



# اسم المادة : البرمجة المرئية

تجمع طلبة كلية التكنولوجيا والعلوم التطبيقية - جامعة القدس المفتوحة

[acadeclub.com](http://acadeclub.com)

وُجد هذا الموقع لتسهيل تعلمنا نحن طلبة كلية التكنولوجيا والعلوم التطبيقية وغيرها من خلال توفير وتجميع **كتب وملخصات وأسئلة سنوات سابقة** للمواد الخاصة بالكلية, بالإضافة لمجموعات خاصة بتواصل الطلاب لكافة المواد:

للوصول للموقع مباشرة اضغط **هنا**

وفقكم الله في دراستكم وأعانكم عليها ولا تنسوا فلسطين من الدعاء

مجمو

عة

من

## 2. مفهوم الحدث (Event)

الحدث هو عبارة عن مجموعة من الأعمال التي يمكن إجراؤها على الكائن مثل النقر عليه، أو النقر المزدوج، أو المرور فوقه باستخدام الفارة، ولكل كائن الأحداث الخاصة به، وكل شيفرة مرتبطة بكائن من الكائنات لا بد من أن بواسطة حدث معين. هناك أحداث متنوعة لكل كائن، مما يعطي قوة إضافية لهذه اللغة ، وهذا يعني إمكانية برمجة الكائن الواحد بعدد الأحداث التي يمتلكها، ويمكن من خلال لغة البرمجة فيجوال بيسك التمييز بين هذه الأحداث، وربط كل حدث بسلوك معين (شيفرة برمجية) يتم من خلاله إجراء عمل ما على الكائن. ويمكن التعرف على الأحداث من خلال الخطوات التالية:

1- إضافة أي أداة (كائن) من جدول الأدوات داخل نموذج.

2- النقر المزدوج على هذا الكائن للانتقال إلى محرر الشيفرة (لاحظ أن عملية النقر المزدوج على كائن معين هي حدث) والشكل (1) يبين هذه الأحداث.

OX

الأحداث

Project1 - Form1 (Code)

Click

Command1

Click

Private Sub Command

**DragDrop**

**DragOver**

**GotFocus**

**End Sub**

**KeyDown**

**KeyPress**

**KeyUp**

**LostFocus**

**MouseDown**

**MouseMove**

**MouseUp**

**OLECompleteDrag**

الشكل (1) أحداث الكائن

ولكل كائن أحداث خاصة به، ويمكننا التعرف على هذه الأحداث من خلال إدراج الكائن المطلوب ثم الانتقال إلى محرر الشيفرة وملاحظة هذه الأحداث.

أسئلة التقويم الذاتي (1)

ما الخطوات التي يمكن عن طريقها التعرف على الأحداث؟

205

أحداث الفأرة ولوحة المفاتيح

3 التعامل مع أحداث الفأرة

تتعامل لغة البرمجة فيجوال بيسك

الأحداث التي تخص الفأرة، فكل عمل

تقوم به الفارة يمكن أن يؤدي إلى استجابة معينة، ويمكن للمبرمج بلغة فيجوال بيسيك أن يحدد هذه الاستجابة، ومن هذه الأحداث:

### 1- حدث النقر (Click)

وهي عملية النقر على كائن (أداة) معين، فتستجيب هذه الأداة بناء على هذا الحدث الأكثر استخداماً؛ والتطبيق التالي يوضح ذلك:

الحدث، وهو

#### التطبيق (1)

سنقوم في هذا التطبيق بالتعرف على حدث النقر Click من خلال مجموعة من الأدوات مثل إضافة زر أمر Command Button، وأداة عنوان Label، وصندوق نص Text Box إلى نموذج، وعند النقر على زر الأمر Click يتم إظهار الرسالة التالية "You Clicked In To Command Button" باستخدام مربع حوار رسائل، وعند النقر على أداة العنوان تصبح خاصية Caption لهذه الأداة

هي

Click، وعند

النقر على صندوق النص يتغير مكانه. والنقر على النموذج يغير لون الخلفية؛ والخطوات التالية توضح ا

### 1- صمم نموذج الموضح بالشكل (2):

ذلك.

X X

1

Form1 &

Click Event

**Label1**

**Text1**

الشكل (2) نموذج يحتوي على زر اواسر وأداة عنوان وصندوق نصل"

2- انتقل إلى محرر الشيفرة بالنقر المزدوج على زر الأمر؛ ستلاحظ أن الحدث

الافتراضي هو Click ثم اكتب الشيفرة الموضحة في الشكل (3)

**Project1 - Form1 (Code) -**

**Text**

**Click**

**()Private Sub Command1\_Click**

**MsgBox ("You Clicked In To Command Button")**

**End Sub**

**()Private Sub Form\_Click**

**Form1.BackColor = vb Red**

**End Sub**

**()Private Sub Label1\_Click**

**"Label1. Caption = "Click**

**End Sub**

**()Private Sub Text1\_Click**

**Text1.Top = 100**

**End Sub**

الشكل (3) الشيفرة اللازمة لحدث النقر

3- نفذ البرنامج ثم انقر على كل الكائنات لتلاحظ ظهور الشكل (4):

Form1

Text1

Click

Click Event

Project 1

X|

You Clicked In To Command Button

OF

الشكل (4) ناتج التنفيذ بعد النقر على كل الأدوات داخل النموذج

2- حدث النقر المزدوج (Double Click)

وهي عملية النقر على كائن (أداة نقرتين متتاليتين، حيث يتم تنفيذ الشيفرة

المتعلقة بهذا الكائن بعد النقر المزدوج عليه؛ و التطبيق التالي يوضح ذلك:

أحداث القارة ولوحة المفاتيح

207

- نفذ البرنامج ثم انقر نقرة مزدوجة على ادارة العنوان لتلاحظ ظهور الشكل (7):

OX

Double Click

Double Click

Copmuter Skills

Visual Basic 6

الشكل (7) ناتج التنفيذ بعد النقر المزدوج على أداة العنوان

## د حدث تحريك القارة (Move)

وهي عملية تحريك مؤشر القارة فوق كائن معين،

فعند

مرور مؤشر القارة

فوق أداة معينة تستجيب هذه الأداة لفعل معين، ويتكون هذا الحدث من الصيغة التالية

التي تكتب تلقائياً عند اختيار الحدث المطلوب:

Private Sub Control Name\_MouseMove(Button As Integer, Shift As

(Integer, X As Single, Y As Single

End Sub

حيث إن:

Control Name: اسم الأداة التي سيجري

عليها الحدث.

MouseMove: حدث تحريك مؤشر القارة.

Button: عدد صحيح يمثل قيمة زر القارة الذي تم الضغط عليه؛ والجدول (1) يمثل

قيم أزرار القارة.

Shift: عدد صحيح يمثل ما إذا كان المفتاح Shift أو Alt مضغوطة عليهما أثناء

الضغط على القارة؛ والجدول (2) يبين قيم هذه المفاتيح.

X و Y: الموقع الأفقي (السيني) والعمودي (الصادي) لمؤشر القارة.

الجدول (١) قيم أزرار القارة

اتجاه الحركة

القيمة

.2

3

4

5

6

7

Left

Right

Left and Right

Middle

Left and Middle

Right and Middle

All

الجدول (2) قيم المفاتيح للمتغير Shift

القيمة

قيم المفاتيح

1

.2

3

4

5



6

7

Shift

Ctrl

Alt

Shift + Ctrl

Shift + Alt

Ctrl + Alt

Shift + Ctrl + Alt

**التطبيق (3)**

سنتعرف في هذا التطبيق على حدث مرور مؤشر الفأرة فوق الكائنات من خلال مروره فوق زر أوامر وفوق نموذج، حيث يتم زيادة حجم الخط داخل صناديق النص عند مرور مؤشر الفأرة فوق زر الأمر Font Size، وتغيير لون خلفية صناديق النص عند مرور مؤشر الفأرة فوق زر الأمر Back Color، ويتم إظهار الإحداثيات السيني والصادي لموقع مؤشر الفأرة فوق النموذج داخل صندوق النص. والخطوات التالية توضح ذلك.

**1- صمم النموذج الموضح بالشكل (8):**

**Form1**

**G**

**Back Color**

**Font Size**

**الشكل (8) نموذج يحتوي على زر أو امر وصندوق نص**

وم

2- انتقل إلى محرر الشيفرة بالنقر المزدوج على أداة العنوان، ثم حدد الحدث MouseMove من مربع الأحداث، ثم اكتب الشيفرة الموضحة في الشكل (9)

Project1. Form1 (Code)

Form

MouseMove

,Private Sub Command1 Mouse Click (Button As Integer

(Shift As Integer, X As Single, Y As Single

Text1. Font Size = Text1. Font Size + 3

Text2. Font Size = Text1. Font Size + 3

End Sub

, Private Sub Command2 Mouse Click (Button As Integer

(Shift As Integer, X As Single, Y As Single

Text1. BackColor = vb Red

Text2. BackColor = vb Red

End Sub

,Private Sub Form MouseMove (Button As Integer

(Shift As Integer, X As Single, Y As Single

Text1.Text = X

Text 2. Text Y

**End Sub**

**الشكل (9) الشيفرة اللازمة لحدث تحريك مؤشر الفأرة**

**3- نفذ البرنامج، ثم مرر مؤشر الفأرة فوق النموذج ولاحظ الأحداث السيني والصادي**

**الموقع مؤشر الفأرة داخل صندوق النص، ثم مرر مؤشر الفأرة فوق أزرار**

**الأوامر ولاحظ ماذا يحدث لحجم الخط ولون الخلفية لصندوق النص؛ والشكل**

**(10) يبين ذلك:**

**G Form1**

**x**

**Font Size**

**Back Color**

**2070**

**1680**

**الشكل (10) ناتج التنفيذ بعد تمرير مؤشر الفأرة فوق النموذج وزري الأوامر.**

**211**

**4- حديث السحب والإفلات (Drag and Drop)**

**وهي عملية سحب كائن معين وتحريكه إلى مكان آخر فوق النموذج ثم إفلاته**

**وهذا الحدث يساعد على نقل الكائنات من مكان إلى آخر على نفس النموذج؛ والصيغة**

**التالية تكتب تلقائيا عند اختيار الحدث المطلوب:**

**,**

**,Private Sub Form\_DragDrop(Source As Control**

**(X As Single, Y As Single**

**End Sub**

حيث إن:

Source : أداة التحكم التي سيتم نقلها.

X و Y : الموقع الحالي لمؤشر الفأرة.

التطبيق (4)

سنتعرف في هذا التطبيق على حدث السحب والإفلات من خلال وضع أداة صندوق صورة ثم اختيار صورة مناسبة داخلة، ثم إمكانية سحب الصورة إلى مكان آخر على نفس النموذج ثم إفلاتها؛ والخطوات التالية توضح ذلك.

1- صمم النموذج الموضح بالشكل (11):

BOX

Form1

الشكل (11) نموذج يحتوي على أداة صندوق صورة

2- انتقل إلى محرر الشيفرة بالنقر المزدوج على النموذج، ثم حدد الحدث DragDrop من مربع الأحداث ثم اكتب الشيفرة الموضحة في الشكل (12).

Project)

-

Formt

)

Code -م

Picture1

MouseDown

,Private Sub Form\_DragDrop (Picture As Control

(X As Single, Y As Single

Picture. Move X, Y

End Sub

,Private Sub Picture1\_MouseDown (Button As Integer

(Shift is Integer, X As Single, Y As Single

picture1. Drag

End Sub

الشكل (12) الشيفرة اللازمة لحدث السحب والإفلات

ذلك

3- نفذ البرنامج، ثم حرك الصورة داخل النموذج؛ والشكل (13) يبين

Form1

X

الشكل (13) ناتج التنفيذ بعد تحريك الصورة.

؟

5- حدث الضغط على زر الفأرة و إفلاته (MouseDown)

/

(Mouse Up

وهي عملية النقر على زر الفأرة فوق كائن معين ثم تحريره، فيستجيب هذا

الكائن أثناء عملية النقر، ويستجيب أثناء عملية الإفلات والصيغة التالية تكتب تلقائيا

عند اختيار حدث الضغط:

Private Sub Control Name \_MouseDown(Button As Integer, Shift As

Integer, X As Single, Y As Single)

أما الصيغة التالية فتكتب تلقائياً عند اختيار حدث الإفلات:

Private Sub Control Name MouseUp(Button As Integer, Shift As

Integer, X As Single, Y As Single)

حيث إن:

Control Name: اسم الأداة التي ينطلق منها الحدث.

MouseDown: حدث الضغط على زر الفأرة.

MouseUp: حدث إفلات زر الفأرة.

Button: عدد صحيح يمثل قيمة زر الفأرة الذي تم الضغط عليه؛ والجدول (1) يمثل

قيم أزرار الفأرة.

Shift: عدد صحيح يمثل ما إذا كان المفتاح Shift أو Alt مضغوطة عليهما أثناء

الضغط على الفأرة؛ والجدول (2) يبين قيم هذه المفاتيح.

X و Y: الموقع الأفقي (السيني) و العمودي (الصادي) لمؤشر الفأرة.

6.

تدريب (1)

قم ببناء نموذج يتعرف على حدث الضغط على زر الفأرة وإفلاته، من خلال أداة

عنوان؛ حيث تصبح خاصية Caption عند الضغط على زر الفأرة "Mouse Down"

ولون الخلفية زرقاء، والخاصية Caption للنموذج تصبح "Mouse Down"، وعل

إفلاته تصبح الخاصية Mouse Up

"

Caption ولون الخلفية حمراء، و

Caption للنموذج تصبح "Mouse Up".

، والخاصية

#### 4. التعامل مع حدث لوحة المفاتيح

يتم التعامل مع ثلاثة أحداث تخص لوحة المفاتيح عند التعامل مع لغة البرمجة فيجوال بيسك، هي: الضغط على مفتاح من لوحة المفاتيح (Key Down)، وحدث إفلات المفتاح (Key Up)، وحدث الضغط على أحد حروف لوحة المفاتيح (Key Press)؛ وسيتم من خلال هذا الجزء التعرف على هذه الأحداث

#### أحدث الضغط على المفتاح (Key Down)

يستجيب هذا الحدث عند النقر على أي مفتاح من لوحة مفاتيح، ولكل مفتاح رمز معين يمكن من خلاله معرفة المفتاح الذي يتم الضغط عليه؛ والصيغة التالية تكتب تلقائياً عند اختيار حدث الضغط Key Down:

Private Sub Control Name\_Key Down(KeyCode As Integer, Shift As Integer)

حيث إن:

Control Name: اسم الأداة التي سيجري عليها حدث الضغط.

KeyDown: اسم الحدث.

Shif: عدد صحيح يمثل ما إذا كان المفتاح Shift أو Alt مضغوط عليهما أثناء

الضغط على الفأرة والجدول (2) يبين قيم هذه المفاتيح

KeyCode: هو عدد صحيح يقابل ثابتاً حرفياً يمثل المفتاح الذي تم الضغط عليه.

#### 2- حدث إفلات المفتاح (Key Up)

يستجيب هذا الحدث عند إفلات المفتاح الذي تم الضغط عليه من لوحة مفاتيح؛

والصيغة التالية تكتب تلقائياً عند اختيار حدث الإفلات Key Up:

**Private Sub Control Name\_KeyUp(KeyCode As Integer, Shift As Integer)**

حيث إن:

**Control Name:** اسم الأداة التي سيجري عليها حدث الإقلاط.

**KeyUp:** اسم الحدث.

**Shift :** عدد صحيح يمثل ما إذا كان المفتاح Shift أو Alt مضغوطة عليهما أثناء الضغط على الفأرة؛ والجدول (2) يبين قيم هذه المفاتيح.

**KeyCode:** هو عدد صحيح يقابل ثابتة حرفيا يمثل المفتاح الذي تم الضغط عليه.

215

أحداث الفأرة ولوحة المفاتيح

### 3- حدث الضغط على الأحرف (Key Press)

يستجيب هذا الحدث عند النقر على أي حرف من لوحة مفاتيح (ولسي

مفتاحا)، ولكل حرف رمز معين يمكن من خلاله معرفة الحرف الذي تم الضغط عليها

والصيغة التالية تكتب تلقائيا عند اختيار حدث الضغط Key Press:

**Private Sub Control Name\_KeyPress(KeyAscii As Integer)**

حيث إن:

**Control Name:** اسم الأداة التي سيجري عليها حدث الضغط على أي حرف.

**KeyPress:** اسم الحدث.

**Key Ascii:** هو عدد صحيح يمثل قيمة الحرف الذي تم الضغط عليه بنظام أسكي

ASCII.



## 5 تطبيقات عملية

( الملبن )

سنقوم في هذا التطبيق بالتعرف على حدث الضغط على أي مفتاح من لوحة المفاتيح (KeyDown) و إفلاته (KeyUp)، فعند الضغط على أي حرف يظهر الحرف بشكل كبير داخل أداة عنوان، و عند إفلاته يظهر الحرف بشكل صغير داخل أداة عنوان أخرى؛ والخطوات التالية توضح ذلك.

1- ممم النموذج الموضح بالشكل (14).

Form1

Press Any Key

Label3

Label2

الشكل (14) نموذج يحتوي على ثلاث أدوات عنوان

2- انتقل إلى محرر الشيفرة بالنقر المزدوج على النموذج، ثم حدد الحدث Key

الأحداث والحدث Key Up، ثم اكتب الشيفرة الموضحة في

Down من مربع

الشكل (15)

Project)

-

Form1

)

Code روا

( -

```

KeyUp
Form
,Private Sub Form_KeyDown (KeyCode As Integer
(Shift As Integer
Labe12.Caption = UCase (Chr (KeyCode))
End Sub
,Private Sub Form_KeyUp (KeyCode As Integer
(Shift As Integer
Labe13.Caption = LCase (Chr (KeyCode))
End Sub

```

الشكل (15): الشيفرة اللازمة لحدثي الضغط وإفلات مفاتيح لوحة المفاتيح

3- نفذ البرنامج، ثم اضغط على أي مفتاح من لوحة المفاتيح ثم حرره (افلته) ليظهر

الشكل (16)

Form 1

X

Press Any Key

q

الشكل (16) ناتج التنفيذ بعد الضغط على زر الفأرة.

كرر عملية الضغط على مفاتيح أخرى ولاحظ النتائج.

التطبيق (6)

سنتعرف في هذا التطبيق على حدث الضغط على أي حرف من حروف لوحة

المفاتيح (KeyPress)، فعند الضغط على أي حرف يظهر الحرف داخل أداة عنوان وشيفرة نظام أسكي ASCII المقابلة له داخل أداة عنوان، والخطوات التالية توضح ذلك.

1- صمم النموذج الموضح بالشكل (17).

EX

Form 1

Label2

Label1

Key Press

الشكل (17): نموذج يحتوي على أداتي عنوان وزر أوامره

2- انتقل إلى محرر الشيفرة بالنقر المزدوج على النموذج، ثم حدد الحدث Key Press من مربع الأحداث، ثم اكتب الشيفرة الموضحة في الشكل (18)

Project1 - Form1 (Code) -

Commandt

KeyPress

Private Sub Commandi\_KeyPress (KeyAscii As Integer)

Label1.Caption = KeyAscii

Label2.Caption = Chr (KeyAscii)

End Sub

الشكل (18): الشيفرة اللازمة لحدث الضغط على أي حرف من حروف لوحة المفاتيح

Key Press

3- نفذ البرنامج، ثم اضغط على أي مفتاح من لوحة المفاتيح ليظهر الحرف وشيفرة أسكي المقابلة له؛ والشكل (19) يبين ذلك.

1

G Form1

65

Key Press

الشكل (19): ناتج التنفيذ بعد الضغط على المفتاح A  
كرر عملية الضغط على مفاتيح أخرى ولاحظ النتائج.

## الوحدة الخامسة

### الإجراءات الفرعية وإجراءات الدوال

# Sub Procedures and Functions Procedures

## 2. مكونات مشروع VB6

يتكون مشروع VB6 في العادة من أجزاء (Modules)، من أبرزها أجزاء النماذج (Form Modules)، التي تتكون بدورها من أجزاء أصغر تسمى إجراءات (Procedures). تأتي الإجراءات عادة على عدة أنواع كما يلي:

أ. إجراءات الحوادث (Events)، وهي الناتجة عن تنفيذ حركة على لوحة المفاتيح (Key Events)، أو تنفيذ حركة على الفأرة (Mouse Events)، وسيتم شرح هذه النوع في وحدة لاحقة.

ب. الإجراءات المدمجة (Built)

-

ج. الإجراءات المعرفة من المستخدم أو المبرمج (User- Defined Procedures)

وهناك نوعان منها هما: الإجراءات الفرعية (Sub Procedures)، وإجراءات الدوال (Function Procedure)، حيث من المعروف أنه مهما احتوت مكتبة أية اللغة على دوال أو إجراءات جاهزة، فإنها لا يمكن أن تلبى جميع احتياجات المبرمج، ولذا كان لا بد من وجود نوع يكتبه المبرمج لتحقيق أهدافه البرمجية المختلفة، وذلك ضمن قواعد اللغة المحددة لذلك بالطبع، وسوف نناقش هذا النوع من الإجراءات ضمن هذه الوحدة كذلك، ولإنشاء إجراء ضمن بيئة (IDE)، نتبع الخطوات التالية:

أ. في وضعية ظهور نافذة برنامج (Code Window)، أي Projectl

-

Forml

(Code)، اختر Add Procedure من قائمة Tools، فيظهر صندوق الحوار

المبين في الشكل (1)

ب. حدد خياراتك الخاصة بالإجراء كما يظهر في صندوق الحوار، باختيار إسم للإجراء، وليكن **Calculate**، ثم النوع (**Type**)، وليكن **Sub**، ثم المجال (**Scope**)، وليكن **Public**، ثم انقر **OK** لتظهر لك نافذة برنامج ( **Code Window**) كما في الشكل (2)، حيث تضاف جميع الإجراءات في الجزء العام (General).

**Project1 - Form1 (Code)**

(

**(General)|**

**Calculate**

**(Private Sub Form\_Load**

**End Sub**

**(Public Sub Calculate**

**End Sub**

**الشكل (2): نافذة برنامج (Code Window)**

ج. اكتب الجمل البرمجية التي تريد بين سطر تعريف الإجراء و سطر **End Sub**

**End Functions**

**138**

= الخامسة

الإجراءات الفرعية وإجراءات الدوال

**Sub Procedures and Functions Procedures**

## في الإجراءات الفرعية (Sub Procedures)

من أهم فوائد هذه الإجراءات أنها تكتب مرة واحدة الموضوع متكرر، ويتم استدعاء الإجراءات كلما قمنا باستخدام جملة نداء في البرنامج الرئيسي، وتكون الصورة العامة للإجراء كما يلي:

**[Accessing Type] Sub Procedure Name ([Parameter List])**

**Statements**

**End Sub**

1

D

حيث:

**[Accessing Type]**: وتأتي على نوعين، إما **Public** أو **Private**، ويتحدد طي

أساسها مجال (**Scope** الإجراء).

**Procedure Name** : إسم يختاره المبرمج للإجراء، وينصح أن يتم اختياره ليبدل

على العمل المفترض أن يقوم به الإجراء.

(: هذه الأقواس لا بد من ذكرها.

**[Parameter List]**: قائمة معاملات وفقا لما يلزم ليقوم الإجراء بعمله، وقد تكون

هذه القائمة فارغة، أي لا تستخدم أية معاملات في الإجراء،

وهذا ما يعنيه استخدام الأقواس المربعة [ ]، أي أن ما

بداخلها اختياري (**optional**).

أما جملة استدعاء أو نداء الإجراء الفرعي لاستخدامه في البرنامج الرئيسي،

فهي جملة من جمل البرنامج الرئيسي تأتي على الصورة العامة التالية:

**Procedure Name ([Parameter List])**

#### 4. إجراءات الدوال (Sub Functions)

يختلف هذا النوع عن سابقه بأننا نستخدم الكلمة المفتاحية (Function) بدلا من Sub، وأننا يمكن أن نعيد قيمة (Return a Value) من خلال اسم الإجراء، ومن هنا لابد من تحديد نوع (Type) لاسم الإجراء، وفيما يلي مثال توضيحي لذلك:

مثال (2)

فيما يلي برنامج ندخل له ثلاث قيم، ويستخدم دالة لإيجاد مجموعها، ودالة أخرى لإيجاد حاصل ضربها.

**(Private Sub Command1 - Click)**

**Dim X,Y, Z, S, P As double**

**X = 2.8**

**Y = 12.5**

**Z = 30.2**

**Print "Summation of X, Y and Z="Sum (X, Y, Z)**

**Print "Product of X,Y, and Z="; Prod (X, Y, Z)**

**End Sub**

**Public Function Sum (A, B, C As double) As double**

**Sum = A+B+C**

**End Function**

**Public Function Prod (A, B, C As double) As double**

**Prod = A\*B\*C**

**End Function**

هناك الكثير من الدوال الجاهزة ضمن مكتبة VB6، وما على المبرمج إلا استخدامها دون الحاجة لكتابة قطعة برمجية لها، ويمكن تمييز عدة فئات من الدوال الجاهزة التي توفرها هذه اللغة، كما يتضح في البنود الفرعية التالية.

#### 1.4 الدوال الرياضية Mathematical Functions

تستخدم هذه الدوال عادة في بعض الأنشطة والحسابات الرياضية، ونلخص في

الجدول التالي مجموعة من هذه الدوال:

الإجراءات الفرعية وإجراءات الدوال



## Sub Procedures and Functions Procedures

## الوحدة الخامسة

اسم الدالة

وظيفة

Abs)

)

x إيجاد القيمة المطلقة للعدد X.

Sqr)

)

x | إيجاد الجذر التربيعي للعدد X.

Sin)

)

x | إيجاد جيب الزاوية x حيث x بالتقدير الدائري

Cos)

)

x | إيجاد جيب تمام الزاوية x، حيث x بالتقدير الدائري

Tan)

)

x إيجاد ظل الزاوية ، حيث x بالتقدير الدائري

Atn)

)

x تعطي الزاوية (بالتقدير الدائري) التي ظلها x

Int)

)

x تعطي أكبر عدد صحيح أقل من أو يساوي x

Sgn)

)

x تعطي إشارة العدد X

Exp)

)

x تعطي القوة x للعدد e، (أي  $e^*$ ).

Log)

)

x إيجاد اللوغريتم الطبيعي للأساس e للعدد X.

Len)

)

s | إيجاد طول العبارة أو السلسلة s

**Fix(x)** تعطي أكبر عدد صحيح أقل من أو يساوي x، إذا كانت x موجبة،

وتعطي أقل عدد صحيح أكبر من أو يساوي x، إذا كانت x سالبة.

تنتج

عبدة عشوائية x حيث 20

$x < 1$ ، علما أن x تكون من نوع

**Single**، ويمكن الضرب بمعامل لتوسيع المدى 0 إلى أحسب

الرغبة.

Val)

)

s تحول الخانات 0 - 9 بما فيها الإشارات + و - و . إلى نوع

عددي (numeric).

Rndo

Round)

)

x | إيجاد الجزء الصحيح من العدد x

Round)

)

x

y تقريب العدد x لأقرب y من الخانات العشرية

X

## 2.4 دوال السلاسل الرمزية String Functions

وتستخدم هذه الدوال للتعامل مع نوع من أنواع البيانات هو السلاسل الرمزية

(String)، ونلخص مجموعة منها في الجدول التالي:

### الوحدة الخامسة

142

## Sub Procedures and Functions Procedures

وظيفتها

Asc)

)

s تعطي القيمة العادية للحرف الأول من السلسلة 5، وذلك وفق نظام

ASCII

HexS)

)

x إن نوع ٦ التي تستخدم مع هذه الدالة هو إما عند أو سلسلة رمزية تحتوي

عددا، وتقوم الدالة بتحويل لا إلى سلسلة من نوع سداسي عشري

(hexadecimal)، أي للأساس 16.

Octs)

)

x إن نوع ٦ التي تستخدم مع هذه الدالة هو إما عند أو سلسلة رمزية تحتوي

عددا، وتقوم الدالة بتحويل إلى سلسلة من نوع ثماني (Octal)، أي

للأساس 8

Chrs)

)

n تعيد الرمز المكافئ للعدد وفقا لنظام ASCII

Str)

\$

)

n تعيد العيد على شكل سلسلة n

String)

)

nx تخزن n من رموز الصلصة x في سلسلة أخرى، وذلك من خلال جملة

تخصيص،

UCaseS)

)

s تحول أحرف السلسلة s إلى أحرف كبيرة (Upper Case) باللغة الإنجليزية

LCases)

)

s تحول أحرف السلسلة S إلى أحرف صغيرة (Lower Casc) باللغة

الإنجليزية

Rights)

)

s

,

m تعيد أول n رمزا ابتداء من يمين السلسلة S.

LeftS)

)

sun | تعيد أول n رمزا ابتداء من يسار السلسلة S.

Mid)

\$

)

s

,

k.n | لاستخلاص n من الرموز من الموقع k من السلسلة 8، ويمكن أن تستخدم

في جملة تخصيص إحلال سلسلة محل الرموز التي تتحدد.

Mid)

\$

)

sk تعيد السلسلة الجزئين من s ابتداء من الموقع k وحتى نهاية 5

Space)

)

n تعيد سلسلة تحتوي n من الفراغات

Spc)

)

n تستخدم عند الطباعة بهدف وضع n من الفراغات قبل بشه طباعة الجزء

المطلوب طباعته

Tab)

)

n لبدء طباعة الجزء المطلوب طباعته ابتداء من الفراغ n

LTrimS(8) تعيد نفس السلسلة s بعد حذف أي فراغات قد تكون على يسار \*

RFrim(5) تعيد نفس السلسلة و بن حذف أي فراغات قد تكون على يمين 5,

. او عطي

أي فراغات قد تكون على

TrimS45) تعيد تقي السلسلة و بد

يسار و

EOF)

)

n تعيد True إذا تم الوصول إلى نهاية الملف (File) ذي الرقم 1، وتعود

مدل Fa ذا لم يتم الرسول بى نهاية الملف

2

Like أو المقارنة بيليتين أو 2 حيث تعيد True في حالة التطابق، وتعيد False

در این افر هواپرا وقت الدوال

Sule Procedures and Function Procedures

### 3.4 دوال التحويل Conversion Functions

تستخدم لتحويل تعبير إلى نوع من أنواع البيانات، وذلك كما يوضح الجدول التالي لبعض هذه الدوال:

اسم الدالة

وظيفتها

CBool)

)

exp | تحول القيمة العددية للتعبير exp إلى بيانات من نوع Boolean

CByte)

)

exp تحول القيمة العددية للتعبير exp إلى بيانات من نوع Byte أي بين 0 و

255

CCur)

)

exp | تحول القيمة العددية للتعبير exp إلى بيانات من نوع Currency

CDate)

)

exp تحول بيانات exp الى بيانات من نوع Date شريطة توافق exp مع

نوع Date

CDec)

)

exp | تحول بيانات exp الى بيانات من نوع عشري (Decimal). شريطة توافق

exp مع نوع عشري (Decimal).

CInt)

)

exp | تحول بيانات exp الى بيانات من نوع صحيح (Integer) شريطة توافق

exp مع نوع صحيح (Integer)

CLng)

)

exp تحول بيانات exp الى بيانات من نوع Date شريطة توافق exp مع نوع

Long Date

اسم الدالة

#### 4.4 دوال أخرى متنوعة Other Functions

هناك العديد من الدوال الأخرى المتنوعة في لغة VB6، ونلخص في الجدول

التالي مجموعة إضافية من هذه الدوال:

وظيفتها

Load Picture)

)

s | تعيد الصورة المعرفة بالتعبير الرمزي s

MsgBox | تعرض هذه الدالة عند استخدامها صندوق حوار يزود المبرمج أو

المستخدم بمعلومات عن حالة تنفيذ البرنامج، كما يمكن للمبرمج تزويد

البرنامج بمعلومات محددة عن طريق الضغط على زر في صندوق |

الحوار، وهناك عدة أنواع وقيم لهذه الأزرار، وتحدث هذه المعلومات إما

على شكل رسالة (Message) أو أيقونة (Icon) أو عن طريق الضغط

على زر (Button)

InputBox | لإنتاج صندوق حوار (Dialog Box) ليقوم المستخدم بإدخال بيانات

نصية في المكان المحدد لذلك.

هذه الدالة لتهيئة العديد من البيانات، مثل الأعداد، أنواع العملات

النسب المئوية، التاريخ والوقت، السلاسل الرمزية، حيث تعتمد طبيعة

التهيئة على العديد من مكونات النافذة المفتوحة لهذه الغاية

VarType تستقبل هذه الدالة نوع بيانات متغير (Variant Data Type)، وينتج

عنها عدد يشير إلى نوع قيمة هذا المتغير (Variant)، مع ملاحظة أن |

Variant | هو نوع بيانات يمكنه تخزين أي نوع من البيانات باستثناء |

Format

تستخدم



الوحدة الخامسة

## Sub Procedures and Functions Procedures

الإجراءات الفرعية وإجراءات الدوال

## Rem

السلاسل الثابتة الطول، ومن الأعداد التي تعيدها هذه الدالة، ومدلول كل منها، أنظر الجدول التالي لهذا الجدول.

توضع في بداية عبارة يهدف المستخدم إلى أن تكون هذه العبارة توضيحا | أو تعليقا أو توثيقا الأمر معين في البرنامج.

InStr)

)

n

,

s1,52 | تستخدم هذه الدالة لإنجاز عملية بحث في السلسلة s (وتسمى السلسلة

الأساس لتقرير فيما إذا كانت تحتوي على السلسلة s2 (وتسمى سلسلة

البحث)، حيث يتم البحث ابتداء من المكان n في n

(

s اختيارية إذا كانت

قيمتها 1)، وتعيد هذه الدالة المكان في s1 الذي تبدأ منه s2 في حالة

وجودها، أو تعيد 0 فيما عدا ذلك،

Beep | لإصدار صوت أو نغمة من سماعات الجهاز

يجدر بالذكر أن هناك العديد من الدوال المدمجة الأخرى، مثل دوال التحقق أو

الاختبار (Checking Functions)، وكثير غيرها، وننصح الدارس بالإطلاع على

محتويات مكتبة VB6 من الإجراءات والدوال المدمجة.

