

## 16 مشروع باستخدام الفيجوال بيسك 6

رقم الصفحة	اسم المشروع	ت
2	آلة حاسبة بسيطة	1
8	آلة حاسبه متكاملة	2
14	تحويل العملات	3
21	أدارة الفنادق	4
26	أسم المستخدم وكلمة السر	5
40	اوامر التكرار	6
53	الرسوم	7
58	الاشكال الهندسية	8
65	محاكاة السرعة	9
68	حساب الفائدة المركبة في البنك	10
75	بعض التطبيقات الرياضيه	11
81	دوال الحوار مع المستخدم	12
87	مصفوفات عناصر التحكم	13
94	التحريك	14
101	بناء البرامج الفرعية	15
110	ربط الفيجوال بيسك بقاعدة البيانات	16

## 1-المشروع الاول

(أسم المشروع : آلة حاسبة بسيطة)

سوف نتعلم من خلال تنفيذ هذا المشروع مايلي:

1. التعامل مع عناصر التحكم الأساسية.

- النموذج Form.
- صندوق النص Text Box.
- التسمية Label.
- زر الأمر Command Button.

2. كيفية إسناد قيمة الى خاصية عنصر تحكم.

3. الدالة Val.

4. العمليات الرياضية الاساسيه في Visual Basic.

5. أحد أشكال الأوامر الشرطية IF.

البرنامج (Source Code) الخاص بهذا المشروع موضح كمايلي:-

```
Private Sub Command1_Click()  
Text3.Text = Val(Text1.Text) + Val(Text2.Text)  
End Sub
```

```
Private Sub Command2_Click()  
End  
End Sub
```

```
Private Sub Command3_Click()  
Text3.Text = Val(Text1.Text) - Val(Text2.Text)  
End Sub
```

```
Private Sub Command4_Click()  
Text3.Text = Val(Text1.Text) * Val(Text2.Text)  
End Sub
```

```
Private Sub Command5_Click()  
Text3.Text = Val(Text1.Text) / Val(Text2.Text)  
End Sub
```

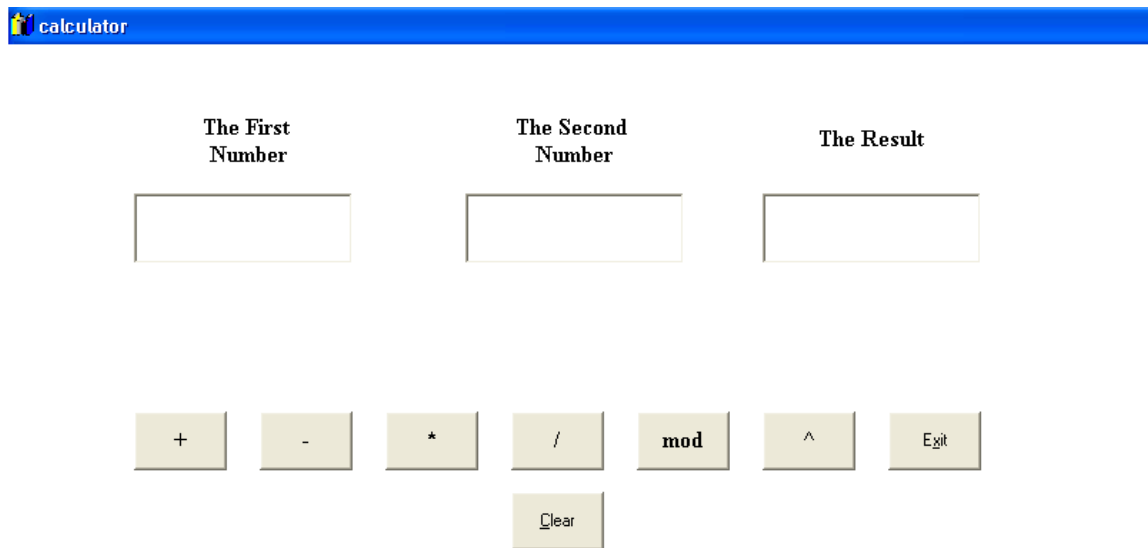
```
Private Sub Command6_Click()  
Text3.Text = Val(Text1.Text) Mod Val(Text2.Text)  
End Sub
```

```
Private Sub Command7_Click()
```

```
Text3.Text = Val(Text1.Text) ^ Val(Text2.Text)
End Sub
```

```
Private Sub Command8_Click()
Text3.Text = ""
Text2.Text = ""
Text1.Text = ""
End Sub
```

نتائج تنفيذ البرنامج أعلاه موضح بالشكل التالي:



1. Properties Of Form.
2. Properties Of Labels.
3. Properties Of Bottuns.

## 1. Properties Of Form1.

<b>PROPERTIES OF FORM1</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Form1
Caption	Calculator
Font	Times New Roman Font Style    Bold Font Size     12
ForeColor	Black   أسود

## 2. Properties Of Labels.

<b>PROPERTIES OF LABEL1</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Label1
Alignment	2-Center
Caption	The First Number
Font	Times New Roman Font Style    Bold Font Size     12
ForeColor	Black   أسود

<b>PROPERTIES OF LABEL2</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Label2
Alignment	2-Center
Caption	The ٥Second Number
Font	Times New Roman Font Style    Bold Font Size     12
ForeColor	Black   أسود

<b>PROPERTIES OF LABEL3</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Label3
Alignment	2-Center
Caption	The Result
Font	Times New Roman Font Style    Bold Font Size    12
ForeColor	Black    أسود

### 3. Properties Of TextBox.

<b>PROPERTIES OF TEXT1</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Text1
Alignment	2-Center
Font	Times New Roman Font Style    Bold Font Size    16
ForeColor	Blue    ازرق
Text	

<b>PROPERTIES OF TEXT2</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Text2
Alignment	2-Center
Font	Times New Roman Font Style    Bold Font Size    16
ForeColor	Blue    ازرق
Text	

<b>PROPERTIES OF TEXT3</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Text3
Alignment	2-Center
Font	Times New Roman Font Style    Bold Font Size    16
ForeColor	Blue    ازرق
Text	

#### 4. Properties Of Command Bottuns.

<b>PROPERTIES OF COMMAND1</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command1
Caption	+
Font	Times New Roman Font Style    Bold Font Size    16

<b>PROPERTIES OF COMMAND3</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command3
Caption	-
Font	Times New Roman Font Style    Bold Font Size    16

<b>PROPERTIES OF COMMAND4</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command4
Caption	*
Font	Times New Roman Font Style    Bold Font Size    16

<b>PROPERTIES OF COMMAND5</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command5
Caption	/
Font	Times New Roman Font Style    Bold Font Size    16

<b>PROPERTIES OF COMMAND6</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command6
Caption	Mod
Font	Times New Roman Font Style    Bold

	Font Size 16
<b>Properties of Command7</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command7
Caption	^
Font	Times New Roman Font Style Bold Font Size 16

<b>PROPERTIES OF COMMAND2</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command2
Caption	E&xit
Font	Times New Roman Font Style Bold Font Size 16

<b>PROPERTIES OF COMMAND8</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command8
Caption	&Clear
Font	Times New Roman Font Style Bold Font Size 16

## 2-المشروع الثاني

-:-----

## الآلة حاسبه بسيطه متكامله

- النموذج Form
- صندوق النص Text Box
- التسمية Label
- زر الامر Command Button
- اسناد قيمه ثابتة الى خاصيه عنصر التحكم Text Box
- الدالة Val
- العمليات الاساسيه الرياضيه في Visual Basic
- الشرط If .. Then .. Else

الهدف من المشروع

سوف نقوم ببناء الآلة حاسبه بسيطه متكامله تسمح للمستخدم تنفيذ العمليات الرياضيه الاساسيه على عددين (يستخدم Text Box) واحد لتنفيذ عمليات الحاسبه هذه وليس ثلاثه كما في المثال السابق.

عند تنفيذ البرنامج سوف نحصل على الشاشة التاليه:



البرنامج الذي يقوم بتنفيذ الحاسبه اعلاه مدون كالتالي:

Dim n1 As Single

[felah\\_najar2000@yahoo.com](mailto:felah_najar2000@yahoo.com)[fallahnajjar@yahoo.com](mailto:fallahnajjar@yahoo.com)[+9647807445833](tel:+9647807445833)



```
Dim n2 As Single
Dim op As String
Private Sub Command1_Click()
Text1.Text = (Text1.Text) + "1"
End Sub
```

```
Private Sub Command10_Click()
Text1.Text = (Text1.Text) + "8"
End Sub
```

```
Private Sub Command11_Click()
Text1.Text = (Text1.Text) + "9"
End Sub
```

```
Private Sub Command12_Click()
Text1.Text = (Text1.Text) + "0"
End Sub
```

```
Private Sub Command13_Click()
n1 = Val(Text1.Text)
op = "+"
Text1.Text = ""
End Sub
```

```
Private Sub Command14_Click()
n1 = Val(Text1.Text)
op = "/"
Text1.Text = ""
End Sub
```

```
Private Sub Command15_Click()
n1 = Val(Text1.Text)
op = "mod"
Text1.Text = ""
End Sub
```

```
Private Sub Command16_Click()
Text1.Text = ""
End Sub
```

```
Private Sub Command17_Click()
End
End Sub
```

```
Private Sub Command18_Click()  
n1 = Val(Text1.Text)  
op = "^"  
Text1.Text = ""  
End Sub
```

```
Private Sub Command19_Click()  
Text1.Text = (Text1.Text) + "."  
End Sub
```

```
Private Sub Command2_Click()  
Text1.Text = (Text1.Text) + "2"  
End Sub
```

```
Private Sub Command20_Click()  
Text1.Text = "Mathematical Calculator"  
End Sub
```

```
Private Sub Command21_Click()  
n1 = Val(Text1.Text)  
op = "-"  
Text1.Text = ""  
End Sub
```

```
Private Sub Command3_Click()  
Text1.Text = (Text1.Text) + "3"  
End Sub
```

```
Private Sub Command4_Click()  
Text1.Text = (Text1.Text) + "4"  
End Sub
```

```
Private Sub Command5_Click()  
Text1.Text = (Text1.Text) + "5"  
End Sub
```

```
Private Sub Command6_Click()  
n2 = Val(Text1.Text)  
If op = "+" Then  
Text1.Text = n1 + n2  
ElseIf op = "*" Then  
Text1.Text = n1 * n2  
ElseIf op = "-" Then  
Text1.Text = n1 - n2  
ElseIf op = "/" Then  
Text1.Text = n1 / n2
```

```

ElseIf op = "^" Then
Text1.Text = n1 ^ n2
ElseIf op = "mod" Then
Text1.Text = n1 Mod n2
Else
End If
End Sub

```

```

Private Sub Command7_Click()
n1 = Val(Text1.Text)
op = "*"
Text1.Text = ""
End Sub

```

```

Private Sub Command8_Click()
Text1.Text = (Text1.Text) + "6"
End Sub

```

```

Private Sub Command9_Click()
Text1.Text = (Text1.Text) + "7"
End Sub

```

نستخدم الخصائص التاليه:

1. الشكل Form

PROPERTIES OF FORM1	
Alphabetic	Categorized
(Name)	Form1
Caption	Calculator
Font	Times New Roman Font Style Bold Font Size 12
Back Color	High Light
ForeColor	Black اسود

2. صندوق النص Text Box

### PROPERTIES OF TEXT BOX - 1

<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Text1
Alignment	0 – Left Justify
Back Color	High Light
Font	Times New Roman Font Style    Bold Font Size     14
Text	

3. زر الامر Command Button

<b>PROPERTIES OF COMMAND1</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command
Caption	1, 2, 3, 4, 5, 6, 7, 8, 9, 0
Font	Times New Roman Font Style    Bold Font Size     14

<b>PROPERTIES OF COMMAND1</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command
Caption	Exit, CE
Font	Times New Roman Font Style    Bold Font Size     14

<b>PROPERTIES OF COMMAND1</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command
Caption	*, /, +, -, ^, Mod, =
Font	Times New Roman Font Style    Bold Font Size     14

<b>PROPERTIES OF COMMAND1</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command

Caption	Calculator
Font	Times New Roman Font Style    Bold Font Size     14

## 3- المشروع الثالث

-:-----

## (تحويل العملات)

سوف نتعلم من خلال هذا المشروع مايلي:-  
1. التعامل مع عناصر التحكم التاليه:-

- النموذج Form
- صندوق النص Text Box
- التسميه Label
- زر الاوامر Command Button
- صندوق القائمه المنسدله Combo Box
- الصوره Image

2. الشكل العام للاوامر الشرطيه If .. then ..else
3. الشكل العام للامر الشرطي Select .. Case

الهدف من المشروع

-:-----

بناء واجهة لتحويل العملات او المبلغ المكافئ بالدولار لقيمة مبلغ بعملة بلد معين،

عند تنفيذ هذا المشروع تظهر الواجه التاليه:



The Amounts   -

The Country

The Result  \$

How many Dollars  How many Dollars  How many Dollars

نستخدم الخصائص التاليه الخاصه بكل جزء من الاجزاء المستخدمه في بناء المشروع:

4. الشكل Form

<b>PROPERTIES OF FORM1</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	FrmEx
Caption	Money Changer
Font	Times New Roman Font Style Bold Font Size 12
Back Color	High Light
ForeColor	Black اسود

5. صندوق النص Text Box

<b>PROPERTIES OF TEXT BOX - 1</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	TxtIn
Alignment	2 - Center
Back Color	High Light
Font	Times New Roman Font Style Bold Font Size 14
Text	

<b>PROPERTIES OF TEXT BOX - 2</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	TxtOut
Alignment	2 - Center
Back Color	High Light
Font	Times New Roman Font Style Bold Font Size 14
Text	

## 6. صندوق القوائم Combo Box

<b>PROPERTIES OF COMBO BOX</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	CboIn
Back Color	Window Background
Font	Times New Roman Font Style    Bold Font Size     14
list	France Germany Italy

## 7. مفتاح الاوامر Command Button

<b>PROPERTIES OF COMMAND1</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command1
Caption	How Many Dollars
Font	Times New Roman Font Style    Bold Font Size     12

<b>PROPERTIES OF COMMAND2</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command2
Caption	How Many Dollars
Font	Times New Roman Font Style    Bold Font Size     12

<b>PROPERTIES OF COMMAND3</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Command3
Caption	How Many Dollars
Font	Times New Roman Font Style    Bold Font Size     12



8. التسميه Label

<b>PROPERTIES OF LABEL</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	LblIn
Alignment	2-Center
Caption	-
Font	Times New Roman Font Style    Bold Font Size    12
ForeColor	Black أسود

<b>PROPERTIES OF LABEL2</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Label2
Alignment	2-Center
Caption	The Amount
Font	Times New Roman Font Style    Bold Font Size    14
ForeColor	Black أسود

<b>PROPERTIES OF LABEL3</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Label3
Alignment	2-Center
Caption	The Country
Font	Times New Roman Font Style    Bold Font Size    14
ForeColor	Black أسود

<b>PROPERTIES OF LABEL4</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Label4
Alignment	2-Center
Caption	The Result
Font	Times New Roman Font Style    Bold Font Size    12
ForeColor	Black أسود

<b>Properties of label1</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	Label1
Alignment	2-Center
Caption	\$
Font	Times New Roman Font Style    Bold Font Size    14
ForeColor	Black أسود

9. صندوق الصور Picture Box

<b>PROPERTIES OF PICTURE BOX</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	ImgIn
AutoSize	True
Visible	False
Picture	Micro Soft Visual Studio, Common, Graphics, Icons, Flags, Choose Country Flag, CTRUSA

<b>PROPERTIES OF PICTURE BOX</b>	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	ImgF
AutoSize	True
Visible	False
Picture	Micro Soft Visual Studio, Common, Graphics, Icons, Flags, Choose Country Flag, FLGFRAN

PROPERTIES OF PICTURE BOX	
Alphabetic	Categorized
(Name)	ImgG
AutoSize	True
Visible	False
Picture	Micro Soft Visual Studio, Common, Graphics, Icons, Flags, Choose Country Flag, FLGGERM

PROPERTIES OF PICTURE BOX	
Alphabetic	Categorized
(Name)	ImgI
AutoSize	True
Visible	False
Picture	Micro Soft Visual Studio, Common, Graphics, Icons, Flags, Choose Country Flag, FLGITALY

## البرنامج Source Code

```

Private Sub Command1_Click()
    If CboIn.Text = "France" Then
        TxtOut.Text = Val(TxtIn.Text) * 0.2
        LblIn.Caption = "FR"
        ImgIn.Picture = ImgF.Picture
    ElseIf CboIn.Text = "Germany" Then
        TxtOut.Text = Val(TxtIn.Text) * 0.6
        LblIn.Caption = "DM"
        ImgIn.Picture = ImgG.Picture
    ElseIf CboIn.Text = "Italy" Then
        TxtOut.Text = Val(TxtIn.Text) * 0.1
        LblIn.Caption = "IL"
        ImgIn.Picture = ImgI.Picture
    Else
        MsgBox "Please Choose The Country From The List", vbCritical, "Error"
    End If
End Sub

```

```
Private Sub Command2_Click()  
If CboIn.ListIndex = 0 Then  
    TxtOut.Text = Val(TxtIn.Text) * 0.2  
    LblIn.Caption = "FR"  
    ImgIn.Picture = ImgF.Picture  
ElseIf CboIn.ListIndex = 1 Then  
    TxtOut.Text = Val(TxtIn.Text) * 0.6  
    LblIn.Caption = "DM"  
    ImgIn.Picture = ImgG.Picture  
ElseIf CboIn.ListIndex = 2 Then  
    TxtOut.Text = Val(TxtIn.Text) * 0.1  
    LblIn.Caption = "IL"  
    ImgIn.Picture = ImgI.Picture  
Else  
    MsgBox "Please Choose The Country From The List", vbCritical, "Error"  
End If  
End Sub
```

```
Private Sub Command3_Click()  
Select Case CboIn.ListIndex  
Case 0  
    TxtOut.Text = Val(TxtIn.Text) * 0.2  
    LblIn.Caption = "FR"  
    ImgIn.Picture = ImgF.Picture  
Case 1:  
    TxtOut.Text = Val(TxtIn.Text) * 0.6  
    LblIn.Caption = "DM"  
    ImgIn.Picture = ImgG.Picture  
Case 2:  
    TxtOut.Text = Val(TxtIn.Text) * 0.1  
    LblIn.Caption = "IL"  
    ImgIn.Picture = ImgI.Picture  
Case Else  
    MsgBox "Please Choose The Country From The List", vbCritical, "Error"  
End Select  
End Sub
```

#### 4-المشروع الرابع

-----:-

#### (أدارة الفنادق)

سوف نتعلم من خلال هذا المشروع

- التعامل مع عناصر التحكم:

- الاطار Frame
- زر الخيار Option Button
- صندوق التحقق Check Box

- تعليمات الشرط

**الهدف من المشروع:-** تكوين برنامج بلغة Visual Basic في أدارة الفندق من حيث كلفة المبيت المعتمده على تصنيف الفندق (من حيث عدد النجوم)، مكان الفندق هل هو في مركز المدينه او في الضواحي، الوجبات التي يريد تناولها (فطور، غداء، عشاء) طريقة الدفع (نقدا او شيك).

عند تنفيذ هذا المشروع سوف تظهر الواجهه التاليه الخاصه بتحديد مواصفات الاقامه في فندق في فرنسا مثلا:

The screenshot shows a Windows application window titled "Hotel". The interface includes the following elements:

- Place:** A group box containing two radio buttons: "Paris" (selected) and "Out".
- Payment:** A group box containing two radio buttons: "Cash" (selected) and "Credit".
- Food:** A group box containing three checked checkboxes: "Break Fast", "Lunch", and "Dinner".
- How Much:** A button labeled "How Much" next to a text box containing the value "990" and the label "Dinar".
- Other:** A dropdown menu with "\*\*\*\*" visible, and a taskbar at the bottom showing the start button and several open applications.

سوف نقوم في هذا البرنامج بحساب كلفة الاقامة وكما يلي:

1. سعر الليله في فندق (\*) 100
- سعر الليله في فندق (\*\*) 200
- سعر الليله في فندق (\*\*\*) 300
- سعر الليله في فندق (\*\*\*\*) 400
- سعر الليله في فندق (\*\*\*\*\*) 500

2. إذا كان الفندق في باريس يضاف 50% للتسعيره السابقه.

3. اسعار الطعام:-

- سعر الفطور 50
- سعر الغداء 100
- سعر العشاء 200

4. اذا كان الدفع نقدا يخصم 10% من السعر الكلي.

البرنامج الذي يقوم بتنفيذ هذه الواجهه مدون ادناه

```
Dim X As Single
Private Sub Command1_Click()
Select Case CboStar.ListIndex
Case 0:
X = 100
Case 1:
X = 200
Case 2:
X = 300
Case 3:
X = 400
Case 4:
X = 500
Case Else
MsgBox "Choose how many stars", vbCritical, "Error"
Exit Sub
End Select
If OptParis.Value = True Then X = X * 1.5
If ChkBreak.Value = 1 Then X = X + 50
If ChkLunch.Value = 1 Then X = X + 100
If ChkDinner.Value = 1 Then X = X + 200
If OptCash.Value = True Then X = X * 0.9
TxtOut.Text = X
End Sub
```

مراحل التنفيذ

-----:  
نفتح مشروعاً قياسياً جديداً .

- ضبط خصائص النموذج وكمايلي:
- ضبط خصائص الشكل Form كمايلي:

PROPERTIES OF FORM1	
Alphabetic	Categorized
(Name)	Form1
Caption	Hotel
Font	Times New Roman Font Style Bold Font Size 14
ForeColor	Black اسود

- ننشي صندوق قائمة منسدلة ونضبط الخصائص كالاتي:

PROPERTIES OF COMBOBOX	
Alphabetic	Categorized
(Name)	CboStar
List	* ** *** **** *****
Text	Choose the stars

- ننشأ ثلاثة عناصر من نوع الاطار Frame ونضبط الخاصية Caption لكل منهما: Place , Payment, Food.

يستخدم الاطار لضم مجموعه من العناصر المترابطة فيما بينها عمليا، يجب انشاء الاطار اولا ثم انشاء العناصر بداخله ( اي لاننشئ مثلا ازرار الخيار في مشروعنا على النموذج ثم نسحب داخل الاطار)، للتأكد من تابعة العناصر للاطار يجب عند سحب الاطار ان تسحب العناصر معه.

يستخدم الاطار بشكل اساسي لضم مجموعه من ازرار الخيار Option Button ويمكن لزر واحد فقط في المجموعه ان يكون محددًا، وعند النقر على زر في المجموعه نفسها يتحدد هذا الزر ويفقد التحديد الذي عليه التحديد.

- ننشئ زر خيار Option Button داخل الاطار Food ونضبط خصائصه كمايلي:

PROPERTIES OF OPTION BUTTON	
Alphabetic	Categorized
(Name)	OptParis
Caption	Paris
Value	True

PROPERTIES OF OPTION BUTTON	
Alphabetic	Categorized
(Name)	OptProv
Caption	Out
Value	False

تصميم الخاصيه Value لعنصر التحكم Option Button تأخذ القيمة True اذا كان زر الخيار محددًا ، اما اذا كان زر الخيار غير محددًا فتأخذ القيمة False.

- ننشئ زر خيار Option Button داخل الاطار Payment ونضبط خصائصه كمايلي:

PROPERTIES OF OPTION BUTTON	
Alphabetic	Categorized
(Name)	OptCash
Caption	Cash
Value	True

PROPERTIES OF OPTION BUTTON	
Alphabetic	Categorized
(Name)	OptCheq
Caption	Credit
Value	False

- ننشئ ثلاثة صناديق تحقق Check Box ضمن الاطار Food ونضبط الخصائص كمايلي:



PROPERTIES OF CHEK BOX	
<b>Alphabetic</b>	<b>Categorized</b>
(Name)	ChkBreak
Caption	Break Fast
Value	1- Checked

PROPERTIES OF CHEK BOX	
<b>Alphabetic</b>	<b>Alphabetic</b>
(Name)	ChkLunch
Caption	Lunch
Value	0 - Unchecked

PROPERTIES OF CHEK BOX	
<b>Alphabetic</b>	<b>Alphabetic</b>
(Name)	ChkDinner
Caption	Dinner
Value	0 - Unchecked

تصميم الخاصية Value لعنصر التحكم Check Box تاخذ القيم الثلاثة التالية

0-UNCHECKED	غير محدد
1-Checked	محدد
2-Grayed	رمادي

- ننشئ صندوق نص لوضع الناتج فيه واسمه (TxtOut) و زر امر لايجاد المبلغ الكلي اسمه (How Much) لاجراء الحساب عند النقر عليه .

## 5- المشروع الخامس

-----:-

اسم المشروع :

( أسم المستخدم User Name وكلمة السر Password)

سوف نتعلم من خلال هذا المشروع مايلي:

❖ التعامل مع المتغيرات Variables

- المتغيرات المحلية للإجراء procedure

- المتغيرات الساكنة Static للإجراء

- المتغيرات العامة للنموذج

- المتغيرات العامة للمشروع

❖ التعامل مع اكثر من نموذج في المشروع

❖ التعامل مع الوحدات النمطية Module

الهدف من المشروع

-----:-

عند تنفيذ المشروع سوف تظهر الواجهة التالية التي تطلب اسم المستخدم وكلمة السر

The screenshot shows a standard Windows-style dialog box titled "Login". It features a light green background and a blue title bar. The dialog contains two text input fields. The first field is labeled "User Name" and contains the text "fallahnajjar". The second field is labeled "Password" and contains four asterisks "\*\*\*\*". Below these fields are two buttons: "Ok" and "Cancel".

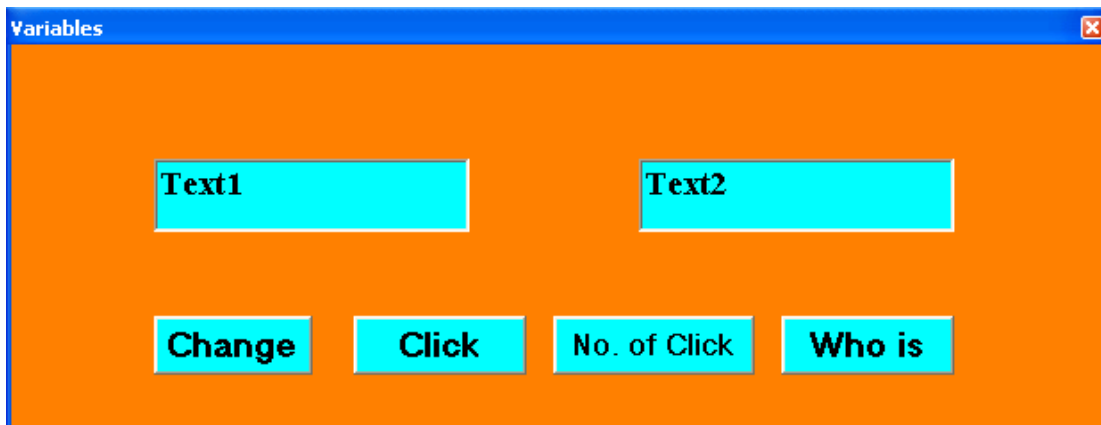
عندما يدخل المستخدم اسمة في مربع النص ثم كلمة السر في مربع النص الثاني التي هي toto في مثالنا هذا، فاذا أخطأ المستخدم في ادخال كلمة السر سوف تظهر العبارة التنبيهية التالية:



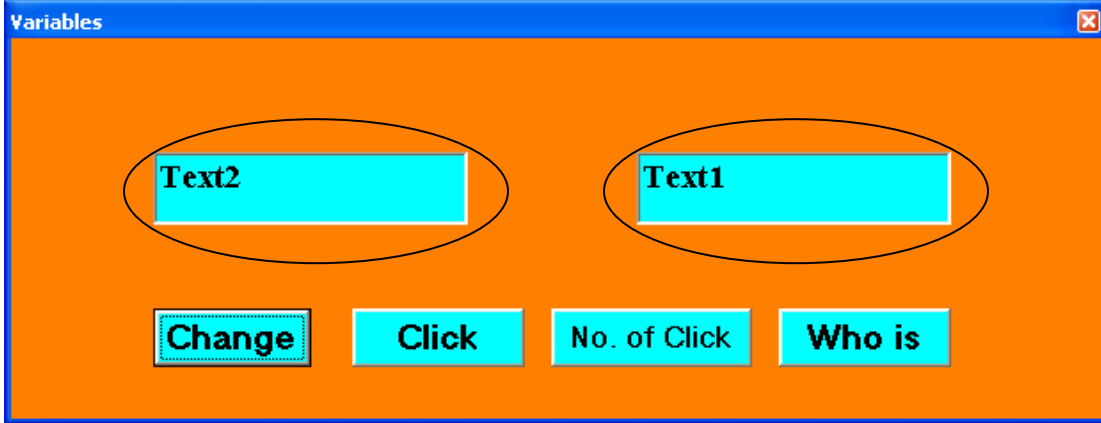
❖ يسمح للمستخدم ان يخطأ مرتين في كلمة السر، اذا أخطأ في المرة الثالثة سوف يظهر صندوق الرسالة التالي ويغلق البرنامج.



❖ في حالة كتابة كلمة السر الصحيحة ، تظهر الواجهة التالية (Form2)



- ❖ عند النقر على مفتاح Change سوف يؤدي الى التبديل بين محتويات الصندوق الاول مع محتويات الصندوق الثاني وكمايلي:



- ❖ يمكن للمستخدم ان ينقر عدد من المرات على المفتاح Click ، بعدها اذا نقر على المفتاح No. of Click سوف يظهر له مربع رسالة يحتوي على عدد النقرات التي قام بها المستخدم على المفتاح Click وكمايلي .:



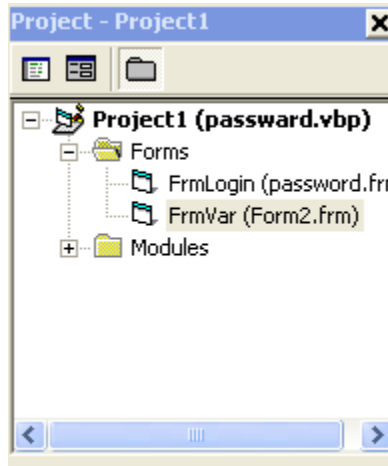
- ❖ عند النقر على مفتاح Who is تظهر رسالة تحتوي على اسم المستخدم الحالي وكمايلي:



## مراحل التنفيذ

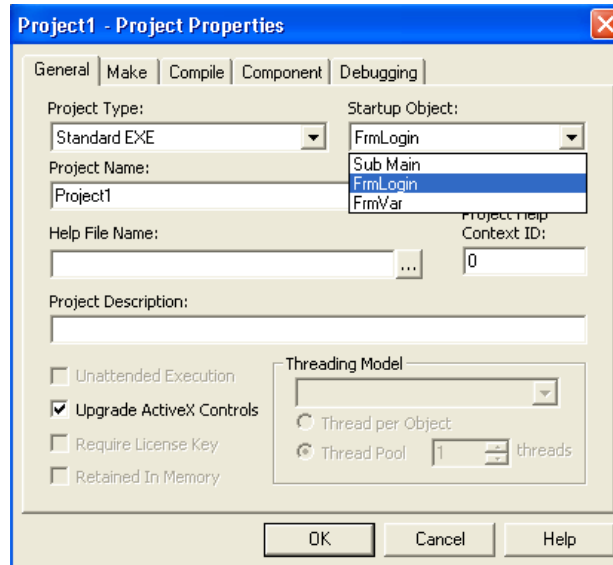
-----:-

1. نفتح مشروعا قياسيا جديدا.
  2. نظيف الى المشروع نموذجا ثانيا Form2، ووحدة نمطية جديدة Module.
  - ❖ لاضافة نموذج اخر الى المشروع ، نختار القائمة Project/Add Form.
  - ❖ لاضافة وحدة نمطية ، نختار القائمة Project / Add Module.
- ملاحظة - 1:- نستخدم مستكشف المشروع Project Explorer للتنقل بين مكونات المشروع المختلفة وكمايلي:



ملاحظه- 2:- نقوم خلال هذا المشروع بحفظ اربعة ملفات ، ملف للنموذج الاول , وملف للنموذج الثاني، وملف للوحدة النمطية module وملف اخر للمشروع.

ملاحظة - 3:- لجعل تنفيذ المشروع يبدأ من نموذج معين ، نختار القائمة Project / Project Properties ثم نختار النموذج المطلوب من القائمة Startup Object



3. ضبط خصائص النموذجين كمايلي: -

<b>FORM1 PROPERTIES</b>	
<b>Name</b>	<b>FrmLogin</b>
<b>BorderStyle</b>	<b>3-FixedDialog</b>
<b>Caption</b>	<b>Login</b>
<b>MinButton</b>	<b>False</b>

<b>FORM2 PROPERTIES</b>	
<b>Name</b>	<b>FrmVar</b>
<b>BorderStyle</b>	<b>2-Sizable</b>
<b>Caption</b>	<b>Variables</b>
<b>MinButton</b>	<b>True</b>

4. تصميم الخاصية **Border Style** : - تستخدم لتحديد شكل اطار النموذج ، حيث ان القيمة الافتراضية لهذه الخاصية هي **(2-Sizable)** اي انه يمكن للمستخدم تغيير ابعاد النموذج. اما اذا وضعنا القيمة **(3-FixedDialog)** فيصبح النموذج غير قابل للتمدد.
5. تصميم الخاصية **MinButton** تستخدم لالغاء او لابقاء مفتاح تصغير النموذج.
6. ننشأ عنصرين من النوع **(TextBox)** صندوق نص على النموذج الاول ونضبط خصائصه كمايلي:-

TEXTBOX PROPERTIES	
Name	TxtUserName
PasswordChar	
Text	

TEXTBOX PROPERTIES	
Name	TxtPassword
PasswordChar	*
Text	

7. تصميم الخاصية **PasswordChar**: تستخدم لجعل حرف ما يظهر عوضا عن الاحرف المدخله(عادة نضع الحرف \*).
8. ننشأ مفتاحين على النموذج الاول ونضبط الخصائص كمايلي:

COMMANDBUTTON	
Name	CmdOk
Cancel	False
Caption	Ok
Default	True

CommandButton	
Name	CmdCancel
Cancel	True
Caption	Cancel
Default	False

9. تصميم الخاصية **Cancel**: تستخدم لتحديد ان مفتاحا معينا على النموذج هو مفتاح الالغاء **Cancel** سوف يتم تنفيذ هذا المفتاح بالنقر عليه او بالضغط على مفتاح **ESC**.
10. تصميم الخاصية **Default**: تستخدم لتحديد ان مفتاحا معينا على النموذج هو المفتاح الافتراضي ، سوف يتم تنفيذ هذا المفتاح بالنقر عليه او بالضغط على المفتاح **Enter**.
11. ننشأ عنصرين من النوع **Label** على النموذج الاول ونضبط الخاصية **Caption** لوضع الاسم وكلمة السر.
12. ننشأ صندوقي نص على النموذج الثاني وندع الاسماء الافتراضيه لهما **Text1, Text2**.
13. ننشأ اربعة مفاتيح على النموذج الثاني ونضبط الخاصية **caption** لهما وكمايلي:  
**.Change, Click, No. of Click, Who is**
14. نبرمج المفتاح **Change** على النموذج الثاني: للتبديل بين محتوى صندوقي النص ، علينا ان نستخدم مكان وسيط لوضع محتوى الصندوق الاول فيه، ثم وضع محتوى الصندوق الثاني في الصندوق الاول، ثم نضع محتوى المكان الوسيط في الصندوق الثاني، يعتبر هذا الوسيط كوسيلة تخزين مؤقت وهو مандعوه بالمتغير **Variable**.

### المتغير Variable

هو عنوان منطقة محجوزه في الذاكرة ، يمكن الكتابة فيها والقراءة منها ، يتم التصريح او التعبير عن المتغير المحلي في داخل الاجراء **Procedure** باستخدام الامر التالي:

**Dim VariableName As Type**



حيث ان VariableName هو اسم المتغير و Type هو نوع المتغير، عند تنفيذ البرنامج سوف يتم حجز المكان الموافق للتغيير في الذاكرة، وسوف يتم تحرير هذا المكان المحجوز عند انتهاء تنفيذ البرنامج، ومن اهم انواع المتغيرات هي:

INTEGER	عدد صحيح
Long	عدد صحيح طويل
Single	عدد عشري
Double	عدد عشري ذو دقة مضاعفه
String	سلسلة نصية

في مثالنا هذا سوف نعرف متغيرا من نوع سلسلة نصية String لتخزين محتوى صندوق النص الاول فيه.

**Dim s As String**

حيث ان S هو الاسم المعطى للمتغير ، ويصبح الاجراء الخاص بالمفتاح Change كالآتي:

**Private Sub Command1\_Click()**

**Dim s As String**

**s = Text1.Text**

**Text1.Text = Text2.Text**

**Text2.Text = s**

**End Sub**

15. برمجة المفتاح Click على النموذج الثاني:-

المطلوب في التمرين كم مرة ينقر فيها المستخدم على هذا المفتاح، نحتاج اذا الى متغير لنضيف اليه واحد في كل مرة يتم النقر فيها على المفتاح Click ، لايمكن في هذه الحالة

استخدام الامر Dim في البرنامج لتعريف المتحول المطلوب ، أذ أن قيمة المتغير ستضيع بعد الخروج من البرنامج الخاص بالاجراء ، كما انه لايمكن الوصول الى هذا المتغير في مفتاح No. of Click .

نحتاج في هذه الحالة الى متغير عام على مستوى النموذج اي انه سوف يخلق عند فتح النموذج ويموت عند اغلاق النموذج ، كما انه تتشارك جميع البرامج الفرعية للنموذج.

لتعريف متغير عام للنموذج يجب وضع الامر Dim للمتغير في قسم التصريحات للنموذج. يكون هذا القسم في رأس صفحة البرمجة للنموذج. في مثالنا هذا التغير العام هو n في قسم التصريحات العامة للنموذج الثاني:

### Dim n As Integer

ونكتب الاجراء التالي على المفتاح Click:

```
Private Sub Command2_Click()
```

```
n = n + 1
```

```
End Sub
```

16. برمجة المفتاح No. of Click :- يهدف البرنامج الفرعي لهذا المفتاح الى اضهار قيمة

المتغير العام الذي يعد عدد النقرات No. of Click على المفتاح Click، نستخدم

الامر MsgBox لاطهار قيمة هذا المتغير وكمايلي:

```
Private Sub Command3_Click()
```

```
MsgBox n
```

```
End Sub
```

17. برمجة المفتاح Ok على النموذج الاول:

نريد ان نبرمج على هذا المفتاح كتابة اجراء يختبر كلمة السر المدخلة صحيحه ، كما

سنسمح للمستخدم المحاولة لثلاث مرات على الاكثر لادخال كلمة السر الصحيحه، اذا

نحتاج الى عد هذه المحاولات (اي عدد مرات النقر على مفتاح Ok)، يمكن ان نستخدم

لذلك متغيرا عاما على مستوى النموذج الاول كما فعلنا في مفتاح Click على النموذج

الثاني ، الا ان هذا المتغير لسنا بحاجة في هذه المره الا ضمن اجراء المفتاح Ok ،  
 لتعريف متغير محلي يحافظ على قيمته من استدعاء لاستدعاء اخر.  
 نستخدم المتغير المعرف باستخدام الامر Static في الاجراء Procedure بحيث  
 يحافظ على قيمته من استدعاء لآخر. مثال :

### Static i As Integer

18. لظهار النموذج Form نستخدم الامر Show ولاخفائه نستخدم الامر Hide.  
 مثلا لظهار النموذج FrmVar نستخدم الامر التالي:

FrmVar.Show

يمكن استخدام Me للدلالة على النموذج الحالي. مثال ذلك:

Me.Hide

تؤدي الى اخفاء النموذج الذي ينفذ فية الامر.

