



VBA Excel v příkladech

Lenka Forstová

Obsah

Odborný obsah knihy VBA Excel v příkladech	6
Funkce	6
Makra	8
Buňky	9
Aplikace	11
Slovo autora	12
Vysvětlivky k použitým prvkům v knize	12
Funkce	16
(F1) První funkce	16
(F2) Sekundy	18
(F3) Jméno a příjmení	19
(F4) Muž, nebo žena?	20
(F5) Rodné číslo	22
(F6) Obrácený text	24
(F7) Adresářová cesta	25
(F8) Náhodná čísla	27
(F9) Náhodná čísla podruhé a lépe	28
(F10) Maximální cifra	30
(F11) Převod dvojkového čísla	31
(F12) Kontrola SPZ	33
(F13) Kontrola data	36
(F14) Průměr podle znaménka	37
Makra	42
(M1) Rámečky	42
(M2) Rostoucí řetězec – pět hvězdiček	44
(M3) Rostoucí řetězec – volitelný počet hvězdiček	46
(M4) Vlož listy	48
(M5) Seznam sešitů	50
(M6) Skrýt listy	51
(M7) Listy podle abecedy	53
(M8) Odstranění listu – ošetření chyby	55
Buňky	58
(B1) Součet z největších čísel	58
(B2) Zápis vzorců	60
(B3) Oprava čísla	62
(B4) Číslo z číslic	64



(B5) Odkazy na buňky.....	66
(B6) Očíslovaná abeceda.....	67
(B7) Velká písmena.....	68
(B8) Skryj označené sloupce.....	70
(B9) Jednoznačnost.....	72
(B10) Jednoznačnost II.....	73
(B11) Jednoznačnost III.....	75
(B12) Jednoznačnost IV.....	76
(B13) Jednoznačnost V.....	77
(B14) Jednoznačnost VI.....	78
(B15) Jednoznačnost VII.....	79
(B16) Okresy.....	81
(B17) Okresy II.....	83
(B18) Kopie řádku.....	85
(B19) Kopie řádku lépe.....	87
Aplikace.....	90
(A1) Import z textového souboru.....	90
(A2) Export do TXT na zadané pozice.....	93
(A3) Export na pozice jinak.....	96
(A4) Databáze kontaktů.....	99
(A5) Vylepšení databáze.....	104
(A6) Tisk vizitky.....	108
(A7) Velký bratr.....	110
Rejstřík.....	113
A co dál	116



Odborný obsah knihy VBA Excel v příkladech

Zde je seznam jednotlivých kapitol. První řádek nastiňuje zadání příkladu, který se v kapitole řeší, kurzívou jsou napsány pojmy, jež jsou na daném příkladu procvičeny. Pro přehlednost obsahu jsou uvedeny přesné odborné termíny, které mohou být v textu pro čtenáře zjednodušeny.

Funkce

Popisuje programování vlastních funkcí v Excelu a jejich zařazení do systému funkcí použitelných k výpočtům v tabulkách.

F1. První vlastní funkce

- VBA modul, hlavička funkce, návratová hodnota, použití vlastní funkce na listu
- Přiřazovací příkaz
- Datové typy, typ návratové hodnoty

F2. Převod časového údaje na sekundy

- Datový typ Integer, Long, Date
- Přejmenování modulu

F3. Jméno a příjmení

- Práce s řetězci, operátor zřetězení
- Přeražení vlastní funkce mezi funkce textové
- Okno editoru Immediate

F4. Pohlaví z rodného čísla

- Deklarace proměnné
- Práce s řetězcem, funkce MID, TRIM
- Podmíněný příkaz
- Automatická konverze – číslo vs. řetězec

F5. Datum narození z rodného čísla

- Délka řetězce
- Typ Date, funkce pro konverzi
- Vlastní krátká nápověda k funkci
- Komentář v kódu

F6. Obrácený text

- Cyklus For
- Direktiva Option Explicit

F7. Adresářová cesta

- Pole parametrů ParamArray, LBound, UBound
- Typ **Variant**, deklarace proměnných na stejné řádce – varování

F8. Náhodná čísla

- Náhodné celé číslo z určitého intervalu
- Aktualizace výpočtu v buňkách, vlastnost funkce Volatile
- Pokračování příkazu na dalším řádku

F9. Dva typy generování náhodného čísla

- Nepovinný parametr vlastní funkce, defaultní hodnota nepovinného parametru
- Funkce IsMissing
- Typ Boolean

