

الباب الخامس

المصفوفات

مفهوم المصفوفة
العمل عن المصفوفات
أنواع المصفوفات

فما سبق تناولنا العديد من الأمثلة التي تبرز أهمية استخدام المتغيرات داخل الكود ، ولكن المتغيرات مثلها مثل أي شيء لها مميزات وعيوب ، فمن مميزاتها أنه يمكن استخدام المتغيرات لتخزين قيمة معينة - سواء كانت قيمة نصية أو رقمية - تمهيداً لإجراء بعض العمليات عليها دون شعور المستخدم بذلك ، لأن معظم هذه العمليات تتم داخل الذاكرة .

ولكن المتغيرات لها بعض العيوب التي تجعل من استخدامها داخل بعض البرامج أمراً معقداً ، فمن أهم عيوب المتغيرات :

✓ لابد من الإعلان عن المتغيرات أثناء كتابة الكود ، ولا يمكن الإعلان عنها أثناء تشغيل البرنامج .

✓ كل متغير يستطيع تخزين قيمة واحدة من البيانات - مما يؤدي إلى عدم وجود مرونة أثناء التعامل مع المتغيرات ، لأنه في حالة الرغبة في تخزين عدد كبير من البيانات بشكل مؤقت فسوف يتطلب الأمر الإعلان عن عدد كبير من المتغيرات واستخدامها داخل الكود ، مما قد يؤدي إلى حدوث الخطأ .

ونتيجة لهذه العيوب فإننا نحتاج إلى استخدام نوع جديد من المتغيرات وهو المصفوفة Array .

ويمكن تشبيه المصفوفة بفندق يتكون من مجموعة من الغرف المتشابهة المتطابقة كل منها يحمل رقماً فريداً يميزها عن غيرها ولكنها في النهاية تقع تحت اسم واحد وهو اسم الفندق .

١-٥ مفهوم المصفوفة

المصفوفة هي عبارة عن متغير يحمل اسماً مميزاً ويكون هذا المتغير من مجموعة من العناصر المتتالية المحجوزة في الذاكرة وهي تحمل نفس النوع

، للإشارة إلى عنصر معين في المصفوفة نقوم بتحديد اسم المصفوفة ورقم مكان هذا العنصر في المصفوفة .

يتم تحديد عناصر المصفوفة باستخدام رقم فريد يطلق عليه فهرس Index ، كل رقم من هذا الفهرس له مساحة من الذاكرة ولذلك عليك الإعلان عن المصفوفة بالحجم المناسب حسب حاجة البرنامج حتى لا تشغل مساحة كبيرة من الذاكرة دون فائدة ، يمكن الإعلان عن المصفوفة بأي نوع من البيانات حسب متطلباتك مع الأخذ في الاعتبار أن جميع عناصر المصفوفة تحمل نفس نوع البيانات .

تستخدم المصفوفة بغرض تقليل عدد الأوامر التي يمكن كتابتها داخل الكود .

٢-٥ الإعلان عن المصفوفات Declaring arrays

✓ تشغل المصفوفات مساحة في الذاكرة ، لذلك يقوم المبرمج بتحديد نوع

المصفوفة (أي نوع البيانات التي تحملها عناصر المصفوفة) وبناءاً

عليه يقوم المترجم Compiler بحجز المساحة المناسبة من الذاكرة .

✓ يمكن الإعلان عن المصفوفات Public في الوحدة النمطية Module

أو Local ، ويمكن الإعلان عنها بصفة عامة باستخدام كلمة Dim أو

Private

Dim ArrayName (10)

Dim ArrayName (1 to 10)

ArrayName اسم المصفوفة ، 10 عدد العناصر الموجودة بالمصفوفة

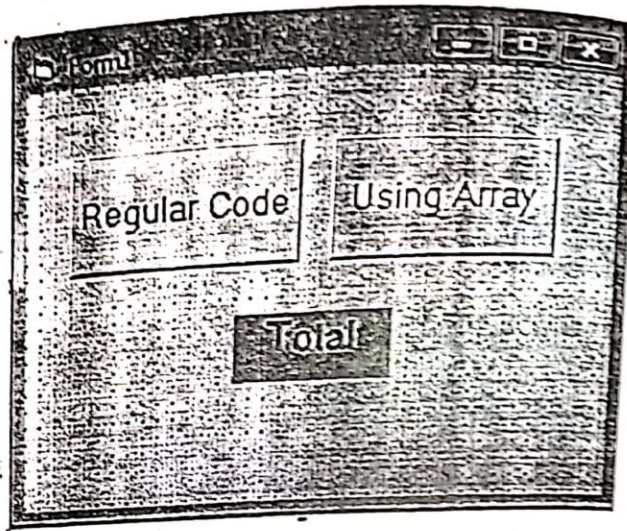
• مثال ١ :

