

MORAVSKÁ VYSOKÁ ŠKOLA OLMOUC

Ústav informatiky a aplikované matematiky

Jiří Trbušek

**Řešení ekonomických úloh pomocí maker a programů
v Excelu**

Solving of Economic Tasks Using Macros and Programs in Excel

Bakalářská práce

Vedoucí práce: Mgr. Aleš Keprt, Ph.D.

Olomouc 2014

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a použil jen uvedené informační zdroje. Prohlašuji, že tištěná verze bakalářské práce se shoduje s elektronickou verzí vloženou do IS/STAG.

Olomouc

Děkuji Mgr. Aleši Keprtovi, Ph.D. za odborné vedení bakalářské práce.

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Jiří TRBUŠEK
Osobní číslo: M11185
Studijní program: B6208 Ekonomika a management
Studijní obor: Podnikové informační systémy
Název tématu: Řešení ekonomických úloh pomocí maker a programů v Excelu
Téma anglicky: Solving of Economic Tasks Using Macros and Programs in Excel
Zadávací katedra: Ústav informatiky a aplikované matematiky

Z á s a d y p r o v y p r a c o v á n í :

Využití Excelu a VBA pro tvorbu programu na zpracování účetnictví, finanční a statistickou analýzu. Praktické využití funkcí integrovaných v Excelu, které mohou manažerovi usnadnit práci a pomoci při rozhodování.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

WEBER, Monika a BREDEN, Melanie. Excel VBA: velká kniha řešení. Vyd. 1. Brno: Computer Press, 2007. 867 s. Programování. ISBN 978-80-251-1453-7.
ČECHOVÁ, Alena. Manažerské účetnictví. 2., aktualiz. a rozš. vyd. Brno: Computer Press, 2011. vi, 194 s. ISBN 978-80-251-2831-2. TULEJA, Pavel. Analýza pro ekonomy. Vyd. 1. Brno: Computer Press, 2007. vi, 336 s. Vysokoškolské učebnice. ISBN 978-80-251-1801-6. BROŽ, Milan a BEZVODA, Václav. Microsoft Excel 2007 pro manažery a ekonomy. Vyd. 1. Brno: Computer Press, 2009. 435 s. ISBN 978-80-251-2116-0. KRÁL, Martin. Excel VBA: výukový kurz. Vyd. 1. Brno: Computer Press, 2010. 504 s. ISBN 978-80-251-2358-4.

Vedoucí bakalářské práce:

Mgr. Aleš KEPRT, Ph.D.

Ústav informatiky a aplikované matematiky

Datum zadání bakalářské práce: 4. února 2013

Termín odevzdání bakalářské práce: 28. března 2014

Podpis studenta: *T. M. J.* Datum: 31. 5. 2013

Podpis vedoucího práce: *A. Kepř* Datum: 11. 2. 2013

Vaněčková
Mgr. et Mgr. Michaela VANĚČKOVÁ, Ph.D.
prorektorka



Mgr. Veronika KAINZOVÁ, Ph.D.
manažer ústavu *[Signature]*

V Olomouci dne 11. února 2013

Obsah

Úvod.....	8
1 Funkce a postupy pro zpracování účetnictví.....	10
1.1 Použití Excelu pro zpracování účetnictví	10
1.2 Rozvržení programu	10
1.3 List s účetní osnovou	11
1.4 List pro ukládání záznamů.....	13
1.5 List pro zadávání záznamů	13
1.6 Tvorba makra.....	15
1.7 Další kroky pro tvorbu programu	19
2 Kontingenční tabulka	19
2.1 Výhody použití kontingenční tabulky	19
2.2 Zdrojová data pro kontingenční tabulku.....	19
2.3 Příklad rozvržení a použití kontingenční tabulky.....	20
2.4 Příklad s použitím počítaného pole	22
2.5 Možnosti nastavení kontingenční tabulky	24
3 Předpovědi budoucích stavů	24
3.1 FORECAST.....	25
3.2 LINTREND	26
3.3 LOGLINTREND	28
4 Funkce pro výpočty úvěrů a investic.....	29
4.1 BUDHODNOTA	29
4.2 SOUČHODNOTA.....	30
4.3 PLATBA.ZÁKLAD a PLATBA.ÚROK	31
4.4 PLATBA.....	32
4.5 POČET.OBDOBÍ	32
4.6 ÚROKOVÁ MÍRA.....	32
5 Funkce pro Cash Flow.....	33
5.1 ČISTÁ.SOUČHODNOTA	33

5.2 MÍRA.VÝNOSNOSTI	34
Závěr.....	36

Úvod

Microsoft Excel (dále jen Excel), případně jiná alternativní tabulková aplikace, je jedním ze zásadních programů, které může ekonom nebo manažer v praxi používat. Jeho zvládnutí umožňuje uživateli provádět efektivní zpracování a analýzu dat, následně jejich grafickou interpretaci a podklady pro různé prezentace atd. Díky své všestrannosti dokáže v mnoha případech dokonce nahradit specializované programy. Navíc je velmi rozšířený a používá ho velké množství firem, takže jeho zvládnutí bývá často podmínkou pro úspěšné ucházení se o funkci ekonoma nebo analytika.

V mé praxi analytika je Excel programem, se kterým pracuji převážnou část pracovní doby. Používám jej pro analýzu dat a přípravu reportů. Prakticky denně se setkávám s problémy ekonomů, kteří potřebují poradit se zpracováním ekonomické úlohy v Excelu, protože neznají funkce, které lze pro výpočet použít nebo je neumějí správně aplikovat. Dané téma jsem si tedy zvolil proto, abych ukázal způsoby, jakými lze v Excelu řešit ekonomické úlohy, jaké funkce se používají a jakou mají syntaxi.

V první části této práce se zaměřím na postupy a metody, které Excel poskytuje, pro zpracování účetnictví. Mým cílem je ukázat, jak pomocí integrovaných funkcí a poměrně jednoduchých maker vytvořit základ účetního programu. Hlavním důvodem volby tohoto programu je to, že na něm lze ukázat funkce, které pracují s textovými řetězci, vyhledávací, podmiňovací apod. Také se budu věnovat tvorbě makra, které je pro chod tohoto programu důležité a zautomatizuje část zadávání záznamů do účetnictví. Uvedu popis struktury tohoto účetního programu, syntaxi používaných funkcí a výhody nebo nevýhody zvolených řešení. Nebudu popisovat veškeré formátování, které jsem použil pro zpřehlednění programu. Po vytvoření základní struktury programu popíši obecně možnosti dalších rozšíření, které jsou sice pro práci s programem potřebné, ale jejich tvorba by vyžadovala opětovné užití postupů, které jsem již uvedl.

Samostatnou kapitolu budu věnovat kontingenční tabulce, která je neopomenutelnou součástí Excelu a umožňuje velmi rychlou analýzu dat díky křížovému náhledu na data a agregačním funkcím. Uvedu popis struktury kontingenční tabulky, práci s daty, možnosti nastavení a příklady použití.

V dalších částech se budu věnovat funkcím, které Excel nabízí, pro analýzu dat. Zaměřím se na funkce pro předpovědi budoucích stavů na základě předchozího vývoje,

výpočty parametrů úvěrů, investic a cash flow. Opět uvedu popis funkcí, syntaxi a příklady použití v praxi.

1 Funkce a postupy pro zpracování účetnictví

1.1 Použití Excelu pro zpracování účetnictví

Účetnictví sleduje stav a pohyb veškerého majetku, závazků, kapitálu, náklady, výnosy a výsledek hospodaření. Používá soustavu účtů, na které provádí záznamy podvojnými zápisy.¹

Vedení účetnictví v Excelu je jednou z možností, jak se vyhnout pořizování specializované aplikace. Účetnictví je ve své podstatě práce s daty v různých tabulkách, tím se pro jeho vedení nabízí využití tabulkového procesoru. Mezi nejznámější a nejpoužívanější patří Excel, který je v dnešní době téměř synonymem pro tabulkový procesor. Nabízí velké množství integrovaných funkcí a také editor jazyku Visual Basic for Applications (dále jen VBA), ve kterém je možné programovat vlastní funkce nebo automatizovat různé rutinní operace. Kód ve VBA je také možné pořídít pomocí záznamu makra, kdy se zaznamenají kroky, které uživatel provádí a poté je lze spouštět pro jejich automatické vykonání. Kód pořizovaný záznamem makra bývá velmi nepřehledný pro další použití a obsahuje mnoho přebytečných částí (zejména označení každé buňky, na kterou uživatel klikl). Tomuto způsobu tvorby kódu se v této práci věnovat nebudu, u maker napíši přímo kód v editoru VBA.

Nejjednodušším příkladem je vedení účetnictví ve zjednodušeném rozsahu, kde stačí mít základní znalosti práce s Excelem. V případě podvojného účetnictví jsou postupy pomocí Excelu složitější a vyžadují více znalostí pro automatizaci zpracování dat, jinak bude účetnictví nejen velmi náročné, ale také se bude zvyšovat riziko chybovosti. U velkých firem s rozsáhlým účetnictvím je specializovaná aplikace samozřejmě na místě a v dnešní době nezbytná. Zaměřím se na postupy, jak zautomatizovat základy vedení podvojného účetnictví, typicky tedy u podnikatele, který vede účetnictví v plném rozsahu.

