

Doporučuje časopis  
**PC WORLD**



soubory na  
[www.grada.cz](http://www.grada.cz)

# Programování v Excelu 2000, 2002, 2003

Jaroslav Černý

- Události v Excelu a práce s grafy pomocí jazyka VBA
- Práce s externími soubory, export a import souborů
- Vlastní nabídky a panely nástrojů
- Spolupráce Excelu s dalšími programy
- Formuláře, databáze a seznamy v Excelu
- Třídy objektů a moduly tříd



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, umísťování 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.

## Obsah

	<b>Úvod.....</b>	<b>10</b>
	<b>Události v Excelu .....</b>	<b>13</b>
	1.1 Využití událostí ve vlastních aplikacích.....	15
	Umístění procedur událostí.....	15
	Jak zabránit výskytu událostí.....	18
	1.2 Události sešitu.....	19
	Přehled událostí sešitu .....	19
	Příklady využití základních událostí sešitu.....	21
	1.3 Události listů .....	23
	Přehled událostí listu.....	24
	Přehled událostí grafů (listu typu graf).....	28
	1.4 Události aplikace Excel .....	31
	<b>Práce s grafy pomocí jazyka VBA .....</b>	<b>33</b>
	2.1 Úvod.....	34
	Grafy v objektovém modelu VBA .....	34
	2.2 Základní postupy při práci s grafy .....	37
	Vytvoření nového grafu pomocí kódu VBA .....	37
	2.3 Práce s datovými řadami .....	42
	Kolekce SeriesCollection .....	42
	Práce s jednotlivými datovými řadami.....	45
	Pokročilé techniky práce s grafy.....	48
	Automatické přizpůsobení hodnot osy Y .....	48
	2.4 Formátování grafu.....	51
	2.5 Přehled vlastností a metod objektů.....	52
	Vlastnosti objektu ChartObject .....	52
	Metody objektu ChartObject .....	54
	Vlastnosti objektu Chart .....	54
	Metody objektu Chart .....	56
	<b>Třídy objektů a moduly tříd.....</b>	<b>57</b>
	3.1 Proč, k čemu, jak .....	58
	3.2 Vytvoření nové třídy a konkrétní instance.....	59
	První pokus .....	60
	Vytváření vlastností a metod .....	62
	Zachycení událostí objektu Application pomocí třídy .....	67
	Zachycení událostí vložených grafů.....	68
	Příklad: použití událostí u vloženého grafu.....	69



## Formuláře ..... 73

4.1	Vytvoření formuláře, jeho zobrazení a ukončení pomocí VBA .....	75
	Modul kódu formuláře .....	76
	Toolbox (souprava nástrojů) .....	76
	Spuštění formuláře .....	77
	Modální a nemoďální formuláře .....	78
	Uzavření formuláře .....	79
4.2	Okno Properties .....	80
4.3	Ovládací prvky .....	82
	Přidání ovládacího prvku do formuláře .....	82
	Výběr ovládacího prvku na formuláři .....	84
	Konvence pro názvy formulářů a ovládacích prvků .....	85
	Pořadí ovládacích prvků na formuláři .....	86
	Dostupnost ovládacích prvků a přesun mezi nimi .....	88
	Úprava ovládacích prvků ve formuláři .....	89
4.4	Přehled typů ovládacích prvků .....	91
	Příkazové tlačítko (CommandButton) .....	91
	Popisek (Label) .....	92
	Textové pole (TextBox) .....	93
	Zaškrťávací políčko (CheckBox) .....	94
	Přepínač (OptionButton) .....	95
	Seznam (ListBox) .....	96
	Rozvírací seznam (ComboBox) .....	99
	Rámeček (Frame) .....	100
	Přepínací tlačítko (ToggleButton) .....	101
	Číselník (SpinButton) .....	101
	Posuvník (ScrollBar) .....	102
	RefEdit .....	102
	Vícenásobná stránka (MultiPage) .....	102
	Karty (TabStrip) .....	103
	Obraz (Image) .....	104
4.5	Odkazy na formuláře a ovládací prvky v kódu VBA .....	105
	Příklad – formulář pro zobrazení informací o listu .....	106
	Další příklad – seznam všech pojmenovaných názvů v sešitu ..	108
	Ukázka práce se zaškrťávacími políčky a přepínači .....	111
4.6	Zpracování událostí souvisejících s klávesnicí a myší .....	113
	Události klávesnice .....	115
4.7	Používání ovládacích prvků přímo na pracovním listu .....	116
4.8	Základní společné vlastnosti a metody ovládacích prvků a uživatelských formulářů .....	119



	Vlastnosti ovlivňující vzhled a chování formuláře či ovládacích prvků .....	119
	Další společné vlastnosti .....	121
	Metody formuláře nebo ovládacích prvků .....	121
4.9	Přehled důležitých vlastností a metod objektu UserForm .....	122
	Vlastnosti .....	122
	Metody .....	124
	Základní události formuláře .....	124
4.10	Sdílení formulářů .....	124

**Práce s externími soubory, export a import souborů ..... 127**

5.1	Práce s externími soubory .....	128
	Základní operace se soubory .....	128
5.2	Jednoduché čtení a zápis textových souborů .....	132
5.3	Vlastní manipulace s textovými soubory .....	136
	Otevření textového souboru .....	137
	Načtení obsahu textového souboru .....	138
	Zápis údajů do textového souboru .....	139
	Import dat do listu Excelu pomocí příkazu Line Input # .....	139
	Načítání textového souboru příkazem Input # .....	141
	Zápis do souborů – příkazy Write # a Print # .....	141
	Export oblasti buněk do textového souboru pomocí příkazu Write # .....	142
	Využití příkazu Write pro zápis údajů o chybách za běhu programu .....	143
5.4	Ukládání různých nastavení do registru Windows .....	144
	Použití příkazů SaveSetting, DeleteSetting a funkcí GetSetting, GetAllSettings .....	145
	Další možnosti pro ukládání nastavení .....	145



**Databáze a seznamy v Excelu ..... 147**

	Základní terminologie .....	149
	Ruční správa seznamu pomocí nabídky Excelu .....	149
6.1	Vytvoření seznamu .....	149
	Pořizování záznamů v seznamu .....	150
6.2	Řazení, filtrování a další operace s databázemi Excelu pomocí VBA .....	150
6.3	Příklad jednoduché aplikace využívající formuláře [Adresář] .....	155
	Jak načítat a ukládat data .....	156
	Činnosti při otevření formuláře .....	157



Obecné procedury pro načtení záznamu, uložení původních hodnot a uložení změn .....	158
Tlačítka pro přechod na další [předchozí záznam] .....	161

## **Pracovní prostředí Excelu – nabídky a panely nástrojů ..... 165**

7.1 Části pracovní plochy Excelu a jejich zobrazení či skrytí pomocí VBA .....	166
7.2 Panely nabídek a panely nástrojů – možnosti manipulace s nimi .....	170
Kolekce CommandBars, objekty CommandBar .....	170
Objekt CommandBarControl a objekty vyjadřující jednotlivé typy ovládacích prvků .....	172
7.3 Práce s nabídkami a panely nástrojů.....	173
Vytváření a odstranění vlastních panelů a nabídek .....	173
Úprava existujících panelů a nabídek .....	175
7.4 Místní nabídky.....	181
Vytvoření, úprava a odstranění nové místní nabídky.....	182
7.5 Přehled vlastností a metod objektů	
CommandBars a CommandBar .....	183
Vlastnosti kolekce CommandBars .....	183
Metody kolekce CommandBars .....	183
Vlastnosti objektu CommandBar.....	184
Metody objektu CommandBar .....	185
Základní vlastnosti a metody ovládacích prvků na panelech příkazů.....	186



## **Doplňky v Excelu..... 187**

8.1 Účel doplňků a přehled doplňků dodávaných přímo s Excelem .....	189
8.2 Obecné zásady pro vytváření doplňků .....	190
8.3 Příklad doplňku – převod do HTML.....	192
Formulář.....	192
Programový kód doplňku .....	193



## **Spolupráce s dalšími programy ..... 197**

9.1 Spouštění a aktivace jiných aplikací .....	198
Funkce Shell .....	198
Spuštění programů pomocí technologie Automation.....	199
Přepínání mezi několika programy .....	202
9.2 Technologie Automation a její využití pro spolupráci s Wordem nebo Outlookem .....	203
Příklady práce s Wordem.....	203
Příklady práce s Outlookem .....	205





<b>Funkce Windows API.....</b>	<b>211</b>
10.1 Co se skrývá za zkratkou API .....	212
Deklarace API funkcí a příkazů .....	212
10.2 Konstanty a uživatelské typy proměnných v API .....	213
Další „lahůdky“ spojené s voláním rutin API .....	215
Několik varování .....	216
10.3 Základní příklady na používání Windows API .....	216
Práce s disky, složkami a soubory .....	217
Práce se systémem Windows.....	219
API pro práci se sítí.....	221
Další vhodné ukázky .....	222
Kde hledat dále? .....	223
<b>Rejstřík .....</b>	<b>225</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.

Tato kniha volně navazuje na knihu s názvem *Excel 2000, 2002, 2003 – záznam, úprava a programování maker (Grada, 2004)*, v níž se věnuji těm záležitostem, které programátor *Excelu* prostě „musí“ znát, a těm technikám, které bude v *Excelu* používat nejčastěji. Kniha, již však držíte v rukou právě teď, je učebnicí pokročilých programovacích technik a objektů.

V celém textu se snažím být maximálně stručný, což je přirozený důsledek dlouholeté zkušeností 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 spousta 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 [sd@s@seznam.cz](mailto:sd@s@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> .
<i>Název</i>	Názvy firem, softwarových produktů, aplikací a jednotlivých objektů programu označuji <i>kurzivou</i> .
Soubor	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 kurzivě</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 odděluje šipkami – např. <b>Nástroje → Možnosti → Zobrazení → Zalomit do okna</b> .
KLÁVESA	Názvy kláves a klávesových zkratk označujeme KAPITÁLKAMI – např. ENTER nebo ALT+S.
Program	Pro výpis zdrojového kódu v příkladech a pro odlišení jednotlivých prvků programu v běžném textu je použito bezpatkové neproportcionální písmo.

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



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



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



*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.*





## **Události v Excelu**

## 1. Události v Excelu

V prvním dílu učebnice byly události zmíněny jen velmi stručně, nyní přišel čas věnovat se jim podrobněji. V dalších kapitolách se s jejich využitím budete setkávat velmi často.

Připomeňme si, co už víte – událost (angl. *event*) je **akce vyvolaná uživatelem nebo systémem, kterou objekt** (obecně, nejenom v *Excelu*) **dokáže rozpoznat**. Mezi takové akce patří např. otevření či uzavření sešitu, klepnutí či poklepání myši na objekt, změna hodnoty v buňce apod.

Každá událost má přiřazenu svoji proceduru (říká se jí **událostní procedura**, angl. *event procedure*), která standardně neobsahuje žádný kód. Napíšete-li však nějaký kód do událostní procedury, bude spuštěn při každém výskytu dané události, tedy např. při otevření nějakého sešitu nebo před jeho uzavřením, před přepočítáním sešitu, po změně hodnoty v buňce apod.

Objekty *Excelu* umí reagovat na velké množství událostí. Rozsah této knihy proto umožní představit si jen některé z nich, stručný přehled všech událostí je uveden vždy v příslušné části kapitoly:

- **Události sešitu:** Probíhají v rámci sešitu. Jako příklad lze uvést událost `Open` (vytvoření nebo otevření sešitu), `BeforeSave` (sešit bude uložen), `BeforePrint` (sešit bude tištěn) nebo `NewSheet` (je přidáván nový list).
- **Události listu:** Jsou vyvolány pro určitý list. Sem patří např. událost `Activate` (list je aktivován), `SelectionChange` (změna výběru na listu) či `Calculate` (list je přepočítáván).
- **Události grafu:** Jsou vyvolány pro určitý list typu graf. Patří mezi ně události `Resize` (po změně velikosti grafu) či `SeriesChange` (změna datového bodu v nějaké datové řadě). Pro zachytávání událostí v grafech vložených na pracovním listu je nutné použít modul třídy (ty budou podrobně probrány ve třetí kapitole).
- **Události celé aplikace:** Probíhají na úrovni celé aplikace (*Excelu*). Mezi nejvýznamnější patří např. `NewWorkbook` (vytvoření nového sešitu) nebo `SheetChange` (proběhla změna buňky v libovolném sešitu). Pro práci s událostmi aplikace je vždy nutný modul třídy.
- **Události formulářů a jejich ovládacích prvků:** Vyskytují se v rámci určitého formuláře nebo ovladače.
- **Události, které neprobíhají v rámci žádného objektu:** Sem patří dvě události na úrovni celé aplikace – `OnTime` a `OnKey`. Funkčně se odlišují od jiných událostí.

Některé akce spouští více událostí za sebou. Např. při otevírání sešitu proběhnou postupně tyto události sešitu:

- `Open`
- `Activate`
- `WindowActivate`

Při uzavírání sešitu probíhají (minimálně) tyto události sešitu:

- BeforeClose
- WindowDeactivate
- Deactivate

Posloupnosti událostí bývají mnohem složitější, než si můžete představit na základě těchto příkladů. V obou ukázkách šlo navíc jen o události na úrovni sešitu. Např. při přidání nového listu do sešitu však proběhnou i události na úrovni listu a celé aplikace. Posloupnost událostí navíc nemusí být vždy zcela logická – při přidání nového listu do sešitu proběhne nejdříve událost `SheetActivate` (aktivace listu) a až poté událost `WorkbookNewSheet` (nový list v sešitu).

Další sekvence můžete zkoumat sami podle libosti, nejjednodušší je zapsat do každé zkoumané událostní procedury příkaz:

```
Debug.Print "Název_Procedury"
```

Bližší průzkum posloupností, v jakých jednotlivé události probíhají, se ovšem v praxi vyplatí pouze v případech, kdy využíváte větší množství událostí. Chcete-li spustit jednu proceduru při otevření sešitu a druhou při přepočtu určitého listu, není třeba se ničeho obávat.

## 1.1 Využití událostí ve vlastních aplikacích

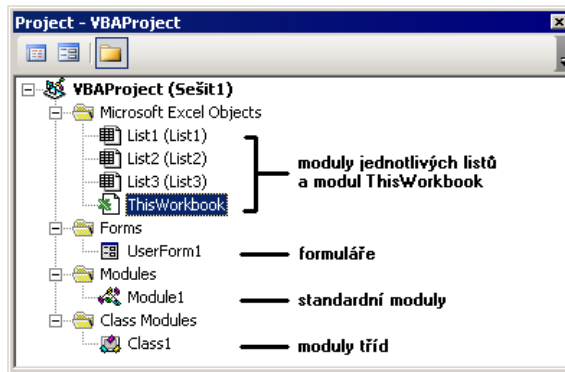
Aplikace řízené událostmi jsou v systému *Windows* rozšířeny tak, že už si to ani neuvědomujeme. Každé dialogové okno je typickou ukázkou – po klepnutí na tlačítko nebo po změně hodnoty určitého ovladače je okamžitě provedena nějaká operace. Klepnutí nebo poklepání myši, pohyb kurzorem nebo stisk klávesy jsou také událostmi, které je každý prvek v dialogovém okně schopen rozeznat.

### Umístění procedur událostí

Velmi častou chybou mnoha začátečníků (v diskusních konferencích narazíte na spoustu dotazů) je fakt, že procedury, které by měly při vzniku nějaké události proběhnout, nejsou spuštěny. Příčina problému je jednoduchá – událostní procedury je třeba vždy zapsat na správné místo.