19. الامر SetFocus :- يستخدم لوضع التحديد على عنصر تحكم . مثال ذلك:

txtPassword.SetFocus

تضع التحديد (مؤشر الماوس على صندوق النص txtPassword)

20. الامر SendKeys يستخدم لاسال مجموعه من الاحرف للنافذه النشطه او الفعاله حاليا  
 كما لو كنا كتبناه من لوحة المفاتيح، مثال

SendKeys "{Home}+{End}"

تشبه النقر على المفتاح Home ثم النقر على المفاتيح معا Shift و End ، سيؤدي ذلك  
 في مثالنا هذا الى تحديد النص المكتوب في صندوق النص.

البرنامج او الاجراء الذي يكتب عند النقر على مفتاح Ok يكون كالاتي:

Private Sub CmdOk\_Click()

Static i As Integer

i = i + 1

UserName = TxtUserName.Text

If TxtPassword = "toto" Then

```

Me.Hide
FrmVar.Show
Else
If i = 3 Then
MsgBox "Finished your try", vbCritical, "Entrance"
End
Else
MsgBox "Try again, it's Wrong", vbExclamation, "Error"
TxtPassword.SetFocus
SendKeys "{Home}+{End}"
End If
End If
End Sub

```

21 برمجة المفتاح Cancel على النموذج الاول : نكتب الاجراء التالي:

```

Private Sub CmdCancel_Click()
End
End Sub

```

22 برمجة المفتاح من هو Who is :

نريد عند النقر على مفتاح Who is في النموذج الثاني لاضهار اسم المستخدم المدخل في مربع النص txtUserName ، في النموذج الاول. نحتاج في هذه الحالة الى متغير عام على مستوى المشروع، اي انه متغير يشارك كافة اجراءات النماذج في المشروع كي نخزن فيه اسم المستخدم من خلال اجراء Ok في النموذج الاول ، ثم نظهر محتوى هذا المتغير من خلال اجراء Who is في النموذج الثاني .

```

Module تعريف متغير عام على مستوى المشروع كلة نعرفه في وحدة نمطيه
باستخدام الامر Public وكمايلي :-

```

**Public Username As String**

سوف تتشارك كافة اجراءات النماذج في المشروع بالمتغير .UserName  
سابقا كتبنا اجراء المفتاح Ok على النموذج الاول الامر التالي:

**Username = txtUserName.Text**

لتخزين اسم المستخدم عند الدخول.وبالتالي يكفي ان نكتب على مفتاح Who is الاجراء التالي:

**Private Sub Command4\_Click()**

**MsgBox UserName**

**End Sub**

والذي سيؤدي تنفيذه الى اظهار صندوق رسالة يحتوي على محتوى المتغير  
.txtUserName

البرنامج كاملا مدون ادناه:-

الوحده النمطية Module

**Public Username As String**

النموذج الاول

**Private Sub CmdCancel\_Click()**

**End**

**End Sub**

**Private Sub CmdOk\_Click()**

**Static i As Integer**

**i = i + 1**

**UserName = TxtUserName.Text**

**If TxtPassword = "aaaa" Then**

**Me.Hide**

```
FrmVar.Show
Else
If i = 3 Then
MsgBox "Finished your try", vbCritical, "Error"
End
Else
MsgBox "Try again, it's Wrong", vbExclamation, "Error"
TxtPassword.SetFocus
SendKeys "{Home}+{End}"
End If
End If
End Sub
```

النموذج الثاني

```
Dim n As Integer

Private Sub Command1_Click()
Dim s As String
s = Text1.Text
Text1.Text = Text2.Text
Text2.Text = s
End Sub
```

```
Private Sub Command2_Click()
n = n + 1
End Sub
```

**Private Sub Command3\_Click()**

**MsgBox n**

**End Sub**

**Private Sub Command4\_Click()**

**MsgBox UserName**

**End Sub**

## 6-المشروع السادس

-----:-

### اوامر التكرار

سوف نتعلم من خلال هذا المشروع مايلي:-

❖ تعليمات التكرار الاساسية.

❖ For .. Statement

❖ Do .. While Statement

❖ Do .. Until Statement

❖ الطباعة على النموذج (Print)

❖ الطباعة على الطابعة (Printer)

الهدف من المشروع

-----:-

عند تنفيذ هذا المشروع سوف تظهر الواجهة التالية:





يؤدي النقر على احد المفاتيح الاربعة WithOut, For, Do While, Do Until الى كتابة الارقام من 0 الى 15 بخط ذي حجم متزايد وكل رقم بلون معين. كذلك يؤدي النقر على مفتاح جدول الضرب (Multi Table 5) 5 الى كتابة جدول الضرب الى حد 10 \* 5 والذي يساوي 50. اما مفتاح جدول الضرب Multi table يكتب على النموذج جداول الضرب العشره وكل جدول بلون يختلف عن الاخر. عند النقر على مفتاح الطباعة Printer سيتم طباعة جدول الضرب على الطباعة الموصولة مع الحاسبة الالكترونية. بينما يؤدي النقر على مفتاح Clear الى مسح النموذج ، وكذلك مفتاح Exit يدل على الخروج من التنفيذ الكلي.

مراحل التنفيذ

-----:-

- نفتح مشروعنا قياسيا جديدا.  
 - ننشئ تسعة مفاتيح ونضبط خاصية العنوان Caption لها كمايلي:-  
 Without, For, Do While, Do Until, Multi table 5, Multi Table, Printer, Clear, Exit.

- الامر Cls : يقوم بمسح النموذج.
- الامر Print:- يستخدم للكتابة على النموذج.
- الامر Me:- للدلالة على النموذج الحالي.
- الخاصية FontSize تحدد حجم الخط للكتابة.
- الخاصة ForeColor تحدد لون خط الكتابة.
- الدالة QBColor(Color\_Number) تعطي عدد صحيحا من Long يمثل اللون الذي رقمة Number\_color، ترقم الارقام من 0 الى 15 كمايبين في الجدول التالي:

0	BLACK	8	GRAY
1	Blue	9	Light Blue
2	Green	10	Light Green
3	Cyan	11	Light Cyan
4	Red	12	Light Red
5	Magenta	13	Light Magenta
6	Yellow	14	Light Yellow
7	White	15	Bright White

❖ يكتب الاجراء التالي على المفتاح بدون تكرار Without لطباعة الارقام من 0 الى 15 بحيث يتزايد حجم الخط من 10 فصاعدا ويكون كل سطر بلون مختلف.

**Private Sub Command1\_Click()**

**Me.Cls**

**Me.FontSize = 10 + 0**

**Me.ForeColor = QBColor(0)**

**Me.Print 0**

**Me.FontSize = 10 + 1**

**Me.ForeColor = QBColor(1)**

**Me.Print 1**

**Me.FontSize = 10 + 2**

**Me.ForeColor = QBColor(2)**

**Me.Print 2**

**Me.FontSize = 10 + 3**

**Me.ForeColor = QBColor(3)**

**Me.Print 3**

**Me.FontSize = 10 + 4**

**Me.ForeColor = QBColor(4)**

**Me.Print 4**

**Me.FontSize = 10 + 5**

**Me.ForeColor = QBColor(5)**

**Me.Print 5**

**Me.FontSize = 10 + 6**

**Me.ForeColor = QBColor(6)**

**Me.Print 6**

**Me.FontSize = 10 + 7**

**Me.ForeColor = QBColor(7)**

**Me.Print 7**

**Me.FontSize = 10 + 8**

**Me.ForeColor = QBColor(8)**

**Me.Print 8**

**Me.FontSize = 10 + 9**

**Me.ForeColor = QBColor(9)**

**Me.Print 9**

**Me.FontSize = 10 + 10**

**Me.ForeColor = QBColor(10)**

**Me.Print 10**

**Me.FontSize = 10 + 11**

**Me.ForeColor = QBColor(11)**

**Me.Print 11**

**Me.FontSize = 10 + 12**

Me.ForeColor = QBColor(12)

Me.Print 12

Me.FontSize = 10 + 13

Me.ForeColor = QBColor(13)

Me.Print 13

Me.FontSize = 10 + 14

Me.ForeColor = QBColor(14)

Me.Print 14

Me.FontSize = 10 + 15

Me.ForeColor = QBColor(15)

Me.Print 15

End Sub

لاحظ اننا قمنا بتكرار الاوامر التالية :

Me.FontSize = 10 + i

Me.ForeColor = QBColor(i)

Me.Print i

من اجل ان نجعل  $i$  تبدأ من الصفر وتتزايد كل مرة بقيمة مقدارها واحد الى ان تصل الى القيمة 15  
توجد في لغة البرمجة Visual Basic عدد من اوامر التكرار ، سوف نعالج ثلاثة من جمل التكرار في  
هذا المثال بهدف استعراض وفهم هذه الحلقات التكرارية.

- برمجة المفتاح For

تاخذ الحلقة التكرارية For الشكل التالي:

```

For counter = start to end [Step]
[statements]
...
[statements]
Next [counter]

```

الاقواس المربعة لاتستخدم في البرمجة بل تستخدم في الشكل العام للامر للدلالة على ان جزءا من الامر يكون اختياري اي يمكن كتابته او تركته.

حيث ان :-

- Counter هو متغير يدعى بالعداد.
- Start القيمة الابتدائية للعداد.
- End القيمة النهائية للعداد.
- Step هو مقدار الزيادة او النقصان لقيمة العداد

ملاحظة

-----:-

يمكن ان يكون Step موجبا او سالبا. اذا كان موجبا فان Start يجب ان يكون اصغر او يساوي End والا فان تعليمات الحلقة لن تنفذ. امام اذا كان سالبا فان start يجب ان يكون اكبر او يساوي كي تنفذ تعليمات الحلقة التكرارية، اذا لم يحدد مقدار الزيادة او النقصان عن طريق الابعاز step فان كل لغات البرمجة تعتبر الزيادة بمقدار وحدة واحدة فقط، البرنامج التالي يقوم بتنفيذ اجراء الحلقة التكرارية For.

```
Private Sub Command2_Click()
```

```
Dim i As Integer
```

```
Me.Cls
```

```
For i = 0 To 15
```

```
Me.FontSize = 10 + i
```

```
Me.ForeColor = QBColor(i)
```

```
Me.Print i
```

```
Next i
```

```
End Sub
```

## - برمجة المفتاح Do While

تأخذ جملة التكرار Do .. While الشكل العام التالي:

**Do While Condition**

[Statement]

...

[Statement]

**Loop**

عندما يصل Visual Basic لتنفيذ هذه الحلقة التكرارية فإنه سيبدأ أولاً باختبار الشرط *Condition* فإذا وجد غير محقق *False* فإنه سيقفز فوراً لما بعد سطر *Loop* أي لن ينفذ أوامر الحلقة التكرارية *Statements* أما إذا وجد *True* فإنه سينفذ التعليمات *Statements* ثم يعود إلى سطر *Do While* ليختبر الشرط ثانية ويعاود التنفيذ إذا كان الشرط *True* أو يخرج من الحلقة التكرارية إذا كان الشرط *false* وهكذا، وبالتالي فإن تعليمات الحلقة التكرارية ستنفذ عدد من المرات طالما بقي الشرط *True* ولن تنفذ أبداً إذا كان الشرط منذ البداية *False*.  
الاجراء التالي الخاص بهذه الحلقة التكرارية موضح كمايلي:-

**Private Sub Command3\_Click()****Dim i As Integer****Me.Cls****Do While i <= 15****Me.FontSize = 10 + i****Me.ForeColor = QBColor(i)****Me.Print i****i = i + 1****Loop****End Sub**

## - برمجة المفتاح Do Until

تأخذ الحلقة التكرارية Do .. until الشكل العام التالي:

**Do until condition**

**[statements]**

...

**[statements]**

**Loop**

يقوم Visual Basic بتكرار هذه الحلقة حتى يصبح الشرط محققا.

الاجراء التالي يقوم بتنفيذ هذه الحلقة التكرارية.

```
Private Sub Command4_Click()
```

```
Dim i As Integer
```

```
Me.Cls
```

```
i = 0
```

```
Do Until i = 15
```

```
Me.FontSize = 10 + i
```

```
Me.ForeColor = QBColor(i)
```

```
Me.Print i
```

```
i = i + 1
```

```
Loop
```

```
End Sub
```

- برمجة المفتاح **Multi table 5** :- يطبع هذا المفتاح جدول الضرب لحد 5 على النموذج ، حيث نستخدم الحلقة لتكرارية For لذلك. الاجراء التالي يحقق الجدول.

```
Private Sub Command5_Click()
```

```
Me.Cls
```

```
Dim i As Integer
```

```
Me.FontSize = 12
```

```
Me.ForeColor = QBColor(0)
```

```
For i = 1 To 10
```

```
    Me.Print i & " * 5 = " & i * 5
```

```
Next i
```

```
End Sub
```

- برمجة المفتاح **Multi Table** :- يطبع هذا المفتاح جدول الضرب على النموذج ، بحيث يكون كل جدول بلون معين. الاجراء التالي يقوم بتنفيذ هذه العملية.

```
Private Sub Command6_Click()
```

```
Dim i As Integer, j As Integer
```

```
Me.Cls
```

```
Me.FontSize = 10
```

```
For i = 1 To 10
```

```
    Me.ForeColor = QBColor(i)
```

```
    For j = 1 To 10
```

```
        Me.CurrentX = 800 * i
```

```
        Me.CurrentY = 500 * j
```

```
        Me.Print j & " * "; i & " = " & i * j
```

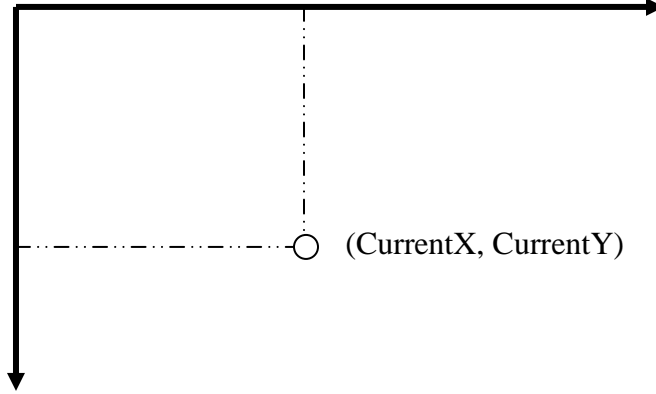
```
    Next j
```

```
Next i
```

```
End Sub
```



- تصميم محاور احداثيات النموذج  
تصمم محاور احداثيات النموذج كمايلي:



الكتابة في مكان معين على النموذج يجب تحديد نقطة محور X عن طريق النقطة CurrentX حيث تتراوح قيم هذا المحور بين الصفر وعرض النموذج width ونقطة محور Y عن طريق النقطة CurrentY التي تتراوح بين الصفر وارتفاع النموذج Height.

يستخدم مصطلح Twip كوحدة قياس اساسية في VB حيث ان كل 576 تويب تعادل واحد سنتيمتر ، في مثالنا هذا نعتبر ان المسافة الافقية الفاصلة بين جدول الضرب واخر تساوي 800 تويب وان المسافة العمودية بين سطر واخر تساوي 500 تويب.

نحتاج في الاجراء الذي يقوم بتنفيذ جدول الضرب الى حلقتين تكراريتين : حلقة تكرارية i من الواحد الى العشرة لجدول الضرب العشرة ، ومن اجل كل جدول ضرب i نحتاج الى j من الواحد الى العشرة لطباعة اسطر جدول ضرب i العشرة. يكون الاجراء بالشكل التالي:

```
Private Sub Command6_Click()
Dim i As Integer, j As Integer
Me.Cls
Me.FontSize = 10
For i = 1 To 10
    Me.ForeColor = QBColor(i)
    For j = 1 To 10
        Me.CurrentX = 800 * i
        Me.CurrentY = 500 * j
        Me.Print j & " * "; i & " = " & i * j
    Next j
Next i
End Sub
```

- برمجة مفتاح الطبع:-

يطبع هذا المفتاح جدول الضرب على الطابعة ، بحيث يكون كل جدول بلون، الاجراء الخاص بتنفيذ هذا الامر موضح كالآتي:-  
لاحظ اوجة التشابهه مع اجراء جدول الضرب.

```
Private Sub Command7_Click()
Dim i As Integer, j As Integer
Printer.FontSize = 10
For i = 1 To 10
Printer.ForeColor = QBColor(i)
For j = 1 To 10
Printer.CurrentX = 800 * i
Printer.CurrentY = 500 * j
Printer.Print j & " * "; i & " = " & i * j
Next j
Next i
End Sub
```

يستخدم ايعاز الطبع Printer للقيام بجميع عمليات الرسم والطباعة على الطابعة، يقبل النموذج جميع الطرق المعرفة على الكائن Printer ، لذا فاننا نقوم عمليا بكتابة اجراءات الرسم والطباعة على النموذج اولا، وبعد الوصول الى الشكل النهائي المطلوب نستبدل اسم النموذج في الاجراءات ب Printer وكما موضح في البرنامج الخاص بالطبع اعلاه.

من اهم خصائص الكائن Printer موضحة في الجدول التالي:

CurrentX	الاحداثيات الافقية
CurrentY	الاحداثيات العمودية
FontName	اسم خط الكتابة
FontSize	حجم خط الكتابة
FontCount	عدد الخطوط المتاحة
Fonts (n)	اسم الخط ذي الرقم n حيث ترقم اسماء الخطوط من 0 الى FontCount-1
Orientation	اتجاه صفحة الطباعة : 1 راسي Portrait و 2 Landscape

ومن اهم طرق الكائن Printer هي:

Circle (X, Y), R, C	رسم دائرة
Line (X1, Y1)-(X2,Y2)	رسم خط
Pset (X,Y),C	رسم نقطة
Print	الطباعة عند الاحداثيات المحددة
printPicture	نقل صورة للطباعة
EndDoc	اخراج الصفحة من الطباعة بعد الطباعة عليها

- مفتاح المسح Clear

يستخدم الاجراء التالي لمسح النموذج

```
Private Sub Command9_Click()
```

```
Me.Cls
```

```
End Sub
```

Exit - مفتاح الخروج

Private Sub Command8\_Click()

End

End Sub

## 7-المشروع السابع

-----:-

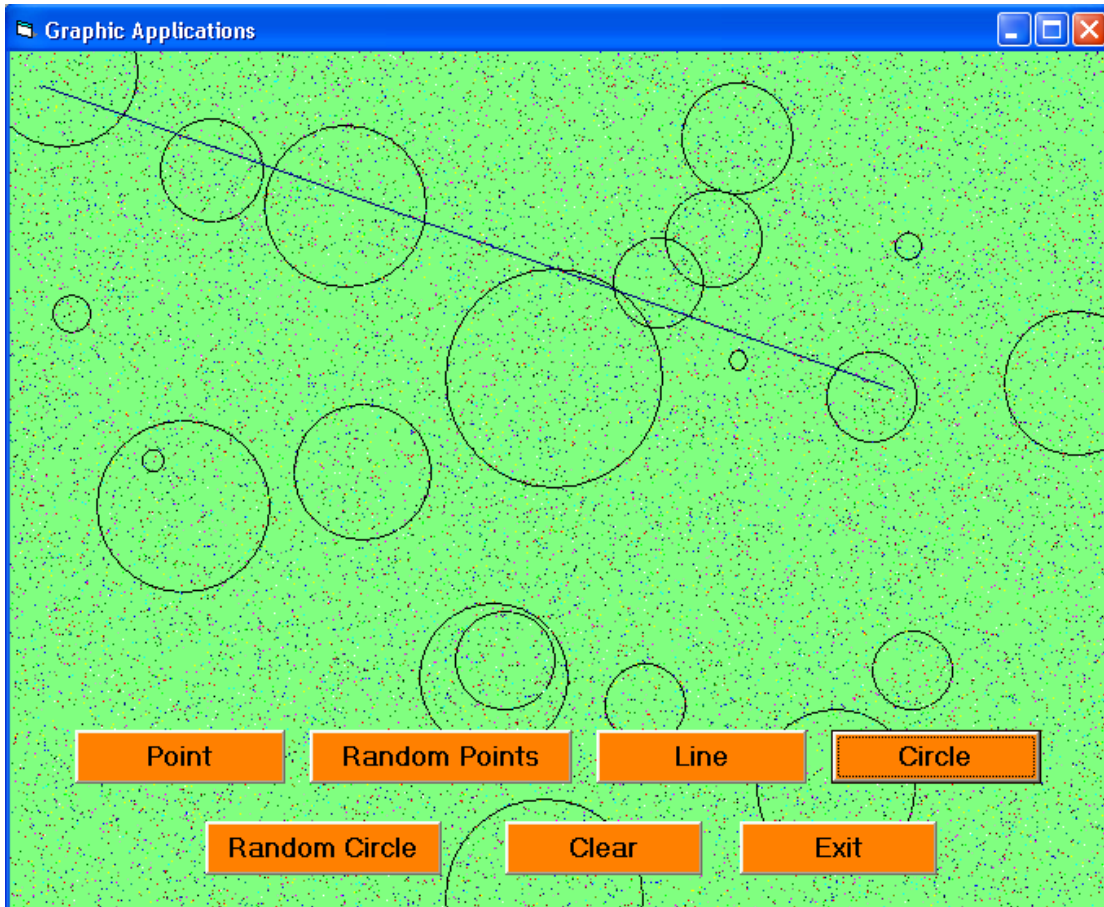
اسم المشروع : (رسم النقطة ، النقاط العشوائية، الخط، الدائره، الدوائر العشوائية)

سوف نتعلم من خلال هذا المشروع مايلي:

- ❖ رسم النقطة.
- ❖ رسم النقاط العشوائية.
- ❖ رسم الخط.
- ❖ رسم الدائره.
- ❖ رسم الدوائر العشوائية.

