Vlastní funkce

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

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

OBSAH PŘEDNÁŠKY

Něco málo o funkcích

Deklarace funkcí

Příklady

Ošetřování chyb

Na co si dát pozor

Základní pojmy

1. funkce pracovního listu

   tyto funkce Excelu není možné modifikovat

2. vlastní funkce

   procedury Function (ještě existují procedury Sub)

   - jsou uživatelské funkce, které si vytváříme sami pomocí programovacího jazyka VBA
   - vlastní funkce si můžeme napsat sobě na míru, pro vlastní potřebu
   - vlastní funkci je možné
     1. použít v pracovním listu
     2. nebo volat z jiné funkce či procedury
   - funkce vždy vrací hodnotu nebo pole hodnot
   - vlastní funkci je nutné umístit do standardního modulu VBA (Insert / Module), když ji dáte někam jinam, nebude fungovat
   - projekt VBA je součástí sešitu

Základní pojmy

- upozornění
  - pomocí vlastní funkce, kterou budete používat na pracovním listu, není možné manipulovat s objekty, prostředím, buňkami apod. a to ani odkazem na jinou proceduru
  - ALE můžete např. zjišťovat vlastnosti objektů a použít je v programu
  - vlastní funkci je možné testovat na listu nebo pomocí jiné procedury
  - volá se stejně jako funkce pracovního listu
  - nachází se v kategorii VLASTNÍ, pokud si ji nezařadíte do jiné kategorie
- použití
  - jako součást výrazů v procedurách VBA
  - ve vzorcích na pracovních listech
- kdy?
  - pro často se opakující výpočty

Deklarace funkce

- klíčové slovo Function, jméno funkce, v kulatých závorkách seznam parametrů

  [Public | Private | Friend] [Static] Function nazev([param])[As typ]

  [prikazy]

  [nazev = …]

  [Exit Function]

  [prikazy]

  nazev =…

  End Function

- proměnné deklarované uvnitř funkce jsou lokální, pokud není definováno jinak

Parametry funkce

- funkce nemusí mít parametry
- funkce může mít pevně daný počet povinných parametrů (1-60)
- funkce může kombinovat povinné a nepovinné parametry
- parametry mohou být
  - konstanty
  - proměnné
  - pole
  - výrazy …

Předávání parametrů

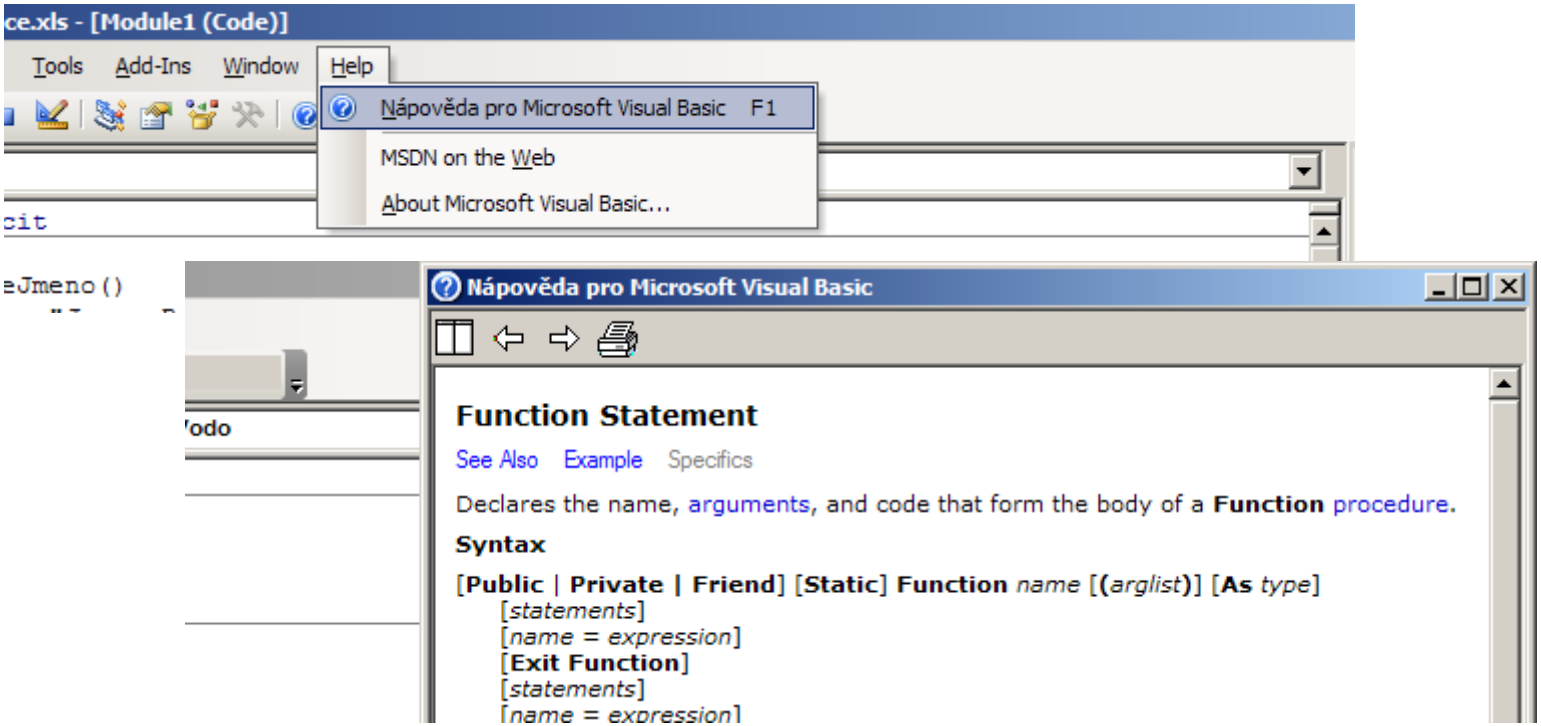
- předávání parametrů hodnotou
  - je potřeba nastavit pomocí klíčového slova ByVal
  - předává proceduře kopii původní proměnné, tzn. že pokud uvnitř procedury provedete nějaké změny parametru, neovlivní to původní proměnnou
- předávání parametrů odkazem
  - je ve VBA implicitní
  - předá funkci adresu proměnné v paměti
  - pro lepší čitelnost složitějšího kódu používejte klíčové slovo ByRef
- vyvarujte se použití funkcí ve složitějších aritmetických výrazech, pokud funkce mění hodnoty proměnných v tomto výrazu
- udržujte pořádek ve jménech proměnných, parametrů, konstant, funkcí, procedur a modulů, při použití stejného jména může dojít ke konfliktu

Volání funkce v kódu

- funkce se volá jménem, za kterým následuje seznam parametrů v kulatých závorkách
  `MojeFunkce(ar1, ar2,...)`
- pomocí klíčového slova `Call`
  `Call MojeFunkce(ar1, ar2,...)`

Nápověda

- v editoru VBA
- např. Function statement
- příklad (example)



Příklady funkcí

- funkce bez parametrů

```
Function MojeJmeno() As String
    MojeJmeno="Jmeno Prijmeni"
End Function
```

- funkce s jedním parametrem

```
Function TCisla(cislo As Integer) As String
  If cislo Mod 2=0 Then
    TCisla="sudé"
  Else
    TCisla="liché"
  End If
End Function
```

Příklady funkcí

- funkce s volitelným parametrem

```
Function Umocnit(c As Double, optional n)As Double
        If IsMissing(n) Then
            'není-li zadán parametr n
            Umocnit=c*c
             Else
            Umocnit=c^n
         End If
 End Function
```

- funkce vracející pole hodnot

```
Function FakVodo()
        Dim Fakulty()
        Fakulty = Array("FCHT","FTOP"…)
        FakVodo = Fakulty
 End Function
```

- pro svislé pole je třeba použít WorksheetFunction.Transpose() a CTRL– Shift – Enter

Příklady funkcí

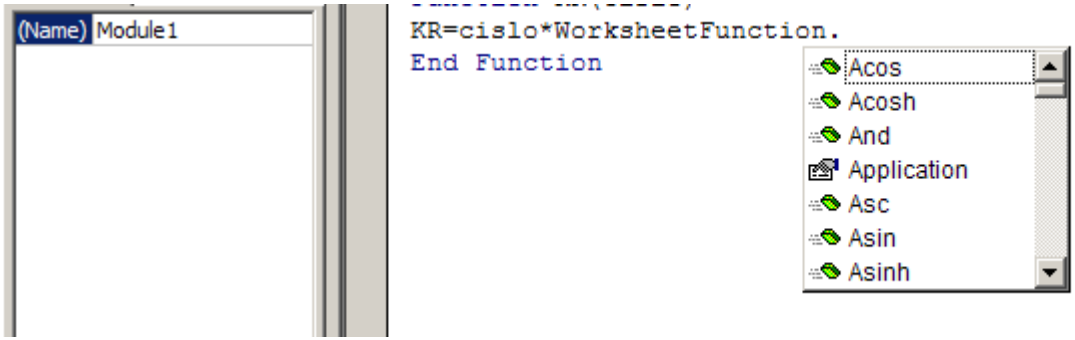
- funkce, kde parametrem je pole

```
Function Pocet(vstup As Range)
    Dim a
    …
    For Each a In vstup
      If a.Value = 1 Then
      …
      End If
    Next a

End Function
```

Funkce listu, matematické funkce ve VBA

1. WorksheetFunction



2. Matematické funkce ve VBA
   v nápovědě Math Functions
   Abs, Atn, Cos, Exp, Fix, Int, Log, Rnd, Sgn, Sin, Sqr, Tan,
   Derived Math Functions

Chybové hodnoty

- funkce pracovních listů mohou vracet jednu z těchto chybových hodnot:
  - #DIV/0!
    - dělení nulou
    - i prázdná buňka je považována za nulovou
  - #NA
    - chybějící údaje
    - vyhledávací funkce nenajdou shodu (např. SVYHLEDAT)
  - #NÁZEV? / #NAME?
    - vzorec obsahuje název buňky nebo oblasti, který není definován
    - vzorec obsahuje text, který Excel vnímá jako nedefinovaný název (překlep v názvu funkce, chybná syntaxe)
  - #NULL!
  - když se vzorec pokusí použít průnik dvou oblastí a tento průnik je prázdný
  - operátorem průniku je mezera, viz =SUMA(B5:B15 A3:F3)

Chybové hodnoty

- funkce pracovních listů mohou vracet jednu z těchto chybových hodnot:
  - #NUM!
    - předáváte nečíselný parametr funkci tam, kde funkce čeká číselný parametr
    - funkce, která je založena na iteračních výpočtech, nekonverguje
    - vzorec vrací příliš velkou nebo příliš malou hodnotu
  - #REF!
    - když vzorec pracuje s neplatným odkazem na buňku
  - #HODNOTA! / #VALUE!
    - parametr funkce má nesprávný datový typ
    - parametr je oblast, ale má to být jedna buňka
    - nebylo stisknuto Ctrl-Shift-Enter u maticového vzorce
    - vlastní funkce není přepočítaná

Konstanty chybový hodnot uvnitř VBA

xlErrDiv0
xlErrNA
xlErrName
xlErrNull
xlErrNum
xlErrRef
xlErrValue

CVErr() je funkce vracející kód chyby

Funkce listu pro zjištění typu hodnoty nebo od

JE.PRÁZDNÉ(hodnota)
JE.CHYBA(hodnota)
JE.CHYBHODN(hodnota)
JE.LOGHODN(hodnota)
JE.NEDEF(hodnota)
JE.NETEXT(hodnota)
JE.ČISLO(hodnota)
JE.ODKAZ(hodnota)
JE.TEXT(hodnota)
CHYBA.TYP(chyba)

POZOR! ve VBA mají tyto funkce jména v angličtině!

Např. WorksheetFunction.IsNumber(arg)

Funkce VBA vhodné pro ošetřování chyb

IsArray(varname)
IsDate(expression)
IsError(expression)
IsEmpty(expression)
IsMissing(argname)
IsNull(expression)
IsNumeric(expression)

Příklad

```
Function deleni(a As Double, b As Double) as Variant

    If b = 0 Then
      deleni = CVErr(xlErrDiv0)
      Exit Function
    End If

    deleni = a / b

End Function
```

Ladění funkcí

- chyby běhu funkce jako programu VBA se nezobrazují v okně s chybovým hlášením
- pokud dojde k chybě běhu funkce, vzorec s funkcí vrátí chybovou hodnotu #VALUE! / #HODNOTA! do buňky listu, odkud je funkce volána (POZOR! to je velmi zrádné)

Doporučený postup ladění:

- otestujte si vlastní funkci voláním z procedury Sub namísto volání ze vzorce na pracovním listu (chyby běhu funkce se budou zobrazovat v okně s chybovým hlášením)
- umístěte příkaz Debug.Print na důležitá místa kódu vlastní funkce kvůli sledování hodnot proměnných a vlastností objektů (příkaz Debug.Print promenna1, promenna2, … vypisuje hodnoty do okna ladění editoru VBA)
- vložte do těla vlastní funkce zarážku (breakpoint) a funkci krokujte příkazem editoru Debug / Step…

Než odevzdáte projekt

- NE vaše chybové hlášky do buněk tabulky
- ANO chybová hlášení Excelu
- POZOR na hlášení v buňce #VALUE! / #HODNOTA!, může to znamenat chybu běhu programu
- jestliže potřebujete, aby funkce vracela v nějaké situaci chybové hlášení, pak je potřeba dát vracenou hodnotu správného datového typu
- vstupní údaje
  - mohou být nulové?
  - mohou být záporné?
- NE MsgBox!!!