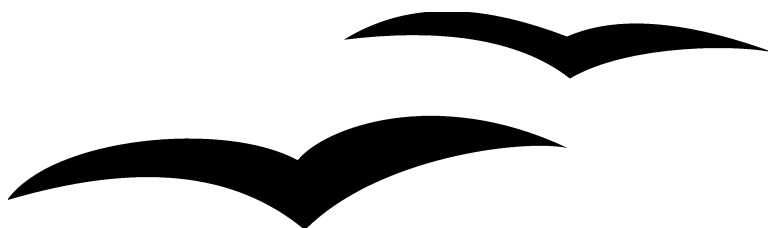


Makrá v OpenOffice.org Dialógy



Obsah

Nezalomiteľné medzery počas písania.....	3
Obsluha klávesnice.....	3
Posledne vložené slovo.....	3
Prvý dialóg.....	6
Organizácia dialógov.....	7
Vlastný návrh dialógového okna.....	7
Dialóg pre nezalomiteľnú medzeru.....	8
Zoznam položiek.....	8
Optické členenie a nápovedné texty.....	9
Zaškrtávacie políčka.....	9
Tlačidlá „OK“ a „Cancel“.....	9
Okrasné a oddeľujúce prvky.....	10
Celkové „kozmetické“ úpravy.....	10
Základné funkcie pre prácu s dialógmi.....	11
Dialóg ako objekt v jazyku StarOffice Basic.....	11
Nastavovanie parametrov pomocou dialógu.....	12
Funkcie pre odpamätanie nastavení do súboru.....	12
Funkcie pre vlastnú prácu s dialógovým oknom.....	14
Vlastné vkladanie nezalomiteľných medzier počas ich písania.....	16
Pružná nezalomiteľná medzera.....	18
Čo na to hovorí matematika.....	19
A čo na to hovoria kódy znakov.....	19
Aké medzery vlastne používať.....	20
Obyčajne potrebujeme všetky tri typy.....	20
Zadefinujeme si nové dialógové okno s prepínacími položkami.....	20
Inicializácia prepínačov.....	27
Ako si v tom udržať poriadok a prehľad.....	27
Automatické spúšťanie počas otvárania súboru.....	28
Niekoľko nedostatkov a chýb.....	28
Niekoľko slov na záver	29

Začíname štvrtú časť seriálu o programovaní makier v OpenOffice.org, ktorú budeme venovať dialógom. Najskôr však trochu odbočíme a znovu sa vrátíme k makru pre vkladanie nezalomiteľných medzier za jednoznakové predložky a spojky.

Nezalomiteľné medzery počas písania

Vkladanie nezalomiteľných medzier už počas písania jednoznakových predložiek a spojok je požiadavka, ktorú sme doteraz márne od OpenOffice.org očakávali. Pretože toto makro sa nám zároveň bude vynikajúco hodiť ako východiskový materiál pre dialógy, začneme najprv s ním.

Obsluha klávesnice

V prípade takejto obsluhy si musíme uvedomiť, že potrebujeme priamo obsluhovať stlačené klávesy. Na tento účel môžeme do systému priradiť vlastné funkcie, ktoré však musia spĺňať určité (UNO) špecifiká. Musia to byť vlastne dve logické funkcie „KeyPressed“ a „KeyReleased“, pričom pomocou prvej z nich obsluhujeme „stlačenie“ a pomocou druhej z nich „pustenie“ klávesy. Obidve tieto funkcie však musia mať rovnaký prefix (t.j. prvú časť názvu), napr. „Nazov_“, pretože práve pomocou tohto prefixu ich zväzujeme so systémom:

```
function Nazov_KeyPressed(StlacenaKlavesa) as boolean  
function Nazov_KeyReleased(StlacenaKlavesa) as boolean  
createUnoListener("Nazov_", "com.sun.star.awt.XKeyHandler")
```

OpenOffice.org predpokladá, že takéto funkcie sú naozaj iba pomocné a väčšinou budeme chcieť využiť aj originálne funkcie na obsluhu klávesnice. Toto zaisťuje automaticky podľa toho, akú výstupnú logickú hodnotu týmto funkciám priradíme. Ak im priradíme hodnotu „TRUE“, potom pôvodné funkcie nevolá (predpokladá, že celú obsluhu sme vykonali sami). Ak im priradíme hodnotu „FALSE“, potom akoby sme žiadnu obsluhu klávesnice nevykonali, resp. naše funkcie sú pokladané naozaj za iba pomocné a volajú sa ešte aj pôvodné (originálne) obslužné funkcie.

Posledne vložené slovo

Pre vlastné sledovanie predložiek si musíme uvedomiť, že ich nebudeme späť vyhľadávať, ale po zadaní zalomiteľnej medzery musíme hneď vedieť, či predchádzajúce slovo bola alebo nebola jednoznaková predložka (spojka) a podľa toho sa zariadime. Preto musíme zo stlačených kláves sledovať zadávané slovo (ale iba posledné slovo). V rámci tohto sledovania si musíme ešte uvedomiť, že pri písaní sa aj mýlime a používame aj opravné klávesy (DEL, BACKSPACE, ...). Ako vidíme, táto problematika nie je nijako jednoduchá a to ešte nehovoríme o tom, že sa môžeme kedykoľvek myšou presunúť na úplne iné miesto v texte, tam opraviť nejaký znak a znovu sa vrátiť na terajšie miesto a pod. Aby sme sa dopracovali k pokiaľ možno čo najjednoduchšiemu riešeniu, obmedzíme sa iba na niektoré zo spomenutých problémov a prípadné rozšírenia už necháme na jednotlivých používateľov.

Skôr ako uvedieme zdrojové texty, ešte pripomeňme, že premenné, cez ktoré budeme sprístupňovať dokument a príslušnú obsluhu klávesnice musia byť viditeľné počas celého písania a preto musíme namiesto deklarácie DIM použiť deklaráciu GLOBAL, pretože takto zaisťujeme, že sa ich definícia a nadobudnuté hodnoty nestratia. Vlastné makrá môžu teraz vyzeráť napríklad takto:

```

global ObsluhaDokumentu, ObsluhaKlavesnice as object
global PosledneSlovo, OddelovaceSlov, JednoznakovePredlozky as string

sub SpustiObsluhuKlavesnice
on error goto next

' Inicializácia premenných
PosledneSlovo = ""
JednoznakovePredlozky="aAiIkKoOsSuUvVzZ"
' Znaký, ktoré ukončujú slová – medzera, nezalomiteľná medzera, tabulátor, CR, LF
OddelovaceSlov=" "+chr(&HA0)+chr(&H09)+chr(&H0A)+chr(&H0D)
' pomlčka, lomítko, bodka, čiarka, výkričník, otáznik, bodkočiarka, trojbodka
OddelovaceSlov=OddelovaceSlov+"-./,!?:;..."
' zátvorky
OddelovaceSlov=OddelovaceSlov+"()[]{}"

' Aktuálny dokument
ObsluhaDokumentu = ThisComponent.getCurrentController
' Vlastná obsluha klávesnice
ObsluhaKlavesnice = createUnoListener("Klavesnica_", "com.sun.star.awt.XKeyListener")
' Priradenie tejto obsluhy k aktuálnemu dokumentu
ObsluhaDokumentu.addHandler(ObsluhaKlavesnice)

end sub

sub UkonciObsluhuKlavesnice
on error goto next
ObsluhaDokumentu.removeKeyListener(ObsluhaKlavesnice)
PosledneSlovo = ""
end sub

sub VlozZnak(Retazec as string)
' Na aktuálnu pozíciu v dokumente vložíme reťazec
dim ViditelnyKurzor, AktualnyText, AktualnyKurzor as object
ViditelnyKurzor = ThisComponent.getCurrentController().getViewCursor()
AktualnyText = ThisComponent.getText()