## 145

الإجراءات الفرعية وإجراءات الدوال

الوحدة الخامسة

Sub Procedures and Functions Procedures

## 5. نداء الإجراءات Procedure's Calling

لاحظنا فيما مضى من نقاش أن الإجراءات من نوع الدوال (Function) تستخدم وكأنها متغير، إذ يمكن أن تدخل ضمن حسابات في التعبيرات المختلفة. أما الإجراءات الفرعية (Sub) فتستخدم جملة نداء (Calling Statement) للتمكن من الاستفادة منها واستخدامها، وذلك كما يلي

[ Call ] Procedure Name ([ Parameter List ] )

حيث:

[ Call ] : اختيارية.

**Procedure Name** : نفس اسم الإجراء الفرعي الذي استخدم عند كتابة الإجراء.

**Parameter List** : ترادف عددا ونوعا وترتيب المعاملات التي ذكرت

واستخدمت في جملة تعريف الإجراء الفرعي.

:

ويبرز هنا موضوع مهم، فما ضرورة وما عمل قائمة المعاملات

(Parameters) التي يتم الاعتناء بها وبترادفها من كل النواحي بين الأجزاء وبين

جملة النداء؟ هذه المعاملات هي الأداة التي تستخدم لتمرير البيانات والقيم بين جملة

النداء والإجراء، وكذلك العكس، إذا لزم الأمر. وتمرير المعلومات يتم بأحد الأسلوبين

التاليين:

### أ. التمرير بالقيمة (Call By Value)

في هذا الأسلوب يتم تمرير قيمة المعامل من البرنامج الرئيسي إلى الإجراء

الفرعي، وذلك دون تمرير عنوان المعامل في الذاكرة، وبناء على ذلك، يتم استخدام

القيمة في أية حسابات ضمن الإجراء الفرعي، إلا أن قيمة المعامل كما

هي مخزنة

ضمن معلومات البرنامج الرئيسي لا تتأثر، حتى لو تغيرت هذه القيمة أثناء الحسابات

في البرنامج الرئيسي، ويفيد هذا الأسلوب بحماية البيانات كما هي مخزنة في الذاكرة

أما كيفية التمرير بهذا الأسلوب فتتم باستخدام الكلمة المحجوزة By Val قبل المعامل،

وذلك إما بجملة النداء في البرنامج الرئيسي، أو بجملة تعريف البرنامج الفرعي. وفيما

يلي مثال توضيحي على ذلك:

## ب. التمرير بالعنوان أو المرجع (Call By Reference)

في هذا الأسلوب يتم تمرير عنوان المعامل من البرنامج الرئيسي إلى الإجراء الفرعي، حيث يكون الإجراء الفرعي قادراً على الوصول إلى القيمة المخزنة في الذاكرة للمعامل، فيستخدمها في الحسابات بالطبع. لكن ليس هذا وحسب، إن أي تغيير يتم على قيمة المعامل بسبب هذه الحسابات ينعكس على قيمة المعامل في الذاكرة وبالتالي في البرنامج الرئيسي. أما كيفية التمرير بهذا الأسلوب، فتتم باستخدام الكلمة المحجوزة By Ref قبل المعامل، وذلك إما بجملعة النداء في البرنامج الرئيسي أو بجملعة تعريف البرنامج الفرعي

147

حدة الخامسة

الإجراءات الفرعية وإجراءات الدوال

Sub Procedures and Functions Procedures

## 6. جمل الخروج (Exit Sub and Exit Function)

توفر لغة VB6 جملتين تمكنان المبرمج من الخروج من الإجراءات قبل إكمال تنفيذها، فإذا كتبت الجملة **Exit Sub** في أي مكان داخل الإجراء الفرعي (**Sub**) فسوف يتم ترك تنفيذ باقي جمل الإجراء التي بعد هذه الجملة، ونقل التحكم مباشرة إلى ما بعد جملة (**End Sub**). وبالمثل فإن كتابة الجملة (**Exit Function**) في أي مكان داخل الدالة (**Function**) يؤدي إلى عدم إكمال تنفيذ باقي جمل الدالة التي بعدهذه الجملة، ونقل التحكم مباشرة إلى ما بعد جملة (**End Function**).

151

الوحدة الخامسة

الإجراءات الفرعية وإجراءات الدوال

Sub Procedures and Functions Procedures

## 7. أصناف التخزين أو فترة الحياة

### (Storage Classes or Lifetime)

إن طريقة الإعلان عن المعرفات (سواء المتغيرات أو أسماء الإجراءات تحدد الوقت الذي يحتفظ خلاله بتخزين هذه المعرفات في الذاكرة، ويمكن التمييز بين نوعين من الإعلان عن المتغيرات، وبالتالي نوعين من أصناف تخزينها أو فترة حياتها في الذاكرة، وذلك كما يلي

أ. الإعلان باستخدام الكلمة المحجوزة **Dim** هكذا:

**Dim A As Integer**

**Dim Sum As Integer, Average As double**

في هذا النوع من الإعلان، يبدأ تخزين المتغيرات المعلن عنها، وهي

**Average)**

,

**Sum**

,

(A من لحظة الإعلان، وتنتهي فترة التخزين بمجرد انتهاء تنفيذ

الإجراء الذي يحتوي التعريف، أي مع لحظة الوصول إلى

**End Sub End Function**

لبدء

الإعلان باستخدام الكلمة المفتاحية **Static**، سواء مع **Dim** أو بدونها هكذا:

**Static Dim X As Integer, Y As double**

**Static W As String**

في هذا النوع من الإعلان، يبدأ تخزين المتغيرات المعلن عنها (وهي W,

,

Y

(X من لحظة الإعلان، وتنتهي فترة التخزين مع انتهاء تنفيذ المشروع (Project)

بكامله.

يجدر بالذكر أن النوع الأول هو المفترض (Default) للمتغيرات، وهو كذلك

المفترض للإجراءات. وفي حالة رغبتنا بأن يكون الإجراء من الصنف الثاني، فلا بد

من استخدام الكلمة المفتاحية المحجوزة Static في السطر الأول الذي يعرف الإجراء  
كما يلي:

**Private Static Sub Find**

Dim x As Integer, Y

## 7. أصناف التخزين أو فترة الحياة

### (Storage Classes or Lifetime)

إن طريقة الإعلان عن المتغيرات (سواء المتغيرات أو أسماء الإجراءات تحدد الوقت الذي يحتفظ خلاله بتخزين هذه المتغيرات في الذاكرة، ويمكن التمييز بين نوعين من الإعلان عن المتغيرات، وبالتالي نوعين من أصناف تخزينها أو فترة حياتها في الذاكرة، وذلك كما يلي

أ. الإعلان باستخدام الكلمة المحجوزة **Dim** هكذا:

**Dim A As Integer**

**Dim Sum As Integer, Average As double**

في هذا النوع من الإعلان، يبدأ تخزين المتغيرات المعلن عنها، وهي

**Average)**

,

**Sum**

,

(A من لحظة الإعلان، وتنتهي فترة التخزين بمجرد انتهاء تنفيذ

الإجراء الذي يحتوي التعريف، أي مع لحظة الوصول إلى

**End Sub End Function**

لبدء

الإعلان باستخدام الكلمة المفتاحية **Static**، سواء مع **Dim** أو بدونها هكذا:

**Static Dim X As Integer, Y As double**

**Static W As String**

في هذا النوع من الإعلان، يبدأ تخزين المتغيرات المعلن عنها (وهي W,

,

Y

(X من لحظة الإعلان، وتنتهي فترة التخزين مع انتهاء تنفيذ المشروع (Project)

بكامله.

يجدر بالذكر أن النوع الأول هو المفترض (Default) للمتغيرات، وهو كذلك

المفترض للإجراءات. وفي حالة رغبتنا بأن يكون الإجراء من الصنف الثاني، فلا بد



من استخدام الكلمة المفتاحية المحجوزة Static في السطر الأول الذي يعرف الإجراء  
كما يلي:

**Private Static Sub Find**

**Dim X As Integer, Y As double**

**End Sub**

ويذكر هنا أن المتغيرات المعرفة بداخل مثل هذا الإجراء تصبح بالضرورة

من نفس النوع (Static).

**152**

الوحدة الخامسة

الإجراءات الفرعية وإجراءات الدوال

Sdi Procedures and Functions Procedures

## 8. قواعد مجال المعاملات (Scope Rules of Parameters)

المجال (Scope) هو ذلك الجزء من البرنامج أو المشروع (Project) المسموح فيه بالتعامل مع المعاملات والوصول إليها، وهناك ثلاثة أنواع لهذا المجال، نوضحها كما يلي:

### أ. المجال المحلي (Local Scope):

وينطبق هذا النوع على المتغيرات التي يتم تعريفها داخل الإجراء الفرعي أو إجراء الدالة، حيث يمكن رؤيته والوصول إليه والتعامل معه داخل هذا الإجراء فقط وينتهي تعريفه والوصول إليه بمجرد وصولنا إلى جملة (End Sub) للإجراء أو (End Function) للدالة، والتي تشير إلى انتهاء تنفيذ الإجراء.

### ب. المجال الخاص (Private Scope):

ويطلق عليه أحيانا المجال المعياري (Module Scope)، حيث تستخدم الكلمة المفتاحية Private الوصف الإجراء الذي نرغب بالتعامل مع معاملاته ضمن المجال الخاص، ويعني هذا المجال بان المعاملات قابلة للرؤية والوصول إليها والتعامل معها قبل جميع إجراءات النموذج (Form)، أي أوسع من المجال المحلي، لكن لم يصل التوسع في المجال إلى إمكانية التعامل مع المعاملات من قبل جميع أجزاء المشروع (Project)، وهذا ما يوفره النوع الثالث من المجالات.

### ج. المجال العام (Public Scope):

يعتبر هذا المجال الأوسع، وتستخدم الكلمة المفتاحية المحجوزة (Public) الوصف الإجراء الذي نرغب بالتعامل مع معاملاته ضمن هذا المجال العام، حيث يمكن رؤية والوصول والتعامل مع هذه المعاملات من قبل أي إجراء آخر ضمن المشروع (Project) بأكمله.

يجدر بالذكر أن هناك متغيرات تعرف ويعلن عنها بوصفها متغيرات محلية (Local variables)، ومثل هذه المتغيرات تخضع للمجال المحلي بغض النظر عن أي اعتبار أو إعلان أو وصف آخر، كما يجدر بالذكر أن الإجراءات كذلك لها المجال (Scope) الخاص بها، فإما أن توصف (Private) ليكون من الممكن الإشارة إليها أو استخدامها من إجراءات النموذج (Form) التي هي فيه، أو أن توصف بـ (Public) ليكون من الممكن الإشارة إليها أو استخدامها من إجراءات كامل المشروع (Project)

## Sub Procedures and Functions Procedures

### الوحدة الخامسة

#### 9. المعاودة أو الاستدعاء الذاتي (Recursion)

مثل معظم لغات البرمجة، فإن لغة VB6 توفر ميزة أن يقوم الإجراء باستدعاء نفسه ذاتياً، وذلك إن كان في موضوع هذا الإجراء صفة التكرار، ويوفر هذا بديلاً له مزاياه لاستخدام تراكيب التكرار. ستخصص هذا البند لمناقشة هذا الموضوع.

التبدأ بمثال مألوف: مستورد برنامجاً يحسب مضروب عدد صحيح موجب  $n$  في

لكن بعد التذكير بالفكرة من الناحية الرياضية، حيث يعرف مضروب العدد  $n$  كما يلي:

$$\begin{aligned}
 & 2 \\
 & + \\
 & +1 \text{ مود } (n-2) \\
 & = \\
 & n \\
 & * \\
 & ) \\
 & n \\
 & - \\
 & 1 \\
 & ( \\
 & * \\
 & ) \\
 & n
 \end{aligned}$$

كما يمكن أن ينظر لهذا التعريف كما يلي:

(Recursive Step)

(Basis Step)

)

$$n! = n * (n - 1)$$

$$1 = 1$$

بمعنى أن حساب  $n!$  ينتهي لو كانت قيمة  $(n - 1)$  موجودة، أي أن مضروب العدد  $(n!)$  هنا معرف بدلالة مضروب العد كذلك لكن لقيمة أقل هي  $(n - 1)$  هنا ومن هنا نصمم البرنامج الذي سيحسب  $n!$  ليستدعي نفسه بتمرير القيمة  $n - 1$  لنفسه ويستمر في الاستدعاء الذاتي المتكرر وتمرير قيمة أقل بواحد كل مرة حتى تكون القيمة الممررة تساوي 1، حيث تنتهي الحسابات ويتحدد الجواب. إذا لنتابع البرنامج المقصود:

**Private Function RecFact (n As Integer) As Long**

**If n <= 1 Then**

**RecFact = 1**

**Else**

**RecFact=n. RecFact (n-1)**

**End If**

**End Function**

**154**

الوحدة الخامسة

الإجراءات الفرعية وإجراءات الدول

**Sub Procedures and Functions Procedures**

الشكل (3): آلية المعادة أو الاستدعاء الذاتي

الشكل (3) يوضح آلية ما يحدث في هكذا برنامج، حيث يبدأ الاستدعاء من  $n$

ويتنازل حتى 1 حيث  $1! = 1$ ، ثم نعود أدراجنا من النحسب ! 2، ثم نعود لنحسب

3!، وهكذا وصولاً إلى حساب  $n!$ .

والآن نتمعن البرنامج التالي الذي يستخدم تركيبة For

/

Next ويكافئ البرنامج

السابق الذي استخدم المعادة:

**Private Function IterativeFact (n As Integer) As Long**

**K As Long**

**Dim**

**K=1**

**j = 2 To n**

**For**

**K \*j=**

**= K**

**Next j**

**IterativeFact = K**

**End Function**

يجدر بالذكر أن أي عملية حسابات فيها تكرار، وتستند إلى عدد تتغير قيمته

من مرحلة حسابات إلى المرحلة التي تليها حتى نقطة توقف محددة، يمكن برمجتها من

خلال أسلوب المعادة، وفيما يلي مثال آخر على ذلك.

155

الإجراءات الفرعية وإجراءات الدوال

**Sub Procedures and Functions Procedures**

الوحدة الخامسة

## مثال (7)

(1) اكتب برنامجا يستخدم المعادة لإيجاد عدد فيبوناشي (Fibonacci Number) الذي

رقمه K في سلسلة هذه الأعداد التي تعرف كما يلي:

العدد الأول F

1-

العدد الثاني  $F_2 = 1$

العدد رقم K يحسب كما يلي:  $F_K = F_{K-1} + F_{K-2}$

(ب) اكتب برنامجا مكافئا يستخدم إحدى تراكيب التكرار لتحقيق نفس الهدف.

الحل:

(1) البرنامج التالي يستخدم المعادة:

**( )Private Sub Command\_Click**

**Dim K As Integer, Fib As Integer**

**"K=InputBox("Enter a Positive Integer", "Input**

**Fib = RecFib (K)**

**Print " The" ;K; "th Fibonacci Number=", Fib**

**End Sub**

**Private Function RecFib (A As Integer) As Integer**

**If (n = 0) or (n = 1) Then**

**RecFib = 1**

**Else**

**RecFib = RecFib (n - 1) + RecFib (n - 2)**

**End I**

**End Function**

(ب) البرنامج التالي يستخدم إحدى تراكيب التكرار وهي **For**

**/**

**Next** لتحقيق نفس

الهدف:

**Private Sub Command Click**

**Dim K As Integer, Fib As Integer**

**K=InputBox("Enter a Positive Integer", "Input")**

**Fib = Iterative Fib (K)**

**Print "The" ;K;"th Fibonacci Number = ", Fib**

**End Sub**

**Private Function IterativeFib(A As Integer) As Integer**

**Dim F1 As Integer, F2 As Integer**

**F1 = 1: F2 = 1**

**For I = 3 To K**

**IterativeFib = F1 + F2**

**F1 = F2**

**F2 = IterativeFib**

**Next i**

**End Function**

**156**

**الوحدة الخامسة**

**Sub Procedures and Functions Procedures**

**الإجراءات الفرعية وإجراءات الدوال**

بسمي

## 10. الأجزاء البرمجية (Code Modules)

نقدم في هذا القسم نمطا آخر من أنماط إعداد الأجزاء البرمجية المختلفة بلغة VB6، وذلك دون استخدام واجهة التخابط الرسومية (GUI)، هذا النمط أيضا الأجزاء المعيارية (Standard Modules)، ومن أهم فوائد هذا النمط أنه يتيح استخدام الإجراءات المختلفة المكتوبة من قبل عدة مشاريع، أو من قبل الأجزاء المختلفة في المشروع الواحد. إن المجال المفترض (Default Scope) هو Private كما يمكن للمبرمج أن يصف الإجراء بأنه Private، مع ملاحظة أنه في هذه الحالة يكون مجال الإجراء ومتغيراته فقط داخل الإجراء، ولذلك لا بد من استخدام Public للإجراء الذي ترغب بأن يكون مجاله في كل المشروع.

الإنشاء أي من هذه الأجزاء البرمجية، نقوم بما يلي:

أ. انقر على Project، ثم اختر Add Module من قائمة Project، فيظهر صندوق

الحوار المبين في الشكل (4)

Add Module

Existing

Now

Poo

Open

Cancel

Heb

Dont show this slog in the future

الشكل (4): صندوق حوار (Add Module)

ب. انقر Existing لإضافة إجراء موجود إلى المشروع، أو انقر New لاستحداث

إجراء جديد ضمن المشروع

ج. انقر Open لفتح الجزء البرمجي المرغوب إضافته أو استحدثه، فيتم فتح نافذة

Project()

-

Modulet



)

Code، كما في الشكل (5)، (مع ملاحظة: أن اختيار

Cancel يسمى

#### 10. الأجزاء البرمجية (Code Modules)

نقدم في هذا القسم نمطا آخر من أنماط إعداد الأجزاء البرمجية المختلفة بلغة

VB6، وذلك دون استخدام واجهة التخابط الرسومية (GUI)، هذا النمط أيضا

الأجزاء المعيارية (Standard Modules)، ومن أهم فوائد هذا النمط أنه يتيح استخدام

الإجراءات المختلفة المكتوبة من قبل عدة مشاريع، أو من قبل الأجزاء المختلفة في

المشروع الواحد. إن المجال المفترض (Default Scope) هو Private كما يمكن

اللمبرمج أن يصف الإجراءات بأنه Private، مع ملاحظة أنه في هذه الحالة يكون مجال

الإجراء ومتغيراته فقط داخل الإجراء، ولذلك لا بد من استخدام Public للإجراء الذي

ترغب بأن يكون مجاله في كل المشروع.

الإنشاء أي من هذه الأجزاء البرمجية، نقوم بما يلي:

أ. انقر على Project، ثم اختر Add Module من قائمة Project، فيظهر صندوق

الحوار المبين في الشكل (4)

Add Module

Existing

Now

Poo

Open

Cancel

Heb

Dont show this slog in the future

الشكل (4): صندوق حوار (Add Module)

ب. انقر Existing لإضافة إجراء موجود إلى المشروع، أو انقر New لاستحداث

إجراء جديد ضمن المشروع

ج. انقر Open لفتح الجزء البرمجي المرغوب إضافته أو استحداثه، فيتم فتح نافذة

Project()

-

## Modulet

)

**Code**، كما في الشكل (5)، (مع ملاحظة: أن اختيار

Cancel يخفي صندوق الحوار، واختيار Help يعرض مساعدة).

157

بسمى

## 10. الأجزاء البرمجية (Code Modules)

نقدم في هذا القسم نمطا آخر من أنماط إعداد الأجزاء البرمجية المختلفة بلغة

**VB6**، وذلك دون استخدام واجهة التخابط الرسومية (**GUI**)، هذا النمط أيضا

الأجزاء المعيارية (**Standard Modules**)، ومن أهم فوائد هذا النمط أنه يتيح استخدام

الإجراءات المختلفة المكتوبة من قبل عدة مشاريع، أو من قبل الأجزاء المختلفة في

المشروع الواحد. إن المجال المفترض (**Default Scope**) هو **Private** كما يمكن

اللمبرمج أن يصف الإجراءات بأنه **Private**، مع ملاحظة أنه في هذه الحالة يكون مجال

الإجراء ومتغيراته فقط داخل الإجراء، ولذلك لا بد من استخدام **Public** للإجراء الذي

ترغب بأن يكون مجاله في كل المشروع.

الإنشاء أي من هذه الأجزاء البرمجية، نقوم بما يلي:

أ. انقر على **Project**، ثم اختر **Add Module** من قائمة **Project**، فيظهر صندوق

الحوار المبين في الشكل (4)

**Add Module**

**Existing**

**Now**

**Poo**

**Open**

**Cancel**

**Heb**

**Dont show this slog in the future**

الشكل (4): صندوق حوار (**Add Module**)

ب. انقر **Existing** لإضافة إجراء موجود إلى المشروع، أو انقر **New** لاستحداث إجراء جديد ضمن المشروع

ج. انقر **Open** لفتح الجزء البرمجي المرغوب إضافته أو استحدثه، فيتم فتح نافذة

**Project**)

-

**Module**

)

**Code**، كما في الشكل (5)، (مع ملاحظة: أن اختيار

**Cancel** يخفي صندوق الحوار، واختيار **Help** يعرض مساعدة).

بسمي

## 10. الأجزاء البرمجية (Code Modules)

نقدم في هذا القسم نمطا آخر من أنماط إعداد الأجزاء البرمجية المختلفة بلغة

**VB6**، وذلك دون استخدام واجهة التخابط الرسومية (**GUI**)، هذا النمط أيضا

الأجزاء المعيارية (**Standard Modules**)، ومن أهم فوائد هذا النمط أنه يتيح استخدام

الإجراءات المختلفة المكتوبة من قبل عدة مشاريع، أو من قبل الأجزاء المختلفة في

المشروع الواحد. إن المجال المفترض (**Default Scope**) هو **Private** كما يمكن

اللمبرمج أن يصف الإجراء بأنه **Private**، مع ملاحظة أنه في هذه الحالة يكون مجال

الإجراء ومتغيراته فقط داخل الإجراء، ولذلك لا بد من استخدام **Public** للإجراء الذي

ترغب بأن يكون مجاله في كل المشروع.

الإنشاء أي من هذه الأجزاء البرمجية، نقوم بما يلي:

أ. انقر على **Project**، ثم اختر **Add Module** من قائمة **Project**، فيظهر صندوق

الحوار المبين في الشكل (4)

**Add Module**

**Existing**

**Now**

**Poo**

**Open**

**Cancel**

Heb

Dont show this slog in the future

الشكل (4): صندوق حوار (Add Module)

ب. انقر **Existing** لإضافة إجراء موجود إلى المشروع، أو انقر **New** لاستحداث

إجراء جديد ضمن المشروع

ج. انقر **Open** لفتح الجزء البرمجي المرغوب إضافته أو استحدثه، فيتم فتح نافذة

Projectl)

-

Modulet

)

**Code**، كما في الشكل (5)، (مع ملاحظة: أن اختيار

**Cancel** يخفي صندوق الحوار، واختيار **Help** يعرض مساعدة).

## 10. الأجزاء البرمجية (Code Modules)

نقدم في هذا القسم نمطا آخر من أنماط إعداد الأجزاء البرمجية المختلفة بلغة VB6، وذلك دون استخدام واجهة التخابط الرسومية (GUI)، هذا النمط أيضا الأجزاء المعيارية (Standard Modules)، ومن أهم فوائد هذا النمط أنه يتيح استخدام الإجراءات المختلفة المكتوبة من قبل عدة مشاريع، أو من قبل الأجزاء المختلفة في المشروع الواحد. إن المجال المفترض (Default Scope) هو Private كما يمكن للمبرمج أن يصف الإجراء بأنه Private، مع ملاحظة أنه في هذه الحالة يكون مجال الإجراء ومتغيراته فقط داخل الإجراء، ولذلك لا بد من استخدام Public للإجراء الذي ترغب بأن يكون مجاله في كل المشروع.

الإنشاء أي من هذه الأجزاء البرمجية، نقوم بما يلي:

أ. انقر على Project، ثم اختر Add Module من قائمة Project، فيظهر صندوق الحوار المبين في الشكل (4)

### Add Module

Existing

Now

Poo

Open

Cancel

Heb

Dont show this slog in the future

الشكل (4): صندوق حوار (Add Module)

ب. انقر Existing لإضافة إجراء موجود إلى المشروع، أو انقر New لاستحداث

إجراء جديد ضمن المشروع

ج. انقر Open لفتح الجزء البرمجي المرغوب إضافته أو استحدثه، فيتم فتح نافذة

Project()

-

Modulet

)

**Code**، كما في الشكل (5)، (مع ملاحظة: أن اختيار

**Cancel** يخفي صندوق الحوار، واختيار **Help** يعرض مساعدة).

**157**

الإجراءات الفرعية وإجراءات الدوال

**Sub Procedures and Functions Procedures**

الوحدة الخامسة

## Project1 - Module 1 (Code)

### (General)

### (Declarations)

الشكل (5): نافذة (Project1)

-

### Module1

(

)

### (Code

يمكن للمشروع أن يحتوي عدة أجزاء برمجية (Code Modules)، كما يمكن له أن يحتوي عدة أجزاء نماذج (Form Module)، وننوه عن أن اسم ملف الجزء البرمجي (Code Module File) ينتهي بـ .bas، وأن الأجزاء البرمجية المختلفة تجمع ضمن المشروع في مجلد (folder) اسمه Modules في الشكل (6).

X

X

## Project - Project 1

### Project1 (Project)

### Forms

### Form1 (Form1)

### Modules

### Module 1 (Module) \*

الشكل (6): مجلد Modules

أما لإظهار نافذة كتابة الجمل البرمجية للإجراء، فيتم ذلك إما بالنقر المزدوج على اسم الجزء البرمجي أو الإجراء، أو بالنقر على هذا الاسم، ثم النقر على **View**

### Code

### الوحدة الخامسة

158

الإجراءات الفرعية وإجراءات الدوال

## Sub Procedures and Functions Procedures



#### 14. مسرد المصطلحات

- إجراءات الدوال (Sub Functions): يختلف هذا النوع عن الإجراءات الفرعية

باننا نستخدم الكلمة المفتاحية (Function) بدلا من (Sub)، وأنا يمكن أن نعيد

قيمة (Return a Value) من خلال اسم الإجراء، ومن هنا لا بد من تحديد نوع

(Type) لاسم الإجراء

- الإجراءات الفرعية (Sub Procedures): هي أجزاء البرنامج التي تكتب مرة

واحدة لموضوع متكرر، ويتم استدعاؤها كلما لزم باستخدام جملة نداء في

البرنامج الرئيس

- التمرير بالعنوان أو المرجع (Call By Reference): في هذا الأسلوب يتم تمرير

عنوان المعامل من البرنامج الرئيس إلى الإجراء الفرعي، حيث يكون الإجراء

الفرعي قادرا على الوصول إلى القيمة المخزنة في الذاكرة للمعامل، فيستخدمها

في حساباتها بالطبع. لكن ليس هذا وحسب، إن أي تغيير يتم على قيمة المعامل

بسبب هذه الحسابات ينعكس على قيمة المعامل في الذاكرة، وبالتالي في

البرنامج الرئيس.

160

#### الوحدة الخامسة

الإجراءات الفرعية وإجراءات الدوال

#### Sub Procedures and Functions Procedures

**التمرير بالقيمة (Call By Value):** في هذا الأسلوب يتم تمرير قيمة المعامل من

البرنامج الرئيس إلى الإجراء الفرعي، وذلك دون تمرير عنوان المعامل في

الذاكرة، وبناء على ذلك يتم استخدام القيمة في أية حسابات ضمن الإجراء

الفرعي، إلا أن قيمة المعامل كما هي مخزنة ضمن معلومات البرنامج الرئيس

لا تتأثر، حتى لو تغيرت هذه القيمة أثناء الحسابات في البرنامج الرئيس.

- **جمل الخروج (Exit Sub and Exit Function):** توفر لغة VB6 جملتين تمكنان

المبرمج من الخروج من الإجراءات قبل إكمال تنفيذها، فإذا كتبت الجملة **Exit**

**Sub** في أي مكان داخل الإجراء الفرعي (**Sub**)، فسوف يتم ترك تنفيذ باقي

الإجراءات التي بعد هذه الجملة، ونقل التحكم مباشرة إلى ما بعد جملة **End**

(**Sub**)

- **دوال التحويل (Conversion Functions):** تستخدم لتحويل تعبير إلى نوع من

أنواع البيانات.

- **الدوال الرياضية (Mathematical Functions):** تستخدم هذه الدوال عادة في

بعض

الأنشطة والحسابات الرياضية.

- **دوال السلاسل الرمزية (String Function):** وتستخدم هذه الدوال للتعامل مع

نوع من أنواع البيانات هو السلاسل الرمزية (**String**).

- **المجال (Scope):** هو ذلك الجزء من البرنامج أو المشروع (**Project**) المسموح

فيه بالتعامل مع المعاملات والوصول إليها.

- **المجال الخاص (Private Scope):** ويطلق عليه أحيانا المجال المعياري (**Module**

**Scope**)، حيث تستخدم الكلمة المفتاحية **Private** الوصف الإجراء الذي نرغب

بالتعامل مع معاملاته ضمن المجال الخاص، ويعني هذا المجال بان المعاملات

قابلة للرؤية والوصول إليها والتعامل معها قبل جميع إجراءات النموذج

(**Form**)، أي أوسع من المجال المحلي، لكن لم يصل التوسع في المجال إلى

إمكانية التعامل المعاملات من قبل جميع أجزاء المشروع (**Project**)، وهذا

ما يوفره النوع الثالث من المجالات.

- **المجال العام (Public Scope):** يعتبر هذا المجال هو الأوسع، وتستخدم الكلمة

المفتاحية المحجوزة (**Public**) لوصف الأجزاء الذي نرغب بالتعامل مع

معاملاته ضمن هذا المجال العام، حيث يمكن رؤية والوصول إلى التعامل مع هذه المعاملات من قبل أي إجراء آخر ضمن المشروع (Project) بأكمله.

161

**التمرير بالقيمة (Call By Value):** في هذا الأسلوب يتم تمرير قيمة المعامل من البرنامج الرئيس إلى الإجراء الفرعي، وذلك دون تمرير عنوان المعامل في الذاكرة، وبناء على ذلك يتم استخدام القيمة في أية حسابات ضمن الإجراء الفرعي، إلا أن قيمة المعامل كما هي مخزنة ضمن معلومات البرنامج الرئيس لا تتأثر، حتى لو تغيرت هذه القيمة أثناء الحسابات في البرنامج الرئيس.

- **جمل الخروج (Exit Sub and Exit Function):** توفر لغة VB6 جملتين تمكنان

المبرمج من الخروج من الإجراءات قبل إكمال تنفيذها، فإذا كتبت الجملة **Exit Sub** في أي مكان داخل الإجراء الفرعي (**Sub**)، فسوف يتم ترك تنفيذ باقي الإجراءات التي بعد هذه الجملة، ونقل التحكم مباشرة إلى ما بعد جملة **End**

(**Sub**)

- **دوال التحويل (Conversion Functions):** تستخدم لتحويل تعبير إلى نوع من أنواع البيانات.

- **الدوال الرياضية (Mathematical Functions):** تستخدم هذه الدوال عادة في بعض

الأنشطة والحسابات الرياضية.

- **دوال السلاسل الرمزية (String Function):** وتستخدم هذه الدوال للتعامل مع نوع من أنواع البيانات هو السلاسل الرمزية (**String**).

- **المجال (Scope):** هو ذلك الجزء من البرنامج أو المشروع (**Project**) المسموح فيه بالتعامل مع المعاملات والوصول إليها.

- **المجال الخاص (Private Scope):** ويطلق عليه أحيانا المجال المعياري (**Module**)

(**Scope**)، حيث تستخدم الكلمة المفتاحية **Private** الوصف الإجراء الذي نرغب بالتعامل مع معاملاته ضمن المجال الخاص، ويعني هذا المجال بان المعاملات

قابلة للرؤية والوصول إليها والتعامل معها قبل جميع إجراءات النموذج

(**Form**)، أي أوسع من المجال المحلي، لكن لم يصل التوسع في المجال إلى

إمكانية التعامل بالمعاملات من قبل جميع أجزاء المشروع (**Project**)، وهذا

ما يوفره النوع الثالث من المجالات.

- **المجال العام (Public Scope):** يعتبر هذا المجال هو الأوسع، وتستخدم الكلمة المفتاحية المحجوزة (**Public**) لوصف الأجزاء الذي ترغب بالتعامل مع معاملاته ضمن هذا المجال العام، حيث يمكن رؤية والوصول إلى التعامل مع هذه المعاملات من قبل أي إجراء آخر ضمن المشروع (**Project**) بأكمله.

161

- **المجال المحلي (Local Scope):** وينطبق هذا النوع على المتغيرات التي يتم تعريفها داخل الإجراء الفرعي أو إجراء الدالة، حيث يمكن رؤيته والوصول إليه والتعامل معه داخل هذا الإجراء فقط، وينتهي تعريفه والوصول إليه بمجرد وصولنا إلى جملة (**End Sub**) للإجراء، والتي تشير إلى انتهاء تنفيذ الإجراء.
- **المعاودة أو الاستدعاء الذاتي (Recursion):** مثل معظم لغات البرمجة، فإن لغة VB6 توفر ميزة أن يقوم الإجراء باستدعاء نفسه ذاتياً، وذلك إن كان في موضوع هذا الإجراء صفة التكرار، ويوفر هذا بديلاً له مزاياه لاستخدام تراكيب التكرار.

