	تتيح إضافة صورة إلى البرنامج ... وادماج هذه الصورة إلى البرنامج
3	Label	هذه الأداة تتيح لنا إضافة عنوان إلى البرنامج ... أو نص في أي مكان من الفورم
4	Text Box	هذه الأداة تستخدم لإدخال أو إخراج بيانات

وهي أداة تتيح لك صنع Frame إطار وتضمن بعض الأدوات بداخل هذا الإطار	Frame	5
يستخدم لتنفيذ عملية معينة عند الضغط عليه	Command Button	6
وهي أداة نقطة إختيار ... وتتيح للمستخدم استخدام خيارات Options التي تضعها في البرنامج	Check Box	7
وهي أداة نقطة إختيار ... وفيها يمكنك أن تختار اختيارا ما من عدة إختيارات	Option Button	8
أداة تضيف إليها قائمة منسدلة ليختار المستخدم منها إحدى القيم	Compo box	9
وهي أداة قائمة List تشبه القائمة السابقة ولكن مع الفارق أن هذه الأداة ليست منسدلة	Listbox	10
وهي أداة شريط الإنزلاق العرضي ... وتستخدم في إنزلاق الصور والكائنات الأكبر من اللازم عرضيا	HScrolBar	11
أداة تشبه السابقة ولكن شريط انزلاق طولي وليس عرضي	VScrollBar	12
وهي أداة الموقت ووظيفتها أنها تقوم بأداء عمل معين أو عدة اعمال معينة بصفة دورية كلما مر زمن معين تحدده	Timer	13
عبارة عن ListBox فيه أقسام القرص الصلب وقسمي القرص المرن والسي دي	DriveListBox	14
وهي عبارة عن أداة ListBox تقوم بعرض المجلدات Folders في مسار معين تحدده أنت	dirListBox	15
وهي عبارة عن أداة Listbox تقوم بعرض الملفات Files في مسار معين	FileListbox	16
وهي عبارة عن أداة رسم شكل	Shape	17
وهي أداة رسم خط على الفورم	Line	18
وهي أداة إضافة صورة وتختلف بعض الإختلافات عن الأداة PictureBox	Image	19
وهي أداة تستخدم في ربط البرنامج بقاعدة بيانات خارجية	data	20
وهي أداة ربط وتضمن ملفات وبرامج خارجية ضمن برنامجك	OLE	21

هناك طريقتان لوضع الادوات على النافذة :

١- وضع الادوات بالنقر المزدوج .

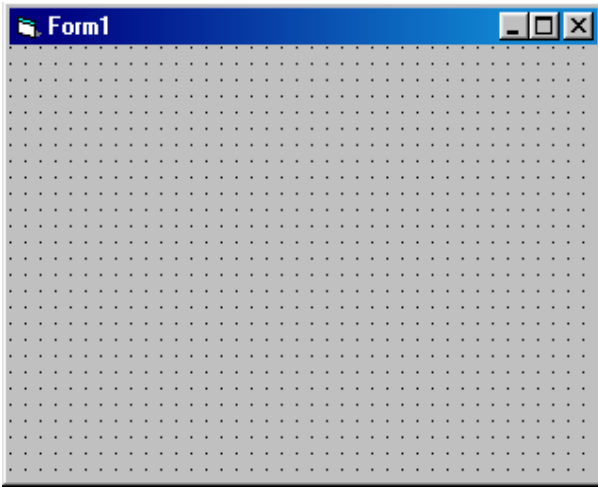
٢- طريقة الرسم .

وضع الادوات بالنقر المزدوج :

أذا نقرت على أي أداة موجودة في نافذة الادوات نقرأ مزدوجا ، ستجد VB قد نقل نسخة الى منتصف نافذة البرنامج واصبحت محاطة بمربعات صغيرة .

طريقة الرسم :

- انقر الاداة التي تريد استخدامها في نافذة الادوات نقرة واحدة فقط .
- حرك المؤشر الى نافذة البرنامج ، ستلاحظ ان شكل المؤشر تغير وهذا معناه ان VB في طور الرسم
- ضع المؤشر في المكان الذي تريد وضع الاداة فيه ثم اضغط زر الفأرة الايسر واحتفظ به مضغوطا ثم اسحب الاداة .
- حرك المستطيل الى ان يصل الحجم المطلوب ثم حرر زر الفأرة .

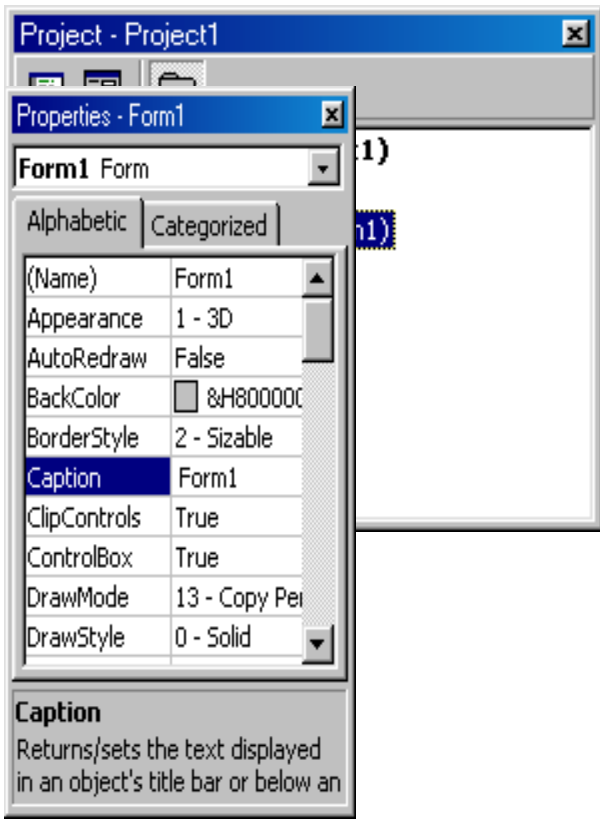


٢. نافذة البرنامج :

او مايسميه VB بالفورم Form و هو عبارة عن نموذج فارغ نكوّن و نضع عليه الرسوم و الصور و الأزرار و عناصر الإدخال و الإخراج و غير ذلك . في الحقيقة هذه النافذة تمثل النافذة التي ستظهر للمستخدم عند انتهاء البرنامج وسيكون معظم عملنا في VB داخل هذه النافذة. لاحظ ان برنامجك قد يحتوي على اكثر من نافذة.

٣. نافذة المشروع :

وتحتوي على الملفات المختلفة التي يتضمنها برنامجك .



٤. نافذة الخصائص Properties Window :

هذه النافذة تحتوي على خصائص جميع الكائنات والأدوات التي تضيفها لمشروعك، بما فيها خصائص الفورم . لاحظ ان اسم الكائن يظهر في اعلى هذه النافذة وان خصائصه تظهر في منتصف النافذة ويظهر في الاسفل شرح هذا الكائن. وهذه النافذة كما ترون تحتوي على الخاصية في اليسار وقيمتها في اليمين . فمثلا الخاصية Caption تجدون أن قيمتها Form1 وهكذا

كل كائن في البرنامج يحتوي على مجموعة من الخصائص التي تختلف من كائن لآخر وكذلك عدد من الخصائص المشتركة بين الكائنات ... فمثلا كل الكائنات تمتلك الخاصية Name وذلك لأن هذه الخاصية من الخواص اللازمة في الفيجول بيسك.. عندما تريد في استعراض او تغيير خصائص احدى النوافذ او الادوات في مشروعك ، انقر فوق هذه النافذة او الاداة لتنشيطها .فيما يلي ترجمة لنافذة الخصائص :

الخاصية	شرحها
Alignment	تحديد مكان النص، في اليمين او الشمال او في الوسط
Appearance	تحديد شكل الاداة اما بشكل عادي او بشكل ثلاثي الابعاد
Autosize	اذا كانت True يعني ان الاداة ستتخذ حجم تلقائي يناسب محتوياتها
BackColor	لون الخلفية
BackStyle	شكل الخلفية اما شفاف او ملون
BorderStyle	سمك إطار النافذة
Caption	العنوان او النص العنوان الذي سيظهر للمستخدم فوق الأداة
control box	ما يظهر في أعلى كل نافذة في أقصى اليمين للتحكم بالنافذة
Datafield	حقل قواعد البيانات
DataFormat	هيئة قواعد البيانات، اما نص او رقم الخ
Datasource	مصدر قواعد البيانات، او الاداة التي تتحكم في قواعد البيانات
Enabled	تمكين الاداة، فهي عادة True ، ولكن عندما تكون False تكون الاداة بلون رمادي وغير قابلة للاستخدام
Font	الخط
ForeColor	لون الخط

طول الاداة	Height
تستخدم اذا كانت الاداة ضمن مصفوفة، فيكون رقم Index هو رقم الاداة داخل المصفوفة	Index
الفترة الزمنية المراد تنفيذ حدث المؤقت بعد مرورها	interval
التغيير الكبير	large change
موقع الاداة الافقي	Left
الحد الأعلى	max
الحد الأدنى	min
ايقونة الماوس	MouseIcon
مؤشر الماوس	MousePointer
تستخدم في البرامج العربية لتحويل الاداة من اليمين الى اليسار	RightToLeft
التغيير الصغير	small change
ترتيب الاداة من حيث التنقل بالضغط على الزر Tab	TabIndex
تستخدم لتخزين اي بيانات اضافية	Tag
تحديد النص الذي سيظهر عند ايقاف الماوس فوق الاداة	ToolTipText
تحديد مكان الاداة العامودي	Top
القيمة	value
اظهار و اخفاء الاداة	Visible
عرض الاداة	Width

الخصائص المشتركة .

خاصية العنوان Caption

تستخدم هذه الخاصية عندما تريد تغيير عنوان زر الامر من command1 الى خروج مثلاً متوفر أثناء التصميم والتنفيذ ، هذه الخاصية متوفرة لجميع الادوات

خاصية الاسم Name

متوفرة أثناء: التصميم فقط

خاصية الاسم خاصية تشترك فيها كل الادوات وهي تحدد الاسم الذي ستستخدمه أنت للإشارة الى الاداة أثناء كتابة البرنامج . فكما قلنا عندما تضع اداة معينة على نافذة البرمجة يقوم VB بتسميتها بطريقة اتوماتيكية باضافة الاسم ثم رقم مثل form1 أو command1 أو label1 كل هذه اسماء يقوم VB بوضعها

خصائص اللون

Forecolor للون الكتابة

Backcolor للون الخلفية

متوفر أثناء : التصميم والتنفيذ

تتحكم هاتان الخاصيتان في لون الكتابة ولون الخلفية فمثلا الامر

label1.ForeColor = vbcolor (١)

وهذا يعني قم يا VB بتغيير لون الكتابة لاداة العنوان الى اللون الازرق اثناء اشتغال البرنامج.

خصائص الخط المستخدم في الكتابة

اسم الخاصية : fontbold, fontitalic, fontname, fontsize, fontstrike, fontunderline

متوفر أثناء التصميم والتنفيذ ،تتحكم هذه الخصائص في اسم وصفات الخط المستخدم في الكتابة .

خاصية الظهور Visible

متوفر أثناء : التصميم والتنفيذ

قد تريد أحيانا أن تخفي بعض الادوات وأن تظهر بعضها تبعاً لطبيعة برنامجك هذه الخاصية تمكنك من عمل ذلك فملاً الأمر

Text1.Visible = false

يؤدي الى اختفاء خانة النص text1 بينما الامر

Text1.Visible = true

يعيد أظهارها

لايظهر تأثير هذه الخاصية الا عند تشغيل البرنامج

خاصية التمكين enabled

متوفرة أثناء : التصميم والتنفيذ

تستخدم هذه الخاصية عندما تريد أن تبقى الاداة ظاهرة على النافذة ولكن لا تريد أن تتيح للمستخدم أن يتعامل معها فاذا غيرت هذه الخاصية الى false لاداة فانها ستظهر خافته على النافذة مما يعطى المستخدم تلميحا الى انه لن يتمكن من استخدامها الان

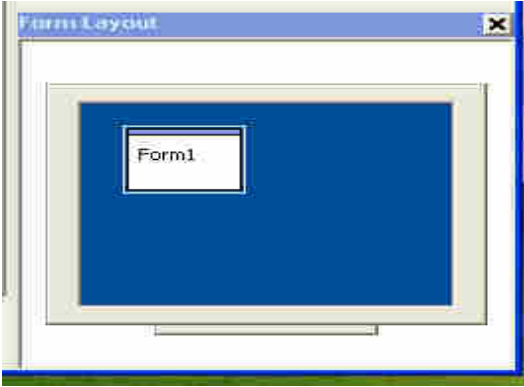
٥. نافذة البرمجة :

تظهر نافذة البرمجة Code Window عند النقر المزدوج على نافذة البرنامج او احد الادوات الموجودة عليها او عند النقر على زر Code View من نافذة المشروع وتستخدم هذه النافذة في اضافة التعليمات Code التي ترغب في تنفيذها .

٦. نافذة اكتشاف الاخطاء :

لا تظهر نافذة اكتشاف الاخطاء Debug Window الا عند تشغيل البرنامج . وهي تلعب دورا حيويا في مرحلة اكتشاف الاخطاء التي توجد في برنامجك .

٧. نافذة تخطيط النوافذ :



هل ترى تلك الشاشة الصغيرة داخل نافذة تخطيط النوافذ ؟؟ انها تمثل شاشة المستخدم والمربع الذي بداخلها يمثل نافذة برنامجك .. كما تلاحظ بالشكل فان نافذة لبرنامجك تقع في الجهة العليا اليسرى وهو المكان الذي ستظهر فيه نافذتك عندما تقوم بتنفيذ البرنامج ويمكن التحكم بمكان النافذة اما عن طريق نافذة تخطيط النوافذ (بالسحب والافلات) او اضافة اوامر برمجية اخرى .

مراحل كتابة البرنامج :

كما هو الحال في البرمجة بأي لغة تأتي الفكرة في المقدمة ، بل ان فكرة البرنامج قد تكون هي العامل الرئيسي في تحديد البيئة التي ستكتب بها ، بعد ان تحدد فكرة البرنامج ، تبدأ في تقسيم البرنامج الى اجزاء مستقلة يقوم كل جزء منها باداء دور معين ، وبعد ان تنتهي من اعداد فكرة البرنامج ، سيمر برنامجك خلال المراحل التالية :

(١) تصميم واجهة البرنامج Drawing the Interface

بعد ان تحدد فكرة البرنامج ، تبدء عملية تصميمه ويجب هنا ان تجيب على عدة اسئلة : كم نافذة يحتاج البرنامج ؟ ماهي الادوات التي تحتاجها ؟
ثم تبدأ في تحديد شكل البرنامج بوضع الادوات المختلفة على نافذة او نوافذ البرنامج .

(٢) ضبط الخصائص Setting Properties

بعد ان تضع الادوات على النافذة تاتي مرحلة تحديد سلوك هذه الادوات توجد لكل أداة من الادوات بما في ذلك نافذة البرنامج عدة خصائص Properties تحدد شكلها ولونها والخط المستخدم فيها وعنوانها وغير ذلك وتقوم في مرحلة ضبط الخصائص بتغيير الخصائص الافتراضية لهذه الادوات لتناسب برنامجك فمثلا يفترض "فيجول بيسك" أنك تريد لون خلفية نافذة برنامجك بنفس اللون الذي يختاره المستخدم فاذا كنت لا تريد أن تظهر خلفية النافذة باللون الاسود مثلا فما عليك سوى أن تغير خاصية لون الخلفية Back Color
label. Forecolor =qbcolor (١)

(٣) كتابة البرمجة Writing Code

بعد أن تنتهي من المرحلتين السابقتين تاتي مرحلة كتابة التعليمات التي من المطلوب من "الفيجول بيسك" أن يقوم بتنفيذها عند وقوع حدث معين .
ضبط الخصائص

هناك طريقتان لضبط الخصائص

١. اثناء تصميم البرنامج .
٢. اثناء تشغيل البرنامج .

ضبط الخصائص أثناء تصميم البرنامج .

كما ذكرنا سابقاً يتم ضبط الخصائص أثناء تصميم البرنامج باستخدام نافذة الخصائص Properties Window وهناك ثلاثة خطوات تمر بها عملية الخصائص هي

١. تحديد الاداة التي تريد ضبط خصائصها (يمكنك تحديد أكثر من أداة) .
٢. تحديد الخاصية التي تريد تغييرها .
٣. ادخال القيمة الجديدة .

كيفية حفظ المشروع ؟

من قائمة File اختر Save Project حدد المسار الذي تريد حفظ المشروع فيه و اعط اسم لمشروعك ثم اضغط على save و بهذا تكون قد حفظت برنامجك .

كيفية فتح مشروع محفوظ مسبقاً؟

لفتح مشروع محفوظ مسبقاً نذهب الى القائمة File و نختار Open Project عندها تظهر لك نافذة اسمها Open Project في تبويب Existing تظهر لك البرامج المحفوظة بالتنسيق الذي تم الحفظ به مسبقاً ، اختر البرنامج المطلوب ثم انقر فوق Open فيفتح برنامجك.

كيف تغير عنوان الفورم من كلمة form1 إلى العنوان الذي نريده ؟

انقر فوق الفورم نقرة واحدة (لتحده أو تختاره) ، ثم اذهب لنافذة الخصائص properties window ثم اخترالخاصية caption و اكتب : " التطبيق الأول " بدلا من form1.

مثال ١ :

برنامجنا اليوم هو برنامج حسابي بسيط نتعرف من خلاله على استخدامات أداتي صندوق النص TextBox والملصق Label وأزرار الاوامر CommandButton .

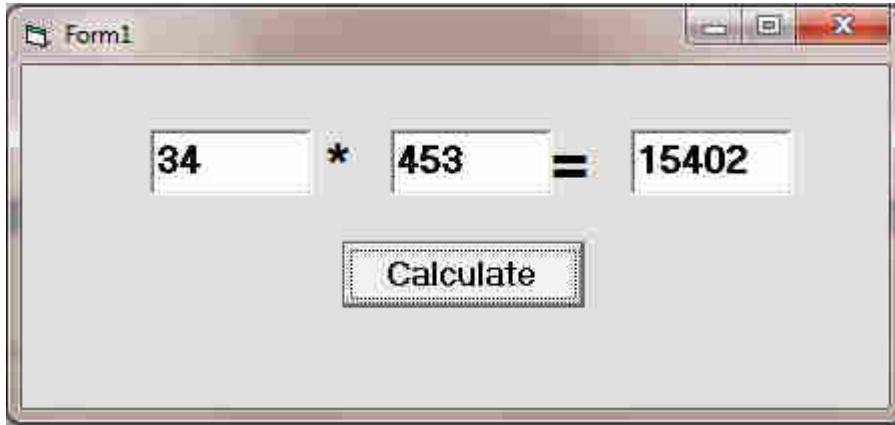
في مشروعنا هذا نفترض أننا نريد أن ننشئ برنامجاً يقوم بإيجاد حاصل ضرب عددين ، سنحتاج الى أداة نص اسمها التلقائي (Text1) ندخل بها العدد المضروب ونحتاج أيضاً ملصق اسمه (Label1) نكتب به علامة الضرب ليعرف من سيتعامل مع البرنامج وظيفته كذلك نحتاج الى أداة نص أخرى اسمها (Text2) أيضاً ناتي بملصق ثان باسم (Label2) نضع فيه علامة المساواة (=) وبعد مربع نص ثالث اسمه (Text3) لنتكتب به النتيجة وبعد أن نرتبها نضع زر الامر (Command1) وبعد أن تكتمل أدواتنا التي نحتاجها في التنفيذ نقوم بتغيير العنوان (وليس الاسم) أي نغير ما يظهر للمستخدم فقط فمثلا Label1 نذهب الى خصائص المشروع ونبحث عن خانة العنوان Caption فنقوم بمسح كلمة Label1 ونكتب بدلاً عنها علامة الضرب (*) ونفس الاجراء مع Label2 حيث نقوم بتبديل عنوانها الى علامة المساواة (=) وكلمة Command1 في زر الامر أيضاً ليست مفهومة فنقوم بتبديلها الى كلمة (احسب) أو (حساب) أو (تنفيذ) ثم ننقل الى ازالة النصوص التي في أدوات النص الثلاثة حيث نبحث عن خاصية Text في خصائص المشروع ونجد أن أمامها نفس مسمى أداة النص بمعنى أن أداة Text1 موجود في خاصية Text كلمة Text1 فنقوم بمسحها وهكذا مع بقية أدوات النص وبعد أن انتهينا من ترتيب الأدوات وتنظيم شكل البرنامج نقوم بكتابة الكود البرمجي وهو بسيط جداً إذا فهمنا فكرته استطعنا اجراء أي عمليات حسابية حيث سنقوم بالتعامل مع مربعات النصوص على أنها متغيرات (س،ص) و أول اجراء نتخذه هو النقر المزدوج على الزر الذي وضعناه واسميناه (احسب) حيث تفتح لنا صفحة فيها سطران بينهما فراغ نكتب به العملية المطلوب تنفيذها فعندنا Text3 هو حاصل ضرب Text1 في Text2 فنعبر عن هذه العملية بالتالي :

Text3.Text = Text1.Text * Text2 . Text

ولاحظ هنا أن علامة الضرب بالبرمجة (*) وبهذه المنطقية يقوم البرنامج باجراء العملية الحسابية ليصبح الكود بصورته النهائية على النحو التالي

```
Private sub command _ click ( )
Text3 . Text = Text1 .Text * Text2 . Text
End sub
```

والان اصبح البرنامج جاهزاً للتنفيذ ضع الارقام وستجد النتيجة مطابقة لها في الالة الحاسبة .
ولاحظ هنا أن علامة الضرب بالبرمجة (*) وبهذه المنطقية يقوم البرنامج باجراء العملية الحسابية ليصبح الكود بصورته النهائية على النحو الموضح والان اصبح البرنامج جاهزاً للتنفيذ ضع الارقام وستجد النتيجة مطابقة لها في الالة الحاسبة .كما موضح في الشكل التالي.



ولتحويل المشروع الى برنامج ذاتي التنفيذ exe توجه الى قائمة ملف ومنه تجد Make Projectl.exe وهو المسئول عن تنفيذ البرامج .

تعلمنا في التطبيق السابق كيفية استخدام بعض الوظائف والادوات فتعلمنا كيف نجري عملية الضرب وبنفس الطريقة يمكننا أن نجري العمليات الأخرى كالطرح والقسمة لكن الجمع يختلف لانه يتعامل مع العملية على أنها صف فاذا كتبت قانون الجمع بهذه الصورة :

Text3 . text = Text1. text + Text2 . text

فان القيم التي يتم ادخالها في خانة Text1 و Text2 ستظهر النتيجة على أن الخانة الاولى احاد والثانية عشرات بمعنى انه اذا جمع ٢+٣ ستظهر لك النتيجة ٢٣ ويكون المخرج من هذه المشكلة بتعديل صيغة القانون ليصبح :

Text3. text = Val (Text1. Text) + Val(Text2. text)

مثال ٢ :

صمم نموذج كما في الشكل، يعمل على حساب مساحة دائرة، حيث يحتوي على صندوق نص (Text1)، و زر أمر (Command1)، بحيث يتم إدخال نصف القطر للدائرة في صندوق النص، ثم يتم حساب المساحة عند النقر على زر الأمر وإظهار النتيجة في أداة العنوان (Label1)، مراعيًا في تنفيذ البرنامج استخدام الثوابت.



الشكل: مرحلة التصميم للنموذج

الشفرة اللازمة لتنفيذ البرنامج كالآتي:

```
Private Sub Command1_Click()  
pi = 3.14  
Label1.Caption = Val(Text1.Text) ^ 2 * pi  
End Sub
```

بعد تنفيذ البرنامج سيظهر كما في الشكل (بالاعتماد على البيانات التي يتم إدخالها في صندوق النص).



الشكل: مرحلة التنفيذ

تمرين ١:

اكتب الشفرة اللازمة لعمل آلة حاسبة بسيطة بالشروط التالية:



يتم تنفيذ وإجراء العملية الحسابية بالنقر على زر الأمر المخصص لها كما في الشكل المرفق، وحسب الأسماء Command1 لعملية الجمع، وCommand2 لعملية الطرح، وCommand3 لعملية الضرب، وCommand4 لعملية القسمة، وCommand5 لمسح محتوى Text1، وText2، وText3، وصاديق

النص الأول Text1 لكتابة الرقم الأول والثاني Text2 لكتابة الرقم الثاني والثالث Text3 لظهار نتيجة العملية المنفذة.

تمرين ٢:

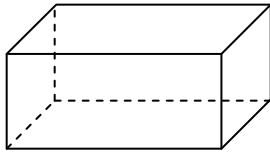
كون برنامج يقوم بحساب الحجم والمساحة الجانبية ومساحة القاعدة لمتوازي المستطيلات.

المعطيات:

الحجم = الطول * العرض * الارتفاع

المساحة الجانبية = ٢ * الارتفاع * (الطول + العرض)

مساحة القاعدة = الطول * العرض

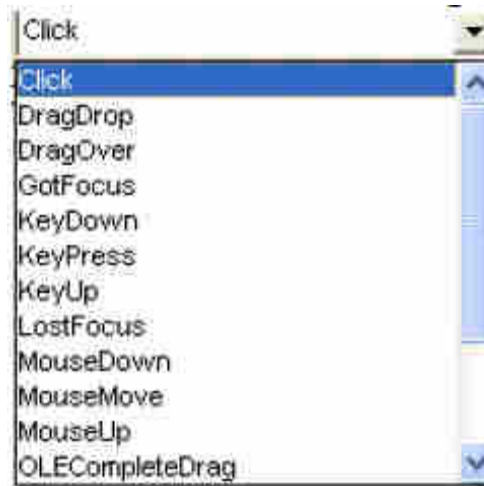


كتابة البرمجة Writing Code

لكي تقوم ببرمجة مفتاح معين لابد وان تظهر نافذة البرمجة حيث ان هذه النافذة تحتوي على الاوامر او الاكواد لبرمجة مفتاح معين. والطريقة السريعة لاطهار نافذة البرنامج ان تقوم بالضغط على المفتاح مرتين متتاليتين فتظهر النافذة بالشكل التالي:-



نلاحظ في هذه النافذة وجود صندوقان للقائمة احدهما يحتوي على جميع الكائنات الموجودة على شاشة النموذج والاخر يحتوي على جميع الاحداث التي تحدث على الزر وهي هنا Click اي عند الضغط على هذا المفتاح.



