القواعد The Rules

السلام عليكم

بالنسبة لموضوع اليوم فلحسن الحظ وقع في يدي ملحق كتاب يتحدث عن القواعد التي يجب ان يتقيد بها فريق العمل قبل البدء بمشروع جديد لهذا وجدت انها فرصة للبدء بتكوين عادات برمجية جيدة حتى تصبح برامجنا مكتوبة بطريقة المحترفين.

حاولت تلخيص الملحق بطريقتي المتواضعة فاعذرونا على التقصير...

يقصد بالقواعد هنا الطريقة التي سيستخدمها المبرمجون في تسمية عناصر البرنامج. مثلا طريقة تسمية الجداول و الحقول و الدوال و المتغيرات و غيرها. كذلك يقصد بها شروط كتابة الشفرة.

تذكر: أفضل المبرمجون هم المتحمسون و الملتزمون بالقواعد

فقبل عمل اي برنامج يجب على فريق العمل الاتفاق على قواعد معينة ليتم الالتزام بها في كتابة البرنامج. فإذا كان كل مبرمج كتب شفرته بطريقته الخاصة فسيؤدي ذلك الي صعوبة فهم الشفرة بين أعضاء الفريق و ربما ادى الي ضياع الكثير من الوقت لفهم مقطع معين من الشفرة . اذا انه حتى اختلاف تسمية الجداول يسبب ارباكا لكثير من المبرمجين. اخبرني كيف ستتعامل مع قاعدة بيانات تحتوي على جدول ذات لاحقة tbl و اخرى بدون لاحقة. و زد على ذلك اذا سميت جداول أخرى باللغة العربية.

لا يعني هذا انه محرم كسر القواعد فأذا كان لديك سبب مقنع فلما لا كن مبدعا. المهم يجب ان تقنع نفسك على الاقل بكسر القاعدة. لهذا اجب على السؤال التالي كلما فكرت بكسر قاعدة ما : لماذا يجب على فعل ذلك؟؟؟

المهم ياشباب دعونا نبدا بالقواعد العامة اولا

١ - القواعد العامة General Rules

==========

١-١ : تقيد بالتسمية من البداية: استخدم البوادئ prefixes الواضحة و الفراغات spaces لجعل حقول البيانات و caption و عناوين رؤس التقرير و اسماء المتغيرات واضحة من اول نظرة. القي نظرة على المثال التالي:

Good:
Field name: CustomerFirstName (table name prefix)
Field caption: First Name (space added for clarity)
Report column header: First Name
Code variable: strFirstName (data type prefix, no spaces)

Bad:
Field name: CustFName
Field caption: Christian Name
Report column header: Given Name
Code variable: strFNm

١-٢ : يجب استخدام امر Exit مرة واحدة فقط في اي دالة او اجراء. لان استخدام اكثر من عبارة Exit يؤدي الي الارباك و تعقيد صيانة الشفرة.

Good:
Function CodeToName(strCategory as string) as string
Dim strReturn as string
IF strCategory = “MAN” Then

```
strReturn = “Manchester”
Elseif strCategory = “LHR” Then
strReturn = “London Heathrow”
Else
strReturn = “Unknown”
End If
CodeToName = strReturn
End Function
```
(Error handling code not shown)

Bad:
```
Function CodeToName( strCategory as string) as string
IF strCategory = “MAN” Then
CodeToName = “Manchester”
Exit Function
Elseif strCategory = “LHR” Then
CodeToName = “London Heathrow”
Exit Function
Else
CodeToName = “Unknown”
Exit Function
End If
End Function
```

ملاحظة:

كن منتبها لطريقة استخدام الامر Exit عند معالجة الاخطاء .error handling code في شفرة الاجراء او الدالة.
لاحظ الطريقة الجيدة و الطريقة السيئة في كتابة الشفرة في الكود التالي:

Good:
```
Private Sub txtFilmYearReleased_Exit(Cancel As Integer)
On Error GoTo ErrorHandler
Me.lblHelp.Caption = ""
CleanUpAndExit:
Exit Sub
ErrorHandler:
Call MsgBox("An error was encountered" & vbCrLf & _
vbCrLf & _
"Description: " & Err.Description & vbCrLf & _
"Error Number: " & Err.Number, , "Error")
Resume CleanUpAndExit
End Sub
```

Bad:
```
Private Sub txtFilmYearReleased_Exit(Cancel As Integer)
On Error GoTo ErrorHandler
Me.lblHelp.Caption = ""
Exit Sub
ErrorHandler:
Call MsgBox("An error was encountered" & vbCrLf & _
vbCrLf & _
"Description: " & Err.Description & vbCrLf & _
"Error Number: " & Err.Number, , "Error")
End Sub
```

١-٣ : سمي الاجراءات و الدوال و المتغيرات و الحقول و اسماء الكائنات Objects و عناصرالتحكم بكلمات مختلطة وبدون فراغات او تسطير Underscores. و لاتنسى تكبير الحرف الاول لكل كلمة.

الحكمة من عدم استخدام التسطير Underscores في اسماء الاجراءات و الدوال هو لتسهيل التفريق بين الاجراءات و الدوال العادية و بين الاحداث Events

اما الغرض من عدم استخدام التسطير Underscores في اسماء المتغيرات و ذلك لتفريق بين المتغيرات و الثوابت Constants

Good:
strCompanyPostCode
txtCompanyPostCode
GetCompanyPostCode()

Bad:
strCompany_Post_Code
txtcompanypostcode
GetCompanypostcode()

١-٤: سمي المتغيرات بصيغة الاسم او بصيغة الاسم مع الفعل
امثلة بصيغة الاسم

strCustomerFirstName
strCustomerLastName
strCustomerPostCode

ملاحظة:
المتغيرات المنطقية Boolean غالبا ماتستخدم الاسم و الفعل معا. انظر للامثلة ادناه:

Good:
blnCustomerIsValid, blnCustomerHasCar

Bad:
blnIsValidCustomer, blnHasCarCustomer

١-٥: سمي الاجراءات و الدوال بالانجليزية المحكية The Spoken English Word
هذه القاعدة تجعل الدوال و الاجراءات قريبة منا و تسهل فهم الغرض منها.

Good:
AddCustomer
DeleteCustomer
CalculateMovingAverage
GetCustomerByPrimaryKey

Bad:
CustomerAdd
CustomerDelete
MovingAverageCalculate
CustomerGetByPrimaryKey

١-٦: لا تستخدم الاختصارات abbreviations على الإطلاق عند تسمية أي عنصر ينتمي لقاعدة البيانات مثل اسماء الجداول و الحقول و المتغيرات و الدوال و غيرها.

Good:
strCustomerFirstName

Bad:
strCustFNm, strCstmFName

١-٧: اعطي دائما اسماء بصيغة المفرد in the singular تجنب استخدام صيغة الجمع.

على سبيل المثال الدالة *GetCompanyType(lngCompanyType)* نعيد صفرا او واحد او اكثر من السجلات التي تحدد نوع الشركة. بدون هذه القاعدة سيتوجب على المبرمج ان يخمن هل يجب عليه استخدام الدالة *GetCompanyType()* (اي بصيغة المفرد) او يستخدم الدالة *GetCompanyTypes() (اي بصيغة الجمع)*

Good:
Table name: Customer

Bad:
Table name: Customers

١-٨: لا تستخدم الارقام المجردة في شفرتك و لكن استخدم الثوابت Constants عوضا عن ذلك:

Good:
GetCompany(WAL_NORTH_WEST_AREA_ONLY)

Bad:
GetCompany(23)

١-٩: حدد نوع بيانات data type لكل وسطاء parameters الدالة و الاجراء.

Good:
Function GetCustomer(lngCustomerID as Long)

Bad:
Function GetCustomer(lngCustom)

١-١٠: صرح عن وسطاء parameters الدوال و الاجراءات ب **ByVal** ما لم يكن لديك سبب وجيه لاستخدام **ByRef**

Good:
Function GetCustomer(ByVal lngCustomerID as Long)

Bad:
Function GetCustomer(lngCustomerID)
Function GetCustomer(ByRef lngCustomerID)

١-١١: لا تستخدم متغيرات من نوع **global** **globally scoped variables.**

اذا ان تطبيق هذه القاعدة يؤدي الي عدم خرق احد مفاهيم البرمجة الكائنية المنحي OOP الا وهو التغليف encapsulation

ملاحظة:
هذه القاعدة لا تطبق على الثوابت العامة .global constants

١-١٢: لا تستخدمة المعامل + مع النصوص Strings ولكن استخدم المعامل & عوضا عن ذلك.

ملاحضة: استخدم المعمل + مع العمليات الرياضية فقط.

١-١٣: تجنب استخدام الامر ***Exit For*** *و كذلك الامر* ***Exit Do***

والغرض من هذه القاعدة هو نفس غرض القاعدة رقم 1-2 اي لتجنب ارباك المبرمج

1-14: استخدم الامر Call عند استدعاء الدوال و الاجراءات.
والغرض من ذلك هو لجعل قراءة الكود اكثر وضوحا

لاحظ الفرق في الاتي:

Without the *Call* statement
strCustomerName = GetCustomerName(lngCustomerID)
DeleteCustomer lngCustomerID
MsgBox “You have successfully deleted customer: “ & strCustomerName

With the *Call* statement
strCustomerName = GetCustomerName(lngCustomerID)

```
Call DeleteCustomer( lngCustomerID)
Call MsgBox( “you have successfully deleted customer: “ & strCustomerName )
```

١-١٥: لا تستخدم الخواص الافتراضية default properties التابعة للكائن.

على سبيل المثال الخاصية الافتراضية لمربع النص Text Box هي Value فاذا استخدمت اسم مربع النص بدون الاشارة اي اي خاصية فان الخاصية Value سيتم اعتمادها وذلك لانها الخاصية الافتراضية لمربع النص.

Good:
txtCustomerFirstName.Value = “John”

Bad:
txtCustomerFirstName = “John”

بهذا نكون قد انهينا القواعد العامة

٢- قواعد تسمية الجداول و الحقول Table and Field naming

٢-١: لا تستخدم ابدا البادئات Prefixes لاسماء الجداول

يرى الكاتب بعدم وجوب استخدام البوادئ Prefixes للجداول و ذلك لان الجداول هي العناصر الرئيسية في قاعدة بيانات. يعني بالعربي الفصيح الجدول غني عن التعريف ببوادئ او ما شابه ذلك.

٢-٢: سمي المفاتيح الرئيسة بصيغة التاليه:

< اسم الجدول > + ID
<table name> + <ID>

Example: Table named *Customer*
Primary Key: *CustomerID*

٢-٣: دائما سمي المفاتيح الاجنبية باسم المفتاح الرئيسي للجدول الذي ترتبط به.

على سبيل المثال:
المفتاح الاجنبي *MediaID في الجدول Film يحمل نفس اسم* المفتاح الرئيسي *MediaID في الجدول Media*.

2-4: في حالة عدم الوضوع , فيمكنك ضم اسم وحدة القياس **the unit of measure** الي اسم الحقل لازالة الغموض.

المثال التالي يوضح كل شي:

في جدول الافلام Film هناك حقل يمثل طول الفلم. المهم كيف يمكن ان نعرف اذا كان طول الفلم بالدقائق او الساعات. هنا لازالة الغموض يمكن اضافة وحدة القياس الي اسم الحقل.

Good: FilmLengthMinutes

Bad: FilmLength

٢-٥: سمي جدول الربط Link Table (وهو الجدول الناتج من كسر العلاقة اطراف بـ اطراق Many to Many) باسمي الجدولين المكونين للعلاقة.

مثال:
الجدول الافلام Film يرتبط بعلاقة اطراف الي اطراف بالجدول الممثلين Actor. هنا سيكون اسم جدول الربط هو اسم جدول الافلام مع اسم جدول الممثلين اي FilmActor

٢-٦: جميع اسماء حقول الجدول تبدأ باسم الجدول الذي تنتمي اليه.

ملاحظة هامة: الاستثناء من هذه القاعدة هي اسماء المفاتيح الاجنبية راجع القاعدة ٢-٣.

مثال:
في جدول الطلاب Student فان:
المفتاح الرئيسي سيكون StudentID
الاسم : StudentName
تاريخ الميلاد : StudentDateOfBirth
محل الميلاد : StudentPlaceOfBirth
وقس على ذلك.

٣- قواعد خواص الحقول Field properties
==========

٣-١: المفاتيح الرئيسية دائما ليست ذات معني **meaningless** وتحمل نوع بيانات Autonumber

٣-٢: القمة الافتراضية Default Value للمفاتيح الاجنبية الغير مطلوبة **non-required foreign keys** يجب ان تكون Null

٣-٣: يجب على الاقل احد حقول الجدول (باستثناء المفتاح الرئيسي) ان يكون مطلوب **be required**

والغرض من ذلك هو لمنع المستخدمين من ادخال سجلات فارغة بطريقة عرضية.او بغير قصد.

٤- قواعد تسمية كائنات الاكسس Access object naming
==========

٤-١: اجعل لكل اسم كائن بادئة معد كائن ا الجدول.

Access Object	Prefix
Table	No Need
Query	qry
Form	Frm
Report	Rpt
Macro	Mcr
Module	mod

٥ – قواعد تسمية عناصر تحكم النموذج Form control naming
==========

٥-١: الجدول يشرح نفسه

Prefix	Object Type	Example
cbo	Combo Box and Drop Down List Box	cboEnglish
chk	Checkbox	chkReadOnly
cmd	Command Button	cmdOK
dat	Data control	datFilm
dir	Directory list box	dirSource
dlg	Common Dialog Control	dlgFileOpen
frm	Form	frmEntry
fra	Frame	fraStyle
img	Image	imgIcon
Lbl	Label	lblHelpMessage

lin	Line	linVertical
lst	List Box	lstPolicyCodes
mnu	Menu	mnuFileOpen
opt	Option Button	optRed
ole	OLE Control	oleWorksheet
pic	Picture	picHotel
spn	Spin Control	spnPages
txt	Text Box	txtLastName
tmr	Timer	tmrAlarm

٦- قواعد تسمية المتغيرات Variable naming
==========

٦-١:اسماء المتغيرات الوحدة النمطية **Module scoped variables** لديها بادئة m بينما اسماء المتغيرات العامة **globally scoped variables** فبادئتها g

```
mstrCustomerName        ‘ Module level
strCustomerName         ‘ Local to sub
gstrApplicationLanguage ‘ Globally visible
```

٦-٢: استخدم بادئة من ثلاثة حروف لكل نوع بيانات

Data Type	Prefix
String	str
Long integer	lng
Boolean	bln
Currency	cur
Double	dbl
Variant	vnt (var also acceptable)
Date and Time	dat (dtm also acceptable)

٧ – قواعد تسمية الثوابت Constants
==========

٧-١: سمي الثوابت بحروف الانجليزية في حالة upper-case و افصل الكلمات ب التسطير underscore

Good: WAL_APPLICATION_STATUS
Bad: conApplicationStatus

٧-٢: اختر بادئة لاسماء الثوابت بحيث يندر ان يستخدمها مبرمج غيرك.

على سبيل المثال انا استخدم البادئة WAL وهي اختصار لاسمي Whale

٨ - قواعد التصريح عن المتغيرات Variable declaration
==========

٨-١: استخدم الخيار *Option Explicit* في كل الوحدات النمطية Modules

٨-٢: حدد نوع بيانات أي متغير حتى لو كان من نوع Variant

Good:
Dim strFirstName as string
Sub DeleteCustomer(lngCustomerID as Long)

Bad: Dim strFirstName
Sub DeleteCustomer(lngCustomerID)

٨-٣: صرح عن المتغيرات كلا على حده و استخدم Dim لذلك.. لا تدمج تصريح متغيرين او اكثر مع Dim واحدة.
اذا خالفت هذه القاعدة فسيصبح نوع بيانات المتغير غير واضح و ربما مشوشا.

Good:
Dim strFirstName as string
Dim strLastName as string

Bad:
Dim strFirstName, strLastName as string ' first variable is a variant

٨-٤: عند التصريح عن متغير ما اعطي تعليق comment يوضح الغرض منه

Example:

```
Dim lngFormMode as long              'Can be TSM_FORM_INSERT or TSM_FORM_UPDATE
Dim lngCompanyCount as long          ' Used to iterate through rsCompanies
Dim blnCompanyIsActive as boolean    ' Flags current credit status
```

٩- قواعد معالجة الأخطاء Error handling
==========

٩-١: معالجة الاخطاء **Error handling** يجب ان يوجد في كل اجراء او دالة بدون استثناء

حتى لو كانت الشفرة بسيطة فان اتباع هذه القاعدة لا يضر بل يجعل برنامجك مستعدا لاي خطء غير متوقع ليبدأ بمعالجته بطريقة السايمة.

٩-٢: استخدم دائما العنوان Label - ErrorHandler: - و العنوان - CleanUpAndExit: - في كل دالة و اجراء.

يرى الكاتب بانه لا يوجد هناك سبب وجية لاستخدام عناوين مختلف في كل اجراء و دالة لمعالجة الاخطاء. اذ ان استخدام نفس عناوين معالجة الاخطاء Error Handler Labels يجعل الكود و اضحا و يزيد من انتاجيتك. اذ تستطيع على سبيل المثال نسخ و لصق كود معالجة الخطاء من اجراء الي اخر.

```
Private Sub MySub()

On Error GoTo ErrorHandler:

'Centeral section of code

CleanUpAndExit:
Exit Sub

ErrorHandler:

'Centeral section of Error handler code
Resume CleanUpAndExit

End Sub
```

٩-٣: اذا استخدمت الامر DoCmd.SetWarnings(False) في الدالة او الاجراء فيجب استخدام الامر DoCmd.SetWarnings(True) في الكود الخروج من الدالة او الاجراء (في هذه الحالة سيكون بعد العنوان CleanUpAndExit)

عدم اتباع هذه القاعدة سيؤدي بان الاكسس لن يعرض رسائل التحدير طوال جلسة العمل the entire Access session.

١٠- قواعد هدم الكائنات Object destruction
==========

١٠-١: كل ال **Recordsets** المفتوحة يجب ان تغلق بشكل صريح **be explicitly closed.**

١٠-٢: اتلف جميع الكائنات بشكل صريح **be explicitly destroyed** عنما لا يكون هناك حاجة الييها.

الكاتب ينصح بعدم الاعتماد على الاكسس ليقوم بالعملية نيابة عنك اذ انه من المصدق ان اسباب تسرب الذاكرة Memory Leaks يعود الي الاعتماد على الاتلاف الاوتماتيك للكائنات automatic object destruction.

وبهذا نكون قد انتهينا

للذين يرغبون بالاطلاع على الملحق باللغة الانجليزية فيمكنهم تحميله من الموقع

/www.learnaccessvba.com

تحياتي لكم جميعا

Whale