



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

DUM 06 téma: Tvorba makra pomocí VBA

ze sady:	03	tematický okruh sady:	Tvorba skript a maker
ze šablony:	10 Algoritmizace a programování	určeno pro:	4. ročník
vzdělávací obor:	18-20-M/01 Informační technologie - Aplikace osobních počítačů		
vzdělávací oblast:	odborné vzdělávání		
číslo projektu:	CZ.1.07/1.5.00/34.0066		
anotace:	DUM seznamující se základními technikami vytváření a strukturami maker kódu VBA		
metodika:	viz metodický list VY_32_INOVACE_10306ml.pdf		
datum tvorby:	15.3.2013		




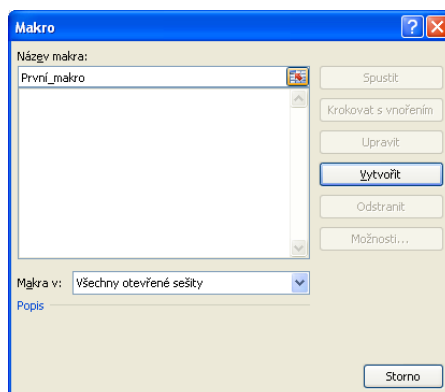
Autorem materiálu a všech jeho částí, není-li uvedeno jinak, je Ing. Daniel Velek.

Materiál je publikován pod licencí Creative Commons

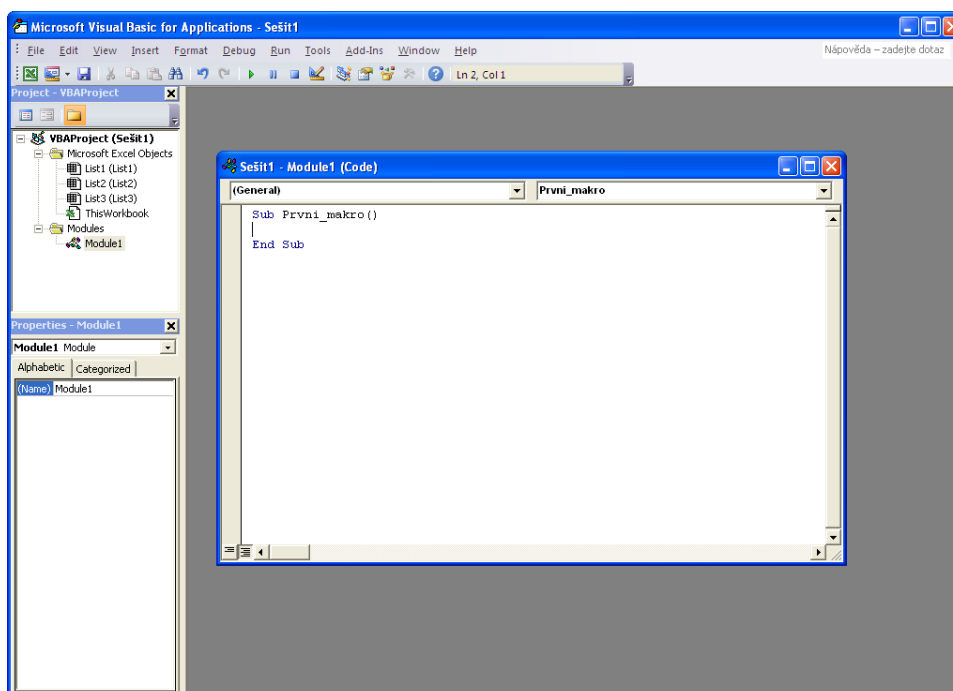


Tvorba makra pomocí VBA

V prostředí MS Excelu si otevřeme pás karet Zobrazení, kde se úplně vpravo nachází kategorie Makra. Klikneme na ikonu makra  nebo si pod tlačítkem otevřeme rozbalovací nabídku, kde vybereme položku Zobrazit makra. Otevře se následující dialogové okno, ve kterém zadáme název nového makra, v našem případě „Prvni_makro“. Na závěr klikneme na vytvořit.




Pokud máme v MS Excelu standardní nastavení, automaticky se spustí dialogové okno editoru kódu Visual Basicu. V opačném případě ho spustíme kliknutím na tlačítko upravit. Jetnotlivá makra jsou v editoru otevřena v samostatných oknech.



Jak je již vidět makro začíná klíčovým slovem Sub, za kterým následuje jeho jedinečný název respektive identifikátor. Na konci řádku jsou prázdné kulaté závorky. Kód ukončujeme pomocí klíčového slova End Sub.

Kód mezi klíčovými slovy Sub a End Sub se nazývá procedura. V proceduře je zapsán již vlastní kód, který je složen z deklarační části a kódové části. V deklarační části jsou nadefinované


Autorem materiálu a všech jeho částí, není-li uvedeno jinak, je Ing. Daniel Velek. Materiál je publikován pod licencí Creative Commons 

proměnné a v kódové části je již vlastní výkonný kód. Makro může být tvořeno z více procedur, které se navzájem mohou volat. Nyní si ukážeme vzájemné volání procedur.

```
(General)
Sub Prvni_makro()
MsgBox "Budeme volat jinou proceduru"
procedura2
MsgBox "Uz jsme v procedure Prvni_makro"
End Sub

Sub procedura2()
MsgBox "Nachazime se v procedure2"
End Sub
```

Po přepsání kódu musíme excelovský sešit nejprve uložit jako sešit s podporou maker. Nyní již lze kód spustit. Pro spuštění máme dvě možnosti:

- Tlačítko spustit v okně, kde jsme si vytvořili prázdné makro. Musíme zavřít nebo minimalizovat editor VBA.
- V editoru VBA se v panelu nástrojů nachází sada tlačítek  na spuštění a zastavování kódu. Tyto tlačítka jsou primárně určena pro ladění kódu VBA.

Popis napsaného kódu

Klíčové slovo vykreslí okno, ve kterém bude vypsáno hlášení v uvozovkách. Proceduru voláme úplně jednoduše pomocí jejího názvu viz. třetí řádek kódu. Makro vypíše hlášení: Budeme volat jinou proceduru, pak je zavolána procedura2. Procedura 2 vypíše hlášení: Nachazime se v procedure2. Procedura 2 je ukončena a vykonávání kódu je vráceno do první procedury pod volání procedury a je vypsáno hlášení: Uz jsme v procedure První_makro. Následně se vykonávání kódu ukončí.

Komentáře

V kódu je uvozujeme apostrofem tedy tímto znakem: ‘ Cokoliv za ním napíšeme nebude v kód zpracováno.

Poznámka: VBA rozlišuje velká a malá písmena tzn. Procedura2 a procedura2 jsou dvě rozdílné procedury.


Části kódu

Moduly

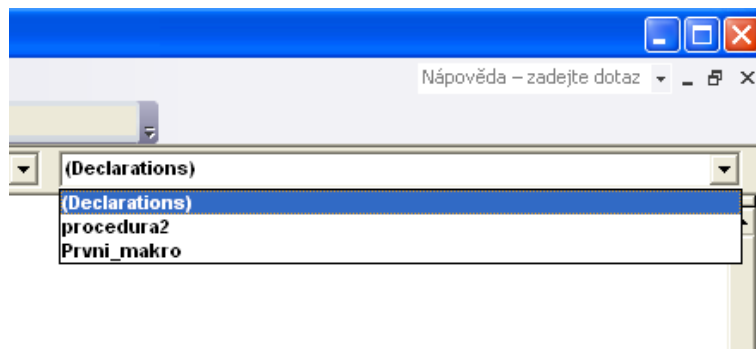
Moduly opět mohou obsahovat obecnou deklarační část, kde mohou být deklarovány proměnné, konstanty a datové typy. Kromě toho v modulu jsou zapsány jednotlivé procedury. Veškeré věci zapsané v obecné deklarační části jsou viditelné z jakékoliv procedury. Modul seskupuje několik



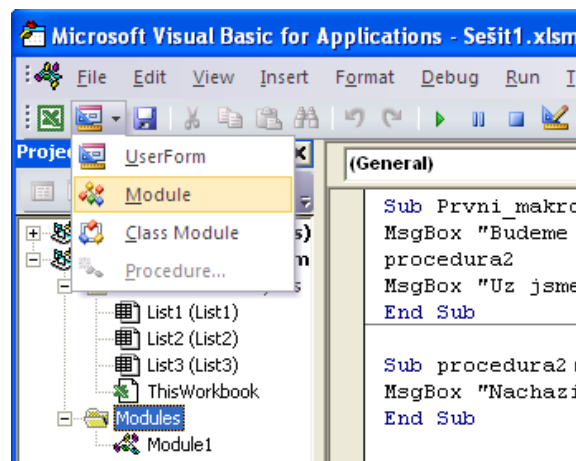
Autorem materiálu a všech jeho částí, není-li uvedeno jinak, je Ing. Daniel Velek

Material je publikován pod licencí Creative Commons 

procedur a ty spolu mohou komunikovat. Obecná deklarační část se nachází úplně na začátku kódu makra. Přepneme se na ni otevřením levé rozbalovací nabídky v okně kódu makra a vybereme položku Declarations.



V rámci tvorby kódu lze vytvořit i nový modul, který vytvoříme v editoru VBA v panelu nástrojů kliknutím na druhou ikonu zleva a z otevřené nabídky vybereme položku Module.



Funkce

Mimo procedur lze do modulů zapsat funkcionální procedury (funkce). Jedná se o části kódu, které jsou zapsány mezi dvojicí příkazů Function a End Function. Mohou být volány stejně jako klasická procedura, tedy svým jménem. Mohou přijímat parametry a **prostřednictvím svého jména vrací hodnotu**. Používají se především ve výpočtech. V závorkách jsou zapsány identifikátory, do kterých budou funkci předány hodnoty. As Double značí, jakého datového typu bude vrácená hodnota a jakého datového typu je očekávána hodnotaB. Poslední řádek příkladu je volání funkce, tedy řádek kódu volající funkci a do proměnné Vysledek je vrácena výsledná hodnota. Na rozdíl od procedur lze funkce volat pouze z procedur nebo z buňky v listu MS Excelu.

Např.: Function Nazev1 (hodnota, hodnotaB As Double) As Double()

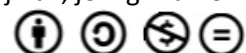
Vlastní kod

End Function



Autorem materiálu a všech jeho částí, není-li uvedeno jinak, je Ing. Daniel Velek

Material je publikován pod licencí Creative Commons



```
Vysledek.Vaulue = Nazev1 (hodnota, hodnotaB)
```

Procedury

Nyní si dopovíme další možnosti při vytváření procedur. Lze zavolat a předat jim atribut dvěma způsoby: odkazem nebo hodnotou.

Předání odkazem

Přivolání proceduře nebude předána hodnota, ale adresa argumentu tzn. proceduře je umožněn přístup k proměnné do paměti a může ji změnit. Tím jsou ovlivněny i další procedury, které s proměnnou pracují.

Volání odkazem.: `JmenoVolaneProcedury (hodnota, hodnotaB)`

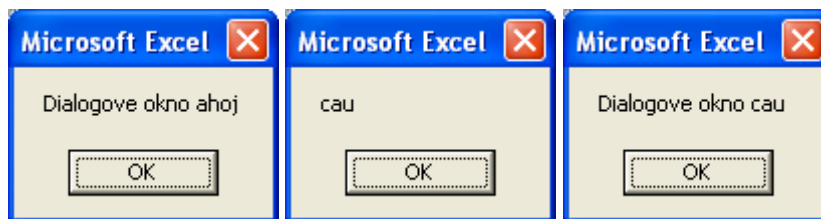
Pro předání pomocí odkazu lze v závorkách volené procedury jako identifikátor zapsat nepovinný atribut `ByRef`.

Praktický příklad na předávání odkazem

```
Sub predavani()
    Dim Hodnota As String
    Hodnota = "ahoj"
    MsgBox "Dialogove okno " & Hodnota

    predano Hodnota
    MsgBox "Dialogove okno " & Hodnota
End Sub

Sub predano(ByRef ZmenHodnota As String)
    ZmenHodnota = "cau"
    MsgBox ZmenHodnota
End Sub
```



Proměnná je předána volané proceduře a tam je její hodnota změněna na nový řetězec. Což po návratu je zobrazeno ve třetím dialogovém okně.

Předání hodnotou

Volaná procedura přijímá pouze hodnotu a tu si ukládá do své lokální proměnné. Všechny změny jsou prováděny pouze s touto lokální proměnnou.

Volání hodnotou.: `JmenoVolaneProcedury (x*0,01,120)`

Pro předání pomocí hodnoty lze v závorkách volené procedury jako identifikátor zapsat nepovinný atribut ByVal.

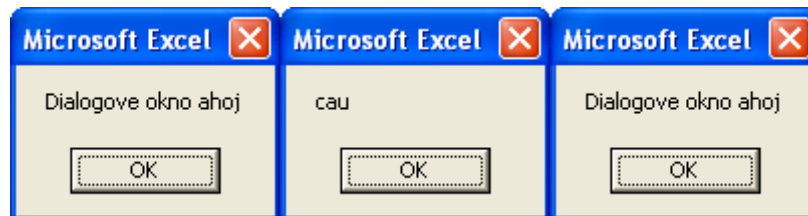
Praktický příklad na předávání hodnotou

```
Sub predavani()
    Dim Hodnota As String
    Hodnota = "ahoj"
    MsgBox "Dialogove okno " & Hodnota

    predano Hodnota
    MsgBox "Dialogove okno " & Hodnota

End Sub

Sub predano(ByVal ZmenHodnota As String)
    ZmenHodnota = "cau"
    MsgBox ZmenHodnota
End Sub
```



Výsledkem činnosti těchto dvou procedur jsou zobrazená dialogová okna. Z těchto dialogových okne plyne, že změna provedená s hodnotou proměnné ve volané proceduře a tato změna platí jen v ní. Ve volající proceduře ke změně hodnoty nedochází a okno zobrazuje stále stejný text.

Životnost hodnot

Při deklaraci procedury lze před klíčové slovo Sub zapsat klíčová slova Private nebo Static.

Private

Procedura je deklarována jako soukromá, patří k modulu, na který je vázána. Její proměnné jsou uchovávány po dobu běhu procedury.

Static

Obsahuje statické proměnné, jejichž hodnota je zachována i pro následující procedury, kde je definována.

Zdroje:

Archiv autora



Autorem materiálu a všech jeho částí, není-li uvedeno jinak, je Ing. Daniel Velek

Material je publikován pod licencí Creative Commons 