

# **Vysoká škola ekonomická v Praze**

## **Fakulta informatiky a statistiky**

**Katedra Informačných technológií**

Študent : **Richard Nagrant**  
Vedúci bakalárskej práce : **Ing. Helena Benáčanová, CSc.**  
Oponent bakalárskej práce : **Ing. Miroslav Lorenc**

### **TÉMA BAKALÁRSKEJ PRÁCE**

## **Návrh vhodného užívateľského rozhrania pri tvorbe aplikácií vo VBA**

**Rok 2011**

**Prehlásenie**

Prehlasujem, že som bakalársku prácu spracoval samostatne, že som uviedol všetky použité pramene a literatúru, z ktorej som čerpal.

V Prahe dňa 9.5.2011

.....

podpis

### **Pod'akovanie**

Touto cestou by som sa rád poďakoval vedúcej mojej bakalárskej práce, Ing. Helene Benáčanovej, CSc., za veľmi cenné rady, pripomienky a hlavne návrhy k zlepšeniu práce.

## **Abstrakt**

Moja bakalárska práca je venovaná problematike používania ovládacích prvkov pomocou jazyka VBA a makier v Exceli. Jej cieľom je podať užívateľom detailný pohľad na tvorbu užívateľského rozhrania s následnou aplikáciou ovládacích prvkov.

V úvode sú uvedené rozdiely verzií MS Excelu 2003 a 2007, k čomu sú pripojené aj výsledky prieskumu. V teoretickej časti sú popísané najpoužívanejšie ovládacie prvky, ich najdôležitejšie vlastnosti a taktiež je uvedený zrozumiteľný príklad.

V praktickej časti som aplikoval využitie jednotlivých ovládacích prvkov v užívateľskom rozhraní, taktiež som kládol dôraz na interakciu ovládacích prvkov a hárkov v zošite. Výsledkom je aplikácia pre dynamickú správu záznamov, z ktorej je možné robiť výstupy v podobe kontingenčných tabuliek alebo exportu do textových súborov.

Táto práca môže slúžiť ako návod pre užívateľov, ktorí nemajú s tvorbou užívateľských rozhraní pomocou VBA skúsenosti.

## **Kľúčové slova**

Excel, makro, VBA, OP, zošit, hárok

## **Abstract**

My bachelor thesis is dedicated to the issue through the use of controls using VBA language and macros in Excel. The goal is to bring users a detailed look at creating user interface followed by application of controls.

The introductory part shows differences between versions of MS Excel 2003 and 2007, to which are attached survey results. The theoretical part describes the most common controls, its most important features and also is given clear example.

In practical part I applied the use of controls in user interface, I also put emphasis on the interaction of controls and worksheets in workbook. The result is an application for dynamic records management, which can be used to make outputs in the form of either pivot tables or text file exports.

This thesis may be used as a guide for users, who don't have experiences with creating of user interfaces using VBA

## **Keywords**

Excel, macro, VBA, control, workbook, sheet

## Obsah

Úvod .....	8
Vymedzenie témy práce .....	8
Cieľ práce .....	8
Štruktúra práce.....	8
2    Softvér .....	9
2.1    Excel 2003 vs. Excel 2007.....	9
2.2    Štatistika .....	11
3    Teoretická časť.....	13
3.1    Predstavenie VBA .....	13
3.1.1    Makrá .....	13
3.1.2    Visual Basic Editor .....	14
3.1.3    Premenné.....	15
3.1.4    Procedúry .....	17
3.1.5    Udalostné procedúry .....	18
3.1.6    Funkcie (Funkčné procedúry).....	18
3.2    Užívateľské rozhranie.....	21
3.2.1    Ovládacie prvky formulárov.....	22
3.2.2    Ovládacie prvky ActiveX.....	23
3.2.3    Konvencia názvov OP.....	24
3.2.4    Práca s oknami.....	25
3.2.4.1    MsgBox a InputBox .....	25
3.2.4.2    Dialógové okno 5.0 .....	28
3.2.4.3    Userform.....	28
3.2.5    Label .....	29
3.2.6    Command button .....	31
3.2.7    TextBox.....	32
3.2.8    Frame .....	33
3.2.9    ListBox.....	34
3.2.10    ComboBox .....	37
3.2.11    OptionButton a CheckBox .....	38
3.2.12    Image.....	40
3.2.13    Spin button .....	40
3.2.14    ScrollBar .....	41
3.2.15    MultiPage a TabStrip .....	41

3.2.16	ToggleButton.....	42
3.2.17	RefEdit .....	43
3.2.18	Calendar .....	43
4	Praktická časť .....	45
4.1	Popis .....	45
4.2	Návrh.....	45
4.2.1	Úvodný formulár (frmStart) .....	46
4.2.2	Informačný formulár (frmInfo).....	47
4.2.3	Formulár prehľad (frmPrehľad) .....	47
4.2.4	Prehľad záznamu (frmPrehliadatTelefony) .....	48
4.2.5	Pridávanie OS a značiek (frmOperacneSystemy a frmZnacky) .....	48
4.3	Realizácia.....	48
4.3.1	Spustenie aplikácie .....	48
4.3.2	Formulár frmStart .....	49
4.3.3	Formulár frmPrehľad.....	50
4.3.4	Formulár frmPrehliadatTelefony .....	51
4.3.5	Formulár frmOperacneSystemy.....	52
4.3.6	Formulár frmZnacky.....	52
4.3.7	Formulár frmVyberVlastnostiZariadeni.....	52
4.3.8	Modul funkcie .....	52
4.3.9	Modul KT.....	53
	Záver .....	55
	Zdroje.....	56
	Zoznam použitých zdrojov .....	56
	Zoznam knižnej literatúry .....	57
	Zoznam elektronických zdrojov .....	57
	Zoznam obrázkov .....	57
	Zoznam tabuliek .....	58
	Zoznam grafov.....	59
	Zoznam použitých skratiek.....	59
	Príloha .....	60
	Zdrojový kód .....	60

## Úvod

V dnešnom svete je jednou z kľúčových vlastností úspechu podniku efektivita práce jeho zamestnancov. Tí sú väčšinou založení na každodennej práci s tabuľkami v Microsoft Excel. Ten je dnes asi najpoužívanější nástroj pre kalkuláciu a výmenu dát. Dôkazom toho je štatistický údaj používania ERP systémov v ČR, ktorý je na úrovni 13,9%. [1]

## Vymedzenie témy práce

Ako už z názvu plynie, táto práca je zameraná na vytváranie užívateľských rozhraní pomocou makier a zabudovaného jazyka VBA v Exceli. To znamená, že sa nebudem zaoberať zložitými výpočtami alebo zacyklenými príkazmi, ale zameriam sa hlavne na správne používanie ovládacích prvkov, pomocou ich typických vlastností. Z ovládacích prvkov vyberiem hlavne ovládacie prvky typu ActiveX, ktorých použitie je omnoho rozsiahlejšie, ako pri type formulára.

## Cieľ práce

Cieľom tejto bakalárskej práce je uľahčiť prácu radovým zamestnancom veľkých spoločností, ktorí dennodenne pracujú v hárkoch Excelu. Títo ľudia často nedisponujú schopnosťami pre automatizáciu ich práce. Touto prácou by som im chcel s ich situáciou pomôcť, poskytnúť prehľad o možnostiach používania ovládacích prvkov, čo sa týka ich vlastností a udalostí ktoré môže byť na nich vykonané. Taktiež na jednotlivých príkladoch demonštrovať ich efektívne využitie pri vytváraní užívateľských rozhraní, ktoré zavŕšim vytvorením typického príkladu.

## Štruktúra práce

Táto práca je rozdelená do dvoch väčších častí, do teoretickej a praktickej. V úvode sa popíšu rozdiely medzi jednotlivými verziami na čo nadväzuje vlastný prieskum preferencií užívateľov

V teoretickej časti sa popíšu najčastejšie využívané ovládacie prvky, ku ktorým budú uvedené krátke príklady so zdrojovým kódom a náhľadom. Jednotlivé prvky na seba nadväzujú, v príkladoch sa neuvádzajú OP, ktoré dosiaľ neboli popísané.

V praktickej časti sa ukáže aplikácia jednotlivých ovládacích prvkov pri interakcii s hárkami zošita. Vytvorí sa aplikácia pre dynamickú správu záznamov o mobilných zariadeniach a zároveň sa budú vykonávať rozličné výstupy založené na určitých vlastnostiach zariadení definované užívateľom. Taktiež budú prebiehať rozsiahle kontroly dát pri ich správe.



## 2 Softvér

Pre túto prácu bol vybraný jazyk Visual Basic for Applications (ďalej len VBA) vo verzii Microsoft Office 2007. Hlavným dôvodom tohto výberu sú vlastné skúsenosti s prácou v tomto prostredí a fakt, že sa jedná o spoločné programovacie prostredie všetkých modulov balíka MS Office, z ktorých bol pre túto prácu vybraný Excel.

### 2.1 Excel 2003 vs. Excel 2007

Tak ako každý softvér na svete prebieha svojim vývojom, zlepšeniami a rôznymi úpravami pre potreby užívateľov, tak je tomu aj u excelovského modulu Microsoft Office.

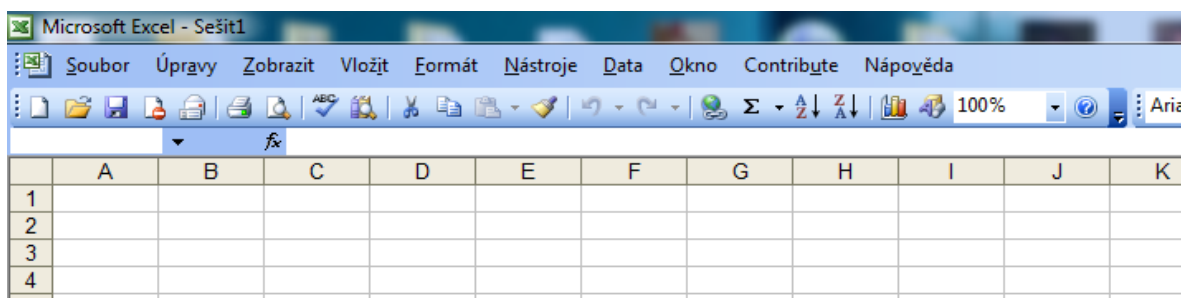
V tejto tabuľke sú rozdiely medzi jednotlivými verziami Microsoft Excel, čo sa týka technického charakteru.

	Excel 2003	Excel 2007
Počet riadkov	65 536 ( $2^{16}$ )	1 048 576 ( $2^{20}$ )
Počet stĺpcov	256 ( $2^8$ )	16 384 ( $2^{14}$ )
Počet listov	Dostupná pamäť	Dostupná pamäť
Dĺžka bunky	32 767	32 767
Rozhranie	Menu a tlačidlá	Ribbony s tlačidlami
Správa pamäte	1 GB	2 GB

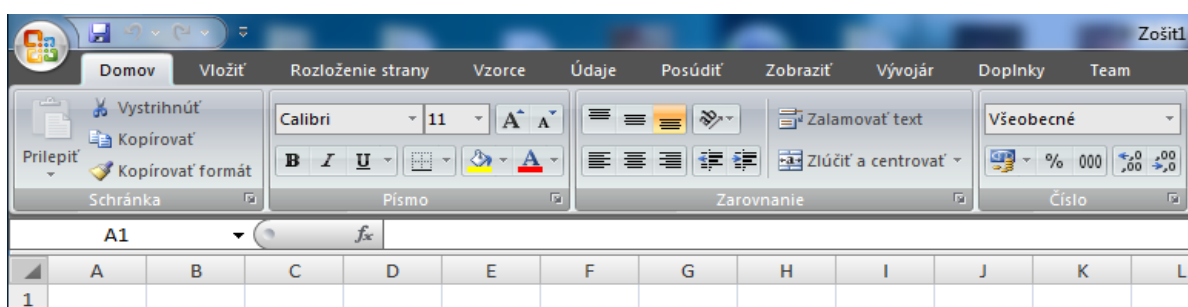
Tabuľka 2.1- Porovnanie verzií Excelu

Tak ako je vidno v tabuľke (tabuľka 2.1), grafické rozhranie prešlo zásadnou zmenou. Panel ponúk v Exceli 2003 (obr. 2.1) sa premenil na pás kariet (Ribbony, obr. 2-2), ktorý je viac intuitívnejší, bohatší a užívateľsky prítiahlivejší. Rôzne príkazy a funkcie boli v staršej verzii dostupné pomocou 3-4 klikov, kdežto v novej si stačí navoliť požadovanú záložku, kde môže užívateľ nájsť hľadanú funkciu.

Zaujímavosťou je pohľad pod povrch, kde väčšina podokien bola ponechaná z verzie 2003 (napr. vlastnosti bunky, nastavenia strany), ba niektoré dokonca ešte zo starších (používanie Microsoft Query a pod.).

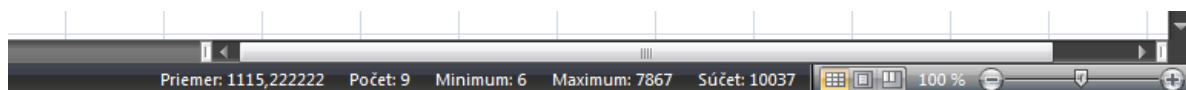


Obrázok 2.1- MS Excel 2003



Obrázok 2.2- MS Excel 2007

Jednou z nových vlastností, ktorá sa využíva a dokáže užívateľom ušetriť kvantum času je používanie rôznych štatistických funkcií v stavovom riadku. Stačí naň kliknúť pravým tlačidlom myši a vybrať požadovanú funkciu, ktorých je tam dostatok, ako je vidieť na obrázku 2.3.



Obrázok 2.3- Stavový riadok

Na podnet užívateľov, ktorí často potrebovali odlíšiť veľké množstvo hodnôt v tabuľkách, došlo k zásahu do funkcie podmieneného formátovania. Zatiaľ čo vo verzii 2003 mohol užívateľ využiť len 3 možnosti, v novej verzii 2007 je to neobmedzené. Tým sa myslí viacero pravidiel, ktoré užívateľ môže rôzne nakonfigurovať pomocou rôznych vizuálnych prechodov, údajových pruhov či množiny ikoniek. Tie sú dostupné pre údaje, zodpovedajúce použitým pravidlám. Zdokonalenia sa dočkalo aj vylepšenie automatického filtra tak, že vo verzii 2007 je možné filtrovať aj podľa formátu.[2]

Niekedy sú zmeny v softvéroch prospešné, inokedy nie. Podľa môjho názoru tento prípad ním nie je, obzvlášť u produktu spoločnosti Microsoft, ktorý sa využíva dennodenne v majoritných podieloch svetových organizácií. To môže byť jedným z dôvodom neimplementovania tejto verzie do svetových

podnikov, abstrahujúc od firemných nákladov. Osobne by som navrhoval pri inštalácii možnosť zvoliť užívateľské rozhranie, resp. nejaký minitutoriál ktorý by naznačil kam sa niektoré funkcie premiestnili (napr. zlom strany vo Word).

Celkové zhrnutie najzaujímavejších nezmienených zmien:

- ✓ 100 úrovní vrátenia akcie späť,
- ✓ až 60 úrovní vnorenia vo vzorcoch,
- ✓ maximálna dĺžka vzorca 8000 znakov,
- ✓ možnosť výstupu do PDF (pomocou doplnku),
- ✓ kontrola kompatibility,
- ✓ smartArt a krajšie grafy.

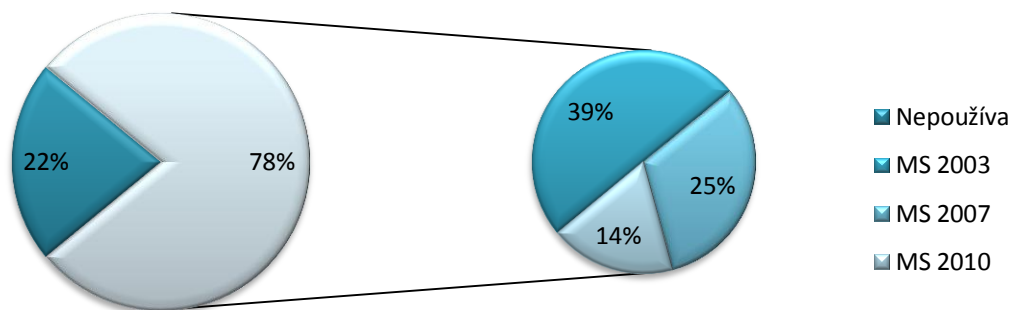
## 2.2 Štatistika

Z dôvodu pravidelných aktualizácií balíka Office Excel od spoločnosti Microsoft bol vytvorený dotazník pomocou nástroja Google Docs. Ten bol pravidelne aktualizovaný na sociálnej sieti Facebook. Pomocou tohto nástroja je možné zistiť s akými verziami sú ľudia spokojní alebo či vôbec používajú Microsoft Office.

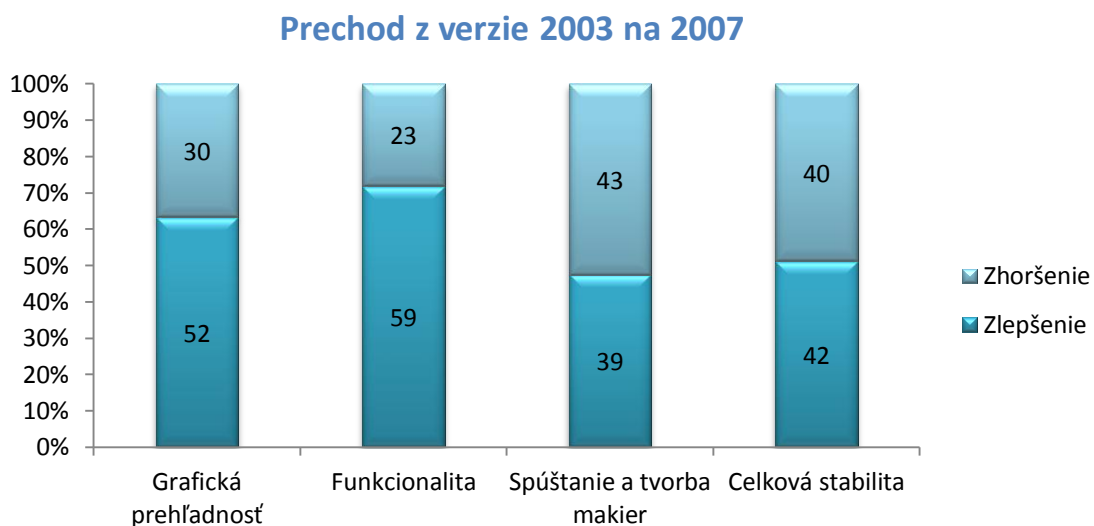
Na dotazník odpovedalo 105 respondentov, z ktorých 82 odpovedalo kladne k spolupráci s Microsoft Excel a dotazník pozostával z otázok:

- ✓ Použitie MS Office Excel v práci,
- ✓ Akú verziu užívateľ používa,
- ✓ Porovnanie v bodoch zlepšenia/zhoršenia vo funkčnosti a stabilite,
- ✓ Hodnotenie verzii 2003 a 2007 na škále od 1 až 10.

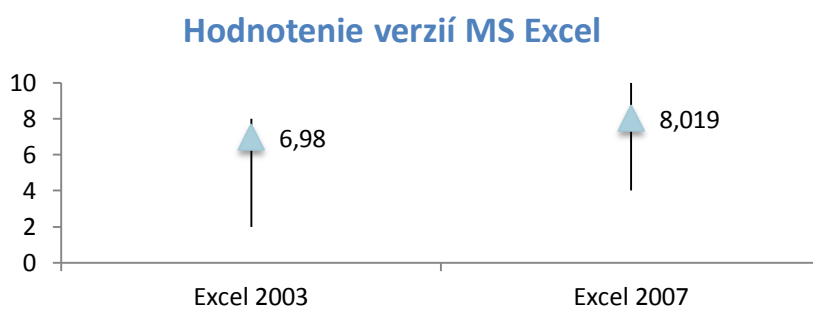
**Podiel používaných verzií MS Office Excel**



**Graf 2.1- Podiel používaných verzií**



Graf 2.2- Prechod z verzie 2003 na 2007



Graf 2.3- Hodnotenie verzií MS Excel

### 3 Teoretická časť

V teoretickej časti sa predstaví čo to vlastne VBA je a popíšu sa základne pojmy. Objasnia sa základy pri vytváraní užívateľských rozhraní pomocou ovládacích prvkov (ďalej len OP). Samozrejmosťou je spresnenie ich vlastností OP, kedy, kde a ako ich používať.

#### 3.1 Predstavenie VBA

Existuje množstvo rôznych definícií od rôznych autorov. Od tých odborných je najvýstižnejšou definíciou definícia pána Walkenbacha:

*„Predstavme si inteligentného robota, ktorý vie všetko o Exceli. Tento robot vykonáva príkazy a narába s Excelom rýchlo a presne. Ak chcete aby robot niečo vykonal, napíšete par riadkov inštrukcií pomocou kódu. Tento robot vykonáva všetko podľa inštrukcií, zatiaľ čo vy sedíte a pijete limonádu. O tom všetkom to je, jazyk pre robotov“[3].*

Z odborného hľadiska sa jedná o objektovo orientovaný programovací jazyk umiestňovaný do hostiteľských aplikácií MS Office. VBA spoločne s Visual Basic (VB) majú spoločný základ, ktorý je súčasťou inštalácie Microsoft Office. Keďže sa nachádza aj v Exceli, v programe,



Obrázok 3.1- Logo VBA

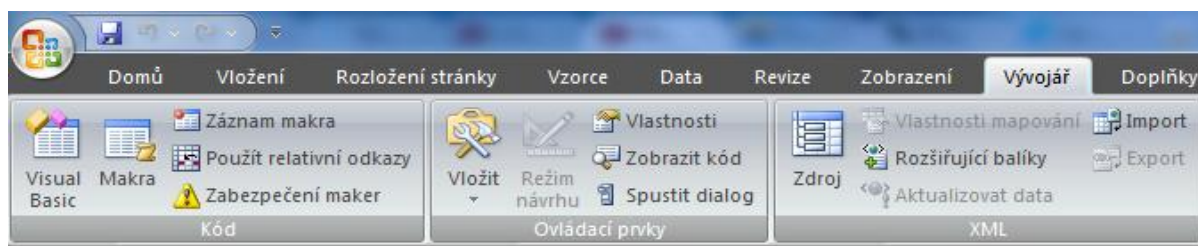
ktorý využíva majoritný počet firiem vo svete pre rôzne výpočty a analýzy, jeho predurčeným osudom je zautomatizovanie práce s rôznymi tabuľkami, výpočtami, vytváranie finančných modelov a pod.

##### 3.1.1 Makrá

Makro je nejaká zaznamenaná udalosť alebo postupnosť príkazov a funkcií, ktoré niekto vytvoril alebo napísal a väčšinou sa nachádza v príslušnom module vo Visual Basic Editore (ďalej len VBE).

Základným a nevyhnutným úkonom pre prácu s nimi, je zaškrtnutá možnosť „Zobraziť kartu vývojár“ (obr. 3.2) v možnostiach excelu a to nasledujúcim spôsobom:

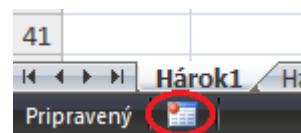
1. Kliknutím na MS Office tlačidlo v ľavom hornom rohu
2. Ďalej kliknutím na “Možnosti programu Excel“
3. V záložke Obľúbené zaškrtneme možnosť “Zobraziť kartu Vývojár”



Obrázok 3.2- Karta Vývojár

Následne je na obrázku vidieť v časti “Kód” možnosť záznamu makra, čomu sa hovorovo hovorí makrokamera. Je to akýsi pozorovateľ, po ktorého spustení zaznamenáva každú jednu akciu, ktorú vykoná užívateľ v Exceli (napr. kliknutie myšou na bunku, skrolovanie myšou a pod.) a uloží ju do príslušného modulu vo VBE. Tam je možno tento kód podľa potreby upravovať. Po spustení zaznamenávania je potrebné tento záznam (makro) vhodne pomenovať, prípadne nastaviť voliteľnú klávesovú skratku a zároveň sa nám tlačidlo kontextovo zmení pre jeho zastavenie.

Ako je vidieť na hore uvedenom obrázku(obr 3.2), bol umožnený prístup k vyššej funkcionalite Excelu, s ktorou sa budem zaoberať v nasledujúcich kapitolách. Pri aktívnej karte Vývojár sa v stavovom riadku zobrazí ikonka pre rýchle spustenie makrokamery (obr. 3.3). Pri ukončovaní práce s Excelom je potrebné, aby bol súbor uložený vo formáte s podporou makier, ináč by sa stratil celý kód z VBE (namiesto klasickej koncovky .xlsx sa uvedie koncovka .xlsm a výkričník v zobrazení ikonky).

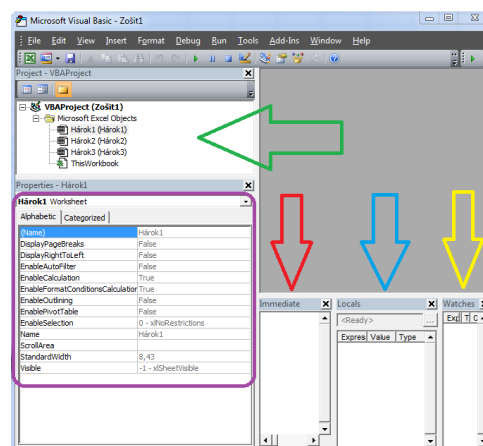


Obrázok 3.3- Spustenie makrokamery

Nevyhnutným úkonom pred spúšťaním makier je uloženie dokumentu, pretože akcie vykonané v Exceli pomocou makier sú **nevratné**. Ak by nastala chyba, tak by sa znovuotvoril dokument bez uloženia.