1.2 Rozvržení programu

Existuje nepřehledné množství možností, jakým způsobem rozvrhnout a vytvořit účetní program. Zvolil jsem takový postup, kdy program bude rozdělen na logické části na samostatných listech, aby byl přehledný a jeho použití intuitivní. Základ programu tedy budou tvořit tři listy. Jeden s účetní osnovou, která je daná zákonem a obsahuje

¹ MRKOSOVÁ, Jitka. Účetnictví 2013: učebnice pro SŠ a VOŠ, s. 3.

informace o syntetických účtech apod. Druhým bude list, na který se budou pořizované doklady ukládat. Třetí list bude určen pro pořizování dokladů.

1.3 List s účetní osnovou

Základem je účetní osnova, kterou je možné najít ve formátu Excelu bezplatně na internetu, například přímo na internetových stránkách Ministerstva financí ČR. Tuto osnovu je vhodné umístit na samostatný list, který bude v dalších částech této práce pojmenován jako *Osnova*. S největší pravděpodobností bude ve staženém souboru vše v jednom sloupci a každá položka osnovy bude zapsána v jedné buňce ve formátu „číslo účtu – název účtu“. Tyto položky jsou také rozčleněny na bloky s pomocí nadepsaných tříd, skupin a prázdných řádků. Pro další práci s jednotlivými položkami osnovy se vyplatí rozdělit obsah každé buňky na části, konkrétně rozdělit každou z položek na třídu, skupinu, číslo účtu a název účtu, jak je zobrazeno na obrázku č. 1.

	A	B	C	D	E
1	Osnova	Třída	Skupina	Číslo účtu	Název účtu
2	Účtová třída 0 - Dlouhodobý majetek				
3					
4	Účtová skupina 01 - Dlouhodobý nehmotný majetek				
5					
6	012 - Nehmotné výsledky výzkumu a vývoje	0	01	012	Nehmotné výsledky výzkumu a vývoje
7	013 - Software	0	01	013	Software

Obrázek č. 1: Účtová osnova na listu v Excelu

Osnova bude rozdělena do čtyř sloupců. Vzorce se zapíše do druhého řádku a poté budou rozkopírovány až po poslední položku účetní osnovy.

První potřebný údaj je číslo účtu, které lze získat například pomocí tohoto vzorce:

`=KDYŽ(JE.ČÍSLO(HODNOTA(ZLEVA(A2;3)));ZLEVA(A2;3);"")`

Zde je použita textová funkce *ZLEVA()*, která má dva parametry. První je textový řetězec, druhý je počet znaků. Funkce tedy vrátí zadaný počet znaků z textového řetězce z levé strany. V tomto případě se jedná o tři znaky. Jak je vidět na obrázku č. 1, ne všechny řádky je potřeba rozdělit. Některé jsou prázdné, v jiných je nadepsaná účtová skupina. Proto je použita logická podmiňovací funkce *KDYŽ()*. Ta má tři parametry. První je podmínka, která bude vyhodnocena s pravdivostní hodnotou *PRAVDA* nebo

NEPRAVDA, druhý je výsledek, který je vrácen při vyhodnocení podmínky jako *PRAVDA* a třetí je výsledek při vyhodnocení *NEPRAVDA*. Jako podmínka je použito vyhodnocení, zda jsou v položce osnovy první tři znaky čísla (pokud ano, odpovídá to formátu *xxx – název účtu*). První tři znaky jsou získány opět použitím funkce *ZLEVA()*. Tato funkce pracuje s textovým řetězcem a také vrací textový řetězec, takže je pomocí funkce *HODNOTA()* převeden na číslo. Samotné pravdivostní vyhodnocení je poté provedeno informační funkcí *JE.ČÍSLO()*, která ověří, zda je zadaný parametr číslo a podle toho vrátí pravdivostní hodnotu *PRAVDA* nebo *NEPRAVDA*. Pokud je podmínka vyhodnocena jako *PRAVDA*, vrátí funkce *KDYŽ()* tři znaky od začátku textového řetězce, pokud je vyhodnocena jako *NEPRAVDA*, vrátí prázdný textový řetězec (dvojitě uvozovky).

Podobným způsobem je získán název účtu:

=*KDYŽ(JE.ČÍSLO(HODNOTA(ZLEVA(A2;3)));ČÁST(A2;7;255);''')*

Funkce je prakticky stejná jako u získání čísla účtu, rozdíl je pouze u vráceného výrazu v případě vyhodnocení podmínky jako *PRAVDA*. Zde je použita funkce *ČÁST()*, která má jako parametry textový řetězec, dále pozici znaku, od kterého má začít a počet znaků, které vrátí. Začíná sedmým znakem (*xxx – název účtu*) a počet znaků pro vrácení je uveden 255 (v případě, že je znaků méně, bude automaticky vrácen pouze dostupný počet znaků, takže není potřeba používat funkce pro výpočet délky řetězce).

Poslední dvě položky, které lze rozdělením získat, jsou třída a skupina. Pro zjednodušení jsou odvozeny z již rozděleného čísla účtu, opět pomocí funkce *ZLEVA()*. Má to tu výhodu, že není potřeba zajišťovat vyhodnocování podmínek apod., protože v případě, kdy bude ve sloupci s číslem účtu prázdný řetězec (je zadán pomocí dvou uvozovek při nesplnění podmínky na číslo), vrátí funkce *ZLEVA()* opět pouze prázdný řetězec. Je to stejné jako u funkce *ČÁST()* v předchozím kroku, tedy pokud je znaků méně, nedojde k chybě, ale vrací se dostupný počet znaků. Tato vlastnost textových funkcí velmi usnadňuje práci a odstraňuje nutnost ošetřovat případné chybové stavy.

Popis jednotlivých částí listu *Nový doklad*:

Do buněk *C3* a *C4* uživatel zadává rok a měsíc. Při vedení účetnictví se účtuje více dokladů ze stejného měsíce, které se liší pouze dnem v měsíci. Proto je vhodné zadat v prvním kroku rok a měsíc, ve kterém se účtuje a poté zadávat při samotném účtování pouze dny. Den je uveden v buňce *B7*.

Oblast *C7* až *G22* je vymezena pro zadávání položek účetního dokladu. U syntetického účtu je možné použít ověření, které Excel nabízí pro omezení zadaných hodnot, kdy je povolen pouze výběr ze seznamu čísel účtu na listu *Osnova*. Pokud uživatel zadá číslo, které není v seznamu, zobrazí se upozornění. To zabrání použití neexistujícího syntetického účtu.

V buňce *A7* je vzorec pro sestavení data účetního případu, protože datum je zadáváno odděleně. Toto datum bude potřeba až při ukládání dokladu pomocí makra, takže je zformátováno tak, aby nebylo viditelné a nepůsobilo rušivě. Sestaveno je pomocí funkce *DATUM()*, která má jako parametry rok, měsíc a den:

$=\text{DATUM}(C3;C4;B7)$

V buňce *D24* je vypočítán počet zadaných položek účetního dokladu, podle počtu vyplněných buněk položky *Syntetický účet*. Použita je funkce *POČET2()*, která vrátí počet neprázdných buněk ze zadané oblasti:

$=\text{POČET2}(C7:C22)$

Ve spodní části je tlačítko s popiskem „*Uložit záznam*“, které spustí proceduru pro uložení dokladu. Jejímu popisu je věnována další část této práce.

V oblasti *H7* až *H22* je nakopírován vzorec, který podle zadaného čísla syntetického účtu dohledá název tohoto účtu, uživateli tak umožňuje kontrolu správnosti zadaného účtu:

$=\text{KDYŽ}(\text{JE.CHYBHODN}(\text{SVYHLEDAT}(C7;\text{Osnova!D:E;2;0});"";\text{SVYHLEDAT}(C7;\text{Osnova!D:E;2;0})))$

V tomto vzorci je použita vyhledávací funkce *SVYHLEDAT()*, která má čtyři parametry. První parametr určuje, jaký údaj má funkce hledat (v tomto případě je to obsah buňky s číslem syntetického účtu), druhý parametr určuje oblast (tabulku), jejíž první sloupec funkce prohledá, třetí parametr určuje pořadí sloupce z prohledávané oblasti, ze kterého funkce vrátí výsledek (ze stejného řádku, na kterém našla požadovaný výraz v prvním sloupci). Čtvrtý parametr určuje logickou hodnotou, zda je požadována přesná shoda s vyhledávaným výrazem. Protože je požadováno vyhledání přesné shody, je zadána nula (ve všech funkcích, kde je vyžadována logická hodnota *PRAVDA* nebo *NEPRAVDA* je možné použít číslo 1 nebo 0). Pokud funkce *SVYHLEDAT()* nenalezne shodu, vrací chybovou hodnotu. To lze ošetřit pomocí podmiňovací funkce *KDYŽ ()*, kde je v podmínce použita informační funkce *JE.CHYBHODN()*, která vyhodnotí, zda je její parametr chybová hodnota a podle toho vrátí hodnotu *PRAVDA* nebo *NEPRAVDA*. Ve výsledku to znamená, že pokud vrací funkce *SVYHLEDAT()* chybovou hodnotu, zobrazí se prázdný textový řetězec, pokud ne, vrátí výsledek hledání.

1.6 Tvorba makra

Tato část práce bude věnována popisu procedury pro uložení záznamu. Vzhledem k omezením velikosti stránek této práce budou rozděleny některé části kódu na více řádků tak, aby byly funkční takto rozdělené i přímo v aplikaci, tedy mezerou a podtržítkem.

VBA je objektově orientovaný jazyk, ale vzhledem k tomu, že makro bude sloužit k automatizaci po sobě následujících kroků, bude kód psán imperativně. Není tedy potřeba znát postupy objektově orientovaného programování pro vytváření vlastních tříd, objektů apod.

Ve více částech kódu budou využívány objekty *Worksheets* a *Range* pro vymezení oblasti buněk, se kterými se pracuje. U *Worksheets* se v závorce uvádí do uvozovek název listu, u *Range* oblast buněk. U objektu *Range* je často potřeba pracovat s oblastí, která není předem definována. V těchto případech je oblast poskládána pomocí částí, které jsou známy (textový řetězec v uvozovkách) a zbytek je doplněn pomocí proměnných nebo konstant (týká se to čísla řádku). Tyto části jsou potom spojeny jako textový řetězec pomocí operátoru *&*.

Nová procedura je zadána takto:

Sub Ulozeni_zaznamu()

Následně jsou deklarovány tři proměnné celočíselného typu integer:

Dim PocetRadku, PocetPolozek, CisloDokladu As Integer

Do proměnné *PocetRadku* je ukládáno číslo prvního prázdného řádku na listu *Databáze*, tedy řádku, na který budou zapisována nově pořízená data. To je řádek o jednotku větší, než je poslední vyplněný řádek. Vypočítán je takto:

*PocetRadku = Worksheets("Databáze").Range("A1").CurrentRegion.Rows. _
Count + 1*

Do proměnné *PocetPolozek* je uložen počet položek na nově pořizovaném dokladu. Tento údaj je uveden přímo v buňce na listu *Nový doklad*, takže do proměnné stačí pouze uložit obsah této buňky pomocí vlastnosti *Value*:

PocetPolozek = Worksheets("Nový doklad").Range("D24").Value

Proměnná *CisloDokladu* bude sloužit k automatickému očíslování nového dokladu, takže je do ní ukládáno číslo o jednotku větší, než je nejvyšší číslo dokladu na listu *Databáze*:

*CisloDokladu = WorksheetFunction.Max(Worksheets("Databáze"). _
Range("A2:A" & PocetRadku - 1)) + 1*

Dále je přidán rozhodovací blok *If*, ve kterém je vyhodnoceno, zda je proměnná *PocetPolozek* rovna nule. Pokud ano, znamená to, že na dokladu nebyla zadána žádná položka a doklad z tohoto důvodu nebude uložen. Uživateli se následně zobrazí zpráva „Nebyl zadán žádný doklad“ a procedura bude ukončena:

```

If PocetPolozek = 0 Then
MsgBox ("Nebyl zadán žádný doklad")
Exit Sub
End If

```

Podobnými rozhodovacími bloky lze ošetřit také případy, kdy se součet Má dáti nerovná součtu Dal apod.

Pokud byla předchozí podmínka vyhodnocena jako nepravda, kód pokračuje dále. V dalším kroku je vhodné zrušit obnovování obrazovky při běhu programu. Není použit automatický záznam makra, který by zvýšil množství označovaných buněk apod., ale přesto se často v kódu označování nelze vyhnout, což by mohlo způsobit (pro uživatele nepříjemné) blikání obrazovky. K tomu dochází tím, že makro běží velmi rychle, při jeho běhu se provádí zápis dat, přepínání listů nebo označování buněk. Pokud je deaktivováno obnovování obrazovky, k problikávání nedojde. Také díky tomu teoreticky dojde ke zrychlení procedury, ale to je při její jednoduchosti a výkonu běžných kancelářských počítačů zanedbatelné. Obnovování obrazovky lze vypnout nastavením vlastnosti *ScreenUpdating* objektu *Application* na logickou hodnotu *False*.²

```

Application.ScreenUpdating = False

```

Nyní je potřeba zadaný doklad uložit na list Databáze. Z velké části jsou sloupce seřazeny ve stejném pořadí, ale u některých položek tomu tak není z důvodu přehlednosti při zadávání, například u automaticky vyplňovaných polí atd.

Jako první je zkopírováno datum účetního dokladu. Vypočítáno je v buňce A7, takže tuto buňku je potřeba překopírovat jako hodnotu na list *Databáze*. Vložena je na všechny řádky ukládaného záznamu, čili začíná se číslem řádku, uloženým v proměnné *PocetRadku* až po řádek o jednotku menší, než je součet proměnných *PocetRadku* a *PocetPolozek*:

```

Worksheets("Nový doklad").Range("A7").Copy
Worksheets("Databáze").Range("B" & PocetRadku & ":B" & PocetRadku _
+ PocetPolozek - 1).PasteSpecial Paste:=xlPasteValues

```

² KRÁL, Martin. Excel VBA: výukový kurz, s. 159.

Dále jsou překopírována data, která byla na doklad zadána ručně, tedy od buňky C po buňku G. Od sedmého řádku do řádku podle počtu položek a vložení jako hodnoty na list *Databáze*:

```
Worksheets("Nový doklad").Range("C7:G" & 6 + PocetPolozek).Copy  
Worksheets("Databáze").Range("C" & PocetRadku).PasteSpecial _  
Paste:=xlPasteValues
```

V mnoha případech je užitečné vědět, kdy byla nějaká položka uložena, takže je na list *Databáze* do sloupce *TimeStamp* zapsáno aktuální datum a čas do řádků, na které se aktuálně vkládají záznamy:

```
Worksheets("Databáze").Range("H" & PocetRadku & ":H" & PocetRadku _  
+ PocetPolozek - 1) = DateTime.Now
```

Poslední záznam, který je zapsán do aktuálně zpracovávaných řádků je číslo dokladu. To bylo na začátku kódu uloženo do proměnné *CisloDokladu*:

```
Worksheets("Databáze").Range("A" & PocetRadku & ":A" & PocetRadku _  
+ PocetPolozek - 1) = CisloDokladu
```

Nyní je možné vymazat obsah oblasti na listu *Nový doklad*, do které byly zadávány účetní položky:

```
Worksheets("Nový doklad").Range("B7:G22").ClearContents
```

Nakonec je vhodné zobrazit zprávu o úspěšném uložení dokladu a v ní i číslo dokladu, pod kterým byl uložen. Následně je možné proceduru ukončit:

```
MsgBox ("Doklad byl uložen pod číslem " & CisloDokladu)  
End Sub
```

1.7 Další kroky pro tvorbu programu

Nyní je program ve stavu, kdy je možné pořizovat a ukládat doklady do databáze. Jako výstupy budou sloužit účetní sestavy, jako je například rozvaha, výkaz zisku a ztrát nebo stavy jednotlivých účtů. Stejně jako osnovu, lze i vzor pro rozvahu stáhnout například z internetových stránek Ministerstva financí ČR. Ke zpracování těchto výstupů bude potřeba matematická funkce *SUMIF()*, sčítající všechny položky ve zvolené oblasti, které splňují zadané kritérium. U těchto výstupů bude kritériem číslo syntetického účtu, takže funkce u každého účtu sečte všechny odpovídající položky z databáze. Tyto výstupy je možné umístit do formulářů na samostatných listech. Většina dalších postupů by vyžadovala opětovné používání funkcí, které již byly uvedeny, takže se dále účetnímu programu v této práci věnovat nebudu. Stěžejním bodem této části byla ukázka použití funkcí a postupů.

2 Kontingenční tabulka

2.1 Výhody použití kontingenční tabulky

Domnívám se, že kontingenční tabulka je „nejsilnějším“ nástrojem, který Excel nabízí. S její pomocí lze velmi rychle analyzovat data díky seskupování, získávání součtů, průměrů a jiných agregací. Kontingenční tabulka umožňuje vytvořit křížový náhled na data, díky kterému je možné získat průniky různých datových oblastí a také automaticky data uspořádat a seskupit podle potřeby. Navíc lze pohledy velmi rychle měnit pouze pomocí jednoduchého přemístění různých položek mezi řádky, sloupci apod.

2.2 Zdrojová data pro kontingenční tabulku

Vyjma dat na listu mohou být zdrojem dat pro kontingenční tabulku také napojené externí databáze nebo textové soubory. Tyto externí zdroje lze načíst přímo na list nebo mohou být pouze propojeny, přičemž omezením je pouze paměť počítače.³ V této práci budou použita pouze data na listu Excelu. Je nezbytné, aby byla data sestavena ve formě seznamu bez prázdných sloupců a řádků. První řádek dat musí obsahovat hlavičku, v následujících řádcích jsou jednotlivé záznamy. Přitom v jednom sloupci mohou být

³ BROŽ, Milan. Microsoft Office Excel 2007: podrobná uživatelská příručka, s. 279.

data pouze jednoho typu, jako např. čísla, text atd. Nelze tedy jako zdrojová data pro kontingenční tabulku použít data ve formě křížového zobrazení apod.

2.3 Příklad rozvržení a použití kontingenční tabulky

Kontingenční tabulka nabízí velkou variabilitu pohledů na data. Pro příklad mějme databázi na listu Excelu, která obsahuje údaje o klientské spokojenosti na základě zpětné vazby. Náhodně vybraní klienti jsou dotazováni a hodnotí spokojenost s řešením daného požadavku příslušným oddělením firmy, které daný požadavek vyřizovalo. Klienti hodnotí spokojenost s řešením ve škále od 1 (nejlepší) do 5 (nejhorší). V datech jsou následující údaje:

- Datum hodnocení
- ID zákazníka
- Hodnocené oddělení
- Obdržená známka od zákazníka

Díky vytvoření kontingenční tabulky z těchto několika údajů lze analyzovat spokojenost zákazníků s jednotlivými odděleními v čase, srovnávat je apod. Pro další práci je vhodné v datech ještě připravit pomocný sloupec, ve kterém je pomocí funkce *MĚSÍC()* získán z data hodnocení pouze měsíc. To pomůže v další analýze, protože výstupy jsou často požadovány v měsíční periodě. Tato funkce má datum jako jediný vstupní parametr a vrací pořadové číslo kalendářního měsíce.

`=MĚSÍC(A2)`

V datech je tedy po dopočítání měsíce šest sloupců ve formě seznamu, jak je uvedeno na obrázku č. 4.

	A	B	C	D	E
1	Datum	ID zákazníka	Oddělení	Hodnocení	Měsíc
2	5.1.2013	1	Reklamace	3	1
3	7.1.2013	2	Prodej	1	1
4	7.1.2013	3	Prodej	1	1
5	7.1.2013	4	Podpora	2	1
6	11.1.2013	5	Reklamace	2	1
7	11.1.2013	6	Reklamace	1	1
8	12.1.2013	7	Prodej	2	1
9	13.1.2013	8	Podpora	3	1
10	13.1.2013	9	Podpora	1	1
11	13.1.2013	10	Reklamace	1	1
12	15.1.2013	11	Podpora	2	1
13	16.1.2013	12	Prodej	3	1
14	17.1.2013	13	Prodej	3	1
15	17.1.2013	14	Prodej	1	1
16	17.1.2013	15	Reklamace	1	1

Obrázek č. 4: Zdrojová data klientské spokojenosti pro kontingenční tabulku

Aktuálně je v databázi 222 záznamů a jejich počet bude výrazně růst, proto by mohlo být vyhodnocování bez použití kontingenční tabulky zdlouhavé. Pokud je vytvořena kontingenční tabulka, lze ihned zobrazit požadované výstupy. Například je možné ji rozvrhnout tak, že do popisků sloupců bude vložen *Měsíc*, do řádků *Oddělení* a do oblasti hodnot *Hodnocení*. Kontingenční tabulka automaticky seskupí *Měsíc* i *Oddělení*, takže každý z těchto údajů je uveden pouze jednou.

Protože *Hodnocení* je číselné pole, automaticky se přednastaví součtová agregace. To je nežádoucí, součet hodnocení ve většině případů nic nevypovídá, takže je potřeba změnit nastavení tohoto pole na průměr. Výsledkem je průměrné hodnocení jednotlivých oddělení chronologicky po měsících, jak je uvedeno na obrázku č. 5.

	A	B	C	D	E	F	G	H	
2									
3	Průměr z Hodnocení	Popisky sloupců							
4	Popisky řádků		1	2	3	4	5	6	Celkový součet
5	Podpora		1,8	1,5	2,1	1,8	1,8	1,0	1,7
6	Prodej		1,9	1,8	1,8	1,9	1,8	1,5	1,8
7	Reklamace		2,1	2,4	1,7	2,2	1,7	2,3	2,1
8	Celkový součet		2,0	1,9	1,9	2,0	1,8	1,7	1,8

Obrázek č. 5: Kontingenční tabulka zobrazující průměrnou spokojenost klientů

Jednoduchými změnami lze získávat další údaje. Pokud se změní nastavení oblasti hodnot na *Počet*, je získána informace o tom, kolikrát byla jednotlivá oddělení každý měsíc hodnocena. Pro tento údaj nemusí být v oblasti hodnot uvedeno *Hodnocení*, může tam být prakticky kterékoliv pole z databáze. Pokud by bylo do oblasti hodnot například vloženo pole *Oddělení*, získán je stejný údaj. Výsledek je stejný bez ohledu na to, zda je zobrazeno, kolikrát se v datech vyskytuje nějaké oddělení, měsíc nebo hodnocení apod. Pole *Oddělení* obsahuje jako data textové řetězce, u kterých kontingenční tabulka automaticky nastavuje agregaci *Počet*. Ve výsledku tedy pouze změnou oblasti hodnot získáme informaci o tom, kolikrát byla jednotlivá oddělení hodnocena po měsících, celkem nebo také kolik hodnocení celkem proběhlo v daném měsíci, jak je uvedeno na obrázku č. 6.

	A	B	C	D	E	F	G	H
2								
3	Počet z Hodnocení	Popisky sloupců						
4	Popisky řádků		1	2	3	4	5	6 Celkový součet
5	Podpora		5	8	11	8	18	11
6	Prodej		8	12	17	14	25	15
7	Reklamace		7	7	11	10	17	18
8	Celkový součet		20	27	39	32	60	44
9								

Obrázek č. 6: Kontingenční tabulka zobrazující počet hodnocení

2.4 Příklad s použitím počítaného pole

Na dalším příkladu budou ukázány možnosti použití výpočtových polí. Kontingenční tabulka nabízí možnost vlastních výpočtů pro případy, kdy nestačí agregace pro jedno pole, kde se pouze vybírá, jak je požadováno oblast hodnot nastavit.

Pro příklad tedy mějme situaci, kdy je na listu Excelu statistika prohovořené doby a počtu hovorů na jednotlivých odděleních po měsících. Požadovaným výstupem je průměrná délka hovoru, kterou lze jednoduše zjistit s pomocí kontingenční tabulky. Podoba dat je uvedena na obrázku č. 7.

	A	B	C	D
1	Timekey	Oddělení	Prohovořená doba [s]	Počet hovorů
2	201301	Transakce	38 964	452
3	201301	Podpora	29 550	380
4	201302	Transakce	45 679	512
5	201302	Podpora	31 101	369
6	201303	Transakce	44 256	498
7	201303	Podpora	30 124	311
8	201304	Transakce	46 845	537
9	201304	Podpora	28 956	280
10	201305	Transakce	51 980	569
11	201305	Podpora	32 850	297
12	201306	Transakce	42 168	471
13	201306	Podpora	30 472	324

Obrázek č. 7: Zdrojová data statistik hovorů pro kontingenční tabulku

Z dat je vytvořena kontingenční tabulka a do popisků řádků vložen měsíc (tedy pole *Timekey*), do sloupců Oddělení. Do oblasti hodnot nebude vloženo nic, zvolena je možnost *Počítané pole*, kde lze vytvořit položku *Průměrná délka hovoru* a zadat do ní vzorec (názvy polí se nemusí vypisovat, stačí je vybrat v nabídce a přidají se do vzorce).

$$= 'Prohovořená doba [s]' / 'Počet hovorů'$$

Ve výsledku bude v kontingenční tabulce zobrazena průměrná délka hovoru po měsících za jednotlivá oddělení i celkové součty, jak je uvedeno na obrázku č. 8.

	A	B	C	D
1				
2				
3	Součet z Průměrná délka hovoru		Popisky sloupců ▾	
4	Popisky řádků ▾		Podpora	Transakce Celkový součet
5	201301		78	86 82
6	201302		84	89 87
7	201303		97	89 92
8	201304		103	87 93
9	201305		111	91 98
10	201306		94	90 91
11	Celkový součet		93	89 91
12				

Obrázek č. 8: Kontingenční tabulka zobrazující průměrnou délku hovoru

2.5 Možnosti nastavení kontingenční tabulky

Kontingenční tabulka nabízí velké množství možností nastavení. Budu se věnovat popisu těch, které jsou dle mého názoru nejdůležitější.

Pro chybové hodnoty zobrazit – v praxi tuto možnost používám nejčastěji. Vzhledem k tomu, že kontingenční tabulka pracuje s křížovým zobrazením a ukazuje tedy průniky, často se stává, že v nějakém průniku žádná data nejsou. To může způsobit problém, pokud se pracuje s Počítaným polem, kdy se může v těchto průnicích zobrazovat například chyba dělení nulou (Excel ji zobrazí jako #DĚLENÍ_NULO!). To působí velmi rušivě a při větším počtu výskytů je celá kontingenční tabulka velmi nepřehledná. Proto lze touto volbou nastavit, co se má v těchto případech zobrazovat (například 0).

Pro prázdné buňky zobrazit – tato volba se nastavuje v případech, kdy v průnicích žádná hodnota není (ale nedochází k žádné chybě při výpočtu apod.). Lze nastavit například nulu pro jednotnou podobu tabulky, aby v ní nebyla prázdná místa nebo například zvýraznění, že se jedná o prázdný průnik v situacích, kdy jsou právě tyto položky důležité.

Celkové součty – tyto volby jsou automaticky přednastaveny pro řádky i sloupce, lze je vypnout v případech, kdy celkové součty nemají význam a je potřeba tabulku zmenšit a zpřehlednit.

Aktualizovat data při otevření souboru – tato položka je standardně vypnuta, je vhodné ji zvolit v případech, kdy je kontingenční tabulka vytvořena dlouhodobě a pracuje se s ní často, aby bylo zajištěné obnovení při každém otevření souboru.

3 Předpovědi budoucích stavů

U mnoha ekonomických problémů je často potřeba odhad budoucího vývoje nějaké veličiny. Může se jednat o finance, počty prodejů, zakázek apod. V Excelu je k dispozici hned několik předpovědních funkcí, které dokáží poskytnout relevantní výsledky v případě, že je k dispozici více než pět vstupních bodů.

3.1 FORECAST

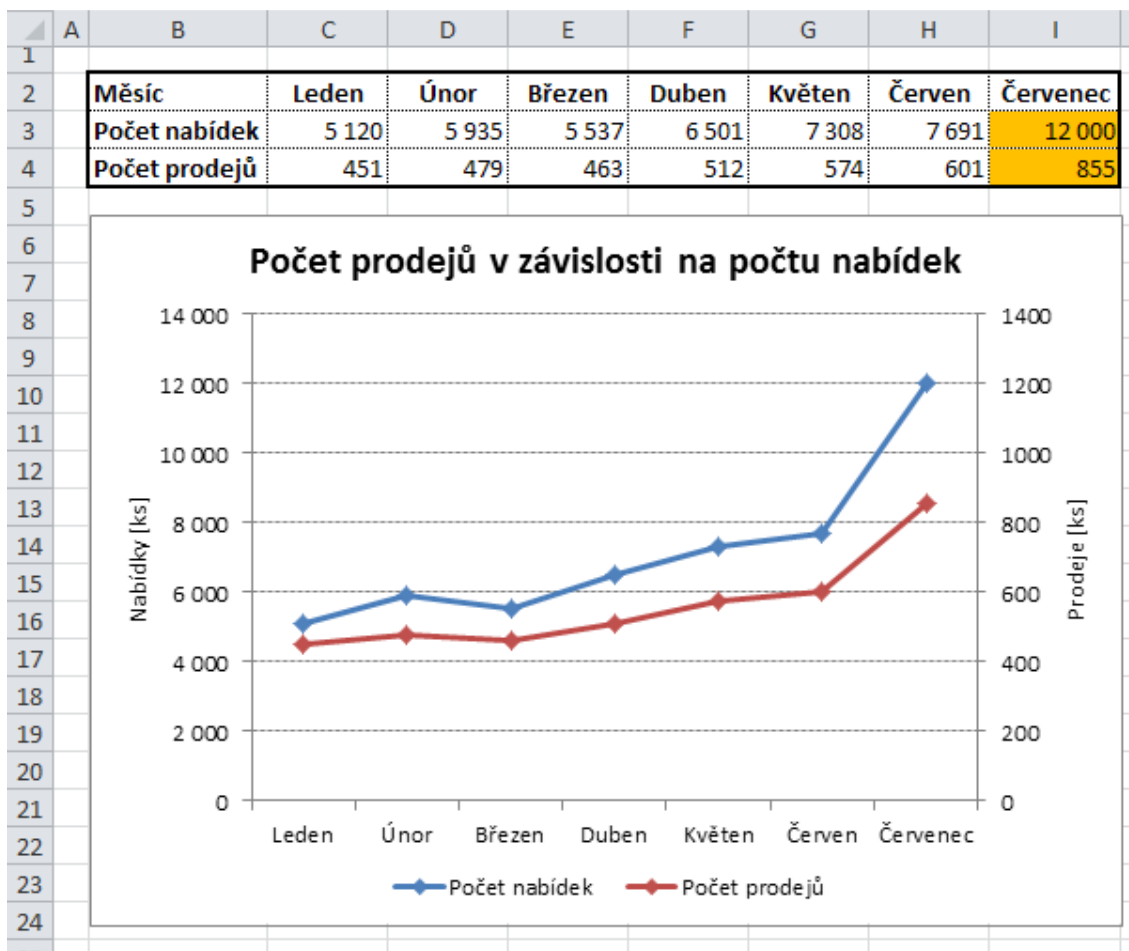
Tato funkce pomocí lineární interpolace odhadne hodnotu y , pro známou hodnotu x , na základě předchozího vývoje obou datových řad x a y .⁴

Pro příklad uvedu situaci v call centru, které vyřizuje příchozí hovory na informační lince a v rámci hovorů také nabízí klientům produkty. Jedná se o servisní část call centra, která vyřizuje převážně příchozí hovory a neprovozuje telemarketingové kampaně apod. Ve velké míře jsou tedy nabízeny produkty pouze v případě, že o ně klient sám přímo projeví zájem nebo například v situacích, kdy operátor vyhodnotí možnost zájmu klienta na základě specifikací jeho požadavku. Vedení call centra ovšem začíná vyvíjet tlak na prodeje a požaduje po operátorech aktivní nabízení produktů ve větší míře, tedy i v případech, kdy z klientova požadavku není přímo zřejmé, že by měl o daný produkt zájem. S rostoucím počtem nabídek, které nejsou přímo cílené na téma hovoru, klesá úspěšnost prodeje a zároveň se prodlužuje délka hovorů, takže není zcela jasné, zda se call centru vyplatí větší tlak na nabízení produktů. Každá nabídka je operátorem zaznamenána, takže vedení má k dispozici přehled vývoje počtu nabídek a prodejů a chce zjistit, kolik prodejů by se uskutečnilo při 12 000 nabídkách měsíčně.

Předchozí vývoj, výsledná hodnota a grafické znázornění je na obrázku č. 9. Ve funkci *FORECAST()*, zadané v buňce *I4*, je v prvním parametru použita hodnota 12 000 jako nová hodnota x , ve druhém parametru je vývoj počtu prodejů jako pole y , ve třetím parametru vývoj počtu nabídek jako pole x .

$=FORECAST(I3;C4:H4;C3:H3)$

⁴ BROŽ, Milan a BEZVODA, Václav. Microsoft Excel 2007 pro manažery a ekonomy, s. 155.



Obrázek č. 9: Odhad vývoje počtu prodejů v závislosti na počtu nabídek

3.2 LINTREND

Stejně jako funkce *FORECAST()*, pracuje na základě lineární regrese. U předchozího příkladu uvedeného u funkce *FORECAST()* by byl výsledek za použití funkce *LINTREND()* stejný. Rozdíl mezi těmito funkcemi je ten, že funkce *LINTREND()* může pracovat s větším počtem nezávislých proměnných.⁵

Příkladem může být situace, kdy obec sestavuje rozpočet a musí v něm uvést příjmy v podobě odměny za zpětný odběr tříděného odpadu. Je zapotřebí tedy zjistit, jaké množství odpadu bude v kontejnerech vyvezeno se zohledněním plánovaného rozšíření počtu kontejnerů a kontejnerových stání oproti předchozím obdobím. Výsledek a grafické znázornění je uvedeno na obrázku č. 10. Ve funkci *LINTREND()*, zadané v buňce C8, je v prvním parametru vývoj počtu svezných tun odpadu jako pole

⁵ BROŽ, Milan a BEZVODA, Václav. Microsoft Excel 2007 pro manažery a ekonomy, s. 157.

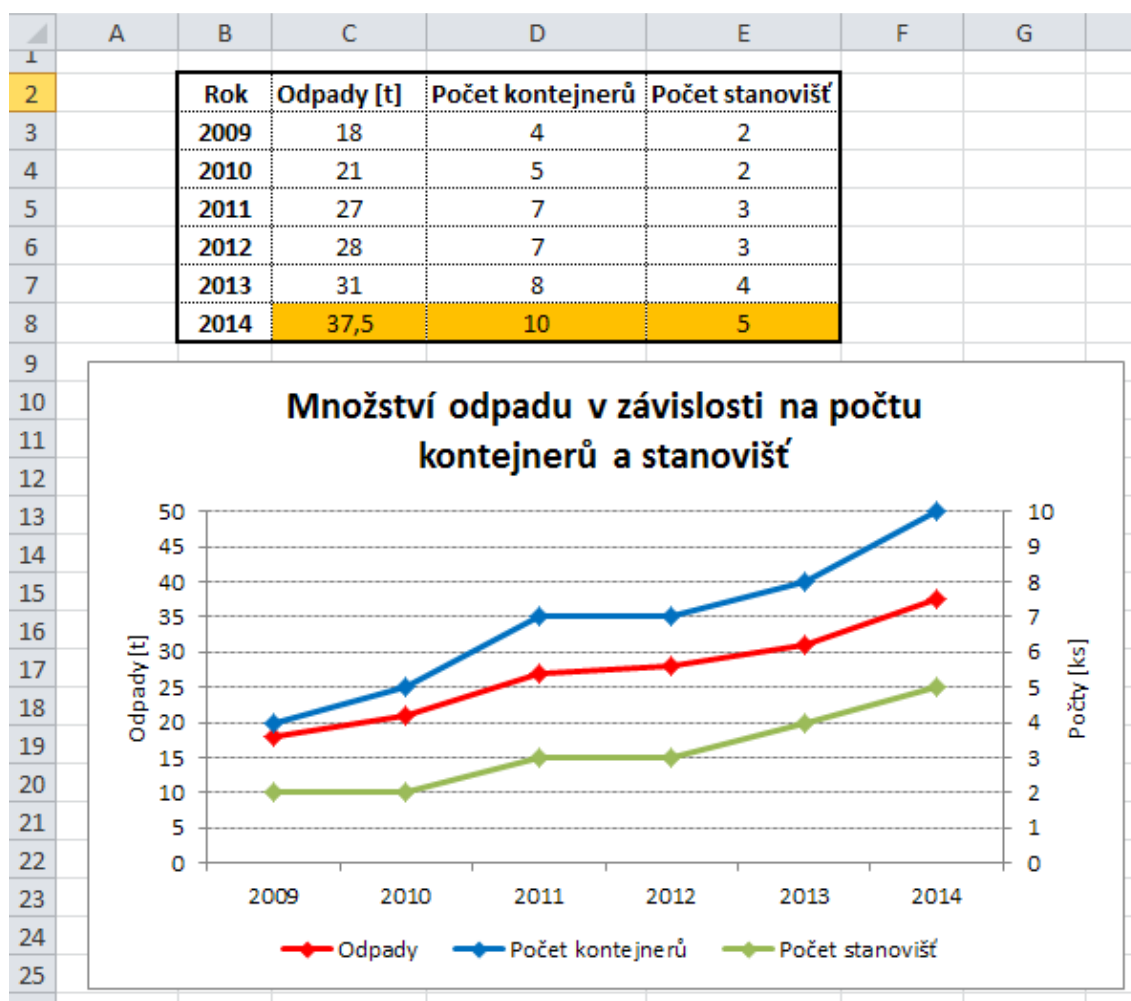
y, ve druhém parametru vývoje počtu kontejnerů a stanovišť jako pole x, a ve třetím parametru jsou nové hodnoty počtu kontejnerů a stanovišť jako nové hodnoty x.

U funkce *LINTREND()* je možné zadat ještě čtvrtý (nepovinný) parametr, kterým je logická hodnota b. Tento parametr určuje, zda se bude počítat s konstantou v následujícím vzorci používaném funkcí *LINTREND()*:

$$y = mx + b$$

Pokud má čtvrtý parametr logickou hodnotu *PRAVDA* nebo je vynechán, s konstantou se ve vzorci počítá, pokud má hodnotu *NEPRAVDA*, je konstanta rovna nule. Výsledný výpočet s použitím funkce *LINTREND()* je následující:

$$=LINTREND(C3:C7;D3:E7;D8:E8)$$



Obrázek č. 10: Odhad počtu svezeného odpadu

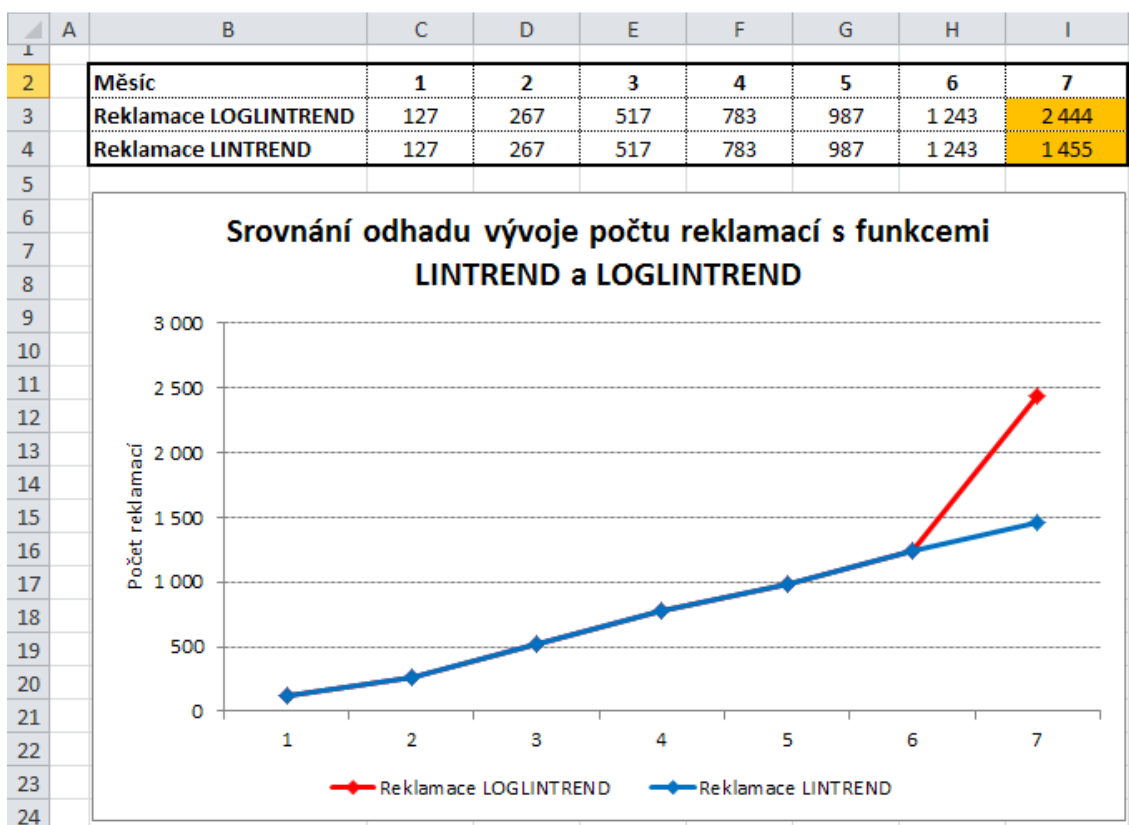
3.3 LOGLINTREND

Používá se podobně jako funkce *LINTREND()*, ale oproti ní pracuje s předpokladem exponenciálního růstu. Je tedy zapotřebí předem promyslet, kterou funkci použít. Funkce *LOGLINTREND()* se používá v případech zavádění nových výrobků na trh, spouštění marketingových kampaní s cílem výrazného zvýšení prodeje apod. Tedy v případě, kdy je očekáván exponenciální nárůst hodnot.⁶

Uvedu příklad odhadu počtu reklamací výrobku při snižování nákladů na výrobu použitím levnějších dílů, na základě předchozího vývoje. Pro srovnání uvedu i variantu s použitím funkce *LINTREND()*.

$=\text{LOGLINTREND}(C3:H3;C2:H2;I2)$

$=\text{LINTREND}(C3:H3;C2:H2;I2)$



Obrázek č. 11: Srovnání odhadů s funkcemi Lintrend a Loglintrend

⁶ BROŽ, Milan a BEZVODA, Václav. Microsoft Excel 2007 pro manažery a ekonomy, s. 161.

Jak je z obrázku č. 11 zřejmé, použití funkcí *LINTREND()* a *LOGLINTREND()* vede při stejném předchozím vývoji ke zcela odlišným výsledkům. Je tedy důležité si před jejich využitím uvědomit jakým způsobem odhad vypočítají na základě znalostí dané problematiky a předpokladu růstu.

4 Funkce pro výpočty úvěrů a investic

V Excelu lze nalézt řadu funkcí, které mohou usnadnit výpočty související s úvěry a investicemi. Mohou velmi usnadnit práci, protože umožňují poměrně jednoduchým způsobem získat požadované informace ohledně splátek, úrokových sazeb, výších úvěrů a investic apod. Tato část práce bude tedy zaměřena na ty funkce, které jsou dle mého názoru v praxi velmi efektivně využitelné. Některé funkce jsou si velmi podobné s tím, že se pouze mění parametr, který známe za jiný, který chceme vypočítat (například výše splátky, úroková sazba, počet splátek apod.). Proto u některých nebudu uvádět podrobnější popis a příklady výpočtů, ale pouze zmíním jejich využití.

4.1 BUDHODNOTA

Jak z názvu funkce vyplývá, počítá se s ní budoucí hodnota vkladu (investice) na základě současné hodnoty, úroku a vkladů.⁷ Druhou možností využití je počítání nesplacené části úvěru, což je situace, která se v literatuře většinou pouze zmiňuje, ale neuvádí se příklad. Dle mého názoru je i toto využití v praxi užitečné, protože můžeme chtít zjistit, jakou částku bychom za určitou dobu potřebovali k doplacení úvěru apod. Proto se zaměřím na oba způsoby využití funkce.

U funkce *BUDHODNOTA()* se jako první parametr zadává úroková sazba, která se případně dělí periodou splácení. Do druhého parametru se zadává počet period, do třetího výše splátky (vkladu), do čtvrtého současná hodnota. Pátý (nepovinný) parametr je logická hodnota, která určuje, zda jsou platby uhrazeny na začátku periody (1) nebo na konci (0 nebo vynechání parametru).

Jako první příklad uvedu investici v podobě pravidelného spoření s úročením zůstatku. Například při počátečním vkladu 100 000 Kč, na konci každého měsíce bude vklad 20 000 Kč, délka spoření čtyři roky a úrok 3,7%. Použitý vzorec bude následující:

$=BUDHODNOTA(3,7\%/12;48;-20000;-100000)$

⁷ BROŽ, Milan a BEZVODA, Václav. Microsoft Excel 2007 pro manažery a ekonomy, s. 180.

Výsledná částka bude 1 148 890,77 Kč. Jelikož se jedná o investici s měsíční periodou, roční úroková sazba je dělena dvanácti a délka spoření je také uvedena v měsících. Dále je důležité zadávat správné znaménko u částek, v tomto případě u počátečního i měsíčního vkladu jako záporné, protože se jedná o finance, které jsou do spoření vkládány. Jedná se o platbu na konci měsíce, takže poslední parametr není nutné uvádět.

Jiným příkladem může být již zmíněný výpočet nesplacené části úvěru. Mějme tedy úvěr ve výši 370 000 Kč, měsíční splátku 5000 Kč a úrokovou sazbu ve výši 5,7%. Chceme vypočítat, jaká bude nesplacená část úvěru po čtyřech letech splácení. Hlavní rozdíl v zadávání funkce je ten, že současná hodnota (tedy poskytnutá výše úvěru) se zadává jako kladná, protože se jedná o prostředky, které jsme obdrželi. Úvěr je splácen měsíčně, takže úroková sazba je opět dělena dvanácti a stejně tak počet period je zadán v měsících. Výsledná částka je -195 650,79 Kč a funkce je zadána takto:

$$=BUDHODNOTA(5,7\%/12;48;-5000;370000)$$

4.2 SOUČHODNOTA

Vrací současnou hodnotu investice (úvěru). Tato funkce se používá v případech, kdy je dána částka, kterou lze splácet úvěr po určitou dobu a známa je také úroková sazba.⁸ Lze tedy s její pomocí zjistit možnou výši poskytnutého úvěru. Parametry jsou podobné jako u předchozí funkce, tedy první je roční úroková sazba dělená periodou, druhý počet period, třetí výše splátky. Čtvrtý (nepovinný) parametr je budoucí hodnota, tedy požadovaná částka po poslední splátce. Pokud není uveden, počítá se jako nula. Pátý (také nepovinný) parametr je logická hodnota určující termín splátky, kdy je stejně jako u funkce *BUDHODNOTA()* výchozí nastavení nula, tedy na konci periody.

Například jsme v situaci, kdy potřebujeme maximální možnou výši úvěru. Můžeme splácet 7 000 Kč měsíčně po dobu šesti let, poskytnutá úroková sazba by byla 6,1%. Výsledkem bude částka 421 175,93 Kč a funkce bude vypadat takto:

$$=SOUČHODNOTA(6,1\%/12;72;-7000)$$

⁸ BROŽ, Milan a BEZVODA, Václav. Microsoft Excel 2007 pro manažery a ekonomy, s. 181.

Pokud bychom v této funkci uvedli kladnou splátku namísto záporné, výsledkem by byla stejná částka, ale záporná. Jednalo by se o pohled z pozice investora, takže současná hodnota by byla záporná, protože by se jednalo o prostředky, které poskytuje a splátka by byla kladná, protože by to byly jeho příjmy.

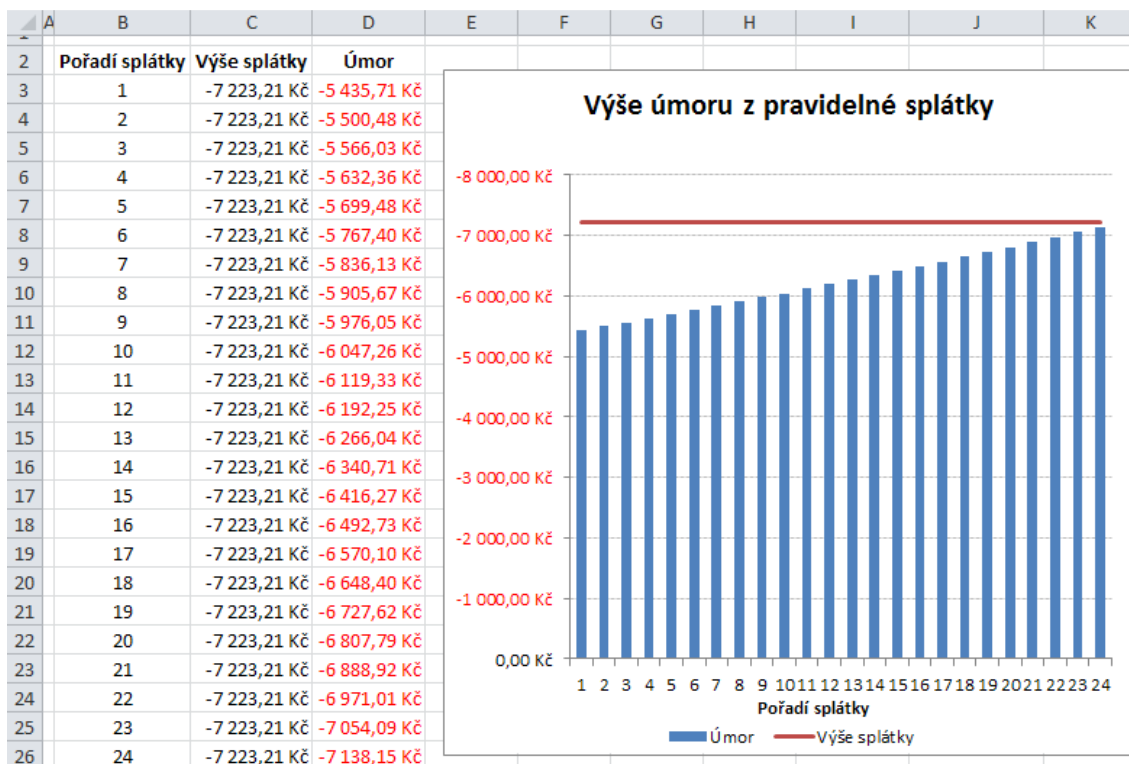
4.3 PLATBA.ZÁKLAD a PLATBA.ÚROK

Jedná se dle mého názoru o velmi zajímavé funkce, pomocí kterých lze poměrně jednoduchým způsobem zjistit z každé splátky výši úmoru, tedy částky, která snižuje jistinu úvěru nebo úroku. S každou další splátkou se zvyšuje podíl úmoru a snižuje podíl úroku. Funkce se zadávají stejně, rozdíl je pouze v tom, že jedna vrací úmorovou část platby, druhá úrok.⁹ Prvním parametrem je opět úroková sazba, druhým je pořadové číslo platby, pro kterou chceme úmor vypočítat, třetím parametrem je počet splátek, čtvrtým současná hodnota. Další dva parametry jsou nepovinné, jedná se o budoucí hodnotu a termín platby stejně jako v předchozích funkcích.

Příkladem může být úvěr ve výši 150 000 Kč, který bude splácen dva roky. Roční úroková sazba činí 14,3%, a výše splátky je 7 223,21 Kč. Chceme zjistit výši úmoru ze všech splátek po celou dobu splácení. V tomto případě je výhodné zadat druhý parametr jako odkaz na buňku, ve které je pořadové číslo platby uvedeno (mějme sloupec s čísly plateb). Poté stačí vzorec jen zkopírovat na požadovanou oblast a získáme výsledné hodnoty pro každou platbu. Výsledek a grafické znázornění rostoucího úmoru je uveden na obrázku č. 12. Použitá funkce v buňce D3 bude následující:

=PLATBA.ZÁKLAD(14,3%/12;B3;24;150000)

⁹ BROŽ, Milan a BEZVODA, Václav. Microsoft Excel 2007 pro manažery a ekonomy, s. 185.



Obrázek č. 12: Výpočet úmoru z pravidelné splátky

4.4 PLATBA

Funkce pro výpočet výše splátky úvěru nebo platby u investice.¹⁰ Parametry jsou opět podobné jako u předchozích funkcí, jedná se o úrokovou sazbu, počet splátek, současnou hodnotu a dva nepovinné parametry v podobě budoucí hodnoty a termínu platby.

4.5 POČET.OBDOBÍ

Tato funkce vrací počet splátek (období) úvěru nebo vkladů do investice při parametrech při parametrech úroková sazba, výše splátky, současná hodnota, budoucí hodnota a logický typ termínu platby.¹¹

4.6 ÚROKOVÁ MÍRA

Funkce pro výpočet úrokové sazby, potřebné pro dosažení požadované hodnoty, při parametrech počet splátek, výše splátky současná hodnota, budoucí hodnota a logický typ termínu platby.¹²

¹⁰ BROŽ, Milan a BEZVODA, Václav. Microsoft Excel 2007 pro manažery a ekonomy, s. 183.

¹¹ Tamtéž, s. 182.

¹² Tamtéž, s. 186.

5 Funkce pro Cash Flow

Cash flow, neboli peněžní tok, je velmi důležitá součást finanční analýzy. Jeho správné sestavení dokáže manažerovi nebo ekonomovi ukázat reálný pohled na pohyby financí. Sestavuje se na úrovni provozní, finanční a investiční.¹³ Excel nabízí několik funkcí, které pomáhají při analýze cash flow a jsou v praxi velmi přínosné a poměrně snadno použitelné.

5.1 ČISTÁ.SOUČHODNOTA

Tato funkce vrací čistou současnou hodnotu, neboli součet všech pravidelných peněžních toků projektu (investice) diskontovaných úrokovou sazbou. Pomocí této funkce lze srovnávat výnosnost různých projektů (investic) a zvolit ten výhodnější.¹⁴

Prvním parametrem je úroková sazba, dalších až 254 parametrů jsou konkrétní peněžní toky (každý z nich může být pole).

Příkladem může být srovnání dvou investic v podobě nákupu nového výrobního zařízení. U obou projektů počítáme se stejnou úrokovou sazbou 10% a stejným součtem celkových příjmů v následujících letech, ale liší se rozložením příjmů v jednotlivých letech a také ve výši počáteční investice. Výpočet je uveden na obrázku č. 13, ze kterého je zřejmé, že i přes vyšší počáteční investici a při stejném součtu příjmů je výhodnější první projekt, jelikož má výhodnější rozložení finančních toků v jednotlivých letech. Funkce použitá pro výpočet prvního projektu je následující:

$$=ČISTÁ.SOUČHODNOTA(10%;C4:C8)+C3$$

Funkce pro druhý projekt:

$$=ČISTÁ.SOUČHODNOTA(10%;D4:D8)+D3$$

¹³ ČECHOVÁ, Alena. Manažerské účetnictví, s. 20.

¹⁴ BROŽ, Milan a BEZVODA, Václav. Microsoft Excel 2007 pro manažery a ekonomy, s. 197.

	A	B	C	D
1				
2			1. projekt	2. projekt
3		Poč investice	-200 000 Kč	-190 000 Kč
4		Příjmy 1. rok	130 000 Kč	80 000 Kč
5		Příjmy 2. rok	90 000 Kč	80 000 Kč
6		Příjmy 3. rok	70 000 Kč	80 000 Kč
7		Příjmy 4. rok	70 000 Kč	80 000 Kč
8		Příjmy 5. rok	40 000 Kč	80 000 Kč
9		Příjmy celkem	400 000 Kč	400 000 Kč
10		Čistá souč. hodnota	117 802 Kč	113 263 Kč
11				

Obrázek č. 13: Výpočet čisté současné hodnoty

Je důležité si uvědomit, že v těchto případech je počáteční investice odečítána zvlášť a není součástí funkce *ČISTÁ.SOUČHODNOTA()*, protože je provedena na začátku projektu a tedy není nijak diskontována, jako by tomu bylo v případě, kdy by byla provedena až na konci roku. V tom případě by byla ve funkci uvedena jako první peněžní tok.

5.2 MÍRA.VÝNOSNOSTI

Používá se pro výpočet výnosnosti investice na základě daných pravidelných peněžních toků.¹⁵

Funkce má pouze dva parametry. Prvním je pole peněžních toků, druhým (nepovinným) je odhad, kterým určujeme předpokládanou výnosnost. Odhad ve většině případů není potřeba uvádět, Excel automaticky používá hodnotu 10%.

Pro příklad uvedu srovnání dvou pětiletých investičních záměrů, u kterých známe počáteční vklad a peněžní toky v následujících letech. Srovnání je uvedeno na obrázku č. 14. Použitá funkce pro první variantu je tato:

=MÍRA.VÝNOSNOSTI(C3:C8)

Funkce pro druhou variantu:

=MÍRA.VÝNOSNOSTI(D3:D8)

¹⁵ BROŽ, Milan a BEZVODA, Václav. Microsoft Excel 2007 pro manažery a ekonomy, s. 198.

	A	B	C	D
1				
2			1. investice	2. investice
3		Poč. investice	-1 200 000 Kč	-1 500 000 Kč
4		1. rok	380 000 Kč	500 000 Kč
5		2. rok	-220 000 Kč	-240 000 Kč
6		3. rok	260 000 Kč	-120 000 Kč
7		4. rok	250 000 Kč	485 000 Kč
8		5. rok	1 400 000 Kč	1 850 000 Kč
9		Míra výnosnosti	14,4%	12,5%
10				

Obrázek č. 14: Výpočet výnosnosti investice

Závěr

Cílem práce bylo ukázat postupy a metody v Excelu, jakými lze při použití integrovaných funkcí a vlastních maker řešit ekonomické úlohy. Problematika Excelu je velmi obsáhlá, z tohoto důvodu jsem se věnoval především popisu funkcí, které jsou dle mého názoru pro ekonoma nebo manažera zásadní. Záměrně jsem uvedl příklady z různých oborů, abych na nich názorně ukázal, jak je použití Excelu rozmanité a může obsáhnout mnoho oblastí.

Neuváděl jsem práci s uživatelským rozhraním apod. Celou práci jsem psal se zaměřením na uživatele, který má základy s prací v Excelu, ale nezná použití funkcí nebo například možnosti využití kontingenční tabulky.

V první kapitole jsem se věnoval tvorbě základu účetního programu, na kterém jsem ukázal použití a syntaxi funkcí pro práci s textovými řetězci, funkce vyhledávací, podmiňovací, informační, datumové apod. Uvedl jsem postup, jak pomocí makra lze některé operace zautomatizovat.

Ve druhé části jsem se věnoval kontingenčním tabulkám. Cílem bylo ukázat, jakým způsobem se pracuje s daty v křížovém zobrazení, jak využívat agregace a počítaná pole.

V dalších částech jsem funkce pro předpovědi budoucích stavů, výpočty úvěrů a cash flow. Uvedl jsem příklady použití z různých oborů a popsal výhody použití daných funkcí.

Excel obsahuje řadu dalších užitečných funkcí, které lze pro řešení úloh použít. Představil jsem ty funkce, které se, dle mých zkušeností z praxe, používají nejčastěji a popsal je tak, aby je mohl ekonom ihned aplikovat díky znalosti jejich syntaxe a způsobu použití.

Anotace

Příjmení a jméno autora:	Trbušek Jiří
Instituce:	Moravská vysoká škola Olomouc
Název práce v českém jazyce:	Řešení ekonomických úloh pomocí maker a programů v Excelu
Název práce v anglickém jazyce:	Solving of Economic Tasks Using Macros and Programs in Excel
Vedoucí práce:	Mgr. Aleš Kepřt, Ph.D.
Počet stran:	39
Počet příloh:	0
Rok obhajoby:	2014
Klíčová slova v českém jazyce:	Excel, funkce, makro, ekonomické úlohy
Klíčová slova v anglickém jazyce:	Excel, function, macro, economic tasks

Práce je zaměřena na popis funkcí, které se používají pro řešení ekonomických úloh v Excelu. Uvedeny jsou jejich syntaxe a příklady použití. Popsány jsou funkce s obecným použitím, funkce přímo zaměřené pro ekonomické výpočty, tvorba jednoduchého makra a kontingenční tabulka.

This thesis is focused on description of the functions that are used to solve economic tasks in Excel. Included are their syntaxes and usage examples. It describes the functions of general use, function directly focused on economic calculations, creating simple macros and pivot table.

Literatura a prameny

BROŽ, Milan a BEZVODA, Václav. *Microsoft Excel 2007 pro manažery a ekonomy*. Vyd. 1. Brno: Computer Press, 2009. 435 s. ISBN 978-80-251-2116-0.

BROŽ, Milan. *Microsoft Office Excel 2007: podrobná uživatelská příručka*. Vyd. 1. Brno: Computer Press, 2007. 407 s. ISBN 978-80-251-1822-1.

ČECHOVÁ, Alena. *Manažerské účetnictví*. 2., aktualiz. a rozš. vyd. Brno: Computer Press, 2011. 194 s. ISBN 978-80-251-2831-2.

KRÁL, Martin. *Excel VBA: výukový kurz*. Vyd. 1. Brno: Computer Press, 2010. 504 s. ISBN 978-80-251-2358-4.

MRKOSOVÁ, Jitka. *Účetnictví 2013: učebnice pro SŠ a VOŠ*. 1. vyd. Brno: Edika, 2013. 291 s. Daně a účetnictví; 2013. ISBN 978-80-266-0176-0.

Seznam obrázků

Obr. 1: Účtová osnova na listu v Excelu	11
Obr. 2: Hlavička listu pro uložené účetní položky na listu v Excelu	13
Obr. 3: List pro zadání nového účetního dokladu	13
Obr. 4: Zdrojová data klientské spokojenosti pro kontingenční tabulku	21
Obr. 5: Kontingenční tabulka zobrazující průměrnou spokojenost klientů	21
Obr. 6: Kontingenční tabulka zobrazující počet hodnocení	22
Obr. 7: Zdrojová data statistik hovorů pro kontingenční tabulku	23
Obr. 8: Kontingenční tabulka zobrazující průměrnou délku hovoru	23
Obr. 9: Odhad vývoje počtu prodejů v závislosti na počtu nabídek	26
Obr. 10: Odhad počtu svezeného odpadu	27
Obr. 11: Srovnání odhadů s funkcemi Lintrend a Loglintrend	28
Obr. 12: Výpočet úmoru z pravidelné splátky	32
Obr. 13: Výpočet čisté současné hodnoty	34
Obr. 14: Výpočet výnosnosti investice	35