

**Bankovní institut vysoká škola Praha**  
**zahraničná vysoká škola Banská Bystrica**

**Tvorba konkrétnej aplikácie v programovacom jazyku**  
**VBA nad MS Excel 2010**

**Bakalárska práca**

**Viktor Kala**

**apríl 2014**

**Bankovní institut vysoká škola Praha**  
**zahraničná vysoká škola Banská Bystrica**  
**Katedra kvantitatívnych metód a informatiky**

**Tvorba konkrétnej aplikácie v programovacom jazyku**  
**VBA nad MS Excel 2010**

**Develop of specific application by using of VBA in MS Excel**  
**2010 environment**

**Bakalárska práca**

**Autor: Viktor Kala**

Informačné technológie

**Vedúci práce: PaedDr. Peter Polakovič Ph.D.**

**Banská Bystrica**

**apríl 2014**

## **Vyhlasenie**

Vyhlasujem, že som bakalársku prácu spracoval samostatne a s použitím uvedenej literatúry.

Svojím podpisom potvrdzujem, že odovzdaná elektronická verzia práce je identická s jej tlačenu verziou a som oboznámený so skutočnosťou, že sa práca bude archivovať v knižnici BIVŠ a ďalej bude prístupná tretím osobám prostredníctvom internej databázy elektronických vysokoškolských prác.

Podpis autora

V Podbrezovej dňa .....

.....

Viktor Kala

## **Pod'akovanie**

Touto cestou by som sa chcel pod'akovať vedúcemu mojej bakalárskej práce PaedDr. Petrovi Polakovičovi, Ph.D., za jeho ochotu, rady a odborné vedenie.

Osobitné pod'akovanie patrí mojej rodine a hlavne priateľke za ich podporu a pochopenie.

## **Anotácia**

**KALA, Viktor: Tvorba konkrétnej aplikácie v programovacom jazyku VBA nad MS Excel 2010.** [Bakalárska práca]. Bankovní institut vysoká škola Praha, zahraničná vysoká škola Banská Bystrica. Katedra kvantitatívnych metód a informatiky. Vedúci práce: PaedDr. Peter Polakovič Ph.D. Rok obhajoby: 2014. Počet strán: 63

Bakalárska práca sa zaoberá tvorbou aplikácie pomocou programovacieho jazyka Visual Basic for Applications nad Microsoft Office Excel a teoreticky sa zameriava na jeho vývojové prostredie. Prvá kapitola obsahuje základné údaje o programovacom jazyku Visual Basic for Applications a zachytáva priebeh vývoja Microsoft Office so zameraním na Microsoft Office Excel až po súčasnosť. Druhá kapitola konkrétne opisuje programovací jazyk Visual Basic for Applications a vysvetľuje jeho prvky. Tretia kapitola sa zaoberá už konkrétnou aplikáciou a podrobne analyzuje časti jej zdrojového kódu.

**Kľúčové slová:** VBA, Microsoft Office, Microsoft Office Excel

## **Annotation**

**KALA, Viktor: Develop of specific application by using of VBA in MS Excel 2010 environment.** [Bachelor's thesis]. Banking Institute University of Prague, Foreign University Banská Bystrica. Department of Quantitative Methods and Information Technology. Thesis supervisor: PaedDr. Peter Polakovič Ph.D. Year of defense: 2014th. Number of pages: 63

The bachelor thesis focuses on building an application using Visual Basic for Applications programming language within Microsoft Office Excel and on the theoretical aspects of this development environment. First chapter encompasses basic information about Visual Basic for Applications programming language and depicts evolution of Microsoft Office until present times, emphasizing Microsoft Office Excel. Second chapter describes Visual Basic for Applications programming language and explains its components. Finally, the third chapter deals with the specific application and thoroughly analyses parts of its source code.

**Keywords:** VBA, Microsoft Office, Microsoft Office Excel

## Obsah

Úvod .....	8
1 Programovací jazyk Visual Basic for Applications.....	9
1.1 História Visual Basic for Applications .....	9
1.1.1 Odvođené programovacie jazyky.....	10
1.1.2 Edície programovacieho jazyka Visual Basic.....	12
1.2 Možnosti využitia jazyka VBA v aplikáciách Microsoft Office .....	13
1.3 História kancelárskeho balíka Microsoft Office .....	16
1.3.1 Microsoft Office od roku 1983 po súčasnosť .....	16
1.3.2 Rozdiel medzi MS Office 2010 a MS Office 2013 .....	19
1.3.3 Aplikácie kancelárskeho balíka MS Office 2013 .....	20
1.4 Výhody a nevýhody použitia VBA v praxi .....	21
1.5 Možnosti využitia jazyka VBA nad MS Excel.....	23
1.5.1 História tabuľkových procesorov .....	24
1.5.2 História Microsoft Excel .....	25
1.5.3 Možnosť využitia makier v Microsoft Excel.....	27
2 MAKROJAZYK V MICROSOFT EXCEL.....	28
2.1 Základné údaje o Visual Basic for Applications .....	28
2.2 Objektový model Microsoft Excel.....	29
2.3 Dátové typy, premenné a konštanty v jazyku VBA.....	30
2.3.1 Deklarovanie premenných v jazyku VBA.....	36
2.3.2 Rozsah platnosti premenných v jazyku VBA.....	38
2.4 Postup pri zaznamenávaní makra v jazyku VBA .....	38
2.5 Začiatok opisu prostredia.....	41
3 Aplikácia – Faktúra .....	44
3.1 List Faktúra .....	48
3.2 List Cenník.....	50
3.3 List Odberateľa.....	54
Záver.....	57
Zoznam použitej literatúry.....	58
Zoznam obrázkov .....	61
Zoznam tabuliek .....	62
Zoznam príloh .....	63

# Úvod

Kancelárske práce boli už v minulosti podmienené použitím mnohých prác uľahčujúcich zariadení ako napríklad písací stroj, kalkulačka, fax či diaľkopis. Postupom času sa vyvíjali aj tieto zariadenia a v súčasnosti sú už mnohé nahradené novšími. Dnešná kancelárska práca sa už nezaobíde bez použitia počítačov, ktoré sú však používané nielen týmito kancelárskymi pracovníkmi, ale aj študentmi, učiteľmi a zákonodarcami. Vývoj v tejto oblasti sa teda v súčasnosti týka najmä uľahčovania práce s počítačmi.

Najpoužívanejším kancelárskym balíkom programov je Microsoft Office, ktorý umožňuje efektívne spracovanie dát v prostredí operačného systému Windows. Aj tento balík prešiel dlhým vývojom, kým dostal podobu, akú momentálne poznáme a používame. Postupom času sa menil tento kancelársky balík, čo sa týka obsiahnutých aplikácií a tieto sa hlavne zdokonaľovali. Zdokonalením môžeme chápať najmä urýchlenie a zautomatizovanie spracovania údajov.

Ešte efektívnejšie dokážu pracovať používatelia rôznych programovacích jazykov, ktorí si konkrétny programovací jazyk môžu vybrať na základe vlastných preferencií a účelu použitia. Ak však chceme ostať v pre nás známom prostredí Microsoft Office, najlepšou voľbou je Visual Basic for Applications.

Visual Basic for Applications je odvodeným jazykom Visual Basic, čo je vlastne vývojový nástroj na zvýšenie produktivity každodennej práce najmä s databázami a riešeniami pre internet vyvinutý spoločnosťou Microsoft.

Visual Basic for Applications je objektovo orientovaný programovací jazyk, ktorý je vstavaný v aplikáciách Microsoft Office. Svojim používateľom uľahčuje prácu tým, že rozširuje možnosti a funkcie spomínaných aplikácií. Najdôležitejšou vlastnosťou tohto jazyka je jednotnosť. Jednotnosť sa týka syntaxe, ktorá sa v jednotlivých aplikáciách nemení, a teda umožňuje opätovné použitie jedného kódu pre viaceré aplikácie Microsoft Office. Riešenie pre viacero aplikácií sa dá zvládnuť len obmenou objektového modelu jazyka.

Cieľom našej bakalárskej práce je preskúmať možnosti programovacieho jazyka Visual Basic for Applications v podnikovej praxi nad MS Excel a vytvorenie konkrétnej aplikácie pre reálne používanie.



# 1 Programovací jazyk Visual Basic for Applications

V súčasnosti poznáme veľa programovacích jazykov a výber jedného konkrétneho, závisí len od daného používateľa, jeho preferencií a hlavne od účelu použitia. Ak hľadáme vlastnosti ako rýchlosť, dynamika a rôznorodosť, najvhodnejší programovací jazyk je Visual Basic for Applications (ďalej len VBA). Čo sa týka VBA, nemôžeme striktne oddeliť Microsoft Excel VBA, Microsoft Word VBA a Microsoft Access VBA, pretože syntax je pre všetky tieto aplikácie rovnaká. Rozdiel je len v objektovom modeli aplikácie. A v tomto vidíme veľkú výhodu programovacieho jazyka VBA, pretože ak sa naučíme programovať v jednej aplikácii Microsoft Office (ďalej len MS Office), tieto vedomosti využijeme aj v jeho ostatných aplikáciách.

Visual Basic je vývojový nástroj, s ktorého pomocou môžeme zostavovať užitočné a skvele vyzerajúce softwarové aplikácie pre najrôznejšie prostredie. Môžeme tak vytvárať aplikácie pre operačný systém Windows, web, príručné zariadenie a pre celú radu ďalších prostredí a nasadení. Najväčšia výhoda jazyka Visual Basic spočíva v tom, že bol navrhnutý pre zvýšenie produktivity každodennej práce vývojára, najmä vtedy, keď potrebujeme pracovať s informáciami v databázach alebo vytvárať riešenia pre Internet. (Halvorson, 2008 s. 17)

## 1.1 História Visual Basic for Applications

História Visual Basic sa začala písať už v 60. rokoch, no je potrebné spomenúť aj dobre známeho a využívaného predchodcu, makrojazyk XLM. Tento jazyk používal na zapisovanie príkazov makro-hárky a bol určený len pre použitie v MS Excel. Napriek tomu, že v roku 1997 bol jazyk XLM nahradený jazykom Visual Basic a možnosť vytvárať makrá už nie je k dispozícii, spúšťať sa stále dajú.

Už v spomínaných 60. rokoch bol tento jazyk určený len na univerzitnú výučbu programovacích techník, o čom svedčí aj jeho vtedajší názov BASIC – Beginner's All-purpose Symbolic Instruction Code. Postupom času sa však rozšíril a v súčasnosti existujú stovky jeho modifikácií.

BASIC sa rokmi vyvíjal a zdokonaľoval. Príkladom môže byť aj skutočnosť, že pôvodne bol interpretovaným jazykom. Každý riadok bol pred vlastným spustením interpretovaný, čo spôsobovalo značné zníženie rýchlosti. Väčšina moderných mutácií jazyka BASIC umožňuje kód preložiť (skompilovať) do binárnej podoby, čo vedie k ďaleko väčšej rýchlosti spustenia. (Walkenbach, 2008 s. 152)

Užívateľské prostredie je dielom spoločnosti Tripod, ktorú reprezentoval Alan Cooper, s ktorým spoločnosť Microsoft podpísala zmluvu na vytvorenie programovacieho prostredia pre Windows 3.0. Spoločnosť Microsoft ho doplnila o programovací jazyk Basic a vznikol tak Visual Basic. Už vtedy bolo zrejmé, že vznikol prevratný produkt programátorstva. V roku 1992 vznikla vylepšená a rýchlejšia verzia, ktorá však bola stále k dispozícii len študentom.

V roku 1993 vznikla tretia verzia a spoločnosť Microsoft sprístupnila tento programovací jazyk pre širokú verejnosť, čo bolo základom pre vývoj samostatných aplikácií pre Windows. Hlavným rozdielom bolo pribudnutie databáz, OLE prvkov a predovšetkým integrácia VBA medzi produkty MS Office. Vďaka tomu je možné vytvárať aplikácie priamo v daných produktoch MS Office.

Štvrtá verzia vydaná o dva roky neskôr, teda v roku 1995, je reakciou na Windows 95, ale súčasťou stále bolo kompilácia zdrojového kódu pre 16 – bitovú aplikáciu. Táto verzia prináša aj nový odvodený jazyk VBScript, čím je použiteľný aj pre webstránky. Piata verzia vyšla v roku 1997 a zmenila podporu zo 16 – bitových systémov na 32 – bitové systémy. Príznačnou bola aj možnosť tvoriť vlastné CTL a OCX prvky. Šiesta verzia bola vydaná už o rok neskôr v roku 1998 a umožňuje využívať Visual Basic.NET.

Siedma verzia, ktorá nahradila predošlú, je významná najmä preto, že podporuje nielen 32 – bitovú ale aj 64 – bitovú verziu systému Office 2010. Táto verzia obsahuje aj dve nové konštanty, a to konštantu VBA 7 a Win64. Konštantu VBA 7 zisťuje, či je daná aplikácia vytvorená pomocou VBA 7 alebo VBA 6 a konštantu Win64 testuje, či je aplikácia naprogramovaná v 32 – alebo 64 – bitovom prostredí.

### **1.1.1 Odvodené programovacie jazyky**

Spoločnosť Microsoft vyvinul odvodené jazyky od Visual Basic, ktoré sa používajú pri skriptovaní. Niektoré z nich:

- Visual Basic for Applications (VBA) – tento jazyk môžeme nájsť najmä v aplikáciách Microsoft (MS Office), ale aj v iných produktoch ako Solid Works, AutoCAD, Word Perfect Office 2002, Sage 300 ERP. Spoločnosť Microsoft skončila vývoj Visual Basic verziou 6.0, ale v roku 2010 predstavila VBA 7.0, ktorý poskytol rozšírené funkcie a 64 – bitovú podporu. V súčasnosti sa využíva VBA 7.1 pre MS Office 2013.
- VBScript– „je skriptovací jazyk určený pre vkladanie kódu do webových stránok a bežné skriptovanie v WSH. Vznik jazyka je spolu so vznikom jazyka JScript považovaný za odpoveď firmy Microsoft na vznik JavaScript.“ (<http://msdn.microsoft.com/>, 21.12.2013)  
Jediný kompatibilný internetový prehliadač pre vykonávanie skriptov je Internet Explorer.
- Visual Basic .NET–pracuje na rozhraní .NET, ktoré výsledný program neskompiluje priamo do strojového kódu, ale do akéhosi medzikódu – CIL (common Intermediate Language). Tento medzikód sa zabalí do súboru EXE a distribuuje sa na počítač k užívateľom v tejto podobe. Nie je to priam strojový kód. Až keď sa program spúšťa, skompiluje sa zo strojového kódu. Navyše sa kompiluje len to, čo je z programu potrebné pred prvým spustením. (<http://dotnetportal.cz/>, 10.11.2013)
- Star Office Basic – známy aj ako Open Office Basic, je druh programovacieho jazyka BASIC, ktorý sa šíri prostredníctvom OpenOffice.org a Libre Office
- Gambas – jeho autorom je Benoit Minisini, programátor, ktorý mal mnoho skúseností s jazykom Visual Basic a takto sa snažiť odstrániť jeho chyby. Gambas je určený pre Linuxové platformy a je licencovaný licenciou GNU GPL. Tento program je pre užívateľov ľahší a prívetivejší, no nie je kompatibilný s jazykom Visual Basic.
- Win Wrap Basic – nízko nákladový a výkonný NET/WPF/COM komponent, ktorý je zároveň alternatívou k ActiveX, VBScript, VB6, VBA. K dispozícii je v 32-bitovej a 64 – bitovej verzii Windows. Softvérový balík WWB sa používa v spojení s vývojovými nástrojmi spoločnosti Microsoft, vrátane Visual Studio, Visual Studio.NET a skriptovacích nástrojov ActiveX. Predvolená prípona pre programy napísané v tomto jazyku je .wwb. (<http://winwrap.com/>, 10.12.2013)
- LotusScript – „je programovací script jazyk, ktorý:
  - Je syntakticky podobný VisualBasicu,
  - Je použiteľný v určitých častiach aplikácie Notes (teda je platformovo nezávislý)

- Sprístupňuje súpravu objektov aplikácie Notes s ich vlastnosťami a metódami,
- Je súčasťou integrovaného vývojového prostredia Lotus Domino Designera,
- Výrazne prekračuje možnosti tzv. @Formula language,
- Umožňuje zoskupovať vlastné funkcie a procedúry do tzv. knižníc LS“ (<http://itnews.sk/>, 13.12.2013)

### 1.1.2 Edície programovacieho jazyka Visual Basic

Visual Basic má využitie u profesionálov, ale aj u amatérov. K jeho základným charakteristikám patrí jednoduchosť, efektnosť návrhu a vyspelé technológie. Z tohto pohľadu rozoznávame niekoľko edícií:

- VB Express Model – tento základný a najjednoduchší model je k dispozícii na stiahnutie zadarmo. Využíva automatické dokončovanie syntaxe a zvýraznenie kódu, ale aj debugger na jemné doladenie kódu a editor vlastností. Je rozdelený na Windows Deskopt a for Web, a teda programovať sa dajú aj deskoptové aplikácie aj aplikácie pre web.
- VB Working Model – táto verzia je dostupná na stiahnutie na oficiálnej stránke Microsoftu, no nevýhodou je, že generovanie EXE súborov tu nie je možné a nie je dostupný ani online pomocník.
- VB Learning Edition – táto verzia je už spoplatnená, rovnako tu nie je možné generovanie EXE súborov, no rozdiel je v tom, že sú k dispozícii príručky Learn VisualBasic Now a príručka Visual Studio. Pridanou hodnotou je možnosť vytvárať internetové aplikácie.
- VB Professional Edition – plnohodnotná verzia obsahujúca kompletnú dokumentáciu. Obsahuje moduly Internet Information Server Application Designer, Integrated Data Tools a Data Environment. Možnosť generovať EXE súbory a tvoriť ActiveX prvky je dostupná.
- VB Enterprise Edition – táto verzia je na rozdiel od predchádzajúcej ešte obohatená o Back Office tools, SQL Server, Microsoft Transaction Server, Internet Information Server, Visual SourceSafe, SNA Server. Tlačенá dokumentácia obsahuje okrem Visual Studio Enterprise aj Microsoft Developer Network CD. (<http://msdn.microsoft.com/>, 13.12.2013)

## 1.2 Možnosti využitia jazyka VBA v aplikáciách Microsoft Office

Aplikácie sa používajú na zautomatizovanie úloh v jednotlivých aplikáciách kancelárskeho balíka MS Office a preto ich môžeme nazvať aj automatizačné aplikácie. Konkrétna definícia automatizačnej aplikácie v teórii neexistuje, no môžeme ju chápať ako jeden alebo viacero užitočných súborov.