V okně **Project** může mít každý sešit tyto části:

- **Objekty jednotlivých listů:** Jejich počet odpovídá počtu pracovních listů v sešitu.
- **Objekty jednotlivých listů typu graf:** Přidáte-li do sešitu graf na samostatném listu.
- **Objekt *ThisWorkbook*:** Představuje celý sešit.
- **Standardní moduly VBA ve složce *Modules*:** Do těchto modulů nikdy neumístíte událostní procedury. Jejich hlavičky se ve standardních modulech ani nevyskytují, musely by se tam ručně dopisovat.
- **Moduly tříd:** Události v modulech tříd si vysvětlíme ve třetí kapitole.
- **Formuláře:** Podrobný výklad událostí u formulářů je ve čtvrté kapitole.



**Obrázek 1.1**  
Různé části projektu Excelu.

Když myší poklepnete na objekt listu nebo objekt `ThisWorkbook`, otevře se okno kódu daného objektu. V jeho horní části jsou dva rozevřací seznamy. Levý seznam obsahuje řádek (General) a řádky Worksheet, Chart nebo Workbook. Část General znáte z klasických modulů: slouží pro deklarování proměnných, které mají svůj rozsah platnosti na úrovni modulu, v praxi tu tedy budete deklarovat proměnné, jež budete používat ve více událostech, probíhajících v daném modulu.



*Pozor na to, že proměnné, které deklarujete v modulu `ThisWorkbook` nebo v modulech listů jako veřejné (`Public`), nejsou dostupné z žádného jiného modulu, tedy ani z jiného modulu listu nebo ze standardních modulů! Veřejné proměnné musíte vždy deklarovat klíčovým slovem `Public` ve standardním modulu!*

*Stejně pravidlo platí i pro všeobecné procedury – ty musí být umístěny také ve standardním modulu. VBA vám sice umožňuje vkládat je i do modulů listů nebo do modulu `ThisWorkbook`, ale takové procedury je možné spouštět jen v rámci daného modulu!*

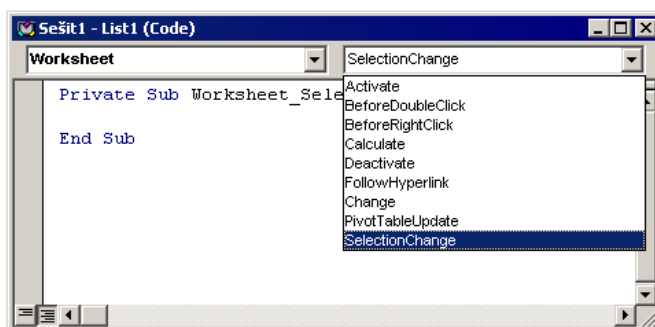
Po krátké odbočce se vraťme zpět k oknu kódu pro moduly listů či modul `ThisWorkbook`. Vyberete-li v levém seznamu druhý řádek (Worksheet, Chart nebo Workbook), objeví se v pravém seznamu všechny události, které můžete pro list či sešit využít.

Jakmile v pravém rozvíracího seznamu vyberete nějakou událost, vloží editor VBA do okna modulu hlavičku odpovídající (prázdné) procedury:

```
Private Sub Workbook_Activate()  
  
End Sub
```

Veškerý kód zapsaný mezi tyto dva řádky bude proveden při každém vyvolání dané události.





**Obrázek 1.2**

V pravém rozevřacím seznamu okna kódu vidíte události, které můžete zachytávat na úrovni listu či sešitu.

Některé událostní procedury obsahují v závorce za názvem procedury ještě seznam parametrů. Ty jsou **vestavěny** a vy jejich deklarace v žádném případě nemůžete měnit! Stejně tak nelze do seznamu parametrů přidávat parametry nové.