الهدف من المشروع:

عند تنفيذ هذا المشروع تظهر الواجهة التالية:



مراحل التنفيذ

-----:-

❖ ننشي نموذجاً قياسياً جديداً، ونضبط خصائصه كمايلي:

FORM1 PROPERTIES	
Name	Graphics_1
BorderStyle	2-Sizable
Caption	Graphic Applications
BackColor	White Green اخضر فاتح
Height	8400
Width	10240
Window State	0 – Normal

❖ ننشي سبعة مفاتيح اوامر Command Buttons ونضبط الخصائص لها كمايلي:

COMMAND BUTTONS 1	
Name	Point
BackColor	Orange
Caption	Point

COMMAND BUTTONS 2	
Name	CmdRandPoint
BackColor	Orange
Caption	Random Point

COMMAND BUTTONS 3	
Name	CmdLine
BackColor	Orange
Caption	Line

<b>command Buttons 4</b>	
<b>Name</b>	<b>CmdCircle</b>
<b>BackColor</b>	<b>Orange</b>
<b>Caption</b>	<b>Circle</b>

<b>COMMAND BUTTONS 5</b>	
<b>Name</b>	<b>Command1</b>
<b>BackColor</b>	<b>Orange</b>
<b>Caption</b>	<b>Random Circle</b>

<b>COMMAND BUTTONS 6</b>	
<b>Name</b>	<b>CmdClear</b>
<b>BackColor</b>	<b>Orange</b>
<b>Caption</b>	<b>Clear</b>

<b>COMMAND BUTTONS 7</b>	
<b>Name</b>	<b>CmdExit</b>
<b>BackColor</b>	<b>Orange</b>
<b>Caption</b>	<b>Exit</b>

❖ البرنامج الخاص بالرسومات موضح كمايلي:

```
Private Sub CmdCircle_Click()  
Graphics_1.AutoRedraw = True  
Circle (5000, 3000), 1000  
End Sub
```

```
Private Sub CmdClear_Click()  
Graphics_1.Cls  
End Sub
```

```
Private Sub CmdExit_Click()  
End  
End Sub
```

```
Private Sub CmdLine_Click()  
Graphics_1.AutoRedraw = True  
Line (300, 310)-(8140, 3100), QBColor(1)  
End Sub
```

```
Private Sub CmdPoint_Click()  
Graphics_1.AutoRedraw = True  
PSet (1500, 600), QBColor(4)  
End Sub
```

```
Private Sub CmdRandPoint_Click()  
Dim r, x, y, c As Integer  
For r = 1 To 20000  
    x = Rnd * Graphics_1.ScaleWidth  
    y = Rnd * Graphics_1.ScaleHeight  
    c = Rnd * 15  
    PSet (x, y), QBColor(c)  
Next r  
End Sub
```



```
Private Sub Command1_Click()  
Dim r, x, y, c As Integer  
For r = 1 To 20  
    x = Rnd * Graphics_1.ScaleWidth  
    y = Rnd * Graphics_1.ScaleHeight  
    c = Rnd * 1000  
    Circle (x, y), c  
Next r  
End Sub
```

8-المشروع الثامن

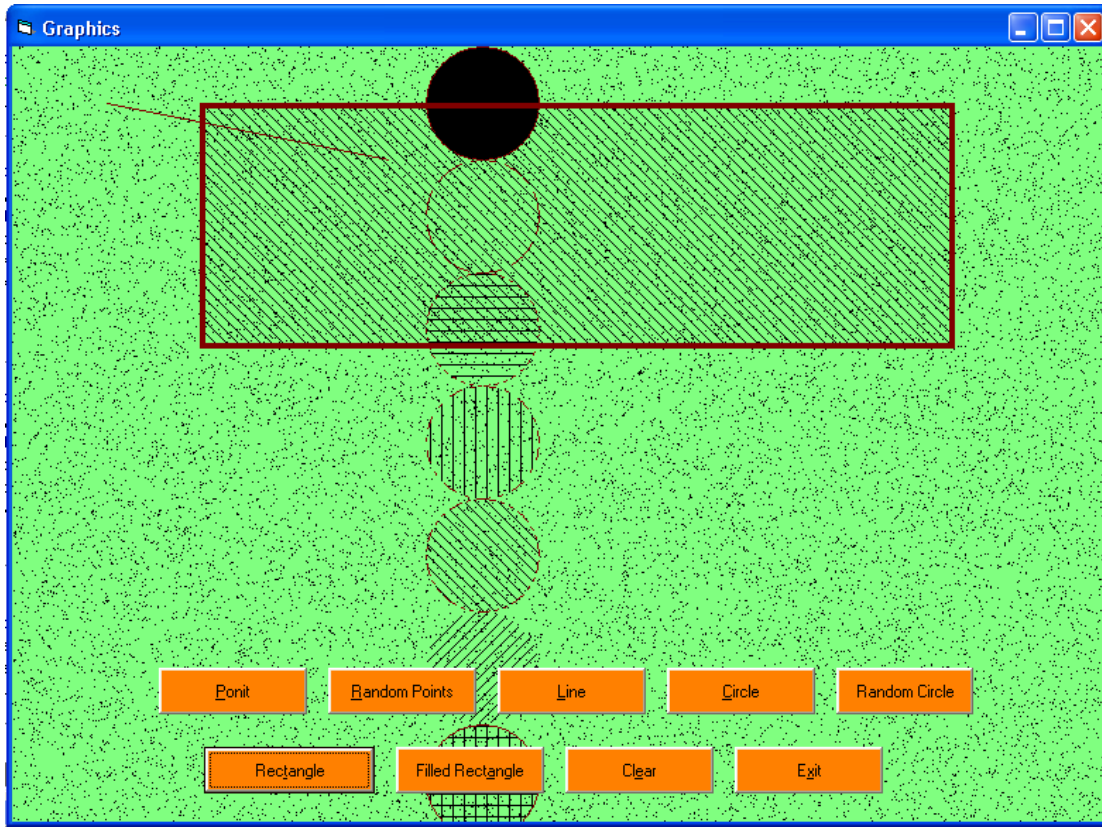
-----:-

(رسم الاشكال الهندسية مع امكانية تغير سمك الخط، لون الخط، مليء الشكل)

الهدف من المشروع

-----:-

عند تنفيذ المشروع تظهر الواجهه التاليه:



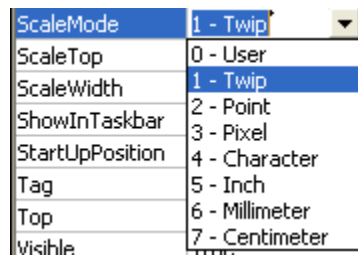
## اوامر الرسوم المباشرة

-----:-

تحتوي **Visual Basic** على مجموعه كبيره من الاوامر التي تمكننا من رسم الاشكال المختلفة فوق النموذج، وقبل ان نبدأ في معرفة هذه الاوامر علينا اولا معرفة احداثيات النموذج، عند بداية تشغيل **Visual Basic** هناك وحدة افتراضية خاصة لقياس احداثيات النموذج تسمى **TWIP** وكل انج واحد يعادل **1440 TWIP**، من الممكن ان تسبب لك هذه الوحدة بعض الازعاجات لعدم ضبط حدود الرسم لذلك يمكن تغييرها الى وحدات قياس اخرى كما مدون في الجدول التالي:

الشرح	الوحده
ويحتوي كل انج على <b>1440 TWIP</b>	<b>TWIP</b>
ويحتوي كل انج على <b>72 POINTS</b>	<b>POINTS</b>
تعتبر هذه الوحده من اكثر الوحدات شيوعا واستخداما منذ بداية التعامل مع لغات البرمجة المختلفة، ويتحدد عدد النقاط <b>PIXELS</b> التي يحتويها النموذج على دقة الشاشة <b>Resolution</b>	<b>PIXELS</b>
حرف	<b>CHARACTER</b>
انج	<b>INCHES</b>
مليمتر	<b>MILIMETERS</b>
سنتيمتر	<b>CENTIMETERS</b>

نتكمن من ضبط وحدات القياس المطلوبة من خلال استخدام الخاصية **Scale Mode** الموجوده ضمن خصائص النموذج الموضحة في الشكل التالي:



❖ الامر PSet

يقوم هذا الامر او الابعاز برسم نقطة على النموذج بالاحداثيات التي تحددها انت.  
الصيغة العامه لها:

Pset (x, y), Color

حيث ان x هي الاحداثي الافقي لمكان النقطة على النموذج.

وان y هي الاحداثي العمودي لمكان النقطة على النموذج.

Color هو لون النقطة ويمكن تغيير لون النقطة عن طريق الدالة QBColor(i)

حيث ان i يمثل لون النقطة من 0 الى 15 لون كما موضح في الجدول التالي:

اللون	الرقم
الاسود	0
الازرق	1
الاخضر	2
البنّي	3
الاحمر	4
البرتقالي	5
الاصفر	6
الابيض	7
الرمادي	8
الازرق الفاتح	9
الاخضر الفاتح	10
السماوي	11
الاحمر الفاتح	12
الروز	13
الاصفر الفاتح	14
الابيض الناصع	15

❖ الدالة Rnd لتوليد النقاط العشوائية اي Random.

❖ الامر Line

يقوم هذا الامر او الابعاز برسم خط على النموذج بالاحداثيات التي تحدد لرسمه  
الصيغة العامة له :

Line (x1, y1) - (x2, y2), QBColor(i)

❖ لرسم مربع او مستطيل نستخدم ابعاز Line مع اضافة الحرف B للدلالة على انه Box وحسب الاحداثيات المعطاة له، الصيغة العامة له:

Line (x1, y1) - (x2, y2), QBColor(i), B

❖ واذا اردنا الصندوق مملوء نستخدم الابعاز BF حيث ان F للدلالة على انه Filled اي مملوء  
❖ ابعاز رسم الدائرة: نستخدم الابعاز Circle لرسم الدائرة، الصيغة العامة له هي:

Circle (x, y), Radius, QBColor(i)

حيث ان Radius هي نصف قطر الدائرة.

❖ الخاصية FillStyle

تستخدم هذه الخاصية لتحديد الشكل الداخلي للشكل المرسوم سواء كان دائرة او مستطيل، القيم الموضحة في الجدول التالي تبين القيم المستخدمة لتحديد الشكل الداخلي للرسم:

الرقم	الاسم
0	Solid معتمه
1	Transparant شفافه
2	Horizontal Lines خطوط أفقية على طول الشكل
3	Vertical Lines خطوط عمودية على طول الشكل
4	UpWard Diagonal Lines خطوط مائلة للأعلى
5	DownWard Diagonal Lines خطوط مائلة للأسفل
6	Cross Hatch خطوط متقاطعه
7	Diagonal Cross Hatch خطوط متقاطعه مائلة

## ❖ الخاصية DrawStyle

تستخدم هذه الخاصية لتحديد شكل الخط المرسوم ، القيم الموضحة في الجدول التالي تبين شكل الخط المطلوب:

الرقم	الاسم
0	خط متصل Solid
1	خط متقطع Dash
2	خط يتكون من مجموعة نقط Dot
3	خط يتكون من خط يليه نقطه Dash Dot
4	خط يتكون من خط يليه نقطتين Dash Dot Dot
5	خط شفاف غير مرئي Transparant
6	خط ممتليء من الداخل Inside Solid

## ❖ الخاصية DrawWidth

تحدد هذه الخاصية سمك الخط للرسم

❖ كل الخواص الموضحة في الاعلى تم استخدامها في البرنامج، يجب علينا ان نمنع في كيفية استخدامها داخل البرنامج وبالتالي نكون ملزمين عن فهم طريقة استخدامها .  
البرنامج التالي يقوم بتنفيذ شكل النموذج الموجود في البداية:

```
Private Sub CmdCircle_Click()
```

```
Dim r, x, y, c, z
```

```
z = 0
```

```
For r = 0 To 6
```

```
    FrmGraphics.FillStyle = r
```

```
    FrmGraphics.DrawStyle = r
```

```
    x = 5000
```

```
    y = 600 + z
```

```
    Circle (x, y), 600, QBColor(4)
```

```
    z = z + 1200
```

```
Next r
```

```
End Sub
```

```
Private Sub CmdClear_Click()
```

```
FrmGraphics.Cls
```

```
End Sub
```

```
Private Sub CmdExit_Click()
```

```
End
```

```
End Sub
```

```
Private Sub CmdLine_Click()
```

```
Line (1000, 600)-(4000, 1200), QBColor(4)
```

```
End Sub
```

```
Private Sub CmdPoint_Click()
```

```
Rem FrmGraphics.DrawWidth = 4
```

```
PSet (1000, 600), QBColor(4)
```

```
End Sub
```

```
Private Sub CmdRndCircle_Click()
```

```
Dim r, x, y, c As Integer
```

```
For r = 1 To 20
```

```
    x = Rnd * FrmGraphics.ScaleWidth
```

```
    y = Rnd * FrmGraphics.ScaleHeight
```

```
    c = Rnd * 1000
```

```
    Circle (x, y), c
```

```
Next r
```

```
End Sub
```

```
Private Sub CmdRndPoints_Click()
```

```
Dim r, x, y, c
```

```
For r = 1 To 20000
```

```
    x = Rnd * FrmGraphics.ScaleWidth
```

```
    y = Rnd * FrmGraphics.ScaleHeight
```

```
PSet (x, y), QBColor(c)
Next r
End Sub

Private Sub Command1_Click()
FrmGraphics.DrawWidth = 4
FrmGraphics.FillStyle = 4
Line (2000, 600)-(10000, 3200), QBColor(4), B
End Sub

Private Sub Command2_Click()
FrmGraphics.DrawWidth = 4

Line (2000, 600)-(10000, 3200), QBColor(4), BF
End Sub
```



## 9-المشروع التاسع

-----:-

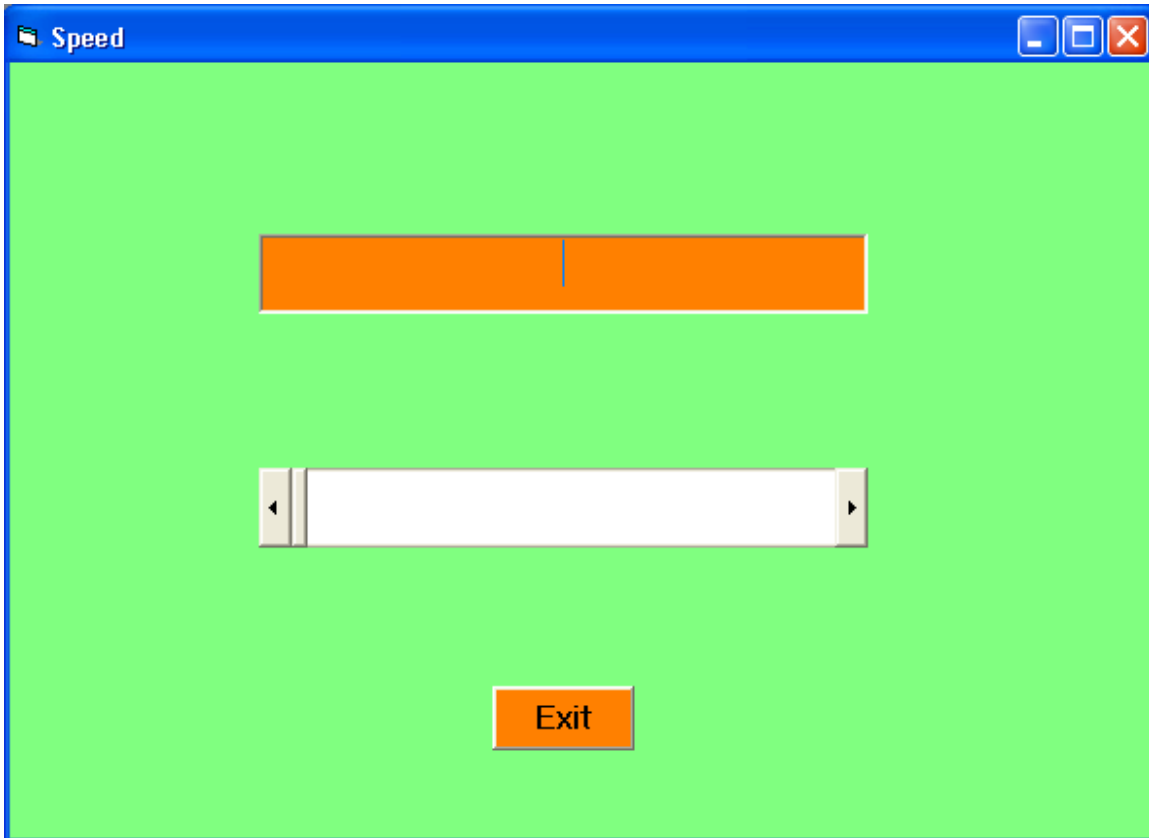
اسم المشروع : محاكاة السرعة Speed

سوف نتعلم من خلال هذا المشروع استخدام الكائن Horizontal Scroll Bar اي شريط الطي الافقي

الهدف من المشروع

-----:-

عند تنفيذ هذا المشروع سوف تظهر الواجهة التالية:



خطوات تنفيذ المشروع

-----:-

❖ ننشيء مشروعاً قياسياً جديداً ونضبط خصائصه كمايلي:

FORM1 PROPERTIES	
Name	Form1
BorderStyle	2-Sizable
Caption	Speed
BackColor	اخضر فاتح White Green
Height	6480
Width	8925
Window State	0 – Normal

❖ ننشيء عنصر واحد من نوع Text Box ونضبط خصائصه كمايلي:

TEXT BOX1	
Name	TxtSpeed
Alignment	2 – Center
BackColor	Orange
Text	

❖ ننشيء عنصر من نوع HScrollBar اي شريط الطي الافقي ونضبط اسمه HSpeedScroll

❖ ننشيء عنصر من نوع Command Button ونضبط عنوانه ب Exit

❖ عند النقر على Text Box نكتب البرنامج التالي:

Private Sub TxtSpeed\_Change()

TxtSpeed.Text = Str(HSpeedScroll.Value) + " km/hour"

End Sub

❖ عند النقر على مفتاح شريط الطي الافقي نكتب الاجراء التالي:

```
Private Sub HSpeedScroll_Change()
```

```
TxtSpeed_Change
```

```
End Sub
```

❖ عند النقر على مفتاح Exit نكتب الاجراء التالي:

```
Private Sub Command1_Click()
```

```
End
```

```
End Sub
```

10 - المشروع العاشر

-----:-

(حساب الفائدة المركبة في البنك)

سنتعلم من خلال هذا المشروع:

- ❖ الحلقات التكرارية
- ❖ حساب الفائدة المركبة

الهدف من المشروع

-----:-

عند تنفيذ هذا المشروع تظهر الواجهة التالية:

تسمح هذه الواجهة باجراء بعض حسابات الفائدة المركبة:

- ❖ في القسم الايمن من الواجهة اعلاه يقوم المستخدم بادخال المبلغ والفائدة السنوية وعدد السنوات، وعند النقر على المفتاح Ok سيتم حساب المبلغ الناتج بعد مرور السنوات المحددة.