```

```
AktualnyKurzor = AktualnyText.createTextCursorByRange(ViditelnyKurzor.getStart())
AktualnyText.insertString(AktualnyKurzor.getStart(), Retazec, true)
```

```
end sub
```

```
function Klavesnica_KeyPressed(StlacenaKlavesa) as boolean
```

```
' Obsluha stlačenej klávesy
```

```
dim Obsluzena as boolean
```

```
'Predpokláme, že neobslúžime klávesnicu
```

```
Obsluzena=False
```

```
if instr(OddelovaceSlov,StlacenaKlavesa.KeyChar)<>0 then
```

```
' Došli sme na koniec slova
```

```
if ZistiPredlozku and (StlacenaKlavesa.KeyChar=" ") then
```

```
' Je to predložka, ktorá končí zalomiteľnou medzerou - túto nahradíme
```

```
VlozZnak(chr$(&HA0)) ' No Break Space
```

```
' Sami sme vložili nezalomiteľnú medzeru - obslúžili sme stlačenú klávesu
```

```
Obsluzena=true
```

```
endif
```

```
' Boli sme na konci slova, už nás nebude zaujímať
```

```
PosledneSlovo=""
```

```
else
```

```
' Neboli sme na konci slova
```

```
' Ešte musíme otestovať riadiace znaky, ktoré nebudeme sledovať v zadávanom slove
```

```
select case StlacenaKlavesa.KeyChar
```

```
case chr$(&H00) to chr$(&H07)
```

```
PosledneSlovo=""
```

```
case chr$(&H08) ' BackSpace
```

```
if len(PosledneSlovo)>0 then
```

```
PosledneSlovo=left(PosledneSlovo,len(PosledneSlovo)-1)
```

```
else
```

```
PosledneSlovo=""
```

```
endif
```

```
case chr$(&H09) to chr$(&H1F)
```

```
PosledneSlovo=""
```

```

case chr(&H20) to chr(&H7E)
    PosledneSlovo=PosledneSlovo+StlacenaKlavesa.KeyChar
case chr(&H7F) ' CTRL BS
    PosledneSlovo=""
case else
    PosledneSlovo=PosledneSlovo+StlacenaKlavesa.KeyChar
end select
endif
Klavesnica_KeyPressed=Obsluzena
end function

function Klavesnica_KeyReleased(StlacenaKlavesa) as boolean
' Obsluha pustenej klávesy
Klavesnica_KeyReleased = False
end function

function ZistiPredlozku as boolean
dim JePredlozka as boolean

JePredlozka=false
if len(PosledneSlovo)=1 then
' Ak je to jednoznakové slovo, tak to môže byť predložka
JePredlozka=instr(1,JednoznakovePredlozky,PosledneSlovo,0)
endif
ZistiPredlozku=JePredlozka

end function

```

Prvý dialóg

Už sme si ukázali viacero spôsobov, ako nastaviť parametre pre interaktívne riadenie makier v OpenOffice.org. Posledná možnosť, ktorej budeme venovať celú túto časť seriálu sú dialógy, alebo dialógové okná. V dnešnom dieli si ukážeme postup, ako do makier vôbec dokážeme pridať vlastné dialógové okno.

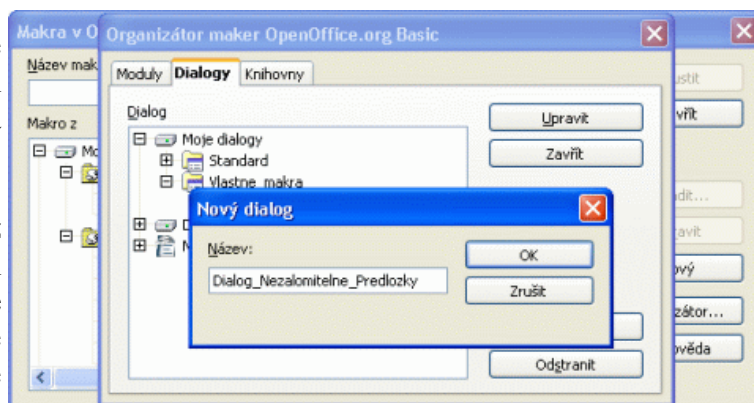
Aby sme dialógy mohli navrhovať, musíme si najprv otvoriť dialógové okno „Makrá v OpenOffice.org Basic“, ku ktorému sa dostaneme cez voľby v menu „Nástroje/Makrá/Správca makier/OpenOffice.org Basic...“. Stlačením tlačidla „Organizátor...“ si sprístupnime dialógové okno „Organizátor makier v OpenOffice.org Basic“, kde vidíme tri záložky: „Moduly“, „Dialógy“ a „Knížnice“. O záložkách „Moduly“

a „Knižnice“ sme si už hovorili na začiatku celého tohto seriálu a preto sa teraz nimi už nebudeme zaoberať, ale pozornosť zameriame na strednú záložku „Dialogy“.

Organizácia dialógov

Aj na tejto záložke nájdeme stromovú štruktúru jednotlivých knižníc, po rozbalení ktorých sa nám zobrazí zoznam dialógov, ktoré v nich máme definované. V organizátorovi máme teraz možnosť pridať názvy nových dialógov, ktoré chceme v budúcnosti definovať, takže napr. do knižnice „Vlastne_makra“ si cez tlačidlo „Nový...“ pridáme nový dialóg,

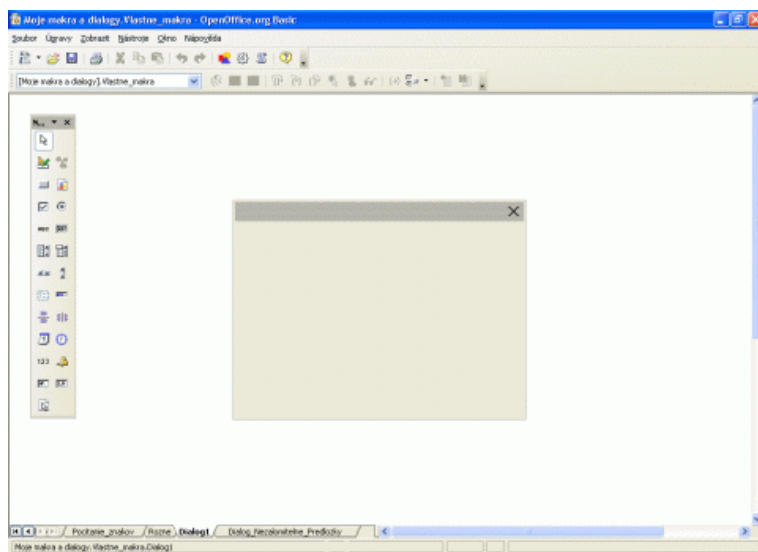
ktorý nazveme napr. „Dialog_Nezalomitelne_Predlozky“. Pozn.: vzhľadom na to, že mám definovaný iný dialóg s takýmto názvom (kde mám definované napr. aj výber medzi vkladáním pružnej a nepružnej nezalomiteľnej medzery), pridal som teraz pracovne znak podčiarknika aj na koniec názvu, aby som to rozoznal. Celý dialóg by však bolo ešte predčasne zverejňovať.



Obrázok 1: Zadanie názvu nového dialógu

Vlastný návrh dialógového okna

Vlastný dialóg môžeme začať navrhovať pomocou tlačidla „Upraviť“. Týmto sa nám otvorí modul Basic, kde vkladáme aj vlastné makrá, pochopiteľne, na záložke, kde máme pripravené



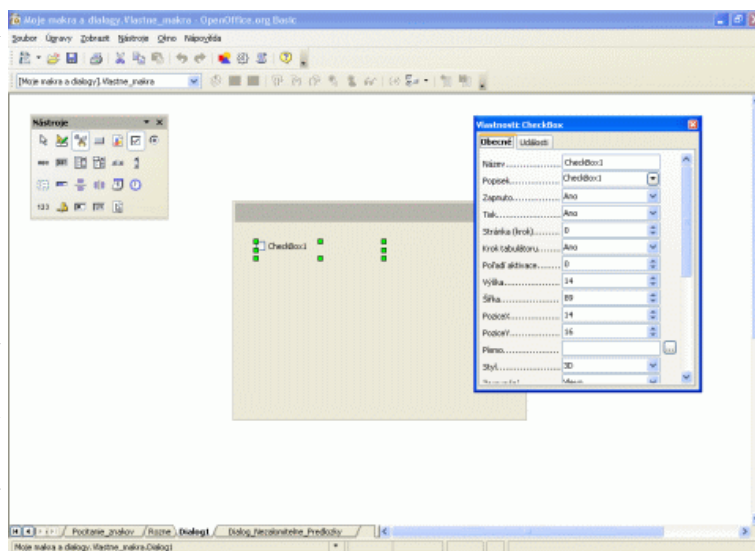
Obrázok 2: Pracovné rozhranie pre návrh dialógu

základné okno budúceho dialógu. Ak si dobre všimnete modul Basic zistíte, že v spodnej časti sú záložky, pomocou ktorých sa môžete ľahko prepínať medzi jednotlivými časťami príslušnej knižnice, t.j. ako medzi dialógmi, tak aj príslušnými modulmi s podprogramami. Zároveň v ľavej hornej časti nájdete rozbaľovacie okno, kde môžete rýchlo prepínať medzi jednotlivými knižnicami, takže vám stačí modul Basic otvoriť raz a veľmi jednoducho sa dostanete na ktorékolvek makro či dialóg v celom vašom systéme makier.