Parametry událostí se vyskytují tam, kde jedna událost může proběhnout pro více různých objektů, nebo kde vám systém předává ještě dodatečné údaje k dané události. Např. u události pohybu myši nad objektem (*MouseMove*) dostanete předány souřadnice kurzoru (při události stisknutí tlačítka myši – *MouseDown* – zase údaj o tom, které tlačítko bylo stisknuto). Dalším příkladem je událost *SheetActivate* (je na úrovni sešitu, proto ji najdete v modulu *ThisWorkbook*) proběhne při aktivaci některého listu v sešitu. Aby bylo možné určit, který list byl aktivován, je odkaz na aktivovaný list (tedy na objekt *Worksheet* nebo *Chart*, deklarovaný ovšem jako *Object*) předán jako parametr procedury:

```
Private Sub Workbook_SheetActivate(ByVal Sh As Object)
End Sub
```

Deklarace parametru jako typ *Object* má v tomto případě určité nepříjemné důsledky. Musíte totiž rozlišovat, zda byl aktivován list typu graf nebo pracovní list. Např. následující kód proběhne v pořádku, bude-li aktivován pracovní list:

```
Private Sub Workbook_SheetActivate(ByVal Sh As Object)
    MsgBox Sh.Name           ' název listu
    MsgBox Sh.Rows.Count     ' počet řádků v listu
End Sub
```

Při aktivaci listu typu graf by pochopitelně došlo k chybě za běhu programu ve druhém řádku, protože list typu graf neobsahuje žádné řádky a tudíž odkaz na kolekci *Rows* není platný. Jak si můžete ověřit, který typ listu byl aktivován? Slouží k tomu operátor *TypeOf*, volaný např. v bloku *If ... End If*:

```
' zde bude operace, kterou je možné provádět jen pro pracovní listy
Private Sub Workbook_SheetActivate(ByVal Sh As Object)
    If TypeOf Sh Is Worksheet Then
    ElseIf TypeOf Sh Is Chart
        ' a zde operace pro listy typu graf
    End If
End Sub
```

V některých událostních procedurách se objevuje logický parametr `Cancel`. Např. deklarace událostní procedury `BeforeClose` v modulu `ThisWorkbook` vypadá takto:

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
End Sub
```

Výchozí hodnota předaného parametru `Cancel` je `False`. Jestliže však ve vlastním kódu procedury nastavíte tento parametr na `True`, nebude sešit uzavřen. Ukázkou možného využití představuje zákaz uzavření sešitu, zapomněl-li uživatel vytisknout denní přehled:

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    dotaz = "Vytisknuli jste denní přehled změn?"
    odpoved = MsgBox(dotaz, vbYesNo, "Ukončení práce...")
    If odpoved = vbNo Then Cancel = True
End Sub
```

Procedura `Workbook_BeforeClose` je spuštěna po vydání příkazu k uzavření sešitu (**Soubor** → **Zavřít**, **Soubor** → **Konec** apod.). Procedura nejdříve zobrazí okno hlášení a odpoví-li uživatel záporně klepnutím na tlačítko **Ne**, bude parametr `Cancel` nastaven na hodnotu `True`. Uzavření sešitu je poté zrušeno.

### Jak zabránit výskytu událostí

V praxi se setkáte se situacemi, kdy máte pro určitou událost napsán kód, ale tento kód může být prováděn jen někdy. Např. při otvírání sešitu nebude z nějakého důvodu žádoucí, aby byl proveden kód události `Activate`. Při každé další aktivaci daného sešitu však už kód proveden být musí.

Obdobně se můžete potkat s případy, kdy v kódu událostní procedury provedete něco, čím danou událost znovu spustíte. Změníte-li např. v události listu `Change` hodnotu buňky v tomto listu (příklad si ukážeme za chvíli), bude událost `Change` generována znovu. V takovém případě ovšem hrozí nekonečné zacyklení kódu.

Existuje dvojí možné řešení těchto situací. Buď zachytávání událostí zcela vypnete, nebo použijete logickou proměnnou, která ukončí provádění kódu. První řešení je „lepší“, protože událost vůbec není vyvolána, ovšem jeho nevýhodou je fakt, že neproběhnou vůbec žádné události. Logická proměnná nic nepotlačuje, jen bezprostředně ukončí provádění kódu v událostní proceduře.

Zachytávání událostí vypnete takto:

```
Debug.Print "Název_Procedury"
```

Vlastnost `EnableEvents` má logické hodnoty `True/False`, za řádkem, který by vyvolal nežádoucí generování událostí, zase zachytávání zapnete.

Ukázka druhého přístupu pomocí logické proměnné potlačí generování události `Activate` při otevření sešitu: