

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta – Katedra fyziky

Visual Basic pro aplikace - využití v bezpečnostních  
systémech

Bakalářská práce

## Anotace

Tato práce se zabývá využitím programovacího prostředí „Visual Basic pro aplikace“ v bezpečnostních systémech, především v aplikacích pro kontrolu vstupu, popř. docházky. Vytvořený program slouží pro rychlou orientaci v záznamech těchto bezpečnostních systémů v prostředí Microsoft Excel, vyhledávání podle uživatelem nastaveného filtru.

### Klíčová slova:

Visual Basic pro aplikace, Microsoft Excel, bezpečnostní systém, systém kontroly vstupu, docházkový systém

## Abstract

This paper deals with application programming environment ,Visual Basic for Applications‘ in security systems, particularly in applications for access control, or attendance. The finished program is for quick reference in the records of these safety systems in Microsoft Excel, set by user search filter.

### Keywords:

Visual Basic for Applications, Microsoft Excel, Security System, System access Kontrol, Attendance System

Prohlašuji, že jsem svoji bakalářskou práci vypracoval samostatně, pouze s použitím zdrojů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Českých Budějovicích dne 22. listopadu 2010

.....  
podpis

## Poděkování

Děkuji Ing. Michalovi Šerému, vedoucímu této bakalářské práce, za odborné vedení, připomínky a cenné rady, které mi při její tvorbě poskytl. Především děkuji mé rodině, zejména manželce za její podporu a trpělivost po celou dobu studia.

---

Úvod.....	6
1 Základní pojmy.....	7
1.1 Visual Basic pro aplikace v bezpečnostních systémech .....	7
1.2 Visual Basic pro aplikace.....	8
2 Popis programu.....	9
2.1 Zákon č. 101/2000 Sb. ....	9
2.2 Vygenerování vstupních dat.....	11
2.3 List „zdroj_dat“ .....	11
2.4 List „vygenerovany“ .....	11
2.4.1 Tlačítko „GENERUJ“ .....	12
2.4.2 Tlačítko „FILTROVÁNÍ“ .....	13
2.4.3 Tlačítko „DOCHÁZKA“.....	15
3 Popis programu.....	17
3.1 Vygenerování vstupních dat.....	17
4 Vytvoření měsíčního výkazu - docházky .....	33
5 Praktická ukázka.....	42
5.1 Generování záznamů.....	42
5.2 Filtrování, práce se záznamy.....	43
5.3 Docházka, podklady pro výpočet mzdy .....	45
6 Závěr.....	47
7 Seznam použitých zdrojů .....	48

## Úvod

Problematikou spojenou s návrhem, instalací a provozováním bezpečnostních systémů se zabývá mnoho let. Tyto systémy byly dříve výsadou pouze velkých podniků, to předurčovala především jejich cena. Vzhledem ke klesání cen elektroniky a dostupnosti nových technologií se sféra zájmu výrobců rozšiřuje i na menší systémy. Tyto lze s výhodou instalovat až už do rodinných domů, nebo menších provozů. Většinou jednou z nevýhod je absence jakéhokoliv softwarového vybavení, které se u velkých systému stalo samozřejmostí. A tento fakt mě přivedl na myšlenku vyvinout pro menší systémy řešení rychlého vyhledávání dat a přípravy podkladů pro zpracování mezd.

Prostředí pro vytvoření aplikace jsem si vybral Microsoft Office Excel a Visual Basic pro aplikace. MS Excel proto, že většina systému umožňuje export dat ve formátu, s nímž umí pracovat jmenovaný software. A Visual Basic pro jeho universálnost a modulárnost. Tento software je od roku 1996 je zdarma obsažen v kancelářském balíku Microsoft Office.

Předkládaná práce čtenáři popíše jakým způsobem aplikaci ovládat a jaká očekávat data na výstupu. Díky komentovaným ukázkám zdrojového kódu získá přehled, jak program pracuje a v praktické ukázce se dozví, čeho se vyvarovat.

Tento program lze pokládat za alfa verzi, která po nasazení v reálné aplikaci, přizpůsobení potřebám klienta bude zpracovávat výstupní data z přístupového či docházkového systému.

# 1 Základní pojmy

## 1.1 Visual Basic pro aplikace v bezpečnostních systémech

Bezpečnostní systémy můžeme chápat jako komplex elektronických zařízení, majících za úkol ochranu zdraví a majetku. Z tohoto pohledu je lze rozdělovat například takto:

- Elektronické zabezpečovací systémy (EZS)
- Kamerové systémy (CCTV)
- Elektrické požární systémy (EPS), atp.
- Přístupové systémy (ACS, popř. SKV)
- Docházkové systémy

Trendem současnosti je tyto systémy slučovat a ovládat z jednoho místa. Potom hovoříme o nadstavbovém systému. Nadstavbový systém je zpravidla software, který při použití vhodného hardwaru integruje bezpečnostní systémy, třeba i různých výrobců. Vzhledem k cenám elektronických technologií zažívají velký rozmach i nadstavbové systémy pro domácnost.

Elektronický zabezpečovací systém je určen především pro ochranu majetku, včasnou detekci narušení střežených zón a následného vyrozumění ať už formou SMS na mobilní telefon, nebo odesláním varovné zprávy na pracoviště bezpečnostní služby. Základem EZS je ústředna, ke které jsou připojeny různé detektory.

Nepostradatelnou součástí všech větších bezpečnostních systémů jsou kamerové systémy. Poskytují rychlou informaci o situaci ve střeženém prostoru, jsou nezbytné pro koordinaci zásahu bezpečnostních složek a ze záznamového zařízení lze následně vyhodnotit poplachovou událost. V současnosti převládají instalace formátu IP (anglicky Internet Protokol). Jejich nespornou výhodou je dostupnost odkudkoliv po síti Internet.

Nedílnou a mnohdy nejdůležitější součástí jsou elektronické požární systémy. EPS disponuje mnoha technologiemi jak včas rozpoznat požár pokud možno již v jeho zárodku, tím zamezit škodám na majetku, ale především ztrátám na životě.

Přístupovým systémem rozumíme aplikaci, pomocí níž je provozovatel takového systému schopen efektivně řídit, popř. regulovat pohyb osob ve střeženém prostoru. Celkem jednoduchým způsobem lze skupině osob, zaměstnanců povolit anebo naopak zakázat vstup do citlivých míst provozu, vytvářet časové rozvrhy a podmínky vstupu atd. Princip spočívá v tom, že zaměstnancům přidělí např. identifikační karty, pomocí nich se přihlašují na snímačích u vstupu do střeženého prostoru a na základě přidělených práv je vstup umožněn nebo zakázán.

Na podobném principu jako přístupový systém pracuje i docházkový. Rozdíl je v tom, že po identifikaci kartou na snímači nedojde k otevření dveří, ale záznam je uložen v elektronické podobě „registračních hodin“. Uložená data se následně zpracovávají pro vytvoření měsíčních výkazu pro mzdovou účtárnu atp.

Data z kartových systémů jsou archivována a následně zpracovávána podle požadavků provozovatele. K tomuto účelu jsou vyvíjeny softwarové aplikace, které jsou většinou určeny pro velké kartové systémy nebo jsou součástí nadstavbových aplikací. Tomu

také odpovídá jejich cena. Pokud se menší firma rozhodne provozovat podobný software, využívá řádově jednotky procent možného výkonu. Taková aplikace se potom stává nerentabilní. I proto vznikl program, prezentovaný v této práci. Nabízí řešení pro menší přístupové a docházkové aplikace.

Program byl vytvořen ve Visual Basic for Applications (Visual Basic pro aplikace), v tabulkovém procesoru MS Excel, protože většina přístupových nebo docházkových systému umožňuje export dat do formátu \*.csv, tedy do formátu se kterým MS Excel pracuje. V této aplikaci lze s daty efektivně pracovat. Lze je jednoduše třídit, vyhledávat na základě uživatelského nastavení filtru atp. Výstupy z takto filtrovaných dat mohou sloužit jako podklady pro rychlé vyhodnocení pohybu osob ve střeženém prostoru zaměstnavatelem, pro potřeby Policie ČR, SÚJB (Státní úřad pro jadernou bezpečnost) atp.

## 1.2 Visual Basic pro aplikace

Visual Basic pro aplikace (anglicky Visual Basic for Applications, popř. VBA) je programovací jazyk, který je společným pro několik aplikací balíku MS Office. Používá se především tam, kde nabídka standardních funkcí končí nebo pokud provádíme opakovaně více funkcí, k automatizaci procesu [1]. Standardně se v této situaci nabízí využití makra, ale použití VBA ve srovnání s makry nabízí vyšší výkon, snáze se modifikuje a spravuje.

VBA je tzv. objektově orientovaným jazykem. Představme si MS Office jako stavebnici s kostkami, objekty. Můžeme říct, že každá kostka má své Vlastnosti (Properties) jako je tvar nebo barva, na které lze uplatnit Metodu (Method), tedy provést s kostkou určitou činnost, například přesunout. Některé objekty, kostky mohou reagovat na Událost (Event) kupříkladu klepnutí myši. Abychom pracovali vždy s požadovanou kostkou, objektový model má svou hierarchii. Pokud tedy chceme v MS Excelu pracovat s buňku A10 v aktivním sešitu aplikace na listě pojmenovaném List2, bude kód vypadat takto:

```
Application.ActiveWorkbook.Sheets("List2").Range("A10")
```

kde:	<code>Application</code>	je ta a většinou jediná spuštěná aplikace, ve které se kód nachází
	<code>ActiveWorkbook</code>	aktivní sešit
	<code>Sheets("List2")</code>	list pojmenovaný List2
	<code>Range("A10")</code>	rozsah, buňka A10



Projekt, prostředí VBA ve kterém se vytváří kód lze dělit na tři základní části:

- Objects obsahuje listy Excelu
- Forms obsahuje Formuláře tohoto sešitu
- Modules obsahuje Moduly, do kterých se zaznamenávají a zapisují makra nebo funkce

## 2 Popis programu

Program, který je v této práci prezentován lze rozdělit na tři hlavní části:

- Vygenerování vstupních dat (bude vysvětleno níže)
- Filtrování dat (práce s daty jako v přístupovém systému)
- Vytvoření měsíčního výkazu (práce s daty jako v docházkovém systém)

Pro osvětlení struktury programu je důležité na začátku vysvětlit několik pojmů z platné legislativy ČR.

### 2.1 Zákon č. 101/2000 Sb.

Vzhledem k tomu, že výstupní údaje z bezpečnostních systémů lze považovat za osobní, je nutné s nimi i tak nakládat a bez souhlasu subjektu, osoby, ke které náleží, dále zpracovávat. Tuto problematiku řeší Zákon č. 101/2000 Sb., o ochraně osobních údajů [2]. Zde jsou uvedena některá nejdůležitější ustanovení.

#### § 3

##### Působnost zákona

(1) Tento zákon se vztahuje na osobní údaje, které zpracovávají státní orgány, orgány územní samosprávy, jiné orgány veřejné moci, jakož i fyzické a právnické osoby.

(2) Tento zákon se vztahuje na veškeré zpracovávání osobních údajů, ať k němu dochází automatizovaně nebo jinými prostředky.

#### § 4

##### Vymezení pojmů

a) osobním údajem jakákoliv informace týkající se určeného nebo určitelného subjektu údajů. Subjekt údajů se považuje za určený nebo určitelný, jestliže lze subjekt údajů přímo či nepřímo identifikovat zejména na základě čísla, kódu nebo jednoho či více prvků, specifických pro jeho fyzickou, fyziologickou, psychickou, ekonomickou, kulturní nebo sociální identitu,

e) zpracováním osobních údajů jakákoliv operace nebo soustava operací, které správce nebo zpracovatel systematicky provádějí s osobními údaji, a to automatizovaně nebo jinými prostředky. Zpracováním osobních údajů se rozumí zejména shromažďování, ukládání na nosiče informací, zpřístupňování, úprava nebo pozměňování, vyhledávání, používání, předávání, šíření, zveřejňování, uchovávání, výměna, třídění nebo kombinování, blokování a likvidace,

f) shromažďováním osobních údajů systematický postup nebo soubor postupů, jehož cílem je získání osobních údajů za účelem jejich dalšího uložení na nosič informací pro jejich okamžité nebo pozdější zpracování,

l) zveřejněným osobním údajem osobní údaj zpřístupněný zejména hromadnými sdělovacími prostředky, jiným veřejným sdělením nebo jako součást veřejného seznamu,

## § 5

(2) Správce může zpracovávat osobní údaje pouze se souhlasem subjektu údajů. Bez tohoto souhlasu je může zpracovávat,

a) jestliže provádí zpracování nezbytné pro dodržení právní povinnosti správce

b) jestliže je zpracování nezbytné pro plnění smlouvy, jejíž smluvní stranou je subjekt údajů, nebo pro jednání o uzavření nebo změně smlouvy uskutečněné na návrh subjektu údajů,

c) pokud je to nezbytně třeba k ochraně životně důležitých zájmů subjektu údajů. V tomto případě je třeba bez zbytečného odkladu získat jeho souhlas. Pokud souhlas není dán, musí správce ukončit zpracování a údaje zlikvidovat,

d) jedná-li se o oprávněně zveřejněné osobní údaje v souladu se zvláštním právním předpisem. Tím však není dotčeno právo na ochranu soukromého a osobního života subjektu údajů,

e) pokud je to nezbytné pro ochranu práv a právem chráněných zájmů správce, příjemce nebo jiné dotčené osoby; takové zpracování osobních údajů však nesmí být v rozporu s právem subjektu údajů na ochranu jeho soukromého a osobního života,

f) pokud poskytuje osobní údaje o veřejně činné osobě, funkcionáři či zaměstnanci veřejné správy, které vypovídají o jeho veřejné anebo úřední činnosti, o jeho funkčním nebo pracovním zařazení, nebo,

g) jedná-li se o zpracování výlučně pro účely archivnictví podle zvláštního zákona.

## 2.2 Vygenerování vstupních dat

Jak je patrné z výše citovaného zákona bylo nutné z praktického důvodu vstupní data vygenerovat.

Zdrojový soubor MS Excel obsahuje několik listů, některé jsou nutné pro řádný běh programu, jiné jsou vytvářeny jako důsledek práce s daty, například při vyhodnocování docházky.

## 2.3 List „zdroj\_dat“

Zde jsou uložena data nezbytná pro vygenerování vstupních záznamů, která jsou následně zpracovávána. Konkrétně je to 200 křestních jmen a stejný počet příjmení jak pro muže, tak pro ženy [3], [4], [5]. Dále je to 30 názvů čteček karet. Pro jednoduchost byly pojmenovány „čtečka číslo 01“ až „čtečka číslo 30“. Jde v podstatě o kontrolované vstupy, dveře, u kterých je instalovaná čtečka identifikačních karet. Jak už bylo zmíněno, systém po identifikaci buď umožní vstup odblokováním zámku dveří anebo naopak vstup znemožní. Ve skutečné aplikaci jsou tyto čtečky popsány charakteristickým textem. Například číslem kontrolovaného vstupu, nebo číslem dveří popřípadě názvem budovy.

## 2.4 List „vygenerovany“

Na tento list se po vygenerování dat ukládají záznamy z „přístupového systému“. Na obr. č. 1 jsou zobrazena tlačítka pro ovládání celého programu.

	A	B	C	D	E	F	G	
1	Jméno	Příjmení	čtečka	Datum	Čas	generuj	filtrování	docházka
2								
3								

Obr. č. 1 Ovládací tlačítka a hlavička tabulky

### 2.4.1 Tlačítko „GENERUJ“

Kliknutím na tlačítko „GENERUJ“ se spustí formulář s názvem „Vstupní\_formular“ viz obr. č. 2.

Obr. č. 2 Vstupni\_formular

Uživatel zvolí, kolik bude vygenerováno jmen mužů a kolik jmen žen (rozuměno dat typu *Jméno, Příjmení*), jaký je maximální počet průchodů pro každé jméno. Každému takto vytvořenému subjektu, který má své jméno a příjmení, program na základě náhodného čísla v rozmezí 1-30 přidělí název čtečky, podle dalšího náhodného čísla datum průchodu a nakonec opět podle náhodných čísel čas příchodu a odchodu. Vygenerovaný záznam pak může vypadat například jako na obr. č. 3.

	A	B	C	D	E
1	Jméno	Příjmení	čtečka	Datum	Čas
2	Čeněk	JANEČEK	čtečka číslo 09	25.8.2010	18:35:38
3	Čeněk	JANEČEK	čtečka číslo 01	26.8.2010	00:00:00

Obr. č. 3 Vygenerovaný záznam

### 2.4.2 Tlačítko „FILTROVÁNÍ“

Kliknutím na tlačítko „FILTROVÁNÍ“ se spustí formulář s názvem „*Filtrování*“ viz obr. č. 4.

Obr. č. 4 Formulář filtrování záznamů

Při spuštění formuláře program automaticky načte položky ComBox, neboli položky rozbalovacího formuláře ze sešitu „vygenerovaný“ a vzestupně seřadí. Formulář filtrování pracuje se sešitem „vygenerovaný“, to znamená, že se všechny nastavení filtru se na listě okamžitě projeví.

#### Tlačítko „FILTRUJ“

Funkci tlačítka si předvedeme nejlépe na příkladu.

Zadání : zjistit pohyb subjektu Janeček Čeněk vstupem čtečka číslo2.

Postup: vyplnit rozbalovací seznamy „Příjmení“, „Jméno“ a „Čtečka“, „Datum“ nechat prázdné. Po kliknutí na tlačítko „FILTRUJ“ zůstanou zobrazeny pouze záznamy, které vyhovují zadaným podmínkám.

#### Tlačítko „ZRUŠ FILTR“

Zruší se filtr na listě „vygenerovaný“.

#### Tlačítko „TISK“

Program vytiskne aktivní obsah listu „vygenerovaný“ na tiskárnu nainstalovanou jako výchozí. Výhodné je tisknout do formátu \*.pdf, který lze následně šířit elektronicky, např. emailem. Proto je v této práci uvažováno o virtuální tiskárně Adobe

PDF jako o tiskárně výchozí. Pokud bude aktivní filtr jako v předchozím případě (Tlačítko „FILTRUJ“), bude vytištěn seznam s podmínkami a datem tisku jako je na obr. č. 5.

Tlačítko „VYMAŽ“

Vymaže zadané hodnoty ve všech rozbalovacích seznamech.

Tlačítko „STORNO“

Zavře formulář „Filtrování záznamů“.

Filtrování ze dne 24.11.2010 21:44:29 podle těchto kritérií

Příjmení: JANEČEK

Jména: Čeněk

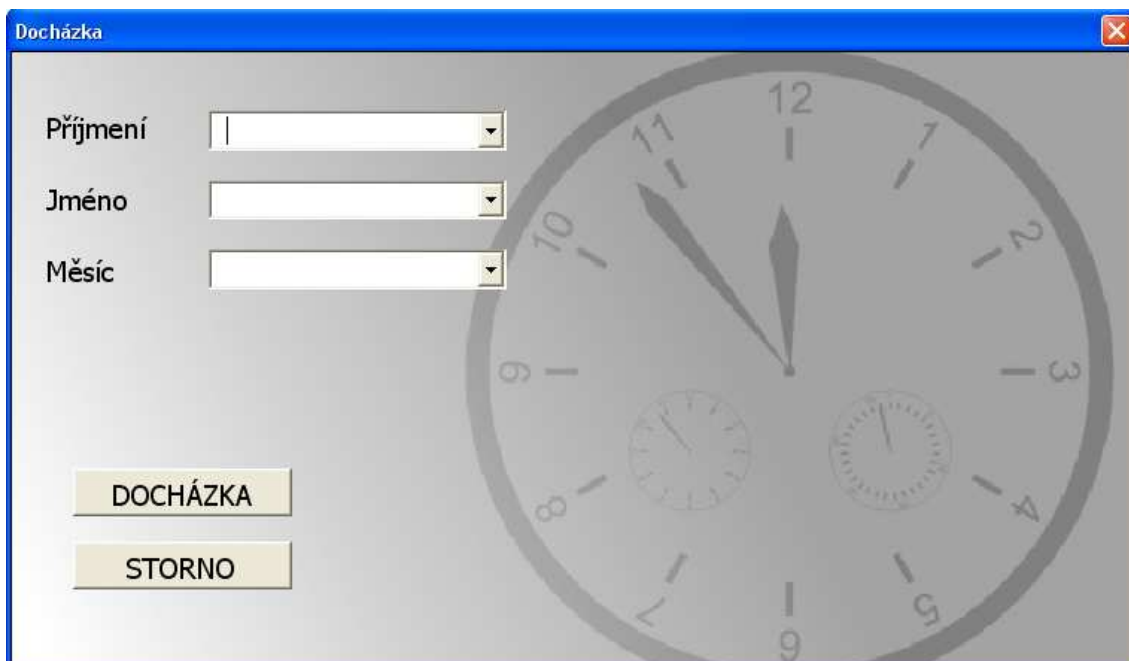
Čtečky: čtečka číslo 02

JMÉNO	PŘÍJMENÍ	ČTEČKA	DATUM	ČAS
Čeněk	JANEČEK	čtečka číslo 02	11.7.2010	01:02:12
Čeněk	JANEČEK	čtečka číslo 02	5.8.2010	01:23:17
Čeněk	JANEČEK	čtečka číslo 02	11.9.2010	04:56:40
Čeněk	JANEČEK	čtečka číslo 02	27.9.2010	06:33:22
Čeněk	JANEČEK	čtečka číslo 02	28.9.2010	08:45:36
Čeněk	JANEČEK	čtečka číslo 02	4.10.2010	09:06:38
Čeněk	JANEČEK	čtečka číslo 02	12.11.2010	11:19:53
Čeněk	JANEČEK	čtečka číslo 02	23.11.2010	12:20:05
Čeněk	JANEČEK	čtečka číslo 02	10.12.2010	18:10:07
Čeněk	JANEČEK	čtečka číslo 02	20.12.2010	20:49:39

Obr. č. 5 Tisk filtrování

### 2.4.3 Tlačítko „DOCHÁZKA“

Tlačítko docházka spouští formulář pro zadání parametrů k vytvoření podkladů pro výpočet mzdy – docházky (obr č. 6).



Obr. č. 6 Formulář docházka

Po zpracování dat, vypočtu odpracovaných hodin, přesčasových hodin a příplatků za práci o víkendu a v noci je v sešitě vytvořen nový list pojmenovaný ve formátu příjmení a jméno (obr. č. 7).



Obr. č. 7 Nový list docházky Janeček Čeněk

Na listu je nejprve vytvořena hlavička s údaji o subjektu, měsíci za který je docházka zpracována, sumarizací odpracovaných hodin a příplatků (obr. č. 8).

	A	B	C	D	E	F	G	H
1	JANEČEK	Čeněk						
2	Docházka	červenec 2010						
3								
4	Hodin celkem:	261,1						
5	Přesčas:	93,1						
6	Příplatek So+Ne:	105,9						
7	Příplatek noční:	56,4						
8								
9	<b>Datum</b>	<b>Příchod</b>	<b>Odchod</b>	<b>Hodin</b>	<b>Přesčas</b>	<b>So+Ne</b>	<b>Noční</b>	
10	4.7.2010	15:53:10						
11	4.7.2010		23:49:49	7,9	0,0	7,9	2,2	

Obr. č. 8 List Janeček Čeněk



## 3 Popis programu

### 3.1 Vygenerování vstupních dat

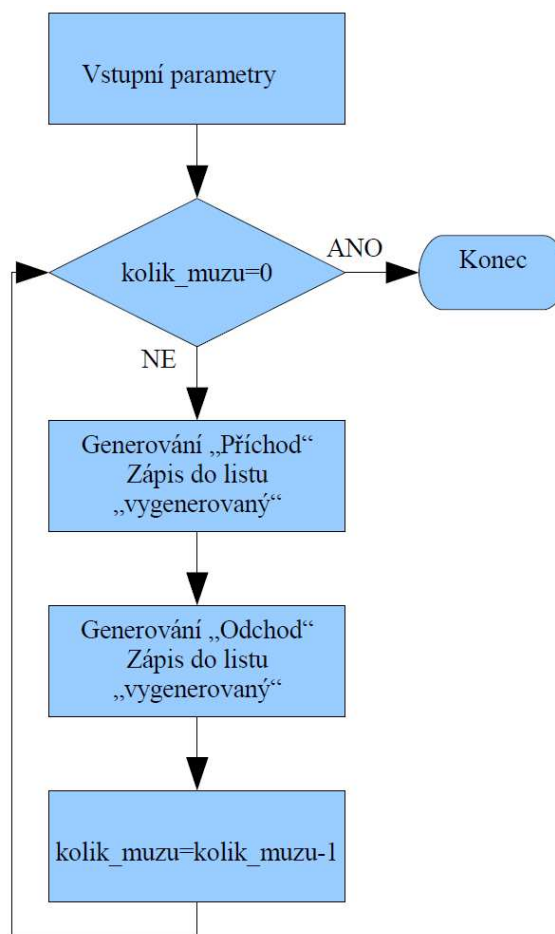
Ve skutečných bezpečnostních systémech by tato část programu byla zbytečná, protože vstupní data bychom jednoduše vyexportovali. Celá řada těchto systémů umožňuje export přímo ve formátu \*.csv, tedy ve formátu který podporuje i MS Excel. Jak už bylo popsáno výše, s takovými údaji nelze dále nakládat bez souhlasu subjektu. Proto tedy tento nezbytný krok.

Po otevření vstupního formuláře, viz obr. č. 2, zadá uživatel tyto parametry:

- Kolik bude mužů:                    program vygeneruje patřičné množství mužských jmen
- Kolik bude záznamů:                zde se udává maximální počet záznamů, průchodů. Aby se vstupní data co nejvíce blížila realitě, program vygeneruje pro každý subjekt jiný počet záznamů. Tato zadaná hodnota je chápána jako maximum.

Stejně podmínky jako v předešlém případě platí pro záznamy žen.

Po kliknutí na tlačítko GENERUJ, program na základě vnitřních podmínek vytvoří „export průchodů“. Pro každé jméno, resp. datum průchodu budou vytvořeny dva záznamy (příchod, odchod). Vnitřní podmínky pro generování záznamů mužů blíže popisuje vývojový diagram (na obr. č. 9), ale především ukázky ze zdrojového kódu programu. Program tedy vygeneruje jméno a příjmení, následně datum, čas a číslo čtečky-tento bude chápán jako příchod. K tomuto záznamu automaticky vytvoří ještě jeden – odchod. Tento záznam je vázán na předchozí, protože z něho vychází. Nejprve je vygenerován čas odchodu, porovnán s příchozím časem, případně upraven společně s datem. Tato podmínka bude blíže popsána v části zdrojového kódu programu. Stejným způsobem jsou vytvořena data „muži“ i „ženy“.



Obr. č. 9 Generování záznamů

Tuto část programu detailně prezentuje následující ukázka zdrojového kódu Visual Basic pro aplikace. Nejprve jsou vymazána předchozí data.

- nastavení oblasti, která bude vymazána.
- první prázdný řádek

```

Do While Not IsEmpty(ActiveSheet.Cells(radek, sloupec))
    Sheets("vygenerovany").Cells(radek, sloupec).Value = ""
  
```

- první prázdný sloupec

```

Do While Not IsEmpty(ActiveSheet.Cells(radek, sloupec + 1))
    Sheets("vygenerovany").Cells(radek, sloupec + 1).Value = ""
    sloupec = sloupec + 1
Loop
radek = radek + 1
sloupec = 1
Loop
  
```

- vstupní nastavení proměnných

```
kolik_muzu = 0
kolik_zen = 0
kolik_zaznamu_muzu = 0
kolik_zaznamu_zeny = 0
radek_cil = 2
```

- načtení proměnných z Vstupni\_formular

```
kolik_muzu = Vstupni_formular.TextBox_muzu
kolik_zen = Vstupni_formular.TextBox_zen
kolik_zaznamu_muzu = Vstupni_formular.TextBox_zaznamu_muži
kolik_zaznamu_zeny = Vstupni_formular.TextBox_zaznamu_zeny
Unload Me
```

- generování záznamu mužů

```
pocet_zaznamu = 2 *olik_zaznamu_muzu
```

- dvojnásobek proto, že se zapisuje příchod i odchod

```
Do While Not kolik_muzu = 0
```

- příkaz ReDim smaže případnou tabulku a nastaví ji na deklarované hodnoty

```
ReDim Preserve tabulka(1 To pocet_zaznamu)
```

- vygenerování čísla a jména muže
- program vygeneruje náhodné číslo do 200, které koresponduje s číslem řádku v sešitu „zdroj\_dat“

```
radek_zdroj = Int((200 * Rnd) + 1)
Jmeno = Sheets("zdroj_dat").Cells(radek_zdroj, 1).Value
```

- obdobným způsobem je vytvořeno příjmení

```
radek_zdroj = Int((200 * Rnd) + 1)
prijmeni = Sheets("zdroj_dat").Cells(radek_zdroj, 3).Value
```

- a nakonec jméno čtečky

```
kolik_cyklu = Int((olik_zaznamu_muzu * Rnd) + 1)
```

- pomocná proměnná j určuje zda je cyklus na začátku, proměnná shoda je nastavena podle toho jestli se datum již vyskytuje ve vygenerovaném seznamu. Program kontroluje, zda vygenerované datum již není na listě zapsané. Je ochrana proti zápis dvou příchodů v jeden den.

```
j =olik_cyklu
shoda = False
x = 1
Do While Notolik_cyklu <= 0
x = x + 1
```

- proměnná `x` je zavedena kvůli ochraně před cyklováním. Pokud by z nějakého důvodu nebyla naplněna podmínka `kolik_cyklu <= 0` program by začal cyklovat, proto se při každém průchodu cyklem `x` inkrementuje.

```
If x > 1000 Then
```

- zápis dat do listu „vygenerovany“

```
sloupec_cil = 1
Sheets("vygenerovany").Cells(radek_cil, sloupec_cil).Value =
    Jmeno
sloupec_cil = sloupec_cil + 1
Sheets("vygenerovany").Cells(radek_cil, sloupec_cil).Value =
    prijmeni
sloupec_cil = sloupec_cil + 1
```

- vygenerování názvu čtečky

```
radek_zdroj = Int((30 * Rnd) + 1)
ctecka = Sheets("zdroj_dat").Cells(radek_zdroj, 5).Value
```

- datum se generuje jako celé kladné číslo, které se formátem na listu převádí na datum. Přičemž číslo 1 reprezentuje datum 1. 1. 1900, další udávají počet dnů od tohoto data. Číslo 40 179 převedené na datum je 1. 1. 2010. Funkce `Rnd` vrací náhodné číslo mezi 0.01 - 0.99, které je násobené počte dnů v roce a je k němu přičtené datum 1. 1. 2010 v číselném formátu (40179). `CDate` vrací formát datum, výsledkem je tedy datum v rozmezí 4. 1. 2010 - 28. 12. 2010.

```
datum = CDate(Int((365 * Rnd) + 40179))
```

- obdobným způsobem se vytváří i čas. Po převodu na formát čas, kde číslo 0 je 00:00:00 a číslo 1 je 24:00:00, program generuje čas v rozmezí 00:14:24 až 23:45:36.

```
Cas = Rnd()
```

- program kontroluje, zda vygenerované datum již není na listě zapsané. Je ochrana protizápis dvou příchodu v jeden den. Při běhu 1. cyklu není nutné kontrolovat shodu dat

```
If j <>olik_cyklu Then
```

- program hledá shodné datum a aktuálně vygenerovaným

```
For i = 2 To radek_cil
shodnedatum = Sheets("vygenerovany").Cells(i, 4).Value
```

- pokud najde shodu, porovná jméno a příjmení. Tímto způsobem vyloučí dva příchody jednoho subjektu v jeden den.

```
If shodnedatum = datum Then
    jmeno2 = Sheets("vygenerovany").Cells(i, 1).Value
```

```
If Jmeno = jmeno2 Then
    prijmeni2 = Sheets("vygenerovany").Cells(i, 2).Value
```

```

If prijmeni = prijmeni2 Then
    shoda = True
Exit For
End If
End If
End If
Next i
End If

```

- shoda=False, tzn. aktuální datum ještě nebylo vygenerované, program zapíše do listu „vygenerovaný“. Pokud by nastala shoda, program zápis přeskočí a vygeneruje nový datum.

```

If shoda <> True Then
Sheets("vygenerovaný").Cells(radek_cil, sloupec_cil).Value =
ctecka
sloupec_cil = sloupec_cil + 1
Sheets("vygenerovaný").Cells(radek_cil, sloupec_cil).Value =
datum
sloupec_cil = sloupec_cil + 1
Sheets("vygenerovaný").Cells(radek_cil, sloupec_cil).Value = Cas
radek_cil = radek_cil + 1
kolik_cyklu =olik_cyklu - 1

```

- 2. zápis (odchod) do nového listu

```

sloupec_cil = 1
Sheets("vygenerovaný").Cells(radek_cil, sloupec_cil).Value =
Jmeno
sloupec_cil = sloupec_cil + 1
Sheets("vygenerovaný").Cells(radek_cil, sloupec_cil).Value =
prijmeni
sloupec_cil = sloupec_cil + 1

```

- program vygeneruje novou čtečku

```

radek_zdroj = Int((30 * Rnd) + 1)
ctecka = Sheets("zdroj_dat").Cells(radek_zdroj, 5).Value
Sheets("vygenerovaný").Cells(radek_cil, sloupec_cil).Value =
ctecka
sloupec_cil = sloupec_cil + 2

```

- čas odchodu Cas2 se generuje tak, že se k času příchodu připočítá 8 hodin jako pracovní doba (0,33), vygeneruje se a přičte čas 2 hodiny jako přesčas (generovaný)

```

generovany = (Int(83 * Rnd) + 1) / 1000
Cas2 = generovany + 0.33 + Cas
sloupec_cil = sloupec_cil - 1

```

- generováním Cas2 může nabývat hodnot větších než 1, z pohledu docházky je to čas po půlnoci, tedy noční směna => upraví se datum odchodu

```

If Cas2 > 1 Then
    Cas2 = Cas2 - 1
    Sheets("vygenerovany").Cells(radek_cil, 5).Value = Cas2
    datum = CLng(datum)
    datum = CDate(datum + 1)
    shoda2 = False
    Sheets("vygenerovany").Cells(radek_cil, 4).Value = datum

```

- tím že se upraví datum, musí se zajistit, že nebude v kolizi s jiným záznamem. V podstatě se musí zajistit to, že čas odchodu jednoho záznamu nebude vyšší než čas příchodu jiného.

```

For i = 2 To radek_cil
    shodnedatum = Sheets("vygenerovany").Cells(i, 4).Value
    If shodnedatum = datum Then
        shoda2 = True
        Cas = Sheets("vygenerovany").Cells(i, 5).Value
        Exit For
    End If
Next i

```

- pokud nalezne shodu (shoda2 = True), pouze pro potřeby programu (na listě „vynenerovany“ zůstane beze změny) od kolizního času odečte 8 hodin, což je povinná přestávka. Pokud by cas nabýval záporných hodnot, nastaví cas2 na hodnotu 00:00:00, jinak vygeneruje nový čas (cas2) a opět ho porovná s takto upraveným časem. Nově vygenerovaný čas musí být v rozmezí 00:00:00 a upraveným časem

```

Cas = Cas - 0.33
If Cas <= 0 Then
    Cas2 = 0
End If

```

- protože je pravděpodobnost cyklování značná, je zavedena proměnná k. Ta je každým cyklem inkrementována a pokud splní podmínku  $k > 100$ , nastaví cas2 na hodnotu 00:00:00

```

k = 1
If Cas2 <> 0 Then
    If Cas < Cas2 Then
        Do While Cas > Cas2
            Cas2 = Rnd()
            k = k + 1
        End While
        If k > 100 Then
            Cas2 = 0
            Exit Do
        End If
    End If
    Sheets("vygenerovany").Cells(radek_cil, 5).Value = Cas2
End If
End If
End If
Sheets("vygenerovany").Cells(radek_cil, 4).Value = datum
Sheets("vygenerovany").Cells(radek_cil, 5).Value = Cas2

```

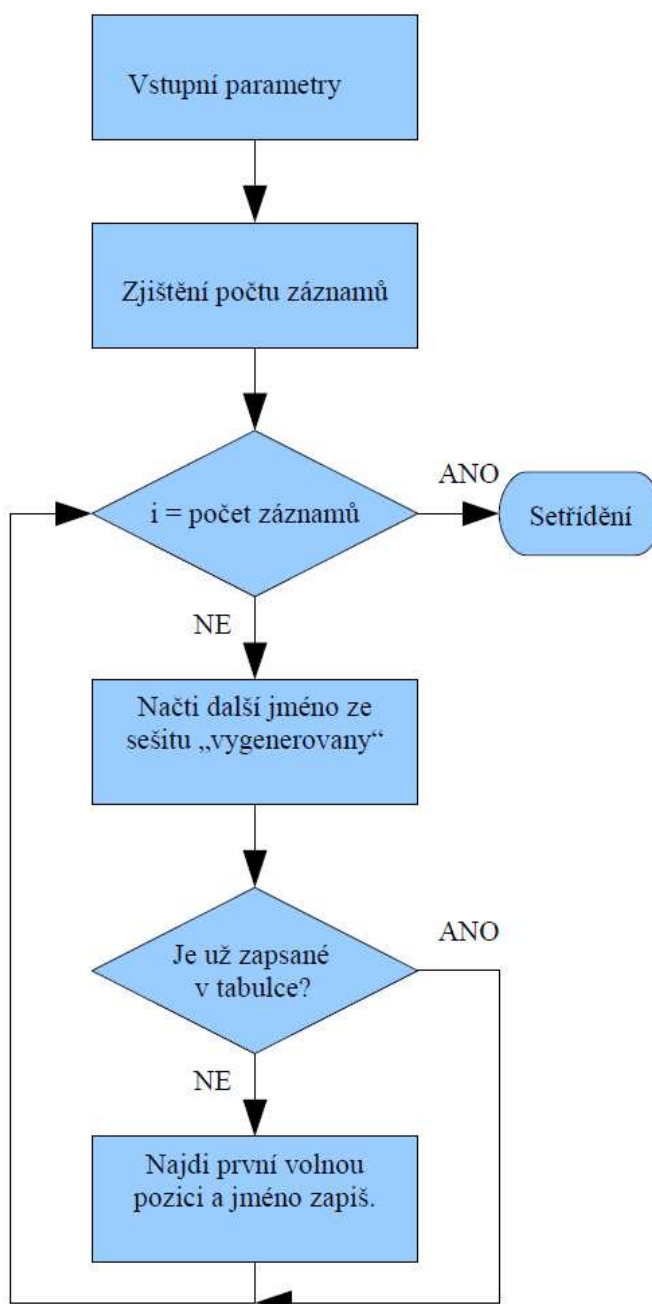
- nastaví se podmínky pro další cyklus

```
radek_cil = radek_cil + 1
kolik_cyklu = kolik_cyklu - 1
End If
shoda = False
End If
Loop
kolik_zaznamu = kolik_zaznamu - 1
kolik_muzu = kolik_muzu - 1
Loop
```

Stejným způsobem jsou generovány i záznamy žen. Jak může vypadat jeden záznam, zobrazuje obr. č. 3. Tato část programu byla zpracována za použití odborné literatury [6], [7], [9].

## Filtrování záznamů

Nejprve se načtou neduplicitní příjmení do tabulky. Program zjistí celkový počet záznamů, načte postupně každé příjmení, porovná je se záznamy v tabulce. Pokud nenajde shodu, zapíše je na poslední místo, resp. první volné místo. Pokud shodu najde, přeskočí zápis do tabulky, načte další v pořadí a porovná (viz obr. č. 10) Tímto způsobem načte i jména a čtečky.



Obr. č. 10 Načtení neduplicitních záznamů

- program zjistí kolik záznamů je na listě „vygenerovaný“ a od druhého řádku prochází seznam, dokud nenajde první prázdnou buňku



```

radek = 2
sloupec = 2
pocet_zaznamu = 0
Do While Not
IsEmpty(Sheets("vygenerovany").Cells(radek,sloupec))
pocet_zaznamu = pocet_zaznamu + 1
radek = radek + 1
Loop

```

- tato část programu načte do Comboxu všechny neduplicitní příjmení, stejný postup platí i pro jména a čtečky. Redimenzování tabulky nastaví meze a zároveň vymaže případná předchozí data.

```

ReDim Preserve tabulka(1 To pocet_zaznamu)
j = 1
radek = 2
sloupec = 2
jmen_v_tabulce = 0
i = 1

```

- načítání položek ze sešitu

```

For i = 1 To pocet_zaznamu Step 1
Jmeno = Sheets("vygenerovany").Cells(radek, sloupec)

```

- porovnávání položek v tabulce

```

For j = 1 To pocet_zaznamu

```

- příjmení už je v tabulce?

```

If tabulka (j) = Jmeno Then

```

- pokud není, zapíše jméno do tabulky a opustí cyklus

```

Exit For
End If
If tabulka(j) = Empty Then

```

- zápis na volnou pozici

```

tabulka(j) = Jmeno
jmen_v_tabulce = jmen_v_tabulce + 1

```

- pokud zapíše jméno do tabulky, opustí cyklus

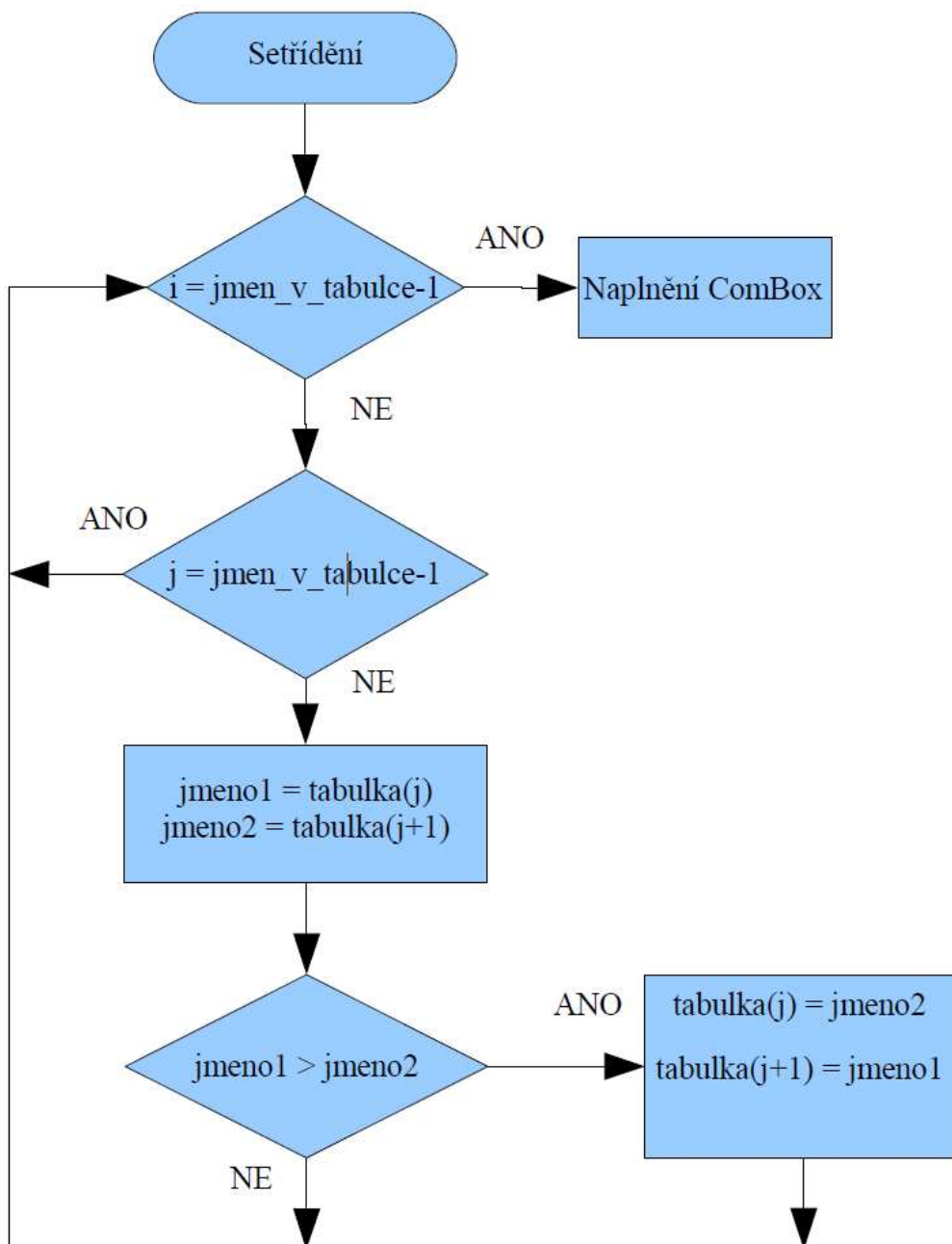
```

Exit For
End If
Next j
radek = radek + 1
Next i

```

Před zápisem dat do rozbalovacího seznamu jsou data abecedně seříděny. Zde byla použita metoda „probublávání“, někdy též Bubble Sort. Tato metoda, v tomto případě vzestupného třídění, spočívá v tom, že jsou porovnány sousední záznamy

v tabulce. Pokud je splněna podmínka, že výše uložený záznam je abecedně až za níže uloženým, tak jsou tyto záznamy prohozeny a je načten další záznam, který je porovnáván s tímto prohozeným. Tak jsou porovnány všechny záznamy s tím, že záznam, který je abecedně poslední „probublává“ tabulkou až na poslední místo (viz obr. č. 10).



Obr. č. 11 Setřídění záznamů

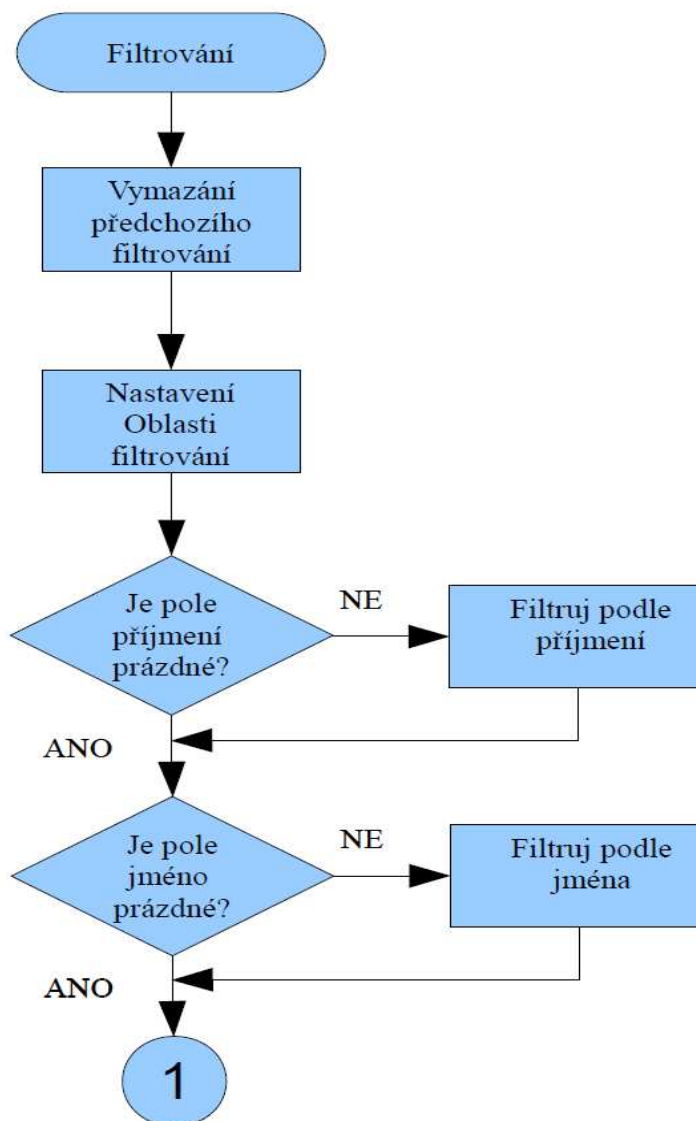
```

For i = 1 To jmen_v_tabulce - 1 Step 1
For j = 1 To jmen_v_tabulce - 1 Step 1
  jmeno1 = tabulka(j)

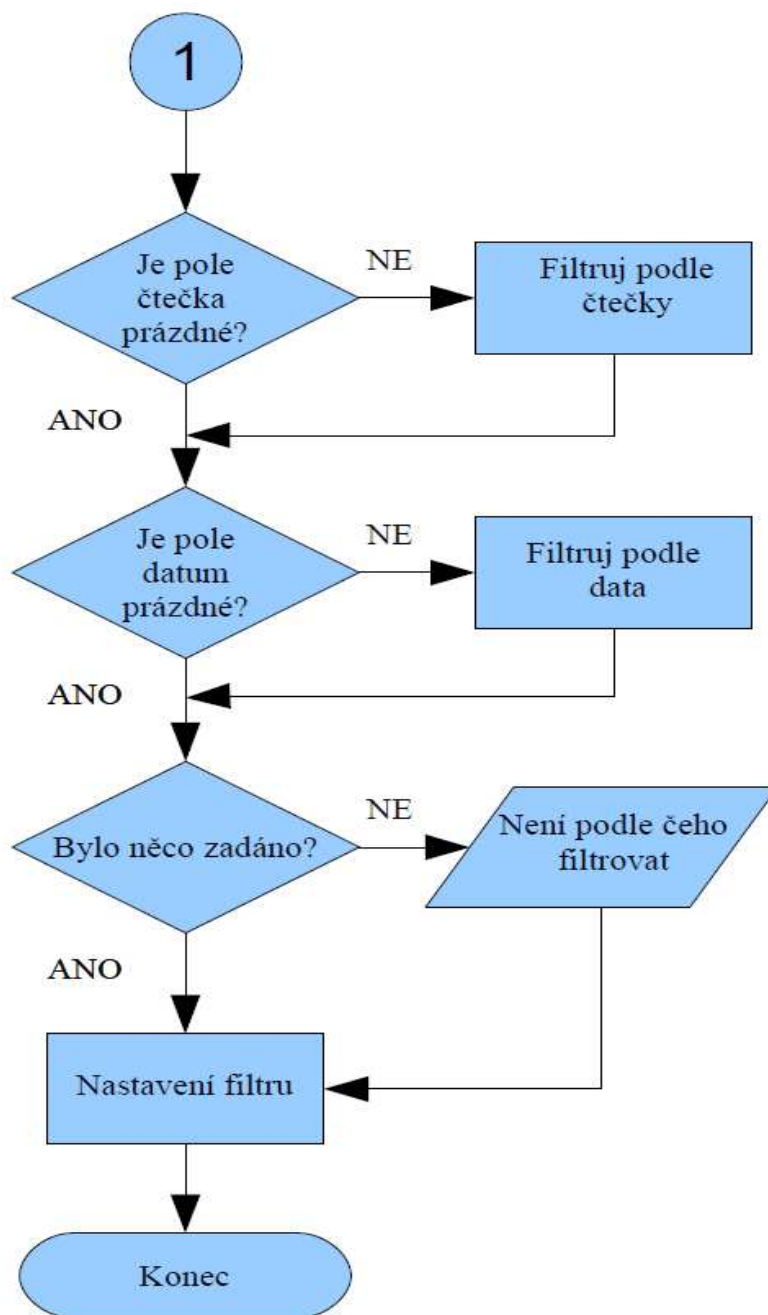
```

```
jmeno2 = tabulka(j + 1)
If jmeno1 > jmeno2 Then
    tabulka(j) = jmeno2
    tabulka(j + 1) = jmeno1
End If
Next j
Next i
```

Takto seříděné záznamy jsou zapsány do rozbalovacího seznamu. Stejný postup platí pro seznam příjmení, jmen i čteček. Uživatel následně z rozbalovacího seznamu vybere kritéria pro filtrování záznamů, viz obr. č. 12 a 13.



Obr. č. 12 Filtrování 1. část



Obr. č. 13 Filtrování 2. část

- vymazání předchozího filtru

```
ActiveSheet.AutoFilterMode = False
Do While Not
IsEmpty(Sheets("vygenerovany").Cells(radek,sloupec))
    pocet_zaznamu = pocet_zaznamu + 1
    radek = radek + 1
Loop
```

- nastavení aktuální oblasti filtrování

```
rozsah = "$A$2:$E$" & pocet_zaznamu
Range("A1:E1").Select
Selection.AutoFilter
```

- podle příjmení

```
If Filtrovani.Pole_prijmeni.Value <> Empty Then
ActiveSheet.Range(rozsah).AutoFilterField:=2,
Criterial:=Filtrovani.Pole_prijmeni.Value
```

- čítač je pro kontrolu, zda jsou zadána kritéria. Odčítá se od hodnoty 4, pokud je na konci podmínek rovna 4, nebylo zadáno žádné kritérium a nelze filtrovat.

```
citac = citac - 1
End If
```

- podle jména

```
If Filtrovani.Pole_jmeno.Value <> Empty Then
ActiveSheet.Range(rozsah).AutoFilter Field:=1,
Criterial:=Filtrovani.Pole_jmeno.Value
citac = citac - 1 'kontrola zda tady prošel
End If
```

- podle čtečky

```
If Filtrovani.Pole_ctecky.Value <> Empty Then
ActiveSheet.Range(rozsah).AutoFilter Field:=3,
Criterial:=Filtrovani.Pole_ctecky.Value
citac = citac - 1 'kontrola zda tady prošel
End If
```

- podle data

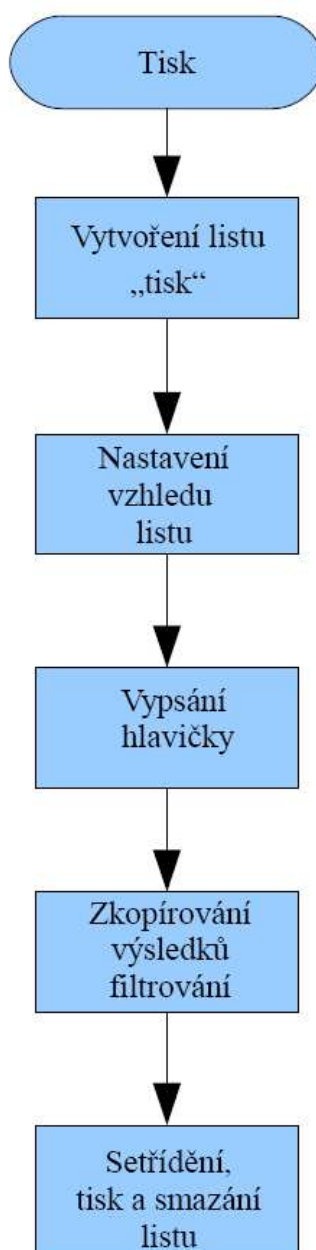
```
If Filtrovani.Pole_datum.Value <> Empty Then
datum se musí načíst do proměnné deklarované jako datum
datum = Filtrovani.Pole_datum.Value
ActiveSheet.Range(rozsah).AutoFilter Field:=4, Criterial:=datum
citac = citac - 1
End If
```

..

- kontrola zda je zadáno alespoň jedno kritérium

```
If citac = 4 Then
Unload Me
MsgBox "Nic nenalezeno..." & vbCrLf & "Není podle čeho
filtrovat", vbInformation + vbOKOnly, "Výsledek filtrování"
End If
```

Takto vyfiltrovaná data lze vytisknout do formátu \*.pdf podle nastaveného formuláře. Program vytvoří nový list, zapíše do hlavičky formuláře datum vytvoření souboru, podmínky, podle kterých se filtrovalo a samozřejmě výsledky filtrování zkopíruje z listu „vygenerovaný“. Provede tisk do \*.pdf s dotazem na místo uložení a potom se list smaže, viz obr. č. 14.



Obr. č. 14 Tisk výsledků filtrování

- vytvoření nového listu „tisk“

```
Sheets.Add After:=Sheets(Sheets.Count)
ActiveSheet.Name = "tisk"
```

- zápis do listu „tisk“ hlavičky podle čeho a kdy se filtrovalo

```
sloupec_cil = 1
radek_cil = 1
datum = Now ' systémové datum a čas
Sheets("tisk").Cells(radek_cil, sloupec_cil).Value = "Filtrování
ze dne " & datum & " podle těchto kritérií"
radek_cil = 3
```

```
If Filtrovani.Pole_prijmeni.Value <> Empty Then
Sheets("tisk").Cells(radek_cil, 1).Value = "Příjmení:"
Sheets("tisk").Cells(radek_cil,2).Value=Filtrovani.Pole_prijmeni
.Value
End If
```

```
If Filtrovani.Pole_jmeno.Value <> Empty Then
radek_cil = radek_cil + 1
Sheets("tisk").Cells(radek_cil, 1).Value = "Jména:"
Sheets("tisk").Cells(radek_cil,2).Value=Filtrovani.Pole_jmeno.
Value
End If
```

```
If Filtrovani.Pole_ctecky.Value <> Empty Then
radek_cil = radek_cil + 1
Sheets("tisk").Cells(radek_cil, 1).Value = "Čtečky:"
Sheets("tisk").Cells(radek_cil,2).Value =
Filtrovani.Pole_ctecky.Value
End If
```

```
If Filtrovani.Pole_datum.Value <> Empty Then
radek_cil = radek_cil + 1
datum = Filtrovani.Pole_datum.Value
Sheets("tisk").Cells(radek_cil, 1).Value = "Data:"
Sheets("tisk").Cells(radek_cil, 2).Value = datum
End If
```

```
Sheets("tisk").Cells(8, 1).Value = "JMÉNO"
Sheets("tisk").Cells(8, 2).Value = "PŘÍJMENÍ"
Sheets("tisk").Cells(8, 3).Value = "ČTEČKA"
Sheets("tisk").Cells(8, 4).Value = "DATUM"
Sheets("tisk").Cells(8, 5).Value = "ČAS"
```

- zápis do listu "tisk" výsledků filtrování

```
rozsah = "$A$2:$E$" & pocet_zaznamu
Sheets("vygenerovany").Range(rozsah).Copy
Sheets("tisk").Select
Range("A9").Select
ActiveSheet.Paste
```

- seřazení podle data a času

```

rozsah = "$D$8:$D$" & radek_cil
ActiveWorkbook.Worksheets("tisk").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("tisk").Sort.SortFields.Add
Key:=Range(rozsah), _
SortOn:=xlSortOnValues, Order:=xlAscending,
DataOption:=xlSortNormal

With ActiveWorkbook.Worksheets("tisk").Sort
.SetRange Range(rozsah)
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom
.SortMethod = xlPinYin
.Apply
End With

rozsah = "$E$8:$E$" & radek_cil
ActiveWorkbook.Worksheets("tisk").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("tisk").Sort.SortFields.Add
Key:=Range(rozsah), SortOn:=xlSortOnValues, Order:=xlAscending,
DataOption:=xlSortNormal

With ActiveWorkbook.Worksheets("tisk").Sort
.SetRange Range(rozsah)
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom
.SortMethod = xlPinYin
.Apply
End With

```

- tisk do pdf

```

rozsah = "$A$1:$E$" & radek_cil - 1
Range(rozsah).Select
ExecuteExcel4Macro "PRINT(1,,,1,,,,,,,,,2,,,TRUE,,FALSE)"

```

- potlačení dotazu zda list smazat

```

Application.DisplayAlerts = False
Sheets("tisk").Delete
Application.DisplayAlerts = True

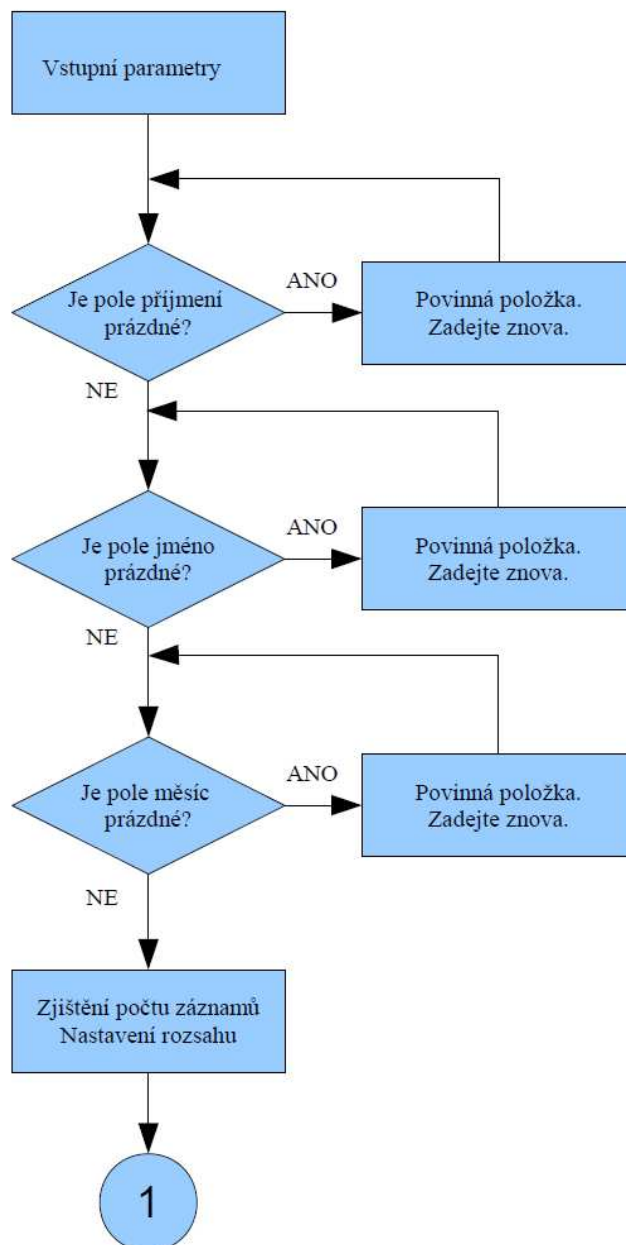
```

Tato část programu byla zpracována za použití odborné literatury [6], [7], [10], [11].

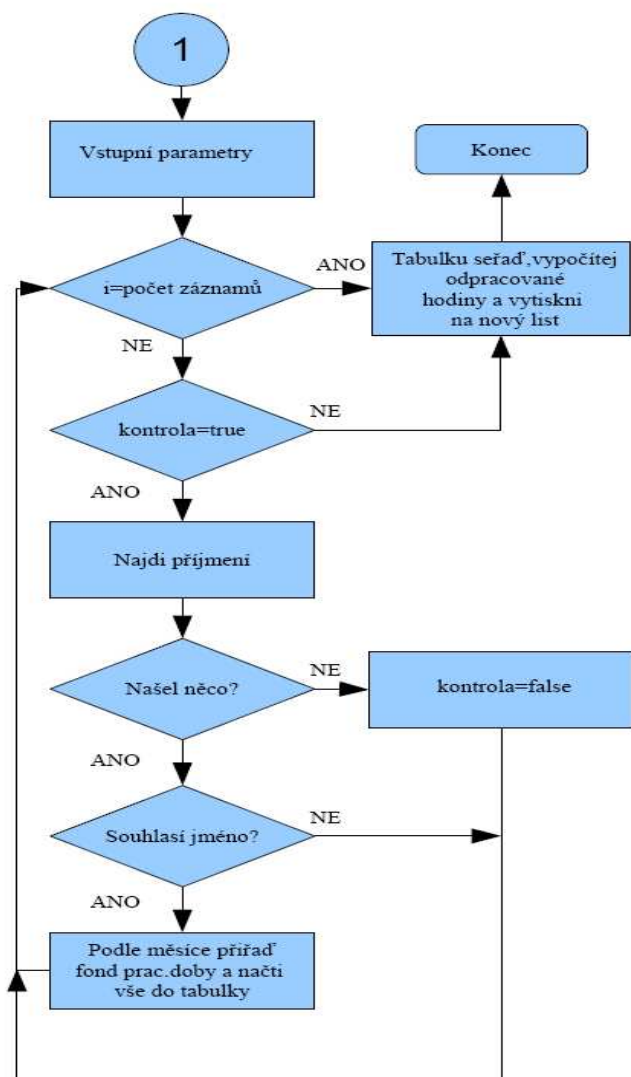


## 4 Vytvoření měsíčního výkazu - docházky

Pro vytvoření docházky se vyplní formulář (obr. č. 6), zkontroluje se, zda jsou zadané všechny položky (obr. č. 15) a přistoupí se ke zpracování vstupních dat.



Obr. č. 15 Docházka 1.část



Obr. č. 16 Docházka 2. část

Po nastavení nezbytných parametrů je vyhledán první záznam s požadovaným příjmením (viz obr. č. 16). Pokud není nalezen žádný takový záznam, proměnná kontrola je nastavena na false, následující procesy jsou přeskočeny a je vytvořen pouze list s hlavičkou, ovšem bez vypsání docházky. Pokud ale ovšem je shodné příjmení nalezeno, kontrola rovná se true, jsou v řádku nalezeného jména načteny proměnné jméno, datum a čas. Jméno je porovnáno s požadovaným a postup se opakuje jako s příjmením. Pokud se shoduje, pokračuje se ve zpracování docházky, pokud ne, je kontrola nastavena na false a v příštím cyklu je proces ukončen. Taktéž je zkontrolováno datum. Výpočet docházky lépe přiblíží následující část zdrojového kódu s komentářem.

- ověření zda nejsou pole prázdná

```

If Dochazka.Pole_prijmeni.Value = Empty Then
MsgBox "Příjmení je povinná položka" & vbCrLf & "Zadejte
znovu.", vbInformation + vbOKOnly, "Docházka: zadání"
Exit Sub
End If
If Dochazka.Pole_jmeno.Value = Empty Then
MsgBox "Jméno je povinná položka" & vbCrLf & "Zadejte znovu.",
vbInformation + vbOKOnly, "Docházka: zadání"
Exit Sub
End If

If Dochazka.Pole_mesic.Value = Empty Then
MsgBox "Měsíc je povinná položka" & vbCrLf & "Zadejte znovu.",
vbInformation + vbOKOnly, "Docházka: zadání"
Exit Sub
End If

```

- proměnné s indexem cíl jsou myšleny jako proměnné vytvářené docházky, kdežto proměnné s indexem zdroj jsou kopírované z listu vygenerovány

```

prijmeni_cil = Dochazka.Pole_prijmeni.Value
jmeno_cil = Dochazka.Pole_jmeno.Value
mesic = Dochazka.Pole_mesic.Value

```

- nastavení vstupních proměnných.

```

radek_zdroj = 2
sloupec = 2
pocet_zaznamu = 1
pocitadlo = 0

```

- proměnná kontrola je typu boolean. Pokud je v nějaké fázi programu nastavena jako false, znamená to, že nastala neshoda mezi nalezeným a zadaným parametrem a cyklus bude ukončen. Proto je na začátku nastaveny jako true.

```

kontrola = True

```

- zjištění počtu záznamů a nastavení rozsahu

```

Do While Not IsEmpty
(Sheets("vygenerovany").Cells(radek_zdroj, sloupec))
pocet_zaznamu = pocet_zaznamu + 1
radek_zdroj = radek_zdroj + 1
Loop
rozsah = "$A$2:$E$" & pocet_zaznamu

```

- prohledává se každý řádek na listě vygenerovaný

```
For i = 1 To pocet_zaznamu
Do While kontrola = True
With Sheets("vygenerovany").Range(rozsah).Cells
```

- hledá další shodné příjmení

```
Set c = . Find(prijmeni_cil, LookIn:=xlValues)
```

- pokud se najde shoda, načtou se informace do proměnné c typu Range

```
If Not c Is Nothing Then
k = k + 1
```

- k je kontrola jestli bylo nalezeno shodné příjmení se zadáním alespoň jednou. Na začátku k = 1.

```
radek = c.Row
sloupec = c.Column
Sheets("vygenerovany").Select
Sheets("vygenerovany").Range("$A$1:$E$" &
pocet_zaznamu).Cells(radek, sloupec).Select
jmeno_zdroj = ActiveCell.Offset(0, -1).Value
radek = radek + 1
rozsah = "$A$" & radek & ":$E$" & pocet_zaznamu
Else:
If k = 1 Then
```

- shoda nastala ani jednou, příjmení se v seznamu nenachází.

```
MsgBox "Takový tady není" & vbCrLf & "Zadejte znovu.",
vbInformation + vbOKOnly, "Docházka: zadání"
kontrola = "false"
Exit For
Else:
```

- shoda nastala alespoň jednou, k <> 1, ale nyní už nebylo nic nalezeno

```
Exit For
End If
End If
End With
```

- počítadlo je ochrana proti zacyklování

```
pocitadlo = pocitadlo + 1
If pocitadlo = > pocet_zaznamu Then
kontrola = False
Exit Do
End If
```

- kontrola, zda souhlasí jméno nalezeného záznamu se zadáním

```
If jmeno_zdroj <> Dochazka.Pole_jmeno.Value Then  
kontrola = False  
End If
```

- načtení data, zapsání čísla měsíce do proměnné kontrolaMesic

```
datum = ActiveCell.Offset(0, 2).Value  
kontrolaMesic = Month(datum)
```

- přiřazení jména měsíce a fondu pracovní doby

```
Select Case kontrolaMesic  
Case 1  
kontrolaMesic = "leden"  
fond = 160  
Case 2  
kontrolaMesic = "únor"  
fond = 160  
Case 3  
kontrolaMesic = "březen"  
fond = 184  
Case 4  
kontrolaMesic = "duben"  
fond = 168  
Case 5  
kontrolaMesic = "květen"  
fond = 168  
Case 6  
kontrolaMesic = "červen"  
fond = 176  
Case 7  
kontrolaMesic = "červenec"  
fond = 160  
Case 8  
kontrolaMesic = "srpen"  
fond = 176  
Case 9  
kontrolaMesic = "září"  
fond = 168  
Case 10  
kontrolaMesic = "říjen"  
fond = 160  
Case 11  
kontrolaMesic = "listopad"  
fond = 168  
Case 12  
kontrolaMesic = "prosinec"  
fond = 176  
End Select
```

- kontrola zda souhlasí měsíc nalezeného záznamu se zadáním

```
If kontrolaMesic = mesic Then
  If kontrola = True Then
```

- souhlasí příjmení, jméno i měsíc se zadáním, jednotlivé položky záznamu jsou zapsány do tabulky

```
ReDim Preserve tabulka(1 To 4, 1 To pocet_zaznamu)
datum = ActiveCell.Offset(0, 2).Value
Cas = ActiveCell.Offset(0, 3).Value
tabulka(1, y) = prijmeni_cil
tabulka(2, y) = jmeno_cil
tabulka(3, y) = datum
tabulka(4, y) = Cas
y = y + 1
End If
End If
Loop
Next i
```

- všechny záznamy, které odpovídaly zadání, jsou zapsány v tabulce
- vytvoření nového listu, který bude pojmenován ve formátu „Příjmení jméno“ předchází smazání případného listu se stejným pojmenováním

```
jmenolistu = prijmeni_cil & " " & jmeno_cil
```

- vypnutí kontroly a porovnání pojmenování všech listů v sešitu

```
Application.DisplayAlerts = False
j = Sheets.Count 'počet listů
For i = 1 To j
  jmenolistu2 = Sheets(i).Name
  If jmenolistu2 = jmenolistu Then
    Sheets(jmenolistu).Delete
  End If
Next i
```

- zapnutí kontroly

```
Application.DisplayAlerts = True
```

- vytvoření nového listu

```
Sheets.Add After:=Sheets(Sheets.Count)
ActiveSheet.Name = jmenolistu
```

- výpočet docházky z tabulky

```
radek_cil = 10
j = 1
```

- y je počet zápisů – řádků v tabulce, viz výše
- hodiny jsou počítány tak, že jsou porovnávány jednotlivé časy a data, tedy jeden jako příchod a druhý jako odchod

```
For i = 1 To y - 1 Step 2
Cas1 = tabulka(4, i)
Cas2 = tabulka(4, i + 1)
```

- Cas1 - příchod, Cas2 - odchod a Cas - odpracováno
- Cas1 < Cas2...odchod před 24:00 hod

```
If Cas1 < Cas2 Then
Cas = Cas2 - Cas1
Else
```

- Cas1 > Cas2...odchod po 24:00 hod

```
Cas = Abs(1 - Cas1) + Cas2
nocni = True
End If
```

- výpočet příplatků noční směna

```
If Cas1 < Cas2 Then
```

- příchod před 22:00 hod (v Excelu číslo 0,9)

```
If Cas1 < 0.9 Then
```

- odchod mezi 22:00 hod a 24:00 hod

```
If Cas2 > 0.9 Then
dopulnoci = Cas2 - 0.9
nocni = True
End If
End If
End If
```

- směna trvala přes 24:00 hod (za podmínky, že nemůže být delší než 24 hod) => příplatek noční směna

```
If Cas1 >= Cas2 Then
```

- = proto kdyby náhodou byly časy stejné

```
nocni = True
```

- příchod před 22:00 hod, ale určitě odchod po 24:00 hod, viz předchozí podmínka

```
If Cas1 < 0.9 Then
```

- příplatek 2 hod, 22:00 hod - 24:00 hod

```
dopulnoci = 2 / 24
End If
```

- příchod po 22:00 hod

```
If Cas1 >= 0.9 Then
dopulnoci = 1 - Cas1
End If
```

- odchod před 6:00 hod

```
If Cas2 <= 0.25 Then
populnoci = Cas2
Else
populnoci = 0.25 'odešel po 6:00
End If
End If
priplatekNocni = (dopulnoci + populnoci) * 24
```

- zápis do nového listu
- datum příchodu

```
Sheets(jmenolistu).Cells(radek_cil, 1).Value = tabulka(3, j)
denVtydnu = Weekday(tabulka(3, j), vbMonday)

Sheets(jmenolistu).Cells(radek_cil, 2).Value = Cas1
j = j + 1
If nocni = False Then
Sheets(jmenolistu).Cells(radek_cil, 3).Value = Cas2
j = j + 1
Else
radek_cil = radek_cil + 1
If IsEmpty(Sheets(jmenolistu).Cells(radek_cil, 1).Value) Then
```

- datum odchodu

```
Sheets(jmenolistu).Cells(radek_cil, 1).Value = tabulka(3, j)
End If

Sheets(jmenolistu).Cells(radek_cil, 3).Value = Cas2
Sheets(jmenolistu).Cells(radek_cil, 7).Value = priplatekNocni
j = j + 1
End If
```

- odpracováno hodin

```
Sheets(jmenolistu).Cells(radek_cil, 4).Value = 24 * Cas
```

- odpracováno přesčas

```
Sheets(jmenolistu).Cells(radek_cil, 5).Value = 24 * (Cas - 0.33)
```

- příplatek sobota (6. den v týdnu) a neděle (7. den v týdnu)

```
Select Case denVtydnu
Case 6
Sheets(jmenolistu).Cells(radek_cil, 6).Value = 24 * Cas
Case 7
Sheets(jmenolistu).Cells(radek_cil, 6).Value = 24 * Cas
End Select
radek_cil = radek_cil + 1
```



```
Next i
Unload Me
```

- seřazení na listu podle data

```
Range("A9").Select
ActiveWorkbook.Worksheets(jmenolistu).Sort.SortFields.Clear
ActiveWorkbook.Worksheets(jmenolistu).Sort.SortFields.Add Key:=
Range("A9"), SortOn:=xlSortOnValues, Order:=xlAscending,
DataOption:= xlSortNormal
With ActiveWorkbook.Worksheets(jmenolistu).Sort
.SetRange Range("$A$10:$G$" & radek_cil - 1)
.Header = xlNo
.MatchCase = False
.Orientation = xlTopToBottom
.SortMethod = xlPinYin
.Apply
End With
```

- sumarizace docházky
- hodin celkem

```
Range("B4").Select
ActiveCell.FormulaR1C1 = "=SUM(R[6]C[2]:R[" & radek_cil &
"C[2])"
```

- příplatek sobota, neděle

```
Range("B6").Select
ActiveCell.FormulaR1C1 = "=SUM(R[4]C[4]:R[" & radek_cil &
"C[4])"
```

- příplatek noční směna

```
Range("B7").Select
ActiveCell.FormulaR1C1 = "=SUM(R[3]C[5]:R[" & radek_cil &
"C[5])"
Selection.Copy
Application.CutCopyMode = False
Range("B4:B7").Select
Selection.NumberFormat = "0.00"
Selection.NumberFormat = "0.0"
```

- příplatek přesčas

```
Range("B5").Select
hodinCelkem = Range("B4").Value
ActiveCell.FormulaR1C1 = hodinCelkem - fond

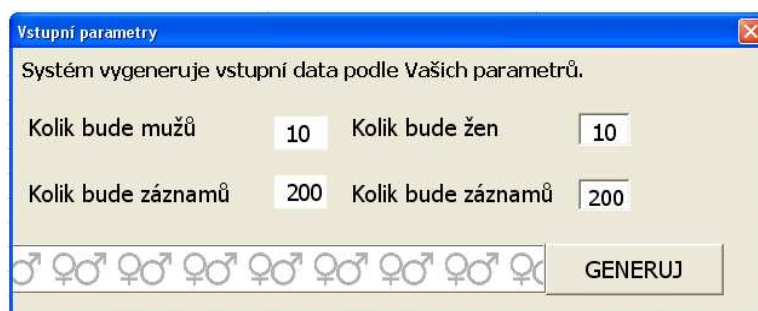
End Sub
```

Tato část programu byla zpracována za použití odborné literatury [6], [7], [8],[12].

## 5 Praktická ukázka

### 5.1 Generování záznamů

Záznamy se vygenerují podle těchto kritérií obr. č. 16



Vstupní parametry

Systém vygeneruje vstupní data podle Vašich parametrů.

Kolik bude mužů  Kolik bude žen

Kolik bude záznamů  Kolik bude záznamů

♂ ♀ ♂ ♀ ♂ ♀ ♂ ♀ ♂ ♀ ♂ ♀ ♂ ♀ ♂ ♀ ♂ ♀ ♂ ♀

GENERUJ

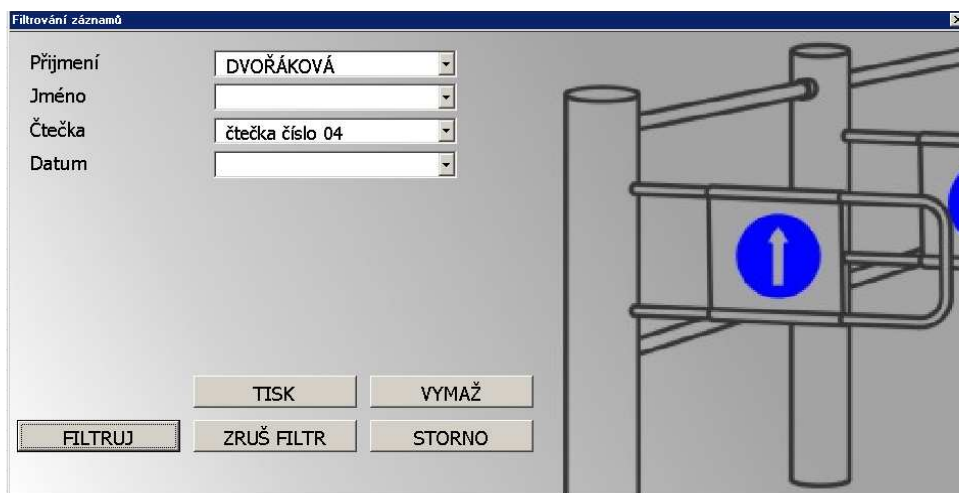
Obr. č. 16 Kritéria generování

- Kolik bude mužů: 10
- Kolik bude záznamů: 200
- Kolik bude žen: 10
- Kolik bude záznamů: 200

Po kliknutí na tlačítko **GENERUJ** se tento proces spustí. Jak bylo popsáno výše, počet záznamů je chápán jako možné maximum. Program vygeneruje pro každé jméno jiný počet záznamů, aby se co nejvíce přiblížil realitě. Znamená to, že pokud by byla zadána stejná kritéria, bude celkový počet záznamů pokaždé jiný. V tomto konkrétním případě bylo vytvořeno 2223 záznamů.

## 5.2 Filtrování, práce se záznamy

Data lze filtrovat podle několika podmínek. Lze filtrovat podle libovolného množství kritérií. To znamená, že pokud bude objektem zájmu pouze pohyb subjektu „Dvořáková“ kontrolovaným vstupem označený jako „čtečka číslo 04“, bude filtr nastaven jako na obr. č. 17.



Obr. č. 17 Filtrování záznamů

Kliknutím na tlačítko „ZRUŠ FILTR“ se zruší předchozí nastavení filtru na listě, jsou zobrazeny všechny záznamy. Zadání ve formuláři filtrování záznamů zůstane nezměněno.

Kliknutím na tlačítko „VYMAŽ“ se zruší pouze aktuální nastavení formuláře filtrování záznamů. Aktivní filtr na listě zůstane nezměněn.

Kliknutím na tlačítko „STORNO“ se zavře formulář filtrování záznamů. Aktivní filtr na listě zůstane nezměněn.

Kliknutím na tlačítko „TISK“ se vytvoří nový list, který má v hlavičce vypsáno kdy a podle čeho se filtrovalo a obsahuje filtrovaná data. Na obr. č. 18 je vidět vytisknutá stránka, kde data byla filtrována podle zadání na obr. č. 17.

Filtrování ze dne 15.12.2010 20:09:49 podle těchto kritérií

Příjmení: DVOŘÁKOVÁ

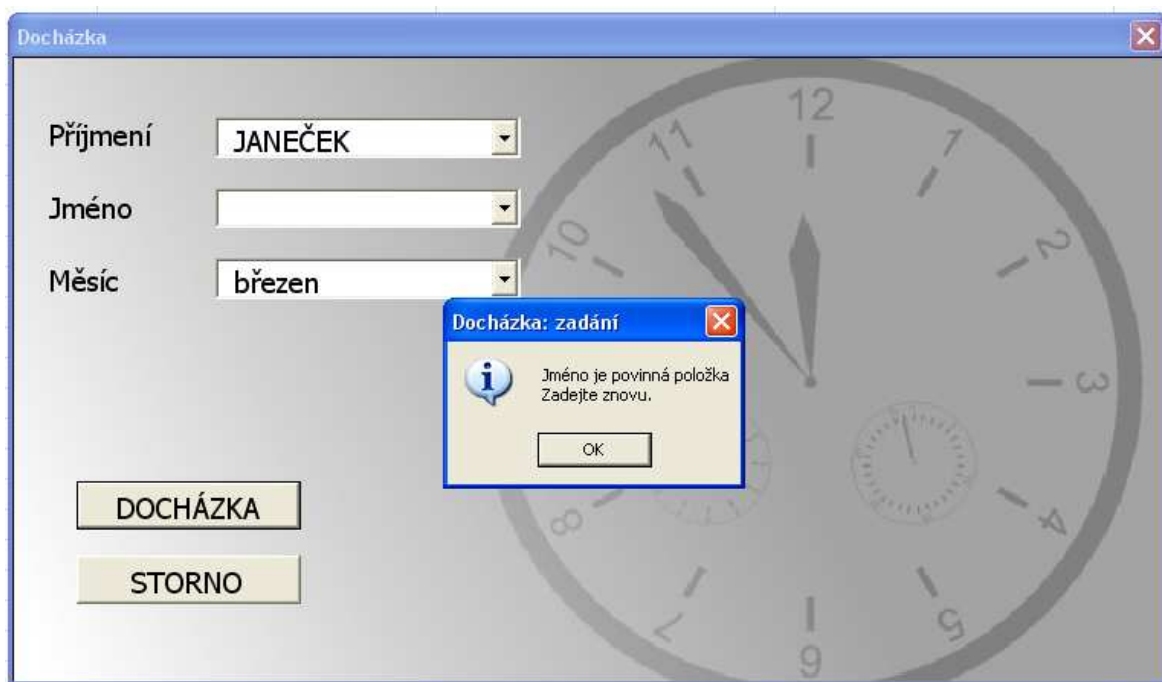
Čtečky: čtečka číslo 04

JMÉNO	PŘÍJMENÍ	ČTEČKA	DATUM	ČAS
Josefa	DVOŘÁKOVÁ	čtečka číslo 04	17.3.2010	00:00:00
Josefa	DVOŘÁKOVÁ	čtečka číslo 04	3.4.2010	00:00:00
Josefa	DVOŘÁKOVÁ	čtečka číslo 04	8.10.2010	07:39:52
Josefa	DVOŘÁKOVÁ	čtečka číslo 04	10.10.2010	10:55:22
Josefa	DVOŘÁKOVÁ	čtečka číslo 04	20.11.2010	11:27:51
Josefa	DVOŘÁKOVÁ	čtečka číslo 04	16.12.2010	22:50:28

Obr. č. 17 Výtisk filtrování

### 5.3 Docházka, podklady pro výpočet mzdy

Na rozdíl od filtrování záznamů, ve formuláři pro docházku jsou všechna pole povinná. Pokud není některé pole vyplněné, program na to upozorní (viz obr. č. 18) a nepokračuje dál, dokud nejsou všechna pole vyplněna.



Obr. č. 18 Docházka-chybné zadání

Jako vstupní data pro vytvoření docházky neboli podkladů pro výpočet mzdy, byla použita vygenerovaná. Proto není docházka do zaměstnání subjektu standardní, ale pro demonstraci chodu programu plně dostačující. Výpočet docházky vychází z reálních hodnot fondu pracovní doby pro rok 2010, které jsou v programu zadány přímo.

Vstupní parametry:

- Příjmení      Janeček
- Jméno        Čeněk
- Měsíc        březen

Na obr. č. 19 je znázorněn list s podklady pro výpočet mzdy dle zadaných parametrů.

JANEČEK	Čeněk
Docházka	březen 2010
Hodin celkem:	51,2
Přesčas:	-108,8
Příplatek So+Ne:	8,6
Příplatek noční:	9,4

Datum	Příchod	Odchod	Hodin	Přesčas	So+Ne	Noční
1.3.2010	19:23:12					
2.3.2010		00:00:00	4,6	-3,3		2,0
4.3.2010	22:15:46					
5.3.2010		00:00:00	1,7	-6,2		1,7
9.3.2010	01:51:52					
9.3.2010		11:42:16	9,8	1,9		
11.3.2010	16:25:03					
12.3.2010		00:00:00	7,6	-0,3		2,0
23.3.2010	14:01:43					
23.3.2010		23:42:03	9,7	1,8		1,7
25.3.2010	14:51:38					
26.3.2010		00:00:00	9,1	1,2		2,0
27.3.2010	11:31:53					
27.3.2010		20:08:51	8,6	0,7	8,6	

Obr. č. 19 Podklady pro výpočet mzdy

## 6 Závěr

Visual Basic pro aplikace je nástroj, pomocí něhož lze významným způsobem zjednodušit ale především zefektivnit práci nejen v prostředí Microsoft Office. Jistě by se dala jmenovat celá řada aplikací, kde VBA zvyšuje užitnou hodnotu software (např. AutoCAD společnosti Autodesk). Tato polemika samozřejmě není předmětem této práce, pouze ukazuje jakým univerzálním nástrojem Visual Basic pro aplikace (VBA) je.

Tato práce se zabývá využitím VBA v bezpečnostních systémech. Program vytvořený v tomto prostředí je velice snadno modifikovatelný potřebám provozovatelů přístupových a docházkových systémů. Jak už bylo zmíněno výše, celá řada bezpečnostních systémů umožňuje exportovat data ve formátu \*.csv (Comma-separated values, hodnoty oddělené čárkami). Existují samozřejmě varianty, které pro oddělování používají i jiné znaky než čárku, nejčastěji středník nebo tabulátor).

Předkládaný program pracuje v prostředí Microsoft Excel. Lze ho rozdělit na tři základní části. V první části je popsáno, jakým způsobem lze získat vstupní data vygenerováním. Data byla generována z hlediska ochrany osobních dat. Jedná se tedy o data, která svým charakterem simulují skutečné výstupy přístupových systémů, přesto o data naprosto náhodná.

Ve druhé části je znázorněno, jakým způsobem lze rychle vyhodnocovat například pohyb konkrétní osoby střeženým prostorem. Výstupem je soubor ve formátu \*.pdf, který podstatným způsobem zefektivňuje práci bezpečnostních složek fyzické ochrany, Policie ČR popřípadě SÚJB (Státní úřad pro jadernou bezpečnost České republiky).

Poslední, třetí část se zabývá vytvořením formuláře například pro mzdovou účtárnu jako podklad pro výpočet mzdy. Z generovaných dat je pro konkrétního pracovníka a konkrétní měsíc vytvořeny sumarizace odpracovaných hodin, včetně příplatků za práci přesčas, příplatků za práci o víkendu i v noci.

Jak bylo zmíněno v úvodu, lze tento program pokládat za alfa verzi, která po nasazení v reálné aplikaci, přizpůsobení potřebám klienta bude zpracovávat výstupní data za přístupového, docházkového systému.

## 7 Seznam použitých zdrojů

- [1] PECHÁČEK, Petr. [Http://officir.ic.cz](http://officir.ic.cz) [online]. 2004/2005 [cit. 2010-11-24]. VBA - vážně to nezvládnete?. Dostupné z WWW: <[http://officir.ic.cz/chipex05/01/excel\\_vba\\_uvod.html](http://officir.ic.cz/chipex05/01/excel_vba_uvod.html)>.
- [2] ČR. Zákon o ochraně osobních údajů. In 101/2000 Sb. 2000,
- [3] Mvcr.cz [online]. 2010 [cit. 2010-12-03]. Ministerstvo vnitra České republiky. Dostupné z WWW: <<http://www.mvcr.cz/clanek/statistiky-cetnost-jmen-a-prijmeni.aspx?q=Y2hudW09Mw%3d%3d>>.
- [4] Křestní jména [online]. 2010 [cit. 2010-12-08]. Seznam a význam jmen. Dostupné z WWW: <<http://krestni-jmena.cz/>>.
- [5] Příjmení u nás [online]. 2010 [cit. 2010-12-08]. Četnost českých příjmení. Dostupné z WWW: <<http://prijmeni.unas.cz/>>.
- [6] DODGE, Mark; STINSON, Craig. Mistrovství v Microsoft Office Excel 2007. Vydání první. Brno: Computer Press, a.s., 2008. 936 s. Dostupné z WWW: <<http://knihy.cpress.cz/>>. ISBN 978-80-251-1980-8, K1487.
- [7] BROŽ, Milan; BEZVODA, Václav. Microsoft Excel: Vzorce, funkce a výpočty. Vydání první. Brno: Computer Press, a.s., 2006. 568 s. Dostupné z WWW: <<http://knihy.cpress.cz/>>. ISBN 80-251-1088-5, K1306.
- [8] WEBER, Monika; BREDEN, Melanie. Excel VBA - Velká kniha řešení Excel VBA - Velká kniha řešení. Brno: Computer Press, 2007. 872 s. ISBN 978-80-251-1453-7.
- [9] WALKENBACH, John. Microsoft Excel 2003 - Programování ve VBA. Brno: Computer Press, 2006. 867 s. ISBN 80-251-0911-9.
- [10] URBÁNEK, Tomáš; ŠKÁRKA, Jaroslav. MS Excel 97 CZ Pro vědce a inženýry. Brno: Computer Press, 2001. 450 s. ISBN 8072260995.
- [11] CORNELL, Gary. Microsoft Visual Basic script. Příručka programátora. Brno: Computer Press, 2001. 294 s. ISBN 8072261444.
- [12] HALVORSON, Michael. Microsoft Visual Basic Professional 6.0 -- Krok za krokem. Brno: Computer Press, 2001. 545 s. ISBN 80-7226-445-1.