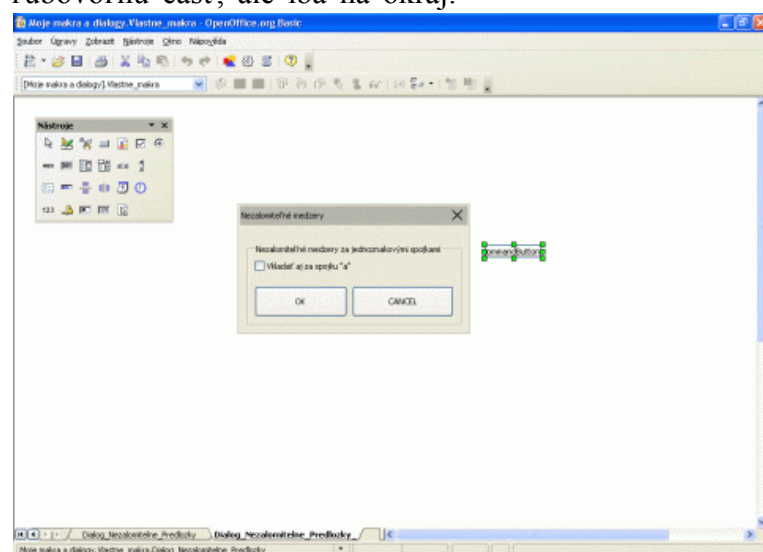
Venujme sa však vlastnému návrhu dialógu. V pracovnom okne vidíme dve časti – zoznam nástrojov a prázdne dialógové okno s jediným prvkom – krížikom „X“, ktorý naznačuje budúce miesto skutočného krížika „X“ na jeho zatváranie. Všetko ostatné si teraz musíme zdefinovať. Vlastné definovanie je veľmi jednoduché - v zozname nástrojov si označíme prvok dialógového okna, ktorý tam chceme vložiť, presunieme sa do nášho navrhovaného okna, stlačíme ľavé tlačidlo myši a potiahnutím vyznačíme oblasť, kde sa má prvok umiestniť. Pravdaže, nikto nepredpokladá, že sa na prvý krát presne trafíme na miesto, kde

prvok chceme mať a že zároveň presne nastavíme aj jeho veľkosť. Okrem toho budeme následne aj tak nastavovať rôzne texty, ktoré si môžu vynútiť zmenu veľkosti nielen tohto prvku, ale aj celého navrhovaného dialógového okna.

Preto si teraz v zozname nástrojov vyberieme výberovú šípku (je to prvý prvok úplne vľavo hore) a označíme si ten prvok, ktorá chceme upravovať. Takýmto istým spôsobom sa dá označiť aj celé dialógové okno, ale nesmieme kliknúť na jeho ľubovoľnú časť, ale iba na okraj.



Obrázok 3: Vložíme prvok a pozrieme si jeho vlastnosti



Obrázok 4: Prvok môžeme umiestniť aj mimo dialógového okna – návrh sa ešte upraví

predpokladá, že sa s veľkosťou ešte „pohráte“, t.j. nakoniec všetko nastavíte tak, aby sa to zmestilo do vytváraného dialógu.

Dialóg pre nezalomiteľnú medzeru

V minulom dieli sme si poukázali na základné možnosti, ktoré nám pri návrhu vlastného dialógového okna ponúka OpenOffice.org. Pokiaľ ste si za domácu úlohu odskúšali, ako sa s týmto modulom pracuje, nebudete mať dnes žiadne problémy.

Zoznam položiek

Ako prvý krok, ktorý si musíme premyslieť, je vlastne zoznam položiek, ktoré v dialógovom okne budeme mať. V prípade, že ideme nastavovať parameter pre vkladanie nezalomiteľnej medzery za spojku „a“ to bude najlepšie zaškrŕavacie políčko, pomocou ktorého nastavíme, či sa za túto spojku má alebo nemá vkladať nezalomiteľná medzera. Ďalej si musíme určiť, ako sa má príslušný parameter nastaviť, t.j. či pri zaškrŕnutom políčku chceme alebo naopak nechceme vkladať túto medzeru, aby sme podľa toho vhodne formulovali príslušné sprievodné texty.

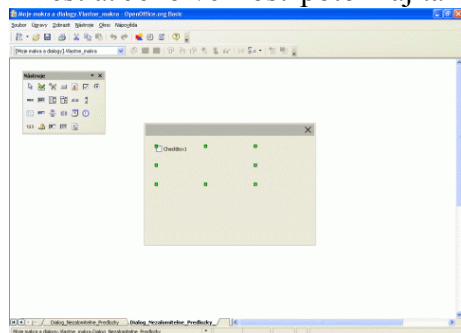
Tým by sa zdalo, že máme všetky políčka vyčerpané, ale nie je to tak. Pokiaľ totiž chceme aj získať zadané hodnoty, musíme dialógové okno korektne zatvoriť pomocou tlačidla „OK“. Toto tlačidlo si však musíme do neho tak isto zadať. Okrem toho je vhodné, aby sme si zadefinovali aj tlačidlo „Cancel“ pre prípad, že nechceme previesť žiadnu zmenu a nevieme, čo všetko sme vlastne pomenili. Toto tlačidlo sa zatiaľ síce môže zdať nadbytočné, v budúcich verziách však nadobudne svoj význam a preto sa ním budeme zaoberať už teraz.

Optické členenie a nápovedné texty

Ako posledné, čo si navrhujeme, bude rôzne optické členenie pomocou rámkov s nápovedným textom. Týmto spôsobom dokážeme veľmi jednoducho opticky označiť určité skupiny tlačidiel, čo sa nám tak isto zídne najmä v budúcnosti pri komplexnejších návrhoch. Pochopiteľne, úplne na záver nesmieme zabudnúť ani na názov nami navrhovaného dialógu.

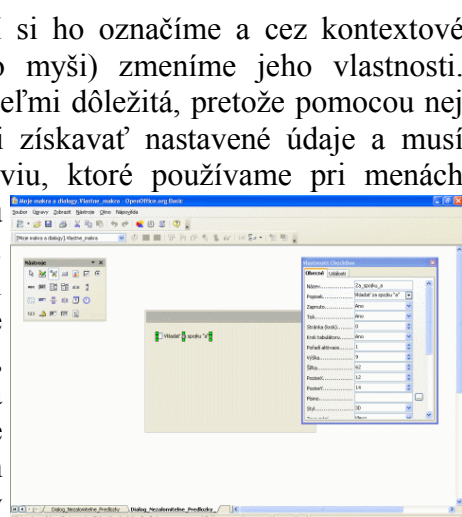
Zaškrŕavacie políčka

Teraz sa však už venujme vlastnému návrhu. Predpokladajme, že sme si otvorili prázdny dialóg, do ktorého teraz postupne budeme vkladať jednotlivé prvky. Aby sme nestratili potrebný prehľad, budeme postupovať presne podľa bodov, ktoré sme si uviedli pred chvíľou a preto ako prvé vložíme zaškrŕavacie políčko. Po jeho výbere v poli nástrojov nakreslíme vo vlastnom dialógovom okne radšej väčší obdĺžnik, aby sme mali istotu, že sa tam všetky údaje zmestia. Jeho veľkosť potom aj tak upravíme.



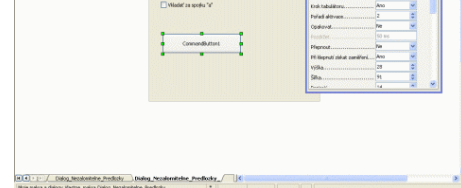
Obrázok 5: Ako prvé vložíme zaškrŕavacie políčko

„Za_spojku_a“. Ďalej zmeníme položku „Popis“, pretože tento text sa vypisuje za vlastným zaškrŕavacím políčkom priamo v dialógovom okne. Povedzme, že chceme, aby zaškrŕnuté políčko znamenalo, že sa má za spojku „a“ vkladať nezalomiteľná medzera a preto si sem vložíme



Obrázok 6: Zobrazíme si vlastnosti vloženého objektu

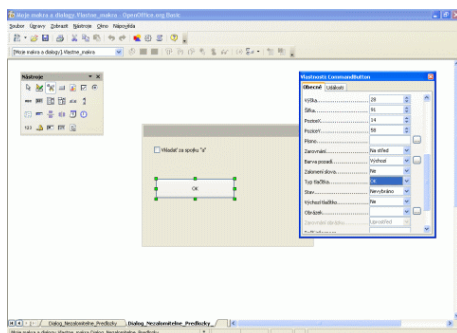
napr. tento text: „Vkladať za spojku „a““. Tým máme všetky potrebné parametre nastavené. Teraz podľa skutočne zobrazeného textu vhodne zmeníme veľkosť zadaného políčka, aby sme ho nemali zbytočne veľký, ale zase aby nebol ani príliš malý. Pokiaľ nám to nevádi, zobrazené údaje „Vlastnosti“ zatvárať nemusíme, aspoň ich budeme mať automaticky prístupné pre všetky ďalšie zadávané políčka, takže ich nebudeme musieť pracne označovať a následne toto okienko otvárať.