J. Walkenbach považuje za automatizačnú aplikáciu súbor, ktorý:

- jedinečným spôsobom dopĺňa hostiteľský program (Word, Excel, Outlook..) o novú funkcionálnu, prostredníctvom ktorej rieši užívateľ určitý problém,
- vykonáva to, čo sa od neho naozaj očakáva,
- vždy poskytuje presné výstupy v čo možno najkratšom čase,
- ponúka vhodné prostredie pre riešenie daného problému,
- je postavený na vhodných a v rámci prostredia efektívnych metódach a algoritmoch,
- zachytáva chyby počas behu programu a príslušným spôsobom ich ošetruje
- je „blbuvzdorný“, t.j. nepoškodí sa pri neodbornej manipulácii,
- disponuje intuitívnym a konzistentným užívateľským rozhraním,
- je dobre zdokumentovaný tak po technickej, ako i užívateľskej stránke (technická a užívateľská príručka),
- je dobre prenositeľný. (Walkenbach, 2008, s. 125)

Aplikácie v MS Office uľahčujú svojim používateľom prácu a preto si priblížime dôvody, prečo je vlastne vhodné ich vytvárať.

Môžeme použiť existujúce zručnosti na vybudovanie aplikácie a využiť známe nástroje, jazyky a hostingové služby. Môžeme použiť ľubovoľný jazyk ako HTML, JavaScript, PHP alebo .NET a nami obľúbené webové rozvojové nástroje, vrátane VisualStudio a nový webovo-založený nástroj “Napa“ Office 365.

Používame akékoľvek hostingové služby na spustenie našej aplikácie. Môžeme pripojiť našu aplikáciu so všetkými súkromnými alebo verejnými webovými službami.

Tvorba, aktualizácia a udržiavanie vlastných aplikácií je zrýchlené v cloude. Publikovanie a distribuovanie schválených aplikácií v rámci organizácie je možné prostredníctvom aplikácie interného katalógu. Zaujímavá je aj možnosť predávať svoje aplikácie v obchode Office.

Webové aplikácie pre MS Office sú aplikácie, ktoré kombinujú to najlepšie z cloudových služieb a internetových technológií v rámci známeho prostredia Office. Aplikácia môže bežať v rôznych prostrediach a klientoch, vrátane cloudu a on-premises.(<http://msdn.microsoft.com/>, 13.12.2013)

Spracovanie veľkého množstva údajov v programoch kancelárskeho balíka MS Office je časovo náročné. Pritom sa dá využiť rutina a opakovanie týchto údajov na urýchlenie a zautomatizovanie tohto spracovania. Na toto sa používajú makrá a programy vytvorené programovacím jazykom Visual basic for Applications.

Visual Basic for Applications je objektovo orientovaný programovací jazyk. Tento programovací jazyk je jediný, ktorý podporujú aplikácie MS Office. VBA uľahčuje svojim používateľom prácu, pretože umožňuje aj úlohy, ktoré Excel a Word neumožňujú. „K týmto úlohám patria:

- navrhovanie formulárov slúžiacich na zber dát a vytváranie ich zostáv,
- dynamické získavanie údajov z externých databáz,
- vytváranie zostavy kontingenčných tabuliek stlačením jedného tlačidla,
- zapisovanie kódu, ktorý dynamicky komunikuje s užívateľom pomocou dialógových okien a vstupných polí,
- dočasné ukladanie údajov do pamäte,
- reagovanie na udalosti, ako je otvorenie listu alebo zmena údajov na liste,
- prípadne riadiť inú aplikáciu, napr. Word, Access.“(Kráľ, 2012, s. 14)

Niektorí používatelia sa stretli s makrami pri otvorení niektorej aplikácie MS Office, a keďže nevedeli ako ich použiť, považovali ich buď za niečo záporné ako vírus alebo niečo náročné ako programovanie. Väčšina makier sa však tvorí ľahšie, ako sa zdá na prvý pohľad. Makrá teda automatizujú opakujúce sa činnosti a tým šetria čas, no rozširujú aj možnosti danej aplikácie MS Office. Väčšina makier vytváraných v MS Office sú napísané v jazyku Visual Basic for Applications. Makro sa môže vytvoriť automatickým zapísaním na základe užívateľom uskutočnených akcií. V kancelárskom balíku MS Office môžeme makrá zaznamenať v aplikáciách MS Word, MS Excel, MS PowerPoint, MS Visio a MS Project. V aplikáciách MS Access, MS Outlook, MS FrontPage a MS Publisher môžeme vytvárať vlastné makrá VBA.

## **Microsoft Word**

V tejto aplikácii môžeme makrá zaznamenávať aj vytvárať. Jazyk VBA tu používa terminológiu, ktorú používateľ, ktorý vie zaznamenávať makrá, pozná z každodenného použitia a tým získava znalosti na úpravu a vytváranie vlastných makier. Pri čom konkrétne pomôže VBA:

- tvorba hromadnej obchodnej korešpondencie
- editácia
- štítky
- porovnávanie textu
- automatické prehl'adávanie textu

## **Microsoft Excel**

V MS Excel sú makrá veľmi užitočné, pretože ich možno použiť na automatizáciu bežných činností, rozšírenie možností už používaných funkcií alebo vytvoriť nové funkcie.

Pri čom konkrétne pomôže VBA:

- import externých dát
- hlásenia mailom podľa zadaných kritérií napr. bazáre, ponuky firiem
- výpočtové operácie
- grafy
- prehľady
- databázy menšieho rozsahu

## **Microsoft Access**

Makrá sú v aplikácii Access veľmi dôležité, pretože môžu významne rozšíriť funkčnosť databáz. Napriek tomu, že tu nie je možné makrá zaznamenávať, môžeme zapisovať kód VBA alebo môžeme makrá vytvárať pomocou nástroja Tvorca makier.

Pri čom konkrétne pomôže VBA:

- rovnaké operácie ako v MS Excel, ale vo väčšom rozsahu
- väčšia stabilita a bezpečnosť
- zdieľanie pre viacerých užívateľov
- porovnávanie dodávateľských dát s fakturáciou
- evidencia skladov

## **Microsoft Outlook**

Pri čom konkrétne pomôže VBA:

- automatické zasielanie a spracovanie mailovej pošty, kontaktov, úloh
- hromadné odosielanie mailov
- práca s nedoručeným mailom
- automatická akceptácia u príjemcov
- nastavenie automatickej odpovedi

## **Microsoft FrontPage a Microsoft Publisher**

Vytvorením makier v týchto publikáciách ušetríme čas automatizovaním činností. V aplikácii FrontPage môžeme vytvoriť a používať doplnky rozširujúce možnosti tejto aplikácie.

### **1.3 História kancelárskeho balíka Microsoft Office**

Microsoft Office je v našich krajinách najpoužívanejším kancelárskym softvérom. Existuje už zhruba dvadsať rokov a za tú dobu si vybojoval svoje miesto na slnku aj v českom a slovenskom prostredí. (Budaj et al, 2010, s. 21)

Kancelársky balík programov Microsoft Office je, ako už z jeho názvu vyplýva, určený najmä pre spracovanie dát v kanceláriách, no využívajú ho aj študenti, vedci, učitelia, či zákonodarcia a advokáti. Z minulosti vieme, že MS Office nahradil mnohé kancelárske zariadenia ako strojopis, kalkulačka, diaľkopies či fax. V súčasnosti je skôr potrebné internetové pripojenie, a to nielen na sťahovanie aktualizácií, ale aj ako prostriedok s vonkajším svetom prostredníctvom elektronickej pošty.

Je skutočným balíkom programov, čo znamená, že sa nejedná len o jednu aplikáciu. Ide o konglomerát programov, ktoré však dokážu operatívne a účinne spolupracovať. Nie sú to v žiadnom prípade programy jednoúčelové, ich možnosti sú veľmi široké. (Pecinovský et al, 2011, s. 17)

#### **1.3.1 Microsoft Office od roku 1983 po súčasnosť**

Prvú verziu má na svedomí programátor Richard Brodie, ktorý ju v roku 1983 vytvoril pre spoločnosť IBM a pre operačný systém DOS. Táto verzia obsahovala viacero chýb a nebola veľmi obľúbená. Druhá vylepšená verzia vydaná v roku 1984 bola určená



pre Macintosh mala možnosť využiť kurzívu a tučné písmo. V roku 1989 vznikol prvý MS Word pre Microsoft Windows, ktorý zvíťazil s podobným produktom Word Perfekt, ktorý verziu pre tento operačný systém nemal.

Office z roku 1990 obsahoval textový editor Word, tabuľkový kalkulátor Excel a PowerPoint. Tieto tri programy sú hlavnými a zakladajúcimi komponentmi súčasného MS Office. V roku 1991 bola vydaná verzia MS Office 1.5, ktorá obsahovala aktualizovaný MS Excel. Už o niekoľko mesiacov neskôr bola vyvinutá ďalšia verzia MS Office 1.6, ktorá obsahovala rozšírenú základňu aplikácií o Mail (súčasný MS Outlook). V roku 1992 vyšla verzia MS Office 3.0 známa aj ako MS Office 92, pretože všetky ďalšie verzie sú označované rokom vydania. Zaujímavosťou je, že je to aj prvá verzia dodávaná na CD. (Predošlé verzie boli dostupné len na diskete.)

Hneď začiatkom roka 1994 sa objavila verzia pod starým názvom MS Office 4.0. Je to prvá verzia vydávaná v dvoch edíciách, a to Professional a Standard. Lacnejší Standard obsahoval aplikácie už predošlých verzií a drahší Professional prvýkrát uviedol MS Access. V tom istom roku vyšla inovovaná verzia 4.3., ktorá bola posledná 16 – bitová verzie podporujúca Windows 3.xx a Windows NT 3.5.

V roku 1995 sa spoločnosť Microsoft opäť rozhodla, že verzie bude označovať rokom vydania, no toto rozhodnutie malo znovu neskoršie výnimky. Aj napriek tomu bola táto verzia prelomová. Charakteristická bola podpora pre Windows 95 a Windows NT 3.51 a vyššie verzie. Prvýkrát boli zosúladené čísla verzie celej sady s číslami jednotlivých programov. Zo súboru programov sa vytratil Mail a pribudol elektronický diár Schedule Plus, ktorý obsahuje tiež znaky terajšieho MS Outlook.

MS Office 97 vychádza už podľa názvu v roku 1997 a prvýkrát sa v nej objavil MS Outlook. Táto verzia ako posledná podporovala Windows NT 3.51.

V roku 1999 vyšla verzia MS Office 2000. Zaujímavosťou je, že je to posledná verzia, keď sa rok vydania nezhoduje s číslom verzie. Ako posledná tiež podporovala Windows 95. Prvýkrát programovú základňu rozšíril program FrontPage určený na tvorbu webových stránok a program PhotoDraw.

V roku 2001 vyšla zatiaľ jediná verzia, ktorá neobsahovala rok vydania alebo číslo verzie a bola to verzia Windows XP. Tento názov bol odvodený od dodnes rozšíreného Windows XP. Spomínaná verzia bola poslednou verziou, ktorá podporovala Windows 98,

Windows Me a Windows NT 4.0. Naopak to však bola prvá verzia MS Office, ktorá podporovala Windows 2000 a Windows XP. Touto verziou boli z MS Office definitívne vylúčené programy pre úpravy fotiek. V edícii Professional Special Edition sa prvýkrát objavil program Share Point.

V roku 2003 vyšla verzia MS Office 2003, ktorá ako posledná podporovala Windows XP. Zaujímavosťou bol nový progresívny dizajn. Najvyššia edícia obsahovala aj nový program InfoParh. Naopak vznikla aj skromnejšia edícia pre študentov a pedagógov, ktorá obsahovala len MS Word, MS Excel, MS PowerPoint a MS Outlook. Editor webových stránok FrontPage sa oddelil od MS Office a je šírený samostatne.

Ďalším míľnikom bol MS Office 2007. Medzi novinky patril koncept Ribbon, čiže pás kariet a zmena užívateľského rozhrania aplikácií. Táto verzia má osem rôznych edícií, ktoré sú delené z hľadiska ceny a komerčného či nekomerčného použitia. Verzia predtým určená pre študentov a pedagógov bola určená už aj pre domácnosti, no ochudobnená o MS Outlook. Najvyššie predovšetkým firemné edície po prvýkrát obsahovali program OneNote a nástroje Groove, Communicator, Enterprise Content Manager a ElectronicForms.

Ďalšia verzia MS Office 2010, nazývaná aj verzia 14.0, pretože 13.0 bola preskočená. Aj keď oficiálne dôvody neboli uvedené, je pravdepodobné, že tak ako u viacerých firiem, ktoré toto číslo preskočili, išlo o poverčivosť. Počet edícií bol zredukovaný na 6 a zreorganizovaný. V predošlých verziách boli programy k edíciám pridávané zmätočne, no táto verzia ako prvá zorganizovala edície tak, že každá vyššia edícia obsahuje všetky programy predošlej nižšej edície. Čo sa týka programov, tak nepribudol žiaden, no program Communicator bol nahradený sofistikovanejším Lync. Táto verzia zmenila a sprehladnila aj rozhranie Ribbon, ktoré bolo rozšírené aj do aplikácie MS Outlook. Zaujímavosťou je, že táto verzia je ako prvá verzia dostupná v 64 – bitovom variante. Odporúčaný je však aj tak 32 – bitový variant z dôvodu kompatibility s tretími stranami. Spoločnosť Microsoft sa už niekoľko rokov pohrávala s myšlienkou prepojiť desktopovú sadu MS Office s online funkciami - zdieľanie a skladovanie dokumentov, tabuliek a prezentácií. Až táto verzia však priniesla túto možnosť. Vyvinutá bola webová kancelárska sada Office Web Apps, ktorú môžeme považovať za odpoveď na GoogleDocs. Táto sada prenáša do internetového prostredia zjednodušenú verziu programov MS Word,

MS Excel, MS PowerPoint a OneNote a je k dispozícii zdarma. Služba SkyDrive funguje ako online úložisko. (<http://lupa.cz/>, 11.12.2013)

### **1.3.2 Rozdiel medzi MS Office 2010 a MS Office 2013**

16. júla 2013 vyšla zatiaľ posledná verzia MS Office 2013, ktorá je zameraná najmä na zjednodušenie práce s údajmi a ukladanie súborov do SkyDrive. Čo sa týka inštalácie, môžeme povedať, že sa v ničom nelíši od verzií 2007 a 2010.

Pri spustení MS Word vidíme prvú zmenu – inovovaná úvodná obrazovka. Pás kariet Ribbon je rovnaký ako v predošlých verziách, no rozdiel je v tom, že názvy kariet sú písané veľkými písmenami. Malý rozdiel je takisto vo farebnosti ikon. Novinkou je karta Design, ktorá obsahuje funkcie, ktoré boli v predošlých funkciách len inde zaradené. Zaujímavou novinkou pre používateľov, ktorí využívajú MS Word na čítanie je možnosť zväčšiť text a posúvať stránky horizontálne.

Po kliknutí na File vidíme, že nová verzia uprednostňuje ukladanie ako prvé SkyDrive a až potom miesta v počítači. Zjednodušená je aj manipulácia s vloženými obrázkami a inými objektami, keď sa vedľa objaví ponuka príkazov na prácu s nimi.

Ďalšie zmeny sú pri tímovej práci s dokumentom na úložisku SkyDrive alebo SharePoint. Prioritne je zmenené zobrazenie Allmarkup. V tejto verzii nie sú viditeľné prečiarknuté časti textu a farebne zvýraznené dopísané časti. Vložené komentáre obsahujú aj fotku autora a informáciu o čase komentovania.

Zmena sa týka aj prehrávania online vložených videí, ktoré sú prehrané vo zväčšenej forme a až po skončení sú zobrazené vo veľkosti zadaného textu. PDF súbory je možné po otvorení a úprave uložiť nielen v pdf formáte.

MS Excel prináša zjednodušenie vytvárania grafov a tabuliek a pomoc pri výbere najvhodnejšej analýzy. Pre skúsenejších používateľov je tu možnosť napojiť sa na SQL server, Power Pivot a Power View. Pribudli aj nové karty a tlačidlá pre navrhované riešenia grafov, tabuliek, formátov, súčtov a minigrafov.

