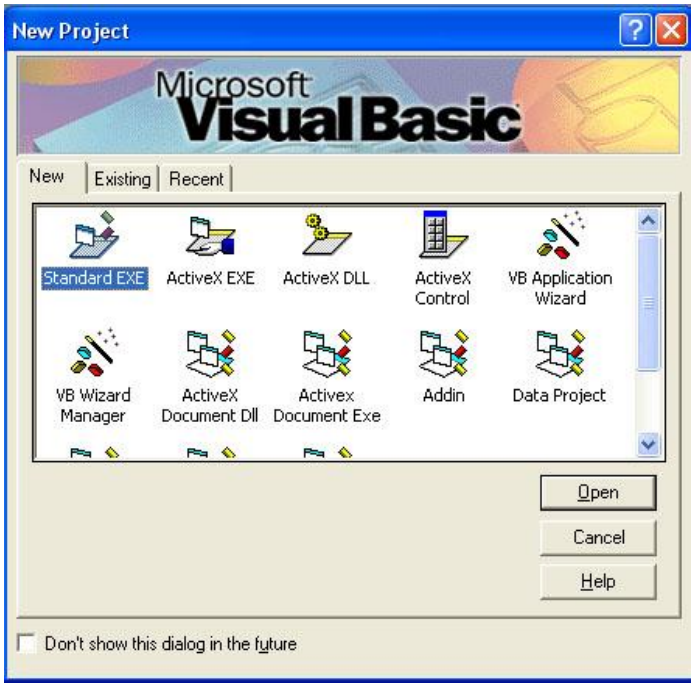
ملخص لماده

مهارات الحاسوب لطلبة التخطيط والإداره

Microsoft Visual Basic 6

Written by: Mohammad Rasheid



Copyrights 2012 ©Mohammad Rasheid

INDEX

CHAPTER ONE

الملحق الأول

CHAPTER TWO

CHAPTER THREE

CHAPTER FOUR

CHAPTER FIVE

CHAPTER SIX

WRITTEN BY

Copyrights 2012 ©Mohammad Rasheid

visual basic 6

chapter one

INDEX

Copyrights 2012 ©Mohammad Rasheid

Inevitable you need to know about Visual Basic

1- **Visual Basic is a tool that allows you to develop Windows (Graphic User Interface - GUI) applications.**
2- **Visual Basic is an event-driven.**
3- **Visual Basic is governed by an event processor.**
4- **All Windows applications are event-driven. For example, nothing happens in Word until you click on a button, select a menu option, or type some text. Each of these actions is an event.**

Event-driven: meaning code remains idle until called upon to respond to some event (button pressing, menu selection)

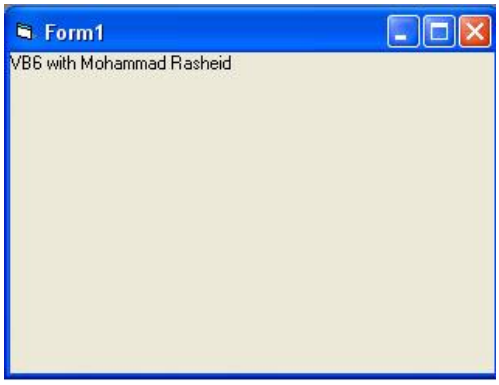
Event processor: Nothing happens until an event is detected. Once an event is detected, the code corresponding to that event (event procedure) is executed. Program control is then returned to the event processor

Basic Programming Constructs

Statements

Statements are special keywords in a programming language that do something when executed. For example, the Print statement in VB prints an expression on the screen.

Print "VB6 with Mohammad Rasheid"



Variables and Data in Visual Basic

Programs use data in writing the program code, these data are specified (محدده) as variables, where variables are memory locations storing the values of these variables.

Copyrights 2012 ©Mohammad Rasheid

تعريف المتغيرات (Declaring Variables):

الصيغه العامة للتعريف: Dim VarName As DataType

حيث أن:

(Dim): من كلمات الفيجوال بيسك المحجوزة وتعني -عين، إجعل-

(VarName): هو إسم المتغير المراد تعريفه.

(As): من كلمات الفيجوال بيسك المحجوزة وتعني -من نوع-

(DataType): نوع المتغير الذي نريد.

الكلمات المحجوزة: هي كلمات تابعه للفيجوال بيسك خصت بأسماء معينه تقوم بمهام خاصه ولا يمكن استخدامها كمتغيرات لبرنامجك مثال للتوضيح: END، هي كلمة محجوزة تعني اوقف عمل البرنامج ولا تنفذ اي امر يأتي بعده.

أنواع المتغيرات (Variable Types):

المتغير يحجز جزء من الذاكره لحفظ البيانات والثوابت بداخلها وفي حال أنك لم تعرف المتغيرات فسيقوم الفيجوال بيسك بتعريفها بنوع Variant والتي بدورها تحجز جزء من الذاكره مقداره 1Bytes + Bytes16 لكل حرف فتخيل لو كانت البيانات المخرجه تتكون من 12 خانه على سبيل المثال فهنا نقول بأن مقدار الجزء الذي سوف يحجز هو ← 12 + 16 =28 Bytes هذا لمتغير واحد ماذا لو كنت تحتاج الى 10 متغيرات ولنفرض أن جميعها تعطي الناتج 12 خانه أي ما يعادل 280 Bytes، تخيل لو كان البرنامج أكبر ويحتاج الى متغيرات أكثر وحجم النتائج أضخم سوف تقوم بالضغط على الذاكره وهنا سوف تؤثر على عمل البرنامج وقد تسبب بطئ في عمليه التشغيل والتعامل مع المدخلات والمخرجات وغيره من العمليات، أما إذا قمنا بتعريفها فسوف تحجز مساحه أقل بالذاكره وكلما كان حجم الذاكره التي يحجزها البرنامج أقل أصبح أسرع بالتعامل فمن هنا ظهرت فائده تعريف نوع المتغيرات. أنظر الشكل الذي يوضح أنواع المتغيرات وحجمها بالذاكره.

Data Type	Memory Requirement	Comments
Integer	2 bytes	Whole numbers
Long	4 bytes	Whole numbers
Single	4 bytes	Decimal numbers
Double	8 bytes	Decimal numbers
Currency	8 bytes	Decimal numbers
String	1 byte per character	Text information
Byte	1 byte	Whole numbers
Boolean	2 bytes	Logical values
Date	8 bytes	Date and time
Object	4 bytes	pictures / objects
Variant	16 bytes + 1 byte per character	any of the preceding data type

Copyrights 2012 ©Mohammad Rasheid

مثال:

```
Dim J As Integer
Dim Ro As Boolean
Dim HHH As Single
```

يمكن تعريف المتغيرات جميعها بنفس السطر ويفضل أن تكون من نفس النوع، وتكون صيغتها العامة كالتالي:

```
Dim VarName1 As DataType , VarName2 As DataType , ..... , VarNameN As DataType
```

مثال:

```
Dim A As String, B As String, C As String
Dim Num1 As Integer, Num2 As Integer, Num3 As Integer
```

يمكن تعريف المتغيرات جميعها بنفس السطر حتى لو من تكن من نفس النوع.

مثال:

```
Dim Name1 As String, PhoneNum As Integer, Married As Boolean
```

تعريف المتغيرات عن طريق (Type Suffixes):

يمكننا تعريف المتغيرات الخاصه بنا عن طريق (Type Suffixes) والتي هي عباره عن إختصارات لأنواع المتغيرات (DataType)، بحيث تكون الصيغه العامة للتعريف بإستخدامها هي:

```
Dim VarNameTypeSuffixes
```

والجدول التالي يوضح إختصارات أنواع المتغيرات

Type	Suffix
Integer	%
Long	&
Single	!
Double	#
Currency	@
String	$
Byte	None
Boolean	None
Date	None
Object	None
Variant	None

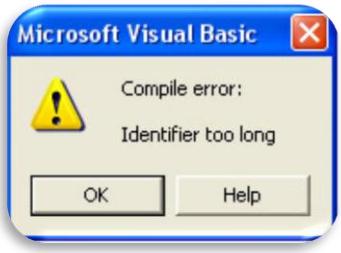
أمثله عليها :

```
Dim Name$
Dim Number%
Dim Avg#
Dim Money@
```


Copyrights 2012 ©Mohammad Rasheid

تسميه المتغيرات (Naming Variables):

لكل شئ في هذا الكون إسم يميزه عن غيره فمثلا: الأسد إسم خاص به ولا يمكننا إطلاقه على حيوان آخر كعصفور مثلا، وحتى في حياتنا عندما نطلق الأسماء فإننا نحاول أن تكون منطقيه ومعبره وذات معنى فلا يمكننا إن نقوم بتسميه طفل صغير بـ X مثلاً، وكذلك الأمر بالنسبه للمتغيرات داخل بيئه الفيجوال بيسك فعندما نقوم بالتسميه لابد لنا من أن نتبع قواعد في التسميه حتى لا تحدث أخطاء في البرنامج ومن هنا لابد لنا من استعراض هذه القواعد وهي:



1 - يجب أن يبتدء إسم المتغير بحرف دائماً وليس رقم أو رمز خاص.

2 - أن لا يحوي إسم المتغير على نقطة.

3 - أن يكون الإسم فريد ضمن الإجراء المتخذ.

4 - أن لا يتجاوز عدد أحرف إسم المتغير 255 حرف.وفي حال زاد عن ذلك تظهر رساله حدوث خطأ.

5 - إذا كان إسم المتغير يتكون من كلمتين فيمكنك الفصل بين الكلمتين بالشرطة السفليه فقط _.

- مجموعه أمثله لأسماء متغيرات مرفوضه في الفيجوال بيسك:

إسم المتغير	مقبول/مرفوض	سبب الرفض إن وجد
X.ca	مرفوض	يحوي على نقطة
2day	مرفوض	ابتدء إسم المتغير برقم
Face book	مرفوض	يحوي على مسافه
Sweet.Girl	مرفوض	يحوي على نقطة

- مجموعه أمثله لأسماء متغيرات مقبوله في الفيجوال بيسك:

إسم المتغير	مقبول/مرفوض	سبب الرفض إن وجد
Love	مقبول	-
Avg	مقبول	-
Face_book	مقبول	-

Copyrights 2012 ©Mohammad Rasheid

تعيين قيم للمتغيرات:

بالتأكيد عند كتابتنا للمتغيرات كان هدفنا تخزين بيانات أو قيم معينه بداخلها، بحيث عندما نقوم بإستداعها تعطيني القيم التي أريد وأتوقع. ومن هنا لابد لنا من معرفه كيفيه تعيين قيم للمتغيرات، بحيث والأغلب يتم وضع إسم المتغير الذي نريد ثم إشاره اليساوي ثم القيمة الخاصه بهذا المتغير وهنا يجب التوضيح بأنه عند تعيين قيم للمتغيرات النصيه فإنها تكتب بين علامتي إقتباس بعكس المتغيرات الرقمية التي تكتب مباشره، طبعاً توجد طريقتين للتعيين:

- **الأولى: باستخدام (Let)**

مثال:

```
Let Name = "Mohammad Rasheid"
Let Number = 1234567890
```

- **الثانيه: مباشره**

مثال:

```
Name = "Mohammad Rasheid"
Number = 1234567890
```

معلومة: نضع النص بين علامتي إقتباس والقيم الرقميه بدونها لماذا؟

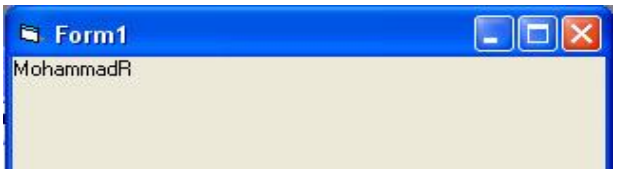
ببساطة لأنني إذا قمت بوضع القيم الرقميه بين علامتي إقتباس فستصبح كأنها قيم نصيه ولن أستطيع تطبيق العمليات الحسابيه عليها مثل: الجمع والطرح ووو .

العمليات الحسابيه لا تستخدم مع القيم النصيه بإستثناء علامة الجمع (+) والتي تعني علامة دمج، أي الدمج بين نصين بدون أن تحوي على فراغ.

مثال:

```
Private Sub Command1_Click()
A = "Mohammad"
B = "R"
Print A + B
End Sub
```

بما أن علامة الجمع وضعت بين قميتين نصيتين فإنها تعامل كأداة دمج وعليه يكون ناتج التنفيذ



وتكافئها من الرموز الخاصه الرمز (&) الذي يعتبر أداه دمج أيضاً مع القيم النصيه ويشابه في عمله إشاره الجمع.

Copyrights 2012 ©Mohammad Rasheid

التعريف بالمتغيرات النصيه

هناك نوعين من التعريف: التعريف المفتوح والتعريف المحدود

أي عندما نقوم بتعريف أي متغير إلى النوع النصي فإنه سيقوم بإستقبال وتخزين النص أو الجمله مهما كان طولها بعكس التعريف المحدود حيث يتم تحديد طول هذا النص ويقوم بإستقبال فقط الطول المحدد له فقط مهما بلغ طول النص. وصيغته العامة هي:

```
Dim VarName As String * strLength
```

مثال على التعريف المفتوح:

```
Dim Name As String
Name = "Mohammad Rasheid"
Print Name
```

الان عند تطبيق الكود سيقوم بتعريف المتغير (Name) الى نوع متغير نصي ومن ثم تخزين النص (Mohammad Rasheid) داخل المتغير (Name) ومن ثم يقوم بطباعته على الشاشه وعليه تكون شاشه المخرجات هي:

لاحظ قام بطباعه كامل النص. الان سنقوم بإعاده تعريف المتغير بحسب خاصيه التعريف المحدودوسيصبح الكود:

```
Dim Name As String * 10
Name = "Mohammad Rasheid"
Print Name
```

الان عند تطبيق الكود سيقوم بتعريف المتغير (Name) الى نوع متغير نصي وأقصى طول للنص سيكون 10 أحرف، ومن ثم تخزين النص (Mohammad Rasheid) داخل المتغير (Name) ولكن وبما أننا حددنا طول النص فإنه سيستقبل أول 10 أحرف فقط من النص المدخل ثم يقوم بطباعته وعليه سيكون الناتج هو:

Copyrights 2012 ©Mohammad Rasheid

نطاقات التعريف:

Dim VarName1 [As VarType1][, Varname2 [As VarType2]]
Private VarName1 [As VarType1][, Varname2 [As VarType2]]
Public VarName1 [As VarType1][, Varname2 [As VarType2]]
Static VarName1 [As VarType1][, Varname2 [As VarType2]]

Dim, ***Private***, ***Static*** and ***Public*** are Visual Basic keywords that define **how and when the variables may be used**. *VarName1* and *VarName2* are the names of the variables you want to declare. The syntax above shows that you may define a number of variables in the same statement but you should be separate between them with a comma, you may define any number of variables .VarType1 and VarType2 are the data type of variable

وتستخدم هذه الكلمات المحجوزة (*Dim*, *Private*, *Static* and *Public*) لتعرف لنا المتغيرات وتبين للفيجوال بيسك كيف ومتى تستخدم هذه المتغيرات.

ويمكننا الاختصار بأن نقول:

نوع التعريف	مكان الكتابه	المدى
Dim	في الـ (General)	على مستوى النموذج بالكامل حيث يتم تخزين القيمة بالمتغير ما دام البرنامج يعمل، ويمكن إستدعاء المتغير من أي مكان بالبرنامج.
	في الإجراء المراد برمجته	على مستوى الإجراء المراد تنفيذه فقط.
Private	في الـ (General)	على مستوى النموذج بالكامل حيث يتم تخزين القيمة بالمتغير ما دام البرنامج يعمل، ويمكن إستدعاء المتغير من أي مكان بالبرنامج.
Public	في الـ (General)	على مستوى النموذج بالكامل حيث يتم تخزين القيمة بالمتغير ما دام البرنامج يعمل، ويمكن إستدعاء المتغير من أي مكان بالبرنامج.
Static	في الإجراء المراد برمجته	على مستوى الإجراء حيث يتم تخزين القيمة بالمتغير ما دام البرنامج يعمل.

Copyrights 2012 ©Mohammad Rasheid

الثوابت (Constants)

الثوابت ومن إسمها تدل على عملها وهي قيم ثابته لا تتغير أثناء تنفيذ البرنامج ومن الأمثله عليها: π = 3.14 حيث أن قيمتها لا تتغير ويمكننا كتابتها في البرنامج في القسم (General) وبحسب الصيغه التاليه:

```
Const CName As DataType = Value
```

حيث أن:

(Const): من كلمات الفيجوال بيسك المحجوزة وتعني ثابت إختصار لـ Constants

(CName): هو إسم الثابت الذي نريد.

(As): من كلمات الفيجوال بيسك المحجوزة وتعني -من نوع-

(DataType): نوع المتغير الذي نريد.

(Value): قيمة هذا الثابت.

مثال:

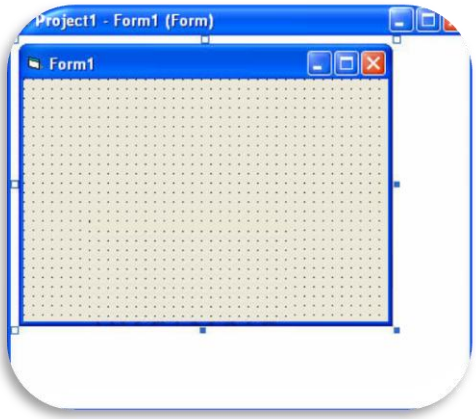
```
Const PI As Double = 3.14
Private Sub Command1_Click()
Print PI
End Sub
```

وعند تنفيذ الكود سيتم طباعه قيمة PI مباشره لأننا قمنا بتعريفها بأنها ثابته، ولكن لو لم نعرفها وطلبنا أن يقوم بطباعه قيمة PI فسيقوم بطباعه صفر، لماذا؟ لأن قيمة المتغير الإفتراضي في حال لم نقم بتعيين قيمة له هي صفر.

