

أساسيات البرمجة بلغة فيجول بيسك

المتغيرات والثوابت Variables and Constants

إن المتغير يشغل حيزاً في الذاكرة، ويأخذ قيمةً مختلفة، شريطة أن تتوافق هذه القيم مع نوع المتغير، فمثلاً إذا كان X متغيراً صحيحاً فلا يجوز أن نُسند إليه قيمة حقيقية. وعندما يجري التصريح عن متغير، فإن VB يخصص له حيزاً من الذاكرة يتناسب مع نوعه.

● أسماء المتغيرات

اسم المتغير هو صف من الحروف يبدأ بحرف إنجليزي، ولا يتخلله فراغ، ولا إشارة خاصة مثل (+، -، :، *، ...).
وليس كلمة محجوزة مثل Dim، ولا تعليمة من تعليمات اللغة مثل For.

● الإعلان عن المتغيرات

يمكننا الإعلان عن المتغيرات بأحد الأوامر التالية:

Public (Global), Private, Protected Dim, Static

ولكل من هذه الأوامر استخدام خاص به إما على مستوى الإجراء، أو النافذة، أو على مستوى المشروع بشكل عام. والجدول التالي يبين ذلك:

الأمر	الاستخدام
Global أو Public	على المستوى العام للمشروع.
Private	على مستوى نافذة واحدة.
Dim	على مستوى الإجراء وعلى مستوى نافذة واحدة.
Static	على مستوى الإجراء
Protected	تستخدم في الوراثة

والسؤال الآن، كيف نصرح أو نعلن عن المتغيرات باستخدام تلك الأوامر. إن عبارة التصريح عن المتغيرات لها الشكل القياسي التالي:

Cmd-name var-name as type

اسم الأمر

اسم المتغير

النوع

أنواع المتغيرات

لدينا عدة أنواع من المتغيرات، منها: عددية، محرفية، منطقية، أو مختلفة وهي ممثلة بالجدول التالي:

النطاق	الحجم	الشرح	نوع البيانات
-32768 to 32767	16 بت	صحيح قصير	Short
0 to 65535	16 بت	صحيح قصير موجب	Ushort
-2147483648 to 2147483647	32 بت	صحيح	Integer
0 to 4,294,967,295	32 بت	صحيح موجب	UInteger
-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	64 بت	صحيح طويل جداً	Long
0 to 18,446,744,073,551,615	64 بت	صحيح طويل جداً موجب	Ulong
-3.4E+38 to +3.4E+38	32 بت	حقيقي ذو دقة عادية	Single
-1.8E+308 to 1.8E+308	64 بت	حقيقي ذو دقة مضاعفة	Double
حقيقي ذو فاصلة ثابتة	128 بت	حقيقي ذو فاصلة ثابتة	Decimal
0 to 255	8 بت	صحيح مختصر	Byte
-128 to 127	8 بت	صحيح مختصر	Sbyte
Unicode code(0 to 65,535)	16 بت	حرف	Char
أي رمز من 0 الى 2 مليار حرف	16 بت عادة لكل حرف	سلسلة نصية(مجموعة حروف)	string
True or false خلال التحويل يتم تحويل الرقم 0 الى false والقيم الأخرى الى true	16 بت	قيمة منطقية	Boolean
9999 31 ديسمبر حتى 0001 1 يناير	64 بت	تاريخ	date
object أي نوع يمكن تخزينه في متغير نوعه object	32 بت	أي نوع من أنواع البيانات	object

أمثلة:

تعريف المتغيرات بالشكل القياسي	تعريف المتغيرات بالواحد
Dim x as integer	متغير صحيح Dim x%
Public w as single	متغير حقيقي Public w!
Private y as String	متغير محرفي أو شريطي Private y\$
Static t as Double	متغير حقيقي ذو دقة مضاعفة Static t#

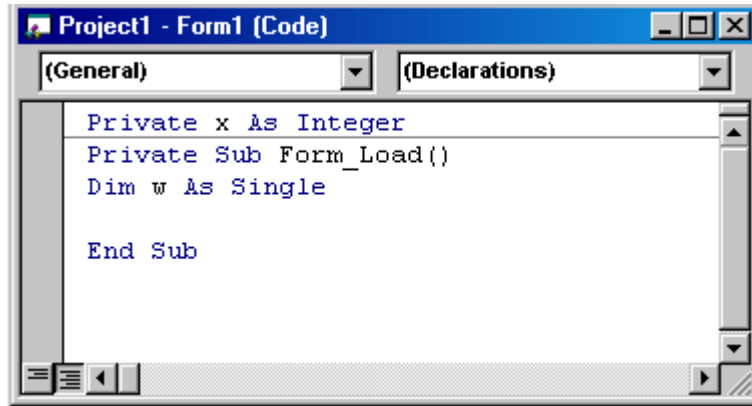
أين يتم التصريح عن هذه المتغيرات ؟ للجواب على ذلك نأخذ المثال التالي:

مثال:

لو طلب منا تعريف متغير عام **Z** من النوع الصحيح الطويل، ومتغير على مستوى النافذة **X** من النوع الصحيح، ومتغير آخر على مستوى الإجراء من النوع الحقيقي **Y**.

يتم التصريح عن المتغيرات العامة لمشروع متعدد النوافذ في ملف خاص اسمه Module يمكن الحصول عليه من

القائمة Project ثم Add Module ومن ثم open، بعد ذلك نكتب **Public Z as Long** للتصريح على متغير على مستوى النافذة أو الإجراء، نقوم بالنقر المزدوج على النافذة Form، فنحصل على النافذة البرمجية، عندئذ نختار General من خانة الأدوات، ثم نكتب التعريفين، أحدهما على مستوى النافذة، والآخر على مستوى الإجراء (Load)، كما في الشكل التالي



• مدى المتغيرات وعمرها

1- المتغيرات العامة

هي المتغيرات التي يمكن أن تستخدم في كل مكان من البرنامج، وتبقى في الذاكرة خلال فترة عمل البرنامج، ولا تحذف من الذاكرة إلا بعد انتهاء البرنامج. وكما ذكرنا يجب أن نعلن عن هذا النوع من المتغيرات في الملف Module، له الامتداد .Bas.

2- المتغيرات المحلية على مستوى النافذة

هذه المتغيرات يمكن التعامل معها فقط في النافذة التي أعلن فيها، وهذا النوع يبقى في الذاكرة خلال فترة عمل البرنامج. ويظهر الفرق جلياً بين المتغيرات العامة والمتغيرات على مستوى النافذة عندما يتكون المشروع من عدة نوافذ، عندها المتغيرات العامة يمكن الوصول إليها في جميع أجزاء البرنامج، بينما المتغيرات على مستوى النافذة لا يمكن الوصول إليها إلا في النافذة التي عرفت فيها.

3- المتغيرات المحلية على مستوى الإجراء

هذه المتغيرات يمكن الوصول إليها فقط ضمن الإجراء الذي أعلنت فيه، ولكن تختلف مدة بقائها في الذاكرة، وذلك تبعاً للأمر الذي أعلنت من خلاله وهو إما Dim، أو Static .
 فإذا أعلنت بالأمر **Dim** مثلاً (Dim x as integer)، فإن VB يزيلها من الذاكرة بعد إنهاء الإجراء، لذلك فهي تفقد قيمها، وإذا استدعي الإجراء مرة أخرى، يعاد حجز الذاكرة لها من جديد وهكذا...
 أما إذا أعلن عن المتغيرات بالأمر **Static** مثلاً (Static x as integer)، فعندها هذه المتغيرات سوف تحتفظ بقيمها، ولا تحذف من الذاكرة.

• الثوابت

هي قيم ثابتة، لا تتغير أثناء تنفيذ البرنامج، تأخذ أنواع المتغيرات، ومداهما يتحدد بالأماكن التي أعلنت فيها، وهي تتبع المتغيرات في قواعدها، ويعلن عنها بالشكل:

Const x=25 : Const y="Hello" : Const z=2.9 ...

الصيغ العامة للتعليمات: Common Formulas of VB Instructions

لهذه الصيغة شكلان هما :

Var-name=New Value

1. تعليمة الإسناد

X=25

مثال:

Object-name . Property = New Value

2. التعليمة

وهذه الصيغة مشهورة وهامة، وتستخدم بكثرة بلغة فيجوال بيسك واستخدمناها في المحاضرات الماضية.