ونلاحظ وجود سطرين في نافذة البرمجة وهما:-

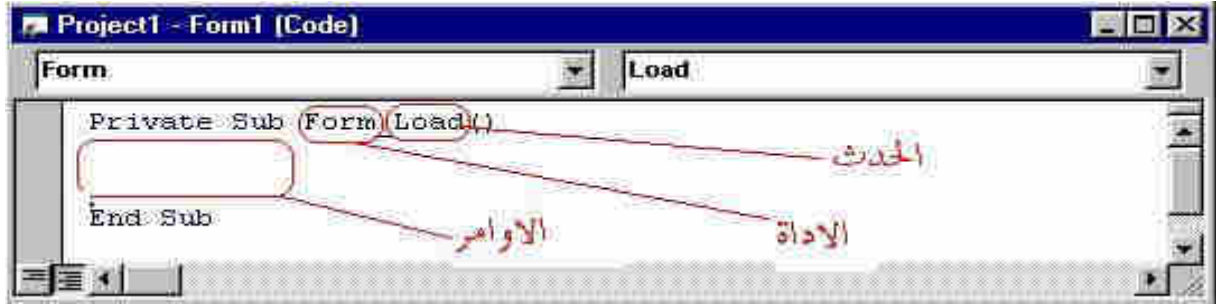
السطر الاول:- Private Sub Command1_Click()

حيث ان هذا السطر يعني ان الحدث خاص سيقع على زر الامر فقط ونلاحظ وجود كلمة command1 اي اسم الزر في خاصية Name ثم وجود Click اي عند الضغط عليه.

السطر الثاني: End Sub وهو يعني انتهاء الحدث.

ما هو الحدث ؟

الحدث Event هو كل مايقوم به المستخدم من حركات بالفأرة كالنقر أو النقر المزدوج و يستجيب لها برنامج فيجوال بيسك ولولا وجودها لما استطعنا ان نحدد للبرنامج متى ينفذ الامر الذي نريده ان ينفذ.



الحدث	وصفة
active	يقع بعد تحميل النافذة أي بعد وقوع حدث load
click	عند النقر المفرد على الفارة
dblclick	عند النقر المزدوج على الفارة
gotfocus	عند اخذ التركيز لاداة معينة
lostfocus	عند فقد التركيز لاداة معينة
keydown	عند نزول الزر الى الاسفل
keypress	عند الضغط على أي زر من لوحة المفاتيح
load	عند تحميل النافذة الرئيسية
mousedown	عند نزول زر الفارة الى الاسفل
mousemove	عند تحرك الفارة على اداة معينة
mouseup	عند طلوع زر الفارة الى الاعلى
paint	عند اعادة رسم النافذة
resize	عند تغيير حجم النافذة
unload	عند الغاء تحميل النافذة



أداة الوقت Timer:

"أداة الوقت أداة غير مرئية" حيث ان المقصود بذلك هي اداة لتوقيت احداث معينة لا يراد للمستخدم معرفة الوقت ويمكن أدرج أداة الوقت Timer في أي مكان في النموذج.
مثال ١: نقوم في هذا البرنامج باستخدام تقنية برمجية مهمة جداً ، ألا و هي استخدام المؤقت " العداد " .

أولاً: قم بإنشاء الكائنات الموجودة في الشكل وغير الخائص وكما يلي:

الكائن	الخاصية	الإعداد
Form1	Caption	برنامج الساعة الرقمية
	Right to lift	True
Label1	Caption	“ “
Timer1	Interval	1000

سوف تظهر نافذة البرنامج كما في الشكل:



ثانياً: كتابة الشيفرة الخاصة بالبرنامج:

كتابة الشيفرة الخاصة بالكائن (Timer1):

```
Private Sub Timer1_Timer()  
Label1.Caption = Time  
End Sub
```

وعند تشغيل البرنامج سوف يصبح كالتالي:



شرح مبسط للكود السابق:

إن العداد مشابه تماماً للعدادات التي نستخدمها في حياتنا العملية ، ولكن العداد الموجود في فيجوال بيسك متطور ومتقدم حيث يمكن ضبط عن طريق الخصائص ، و يتم استخدام هذا العداد في التحكم ببعض كائنات البرنامج ، فعلى سبيل المثال قمنا بجعل العداد يتحكم بأداة العنوان (Label1) بحيث تظهر الوقت ، فيقوم العداد بتغيير الوقت كل ٠.٠٠١ من الثانية ، فيتغير الوقت تلقائياً بعد مرور ٠.٠٠١ من الثانية ، فنكون بذلك قد كوننا ساعة رقمية.

مثال ٢:

سوف نقوم بعمل برنامج بسيط يحتوي على مفاتيح لإظهار التاريخ والوقت ورسالة ترحيبية ومفتاح للمسح واخر للخروج من البرنامج .
أولاً: قم بإنشاء الكائنات الموجودة في الجدول .

الكائن	الخاصية	الإعداد
Form1	Caption	البرنامج الترحيبي
	Right to lift	True
Text1	Caption	“ “
Command1	Caption	أظهر الوقت الحالي
Command2	Caption	أظهر التاريخ الحالي
Command3	Caption	أظهر الترحيب الخاص
Command4	Caption	مسح
Command5	Caption	خروج

سوف تظهر نافذة البرنامج كما في الشكل



ثانياً: كتابة الشيفرة الخاصة بالبرنامج:

ملاحظة: إن كتابة الشيفرة الخاصة بالبرنامج تعني كتابة الأوامر واستخدامها في برمجة البرنامج .

كتابة الشيفرة الخاصة بزر إظهار الوقت الحالي (Command1):

Text1.Text = Time

كتابة الشيفرة الخاصة بإظهار التاريخ الحالي (Command2):

Text1.Text = Date

كتابة الشيفرة الخاصة بإظهار النص الترحيبي (Command3):

Text1.Text = "مرحباً بك في عصر فيجوال بيسك"

كتابة الشيفرة الخاصة بمسح محتويات مربع النص (Command4):

Text1.Text = ""

كتابة الشيفرة الخاصة بالخروج من البرنامج (Command5):

End

شرح مبسط للكود السابق:

في كتابة الشيفرة للزر الخاص بإظهار الوقت الحالي اسندنا قيمة الوقت والمعرفة في لغة فيجوال بيسك باسم (Time) إلى صندوق النص ، وفي الكود الخاص بالتاريخ قمنا بعملية مشابهة ، حيث اسندنا قيمة التاريخ لصندوق النص ، وفي الكود الخاص بالنص الترحيبي قمنا بإسناد قيمة حرفية لصندوق النص ، وعندما أردنا إخلاء صندوق النص أمرنا فيجوال بيسك بأن يجعل في هذا الصندوق قيمة خالية ، أي يجعله فارغاً . ملاحظة عندما كتبنا كود الخروج من البرنامج كتبنا كلمة (End) وهي تقوم بإغلاق البرنامج بجميع محتوياته.

مربعات سرد محركات الأقراص والدلائل والملفات

مربعات سرد محركات الأقراص التي تتيح لك رؤية محركات الأقراص الصالحة في الكمبيوتر و مربعات سرد الدلائل التي تتيح لك التنقل بين المجلدات الموجودة في محرك أقراص معين، ومربعات سرد الملفات التي تتيح لك انتقاء ملف معين في مجلد.

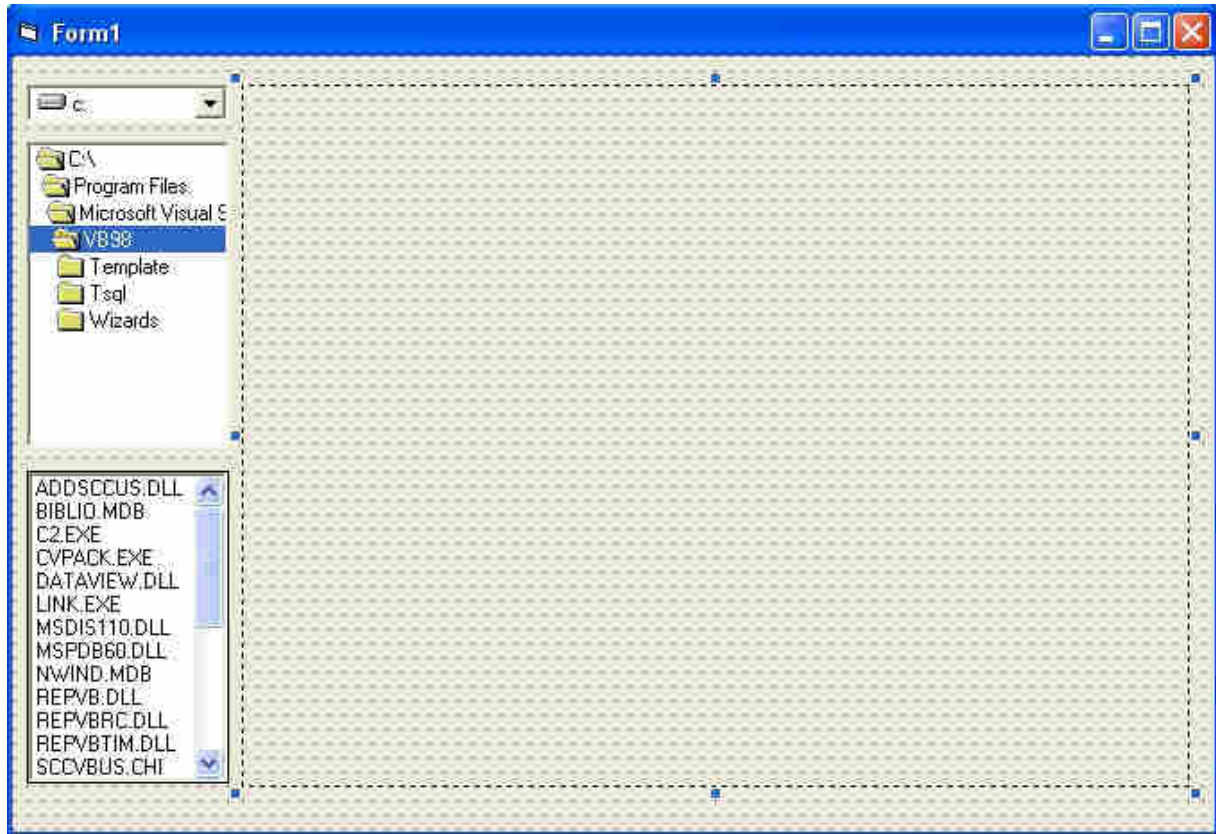
في التمرين التالي ستستعمل هذه الكائنات الثلاثة لبناء برنامج Photo Browser يحدد موقع الملفات التي تحتوي على أعمال فنية في الكمبيوتر ويعرضها.

مثال ٣ : البرنامج Photo Browser

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

بناء البرنامج Photo Browser

١. أنشأ برنامج وصم الواجهة التالية :



٢. اضبط خصائص النموذج وكائناته حسب الجدول التالي:

الكائن	الخاصية	الإعداد
File1	Pattern	*.bmp; *.jpg; *.gif *.ico;
Image1	Stretch	True

٣. اكتب البرمجة كما يلي :

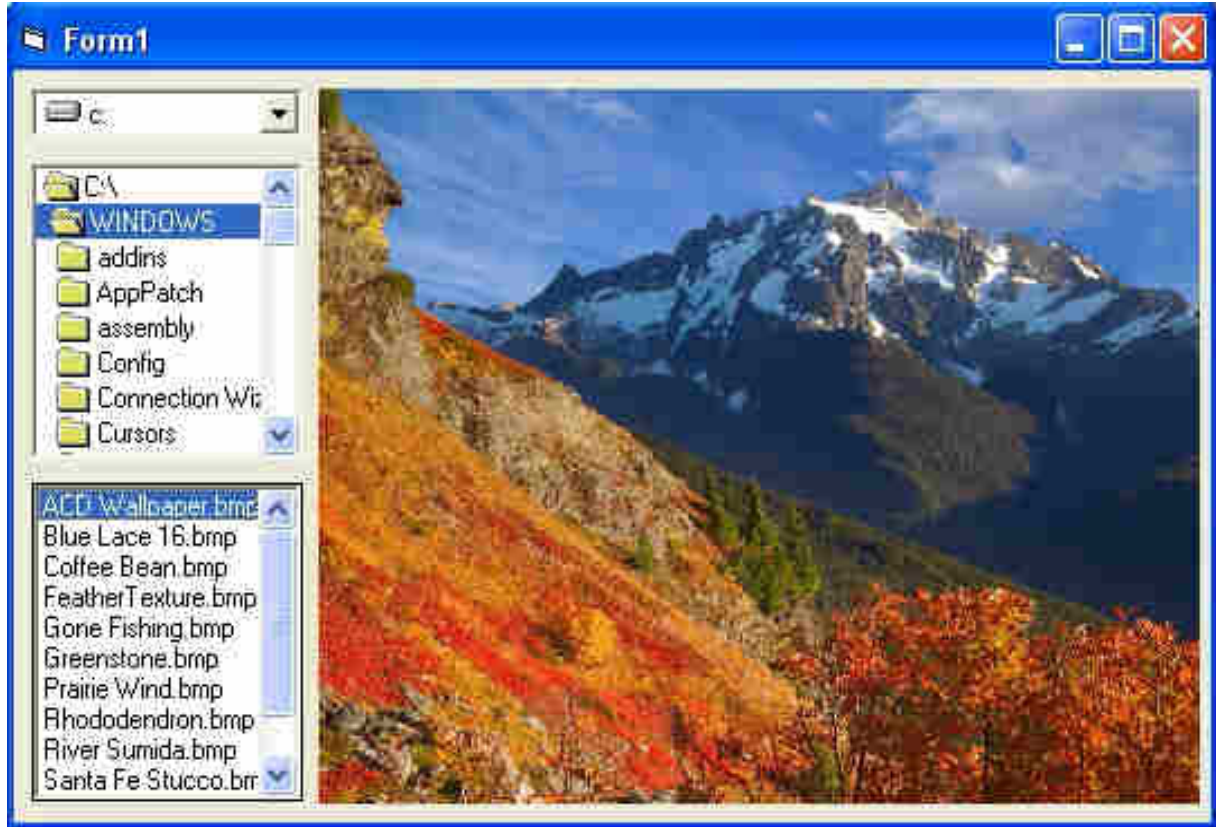
```

Project1 - Form1 (Code)
File1 Click
Private Sub Dir1_Change ()
File1.Path = Dir1.Path
End Sub

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

Private Sub File1_Click ()
SelectedFile = File1.Path & "\" & File1.FileName
Image1.Picture = LoadPicture (SelectedFile)
End Sub
    
```

٤ . وعند تنفيذ البرنامج يحدث التالي :



Mathematics Functions الدوال الرياضية

ما هي الدوال :

الدوال هي أسماء محجوزة ومعرفة من قبل الفيجوال بيسك لتقوم بعمل معين مثل المصفوفات والقيم المطلقة وغيرها. أو بمعنى أبسط واعم هي عبارة عن برنامج صغير مكتوب مسبقا و محفوظ داخل لغة الفيجوال بيسك يمكن استدعاه من خلال برنامجك والاستفادة منه ، إضافة إلى ذلك يمكنك أن تكتب دالة وتستدعيها أكثر من مرة داخل برنامجك . وهناك عدة انواع من الدوال ومنها الدوال الرياضية .

الدوال الرياضية

الدالة **Abs** : تجد القيمة المطلقة لأي عدد والقيمة المطلقة هي قيمة العدد بدون إشارة فالقيمة المطلقة ل (-13) مثلا هي (13) وهكذا، فمثلا لو كتبنا الكود التالي.

A=Abs(-45.6)
Label1.caption = A

فإن نتيجة تنفيذ الدالة هي A=45.6
ولاحظ أن القيمة المدخلة للدالة لا بد أن تكون عدد فإذا كانت القيمة المدخلة للدالة Null ستكون النتيجة Null وإذا كانت القيمة المدخلة للدالة متغير فارغ أو لم يتم تعيين قيمة له ستكون النتيجة خطأ .
يمكن استخدام صندوق النص Text لادخال اي قيمة للبرنامج كما يمكن استخدام ملصق Label لاجراج اي ناتج:

Label1.caption = Abs(text1.text)

الدالة **Sqr** : تستخدم هذه الدالة في تحديد الجذر التربيعي لرقم معين وتأخذ الصورة العامة التالية.

A=Sqr(25)

فإن نتيجة تنفيذ الدالة هي A=5

الدالة **Log**: تستخدم هذه الدالة في تحديد قيمة اللوغاريتم العشري لرقم وتأخذ الصورة العامة التالية :

A=Log (20)

فإن نتيجة تنفيذ الدالة هي A=2.9957327

الدالة **Exp**: تستخدم هذه الدالة في تحديد القيمة (e) وهي قاعدة اللوغاريتم الطبيعي مرفوعة بقوة الرقم الذي تتضمنه حيث (e) تساوي تقريبا 2.7182818 وتأخذ الصورة العامة التالية:

A=Exp (رقم)

الدالة **Rnd** : وتستخدم هذا الدالة في توليد أرقام عشوائية تقع ما بين الصفر و واحد بحد أقصى ١٥ رقما عشريا وتأخذ الصورة العامة التالية :

A=Rnd

فمثلا الدالة Rnd قد تعطي رقما مثل 0.7055475 وعند تشغيل الدالة مرة أخرى ينتج رقما آخر مثل 0.533424 وهكذا.

لتوليد أرقام عشوائية من واحد إلى عشرة .

A=Int(10 * Rnd) + 1

ولتوليد أرقام عشوائية من واحد إلى مئة .

A=Int(100 * Rnd) + 1

الدالة **Sin** : تستخدم هذه الدالة في تحديد قيمة جيب زاوية معينة وتأخذ الصورة العامة التالية:

$$A = \text{Sin}(\text{رقم})$$

الدالة **Cos** : تستخدم هذه الدالة في تحديد قيمة جيب تمام الزاوية معينة وتأخذ الصورة العامة التالية:

$$A = \text{Cos}(\text{رقم})$$

الدالة **Tan** : تستخدم هذه الدالة في تحديد قيمة ظل زاوية معينة وتأخذ الصورة العامة التالية:

$$A = \text{Tan}(\text{رقم})$$

الدالة **Atn** : تستخدم هذه الدالة في حساب مقلوب ظل الزاوية "ظنا" للرقم الذي تشتمل عليه مقدار بالتقدير الدائري وتأخذ الصورة العامة التالية:

$$A = \text{Atn}(\text{رقم})$$

الدالة **Round** : وهي دالة التقريب التي من خلالها يمكنك تحديد عدد الأرقام العشرية وتأخذ الصورة التالية:

$$A = \text{Round}(4.52696, 2)$$

حيث 4.52696 الرقم المراد تقريبه أما الرقم 2 فهو عدد المراتب التي ستقرب بعد العلامة العشرية وسيكون الناتج 4.53 .

الدالة **Int** : وتستخدم هذه الدالة لحساب الجزء الصحيح فقط من رقم يشتمل على أرقام صحيحة وعشرية أو بعبارة آخر لحذف الأرقام العشرية الموجودة بعد العلامة العشرية بدون تقريب وتأخذ الصورة التالية:

$$A = \text{Int}(332.54)$$

فإن نتيجة تنفيذ الدالة هي $A = 332$

الدالة **Fix** : وهي تشبه الدالة **Int** ولكنها تستخدم لحساب الجزء الصحيح فقط وتأخذ الصورة التالية:

$$A = \text{Fix}(4.52696)$$

فإن نتيجة تنفيذ الدالة هي $A = 4$

الدالة **Val** : وهي تحول السلاسل النصية إلى قيمة رقمية وهي عكس الدالة **Str()** التي تحول الأرقام إلى سلاسل نصية وتأخذ الصورة التالية:

$$A = \text{Val}(\text{text1}.text)$$

العمليات الحسابية

إن العمليات المستخدمة في التعبيرات الحسابية هي : الجمع + ، الطرح - ، الضرب * ، القسمة / ، القسمة الصحيحة \ ، الرفع للقوة ^ ، الأقواس () .
فمثلا العبارات الرياضية تكتب بلغة البيسك وكما يلي :

رياضياً	بلغة البيسك
$\frac{3 \cos(2x)}{1+x^2}$	$3 * \cos(2 * x) / (1 + x^2)$
$\frac{1+t^3}{\sin(4t+7)}$	$(1+t^3) / \sin(4*t+7)$
$\frac{ x-4 }{x^2+6}$	$\text{abs}(x-4) / (x^2+6)$
$\sqrt{1+ab}$	$\text{sqr}(1+a*b)$

أولوية العمليات الحسابية

ويقصد بالأولوية أي تسلسل العمليات التي تنفذ أولاً فمثلاً العبارة : $2 * 7 / 5^2 + 1$ لا نستطيع تحديد قيمتها قبل معرفة ترتيب التنفيذ للعمليات الحسابية الموجودة بالعبارة. كما في اغلب لغات البرمجة تكون أولويات تنفيذ العمليات الحسابية في الفيجوال بيسك كما يلي :-

- ١- فك ما بين الاقواس ()
- ٢- عمليات الرفع للأسس.
- ٣- عمليات الضرب او القسمة او ايهما اولاً.
- ٤- قسمة الاعداد الصحيحة .
- ٥- الجمع او الطرح او ايهما اولاً .

فمثلا ناتج المعادلات التالية يكون كما يلي :

$$1) \quad 2 + 3 * 2 \rightarrow 8$$

$\underbrace{\hspace{1.5cm}}_2$
 $\underbrace{\hspace{0.5cm}}_1$

$$2) \quad 2 * 3^2 + 1 \rightarrow 19$$

$\underbrace{\hspace{1.5cm}}_2$
 $\underbrace{\hspace{0.5cm}}_1$
 $\underbrace{\hspace{2.5cm}}_3$

مثال ١ : المطلوب عمل برنامج لحساب معدلات الجريان الحجمية والوزنية في برنامج واحد مع إضافة مفتاح لغلق البرنامج علما ان المعلومات المدخلة للبرنامج هي القطر والسرعة والكثافة. علما ان:

$$\text{Area} = (3.14/4) * \text{Diameter}^2$$

$$\text{Volumetric flow rate} = \text{Area} * \text{Velocity}$$

$$\text{Mass flow rate} = \text{Density} * \text{Volumetric flow rate}$$

لعمل البرنامج نحتاج ثلاثة صناديق نص Text لادخال قيم القطر والسرعة في الانبوب والكثافة للسائل وكذلك نحتاج الى مجموعة من الملصقات Label وزري امر Command . ثم نضبط خصائص الادوات وكما في التالي.

عزيزي الطالب : في المثال الحالي والامثلة اللاحقة لن نقوم بشرح عملية التصميم وضبط الخصائص لادوات لان ذلك اصبح من البديهي لديك.

بعد ضبط الخصائص اكتب الكود البرمجي الموجود في الشكل التالي :

```

Command2 Click
Private Sub Command1_Click()
A = (3.14 / 4) * Text1 ^ 2
Label7 = Text2 * A
Label8 = Label7 * Text3
End Sub
Private Sub Command2_Click()
End
End Sub

```

و عند ادخال قيم القطر والسرعة والكثافة في مربعات النص وتنفيذ البرنامج يحدث التالي :



مثال ٢ : المطلوب عمل برنامج لحساب قيم جيب و جيب تمام و ظل زاوية معينة و ايجاد القيم العشوائية و تفسير القيم.
في البداية ضع مربعي نص Text لادخال قيمة الزاوية واخراج الناتج وخمسة ازرار Command لتنفيذ الاوامر . ثم اضبط الخصائص كما في الشكل التالي .



اكتب الكود البرمجي الموجود في الشكل التالي :

```
Command1 Click
Private Sub Command1_Click()
Text2 = Sin(Text1)
End Sub

Private Sub Command2_Click()
Text2 = Cos(Text1)
End Sub

Private Sub Command3_Click()
Text2 = Tan(Text1)
End Sub

Private Sub Command4_Click()
Text1 = ""
Text2 = ""
End Sub

Private Sub Command5_Click()
Text2 = Rnd
End Sub
```



التنفيذ عند ضغط مفتاح Sin



التنفيذ عند ضغط مفتاح Cos



التنفيذ عند ضغط مفتاح Tan



التنفيذ عند ضغط مفتاح التصفير
C



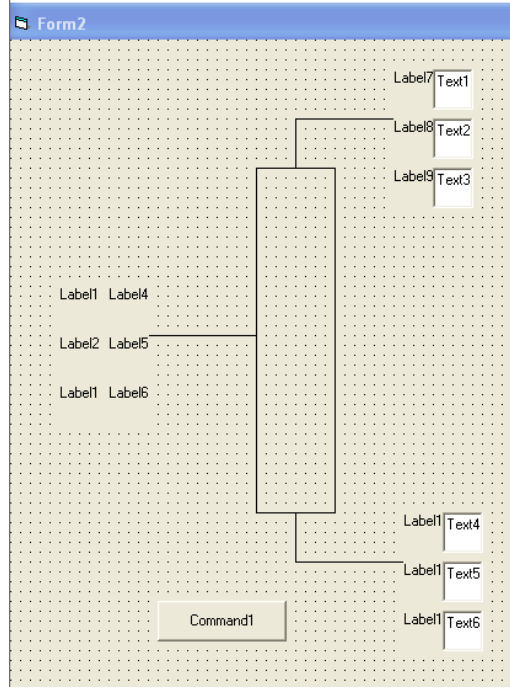
التنفيذ عند ضغط مفتاح القيمة
العشوائية Rnd

البرمجة بلغة فيجوال بيسك
الدوال الرياضية

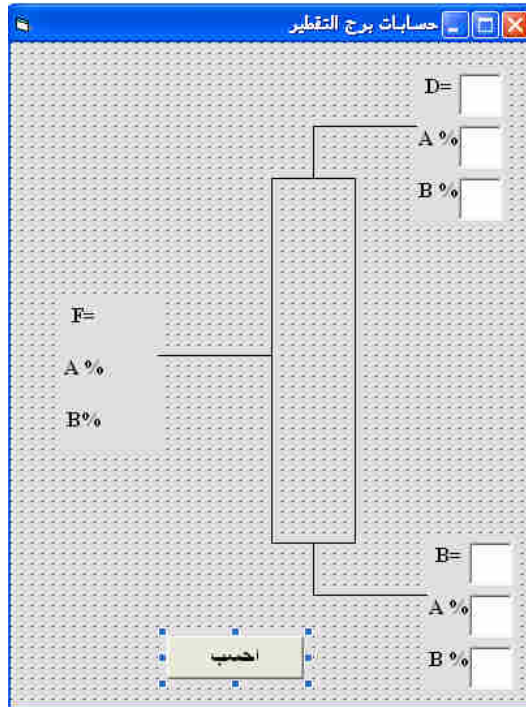
قسم الهندسة الكيماوية
المرحلة الأولى

مثال ٣ :المطلوب عمل برنامج يقوم بحساب معدل جريان وتراكيز المواد الداخلة الى برج التقطير لفصل المادتين A و B بالاعتماد على قيم معدلات الجريان وتراكيز المواد الخارجة من اعلى برج التقطير (المتقطر) واسفل برج التقطي (المتبقي) .