من أجزاء الشاشه الرئيسيه

تتكون الشاشه الرئيسيه لبرمجيه الفيجوال بيسك من أجزاء عده إخترنا أهمها بحسب الماده المقرره في (حاسوب 2 لطلبه التخطيط) وهي:

1 - حيز العمل (Work Space): والذي من خلاله نقوم بإضافه الأدوات والصور وغيرها وهي ما تظهر للمستخدم عند تشغيليها.



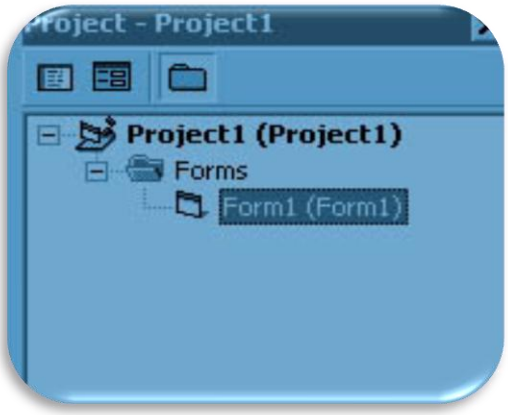
Copyrights 2012 ©Mohammad Rasheid

2 - صندوق الأدوات (ToolBox): والذي يوفر عدد لا بأس به من الأدوات الأساسيه لأي مبرمج يرغب بتصميم برنامجه

NO.	Name	Description
01	Pointer	المؤشر ويساعد في تحديد ومسك الأدوات.
02	Label	أداه لكتابه النص الثابت على النموذج.
03	Frame	أداه تساعد في تحديد وجمع الأدوات مع بعضها.
04	CheckBox	أداه الإختيار المتعدد.
05	ComboBox	أداه تسمح للمستخدم بإختيار خيار واحد فقط.
06	HScrollarBar	أشرطة التمرير الأفقيه.
07	Timer	مؤقت النظام، حيث يسمح بتنفيذ الأوامر في أوقات محدده.
08	DirListBox	تقوم بعرض المجلدات الموجوده في الجهاز.
09	Shape	تسمح برسم عده أشكال على النموذج.
10	Image	أداه تسمح بإضافه صوره بدون إطار.
11	OLE	أداه تسمح بدمج برامج أخرى داخل النموذج الخاص بك.
12	PictureBox	أداه تسمح بإضافه صوره داخل إطار.
13	TextBox	أداه تسمح بإدخال بيانات أو تقوم بعرض بيانات.
14	Command	زر الأمر الذي يستخدم لتنفيذ الأوامر بمجرد الضغط عليه.
15	OptionBox	أداه الإختيار الواحد.
16	ListBox	تقوم بعرض جميع الخيارات تحت بعضها ةتسمح لك بالتحديد على خيار.
17	VScrollarBar	أشرطة التمرير العاموديه.
18	DriveListBox	تقوم بعرض الأقراص الموجوده في الجهاز.
19	FileListBox	تقوم بعرض الملفات الموجوده في الجهاز.
20	Line	تسمح برسم خط على النموذج.
21	Data	تقوم بتصفح وعرض البيانات الموجوده في قاعده البيانات.

Copyrights 2012 ©Mohammad Rasheid

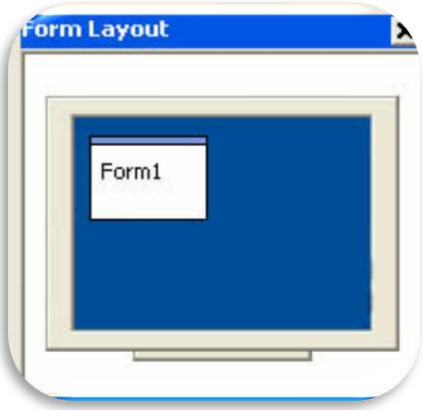
3 - مستعرض المشروع (Project Explorer): حيث يقوم بعرض جميع النوافذ الموجوده بالمشروع.



4 - نافذه الخصائص (Properties Windows): حيث تقوم بالتحكم بخصائص النموذج والأدوات التي سوف تظهر للمستخدم، هناك خصائص مشتركة بين الأدوات وهناك خصائص غير مشتركة.

مثال على الخصائص المشتركة: Visible **مثال على الخصائص الغير مشتركة: Icon**

5 - نافذه موقع النموذج (Form Layout): حيث تظهر مكان النموذج على الشاشه عند تشغيليه وبإمكانك تغير موقعه بالضغط على النموذج الموجوده داخل الشاشه كما بالشكل وتحريكه لأي موقع تريد.



Copyrights 2012 ©Mohammad Rasheid

تغيير الخصائص عند تشغيل البرنامج (Set Properties at Run Time)

يمككنا التحكم بخصائص الأدوات المضافه على النموذج من خلال برمجتها داخل الحدث (Form_Load) الذي يقوم بتنفيذ ما يحوي من أوامر بمجرد تشغيل البرنامج، وطبعاً يتم ذلك بحسب الصيغه التالية:

ObjectName.Property = NewValue

حيث أن:

(ObjectName): إسم الأداه التي نريد.

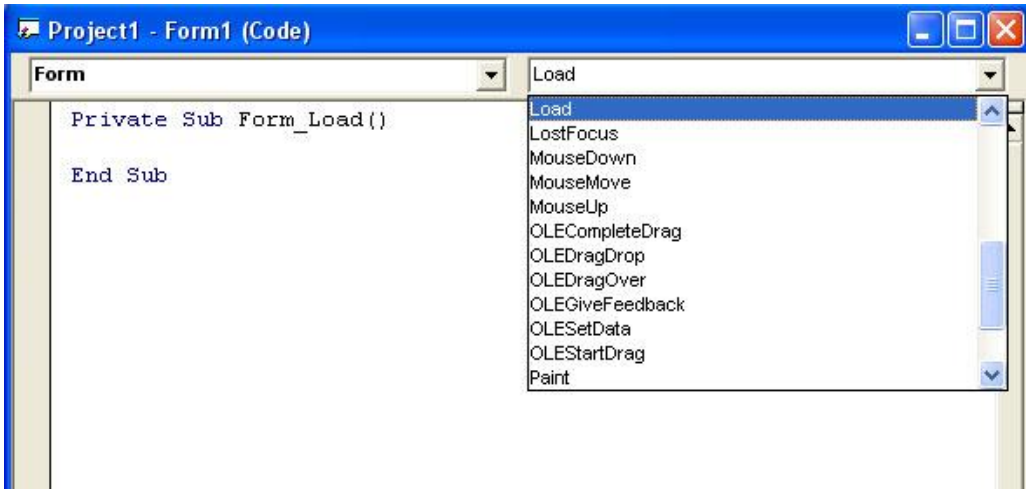
(Property): إسم الخاصيه المراد التحكم بها.

(NewValue): القيمة الجديده لهذه الخاصيه.

مثال:

```
Form1.Caption = "Jordan"     --->     تغير إسم النموذج
Command1.Enabled = False     --->     تفعيل أو تعطيل زر الأمر
Form1.BackColor = vbGreen    --->     تعبئة النموذج باللون الأخضر
```

الأحداث الخاصه بالأدوات: يتميز كل حدث عن الآخر بكيفيه تنفيذه للأكواد ومن أشهرها (Load) و (Click) و (DblClick).



Copyrights 2012 ©Mohammad Rasheid

يوجد أيضاً أحداث خاصه بكل من لوحه المفاتيح و الماوس وهي كالتالي:

الأحداث الخاصه بلوحة المفاتيح:

KeyDown , KeyPress , KeyUp

حيث أن:

(KeyPress): تقوم بتنفيذ الأوامر بمجرد الضغط على أي زر من أزرار الكيبورد.

(KeyDown):تقوم بتنفيذ الأوامر بمجرد الضغط على الزر نحو الأسفل.

(KeyUp):تقوم بتنفيذ الأوامر بمجرد رفع الأصبع عن الزر.

الأحداث الخاصه بالماوس:

MouseMove , MouseDown , MouseUp

حيث أن:

(MouseMove): تقوم بتنفيذ الأوامر بمجرد تحرك الماوس على النموذج.

(MouseDown):تقوم بتنفيذ الأوامر بمجرد الضغط على زر الماوس نحو الأسفل.

(MouseUp):تقوم بتنفيذ الأوامر بمجرد رفع الأصبع عن زر الماوس.

⭐⭐ The End of Chapter one ⭐⭐

Copyrights 2012 ©Mohammad Rasheid

visual basic 6

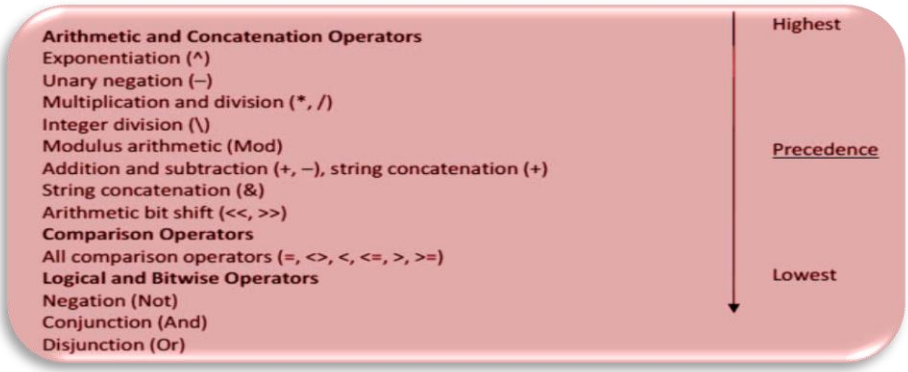
الملحق الأول

INDEX

Copyrights 2012 ©Mohammad Rasheid

العمليات الحسابيه:

لمعرفه طريقه إستخراج ناتج عمليه حسابيه لابد لنا أن نحاكي عمل الآله الحاسبه والتي تعمل وفق أولويات ثابته، والشكل في الأسفل يوضح هذه الأولويات:



دائماً وفي حال وجود متغيرات في العمليات الحسابيه قم بتعويض قيم هذه المتغيرات إن وجدت وفي حال لم تعطى قيم هذه المتغيرات تكون قيمتها الإفتراضيه تساوي صفر.

جد ناتج كل من العمليات الحسابيه التاليه:

```
Const x = 5, y = 4, z = 3
Private Sub Form_Load()
MsgBox ((x ^ k) + y - x * -z) * 0.5
End Sub
```

10

```
Const Num1 = 2, Num2 = 4, Num3 = 3
Private Sub Form_Load()
MsgBox (((Num1 ^ Num2) / Num1) * Num3) \ 5
End Sub
```

4

```
Const A = 2, B = 4, X = 3
Private Sub Command1_Click()
y = 4 / B * 3 ^ 2 + 7 * A - 8
Print y
End Sub
```

15

```
Const A = 2, B = 4, X = 3
Private Sub Command1_Click()
y = 3 * (6 \ B) + (6 - A) ^ A
Print y
End Sub
```

19

```
Const A = 2, B = 4, X = 3
Private Sub Command1_Click()
y = (2 + A Mod B) ^ 2 / B
Print y
End Sub
```

4

دائماً عوض قيم المتغيرات ثم قم بحل الأولويات واحده واحده.

Copyrights 2012 ©Mohammad Rasheid

عمليات المقارنه والعمليات المنطقيه:

1- عمليات المقارنه: تكون أولوياتها بعد أولويات العمليات الحسابيه So قم بتعويض قيم المتغيرات إن وجدت ثم إستخرج ناتج العمليات الحسابيه بحسب الأولويات ومن ثم توجه إلى عمليات المقارنه (<> , <= , >= , = , > , <) وتكون نتيجتها إما صواب (True) أو خطأ (False).

ما ناتج كل من البرامج التاليه:

```
Private Sub Command1_Click()
Print 7 > 5
End Sub
```

```
Private Sub Command1_Click()
Print 22 * 3 > 100
End Sub
```

```
Private Sub Command1_Click()
Print ((5 * -1) / -1) ^ 3 > 100
End Sub
```

2- العمليات المنطقيه: تستخدم العمليات المنطقيه مع كل من العمليات الحسابيه وعمليات المقارنه معاً بحيث يشكل إجتماعهم معاً معادله معقده نوعاً ما ولكن كن على ثقه بأن المعادله المعقده ما هي إلا مجموعه من العمليات والمعادلات الصغيره والبسيطه ويبقى على الذكي تفكيكها وحلها.

ولكن وقبل أخذ الأمثله لابد لنا من التطرق إلى المعاملات المنطقيه وهي (Not , And , OR) وحتى نتعلم كيفيه التعامل معها لابد من دراسه جداول الصواب والخطأ لكل واحده. أنظر إلى الجداول المرفقه.

X	NOT X
F	T
T	F

X	Y	X And Y
F	F	F
T	F	F
F	T	F
T	T	T

X	Y	X OR Y
F	F	F
T	F	T
F	T	T
T	T	T

كما نلاحظ من الجداول السابقه أن المعامل (Not) يعطي عكس الإجابه فيحول الصحيح إلى خاطئ والخاطئ إلى صحيح.

أما المعامل (And) فيجب أن تكون الإجابيتين صحيحتين حتى يعطي (True)، ولا يكفي توفر إجابه صحيحه.

أما المعامل (OR) فيجب أن تكون كلتا الإجابتي خاطئه حتى تعطي (False) بعكس (And) وهنا وبمجرد توفر إجابه واحده صحيحه فإنها تعطي (True).

Copyrights 2012 ©Mohammad Rasheid

ما ناتج كل من البرامج التاليه:

```
Const a = 5, b = 7, c = 2, d = 4
Private Sub Command1_Click()
Print a <= b And c + 1 < b + 1
End Sub
```

```
Const a = 5, b = 7, c = 2, d = 4
Private Sub Command1_Click()
Print a - c <> a Or b <= d
End Sub
```

```
Const a = 5, b = 7, c = 2, d = 4
Private Sub Command1_Click()
Print Not (a ^ 2 + 4 > b * d)
End Sub
```

تدريب: جد ناتج كل من البرامج التاليه:

```
Const a = 5, b = 7, c = 2, d = 4
Private Sub Command1_Click()
Print Not (b < c And (d <> a - c Or a > -d * c))
End Sub
```

```
Const a = 5, b = 7, c = 2, d = 4
Private Sub Command1_Click()
Print a + b Mod 5 \ c > d Or 6 ^ 2 <> d + 2 * 2 And b < 2
End Sub
```

Copyrights 2012 ©Mohammad Rasheid

الدوال الجاهزه (Built-In Function):

وهي عباره عن كلمات محجوزة تؤدي وظائف معينه، وسنقسمها إلى:

1 - (String Function) دوال السلاسل الرمزيه.
2 - (Math Function) الدوال الرياضيه.
3 - (Time & Date Function) دوال الوقت والتاريخ.