F10. Maximální cifra

- Operátor Like

F11. Převod dvojkového čísla

- Sestupný For cyklus
- Využití funkce MaximalniCifra, direktiva Private
- Exit Function

F12. Kontrola SPZ

- Porovnání řetězců, kódování textových vzorů pro operátor Like
- Direktiva Option Compare

F13. Kontrola data

- Kontrola všech buněk v oblasti, cyklus For Each, parametr typu Range, hodnota buňky, operátor negace NOT

F14. Průměr kladných a záporných hodnot

- Chyba jako návratová hodnota

Funkce

Makra

Buňky

Aplikace

Rejstřík



Makra

Od nahrání jednoduchých maker jako posloupnosti příkazů přes jejich upravování až po psaní kódu vlastních jednodušších maker

M1. Rámečky

- Záznamník maker, prohlížení nahraného, změna klávesové zkratky

M2. Hvězdičky pod sebe

- Relativní adresy při nahrávání
- Pozorování nahrávání „on-line“, čtení a porozumění kódu

M3. Volitelný počet hvězdiček

- Volitelný počet hvězdiček
- Úprava nahraného makra
- Funkce InputBox
- Ladění makra (krokování, sledování hodnot proměnné, ...)

M4. Přidej nové listy

- Nahrání makra a doplnění ošetření vstupních hodnot
- Do Loop cyklus
- Funkce IsNumeric, Val

M5. Seznam sešitů

- Kolekce objektů, For Each cyklus, Workbooks
- MsgBox, vbNewLine, Workbook.Name

M6. Skrýt listy

- Indexace kolekce, Sheets.Count, For cyklus a kolekce
- Visible, xlHidden, xlVeryHidden

M7. Seříd' listy podle abecedy

- Algoritmus třídění
- Worksheet.Move
- Přiřazení hodnoty parametru jménem (:=), parametry funkce vs. podprogramu

M8. Odstranění listu

- Ošetření chyb – On Error GoTo ..., objekt Err
- DisplayAlerts

Buňky

Různé způsoby práce s oblastí buněk a jejich adresace

B1. Součet největších čísel

- Deklarace pole, ReDim
- WorksheetFunction

B2. Zápis vzorce do buňky

- Formula, FormulaLocal
- Selection

B3. Oprava čísla

- Nadbytečnost kódu při použití záznamníku maker
- Konstrukce With ... End With

B4. Složení čísla z číslic

- Range.Offset

B5. Odkazy na buňky

- Objektová hierarchie, rodičovské objekty
- ActiveCell, ActiveSheet, ...

B6. Očíslovaná abeceda

- Kolekce Worksheet.Cells
- Funkce Asc a Chr, For cyklus s abecedou

B7. Převod na velká písmena

- UCase, LCase
- Kolekce Range.Cells, Range.Rows, Range.Columns

Funkce

Makra

Buňky

Aplikace

Rejstřík



B8. Skryj označené sloupce

- Přiřazovací příkaz pro objekty (Set)
- EntireRow, EntireColumn
- Defaultní vlastnost objektu – varování

B9. Jednoznačnost ve vybrané oblasti

- Cells s jedním parametrem

B10. Kontrola třetího sloupce v pojmenované oblasti

- Range.Columns(i)

B11. Kontrola sloupce v souvislé oblasti

- Range.CurrentRegion

B12. Kontrola sloupce v použité oblasti

- Worksheet.UsedRange

B13. Kontrola aktuálního sloupce

- Průnik oblastí – Intersect

B14. Kontrola vícenásobné oblasti

- Kolekce Areas

B15. Kontrola souvislého úseku

- Range.End(xlUp), ...
- Vymezení oblasti Range (odkud, kam)

B16. Kontrola číselníku okresů

- Algoritmus vyhledávání v oblasti
- Range.Find, Nothing

B17. Kontrola číselníku v jiném sešitě

- Workbooks.Open, Workbook.Path, Workbook.Activate
- Range.Parent



B18. Kopie řádku – vybrané sloupce na jiné nové pozice

- Zřetězení objektů typu Range, čitelnost kódu

B19. Kopie řádku – vylepšení

- Nezávislost pozice tabulky na listu

Aplikace

Programování jednoduchých aplikací a jejich komunikace s uživatelem – dialogová okna, uživatelské formuláře, reakce na události...

A1. Import textového souboru

- Vestavěný dialog GetOpenFileName, filtr pro typy souborů
- Workbooks.OpenText, FieldInfo

A2. Export do TXT na uživatelem dané pozice (doplnění mezer)

- Metoda InputBox, kontrola typu vstupní hodnoty
- Operátor Is
- GetSaveAsFilename
- Zápis a práce s textovým souborem, Open, Print, Close, ...

A3. Jiný způsob zápisu na pozice v TXT souboru

- Print Tab(n)
- Spouštění makra příkazovým tlačítkem, reakce na události

A4. Databáze kontaktů

- Uživatelský formulář

A5. Vylepšení databáze kontaktů

- Kontrola údajů
- ListBox, doplnění položek za běhu programu

A6. Tisk vizitky z databáze

- Kolekce vestavěných dialogů – Dialogs, Dialogs(xlDialogPrint)
- ObjectBrowser

A7. Sledování změn v tabulce

- Reakce na události Worksheet.Change, Workbook_Open, BeforeSave

Funkce

Makra

Buňky

Aplikace

Rejstřík

<p>Funkce ve VBA Excelu</p> <ul style="list-style-type: none"> • První vlastní funkce • Sekundy • Jméno a příjmení • Muž, nebo žena? • • Rodné číslo • Obrácený text • Adresářová cesta • Náhodná čísla • • Náhodná čísla podruhé a lépe • Maximální cifra • Převod dvojkového čísla • • Kontrola SPZ • Kontrola data • Průměr podle znaménka • 	<p>1</p> <p>Funkce</p>
<p>Makra ve VBA Excelu</p> <ul style="list-style-type: none"> • Rámečky • Vlož listy • Seznam sešitů • Skrýt listy • Listy podle abecedy • • Rostoucí řetězec – pět hvězdiček • Rostoucí řetězec – volitelný počet hvězdiček • • Odstranění listu – ošetření chyby • 	<p>2</p> <p>Makra</p>
<p>Buňky ve VBA Excelu</p> <ul style="list-style-type: none"> • Součet z největších čísel • Zápis vzorců • Oprava čísla • Číslo z číslic • • Odkazy na buňky • Očíslovaná adresa • Velká písmena • Skryj označené sloupce • • Jednoznačnost • Okresy • Kopie řádku • Kopie řádku lépe • 	<p>3</p> <p>Buňky</p>
<p>Aplikace</p> <ul style="list-style-type: none"> • Import z textového souboru • Export do TXT na zadané pozice • • Export na pozice jinak • Databáze kontaktů • Vylepšená databáze • • Tisk vizitky • Velký bratr • 	<p>4</p> <p>Aplikace</p>
<p>Rejstřík</p>	<p>5</p> <p>Rejstřík</p>

1

Funkce ve VBA Excelu

- První vlastní funkce • Sekundy • Jméno a příjmení • Muž, nebo žena? •
- Rodné číslo • Obrácený text • Adresářová cesta • Náhodná čísla •
- Náhodná čísla podruhé a lépe • Maximální cifra • Převod dvojkového čísla •
- Kontrola SPZ • Kontrola data • Průměr podle znaménka •

Funkce

Makra

Buňky

Aplikace

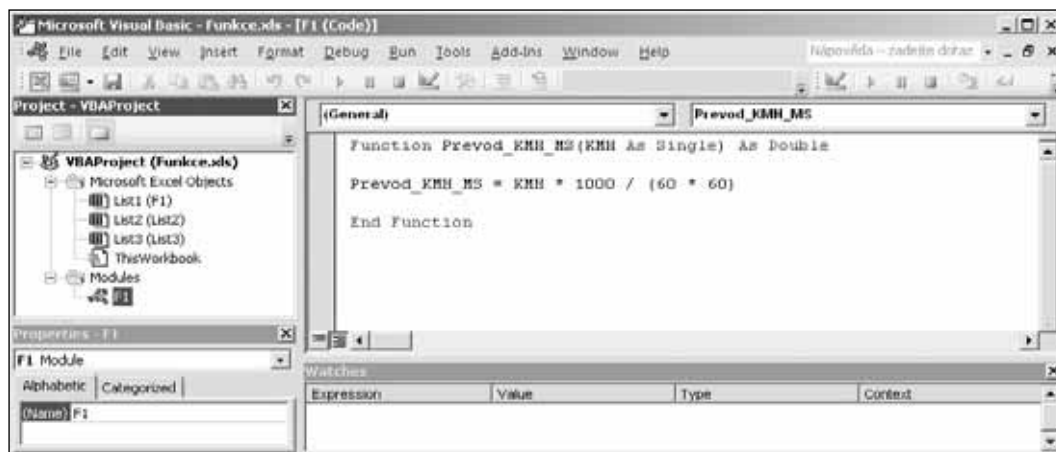
Rejstřík

Funkce

Excel nabízí celou řadu vestavěných tabulkových funkcí, ale každý pokročilejší uživatel zná situaci, kdy byl nucen použít volání několika funkcí zřetězených nebo vložených do sebe v nepřehledném složitém vzorci a řekl si, jak pěkné by bylo, kdyby na takto složitý výpočet existovala jednoduchá vestavěná funkce. Výpočet ostatně nemusí být ani komplikovaný, ale pokud se stále opakuje, může být příjemné nahradit jej jedinou funkcí. Řešení existuje a není ani moc složité – naprogramovat si vlastní tabulkovou funkci. A právě to je obsahem tohoto oddílu.

Všechny zde vytvořené funkce si můžete prohlédnout a jejich použití vyzkoušet v souboru **Funkce.xls**.

(F1) První funkce



Obrázek 1

Co to dělá:

Funkce převede údaj o rychlosti v kilometrech za hodinu na údaj v metrech za sekundu.

Jak na to:

```
Function Prevod_KMH_MS(KMH As Single) As Double
    Prevod_KMH_MS = KMH * 1000 / (60 * 60)
End Function
```



Funkce.xls

Vysvětlení:

- Chcete-li vytvořit vlastní funkci, musíte její kód zapsat do **modulu**. Modul je součástí sešitu Excelu a můžete si jej představit jako speciální list, který ovšem nevidíte přímo z Excelu, ale pouze editorem VBA. Do editoru se z Excelu přepnete například stiskem kombinace kláves **ALT+F11**. V editoru si můžete zobrazit několik různých podoken. Pokud na obrazovce nějaké právě potřebné podokno nevidíte, použijte nabídku **View**. Jedním z podoken je i **Project Explorer** se strukturou jednotlivých otevřených sešitů. Každý list sešitu zde má svůj modul a navíc je zde také modul pro **ThisWorkbook** odpovídající celému sešitu. Tyto moduly jsou zahrnuty ve skupině **Microsoft Excel Objects**. Moduly této skupiny mají svoji zvláštní úlohu a nelze je použít k programování vlastních tabulkových funkcí. Vy si potřebujete založit nový **uživatelský modul**, který bude příslušet skupině **Modules**. Provedete to například příkazem menu **Insert/Modul**. Uživatelský modul se objeví ve struktuře sešitu (zkontrolujte, že správného) a zároveň se otevře textové okno, do kterého můžete začít psát text své funkce.
- Každá funkce musí začínat klíčovým slovem **Function** a končit **End Function**. Funkce se musí nějak jmenovat a **její název nesmí obsahovat nedovolené znaky, jako je mezera, tečka, vykřičník a některé další**. Je lepší nepoužívat ani znaky s diakritikou. Pokud se chcete vyhnout potenciálním problémům, používejte pouze znaky anglické abecedy, číslice a podtržítka (_). Jméno funkce musí začínat písmenem. Daná funkce se jmenuje **Prevod_KMH_MS**.
- Za názvem funkce v závorce následuje parametr, může jich být víc (oddělují se čárkou). Parametry slouží k zadávání vstupních hodnot a znáte je už z vestavěných tabulkových funkcí v Excelu. Každý parametr musíte pojmenovat a můžete i určit, jakého typu bude zadávaná hodnota (celé číslo, desetinné číslo, datum, text, logická hodnota,...). Zde je to parametr **KMH** a je typu **Single**, což znamená desetinné číslo. Specifikace typu se v jazyce VBA uvozuje klíčovým slovem **As**.
Funkce ovšem nemusí mít žádný parametr. Příkladem takové funkce je tabulková funkce **RND**.
- Výsledkem funkce je vypočtená hodnota a rovněž pro ni můžete specifikovat, jakého bude typu. Uvedená funkce pro převod rychlostí vrací hodnotu typu **Double**, což znamená hodnotu, která může mít dvakrát víc cifer než hodnota typu **Single**. Specifikace typu návratové hodnoty funkce je rovněž uvozena klíčovým slovem **As**.
- Uvnitř funkce se vypočte a funkci přiřadí návratová hodnota. Přiřazení se provede znakem **=** (rovnítko).
- Funkci můžete použít v tabulce stejným způsobem jako vestavěné funkce. V průvodci funkcí najdete svou novou funkci v kategorii **Vlastní**.

Funkce

Makra

Buňky

Aplikace

Rejstřík

(F10) Maximální cifra

	A	B	C	D
1				
2		Řetězec	Max číslo	
3		123474321	7	
4		543212345	5	
5		1dgf7gg57	7	
6		hfhgfhfhg	-1	
7				

Obrázek 10

Co to dělá:

Funkce zjistí hodnotu největší číslice v řetězci.

Jak na to:

Funkce postupně probere všechny znaky v řetězci a vybere z nich největší číslici. Pokud se v řetězci nevyskytují žádná číslice, funkce vrátí hodnotu -1.



Funkce.xls

```

Function MaxCifra(S As String) As Integer
    Dim i As Integer
    Dim Nejvetsi As Integer
    Dim Znak As String
    Nejvetsi = -1
    For i = 1 To Len(S)
        Znak = Mid(S, i, 1)
        If Znak Like "#" Then
            If Znak > Nejvetsi Then
                Nejvetsi = Znak
            End If
        End If
    Next i
    MaxCifra = Nejvetsi
End Function
    
```

Vysvětlení:

- V tomto případě je hojně využívána automatická konverze, která v případě potřeby zajistí převod řetězce na číslo.
- Operátor **Like** se používá k porovnání řetězce s určitým vzorem (předlohou, předpisem,...) reprezentujícím skupinu řetězců. Nejspíš umíte použít zástupné znaky „*“ (hvězdička nahrazuje libovolný řetězec) a „?“ (otazník zastupuje jeden libovolný znak). V tomto případě byl použit znak „#“, který symbolizuje libovolnou číslici. K použití operátoru **Like** se podrobněji dostaneme v příkladu F12.

(F11) Převod dvojkového čísla

	A	B	C	D	E
1					
2			Dvojkové	Desítkové	
3			110	6	
4			101010	42	
5			101211	-1	
6					

Obrázek 11

Co to dělá:

Převede číslo zapsané ve dvojkové soustavě na číslo desítkové.

Jak na to:

Číslo se z dvojkové do desítkové soustavy převádí tak, že se jednotlivé číslice dvojkového čísla násobí odpovídající mocninou čísla dvě, podle pozice číslice ve dvojkovém čísle, a násobky se sečtou.

$$(11011)_2 \Rightarrow 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = (27)_{10}$$

Nejprve je třeba zkontrolovat, zda číslo neobsahuje příliš velké číslice, pokud ano, funkce vrátí hodnotu **-1**. Dvojkové číslo je probíráno po znacích od konce, aby bylo možné postupně počítat odpovídající mocniny dvojků.



Funkce.xls

```
Function Prevod2(Dvojkove As String) As Integer
Dim i As Integer
Dim Vysledek As Integer
Dim Mocnina As Integer
If MaxCifra(Dvojkove) >= 2 Then
```

```
    Prevod2 = -1
    Exit Function
End If
Vysledek = 0
Mocnina = 1
For i = Len(Dvojkove) To 1 Step -1
    Vysledek = Vysledek + Mid(Dvojkove, i, 1) * Mocnina
    Mocnina = Mocnina * 2
Next i
Prevod2 = Vysledek
End Function
```

Vysvětlení:

- Ověření maximální číslice se provede pomocí funkce **MaxCifra** z minulého příkladu. Funkce, které si sami naprogramujete, nemusíte používat pouze v tabulkách, ale i v kódu, a to stejným způsobem, jako používáte funkce Visual Basicu.
- Cyklus **For** má nepovinnou část s klíčovým slovem **Step** následovaným číslem. Číslo udává hodnotu, o kterou se změní hodnota řídicí proměnné cyklu v jednom kroku. V tomto případě se po každém jednotlivém průchodu cyklem zmenší hodnota proměnné *i* o jedna. Proměnná *i* má na začátku hodnotu délky zadaného čísla a postupně se zmenšuje až do hodnoty jedna.
- Pokud hodnota nevyhovuje, výpočet funkce je možné předčasně ukončit provedením příkazu **Exit Function**.

K zamyšlení:

Upravte funkci tak, aby převáděla číslo z libovolné soustavy o základu 1–9 do desítkové soustavy.



Poznámka: Pokud byste chtěli vytvořit funkci pouze pro použití v kódu a nechcete, aby se nabízel pro vložení do buňky, musíte použít klíčové slovo **Private**.

```
Private Function MaxCifra(S As String) As Integer
```



Poznámka: Tabulkové funkce jsou určeny k tomu, aby vypočítaly hodnotu. V žádném případě nesmějí měnit vlastnosti a nastavení objektů (buňky, listu, okna,...) v Excelu. Proto tabulková funkce nesmí změnit například formát buňky, do které se zapisuje výsledek, i když to k tomu někdy velmi svádí. Funkce, které mění vlastnosti objektů, musí být deklarovány jako **Private**. Takovéto funkce se pak nezobrazují v seznamu tabulkových funkcí.

3

Buňky ve VBA Excelu

- Součet z největších čísel • Zápis vzorců • Oprava čísla • Číslo z číslic •
- Odkazy na buňky • Očíslovaná adresa • Velká písmena • Skryj označené sloupce •
- Jednoznačnost • Okresy • Kopie řádku • Kopie řádku lépe •

Funkce

Makra

Buňky

Aplikace

Rejstřík

Buňky

Jistě souhlasíte, že práce s buňkou, resp. s oblastí buněk, je v Excelu to nejdůležitější. Proto je celá jedna část vyhrazena příkladům zabývajícím se buňkami a především odkazováním se na ně. Pokud chcete makrem změnit vlastnosti nějaké konkrétní buňky, lze se ve VBA na tuto buňku odkázat mnoha způsoby, což může být (především pro začínající programátory) až na obtíž. Proto je zde při řešení příkladů důsledně používán jeden určitý styl. Nebudete tím nijak ošizeni – použitý styl je dostatečně silný pro naprostou většinu úloh, a to nejen v této knize. Pokud vás přesto zajímají i další možnosti, snadno je naleznete v nápovědě. Měli byste je však začít hledat až v okamžiku, kdy se v problematice dobře zorientujete a nebude již hrozit, že zabloudíte. Možná si pak také vyberete jiný styl, kterým budete psát svoje vlastní programy. Postupem času by si každý programátor měl vytříbit svůj systém (nejen) odkazů na buňky tak, aby zápis jeho programů byl dostatečně přehledný a srozumitelný.

(B1) Součet z největších čísel

C11		fx =NejSoucet(C3:C10;5)					
	A	B	C	D	E	F	G
1							
2			Lukáš	Máša	Tonda	Albert	
3		1.závod	5	-	4	-	
4		2.závod	4	-	5	0	
5		3.závod	5	5	2	4	
6		4.závod	3	5	4	5	
7		5.závod	0	5	0	5	
8		6.závod	-	5	0	-1	
9		7.závod	3	-	0	-2	
10		8.závod	4	-	0	-3	
11		Součet pěti nejlepších	21	20	15	14	
12							

Obrázek 24

Co to dělá:

Funkce spočítá součet pouze určitého počtu největších kladných čísel v oblasti.

Jak na to:

Makro použije pomocné pole, do kterého nakopíruje kladná čísla z buněk v oblasti. Ostatní hodnoty nahradí nulou. Postupně hledá největší hodnoty. Nalezené maximum přičte do výsledného součtu a jeho hodnotu v poli pak nahradí nulou, aby se maximum mohlo stát menší číslo.



Bunky_1-8.xls

```

Function NejSoucet(Oblast As Range, N As Integer) As Double
Dim Hodnoty() As Double      'pole, kam si uložím
                              'hodnoty z buněk oblastí

Dim Bunka As Range
Dim i As Integer
Dim j As Integer
Dim Max As Double           'největší hodnota
Dim MaxJ As Integer        'index největší hodnoty
Dim Soucet As Double

    'načtení hodnot do pole
ReDim Hodnoty(Oblast.Count)
i = 1
For Each Bunka In Oblast
    If IsNumeric(Bunka.Value) Then
        Hodnoty(i) = WorksheetFunction.Max(Bunka.Value, 0)
    Else
        Hodnoty(i) = 0
    End If
    i = i + 1
Next Bunka

'součet N největších
For i = 1 To N

    'hledání maxima
    Max = 0
    For j = 1 To Oblast.Count
        If Max < Hodnoty(j) Then
            Max = Hodnoty(j)
            MaxJ = j
        End If
    Next j

```

Funkce

Makra

Buňky

Aplikace

Rejstřík

```

Soucet = Soucet + Max
Hodnoty(MaxJ) = 0
Next i

NejSoucet = Soucet

End Function

```

Vysvětlení:

- Objekt typu **Range** má vlastnost **Count**. Hodnota **Oblast.Count** prozradí, kolik buněk oblast obsahuje.
- Pole **Hodnoty** je možné deklarovat bez uvedení počtu položek pole. Velikost pole pak určí až zavolání funkce **Redim** během provádění programu.
- Programovací jazyk Visual Basic poskytuje pro výpočty výrazně menší množství vestavěných funkcí, než jste zvyklí používat v tabulkách Excelu. Nicméně většina tabulkových funkcí je zpřístupněna objektem **WorksheetFunction**. Objekt pracuje pouze s anglickými názvy funkcí a neposkytuje funkce, které mají přímo ve Visual Basicu svůj ekvivalent.

(B2) Zápis vzorců

D9		fx =PRŮMĚR(D5:D7)				
	A	B	C	D	E	F
1						
2						
3			Před spuštěním	Po spuštění	Po spuštění	
4			text	Formula	FormulaLocal	
5			=5+4		9	9
6			+7+9	+7+9	+7+9	
7				7		7
8			=průměr(d5:d7)	#NÁZEV?		8
9			=average(d5:d7)		8	#NÁZEV?
10			=nesmysl()	#NÁZEV?		#NÁZEV?
11			sdgfsdagsdfa	sdgfsdagsdfa		sdgfsdagsdfa

Obrázek 25

4

Aplikace

- Import z textového souboru • Export do TXT na zadané pozice •
- Export na pozice jinak • Databáze kontaktů • Vylepšená databáze •
- Tisk vizitky • Velký bratr •

Funkce

Makra

Buňky

Aplikace

Rejstřík

Aplikace

Následující příklady jsou již složitější. Jedná se o ucelené drobné aplikace zaměřené především na ukázkou různých způsobů a možností komunikace mezi vaším programem a jeho uživatelem. Zároveň jsou ukázkou použití programovacích technik probíraných dříve, především práce s buňkou a oblastí buněk.

(A1) Import z textového souboru

	A	B	C	D	E	F	G	H
1								
2		Import						
3		Jméno	Příjmení	Firma	Telefon	E-mail	poslední kontakt	
4		František	FOJTÍK	Farmmed	732 121 972	Frantisek.FOJTIK@Farmmed.cz	12.4.2003	
5		Martin	GOLA	Graas	425 091 986	Martin.GOLA@Graas.cz	3.11.2000	
6		Pavel	ADAMEC	Alego	221 914 210	Pavel.ADAMEC@Alego.cz	7.8.2001	
7		Radovan	CÍSAŘ	Cest	790 401 989	Radovan.CISAR@Cest.cz	15.6.2007	
8		Milan	BRUNCLÍK	Bramo	270 201 964	Milan.BRUNCLIK@Bramo.cz	11.5.2002	
9		Ondřej	EKRT	Ello	771 091 974	Ondrej.EKRT@Ello.cz	1.3.2005	
10		Dana	DVOŘÁKOVÁ	Dafa	708 200 100	Dana.DVORAKOVA@Dafa.cz	16.8.2005	
11								

Obrázek 44

Co to dělá:

Po stisku tlačítka vymaže obsah listu a poté importuje data z textového souboru. Uživatel zadává jméno a umístění importovaného souboru pomocí standardního dialogu.

Jak na to:

Nejprve si můžete pomoci nahráním makra, které otevře textový soubor a rozdělí data do buněk. Přidejte i nahrání příkazu pro nastavení šířky sloupců.

Data z textového souboru se otevrou v novém sešitě a vy je budete muset překopírovat. Buňka, která je aktuální v okamžiku spuštění makra, určí horní levý roh oblasti, kam bude makro importovat data. Pomocný sešit s importovaným textem nesmíte zapomenout zavřít metodou **Close**.



Apl_1-3.xls

```
Sub ImportText()
    Dim JmenoSouboru As String
    Dim CilBunka As Range      'levý horní roh cílové oblasti
    Dim CilWS As Worksheet    'list, kam kopíruji
    Dim ZdrojWB As Workbook   'sešit, odkud kopíruji
```

```

JmenoSouboru = Application.GetOpenFilename(FileFilter:= _
    "Textové soubory (*.txt; *.prn), *.txt; *.prn, " _
    & "Všechny soubory (*.*), *.*", _
    FilterIndex:=1, Title:="Import souboru")
If UCase(JmenoSouboru) = "FALSE" Then
    Exit Sub
End If

Application.ScreenUpdating = False

Set CilWS = ActiveSheet
Set CilBunka = ActiveCell

On Error GoTo Konec
Workbooks.OpenText FileName:=JmenoSouboru, _
    Origin:=xlWindows, StartRow:=1, _
    DataType:=xlFixedWidth, _
    FieldInfo:= Array(Array(0, 2), _
    Array(20, 2), Array(40, 2), _
    Array(60, 9), Array(75, 2), _
    Array(87, 2), Array(137, 4))

On Error GoTo 0

CilWS.UsedRange.Clear
Set ZdrojWB = ActiveWorkbook
ZdrojWB.Worksheets(1).UsedRange.Copy
CilBunka.PasteSpecial xlPasteAll
CilWS.Activate
Selection.Columns.AutoFit

ZdrojWB.Close SaveChanges:=False

Konec:
Application.ScreenUpdating = True

End Sub

```

Funkce

Makra

Buňky

Aplikace

Rejstřík



Makra lze spouštět také kliknutím například na tlačítko. V tomto případě je vhodné umístit formulářové tlačítko na list, na který se budou importovat data.

- Nejprve nakreslete tlačítko.

Postup pro Excel 2003 a starší: menu **Zobrazit** ▶ **Panely nástrojů** ▶ **Formuláře** ▶ **Tlačítko**

Postup pro Excel 2007: karta **Vývojář** ▶ skupina **Ovládací prvky** ▶ **Vložit** ▶ **Ovládací prvky formuláře** ▶ **Tlačítko**

- Upravte nápis na tlačítku. V kontextové nabídce (klepněte pravým tlačítkem myši na tlačítko) vyberte příkaz **Přiřadit makro...** a přiřadte správný název makra.

Vysvětlení:

- Metoda **GetOpenFilename** spustí standardní dialog pro otvírání souborů, ale soubor neotevře, jen předá informaci o souboru, který si uživatel vybral. Její parametr **FileFilter** je textový řetězec tvořený dvojicemi popisujícími určitý filtr. Jednotlivé dvojice jsou odděleny čárkou. Dvě části jedné dvojice popisující určitý filtr jsou také oddělené čárkou. V první části se zadává text, který se zobrazí v editačním okně **Soubory typu**, a v druhé filtr pro nabízené soubory.

Je-li potřeba v jednom filtru použít více přípon zároveň, oddělují se středníkem.

- Jednou z metod, které do kolekce **Workbooks** přidají nový sešit, je metoda **OpenText**. Načte textový soubor tak, že řádky textu rozdělí do řádků tabulky, případně dále do jednotlivých buněk.
- Parametr **FieldInfo** metody **OpenText** určí, jak rozdělit jeden záznam (řádek textu) na položky (do buněk). Každá položka je popsána dvojicí čísel. První udává pozici začátku položky (číslováno od nuly), druhá určuje typ importovaných dat. Čísla pro jednotlivé typy najdete v nápovědě nebo ještě lépe – využijte nahrané makro.
- Ukončí-li uživatel dialog stisknutím tlačítka **Storno**, metoda vrací hodnotu **False**. Pokud výsledek metody ukládáte do proměnné typu **String**, dojde k automatické konverzi a hodnota **False** se převede na řetězec "False".
- Jelikož otvíráním a zavíráním sešitů se mění, které sešity jsou aktivní a které pasivní, měli byste pro jednotlivé sešity používat proměnné daného typu a nespolehat se jen na to, že víte, který sešit je právně aktivní.
- Akce, jako je například otvírání nového sešitu nebo kopírování metodou **Copy-Paste**, jsou viditelné akce, které způsobují překreslení okna aplikace. Pokud se takovýchto akcí v průběhu makra provádí víc rychle za sebou, způsobí to nepříjemné blikání obrazovky. Může být proto užitečné použít metodu **ScreenUpdating**, která zabrání překreslování obrazovky.