في هذا البرنامج سوف نستخدم اداة Shape لرسم المستطيل واداة Line لرسم الخطوط وبعد ذلك نضيف الادوات Text و Label و Command كما في الشكل التالي:



اضبط خصائص النموذج وكائناته كما في الشكل التالي:



اكتب البرمجة كما يلي :

```
Project1 - Form2 (Code)
Command1 Click
Private Sub Command1_Click()
Label4 = Val(Text1) + Val(Text4)
Label5 = (Text1 * Text2 + Text4 * Text5) / (Val(Text1) + Val(Text4))
Label6 = (Text1 * Text3 + Text4 * Text6) / (Val(Text1) + Val(Text4))
End Sub
```

و عند تنفيذ البرنامج يحدث التالي :

ملاحظة : عزيزي الطالب يمكن تطبيق الخطوات السابقة لحل اي مشكلة قد تصادفك في اختصاصك بالهندسة الكيماوية ويمكنك مستقبلا تعلم الكثير من اوامر البرمجة للاستفادة منها في حل المشاكل الرياضية.

الجملة الشرطية (If)

تعتبر الجملة الشرطية من أهم الأوامر التي لها القدرة على اتخاذ القرارات وتستخدم في معاملات المقارنة ، و يتم تنفيذ أو عدم تنفيذ الأوامر بناء على صحة الشرط . وهناك عدة صيغ لتطبيق الجملة الشرطية.

• صيغة (If –Then) :

الجملة في حال تحقق الشرط Then الشرط If

الصيغة العامة :

مثال ١ :

If g >= 50 Then Text1="ناجح"

• صيغة (If – Then – End If) :

Then الشرط If
الجملة أو مجموعة من الجمل في حال تحقق الشرط
End If

الصيغة العامة :

مثال ٢ :

If g >= 50 Then
Text1 = "مبروك ناجح"
End If

• صيغة (If – Then – Else) :

Then الشرط If
الجملة أو مجموعة من الجمل في حال تحقق الشرط
Else
الجملة أو مجموعة من الجمل في حال عدم تحقق الشرط
End If

الصيغة العامة :

مثال ٣ :

If g >= 50 Then
Text1 = "ناجح"
Else
Text2 = "راسب"
End If

• صيغة (If – Then –ElseIf) :

الصيغة العامة :

```

If الشرط Then
الجملة أو مجموعة الجمل في حال تحقق الشرط الاول
ElseIf
الجملة أو مجموعة الجمل في حال تحقق الشرط الثاني
ElseIf
الجملة أو مجموعة الجمل في حال تحقق الشرط الثالث
:
:
:
Else
الجملة أو مجموعة الجمل في حال عدم تحقق الشروط السابقة كلها
End If
    
```

ملاحظات :

- لا داعي لكتابة الامر else if في حال عدم الحاجة إليه .
- لا داعي لكتابة الامر else في حال عدم الحاجة إليه .

مثال ٤ : كون مشروعاً يطلب من المستخدم إدخال درجته في مادة البرمجة وعند ادخال الدرجة يعطينا البرنامج التقييم تبعاً للجدول التالي :-

التقدير	العلامة
ممتاز	اكبر أو تساوي ٩٠
جيد جداً	٨٠ - ٨٩.٩٩٩
جيد	٧٠ - ٧٩.٩٩٩
متوسط	٦٠ - ٦٩.٩٩٩
مقبول	٥٠ - ٥٩.٩٩٩
ضعيف	اقل من ٥٠

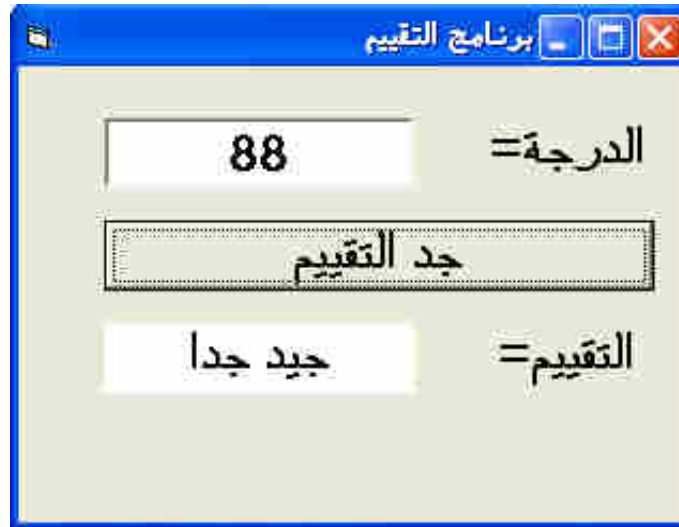
لنبدأ الان بتصميم النموذج ستحتاج إلى مربع نص و زر أمر Command و ثلاثة ملصقات و نرتب النموذج على

النحو التالي :

ثم نكتب الاوامر البرمجية التالية في مفتاح الامر Command :

```
g=val(Text1)
If g >= 90 Then
Label3 ="ممتاز"
ElseIf g >= 80 Then
Label3 ="جيد جدا"
ElseIf g >= 70 Then
Label3 ="جيد"
ElseIf g >= 60 Then
Label3 ="متوسط"
ElseIf g >=50 Then
Label3 ="مقبول"
Else
Label3 ="راسب"
End If
```

عند تشغيل البرنامج ووضع قيمة الدرجة والضغط على زر الامر يحدث التالي :



الجملة الشرطية (Select Case)

تعمل هذه الجملة الشرطية نفس عمل الجملة الشرطية If – Then –Elseif ولاكن يفضل استخدام هذه الجملة عندما يكون للشرط عدة احتمالات.

الصيغة العامة :

```
Select Case التعبير الذي نريد اختباره
Case الاحتمال الأول
جملة أو مجموعة الجمل تنفذ في حال تحقق الاحتمال الأول
Case الاحتمال الثاني
جملة أو مجموعة الجمل تنفذ في حال تحقق الاحتمال الثاني
Case الاحتمال الثالث
جملة أو مجموعة الجمل تنفذ في حال تحقق الاحتمال الثالث
:
:
:
Case Else
جملة أو مجموعة الجمل تنفذ في حال عدم تحقق أي من الاحتمالات أعلاه
End Select
```

مثال ٥ : اعد برمجة المثال السابق باستعمال الجملة الشرطية (Select Case) ؟

```
Select Case Text1
Case 90 To 100
Label3 = "ممتاز"
Case 80 To 89.999
Label3 = "جيد جدا"
Case 70 To 79.999
Label3 = "جيد"
Case 60 To 69.999
Label3 = "متوسط"
Case 50 To 59.999
Label3 = "مقبول"
Case Else
Label3 = "راسب"
End Select
```

فيما يلي جدولاً بالمعاملات الشرطية المستخدمة مع جمل الشرط :

الرمز	العملية	المعنى
=	المساواة	إذا كانت القيم متساوية
<	أقل من	إذا كانت القيمة على اليسار اقل من القيمة على اليمين الاشارة
<=	أقل من او يساوي	إذا كانت القيمة على اليسار اقل او تساوي القيمة على اليمين
>	اكبر من	إذا كانت القيمة على اليسار اكبر من القيمة على اليمين الاشارة
>=	اكبر من او يساوي	إذا كانت القيمة على اليسار اكبر او تساوي القيمة على اليمين
<>	لا يساوي	إذا كانت القيمتين غير متساويتين

الشرط المركب (من جملي شرط) يمكن ان يحتوي على احد التعابير التالية :

And	عندما تكون جملي الشرط صحيحتان تكون النتيجة صح
Or	عندما تكون اي من جملي الشرط صحيحة تكون النتيجة صح
Not	عندما تكون جملي الشرط خطأ تكون النتيجة صح
Xor	عندما تكون احدى جملي الشرط صحيحة تكون النتيجة صح . واذا كانت جملي الشرط صحيحة او جملي الشرط خطأ تكون النتيجة خطأ

مثال ٦ : المطلوب عمل برنامج لحساب رقم رينولد بالاعتماد على ادخال قيم الكثافة والسرعة و القطر واللزوجة . حيث يعطينا البرنامج حالة الجريان وحسب ما يلي :

Laminar if Re. No <2000
Turbulent if Re. No \geq 2000

في البداية صمم النافذة الاتية :

اكتب الكود التالي لحساب رقم رينولد و ايجاد حالة الجريان .

```
Project1 - Form1 (Code)
Command1 Click
Private Sub Command1_Click()
Re = (Text1 * Text2 * Text3) / Text4
If Re < 2000 Then
Label6.Caption = "laminar"
Else
Label6.Caption = "Turbulent"
End If
End Sub
```

والان ادخل قيم الكثافة والسرعة و القطر واللزوجة في الحقول المخصصة لها واضغط على مفتاح الامر:

The screenshot shows a window titled "Flow Type" with a light gray background. At the top, there are four labels: "Density", "Vilocity", "Diameter", and "Viscosity". Below each label is a text box containing a numerical value: "1000", "0.5", "0.02", and "0.01" respectively. In the center of the window is a button labeled "Decide Flow Type". Below the button, the text "laminar" is displayed.

The screenshot shows the same "Flow Type" window. The input values in the text boxes are "1000", "0.5", "0.05", and "0.01". The "Decide Flow Type" button is still present, and now the text "Turbulent" is displayed below it.

الانتقال (GoTo)

هناك نوعان للانتقال وهما :

١. الانتقال الغير مشروط

الصيغة العامة :

حيث line رقم السطر الذي سوف تذهب إليه بدون قيد معين .

مثال ٧ :

GoTo line

Name="أحمد"

GoTo A

.....

A :

Grade=45

.....

٢. الانتقال المشروط

الصيغة العامة :

حيث line رقم السطر الذي سوف تذهب إليه في حال تحقق الشرط .

مثال ٨ :

If g >= 50 Then GoTo A

If g < 50 Then GoTo B

A :

Text1="ناجح"

GoTo C

B :

Text1="راسب"

GoTo C

C :

Print Text1

مربع اختيار Checkbox

تستخدم هذه الاداة من اجل اختيار واحد او مجموعة خيارات من مجموعة.

اهم خصائص هذه الاداة هي :

Alignment : تغيير هذه الخاصية اتجاه الكتابة من اليسار الى اليمين او بالعكس.

Value : هذه الخاصية تملك القيم التالية:

- 0-Unchecked : اي ان خاصية Value للاداة تحمل القيمة 0 في حالة عدم التأشير داخل المربع.

Check1.value=0 → Check1

- 1-checked : اي ان خاصية Value للاداة تحمل القيمة 1 في حالة التأشير داخل المربع.

Check1.value=1 → Check1

زر اختيار Option button

تستخدم هذه الاداة لاختيار عنصر واحد فقط من بين عدة عناصر ، وهنا على عكس Check box من المستحيل اختيار اكثر من عنصر واحد.

اهم خصائص هذه الاداة هي:

Alignment : تغيير هذه الخاصية اتجاه الكتابة من اليسار الى اليمين او بالعكس .

Value : هذه الخاصية تملك القيم التالية:

- False : اي ان خاصية Value للاداة تحمل القيمة False في حالة عدم التأشير داخل الدائرة .

Option1.value=False Option1

- True : اي ان خاصية Value للاداة تحمل القيمة True في حالة التأشير داخل الدائرة .

Option1.value=True Option1

الاداة Frame

تستخدم هذه الاداة كوعاء لباقي الادوات الاخرى وذلك من اجل فرز العمليات وتحسين المظهر ويجب ان نستخدم طريقة الرسم عند وضع الادوات عليها لأن طريقة النقر المزدوج غير فعالة.

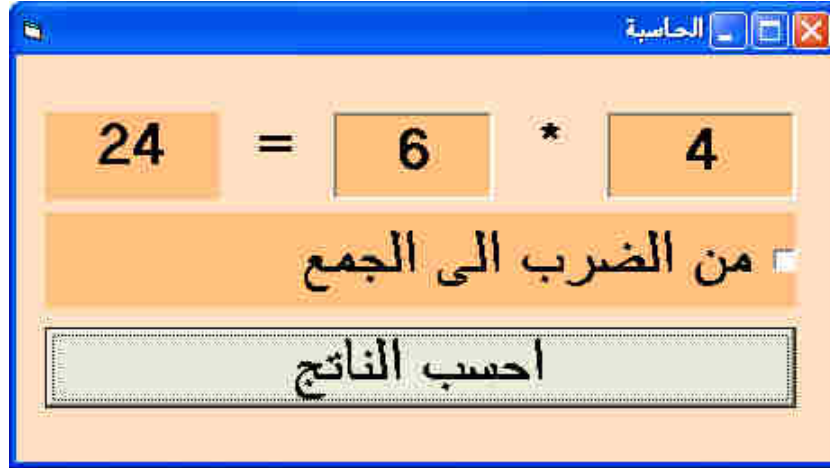
مثال ٩ : المطلوب عمل برنامج يجري عمليتي الجمع والضرب بين عددين باستخدام زر امر واحد مع اتاحة الخيار للمستخدم لتحويل العملية من الجمع الى الضرب او بالعكس.
تصميم النافذة : في هذا البرنامج سوف نحتاج الى مربعي نص Text وثلاثة ملصقات Label وزر امر Command ومربع خيار Check box ثم نضبط خصائص الادوات كما في الشكل التالي:

كتابة الكود :

```
Project1 - Form1 (Code)
Command1 Click
Private Sub Command1_Click()
If Check1.Value = 1 Then
Label1.Caption = "+"
Label3.Caption = Val(Text1) + Val(Text2)
Else
Label1.Caption = "*"
Label3.Caption = Text1 * Text2
End If
End Sub
```

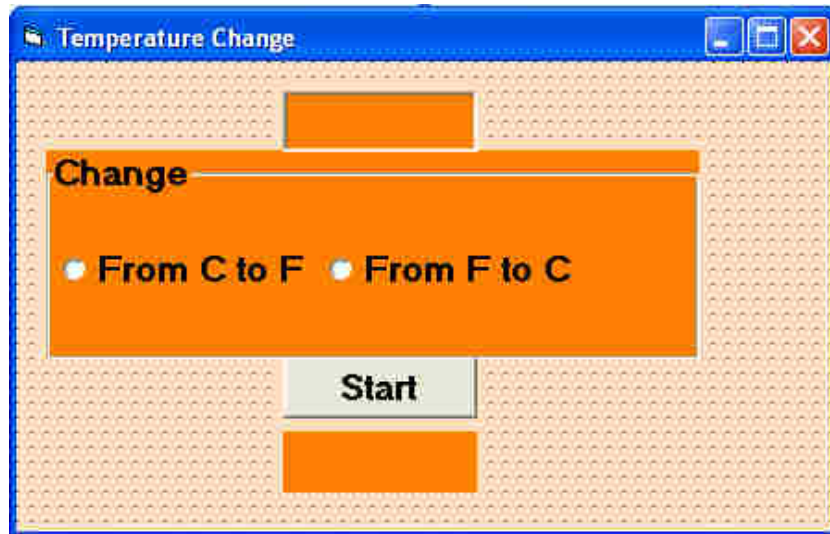
التنفيذ لعملية الجمع :

التنفيذ لعملية الضرب :

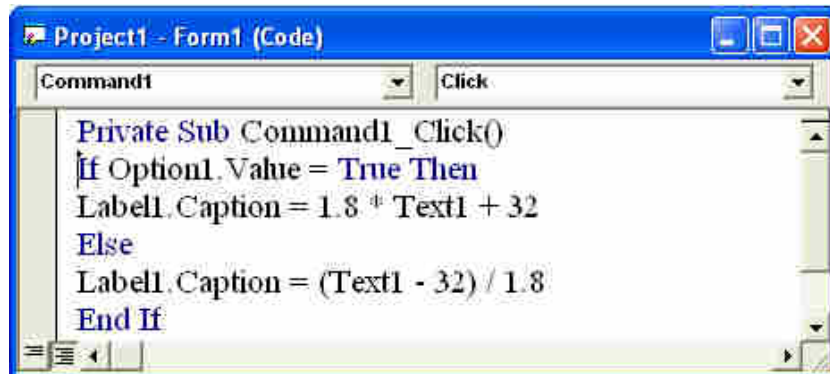


مثال ١٠ : المطلوب عمل برنامج يقوم بتحويل درجات الحرارة من المئوي الى الفهرنهايت او بالعكس.

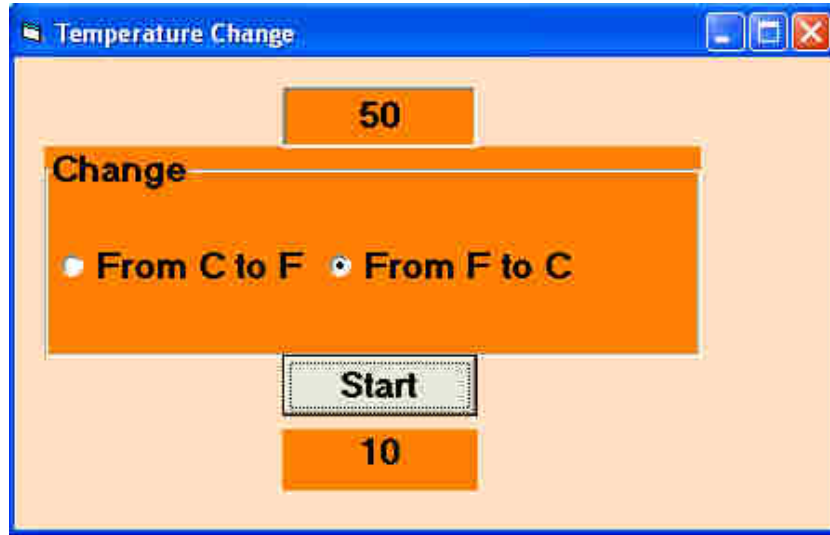
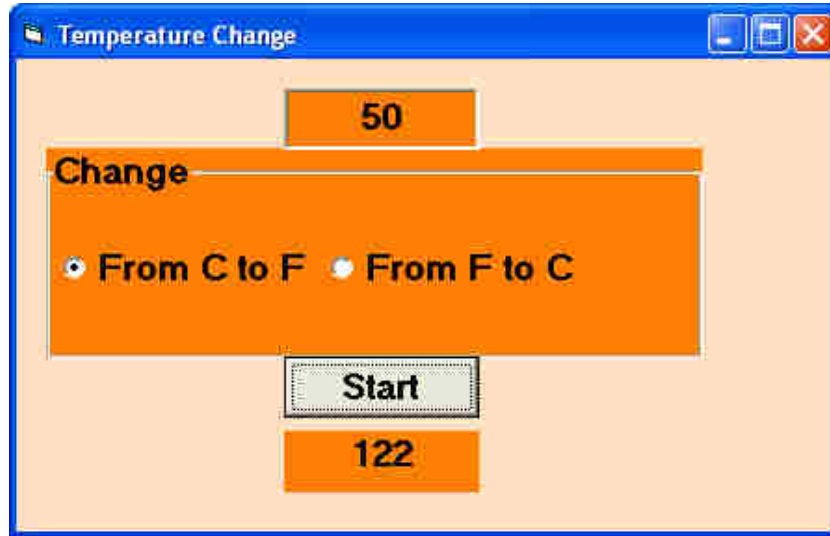
تصميم النافذة : في هذا البرنامج سوف نحتاج الى مربع نص Text وملصق واحد Label و زر امر Command و اداة الاطار Frame و زري اختيار Option ثم نضبط الخصائص للادوات وكما في الشكل التالي:



كتابة الكود :

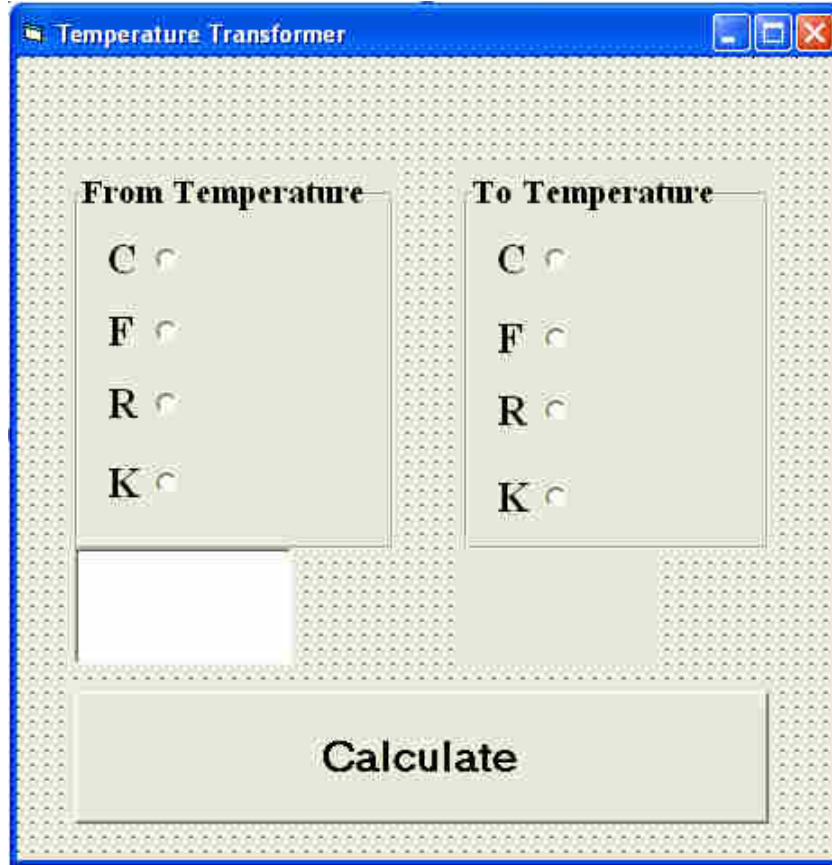


التنفيذ :



مثال ١١ : المطلوب عمل برنامج لتغيير درجة الحرارة يتيح للمستخدم اختيار واحد من C او F او K او R وتغييره الى واحد من C او F او K او R.

تصميم النافذة : في هذا البرنامج سوف نحتاج الى مربع نص Text وملصق واحد Label و زر امر Command واداتي اطار Frame و ثمانية ازرار اختيار Option حيث يجب وضع كل اربعة ازرار اختيار Option في الاطار المخصص Frame باستخدام طريقة الرسم فقط لان استخدام طريقة النقر المزدوج سوف لن يتيح عمل ازرار الاختيار بصورة صحيحة . ثم نضبط الخصائص للادوات وكما في الشكل التالي :



كتابة الكود : يجب كتابة هذا الكود في نافذة البرمجة لل Command

```
Private Sub Command1_Click()  
If Option1.Value = True And Option5.Value = True Then  
Label1 = Text1.Text * 1  
End If  
If Option1.Value = True And Option6.Value = True Then  
Label1 = Text1.Text * 1.8 + 32  
End If  
If Option1.Value = True And Option7.Value = True Then  
Label1 = Text1.Text + 460  
End If  
If Option1.Value = True And Option8.Value = True Then  
Label1 = Text1.Text + 273.15  
End If  
  
If Option2.Value = True And Option5.Value = True Then  
Label1 = (Text1.Text - 32) / 1.8  
End If  
If Option2.Value = True And Option6.Value = True Then  
Label1 = Text1.Text * 1  
End If
```

```
If Option2.Value = True And Option7.Value = True Then  
Label1 = (Text1.Text - 32) / 1.8 + 460  
End If
```

```
If Option2.Value = True And Option8.Value = True Then  
Label1 = (Text1.Text - 32) / 1.8 + 273.15  
End If
```

```
If Option3.Value = True And Option5.Value = True Then  
Label1 = Text1.Text - 460  
End If
```

```
If Option3.Value = True And Option6.Value = True Then  
Label1 = (Text1.Text - 460) * 1.8 + 32  
End If
```

```
If Option3.Value = True And Option7.Value = True Then  
Label1 = Text1.Text * 1  
End If
```

```
If Option3.Value = True And Option8.Value = True Then  
Label1 = Text1.Text - 460 + 273.15  
End If
```

```
If Option4.Value = True And Option5.Value = True Then  
Label1 = Text1.Text - 273.15  
End If
```

```
If Option4.Value = True And Option6.Value = True Then  
Label1 = (Text1.Text - 273.15) * 1.8 + 32  
End If
```

```
If Option4.Value = True And Option7.Value = True Then  
Label1 = Text1.Text - 273.15 + 460  
End If
```

```
If Option4.Value = True And Option8.Value = True Then  
Label1 = Text1.Text * 1  
End If
```

```
End Sub
```

الملاحظ في هذا الكود استخدام الامر And وذلك للفصل بين جملتي شرط والتي بصحة هاتين الجملتين فقط سوف تنفذ الاوامر داخل الجملة الشرطية.

التنفيذ :

Temperature Transformer

From Temperature

C

F

R

K

25

To Temperature

C

F

R

K

77

Calculate

Temperature Transformer

From Temperature

C

F

R

K

25

To Temperature

C

F

R

K

25

Calculate

تمرين ١ : المطلوب عمل برنامج يشابه المثال السابق ولكنه يستخدم لتغيير العملة بين الدولار والدينار واليورو بالاعتماد على ادخال ما ياتي:

- المبلغ المراد تحويله
- سعر صرف الدولار مقابل بالدينار العراقي
- سعر صرف اليورو مقابل الدينار العراقي

تمرين ٢ : المطلوب عمل برنامج لاختبار الحظ بالمواصفات التالية:

استخدام ثلاثة ملصقات يظهر في كل ملصق عدد عشوائي يتراوح بين ١ و ١٠ بالاعتماد على الامر Rnd ومفتاح امر واحد لتنفيذ البرنامج واستخدام صيغة للشرط تظهر عبارة "انت محظوظ جدا" في ملصق رابع في حالة تطابق الاعداد الثلاثة و عبارة "انت محظوظ" في حالة تطابق اي عددين فقط وعبارة "انت غير محظوظ" في حالة عدم تطابق الاعداد الثلاثة.

الرسائل

صناديق الإدخال InputBox

صناديق الإدخال إحدى أهم الأشياء في فيجوال بيسك و تستخدم صناديق الإدخال لإدخال القيم من قبل المستخدم ، وذلك عن طريق عرض صندوق حوار يحتوي سؤال ومربع نص لإدخال الإجابة وزري الأمر Ok /cancel للموافقة على الإدخال أو إلغاء الأمر.