Zmeny v používateľskom prostredí nie sú až také markantné ako pri MS Office 2007. Grafika a farebnosť je len mierne zmenená, čiže používateľ by ho mal vedieť bez problémov používať. (<http://zive.sk/>, 25.11.2013)

### 1.3.3 Aplikácie kancelárskeho balíka MS Office 2013

MS Office 2013 obsahuje všetky aplikácie ako predošlá verzia, a takisto je členená do rôznych balíkov s rôznym súborom dostupných aplikácií. Ceny týchto balíkov sa odvíjajú od počtu zahrnutých aplikácií a aj od toho, či je balík určený pre jeden alebo viac počítačov (viď tabuľka 1). Samostatne sa dajú zakúpiť aj jednotlivé aplikácie, čo sa však finančne veľmi neoplatí. Jednotlivé balíky teda môžu obsahovať jednu samotnú aplikáciu alebo aj všetky aplikácie z týchto:

- Microsoft Office Word – textový proces, používaný na spracovanie textu. Ide však o integrujúcu aplikáciu, pretože je v nej možné spustiť dáta spracované aj inými aplikáciami, a to napr. tabuľku, graf, obrázok, či databázu.
- Microsoft Office Excel – tabuľkový kalkulátor, používaný prednostne na prácu s tabuľkami a grafmi.
- Microsoft Office Outlook – elektronický diár, ktorý v sebe nesie správu elektronickej pošty, kalendár, plánovač a adresár.
- Microsoft Office PowerPoint – program používaný na tvorbu prezentácií, ale aj na tvorbu vektorových obrázkov, ktoré môžu byť exportované vo formáte WMF.
- Microsoft Office Access – program používaný na vytváranie a správu veľkých súborov s dátami hromadného charakteru, teda databáz.
- Microsoft Publisher – program určený na tvorbu profesionálnych marketingových a komunikačných publikácií.
- Microsoft OneNote – digitálny poznámkový blok, určený na operatívne zaznačovanie informácií a poznámok.
- InfoPath – aplikácia určená na zhromažďovanie podnikových informácií do vytváraných formulárov.
- Lync – aplikácia určená na zjednodušenie komunikácie a spolupráce so vzdialenými používateľmi. (Weverka, 2013 s. 9)

Tabuľka 1 Porovnanie cien kancelárskeho balíka Microsoft Office 2013

	Cena za aplikáciu (v EUR)	Office 2013 pre jeden PC			Multilicenčné balíky (5 a viac licencií)	
		Office 2013 pre študentov a domácnosti	Office 2013 pre podnikateľov	Office 2013 professional	Office 2013 standard	Office 2013 Professional Plus
Word	135	X	X	X	X	X
Excel	135	X	X	X	X	X
PowerPoint	135	X	X	X	X	X
Outlook	135		X	X	X	X
Access	135			X	X	X
Publisher	135			X	X	X
OneNote	69	X	X	X	X	X
InfoPath						X
Lynce						X
Cena balíka (v EUR)		139	269	539	1	2

<sup>1</sup> Cena sa uvádza v mesačných splátkach a závisí od počtu licencií

<sup>2</sup> Cena sa uvádza v mesačných splátkach a závisí od počtu licencií

Zdroj: Vlastné spracovanie

## 1.4 Výhody a nevýhody použitia VBA v praxi

Napriek tomu, že je tento jazyk jeden z najpoužívanejších a má teda svojich zástancov a používateľov, ktorí vidia a preferujú hlavne jeho výhody, nájdu sa samozrejme aj jeho odporcovia, pre ktorých sú viditeľnejšie jeho nevýhody, pre ktoré ho nepreferujú alebo vôbec nepoužívajú.

Medzi jeho najlepšie charakteristiky patrí rýchlosť, jednoduchosť použitia a spolupráca s inými aplikáciami Windowsu. Nemôžeme nespomenúť aj výhodu šetrenia času, ktorú dosahuje VBA automatizovaním opakovaných činností. Ak chceme lepšie pochopiť jeho výhody, je dobré ho porovnať s iným programovacím jazykom. Na porovnanie sme si vybrali AutoLISP.

Ako prvú výhodu VBA oproti AutoLISPU vidíme už spomenutú integráciu s inými aplikáciami, ktorá AutoLISPU chýba, a teda nám VBA ukazuje, aká je táto vlastnosť dôležitá a využiteľná. Je zrejmé, že VBA je sofistikovanejší programovací jazyk, ktorý už nemá toľko nevýhod ako AutoLISP, no stále si zachováva jednoduchosť programovania. Medzi jeho výhody môžeme zaradiť aj vizuálnosť nástrojov tvorby dialógových boxov, u ktorých nie je potrebné programovať užívateľské prostredie, ale jednoducho ho tvoriť z ponúknutých nástrojov, ktoré sa dajú následne meniť a nastavovať.

Nespornou výhodou VBA je aj jeho nízka cena, ktorá vyplýva zo skutočnosti, že na vytvorenie programu nie je potrebné kupovať finančne náročný licencovaný program, pretože tento je implementovaný v aplikáciách MS Office. Ďalším plusom pre VBA je skutočnosť, že nie je potrebná inštalácia vývojového prostredia, ktoré sa nachádza v aplikáciách MS Office a tým pádom je uľahčená aj jeho distribúcia, ktorá sa uskutočňuje len prostredníctvom kopírovania daného súboru.

Za výhodu považujeme aj to, že syntax jazyka sa nemení, a tak môže používateľ jej znalosťou vytvárať program v rôznych aplikáciách MS Office. K mnohým spomenutým výhodám však musíme objektívne pridať aj nevýhody, ktoré VBA má. Jednou z najviac vytýkaných vlastností VBA je limitované množstvo prvkov ActiveX.

VBA pracuje len v interpretovanom móde, čo je tiež považované za nevýhodu. Nevýhodou je aj nižšia rýchlosť, ktorá však nie je badateľná pri bežných kancelárskych aplikáciách. Medzi nevýhody patrí aj skutočnosť, že VBA pracuje priamo so zdrojovým kódom, ktorý je teda nechránený.

Na jednej strane môžeme považovať implementáciu VBA v aplikáciách MS Office za výhodu, no existuje tu aj nevýhoda z tejto skutočnosti. Keďže MS Office sa pravidelne a dosť často aktualizuje, môže nastať problém s kompatibilitou starších a novších verzií.

Pre jasnejšie a prehľadnejšie zhrnutie výhod a nevýhod VBA sme sa rozhodli vložiť ich do tabuľky.

Tabuľka 2 Výhody a nevýhody programovacieho jazyka VBA

VBA	
Výhody	Nevýhody
Rýchlosť	Limitované množstvo prvkov ActiveX
Jednoduchosť použitia	Interpretovaný mód
Spolupráca medzi aplikáciami	Nižšia rýchlosť
Šetrenie času	Nechránenosť zdrojového kódu
Jednoduchosť tvorby užívateľského prostredia	Kompatibilita verzií
Cena	
Bez nutnosti inštalácie	
Distribúcia kopírovaním súborov	
Nemennosť syntaxe	

Zdroj: Vlastné spracovanie

## 1.5 Možnosti využitia jazyka VBA nad MS Excel

MS Excel patrí medzi tabuľkové procesory. Ako už z názvu vyplýva, ide o program, ktorý spracúva vzájomne prepojené údaje v tabuľkách vo forme riadkov a stĺpcov. Nejde len o ukladanie údajov, ale aj o prácu s nimi, ako sú výpočty, prehľady a následné prezentovanie výstupov pomocou grafického zobrazenia.

Medzi základné funkcie tabuľkových procesorov podľa Poura patria:

- vytváranie tabuľky, jej aktualizácie, zobrazenie, tlač,
- spracovanie všetkých potrebných kalkulácií s numerickými dátami uloženými v bunkách tabuľky,
- spracovanie najrôznejších typov grafov nad hodnotami uloženými v tabuľke,
- formátovanie dokumentu a jednotlivých buniek, nastavenie výšky riadkov, šírky stĺpcov, formátovanie nadpisov
- spracovanie záhlavia a päty tabuľky, vrátane číslovania strán
- vkladanie schém a obrázkov do tabuľky,
- výberové filtre, súhrny a ďalšie dátové operácie,
- spracovanie kontingenčných tabuliek,

- vytváranie akcií (makier), ktoré je možné vyvolať na základe stlačenia tlačidla, vytváranie základných tabuľkových aplikácií,
- ďalšie funkcie rozširujúce možnosti práce s dátami. (Pour et al, 2009, s. 95)

### 1.5.1 História tabuľkových procesorov

V súčasnosti sme už zvyknutí na existenciu tabuľkových procesorov, no môžeme ich považovať za náhradu a hlavne vylepšenie v minulosti používaných obrovských počítačov a kalkulátorov.

Za prvý tabuľkový procesor sa považuje VisiCalc vytvorený v roku 1978. Zaujímavosťou je, že v tej dobe sa nevyskytovali žiadne osobné počítače. Aj napriek tomu, že VisiCalc bol navrhnutý pre Apple II, ktorý sa s dnešnými počítačmi nedá ani porovnať, základ tabuľkového procesoru už vtedy tvorili riadky, stĺpce a zápis vzorcov, ktoré sa používajú dodnes. (Breden, 2009 s. 23)

Zároveň bol vyvíjaný aj iný tabuľkový procesor SuperCalc pre počítače s operačným systémom CP/M. Zlomovým rokom bol rok 1981, kedy IBM zahájil éru osobných počítačov, a tak bolo potrebné VisiCalc portovať na novú hardwarovú platformu, čo sa týkalo aj SuperCalcu. Aj napriek zmenám, nemôžeme vtedajšie procesory porovnávať so súčasnými najmä preto, že text v bunke nemohol byť dlhší ako samotná bunka. Ak by bol text dlhší, musel by sa napísať do vedľajšej bunky. No až v roku 1983 začal poriadny konkurenčný boj na poli s tabuľkovými procesormi. Firma Lotus vyvinula produkt 1-2-3, ktorý bol síce finančne náročný, no ihneď predbehol VisiCalc, čo mu ostalo ešte niekoľko rokov. Môžeme povedať, že úspech 1-2-3 bol zaslúžený, pretože vylepšil všetky základné nástroje VisiCalcu a SuperCalcu a navyše bol určený pre 16 – bitovú verziu IBM PC AT. Medzi najväčšie výhody tohto editoru boli považované rýchlosť spôsobená zapisovaním textu priamo do obrazovej pamäti a systém interaktívnej nápovedy.

Ďalšie úspešné produkty firmy Lotus boli 1-2.3 Release, 1-2-3 Release 1A a Release 2. Release 2 ako prvý tabuľkový procesor predstavil doplnky, ktoré rozširovali funkcie aplikácie a mal aj vylepšenú správu pamäti (štyrikrát viac riadkov ako predošlá verzia). Úspech týchto produktov prilákal aj nie veľmi poctivých konkurentov, u ktorých nebolo výnimkou porušovanie autorských práv s cieľom znížiť cenu svojich vlastných



výrobkov. Takýmto konkurentom bola spoločnosť Paperback s jeho produktom VP Planner.

V roku 1989 prišla s verziou 1-2-3 Release 3 závažná zmena vo forme viacstránkových pracovných listov, no používatelia si stále viac uvedomovali nedostatky pri kvalite tlačeného výstupu. V roku 1990 zaznamenala spoločnosť Lotus neúspech, pretože nielenže neskoro reagovali na uvoľnenie Windowsu 3.0, ale ich reakcia vo forme 1-2-3 for Windows nemala vôbec žiadny úspech a znamenala sklamanie pre väčšinu používateľov, z čoho vyplýva, že sa ani neuplatnila.

Adekvátnou náhradou a očistením mena Lotusu bola verzia 1-2-3 Release pre Windows. Existovala takisto aj verzia 1-2-3 pre DOS, no kvôli nekompatibilitě s produktom Windows to bola aj posledná verzia pre DOS. V roku 1994 tvorili tabuľkové procesory rôzne firmy ako Word Perfect International a Borland, no tie boli skupované firmami Novell a Corel Corporation.

### **1.5.2 História Microsoft Excel**

História produktu Microsoft Excel siaha až do 80. rokov, kedy v roku 1982 spoločnosť Microsoft zverejnila svoj prvý tabuľkový procesor MultiPlan. Tento bol určený pre operačný systém CP/M a bol príliš zložitý, čo zapríčinilo jeho neúspech.

V roku 1985 bola vyvinutá prvá verzia MS Excel, ktorý bol na rozdiel od MultiPlanu graficky orientovaný a určený pre Macintosh.

V roku 1987 bola distribuovaná prvá verzia MS Excel pre Windows. Problémom bola nedostupnosť Windowsu, a tak bol MS Excel šírený aj so špeciálnou verziou Windowsu, ktorá dovoľovala spustiť len MS Excel. Táto verzia bola nazvaná MS Excel 2.0 a bola inovovaná verziami 2.1 a 2.2. V druhej verzii sa ako prvý objavil jazyk makier (XLM). Jazyk XLM bol neskôr nahradený jazykom Visual Basic for Applications. V roku 1990 vyšla verzia MS Excel 3 pre Windows, ktorá mala vylepšený vzhľad aj funkčnosť a navyše obsahovala panel nástrojov, možnosti kreslenia a výkonný nástroj pre optimalizáciu.

MS Excel 4 vyšiel v roku 1992 a bol charakterizovaný zjednodušením používania, zvýšením výkonnosti a prepracovanejším prostredím.

O dva roky neskôr, v roku 1994, sa objavil MS Excel 5, ktorý bol konkurencieschopný v porovnaní s produktom 1-2-3 Release 5. Táto verzia je prvá, v ktorej sa objavil jazyk VBA. MS Excel 95, známy tiež ako MS Excel 7, vyšiel súčasne s Windows 95. Vzhľadom sa veľmi nelíšil od svojho predchodcu, no väčšina kódu jadra aplikácie bola prepísaná, čo zabezpečilo zvýšenie rýchlosti. Používaný súborový formát sa nezmenil, čo však skomplikovalo kompatibilitu s vylepšeným jazykom VBA.

V roku 1997 vyšiel MS Excel 97 ako súčasť kancelárskeho balík MS Office 97, ktorý obsahoval veľa noviniek a nové rozhranie pre vývoj aplikácií v jazyku VBA. MS Excel 2000 bol uvoľnený už začiatkom roku 1999 a jeho rozšírenie sa týkalo predovšetkým programovania a možností Internetu.

MS Excel 2002 známy aj ako MS Excel XP neponúka veľa noviniek, no za zmienku stojí možnosť obnoviť poškodené súbory. MS Excel 2003 mal len zanedbateľné množstvo noviniek, ku ktorým patrili prvky správy užívateľských oprávnení a nový systém nápovedy.

Najprepracovanejšou verziou je verzia MS Excel 2007, kde došlo k prepracovaniu užívateľského rozhrania s pásom kariet, tisícnásobnému zväčšeniu mriežky, zdokonaleniu tabuliek, zmene podmieneného formátovania, zásadným úpravám vzhľadu grafov a tém dokumentov. (Walkenbach, 2008, s. 46)

Prioritou MS Excel 2010 bolo zvýšenie rýchlosti a efektivity a nie dramatická zmena vzhľadu. „Vylepšenia boli sústredené predovšetkým do oblasti spracovania veľmi rozmerných dát či kontingenčných tabuliek. MS Excel sa tak dočkal spresnenia výsledkov matematických a štatistických funkcií, čo využijú predovšetkým užívatelia, ktorí bežne pracujú s tisícami riadkov a stĺpcov. S tým súvisí aj novinka v podobe 64bitovej podpory, ktorá má umožniť prácu až s 4GB tabuľkami.“(Budai et al, 2010, s. 89)

MS Excel 2013 je súčasťou balíka Microsoft Office 2013. Táto verzia ako prvá je jednotná pre bežné osobné počítače a zároveň pre smartfóny, tablety a webové prostredie.

Jeho aplikácie vedú ukladať dokumenty na web a sú previazané so sieťovým úložiskom SkyDrive. K vylepšeniu došlo tiež pri zdieľaní listov medzi užívateľmi a u ich súčasnej spolupráci v reálnom čase. Celé prostredie je lepšie optimalizované pre dotykové zariadenia.(Barilla et al, 2013, s. 482)

### 1.5.3 Možnosť využitia makier v Microsoft Excel

Ako sme už spomínali pre celý balík MS Office, a teda aj pre MS Excel existuje možnosť automatizácie opakujúcich činností, čo zabezpečuje šetrenie času. Táto výhoda sa robí cez makrá, ktoré sa následne ukladajú do jazyka VBA, kde ich môžeme ďalej upravovať.

Okrem funkcií môžeme v MS Excel vytvárať aj programy (nepresne nazývané makrá). Podprogram, na rozdiel od funkcie, nevracia žiadnu hodnotu, je to len postupnosť príkazov. Tie sa zapisujú do modulov podobne ako funkcie a pomocou nich je možné robiť zložitejšie operácie s objektami Excelu. (Forstová, 2010, s. 42)

Tvoriť makrá v MS Excel sa dá viacerými spôsobmi. Prvý spôsob je zaznamenanie všetkých krokov potrebných k určitej akcii. Prácu si však môžeme uľahčiť tým, že nahráme niektoré alebo všetky príkazy. Kombináciou týchto dvoch spôsobov je nahranie príkazu a následné dopísanie v editore. Zatiaľ čo prvý spôsob je náročný na vedomosti používateľa, druhý uľahčený spôsob má výhodu aj v tom, že sa ľahšie hľadajú podrobnosti v nápovede. Nevýhodou je, že tieto nahrané makrá v sebe môžu niesť už zadané znaky a vlastnosti, ktoré nemusíme vedieť ovládať. Existujú aj tzv. makrovírusy, čo sú vlastne makrá, ktoré sú zámerne napísané tak, aby poškodili jednotlivé súbory v počítači, do ktorého sú prekopírované, alebo je ich cieľom rozšíriť vírus.

Určitú skupinu príkazov v MS Excel môžeme zaznamenať pomocou makier, ktoré môžeme následne spustiť buď klávesovou skratkou, alebo vyvolaním príslušného makra. Záznam makier je pomerne jednoduchý v porovnaní s programovaním vo VBA, ktoré si už vyžaduje aspoň minimálne programátorské zručnosti. (Jelen, 2008, s. 145)

## 2 MAKROJAZYK V MICROSOFT EXCEL

Existujú situácie, keď nástroje a funkcie aplikácie MS Excel nie sú pre naše potreby vhodné či dostatočné, a preto je dobré, že si ich môžeme prispôbiť alebo rozšíriť. Na toto používame makrá, ktoré automatizujú činnosti MS Excel. Makro slúži na zaznamenanie určitej skupiny príkazov uloženej v module jazyka Visual Basic for Applications. Tento jazyk obsiahnutý v aplikácii MS Excel nepracuje len v spomínanej aplikácii, ale tvorí skupinu Microsoft Visual Basic Programming System, Applications Edition, ktorá je obsiahnutá skoro vo všetkých aplikáciách firmy Microsoft. Pri vyskytnutí problému najskôr preveríme možnosti vstavaných funkcií a v prípade, že tieto nie sú dostatočné, môžeme využiť VBA. VBA teda využijeme najmä keď:

- opakovane robíme sled rovnakých úkonov,
- používame špecifickú analýzu dát a vlastné funkcie,
- potrebujeme previesť užívateľov určitým procesom. (<http://officir.ic.cz/>, 23.11.2013)

### 2.1 Základné údaje o Visual Basic for Applications

Visual Basic for Applications je objektovo orientovaný programovací jazyk vyvinutý v prostredí Visual Basic spoločnosťou Microsoft. Tento jazyk je vstavaný v takmer všetkých aplikáciách MS Office, ale aj v iných. Ako sme už spomenuli, VBA automatizuje opakujúce sa činnosti, ktoré nie je možné spravovať bežnými funkciami aplikácie.

Objektovo orientované programovanie sa snaží efektívne riešiť jeden zo základných problémov štruktúrovaného (procedurálneho) programovania, ktorým bola separácia dátových objektov a funkcií (alebo podprogramov), ktoré sa týmito dátovými objektami manipulovali. (Hanák, 2011, s. 148)

Základom použitia VBA je zvládnuť objektový model, ktorý sa ako jediný mení v jednotlivých aplikáciách MS Office, pretože syntax jazyka ostáva pre všetky aplikácie MS Office rovnaká. Jednotnosť jazyka VBA umožňuje opätovné použitie kódu pre viaceré aplikácie MS Office a umožňuje vytvárať riešenia pre viacero aplikácií MS Office naraz. Programovacie prostredie jazyka VBA je rovnaké pre všetky aplikácie.

## 2.2 Objektový model Microsoft Excel

Objektový model je hierarchia všetkých objektov aplikácie. Napríklad v Exceli je nasledujúca hierarchia objektov:

*Application – Workbook – Worksheet – Range*

Informácie o objektovom modeli aplikácie sú dostupné v editore VBA alebo online. Pri programovaní vo VBA väčšinou nie je nutné vytvárať vlastné objekty, ale stačí len využívať už vytvorené objekty.

MS Excel obsahuje viac ako 100 objektov, ktoré sú zoradené hierarchicky a najvyššia z nich je aplikácia MS Excel, nižšie sú zošit, hárok, oblasť. Z toho vyplýva, že každý objekt buď obsahuje iné objekty, alebo je obsiahnutý v inom objekte. Celkovú objektovú mapu nájdeme v pomocníkovi MS Visual Basic. Kliknutím na daný objekt si zobrazíme všetky informácie ohľadne daného objektu. S objektom teda môžeme v rámci aplikácie rôzne pracovať. Najčastejšie úkony spojené s objektmi sú:

- čítať a meniť vlastnosti objektu,
- prinútiť objekt k vykonaniu akcie aktiváciou metódy, ktorú objekt ovláda,
- napísať program, ktorý sa spustí pri danej udalosti.

Na tieto úkony nám slúžia štyri pomôcky:

1. Vlastnosti – popisujú charakteristiku objektu. Dôležitou vlastnosťou objektu je meno, ktoré slúži k identifikácii. Svoje meno má zošit, list, tlačidlo. Zmeniť názov sa nemusí len vo VBA, stačí zmeniť pravým tlačidlom myši, a teda už nebude napr. List1, ale Môj názov. Ak chceme zmeniť vlastnosť vo VBA, najprv musíme definovať objekt a potom vlastnosť, ktorú chceme zmeniť a jej hodnotu. Dá sa zmeniť – meno, font, veľkosť písma, hodnota atď.
2. Metódy – sú akcie, ktoré môže daný objekt vykonať, napríklad vybrať, vymazať, kopírovať, prepočítať.
3. Udalosti – ak nastane požadovaná udalosť, MS Excel na ňu môže reagovať a spustiť nejaký kód. Udalosti ktoré MS Excel rozpozná – otvorenie zošitu, zavretie zošitu, uloženie zošitu, prepočítanie zošitu, zmena dát v bunke atď. MS Excel teda zareaguje napríklad po otvorení, po prepočítaní, pred tlačením.
4. Kolekcia – je súbor objektov obsiahnutých v inom objekte. Využijeme to napríklad vtedy, ak chceme do zápätia listov pripísať poznámku o autorovi, ktoré sa môže

vykonať aj pomocou cyklov, no ak by sme neskôr pridali ďalší list, na ňom už poznámka o autorovi nebude. Je teda vhodné to urobiť pomocou kolekcie. (<http://office.lasakovi.com/>, 21.12.2013)

## 2.3 Dátové typy, premenné a konštanty v jazyku VBA

Premenná je pomenované miesto v pamäti počítača, kde sa nachádza kúsok dát, s ktorými chceme pracovať. Dáta môžu mať samozrejme rôznu podstatu a podobu, ako je celé číslo, záporný zlomok, iracionálne číslo (napr. druhá odmocnina z dvoch, číslo  $\pi$  či základ prirodzených algoritmov, známy tiež ako Eulerovo číslo  $e$ ), čiastka v domácej mene, zoznam hodnôt, dátum, text, zošit Excelu, graf na liste Excelu, alebo dokonca celá bežiacia aplikácia Microsoft Excel. (Kráľ, 2012, s. 22)

Premenné v sebe môžu ukladať širokú paletu rôznych dátových typov – od jednoduchých logických hodnôt (*True* alebo *False*) až po veľké čísla s dvojitou presnosťou. Hodnotu do premennej priradíme pomocou operátora rovnosti.

Do premennej môžeme priradiť:

- vlastnosť objektu spôsobom *Premenná = Objekt, Vlastnosť*
- objekt spôsobom *Set Premenná = Objekt*

Premenné by sa mali nazývať tak, aby ich názvy vyjadrovali ich účel alebo obsah. VBA obsahuje pravidlá pre názvy premenných:

- v názvoch premenných musí byť prvý znak písmeno a ďalšie môžu byť písmená, číslice a aj niektoré interpunkčné znamienka,
- VBA nerozlišuje veľké a malé písmená, tieto sa používajú len na lepšie pochopenie názvu,
- nie je možné používať medzery alebo bodky, ale pre lepšiu čitateľnosť môžeme použiť podtržník,
- špeciálne znaky pre deklaráciu dátového typu premennej ( $\#$ ,  $\$$ ,  $\%$ ,  $\&$   $@$ ,  $!$ ) nie je možné použiť v názve premennej,
- názov premennej môže mať až 254 znakov,
- VBA obsahuje aj niektoré vyhradené slová, ktoré nemôžeme použiť ako názov premennej. Takýmto slovom je napríklad *Next*, ktoré keď sa pokúsime použiť v názve premennej ohlásí editor chybu. (Walkenbach, 2010, s. 205)

Dátové typy premenných hovoria o tom, akým spôsobom budú dáta uložené v pamäti – ako celé čísla, reálne čísla, reťazec atď. (viď tabuľka 3). Aj keď sa VBA dokáže automaticky starať o dátové typy, nie je to efektívne, pretože to spomaľuje uskutočňovanie príkazov a zväčšuje to množstvo spotrebovanej pamäti. Druhou možnosťou je explicitná deklarácia dátových typov, kde môžeme behom kompilácie kódu robiť prídavné kontroly možných chýb, ktoré sa inak dajú len veľmi ťažko odhaliť.

Konštanty sa delia na vstavané a definované programátorom. Mnoho vlastností a metód má svoje vlastné preddefinované konštanty. Pri objektoch Excelu sa tieto konštanty začínajú písmenami *xl* a pri objektoch VBA sa začínajú *vb*.

Tabuľka 3 Dátové typy v programovacom jazyku VBA

Dátový typ	Rozsah hodnôt	Veľkosť v pamäti
Boolean (logický)	True alebo False	2 byty
Byte	0-255	1 byte
Currency (menový)	Približne ± 9 biliónov	8 bytov
Date (dátum)	1.január 100 až 31.december Čas 0:00:00 až 23:59:59	8 bytov
Decimal (desatinný)	Maximálna hodnota ±79,228,162,514,264,337,593,543,950,335 s desatinou čiarkou na ľubovoľnom čísle	12 bytov
Double (dvojitá presnosť)	-1.79769313486231 až -4.94065645841247E-324 pre záporné čísla, 4.94065645841247E-324 do 1.79769313486232E308 pre kladné čísla	8 bytov
Integer (krátke celé číslo)	-32,768 až 32,767	2 byty
Long (dlhé celé číslo)	-2,147,483,648 až 2,147,483,647	4 byty
Object (objekt)	Uložený ukazovateľ (pointer) na ľubovoľný objekt v pamäti	4 byty
Single (jednoduchá)	-3.402823E38 až -1.401298E-45 pre záporné hodnoty 1.401298E-45 až 3.402823E38 pre kladné hodnoty	4 byty

presnosť)		
String (reťazec „Text“)	Môže obsahovať až 2 31 znakov, môže mať aj konštantnú dĺžku až do približne 64 000 znakov	Rôzna
User Defined (užívateľský)	Môže obsahovať jeden alebo viac rôznych typov	Rôzna
Variant	Môže obsahovať rôzne hodnoty a objekty	Rôzna

Zdroj: Král, 2012, s. 22.

Najlepšie je používať taký dátový typ, ktorý používa čo najmenší počet bajtov, pri ktorom je ešte schopný obsiahnuť všetky dáta, ktoré mu budú priradené.

Medzi najpoužívanejšie typy premenných patria tieto:

### Premenná typu Boolean

Premenná obsahuje buď hodnotu True (Pravda) alebo False (Nepravda). Použijete ju ako podmienku v rozhodovacích príkazoch a ako návratovú hodnotu mnohých užívateľských funkcií (viď obrázok 1).

Obrázok 1 Premenná typu Boolean

```
Public Function KontrolaCisla(CellValue) As Boolean
    If IsNumeric(CellValue) Then
        KontrolaCisla = True
        Call MsgBox("je to cislo")
    Else
        KontrolaCisla = False
        Call MsgBox("nie je cislo")
    End If
End Function
```

Zdroj: Vlastné spracovanie

Funkcia vráti True, ak list zadaného mena v aktívnom zošite existuje.

### Premenná typu Long

Okrem premenných, ktoré ukladajú ukazovateľ (pointer) na objekt, budeme tento typ premennej používať asi najčastejšie. Typické použitie čítač v cykle a uloženie čísla riadku alebo stĺpca (viď obrázok 2).



### Obrázok 2 Premenná typu Long

```
Sub TestLong()  
  
'// Kod vynasobi kazdou druhou bunku ve sloupci A  
'// aktivniho listu dvema, pocinaje druhym radkem  
Sheets("Long").Activate  
  
Dim i As Long 'Citac  
Dim lLastRow As Long 'posledni radek  
  
lLastRow = Cells(Rows.Count, 1).End(xlUp).Row  
  
For i = 2 To lLastRow Step 2  
    Cells(i, "A") = Cells(i, "A") * 2  
Next i  
  
End Sub
```

Zdroj: Král, 2012, s. 24.

Kód vynásobí každú druhú bunku v stĺpci A aktívneho listu dvoma, začínajúc druhým riadkom.

### Premenná typu String

Túto premennú budeme používať pre uloženie reťazcových premenných, čiže textu. Operácie s textom sú jedny z najpomalších vo VBA a treba ich používať len v cykloch (viď obrázok 3).

### Obrázok 3 Premenná typu String

```
Sub TestString2()  
  
'//spojeni slov v jeden text bez cyklu  
Dim y  
Dim strText ' vysledny text  
  
y = Application.Transpose(Sheets("String").Range("A1:A5"))  
strText = Join(y, ",")  
Debug.Print strText  
  
End Sub
```

Zdroj: Král, 2012, s. 24.

Funkcia spojí slová z buniek A1:A5 do jedného reťazca oddeleného čiarkou.

## Premenná typu Double a Currency

S týmito typmi sa pracuje menej, pretože pri programovaní aplikácie Excel sú hodnoty väčšinou vypočítavané vzorcom na liste. Používajú sa však napríklad pri výpočte degresívnej provízie v závislosti na predaji. Double má jednu vlastnosť, ktorú môžeme pochopiť ako výhodu aj nevýhodu zároveň (viď obrázok 4). Výhodou je, že rozsah hodnôt je skutočne obrovský, pretože obsiahne akékoľvek zmysluplné číslo, no na druhej strane, je týchto čísel nekonečne veľa a premenná ma k dispozícii len obmedzený počet bitov. Preto nekonečné množstvo čísel nemôže byť zobrazené s absolútnou presnosťou. Už relatívne bežné čísla môžu viesť k neočakávaným výsledkom.

Obrázok 4 Premenná typu Double

```
Sub TestDouble ()  
  
Dim a1 As Double, a2 As Double, a3 As Double  
  
a1 = 1  
a2 = 0.3451  
a3 = 0.6549  
Debug.Print (a1 - a2 - a3) = 0  
Debug.Print a1 = a2 + a3  
Debug.Print a1 - a2 - a3  
  
End Sub
```

Zdroj: Král, 2012, s. 25.

Pretože  $a1 = a2 + a3$ , na výpise v okne Immediate (Ctrl+G, ak sa nachádzame v VB Editre) by sme čakali nasledujúce údaje:

True

True

0

Výstup však zobrazí čiastočne iné výsledky:

False

True

-5,55111512312578E-17

Prvý výsledok je úplne zle a tretí je síce veľmi blízko nule, ale nula to nie je. Chyba sa takzvané propaguje do výpočtu, pretože jej vplyv sa s nárastom počtu operácií s chybným číslom zväčšuje. Ešte väčším nebezpečenstvom je použitie premennej typu Double ako podmienky pre ukončenie cyklu, pretože nikdy nemusí nadobudnúť presne

očekávané hodnoty. Vo väčšine nefyzikálnych výpočtov je riešením nahradiť typ Double typem Currency.

### **Premenná typu Object**

Premenná tohto typu bude pravdepodobne jedna z najčastejších premenných, s ktorými sa vo VBA kóde pre Excel pracuje. A to z dvoch dôvodov – rýchlosť a čitateľnosť kódu.

Existujú dva základné druhy použitia:

```
Dim oObject As Object
```

```
Dim wdApp as Word.Application
```

```
Dim ws As Worksheet
```

```
Dime rg As Range
```

V prvom prípade dopredu nevieme, ktorého typu objekt bude, a teda, ktorú triedu mu priradiť. Preto premennú deklarujeme s obecnou triedou Object. V ostatných prípadoch poznáme, akej triedy objekt bude a zároveň je to knižnica obsahujúca tento objekt zahrnutá v referenciách zošitu, v ktorom je tento kód zapísaný (VBE=>Tools=>References)

### **Premenná typu Variant**

Variant je flexibilný typ premennej, ktorý môže obsahovať akúkoľvek hodnotu okrem reťazca o pevnej danej dĺžke (fixed-length string), vrátane hodnôt a objektov Empty, Error, Nothing a Null. Aj keď od verzie Excelu 2002 môže byť do premennej typu Variant priradený aj užívateľský typ premennej, toto použitie je veľmi vzácné.

Vo všetkých týchto prípadoch bude premenná var typu Variant:

```
Dim var As Variant
```

```
Dim var, i As Long
```

```
Dim var
```

Pokiaľ nie je v záhlaví modulu uvedené Option Explicit, je možné i deklaráciu vynechať a premennú použiť rovno v kóde. Bude automaticky typu Variant:

```
Var = var + 1
```

Napriek tomu použitie premennej typu Variant môže vyzerat' lákavo, nepoužívame ju okrem prípadov, keď je to nutné. Premenná typu Variant je jedným z dôvodov, prečo VBA nie je „ozajstnými programátormi“ považovaný za plnohodnotný programovací jazyk. Visual Basic sa im zdá byť príliš benevolentný, čo vedie k zvýšenému riziku výskytu ťažko odhaliteľných chýb.

### 2.3.1 Deklarovanie premenných v jazyku VBA

Premenné používané ako vstupné parametre funkcie alebo ako medzivýsledky môžu obsahovať celé alebo desatinné čísla, textové reťazce, dátum alebo hodnoty True a False. Typ premennej sa určí pri prvom dosadení hodnoty alebo je možné určiť ju dopredu pomocou deklarácie premennej. Deklaráciu zapíšeme príkazom:

```
Dim <premenná> As <typ>
```

Kľúčové slovo As: klauzula As je dôležitá najmä preto, že uvádza dátový typ premennej. Každá definovaná premenná je premennou istého dátového typu. Prostredníctvom kľúčového slova As a nasledujúceho názvu požadovaného dátového typu môžeme účinne vymedziť rozsah hodnôt, ktoré bude možné neskôr do vytvorenej premennej uložiť. (Hanák, 2011, s. 59)

Ak zapíšeme klauzulu As a medzeru, editor VBA ponúkne kontextové menu, z ktorého je možné potrebný typ vybrať. Hľadanie urýchľuje a uľahčuje zápis prvého písmena.

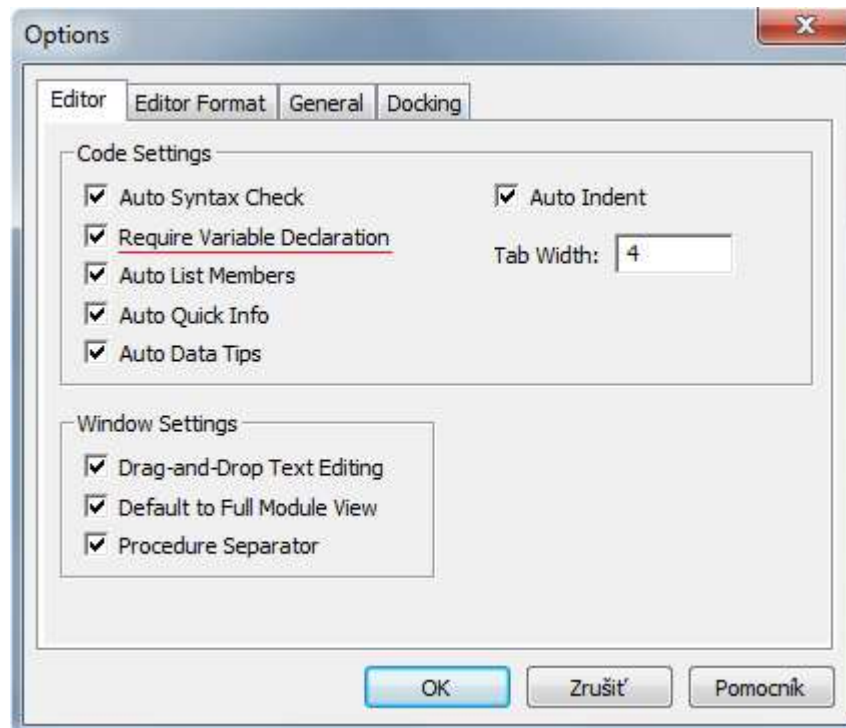
Premenné sa definujú za kľúčovým slovom *Dim*. Tak ako kľúčové slovo *Const*, môžeme aj kľúčové slovo *Dim* nájsť v zozname nápovedy k Microsoft Visual Basic v ponuke *Microsoft Visual Basic Documentation – Visual Basic Language Reference – Statements*.

Ak na začiatok modulu (pred zápis prvého makra alebo funkcie) zapíšeme príkaz Option Explicit, všetky používané premenné musia byť deklarované (viď obrázok 5), inak editor VBA vyhlási chybu. Editor VBA môžeme nastaviť tak, aby sa uvedený príkaz vložil automaticky na začiatok každého modulu:

1. Použijeme príkaz Tools – Option.
2. Klikneme na záložku prvej karty Editor.

3. Označíme voľbu Require Variable Declaration a potvrdíme tlačidlom OK.  
(Laurenčík, 2011, s. 30)

Obrázok 5 Voľby editora VBA



Zdroj: Vlastné spracovanie

Pri deklarácii premenných by sme mali zároveň zadávať aj typ premennej. Ak to neurobíme, použije sa automaticky dátový typ Variant, čo môžeme vidieť takto:

```
Dim varVariable1, strVariable2 As String
```

Na rozdiel od iných programovacích jazykov, ako je napríklad VB.Net, deklaruje sa takto ako dátový typ String iba premenná strVariable2, pri druhej premennej sa nastaví dátový typ Variant – pretože u nej chýba kľúčové slovo As. Správna deklarácia oboch premenných vyzerá takto:

```
Dim varVariable1 As String
```

```
Dim varVariable2 As String
```

Na jednom riadku sa potom deklarácia urobí takto:

```
Dim varVariable1 As String, varVariable2 As String (Breden, 2009, s. 18)
```

Štandardný dátový typ Variant má tú nevýhodu, že VBA musí opakovane robiť časovo náročné kontroly skutočného dátového typu a bude pre premennú rezervovať viac

pamäti, ako je v skutočnosti treba. Používaním príkazu `Option Explicit`, ktorý nás prinúti deklarováť všetky premenné sa vyhneme problémom s preklepmi v názvoch premenných, ktoré vedú k nesprávnym výsledkom procedúry.

### **2.3.2 Rozsah platnosti premenných v jazyku VBA**

Rozsah platnosti premennej určuje, v ktorých moduloch a procedúrach môže byť daná premenná použitá. Možné sú tri druhy rozsahu platnosti:

Lokálne premenné – je to premenná deklarovaná vo vnútri procedúry, len v ktorej môže byť aj používaná. Po skončení tejto procedúry, skončí aj existencia premennej, čo uvoľní zodpovedajúcu časť pamäti v MS Excel. Poznáme možnosť na zachovanie tejto premennej, a to jej deklarováním za statickú pomocou príkazu `Static`. Deklarácia premennej sa robí pridaním príkazu `Dim` medzi príkazy `Sub` a `End Sub`. Rôzne procedúry v rámci jedného modulu môžu používať lokálne premenné s tým istým názvom, no tieto budú v rámci svojej procedúry jedinečné. Tento typ premennej je najefektívnejší z dôvodu uvoľňovania pamäte hneď po skončení procedúry.

Modulová premenná – je taká premenná, ktorá je dostupná pre všetky procedúry v danom module. Túto premennú musíme deklarovať ešte pred prvou procedúrou v module. Modulová premenná sa po ukončení procedúry nezmení, zmení sa len keď je procedúra zastavená príkazom `End`, po ktorom sú stratené hodnoty všetkých modulových premenných.

Verejné premenné – sú premenné, ktoré sú dostupné všetkým procedúram vo všetkých moduloch VBA určitého projektu. Deklarovanie musí byť na úrovni modulu, pomocou kľúčového slova `Public`.

Statické premenné – sú špeciálne premenné, ktoré sú deklarované na úrovni procedúr a uchovávajú si hodnotu po normálnom ukončení procedúry, ale strácajú svoju hodnotu po ukončení procedúry príkazom `End`. (Walkenbach, 2008 s. 210)

## **2.4 Postup pri zaznamenávaní makra v jazyku VBA**

V MS Excel môžeme vytvárať makrá, nazývané aj podprogramy, s ktorými môžeme robiť zložitejšie operácie s objektami MS Excel. Záznamník makier zaznamenáva príkazy priamo v jazyku Visual Basic for Applications, vďaka čomu nemusíme makrá

vytvárať len zápisom ich kódov v editore jazyka VBA. Editor VBA je súčasťou každého zošitu. V MS Excel môžeme vytvárať makrá spôsobmi:

**Užívateľský** – v zošite si otvoríme záznamník makra, ktorý všetky uskutočnené akcie zaznamená pomocou kódu VBA a tento kód uloží ako súčasť zošitu v editore VBA. Makro môže obsahovať len také akcie, ktoré sme schopní urobiť aj bežným spôsobom, no toto šetrí čas práce so zošitom.

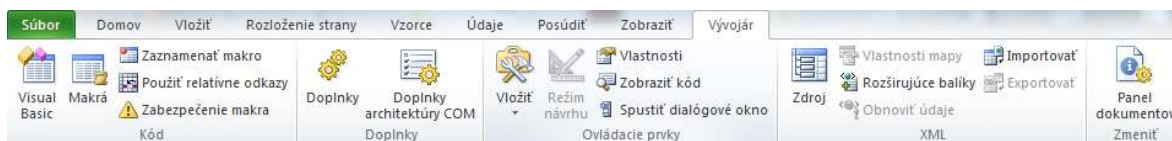
**Programátorský** – zobrazíme si editor VBA, v ktorom potrebný kód zapíšeme ručne. Možnosti pri tomto spôsobe vytvárania makier:

- spracovávať údaje na liste spôsobom, ktorý by bol pri užívateľskej práci časovo náročný alebo nemožný,
- vytvárať nové vlastné funkcie,
- spúšťať automatické akcie pri otvorení a zatvorení zošitu,
- vytvárať vlastné dialógové okná pre zadávanie vstupných údajov.

**Kombinovaný** – makro môžeme zaznamenať v záznamníku makra a upraviť priamym zápisom v editore VBA. (Laurenčík, 2011, s. 13)

Pri práci s makrami si musíme zobrazit' kartu Vývojár (vid' obrázok 6), ktorá je pri MS Excel 2010 automaticky zobrazená už pri prvom spustení programu. V tejto verzii MS Excel budeme programovať našu aplikáciu, ktorá je súčasťou praktickej časti tejto bakalárskej práce.

Obrázok 6 Karta Vývojár v Exceli 2010



Zdroj: Vlastné spracovanie

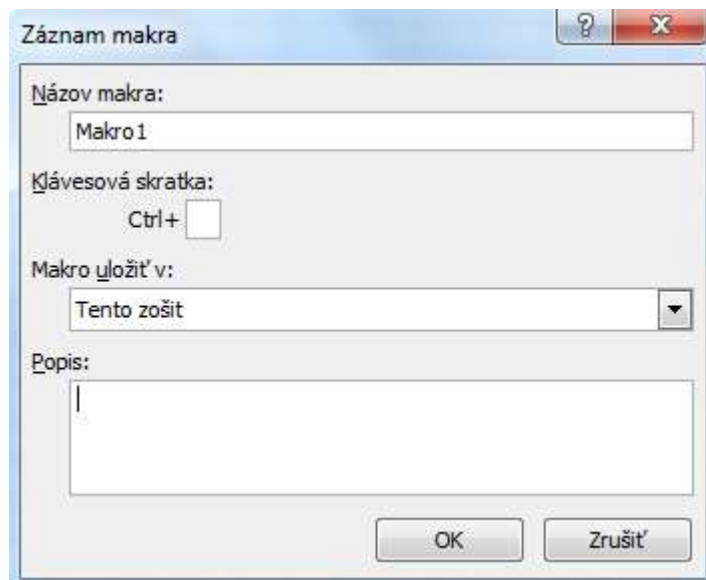
Postup pri zázname makra, ktorý orámuje vybranú oblasť je nasledovný:

1. Označíme oblasť, ktorú budeme chcieť orámovat' (táto oblasť musí mať aspoň dva riadky a dva stĺpce).
2. Príkazom Záznam makra spustíme nahrávanie.

3. Skôr ako sa začne skutočne nahrávať, zobrazí sa dialógové okno (vid' obrázok 7), kde vyplníme informácie o makre. Až po odsúhlasení dialógového okna tlačidlom OK sa začne zaznamenávať.
4. Vykonáme požadované orámovanie štandardným postupom.
5. Príkazom Zastaviť záznam ukončíme nahrávanie.
6. Nahrané makro je možné spustiť kedykoľvek po stlačení príkazu Spustiť z dialógového okna Makrá alebo klávesovou skratkou, ak sme ju zadali.

Výsledkom je orámovanie oblasti tak, že vnútorné čiary budú tenké a vonkajšie čiary budú tučné.

Obrázok 7 Záznam makra

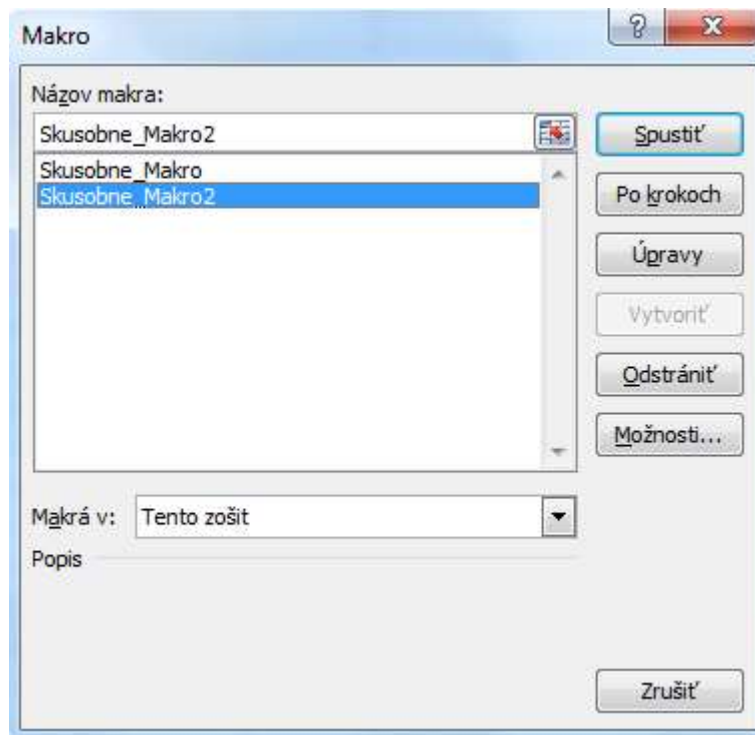


Zdroj: Vlastné spracovanie

Ďalšie akcie spojené s makrom, okrem spomínaného záznamu, sú upraviť makro, vytvoriť makro, spustiť makro a odstrániť makro (vid' obrázok 8). K týmto sa dostanem klávesovou skratkou Alt+F8.



Obrázok 8 Diológové okno makra

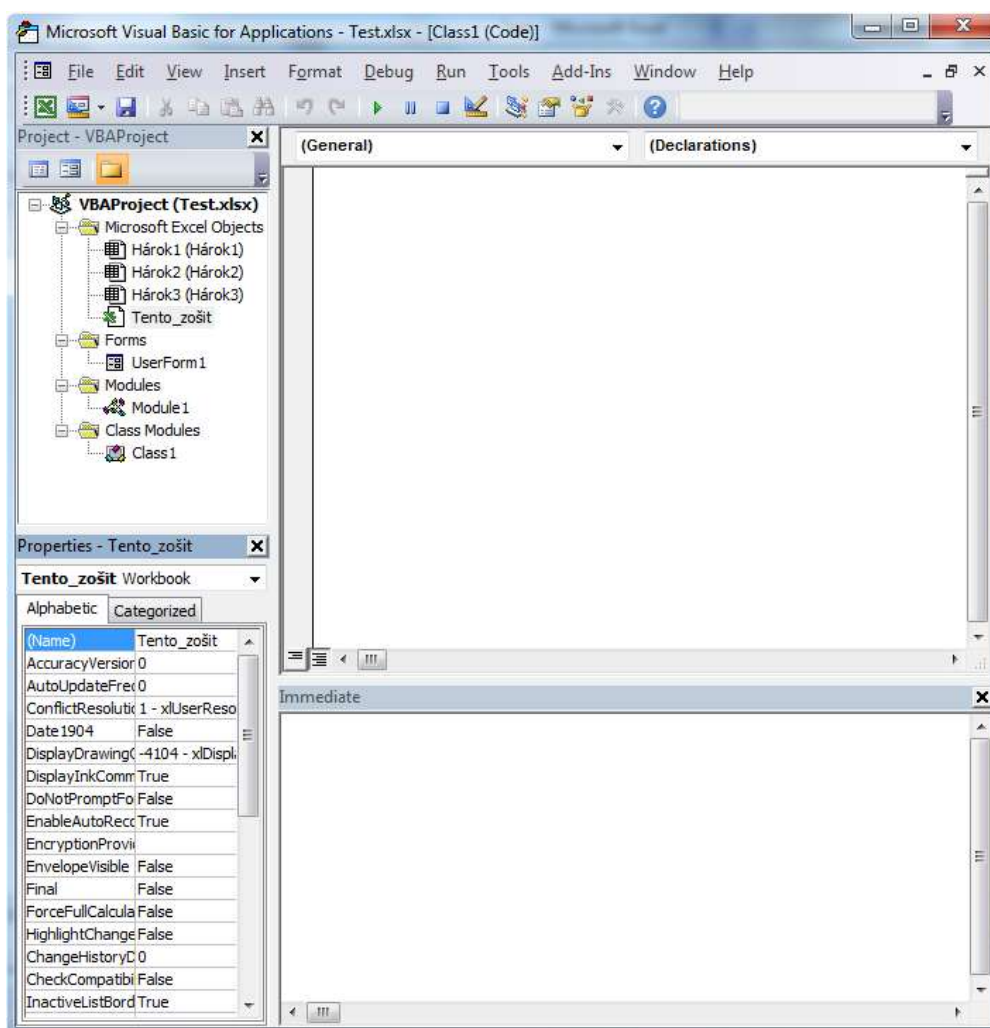


Zdroj: Vlastné spracovanie

## 2.5 Začiatok opisu prostredia

Vývojovým prostredím pre programovací jazyk VBA je editor VBA (viď obrázok 9). Z MS Excel sa do editoru VBA dostaneme buď stlačením klávesovej skratky Alt+F11, alebo cez kartu vývojár, stlačením ikony Visual Basic.

Obrázok 9 Editor programovacieho jazyka VBA



Zdroj: Vlastné spracovanie

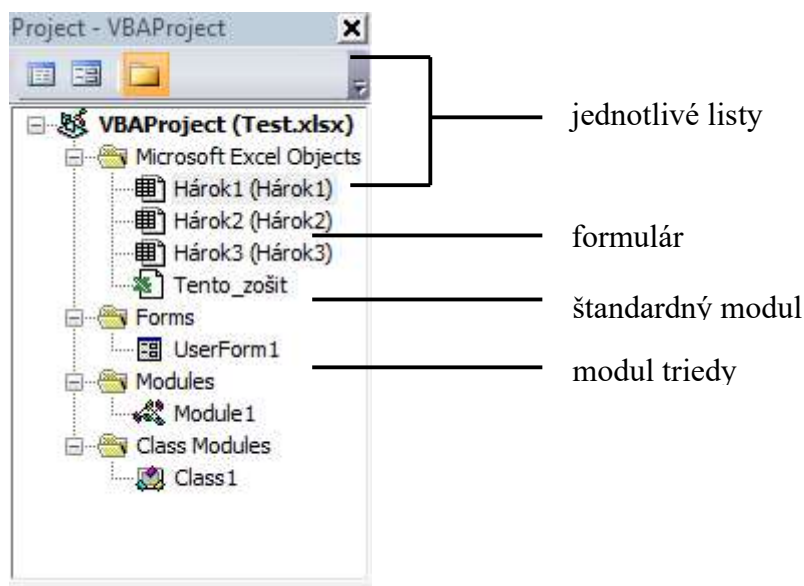
Editor VBA je v angličtine a tomu musíme prispôbiť zápis kódu:

- ako oddeľovač desatinných miest nepoužijeme čiarku, ale bodku,
- ako oddeľovač v zozname nepoužijeme bodkočiarku, ale čiarku.

V okne editor VBA sa nachádzajú tri podokná:

- okno projektu – obsahuje zoznam listov v zošite, zoznam modulov a zoznam užívateľských formulárov (vid' obrázok 10),

Obrázok 10 Okno Projektu



Zdroj: Vlastné spracovanie

- okno properties – využíva sa najmä pri tvorbe formulárov a obsahuje len názov modulu,
- okno kódu – slúži na vlastný zápis kódu. Toto podokno funguje ako poznámkový blok. Jednotlivé príkazy ukončíme klávesou Enter, prázdne riadky sú ignorované a funkčné sú kombinácie Ctrl+C, Ctrl+X a Ctrl+V.

Pri rozdelení príkazu na dva riadky ukončíme prvú časť medzerou a podtržníkom a ďalší riadok je chápaný ako pokračovanie príkazu. Riadky uvedené apostroфом sú komentáre a pri spúšťaní kódu sa ignorujú. Komentáre začínajúce apostroфом je tiež možné zapísať za výkonný príkaz. Editor VBA zobrazuje zapísaný kód pomocou štyroch farieb:

- modrá farba – kľúčové slová jazyku VBA,
- čierna farba – názvy funkcií a premenných,
- zelená farba – komentáre,
- červená farba – chybné riadky.

### 3 Aplikácia – Faktúra

V spomínanom programovacom jazyku VBA sme sa rozhodli naprogramovať aplikáciu na vytvorenie faktúry. Pre túto aplikáciu sme sa rozhodli z dôvodu šetrenia času pri vystavovaní faktúr v menších spoločnostiach, ktoré obchodujú v rámci Slovenska. Ďalší dôvod, prečo sme si pre tento programovací jazyk vybrali práve aplikáciu faktúry je fakt, že tento programovací jazyk využíva prostredie MS Office, ktoré väčšina používateľov už dobre pozná.

Všetky povinné náležitosti a pravidlá pre vyhotovovanie faktúr sú upravené zákonom 222/2004 Z. z. o dani z pridanej hodnoty v znení neskorších predpisov. Harmonizácia predpisov všetkých členských štátov Európskej únie je zabezpečená smernicou Rady 2010/45/EÚ o nových pravidlách fakturácie z 13. júla 2010, ktorou sa mení a dopĺňa smernica 2006/112/ES o spoločnom systéme dane z pridanej hodnoty. Túto smernicu museli prijať všetky členské štáty od 01. 01. 2013. Novou smernicou sa menia paragrafy 71 – 76.

Význam nových pravidiel fakturácie je:

- zlepšiť kontrolu daní zmenením a doplnením náležitostí faktúr,
- zjednotiť zaobchádzanie medzinárodných a vnútroštátnych transakcií,
- harmonizovať lehotu na vyhotovovanie medzinárodných faktúr,
- rozšíriť elektronickú fakturáciu. (<http://financnasprava.sk/>, 21.3.2014)

Podľa § 71 ods.1 zákona č. 222/2004 Z. z. o dani z pridanej hodnoty v znení neskorších predpisov sa faktúrou rozumie každý doklad alebo oznámenie, ktoré je vyhotovené v listinnej alebo elektronickej forme. Elektronickou faktúrou je faktúra, ktorá obsahuje povinné údaje a je vydaná a prijatá v akomkoľvek elektronickej formáte. Elektronickú faktúru možno vydať len so súhlasom príjemcu tovaru alebo služby. Vierohodnosťou pôvodu faktúry sa rozumie potvrdenie totožnosti dodávateľa tovaru alebo služby, alebo osoby, ktorá v mene dodávateľa vyhotovila faktúru.

Podľa § 72 zákona č. 222/2004 Z. z. o dani z pridanej hodnoty v znení neskorších predpisov je faktúru povinná vyhotoviť osoba, ktorá je platiteľom DPH pri:

- a) dodaní tovaru alebo služby s miestom dodania v tuzemsku inej zdaniteľnej osobe alebo právnickej osobe, ktorá nie je zdaniteľnou osobou,
- b) dodaní tovaru alebo služby s miestom dodania v inom členskom štáte, keď je osobou povinnou platiť daň príjemca tovaru alebo služby, a to aj, ak je dodanie tovaru alebo služby oslobodené od dane,
- c) dodaní tovaru alebo služby s miestom dodania v treťom štáte pre zdaniteľnú osobu,
- d) dodaní tovaru formou zásielkového predaja s miestom dodania v tuzemsku,
- e) prijatí platby pred dodaním tovaru podľa písmen a) až d),
- f) prijatí platby pred dodaním služby podľa písmen a) až c).

Podľa § 73 zákona č. 222/2004 Z. z. o dani z pridanej hodnoty v znení neskorších predpisov musí byť faktúra vyhotovená do 15 dní

- odo dňa dodania tovaru alebo služby,
- odo dňa prijatia platby pred dodaním tovaru alebo služby,
- od konca kalendárneho mesiaca, v ktorom bol dodaný tovar oslobodený od dane,
- od konca kalendárneho mesiaca, v ktorom bola dodaná služba alebo prijatá platba pred dodaním služby s miestom dodania v inom členskom štáte,
- od konca kalendárneho mesiaca, v ktorom nastala skutočnosť rozhodná pre vykonanie opravy základu dane.

Podľa § 74 zákona č. 222/2004 Z. z. o dani z pridanej hodnoty v znení neskorších predpisov sú povinnými údajmi faktúry:

- meno a priezvisko zdaniteľnej osoby alebo názov zdaniteľnej osoby, adresu jej sídla, miesta podnikania, prevádzkarne, bydliska alebo adresu miesta, kde sa obvykle zdržiava a jej identifikačné číslo pre daň, pod ktorým tovar alebo službu dodala,
- meno a priezvisko príjemcu tovaru alebo služby alebo názov príjemcu tovaru alebo služby, adresu jeho sídla, miesta podnikania, prevádzkarne, bydliska alebo adresu miesta, kde sa obvykle zdržiava a jeho identifikačné číslo pre daň, pod ktorým mu bol dodaný tovar alebo pod ktorým mu bola dodaná služba,
- poradové číslo faktúry,

- dátum, keď bol tovar alebo služba dodaná, alebo dátum, keď bola platba prijatá, ak tento dátum možno určiť a ak sa odlišuje od dátumu vyhotovenia faktúry,
- dátum vyhotovenia faktúry,
- množstvo a druh dodaného tovaru alebo rozsah a druh dodanej služby,
- základ dane pre každú sadzbu dane, jednotkovú cenu bez dane a zľavy a rabaty, ak nie sú obsiahnuté v jednotkovej cene,
- uplatnenú sadzbu dane alebo oslobodenie od dane, pri oslobodení od dane sa uvedie odkaz na ustanovenie zákona alebo smernice, alebo informácia „dodanie je oslobodené od dane“,
- výšku dane spolu v eurách, ktorá sa má zaplatiť,
- slovnú informáciu „vyhotovenie faktúry odberateľom“, ak faktúru vyhotovuje odberateľ, ktorý je príjemcom tovaru alebo služby,
- slovnú informáciu „prenesenie daňovej povinnosti“, ak osobou povinnou platiť daň je príjemca tovaru alebo služby.

Nami vytvorená aplikácia Faktúra je naprogramovaná v prostredí MS Office Excel s využitím programovacieho jazyka VBA. Pri návrhu zvolenej aplikácie sme sa snažili dodržať súčasný trend v aplikáciách a vyhnúť sa zbytočnému skomplikovaniu používania. Pravidlá, ktorých sme sa držali, aby bola vytvorená prehľadná a ľahko pochopiteľná aplikácia pre každého používateľa so základnými zručnosťami v MS Office Excel:

- vyhnúť sa zbytočnému skrášľovaniu prostredia (príliš mnoho farieb),
- ľahko čitateľné písmo (Arial),
- dostatočná veľkosť písma (aspoň 12b),
- vyhnúť sa zvýrazňovaniu písma (kurzíva, podčiarkovanie),
- systematicky rozdeliť údaje pre zabezpečenie prehľadnosti,
- vyhnúť sa cudzojazyčným názvom tlačidiel (Print/Tlačiť),
- ošetrovanie vstupov (IČO: 12345K78 - chybové hlásenie),
- dodržiavanie názvov premenných (LsText – Lokal String name),
- zamknúť listy, aby sme predošli k nechcenému mazaniu alebo dopisovaniu.

Naša aplikácia je rozdelená na 3 časti (listy):

- Faktúra
- Cenník

- Odberatelia

2 formuláre:

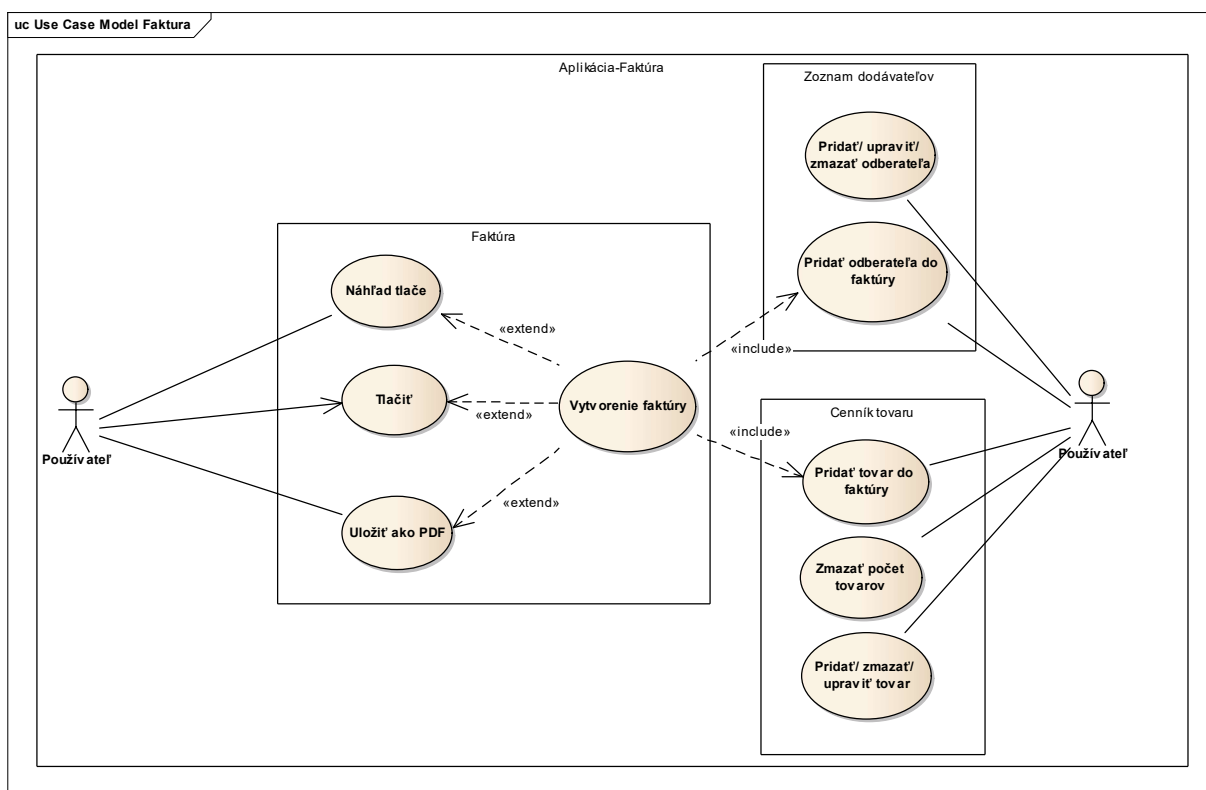
- Odberateľ
- Položka

a štyri moduly:

- Cenník
- Faktúra
- Odberateľ
- Support

Pre lepšie pochopenie používania našej aplikácie sme vytvorili Use Case Diagram.

Obrázok 11 Use Case Diagram aplikácie Faktúra



Zdroj: Vlastné spracovanie

V ďalších častiach práce si podrobne vysvetlíme jednotlivé moduly, ich funkčnosť a časti zdrojového kódu.

### 3.1 List Faktúra

List Faktúra považujeme za podstatu našej aplikácie, pretože cieľom aplikácie je práve výstup z tohto listu. Na tomto liste sa nachádza samotná faktúra so všetkými povinnými náležitosťami podľa zákona 222/2004 Z. z. o dani z pridanej hodnoty v znení neskorších predpisov. Jednotlivé náležitosti na faktúre sú zoradené podľa súčasného trendu tvorby faktúry. To znamená, že všetky informácie o dodávateľovi (názov, adresa, IČO, DIČ, bankové spojenie), odberateľovi (názov, adresa, IČO, DIČ, bankové spojenie), informácie o objednaných tovaroch alebo službách (názov a popis, jednotka, množstvo, jednotková cena bez DPH, jednotková cena s DPH, suma DPH, celková suma), dátumy (vystavenia, dodania, splatnosti) a podpis sú zoradené tak, aby boli čo najviac prehľadné (viď obrázok 12). Tento list sme upravili jednoduchým zlúčením jednotlivých buniek, rôznymi druhmi orámovania, prispôbením veľkosti buniek a rôznymi veľkosťami písma, a tým sme vytvorili šablónu, do ktorej sa budú presúvať zvolené informácie z dvoch ostatných listov (Cenník, Odberatelia).

Obrázok 12 Rozmiestnenie údajov na faktúre



Zdroj: Vlastné spracovanie

Okrem nami vytvorenej šablóny sa na tomto liste nachádzajú aj tri ovládacie prvky AcitveX (CommandButton). Prvý z nich má názov Uložiť ako PDF. Toto tlačidlo slúži na to, aby sme si hotovú faktúru, mohli uložiť vo formáte PDF, pričom názov faktúry sa bude zhodovať s jej číslom. Toto sa vykoná vďaka procedúre “ExportToPDF“. Pri spustení



procedúry, ktorá je nahratá na tomto tlačidle, sa používateľovi zobrazí dialógové okno s možnosťou výberu umiestnenia dokumentu, prípadne zmenu formátu, v ktorom chceme, aby bola daná aplikácia uložená. Pri programovaní procedúry, ktorá nám umožní toto ukladanie, sme využili možnosť nahrávania makra, ktoré sme si po nahratí upravili na svoj požadovaný stav. Úprava bola potrebná, pretože nahraté makro nezobrazovalo možnosť výberu miesta uloženia a formátu, ale napevno bol stanovený formát aj adresa uloženia.

Obrázok 13 Zdrojový kód procedúry ExportToPDF

```
Public Sub ExportToPDF()  
  
Dim LsFileName As String  
Dim fileSaveName As Variant  
  
LsFileName = ActiveSheet.Range("L4").Text  
fileSaveName = Application.GetSaveAsFilename(fileFilter:="PDF Files (*.pdf),_*  
*.pdf", _ InitialFileName:="faktura_" & LsFileName & ".pdf")  
  
If fileSaveName <> False Then  
    ActiveSheet.ExportAsFixedFormat Type:=xlTypePDF, Filename:=fileSaveName, _  
    Quality:=xlQualityStandard, IncludeDocProperties:=True, _  
    IgnorePrintAreas:=False,  
    OpenAfterPublish:=False  
    MsgBox "Save as " & fileSaveName  
End If  
End Sub
```

Zdroj: Vlastné spracovanie

Druhé tlačidlo s názvom Tlačiť (viď obrázok 14) slúži na vytlačenie aktuálneho listu Faktúra. Stlačením tohto tlačidla sa vytlačí len oblasť, ktorú sme si nastavili v MS Excel.

Obrázok 14 Zdrojový kód procedúry Tlacit

```
Public Sub Tlacit()  
    ActiveSheet.PrintOut  
End Sub
```

Zdroj: Vlastné spracovanie

Tretie tlačidlo Náhl'ad Tlače (viď obrázok 15) je podobné tlačidlu Tlačiť. Rozdiel je v tom, že aktuálny list nevytlačí, ale len zobrazí oblasť, ktorá bude vytlačená. Toto tlačidlo slúži na to, aby sme sa uistili, že daný dokument bude vytlačený správne a nedôjde k chybám.

Obrázok 15 Zdrojový kód procedúry NahladTlace

```
Public Sub NahladTlace()  
    ActiveSheet.PrintPreview  
End Sub
```

Zdroj: Vlastné spracovanie

### 3.2 List Cenník

Na liste cenník sa nachádzajú tovary alebo služby, ktoré dodáva dodávateľ a všetky informácie o nich v jednotlivých stĺpcoch (názov, počet, cena, merná jednotka). Okrem tohto sa tu nachádza 5 prvkov ActiveX (CommonButton). Prvý z nich je tlačidlo Zmazať počet (viď obrázok 16). Toto tlačidlo slúži na vymazanie počtu kusov všetkých tovarov/služieb a využíva sa pri tvorbe novej faktúry, aby sa v nej nenachádzali informácie z predošlej faktúry. V danej procedúre "ZmazatKS" je zadefinovaný rozsah stĺpca, ktorý sa ma vymazať (B4:B1000), to znamená, že ak by počet tovarov/služieb bol väčší ako 1000, bolo by potrebné toto nastavenie zmeniť. Okrem toho nám táto aplikácia odomkne aktívny list a potom ho následne zamkne. S týmto postupom sa budeme stretávať aj v iných procedúrach.

Obrázok 16 Zdrojový kód procedúry ZmazatKs

```
Public Sub ZmazatKs()  
    ActiveSheet.Unprotect  
    ActiveSheet.Range("B4:B1000").ClearContents  
    ActiveSheet.Protect  
End Sub
```

Zdroj: Vlastné spracovanie

Druhé tlačidlo Tovar do faktúry prekopíruje všetky tovary/služby (viď obrázok 17), pri ktorých je zadaný počet do listu Faktúra. Táto procedúra, ktorá je vyvolaná stlačením tlačidla Tovar do faktúry má niekoľko krokov. Najskôr vymaže oblasť produktov na liste Faktúra, potom skontroluje stĺpec s názvom Počet, následne sa prekopírujú tovary/služby, pri ktorých je zadaná číselná hodnota do listu Faktúra na príslušné bunky. Pri každom chode tejto procedúry sa upraví dátum na liste Faktúra vďaka funkcii Date a vygeneruje sa unikátne číslo faktúry.

Obrázok 17 Priebeh procedúry ZapisTovaruDoFaktury

```
Sheets (LsFaktSheetName) .Range ("C24:O39") .ClearContents
```



```
For LLI = START_ROW_POLOZKA To 1000  
    If (ActiveSheet.Cells(LLI, 2).Value <> "") And (poc < 17) Then  
        ...Vykoná sa kopírovanie...  
    End If  
Next LLI
```



```
Sheets (LsFaktSheetName) .Range ("O17") = Format (Date, "d.m.yyyy")  
Sheets (LsFaktSheetName) .Range ("O19") = Format (Date + 21, "d.m.yyyy")  
Sheets (LsFaktSheetName) .Range ("O18") = Format (Date, "d.m.yyyy")  
Sheets (LsFaktSheetName) .Range ("L4") = CStr (CLng (Sheets (LsFaktSheetName) _  
.Range ("L4") .Value) + 1)
```

Zdroj: Vlastné spracovanie

Tretie tlačidlo Pridať slúži na pridanie nového tovaru/služby do listu Cenník. Pri stlačení tohto tlačidla sa vyvolá procedúra, ktorá otvorí formulár Pridať položku (viď obrázok 18). V tomto formulári sa nachádzajú dva prvky TextBox a jeden prvok ComboBox.

Obrázok 18 Formular pre pridanie položky

Zdroj: Vlastné spracovanie

V tomto formulári sme ošetrili vstupy, aby bolo vylúčené duplicitné zadanie tovaru/služby. Na toto ošetrenie nám slúži funkcia “KontrolaNaZhodu“ (viď obrázok 19), ktorá je napísaná v module Support a je typu Public, takže ju môžeme využívať vo všetkých moduloch našej aplikácie.“ KontrolaNaZhodu“ je funkcia, pretože vracia hodnotu, čím sa líši od procedúry, ktorá hodnotu nevracia.

Obrázok 19 Zdrojový kód funkcií KontrolaNaZhodu

```
Public Function KontrolaNaZhodu(ByVal l1Od As Long, ByVal  
    l1Do As Long, ByVal l1IndexStlpca As Long, ByVal IsText As_  
String) As Boolean  
  
Dim l1I As Long  
    KontrolaNaZhodu = True  
  
For l1I = l1Od To l1Do  
    If StrComp(ActiveSheet.Cells(l1I, l1IndexStlpca).Text, IsText,_  
vbTextCompare) = 0  
        Then  
            KontrolaNaZhodu = False  
        End If  
    Next l1I
```

Zdroj: Vlastné spracovanie

Cenu sme ošetrili vďaka príkazu IF a správne zadefinovanej podmienky (viď obrázok 20), vďaka ktorej môžeme zapisovať len číslice od 0 po 9, poprípade bodku na oddelenie desatinných čísel. Hodnota z prvku “txtCena“ sa prekopíruje do stĺpca, ktorý má účtovnícky formát čísla, teda číslo bude mať automaticky dve desatinné miesta plus znak eura.

Obrázok 20 Podmienka pre zadávanie ceny položky

```
If ((KeyAscii < 48) Or (KeyAscii > 57)) And (KeyAscii <> 46)_  
    Then  
  
    KeyAscii = 0  
    Call MsgBox("Zadali ste zlu hodnotu," & vbCrLf & "musite zadat_  
celé číslo,"  
    & vbCrLf & "a ako odelovač pouzít bodku (.)", vbExclamation,_  
    "Nespravna cena")  
  
End If
```

Zdroj: Vlastné spracovanie

ComboBox s názvom “cboMernaJednotka“ uľahčuje prácu pri zadávaní mernej jednotky (viď obrázok 21), pretože sme si vopred zadefinovali základné druhy merných

jednotiek (ks, kg, hod, atď.). Ak používateľ nenájde vhodnú preddefinovanú mernú jednotku, môže si ju dopísať do tohto ComboBoxu.

Obrázok 21 Zdrojový kód ComboBoxu

```
Private Sub UserForm_Initialize()  
  
    cboMernaJednotka.Clear  
    Call cboMernaJednotka.AddItem("Kg")  
    Call cboMernaJednotka.AddItem("Ks")  
    Call cboMernaJednotka.AddItem("l")  
    Call cboMernaJednotka.AddItem("ml")  
  
End Sub
```

Zdroj: Vlastné spracovanie

Na štvrtom tlačidle s názvom Upraviť sa nachádza procedúra “UpravitPolozku“ (viď obrázok 22), ktorá zobrazí formulár s vyplnenými údajmi o tovare/službe, na ktoré sa nachádzajú na aktuálnom riadku. V tejto procedúre “UpravitPolozku“ nastavíme premennú “MbJeUprava“ deklarovanú na formulári “frmCennik“ na True, aby sme rozlíšili režim, že ide o úpravu položky. Ak by premenná “MbJeUprava“ mala hodnotu False zobrazil by sa prázdny formulár, ako v prípade tlačidla Pridať.

Obrázok 22 Zdrojový kód procedúry UpravitPolozku

```
Public Sub UpravitPolozku()  
    frmPolozka.MbJeUprava = True  
    frmPolozka.Caption = "Upraviť položku"  
    frmPolozka.Show  
End Sub
```

Zdroj: Vlastné spracovanie

Posledné tlačidlo na liste Cennik je tlačidlo Zmazať. K tomuto tlačidlu je pridaná procedúra “ZmazatPolozku“ (viď obrázok 23), ktorá najskôr zistí polohu aktívneho riadku a porovná ju s konštantou “START\_ROW\_POLOZKA“, ktorá má hodnotu 4, čiže ak sa nachádzame na treťom riadku, procedúra sa nevykoná. Procedúra pokračuje zistením, či sa v danom aktívnom riadku nachádza nejaký znak (funkcia Len) a ak áno, tak sa zobrazí MsgBox s otázkou „Naozaj chcete odstrániť položku ... z cenníka?“. Ak používateľ stlačí áno, vymaže sa celý riadok. Ak stlačí nie, tak sa celá procedúra zruší.

Obrázok 23 Ukážka z kódu procedúry ZmazatPolozku

```
LlRow = ActiveCell.Row
If LlRow < START_ROW_POLOZKA Then Exit Sub

If Len(ActiveSheet.Cells(LlRow, 1).Text) = 0 Then Exit Sub

LsText = "Naozaj chcete odstrániť položku '" _
& ActiveSheet.Cells(LlRow, 1).Text & "' z cenníka?"

If MsgBox(LsText, vbYesNo, "Odstránenie položky") = vbYes Then
    ActiveSheet.Unprotect
    ActiveCell.EntireRow.Delete
    ActiveSheet.Protect
End If
```

Zdroj: Vlastné spracovanie

### 3.3 List Odberatelia

Posledný z listov našej aplikácie je list Odberatelia. Tento list je vlastne zoznamom všetkých súčasných odberateľov daného dodávateľa so všetkými základnými informáciami (názov, adresa, IČO, DIČ). Pri tvorbe novej faktúry sa jedným klikom prenesú informácie o zvolenom odberateľovi na list Faktúra. Informácie o súčasných odberateľoch sa dajú upraviť alebo vymazať, prípadne pridávať nových odberateľov pomocou prvkov ActiveX, ku ktorým sú pridané jednotlivé procedúry. Tieto procedúry sú obdobné ako na liste Cenník.

Prvým tlačidlom je tlačidlo s názvom Pridať firmu do faktúry, ku ktorému je pridaná procedúra "OdberatelDoFaktury" (viď obrázok 24). Táto procedúra prekopíruje informácie o zvolenom odberateľovi z aktívneho riadku do príslušnej oblasti na list Faktúra v prípade, že aktívna oblasť nie je prázdna. Ak je aktívna oblasť prázdna, zobrazí sa chybové hlásenie „Vybraná zlá bunka“.

Obrázok 24 Zdrojový kód procedúry OdberatelDoFaktury

```
Public Sub OdberatelDoFaktury()  
Dim Riadok As Variant  
Dim LsFaktSheetName As String  
Dim LsOdbSheetName As String  
  
LsOdbSheetName = "Odberatelia"  
LsFaktSheetName = "Faktúra"  
  
If ActiveCell = "" Then  
    Call MsgBox("Vybraná zlá bunka," & vbCrLf & "musíte byť na bunke," _  
        & vbCrLf & "ktorá nie je prázdna.", vbExclamation, "Zlá bunka"  
    Exit Sub  
End If  
  
Riadok = ActiveCell.Row  
    Sheets(LsFaktSheetName).Unprotect  
    ...Vykoná prekopírovanie...  
    Sheets(LsFaktSheetName).Select  
    Sheets(LsFaktSheetName).Protect  
End Sub
```

Zdroj: Vlastné spracovanie

Nasledujúce tri tlačidlá Pridať, Upraviť, Zmazať obsahujú obdobné procedúry, ako tlačidlá s rovnakými názvami na liste Cenník, len sa vzťahujú na odberateľov a nie na produkty. Menšie rozdiely nachádzame vo formulári Odberateľ, ktorý je vyvolaný stlačením tlačidiel Pridať a Upraviť. Tento formulár je tvorený piatimi TextBox prvkami, do ktorých používateľ zadáva príslušné údaje (viď obrázok 25).

Obrázok 25 Formulár na pridanie odberateľa



Zdroj: Vlastné spracovanie

Takisto je ošetrený aj tento formulár, konkrétne vstupy na dvoch prvkoch TextBox IČO a DIČ. IČO sme ošetrili (viď obrázok 26), aby sa nenachádzalo jedno IČO dvakrát v zozname odberateľov. V prípade ak by sme chceli zadať rovnaké IČO, ktoré už je v zozname odberateľov, vyhodí chybové hlásenie „Firma s daným IČO už existuje“. Tento TextBox je chránený proti zápisu písmen a zároveň IČO musí obsahovať 6 alebo 8 čísel,

pretože na Slovensku je správne IČO len v tomto formáte. Takisto sa používateľovi zobrazí chybové hlásenie pri nesplnení jednej z týchto podmienok.

Obrázok 26 Kód na ošetrovanie IČO

```
Private Sub txtICO_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

If (KeyAscii < 48) Or (KeyAscii > 57) Then
    KeyAscii = 0
    Call MsgBox("Zadali ste zlu hodnotu," & vbCrLf & "musite zadat cele_
    cislo.", vbExclamation, "Nespravne ICO")
End If
End Sub
```

Zdroj: Vlastné spracovanie

Zadávanie hodnôt do prvku TextBox DIČ je ošetrované (viď obrázok 27), aby nebolo možné zadať nič iné len 10 číslic, pretože len tento formát je na Slovensku správny. Ošetrovanie na zadávanie len číselných hodnôt je podobné ako na predchádzajúcom obrázku preto si ukážeme ošetrovanie na zadanie len desať číslic. To znamená, že používateľ môže zadať len desať číslic.

Obrázok 27 Kód na ošetrovanie počtu čísla v DIČ

```
If (Len(txtDIC.Text) <> 10) Then
    Call MsgBox("DIČ musí obsahovať 10 číslic.",_
    vbExclamation, "Nespravne DIČo")
Exit Sub
End If
```

Zdroj: Vlastné spracovanie



## Záver

V súčasnosti je v oblasti spracovania údajov najdôležitejšie urýchlenie a zautomatizovanie činností. Keďže sa spracovanie údajov vykonáva najčastejšie v jednotlivých aplikáciách balíka Microsoft Office, musí sa toto zdokonalenie týkať práve týchto aplikácií. Zdokonalenie je samozrejme možné aj pomocou programovacích jazykov, no najefektívnejšie, je ostať v prostredí Microsoft Office, ktoré je používateľom dobre známe. Toto prostredie podporuje programovací jazyk Visual Basic for Applications, ktorý teda spája zdokonalenie spracovania údajov použitím programovacieho jazyka a zároveň používateľ využíva známe aplikácie Microsoft Office. Naša práca sa zaoberá rozborom spomínaného programovacieho jazyka Visual Basic for Applications.

Hlavným cieľom práce bolo preskúmať možnosti programovacieho jazyka Visual Basic for Applications v praxi nad MS Excel a vytvorenie konkrétnej aplikácie pre reálne používanie. Tento cieľ bol z veľkej časti splnený prostredníctvom troch kapitol, ktoré tvoria našu prácu.

V prvej kapitole rozoberáme programovací jazyk Visual Basic for Applications, jeho históriu, odvodené programovacie jazyky a edície. Ďalej tu poukazujeme na možnosti využitia tohto programovacieho jazyka v aplikáciách Microsoft Office. Zároveň opisujeme históriu tohto kancelárskeho balíka až po súčasnosť a tiež približujeme hlavné rozdiely v jeho najnovšej verzii. Prvá kapitola obsahuje aj zoznam výhod a nevýhod použitia programovacieho jazyka Visual Basic for Applications. Z aplikácii Microsoft Office sme si pre tvorbu aplikácie vybrali Microsoft Excel, a tak tu vysvetľujeme aj možnosti využitia programovacieho jazyka Visual Basic for Applications v tejto aplikácii.

Druhá kapitola analyzuje nami vybraný programovací jazyk vo vybranej aplikácii Microsoft Excel. Charakterizujeme tu jeho hlavné časti ako objektový model, dátové typy, premenné či konštanty. Rozoberáme tu aj možnosti zaznamenávania makier v tomto programovacom jazyku a tiež tu už opisujeme jeho programovacie prostredie.

Tretiu kapitolu tvorí konkrétna aplikácia pre reálne používania a to Faktúra. Táto aplikácia je naprogramovaná v Microsoft Excel a tvoria ju tri listy, ktoré popisujeme práve v tejto kapitole. Podrobne vysvetľujeme funkcionality a opisujeme význam jednotlivých tlačidiel.

Domnievam sa, že nami vytvorená aplikácia poslúži na urýchlenie a zautomatizovanie činností. Do budúcnosti by bolo možné zlepšiť túto aplikáciu prepojením s Microsoft SQL, čím by sa rozšírili možnosti pre databázy.

# Zoznam použitej literatúry

## Knižné zdroje

1. **BARILLA, J., SIMR, P., SÝKOROVÁ, K. 2013.** *Excel 2013 Podrobná uživatelská príručka*. Brno : Computer Press, 2013. ISBN 978-80-251-4114-4.
2. **BREDEN, M., SCHWIMMER, M. 2009.** *Excel 2007 VBA*. Brno : Computer Press, a.s., 2009. ISBN: 978-80-251-2698-1.
3. **BUDAI, D., BROŽA, P., POLZER, J. 2010.** *Bible Microsoft Office 2010*. Brno : Extra Publishing, s.r.o., 2010. ISBN 978-80-7413-118-9.
4. **FORSTOVÁ, L. 2010.** *VBA Excel v príkladech* . Kralice na Hané : Computer Media s.r.o., 2010. ISBN 978-80-7402-042-1.
5. **GÁLA, L., POUR, J., ŠEDIVÁ, Z. 2009.** *Podniková informatika*. Praha : Grada Publishing, 2009. ISBN 978-80-247-2615-1.
6. **HALVORSON, M. 2008.** *Microsoft Visual Basic 2008 Krok za krokom*. Brno : Computer Press, a.s., 2008. ISBN 978-80-251-2221-1.
7. **HANÁK, J. 2011.** *Programování v jazyce Visual Basic 2010*. Kralice na Hané : Computer Media s.r.o., 2011. ISBN 978-80-7402-112-1.
8. **JELLEN, B. 2008.** *Excel 2007 Miracles Made Easy*. Ohio : Holy Macro! Books Inc., 2008. ISBN: 978-1-932802-25-2.
9. **KRÁL, M. 2012.** *Excel VBA - Výukový kurz*. Brno : Computer Press, 2012. ISBN 978-80-251-2358-4.
10. **LAURENČÍK, M. 2011.** *Programování v Excelu 2007 & 2010*. Praha : Grada Publishing 2011, a.s., 2011. ISBN 978-80-247-3448-4.
11. **PECINOVSKÝ, J., PECINOVSKÝ, R. 2011.** *Office 2010 - podrobný průvodce*. Praha : Grada Publishing, a.s., 2011. ISBN 978-80-247-3620-4.
12. **WALKENBACH, J. 2010.** *Excel 2010 Power Programming with VBA*. Indiana : Wiley Publishing, Inc, 2010. ISBN 978-0-470-47535-5.
13. **WALKENBACH, John. 2008.** *Microsoft Office Excel 2007. Programování ve VBA*. Brno : Computer Press, 2008. s. 912. ISBN 978-80-251-2011-8.
14. **WEVERKA, P. 2013.** *Microsoft Office 2013 All in One*. New Jersey : Wiley, 2013. ISBN 978-1-118-51636-2.

## Internetové zdroje

1. **DOTNETPORTAL**.2013.Úvod, vývojové prostredí a základní pojmy.dotnetportal.cz [Online]. 2013 [cit. 2013-11-10]. Dostupné na internete: <<http://www.dotnetportal.cz/clanek/2/Uvod-vyvojove-prostredi-a-zakladni-pojmy>>
2. **FINANČNÁ SPRÁVA**. 2014. *Vyhotovovanie faktúr*. [Online]. 2014 [cit. 2014-03-21]. Dostupné na internete: <<https://www.financnasprava.sk/sk/podnikatelia/dane/dan-z-pridanej-hodnoty/vyhotovovanie-faktur>>
3. **ITNEWS**.2010. *LOTUS SCRIPT: Základy*.itnews.sk [Online]. 2010 [cit. 2013-12-13]. Dostupné na internete: <<http://www.itnews.sk/tituly/infoware/2010-04-24/c140640-lotus-script-zaklady-1.-cast>>
4. **LUPA**. 2012. *Microsoft office od roku 1989 do éry smartphonu a cloudu*. Lupa.cz [Online]. 2012 [cit. 2013-12-11]. Dostupné na internete: <<http://www.lupa.cz/clanky/microsoft-office-od-roku-1989-do-ery-smartphonu-tabletu-a-cloudu/>>
5. **MICROSOFT**. 2012. *Windows script host Version Information*. Microsoft.com [Online]. 2012 [cit. 2013-12-21]. Dostupné na internete: <<http://msdn.microsoft.com/cs-cz/library/x66z77t4%28v=vs.85%29.aspx>>
6. **MICROSOFT**. 2013. *Visual Basic Editions*. Microsoft.com [Online]. 2013 [cit. 2013-12-13]. Dostupné na internete: <<http://msdn.microsoft.com/en-us/library/aa232760%28v=vs.60%29.aspx>>
7. **OFFICE**. 2012. *Produkty Office*. Office.com [Online]. 2012 [cit. 2013-12-21]. Dostupné na internete: <<http://office.microsoft.com/sk-sk/business/>>
8. **OFFICE**. 2012. *Porovnanie multilicenčných balíkov Microsoft Office*. Office.com [Online]. 2012 [cit. 2013-11-23]. Dostupné na internete: <<http://office.microsoft.com/sk-sk/business/porovnanie-multilicencnych-balikov-microsoft-office-FX101812899.aspx>>
9. **WINWRAP**. 2013. *VBA and VB.NET*. winwrap.com [Online]. 2013 [cit. 2013-12-10]. Dostupné na internete: <<http://www.winwrap.com/web/basic/default.asp> >

10. **ZIVE.** 2012. *Krok za krokom Visual Basicom.* Zive.sk [Online]. 2012 [cit. 2013-11-25]. Dostupné na internete:  
<<http://www.zive.sk/krok-za-krokom-visual-basicom--cast-i/sc-3-a-253370/default.aspx>>
11. **ZIVE.** 2012. *Prvý pohľad na Microsoft Office 2013.* Zive.sk [Online]. 2012 [cit. 2013-12-15]. Dostupné na internete:  
<<http://www.zive.sk/prvy-pohlad-na-microsoft-office-2013/sc-3-a-302542/default.aspx?textart=1>>

## Zoznam obrázkov

Obrázok 1 Premenná typu Boolean.....	32
Obrázok 2 Premenná typu Long.....	33
Obrázok 3 Premenná typu String .....	33
Obrázok 4 Premenná typu Double.....	34
Obrázok 5 Voľby editora VBA .....	37
Obrázok 6 Karta Vývojár v Exceli 2010 .....	39
Obrázok 7 Záznam makra.....	40
Obrázok 8 Diológové okno makra .....	41
Obrázok 9 Editor programovacieho jazyka VBA.....	42
Obrázok 10 Okno Projektu.....	43
Obrázok 11 Use Case Diagram aplikácie Faktúra.....	47
Obrázok 12 Rozmiestnenie údajov na faktúre.....	48
Obrázok 13 Zdrojový kód procedúry ExportToPDF.....	49
Obrázok 14 Zdrojový kód procedúry Tlacit .....	49
Obrázok 15 Zdrojový kód procedúry NahladTlace.....	50
Obrázok 16 Zdrojový kód procedúry ZmazatKs.....	50
Obrázok 17 Priebeh procedúry ZapisTovaruDoFaktury .....	51
Obrázok 18 Formular pre pridanie položky .....	51
Obrázok 19 Zdrojový kód funkcií KontrolaNaZhodu.....	52
Obrázok 20 Podmienka pre zadávanie ceny položky.....	52
Obrázok 21 Zdrojový kód ComboBoxu.....	53
Obrázok 22 Zdrojový kód procedúry UpravitPlozku.....	53
Obrázok 23 Ukážka z kódu procedúry ZmazatPolozku .....	54
Obrázok 24 Zdrojový kód procedúry OdberatelDoFaktury .....	55
Obrázok 25 Formulár na pridanie odberateľa .....	55
Obrázok 26 Kód na ošetrenie IČO .....	56
Obrázok 27 Kód na ošetrenie počtu čísla v DIČ .....	56

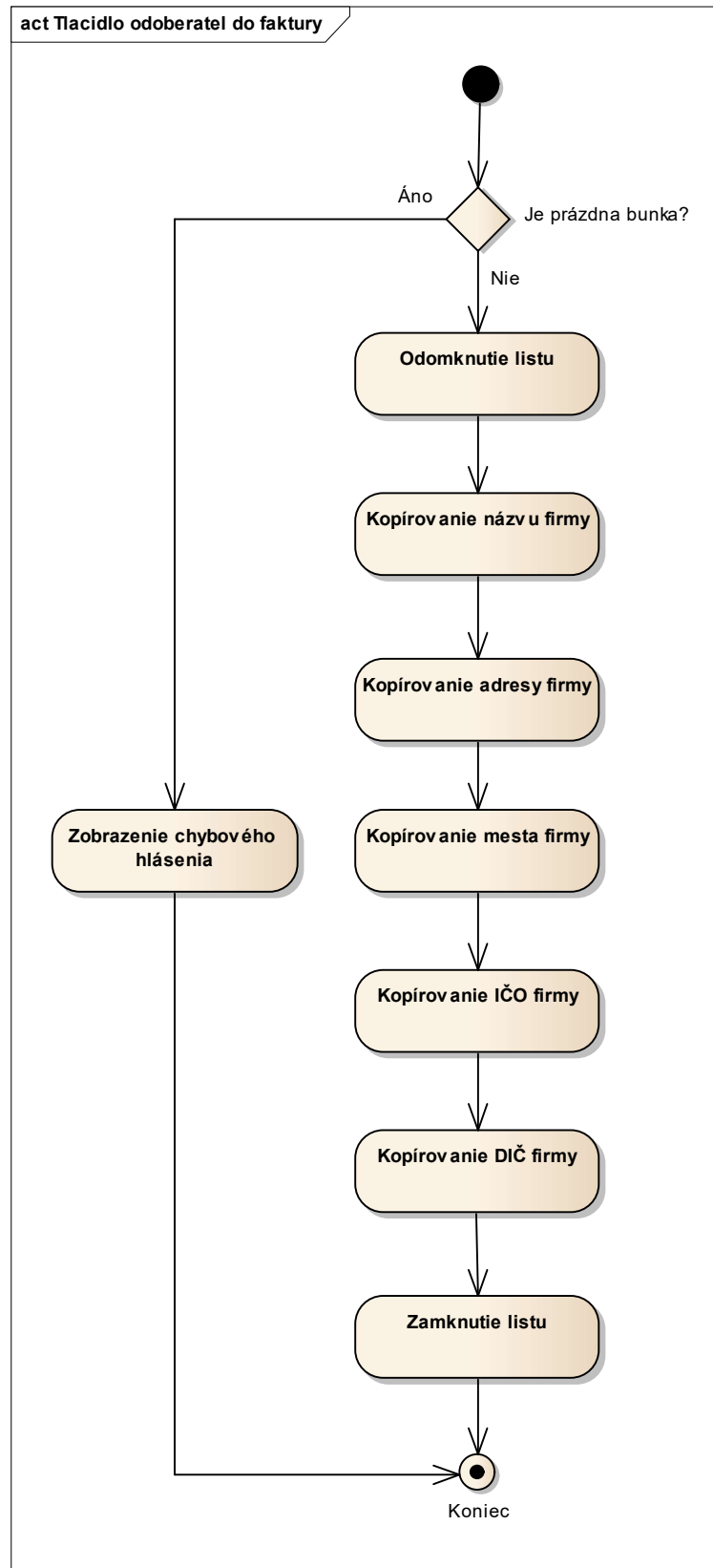
## **Zoznam tabuliek**

Tabuľka 1 Porovnanie cien kancelárskeho balíka Microsoft Office 2013 .....	21
Tabuľka 2 Výhody a nevýhody programovacieho jazyka VBA .....	23
Tabuľka 3 Dátové typy v programovacom jazyku VBA.....	31

## **Zoznam príloh**

Príloha 1 Aktivita diagram pre tlačidlo odberateľ do faktúry .....	64
Príloha 2 Aktivita diagram pre tlačidlo pridať odberateľa .....	65
Príloha 3 Aktivita diagram pre tlačidlo pridať položku .....	66
Príloha 4 Aktivita diagram pre tlačidlo tovar do faktúry .....	67

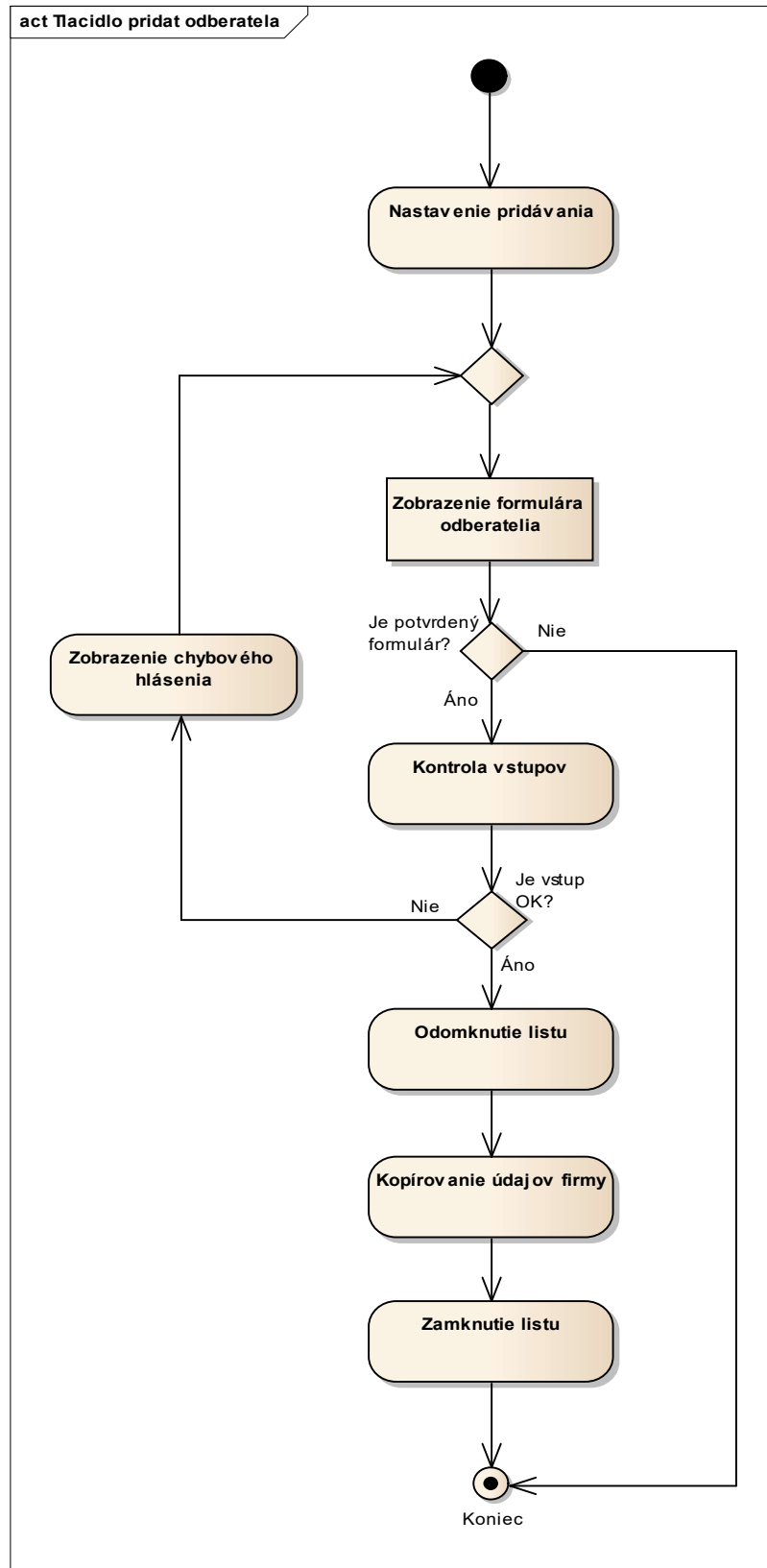
## Aktivity diagram pre tlačidlo odberateľ do faktúry



Zdroj: Vlastné spracovanie

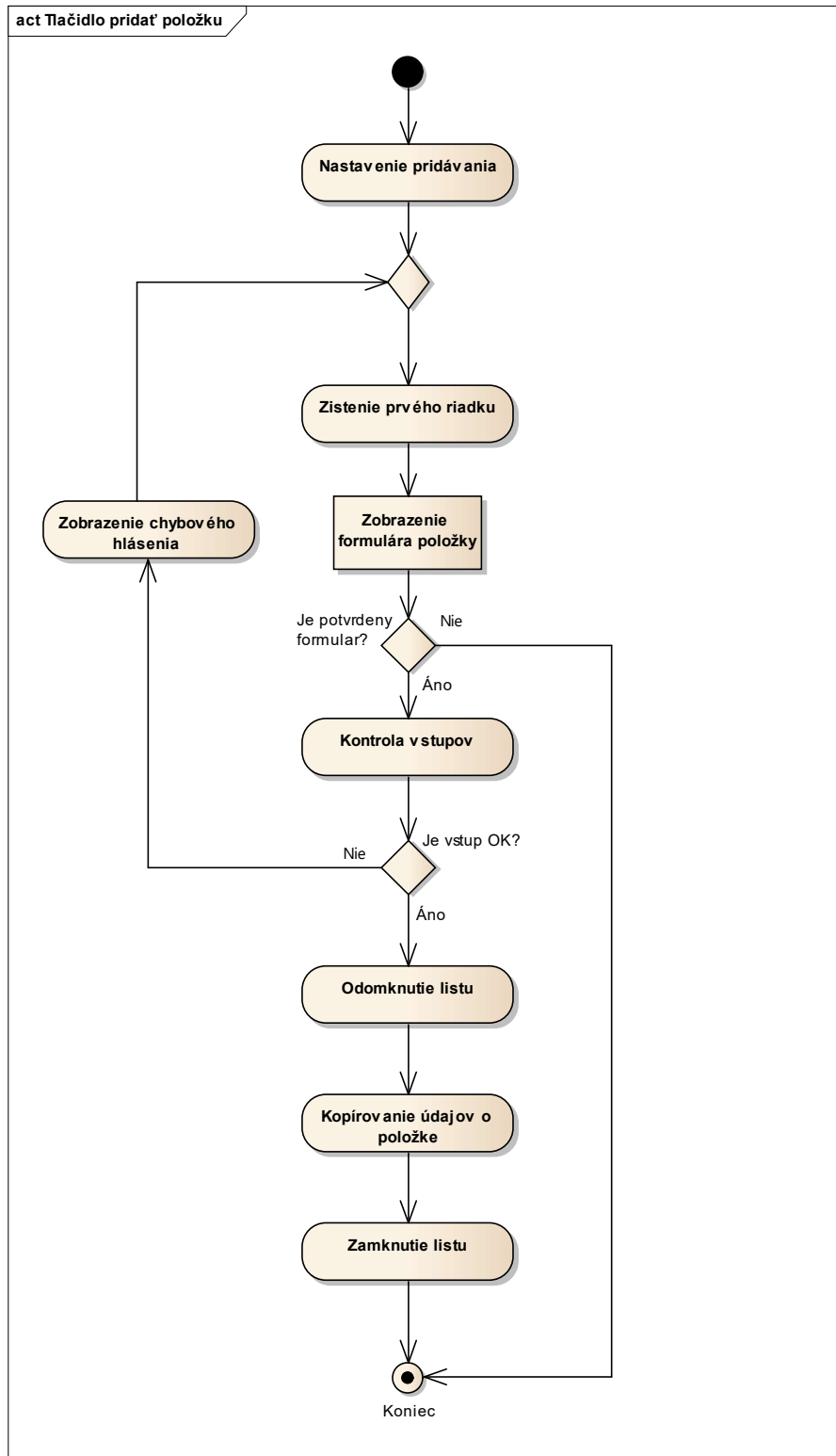


## Aktivity diagram pre tlačidlo pridať odberateľa



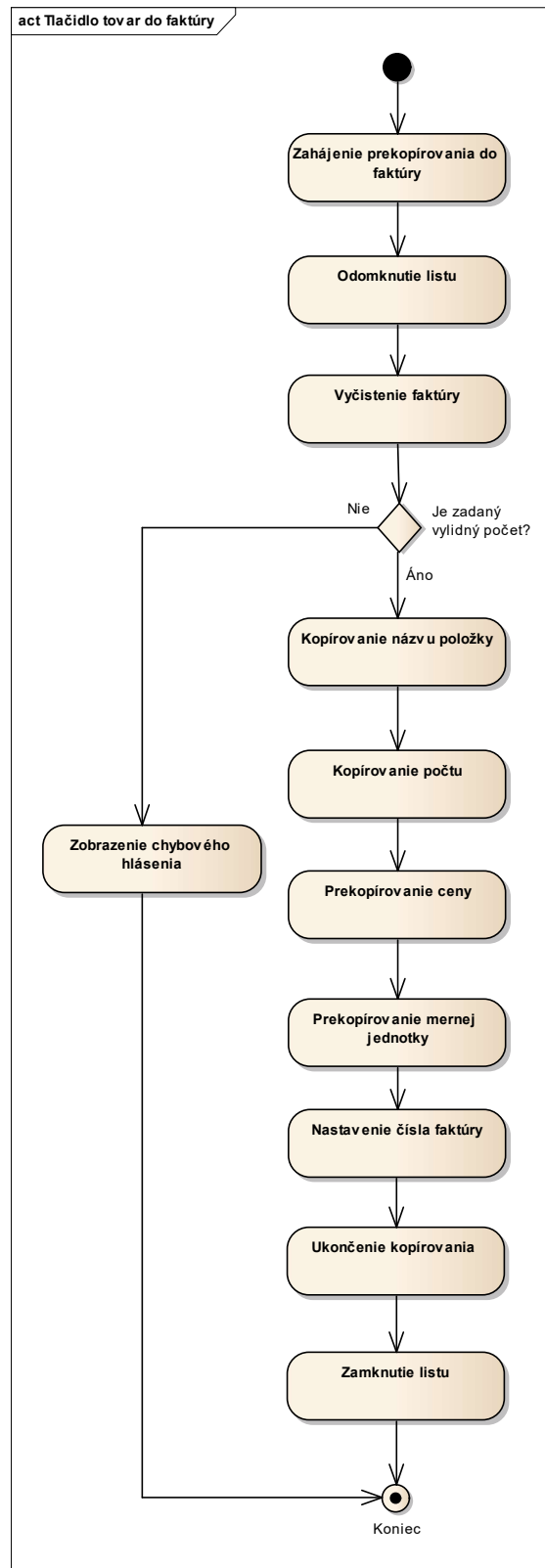
Zdroj: Vlastné spracovanie

## Aktivity diagram pre tlačidlo pridať položku



Zdroj: Vlastné spracovanie

## Aktivita diagram pre tlačidlo tovar do faktúry



Zdroj: Vlastné spracovanie