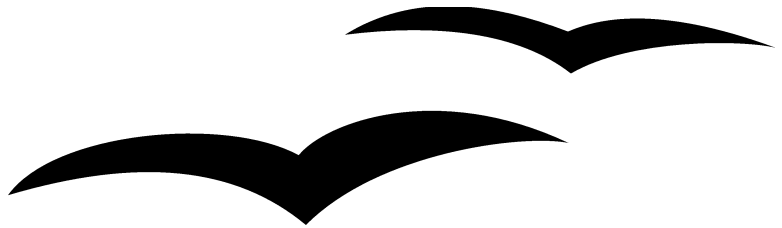


*Makrá v OpenOffice.org
Formátujeme dokumenty*



Obsah

Formátujeme dokumenty.....	3
Znovu odstraňujeme medzery.....	3
Funkcie a procedúry.....	4
Nezalomiteľná medzera.....	8
Nezalomiteľné medzery podľa typografických pravidiel.....	8
Nezalomiteľné medzery za spojkami.....	9
Znovu používame regulárne výrazy.....	12
Ale spojky a predložky nebývajú vždy osamotené.....	13
Ďalšie formátovacie funkcie.....	14
„Nezalomiteľné“ akademické tituly.....	14
Vlastná funkcia pre výmeny.....	16
„Nezalomiteľné“ dátumy.....	16
Končíme s nezalomiteľnými medzerami.....	20
„Nezalomiteľné čísla“.....	20
„Nezalomiteľné jednociferné čísla“.....	21
„Nezalomiteľné telefónne čísla“.....	21
„Nezalomiteľné merné jednotky“.....	21
Nadbytočné medzery.....	22
Čo nám hovoria normy.....	23
Nadbytočné medzery pri interpunkčných znamienkach.....	23
Chýbajúce medzery pri interpunkčných znamienkach.....	24
Malá rekapitulácia.....	27
Záverečné úpravy.....	32

Formátujeme dokumenty

Pri praktickom písaní sa mnohokrát stretávame s problémom potreby neskoršej automatickej úpravy textu podľa našich požiadavok. Pokiaľ píšete pomerne veľa (a to nielen ako ja recenzie či návody, ale aj romány, listy, dokumentáciu a pod.), postupom času zistíte, že OpenOffice.org v niektorých smeroch neposkytuje taký komfort ako iné, konkurenčné nástroje. Ako príklad spomeňme automatické vkladanie nezalomiteľných medzier za predložky. K tomu sa pridávajú ďalšie potreby, ktoré, samozrejme, už nemusia záležať od kancelárskeho balíka, pretože sú zamerané napríklad na automatické opravy preklepov, z ktorých znovu spomeňme opravu viacnásobných medzier. Všetky takéto úpravy môžeme súhrne označiť ako jednoduché formátovanie či upravovanie (opravovanie) dokumentu.

Prvé makro z tejto série sme už naznačili v predchádzajúcich dieloch nášho seriálu – makro pre odstraňovanie viacnásobných medzier. Aby sme „nevyšli z cviku“, začneme znovu týmto makrom, ktoré znovu trochu vylepšíme a pozmeníme.

Znovu odstraňujeme medzery

V poslednej verzii makra pre odstraňovanie viacnásobných medzier sme s využitím regulárnych výrazov dokázali v jednom kroku odstrániť všetky viacnásobné medzery, k čomu sme ešte pridali odstránenie medzier na začiatku a konci odstavca. Pravdaže, ani toto riešenie ešte nezahŕňa všetky možnosti takýchto preklepov, pretože môžeme mať rôzne kombinácie nezalomiteľných a zalomiteľných medzier, tabelátorov s medzerami, nezalomiteľné medzery alebo tabelátory na začiatku alebo konci odstavca a pod. Okrem toho sme medzery na začiatku alebo na konci odstavca hľadali v dvoch krokoch. Pokiaľ zohľadníme tieto pripomienky, môže makro pre vymazávanie viacnásobných medzier vyzerať takto:

SUB Viacnasobna_medzera

DIM Dokument, Vymena AS object

DIM kolko AS Long

DIM NM AS string

Dokument=ThisComponent

Vymena=Dokument.createReplaceDescriptor()

NM=chr\$(&HA0) ' Nezalomiteľná medzera (ctrl+space)

' Budeme hľadať pomocou regulárnych výrazov

Vymena.SearchRegularExpression=True

'Hľadáme viacnásobné zalomiteľné medzery

Vymena.SearchString=" +"

'A nahradíme ich jednoduchou medzerou

Vymena.ReplaceString=" "

kolko=Dokument.replaceAll(Vymena)

Vymena.SearchRegularExpression=True

'Hľadáme kombinácie zalomiteľná+nezalomiteľná+zalomiteľná medzera,

'nezalomiteľná+zalomiteľná a zalomiteľná+nezalomiteľná medzera

Vymena.SearchString=" "+NM+" |"+NM+" | "+NM

```
' ktoré nahradíme nezalomitelnou medzerou
Vymena.ReplaceString=NM
kolko=kolko+Dokument.replaceAll(Vymena)
```

```
'Hľadáme kombinácie medzera+tabelátor+medzera,
'tabelátor+medzera a medzera+tabelátor
Vymena.SearchString=" \t |\t | \t"
' ktoré nahradíme tabelátorom
Vymena.ReplaceString="\t"
kolko=kolko+Dokument.replaceAll(Vymena)
```

```
Vymena.SearchRegularExpression=True
'Hľadáme zbytočné medzery na začiatku alebo na konci odstavca
Vymena.SearchString="^[[:space:]]*|[:space:]*$"
' ktoré vymažeme
Vymena.ReplaceString=""
kolko=kolko+Dokument.replaceAll(Vymena)
```

```
Vymena.SearchRegularExpression=True
'Hľadáme zbytočné tabelátory na začiatku alebo na konci odstavca
Vymena.SearchString="^\t*|\t*$"
' ktoré vymažeme
Vymena.ReplaceString=""
kolko=kolko+Dokument.replaceAll(Vymena)
```

```
msgbox("Nahradených "+kolko+" viacnásobných medzier.",0,"Viacnásobne medzery")
END SUB
```

Funkcie a procedúry

V minulom dieli sme si ukázali makro pre vymazávanie viacnásobných medzier, kde sme pridali aj vymazávanie nepotrebných znakov na začiatku či konci odstavca a pod. Pokiaľ si dobre všimnete toto makro, zistíte, že sa tam viac krát opakuje vlastne tá istá postupnosť týchto príkazov:

```
Vymena.SearchRegularExpression=True
Vymena.SearchString= ... ' hľadaný reťazec
Vymena.ReplaceString= ... ' reťazec, ktorým nahrádzame
kolko=Dokument.replaceAll(Vymena)
```

V prípade ďalších zámen (ktoré budeme ešte programovať) sa takýto program zbytočne predlžuje a stráca prehľadnosť. Preto je vhodné, aby sme často opakované postupnosti rovnakých príkazov zadefinovali osobitne ako funkciu alebo procedúru, ktoré spoločným názvom nazývame aj podprogramy. Procedúru už vlastne používame, pretože

aj už naprogramované makrá pre výmenu viacnásobných medzier sú vlastne procedúry. Je to vlastne akoby definícia nového príkazu. Samotná definícia má takýto obecný tvar:

```
SUB názov(zoznam parametrov)
```

```
...
```

```
END SUB
```

Funkcie nám tiež nie sú neznáme, pretože poznáme napr. matematické funkcie a pod. V našom makre pre výmenu viacnásobných medzier sme používali napr. systémovú funkciu Dokument.replaceAll. Samotná definícia má takýto obecný tvar:

```
FUNCTION názov(zoznam parametrov) AS typ výsledku
```

```
...
```

```
názov=výsledok
```

```
END FUNCTION
```

Ako vidíme, vo funkcii musíme mať aspoň jeden špeciálny priradovací príkaz, ktorým priradíme jej výsledok do názvu funkcie. Dosť však bolo teórie, pozrime sa na naše makro pre výmenu viacnásobných medzier. Uvedené štyri riadky by sme mohli naprogramovať potom takto:

```
FUNCTION Vymen(Co, Za as string, Regularne as Boolean) as long
```

```
rem Funkcia vymení reťazec Co za reťazec Za a vráti počet výmien
```

```
rem Regularne - určuje, či vyhľadávaný výraz je (TRUE) alebo nie je (FALSE)  
regulárny výraz
```

```
rem Ako výsledok bude počet zámen
```

```
Vymena.SearchString=Co
```

```
Vymena.ReplaceString=Za
```

```
Vymena.SearchRegularExpression=Regularne
```

```
Vymen=Dokument.replaceAll(Vymena) ' Do názvu funkcie priradujeme počet výmen
```

```
END FUNCTION
```

Vlastnú procedúru pre zámenu môžeme teraz naprogramovať takto:

```
SUB Viacnasobna_medzera
```

```
DIM kolko AS Long
```

```
DIM NM AS string
```

```
NM=chr$(&HA0) ' Nezalomitel'ná medzera (ctrl+space)
```

```
' Výmena viacnásobných medzier za jednoduchú
```

```
kolko=Vymen(" +", " ", TRUE)
```

```
' Kombinácie nezalomitel'ných a zalomitel'ných medzier
```

```
kolko=kolko+Vymen(" "+NM+" |"+NM+" | "+NM, NM, TRUE)
```

```
' Kombinácie tabelátorov a medzier
```

```
kolko=kolko+Vymen(" \t |\t | \t", "\t", TRUE)
```

```
' Medzery na začiatku a konci odstavca
kolko=kolko+Vymen("^[:space:]*[:space:]*$", "", TRUE)
' Tabelařory na začiatku a konci odstavca
kolko=kolko+Vymen("^\t*|\t*$", "", TRUE)
```

```
msgbox("Nahradených "+kolko+" viacnásobných medzier.",0,"Viacnásobne medzery")
```

END SUB

Ako vidíme, makro sa naozaj zjednodušilo a obsahuje okrem poznámok presne deväť riadkov. Pravdaže, dá sa zapísať ešte kratšie – na päť riadkov s použitím jedného jediného priradovacieho príkazu, ale to je už trochu na úkor čitateľnosti a prehľadnosti:

```
SUB Viacnasobna_medzera
```

```
DIM kolko AS Long
```

```
DIM NM AS string
```

```
NM=chr$(&HA0) ' Nezalomitel'ná medzera (ctrl+space)
```

```
kolko=Vymen(" +", " ", TRUE)+Vymen(" "+NM+" |"+NM+" |
"+NM,NM,TRUE)+Vymen(" \t |\t | \t", "\t", TRUE)+Vymen("^[:space:]*|
[:space:]*$", "", TRUE)+Vymen("^\t*|\t*$", "", TRUE)
```

```
msgbox("Nahradených "+kolko+" viacnásobných medzier.",0,"Viacnásobne medzery")
```