الصيغة العامة لصندوق الإدخال InputBox تكون كما يلي :

لإدخال المتغيرات الرقمية
متغير=Inputbox("رسالة", "عنوان النافذة", "قيمة افتراضية")

لإدخال المتغيرات الحرفية
متغير\$=Inputbox\$("رسالة", "عنوان النافذة", "قيمة افتراضية")

وفيما يلي توضيح لمفردات صندوق الإدخال InputBox :

متغير : اسم المتغير لتخزين القيمة المستلمة من المستخدم .

رسالة : تعني الجملة التي يتم عرضها على المستخدم .

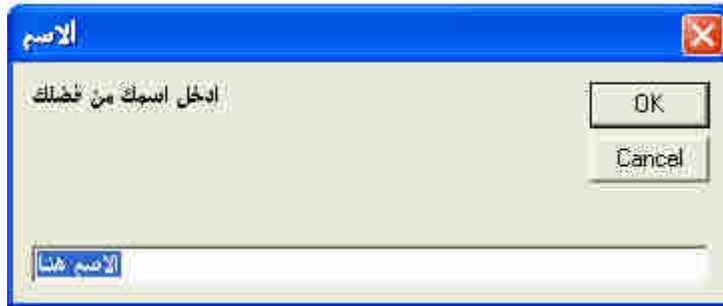
عنوان النافذة : هو شريط العنوان في النافذة التي تعرض على المستخدم .

قيمة افتراضية : هي القيمة التي يتم وضعها للمستخدم .

مثال ١ : ضع الجملة التالية في مفتاح امر command ثم نفذ البرنامج .

YourName\$ = InputBox\$("ادخل اسمك من فضلك", "الاسم", "الاسم هنا")

سيظهر الصندوق كالتالي :

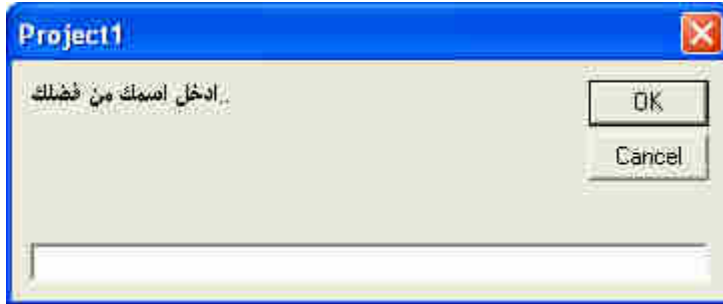


كلمة YourName\$ في بداية الكود تحمل قيمة المتغير الذي سيقوم المستخدم بإدخاله و الرمز \$ يستخدم لتمييز المتغير الحرفي عن المتغير الرقمي .

ويمكن استخدام صيغة مختصرة لصناديق الإدخال حيث يمكن الاكتفاء بالرسالة فقط وكما يلي :

YourName\$ = InputBox\$("ادخل اسمك من فضلك")

سيظهر الصندوق كالتالي:



لاحظ عزيزي الطالب عند استخدام الصيغة المختصرة لظهور صناديق الإدخال InputBox لن تكون هناك قيمة بدائية و عنوان النافذة سوف يكون نفس عنوان المشروع .

صناديق الرسائل (Message Box)

كيف تعرض رسالة على المستخدم؟

قد ترغب في أحيان كثيرة أثناء سير البرنامج ، أن تطلب من المستخدم أن يتخذ قراراً بشأن مسألة ما . فمثلاً إذا أراد المستخدم حفظ ملف باسم معين، ثم كان هذا الملف موجوداً على القرص، فقد ترغب في سؤاله إن كان يريد الكتابة فوقه أم لا . في هذه الحالة تعرض له رسالة وتترك له حرية اختيار الإجراء المناسب .

كيف تعرض هذه الرسالة؟

هناك شكلان لعرض الرسائل . الأول إذا كنت تريد عرض رسالة على المستخدم فقط ولا تطلب من المستخدم أن تحدد رأيه في مسألة معينة، في هذه الحالة استخدام الصيغة التالية :

"عنوان مربع الحوار"، الأزرار والرموز ، "الرسالة المطلوبة" MsgBox

أما إذا كنت تريد أن تعرف رأيه بشئ معين قم بوضع الأمر بالطريقة التالية

("عنوان مربع الحوار"، الأزرار والرموز ، "الرسالة المطلوبة") = MsgBox المتغير

وفي هذه الحالة سيحمل المتغير رد فعل المستخدم على الرسالة . وفيما يلي توضيح لمفردات الرسالة :

الرسالة المطلوبة : الجملة التي تريد عرضها على المستخدم .





عنوان مربع الحوار : عنوان شريط عنوان مربع الحوار .

الأزرار والرموز : يحدد نوعية الأزرار التي ترغب في عرضها .

لتحديد الأزرار التي سوف تظهر في الرسالة نستخدم الجدول التالي :

كود الأزرار	الكود البديل للأزرار	يستخدم لعرض الأزرار التالية
0	vbokonly	ok
1	vbokcancel	Ok , cancel
2	VBAbortRetryIgnore	Abort , Retry , Ignore
3	VBYesNoCancel	Yes , No , Cancel
4	VBYesNo	Yes , No
5	VBRetryCancel	Retry , cancel

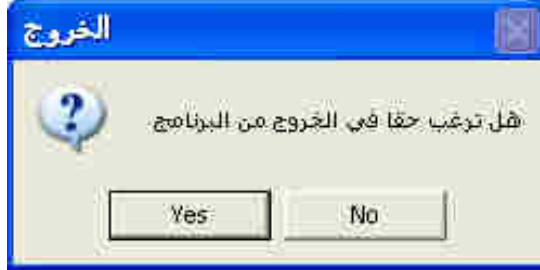
ولإظهار الرموز التي تظهر في الرسالة نستخدم الجدول التالي :

كود الرموز	الكود البديل للرموز	الاستخدام
16	VBCritical	لعرض رمز قف  و تستخدم عند حدوث خطأ
32	VBQuestion	عرض رمز علامة الاستفهام  و تستخدم عندما يطلب النظام معلومات اضافية من المستخدم
48	VBExclamation	عرض رمز علامة التعجب  و تستخدم عند حدوث خطأ ويطلب من المستخدم التدخل لتصحيحه
64	VBInformation	عرض رمز علامة الاستعلام  و تستخدم لإخبار المستخدم بمعلومة تفيد بأن الأمور تسير على ما يرام

في كلا الجدولين السابقين انت مخير في استخدام العمود الاول او الثاني لاطهار الازرار والرموز . فمثلا يمكن استخدام الكودين التاليين لاطهار الرسالة التالية:

A=msgbox (" الخروج", 36, "هل ترغب حقا في الخروج من البرنامج ")

A=msgbox (" الخروج", vbYesNo + vbQuestion, "هل ترغب حقا في الخروج من البرنامج ")



لاحظ اننا استخدمنا في الجملة الاولى الرقم 36 وهو مجموع العددين 4 لاطهار مفاتيح Yes , No و العدد 32 لاطهار رمز علامة الاستفهام . والجدول التالي يمثل قيم المتغيرات والتي تمثل رد فعل المستخدم عند الضغط على المفاتيح :

قيمة المتغير للمفتاح	المفتاح
1	Ok (موافق)
2	Cancel (إلغاء أمر)
3	Abort (توقف)
4	Retry (إعادة المحاولة)
5	Ignore (تجاهل)
6	Yes (نعم)
7	No (لا)

فمثلا في الرسالة السابقة عند الضغط على المفتاح yes تكون قيمة A هي 6 وعند الضغط على المفتاح No تكون قيمة A هي 7 .

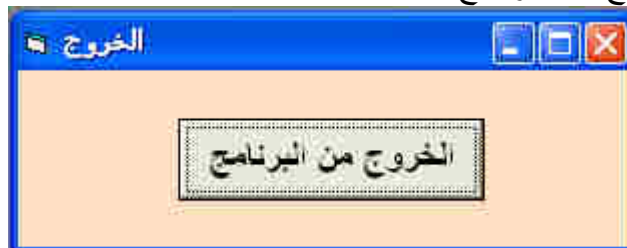
ملاحظة : كما في صناديق الادخال يمكن استخدام صيغة مختصرة للرسائل وكما يلي :

(" شكرا لاستخدامك برنامج فيجوال بيسك ") MsgBox

حيث ستظهر الرسالة التالية :



مثال ٢ : برمجة مفتاح الخروج من البرنامج .



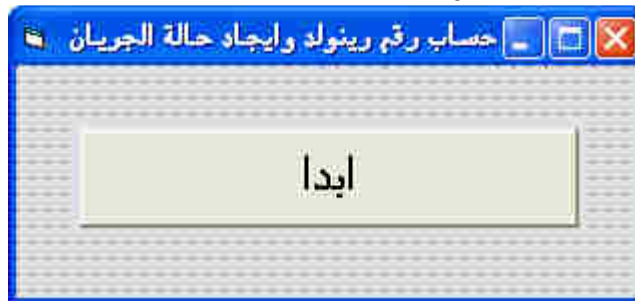
ضع الكود التالي في مفتاح امر command ثم نفذ البرنامج .


```
Project1 - Form1 (Code)
Command1 Click
Private Sub Command1_Click()
A = MsgBox("الخروج من البرنامج", 36, "هل ترغب حقا في إنهاء البرنامج")
If A = 6 Then
End
End If
endsub
```

سوف تظهر الرسالة التالية والتي من خلالها وبالاعتماد على رد فعل المستخدم سيتم الخروج من البرنامج في حالة الضغط على المفتاح Yes او الرجوع الى النافذة الاصلية في حالة الضغط على مفتاح No.



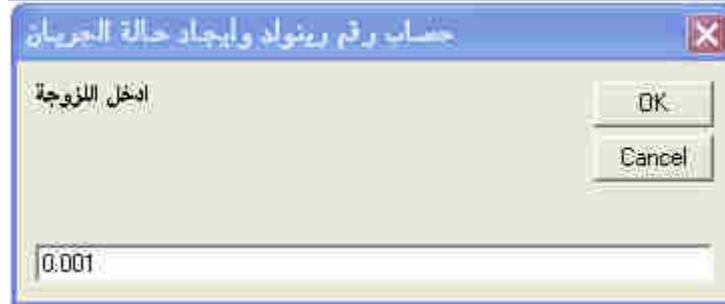
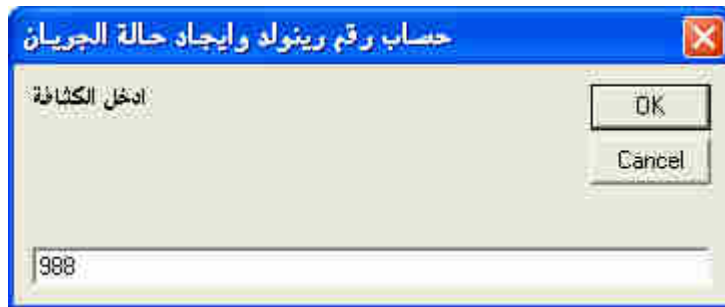

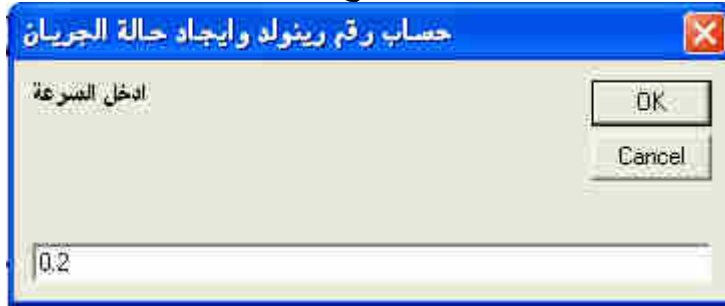
مثال ٣ : المطلوب عمل برنامج لحساب رقم رينولد بالاعتماد على ادخال قيم الكثافة والسرعة و القطر وللزوجة من خلال صناديق ادخال InputBox . حيث يعطينا البرنامج حالة الجريان باستخدام MsgBox : تصميم النافذة : نافذة بمفتاح Command واحد فقط



كتابة الكود :

```
Project1 - Form1 (Code)
Command1 Click
Private Sub Command1_Click()
U = InputBox("ادخل السرعة", "حساب رقم رينولد وايجاد حالة الجريان")
D = InputBox("ادخل قطر الأنبوب", "حساب رقم رينولد وايجاد حالة الجريان")
P = InputBox("ادخل الكثافة", "حساب رقم رينولد وايجاد حالة الجريان")
M = InputBox("ادخل اللزوجة", "حساب رقم رينولد وايجاد حالة الجريان")
Re = (U * D * P) / M
If Re < 2000 Then
MsgBox "حساب رقم رينولد وايجاد حالة الجريان", 0, "نوع الجريان هو انسيابي"
Else
MsgBox "حساب رقم رينولد وايجاد حالة الجريان", 0, "نوع الجريان هو مضطرب"
End If
```

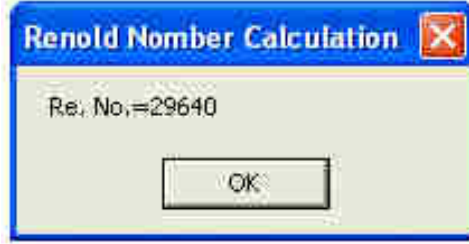
التنفيذ : عند الضغط على المفتاح ابدأ سوف تظهر صناديق الادخال وحسب التسلسل الموجود في الكود حيث على المستخدم في كل مرة ادخال قيمة والضغط على مفتاح Ok.



عندها سوف تظهر الرسالة التالية والتي تخبرنا عن نوع الجريان .



لاحظ عزيزي الطالب انه لوكان المطلوب اخراج قيمة رقم رينولد فيمكن اضافة الجملة التالية الى الكود :
MsgBox "Re. No.=" + Re\$,0,"Renold Number Calculation"
وعند التنفيذ سوف تظهر الرسالة التالية :



لاحظ اننا قمنا بتحويل قيمة Re من رقم الى نص وذلك باستخدام \$ من اجل الجمع بين النص = Re. No مع النص 29640 والتمثل بالرمز الحرفي Re\$.

مثال ٤ : المطلوب عمل برنامج لحساب مجموع ومعدل اربعة دروس مع عمل شرط في البرنامج لاجبار المستخدم برسالة عندما تكون احدى القيم اكثر من ١٠٠ حيث تتيح هذه الرسالة للمستخدم الخيار اما الخروج من البرنامج او الرجوع للنافذة الاصلية وتصفير قيم الدروس .
تصميم النافذة :



كتابة الكود :

```

Project1 - Form1 (Code)
Command2 Click
Private Sub Command1_Click()
If Text1 > 100 Or Text2 > 100 Or Text3 > 100 Or Text4 > 100 Then
A = MsgBox("الخروج من البرنامج", 36, "احدى الدرجات اكثر من 100 هل ترخت في الخروج من البرنامج")
If A = 6 Then
End
Else
Text1 = ""
Text2 = ""
Text3 = ""
Text4 = ""
Text5 = ""
Text6 = ""
End If
Else
Text5.Text = Val(Text1) + Val(Text2) + Val(Text3) + Val(Text4)
Text6.Text = Text5 / 4
End If
End Sub
Private Sub Command2_Click()
End
End Sub

```

التنفيذ ١ :

رياضيات	موالغ	برمجة	رسم
55	63	67	72
257	المجموع =		احسب
64.25	المعدل =		خروج

التنفيذ ٢ : في حالة وضع قيمة اكثر من مئة في احد مربعات النص والضغط على مفتاح (احسب) سوف تظهر الرسالة التي تخير المستخدم اما الخروج من البرنامج او الرجوع الى النافذة الاصلية وتفسير القيم.

رياضيات	موالغ	برمجة	رسم
56	67	103	75
	المجموع =		احسب
	المعدل =		خروج

سوف تظهر الرسالة التالية والتي من خلالها وبالاعتماد على رد فعل المستخدم سيتم الخروج من البرنامج في حالة الضغط على المفتاح Yes او الرجوع الى النافذة الاصلية وتفسير مربعات النص في حالة الضغط على مفتاح No .

الخروج من البرنامج

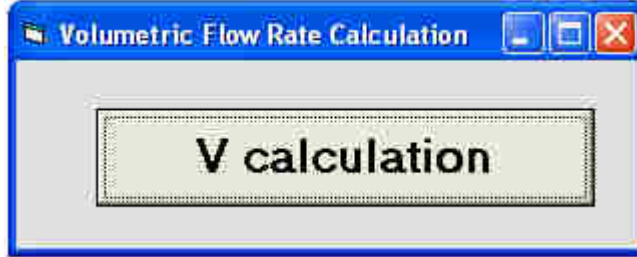
هل ترغب في الخروج من البرنامج?

Yes No

مثال ٥ : المطلوب عمل برنامج لحساب معدل الجريان الحجمي بالمواصفات التالية :

- يحتوي مفتاح الامر command على رقم سري وهو 1234 وعند ادخاله فقط سوف يعمل البرنامج.
- يتم في البرنامج ادخال قيم قطر الانبوب والسرعة من خلال صناديق ادخال InputBox . حيث يعطينا البرنامج معدل الجريان الحجمي من خلال MsgBox .

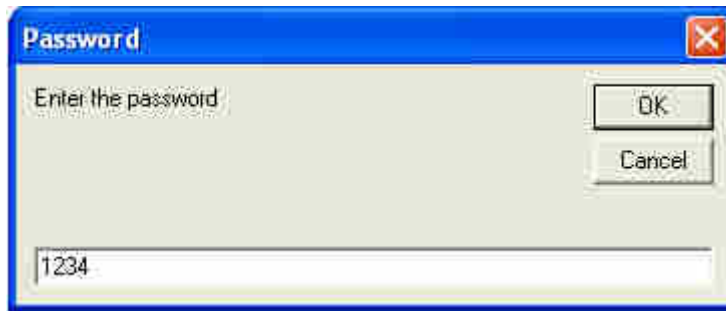
تصميم النافذة : نافذة بمفتاح Command واحد فقط .

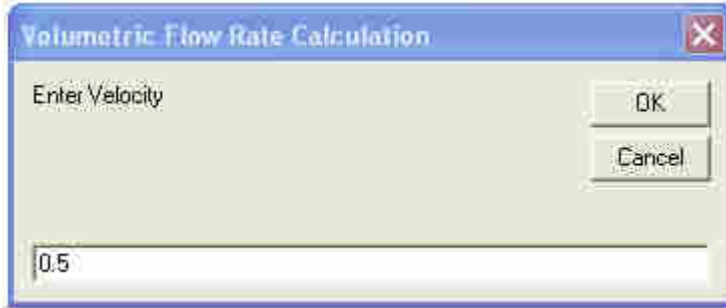
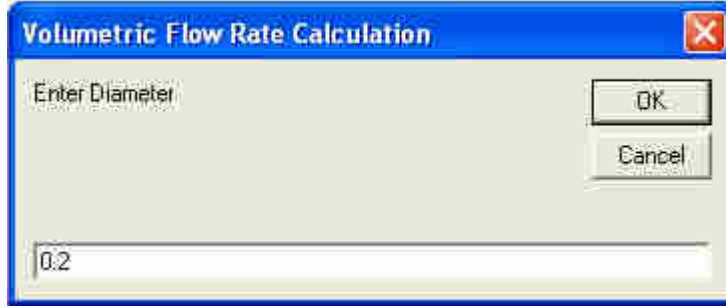


كتابة الكود :

```
Project1 - Form1 (Code)
Command1 Click
Private Sub Command1_Click()
A$ = InputBox("Enter the password", "Password", "")
If A$ = "1234" Then
D = InputBox("Enter Diameter", "Volumetric Flow Rate Calculation", "")
U = InputBox("Enter Velocity", "Volumetric Flow Rate Calculation", "")
Area = (3.14 / 4) * D ^ 2
Q$ = Area * U
MsgBox " Volumetric Flow Rate= " + Q$ + " m3 ", 0, "Volumetric Flow Rate Calculation"
Else
End
End If
```

التنفيذ : عند تشغيل البرنامج يظهر صندوق الحوار التالي والذي يطالبنا بكلمة السر وعند ادخال كلمة سر خاطئة سوف يغلق البرنامج ولاكن عند ادخال كلمة السر الصحيحة والضغط على مفتاح Ok سوف تستمر بقية صناديق الإدخال للقطر والسرعة بالظهور .





في النهاية يظهر الناتج كما في الرسالة التالية :



في الرسالة السابقة قمنا باستعمال علامة الجمع + وذلك لجمع النصوص "Volumetric Flow Rate" مع "0.0157" مع "m3/s".

تمرين ١ : المطلوب عمل كلمة سر Password قيمته AAC24BW) لنافذة البرنامج Form . حيث لن يتم تحميل البرنامج Load مالم يتم ادخال كلمة السر الصحيحة من خلال صندوق ادخال .

تمرين ٢ : المطلوب عمل برنامج لايجاد معدل دروس المرحلة الثانية في قسم الهندسة الكيماوية . استخدم صناديق ادخال لادخال درجة كل درس وصندوق رسائل لأخراج ناتج المعدل .

حلقات التكرار

التكرار يعني تكرار مجموعة من الأوامر عدد معين من المرات .

قد تحتاج إلى تكرار عملية معينة عدد كبير من المرات ، كإعادة حساب مجموعة من المتغيرات مئات المرات فان اعتمدت الطريقة اليدوية التقليدية فان ذلك سيكلفك الكثير من الجهد والوقت ، لكن في عالم البرمجة هناك مخرج من هذه المشكلة وهو بناء حلقات التكرار ، حيث هناك قاعدتين للتكرار (تستخدمان للتكرار المحدود والتكرار غير محدود) وكما يلي:

قاعدة (For – Next) : تستخدم لعدد معروف من عمليات التكرار.

الصيغة العامة : [مقدار الزيادة Step] النهائية To البداية = متغير يخزن فيه عدد مرات التكرار For
.....
.....
Next

مثال ١ : برنامج لطباعة الأرقام الفردية المحصورة بين ١ و ١٠ .

اكتب الكود التالي داخل مفتاح الأمر Command أو داخل نافذة النموذج Form ثم نفذ البرنامج.

```
For m=1 To 10 Step 2
Print m
Next
```

مثال ٢ : برنامج لإدخال رقم موجب قيمته X وحساب مجموع المتسلسلة التالية :

$$S=1 + 2 + + X$$

اكتب الكود التالي داخل مفتاح الأمر Command حيث يتم إدخال قيمة X من خلال مربع نص Text وإخراج الناتج النهائي من خلال ملصق Label.

```
X = val(text1.text)
S=0
For i=1 to X
S= S+i
Next
Label1=S
```

مثال ٣ : اكتب برنامجا لإيجاد قيمة y في المتسلسلة التالية :

$$y = x + \frac{x^2}{2} + \frac{x^3}{3} + + \frac{x^n}{n}$$

ملاحظة : استخدم مربعي نص text لإدخال قيم x و n .

تصميم النافذة :

حساب قيمة

$$y = x + \frac{x^2}{2} + \frac{x^3}{3} + \dots + \frac{x^n}{n}$$

X=

n=

Calculate Y Value

Y=

كتابة الكود :

```
Private Sub Command1_Click()  
x = Text1.Text  
n = Text2.Text  
y = 0  
For i = 1 To n  
m = (x ^ i) / i  
y = y + m  
Next  
Label5.Caption = Round(y, 2)  
End Sub
```

التنفيذ :

حساب قيمة

$$y = x + \frac{x^2}{2} + \frac{x^3}{3} + \dots + \frac{x^n}{n}$$

X= 2.5

n= 4

Calculate Y Value

Y= 20.6

مثال ٤ : المطلوب عمل برنامج لحساب الحجم المولي لمادة الميثيل كلورايد (methyl chloride) باستخدام معادلة (Redlich/Kwong) :

$$a = \frac{0.42748 \times R^2 \times T_c^{2.5}}{P_c} \quad \text{and} \quad b = \frac{0.08664 \times R \times T_c}{P_c}$$

$$V(i+1) = \left(\frac{R \times T}{P} \right) + b - \frac{(a \times (V(i) - b))}{(T^{0.5} \times P \times V(i) \times (V(i) + b))}$$

$$R = 83.14$$

$$T_c = 416.3 \text{ K}$$

$$P_c = 66.8 \text{ bar}$$

علما إن :

ملاحظة : استخدم مربعي نص لإدخال قيم درجة الحرارة T و الضغط P .

تصميم النافذة :

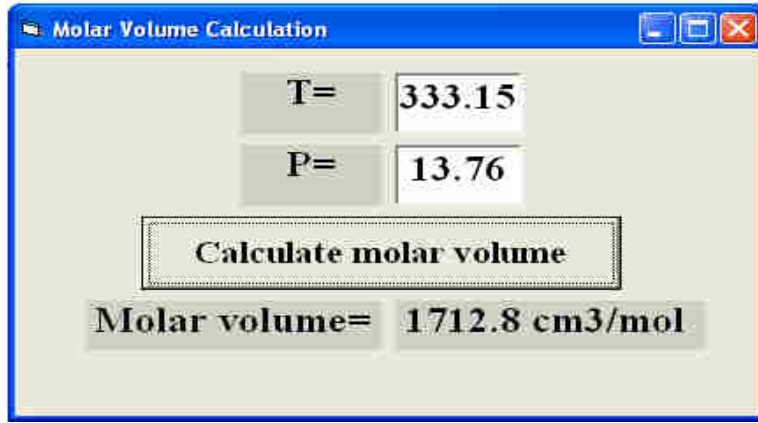
كتابة الكود :

```

Project1 - Form1 (Code)
Command1 Click
Private Sub Command1_Click()
    T = Text1
    P = Text2
    R = 83.14
    Tc = 416.3
    Pc = 66.8
    a = 0.42748 * R ^ 2 * Tc ^ 2.5 / Pc
    b = 0.08664 * R * Tc / Pc
    V = 1
    For i = 1 To 100
        V = (R * T / P) + b - (a * (V - b)) / (T ^ 0.5 * P * V * (V + b))
    Next
    Result$ = Round(V, 2)
    Label4 = Result$ + " cm3/mol "
End Sub

```

التنفيذ :



لاحظ انه عند التنفيذ ومهما كانت القيمة البدائية ل V فان القيمة النهائية الصحيحة سوف تكون واحدة.

