Skripta ke školení

Základy VBA

vypracoval: **Tomáš Herout**

e-mail: **herout@helpmark.cz**

tel: **739 719 548**

© 2016

Obsah

© 2016 Tomáš Herout

Trocha teorie VBA

Zkratka VBA znamená Visual Basic for Application.

Většinou se používá k:

1. Opakování stále stejných operací, kdy se dá při první operaci spustit nahrávání makra, a toto makro pomocí jazyka VBA zaznamenává operace. Kdykoli se pak dá spustit a provede totéž
2. Pro práci s formuláři, kdy uživatel může data vkládat do formuláře a až formulář vytvoří strukturu dat
3. Za účelem automatické tvorby grafů
4. Práce se soubory
5. K vytváření vlastních funkcí nad rámec funkcí, které jsou v Excelu integrovány

Způsob zápisu ve VBA

1. VBA nerozlišuje velikost písmen
2. Příkaz se ukončuje tím, že začnete psát na nový řádek (ne středníkem). V případě, že si budete chtít kód zpřehlednit zalomením na nový řádek, musíte použít podtržítko
3. Používá tzv. tečkovou notaci. Představte si ji jako: `cesko.praha`

Co je to funkce

Funkce je kus obaleného kódu, kterému dáme název. Díky tomu názvu ji můžeme opakovaně používat kdykoli to budeme potřebovat. Bude stačit tu funkci jen zavolat a sdělit jí, co po ní chceme. Tomu předávání informací obsahující to, co po funkci chceme, se odborní říká **argumenty**. Při volání funkce se argumenty zapisují do závorek. Některé funkce mohou přebírat více argumentů. Jeden argument od druhého se uvnitř závorek odděluje pomocí středníků.

Funkce nám vrací nějakou hodnotu. Třeba číslo, zda je to pravda apod.

Ukázka zavolání funkce:

```
=NAZEVFUNKCE(argument-jedna; argument-dva; …)
```

Ukázka vytvoření funkce:

```
Function NazevFunkce(argument)
' nějaký kód
End Function
```

Co je to procedura

Je to také obalený kód, který se spustí, když ho potřebujeme, ale nevrací žádnou hodnotu.

Ukázka zápisu procedury:

```
Sub Nazev()
' nějaký kód
End Sub
```


© 2016 Tomáš Herout

Základní stavební kameny VBA

Proměnné

Proměnné se používají vždy, pokud má kód pracovat s nějakými daty (třeba čísli) vloženými uživatele. Používají se v praxi i k mnoha dalším věcem. Například ke zpřehlednění kódu. U VBA je třeba nejprve určit druh proměnné, než ji v reálu použijete.

Příklad:

```
Dim NazevPromenne As Integer
NazevPromenne = 1
```

Platnost proměnné

Název	Popis
`Dim`	Platí jen dočasně po dobu, kdy se vykonává kód procedury
`Private`	Dočasná, ale je vidět ve více procedurách v jednom modulu
`Public`	Zachová si platnost i po skončení operací
`Static`	Je neměnná (když k ní něco přičtete, tak zůstane stejná)

Datové typy

Název	Popis
`Integer`	Menší čísla od -32 768 do 32 767
`Long`	Větší čísla od -2 147 483 648 do 2 147 483 647
`Single`	Desetinná místa 6 míst (např. 0,654321)
`Double`	Desetinná místa 12 míst
`Decimal`	Desetinná místa 28 míst
`Date`	Datum např. 1. 1. 2001
`String`	Textový řetězec
`Boolean`	Pravdivostní (pravda / nepravda)
`Variant`	Ostatní proměnné, třeba `Error`

© 2016 Tomáš Herout

Pole

Pole jsou něco podobného, jako proměnné. Rozdíl je v tom, že v jednom poli může být více proměnných. Každá z proměnných uvnitř pole má své číslo (index). Pozor, čísla začínají nulou. Tedy např:

```
Dim Pole(3) As Variant
Pole(0) = "Muž"
Pole(1) = "Žena"
Pole(2) = "neurčeno"
MsgBox Pole(0) & " " & Pole(1) & " " & Pole(2)
```

Křižovatky (podmínky)

IF

Už název napovídá, že se jedná o podmínku. Pokud je podmínka splněna (je pravda), tak se vykoná kód umístěný v podmínce. Příklad zápisu:

```
If x > 0 Then
     MsgBox x "> 0"
End If
```

Může se stát případ, kdy budeme chtít něco vykonat i v případě, kdy podmínka platná nebude.
K tomu nám slouží `Else`. Příklad:

```
If x > 0 Then
     MsgBox x "> 0"
Else
     MsgBox x "< nebo = 0"
End If
```

Můžeme také chtít ověřit více než jen jednu podmínku. Pak použijeme `ElseIf`.

```
If x > 0 Then
     MsgBox x "> 0"
ElseIf x = 0
     MsgBox x " = 0"
Else
     MsgBox x "< 0"
End If
```

