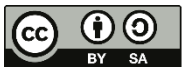
VLASTNÍ MAKRA

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY



OBSAH PŘEDNÁŠKY

Něco málo o makrech

Parametry maker a jejich předávání

Spouštění maker

Komunikace maker s uživateli pomocí vestavěných oken

Ošetřování chyb

Uznávaná praxe doporučuje

NĚCO MÁLO O MAKRECH

- makro je procedura Sub, která se používá k automatizování činností v aplikaci Excel
- makro na rozdíl od funkce
  - může měnit vlastnosti a volat metody objektů, které mění stav objektového modelu Excelu (např. může formátovat buňky, přesouvat listy, otevírat soubory, ovládat dialogová okna a jejich části, měnit místní nabídky apod.)
  - nemůže nic vracet
  - nemůže být voláno z buňky (vzorce) pracovního listu
  - může komunikovat s uživateli prostřednictvím dialogových oken

PARAMETRY MAKER A JEJICH PŘEDÁVÁNÍ

- parametry poskytují data, která makro ve svém těle zpracovává
- makro
  - nemusí vyžadovat žádné parametry
  - může očekávat pevně stanovený počet parametrů
  - může přijímat předem neurčený počet parametrů
  - může některé parametry vyžadovat a některé může mít nepovinné
  - může mít všechny parametry nepovinné
- parametry se mohou předávat dvěma způsoby:
  - (implicitně) odkazem
  - hodnotou

    parametr, v hlavičce makra uvozený slovem ByVal, předá makru kopii proměnné, konstanty, pole nebo objektu ⇒ makro s nimi bude pracovat, ale neovlivní jejich původní hodnotu

SPOUŠTĚNÍ MAKER

- makra lze spouštět
  - z okna Editoru jazyka VB pomocí nabídky Run / Run Sub / UserForm
  - z podokna Immediate Editoru jazyka VB
  - z jiné vlastní procedury
  - z okna Excelu pomocí dialogového okna Vývojář / Kód / Makra*
  - z okna Excelu pomocí klávesové zkratky*
  - klepnutím na ovládací prvek *ActiveX*, formulářový ovládací prvek, obrázek, klipart, tvar, SmartArt, graf, WordArt nebo jiný vložený objekt*
  - z panelu Rychlý přístup*
  - prostřednictvím tlačítek nebo jiných ovládacích prvků na pásu karet*
  - z libovolné místní nabídky Excelu*
  - jako reakci na výskyt nějaké události

* pouze veřejná makra bez parametrů

SPOUŠTĚNÍ MAKER Z VLASTNÍCH PROCEDUR

- z jiných procedur lze spouštět makro
  - pomocí klíčového slova Call podle vzoru
    Call *Makro(Parametr1,Parametr2,...)*
  - pomocí metody Run objektu Application podle vzoru
    Application.Run "*Makro*",*Parametr1,Parametr2,...*
- má-li se volat makro umístěné v jiném modulu téhož projektu, je nutné před název makra přidat název modulu podle vzoru
  Call *Modul.Makro(Parametr1,Parametr2,...)*
  Application.Run "*Modul.Makro*",*Parametr1,Parametr2,...*
- má-li se volat makro umístěné v jiném otevřeném (!) sešitu, je nutné použít metodu Run objektu Application a před název makra přidat název sešitu podle vzoru
  Application.Run "'*Sešit*.xlsm'!*Makro*",*Parametr1,Parametr2,...*

SPOUŠTĚNÍ MAKER Z VLASTNÍCH PROCEDUR

- má-li se volat makro, jehož název je uložen v *Proměnné* typu String, je nutné použít vždy metodu Run objektu Application podle vzoru Application.Run *Proměnná,Parametr1,Parametr2,...*

SPOUŠTĚNÍ MAKER POMOCÍ KLÁVES. ZKRAT

- veřejné makro bez parametrů lze spouštět pomocí klávesové zkratky*, kterou lze přidělit
  - před spuštěním nahrávání pomocí dialogu Vývojář / Kód / Záznam makra
  - pomocí dialogu Vývojář / Kód / Makra
  - v kódu pomocí metody MacroOptions objektu Application podle vzoru Application.MacroOptions Macro:="*Makro*" HasShortcutKey:=True _ ShortcutKey:="*klávesa*"
- kombinace kláves CTRL+E, J, M a Q nejsou v Excelu obsazeny; naopak obsazeny jsou kombinace CTRL+SHIFT+F, L, N, O, P a W
- makro, které bude mít přiřazenu klávesovou zkratku obsazenou Excelem, bude mít přednost

* malé písmeno odpovídá kombinaci CTRL+*klávesa*, velké písmeno kombinaci CTRL+SHIFT+*klávesa*

KOMUNIKACE MAKER S UŽIVATELEM

- je-li třeba, makro může s uživatelem komunikovat – může od uživatele informace vyžadovat (vstupní data, potvrzení, …) nebo může uživateli informace sdělovat (výsledky, chyby, …)
- ke komunikaci s uživatelem lze využít vlastní dialogová okna *UserForms* nebo dialogová okna generovaná vestavěnými nástroji VBA:
  - k zadání vstupní hodnoty lze využít okno, které generuje funkce InputBox nebo metoda Application.InputBox
  - ke sdělení informace (textu, hodnot proměnných a vlastností objektů) anebo k položení dotazu lze využít okno, které generuje funkce MsgBox
- okna InputBox jsou aplikačně modální a okna MsgBox aplikačně a příp. i systémově modální, tj. okna je nutné uzavřít, aby mohl pokračovat kód makra ve vykonávání a uživatel v ovládání Excelu příp. operačního systému

FUNKCE INPUTBOX

Proměnná = InputBox(Prompt:="*Výzva*", Title:="*TitulekOkna*")

- generuje dialogové okno implicitně s jedním vstupním polem, tlačítky OK, Cancel a volitelně tlačítkem Nápověda
- umožňuje uživateli vložit hodnotu z klávesnice
- *Proměnná* je typu String ⇒ pokud uživatel vloží číslo, musí být převedeno na skutečné číslo pomocí funkcí pro konverzi datových typů
- pokud uživatel stiskne Cancel, vrací prázdný řetězec
- více viz on-line nápověda pod heslem InputBox Function

METODA APPLICATION.INPUTBOX

Proměnná = Application.InputBox(Prompt:="*Výzva*", Title:="*TitulekOkna* ", _
Type:=*Číslo*)

- generuje dialogové okno implicitně s jedním vstupním polem, tlačítky OK a Storno a volitelně tlačítkem Nápověda
- umožňuje uživateli vložit hodnotu z klávesnice i odkazem na buňku, a to datového typu, který lze specifikovat v číselném parametru Type (je-li třeba povolit více typů, jeho hodnota bude dána součtem čísel pro jednotlivé typy)
- datový typ se během zadávání automaticky ověřuje; v případě selhání se zobrazí okno s varováním, které dovoluje uživateli zadání hodnoty správné typu zopakovat
- pokud uživatel stiskne Storno, metoda vrací logickou hodnotu False
- více viz on-line nápověda pod heslem InputBox Method

FUNKCE MSGBOX

Proměnná = MsgBox(Prompt:="*SděleníNeboDotaz*",
Buttons:=vbQuestion+vbYesNo+ _ vbDefaultButton2, Title:="*TitulekOkna*")

- generuje dialogové okno implicitně bez ikony a s tlačítkem OK, volitelně s jednou ze čtyř ikon a až čtyřmi různými tlačítky (ikonu, tlačítka a výchozí tlačítko lze specifikovat pomocí čísel nebo názvů ekvivalentních VBA konstant, které se sčítají v parametru Buttons)
- *Proměnná* je typu Integer a lze podle její hodnoty zjistit, kterým tlačítkem bylo dialogové okno uzavřeno (není-li nutné *Proměnnou* v kódu dále zpracovávat, lze funkci volat bez uložení návratové hodnoty a bez kulatých závorek)
- více viz v on-line nápověda pod heslem MsgBox Function

OŠETŘOVÁNÍ CHYB BĚHU PROGRAMU

- během provádění procedur VBA se mohou vyskytnout chyby – chyby za běhu programu [*run-time errors*]
- dojde-li k chybě za běhu programu, VBA za normálních okolností
  - přiřadí chybě číslo, číslo uloží do vlastnosti Number a anglický popis chyby do vlastnosti Description objektu Err
  - ukončí provádění kódu
  - zobrazí dialogové okno s číslem a anglickým popisem chyby
- každou chybu za běhu programu lze programově zachytit a poté (není-li závažná) ignorovat anebo (je-li závažná) zpracovat na místě nebo ve speciální části kódu – chybové rutině [*error handler*]
- k zachycení chyby, definování způsobu zpracování chyby a způsobu pokračování ve vykonávání kódu lze použít objekt Err a/nebo kombinaci příkazů On Error, Resume, GoTo a Next

OŠETŘOVÁNÍ CHYB BĚHU PROGRAMU

- příkazy On Error, Resume a Exit Sub vlastnosti objektu Err automaticky nulují nebo nastavují na řetězec nulové délky (""); nebudou-li bezprostředně po ošetření chyby tyto příkazy vykonány, je nutné »zresetovat« objekt Err metodou Clear »ručně«

OŠETŘOVÁNÍ CHYB BĚHU PROGRAMU

Kód, který chybu ignoruje:

```
Sub
    ...
    On Error Resume Next          'nastaví, aby se chyba přeskočila
    'příkaz, který generuje chybu
    'příkaz, kterým makro pokračuje
    ...
    'příkaz, který generuje chybu
    'příkaz, kterým makro pokračuje
    ...
    On Error GoTo 0               'obnoví výchozí zpracování chyb
    ...
End Sub
```

OŠETŘOVÁNÍ CHYB BĚHU PROGRAMU

Kód, který pokračuje následujícím příkazem, který chybu ošetří:

```
Sub
    ...
    On Error Resume Next          'nastaví, aby se chyba přeskočila
    'příkaz, který generuje chybu
    If Err.Number <> 0 Then
        'příkazy, které se provedou v případě chyby
        Err.Clear                 'vynuluje objekt Err
    End If
    'příkaz, kterým makro pokračuje
    ...
    On Error GoTo 0               'obnoví výchozí zpracování chyb
    ...
End Sub
```

OŠETŘOVÁNÍ CHYB BĚHU PROGRAMU

Kód, který odskočí do rutiny na konci procedury, která chybu ošetří a pokračuje dalším řádkem:

```
Sub
    ...
    On Error GoTo ErrorHandler    'odskočí do rutiny s návěštím
       „ErrorHandler“
    'příkaz, který generuje chybu
    'příkaz, kterým makro pokračuje
    ...
    On Error GoTo 0    'obnoví výchozí zpracování chyb
    Exit Sub
ErrorHandler:    'návěští chybové rutiny
    'příkazy, které se provedou v případě chyby
    Resume Next    'vrátí se na následující řádek za ten, který generoval
       chybu
End Sub
```

OŠETŘOVÁNÍ CHYB BĚHU PROGRAMU

Kód, který odskočí do rutiny na konci procedury, která chybu ošetří a zopakuje příkaz na řádku s chybou:

```
Sub
    ...
    On Error GoTo ErrorHandler    'odskočí do rutiny s návěštím
       „ErrorHandler“
    'příkaz, který generuje chybu a který se po ošetření chyby zopakuje
    ...
    On Error GoTo 0    'obnoví výchozí zpracování chyb
    Exit Sub
ErrorHandler:      'návěští chybové rutiny
    'příkazy, které se provedou v případě chyby
    Resume      'vrátí se na příkaz, který chybu vygeneroval
End Sub
```

UZNÁVANÁ PRAXE DOPORUČUJE

- kvůli přehlednosti a údržbě kódu psát makra
  - modulárně, tj. vytvářet malá a jednoduchá makra oproti velkým a složitým a členit je do více modulů
  - strukturovaně, tj. s jedním vstupním bodem (Sub, Function), jedním výstupním bodem (End Sub, Exit Sub, End Function, Exit Function) a tělem, které se vykonává systematicky řádek po řádku ⇒ použití příkazu pro odskok GoTo praxe toleruje jen v případě ošetřování chyb za běhu programu!!!
- kód optimalizovat, tzn. používat takové způsoby zápisu, které kód zrychlují – viz doporučený informační zdroj Optimize Slow VBA Code. Speeding Up Slow Excel VBA Code

UZNÁVANÁ PRAXE DOPORUČUJE

- každou chybu běhu programu programově ošetřit tak, aby se uživatel se standardním hlášením VBA nikdy nesetkal anebo aby byl alespoň upozorněn uživatelsky přívětivější zprávou než je např. hlášení *Error 1043: Object is not defined…*