قاعدة (Do – Loop) : تستخدم الحلقات الشرطية Do – Loop في حال عدم معرفتنا لعدد مرات التكرار المطلوبة و يتم التوقف عن التكرار عند تحقق شرط معين ولهذا النوع من الحلقات صيغتين :

الصيغة الأولى :

Do
الجملة أو مجموعة الجمل تنفذ لمرة واحدة أو طالما كان الشرط صحيحاً
Loop While الشرط

الصيغة الثانية :

Do While الشرط
الجملة أو مجموعة من الجمل تنفذ طالما كان الشرط صحيحاً
Loop

لكل صيغة من الصيغتين السابقتين تأثيرا مختلفا على سير البرنامج . ففي الصيغة الأولى يتم تنفيذ مجموعة الأوامر على الأقل لمرة واحدة سواء أتحقق الشرط أم لم يتحقق (في هذه الحالة يقوم الفيچوال بيسك بتنفيذ التكرار مرة واحدة ثم يختبر الشرط فان كان خاطئا فانه يوقف التكرار ، وان كان صحيحا فانه يقوم بتكرار مجموعة الأوامر المحصورة بين Do و Loop)

بينما في الصيغة الثانية إذا لم يتحقق شرط الدوران لن يتم تنفيذ إي أمر من الأوامر (في هذه الحالة يقوم الفيچوال بيسك بتقييم الشرط ، فان كان خاطئا فلا يقوم بتنفيذ التكرار مطلقا . إما إذا كان صحيحا ، فانه يقوم بتنفيذ مجموعة من الأوامر المحصورة بين Do و Loop ثم يعود و يختبر الشرط مرة أخرى .. وهكذا حتى يكون الشرط خاطئا فيخرج من التكرار) .

مثال ٥ : طباعة قيمة المتغير M والذي يزداد من ١ إلى ١٠ حيث يتوقف التكرار عندما تكون قيمة المتغير أكثر من ١٠ ؟ يمكن استخدام الصيغتين السابقتين لطباعة الإعداد من ١ الى ١٠ وكما يلي :

الصيغة الثانية	الصيغة الأولى
M=0 Do While M<=10 M=M+1 Print M Loop	M=0 Do M=M+1 Print M Loop While M<=10

وألان اعد كتابة الكود السابق بالشكل التالي :

الصيغة الثانية	الصيغة الأولى
M=20 Do While M<=10 M=M+1 Print M Loop	M=20 Do M=M+1 Print M Loop While M<=10

حيث سيتم تنفيذ الأوامر في الصيغة الأولى فقط لمرة واحدة إي طباعة العدد ٢١ ، بينما لن يتم تنفيذ إي من الأوامر الخاصة بالصيغة الثانية .

مثال ٦ : أكتب برنامجاً لطباعة الأعداد الزوجية بين ١٠٠ و ٢٠٠ باستخدام صيغة قاعدة (Do – Loop)

```
i = 100
Do While i < 200
Print i
i = i + 2
Loop
```

مثال ٧ : أكتب برنامجاً لحساب معدل درجات 15 طالباً؟

```
Sum = 0
i = 1
Do While i <=15
m = InputBox("ادخل درجة الطالب")
Sum = Sum + m
i=i+1
loop
avr = Sum / 15
Print avr
```

مثال ٨ : برنامج لإدخال قيم المتغير x من خلال صندوق إدخال Inputbox حيث يقوم بإعادة التكرار طالما المتغير x لا يساوي صفر .

```
x=1
Do While x<>0
x=inputbox("Enter a number")
Loop
```

تداخل الحلقات

أحيانا نحتاج لتكرار داخل تكرار بغية الحصول على المطلوب لذلك يمكننا استخدام حلقة داخل حلقة أو أكثر من ذلك وعندها يجب أن تكون الحلقة الداخلية بكاملها داخل الحلقة الخارجية أي محتواة بالكامل.

مثال ٩ : برنامج جدول الضرب .

```

→ For i = 1 To 10
→ For j = 1 To 10
  A = i * j
  Print A
Next j
Next i
    
```

الخروج من التكرار

يمكن الخروج من حلقة التكرار قبل انتهائها وذلك عند تحقق الغاية المطلوبة (أي لا حاجة للاستمرار) وذلك باستخدام الأوامر التالية بعد جملة الشرط داخل التكرار :

جملة الخروج من التكرار	اسم التكرار
Exit for	For – Next
Exit loop	Do – Loop

مثال ١٠ : المطلوب عمل برنامج لحساب درجة الغليان لأي تركيز من مزيج مكون من الايثانول والماء ، علما إن الضغط البخاري لمكونات المزيج يحسب باستخدام المعادلات التالية :

$$P_e^0 = \exp(18.5242 - 3578.91 / (T - 50.5)) \quad \text{الايثانول}$$

$$P_w^0 = \exp(18.3036 - 3816.44 / (T - 46.13)) \quad \text{الماء}$$

$$K_i = P_i^0 / P_t$$

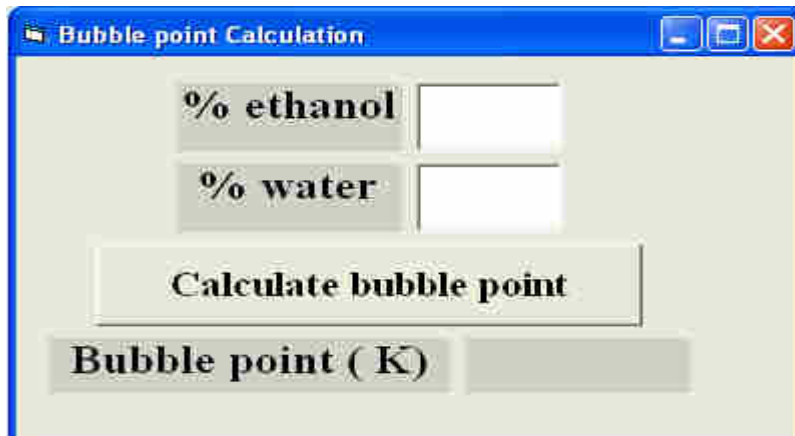
$$P_t = 760$$

$$y_i = K_i \times x_i$$

$$\sum y_i = \sum K_i \times x_i = 1 \quad (\text{عند الغليان})$$

علما أن

تصميم النافذة :



كتابة الكود :

```
Project1 - Form1 (Code)
Command1 Click
Private Sub Command1_Click()
Xe = Text1 / 100
Xw = Text2 / 100
For T = 273.15 To 450 Step 0.01
Pe = Exp(18.5242 - 3578.91 / (T - 50.5))
Pw = Exp(18.3036 - 3816.44 / (T - 46.13))
Ke = Pe / 760
Kw = Pw / 760
Sum = Ke * Xe + Kw * Xw
If Sum >= 1 Then
Exit For
End If
Next
Label4 = Round(T, 2)
End Sub
```

التنفيذ :

Bubble point Calculation

% ethanol 60

% water 40

Calculate bubble point

Bubble point (K) 358.03

تمرين ١ : أكتب الكود المستخدم لإيجاد قيمة n فى المتسلسلة التالية حتى يصبح المجموع S أقل من 0.0001 .

$$S = X - \frac{X^3}{3} + \frac{X^5}{5} - \frac{X^7}{7} \dots \dots \dots \frac{X^N}{N}$$

ملاحظة : استخدم صندوق نص لإدخال قيمة X

Tutorial 2 : Consider the following direct substitution problem, which results from an energy balance problem.

$$T_{n+1} = \frac{15.04 \times T_n}{\sqrt{(0.716 - 4.257 \times 10^{-6} T_n)}}$$

Use loop with 20 step in a program (design and code) to calculate the value of T.
Note: Use a textbox to enter the initial temperature and label to show the final temperature.

Tutorial 3 : A simple force balance on a spherical particle reaching terminal velocity in a fluid is given by;

$$V_t = \sqrt{\frac{4g(\rho_p - \rho)D_p}{3C_D\rho}}$$

Where

V_t : Terminal velocity in m/s

g : Acceleration of gravity

ρ_p : Particle density

D_p : The diameter of the spherical particle in m

C_D : Dimensionless drag coefficient.

The drag coefficient on a spherical particle at terminal velocity varies with Reynolds number (Re) as followings:

$C_D = 24/Re$ for $Re < 0.1$

$C_D = 24 * (1 + 0.14 * Re^{0.7}) / Re$ for $0.1 \leq Re < 1000$

$C_D = 0.44$ for $1000 < Re < 350000$

$C_D = 0.19 - 80000/Re$ for $350000 < Re$

Where,

$Re = (D_p v_t \rho) / \mu$ $g = 9.80665 \text{ m/s}^2$

$\rho = 994.6 \text{ kg/m}^3$ $\rho_p = 1800 \text{ kg/m}^3$

$\mu = 0.0008931 \times \text{kg/m.s.}$ $D_p = 0.000208 \text{ m}$

Calculate the terminal velocity of spherical particle using visual basic program?

البيانات DATA

البيانات التي يتم إدخالها للكمبيوتر إما أن تكون بيانات متغيرة (متغيرات) أو بيانات ثابتة (ثوابت).
فمثلا : عندما نقول إن هناك برنامج يحسب مساحة المستطيل (المساحة = الطول × العرض) فإن البرنامج في كل مرة يحسب فيها مساحة أي مستطيل سيحتاج إن ندخل له الطول والعرض ليخرج المساحة وفي كل مرة ومع اختلاف المستطيلات ستتغير قيمة الطول والعرض وبالتالي المساحة وبذلك فإننا نتعامل مع متغيرات الطول والعرض والمساحة ولو كان البرنامج يحسب مساحة الدائرة (المساحة = ط نق²) في هذه الحالة نق (نصف القطر سيختلف من دائرة لأخرى) وبالتالي فهو متغير ولكن ط = ٣.١٤ مقدار ثابت لن يتغير في كل مرة وبالتالي فإننا نتعامل مع ثابت.

أنواع المتغيرات

الجدول التالي يمثل أهم أنواع المتغيرات المستخدمة في الفيجوال بيسك

نوع المتغير	قيمة المتغير	الحجم في الذاكرة
Byte	يسمح باستخدام أرقام صحيحة بين 0 و 255	يحجز 1 بايت من الذاكرة
Integer	يسمح باستخدام أرقام صحيحة بين -32767 و 32767	يحجز 2 بايت
Long	يسمح باستخدام أرقام صحيحة بين -2147483647 إلى 2147483647	يحجز 4 بايت
Single	يسمح باستخدام أرقام كسرية وعشرية بين -3.402E+38 إلى 3.402E+38	6 بايت
Double	يسمح باستخدام أرقام كبيرة جدا تصل إلى 1E+9000	10 بايت
string	(من 0 إلى 65535) حرف	1 بايت لكل حرف
Currency	من -922337203685477.5808 إلى 922337203685477.5807	8 بايت
Date	من 1/1/100 إلى 31/12/9999	8 بايت
Variant	كل نطاق أنواع البيانات	16 بايت
Boolean	False أو True	2 بايت

ملاحظات هامة:

- عندما يتم ادخال البيانات بواسطة وحدات الادخال فانها تخزن داخل الذاكرة الداخلية للحاسوب .
- تتكون ذاكرة الحاسوب من ملايين الخلايا (الاماكن) المتساوية وتسمى هذه الخلايا بالبايت (byte) وتحمل هذه الخلايا ارقاما متسلسلة تبدأ من الصفر .
- تمثل عدد الخلايا (byte) حجم ذاكرة الحاسب وكلما زاد عدد هذه الخلايا كانت ذاكرة الجهاز اكبر وبالتالي فان سعة تخزين البيانات تكون اكبر.

فمثلا عند خزن بيان ما في الذاكرة فانه يشغل مساحة حجمها خلية واحدة (واحد بايت) وبيان اخر يشغل مساحة حجمها ٦ خلايا (٦ بايت) وهذا الاختلاف في المساحة المحجوزة للبيان يرجع الى اختلاف نوع البيان المخزن داخل هذه الخلايا . فلو اردنا تخزين العدد الصحيح ٢٠٠ فاننا يمكن تخزينه في كل من byte ,short , integer , long ولكن الاصح تخزينه من نوع byte لانه بذلك سيشغل مساحة اقل في الذاكرة .

مثال توضيحي : لو اردنا تخزين العدد ٣٢١٤٥٦.١٥٤٧٨٩٦١ داخل متغيرين من نوع single , double نجد ان القيمة التي تخزن داخل متغير من نوع single هي : ٣٢١٤٥٦.٢ والقيمة التي تخزن في متغير من نوع double هي ٣٢١٤٥٦.١٥٤٧٨٩٦١

اختيار اسم المتغير

يمكن استخدام أي اسم للمتغير بشرط تحقق ما يلي:

- يجب أن يبدأ اسم المتغير بحرف ولا يجوز أن يبدأ برقم .
- يجب ألا يزيد اسم المتغير عن ٤٠ حرفاً .
- ألا يحتوي اسم المتغير على فراغات (لأن يمكن إن يحتوي على الرمز _) .
- ألا يحتوي اسم المتغير على الرموز (* / ÷ + -) .
- ألا يحتوي اسم المتغير على كلمة من الكلمات المحجوزة لأوامر لغة الفيجوال بيسك مثل print ,copy .cls

طريقة الإعلان عن المتغيرات

- تعريف المتغيرات باستخدام الأمر Dim

الصيغة العامة :

نوع المتغير as اسم المتغير Dim

مثال ١ :

Dim name as string
Dim a as integer
Dim num as long

- تعريف المتغيرات العامة باستخدام الأمر Global

المتغير العام هو المتغير الذي يمكن استخدامه من أي مكان في البرنامج ، أي أن مداه يشمل جميع ملفات ونوافذ البرنامج . ويبقى في الذاكرة طوال فترة عمل البرنامج ولا يحذف من الذاكرة إلا بعد انتهاء البرنامج .
طريقة إدخال المتغيرات التي ترغب في جعلها متغيرات عامة هي

نوع المتغير as اسم المتغير Global

بعد هذا الإعلان ستمكن من استخدام المتغير من أي مكان داخل البرنامج وستجد أنه يحتفظ بقيمته طوال فترة

عمل البرنامج .

طريقة الإعلان عن الثوابت :

لتعريف الثوابت نستخدم الأمر Const

الصيغة العامة :

القيمة التي سوف تخزنها في هذا الثابت = اسم الثابت Const

مثال ٢ :

Const x=50

Const mark=100

المصفوفات

المصفوفة هي مجموعة من العناصر المرتبة وفق نسق معين وتحتوي على بيانات عديدة من نفس النوع .
والمصفوفة يمكن أن تكون أحادية البعد على شكل صف أو عمود ويمكن أن تكون متعددة الأسطر والأعمدة .
كل مصفوفة لها اسم واحد يمكن استخدامه للرجوع إلى أي عنصر فيها وذلك باقتران هذا الاسم بدليل يمثل مكان
العنصر في المصفوفة ، ويمكن إنشاء مصفوفة لاحتواء أي نوع من أنواع البيانات مثل النصوص والإعداد الحقيقية
والصحيحة وغيرها، فأنواع البيانات المتوفرة في الفيجوال بيسك هي : Integer , Long , Single , Byte
Variant , Boolean , Double , Currency , Data , String .
وإستخدام المصفوفات في البرمجة يساعد في كتابة برامج قصيرة وبسيطة ذات قوة كبيرة لأنه يمكن بناء حلقات
تكرار Loops تتعامل بكفاءة مع المصفوفات مهما كان عدد عناصرها .

الخصائص الأساسية للمصفوفة في الفيجوال بيسك:

- ١ . اسم المصفوفة يمثل عنوان في الذاكرة ولا يمكن تغييره إثناء تنفيذ البرنامج .
- ٢ . يمكن الإعلان عن مصفوفة لأي نوع من أنواع البيانات .
- ٣ . كل وحدة بيانات منفردة في المصفوفة تسمى عنصر .
- ٤ . جميع العناصر تكون من نفس النوع .
- ٥ . جميع العناصر تكون مخزونة على التتابع في ذاكرة الحاسوب ودليل أول عنصر هو الصفر ، ويمكن جعله
(1) باستخدام جملة Option Base .
- ٦ . يحدد موقع العنصر بواسطة رقم الصف ورقم العمود الذي يقع فيه .
- ٧ . لكل مصفوفة حد أعلى وحد أدنى ، وعناصر المصفوفة تكون محصورة بين هذين الحدين .
- ٨ . من الممكن إن تكون المصفوفة ذات بعد واحد أو متعددة الإبعاد .

أوامر الإعلان عن المصفوفات

إن أوامر الإعلان عن المصفوفات لا تختلف كثيراً عن أوامر الإعلان عن المتغيرات البسيطة التي درسناها سابقاً وهي: Public (Global), Private , Dim

- الأمر Dim
يستخدم هذا الأمر على مستوى النافذة الواحدة أو الإجراء الواحد .
- الأمر Private
يستخدم هذا الأمر على مستوى النافذة الواحدة فقط .
- الأمر Public أو Global
ويستخدم للتصريح عن المصفوفات العامة للمشروع في حالة عدة نوافذ .
يتم تعريف المصفوفات كما يلي :

نوع المتغير as (النهاية to البداية) اسم المتغير [Public (Global) , Private, Dim]

هناك نوعين من المصفوفات

1. المصفوفة ذات الحجم الثابت (Fixed- size array) : وهي التي يظل حجمها ثابت إثناء تشغيل البرنامج .
2. المصفوفة ذات الحجم المتغير (Dynamic- size array) : وهي التي يمكن تغيير حجمها إثناء تشغيل البرنامج .

المصفوفات ثابتة الحجم Fixed Size Arrays

هي المصفوفات التي يتم تحديد عدد عناصرها و تثبيت حجمها في الذاكرة من بداية المشروع . والمصفوفات من ناحية الأبعاد تنقسم أيضاً إلى نوعين وهما :

1. المصفوفات أحادية البعد (Single- Dimensional Arrays)

2. المصفوفة متعددة الأبعاد (Multi- Dimensional Arrays)

المصفوفات أحادية البعد (Single- Dimensional Arrays)

لنأخذ المصفوفة الأحادية البعد A(5) والمتمثلة بالشكل التالي :

15	20	50	30	10	40
----	----	----	----	----	----

بحيث $A(0)=15$ و $A(1)=20$ وهكذا ... $A(6)=40$.

فإذا عرفنا هذه المصفوفة داخل البرنامج، فإنها تحجز ستة عناصر، وهذا يدل على أن المصفوفة عبارة عن متغير مركب (مكون من عدة عناصر) ولكل عنصر دليل خاص به ، فإذا أخذنا العنصر الثالث مثلاً فإن له الدليل 2 ، كما أن جميع العناصر يجب أن تكون من نوع واحد وفي حالتنا هذه جميع العناصر من النوع الصحيح.

مثال 3 : إعلان عن مصفوفة Student

Dim Student (1000) As integer

وبذلك يكون عدد عناصر هذه المصفوفة 1001 عنصر . ولجعلها مصفوفة عامة ، نستبدل Dim ب Public :

Public Student (1000) As Integer

مثال ٤ : الإعلان عن مصفوفة من 51 عنصر مرقمة من 0 إلى 50 من دون تحديد نوع عناصرها ، في هذه الحالة سيعتبر النوع مباشرة Variant .

Dim Day (50)

أمثلة :

Dim name(50) as string

Dim m(10) as integer

Public A(6) as single

Dim Z(10) as Double

المصفوفة متعددة الأبعاد (Multi- Dimensional Arrays)

يقصد بكلمة بعد هو عدد الدلائل المستخدمة للرجوع إلى عنصر في المصفوفة . تستخدم المصفوفات متعددة الأبعاد لتخزين بيانات مرتبطة ببعضها البعض . لنأخذ المصفوفة $A(3,2)$ والتي تحمل اثنا عشر عنصراً موزعة على أربعة أسطر وثلاثة أعمدة وهي : $A(3,0), A(3,1), A(3,2)$; . . . ; $A(0,0), A(0,1), A(0,2)$. يتم الإعلان عن مصفوفة متعددة الأبعاد بذكر قيمة لإبعادها بعد اسمها مباشرة وداخل الأقواس ، يتم فصل قيمة كل بعد بالفاصلة ، سواء كانت ذات بعدين أو ثلاثة إبعاد كما ستوضح الأمثلة الآتية:

Public A(3,4) As Double

Public A (1 To 10 , 1 To 10) As Integer

Dim B (3, 1 To 10 , 1 To 15)

Private X(3,4) , y(9) as integer

ملاحظة:

- عدد عناصر المصفوفة متعددة الأبعاد يمكن معرفته بضرب قيم الأبعاد ببعضها .
- ينبغي الاحتراس وتقادي استخدام النوع Variant قدر الإمكان لما يتطلبه من مساحة تخزينية كبيرة .

استخدام الحلقات Loops لمعالجة المصفوفات :

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

مثال ٥ : الجمل التالية سوف تعطي قيمة لكل عنصر من عناصر المصفوفة A وهذه القيمة تكون مرتبطة بموقع العنصر داخل المصفوفة كما سنرى :

```
Public 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
```

خيارات حجز المصفوفات

١- بداية دليل المصفوفة من الصفر، وهي الحالة الافتراضية، مثال :

Dim a(4) as integer	حجز خمسة عناصر من النوع الصحيح
Dim a(3,4) as single	حجز عشرين عنصراً من النوع الحقيقي

٢- بداية دليل المصفوفة من الواحد، مثال :

Option Base 1 Dim a(4) as integer	حجز أربعة عناصر من النوع الصحيح
Option Base 1 Dim a(4,5) as single	حجز عشرين عنصراً من النوع الحقيقي

٣- بداية حجز المصفوفة اختيارياً من المبرمج، مثال :

Dim a(1 to 4) as integer	حجز خمسة عناصر من النوع الصحيح
Dim a(1 to 4,1 to 5) as single	حجز عشرين عنصراً من النوع الحقيقي

المصفوفات الديناميكية أو متغيرة الحجم Dynamic Arrays

في بعض الأحيان ، لا نعرف مسبقاً حجم المصفوفة التي سنستخدمها في البرنامج بالضبط ، وقد نريد تغيير حجم المصفوفة أثناء تشغيل البرنامج ، هنا سنحتاج إلى المصفوفات ذات الحجم المتغير حيث يمكننا تغيير حجمها في وقت لاحق Dynamic . تعتبر المصفوفات متغيرة الحجم احد مميزات فيجوال بيسك ، وهي تساعد في تنظيم الذاكرة بكفاءة فمثلاً ، يمكن استخدام مصفوفة كبيرة لوقت قصير ثم إعادة تحجيمها لتحرير مساحة من الذاكرة عندما لا نحتاجها . وهذا من شأنه تسريع المعالجة .

- جملة ReDim تستخدم في إعادة تخصيص مساحة تخزينية لمصفوفة متغيرة الحجم .
- يمكن تكرار استخدام جملة ReDim لتغيير عدد العناصر والإبعاد لمصفوفة ، ومع ذلك لا يمكن الإعلان عن مصفوفة بنوع معين من البيانات ثم إعادة تعريفها لاحقاً لنوع آخر من البيانات.
- □ إذا كان للمصفوفة بعد واحد فيمكن إعادة تحجيم هذا البعد لأنه البعد الأخير والوحيد بالمصفوفة . وإذا كان للمصفوفة بعدين أو أكثر فيمكن فقط تغيير حجم البعد الأخير مع الاحتفاظ بمحتويات المصفوفة.
- إذا صنعنا مصفوفة اصغر مما كانت فأن بيانات العناصر المخزنة سوف تفتقد . ولإعادة تعريف المصفوفة مع خزن البيانات يمكن استخدام كلمة Preserve في إعادة التعريف .

مثال ٦ : نعرف مصفوفة أحادية فارغة بالطريقة التالية:

- Dim A() as integer
-

وعند إسناد القيم إلى المصفوفة يجب علينا تحديد عدد العناصر المراد إسنادها، فمثلاً لإسناد خمس قيم نكتب:

- ReDim A(4) as integer
- A(0)=10
- A(1)=40
-
- A(4)=70

• ثم إذا أردنا إضافة قيم جديدة نقوم بالطريقة السابقة ذاتها، ومثال على ذلك، لإضافة عنصرين جديدين للمصفوفة السابقة نكتب:

- ReDim A(6) as integer
- A(5)=30
- A(6)=20

مثال ٧ : لصناعة مصفوفة M كمصفوفة متغيرة الحجم ، نعلن عنها أولاً على النحو التالي

Dim M () As Integer

ثم نعيد إعلان عنها على النحو التالي :

ReDim M(9,15)

مثال ٨ : يبين المثال التالي كيف يمكن زيادة حجم البعد الأخير للمصفوفة متغيرة الحجم بدون محو البيانات الموجودة فيها:

Dim X(10,10,6)

ReDim Preserve X(10,10,15)

ReDim Preserve X(10,10,1)

مثال ٩ : اكتب برنامجاً لقراءة أسماء الأشهر وتخزينها في مصفوفة أحادية ثم طباعتها مبينا رقم الشهر واسمه؟
البرنامج بالشكل التالي :

Private Sub Form_Load()

Dim month(12) As String

For i=1 To 12

month(i)=InputBox("أدخل اسم الشهر")

Next

For i=1 to 12

Print month (i),i

Next i

End Sub

مثال ١٠ : اكتب برنامجاً لقراءة أسماء ودرجات عشرة طلاب في مادة دراسية واحدة ، ويطبع اسم ودرجة

الطالب الحاصل على أعلى درجة.

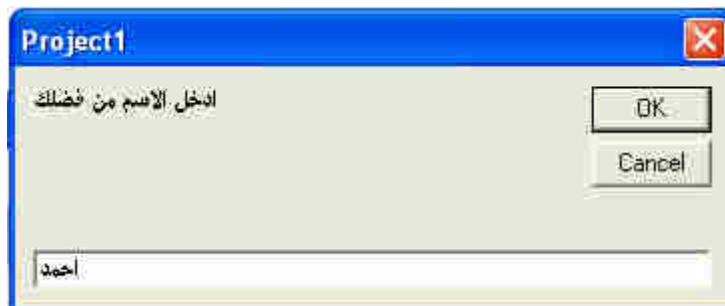
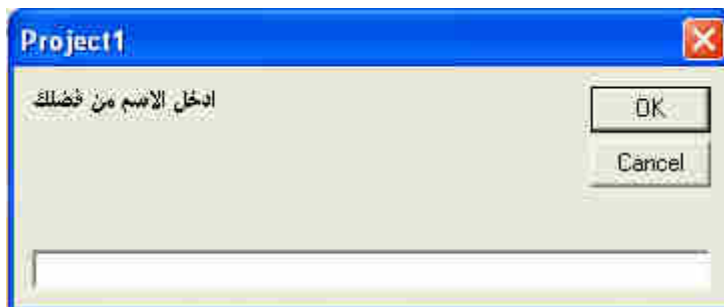
```
Private Sub Form_Load
Dim stu_name(10) As String
Dim mark(10) As Single
For i=1 To 10
Stu_name(i)=InputBox("اسم الطالب")
Mark(i)=InputBox("درجة الطالب")
Next i
Max=0
For i=1 To 10
If mark(i)>=Max Then
Max = mark(i)
Nam = stu_name(i)
End if
Print Nam, Max
End sub
```

مثال ١١ : المطلوب عمل برنامج بيانات الطلاب يحتوي على ثلاثة مصفوفات للاسم والعمر والمرحلة الدراسية حيث يمكن إظهار بيانات إي طالب عند إدخال اسم الطالب فقط؟

تصميم النافذة : نصمم النافذة التالية للبرنامج حيث نضع فيها زرین Command احدهما يقوم بتنفيذ البرنامج ، بينما الثاني للخروج من البرنامج ونضع ستة ملصقات Label ، ثلاثة للاسم والعمر والمرحلة الدراسية والثلاثة الباقية للمعلومات المقابلة .