END SUB

Alebo pre absolutistov ešte kratšie na tri riadky:

```
SUB Viacnasobna_medzera
```

```
DIM NM AS string
```

```
NM=chr$(&HA0) ' Nezalomitel'ná medzera (ctrl+space)
```

```
msgbox("Nahradených "+Vymen(" +", " ", TRUE)+Vymen(" "+NM+" |"+NM+" |
"+NM,NM,TRUE)+Vymen(" \t |\t | \t", "\t", TRUE)+Vymen("^[:space:]*|
[:space:]*$", "", TRUE)+Vymen("^\t*|\t*$", "", TRUE)+" viacnásobných
medzier.",0,"Viacnásobne medzery")
```

END SUB

Alebo ak chcete, tak absolútne na jeden riadok:

```
SUB Viacnasobna_medzera
```

```
msgbox("Nahradených "+Vymen(" +", " ", TRUE)+Vymen(" "+chr$(&HA0)+"
|"+chr$(&HA0)+" | "+chr$(&HA0),chr$(&HA0),TRUE)+Vymen(" \t |\t |
\t", "\t", TRUE)+Vymen("^[:space:]*|[:space:]*$", "", TRUE)+Vymen("^\t*|
\t*$", "", TRUE)+" viacnásobných medzier.",0,"Viacnásobne medzery")
```

END SUB

Posledné úpravy však nevedú (ak vôbec vedú) k zrýchleniu podprogramu a sú preto určené iba ako ukážka možností volania funkcií s prípadným šetrením pamäte, ktoré sme získali definíciou menšieho (alebo nulového) počtu premenných.

Ak sa pozrieme na zadefinované podprogramy zistíme, že sme kdesi „stratili“ definíciu premennej Vymena a vlastne nikde nepriradujeme výmenu k aktuálnemu dokumentu. Niečo nám tam teda ešte chýba. Najprv si však povedzme, ako je to vôbec s premennými – ktoré poznáme iba vo vnútri podprogramu a ktoré aj mimo? Je to jednoduché – všetko záleží od miesta, kde ich definujeme. Ak zadefinujeme premenné iba vo vnútri podprogramu, potom ich poznáme iba tam – môžeme ich nazvať lokálne premenné. Ak ich zadefinujeme „vonku“, poznáme ich všade – môžeme ich nazvať ako globálne, alebo spoločné premenné.

Takže v tomto zmysle teraz zadefinujeme chýbajúce premenné a následne priradíme aktuálny dokument. Celé makro zo všetkými funkciami a procedúrami potom môžeme naprogramovať napríklad takto:

REM Definície spoločných premenných sú úplne pred definíciou podprogramov

dim Dokument, Vymena as object

' Nezalomitelnú medzeru budeme používať aj v iných makrách, preto presunieme definíciu sem

dim NM as string

' V osobitnej procedúre inicializujeme spoločné premenné.

' Túto procedúru musíme potom volať ako prvú, inak neurobíme potrebné priradenia.

sub Init

rem Inicializácia spoločných premenných

Dokument=ThisComponent

Vymena=Dokument.createReplaceDescriptor()

NM=chr\$(&HA0) ' Nezalomitelná medzera – hexadecimálny kód A0

end sub

FUNCTION Vymen(Co, Za as string, Regularne as Boolean) as long

rem Funkcia vymení reťazec Co za reťazec Za a vráti počet výmien

**rem Regularne - určuje, či vyhľadávaný výraz je (TRUE) alebo nie je (FALSE)
regulárny výraz**

rem Ako výsledok bude počet zámen

Vymena.SearchString=Co

Vymena.ReplaceString=Za

Vymena.SearchRegularExpression=Regularne

Vymen=Dokument.replaceAll(Vymena) ' Do názvu funkcie priradujeme počet výmen

END FUNCTION

SUB Viacnasobna_medzera

DIM kolko AS Long

' Inicializácia spoločných premenných

Init

' Výmena viacnásobných medzier za jednoduchú

kolko=Vymen(" +", " ",TRUE)

' Kombinácie nezalomiteľných a zalomiteľných medzier

kolko=kolko+Vymen(" "+NM+" |"+NM+" | "+NM,NM,TRUE)

' Kombinácie tabelátorov a medzier

kolko=kolko+Vymen(" \t |\t | \t", "\t",TRUE)

' Medzery na začiatku a konci odstavca

kolko=kolko+Vymen("^[:space:]*[[:space:]]*\$", "",TRUE)

' Tabelátory na začiatku a konci odstavca

kolko=kolko+Vymen("^\t*|\t*\$", "",TRUE)

msgbox("Nahradených "+kolko+" viacnásobných medzier.",0,"Viacnásobne medzery")

END SUB

Toto je minule sľúbené makro pre výmenu viacnásobných medzier na osem riadkov. A hoci to tak nevyzerá, ešte stále to nie je jeho posledná verzia. Napokon, stále platia staré programátorské zákony – v každom programe je aspoň jedna chyba a v každom programe je aspoň jeden riadok zbytočný (len to neberte absolútne, lebo potom by sme nemali čo programovať :-).

Nezalomiteľná medzera

V dnešnom dieli nášho seriálu začneme rozširovať funkcie, ktorými budeme formátovať dokument. A začneme hneď jedným z najviac žiadaných makier, pomocou ktorého budeme podľa typografických pravidiel zamieňať zalomiteľné medzery za nezalomiteľné.

V minulom dieli sme si naprogramovali funkciu „Vymen“ a procedúru „Init“, pomocou ktorých sme následne značne zjednodušili naše makro pre opravu viacnásobných medzier. V nasledujúcich dieloch nášho seriálu tieto podprogramy už budeme štandardne využívať, takže sa k nim budeme vracieť iba v nevyhnutných prípadoch, keď ich budeme meniť – teda napríklad aj v dnešnom dieli.

Nezalomiteľné medzery podľa typografických pravidiel

Pozrime sa teraz bližšie na ďalší problém, z ktorým sa pri praktickej práci stretávame. Je to ten fakt, že OpenOffice.org nevkladá automaticky nezalomiteľné medzery napr. za jednoznakové predložky. Toto však odporuje zásadám typografie, v ktorých sa o.i. hovorí, že k zalomeniu riadku nemá dôjsť pred jednociferným číslom bez bernej jednotky (napríklad „1 človek“), medzi rádovými skupinami čísiel vo vnútri čísla alebo pri telefónnych číslach (napr. „1 587 235“), medzi číslom a značkou mernej jednotky (napr. „123 km“), medzi skratkami akademických titulov, vojenskými a vedeckými hodnosťami a menom (napr. „Ing. Pastierik“), medzi skratkou krstného mena a priezviskom (napr.

„J. Pastierik“), za číslicami v dátumoch (napr. „16. 1. 2006“) a ani za jednoznakovou predložkou alebo spojkou.

V prípade predložiek a spojok je otázka nezalomiteľných medzier trochu nejednoznačná. Ide o to, že niektoré zdroje (napr. <http://www.typo.cz>) uvádzajú, že spojka „a“ (ale nie verzálka „A“) je výnimka a ďalej, že v prípade úzkej sadzby (do 25 znakov) sú možné výnimky aj za ostatnými jednoznakovými predložkami a spojkami. Iné pramene popisujú ako výnimku iba spojku „a“ a aj to iba v prípade úzkej sadzby. Pretože rozhodnutie leží na každom používateľovi, budeme sa snažiť vyriešiť otázku okolo spojky „a“ tak, aby si každý mohol urobiť jednoduchú úpravu podľa svojich požiadavok.

Nezalomiteľné medzery za spojkami

Ako vidíme, problematika nezalomiteľných medzier je pomerne široká a preto sa ňou budeme zaoberať postupne. Ako prvé, čo si budeme v spomínanej oblasti programovať, bude makro pre zámenu zalomiteľných medzier za jednoznakovými predložkami a spojkami, pretože toto je najčastejšie požadované od používateľov OpenOffice.org. Jednoduchá verzia tohto makra by mohla vyzeráť takto (predložky rozpoznáme tak, že pred nimi a za nimi je medzera):

```
sub nezalomitelne_Spojky
```

```
dim predlozky() as string
```

```
dim i, kolko as long
```

```
predlozky()=array("a","i","k","o","s","u","v","z")
```

```
kolko=0
```

```
REM pre všetky predložky
```

```
for i=lbound(predlozky()) to ubound(predlozky())
```

```
kolko=kolko+vymen(" "+predlozky(i)+" "," "+predlozky(i)+NM,false)
```

```
next i
```

```
msgbox("Nahradených "+kolko+" zalomiteľných medzier.",0,"Nezalomiteľné medzery")
```

```
end sub
```

Podobne, ako v prípade prvej verzie makra pre mazanie viacnásobných medzier, aj toto makro má veľa nedostatkov. Ten najhlavnejší je však ten, že ak máme spojku napísanú veľkými písmenami, tak ju zamení za malé písmeno. Je jasné, že pri zámene musíme rozpoznávať veľké a malé písmena. Na tento účel si preto najprv musíme upraviť funkciu „Vymen“. Aby sme jej úpravu nemuseli robiť tak často, hneď si uvedieme aj ďalší parameter, pomocou ktorého budeme rozpoznávať, či hľadáme celé slová, alebo iba ich časti:

```
function Vymen(Co, Za as string, Regularne, Cele_slova, Velke_pismena as Boolean) as long
```

```
rem Funkcia vymení reťazec Co za reťazec Za a vráti počet výmien
```

```
rem Regularne - určuje, či vyhľadávaný výraz je (true) alebo nie je (false) regulárny výraz
```

```
rem Cele_slova - určuje, či vyhľadávaný výraz je (true) alebo nie je (false) celé slovo
```

rem Velke_pismena - určuje, či sa majú (true) alebo nemajú (false) rozoznávať veľké a malé písmená

```
Vymena.SearchString=Co
Vymena.ReplaceString=Za
Vymena.SearchRegularExpression=Regularne
Vymena.SearchWords=Cele_slova
Vymena.SearchCaseSensitive=Velke_pismena
Vymen=Dokument.replaceAll(Vymena)
end function
```