162

الوحدة السادسة

المصفوفات في لغة فيجوال بيسك 6

**Arrays in Visual Basic 6**

## 2. مقدمة عامة

عندما نتعامل مع متغير معين، فإننا نقوم بالإعلان عنه باستخدام الجملة البرمجية التالية **Dim Variable as Data\_Type**، وهذا يعني حجز موقع في الذاكرة

له اسم ومحتوى وحجم معين حسب النوع البياني، ويمكن تخزين قيمة واحدة فقط في كل

مرة داخل هذا المتغير، وهذا ما يسمى بالمتغير البسيط ( **Simple Variable**). وعند

التعامل مع متغيرين، فإننا نعلن عنهما وتعامل معهما بنفس الأسلوب، وهكذا إذا كان

عدد المتغيرات بسيطاً. لكن ماذا يحدث إذا كان عدد المتغيرات ؟؟ فعلى سبيل

المثال إذا أردنا حساب معدل خمسين طالبا فإننا نحتاج إلى الإعلان عن خمسين

متغيرة، وخمسين جملة إدخال العلامات كل طالب، وخمسين جملة معالجة (حساب

المعدل)، وخمسين جملة إخراج (طباعة)؛ وهذا يعني أن عملية البرمجة أصبحت

طويلة ومملة، فماذا يحدث إذا كان عدد الطلاب أكبر من ذلك؟ ولهذه الغاية أوجدت

لغة البرمجة فيجوال بيسك - كما في لغات البرمجة عالية المستوى - حلاً لهذه المشكلة

التعامل مع عدد كبير من المتغيرات)، وهي إمكانية تخزين مجموعة من البيانات

ذات النوع البياني الواحد داخل الذاكرة تحت اسم واحد، بحيث يمكن التعامل مع كل

عنصر من عناصر المجموعة بصورة منفصلة باستخدام المصفوفاته والمصفوفة مجموعة من العناصر يكون لها نفس النوع البياني، وترتبط بعضها مع بعض بطريقة معينة ويكون لها الاسم نفسه، ولكل عنصر من عناصر

المصفوفة، فهرس معين يميزه عن بقية العناصر (موقع العنصر داخل المصفوفة). أما

فهرس المصفوفة، فهو رقم متسلسل يبين ترتيب العنصر داخل المصفوفة، ويجب أن

يكون عندة صحيحة موجبة ( $0,1,2,3,\dots$ )، والشكل التالي يوضح آلية خزن المصفوفة

داخل الذاكرة

الذاكرة

الفهرس (Subscript)

(0)

20

12

X(1)

50

40

X(99)

X(100)

نلاحظ من الشكل السابق ما يلي

1- اسم المصفوفة هو X

- حجم المصفوفة (طولها عدد العناصر) هو 101.

169

المصلوقة في لفة فيجوال بيسك 6

الوحدة السادسة

**Arrays in Visual Basic 6**



3- عملية فهرسة المصفوفة تبدأ من الفيرس رقم صفر ( Zero Based ) في هذا المثال.

4- كل عنصر من عناصر المصفوفة له فهرس وله محتوى، فمثلا 20  
.X(0)

وبذلك فقد سهلت المصفوفات إمكانية التعامل أعداد كبيرة من المتغيرات شريطة أن يكون لها نفس النوع البياتي دون الحاجة إلى الإعلان عن كل متغير بشكل

مستقل وإنما الإعلان عن متغير واحد من نوع مصفوفة، وأصبح بالإمكان استخدام

البيانات المخزنة داخل المصفوفة في أكثر من موقع داخل البرنامج مما أدى إلى زيادة

فاعلية البرنامج، وسنتعرف من خلال هذه الوحدة على كيفية التعامل مع المصفوفات

من خلال الإعلان عنها، وإدخال قيم إليها، وطباعة عناصرها؛ للمصفوفات أحادية

البعد، ومتعددة الأبعاد، والعمليات التي يمكن إجراؤها على المصفوفات.

10

170

المصارفك في نفة فرورل هومة )

Arrays in Visual Basic 6

### 3. مصفوفة البعد الواحد والإعلان عنها

عناصر المتتالية التي تقع على نفس السطر، أو نفس العمود، تسمى مصفوفة أحادية البعد (**One Dimensional Array**)، ويمكن الإعلان عنها باستخدام الجملة التالية:

**Dim Array\_Name(Subscript) As Data\_Type**

حيث إن:

**Dim**: جملة الإعلان، ويمكن استبدالها بإحدى الكلمات التالية **Public**,

,

**(Private**

**(Static**

**Array\_Name**: اسم المصفوفة ويخضع لقواعد التسمية.

**Subscript**: الفهرس، ويمثل أعلى رقم نريده المتغيرات المصفوفة (طول المصفوفة).

**Data\_Type**: أحد الأنواع البيانية التي توفرها اللغة.

تبدأ عملية الفهرسة للمصفوفة من الفهرس رقم صفر، وتستمر حتى تصل إلى

رقم الفهرس الذي يمثل الحد الأخير في المصفوفة، ويمكن التحكم بالقيمة الابتدائية في

الفهرس كما سنرى لاحقاً.

## مثال (1)

Dim Array(100) As Integer

الحد الأعلى

ونلاحظ من جملة الإعلان السابقة ما يلي:

نوعها

الحد الأدنى

نوعها

حجمها

المصفوفة

البياني

أحادية البعد | Integer

الفهرس

الفهرس

Array

101

0

100

ال

duan

## Obat Ibon

ملاحظات:

- كل مصفوفة تستخدم داخل البرنامج يجب الإعلان عنها، وفي حال استخدام

مصفوفة غير معلن عنها فإن رسالة الخطأ التالية ستظهر **Sub or**

**Function Not**

از "

يجب التقيد بالمدى المحدد للمصفوفة، فمثلاً:

**"Defined.**

**Dim A(5) As Integer**

**A(6)=10**

خارج المدى لأن رقم الفهرس للعنصر النهائي هو 5، وستظهر رسالة الخطأ

التالية

**"Sub Script out of Range"**

**171**

المصفوفات في لغة فيجوال بيسك 6

الوحدة السادسة

**Arrays in Visual Basic 6**



- لا علاقة لقيمة الفهرس بمحتوى العنصر، فيمكن وضع أي قيمة داخل العنصر

حسب النوع البياني، فمثلا:  $A(6)=10$  لا علاقة فيه بين الفهرس ومحتوى العنصر.

- عناصر المصفوفة داخل الذاكرة متتابة، أي أنها مرتبة حسب الفهرس، وهذا لا

يمنع الرجوع إلى أي عنصر بشكل عشوائي فمثلا:  $X(8)$

=

A

)

5

(

+

A هي ناتج جمع

الموقع الخامس مع الموقع الثامن،

- يجب التقيد بالنوع البياني لعناصر المصفوفة عند إدخال عناصرها.

**كيفية التعامل مع فهرس المصفوفة**

يمكن أن يبدأ فهرس المصفوفة بالقيم التالية:

1- في الوضع الاعتيادي يبدأ فهرس المصفوفة بالقيمة صفر، فمثلا

**Dim A(10) As Integer**

القيمة الابتدائية

القيمة النهائية

الحجم

0

10

11

2- يمكن أن تبدأ عملية الفهرسة من القيمة واحد (One Based)، وذلك باستخدام جملة

1

**Option Base** التي توضع قبل بداية البرنامج.

مثال (2)

عند الإعلان التالي فإن قيمة الفهرس هي كما يبين الجدول التالي:

**Option Base 1**

**Private Sub Command1 Click**

**Dim A(10) As Integer**

القيمة الابتدائية

القيمة النهائية

الحجم

10

11

3- يمكن أن تبدأ قيمة الفهرس بأي قيمة، وذلك باستخدام الكلمة  
المحجوزة To

والجدول التالي يبين قيمة الفهرس لجملة الإعلان التالية:

**Dim A( 10 To 100) As Integer**

القيمة الابتدائية

القيمة النهائية

الحجم

**10**

**100**

**91**

**172**

الوحدة السادسة

المصفوفات في لغة فيجوال بيমে (

Arrays in Visual Basic 6



يمكن حساب حجم المصفوفة من خلال المعادلة التالية  
حجم المصفوفة - قيمة الفهرس النهائية - القيمة الابتدائية + 1

### مثال (3)

اعلن عن المصفوفات التالية:

-1

X حجم 40 من نوع صحيح بحيث يشير الفهرس 1 إلى العنصر الأول.

- Y حجم 100 من نوع رمزي.

- 2 حجم 150 من نوع حقيقي بحيث يشير الفهرس 10 إلى العنصر الأول.

الحل:

[

Option Base 1 .1

Dim X(1241) As Integer

Dim Y(99) As String .2

Dimn Z(10 To 159) As Single .3

Real

double

%0.0

173

المصفوفك في لغة فيجوال بيسك 6

#### 4. مصفوفة البعدين ومتعددة الأبعاد والإعلان عنها

تدعم لغة البرمجة فيجوال بيسك إمكانية التعامل مع المصفوفات ذات البعدين (Two Dimensional Arrays) التي تتكون من مجموعة صفوف ومجموعة

أعمدة، وتحتاج إلى فهرسين للتعامل مع عناصرها، أما المصفوفات متعدد

الأبعاد (Multi Dimensional Arrays)، فهي تتكون من أكثر من بعدين وتحتاج إلى

فهارس بعند الأبعاد، ويتم الإعلان عن هذه المصفوفات بالطريقة نفسها التي استخدمت

للإعلان عن المصفوفات أحادية البعد، مع إضافة فهرس لكل بعد حسب الصيغة

التالية:

**Dim Array\_Name (Index1, Index2) As Data\_Type**

جملة الإعلان عن المصفوفة ثنائية البعد حيث [Index الفهرس الأول و Index2 الفهرس الثاني].

**Dim Array\_Name(Index1,Index2,...) As Data\_Type**

جملة الإعلان عن المصفوفة متعددة الأبعاد، حيث يتم تحديد عدد الفهارس بند

الأبعاد.

#### مثال (4)

أعلن عن مصفوفة ثنائية الأبعاد تتكون من 4 صفوف ود أعمدة من نوع

صبح

**Dim X(3,4) As Integer**

جملة الإعلان السابقة تحجز 20 موقعا في الذاكرة حسب الشكل التالي:

العمود 4

العمود 3

العمود 2

العمود 1

العمود 0

الصف 0

N(0.4)

X(14)

x(24)

X(0,3)

X(13)

X(23)

$X(33)$

الصف 1

$X(0,2)$

$X(1,2)$

$(22)$

$X(3,2)$

$X(0,1)$

$X(1,1)$

$X(2,1)$

$X(3,1)$

$X(0,0)$

$X(1,0)$

$X(20)$

$X(30)$

الصف 2

$X(34)$

الصف 3

لاحظ أن عملية التعامل مع الفهرس في المصفوفات أحادية البعد تشبه  
عملية

التعامل معه في المصفوفات ثنائية ومتعددة الأبعاد.

الوحدة السادسة

**174**

المصفوفات في لغة فيجوال بيسك 6

Arrays in Visual Basic 6

## مثال (5)

أعلن عن المصفوفات التالية:

- مصفوفة ثنائية البعد عدد صفوفها 3 وعدد الأعمدة 5 من نوع رمزي.

2- مصفوفة ثنائية البعد عدد عناصرها 100 تستخدم لخزن معدلات الطلبة.

- مصفوفة ثلاثية الأبعاد بعدها الأول 4، والثاني 3 والثالث 2 من نوع صحيح.

الحل:

**Dim X(1 to 3,1 to 4) As String (1)**

**Or**

**Dim X(2,4) As String**

**Dim X(20,5) As Double (2)**

**Dim X(1 to 4,1 to 3,1 to 2) As Integer (3)**

**or**

**Dim X(3,2,1) As Integer**

يمكن الوصول إلى أي عنصر من عناصر المصفوفة بتحديد فهرسيه،  
فعلى

سبيل المثال العنصر  $X(1,2)$  يقع في الصف الثاني والعمود الثالث،  
ويمكن استخدام

**Option Base** لبدء الفهرس من القيمة 1 وليس 0.

**175**

**المصفوفات في لغة فيجوال بيسك 6**

**الوحدة السادسة**

**Arrays in Visual Basic 6**

**Empty)**

## 5 قراءة وطباعة محتويات مصفوفة

عملية الإعلان عن المصفوفة تحجز موقعا في الذاكرة أيمنه الابتدائية  
(Default) صفر عندما تكون المصفوفة من نوع عادي، ويأخذ القيمة  
..

**String)** عندما تكون المصفوفة رمزية، ويمكن إدخال قيم إلى عناصر  
المصفوفة

باستخدام جملة لتخصيص كما يلي

**Dim X(6) As Single**

جملة الإعلان المسابقة تحجز 7 مواقع داخل الذاكرة وفق الشكل التالي:

**RAM**

**X(0)**

**N(1)**

**X(2)**

**X(3)**

**X(4)**

**X(5)**

**X(6)**

ولاتخل قيم إلى المصفوفة السابقة تستخدم جمل التخصيص التالية:



$$X(0)=5$$

$$X(1)=6$$

$$X(2) \ 42$$

$$X(3) \ 81$$

$$X(4) \ 23$$

$$X(5)=74$$

$$X(6) \ 34$$

تصبح الذاكرة بعد عملية التخصيص بالشكل التالي:

$$X(0)$$

$$(1)$$

$$X(2)$$

$$X(3)$$

$$X(4)$$

$$X(5)$$

$$X(6)$$

RAM

5

6

42

81

23

34

ويمكن إجراء أي عملية معلمة على عناصر المصفوفة كما يلي

$Y=\{2\}+X(4)$

$UX(23)X(5)$

**Print X(6),Y,Z**

176

الوحدة الاوة

المصلوكت في لفة فيجوال بيسة «

**Arrays in Visual Basic 6**

## مثال (6)

أعلن عن مصفوفة ثنائية البعد: عدد صفوفها 3 وعدد الأعمدة 4، ثم  
خصص

قيمة لعناصرها.

**Dim Y(2,3) As Integer**

$$Y(0,0) = 5$$

$$Y(0, 1) = 4$$

$$Y(0,2)=8$$

$$Y(1,0) = 10$$

$$Y(1,1) = 11$$

$$Y(1,2)= 18$$

$$Y(2,0) = 20$$

$$Y(2,1) = 68$$

$$Y(2,2)= 74$$

$$Y(0,3) = 000$$

$$Y(1,3) = 000$$

$$Y(2,3) = 000$$

## مثال (7)

أعلن عن مصفوفة ثلاثية الأبعاد: البعد الأول 2 والثاني 3 والثالث 2: ثم

خصص قيمة لعناصرها.

**Dim Z(1,2,1) As Integer**

$$Y(0, 0, 0) = 5$$

$$Y(0,1,0) = 4$$

$$Y(0, 2,0) = 8$$

$$Y(0,2,0) = 10$$

$$Y(1,0,0) = 11$$

$$Y(1,1,0) = 18$$

$$Y(1,2,0) = 20$$

$$Y(0, 0,1 = 20$$

$$Y(0,1,1) = 20$$

$$Y(0,2,1 = 20$$

$$Y(1,0,1) = 20$$

$$Y(1,2,1 = 20$$

$$Y(1,2,1 = 10$$

يمكن معرفة عدد عناصر المصفوفة متعددة الأبعاد بضرب أبعادها هكذا

البعد الأول : البعد الثاني

عملية تخصيص عناصر مصفوفة عملية غير فعالة عندما يكون حجم

المصفوفة كبيراً، ولهذا الغرض نستخدم إحدى جمل التكرار كما سافرى  
لاحقاً

**177**

**المصفوفات فى لغة فيجوال بيسك 6**

**الوحدة السادسة**

**Arrays in Visual Basic 6**

## 6. المصفوفة الديناميكية

قد يحتاج المبرمج إلى زيادة حجم المصفوفة أو إنقاصها بعد الإعلان عنها، أو

أنه لا يعرف الحجم المناسب للمصفوفة التي سيستخدمها، وبالتالي فإن الطريقة التقليدية

التعريف المصفوفة (الطريقة الثابتة) لا تفي بالغرض، ولهذا السبب وفرت لغة البرمجة

فيجوال بيسك إمكانية التحكم بإعادة تعريف المصفوفة، وتحديد الحجم المطلوب بطريقة

ديناميكية على مستوى الإجراء باستخدام الجملة **ReDim** كما يلي:

- يمكنك زيادة حجم مصفوفة تم الإعلان عنها وتغيير نوعها البياني وكما في المثال التالي:

**Dim X(5) As Integer**

**10=1**

**ReDim X(1) As Integer**

**ReDim X(100) As Single**

في جملة الإعلان الأولى تم تحديد مصفوفة حجمها 6، ثم إعادة تعريفها بحجم

جديد هو 11 عنصرا مع إلغاء جميع القيم القديمة، وإنشاء مصفوفة فارغة بنفس النوع

البياني، أما جملة الإعلان الثالثة فتم فيها تغيير حجم المصفوفة والنوع البياتي، وهذا

يعني مصفوفة جديدة بحجمها ونوعها مع بقاء اسم المصفوفة القديم. يمكن الاحتفاظ بالقيم القديمة التي تم تخزينها داخل المصفوفة عند إعادة تعريفها ،

وزيادة حجمها باستخدام الكلمة المحجوزة **Preserve** بعد الكلمة المفتاحية

**ReDim**: ومثال ذلك:

**ReDim Preserve Book (5)**

**ReDim Book (100) As Integer**

جملة الإعلان الأولى قامت بحجز ستة مواقع في الذاكرة، ثم تم تخصيص قيم

لهذه العناصر، ويتم الاحتفاظ بهذه المواقع عند إعادة تعريف المصفوفة بسين

الكلية المحجوزة **Preserve** ولا داعي لتحديد النوع البياني للمصفوفة، أن نظم

فيجوال بيسك يفترض انك تعيد عملية تعريفها وتحديد النوع البياني المطلوب، وفي

جملة الاعلان الثانية تم زيادة حجم المصفوفة مع بقاء القيم القديمة مخزنة داخلها ؛

المصفوفات في ثقة فيجوال بيسك (

**Arrays in Visual Basic 6**



ويمكن تخصيص قيم للعناصر الجديدة ابتداء من العنصر Book(6)  
يمكن الإعلان عن مصفوفة معينة دون تحديد حجمها (عدد عناصرها)؛  
ومثال  
ذلك:

**Dim X() As Integer**

**ReDim X(10)**

جملة الإعلان الأولى قامت بتعريف مصفوفة لم يتم تحديد عدد  
عناصرها، أما

جملة الإعلان الثانية فقد تم فيها تحديد عدد عناصر المصفوفة، ولم يتم  
تحديد نوعها

البياني لأن النوع البياني تم تحديده في جملة الإعلان الأولى، وبالتالي  
فإن حجم

المصفوفة هو 11 عنصرا.

- يمكن تحرير مواقع الذاكرة المحجوزة لعناصر المصفوفة وإعادةتها إلى  
الذاكرة ليتم

استخدامها لأغراض أخرى باستخدام جملة (Erase)؛ والمثال التالي  
يوضح ذلك:

8

**Dim Sun(10) As Integer**

**ReDim Sun (100)**

## Erase Sum

تم الإعلان عن مصفوفة، ثم إعادة تعريفها، وفي الجملة الأخيرة تم تحرير

المصفوفة كاملة

من الذاكرة لاستخدامات أخرى، أما في حال استخدام المصفوفة،  
الثابتة فيتم مسح عناصرها ليتم التخزين عناصر جديدة داخل نفس  
المصفوفة، ولا يتم

تحريرها من الذاكرة؛ ومثال ذلك:

## Dim Num (100)

## Erase Num

في الجملة الأخيرة تم مسح جميع عناصر المصفوفة Num ليتم إعادة  
تخزينها

بقيم جديدة

إن ما ينطبق على المصفوفات الديناميكية أحادية البعد ينطبق على  
المصفوفات

الديناميكية متعددة الأبعاد.

179

المصفوفات في لغة فيجوال بيسك 6

الوحدة السادسة

## Arrays in Visual Basic 6



## 7. المصفوفات وجمل التكرار

إن جمل التكرار قد سهلت كثيرا من عملية التعامل مع المصفوفات، والوصول

إلى جميع عناصر المصفوفة بجمل محدودة، فعندما تكون المصفوفة كبيرة الحجم 100

(عنصر مثلا) فإننا نحتاج إلى 100 جملة تخصيص الإدخال قيم إليها، وتحتاج إلى

100 جملة لطباعتها؛ وهذا يعتبر مم" وشاقا، ولهذا الغرض يمكننا استخدام تراكيب

التكرار كما في الأمثلة التالية:

**Dim Num (100) As Integer**

**For I=0 To 100**

**Num(I)=10**

**Print Num(I)**

**Next I**

في مقطع البرنامج السابق تم الإعلان عن مصفوفة حجمها 101 عنصر، وتخصيص القيمة 10 لكل عنصر من عناصرها، ثم طباعة عناصر هذه المصفوفة

باستخدام تركيبة التكرار **For**

/

**. Next**

**Dim X (1 To 20) As Integer**

**For I=1 To 20**

**X(I)=I^2**

**Print I, X(I)**

**Next I**

في مقطع البرنامج السابق تم الإعلان عن مصفوفة حجمها 20 عنصر،  
وتخصيص مربع قيمة الفهرس لكل عنصر من عناصرها، ثم طباعة  
قيمة الفهرس

وعناصر هذه المصفوفة باستخدام تركيبة التكرار For

/

**.Next**

**Dim Y (5 To 25) As String**

**For I=5 To 25**

**Y(D=InputBox('Enter Elements Of Array')) Print I, Y(I)**

**Next I**

في مقطع البرنامج السابق تم الإعلان عن مصفوفة قيمة الفهرس  
الابتدائية لها

هي 5 والنهائية هي 25، ثم إدخال عناصر المصفوفة باستخدام صندوق بحيث يظهر

هذا الصندوق عند كل عملية إدخال، ثم تم طباعة قيمة الفهرس وعناصر هذه

المصفوفة باستخدام تركيبة التكرار **For**

/

**Next** مع ملاحظة أن القيمة الابتدائية

والنهائية للفهرس هي القيمة الابتدائية والنهائية لتركيبية التكرار. وعند إجراء أية عملية

على المصفوفة فإنه يمكن استخدام تركيبة تكرار كما سنرى لاحقا.

الوحدة السادسة

**180**

المصفوفات في لغة فيجوال بيسك 6

**Arrays in Visual Basic 6**

## مثال (8)

سنقوم في هذا التطبيق بما يلي:

- أ- الإعلان عن مصفوفة من نوع صحيح حجمها 10.
- ب- إدخال عناصر المصفوفة بحيث يكون كل عنصر هو ناتج تكعيب فهرسه.
- ج- طباعة عناصر المصفوفة داخل صندوق صورة
- صمم النموذج الموضح بالشكل (1)

لا |

**Form1**

**a**

**Printing Array**

**Enter Array**

الشكل (1): نموذج يحتوي على زري أولمر وأداة صندوق صورة

181

الصفوف في نفة قبول است

**Arrays in Visual Basic 6**

2- انتقل إلى محرر الشيفرة بالنقر المزدوج على زري الأوامر، ثم اكتب الشيفرة

الموضحة في الشكل (2).

**Project 1 - Form1 (Code)**

**Commrrano2 |**

**Click**

**Option Base 1**

**Dim X(10) As Integer**

**( )Private Sub Command1\_Click**

**For I = 1 To 10**

**(I) = I 43\***

**Next I**

**End Sub**

**( )Private Sub Command2\_Click**

**"Picture1. Print "Index", "Array**

**٣٢ picture1.Print " = ===== , " = = = = =**

**For I 1 To 10**

**Picture1. Print I, X(I)**



**Next I**

**End Sub**

**الشكل (2): الشيفرة اللازمة لإدخال وطباعة عناصر مصفوفة.**

**3- نفذ البرنامج ثم انقر على زر الأمر (Enter Array) ثم انقر على**

**(Printing**

**Array) لتلاحظ ظهور الشكل (3)**

**S Form1**

**EX x**

**Enter Array**

**Printing Array**

**Index**

**Array**

**1**

**2**

**1**

**8**

**27**

**E**

125

216

343

512

729

1000

E

8

9

10

**الشكل (3): ناتج إدخال وطباعة عناصر المصفوفة.**

**182**

الوحدة السادسة

المصفوفات في لغة فيجوال بيسك 6

Arrays in Visual Basic 6

وللتعامل مع المصفوفات متعددة الأبعاد نحتاج إلى تراكيب تكرارية متداخلة

**(Nested For) بعدد الأبعاد؛ ومثال ذلك :**

```
Dim Y (5,4) As String
```

```
For I=0 To 5
```

```
For J= 0 To 4
```

```
Y(1,1)=InputBox("Enter Elements Of Array") Print I, J,
```

```
Y(I,J)
```

```
Next J
```

```
Next I
```

لاحظ أنه تم استخدام فهرسين لتحديد عناصر المصفوفة ثنائية الأبعاد، وهذا ما

ينطبق على ثلاثية الأبعاد، فهي تحتاج إلى ثلاثة تراكيب تكرارية متداخلة.

**183**

المصفوفات في لغة فيجوال بيسك 6

Arrays in Visual Basic 6

## 8. المصفوفات والإجراءات

كما تقدم، فإن الإجراءات قد تحتاج إلى تمرير معاملات إليها من أجل الخال أيم

إلى الإجراءات، أو إجراء عمليات معالجة على هذه المعاملات، أو إعادة تقييم فيها

وهناك نوعان من التمرير هما: التقرير بالعنوان ( Call BY Reference ) حيث يستم

إرسال عنوان المتغير إلى الإجراء، وأي عملية تعديل تتم على المعاملات الشكلية

داخل الإجراء، تؤثر على المعاملات الحقيقية والتمرير بالقيمة ( call By Value

حيث يتم إرسال قيمة المتغير إلى الإجراء، وهذا يعني أن أي عملية تعديل على

المعاملات الشكلية داخل الإجراء لا تؤثر على المعاملات الحقيقية، ما العمل إذا كان

المتغير المراد إرساله إلى الإجراءات هو مصفوفة ؟

عند إرسال متغيرات إلى الإجراء Sum نستخدم الجملة التالية:

**Private Sub Sum(X,Y)**

أما في حال إرسال مصفوفة إلى إجراء فتتم وفق الجملة التالية:

**Dim Array(1 to 10) As Integer**

**Private Sub Sum(Array)**

لاحظ أنه لم يتم تحديد عدد عناصر المصفوفة داخل الإجراء، وذلك لأن نظام

فيجوال بيسك سيحدد من خلال القوسين أن المعامل هو مصفوفة فيبدا بالبحث عن هذه

المصفوفة، داخل الذاكرة ثم يتعامل معها حسب الغرض من الإجراء (أي أنه تم إرسال

عنوان المصفوفة وليس قيمها)، وهذا يعني أن أي تعديل على المصفوفة المرة

سيؤثر على المصفوفة الحقيقية، ولإرسال قيم عناصر المصفوفة يجب وضع كلمة

**(BY Val)** في الإجراء كما في الجملة التالية:

**Private Sub Sum(By Val Array)**

ويمكن إرسال قيم او عناوين أحد عناصر المصفوفة كما في المثال التالية

**Private Sub Sum(By Val Array(5))**

ففي الجملة السابقة تم إرسال قيمة العنصر (5) Array إلى الإجراء Sum

**Private Sub Sum(Array(5))**

وفي هذه الجملة تم إرسال عنوان العنصر (5) Array إلى الإجراء Sumn

الوحدة السادسة

المصفوفات في لغة فيجوال بيسك 6

## Arrays in Visual Basic 6

### مثال (9)

سنقوم في هذا التطبيق بإدخال عناصر مصفوفة عن طريق تربيع قيمة  
الفهرس

ثم زيادته بالقيمة 5، ثم إرسال المصفوفة إلى إجراء يقوم بطباعة  
المصفوفة داخل

صندوق صورة بعد أن يقوم بزيادة القيمة 10 لكل عنصر من عناصر  
المصفوفة

والخطوات التالية توضح ذلك:

### 1- صمم النموذج الموضح بالشكل (4)

Form1

COX

Enter The Number Of Elements

Read And Print

الشكل (4): نموذج يحتوي على زر أوامر وأداة صندوق صورة  
وصندوق نص وأداة عنوان.

### 2- انتقل إلى محرر الشيفرة بالنقر المزدوج على زر الأوامر ( Read And Print ) ثم

اكتب الشيفرة الموضحة في الشكل (5)

Project1 - Form1 (Code) -

(General)

printing

Option Base 1

Dim x0 As Integer

( )Private Sub Cominand1\_Click

N - Text1.Text

ReDim X(N)

For I = LBound (x) To UBound (X)

(I) = I \* 2 + 5 \*

Next I

Call printing (X())

End Sub

Private Sub printing (X() As Integer)

For 1 = LBound (x) To UBound (X)

(I) 10 = (1)

Picture1. Print X(I)

Next I

End Sub

**الشكل (5): الشيفرة اللازمة لإدخال وطباعة عناصر مصفوفة باستخدام الإجراءات**

المصفوفات في لغة فيجوال بيسك 6

**185**

Arrays in Visual Basic 6

الوحدة السادسة

نلاحظ من الشيفرة السابقة أنه تم تحديد عدد عناصر المصفوفة عن طريق

المستخدم بعد إدخالها، وقد استخدمنا مفهوم المصفوفة الديناميكية، وقد تم استخدام

الدالتين LBound و UBound لتحديد الحد الأدنى والحد الأعلى للمصفوفة على

التوالي، وبعد تحديد عناصر المصفوفة تم إرسالها بالعنوان إلى الإجراء printing

الطباعتها داخل صندوق صورة.

3- نفذ البرنامج ثم حدد عدد عناصر المصفوفة داخل صندوق النص، ثم انقر على

**زر الأمر (Read And Print) لتلاحظ ظهور الشكل (6)**

Form 1



Enter The Number Of Elements

5

Read And Print

16

19

24

31

40

الشكل (6): ناتج إدخال وطباعة عناصر المصفوفة.

الوحدة السادسة

186

المصفوفات في لغة فيجوال بيسك 6

Arrays in Visual Basic 6

## 9. العمليات على المصفوفات

بعد الانتهاء من إدخال عناصر المصفوفة أحادية البعد أو متعددة الأبعاد، يمكن

إجراء كثير من العمليات على العناصر المدخلة. ومن هذه العمليات:

1- نسخ مصفوفة إلى مصفوفة أخرى: ويقصد بها نقل عناصر المصفوفة  $x$  إلى

المصفوفة  $y$ ، ويجب أن تكون المصفوفتان قد تم الإعلان عنهما ولهما نفس

الحجم والنوع البياني.

2- إيجاد مجموع عناصر مصفوفة ومعدلها،

3- إيجاد أكبر وأصغر عنصر من عناصر المصفوفة.

4- إيجاد عدد العناصر التي تحقق شرطاً معينة (عدد العناصر السالبة / عدد العناصر

الزوجية | عدد العناصر الأكبر أو الأصغر من قيمة معينة / ...).

5- البحث عن عنصر داخل مصفوفة.

والتطبيق التالي يوضح هذه العمليات:

### مثال (10)

سنقوم في هذا التطبيق بإدخال رواتب عدد  $N$  من الموظفين ثم إيجاد ما يلي:

1- إيجاد مجموع الرواتب ومعدلها.

- 2- إيجاد أعلى راتب وأدنى راتب.
- 3- إيجاد عدد الموظفين الذين تزيد رواتبهم عن 150 ديناراً.
- 4- أضف عشرة دنانير إلى كل موظف ثم انسخ الناتج إلى مصفوفة جديدة.
- 5- طباعة المصفوفتين قبل الزيادة وبعد الزيادة.

**187**

الوحدة السادسة

## أ- صمم النموذج الموضح بالشكل (7)

Form1

Enter Number Of Employer

Read Array

Sumation

Average

Max

Min

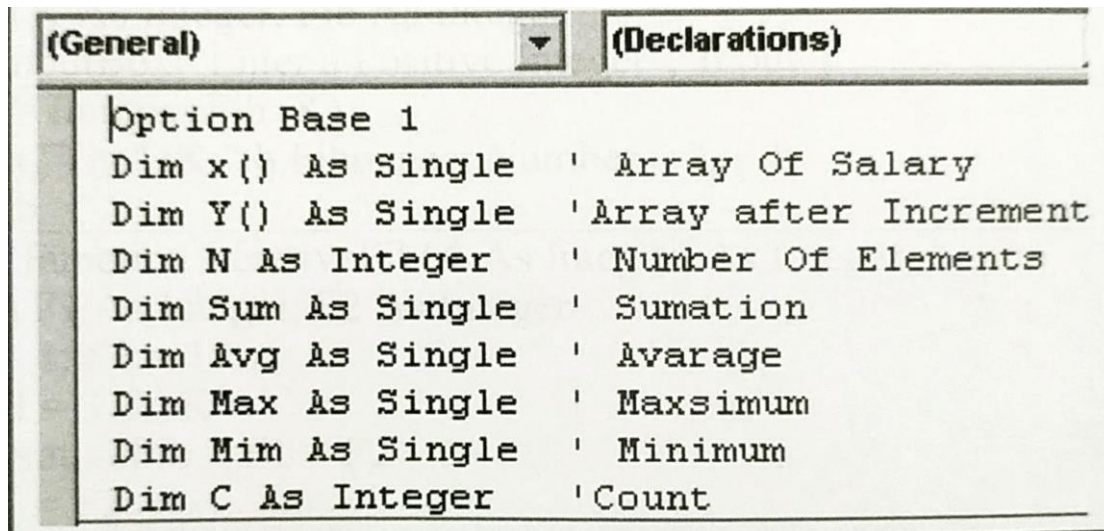
Increment

Copy And Print

Number Of Salary Grater Than 150

الشكل (7): نموذج يحتوي على أزرار أوامر وأداتي صندوق صورة  
وصندوق نص  
وأداة عنوان.

2- انتقل إلى محرر الشيفرة داخل قسم الإعلانات General  
Declaration وأعلن عن المتغيرات كما هو موضح بالشكل (8)



(

الشكل (8): الاعلان عن المتغيرات في قسم الاعلانات.