خروج	ابدأ
	الاسم
	العمر
	المرحلة الدراسية

```
Private Sub Command1_Click()  
Dim x(4) As Integer  
Dim y(4) As String  
Dim z(4) As String  
y(1) = "فلاح"  
x(1) = 19  
z(1) = "الاول"  
y(2) = "احمد"  
x(2) = 20  
z(2) = "الثاني"  
y(3) = "طلال"  
x(3) = 22  
z(3) = "الرابع"  
y(4) = "روى"  
x(4) = 19  
z(4) = "الثاني"  
A$ = InputBox$("ادخل الاسم من فضلك")  
For i = 1 To 4  
If y(i) = A$ Then  
Label3 = y(i)  
Label5 = x(i)  
Label7 = z(i)  
End If  
Next i  
End Sub  
Private Sub Command2_Click()  
End  
End Sub
```



التنفيذ :

لاحظ عزيزي الطالب في الكود السابق إن البرنامج سوف يقارن الاسم المدخل من خلال صندوق الإدخال مع كل الأسماء الموجودة في المصفوفة الخاصة بالأسماء وعند تطابق الاسم المدخل مع الاسم الموجود في المصفوفة يتم عرض المعلومات الخاصة بالاسم المدخل .



مثال ١٢ : المطلوب عمل دليل هاتف لسبعة أشخاص ، بثلاثة مصفوفات للاسم الأول والاسم الثاني ورقم الهاتف ؟

تصميم النافذة :

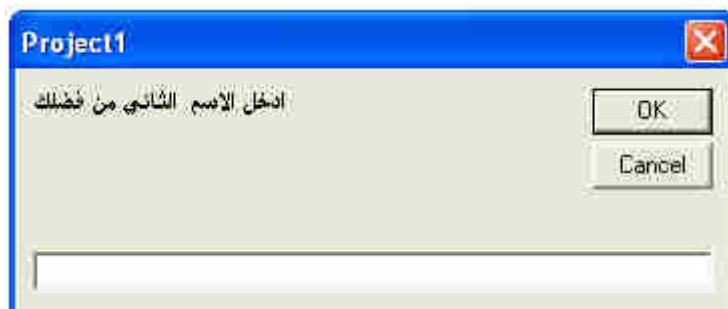
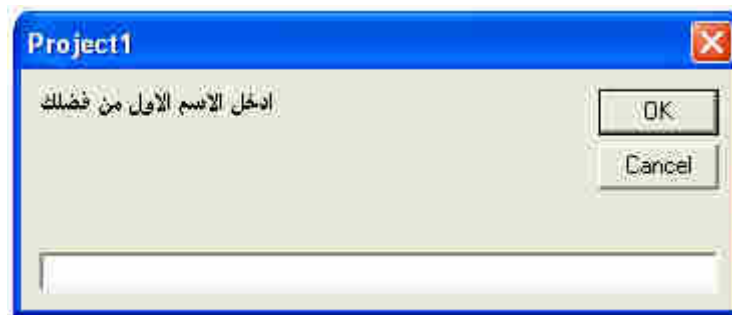


كتابة الكود :

```
Private Sub Command1_Click()  
Dim x(7) As String  
Dim y(7) As Long  
Dim z(7) As String  
x(1) = "فلاح"  
x(2) = "عمر "  
x(3) = "نور"  
x(4) = "ياسر"
```



```
x(5) = "فوزي"  
x(6) = "نمير"  
x(7) = "امل"  
y(1) = 7732442  
y(2) = 8162321  
y(3) = 4434551  
y(4) = 7167712  
y(5) = 4256929  
y(6) = 4422634  
y(7) = 7165323  
z(1) = "حسن"  
z(2) = "عقيل"  
z(3) = "ستار"  
z(4) = "ياسر"  
z(5) = "علي"  
z(6) = "احمد"  
z(7) = "محمد"  
A$ = InputBox$("ادخل الاسم الأول من فضلك")  
A1$ = InputBox$("ادخل السم الثاني من فضلك")  
For i = 1 To 7  
If A$ = x(i) Then  
If A1$ = z(i) Then  
Label3 = x(i)  
Label5 = z(i)  
Label7 = y(i)  
End If  
End If  
Next i  
End Sub
```



التنفيذ :



Tutorial 1 : Write a program (design and code) which can be used as information source for Atomic weights of chemical elements with the following list of information.

chemical elements	H	He	Na	Al	Cl	O	Cr	Fe
Atomic Weights (gm/mol)	1.008	4.003	22.990	26.982	35.453	15.999	51.996	55.847

Note: Use Inputbox to enter the chemical element symbol.

Tutorial 2 : Write a program (design and code) which can be used as information source for critical properties for several chemical components.

Component name	Formula	Tc(K)	Pc(MPa)	Zc
Methane	CH ₄	190.6	4.604	0.288
Ethane	C ₂ H ₆	305.4	4.88	0.284
Propane	C ₃ H ₈	369.8	4.249	0.281
Butane	C ₄ H ₁₀	425.2	3.797	0.274
Pentane	C ₅ H ₁₂	469.7	3.369	0.269

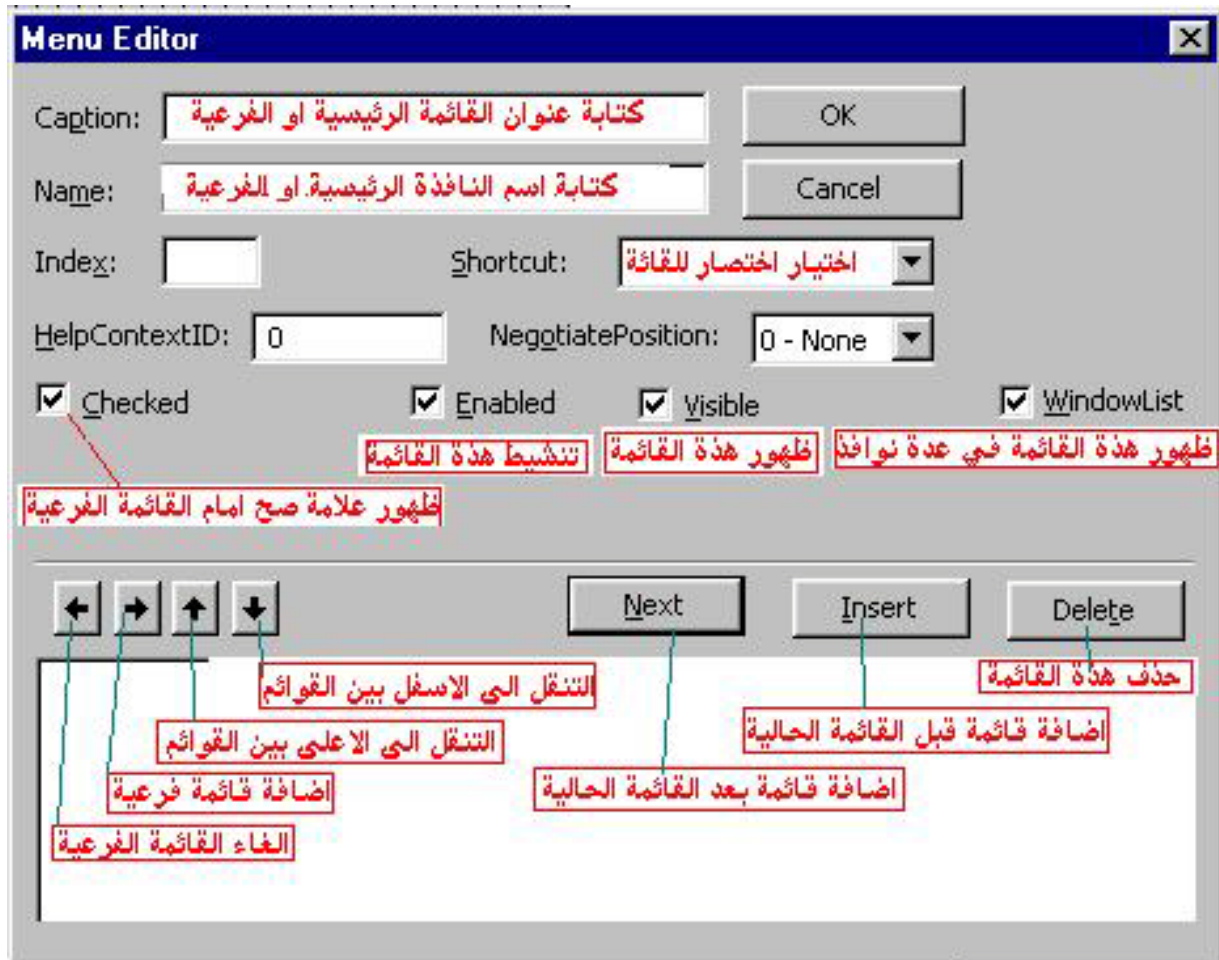
Note: Use Inputbox to enter the chemical component name.

Symbol

القوائم المنسدلة menu

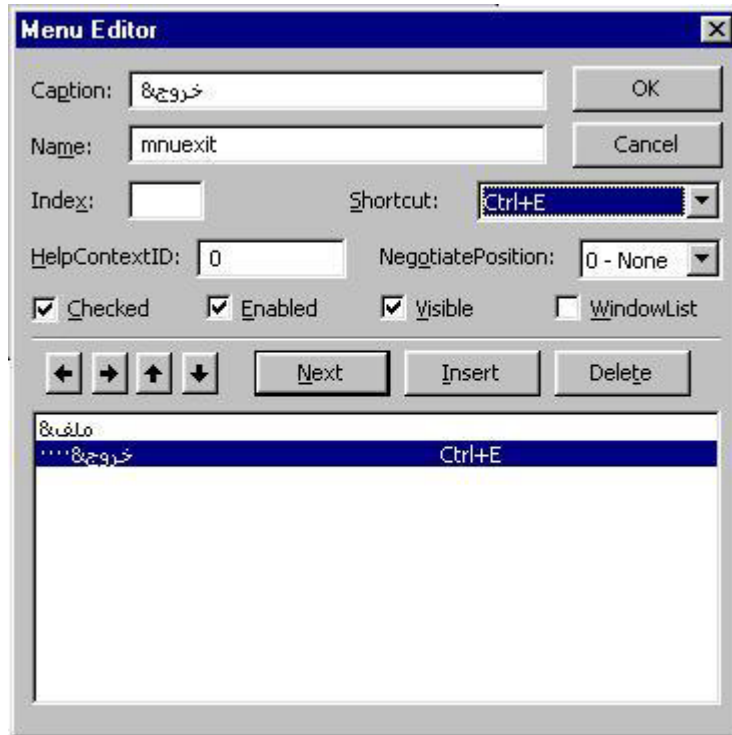
القوائم هي مجموعة الخيارات التي تجدها في معظم البرامج في أعلى الصفحة و أشهرها القائمة File . لا يمكن الاستغناء بأي حال من الأحوال عن القوائم في أي برنامج على الرغم من إمكانية التعامل من خلال أزرار الأوامر (كما تعلمنا سابقاً) إلا أن للقوائم سحرها الخاص في البرنامج لما توفره من إمكانيات ومكان ضمن البرنامج. يتم إنشاء القوائم باستخدام صندوق الحوار Menu Editor الذي يتم فيه إنشاء كل أنواع و عناصر القوائم من قوائم فرعية وخطوط فاصلة وقوائم سريعة . بينما يتم تحديد خصائص هذه القوائم من نافذة الخصائص، و على اعتبار إن كل عنصر من عناصر القائمة هو كائن مستقل فإنه يتم تحديد خصائص و برمجة كل واحد على حده.

إنشاء القوائم: يمكن فتح محرر القوائم من قائمه Tools ثم Menu Editor أو من خلال الضغط بالزر الأيمن للماوس على الفورم ثم Menu Editor سوف تظهر لك النافذة التالية:



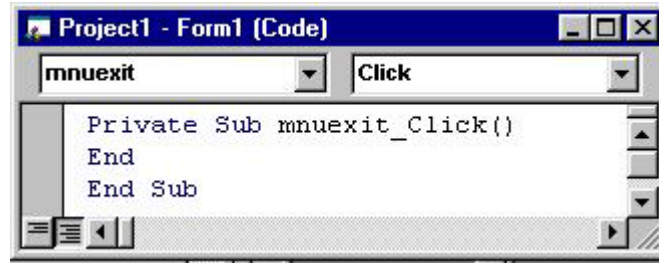
- الخاصية (Caption) : هذه الخاصية تمكنك من كتابه الاسم الذي تريده مثل فتح ، جديد ، ملف ، وغيرها من القوائم.
- الخاصية (Name) : وهي الاسم الذي سوف يظهر في صفحه الكود ويجب إن لا يتكرر ولا يكون نفس الاسم الذي في Caption وان لا يوجد مسافة بين الاسم.
- الخاصية (Shortcut) : وهي تجعلك تختار اختصارا لصفوف القائمة (مثلاً الاختصار Ctrl+s للحفظ).

- الخاصية (Checked) : وتجعلك تختار بين إن تكون عنصر من عناصر القائمة مختارة أو لا (إي يكون بجانبها علامة صح).
 - الخاصية (Enabled) : وتمكنك من إن تجعل عنصر من عناصر القائمة فعال أو لا.
 - الخاصية (Visible) : تمكنك من إن تجعل عنصر من عناصر القائمة مرئي أو لا.
 - الخاصية (Windows list) : وتمكنك من استخدام القائمة في أكثر من نموذج.
 - الأسهم الثلاثة تمكنك من التحرك بين القوائم أو داخل القوائم.
 - الزر (Next) : وهو ليجعلك تهبط للصف الأسفل.
 - الزر (Insert) : يمكنك من إضافة صف جديد في القائمة.
 - الزر (Delete) : يمكنك من مسح صف من القائمة.
- قم بكتابة ما هو موجود في الشاشة التالية



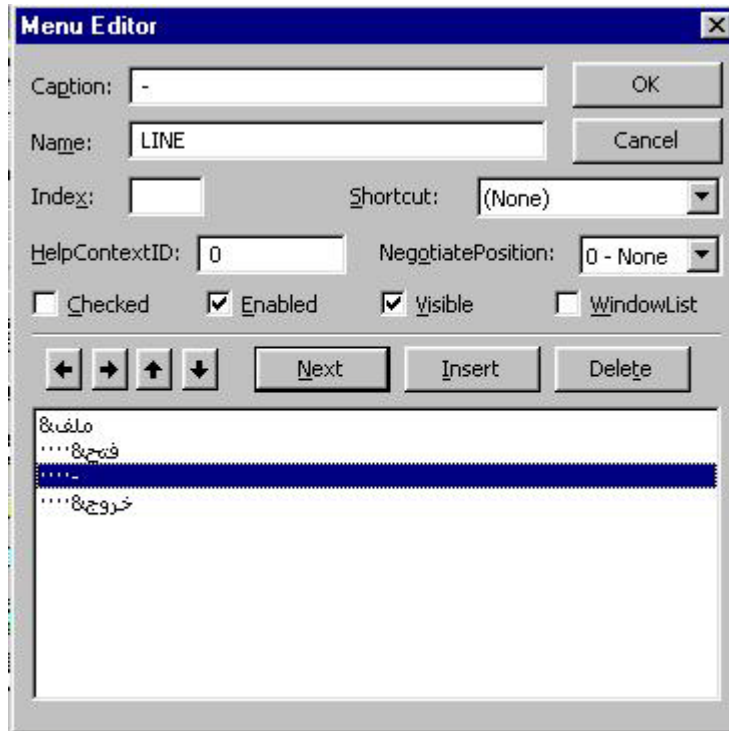
ثم نفذ البرنامج لكي يظهر لك الأتي :

كتابة الأمر في الشاشة التالية



```
Project1 - Form1 (Code)
mnuexit Click
Private Sub mnuexit_Click()
End
End Sub
```

ولإضافة خط فاصل في القوائم المنسدلة قم بإدخال الرمز (-) في الصندوق وكما يلي :



Menu Editor

Caption: - OK

Name: LINE Cancel

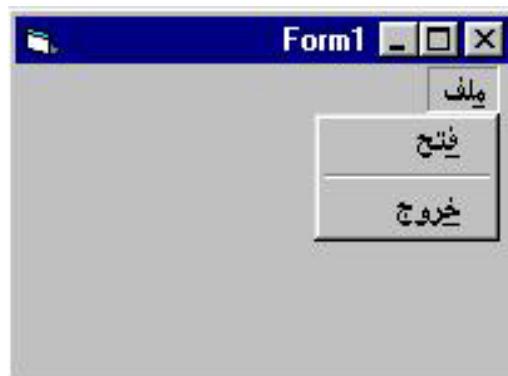
Index: Shortcut: (None)

HelpContextID: 0 NegotiatePosition: 0 - None

Checked Enabled Visible WindowList

Next Insert Delete

ملف
فتح...
...
خروج...



Form1

ملف

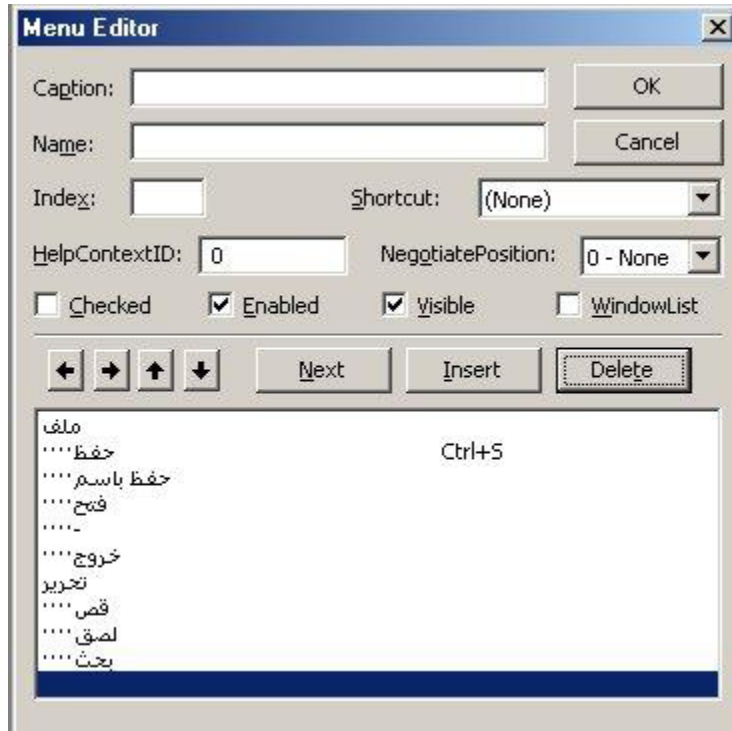
فتح

خروج

وبعد التنفيذ

مثال 1 : عمل قائمتي ملف وتحرير؟

لعمل القائمتين قم بكتابة ما موجود في الشاشة التالية :



ستظهر نتيجة المثال السابق بهذا الشكل



ملاحظة : يمكن تحويل اتجاه تنسيق النافذة إلى اليمين في البرامج العربية إما بتغيير الخاصية RightToLeft الموجودة في نافذة خصائص النافذة (Form) إلى True بكتابة الأمر الآتي في النافذة :

Form1.RightToLeft = True

بعد



قبل



أشرطة التمرير ScrollBars

أشرطة التمرير العمودية Horizontal Scroll Bars والأفقية Vertical Scroll Bars



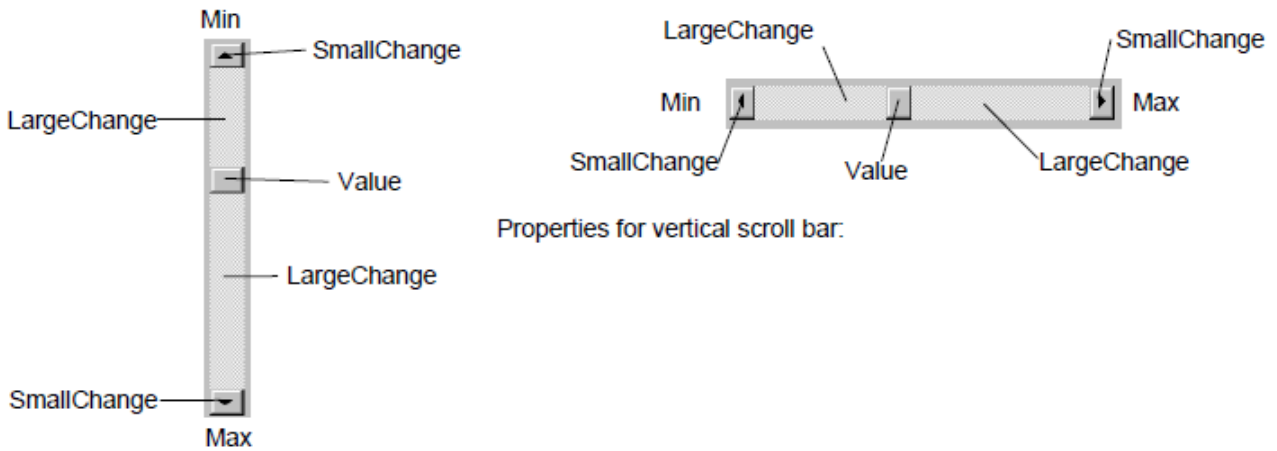
تستخدم أشرطة التمرير بكثرة في نظام النوافذ وذلك عندما تختفي بعض المعلومات في النافذة فنستخدم هذه الأشرطة لعرض باقي المعلومات أو في برامج الصوت لغرض تحديد شدة الصوت . حيث يقوم المبرمج بتحديد مدى من القيم ويقوم المستخدم باختيار أحد هذه القيم فمثلاً يمكن ربط شريط التمرير العمودي مع الأداة Picture بحيث يكون مرتبطاً بخاصية حجم الصورة فإذا حرك المستخدم الشريط لجهة فسيكبر حجم الصورة وإذا حركه لجهة أخرى فسيصغر حجم الصورة. ومن أهم خصائص أشرطة التمرير:

Max : يحدد من خلالها القيمة النهائية للشريط.

Min : يحدد من خلالها القيمة الابتدائية للشريط.

LargeChange : تمثل مقدار التغير للعمود عند الضغط Click على المنطقة الفارغة في الشريط.

SmallChange : تمثل مقدار التغير للعمود عند الضغط Click على الأسهم.



مثال 2 : برنامج مزج الألوان (الأحمر و الأخضر والأزرق)

تصميم النافذة : ضع هذه الأدوات على النافذة وغير خصائصها كما في الشكل التالي :



كتابة الكود :

```

Project1 - Form1 (Code)
HScroll2 Change
Private Sub Form_Load()
HScroll11.Min = 0
HScroll11.Max = 255
HScroll11.SmallChange = 30
HScroll11.LargeChange = 50

HScroll12.Min = 0
HScroll12.Max = 255
HScroll12.SmallChange = 30
HScroll12.LargeChange = 50

HScroll13.Min = 0
HScroll13.Max = 255
HScroll13.SmallChange = 30
HScroll13.LargeChange = 50
End Sub
Private Sub HScroll11_Change()
Picture1.BackColor = RGB(HScroll11.Value, HScroll12.Value, HScroll13.Value)
End Sub
Private Sub HScroll12_Change()
Picture1.BackColor = RGB(HScroll11.Value, HScroll12.Value, HScroll13.Value)
End Sub
Private Sub HScroll13_Change()
Picture1.BackColor = RGB(HScroll11.Value, HScroll12.Value, HScroll13.Value)
End Sub

```

الأمر RGB يقوم بمزج الألوان (الأحمر والأخضر والأزرق) بالاعتماد على شدة اللون الذي يتراوح بين 0 و 255 .

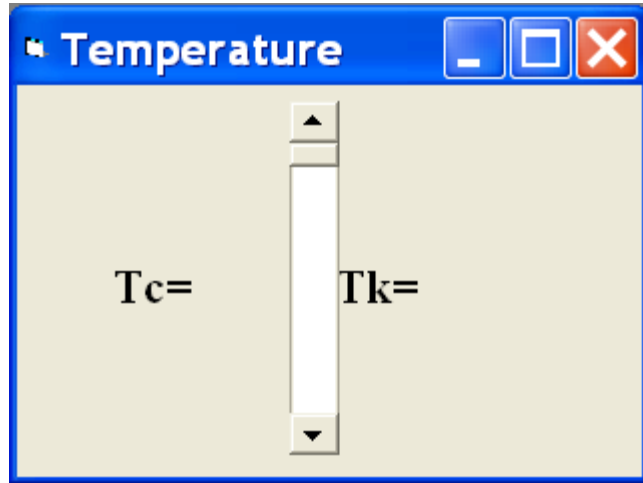
تنفيذ البرنامج : نفذ البرنامج وغير في اشروطه التمرير سوف تظهر لك الأشكال التالية :



مثال 3 : برنامج لتغيير درجة الحرارة من السيليزية (ضمن مدى من 0 إلى 100) إلى الفهرنهايتية مع تغيير لون خلفية البرنامج من الأحمر عند المائى سيليزية الى الأزرق عند الصفر سيليزية.