Operátory používané u podmínek

Operátor	Popis
>	x větší než y
<	x menší než y
>=	x větší nebo rovno než y
<=	x menší nebo rovno než y
=	x rovná se y

© 2016 Tomáš Herout

`<>`	x nerovná se y
`Like`	Porovnání řetězců
`Is`	Porovnání proměnných odkazující na objekty
`And`	Spojení podmínek, např.: x > 0 And x < 100
`Or`	Buď jedno, nebo druhé x > 0 Or x < 0

IIf

V jádru totéž, co `If`, ale jednodušší zápis. Ukázka:

```
vstup = MsgBox("Chcete vstoupit?", vbYesNo, "Vstup")
MsgBox (iif(vstup = vbYes, "Vítejte a pokračujte", "Nashledanou"))
```

Select Case

Používá se ve stejných situacích, jako `If` a `Elseif`. Jeho použití je tam, kde je „křižovatka“ s mnoha odbočkami.

```
Select Case promenna
     Case pondělí
            MsgBox ("pondělí")
     Case úterý
            MsgBox ("úterý ")
     Case středa
            MsgBox ("středa ")
 End Select
```

Cykly

Cykly provádějí nějaký kód tak dlouho, dokud je to potřeba. Pokud např. chceme vypsat celé pole (více proměnných pod jedním názvem), můžeme použít cyklus.

For

Zde příklad:

```
Dim Pole(3) As Variant
     Pole(0) = "Muž"
     Pole(1) = "Žena"
     Pole(2) = "neurčeno"

Dim i As Integer
Dim ii As Integer
Dim bunka As Variant
ii = 1
For i = 0 To UBound(Pole)
    bunka = "A" & ii
    Range(bunka) = Pole(i)
    ii = ii + 1
Next i
```


© 2016 Tomáš Herout

Popis příkladu:
1. řádek vytváří pole pojmenované jako „Pole“ a definuje mu tři hodnoty
2. až 4. řádek vytváří hodnoty v polích
5. až 7. řádek vytváří proměnné
8. řádek definuje proměnnou `ii`, kterou použijeme pro vytváření dynamických adres buněk
9. řádek otevírá cyklus, stanovuje proměnnou `i` na hodnotu 0 a za `To`, vypíše počet opakování, který nejprve zjišťuje přes funkci `UBound(Pole)`.
10. řádek vytváří proměnnou `bunka`, která se při každém cyklu bude měnit
11. řádek zajistí, že se při každém cyklu zvýší proměnná `ii`
12. řádek ukončuje cyklus a vrací ho na 9. řádek

For Each

Je přímo určen k procházení polí. Má kratší zápis:

```
Dim Bun_hod As String

For Each Bun In ActiveSheet.Range("A1:A5")
Bun_hod = Bun_hod & vbNewLine & Bun.Address & " - " & Bun.Value
Next Bun

MsgBox Bun_hod
```


© 2016 Tomáš Herout

Operace s VBA

Načítání hodnot z tabulek a vkládáno do tabulek

VBA samozřejmě musí umět pracovat s tabulkami Excelu, jinak by to nedávalo smysl.

MsgBox

Otevře okno nad Excelem a vypíše do něj libovolnou hodnotu, kterou chcete.

Range(AdresaBunky)

Pokud do závorek vložíte adresu buňky (třeba „A1“), tak tuto buňku označíte a budete s ní moci pracovat, třeba do ní zapsat nějakou hodnotu. Příklad:

```
Range("A1") = "test"
```

Pokud byste naopak chtěli načíst hodnotu z nějaké buňky a uložit ji do proměnné, tak by zápis mohl vypadat zase takto:

```
Dim NacteniZBunky As Integer
NacteniZBunky = Range("A1").Value
```

Vysvětlení: První řádek jen definuje budoucí proměnnou. Udává, že bude vidět pouze v rámci procedury a pak zanikne (šetří se tím výkon). Jako druh je určené číslo. Druhý řádek ukládá do proměnné hodnotu z buňky A1. Získávat hodnoty můžeme pomocí více příkazů:

1. Value – vrátí hodnotu, ale nemusí být zcela přesná (např. zaokrouhlená)
2. Value2 – vrátí přesnější hodnotu. To se projevuje u data a měn
3. Text – vrátí přesný obsah buňky

Pokud bychom chtěli získat hodnotu z jiného listu, tak zde máte příklad:

```
Dim NacteniZBunky As Integer
NacteniZBunky = Worksheets("List1").Range("A1").Value
```

Můžeme také chtít načíst buňku, která se nachází v jiném sešitě. V tom případě by to mohlo vypadat takto:

```
Application.Workbooks.Open ("C:\adresar\sesit.xlsx")
Dim x As Integer
X = Workbooks("sesit.xlsx").Worksheets("List1").Range("A1").Value
```

Sledování událostí

Sledování událostí slouží k tomu, aby byl spuštěn nějaký kód (v proceduře) na základě nějaké události, jako je např. otevření sešitu.

```
Private Sub Workbook_Open()
    MsgBox "Vítáme vás"
End Sub
```


© 2016 Tomáš Herout