

Doporučuje časopis

PC WORLD



příklady na
www.grada.cz

OpenOffice.org

tipy a triky pro záznam a úpravu maker

Milan Sobek

- Seznámení s jazykem Basic
- Práce s moduly a knihovnamí
- Proměnné, funkce a příkazy
- Vývojové prostředí IDE
- Aplikační programovací rozhraní
- Práce s tabulkovými dokumenty



GRADA

Upozornění pro čtenáře a uživatele této knihy

Všechna práva vyhrazena. Žádná část této tištěné či elektronické knihy nesmí být reprodukována a šířena v papírové, elektronické či jiné podobě bez předchozího písemného souhlasu nakladatele. Neoprávněné užití této knihy bude **trestně stíháno**.

Používání elektronické verze knihy je umožněno jen osobě, která ji legálně nabyla a jen pro její osobní a vnitřní potřeby v rozsahu stanoveném autorským zákonem. Elektronická kniha je datový soubor, který lze užívat pouze v takové formě, v jaké jej lze stáhnout s portálu. Jakékoliv neoprávněné užití elektronické knihy nebo její části, spočívající např. v kopírování, úpravách, prodeji, pronajímání, půjčování, sdělování veřejnosti nebo jakémkoliv druhu obchodování nebo neobchodního šíření je zakázáno! Zejména je zakázána jakákoliv konverze datového souboru nebo extrakce části nebo celého textu, umístování textu na servery, ze kterých je možno tento soubor dále stahovat, přitom není rozhodující, kdo takovéto sdílení umožnil. Je zakázáno sdělování údajů o uživatelském účtu jiným osobám, zasahování do technických prostředků, které chrání elektronickou knihu, případně omezují rozsah jejího užití. Uživatel také není oprávněn jakkoliv testovat, zkoušet či obcházet technické zabezpečení elektronické knihy.





Copyright © Grada Publishing, a.s.

Obsah

Úvod.....	11
Část I: Práce s jazykem Basic.....	15
1. Naše první makro	17
1.1 Co je to makro, jak vzniká a k čemu slouží.....	17
1.2 Vytvoření makra s využitím záznamníku maker	18
1.3 Makro „Zdravím svět“	22
2. Základní informace o makrech.....	25
2.1 Makro jako základní programovací jednotka	25
2.2 Tvorba maker, jejich uložení a spouštění	26
2.3 Uživatelské dialogy	26
3. Práce s moduly a knihovnami.....	27
3.1 Vytváření a mazání knihoven a modulů	29
3.2 Kopírování knihovny mezi dvěma dokumenty.....	30
3.3 Kopírování knihovny mezi dokumentem a aplikačním knihovním kontejnerem.....	31
4. Forma zápisu procedur	33
4.1 Řádkování	33
4.2 Mezery	34
4.3 Poznámky	34
4.4 Velká a malá písmena.....	34
4.5 Názvy tvořené uživatelem	34

4.6 Příkazy	34
4.7 Funkce	35
5. Proměnné	37
5.1 Názvy proměnných	38
5.2 Typ proměnných	38
5.3 Deklarace proměnných	41
5.4 Určení typu proměnných	42
5.5 Práce s proměnnými různého typu	43
5.6 Záměna typu proměnných	44
5.7 Funkce pro práci s typy proměnných a s typy hodnot	44
5.7.1 Určení typu proměnných	45
5.7.2 Určení typu hodnot uložených v proměnných	47
5.7.3 Převod typu hodnot uložených v proměnných	48
5.7.4 Platnost proměnných	48
5.7.5 Vztah globálních a lokálních proměnných	49
5.7.6 Operace s proměnnými	50
5.7.7 Pole proměnných	51
6. Funkce a příkazy	53
6.1 Komunikace s uživatelem	54
6.1.1 Příkaz Print	54
6.1.2 Příkaz/funkce MsgBox	56
6.1.3 Funkce InputBox	58
6.2 Řízení chodu makra	58
6.2.1 Příkaz If ... Elseif ... Else ... End If	58
6.2.2 Příkaz Select Case ... Case ... End Select	60
6.2.3 Příkaz Do ... Loop	62
6.2.4 Příkaz For ... To ... Step ... Next	64
6.2.5 Příkaz skoku GoTo a příkaz k ukončení makra Exit	65
6.3 Práce s textovými řetězci	67
6.4 Datové a časové funkce	70

6.5	Číselné funkce	73
6.5.1	Trigonometrické funkce	73
6.5.2	Exponenciální funkce	73
6.5.3	Celá čísla.....	73
6.5.4	Ostatní funkce	73
6.6	Převod typu proměnných	74
6.7	Zpracování chyb.....	75
6.8	Ostatní příkazy a funkce.....	77
7.	Spolupráce maker.....	79
7.1	Makro jako příkaz nebo funkce	79
7.2	Předávání proměnných mezi makry	81
7.3	Povinné a nepovinné parametry	82
8.	Vývojové prostředí IDE	85
8.1	Pohyb v projektu	86
8.2	Práce s textem.....	86
8.3	Uložení modulu jako samostatného souboru	87
8.4	Ovládání chodu makra	87
8.4.1	Spustit BASIC.....	87
8.4.2	Zkompilovat	87
8.4.3	Zastavit makro.....	87
8.4.4	Krok přes, dovnitř, ven.....	87
8.4.5	Bod přerušení zap/vyp	88
8.4.6	Správa bodů přerušení.....	88
8.5	Kukátko	89
8.6	Přiřazení makra události.....	89
8.7	Přiřazení makra klávese	90
8.8	Přiřazení makra k tlačítku.....	91