تصميم النافذة : ضع أربعة ملصقات وشرط تمرير عمودي على النافذة وغير خصائصها كما في الشكل التالي :

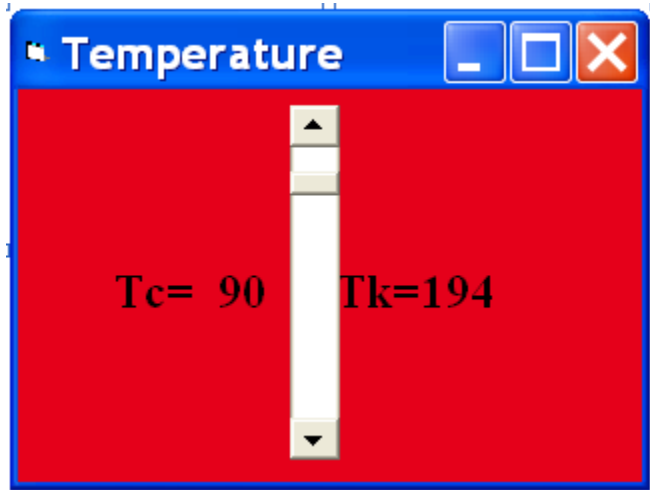
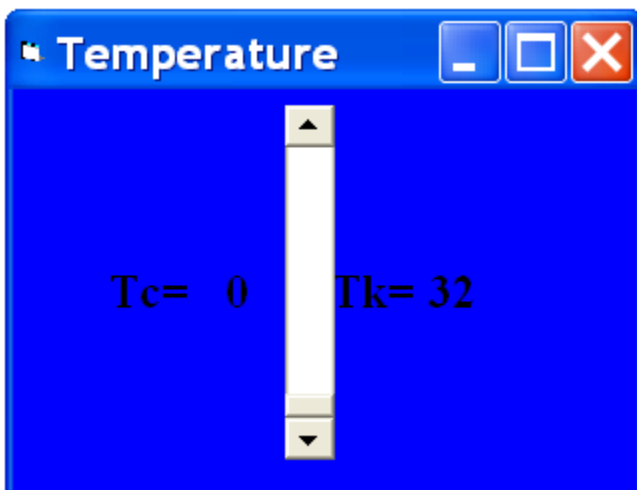
:



كتابة الكود :

```
Project1 - form1 (C...
VScroll1 Change
Private Sub Form_Load()
VScroll1.Min = 0
VScroll1.Max = 100
VScroll1.SmallChange = 5
VScroll1.LargeChange = 10
End Sub
Private Sub VScroll1_Change()
Label2 = 100 - Val(VScroll1.Value)
Label4 = Val(Label2) * 1.8 + 32
red = 2.55 * Val(Label2)
blue = 255 - red
form1.BackColor = RGB(red, 0, blue)
Label1.BackColor = RGB(red, 0, blue)
Label2.BackColor = RGB(red, 0, blue)
Label3.BackColor = RGB(red, 0, blue)
Label4.BackColor = RGB(red, 0, blue)
End Sub
```

تنفيذ البرنامج : نفذ البرنامج وغير في شريط التمرير سوف تظهر لك الأشكال التالية :



أداة القائمة ListBox

تعرض هذه الأداة مجموعة من الخيارات في داخل صندوق يحتوي على أشرطة تمرير ScrollBars فإذا قام المستخدم باختيار أحد هذه الخيارات فإنها تقوم بعمل مهمة معينة.
يتم تعبئة محتويات الأداة في وقت التصميم عن طريق الخاصية List أو وقت التنفيذ باستخدام الأمر AddItem وكما يلي:

“الاول”List1.AddItem

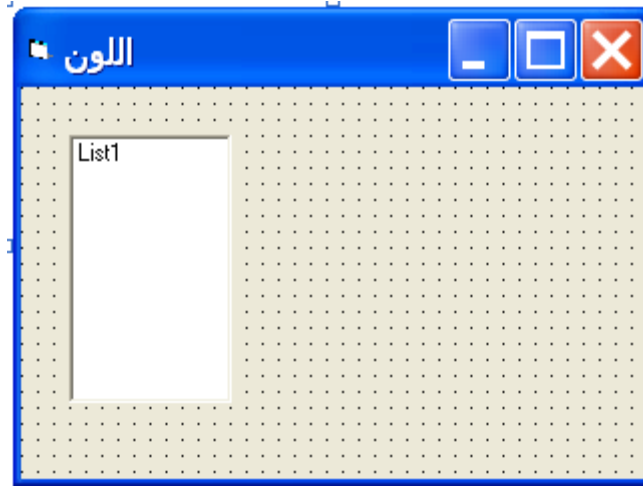
“الثاني”List1.AddItem

“الثالث”List1.AddItem

الخاصية Sorted تقوم بفرز محتويات الأداة List Box فرز تصاعدي بالاستناد على حروفها الأبجدية، مع العلم أن الحروف العربية ستكون أسفل الحروف الانجليزية.

مثال 4 : استخدم أداة ListBox لتغيير لون خلفية النموذج form بالاعتماد على الألوان (الأحمر والأخضر والأزرق).

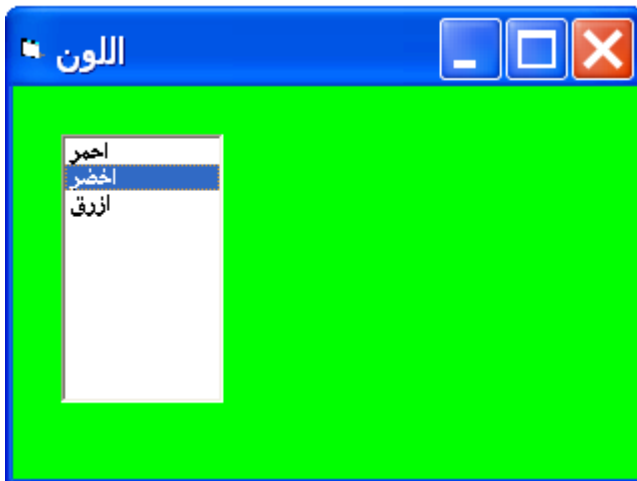
تصميم النافذة : ضع الأداة List Box على النافذة كما في الشكل التالي :



الكود :

```
Project1 - form1 (Code)
List1 Click
Private Sub Form_Load()
List1.AddItem "احمر"
List1.AddItem "اخضر"
List1.AddItem "ازرق"
End Sub
Private Sub List1_Click()
If List1.Text = "احمر" Then
form1.BackColor = RGB(255, 0, 0)
ElseIf List1.Text = "اخضر" Then
form1.BackColor = RGB(0, 255, 0)
ElseIf List1.Text = "ازرق" Then
form1.BackColor = RGB(0, 0, 255)
End If
End Sub
```

تنفيذ البرنامج : عند اختيار إي لون من الألوان الموجودة في قائمة List Box فان الحدث Click الخاص بال List Box سيقوم بتغيير لون الخلفية للنموذج.



صندوق القوائم المضغوطة Combo Box

تحتوي أداة Combo Box قائمة منسدلة تحتوي على خيارات متعددة فعند اختيار أحد الخيارات منها فإنها تقوم بتنفيذ أمر معين حسب ما يطلبه المبرمج وهي مثل الأداة List Box ولأكن تضغط هذه الخيارات بحيث تظهر صغيرة الحجم فتوفر في المساحة ولها استخدامات عديدة . معظم الخصائص للأداة ListBox السابقة موجودة في الأداة ComboBox، وذلك لان الأداة ComboBox عبارة عن أداة ListBox قياسية وتحتوي على خانة نص TextBox إضافية في أعلاها.

مثال 5 : اعد المثال السابق باستخدام الأداة ComboBox.

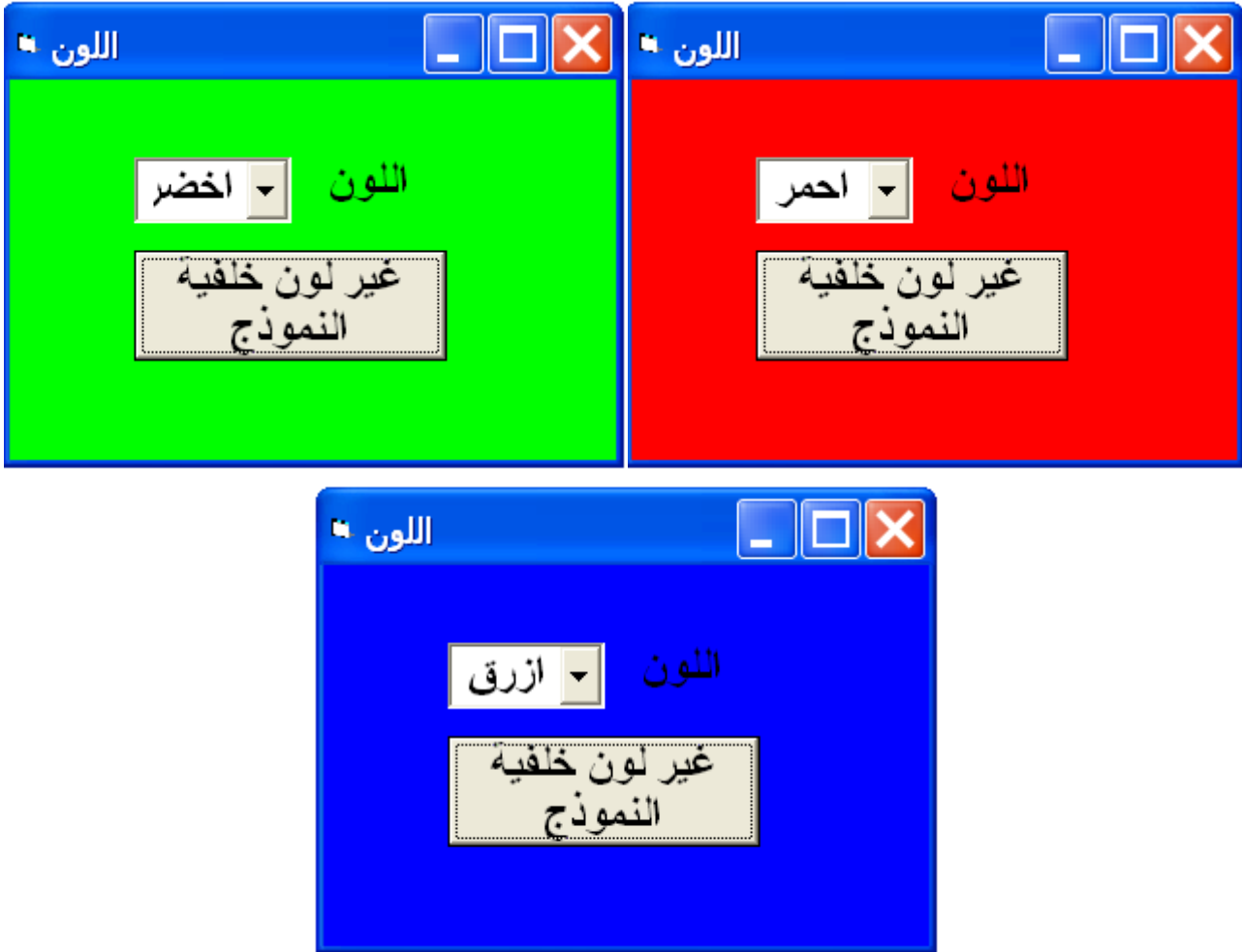
تصميم النافذة :



كتابة الكود :

```
Project1 - form1 ...
Command1 Click
Private Sub Form_Load()
    Combo1.AddItem "احمر"
    Combo1.AddItem "اخضر"
    Combo1.AddItem "ازرق"
End Sub
Private Sub Command1_Click()
    If Combo1.Text = "احمر" Then
        form1.BackColor = RGB(255, 0, 0)
        Label1.BackColor = RGB(255, 0, 0)
    ElseIf Combo1.Text = "اخضر" Then
        form1.BackColor = RGB(0, 255, 0)
        Label1.BackColor = RGB(0, 255, 0)
    ElseIf Combo1.Text = "ازرق" Then
        form1.BackColor = RGB(0, 0, 255)
        Label1.BackColor = RGB(0, 0, 255)
    End If
End Sub
```

التنفيذ :

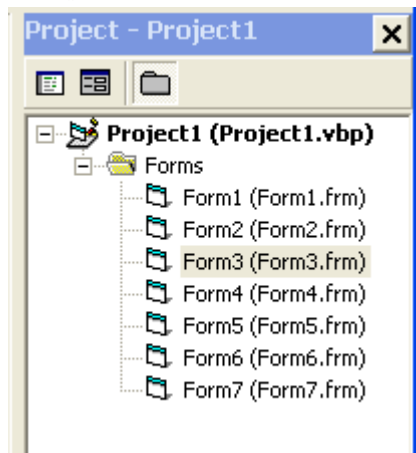


التعامل مع برنامج يحتوي على مجموعة من النوافذ

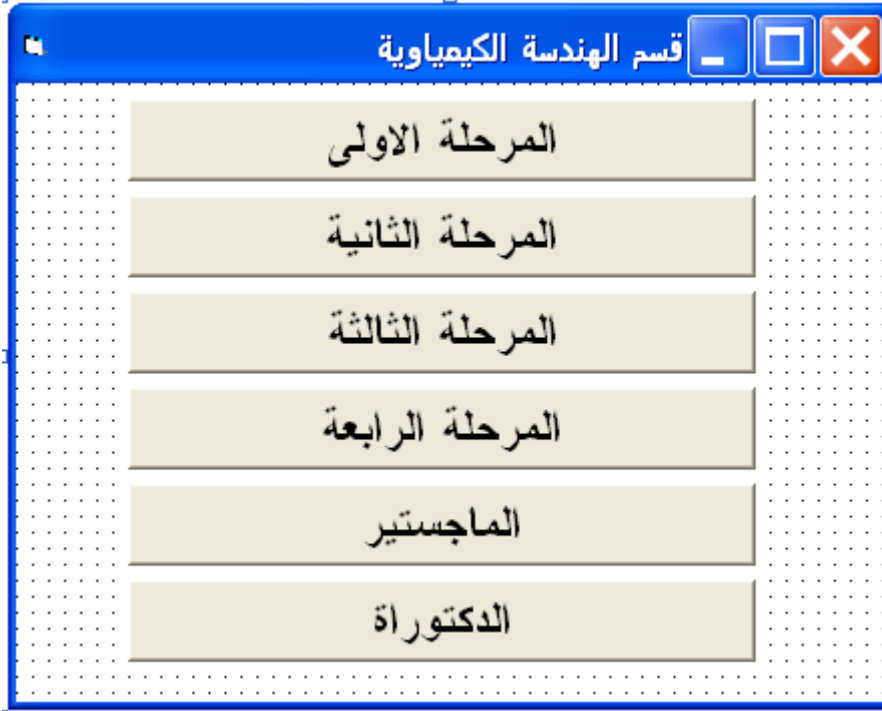
مثال 6 : المطلوب عمل برنامج خاص باللجنة الامتحانية لقسم الهندسة الكيماوية يمكنه حساب معدل ومجموع

الدروس لكل مرحلة من مراحل الدراسة في القسم من المرحلة الأولى إلى الدكتوراه.

تصميم المشروع : يحتوي المشروع على سبعة نوافذ Form وكما يلي :



تصميم النافذة الأولى : تحتوي النافذة الأولى Form1 على ستة أوامر (Command) للانتقال إلى النوافذ الستة الباقية والخاصة بالمراحل الأولى والثانية والثالثة والرابعة والماجستير والدكتوراه وكما في الشكل.



كود النافذة الأولى :

```
Project1 - Form1 (Code)
Command1 Click
Private Sub Command1_Click()
Form2.Show
Unload Me
End Sub
Private Sub Command2_Click()
Form3.Show
Unload Me
End Sub
Private Sub Command3_Click()
Form4.Show
Unload Me
End Sub
Private Sub Command4_Click()
Form5.Show
Unload Me
End Sub
Private Sub Command5_Click()
Form6.Show
Unload Me
End Sub
Private Sub Command6_Click()
Form7.Show
Unload Me
End Sub
```

الأمر Form2.show يقوم باظهار النافذة الثانية بينما الامر Unload Me يقوم بإلغاء تحميل النافذة الحالية.

تصميم النافذة الثانية : تحتوي هذه النافذة Form2 على الأدوات كما يلي :

كود النافذة الثانية :

```

Project1 - Form2 (Code)
Command2 Click
Private Sub Command1_Click()
Label10.Caption = Val(Text1) + Val(Text2) + Val(Text3) + Val(Text4) + Val(Text5) + Val(Text6) +
Label12.Caption = Label10 / 8
End Sub
Private Sub Command2_Click()
Text1 = ""
Text2 = ""
Text3 = ""
Text4 = ""
Text5 = ""
Text6 = ""
Text7 = ""
Text8 = ""
End Sub
Private Sub Command3_Click()
End
End Sub
Private Sub Command4_Click()
Form1.Show
Unload Me
End Sub

```

بنفس الطريقة يتم تصميم وضبط خصائص وكتابة الكود للنوافذ الخمسة الباقية. وكل نافذة من النوافذ الستة

الثانوية يجب إن تحتوي على أمر command يمكن الرجوع إلى النافذة الأولى وفيه الكود البرمجي التالي:

Form1.Show

Unload Me

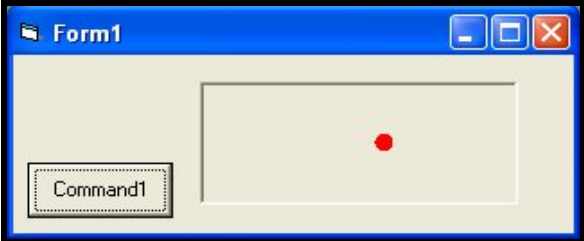
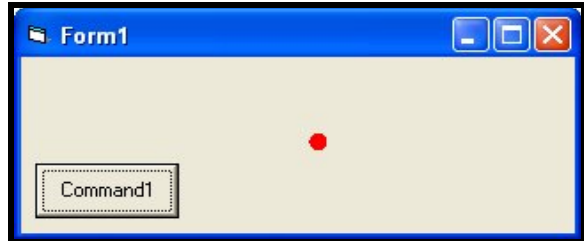
التنفيذ : يمكن عند تنفيذ البرنامج الانتقال من النافذة الأولى الانتقال إلى أي من النوافذ الستة الباقية وحساب

معدل الدروس وثم الرجوع إلى النافذة الأولى بالضغط على مفتاح الرجوع إلى النافذة الأصلية.

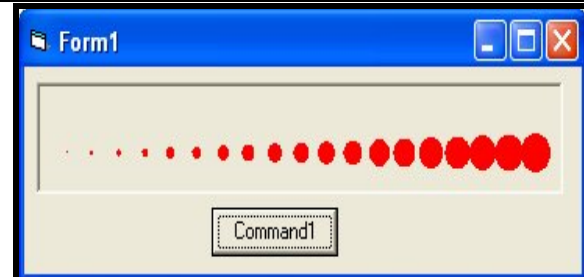
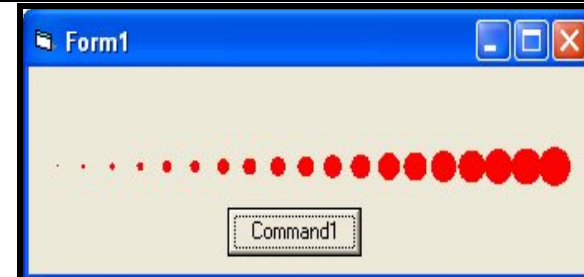
ايعازات الرسم في الفجوال بيسك :-

تمتلك الفجوال بيسك مجموعة من الطرق تمكن المستخدم من الرسم على ارضية النموذج او صندوق الصور.

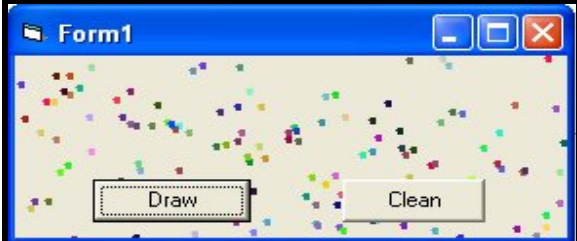
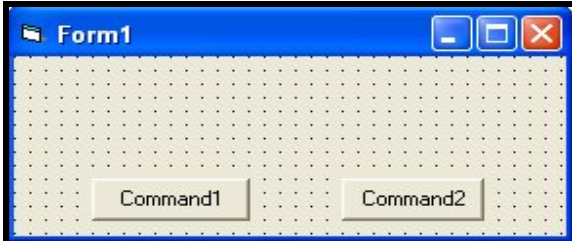
١. رسم نقطة :- تستخدم الكلمة المحجوزة Pset لتمكين المستخدم من رسم نقطة و حسب الأحداثيات التي يحددها و كما يلي :

الشكل العام لرسم النقطة	
Object Name . Pset (X,Y) , Color	
رسم نقطة على صندوق الصورة	رسم نقطة على النموذج
Picture1.Pset (X,Y) , Color	Form1.Pset (X,Y) , Color
Private Sub Command1_Click() Picture1.PSet (2500, 750), vbRed End Sub	Private Sub Command1_Click() Form1.PSet (2500, 750), vbRed End Sub
	


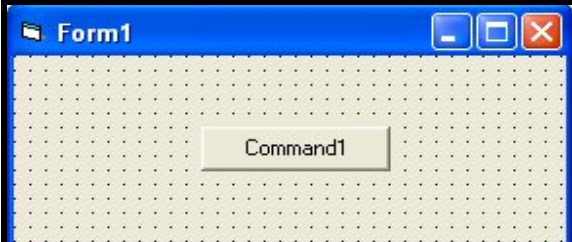
ملاحظة :- لرسم النقطة بمساحة اكبر ممكن ضبط الخاصية Drawwidth التابعة للنموذج او صندوق الصورة و كما موضح في المثال التالي:

Private Sub Command1_Click() For i = 2 To 20 x = x + 250: y = 500 Picture1.PSet (x, y), vbRed Picture1.DrawWidth = i Next End Sub	Private Sub Command1_Click() For i = 2 To 20 x = x + 250: y = 500 Form1.PSet (x, y), vbRed Form1.DrawWidth = i Next End Sub
	

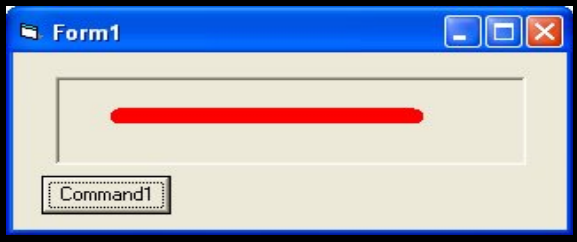

مثال :- صمم نافذة تحتوي على زري امر الأول لتمكين المستخدم من رش مجموعة من النقاط على النموذج عند الحدث click. يتم مسح النقاط المرسومة عند الحدث clic لزر الأمر الثاني؟

مرحلة التنفيذ	مرحلة التصميم
	
<pre> Private Sub Form_Load() Command1.Caption = "Draw" Command2.Caption = "Clean" Me.DrawWidth = 4 End Sub Private Sub Command1_Click() x = Form1.ScaleWidth: y = Form1.ScaleHeight Me.PSet (Rnd * x, Rnd * y), RGB(Rnd * 255, Rnd * 255, Rnd * 255) Me.PSet (Rnd * x, Rnd * y), RGB(Rnd * 255, Rnd * 255, Rnd * 255) Me.PSet (Rnd * x, Rnd * y), RGB(Rnd * 255, Rnd * 255, Rnd * 255) Me.PSet (Rnd * x, Rnd * y), RGB(Rnd * 255, Rnd * 255, Rnd * 255) End Sub Private Sub Command2_Click() Me.Cls End Sub </pre>	


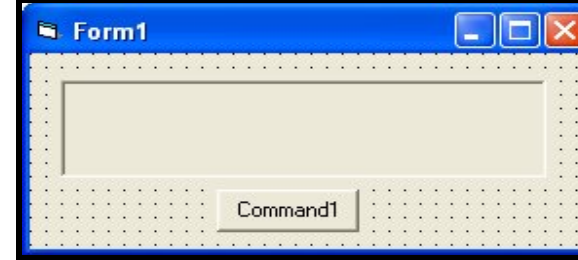
مثال :- اكتب شفرة برمجية بلغة الفجوال بيسك لرسم نقاط على شكل دائرة على النموذج؟

مرحلة التنفيذ	مرحلة التصميم
	
<pre> Dim x As Single, y As Integer Private Sub Form_Load() Me.BackColor = vbwhite : x = Me.ScaleWidth / 2: y = Me.ScaleHeight / 2 : Me.DrawWidth = 10 End Sub Private Sub Command1_Click() For i = 0 To 60 PSet (x + x * (Sin(i * 4 * Atn(1) / 30)), y + y * (Cos(i * 4 * Atn(1) / 30))), RGB(Rnd * 255, Rnd * 255, Rnd * 255) Next End Sub </pre>	

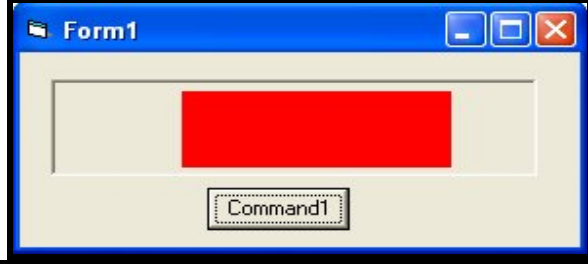

٢. رسم خط مستقيم :- تستخدم الكلمة المحجوزة **Line** لتمكين المستخدم من رسم خط و حسب احداثيات نقطتي البداية و النهاية التي يحددها و كما يلي :

الشكل العام لرسم الخط	
Object Name . Line (X,Y) – (X1,Y1) , Color , [B or BF]	
Object Name :- Form او Picture المراد رسم الخط عليه وهو اما Picture او Form	
Line :- كلمة محجوزة لرسم خط	
(X,Y):- احداثيات نقطة البداية	
(X1,Y1):- احداثيات نقطة النهاية	
Color :- لون الخط المرسم	
B :- لرسم مستطيل فارغ	
BF :- لرسم مستطيل مملوء	
لرسم نقطة على صندوق الصورة	لرسم نقطة على النموذج
Picture1 . Line (X,Y) – (X1,Y1) , Color , [B or BF]	Form1 . Line (X,Y) – (X1,Y1) , Color , [B or BF]
Private Sub Command1_Click() x = Picture1.CurrentX: y = Picture1.CurrentY Picture1.Line (x + 500, y + 350)-(x + 3000, y + 350), vbRed End Sub	Private Sub Command1_Click() x = CurrentX: y = CurrentY Me.Line (x + 500, y + 750)-(x + 4000, y + 750), vbRed End Sub
	

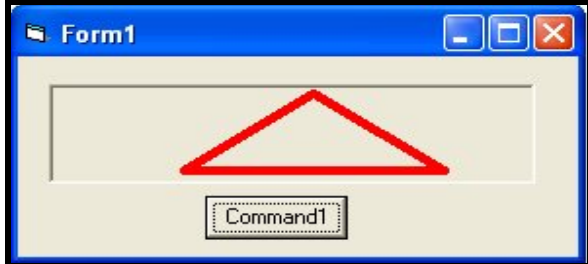
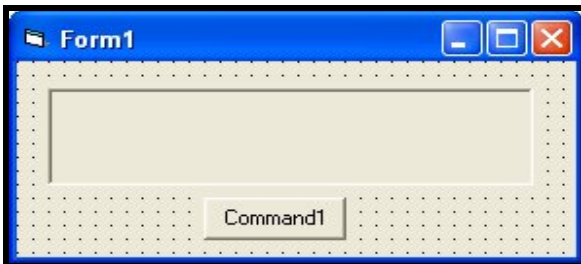
مثال :- اكتب شفرة برمجية لرسم مستطيل على صندوق صورة مستخدما الكلمة المحجوزة Line ؟

