المحاضرة الثالثة

النماذج والأدوات

النموذج هو مصطلح خاص بالفيجوال بيسك مرادف للنافذة في نظام التشغيل ويندوز, وهو بالفعل نافذة تقوم أنت بتصميمها وتظهر في وقت التنفيذ كسائر نوافذ البرامج التطبيقية. أما الأدوات فهي كائنات توضع داخل النماذج وتحتضن فيها . ومن المهم أن نلفت النظر إلى وجود نوعين من الأدوات :

1- **الأدوات الداخلية** :وهي الأدوات العشرون التي توجد في صندوق الأدوات عند إنشائك مشروع قياسي Standard EXE.

2- **أدوات Active X Control**:وهي أدوات خارجية لها الامتداد OCX يمكن إضافتها عن طريق اختيار الأمر Components من القائمة Project أو بالنقر باليمين على صندوق الأدوات واختيار الأمر Components .

كل النماذج والأدوات تعتبر كائنات ولذلك هناك ثلاثة عناصر تتحكم بهذه الكائنات هي :

الخصائص(Properties),الطرق(Methods),الأحداث(Events)وكل كائن من هذه الكائنات له عناصره الخاصة به ولكن نلفت الانتباه إلى أن هناك عناصر مشتركة كثيرة بين النماذج والأدوات التي توضع عليها .

الخصائص المشتركة :

الخاصية هي قيمة تؤثر إما في شكل الكائن الخارجي مثل الخاصية BackColor أو الخاصية Font أو في سلوكه مثل الخاصية Enabled.

نافذة الخصائص هي المكان الذي يمكنك من تغيير قيمة الخاصية وقت التصميم أما وقت التنفيذ فتتغير الخصائص عن طريق كتابة الشيفرة اللازمة لذلك مثل التالي :

```
PictureBob1.BackColor=0

Label1.Caption="مرحباً بكم"
```

أو استخدام الكلمة المحجوزة With لتغيير مجموعة من الخصائص لكائن معين دفعة واحدة دون إعادة تكرار كتابة اسم الكائن :

```
With text1
        .text="نقابة المهندسين"
        .Font.Bold=True
        .BackColor=VBBlack
        .ForColor=VBWhite
End With
```

تتميز بعض الأدوات الداخلية بوجود خاصية افتراضية تغنيك عن كتابة هذه الخاصية بعد اسم الكائن فمثلاً خاصية Caption هي الخاصية الافتراضية للأداة Label عندها يمكن أن نكتب التالي :

```
Label1="نقابة المهندسين"
Text1="فرع حماة"
```

بالنسبة لنافذة النموذج يمكنك الوصول إلى خصائصها دون كتابة اسم النموذج باستخدام الكلمة المحجوزة Me أو حتى دون كتابتها نهائياً فكل العبارات التالية صحيحة :

```
Form1.caption="XXXX"
Me.Caption="XXXX"
.Caption="XXXX"
```

خاصية Name:

تعديل هذه الخاصية ممكن وقت التنفيذ فقط , وهي تمثل الاسم البرمجي للكائن أي هو الاسم الذي سنخاطب به الكائن أثناء كتابة الشيفرة وهنا ننصح بعدم الاعتماد على الأسماء الافتراضية التي يعطيها فيجوال بيسك مثل .....Form1,Form2,command1 وذلك لآنها تسبب تشويش المبرمج خاصة في البرامج الكبيرة وهنا لابد من ذكر الشروط الواجب أن نتبعها في تسمية الكائن :

- لا يبدأ برقم
- لا يزيد عن 40 حرف
- لا يحتوي على رموز خاصة ومسافات(@,$,&.....)

- أن لا يكون محجوزاً من قبل اللغة لاسم دالة أو اسم إجراء أو تابع .....

خصائص الموقع والحجم :

خصائص الموقع والحجم موجودة في جميع الأدوات القابلة للظهور(أي التي تحتوي على خاصية Visible)أما الأدوات الأخرى مثل أداة Timer فمن البديهي أن تكون هذه الخصائص غير موجودة .

خصائص الموقع للكائن(Left ,Top)تحددان موقع الزاوية العلوية اليسرى للأداة بالنسبة للأداة الحاضنة لها أو موقع الزاوية العليا اليسرى لنافذة النموذج بالمسبة للشاشة, ويمكن ضبط موقع الأداة عن طريق الشيفرة البرمجية مثل توسيط زر أمر وسط النموذج كالتالي :

```
Command1.Left=(Me.Scalewidth-Command1.width)/2

Command1.Top=(Me.ScaleHeight-Command1.Height)/2
```

خاصية Font:

جميع الأدوات التي تعرض نصوصاً على جبهتها تحتوي على هذه الخاصية والتي تحدد فيها نوع وحجم الخط المعروض على جبهة الأداة , ويمكن ضبط هذه الخاصية وقت التصميم من نافذة الخصائص أو وقت التنفيذ حسب الشيفرة التالية مثلاً :

```
With Text1
    .Font.name="Tahoma"
    .Font.Bold=True
    .Font.Size=20
End with
```

ومن المرونة التي يوفرها الكائن Font هي فكرة نسخ جميع خصائص الخط من اداة لأخرى كالتالي :

```
SetLabel1.Font=Label2.Font
```

وعند تعديل أي خاصية من خصائص الكائن Label2 يتم تعديل نفس الخاصية للكائن Label1 آلياً

خصائص اللون:

الخاصيتان BackColor,ForColor تمثلان لون الخلفية ولون الخط للأداة المحددة ,بعض الأدوات كـ ScrollBar لا تدعم هذه الخاصية فألوانها قياسية من ألوان النظام.

بعض الأدوات كـ Label لا تلحظ تغيير أي شيء إذا غيرت BackColor إلا إذا كانت قيمة الخاصية Backstyle هي 1 كذلك الحال مع الأداةCommandButton فلن تتمكن من مشاهدة التغيير اللوني لخلفية الأداة إلا إذا حولت قيمة الخاصية Styleإلى Graphical.

و الألوان تقسم إلى قسمين : ألوان قياسية Standard وألوان خاصةCustom ويفضل استخدام الألوان القياسية التي تمثل ألوان النظام ويندوز أما إذا رغبت في ألوان ثابتة لا تتغير مع ألوان النظام من لوحة التحكم فاستخدم الألوان الخاصة من نافذة الخصائص أو أحد الأكواد التالية :

```
Me.BackColor=VBGreen

Me.BackColor=VBBlue

Me.BackColor=RGB(255,0,0)
```

خصائص الجدولة :

معظم مستخدمي ويندوز يفضلون استخدام مفتاح الجدولة Tab للتنقل بين الأدوات , مع العلم أنك تستطيع معرفة الأداة المنتقاة من انتقال التركيز إليها مثلا شريط التمرير إن كان عمودياً او أفقياً يومض هذا الشريط أو يوضع مستطيل منقط حول الأداة المختارة .... إذا كل اداة قابلة للتركيز عليها لها خصائص الجدولة هي TabStop,TabIndex فالخاصية TabStop تحدد فيما إذا كانت الأداة يمكن استخدام المفتاح Tab معها أم لا بينما الخاصية الثانية TabIndex فتحدد ترتيب هذه الأداة مع مفتاح Tab مع العلم أن ترتيب الفهرسة يبدأ من الصفر .

ملاحظـــة: حتى لو كانت قيمة الخاصية TabStop هي False فإن المستخدم يستطيع أن ينقل التركيز على الأداة باستخدام الفأرة

مثـــــال :قد يرغب المبرمج أن يجبر المستخدم على كتابة شيء داخل صندوق النص وعدم مغادرته لصندوق النص هذا إلا بكتابته شيئاً فيه يمكن جعل ذلك ممكناً من خلال الكود التالي :

```
Private Sub Text1_LostFocus()
  If Trim(Text1.Text) = "" Then
    Text1.SetFocus
  End If
End Sub
```

الأداة checkbox:

تعطي هذه الأداة فرصة للمستخدم لتحديد اختيار معين إما بتفعيله أو لا , لذلك هذه الاداة سهلة التعلم ومهمة في نفس الوقت ,سنقوم بشرح كيفية استخدامها بواسطة مثال مع العلم أن قيمة التفعيل تحتجز في الخاصية Value والتي تأخذ إحدى القيم التالية :

Unchecked -0

Checked -1

Grayed-2 (رمادي)

مثال :

```
Private Sub Chcek1_Click
If Check1.Value=1 then
Image1.visible=True
End If
End Sub
```

الأداة OptionButton:

تسمى هذه الاداة في بعض الكتب Radio Button وعملها شبيه بالأداة checkbox إلا أن قيمة الخاصية Value هنا تكون trueأو False ولن تستطيع جعل قيمة Value تساوي True لأكثر من زر اختيار واحد فقط , لذلك يفضل وضع هذه الأزرار في اداة إطار Frame.

اداة القائمة ListBox:

تعرض هذه الأداة مجموعة من النصوص داخل صندوق يحوي على أشرطة تمرير scrollBars .الخاصية Sorted تقوم بفرز محتويات الأداة فرز تصاعدي بالاستناد إلى حروفها الأبجدية.تستطيع تعبئة محتويات الأداة في وقت التصميم عن طريق الخاصية List أو وقت التنفيذ باستخدام الطريقة AddItem

```
List1.AddItem "الأول"
List1.AddItem "الثاني"
```

ملاحظـة:

إذا كنت ستضيف مئات أو آلاف العناصر وقت التنفيذ فينصح بإخفاء الأداة مؤقتاً وبعد الاضافة تعيد إظهار الأداة من جديد , وذلك لأن الأداة تعيد رسم نفسها تلقائياً مع إضافة أي عنصر إليها مما يسبب بطئ الجهاز وارتعاش الأداة نفسها

```
List1.visible=False
For x=0 to 10000
  List1.AddItem x
Next
list1.Visible=True
```

العناصر الجديدة تضاف إلى نهاية سلسلة العناصر إذا كانت قيمة الخاصية Sorted هي False .كما يمكنك حذف عنصر من القائمة باستخدام الخاصيةRemoveItem أو حذف جميع العناصر بالطريقة Clear

```
List1.RemoveItem 0
List1.Clear
```

الخاصيةlistIndex :

تعود بقيمة العنصر المحدد في الأداة وتعود بالقيمة(-1)إن لم يكن هناك أي عنصر محدد أما الخاصية Text فتعود بنص العنصر المحدد

```
list1.ListIndex=0
Print List1.text
```

الخاصية listCount :

تعود هذه الخاصية بعدد جميع العناصر الموجودة والتي تستخدم بكثرة مع الخاصية List التي تمكنك من الوصول إلى العنصر

```
For x=0 to list1.listCout
  Print list1.list(0)
Next
```

بالنسبة للخاصية MultiSelect تمكن المستخدم من تحديد عدة عناصر متتالية في الاداة إذا كانت قيمتها 1-Simple أو عدة عناصر غير متتالية إذا كانت قيمة الخاصية 2-Extended.

اداة القائمة ComboBox:

معظم الخصائص والطرق للأداة listBox السابقة موجودة في الأداة ComboBox وذلك لأن الاداة ComboBoxهي الأداة listBox القياسية ولكن تحتوي على خانة نص اعلاها .تستطيع التحكم بعرض خانة النص بعدة طرق باستخدام الخاصية Style .

1- إذا كانت قيمة الخاصية Style تساوي 0-dropdown فإن أداة النص ستظهر مع سهم يؤدي النقي عليه إلى ظهور الجزء الثاني من الاداة .

2- إذا كانت قيمة الخاصية Style تساوي 1-Simple فكلا الجزأين سيظهر للمستخدم

3- أما إذا كانت قيمة الخاصية تساوي 2-dropDown list فهي نفس الخاصية أو القيمة الأولى سوى أن المستخدم لن يتمكن من الكتابة في صندوق النص .

مثال :

```
Private Sub Form1_Load()

Combo1.AddItem "مصطفى"

Combo1.AddItem "خالـــد"

End Sub
```

وبعد وضع العناصر في القائمة يمكن اختيار أحدها وتنفيذ مجموعة تعليمات تنفذ نتيجة لهذا الاختيار

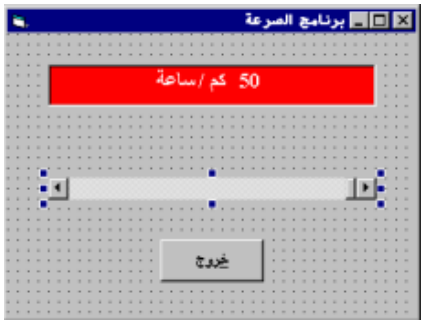
```
Private Sub Combo1_Click()

Select Case Combo1.ListIndex

Case 0
مجموعة تعليمات
Case 1
مجموعة تعليمات اخرى

End Select

End Sub
```

أشرطة التمرير ScroolBars:

تمكنك الأداتين HscrollBar و VscrollBar من محاكاة أشرطة التمرير المنتشرة في نوافذ وتطبيقات Windows .إبدا بالخاصيتين Min,Max لتحديد مجال القيم التي يمكنك قراءتها أو كتابتها عن طريق الخاصية Value والتي تمثل الموقع الحالي للمنزلقة المتحركة على شريط التمرير بعد ذلك حدد قيمة التغيير البسيط عن طريق الخاصية smallchange وهي مقدار التغيير في القيمة عندما يقوم المستخدم بالنقر على أحد أزرار أشرطة التمرير ,أما الخاصية LargeChange فهي مقدار التغيير في القيمة عندما يقوم المستخدم بالنقر على شريط التمرير نفسه.

برنامج السرعة :

يوضح المثال التالي كيفية استخدام شريط التمرير للحصول على قيمة معينة من المستخدم بحيث يظهر مؤشر شريط التمرير في مركز الشريط عند تشغيل البرنامج وتظهر عبارة كم/سا في مربع النص وعند تغيير موضع منزلقة شريط التمرير يظهر هذا التغيير في صندوق النص



كما جرت العادة سيكون هناك تمثيل مرئي للبرنامج والمرحلة الثانية هي كتابة الشيفرة

أولاً التمثيل المرئي :

القيمة	الخاصية	اسم الكائن
FrmPeed	Name	Form
برنامج السرعة	Caption	
TRUE	R To L	
CmdExit	Name	CommandButton
&خروج	Caption	
Hsbspeed	Name	HscrollBar

	Min	0
	Max	100
TextBox	Name	txtspeed
	Alignment	2-Center
	BackColor	Red
	ForColor	White
	Text	50كم/سا
	R To L	TRUE

نذهب الآن إلى نافذة الخصائص ونختار الخاصية Value لشريط التمرير ونعطيها القيمة 50 حتى توضع المنزلقة في منتصف شريط التمرير في بداة تشغيل البرنامج , ثم نختار الخاصية Min ونعطيها القيمة (0) وكذلك الخاصية Max نعطيها القيمة (100) الان لو ضغطنا على مفتاح F5 من لوحة المفاتيح سنرى شريط التمرير وقد وضع المنزلقة في منتصفه حرك المنزلقة إلى اليمين و اليسار لن ترى أي تغيير سيحصل (يمكنك تحريك المنزلقة باستخدام مفاتيح الأسهم لليمين واليسار)ولهذا ننقر على شريط التمرير الافقي مرتين للانتقال إلى مرحلة كتابة الكود.

ثانياً كتابة الكود:

في حدث change لشريط التمرير الأفقي نكتب العبارة التالية :

Txtspeed.text=Str(hsbspeed.Value)+"كم / سا"

عند التجريب تلاحظ أن مربع النص لا يتغير محتواه إلا بعد تحرير زر الفأرة ونحن نريد أن يتغير المحتوى اثناء تحريك المنزلقة إلى اليمين أو اليسار لهذا نضع العبارة التالية عند الحدث Scroll للكائن hsbspeed

Hsbspeed_change

أي نفذ الاجراء التالي عند انزلاق المنزلقة على الشريط .

التحكم في سير البرنامج :

المقصود بعملية التحكم في سير البرنامج هو تغيير سير تنفيذ التعليمات إذا تحقق شرط معين أو صادف حدوث حدث ما (ضغط المستخدم على زر Esc من لوحة المفاتيح مثلا) عندها يتم تنفيذ مجموعة تعليمات جديدة ,والجملة البرمجية المستخدمة للتحكم في سير البرنامج هي جملة IF.

هذه الجملة لا يستغني عنها أي مبرمج يعمل في أي لغة برمجية كانت ولهذا يمكن القول انها من اكثر العبارات البرمجية استخداماً هذه العبارة إما أن تنجز في سطر واحد أو عدة أسطر وهو الخيار المفضل .

امثلة :

```
1- IF X>0 THEN Y=0

2- IF X.0 THEN Y-0 ELSE Y=X
```

كما هو ملاحظ اننا لم نضع عبارة END IFفي نهاية الجملة البرمجية لأن عبارة IF وضعت على سطر واحد ولكن إذا كنا سنستخدم عبارة IF في عدة أسطر فيجب وضع عبارة END IF في نهاية التحقق من شرط معين .

أمثلة:

```
1- IF X=0 THEN
     Y=0
   END IF
2- IF M>0 THEN
     T=1
     ELSE
     T=-1
   END IF
3- IF M>0 THEN
     T=1
     ELSE IF M<0 THEN
```

```
    T=-1
    ELSE
    T=0
  END IF
```

الدالـــة IIF:

هي كاختصار لدالة IF.....ELSE كالتالي :

```
X=TEXT1.TEXT
MSGBOX IIF(X=7,"X=7","X<>7")
```

أي اجعل محتوى صندوق النص TEXT1.TEXT في المتحول X ثم أظهر رسالة بعد مناقشة حالة X إذا كانت =7فاكتب X=7 وإلا فاكتب X<>7.

التفرع باستخدام SELECT CASE:

تصلح عبارة الشرط IF إذا كان جواب الشرط احتمالين أو ثلاثة وأما إذا كنت تتوقع احتمالات كثيرة فمن المفضل أن تستخدم عبارة SELECT CASE يليها اسم المتغير الذي سيتم اختباره تأتي بعد ذلك احتمالات CASE بعد كل منها تأتي احدى قيم المتغير الذي ستتم مقارنته ثم تعقبها التعليمات التي ستنفذ إذا كان الشرط صحيحاً أو كان المتغير بهذه القيمة . واخيراً تأتي عبارة CASE ELSE ومعناها إذا كان المتغير لا يساوي أياً من القيم السابقة أو إذا لم يكن الشرط صحيحاً فإن التعليمات التي تلي ELSE هي التي تنفذ :

```
SELECT CASE TEXT1.TEXT
    CASE "محمد"
MSGBOX "مرحباً يا محمد"
   CASE "خالد"
MSGBOX "مرحباً يا خالد"
  CASE ELSE
MSGBOX "الاسم المدخل غير موجود"
END SELECT
```

ملاحظـــة :

إذا كنا سنستخدم عبارات مقارنة بعد Case فلا بد من استخدام IS معها