مثال:

Textbox1.backColor = color.salmon	اجعل خلفية النص بلون salmon
Button1.Enabled = false	اجعل زر الأمر معطلاً
Label1.Autosize = true	اجعل حجم أداة العنوان بحجم النص

أنواع العمليات: Operators Types

- العمليات الحسابية وهي
الجمع + ، الطرح - ، الضرب * ، القسمة / ، والرفع إلى قوة ^ ، باقي القسمة mod القسمة الصحيحة \ .
- عمليات المقارنة
تستخدم هذه العمليات بين الأعداد وبين الحروف، وهي:
= ، < > ، < ، > ، <= ، >= .
- العمليات المنطقية not ، or ، and
- الدوال الجاهزة: Ready Functions

1. الدالة Parse()

تقوم هذه الدالة بتحويل سلسلة المحارف إلى قيم عددية، وبما أن الأداة TextBox، هي الأداة التي من خلالها يتم إدخال البيانات، فإن كل البيانات نصية، وليست عددية، ولذلك نحتاج إلى دالة التحويل Parse. مثال:

X=Integer.Parse(Textbox1.text)

Y= Integer.Parse(Textbox2.text)

2. الدالة ToString

هذه الدالة التي تقوم بتحويل القيم العددية إلى سلسلة محارف. مثال:

Label1.text = X.ToString

Label2.text = y.ToString

3. الدالة mod

لها الشكل $I \text{ mod } j$ باقي القسمة الصحيحة لـ I على j. مثال:

$$17 \text{ mod } 3 = 2 ; \quad 5 \text{ mod } 7 = 5 ; \quad 20 \text{ mod } 7 = 6$$

1. أداة التاريخ والوقت DateTimePicker

تقوم بعرض التاريخ والوقت.

2. صندوق الصور PictureBox

يستخدم كوعاء توضع الصور بداخله.

3. صندوق الخيارات CheckBox

يستخدم عند وضع مجموعة من الخيارات ويمكن للمستخدم اختيار أكثر من خيار

4. زر اختيار RadioButton

يستخدم عند وضع مجموعة من الخيارات ويمكن للمستخدم اختيار خيار واحد فقط وعند اختيار أحد الأزرار تزول علامة الاختيار عن باقي الأزرار.

5. صندوق المجموعات GroupBox

يستخدم كحاوية لبعض الأدوات التي يتم التعامل معها كمجموعة واحدة عندما نريد تحريك مجموعة من الأدوات معا توضع داخل صندوق المجموعات

وضع الأدوات داخل GroupBox (وبشكل عام داخل حاوية Container)

يتم بإحدى الطرق التالية:

- (1) بتحديد الحاوية Container التي قمت بإضافتها إلى النموذج ثم من قائمة الأدوات تقوم بالنقر المزدوج على الأداة التي تريد أن تضيفها على Container.
- (2) بواسطة السحب والإفلات من قائمة الأدوات إلى داخل ال Container.
- (3) بسحب الأداة الموجودة على النموذج إلى داخل ال Container.
- (4) بواسطة القص والاصق من النموذج إلى داخل ال Container.

ملاحظة:

إذا رغبتنا إنشاء مجموعة كبيرة من الأزرار (أو أي أداة) على النموذج فيتم ذلك بالضغط المتواصل على مفتاح Ctrl ثم النقر نقرة واحدة بالماوس على الأداة التي تريد إضافة عدداً منها، ثم في كل مرة تنقر على النموذج سيتم إضافة أداة جديدة وباسم جديد.

برنامج لتصميم آلة حاسبة

a. مرحلة التصميم المرئي

1. الأدوات

Label1, Label2, Label3—textbox1, textbox2, textbox3, GroupBox1
Button1, Button2, Button3, Button4, Button5

2. النموذج



b. مرحلة البرمجة

تعريف المتغيرات كمتغيرات عامة على مستوى النافذة

```
Public Class Form1
    Dim x As Integer
    Dim y As Integer
```

كتابة الاكواد البرمجية لازرار اجراء العمليات الحسابية

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles Button4.Click
    x = Integer.Parse(TextBox1.Text)
    y = Integer.Parse(TextBox2.Text)
    TextBox3.Text = (x + y).ToString
End Sub
```

اساسيات البرمجة

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    x = Integer.Parse(TextBox1.Text)
    y = Integer.Parse(TextBox2.Text)
    TextBox3.Text = (x - y).ToString
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
    x = Integer.Parse(TextBox1.Text)
    y = Integer.Parse(TextBox2.Text)
    TextBox3.Text = (x * y).ToString
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
    x = Integer.Parse(TextBox1.Text)
    y = Integer.Parse(TextBox2.Text)
    TextBox3.Text = (x / y).ToString
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button7.Click
    x = Integer.Parse(TextBox1.Text)
    y = Integer.Parse(TextBox2.Text)
    TextBox3.Text = (x Mod y).ToString
End Sub
```