## 3.1.2 Visual Basic Editor

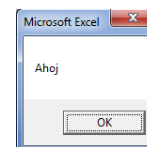
Editor jazyka VB je možné spustiť dvoma spôsobmi, a to kliknutím na ikonku „Visual Basic“ (obr.3.2) alebo klávesovou skratkou Alt + F11. Okamžite sa zobrazí VBE (obr 3.5). Ako je vidieť ,pozostáva z viacerých podokien. **Zelená šípka** vyznačuje projektový prieskumník, **bordový obdĺžnik** oblasť kde sa nachádzajú vlastnosti jednej z vybraných komponent, ktoré sa dynamicky menia v závislosti na aktívnom prvku. **Červená šípka** smeruje na „Okamžité okno“, kde môžeme priamo písať príkazy. Môže sa tam napísať najjednoduchší príkaz : MsgBox “Ahoj“ a stlačiť Enter.



Obrázok 3.4- VBE

Okno so správou sa okamžite zobrazí (obr.3.4). Okno „Locals“, označené **modrou šípku**, zobrazuje

všetky premenné, ktoré sa nachádzajú v danom projekte a zaznamenáva ich správanie, poprípade zmeny hodnôt. Okno označené **žltou šípkou** „Watches“ je najčastejšie využívané okno, do ktorého si môže užívateľ „nahádzať“ žiadané premenné, ktorých zmeny má v pláne pozorovať. Tzn. nemusí strácať čas skrolovaním medzi všetkými premennými v okne „Locals“ a bezproblémovo môže sledovať zmeny žiadaných hodnôt.



**Obrázok 3.5-**  
**Okno so**  
**správou**

### 3.1.3 Premenné

Často je potrebné ukladať dáta, s ktorými sa pracuje. Aby sa nemuseli vždy načítavať a tým zahlcovať pamäť, používajú sa vo VBA premenné. Je to akýsi pojem, do ktorého je možné uložiť napr. číslo, text, odkaz na bunku, s ktorou má pracovať a pod.

Pre vytváranie premenných sa dodržiavajú isté zásady:

- ✓ musí začínať písmenom a ničím iným,
- ✓ nesmie byť dlhšia ako 250 znakov,
- ✓ nesmie obsahovať anglické kľúčové slová Excelu (Workbook, Sheet a pod.),
- ✓ názov nesmie obsahovať medzeru, to sa rieši zväčšením začiatočného písmena alebo poprípade iným znakom (najčastejšie to býva podtržítka “\_”),
- ✓ neobsahuje diakritiku.

Z týchto zásad je možné usúdiť, že pre pomenovanie premenných je možné použiť akékoľvek názvy.

K definovaniu týchto premenných sa využívajú dátové typy. V nasledujúcej tabuľke sú uvedené najpoužívanejšie:

Dátový typ	Miesto v pamäti	Rozsah
Byte	1 byte	<0;255>
Boolean	2 byty	True alebo False
Integer	2 byty	<-32768;32767>
Long	4 byty	<-2147483648;2147483647>
Double	8 bytov	<4.94065645841246544E-324; 1.79769313486231570E+308> pre kladné
		čísla a <-1.79769313486231570E+308; 4.94065645841246544E-324> pre záporne čísla
String	10 bytov + dĺžka	Textový reťazec

<b>Date</b>	8 bytov	Dátum od 1.1.1900-31.12.9999
<b>Variant</b>	Najmenej 16 bytov	Tento typ sa nedefinuje
<b>Object</b>	8 bytov	Odkaz na objekt
<b>Date</b>	8 bytov	Dátum od 1.1.1900 do 31.12.9999

**Tabuľka 3.1- Dátové typy [4]**

Týmto sú položené základy definovania premenných. Pre ich deklaráciu v kóde sa používa nasledujúca syntax: **Dim** premenná **as** dátový typ.

```
Dim nazov_domu as string
Dim cislo_domu as Integer
```

Hore uvedeným deklarovaním sa dalo jasne najavo, že do premennej `nazov_domu` sa bude ukladať text a do premennej `cislo_domu` číslo. Výraz **as integer** sa nemusí písať, tým by sa tejto premennej automaticky priradil typ Variant. Ale keď sa vezme v úvahu 2. stĺpec v tabuľke 3.1, v ktorom sú uvedené jednotlivé veľkosti premenných, tak typ Variant zaberá 8 krát viac pamäte ako typ integer, čo pri menej výkonných počítačoch nie je veľmi vhodné a hlavne zdĺhavé. Preto je potrebné dopredu premyslieť v akých typoch sa budú hodnoty ukladať, aby si Excel alokoval optimálne miesto v pamäti.

Určite existuje veľa otázok, prečo sú vlastne premenné používané, na čo sú vlastne určené. Nasledujúci príklad objasní všetky nejasnosti. Napravo od znaku “=” sa nachádza hodnota, ktorá sa ukladá a naľavo kam bude uložená.

```
Sheets("Hárok1").Range("A1").Value= Sheets("Hárok1").Range("B2").Value
Sheets("Hárok1").Range("A2").Value = Sheets("Hárok1").Range("B2").Value + 1
Sheets("Hárok1").Range("A3").Value = Sheets("Hárok1").Range("B2").Value + 2
Sheets("Hárok1").Range("A4").Value = Sheets("Hárok1").Range("B2").Value + 3
```

V hore uvedenom kóde sa ukladá hodnota z hárku “Hárok1”, konkrétne z bunky “B2” do bunky “A1” v tom istom hárku. Následne sú do buniek A2-A4 sa ukladajú hodnoty z buniek B2 postupne zvyšované o 1 až 3. Veľkou nevýhodou tohto kódu je, že Excel bude načítavať hodnotu z bunky B2 4 krát. To by znamenalo že pri viacerých procedúrach, kde by sme využívali túto hodnotu, by ju Excel musel načítavať ešte viac krát, a to abstrahujeme od jeho následnej editácie. Preto je najvýhodnejšie uložiť hodnotu z bunky B2 do premennej “hodnotaB2”.

```
Dim hodnotaB2 as Integer
hodnotaB2 = Sheets("Hárok1").Range("B2").Value
Sheets("Hárok1").Range("A1").Value = hodnotaB2
Sheets("Hárok1").Range("A2").Value = hodnotaB2 + 1
Sheets("Hárok1").Range("A3").Value = hodnotaB2 + 2
Sheets("Hárok1").Range("A4").Value = hodnotaB2 + 3
```

Jediným rozdielom tohto kódu od predošlého je, že na začiatku je raz načítavaná hodnota z bunky B2 do premennej “hodnotaB2” a následne je tam uložená počas celého behu kódu. Po spustení tohto kódu sa

	A	B	C
1		100	
2	101	100	
3	102		
4	103		
5			
6			
7			

**Obrázok 3.6- Premenné**



vypíšu hodnoty do príslušných buniek, ako je vidieť na obrázku 3.6.

S premennými sa spája pojem **Option Explicit**, ktorý neznamena nič iné ako povinné deklarovanie premenných. Ak by sa pri písaní procedúr zabudlo na deklarovanie premennej, chybové hlásenie na to upozorní. Uvádza sa to ako prvý príkaz v modulu alebo sa v možnostiach panelu Tools, následne Options zaškrtnie možnosť **Require variable declaration**, ktorá ho bude automaticky vkladať na začiatok kódu.

### 3.1.4 Procedúry

Pojem procedúra sa chápe istý ohraničený úsek kódu ktorý vykonáva žiadaný efekt. To znamená že každý riadok kódu musí byť začlenený do istého (ohraničenia), nezahŕňajúc deklaráciu premenných. V predchádzajúcej stati (3.1.3.) sa vysvetlilo, prečo je používanie premenných nevyhnutné. Keby sa ten kód vložil do VBE, tak by to vyhodilo chybu. Preto je nutné ho ohraničiť, aby Excel vedel, od ktorého riadku po ktorý ma vykonávať, resp. aby v tom mal poriadok.

Začiatok procedúry sa označuje kľúčovým slovom **sub** (z angl. slova subroutine) s názvom procedúry, koniec procedúry označíme spojením **end sub**. Takže funkčný kód z predchádzajúcej stati by vyzeral nasledovne:

```
Public Sub procedura()  
Dim hodnotaB2 as Integer  
hodnotaB2 = Sheets("Hárok1").Range("B2").Value  
Sheets("Hárok1").Range("A1").Value = hodnotaB2  
Sheets("Hárok1").Range("A2").Value = hodnotaB2 + 1  
Sheets("Hárok1").Range("A3").Value = hodnotaB2 + 2  
Sheets("Hárok1").Range("A4").Value = hodnotaB2 + 3  
End Sub
```



Obrázok 3.7-  
Spustenie makra

Pred kľúčové slovo sub sa definuje o aký typ procedúry pôjde. Buď **public** (verejná), to znamená že je možné ju spustiť z akéhokoľvek modulu v kóde alebo **private** (súkromná), ktorá patrí danému modulu, kde bola vytvorená alebo sa nezadá nič, vtedy je dostupná zo všetkých modulov.

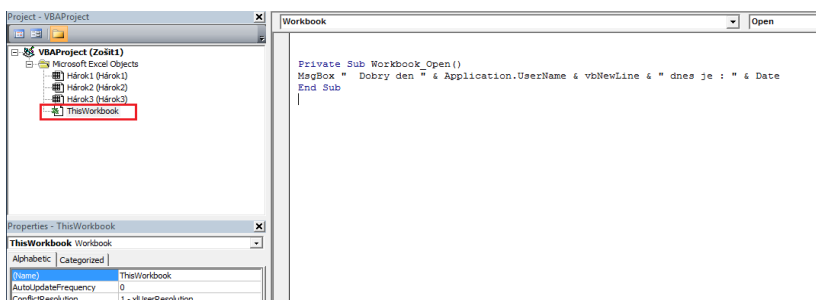
Na spustenie tohto kódu stačí jednoduché stlačenie zeleného tlačidla (obr. 3-7) alebo použitie skratky F5. Modré tlačidlo slúži na jeho zastavenie.

Makro a procedúra sú totožné pojmy, ktoré pozostávajú z riadkov kódu. Ich rozdielmi sú ich umiestnenia, mená a ich pôvod. Makrá sú väčšinou nahrávané pomocou makrokamery, zatiaľ čo procedúry sa píše ručne a musí sa do nich implementovať kód z nahratých makier. Už len z dôvodu ich ohraničenia, tzn. označenia, kde kód začína a kde končí. V inom prípade by Excel nevedel čo spúšťať, odkiaľ pokiaľ a následne by sa začali objavovať chybové hlásenia.

## 3.1.5 Udalostné procedúry

Jediným rozdielom týchto udalostných procedúr oproti obyčajným je, že tie spúšťajú procedúru, ako reakciu na nejakú udalosť, ktorá nastane. Pri vyvolaní procedúry sa tak vykonávajú príkazy, ktoré sú tejto udalosti priradené.

Ako príklad je uvedená udalosť typu **Workbook\_open**, to znamená že sa priradí kód udalosti otvorenia zošitu:

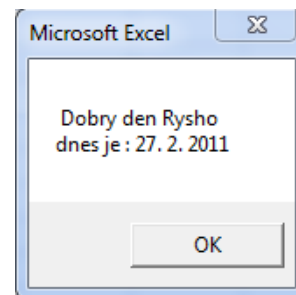


Obrázok 3.8- Udalostná procedúra

Komentár:

- obyčajný text dávame do úvodzoviek, medzery medzi úvodzovkami slúžia pre krajšie zarovnanie
- **Application.UserName** – stromové odkazovanie na užívateľa Excelu
- výraz "&" slúži na spájanie výrazov (tzv. reťazenie)
- **vbNewLine**- konštanta prechodu na nový riadok (alternatívou je **vbCrLf** alebo funkcia **chr(10)**)
- **Date**- dátumová funkcia Excelu, ktorá určí aktuálny dátum

Prvým krokom je kliknutie do projektového okna na možnosť "ThisWorkbook", tým sa v pravej časti okna zjaví priestor pre kód. Nad kódom sú vidieť 2 roletky (obr. 3.8). Tá vľavo je určená pre objekt, v našom prípade Workbook, a v tej napravo je udalosť ktorej sa priradzuje kód. S uvedeným kódom sa docielilo, že pri každom spustení zošita, po akceptovaní spúšťania makier, vybehne správa s informáciami (obr. 3.9).



Obrázok 3.9- Správa

V nasledujúcej tabuľke sú uvedené najpoužívannejšie udalosti zošita Excelu:

Udalosť	Popis
<b>Workbook_BeforeClose</b>	Udalosť pred zatvorením zošita
<b>Workbook_BeforePrint</b>	Udalosť pred spustením tlače
<b>Workbook_BeforeSave</b>	Udalosť pred uložením zošita
<b>Workbook_Open</b>	Udalosť pred otvorením zošita

Tabuľka 3.2- Udalosti Workbooku

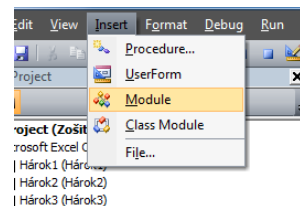
## 3.1.6 Funkcie (Funkčné procedúry)

Medzi ďalšie typy procedúr patria funkčné. Tieto disponujú možnosťou priechodu hodnoty (parametru) cez ich definíciu (u procedúr je to taktiež možné) a na konci kódu nám vracajú hodnotu. Funkcie a procedúry nemajú rovnaké schopnosti. Spúšťanie procedúr je zväčša vyvolávané stláčaním

určitých tlačidiel, poprípade vykonaním istých udalostí, zatiaľ čo u funkcií je to iné. Tie sú najčastejšie vytvárané pre vykonávanie výpočtov (viď príklad kvadratickej rovnice). Poprípade sú dostupné z buniek hárkov ako užívateľsky definované funkcie (UDF).

Deklarácia funkcie:

```
Function (premenná as Dátový typ) as Dátový typ
....
End Function
```



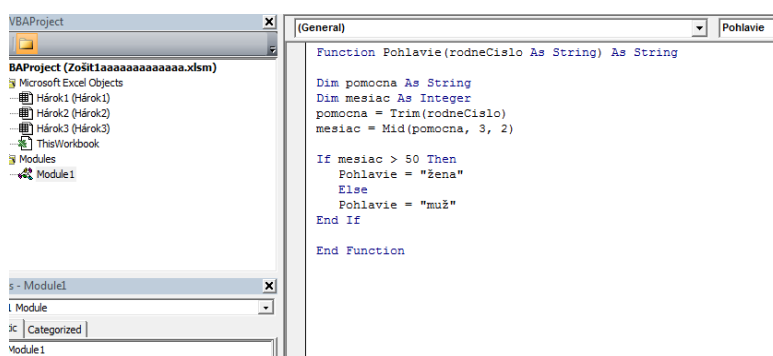
Obrázok 3.10- Modul

Excel pozostáva z veľkého počtu funkcií. Tie môžeme rozdeliť do dvoch skupín, podľa toho odkiaľ sú spúšťané:

- spúšťané z buniek hárkov,
- spúšťané z procedúr.

Pri funkciách je potrebné aby sa nachádzali v moduloch. Ten vytvoríme pomocou možnosti Insert a následne zvolíme Module, obr. 3.10.

Čo sa týka spúšťania funkcií z hárkov, uvedie sa príklad pre zistenie pohlavia osoby podľa rodného čísla (obr.3.11).

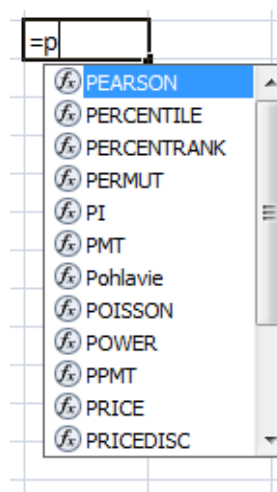


Obrázok 3.11- Funkcia pohlavie

```
Function Pohlavie(rodneCislo As String) As String
Dim pomocna As String
Dim mesiac As Integer
pomocna = Trim(rodneCislo)
mesiac = Mid(pomocna, 3, 2)
If mesiac > 50 Then
    Pohlavie = "žena"
Else
    Pohlavie = "muž"
End If
End Function
```

Komentár:

- Funkcia **Trim** slúži pri textových premenných na odobratie prípadných medzier v texte (prečistenie)
- Funkcia **Mid**, funguje ako doslova "vyťahovanie" jednotlivých znakov z textu, ktorá má tvar (daná textová premenná, znak od ktorého má začať počítať, počet znakov ktoré má "vytiahnuť"), pri číslach ich prevedie
- Funkcia **IF**, ak je podmienka splnená ( $Mesiac > 50$ ) vloží do funkcie Pohlavie "žena", ak nie tak vloží "muž"



Obrázok 3.12- Výber funkcie

V projektovom okne sa vytvorila podzložka Modules, kam sa budú ukladať všetky moduly, ktoré sa vytvoria. Pri použití funkcií stačí vpísať do ktorejkoľvek bunky znak "=" a ihneď sa zaktivuje zhruba

340 vstavaných funkcií. Ako je vidieť v kóde, uvedená užívateľská funkcia sa volá "Pohlavie", tak stačí stlačiť p (obr.3.12) a zvoliť ju. Keď sa človek bližšie pozrie na kód, tak vidí že premenná "rodneCislo" nám vstupuje do funkcie ako parameter, tak sa zapíše do zátvoriek a potvrdí enterom. V bunke sa na to zjaví výsledok funkcie.

Možno sa zdá byť nezmyselné, že rodne číslo vchádza do funkcie ako text, žiaľ funkciu **Trim** je možné použiť len na text. Ak by sa niečo počítalo, sčítavalo a pod. tak by to bol problém, v tomto prípade to nie je.

Čo sa týka funkcií spúšťaných v rámci procedúr, ako príklady sa zvyknú uvádzať jednoduché výpočty napr. výpočet diskriminantu kvadratickej rovnice, kde predpokladáme zadanie všetkých nenulových argumentov:

```
Option Explicit
Dim diskriminant As Integer
Sub kvadraticka_rovnica()
Dim a As Integer, b As Integer, c As Integer, vysledok As Integer, vysledokKorene_
As Integer
a = Application.InputBox("Zadajte hodnotu a", , , , , , 1)
b = Application.InputBox("Zadajte hodnotu b", , , , , , 1)
c = Application.InputBox("Zadajte hodnotu c", , , , , , 1)

If a = 0 Or b = 0 Or c = 0 Then
    MsgBox "Nastala chyba"
Else
    diskriminant = (b * b) - (4 * a * c)
    vysledok = MsgBox("Diskriminant je číslo " & diskriminant)
    If diskriminant > 0 Then
        vysledokKorene = MsgBox(" Kvadratická rovnica má korene: x1= " & _Koren1(a, _
        b, odmocninaDiskriminant(diskriminant)) & " a x2= " & _Koren2(a, b, _
        odmocninaDiskriminant(diskriminant)))
    ElseIf diskriminant = 0 Then
        vysledokKorene = MsgBox(" Kvadratická rovnica má dvojnásobný koreň: _ x1,2= " _
        & _Koren1(a, b, diskriminant))
    Else
        MsgBox ("Kvadratická rovnica má riešenie v obore komplexných čísel")
    End If
End If
End Sub
Function odmocninaDiskriminant(diskriminant As Integer)
odmocninaDiskriminant = Sqr(diskriminant)
End Function
Function Koren1(a As Integer, b As Integer, odmocninaDiskriminant As Integer)
If diskriminant > 0 Then
Koren1 = (-b + odmocninaDiskriminant) / (2 * a)
ElseIf diskriminant = 0 Then
Koren1 = -b / (2 * a)
End If
End Function
Function Koren2(a As Integer, b As Integer, odmocninaDiskriminant As Integer)
Koren2 = (-b - odmocninaDiskriminant) / (2 * a)
End Function
```

Ako je vidieť celý výpočet obsahuje jednu procedúru a 3 funkcie pre výpočet, odmocniny z diskriminantu, prvého a druhého koreňa. V procedúre sa načítajú hodnoty argumentov a, b, c, a vypočíta sa diskriminant, ak boli všetky hodnoty zadane. Následne v každej funkcii vstupujú určité

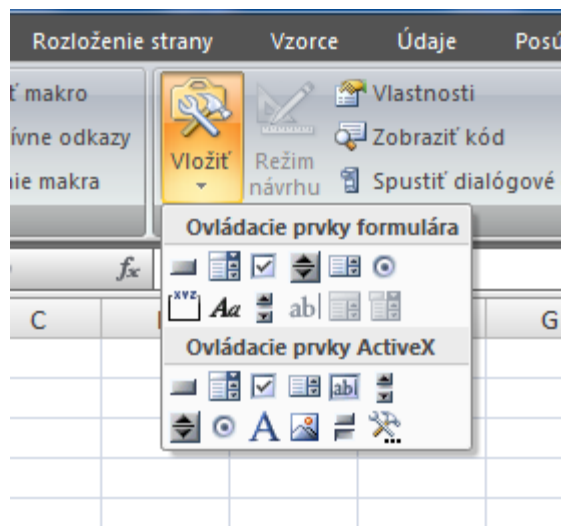
argumenty odvodené z týchto základných ako parametre. To je vo funkciách výhodné, tzn. že tieto funkcie nie sú “šité na mieru” pre procedúru kvadraticka\_rovnica, ale je možné vytvárať jej odvodené verzie v iných moduloch a využívať tieto funkcie pre výpočet jednotlivých veličín. Predpokladom by bol v odvodených verziách rovnaký názov premenných, aký je uvedený ako parameter. Funkcia InputBox je vysvetlená v stati 3.2.4.1.



*Neodporúča sa používanie klauzuly private pred funkciou. Tým sa zneviditeľní voči iným modulom a aj hárku, čo býva častou chybou užívateľov.*

### 3.2 Užívateľské rozhranie

Význam tohto výrazu pochádza z anglického spojenia “user interface“, a môže sa pod ním rozumieť vzhľad (štýl), zobrazenie alebo ovládanie, v tomto prípade dát v tabuľkách pomocou prvkov. Či už tlačidlá okolo dát v hárkoch alebo užívateľské okno s jednotlivými vytvorenými možnosťami, ktoré uľahčujú prácu, prejde sa oboma možnosťami. Obzvlášť sa práca zameria na typ s užívateľským oknom (formulárom), v ktorom sa skrýva nespočetne veľa možností a hlavne je užívateľský “prítulnejšie“. Ukážu sa základne zásady práce s ovládacími prvkami, čomu sa vyvarovať, čo kam umiestniť a najmä triky, ktoré určite skrátia čas tvorenia vašej aplikácie.



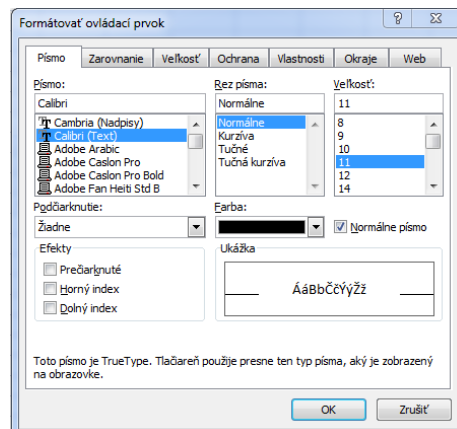
Obrázok 3.13- Ovládacie prvky

Majoritný podiel na vzhľade užívateľského prostredia majú ovládacie prvky (ďalej OP). Delia sa na OP formulárov a OP ActiveX. V súčasnosti sa pri programovaní vo VBA uprednostňujú ActiveX prvky. Tieto prvky sa dajú použiť pre ošetrovanie rôznych udalostí, ktoré nastávajú vo chvíli, keď sa daný prvok použije. OP z panelu nástrojov formulára sú kompatibilné s predchádzajúcimi verziami Excelu a okrem toho je možné ich použiť nielen v hárkoch zošita ale ako jediné aj v grafoch vložených ako zvláštny hárok. Tieto prvky reagujú len na udalosť **Click()**, kedy vykonajú makro, ktoré je udalosti priradené[5].

V oboch prípadoch sa však OP vkladajú do hárku rovnako. Kliknutím ľavým tlačidlom myši na ľubovoľný OP a následné kliknutie do hárku. Ďalšou možnosťou je konkrétny OP na hárok pretiahnuť, podržaním ľavého tlačidla a umiestniť na hárok. Vhodným doplnkom je podržanie klávesy Alt, ktorá zarovná daný OP k okrajom najbližšej bunky [6]. Pre editáciu OP formulárov postačí jednoduché kliknutie na nich, ale pri OP ActiveX je nevyhnutné mať zapnutý režim návrhu (obr.3.2), ktorý je na danom obrázku nečinný, pretože zatiaľ nebol vložený žiaden OP.

## 3.2.1 Ovládacie prvky formulárov

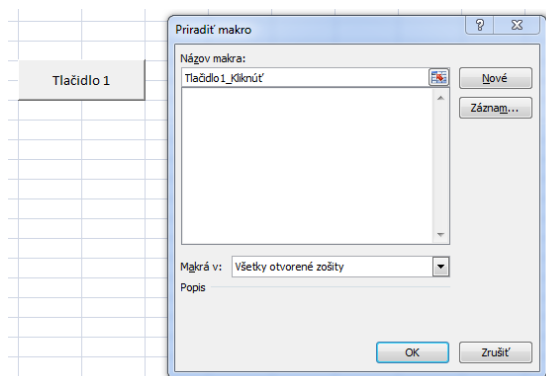
Tieto prvky je možné nájsť na obr. 3.13 v hornej časti, kam je možné sa dostať kliknutím na tlačidlo “Vložiť” v karte Vývojár. Po ich vložení si Excel automaticky vypýta priradenie makra, kde sa predpokladá, že je vytvorené. V prípade ak nie je, tak je vždy cesta ako sa dostať späť k tejto obrazovke, stačí kliknúť na hranu daného OP, pri ktorom sa aktivuje kontextové menu a zvolíte možnosť “Priradiť makro”. Tieto prvky sa zväčša používajú vtedy, ak nie je v úmysle ich kód písať (tvoriť), poprípade upravovať, ale len nahrat makrokamerou. Ako už bolo poznamenané, tieto prvky reagujú len na akciu **Click()**, ktorá sa od verzie 2007 poslovenčila, samozrejme aj s názvom prvku ako je vidieť na obrázku 3.15.



Obrázok 3.14- Vlastnosti OP formulára

Z obrázku je vidno, že žiadne makrá nie sú vytvorené. Tie je možné vytvoriť stlačením tlačidla “Záznam”, čo užívateľa priamo presmeruje na makrokameru alebo tlačidlom “Nové”, ktoré presmerováva do novovytvoreného modulu vo VBE. To znamená, že vytvorí začiatok a koniec procedúry. Do tela tejto procedúry je možné následne nakopírovať akýkoľvek kód, ktorý sa predtým nahral, poprípade napísal.

V prípade záujmu o úpravu vzhľadu<sup>1</sup> alebo funkčnosti OP formulára, v tomto prípade tlačidla, je možné naň kliknúť pravým tlačidlom a vybrať z kontextovej ponuky možnosť “Formátovať ovládací prvok”, obr. 3-14. Následne vyskočí okno z viacerými záložkami, ktoré sú dosť intuitívne. Ako napr. veľkosť textu, jeho zafarbenie alebo farba tlačidla. Z týchto vlastností je asi najužitočnejšou zmena veľkosti vzhľadom k veľkosti bunky. Je to takto implicitne nastavené, čo môže niekedy zmiast neskúseného užívateľa, ktorý si napr. skryje stĺpec na ktorom sa nachádza daný prvok, tým sa aj on zneviditeľní. Poprípade sa to môže nechať takto nastavené, ale prvky by sa mali nechať umiestnené v ukotvených stĺpcoch, poprípade riadkoch.



Obrázok 3.15- Vlastnosti OP formulára

V nasledujúcej tabuľke je prehľad použiteľných OP formulára s popisom:

<sup>1</sup> Možnosť formátovania a popisu OP formulára je možné len u tlačidla

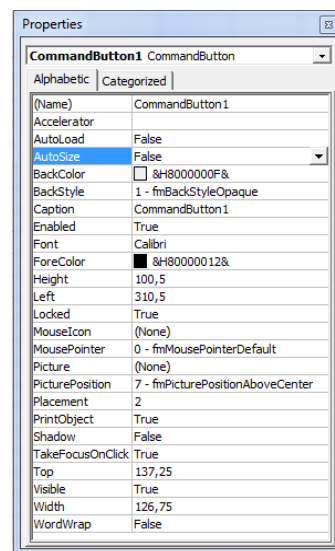
Názov	Typ	Popis
Tlačidlo	Button	Tlačidlo, ktoré po stlačení vyvolá dané makro
Pole so zoznamom	DropDown	Rozbaľovacie pole so zoznamom položiek
Zaškrtávacie políčko	CheckBox	Aktivuje alebo deaktivuje danú(é) položku(y)
Číselník	Spinner	Zvyšuje alebo znižuje hodnotu
Zoznam	ListBox	Zobrazuje zoznam položiek
Prepínač	OptionButton	Umožní vybrať zo skupiny združených prvkov len jednu možnosť
Skupinový rámček	GroupBox	Združuje súvisiace prvky do jedného ohraničenia
Popisok	Label	Popisuje nejaký prvok a pod.

Tabuľka 3.3- Ovládacie prvky formulára [7]

Typ TextField a ComboList sú dostupné len v dialógovom okne 5.0

## 3.2.2 Ovládacie prvky ActiveX

Tieto ovládacie prvky sú porovnateľné s tými, ktoré sa používajú v programovacích jazykoch typu Microsoft Visual Basic. Nachádzajú sa na obr. 3.13, v spodnej časti kontextového okna. Po pridaní do hárku je možné si všimnúť ako prvé zmenu jazyka, z dôvodu napojenia na VBE, ktorý je výhradne v jazyku anglickom a následnú aktiváciu **režimu návrhu**, tzn. že len pri aktívnom režime návrhu je možná editácia daného OP. V opačnom prípade prebieha pri kliknutí naň vykonanie priradeného kódu. Možnosti formátovania OP ActiveX obsahujú len 4 záložky (Veľkosť, Ochrana, Vlastnosti a Web) OP formulárov (obr. 3.14). Zároveň disponujú novou možnosťou kontextového menu OP. Nazýva sa “Vlastnosti“ (obr. 3.16), ktorá umožňuje meniť farbu pozadia prvkov alebo vzhľad kurzora myši, na ktorý sa má ukazovateľ zmeniť, pokiaľ užívateľ naň prejde myšou a pod. Pri neznalosti niektorého z atribútov prvku stačí na daný atribút (názov) kliknúť a stlačiť klávesu F1, ktorá aktivuje nápovedu.



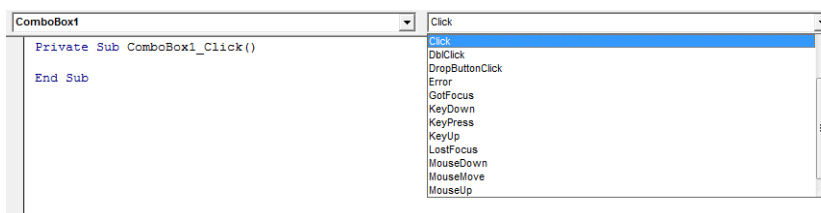
Obrázok 3.16- Vlastnosti OP ActiveX

V skutočnosti ide o totožné prvky, ktoré je možné vkladať do formulárov vytváraných vo VBE. Pokiaľ sa vložia do hárku zošita Excelu OP ActiveX, vytvorí zdrojový kód makra, ktorý sa vzťahuje



k identifikačnému číslu prvku. Vo chvíli, kedy užívateľ použije prvok formulára, spustí nami zapísaný kód. Pod pojmom použitie sa chápu rôzne udalosti, ktoré môžu nastať pri narábaní s daným OP.[8]

U prvkov je možné napísať kód, na ktorý budú reagovať skryté udalosti prvkov ActiveX. Roletový zoznam vľavo obsahuje prehľad všetkých OP aktívneho hárku (obr. 3.17). Po



Obrázok 3.17- Roletkový zoznam udalostí OP

následnom výbere ľubovoľného objektu z ľavej roletky sa nám v pravom roletovom zozname zobrazia všetky udalosti vzťahujúce sa k tomuto OP. Kliknutím na ľubovoľnú udalosť sa vo VBE vytvorí procedúra vzťahujúca sa k danému OP a jeho udalosti, ako je vidieť na obrázku. Do tela tejto procedúry stačí vložiť požadovaný kód a je vytvorený funkčný OP.

Pre zmenu vlastností sa používa panel vlastností OP (obr. 3.16), ale týka sa to vlastností pri spustení programu. Počas behu programu je možné taktiež meniť ich vlastnosti (zväčšiť písmo, skryť a pod.). Ak meníte viacero vlastností rovnakého OP je vhodné použiť klauzulu **with OP** a **end with**, ktorá umožní vykonať viacero operácií s jedným OP, použitá na str. 34.

### 3.2.3 Konvencia názvov OP

V tejto tabuľke sa priradzuje každému OP istá skratka pre rýchlejšiu a efektívnejšiu prácu s kódom. Väčšinou sú to isté spoluhlásky z anglického názvu.

Skratka OP	Názov	Slovenský ekvivalent
chkb	CheckBox	Zaškrtávacie políčko
cmb	ComboBox	Rozbaľovacie textové pole
cmd	CommandButton	Tlačidlo
frm	UserForm	Užívateľský formulár
img	Image	Obrázok
lbl	Label	Popisok (menovka)
lst	ListBox	Zoznam
opt	OptionButton	Prepínač
fra	Frame	Zoskupenie
spn	SpinButton	Prepínacie tlačidlo
tbs	TabStrip	Záložka
txt	TextBox	Textové pole
mlt	MultiPage	Viacnásobná stránka



Tabuľka 3.4- Konvencia OP

## 3.2.4 Práca s oknami

Práca s oknami bola rozdelená do 3 častí. Aj keď MsgBox a InputBox nepatria medzi OP, je veľmi vhodné, niekedy nevyhnutné používať ich pri vytváraní kódu.

### 3.2.4.1 MsgBox a InputBox

Tieto funkcie nepatria medzi OP ale je veľmi vhodné, niekedy nevyhnutné používať ich pri vytváraní kódu.

Funkciu **MsgBox** sa využíva v prípadoch informovania užívateľa o určitej zmene stavu (obr. 3.4), v tomto prípade zmenou stavu bolo otvorenie zošita. Je možné vložiť ju na akékoľvek miesto v kóde, môže nadobúdať rôzne významy (rôzne významy znakov v správe). Slúži ako na strane vstupnej, tak aj na výstupnej.

Syntax:

- a.) pre prípad jednoduchšej správy, ktorá zobrazí len tlačidlo "OK" (obr. 3.4)

```
MsgBox "Názov správy"
```

- b.) v prípade zložitejšej správy (nevyhnutné vytvoriť textovú premennú)

```
Dim sprava as String
sprava = MsgBox("Názov správy", typ ikonky + typ tlačidiel, "Názov okna_
so správou")
```

Zložitejšou správou sa rozumie správa, v ktorej definujeme žiadané tlačidlá alebo ikonky pre zobrazenie. Názvom správy sa definuje, čo sa v danom okne ako text má nachádzať. V ďalšej časti zátvorky sa definuje žiadaný typ tlačidiel a ikonky, ktoré ak chceme kombinovať musia byť spojené spojovníkom "+" a na konci uvedieme názov okna so správou.

Konštanta	Popis
<b>vbOKOnly</b>	Zobrazí len tlačidlo OK (obr. 3.4)
<b>vbOKCancel</b>	Zobrazí tlačidlá OK a Zrušiť
<b>vbAbortRetryIgnore</b>	Zobrazí tlačidlá Späť, Znova a Ignorovať
<b>vbYesNoCancel</b>	Zobrazí tlačidlá Áno, Nie a Zrušiť
<b>vbYesNo</b>	Zobrazí tlačidlá Áno, Nie
<b>vbRetryCancel</b>	Zobrazí tlačidlá Znovu a Zrušiť
<b>vbCritical</b>	Zobrazí v správe kritickú ikonku
<b>vbQuestion</b>	Zobrazí v správe otáznik
<b>vbExclamation</b>	Zobrazí v správe varovnú ikonku
<b>vbInformation</b>	Zobrazí v správe informačnú ikonku
<b>vbOK</b>	Potvrdenie správy tlačidlom OK
<b>vbCancel</b>	Potvrdenie správy tlačidlom Zrušiť
<b>vbAbort</b>	Potvrdenie správy tlačidlom Späť
<b>vbRetry</b>	Potvrdenie správy tlačidlom Znova
<b>vbIgnore</b>	Potvrdenie správy tlačidlom Ignorovať

**vbYes**

Potvrdenie správy tlačidlom Áno

**vbNo**

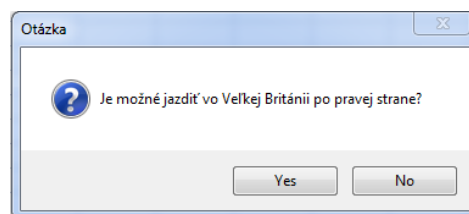
Potvrdenie správy tlačidlom Nie

Tabuľka 3.5- Jednotlivé konštanty pre prácu s MsgBoxom

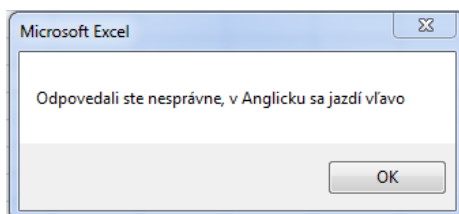
**Prvá skupina** popisuje typ tlačidiel zobrazených v MsgBox-e, **druhá** určuje štýl ikony. Posledná skupina dáva možnosť zistenia na ktoré tlačidlo užívateľ klikol, ako odpovedal. To je možné zistiť jednoduchým príkazom **If** na konkrétnom príklade, v ktorom sa bude zisťovať či užívateľ vie, že sa vo Veľkej Británii jazdí vľavo:

```
Sub YesNo()  
Dim odpoved As Byte  
Dim sprava As String  
sprava = "Je možné jazdiť vo Veľkej Británii po pravej strane?"  
odpoved = MsgBox(sprava, vbQuestion + vbYesNo, "Otázka")  
If odpoved = vbYes Then  
    MsgBox "Odpovedali ste nesprávne, v Anglicku sa jazdí vľavo"  
Else  
    MsgBox "Odpovedali ste správne"  
End If  
End Sub
```

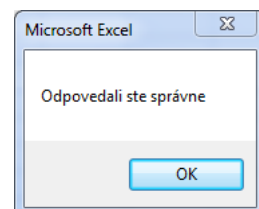
Pri spustení procedúry vybehne správa s otázkou (obr. 3.18), či je možné jazdiť vo Veľkej Británii po pravej strane. Po kliknutí na odpoveď Yes (obr. 3.19) vybehne správa s chybnou odpoveďou, pri odpovedi No (obr. 3.20) vybehne správa so správnou odpoveďou.



Obrázok 3.18- MsgBox otázka



Obrázok 3.19- Odpoveď Yes



Obrázok 3.20- Odpoveď No

Je možné použiť aj inú variantu kódu cyklom Case a dosiahnuť rovnaký výsledok, blok príkazu If je možné zameniť za blok nasledovného cyklu :

```
Select Case odpoved  
    Case vbYes  
        MsgBox "Odpovedali ste nesprávne, v Anglicku sa jazdí vľavo"  
    Case vbNo  
        MsgBox "Odpovedali ste správne"  
End Select
```

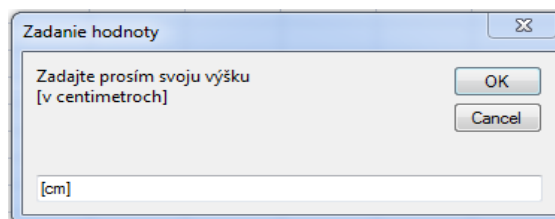
Pri potrebe zadania určitých hodnôt sa na strane vstupnej používa funkcia **InputBox** (obr.3.21). Je to akési dialógové okno s textovým poľom, ktoré čaká na zadanie vstupu užívateľa. Inými slovami je to interakcia s užívateľom na vloženie hodnôt alebo znakov. Vo väčšine prípadoch sú to len číselne znaky, a to z dôvodu nemožnosti obmedzenia textového poľa na určitý počet znakov, ktoré by sme

potrebovali. Je možné ju ukladať buď do premennej alebo do nejakej bunky v hárku. V prípade ukladania do buniek sa nemáme čoho obávať, ale v prípade premennej, keď užívateľ zvolí iné potvrdzovacie tlačidlá okrem OK, a tak nastáva problém, pretože tie vracajú prázdny reťazec a celkovo to končí chybou. To sa vyrieši použitím objektu application, takže stačí pridať application. pred inputBox (obr. 3.22) a zobrazí sa nám totožné dialógové okno menších rozmerov.

Syntax:

```
premenná = InputBox ("Text okna", "Názov okna", "Predefinovaný text", X-pozícia, _  
Y- pozícia)
```

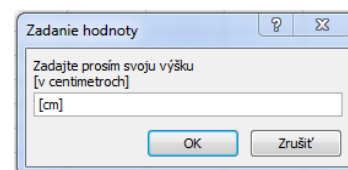
Podľa syntaxe je vidieť, že hodnota, ktorú zadáme do InputBoxu sa nám automaticky uloží do premennej. Ako parameter funkcie uvádzame text ktorý sa nám zobrazí v okne, názov tohto okna. Ďalej tam používame predefinovaný text, to znamená že v textovom poli sa bude vyskytovať



Obrázok 3.21- InputBox

hodnota alebo nejaké znaky, zväčša v prípade ak chcete niekoho ovplyvniť pre vloženie hodnoty v určitom formáte. Nakoniec sa tam môžu zadať 2 čísla, prvé je X-ová vzdialenosť od okraja obrazovky a druhé je Y-ová.

Ako príklad druhého typu sa uvedie zadanie výšky užívateľa a okno zobrazíme úplne vľavo hore obrazovky (obr. 3.22):



Obrázok 3.22- Application InputBox

```
sprava = Application.InputBox(„Zadajte prosím svoju  
výšku“ & vbCrLf & „[v_ centimetroch]“, „Zadanie  
hodnoty“, „[cm]“, 1, 1, , , 1)
```

Z týchto dvoch možností je výhodnejšie využívať možnosť podľa obrázka 3.22. Problémy s tlačidlami zrušiť a križik tu odpadajú. Navyše výhodou tohto typu InputBoxu je, že sa dá obmedziť na istý typ dát. Pri deklarovaní InputBoxu sa do parametru na posledné miesto zadá určitá hodnota z tabuľky 3.6.

Inými slovami sa to dá nazvať kontrolou správnosti zadávaných dát. Najdôležitejšou hodnotou z tabuľky (tabuľka 3.6) je hodnota 2, obmedzenie na číslo. Pri definovaní vstupnej hodnoty ako text sa zvykne používať funkcia Left (Zľava), ktorá nám tu premennú oreže, poprípade funkcia Trim (Prečistiť). Dôvodom je, neexistencia možnosti obmedzenia InputBoxu na určitý počet znakov.

Hodnota	Význam
0	Vzorec
1	Číslo
2	Text
4	Logická hodnota
8	Odkaz
16	Chybová hodnota
4	Pole hodnôt

Tabuľka 3.6- Tabuľka hodnôt[9]

## 3.2.4.2 Dialógové okno 5.0

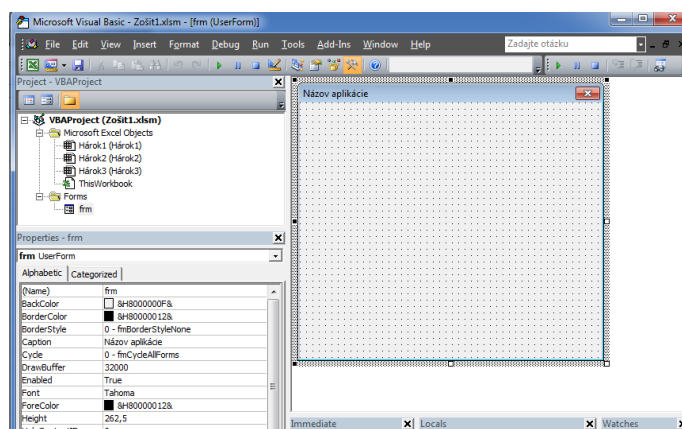
Dialógové okno verzie 5.0 sa vytvára formou vloženia do listu v hárku. Stačí pravým tlačidlom kliknúť na daný hárok, možnosť vložiť a následne vybrať možnosť “Dialógové okno MS Excel 5.0“. Je to v podstate mriežka, ktorá môže obsahovať text a ovládacie prvky. V Exceli 95 sa používali pre interakciu s užívateľom formou užívateľských dialógov. V Exceli 97 boli zavedené užívateľské formuláre, ktoré svojou funkčnosťou a vlastnosťami jednoznačne prevyšujú a nahradili ich (viď nasledujúca kapitola).

## 3.2.4.3 Userform

Userform, slovensky formulár, spoločne s hárkami sú základnou ponukou užívateľského rozhrania aplikácie. Inými slovami Userform je akýmsi kontajnerom pre ostatné OP ActiveX. Jeho vzhľad či použitie môžu byť rôzne, od zobrazenia jednoduchých hlásení, cez zobrazenia výsledkov až ku komplikovaným formulárom pre zadávanie dát.

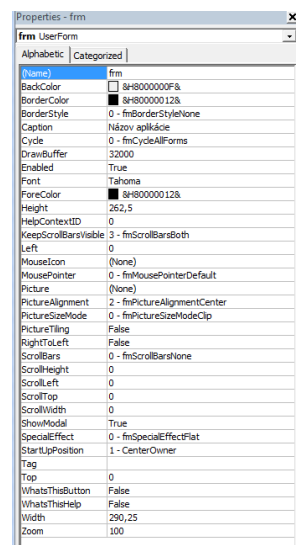
Základné rysy návrhu formulára:

- ✓ Použitelnosť,
- ✓ Intuitívnosť,
- ✓ Vhodné názvoslovie OP,
- ✓ Združovanie prvkov,
- ✓ Rôzne typy úloh do rôznych kariet,
- ✓ Bezchybnosť.



Obrázok 3.23- Userform

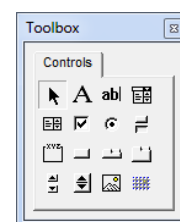
Pre vytvorenie užívateľského formulára sa užívateľ prepne do VBE a z menu vyberie možnosť Insert a následne Userform. Jestvuje aj iná možnosť a tou je kliknutie pravým tlačidlom myši v okne projektov na príslušný projekt a výberom položky Insert a následne Userform (obr. 3.23). Týmto sa vložil formulár do VBE, to znamená, že môžeme začať meniť jeho vlastnosti (atribúty) (obr. 3.24). Základným krokom je vhodné pomenovanie formulára v rámci kódu (vlastnosť **Name**), čo je v našom prípade “frm“, podľa konvencií (tab. 3.4). Ďalším intuitívnym krokom je zmena veľkosti formulára, kde jestvujú 2 spôsoby. Prvým je kliknutie myšou na pravý dolný roh formulára a následným posúvaním myšou sa zmení jeho veľkosť.



Obrázok 3.24- Vlastnosti OP ActiveX

Ďalšou možnosťou je vlastnosť **Height** čo je výška a **Width** šírka. Názov formulára zabezpečuje vlastnosť **Caption**, o podfarbenie sa stará vlastnosť **BackColor** alebo vlastnosť **Picture**, kde je možné nastaviť rôzne obrázky ako pozadie. Ak užívateľovi nevyhovuje znak myši pri prechádzaní myšou nad formulárom tak si stačí vybrať jednu z možností pri vlastnosti **MousePointer**. Vlastnosť **Enabled** sa nezvykne meniť, jedine že by bolo potrebné deaktivovať formulár, to znamená, že by nebolo možné vykonať udalosti nad daným formulárom. **StartupPosition** sa zvykne nechať na implicitnej hodnote, čo znamená, že sa daný formulár zobrazí v strede obrazovky. V prípade použitia viacerých formulárov, je možné manuálne nastaviť súradnice zobrazenia ľavého horného bodu. Vlastnosť **Top** je vzdialenosť horného okraja a **Left** ľavého.

Pre aktiváciu okna s OP, ktoré sa nazýva **ToolBox** (obr. 3.25) stačí kliknúť na formulár. Ak by nastala situácia, že by sa nezobrazilo, tak je potrebné využiť možnosť v menu **View** a následne **Toolbox**. Následným kliknutím a pretiahnutím myšou na formulár daný OP vložiť. Po pridaní OP sa jasne zmenia vlastnosti daného OP. Pre výber konkrétneho OP naň stačí kliknúť, v prípadoch keď to nie je možné, stačí použiť malé rozbaľovacie okno nad vlastnosťami daného OP (obr. 3.24).



Obrázok 3.25-  
ToolBox

Jednotlivé OP je možné postupne aktivovať pomocou klávesy **Tab**, poradie týchto prvkov sa riadi pomocou vlastnosti **TabIndex**. Pri pridaní každého nového OP sa hodnota indexu zvýši o 1, čo znamená že dvojité priradenie rovnakého indexu v jednom kontajneri (formulári) nie je možné. [10]

Po nastavení všetkých potrebných vlastností je dobré si uvedomiť akú funkcionálnosť bude daný formulár vykonávať. Ak je to obyčajné menu s niekoľkými OP, tak postačí ponechať ako hlavný kontajner formulár. V prípade väčšej zložitosti (viacerých záložiek) alebo z dôvodu systematického delenia, sa zvykne používať OP **Multipage**, ktorému sa venuje stať 3.2.15.



*Je zbytočné nechávať prázdne miesta po okrajoch. Pre ich vytváranie sa treba držať zásad návrhu formulára, pri podfarbovaní OP je vhodné podfarbiť aj kontajner, kde sa daný OP nachádza.*

### 3.2.5 Label

Label alebo tzv. popisok slúži pre zobrazenie určitého textu kdekoľvek vo formulári. Teda bez nutnosti nejakých úprav je možné ho použiť k úprave vlastností ako nadpisov (popisov) k OP. Vlastnosťou **Name** sa pomenuje daný prvok, do vlastnosti **Caption** sa zadáva zobrazovaný text, ktorý je možné ľubovoľne formátovať pomocou vlastnosti **Font**. Ďalej je možné použiť vlastnosti ako **BorderStyle** kde sa nastaví spôsob ohraničenia a **BackColor**, či **Picture**, kde je možné nastaviť pozadie formou farby či obrázku.[11]

V nasledujúcej tabuľke sú uvedené rôzne príklady popiskov s ich detailným popisom a vhodnosťou použitia:

Label	Popis	Vhodnosť
	Základný typ popisku	Áno
	Modrá farba písma, typ písma tučné (Font)	Áno
	Typ písma tučné, kurzíva (Font)	Áno
	Zväčšené písmo (Font), modré podfarbenie (BackColor), červené orámovanie (BorderStyle, BorderColor)	Nie
	Základný typ popisku, obyčajné orámovanie BorderStyle), špeciálny efekt " Etched" (SpecialEffect)	Áno

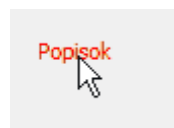
Tabuľka 3.7- Typy OP typu Label

Ako vidieť z tabuľky 3.7, štvrtý prípad popisku je dosť nevhodný. Všeobecne sa vlastnosť **BackColor** a **Picture** u typu Label nezvykne meniť a necháva sa implicitné podfarbenie formulára. Nepôsobí to totiž prívetivo. Výnimkou by mohol byť prípad, ak by sa zmenilo na totožné podfarbenie OP na ktorom sa daný popisok nachádza.

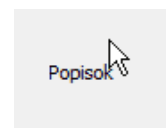
Ďalej ako každý iný OP disponuje udalostnými typmi procedúr. Pri type OP Label sa ich vyskytuje presne 8. V prvom príklade je využitá udalosť **MouseMove**:

```
Private Sub lbl_MouseMove(ByVal Button As Integer, ByVal Shift_
    As Integer, ByVal X As Single, ByVal Y As Single)
    lbl.ForeColor = &HFF&
End Sub
Private Sub frm_MouseMove(ByVal Button As Integer, ByVal Shift_
    As Integer, ByVal X As Single, ByVal Y As Single)
    lbl.ForeColor = &H80000012
End Sub
```

Táto udalostná procedúra ma čisto vzhľadový charakter. Po prejdení myšou na daný OP v našom prípade lbl (**zelená farba**) sa jeho vlastnosť **ForeColor** zmení na hodnotu &HFF&, čo je kódové označenie červenej farby (obr. 3.27). Automaticky, keď opustíte oblasť OP lbl a prejdete na formulár UserForm (**modrá farba**) tak sa vlastnosť OP lbl **ForeColor** zmení na pôvodnú &H80000012, čo je čierna (obr.3.26).



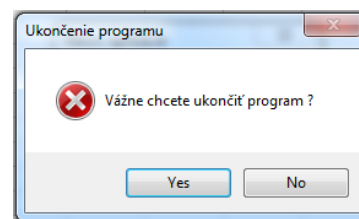
Obrázok 3.27- MouseMove 1



Obrázok 3.26- MouseMove 2

Po funkčnej stránke sa dá popisok využiť ako tlačidlo. Pomocou udalostnej procedúry **DbClick** alebo v našom prípade **Click**.

```
Private Sub lbl_Click()
    Dim text As String
    Dim sprava As Byte
    Dim nadpis As String
    text = "Vážne chcete ukončiť program ?"
    nadpis = "Ukončenie programu"
    sprava = MsgBox(text, vbCritical + vbYesNo, nadpis)
```



Obrázok 3.28- Ukončenie programu

```
If sprava = vbYes Then  
    Application.Quit  
Else  
    Exit Sub  
End If  
End Sub
```

Pri kliknutí sa vybehne okno s otázkou. Pri kliknutí na Yes sa vykoná Quit, čo znamená že program sa ukončí. Pri kliknutí na No sa vykoná Exit Sub, čo znamená že procedúra skončí na tomto riadku, tzn. že sa nič nevykoná.

Pri zobrazovaní textu je ešte dôležitá vlastnosť **WordWrap**, ktorá udáva, či sa má text na konci riadku zalomiť.



*Pri použití popiskov ako názvov OP ako napr. textového poľa a pod. sa odporúča používať veľkosť nie väčšiu ako je pomenovávaný OP. Môže dôjsť k zakrývaniu iných OP. Veľkosť sa nastavuje rovnako ako u formulára.*

### 3.2.6 Command button

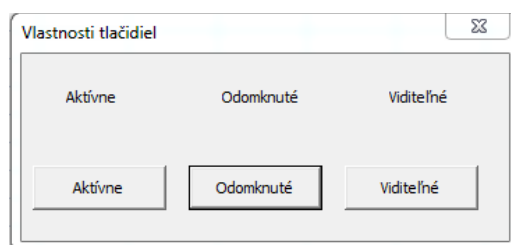
U príkazového tlačidla ide asi o najčastejšie používaný OP. Aj keď je preň možné definovať veľké množstvo rôznych udalostí, najčastejšou je udalosť **Click()**. Pokiaľ je k tejto udalosti k dispozícii procedúra, vykoná sa v momente kliknutia ľavým tlačidlom myši, poprípade stlačením klávesy enter v okamžiku, keď je OP aktívny (orámovanie po hranách).

Najzaujímavejšou vlastnosťou u tlačidla je možnosť jeho ochrany pred nechceným spustením. Tým je možné užívateľa usmerniť, že nejaké údaje nie sú v správnom tvare alebo podmienka pre jeho zaktívnenie nebola splnená a pod. Zaručuje to vlastnosť **Enabled**, ktorá je implicitne nastavená na hodnotu True, to znamená, že je možné ho stlačiť. Pri hodnote False sa zneaktívni štýlom, pri ktorom sa zmení farba písma na sivú a nie je možné ho stlačiť.

Podobnou vlastnosťou je **Locked**, ktorá umožní stlačiť tlačidlo pri oboch hodnotách ale pridelená procedúra sa vykoná len pri hodnote False. Pri najvyššej úrovni ochrany je možné použiť vlastnosť **Visible**, ktorá ho zneviditeľní.

Ako príklad uvedieme 3 tlačidlá a 3 popisky, ktorými budeme meniť jednotlivé vlastnosti tlačidiel pomocou akcie **Click()** na jednotlivé popisky:

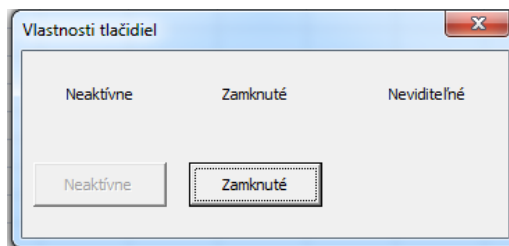
```
Private Sub lbl1_Click()  
If cmd1.Enabled = True Then  
    lbl1.Caption = "Neaktívne"  
    With cmd1  
        .Enabled = False  
        .Caption = "Neaktívne"  
    End With  
Else  
    lbl1.Caption = "Aktívne"
```



Obrázok 3.29-Vlastnosti hodnot True



```
With cmd1
    .Enabled = True
    .Caption = "Aktívne"
End With
End If
End Sub
```



Obrázok 3.30- Vlastnosti tlačidiel False

V tejto procedúre sa zisťuje, či je vlastnosť **Enabled** tlačidla cmd1 True. Ak je, tak sa zmení pri stlačení na False, názvy (**Caption**) popisku a tlačidla na

neaktívne. Ak nie je tak opačne. Táto procedúra sa týka prvého prípadu obrázku, kde meníme vlastnosť **Enabled**. V druhom prípade meníme vlastnosť **Locked** a v poslednom **Visible**. Samozrejmosťou sú zmeny mien jednotlivých mien (vlastnosť **Name**) popiskov a tlačidiel. V druhom prípade cmd2, lbl2 a v treťom cmd3, lbl3.

Pri väčšom počte tlačidiel vo formulári je niekedy pre nového užívateľa nevyhnutné použiť nejaký popis k tlačidlám. Na to slúži vlastnosť **ControlTipText**. Ide vlastne o text ktorý sa zobrazí v bubline, ak sa dlhšie podrží kurzor myši nad daným OP. [12] Podobným OP je prvok **ToggleButton** popísaný v kapitole 3.2.16.



*Pre usmernenie užívateľa, vykonať nejaký úkon, aby sa zaktivovalo tlačidlo je najlepšie používať vlastnosť **Enabled**, kde užívateľ dokáže odlišiť voľným okom hodnotu tejto vlastnosti.*

### 3.2.7 TextBox

Textové pole stojí vo VBA ako na strane vstupnej tak aj na výstupnej. Na strane vstupnej nám slúži pre pohodlné zadávanie hodnôt a na výstupnej pre ich zobrazovanie. Pokiaľ ich použijeme pre zobrazovanie, poskytnú nám oproti popiskom možnosť označiť časť textu a uložiť ho do schránky[13].

Základnou vlastnosťou textového poľa je **Value**, do ktorej sa ukladá vstup z textového pola. Táto vlastnosť je totožná s vlastnosťou **Text**. Pre získanie tejto hodnoty sa používa klasické stromové odkazovanie (napr. txt.Value) alebo si vytvorí premennú, kam sa to bude ukladať. Pri práci s číslami je dôležité si uvedomiť, že textové pole je vo formáte textu, čo znamená, že sa to musí skonvertovať na číslo, na čo existuje šikovní spôsob. Namiesto txt1.Value tam funkciu **Hodnota** v tvare Val(txt.text). Čo by sa dalo doslovne preložiť, že vyťahujeme value (hodnotu) z textového poľa.

Ako jediný prvok disponuje vlastnosťou **PasswordChar**. Pokiaľ sa do nej nastaví nejaký znak, bude sa tento znak zobrazovať namiesto skutočne zadávaného textu. Je to využiteľné v prípade zadávania hesiel alebo iných citlivých údajov, bez nebezpečia, aby sa zobrazovali na monitore v čitateľnom tvare [14].



Ako príklad sa uvedie vyplňanie informácií o osobe s následnou kontrolou správnosti zadávaných dát (obr. 3.31) :

```
Private Sub cmdOK_Click()  
If Trim(txtMeno.Value) = "" Or  
Trim(txtPriezvisko.Value) = "" Or Trim(txtUlica.Value) = "" Or_  
(txtCislo.Value) = "" Or Trim(txtMesto.Value) = "" Then  
    MsgBox "Zabudli ste vyplniť údaje"  
ElseIf Not IsNumeric(txtCislo) Then  
    MsgBox "Ulica je číselný údaj"  
Else  
    MsgBox "Vaše údaje boli uložené"  
    frm.Hide  
End If  
End sub
```

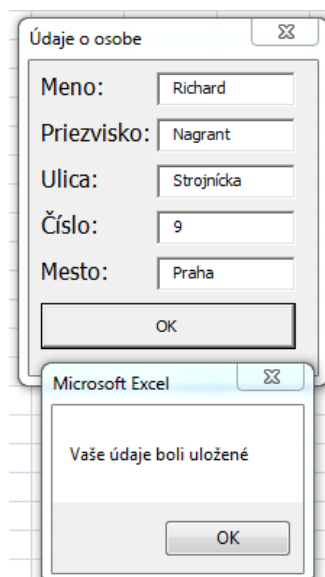
V horeuvedenej procedúre sa na začiatku príkazu If overuje obsah každého jedného textového poľa funkciou **Trim** (odstráni nám medzery), následne kontroluje negovanou funkciou **IsNumeric**, či údaj ktorý sa nachádza pri položke číslo je skutočne číslo. Chyby sú ošetrené jednotlivými správami. Ak je všetko správne vyplnené, vyskočí okno so správou a formulár sa skryje (frm.Hide).

Vlastnosť nevyhnutná pre prácu v praktickej časti a v bežnom živote je **ControlSource**, tzv. kontrolovaný zdroj. Je to akési prepojenie textového poľa vo formulári s nejakou bunkou v liste. Po ich prepojení (zapísanie názvu hárku s výkričníkom a súradnicou bunky), ak sa niečo zapíše do textového poľa, automaticky sa to zapíše do bunky, pokiaľ to následne nevymažeme vo vlastnosti a naopak.

Hodnota True **MultiLine** a **Wordwrap** v prvom prípade umožní viacriadkový vstup a v druhom automatické zalamovanie riadkov. Vlastnosť **MaxLength** umožní obmedziť užívateľa pre vloženie určitého počtu znakov (nastaví maximálnu dĺžku). Pri písaní dlhších viacriadkových textov do textového poľa je priam nevyhnutné aktivovať posuvníky. Zabezpečuje to vlastnosť **ScrollBars**, ktorá nadobúda 4 hodnoty. Pri hodnote 0 nie sú zobrazené žiadne, pri 1 sa pripojí len horizontálny, pri 2 len vertikálny a pri 3 oba.



*Pri viacerých textových poliach, napr. pri vyplňovaní nejakých formulárov, je dobré mať nastavenú hodnotu **TabIndex** v súlade s nimi. Tým je možné rapídne urýchliť vyplňanie textových polí prepínaním sa klávesou Tab.*



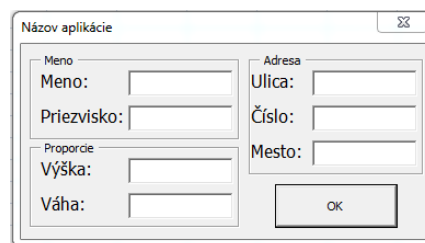
Obrázok 3.31- TextBox príklad

### 3.2.8 Frame

Rámčeky disponujú čisto estetickým charakterom využitia. V prípade šikovného použitia príkazu for je možné ho použiť aj na stránku funkčnú, pri ktorej je možné meniť vlastnosti viacerých OP, ktoré sa

nachádzajú v danom rámečku. Hlavným prínosom rámečkov je vylepšenie vzhľadu formulára a orientácie na ňom. Základnou vlastnosťou je **Caption**, čo je názov v záhlaví rámečku.

Ako príklad estetického využitia sa uvedie vyplnenie predošlých údajov zo stati textového poľa. Ako vidieť na obr. 3.32, údaje sú oddelené podľa logickej spojitosti do rámečkov Meno, Adresa a Proporcie.

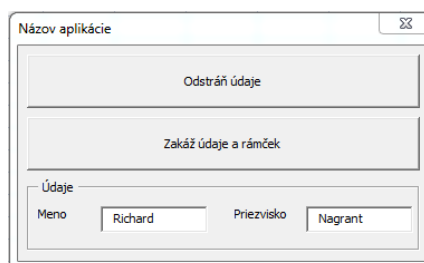


Obrázok 3.32- Frame vzhľad

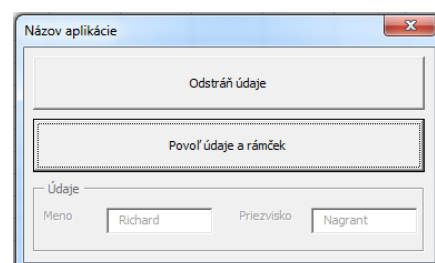
Čo sa týka funkčného charakteru, vysvetlí sa to na príklade, v ktorom je nevyhnutné použitie príkazu **For**:

```
Private Sub cmdOdstran_Click()  
txtMeno.text = ""  
txtPriezvisko = ""  
End Sub  
Private Sub cmdPovol_Click()  
Dim op As Control  
frame1.Enabled = Not frame1.Enabled  
For Each op In frame1.Controls  
op.Enabled = Not op.Enabled  
Next  
If cmdPovol.Caption = "Zakáž údaje a rámček" Then  
cmdPovol.Caption = "Povoľ údaje a rámček"  
Else  
cmdPovol.Caption = "Zakáž údaje a rámček"  
End If  
End Sub
```

Na konkrétnom príklade sa nachádzajú dve tlačidlá, s ktorými v jednom prípade mažeme textové polia a v druhom deaktivujeme všetky OP v rámečku. Výhodou je zrýchlenie pomocou príkazu **for**, ktorý v danom momente mení vlastnosti všetkých OP v uvedenom kontajneri (frame).



Obrázok 3.33- Frame funkčnosť 1



Obrázok 3.34- Frame funkčnosť 2



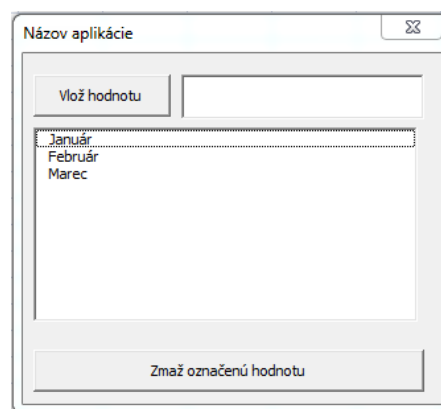
Zaujímavosťou je zmena vlastnosti *BorderStyle* z *fmBorderStyleNone* na *fmBorderStyleSingle* a späť. Tým sa totálne zneviditeľní rámček a je možné využívať jeho funkcionalitu bez viditeľnosti do formulári.

### 3.2.9 ListBox

Zoznam je akési pole riadkov, ktoré sa používa pre pohodlné zobrazovanie dát a pre prácu s nimi. Okrem všeobecných vlastností, ktoré sa vyskytujú u všetkých prvkov, disponujú aj inými zaujímavými vlastnosťami. Tie dáta je možné vložiť dvoma spôsobmi. Buď použitím vlastnosti **RowSource** alebo ručným pridaním v procedúre.

Čo sa týka ručného zadávania dát, to prebehne v procedúre inicializácie formulára, čo je jedinou možnosťou, vyhnúť sa vykonaniu istej udalosti. Tzn., že pri zobrazení formulára už budú položky v zozname (obr. 3.35) viditeľné:

```
Private Sub UserForm_Initialize()
With lst1
.AddItem ("Január")
.AddItem ("Február")
.AddItem ("Marec")
End With
End Sub
Private Sub cmdVloz_Click()
Dim hodnota As String
hodnota = txt.text
lst1.AddItem (hodnota)
txt.Text = ""
End Sub
Private Sub cmdZmaz_Click()
If lst1.Value = lst1 Then
lst1.RemoveItem (lst1.ListIndex)
Else
MsgBox "Prosím označte položku zoznamu na odstránenie", vbOKOnly + vbCritical, "Chyba"
End If
End Sub
```



Obrázok 3.35- ListBox ručne

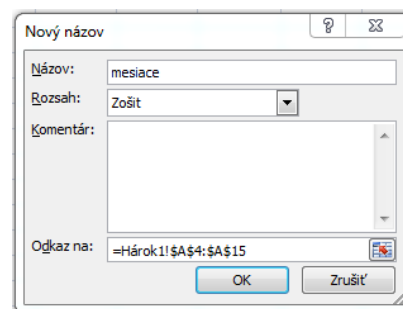
V procedúre na obr. 3-35 sa automaticky vložia hodnoty do zoznamu (počas inicializácie). Textovým poľom a následným potvrdením tlačidlom je možné ručne vložiť položku do zoznamu. Následne je možné ju označením a stlačením tlačidla "odstrániť" odstrániť. Každá položka v zozname disponuje vlastnosťou **ListIndex**, pomocou ktorej daný označený záznam odstránime. Pre prvý záznam (položku) je hodnota tejto vlastnosti 0, takže je nutné ju vždy zvýšiť o 1.

Vlastnosť **RowSource** je založená na prepojení istých buniek (napr. Hárrok1!A4:A15) v hárkoch alebo predefinovaných názvov oblastí (napr. mesiace, obr. 3.36 a následne obr. 3.37). Lepším riešením sú predefinované názvy, keďže pri vkladaní nových položiek, ktoré vložíme niekam medzi bunky názvu, automaticky sa zahrnú do daného názvu. Tzn., že sa nemusí nanovo vytvárať názov. Stačí vytvoriť predefinovaný názov v Exceli a následne tento

3	
4	január
5	február
6	marec
7	april
8	máj
9	jún
10	júl
11	august
12	september
13	október
14	november
15	december
16	

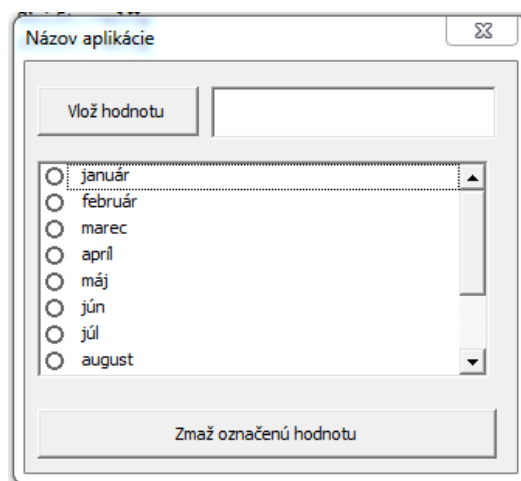
Obrázok 3.36- Označenie hodnôt

názov (pás kariet Vzorce => Definovať názov) uviesť ako hodnotu pri vlastnosti **RowSource**. V tomto prípade, použitím predchádzajúceho príkladu sa užívateľ vyhne procedúre inicializácie formulára. Pri použití tejto metódy nie je možné ďalšie pridávanie hodnôt pomocou metódy **AddItem**. V tomto prípade je pridávanie a odstraňovanie hodnôt o trochu zložitejšie. Pri pridávaní sa po stlačení tlačidla nad posledný záznam v názve pridá prázdny riadok a následne sa tam vloží hodnota z textového poľa. Ak by sa v textovom poli



Obrázok 3.37- Názov mesiace

nenachádzalo nič, tak by sme ukončili procedúru (:Exit Sub), doslovne by sme vyskočili z celej procedúry a nič nasledovné by sa nevykonalo. Pri odstraňovaní stačí označiť hodnotu na odstránenie a stlačiť tlačidlo. Všeobecne vlastnosť **ListIndex** slúži pre odkazovanie na zvolenú hodnotu. Okno so správou sa nachádza v príkaze **With**, pretože ak by ostal na konci, hodnota by už bola odstránená, a nebolo by možné ju už použiť. Konštanta **vbNullString** je prázdny reťazec a ekvivalentom ku "".



Obrázok 3.38- Listbox plnený RowSource

```
Private Sub cmdVloz_Click()
Dim hodnota As String
If Trim(txt) = "" Then MsgBox "Nemôžeme vložiť prázdnu hodnotu": Exit Sub
hodnota = txt.text
Sheets("Hárok1").Select
Range("mesiace").Select
Selection.End(xlDown).Select
Selection.EntireRow.Insert
Selection = hodnota
MsgBox "Hodnota " & hodnota & " bolá vložená"
txt.text = ""
End Sub
Private Sub cmdZmaz_Click()
Dim rozsah As String
With lst1
    rozsah = .RowSource
    MsgBox "Hodnota " & Range(rozsah).Cells(lst1.ListIndex + 1, 1)_
    & " bola odstránená"
    Range(rozsah).Cells(.ListIndex + 1, 1).Delete Shift:=xlUp
    .RowSource = vbNullString
    .RowSource = rozsah
End With
lst1.Selected(lst1.ListCount - 1) = True
End Sub
```

Medzi vlastnosti, o ktorých tu nebola zmienka, patrí **Multiselect**, ktorej hodnotu v predošlých príkladoch sme mali nastavenú na fmMultiSelectSingle. Ak by sa zmenila na fmMultiSelectMulti, tak by bolo možné vybrať viacero hodnôt zo zoznamu. Medzi obrázkami 3.35 a 3.38 je taktiež rozdiel vo vlastnosti **ListStyle** kde na obrázku 3.35 je nastavená hodnota fmListStylePlain a označenie položky sa prejaví modrým podfarbením. Pri obrázku 3.40 je hodnota nastavená na fmListStyleOption, to sa prejaví jednotlivými kruhmi vedľa položiek, ktoré sa taktiež vyfarbia a celý riadok podfarbí na modro, pri označení.

Vlastnosť **Selected** vracia, poprípade udáva stav výberu jednotlivých položiek zoznamu, pričom ako parameter využíva hodnoty ich indexov [15]. Obdobnou vlastnosťou je **ControlSource**, do ktorej stačí vložiť ľubovoľnú bunku v hárku a bude sa tam ukladať hodnota **Value** zo zoznamu.

Pri tomto OP je možné zobrazit' viacero stĺpcov. Zabezpečuje to vlastnosť **ColumnCount**, kde stačí priradiť hodnotu v podaní počtu stĺpcov. Je možné k týmto stĺpcom pridať aj hlavičky, to znamená že sa vlastnosť **ColumnHeads** nastavíme na True a pomocou kódu pridáme názvy. Pri dlhších názvoch je možné manuálne zvolit' šírky stĺpcov (**ColumnWidths**). Pri používaní viacerých stĺpcov sa na označenú hodnotu (vlastnosť `listbox.value`) implicitne načíta hodnota z prvého stĺpca, čo sa dá ovplyvniť vlastnosťou **BoundColumn**, kde je možné nastaviť požadovaný stĺpec číslom v poradí.



*Ako jediný OP dokáže plnohodnotne zobrazit' data vo viacerých stĺpcoch a riadkoch. Atraktívnou vlastnosťou je MultiSelect, no na druhej strane absentuje vlastnosť pre obnovenie zoznamu (pri plnení pomocou vlastnosti RowSource).*

### 3.2.10 ComboBox

Tento prvok, ktorého slovenským ekvivalentom je pole so zoznamom, pochádza z istej kombinácie vlastností a funkčnosti prvkov Zoznamu (ListBox) a Textového poľa (TextBox). Jeho základným rysom je vzhľad ako textové pole, na ktorého konci sa nachádza trojuholník, po ktorého stlačení sa rozbalí zoznam s položkami. Hlavným rozdielom oproti ListBoxu je, že tento OP nedisponuje možnosťou **MultiSelect**.

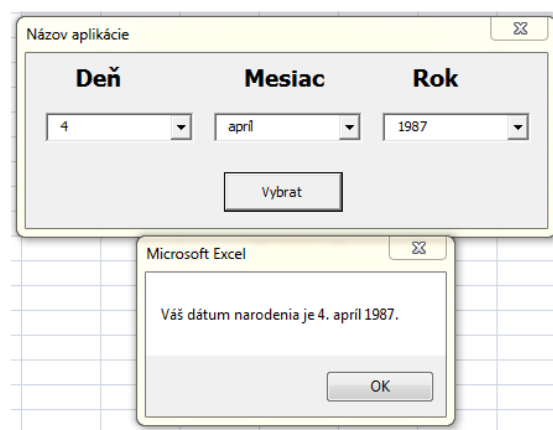
Napíňa sa rovnako ako ListBox (obr. 3-35 a 3-38), tiež môže obsahovať viac stĺpcov. V tom prípade sa predpokladá nastavenie hodnoty **ColumnCount** na požadovaný počet a následne **BoundColumn**, ktorá nám určí z ktorého stĺpca sa nám budú vyberať hodnoty, ktoré naplnia vlastnosť Value [16].

Vlastnosť **ListRows** nám udáva koľko riadkov položiek je možné zobrazit' , ktorá je implicitne nastavená na max. 8. K tejto vlastnosti je logicky spojené obmedzenie na zoznam, tzn. že užívateľovi nepovolí vložiť hodnoty mimo zoznam. To zabezpečí hodnota `fmStyleDropDownList` vlastnosti **Style**.

Najpoužívanejšou vlastnosťou, tak ako aj u textového poľa, je **Value**, ktorá vracia vybranú hodnotu z poľa so zoznamom.

Ako príklad sa uvedie klasický výber z 3 prvkov pre určenie dátumu narodenia (obr. 3-39). Na začiatku sa vytvoria názvy pre jednotlivé OP (dni, mesiace, roky). Následne sa vložia do vlastností

**RowSource** každého OP. V tomto príklade sa v okne so správou kód odkazuje na jednotlivé vlastnosti **Value** ovládacích prvkov.



Obrázok 3.39- Výber dátumu narodenia

```
Private Sub cmdVybrat_Click()  
MsgBox "Váš dátum narodenia je " & _ cmbDen.Value & ". " & cmbMesiac.Value & " " &  
cmbRok.Value & "."  
End Sub  
Private Sub UserForm_Initialize()  
cmbDen.Style = fmStyleDropDownList  
cmbMesiac.Style = fmStyleDropDownList  
cmbRok.Style = fmStyleDropDownList  
End Sub
```



Vo všeobecnosti je používanie poľa so zoznamom u užívateľov obľúbenejšie, a to z dôvodu, že nezaberá tak veľa miesta na formulári ako listBox. Najčastejšou chybou u užívateľov je, že nenastavia výber na zoznam, tzn. že comboBox je editovateľný. Tomu sa vyhneme možnosťou `fmStyleDropDownList` vlastnosti `Style`. Pre výber dátumu narodenia je najvhodnejšie používať OP kalendár, popísaný v stati 3.2.18.

### 3.2.11 OptionButton a CheckBox

OP prepínač sa zvykne využívať v prípadoch, ak sa potrebuje vybrať jedna možnosť z viacerých (skupina prvkov). Napr. vybrať pohlavie a pod. Prepínače sa v rámci skupiny vylučujú, tzn. že nemôžeme vybrať z jednej skupiny viac ako 1 prepínač, pri vybratí prvého a následne druhého sa prvý deaktivuje. Vzájomné vylučovanie prepínačov sa týka výlučne jedného kontajnera, ktorým sa myslí objekt, ktorý môže obsahovať iné objekty (formulár, rámček). Príslušnosť ku skupine prvkov nastavíme pomocou vlastnosti **GroupName** tak, že zapíšeme rovnakú hodnotu u všetkých tlačidiel.

Aktívnosť prepínača sa zistí podľa ikonky, ktorá môže byť vyplnená. Vtedy je prvok aktívny, tzn že jeho vlastnosť **Value** je `True`, v opačnom prípade je `False`. Vhodným spôsobom ako zoskupiť jednotlivé prepínače sú rámčeky, ktorými je možné aj vhodne pomenovať účel týchto prepínačov. Ako u väčšiny prvkov býva zvykom, aj tu sa nachádza vlastnosť **ControlSource**, ktorou je možné prepojiť prepínač spoločne s bunkou v hárkoch. Do tejto bunky sa ukladá hodnota PRAVDA, či NEPRAVDA, v závislosti od prvku, či je vybraný alebo nie [17].

Čo sa týka **zaškrťavacieho tlačidla**, tie slúžia k výberu viacerých možností. Oproti prepínačom, ktoré majú ikonku guľatú, prepínače majú štvorce. Pri klepnutí na nevybrané tlačidlo sa jeho hodnota nastaví na `True`, čo sa pozná zaškrtnutím štvorca pri tlačidle. Po ďalšom kliknutí sa odškrtnie, hodnota zaškrťavacieho tlačidla sa zmení na `False`. Aktuálnu hodnotu reprezentuje vlastnosť **Value** a taktiež funkčnosť vlastnosti **ControlSource** funguje úplne rovnako ako v prípade prepínača.

Ako príklad sa uvedie výpočet mzdy zamestnanca:

```
Dim sadzba As Integer  
Private Sub UserForm_Initialize()  
optOperator.Value = True  
cmbMeno.Style = fmStyleDropDownList  
lblCP.TextAlign = fmTextAlignCenter  
lblDan.TextAlign = fmTextAlignCenter  
lblHP.TextAlign = fmTextAlignCenter  
With cmbMeno
```

```
.AddItem "Richard Nagrant"
.AddItem "Daniel Host"
End With
End Sub
Private Sub cmdPotv_Click()
If Val(txtPH.text) > 744 Then
    MsgBox "Nie je realne aby mal mesiac viac ako 31 dni"
Else
    If Trim(cmbMeno.Value) = "" Or Trim(txtPH.text) = "" Then
        MsgBox "Zabudli ste vyplniť údaje"
    ElseIf Not IsNumeric(txtPH) Then
        MsgBox "Počet hodín je číselný údaj"
    ElseIf chkStudent.Value = True Then
        lblHP.Caption = Val(txtPH) * sadzba
        lblDan.Caption = 0.1 * Val(lblHP.Caption)
        lblCP.Caption = Val(lblHP.Caption) - Val(lblDan.Caption)
    Else
        lblHP.Caption = Val(txtPH) * sadzba
        lblDan.Caption = 0.3 * Val(lblHP.Caption)
        lblCP.Caption = Val(lblHP.Caption) - Val(lblDan.Caption)
    End If
End If
End Sub
Private Sub chkStudent_Click()
If chkStudent.Value = True Then
    frmMzda.Caption = "Študentská mzda"
Else
    frmMzda.Caption = "Mzda"
End If
End Sub
Private Sub optAnalytik_Click()
sadzba = 120
End Sub
Private Sub optManual_Click()
sadzba = 60
End Sub
Private Sub optOperator_Click()
sadzba = 100
End Sub
```

Ako je vidieť, na začiatku kódu sa nachádza deklarácie globálnej premennej z dôvodu použitia vo viacerých procedúrach. Po spustení formulára prebieha jeho inicializácia, kde sa manuálne pridávajú osoby do poľa so zoznamom, nastaví sa nutnosť výberu zo zoznamu a text v popiskoch sa zarovná na stred. Na to prebieha klasická kontrola, či sú polia vyplnené, resp. či náležité pole obsahuje čísla. Nakoniec prebiehajú výpočty, najprv ak je osoba študentom a nakoniec ak nie je. V posledných 3 procedúrach sa priradujú hodinové sadzby do premennej sadzba, v závislosti, ktorý prepínač je aktívny.



*Pri zoskupovaní tlačidiel pomocou vlastnosti GroupName, nastáva niekedy u užívateľov chaos v ich názvoch, preto sa odporúča používať rámčeky pre ich vyčlenenie. Tie sú na rozdiel od vlastnosti GroupName aj viditeľné vo formulároch.*

Obrázok 3.40- Mzda

Obrázok 3.41- Študentská mzda



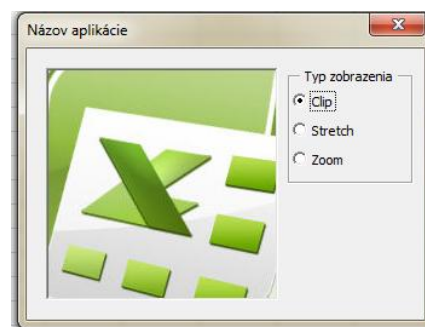
## 3.2.12 Image

OP typu Image slúži na zobrazovanie obrázkov vo formulári. Po pridaní OP typu Image do formulára a následným kliknutím na vlastnosť **Picture** môžeme vybrať požadovaný obrázok a načítať ho do OP. Tento obrázok sa stane súčasťou excelovského súboru. Podporované sú takmer všetky grafické formáty (.bmp, .ico, .gif, .jpg, .jpeg a pod.)

Druhou vlastnosťou, ktorou sa pri práci s týmto OP budeme zaoberať je **PictureSizeMode**, ktorý určuje podobu zobrazenia obrázku. K dispozícii sú 3 možnosti, a to **fmPictureSizeModeClip**, **fmPictureSizeModeStretch** a **fmPictureSizeModeZoom**. Pri hodnote **Clip** sa oreže časť, ktorá presahuje OP, v ďalších dvoch prípadoch sa rozmery obrázku upravujú tak aby sa vošli do OP. V prípade **Stretch** dochádza pri veľkých obrázkoch ku skresleniu a v prípade **Zoom** dochádza k zachovaniu pôvodných rozmerov strán. Taktiež ku skrášleniu rámu môžeme zmeniť vlastnosť **SpecialEffect**, ktorým je možné upraviť vzhľad OP.

V príklade je uvedený vložený obrázok s prepínačmi, ktorými budeme meniť vlastnosť **PictureSizeMode**:

```
Private Sub optClip_Click()
imgExcel.PictureSizeMode = fmPictureSizeModeClip
End Sub
Private Sub optStretch_Click()
imgExcel.PictureSizeMode = fmPictureSizeModeStretch
End Sub
Private Sub optZoom_Click()
imgExcel.PictureSizeMode = fmPictureSizeModeZoom
End Sub
```

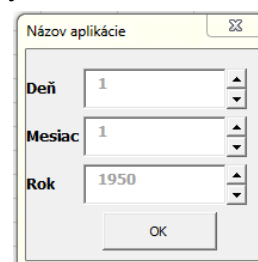


Obrázok 3.42- Image

Pri zobrazovaní obrázkov pomocou OP Image sa neodporúča používať hodnota **fmPictureSizeModeZoom** vlastnosti **PictureSizeModeClip**, pretože sa odreže tá časť obrázku, ktorá presahuje okraj. Preto je vhodné prispôbiť rozmery OP Image zobrazovanému obrázku. Na prepínanie typu zobrazenia sa používajú len prepínače, z dôvodu vzájomného vylúčovania vo rámečku.

## 3.2.13 Spin button

Prvok typu číselník (obr. 3-43) slúži pre pohodlné zvyšovanie, či znižovanie číselných hodnôt zadávaných myšou. Skladá sa z 2 tlačidiel, ktoré sú osadené šípkami v opačných smeroch. Hlavná vlastnosť **Value** sa väčšinou prepája s textovým poľom, do ktorého vlastnosti **Value** sa zobrazuje vlastnosť **Value** číselníka. Môžeme taktiež nastaviť spodnú hranicu **Min** (minimálna hodnota vlastnosti Value) a hornú hranicu **Max** (maximálna hodnota vlastnosti Value), ktoré sa nachádzajú v rozmedzí dátového typu **Long**. Orientácia číselníka sa dá prepnúť vo vlastnosti **Orientation**. Implicitné zvyšovanie hodnoty je nastavené na 1. Pre jeho zmenu použijeme vlastnosť **SmallChange**. Samozrejmosťou je



Obrázok 3.43- Číselník



možnosť prepojiť číselník s bunkou v hárkoch pomocou vlastnosti **ControlSource**, do ktorej hodnoty sa zadá cesta k bunke (napr. Hárrok1!A1) [18]

```
Private Sub UserForm_Initialize()  
    spnDen.Min = 1  
    spnDen.Max = 31  
    spnMesiac.Min = 1  
    spnMesiac.Max = 12  
    spnRok.Min = 1950  
    spnRok.Max = 2000  
End Sub  
Private Sub cmdOK_Click()  
    MsgBox "Váš dátum narodenia je : " & txtDen.text & "." & txtMesiac.text & "." & _  
    txtRok.text  
End Sub  
Private Sub spnDen_Change()  
    txtDen.text = spnDen.Value  
End Sub  
Private Sub spnMesiac_Change()  
    txtMesiac.text = spnMesiac.Value  
End Sub  
Private Sub spnRok_Change()  
    txtRok.text = spnRok.Value  
End Sub
```

V procedúre inicializácie formulára sa nastavujú minimálne a maximálne hodnoty jednotlivých číselníkov. V posledných 3 procedúrach sa vkladajú vlastnosti Value číselníkov do príslušných textových polí.



*Celkovo je tento typ OP veľmi užitočný najmä u užívateľov, ktorí neradi prekladajú ruku z myši na klávesnicu, tzn. že dokážu pohodlne zadať hodnotu pomocou myši. Vlastnosti jednotlivých minim a maxim by sa mali načítavať v procedúre inicializácie formulára.*

### 3.2.14 ScrollBar

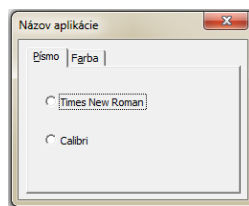
Posuvník je veľmi podobný panelom, ktoré sa zobrazujú v rôznych OP, aby bol viditeľný celý obsah OP. V porovnaní s číselníkmi je tu len jeden rozdiel a to je bežec medzi tlačidlami so šípkami. Umiestnenie tohto bežca vypovedá aktuálnej hodnote medzi **Min** a **Max**. Pre posun vyšší než je nastavená vlastnosť **SmallChange** je tu vlastnosť **LargeChange**, ktorá sa zaktívuje uchopením bežca.

### 3.2.15 MultiPage a TabStrip

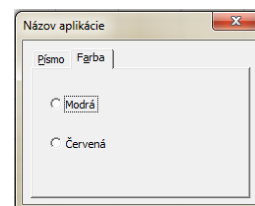
MultiPage, teda viacnásobná stránka, sa po formulári (UserForm) používa ako druhý najčastejší kontajner pre ovládacie prvky. Využívame ho taktiež pre lepšie usporiadanie prvkov, podobne ako rámček, len na vyššej úrovni. Môže obsahovať niekoľko stránok, pričom každá z nich zobrazuje iné typy OP. Vidieť sú len prvky na práve zobrazenej záložke, ktorých počet pri vložení je 2. Všeobecne sa dá o ňom povedať že poskytuje opticky lepší prehľad nad funkcionalitou celej aplikácie.

Pre pridanie stránky stačí kliknúť pravým tlačidlom do priestoru názvov stránok a z kontextového menu zvoliť nová stránka. Pre jej odstránenie je postup rovnaký. Pri premenovaní danej stránky stlačíme možnosť premenovať (rename), kde sa nám naskytnú nové možnosti.

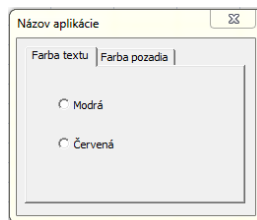
Vlastnosť **ControlTipText**, čo je bublina pri prechode myšou nad názvom stránky. Zvyšnou možnosťou je klávesová skratka, v spolupráci s klávesou **alt** a následne užívateľsky definovanou. Zároveň ak sa použije písmeno ktoré obsahuje daný názov, podčiarkne sa. Pre nastavenie aktívnej (stránky, ktorá sa má zobrazit', napr. pri spustení) stránky, sa používa vlastnosť **Value**, ktorej hodnoty začínajú od 0 pre prvú stránku (Page1). Taktiež je možné nastaviť umiestnenie názvov záložiek pomocou vlastnosti **TabOrientation**. V tomto prípade hodnota tejto vlastnosti je **fmTabOrientationTop**, kde medzi ďalšie možnosti patria **\*Right**, **\*Left** alebo **\*Bottom**. [19].



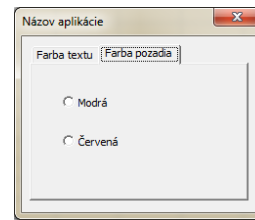
Obrázok 3.44- MultiPage 1



Obrázok 3.45- MultiPage 2



Obrázok 3.46- Tabstrip 1



Obrázok 3.47- Tabstrip 2

Ako je vidieť na obrázkoch 3.44 a 3.45, na oboch kartách sú rozdielne OP, ktoré sú súčasťou jednotlivých stránok.

Čo sa týka prvku Záložka (Tabstrip), slúži k vykonaniu rôznych operácií s rovnakými ovládacími prvkami, v závislosti na výbere záložky. Zaujímavou vlastnosťou je orientácia jednotlivých záložiek pomocou vlastnosti **TabOrientation**. Na rozdiel od viacnásobnej stránky tento prvok nie je vlastným kontajnerom, tzn. že nemôže mať vlastné ovládacie prvky. Ako príklad na obr. 3.46 a 3.47 môžeme vidieť, že aj napriek zmene záložky, OP ostali tie isté. To je z dôvodu že OP (prepínače) sú viazané na formulár. Viditeľné by to bolo pri presunutí záložiek.

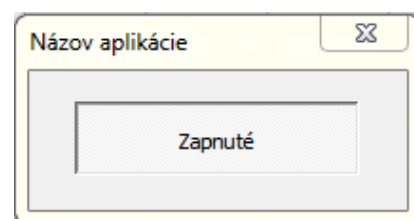


*Pre lepšiu spoluprácu, vzhľad a funkčnosť je lepšie používať ako základný kontajner viacnásobnú stránku.*

### 3.2.16 ToggleButton

Prepínacie tlačidlo je akýmsi krížencom príkazového tlačidla, odkiaľ pochádza jeho vzhľad a zaškrtnutého tlačidla, odkiaľ pochádza jeho funkcionálnosť. V základnom tvare je hodnota jeho vlastnosti **Value** **False** a vyzerá ako akýkoľvek iné príkazové tlačidlo. Po jeho stlačení sa vlastnosť **Value** zmení na **True** a tlačidlo ostane stlačené, následným stlačením sa vráti do pôvodného stavu. Taktiež obsahuje vlastnosť **ControlSource**, tzn. že dokáže do bunky uložiť hodnotu **PRAVDA** alebo **NEPRAVDA** v závislosti na stave tlačidla.

```
Private Sub tgl_Click()  
If tgl.Caption = "Vypnuté" Then  
    tgl.Caption = "Zapnuté"  
Else  
    tgl.Caption = "Vypnuté"  
End If
```



Obrázok 3.48- Stlačené prepínacie tlačidlo

```
End Sub
Private Sub UserForm_Initialize()
    tgl.Caption = "Vypnuté"
End Sub
```

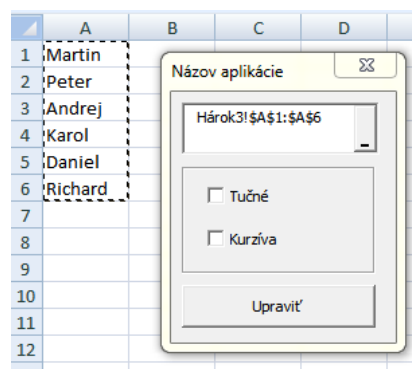


Použitie tohto prvku pri formulároch nie je najvhodnejšie a to hlavne zo vzhľadovej stránky, lepšie je použitie zaškrťavacieho tlačidla, pri ktorom je jasne viditeľná jeho hodnota. Prepínacie tlačidlo sa skôr zvykne používať v hárkoch, napr. pre zapínanie filtra, skrývanie riadkov a pod.

### 3.2.17 RefEdit

Z názvu plyní účel tohto OP. Názov je zložený z 2 slov "Reference Edit", čo v preklade znamená upraviť odkaz. Prostredníctvom tohto prvku je možné vybrať pomocou myši alebo klávesnice z hárku oblasť a následne ju získať v textovej podobe ako adresu.[20].

```
Private Sub cmdUpravit_Click()
    Call upravOblast
End Sub
Sub upravOblast()
    Dim refRozsah As Range
    If refRozsah Is Nothing Then Exit Sub
    Set refRozsah = Range(RefEdit.Value)
    If chkbTucne Then
        refRozsah.Font.Bold = chkbTucne
    Else
        refRozsah.Font.Bold = False
    End If
    If chkbKurziva Then
        refRozsah.Font.Italic = chkbKurziva
    Else
        refRange.Font.Italic = False
    End If
End Sub
```



Obrázok 3.49- Úprava písma výberu

V procedúre cmdUprav sa vykonáva volanie procedúry upravOblast, tzn. že túto procedúru by mohli využívať aj iné tlačidlá. V deklarácii sa definuje premenná ako objekt Range. Kľúčové slovo Set sa využíva pri naplňovaní objektov. Do konkrétneho objektu refRozsah sa naplnil rozsah vlastnosti Value OP refEdit (tzn. to čo sa označí, s tým je možné narábať). V tomto prípade sa mení vlastnosť Font.Bold a Font.Italic objektu refRozsah.



Ovládací prvok použijeme v prípade keď potrebujeme užívateľovi dať možnosť zadať odkazy na oblasť buniek v hárku. Čo sa týka jeho funkčnosti, tá na vysokej úrovni, keďže pre vyznačenie údajov pre napr. editáciu, využíva objekt typu Range.

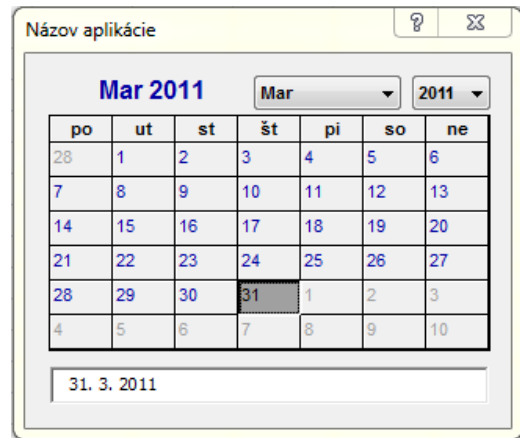
### 3.2.18 Calendar

OP typu kalendár sa nenachádza medzi základnými OP MS Forms, čo je škoda, pretože dokáže ušetriť veľa práce. Pridá sa kliknutím pravým tlačidlom na panel ToolBox a následne sa zvolí možnosť "Additional controls". Následným výberom prvku Kalendár zo zoznamu a potvrdením sa vloží do ponuky.

Je to akési dialógové okno, prostredníctvom ktorého je možné pohodlne vybrať dátum z ponuky. Vlastnosť **Value** nám vyjadruje dátum vyznačený v kalendári. Preto je vhodné v inicializácii formulára nastaviť pre vlastnosť **Value** hodnotu `Date()`, čo je zabudovaná funkcia Excelu, ktorá nastaví v kalendári aktuálny dátum. Vlastnosťou **FirstDay** si nastavíme prvý deň zobrazovaný v kalendári. U nás je to zrejme hodnota `Po`.

```
Private Sub cal_Click()  
    TextBox1.text = Calendar1.Value  
End Sub
```

```
Private Sub frm_Initialize()  
    Calendar1.Value = Now()  
    TextBox1.text = Calendar1.Value  
End Sub
```



Obrázok 3.50- Kalendár



Pri tomto OP nie je veľa toho čo nastavovať, jedine nastaviť aktuálny dátum ako hodnotu `Value` kalendára. Pôsobí celkovo príjemným dojmom, pomocou 2-3 klikov je možné nastaviť požadovaný dátum.

### 4 Praktická časť

V praktickej časti sa bude aplikovať použitie určitých OP pre prácu s hárkami zošita na konkrétnom príklade. Zúročia sa tu všetky nadobudnuté znalosti z teoretickej časti, pridajú sa ešte nejaké špecifické udalosti, objekty a pod. Užívateľské rozhranie sa bude navrhovať s použitím pravidiel maximálnej funkčnosti pomocou najjednoduchšieho ovládania. Táto vytvorená aplikácia by mala predovšetkým užívateľom uľahčiť prácu, tzn. že užívateľ by sa mal vedieť intuitívne preklikať medzi jednotlivými oknami.

#### 4.1 Popis

V tomto príklade sa bude vytvárať aplikácia pre správu mobilných zariadení (telefónov). Celková funkčnosť aplikácia je založená na spoločnej interakcii ovládacích prvkov ActiveX a jednotlivých hárkov, v ktorých sa dlhodobé záznamy budú ukladať. Tieto zariadenia bude možné jednotlivito pridávať, či mazať. Pri pridávaní sa nastavia isté polia nutné pre vyplnenie, z dôvodu konzistencie databáze pre neskoršie výstupy v podaní kontingenčných tabuliek. Taktiež bude možné pridávať operačné systémy a značky, z ktorých bude neskôr možné vytvárať mobilné telefóny. Po ukončení pridávania či odstraňovania telefónov je možné tieto telefóny filtrovať, a to buď podľa značky alebo podľa cenového rozdelenia a následne ich exportovať do súboru typu txt. Medzi ďalšie výstupy patrí percentuálny prehľad zastúpenia jednotlivých značiek v databáze, zastúpenie počtu zariadení v cenových intervaloch a zastúpenie jednotlivých telefónov u operátorov na základe funkčnosti, podľa zadanych vlastností.

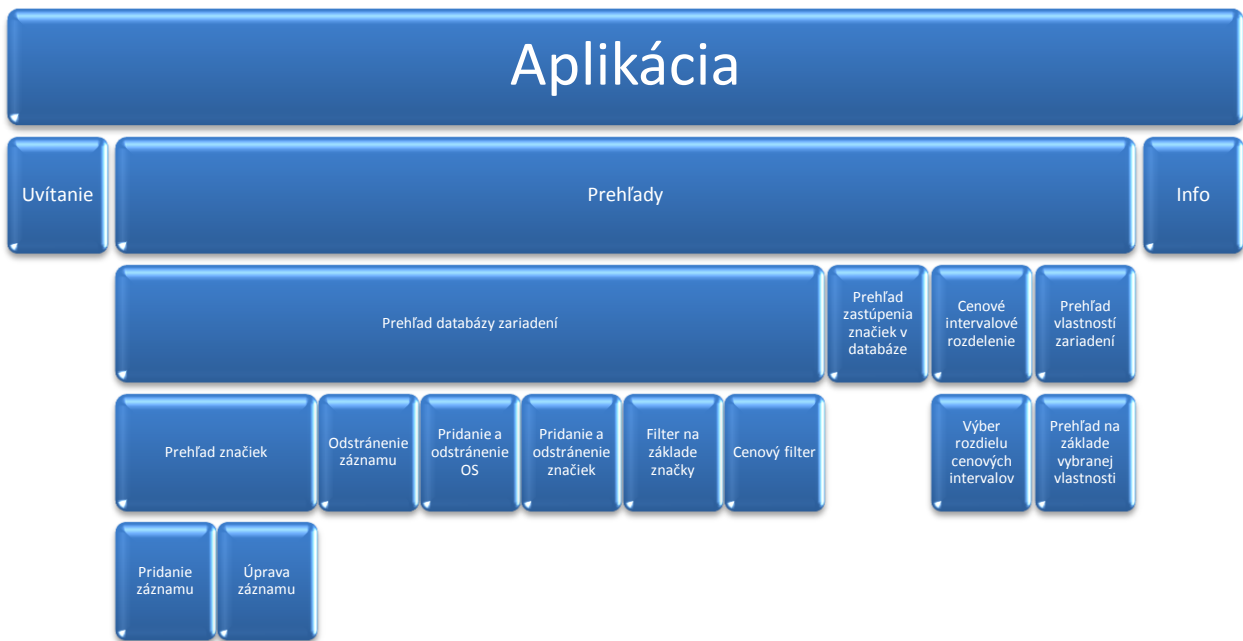
#### 4.2 Návrh

Celkový návrh užívateľského rozhrania je veľmi komplexná záležitosť. Prípadne zmeny počas tvorby aplikácie by mohli byť zbytočne komplikované. Na začiatku každého návrhu aplikácie je potrebné stanoviť si požiadavky, ktoré musí spĺňať. To znamená jasne si nadefinovať:

- ✓ účel aplikácie,
- ✓ konkrétne výstupy,
- ✓ štruktúru dát,
- ✓ konvenciu pre názvy OP,
- ✓ pomenovávanie oblastí a listov,
- ✓ štruktúru jednotlivých úrovní menu,
- ✓ skrývanie, odkrývanie formulárov,
- ✓ overenie a ochrana dát,
- ✓ vratnosť akcií,
- ✓ a pod.

Nie je potrebné vytvárať zbytočne zložité formuláre, dobré riešenia bývajú väčšinou najjednoduchšie.

Konkrétny návrh úrovni aplikácie:



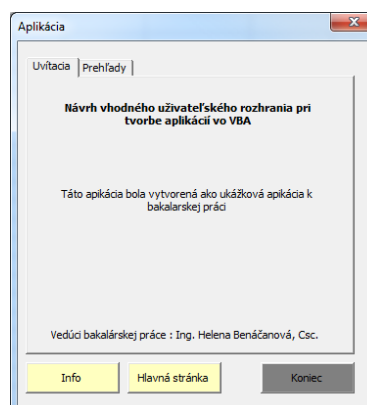
Obrázok 4.1- Návrh aplikácie

Celkovo je použitých 9 rôznych OP a 6 rôznych formulárov v celkových 8 modifikáciách.

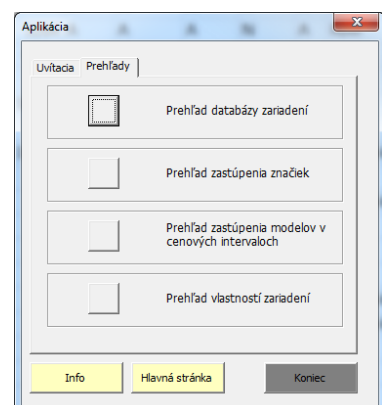
Pre spustenie aplikácie existujú dve riešenia. Prvým je nastaviť pre udalosť Workbook\_Open zobrazenie úvodného formulára alebo ponechať zobrazenie hárku, do ktorého sa vložia tlačidlá. V tejto aplikácii sa využije možnosť druhá, kde sa budú nachádzať 2 tlačidlá (ActiveX), jedno pre zobrazenie formulára a druhé pre ukončenie aplikácie. Udalosť Workbook\_Open bude obsahovať procedúru pre pozdravenie užívateľa (v závislosti na čase) a spustenie procedúry prispôbenia obrazovky.

## 4.2.1 Úvodný formulár (frmStart)

Na úvodnej formulári je zobrazený OP Multipage, ktorý obsahuje 2 karty, uvítaciu a kartu prehľadov. Pod ním sa nachádzajú 3 tlačidlá. Prvé s názvom Info je pre zobrazenie formulára frmInfo, druhé pre prechod na uvítací hárku a tretie pre zavretie formulára.



Obrázok 4.2- Uvítacia karta 1



Obrázok 4.3- Uvítacia karta 2

Prvá karta formulára má informatívny charakter, druhá obsahuje 4 tlačidlá. Prvé spúšťa formulár Prehľad (frmPrehľad), zvyšné presmerujú na hárok "KT", kde sa makrom spúšťajú príslušné prehľady v podobe kontingenčných tabuliek.

Úvodný formulár sa po vybratí akejkoľvek možnosti skrýva. Pre jednotlivé rámčeky a tlačidlá platí, že ich vlastnosti Caption sú prázdne. Hlavnou vysvetlivkou pre tlačidlo je popisok, v príslušnom rámčeku.

## 4.2.2 Informačný formulár (frmInfo)

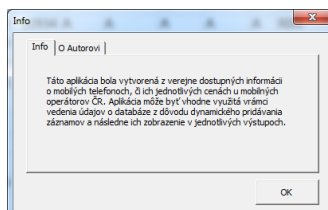
Informačný formulár obsahuje taktiež

OP multipage s dvoma kartami.

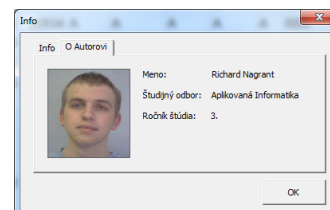
V prvej sú informácie o databáze a o pôvode dát, v druhej informácie

o autorovi. Druhá karta obsahuje aj OP Image kde je vložená príslušná fotka.

Tlačidlom OK sa daný formulár skryje.



Obrázok 4.5- Informačná karta 1



Obrázok 4.4- Informačná karta 2

## 4.2.3 Formulár prehľad (frmPrehľad)

V tomto prípade ide

o hlavný formulár

s najvyššou funkcionalitou

zo všetkých. Najväčší

priestor zaberá OP

ListBox, ktorého vlastnosti

RowSource je naplnená

oblasťou (názvom) "data".

Značka	Model	O2	Vodafone	T-Mobile	Twist	MMS	Foto	Portal	UMTS	OS	Typ OS	Min. cena
Apple	iPhone 3GS 16GB		14977	14999	10999	A	A	A	A	A	iPhone OS	14977
Apple	iPhone 3GS 32GB		16977			A	A	A	A	A	iPhone OS	16977
Apple	iPhone 4 16 GB	17295	15577	16999	17049	A	A	A	A	A	iPhone OS	15577
Apple	iPhone 4 32 GB	20395	18477			A	A	A	A	A	iPhone OS	18477
BlackBerry	8520	6695		5799	5264	A	A	A	N	A	RIM	5799
BlackBerry	8960			7199	6524	A	A	A	N	A	RIM	7199
BlackBerry	9520	13795				A	A	A	A	A	RIM	13795
BlackBerry	9700	12995		12099	10934	A	A	A	A	A	RIM	12099
CPA	Halo 4			1399	1304	N	N	N	N	N	-	1399
HTC	Desire	11495	10977	11499	10394	A	A	A	A	A	Android	10977
HTC	HD2		4977	67678		A	A	A	A	A	WM	67678
HTC	Magic					A	A	A	A	A	Android	4977
HTC	Smart	3995		5577		A	A	A	A	N	-	3995
HTC	Touch 2					A	A	A	A	A	WM	5577
HTC	Touch Pro 2		10577			A	A	A	A	A	WM	10577
HTC	Wildfire	5895	5477	5299		A	A	A	A	A	Android	5299
Huawei	U8100	3795				A	A	A	A	A	Android	3795
LG	G200	1595				A	A	A	A	N	-	1595
LG	G200 mini					A	A	A	A	N	-	
LG	G200 Pop	3295	2777	3199		A	A	A	A	N	-	2777
LG	GM205	1995	6577			A	A	A	A	N	-	1995
LG	GM360 Viewity Snap	3695		3199	2924	A	A	A	A	N	-	3199
LG	GS101		677	2199	2024	N	N	N	N	N	-	677
LG	GS290 Cookie Fresh					A	A	A	A	N	-	2199
LG	GT350 Town		3177			A	A	A	A	N	-	3177
LG	GT300			3899	3554	A	A	A	A	N	-	3899

Obrázok 4.6- Formulár prehľad

Obsahuje 13 stĺpcov dát prepojených s hárkom

"Data". Pod ním sa nachádza viacero tlačidiel. Prvý rámček "Záznamy" zľava obsahuje 2 tlačidlá. To prvé slúži pre zobrazenie formulára pre prehliadanie záznamov a pre ich pridanie. To druhé je pre ich odstránenie z listboxu a samozrejme zo zdroja v hároku "Data". Druhý rámček "Pridávanie" slúži pre správu operačných systémov a značiek v databáze. Posledný rámček "Filtre" môže nadobúdať 2 podoby. Prvou a implicitnou možnosťou je filtrovanie dát podľa značky. Druhou je možnosť filtrovania záznamov podľa cien, ktorá sa zaktívuje výberom z pola s zoznamom. Po každom filtrovaní je možné vyfiltrované záznamy vyexportovať do textového dokumentu. Posledným OP je tlačidlo OK, ktoré uzavrie tento formulár a následne sa zobrazí úvodný.



## 4.2.4 Prehľad záznamu (frmPrehliadaťTelefony)

Tento formulár slúži pre jednotlivé prehliadanie a pridávanie záznamu. Obsahuje viacero OP ako pole so zoznamom pre výber značky, či textové pole pre názov modelu. V rámci "Vlastnosti" sa nachádzajú jednotlivé polia so zoznamom, ktoré definujú vlastnosti záznamu, z pohľadu funkčnosti. Na pravej strane od rámčeka sa nachádzajú textové polia s cenami u jednotlivých operátorov. V spodnej časti sa nachádza nepomenovaný

Obrázok 4.7- Prehliadanie záznamu

rámček, ktorý zoskupuje prvky podľa logickej návaznosti. Textové pole v ňom nám dáva najavo ktorý záznam v poradí prehliadame a číselník slúži na ich prepínanie. Posledným prvkom v rámci je tlačidlo pre vytvorenie nového záznamu, po ktorého kliknutí sa vymažú všetky polia. Pre úspešné pridanie telefónu je nutné vyplniť značku, model a jednotlivé vlastnosti v rámci "Vlastnosti". Voliteľné sú textové polia operátorov, ktoré sú obmedzené len na číselné údaje. Pri vytváraní záznamu sa tlačidlo "Nový" zmení na tlačidlo "Zrušiť", po ktorého stlačení zruší vytváranie záznamu. Tlačidlom OK sa uložia všetky zmeny vykonané v jednotlivých OP, či pri prehliadaní alebo pridávaní. Pre správne zobrazovanie a vyhnutie sa chybám je potrebné aby značka a operačný systém, ktoré sú uvedené v databáze existovali aj v danej oblasti.

## 4.2.5 Pridávanie OS a značiek (frmOperacneSystemy a frmZnacky)

Tieto formuláre sú funkcionalitou a vzhľadom podobné. V oboch je možné pridať nový záznam po stlačení tlačidla pridať alebo odstrániť vybraný. V prípade operačných systémov je znemožnené odobratie položky "-", pretože to je implicitne vypisované v záznamoch, ktoré nedisponujú operačným systémom. Záznamy v oboch prípadoch sú načítavané cez vlastnosť RowSource názvami príslušných oblastí. Oba prípady sú ošetrené, aby v zozname neostal počet záznamov menší ako 3.

Obrázok 4.9- Správa OS

Obrázok 4.8- Správa značiek

## 4.3 Realizácia

V tejto časti sa popíše jednotlivo zdrojový kód celej aplikácie nachádzajúci sa v prílohe. Jednotlivé riadky kódov sú číslované v rámci sekcií, pri každom vysvetlení sa odkáže na určitý rozsah riadkov, vo formáte [číslo sekcie, číslo riadka(ov)].

### 4.3.1 Spustenie aplikácie

Po otvorení zošita je nevyhnutné pre spustenie makier prijať výstrahu, záležiac na konkrétnych nastaveniach. Následne prebehne prispôbenie okna, jednotlivo sa skryjú hárky zošita, nadpisy,



panel so vzorcom, stavový riadok, mriežka a posuvníky.[1,3-13] Po ich prijatí sa zobrazí správa s pozdravom, ktorá podľa času určí pozdrav a k tomu pripojí aj meno užívateľa[1,15-23]. Následne sa zavolá z tejto procedúry procedúra v module rozlisenie[12]. V tomto module sa na začiatku definuje funkcia **GetSystemMetrics** s parametrom, vďaka ktorej je možné zistiť systémové vlastnosti. Z nich sú potrebné **SM\_cxScreen** a **SM\_cyScreen** pre rozlíšenie obrazovky. Tieto verejné konštanty sa použijú ako parametre, z ktorých je tá prvá pre šírku a druhá pre výšku[12,1-12]. Po tejto správe prebehne príkaz **select case**, ktorý vyhodnotí na základe šírky monitora aký koeficient priblíženia pre všetky hárky nastaví.

Pri každom úkone v ktorom sa vyžaduje práca s hárkom (zapísanie do bunky a pod) preblikne obrazovka (prejde na daný hárok a pod.), to je možné potlačiť vlastnosťou **ScreenUpdating** triedy **Application**. Na začiatku procedúry ju nastavíme na **False** a na konci na **True**

### 4.3.2 Formulár **frmStart**

Po úvodnej správe sa užívateľ nachádza na uvítacom liste, kde sa nachádzajú 2 tlačidlá (**ActiveX**). To prvé slúži pre zobrazenie hlavného formulára (**frmStart**) a to druhé pre ukončenie aplikácie. Pre tlačidlo ukončenia je vytvorená správa, či chce užívateľ skutočne ukončiť aplikáciu a na to klasická systémová otázka o uložení. Tie systémové správy je možné potlačiť (**Application.DisplayAlerts**) nastavením na začiatku na **True** a na konci procedúry späť na **False**, no v tomto projekte sa neriešili. Ak by sa riešili, tak by vybehlo okno s otázkou o uložení (nami vytvorené), s 3 tlačidlami **vbYesNoCancel**, kde by možnosť **Yes** uložila a ukončila projekt, možnosť **No** by neuložila a ukončila a posledná možnosť **Cancel** by zavrela okno so správou a nevykonalo by sa nič.

Implicitná hodnota **OP Multipage** je nastavená na 0, tzn. že sa ako úvodná karta zobrazí uvítacia s informáciami o projekte. Na nej sa nachádzajú 3 popisky zarovnané na stred, opticky to pôsobí priateľivo a výstižne, najmä zvýraznenie tučným písmom.

Na druhej karte sa nachádzajú 4 tlačidlá, 4 rámčeky a 4 popisky. Tlačidlá vyššej funkcionality by nemali obsahovať názvy, ale tie zadať do popisku neďaleko nich. Za prvé to pôsobí výstižne a najmä je možné vytvoriť dlhší text o rôznych veľkostiach, ktorý by do tlačidla nevošiel. Po stlačení každého tlačidla sa vykonajú rôzne akcie, ktorým predchádza skrytie formulára (**frmStart.hide**).

Po stlačení prvého tlačidla (**cmdPrehľadTelefonov**) sa zobrazí formulár (**frmPrehľad**). Udalosť kliknutia na druhé tlačidlo (**cmdPrehľadZnacie**) [3,27-30] vyvolá spustenie procedúry **KT1**, ktorá najprv v hárku "KT" zmaže existujúcu a následne vytvorí novú kontingenčnú tabuľku. Tabuľka vyjadruje percentuálne zastúpenie značiek v databáze. Tretie tlačidlo (**cmdPrehľadCenovýchInterv**) zaktivuje **inputBox** pre zadanie cenového rozdielu intervalov a jeho následné zobrazenie v kontingenčnej tabuľke [3,12-22]. **InputBox** je ošetrený proti zadaniu iných hodnôt ako čísel pomocou posledného argumentu a následne príkazom **IF** pre zadanie hodnoty menšej alebo rovnjej 0

[3,16-22]. Posledné tlačidlo (**cmdPrehľadVlastnostiZariadení**) zobrazí miniformulár (**frmVyberVlastnostiZariadeni**), pomocou ktorého si užívateľ vyberie vlastnosť záznamu (telefónu), ktorá sa uvedie v kontingenčnej tabuľke.

### 4.3.3 Formulár **frmPrehľad**

Tento formulár je značne zastúpený OP **ListBox** (**lstData**), ktorý je pri inicializácii formulára implicitne naplnený pomocou vlastnosti **RowSource** oblasťou (názvom) "data". Vlastnosť pre počet stĺpcov (**ColumnCount**) je nastavená na 13 a samozrejme vlastnosť pre hlavičky (**ColumnHeads**) je nastavená na **True**. Následne sa manuálne upravili šírky jednotlivých stĺpcov (**ColumnWidths**). Pod **ListBoxom** sa nachádza niekoľko tlačidiel, prvé zľava je tlačidlo (**cmdPrehľad**) pre zobrazenie formulára (**frmPrehliadatTelefony**). Druhým je tlačidlo (**cmdOdstranit**) určené pre odstránenie záznamu. Po označení záznamu v zozname a následnom stlačení sa vykoná kontrola (počet záznamov nesmie byť menší ako 3) a následne sa zobrazí okno s poverením pre odstránenie záznamu. Pri vymazávaní zo zoznamu sa využíva vlastnosť **listBoxu** **ListIndex**, ktorej hodnota sa predáva objektu typu **Range** a jeho vlastnosti **Rows** a následne sa odstráni [4,20-21]. Následne na to využijeme vlastnú funkciu pre obnovenie zoznamu, ktorej ako parametre predávame názov zoznamu, ktorý obnovujeme a názov oblasti, ktorá sa doň načítava [4,22]. Ďalšie dve tlačidlá (**cmdPridanieOS**) a (**cmdZoznamZnacie**) slúžia pre zobrazenie príslušných formulárov pre jednotlivú správu. Posledný rámček nadobúda 2 podoby, a to na základe hodnoty poľa so zoznamom (**cmbFiltre**), kde sa hodnoty naplňajú inicializáciou formulára. Pri implicitnej hodnote "Značka" sa ponúka filtrovanie priamo v zozname podľa značky, ktorú je možné vybrať OP (**cmbVyberZnacky**), tá sa taktiež naplňa pri inicializácii. Pred každým filtrovaním prebehne procedúra pre vymazanie minulých vyfiltrovaných záznamov.

Filtrovanie funguje na princípe prepojenia vlastnosti **ControlSource** OP (**cmbVyberZnacky**) na pozíciu **krit!B3**. Tá spoločne s hlavičkou je definovaná oblasťou "krit\_znacka" [4,126-128]. Obe polia majú nastavenú hodnotu vlastnosti **Style** na **fmStyleDropDownList**, ktorá usmerní užívateľa na hodnoty zo zoznamu. Inou možnosťou filtrovania je na základe vloženia cien. Obe polia sú ošetrené voči zadávaniu nečíselných alebo nevhodných znakov [4,69-104]. Samozrejmosťou je kontrola pred vložením hodnôt do zoznamu, z dôvodu nulového výsledku filtru [4,80-84]. V oboch prípadoch odfiltrovanie prebehne automaticky pomenovanie vyfiltrovaných záznamov názvom "vyfiltrovanéData". Na to vybehne správa s oznámením pre užívateľa [4,191-216]. Typ správy závisí od hodnoty premennej **cena** typu **Boolean**. V prípade hodnoty **True** je správa prispôbená príkazom **If** pre cenové rozdelenie, v prípade **False** pre výber podľa značky. Po každom odfiltrovaní prebehne označenie prvého záznamu v zozname (**ListIndex** = 0).

Po odkliknutí správy je užívateľovi ponúknutá možnosť exportu vyfiltrovaných záznamov do textového súboru [4,153-176]. Tam sa využíva objekt typu **FileSystemObject** (**FSO**) pre narábanie so

súbormi, kde následne pomocou jeho metódy **CreateTextFile** vytvoríme súbor na požadovanom disku s požadovaným názvom (záleží na užívateľovi). Zapisovanie funguje pomocou cyklu `for[4,166-172]`, kde sa najprv vezme prvý riadok filtrovaných dát a v rámci neho sa ide po stĺpec za stĺpcom. Každý riadok sa osobitne zapisuje pomocou premennej pomocná do objektu **objText**, ktorý to zapisuje priamo do súboru. Tlačidlom OK zavrieme formulár.

### 4.3.4 Formulár frmPrehliadaťTelefony

Tento formulár funguje na princípe načítania a ukladania záznamov. Využíva sa objekt typu **Range uloziskoDat** a premenná **jednotliveDataUloziska** typu **Variant** (viacero typov hodnôt pri zázname). Pre ukladanie[5,135-149] platí, že každé textové pole a pole so zoznamom má svoje miesto v premennej **jednotliveDataUloziska** (riadok, stĺpec), v našom prípade používame 1 riadok a 12 stĺpcov. Po uložení hodnôt z OP do premennej prebehne priradenie tejto premennej o 12 políach vlastnosti **Value** objektu **uloziskoDat**. Princíp načítania dát[5,120-134] funguje na opačnom princípe. Vlastnosť **value** objektu **uloziskoDat** sa načíta do premennej **jednotliveDataUloziska** a tie sa priradia do OP.

Inicializáciou tohto formulára sa spočiatku kontroluje, či je nejaká hodnota v zozname označená [5,101-103] a ak nie je, tak označí prvú. Následne priradíme objektu **uloziskoDat** záznam z hárku, ktorý závisí na aktuálnej hodnote číselníka (**spn**). Na to sa načítajú záznamy z objektu do jednotlivých OP konkrétneho formulára, pre číselník sa určí maximum a súčasná hodnota závislá na premennej **riadokLB** (označený riadok v zozname). Pri poli so zoznamom (**cmbOS**) je nastavené automatické skrývanie OP (**cmbTypOS**) v prípade ak daný telefón OS nemá (nastavenie hodnoty “- “).

Pre prepínanie sa medzi záznamami slúži číselník, ktorého zmena vyvolá procedúru **spn\_change**[5,79-98], tzn. že uloží prípadné zmeny v danom zázname, objektu **uloziskoDat** sa priradí hodnota vyššia/nížšia a následne sa daný záznam načíta. Potom prebieha aktualizácia textového poľa **txtZaznamy**, ktoré informuje užívateľa o aktuálnom zázname. Taktiež prebieha pravidelné obnovenie zoznamu (**lstData**), aby v ňom boli vždy aktuálne hodnoty z hárku.

Tlačidlo nový (**cmbNovyZaznam**) slúži pre vytvorenie nového záznamu [5,14-25]. Po jeho stlačení sa deaktivuje číselník a nad posledný záznam v oblasti “data“ v hárku “Data“ sa vloží prázdny riadok. Tieto prázdne hodnoty sa uložia do objektu **uloziskoDat** a následne načítajú do OP. Po vyplnení OP sa záznam uloží tlačidlom (**cmdOK**). Procedúra udalosti tohto tlačidla je ošetrená pre nutné zadanie všetkých záznamov okrem cien operátorov, tie sú voliteľné ale len pre číselné hodnoty [5,33-36]. Avšak stlačenie tohto tlačidla môže nastať v 2 prípadoch, klasické prepínanie medzi záznamami a pri pridani záznamu. To sa ošetrí pomocou premennej **vytvaranieZaznamu** (**Boolean**), ktorá ak je **True** a všetky náležitosti sú splnené, tak prebieha prvá časť [5,43-55] a ak sa nejedná o pridanie záznamu tak sa vykoná [5,63-68]. Po oboch variantách prebieha uloženie záznamov, zapísanie vzorcov do posledného stĺpca a zoradenie pomocou vlastnej funkcie **zoradZaznamy**.

So stlačením tlačidla (**cmbNovyZaznam**) sa zviditeľní tlačidlo (**cmdZrusVytvaranieZaznamu**). To slúži pre odstránenie tohto prázdneho riadka[5,71-78] a následne načíta príslušný záznam do objektu a ten do OP.

Pre stlačenie krížika v rohu je nastavená udalosť **QueryClose** tohto formulára, ktorá ho uzavrie.

### 4.3.5 Formulár frmOperacneSystemy

---

Zobrazením tohto formulára sa označí prvá hodnota v ListBoxe (**lstOperacneSystemy**). Hodnoty sú doň implicitne dané z oblasti "operacneSystemy".

Pre pridanie OS slúži tlačidlo (**cmdPridatOS**) po ktorého stlačení sa zobrazí inputBox pre vloženie názvu, ktorý sa uloží do premennej pridanieOperacnehoSystemu[7, 30-50]. Tá je následne ošetrovaná pre prázdnu hodnotu a hodnotu False, ktorú by nadobudla pri stlačení krížika. Pri správnom názve sa v oblasti "operacneSystemy" vloží nový riadok a doň uloží operačný systém. Následne sa zoradia tieto záznamy, obnovia v listBoxe(**lstOperacneSystemy**) a taktiež v poli so zoznamom (**cmbTypOs**), kde musia byť k dispozícii pre prípadne zmeny.

Pre odstránenie slúži tlačidlo (**cmdOdstranitOS**), po ktorého stlačení sa zobrazí správa uisťujúca, či chce užívateľ záznam odstrániť[7,1-29]. V príkaze If je zakázané odstrániť možnosť "- ", ktorá označuje v záznamoch absenciu OS (**cmbTypOS**) a taktiež sa udržiava minimálny počet záznamov na čísle 3. Pri splnení podmienok pre odstránenie sa odstráni záznam podľa hodnoty označenej v listBoxe(**lstOperacneSystemy**). Pred odstránením vybehne informačná správa o odstránenom zázname, ktorý sa v skutočnosti vymaže až následne, inak by sa nebolo možné na čo odkázať. Potom sa vlastnosť vyprázdni konštantou **vbNullString** a následne sa do nej nanovo priradí aktualizovaná oblasť. Nato sa záznamy zoradia a obnovia v príslušných OP (**lstOperacneSystemy**, **cmbTypOs**) a označí sa prvá hodnota v listBoxe.

### 4.3.6 Formulár frmZnacky

---

Tento formulár funguje analogicky rovnako ako predchádzajúci, s tým rozdielom že sa tam nachádzajú iné OP a iné oblasti, konkrétne **lstZnacky**, **cmbZnacky** a oblasť "znacky"[6,1-52].

### 4.3.7 Formulár frmVyberVlastnostiZariadeni

---

Tento formulár slúži pre výber parametru na zobrazenie kontingenčnej tabuľky v hárku "KT". Obsahuje pole so zoznamom, ktoré sa naplňa pri inicializácii poľami vlastností[8,10-20].

### 4.3.8 Modul funkcie

---

Zoradenie záznamov, užívateľsky definovaná funkcia slúži pre zoradovanie záznamov [10,1-23]. Ako parametre sa jej predávajú:

- ✓ **menoHarku** as String (názov hárku, v ktorom prebieha zoradenie)
- ✓ **klucRadenia** as String (označenie stĺpca podľa ktorého sa majú záznamy zoradiť("B3"))

- ✓ **nazovRange** as String (názov oblasti (názvu), v ktorom bude prebiehať zoradenie)
- ✓ **druhyKluc** as String (voliteľný druhý kľúč radenia)

Funkcia vznikla použitím makrokamery, v ktorej sa zmenili polia ktoré sa počas rôznych radení menili.

Funkcia obnovenia záznamov slúži pre obnovenie OP listBox danou oblasťou (názvom) [10,21-23].

Ako parametre sa jej predávajú:

- ✓ **zoznam** as Object (názov daného listBoxu, ktorý má byť obnovený)
- ✓ **nazovRange** as String (názov danej oblasti(názvu), ktorá sa má načítať do listBoxu)

### 4.3.9 Modul KT

Tento modul slúži pre spúšťanie jednotlivých kontingenčných tabuliek v projekte. Kód bol tvorený použitím makrokamery a to nasledujúcim spôsobom:

1. Na želané miesto, kde sa bude vytvárať sa vytvorí ľubovoľná KT
2. Prejde sa na inú kartu
3. Zapne sa makrokamera
4. Prejde sa ku skúšobnej KT, klikne sa do jej oblasti a použitím klávesovej skratky ctrl + \* sa celá označí
5. Vymaže sa
6. Vloží sa požadovaná KT
7. Klikne sa mimo KT
8. Vypne sa makrokamera

Tento spôsob je najjednoduchší pre rýchlu tvorbu kódu. Je potreba byť obozretný pri uvedených názvoch v kóde, aby sa zhodovali s vytváranými, z dôvodu vyhnutia sa zbytočným chybám.

Prvá KT je vytvorená s minimálnou úpravou užívateľa, pri potrebe sa zmení jej názov v častiach `ActiveSheet.PivotTables()`, aby sa v zátvorke nachádzal rovnaký názov KT[11,1-33]. KT udáva prehľad o zastúpení jednotlivých značiek v databáze. Táto KT sa vytvorí stlačením tlačidla (**cmdZastupenieZnacie**).

Druhej KT sa predáva parameter interval ako číslo. Táto KT slúži pre zobrazenie počtu telefónov v jednotlivých cenových intervaloch, ktorý si užívateľ sám zadá[11,34-68]. Na konci procedúry sa taktiež zapíše do bunky informatívny text pre užívateľa o parametroch zobrazenia KT. Táto KT sa vytvorí stlačením tlačidla (**cmdPrehľadVCenovychInterv**) a uvedením intervalu.

Tretej KT sa predáva parameter vlastnosť ako text, čo je jednou z polí vlastností telefónov, nad ktorou má byť v KT vytvorený prehľad. Na konci procedúry sa zapíše do bunky informatívny text pre

užívateľa s názvom vlastností, ktorá je uvedená v KT. Táto KT sa vytvorí stlačením tlačidla (**cmdPrehľadVlastnostiZariadeni**).

Všetky tlačidla pre vytvorenie KT sú dostupné vo formulári (**frmStart**) na karte 1. Pre spätné zobrazenie sa použije tlačidlo (**cmdKTSpät**), ktoré spätne zobrazí formulár (**frmStart**).

### Záver

V úvode práce som zhodnotil rozdiely vo verziách MS Excel 2003 a 2007 , ku ktorým som doložil aj výsledku reálneho prieskumu respondentov.

Na začiatku teoretickej časti som predstavil základné pojmy, ktoré sa vyskytujú v tejto práci. Detailne som ich popísal a najmä ku každému pojmu som uviedol prehľadný príklad. V ďalšej časti som vysvetlil čo je to vlastne užívateľské rozhranie, aké typy OP existujú a vytvoril istú konvenciu pre efektívnejšiu prácu . Následne na to som aplikoval použitie jednotlivých OP, zväčša typu ActiveX. K týmto prvkom som pridal ich popis a situácie, pri ktorých ich použitie je najvhodnejšie. Taktiež som uviedol najužitočnejšie vlastnosti, ktorými sa dá OP efektívne využiť. Každý OP obsahuje moje zhodnotenie z pohľadu funkčnosti alebo vzhľadu.

V praktickej časti som zostavil rozsiahlejší príklad na ktorom som aplikoval OP a ich jednotlivé využitie. Na jej začiatku som pripravil detailný popis a grafický návrh, ku ktorému som pripojil aj detailné náhľady všetkých používaných okien. Každé okno som jednotlivo popísal, vysvetlil aké akcie v ňom prebiehajú. V ďalšej fáze som popísal funkcionality a princípy celej aplikácie s odkazovaním sa čísla riadkov zdrojového kódu uvedeného v prílohe.

Prínosy tejto práce vidím hlavne vo sfére zberu dát, kde je potrebné vytvárať konzistentné záznamy. Tie docielime vhodnými overeniami dát a kontrolami OP. Taktiež z tejto práce môže čerpať informácie a zúročiť ich ktokoľvek, kto denne pracuje s Excelom a tým si môže zautomatizovať každodennú rutinu jedným klikom.

## Zdroje

---

### Zoznam použitých zdrojov

---

- [1] <http://www.erpforum.cz/erp-trendy/cesky-erp-trh-roste-i-v-obdobi-hospodarske-krize.html>
- [2] <http://office.microsoft.com/sk-sk/excel-help/nove-funkcie-v-programe-microsoft-office-excel-2007-HA010073873.aspx#BMui>
- [3] (Walkenbach, 2007)
- [4] (Benáčanová, 2009), str. 18-19
- [5] (Bredden & Schwimmer, 2009), str. 217-218
- [6] (Bredden & Schwimmer, 2009), str. 217-218
- [7] (Bredden & Schwimmer, 2009), str. 219
- [8] (Bredden & Schwimmer, 2009), str. 223-224
- [9] (Benáčanová, 2009), str. 56
- [10] (Bredden & Schwimmer, 2009), str.508
- [11] (Bredden & Schwimmer, 2009), str. 517
- [12] (Bredden & Schwimmer, 2009), str.517
- [13] (Bredden & Schwimmer, 2009), str. 518
- [14] (Bredden & Schwimmer, 2009), str. 518
- [15] (Bredden & Schwimmer, 2009), str.523
- [16] (Benáčanová, 2009), str.47
- [17] (Benáčanová, 2009), str. 46
- [18] (Bredden & Schwimmer, 2009), str.560
- [19] (Bredden & Schwimmer, 2009), str. 550-553
- [20] (Bredden & Schwimmer, 2009), str.546



## Zoznam knižnej literatúry

BENÁČANOVÁ, Helena. *Tvorba aplikácií v Microsoft Excel : 2. časť Možnosti VBA*. První. Praha : Oeconomica, 2009. 125 s. ISBN 978-80-245-1602-8.

BREDEN, Melanie; SCHWIMMER, Michael. *Excel 2007 VBA : Velká kniha řešení*. První. Brno : Computer Press, a.s., 2009. 685 s. ISBN 978-80-251-2698-1.

KRÁL, Martin. *Excel VBA : Výukový kurz*. První. Brno : Computer Press, a.s., 2010. 504 s. ISBN 978-80-251-2358-4.

WALKENBACH, John. *Excel 2007 VBA Programming : For dummies*. Indianapolis : Wiley Publishing, Inc., 2007. What is VBA?, s. 20. ISBN 978-0-470-04674-6.

## Zoznam elektronických zdrojov

<http://www.erpforum.cz/erp-trendy/cesky-erp-trh-roste-i-v-obdobi-hospodarske-krize.html>

<http://office.microsoft.com/sk-sk/excel-help/nove-funkcie-v-programe-microsoft-office-excel-2007-HA010073873.aspx#BMui>

## Zoznam obrázkov

Obrázok 2.1- MS Excel 2003 .....	10
Obrázok 2.2- MS Excel 2007 .....	10
Obrázok 2.3- Stavový riadok.....	10
Obrázok 3.1- Logo VBA.....	13
Obrázok 3.2- Karta Vývojar .....	14
Obrázok 3.3- Spustenie makrokamery .....	14
Obrázok 3.4- Okno so správou .....	15
Obrázok 3.5- VBE .....	14
Obrázok 3.6- Premenné.....	16
Obrázok 3.7- Spustenie makra.....	17
Obrázok 3.8- Udalostná procedúra .....	18
Obrázok 3.9- Správa .....	18
Obrázok 3.10- Modul.....	19
Obrázok 3.11- Funkcia pohlavie.....	19
Obrázok 3.12- Výber funkcie .....	19
Obrázok 3.13- Ovládacie prvky.....	21
Obrázok 3.14- Vlastnosti OP formulára.....	22
Obrázok 3.15- Vlastnosti OP formulára.....	22
Obrázok 3.16- Vlastnosti OP ActiveX.....	23
Obrázok 3.17- Roletkový zoznam udalostí OP .....	24
Obrázok 3.18- MsgBox otázka .....	26
Obrázok 3.19- Odpoveď Yes .....	26
Obrázok 3.20- Odpoveď No.....	26
Obrázok 3.21- InputBox.....	27
Obrázok 3.22- Application InputBox.....	27
Obrázok 3.23- Userform .....	28
Obrázok 3.24- Vlastnosti OP ActiveX.....	28
Obrázok 3.25- ToolBox .....	29

Obrázok 3.26- MouseMove 2 .....	30
Obrázok 3.27- MouseMove 1 .....	30
Obrázok 3.28- Ukončenie programu.....	30
Obrázok 3.29- Vlastnosti hodnot True.....	31
Obrázok 3.30- Vlastnosti tlačidiel False .....	32
Obrázok 3.31- TextBox príklad.....	33
Obrázok 3.32- Frame vzhľad.....	34
Obrázok 3.33- Frame funkčnosť 1 .....	34
Obrázok 3.34- Frame funkčnosť 2.....	34
Obrázok 3.35- ListBox ručne .....	35
Obrázok 3.36- Označenie hodnôt .....	35
Obrázok 3.37- Názov mesiace.....	35
Obrázok 3.38- Listbox plnený RowSource .....	36
Obrázok 3.39- Výber dátumu narodenia .....	37
Obrázok 3.40- Mzda .....	39
Obrázok 3.41- Študentská mzda .....	39
Obrázok 3.42- Image .....	40
Obrázok 3.43- Číselník .....	40
Obrázok 3.44- MultiPage 1 .....	42
Obrázok 3.45- MultiPage 2 .....	42
Obrázok 3.46- Tabstrip 1 .....	42
Obrázok 3.47- Tabstrip 2 .....	42
Obrázok 3.48- Stlačené prepínacie tlačidlo.....	42
Obrázok 3.49- Úprava písma výberu .....	43
Obrázok 3.50- Kalendár .....	44
Obrázok 4.1- Návrh aplikácie.....	46
Obrázok 4.2- Uvítacia karta 2 .....	46
Obrázok 4.3- Uvítacia karta 1 .....	46
Obrázok 4.4- Informačná karta 1 .....	47
Obrázok 4.5- Informačná karta 2.....	47
Obrázok 4.6- Formulár prehľad.....	47
Obrázok 4.7- Prehliadanie záznamuu .....	48
Obrázok 4.8- Správa značiek.....	48
Obrázok 4.9- Správa OS .....	48

### Zoznam tabuliek

Tabuľka 2.1- Porovnanie verzií Excelu .....	9
Tabuľka 3.1- Dátové typy [4].....	16
Tabuľka 3.2- Udalosti Workbooku.....	18
Tabuľka 3.3- Ovládacie prvky formulára [7] .....	23
Tabuľka 3.4- Konvencia OP.....	25
Tabuľka 3.5- Jednotlivé konštanty pre prácu s MsgBoxom.....	26
Tabuľka 3.6- Tabuľka hodnôt[9] .....	27
Tabuľka 3.7- Typy OP typu Label.....	30

## Zoznam grafov

Graf 2.1- Podiel používaných verzií .....	11
Graf 2.2- Prechod z verzie 2003 na 2007 .....	12
Graf 2.3- Hodnotenie verzií MS Excel .....	12

## Zoznam použitých skratiek

Skratka	Popis
VBA	Visual Basic for Applications
A pod.	A podobne
Napr.	Napríklad
VBE	Visual Basic Editor
OP	Ovládacie prvky
Resp.	Respektíve
Atd.	A tak ďalej
Tzn	To znamená
MS	Microsoft
OS	Operačný systém (mobilný)
KT	Kontingenčná tabuľka

## Príloha

---

Z dôvodu veľkej kapacity komentárov nie sú uvedené v kóde, ale jednotlivo odkazované z realizácie praktickej časti.

## Zdrojový kód

---

### 1.Otvorenie zošita

```
1 Private Sub Workbook_Open()  
2 Dim text As String  
3 With Application  
4     .DisplayFormulaBar = False  
5     .DisplayStatusBar = False  
6 End With  
7 With ActiveWindow  
8     .DisplayGridlines = False  
9     .DisplayHeadings = False  
10    .DisplayWorkbookTabs = False  
11    .DisplayHorizontalScrollBar = False  
12    .DisplayVerticalScrollBar = False  
13 End With  
14 Application.DisplayFullScreen = True  
15 Select Case Hour(Time)  
16     Case Is < 8  
17         text = "Dobrý večer " & Application.UserName  
18     Case Is < 12  
19         text = "Dobré ráno " & Application.UserName  
20     Case Is < 18  
21         text = "Dobrý deň " & Application.UserName  
22 End Select  
23 MsgBox text  
24 Call nastavRozlisenie  
25 End Sub  
  
26 Private Sub Workbook_BeforeClose(Cancel As Boolean)  
27 With Application  
28     .DisplayFormulaBar = True  
29     .DisplayStatusBar = True  
30 End With  
31 With ActiveWindow  
32     .DisplayGridlines = True  
33     .DisplayHeadings = True  
34     .DisplayWorkbookTabs = True  
35     .DisplayHorizontalScrollBar = True  
36     .DisplayVerticalScrollBar = True  
37 End With  
38 Application.DisplayFullScreen = False  
39 End Sub
```

### 2.Hárok Vitajte

```
1 Private Sub cmdSpustit_Click()  
2 frmStart.Show  
3 End Sub  
  
4 Private Sub cmdUkoncit_Click()  
5 Dim konec As Byte  
6 konec = MsgBox("Vážne chcete ukončiť aplikáciu ? ", vbYesNo + vbQuestion, "Koniec_  
7 aplikácie")  
8 If konec = vbYes Then ActiveWorkbook.Close  
9 End Sub
```

### 3.Zobrazenie a inicializácia formulára frmStart

```
1 Private Sub cmdHlavnaStranka_Click()  
2 Sheets("Vitajte").Select  
3 frmStart.Hide  
4 End Sub  
  
5 Private Sub cmdKoniecAplikacie_Click()  
6 End  
7 End Sub  
  
8 Private Sub cmdPrehľadTelefonov_Click()  
9 frmStart.Hide  
10 frmPrehľad.Show  
11 End Sub  
  
12 Private Sub cmdPrehľadVCenovýchInterv_Click()  
13 Dim interval As Integer  
14 interval = Application.InputBox("Zadajte nezáporný rozdiel hodnot intervalu",_  
15 "Interval", , , , , 1)  
16 If interval <= 0 Then  
17     MsgBox "Nesprávny rozdiel hodnot"  
18     Exit Sub  
19 Else  
20     Call KT2(interval)  
21     frmStart.Hide  
22 End If  
23 End Sub  
  
24 Private Sub cmdPrehľadVlastnostiZariadeni_Click()  
25 frmVyberVlastnostiZariadeni.Show  
26 End Sub  
  
27 Private Sub cmdZastupenieZnacie_k_Click()  
28 Call KT1  
29 frmStart.Hide  
30 End Sub  
  
31 Private Sub cmdZobrazInfo_Click()  
32 frmStart.Hide  
33 frmInfo.Show  
34 End Sub
```

#### 4.Zobrazenie a inicializácia formulára frmPrehľad

```
1 Dim vytvaranieZaznamu As Boolean
2 Dim zoznam As Range
3 Dim znak As String
4 Dim cena As Boolean

5 Private Sub cmdOdstranit_Click()
6 Dim sprava As Byte
7 Application.ScreenUpdating = False
8 cmdOdstranit.BackColor = &H8000000F
9 cmdOdstranit.ForeColor = vbBlack
10 If lstData.ListCount > 3 Then
11     sprava = MsgBox("Vážne chcete odstrániť záznam č. " & lstData.ListIndex + 1 & _
12     ", " & Range("data").Cells(frmPrehľad.lstData.ListIndex + 1, 1) & " " & _
13     Range("data").Cells(frmPrehľad.lstData.ListIndex + 1, 2) & " ?", vbCritical + _
14     vbYesNo, "Odstránenie záznamu")
15     If sprava = vbYes Then
16         MsgBox "Záznam " & Range("data").Cells(frmPrehľad.lstData.ListIndex + 1,1)_
17         & " " & Range("data").Cells(frmPrehľad.lstData.ListIndex + 1, 2) & " bol _
18         odstránený.", vbInformation
19         Sheets("Data").Select
20         Range("data").Rows(frmPrehľad.lstData.ListIndex + 1).Select
21         Selection.Delete Shift:=xlUp
22         Call obnovZoznam(lstData, "data")
23         lstData.ListIndex = 0
24     End If
25     Call obnovZoznam(lstData, "data")
26 Else
27     MsgBox "Počet záznamov nesmie byť menší ako 3 ", vbInformation
28 End If
29 Application.ScreenUpdating = True
30 End Sub

31 Private Sub cmdOdstranit_MouseMove(ByVal Button As Integer, ByVal Shift As Integer,_
32     ,_ ByVal X As Single, ByVal Y As Single)
33 cmdOdstranit.BackColor = vbBlue
34 cmdOdstranit.ForeColor = vbWhite
35 End Sub

36 Private Sub cmdOkSkryFrmPrehľad_Click()
37 Call obnovZoznam(lstData, "data")
38 frmPrehľad.Hide
39 frmStart.Show
40 End Sub

41 Private Sub cmdPrehľad_Click()
42 Call obnovZoznam(lstData, "data")
43 cmdPrehľad.BackColor = &H8000000F
44 cmdPrehľad.ForeColor = vbBlack
45 frmPrehľad.Hide
46 frmPrehľadatTelefony.Show
47 End Sub

48 Private Sub cmdPrehľad_MouseMove(ByVal Button As Integer, ByVal Shift As Integer,_
49     ByVal X As Single, ByVal Y As Single)
50 cmdPrehľad.BackColor = vbBlue
51 cmdPrehľad.ForeColor = vbWhite
52 End Sub

53 Private Sub cmdPridanieOS_Click()
54 frmOperacneSystemy.Show
55 Call obnovZoznam(lstData, "data")
56 End Sub

57 Private Sub cmdPridanieOS_MouseMove(ByVal Button As Integer, ByVal Shift As _
58     Integer, ByVal X As Single, ByVal Y As Single)
```

## Návrh užívateľského rozhrania pri tvorbe aplikácii vo VBA

---

```
59 cmdPridanieOS.BackColor = vbBlue
60 cmdPridanieOS.ForeColor = vbWhite
61 End Sub

62 Private Sub cmdPridat_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, _
63 ByVal X As Single, ByVal Y As Single)
64 cmdPridat.BackColor = vbBlue
65 cmdPridat.ForeColor = vbWhite
66 End Sub

67 Sub cmdVybratRozpatie_Click()
68 Application.ScreenUpdating = False
69 If Not IsNumeric(txtCenaOd) Or Not IsNumeric(txtCenaDo) Then MsgBox "Používajte_
70 prosim čísla": Exit Sub
71 If Val(txtCenaOd) < Val(txtCenaDo) Then
72     Call vymazFiltrovaneData
73     Sheets("Data").Select
74     Range("D4:P4").Select
75     Range(Selection, Selection.End(xlDown)).Select
76     With Selection
77         .AdvancedFilter Action:=xlFilterCopy, CriteriaRange:=Range("krit_cena"), _
78 CopyToRange:=Range("AF4:AR4"), Unique:=False
79     End With
80     If Range("AF5") = "" Then
81         MsgBox "Vašim kritériam nevyhovujú žiadne záznamy"
82         Call obnovZoznam(lstData, "data")
83         Exit Sub
84     End If
85     Range("AF5:AR5").Select
86     Range(Selection, Selection.End(xlDown)).Select
87     Selection.Name = "filtrovaneData"
88     Call obnovZoznam(lstData, "filtrovaneData")
89     cena = True
90     Call vyhodSpravu
91     cena = False
92     txtCenaOd.text = ""
93     txtCenaDo.text = ""
94     sprava = MsgBox("Máte záujem vyexportovať vyfiltrované data do txt súboru ?", _
95 vbYesNo + vbInformation, "Export dát")
96     If sprava = vbYes Then
97         Call exportDat
98     End If
99     lstData.ListIndex = 0
100 Else
101     MsgBox "Zadajte vhodnejšie kritéria."
102 End If
103 Application.ScreenUpdating = True
104 End Sub

105 Sub vymazFiltrovaneData()
106 Application.ScreenUpdating = False
107 If Range("AF5") = "" Then
108     Exit Sub
109 Else
110     Range("AF5:AR5").Select
111     Range(Selection, Selection.End(xlDown)).Select
112     Selection.ClearContents
113 End If
114 Application.ScreenUpdating = True
115 End Sub

116 Private Sub cmdVybratZnacka_Click()
117 Application.ScreenUpdating = False
118 Dim sprava As Byte
119 If cmbVyberZnacky.Value = "" Then
120     MsgBox "Je potrebné vybrať značku z ponuky"
121 Else
122     Call vymazFiltrovaneData
```

## Návrh užívateľského rozhrania pri tvorbe aplikácii vo VBA

---

```
123     Sheets("Data").Select
124     Range("D4:P4").Select
125     Range(Selection, Selection.End(xlDown)).Select
126     With Selection
127         .AdvancedFilter Action:=xlFilterCopy, CriteriaRange:=Range("krit_znacka"), _
128     CopyToRange:=Range("AF4:AR4"), Unique:=False
129     End With
130     If Range("AF5") = "" Then
131         MsgBox "Vašim kritériam nevyhovujú žiadne záznamy"
132         Call obnovZoznam(lstData, "data")
133         Exit Sub
134     End If
135     If Range("AF6") = "" Then
136         Range("AF5:AR5").Select
137     Else
138         Range("AF5:AR5").Select
139         Range(Selection, Selection.End(xlDown)).Select
140     End If
141     Selection.Name = "filtrovaneData"
142     Call obnovZoznam(lstData, "filtrovaneData")
143     Call vyhodSpravu
144     sprava = MsgBox("Máte záujem vyexportovať vyfiltrované data do txt súboru ?", _
145     vbYesNo + vbInformation, "Export dát")
146     If sprava = vbYes Then
147         Call exportDat
148     End If
149     lstData.ListIndex = 0
150 End If
151 Application.ScreenUpdating = True
152 End Sub

153 Sub exportDat()
154 Application.ScreenUpdating = False
155 Dim objFs As Object
156 Dim objText As Object
157 Dim strData As String
158 Dim n As Integer
159 Dim i As Integer
160 Dim nazovSuboru As String
161 Dim znak As String
162 Set objFs = CreateObject("Scripting.FileSystemObject")
163 nazovSuboru = Application.InputBox("Zadajte názov súboru, ktorý sa vytvorí na _
164 disku C ", "Nazov súboru")
165 Set objText = objFs.CreateTextFile("C:\\" & nazovSuboru & ".txt", 1)
166 For n = 0 To Sheets("Data").Range("filtrovaneData").Rows.Count - 1
167     For i = 0 To Sheets("Data").Range("filtrovaneData").Columns.Count - 1
168         pomocna = pomocna & ActiveSheet.Range("AF4").Offset(n, i).text & "|"
169     Next i
170     objText.WriteLine (pomocna)
171     pomocna = ""
172 Next n
173 objText.Close
174 Set objFs = Nothing
175 Application.ScreenUpdating = True
176 End Sub

177 Private Sub UserForm_Initialize()
178 Application.ScreenUpdating = False
179 txtCenaOd.text = ""
180 txtCenaDo.text = ""
181 With cmbFiltre
182     .AddItem "Značka"
183     .AddItem "Cenové rozpatie"
184     .Value = "Značka"
185     .Style = fmStyleDropDownList
186 End With
187 cmbVyberZnacky.RowSource = "znacky"
188 cmbVyberZnacky.Style = fmStyleDropDownList
```



## Návrh užívateľského rozhrania pri tvorbe aplikácii vo VBA

---

```
189 Application.ScreenUpdating = True
190 End Sub

191 Sub vyhodSpravu()
192 If cena Then
193     If Range("filtrovaneData").Rows.Count = 1 Then
194         MsgBox "Vyfiltrovaný bol 1 záznam v cenovom rozsahu od " & txtCenaOd.text _
195 & " do " & txtCenaDo.text
196     ElseIf Range("filtrovaneData").Rows.Count > 1 And_
197 Range("filtrovaneData").Rows.Count < 5 Then
198         MsgBox "Vyfiltrované boli " & Range("filtrovaneData").Rows.Count & " _
199 záznamy v cenovom rozsahu od " & txtCenaOd.text & " do " & txtCenaDo.text
200     Else
201         MsgBox "Vyfiltrovaných bolo " & Range("filtrovaneData").Rows.Count & " _
202 záznamov v cenovom rozsahu od " & txtCenaOd.text & " do " & txtCenaDo.text
203     End If
204 Else
205     If Range("filtrovaneData").Rows.Count = 1 Then
206         MsgBox "Vyfiltrovaný bol 1 záznam podľa značky " & cmbVyberZnacky.Value
207     ElseIf Range("filtrovaneData").Rows.Count > 1 And_
208 Range("filtrovaneData").Rows.Count < 5 Then
209         MsgBox "Vyfiltrované boli " & Range("filtrovaneData").Rows.Count & " _
210 záznamy podľa značky " & cmbVyberZnacky.Value
211     Else
212         MsgBox "Vyfiltrovaných bolo " & Range("filtrovaneData").Rows.Count & " _
213 záznamov podľa značky " & cmbVyberZnacky.Value
214     End If
215 End If
216 End Sub

217 Private Sub cmdZoznamZnaciekJ_Click()
218 frmZnacky.Show
219 Call obnovZoznam(lstData, "data")
220 End Sub

221 Private Sub cmbFiltre_Change()
222 If cmbFiltre.Value = "Značka" Then
223     frameCenoveRozdelenie.Visible = False
224 Else
225     frameCenoveRozdelenie.Visible = True
226 End If
227 Call obnovZoznam(lstData, "data")
228 End Sub

229 Private Sub frameZaznamy_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, _
230 , ByVal X As Single, ByVal Y As Single)
231 cmdOdstranit.BackColor = &H8000000F
232 cmdOdstranit.ForeColor = vbBlack
233 cmdPrehľad.BackColor = &H8000000F
234 cmdPrehľad.ForeColor = vbBlack
235 End Sub

236 Private Sub UserForm_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, _
237 ByVal X As Single, ByVal Y As Single)
238 cmdOdstranit.BackColor = &H8000000F
239 cmdOdstranit.ForeColor = vbBlack
240 cmdPrehľad.BackColor = &H8000000F
241 cmdPrehľad.ForeColor = vbBlack
242 cmdZoznamZnaciekJ.BackColor = &H8000000F
243 cmdZoznamZnaciekJ.ForeColor = vbBlack
244 cmdPridanieOS.BackColor = &H8000000F
245 cmdPridanieOS.ForeColor = vbBlack
246 End Sub

247 Private Sub cmdZoznamZnaciekJ_MouseMove(ByVal Button As Integer, ByVal Shift As _
248 Integer, ByVal X As Single, ByVal Y As Single)
249 cmdZoznamZnaciekJ.BackColor = vbBlue
250 cmdZoznamZnaciekJ.ForeColor = vbWhite
```

```
251 End Sub

252 Private Sub Frame1_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, _
253 ByVal X As Single, ByVal Y As Single)
254 cmdZoznamZnaciek.BackColor = &H80000000F
255 cmdZoznamZnaciek.ForeColor = vbBlack
256 cmdPridanieOS.BackColor = &H80000000F
257 cmdPridanieOS.ForeColor = vbBlack
258 End Sub

259 Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
260 frmPrehľad.Hide
261 frmStart.Show
262 End Sub
```

## 5. Formulár frmPrehliadatTelefony

```
1 Dim uloziskoDat As Range
2 Dim jednotlivDataUloziska As Variant
3 Dim riadokLB As Integer
4 Dim vytvaranieZaznamu As Boolean
5 Dim vytvorenyZaznam As Boolean

6 Private Sub cmbOS_Change()
7   If cmbOS.Value = "N" Then
8     cmbTypOs.Visible = False
9     cmbTypOs.Value = "-"
10  Else
11    cmbTypOs.Visible = True
12  End If
13 End Sub

14 Private Sub cmdNovyZaznam_Click()
15   vytvaranieZaznamu = True
16   spn.Enabled = False
17   Sheets("Data").Range("data").Select
18   Selection.End(xlDown).Select
19   Selection.EntireRow.Insert
20   Call obnovZoznam(frmPrehľad.lstData, "data")
21   Set uloziskoDat = Range("data").Rows(frmPrehľad.lstData.ListCount - 1)
22   Call nacitajZaznamy
23   cmdZrusVytvaranieZaznamu.Visible = True
24   cmdNovyZaznam.Visible = False
25 End Sub

26 Private Sub cmdOK_Click()
27   If vytvaranieZaznamu Then
28     If cmbZnacky.Value <> "" And txtModel.text <> "" Then
29       If cmbMMS.Value = "" Or cmbFoto.Value = "" Or cmbPortal.Value = "" Or_
30       cmbOS.Value = "" Or cmbUmts.Value = "" Then
31         MsgBox "Vlastnosti musia byť tiež vyplnené", vbCritical
32       Else
33         If (Len(Trim(txtO2.text)) > 0 And Not IsNumeric(txtO2.text)) Or_
34         (Len(Trim(txtTmobile.text)) > 0 And Not IsNumeric(txtTmobile)) Or_
35         (Len(Trim(txtVodafone.text)) > 0 And Not IsNumeric(txtVodafone)) Or_
36         (Len(Trim(txtTwist.text)) > 0 And Not IsNumeric(txtTwist)) Then
37           MsgBox "Jednotlivé polia operátorov sú čísla", vbCritical
38         Else
39           If cmbOS.Value = "A" And cmbTypOs.Value = "-" Then
40             MsgBox "Ak má telefon operačný systém, tak je potrebné zvoliť_
41 konkrétne typ z ponuky", vbInformation
42           Else
43             frmPrehliadatTelefony.Hide
44             Call ulozZaznamy
45             Call vzorce
46             MsgBox "Hodnota " & jednotlivDataUloziska(1, 1) & " " &_
47             jednotlivDataUloziska(1, 2) & " bola vložená"
48             Call zoradZaznamy("Data", "D5", "data", "E5")
49             Call obnovZoznam(frmPrehľad.lstData, "data")
50             vytvaranieZaznamu = False
51             cmdNovyZaznam.Visible = True
52             cmdZrusVytvaranieZaznamu.Visible = False
53             spn.Enabled = True
54             vytvorenyZaznam = True
55             frmPrehľad.Show
56           End If
57         End If
58       End If
59     Else
60       MsgBox "Nie sú vyplnené povinné polia", vbCritical
61     End If
62   Else
63     frmPrehliadatTelefony.Hide
```

## Návrh užívateľského rozhrania pri tvorbe aplikácii vo VBA

---

```
64      Call ulozZaznamy
65      Call vzorce
66      Call zoradZaznamy("Data", "D5", "data", "E5")
67      Call obnovZoznam(frmPrehľad.lstData, "data")
68      frmPrehľad.Show
69  End If
70  End Sub

71  Private Sub cmdZrusVytvaranieZaznamu_Click()
72      vytvaranieZaznamu = False
73      spn.Enabled = True
74      Selection.EntireRow.Delete
75      Set uloziskoDat = Range("data").Rows(riadokLB)
76      Call nacitajZaznamy
77      cmdNovyZaznam.Visible = True
78  End Sub

79  Private Sub spn_Change()
80      With spn
81          .Min = 1
82          .Max = frmPrehľad.lstData.ListCount
83      End With
84      If cmbOS.Value = "N" Then
85          cmbTypOs.Visible = False
86          cmbTypOs.Value = "-"
87      Else
88          cmbTypOs.Visible = True
89      End If
90      If vytvorenyZaznam = False Then
91          Call ulozZaznamy
92      End If
93      Set uloziskoDat = Range("data").Rows(spn.Value)
94      Call nacitajZaznamy
95      txtZaznamy.text = "Záznam č. " & spn.Value & " z " & spn.Max
96      Call obnovZoznam(frmPrehľad.lstData, "data")
97      vytvorenyZaznam = False
98  End Sub

99  Private Sub UserForm_Initialize()
100      cmdNovyZaznam.Visible = True
101      vytvorenyZaznam = False
102      If riadokLB = 0 Then
103          riadokLB = 1
104      End If
105      riadokLB = frmPrehľad.lstData.ListIndex + 1
106      Sheets("Data").Select
107      Set uloziskoDat = Range("data").Rows(riadokLB)
108      Call nacitajZaznamy
109      With spn
110          .Value = riadokLB
111          .Max = frmPrehľad.lstData.ListCount
112      End With
113      If cmbOS.Value = "N" Then
114          cmbTypOs.Visible = False
115          cmbTypOs.Value = "-"
116      Else
117          cmbTypOs.Visible = True
118      End If
119  End Sub

120  Sub nacitajZaznamy()
121      jednotliveDataUloziska = uloziskoDat.Value
122      cmbZnacky.Value = jednotliveDataUloziska(1, 1)
123      txtModel.text = jednotliveDataUloziska(1, 2)
124      txtO2.text = jednotliveDataUloziska(1, 3)
125      txtVodafone.text = jednotliveDataUloziska(1, 4)
126      txtTmobile.text = jednotliveDataUloziska(1, 5)
127      txtTwist.text = jednotliveDataUloziska(1, 6)
```

## Návrh užívateľského rozhrania pri tvorbe aplikácii vo VBA

---

```
128 cmbMMS.Value = jednotlivDataUloziska(1, 7)
129 cmbFoto.Value = jednotlivDataUloziska(1, 8)
130 cmbPortal.Value = jednotlivDataUloziska(1, 9)
131 cmbUmts.Value = jednotlivDataUloziska(1, 10)
132 cmbOS.Value = jednotlivDataUloziska(1, 11)
133 cmbTypOs.Value = jednotlivDataUloziska(1, 12)
134 End Sub

135 Sub ulozZaznamy()
136 jednotlivDataUloziska(1, 1) = cmbZnacky.text
137 jednotlivDataUloziska(1, 2) = txtModel.text
138 jednotlivDataUloziska(1, 3) = txtO2.text
139 jednotlivDataUloziska(1, 4) = txtVodafone.text
140 jednotlivDataUloziska(1, 5) = txtTmobile.text
141 jednotlivDataUloziska(1, 6) = txtTwist.text
142 jednotlivDataUloziska(1, 7) = cmbMMS.Value
143 jednotlivDataUloziska(1, 8) = cmbFoto.Value
144 jednotlivDataUloziska(1, 9) = cmbPortal.Value
145 jednotlivDataUloziska(1, 10) = cmbUmts.Value
146 jednotlivDataUloziska(1, 11) = cmbOS.Value
147 jednotlivDataUloziska(1, 12) = cmbTypOs.Value
148 uloziskoDat.Value = jednotlivDataUloziska
149 End Sub

150 Sub vzorce()
151 Range("R5").Select
152 Selection.Copy
153 Range("O5").Select
154 Selection.End(xlDown).Select
155 ActiveCell.Offset(, 1).Select
156 Range(Selection, Selection.End(xlUp)).Select
157 Range(Selection, Selection.End(xlUp)).Select
158 Range(Selection, Selection.End(xlUp)).Select
159 Range(Selection, Selection.End(xlUp)).Select
160 ActiveSheet.Paste
161
162 Range("R7").Select
163 Application.CutCopyMode = False
164 Selection.Copy
165 Range("P4").Select
166 ActiveSheet.Paste
167 Range("P1:P3").Select
168 Application.CutCopyMode = False
169 Selection.Clear
170 End Sub

171 Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
172 If CloseMode = vbFormControlMenu Then
173     Cancel = True
174 End If
175 End Sub
```

## 6.Zobrazenie a inicializácia formulára frmZnacky

```
1 Private Sub cmdOdstranitZnacku_Click()  
2 Application.ScreenUpdating = False  
3 Dim strRange As String  
4 Dim sprava As Byte  
5 If lstZnacky.ListCount > 3 Then  
6     With lstZnacky  
7         strRange = .RowSource  
8         If Range(strRange).Cells(lstZnacky.ListIndex + 1, 1) = "" Then Exit Sub  
9         sprava = MsgBox("Vážne chcete odstrániť značku " &_  
10 Range(strRange).Cells(lstZnacky.ListIndex + 1, 1) & " ?", vbYesNo, "Odstránenie_  
11 značky")  
12         If sprava = vbYes Then  
13             MsgBox "Hodnota " & Range(strRange).Cells(lstZnacky.ListIndex + 1,_  
14 1) & " bola odstránená"  
15             Range(strRange).Cells(.ListIndex + 1, 1).Delete Shift:=xlUp  
16             .RowSource = vbNullString  
17             .RowSource = strRange  
18         End If  
19     End With  
20     Call zoradZaznamy("Sznacky", "B3", "znacky")  
21     Call obnovZoznam(lstZnacky, "znacky")  
22     Call obnovZoznam(frmPrehliadatTelefony.cmbZnacky, "znacky")  
23     lstZnacky.ListIndex = 0  
24 Else  
25     MsgBox "Počet záznamov nesmie byť menší ako 3", vbInformation  
26 End If  
27 Application.ScreenUpdating = True  
28 End Sub  
  
29 Private Sub cmdPridatZnacku_Click()  
30 Application.ScreenUpdating = False  
31 pridanieZnacky = Application.InputBox("Zadajte názov značky", "Pridanie značky")  
32 If pridanieZnacky = "" Then  
33     MsgBox "Nezadali ste názov!"  
34 Else  
35     Sheets("Sznacky").Select  
36     Range("znacky").Select  
37     Selection.End(xlDown).Select  
38     Selection.EntireRow.Insert  
39     Selection = pridanieZnacky  
40     MsgBox "Hodnota " & pridanieZnacky & " bolá vložená"  
41     Call zoradZaznamy("Sznacky", "B3", "znacky")  
42     Call obnovZoznam(lstZnacky, "znacky")  
43     Call obnovZoznam(frmPrehliadatTelefony.cmbZnacky, "znacky")  
44 End If  
45 Application.ScreenUpdating = True  
46 End Sub  
  
47 Private Sub UserForm_Initialize()  
48 lstZnacky.ListIndex = 0  
49 End Sub  
  
50 Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)  
51 Unload frmZnacky  
52 End Sub
```

## 7.Zobrazenie a inicializácia formulára frmOperacneSystemy

```
1 Private Sub cmdOdstranitOS_Click()  
2 Application.ScreenUpdating = False  
3 Dim strRange As String  
4 Dim sprava As Byte  
5 If lstOperacneSystemy.ListCount > 3 Then  
6     With lstOperacneSystemy  
7         strRange = .RowSource  
8         If Range(strRange).Cells(lstOperacneSystemy.ListIndex + 1, 1) = "-" Or_  
9 Range(strRange).Cells(lstOperacneSystemy.ListIndex + 1, 1) = "" Then Exit Sub  
10         sprava = MsgBox("Vážne chcete odsrániť operačný systém " &  
11 Range(strRange).Cells(lstOperacneSystemy.ListIndex + 1, 1) & " ?", vbYesNo, _  
12 "Odstránenie operačného systému")  
13         If sprava = vbYes Then  
14             MsgBox "Hodnota " &  
15 Range(strRange).Cells(lstOperacneSystemy.ListIndex + 1, 1) & " bola odstránená"  
16             Range(strRange).Cells(.ListIndex + 1, 1).Delete Shift:=xlUp  
17             .RowSource = vbNullString  
18             .RowSource = strRange  
19         End If  
20     End With  
21     Call zoradZaznamy("SoperacneSystemy", "B3", "operacneSystemy")  
22     Call obnovZoznam(lstOperacneSystemy, "operacneSystemy")  
23     Call obnovZoznam(frmPrehliadatTelefony.cmbTypOs, "operacneSystemy")  
24     lstOperacneSystemy.ListIndex = 0  
25 Else  
26     MsgBox "Počet záznamov nesmie byť menší ako 3", vbInformation  
27 End If  
  
28 Application.ScreenUpdating = True  
29 End Sub  
  
30 Private Sub cmdPridatOS_Click()  
31 Application.ScreenUpdating = False  
32 pridanieOperacnehoSystemu = Application.InputBox("Zadajte názov operačného_  
33 systému", "Pridanie operačného systému")  
34 If pridanieOperacnehoSystemu = "" Then  
35     MsgBox "Nezadali ste názov!"  
36 ElseIf pridanieOperacnehoSystemu = False Then  
37     Exit Sub  
38 Else  
39     Sheets("SoperacneSystemy").Select  
40     Range("operacneSystemy").Select  
41     Selection.End(xlDown).Select  
42     Selection.EntireRow.Insert  
43     Selection = pridanieOperacnehoSystemu  
44     MsgBox "Hodnota " & pridanieOperacnehoSystemu & " bolá vložená"  
45     Call zoradZaznamy("SoperacneSystemy", "B3", "operacneSystemy")  
46     Call obnovZoznam(lstOperacneSystemy, "operacneSystemy")  
47     Call obnovZoznam(frmPrehliadatTelefony.cmbTypOs, "operacneSystemy")  
48 End If  
49 Application.ScreenUpdating = True  
50 End Sub  
  
51 Private Sub UserForm_Initialize()  
52 lstOperacneSystemy.ListIndex = 0  
53 End Sub  
  
54 Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)  
55 Unload frmOperacneSystemy  
56 End Sub
```

## 8.Zobrazenie a inicializácia formulára frmVyberVlastnostiZariadeni

```
1 Private Sub cmdVybratVlastnostZariadenia_Click()  
2 If cmbVlastnostZariadenia.Value = "" Then  
3     Exit Sub  
4 Else  
5     frmVyberVlastnostiZariadeni.Hide  
6     frmStart.Hide  
7     Call KT3(cmbVlastnostZariadenia.Value)  
8 End If  
9 End Sub  
  
10 Private Sub UserForm_Initialize()  
11 With cmbVlastnostZariadenia  
12     .AddItem "MMS"  
13     .AddItem "Foto"  
14     .AddItem "Portal"  
15     .AddItem "UMTS"  
16     .AddItem "OS"  
17     .AddItem "TypOS"  
18     .Style = fmStyleDropDownList  
19 End With  
20 End Sub  
  
21 Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)  
22 If CloseMode = vbFormControlMenu Then  
23     Cancel = True  
24 End If  
25 End Sub
```

## 9.Zobrazenie formulára frmInfo

```
1 Private Sub cmdInfoOK_Click()  
2 Unload frmInfo  
3 frmStart.Show  
4 End Sub
```

## 10.Modul funkcie

```
1 Function zoradZaznamy(menoHarku As String, klucRadenia As String, nazovRange As_  
2 String, Optional druhyKluc As String)  
3 With ActiveWorkbook.Worksheets(menoHarku).Sort.SortFields  
4     .Clear  
5     .Add Key:=Range(klucRadenia), SortOn:=xlSortOnValues, Order:=xlAscending,_  
6 DataOption:=xlSortNormal  
7     If Len(druhyKluc) > 0 Then  
8         .Add Key:=Range(druhyKluc), SortOn:=xlSortOnValues, Order:=xlAscending,_  
9 DataOption:=xlSortNormal  
10 End If  
11 End With  
12 With ActiveWorkbook.Worksheets(menoHarku).Sort  
13     .SetRange Range(nazovRange)  
14     .Header = xlNo  
15     .MatchCase = False  
16     .Orientation = xlTopToBottom  
17     .SortMethod = xlPinYin  
18     .Apply  
19 End With  
20 End Function  
  
21 Function obnovZoznam(zoznam As Object, nazovRange As String)  
22 zoznam.RowSource = nazovRange  
23 End Function
```



## 11.Modul KT

```
1 Sub KT1 ()
2     Application.ScreenUpdating = False
3     Sheets("KT").Select
4     Range("A5").Select
5     Selection.CurrentRegion.Select
6     Selection.Clear
7     Sheets("KT").Select
8     Range("A5").Select
9     ActiveWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:= _
10         "data_pre_KT", Version:=xlPivotTableVersion12).CreatePivotTable _
11         TableDestination:="KT!R5C1", TableName:="Kontingenčná tabuľka2", _
12         DefaultVersion:=xlPivotTableVersion12
13     Sheets("KT").Select
14     Cells(5, 1).Select
15     With ActiveSheet.PivotTables("Kontingenčná tabuľka2").PivotFields("Značka")
16         .Orientation = xlRowField
17         .Position = 1
18     End With
19     ActiveSheet.PivotTables("Kontingenčná tabuľka2").CompactLayoutRowHeader = _
20         "Značka"
21     ActiveSheet.PivotTables("Kontingenčná tabuľka2").AddDataField ActiveSheet. _
22         PivotTables("Kontingenčná tabuľka2").PivotFields("Model"), "Počet z Model" _
23     ,xlCount
24     With ActiveSheet.PivotTables("Kontingenčná tabuľka2").PivotFields("Počet z _
25     Model")
26         .Caption = "Podiel"
27         .Calculation = xlPercentOfTotal
28         .NumberFormat = "0%"
29     End With
30     Sheets("KT").Range("D2").Value = "Prehľad zastúpenia jednotlivých značiek"
31     Range("D2").Select
32     Application.ScreenUpdating = True
33 End Sub

34 Sub KT2(interval As Integer)
35     Sheets("KT").Select
36     Range("A5").Select
37     Selection.CurrentRegion.Select
38     Selection.Clear
39     Sheets("KT").Select
40     Range("A5").Select
41     ActiveWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:= _
42         "data_pre_KT", Version:=xlPivotTableVersion12).CreatePivotTable _
43         TableDestination:="KT!R5C1", TableName:="Kontingenčná tabuľka4", _
44         DefaultVersion:=xlPivotTableVersion12
45     Sheets("KT").Select
46     Cells(5, 1).Select
47     With ActiveSheet.PivotTables("Kontingenčná tabuľka4").PivotFields("Min cena")
48         .Orientation = xlRowField
49         .Position = 1
50     End With
51     ActiveSheet.PivotTables("Kontingenčná tabuľka4").CompactLayoutRowHeader = _
52         "Cenový interval"
53     ActiveSheet.PivotTables("Kontingenčná tabuľka4").AddDataField ActiveSheet. _
54         PivotTables("Kontingenčná tabuľka4").PivotFields("Model"), "Počet z Model" _
55     ,xlCount
56     ActiveSheet.PivotTables("Kontingenčná tabuľka4").PivotFields("Počet z _
57     Model").Caption = "Počet"
58     Range("A9").Select
59     Selection.Group Start:=True, End:=True, By:=interval
60     Sheets("KT").Range("D2").Value = "Prehľad zastúpenia mobilných telefónov v _
61     cenových intervaloch " & interval & " CZK"
62     Range("A5").Select
63     Range("D2").Select
64     Selection.CurrentRegion.Select
65     Selection.Name = "data_pre_graf_cenove_intervaly"
```

## Návrh užívateľského rozhrania pri tvorbe aplikácii vo VBA

---

```
66 Application.ScreenUpdating = True
67 End Sub

68 Sub KT3(vlastnost As String)
69 Sheets("KT").Select
70 Range("A6").Select
71 Selection.CurrentRegion.Select
72 Selection.Clear
73 Range("A5").Select
74 ActiveWorkbook.PivotCaches.Create(SourceType:=xlDatabase, SourceData:= _
75     "data_pre_KT", Version:=xlPivotTableVersion12).CreatePivotTable _
76     TableDestination:="KT!R5C1", TableName:="Kontingenčná tabuľka2", _
77     DefaultVersion:=xlPivotTableVersion12
78 Sheets("KT").Select
79 Cells(5, 1).Select
80 ActiveWorkbook.ShowPivotTableFieldList = True
81 With ActiveSheet.PivotTables("Kontingenčná tabuľka2").PivotFields(vlastnost)
82     .Orientation = xlRowField
83     .Position = 1
84 End With
85 If frmVyberVlastnostiZariadeni.cmbVlastnostZariadenia = "TypOS" Then
86     ActiveSheet.PivotTables("Kontingenčná tabuľka2").CompactLayoutRowHeader = _
87 "Operačný systém"
88 Else
89     ActiveSheet.PivotTables("Kontingenčná tabuľka2").CompactLayoutRowHeader = _
90 "Funkčnosť"
91 End If
92 With ActiveSheet.PivotTables("Kontingenčná tabuľka2")
93     .AddDataField ActiveSheet.PivotTables("Kontingenčná_
94 tabuľka2").PivotFields("O2"), "Počet z O2", xlCount
95     .PivotFields("Počet z O2").Caption = "O2 počet"
96     .AddDataField ActiveSheet.PivotTables("Kontingenčná_
97 tabuľka2").PivotFields("Vodafone"), "Počet z Vodafone", xlCount
98     .PivotFields("Počet z Vodafone").Caption = "Vodafone počet"
99     .AddDataField ActiveSheet.PivotTables("Kontingenčná_
100 tabuľka2").PivotFields("T-Mobile"), "Počet z T-Mobile", xlCount
101     .PivotFields("Počet z T-Mobile").Caption = "T-Mobile počet"
102 End With
103 Sheets("KT").Range("D2").Value = "Prehľad zariadení podľa vlastnosti " &_
104 vlastnost
105 Range("D2").Select
106 End Sub
```

## 12.Modul rozlisenie

```
1  Declare Function GetSystemMetrics Lib "user32" (ByVal nIndex As Long) As Long
2  Public Const SM_cxscreen = 0
3  Public Const SM_cyscreen = 1
4  Sub nastavRozlisenie()
5  Dim sirka
6  Dim vyska
7  Dim zoomKoef
8  Dim ws As Worksheet
9  sirka = GetSystemMetrics(SM_cxscreen)
10 vyska = GetSystemMetrics(SM_cyscreen)
11 zoomKoef = Application.ActiveWindow.Zoom
12 For Each ws In Worksheets
13     Select Case sirka
14         Case "1920"
15             zoomKoef = 110
16         Case "1680"
17             zoomKoef = 120
18         Case "1440"
19             zoomKoef = 100
20         Case "1280"
21             zoomKoef = 91
22         Case "1280"
23             zoomKoef = 90
24         Case "1280"
25             zoomKoef = 90
26         Case "1280"
27             zoomKoef = 90
28         Case "1280"
29             zoomKoef = 83
30         Case "1152"
31             zoomKoef = 83
32         Case "1024"
33             zoomKoef = 75
34         Case "800"
35             zoomKoef = 58
36         Case Else
37             zoomKoef = 50
38     End Select
39 Next ws
40 Application.ActiveWindow.Zoom = zoomKoef
41 End Sub
```