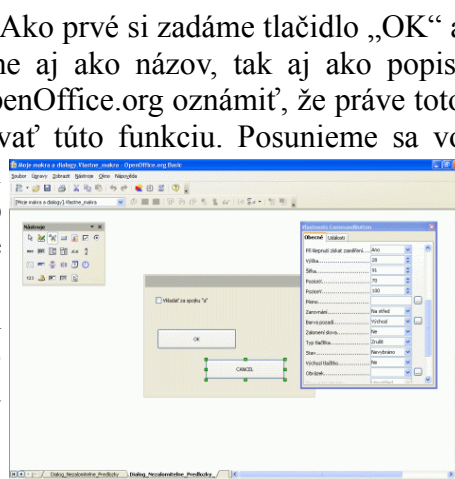
Obrázok 7: Ako druhé vložíme tlačidlo

Tlačidlá „OK“ a „Cancel“

Teraz vložíme do dialógového okna tlačidlá „OK“ a „Cancel“. V zozname nástrojov si vyberieme symbol tlačidla a následne nakreslíme primerane veľké tlačidlo do nášho návrhu. Vo vlastnostiach zmeníme znovu názov



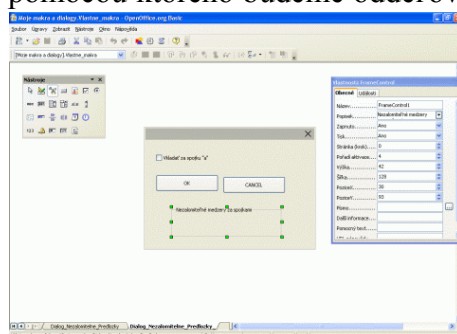
Obrázok 8: A nastavíme ho na "OK" a v položke „Typ tlačidla“ vyberieme hodnotu „OK“.



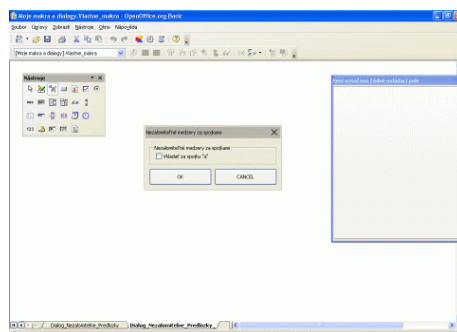
Obrázok 9: Nesmieme zabudnúť ani na tlačidlo "Cancel"

Okrasné a oddel'ujúce prvky

Ako posledný typ sme si povedali, že zadáme rámik, pomocou ktorého budeme oddel'ovať jednotlivé skupiny prvkov dialógového okna. V prípade tohto typu obvykle nemejme jeho názov, pretože sa naň prakticky nikde neodvoláva a nastavujeme iba popis, ktorý v stručnosti vystihuje orámované prvky dialógového okna. Nakoniec si označíme hlavné dialógové okno a zadáme jeho popis.



Obrázok 10: Ako posledný si vložíme ozdobný rámik

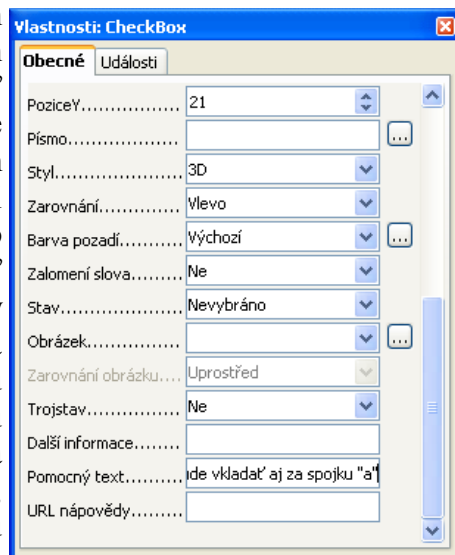


Obrázok 11: Dialógové okno nakoniec upravíme podľa nášho vkusu

„Pomocný text“ aj ľubovoľný nápovedný text, ktorý sa bude zobrazovať v žltom okienku, keď sa myšou premiestnime nad tento prvok, napr. k zaštrávaciemu okienku napíšeme takýto nápovedný text: „Nezalomitelná medzera sa bude vkladať aj za spojku "a"“. Pravdaže, tieto nápovedné texty môžeme pridať k ľubovoľnému prvku (teda aj k textovému rámkovi, tlačidlu „OK“, ...), čím môžeme pripraviť dialógové okno aj pre menej zdatných používateľov.

Celkové „kozmetické“ úpravy

Pretože máme zadané už všetky prvky a texty, prevedieme si teraz celkové rozmiestnenie tak, aby sme dosiahli nami požadovanú estetickú úroveň, t.j. jednotlivé prvky teraz budeme postupne označovať, premiestňovať a meniť ich veľkosť. Pokiaľ chceme, môžeme veľkostné údaje a umiestnenie zadávať aj číselne priamo v parametroch tlačidiel, takže návrh môžeme previesť naozaj veľmi detailne a podrobne. Okrem toho môžeme pri každom prvku v jeho vlastnostiach zadať do položky



Obrázok 12: Do pomocného textu môžeme zadať stručnú nápovedu

Základné funkcie pre prácu s dialógmi

V minulom dieli nášho seriálu sme si navrhli prvé dialógové okno, pomocou ktorého chceme nastavovať parameter pre riadenie vkladania nezalomiteľnej medzery za spojku „a“. Tento návrh vlastne spočíval z výberu vhodných položiek (políček, prvkov), ktoré sme pre náš dialóg požadovali a ich následnej grafickej úpravy vrátane vhodného textového sprievodu.

Pochopiteľne, takýto návrh musíme vždy urobiť ako prvý, pretože až na jeho základe môžeme s takto pripraveným dialógovým oknom pracovať. A práve na túto prácu – t.j. otvorenie dialógového okna, nastavovanie a čítanie údajov z jednotlivých políček až po jeho zatvorenie sa zameriame práve dnes.

Dialóg ako objekt v jazyku StarOffice Basic

Z hľadiska programovacieho jazyka StarOffice Basic je dialóg špecifický objekt s množinou preddefinovaných funkcií. Teraz sa musíme vrátiť k druhému dielu, v ktorom sme spomínali, že všetky dialógy máme vlastne uložené v nami zadaných knižniciach, citujem: „... Aj na tejto záložke nájdeme stromovú štruktúru jednotlivých knižníc, po rozbalení ktorých sa nám zobrazí zoznam dialógov, ktoré v nich máme definované. V organizátorovi máme teraz možnosť pridať názvy nových dialógov, ktoré chceme v budúcnosti definovať, takže napr. do knižnice "Vlastne_makra" si cez tlačidlo "Nový..." pridáme nový dialóg, ktorý nazveme napr. "Dialog_Nezalomitelne_Predlozky_"..."