String Function النصوص

- **LEN** حساب طول النص سواء بمتغير أو بين علامتي إقتباس

```
Print Len("Jordan First")
Print Len("Yossef Arafat")
```

```
12
13
```

- **Mid** بتحديد النص التي تريد وتحدد من أين يبدأ القراءة وبعد أي حرف يتوقف

```
Print Mid("I Love You", 2, 5)
Print Mid("Jordan", 1, 3)
```

```
Love
Jor
```

```
x = "Yossef JO-First Arafat"
Mid(x, 1, 6) = "xxxxxxx"
Mid(x, 17, 6) = "xxxxxxx"
Print x
```

```
xxxxxx JO -First xxxxxx
```

- **Left** تبدأ من الجانب الأيسر وتحدد عدد الأحرف التي سوف تأخذها وما تبقى يحذف

```
Print Left("Love Above Hate", 4)
Print Left("Hate Me", 1)
```

```
Love
H
```

- **Right** تبدأ من الجانب اليمين وتحدد عدد الأحرف التي سوف تأخذها وما تبقى يحذف

```
Print Right("Love Above Hate", 4)
Print Right("Hate Me", 1)
Print Right("Don't Ask me", 6)
```

```
Love
H
Ask Me
```


Copyrights 2012 ©Mohammad Rasheid

- **UCase** تحويل حاله الاحرف من الصغيره الى الكبيره

```
Print UCase("The voice of Jordan 2012")
Print UCase("Yossef Arafat")
```

```
THE VOICE OF JORDAN 2012
YOSSEF ARAFAT
```

- **LCase** تحويل حاله الاحرف من الكبيره الى الصغيره

```
Print LCase("VISUAL BASIC")
Print LCase("AMMAN")
```

```
visual basic
amman
```

- **LTrim** حذف الفراغ إن وجد من يسار النص

```
Print LTrim(" VISUAL BASIC ")
Print LTrim(" Dr91")
```

```
VISUAL BASIC
Dr91
```

- **RTrim** حذف الفراغ إن وجد من يمين النص

```
Print RTrim(" VISUAL BASIC ")
Print RTrim(" Dr91")
```

```
 VISUAL BASIC
 Dr91
```

- **Trim** حذف الفراغ من يسار ويمين النص إن وجد

```
Print Trim("        Xmen        ")
```

```
Xmen
```

- **ASC** يعطيك الرقم الذي يكافئ الحرف حسب جدول **ASCII**

```
Print Asc("A"), Asc("a"), Asc(0)
```

```
65   97   48
```

- **Chr** تحول الرقم الخاص من جدول آسكي إلى الحرف الخاص به

```
Print Chr(70)
```

```
F
```


Copyrights 2012 ©Mohammad Rasheid

- **Abs** تحويل الارقام السالبه الى أرقام موجبه

```
Print Abs(-125)
Print Abs(-0.2)
```

```
125
0.2
```

- **Sqr** تعطي جذر الرقم المطلوب

```
Print Sqr(81)
Print Sqr(3)
```

```
9
1.732
```

- **Rnd** تعطي أرقام عشوائيه بين 0 و 1

```
Print Rnd()
Print Rnd() * 10
```

- **Int** إعطاء الجزء الصحيح من الرقم وحذف الباقي

```
Print Int(23.2240245)
Print Int(0.565865485)
```

```
23
0
```

- **Round** تقريب العدد المدخل حيث يمكننا تحديد عدد الخانات العشريه

```
Print Round(25.454)
Print Round(25.454, 0)
Print Round(25.454, 1)
Print Round(95.175, 2)
```

```
25
25
25.5
95.18
```

- **Val** تقوم بإعاده الجزء الرقمي من النص المدخل

```
Print Val("2text")
Print Val("91Dr")
Print Val("Damar2")
```

```
2
91
0
```


Copyrights 2012 ©Mohammad Rasheid

- **Date** تقوم بإعطاء تاريخ اليوم

```
Print Date      3/22/2012
```

- **Time** تقوم بإعطاء الوقت الحالي

```
Print Time      2:36:38 AM
```

- **Now** تقوم بإعطاء تاريخ ووقت اليوم الحالي

```
Print Now       3/22/2012 2:37:20 AM
```

- **Hour** تقوم بإعطاء الساعه من الوقت الحالي بنظام 24

```
Print Hour(Time)        2
```

- **Minute** تقوم بإعطاء الدقائق من الوقت الحالي

```
Print Minute(Time)      40
```

- **Second** تقوم بإعطاء الثواني من الوقت الحالي

```
Print Second(Time)      52
```

- **WeekDay** تقوم بإعطاء موقع اليوم من الأسبوع ولكن بعد تحديد بدايه الأسبوع

```
Print Weekday(Date, vbFriday)       7
```

- في حال لم تحدد بدايه الأسبوع فيكون إفتراضياً يوم الأحد هو بدايه الأسبوع

نهايه الملحق الأول

Copyrights 2012 ©Mohammad Rasheid

visual basic 6

chapter two

INDEX

Copyrights 2012 ©Mohammad Rasheid

DIALOGUE BOXES – صناديق الحوار

تقسم صناديق الحوار إلى ثلاث أنواع:

1 - صندوق الإدخال (Input Box)
2 - صندوق الرسائل (Message Box)
3 - صناديق الحوار الداخليه (Built-in Dialogue Boxes)

صناديق الإدخال (InputBox):

وهي عباره عن صناديق حوار تظهر للمستخدم تطلب منه بأن يقوم بإدخال معلومات معينه يحتاجها البرنامج حتى يستطيع أن يكمل عمله ومن الأمثله العمليه على هذا الصندوق برنامج إحتساب الأعمار أو برنامج إدخال أسماء العمال في المصنع وغيرها الكثير.

الصيغه العامة للتعريف:	VarName = InputBox(Message, Title)

حيث أن:

(VarName): هو إسم المتغير المراد تخزين البيانات المدخله من المستخدم فيه.

(InputBox): من كلمات الفيجوال بيسك المحجوزة والخاصه بإظهار صندوق الإدخال.

(Message): نص الرساله التي ستظهر في صندوق الإدخال وتطلب من المستخدم كتابه البيانات المطلوبة.

(Title): عنوان الرساله التي ستظهر في صندوق الإدخال.

مثال تطبيقي:

قم بفتح برمجيه الفيجوال بيسك وقم بإضافه زر أمر وصندوق نص كما بالشكل:

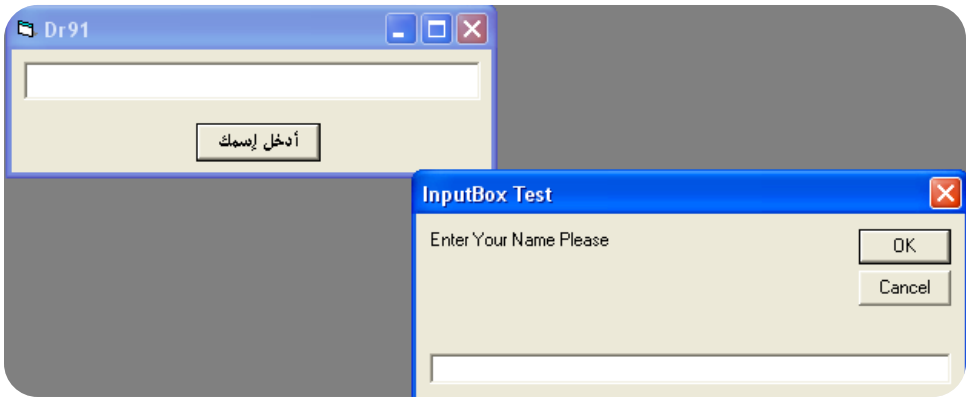
إفتح شاشه الكود الخاصه بزر الأمر (أدخل إسمك) وأكتب الكود التالي:

```
Private Sub Command1_Click()
UserName = InputBox("Enter Your Name Please", "InputBox Test")
Text1.Text = "Mr." & UserName
End Sub
```


Copyrights 2012 ©Mohammad Rasheid

شرح الكود الأول: في السطر الأول تم كتابة الأمر الخاص بصندوق الإدخال بحسب الصيغه الخاصه بها وسيتم تخزين ما يدخله المستخدم داخل المتغير (UserName)، أما السطر الثاني سيتم إدخال النص (.Mr) ومن ثم إسم المستخدم الذي قام بإدخاله وقد إستخدمنا الرمز (&) الذي تم شرحه سابقاً والذي يعني أداة دمج.

والان عند التطبيق قم بالضغط على زر الأمر (أدخل إسمك) لاحظ سوف يظهر صندوق الإدخال كما بالشكل



قم بإدخال إسمك وليكن على فرض (Sultan Al-Khatib) ثم إضغط (OK)، لاحظ تم إضافه الإسم في صندوق النص مسبوقاً بـ (.Mr) كما بالشكل

⭐ لاحظ لو قمنا بتعريف المتغير (UserName) بحسب التعريف المحدود الخاص بالمتغيرات النصيه أي قمنا بإضافه هذا الكود قبل الكود الخاص بصندوق الإدخال وتطبيق ما قمنا به وعلى فرض كان النص المدخل هو نفسه فسيكون الناتج

Copyrights 2012 ©Mohammad Rasheid

صندوق الرسائل (Message Box):

وتنقسم صناديق الرسائل إلى نوعين الأول الذي يهدف إلى إظهار رساله للمستخدم ولا يهتم إلى رده والنوع الثاني الذي يهدف إلى معرفه رد المستخدم وسيقوم بالتنفيذ بحسب رده.

النوع الأول: هذا النوع وكما قلنا لا يهتم برد المستخدم كل ما يهمة هو إظهار رساله للمستخدم قد تفيد بوقوع خطأ أو إظهار معلومة ومن ثم يكمل عمل البرنامج.

الصيغه العامة: MsgBox Message, [Buttons], Title

حيث أن:

(MsgBox): من كلمات الفيجوال بيسك المحجوزة والتي تعني أمر بإظهار رساله للمستخدم.

(Message): هو نص الرساله الذي تريد إظهاره للمستخدم.

(Buttons): الأزرار التي ستظهر مع الرساله وفي حال لم تقم بتحديدها تكون إفتراضياً (OK) مع تحديد أيقون الرساله (نوعها).

(Title): عنوان الرساله التي ستظهر للمستخدم.

الأنواع الأيقون وقيمها أنظر الجدول:

Table 2: Message Icon

Icon	Value	Constant	Comment
	32	vbQuestion	**Asking the user some kind of information.**
	16	vbCritical	**Used when an error occurred.**
	48	vbExclamation	**Used when an error occurred, and the user must cover it.**
	64	**vbInformation**	**Telling the user some kind of information.**

وهنا يمكنك أن تكتب قيمة هذه الأيقون مباشره مكان كلمة (Buttons) أو أن تكتب صيغتها مثال (vbInformation).

ويمكنك أن تضيف الأزرار الخاصه بهذه الرساله من خلال كتابه قيمها أو صيغتها أنظر الجول:

Table 3: Command Button Values

Command Button	Value	Constant
OK	**0**	vbOkOnly
Ok and Cancel	**1**	vbOkCancel
Abort, Retry and Ignore	**2**	vbAbortRetryIgnore
Yes, No and Cancel	**3**	vbYesNoCancel
Yes and No	**4**	vbYesNo
Retry and Cancel	5	vbRetryCancel

عند إختيار الأيقون والأزرار يجب أن نفصل بينهم بإشاره الجمع (+).

Copyrights 2012 ©Mohammad Rasheid

مثال: قم بإضافه زر أمر وأفتح شاشه الكود الخاصه وقم بكتابه الكود التالي

```
MsgBox "I'm Jordanian and I pride by this", vbOKOnly + vbInformation, "Dr91"
```

لاحظ عند تنفيذ الكود سوف تظهر رساله للمستخدم كما بالشكل

ويمكننا كتابه الكود السابق عن طريق قيمها فيكون الكود:

```
MsgBox "I'm Jordanian and I pride by this", 64, "Dr91"
```

النوع الثاني: الذي يهدف إلى معرفه رد المستخدم وسيقوم بالتنفيذ بحسب رده، كيف ذلك؟ ببساطة عند قيام المستخدم بالضغط على أي زر من أزرار الرساله فإن هذا الزر يقوم بإرجاع قيمة سيتم تخزينها داخل متغير وعلى أساسه هذه القيمة سيتم التنفيذ ومن الأمثله العمليه على هذا النوع من صناديق الرسائل عندما تود الخروج من برنامج معين وليكن (Microsoft Word) لاحظ ستظهر أمامك رساله تؤكد عمليه الخروج وبحسب ما تختاره تقوم بالتنفيذ.

الصيغه العامة: VarName = MsgBox (Message, [Buttons], Title)

حيث أن:

(VarName): إسم المتغير الذي سيقوم بحفظ القيمة المرتجعه.

(MsgBox): من كلمات الفيجوال بيسك المحجوزة والتي تعني أمر بإظهار رساله للمستخدم.

(Message): هو نص الرساله الذي تريد إظهاره للمستخدم.

(Buttons): الأزرار التي ستظهر مع الرساله وفي حال لم تقم بتحديدها تكون إفتراضياً (OK) مع تحديد أيقون الرساله (نوعها).

(Title): عنوان الرساله التي ستظهر للمستخدم.

يجب معرفه القيم الذي تعيدها الأزرار والتي يتم تخزينها داخل المتغير لذلك لاحظ الجدول التالي:

Table 5:(buttons return values)

Button	Value
Ok	1
Cancel	2
Abort	3
Retry	4
Ignore	5
Yes	6
No	7

Copyrights 2012 ©Mohammad Rasheid

⭐ لن يتم كتابة أي مثال على هذا النوع من الرسائل لأنه سيتم التطرق لها في (Chapter 3) لأن هذا النوع من الرسائل يأتي مع جمل الشرط .

⚡ صناديق الحوار الداخليه (Built-in Dialogue Boxes)

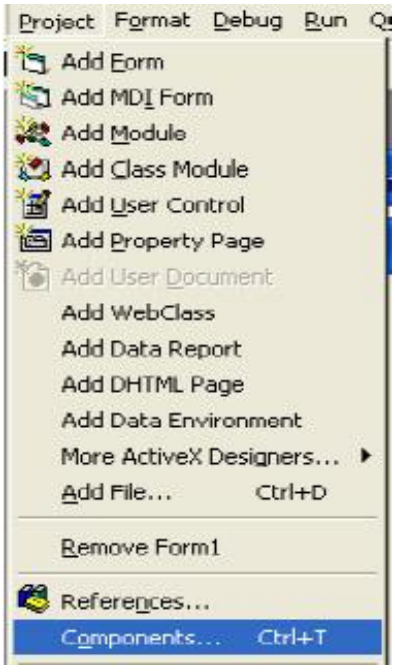
تستخدم هذه الصناديق إما لإظهار صندوق:

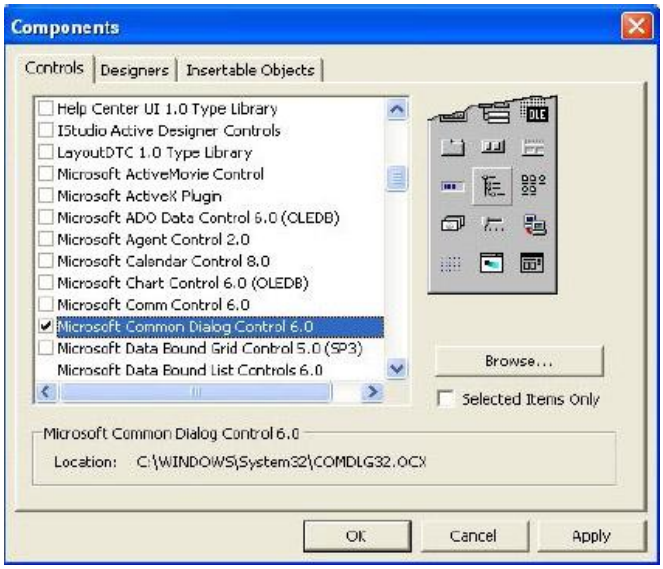
1 - صندوق الألوان (ShowColor)
2 - صندوق الخطوط (ShowFont)
3 - صندوق الطباعه (ShowPrinter)
4 - صندوق فتح ملف (ShowOpen)
5 - صندوق حفظ ملف (ShowSave)

لن أقوم بشرحها بل سأكتفي بوضع أسئله بخصوصها (بصراحة لا تأتي عليها أسئله – مع هيك أدرسها)

س1: كيف أقوم بإضافه الأداه المساعدة Common Dialog ؟

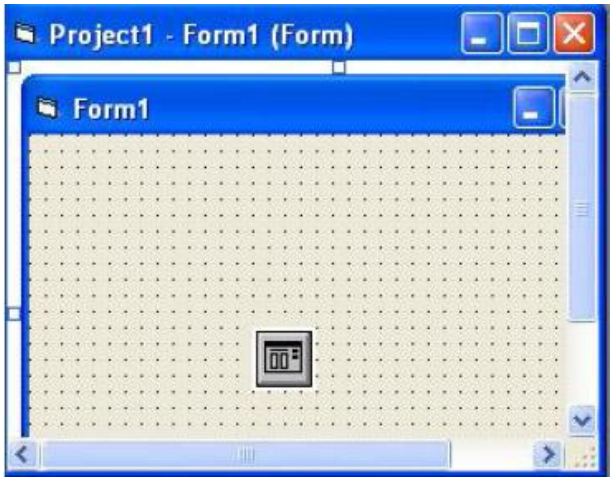
1 - من قائمة اللوائح إختر (Project) ومنها إختر (Components) ستلاحظ ظهور صندوق حوار (Components) سوف تظهر العديد من الملفات إختر منها (Microsoft Common Dialog Control 6.0) ثم موافق.





Copyrights 2012 ©Mohammad Rasheid

2 - لاحظ ظهور أداة جديده في صندوق الأدوات أنقر عليها نقراً مزدوج ولاحظ ظهورها في حيز العمل.



⭐⭐ The End of Chapter Two ⭐⭐

Copyrights 2012 ©Mohammad Rasheid

visual basic 6

chapter three

INDEX

Copyrights 2012 ©Mohammad Rasheid

جمل الشرط

تقدم لغة الفيجوال بيسك أيضاً القدره على السيطره بشكل أكبر من خلال جمل شرط تستطيع من خلالها أن تتحكم في المخرجات بحسب البيانات المدخله فلو أردت أن تقوم بعمل برنامج إذا كان الرقم المدخل أقل من صفر يطبع بأن الرقم المدخل سالب والعكس الرقم موجب، لاحظ بأنك تحتاج كود من خلاله تستطيع مقارنه الارقام المدخله بما تريده وطباعه الأمر بحسب الرقم المدخل!، لذلك يوجد في بيئة الفيجوال بيسك هذه الجمل المعروفه بجمل الشرط والتي تقسم إلى:

1 - If then/else
2 - Select case
3 - Switch
4 - Choose

إذا لدينا في هذا القسم أربع جمل شرط، كل واحده أبسط من الأخرى وحتى تستطيع أن تتعامل معها يجب عليك التتبع وسنرى كيف يمكننا ذلك في هذا القسم. ولكن وقبل الشروع في شرح هذه الجمل لا بد لنا من التنويه بأن الشرط المستخدم في جمل الشرط قد يكون عمليات حسابيه أو منطقية وقد تم التطرق وشرح كيفيه التعامل مع هذا النوع من الأسئلة ودراسة جداول الصواب والخطأ لذلك سأنتقل مباشره إلى جمل الشرط.

جملة الشرط (IF Then Else)

تعتبر هذه الجملة من جمل الشرط الأكثر إستخداماً ونظام عملها ينص على المقارنه وفي حال كان جواب عمليه المقارنه صائب فيتم الإنتقال وتنفيذ الأوامر الموجوده بعد الشق (Then) وفي حال كان ناتج عملية المقارنة خاطئ فيتم الإنتقال وتنفيذ الأوامر الموجوده بعد الشق (Else).

الصيغه العامة:	IF Condition Then vbStatement Else vbStatement

حيث أن:

(IF): من كلمات الفيجوال بيسك المحجوزة وتعني إذا.

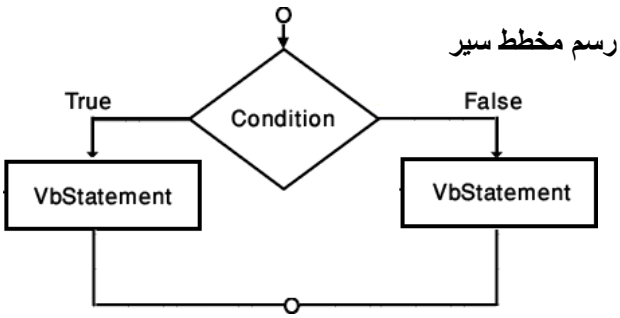
(Condition): عمليه المقارنه التي على أساس ناتجها يتم تنفيذ الأكواد.

(Then):من كلمات الفيجوال بيسك المحجوزة ويتم تنفيذ ما بعدها من أكواد في حال كان جواب هذه العمليه صحيح.

(Else):من كلمات الفيجوال بيسك المحجوزة ويتم تنفيذ ما بعدها من أكواد في حال كان جواب هذه العمليه خاطئ.

(vbStatement): الأوامر التي تريد تنفيذها.

لاحظ الشكل التالي الذي يوضح آليه جملة الشرط (IF Then Else) عن طريق رسم مخطط سير العمليات الخاص بها.

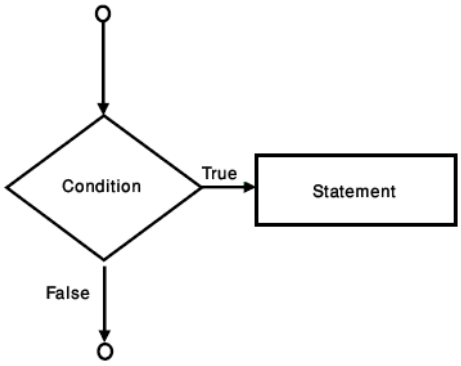


Copyrights 2012 ©Mohammad Rasheid

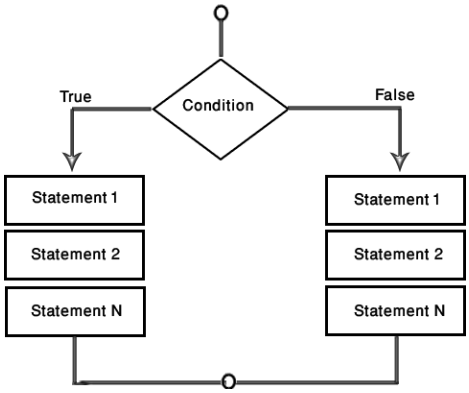
لاحظ بأنه يمكننا أن نستغني عن الشق (Else) ليصبح الشكل العام للجمله كالتالي:

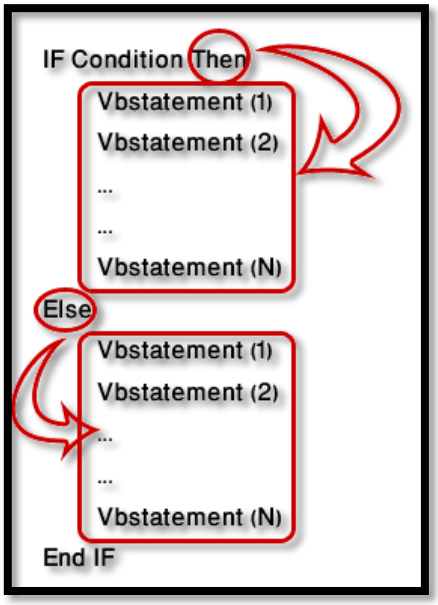
```
IF Condition Then vbStatement
```

بحيث بعد عمليه المقارنه إذا كان ناتج عملية المقارنة صحيح فيتم تنفيذ ما بعد الشق (Then)، أما إذا كان الناتج خاطئ فيتم الخروج من جمله الشرط وتنفيذ ما تبقى من أكواد إن وجد. لاحظ الشكل التلي الذي يوضح آليه العمل من خلال رسم مخطط سير العمليات الخاص بها.

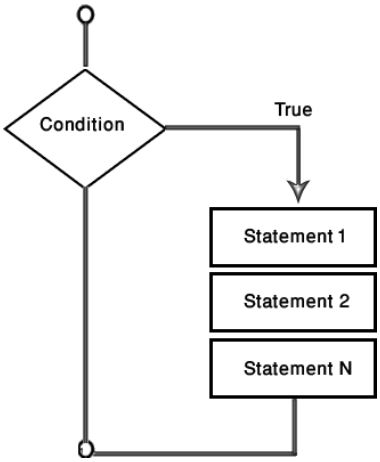


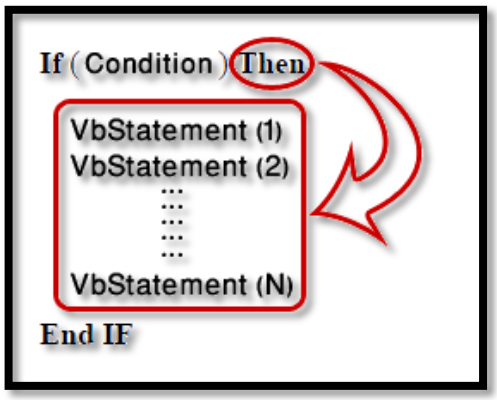
يمكننا أيضاً أن نضيف المزيد من الأوامر داخل كل من الشقين (Then , Else) بحيث أن كل من الصيغتين السابقتين تسمح بوضع أمر واحد داخل كل شق، ونم هنا وجب أن توضع صيغة خاصة تسمح بتعدد الأوامر داخل كل شق.





وكما أسلفنا يمكننا أن نستغني عن الشق (Else) وتكون صيغته كما بالشكل:





Copyrights 2012 ©Mohammad Rasheid

أمثله على جملة (IF Then Else) بصيغها:

```
Const B As Integer = 4, C As Integer = 6
Private Sub Command1_Click()
Dim A As Integer
A = 2
If (x ^ A + B ^ A) - C = 10 Then A = ((C + 4) ^ 2) - 87
MsgBox A
End Sub
```

نبدأ بشرح سريع لآلية التتبع في البرنامج بشكل صحيح، لاحظ في القسم (General) تم وضع (constant) الثوابت والتي تم التطرق لها فيما سبق بأن قيمتها لا تتغير في البرنامج، ثم إنتقلنا إلى الحدث (Command1_Click) وتم تعريف المتغير A من نوع رقم وفي السطر الذي يليه تم وضع قيمة له، أما السطر الثالث كتبنا جملة شرط وكان الشرط (Condition) عباره عن عمليات حسابيه مع مقارنه وبحسب الأولويات الحسابيه سيكون الناتج إما صواب أو خطأ وعليه يتم التنفيذ وبعد الإنتهاء من جمله الشرط سوف تظهر رساله للمستخدم تحوي قيمة المتغير A، هذه نظره سريعه للكود الآن نبدأ بالتتبع.

حتى نتتبع بشكل دقيق أفضل أن نكتب على هامش ورقه الإجابة كلمة (الذاكره) وأن تحوي تحتها جميع المتغيرات وقيمها ونمشي سطر سطر في الكود وفي حال تم تغير قيمة أحد المتغيرات نقوم بحذف قيمة المتغير بالذاكره بوضع شرطة عليه وأن نكتب تحته القيمة الجديده لأن القيمة الجديده للمتغير تلغي القيمة القديمة. كما بالشكل التالي

الذاكرة

A	B	C
~~2~~	4	6
13		

لاحظ أن قيمة كل من (B , C) هي قيم ثابته ولن تتغير قيمها أثناء تنفيذ البرنامج، الآن ننتقل إلى الحدث (Command1) حيث نجد قيمة للمتغير A فنقوم بتخزينها بالذاكره التي كتبناها على الهامش، الآن ندخل إلى جمله الشرط حيث نجد شرط وهو عباره عن عمليات حسابيه ومقارنه فنقوم بحلها بحسب الأولويات الحسابيه فيكون الناتج النهائي هو (True) وبما أن ناتج الشرط صحيح فسيتم تنفيذ الأوامر الموجوده بالشق (Then) وهي عباره عن علمية تعيين قيمة جديده للمتغير A عن طريق عمليه حسابيه فنقوم بإخراج الناتج بحسب الأولويات وتعيين القيمة الجديده مكان القيمة القديمة، ثم ننتقل إلى السطر الأخير وهو عباره عن رساله للمستخدم تحوي قيمة المتغير A وبحسب الذاكره التي كتبناها فإن آخر قيمة للمتغير A هي 13 وعليه ستحوي الرساله الرقم 13.

```
Private Sub Command1_Click()
Dim Password As String
Password = "VB6"
CheckPass = InputBox("Enter password Please:", "Login")
If CheckPass = Password Then MsgBox "Login successful" Else MsgBox "Login fail"
End Sub
```

وهذا الكود هو تمثيل بسيط لآليه عمل برامج التحقق من كلمات المرور مثل برامج الشات لدخولها يجب أن يكون الرقم السري صحيح، ففي هذا المثال نلاحظ بأنه تم تعريف المتغير (Password) من نوع نص وتم تعيين قيمة له (VB6) في السطر الثالث تم عمل صندوق إدخال يطلب بإدخال الرقم السري من المستخدم حيث سيتم تخزينه بالمتغير (CheckPass) ومن ثم ندخل في جمله شرط حيث سيتم التحقق من أن النص المدخل يساوي النص المحفوظ لدينا أي (CheckPass = Password) وفي حال كان النص المدخل صحيح سيتم تنفيذ الأمر بالشق (Then) وإظهار رساله للمستخدم تقول (Login Successful) وفي حال كان النص المدخل لا يساوي الباس فسيتم تنفيذ الأمر بالشق (Else) وإظهار رساله (Login fail).

Copyrights 2012 ©Mohammad Rasheid

```
Private Sub Command1_Click()
A = 5
B = 15
C = 20
If (A ^ 2 > B * C Or B ^ 3 > A * C) Then
X = B * C - A ^ 0
Y = A * C ^ 2 - B
Print X
Print Y
End If
End Sub
```

الذاكره

A	B	C	X	Y
5	15	20	299	1985

في الكود التالي تم تعيين قيم للمتغيرات الثلاثه (A , B , C) ثم إنتقلنا إلى جمله الشرط التي إحتوت على شرط وهو عباره عن عمليات حسابيه وعمليات مقارنه ومعاملات منطقيه ونحلها بحسب الأولويات الحسابيه فيكون الناتج (True) فسينتقل إلى تنفيذ الأوامر التي تلي الشق (Then) والسطرين الأول والثاني تم تعيين قيم للمتغيرات الجديده (X , Y) وأخيراً أمر بطباعه كل من قيم X و Y وعليه يكون الناتج وبحسب الذاكره التي كتبناها وعملنا عليها بالهامش 299 ثم 1985.

```
Private Sub Command1_Click()
Dim ExitMsg As Integer
ExitMsg = MsgBox("Are You Sure to Exit?", vbQuestion + vbYesNo, "Dr91")
If ExitMsg = 6 Then
MsgBox "You Select Yes So Bye :(", vbInformation, "Dr91"
End
Else
MsgBox "You Select No So Welcome Back :)", vbInformation, "Dr91"
End If
End Sub
```

سوف ننهي بهذا المثال الجمله الأولى من جمل الشرط والذي هو عباره عن تطبيق عملي بسيط لآليه رسائل تأكيد الخروج من البرامج، حيث تم تعريف المتغير (ExitMsg) من نوع رقم وفي السطر الذي يليه تم تعيين قيمة لهذا المتغير وهي رساله للمستخدم (النوع الثاني من الرسائل- التي تهتم برد المستخدم وتقوم بالتنفيذ بحسب رده) من نوع سؤال وتحوي الزرين (Yes , No) ثم تأتي جمله الشرط حيث يتم المقارنه هل القيمة العائده من الرساله تساوي 6 أي تم الضغط على الزر (Yes) إذا كان ناتج المقارنه صحيح فسيتم تنفيذ ما بعد الشق (Then) وإظهار رساله تفيد بأنه ضغط على الزر (Yes) ويتم الخروج من البرنامج وفي حال لم يكن ناتج المقارنه صحيح فسيتم تنفيذ الأوامر التي بالشق (Else) وإظهار رساله للمستخدم تفيد بذلك ويتم الرجوع للبرنامج.

Copyrights 2012 ©Mohammad Rasheid

جملة الشرط (Select Case)

يعتبر هذا النوع من جمل الشرط الأكثر إستعمالاً نسبياً بين المبرمجين بحيث يتميز بسرعه التنفيذ.

صيغتها العامة:

```
Select Case VarName (Or arithmetic expression)
Case 1: VBStatement1
Case 2: VBStatement2
..
..
Case N: VbStatementN
End Select
```

بحيث أن:

(Select Case): من كلمات الفيجوال بيسك المحجوزة وتعني تحديد الحاله.

(VarName): هو إسم المتغير الذي نريد أن نقارن قيمته.أو تكون عمليه حسابية.

(Case): من كلمات الفيجوال بيسك المحجوزة وتعني حاله

(VBStatement1): الأمر المراد تنفيذه.

(End Select):من كلمات الفيجوال بيسك المحجوزة وتعني نهاية جمله الشرط.

```
x = InputBox("Enter Number", "JOR")
Select Case x
Case 1 To 10: MsgBox "Your Number Between 1 to 10", vbInformation, "JOR"
Case Is > 10: MsgBox "Enter Number Between 1 to 10", vbExclamation, "JOR"
End Select
```

نلاحظ من هذا المثال بأنه تم تعيين المتغير (x) لتخزين القيمة المدخله من المستخدم في صندوق الإدخال ومن ثم ننتقل إلى جملة الشرط ليقارن في الحاله الأولى هل الرقم المدخل بين 1 و 10 إذا كان كذلك فسيتم تنفيذ الرساله التي تليها مباشره وفي حال لم تكن كذلك فيتم الانتقال الى السطر الذي يليه وهي مقارنه أخرى هل الرقم المدخل اكبر من 10 إذا كان كذلك فسيتم تنفيذ الرساله التي تليها مباشره وفي حال لم تكن كذلك فيتم الانتقال الى السطر الذي يليه وهي عباره عن نهاية جملة الشرط وبذلك لن يتم تنفيذ أي أمر.

تدريب: ما هي قيمة المتغير (tax) بالذاكره بعد تنفيذ البرنامج؟

```
Gross  = 300
Select case  salary
        Case is  < Gross : tax  = 0.03 * salary
        Case is >= gross :      tax = 0.05 * salary
End Select
```


Copyrights 2012 ©Mohammad Rasheid

جملة الشرط (Switch)

صيغتها العامة: Switch(Condition-1,Action-1, condition-2 , Action-2,…,Condition-n, Action-n)

حيث أن:

(Switch): من كلمات الفيجوال المحجوزه وهي بدايه الشرط.

(Condition-1): عمليه مقارنه إذا كانت نتيجه المقارنه صحيحه فإنه ينفذ الأمر الذي بجانبه (Action-1) وإذا كان الناتج غير صحيح فإنه ينتقل إلى المقارنه الثانيه (condition-2).

(condition-2): عمليه مقارنه إذا كانت نتيجه المقارنه صحيحه فإنه ينفذ الأمر الذي بجانبه (Action-2) وإذا كان الناتج غير صحيح فإنه ينتقل إلى المقارنه الثالثه ... وهكذا (Condition-n).

```
Private Sub Command1_Click()
Dim x%
x = 60
MsgBox Switch(x >= 50, "Pass :)", x < 50, "Faild :("), vbInformation, "Dr91"
End Sub
```

في السطر الأول تم تعريف المتغير (x) بأنه من نوع رقم (Integer) وفي السطر الذي يليه تم تعيين قيمة لهذا المتغير وأخيراً لدينا الأمر بإظهار رساله للمستخدم (MsgBox) ولكن إحتوى على جمله الشرط (Switch) حيث سيتم المقارنه هل قيمة المتغير x أكبر من أو تساوي 50 إذا كان ناتج المقارنه صحيح فسيتم إظهار الرساله التي بجانبه وإذا كانت خاطئه فسيتم الإنتقال إلى المقارنه الثانيه هل قيمة المتغير x أقل من أو تساوي 50، وبحسب المثال السابق فسيكون ناتج المخرجات: (Pass ☺)

جملة الشرط (Choose)

صيغتها العامه: Choose (Parameter , Value-1 , Value-2,…..,Value-n)

حيث أن:

(Choose): من كلمات الفيجوال بيسك المحجوزة وتعني بدايه جمله الشرط.

(Parameter): وهو المتغير الذي سيقارن القيمة الخاصه به مع القيم الموجوده وسيقوم بتنفيذ الامر.

- على فرض كان الـ (Parameter) يساوي صفر فستكون النتيجه (Null) لأن الخانات تبدأ بـ 1 وليس 0 وكذلك الحال اذا كان (Parameter) يساوي خانه غير موجوده فستكون النتيجه (Null) ايضاً.

Copyrights 2012 ©Mohammad Rasheid

مثال:

```
Private Sub Command1_Click()
Dim X As Integer
X = 2
Print Choose(X, 2, 5, "Man", 20, "Love")
End Sub
```

في المثال السابق وعند تنفيذ الكود سوف يطبع قيمة (X) الموجوده بالخانه رقم 2 وهي الرقم 5، ولو فرضنا بأن قيمة X هي 3 فسيطبع Man وهكذا، أما إذا كانت قيمة X غير موجوده فسيقوم بطباعه Null.

تدريب: ما ناتج البرنامج التالي:

- 1

```
Dim X As Integer
X = 0
Print Choose(X, 1, 2, 3, 4, 5, 6)
```

- 2

```
Dim X As Integer
X = 4
MsgBox Choose(X, 6, 34, 0, 76, 56, 65), vbInformation, "Dr91"
```

⭐⭐The End of Chapter Three⭐⭐

Copyrights 2012 ©Mohammad Rasheid

visual basic 6

chapter four

INDEX

Copyrights 2012 ©Mohammad Rasheid

جمل التكرار والدوران

قد تحتاج أحياناً وأثناء برمجة برنامجك إلى تكرار أمر أو مجموعه أوامر عده مرات فهل يعقل أن تعيد كتابة هذه الأوامر في كل مره حتى تحصل على التكرار المطلوب ؟ بالتأكيد لا ومن هنا جائت الفيجوال بيسك لتقدم لنا جمل التكرار التي تسهل علينا الوقت والجهد وتقوم بتكرار ما نريد وبحسب عدد المرات التي نريد وهذه الجمل هي:

1. For ... Next
2. Do While ... Loop
3. Do ... Loop While
4. Do Until ... Loop
5. Do ... Loop Until
6. While ... Wend

جملة التكرار (For ... Next)

صيغتها العامة:

```
For Index = start To end Step [step]
Statement(s) to be repeated each loop
Next Index
```

حيث أن:

(For): من كلمات الفيجوال بيسك المحجوزة وتعني بداية التكرار.

(Index): هو إسم المتغير الذي نريد ويعتبر كعداد.

(Start): الرقم الذي نريد أن يبدأ التكرار العد منه.

(To): من كلمات الفيجوال بيسك المحجوزة وتعني إلى.

(End): الرقم الذي نريد أن يتوقف التكرار عنده.

(Step): من كلمات الفيجوال بيسك المحجوزة وتعني مقدار الخطوة التي نريد.

([Step]): هو مقدار الخطوة التي نريد وفي حال لم نحدد يكون إفتراضياً 1.

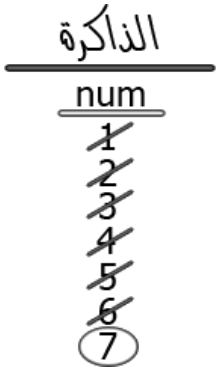
(Next): من كلمات الفيجوال بيسك المحجوزة وتعني العودة إلى بداية التكرار بعد أن يكون قد أضاف مقدار الـ (Step) إلى المتغير الذي بجواره.

مثال: جد ناتج التطبيق التالي:

```
Private Sub Command1_Click()
Dim num As Integer
For num = 1 To 6
Print "**-";
Next num
End Sub
```


Copyrights 2012 ©Mohammad Rasheid

لاحظ في السطر الأول تم تعريف المتغير (num) من نوع رقم (Integer) وفي السطر الذي يليه تم الدخول إلى التكرار الخاص بجمله (For … Next) وتم تعيين القيمة 1 للمتغير (num) كقيمة بداية ثم دخل إلى جسم التكرار ليقوم بطباعه النص (-**) على جسم النموذج ومن ثم إنتقل إلى السطر الذي يعيده إلى بدايه التكرار ولكن بعد إضافه قيمة الـ (Step) إلى المتغير وهنا بما أننا لم نحدد قيمة الـ (Step) فإن قيمته الإفتراضيه هي (1) وعليه يكون قيمة المتغير (num) قبل هي (1) وقيمته بعد الزياده هي (2)... وهكذا إلى أن يكون الناتج هو كما بالشكل وقيمة المتغير (num) بالذاكره هي: (7)



تدريب ما ناتج كل من البرامج التاليه:

```
Private Sub Command1_Click()
Dim index As Integer
For index = 2 To 8 Step 2
Print "Jordan First"
Next index
MsgBox index
End Sub
```

```
Private Sub Command1_Click()
Dim dede As Integer
For dede = 1 To 5 Step 1
Print dede & "- Jordan First"
Next dede
End Sub
```

```
Private Sub Command1_Click()
Dim Ra As Integer
For Ra = 10 To 5 Step -1
Print "BAU - Salt City"
Next Ra
End Sub
```


Copyrights 2012 ©Mohammad Rasheid

جملة التكرار (Do While ... Loop)

صيغتها العامة:

```
Do While condition
'Statement(s) to be executed as long as the condition is true
Loop
```

حيث أن:

(Do): من كلمات الفيجوال بيسك المحجوزة وتعني بداية التكرار "نفذ" .

(While): من كلمات الفيجوال بيسك المحجوزة وتعني عندما.

(Condition): الشرط أو عمليه المقارنه أي إذا كانت عمليه المقارنه صحيحه فسيدخل إلى جمله التكرار وإذا كانت خاطئه فلن يدخل إلى جسم التكرار.

(Statement): الأوامر التي سينفذها إذا دخل إلى جسم التكرار.

(Loop): من كلمات الفيجوال بيسك المحجوزة وتعني تكرار أو العوده إلى بدايه التكرار.

مثال: ما عدد المرات التي ستظهر فيها رساله المسج بوكس ؟

```
Private Sub Command1_Click()
Dim test As Integer
Do While test < 10
MsgBox "Dr91 ;) "
test = test + 1
Loop
End Sub
```

في السطر الأول تم تعريف المتغير (test) من نوع رقم (Integer) ، أما السطر الثاني فقد دخلنا إلى جمله التكرار (Do while ... Loop) وفيها عمليه مقارنه إذا كانت قيمة المتغير (test) أقل من 10 فدخل إلى جسم التكرار ونفذ الأوامر التي فيه وإلا أخرج من جسم التكرار وبما أننا لم نحدد قيمة المتغير (test) فإن قيمته الإفتراضيه 0 وبحسب جمله المقارنه (test < 10) والصفر أقل من 10 فسيدخل إلى جمله التكرار وينفذ الجمله الأمر الأول وهو رساله للمستخدم تحوي (Dr91 ;) ثم ينتقل إلى السطر الذي يليه وهو عباره عن أمر لزياده قيمة المتغير (test) في كل عمليه دوران لأننا لو لم نفعل ذلك فستكون الرساله لا نهائيه التكرار، وهنا ستصبح قيمة المتغير (test) تساوي 1 وينتقل مجدداً إلى بداية التكرار ويدخل من جديد في عمليه المقارنه وسيصبح السؤال هل قيمة المتغير (test) الجديده أقل من 10 إذا كانت الإجابه نعم فسيدخل إلى جسم التكرار وإذا لم تكن مساويه فسيخرج من التكرار وهكذا إلى أن ينتهي البرنامج وعليه ستظهر رساله المسج بوكس 10 مره وستكون قيمة المتغير (test) النهائيه هي: (10).

الذاكره

test

~~0~~
~~1~~
~~2~~
~~3~~
~~4~~
~~5~~
~~6~~
~~7~~
~~8~~
~~9~~
10

Copyrights 2012 ©Mohammad Rasheid

تدريب ما ناتج كل من البرامج التاليه:

```
Private Sub Command1_Click()
Dim size As Integer
Dim counter As Integer
size = 100
Do While size <= 1000
counter = counter + 1
size = size + 100
Print " Done ! "
Loop
End Sub
```

```
Private Sub Command1_Click()
Dim X As Integer
X = 0
Do While X < 10
X = X + 1
Print X
If X = 5 Then Exit Do  في حال تحقق الشرط يتم الخروج من جسم التكرار '
Loop
End Sub
```

```
Private Sub Command1_Click()
Dim معقول As Integer
معقول = 12
Do While معقول Mod 2 = 0
معقول = معقول - 2
Print معقول
If معقول <= 0 Then Exit Do   في حال تحقق الشرط يتم الخروج من جسم التكرار '
Loop
End Sub
```

```
Private Sub Command1_Click()
Dim n%
If 5 + 6 ^ 0 * 9 = 14 Then
Y = 14
Do While Y > 5
Print "Nice"
Y = Y - 2
Loop
Else
For X = 1 To 5
Print "Try A gain"
Next X
End If
End Sub
```


Copyrights 2012 ©Mohammad Rasheid

جملة التكرار (Do ... Loop While)

صيغتها العامة:

```
Do
 Statement(s) to be executed as long as the condition is true
Loop While condition
```

حيث أن:

(Do): من كلمات الفيجوال بيسك المحجوزة وتعني "نفذ مباشره" .

(Statement): الأوامر التي سينفذها داخل جسم التكرار.

(Loop): من كلمات الفيجوال بيسك المحجوزة وتعني تكرار أو العوده إلى بدايه التنفيذ والتكرار.

(While): من كلمات الفيجوال بيسك المحجوزة وتعني عندما.

(Condition): الشرط أو عمليه المقارنه أي إذا كانت عمليه المقارنه صحيحه فسيدخل إلى جمله التكرار وإذا كانت خاطئه فلن يدخل إلى جسم التكرار.

مثال ما ناتج البرنامج التالي:

```
Private Sub Command1_Click()
Do
Print "JOJO"
x = x + 1
Loop While x < 3
End Sub
```

في السطر الأول إبتدء البرنامج بالأمر (Do) أي نفذ مباشره ما يوجد تحتي من أوامر ، فينتقل للتنفيذ فيجد أمر بطباعه النص (JOJO) على النموذج وبلبه عمليه تعيين قيمة جديده للمتغير (x) بحيث يتم إضافه 1 في كل دوره تكرار وأخيراً يتوجه لـ (Loop While) حيث سيتم إعاده التكرار في حال كانت نتيجه عمليه المقارنه صحيحه وإلا فسوف يخرج من التكرار مباشره. وعليه سوف يطبع النص ثلاث مرات وتكون القيمة النهائيه للمتغير (x) بالذاكره تساوي 3.

جملة التكرار (Do Until ... Loop)

صيغتها العامة:

```
Do Until Condition
'Statement(s) to be executed as long as the condition is false
Loop
```

حيث أن:

(Do): من كلمات الفيجوال بيسك المحجوزة وتعني "نفذ" .

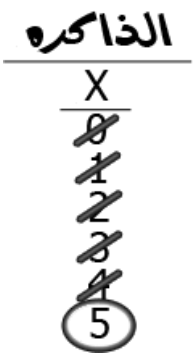
(Until): من كلمات الفيجوال بيسك المحجوزة وتعني عندما لا.

(Condition): الشرط أو عمليه المقارنه.

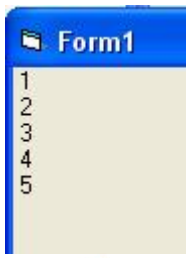
Copyrights 2012 ©Mohammad Rasheid

مثال ما ناتج البرنامج التالي:

```
Private Sub Command1_Click()
Dim X As Integer
X = 0
Do Until X > 10
X = X + 1
Print X
If X = 5 Then       ' الخروج من جسم التكرار في حال تحقق الشرط
Exit Do
End If
Loop
End Sub
```



في السطر الأول تم تعريف المتغير (X) من نوع رقم والسطر الثاني تم تعيين قيمة للمتغير (Integer) قيمتها 0 وفي السطر الذي يليه تم الدخول إلى جسم التكرار للجمله (Do Until ... Loop) وهنا يكون السؤال نفذ عندما لا تكون قيمة المتغير (X) أكبر من 10، بلغه أبسط (نفذ إذا كانت نتيجه المقارنه خاطئه False)، وبما أن 0 ليس أكبر من 10 أي نتيجه المقارنه خاطئه فسيتم الدخول إلى جسم التكرار ويبدأ بتنفيذ الأوامر التي بداخله وبحسب الأمر الأول فقد تم شرحه سابقاً ثم أمر بطباعه قيمة المتغير (X) وأخيراً جمله شرط تتأكد بأن قيمة المتغير (X) تساوي 5 أم لا وفي حال كانت مساويه فسيتم الخروج من التكرار وإلا فسيكمل التكرار ثم ينتهي بـ (Loop) التي تعيده مجدداً إلى بدايه جمله التكرار وهكذا حتى يخرج من الجمله وينهي تنفيذ البرنامج. وعليه ستكون النتيجة على شاشه المخرجات هي كما بالشكل التالي وتكون قيمة المتغير (X) بالذاكره تساوي .



جملة التكرار (Do ... Loop Until)

صيغتها العامة:

```
Do
'Statement(s) to be executed as long as the condition is false
Loop Until Condition
```

مثال:

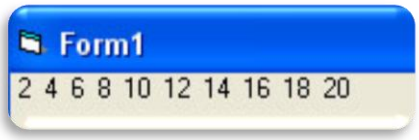
```
Private Sub Command1_Click()
Dim I As Integer
I = 2
Do
Print I;
I = I + 2
Loop Until I > 20
End Sub
```

في السطر الأول تم تعريف المتغير (I) من نوع رقم (Integer) وفي السطر الثاني تم تعيين قيمة للمتغير (I) تساوي 2، ثم دخلنا إلى جسم التكرار وبدأ بالتنفيذ الأوامر حيث سيتم طباعه قيمة المتغير على شاشه النموذج ثم ينتقل الى السطر الذي يليه لزياده قيمة المتغير ثم ينتقل

Copyrights 2012 ©Mohammad Rasheid

أخيراً إلى (Loop Until) حيث يكون السؤال أعد التكرار عندما لا تكون قيمة المتغير (I) أكبر من 20 أو بلغه أبسط إذا كانت نتيجه المقارنه خاطئه أعد التكرار وإذا كانت صحيحه أخرج من التكرار.

وعلى ما سبق ستكون نتيجه المخرجات:



جملة التكرار (While ... Wend)

صيغتها العامة:

```
While condition
'Statement(s) to be executed as long as the condition is true
Wend
```

حيث أن:

(While): من كلمات الفيجوال بيسك المحجوزة وتعني "عندما" .

(Condition): الشرط أو عمليه المقارنه.

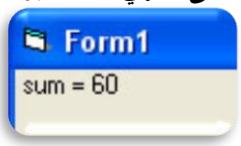
(Wend): من كلمات الفيجوال بيسك المحجوزة وتعني نهايه جمله التكرار والعوده إلى بدايتها.

مثال: جد ناتج البرنامج التالي

```
Private Sub Command1_Click()
sum = 10
While sum <= 50
sum = sum + 10
Wend
Print " sum = " & sum
End Sub
```

في السطر الأول تم تعيين قيمة للمتغير (Sum) وهي 10 ثم دخلنا إلى جمله التكرار (While ... Wend) حيث وجدنا الشرط يقول نفذ إذا كانت قيمة المتغير أقل من أو تساوي 50 وبما أن 10 أقل من 50 فسيدخل إلى جسم التكرار لينفذ الأوامر الموجوده ليجد أمر تعيين قيمة جديده للمتغير +10 وثم نهايه جمله التكرار ليعود به الى البدايه ليصبح السؤال هل 20 أقل من أو تساوي 50 إذا نعم سيدخل التكرار للتنفيذ وإلا سيوقف التكرار وهكذا حتى ينتهي تنفيذ البرنامج.

وعليه سيكون الناتج:



⭐⭐ The End of Chapter Four ⭐⭐

Copyrights 2012 ©Mohammad Rasheid

visual basic 6

chapter five

INDEX

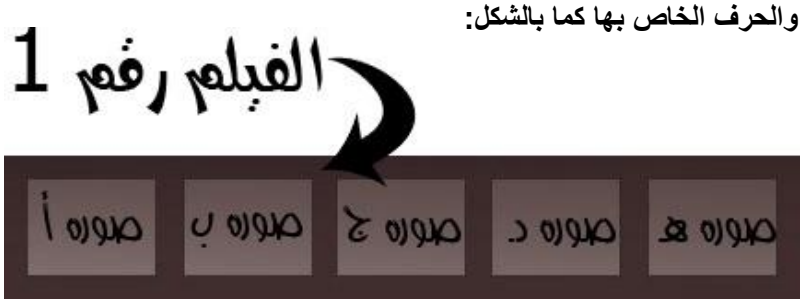
Copyrights 2012 ©Mohammad Rasheid

Arrays

ما هي المصفوفه ؟

قبل البدء بتعريف المصفوفه وما هي دعنا نحاول تصور المصفوفه حتى يكون التعريف أبسط وأوضح، وعليه يمكنني تشبيه المصفوفه بشريط الفيلم الخاص بكاميرا التصوير قديماً، لاحظ الشكل التالي:

لاحظ أن الفيلم واحد ولكن يحوي بداخله الكثير من المقاطع وفي كل مقطع توجد فيه صوره، فلو أطلقنا على الفيلم ككل إسم الفيلم رقم 1، وكل مقطع تم تسميته بـ صوره والحرف الخاص بها كما بالشكل:



لاحظ عندما نريد أن تنظر إلى صوره معينه لن نقول أنظر الفيلم رقم 1 لماذا؟ لأنه يحوي بداخله الكثير من المقاطع (الصور) ولكن نقول أنظر الفيلم رقم 1 الصوره ج، فلاحظ هنا قمت بتحديد إسم الفيلم والذي يمثل هنا المصفوفه ومن ثم تحديد الخانه التي أريد وهي (Element).

هذا هو تقريباً المثال الأقرب للمصفوفه، وعليه نقول المصفوفه هي: عباره عن متغير يحوي مجموعه من العناصر في الذاكره تتشابه في الإسم (إسم المصفوفه) وتختلف في إسم العنصر (رقمه) ويمكننا إن نعرف المصفوفه كما نريد كما هو الحال بالنسبه للمتغيرات وعليه لو قمنا بتعريف المتغير X مثلاً بأنه مصفوفه من النوع الرقمي (Integer) فسيقوم الفيجوال بيسك مباشره بتعريف كل العناصر بأنها من نوع رقمي لماذا؟ لأن المصفوفه الرئيسيه التي يتبع لها هذا العنصر من نوع رقمي .

وتعبر المصفوفات حالياً هي الأكثر إستعمالاً بين محترفي البرمجه بحث تسهل التعامل والتعريف فلو قلنا أننا بحاجه إلى تعريف 1000 متغير من نوع نص (String) فهل يعقل أن تبدأ بتعريفهم متغير متغير ؟ بالتأكيد لا فهو أمر مرهق وغير منطقي أبداً ومن هنا وجدت الفيجوال بيسك أن الأفضل للمبرمجين هو وجود المصفوفات، فعندما نقول بأننا سنعرف متغير واحد يحوي 1000 عنصر بأنه من نوع نص ستقول أن هذا هو المنطق وبعدها تقوم بتعيين ما تريد من قيم لهذه العناصر.

تعريف المصفوفه:

إذا أردنا أن نتعلم كيف نقوم بكتابه تعريف المصفوفه فلا بد لنا من معرفه الصيغه العامة لها وهي :

```
Dim Array_Name(value) As DataType
```


Copyrights 2012 ©Mohammad Rasheid

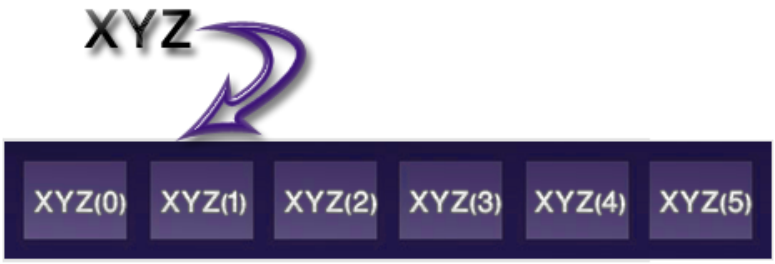
وبناء على الصيغه السابقه إذا أردنا تعريف المتغير XYZ بأنه مصفوفه مكونه من خمس عناصر من نوع نص (String) فسنقول:

```
Dim XYZ(5) As String
```

ومن الأمثله على المصفوفات أيضاً:

```
Dim P_Name(40) As String

Dim P_Number(10) As Integer

Dim P_Email(23) As String

Dim P_Married(9) As Boolean

Dim BAU(30) As String
```

بالرجوع إلى المثال أعلاه (XYZ) إذا أردنا أن نرى تمثيل له بالذاكره فسيكون كالتالي:



قد تتسائل الآن من أين جائت XYZ(0) سأقول لك بأن الفيجوال بيسك يبدأ العد من 0 إلى الرقم الذي قمت بتحديده وعليه يكون:

عدد عناصر المصفوفه = العدد الذي قمت بتحديده + 1

وعند تطبيق هذه القاعده في المثال أعلاه ستلاحظ أن عدد العناصر 5 + 1 = 6، وفعلا إذا قمت بعد عناصر المصفوفه الموجوده بالشكل السابق ستلاحظ أنها 6. أي لو أنك أردت أن تعرف مصفوفه تحوي 5 عناصر فقط وفعليه وبحسب ماقد شاهدناه في المثال أعلاه حيث أن الفيجوال بيسك يبدأ العد من 0 فسيكون التعريف :

```
Dim XYZ(4) As String
```

وعند تطبيق قاعده معرفه عدد العناصر سيكون 4 + 1 = 5 وهو المطلوب.

Copyrights 2012 ©Mohammad Rasheid

الخاصيه Option base

تستخدم هذه الخاصيه الرائعه من أجل إجبار الفيجوال بيسك بأن يبدأ العد من الرقم 1 أو الرقم 0، فكما لاحظنا بأن الفيجوال بيسك يبدأ من 0 إلى الرقم الذي قمت بتحديده، وقد قمنا بكتابه قاعده معرفه عدد العناصر وقلنا هي: عدد عناصر المصفوفه = العدد الذي قمت بتحديده + 1 جميل، ولكن هل يعقل في كل مره أريد أن أعرف مصفوفه يجب أن آخذ بعين الإعتبار هذه الملاحظة فلو فرضنا بأنني أريد كتابه برنامج يقوم بإدخال أسماء عمال في مصنع وعلى فرض 100 عامل، فهل يعقل عند التعريف أن أكتب Dim P_Name(99) As String قد تقول نعم ولكن تخيل لو أنك بحاجة إلى أكثر من مصفوفه هل في كل مره ستقوم بطرح 1 من القيمة التي تريد ؟ بالتأكيد هذا كلام غير منطقي!

ومن هنا جائت هذه الخاصيه التي نقوم بكتابتها في القسم (General) ونحدد من أين يبدأ الفيجوال بيسك العد هل من 1 أو من 0 وبالأغلب تكون 1. وصيغتها هي:

```
Option Base N
```

وقيمة (N) هنا هي (1) أو (0).

رأينا كيف يمكننا أن نقوم بتعريف المصفوفه ورأينا أيضاً كيف يمكننا أن نستخدم الخاصيه (Option Base) لتحديد العنصر الذي سنجبر الفيجوال بيسك بالعد منه، وسنرى الآن طريقه أخرى لتعريف المصفوفه والتي يمكننا من خلالها أن نحدد البدايه أي وبلغه أخرى أن نحدد العنصر الذي نريد أن يبدأ العد من خلاله ولكن هذه المره ليس 1 أو 0 بل أي عدد أريد، إلى الآن أنا متأكد بأنك لم تستطع أن تفهم ما أقول ولكن لا تقلك فإذا شعرت بذلك فهذا خطأي وليس خطأك وحتى نستطيع تصحيح هذا الخطأ لابد لنا من معرفه الصيغه العامة للتعريف الجديد:

الصيغه العامة:

```
Dim Array_Name(Start to End) as String
```

حيث أن:

(Start): هي قيمة البدايه التي سيقوم الفيجوال بيسك بالعد والقراءه منها.

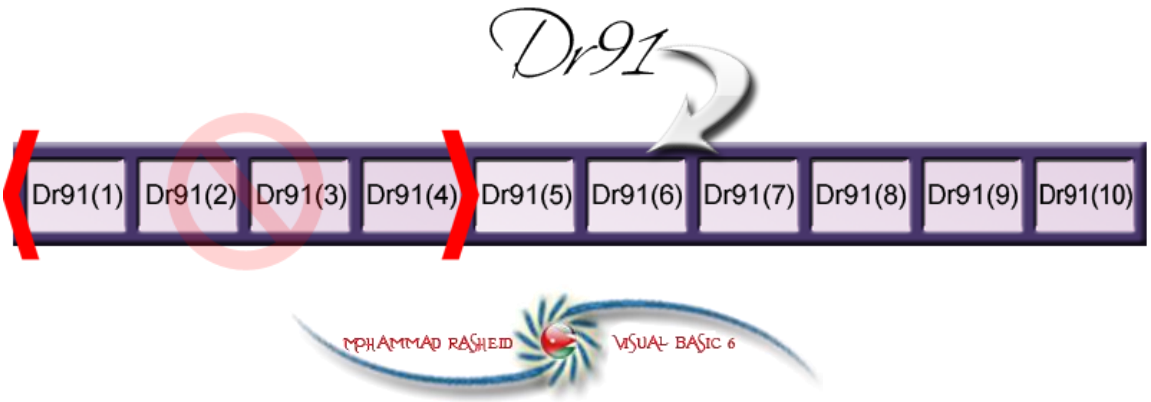
(End): هي قيمة النهايه التي سيقوم الفيجوال بيسك بالتوقف عندها.

Copyrights 2012 ©Mohammad Rasheid

مثال عليها:

```
Dim Dr91(5 To 10) As Integer
```

وهنا نلاحظ بأنه تم تعريف المصفوفه (Dr91) بأنها من نوع رقمي (Integer) وتبدأ من العنصر (5) وتنتهي بالعنصر (10) وهذا يعني بأن جميع العناصر التي تأتي قبل العنصر الذي قمنا بتحديده (العنصر 5) هي قيم محذوفه وهي خارج المدى والشكل التالي يوضح هذه الفكره



تعيين قيم لعناصر المصفوفه:

لابد لك بأنك تتذكر كيف كنا نقوم بتعيين قيم للمتغيرات حيث كنا نقوم بكتابه إسم المتغير ثم إشاره المساواة وبعدها نضع قيمة هذا المتغير كما بالصيغه التاليه:

```
VarName = NewValue
```

وتعيين قيمة للمصفوفه لا يختلف عن تعيين قيمة للمتغير العادي بحيث أنك تقوم بكتابه إسم المصفوفه وتحدد رقم العنصر المراد تعيين قيمة له بين قوسين ومن ثم إشاره المساواة وبعدها قيمة هذا المتغير. كما بالصيغه التاليه:

```
Array_Name(Value) = NewValue
```

ومن الأمثله عليها:

```
Dr91(5) = 1991

P_Name(35) = “Mohammad Rasheid”

P_Married(7) = Flase

Dr91(9) = Dr91(5)
```


Copyrights 2012 ©Mohammad Rasheid

الخاصيه (LBound) و (UBound):

تستخدم هذه الخاصيه لتحديد قيمة البداية والنهايه للمصفوفه التي تريد، وصيغتها العامة:

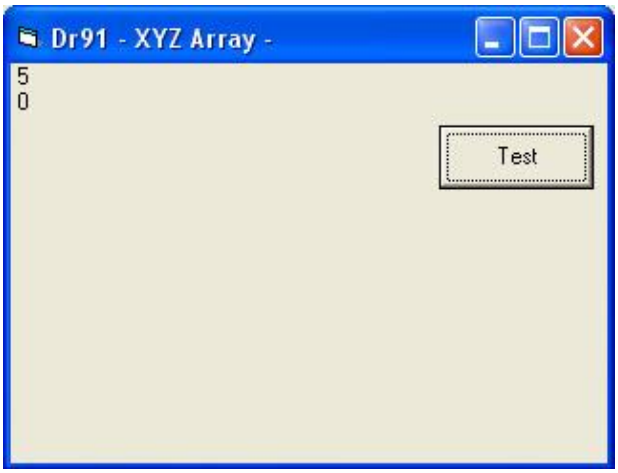
```
LBound (Array_Name)
UBound (Array_Name)
```

ولتطبيق هذا الكلام على المصفوفه (XYZ) التي مررنا بها سابقاً سنأخذ المثال التالي:

```
Private Sub Command1_Click()
Dim XYZ(5) As String
Print UBound(XYZ)
Print LBound(XYZ)
End Sub
```

بحيث تم تعريف المصفوفه (XYZ) بأنها مصفوفه مكونه من 5 عناصر من نوع نصي وفي السطرين التاليين طلبنا أن يقوم بطباعه (LBound) و (UBound) للمصفوفه والشكل التالي يوضح المصفوفه في الذاكره وكيف تم تحديد (LBound) و (UBound)

لاحظ ومن الشكل أعلاه يجب أن يقوم بطباعه 0 والتي تمثل (LBound) و 5 التي تمثل (UBound)، وحتى نتأكد من ذلك قمنا بتطبيق البرنامج وعند تشغيله والضغط على زر الأمر طبع ما يلي:



وهذا يؤكد المبدأ الذي قمنا بإتباعه.

ملاحظة: يتم إستخدام (LBound) و (UBound) أيضاً مع جمل التكرار وسيتم تناول أمثله كافيه على هذا الموضوع بعد إنتهاء الدرس.

Copyrights 2012 ©Mohammad Rasheid

أنواع المصفوفات:

للمصفوفات أنواع كثيره منها المصفوفه ذات البعد الواحد وهي ما تمت دراستها قبل قليل والمصفوفه ذات الأبعاد المتعدده وهي ما سيتم تناولها بعد قليل، لا تختلف كثيراً عن المصفوفه ذات البعد الواحد ولكن ما يهمك هنا سواء كانت المصفوفه أحاديه أو متعدده، ثلاث أشياء:

1 - ورقه وقلم (أي لا تقم بتتبع البرنامج ذهنياً مهما كان مستواك !)

2 - الذاكره (طريقه ملكيه من وجه نظري!)

3 - التركيز

المصفوفه ذات الأبعاد المتعدده (Multi dimensional arrays):

يمكننا تصنيف المصفوفات ذات الأبعاد إلى نوعين: مصفوفه ذات البعدين وهي ما سوف ندرسه الآن ونقوم بشرحه، والمصفوفه المتعدده وتشمل (المصفوفه ذات الثلاث أبعاد فأكثر).

المصفوفه ثنائيه البعد (Two dimensional arrays)

ويمكننا تشبيهها بجدول له صفوف وأعمده ولتكن على فرض 4 أعمده و 3 صفوف، فتقاطع كل عمود مع صف يكون خليه هذه الخليه يتم تسميتها بإسم الصف والعمود أي مثلاً: الخليه(الصف 4 مع العمود 2) وهي أشبه بنظام (Microsoft Excel).

تعريف المصفوفه الثنائيه:

تعريف المصفوفه الثنائيه لا يختلف عن تعريف المصفوفه الأحاديه، ولكن في الثنائيه نقوم بتحديد عدد الصفوف ومن ثم فاصله وعدد الأعمده وصيغتها العامة للتوضيح هي:

```
Dim Array_Name(Rows_Number, Coulmns_Number) As DataType
```

حيث أن:

(Array_Name): هي إسم المصفوفه التي نريد.

(Rows_Number): عدد الصفوف، ولا تنسى "الفيجوال بيسك يبدأ بالعد من الصفر مالم أحدد غير ذلك".

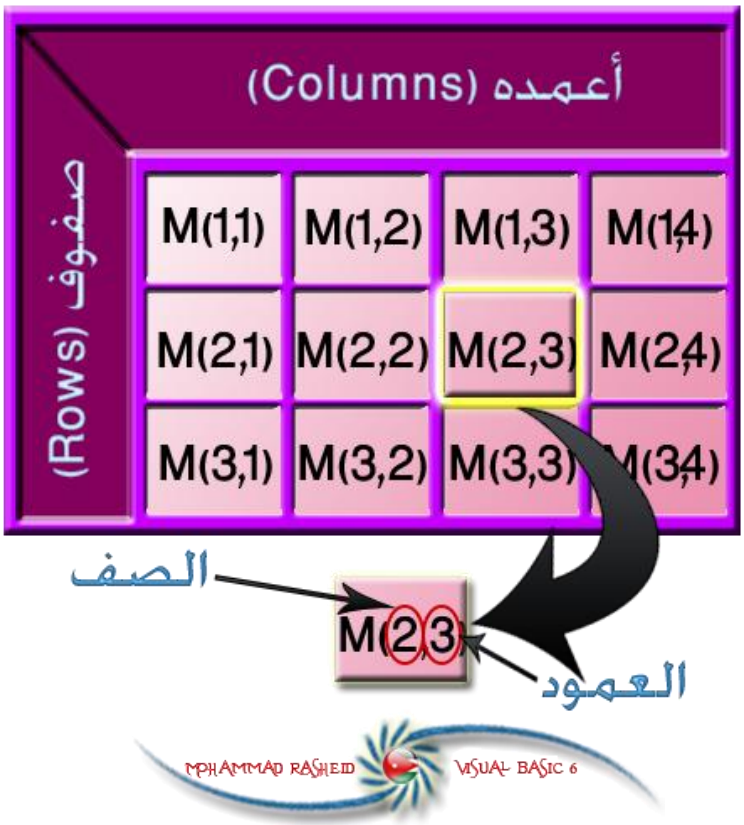
(Coulmns_Number):عدد الأعمدة، ولا تنسى "الفيجوال بيسك يبدأ بالعد من الصفر مالم أحدد غير ذلك".

Copyrights 2012 ©Mohammad Rasheid

وحتى نوضحها أكثر وجب علينا أخذ مثال، فلو أردنا تعريف المتغير (M) من نوع رقمي (Integer) بأنه مصفوفه ثنائيه البعد تتكون من 3 صفوف و 4 أعمده. فسيكون كود التعريف هو:

```
Option Base 1
Private Sub Command1_Click()
Dim M(3, 4) As Integer
End Sub
```

الـ(Option Base 1) لإجبار الفيجوال بيسك بالعد من الواحد. وعند تمثيلها بالذاكره ستكون كما بالشكل التالي:



MOHAMMAD RASHEID VISUAL BASIC 6

تعيين قيمة لعناصر المصفوفه ثنائيه البعد:

وتعيين قيمة للمصفوفه لا يختلف عن تعيين قيمة لعناصر المصفوفه الأحاديه البعد بحيث أنك تقوم بكتابه إسم المصفوفه وتحدد رقم الصف والعمود التي تتواجد فيه الخلية المراد تعيين قيمة لها بين قوسين وتفصل بينهما فاصله ومن ثم إشاره المساواة وبعدها قيمة هذا المتغير. كما بالصيغه التاليه:

Array_Name(Row_Num , Coulmn_Number) = NewValue

فلو أردنا أن نقوم بتعيين قيمة للمتغير بالشكل السابق فسيكون الكود هو:

M(2,3) = 1991

MOHAMMAD RASHEID VISUAL BASIC 6

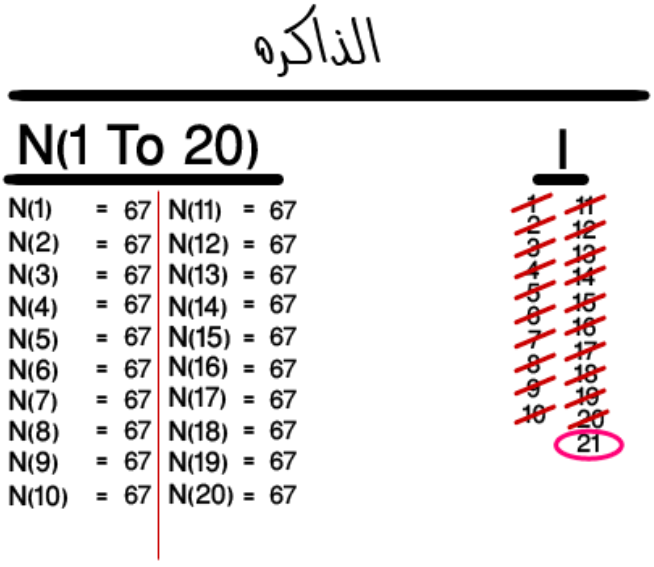
Copyrights 2012 ©Mohammad Rasheid

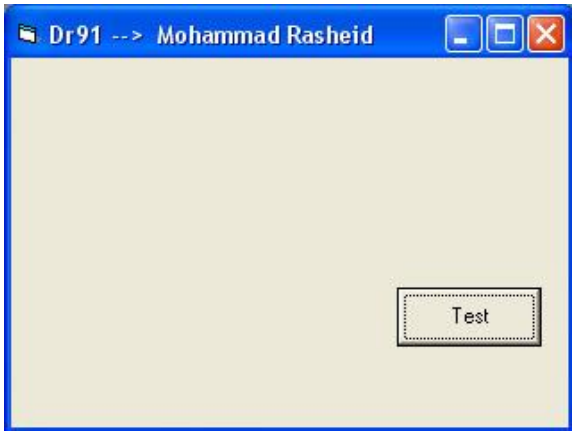
إلى هنا وأكتفي بالمعلومات التي قمت بشرحها حييث أنني لا أقوم بشرح التفصيلي لأني على ثقه بأن لديك المعلومة من الدكتور المشرف عليك وما هذه الشروحات إلى لتثبيت ما قام دكتورك بشرحه، أتمنى أن أكون قد وفقت بإيصال وشرح ولو جزء بسيط من هذه الماده بالشكل المناسب وأختم أطروحتي بمجموعه من الأسئله ستقسم إلى قسيمين أسئله مشروحه (أي سنقوم بشرحها سوياً) وتدريبات أتركها لك لتختبر وتقيّم نفسك بها.

أسئله مشروحه:

```
Private Sub Command1_Click()
Dim N(1 To 20) As Integer
Dim I As Integer
For I = 1 To 20
N(I) = 67
Next I
End Sub
```

نبدأ بالمثال الأول، السطر الأول قمنا بتعريف المتغير N بأنها مصفوفه من نوع رقمي (Integer) ومداها من 1 الى 20، في السطر الثاني تم تعريف المتغير I بأنه متغير رقمي (Integer)، في السطر الثالث تم الدخول إلى جمله التكرار (For)، تم تعيين قيمة ابتدئيه للمتغير I (1) فنذهب مباشره ونرسم الذاكره على هامش الورقه ونسجل أول بأول قيم المتغيرات وكما تعلمنا القيمة الجديده تلغي القيمة القديمة، الآن ندخل إلى جسم التكرار نلاحظ أمر بتعيين قيمة للمتغير N(I) ولكن ماذا نعني بـ N(I) ؟ أي عوض مكان المتغير I الموجود بين قوسين قيمة المتغير I ثم عين القيمة له، طيب بروح وبنسأل حالي شو قيمة المتغير I؟ إرجع للذاكره عالسريع راح إتشوف أنه قيمه المتغير I هي القيمة الابتدائيه إلي قمنا بتدوينها قبل قليل وهي 1، مباشره بعوضها فتصبح N(1) = 67 حلو أصبحت واضحه الآن وهي عباره عن أمر بتعيين قيمة للعنصر N(1) بالمصفوفه N إذهب للذاكره وسجل أن قيمة هذا العنصر تساوي 67 تمام، نكمل إنتقل للسطر الذي يليه سنجد Next I أي أضف للمتغير I قيمة الـ Step ومن ثم إرجع إلى بدايه جمله التكرار فتصبح قيمة المتغير I الجديده = قيمة المتغير I القديمة + مقدار الـ Step وعند التطبيق ستصبح قيمة المتغير I تساوي 2 فنذهب للذاكره ونحذف 1 ونضع مكانه القيمة الجديده للمتغير وهي 2، أكمل على هذا المسار حتى تصبح قيمة المتغير I تساوي 21 ويخرج من جمله التكرار ولاحظ الذاكره بعد إنتهائك منها أن جميع عناصر المصفوفه N تحمل نفس القيمة وذلك لأننا بالسؤال جعلنا الفيجوال بيسك يقوم بتعيين قيمة ثابته للجميع فلو أردنا أن نرى ناتج شاشه المخرجات للمثال السابق فسيكون كما بالشكل التالي:





لاشئ لماذا؟ لأننا لم نضف بالكود السابق أي أمر لإخراج النتائج ^.^ فسيكون الجواب هو (Nothing)، سؤال رائع لف لف لف وفي النهايه (Nothing).

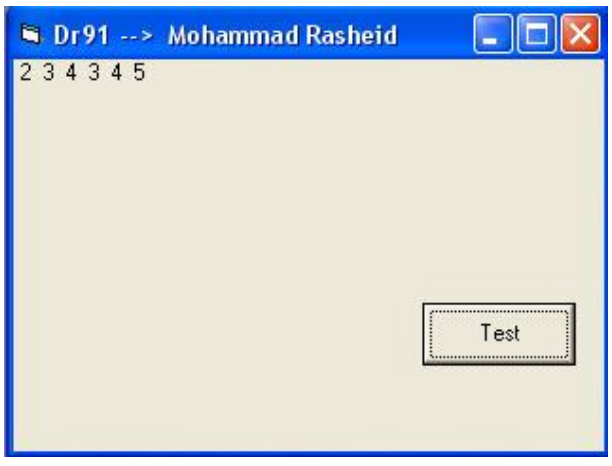
Copyrights 2012 ©Mohammad Rasheid

```
Private Sub Command1_Click()
Dim M(1 To 2, 1 To 3) As Integer
Dim X%, Y%
For X = 1 To 2
  For Y = 1 To 3
  M(X, Y) = X + Y
  Print M(X, Y);
  Next Y
Next X
End Sub
```

نبدأ بالمثال الثاني،لاحظ معي في السطر الأول تم تعريف المصفوفه M بأنها مصفوفه ثنائيه البعد من نوع رقمي (Integer) تحوي على صفين و عمودان وفي السطر الذي يليه تم تعريف المتغيران X و Y أيضاً بأنهما من نوع رقمي (Integer)، ندخل الآن الى السطر الثالث بحيث يحوي على جمله تكرار ندخل إليها لنجد بأنه تم تعيين قيمة للمتغير X هي 1 فنذهب للذاكره ونسجل ثم ندخل إلى جسم التكرار فنجد جمله تكرار أخرى ! وقد تم تعيين قيمة إبتدائيه للمتغير Y تساوي 1 أيضاً فنذهب للذاكره ونسجل ثم ندخل إلى جسم التكرار الثاني لنجد أمر بتعيين قيمة لعنصر المصفوفه M وهو M(X,Y) = X +Y لمعرفه العنصر المطلوب وناتج العمليه قم بتعويض المتغيرات الموجوده لتصبح M(1,1) = 1+1 حلو أي عين للعنصر M(1,1) ناتج العمليه 1+1 أي 2 فنذهب للذاكره ونسجل M(1,1) = 2 ، ننتقل إلى السطر الذي يليه لنجد أمر بطباعه عنصر من المصفوفه M(X,Y); كما إتفقنا قم بتعويض المتغيرات لتصبح M(1,1); وبحسب ما قد قمنا بتسجيله بالذاكره سيطبع 2 وبما أن أمر الطباعه إنتها بالفاصله المنقوطة فستكون جميع النتائج على نفس السطر يفصل بينها فراغ، أوكي نذهب إلى السطر الذاي يليه لنجد أمر Next Y أي إرجع إلى بدايه التكرار بعد أن تقوم بزياده قيمة المتغير Y بقيمة الـ Step وعليه سيكون قيمة المتغير Y الجديده يساوي Y القديمة + الـ Step وعليه 1+1 = 2 فنذهب للذاكره ونحذف القيمة القديمة للمتغير Y ونسجل الجديده، نذهب الآن إلى بدايه التكرار وسيكون For Y = 2 To 3 بما أن 2 داخل المدى 1 To 3 فسيدخل التكرار ثم يذهب لتعيين قيمة للعنصر الجديد M(X,Y) = X+Y عوض الآن المتغيرات حتى نعرف العنصر المطلوب وقيمته الجديده M(1,2) = 1 + 2 وعليه سيكون الناتج M(1,2) = 3 توجه للذاكره وسجل ذلك ثم يذهب للأمر الذي يليه ويقوم بطباعته، تمام الان نرجع مجدداً إلى Next Y وكما قلنا إرجع إلى بدايه التكرار بعد أن تقوم بزياده قيمة المتغير Y بقيمة الـ Step وعليه سيكون قيمة المتغير Y يساوي 2 + 1 = 3 فنذهب للذاكره ونحذف القيمة القديمة للمتغير Y ونسجل الجديده، نرجع إلى بدايه التكرار For Y = 3 To 3 وبما أن 3 داخل المدى 1 To 3 فبالتأكيد سيدخل إلى التكرار وسينفذ ما قمنا بفعله للمتغيرات التي قبله وسيكون الناتج تعيين قيمة للعنصر M(1,3) = 4 ومن ثم طباعه الناتج على شاشه المخرجات ثم يذهب للأمر Next Y ويضف قيمة الـ Step للمتغير Y لتصبح قيمته الجديده = 3 + 1 = 4 فنذهب للذاكره ونحذف القيمة القديمة ونسجل الجديده الان سيرجع إلى بدايه التكرار ولكن هل 4 داخل المدى 1 To 3 ؟ بالتأكيد لا فسيخرج من التكرار الثاني ويعود إلى التكرار الأول ليجد الأمر Next X أصبح هذا الأمر معروف وستصبح قيمة المتغير X = 2 فنذهب للذاكره ونحذف القيمة القديمة ونسجل الجديده ونذهب إلى بدايه التكرار الأول ونسأل أنفسنا هل 2 داخل المدى 1 To 2 بالتأكيد نعم إذن سيدخل التكرار الأول ليجد أمامه مره أخرى التكرار الثاني فسيدخل فيه ويقوم بحذف القيمة القديمة للمتغير Y والتي تساوي 4 ويسجل القيمةالجديده مكانها وهي 1 ويعيد ما تم شرحه قبل قليل حتى ينهي عمل البرنامج وعليه سيكون ناتج شاشه المخرجات:

الذاكره

M(1 To 2 , 1 To 3)	X	Y
M(1,1) = 2	~~1~~	~~1~~
M(1,2) = 3	~~2~~	~~2~~
M(1,3) = 4	3	~~3~~
M(2,1) = 3		~~4~~
M(2,2) = 4		~~1~~
M(2,3) = 5		~~2~~
		~~3~~
		4



Copyrights 2012 ©Mohammad Rasheid

تدريب: جد ناتج كل من البرامج التاليه:

1 - كم مره ستظهر رساله MsgBox ؟

```
Option Base 1
Private Sub Command1_Click()
Dim dr91$(5)
Dim x%
For x = LBound(dr91) To UBound(dr91)
dr91(x) = (x ^ 0 + 5) \ 2
MsgBox dr91(x), vbInformation, "Mohammad Rasheid"
If x = 3 Then Exit For
Next x
End Sub
```

2 - ما ناتج شاشه المخرجات ؟

```
Option Base 1
Private Sub Command1_Click()
Dim dr91%(10, 1), Y%, X
For Y = 1 To 1 Step 1
    For X = 1 To 10
    dr91(X, Y) = 5 * X
    Print "5 " & "* " & X & " = " & dr91(X, Y)
    Next X
Next Y
End Sub
```

3 - ما ناتج شاشه المخرجات ؟

```
Option Base 1
Private Sub Command1_Click()
Dim x(1, 1 To 3) As Integer
x(1, 2) = 5
x(1, 1) = 3
x(1, 3) = x(1, 2) + x(1, 1)
x(1, 3) = x(1, 3) - x(1, 1)
x(1, 1) = (((8 - 3) * 2) / 5) + 3
If (x(1, 1) + x(1, 2)) / 2 = x(1, 3) Then Exit Sub
Print x(1, 1), x(1, 2), x(1, 3)
End Sub
```

NOTHING

⭐⭐ The End of Chapter Five ⭐⭐

Copyrights 2012 ©Mohammad Rasheid

visual basic 6

chapter six

INDEX

Copyrights 2012 ©Mohammad Rasheid

Procedures And Functions

وجدت صعوبه كبيرةً في كيفيه البدايه في شرح هذا القسم لأن فكره هذا القسم من الماده لم ولن يتم شرحه بشكل كامل وثق بأنني لن أقوم بتعقيدها ولكن هذا القسم يحوي أفكاراً كثيره بالرغم من أن هذا القسم من الماده المقرره يحوي 11 صفحه فقط! لم أجدها من وجه نظري كافيه ولكن سأسعى بأن أختصر وأشمل قدر الإمكان، لاحظ معي بأننا نستطيع كتابه الكود كاملاً في زر أمر واحد وأن يشمل هذا الزر جميع عمليات الحسابيه والمقارنه وو ... الخ ولكن هل سألت نفسك لو كان هذا البرنامج أكبر حجماً ويحتاج إلى عمليات حسابيه كثيره فهل يعقل بأن الزمن الذي يستغرقه في التنفيذ سيكون أقل من البرنامج الذي العمليات الذي يمر بها أثناء التنفيذ قليله ؟ بالتأكيد لا فمن هنا كان لابد من وجود طريقه تقوم على تجزئه الكود وتقليص حجمه بحيث لو إحتجت إلى قسم معين يكفي بأن أطلب إستدعائه وهنا سيكون تنفيذ الكود أسرع بكثير وحجم الكود داخل الزر صغير، ومن هنا جائت أهميه الإجراءات (Procedures) بحيث تسمح لك بتقسيم الكود وكتابه إجراءاتك الخاصه وإستخدامها وإستدعائها متى تريد وهو محور دراستنا الآن.

الصيغه العامة لكتابه الإجراءات (Procedures):

الصيغه العامة:

```
[Private|Public] Sub SubName(Arg1 as Type1,……, Argn2 as Typen)
Sub Body
End Sub
```

حيث أن:

([Private|Public]): مستوى تعريف الإجراء عام أم خاص.

(Sub): من كلمات الفيجوال بيسك المحجوزه.

(SubName): هو إسم الإجراء الذي تريد ويفضل أن يكون ذا معنى حتى يسهل تتبع البرنامج.

(Arg1 as Type1 , …., Arg2 as TypeN): الباراميترز وسترى كيفيه التعامل معها.

(Sub Body): جسم الإجراء ويمكنك أن تكتب ما يحلو لك من أوامر داخله.

(End Sub): من كلمات الفيجوال بيسك المحجوزة وتعني نهايه الإجراء.

كيفيه كتابه الإجراءات / إضافتها:

توجد طريقتين لكتابه الإجراءات إما بشكل يدوي أو بشكل مرئي وسوف نتناول الطريقتين علماً بأن الطريقه اليدويه غير مطلوبه منك ولكن من باب المعرفه وجب إضافتها.

Copyrights 2012 ©Mohammad Rasheid

1 - كتابه الإجراءات (يدوياً): كتابه الإجراءات بسيطه جداً ويمكنك كتابتها بناءً على الصيغه العامة التي درسناها قبل قليل ما عليك سوى كتابه مدى الإجراء الذي تريد وإسم الإجراء والباراميترز إن وجد (بحسب عمل برنامجك) ثم تضف ما تريد من أكواد برمجيه والمثال التالي يوضح ما أقصد لاحظ معي:

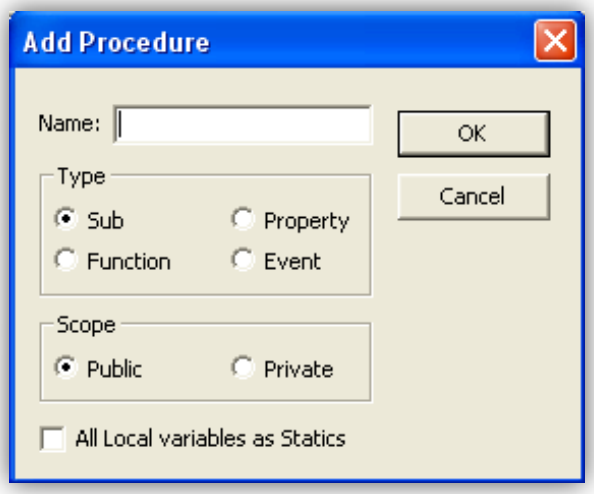
```
Public Sub Mohammad_Rasheid()
MsgBox "Welcome In my World", 64, "The Wolf Of Visual Basic"
End Sub

Private Sub Dr91()
MsgBox "• Mohammad Waleed × Hi Man •", 64, "The Wolf Of Visual Basic"
End Sub
```

• بسيطه صح ☻ !

2 - إضافه الإجراءات بواسطه (Add Procedures)

تعتبر أكثر شيوعاً بين الأشخاص الكسالى، كل ما عليك هو أن تتبع الخطوات التاليه:

I. إفتح شاشه الكود بالنقر المزدوج في أي مكان خالي في الشاشه.
II. من قائمة اللوائح إختر (Tools) ومنها إختر (Add Procedures).
III. ستلاحظ ظهور صندوق (Add Procedures)، كما بالشكل التالي:



IV. في الخانه (Name) قم بكتابه إسم الإجراء الذي تريد ويفضل أن يكون ذا معنى.
V. في القسم (Type) إختر نوع الإجراء.
VI. في قسم (Scope) إختر مدى الإجراء عام أم خاص.
VII. الخيار (All Local variable as Statics) أي هل تريد أن تكون المتغيرات التي يتم تعريفها داخل هذا الإجراء من نوع (Static) أي تحفظ قيمة المتغير بداخلها ما دام البرنامج يعمل ؟
VIII. بعد الإنتهاء من تعبئه النموذج إضغط على زر موافق (OK) لتلاحظ نزول الإجراء داخل شاشه الكود، كل ما تبقى عليك هو التوجه إلى (Sub Body) لكتابة ما تريد من أكواد.

Copyrights 2012 ©Mohammad Rasheid

⭐ كيفيه إضافه وظيفه (Function):

ما راح أشرح كيف زي العنجد على شان أعمل حالي فهمان وأكتب صفحه عليها، لالا القصه سهله شفت الخطوات إلي فوق نفسها إتبعها ولما تصل لقسم (Type) إختر نوع الإجراء (Function) وسلامتك هيك بتكون أضفت (Function) وسلامتك لاحظ الكود التالي:

```
Public Function HOHOH()

End Function
```

الإجراء (H0H0H) مداه (Public)، لاحظ بدل (Public Sub) أصبحت (Public Function) وبدل (End Sub) أصبحت (End Function).

⚡ آليه عمل الإجراءات:

لاحظ معي المثال التالي الذي من خلاله سنوضح آليه عمل الإجراءات، حيث يحوي المثال على إجرائين الأول (Dr91) والثاني (Mohammad_Rasheid)، ولذي وبحسب الكود المرفق زر أمر يحوي على أمرين كل أمر يقوم بإستدعاء الإجراء الذي يريد.

```
Private Sub Command1_Click()
Call Dr91
Mohammad_Rasheid
End Sub
Public Sub Dr91()
MsgBox "Nice"
End Sub
Private Sub Mohammad_Rasheid()
Form1.BackColor = vbRed
End Sub
```

الآن لنحاكي عمل الفيجوال بيسك في البدايه عندما يقوم المستخدم بالضغط على زر الأمر فسيقوم الفيجوال بيسك بالدخول إلى جسم زر الأمر ليقوم بتنفيذ ما سيجده من أكواد وسيجد الآن أمر بإستدعاء الإجراء (Dr91) فسيقوم بالإنتقال إلى الإجراء (Dr91) كما بالسهم رقم 1 ليقوم ما سوف يجده من أوامر داخل الإجراء، الآن عندما دخل إلى جسم الإجراء وجد أمر بإظهار رساله (MsgBox) فسيقوم بإظهار رساله للمستخدم تحوي النص (Nice) كما بالشكل التالي:

الآن عند ضغط المستخدم على زر أوكي سيكمل تنفيذ الفيجوال بيسك تنفيذ الأكواد ولكنه لايجد سوى أمر نهايه الإجراء (End Sub) فسيقوم الفيجوال بيسك بالرجوع لتنفيذ الأوامر التي تلي الأمر الذي قام بإستدعاء الإجراء (Dr91) فكما تعرفنا في (Chapter One) بأن لغه الفيجوال بيسك لغه تتبعيه أي تنفذ سطر سطر، وعليه سيعود إلى الإجراء (Command1_Click) كما بالسهم رقم 2 ليكمل ما تبقى من كواد إن وجد، ويلاحظ هنا بأنه وجد أمر آخر لإستدعاء الإجراء (Mohammad_Rasheid) وعليه سيقوم بالإنتقال إلى الإجراء كما بالسهم رقم 3 ليدخل إلى جسم الإجراء ويجد الأمر الخاص بتغيير لون الفورم (النموذج- Form) إلى اللون الأحمر كما بالشكل التالي:

ثم يكمل تنفيذ الأكواد إن وجدت فلا يجد أكواد إضافيه فيتوجه إلى (End Sub) أي نهـــايه الإجـــراء وسيعود إلى ما بعد الأمر الذي قام بإستدعاء الإجراء الثاني أي مع السهم رقم 4 ثم يكمل تنفيذ الأكواد إن وجدت ولكن لايجد سوى الأمر (End Sub) أي نهايه الإجراء فينهي الإجراء وينهي معه البرنامج وعليه سيكون الناتج هو ظهور رساله للمستخدم مع تغير لون خلفيه البرنامج إلى الأحمر.

Copyrights 2012 ©Mohammad Rasheid

كيفيه إستدعاء الإجراءات:

الإستدعاء نوعين الأول بوجود باراميترز والذي يقسم أيضاً إلى قسمين والثاني في حال عدم وجود باراميترز وهذا ما سأقوم بشرحه أولاً لسهولته.

A. إستدعاء إجراء في حال عدم وجود باراميترز:

لإستدعاء أي إجراء تريد كل ما عليك هو أن تكتب إسم الإجراء الذي تريد مباشره في المكان الذي تريد أو أن تكتب إسم الإجراء بعد إستخدام الأمر (Call)، كما لاحظنا بالمثال السابق.

B. إستدعاء إجراء في حال وجود باراميترز:

الإستدعاء إما أن يكون بواسطه القيمة (By Value) أو بواسطه الإشاره (By Reference).

1 - الإستدعاء بواسطه القيمة (By Value):

بحيث عن الإستدعاء إتفقنا على كتابه الأمر Call ثم اسم الإجراء، الان عند وضع الباراميترز قلنا بأننا نكتبها داخل قوسين وبما أننا سنستعمل الطريقه By Value فكل قيمة توضع أيضاً بين قوسين ومثال عليها الكود التالي:

```
Call Dr91( (9) , (3) )
```

لاحظ معي إذا كانت القيم بين قوسين كما بالمثال السابق فبلغه الإختصار (من الآخر ريحني وريحك) بتروح عالإجراء وبتنفذ كل شي وبترجع إتكمل تنفيذ الأكواد بدون ما يصير أي تغير أو تعديل على الأرقام (الباراميترز)، عارف راح إتقولي طلاسم بقولك شوف هالمثال.

```
Private Sub Command1_Click()
Dim x%, y%
x = 5
y = x + 2
Print "By Value راح إنطير هلا للإجراء Dr91 وطريقه الطيران بإستخدام"
Call Dr91((x), (y))
Print "Command1_Click تم الرجوع إلى الإجراء"
End Sub

Private Sub Dr91(n1 As Integer, n2 As Integer)
Print "Dr91 إنت الآن في مطار الإجراء"
Sum = n1 + n2
av = Sum / 2
Print "Sum= " & Sum, "Avg= " & av
n1 = n1 + 5
n2 = n1 + 2
Print "تم الإنتهاء من الإجراء وحان موعد الرجوع الى الإجراء الرئيسي"
End Sub
```

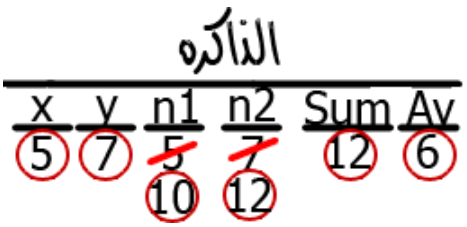
لاحظ معي في الكود السابق داخل كود زر الامر (Command1) تم تعريف المتغيرين (x,y) بأنهما من نوع (Integer) ومن ثم تعيين قيم للمتغيرين كما بالسطر الثاني والثالث، الان ينتقل للسطر الرابع ليقوم بتنفيذ جمله الطباعه وسيقوم بطباعه النص المرفق داخل علامتي الإقتباس، ثم ينتقل للسطر الذي يليه والذي تم تحديده باللون الأصفر لاحظ بأنه أمر بإستدعاء الإجراء Dr91 الأهم من هذا لاحظ أن كل باراميترز تم وضعه داخل قوسين وهذا يعني بأن طريقه الإستدعاء تنص على طريقه (By Value) أي بالقيمة، الان نتوجه لتنفيذ الإجراء Dr91 لكن قبل البدء بالسطر الأول من هذا الإجراء لاحظ بعد (Private Sub Dr91) تم تعريف متغيرين (n1 , n2) بأنهما من نوع (Integer) ومن هذا نستنتج بأن الباراميترز يجب أن يتم حفظهم داخل متغيرات جديده بشرط أن تكون من نفس نوع المتغير الذي تم إرساله بواسطه الأمر (Call)، أوكي إذن وعلى ما سبق سيتم حفظ قيمة المتغير x داخل المتغير n1 وقيمة المتغير y داخل المتغير n2، الان سيبدأ بتنفيذ الأكواد ومن السطر الأول نجد جمله طباعه سيقوم بتنفيذ وطباعه النص المرفق بين علامتي الإقتباس ثم يتوجه للسطر الذي يليه

Copyrights 2012 ©Mohammad Rasheid

وسيقوم بتخزين مجموع المتغيرين (n1 , n2) داخل المتغير Sum ومن ثم إيجاد معدل العديين وتخزينه داخل المتغير av، يتوجه بعدها لطباعه النص (Sum=) ثم مجموع العددين و (Avg=) ومعدل العددين الان لاحظ معي تم تحديد السطرين التاليين باللون الأصفر لذلك ركز معي هنا، في السطر الأول تم إعاده تعيين قيمة المتغير n1 بأن أصبح القيمة الجديده = القيمة القديمة + 5 أي أصبح = 10 وكذلك الحال بالنسبه للمتغير n2 أي أصبحت قيمته الجديده = 9 ثم ينتقل الى السطر الأخير ويقوم بطباعه النص المرفق بين علامتي إقتباس وبما أنه لم يبقى أي أكواد أخرى للتنفيذ فسوف يقوم بالرجوع إلى الإجراء الرئيسي (Command1) الآن بيت القصيد هنا بأننا قلنا بأن n1 تعادل x و n2 تعادل y للتذكير (Private Sub Dr91(n1 as integer , n2 as integer)) الان السؤال هنا هل قيمة x الآن أصبحت نفس قيمة المتغير n1؟ أي أصبحت تساوي 10؟ وقيمة المتغير y أصبحت نفس قيمة المتغير n2؟ أي أصبحت تساوي 12؟ حتى نعرف ذلك لابد للرجوع للإجراء الرئيسي والتأكد من جمله الإستدعاء هل كانت من نوع (By Value) أم من نوع (by Reference)؟ لاحظ معي:

((y), (x))Call Dr91 وإتفقنا أن البراميترز إذا وضع كل واحده داخل قوسين فهذا يعني بأنها تتبع طريقه (By Value) وعليه بما أن الطريقه تعتمد على الإستدعاء بالقيمة فلن تتأثر قيمة المتغيرين x و y أبداً وستبقى قيمة المتغير x = 5 وقيمة المتغير y=7 .

وعليه عند تنفيذ البرنامج سيكون ناتج شاشه الخرجات كما بالشكل وعند تمثيله بالذاكره سيكون كما بالشكل التالي:



2 - الإستدعاء بواسطه القيمة (By Reference):

إتفقنا على كتابه الأمر Call ثم اسم الإجراء، الان عند وضع البراميترز قلنا بأننا نكتبها داخل قوسين وبما أننا سنستعمل الطريقه By Reference فإننا نكتب البراميترز تفصل بينهما فقط الفاصله بدون أن توضع كل واحده على حدا داخل قوسين كما في الإستدعاء بالقيمة (By Value)، ويجب التنويه بأنه في حال إستخدام الإستدعاء بالإشاره (By Reference) فأي تعديل أو تغير على قيمة المتغير في الإجراء الفرعي ستؤثر وتغير قيمة المتغير في الإجراء الرئيسي ومثال عليها الكود التالي:

```
Call Dr91(9 , 3)
```

سوف نأخذ نفس المثال السابق مع تعديل بسيط وفي الإجراء الرئيسي وهو طباعه القيمة النهائيه اللمتغيرات بعد الإنتهاء من إستدعاء وتنفيذ الإجراء، لاحظ المثال.

Copyrights 2012 ©Mohammad Rasheid

```
Private Sub Command1_Click()
Dim x%, y%
x = 5
y = x + 2
Print "By Reference راح إنطير هلا للإجراء Dr91 وطريقه الطيران بإستخدام"
Call Dr91(x, y)
Print "Command1_Click تم الرجوع إلى الإجراء"
Print "-----------------------------------------------------"
Print " =قيمة المتغير x الجديده" & x
Print " =قيمة المتغير y الجديده" & y
End Sub
```

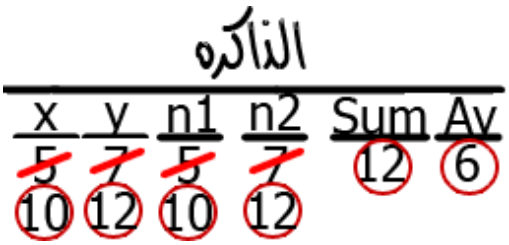
```
Private Sub Dr91(n1 As Integer, n2 As Integer)
Print "Dr91 إنت الآن في مطار الإجراء"
Sum = n1 + n2
av = Sum / 2
Print "Sum= " & Sum, "Avg= " & av
n1 = n1 + 5
n2 = n1 + 2
Print "تم الإنتهاء من الإجراء وحان موعد الرجوع الى الإجراء الرئيسي"
End Sub
```

لن أقوم بإعاده شرح هذا المثال السبب لأنه نفس المثال السابق ولكن سأوضح الفرق بين الكودين فقط أضف إلى ذلك بأن جميع ما سيرد داخل هذا الكود قم تمت دراسه في فصول سابقه! ، لاحظ أمر الإستدعاء (Call Dr91 (x ,y))، لم توضع البراميترز كل واحده على حدا داخل قوسين وهذا يعني بأن طريقه الإستدعاء المتبعه هي (By Reference) وهذا يعني أيضاً أن أي تعديل على قيمة المتغيرات سوف تغير معها القيمة الخاصه بالمتغير الأصلي، الآن ننتقل إلى الإجراء Dr91 ننتقل مباشره إلى السطرين:

n1 = n1 +5

n2 = n1 +2

لاحظ بأنه سيقوم بإعاده تعيين قيم للمتغيرات n1 التي تكافئ المتغير x والمتغير n2 يكافئ المتغير y وعليه ستصبح قيمة المتغير n1 = 10 وقيمة المتغير n2 = 12، وبما أننا إستخدمنا الإستدعاء بواسطه الإشاره (By Reference) فإن قيمة المتغيرين x و y ستتغيران أيضاً بحيث ستصبح قيمة المتغير x الجديده = 10 والقيمة الجديده للمتغير y = 12، الان عند العوده للإجراء الرئيسي سيقوم بطباعه كل من القيم الجديده للمتغيرين وليس القيم القديمة لأننا إستخدمنا الإستدعاء بالإشاره (By Reference)، وحتى نثبت ما قلنا لاحظ ناتج شاشه المخرجات عند التنفيذ وقيم المتغيرات بالذاكره.



Copyrights 2012 ©Mohammad Rasheid

تدريب: جد ناتج البرنامج التالي في حال:

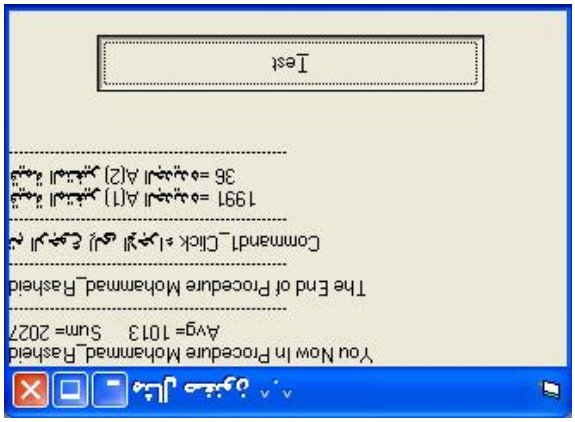
1 - كان الرقم المدخل من قبل المستخدم 3 2- كان الرقم المدخل من قبل المستخدم 6

```
Option Base 1
Private Sub Command1_Click()
Dim A%(3)
A(1) = 1991
A(2) = ((5 + 8 ^ 1) \ 2) ^ 2
A(3) = InputBox("Input Numer Between 1 To 10", "Dr91")
Select Case A(3)
Case 1 To 5: Call Mohammad_Rasheid(A(1), A(2))
Case 6 To 10: Call Mohammad_Rasheid((A(1)), (A(2)))
End Select
Print "-----------------------------------------------------"
Print "Command1_Click تم الرجوع إلى الإجراء"
Print "-----------------------------------------------------"
Print " =قيمة المتغير A(1) الجديدة" & A(1)
Print " =قيمة المتغير A(2) الجديدة" & A(2)
Print "-----------------------------------------------------"
End Sub
```

```
Private Sub Mohammad_Rasheid(Num1 As Integer, Num2 As Integer)
Print "You Now In Procedure Mohammad_Rasheid"
Sum = Num1 + Num2
av = Sum \ 2
Print "Sum= " & Sum, "Avg= " & av
Print "-----------------------------------------------------"
Num1 = Num1 + 5
Num2 = Num1 + 2
Print "The End of Procedure Mohammad_Rasheid"
End Sub
```

إجابه الفرع الأول

إجابه الفرع الثاني



The End of Chapter Six

Copyrights 2012 ©Mohammad Rasheid

Mohammad Rasheid

عزيزي الباحث لك مطلق الحرية بنقل هذا الملخص إلى أي منتدى أو موقع بهدف نشر الفائده، ولكن أرجو ذكر الحقوق وليس الإكتفاء بذكر أنها منقوله! لأن هذا الملخص كلف من الوقت والجهد الكثير لذلك ساهم معنا في حفظ حقوق هذا المرجع بالإشاره الواضحه لمصدره وشكراً.

18/04/2012

INDEX

Copyrights 2012 ©Mohammad Rasheid