مرحلة التنفيذ	مرحلة التصميم
	
Private Sub Form_Load() Picture1.DrawWidth = 5 End Sub Private Sub Command1_Click() x = Picture1.CurrentX: y = Picture1.CurrentY Picture1.Line (x + 1000, y + 100)-(x + 3000, y + 700), vbRed, B End Sub	

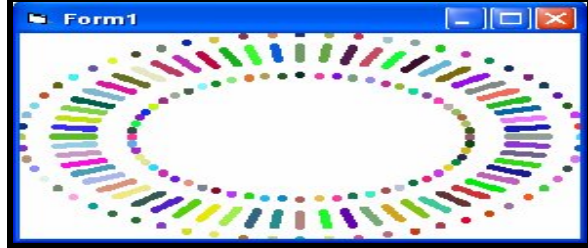
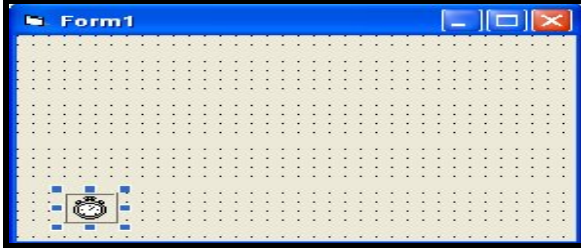
مثال :- اكتب شفرة برمجية لرسم مستطيل مملوء على صندوق صورة مستخدما الكلمة المحجوزة Line ؟

مرحلة التنفيذ	مرحلة التصميم
	
<pre>Private Sub Form_Load() Picture1.DrawWidth = 5 End Sub Private Sub Command1_Click() x = Picture1.CurrentX: y = Picture1.CurrentY Picture1.Line (x + 1000, y + 100)-(x + 3000, y + 700), vbRed, BF End Sub</pre>	

مثال :- اكتب شفرة برمجية لرسم مثلث على صندوق صورة مستخدما الكلمة المحجوزة Line ؟

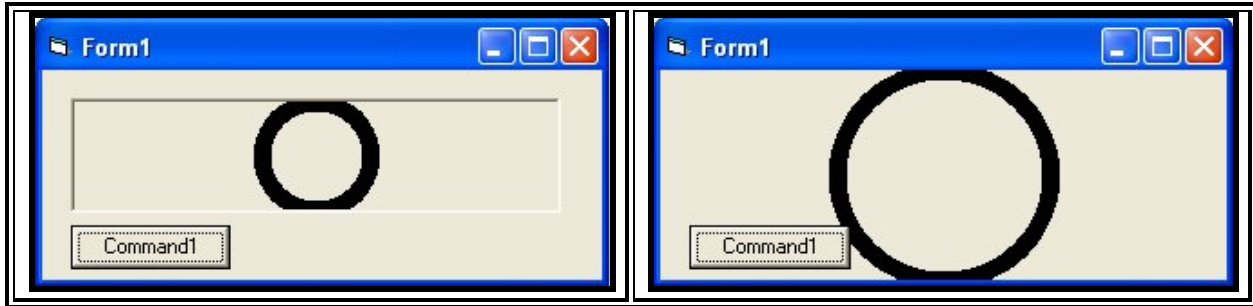
مرحلة التنفيذ	مرحلة التصميم
	
<pre>Private Sub Form_Load() Picture1.DrawWidth = 5 End Sub Private Sub Command1_Click() Picture1.Line (2000, 50)-(3000, 700), vbRed Picture1.Line (3000, 700)-(1000, 700), vbRed Picture1.Line (1000, 700)-(2000, 50), vbRed End Sub</pre>	
<p>يمكن كتابة الشفرة البرمجية بالشكل التالي</p> <pre>Private Sub Command1_Click() Picture1.Line (2000, 50)-(3000, 700), vbRed Picture1.Line (Picture1.CurrentX, Picture1.CurrentY)-(1000, 700), vbRed Picture1.Line (Picture1.CurrentX, Picture1.CurrentY)-(2000, 50), vbRed End Sub</pre>	

مثال :- اكتب شفرة برمجية لرسم خطوط مستقيمة لتشكيل دائرة على النموذج مستخدما الكلمة المحجوزة Line ؟

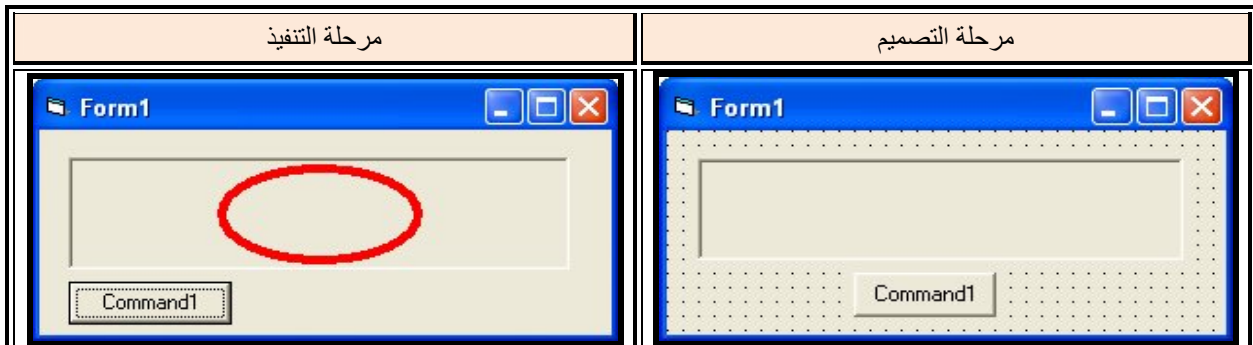
مرحلة التنفيذ	مرحلة التصميم
	
<pre> Dim x As Single, y As Integer Private Sub Form_Load() Me.Show Me.ScaleWidth = Me.ScaleHeight Me.DrawWidth = 5: x = Me.ScaleWidth / 2: y = Me.ScaleHeight / 2: Me.BackColor = vbWhite End Sub Private Sub Timer1_Timer() For i = 0 To 60 PSet (x + x * (Sin(i * 4 * Atn(1) / 30)), y + y * (Cos(i * 4 * Atn(1) / 30))), RGB(Rnd * 255, Rnd * 255, Rnd * 255) Line (x + 1600 * (Sin(i * 4 * Atn(1) / 30)), y + 1600 * (Cos(i * 4 * Atn(1) / 30)))-(x + 1900 * (Sin(i * 4 * Atn(1) / 30)), y + 1900 * (Cos(i * 4 * Atn(1) / 30))), RGB(Rnd * 255, Rnd * 255, Rnd * 255) PSet (x + 1300 * (Sin(i * 4 * Atn(1) / 30)), y + 1300 * (Cos(i * 4 * Atn(1) / 30))), RGB(Rnd * 255, Rnd * 255, Rnd * 255) Next End Sub </pre>	

٣. رسم دائرة :- تستخدم الكلمة المحجوزة Circle لتمكين المستخدم من رسم دائرة و حسب نقطة المركز ونصف القطر الدائر المراد رسمها و كما يلي :

الشكل العام لرسم الخط	
Object Name . Circle (X,Y) , Radius , Color , Start Angle , End Angle , Proportion	
اسم العنصر المراد رسم الخط عليه وهو اما Picture او Form :- Object Name	
كلمة محجوزة لرسم خط :- Circle	
احداثيات المركز :- (X,Y)	
نصف القطر :- Radius	
لون الرسم :- Color	
بداية رسم قوس من الدائرة :- Start Angle	
نهاية رسم القوس :- End Angle	
نسبة ارتفاع الدائرة الى عرضها :- Proportion	
رسم دائرة على صندوق الصورة	رسم دائرة على النموذج
<pre> Private Sub Command1_Click() x = Picture1.ScaleWidth / 2: y = Picture1.ScaleHeight / 2 Picture1.Circle (x, y), y End Sub </pre>	<pre> Private Sub Command1_Click() x = Me.ScaleWidth / 2: y = Me.ScaleHeight / 2 Me.Circle (x, y), y End Sub </pre>

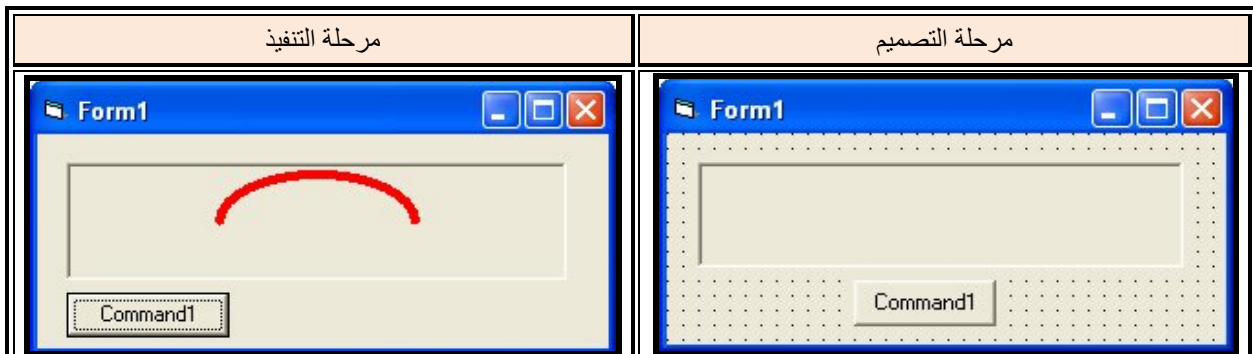


مثال :- اكتب شفرة برمجية لرسم شكل بيضوي على صندوق صورة مستخدما الكلمة المحجوزة Circle ؟




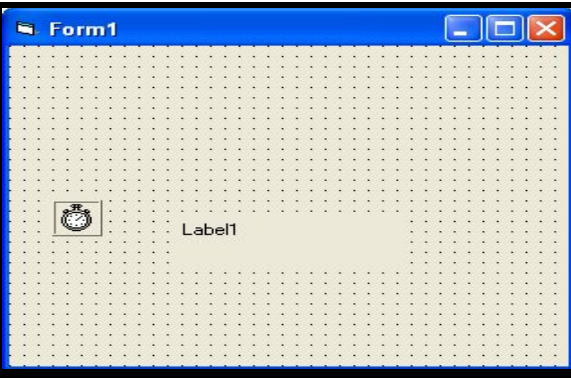
```
Private Sub Command1_Click()
Picture1.DrawWidth = 5
x = Picture1.ScaleWidth / 2: y = Picture1.ScaleHeight / 2
Picture1.Circle (x, y), 800, vbRed, , , 0.5
End Sub
```

مثال :- اكتب شفرة برمجية لرسم شكل بيضوي على صندوق صورة مستخدما الكلمة المحجوزة Circle ؟



```
Private Sub Command1_Click()
Picture1.DrawWidth = 5
x = Picture1.ScaleWidth / 2: y = Picture1.ScaleHeight / 2
Picture1.Circle (x, y), 800, vbRed, 0, 4 * Atn(1), 0.5
End Sub
```

مثال :- صمم نافذة تحتوي على اداة للعنوان الأول لعرض ساعة رقمية . ثم اكتب شفرة برمجية لعرض ساعة غير رقمية
مستعينا بادوات الرسم التي درستها؟

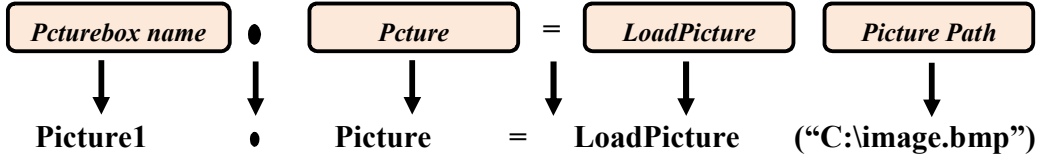
مرحلة التنفيذ	مرحلة التصميم
	
<pre> Dim x, y Private Sub Form_Load() y = Me.ScaleHeight / 2 x = Me.ScaleWidth / 2 Me.BackColor = vbBlack Label1.BackColor = vbBlack: Label1.ForeColor = vbGreen Label1.FontSize = 14: Label1.FontBold = True End Sub Private Sub Timer1_Timer() Label1.Caption = Time Me.Cls DrawWidth = 6: Me.Line (x, y)-(x + (900 * Sin(Hour(Now) * 3.1415 / 6)), y - (900 * Cos(Hour(Now) * 3.1415 / 6))), vbRed DrawWidth = 6: Me.Line (x, y)-(x + (1200 * Sin(Minute(Now) * 3.1415 / 30)), y - (1200 * Cos(Minute(Now) * 3.1415 / 30))), vbRed DrawWidth = 1: Me.Line (x, y)-(x + (1500 * Sin(Second(Now) * 3.1415 / 30)), y - (1500 * Cos(Second(Now) * 3.1415 / 30))), vbRed Me.DrawWidth = 5 For i = 0 To 60 Line (x + 1500 * (Sin(i * 4 * Atn(1) / 30)), y + 1500 * (Cos(i * 4 * Atn(1) / 30)))-(x + 1600 * (Sin(i * 4 * Atn(1) / 30)), y + 1600 * (Cos(i * 4 * Atn(1) / 30))), vbGreen Next End Sub </pre>	

صندوق الصورة (Picture Box) :

تستخدم هذه الأداة لعرض الصور عن طريق الخاصية Picture وكذلك يمكن استخدامها كحاوية لباقي عناصر التحكم.

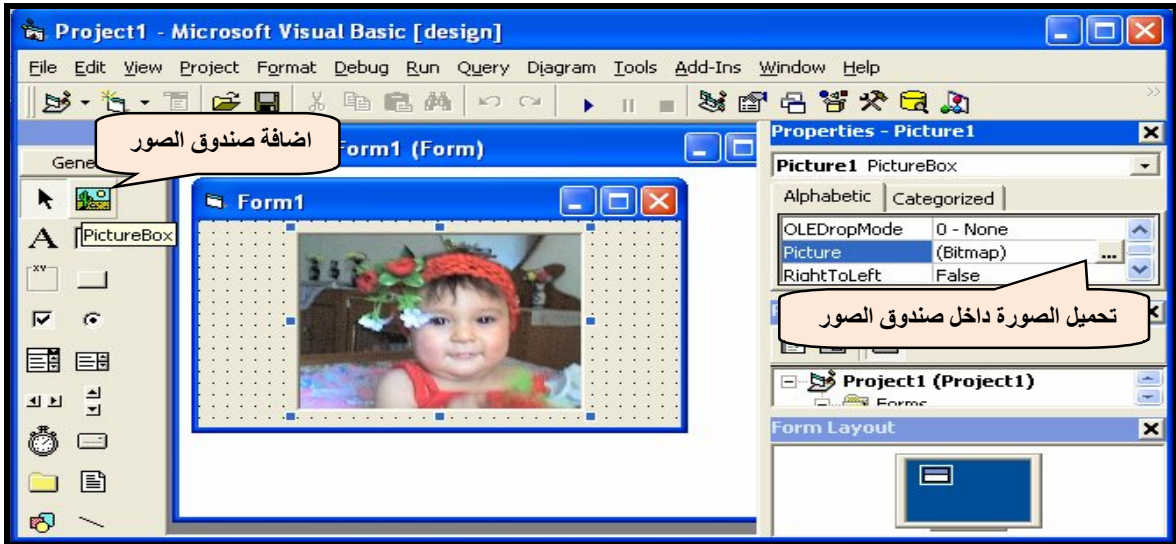
أهم الخصائص

- إضافة صورة جديدة:-- يتم ذلك من خلال استعمال الجملة البرمجية التالية:



يؤدي كتابة هذه الشيفرة الى تحميل الصورة الموجودة في المسار المحدد الى صندوق الصورة الذي اسمه Picture1 .

لاضافة الصورة اثناء مرحلة التصميم تتبع الخطوات المبينة في الشكل التالي



- الخاصية **AutoSize** : تاخذ هذه الخاصية القيمة المنطقية (True or False)

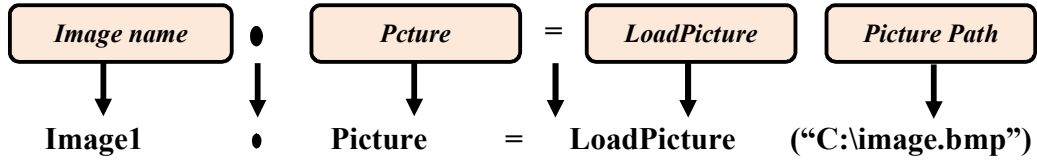


اداة الصورة (Image) :

تستخدم هذه الأداة لعرض الصور عن طريق الخاصية Picture ولا يمكن استخدامها كحاوية لباقي عناصر التحكم.

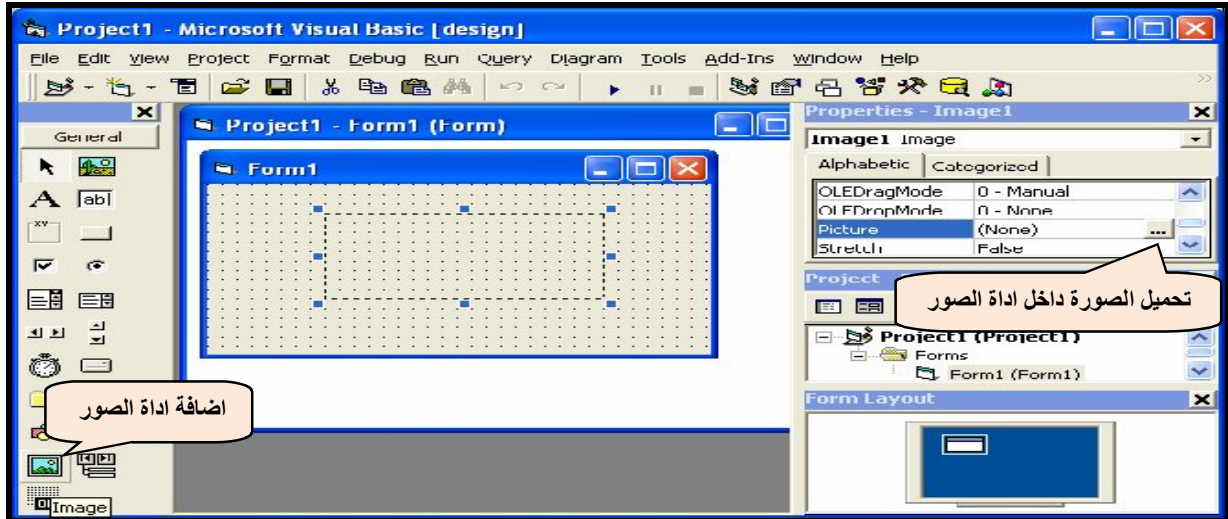
أهم الخصائص

- اضافة صورة جديدة:- يتم ذلك من خلال استعمال الجملة البرمجية التالية:

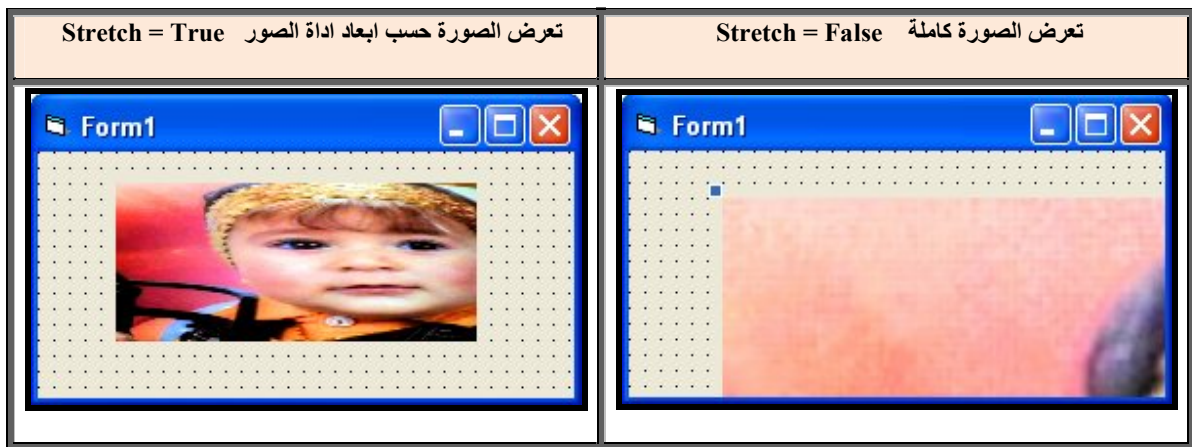


يؤدي كتابة هذه الشيفرة الى تحميل الصورة الموجودة في المسار المحدد الى اداة الصورة الذي اسمه Image1 .

لاضافة الصورة اثناء مرحلة التصميم نتبع الخطوات المبينة في الشكل التالي

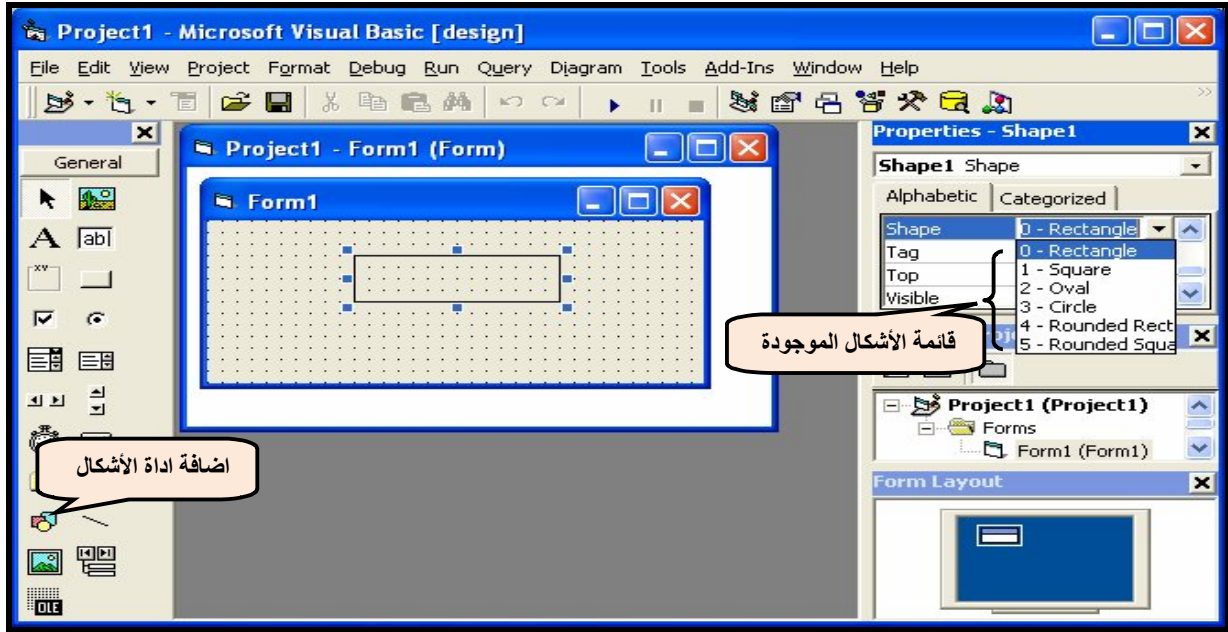


- الخاصية Stretch: تأخذ هذه الخاصية القيمة المنطقية (True or False)



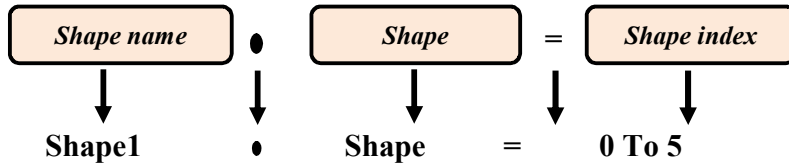
أداة الأشكال (Shape) :







تستخدم هذه الأداة لعرض عدد من الأشكال الهندسية عن طريق الخاصية Shape.



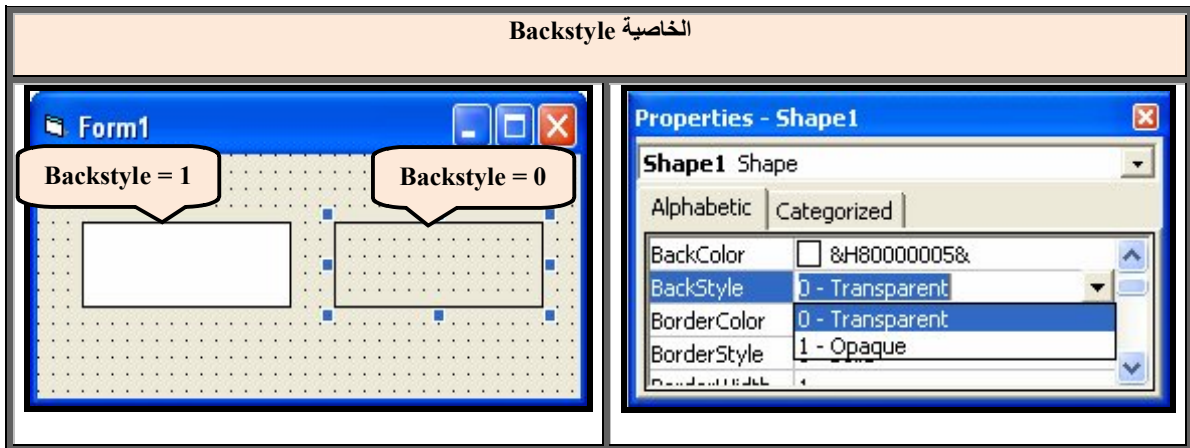
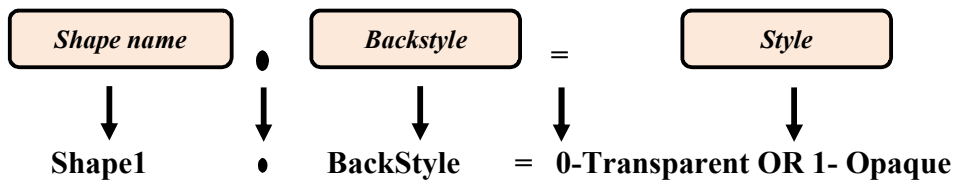
أهم الخصائص

- الخاصية **Shape**: والتي يمكن تغييرها في مرحلة التصميم أو التنفيذ و يتم ذلك من خلال استعمال الجملة البرمجية التالية:



Shape1.Shape = 0	Rectangle	
Shape1.Shape = 1	Square	
Shape1.Shape = 2	Ovel	
Shape1.Shape = 3	Circle	
Shape1.Shape = 4	Rounded Rectangle	
Shape1.Shape = 5	Rounded Square	

- **الخاصية Backstyle**: تحدد هذه الخاصية نمط خلفية الشكل شفافة او غير شفافة.



- **الخاصية Backcolor**: تعطي هذه الخاصية اللون الخلفي للشكل. (يجب ان تكون Backstyle = 1)