❖ في القسم الايسر يقوم المستخدم بادخال المبلغ والفائدة لسنوية والمبلغ المطلوب، وعند النقر على المفتاح Years سيتم حساب عدد السنوات اللازم كي يصبح المبلغ اكبر او يساوي المبلغ المطلوب.

مراحل التنفيذ \_\_\_\_\_ ذ

-----:-

- ❖ نفتح مشروعا قياسييا جديدا.
- ❖ ننشئ عنصرين من النوع Shape على شكل مستطيل ذي حواف مستديرة، نضع عناصر كل مسألة في مستطيل.
- يستخدم عنصر التحكم Shape لرسم شكل هندسي على النموذج، وتحدد الخاصية Shape للشكل الهندسي وكمايلي:

0 - RECTANGLE	مستطيل
1 - Square	مربع
2 - Oval	قطع ناقص
3 - Circle	دائرة
4 - Rounded Rectangle	مستطيل ذو حواف مستديرة
5 - Rounded Square	مربع ذو حواف مستديرة

❖ ضبط خصائص النموذج كمايلي:

FORM1 PROPERTIES	
Name	Form1
BorderStyle	2-Sizable
Caption	Benefit
BackColor	اخضر فاتح White Green
Height	7710
Width	11025
Window State	0 - Normal

❖ ننشئ شكلين من النوع Shape ونضبط الخصائص كمايلي:

SHAPE	
Name	Shape1
Shape	4 – Rounded Rectangle

SHAPE	
Name	Shape2
shape	4 – Rounded Rectangle

❖ ننشئ اربعة صناديق نص لوضع المبلغ (TxtAmount) والفائدة (TxtRate) وعدد السنين (TxtYears) والمبلغ الناتج (TxtOut) في المستطيل الايمن. نضبط الخصائص كمايلي:

TEXT BOX	
Name	TxtAmount
Text	

TEXT BOX	
Name	TxtRate
Text	

TEXT BOX	
Name	TxtYears
Text	

TEXT BOX	
Name	TxtOut
Text	

- ❖ ننشئ مفتاح الاوامر Ok في المستطيل الايمن لاجراء عملية الحساب.
- ❖ ننشئ اربعة صناديق نص لوضع المبلغ (TxtAmount1) والفائدة (TxtRate1) وعدد السنين (TxtYears1) والمبلغ الناتج (TxtOut1) في المستطيل الايسر.

TEXT BOX	
Name	TxtAmount1
Text	

TEXT BOX	
Name	TxtRate1
Text	

TEXT BOX	
Name	TxtYears1
Text	

TEXT BOX	
Name	TxtOut1
Text	

- ❖ ننشئ مفتاح الاوامر Command Button ونضبط العنوان باسم Years في المستطيل الايسر لاجراء عملية الحساب.
  - ❖ ننشئ التسميات Labels الظاهره في الواجهة ونضبط عناوينها Caption.
  - ❖ برمجة المفتاح Ok:
- لحساب المبلغ الناتج بعد عدة سنوات، يجب ان يكون من اجل كل سنة اضافة الفائدة الى المبلغ الناتج الذي يكون مبلغ اول سنة قادمة (فائدة مركبة او تراكمية) وهكذا لبقية السنين.  
اي انة مثلا اذا كان المبلغ هو 100 والفائدة هي 10 وعدد السنوات 5 سوف يتم الحساب كمايلي:

السنة	مبلغ اول سنة	الفائدة	مبلغ اخر سنة
1	100	$10=100\backslash 10*100$	$110=10+100$
2	110	$11=100\backslash 10*110$	$121=11+110$
3	121	$12,1=100\backslash 10*121$	$133,1=12,1+121$
4	133,1	$13,31=100\backslash 10*133,1$	$146,41=13,31+133,1$
5	146,41	$14,641=100\backslash 10*146,41$	$161,051=14,641+146,41$

اي انه يجب تكرار العملية التالية بعدد السنوات:

مبلغ نهاية السنة = مبلغ بداية السنة + مبلغ بداية السنة  $\times$  الفائدة \ 100.

من الانسب في هذه الحالة استخدام تعليمة التكرار For ، اذا اردنا ان نعرف عدد مرات التكرار ( اي عدد السنوات).

نكتب الاجراء التالي على مفتاح الاوامر Ok وكمالي:

```
Private Sub Command1_Click()
```

```
Dim a As Single
```

```
Dim r As Single
```

```
Dim y As Integer
```

```
Dim b As Single
```

```
Dim i As Integer
```

```
a = Val(TxtAmount.Text)
```

```
r = Val(TxtRate.Text)
```

```
y = Val(TxtYears.Text)
```

```
b = a
```

```
For i = 1 To y
```

```
    b = b + b * r / 100
```

```
Next i
```

```
TxtOut.Text = b
```

```
End Sub
```



❖ برمجة المفتاح Years :

في هذه الحالة لدينا المبلغ والفائدة ، ونريد ان نحسب عدد السنوات اللازمة كي يصبح هذا المبلغ اكبر من مبلغ معطى

اي اننا يجب ان نكرر عملية اضافة الفائدة الى المبلغ الى ان يصل الى اكبر من المبلغ المطلوب، لاحظ اننا لانعرف مسبقا عدد مرات التكرار ، وبالتالي لايمكن في هذه الحالة استخدام الحلقة التكرارية For .  
الحلقة التكرارية المناسبة في هذه الحالة هي الحلقة التكرارية Do .. While اذا سنكرر عملية اضافة الفائدة الى المبلغ طالما كان اصغر من المبلغ المطلوب، وعندما يصبح المبلغ اكبر او يساوي المبلغ المطلوب سوف تتوقف الحلقة التكرارية وسوف يكون عدد مرات التكرار هو بقدر عدد السنوات.  
نكتب الاجراء التالي بعد الضغط على مفتاح الاوامر Years:

Private Sub Command2\_Click()

Dim a As Single

Dim r As Single

Dim y As Integer

Dim b As Single

a = Val(TxtAmount1.Text)

r = Val(TxtRate1.Text)

b = Val(TxtOut1.Text)

y = 0

Do While a &lt; b

a = a + a \* r / 100

y = y + 1

Loop

TxtYears1.Text = y

End Sub

❖ ننشئ مفتاح الاوامر Command Button رقم 3 ونضبط العنوان له ب Clear

نبرمج هذا المفتاح كمايلي لغرض المسح في حالة اعادة تنفيذ البرنامج مرة اخرى لغرض اجراء عملية حسابات اخرى:

```
Private Sub Command3_Click()  
    TxtAmount = ""  
    TxtRate = ""  
    TxtYears = ""  
    TxtOut = ""  
    TxtAmount1 = ""  
    TxtRate1 = ""  
    TxtYears1 = ""  
    TxtOut1 = ""  
End Sub
```

11 - المشروع الحادي عشر

-----:-

(بعض التطبيقات الرياضيه)

سوف نتعلم من خلال هذا المشروع مايلي:-

- ❖ اوامر التكرار
- ❖ برمجة عدة مسائل رياضية
- ❖ حساب المفكوك Factorial
- ❖ أختبار هل ان العدد اولي ام لا
- ❖ المضاعف المشترك البسيط لعددين
- ❖ القاسم المشترك الاعظم لعددين

الهدف من المشروع

-----:-

عند تنفيذ المشروع تظهر الواجهه التاليه:



نتمكن من خلال هذه البرنامج حساب مايلي:

1. المفقوك Factorial
2. اختبار هل ان العدد اولي ام لا Primary or not
3. حساب المضاعف المشترك البسيط لعددین Simple Multiple
4. حساب القاسم المشترك الاعظم لعددین Max Divider

مراحل التنفيذ

-----:-

- ❖ نفتح مشروعاً قياسياً جديداً.
  - ❖ ننشئ عناصر التحكم الظاهره في النموذج أعلاه.
  - ❖ برمجة المفقوك Factorial
- المفقوك هو حاصل ضرب الاعداد من 1 الى n :  $1 * 2 * 3 * \dots * n$
- لحساب المفقوك لعدد n نحتاج الى حلقة تكرارية For من 1 الى n لتراكم ضرب الاعداد من 1 الى n .
- نكتب الاجراء التالي الخاص بحساب المفقوك:

```
Private Sub Command1_Click()  
Dim n As Integer  
Dim nf As Double  
Dim i As Integer  
n = Val(Text1.Text)  
nf = 1  
For i = 1 To n  
    nf = nf * i  
Next i  
Text2.Text = nf  
End Sub
```

## ❖ برمجة المفتاح Primary No

العدد الاولي هو العدد الذي لا يقبل القسمة على اي عدد (ماعدا على الواحد او على العدد نفسه).

لاختبار ان العدد  $n$  هو عدد اولي نقسمة على كل الاعداد من 2 الى  $n - 1$

❖ اذا كان هناك باقي القسمة على جميع الاعداد من 2 الى  $n - 1$  عند ذلك يكون العدد اولي.

❖ اما اذا قبل  $n$  القسمة على احد الاعداد من 2 الى  $n - 1$  يكون العدد  $n$  غير اولي.

نكتب الاجراء التالي على المفتاح Primary No.

Private Sub Command2\_Click()

Dim n As Integer

n = Val(Text3.Text)

Dim i As Integer

Option1.Value = True

For i = 2 To n - 1

If n Mod i = 0 Then

Option1.Value = False

Exit For

End If

Next i

End Sub

## ❖ برمجة المفتاح Simple Multiple

المضاعف المشترك البسيط لعددين هو اصغر عدد يقبل القسمة على كلا العددين وبالتالي فهو

محصور بين اكبر عدد وحاصل ضرب العددين (حاصل ضرب العددين حتما يقبل القسمة عليهما)

مثلا المضاعف المشترك البسيط لعددين 4 و 6 هو 12

الخوارزمية الخاصة بهذا الاجراء موضحة بالشكل التالي:

❖ ليكن العددين  $n, m$

❖ نوجد اكبر عدد بينهما وليكن  $\max$

❖ نجري الحلقة التكرارية For i →  $\max$  to  $n*m$

❖ في كل تكرار نختبر فيما اذا كان متغير الحلقة  $i$  يقبل القسمة على  $n, m$

❖ اذا قبل  $i$  القسمة على  $n$  و  $m$  يكون المضاعف المشترك الاصغر وعندها ننهي

Exit For الحلقة التكرارية

❖ نكتب الاجراء التالي على مفتاح Simple Multiple

```
Private Sub Command3_Click()
Dim n As Integer
Dim m As Integer
Dim i As Integer
Dim max As Integer
n = Val(Text4.Text)
m = Val(Text5.Text)
If n > m Then max = n Else max = m
  For i = max To n * m
    If (i Mod n = 0) And (i Mod m = 0) Then
      Exit For
    End If
  Next i
  Text6.Text = i
End Sub
```

❖ برمجة مفتاح القاسم المشترك الاعظم Max Divider

القاسم المشترك الاعظم لعددتين هو اكبر عدد يقسم كلا العددين بدون باقي وبالتالي فهو محصور بين اصغر عدد بينهما وبين العدد 1 مثلا القاسم المشترك الاعظم بين 4 و 6 هو 2. لحل هذه المسألة نكتب الخوارزمية التاليه:

❖ ليكن العددان n و m

❖ نوجد اصغر عدد بينهما وليكن min

❖ ننفذ الحلقة التكرارية For من العدد min الى العدد 1

For i → min to 1 step - 1

❖ في كل تكرار نختبر فيما اذا كان المتغير i يقسم كلا العددين n و m

❖ اذا قسم العدد i العددين n و m يكون القاسم المشترك الاعظم وعندها ننهي الحلقة التكرارية

❖ نكتب الاجراء التالي على مفتاح Max Divider وكمالي:

```
Private Sub Command4_Click()  
Dim n As Integer  
Dim m As Integer  
Dim i As Integer  
Dim min As Integer  
n = Val(Text7.Text)  
m = Val(Text8.Text)  
If n < m Then min = n Else min = m  
    For i = min To 1 Step -1  
        If (n Mod i = 0) And (m Mod i = 0) Then  
            Exit For  
        End If  
    Next i  
    Text9.Text = i  
End Sub
```

❖ برمجة المفتاح Clear

لتنفيذ برمجة المفتاح Clear نكتب الاجراء التالي:

```
Private Sub Command5_Click()  
Text1.Text = ""  
Text2.Text = ""  
Text3.Text = ""  
Text4.Text = ""  
Text5.Text = ""  
Text6.Text = ""  
Text7.Text = ""  
Text8.Text = ""  
Text9.Text = ""  
End Sub
```

❖ برمجة مفتاح الخروج Exit

للخروج من البرنامج نكتب الاجراء التالي:

**Private Sub Command6\_Click()**

**End**

**End Sub**



12 - المشروع الثاني عشر

-----:-

(دوال الحوار مع المستخدم)

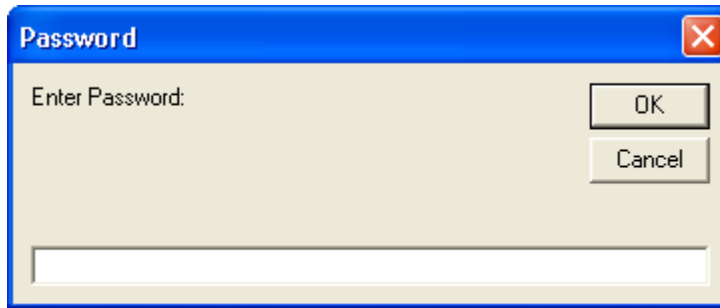
سوف نتعلم من خلال هذا المشروع مايلي:

- ❖ دوال الحوار مع المستخدم.
- ❖ صندوق الادخال InputBox
- ❖ صندوق الرسالة MsgBox
- ❖ الحلقات التكرارية.

الهدف من المشروع

-----:-

عند تنفيذ هذا المشروع سوف يظهر صندوق الحوار التالي الذي يطلب من المستخدم ادخال كلمة السر.



إذا ادخل المستخدم كلمة السر الصحيحة toto في مثالنا هذا ثم نقر على المفتاح Ok او المفتاح Enter سوف يظهر نموذج المشروع التالي:



أما إذا أخطأ المستخدم في إدخال كلمة السر الصحيحة فسوف يظهر له صندوق الرسالة التالية



- ❖ إذا نقر المستخدم على المفتاح **Cancel** سوف يتم إغلاق البرنامج، اما إذا نقر المستخدم على مفتاح **Retry** فسوف يظهر صندوق إدخال كلمة السر ليحرب إدخال كلمة السر مرة أخرى.
- ❖ يسمح البرنامج للمستخدم ان يخطيء مرتين في كلمة السر، أما إذا أخطأ في المرة الثالثة فسيتم إغلاق البرنامج.
- ❖ يحوي نموذج هذا المشروع التعليمي مفتاحا وحيدا لإغلاق البرنامج **Exit** عند النقر على هذا المفتاح يظهر صندوق الحوار التالي:



- ❖ إذا نقر المستخدم على المفتاح **No** لا يحدث أي شيء، أما إذا نقر على مفتاح **Yes** سيتم إغلاق البرنامج.

#### مراحل التنفيذ

-----:-

- ❖ برمجة الدالة **InputBox** : تستخدم هذه الدالة لإظهار صندوق الحوار الذي يطلب قيمة من المستخدم أي تنتظر الدالة من المستخدم ان يكتب نصا في صندوق النص او ينقر على احد المفتاحين **Ok** او **Cancel** حيث تعيد الدالة سلسلة حرفية **String** هي عبارته عن محتوى صندوق النص، تأخذ الدالة الشكل العام التالي:

**InputBox(Prompt[,title][,Default][,xpos][,ypos][,helpfile,context])**

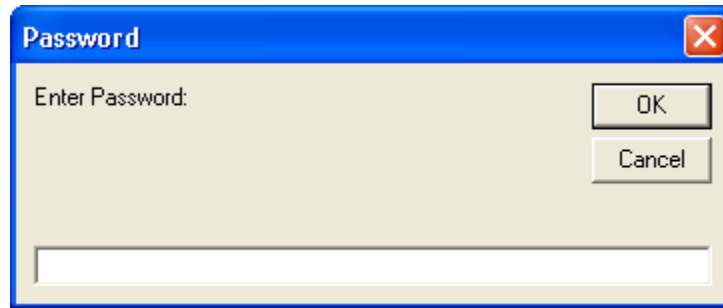
حيث ان

- ❖ Prompt تشير الى الرسالة الظاهره في الصندوق.
- ❖ Title عنوان الصندوق.
- ❖ Default القيمة الافتراضية التي تظهر في صندوق نص نافذة الحوار.
- ❖ Xpos, Ypos أحداثيات الصندوق.
- ❖ Helpfile, context تستخدم مع ملفات المساعدة.

الايغاز التالي يمثل كيفية استخدام الدالة `InputBox` ضمن البرنامج لظهار صندوق الحوار أعلاه:

`P = InputBox("Enter Password: ", "Password")`

عند النقر على مفتاح `Ok` او الضغط على مفتاح `Enter` فإن القيمة المرجعة للدالة `InputBox` تكون النص المكتوب في صندوق النص، أما إذا نقر المستخدم على مفتاح `Cancel` فإن القيمة المرجعه هي لاشيء مهما كان النص الموجود ضمن صندوق النص، وكما موضح في الشكل التالي:



- ❖ برمجة الدالة `MsgBox` تستخدم لظهار رسالة حوار ضمن صندوق خاص بها ، تنتظر هذه الدالة من المستخدم ان ينقر على احد المفاتيح التي يحتويها الصندوق، تعيد هذه الدالة قيمة رقمية `Integer` تمثل المفتاح الذي تم النقر عليه.
- تأخذ هذه الدالة الشكل العام التالي:

`MsgBox(Prompt[,buttons][,title][,helpfile,context])`

حيث ان :

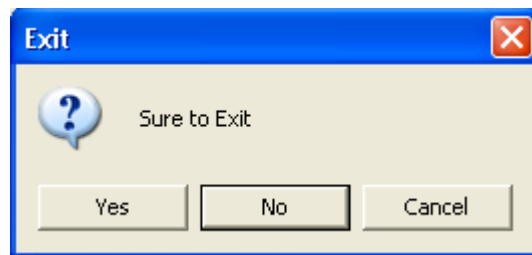
- ❖ Prompt تمثل الرسالة الظاهره في الصندوق.
- ❖ Title عنوان الصندوق.
- ❖ Helpfile, context تستخدم مع ملفات المساعدة.