Pravdaže, teraz si musíme upraviť aj makro pre zámenu viacnásobných medzier, pretože sme tam nepočítali s novými parametrami:

```
SUB Viacnasobna_medzera
```

```
    DIM kolko AS Long
```

```
    ' Inicializácia spoločných premenných
```

```
    Init
```

```
    ' Výmena viacnásobných medzier za jednoduchú
```

```
    kolko=Vymen(" +", " ", TRUE, false, false)
```

```
    ' Kombinácie nezalomiteľných a zalomiteľných medzier
```

```
    kolko=kolko+Vymen(" "+NM+" |"+NM+" | "+NM,NM,TRUE,false,false)
```

```
    ' Kombinácie tabelátorov a medzier
```

```
    kolko=kolko+Vymen(" \t |\t | \t", "\t", TRUE, false, false)
```

```
    ' Medzery na začiatku a konci odstavca
```

```
    kolko=kolko+Vymen("^[:space:]*[[:space:]]*$", "", TRUE, false, false)
```

```
    ' Tabelátory na začiatku a konci odstavca
```

```
    kolko=kolko+Vymen("^\t*|\t*$", "", TRUE, false, false)
```

```
    MsgBox("Nahradených "+kolko+" viacnásobných medzier.", 0, "Viacnásobne medzery")
```

```
END SUB
```

Makro pre zámenu zalomiteľných medzier za predložkami a spojkami by teraz mohlo vyzeráť takto:

```
sub nezalomiteľne_Spojky
```

```
    dim predlozky() as string
```

```
    dim i, kolko as long
```

```

predlozky()=array("a","i","k","o","s","u","v","z")

kolko=0

REM pre všetky predložky
for i=lbound(predlozky()) to ubound(predlozky())
  rem malé predložky
  kolko=kolko+vymen(" "+predlozky(i)+" "," "+predlozky(i)+NM,false,false,TRUE)
  rem veľké predložky – funkcia uCase prevedie malé znaky na veľké
  kolko=kolko+vymen(" "+ucase(predlozky(i))+" "," "+ucase(predlozky(i))+NM,false,false,TRUE)
next i

  msgbox("Nahradených "+kolko+" zalomiteľných medzier.",0,"Nezalomiteľné medzery")
end sub

```

V tejto verzii makra sa však vkladá nezalomiteľná medzera aj za spojku „a“, čo nie každému vyhovuje. Tento stav sa dá ošetriť napríklad tak, že budeme túto výnimku testovať:

```

sub nezalomitelne_Spojky
  dim predlozky() as string
  dim i, kolko as long

  predlozky()=array("a","i","k","o","s","u","v","z")

  kolko=0

  REM pre všetky predložky
  for i=lbound(predlozky()) to ubound(predlozky())
    rem malé predložky
    if predlozky(i)<>"a" then
      kolko=kolko+vymen(" "+predlozky(i)+" "," "+predlozky(i)+NM,false,false,TRUE)
    end if
    rem veľké predložky – funkcia uCase prevedie malé znaky na veľké
    kolko=kolko+vymen(" "+ucase(predlozky(i))+" "," "+ucase(predlozky(i))+NM,false,false,TRUE)
  next i

  msgbox("Nahradených "+kolko+" zalomiteľných medzier.",0,"Nezalomiteľné medzery")
end sub

```

Alebo, ešte jednoduchšie, vymenovaním všetkých malých aj veľkých spojok a predložiek osobitne:

```
sub nezalomitelne_Spojky

dim predlozky() as string
dim i, kolko as long

predlozky()=array("A","i","I","k","K","o","O","s","S","u","U","v","V","z","Z")

kolko=0

REM pre všetky predložky
for i=lbound(predlozky()) to ubound(predlozky())
    kolko=kolko+vymen(" "+predlozky(i)+" "," "+predlozky(i)+NM,false,false,TRUE)
next i

msgbox("Nahradených "+kolko+" zalomiteľných medzier.",0,"Nezalomiteľné medzery")
end sub
```

Ten, kto chce vkladať aj za spojku „a“ nezalomiteľnú medzeru, si ju môže do zoznamu jednoducho doplniť. Ani toto riešenie nie je však úplne dobré, pretože niekto môže chcieť, aby sa za touto spojku nevkladala nezalomiteľná medzera iba v prípade úzkej sadzby. Toto však budeme riešiť až niekedy inokedy.

Znovu používame regulárne výrazy

Makro pre vkladanie nezalomiteľných medzier za predložky, ktoré sme si naprogramovali naposledy, má niekoľko nedostatkov. Prvý, ktorý si môžeme všimnúť, je ten, že nevkladá medzery za tie predložky, ktoré sa nachádzajú na začiatku odstavca. Pravdaže, tento nedostatok nemusí byť až taký závažný, pretože na tomto mieste nedochádza k nevhodnému zalomeniu. Na druhej strane, ak zlúčime dva odstavce do jedného, tento problém sa už vyskytnúť môže. Môžeme to však jednoducho vyriešiť pomocou regulárnych výrazov, napríklad takto:

```
sub nezalomitelne_Spojky

dim predlozky() as string
dim i, kolko as long

predlozky()=array("A","i","I","k","K","o","O","s","S","u","U","v","V","z","Z")

kolko=0

REM pre všetky predložky
for i=lbound(predlozky()) to ubound(predlozky())
```

```

kolko=kolko+vymen(" "+predlozky(i)+" "," "+predlozky(i)+NM,false,false,TRUE)
REM Vyhľadáme predložky aj na začiatku odstavca
kolko=kolko+vymen("^"+predlozky(i)+" ",predlozky(i)+NM,TRUE,false,TRUE)
next i

```

```

msgbox("Nahradených "+kolko+" zalomiteľných medzier.",0,"Nezalomiteľné
medzery")
end sub

```

Ale spojky a predložky nebývajú vždy osamotené

Pozrime sa však na druhý problém, ktorý nám pri používaní tohto makra vznikol – v prípade, že máme v texte viacej jednoznakových spojok a predložiek tesne za sebou, nezamení všetky potrebné medzery. Takéto prípady sú pomerne časté a nie je problém sa stretnúť aj s tromi predložkami a spojkami za sebou, napr. v texte „... a i s tým sa musíme vyrovnáť ...“. Je to vlastne preto, lebo konkrétne v uvedenom príklade sa za spojku „a“ vloží nezalomiteľná medzera a preto v ďalšom kroku pri teste spojky „i“ túto už nenájde. V makre totiž hľadáme reťazec zalomiteľná medzera + spojka + zalomiteľná medzera a my tam v skutočnosti máme reťazec nezalomiteľná medzera + spojka + zalomiteľná medzera. Naše makro preto rozšírime o ďalší test:

```

sub nezalomitelne_Spojky

```

```

dim predlozky() as string
dim i, kolko as long

```

```

predlozky()=array("A","I","I","K","K","O","O","S","S","u","U","v","V","z","Z")

```

```

kolko=0

```

```

REM pre všetky predložky

```

```

for i=lbound(predlozky()) to ubound(predlozky())

```

```

kolko=kolko+vymen(" "+predlozky(i)+" "," "+predlozky(i)+NM,false,false,TRUE)

```

```

REM pred predložkou môže byť aj nezalomiteľná medzera

```

```

kolko=kolko+vymen(NM+predlozky(i)+" ",NM+predlozky(i)+NM,false,false,TRUE)

```

```

REM Vyhľadáme predložky aj na začiatku odstavca

```

```

kolko=kolko+vymen("^"+predlozky(i)+" ",predlozky(i)+NM,TRUE,false,TRUE)

```

```

next i

```

```

msgbox("Nahradených "+kolko+" zalomiteľných medzier.",0,"Nezalomiteľné
medzery")
end sub

```

Toto makro už funguje správne. Ako však vidíme, každú jednoznakovú spojku alebo predložku testujeme až tri krát. Je to tak preto, lebo sme si definovali jednoznakové spojky a predložky ako reťazec medzera + písmeno + medzera. Predložky však môžeme

definovať aj inak – sú to vlastne jednoznakové slová, t.j. hľadané písmeno sa musí zároveň nachádzať na začiatku slova. Na rad teda znovu prichádzajú regulárne výrazy:

```
sub nezalomitelne_Spojky
```

```
dim predlozky() as string
```

```
dim i, kolko as long
```

```
predlozky()=array("A","I","K","O","S","U","V","Z")
```

```
kolko=0
```

```
REM pre všetky predložky
```

```
for i=lbound(predlozky()) to ubound(predlozky())
```

```
REM hľadáme jednoznakové spojky, ktoré sú na začiatku slova
```

```
kolko=kolko+vymen("\<" + predlozky(i) + " " + predlozky(i) + NM, TRUE, false, TRUE)
```

```
next i
```

```
msgbox("Nahradených " + kolko + " zalomiteľných medzier.", 0, "Nezalomiteľné medzery")
```

```
end sub
```

Toto makro rieši v jednom teste všetky problémy, ktoré sme dnes načrtli, pretože jednoznakové predložky a spojky, ktoré sa nachádzajú na začiatku odstavca sa zároveň nachádzajú aj na začiatku slova. Zároveň sa týmto makrom rieši aj ten problém, keď pred jednoznakovou spojku a predložkou nie je medzera, ale tabulátor.

Ďalšie formátovacie funkcie

V doterajších dieloch nášho seriálu sme sa venovali dvom základným problémom – vymazávaniu viacnásobných medzier a vkladaniu nezalomiteľných medzier za jednoznakové predložky a spojky. Pravdaže, týmito príkladmi sme nevyriešili všetky problémy, ktoré sú spojené s touto problematikou. Napríklad, pod odstraňovanie viacnásobných medzier by sme mohli zaradiť aj odstraňovanie zbytočných medzier, resp. takých medzier, ktoré nezodpovedajú typografickým pravidlám. Tu patria napr. nesprávne vložené medzery medzi úvodzovky a text, zátvorky a text a pod. Podobne sme ešte nevyriešili celú problematiku okolo nezalomiteľných medzier, pretože tie sa vkladajú (ako sme už spomínali v tretej časti) aj medzi skratkami akademických titulov, za číslicami v dátumoch a pod. A práve tejto problematike sa teraz budeme postupne venovať.

„Nezalomiteľné“ akademické tituly

Ako prvej sa budeme teraz venovať problematike vkladania nezalomiteľných medzier medzi akademické tituly a meno. Hoci je problematika podobná jednoznakovým predložkám, má svoje osobitosti, ktoré si musíme rozobrať. Na rozdiel od predložiek má väčšina titulov pevný tvar veľkých a malých písmen (napr. magister farmácie – „PhMr.“). Nemusíme sa však zaoberať tým, ako boli napísané, ale ich vždy prepíšeme správnym tvarom, čím zároveň dosiahneme aj to, že automaticky opravíme ich prípadný nesprávny zápis. Toto však neplatí úplne pre všetky tituly, pretože skratky titulov akademik – „akad.“, architekt – „arch.“, docent – „doc.“ a profesor – „prof.“ sa píše s malými

písmenami, pravdaže, okrem prípadu, keď sa nachádzajú na začiatku vety (vtedy je prvé písmeno veľké).