188

الوحدة السادسة

المصفوفات في لغة فيجول بيسك 6

Arrays in Visual Basic 6

3\_ انتقل إلى محرر الشيفرة بالنقر المزدوج على أزرار الأوامر  
(Read Array) ثم اكتب الشيفرة الموضحة في الشكل (٩)

```
Private Sub Command1_Click()  
N = Text1.Text  
ReDim x(N)  
For I = LBound(x) To UBound(x)  
x(I) = InputBox("Enter The Salary")  
Next I  
End Sub
```

الشكل (٩): الشيفرة اللازمة لإدخال عناصر مصفوفة

٤\_ انتقل إلى محرر الشيفرة بالنقر المزدوج على زر الأوامر  
(Sumation) ثم اكتب الشيفرة الموضحة في الشكل (١٠).

```
Private Sub Command2_Click()  
Sum = 0  
For I = LBound(x) To UBound(x)  
Sum = Sum + x(I)  
Next I  
Text2.Text = Sum  
End Sub
```

الشكل (١٠): الشيفرة اللازمة لإيجاد مجموع عناصر مصفوفة

٥\_ انتقل إلى محرر الشيفرة بالنقر المزدوج على زر الأوامر (Average) ثم اكتب الشيفرة الموضحة في الشكل (١١)

```
Private Sub Command3_Click()  
Avg = Sum / UBound(x)  
Text3.Text = Avg  
End Sub
```

الشكل (١١): الشيفرة اللازمة لإيجاد معدل عناصر مصفوفة

٦\_ انتقل إلى محرر الشيفرة بالنقر المزدوج على زر الأوامر )

```
Private Sub Command4_Click()  
Max = x(1)  
For I = LBound(x) To UBound(x)  
If Max < x(I) Then  
Max = x(I)  
End If  
Text4.Text = Max  
Next I  
End Sub
```

7- انتقل إلى محرر الشيفرة بالنقر المزدوج على زر الأوامر (Min) ثم اكتب

الشيفرة الموضحة في الشكل (13).

()

Private Sub Commands\_Click

in = x (1)

For I = LBound (x) TO UBound (x)

If Min > X(I) Then

Min = x (I)

End If

Text5. Text = Min

Next I

End Sub

الشكل (13): الشيفرة اللازمة لإيجاد أصغر عنصر من عناصر مصفوفة.

8- انتقل إلى محرر الشيفرة بالنقر المزدوج على زر الأوامر (Increment) ثم اكتب

الشيفرة الموضحة في الشكل (14)

Private Sub Command6\_Click



**ReDim Y(N)**

**For I = LBound (x) TO UBound (x)**

**Y(I) = x (1) + 10**

**Next I**

**End Sub**

**الشكل (14): الشيفرة اللازمة لزيادة عناصر المصفوفة بالقيمة 10.**

**و- انتقل إلى محرر الشيفرة بالنقر المزدوج على زر الأوامر ( Copy And Print ) ثم**

**اكتب الشيفرة الموضحة في الشكل (15)**

**( )Private Sub Command7\_Click**

**"Picture1. Print "Befor Increment**

**"Picture2.Print "After Increment**

**For I = LBound (x) TO UBound (x)**

**Picture1.Print x(I)**

**Picture2.Print Y(I)**

**Next I**

**End Sub**

**الشكل (15): الشيفرة اللازمة لطباعة عناصر المصفوفة داخل صندوق صورة**

## الوحدة السادسة

## المصفوفات في لغة فيجوال بيسك 6

10- انتقل إلى محرر الشيفرة بالنقر المزدوج على زر الأوامر

(Number Of Salary

150)

Grater Than ثم اكتب الشيفرة الموضحة في الشكل (16).

( )Private Sub Command8\_Click

c = 0

For l = LBound (x) TO UBound (x)

If x(l) > 150 Then

c = c

الشكل (16): الشيفرة اللازمة لإيجاد عدد الإعداد التي تحقق شرط معينة

أكبر من 150).

11- نفذ البرنامج، ثم ادخل في مربع النص الذي يحدد عدد العناصر 4،

ثم انقر على

زر الأمر Read Array يظهر لك أربعة صناديق إدخال كما يبين الشكل

(17).

أدخل العناصر التالية: 160,200,400,100 ثم انقر على كل الأوامر  
في النموذج

**Formt**

**Read Asas**

**Enter Number Of Employer**

**Mas**

**Average**

**Sumaton**

**100**

**400**

**215**

**860**

**Copy And Print**

**Increment**

**Number Of Salary**

**Grater Than 150**

**Befor Incremen After Increment**

**170**

**160**

**210**

200

410

400

110

100

3

الشكل (18): تنتج التنفيذ بعد بحال القيم السابقة

ما ينطبق على العمليات التي تمت على المصفوفات أحادية البعد ينطبق على

المصفوفات متعددة الأبعاد، ولكن باستخدام التركيب التكرارية المتداخلة.

سنقوم في هذا التطبيق بالتعامل مع المصفوفات ثلاثية الأبعاد من خل ما يلي:

- الإعلان عن مصفوفة ثنائية الأبعاد تكون من 3 صفوف و 4 أعمدة.

- بخل عناصر المصفوفة وطباعتها.

- ايجاد مجموع العناصر التي تقع على القصر الرئيسي

- ايجاد مجموع العناصر التي تقع على القطر الثانوي.

5- اوجد عدد العناصر التي تقع على الإطار الخارجي للمصفوفة.

192

**المصفوفات في لغة فيجوال بيت 6**

**Arrays in Visual Basic 6**

الشكل (18) يبين ذلك.

Array Complet

X|

Enter The Salary

OK

Cancel

100

الشكل (17): صندوق إدخال الرواتب.

191

المصفوفات في لغة فيجوال بيسك 6

Arrays in Visual Basic 6

الوحدة السادسة

```
Dim A(2, 3) As Single
()Private Sub Command1_Click
    For i = 0 To 2
        For j = 0 To 3
            A(i, j) = InputBox("Enter The Elements of Array")
        Print A(1,1)
        Next j
    Next i
End Sub
)Private Sub Command2_Click
    Dim sum1 As Single
    sum1 = 0
    For i = 0 To 2
        For j = 0 To 3
            If i=j Then
sum1 = sum1 +A(1,3)
            End If
        Next j
```



```
Next i
Print sum1
End Sub
()Private Sub Command3_Click
    Dim sum2 As Single
    sum2=0
    For i = 0 To 2
        For j = 0 To 3
            If (i+j) = 3 Then
                sum2 = sum2 + A(1,1)
            End If
        Next j
    Next i
    Print sum2
End Sub
)Private Sub Command4_Click
    Dim sum2 As Single
    sum2=0
```

```
For i = 0 To 2
For j = 0 To 3
If (i = 0) or (j = 0) or (i = 3) Or (j = 3) Then
sum3 = sum3 + A(1,1)
End If
Next j
Next i
Print sum3
End Sub
```

193

المصفوفات في لغة فيجوال بيسك 6

Arrays in Visual Basic 6

الوحدة السادسة

العناصر

10. طرق الإضافة والإلغاء من القوائم (Add

/

(Remove Item

يستخدم صندوق القائمة (ER

)

List Box للتعامل مع مجموعة من

مصفوفة)، حيث يمكن إضافة وحذف ومسح عناصر القائمة، ومن أهم خصائص

صندوق القائمة ما يلي

- Name: لتحديد اسم القائمة.

-2

List: لتحديد عناصر القائمة أثناء التصميم

-3

List Index: لتحديد فهرس العنصر المشار إليه، وهو يشبه فهرس المصفوفة لكنه

يبدأ بالقيمة صفر دائما

ويمكن إجراء مجموعة من العمليات على صندوق القائمة، ومن هذه العمليات: إضافة

عنصر إلى القائمة

- إضافة عنصر إلى القائمة (List 1. AddItem X)  
)

Additem

حذف عنصر من القائمة (RemoveItem)

مسح عناصر القائمة (Clear)

يمكن إضافة عنصر X إلى القائمة List1 باستخدام الجملة البرمجية التالية:

List 1. AddItem X(AddItem)

يمكن حذف عنصر من عناصر القائمة Listi باستخدام الجملة البرمجية التالية:

List1.Removeitem (list1.ListIndex)

يمكن مسح عناصر القائمة List1 باستخدام الجملة البرمجية التالية:

List1.Clear

يمكن إضافة عناصر إلى القائمة أثناء عملية التصميم من خلال الخاصية List؛

والشكل (19) يبين ذلك:

(ع)

Form1

**List**

**Mouselcon AB**

**MousePointer ALDA**

**MultiSelect**

**OLEDragMode**

**OLEDrop Mod**

**RightToLeft**

**TOS**

**Sorted**

**Standard - 0**

**Style**

**AS**

**HUDA**

الشكل (19): إضافة عناصر إلى القائمة أثناء عملية التصميم

**194**

المصفوفات في لغة فيجوال بيسك 6

**Arrays in Visual Basic 6**



## مثال (12)

سنقوم في هذا التطبيق بالتعامل مع صندوق القائمة من خلال ما يلي:

- إضافة مصفوفة إلى القائمة.

2- حذف عنصر من عناصرها.

عناصر القائمة.

مسح

4- البحث عن عنصر داخل القائمة مع تحديد موقعه إذا كان موجودا، أو

إعطاء

رسالة في حال عدم وجوده، و الخطوات التالية توضح ذلك:

1- صمم النموذج الموضح بالشكل (20)

3-

Formt

**List**

**Create List**

**Search**

**Result**

**Index**

الشكل (20): نموذج يحتوي على أداة عنوان وصندوق نص و زر

أوامر وصندوق قائمة.

انتقل إلى محرر الشيفرة ثم اكتب الشيفرة الموضحة في الشكل (21).

2-

(سعا)

7

**Project1 - Formé (Code)**

**Command2**

**Click**

**Dim (1 To 5) As Integer**

**(Private Sub Command1\_Click**

**For1 = 1 To 3**

**InputBox("Enter The Element of Array") - (1)**

**List1.AddItem A(1)**

**Next 1**

**End Sub**

**(Private Sub Command2\_Click**

**N = Text1.Text**

**Talse -**

**Location - 0**



**Yor1=1re 3**

**IL N-A (1) Then**

**Y - True**

**- Location**

**Exit Yor**

**End IC**

**Next 1**

**It - True Then**

**"Text2. Text - "Yound**

**rex Text - Location**

**zine**

**Text2. Text - "Not Yound**

**Ind IE**

**End Sub**

**الشكل (21): الشيفرة اللازمة لإضافة مصفوفة إلى صندوق قائمة  
والبحث عن عنصر داخلها ,**

**195**

**المصفوفات في لغة فيجوال بيسك 6**

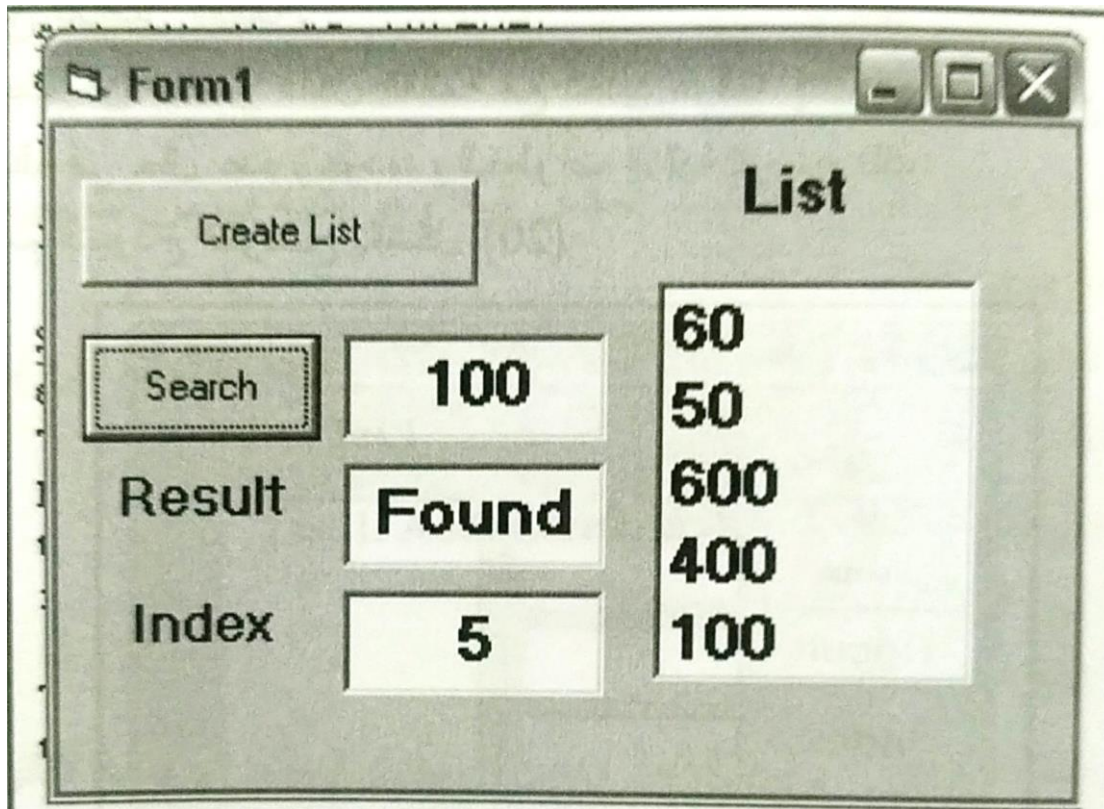
## Arrays in Visual Basic 6

٣\_ نفذ البرنامج ثم انقر على زر الأمر **Create List** ليظهر صندوق إدخال العناصر.

حدد العناصر التالية (٦٠, ٥٠, ٦٠٠, ٤٠٠, ١٠٠) ثم حدد العنصر (100) للبحث

عنه ، ثم انقر على زر الأمر **Search** لتلاحظ ظهور الشكل (22) ثم أدخل القيمة

٩٠ ليتم البحث عنها لتلاحظ ظهور الشكل (23)



The screenshot shows a Windows form titled "Form1" with a search interface. The form contains the following elements:

- A button labeled "Create List".
- A button labeled "Search".
- Three input fields:
  - The first field is labeled "Result" and contains the text "Found".
  - The second field is labeled "Index" and contains the number "5".
  - The third field is labeled "List" and contains a list of numbers: 60, 50, 600, 400, 100.

الشكل (2٢): ناتج التنفيذ بعد إضافة عناصر المصفوفة إلى القائمة

والبحث عن القيمة 100

Form1

Create List

Search

Result

Index

Value: 90

Not Found

List

- 60
- 50
- 600
- 400
- 100

الشكل (23): ناتج التنفيذ بعد البحث عن القيمة (90)

196

المصفوفات في لغة فيجول بيسك ٦

Atrays is Vases Basie

### 13. مسرد المصطلحات

- مصفوفة البعد الواحد (One Dimensional): مجموعة العناصر المتتالية التي تقع

على سطر واحد أو عمود واحد.

- مصفوفة البعدين (Two Dimensional): هي مصفوفة تتكون من مجموعة

صفوف ومجموعة أعمدة، وتحتاج إلى فهرسين للتعامل مع عناصرها،  
أما

المصفوفات متعددة الأبعاد (Multi Dimension)، فهي تتكون من  
أكثر من

بعدين وتحتاج إلى فهرس بعدد الأبعاد.

- مصفوفة متعددة الأبعاد (Multi Dimensional Array): هي  
مصفوفة تتكون من

أكثر من بعدين، وتحتاج إلى فهرس بعدد الأبعاد.

: المصفوفة الديناميكية (Dynamic Array): هي مصفوفة يمكن زيادة  
حجمها أو

إنقاصه بعد الإعلان عنها، أو إذا كنا لا نعرف الحجم المناسب لها.

## المصفوفات في لغة فيجول بيسك ٦

## Arrays in Visual Basic 6

# انتهت الوحدة

2. البيانات وأنواعها في لغة VB6

تتعامل لغة VB6 مع أنواع مختلفة من البيانات، وتلك مثل غيرها من لمسات

البرمجة، وتنقسم البيانات وفق لغة V136 إلى عدة أنواع كما يلي:

(1) البيانات العددية: تتعامل لغة VB6 مع أنواع مختلفة من البيانات العددية، نوجزها

فيما يلي:

(1) البيانات العددية الصحيحة (Integer): وتشمل الأعداد الصحيحة الموجبة والسالبة

والصفر، وتعتبر عنها عادة مجموعة الأعداد الصحيحة

{ ..., -3, -2, -1, 0, 1, 2, 3, ... }

ويخزن العدد منها في الذاكرة في 2 بايت، وبالتالي تنحصر قيم هذه الأعداد

التي يمكن للغة VB6 التعامل معها ما بين -35768 حدا أدنى، وحتى +35767 حدا

اعلى

(ب) البيانات من نوع بايت (Byte): و هي جزء من الأعداد الصحيحة الموجبة فقط،

ويخصص لها بايت لتخزينها، وعليه فتتراوح قيمها ما بين صفر وحتى 255.

(ج) البيانات العددية الصحيحة الكبيرة (Long): وهي تلك الأعداد الصحيحة

الموجبة أو السالبة أو الصفر، التي تتراوح قيمها من 2,147,483,648 إلى

+2,147,483,648 وتخزن في 4 بايت في الذاكرة

(د) البيانات العددية العشرية (Decimal):

وهي بيانات عددية موجبة أو سالبة، صحيحة أو عشرية، يخصص لها 4 بايت

التخزين، ويمكن التمييز بين نوعين منها:

(1) البيانات العددية العشرية ذات الدقة الأحادية (Single): وهي بيانات عددية موجبة

أو سالبة، صحيحة أو عشرية، يخصص لها 4 بايت للتخزين، وتوصف بأنها  
بيانات عددية ذات دقة أحادية معيارية (Single Precision)، أمامدى هذه  
البيانات فهو كما يلي:

مدى البيانات السالبة: من E383.402823- إلى 45  
401298E

-1

مدى البيانات الموجبة : من 45 - E1.401208 إلى E383.402823  
حيث Exx تعنى

(2) البيانات العددية العشرية ذات الدقة المضاعفة (Double): وهي بيانات عددية  
موجبة لرسالية، محيحة أو عشرية، يخصص 8 بايت لتخزين الواحد منها  
وتوصف بأنها بيانات عددية ذات دقة مضاعفة (Double Precision)، أما مدى

10.

51

هذه البيانات فهو كما يلي:

مدى البيانات المالية: من E30879769313486231 إلى  
1.

إلى E94065645841247-324

4

: مدى البيانات الموجبة: من E44.94065645841247-324

إلى E3081.79769313486232+.

حيث Exx تعني "10ء

(2) بيانات السلاسل الرمزية (String)

وهي تستخدم أرقاما (Numerals) وحروفا (Letters) وبعض الإشارات

الخاصة (Special Characters)، ويمكن تسمية أي منها باسم رمز (Character)،

ويخزن في ابايت، وتسمح لغة VB6 بالتعامل مع سلاسل رمزية ثابتة الطول.

(Fixed Length) وتعطى نوع (String)، ويتراوح عدد رموزها من 1 إلى 65000،

أو بنوع متغير الطول، ويعطى نوع (Dynamic String)، ويتراوح عدد رموز هذا

النوع من 1 إلى 2 بليون رمز، وتبدو هذه الأطوال كبيرة جدا كما نلاحظ، إلا أن ذلك مرونة توفرها اللغة للمبرمجين ليستخدموا ما يرونه مناسباً لأغراض برامجهم ومشاريعهم.

(3) بيانات التاريخ (Date):

مع تاريخ الأيام والأشهر والسنوات، ويخصص 8 بايت

التخزين الواحد منها، ويكون مداها من 1 - 100

January إلى 31

December

9999، وعند ذكر تاريخ معين، تشترط - لغة VB6 وضعه بين إشارتين من النوع #

مثل #1950, 28

. #October

(4) بيانات كينونية (Object):

تستخدم للتعامل مع كينونات تشكل مرجعية لعناصر أخرى مثل الصورة

ادارت تحكم، وما إلى ذلك، ويخصص 8 بايت لتخزين الواحد منها.

(5) بيانات متنوعة (Variant):

وتستخدم عندما يكون المبرمج غير محدد بوضوح لنوع البيانات التي يريده

وحيث يمكن أن يسمح هذا النوع بأي نوع من أنواع البيانات التي ذكرنا أعلاه

ويخصص 16 بايت لتخزين الواحد منها، ولذا ينصح بعدم اللجوء إلى استخدام هذا

النوع إلا عند الضرورة، بسبب ضخامة سعة التخزين التي يحتاجها. يذكر أنه إذا لم

يعلن عن النوع في برامجنا، فإن لغة VB6 تعتبره من نوع Variant..

تستخدم

للتعامل

52

بعد استعراض أنواع البيانات في لغة VB6 ، فمن المفيد أن تعرف عزيزي

الدارس، بأن لغة VB6 تفسح لك استخدام هذه الأنواع من البيانات لتحديد قيم ثابتة

عدية تحتاجها في برامجك مثل:

201, -25, 1.495, -2.058

أو قيم ثابتة رمزية، وهي كل ما تضعه بين إشارتي اقتباس من النوع الثنائي"،



(0)

### 3 المتغيرات (Variables)

المتغير في لغة البرمجة يعني اسما مما تسمح به اللغة، ويرادف موقعا محددا  
الذاكرة، يسمح بتخزين بيانات من أي نوع مما ذكرنا أعلاه، والقيمة المخزنة في  
موقع متغير ما قابلة للتعديل والتغيير وفقا للحسابات التي تحدث نتيجة تنفيذ جمل  
البرنامج، وهنا نذكرك عزيزي الدارس بالنقاط الهامة التالية:  
يجب أن يبدأ اسم المتغير بحرف، فمثلا :

L210

,

W50

,

X

,

sum : مقبولة

sun

,

B10

.

A

,

5W: مرفوضة

ولا يجوز أن يتخلل اسم المتغير أي فراغ أو نقطة، فمثلا

SUM.W.5

,

ABC

,

L: مرفوضة.

ب) هناك أحرف إذا بدأت بها أسماء المتغيرات، فإنها تساعد وتشير إلى أن اسم المتغير له مدلول محدد شائع الاستخدام، فمثلا لو بدأت المتغيرات التي ستخزن قيمة من نوع (Double) بالحرف D، كان نقول DSUM، أو بدأت المتغيرات التي ستخزن قيمة من نوع (Long) بالحرف L كأن نقول LValue، أو بدأت المتغيرات التي ستخزن قيمة منطقية (Boolean) بالحرف B، كان نقول BAnswer، لكن هذا كله يؤدي إلى وضوح كبير واتفاق مريح على محلول الأسماء. وهناك مجموعة أخرى شائعة من هذه البدايات نلخص بعضها في

الجدول رقم (1):

الجدول رقم (1): مدلول بعض بدايات اسماء المتغيرات

البداية (مرتبة هجائيا)

المدلول

AMyArray المتغير من نوع Array

A

ANI

B

)

CHIK

Animation Button je wall ANILogo

BAnswer

Currency

Picture Clip FullCLIPMy Picture

Controle will CTRCompare

Boolean من نوع

( IrTency المتغير من نوع CPnnicipal

^Check Bo للتعبير عن CHINAge

( MIDE xil | للتعبير عرا Button )

(للتعبير )

CLIP

CMD

CTR

54

D

DValue

DIR

DLG

DT

FRM

المتضر من نوع Dille

Data FirstValue للتعبير عن Data

Data Base ع للتعبير | DI3Student Rscord D}}

Directory عن للتعبير | DIRMyProgram

Dialog عن للتعبير | DL GEnter

DateTime عن للتعبير | DTMyDate

Form عن للتعبير | FRMModule

Global المتغير يعتبر GSum

Integer من نوع المتغير

Long من نوع المتغير

Label عن للتعبير | LBL MyLabel

Single من نوع المتغير

String من نوع المتغير

|

IGrade

L

LAmount

LBL

S

SInterest

STRItem Number STR

ج) تسمح لغة VB6 باستخدام رموز خاصة في نهاية أسماء المتغيرات ليكون لها

مدلول خاص، وذلك كما يوضح الجدول رقم (2):

الجدول رقم (2): مدلول بعض نهايات أسماء المتغيرات

النهاية

%

منال

المدلول

Amounta | المتغير من نوع Currency

Age | المتغير من نوع Integer

Salary | المتغير من نوع Long

Grade! | المتغير من نوع Single

Value | المتغير من نوع Double

Namics | المتغير من نوع String

%

&

!

!

\$

55

4 جمل الإعلان في لغة VB6

يتم الإعلان عن أنواع المتغيرات بأكثر من جملة إعلان، وذلك كما يلي، مع

ملاحظة أن ما هو موضوع داخل الأقواس الكبيرة [] هو اختياري إذا وجد ما يحدد

النوع بدونه؛

(1) جملة الإعلان

ومن الأمثلة على ذلك:

Dim VarName [As Vartype]

Dim SUM As Double

Dim Age As Single

Dim K As Integer

Dim Citys

@Dim Money

(ب) ومن الجمل الأخرى للإعلان عن المتغيرات:

Private VarName [As VarType]

Static VarName [As VarType]

Public VarName [As VarType]

يجدر بالذكر أن هناك كلمات محجوزة في لغة VB6 ، لتؤدي معنى أو غرضا

محددة، والكلمات المحجوزة التي وردت حتى الآن هي: Static

,

Private

,

Dim

As,

,

Public، حيث كلمة As تعني "مثل"، لتذكر بعدها نوع المتغير، ونوضح معنى

الكلمات Public Static Private

,

Dim في الجدول رقم (3):

Dim

الجدول رقم (3): مدلول بعض الكلمات المفتاحية للإعلان عن المتغيرات.

الكلمة

المعنى

تستخدم للإعلان عن متغيرات، وذلك إما ضمن إجراء أو دالة، ويعتبر المتغير في

الحالة محليا (Local)، ولا يستخدم إلا داخل الإجراء أو الدالة المعلن عنه فيها، ولذا قد يتكرر اسم متغير مثل \* في أكثر من إجراء أو دالة دون حدوث تناقض. أو يتم الإعلان عن المتغيرات بهذه الكلمة ضمن منطقة الإعلان العام ( General Declaration)، ويأخذ المتغير في هذه الحالة صفة (Module Scope) و (Private Scope)، حيث تتم رؤيته واستخدامه من جميع أجزاء ال Module، وهذا النوع يحتفظ بقيمته عند الخروج من الإجراء، بعكس النوع الأول الذي لا يحقه بالقيمة التي اكتسبها ضمن الإجراء.

لا يعامل المنصر المعلن عنه بهذه الكلمة معاملة المنفير الذي له صفة (le Scope) أو (Private Scope) كما ذكرنا في Dim أعلاه.

يتم اجراء تعديل لغوي بسيط (محلى Local، الجزء الآخر من هذا المشروع) ان عن متغير عمومي تتم راته واستخدامه من جميع

أجزاء

اء المشروع

56

أساسيات لغة فيجوال بينه

mentals of Visual Basic 6

الورد الآن بعض الأمثلة التوضيحية عن الاعلان عن المتغيرات، ونعلق عليها

%

التعريف

abc اسم Integer | تعريف متغير محلى من نوع Integer اسم abc

Dim abc تعريف متغير محلى من نوع Integer اسم abc

Public x As Double | تعريف متغير عمومي من نوع Double اسم x

Private y As Long | تعريف متغير محلى من نوع Long اسم y

تعريف متغير يحتفظ بقيمته من نوع String اسم my name

Private mynames | تعريف متغير محلى من نوع String اسم myname

y

myname As String

Static

هناك فكرة جديدة بالذكر ، تتعلق بالإعلان عن المتغيرات، حيث بالإمكان ترتيب بعض الخيارات لتجعل نظام VB6 يسألك عن نوع متغير ما تستخدمه وأنت تبرمج، حتى لو لم تقم في الأساس بعملية الإعلان عنه، وتقوم بعد ظهور السؤال عن النوع بتحديد النوع الذي يلزم للمتغير. لتنفيذ هذا الترتيب اتبع الخطوات التالية:

1. اختر Options من قائمة Tool Menu، فيظهر الشكل رقم (1)

x

Options

Editor Editor Format General Docking Environment Advanced

Code Settings

Auto Indent ✓

Auto Syntax Chess

Require Variable Declaration

Tab Width:

Auto Use Members

Auto Qua Info

Auto Data To

Window Settings

Drag and Drop Text Editing

Default to Redundant View

procedure Separator

Help

Cancel

الشكل رقم (1): صندوق الخيارات (Options) من قائمة Tool Menu

من قائمة الإعدادات في الشكل، اختر Require Variable Declaration

. انقر Ok.

57

اساسيات لغة الفيجوال بيسك 6

## Fundamentals of Visual Basic 6

ومن النصائح الأخرى المفيدة لتسمية المتغيرات، نذكر ما يلي:

(1) عند تعريف متغير فإنك تحجز له مساحة من ذاكرة الحاسوب، فإذا كان لديك متغير لن تستخدمه إلا في جزء بسيط من البرنامج فلا تقم بتعريفه Public حتى لا يحجز المساحة المخصصة له طيلة فترة تشغيل البرنامج، وإنما قم بتعريفه اسطة Dim أو Private حتى يحجز المساحة المخصصة له لأقل فترة زمنية ممكنة

(ب) قم بتسمية المتغيرات بمسميات ذات دلالة على معناها الوظيفي، فذلك يساعدك على تذكر دور المتغيرات في برنامجك، كما أنه يساعد أي شخص لم يكتب البرنامج معك على تتبع البرنامج وفهم معناه بشكل أسهل، أي أنه ينصح باستخدام أسماء للمتغيرات ذات دلالة على ما سيخزن فيها، مثل age للعمر، sum للمجموع، length للطول، رغم أنه يجوز لك غير ذلك طبعاً، كان تستخدم X للمجموع، أو تستخدم W50 للطول مثلاً. كما يمكننا لغة VB6 من الإعلان عن الثوابت، وذلك باستخدام الكلمة المفتاحية Const، التي هي اختصار بطبيعة الحال لكلمة Constant، وذلك كما يلي:

Const Constname1 = Value 1

Constname2 = Value2

Constname k = Valuek

والثوابت 1

Constname 2

،

Constname، ..... المعلن عنها بهذه الطريقة

لا تقبل تغير القيم Value2

،

Value1، ..... التي خزنت في كل منها، كما أن قيمتها

هذه قابلة للاستخدام في أي مكان من أجزاء المشروع. وتسمح اللغة كذلك بنوع آخر

يحدد نوع الثابت كما يلي:

Const Constname [As Const Type] = Value



مع ملاحظة أنه إذا استخدمنا النهايات الموضحة في جدول رقم (2) أعلاه،  
فيصبح استخدام As Const Type لا لزوم له. ويجدر بالذكر عزيزي الدارس أن لغة  
VB6 تشتمل على ثوابت معرفة تلقائياً، وتبدأ كل هذه الثوابت بالبداية vb، ويمكن  
استخدامها و استخدامها قيمها في برنامجك دون الإعلان عنها، والجدول رقم (4) يوضح  
هذه الثوابت والقيم المخزنة بها تلقائياً:

58

اساسيات لغة فول سد

Fundamentals of Visual Basic 6

vbBack

vbBlue

الجدول رقم (4): ثوابت معن عنها وعن أيمها تلقائياً

الثابت

قيمه أو دوره

ثابت يدل على كبسة Back Space

vbNewLine ثابت يدل على النزول لسطر جديد عند الطباعة

vbTab / ثابت يدل على كبسة Tab

ثابت يدل على اللون الأزرق

vbCancel ثابت يدل على أن كبسة Cancel في نموذج معين ثم الضغط عليها

vbEOF ثابت يدل على الوصول لنهاية ملف

vbKey4 ثابت يدل على كبسة الرقم 4

vbKeyB ثابت يدل على كبسة الحرف B

vbKeyF1 ثابت يدل على كبسة f1

vbKeyLeft ثابت يدل على كبسة السهم الأيسر

ثابت يدل على أنه لا يجوز الانتقال من النموذج الأول إلى نموذج

آخر إلا بعد الخروج من النموذج الأول

vbModal

وتقوم لغة VB6 تلقائياً كذلك بالاعتناء بعملية التوفيق بين الأنواع المتباينة في

هذه الحالة

أساسيات لغة فيها

## 5 العمليات والتعبيرات الحسابية والمنطقية

نستخدم في برامجنا نوعين من العمليات الأساسية، هما العمليات الحسابية

(Arithmetic Operations) والعمليات المنطقية (Logical Operations)، ولكل

عملية رمز يدل عليها ويؤدي العمل المتوقع من العملية، ونوضح في الجدول رقم (5)

العمليات الحسابية ورموزها، كما نوضح في الجدول رقم (6) العمليات المنطقية

ورموزها.

الجدول رقم (5): التعريف بالعمليات الحسابية ورموزها وأولوياتها فيما بينها.

العمليات الحسابية رموزها

الألوية

الأقواس

الرفع للأسس

()

2

٨

3

X

الضرب

3

4

القسمة الصحيحة

Mod

١

باقي القسمة

+

الجمع

الطرح

الجدول رقم (6): التعريف بالعمليات المنطقية ورموزها، حيث الأولوية للأقواس ان وجدت ثم من اليسار إلى اليمين.

العملية المنطقية

رمزها

يساوي

لا يساوي

أقل من

أقل من أو يساوي

أكبر من

أكبر من أو يساوي

60

of Visual Basic 6

أ. أما التعبير الحسابي، فهو عبارة عن سلسلة من المتغيرات والثوابت العددية

تفصل بين أي اثنين منها عملية حسابية.

ومن الأمثلة على التعبيرات الحسابية:

$$W^2 + 5/n * m^{\circ}$$

$$Y5Y-20$$

$$b3 + ((3y-20) - 4*4)$$

$$\text{num} / (2y + 8)$$

ولتوضيح أولوية أجراء العمليات الحسابية، نتابع معا إجراء العمليات حسب

الأولوية في التعبير قبل الأخير المذكور أعلاه، كما يلي:

)

$$4d$$

$$y - 20)) + b*3$$

.3)

1

2

6

لاحظ كيف تم إعطاء الأولوية للأقواس الداخلية، ثم الأقواس التي تليها،

وهكذا.

التعبير المنطقي ينتج عن إجراء مقارنة بين عنصرين قد يكونا من المتغيرات المعروفة القيم أو ثوابت، أو خليطا منهما، والتعبير المنطقي، يكون جوابه إما True أو False، وذلك بعد إتمام عمليات المقارنة المنطقية استنادا إلى قيم مكونات التعبير، ولنضرب مثالا على ذلك، فلنفرض ما يلي:

$$X = 10, Y = 12, z = -3, W = 2.5$$

وبناء على هذه القيم، نورد التعبيرات المنطقية التالية مع قيمة كل منها في

الجدول رقم (7)

حدد أولوية إجراء العمليات في التعبير الحسابي التالي

$$* A + B \bullet (C \text{ Moul } D - (X + Y / W))$$

61

اساسيات لغة فيجوال بيسك 6

Fundamentals of Visual Basic 6

الجدول رقم (7): بعض التعبيرات المنطقية وقيمها، حيث  $W = 2.5$  ,

(10,112, 25

It's Value

The Expression

True

X=25

True

XZ

True

X457|

True

$$Y \ 274 * W = 0$$

False

$$X-3Y/9 - 16+2$$

True

$$2N3Y9 = 6$$

$$X-3*Y / 9 = 16 \text{ True}+2$$

$$2X3Y9 = 16 \text{ True}|$$

لاحظ أنه في حالة اجتماع عمليات حسابية، وأخرى منطقية في نفس التعبير، فإن الأولوية تعطى لإجراء العمليات الحسابية قبل المنطقية، ويجدر بالذكر أن جميع التعبيرات المذكورة في الجدول رقم (7) تسمى تعبيرات منطقية بسيطة (Simple) كونها تشمل مقارنة منطقية واحدة. ويمكننا كذلك التعامل مع تعبيرات منطقية مركبة (Compound)، حيث يتكون التعبير المنطقي المركب من تعبيرين بسيطين أو أكثر يربط بين كل اثنين أداة ربط منطقية (Connector)، الجدول رقم (8) يوضح أدوات الربط في VB6، ورموزها وأولوياتها.

الجدول رقم (8): التعريف بأدوات الربط بين التعبيرات المنطقية ورموزها وأولوياتها فيما بينها.

أداة الربط المنطقية

رمزها

الأولوية

النفي

NOT

AND

2

3

OR

أو

أو التخالفية

XOR

3

أما الجدول رقم (9)، فيمثل قيم الصواب والخطأ عند استخدام أدوات الربط

المختلفة، حيث T تعني True و F تعني False، علماً أن كلا من Q

،

P ترمز إلى

تعبير منطقي بسيط، كما نقدم في الجدول رقم (10) بعض الأمثلة على التعبيرات المنطقية المركبة وقيمها.

الجدول رقم (9): جدول الصواب والخطأ لبعض أدوات الربط المنطقية

NOT

P

F

F

T

T

POR O

T

T

T

F

PANDO O

T

F

F

T

F

F

P

T

T

F

F

62

اساسيات لغة فيجوال بيسك (

Fundamentals or Visual Basic 6

الجدول رقم (10): بعض الأمثلة على التعبيرات المنطقية المركبة وقيمها، حيث

Y 12, 25, W=2.5 ,10

It's Value

The Expression

T

((X<= 25) AND X > Z

T

((X<=25) OR X >Z

T

(X = 25) IMP (X >Z)

F

(X> = Z) IMP (X<= 25)

T

(X<= 25) EQV (X+5> Z)

F

(X<= 25)XOR (X> = Z)

T

(X<= Z) XOR (X+5> Z)

ومن العمليات الأخرى في لغة VB6 ، عملية تستخدم لوصل السلاسل الرمزية

(Concatenation)، ورمزها &، حيث تستخدم كعملية ثنائية بين سلسلتين لتنتج منهما

سلسلة رمزية واحدة، وبالطبع يمكن تكرار هذه العملية لوصل عدة سلاسل، شريطة

وجود عملية بين أي سلسلتين منها، كما يمكن أن يكون مكان سلسلة ما في التعبير

الرمزي ثابتا رمزية، تماما كما يحدث في التعبير الحسابي، هناك مجموعة من الدوال

(Functions) الرمزية المستخدمة في هذا المجال، والتي تعيد (Retur) قيمة رمزية

وبالتالي يمكن استخدامها في التعبيرات الرمزية.

وحتى تكتمل الصورة، وترتبط أفكار العمليات المختلفة من حيث الأولوية

بعضها مع بعض، نورد في الجدول رقم (1) تعريفا بأولوية العمليات إذا اجتمعت

في تعبير مركب.

الجدول رقم (11): أولويات العمليات إذا اجتمعت في تعبير مركب، علما أن الأولوية

الأولى للأقواس الداخلية ثم التي تليها، وهكذا

العملية

أولوياتها

رمزها

1

الأقواس

0

2

3

الرفع للاس

الإشارة السالبة

الضرب والقسمة

القسمة الصحيحة

4

5

\

MOD

6

باقي القسمة

7

8

&



الجمع والطرح

الإلحاق الرمزي

المقارنات

9

NOT

10

النفي

AND

11

از

12

OR, XOR

او، او التخالفية

63

اساسيات لغة فيجوال بيسك 6

Fundamentals of Visual Basic 6

6. جملة التعيين أو التخصيص (Assignment Statement)

ان الصورة العامة لهذه الجملة هي كما يلي:

Variable Name = Expression

حيث الطرف الأيسر منها هو متغير، ولا يجوز غير ذلك، والطرف الأيمن هو

تعبير، ويمكن أن يكون حسابية أو منطقية أو ثابتة رمزية أو استدعاء أو استخدام الدالة

فرعية تعتبر قيمة محددة، شريطة أن يتوافق مع أنواع المتغير في الطرف الأيسر.

وفصل بين الطرفين رمز التخصيص =، وفيما يلي أمثلة على هذه الجملة:

$X = 2 * Y + 3 * 7 - 25$

$SUM = SUM + a * b$

"Name \$ = "Palestine

K% = 100

ومفهوم هذه الجملة هو أن يتم إيجاد قيمة التعبير في الطرف الأيمن، وذلك بعد إجراء العمليات إن وجد، ثم يتم تخصيص أو تعيين هذه القيمة للمتغير في الطرف الأيسر.

#### 7. جملة الملاحظة Comment Statement

من المهارات التي ينصح بها في البرمجة تزويد البرنامج بما يلزمه من ملاحظات عن هدف البرنامج، وتاريخ كتابته، ومن كتبه، وتعريف بمتغيرات البرنامج وأجزائه، حيث يفيد ذلك كله عند عودة المبرمج بعد فترة لمراجعة برنامجه، فيتمكن من فهمه بسرعة، كما يفيد الآخرين الذين يطلعون على البرنامج، ويتم ذلك في لغة VB6 بوضع إشارة اقتباس منفردة في بداية كتابة الملاحظة على يسارها، لكتابة تعريف عنها، وهذه الجملة غير تنفيذية (Not Executable) ولا يقوم مترجم اللغة بترجمتها إلى لغة الآلة. وهذه أمثلة على جملة الملاحظة:

\* This Is Our First Program.

\* This Program is to SUM two Values.

Program is: Ahmad Salim.

\* This Function is ti sort an array.

\* .Date: 1/1/2006.

#### 8. النافذة الفورية Immediate Window

عند تشغيل لغة VB6 واختيار Standard.EXE، تظهر نافذة في اسفل الشاشة معنونة Immediate، وإن لم تظهر، فيمكنك إظهارها، وذلك باختيارها من قائمة View، أو بالضغط على مفتاحي Ctrl و G معا. تستخدم هذه النافذة الرؤية نتائج فورية الجملة بلغة VB6 تكتبها في هذه النافذة وتضغط Enter بعد كتابتها، حتى ترى في النافذة نتيجة تنفيذ هذه الجملة، و الجدول رقم (12) يشمل أمثلة توضيحية على ذلك:

الجدول رقم (12)

Result

VB is nice

Press Statement

"Enter Print "VB is nice

"Enter MsgBox "Good Luck

18

Enter X = 10

Y=8

Z=X+Y

Print Z

false

Enter A = 3

Y = 4

Print A>Y

6

Enter Print 12.4/2

14.4

Enter Print 12.4+2

True

Enter Print 12.4>2

False

Enter | Print 12.4 <= 2

"My Name Is Sadeq Enter Print "My Name" & "Is Sadeq

True

) Enter | Print (12.4 <2) Or

(52

True

Enter Print True And (12.4 > 2)

9. الطباعة وتهييننما (Print and Format)

إذا أردت أن تعد تقارير برنامجك باستخدام الطابعة (Printer)، فبإمكانك

استخدام الصيغة العامة التالية التي توفرها لغة VB6 ،

[ Printer.] Print [ spc(n): Tab (n) ], expression

مع ملاحظة ما يلي:

الجدول رقم (13)

]

[ Printer. جزء اختياري من الجملة (لاحظ الأقواس [ ] )، وهو كائن

(Object) يرمز إلى أننا نطلب النتائج أو التقارير على الطباعة.

Spc]

)

n

(

:

[Tabcn وهذا أيضا جزء اختياري من الجملة، ويعني: نريد استخدام

الدالة الجاهزة Spc لترك n من الفراغات أو (هذا ما تعنيه)

نريد استخدام الدالة الجاهزة Tab لتحديد أننا نريد بدء الطباعة

من العمود رقم 1، حيث n قيمة عددية صحيحة موجبة.

يرمز إلى التعبير الذي نريد طباعته أو طباعة قيمته أو نتيجته،

لاحظ استخدام أو بين Spc أو Tab وبين رمز التعبير، حيث

يمكن استخدام أو بدلا منها كذلك.

Print | كلمة محجوزة الإصدار أمر الطباعة.

Experssion

( ) Private Sub Command 1\_Clickit

X = 12

Y=3

Printer. Print Tab(5); "X="; X

Printer. Print Tab(5); "Y=": Y

Printer. Print Tab(5); "Sum="; X+Y

Printer. Print Tab(5); "Product="; XY

Print Tab(5): "Quotient=" X/Y

Print Tab (5); "X Mod Y =" : X Mod Y

End Sub

وننصح الدارس الكريم أن ينفذ هذا البرنامج، ويدقق النتائج التي يطبعها بعد تنفيذ هذا البرنامج، يجدر بنا توضيح عدة نقاط حول استخدام جملة الطباعة:

1. إن استخدام الفاصلة المنقوطة : بين الأجزاء المطلوب طباعتها يؤدي إلى ترك الفراغات، علما أن الفراغ الذي يشاهد قبل القيمة العددية المطبوعة، ليس فراغا في الحقيقة، ولكنه محجوز لإشارة القيمة، إما \* وإما -

67

أساسات لذة لجرال سه را

## Fundamentals of Visual Basic 6

2. أما استخدام الفاصلة العادية , بين الأجزاء، فيؤدي إلى تقسيم سطر الطباعة إلى مناطق (Zones) محل كل منها مكون من 14 - 16 فراغا في العادة، وتخصص كل منطقة الجزء من الأجزاء المطلوب طباعتها.

3. الأصل أن تنفذ كل جملة طباعة بسطر منفصل، إما إذا وضعت ، أو في آخر جملة الطباعة، فهذا يؤدي إلى تنفيذ الطباعة التالية في نفس السطر، ولا يتم الانتقال إلى سطر جديد.

4. الجملة Print تطبع سطرا فارغا، وإذا كان مكانها بعد جملة Print منتهية بالرمز، أو الرمز ; فإنها تلغي البقاء في نفس السطر وتنقل الطباعة التالية إلى سطر جديد.

5. إن أي جزء يوضع بين إشارتي الاقتباس " " يعتبر ثابتا رمزيا، ويطبع كما هو تماما.

إضافة إلى ذلك، فإن لغة VB6 توفر العديد من الإمكانيات المتقدمة للتحكم بالمادة المطبوعة، وفيما يلي توضيح لأهمها:

1. هناك مجموعة من الكلمات المحجوزة التي تستخدم قبل جملة طباعة ما، وتؤدي الغرض الموضح إزاء كل منها في الجدول:

الكلمة

الجدول رقم (14): الكلمات المستخدمة قبل جملة Print ومعانيها.

صيغة استخدامها

الغرض من استخدامها

التوجيه الطباعة التالية إلى بداية صفحة جديدة

أو

New Page

New Page

Printer. New Page

Print Quality

Orientation

تحدد جودة الطباعة، وتستخدم القيم من 1 إلى 4

لتحديد الجودة، حيث 1 الأقل جودة، 4 الأكثر جودة

للحصول على طباعة إما أفقية أو عمودية، حيث:

. تحدد الطباعة عموديا.

2. تحدد الطباعة أفقيا

تحدد عدد النسخ المطلوب طباعتها

تحدد نوع الطباعة، حيث:

1. تعني ابيض واسود

Copies

Color Mode

2. تعني ملونة

Font Bold

Font Bold: True

تظهر الطباعة التالية بخط غامق.

تظهر الطباعة التالية بخط غير غامق

تظهر الطباعة التالية بخط مائل.

تظهر الطباعة التالية بخط غير مائل

Font Bold: False

Italic = True

Italic = False

Font Italic

Font Underline

68

اساسيات لغة فيجوال بيسك 6

Fundamentals of Visual Basic 6

2. استخدام جملة التهيئة Format:

إن الصيغة العامة لهذه الجملة هي:

حيث StrFormat يمكن أن يكون إحدى الكلمات "General Number"،

Scientific"

"

،

"

Percent

"

،

"

Fixed

"

،

"

Currency"، ولكل منها معنى محدد.

Format (Expression, StrFormat)

""

، . " "" ""

نشاط (1)

ابحث في المعنى المحدد لكل من الكلمات المذكورة أعلاه.

## 10. الخاصة

### الغة VB6 في

لقد خصصنا هذه الوحدة، عزيزي الدارس، لمناقشة أساسيات لغة VB6، حيث تكونت الوحدة من ثمانية أقسام رئيسية، تم تخصيص القسم الأول منها لمناقشة البيانات وأنواعها في لغة VB6، ثم عرضنا أساسيات التعامل المتغيرات في القسم الثاني. أما في القسم الثالث فقد ناقشنا فيه كيفية الإعلان عن أي متغير من خلال التعرف إلى الصيغة العامة لجمل الإعلان وبعد استعراض معلومات البيانات والمتغيرات وجمل الإعلان، أصبح ممكناً أن نجري عمليات حسابية ومنطقية فيما بينها، ونكون منها تعبيرات حسابية أو منطقية، وهذا ما تم مناقشته في القسم الرابع من الوحدة. أما الأقسام الخامس والسادس، فقد ناقشنا من خلالها جمل التخصيص والملاحظة مع التوالي، ثم ناقشنا استخدام النافذة الفورية لإظهار نتائج جمل بمجرد كتابتها والضغط على Enter، وذلك في القسم السابع، وخصصنا القسم الثامن لتقديم أفكار عن طباعة المشروع ونتائجه وجمل التهيئة ذات العلاقة.

### 11. لمحة عن الوحدة الدراسية الرابعة

تواصل في هذه الوحدة - عزيزي الدارس - عملية تعلم وفهم المزيد من قواعد التعامل مع لغة VB6، حيث تخصص الوحدة لمناقشة تراكيب التحكم في اللغة ليتم توظيف المعلومات المتراكمة لحل مسائل من خلال البرمجة بلغة VB6، لذلك استعرض أفكاراً من الخوارزميات و خرائط سير العمل التي تشكل قاعدة هامة لحل المسائل، ثم ناقش تراكيب التسلل والاختيار والتكرار في لغة V136، مما يمكننا من كناية مشاريع وبرامج متنوعة بهذه اللغة.

69

انسانيت لفة فيجوال بيسك 6

Fundamentals of Visual Basic 6

### 12. إجابات التدريبات

تكرير

$$A+B * ((( \text{Mod } D (X+/W))) / 3$$



## 13. مسرد المصطلحات

بيانات التاريخ Date: تستخدم للتعامل مع تاريخ الأيام والأشهر والسنوات

ويخصص 8 بايت لتخزين الواحد منها، ويكون مداها من 100 - 13

January

إلى 9999\* 31

December، وعند ذكر تاريخ معين، تشترط - لغة VB6

وضعه بين إشارتين من النوع #، مثل #1950, 28,

#October.

بيانات السلاسل الرمزية String: وهي تستخدم أرقاماً (Numerals) و حروفاً

(Letters) وبعض الإشارات الخاصة (Special Characters)، ويمكن تسمية

أي منها باسم رمز (Character)، ويخزن في 1 بايت.

البيانات العددية ذات الدقة الأحادية Single: وهي بيانات عددية موجبة أو سالبة،

صحيحة أو عشرية، يخصص لها 4 بايت للتخزين، وتوصف بأنها بيانات

عددية ذات دقة أحادية (Single Precision).

- البيانات العددية ذات الدقة المضاعفة Double: وهي بيانات عددية موجبة أو

سالبة، صحيحة أو عشرية، يخصص 8 بايت لتخزين الواحد منها، وتوصف

بأنها بيانات عددية ذات دقة مضاعفة (Double Precision).

البيانات العددية الصحيحة Integer: وتشمل الأعداد الصحيحة الموجبة والسالبة

والصفر، وتعتبر عنها عادة مجموعة الأعداد الصحيحة { ..., 1, 0, -1, -2, -3

, 2, 3 } = Z ويخزن العدد منها في الذاكرة في 2 بايت، وبالتالي تنحصر قيم

هذه الأعداد التي يمكن للغة VB6 التعامل معها ما بين -35768 - حداً أدنى،

وحتى +35767 حداً أعلى.

البيانات العددية الصحيحة الكبيرة Long: وهي الأعداد الصحيحة الموجبة أو

السالبة أو الصفر، والتي تتراوح قيمها من 648، 483، 147، -2 إلى 648،

483، 147، +2 وتخزن في 4 بايت في الذاكرة

- البيانات العددية العشرية Decimal: وهي بيانات عددية موجبة أو سالبة، صحيحة

أو عشرية، يخصص لها 4 بايت للتخزين.

- بيانات كينونية Object: تستخدم للتعامل مع كينونات تشكل مرجعية لعناصر

أخرى مثل الصور، أدوات تحكم، وما إلى ذلك، ويخصص 8 بايت لتخزين

الواحد منها.

- بيانات متنوعة Variants: وتستخدم عندما يكون المبرمج غير محدد بوضوح

النوع البيانات التي يريد، وحيث يمكن أن يسمح بأي نوع من أنواع البيانات

التي ذكرنا أعلاه، ويخصص 16 بايت لتخزين الواحد منها ولذا ينصح بعدم

اللجوء إلى استخدام هذا النوع إلا عند الضرورة، بسبب ضخامة سعة التخزين

التي يحتاجها. يذكر أنه إذا لم يعلن عن النوع في برنامجنا، فإن لغة VB6

تعتبره من نوع Variant.

- البيانات من نوع بايت Byte: وهي جزء من الأعداد الصحيحة الموجبة فقط،

ويخصص لها بايت لتخزينها، وعليه تتراوح قيمها ما بين صفر حتى 255.

التعبير الحسابي Arithmetic Expression: هو عبارة عن سلسلة من المتغيرات

والثوابت العددية، تفصل بين أي اثنين منها عملية حسابية.

- التعبير المنطقي Logical Expression: ينتج عن إجراء مقارنة بين عنصرين قد

يكونا من المتغيرات المعروفة القيم أو ثوابت، أو خليط منهما، والتعبير

المنطقي، يكون جوابه إما True أو False، وذلك بعد إتمام عمليات المقارنة

المنطقية استنادا إلى قيم مكونات التعبير.

- الثوابت Constants: الثابت يرادف جزءا من الذاكرة يخزن فيه قيمة ثابتة لا

تتغير.

جملة الإعلان Declaration Statement: جملة تستخدم للإعلان عن المتغيرات

في اللغة.

جملة التعيين Assignment Statement: جملة تستخدم لإعطاء متغير قيمة

معينة، ويستخدم فيها مزيج بين المتغيرات والثوابت والعمليات الحسابية والمنطقية.

جملة الملاحظة Connment Statement : جملة تستخدم لتزويد البرنامج بما يلزمه من ملاحظات عن هدف البرنامج، وتاريخ كتابته، ومن كتبه، وتعريف

71

أساسيات لغة فيجوال بيسك 6

## Fundamentals of Visual Basic 6

بمتغيرات البرنامج واجزائه، حيث يفيد ذلك كله عند عودة المبرمج بعد فترة المراجعة برنامجه، فيتمكن من فهمه بسرعة، كما يفيد الآخرين الذين يطلعون على البرنامج، ويتم ذلك في لغة VB6 بوضع إشارة اقتباس مفردة في بداية كتابة الملاحظة على يسارها، لكتابة تعريف عنها، وهذه الجملة غير تنفيذية (Not Executable) ولا يقوم مترجم اللغة بترجمتها إلى لغة الآلة.

- المتغيرات Variables : المتغير في لغة البرمجة يعني اسما مما تسمح به اللغة ويرادف موقعا محددا في الذاكرة، يسمح بتخزين بيانات من أي نوع مما ذكرنا أعلاه،

والقيمة المخزنة في موقع متغير قابلة للتعديل والتغير وفقا للحسابات التي تحدث نتيجة تنفيذ جمل البرنامج.

النافذة الفورية Immediate Window: تستخدم هذه النافذة لرؤية نتائج فورية الجملة تكتبها في البرنامج وتضغط Enter بعد كتابتها، حتى ترى في النافذة

الفورية نتيجة تنفيذ هذه الجملة.

Dim: تستخدم للإعلان عن متغيرات، وذلك إما ضمن إجراء أو دالة، ويعتبر المتغير في هذه الحالة محلية (Local)، ولا يستخدم إلا داخل الإجراء أو الدالة المعلن عنه فيها، ولذا قد يتكرر اسم متغير مثل x في أكثر من إجراء أو دالة دون حدوث تناقض. أو يتم الإعلان عن المتغيرات بهذه الكلمة ضمن منطقة الإعلان العام (General Declaration)، ويأخذ المتغير في هذه الحالة صفة (Module Scope) أو (Private Scope)، حيث تتم رؤيته واستخدامه من

جميع أجزاء ال Module، وهذا النوع يحتفظ بقيمته عند الخروج من الإجراء، بعكس النوع الأول الذي لا يحتفظ بالقيمة التي اكتسبها ضمن الإجراء.

Private: يعامل المتغير المعلن عنه بهذه الكلمة معاملة المتغير الذي له صفة (Module Scope) أو (Private Scope) كما ذكرنا في Dim أعلاه.

Public : للإعلان عن متغير "عمومي" تتم رؤيته واستخدامه من جميع أجزاء المشروع

Static: يتم إجراء تعديل لغوي بسيط (محلي Local، الجزء الآخر من هذا المشروع).

العمليات Operations: هي إما عمليات حسابية (Arithmetic Operation)، وتتم بين تغييرين أو ثابتين أو خليط منهما، وإما عمليات أو مقارنات منطقية (Logical Operation)، وتتم بين طرفين حسابيين لنحصل على نتيجة صواب أو خطأ

72

أساسيات لغة فيجوال بيسك (

Fundamentals of Visual Basic 6

### String ( الرمزية والسلاسل ) Characters الرموز

باستخدام المستويه عالمية اخري لغة باي أو ، بيسك فيجوال بلغة البرامج تكتبها ذات والثورات والمتغيرات الكلمات من مجموعة من تتكون التي الأوامر من مجموعة أساسية بثورة من والتوابته والمتغيرات الكلمات وتتكون المستخدمة، اللغة في الدولة ومل اللغة تستخدمها التي الرموز من رمز وكل ، Characters الحرف الرمز تسمى رمزا 250 تمثيل نستطيع فإننا وبالتالي ،)بت ثمانية أي ( Byte ( بالبايت يسمى بما ( بين محصورة صحيحة بامداد الرموز هذه تمت يمكن حيث

| هذا ويعرف 2550

( American Standard Code for Information ASCII hi Lill )

. Interchange )

47 القيمة وتستخدم ، " A " الحرف لتمثيل 65 القيمة تستخدم المثال سبيل فعلى وهكذا ، "الرمز التمثيل

( 1 ) الذاتي التقويم أسئلة

والرموز؟ بالسلاسل ASCII نظام علاقة ما

تتكون التي ( Characters ( الرموز من مجموعة فهي الحرفية، السلاسل أما  
( Special ( الخاصة والرموز ) Digits ( والأرقام ) Letters ( الهجائية الحروف من  
:هما الحرفية السلاسل من نوعان وهناك . Characters (

أو أرقام أو حروف عن عبارة وهي : (الرمزية الثوابت ( الثابتة الحرفية السلاسل -  
:ذلك توضح التالية والأمثلة ؛ اقتباس علامتي بين توضع خاصة رموز

" Where Do You Live ?

" String & Characters

7 # \*- 9 + 22

ة عددي ثابتة لا رمزية ثابتا يعتبر اقتباس علامتي بين رقم وضع عند أنه لاحظ

128 # 128

Print : جملة باستخدام الرمزية الثوابت مع التعامل ويمكن

Print \* # # # # # "

265

والصوم والعمل الرموز

Characters, Strings and Text

اختر تستحم ذاكرة مواقع وهي :الرمزية المتغيرات ( المتغيرة الحرفية السلاسل -  
التاليتين الجملتين إحدى باستخدام عنها الإعلان ويمكن الرمزية، الكويت

Din Variable as String

Dust Variables

منا

Dim X As String

Dim YS

X = "String "

Y = "Characters "

Print ""; X, "Y="; Y

.تاليا سنرى كما الحرفية السلاسل على العمليات من كثير إجراء ويمكن

,+ ,& Compare, Like, StringRev , Upper Case, Lower Case

ة الرمزي السلاسل على العمليات 1.2

,

,

يمكن والتي ببسك، فيجوال البرمجة لغة توفرها التي العمليات من الكثير هناك  
: العمليات هذه ومن .الحرفية السلال على إجراؤها

& ، باستخدام الرمزية السلاسل بين الدمج - 1

التالي والمثال ؛& \* الدمج أداتي باستخدام الرمزية السلاسل بين الدمج يمكن  
ذلك؟ توضح

Private Sub Commandi\_Click ) (

stri - "ABC "

str2 = "DEFG "

str3 = "HIJ "

Str4 = stri + str2

str5 = stri & str2 str3

Print str4

Print str5

End Sub

ر يظه حيث ، Str2 و Srl بطول هو str4 الرمزي المتغير طول أن يعني وهذا  
لم عند تماما مشابه & اء من كل وعمل Str طباعة عند Str S ملين على  
كل المختلفة، البيانية الأنواع بين ما استخدام يمكن ولا لرمزية، والقوات المتغيرات  
ه رمزك ومتفقي علي من بل الجمع

266

النصر و والسلاسل الرموز

Characters, Strings and Text

خاطفة جملة

X5

Y="A "

2XY

..الأبجدي الترتيب

الأبجدي ترتيبها حسب الرمزية السلاسل بين المقارنة تتم حيث الرمزية؛ السلاسل  
بين للمقارنة < > = = = ( المقارنة عمليات جميع إجراء ويمكن

AI الشرط فمثلا

"

<

"

حلب " Zaid تسبق ' Ali " لأن وذلك صحيحا، يعتبر Zaid

( Conspare ) الحربية السلاسل بين المقارنة 2

الرمزية السلاسل بين للمقارنة التالية الدالة بيسك فيجوال البرمجة لغة توفر

:التالية الصيغة وفق ) < > = = = ( المقارنة عمليات استخدام من بدلا

هي String2 و الأولى، الحرفية السلسلة هي String1 إن حيث

: التالية القيم الدالة هذه وتعيد الثانية؛

ة الثاني الحرفية السلسلة من أكبر String1 الأولى الحرفية السلسلة كانت إذا :

StrComp(String1, String2)

ة الحرفي المالية

2

. String2

ة الثاني الحرفية السلسلة من أصغر String1 الأولى الحرفية السلسلة كانت إذا :-

2

. String

String2 ة الثاني الحرفية السلسلة تساوي String1 الأولى الحرفية السلسلة كانت إذا :

التالية المثال نطبق الدالة هذه عمل ولتوضيح

( 1 ) تطبيق

مدخلتين رمزيتين سلسلتين بين المقارنة عملية بإجراء التطبيق هذا في سنقوم  
توضح التالية والخطوات عنوان، أداة في النتيجة ويظهر نص، صندوق باستخدام  
ذلك

( 1 ) بالشكل الموضح النموذج صمام -أ

3 Form1

Enter String

Ene Sun.2

Compa

Rent

المصمم النموذج : 1 ( الشكل

والنصوص والمسلسل الرموز

267

Characters, Strings and Text

ركسي ( 2pare ) رامسر على المزوج بالنقر الشيفرة محرز بي الشق -

( 2 ) بالشكل الموضحة الشيفرة

ا ر )

Project1. Form1 (Code)

Commandi

Click

Private Sub Commandi\_Click ) (

Dim stisString

Dim st2 As String

Dit result is Integer

stri - Text1. Text

ste2 - Text2. Text

resultStrComp (stri, str2)

IL Lesult - 1 Then

Labe14. Caption - "STR1 GRATER THAN ,

Elselt result --1 Then

Labe 14. Caption - "STR1 LESS THAN STI

Elselt result = 0 Then

Labe 14. Caption - "STRI EQUAL STR2 "

End If

End Sub

رمزيتين سلسلتين بين المقارنة شيفرة ) 2 ( الشكل

والشكل ( Compare ) ( الأمر زر على انقر ثم ) FS ( على بالنقر السابق البرنامج نفذ - 3

.المقارنة نتيجة يظهر ) 3 (

Form

Enter Stina

CAR

Erla Sun2 CAT

Compare

STRI LESS THAN STA2

المقارنة نتيجة : 3 ( الشكل

والنصوص والعسل الرموز

Characters, Strings and Text

268

النتائج وملاحظة مختلفة رمزية

سلام على السابق التطبيق خلال من المقارنة عمليات من العديد إجراء يمكن

Like المعامل

- 3

ة التالي العامة الصيغة وفق الرمزية السلاسل بين للمقارنة التالية الدالة تستخدم

" String 1" Like "String 2

متشابهتان، الرمزيتان السلسلتان كانت إذا ( True ( القيمة الدالة هذه تعطي حيث

توضح التالية والأمثلة مختلفتين؛ الرمزيتان السلسلتان كانت إذا ( false ( القيمة وتعطي

ذلك:

( 2 ) تطبيق

باستخدام الرمزية السلاسل بين المقارنة عملية بإجراء التطبيق هذا في سنقوم

:التالية الخطوات وفق Like المعامل

(. 4 ) بالشكل الموضح النموذج صمم - 1

1

Form1

Enter Strins1

Enter Strins2

Like

Result

.المصمم النموذج : 4 ( الشكل

الشئ واكتب ( Like ( الأمر زر على المزدوج بالنقر الشيفرة محرر إلى انتقل - 2

(. 5 ) بالشكل الموضحة

والنصوص والسلاسل الرموز

acters, Stri

E Project Form1 (Code)

command

Click

Private Sub Contrand1\_Click )(

) (

Dim sti As String

Dim st2 String

Dim result As Boolean

stri - Text1. Text

str2- Text2. Text

result – stri Like strz

It result Then

Labe 14. Caption "Similar "

Else



#### Labe 14. Caption - "not Similar "

End If

End Sub

like المعامل شيفرة (: 5 ( الشكل

( Text النص مربع داخل القيم أدخل ثم ) F5 ( على بالنقر السابق البرنامج نفذ 3

زرار على انقر ثم الثاني، ) Text Box ( النص مربع داخل رقيم الأول Box )

( 6 ( بالشكل موضح هو كما ) Like ( الأمر

Fort

Enter Sun CAR

Enter Sun CAT

Rou

Like المعامل باستخدام المقارنة نتيجة (: 6 ( الشكل

سمل طى السابق التطبيق خلال من المقارنة عمليات من العديد إجراء يمكن  
النتائج وملاحظة مختلفة رمزية

270

وتمو والمال الرموز

Characters, Strings and Test

الدالة هذه داخل التالية الرموز استخدام يمكن

( ٤ ) الزمر |

؛ المقارنة عملية من تسى الرموز من سلسلة من بدلا الرمز هذا يستخدم  
، توضح التالية والأمثلة

" I Am Student " Like " \* Student

(

" I Am Student " Like "I Am

السلسلتين نهاية بين تتم المقارنة لأن ) True ( القيمة الجملة هذه تعطي

| \* |

السلسلتين بداية بين تتم المقارنة لأن ) True ( القيمة الجملة هذه تعطي

) |

" "

السلسلتين ونهاية بداية بين تتم المقارنة لأن ) True ( القيمة الجملة هذه تعطي

" I Am Student " Like "I \* Student' le

( 4 ( الرمز - 2

ة الرمزي السلسلة داخل معين رمز موقع كان إذا ما الاختبار الرمز هذا يستخدم  
: ذلك توضح التالية والأمثلة حروف؛ وليس رقم الأولى

Visual Basic 6 "Like" Visual Basic ('#

السلسلة في # الرمز المقابل الموقع لأن ) True ( القيمة الجملة هذه تعطي

. 6 رقم هو الأولى

( # L ب

"

السلسلة في # للرمز المقابل الموقع لأن ( False ) القيمة الجملة هذه تعطي  
ه حرف هو الأولى

Visual Basic L "Like" Visual Basic ( "#

? الرمز

الذي الموقع إهمال مع رمزيتين، سلسلتين بين للمقارنة الرمز هذا يستخدم  
توضح التالية والأمثلة الثانية؛ الرمزية السلسلة في (حرف/رقم) ؟ الرمز فيه يوضع  
: ذلك

Computer science" Like "Computer ?cience

من الثانية الكلمة من الأول الحرف لأن ( True ) القيمة الجملة هذه تعطي  
ة عملي أثناء إهماله يمكن (رقما أو حرفا يكون أن يمكن الثانية الرمزية السلسلة

Computer Skills" Like "Computer Skill ( "??

السلسلة من الثانية الكلمة من حرفين آخر لأن ( True ) القيمة الجملة هذه تعطي  
الأحسرت بقية تكون أن ويجب المقارنة، عملية أثناء إهمالهما يمكن الثانية الرمزية  
[ ] الكبيرة الأقواس - 4

ه ب المقارنة المراد الرموز ضمنه يقع الذي المدى لتحديد الأقواس هذه تستخدم  
: ذلك توضح التالية والأمثلة ؛ [ 3 - 7 ]

A [ المدى ضمن C الحرف يقع المثال سبيل فعلى

-

المدى في يقع 5 الرقم أن أو [ N

" Computer Skills" Like "Computer S[i-n]ills ( ')

" Computer Skills" Like "Computer S[a-e]ills ( "

fi [ المدى ضمن يقع k الحرف لأن ( True ) القيمة الجملة هذه تعطي

-

i

"

"

S

-

c

]

a [ المدى ضمن يقع لا k الحرف لأن ( False ) القيمة الجملة هذه تعطي

-

e]

! الرمز - 5

معين مدى ضمن رقم، او معين حرف وجود لنفي الرمز هذا استخدام يمكن  
A [المدى ضمن يقع لا X أن أي

-

ذلك بوضع التالي المثال ؛! D

' Computer Skills" Like "C[i-p]mputer Ski[!i-p]ls "

ضمن يقع الأولى السلسلة من 0 الحرف لأن ) False ( القيمة الجملة هذه تعطي

i [ المدى

-

i [ المدى ضمن يقع ا الحرف لكن [ p

-

افان النفسي علامة وجود ومع ، [ p

) False ( هو الناتج

؟

( 2 ) الذاتي التقويم أسئلة

Like دالة في !و ؟ استخدام بين الفرق ما

( StringRev ) الرمزية السلاسل ترتيب عكس دالة 4

ة الرمزي السلاسل ترتيب عكس إمكانية بيك فيجوال البرمجة لغة توفر

:التالية الدالة باستخدام

StringRev(String)

التالي والتطبيق رموزها عكس المراد الحرفية السلسلة هي tring ان حيث

.ذلك واضح

التد الوحدة

272

النسومر و والجلال الرموز

Characters, Strings and Text

( 1 ) تدريب

( 7

.عكسية بصورة وإعادتها رمزية سلسلة بإدخال طريقه عن يقوم برنامجا اين

( UpperCase ) كبيرة حروف إلى صغيرة حروف من التحويل دالة -

الصيغة وفق كبيرة حروف إلى الصغيرة الحروف لتحويل الدالة هذه تستخدم

:التالية

UCase(String)

كبيره حرف إلى الرمزية السلسلة حروف من صغير حرف كل تحويل يتم حيث

UCase" ( مثال

(

"

Computer Skills

COMPUTER SKILLS هو الدالة هذه ناتج

( Lower Case ) صغيرة حروف إلى كبيرة حروف من التحويل دالة - 6

الصيغة وفق صغيرة حروف إلى الكبيرة الحروف لتحويل الدالة هذه تستخدم

التالية:

**LCase(String)**

صغير حرف إلى الرمزية السلسلة حروف من كبير حرف كل تحويل يتم حيث

**LCase** " ( مثال

(

"

**COMPUTER SKILLS**

**computer skills** هو الدالة هذه ناتج

( 3 ) التطبيق

ف وأحر كبيرة بأحرف كتابتها ثم رمزية، سلسلة بإدخال التطبيق هذا في سنقوم  
مغيرة.

( 7 ) بالشكل الموضح النموذج صمم -

**Form1**

**Enter thing**

**Upper Cato**

( **Upper Case** ) الامر زر على المزدوج بالنقر الشيفرة محرر إلى انتقل 2-

( 8 ) بالشكل الموضحة الشيفرة واكتب ( **Lower Case** ) و

\* \_

- **Project1 - Form1 (Code)**

**Click**

**Command2**

( **Private Sub Command1\_Click** )

**Text2.Text = UCase (Text1.Text)**

**End Sub**

( **Private Sub Comand2\_Click** )

**Text3.Text = LCase (Text1.Text)**

**End Sub**

الكبيرة والحروف الصغيرة الحروف شيفرة : ( 8 ) الشكل

داخل رمزية سلسلة أدخل ثم ، ( **F5** ) على بالنقر السابق البرنامج نفذ - 3

هو كما ( **Lower Case** ) و ( **Upper Case** ) الأمر زري على انقر ثم ( **Text Box** )

( 9 ) بالشكل موضح

**EX**

**Form1**

**Computer**

**Entet string**

**Lower Care**

**Upper Case**

**computer**

**COMPUTER**

وبالعك، الكبيرة إلى الصغيرة الحروف من التحويل التي نتيجة : ( 9 ) الشكل

## ة الجزئي الرمزية السلاسل معالجة 2.2

Mids, Lens, Rights, Instr, InstrRev, Trim, LTrims, RTrims, strings ,  
Spaces

التي الجاهزة، الضمنية الدوال من مجموعة بيسك فيجوال البرمجة لغة توفر  
الدوال هذه ومن. الرمزية السلاسل رموز مع التعامل خلالها من يمكن

Nids دالة -

: التالية الصيغة وفق رمزية سلسلة من جزء بإعادة الدالة هذه تقوم

Mid(String, Start, [Length])

274

الآلة الوحدة

والنصوص الهلال و الرموز

Characters, Strings and Text

String: ة الرئيسي الرمزية السلسلة هي

ه إعادت سيتم الرئيسية الرمزية السلسلة في رمز أول موقع : Stan

سيتم التي الرمزية السلسلة طول تحديد خلالها من ويتم اختيارية، في : Length  
السن البداية نقطة من الرمز تعد الدالة فان كتابتها علم حب وفي إعادتها،

. الرئيسية الرمزية السلسلة نهاية

الدالة هذه عمل توضح التالية والأملة

Mid("ABCDEF", 1, 3) (

Mid("Visual Basic", 8 ((

ABC :هي النتيجة أن أي ؛ 3 يطول أ الموقع من المعالجة عملية ستبدأ

8 ب (

"

ه لأن الرئيسية، الرمزية السلسلة نهاية إلى 8 الموقع من المعالجة عملية تبدأ

Basic :هي النتيجة فان وبالتالي المطلوب؛ الطول تحديد يتم لم

Mid ", 4,2) ( ج (

" (

Student

de :هي النتيجة أن أي 2 يطول لا الموقع من المعالجة عملية ستبدأ

Les دالة -

: التالية الصيغة وفق رمزية سلسلة من الأيسر الجزء لإعادة الدالة هذه تستخدم

Lift(String, Length)

حيث

ة الرئيسي الرمزية السلسلة هي : String

الجزء من إعادتها سيتم التي الرمزية السلسلة طول تحديد خلالها من ويتم : Length

.الرمزية السلسلة الأيسر

:ذلك تبين التالية والأمثلة

Left("ABC D E", 4) (

للسلسلة الأيسر الجزء من تبدأ رموز 4 بطول رمزية سلسلة الدالة هذه ستعيد

ABC : هي النتيجة أن أي الرئيسية؛ الرمزية

"" (ب

Left("Programming Language", 11) (

ال الأيسر الجزء من فدا رمزا 11 بطول رمزية مراسلة الدالة هذه ستعيد

Programming : هي النتيجة أن أي الرئيسية؛ الرمزية

وشموع والسلاسل الرموز

275

Characters, Strings and Text

التاسعة الوحدة

Left("CH+Language", 3) (

اتيلن الجزرهای من تبدأ رموز، 3 بطول رمزية سلسلة الدالة هذه استعيد

ج بي : هي النتيجة أن أي الرئيسية؛ الرمزية

Rights دالة - 3

التالية الصيغة وفق رمزية سلسلة من الأيمن الجزء لإعادة الدالة هذه تستخدم

Right (String, Length)

:حيث

ة الرئيسي الرمزية السلسلة هي : String

الجزر من إعادتها سيتم التي الرمزية السلسلة طول تحديد خلالها من ويتم : Length

. الرمزية للسلسلة الأيمن

:ذلك تبين التالية والأمة

Right ("A BCD E", 4)

Right ("Programming Language", 8) (

للسلسلة الأيمن الجزء من تبدأ رموز، 4 بطول رمزية سلسلة الدالة هذه ستعيد

CDE : هي النتيجة أن أي الرئيسية؛ الرمزية

"" (ب

للسلسلة الأيمن الجزء من تبدأ رموز، 8 بطول رمزية سلسلة الدالة هذه ستعيد

Language : هي النتيجة أن أي الرئيسية؛ الرمزية

"," (ج

للسلسلة الأيمن الجزء من تبدأ رموز، 7 بطول رمزية سلسلة الدالة هذه ستعيد

science : هي النتيجة أن أي الرئيسية؛ الرمزية

Right ("Computer science", 7) la

Instr ( البحث دوال - 4

,

InstrRev )

سلسلة داخل رمزية سلسلة عن البحث إمكانية بيسك فيجوال البرمجة لغة توفر

:التاليتين الدالتين إحدى باستخدام أخرى رمزية

Inslr دالة -

حي أخرى، رمزية سلسلة داخل رمزية سلسلة عن للبحث الدالة هذه تستخدم

التالية الصيغة وفق اليمين إلى اليسار من البحث عملية تبدأ

Instr( [Start],String1,String2, [Compare])

276

حديث:

ت و

أ يبد البداية نقطة تحديد عدم حال وفي :اليسار من البحث بداية نقطة تحديد : Start

.اليسار أقصى من البحث

..داخلها البحث المراد الرئيسية الرمزية السلسلة : String

.عنها البحث المراد الفرعية الرمزية السلسلة : String2

:هما قيمتين إحدى وتأخذ اختيارية هي Compare

أي ( البحث عملية في والكبيرة الصغيرة الحروف بين فرق هنالك يكون لا عندما :

أن A

=

a )

أن أي ( البحث عملية في والكبيرة الصغيرة الحروف بين فرق هنالك يكون عندما :

( A<>a

:التاليتين القيمتين إحدى الدالة هذه وتعطي

. String1 داخل String2 موقع تمثل صحيح عدد

. String1 داخل String2 وجود عدم عند : 0

هي : N

InstrRev دالة -

إلى اليمين من البحث بعملية تبدأ أنها إلا ، Instr دالة تماما تشبه الدالة هذه

:التالية الصيغة وفق اليسار

InstrRev(Start, String 1,[ String2],[Compare])

.السابقتين الحاليتين عمل توضح التالية والأمثلة

( InStr(1, "Aeaba", "A")

مسن الفرعية الرمزية السلسلة موقع تمثل التي 1 القيمة الدالة هذه تعطي

. الرئيسية الرمزية السلسلة

1 , "ب"

,

"

"

,

a

السلسلة من الفرعية الرمزية السلسلة موقع تمثل التي 3 القيمة الدالة هذه تعطي

. الرئيسية الرمزية

. A عن يختلف a الحرف أن لاحظ

( 3 ج

'  
"

السلسلة من الفرعية الرمزية السلسلة موقع تمثل التي 5 القيمة الدالة هذه تعطي

a

InStr(3, "ZZASWAQRF", "WA") (

. الرئيسية الرمزية

والنصوص والسلال الرموز

Characters, Strings and Text

277

التاسعة الوحدة

InStrRev " ( ) د

(  
"

How Are You

"

,  
"

r

السلسلة من الفرعية الرمزية السلسلة موقع تمثل التي 6 القيمة الدالة هذه تعطي

. الرئيسية الرمزية

InStrRev " ( ) هـ

(  
"

How Are You

"

,  
"

z

في موجودة غير الفرعية الرمزية السلسلة لأن 0 القيمة الدالة هذه تعطي

. الرئيسية الرمزية السلسلة

Trin ( الدوال - 5

,

LTrin

,

RTrim )

أو الرمزية، الثوابت ويسار يمين على الفراغات لحذف الدوال هذه تستخدم

محدود نص صندوق في رسائل عرض نود عندما الدوال هذه وتستخدم كليهما، على

:التالية الصيغة وفق الرسالة جانبي على الزائدة الفراغات حذف يستدعي مما الطول،



( Trim

(

String الرمزية السلسلة جانبي على الفراغات لإزالة تستخدم : String

( LTrim

(

String الرمزية السلسلة يسار على الفراغات لإزالة تستخدم : String

( RTrim

(

String الرمزية السلسلة يمين على الفراغات لإزالة تستخدم : String

:ذلك توضح التالية والأمثلة

11

S=" Computer Science

Print Trim(S)

Print LTrim(S)

Print RTrim(S)

Computer Science الرمزية السلسلة ويسار يمين على الفراغات إزالة هو الأولى الجملة ناتج

Computer Science الرمزية السلسلة يسار على الفراغات إزالة هو الثانية الجملة ناتج

Computer Science الرمزية السلسلة يمين على الفراغات إزالة هو الأولى الجملة ناتج

Space ( 0 ) ( الدوال - 6

(

)

,

String )

تكون أن على طول، بأي رمزية سلسلة خلق السابقتين الدالتين باستخدام يمكن  
وفق واحد رمز من مكونة رمزية سلسلة أو ( Space ) فراغات الرمزية السلسلة هذه

:التالية الصيغة

( Space

(

n بطول فراغات لترك تستخدم : n

( String

(

n

,

الرمز من مكونة n طولها رمزية سلسلة لإنشاء تستخدم : Character

Charact

توضح التالية والأمثلة

د ز ا

Print "Computer", Space(10), "Skills")"

Skills فراغات 10 بينهما الكلمتين طباعة الجملة هذه ناتج

( , "ب

AAAAA مرات خمس " A " الحرف طباعة الجملة هذه ناتج

( Print String(5,"A") (

Computer

Replace دالة باستخدام السلاسل إحلال 3.2

: التالية الدالة باستخدام أخرى رمزية بسلسلة رمزية سلسلة استبدال يمكن

Replace(String, Str1, Str2)

حيث:

String : هي

ة رمزي بسلسلة منها جزء استبدال سيتم التي الرئيسية الرمزية السلسلة  
أخرى.

Str1 : هي

Str2 : هي

. استبدالها سيتم التي الرئيسية الرمزية السلسلة ضمن موجودة حرفية سلسلة  
مكان الرئيسية الرمزية السلسلة في وضعها سيتم جديدة حرفية سلسلة  
ذلك يوضح التالي التطبيق و Str1 الرمزية السلسلة

( 4 ) التطبيق

ة مدخل حرفية سلسلة على والاستبدال البحث عملية التطبيق هذا في سنجري  
:التالية الخطوات وفق

الأول النص صندوق أن ملاحظة مع ( 10 ) بالشكل الموضح النموذج

معهم

أكثر إدخال من لنتمكن MultiLine خاصية يستخدمان الرابع 2 النص وصندوق  
النص صندوق نفس في سطر من

- 1

Form

te

End

Replacement

Replace with

Test Altes Replacement

. المصمم النموذج (: 10 ) الشكل

والنصرم والسلاسل الرموز

Characters, Strings and Text

279

انواعها

والكتاب ( Replace ) الأمر زر على المزدوج بالنقر الشيفرة محرر إلى انتقل -

( 11 ) بالشكل الموضحة الشيفرة

Project1 Form1 (Code)

Command

Click

Private Sub Command1\_Click ) (

Dim str As String

Dim stri As String

Dim str 2 As String

str = Text1. Text

stri = Text2. Text

str2 = Text 3. Text

Text 4. Text - Replace (str, stri, str2)

End Sub

الكبيرة والحروف الصغيرة الحروف شيفرة : 11 ( الشكل

النص مربع داخل رمزية سلسلة أدخل ثم ، F5 ( على بالنقر السابق البرنامج نفذ - 3

د المرا الرمزية والسلسلة عنها البحث المراد الرمزية السلسلة و " Text Box (

( 12 ( بالشكل موضح هو كما ) Replace ( الأمر زر على انقر ثم استبدالها،

Form1

سه

Text

Computer Science

Find

Science

Replacement

Replace with

Skills

Text After Replacement

Computer Skills

الاستبدال دالة نتيجة : 12 ( الشكل

280

السلاسل تحويل في

Asc, Chr\$, IsNumeric, Val, Str\$, Hex\$, Oct\$

خلالها من يمكن التي الجاهزة، الضمنية الدوال من مجموعة البرمجة لغة توفر

التحويلية الدوال بعض يبين ( 1 ( والجدول آخر؛ إلى شكل من البيانات تحويل

التحويلية الدوال بعض : 1 ( الجدول

الصيغة

الوصل

'( ASC

(

"

للمر ASCII شيفرة تعيد Character

( Chr

(

شيفرة من N لقيمة المقابل الرمز تعيد N

ASC

65

Print Asc("A")

Print Chr(65)

A

Chr

ASCII

True

IsNumeric

( IsNumeric

(

Print IsNumeric ( 457 ) ( كان إذا True القيمة تعطي منطقية دالة Exprission

القيمة تعطي ذلك عكس و عادية التعبير

False

Val

Print Val("786")

786

Print Str789 " 789 )

Str

Print Hex(20)

14

"( Val

(

"

ة عددي إلى الرمزية السلسلة تحول String

( Str

(

السلسلة إلى العادي المتغير تحول Nuthber

ة الرمزي

( Hex

(

إلى العشري النظام من العدد تحول | Number

عشر السداسي النظام

( Oer

(

إلى العشري النظام من لعدد اتحول Number

ي الثان النظام

Hex

Print Oct (20)

24

Oct

281

والنصوص والسلام الرموز

Characters, Strings and Text

السلاسل تهيئة

يستخدم كان معينة، صيغ باستخدام قيمة إظهار إلى المبرمج يحتاج ما كثيرا  
فيجوال البرمجة لغة توفر الغاية ولهذه وغيرها؛ % المنوية النسبة وعلامة بسمه  
ة التهيئ دالة الدوال هذه أشهر ومن .مناسبة بصيغ معينة قيمة عرض إمكانية لك

: التالية العامة الصيغة وفق معينة بصيغة المقترنة ( Form

Format(Value, "Formatting General")

Val : معينة بصيغة عرضها المراد القيمة هي :

من مجموعة وهناك القيمة، فيها ستظهر التي الصيغة هي : Formatting Gene -  
الصيغ هذه بعض يبين ( 2 ) الجدول .بيسك فيجوال البرمجة لغة توفرها التي سيغ

Format(100, "Currency")

00

" Currenc

Format(545.589, "Fixed")

545.59

" Fixed "

847.88

" Standard

Format(847.875, "Standard")

اللغة توفرها التي الصيغ بعض : ( 2 ) الجدول

الوصف

مثال

على \$ شارة تظهر العملة صيغة

ن عشريتا ومنزلتان القيمة يسار

العشرية الفاصلة يمين على

| القيمة تظهر الثابتة الصيغة

ط فق بقيمتين العشرية

ة القيم تظهر المعيارية الصيغة

ة فاصل ووضع فقط بقيمين الشرية

للآلات

القيمة عرض العلمية الصيغة

\*\*\* E )\*\*\* (العلمي بالأسلوب

ة القيم عرض المنوية النسبة صيغة

اطهار مع 100 % في مضروبة

يمين على المنوية النسبة الإشارة

.عشريتين ومنزلتين الرقم

( م نع ) المنطقية الصيغة

No تعطي 0 Yes: اتعطي

( أخطا صح ) المنطقية الصيغة

False تعني 0 True: اتعطي

4E02

Format( 847, "Scientific")

" Scientific

4700.00 %

" Percent

Format(475, "Percent")

" Yes/No "

Format(0 ,

" Yes/No )"

raise

" True/False

Format(0, "True/False")

" On/OIP

Format(1, "On/Off")

(مفصول موصول) المنطقية الصيغة

OR تعلي 0: On تعطي

General numnt العالية الصينية

47

"

Format(847, "General number")

282

واللصوم والسلاسل الرموز

Characters, Strings and Text

يمين على الخانات عدد تحديد خلالها من يمكن التي الصيغ بعض وهناك

ذلك يبين ( 3 ) والجدول العشرية؛ الفاصلة ويسار

النتائج

والصحيحة العشرية الخانات عدد تحدد التي الصيغ بعض (: 3 ) الجدول

الوصف

ة منزل أي بدون القيمة تظهر

د عي قرب إلى ومقربة عشرية

الأقل على يظهر بحيث صحيح

الكاملة يسار على واحدة منزلة

العشرية

عشريتين بمنزلتين القيمة تظهر

Format(-147,954, 0)

- 148

" 0.00 "

Format(758.587, 0.00")

758.59

### 0.00 "

54872.87.00

عشريتين بمنزلتين القيمة تظهر

على واحدة صحيحة ومنزلة فقط

فواصل وضع ضرورة مع الأول

للان

Format(54872.875 ,

#.# 0.00

الفتح

.100\$

" 0 "%

| بالقيمة مضروبة القيمة تظهر

100

ة عدني قيمة إلى ومقربة %

عشرية منازل دون صحيحة

المنوية النسبة بإشارة متبوعة و

Format(6,451, ( )

645 %

545 .

847

( 5 ) تطبيق

ق وف قائمة صندوق داخل مختلفة صيغ بإظهار التطبيق هذا في سنقوم

:التالية الخطوات

-

( 13 ) بالشكل الموضح النموذج صمم

|

Formt

Format

المصمم النموذج : ( 13 ) الشكل

والنصوص والبلامل الرموز

Characters, Strings and Text

283

المهام عدد

ة الصيني والمعالجة ) Time ( والوقت ) ate [ التارية :

الدولة هذه يبين ( 4 ) والجدول والوقت التاريخ التعامل خلالها من يمكن

والتي ببسك، فيجوال البرمجة لغة توفرها التي الجاهزة الضمنية الدوال من

.والتاريخ الوقت دوال : ( 4 ) الجدول

الوصف

ز تجي التي تاريخ شد

التاريخ من الحالي ليوم تعيد Dya (

لجهاز الحلي لوتي

الحالي الوقت من تساعة تي HouT (

تحاليين وتاريخ نوت تي Now

العامة الهينة

Date

Day

Time  
 How  
 Now  
 Print Date  
 30.01.2006  
 Print Day(date)  
 30  
 10:41:14 AM Print Time  
 Print Hour(Time)  
 10  
 Print Now  
 30.01.2006  
 01:14 AM  
 Print Month(Date)  
 T  
 January  
 ( Date ) )  
 Moot  
 Month Nam  
 ( Won Date الحالي تاريخ من شهر تعد  
 ( MorthkamaDate Print MonthNameMonth التاريخ من الشير قسم تين  
 الحني  
 ( Wedda date محصورة علي قيمة ت  
 الحالي اليوم تمت - مين  
 قحاني تموم لسم تي )  
 Weekday  
 Print Weekday(date)  
 Whale WeekdayName(Date) Weekday name  
 Sunday  
 Print  
 WeekdayName(Weekday  
 ( Date )  
 Print year(Date)  
 IsDate 3/2/1990 "  
 2006  
 Year  
 True  
 IsDate  
 ( Year  
 (  
 Date الحالية السنة ني  
 iiDa Variable True تنقية تي منطقية دل  
 التغير تحويل بالإمكان كان ذا  
 ن وك Date بيتي نوع بني



false تتية تعيد نت

( الوقت من الحالية الدقيقة تعيد Minute Time )

الوت من الحالية نية ني Saudi Time ( )

متصف من والتي د تا

النتي تو كي تيل

41

Print Minute(Time)

Print Second(Time)

14

Second

Print Timer

7581247

Timer

:ملاحظات

يبين ( 5 ) والجدول رمزية؛ وقيم عديدة كثوابت ببسك فيجوال لغة في الأيام تعرف

ذلك.

الجمال خلال من VBFriday (الجمعة)

م الأول باليوم التحكم ويمكن ، VBSunday هي الأول لليوم الافتراضية القيمة

المثال سبيل فعلى الأول؛ اليوم يمثل Weekday دالة إلى معامل إضافة خلال

هو الأول اليوم جعل يمكن

- onth Name الشهر واسم Weekday Name اليوم اسم دالتي تعيد أن يمكن

. Weekday(Date, VBFriday)

والنصوص والسلاسل الرموز

285

Taracters, Strings and Text

في ( True ) Jan ( القيمة يأخذ السابقتين الدالتين إلى منطقي معامل بإضافة وذلك

تصبح ) . January ( و ) Sun ( تصبح ) Sunday ( والشهر لليوم مختصرة أسماء

استخدام عدم حال في False والقيمة الاختصارات استخدام حال

توضح التالية والأمثلة الاختصارات،

ذاك

)

WeekdayName(Weekday (Date),True)

MonthName (

Month (Date), True )

.الأسبوع أيام معرفات (: 5 ) الجدول

ة الحرفي القيمة الصحيحة القيمة

الأول

1

الثاني

2

3

4

VBSunday  
VB Monday  
VBTuesday  
VBWednesday  
VBThursday  
VBFriday  
VBSaturday

ع الراب  
الخامس  
السادس  
السابع

5  
6  
7

**Date Value Ulls -1**

والتاريخ الوقت صيغ دوال  
دالة

: ذلك توضح التالية والأمثلة التاريخ، بصيغة للتحكم الدالة هذه استخدام يمكن

**DateValue("20-10-1979") (**

إلى) (\*\*\*\*\*) \* (الصيغة تحويل تم حيث 1979 / 10 / 20 الدالة هذه استعيد  
(\*\*\*\*/\*\*) (الصيغة)

'ب

الشهر رقم إلى الشهر اسم تحويل تم حيث 1988 / 2 / 15 الدالة هذه ستعيد

**DateValue("February 15,1988") (**

**DateValue("21-Feb-88") (a**

والإشارة رقم إلى الشهر اختصار تحويل تم حيث 88 / 2 / 21 الدالة هذه ستعيد  
الإشارة إلى

**Time Value - 2**

From 0:00:00 التالية بالصيغة الوقت الدالة هذه تعيد

(

**12 AM**

)

**To)**

**( PM**

يوضح التالي والمثال ( 12:59:59

:

**TimeValue("17:45")**

AM التالية بالصيغة الوقت الحالة هذه تعد

**05:45:00**

**286**

والسلاسل الرموز

rings and Text

10 .

المصطلحات مسرد .

ف حرو إلي الكبيرة الحروف لتحويل الدالة هذه تستخدم : LowerCase ( دالة - صغيرة

ي حروف إلى الصغيرة الحروف لتحويل الدالة هذه تستخدم : UpperCase ( دالة - كبيرة

الكلمات منها تتكون التي الأساسية البنية هو : Character (الحرف) الرمز - والثوابت والمتغيرات

Characters ( الرموز من مجموعة : Character Strings الحرفية السلاسل  
الخامسة والرموز ) Digits ( والأرقام ) Letters ( الهجائية الحروف من تتكون  
(

( Special Characters )

- ASCII PULI (American Standard Code Information Interchange) :

( نظام

باعداد الرموز هذه تمثيل يمكن حيث رمزا، 256 لتمثيل يستخدم نظام هو

صحيحة محصورة بين \_\_. 255 - 0

2. مربعات الرسائل Messages Boxes

12 عرض مربعات الرسائل

تستخدم مربعات الرسائل العرض رسائل معينة على الشاشة، وتأخذ الشكل

البرمجي

التالي:

"MsgBox "Message", Symbol, "Title

حيث:

MsgBox: تدل على أن العملية هي إظهار مربع رسائل (Messages Box) .

Message: الرسالة التي ستظهر داخل مربع الرسالة.

Symbol: يمكن إظهار أيقونات (Icons) أو أزرار أوامر (Command Buttons)

داخل مربع الرسائل. ولكل أيقونة أو أمر قيمة صحيحة (Integer) أو قيمة

ثابتة حرفية (Constant). ويمكن التعامل مع الأيقونات أو الأوامر من

خلال قيمها الصحيحة أو الحرفية كما سنرى لاحقا.

Title: هو عنوان الرسالة الذي سيظهر في شريط العنوان الخاص بمربع الرسائل.

يمثل الجدول (1) مجموعة من الأيقونات والقيمة الصحيحة والحرفية لكل منها.

الأيقونة

الجدول (1): الأيقونات (Icons) وقيمها الصحيحة والحرفية.

القيمة الصحيحة القيمة الحرفية

الاستخدام

عند حدوث خطأ

X

16

vbCritical

?

32

vbQuestion

عند طلب النظام معلومات

إضافية من المستخدم

عند حدوث خطأ،

والمستخدم يقوم بتصحيحه

عند اخبار المستخدم

بمعلومات حول التطبيق

48

vbExclamation

64

vbInformation

بينما يظهر الجدول (2) مجموعة من الأوامر والقيمة الصحيحة والحرفية لكل منها.

vbExclamation ؟ أعط بعض الأمثلة

اسئلة التقويم الذاتي (1)

ما الفرق بين

vbCritical

و

الجدول (2): أزرار الأوامر (Command Buttons) وقيمها الصحيحة والحرفية.

القيمة الصحيحة

القيمة الحرفية

OK

0

VbOKOnly

VbOKCancel

OK and Cancel

1

vbAbortRetryIgnore

2

Abort, Retry and

Ignore

Yes, No and Cancel

3

vb YesNoCancel

vb Yes No

Yes and No

4

vbRetryCancel

Retry and Cancel

5

مثال (1)

الجملة البرمجية التالية تقوم بإظهار مربع رسائل يحتوي على الرسالة التالية:

New Messages Box

"MsgBox "New Message Box

لاحظ أن عنوان مربع الرسائل، وظهور الأيقونات و أزرار الأوامر اختيارية؛

والشكل (1) يبين مربع الرسائل الناتج.

Project 1

New Message Box

Ck

الشكل (1): مربع الرسائل الناتج من مثال 1

تطبيق (1)

سنقوم في هذا التطبيق بإظهار مربع رسائل يحتوي على الرسالة التالية:

New Message Box والعنوان Visual Basic ويحتوي على زري الأمر OK

Cancel.

232

الوحدة الثامنة

البناء هذا التطبيق نتبع الخطوات الآتية:

آه منم النموذج الموضح في الشكل (2).

O x

Form 1

٣

Message Box

In I

E

الشكل (2): نموذج تصميم مربع الرسائل

2- انتقل إلى محرر الشيفرة بالنقر المزدوج على زر الأمر (Message Box) واكتب

الشيفرة الموضحة في الشكل (3).

Project)

-

Format

)

Code را

-

Click

Command1

(Private Sub Command1\_Click

"MsgBox "New Message Box", VbOKCancel, "Visual Basic

End Sub

الشكل (3): الشيفرة باستخدام القيم الحرفية الثابتة.

3- نفذ البرنامج السابق بالنقر على (5)، ثم انقر على زر الأمر (Message Box)

ليظهر الشكل (4)

لاحظ عزيزي الدارس أنه يمكن استخدام القيم الصحيحة بدلا من القيم الحرفية

الثابتة، فمثلا يمكن استخدام القيمة 0 بدلا من القيمة الحرفية vbOkOnly يمكن

استخدام القيمة 1 بدلا من القيمة الحرفية vbOkCancel وهكذا... هذه الإمكانيات

موضحة من خلال التطبيق (2)

تطبيق (2)

سنقوم في هذا التطبيق بإدخال قيمة من لوحة المفاتيح داخل مربع نص Text

(Box)، وإظهار القيمة المقابلة لها باستخدام شيفرة ASCII داخل مربع رسائل يحتوي

على أيقونة عنوانه ASCII CODE.

1- صمم النموذج الموضح في الشكل (5).

Form1

E

Enter Value

ASCII

أمر

الشكل (5): وضع مربع نص (Text Box) وأداة عنوان (Label) وزرا

(Command Button) داخل نموذج.

234

ادا

- اضبط الخصائص التالية:

Object

Label

Setting

Labell

Center

Tahoma

Bold

14

Text1

Center

Tahoma

Property

Name

Alignment

Font

Font style

Font Size

Name

Alignment

Font

Font style

Font Size

Text Box

Bold

14

- انتقل إلى محرر الشيفرة بالنقر المزدوج على زر الأمر (Command Button)،

واكتب الشيفرة الموضحة في الشكل (6). لاحظ إمكانية استخدام القيم الصحيحة

بدلاً من القيم الحرفية الثابتة.



-

Project)

-

Form1

)

Code را

Click

Command1

(Private Sub Command1\_Click

msg = 64

"MsgBox Asc (Text1.Text), msg, "ASCII CODE

End Sub

الشكل (6): شيفرة زر الأوامر (Command Button)

4 نفذ البرنامج السابق بالنقر على (F5)، ثم أدخل القيمة (A) داخل مربع النص، ثم

انقر على زر الأمر (ASCII) كما هو موضح في الشكل (7) ليظهر مربع

الرسائل الموضح في الشكل (8).

Form1

A

Enter Value

ASCII

الشكل (7): نموذج الإدخال

235

أميرات الحوار

الشكل

5- كرر التنفيذ على قيم أخرى.

الاحظ أيضا عزيزي الدارم، أنه يمكن الدمج بين الأيقونات وأزرار الأوامر

داخل مربع الرسائل باستخدام الرمز (+). التطبيق (3) يبين هذه الفكرة.

التطبيقات

سنقوم في هذا التطبيق بإظهار مربع الرسائل الذي يحتوي على أيقونات وأزرار أوامر معا باستخدام الرمز (+) حيث يحتوي على:

العنوان: EXIT

زر الأمر: Yes and No

الرسالة: Are You Sure

الأيقونة: vbQuestion

:

التنفيذ هذا التطبيق نتبع النقاط الآتية:

1- صمم نموذجاً مناسبة

2- الشيفرة في الشكل (9) توضح ذلك

Project1)

-

Form1

)

با Code

Click

Command1

(Private Sub Command1\_Click

msg = vb YesNo + vbQuestion

"MsgBox "Are You Sure", msg, "Exit

End Sub

الشكل (9): استخدام القيم الصحيحة والحرفية معا .

236

3- نفذ البرنامج لتلاحظ ظهور الشكل (10):|

x

Exit

?Are You Sure

?

No

Yes

الشكل (10): مربع رسائل يحتوي على أيقونات وأزرار أوامر  
و يمكن استبدال القيم الحرفية بقيم عددية صحيحة كما هو موضح بالشفيرة الظاهرة  
في الشكل (11)، لتعطي نفس مربع الرسائل الظاهر في الشكل (10):

X

Project1 - Form1 (Code) -

Click

Command1

(Private Sub Command1\_Click

msg = 36

"MsgBox "Are You Sure", msg, "Exit

End Sub

الشكل (11): شيفرة التطبيق (3) باستخدام قيم عددية  
حيث يتم إيجاد مجموع قيمة vbQuestion مع قيمة vbYesNo وهي (4+32)  
= 36

فمثلا القيمة 33 تستخدم لإظهار مربع الرسائل الظاهر في الشكل (12)،  
والقيمة 67 تستخدم لإظهار مربع الرسائل الظاهر في الشكل (13) وهكذا...

X

Exit

?Are You Sure

?

CI

Cancel

الشكل (12): مربع رسائل قيمته 33

## 22 استخدام دالة مربعات الرسائل Message Boxes function

قد نحتاج إلى القيمة التي يعيدها مربع الرسائل، وبالتالي لا بد من استخدام دالة تعيد هذه القيمة وهي دالة مربع الرسالة (MSGBOX)؛ حيث تختلف هذه الدالة عن الرسائل في أنها تعيد قيمة يمكن الاستفادة منها، وتنتهي بقوسين للدلالة على أنها دالة؛ والشكل البرمجي التالي يوضح الصيغة العامة لهذه الدالة:

Variable = MsgBox(Message, Symbol, Title)

حيث إن:

Variable: هو المتغير الذي تعاد فيه قيمة الدالة، وهي إحدى القيم الصحيحة الواردة

في الجدول (3)

أما بقية عناصر الدالة، فهي تشبه ما تم شرحه في مربع الرسائل.

الجدول (3): القيم التي تعيدها دالة MsgBox()

القيمة الصحيحة التي تعيدها

القيمة الحرفية الثابتة

الدالة

زر الأمر

OK

Cancel

Abort

Retry

Ignore

Yes

No

1

2

3

4

5

VBOK

vbCancel

vbAbort

vbRetry

vbIgnore

vb Yes

vbNo

7

؟

أسئلة التقويم الذاتي (2)

1. ما أهمية دالة مربع الرسالة (MSGBOX)؟

2. ما الفرق بين مربع الرسالة ودالة مربع الرسالة (MSGBOX)؟

تطبيق (4)

سنقوم، عزيزي الدارس، في هذا التطبيق بإظهار مربع رسائل (Dialog

Box). يطلب منك تأكيد عملية مسح محتويات مربع نص (Text Box) قبل إتمام عملية

مسح محتوياته وإظهار مربع رسائل لتأكيد عملية إنهاء البرنامج. لتنفيذ هذا التطبيق

اتبع الخطوات الآتية:

3. مربعات الحوار الشائعة Common Dialog Control

توفر لغة فيجوال بيسك مجموعة من مربعات الحوار الشائعة or

(Dialog Control) دون الحاجة لكتابة برامج خاصة بها، حيث تستطيع استخدام لسنه

المربعات أثناء تنفيذ البرنامج، وعندما تحتاج إلى استخدام مربعات الرسائل العامة

داخل برنامجك يجب عليك القيام بما يلي:

- أنشئ نموذجاً جديداً.

2- من لائحة Project إختَر Components كما هو مبين في الشكل (19).

Project Format Debug Run

Add Form

Add MDI Form

Add Module

Add Class Module  
Add User Control  
Add Property Page  
Add User Document  
Add WebClass  
Add Data Report  
Add DHTML Page  
Add Data Environment  
...More ActiveX Designers  
Add File... Ctrl+D  
Remove Formi  
...References  
...Components  
Ctrl+T  
...Project Properties

الشكل (19): اختيار الأمر Components من لائحة Project

### 1.3 مربع حوار فتح ملف Open Dialog

يمكن فتح أي ملف من داخل برمجية فيجوال بيسك، بعد إظهار مربع الحوار  
)

الخاص بفتح الملفات (Open Dialog)، من خلال الخاصية (Show Open) لأداة  
(Common Dialog) التي يمكن إظهارها على شريط أدوات التحكم، باستخدام  
الخطوات التي تم ذكرها سابقا .

#### تطبيق (5)

سنضيف في هذا التطبيق أداة Common Dialog داخل شريط أدوات التحكم،  
ثم نستخدم مربع حوار فتح الملف لتحديد اسم الملف المراد فتحه. لتنفيذ هذا التطبيق  
نتبع الخطوات الآتية:

1. صمم النموذج الموضح في الشكل (22):

Form 1

The File Selection

Open File

List Clear

الشكل (22): نموذج يحتوي على صندوق قائمة (List Box) وأداة عنوان (Label) وزري امر (Command Button)

2 انتقل إلى محرر الشيفرة داخل زري الأوامر List Clear و KOpen File ثم

اكتب الشيفرة الموضحة في الشكل (23)

PTT

k- Project1 - Form1 (Code)

Command2

Click

(Private Sub Command1\_Click

CommonDialog1.ShowDialog

= If CommonDialog1.FileName

"MsgBox " No File Is Selected

Else

List 1. Add Item CommonDialog1.FileName

End If

End Sub

Then

(Private Sub Command2\_Click

List1.Clear

End Sub

الشكل (23): استخدام مربع الحوار فتح ملف (Open File)

3 نفذ البرنامج ثم انقر على زر الأمر (Open File) لتلاحظ ظهور الشكل (24).

اختر إحدى الملفات المعروضة لتلاحظ ظهور اسمه داخل صندوق القائمة (List Box) كما في الشكل (25)، وعند عدم اختيارك لأي ملف من الملفات المعروضة تلاحظ ظهور مربع الرسائل كما في الشكل (26)، وعند النقر على الزر (List Clear) يتم مسح عناصر القائمة.

Open

Look in

DVB98

font

font}

My Recent

Documents

Setup

Template

Tsal

Wizards

Addscous.di

Ifont

Desktop

e INSTALL

LINK

MSDIS110.DLL

msg

MSPD860.DLL

(MSSCCPR

NWIND

Pr

m Pr



Project

Projects

READMEDT

READMERP

READMESS

READMEVB

REDIST

REPV.B.DLL

Repvbrc.dll

REPVB.TIM.DLL

BX.SCCVBUS

SCCVBUS.CHI

VB6

VB6.OLB

VB6DEBUG.DLL

m VB6EXT.OLB

Vbóide.DLL

VBA6.DLL

VBAEXE6

VBDATA.WW

VBEXT.SRG

VBHELP.SRG

VBSCC.DLL

Visdata

Sasa

BIBLIO

c2

b. doar

CVPACK

Dataviow.de

My Documents

My Computer

File name

My Network

Places

Fies of type

Open

Opon as tead only

Cancel

الشكل (24): مربع الحوار فتح ملف (Open File).

Form 1

X

The File Selection

Open File

D:\Program Files\Micro

D:\Program Files\Micro

D:\Program Files\Micro

List Clear

الشكل (25): إضافة أسماء الملفات المختارة إلى صندوق القائمة

Project 1

No File Selected

OK

الشكل (26): مربع رسائل يظهر رسالة عدم اختيار أي ملف من الملفات المعروضة

يمكن التعامل مع الخاصية Action لإظهار مربع الحوار (فتح ملف) كما في

الجملة التالية بدلا من الخاصية Show Open:

```
CommonDialog1.Action = 1
```

```
]/
```

تدريب (2)

أضف أداة Common Dialog داخل شريط أدوات التحكم، ثم استخدم مزيم

حوار (فتح الملف) لتحديد اسم الملف المراد فتحه. استخدم الخاصية Action الظهر

مربع الحوار (فتح ملف) بدلا من الخاصية Show Open.

يمكن التعامل

خصائص مربع الحوار من خلال خاصية Custom حيث

يظهر الشكل (27)، بعد النقر على هذه الخاصية من مربع الخصائص (properties)

Help

Property Pages

Font Print    Open / Save As color

Dialog Title

Flags

Default Ext

FileName

260

MaxFileSize

Initir

0

Fillerindex

Filter

CancelError

الشكل (27): خصائص مربع الحوار

أسئلة التقويم الذاتي (4)

عدد الخصائص التي يمكن استخدامها عند استخدام مربعات الحوار الشائعة.

### 2.3 مربع حوار حفظ الملفات Save Dialog

يمكن من خلال مربع الحوار (حفظ) إجراء عملية حفظ الملفات التي يحتاجها

مستخدمو فيجوال بيسك؛ وذلك ب تحديد اسم الملف المراد حفظه، وهي تشبه عملية فتح ملف.

#### تطبيق (6)

سنقوم في هذا التطبيق بإظهار مربع رسائل يطلب منك تأكيد عملية الحفظ عند

النقر على زر الإغلاق (X)؛ وعند النقر على الأمر Yes، يظهر مربع حوار حفظ

لإتمام عملية الحفظ، وعند النقر على الأمر No يقوم بإغلاق البرنامج. لتنفيذ هذا

التطبيق نتبع الخطوات الآتية:

- 1- صمم النموذج الموضح في الشكل (28)
- 2- 3.3 مربع حوار اللون Color Dialog
- 3- يمكن التحكم بلون النص المكتوب داخل صناديق النصوص، أو أدوات العنوان
- 4- وغيرها من الكائنات، من خلال مربع حوار الألوان الذي يمكن التعامل معه كما في
- 5- مربعات الحوار السابقة. فلو أردنا التحكم بلون النص المكتوب داخل مزييس
- 6- النص (Text Box) الظاهر في الشكل (32)، بعد النقر على زر الأمر Color نكت
- 7- الشيفرة الموضحة في الشكل (33) ليظهر مربع الحوار (الوان) الظاهر في الشكل
- 8- (348)، ليتم من خلاله اختيار اللون المطلوب (وليكن الأزرق مثلا) ثم النقر على
- 9- الأمر OK ليتم تغيير لون النص حسب اللون المختار وكما هو موضح في الشكل
- 10- (35)
- 11- X
- 12- Form 1
- 13- T
- 14- H
- 15- Color
- 16- الشكل (32): نموذج يحتوي على مربع نص (Text Box) وزر أمر (Command Button)
- 17- (Button)
- 18- X
- 19- Form1
- 20- Visual Basic 6
- 21- Color
- 22- الشكل (35): لون النص بعد اختياره من مربع الحوار ألوان
- 23- يمكن التعامل مع الخاصية Action لإظهار مربع الحوار ألوان كما في الجملة
- 24- التالية بدلا من الخاصية Show Color:
- 25- CommonDialog1.Action = 3
- 26- 4.3 مربع تحديد شكل الحرف Font Dialog
- 27- يمكن التحكم بحجم ونوع ونمط الخط للنص المكتوب داخل صناديق النصوص
- 28- أو أدوات العنوان وغيرها من الكائنات من خلال مربع تحديد شكل الحرف (Font)
- 29- 251
- 30- استخدام مربعات الحوار

الأخذ

(Dialog) الذي يمكن التعامل معه كما في مربعات الحوار السابقة ويجب ا  
الاعتبار القيم الثابتة (Constants) التي تحدد فيما إذا كان الخط يخص الشاشة  
(Screen) أو الطابعة (Printer) أو كليهما كما هو مبين بالجدول (4). فلو أردنا  
التحكم بخط النص المكتوب داخل مربع النص (Text Box) الظاهرة في الشكل (36)  
بعد النقر على زر الأمر (Font) نقوم بكتابة الشيفرة الموضحة في الشكل (37) ليظهر  
مربع تحديد شكل الحرف الظاهر في الشكل (38) ليتم من خلاله اختيار نوع ونمط  
الخط المطلوب وليكن الحجم 14 والنوع (Tahoma) والنمط (Bold Italic) ثم  
النقر على الأمر OK ليتم تغيير النص حسب القيم المختارة وكما هو موضح في  
الشكل (35).

وحجم

الجدول (4) ثوابت الشاشة والطابعة

الثابت

القيمة

اعداد الخط

1

cdICFscreenFonts

الشاشة

2

cdICFPrinter Fonts

الطابعة

الشاشة والطابعة

3

cdICFBoth

يمكن التعامل

الجملة التالية بدلا من الخاصية Show Font:

الخاصية Action لإظهار شكل الحوار (طباعة) كما سى

CommonDialog1. Action = 4

Print Dialog 5.3 مربع حوار الطباعة

يمكن التعامل مع مربع الحوار الخاص بالطباعة (Print Dialog) من حد

الخاصية Show Printer، التي تقوم بعرض مربع حوار الطباعة الشاهد في كل

(40) بعد كتابة الجملة التالية:

CommonDialog1.Showprinter

Print

General

Select Printer

Add Printer

HP DeskJet

2134

=====

Preferences

Print to file

Ready

Status

Location

Comment

Find Printer

Page Range

:Number of copies

Pega

Selection

3

Cancel

الشكل (40) مربع حوار طباعة

يمكن التعامل مع الخاصية Action لإظهار مربع الحوار (طباعة) كما في

الجملة التالية بدلا من الخاصية Show Printer:

CommonDialog1.Action = 5

## الوحدة الرابعة

**1.2 حل مسألة بواسطة الحاسوب ( Problem Solving by Computers )** تعتبر عملية استخدام الحاسوب في حل المسائل من أهم استخدامات هذا الجهاز ، ولإتمام عملية الحل هذه ، فلا بد أن يكون لهذه المسألة طريقة حل على شكل خطوات مرتبة منطقيا ، وهذا ما يسمى بخوارزمية الحل ، ويعتبر عملية وجود هذه الخوارزمية لمسألة ما أحد خطوات حل المسألة بواسطة الحاسوب وتعتبر عملية البرمجة خطوة أخرى وتوضيحا لهذه الأفكار ، نورد أولا بعض التعريفات ذات العلاقة . - خوارزمية ( ALGORITHM ) هي طريقة حل المسألة ما ، موصوفة خطوة خطوة بشكل محدد ، وتكون قابلة البرمجة والتنفيذ في وقت محدد من البرنامج ( PROGRAMM ) وهو مجموعة من التعليمات أو الإيعازات المرتبة ( INSTRUCTIONS ) ، مكتوبة بإحدى لغات البرمجة ، وتهدف إلى تنفيذ غرض محدد ( Task ) في وقت محدد . - البرمجة ( PROGRAMMING ) هي عملية كتابة البرنامج ، وهذه العملية تحتاج إلى معرفة خلفية علمية وتترسخ بزيادة الممارسة من أجل اكتساب المهارة الكافية .

أما خطوات حل مسألة بواسطة الحاسوب ، فيمكن تلخيصها بما يلي : 1 ، مرحلة فهم وتحليل المسألة ( Understanding and Analysing of the Problem ) في هذه الخطوة يتم جمع المعلومات اللازمة عن المسألة ليكون من الممكن صياغتها وتعريفها بشكل إنشائي أو غيره ، بحيث تتحدد ثوابتها ومتغيراتها ، وننكر الأهداف المطلوب تحقيقها لإتمام عملية الحل . مرحلة تصميم ( أو اختيار ) خوارزمية Algorithm ( Designing or choosing An ) إذا كانت هناك عدة طرق متوفرة لحل المسألة التي أتممنا صياغتها في الخطوة السابقة ، فيتوجب في هذه الخطوة اختيار طريقة منها للمضي قدما في عملية الحل . لما إذا لم نكن نعلم بطريقة معروفة لحل المسألة ، فعلى المختصين تصميم طريقة حل مناسبة . 3. مرحلة تمثيل الخوارزمية لتوضيحها ( Representation of the Algorithm ) تعتبر هذه الخطوة اختيارية ، وهي توطئة للخطوة القادمة ، والمقصود منها هو الانتقال من النص الكلامي الإنشائي للخوارزمية إلى صورة ما تقترب بنا أكثر إلى هدف الخطوة القادمة ، وتستخدم هنا فكرة " مخطط سير العمل FLOWCHART أو فكرة شبه البرنامج ( PSEUDOCODE ) للقيام بذلك

4. مرحلة البرمجة ( Programming Phase ) • برمجة الخوارزمية ( Coding the Algorithm ) هنا تستخدم إحدى لغات البرمجة ، مثل فيجوال بيسك 6 ، لكتابة برنامج بهدف تنفيذه على جهاز الحاسوب . : اختبار البرنامج ( Testing the Program ) بعد التأكد من صحة البرنامج من الناحية القواعدية ، والذي تحقق في الخطوة السابقة ، يبقى التحقق من أن البرنامج يعطي النتائج المرجوة بشكل صحيح ، ويفترض في البرنامج أن يعطي نتائج صحيحة عند تعرضه لمختلف أنواع المدخلات ، فإن اجتاز مثل هذا الاختبار ، بحيث أعطى نتائج صحيحة لكل الأنواع الممكنة للمدخلات اللازمة له ، أعتبر برنامجا صحيحا و التوثيق ( Documentation ) من العادات الجيدة أن نكتب ملاحظات توضيحية عن البرنامج : من كتبه ، بأي تاريخ كتب ، بأي لغة كتب ، لأي غرض كتب ، التعريف بالمتغيرات ، والتعريف بأجزاء البرنامج المختلفة وما يؤديه كل جزء هذه العملية تسمى عملية توثيق البرنامج ،

6 وتضاف مثل هذه الملاحظات للبرنامج في أي مكان منه وذلك عن طريق جمل الملاحظات في اللغة .  
در مرحلة صيانة الحل والبرنامج ( Maintenance ) إن أحد أوجه عملية صيانة البرنامج ، هو التمكن من إجراء تعديلات في البرنامج من وقت لآخر ، وذلك على ضوء ما يحدث من تغيرات في المسألة التي كتب البرنامج لحلها ، وهذا ضروري ليغنينا عن عملية كتابة برنامج جديد عند حدوث تغيرات في المسألة ، إذ يكفي أن نقوم بصيانة البرنامج ليبقى صالحا لحل نفس المسألة بعد التعديل .  
6. تطبيق الحل ( Implementing the solution ) عند الحصول على حل المسألة بعد كل المراحل السابقة ، يكون من المنطقي أن انتقل الحل النظري الذي حصلنا عليه إلى الواقع العملي الذي نتجت منه المسألة الأساسية ، ففي الحقيقة نحن نريد حلا قابلا للتطبيق والتنفيذ على أرض الواقع من أجل تحقيق الفوائد المتوقعة من الحصول على حل ، ولا نريد حلا نظريا على صفحات الورق فقط ، لا يصمد نقله إلى أرض الواقع ، كأن يكون غير ملائم أو غير معقول من وجهة النظر التطبيقية .

بقي أن نقول : إن الهدف من هذا المقرر تعليمي ، ولذلك فإن ما سنكتبه من برامج في الفصول اللاحقة لا يشترط أن تمر بجميع الخطوات السابقة ، إذ إننا نهدف إلى تعليم لغة فيجوال بيسك من خلال برامج قد تطول أو تقصر حسب الضرورة والهدف ، فالتركيز إذا ليس على حل مسألة محددة من الألف إلى الياء . ويجدر بالذكر كذلك أن حل مسائل عملية كثيرة يرقى إلى مستويات وأحجام بحيث يمكن تسميته " إعداد نظام متكامل لتلك المسألة " ، والذي غالبا ما تبرز فيه روح الفريق ، بمعنى أن يتعاون فريق من عدة أفراد لإتمام تصميم وتنفيذ نظام متكامل لمسألة معينة تطبيقية كبيرة ويمكن تمييز عدة طرق مستخدمة في كتابة البرامج ، من أهمها : أ . البرمجة من أعلى إلى أسفل ( TOP - DOWN PROGRAMMING ) . ب . البرمجة بالأجزاء ( MODULAR PROGRAMMING ) . ج . البرمجة بالتشفير المركب ( STRUCTURED CODING ) ولكون هذا المقرر غير مخصص للتوسع بهذه المواضيع ، نأتي على ذكر معلومات إضافية عن طريقة واحدة فقط وهي الطريقة الأولى



تصور أن لديك مسألة تريد كتابة برنامج لحلها بواسطة الحاسوب ، فإذا كانت المسألة كبيرة ، فقد نحتار في كيفية البدء ، ومن أين ، إذا لنحاول تقسيم المسألة إلى عدة مسائل فرعية ، وإذا كانت المسائل الفرعية الناتجة ذات حجم مناسب بحيث تسيل برمجتها ، نقوم ببرمجة كل مسألة ، ونجمع البرامج الناتجة بأسلوب معين ، لينتج برنامج المسائل الأساسية . بالطبع قد نلاحظ أن مسألة من المسائل الفرعية الناتجة أو أكثر هي من حجم يستحق التجزئة لي مسائل اصغر . وفي الحقيقة لا بوجل تستمر بعملية التقسيم هذه حتى الانتهاء بأفرع صغيرة لمسائلنا ، تكتب برامجها ببسر وسهولة ، يلي ذلك مرحلة تجميع البرامج الصغيرة بأسلوب مناسب حتى الحصول على برنامجنا المقصود وفيما يلي بعض الأمثلة التوضيحية على أسلوب التجزئة هذا : ما يمنع مثال ( 1 ) تريد إيجاد معدل ثلاثة أختبارات ، حيث يكون لكل اختبار وزنه الخاص به وطباعة المعلومات المعطاة ، والمعدل الذي تجده . الحل : يمكن النظر إلى البرنامج على أنه جزء واحد يسمى البرنامج الرئيسي ، وعند التفكير بكتابته يمكن أن نميز عدة أجزاء فرعية له كما يبدو في الشكل ( 1 ) التالي :

بعد ذلك يمكن أن نهتم بكل جزء على حدة ، ونركز عليه بغية كتابة جمل البرمجة التي تؤدي إلى الغرض المطلوب في هذا الجزء ، فمثلا جزء الدخل البيت ) يمكن وضع تفصيل له كما يلي أدخل البيانات - اقرا درجة الاختبار الأول ووزنه - اقرا درجة الاختبار الثاني ووزنه - اقرا درجة الاختبار الثالث ووزنه . أما جزء ( أطبع البيانات ) ، فيمكن تفصيله كما يلي - أطبع البيانات الكتب عناوين مناسبة أكتب درجة الاختبار الأول ووزنه - الكتب درجة الاختبار الثاني ووزنه - اكتب درجة الاختبار الثالث ووزنه \*\* أما الجزء الثالث ، فلا يحتاج إلى تقليل كثيرة ، كما يلي أحب المعدل المعدل - ) الدرجة الأولى \* وزنها الدرجة الثانية \* وزنها + الدرجة الثالثة وزنها ( مجموع الأوزان الجزء الأخير تذكر الأتية أطبع المعدل - اكتبه علوانا ملاسماء الكتب المعدل . مثال ( 2 ) تقع مدرس في قسم الحاسوب طريقته الخاصة اصدار قائمة بأسماء الطلبة الذين يحضرون امتحاناته ، وقائمة بأسماء المتخلفين عنها ، وفي طريقته الخاصة هذه يستخدم جهاز الحاسوب المتنقل الخاص به ، والذي بلفظ عليه بزانة مجا ، للقيام بهذه المهمة ، فقط يطلب من طليته أنه على كل من يدخل قاعة الامتحان أن يتخل معه عن

**2.2 مخططات سير العمل ( Flowcharts )** عندما يمتلك القارى معلومات أساسية جيدة عن لغة برمجة معينة ، فإن بإمكانه كتابة برامج لحل مسائل سهلة وذات حجم صغير ، وذلك دون المرور بخطوات تمهيدية . أما عندما يزداد حجم المسائل التي نريد كتابة برامج لها ، وبالتالي يزداد حجم البرنامج نفسه ، فإن عملية تتبع منطق الحل من البداية حتى النهاية يزداد تعقيداء ولذا ينصح في كثير من الأحيان باستخدام رسومات مساعدة لتوضيح خطوات العمل عند حل مسألة معينة . والشكل الناتج من هذه الرسومات يسمى مخطط سير العمل ( Flowchart ) . وهذه الخطوة تعتبر توطئة لكتابة البرنامج ، علما أنه ليس من الضروري المرور بها إلا عند الرغبة ، وعندما تكون مناسبة لنوع المسألة . وتستعمل

في هذه المخططات رموز ذات دلالة محددة سبق وأن درستها في مقرر سابق ومستورد الآن بعض الأمثلة لتوضح استخدام مخططات سير العمل هذه ، وذلك من باب التذكير وإنعاش المعلومة الدين الدارس ، مثال ( 3 ) فيما يلي مخطط سير عمل يتم فيه بحال قيمتين لمتغيرين ، ثم حسان مجموعهما وحامل ضربهما ، وطباعة النتائج . البداية اقرا او الحبه == ابن القيم السفة في د , H المجمل هو حاصل الضرب بالو : ١ الشكل ( 3 ) : مخطط سير العمل تمثال رقم ( 3 ) مثال ( 4 ) فيما شي مخطط سير عمل يتم فيه دخل مبلغ من العمل بعمل ما يملكه مسلم بر لعبه في تأدية زكاة ماله ، ثم يصف مقدار الزكاة التي تبلغ 2.5 % من المبلغ ، ويطيع

عهده القامى اما 1 م و فن ، == اهل الميت را كه مادة الى ( 4 / مقطط سير العمل لمن رقم ( 4 ) الشكل ( 5 ) : مخطط سير السل تمثال رقم ( 5 ) من العذاب أن تمتنع هذا نتيجة مهمة توضح احد مسارات التحكم في تغية تير بامج ، وهو من المفتي او النمل ( Sauumntial ) في تمد البرنامج ، وهذا من تحت في ماي 3 ، 4 أعد بحث با نان ثم انتقل إلى الخطوة التالية والتي لها وقا مثل شليز ومثل شي بعت لنهاية أدى التدقيق في من ؟ 5 اعلا نستني أننا استغنا : اتخذ قرار من أجل اللي على عدة لغتيران ، حيث يتم اختيار أحدهما النقد استيا إلى حالة شرطية مشددة ، وهذا النوع من التحكم في تغطية حمل البرنامج يستخدم بما يطلق عليه تراكيب الاختيار ( Salatian Fantaames ) في أغة البرمجة ، وسوف تاتي اي تفاصيل هذه اتر لقيب أي قسم لاحق من هو الوحدة الثالثة والنهاية الكورية و العملة الي أن النية ار تي عمل ام لانه لم يتم إيدائها لشرح ، تم كتابة القوه مثال ( 6 ) مخطط سير العمل انتي برشح طريقة ايجاد معطل 3 قيمة عقدية تباعا إلي البرنامج وإخراج القيم الثري و محتتهاء

الشكل ( 6 ) : مخطط سير العمل المثال رقم ( 6 ) ولدى التدقيق في مثال 6 أعلاه نستنتج أننا استخدمنا فكرة العداد ، وتتالت الخطوات حتى وصلنا إلى خطوة اتخاذ قرار فيما إذا كان العداد أكبر من أو يساوي 20 أم لا . وفي كل مرة لا يتحقق فيها هذا الشرط ، يكون الجواب عن التساؤل هو لا ، حيث نعود إلى خطوة سابقة من أجل الإعادة ، وتكرر الإعادة ما دام الجواب على التساؤل هو لا ، وهذا يعني أننا لم نكمل قراءة القيم العشرين وإضافتها إلى المجموع . وبمجرد قراءة آخر قيمة وإضافتها إلى المجموع ، نلاحظ أن قيم العداد أصبحت مساوية 20 ، ويصبح جواب التساؤل هو نعم ، عندها يصبح قرارنا بأن لا ضرورة للإعادة وما علينا إلا أن نقسم المجموع على 20 لإيجاد المعدل وطباعته ثم إلى النهاية ، إن هذا النوع من التحكم في تنفيذ جمل البرنامج يستخدم ما يطلق عليه تراكيب التكرار ( Repetition Structures ) في لغة البرمجة ، وسوف نأتي على تفاصيل ذلك في بند لاحق من هذه الوحدة

3.2 أشباه البرامج ( Pseudocodes ) بعد أن قدمنا في البند السابق طريقة استخدام مخططات سير العمل كخطوة تمهيدية تساعد في الوصول إلى كتابة البرنامج ، نقدم في هذا البند طريقة أخرى تساعد في تحقيق نفس الغاية ، ألا وهي طريقة كتابة شبه برنامج ( Pseudocode ) لحل المسألة . وشبه البرنامج هو وصف لحل المسألة باللغة العادية ، وقد يتخلل هذا الوصف بعض الأجزاء أو الأسطر التي تقترب من تراكيب لغة البرمجة ، ويقل ذلك أو يكثر رغبة المبرمج ، وكلما اقتربنا من استخدام وصف مشتق من تراكيب لغة البرمجة ، وابتعدنا عن اللغة العادية الإنشائية ، كلما اقترب الوضع إلى

البرنامج المنشود ، ومن المهم أن نلاحظ هنا أنه لا توجد ضوابط محددة ملزمة للمبرمج عند كتابته لشبه البرنامج ، وتكرر الآن الأمثلة السابقة من 3 إلى 7 ونكتب شبيه برنامج لكل منها : حسب

3 تراكيب الاختيار Selection Structures كما ذكرنا في البند السابق ، فإن لغة VB6 توفر تراكيب تحكم في البرنامج ، ومن ضمنها تراكيب الاختيار ، ونخصص هذا البند لشرح هذه التراكيب وبعض الأفكار الأخرى ذات العلاقة . 1.3 تركيبة IF THEN إن الشكل العام لهذه التركيبة هو كما يلي : If / Condition THEN Statement حيث Condition : هو شرط منطقي بصور متعددة ، تكون قيمته أو الجواب عنه إما صواب ( True ) أو خطأ ( False ) ، وقد يكون هذا الشرط بسيطاً ( Simple ) أو قد يكون مركباً ( Cornipound ) والذي ينتج من ربط شرطين بسيطين أو أكثر بإحدى أدوات الربط مثل OR ، AND وووووووو Staleilent : جملة من جملة VB6 . IF THEN : كلمتان محجوزتان ( Reserved Words ) في لغة VB6 . وعندما يكون جواب الشرط صوابية ، يتم تنفيذ الجملة بعد THEN ثم الانتقال مباشرة إلى ما بعد هذه التركيبة ، وعندما يكون الجواب خطأ ، لا يتم تنفيذ الجملة بعد them ، بل يتم الانتقال مباشرة إلى ما بعد هذه التركيبة ، ومخطط سير ( 11 ) التالي يوضح ما ذكرناه : العمل في

قد تأخذ هذه التركيبة صورة أخرى كما يلي : IF Condition THEN Statement 1 Statement k END IF وفي هذه الحالة تنفذ كتلة الجمل من Statement 1 إلى Statement k في حالة كون قيمة الشرط صواباً ، ثم إلى ما بعد END IF ، في حالة كون قيمة الشرط خطأ ، فلا تنفذ هذه الكتلة أبداً ، وينتقل التنفيذ إلى ما بعد END IF ، ومخطط سير العمل في الشكل ( 12 ) التالي يوضح ذلك .

/ 2.3 تركيبة IF THEN / ELSE / ENDIF إن الصورة العامة لهذه التركيبة هي كما يلي : IF Condition THEN Statement ( s ) 1 ELSE Statement ( s ) 2 END IF حيث يتم تنفيذ مجموعة الجمل 1 في حالة كون قيمة الشرط صواباً ، ثم الانتقال إلى ما بعد ENDIF ، أو يتم تنفيذ مجموعة الجمل 2 في حالة كون قيمة الشرط خطأ ، الانتقال إلى ما بعد ENDIF ، والمخطط في الشكل ( 13 ) التالي يوضح هذه التركيبة :

3.3 تركيبة SELECT CASE / END SELECT إن الصورة العامة لهذه التركيبة هي كما يلي :  
: SELECT CASE Experssion CASE Value 1 : Statement ( s ) 1 CASE Value 2

Statement ( s ) 2 CASE Value k : Statement ( s ) k CASE ELSE Statement ( s ) J  
END SELECT

4.3 استخدام دوال في عملية الاختيار هناك مجموعة من الدوال تستخدم كذلك في عملية القرع والاختيار ، لذكرها فيما يلي الدالة IIF : أن الصورة العامة لاستخدام هذه الدالة هي كما يلي : ( TIF CONDITION , CHOICE1 , CHOICE2 ) ويكافئ ذلك تركيبة IF THEN ELSE التالية : IF CONDITION THEN CHOICE 1 ELSE 2 CHOICE 2 ENDIF حيث يتم تقييم الشرط ( CHOICE 1 ، فإذا كانت القيمة صواب يتم تنفيذ CHOICE 1 ، وإذا كانت قيمة خطأ يتم تنفيذ 2

**CHOICE** ب . الدالة **SWITCH** : إن الصورة العامة لاستخدام هذه الدالة يكون في جملة تخصيص ، وذلك كما يلي **VARIABLE = SWITCH ( CONDITION 1 , CHOICE1 , CONDITION2 , CHOICE 2 , CONDITION K , CHOICE K )** ويمكنك عزيزي الدارس مقارنة استخدام هذه الدالة مع استخدام تراكيب متداخلة من تركيبية

**IF / THEN / ELSE / END IF** التجذ أنهما متكافئان الدالة **CHOOSE** : أن الصورة العامة لاستخدام هذه الدالة تكون ضمن جملة تخصيص ، وذلك كما يلي : **VARIABLE = CHOOSE ( CHICICEn ( PARAMETER , CHOICE1 , CHOICE2 , ... , PARAMETER : من نوع INTEGER و مترادف قيمه مع الخيارات التي تليه في الجملة المذكورة ، وتبدأ هذه القيم بالقيمة القرانف تنقيد 1 CHOICE ، ثم 2 لترادف تنفيذ 2 CHOICE ، وهكذا . CHOICE 1 إلى CHOICE : الخيارات التي يتم تنفيذ واحد منها استنادا إلى القيمة التي باخذها PARAMETER**

**F [** و تراكيب التكرار **Repetition Structures** من تراكيب التحكم التي توفرها لغة **VB6** ، تناقش في هذا القسم تراكيب التكرار ، وذلك للتمكن من التعامل مع الحالات البرمجية التي يمر من مها إعادة أو تكرار جزء محنت من الجمل مرت عنيدة 1.4 تركيبية **FORNEXT** ان الصورة العامة لهذه التركيبية هي كما يلي **FOR VAR = EXP1 TO EXP2 STEP R " STATEMENT 1** **STATEMENTN NEXT VAR** مع ملاحظة ما يلي **R ( ) ( EXPI : VAR** متغير على العدد الذي يتم وفقا اقيمه تكرار مكون من الجمل التي بين **FIR** و **NTXT** ) تعبير تحدد أهدته القيمة الألمانية ( **INITIAL VALLE** ) تعداز **VB - EXP** : تعبير تحنن قعله التهمة النهاية ( **FINAL VALLE** ) العدد **WAR** قيمة محددة تمثل التغير في البيعة العداد **٩٧٠٠** بعد كل مرة تكرار . أم إن **STEPR** اختيارية في حالة كون **R = ١** ان لكر **VAR** بعد **NEXT** في السطر الأخير أختهاري . الكلمات **STEI , NEXT , TO , FOR** هي كلمات متاحة محجوزة في لغة 16 . ( المخطط في الشكل ( 14 ) يوضح عمل تركية **FOR NIXT**

## 2. مفهوم البرمجة المرئية

من المعروف أن اللغات التقليدية التي لا تعتمد أسلوب البرمجة المرئية) تستخدم النصوص لبرمجة الحاسوب، أما لغات البرمجة المرئية فيمكن أن تستخدم الرسم والرسومات وواجهة رسومية (Graphical User Interface: GU1) لإصدار تعليمات للحاسوب، ومن التوضيحات المنطقية لهذا الأسلوب، أن الإنسان يشاهد ما حوله من الظواهر المختلفة على شكل صور، ثم تتم ترجمة ما يرى إلى نصوص معبرة عنها، وبالتالي، لماذا لا تعطي هذه الصور كتعليمات للحاسوب بدلا من المرور

حلة نقلها إلى نص. كما أن كثيرا من التطبيقات العلمية وغيرها، وبرامج التعلم التفاعلية تحتاج إلى البيئة المرئية لتقوم بدورها بفاعلية.

إن هذا النوع من البرمجة يوسع دائرة استخدام الحاسوب، ويزيد من قوة وفاعلية البرمجة ويطورها، وقد شهدت العقود الثلاثة الأخيرة تطورات واسعة في هذا الميدان، ويمكن القول إن حقل البرمجة المرئية نما وترعرع نتيجة تمازج رائع بين

هي: اللغات البرمجة (Programming Languages)

الرسم بالحاسوب (Computer Graphics)

تفاعل الإنسان مع الحاسوب (Human Computer Interface ( HCI))

ولنستعرض التعريفات التالية وصولاً إلى تعريف لغة البرمجة المرئية:

أيقونة: كينون ذو معلولين: المدلول المنطقي (Logical) وهو المعنى المقصود منه، والمدلول الفيزيائي (Physical) وهو الصورة التي تمثلها.

نظام أيقونية: مجموعة تركيبية من أيقونات ذات علاقة بعضها مع بعض.

جملة أيقونية: تنظيم من الأيقونات من نظام أيقوني.

لغة البرمجة المرئية: مجموعة من الجمل الأيقونية بنيت بقواعدية (Syntax) يرتبط معها تحليل لهذه الجمل لتحديد تركيبها القواعدي، ومعنى (Semantic) يرتبط معها تحليل لهذه الجمل لتحديد المعنى المقصود منها.

وفي لغات البرمجة المرئية، يمكن أن نميز بين نوعين من الأيقونات هي:

أيقونات العمليات (Process Icons) تدل على عمليات أو حسابات من نوع ما.

الأيقونات الكينونية (Object Icons)، تتكون بدورها من نوعين هما: البسيطة (Elementary)، التي تمثل كينونات أساسية، والمركبة (Composite)، التي تمثل كينونات مركبة، يتم تكوينها من عدة أيقونات بسيطة محددة.

إن البرمجة المرئية هي عبارة عن أسلوب حديث نسبياً للبرمجة، تستخدم فيه برامج مساعدة لتصميم واجهة استخدام رسومية (الأزرار والنصوص...) وربطها بالشفيرة البرمجية (Code)، وتسمى هذه البرامج المساعدة بيئة التطوير المتكاملة (Integrated Development Environment : IDE)، ويستخدم هذا النوع من البرمجة تعبيرات مرئية ( Visual Expressions) في عملية البرمجة، أو تعالج معلومات مرئية، أو قد تدعم التفاعل المرئي بين المستخدم والحاسوب.

في السابق كانت البرامج تستخدم ما يسمى بسطر الأوامر (Command Line)، فكان يظهر البرنامج في صورة نصوص من عدة أسطر، ويمكن القول إن لغات البرمجة التي تعتمد النصوص فقط صعبة على العديد من المستخدمين. أما الآن مع الواجهات الرسومية، فتظهر أمام المستخدم مجموعات من الأزرار والخيارات والقوائم وغيرها، ولا يمكن للبرنامج أن يتوقع ما الذي سيحدث في الخطوة التالية، لذا فإن البرنامج يقسم إلى عدة أجزاء، ولكل جزء وظيفة محددة، ينفذ كل منها عند تنفيذ

بالحدث (Event). فعلى سبيل المثال، تعتبر نقرة زر الفأرة

حدثاً، وضغط أحد المفاتيح يعتبر حدثاً، والاتصال بالإنترنت يعتبر حدثاً، كل هذه تعتبر أحداث، وتسمى الدالة التي تعمل عند حدوث الحدث بالدالة المرتبطة بالحدث. ومما يعزز أهمية البرمجة المرئية كذلك أن الكثير من الناس يستخدمون فكرة الصور (Pictures) في تفكيرهم وتوضيحاتهم، وأن كثيراً من التطبيقات تتوافق كثيراً مع استخدام الرسومات والواجهات الرسومية، وبأسلوب آخر فإن

العقل البشري سريع التأثر بالصورة والتعامل معها، وربما نقلت صورة معلومات أكثر بكثير مما تنقله النصوص، كما أن هذا النوع من البرمجة يسهل عملية البرمجة لغير المختصين مما يوسع انتشار الحواسيب وبرمجتها.

ومن أشهر بيئات التطوير الرسومية

Visual C++

Visual j++

Delphi

Visual Basic

Visual Basic.Net

Java Builder

وغيرها الكثير، وتستخدم هذه البرامج نسخة محسنة من لغات البرمجة العادية وتدمجها في بيئة التطوير الخاصة بها، لذلك فإن Delphi مثلا ليست لغة برمجة بمعنى الكلمة، إنما هي بيئة تطوير تستخدم نسخة محسنة من Pascal تتميز باستخدام الكائنات (Objects) وميزات أخرى.

وتعرف لغة البرمجة المرئية بأكثر من أسلوب، فيمكن القول إنها لغة تستخدم الفأرة (Mouse) والأيقونات (Icons) والرموز التي على الشاشة وقوائم الاختيار (Menus) لإعداد أو تطوير البرامج. كما يمكن القول إنها لغة تستخدم التمثيل المرئي (Visual Representation) لكنونات منطقية (Objects Logical) لمعالجة معلومات مرئية (Visual Information) وتدعم التفاعل المرئي في عملية البرمجة (Visual Interaction) وتستخدم التعبيرات المرئية (Visual Expressions).

3.مزايا البرمجة المرئية

تصف عملية البرمجة المرئية بمزايا إيجابية منها:

/\* أفكار البرمجة فيها أقل من غيرها.

/\* التماسك الكبير بين أجزائها.

/\* العلاقة بين أجزائها تظهر بشكل صريح.

/\* مشاهدة النتائج المرئية بشكل فوري وسريع

/\* لا يشترط تصور البرامج فيها بشكل متسلسل.

/\* يمكن الاستغناء عن خطوات أو مراحل وسطية.

/\* اهتمام أقل بالأمر القواعدية

/\* للبرامج فيها هيكلية قابلة للاستعراض (Navigable) .

/\* يمكن تنفيذ أجزاء محددة من البرامج.

/\* تكاملية القواعد الصورية لها.

/\* وجود مكونات (Components) قابلة لإعادة الاستخدام، حيث تدمج مع غيرها الإنتاج نظام تطبيقي كامل

وبالرغم من وجود كل هذه المزايا، فلا يخلو الأمر من وجود سلبيات، مثل:

أ. حاجة البرنامج لمساحة كبيرة لعرض أجزاءه الرسومية.

ب. وجود إشكالية في المسميات لبعض الأجزاء الرسومية، كأن نختار في تسمية

الجزء (Stop) على أنه إجراء (Action) أمر بالتوقف، أو أنه كينون (Object) الإشارة توقف مثلا World .Pygmalion ding

#### 4. بعض لغات البرمجة المرئية

يمكن تصنيف لغات البرمجة المرئية إلى عدة أصناف كما يأتي:

(أ) اللغات المستندة إلى النماذج (Form - Based): يلاحظ هذا النوع في التعامل مع الجداول الإلكترونية واستخداماتها، وتستخدم للتأثير على مجموعة من الخلايا المترابطة، ومثال ذلك لغة Forms/3.

(ب) اللغات الخليطة النصية (Textual و المرئية Visual): كما يشير الاسم فإنها تمزج بين النص والصورة، ومثال ذلك لغة C ، C++ ، Rehearsal World.

(ج) لغات البرمجة بالمثال (Example- Programming- by): من خلالها يزود المستخدم النظام بكيفية أو آلية أداء مهمة محددة، ومثال ذلك لغة Rehearsal

World ولغة Pygmalion

(د) اللغات الموجهة بالقيود (Constraint- Oriented): يتعامل هذا النوع مع كينونات فيزيائية في البيئة المرئية ضمن قيود محددة تكافئ قوانين طبيعية، وتناسب هذه اللغات عملية التصميم بالمحاكاة (Simulation Design)، ومثال ذلك لغة Thinklab، ولغة Reality Kit Alternative (ARK)

(هـ) لغات البرمجة المرئية البحتة أو الكاملة (Pure Visual): يدعم هذا النوع من اللغات عملية البرمجة بالصور بشكل كامل، ومثال ذلك برمجيات ( Lap , Pict/D Cube , Prograph , View ) ويجدر بالذكر أن البرنامج هنا ينتج كاملا من خلال معالجة الأيقونات ورسومات أخرى، وتتم ترجمته (Compiled) مباشرة من خلال التمثيل المرئي، ويرتبط بها مصطلح الرسومات التنفيذية (Executable Graphics)، كما يمكن القول بأن لغات مثل (Visual Basic Visual C++) لا تعتبر لغات برمجة مرئية بحتة، بل هي في الأغلب لغات نصية معرفة ومنظمة بشكل جيد، تدعم الرسومات وتستخدم واجهة استخدام رسومية (GUI) لتنتج برمجة منسقة جذابة ومريحة للمبرمج والمستخدم

ومن الناحية التاريخية، يمكن تلخيص ظهور مجموعات من لغات البرمجة المرئية بما يأتي:

أ. البدايات الأولى (1966-1980): ظهرت خلال هذه الفترة مجموعة من اللغات مثل:

Plan 2D

Tiner

Ambit

Outline

Grail

Pygmalion

Auto programmer

Query by Example

(ب) أنظمة مرئية (1981-1990): ظهرت خلال هذه الفترة مجموعة أخرى من الأنظمة المرئية مثل:

ML- Like VL

Hi- Visual

Prograph

Tinker Toy

Pict

PC- Titels

GAL

C..2

formal

Thinklab

PIGS

Sun PICT

SmallStar

ARK

Show and Tell

pGiumoP

(ج) المرحلة الحديثة (1991-): ظهرت خلال هذه الفترة مجموعة إضافية من الأنظمة المرئية مثل:

Cube

CODE 2.0

Form/3

Hyper Pascal



Cantata  
Iconicode  
AVS  
Vampire  
Hence 1.4  
Viasvis  
Mondrian  
VIPR  
Scheme Paint  
MViews  
Chem Train  
SPE

أيقونة Icon: كينون ذو مدلولين: المدلول المنطقي (Logical) وهو المعنى المقصود منه، والمدلول الفيزيائي (Physical) وهو الصورة التي تمثلها.

- البرمجة المرئية Visual Programming : أسلوب حديث نسبيا للبرمجة، تستخدم فيه برامج مساعدة لتصميم واجهة استخدام رسومية (الأزرار والنصوص...) وربطها بالشفيرة البرمجية (Code)، وتسمى هذه البرامج المساعدة ببيئة التطوير المتكاملة (Integrated Development Environment : IDE)، ويستخدم هذا النوع من البرمجة تعبيرات مرئية ( Visual Expressions) في عملية البرمجة، أو تعالج معلومات مرئية، أو قد تدعم التفاعل المرئي بين المستخدم وجهاز الحاسوب.

بيئة التطوير المتكاملة Integrated Development Environment: وهي بيئة لاحتضان عملية البرمجة، حيث يتوفر فيها مجموعة من الأدوات المختلفة والنماذج التي توفر للبرمجة تسهيلا كبيرا في بناء تطبيقات البرمجة.

- تفاعل الإنسان مع الحاسوب Human Computer Interaction: رغم أنه لا يوجد اتفاق على تعريف واضح لهذا الموضوع، فإنه يمكن القول بأنه حقل يعني بتصميم وتقييم وتنفيذ أنظمة الحسابات التفاعلية الموجهة لاستخدام الإنسان، كما يعني بدراسة مختلف الظواهر المحيطة بهذه الأنظمة. التعبير المرئي Vistral Expression: استخدام نماذج مرئية مختلفة، خاصة الصور، للتعبير عن معلومة معينة مثل الشعارات (Logos)، بما في ذلك استخدام الألوان والحركة.

- التفاعل المرئي Visual Interaction: تفاعل الإنسان مع الحاسوب دون اللجوء إلى الأسلوب الكتابي، لكن باستخدام الإشارة أو تعابير بعض أعضاء الإنسان كالعين وغيرها، للطلب من الحاسوب المعلومة المطلوبة.

الرسم بالحاسوب Computer Graphics: استخدام الحاسوب من معدات وبرمجيات الإنجاز رسومات متنوعة وجذابة.

قواعد تراكيب اللغة Syntax: مجموعة من التراكيب الكتابية جمل اللغة، يرتبط معها تحليل لهذه الجمل لتحديد تركيبها القواعدي وصحة توافقه مع تراكيب اللغة.

- لغة البرمجة المرئية Visual Figraturing Language : مجموعة من الجمل الأيقونية بنيت بقواعدية (Syntax) يرتبط معها تحليل لهذه الجمل التحديد تركيبها القواعدي، ومعنى (Semantic) يرتبط معها تحليل لهذه الجمل لتحديد المعنى المقصود منها.

- المعلومات المرئية Vistral Information : معلومات تأتي على شكل مصور مرئي ناتجة عن التحليل المرئي للبيانات، مما يزيد في فهم العلاقة بين المصادر المتوفرة والأداء المتعلق بها.

معني تراكييب اللغة Sexuantic: المعنى المرتبط بتراكيب جمل اللغة يحدد ما يفترض أن تؤديه الجملة، ويرتبط بتحليل لهذه الجمل لتحديد المعنى الذي يقصد منها.

- واجهة المستخدم الرسومية Graphical User Interface GUI: شاشة تبدأ بها الكثير من البرمجيات الحديثة وتفرعاتها، وتوفر للمستخدم مجموعة من الأيقونات والأدوات والخيارات التي تسهل عليه كثيرا عملية استخدام البرمجية.

2. نسخ مختلفة من VB6

كأي لغة من لغات البرمجة، تعتبر VB6 من أحد إصدارات لغة فيجوال بيسك (Visual Basic) والتي هي لغة برمجة النوافذ (Windows Programming Language)، من إنتاج شركة مايكروسوفت وهناك عدة نسخ متوفرة من VB6 نفسها، تختلف فيما بينها بما توفره كل منها من مواصفات وخصائص وإمكانات للمبرمج من خلال بيئة التطوير المتكاملة (IDE)، فقد أصدرت مايكروسوفت النسخ التالية:

أ. النسخة التعليمية (Learning Edition)

توفر للمبرمج إمكانات البرمجة الأساسية.

ب. نسخة الاحتراف (Professional Edition) توفر إمكانات النسخة السابقة، إضافة إلى مجموعة جيدة أخرى من الإمكانيات، وهي الأكثر شيوعا واستخداما من المبرمجين.

ج. النسخة المميزة (Enterprise Edition)

توفر جميع الإمكانيات كسابقتيها، إضافة إلى مجموعة إضافية، مما يجعل منها لغة تصلح لتطوير برمجيات وأنظمة على مفتوي واسع، وحسب متطلبات المستخدمين على أرض الواقع والتطبيق.

ورغم اختلاف هذه النسخ بما يتوفر فيها من خلال بيئة التطوير المتكاملة (IDE) الخاصة بها، فإن هذه البيئة تسهل على المبرمج إنتاج البرنامج بجهد أقل ووقت أقل كذلك، ويتم كل هذا من خلال واجهة تطبيق رسومية (Graphical User Interface)، وبأسلوب تفاعلي (Interactive) مما يسهل عملية البرمجة ويجعلها أكثر تشويقا.

3. الملفات والمشاريع (Files and Projects)

ينظر إلى برنامج فيجوال بيسك على أنه مشروع (Project)، يتكون بدوره من مجموعة من الملفات. ونفترض أن تكون لغة VB6 محملة على جهازك، وإن لم يكن الأمر كذلك، فعليك توفيرها على جهازك.

1.3 تشغيل لغة VB6

نقوم بتشغيل برنامج لغة فيجوال بيسك بالأسلوب المعتاد لتشغيل البرمجيات المشابهة، هكذا:

1. انقر "ابدأ" (Start) من شريط الحالة

2. Status Bar. انقل المؤشر إلى البرامج (Programs).

3. ابحث عن أيقونة VB6، وانقر عليها، لتلاحظ ظهور شاشة أو نافذة مشروع جديد

كما يظهر في الشكل (1). تحتوي هذه النافذة على أيقونات عديدة تمثل كل منها نوع المشروع بملفاته المترابطة، وهذا المشروع بملفاته يمثل برنامج VB6 كما ذكرنا، وتستخدم كل أيقونة للتعامل مع مشروع بإمكانات مختلفة. تفتح هذه النافذة عادة بحيث يشير الاختيار إلى الأيقونة Standard EXE وهي ما سوف نكمل نقاشنا باستخدامها هنا.

2.3 محتويات نافذة مشروع جديد

يمكن ملاحظة أن محتويات نافذة مشروع جديد هي كما يلي: أ. أيقونات مختلفة للتعامل مع مشاريع VB6 إمكانيات مختلفة، ومنها كما قلنا

Standard EXE ومنها Active EXE ، وغيرها الكثير .تفقد النافذة لتتعرف عليها.

ب. ثلاثة أقسام (Tabs) وهي:

-New: لإنشاء مشروع جديد.

-Existing: لفتح مشروع موجود ومخزن.

-Recent: لفتح مشروع موجود، حيث تظهر الملفات حسب وقت إنشائها الأحدث أولاً.

ج. صندوق اختيار (Checkbox)، مكتوب بجانبه ما يلي:

Don't show this dialog in the future.

حيث باختباره، لا تعود نافذة مشروع جديد المبينة في شكل (1) تظهر عند تشغيل VB6.

د. ثلاثة أزرار (Buttons)، وهي:

Open: لفتح نوع مشروع، مثل Standard EXE ، وذلك بالنقر على نوع المشروع، ثم النقر على هذا الزر.

Cancel: يغلق نافذة مشروع جديد، وذلك بالنقر عليه Help - .يفتح المساعدة المباشرة On\_line

لنستمر الآن في التعرف على مكونات بيئة التطوير IDE، حيث سنستخدم نوع المشروع Standard EXE كما ذكرنا عند فتح مشروع جديد من هذا النوع، وذلك بالنقر المزدوج على النوع، أو بالنقر عليه مرة واحدة ثم النقر على Open، هكذا تفتح لنا شاشة أخرى ضمن بيئة التطوير IDE، وهي المبينة في الشكل (2)، وهي غنية بعدة نوافذ ومكونات وأشرطة خيارات وأدوات مختلفة وجدت لخدمة المبرمج وتنفيذ طلباته

3.3 مكونات شاشة مشروع تم إنشاؤه

بعد إنشاء مشروع جديد وظهور الشاشة المبينة في الشكل (2)، يمكننا تمييز الأجزاء والمكونات التالية:

1. نافذة 1- Form 1 ( Form 2) .Project

2. نافذة Form Layout .

3. نافذة Properties- Forml .

4. نافذة 1 - Project - Project

5. صندوق الأدوات Tool Box

6. شريط الخيارات Menu Bar

7. شريط الأدوات Tool Bar

فيما يلي نبذة مختصرة عن كل من هذه الأجزاء:

1. نافذة 1- (Form -) Project .Forml

تحتوي هذه النافذة نموذج (Form) يسمى 1

Form في هذا الجزء يتم عرض الجزء المرئي (Visual) من المشروع وهو الجزء المعروف باسم واجهة التخابط الرسومية (GUI) وما يمكن أن تحويه من أزرار وقوائم وغيرها، وهنا أيضا يدخل المستخدم مدخلات (Input) البرنامج، كما ينتج البرنامج النتائج المتوقعة output هنا كذلك.

## 2. نافذة Form Layout

يحدد المبرمج فيها مكان النموذج (Form's Position) على الشاشة أثناء تنفيذ البرنامج، وكما يبين الشكل (4) فتظهر صورة الشاشة بداخلها صورة النموذج، وبإمكانك النقر والسحب على صورة النموذج لتضعها في المكان الذي تريد.

## 3. نافذة الخصائص Properties- Form

وتعرض هذه النافذة خصائص النموذج أو أداة التحكم، مثل الحجم (Size)، نوع الخط (Font Style)، اللون (Color)، مكان النموذج (Position)، ومن خلالها يمكن تحديد هذه الخصائص سواء للنموذج أو الأدوات التحكم.

ويجدر بالذكر أن النموذج وأدوات التحكم قد تشترك في بعض الخصائص، مثل العرض (Width) والارتفاع (Height)، كما تنفرد كل منها بخصائص أخرى، الشكلان (6أ)، (6ب) يظهران نافذة الخصائص مصنفة بأسلوبين: إما مرتبة هجائيا (Alphabetically) أو وفق مجموعات (Categorized)، وبالإمكان التغيير من أسلوب الآخر باستخدام أحد الأقسام (Tabs) المبينة في أعلى شاشة الخصائص.

## 4. نافذة Project - Project1

تجمع هذه النافذة ملفات المشروع الواحد وتبينها وفقا لنوعها (Type)، وتسمى كذلك مستكشف المشروع (Project Explorer).

كما تحتوي نافذة المشروع ثلاثة أزرار:

أ. زر رؤية البرنامج (View Code): وعند النقر عليه تفتح نافذة تمكن من كتابة البرنامج أو رؤيته عندما يكون مكتوبا.

ب. زر رؤية الكينون (View Object): وعند النقر عليه يتم عرض النموذج (Form)، ويجدر بالذكر أن تفعيل واستخدام (View Co(de)، (View Object) يستوجب اختيار النموذج Form 1، كما يظهر في الشكل (7).

ج. زر إظهار وإخفاء دليل النماذج (Toggel Folders Button): وبالنقر عليه بالتتابع، مرة يظهر دليل النموذج (Formns Folder)، والأخرى يخفيه، ودليل النماذج يحتوي جميع النماذج: من نموذج واحد في البرامج البسيطة، إلى عدة نماذج في البرامج أو المشاريع الأكبر.

## 5. صندوق الأدوات Tool Box

يحتوي مجموعة من أدوات التحكم في النموذج، وهذه الأدوات قد تكون: أزرار (Buttons)، صناديق اختيار (Check Boxes)، صندوق نص (Text Box) صندوق قوائم (List Box)، إطارا (Frame). لاحظ أن وضع مؤشر الفأرة على إحدى هذه الأدوات يؤدي إلى عرض معلومة عن الأداة، وتعتبر هذه المعلومة تعريفا بالأداة (Tool Tip)، وهذا النوع من التعريف موجود كذلك في مناطق أخرى من بيئة IDE.

أدوات صندوق الأدوات

\* المؤشر : تستخدم لاختيار أدوات التحكم

\* صندوق الصور : تستخدم لعرض الصور ، واحتواء الأدوات الأخرى

\* العنوان A : لإنتاج منطقة العرض عنوان يتم إدخاله عن طريقها أو عن طريق الصفة Caption

\* صندوق النص : تستخدم لإنتاج صندوق يمكن أن يكتب فيه نص

\* الاطار : تستخدم لاحتواء الأدوات الأخرى

\* زر الأمر : تستخدم لإنتاج زر أمر، للتحكم بالتطبيقات والإجراءات في المشروع.

\* صندوق الفحص : تستخدم لإنتاج صندوق فحص، يستخدمه المستخدم في اختياراته لاحق

\* زر الخيار : تستخدم لعرض بدائل مختلفة ليختار منها المستخدم

\* صندوق السرد والحوار : تستخدم لاختيار بدائل من قوائم \*صندوق القوائم : تستخدم لبناء قوائم نصية باستخدام دوال محددة

\* شريط الاستعراض الأفقي : تستخدم لاستعراض واختيار قيمة من مجموعة من القيم، ويكون الاستعراض أفقياً، وهذه القيم قد تكون ألواناً أو أعداداً وهكذا حيث يؤدي الاختيار إلى تغييرات في النموذج.

\* شريط الاستعراض العمودي : تستخدم لاستعراض واختيار قيمة من مجموعة من القيم، ويكون الاستعراض عمودياً، وهذه القيم قد تكون ألواناً أو أعداداً وهكذا حيث يؤدي الاختيار إلى تغييرات في النموذج.28

## 6. شريط الخيارات Menu Bar

يوفر هذا الشريط مجموعة خيارات تساعد المبرمج على إنشاء برامجه وصيانتها وتنفيذها، الشكل (9) يبين هذه الخيارات في حالة استخدام Standard EXE والجدول (2) يعطي فكرة مختصرة عن كل منها:

File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help

وصف شريط الخيارات

\* File Menu : تحتوي على أوامر لعمل مشروع جديد أو لفتح مشروع موجود أو لتخزين مشروع أو لطباعة الجمل البرمجية وشاشات المشروع أو لعمل ملف تنفيذي للمشروع.

\* Edit Menu : تحتوي على أوامر لنسخ ولصق الأشياء أو للبحث

\* View Menu : تحتوي على أوامر لإظهار وإخفاء بعض النوافذ ف شاشة فيجوال بيسك.

\* Project Menu :تحتوي على أوامر لإضافة شاشات وأدوات جديدة للمشروع.

\* Format Menu : تحتوي على أوامر للتحكم بأماكن الأشياء الموجودة ف الشاشات التي يتم عملها في المشروع

\* Debug Menu: تحتوي على أوامر تساعد على تتبع تنفيذ المشروع ومعرفة أماكن الأخطاء في الجمل البرمجية.

\* Run Menu : تحتوي على أوامر للتحقق من عدم وجود أخطاء في الجمل البرمجية وتنفيذ البرنامج.

\* Query Menu : تحتوي على أوامر تسهل التعامل مع قواعد البيانات الموجودة في المشروع.

\* Diagram Menu : تحتوي على أوامر للمساعدة في الأمور التنظيمية في المشروع

\* Tools Menu: تحوي على أوامر تساعد على إضافة أجزاء تنفيذية البرنامج وإضافة قوائم النوافذ الموجودة في المشروع

\* Add-Ins Menu:تحتوي على أوامر تساعد على إضافة أدوات مساعدة تتعلق بقواعد البيانات وغيرها

\* Window Menu: تحتوي على أوامر لترتيب النوافذ بشكل متناسق.

\* Help Menu : تحتوي على أوامر تساعد على معرفة معلومات عن أي جزء من لغة الفيجوال بيسك.

## 7. شريط الأدوات: Tool Bar

يوفر هذا الشريط بديلا مباشرة وسهلا لاستخدام الخيارات الأكثر استخداما، وذلك بوجود أيقونة لكل من هذه الخيارات

### 4. بناء مشروع أو برنامج جديد

بعد أن تعرفنا أهم عناصر بيئة التطوير IDE، فمن المناسب والمفيد أن نبدأ برنامجنا الأول بلغة VB6 ، سوف نبني مشروعا جديدا يضع العبارة التالية:

Welcome To AL -Quds Open University And Avecinna Virtual Campus

في نموذج (Form)، وإنشاء زر (Button) باسم (OK) يمكنك من الخروج من المشروع.

### 1.4 خطوات إنجاز مشروع جديد

بعد تشغيل برمجية VB6 ، وظهور شاشة نافذة مشروع جديد التي أوردناها في الشكل (1)، واختيار Standard EXE، كما ذكرنا سابقا، وظهور الشاشة التي أوردناها في الشكل (2)، ننفذ الخطوات التالية:

1. من صندوق الأدوات، انقر نقرا مزدوجا على زر الأمر (Command Button).

لاحظ ظهور Command1 وسط نافذة النموذج، ويمكنك تحريك مكان الأمر بالفقر والسحب إلى المكان المناسب، في أسفل النافذة

2. من صندوق الأدوات كذلك، انقر نقرا مزدوجا على زر العنوان (Label). لاحظ ظهور Label1 في نافذة النموذج، ضعه في وسط وأعلى النافذة، فتظهر الشاشة

3. اختر زر Command بالنقر عليه، ثم من نافذة الخصائص، انقر نقرا مزدوجا على (Caption)، بعد ذلك أكتب OK بدلا من

Command1. بنفس الأسلوب غير الاسم (Name) من Command 1 إلى Cmd

4. أعد خطوة 3، لكن هذه المرة للزر 1

Label بحيث تغير Caption الخاص به إلى العبارة التالية:

Welcome To AL-Quds Open University And Avicenna Virtual Campus

وتغير Name الخاص به إلى Lb1Welcome

5. من نافذة النموذج، انقر نقرا مزدوجا فوق الأمر OK ، حيث ستظهر نافذة البرنامج (Code) كما في الشكل (14)، كما يمكن إظهار نفس الشاشة بالنقر المزدوج على النموذج نفسه، أو على أي زر (كينون) موجود على سطح النموذج

ويجدر بالذكر أنه لا يلزم كتابة جمل تنفيذية في هذا المثال، فقط اطبع كلمة End بين سطري Private Sub و End Sub، إن عملية كتابة جمل VB عند الحاجة تتم باستخدام محرر (Editor) يأتي مرافقا للغة، ويمكنك ملاحظة صندوقين موجودين أعلى شاشة البرنامج:

الأول: صندوق الإجراءات (Procedure Box)، ويقع أعلى يمين الشاشة، ومكتوبا فيه

Click في العادة

الثاني: صندوق الكينونات (Object Box) ، ويقع أعلى يسار الشاشة، ومكتوب فيه cmdWelcome في حالة برنامجنا.

### 2.4 تنفيذ البرنامج وإنهاؤه

1. من قائمة Run ، انقر على زر Start ، وذلك لتنفيذ التطبيق،

2. يتم إنهاء البرنامج باختيار End من قائمة Run ، أو بالنقر على أداة End من شريط الأدوات.

#### 3.4 عملية تخزين مشروع وفتحه

كما تعلم عزيزي الدارس، نلجأ إلى تخزين برامجنا وملفاتنا، وذلك للاستفادة منها لاحقاً باسترجاعها عند اللزوم، وبعد أن قمنا ببناء مشروع أو برنامج، واختبرنا تنفيذه، فيمكن تخزينه كما يلي:

1. من قائمة File، انقر زر Save Project، فيظهر مربع حوار تخزين ملف بعنوان Save File As

2. ادخل اسم الملف في مربع File Name ، وليكن FrmALQUDES ثم انقر زر Save ليظهر لك مربع حوار تخزين مشروع. أدخل كلمة ALQUDES في مربع Project Name ثم انقر زر Save. إن عملية التخزين الموصوفة في الخطوتين 2 ، 1 تؤدي إلى تخزين المشروع وملف نافذة Form والأوامر والخصائص والبرامج ذات العلاقة بالمشروع قيد النقاش

تذكر عزيزي الدارس ما ذكرناه حول فتح مشروع موجود، ويتم ذلك إما باستخدام Recent من نافذة مشروع جديد، أو باستخدام Existing من تحميل المشروع في بيئة IDE وفتحه، ونطلب منك عزيزي الدارس أن تتفقد ذلك على المشروع الذي ناقشناه.

#### 4. طباعة مشروع

من قائمة ملف File، انقر الزر Print فيظهر مربع حوار الطباعة، تستطيع من هنا أن تقوم بما يلي:

\* طباعة ما تم تظليله Selection: تختار Selection إذا أردت طباعة جزء معين ظالته مسبقاً من التعليمات البرمجية في المشروع.

\* طباعة الشاشة الحالية Current Module : تختار Current Module إذا أردت طباعة الشاشة التي تعمل عليها حالياً والجميل البرمجية المتعلقة بها، ثم تختار Code إذا أردت طباعة الجمل البرمجية المتعلقة بالشاشة فقط، أو تختار Image form إذا أردت طباعة شكل الشاشة فقط، وتستطيع اختيار الخيارين معاً.

\* طباعة المشروع الحالي Current Project: تختار Current Project إذا أردت طباعة كامل المشروع، ثم تختار Code إذا أردت طباعة الجمل البرمجية في مشروعك فقط، أو تختار Form Image إذا أردت طباعة أشكال الشاشات التي صنعتها في مشروعك فقط، وتستطيع اختيار الخيارين معاً.

#### 5.4 أنواع الملفات وامتداد اسمائها

عزيزي الدارس، ذكرنا سابقاً أن مشروع VB6 يتكون من عدة ملفات مترابطة، وتظهر هذه الملفات في نافذة مستكشف المشروع (Project Explorer)، ويمكننا تمييز ثلاثة أنواع من الملفات كما يلي:

1. ملف المشروع (The Project File)

ينتهي اسم هذا الملف بالامتداد (.vbp)، ويحتوي على جميع الملفات والمعلومات ذات العلاقة بالمشروع.

2. ملف النموذج (The Form File)

ينتهي اسم هذا الملف بالامتداد (.frm)، ويحتوي على وصف للنموذج وما يتعلق به من أدوات وبرامج.

3. ملف الأجزاء (Module File)

ينتهي اسم هذا الملف بالامتداد (.bas)، ويحتوي على الجمل والتراكيب البرمجية المكتوبة بلغة VB6، وذلك ضمن برنامج فرعي (Sub) أو أكثر، ولا ترتبط هذه الجمل والتراكيب بأية نماذج أو أدوات أخرى.

#### 6.4 فتح مشروع موجود وإجراء تغييرات عليه

بإمكانك عزيزي الدارس، فتح مشروع VB6 موجود، وإجراء أية تعديلات عليه إن رغبت، وذلك بإحدى طريقتين كما يلي:

من قائمة File، اختر Open Project، فيظهر صندوق الحوار

\* أو بعد تشغيل VB6، وظهور شاشة تنافذة مشروع جديد"، انقر على Existing فيظهر صندوق الحوار المبين

\*. حدد مكان تخزين المشروع الذي تريد فتحه، وذلك في شريط Look in.

\*. اختر اسم المشروع الذي تريد فتحه بالنقر عليه، ثم انقر Open، أو انقر نقرا مزدوجا على اسم المشروع المعني، وفي كلتا الحالتين سيتم فتح المشروع.

\*. قم بإجراء التغييرات التي تريد على المشروع، شريطة أن تكون قد اخترت عنصر المشروع الذي تريد أن تغير فيه ضمن إطار المشروع.