- ❖ **Buttons** تعبير رقمي: يمثل مجموع ثوابت رقمية تحدد ايقونة ومفاتيح صندوق الحوار والمفتاح الافتراضي في الصندوق.  
مثلا : لظهور ايقونة الاستفهام مع المفاتيح (Yes, No, Cancel) بحيث يكون المفتاح الثاني هو المفتاح الافتراضي، يكون هذا التعبير كمايلي:

**vbQuestion + vbYesNoCancel + vbDefaultButton2**

وكما موضح في الشكل التالي:

**R = MsgBox("Sure to Exit", vbQuestion + vbYesNoCancel + vbDefaultButton2, "Exit")**



- ❖ تحدد الثوابت التالية الايقونة الظاهرة في صندوق الحوار عند استخدامها في البرمجة وكمايلي:

<b>VBQUESTION</b>	أشارة الاستفهام
<b>vbExclamation</b>	أشارة التعجب
<b>vbCritical</b>	اشارة الخطأ
<b>vbInformation</b>	أشارة المعلومات

- ❖ تحدد الثوابت التالية مفاتيح صندوق الرسائل **MsgBox**

<b>VBOKONLY</b>	<b>OK</b>
<b>vbOKCancel</b>	<b>OK, Cancel</b>
<b>vbAbortRetryIgnore</b>	<b>Abort, Retry, Ignore</b>
<b>vbYesNoCancel</b>	<b>Yes, No, Cancel</b>
<b>vbYesNo</b>	<b>Yes, No</b>
<b>vbRetryCancel</b>	<b>Retry, Cancel</b>

❖ تحدد الثوابت التالية المفتاح الافتراضي

VBDEFAULTBUTTON1	المفتاح الاول هو المفتاح الافتراضي
VbDefaultButton2	المفتاح الثاني هو المفتاح الافتراضي
VbDefaultButton3	المفتاح الثالث هو المفتاح الافتراضي

- ❖ نضغط على النموذج فيظهر حدث تحميل النموذج Form\_Load ، سوف نكتب اجراء طلب كلمة السر للدخول الى المشروع في هذا الحدث.
- ❖ برمجة الحدث Form\_Load يحصل عند تحميل النموذج في الذاكرة، نحتاج في هذا الاجراء الى حلقة تكرارية لتكرار السؤال عن كلمة السر ثلاثة مرات على الاكثر.
- ❖ في حالة معرفة المستخدم لكلمة السر يتم انهاء الحلقة التكرارية.
- ❖ اذا ضغط المستخدم على Cancel في نافذة طلب كلمة السر او في نافذة اعادة المحاولة يتم اغلاق البرنامج.
- ❖ اذا وصل عدد المحاولات الى ثلاث دون معرفة كلمة السر يغلق البرنامج.
- ❖ نكتب الاجراء التالي على حدث تحميل النموذج:

```

Private Sub Form_Load()
Dim P As String
Dim R As Integer 'Retry or Cancel
Dim i As Integer
Do While True
i = i + 1
P = InputBox("Enter Password: ", "Password")
If P = "toto" Then Exit Do
If P = "" Then End
If i = 3 Then End
R = MsgBox("Wrong Password", vbQuestion + vbRetryCancel +
vbDefaultButton1, "Error")
'R = MsgBox("Wrong Password", vbExclamation + vbRetryCancel +
vbDefaultButton1, "Error")
'R = MsgBox("Wrong Password", vbCritical + vbRetryCancel +
vbDefaultButton1, "Error")
'R = MsgBox("Wrong Password", vbInformation + vbRetryCancel +
vbDefaultButton1, "Error")
If R = vbCancel Then End
Loop

End Sub

```

❖ انشاء مفتاح Exit للخروج من البرنامج

نريد قبل إغلاق البرنامج أظهار صندوق الحوار للتأكد من أن المستخدم يريد إغلاق البرنامج، يحوي هذا الصندوق على مفتاح Yes ومفتاح No وهو المفتاح الافتراضي. لن يتم إغلاق البرنامج الا اذا نقر المستخدم على مفتاح Yes.

❖ نكتب الاجراء التالي على مفتاح Exit

```
Private Sub Command1_Click()
Dim R As Integer
R = MsgBox("Sure to Exit", vbQuestion + vbYesNo + vbDefaultButton2,
"Exit")
'R = MsgBox("Sure to Exit", vbQuestion + vbYesNoCancel +
vbDefaultButton2, "Exit")
'R = MsgBox("Sure to Exit", vbQuestion + vbOKCancel + vbDefaultButton2,
"Exit")
If R = vbYes Then End
'If R = vbYes Then End
'If R = vbOk Then End
End Sub
```

13 - المشروع الثالث عشر

-----:-

مصفوفات عناصر التحكم

سوف نتعلم من خلال هذا المشروع مايلي:

- ❖ انشاء مصفوفة من عناصر التحكم.
- ❖ القيام بالعمليات التالية على مجموعة من الارقام.
- ❖ إيجاد المعدل (الوسط الحسابي).
- ❖ إيجاد القيمة العظمى.
- ❖ إيجاد القيمة الصغرى.
- ❖ الترتيب التصاعدي ( الفرز ) لمجموعة من الارقام.

الهدف من المشروع

-----:-

عند تنفيذ هذا المشروع تظهر الواجهة التالية:



- ❖ يمكن للمستخدم ان يدخل ارقاما في صناديق النص، ثم يطلب ايجاد الوسط الحسابي ( المعدل Average ) والقيمة العظمى (Max) والقيمة الصغرى (MIN) لهذه الارقام. عند النقر على المفتاح تلوين (Color) سوف يتلون العدد الزوجي بلون احمر والعدد الفردي بلون ازرق، عند النقر على مفتاح ترتيب (Ordered) سيتم ترتيب الارقام تصاعديا.

### مراحل التنفيذ

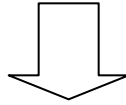
-----:-

- ❖ نفتح مشروعا قياسيا جديدا.
- ❖ ننشئ مصفوفة من (10) مربعات نص (Text Box):  
M(0), M(1), M(2), ..., M(9)
- ❖ تصميم المصفوفة: وهي عبارة عن مجموعة من عناصر التحكم من نفس النوع، لها نفس الاسم، حيث يكون لكل منهما فهرس (Index) متسلسل يبدأ من الصفر. وبالتالي يمكن لنا استخدام الحلقة التكرارية (For) للتعامل مع جميع عناصر المصفوفة.  
لانشاء هذه المصفوفة، نقوم بانشاء اول عنصر ونسمية ونضبط خصائصه ثم نقوم بعملية نسخة (Copy) ثم نقوم بعملية لصقه (Paste). مثلا لانشاء مصفوفة من صناديق النص نقوم بمايلي:  
- ننشئ صندوق نص ، نسمية (M) ونضبط خصائصه (مع ملاحظة ان الخاصية Index له فارغه  
- نحدد صندوق النص (M) ومن ثم ننسخه بأختيار القائمة (Edit/Copy).  
- نقوم بعملية لصقه (Paste) من خلال القائمة (Edit/Paste)، سوف يظهر صندوق حوار يحتوي على مفتاح "Yes" or "No" والرسالة التالية:  
You already have a control named 'M'  
Do you want to create a control array?  
- اذا اخترنا "No" فسوف ينشئ صندوق نص لاعلاقة له ب (M).  
- أما اذا اخترنا "Yes" تصبح الخاصية (Index) لصندوق النص الاول مساويا الى صفرا.  
- ننشئ صندوق نص جديد اسمة (M) عن طريق الاستنساخ واللصق (Copy/Paste) والخاصية (Index) له مساويا الى العدد ( 1 )، وهكذا نكرر العملية حتى صندوق النص M(9) اي بعدد عناصر المصفوفه.



M(0) TEXT BOX	
Name	M
Index	0
Text	

M(0) TEXT BOX	
Name	M
Index	1
Text	



M(0) TEXT BOX	
Name	M
Index	9
Text	

- ❖ ننشيء مفتاح لايجاد المعدل (Average) وصندوق نص (TxtAvg).
- ❖ ننشيء مفتاح لايجاد العدد الاكبر (Max)، وصندوق نص (TxtMax).
- ❖ ننشيء مفتاح لايجاد العدد الاصغر (Min) وصندوق نص (TtMin).
- ❖ ننشيء مفتاح للتلوين (Color).
- ❖ ننشيء مفتاح لاجراء عملية الترتيب (Ordered).

#### ❖ برمجة المفتاح (Average)

لحساب الوسط الحسابي (المعدل Average) لمجموعة من الاعداد نقسم مجموع تلك الارقام على عددها. عند ذلك نحتاج الى متغير (Avg) لتراكم او جمع الارقام المكتوبة في عناصر المصفوفة، وللوصول الى عناصر تلك امصفوفة نحتاج الى حلقة تكرارية For من صفر الى 9 ، لان عناصر المصفوفة العشرة مفهرسة من صفر الى 9 .

## ❖ برمجة الخاصية Count

تعيد هذه الخاصية عدد عناصر المصفوفة.

حيث ان عدد عناصر المصفوفة 10 بالاعتماد على  $M.Count = 10$ ، وبالتالي تكون حلقة For من صفر الى  $M.Count - 1$ ، أذ أن فهرسة عناصر المصفوفة تبدأ من الصفر.  
البرنامج الخاص بهذا الاجراء موضح بالشكل التالي:

```
Private Sub Command1_Click()
Dim i As Integer
Dim Avg As Single
Avg = 0
For i = 0 To M.Count - 1
    Avg = Avg + Val(M(i).Text)
Next i
    TxtAvg.Text = Avg / M.Count
End Sub
```

## ❖ برمجة المفتاح Max

لايجاد القيمة العظمى لمجموعة من الارقام نستخدم الطريقة التالية:

- نفرض ان اول عدد هو القيمة العظمى ونضعه في متغير Max.
  - نعمل حلقة تكرارية على بقية الارقام وكلما وجدنا رقما اكبر من Max نضعه في Max.
- نكتب الاجراء التالي على مفتاح Max وكمايلي:

```
Private Sub Command2_Click()
Dim i As Integer
Dim Max As Single
Max = Val(M(0))
For i = 1 To M.Count - 1
If Val(M(i)) > Max Then
    Max = Val(M(i))
End If
Next i
    TxtMax.Text = Max
End Sub
```

## ❖ برمجة المفتاح Min

بنفس المبدأ المعتمد في حساب القيمة العظمى، نكتب الاجراء التالي على المفتاح Min وكمالي:

```
Private Sub Command3_Click()
Dim i As Integer
Dim Min As Single
Min = Val(M(0))
For i = 1 To M.Count - 1
If Val(M(i)) < Min Then
    Min = Val(M(i))
End If
Next i
TxtMin.Text = Min
End Sub
```

## ❖ برمجة المفتاح تلوين Color

لاختبار ان عددا ما زوجي ام فردي نختبر باقي قسمته على 2 باستخدام Mod فاذا كان باقي القسمة صفرا يكون العدد زوجيا والا كان فرديا.

نكتب الاجراء التالي على المفتاح تلوين Color

```
Private Sub Command5_Click()
Dim i As Integer
Dim j As Integer
Dim x As Single
For i = 0 To M.Count - 1
    For j = i + 1 To M.Count - 1
        If Val(M(j)) < Val(M(i)) Then
            x = Val(M(j))
            M(j) = M(i)
            M(i) = x
        End If
    Next j
Next i
End Sub
```

## ❖ برمجة المفتاح ترتيب تصاعدي Ordered

يوجد العديد من الطرق لترتيب مجموعة من الارقام في مصفوفة من احد هذه الطرق:

- نبدأ بأول عنصر في المصفوفة.
- نعمل حلقة تكرارية على باقي عناصر المصفوفة (من العنصر الثاني الى اخر عنصر) وكلما وجدنا عنصرا اصغر من اول عنصر نقوم بتبديله مع اول عنصر، وفي نهاية هذه الدورة سيكون اول عنصر هو العدد الاصغر في المصفوفة.
- ننتقل الى العنصر الثاني في المصفوفة.
- نعمل حلقة تكرارية على باقي عناصر المصفوفة (من العنصر الثالث الى اخر عنصر) وكلما وجدنا عنصرا اصغر من العنصر الثاني بادلنا بينه وبين ثاني عنصر. في نهاية هذه الحلقة سيكون اول عنصر هو العدد الاصغر في المصفوفة وثاني عنصر هو العدد الذي يليه وهكذا حتى الوصول الى اخر عنصر في المصفوفة.
- نحتاج في هذا الاجراء الى حلقتي For وهما

For i=0 to M.Count-1

For j=i+1 to M.Count

- الحلقة الاولى I للدوران على كل عناصر المصفوفة، والحلقة الثانية j للدوران من اجل كل I على عناصر المصفوفة من i+1 الى اخر عنصر في المصفوفة.
- يصبح الاجراء الخاص بالمفتاح ترتيب تصاعدي Ordered كمايلي:

```
Private Sub Command5_Click()
Dim i As Integer
Dim j As Integer
Dim x As Single
For i = 0 To M.Count - 1
For j = i + 1 To M.Count - 1
If Val(M(j)) < Val(M(i)) Then
x = Val(M(j))
M(j) = M(i)
M(i) = x
End If
Next j
Next i
End Sub
```

عملية التنفيذ لكافة الاجراءات موضحة في الشكل التالي:



14 - المشروع الرابع عشر

-----:-

## التحريك

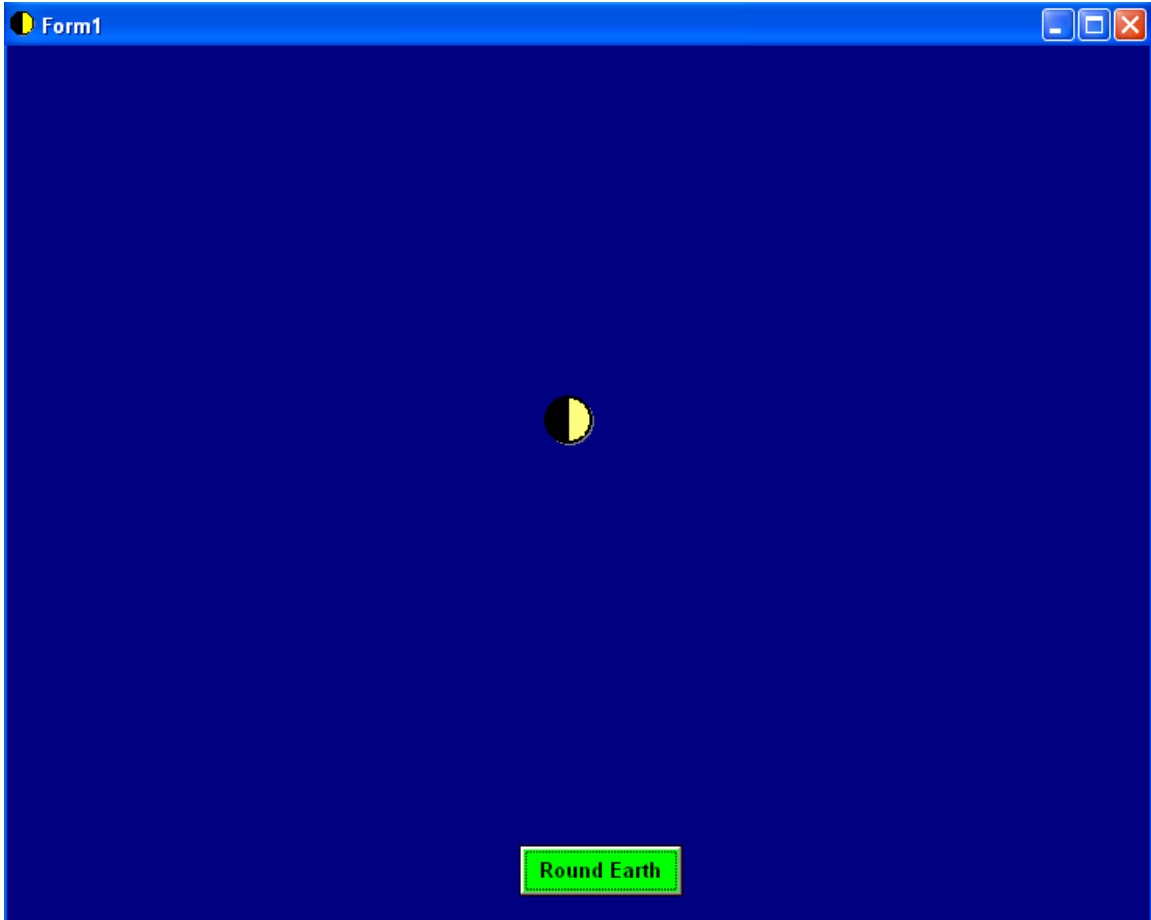
سنتعلم من خلال هذا المشروع :-

- ❖ التعامل مع المؤقت Timer.
- ❖ التعامل مع مصفوفات عناصر التحكم.
- ❖ المتغيرات الساكنة Static .
- ❖ اشرطة الانزلاق Scroll Bars.
- ❖ توليد رقم عشوائي Rnd.

## الهدف من المشروع

-----:-

- ❖ عند تنفيذ هذا المشروع، يظهر النموذج الاول من المشروع والذي يحتوي على صورة قمر يدور حول نفسه (كذلك نلاحظ دوران ايكونة النموذج). كما يتغير لون خلفية النموذج كل نصف ثانية.



- ❖ عند النقر على مفتاح حول الارض Round Earth في النموذج الاول، يظهر النموذج الثاني والذي يحوي قمرا يدور حول الارض، يمكن التلاعب بسرعة دوران القمر حول الارض عن طريق شريط الانزلاق العمودي، كذلك يمكن عكس اتجاه دوران القمر بالنقر على مفتاح Right/Left.

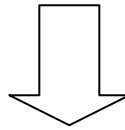
### مراحل التنفيذ

-----:-

- ❖ نفتح مشروعنا قياسيا جديدا.
- ❖ ننشئ نموذجين.
- ❖ ننشئ في النموذج الاول مصفوفة ونسميها (Moon) من ثمانية صور Image وكمايلي:

MOON (0) IMAGE	
Name	M
Index	0
Picture	Icon

MOON (1) IMAGE	
Name	M
Index	1
Picture	Icon



MOON (7) IMAGE	
Name	M
Index	7
Picture	Icon

- ❖ نضع في عناصر المصفوفة Moon صور الاقمار باوضاعها المختلفة والتي نجدها في المجلد التالي:  
C:\Program Files\Microsoft Visual Studio\Common\Graphics\Icons\Elements
- ❖ ننشئ صورة (التي سوف تدور فيها الاقمار) ونسميها ImgMoon.

- ❖ ننشئ مؤقتا Timer ونضبط مجاله الزمني Interval على 500 ميلي ثانية مثلا.
- ❖ المؤقت Timer يستخدم لتنفيذ اجراء كل فترة زمنية محددة.
- ❖ تستخدم الخاصية Interval لتحديد الفترة الزمنية للمؤقت، أي الفترة الزمنية الفاصلة بين تنفيذ اجراء المؤقت. يقاس عادة المجال بالميلي ثانية ( اي اذا وضعنا القيمة 1000 مثلا، يكون الزمن للمؤقت عبارة عن ثانية واحدة، اذا كانت قيمة المجال صفرا لن يعمل المؤقت، تتراوح قيمة المجال الزمني بين الصفر و القيمة 65535 ميلي ثانية اي اكثر من الدقيقة بقليل.
- ❖ مثال: ضع مؤقتا على النموذج، واضبط مجاله الزمني على 500 ميلي ثانية، واكتب الامر التالي على اجراء المؤقت:

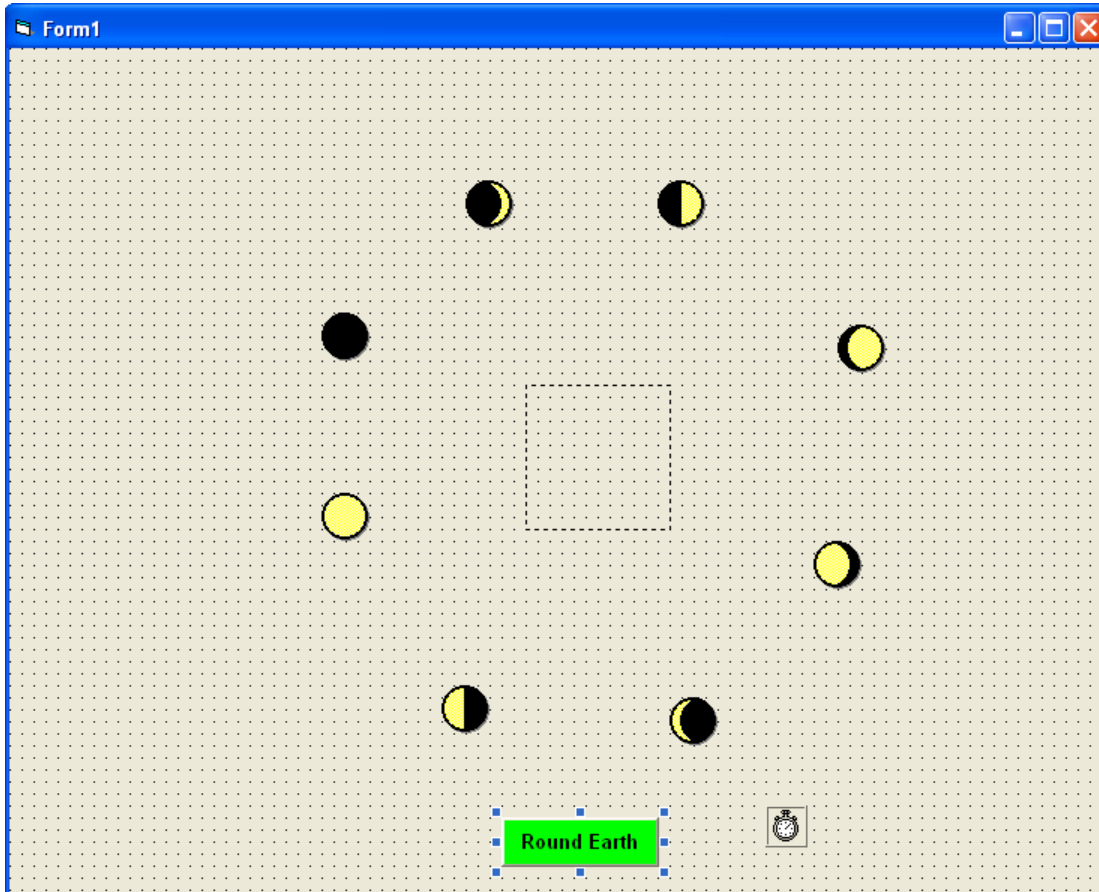
Me.BackColor = QBColor(15\*Rnd)

- ❖ عند التنفيذ سيتغير لون النموذج بشكل عشوائي كل نصف ثانية.
- ❖ برمجة الدالة Rnd تعيد عند كل استدعاء رقما عشوائيا محصورا بين الصفر والواحد ، في الامر السابق نستخدم لتوليد لون الخلفية الدالة QBColor التي تأخذ شكل متغير أذخار رقما محصورا بين قيمة الصفر والقيمة 15، القيمة الناتجة من حاص الضرب (15 \* Rnd) تعيد عند كل استدعاء رقما عشوائيا محصورا بين قيمة الصفر والقيمة 15.
- ❖ برمجة المؤقت Timer في النموذج الاول : من خلال استخدام المؤقت نحصل على صور متحركة مختلفة ومتسلسلة بفترة زمنية قصيرة نسبيا، تبدو الصورة وكأنها تتحرك.
- في مثالنا هذا نقوم بعرض صور الاقمار الثمانية في ImgMoon بشكل مستمر:
- نضع اول الصورة الاولى Moon(0) في ImgMoon.
- بعد نصف ثانية نضع الصورة الثانية Moon(1) في ImgMoon.
- بعد نصف ثانية ايضا نضع الصورة الثالثة Moon(2) في ImgMoon.
- وهكذا حتى عرض الصورة الاخيرة Moon(7) عندها نحتاج الى الرجوع الى الصورة الاولى Moon(0) ، نحتاج الى ذلك متغيير ثابت Static أسمة I كي يحافظ على قيمته من استدعاء لآخر ، سنزيد هذا المتغيير بمقدار واحد في كل مرة وعندما يصل I الى ثمانية نعيدة الى الصفر.
- يكون اجراء مؤقت النموذج الاول كمايلي:

```
Private Sub Timer1_Timer()
Me.BackColor = QBColor(15 * Rnd)
Static i As Integer
ImgMoon.Picture = M(i).Picture
Me.Icon = M(i).Picture
i = i + 1
If i = 8 Then i = 0
End Sub
```



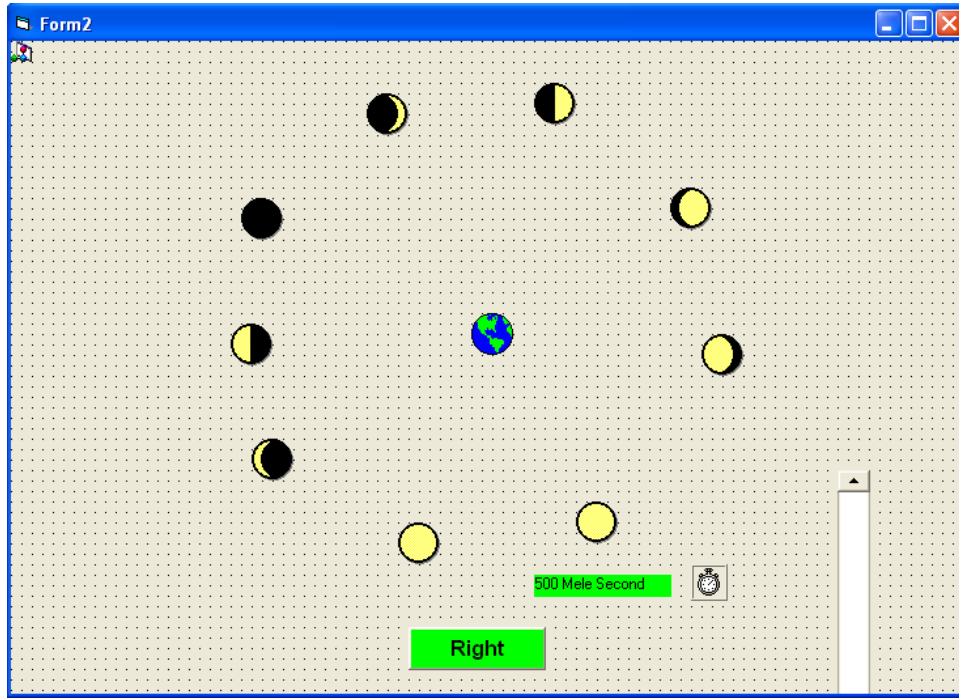
الخطوات في الاعلى موضحة في النموذج التالي:



❖ ننشئ مفتاح حول الارض Round Earth على النموذج الاول ونكتب في أجهزته امر اضهار النموذج الثاني:

```
Private Sub Command1_Click()
Me.Hide
Form2.Show
End Sub
```

❖ ننشئ صورة الارض على النموذج الثاني ImgEarth ونضع حولها مصفوفة من ثمانية صور للقمر Moon بشكل دائري، نجعل جميع عناصر هذه المصفوفة مخفية عن طريق الخاصية Visible = False، وكما موضح في الشكل التالي:



❖ ننشئ مفتاح عنوانه الابتدائي Right ، كلما نقرنا على هذا المفتاح يتغير عنوانه Right/Left الاجراء التالي يوضح ذلك:

```
Private Sub Command1_Click()
If Command1.Caption = "Right" Then
Command1.Caption = "Left"
Else
Command1.Caption = "Right"
End If
End Sub
```

- ❖ ننشئ مؤقتا على النموذج الثاني، ونضبط مجاله الزمني على 500 ميلي ثانية.
- ❖ ننشئ تسمية Label ونضع في خاصية العنوان له قيمة مجال المؤقت 500 Mele Second.
- ❖ ننشئ شريط الانزلاق العمودي Vertical Scroll Bar ونضبط خصائصه كالآتي:

VSCROLL1 VSCROLLBAR	
Name	VScroll1
Max	10000
Min	1
Value	500

- ❖ يستخدم شريط الانزلاق Scroll Bar لتوليد قيمة Value محصورة بين قيمة دنيا Min وقيمة عظمى Max.
- ❖ في مثالنا هذا نستخدم شريط الانزلاق لتغيير مجال مؤقت النموذج الثاني من 1 ميلي ثانية الى 10000 ميلي ثانية أي (10) ثواني.
- ❖ نكتب الاجراء التالي على حدث تغيير Change قيمة شريط الانزلاق:

```
Private Sub VScroll1_Change()
Label1.Caption = VScroll1.Value
Timer1Sub.Interval = VScroll1.Value
End
```

- ❖ برمجة مؤقت النموذج الثاني: لجعل الاقمار تبدو وكأنها تدور حول الارض:
- ❖ في البداية تكون الاقمار مخفية.
- ❖ بعد نصف ثانية يظهر القمر الاول Moon(0).
- ❖ بعد نصف ثانية يختفي القمر الظاهر Moon(0) ويظهر القمر Moon(1).
- ❖ وهكذا حتى الوصول الى القمر Moon(7) بعدها نعود الى القمر الاول Moon(0).
- ❖ نحتاج في اجراء المؤقت الى متغير ساكن I كي يحافظ على قيمته من استدعاء لاستدعاء اخر، سيزيد هذا المتغير بقيمة واحد في كل مرة اذا كان عنوان المفتاح هو Right وعندما نصل الى ثمانية نعيدة للصفر، اما اذا كان عنوان المفتاح Left ينقص من المتغير بمقدار واحد وعندما يصل هذا المتغير الى - 1 نعيده الى سبعة، في كل مرة نخفي جميع الاقمار ونظهر القمر الذي رقمه i . يكون اجراء المؤقت كمايلي:

```
Private Sub Timer1_Timer()  
Static i As Integer  
Dim j As Integer  
For j = 0 To 7  
    M1(j).Visible = False  
Next j  
    M1(i).Visible = True  
If Command1.Caption = "Right" Then  
i = i + 1  
Else  
i = i - 1  
End If  
If i = 8 Then i = 0  
If i = -1 Then i = 7  
End Sub
```

15 - المشروع الخامس عشر

بناء البرامج الفرعية

- ❖ الاجراءات Sub.
- ❖ الدوال Function.

الهدف من المشروع

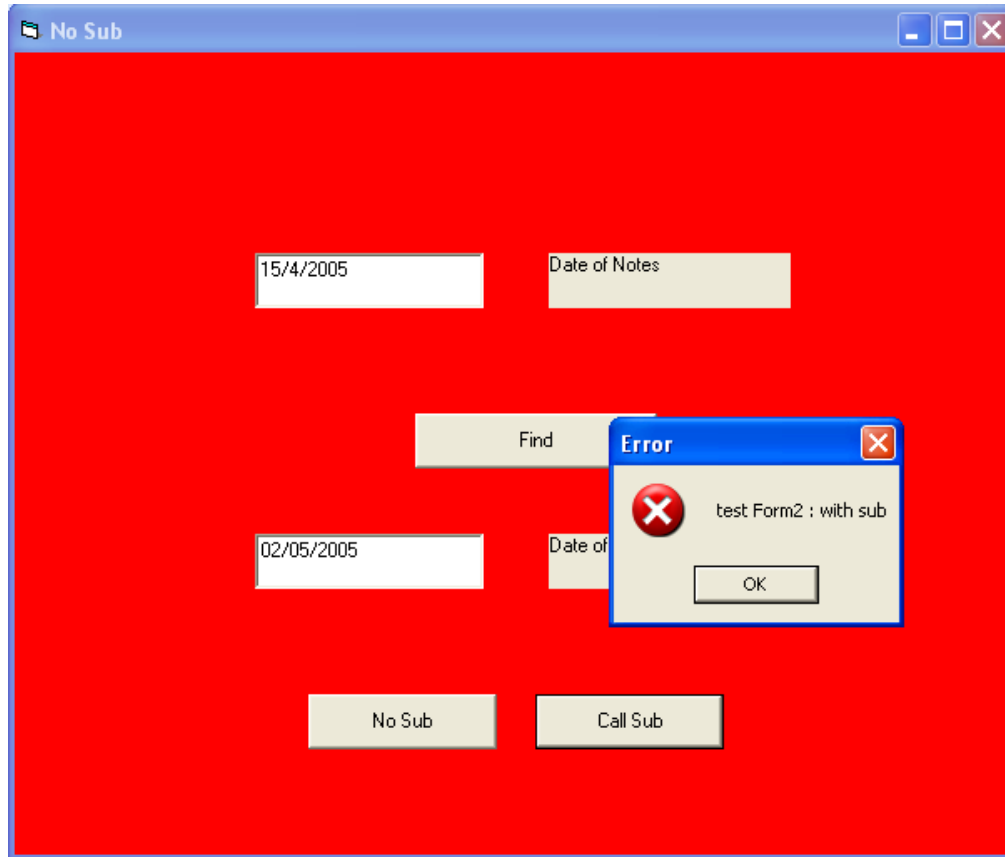
- ❖ عند تنفيذ هذا المشروع سوف يظهر النموذج الاول من المشروع، وعند النقر على مفتاح Form2 يظهر النموذج الثاني، كما موضح في الاشكال التالية:

The screenshot shows a Windows-style window titled 'Form1' with a blue title bar and standard minimize, maximize, and close buttons. The main area has an orange background. It contains the following elements:

- A text box containing '12/2/2004' and a label 'Date' to its right.
- A button labeled 'Find' in the center.
- A text box containing 'Thursday' and a label 'Day' to its right.
- Three buttons at the bottom: 'Form2', 'No Sub', and 'Call Sub'.

❖ عند النقر على مفتاح **No Sub** في احد النموذجين، يتلون النموذج باللون الاحمر ويظهر صندوق رسالة خطأ كما في الشكل التالي:

- ❖ عند النقر على مفتاح **Call Sub** في احد النموذجين، يتلون النموذج أيضا باللون الاحمر ويظهر صندوق رسالة خطأ كما في الشكل التالي:



- ❖ بعد كتابة تاريخ في صندوق النص الاول في النموذج الاول، ومن ثم النقر على المفتاح أوجد Find ، يظهر في صندوق النص الثاني أسم اليوم الموافق للتاريخ.
- ❖ بعد كتابة تاريخ في صندوق النص الاول في النموذج الثاني، ومن ثم النقر على مفتاح أوجد Find يظهر في صندوق النص الثاني تاريخ اول يوم ليس عطلة (سبت او أحد) من الشهر التالي:

### مراحل التنفيذ

-----:-

- ❖ نفتح مشروعاً قياسياً جديداً.
- ❖ ننشئ نموذجين.
- ❖ ننشئ على النموذجين عناصر التحكم الظاهره في الاشكال السابقة.
- ❖ برمجة المفتاح No Sub على النموذج الاول:

الفكرة الرئيسية من خلال تنفيذ هذا الاجراء هو اننا نريد عند كل ظهور ل صندوق رسالة MsgBox ان نلون النموذج باللون الاحمر لشد انتباه المستخدم، وبعد اغلاق الصندوق يعود النموذج الى اللون الاول.

### ❖ نكتب الاجراء التالي على مفتاح No Sub في النموذج الاول:

```
Private Sub Command3_Click()
Dim c As Long
c = Me.BackColor
Me.BackColor = vbRed
MsgBox "Test Form1 : WithOut Sub", vbCritical, "Error"
Me.BackColor = c
End Sub
```

### ❖ برمجة المفتاح No Sub على النموذج الثاني:

نكتب الاجراء التالي الذي يكون شبيها بالاجراء السابق:

```
Private Sub Command2_Click()
Dim c As Long
c = Me.BackColor
Me.BackColor = vbRed
MsgBox "Test Form2 : WithOut Sub", vbCritical, "Error"
Me.BackColor = c
End Sub
```

### ❖ كتابة الاجراء MyError

نلاحظ أننا في الاجرائين السابقين استخدمنا نفس التعليمات او الاوامر، ما يختلف في التعليمات هو أسم الاجراء ورسالة الخطأ.

سوف نقوم عوضا عن كتابة نفس هذه التعليمات في اكثر من مكان، بكتابة اجراء عام ندعوه MyError ومن ثم نستعدى هذا الاجراء حيثما نريد في المشروع، سنمرر لهذا الاجراء عند كل استدعاء النموذج المنفذ عليا ورسالة الخطأ التي نريد أظهارها في صندوق الرسالة.

### ❖ برمجة البرامج الفرعية

تستخدم البرامج الفرعية عادة لاتمام المهام التي نقوم بها بشكل متكرر، مثلا يمكن أن ننشئ دالة ونستدعيها في عدة تعابير في أماكن متعددة.



يمكن أن نستخدم البرامج الفرعية أيضا في حال أن عدة إجراءات (أحداث Events) تقوم بنفس العمليات، عندها يمكن ان نضع هذه العمليات في برنامج فرعي واحد ونجعل جميع إجراءات الاحداث تستدعي هذا البرنامج الفرعي، مما سيؤدي الى الغاء التعليمات المتكررة كما سيجعل التعديل أسهل فيما بعد، أذ أن التعديل سيكون في البرنامج الفرعي فقط وليس في كل إجراءات الاحداث، تكون البرامج الفرعية اما إجراءات Sub او دوال Function.

### ❖ برمجة الإجراءات Sub

الشكل العام للإجراء هو

[Public|Privates] Sub name [(arglist)]

[statements]

[Exit Sub]

[statements]

End Sub

حيث ان :

❖ **Public** تحدد ان الاجراء هو عام، أي قابل للاستدعاء من جميع أجزاء المشروع(وهي الحالة الافتراضية أن لم نحدد نوع الاجراء عاما ام خاصا).

❖ **Private** تحدد أن الاجراء خاص، لا يستدعى الا في نفس مكان تعريفه(صفحة البرمجة لنموذج أو وحدة نمطية).

❖ **Name** اسم الاجراء.

❖ **Statements** وهي الاوامر او التعليمات الخاصة بالاجراء التي ستنفذ عند استدعاء الاجراء.

❖ **Exit Sub** وهي أمر الخروج الفوري من الاجراء (كسر الاجراء).

❖ **Arglist** هي أسماء متغيرات الادخال للاجراء، ويفصل بينهما بفاصلة (،) في حالة وجود أكثر من متغير ادخال، كل متغير ادخال يعرف كتعريف أي متغير وسيعامل كمتغير في تعليمات الاجراء، الشكل العام لتعريف أي متغير ادخال هي:

[By Val]|By Ref] varname[()] [As type]

حيث أن:

❖ **By Val** تحدد أن تمرير المتغير هي التمرير بالقيمة، أي لا تتغير قيمة المتغير الممرر للاجراء.

❖ **By Ref** تحدد أن طريقة تمرير المتغير هي التمرير بالعنوان، أي يمكن أن تتغير قيمة المتغير الممرر للاجراء ( وهي الطريقة الافتراضية أن لم نحدد طريقة التمرير).

❖ **Varname** أسم المتغير ، حيث أن الاقواس بعد اسم المتغير ( ) تدل على أنه مصفوفة (Array) . عند استدعاء الاجراء نعطي قيما معينة لمتغيرات الدخل، يتم تنفيذ تعليمات الاجراء من أجلها(أي سيتم أستبدال كل متغير أدخل في تعليمات الاجراء بالقيمة المررخ).

❖ **Type** يمكن أن يكون النوع أي نوع بيانات اساسي:

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

أو **Variant** او نوع كائن Object Type ، اذا لم نحدد نوع البيانات الخاصة بمتغير الادخال، فإنه سيأخذ النوع variant أي أنه أي نوع من البيانات .

❖ يتم استدعاء الاجراء بأستخدام الامر

[Call] nme [argumentlist]

❖ **Name** أسم الاجراء.

❖ **Argumentlist** قائمة قيم متغيرات الادخال للاجراء.

❖ **Call** اختيارية، اذا وضعت يجب أن نضع قائمة متغيرات الدخل ضمن قوسين ( ) و الا فلا يجب وضع الاقواس.

❖ **برمجة**: يكتب البرنامج الفرعي العام في وحدة مطية Module ويصبح قابلا للاستدعاء من مختلف نماذج المشروع، يمكن في الوحدة النمطية كتابة عدد من الاجراءات والدوال.

❖ **تصميم**: لاضافة وحدة نمطية الى المشروع نختار القائمة Project/Add Module فتظهر صفحة تحرير، يمكن كتابة الاجاء فيها. تحفظ الوحدة النمطية في ملف مستقل ينتهي بالامتداد (.bas) ، ننشئ وحدة نمطية جديده Module1 ونكتب الاجراء MyError التالي فيها. يأخذ هذا الاجراء متغيرات الادخال التالية:

**F as Form** نمرر في هذا المتغير النموذج المراد تلوينة.

**S As String** نمرر في هذا المتغير رسالة الخطأ.

Sub MyError(f As Form, s As String)

Dim c As Long

c = f.BackColor

f.BackColor = vbRed

MsgBox s, vbCritical, "Error"

f.BackColor = c

End Sub

## ❖ برمجة المفتاح Call Sub على النموذج الاول:

نستدعي في إجراء هذا المفتاح الاجراء العام MyError ، ونمرر له النموذج الاول Form1 ورسالة الخطأ

```
Private Sub Command2_Click()
    MyError Form1, "Test Form1 : with Sub"
End Sub
```

## ❖ برمجة المفتاح "Call Sub" على النموذج الثاني

نستدعي في إجراء هذا المفتاح الاجراء العام MyError ونمرر له النموذج الثاني Form2 ورسالة الخطأ.

```
Private Sub Command2_Click()
    Call MyError(Form2, "Test Form2: with Sub")
End Sub
```

## ❖ كتابة الدالة DateDay في وحدة نمطية Module2

## ❖ برمجة الدوال Function

الشكل العام لتعريف الدالة:

```
[Public| Private] Function name [(arglist)]
[statements]
[name = expression]
[Exit Function]
[statements]
[name = expression]
End Function
```

تعمل متغيرات الادخال في الدالة تماما كما تعمل متغيرات الادخال في الاجراء. هناك ثلاثة فروق اساسية بين الاجراء والدالة:

- ❖ يجب أن نضع دائما أقواسا بعد أسم الدالة عند أستدعاها .
- ❖ للدوال متغيرات نوع بيانات يحدد نوع القيمة المرجعة.
- ❖ تنسب القيمة المرجعة في الدالة لاسم الدالة، وبالتالي يمكن أستخدام الدالة في التعبيرات.
- ❖ في مشروعنا هذا نريد كتابة دالة نمرر لها تاريخا ، فتعيد لنا أسم اليوم الموافق لهذا التاريخ، نكتب الدالة DateDay التالية في وحدة نمطية ثانية Module2.

```

Function DateDay(d As Date) As String
Select Case Weekday(d)
Case 1: DateDay = "Sunday"
Case 2: DateDay = "Monday"
Case 3: DateDay = "Tuesday"
Case 4: DateDay = "Wednesday"
Case 5: DateDay = "Thursday"
Case 6: DateDay = "Friday"
Case 7: DateDay = "Saturday"
End Select
End Function

```

- تأخذ الدالة DateDay متغير إدخال وحيد هو D من النوع date ، وتعيد سلسلة نصية String .
- ❖ برمجة الدالة WeekDay (Date) من دوال VB تأخذ تاريخا Date وتعيد رقم اليوم في الاسبوع الموافق لهذا التاريخ (1 الاحد، 2 الاثنين، ...).
  - ❖ استدعاء الدالة DateDay على المفتاح اوجد Find في النموذج الاول:
- المطلوب أن يكتب المستخدم تاريخا ما في صندوق النص Text1. بعد النقر على المفتاح Find، يظهر في صندوق النص Text2 أسم اليوم الموافق للتاريخ المدخل. نستدعي في إجراء المفتاح أوجد Find الدالة السابقة DateDay ، نمرر لها Text1.Text كقيمة متغير إدخال للدالة ونسند نتيجة الدالة الى Text2. نختبر قبل استدعاء الدالة النص المدخل في Text1 هو تاريخ مقبول بأستخدام الدالة IsDate والا فتظهر عبارة خطأ في صندوق النص Text2.
- ❖ نكتب على مفتاح أوجد Find الاجراء التالي:

```

Private Sub Command1_Click()
If IsDate(Text1.Text) = True Then
    Text2 = DateDay(Text1.Text)
Else
    Text2 = "Error"
End If
End Sub

```

- ❖ برمجة الدالة IsDate(expr) من دوال VB تختبر أن تعبيراً ما expr هو تاريخ فتعيد True والا فتعيد False .

## ❖ كتابة الدالة DueDate في وحد نمطية Module2:

- المطلوب أن نكتب دالة DueDate نمرر لها تاريخا (تاريخ فاتورة مثلا) فتعيد تاريخ أول يوم من الشهر القادم (تاريخ وجوب الدفع) مع مراعاة مايلي:
- إذا كان أول يوم من الشهر القادم يوم أحد، فستعيد عوذا عنه يوم الاثنين التالي:
- إذا كان أول يوم من الشهر القادم يوم سبت، فستعيد عوذا عنه يوم الاثنين التالي.
- إذا كان أول يوم من الشهر القادم يوم جمعة، فستعيد عوذا عنه يوم الخميس السابق.

```
Function DueDate(d As Date) As Date
Dim Result As Date
Result = DateSerial(Year(d), Month(d) + 1, 1)
Select Case Weekday(Result)
Case 1: DueDate = Result + 1
Case 7: DueDate = Result + 2
Case 6: DueDate = Result - 1
Case Else
DueDate = Result
End Select
End Function
```

- ❖ برمجة الدالة Year(Date) من دوال VB نمرر لها تاريخا Date فتعيد عام هذا التاريخ . الدالة Month(Date) نمرر لها تاريخا Date فتعيد شهر ذلك التاريخ.
- ❖ برمجة الدالة DateSerial(year, month, day) من دوال VB نمرر لها عام year وشهرا month ويوما day ، فتعيد التاريخ الموافق.
- ❖ استدعاء الدالة DueDate على المفتاح أوجد Find في النموذج الثاني:
- المطلوب أن يكتب المستخدم تاريخا ما في صندوق النص Text1 (تاريخ الفاتورة) . بعد النقر على مفتاح أوجد Find يظهر في صندوق النص الثاني Text2 تاريخ أول يوم من الشهر القادم (تاريخ وجوب الدفع). نستدعي في إجراء المفتاح أوجد Find الدالة السابقة DueDate ، نمرر لها Text1.Text كقيمة متغير أدخل الدالة ونسند ناتج الدالة الى Text2. نختبر قبل استدعاء الدالة أن النص المدخل في Text1 هو تاريخ مقبول بأستخدام الدالة Isdate ، والا فتظهر عبارة خطأ في صندوق النص Text2.
- ❖ نكتب على المفتاح أوجد Find الاجراء التالي:

```
Private Sub Command1_Click()
If IsDate(Text1.Text) Then
Text2 = DueDate(Text1.Text)
Else
Text2 = "Error"
End If
End Sub
```

## 16 - المشروع السادس عشر

-----:-

## ربط الفيجوال بيسك بقاعدة البيانات

سنتعلم من خلال هذا المشروع مايلي:

- ❖ الاضافة Add
- ❖ البحث Search
- ❖ الحذف Delete
- ❖ العرض Display
- ❖ الترتيب Ordere

الهدف من المشروع

-----:-

ربط قاعدة البيانات مع الفيجوال بيسك ضمن برنامج خاص بمعلومات الموظفين، كما مبين في النموذج التالي:

ID	name	ger
1	ali	ma
2	muhamaad	ma
3	nada	ferr
4	suha	ferr
5	fadhil	ma
6	hadi	ma
7	salman	ma

مراحل التنفيذ

-----:-

- ❖ نفتح مشروعا قياسييا جديدا.
- ❖ نضع على النموذج عناصر التحكم المبينة في الشكل أعلاه :
- ❖ برمجة الدالة Data1

DATABASENAME	C:\DOCUMENTS AND SETTINGS\COMPUTER SYSTEM\MY DOCUMENTS\DB297.MDB
RecordSource	Table1
Visible	False

- ❖ النص الاول TextBox1 نحدد خصائصه كمايلي:

DATA FIELD	ID
Data Source	Data1
Text	فراغ

- ❖ النص الثاني TextBox2 نحدد خصائصه كمايلي:

DATA FIELD	NAME
Data Source	Data1
Text	فراغ

- ❖ النص الثالث TextBox3 نحدد خصائصه كمايلي:

DATA FIELD	GENDER
Data Source	Data1
Text	فراغ

❖ النص الرابع TextBox4 نحدد خصائصه كمايلي:

DATA FIELD	AGE
Data Source	Data1
Text	فراغ

❖ النص الخامس TextBox5 نحدد خصائصه كمايلي:

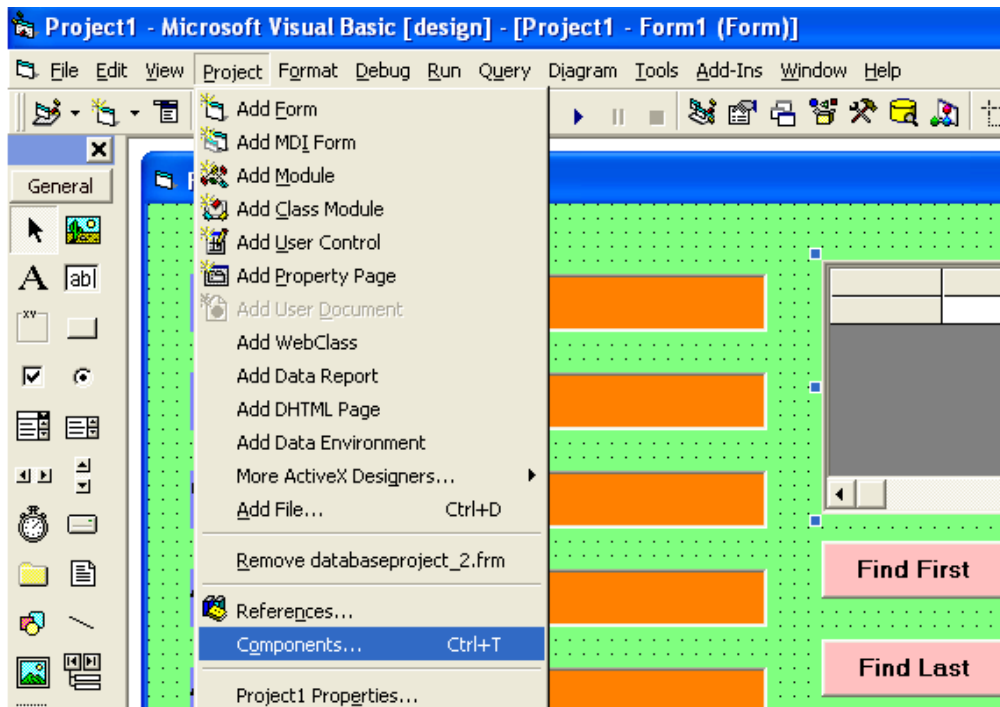
DATA FIELD	ADDRESS
Data Source	Data1
Text	فراغ

❖ النص السادس TextBox6 نحدد خصائصه كمايلي:

DATA FIELD	SALARY
Data Source	Data1
Text	فراغ

❖ برمجة الاداة Microsoft FlexGrid Control 6.0

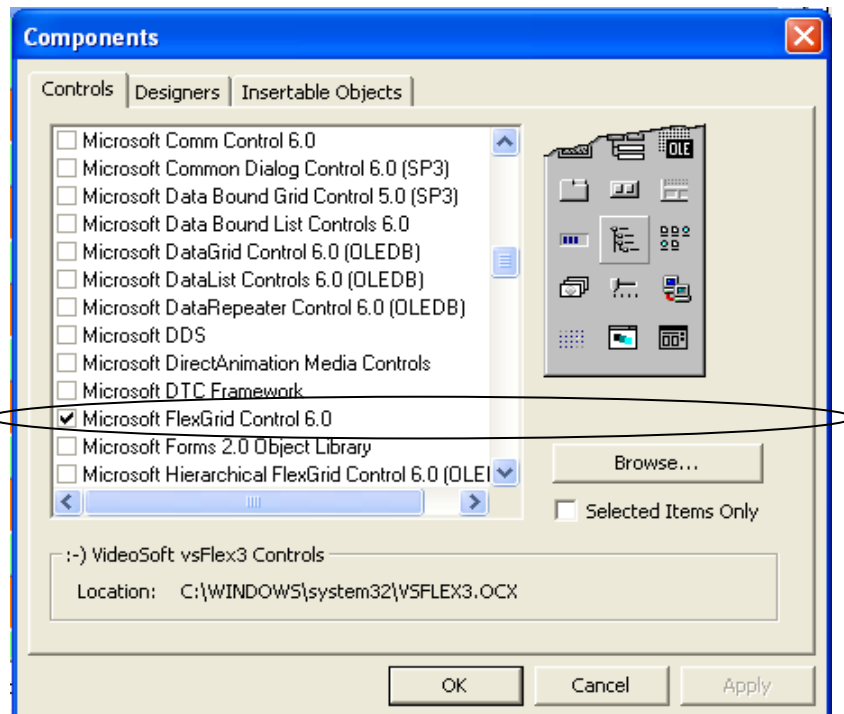
نفعل هذه الاداة MsFlexGrid عن طريق الدخول الى الامر Project ثم الامر Components عند ذلك نفعل MsFlexGrid كما موضح في الشكل التالي:





❖ ومن ثم تظهر الشاشة التالية:

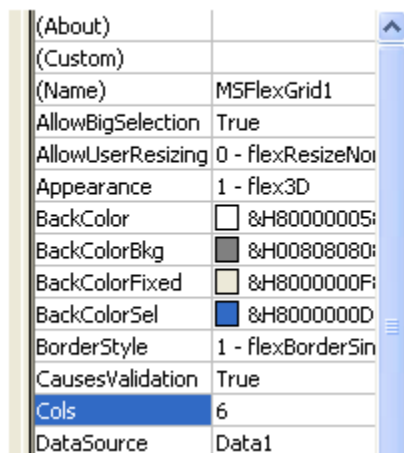
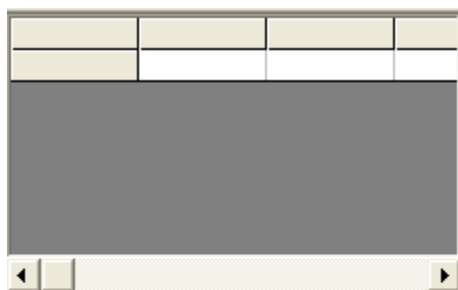
نؤشر الاختيار المطلوب بعلامة صح وكما موضح في الشكل التالي:



❖ ومن ثم نحدد خصائص هذه الدالة بتحديد الخصائص التالية:

COLS	6
DataSource	Data1

وكما موضح في الشكل التالي:



❖ برمجة المفتاح التالي Next : عند النقر على المفتاح Next نكتب الاجراء التالي:

```
Private Sub Command1_Click()  
On Error Resume Next  
Data1.Recordset.MoveNext  
If Data1.Recordset.EOF Then  
Data1.Recordset.MoveLast  
End If  
End Sub
```

❖ برمجة المفتاح السابق Previous: عند النقر على المفتاح Previous نكتب الاجراء التالي:

```
Private Sub Command2_Click()  
On Error Resume Next  
Data1.Recordset.MovePrevious  
If Data1.Recordset.BOF Then  
Data1.Recordset.MoveNext  
End If  
End Sub
```

❖ برمجة مفتاح الانتقال للسجل الاول First Record : عند النقر على مفتاح First Record نكتب  
الاجراء التالي:

```
Private Sub Command3_Click()  
Data1.Recordset.MoveFirst  
End Sub
```

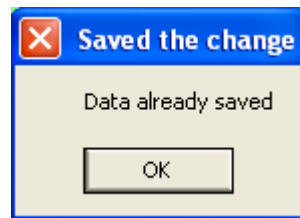
❖ برمجة مفتاح الانتقال الى السجل الاخير Last Record : عند النقر على مفتاح Last Record اي  
الانتقال الى السجل الاخير نكتب الاجراء التالي:

```
Private Sub Command4_Click()  
Data1.Recordset.MoveLast  
End Sub
```

❖ إضافة سجل Add: عند النقر على مفتاح إضافة (Add) نكتب الاجراء التالي:

```
Private Sub Command5_Click()
Data1.Recordset.AddNew
End Sub
```

❖ عملية الخزن Save عند النقر على مفتاح save سوف تظهر الواجهة التالية :



نكتب الاجراء التالي عند النقر على مفتاح Save

```
Private Sub Command6_Click()
On Error Resume Next
Data1.Recordset.Update
MsgBox "Data already saved", vbOKOnly + vbMsgBoxRight + vbMsgBoxRtlReading,
"Saved the change"
End Sub
```

❖ برمجة مفتاح الحذف Delete : عند النقر على مفتاح Delete نكتب الاجراء التالي :

```
Private Sub Command7_Click()
On Error Resume Next
x = MsgBox("Are you want to delete this record", vbOKCancel)
If x = vbOK Then
Data1.Recordset.Delete
End If
End Sub
```

❖ برمجة المفتاح Count اي حساب عدد القیود الموجوده في ملف قاعدة البيانات ، عند النقر على مفتاح Count مرتين متتاليتين نكتب الاجراء التالي:

```
Private Sub Command8_Click()
On Error Resume Next
MsgBox Data1.Recordset.RecordCount
End Sub
```

❖ برمجة المفتاح اوجد السجل الاول Find First: عند النقر على مفتاح Find First نكتب الاجراء التالي:

```
Private Sub Command9_Click()
xx = InputBox("Enter name of the employment to be search")
yy = "Name = " & xx & ""
Data1.Recordset.FindFirst yy
If Data1.Recordset.NoMatch Then
MsgBox ("The name is not found")
End If
End Sub
```

❖ برمجة المفتاح اوجد السجل الاخير Find last:

```
Private Sub Command10_Click()
On Error Resume Next
xx = InputBox("Enter name of the employment to be search")
yy = "Name = " & xx & ""
Data1.Recordset.FindLast yy
If Data1.Recordset.NoMatch Then
MsgBox ("The name is not found")
End If
End Sub
```

❖ اوجد السابق Find Previous : عند النقر على مفتاح اوجد السابق نكتب الاجراء التالي:

```
Private Sub Command11_Click()
On Error Resume Next
xx = InputBox("Enter name of the employment to be search")
yy = "Name = " & xx & ""
Data1.Recordset.FindPrevious yy
If Data1.Recordset.NoMatch Then
MsgBox ("The name is not found")
End If
End Sub
```

❖ برمجة المفتاح اوجد اللاحق Find Next : عند النقر على مفتاح اوجد اللاحق Find Next نكتب الاجراء التالي:

```
Private Sub Command12_Click()
On Error Resume Next
xx = InputBox("Enter name of the employment to be search")
yy = "Name = " & xx & ""
Data1.Recordset.FindNext yy
If Data1.Recordset.NoMatch Then
MsgBox ("The name is not found")
End If
End Sub
```

❖ برمجة المفتاح عرض السجلات Display Record : عند النقر على مفتاح عرض السجلات Display Record مرتين متتاليتين نكتب الاجراء التالي:

```
Private Sub Command13_Click()
Dim xxx
xxx = InputBox("Enter name of employment to be search")
Data1.RecordSource = " select * from table1 where name like " & xxx & ""
Data1.Refresh
End Sub
```

❖ برمجة المفتاح اصغر راتب Ordered Min: عند النقر على هذا المفتاح مرتين متتاليتين نكتب الاجراء التالي:

```
Private Sub Command14_Click()
On Error Resume Next
Data1.RecordSource = " select * from table1 order by salary "
Data1.Refresh
End Sub
```

❖ برمجة المفتاح البحث عن الراتب Search salary: عند النقر على مفتاح البحث عن الراتب Search Salary نكتب الاجراء التالي:

```
Private Sub Command15_Click()
On Error Resume Next
Dim x, y
x = InputBox("Enter salary to be search")
y = " select * from table1 where salary like '" & x & "'*"
Data1.RecordSource = y
Data1.Refresh
End Sub
```

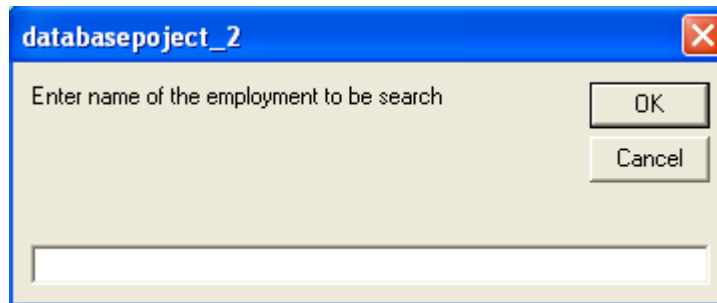
❖ برمجة المفتاح اوجد اسم الموظف الذي يمتلك اقل راتب Search Name: عند النقر على مفتاح Search Name نكتب الاجراء التالي:

```
Private Sub Command16_Click()
On Error Resume Next
Data1.RecordSource = "select * from table1 order by salary"
Data1.Refresh
End Sub
```

❖ عند النقر على احد هذه المفاتيح



❖ سوف تظهر الشاشة التالية:



❖ اذا كان الاسم غير موجود تظهر الشاشة التالية:



❖ وعند النقر على المفاتيح التالية:



❖ سوف تظهر الشاشة التالية:

