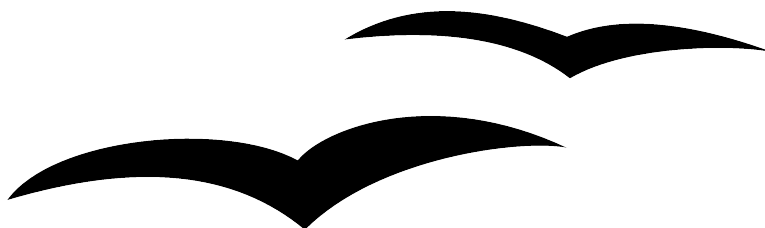


Makrá v OpenOffice.org

Makromoduly a drobnosti



Obsah

Makromodul (nielen) pre vzorce – Dmaths.....	3
Počítanie znakov.....	6
Drobnosti, ktoré potešia.....	11
Čarujeme s koncami odstavcov.....	14
Rôzne zámeny.....	19
Vlastný dialóg pre zámeny.....	21
Rozšírenie pre typografické úpravy textu – TypoJTB.....	24
Typografické úpravy.....	25
Kontrola úvodzoviek a zátvoriek.....	26
Hľadanie a nahradzovanie typografických znakov.....	26
Nastavenie šírky medzier a znakov.....	26
Vkladanie špeciálnych znakov.....	27
Informácie o kódach znakov.....	27
Rozdiely oproti štýlom.....	27
Štatistika použitých fontov.....	28
Vloženie (zrušenie) zalomenia stránky.....	28
Nedeliť slovo.....	28
Zdroje informácií.....	28
Analýza premenných.....	29
Dokumentácia.....	31
Internet.....	32
Dokumentácia SUN StarOffice:.....	32

V šiestej časti seriálu o programovaní makier v OpenOffice.org sa budeme venovať rôznym makrá, ktoré spríjemnia našu každodennú prácu. Nie všetky budeme programovať, ale predstavíme si aj niekoľko zaujímavých makier či celých makromodulov od iných autorov.

V doterajších častiach o programovaní makier pre OpenOffice.org sme sa venovali najmä formátovaniu dokumentov s tým, že sme si na týchto príkladoch postupne predstavili jazyk StarOffice Basic, typy premenných, prácu s dialógmi, možnosti nastavovania parametrov s ich pamätaním v rôznych profiloch a pod. Teraz zúročíme tieto poznatky a uvedieme si rôzne (nielen) drobné makrá, aby sme videli, že nie všetko sa musí venovať formátovaniu.

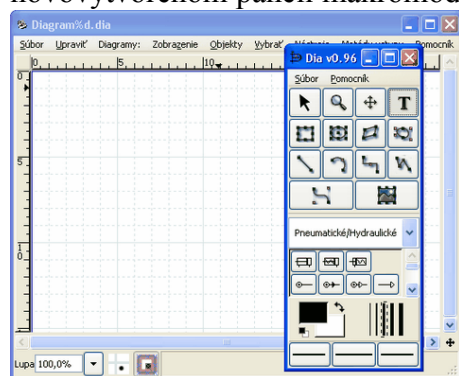
Makromodul (nielen) pre vzorce – Dmaths

Aby sme začali trochu netradične, dnes nebudeme programovať, ale predstavíme si francúzsky makromodul Dmaths. Tento makromodul je určený najmä pre tých, ktorí často pracujú so vzorcami. Určite mnohí z nich už neraz „nadávali“, že pre vloženie vzorca musia pracne vchádzať do modulu Math a prečo OpenOffice.org neumožňuje ich zadanie jednoduchšie, najmä ak dobre ovládajú skriptovací jazyk pre ich zadávanie.

A práve na riešenie takýchto problémov je ako stvorený predstavovaný makromodul Dmaths. Pretože jeho inštalácia je trochu špecifická, popíšeme si ju. Na tomto mieste sa ospravedlňujem majiteľom počítačov Apple, ale pretože ich nevlastním, nepopíšem inštaláciu pre ich operačný systém.

Po rozbalení distribučného ZIP balíka otvoríme súbor install.odt (pravdaže, musíme povoliť spúšťanie makier z tohto súboru). Následne klikneme na jazyk, v ktorom chceme, aby sa makromodul nainštaloval. Tu nás poteší, že medzi nimi nájdeme aj slovenčinu. Na záver súbor zavrieme. V adresári, kde sme súbor rozbalili sa nám medzitým vytvoril súbor „instalmono.bat“ (pre operačný systém Windows) alebo súbor „instalmono.sh“ (pre operačný systém Linux).

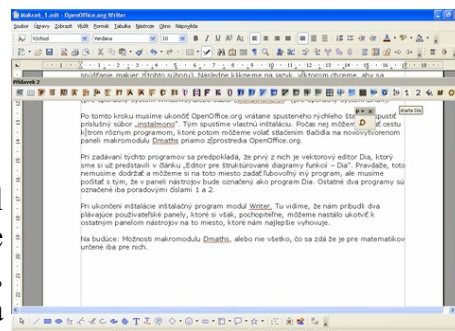
Po tomto kroku musíme ukončiť OpenOffice.org vrátane spusteného rýchleho štartu a spustiť príslušný súbor „instalmono“. Tým spustíme vlastnú inštaláciu. Počas nej môžeme zadať cestu k trom rôznym programom, ktoré potom môžeme volať stlačením tlačidla na novovytvorenom paneli makromodulu Dmaths priamo z prostredia OpenOffice.org.



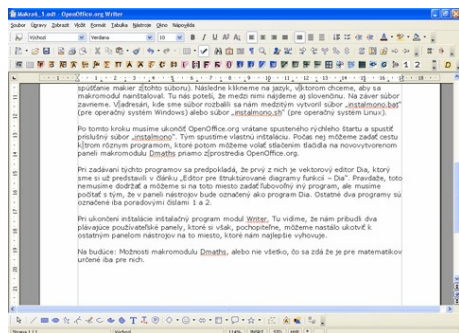
Obrázok 1: Vektorový editor DIA

Pri ukončení inštalácie inštalčný program modul Writer. Tu vidíme, že nám pribudli dva plávajúce používateľské panely, ktoré si však, pochopiteľne, môžeme nastálo ukotviť k ostatným panelom nástrojov na to miesto, ktoré nám najlepšie vyhovuje.

Pri zadávaní týchto programov sa predpokladá, že prvý z nich je vektorový editor Dia, ktorý sme si už predstavili v článku „Editor pre štruktúrované diagramy – Dia“. Pravdaže, toto nemusíme dodržať a môžeme si na toto miesto zadať ľubovoľný iný program, ale musíme počítať s tým, že v paneli nástrojov bude označený ako program Dia. Ostatné dva programy sú označené iba poradovými číslami 1 a 2.



Obrázok 2: Používateľské panely makromodulu Dmaths



Obrázok 3: Ukotvený pracovný panel Dmaths

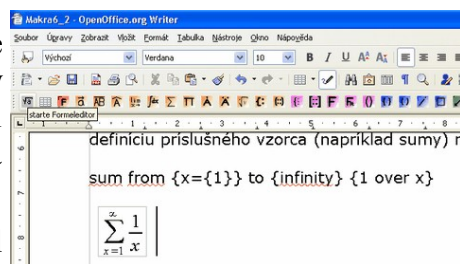
Teraz si ukážeme možnosti, ktoré nám tento makromodul ponúka. Pri tomto popise sa budeme odvolávať na tlačidlá, ktoré sa nachádzajú v jeho pracovnom paneli. Pravdaže, ku mnohým z nich existujú aj klávesové skratky, ich používanie necháme už na láskavého čitateľa.

Pre tých, ktorí dobre ovládajú skriptovací jazyk modulu Math bude najzaujímavejšia prvá možnosť – vloženie vzorca priamo z prostredia textového procesora Writer. Stačí, ak napíšeme definíciu príslušného vzorca (napríklad sumy) na to miesto, kde ho chceme vložiť:

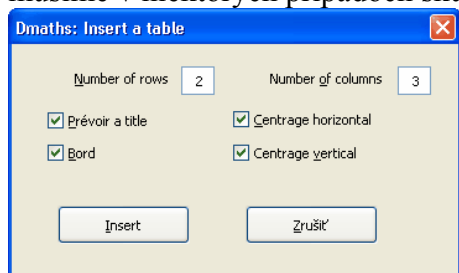
sum from {x={1}} to {infinity} {1 over x}

Definíciu potom označíme a stlačíme tlačidlo „starte Formeleditor“. Následne sa nám do textu automaticky vloží vzorec tak isto, ako keby sme ho zadali cez modul Math. Pokiaľ toto tlačidlo stlačíme bez označenia definície vzorca, otvorí sa nám modul Math.

Ďalšia možnosť, ktorú nám ponúka makromodul Dmaths je vkladanie tabuliek. Žiaľ, už v tomto dialógu vidíme, že preklad do slovenčiny nie je urobený na všetkých miestach dokonale, ale nájdeme tu konkrétne anglicko-francúzsky text. Preto musíme v niektorých prípadoch skúšať, čo vlastne znamenajú príslušné kolonky.



Obrázok 4: Priame vloženie vzorca v textovom procesore



Obrázok 5: Vloženie tabuľky

Teraz sa budeme venovať oranžovým tlačidlám. Tieto funkcie sú určené pre zrýchlené zadávanie často používaných funkcií, za ktoré autori tohto modulu považujú vektory, limity, integrály, sumy, odmocniny, matice a pod. V týchto prípadoch stačí, ak zadáme parametre príslušných funkcií oddelené bodkočiarkou. Údaje nemusíme ani označovať, ale stačí, ak tesne za nimi vyberieme príslušnú funkciu. Takto vyzerá napríklad zjednodušená definícia matice:

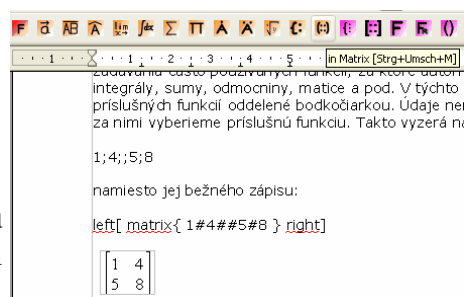
1; 4;; 5; 8

namiesto jej bežného zápisu:

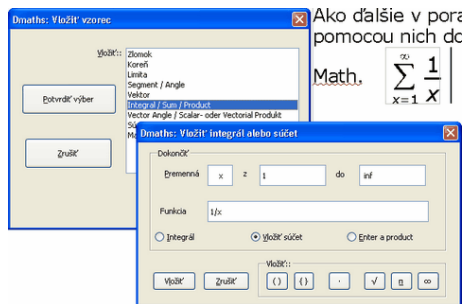
left[matrix{1#4##5#8} right]

Ako ďalšie v poradí nájdeme fialové tlačidlá. Tieto sú určené najmä pre začiatočníkov, pretože pomocou nich dokážeme zadávať vzorce bez akejkoľvek znalosti (resp. iba so základnou znalosťou používania zloženej zátvorky pre zoskupovanie) skriptovacieho jazyka modulu Math.

Tmavomodré tlačidlá sú určené pre zmenu vybraného textu (nie vzorcov). Po označení textu ho môžeme vložiť do obyčajných alebo zložených zátvoriek, zmeniť ho na kurzívu alebo uložiť do rámca.

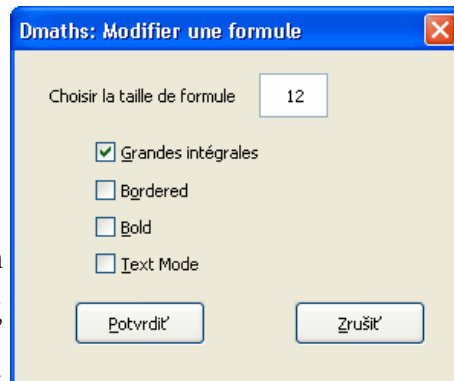


Obrázok 6: Jednoduché vloženie matice

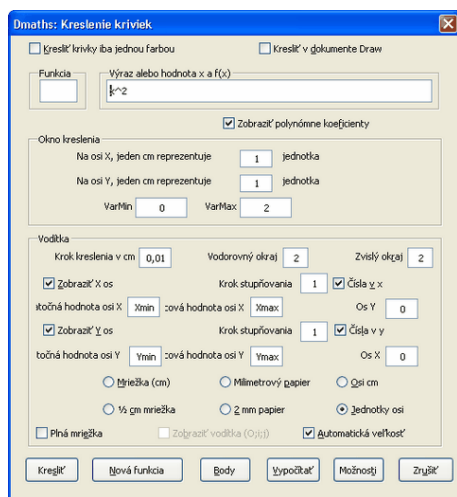


Obrázok 7: Vloženie vzorca bez potreby jeho definície

Svetlomodré tlačidlá ukrývajú možnosti, ktoré nám bežne OpenOffice.org (a nielen on) neposkytuje. Pomocou nich totiž môžeme kresliť grafy funkcií, vkladať milimetrový papier, prevádzať štatistické výpočty atď. Upozorňujem, že graf sa vždy nakreslí do ľavého horného rohu strany a preto ho musíme následne presunúť na to miesto, kde ho potrebujeme umiestniť.

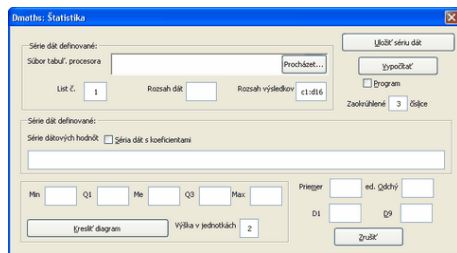


Obrázok 8: Formátovanie vzorca



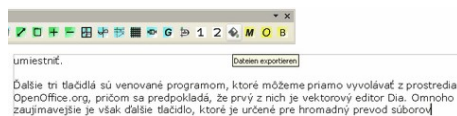
Obrázok 9: Definovanie grafu funkcie

textové do textových a pod. Modul podporuje prevod do týchto formátov:



Obrázok 11: Štatistické výpočty

Nakoniec sa v paneli nástrojov nachádzajú tlačidlá pre nastavenie pracovných parametrov a zálohovanie používateľských adresárov OpenOffice.org. Týmto funkciám sa už nebudeme špeciálne venovať.

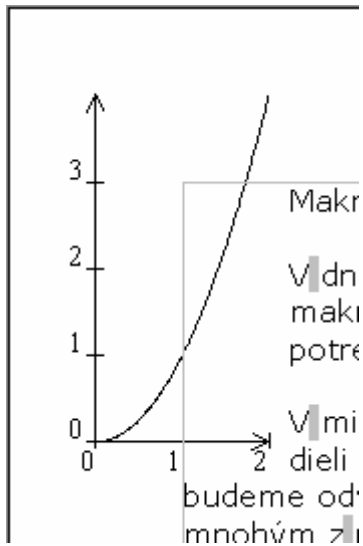


Obrázok 12: Tlačidlo pre hromadnú konverziu súborov

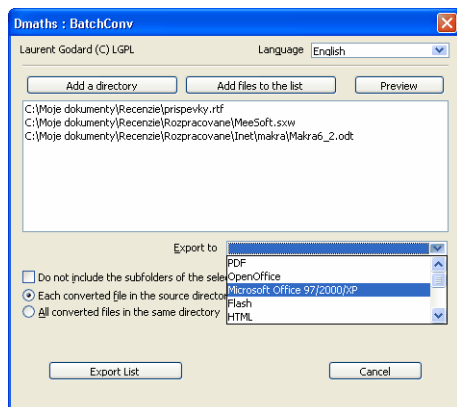
Zelené tlačidlá sú určené pre formátovanie už vloženého vzorca. Môžeme zmeniť jeho veľkosť, rez, resp. ho orámovať.

Ďalšie tri tlačidlá sú venované programom, ktoré môžeme priamo vyvolávať z prostredia OpenOffice.org, pričom sa predpokladá, že prvý z nich je vektorový editor Dia.

Omného zaujímavejšie je však ďalšie tlačidlo, ktoré je určené pre hromadný prevod súborov. Program dokáže naraz konvertovať súbory z rôznych vstupných formátov do jednotného výstupného formátu. Pravdaže, vstupné formáty musia zodpovedať súborom rovnakého druhu, t.j. PDF, OpenOffice.org, Microsoft Office 97/2000/XP, Flash, HTML, Fichier Texte, RTF, Microsoft Word 95, Microsoft Winword 6.0, Microsoft Excel 95, Microsoft Excel 5.0, JPG, PNG, TIFF, SVG, EPS.



Obrázok 10: Vložený graf



Obrázok 13: Hromadný prevod súborov

Na záver už iba uvedme, že makromodul Dmaths je šírený pod licenciou GNU General Public Licence. Úplné znenie licencie si môžeme prečítať tak, že si otvoríme nový súbor, napíšeme text „copie“ a stlačíme tlačidlo F3.

Internet: www.dmaths.com

Počítanie znakov

Ani vám nepostačujú štandardné možnosti, ktoré ponúka OpenOffice.org pri počítaní znakov? Dnes si ukážeme makro, pomocou ktorého dokážeme vypočítať o.i. napr. aj počet normostrán.

Tí, ktorí prispievajú do rôznych časopisov (papierových či internetových), alebo píšú romány sa ešte stále stretávajú s tým, že sú odmeňovaní podľa počtu normostrán, napísaných znakov ale bez medzier a pod. Podobne, pri rôznych súťažiach je požadované, aby napr. román mal minimálne toľko a toľko normostrán. Žiaľ, OpenOffice.org nám ponúka v bežnej štatistike iba základné údaje o počte znakov a vyššie uvedené možnosti nemá.

Okrem toho niekedy (napr. pri rozdeľovaní dlhého textu na viacero kratších pokračovaní) potrebujeme spočítať uvedené veci iba pre vybranú časť textu, resp. naraz aj pre celý text aj pre označený text. Preto si dnes uvedieme makro, ktoré nám dokáže spočítať počet slov, počet všetkých znakov, počet všetkých znakov bez medzier, počet normostrán zo všetkých znakov a počet autorských hárkov z normostrán a to naraz ako pre celý dokument, tak aj pre vybranú časť textu (pokiaľ nejakú označíme).

Hoci by sme pre vlastné počítanie niektorých znakov (v našom prípade to budú medzery) mohli teoreticky použiť aj náhradu znaku za ten istý znak, je to nepraktické (zbytočne by sme vykonávali náhradu). Preto si zadefinujeme vlastnú funkciu pre počítanie znakov, kde nebudeme používať štandardnú metódu pre nahrádzanie, ale pre vyhľadávanie.

Vzhľadom na to, že makrá sú, ako obvykle, čiastočne okomentované, nebudeme ich už viac rozoberať a uvedieme si priamo ich zdrojové texty. Ešte podotýkam, že v zozname makier ich mám uložené v osobitnom module „Pocitanie_znakov“ a pre vlastné spočítanie sa volá procedúra „Pocitanie_znakov“. Tím, ktorí budú túto procedúru používať často (u mňa je to každodenná náležitosť a vlastne vôbec nepoužívam počítanie znakov, ktoré má OpenOffice.org štandardne k dispozícii), odporúčam jej umiestnenie na niektorom paneli nástrojov.

```
public medzery, medzery_regularne as string ' Oddel'ovacie znaky medzi slovami
public oddelovac_slov as string ' Znak, ktorý ukončuje slovo
public nepocitane_znaky as string ' Znaky, ktoré sa nikdy nepočítajú
public normo_strana as single ' Počet znakov na normostranu
```

```

public autorsky_harok as single ' Počet normostrán na jeden AH

dim Dokument, Hladaj, vyber as object

REM Inicializácia premenných
sub init_znaky

medzery=chr(&H20)+chr(&HA0)+chr(&H09) ' Space, Hard space, Tab
' Definícia medzier pre vyhľadávanie pomocou regulárnych výrazov
medzery_regularne="(" + chr(&H20) + "|" + chr(&HA0) + "|" + chr(&H09) + ")"

oddelovac_slov=medzery+chr(&H0A)+chr(&H0D) ' Medzery, CR, LF
' Napr. Frídek-Místek sú dve slova, preto tu je aj pomlčka ' alebo lomítko a bodka napr. pr
e http odkazy
oddelovac_slov=oddelovac_slov+"-"/" "+"."
' alebo lomítko a bodka napr. pre http odkazy

nepocitane_znaky=chr(&H0A)+chr(&H0D) ' CR, LF

' Ak sa zmení norma, alebo niekto potrebuje tieto údaje počítať inak, môže si zmeniť tieto č
ísla
normo_strana=1800 ' Počet znakov na normostranu je 1800
autorsky_harok=16 ' Počet normostrán na autorský hárok je 16

Dokument=ThisComponent
Hladaj=Dokument.createSearchDescriptor()
vyber=dokument.getCurrentSelection() ' aktuálny vyber
end sub

REM Funkcia spočíta počet výskytov reťazca Co
function Spocitaj_hladanim(Co as string, Regularne, Cele_slova, Velke_pismena as Boolea
n) as long
dim nasiel as object
dim kolko as long
Hladaj.searchString=Co
Hladaj.SearchRegularExpression=Regularne
Hladaj.SearchWords=Cele_slova

```

```

Hladaj.SearchCaseSensitive=Velke_pismena
kolko=0
' Vyhľadanie prvého výskytu hľadaného reťazca
nasiel = Dokument.findFirst(Hladaj)
' Pokiaľ sa reťazec našiel, tak
Do While NOT isNull(nasiel)
    if len(nasiel.String)>0 then
        kolko=kolko+1
    end if
' Vyhľadanie ďalšieho výskytu hľadaného reťazca
nasiel = Dokument.findNext(nasiel.End, Hladaj)
Loop
Spocitaj_hladanim=kolko
end function

```

rem Procedúra pre spočítanie znakov a slov v celom dokumente a vo výberoch

Sub Pocitanie_znakov

init_znaky

' Štatistické údaje o celom dokumente

dim vsetky_znaky, vsetky_slova, vsetky_znaky_bez as long

dim vsetky_normo, vsetky_normo_bez, vsetky_AH, vsetky_AH_bez as single

' Štatistické údaje o výbere

dim vyber_pocet, vyber_znaky, vyber_znaky_bez, vyber_slova as long

dim vyber_normo, vyber_normo_bez, vyber_AH, vyber_AH_bez as single

dim riadok as string

dim vyber_ktory, dlzka_riadku, i as long

dim slovo as boolean

dim znak, CR, real_format as string

CR=chr(&H0D)

' Formát pre zobrazenie počtu nájdených a vypočítaných údajov (na dve desatinné miesta)

real_format= "#####0.00"

```

' Údaje pre celý dokument
' Celkový počet slov
vsetky_slova=dokument.WordCount

' Celkový počet znakov
vsetky_znaky=dokument.CharacterCount
' Celkový počet znakov bez medzier
vsetky_znaky_bez=vsetky_znaky-Spocitaj_Hladanim(medzery_regularne, true, false,
false)

' Výpočet normostrán, AH a to s medzerami aj bez nich pre celý dokument
vsetky_normo=vsetky_znaky/normo_strana ' počet normostrán
vsetky_normo_bez=vsetky_znaky_bez/normo_strana ' počet normostrán bez medzier
vsetky_AH=vsetky_normo/autorsky_harok ' počet autorských hárkov
vsetky_AH_bez=vsetky_normo_bez/autorsky_harok ' počet AH bez medzier

' Údaje pre vybrané časti textu

' Počet výberov (naraz môžeme označiť viacero nezávislých častí textu)
vyber_pocet=vyber.getCount()

vyber_slova=0 ' počet slov
vyber_znaky=0 ' počet znakov s medzerami
vyber_znaky_bez=0 ' počet znakov bez medzier

' Pre všetky výbery
for vyber_ktory=0 to vyber_pocet-1
' Text príslušného výberu
riadok = vyber.getByIndex(vyber_ktory).getString()
dlzka_riadku=len(riadok)

' Tvorba slova pre počítanie ich počtu
slovo=false ' slovo neobsahuje ešte žiadny znak

' Pre všetky znaky z príslušného výberu
for i=1 to dlzka_riadku

```

```

znak=mid(riadok, i,1) ' spracovávame i-ty znak z textu
if instr(oddelovac_slov, znak)=0 then
    ' znak nie je oddeľovač slov, t.j. je časťou slova
    slovo=true ' slovo určite obsahuje nejaký znak
Elseif slovo then
    ' znak je oddeľovač slova a slovo obsahovalo nejaký znak – zvyšuje sa počet slov
    vyber_slova=vyber_slova+1
    ' Slovo bolo ukončené – znovu začíname tvoriť ďalšie
    slovo=false
endif
if instr(nepocitane_znaky, znak)=0 then
    ' znak sa nenachádza v zozname nepočítaných znakov, t.j. počítame ho
    vyber_znaky=vyber_znaky+1
    if instr(medzery, znak)=0 then
        ' znak nie je medzera, počítame ho teda aj medzi všetky ostatné znaky
        vyber_znaky_bez=vyber_znaky_bez+1
    endif
endif
endif
next i
next vyber_ktory

' ak sme na konci posledného slova nemali oddeľovač,
' tak toto slovo ešte nemáme započítané v celkovom počte slov
if slovo then
    vyber_slova=vyber_slova+1
endif

' Výpočet normostrán, AH a to s medzerami aj bez nich pre vybranú časť dokumentu
vyber_normo=vyber_znaky/normo_strana
vyber_normo_bez=vyber_znaky_bez/normo_strana
vyber_AH=vyber_normo/autorsky_harok
vyber_AH_bez=vyber_normo_bez/autorsky_harok

' Využijeme premennú "znak" na vytvorenie výstupného formulára

znak="Počet všetkých slov: "+vsetky_slova+CR+CR

```

```
znak=znak+"Počet všetkých znakov: "+vsetky_znaky+CR
znak=znak+"Počet všetkých normostrán: "+format(vsetky_normo, real_format)+CR
znak=znak+"Počet všetkých autorských hárkov: "+format(vsetky_AH, real_format)+CR
+CR
```

```
znak=znak+"Počet všetkých znakov bez medzier: "+vsetky_znaky_bez+CR
znak=znak+"Počet všetkých normostrán bez medzier: "+format(vsetky_normo_bez,
real_format)+CR
```

```
znak=znak+"Počet všetkých autorských hárkov bez medzier: "+format(vsetky_AH_bez,
real_format)+CR+CR+CR
```

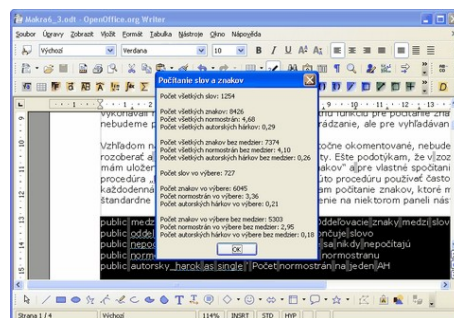
```
znak=znak+"Počet slov vo výbere: "+vyber_slova+CR+CR
```

```
znak=znak+"Počet znakov vo výbere: "+vyber_znaky+CR
znak=znak+"Počet normostrán vo výbere: "+format(vyber_normo, real_format)+CR
znak=znak+"Počet autorských hárkov vo výbere: "+format(vyber_AH, real_format)+CR
+CR
```

```
znak=znak+"Počet znakov vo výbere bez medzier: "+vyber_znaky_bez+CR
znak=znak+"Počet normostrán vo výbere bez medzier: "+format(vyber_normo_bez,
real_format)+CR
znak=znak+"Počet autorských hárkov vo výbere bez medzier: "+format(vyber_AH_bez,
real_format)
```

```
msgbox(znak, 0, "Počítanie slov a znakov")
```

```
End Sub
```



Obrázok 14: Počítanie znakov

Drobnosti, ktoré potešia

V dnešnom pokračovaní seriálu o programovaní makrií v OpenOffice.org si uvedieme rôzne naozaj drobné makrá, ktoré nám pomôžu pri formátovaní dokumentu.

V doterajších dieloch sme si o.i. uviedli makrá, pomocou ktorých sme vkladali nezalomiteľné medzery za jednoznakové predložky, akademické tituly, do vnútra napr. telefónnych čísiel a pod. V bežnej praxi sa však mnohokrát stretávame aj s ďalšími potrebami v tejto oblasti. Napríklad, pokiaľ chceme použiť úzku sadzbu, potom nám nezalomiteľné medzery naopak prekážajú a potrebujeme ich nahradiť zalomiteľnými.

S týmto istým (alebo podobným) problémom sa môžeme stretnúť napr. aj pri potrebe zaslania rukopisu, keď si vydavateľ žiada, aby sme v texte nezalomiteľné medzery

nepoužívali, pretože ich prípadné vloženie (t.j. zalomenie textu) si robí sám pomocou iných nástrojov.

Podobná problematika vzniká napr. aj vtedy, keď potrebujeme spracovať text, ktorý bol pripravený napr. v textovom procesore MS Word. Tento program vkladá nezalomiteľné medzery automaticky, čo nám však nemusí vyhovovať, pretože mi ich chceme mať napr. iba tam, kde sme si to nastavili v našom pracovnom profile. To je aj prípad, keď musíme upravovať text, ktorý pripravil niekto iný.

Osobne sa stretávam aj s ďalším problémom – potrebujem nahradiť úplne všetky medzery sa nezalomiteľné. Jedná sa napr. aj o príspevky do tohto seriálu o programovaní makier – konkrétne o zdrojové texty. Problém je vlastne v tom, že potrebujem dosiahnuť, aby zdrojové texty boli odsadené podľa štruktúry príkazov. Pokiaľ tam nechám obyčajné medzery (ktoré sa v zdrojovom texte automaticky nachádzajú po jeho skopírovaní z modulu Basic), tieto mi redakčný systém na začiatku riadku „odreže“. Preto ich musím nahradiť medzerami nezalomiteľnými.

Pravdaže, zároveň chcem, aby jeden programový riadok bol aj zobrazený ako jeden riadok a preto musím nahradiť zalomiteľné medzery za nezalomiteľné aj vo vnútri vlastných príkazov. Samozrejme, tí používatelia, ktorí si prípadne skopírujú takto uvedený zdrojový text musia počítať s tým, že im modul Basic následne vyhlási chybu (t.j. nebudú priamo fungovať) a musia si ho opraviť. Je to však seriál o programovaní a nie o kopírovaní hotových zdrojových textov, takže si každý môže dať tú námahu, aby si ich aspoň prepísal...

Ako posledná problematika, ktorej sa budeme dnes venovať je problematika rôznych nezalomiteľných medzier. V predchádzajúcich dieloch sme si uviedli dva typy – bežnú nezalomiteľnú medzeru (znak „Non Break Space“) a tzv. pružnú nezalomiteľnú medzeru (znak „Word Joiner“). Pri bežnej práci sa nám môžu hodiť makrá, ktoré dokážu automaticky zamedziť vzájomne tieto znaky, t.j. podľa potreby môžeme previesť jeden typ na druhý a opačne. Pravdaže, keď sme už spomínali zámenu všetkých medzier za nezalomiteľné medzery, aj v tých makrách uvedieme obidve tieto možnosti.

Tak, ako v predchádzajúcich dieloch, vlastné makrá nebudeme osobitne popisovať, pretože sú (podľa môjho názoru) dostatočne okomentované. Iba pripomeňme, že v nich používame funkcie, ktoré sme už uviedli pri formátovaní dokumentu (konkrétne funkcie „Init“ a „Vymen“), pri ktorých budeme predpokladať, že sú uložené v module „Formatovanie_funkcie“ – t.j. voláme ich Formatovanie_funkcie.init a Formatovanie_funkcie.Vymen. Dnes uvedené funkcie si uložíme do modulu „Rozne“.

```
REM Makro pre nahradenie nezalomiteľných medzier zalomiteľnými
REM Musíme nahradiť ako znak NBSP – hexadecimálny kód A0,
REM tak aj znak WJ – hexadecimálny kód 2060
function f_Zalomitelna_Medzera as long
Formatovanie_funkcie.init
f_Zalomitelna_Medzera=Formatovanie_funkcie.Vymen(chr$(&HA0)," ", false, false, false)
)+_
Formatovanie_funkcie.Vymen(chr$(&H2060),"", false, false, false)
end function
```

```

REM Makro pre nahradenie zalomitel'nych medzier nezalomitel'nymi medzerami NBSP
function f_Nezalomitelna_Medzera_NBSP as long
    Formatovanie_funkcie.init
    f_Nezalomitelna_Medzera_NBSP=Formatovanie_funkcie.Vymen(" ", chr$(&HA0), false,
false, false)
end function

```

```

REM Makro pre nahradenie zalomitel'nych medzier nezalomitel'nymi medzerami WJ
function f_Nezalomitelna_Medzera_WJ as long
    Formatovanie_funkcie.init
    f_Nezalomitelna_Medzera_WJ=Formatovanie_funkcie.Vymen(" ", chr$(&H2060)+" ", fal
se, false, false)
end function

```

```

REM Makro pre zámenu NBSP za WJ
function f_NBSP_za_WJ as long
    Formatovanie_funkcie.init
    f_NBSP_za_WJ=Formatovanie_funkcie.Vymen(chr$(&HA0), chr$(&H2060)+" ", false, f
alse, false)
end function

```

```

REM Makro pre zámenu WJ za NBSP
function f_WJ_za_NBSP as long
    Formatovanie_funkcie.init
    f_WJ_za_NBSP=Formatovanie_funkcie.Vymen(chr$(&H2060)+" ", chr$(&HA0), false, f
alse, false)
end function

```

REM Funkcie môžeme volať aj priamo, ale vtedy nám nevypíše počet zámen.

REM Ak chceme vidieť aj tento počet, musíme si na to naprogramovať procedúry.

```

sub Zalomitelna_Medzera
    ' Nahradenie nezalomitel'nych medzier zalomitel'nymi
    msgbox("Zmenených "+f_Zalomitelna_Medzera+" nezalomitel'nych medzier.", 0,"Zalomit
el'né medzery")
end sub

```

```

sub Nezalomitelna_Medzera_NBSP
' Nahradenie zalomitel'nych medzier nezalomitel'nymi medzerami NBSP
msgbox("Zmenených "+f_Nezalomitelna_Medzera_NBSP+" zalomitel'nych medzier.", 0,"Nezalomitel'né medzery NBSP")
end sub

sub Nezalomitelna_Medzera_WJ
' Nahradenie zalomitel'nych medzier nezalomitel'nymi medzerami WJ
msgbox("Zmenených "+f_Nezalomitelna_Medzera_WJ+" zalomitel'nych medzier.", 0,"Nezalomitel'né medzery WJ")
end sub

sub NBSP_za_WJ
' Záměna NBSP za WJ
msgbox("Zmenených "+f_NBSP_za_WJ+" NBSP za WJ.", 0,"Záměna NBSP za WJ")
end sub

sub WJ_za_NBSP
' Záměna WJ za NBSP
msgbox("Zmenených "+f_WJ_za_NBSP+" WJ za NBSP.", 0,"Záměna WJ za NBSP")
end sub

```

Čarujeme s koncami odstavcov

V ďalšom pokračovaní seriálu o programovaní makier v OpenOffice.org si naprogramujeme dve podobné makrá, pomocou ktorých budeme upravovať konce odstavcov.

Pri písaní textu bežne používame viacero odstavcov, t.j. odstavec ukončujeme klávesou „Enter“. V niektorých prípadoch to však nie je žiadúce, pretože potrebujeme, aby sme v dokumente vlastne nemali klasické odstavce, ale aby sa celý dokument správal v skutočnosti ako jediný odstavec (t.j. ako keby sme namiesto klávesy „Enter“ stláčali kombináciu „Shift + Enter“, ktorá predstavuje zalomenie riadku).

Samozrejme, môžete sa opýtať, v akých prípadoch to niekto vôbec môže potrebovať. Odpoveď je veľmi jednoduchá – napr. aj ja pri písaní príspevkov na Inet.sk. Je to tak preto, lebo redakčný systém, pomocou ktorého sa príspevky vkladajú, pri klasickom ukončení odstavca vkladá iné odsadenie, ako pri zalomení riadku. Pri obyčajnom texte by to ešte nebolo na prekážku, ale pri vkladaní zdrojových textov programov by to dopadlo priam katastrofálne, pretože by sa tento nedal prakticky nastaviť ako osobitný štýl (ten totiž aplikuje redakčný systém na označený text, pričom však rešpektuje aj zadané odstavce, takže každý riadok by bol uvedený v osobitnom rámičku).

V makrách pre formátovanie dokumentu sme si o. i. uviedli makro pre odstraňovanie prázdnych odstavcov. A tu sa znovu dostávame k opačnému problému – niekedy potrebujeme

mať prázdne odstavce medzi ostatnými odstavcami. Pre príklad takejto potreby nemusíme chodiť ďaleko. Znovu sa vrátíme k problematike uverejňovania textov na internete, kde ich jednoducho inak nedokážeme opticky oddeliť (t.j. príslušný redakčný systém musí medzi ne vložiť dvakrát ukončenie riadku –

). Pokiaľ však máme text pripravený tak, že odstavce sme si odsadzovali iba nastavením vzdialenosti medzi nimi, je to pre takéto (ale nielen pre takéto) spracovanie naozaj nepostačujúce.

Dost' však bolo teórie, uveďme si teraz vlastné makrá pre upravovanie odstavcov. Na tomto mieste musíme upozorniť, že vzhľadom na možnosti, ktoré nám ponúka aplikačné programovacie rozhranie OpenOffice.org sa tieto zmeny dajú previesť iba v označenom texte a preto, ak ich chceme aplikovať na celý dokument, musíme si ho najprv celý označiť (napr. klávesovou skratkou „Ctrl + A“).

Ešte upozorníme, že makrá pre úpravu odstavcov menia aj formát celého označeného textu na formát posledného odstavca. Táto vlastnosť je automatická a programovo by sa riešila naozaj veľmi obtiažne (pokiaľ by sa vôbec dala vyriešiť). Vzhľadom na to sa v makrách vôbec nerieši a preto s ňou musíme počítať ešte pred tým, ako makro spustíme. Z uvedeného dôvodu som do makier pridal aj upozornenie s možnosťou jeho ukončenia bez prevedenia týchto zmien.

Preto, pokiaľ chcete v texte používať rôzne formáty textu, najprv spustíte tieto makrá a až následne meňte formáty textu. Pokiaľ vám to nevyhovuje, nezostáva vám nič iné, ako príslušné zmeny previesť manuálne, čo je však pri dlhých dokumentoch časovo naozaj náročná činnosť. Na druhej strane práve táto vlastnosť môže mnohým vyhovovať – jednoduchým spôsobom si práve pomocou tohto makra môžu zmeniť formát celého dokumentu, napr. v prípade, že spracovávali podklady od viacerých autorov resp. s rôznych zdrojov.

Ak náhodou nepotrebujú ich ďalšie vlastnosti (ako je vloženie prázdnych odstavcov či zmenu koncov odstavcov na zalomenie riadku), môžu v ďalších krokoch tieto zmeny vrátiť späť. Na to môže slúžiť napr. formátovanie dokumentu, kde nájdeme možnosť odstránenia prázdnych odstavcov, alebo v prípade zalomenia riadku použiť náhradu reťazca „\n“ za reťazec „\n“ s využitím regulárnych výrazov (v budúcom dieli si na to uvedieme krátke makro).

REM Makro pre zámenu odstavcov za line break (odstráni znaky CR z textu)

REM Makro funguje iba v označenom (vybranom) texte

Sub Zrusenie_Odstavcov

' Na konci odstavcov je kombinácia znakov CR+LF,

' na konci riadkov (ručné zalomenie - shift+enter) iba znak LF.

' Preto zrušenie odstavcov spočíva vlastne v odstránení znakov CR.

dim dokument, vyber as object

dim vyber_ktory, dlzka_riadku, i as long

dim pocet_CR as long

dim riadok as string

dim znak, CR, slovo as string

```

dim zamena as boolean

pocet_CR=0
CR=chr(&H0D)

' Na začiatku upozorníme, že makro mení formát odstavcov
slovo="Pozor, funkcia nastaví formátovanie odstavcov podľa posledného z nich!"&CR
+CR
slovo=slovo+"
Naozaj ju chcete spustiť?"&CR&CR
ooo=msgbox(slovo,4,"Zrušenie odstavcov") ' Dialógové okno s tlačidlami „Áno/Nie“

if ooo=6 then
' Stlačené tlačidlo "ÁNO"

Dokument=ThisComponent
vyber=dokument.getCurrentSelection() ' aktuálny výber
vyber_pocet=vyber.getCount() ' Počet výberov

for vyber_ktory=0 to vyber_pocet-1
riadok = vyber.getByIndex(vyber_ktory).getString() ' reťazec z výberu
dlzka_riadku=len(riadok)
zamena=false

rem pre všetky znaky z príslušného výberu
for i=1 to dlzka_riadku
znak=mid(riadok,i,1) ' spracovávame i-ty znak z výberu
if instr(CR,znak)<>0 then
' znak je CR, odstránime ho a spočítame počet odstránených odstavcov
mid(riadok,i,1,"") ' Nájdený znak CR nahradíme prázdny znakom
dlzka_riadku=len(riadok)
pocet_CR=pocet_CR+1 ' Pripočítame práve odstránený znak CR, t.j. odstránený odstav
ec
zamena=true ' Indikácia, že sme odstránili znak CR
endif
next i
if zamena then

```

' Ak sme našli znak CR (t.j. koniec odstavca), zameníme starý text za nový, čím odstráni me odstavce

```
vyber.getByIndex(vyber_ktory).setString(riadok)
```

```
endif
```

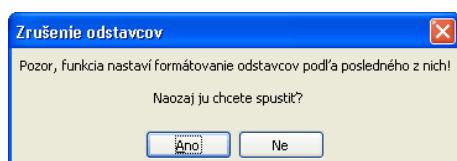
```
next vyber_ktory
```

```
endif
```

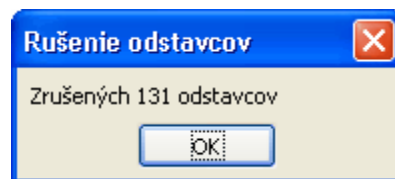
' Na konci si vypíšeme počet odstránených odstavcov

```
msgbox("Zrušených "+pocet_CR+" odstavcov",0,"Rušenie odstavcov")
```

End Sub



Obrázok 16: Upozornenie, že dôjde k preformátovaniu odstavcov



Obrázok 15: Oznámenie o počte zrušených odstavcov

REM Makro pre pridanie prázdnych odstavcov za každý iný odstavec

REM Makro funguje iba v označenom (vybranom) texte

Sub Zdvoynasobenie_Odstavcov

' Na konci odstavcov je kombinácia znakov CR+LF,

' preto za znakom LF pridáme ďalšiu kombináciu CR+LF.

dim dokument, vyber as object

dim vyber_ktory, dlzka_riadku, i as long

dim pocet_CR as long

dim riadok as string

dim znak, CR, LF, slovo as string

dim zamena as boolean

pocet_CR=0

CR=chr(&H0D)

LF=chr(&H0A)

' Na začiatku upozorníme, že makro mení formát odstavcov

slovo="Pozor, funkcia nastaví formátovanie odstavcov podľa posledného z nich!" + CR
+ CR

```

slovo=slovo+"
cete spustit?">CR+CR
    ooo=msgbox(slovo,4,"Zdvojnásobenie odstavcov") ' Dialógové okno s tlačidlami „Áno/Ni
e“

if ooo=6 then
    ' Stlačené tlačidlo "ÁNO"

    Dokument=ThisComponent
    vyber=dokument.getCurrentSelection() ' aktuálny vyber
    vyber_pocet=vyber.getCount() ' Počet výberov

    for vyber_ktory=0 to vyber_pocet-1
        riadok = vyber.getByIndex(vyber_ktory).getString() ' reťazec z výberu
        dlzka_riadku=len(riadok)
        zamena=false

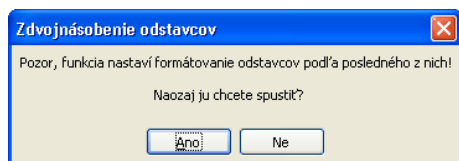
        rem pre všetky znaky z výberu
        for i=1 to dlzka_riadku
            znak=mid(riadok,i,1) ' spracovávame i-ty znak z výberu
            if instr(LF,znak)<>0 then
                ' znak je LF, nahradíme ho LF+CR+LF čím pridáme odstavec a spočítame počet zámen
                mid(riadok,i,1,LF+CR+LF)
                ' Posunieme index o dva vložené znaky (CR+LF)
                i=i+2
                dlzka_riadku=len(riadok)
                pocet_CR=pocet_CR+1 ' Počet vložených prázdnych odstavcov
                zamena=true ' Indikácia vloženia prázdneho odstavca
            endif
        next i
        if zamena then
            ' Do textu sme vložili prázdny odstavec, preto nahradíme starý text za nový, čím vlastne
            prevedieme vlastnú úpravu
            vyber.getByIndex(vyber_ktory).setString(riadok)
        endif
    next vyber_ktory
endif

```

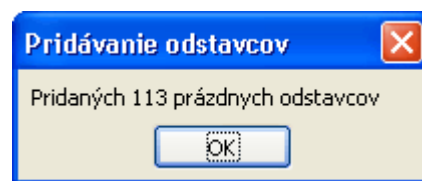
' Na konci si vypíšeme počet vložených prázdnych odstavcov

msgbox("Pridaných "+pocet_CR+" prázdnych odstavcov",0,"Pridávanie odstavcov")

End Sub



Obrázok 17: Upozornenie, že dôjde k preformátovaniu odstavcov



Obrázok 18: Oznámenie o počte vložených odstavcov

Rôzne zámeny

V dnešnom pokračovaní seriálu o programovaní makier v OpenOffice.org si znovu uvedieme makrá, pomocou ktorých dokážeme formátovať dokument a to konkrétne v oblasti zámen postupnosti znakov za typografické znaky.

Pri písaní textu bežne používame tri bodky pre označenie pokračovania textu (niekedy so skrytým podtextom). V typografii sa však pri sádzaní textu nepoužívajú tri bodky, ale špeciálny znak, ktorý nazvem „trojbodka“ (prosím, keby ma jazykovedci nekameňovali za tento názov, pre bežných používateľov je však dosť názorný). Hoci sa dá v OpenOffice.org nastaviť, aby sa tri bodky automaticky nahradzovali týmto znakom, nedôjde k tomu v tom prípade, ak nasledujú tesne za predchádzajúcim slovom (t.j. nie je medzi nimi a slovom medzera), čo je z gramatického hľadiska veľmi častý prípad. Tak isto k tejto zmene nedôjde vtedy, ak po ich napísaní nestlačíme medzeru či klávesu „Enter“ (napr. na konci textu).

Podobná problematika sa týka aj spojovníkov, pomlčiek a rozdeľovníkov. Z typografického hľadiska majú všetky tieto znaky rôzne dĺžky – spojovník a rozdeľovník je v skutočnosti kratšia čiarka, ktorá nemá medzeru medzi sebou a slovom, kým pomlčka je dlhšia a pred a aj za ňou sa nachádza medzera. OpenOffice.org automaticky dokáže upraviť spojovník na pomlčku (pre ktorú vlastne nemáme priamu klávesu) v prípade, ak sa nachádza medzi medzerami. Pokiaľ však pomlčku píšeme za text, ktorý je ukončený bodkou (napr. nejaká skratka), potom túto zámenu automaticky neprevedie. Pravdaže, tento problém dokážeme riešiť aj tým, že si zadefinujeme do automatických opráv napr. náhradu dvoch spojovníkov za pomlčku. Toto riešenie však nemusí vyhovovať každému a preto je vhodné, aby sme spolu so zámenou troch bodiek za „trojbodku“ naprogramovali aj zámenu spojovníka za pomlčku, ak sa nachádza medzi medzerami.

A aby sme nezabudli na sľub, ktorý sme si dali minule, uvedieme si dnes aj makro, pomocou ktorého dokážeme zameniť ručné zalomenie riadku (klávesová skratka „Shift + Enter“) za zalomenie odstavca (klávesa „Enter“). Podobne, ako sme to robili aj v minulosti, aj teraz použijeme makrá, ktoré sme si už doteraz naprogramovali a ktoré sa nachádzajú v module „Formatovanie_funkcie“.

REM Makro pre rôzne zámeny (napr. troch bodiek za „trojbodku“)

function f_Rozne_zameny as long

dim V_com(), Co(), Za() as string

```

dim n as long
dim kolko as long

Formatovanie_funkcie.init
' Budeme zamieňať tri bodky za trojbodku
' a obyčajné pomlčky za dlhú pomlčku medzi dvomi medzerami.
' Pri hľadaní použijeme regulárne výrazy, lebo spojovníky môžu začínať a končiť
' ľubovoľnou medzerou - preto je definícia vyhľadávacích reťazcov trochu komplikovanejš
ia

V_com()=array("\.\.", "[space:]+-[space:]+")
Co()=array("...", "-") ' tri bodky a spojovník
Za()=array("...", "-,") ' „trojbodka“ a pomlčka

kolko=0
for n=lbound(V_com()) to ubound(V_com())
    kolko=kolko
+Formatovanie_funkcie.Vymen_hladanim(V_Com(n),Co(n),Za(n),true,false,false)
next n
f_Rozne_zameny=kolko
end function

REM Pokiaľ si chceme vypísať počet zámen, môžeme volať predchádzajúcu funkciu touto
procedúrou
sub Rozne_zameny
    msgbox("Prevedených "+f_Rozne_zameny+" zámen.",0,"Rôzne zámeny")
end sub

REM Makro pre zámenu zalomenia riadku za koniec odstavca
function f_Koniec_odstavca as long
    Formatovanie_funkcie.init
    f_Koniec_odstavca =Formatovanie_funkcie.Vymen("\n","\n",true,false,false)
end function

REM Pokiaľ si chceme vypísať počet zámen koncov riadkov za odstavce
sub Koniec_odstavca

```

```

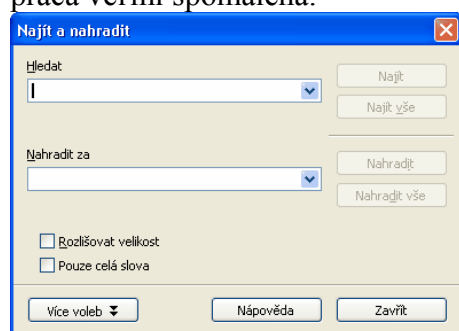
msgbox("Prevedených "+f_Koniec_odstavca+" zámien.",0,"Zámena konca riadku za odsta
vec")
end sub

```

Vlastný dialóg pre zámenny

V ďalšom pokračovaní seriálu o programovaní makier v OpenOffice.org si vytvoríme vlastný dialóg pre hľadanie a nahrádzanie textu.

Pri praktickej práci s OpenOffice.org často používame dialógové okno „Nájsť a nahradit“ (klávesová skratka „Ctrl + F“). Mnohým však nevyhovuje to, že napríklad zaškrŕtávacie políčko pre regulárne výrazy sa zobrazí až po stlačení tlačidla „Viac volieb“, pričom si OpenOffice.org nepamätá ani jeho predchádzajúce nastavenie. Toto platí pre každé otvorenie tohto dialógového okna, takže ak často používame práve regulárne výrazy, je táto práca veľmi spomalená.

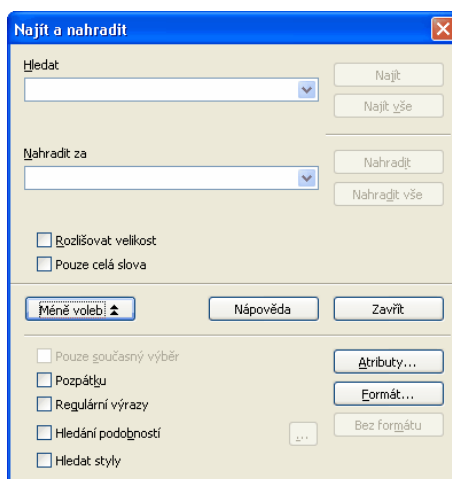


Obrázok 19: Štandardný dialóg pre zámenny

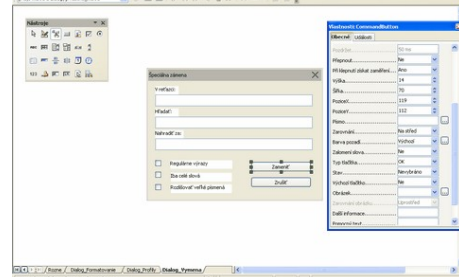
Preto si teraz naprogramujeme vlastné dialógové okno pre nahrádzanie textu. V doterajších dieloch sme si naprogramovali dve vlastné funkcie, pomocou ktorých sme nahrádzali jeden reťazec druhým. Konkrétne ide o funkcie „Vymen“ a „Vymen_hladanim“, ktoré sme si uložili do modulu makier „Formatovanie_funkcie“.

V stručnosti si pripomeňme, že funkcia „Vymen“ využíva štandardné služby pre zámenu reťazcov s tým, že môžeme nastaviť tri logické parametre pre vyhľadávanie – regulárne výrazy, celé slová a rozlišovanie veľkosti písmen. Funkcia „Vymen_hladanim“ je už zložitejšia, pretože ako parametre nemáme dva, ale tri reťazce. Je to tak preto, lebo v tejto funkcii zamieňame celý nájdený reťazec, ale iba jeho vnútornú časť. Pravdaže, aj tu máme možnosť použiť už spomenuté tri logické parametre pre vyhľadávanie.

Tieto funkcie sa nám preto priamo ponúkajú na to, aby sme si pripravili vlastný dialóg pre vyhľadávanie. Dialóg nazveme „Dialog_Zamena“ a musí obsahovať aspoň tieto položky: tri textové polia, ktoré nazveme „V_retazci“, „Hladat“ a „Nahradit za“; tri zaškrŕtávacie políčka – „Regularne_vyrazy“, „Cele_slova“ a „Velke_pismena“; dve tlačidlá – „Zamenit“, ktorému priradíme typ tlačidla „OK“ a „Zrusit“, ktorému priradíme typ



Obrázok 20: Rozšírené možnosti štandardného dialógu pre zámenny



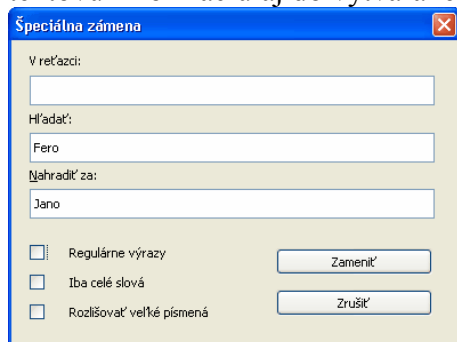
Obrázok 21: Návrh vlastného dialógu pre zámenny

„Cancel“ („Zrušiť“). Okrem toho si jednotlivé položky doplníme vhodnými textovými popismi.

Vlastnej tvorbe dialógu sa už, samozrejme, nebudeme venovať, pretože na to boli určené predchádzajúce diely, kde sme sa tomu podrobne venovali. Teraz nám

nezostáva už nič iné, iba naprogramovať vlastné makrá pre obsluhu tohto dialógu s následným volaním už spomínaných vyhľadávacích funkcií.

Aby sme nemuseli zadávať špeciálne zaškrŕavacie políčko pre rozlíšenie situácie, kedy chceme využiť funkciu „Vymen“ a kedy funkciu „Vymen_hladanim“, rozlíšime to automaticky podľa toho, či zadáme text do dialógovej textovej premennej V“_retazci“. Ak áno, tak budeme používať funkciu „Vymen_hladanim“, ak nie, tak funkciu „Vymen“. Napokon, z vlastného makra je dosť jasné a prípadne si toto môžete uviesť ako doplňujúcu textovú informáciu aj do vytváraného dialógu.



Obrázok 22: Obyčajná záměna pomocou vlastného dialógu

Makrá si pre lepšiu orientáciu uložíme do modulu „Specialna_zamena“. Pre vlastné vyvolanie s pustením dialógu so záměnami voláme procedúru „Specialne_zamen“, pričom na konci si vypíšeme, koľko výmen bolo urobených.

REM Premenné si zadefinujeme ako global, aby sa pamätali aj pri ďalšom spustení

REM Pozor! Pri prvom spustení môžu obsahovať náhodné veličiny

global zamen_V_retazci, zamen_Hladat, zamen_Nahradiť_za as string

global zamen_Regularne_vyrazy, zamen_Cele_slova, zamen_Velke_pismena as boolean

sub Init_dialog_zaměny(oDialog as object)

REM Inicializácia prvkov dialógu

oDialog.model.V_retazci.Text = zamen_V_retazci

oDialog.model.Hladat.Text = zamen_Hladat

oDialog.model.Nahradiť_za.Text = zamen_Nahradiť_za

if zamen_Regularne_vyrazy then

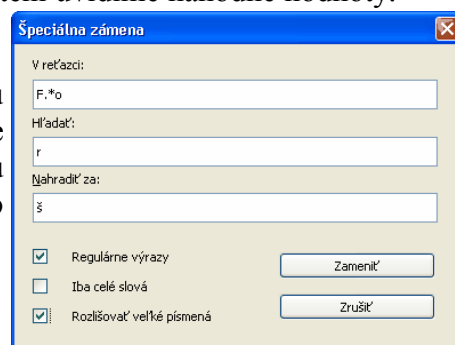
oDialog.model.Regularne_vyrazy.State=1

else

oDialog.model.Regularne_vyrazy.State=0

endif

if zamen_Cele_slova then



Obrázok 23: Špeciálna záměna pomocou vlastného dialógu

```

oDialog.model.Cele_slova.State=1
else
oDialog.model.Cele_slova.State=0
endif
if zamen_Velke_pismena then
oDialog.model.Velke_pismena.State=1
else
oDialog.model.Velke_pismena.State=0
endif

end sub

sub vykonaj_zamenu(oDialog as object)
REM Vlastná zámena podľa zadaných hodnôt v dialógu

dim kolko as long

REM Príprava globálnych premenných
zamen_V_retazci = oDialog.model.V_retazci.Text
zamen_Hladat = oDialog.model.Hladat.Text
zamen_Nahradit_za = oDialog.model.Nahradit_za.Text
zamen_Regularne_vyrazy = oDialog.model.Regularne_vyrazy.State=1
zamen_Cele_slova = oDialog.model.Cele_slova.State=1
zamen_Velke_pismena = oDialog.model.Velke_pismena.State=1

REM Príprava dokumentu pre zámenu
Formatovanie_funkcie.Init
if zamen_V_retazci="" then
REM Ak nezádáme reťazec "V_retazci", zamieňame priamo
kolko=Formatovanie_funkcie.Vymen(zamen_Hladat, zamen_Nahradit_za,
zamen_Regularne_vyrazy, zamen_Cele_slova, zamen_Velke_pismena)
else
REM Ak zadáme reťazec "V_retazci", zanieňame špeciálne
kolko=Formatovanie_funkcie.Vymen_hladanim(zamen_V_retazci, zamen_Hladat,
zamen_Nahradit_za, zamen_Regularne_vyrazy, zamen_Cele_slova, zamen_Velke_pismena)
endif
msgbox("Prevedných "+kolko+" zámien.", 0,"Špeciálne zámeny")

```

```
end sub
```

```
sub Specialne_zamen
```

```
dim oDialog as object
```

```
oDialog=CreateUnoDialog(DialogLibraries.Vlastne_makra.Dialog_Zamena)
```

```
Init_dialog_zameny(oDialog)
```

```
if oDialog.Execute()=1 then
```

```
vykonaj_zamenu(oDialog)
```

```
endif
```

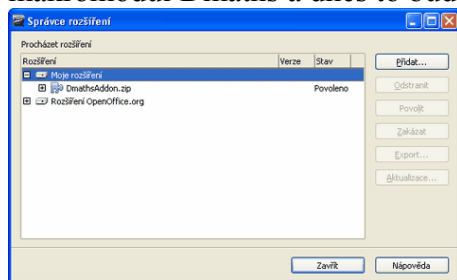
```
oDialog.dispose()
```

```
end sub
```

Rozšírenie pre typografické úpravy textu – TypoJTB

V doterajších dieloch sme sa venovali predovšetkým úpravám textu podľa typografických pravidiel. Predstavme si preto makromodul TypoJTB z dielne Tomáša Bílka, ktorý je venovaný práve tejto problematike.

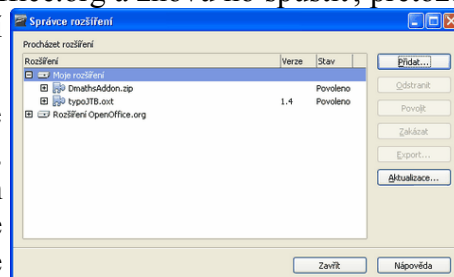
Ako sme povedali už v prvom dieli tejto časti seriálu o programovaní makier pre OpenOffice.org, predstavíme si aj niekoľko zaujímavých makier, makromodulov (či lepšie povedané rozšírení) od iných autorov. Prvou lastovičkou v tejto oblasti bol francúzsky makromodul Dmaths a dnes to bude české rozšírenie TypoJTB.



Obrázok 24: Inštalácia TypoJTB

Toto rozšírenie je distribuované vo formáte „OXT“ a do prostredia sa inštaluje cez správcu rozšírení (postupnosť volieb menu „Nástroje – Správca rozšírení“). Tam si vyberieme časť, kde chceme rozšírenie nainštalovať (t.j. či pre seba, alebo pre celý OpenOffice.org), stlačíme tlačidlo „Pridať“ a po vyhľadání otvoríme príslušný súbor (v tomto prípade „typoJTB.oxt“). Rozšírenie sa následne automaticky nainštaluje. Pre jeho spustenie je potrebné zavrieť OpenOffice.org a znovu ho spustiť, pretože údaje o rozšíreniach sa načítavajú pri spúšťaní OpenOffice.org.

Hoci to sem priamo nepatrí, uvedme si ešte, že v prípade, ak chceme nejaké rozšírenie odinštalovať, urobíme to tak isto v správcovi rozšírení stlačením tlačidla „odstrániť“. Pokiaľ chceme, môžeme niektoré rozšírenia dočasne zakázať a znovu povoliť – príslušné funkcie sú dosť jasné z možností, ktoré vidíme v tomto



Obrázok 25: Zoznam rozšírení po nainštalovaní TypoJTB

správcovi. Pozor, všetky zmeny sa prejavia, podobne ako pri inštalácii, až po opätovnom spustení OpenOffice.org.



Obrázok 26: Lišta TypoJTB

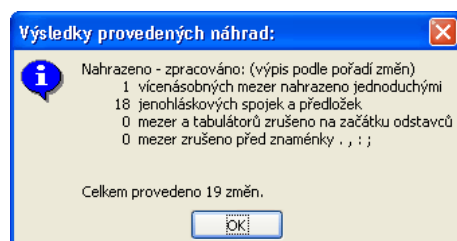
Po nainštalovaní sa v prostredí OpenOffice.org zobrazí nová nástrojová lišta TypoJTB, kde sa nachádzajú ikony sprístupňujúce jeho všetky funkcie. Tieto môžeme rozdeliť do troch častí (takto sú aj usporiadané na nástrojovej lište) – úpravy textu, informácie (štatistika) a pomocné funkcie. Pozrime sa teraz na ne podrobnejšie.

Typografické úpravy

Táto funkcia určite patrí medzi tie najzaujímavejšie, pretože práve jej niektorým možnostiam sme sa venovali aj my pri programovaní makier. Umožňuje úpravy, ako je mazanie viacnásobných medzier, vloženie nezalomiteľných medzier za jednoznakové predložky a spojky, zámenu troch bodiek za znak výpusťky (doteraz sme tento znak označovali ako „trojbodka“), opraviť medzery okolo interpunkčných znamienok (zrušiť medzery pred a pridať medzery za nimi) atď.

Makro umožňuje jednotlivé nastavenia uložiť, takže pri budúcom spustení ich nemusíme nastavovať znovu. Momentálne ešte makro neumožňuje vytvárať používateľské profily, ale autorovi som už navrhol túto možnosť, takže je možné, že táto funkcia bude do tejto časti implementovaná, čo by určite mnohí používatelia privítali.

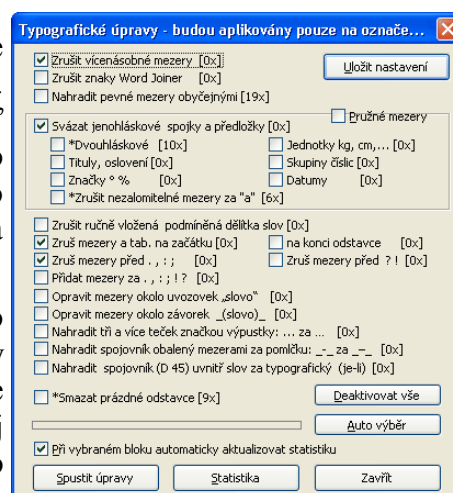
Veľmi zaujímavé a potrebné je aj to, že pri tejto funkcii sa automaticky rozoznáva, či máme označený



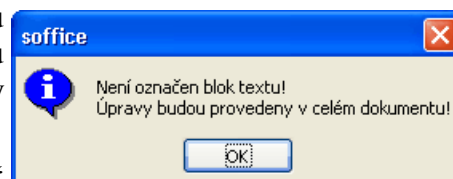
Obrázok 28: Oznam o počte provedených úprav

v celom dokumente.

Pokiaľ po výbere funkcie „Typografické úpravy“ prevedieme tzv. štatistiku (pri označenom texte môžeme využiť aj možnosť jej automatickej prípravy), v hranatých zátvorkách, ktoré sa nachádzajú pri jednotlivých možnostiach uvidíme počet výskytov prípadných úprav. Teraz môžeme pomocou tlačidla „Auto výber“ automaticky označiť iba tie úpravy, kde máme nenulový výskyt príslušných úprav. Samozrejme, následným ručným nastavením môžeme zabrániť prevedeniu prípadných neželaných zmien. Pomocou tejto možnosti však môžeme urýchliť následnú úpravu, lebo sa zbytočne naprázdno nespúšťajú tie úpravy, ktoré de facto nie sú potrebné.



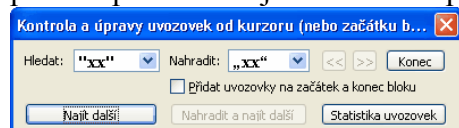
Obrázok 27: Možnosti typografických úprav



Obrázok 29: Oznámenie, že nebol označený text

Kontrola úvodzoviek a zátvoriek

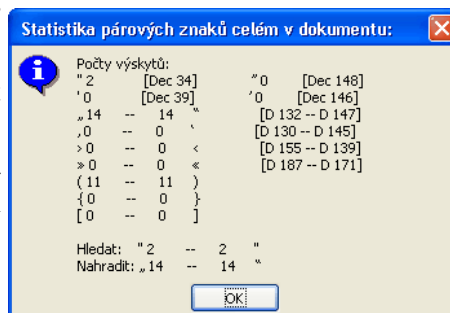
Druhá funkcia sa venuje úvodzovkám a zátvorkám, pričom umožňuje ich rôzne vzájomné nahrádzanie (dokonca aj úvodzovky za zátvorky či naopak). Toto ocenia napr. tí, ktorí píšú zdrojové texty programov v prostredí OpenOffice.org, pričom práve napr. textové úvodzovky „“, ktorými OpenOffice.org automaticky ohraňuje text nie sú vhodné, pretože v programe potrebujú úvodzovky ". Počas programovania sa však nemusia zaoberať touto „drobnosťou“, pretože pomocou tejto funkcie na pár krokov dokážu urobiť všetky potrebné zmeny.



Obrázok 30: Kontrola úvodzoviek a ohľadu na text, ktorý sa nachádza vo vnútri. Ďalej tu zátvoriek

nájde možnosť uzatvorenia označeného textu do úvodzoviek alebo zátvoriek. Výber znakov, ktorými sa ohraňuje tento text prevádzame v časti „Nahradiť“ a vlastné ohrazenie vykonáme zaškrtnutím voľby „Pridať úvodzovky na začiatok a koniec bloku“ a následným potvrdením pomocou tlačidla „Pridať okolo bloku“.

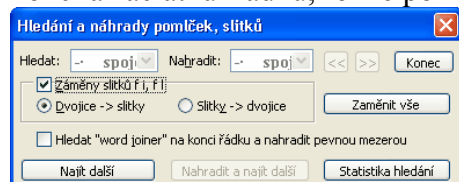
V rámci tejto funkcie je určite zaujímavá aj štatistika, kde nám makro vypíše všetky výskyty znakov, ktoré môžeme vyhľadávať a nahrádzať. Zároveň nám vypíše počet výskytov znakov „Hľadať“ a aj počet výskytov znakov „Nahradiť“, takže hneď vidíme, či musíme prípadnú zámenu vôbec spúšťať.



Obrázok 31: Štatistika úvodzoviek a zátvoriek

Hľadanie a nahradzovanie typografických znakov

Táto funkcia umožňuje vyhľadať dvojice znakov „fi“ a „fl“ a nahradiť ich príslušnými zliatkami, alebo naopak. Ďalej umožňuje nájsť a vzájomne nahradiť rôzne iné typografické značky (spojovník, typografický spojovník – t.j. taký, ktorý sa v prípade potreby opakuje na konci a začiatku riadku, rôzne pomlčky a pod.

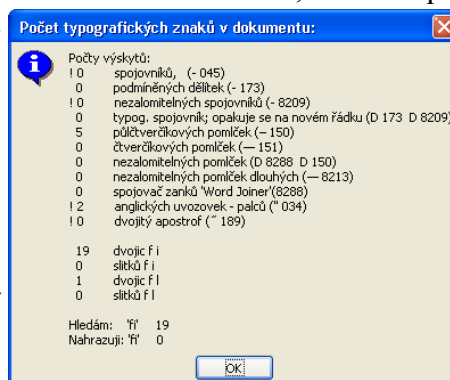


Obrázok 32: Hľadanie a nahradzovanie typografických znakov

Pokiaľ používate „pružnú“ nezalomiteľnú medzeru (t.j. znak „Word Joiner“), mohli ste sa už stretnúť s prípadom, kedy nefungovala úplne správne a spojenie, ktoré malo zostať na jednom riadku bolo rozdelené do dvoch riadkov. Táto funkcia dokáže nájsť tieto prípady a príslušné znaky „Word Joiner“ nahradí nezalomiteľnou medzerou, čím sa uvedené problémy odstránia. Podobne, ako pri predchádzajúcej funkcii, aj tu máme možnosť zobrazenia štatistiky, t.j. počtu výskytov znakov, ktoré môžeme v rámci tejto funkcie vyhľadávať a vzájomne zamieňať.

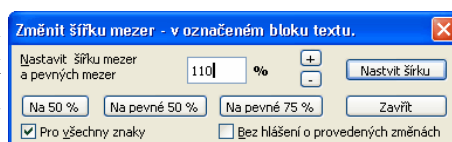
Nastavenie šírky medzier a znakov

V poradí štvrtá funkcia, pomocou ktorej môžeme upravovať text, je zameraná na šírku písma. Umožňuje vo vybranej časti dokumentu nastavovať šírku medzier a prípadne aj ostatných znakov. Ak neoznačíme žiaden text, zmeny sa budú prevádzať v celom dokumente, ale iba pre medzery a aj to fungujú iba prednastavené hodnoty na troch tlačidlách – zmenšenie medzier na 50%, prevod



Obrázok 33: Štatistika typografických znakov

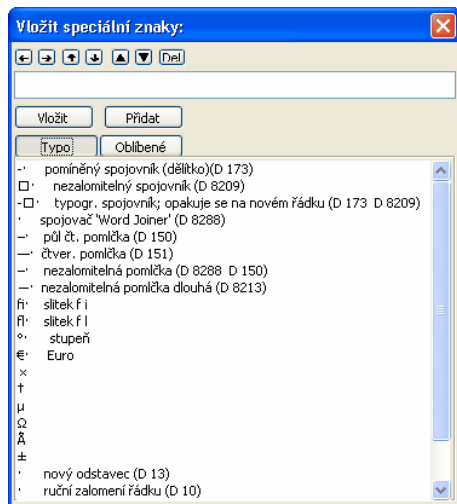
obyčajných medzier na nedeliteľné (pevné) medzery a ich zmenšenie na 50% alebo 75%. Tieto tlačidlá však nemôžeme použiť v prípade, keď chceme zmeniť šírku všetkých znakov. Vtedy musíme príslušnú percentuálnu zmenu uviesť priamo zadáním jej číselnej hodnoty.



Obrázok 34: Nastavenie šírky medziera znakov

Vkladanie špeciálnych znakov

Hoci je vkladanie špeciálnych znakov prístupné aj cez menu („Vložiť – Špeciálny symbol“), táto možnosť je dosť nepraktická vtedy, ak často vkladáme iba pár znakov. A práve vtedy dokážeme výhodne použiť funkciu rozšírenia TypoJTB, kde nájdeme niekoľko týchto znakov preddefinovaných. Pracovné okno tejto funkcie je rozdelené do dvoch záložiek – „Typo“ a „Oblíbené“.



Obrázok 35: Vkladanie špeciálnych znakov

Ide v podstate o delenie čisto formálne, pretože do ktorejkoľvek z týchto záložiek si môžeme pridávať vlastné znaky či dokonca celé texty. Znaký môžeme zapisovať priamo, alebo zadáním ich hexadecimálneho alebo dekadického kódu. Ku každému znaku alebo textu môžeme pridať aj poznámku, ktorá sa bude zobrazovať pri ich popise, ale do vlastného textu sa nebude vkladat'. Pravdaže, vložené (a aj preddefinované znaky) či texty, pokiaľ nám nevyhovujú alebo ich už nepoužívame, môžeme kedykoľvek vymazať. Vlastné vloženie potrebného znaku prevádzame nakoniec obyčajným dvojklikom.

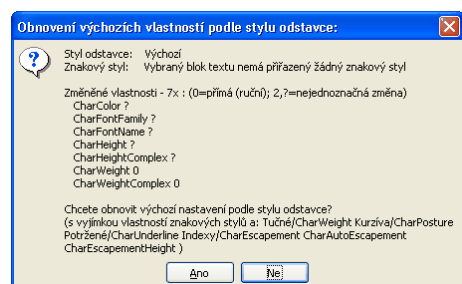
Informácie o kódach znakov

Pokiaľ potrebujete zistiť hexadecimálne a dekadické kódy znakov, stačí označiť príslušný text a spustiť túto funkciu. Následne sa vypíšu všetky vybrané znaky pod sebou s tým, že sú očíslované, výpis obsahuje dekadický a hexadecimálny kód a na koniec aj znak, ktorému tento kód prislúcha.



Obrázok 36: Informácie o kódach znakov

Rozdiely oproti štýlom

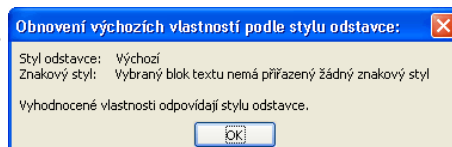


Obrázok 37: Obnovenie podľa pôvodného štýlu

nesprávne manuálne zasiahol.

Pozor, funkcia spätného nastavenia funguje iba pre vybraný text (resp. znak, na ktorom sa práve nachádza

Pokiaľ pri písaní zmeníme v texte nastavenia, ktoré nie sú zhodné s nastaveniami, ktoré sú definované v príslušnom použitom štýle, táto funkcia nám ich vypíše a zároveň nám ponúkne možnosť spätného nastavenia podľa príslušného štýlu. Túto funkciu ocenia najmä tí používatelia, ktorí musia dávať dohromady podklady z rôznych zdrojov, čím veľmi rýchlo a na jeden krok urobia všetky potrebné zmeny bez toho, aby museli pracne vyhľadávať, kde používateľ

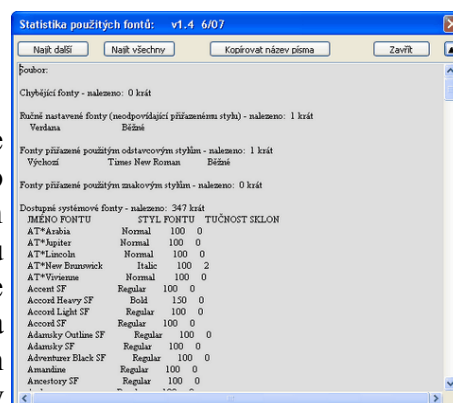


Obrázok 38: Oznámenie, že nemusíme robiť obnovenie

kurzor), takže pokiaľ ju chceme aplikovať na celý dokument, musíme ho celý označiť. Pokiaľ sa žiadne zmeny nenájdu, program nám to oznámi jednoduchým výpisom a žiadne spätné úpravy (lebo nie je aké) nám neumožní vykonať.

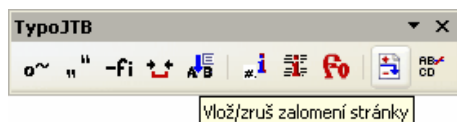
Štatistika použitých fontov

Posledná štatistická funkcia, ktorú nám rozšírenie TypoJTB ponúka, je štatistika použitých fontov. Táto funkcia je veľmi potrebná najmä pri prezeraní cudzích dokumentov, pretože nám povie nielen to, ktoré fonty sú v dokumente použité a aké fonty máme v systéme nainštalované, ale, čo je hľadám najdôležitejšie, aj to, či sa v dokumente náhodou nepoužíva font, ktorý náš systém nepozná (t.j. ktorý by nám mohol spôsobovať problémy pri následnej tlači, exporte do PDF formátu a pod.).



Obrázok 39: Štatistika použitých fontov

Vloženie (zrušenie) zalomenia stránky

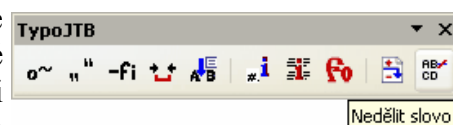


Obrázok 40: Tlačidlo pre vloženie/zrušenie zalomenia stránky

Táto funkcia vloží ručné zalomenie stránky za aktuálny odstavec, resp. zruší toto zalomenie, ak sa pred ním nachádza. Pretože pri jej použití sa vôbec nemusíme premiestňovať na začiatok alebo koniec odstavca, je toto vkladanie alebo mazanie naozaj veľmi pohodlné a značne uľahčuje prácu.

Nedeliť slovo

Posledná doplnková funkcia, ktorú nám rozšírenie TypoJTB poskytuje, je možnosť označenia slov, ktoré nechceme, aby sa delili. Makro túto situáciu rieši veľmi jednoducho – pre príslušné slovo, na ktorom sa nachádzame (alebo môžeme mať označených aj viacej slov) nastaví jazyk na hodnotu „žiadny“, takže pri následnom delení slov sa nemôže použiť príslušný slovník a slovo sa nerozdelí. Toto riešenie však prináša aj určitú nevýhodu – takéto slovo nepodlieha kontrole gramatiky, ale to je už „programátorská“ daň, ktorú musíme za túto možnosť „zaplatiť“.



Obrázok 41: Tlačidlo pre nastavenie nedeliteľnosti slova

Internet: www.volny.cz/macrojtb/index.html

Zdroje informácií

Chcete si naprogramovať nejaké makro, potrebujete zistiť premenné metódy, ktoré používa OpenOffice.org a neviete ako na to? Dnes si ukážeme niekoľko možností, ako to dokážete zistiť.

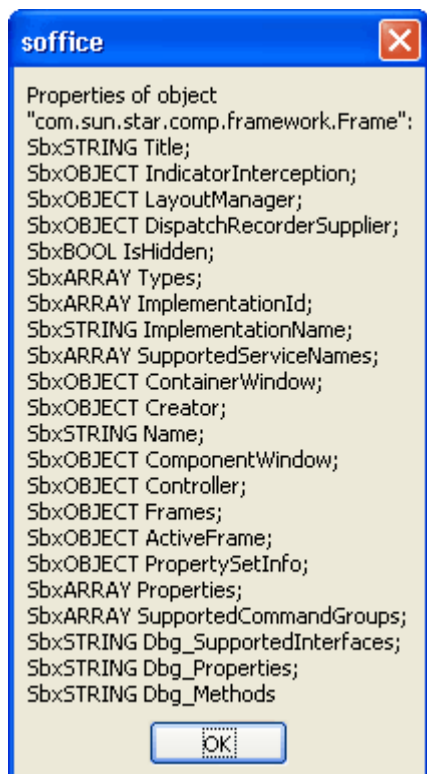
V predchádzajúcich 62 dieloch (áno, bolo ich presne 62) seriálu o programovaní makier sme si ukazovali rôzne makrá, pričom sme používali najrôznejšie premenné a metódy, ktoré nám ponúka OpenOffice.org. Samozrejme, mnohým už prišla na um otázka, ako a kde sa dajú získať podrobnejšie informácie.

Analýza premenných

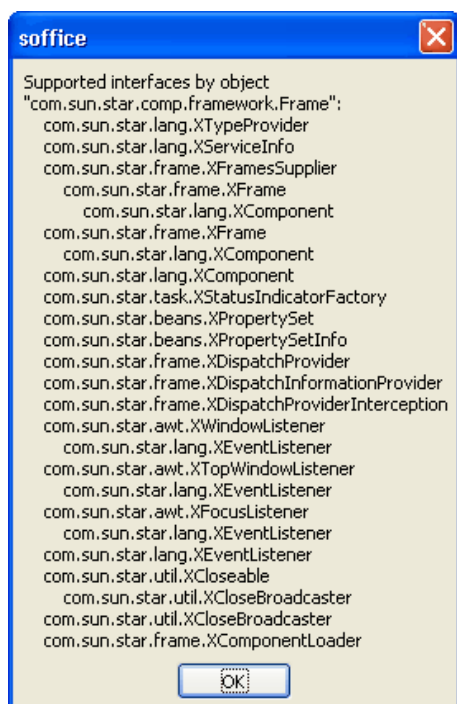
Pokiaľ poznáme premennú, ktorú potrebujeme iba preskúmať, ponúka nám nástroj priamo jazyk StarOffice Basic. Objekty majú totiž v sebe zabudované aj dva reťazce, ktoré obsahujú základné informácie:

DBG_properties – reťazec, ktorý obsahuje všetky vlastnosti objektu

DBG_methods – reťazec, ktorý obsahuje všetky metódy objektu



Obrázok 42: Zoznam vlastností



Obrázok 44: Zoznam rozhraní

Väčšina objektov má zabudovaný aj tretí reťazec DBG_supportedInterfaces, ktorý obsahuje informácie o rozhraniach, ktoré objekt používa.

Napríklad pomocou nasledujúceho makra dokážeme zistiť potrebné údaje pre objekt „ThisComponent.CurrentController.Frame“:

sub metody

dim premenna as object

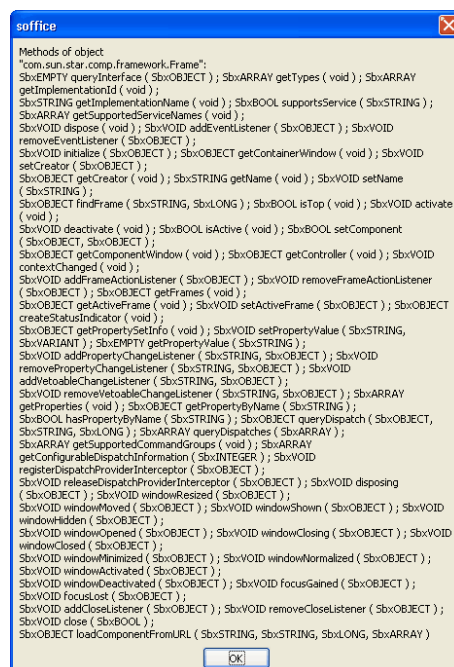
premenna=ThisComponent.CurrentController.Frame

msgbox premenna.DBG_properties

msgbox premenna.DBG_methods

msgbox premenna.DBG_supportedInterfaces

end sub



Obrázok 43: Zoznam metód

Nevýhoda tohto riešenia je taká, že chyba akýkoľvek bližší popis a preto mnohokrát nezostáva nič iné, iba metódou pokusov a omylov zisťovať, na čo sa ktorá vlastnosť alebo metóda používa.

Ako vhodnejší spôsob sa javí použiť už hotové makro XrayTool, ktoré nielenže dokáže vypisovať tieto údaje, ale ich vypisuje aj v omnoho prehľadnejšej forme. Makro je možné stiahnuť v češtine zo stránok Tomáša Bílka, autora minule predstaveného makra TypoJTB. Jeho inštaláciu si schválne nebudeme uvádzať, pretože už musíme predpokladať, že čitatelia tohto seriálu to dokážu vykonať sami.

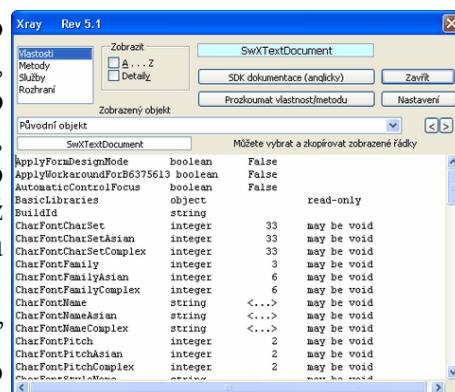
Po jeho nainštalovaní je potrebné ešte naprogramovať dve malé procedúry. V prvej budeme inicializovať makro XrayTool a v druhej zisťovať údaje o konkrétnej premennej. Osobne mám naprogramované tieto procedúry v nasledujúcej podobe (pre spúšťanie makra používam procedúru ZistiMetody):

```
sub LoadingLibrariesXrayTool
    BasicLibraries.LoadLibrary("XrayTool")
End Sub
```

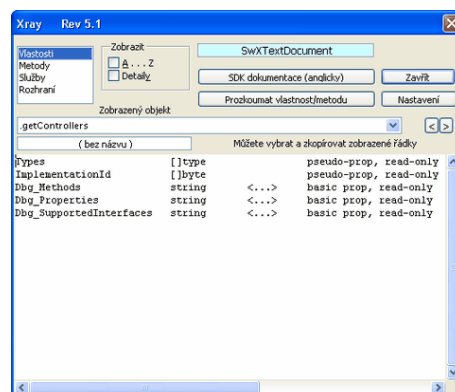
```
sub ZistiMetody
    LoadingLibrariesXrayTool
    dim premenna as object
```

```
REM Nastav podľa potreby
    premenna=ThisComponent
```

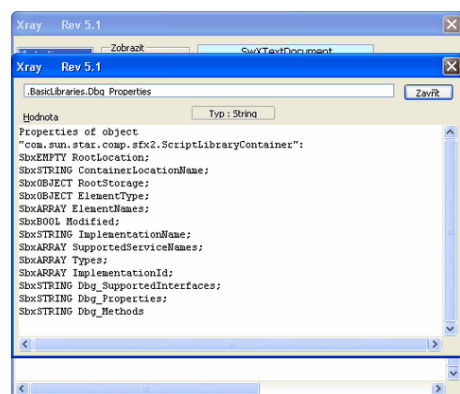
```
    xray premenna
end sub
```



Obrázok 45: Makro XrayTool



Obrázok 46: Prechádzanie po štruktúre objektovej premennej



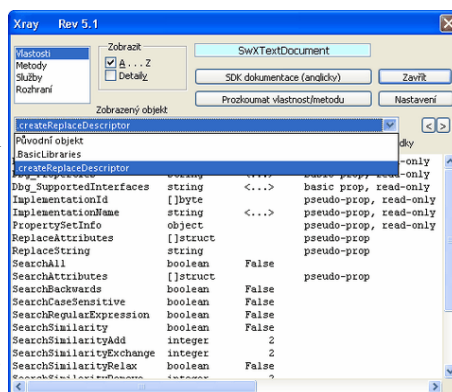
Obrázok 47: Popis vlastností

Pri takomto nastavení dokážeme zistiť údaje o všetkých premenných, pretože makro XrayTool umožňuje prechádzať po štruktúre. Pokiaľ sa presunieme na nejakú vlastnosť alebo metódu (stačí kliknúť na príslušnom riadku), pomocou tlačidla „Prozkoumat vlastnost/metodu“ sa nám zobrazia buď jej obsah, alebo vnútorná štruktúra, ktorú môžeme skúmať ďalej. Pravdaže, nájdeme tu aj už spomínané premenné `DBG_properties`, `DBG_methods` a `DBG_supportedInterfaces`.

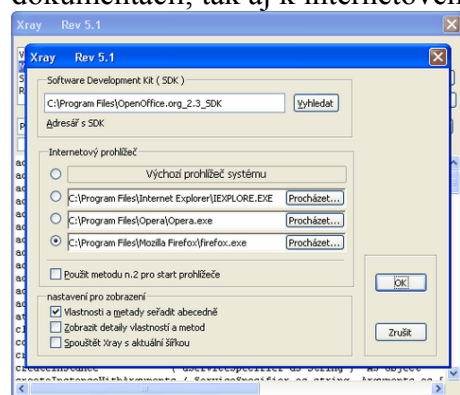
Pri vlastnom skúmaní máme možnosť prepínať medzi štyrmi údajmi – vlastnosťami, metódami, službami a rozhraním. Pokiaľ prechádzame po vnútornej štruktúre, príslušné premenné sa pamätajú v zozname zobrazených objektov, takže sa môžeme kedykoľvek medzi nimi rýchlo presúvať.

Dokumentácia

To, čo je však na makre XrayTool najzaujímavejšie, je spolupráca s „Software Development Kit (SDK)“. Pokiaľ si stiahneme a nainštalujeme túto dokumentáciu napr. zo servera ftp.linux.cz (žiaľ, je iba v angličtine), potom nám makro po stlačení tlačidla „SDK dokumentace (anglicky)“ dokáže zobrazit v nastavenom internetovom prehliadači informácie o príslušnej časti, pokiaľ sú k dispozícii (nie všetko sa v tejto dokumentácii nachádza). Nastavenie príslušných ciest (ako k inštalovanej dokumentácii, tak aj k internetovému prehliadaču) vykonáme cez tlačidlo „Nastavení“.



Obrázok 48: Zoznam zobrazených objektov



Obrázok 49: Nastavenie ciest a parametrov v XrayTool

„Useful Macro Information For OpenOffice“, ktorá je viac než len malá programátorská príručka. Osobne som z tejto dokumentácie čerpal inšpiráciu pre on-line dopĺňanie neزالomiteľných medzier.

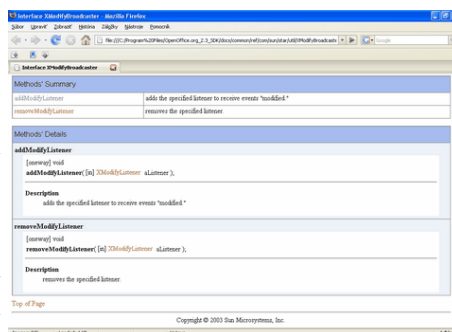
Ďalej nesmieme zabudnúť ani na základnú dokumentáciu k jazyku StarOffice Basic, ktorú na svojich stránkach docs.sun.com/app/docs/doc ponúka priamo firma SUN. Z osobných skúseností vám odporúčam stiahnuť si dokumentáciu pre StarOffice Basic od verzie 6 až po verziu 8. Prečo všetky tri? Preto, lebo počas vývoja OpenOffice.org už došlo k situácii, keď najprv platili veci, ktoré boli popísané v príručke pre „sedmičkovú“ verziu a potom sa vrátili k verzii „šestkovej“ (konkrétne sa jednalo o niektoré rozhrania pre dokumenty OpenOffice.org Calc).

Všetkým používateľom OpenOffice.org (t.j. nielen programátorom) odporúčam stránky wiki.services.openoffice.org/wiki/Documentation, kde nájdú podrobnú on-line dokumentáciu tohto produktu. Samozrejme, na svoje si prídu aj programátori, pretože časť „BASIC Guide“ je venovaná výhradne pre nich.

Týmto sme ukončili prvú veľkú časť seriálu o programovaní makier pre OpenOffice.org, v ktorej sme sa venovali predovšetkým modulu Writer. Nastal čas pre krátku dovolenku, aby si čitatelia mohli v pokoji stiahnuť a preštudovať odporúčenú dokumentáciu a aby sme mohli

Okrem uvedených možností, môžeme pri programovaní použiť aj ďalšie dokumentácie, ktoré si môžeme stiahnuť z internetu. Okrem toho nám ako inšpirácia môžu poslúžiť aj iné makrá, kde nájdeme ukrytých nielen veľa dobrých myšlienok, ale aj praktické použitie niektorých metód.

Na tomto mieste preto nesmieme zabudnúť na makrá od Andrewa Pitonyaka (www.pitonyak.org), ktoré sú naozaj na veľmi vysokej úrovni. Všetkým záujemcom odporúčam, aby si stiahli jeho dokumentáciu



Obrázok 50: Zobrazené informácie z SDK

načerpať nové inšpirácie pre pripravovanú druhú časť, ktorú budeme venovať ostatným modulom OpenOffice.org. Je pravda, že sme ani v rámci modulu Writer nespomenuli množstvo vecí (napr. makrá pre prácu s tabuľkami, makrá pre prácu so súbormi...), ale naozaj nie je možné uviesť všetko naraz.

Na úplný záver si však neodpustíme ešte jedno makro – a to makro, pomocou ktorého dokážeme spúšťať iné makro, ktorého názov mu odovzdáme ako parameter. Využitie tohto makra už ponecháme na láskavých čitateľov, ktorí si určite budú vedieť predstaviť napr. tabuľku, kde v jednej bunke zadajú názov a v druhej budú volať príslušné makro.

```
sub spusti_makro(nazov as string)
    dim document as object
    dim dispatcher as object

    document = ThisComponent.CurrentController.Frame
    dispatcher = createUnoService("com.sun.star.frame.DispatchHelper")

    dispatcher.executeDispatch(document, "vnd.sun.star.script:"+nazov+"?
language=Basic&location=application", "", 0, Array())

end sub
```

Ako konkrétny príklad použitia si uvedieme volanie makra pre počítanie znakov:

```
sub volanie_ineho_makra
    spusti_makro("Vlastne_makra.Pocitanie_znakov.Pocitanie_znakov")
end sub
```

Internet

XrayTool – www.volny.cz/macrojtb/index.html

Software Development Kit (SDK) pre verziu OpenOffice.org 2.3.0 – OOo-SDK_2.3_windows_install_en-US.exe

Dokumentácia Adrewa Pitonyaka – www.pitonyak.org/AndrewMacro.odt

On-line dokumentácia OpenOffice.org – wiki.services.openoffice.org/wiki/Documentation

On-line dokumentácia OpenOffice.org Basic – wiki.services.openoffice.org/wiki/Documentation/BASIC_Guide

Dokumentácia SUN StarOffice:

Basic Programmer's Guide StarOffice. 6.0 – 817-1826-10.pdf

StarOffice 7 Office Suite Basic Programmer's Guide – 817-1826.pdf

StarOffice 8 Programming Guide for BASIC – 819-0439.pdf