

PROVOZCE

Doporučuje časopis

**PC WORLD**

# Microsoft **Excel**

## 2000, 2002, 2003

**záznam, úprava a programování maker**

Jaroslav Černý

- Vhodné oblasti pro použití maker
- Krátký kurz nahrávání maker
- Objektový model Excelu
- Řešení chyb, k nimž může dojít
- Často používané konstrukce
- Editor jazyka Visual Basic for Applications

GRANIP

# 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.







Copyright © Grada Publishing, a.s.



Copyright © Grada Publishing, a.s.

## Obsah

	<b>Úvod.....</b>	<b>11</b>
	<b>1. Makra v Excelu .....</b>	<b>14</b>
	1.1 Proč je v Excelu obsažen i programovací jazyk.....	14
	Ve kterých případech je vhodné používat makra? .....	14
	Jak se dají makra vytvořit? .....	15
	Kdo makra obvykle vytváří? .....	15
	Jak se dají hotová makra spouštět? .....	15
	1.2 Něco málo o vývoji Excelu z hlediska programování .....	16
	1.3 Základní údaje o zastaralém makrojazyku XLM .....	16
	1.4 Typy listů v sešitu a jejich vztah k programování .....	17
	<b>2. Nahrávání maker .....</b>	<b>20</b>
	2.1 Proč makra nahráváme?.....	20
	2.2 Začínáme s nahráváním maker .....	20
	První nahrané makro: formátování buněk .....	21
	Zkoumání nahraných maker .....	24
	Problém názvů listů či jiných objektů .....	27
	2.3 Relativní a absolutní odkazy na buňky a jejich použití při nahrávání maker .....	29
	<b>3. První setkání s editorem jazyka Visual Basic .....</b>	<b>32</b>
	3.1 Jak nahraná makra upravit? .....	32
	3.2 Spuštění editoru jazyka Visual Basic .....	32
	3.3 Vzhled editoru jazyka Visual Basic .....	34
	Kde všude může být kód VBA? .....	37
	3.4 Režimy práce v editoru Visual Basicu.....	39
	3.5 Rozbor nahraných maker .....	39
	Hlavička makra .....	41
	Komentáře.....	41
	Vlastní kód makra Pokus.....	42
	Krátce k barvám kódu.....	44
	3.6 Okno Immediate .....	44
	<b>4. Úvod do objektů ve VBA .....</b>	<b>48</b>
	4.1 Principy programování v Excelu .....	48
	4.2 Objekty v jazyku VBA .....	48
	Pračka jako typický příklad objektu.....	49



	Třídy objektu.....	49
	Kolekce a jejich vztah k samostatným objektům.....	50
	Kontejnery jako „nádoby“ na jiné objekty a jejich hierarchie...50	
	Odkazy na objekty v kódu .....	52
	Zkrácené odkazy na objekty .....	54
	Rozdíl mezi vlastnostmi a metodami v kódu .....	55
4.3	Události .....	56
4.4	Nápověda k jednotlivým objektům.....	56
	Kontextová nápověda .....	57
<b>5.</b>	<b>Začínáme psát vlastní makra.....</b>	<b>60</b>
5.1	Části standardního modulu kódu .....	60
5.2	Procedury a funkce .....	60
	Jaký je rozdíl mezi makrem a procedurou?.....	61
	Jak vytvoříte novou proceduru či funkci? .....	61
5.3	Jednotlivé části podprogramů .....	63
	Struktura vlastní funkce.....	64
5.4	Příkazy a jejich zápis .....	64
5.5	Proměnné .....	65
	Přiřazení hodnoty do proměnné .....	65
5.6	Konstanty .....	66
5.7	Názvy objektů, proměnných a konstant .....	66
5.8	Výrazy .....	67
5.9	Operátory .....	67
	Ukázky operátorů a jejich vyhodnocování .....	69
5.10	Datové typy proměnných a konstant.....	69
	Datové typy proměnných a konstant.....	69
5.11	Rozsah platnosti a životnost .....	72
	Životnost proměnných .....	73
5.12	Deklarace proměnných .....	73
	Deklarace konstant a jejich typ.....	75
5.13	Praktické ukázky deklarací proměnných, jejich platnosti a životnosti .....	75
	Platnost proměnných.....	75
	Statické proměnné.....	77
5.14	Způsoby zápisu hodnot do proměnné.....	77
	Textové údaje .....	77
	Číselné údaje .....	78
	Datum a čas.....	78
	Logické údaje .....	79

5.15	Další poznámky k datovým typům .....	79
	Datový typ Variant .....	79
	Datový typ String .....	79
	Logický typ Boolean .....	80
	Převody mezi jednotlivými typy hodnot .....	80
5.16	Pole proměnných .....	81
	Statická pole .....	81
	Dynamická pole .....	81
	Vícerozměrné pole .....	82
	Přiřazení hodnoty do prvku pole .....	82
	Pole proměnných v proměnné typu Variant – funkce Array .....	82
5.17	Parametry a jejich druhy .....	82
	Parametry ve vlastních procedurách či funkcích .....	84
5.18	Rozhodovací bloky .....	86
	Příkaz Select Case .....	86
	Rozhodovací blok If...Then...Else .....	87
5.19	Cykly .....	89
	Cyklus For...Next .....	89
	Cyklus Do...While   Until .....	89
	Cyklus For Each...Next .....	90
5.20	Další možnosti pro spouštění procedur .....	91
	Spouštění procedur přímo v modulu .....	91
	Spouštění procedur z jiných procedur či funkcí .....	91
	Spouštění maker z panelů nástrojů .....	92
	Spouštění maker z nabídky Excelu .....	94
	Spouštění maker pomocí příkazových tlačítek v listech .....	94
<b>6.</b>	<b>Často používané příkazy, funkce a konstrukce.....</b>	<b>98</b>
6.1	Vestavěné příkazy a funkce obecně .....	98
	Funkce pracovního listu v kódu VBA .....	98
6.2	Jak z kódu informovat uživatele – příkaz a funkce MsgBox .....	99
6.3	Zadávání hodnot – funkce a metoda InputBox .....	103
6.4	Funkce pro práci s textovými hodnotami .....	105
	Funkce, které vrací část hodnoty proměnné .....	105
	Funkce pro hledání v textu a porovnání řetězců .....	106
	Převod na velká a malá písmena .....	107
6.5	Funkce pro práci s číselnými hodnotami .....	107
	Zaokrouhlování desetinných čísel .....	107
	Trigonometrické, logaritmické a jiné funkce .....	108
6.6	Vlastní funkce, které můžete použít na pracovních listech .....	108







6.7	Další časté konstrukce .....	110
	Příkaz With...End With .....	110
	Odkazy na objekty pomocí proměnných .....	111
<b>7.</b>	<b>Pracujeme s objektem Range.....</b>	<b>114</b>
7.1	Co všechno může být objektem typu Range.....	114
7.2	Jak určit oblast buněk .....	114
7.3	Odkazy na objekt typu Range .....	114
	Odkazy stylem A1 – vlastnost Range .....	115
	Odkaz vlastností Cells.....	116
	Vlastnosti Range a Cells, použité na jiných objektech Range ..	117
	Odkazy na celé řádky a sloupce – vlastnosti Columns a Rows ....	117
	EntireColumn a EntireRow .....	118
	Odkaz na vybranou oblast buněk – Selection .....	119
	Odkaz pomocí vlastnosti Offset .....	119
	Odkaz pomocí vlastnosti CurrentRegion a UsedRange .....	119
	Odkaz pomocí vlastnosti Areas .....	120
	Odkaz na buňky od pevné počáteční buňky do určitého místa....	121
	Odkaz pomocí vlastností TopLeftCell a BottomRightCell .....	121
7.4	Pracujeme s oblastmi buněk – základní vlastnosti a metody .....	121
	Nejdůležitější vlastnosti objektu Range .....	122
7.5	Příklady různých technik pro práci s buňkami.....	125
	Kopírování buněk .....	125
	Přesun buněk na jiné místo.....	125
	Vkládání ze schránky jinak – metoda PasteSpecial.....	125
	Mazání obsahu či formátu buněk.....	126
	Vkládání buněk do listu a jejich odstraňování.....	127
	Vkládání vzorců do buněk .....	127
	Práce s komentáři v buňce .....	127
	Automatické vyplnění určité oblasti.....	128
	Odkaz na speciální oblasti buněk .....	128
	Prohledávání a zaměňování obsahu buněk .....	130
	Setřídění určité oblasti.....	132
	Formátování obsahu buněk .....	133



<b>8.</b>	<b>Pracujeme s objekty Workbook a Worksheet .....</b>	<b>138</b>
	Jak jsou listy v sešitu vyjádřeny objektově?.....	138
	Odkazy na aktivní sešit či list .....	138
8.1	Pracovní listy .....	139
	Kolekce Worksheets .....	139
	Základní vlastnosti objektu Worksheet .....	140



	Základní metody objektu Worksheet.....	141
	Odstranění listu .....	142
	Zkopírování nebo přesun celého listu.....	142
	Vložení obsahu schránky na list .....	142
	Vložení obsahu schránky na list v určitém formátu .....	143
	Vytištění listu a tiskový náhled .....	144
	Uložení listu do souboru jiného typu .....	144
	Ochrana listu proti změnám.....	145
8.2	Práce s objektem PageSetup a tisk listů .....	147
	Změna orientace papíru .....	149
	Jak nastavit okraje stránky .....	149
	Jak vymezit tiskovou oblast listu.....	149
	Zvětšení či zmenšení obsahu listu při tisku.....	149
	Vložení ručního konce stránky .....	150
8.3	Pracujeme se sešity .....	150
	Kolekce sešitů Workbooks .....	151
	Vlastnosti a metody objektu Workbook.....	152
	Základní postupy pro práci se sešity.....	155
8.4	Odpověď na hádanku.....	159
	<b>9. Objekt Application a jeho význam.....</b>	<b>162</b>
9.1	Užitečné metody a vlastnosti objektu Application .....	162
	Jak urychlit makro potlačením vykreslování změn.....	162
	Potlačení některých výstražných dialogů.....	163
	Zobrazování zpráv o průběhu makra .....	163
	Změna tvaru ukazatele myši .....	163
	Vlastnosti prostředí Excelu.....	164
	Metody objektu Application, které jsme již poznali .....	165
	Přechod na jiné místo sešitu.....	166
	Metoda OnKey – spuštění procedury klávesovou zkratkou ..	166
	Metoda OnTime – spuštění procedury v určitý čas.....	167
	Pozastavení makra na určitou dobu.....	167
	Uložení pracovního prostoru .....	167
	Ukončení Excelu .....	167
9.2	Zobrazení vestavěných dialogových oken .....	168
	<b>10. Jak řešit chyby v makrech .....</b>	<b>172</b>
	Chyby během kompilace .....	173
10.1	Chyby za běhu programu.....	173
10.2	Chybové rutiny v kódu .....	174
	„Vnitřek“ chybové rutiny .....	175
	Proč je důležité mít chybové rutiny ve všech procedurách ..	176
	Konkrétní ukázka chybové rutiny s několika „figly“ .....	177

10.3	Ladící nástroje Excelu .....	179
	Stručně o režimu přerušení .....	180
	Rychlé zobrazení hodnoty nějakého výrazu či proměnné ....	180
	Okna související s laděním chyb .....	181
	Testování hodnot v okně Immediate .....	181
	Jak používat okno Locals .....	182
	Kukátka .....	182
	Rychlé kukátko .....	184
	Zarážky (breakpoint) .....	184
	Krokování procedur .....	184
	Možné vynechání určitých míst v kódu .....	185
10.4	Stručný přehled nejdůležitějších chyb .....	186
	<b>Rejstřík .....</b>	<b>189</b>

## Úvod

Kniha se zaměřuje na základy programování v *Excelu 2003*; pokud máte starší verze *Excelu (97, 2000, 2002)*, můžete se podle ní učit také, z hlediska základů programování se tyto verze neliší.

V prvním dílu knihy se věnuji těm záležitostem, které programátor *Excelu* prostě „musí“ znát, a těm technikám, které bude v *Excelu* používat nejčastěji. Navazující druhý díl je učebnicí „pokročilých“ programovacích technik a objektů.

V celém textu se snažím být maximálně stručný, což je přirozený důsledek dlouholeté zkušenosti lektora, který ví, že jeho posluchači (čtenáři) se chtějí něco naučit a ne poslouchat marketingové a jiné podobné bláboly. Sem tam na některé věty či pasáže narazíte vícekrát – opakování je matka moudrosti.

Programování maker není záležitostí pro úplné počítačové začátečníky a také ne pro ty, kdo si jen rádi hrají. V celém textu knihy předpokládám, že umíte *Excel* ovládat a víte, k čemu všemu se dá použít. Např. v části věnované programování grafů rozhodně nenajdete výklad toho, jak se graf vytváří ručně v uživatelském rozhraní ...

Počítejte také s tím, že bez znalosti angličtiny se toho moc nenaučíte, pro zvládnutí pokročilých témat je nezbytná, ale i u základů se vám bude hodit. Česky toho totiž příliš nevyšlo a spoustu materiálů seženete na internetu jen v angličtině. Schází především česká referenční příručka s **kompletním** českým překladem nápovědy k objektům, vlastnostem, metodám a událostem, i když se v tomto směru stále dá doporučit výtečná knížečka „Programování Office 97“ od Markéty a Petra Šitínových, kterou vydalo nakladatelství *Grada* v roce 1998.

Je přirozené, že váš pohled na knihu bude jiný než můj. Pokud zjistíte, že v knize schází některá fakta, bez kterých si nevíte rady, napište na adresu [sdds@seznam.cz](mailto:sdds@seznam.cz) a já se pokusím vám stručně vysvětlit souvislosti.

### Používané konvence

Jak jste již z publikací nakladatelství *Grada* zvyklí, orientaci v textu vám budou usnadňovat různé typografické prvky:

<b>Pozor!</b>	Důležité pojmy a pasáže textu, které je třeba zvýraznit, jsou vysazeny <b>tučně</b> .
<b>Název</b>	Názvy firem, softwarových produktů, aplikací a jednotlivých objektů programu označuji <i>kurzívou</i> .
<b>Soubor</b>	Názvy souborů, složek a internetové adresy.
<b>Storno</b>	Texty, které se objevují v uživatelském prostředí aplikací <i>Windows</i> , jsou vysazeny <b>jako běžný text, ale v tučné kurzívě</b> . Takto jsou odlišeny příkazy nabídek, popisky ovládacích tlačítek, názvy dialogových oken a další „citace z obrazovky“.

<b>Vložit → Rám</b>	Jednotlivé příkazy v posloupnosti příkazů zadávané v nabídkách, podnabídkách a následně otevíraných dialogových oknech oddělujeme šipkami – např. <b>Nástroje → Možnosti → Zobrazení → Zalomit do okna</b> .
<b>KLÁVESY</b>	Názvy kláves a klávesových zkratk označujeme KAPITÁLKAMI – např. ENTER nebo ALT+S.
<b>Program</b>	Pro výpis zdrojového kódu v příkladech a pro odlišení jednotlivých prvků programu v běžném textu je použito bezpatkové neproporcionální písmo.

V textu se budete často setkávat s odstavci označenými ikonou, která bude charakterizovat druh informace v daném odstavci:



*Pišící ruka označuje poznámku, která není nezbytná k pochopení dané problematiky, ale týká se tématu a prozrazuje další souvislosti.*



*„Usměvák“ vás upozorní na různé tipy a triky, kterými si můžete usnadnit některé činnosti nebo které vám umožní dosáhnout efektivních výsledků.*



*„Mračoun“ označuje text, v němž vás upozorňuji na něco, co nefunguje tak, jak byste očekávali nebo jak popisuje dokumentace.*

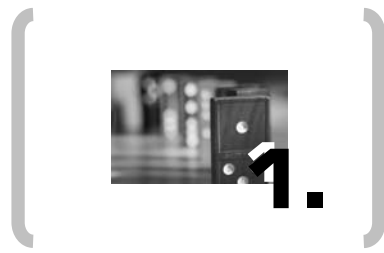


*Varovně vztyčený prst označuje text, který vás upozorňuje na něco, na co byste si měli dát pozor, co vás může nepříjemně překvapit nebo co by vám mohlo způsobit problémy.*



*Bomba je předzvěstí katastrofy nebo alespoň velkých nepříjemností. Tato ikona totiž označuje text upozorňující na skutečnosti, vedoucí ke ztrátě dat, zhroucení systémů a podobným havarijním stavům.*

Takto označené pasáže textu jsou použity v případě, kdy je třeba na chvíli přerušit normální běh výkladu a upozornit na skutečnosti, které je třeba znát pro pochopení další látky, nebo na skutečnosti, které by mohly být pro čtenáře zajímavé, ale které s vykládanou látkou souvisí pouze nepřímo.



## **Makra v Excelu**

## 1. Makra v Excelu

Nejen u počítačů, ale i u jiných oborů se vyplatí držet se přísloví „dvakrát měř a jednou řež“. U programování však nejde jen o to, abyste přílišným spěchem nevyrobili příliš mnoho chyb, jak říká původní smysl staré lidové moudrosti. Základní otázkou tu je, zda má vůbec smysl se do něčeho podobného pouštět. A náplní první kapitoly jsou právě odpovědi na otázky: „K čemu je programování? Proč bych to měl dělat? Jak se do toho pustit?“

### 1.1 Proč je v Excelu obsažen i programovací jazyk

Čím více zkušeností s prací v *Excelu* budete mít, tím rychleji zjistíte, že mnohdy určité činnosti a operace stále a stále opakuje – a to tak často, že zatoužíte po nějakém „pomocníkovi“, který by tyto operace dělal za vás.

V mnoha případech vám bude některá operace připadat příliš složitá – např. zapisování vstupních údajů do složitě strukturovaných tabulek, ve kterých budete muset pro každý nový údaj vyhledat příslušný sloupec či řádek. Ztráta času bude dost velká, a vy opět začnete uvažovat o „pomocníkovi“, který by celou činnost zrychlil.

Tyto pocity zažívaly tisíce uživatelů před vámi, takže do většiny populárních kancelářských (i jiných) programů byly začleněny i prostředky pro psaní maker (má je nejen *Excel*, ale i *Word* a další programy). *Makro* je sled příkazů, jehož cílem je automatizovat některé pracovní postupy, vyloučit tak opakované operace a snížit počet případných chyb při zpracování dat.

#### Ve kterých případech je vhodné používat makra?

Otázku je možné položit i jinak – co můžete pomocí maker udělat? Možností je mnoho, přibližně si alespoň ty nejdůležitější:

- 1. Automatizace často prováděných úkonů.** Zkuste si sami vzpomenout, které činnosti provádíte vždy před odchodem z práce, na konci týdne, na konci měsíce, po vytvoření sešitu ... patřit sem může např.:
  - Otevření sešitu, vytištění jeho části a následné uzavření.
  - Otevření více sešitů najednou, sloučení jejich obsahu do nového sešitu a jeho uložení.
  - Vytváření grafů.
  - Stejně formátování buněk.
  - Tisk určité části sešitu.
- 2. Vytvoření vlastní funkce či příkazu.** Pomocí vlastní funkce se dají zjednodušit některé vzorce, vlastním příkazem můžete zkombinovat několik příkazů z nabídek *Excelu*.
- 3. Vytvořit celé nové aplikace.** Programovací jazyk většinou nabízí i vlastní dialogová okna nebo panely nástrojů, která výrazně zjednoduší práci s programem.

### Jak se dají makra vytvořit?

V zásadě jsou možné dva způsoby :

- Makro si můžete **nahrát** pomocí záznamníku maker. Při nahrávání makra jsou všechny činnosti, které v *Excelu* provádíte, automaticky zapisovány do tzv. modulu. Po zastavení záznamníku můžete nahrané makro okamžitě spouštět, ve většině případů je ovšem nutné ho trochu upravit. Výhody a nevýhody záznamníku maker jsou popsány ve druhé kapitole.
- Makro můžete napsat ručně, ať již celé, nebo pouze upravit jiné, již existující. Tento přístup má mnoho výhod, ale jednu zásadní nevýhodu – musíte ovládat daný programovací jazyk.

### Kdo makra obvykle vytváří?

- Každodenní uživatelé *Excelu*, ovládající programování špatně či vůbec, kteří si ale dokáží makro nahrát a urychlit tak některé úkoly.
- Vývojáři (programátoři), jejichž úkolem je vytvářet rozsáhlejší tabulkové aplikace pro pracovníky vlastního podniku či zákazníky.

### Jak se dají hotová makra spouštět?

Hotové makro je uloženo v modulu, kde ho lze upravit či spustit, běžný uživatel se ovšem s modulem v praxi přímo neseťká (to by musel otevřít editor jazyka *Visual Basic*). Proto je nutné, aby mohl spouštět makra přímo z uživatelského prostředí. *Excel* v tomto ohledu nabízí dostatečné množství variant:

- **Makro můžete vždy spustit z dialogového okna Makro**, které otevřete příkazem **Nástroje → Makro → Makra** (nebo klávesovou zkratkou ALT+F8). V dialogovém okně **Makro** vidíte seznam všech dostupných maker, pokud některé z nich v seznamu označíte, stačí pak už jen stisknout tlačítko **Spustit**.
- **Makru můžete přiřadit klávesovou zkratku**. Z určitých nepochopitelných důvodů vám však *Excel* v dialogovém okně možností makra nabízí jen zkratky typu CTRL+PÍSMENO, případně CTRL+SHIFT+PÍSMENO. Neznamená to, že byste makro nemohli spouštět i jinými kombinacemi kláves, ale ty již musíte určitým způsobem naprogramovat.
- **Makro se dá spouštět určitým příkazem nabídky**. Příkaz pro spuštění makra se dá přidat do libovolné nabídky, dokonce si můžete vytvořit i vlastní nabídku.
- **Makro můžete přiřadit nějakému tlačítku na panelu nástrojů**. Jde o velmi často používanou možnost, protože toto tlačítko můžete graficky upravit tak, aby vizuálně naznačovalo, jaké makro se po jeho stisknutí spustí.
- **Makro dále můžete přiřadit tlačítku umístěnému v listu či na formuláři, nebo libovolnému grafickému objektu**. I tato možnost se hojně využívá, zvláště u maker určených k provádění operací na konkrétním listu.

Kromě toho umí *Excel* spustit makro (programový kód) i sám, bez přispění uživatele. Využívá při tom tzv. **události**, které probíhají při určitých akcích, např. otevření sešitu, tisk sešitu, aktivace jiného listu apod.

## 1.2 Něco málo o vývoji Excelu z hlediska programování

Excel má dost bohatou historii verzí, i když v dnešní době ty starší (2, 3, 4, 5, 95) jsou již dávno mimo hru; těžko se najde někdo, kdo by je ještě používal (*Microsoft* je již také žádným způsobem nepodporuje).

Ve starých verzích Excelu byl programovací jazyk XLM (viz další část), v Excelu 5 se poprvé objevil jazyk *Visual Basic for Applications* (dále VBA).

Pro nás je výchozí až Excel 97, který se objevil začátkem roku 1997, buď samostatně nebo jako součást balíku aplikací *Office 97*. (Někdy je označován jako *Excel 8*, hlavně v názvech souborů nápověd.) Excel 97 totiž přišel se zcela novým uživatelským rozhraním pro programování – s moduly maker se pracuje ve zvláštním editoru *Visual Basicu*, starý způsob psaní kódu přímo v uživatelském prostředí Excelu byl opuštěn (moduly jsou ovšem stále ukládány jako součást sešitů XLS). Značně se tak zvýšila efektivita práce programátorů, protože staré nástroje se s pohodlím nového editoru nemohly srovnávat. Kupodivu, tento editor byl v českém Excelu 97 také přeložen do češtiny, zatímco v české verzi Excelu 2000 a novějších zůstal anglický.

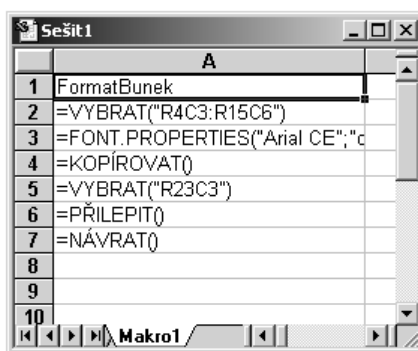
V dalších verzích Excelu se již nic zásadního nedělo, samozřejmě každá verze přinesla pár novinek, obvykle souvisejících s vylepšenými nebo novými funkcemi.

## 1.3 Základní údaje o zastaralém makrojazyku XLM

Přestože v této knize se jazyk XLM učit nebudete, vysvětlím stručně, o co šlo. Makra v jazyku se zapisovala do zvláštních typů listů (oficiálně se tento typ listu jmenuje *Makro Microsoft Excel 4.0*) jako posloupnosti volání jednotlivých funkcí.

V Excelu 97 a novějších verzích již nelze makra v jazyku XLM nahrávat, již existující makra však můžete spouštět nebo upravovat (listy s těmito makry se objeví přímo v Excelu, není nutné spouštět editor *Visual Basicu*).

A proč se tu o tomto jazyku vůbec zmiňuji? Inu, máte šanci – asi tak půl promile – na to, že se k vám někdy sešit s makry napsanými v tomto jazyku dostane. Tak abyste nebyli překvapeni. Jak by takové makro vypadalo, vidíte na následujícím obrázku.



Obr. 1.1  
Ukázka jednoduchého makra v jazyku XLM.



## 1.4 Typy listů v sešitu a jejich vztah k programování

Celkem existuje pět typů listů, které se v sešitu *Excelu* mohou objevit:

- pracovní listy (angl. **worksheets**)
- listy typu graf (**charts**)
- listy maker *XLM* (zastaralé, stále se však podporují)
- listy dialogů *Microsoft Excel 5.0* (zastaralé, jsou však stále podporovány)
- mezinárodní list maker (relikt z prastarých verzí *Excelu*; jeho význam je utajen již v ná-povědě k *Excelu 95*)

Nejběžnějším typem listu je tzv. **pracovní list**. Tento list má většina lidí na mysli, pokud mluví o tabulkách. Každý pracovní list *Excelu 97* a vyšších verzí má 256 sloupců a 65 536 řádků (což je čtyřnásobek počtu řádků, který byl k dispozici ve starších verzích).

S prvními dvěma typy listů pracujete přímo v sešitu. Makra jazyka *VBA* jsou ukládána výhradně v modulech, se kterými je možné pracovat jen v programovacím editoru.

Kromě maker, která jsou složena jen z kódu, můžete naprogramovat i vlastní dialogová okna, tedy grafický prvek uživatelského rozhraní, který lze zobrazit přímo nad pracovním listem *Excelu*. Ve starších verzích *Excelu* se tato dialogová okna vytvářela na speciálních listech dialogů. Novější verze (počínaje *Excetem 97*) sice stále tyto listy podporují, nabízejí však daleko lepší alternativu: formuláře **UserForm**.





## **Nahrávání maker**