Část II: Práce s tabulkovým dokumentem	95
9. Popis API	97
9.1 Objekty v API	98
9.2 Uspořádání informací.....	98
9.3 Prvky použité v API	99
9.3.1 Moduly	99
9.3.2 Služby.....	99
9.3.3 Rozhraní.....	99
9.3.4 Výčtové typy a skupiny konstant	99
9.3.5 Struktury	100
9.3.6 Sekvence.....	100
9.4 Kolekce prvků	101
9.5 Zápis textu do buňky v listu	101
10. Informace o prvcích API.....	103
10.1 Určení typu proměnné.....	103
10.2 Informace o objektu.....	104
10.3 Podporované služby	107
11. Tabulkové dokumenty.....	109
12. Listy tabulkového dokumentu	117
12.1 Kontejner listů	117
12.2 Vlastnosti listů	119
12.3 Práce s vlastnostmi objektů	121
13. Buňky, oblasti buněk	123
13.1 Jednotlivé buňky	123
13.1.1 Získání objektu buňky.....	123
13.1.2 Umístění buňky.....	124
13.1.3 Hodnoty v buňkách	125
13.1.4 Vlastnosti buněk	126

13.2 Oblasti buněk	127
13.2.1 Objekty oblastí	127
13.2.2 Tabulkové funkce	132
13.2.3 Řazení buněk v oblasti	132
13.2.4 Kopírování a přesun oblastí	133
13.2.5 Vyhledávání v oblasti a náhrada výrazů	134
13.2.6 Filtrace dat	136
13.2.7 Využitá oblast.....	139
13.2.8 Zobrazená oblast	139
14. Uživatelská dialogová okna.....	141
14.1 Vytvoření uživatelského dialogu	141
14.2 Ovládání uživatelského dialogu.....	142
15. Události v dokumentu	145
15.1 Registrace událostí v dokumentu.....	145
15.2 Zpracování událostí.....	146
Závěr	149
Rejstřík	150

Úvod

Příručka je určena pro pokročilejší uživatele kancelářského balíku OpenOffice.org, kteří již zvládli základní dovednosti nezbytné pro práci s tímto programem a s jeho tabulkovým procesorem. Poskytuje informace o vytváření maker, s jejichž pomocí lze možnosti a uplatnění tohoto programu podstatně rozšířit. **Makra** umožňují automatizovat rutinní operace, usnadňují zpracování a vyhodnocování dat. Významná je i možnost účinné kontroly. Předností maker oceníme zejména při práci s rozsáhlejšími datovými soubory.

V příručce se budeme věnovat tabulkovému procesoru OpenOffice.org Calc a tvorbě maker s využitím programovacího jazyka OpenOffice.org Basic. Předpokládáme práci v prostředí Windows.

Makra představují vždy pouze určitý doplněk základního programu a mohou pracovat pouze s tímto programem. OpenOffice.org Basic neumožňuje vytvářet samostatné programy, které by pracovaly mimo rámec balíku OpenOffice.org.

Kniha u čtenáře nepředpokládá předchozí zkušenosti s programováním. V jednotlivých kapitolách jsou postupně vysvětleny základní pojmy a programovací postupy. Výklad je doplněn praktickými příklady, na nichž je možno nové poznatky ověřit. Makra uváděná v příkladech jsou také k dispozici ve formě tabulkových dokumentů.

OpenOffice.org je rozsáhlá aplikace, která zahrnuje několik programů vhodných zejména pro kancelářské využití. I když všechny programy mají řadu prvků společných, každý z nich je zaměřen na určitou oblast, která představuje jeho silnou stránku. U tabulkového procesoru je to především práce se soubory hodnot, které jsou často uspořádány ve formě tabulek.

Když spustíme aplikaci, tak prvním krokem bývá otevření již existujícího dokumentu nebo vytvoření dokumentu nového, v našem případě tabulkového. S dokumentem můžeme začít hned pracovat, protože program k tomu poskytuje celou řadu nástrojů. Máme k dispozici příkazy z panelu nabídek, ikony na panelech nástrojů a klávesové zkratky. Řadu činností můžeme vykonávat přímo myší. V převážné většině se jedná o univerzální postupy, které používáme k řešení našich konkrétních úkolů. Program ovládáme prostřednictvím klávesnice nebo myši a pro každou situaci je v programu zpracováno řešení ve formě instrukce, která určuje reakci počítače na naše zadání. Když klepneme na ikonu **Uložit**, počítač uloží do paměti aktuální stav dokumentu. A nás jako uživatele ani nemusí zajímat, jak to provede. Stačí vědět, že dokument je možno uložit, a znát způsob, jak to zajistit.

I když je nabídka nástrojů opravdu bohatá, nemůže nikdy zcela pokrýt naše individuální potřeby. Představme si situaci, kdy máme rozsáhlou tabulku s tisícem řádků a kdy potřebujeme vymazat obsah všech buněk, které leží v sudých řádcích ve druhém sloupci. Vymazat obsah jedné buňky není problém. Stačí vybrat buňku, stisknout klávesu DELETE, zvolit způsob mazání a klávesou ENTER vymazání potvrdit. Pokud se ale jedná o 500 buněk, je to už poněkud obtížnější. Minimálně to zabere určitý čas, kromě toho se můžeme snadno splést, některou buňku přeskočit, nebo naopak vymazat buňku v lichém řádku. V této situaci, a zvláště kdybychom měli mazat buňky opakovaně, bychom asi uvítali, kdyby existoval příkaz „Vymazat každou buňku ve druhém sloupci v sudých řádcích až do řádku 1000“. Takový příkaz bohužel v nabídce nenajdeme, program nám ale poskytuje možnost si takový příkaz vytvořit. Vytvořením příkazu rozšiřujeme funkčnost programu a původně zcela univerzální program si tak přizpůsobujeme pro naše potřeby. A právě v tom spočívá smysl tvorby našich individuálních příkazů, které souhrnně označujeme jako makra.

Pokud bychom chtěli někoho požádat, aby buňky vymazal místo nás, mohli bychom požadavek zformulovat třeba následujícím způsobem:

1. vyber buňku ve druhém sloupci, druhém řádku a vymaž její obsah,
2. ve stejném sloupci vyber buňku v dalším sudém řádku a opět vymaž obsah,
3. postup podle bodu 2 opakuj až do řádku 1000.

Zformulovali jsme soubor instrukcí, jejichž provedení vede k požadovanému výsledku. Počítači můžeme stejnou práci zadat obdobným způsobem. Instrukce ale musí být pro počítač srozumitelné, a proto k jejich zapsání musíme použít některý z vhodných programovacích jazyků. My budeme používat OpenOffice.org Basic, nebo jen zkráceně Basic, a zápisy instrukcí v tomto jazyku budeme označovat jako procedury.

Pro zápis procedur je v aplikaci k dispozici speciální textový editor, který kromě vlastního zápisu poskytuje i celou řadu dalších podpůrných funkcí. Editor se označuje jako integrované vývojové prostředí, zkráceně IDE, z anglického Integrated Development Environment.

Pro získání celkového přehledu je vhodné také uvést, jaké jsou možnosti pro zadávání našich požadavků aplikaci OpenOffice.org. Základní možností je ovládání programu klávesnicí a myší. V tomto případě hovoříme o práci s **uživatelským rozhraním**. Zobrazené příkazy a ikony ovládáme přímo, obdobně jako ovládáme například tlačítka na televizoru. Kromě toho ale existuje i další způsob, a sice využití **aplikačního programovacího**

rozhraní, zkráceně API, z anglického Application Programming Interface. Práci s API lze přirovnat k použití dálkového ovladače u televizoru. Ovladačem můžeme rovněž ovlivňovat funkci televizoru. Na rozdíl od mačkání tlačítek přímo na televizoru ovládáme funkce světelným paprskem, který vysílá ovladač. Mohli bychom tedy říci, že pracujeme se světelným rozhraním. Je zřejmé, že televizor bude reagovat jen na světelné signály, kterým bude rozumět. A obdobné je to u naší aplikace. Již jsme uvedli, že makra musí být zapsána v programovacím jazyku. Kromě toho ale musí být zformulována tak, aby jim aplikace rozuměla. Informace, které k tomu potřebujeme, můžeme získat různým způsobem, ale jejich základním zdrojem je **popis API**.

Do popisu API se nutně promítá způsob, jakým byla vytvořena sama aplikace OpenOffice.org. Při její tvorbě byla využita **technologie UNO**, což je zkratka anglického Universal Network Objects (univerzální síťové objekty). Univerzalita této technologie spočívá v tom, že s objekty vytvořenými touto technologií je možno pracovat v různých prostředích a s použitím různých programovacích jazyků. To platí i pro aplikaci OpenOffice.org, a tedy i pro API. Z existujících možností byla v této příručce vybrána možnost nejsnazší, a sice použití jazyka Basic. Tomuto jazyku je věnována první část knihy. Práci s API se budeme věnovat ve druhé části.

V příručce je kladen důraz především na vysvětlení základních pojmů a pracovních postupů. Dalším cílem je poskytnout návod, jak získat další informace a jak s nimi pracovat. Díky technologii UNO lze získané poznatky v široké míře využít i pro práci s dalšími komponentami aplikace, např. s textovým editorem Writer.

Základním zdrojem informací o aplikaci jsou webové stránky <http://www.openoffice.org> a <http://www.openoffice.cz>. Zde můžeme zadarmo získat instalační soubory aplikace a k dispozici je zde rovněž balíček pro návrháře SDK (Software Development Kit). V balíčku je kromě jiného obsáhlý a velmi podrobný průvodce pro návrháře Developer's Guide s informacemi o práci s API. Ve složce **docs\common\ref** je uložen pod názvem **module-ix.html** vlastní popis API. Aktuální popis API je uveden též na adrese <http://api.openoffice.org/docs/common/ref/com/sun/star/module-ix.html>.

Řadu cenných informací lze získat i v dokumentaci společnosti Sun Microsystems a v příspěvcích uživatelů aplikace. Velmi cenné praktické informace jsou k dispozici např. na adrese <http://www.pitonyak.org>. Určitou nevýhodou uváděných zdrojů je skutečnost, že jsou vesměs v angličtině. Práci s průvodcem pro návrháře rovněž ztěžuje okolnost, že většina uváděných příkladů je napsána v jazyku JAVA, který se od jazyka Basic značně odlišuje. Myšlenka otevřeného software si však získává stále více příznivců, a tak je velmi pravděpodobné, že se bude i podpora pro uživatele neustále zlepšovat.

OpenOffice.org je otevřený systém, který se neustále vyvíjí a zdokonaluje. Dodržuje se však zásada, že poznatky a metody z předchozích verzí lze uplatnit i ve verzích následujících. V knize je popsána práce s verzí 2 podle stavu ke konci roku 2005.

Knihy je vybavena příklady ke stažení na www.gradac.cz. Protože v některých kapitolách se na tyto příklady odvoláváme a pracujeme s nimi, doporučujeme čtenáři, aby si je ze zmíněné adresy stáhl a umístil na svém disku do složky **C:\Příklady OO.o** (například). Příklady jsou ve formě tabulkových dokumentů, které získáte po rozbalení (odzipování) staženého souboru.

Část I

Práce s jazykem Basic

V první části knihy se budeme věnovat programovacímu jazyku OpenOffice.org Basic. V porovnání s ostatními programovacími jazyky je Basic jednodušší a proto je vhodný pro psaní makr, jež nevytvářejí samostatné aplikace, ale programy pouze doplňují a rozšiřují.

Basic disponuje řadou příkazů a funkcí, které umožňují zejména jednoduchou komunikaci s uživatelem prostřednictvím dialogových oken a zpracování textových i číselných hodnot. Dále je k dispozici větvení, což znamená, že makro může probíhat různým způsobem, v závislosti na zadáných hodnotách nebo na rozhodnutí uživatele. Užitečné jsou i příkazy, které jednoduše zajišťují opakování jedné nebo několika instrukcí, a řada dalších příkazů a funkcí.

V rámci aplikace OpenOffice.org představuje Basic relativně samostatný uzavřený celek bez těsnější vazby na dokumenty OpenOffice.org. Pro uživatele je velmi výhodné, že má přímo v rámci aplikace k dispozici dobře zpracovanou nápovědu, ve které jsou stručnou formou uvedeny všechny informace nezbytné pro práci s tímto jazykem.

První část knihy je do určité míry přípravná a jejím cílem je především seznámit čtenáře s jazykem a připravit ho na práci s objekty OpenOffice.org ve druhé části knihy.

1.

Naše první makro

V první kapitole uvedeme podrobněji, co je to makro, jakým způsobem se vytváří a k čemu může sloužit. Dále vytvoříme jednoduché makro s využitím záznamníku maker a napíšeme klasické makro s pozdravem.

1.1 Co je to makro, jak vzniká a k čemu slouží

Pro práci s počítačem existuje v současné době velké množství programů. Tyto programy mohou mít nejrůznější charakter. Může se jednat o specializované programy zaměřené na určitou problematiku, jako jsou například hry, prohlížeče internetu nebo programy pro účetnictví, anebo se setkáváme s univerzálními programy, které umožňují velmi široké použití. Příkladem může být textový editor nebo tabulkový procesor. Každý typ programu má své přednosti i nedostatky. Specializované programy jsou zpravidla navrženy tak, aby maximálně usnadnily určitou činnost. Jejich nevýhodou je většinou poměrně nízká flexibilita a dosti vysoké náklady spojené s pořízením. Univerzální programy umožňují řešit širokou škálu úkolů, jejich všestrannost ale bývá vykoupena větší pracností při řešení konkrétního úkolu. Zpracování specializovaných programů „na míru“ je dosti nákladné, a proto se v řadě

případů musíme spokojit s některým z univerzálních programů. Velmi široce používané jsou tak zvané kancelářské balíky, které zpravidla obsahují textový editor, tabulkový procesor a prezentační program. Jedním z takových balíčků je i sada OpenOffice.org. Její velkou předností je volná přístupnost a tedy prakticky nulové pořizovací náklady.

Univerzální programy prošly poměrně dlouhým vývojem a dnes již disponují širokou paletou nástrojů. Při obsluze programu z klávesnice (nebo myši) však musíme jednotlivé příkazy vždy znovu manuálně zadávat. Opakování většího množství příkazů je proto únavné a časově náročné. I u univerzálních programů však existuje možnost jejich modifikace tak, aby se přiblížily specializovaným programům. Touto možností je použití maker. Makra představují určitou nadstavbu nebo doplnění základního programu. Samotné slovo makro vzniklo jako zkratka slova makroinstrukce. Počítač pracuje na základě zadaných instrukcí. Instrukcí může být například stisk klávesy nebo volba určitého příkazu z panelu nabídek. Instrukcí může být také příkaz napsaný v programovacím jazyku. A soubor několika takových instrukcí potom tvoří makroinstrukci, tedy naše makro.

Makra mohou vznikat dvojím způsobem. Některé programy, a to se týká i OpenOffice.org, jsou vybaveny záznamníkem instrukcí. Při použití záznamníku postačuje spustit záznam a následně vykonat určitou činnost, např. zadat údaje do buňky a změnit její formát. Po ukončení záznamu máme k dispozici nově vytvořené makro se záznamem provedených činností. Při každém spuštění makra se zaznamenané činnosti vždy znovu provedou. Dosažený výsledek závisí na dokonalosti záznamníku a jeho schopnosti zachytit všechny provedené činnosti.

Druhou možností je vytvoření makra ve vhodném programovacím jazyku. Vytvoření makra tímto způsobem je sice náročnější, ale výsledné makro může zajišťovat i činnosti, které jsou pro uživatele jinak nedostupné. Příkladem může být použití dialogových oken pro obousměrnou komunikaci s počítačem nebo nastavení různé odezvy programu v závislosti na aktuální situaci. Makro totiž může reagovat různým způsobem, například podle toho, jaká je pozice aktivní buňky v listu. Oba způsoby lze i kombinovat a zaznamenané makro následně upravit a doplnit.

Užití maker je velmi široké. Od jednoduchých maker, která zachycují několik příkazů zadaných z klávesnice, až po obsáhlé soubory maker, které mohou zvýšit výkonnost univerzálního programu na úroveň blížící se jednoúčelovým specializovaným programům.

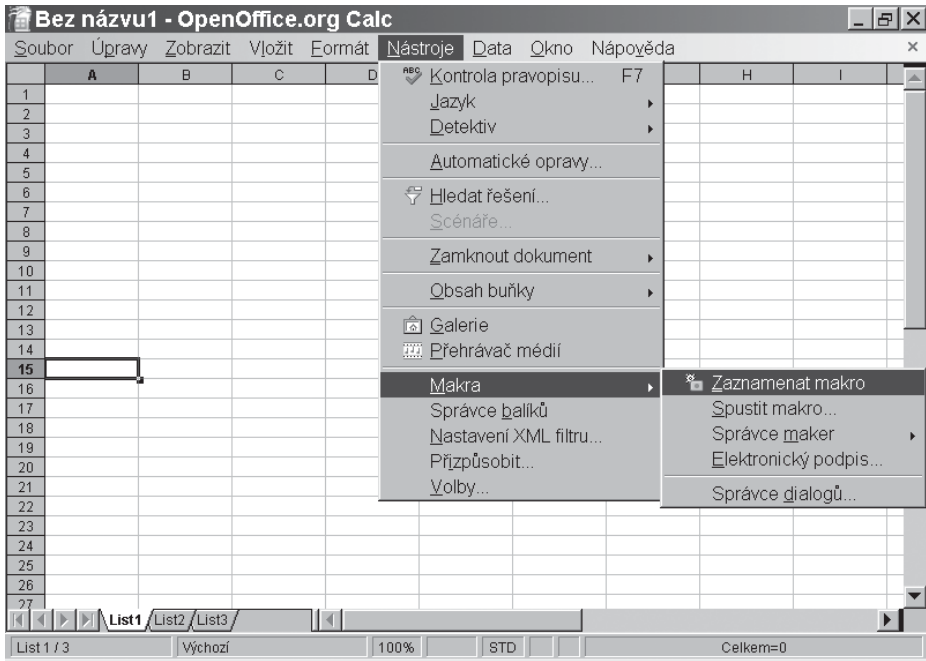
Možnosti programovacích jazyků jsou dnes natolik bohaté, že budeme těžko hledat zadání, které by nebylo možno makrem zajistit.

1.2 Vytvoření makra s využitím záznamníku maker

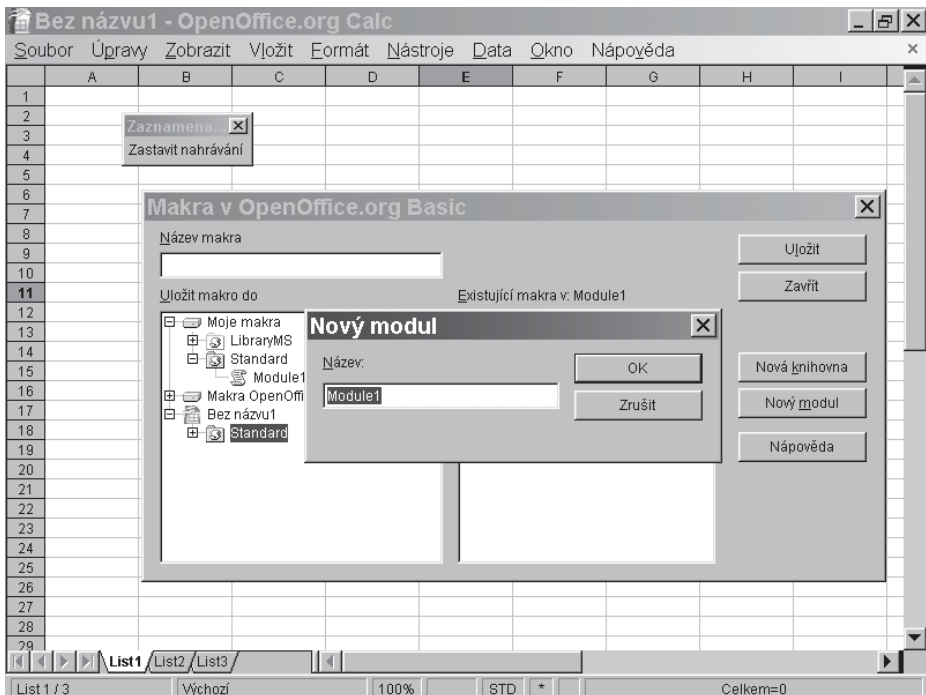
V malé ukázkě vytvoříme makro, které po spuštění vybere buňku E10, změní barvu pozadí, do buňky zapíše text a nakonec aktivuje buňku E11. Nejdříve spustíme tabulkový procesor aplikace OpenOffice.org. Při standardní instalaci jsou programové soubory uloženy ve složce **C:\Program Files\OpenOffice.org 2.0**. Zde ve složce **program** najdeme a vybereme soubor **scalc.exe**. Stiskem klávesy ENTER nebo poklepáním soubor otevřeme. Vytvoří se tabulkový dokument s názvem **Bez názvu1**. Zobrazí se list **List1** s aktivní buňkou A1. Na panelu nabídek vybereme příkaz **Nástroje** → **Makra** a potvrdíme řádek **Zaznamenat makro** (viz obrázek 1.1).

Zobrazí se malý panel s nadpisem **Zaznamenat...** a s jedním příkazovým řádkem **Zastavit nahrávání**. Od této chvíle se naše akce budou zaznamenávat do makra. Vybereme myší buňku E10, pomocí příkazů z nabídky **Formát** → **Buňky...** zobrazíme dialogové okno se záložkami. Vybereme záložku **Pozadí** a zde nastavíme odstín pozadí na hodnotu Zelená 7.

1. Naše první makro



Obr. 1.1: Zabíjení záznamu makra



Obr. 1.2: Ukládání nabraného makra

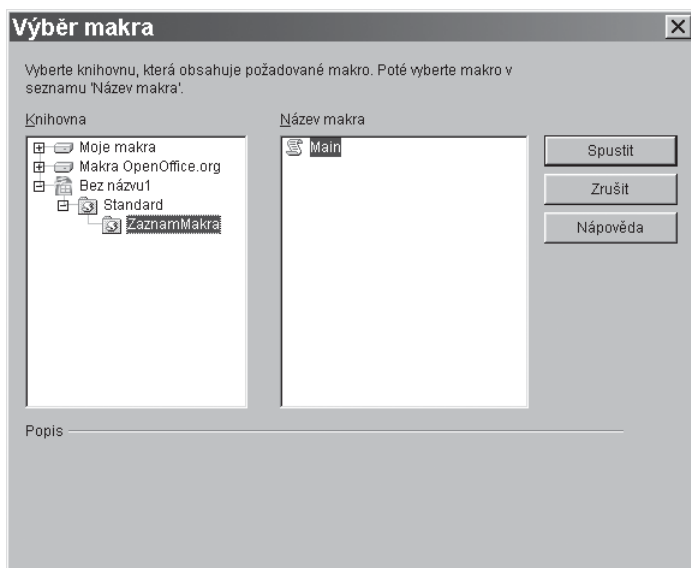
1.2 Vytvoření makra s využitím...

Potvrzením volby tlačítkem **OK** se dialogové okno uzavře. Do buňky zapíšeme text „buňka E10“ a stisknutím klávesy **ENTER** aktivujeme buňku E11. Záznam makra ukončíme klepnutím na řádek **Zastavit nahrávání** v malém dialogovém okně.

Okno signalizující nahrávání makra se uzavře a místo něj se zobrazí nové dialogové okno nazvané **Makra**. Pomocí tohoto okna určíme, kam se nahrané makro uloží. Soustředíme se na levé dolní okno nazvané **Uložit makro do**. Svislým posuvníkem se přesuneme na dolní okraj seznamu a zde klepnutím otevřeme soubor **Bez názvu1** a vybereme řádek **Standard**. Dále klepneme na tlačítko **Nový modul** a zobrazí se dialogové okno se stejným názvem. V textovém poli je nabídnut název modulu **Module1** (viz obrázek 1.2).

Místo něj zapíšeme text např. „ZaznamMakra“ a klepneme na tlačítko **OK**. Okno se zavře a vrátíme se zpět do okna **Makra**. Zde v levé části přibyl pod řádkem **Standard** nový řádek s právě zapsaným názvem **ZaznamMakra**. Uložení makra ukončíme klepnutím na tlačítko **Uložit**. Na dotaz, zda chceme přepsat makro **Main**, odpovíme kladně stiskem tlačítka **Ano**.

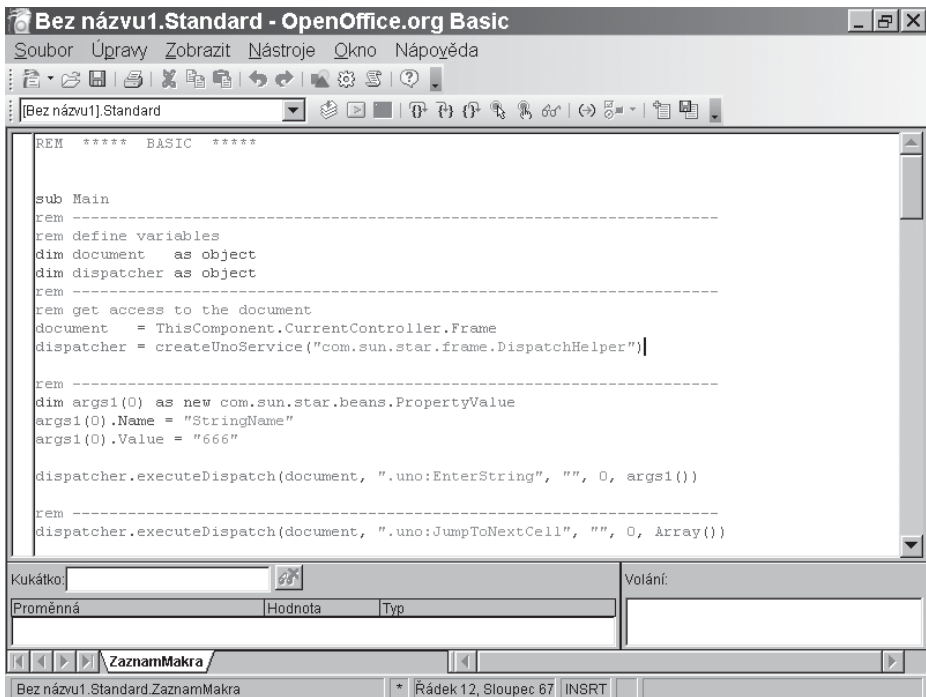
Makro s názvem **Main** je zaznamenáno a uloženo. A nyní funkci makra ověříme. Nejdříve vymažeme text v buňce E10 a nastavíme původní barvu pozadí. S využitím příkazu **Nástroje** → **Makra** → **Spustit makro...** zobrazíme okno **Výběr Makra**. V levé části vybereme soubor **Bez názvu1** a klepnutím na řádek **Standard** zobrazíme řádek **ZaznamMakra**. V pravém okně se zobrazí řádek s názvem **Main**, jak ukazuje obrázek 1.3. Po klepnutí na tlačítko **Spustit** se makro spustí a provedou se stejné operace jako při jeho nahrávání.



Obr. 1.3: Spuštění makra

Asi budeme zvědaví, jak takové makro vypadá, a proto se na jeho záznam podíváme. Obdobně jako při spouštění makra zobrazíme příkazem **Nástroje** → **Makra** → **Správce maker** → **OpenOffice.org Basic...** okno **Makra**. V levé části otevřeme soubor **Bez názvu1** a klepnutím na řádek **Standard** zobrazíme řádek **ZaznamMakra**. V pravém okně se zobrazí řádek s názvem **Main**. Tentokrát ale klepneme na tlačítko **Upravit**. Místo tabulky se zobrazí jednoduchý textový editor (jako na obrázku 1.4) s textem uvedeným v příkladu 1.1.

1. Naše první makro



Obr. 1.4: Záznam makra v integrovaném vývojovém prostředí

Spouštění tabulkového procesoru si můžeme usnadnit umístěním spouštěcí ikony na ploše. Ve složce **C:\Program Files\OpenOffice.org 2.0\program** vybereme soubor **scalx.exe** a stiskneme pravé tlačítko myši. V otevřené nabídce vybereme a potvrdíme řádek **Vytvořit zástupce**. Nově vytvořený soubor zástupce přetáhneme na plochu. Ke spouštění tabulkového procesoru nyní stačí poklepat na ikonu.



Příklad 1.1: Procedura vytvořená záznamníkem maker

```

Sub Main
rem -----
rem define variables
dim document as object
dim dispatcher as object
rem -----
rem get access to the document
document = ThisComponent.CurrentController.Frame
dispatcher =
    createUnoService("com.sun.star.frame.DispatchHelper")

rem -----
dim args1(0) as new com.sun.star.beans.PropertyValue
args1(0).Name = "ToPoint"
args1(0).Value = "$E$10"

```

```

dispatcher.executeDispatch(document, ".uno:GoToCell", "", 0,
    args1())

rem -----
dim args2(4) as new com.sun.star.beans.PropertyValue
args2(0).Name = "BackgroundPattern.Transparent"
args2(0).Value = false
args2(1).Name = "BackgroundPattern.BackColor"
args2(1).Value = 4057917
args2(2).Name = "BackgroundPattern.URL"
args2(2).Value = ""
args2(3).Name = "BackgroundPattern.Filtername"
args2(3).Value = ""
args2(4).Name = "BackgroundPattern.Position"
args2(4).Value = com.sun.star.style.GraphicLocation.NONE

dispatcher.executeDispatch(document, ".uno:BackgroundPattern",
    "", 0, args2())

rem -----
dim args3(0) as new com.sun.star.beans.PropertyValue
args3(0).Name = "StringName"
args3(0).Value = "Bunka E10"

dispatcher.executeDispatch(document, ".uno:EnterString", "", 0,
    args3())

rem -----
dispatcher.executeDispatch(document, ".uno:JumpToNextCell", "",
    0, Array())

End Sub

```

Popis:

Možná nás překvapí délka textu nezbytná pro záznam několika jednoduchých úkonů. Text vypadá také dost tajuplně, ale tím se nenechme odradit. Důležité je, že jsme dokázali vytvořit a spustit první jednoduché makro. A i jednoduchá makra vytvořená tímto způsobem nám někdy mohou posloužit jako užitečný pomocník. Zatím si pouze všimněme, že zápis makra začíná slovem **Sub**, za nímž následuje název makra **Main**, a končí slovy **End Sub**. Slůvko **Sub** je zkratkou anglického subroutine, což se překládá jako podprocedura, **End Sub** značí konec podprocedury. Protože v jazyku OpenOffice.org Basic není formální rozdíl mezi procedurami a podprocedurami, slovo podprocedura se většinou nepoužívá a o zápisech maker hovoříme jako o procedurách.