Predpokladajme, že sme dodržali túto štruktúru. K nášmu dialógu sa potom dostaneme pomocou nasledujúcej postupnosti príkazov:

REM Definícia premennej, pomocou ktorej budeme pristupovať k nášmu dialógu

dim dlg as object

REM Sprístupnenie knižnice Vlastne_makra

DialogLibraries.LoadLibrary("Vlastne_makra")

REM Priradenie dialógu Dialog_Nezalomitelne_Predlozky_ do premennej dlg

dlg=CreateUnoDialog(DialogLibraries.Vlastne_makra.Dialog_Nezalomitelne_Predlozky_)

Keď už máme dialógové okno sprístupnené, môžeme ho otvoriť a zatvoriť. Zatvorenie je jednoduché, pretože sa môže urobiť kliknutím na štandardné zatváracie tlačidlo okien (de facto sa správa ako tlačidlo „Cancel“), alebo, ak sme definovali tlačidlá „OK“ a „Cancel“, tak kliknutím na jedno z nich. Pravdaže, v programe nás bude následne zaujímať, ako sme dialóg vlastne ukončili. Všetky tieto činnosti (t.j. otvorenie a oznámenie, ako sme dialógové okno zatvorili) nám zabezpečuje jedna jediná funkcia „Execute“, ktorá je súčasťou dialógu:

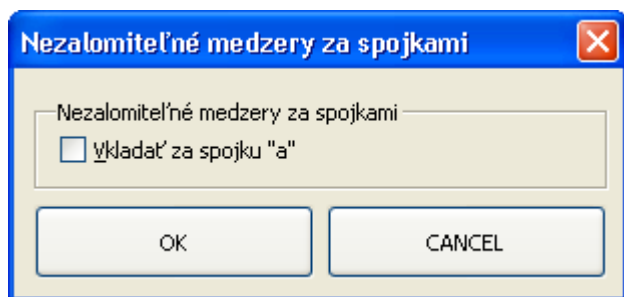
if dlg.Execute()=1 then

REM Dialógové okno sme ukončili klávesou „OK“

else ' nadobúda hodnotu 0

REM Dialógové okno sme ukončili klávesou „CANCEL“

endif



Obrázok 13: Takto vyzerá naše dialógové okno po jeho zavolaní

stále sme sa však nedostali k jednotlivým položkám (okrem tlačidiel „OK“ a „Cancel“), ktoré máme v dialógovom okne definované a ktoré nás, pochopiteľne, zaujímajú najviac. Všetky položky, ktoré sme do dialógového okna vkladali sú prístupné v časti „model“, za ktorým uvádzame ich názov. Ak máme dialógové okno prístupné (predpokladajme že v premennej `dlg`), k nami zadanému zaškrtávaciemu okienku „Vkladať za spojku a“, ktoré sme minule vo vlastnostiach nazvali „Za_spojku_a“ sa dostaneme nasledovne:

```
dlg.model.Za_spojku_a
```

Týmto spôsobom sa sprístupňuje celý rad parametroch a hodnôt, ktoré pre túto položku môžeme nastavovať. Zatiaľ sa však budeme zaoberať iba jednou z nich – jej hodnotou. Táto je následne prístupná cez položku „State“:

```
dlg.model.Za_spojku_a.State
```

Pochopiteľne, programovací jazyk StarOffice Basic má štandardne nastavené počiatočné hodnoty všetkých typov položiek, ktoré do dialógového okna môžeme vložiť. V prípade zaškrtávacieho okienka je to hodnota 0, t.j. okienko nie je zaškrtnuté. Vlastnému nastavovaniu a jeho následnému odpamätávaniu do pracovného súboru na disku sa však budeme venovať až neskôr.

Nastavovanie parametrov pomocou dialógu

V poslednom dieli nášho seriálu sme si stručne načrtli funkcie pre prácu s dialógmi. Na základe toho môžeme teraz pristúpiť k ďalšiemu kroku, v ktorom dialóg využijeme pre vlastné nastavenie parametra v našich podprogramoch ohľadne predložky „a“. Pretože všetky potrebné funkcie sme v nižších verziách už uviedli, môžeme priamo pristúpiť k vlastnému programovaniu bez toho, aby sme obzvlášť museli komentovať jednotlivé podprogramy.

Funkcie pre odpamätanie nastavení do súboru

```
global ObsluhaDokumentu, ObsluhaKlavesnice as object
```

```
global PosledneSlovo, OddelovaceSlov, JednoznakovePredlozky as string
```

```
global Priznak_nezalomitelna_a as integer
```

```
REM Funkcie pre prácu so súborom, kde sú nastavené príznaky
```

Pokiaľ už s dialógovým oknom nebudeme pracovať, musíme ho nakoniec uvoľniť z pamäte príkazom:

```
dlg.dispose()
```

To by z funkcií pre prácu s celým dialógovým oknom mohlo zatiaľ stačiť. Ešte

```

function Subor_nezalomitelne_predlozky as string
    Subor_nezalomitelne_predlozky="file:///c:/Program files/OpenOffice.org 2.0/Nezalomitel
na_Predlozka_a.dat"
end function

sub Init_zoznam_predloziek
    JednoznakovePredlozky="AiIkKoOsSuUvVzZ"
    if Priznak_nezalomitelna_a=1 then
        JednoznakovePredlozky="a"+JednoznakovePredlozky
    endif
end sub

sub Uloz_nastavenie_predlozky
    dim i, subor as integer

    subor = Freefile
    open Subor_nezalomitelne_predlozky for Output as #subor
    print #subor,str(Priznak_nezalomitelna_a)
    close #subor
end sub

sub Citaj_nastavenie_predlozky
    dim i, subor as integer
    dim riadok as string
    ' Inicializácia na štandardné štandardné hodnoty
    Priznak_nezalomitelna_a=1
    next i

    ' Ak nastavovací súbor neexistuje, vytvoríme ho a uložíme do neho štandardné hodnoty
    if not fileexists(Subor_nezalomitelne_predlozky) then
        Uloz_nastavenie_predlozky
    endif

    ' Teraz súbor určite existuje - načítame ho
    subor = Freefile

```

```

open Subor_nezalomitelne_predlozky for Input as #subor
i=0
if not eof(subor)
i=i+1
Line Input #subor, riadok
Priznak_nezalomitelna_a=val(riadok)
endif
close #subor

' Ak v súbore chýbal nejaký údaj, tak ho opravíme
if i<1 then
Uloz_nastavenie_predlozky
endif
Init_zoznam_predloziek

end sub

```

Funkcie pre vlastnú prácu s dialógovým oknom

Teraz si napíšeme funkcie pre vlastnú prácu s dialógovým oknom „Dialog_nezalomitelne_predlozky“. Pretože sme sa tomuto ešte nevenovali podrobne, budeme príslušné funkcie trochu viac komentovať.

REM Procedúra, pomocou ktorej nastavíme počiatočné hodnoty príslušných položiek dialógového okna

```
sub init_dialog_predlozky(oDialog as object)
```

```
REM Načítame ich zo súboru
```

```
Citaj_nastavenie_predlozky
```

REM Nastavenie počiatočnej hodnoty podľa toho, ako ju máme odpamätanú v inicializačnom súbore

```
oDialog.model.Za_spojku_a.State=Priznak_nezalomitelna_a
```

```
end sub
```

REM Procedúra, pomocou ktorej zapíšeme získané hodnoty z príslušných položiek dialógového okna do inicializačného súboru

```
sub zapis_dialog_predlozky(oDialog as object)
```

```
Priznak_nezalomitelna_a=oDialog.model.Za_spojku_a.State
```

```
Uloz_nastavenie_predlozky
```

REM Po uložení do súboru je vhodné, aby sme si zmenené nastavenie aj zmenili v globálnych premenných

```
Init_zoznam_predloziek
```

```
end sub
```

REM Vlastné volanie dialógu s príslušnými nastaveniami

```
sub Nastav_Priznak_Nezalomitelne_a
```

```
dim dlg as object
```

REM Sprístupnenie knižnice

```
DialogLibraries.LoadLibrary("Vlastne_makra")
```

REM Sprístupnenie dialógového okna

```
dlg=CreateUnoDialog(DialogLibraries.Vlastne_makra.Dialog_Nezalomitelne_Predlozky_  
)
```

REM Inicializácia počiatočných hodnôt

```
init_dialog_predlozky(dlg)
```

REM Vlastné volanie dialógového okna

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

REM Ak sme ukončili tlačidlom „OK“, tak zmenené hodnoty zapíšeme

```
zapis_dialog_predlozky(dlg)
```

```
endif
```

REM Uvoľnenie dialógového okna z pamäte

```
dlg.dispose()
```

```
end sub
```

Vlastné vkladanie nezalomiteľných medzier počas ich písania

Celé nastavenie vykonáme zavolaním poslednej procedúry „Nastav_Priznak_Nezalomitelne_a“. Teraz už môžeme pridať procedúry pre vkladanie nezalomiteľných medzier za jednoznakové predložky počas ich písania. Na rozdiel od už uverejnenej verzie budeme teraz nastavovať zoznam predložiek podľa inicializačného súboru (volanie funkcie „Citaj_nastavenie_predlozky,,“).

```
sub SpustiObsluhuKlavesnice
```

```
on error goto next
```

```
' Inicializácia premenných
```

```
Citaj_nastavenie_predlozky
```

```
PosledneSlovo = ""
```

```
' Znak, ktoré ukončujú slová – medzera, nezalomiteľná medzera, tabulátor, CR, LF
```

```
OddelovaceSlov="" +chr(&HA0)+chr(&H09)+chr(&H0A)+chr(&H0D)
```

```
' pomlčka, lomítko, bodka, čiarka, výkričník, otáznik, bodkočiarka, trojbodka
```

```
OddelovaceSlov=OddelovaceSlov+"-./,!?;..."
```

```
' zátvorky
```

```
OddelovaceSlov=OddelovaceSlov+"()[]{}"
```

```
' Aktuálny dokument
```

```
ObsluhaDokumentu = ThisComponent.getCurrentController
```

```
' Vlastná obsluha klávesnice
```

```
ObsluhaKlavesnice = createUnoListener("Klavesnica_", "com.sun.star.awt.XKeyHandler")
```

```
' Priradenie tejto obsluhy k aktuálnemu dokumentu
```

```
ObsluhaDokumentu.addHandler(KeyHandler(ObsluhaKlavesnice))
```

```
end sub
```

```
sub UkonciObsluhuKlavesnice
```

```
on error goto next
```

```
ObsluhaDokumentu.removeKeyHandler(ObsluhaKlavesnice)
```

```
PosledneSlovo = ""
```

```
end sub
```

```
sub VlozZnak(Retazec as string)
```

```
' Na aktuálnu pozíciu v dokumente vložíme reťazec
```

```
dim VidelnyKurzor, AktualnyText, AktualnyKurzor as object
```

```
ViditelnyKurzor = ThisComponent.getCurrentController().getViewCursor()  
AktualnyText = ThisComponent.getText()  
AktualnyKurzor = AktualnyText.createTextCursorByRange(ViditelnyKurzor.getStart())  
AktualnyText.insertString(AktualnyKurzor.getStart(), Retazec, true)
```

```
end sub
```

```
function Klavesnica_KeyPressed(StlacenaKlavesa) as boolean
```

```
' Obsluha stlačenej klávesy
```

```
dim Obsluzena as boolean
```

```
'Predpokláame, že neobsľúžime klávesnicu
```

```
Obsluzena=False
```

```
if instr(OddelovaceSlov,StlacenaKlavesa.KeyChar)<>0 then
```

```
' Došli sme na koniec slova
```

```
if ZistiPredlozku and (StlacenaKlavesa.KeyChar=" ") then
```

```
' Je to predložka, ktorá končí zalomiteľnou medzerou - túto nahradíme
```

```
VlozZnak(chr$(&HA0)) ' No Break Space
```

```
' Sami sme vložili nezalomiteľnú medzeru - obsľúžili sme stlačenú klávesu
```

```
Obsluzena=true
```

```
endif
```

```
' Boli sme na konci slova, už nás nebude zaujímať
```

```
PosledneSlovo=""
```

```
else
```

```
' Neboli sme na konci slova
```

```
' Ešte musíme otestovať riadiace znaky, ktoré nebudeme sledovať v zadávanom slove
```

```
select case StlacenaKlavesa.KeyChar
```

```
case chr$(&H00) to chr$(&H07)
```

```
PosledneSlovo=""
```

```
case chr$(&H08) ' BackSpace
```

```
if len(PosledneSlovo)>0 then
```

```
PosledneSlovo=left(PosledneSlovo,len(PosledneSlovo)-1)
```

```
else
```

```
PosledneSlovo=""
```

```
endif
```

```

case chr(&H09) to chr(&H1F)
    PosledneSlovo=""
case chr(&H20) to chr(&H7E)
    PosledneSlovo=PosledneSlovo+StlacenaKlavesa.KeyChar
case chr(&H7F) ' CTRL BS
    PosledneSlovo=""
case else
    PosledneSlovo=PosledneSlovo+StlacenaKlavesa.KeyChar
end select
endif
Klavesnica_KeyPressed=Obsluzena
end function

function Klavesnica_KeyReleased(StlacenaKlavesa) as boolean
' Obsluha pustenej klávesy
Klavesnica_KeyReleased = False
end function

function ZistiPredlozku as boolean
dim JePredlozka as boolean

JePredlozka=false
if len(PosledneSlovo)=1 then
' Ak je to jednoznakové slovo, tak to môže byť predložka
JePredlozka=instr(1,JednoznakovePredlozky,PosledneSlovo,0)
endif
ZistiPredlozku=JePredlozka

end function

```

Pružná nezalomiteľná medzera

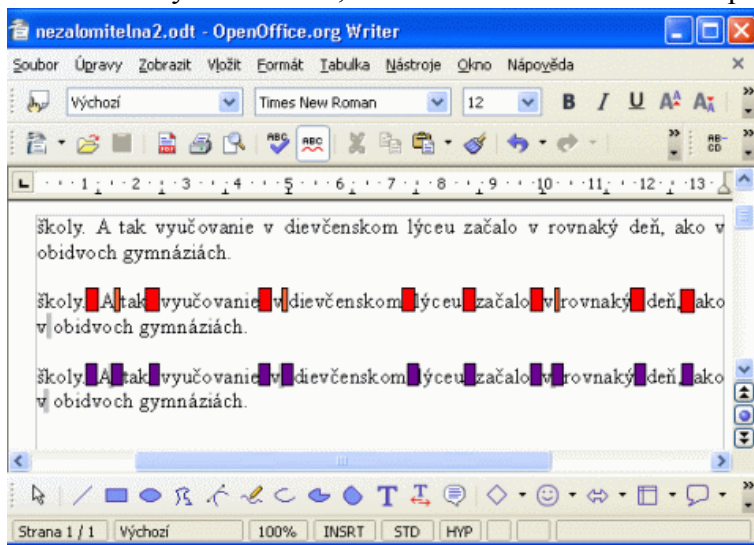
Hneď na úvod si zopakujme, čo to vlastne nezalomiteľná medzera je. Podľa slovenských a českých typografických pravidiel sa na konci riadku nemajú vyskytovať jednoznakové predložky a spojky (a, i, k, o, s, u, v, z). Z hľadiska práce na počítači je preto potrebné za ne vložiť vhodný znak (nezalomiteľnú medzeru), ktorá zabezpečí, že ak sa vyskytnú na konci riadku, tak sa automaticky zaradia na riadok nový.

Pokiaľ používame zarovnanie textu do bloku (a toto zarovnanie sa používa veľmi často), textový editor musí dosiahnuť, že slová budú spolu a zároveň rozťahnuté na celý riadok. To sa

dá dosiahnuť iba jedným spôsobom – „roztiahnutím“ medzier. Pravdaže, každý textový editor, ktorý umožňuje zarovnávať text do bloku to aj robí. Jedná sa vlastne o veľmi jednoduchú „matematiku“ stačí zistiť, koľko voľného priestoru medzi slovami máme a vydeliť ho počtom medzier. Pravdaže, v praxi sa stane, že toto číslo nebude celé, tak program potom upraví jednotlivé medzery tak, že niektoré urobí o jeden bod väčšie.

Čo na to hovorí matematika

Matematicky vynikajúce riešenie... pokiaľ však nepoužijeme nezalomiteľné medzery. Pokiaľ si dobre všimnete, žiadny z textových editorov tieto medzery nezväčšuje, t. j. ponecháva im konštantnú veľkosť a zväčšuje iba „obyčajné“ medzery. Týmto spôsobom sa však text značne deformuje a mnohokrát pôsobí opticky veľmi disharmonicky. Pravdaže, mnohým to nevaďí, ale ak budete čítať desiatky strán textu, oči sa omnoho skôr unavia pri nerovnomernom rozložení textu. Na nasledujúcom obrázku je použitý príklad, kde v prvom riadku vidíte obyčajné medzery, v druhom nezalomiteľné a v treťom pružne roztiahnuté nezalomiteľné medzery. Na tomto mieste upozorňujem, že sa nejedná o vymyslený text, ale o príklad z rukopisu románu s presne nastavenými okrajmi, ktorý spolu s manželkou vydávame ku koncu tohto roku. Myslím si, že farebné rozdiely vyjadrujúce veľkosť príslušných medzier sú dostatočne výrazné.



Obrázok 14: Rozdiely medzi rôznymi typmi medzier

A čo na to hovoria kódy znakov

Dost' však bolo teórie, ale pozrime sa na to, ako sa dá dosiahnuť „pružná nezalomiteľná“ medzera. Po dlhom hľadaní som v zozname unikódov (verzia 4.1) našiel znak, ktorý umožnil do textu vkladať takéto medzery. Prísne vzaté, nejedná sa o žiadnu medzeru, ale o znak „Word Joiner“ (hexadecimálny kód 2060 – v prípade kódov budeme väčšinou používať iba hexadecimálne kódy), ktorý vlastne do textu vkladá nezalomiteľnú medzeru nulovej veľkosti, čím dokáže spojiť dve za sebou idúce slová. Žiaľ, mnoho programov nedokáže dobre zobrazovať tieto unikódy. Medzi tie, ktoré ho zobrazujú správne patrí okrem OpenOffice.org napr. Adobe Reader, Foxit Reader, Tomahawk, Mozilla Firefox, GSView a pod. Medzi tie, ktoré ho správne nezobrazujú patria programy MS Office, Opera a pod. Aby ste si mohli odskúšať, ako sa zobrazuje tento znak vo vami obľúbených (či neobľúbených) programoch vrátane internetového prehliadača, príkladám teraz tri riadky textu. V prvom sú použité iba obyčajné medzery, v druhom iba nezalomiteľné a v treťom znak „Word Joiner“ spolu s obyčajnou medzerou za ním.

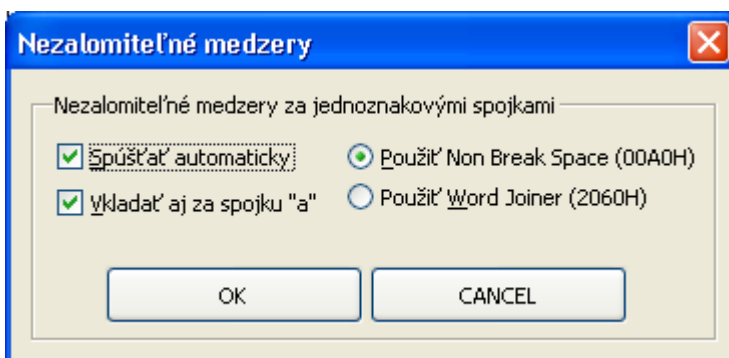
V tomto riadku sú všetky medzery obyčajné.

V tomto riadku sú všetky medzery nedeliteľné.

V tomto riadku je pred každou medzerou znak „Word Joiner“.

ktorým budeme nastavovať aj tento parameter. Celkové dialógové okno môže potom po spustení vyzeráť napríklad takto:

Teraz nám už nič nebráni v tom, aby sme si naprogramovali jednotlivé funkcie pre vlastnú prácu s dialógovým oknom a vkladaním nezalomiteľných medzier. Pravdaže, vzhľadom na väčší počet nastavovaných parametrov je potrebné vymazať príslušný dátový súbor, v ktorom máme zapamätané doterajšie nastavenia (však sa hneď pri prvom spustení obnoví):



Obrázok 16: Záverečný vzhľad dialógového okna pre nastavenie parametrov nezalomiteľných medzier

global ObsluhaDokumentu, ObsluhaKlavesnice as object

global PosledneSlovo, OddelovaceSlov, JednoznakovePredlozky as string

global Priznaky_nezalomitelne_medzery(3) as integer

global Obsluhovat_Klavesnicu as boolean

function Subor_nezalomitelne_predlozky as string

Subor_nezalomitelne_predlozky="file:///c:/Program files/OpenOffice.org 2.0/Nezalomitel
ne_Predlozky.dat"

end function

sub Init_zoznam_predloziek

JednoznakovePredlozky="AiIkKoOsSuUvVzZ"

if Priznaky_nezalomitelne_medzery(2)=1 then

JednoznakovePredlozky="a"+JednoznakovePredlozky

endif

end sub

sub Uloz_nastavenie_predlozky

dim i, subor as integer

subor = Freefile

open Subor_nezalomitelne_predlozky for Output as #subor

for i=1 to 3

print #subor,str(Priznaky_nezalomitelne_medzery(i))

next i

```

close #subor
end sub

sub Citaj_nastavenie_predlozky
dim i, subor as integer
dim riadok as string
' Inicializácia na štandardné štandardné hodnoty
for i=1 to 3
    Priznaky_nezalomitelne_medzery(i)=1
next i

' Ak nastavovací súbor neexistuje, vytvoríme ho a uložíme do neho štandardné hodnoty
if not fileexists(Subor_nezalomitelne_predlozky) then
    Uloz_nastavenie_predlozky
endif

' Teraz súbor určite existuje - načítame ho
subor = Freefile
open Subor_nezalomitelne_predlozky for Input as #subor
i=0
While not eof(subor)
    i=i+1
    Line Input #subor, riadok
    Priznaky_nezalomitelne_medzery(i)=val(riadok)
wend
close #subor

' Ak v súbore chýbal nejaký údaj, tak to opravíme
if i<3 then
    Uloz_nastavenie_predlozky
endif
Init_zoznam_predloziek
end sub

sub init_dialog_predlozky(oDialog as object)
    Citaj_nastavenie_predlozky

```

```

oDialog.model.Automaticky.State=Priznaky_nezalomitelne_medzery(1)
oDialog.model.Za_spojku_a.State=Priznaky_nezalomitelne_medzery(2)
oDialog.model.TabIndex=3
oDialog.model.TabIndex=4
select case Priznaky_nezalomitelne_medzery(3)
case 2
oDialog.model.Volba1.State=0
oDialog.model.Volba2.State=1
case else
oDialog.model.Volba1.State=1
oDialog.model.Volba2.State=0
end select
end sub

```

```

sub zapis_dialog_predlozky(oDialog as object)
Priznaky_nezalomitelne_medzery(1)=oDialog.model.Automaticky.State
Priznaky_nezalomitelne_medzery(2)=oDialog.model.Za_spojku_a.State
if oDialog.model.Volba1.State=1 then
Priznaky_nezalomitelne_medzery(3)=1
else
Priznaky_nezalomitelne_medzery(3)=2
endif
Uloz_nastavenie_predlozky
' A ešte podľa nového nastavenia zmeníme zoznam predložiek
Init_zoznam_predloziek
end sub

```

```

sub NastavPriznakyNezalomitelnePredlozky
dim dlg as object

```

```

DialogLibraries.LoadLibrary("Vlastne_makra")

```

```

dlg=CreateUnoDialog(DialogLibraries.Vlastne_makra.Dialog_Nezalomitelne_Predlozky)
init_dialog_predlozky(dlg)

```

```

if dlg.Execute()=1 then

```

```

    zapis_dialog_predlozky(dlg)
endif
dlg.dispose()
end sub

sub SpustiObsluhuKlavesnice
on error goto next
if ThisComponent.supportsService("com.sun.star.text.TextDocument") then
' Sme v textovom dokumente, kde chceme nahrádzať medzery
Obsluhovat_Klavesnicu=True
PosledneSlovo = ""
' medzera, nezalomiteľná medzera, tabulátor, CR, LF
OddelovaceSlov=" "+chr(&HA0)+chr(&H09)+chr(&H0A)+chr(&H0D)
' pomlčka, lomítko, bodka, čiarka, výkričník, otáznik, bodkočiarka, trojbodka
OddelovaceSlov=OddelovaceSlov+"-./!?:;..."
' zátvorky
OddelovaceSlov=OddelovaceSlov+"()[]{}"
ObsluhaDokumentu = ThisComponent.getCurrentController
ObsluhaKlavesnice = createUnoListener("Klavesnica_", "com.sun.star.awt.XKeyHandler")
ObsluhaDokumentu.addHandler(KeyHandler(ObsluhaKlavesnice))
endif
end sub

sub UkonciObsluhuKlavesnice
on error goto next
if ThisComponent.supportsService("com.sun.star.text.TextDocument") then
' Sme v textovom dokumente a chceme ukončiť obsluhu klávesnice
ObsluhaDokumentu.removeKeyHandler(ObsluhaKlavesnice)
PosledneSlovo = ""
endif
end sub

function VlozZnak(Retazec as string)
' Na aktuálnu pozíciu v dokumente vložíme reťazec
dim ViditelnyKurzor, AktualnyText, AktualnyKurzor as object

```

```

ViditelnyKurzor = ThisComponent.getCurrentController().getViewCursor()
AktualnyText = ThisComponent.getText()
AktualnyKurzor = AktualnyText.createTextCursorByRange(ViditelnyKurzor.getStart() )
AktualnyText.insertString(AktualnyKurzor.getStart(), Retazec, true)

```

end function

```

function Klavesnica_KeyPressed(StlacenaKlavesa) as boolean
' Obsluha stlačenej klávesy
dim Obsluzena as boolean

'Predpokláame, že nebudeme vkladať nezaľomiteľnú medzeru
Obsluzena=False
if instr(OddelovaceSlov,StlacenaKlavesa.KeyChar)<>0 then
' Došli sme na koniec slova
if Obsluhovat_Klavesnicu then
if ZistiPredlozku and (StlacenaKlavesa.KeyChar=" ") then
' Je to predložka, ktorá končí zaľomiteľnou medzerou - túto nahradíme
if Priznaky_nezaľomitelne_medzery(3)=1 then
VlozZnak(chr$(&HA0)) ' No Break Space
' Sami sme vložili medzeru - obslúžili sme stlačenú klávesu
Obsluzena=true
else
VlozZnak(chr$(&H2060)) ' Word Joiner
' V tomto prípade sa musí ešte vložiť aj zadaná medzera a preto sme neobslúžili stlačenú
klávesu
endif
endif
endif
PosledneSlovo=""
else
' Ešte musíme otestovať riadiace znaky, ktoré nebudeme sledovať v zadávanom slove
select case StlacenaKlavesa.KeyChar
case chr$(&H00) to chr$(&H07)
PosledneSlovo=""
case chr$(&H08) ' BS

```

```

if len(PosledneSlovo)>0 then
    PosledneSlovo=left(PosledneSlovo,len(PosledneSlovo)-1)
else
    PosledneSlovo=""
endif
case chr$(&H09) to chr$(&H1F)
    PosledneSlovo=""
case chr$(&H20) to chr$(&H7E)
    PosledneSlovo=PosledneSlovo+StlacenaKlavesa.KeyChar
case chr$(&H7F) ' CTRL BS
    PosledneSlovo=""
case else
    PosledneSlovo=PosledneSlovo+StlacenaKlavesa.KeyChar
end select
endif
Klavesnica_KeyPressed=Obsluzena
end function

function Klavesnica_KeyReleased(StlacenaKlavesa) as boolean
' Obsluha pustenej klávesy
Klavesnica_KeyReleased = False
end function

function ZistiPredlozku as boolean
dim JePredlozka as boolean

JePredlozka=false
if len(PosledneSlovo)=1 then
    ' Ak je to jednoznakové slovo, tak to môže byť predložka
    JePredlozka=instr(1,JednoznakovePredlozky,PosledneSlovo,0)
endif
ZistiPredlozku=JePredlozka

end function

```

Inicializácia prepínačov

Teraz k vlastným prepínačom. Inicializovali sme ich týmito dvomi príkazmi:

```
oDialog.model.TabIndex=3
```

```
oDialog.model.TabIndex=4
```

OpenOffice.org sa správa tak, že ak uvedieme bezprostredne za sebou bez prerušenia iným príkazom zoznam prepínačov, považuje ich za jeden zoznam a bude prepínať automaticky medzi nimi, t.j. označením jednej položky zruší označenie inej. K uvedeným funkciám ešte musíme naprogramovať aj podporné obslužné programy, resp. ich priradiť k akciám ako je otvorenie či zatvorenie súboru, ale tomu všetkému sa budeme venovať až na budúce.

Ako si v tom udržať poriadok a prehľad

V minulom dieli sme si ukázali pomerne rozsiahle funkcie pre prácu s dialógovým oknom, v ktorom nastavujeme parametre pre vkladanie nezalomiteľných medzier za jednoznakové spojky a predložky. Pravdaže, priame volanie týchto funkcií nie je práve pohodlné a okrem toho sme si do dialógového okna zaradili nový parameter – „Spúšťať automaticky pri otvorení dokumentu“. Táto funkcia je určite mnohými požadovaná, pretože sa nechcú zaoberať tým, aby po otvorení dokumentu ešte manuálne museli spúšťať nejaké makro pre vkladanie (zámenu) medzier.

Preto bude vhodné, ak si všetky funkcie, ktoré sme si uviedli minule uložili do osobitného modulu „Nezalomitelne_predlozky_funkcie“ a dnes si vytvoríme nový modul „Nezalomitelne_predlozky“, do ktorého si zadáme iba jednoduché programy, ktoré budú už príslušné kompletne funkcie v ostatných moduloch volať za nás:

```
sub Nastav_priznaky_nezalomitelne_predlozky
```

```
Nezalomitelne_predlozky_funkcie.NastavPriznakyNezalomitelnePredlozky
```

```
end sub
```

```
sub Automaticky_Obsluhuj_Klavesnicu
```

```
Nezalomitelne_predlozky_funkcie.Citaj_nastavenie_predlozky
```

```
if Nezalomitelne_predlozky_funkcie.Priznaky_nezalomitelne_medzery(1)=1 then
```

```
Nezalomitelne_predlozky_funkcie.SpustiObsluhuKlavesnice
```

```
endif
```

```
end sub
```

```
sub Obsluhuj_Klavesnicu
```

```
Nezalomitelne_predlozky_funkcie.Citaj_nastavenie_predlozky
```

```
Nezalomitelne_predlozky_funkcie.SpustiObsluhuKlavesnice
```

```
end sub
```

```

sub Neobsluhuj_Klavesnicu
    Nezalomitelne_predlozky_funkcie.UkonciObsluhuKlavesnice
end sub

sub Povol_Klavesnicu
    Nezalomitelne_predlozky_funkcie.Obsluhovat_Klavesnicu=True
end sub

sub Pozastav_Klavesnicu
    Nezalomitelne_predlozky_funkcie.Obsluhovat_Klavesnicu=False
end sub

```

Automatické spúšťanie počas otvárania súboru

Pre tých, ktorí potrebujú spúšťať automatické nahrádzanie medzier už počas otvárania súborov je tu pripravená funkcia „Automaticky_Obsluhuj_Klavesnicu“ a funkcia „Neobsluhuj_Klavesnicu“. Práve pre nich je určená aj zaujímavá dvojica funkcií „Povol_Klavesnicu“ a „Pozastav_Klavesnicu“. Tieto funkcie sú určené na to, aby sa dalo občas (z ľubovoľného dôvodu) na určitý čas prerušiť (funkcia „Pozastav_Klavesnicu“) a znovu povoliť (funkcia „Povol_Klavesnicu“) automatické nahrádzanie. Všetci ostatní používatelia si tu napokon nájdu dvojicu funkcií „Obsluhuj_Klavesnicu“ a „Neobsluhuj_Klavesnicu“.

Pravdaže, ešte sme sa nezmienili o tom, ako vlastne zabezpečiť automatické spúšťanie obsluhy klávesnice. Za týmto účelom musíme v dialógovom okne „Nástroje-Prispôbiť“ na záložke „Udalosti“ priradiť k položke „Vytvoriť dokument“ a k položke „Otvoriť dokument“ volanie makra „Automaticky_Obsluhuj_Klavesnicu“. Zároveň je nutné, aby sme k položke „Zatvoriť dokument“ priradili volanie makra „Neobsluhuj_Klavesnicu“.

Niekoľko nedostatkov a chýb

Na tomto mieste musíme upozorniť na niektoré chyby, ktoré uvedené makrá ešte stále majú. Používajú sa v nich globálne premenné a preto, ak spustíte nahrádzanie vo viacerých dokumentoch, bude to viesť ku chybe. Pri manuálnom spustení nahrádzania je ďalej potrebné toto pred uzatvorením dokumentu ukončiť, pretože to vedie ku chybe, že chceme uzatvoriť dokument, v ktorom máme spustené nejaké makro. Podobne, ak si otvoríte niekoľko dokumentov a v jednom povolíte nahrádzanie, je pred uzatvorením ktoréhokolvek z nich v tomto dokumente nahrádzanie ukončiť a až potom prípadne znovu povoliť.

Viem, že tieto chyby môžu na prvý pohľad vyzerat' hrôzostrašne. Žiaľ, zatiaľ sa mi nepodarilo prísť na to, ako ich odstrániť. Napriek tomu však verím, že uvedené makrá mnohým pomôžu aj v takom tvare, v akom sú dané k dispozícii, pretože nie každý pracuje naraz s viacerými dokumentami.

Niekoľko slov na záver ...

Týmto dielom sme ukončili základnú časť práce s dialógmi a tým vlastne aj zo základmi programovania makier v OpenOffice.org. Pred skoro trištvrte rokom sme začali tento seriál s trochu neskromným cieľom – naučiť sa programovať makrá. Ak si pamätáte, začali sme „nahrávaním“ makra pre automatické vkladanie textu a dnes končíme pomerne rozsiahlym komplexom programov, v ktorom nájdete aj automatické nahrádzanie medzier už počas ich písania. Viem, že cez internet sa ťažko učí programovať. Ak sa však v tomto „školskom“ roku našiel aspoň jeden spokojný používateľ, ktorý si na to „trúfol“, splnili sme cieľ, ktorý sme si na začiatku položili. Je 29.6.2006 a pozajtra začnú aj mojím deťom školské prázdniny. Oddýchnime si, načerpajme sily a začiatkom septembra sa znovu teším na stretnutie so všetkými „programovaniachtivými“ používateľmi OpenOffice.org.