بفرض أننا نقوم بتصميم برنامج لسوبر ماركت ، الغرض منه هو إظهار ١٠

رسائل من نوع Input Box ، حيث يقوم المستخدم بإدخال أسعار ١٠

سلع ، ثم يتم حساب إجمالي هذه السلع وعرضه داخل أداة Label ... وسوف

نقوم بتنفيذ هذا الكود من خلال أساليب البرمجة العادية وباستخدام المصفوفة ، بغرض توضيح الفرق بين الأسلوبين .
 قم بإنشاء مشروع جديد ثم أضف اثنين من الاداة Command Button
 بالإضافة إلى اداة label إلى سطح النموذج حتى يظهر بهذا الشكل :



قم بكتابة الكود التالي داخل المفتاح Regular Code وهذا الكود يوضح كيفية تنفيذ البرنامج باستخدام أسلوب البرمجة التقليدي

```
Private Sub Command2_Click()
```

```
item1 = InputBox("Enter a Price ", "Item 1")
```

```
item2 = InputBox("Enter a Price ", "Item 2")
```

```
item3 = InputBox("Enter a Price ", "Item 3")
```

```
item4 = InputBox("Enter a Price ", "Item 4")
```

```
item5 = InputBox("Enter a Price ", "Item 5")
```

```
item6 = InputBox("Enter a Price ", "Item 6")
```

```
item7 = InputBox("Enter a Price ", "Item 7")
```

```
item8 = InputBox("Enter a Price ", "Item 8")
```

```
item9 = InputBox("Enter a Price ", "Item 9")
```

البرمجة الموجهة
 item10 = InputBox("Enter a Price ", "Item 10")
 Total = Val(item1) + Val(item2) + Val(item3) +
 Val(item4) + Val(item5)
 Total = Total + Val(item6) + Val(item7) + Val(item8) +
 Val(item9) + Val(item10)
 Label1.Caption = Total
 End Sub

نلاحظ أن ١٠ أسطر في الكود السابق الغرض منهم هو عرض مجموعة رسائل متتابعة ليقوم المستخدم بإدخال أسعار ١٠ سلع .
 ثم يعمل السطر الحادي عشر والثاني عشر على حساب مجموع هذه الاسعار وتخزينها داخل المتغير Total ، بعد ذلك ينتقل الكود إلى السطر الثالث عشر الذي يقوم بعرض ناتج المتغير Total داخل الاداة Label1 .
 الكود التالي باستخدام المصفوفة Array ، قم بكتابة هذا الكود داخل المفتاح : using Array

```
Private Sub Command2_Click()
Dim item(10) As Integer
total = 0
For i = 1 To 10
item(i) = InputBox("Enter aPrice", "Item" & i)
total = total + item(i)
Next i
Label1.Caption = total
End Sub
```


نلاحظ في الكود السابق أن السطر الأول يستخدم للإعلان عن المتغير Item وهو عبارة عن مصفوفة مكونة من ١٠ عناصر من النوع integer ليتمكن التعامل مع الاسعار التي سيقوم المستخدم بإدخالها على أنها أرقام مباشرة ، بدلا من استخدام الدالة Val كما في الكود السابق .

أما السطر الثاني من الكود فالغرض منه استخدام المتغير Total الذي سوف يحتوي على مجموع الاسعار التي سيتم ادخالها ولقد تم تعيين قيمة افتراضية لهذا المتغير وهي الصفر .

السطر الثالث فيه يتم استخدام الجملة For لتكرار الكود ١٠ مرات ثم ينتقل إلى السطر الرابع وفيه نستخدم الدالة InputBox لعرض ١٠ رسائل متتالية وتخزين قيمة كل رسالة داخل المتغير Item(i) - حيث i تمثل عدد عناصر المصفوفة - وهي ١٠ .

والسطر الخامس يقوم باضافة السعر الذي سيقوم المستخدم بإدخاله في كل رسالة إلى قيمة المتغير Total ، وفي النهاية سيتم عرض قيمة المتغير Total داخل الاداة Label .

نلاحظ أن هذا الكود الذي قمنا باستخدام المصفوفة به أبسط من الكود بالبرمجة التقليدية وخاصة إذا قمنا بعمل برنامج لحساب مجموع أسعار ١٠٠ منتج مثلا أو أكثر .

٣-٥ أنواع المصفوفات

✓ هناك نوعان من المصفوفات في الفيجوال بيسك :

- Fixed size array مصفوفة ذات حجم ثابت : في هذا النوع يظل حجم المصفوفة ثابت أثناء تنفيذ البرنامج .

Dim numbers(5) As Integer

Numbers تشير إلى اسم المصفوفة ، الرقم 5 الموجود بين الأقواس يشير إلى أن عدد عناصر المصفوفة 6 عناصر بالترقيم من 0 إلى 5 .
 Dynamic array مصفوفة متغيرة الحجم : يمكن التعديل في حجم المصفوفة أثناء وقت التشغيل .

Dim DynamicArray () As String

ولكن قبل البدء في وضع القيم لعناصر هذه المصفوفة ، عليك استخدام الكلمة المحجوزة ReDim أولاً مع ذكر الحجم :

ReDim DynamicArray (2)

DynamicArray (0)= "Mona"

DynamicArray (1)="Mohammed"

DynamicArray (2)="Ali"

إذا أردت زيادة أو تقليص حجم المصفوفة، استخدم ReDim مرة أخرى
 وعليك معرفة أن جميع محتويات المصفوفة سوف تلغى:

ReDim DynamicArray (4)

DynamicArray (3)="Sayed"

DynamicArray (4)="Essam"

Print DynamicArray (4) تطبع " Essam "

Print DynamicArray (2) لا تطبع شيء

يمكن التحكم في عدم تغيير أبعاد المصفوفة الديناميكية باستخدام Preserve:

ReDim Preserve OneDim (3)

يمكنك Visual Basic من إنشاء والتعامل مع المصفوفات Arrays سواء كانت أحادية البعد أو متعددة الأبعاد .

المصفوفة أحادية البعد : هي عبارة عن مجموعة من القيم أو العناصر المنظمة من خلال صف واحد وعدة أعمدة أو عمود واحد وعدة صفوف .

المصفوفة متعددة الأبعاد : عبارة عن مجموعة من القيم منظمة في صفوف وأعمدة ، لتحديد عنصر في مصفوفة ذات بعدين نستخدم فهرسين إحدهما يشير إلى الصف والآخر للعمود ، قد تصل المصفوفة المتعددة الأبعاد في الفيجوال بيسك إلى ١٠ أبعاد ولكن الأكثر شيوعاً في الاستخدام المصفوفة ذات بعدين والمصفوفة ذات الثلاثة أبعاد .

مثال ٢ :

Dim AvgMarks (50, 50)

Dim OneDim (99) As Integer (١٠٠ عنصر)

Dim TwoDim (4, 9) As Integer (ثنائية الأبعاد)

Dim ThreeDim (2, 2, 2) As Integer (ثلاثية الأبعاد)

Dim OneDArray(0 To 10) As String

Dim TwoDArray(0 To 10, 0 To 10) As Long

Dim OneDArray(15 To 22) As String

يمكنك الآن استناد القيم للمصفوفة بعد عملية الإعلان مع العلم أن فهرس

المصفوفة Array Index يبدأ من صفر .

OneDim (0) = 100

OneDim (1) = 200

مثال ٣ :
اكتب برنامج لطباعة مصفوفة ذات بعدين (5)(5) مع ادخال اي رقم
لعناصرها .

```
Sub main()
```

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

```
Dim arr(5, 5) As Integer
```

```
For i = 0 To 5
```

```
For j = 0 To 5
```

```
arr(i, j) = 9
```

```
Debug.Print arr(i, j) & " " ;
```

```
Next j
```

```
Debug.Print vbNewLine
```

```
Next i
```

```
End Sub
```

أسئلة عامة على الباب الخامس

١. اذكر المقصود بالمصفوفة .
٢. ما هي أنواع المصفوفات .
٣. اكتب برنامج لحساب حاصل ضرب ١٠ أعداد باستخدام المصفوفات .