1.3 Makro „Zdravím svět“

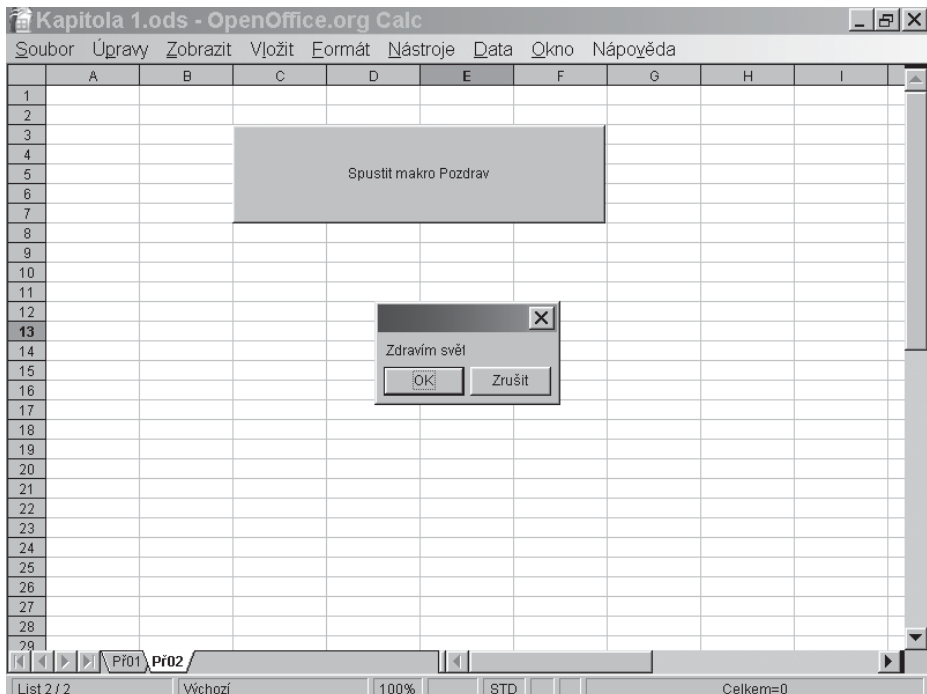
Podstatně širší možnosti než nahrávání poskytuje tvorba maker v integrovaném vývojovém prostředí IDE. Do tohoto prostředí jsme již nahlédli v kapitole 1.2, když jsme prohlíželi záznam makra vytvořený záznamníkem. Opět tedy přejdeme do tohoto prostředí a zapíšeme zde makro s pozdravem, který se po spuštění makra objeví v dialogovém okně.

1. Naše první makro

Stejně jako v kapitole 1.2 vytvoříme nový tabulkový dokument s názvem **Bez názvu1**. Na panelu nabídek vybereme příkazem **Nástroje** → **Makra** → **Správce maker** → **OpenOffice.org Basic...** okno **Makra**. V levé části otevřeme soubor **Bez názvu1** a vybereme řádek **Standard**. Klepnutím na tlačítko **Nový** zobrazíme okno **Nový modul**. Do textového pole zapíšeme název modulu, v němž budeme vytvářet naše makro, například „ModulPozdrav“, a potvrdíme tlačítkem **OK**. Zobrazí se rozhraní IDE s modulem **ModulPozdrav**. Název modulu je uveden ve spodní záložce. V části textového editoru uvidíme automaticky zapsaný text:

```
REM ***** BASIC *****
Sub Main
End Sub
```

Zápis upravíme tak, že název **Main** změníme na **Pozdrav** a přidáme řádek s textem **Print** „**Zdravím svět**“. Zápis po úpravách je uveden v příkladu 1.2. Makro spustíme obdobně jako v kapitole 1.2. Přejdeme do tabulkového dokumentu a s využitím příkazu **Nástroje** → **Makra** → **Spustit makro...** zobrazíme okno **Výběr Makra**. V levé části otevřeme soubor **Bez názvu1** a klepnutím na řádek **Standard** zobrazíme řádek **ModulPozdrav**. V pravém okně se zobrazí řádek s názvem **Pozdrav**. Po klepnutí na tlačítko **Spustit** se makro spustí a zobrazí se dialogové okno s textem pozdravu „Zdravím svět“ jako na obrázku 1.5. Naše makro je podstatně jednodušší než záznam makra z kapitoly 1.2. A přesto nám poskytuje novou funkčnost, kterou bez maker nelze získat, a sice zobrazení dialogového okna se zadanou informací.



Obr. 1.5: Pozdrav zobrazený po spuštění makra z příkladu 1.2



Mezi prostředím IDE a tabulkovým dokumentem přebházíme nejsnadněji pomocí kláves ALT a TAB. Stiskneme klávesu ALT, klávesu držíme stisknutou a krátce stiskneme klávesu TAB. Zobrazí se dialogové okno s ikonami aktuálně otevřených oken. Klávesu ALT držíme stále stisknutou a opakovaným stiskem klávesy TAB vybereme okno, které chceme zobrazit. Vybrané okno se zobrazí po uvolnění klávesy ALT.

Příklad 1.2: Makro „Zdravím svět“

```
REM ***** BASIC *****
Sub Pozdrav
  Print "Zdravím svět"
End Sub
```

Popis:

Makro začíná opět slovem **Sub**, za nímž následuje název makra **Pozdrav**, a končí slovy **End Sub**. Uvnitř makra je zapsán jediný příkaz **Print**, který zajišťuje zobrazení dialogového okna s textem zapsaným v uvozovkách za příkazem. Příkaz a text jsou odděleny mezerou. V prvním řádku je zápis začínající slůvkem **REM**. Jedná se o zkratku anglického Remark (poznámka). Do poznámek se zapisují informace o makru a o jeho funkci. Tyto poznámky usnadňují používání maker a jsou dobrým pomocníkem pro orientaci v makru při hledání případných chyb nebo při provádění dodatečných úprav. Počítač při provádění maker slůvko **REM** i text, který následuje až do konce řádku, ignoruje. Stejnou funkci jako **REM** má i apostrof ‘.

Napsáním vlastního makra jste učinili první, a asi nejdůležitější krok na cestě k daleko lepšímu ovládnutí počítače, než které umožňují hotové programy. Získáváte mocný nástroj, k jehož zvládnutí by vám chtěla tato knížka napomoci.