Okrem titulov, ktoré sa uvádzajú pred menom existujú aj tituly, ktoré sa uvádzajú za menom, t.j. pri ktorých musíme vkladať nezalomiteľnú medzeru pred ne (napr. kandidát vied – „CSc.“). Okrem toho, pri tituloch za menom existuje možný dvojtvar pre skratku titulu čestného doktorátu (honoris causa) – h.c. (bez medzery)¹ alebo h. c. (s nezalomiteľnou medzerou).

Pri programovaní už nebudeme uvádzať riešenia, ktoré nehľadajú potrebné reťazce na začiatku alebo konci odstavca, pretože sme si túto problematiku už dosť objasnili v predchádzajúcich dieloch nášho seriálu. Pri tituloch sa však stretávame s novým problémom – vždy končia bodkou, ktorú musíme zahrnúť do vyhľadávania. Vzhľadom na to, že budeme hľadať pomocou regulárnych výrazov, kde má bodka špecifické postavenie (nahrádza akýkoľvek znak), musíme ju zadávať kombináciu znakov „\.“ a z tohto dôvodu ju nebudeme v definícii titulov uvádzať.

V tejto súvislosti nám ešte spôsobia problémy skratky „h. c.“ a „h.c.“, pretože tu sa bodka nachádza aj v ich vnútri, a preto ich musíme riešiť úplne osobitne. Makro, kde sú zahrnuté všetky uvedené problémy, by mohlo vyzeráť potom takto:

sub Nezalomitelne_tituly

REM program prevedie zámenu zalomiteľnej medzery medzi menom a titulom

REM zároveň sa opraví nesprávne napísané tituly okrem titulov

REM akad., arch., doc., prof., ktoré sa píše malým písmenom,

REM ale na začiatku vety majú prvé písmeno veľké, t.j. Akad., Arch., Doc., Prof.

dim Tituly_pred(), Tituly_pred_spec(), Tituly_za() as string

dim i, kolko as long

REM Akademické tituly pred menom

Tituly_pred()=array("Bc", "Dr", "Ing", "JUDr", "Mgr", "MUDr", "MVDr", "PaedDr",
"PharmDr", "PhDr", "PhMr", "RNDr", "ThDr")

REM Akademické tituly pred menom, kde sú rôzne prvé písmená

Tituly_pred_spec()=array("akad", "Akad", "arch", "Arch", "doc", "Doc", "prof", "Prof")

REM Akademické tituly za menom

Tituly_za()=array("CSc","DrSc")

REM Mimo ostatných skratiek je skratka h.c.

REM Najprv vložíme nezalomiteľnú medzeru do skratiek h. c.

Kolko=Vymen("h. c.", "h." + NM + "c.", false, false, false)

REM nezalomiteľná medzera pred skratku h.c. bez medzier

Kolko=Vymen(" h.c.", NM + "h.c.", false, false, false)

REM nezalomiteľná medzera pred skratku h. c. s medzerou

Kolko=Vymen(" h." + NM + "c.", NM + "h." + NM + "c.", false, false, false)

1 Tvar bez medzery „h.c.“ nie je správny, ale o tom píšeme neskôr (str. 23).

```

REM Pre všetky tituly pre menom, ktoré sa zároveň opravujú na správny tvar
for i=lbound(Tituly_pred()) to ubound(Tituly_pred())
    kolko=kolko+Vymen("\<" + Tituly_pred(i) + "\. ", Tituly_pred(i) + "." + NM, true, false, false)
)
next i

REM Pre všetky tituly pred menom, ktoré sa neopravujú
for i=lbound(Tituly_pred_spec()) to ubound(Tituly_pred_spec())
    kolko=kolko+Vymen("\<" + Tituly_pred_spec(i) + "\. ", Tituly_pred_spec(i) + "." + NM, true, false, true)
next i

REM Pre všetky tituly, ktoré sú za menom
for i=lbound(Tituly_za()) to ubound(Tituly_za())
    kolko=kolko+Vymen(" " + Tituly_za(i) + ".", NM + Tituly_za(i), false, false, false)
next i

msgbox("Nahradených " + kolko + " zalomiteľných medzier.", 0, "Nezalomiteľné tituly")
end sub

```

Ako vidíme, toto makro je na rozdiel od nezalomiteľných medzier za predložkami pomerne zložitá, no napriek tomu dúfame, že ešte stále pochopiteľná a prehľadná. Vieme, že v ňom nie sú uvedené všetky možné tituly (napr. chýbajú vojenské hodnosti), predpokladáme, že tí, ktorí s nimi v texte pracujú si ich určite ľahko doplnia. Okrem toho sa môže do makra doplniť aj výmena dvoch rôznych tvarov titulov čestného doktorátu za jeden – ten, ktorý sa komu viac páči. Toto však necháme prípadným záujemcom ako domácu úlohu.

Vlastná funkcia pre výmeny

V minulom dieli sme uverejnili makro pre vkladanie nezalomiteľných medzier za akademické tituly. Dnes tieto možnosti rozšírime o ďalšie typografické pravidlá, pričom si naprogramujeme vlastnú funkciu pre zamieňanie nájdeného textu.

„Nezalomiteľné“ dátumy

Ďalším problémom, s ktorým sa pri dodržiavaní typografických pravidiel môžeme stretnúť. Je vkladanie nezalomiteľných medzier za číslice v dátumoch. Najjednoduchšie je tento problém vyriešiť tak, že automaticky vložíme nezalomiteľnú medzeru za každé číslo, ktoré končí znakom bodky:

```

sub Nezalomitelne_datumy

    dim Cislovky() as string
    dim i, kolko as long

    Cislovky()=array("1.", "2.", "3.", "4.", "5.", "6.", "7.", "8.", "9.", "0.")

```



```

kolko=0
for i=lbound(Cislovky()) to ubound (Cislovky())
    kolko=kolko+Vymen(Cislovky(i)+" ",Cislovky(i)+NM,false,false,false)
next i
msgbox("Nahradených "+kolko+" zalomiteľných medzier.",0,"Nezalomiteľné dátumy")
end sub

```

Pravdaže, toto riešenie nie je optimálne, pretože vloží nezalomiteľnú medzeru za úplne každé radové číslo bez ohľadu na to, kde sa vyskytuje. Pokiaľ ho teda budeme mať napr. na konci vety (napr. „ ... a tento pretekár skončil 123.“, úplne zbytočne nám tu vloží nezalomiteľnú medzeru. Tento problém by sme mohli vyriešiť tak, že budeme postupne testovať všetky dvojice číslíc „1. 1“, „1. 2“, ... „0. 0“. napr. takto:

```

sub Nezalomitelne_datumy

    dim Cislovky() as string
    dim i, j, kolko as long

    Cislovky()=array("1","2","3","4","5","6","7","8","9","0")

    kolko=0
    for i=lbound(Cislovky()) to ubound (Cislovky())
        for j=lbound(Cislovky()) to ubound (Cislovky())
            kolko=kolko+Vymen(Cislovky(i)+".
+Cislovky(j),Cislovky(i)+". "+NM+Cislovky(j),false,false,false)
        next j
    next i
    msgbox("Nahradených "+kolko+" zalomiteľných medzier.",0,"Nezalomiteľné dátumy")
end sub

```

Toto riešenie je však veľmi pomalé, pretože celý dokument prehladáme presne sto krát, kým neotestujeme a neopravíme všetky kombinácie a preto je v praxi viac-menej nepoužiteľné. Pravdaže, pri vyhľadávaní by sme mohli použiť regulárne výrazy, napr. „[0-9]\. [0-9]“, pomocou ktorých dokážeme vyhľadať všetky potrebné kombinácie na jeden krok. Problém je však v tom, ako zabezpečiť výmenu medzery, ktorá sa nachádza niekde uprostred hľadaného výrazu, pričom ani nevieme, ktorú konkrétnu dvojicu číslíc takto nájdeme.

Toto sa dá riešiť tak, že keď nájdeme niektorú dvojicu dátumových číslíc, následne v nej pomocou funkcií pre prácu s reťazcami opravíme potrebnú medzeru a až potom takto pripraveným reťazcom nahradíme pôvodný. Na to nám však nebude stačiť štandardná metóda pre zámenu, ktorú sme doteraz používali:

```
Vymena=Dokument.createReplaceDescriptor()
```

ale najprv musíme vyhľadať potrebný reťazec metódou pre hľadanie:

```
Hladaj=Dokument.createSearchDescriptor()
```

Preto sa teraz musíme vrátiť k definícii premenných a procedúre „Init“, aby sme tam doplnili požadované údaje:

```
REM Definície spoločných premenných sú úplne pred definíciou podprogramov
dim Dokument, Vymena, Hladaj as object
dim NM as string
```

```
sub Init
```

```
Dokument=ThisComponent
Vymena=Dokument.createReplaceDescriptor()
Hladaj=Dokument.createSearchDescriptor()
NM=chr$(&HA0) ' Nezalomitel'ná medzera – hexadecimálny kód A0
```

```
end sub
```

Po nájdení budeme vo vnútri dátumu vymieňať zalomitel'nú medzeru za nezalomitel'nú. Na takéto zámény si zdefinujeme obecnú funkciu, pomocou ktorej dokážeme zameniť ľubovoľný vnútorný reťazec za iný, pretože v takomto tvare sa nám bude hodiť aj pre iné účely:

```
function Zamen_Vo_Vnutri (V_Com, Co, Za as string) as string
REM Funkcia vyhľadá v reťazci V_Com reťazec Co a nahradí ho reťazcom Za.
REM Ak tento reťazec nenájde, ponechá vo výsledku pôvodný reťazec V_Com.

dim pom as string
dim i, j as long

REM Pre prípad, že sa reťazec nenájde, necháme vo výsledku pôvodný reťazec
pom=V_Com

REM pozícia hľadaného reťazca
i=instr(V_Com,Co)

REM veľkosť zvyšku pravej strany pôvodného reťazca, ktorá je za hľadaným reťazcom
j=len(V_Com)-i-len(Co)+1

REM ak sa reťazec Co našiel, je jeho pozícia väčšia ako 0
if i>0 then
REM do výsledku priradíme ľavú časť pôvodného reťazca plus nahrádzaný reťazec
pom=left(V_Com,i-1)+Za

REM ak je za hľadaným reťazcom ešte nejaký zvyšok pôvodného reťazca
if j>0 then
```

```
REM do výsledku priradíme aj tento zvyšok
pom=pom+right(V_Com,j)
end if
end if
```

```
Zamen_Vo_Vnutri=pom
end function
```

Teraz už môžeme zdefinovať vlastnú funkciu na výmenu, kde najprv nájdeme hľadaný výraz, potom v ňom zameníme vnútorný reťazec za iný a nakoniec upraveným reťazcom nahradíme pôvodný. Pravdaže, pri hľadaní využijeme možnosť nastavenia hľadania pomocou regulárnych výrazov, celých slov a rozlišovania veľkosti písmen. Ako výsledok tejto funkcie bude počet prevedených zámien.

```
function Vymen_hladanim(V_Com, Co, Za as string, Regularne, Cele_slova,
Velke_pismena as Boolean) as long
```

REM procedúra vyhľadá reťazec V_Com, pričom až v skutočne nájdenom reťazci zamení vnútorný reťazec Co za reťazec Za.

```
dim nasiel as object
dim kolko as long
```

```
Hladaj.searchString=V_Com
Hladaj.SearchRegularExpression=Regularne
Hladaj.SearchWords=Cele_slova
Hladaj.SearchCaseSensitive=Velke_pismena
```

```
kolko=0
```

```
REM objekt s nájdeným reťazcom priradíme do premennej Nasiel
nasiel = Dokument.findFirst(Hladaj)
```

```
rem opakuj, pokiaľ tento objekt obsahuje nejaké údaje
Do While NOT isNull(nasiel)
```

```
rem ak sa našiel reťazec
if len(nasiel.String)>0 then
rem ak sa našiel reťazec, tak urobíme zámenu a výsledkom nahradíme pôvodný
reťazec
nasiel.String=Zamen_Vo_Vnutri(nasiel.String,Co,Za)
kolko=kolko+1
end if
```

```
rem vyhľadáme ďalší reťazec
nasiel = Dokument.findNext( nasiel.End, Hladaj)
Loop
```

```
Vymen_hladanim=kolko
```

```
end function
```

Procedúra pre výmenu nezalomiteľných medzier vo vnútri dátumov potom môže vyzerať takto:

```
sub Nezalomitelne_datумы
```

```
dim kolko as long
```

```
kolko=Vymen_hladanim("[0-9]\. [0-9]", " ",NM,true,false,false)
```

```
msgbox("Nahradených "+kolko+" zalomiteľných medzier.",0,"Nezalomiteľné dátумы")
```

```
end sub
```

V prípade, že máme v texte uvedený dátum s údajom dňa, mesiaca a roku nedokáže toto makro vložiť nezalomiteľnú medzeru medzi číselný údaj mesiaca a roku. Toto však môžeme riešiť opakovaným vyhľadávaním:

```
sub Nezalomitelne_datумы
```

```
dim kolko as long
```

```
kolko=Vymen_hladanim("[0-9]\. [0-9]", " ",NM,true,false,false)
```

```
kolko=kolko+Vymen_hladanim("[0-9]\. [0-9]", " ",NM,true,false,false)
```

```
msgbox("Nahradených "+kolko+" zalomiteľných medzier.",0,"Nezalomiteľné dátумы")
```

```
end sub
```

Končíme s nezalomiteľnými medzarami

V dnešnom dieli dokončíme množinu funkcií pre vkladanie nezalomiteľných medzier, čím, pochopiteľne, ešte stále nevyčerpáme celú túto problematiku. Rôzne iné možnosti však necháme ako domácu úlohu.

„Nezalomiteľné čísla“

Ako posledné funkcie, ktoré si v oblasti nezalomiteľných medzier naprogramujeme budú procedúry pre vkladanie týchto medzier za čísla. Pripomeňme si, že nezalomiteľné medzery by mali byť za jednociferným číslom bez bernej jednotky (napr. „9 ľudí“), medzi rádovými skupinami cifier vo vnútri čísla alebo pri telefónnych číslach (napr. „5 784 452“) a medzi číslom a značkou mernej jednotky (napr. „15 kg“). Úplne najjednoduchšie riešenie je také, že nezalomiteľnú medzeru vložíme za úplne každé číslo, napríklad takto:

```
sub Nezalomitelne_cisla
```

```
dim kolko as long
```

```
kolko=Vymen_hladanim("[0-9] "," ",NM,true,false,false)
```

```
msgbox("Nahradených "+kolko+" zalomitelných medzier.",0,"Nezalomitelné čísla")
```

```
end sub
```

Toto riešenie bude určite mnohým postačovať, no na druhej strane vkladá nezalomitelné medzery za čísla aj vtedy, keď to nie je potrebné, napr. v texte „12 stromov“. Preto musíme jednotlivé možnosti naprogramovať osobitne.

„Nezalomitelné jednociferné čísla“

Riešenie problematiky jednociferných čísiel je veľmi jednoduché, pretože za každým jednociferným číslom by mala nasledovať nezalomitelná medzera. Jednociferné číslo zistíme podobne, ako sme zisťovali predložky – t.j. číslo za ktorým je medzera sa musí nachádzať na začiatku slova:

```
sub Nezalomitelne_jednociferne_cisla
```

```
dim kolko as long
```

```
kolko=Vymen_hladanim("\<[0-9] "," ",NM,true,false,false)
```

```
msgbox("Nahradených "+kolko+" zalomitelných medzier.",0,"Nezalomitelné jednociferné čísla")
```

```
end sub
```

„Nezalomitelné telefónne čísla“

V prípade rádoých skupín cifier a telefónnych čísiel musíme zameniť medzeru, ktorá sa nachádza medzi dvomi číslami:

```
sub Nezalomitelne_telefonne_cisla
```

```
dim kolko as long
```

```
kolko=Vymen_hladanim("[0-9] [0-9]"," ",NM,true,false,false)
```

```
msgbox("Nahradených "+kolko+" zalomitelných medzier.",0,"Nezalomitelné telefónne čísla")
```

```
end sub
```

„Nezalomitelné merné jednotky“

Na koniec sme si nechali problematiku nezalomitelných medzier, ktoré sa nachádzajú medzi číslom a značkou mernej jednotky. Aby sme mohli vyriešiť túto problematiku, musíme, podobne ako pri akademických tituloch vymenovať všetky merné jednotky, ktoré

sa za číslom môžu vyskytovať. Už v tomto však objavíme problém, pretože celkový počet rôznych merných jednotiek (vrátane ich kombinácií s násobkami „kilo“, „mega“, „piko“ ...) dosahuje rádovo stovky možností. Pravdaže, málokto v praxi používa skutočne všetky a preto, ak ich počet bude ešte únosný, môže použiť nasledovné makro, kde sme ako príklad použili jednotky dĺžky:

```
sub Nezalomitelne_merne_jednotky
```

```
dim Merne_jednotky() as string
```

```
dim i, kolko as long
```

```
Merne_jednotky()=array("m", "km", "mm", "cm")
```

```
for i=lbound(Merne_jednotky()) to ubound(Merne_jednotky())
```

```
kolko=kolko+Vymen_hladanim("[0-9] "+Merne_jednotky(i)," ",NM,true,false,false)
```

```
next i
```

```
msgbox("Nahradených "+kolko+" zalomiteľných medzier.",0,"Nezalomiteľné merné jednotky")
```

```
end sub
```

Počet merných jednotiek, pri ktorých bude uverejnené makro ešte dostatočne rýchle závisí najmä od veľkosti spracovávaných dokumentov a preto sa nedá jednoznačne určiť ich počet. Vzniká otázka, či by sa to predsa len nedalo vyriešiť inak, pretože testovanie rôznych možností jednotiek v rámci jedného cyklu je určite rýchlejšie, ako opakované prehľadávanie celého dokumentu. Našťastie, aj v tomto smere nám riešenie poskytujú priamo regulárne výrazy, kde máme možnosť zadať logickú podmienku „alebo“:

```
sub Nezalomitelne_merne_jednotky
```

```
dim kolko as long
```

```
kolko=kolko+Vymen_hladanim("[0-9] (m|km|mm|cm)", " ",NM,true,false,false)
```

```
msgbox("Nahradených "+kolko+" zalomiteľných medzier.",0,"Nezalomiteľné merné jednotky")
```

```
end sub
```

Toto riešenie môžeme použiť aj na iné funkcie v oblasti formátovania nezalomiteľných medzier. O tom si však budeme hovoriť až niekedy inokedy.

Nadbytočné medzery

Pokiaľ si pamätáte, v tretej časti sme uviedli zatiaľ poslednú verziu makra pre odstraňovanie viacnásobných medzier, v ktorej sme počítali aj s prázdnyimi medzerami na začiatkoch a koncoch odstavcov. Týmto sme, pochopiteľne, neopravili všetky možné výskyty nadbytočných medzier, pretože sa s nimi môžeme stretnúť aj v iných častiach dokumentu, napr. v textoch v zátvorkách, medzi úvodzovkami a pod.

Čo nám hovoria normy

Aby sme vedeli, ako správne písať medzery, pozrime sa na to, čo nám v tomto smere hovorí norma pre úpravu písomností (STN 01 6910, ktorá je prakticky zhodná s normou ČSN 01 6910). Tam sa o. i. dozvieme, že interpunkčné znamienka bodka, čiarka, dvojbodka, bodkočiarka, výkričník a otáznik sa pripájajú k textu bez medzery (napr. „A preto, hoci nerád, to urobil.“). Za tieto znaky patrí medzera okrem tých prípadov, keď ich nasleduje viac za sebou (napr. „bolo to vo firme FIRMA s r. o., takže ...“). Medzera sa nedáva ani za bodku v peňažných číslach (napr. „12.000 Sk“), za bodku pri číselnom označovaní časti textu (napr. „kapitola 2.7“), za čiarku pri oddelovaní desatinných miest (napr. „3,14“) a za dvojbodku pri písaní časových údajov (napr. „8:48 h“). Zátvorky a úvodzovky sa na začiatku a konci textu, ktorý dávame do zátvoriek alebo do úvodzoviek, neoddeľujú od tohto textu medzerami (ako môžeme vidieť vo všetkých príkladoch, ktoré sme v tomto odstavci pred chvíľou uviedli). Medzery sa nepoužívajú ani pri spojovníku (napr. „slovensko-nemecký slovník“).

Ako vidíme, aj táto problematika – podobne ako problematika nezalomiteľných medzier – je pomerne široká. V tejto súvislosti sa ešte musíme vrátiť k piatej časti tohto seriálu, kde som spomínal dva možné tvary akademického titulu čestného doktorátu „honoris causa“ – „h.c.“ (bez medzery) alebo „h. c.“ (s nezalomiteľnou medzerou). Tieto dva tvary sa uvádzali v materiáloch, z ktorých som čerpal podklady pre zoznam akademických titulov. Tvar bez medzery je však v rozpore so spomínanými normami pre úpravu písomností, čo som si vtedy neuvedomil. Týmto sa všetkým za uvedený omyl ospravedlňujem. Predmetný článok však už nebudem opravovať, pretože všetky uvedené makrá sú zároveň príkladmi riešenia určitého typu problematiky – v tomto prípade viacerých výnimiek z nejakého pravidla.

Nadbytočné medzery pri interpunkčných znamienkach

Ako prvé, čo si z uvedenej problematiky budeme programovať bude odstraňovanie nadbytočných medzier pred a za interpunkčnými znamienkami. Nebudeme sa už venovať pomalým náhradám cez cykly, ale využijeme rýchlejšie riešenie pomocou regulárnych výrazov.

sub Zmaz_interpunkcia_medzera

dim kolko as long

REM Vymazanie medzier za ľavými znakmi – úvodzovkami a zátvorkami

REM Tu patrí týchto šesť znakov: „ « " ([{

REM Úvodzovky " musíme zadať pomocou ich kódu, zátvorky (a [musíme zadať pomocou uvádzajúceho znaku \

```
kolko=Vymen_hladanim(",|«|"+chr$(&H22)+"|\(|\{|} " , " " , "" , true, false, false)
```

REM Vymazanie medzier pred pravými zátvorkami a úvodzovkami a pred interpunkciou

REM Tu patrí týchto šesť znakov: " » ")] }

REM a týchto šesť interpunkčných znamienok: , . ! ? ; :

REM Úvodzovky " musíme zadať pomocou ich kódu, znaky , . ?)] musíme zadať pomocou uvádzajúceho znaku \

```
kolko=Vymen_hladanim(" (|»|" +chr$(&H22)+"|\)|\}|\,|\.|!|\?|;|:)",  
",",true,false,false)
```

```
msgbox("Vymazaných "+kolko+" nadbytočných medzier.",0,"Nadbytočné medzery")
```

```
end sub
```

Chýbajúce medzery pri interpunkčných znamienkach

V minulom dieli sme si spomínali, že v normách pre úpravu písomností sa spomína, že okrem niekoľkých výnimiek sa za interpunkčné znamienka vkladá medzera. A práve na tento problém sa teraz zameriame. Aby sme ho mohli vyriešiť, znovu si pripomeňme, že za interpunkčné znamienka sa vkladá medzera okrem tých prípadov, keď ich nasleduje viac za sebou, medzera sa nevkladá ani za bodku oddeľujúcu rády v číslach, desatinnú čiarku a za dvojbodku v časových údajoch. Z hľadiska programovania to znamená, že medzeru budeme vkladať za diakritické znamienka vtedy, ak za nimi bezprostredne nasleduje písmeno a v prípade diakritických znamienok otáznika, výkričníka a bodkočiarky aj vtedy, ak za nimi bezprostredne nasleduje aj číslo.

Na rozdiel od predchádzajúcich makier nebudeme teraz nič mazať, ale iba vkladať. Preto si najprv musíme naprogramovať funkciu pre vloženie reťazca do nájdeneho textu:

```
function Vloz_hladanim(V_Com, Co as string, Kde as long, Regularne, Cele_slova,  
Velke_pismena as Boolean)
```

```
REM funkcia vyhledá reťazec V_Com a na pozíciu Kde do neho vloží reťazec Co
```

```
dim nasiel as object
```

```
dim kolko, i, j as long
```

```
dim pom as string
```

```
Hladaj.searchString=V_Com
```

```
Hladaj.SearchRegularExpression=Regularne
```

```
Hladaj.SearchWords=Cele_slova
```

```
Hladaj.SearchCaseSensitive=Velke_pismena
```

```
kolko=0
```

```
REM úprava parametra Kde (pozícia 0 - vloženie pred reťazec sa musí upraviť na 1)
```

```
i=kde+1
```

```
REM objekt s nájdeným reťazcom priradíme do premennej Nasiel
```

```
nasiel = Dokument.findFirst(Hladaj)
```

```
rem opakuj, pokiaľ tento objekt obsahuje nejaké údaje
```

```
Do While NOT isNull(nasiel)
```



```

rem ak sa našiel reťazec
if len(nasiel.String)>0 then
  REM veľkosť zvyšku pravej strany pôvodného reťazca
  j=len(nasiel.String)-i+1
  REM do výsledku priradíme ľavú časť pôvodného reťazca plus vkladací reťazec
  pom=left(nasiel.String,i-1)+Co
  REM ak je za hľadaným reťazcom ešte nejaký zvyšok pôvodného reťazca
  if j>0 then
    REM do výsledku priradíme aj tento zvyšok
    pom=pom+right(nasiel.String,j)
  end if
  nasiel.String=pom
  kolko=kolko+1
end if

```

```

rem vyhľadáme ďalší reťazec
nasiel = Dokument.findNext( nasiel.End, Hladaj)
Loop

```

```
Vloz_hladanim=kolko
```

```
end function
```

S využitím tejto funkcie môžeme potom naprogramovať makro pre vkladanie medzier napríklad takto:

```
sub Vloz_interpunkcia_medzera
```

```
dim kolko as long
```

```

REM a týchto šesť interpunkčných znamienok: , . ! ? ; :
REM Znak , . ? musíme zadať pomocou uvádzajúceho znaku \

```

```

REM Za znakmi ? ! ; hľadáme ľubovoľný alfanumerický znak
kolko=Vloz_hladanim("(\\?|!|;)[:alnum:]{1}", " ",1,true,false,false)

```

```

REM Za znakmi , . : hľadáme ľubovoľné písmeno
kolko=kolko+Vloz_hladanim("(\\,|\\.|:)[:alpha:]{1}", " ",1,true,false,false)

```

```
msgbox("Vložených "+kolko+" medzier.",0,"Vložené medzery")
```

```
end sub
```

Toto makro však nevkladá medzeru medzi bodku (čiarku, dvojbodku) a číslo v tom prípade, ak sa pred ním nachádza iný znak ako číslo (napr. „... sa zúčastnili všetci.12 ľudí potom ...“). Komu to nevyhovuje, musí si makro príslušne upraviť:

```
sub Vloz_interpunkcia_medzera
```

```
dim kolko as long
```

```
REM a týchto šesť interpunkčných znamienok: , . ! ? ; :
```

```
REM Znaky , . ? musíme zadať pomocou uvádzajúceho znaku \
```

```
REM Za znakmi ? ! ; hľadáme ľubovoľný alfanumerický znak
```

```
kolko=Vloz_hladanim("(\\?!|;)[:alnum:]{1}"," ",1,true,false,false)
```

```
REM Za znakmi , . : hľadáme ľubovoľné písmeno
```

```
kolko=kolko+Vloz_hladanim("(\\,|\\.|:)[alpha:]{1}"," ",1,true,false,false)
```

```
REM Ak pred znakmi , . : nie je číslica a za nimi je číslica, vložíme medzeru
```

```
kolko=kolko+Vloz_hladanim("[^0-9](\\,|\\.|:)[digit:]{1}"," ",2,true,false,false)
```

```
msgbox("Vložených "+kolko+" medzier.",0,"Vložené medzery")
```

```
end sub
```

Teraz však prehľadávame dokument v prípade znakov bodky, čiarky a dvojbodky dvakrát. Toto sa dá vyriešiť napríklad takto:

```
sub Vloz_interpunkcia_medzera
```

```
dim kolko as long
```

```
REM a týchto šesť interpunkčných znamienok: , . ! ? ; :
```

```
REM Znaky , . ? musíme zadať pomocou uvádzajúceho znaku \
```

```
REM Za znakmi ? ! ; hľadáme ľubovoľný alfanumerický znak
```

```
kolko=Vloz_hladanim("(\\?!|;)[:alnum:]{1}"," ",1,true,false,false)
```

```
REM Ak pred znakmi , . : nie je číslica a za nimi je ľubovoľný alfanumerický znak
```

```
kolko=kolko+Vloz_hladanim("[^0-9](\\,|\\.|:)[alnum:]{1}"," ",2,true,false,false)
```

```
msgbox("Vložených "+kolko+" medzier.",0,"Vložené medzery")
```

```
end sub
```

Malá rekapitulácia

V dnešnom dieli seriálu o programovaní makier v OpenOffice.org si zrekapitulujeme všetky makrá pre formátovanie dokumentov, ktoré sme doteraz vytvorili a trochu (teda dosť podstatne) ich upravíme.

Na úvod si urobme krátky prehľad doteraz naprogramovaných makier pre formátovanie dokumentov:

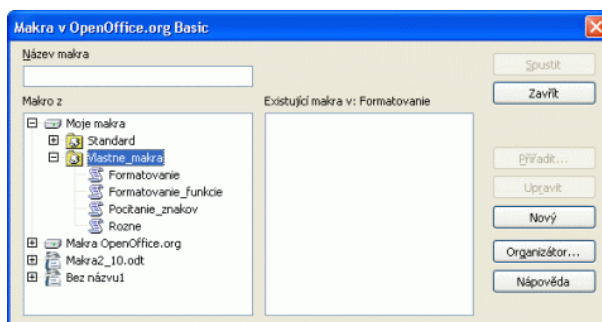
- makro pre odstraňovanie viacnásobných medzier „Viacnasobna_Medzera“
- makro pre vkladanie nezalomiteľných medzier za predložky a spojky „Nezalomitelne_Spojky“
- makro pre vkladanie nezalomiteľných medzier pred a za akademické tituly „Nezalomitelne_tituly“
- makro pre vkladanie nezalomiteľných medzier do dátumov „Nezalomitelne_datumy“
- makro pre vkladanie nezalomiteľných medzier medzi jednociferné čísla a text „Nezalomitelne_Jednociferne_Cisla“
- makro pre vkladanie nezalomiteľných medzier v telefónnych číslach „Nezalomitelne_Telefonne_Cisla“
- makro pre vkladanie nezalomiteľných medzier medzi čísla a merné jednotky „Nezalomitelne_Merne_Jednotky“
- makro pre mazanie nadbytočných medzier pri interpunkčných znamienkach „Zmaz_Interpunkcia_Medzera“
- makro pre vkladanie chýbajúcich medzier pri interpunkčných znamienkach „Vloz_Interpunkcia_Medzera“

K tomu ešte musíme pridať makro pre zámenu troch bodiek za „trojbodku“, ktoré sme naznačili ešte v štvrtom pokračovaní prvej časti tohto seriálu a ktoré môžeme nazvať „Rozne_Zameny“.

Pri praktickej úprave dokumentov je veľmi nepohodlné, aby sme každé makro spúšťali osobitne a tak isto nechceme, aby sme museli pri ich spoločnom spustení osobitne potvrdzovať zistené počty jednotlivých prevedených zámien. Na druhej strane však niekedy potrebujeme práve osobitné, resp. selektívne spúšťanie. Aby sme toto všetko mohli vyriešiť, musíme si uvedené makrá prispôbiť.

Každé makro rozdelíme na dve časti. Prvá časť bude funkcia, ktorá prevedie príslušné formátovanie a ako výsledok nám vráti počet výmien (opráv) bez toho, aby ho vypisovala na obrazovku. Druhá časť bude malý podprogram, ktorý bude predmetnú funkciu volať a následne aj vypisovať počet prevedených opráv.

Aby sme si udržali potrebný prehľad, rozdelíme si tieto dve časti do osobitných modulov v rámci skupín makier. Cez menu „Nástroje – Makrá – Správca makier – OpenOffice.org Basic...“ si otvoríme dialógové okno „Makrá v OpenOffice.org Basic“. V skupine makier „Vlastne_makra“ si vytvoríme dva moduly, ktoré nazveme „Formatovanie“ a „Formatovanie_funkcie“. V module „Formatovanie“ budeme mať malé podprogramy, ktoré budú volať vlastné formátovacie funkcie, ktoré uložíme do modulu „Formatovanie_funkcie“.



Obrázok 1: Pre makrá vytvoríme osobitné moduly

Aby sme rozlíšili funkciu od podprogramu, vložíme na jej začiatok písmeno „f“. Celý modul „Formatovanie_funkcie“ bude vyzeráť nasledovne:

REM Definície spoločných premenných sú úplne pred definíciou podprogramov

```
dim Dokument, Vymena, Hladaj as object
dim NM as string
```

REM Inicializácia spoločných premenných

```
sub Init
  Dokument=ThisComponent
  Vymena=Dokument.createReplaceDescriptor()
  Hladaj=Dokument.createSearchDescriptor()
  NM=chr$(&HA0) ' Nezalomitelná medzera – hexadecimálny kód A0
end sub
```

REM Výmena reťazca Co za reťazec Za

```
function Vymen(Co, Za as string, Regularne, Cele_slova, Velke_pismena as Boolean)
as long
  Vymena.SearchString=Co
  Vymena.ReplaceString=Za
  Vymena.SearchRegularExpression=Regularne
  Vymena.SearchWords=Cele_slova
  Vymena.SearchCaseSensitive=Velke_pismena
  Vymen=Dokument.replaceAll(Vymena)
end function
```

REM Vlastná výmena reťazca Co za reťazec Za vo vnútri reťazca V_Com

```
function Zamen_Vo_Vnutri (V_Com, Co, Za as string) as string
  dim pom as string
  dim i, j as long
  pom=V_Com
  i=instr(V_Com,Co)
  j=len(V_Com)-i-len(Co)+1
  if i>0 then
    pom=left(V_Com,i-1)+Za
    if j>0 then
      pom=pom+right(V_Com,j)
    end if
  end if
  Zamen_Vo_Vnutri=pom
end function
```

```

REM Výmena vnútorného reťazca Co za reťazec Za hľadanim reťazca V_Com
function Vymen_hladanim(V_Com, Co, Za as string, Regularne, Cele_slova,
Velke_pismena as Boolean) as long
    dim nasiel as object
    dim kolko as long
    Hladaj.searchString=V_Com
    Hladaj.SearchRegularExpression=Regularne
    Hladaj.SearchWords=Cele_slova
    Hladaj.SearchCaseSensitive=Velke_pismena
    kolko=0
    nasiel = Dokument.findFirst(Hladaj)
    Do While NOT isNull(nasiel)
        if len(nasiel.String)>0 then
            nasiel.String=Zamen_Vo_Vnutri(nasiel.String,Co,Za)
            kolko=kolko+1
        end if
        nasiel = Dokument.findNext( nasiel.End, Hladaj)
    Loop
    Vymen_hladanim=kolko
end function

```

```

REM Vyhľadanie reťazca V_Com a na pozíciu Kde do neho vloží reťazec Co
function Vloz_hladanim(V_Com, Co as string, Kde as long, Regularne, Cele_slova,
Velke_pismena as Boolean) as long
    dim nasiel as object
    dim kolko, i, j as long
    dim pom as string
    Hladaj.searchString=V_Com
    Hladaj.SearchRegularExpression=Regularne
    Hladaj.SearchWords=Cele_slova
    Hladaj.SearchCaseSensitive=Velke_pismena
    kolko=0
    i=kde+1
    nasiel = Dokument.findFirst(Hladaj)
    Do While NOT isNull(nasiel)
        if len(nasiel.String)>0 then
            j=len(nasiel.String)-i+1
            pom=left(nasiel.String,i-1)+Co
            if j>0 then
                pom=pom+right(nasiel.String,j)
            end if
        end if
        nasiel = Dokument.findNext( nasiel.End, Hladaj)
    Loop
    Vloz_hladanim=pom
end function

```

```

end if
nasiel.String=pom
kolko=kolko+1
end if
nasiel = Dokument.findNext( nasiel.End, Hladaj)
Loop
Vloz_hladanim=kolko
end function

```

REM Makro pre odstraňovanie viacnásobných medzier

```

function f_Viacnasobna_Medzera as long
dim kolko AS Long
kolko=Vymen(" +", " ", TRUE, false, false)
kolko=kolko+Vymen(" "+NM+" |"+NM+" | "+NM, NM, TRUE, false, false)
kolko=kolko+Vymen(" \t |\t | \t", "\t", TRUE, false, false)
kolko=kolko+Vymen("^[:space:]*[:space:]*$", "", TRUE, false, false)
kolko=kolko+Vymen("^\t*\t*$", "", TRUE, false, false)
f_Viacnasobna_Medzera=kolko
end function

```

REM Makro pre vkladanie nezalomiteľných medzier za predložky a spojky

```

function f_Nezalomitelne_Spojky as long
f_Nezalomitelne_Spojky=Vymen_hladanim("\<(a|i|k|o|s|u|v|z) ", "
", NM, true, false, false)
end function

```

REM Makro pre vkladanie nezalomiteľných medzier pred a za akademické tituly

```

function f_Nezalomitelne_Tituly as long
dim Tituly_pred(), Tituly_pred_spec(), Tituly_za() as string
dim i, kolko as long
Tituly_pred()=array("Bc", "Dr", "Ing", "JUDr", "Mgr", "MUDr", "MVDr", "PaedDr",
"PharmDr", "PhDr", "PhMr", "RNDr", "ThDr")
Tituly_pred_spec()=array("akad", "Akad", "arch", "Arch", "doc", "Doc", "prof", "Prof")
Tituly_za()=array("CSc", "DrSc")
kolko=Vymen("h. c.", "h."+NM+"c.", false, false, false)
kolko=kolko+Vymen(" h.c.", NM+"h.c.", false, false, false)
kolko=kolko+Vymen(" h."+NM+"c.", NM+"h."+NM+"c.", false, false, false)
for i=lbound(Tituly_pred()) to ubound(Tituly_pred())
kolko=kolko+Vymen("\<" + Tituly_pred(i) + "\.
", Tituly_pred(i) + "." + NM, true, false, false)

```

```

next i
for i=lbound(Tituly_pred_spec()) to ubound(Tituly_pred_spec())
kolko=kolko+Vymen("\<" + Tituly_pred_spec(i) + "\.
", Tituly_pred_spec(i) + "." + NM, true, false, true)
next i
for i=lbound(Tituly_za()) to ubound(Tituly_za())
kolko=kolko+Vymen(" " + Tituly_za(i) + ". ", NM + Tituly_za(i), false, false, false)
next i
f_Nezalomitelne_Tituly=kolko
end function

```

REM Makro pre vkladanie nezalomitel'nych medzier do dátumov

```

function f_Nezalomitelne_Datумы as long
dim kolko as long
kolko=Vymen_hladanim("[0-9]\. [0-9]", " ", NM, true, false, false)
kolko=kolko+Vymen_hladanim("[0-9]\. [0-9]", " ", NM, true, false, false)
f_Nezalomitelne_Datумы =kolko
end function

```

REM Makro pre vkladanie nezalomitel'nych medzier medzi jednociferne čísla a text

```

function f_Nezalomitelne_Jednociferne_Cisla as long
f_Nezalomitelne_Jednociferne_Cisla =Vymen_hladanim("\<[0-9] ", "
", NM, true, false, false)
end function

```

REM Makro pre vkladanie nezalomitel'nych medzier v telefónnych číslach

```

function f_Nezalomitelne_Telefonne_Cisla as long
f_Nezalomitelne_Telefonne_Cisla =Vymen_hladanim("[0-9] [0-9]", "
", NM, true, false, false)
end function

```

REM Makro pre vkladanie nezalomitel'nych medzier medzi čísla a merné jednotky

```

function f_Nezalomitelne_Merne_Jednotky as long
f_Nezalomitelne_Merne_Jednotky=Vymen_hladanim("[0-9] (m|km|mm|cm)", "
", NM, true, false, false)
end function

```

REM Makro pre mazanie nadbytočných medzier pri interpunkčných znamienkach

```

function f_Zmaz_Interpunkcia_Medzera as long
dim kolko as long
kolko=Vymen_hladanim("(,|<|" + chr$( &H22) + "| \ ( | | { ) ", " ", " ", true, false, false)

```

```

kolko=kolko+Vymen_hladanim(" (|»|" +chr$(&H22)+"|\)|\]}|\,|\.\!|\?|;|:)",
",",true,false,false)
f_Zmaz_Interpunkcia_Medzera=kolko
end function

```

REM Makro pre vkladanie chýbajúcich medzier pri interpunkčných znamienkach

```

function f_Vloz_Interpunkcia_Medzera as long
dim kolko as long
kolko=Vloz_hladanim("(\\?!|;)[:alnum:]{1}", " ",1,true,false,false)
kolko=kolko+Vloz_hladanim("[^0-9](\\,|\\.|:)[:alnum:]{1}", " ",2,true,false,false)
f_Vloz_Interpunkcia_Medzera=kolko
end function

```

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

```

function f_Rozne_zameny as long
dim Co(), Za() as string
dim n as long
dim kolko as long
Co()=array("...")
Za()=array("...")
kolko=0
for n=lbound(Co()) to ubound(Za())
kolko=kolko+Vymen(Co(n),Za(n),false,false,false)
next n
f_Rozne_zameny=kolko
end function

```

Všimnite si zaujímavú zmenu makra pre vkladanie nezalomiteľných medzier za jednoznakové predložky a spojky, pretože sme aj pri ňom využili našu vlastnú funkciu pre výmenu.

Záverečné úpravy

V poslednom dieli druhej časti seriálu o programovaní makier v OpenOffice. org dokončíme prvú (ale nie poslednú) komplexnú verziu makier pre formátovanie dokumentov.

Naposledy sme si povedali, že makrá pre formátovanie dokumentov rozdelíme do dvoch modulov, pričom sme si uviedli zdrojové texty modulu „Formatovanie_funkcie“. Dnes si ukážeme, ako budeme tieto funkcie volať v procedúrach, ktoré uložíme do modulu „Formatovanie“. Prístup k podprogramom, ktoré sa nachádzajú v iných moduloch nám umožňuje zápis, ktorý je podobný zápisu pri volaní metód v rámci objektových premenných:

```
meno_modulu.meno_podprogramu
```


Ako sme si spomínali minule, chceme zároveň naprogramovať procedúru, ktorá bude volať všetky formátovacie funkcie. Pri ich volaní však nemôžeme použiť hocikaké poradie, pretože potom by sme nedosiahli požadovaný výsledok. Ako prvé musia byť prevedené rôzne zámeny, pretože napr. pred „trojbodkou“ nechávame medzeru, ale pred bodkou nie. Následne je potrebné vymazať viacnásobné medzery, lebo potom by v prípade ich výskytu napr. za spojku zostala kombinácia nezalomiteľná + zalomiteľná medzera. Ešte pred opravou zalomiteľných medzier za spojkami a predložkami sa najprv musí vložiť medzera za interpunkčné znamienka, lebo potom by mohla zostať zalomiteľná medzera za spojku, ak predtým pred ňou chýbala. Takisto pre opravu predložiek a spojok musíme zmazať nadbytočné medzery pri interpunkčných znamienkach, pretože v prípade, že napr. na konci zátvorky je iba spojka, zbytočne sa najprv zamení medzera za nezalomiteľnú, aby sa za chvíľu úplne vymazala. Pretože sme vo funkcii pre mazanie nadbytočných medzier pri interpunkčných znamienkach neuvažovali s možnosťou nezalomiteľnej medzery a tabelátora, musíme si najprv v module „Formatovanie_funkcie“ opraviť túto funkciu:

```
function f_Zmaz_Interpunkcia_Medzera as long
    dim TAB as string
    dim kolko as long
    TAB=chr$(&H09) ' Tabelátor
    kolko=Vymen_hladanim(",|<|" + chr$(&H22) + "\(\[|{) ", " ", "", true, false, false)
    kolko=kolko+Vymen_hladanim(",|<|" + chr$(&H22) + "\(\[|{)" + NM, NM, "", true, false, false)
    kolko=kolko+Vymen_hladanim(",|<|" + chr$(&H22) + "\(\[|{)" + TAB, TAB, "", true, false, false)
    kolko=kolko+Vymen_hladanim(" (|>|" + chr$(&H22) + "\)\]|}|\,|\.|!\|?|;|:)", "", "", true, false, false)
    kolko=kolko+Vymen_hladanim(NM + " (|>|" + chr$(&H22) + "\)\]|}|\,|\.|!\|?|;|:)", NM, "", true, false, false)
    kolko=kolko+Vymen_hladanim(TAB + " (|>|" + chr$(&H22) + "\)\]|}|\,|\.|!\|?|;|:)", TAB, "", true, false, false)
    f_Zmaz_Interpunkcia_Medzera=kolko
end function
```

V prípade, že budete potrebovať znak tabelátora aj v iných funkciách, je vhodné zaradiť jeho definíciu medzi spoločné premenné a následne jeho inicializáciu zaradiť do podprogramu Init. To však ponecháme prípadným záujemcom ako domácu úlohu. Modul „Formatovanie“, kde sú definované podprogramy pre formátovanie dokumentu môže pri dodržaní vyššie uvedených podmienok vyzeráť napr. takto:

```
REM Makro pre odstraňovanie viacnásobných medzier
sub Viacnasobna_Medzera
    Formatovanie_funkcie. Init
    msgbox("Vymazanie" + Formatovanie_funkcie.
f_Viacnasobna_Medzera() + "viacnásobných medzier.", 0, "Viacnásobné medzery")
end sub

REM Makro pre vkladanie nezalomiteľných medzier za predložky a spojky
sub Nezalomitelne_Spojky
    Formatovanie_funkcie. Init
```

```
    msgbox("Vložených"+Formatovanie_funkcie.  
f_Nezalomitelne_Spojky()+"nezalomiteľných medzier.", 0,"Nezalomiteľné spojky a  
predložky")  
end sub
```

```
REM Makro pre vkladanie nezalomiteľných medzier pred a za akademické tituly  
sub Nezalomitelne_Tituly  
    Formatovanie_funkcie. Init  
    msgbox("Vložených"+Formatovanie_funkcie.  
f_Nezalomitelne_Tituly()+"nezalomiteľných medzier.", 0,"Nezalomiteľné tituly")  
end sub
```

```
REM Makro pre vkladanie nezalomiteľných medzier do dátumov  
sub Nezalomitelne_Datумы  
    Formatovanie_funkcie. Init  
    msgbox("Vložených"+Formatovanie_funkcie.  
f_Nezalomitelne_Datумы()+"nezalomiteľných medzier.", 0,"Nezalomiteľné dátумы")  
end sub
```

```
REM Makro pre vkladanie nezalomiteľných medzier medzi jednociferné čísla a text  
sub Nezalomitelne_Jednociferne_Cisla  
    Formatovanie_funkcie. Init  
    msgbox("Vložených"+Formatovanie_funkcie.  
f_Nezalomitelne_Jednociferne_Cisla()+"nezalomiteľných medzier.", 0,"Nezalomiteľné  
jednociferné čísla")  
end sub
```

```
REM Makro pre vkladanie nezalomiteľných medzier v telefónnych číslach  
sub Nezalomitelne_Telefonne_Cisla  
    Formatovanie_funkcie. Init  
    msgbox("Vložených"+Formatovanie_funkcie.  
f_Nezalomitelne_Telefonne_Cisla()+"nezalomiteľných medzier.", 0,"Nezalomiteľné  
telefónne čísla")  
end sub
```

```
REM Makro pre vkladanie nezalomiteľných medzier medzi čísla a merné jednotky  
sub Nezalomitelne_Merne_Jednotky  
    Formatovanie_funkcie. Init  
    msgbox("Vložených"+Formatovanie_funkcie.  
f_Nezalomitelne_Merne_Jednotky()+"nezalomiteľných medzier.", 0,"Nezalomiteľné merné  
jednotky")  
end sub
```

```

REM Makro pre vloženie nezalomiteľných medzier za čísla ako také
sub Nezalomitelne_Cisla
dim kolko as long
Formatovanie_funkcie. Init
kolko=Formatovanie_funkcie. f_Nezalomitelne_Datумы()
kolko=kolko+Formatovanie_funkcie. f_Nezalomitelne_Jednociferne_Cisla()
kolko=kolko+Formatovanie_funkcie. f_Nezalomitelne_Telefonne_Cisla()
kolko=kolko+Formatovanie_funkcie. f_Nezalomitelne_Merne_Jednotky()
msgbox("Vložených"+kolko+"nezalomiteľných medzier.", 0,"Nezalomiteľné čísla")
end sub

```

```

REM Makro pre mazanie nadbytočných medzier pri interpunkčných znamienkach
sub Zmaz_Interpunkcia_Medzera
Formatovanie_funkcie. Init
msgbox("Vymazaných"+Formatovanie_funkcie.
f_Zmaz_Interpunkcia_Medzera()+"nadbytočných medzier.", 0,"Nadbytočné interpunkčné
medzery")
end sub

```

```

REM Makro pre vkladanie chýbajúcich medzier pri interpunkčných znamienkach
sub Vloz_Interpunkcia_Medzera
Formatovanie_funkcie. Init
msgbox("Vložených"+Formatovanie_funkcie.
f_Vloz_Interpunkcia_Medzera()+"chýbajúcich medzier.", 0,"Chýbajúce interpunkčné
medzery")
end sub

```

```

REM Makro pre opravu (vloženie/vymazanie) medzier pri interpunkčných znamienkach
sub Oprav_Interpunkcia_Medzera
Formatovanie_funkcie. Init
msgbox("Opravených"+Formatovanie_funkcie.
f_Zmaz_Interpunkcia_Medzera()+Formatovanie_funkcie.
f_Vloz_Interpunkcia_Medzera()+"medzier.", 0,"Nesprávne interpunkčné medzery")
end sub

```

```

REM Makro pre rôzne záměny (napr. troch bodiek za „trojbodku“)
sub Rozne_zaměny
Formatovanie_funkcie. Init
msgbox("Vykonaných"+Formatovanie_funkcie. f_Rozne_zaměny()+"výmien.",
0,"Rôzne záměny")
end sub

```

```

REM Makro pre celkové formátovanie dokumentu
sub Formatuj_Dokument
dim kolko as long
Formatovanie_funkcie. Init
' Prvé musia byť rôzne zámeny
kolko=Formatovanie_funkcie. f_Rozne_Zameny()
' Potom odstránime viacnásobné medzery
kolko=kolko+Formatovanie_funkcie. f_Viacnasobna_Medzera()
' Opravíme medzery pri interpunkčných znamienkach
kolko=kolko+Formatovanie_funkcie.
f_Zmaz_Interpunkcia_Medzera()+Formatovanie_funkcie. f_Vloz_Interpunkcia_Medzera()
' Vložíme nezalomitelné medzery k číslam
kolko=kolko+Formatovanie_funkcie.
f_Nezalomitelne_Datумы()+Formatovanie_funkcie.
f_Nezalomitelne_Jednociferne_Cisla()+Formatovanie_funkcie.
f_Nezalomitelne_Telefonne_Cisla()+Formatovanie_funkcie.
f_Nezalomitelne_Merne_Jednotky()
' Opravíme nezalomitelné medzery pri akademických tituloch
kolko=kolko+Formatovanie_funkcie. f_Nezalomitelne_Tituly()
' A nakoniec opravíme nezalomitelné medzery pri spojkách a predložkách
kolko=kolko+Formatovanie_funkcie. f_Nezalomitelne_Spojky()
msgbox("Prevedných"+kolko+"náhrad.", 0,"Formátovanie textu")
end sub

```

Skončili sme druhú veľkú časť seriálu o programovaní makier v OpenOffice. org. Ako ste si mohli všimnúť aj v dnešnom dieli, uvedené funkcie určite neposkytujú všetky možnosti, ktoré pri práci potrebujeme. Na základe doterajších príkladov si však už určite dokážete naprogramovať aj prípadné úpravy, pomocou ktorých vhodne upravíte alebo rozšírite ich možnosti. V našom seriáli budeme teraz pokračovať treťou veľkou časťou, v ktorej si naznačíme možnosti programovania makier pre OpenOffice.org v oblasti vstupov a výstupov, t.j. budeme programovať rôzne vstupné formuláre (dialógové okná) a pod. Pri tomto programovaní sa stále budeme zaoberať funkciami pre formátovanie dokumentov, takže tieto znovu prekonajú niekoľko zmien.