

MORAVSKÁ VYSOKÁ ŠKOLA OLOMOUC
ÚSTAV INFORMATIKY A APLIKOVANÉ MATEMATIKY

Jaroslav Vrána

**Návrh databázového systému "Cestovní příkazy"
v Microsoft Accessu**

Design of a Travel Orders System Database in Microsoft Access

Bakalářská práce

Vedoucí práce: Mgr. Zdeňka Krišová

Olomouc 2013

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a použil jen uvedené
informační zdroje.

Olomouc 16. 3. 2013

vlastnoruční podpis

Děkuji Mgr. Zdeňce Krišové za odborné vedení bakalářské práce a za cenné rady při zpracování bakalářské práce.

Obsah

Úvod.....	6
1 Teoretická část.....	8
1.1 Agenda cestovní příkazy.....	8
1.2 Microsoft Access.....	9
1.2.1 Edice MS Office.....	10
1.2.2 Popis MS Access.....	10
1.2.3 Tabulky.....	12
1.2.4 Dotazy.....	13
1.2.5 Formuláře.....	14
1.2.6 Sestavy.....	14
1.2.7 Makra.....	14
1.2.8 Moduly a VBA kód.....	15
2 Praktická část.....	21
2.1 Případy užití.....	21
2.2 Popis řešení.....	25
2.2.1 Back-end databáze.....	25
2.2.2 Front-end databáze.....	29
2.2.3 Pás karet.....	29
2.2.4 Propojení tabulek.....	31
2.2.5 Způsoby práce s daty.....	32
2.2.6 Akční dotazy v run-time modulu.....	36
2.2.7 Postup vytváření cestovního příkazu.....	37
2.2.8 Tiskové výstupy.....	40
Závěr.....	44
Anotace.....	46

Seznam literatury.....	47
Seznam obrázků	48

Úvod

Cílem mé práce je navrhnout jednoduchý databázový systém v prostředí Microsoft Office, konkrétně v Microsoft Access a vytvořit v praxi využitelný softwarový projekt, který umožní efektivní pořízení cestovních příkazů. Motivací mi byla vlastní patnáctiletá zkušenost při ručním vyplňování těchto dokladů. V dnešní době je mnoho profesí, kde pracovníci jezdí denně pro zaměstnavatele vlastním vozidlem. Poté musí tyto cesty administrativně završit vyplněním příslušného formuláře. Je to činnost nezáživná, neustále se opisuje mnoho údajů a propočítávají náhrady. Právě tuto část by měl můj program zefektivnit. Prostedí Microsoft Access je velmi podceňovanou součástí kancelářského balíku Microsoft Office, a proto jsem se snažil popsat možnosti, které může programátor pro programy obdobného rozsahu využít.

V bakalářské práci jsem použil dvě metody – analýzu a syntézu. Obě tyto metody jsem použil ve více krocích, podle toho, jakou hloubku problému jsem řešil.

Analýza je důležitá před počáteční fází návrhu databáze. Bylo nutno provést dekompozici problému, tedy postupný rozbor na dílčí až atomické části – elementy. Nejdříve šlo o analýzu činností zpracovávaných běžným, tedy papírovým způsobem. Tím jsem získal základní rozdělení – zápis cestovních příkazů, zápis množství kilometrů určených k fakturaci zákazníkům, tisk cestovních příkazů, podkladů k fakturaci a podkladů k výplatě cestovného. Druhý krok již vedl k základní datové struktuře, tzn. seznamu tabulek. Třetí fáze analýzy poskytla konkrétní strukturu tabulek včetně datových typů.

Syntéza je druhá důležitá metoda v procesu vývoje software. Při zkoumání vztahů mezi jednotlivými entitami databáze jsem nastavil relace mezi tabulkami a ty v dalších krocích vedly k definování integritních omezení včetně referenční integrity pomocí primárních a cizích klíčů. Zanedbání syntézy vede v případě vytváření databázových objektů k chybně normalizované databázi, tedy k redundanci dat, jejímž následkem dochází v praxi často k nekonzistenci dat. Mým záměrem bylo navrhnout databázi, jejíž tabulky budou ve třetí normální formě. Návrh SQL dotazů, tvorba formulářů a sestav už byl finálním krokem syntézy řešeného problému.

Využití analýzy a syntézy tedy umožnilo konverzi části papírově orientovaného zpracování agendy cestovních příkazů do počítačové podoby.

1 Teoretická část

Teoretická část se zabývá přiblížením agendy cestovních příkazů a popisem vývojového prostředí programu Microsoft Access.

1.1 Agenda cestovní příkazy

Cestovní příkazy patří mezi základní formuláře denní podnikové administrativy. Jedná se o prvotní účetní doklad, který musí splňovat náležitosti podle zákona č. 563/1991 Sb. o účetnictví, ve znění pozdějších předpisů. Oblast cestovních náhrad je popsána v příslušných ustanoveních zákona č. 262/2006 Sb., zákoníku práce (dále jen ZP), ve znění pozdějších předpisů. V §42, odstavec 1 ZP je definována pracovní cesta jako časově omezené vyslání zaměstnance zaměstnavatelem k výkonu práce mimo sjednané místo výkonu práce. Z toho nepřímo vyplývá několik dalších nutných náležitostí, které musí Cestovní příkaz obsahovat. Jde především o tyto údaje:

- identifikaci pracovníka vykonávajícího práci (příjmení, jméno, titul, bydliště, telefon, osobní číslo, útvar, pracovní doba od-do),
- datum a podpis pracovníka oprávněného služební cestu povolit (nařídil) a pracovníka, který služební cestu vykonává,
- identifikaci pracovní cesty (počátek cesty a konec pracovní cesty včetně místa, data a času, místo výkonu práce a účel pracovní cesty),
- seznam spolucestujících,
- identifikaci použitého dopravního prostředku (druh dopravního prostředku, SPZ, spotřeba dle Technického průkazu, druh a cena pohonných hmot,
- zálohu (předpokládaná částka výdajů, částka, datum a číslo pokladního dokladu povolené zálohy),
- zprávu o pracovní cestě (datum podání a datum souhlasu a podpis oprávněného pracovníka),
- výdajový pokladní doklad (číslo, úprava částky, odpočet zálohy, účtovací předpis, datum a podpis příslušných pracovníků),
- detailní rozpis pracovní cesty (datum, čas a místo odjezdu a příjezdu, druh použitého dopravního prostředku, vzdálenost, začátek a konec výkonu práce,

1.2.1 Edice MS Office

Microsoft Office 2010 existuje v několika edicích, které se od sebe odlišují způsobem licencování a vybaveností. Microsoft Access je dostupný od edice Professional nebo jej lze zakoupit samostatně.

Přehled sad Office 2010

Aplikace	Balené sady			Multilicenční sady		
	Pro studenty a domácnosti	Pro podnikatele	Professional	Professional Academic	Standard	Professional Plus
Word	Ano	Ano	Ano	Ano	Ano	Ano
Excel	Ano	Ano	Ano	Ano	Ano	Ano
PowerPoint	Ano	Ano	Ano	Ano	Ano	Ano
OneNote	Ano	Ano	Ano	Ano	Ano	Ano
Outlook**		Ano	Ano	Ano	Ano	Ano
Publisher			Ano	Ano	Ano	Ano
Access			Ano	Ano		Ano
InfoPath						Ano
Communicator						Ano
SharePoint Workspace						Ano
Technická podpora	90 dní*	90 dní*	1 rok*	90 dní*	Poskytuje prodejce	

* Platí od data aktivace produktu.

** Business Contact Manager obsažen jen v multilicenčních sadách.

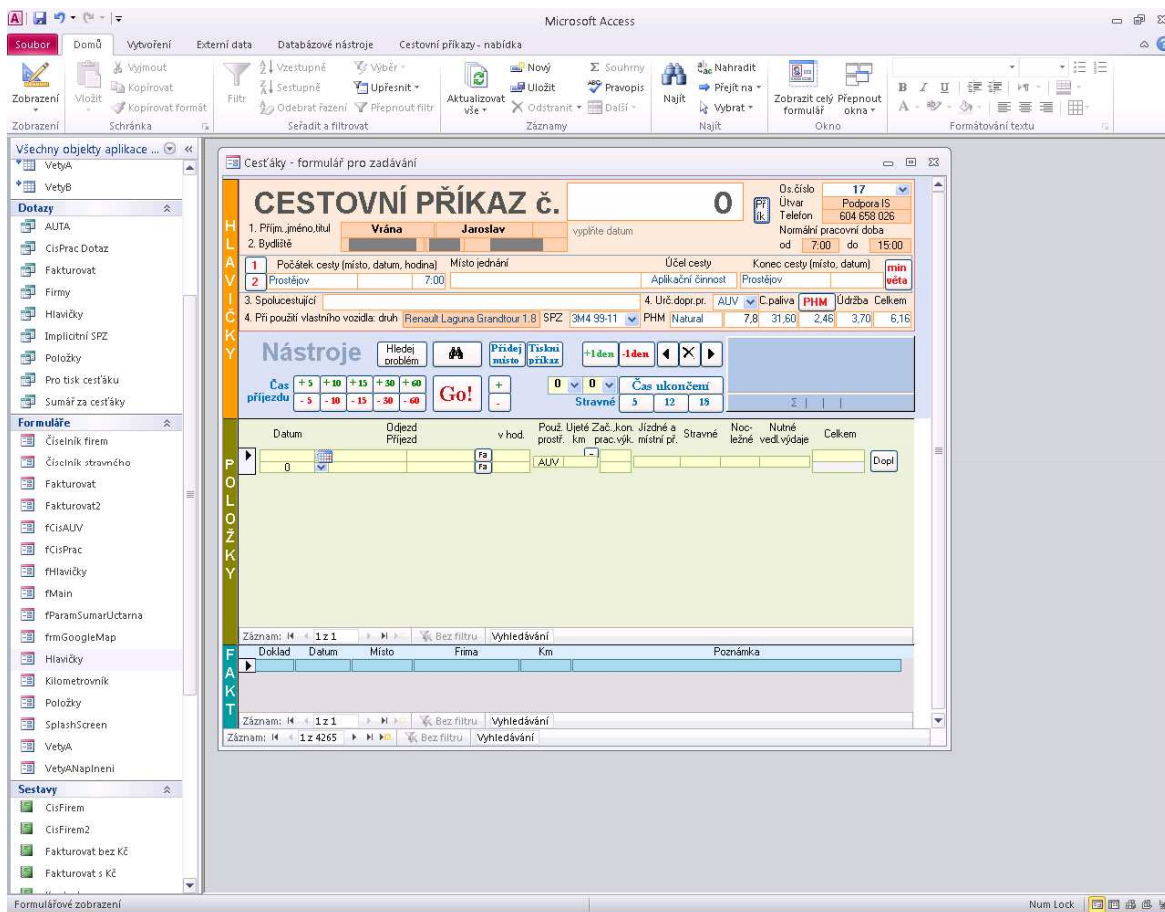
- Sada Office Professional Academic 2010 je určena jen pro vzdělávací instituce, jejich učitele, studenty a zaměstnance a dále pro veřejné knihovny, veřejná muzea a charitativní organizace.
- Sada Office 2010 pro studenty a domácnosti není licencována pro využití při komerčních, neziskových ani ziskových obchodních činnostech ani v organizacích veřejné správy.

Obr. 2 – Přehled sad Office 2010, zdroj ¹

1.2.2 Popis MS Access

Microsoft Access je integrované vývojové prostředí. Obsahuje vlastní relační databázový systém a nástroje pro vytváření tabulek, dotazů, formulářů, sestav, maker a zdrojových kódů ve Visual Basicu for Application.

¹ <http://www.svetsoftware.cz/microsoft-office-2010-pro-podnikatele/porovnani-edici>



Obr. 3 – Uživatelské prostředí, zdroj vlastní obrázek

Uživatelské rozhraní se skládá z:

- pásu nástrojů pro rychlý přístup (nahore vedle tlačítka **Office**),
- pásu karet nahore,
- navigačního podokna (vlevo),
- podokno pro otvírané objekty aplikace MS Access.

Na této obrazovce se odehrává hlavní část návrhu programu. Obvykle budeme pracovat s daty uloženými v tabulkách, takže je potřeba tyto tabulky založit nebo je vytvořit propojením na jiné zdroje. S daty lze pracovat přímo v tabulkách nebo si můžeme vytvořit editační formuláře. Pro prezentaci dat vytvoříme sestavy a grafy. Mnoho operací s daty i s grafickým návrhem prostředí lze provádět pomocí maker a především ve VBA kódu. Blíže jsou tyto části návrhu aplikace popsány dále.

1.2.3 Tabulky

Pro uložení dat je tedy možné využít zmíněný vlastní databázový systém Microsoft Jet Database Engine (MS JDBE). Mimo to lze importovat nebo propojit tabulky i z jiných databázových systémů a prostředí prostřednictvím ODBC ovladačů. Takto lze pracovat s tabulkami z Oracle, 4D, dBase, Paradox, MySQL, MS Excel či třeba v textovém tvaru. Další možnosti jak vytvořit tabulku je vyplněním prázdné tabulky hodnotami. MS Access se v tomto případě pokusí určit datové typy. Ne vždy je však vzorek dat dostatečně reprezentativní. Nativní databáze je pro menší projekty v mnoha případech naprosto dostačující. Obsahuje standardní datové typy, může být normalizovaná podle běžných kritérií a podporuje SQL jazyk. Pro větší projekty je však vhodnější umístění dat v databázi na SQL serveru. Přírozenou volbou by mohla být třeba Express edice MS SQL (dříve MSDE).

Datové typy v tabulkách:

- Text obsahuje alfanumerické znaky a může dosahovat velikosti maximálně 2^8 , tedy 255 bajtů.
- Memo použijeme v případě, že nestačí datový typ Text a může nabývat velikosti maximálně 2^{16} , tedy 65536 bajtů.
- Číslo má velikost 1, 2, 4 nebo 8 bajtů. Je-li velikost pole nastavena na replikační identifikátor, zabírá 16 bajtů.
- Automatické číslo je náhodné nebo sekvenční unikátní číslo zvyšované o 1, má velikost 4 bajty nebo 16, je-li velikost pole nastavena na replikační identifikátor.
- Datum a čas zabírá 8 bajtů.
- Ano/ne slouží pro vyjádření logické hodnoty pravda/nepravda a má velikost 1 bit.
- Měna má velikost 8 bajtů. Používá se k ukládání peněžních hodnot.
- Hypertextový odkaz se může skládat ze čtyř částí, každá z nich může být maximálně 2048 znaků dlouhá. Jedná se o:
 - o text to display - text k zobrazení v poli,
 - o address - adresu obsahující UNC cestu nebo URL,
 - o subaddress – umístění v souboru nebo na stránce,
 - o screentip – zobrazí se jako popisek.
- Objekt OLE o velikosti maximálně 1 gigabajt slouží k propojení jiných objektů

přes OLE/DDE protokol. Obvykle se jedná o dokumenty MS Word, MS Excel, obrázky, apod. vytvořené v jiných programech.

- Příloha slouží k připojení jakéhokoliv podporovaného typu souboru obdobně jako je tomu u e-mailových zpráv.
- Počítané² - obsahuje výraz, je to tedy vypočítaný údaj. Tento datový typ není uložen a vyhodnocuje se ve chvíli, kdy je požadovaná jeho hodnota.
- Průvodce vyhledáváním umožňuje zvolit hodnotu výběrem. Zdrojem pro tento výběr může být jiná tabulka, dotaz nebo výčet hodnot.

1.2.4 Dotazy

Query (dotazy) mají syntaxi založenou na jazyku SQL. Umožňují DDL i DML skripty, tedy definici dat i manipulaci s daty. Díky této skutečnosti je MS Access snadno použitelný pro programátory i zkušenější uživatele, kteří jazyk SQL znají. Vlastní SQL skript je možné přímo zapsat buďto ručně nebo se generuje automaticky, pokud dotaz tvoříme v návrhovém režimu. Nejčastějším využitím je selekce a projekce dat, tedy výběr podmnožiny řádků a sloupců pro formuláře a sestavy a to včetně zahrnutí údajů z jiných tabulek přes cizí klíče. Lze jej také použít pro odstranění, přidání a modifikaci záznamů. Jazyk bohužel nemá některé součásti v jazyku SQL běžné, jako třeba pohledy, uložené procedury a triggery, takže je nutné na tyto skutečnosti brát ohled a chybějící funkcionalitu nahradit ve VBA kódu. MS Access umožňuje parametrické dotazy, kdy je možno předat hodnoty parametrů nebo se na ně dotázat formou dialogových oken.³ Toto umožňuje znovupoužitelnost skriptů.

Dotazy v MS Accessu jsou:

- výběrové
 - o z jedné tabulky,
 - o z více tabulek,
 - o souhrnné,
 - o křížové,
 - o parametrické,
- akční

² PECINOVSKÝ, J., *Excel a Access 2010: efektivní zpracování dat na počítači.*, s. 129.

³ KUBÁLEK, T., a KUBÁLKOVÁ M., *Databázový systém Microsoft Office Access verze 2007 CZ*, s. 86.

- vytvářecí,
- aktualizací,
- odstraňovací,
- přidávací.

Výsledek dotazu je běžně používán k naplnění hodnot v tabulce nebo je vstupem dalšího dotazu. S ohledem na potřebu normalizace databází se bez dotazů prakticky neobejdeme ve formuláři, sestavě, kontingenční tabulce i kontingenčním grafu.

1.2.5 Formuláře

Formuláře jsou objekty, které obsahují grafické prvky, editační pole a ovládací prvky. Umožňují řízení běhu programu pomocí událostí. Pomocí těchto nástrojů je možné vytvořit graficky přehledné i působivé obrazovky, ve kterých se uživatel snadno zorientuje. Nejčastěji jsou využity pro zobrazení nebo editaci dat z tabulek. Vybrat si můžeme různé editační a ovládací prvky podle charakteru dat. Pro čísla a text se využívá TextBox (textové pole), pro pravdivostní hodnoty CheckBox (zaškrtačací políčko), pro výběr z více hodnot RadioButton (přepínač), ListBox (seznam) nebo ComboBox (rozevírací seznam). Pro zobrazení master-slave formuláře (hlavička a položky) je vhodné použít prvek Podformulář-podsestava. Většinu těchto prvků jsem v návrhu programu využil s cílem co nejvíce ho zpřehlednit a usnadnit zadání dat. Obvykle je zdrojem záznamů formuláře SQL dotaz nebo přímo tabulka. Mimo to lze formuláře využít i bez přímé vazby na data, například na tvorbu menu, splash screen oken, zadání parametrů požadovaných programem od uživatele, apod.

1.2.6 Sestavy

Sestavy slouží k přehledné prezentaci dat. Lze do nich vložit prostá data z tabulek či dotazů nebo také různé typy grafů. Tvorba sestav v MS Accessu je, jako téměř vše, velmi jednoduchá a přitom umožňuje vytvořit profesionální výstup. Můžeme využít skupinování, řazení, součtování dat, řízení jejich zobrazení pomocí VBA skriptů, omezující podmínky pomocí filtrů nebo dotazů.

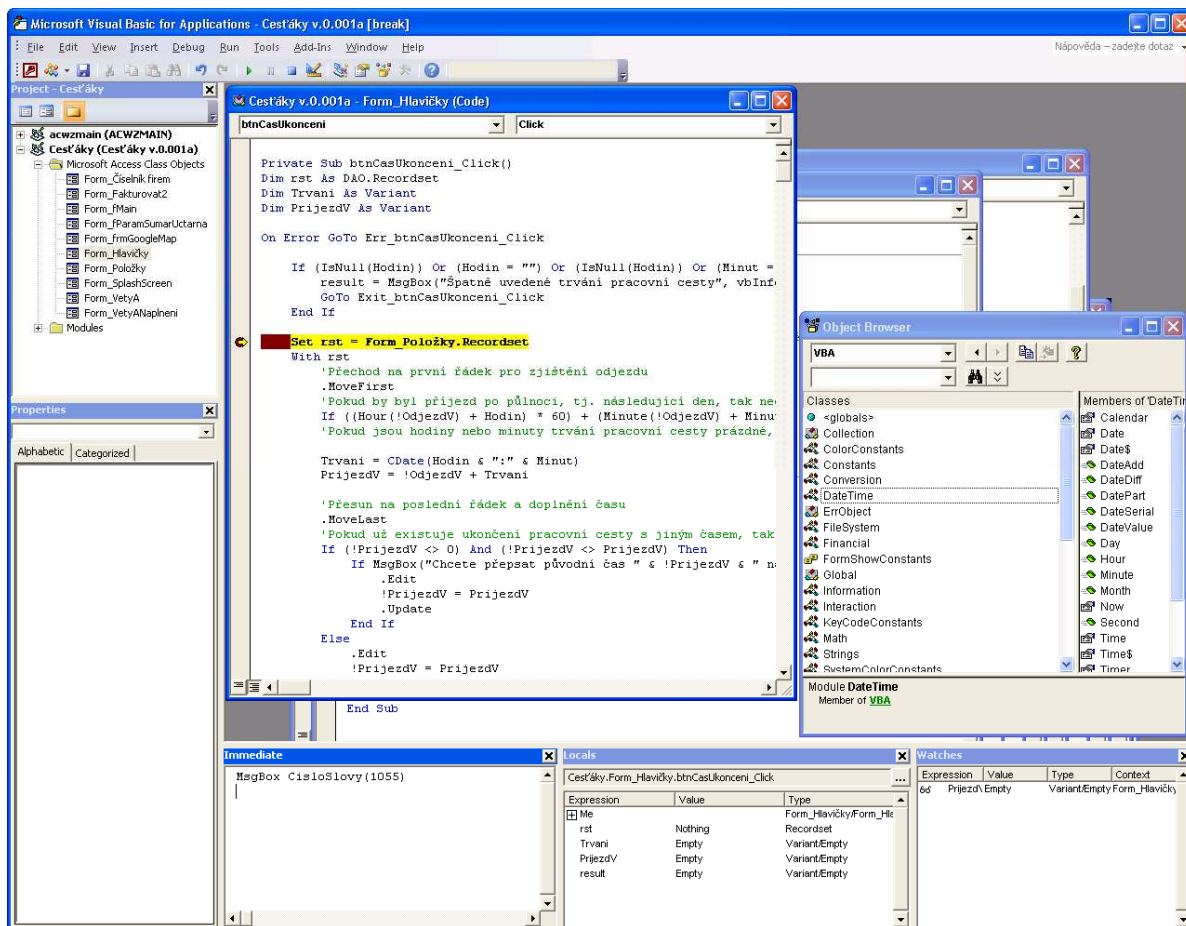
1.2.7 Makra

Makra jsou v podání MS Accessu mocným nástrojem k provádění různých činností. V mnoha případech mohou nahradit zdrojový VBA kód. Lze jimi realizovat vyvolání

formulářů, sestav či dotazů. Využitelné jsou při tvorbě vlastního menu a kontextových nabídek. Ve spojení s tlačítky na formuláři poslouží k navigaci mezi záznamy nebo třeba doplnění hodnot. Samozřejmě je často lepší využít VBA kód, protože v něm má programátor běh programu více pod kontrolou.

1.2.8 Moduly a VBA kód

Vlastní programové části je možné zapisovat, podobně jako v dalších programech kancelářského balíku Microsoft Office, ve VBA kódu (Visual Basic for Application). Tento jazyk je pro běžné účely dostatečně silný nástroj.



Obr. 4 – Prostředí pro VBA, zdroj vlastní obrázek

Rozhraní editoru pro VBA sestává z horizontální nabídky, nástrojové lišty, okna projektu (Project Explorer), okna vlastností (Properties), okna pro sledování hodnot při ladění (Watches a Locals) a okna pro editaci modulů. Vyvolat lze i prohlížeč objektů (Object Browser) či okno pro okamžité vykonání vloženého příkazu (Immediate).

Program je organizován do modulů a ty dále do procedur a funkcí. Rozsah platnosti je Public, Private nebo Static. Pokud rozsah není určen explicitně je Public. Procedurám a funkcím mohou být předávány parametry pomocí klíčových slov Optional jako volitelný nepovinný parametr, ByVal – volání hodnotou, ByVal – volání odkazem anebo ParamArray – pole hodnot typu variant.

Při programování se málokdy obejdeme bez proměnných. Tyto slouží k uchování hodnoty, jsou pojmenované a zabírají určitou část paměti. Někdy hodnotu neuchovává přímo proměnná, ale jejím obsahem je adresa (ukazatel) na místo v paměti s danou hodnotou. Důležitou vlastností proměnné je datový typ. Ten určuje charakter uchovávané hodnoty a prostředí obvykle kontroluje přípustnost vkládaných dat. Ne vždy je však tato kontrola naprosto spolehlivá, takže často může dojít k přetečení rozsahu proměnné. Rozsah platnosti proměnné určíme pomocí klíčových slov Dim, Private, Public nebo Static.

Důležitou vlastností programovacích jazyků jsou příkazy větvení a cyklu. VBA nám dává k dispozici konstrukce pro větvení if-then-else na základě predikátu a select-case podle hodnoty testovaného výrazu.

Uvedený příklad řeší stejný problém různými způsoby:

```
If cislo < 0 Then
    MsgBox "záporné"
ElseIf Cislo = 0 Then
    MsgBox "nula"
Else
    MsgBox "kladné"
End If

Select Case cislo
    Case Is < 0: MsgBox "záporné"
    Case 0: MsgBox "nula"
    Case Else: MsgBox "kladné"
End Select
```


Pro cykly můžeme použít struktury for-to-next, for-each-next, do-while-loop, do-next-loop, do-loop-next, do-loop-while, while-wend⁴.

Příklad stejné funkcionality:

```
Private Sub btnCykly_Click()  
    Dim DnyTydne, MujText  
    DnyTydne = Array("po", "út", "st", "čt", "pá", "so", "ne")  
  
    MujText = "Zkratky dnů v týdnu: " & vbCrLf  
    For i = 0 To 6  
        MujText = MujText & DnyTydne(i) & vbCrLf  
    Next i  
    MsgBox MujText  
  
    MujText = "Zkratky dnů v týdnu: " & vbCrLf  
    For Each den In DnyTydne  
        MujText = MujText & den & vbCrLf  
    Next den  
    MsgBox MujText  
  
    i = 0  
    MujText = "Zkratky dnů v týdnu: " & vbCrLf  
    Do Until i = 7  
        MujText = MujText & DnyTydne(i) & vbCrLf  
        i = i + 1  
    Loop  
    MsgBox MujText  
  
    i = 0  
    MujText = "Zkratky dnů v týdnu: " & vbCrLf  
    Do Until i = 7  
        MujText = MujText & DnyTydne(i) & vbCrLf
```

⁴ KOROL, Julitta. *Access 2007 programming by example with VBA, XML, and ASP.*, s 112-123.

```

        i = i + 1
Loop
MsgBox MujText

i = 0
MujText = "Zkratky dnů v týdnu: " & vbCrLf
Do
    MujText = MujText & DnyTydne(i) & vbCrLf
    i = i + 1
Loop While i < 7
MsgBox MujText

i = 0
MujText = "Zkratky dnů v týdnu: " & vbCrLf
While i < 7
    MujText = MujText & DnyTydne(i) & vbCrLf
    i = i + 1
Wend
MsgBox MujText
End Sub

```

VBA obsahuje dispozici dostatek datových typů:

- Byte obsahuje číslo v rozsahu 0 až 255 a zabírá 2 bajty.
- Boolean obsahuje logickou hodnotu pravda/nepravda a zabírá 2 bajty.
- Integer s rozsahem -32768 to 32767 zabírá 2 bajty.
- Long obsahuje celá čísla mezi -2,147,483,648 a 2,147,483,647 a má velikost 4 bajty.
- LongLong je platný datový typ na platformě x64 zabírá 8 bajtů a nabývá hodnot v intervalu -9 223 372 036 854 775 808 to 9 223 372 036 854 775 807.
- Single -3,402823E38 až -1,401298E-45 pro záporné hodnoty; 1,401298E-45 do 3,402823E38 pro kladné hodnoty.
- Double -1,79769313486231E308 až -4,94065645841247E-324 pro záporné

hodnoty; 4.94065645841247E-324 do 1.79769313486232E308 pro kladné hodnoty.

- Currency pro uložení měny -922 337 203 685 477,5808 až 922,337,203,685,477.5807 a má velikost 8 bajtů.
- Decimal obsahuje čísla v rozsahu +/-79 228 162 514 264 337 593 543 950 335 bez desetinné čárky +/-7.9228162514264337593543950335 s 28 místy vpravo od desetinné čárky; nejmenší nenulové číslo je +/-0.00000000000000000000000001 a má velikost 14 bajtů.
- Date obsahuje datum od 1. ledna 100 do 31. prosince 9999 a má velikost 8 bajtů.
- Object
- String proměnlivé délky slouží k uchování alfanumerických znaků a má velikost až do přibližně 2 miliard znaků. Zabírá velikost 10 bajtů + délku řetězce.
- String pevné délky může být dlouhý až 65536 znaků.
- Variant může obsahovat mimo stringu pevné délky jakýkoliv datový typ. Umožňuje uchovávat i speciální hodnoty jako je Empty, Error, Nothing a Null.
- Uživatelsky definovaný datový typ může obsahovat jeden nebo více prvků standardních datových typů, pole nebo dříve definované uživatelské typy. Tento datový typ slouží obvykle k definici datových víceprvkových struktur, např.

```
Type Osoba
    Jmeno as String
    Prijmeni as String
    Titul as String
    DatNar as Date
End Type
```

Proměnné vytváříme proto, abychom za běhu programu mohli jejich hodnotu měnit. Často však potřebujeme mít zapamatovány hodnoty, které měnit nechceme, přesto je opakovaně využíváme. Pro tento účel VBA obsahuje možnost deklarovat konstanty. Ty mohou nabývat rozsah platnosti Public nebo Private.

Vlastní volání procedur a funkcí se provádí buďto z objektů MS Access, tedy například kliknutí na tlačítko, pomocí časovače nebo voláním z jiné procedury či funkce. Nejčastější je právě propojení VBA kódu na prvek formuláře přiřazením k některé

události tohoto prvku. Běžně je také vyvolávána procedura při otevření nebo zobrazení formuláře.

Pro odladění správné funkčnosti programu jsou k dispozici nástroje jako zarážky (Breakpoints) a ladění s trasováním (Debug). Přitom můžeme sledovat hodnoty proměnných a objektů v okně Locals a Watches. Ošetření chybových stavů můžeme provést pomocí konstrukcí On-Error-Resume-Next a On-Error-GoTo.

2 Praktická část

V praktické části se věnuji tvorbě programu. Na základě analýzy požadavků jsem navrhl datové struktury, formuláře a tiskové výstupy. Makra a VBA kód jsem vytvářel s ohledem na zpracované případy užití.

2.1 Případy užití

Budoucí funkcionalitu systému je vhodné popsat předem formou případů užití⁵ pro jednotlivé aktéry. Aktér v tomto programu vyjadřuje roli uživatele⁶. Nepodcenění této fáze výrazně zjednodušuje, zpřesňuje a zrychluje následný vývoj programu. V dalším textu uvádím klíčové případy užití.

Název případu užití	Vytvoření příkazu
Identifikace příkazu užití	UC001
Aktéři	Pracovník vyúčtovávající svou pracovní cestu
Omezení na stav systému před spuštěním případu užití	Pracovní cesta již byla dokončena.
Kroky případu užití	<ol style="list-style-type: none">1. Uživatel spustí volbu Přidat cestovní příkaz.2. Systém zobrazí formulář cestovního výkazu a zpřístupní údaje pro editaci.3. Uživatel vloží číslo své pracovníka.4. Systém doplní implicitní údaje.<ul style="list-style-type: none">- o pracovníkovi (útvár, telefon, pracovní dobu, jméno, bydliště),- o pracovní cestě (výchozí a koncové místo, datum počátku o 1 den vyšší než předchozí nejvyšší a konce, čas počátku, účel cesty),- o dopravním prostředku (typ určeného dopravního prostředku, druh vozidla, SPZ),

⁵ KANISOVÁ, H., a MÜLLER, M., *UML srozumitelně*, s. 37.

⁶ ARLOW, J., a NEUSTADT, I., *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*, s. 93.

- náhradu za vozidlo (druh PHM, spotřebu, cenu paliva, sazbu za 1 km za údržbu, PHM a jejich součet).
- 5. Uživatel upraví výchozí hodnoty v hlavičce, úpravu času odjezdu může řešit pomocí alternativního postupu v nástrojích pro zefektivnění pořízení (Use Case UC002).
- 6. Uživatel vyplní seznam míst oddělených čárkou bez mezer v údaji Místo jednání ručně nebo pomocí tlačítka **Přidej místo** výběrem z číselníku firem (Use Case UC003).
- 7. Uživatel v nástrojích pro zefektivnění pořízení nastaví dobu trvání pracovní cesty (hodiny, minuty).
- 8. Uživatel vyplní jednotlivé řádky pracovní cesty nebo použije tlačítko **Go!** pro generování podle míst uvedených v hlavičce v údaji Místo jednání (Use Case UC004).
- 9. Doplnění případně ručně nedoplněných údajů do posledního řádku pomocí nástrojů (čas ukončení, výše stravného).
- 10. Doplnění zákazníka do podkladu pro fakturaci cestovného. Lze vybírat tlačítkem vedle místa příjezdu nebo odjezdu v řádcích výkazu.

Omezení na stav systému po ukončení případu užití Cestovní příkaz je založen v systému.

Název případu užití	Úprava času odjezdu
Identifikace příkazu užití	UC002
Aktéři	Pracovník vyúčtovávající svou pracovní cestu
Omezení na stav systému	Čas odjezdu není podle skutečnosti a uživatel chce využít nástroje pro před spuštěním případu užití zefektivnění pořízení.

Kroky případu užití	5a1. Uživatel zvyšuje čas v minutách klikem myši na tlačítka +5 (10,15,30,60) nebo snižuje klikem myši na tlačítka -5 (10,15,30,60) . 5a2. Systém zvýší/sníží čas odjezdu o zadaný počet minut.
Omezení na stav systému po ukončení případu užití	Čas odjezdu je nastaven podle skutečnosti.

Název případu užití	Výběr z existujících firem
Identifikace příkazu užití	UC003
Aktéři	Pracovník vyúčtovávající svou pracovní cestu
Omezení na stav systému před spuštěním případu užití	Místo jednání není vyplněno nebo není úplné.
Kroky případu užití	6a1. Uživatel stiskne tlačítko Přidej místo . 6a2. Systém otevře číselník firem. 6a3. Uživatel dvojklikem vybere zákazníka. 6a4. Systém přenesse alias zákazníka do údaje Místo jednání. Jednotlivá místa jsou oddělena čárkou bez mezer. Takto zadaná místa budou sloužit k automatickému generování řádků výkazu.
Omezení na stav systému po ukončení případu užití	Místo jednání je úplné.

Název případu užití	Tisk jednoho cestovního příkazu
Identifikace příkazu užití	UC004
Aktéři	Pracovník vyúčtovávající svou pracovní cestu

Omezení na stav systému před spuštěním případu užití	Požadovaný cestovní příkaz je vyplněn úplně.
Kroky případu užití	<ol style="list-style-type: none"> 1. Uživatel spustí volbu Přidat cestovní příkaz. 2. Systém zobrazí formulář cestovního výkazu. 3. Uživatel najde konkrétní cestovní příkaz listováním nebo pomocí tlačítka pro hledání nebo vestavěných nástrojů prostředí. 4. Systém zobrazí hledaný příkaz. 5. Uživatel stiskne tlačítko Tiskni příkaz. 6. Systém zobrazí formulář pro tisk v módu Náhled.
Omezení na stav systému po ukončení případu užití	Systém je připraven k tisku cestovního příkazu na zvolenou tiskárnu.

Název případu užití	Tisk intervalu cestovních příkazů
Identifikace příkazu užití	UC005
Aktéři	Pracovník vyúčtovávající svou pracovní cestu
Omezení na stav systému před spuštěním případu užití	Požadovaný cestovní příkaz je vyplněn úplně.
Kroky případu užití	<ol style="list-style-type: none"> 1. Uživatel zvolí tisk příkazů. 2. Systém zobrazí dialog pro zadání čísla pracovníka a intervalu čísel dokladů a datumů. 3. Uživatel vyplní (vybere) své osobní číslo a intervaly. Pokud nebude vyplněn některý z intervalů, nebude výběr omezovat. Všechny podmínky se aplikují současně, jedná se tedy o logický součin a stiskne tlačítko Tisk. 4. Systém přehodí případně chybně zadané intervaly (od je vyšší než do) a zobrazí požadovaný interval příkazů v módu Náhled.

Omezení na stav systému System je připraven k tisku cestovních příkazů na
po ukončení případu užití zvolenou tiskárnu.

2.2 Popis řešení

Pro svoji práci jsem si zvolil nativní tabulky, tedy přímo v prostředí MS JDBE. Vedly mě k tomu především dva důvody:

- chci, aby projekt bylo možno jednoduše zprovoznit bez jakýchkoliv dalších instalačních programů, aby ho bylo možné jednoduše přenášet a nebylo nutné používat žádné ODBC ovladače,
- chci splnit celý cíl, tedy prokázat, že je možné pomocí dosti rozšířeného vývojového prostředí vytvořit dostatečně robustní aplikaci.

Projekt jsem proto rozdělil na dvě části:

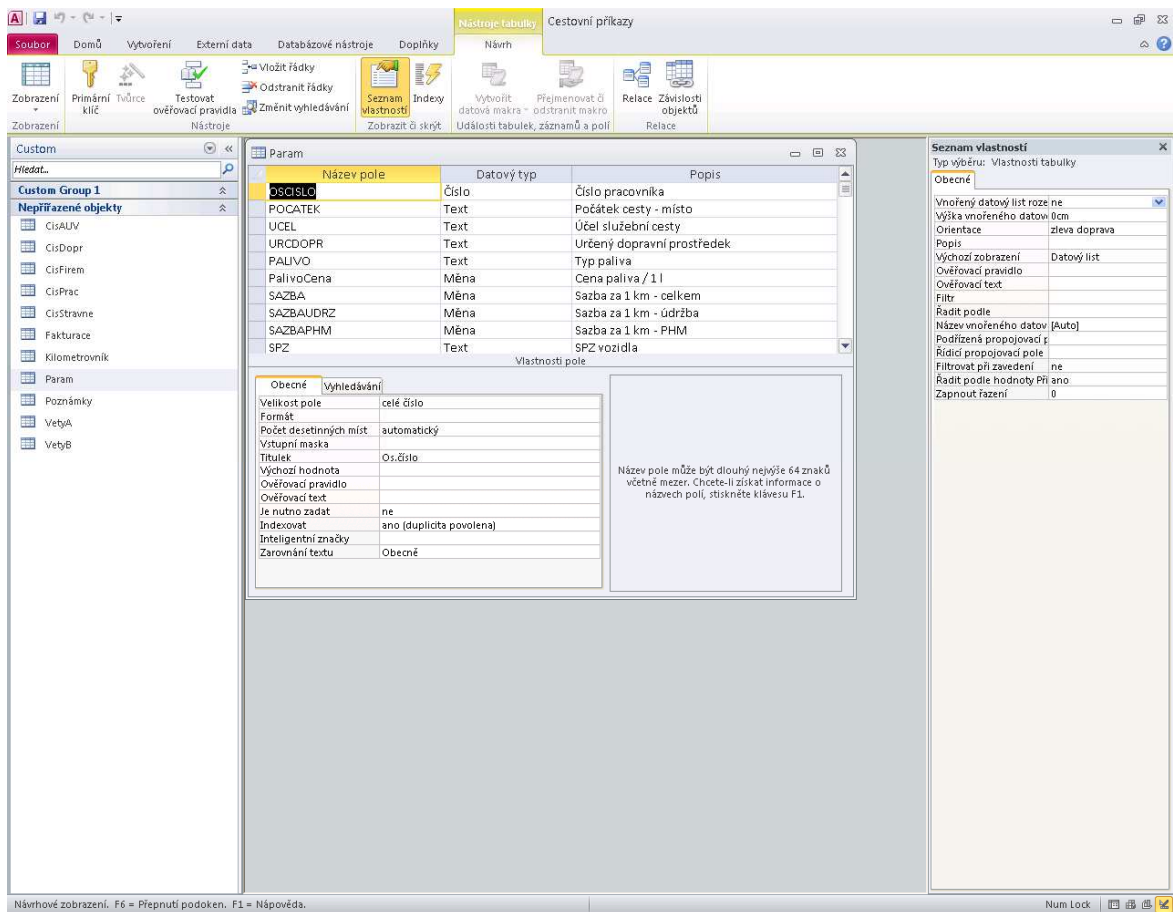
- Back-end databázi,
- Front-end databázi.

2.2.1 Back-end databáze

Back-end databáze je část projektu, která obsahuje pouze data. Pokud by v praxi nastala situace, že by výkon MS JDBE nebyl, s ohledem na množství dat, dostačující, bylo by možné tuto část projektu opustit a propojení tabulek provést ve Front-end databázi pomocí ODBC ovladačů na některý SQL server.

Back-end databáze obsahuje jednotlivé tabulky projektu a v nich jsou definovány sloupce (údaje). Z dalších vlastností údajů jsou uvedeny: datový typ, popis, rozsah, případný formát, výchozí hodnota a omezující podmínky. Kvůli efektivní propojitelnosti tabulek a kvůli rychlejšímu přístupu k záznamům jsou definované primární klíče a indexy⁷.

⁷ PÍSEK, S., *Access 2007: podrobný průvodce*, s. 47.



Obr. 5 – Návrh tabulky, zdroj vlastní obráček

Při návrhu databáze jsem se snažil dodržet pravidla normalizace⁸. První tři jsou popsána takto:

- 1. normální forma – relace R je v první normální formě (1NF), jsou-li všechny její atributy atomické (dále již nedělitelné).
- 2. normální forma - relace R je ve druhé normální formě (2NF), je-li v první normální formě a jestliže pro každý neklíčový atribut platí, že je úplně funkčně závislý na primárním klíči.
- 3. normální forma - relace R je v třetí normální formě (3NF), je-li ve druhé normální formě a platí-li, že žádný neklíčový atribut není tranzitivně závislý na žádném klíči relace R.

Snažil jsem se dodržet návrh databáze ve třetí normální formě, proto jsem rozdělil data do několika tabulek, aby byly všechny neklíčové atributy vzájemně nezávislé.

⁸ <http://databases.about.com/od/specificproducts/a/normalization.htm>

Databázi jsem tedy rozdělil na následující tabulky:

CisAUV

Jedná se o číselník dopravních prostředků typu vlastních automobilů (AUV). Primárním klíčem tabulky je SPZ vozidla. Dále tabulka obsahuje název vozidla, druh paliva, kombinovanou spotřebu dle TP, implicitní sazbu za údržbu a implicitní cenu paliva. Tyto implicitní hodnoty se při pořízení cestovního příkazu pouze nabízí, protože se mění buďto zákonem (sazba za údržbu) nebo na trhu (cena paliva) a je možné je přepsat.

CisDopr

Tento číselník je seznamem použitelných dopravních prostředků. Primárním klíčem je SPZ a obsahuje také název vozidla.

CisFirem

Jedná se o číselník míst výkonu práce. Primárním klíčem je složený klíč z údajů Podnik a Obec. Dále je obsahem zkratka místa výkonu práce pro zjednodušení zadávání a vzdálenost od místa vlastní firmy. Vzdálenost je využita pro automatické doplnění při generování prvního, příp. posledního řádku cestovního příkazu za předpokladu, že je počátek, příp. konec pracovní cesty v místě sídla vlastní firmy.

CisPrac

Program počítá s možností vytváření cestovních příkazů pro více osob. Proto jsem vytvořil tabulku CisPrac. Tento číselník obsahuje seznam pracovníků a mimo vazby na opakující se údaje (např. jméno) v dotazech založených na tabulce VetyA, slouží především k dosazování implicitních hodnot. Primárním klíčem je osobní číslo pracovníka. Dále jsou zde další potřebné údaje:

- jméno, příjmení titul,
- ulice, PSČ, obec,
- telefon, pracovní doba od-do,
- implicitní počátek a konec výkonu práce,
- implicitní účel pracovní cesty,

- implicitní určený dopravní prostředek.

CisStravne

K doplnění správné sazby za stravné slouží tento číselník náhrad stravného. Obsahuje částky podle jednotlivých roků a pásem a program v něm vyhledává na základě hodnoty roku a intervalu doby výkonu práce.

Fakturace

Tato tabulka obsahuje průběžně ukládané záznamy pro fakturaci cestovného zákazníkům. Neobsahuje unikátní primární klíč, je zde pouze vazba na konkrétní cestovní příkaz. Dále je obsahem tabulky název a místo firmy, počet fakturovaných kilometrů a popis práce.

Kilometrovnik

Jedná se o pomocnou tabulku se vzdálenostmi mezi jednotlivými místy výkonu práce. Primárním klíčem je složený klíč z údajů firma a sídlo odjezdu a příjezdu. Dalším údajem je vzdálenost a obvyklá doba jízdy osobním automobilem. V tabulce se hledá oboustranně, protože vzdálenost z místa A do místa B je obvykle stejná.

Param

Tato tabulka obsahuje globální parametry pro celý program. Neobsahuje žádný klíč a přístup k údajům je prováděn pomocí doménové agregační funkce DLookup. Hlavním účelem tohoto souboru je doplňování výchozího osobního čísla a zadávání datumového intervalu pro tisk sestav.

VetyA

Tato tabulka obsahuje hlavičky cestovních příkazů. Primárním indexem je číslo příkazu. Dále tabulka obsahuje číslo pracovníka, místo, datum a čas výkonu práce, účel pracovní cesty, spolucestující, typ a SPZ určeného dopravního prostředku, použité palivo, cena paliva a sazba za údržbu.

VetyB

Tato tabulka obsahuje položky cestovních příkazů. Tabulka neobsahuje unikátní index, vazbu na hlavičky zajišťuje údaj číslo dokladu. Dalším sloupcem je pořadí řádku

v rámci příkazu, datum cesty, čas a místo odjezdu a příjezdu, čas výkonu práce, typ dopravního prostředku, který může být různý při kombinované dopravě, hodiny ztráty času, ujeté kilometry, jízdné, stravné, nocležné a částku vedlejších výdajů.

2.2.2 Front-end databáze

Front-end databáze je vlastně klientská aplikace, ve které jsou propojeny tabulky z Front-end databáze, obsahuje formuláře, sestavy, makra a zdrojový VBA kód. Výhodou a taky hlavním důvodem tohoto rozdělení je to, že mohu uživateli aktualizovat jeho program, aniž bych potřeboval jeho data a on může do poslední chvíle pracovat. Obvykle autor více zasahuje do vlastního programu než do datové základny.

2.2.3 Pás karet

V každém prostředí je vhodné dodržovat zavedené zvyklosti a design. Řešil jsem proto problém jak použít pás karet v uživatelském rozhraní Fluent zavedeném od verze Microsoft Office 2007⁹. Vytvořil jsem a naplnil jsem speciální systémovou tabulku uSysRibbons, která v běžném zobrazení není vidět. Je nutné povolit zobrazení systémových objektů. Tabulka obsahuje údaje ID, RibbonName, RibbonXML a Describe. Definice pásu je ve sloupci RibbonXML ve formátu XML.

```
<customUI
xmlns="http://schemas.microsoft.com/office/2006/01/customui">

  <ribbon startFromScratch="false">

    <tabs>

      <tab id="dbCustomTab" label="Cestovní příkazy - nabídka"
visible="true">

        <group id="dbCiselniky" label="Číselníky">

          <button id="DataCisFirem" label="Firmy"
imageMso="ViewsFormView" onAction="RibbonMenu.frmCisFirem"/>

          <button id="DataKilometrovnik" label="Kilometrovník"
imageMso="ViewsFormView" onAction="RibbonMenu.frmKilometrovnik"/>

          <menu id="MyMenuNextLevel1" label="Další"
imageMso="ViewsFormView" itemSize="normal">

            <button id="CisAUV" label="Auta"
imageMso="ViewsFormView" onAction="RibbonMenu.frmCisAUV"/>

            <button id="CisPrac" label="Pracovníci"
imageMso="ViewsFormView" onAction="RibbonMenu.frmCisPrac"/>


```

⁹ PIERCE, J., *Mistrovství v Microsoft Office 2007*, s. 744.

```

        <button id="CisStravne" label="Sazby stravného"
imageMso="ViewsFormView" onAction="RibbonMenu.frmCisStravne"/>
    </menu>
</group>
<group id="dbData" label="Data">
    <button id="DataHlavicky" label="Formulář pro zadávání"
onAction="RibbonMenu.frmHlavicky" size="large"
imageMso="ViewsFormView" supertip="Úplný formulář pro zadávání
cestovních příkazů."/>
</group>
<group id="dbSestavy" label="Sestavy">
    <button id="SestTiskVsechCestaku" label="Všechny cestáky"
onAction="RibbonMenu.rptTiskVsechCestaku" imageMso="ViewsReportView"/>
    <menu id="MyMenuNextLevel2" label="Další"
imageMso="ViewsReportView" itemSize="normal">
        <button id="SestCisFirem" label="Seznam firem"
onAction="RibbonMenu.rptCisFirem" imageMso="ViewsReportView"/>
        <menuSeparator id="myMenuSep1"/>
        <button id="SestFakturovatKc" label="Fakturace s Kč"
onAction="RibbonMenu.rptFakturaceKc" imageMso="ViewsReportView"/>
        <button id="SestSumarUctarna" label="Sumář pro
účtárnu" onAction="RibbonMenu.rptSumarUctarna"
imageMso="ViewsReportView"/>
    </menu>
</group>
<group id="dbExporty" label="Exporty">
    <control idMso="ExportExcel" label="Export do Excelu"
enabled="true"/>
</group>
<group id="GroupCommands" label="Příkazy">
    <button idMso="FilePrintQuick" size="large"/>
    <button idMso="FileSendAsAttachment" size="large"/>
    <button idMso="PublishToPdfOrEdoc" size="large"/>
</group>
<group idMso="GroupClipboard"></group>
<group idMso="GroupSortAndFilter"></group>
<group idMso="GroupFindAccess"></group>
</tab>

```

```
</tabs>  
</ribbon>  
</customUI>
```

Poté se tento pás karet nastaví ve volbě **Možnosti** aplikace Access, v položce **Aktuální databáze**, v části **Možnosti pásu a panelu nástrojů**. Velkým zjednodušením ve verzi 2010 je možnost vytvářet vlastní pás karet ve volbě **Přizpůsobit pás karet**¹⁰.

2.2.4 Propojení tabulek

Po spuštění programu jsem nastavil jako výchozí modální formulář SplashScreen. U něj je ošetřena událost Form_Load tak, aby bylo zajištěno, že budou tabulky z Back-end databáze propojeny do Front-end databáze nacházející se ve stejném adresáři. Tím je zabezpečeno, že pokud přesuneme data do jiné složky, program si tabulky automaticky znovu připojí (viz následující ukázka části kódu procedury):

```
db.TableDefs(i).Connect <> ""  
Dim db As Database, path As String, i As Integer  
Set db = DBEngine.Workspaces(0).Databases(0)  
path = Application.CurrentProject.path & "\Cestáky.mdb"  
txtOut1 = "Program pro zpracování cestovních příkazů" & vbCrLf & _  
        "(C) 2012-2013 by Jaroslav Vrána" & vbCrLf & _  
        "Umístěn v "&Application.CurrentProject.FullName & vbCrLf  
txtOut2 = "Reconnect tabulek: "  
For i = 0 To db.TableDefs.Count - 1  
    If db.TableDefs(i).Connect <> "" Then  
        txtOut2 = txtOut2 & IIf(i>0, ", ", "") & db.TableDefs(i).Name  
        db.TableDefs(i).Connect = ";Database=" & path  
        db.TableDefs(i).RefreshLink  
    End If  
Next i
```

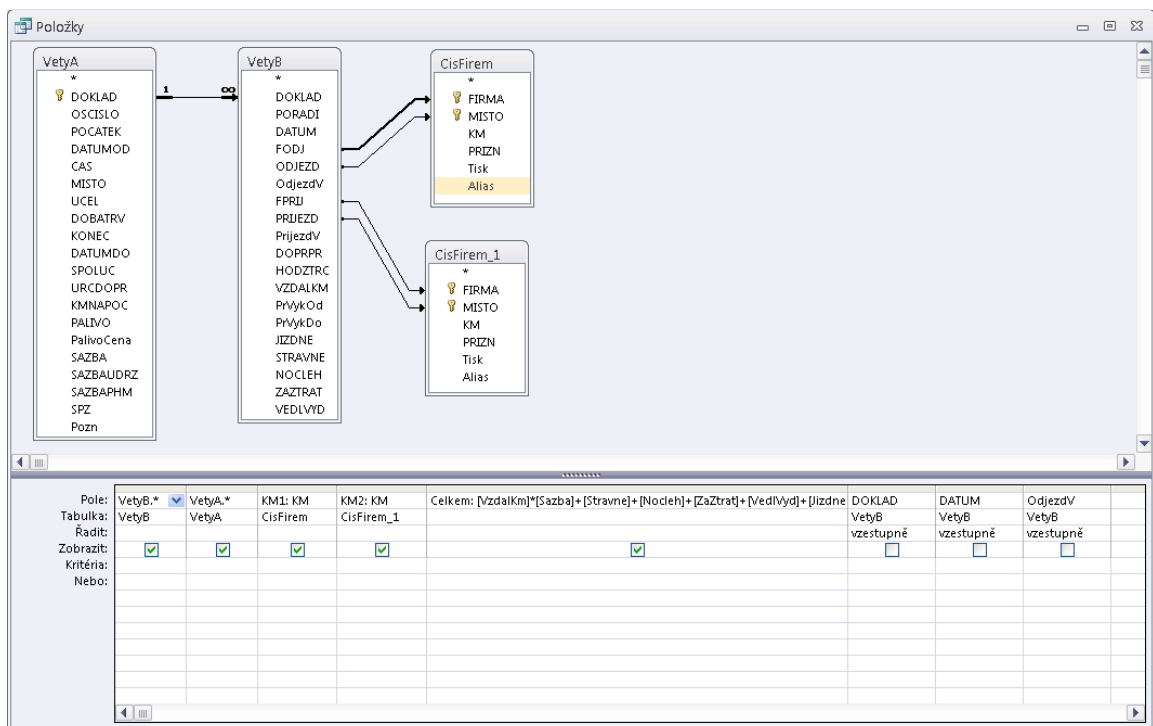
¹⁰ PÍSEK, S., *Access 2010: podrobný průvodce*, s. 16.

2.2.5 Způsoby práce s daty

Pro práci s daty jsem se snažil využít různé možnosti, které MS Access nabízí. Například zdroj záznamů pro položky cestovních příkazů řeší tento SQL skript:

```
SELECT
    VetyB.*, VetyA.*,
    CisFirem.KM AS KM1, CisFirem_1.KM AS KM2,
    [VzdalKm]*[Sazba]+[Stravne]+[Nocleh]+[ZaZtrat]+[VedlVyd]+[Jizdne] AS
    Celkem
FROM
    VetyA LEFT JOIN ((CisFirem RIGHT JOIN VetyB ON (CisFirem.FIRMA =
    VetyB.FODJ) AND (CisFirem.MISTO = VetyB.ODJEZD)) LEFT JOIN CisFirem AS
    CisFirem_1 ON (VetyB.FPRIJ = CisFirem_1.FIRMA) AND (VetyB.PRIJEZD =
    CisFirem_1.MISTO)) ON VetyA.DOKLAD = VetyB.DOKLAD
ORDER BY
    VetyB.DOKLAD, VetyB.DATUM, VetyB.OdjezdV;
```

Výsledkem je sada záznamů se všemi řádky z tabulky VetyB a ke každému z nich jsou přidány údaje z tabulky VetyA a implicitní kilometry ze sídla firmy do místa výkonu práce a zpět. V návrhovém zobrazení takové propojení vypadá následovně:

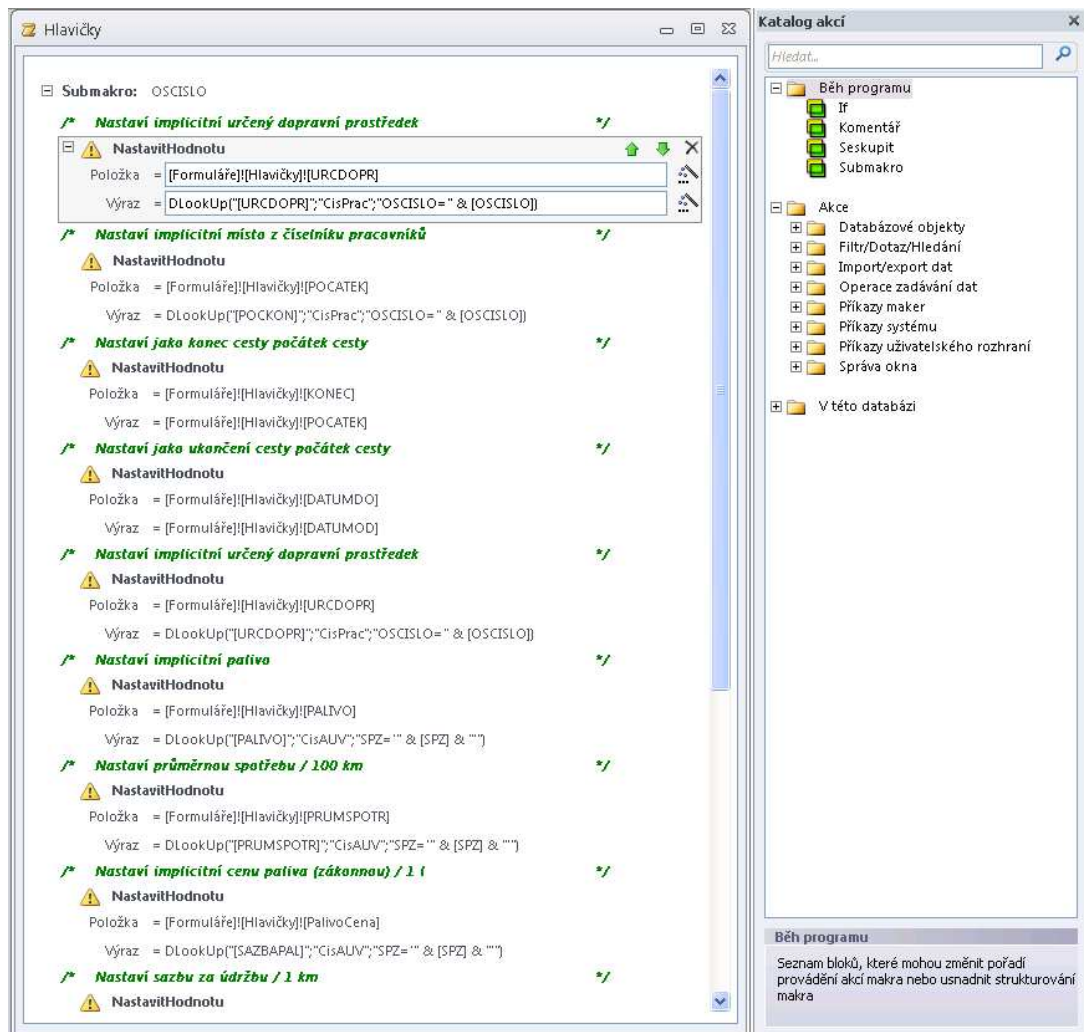


Obr. 6 – Návrh dotazu, zdroj vlastní obrázek

Sumarizaci částek jednotlivých příkazů zajišťuje dotaz:

```
SELECT
VetyB.DOKLAD, First(VetyA.DATUMOD) AS FirstOfDATUMOD,
First(VetyA.MISTO) AS FirstOfMISTO, First(VetyA.UCEL) AS FirstOfUCEL,
First(VetyA.Pozn) AS FirstOfPozn, Sum(VetyB.VZDALKM) AS SumOfVZDALKM,
Sum(VetyB.STRAVNE) AS SumOfSTRAVNE,
Sum([VzdalKm]*[Sazba]+[Stravne]+[Nocleh]+[ZaZtrat]+[VedlVyd]) AS
CelkemZaKm,
Sum([VzdalKm]*[Sazba]+[Stravne]+[Nocleh]+[ZaZtrat]+[VedlVyd]) AS
Celkem,
Int([Celkem]+0.49999) AS CelkemZ,
First(Month([Datum]))/100+First(Year([Datum])) AS Obdobi,
VetyA.OSCISLO
FROM VetyA LEFT JOIN VetyB ON VetyA.DOKLAD = VetyB.DOKLAD
GROUP BY VetyB.DOKLAD, VetyA.OSCISLO;
```

Dále jsou v programu využita makra. Pomocí nich je řešeno například doplnění implicitních hodnot v nové hlavičce nebo akce ve vlastním pruhu nabídek (RibbonMenu). K naplnění výchozích hodnot slouží následující makro Hlavičky:



Obr. 7 – Návrh makra, zdroj vlastní obrázek

V programu je mnoho operací se záznamy v tabulkách prováděno přímo ve VBA kódu. V něm jsou možnosti manipulace s daty zřejmě největší. Například lze volat již vytvořený SQL dotaz nebo ho lze nejprve sestavit, jak dokládá část kódu procedury pro doplnění kilometrovníku novými hodnotami přímo z pořízení položek pomocí příkazu DoCmd.RunSQL:

```
DoCmd.RunSQL ("INSERT INTO
Kilometrovník (FirmaZ, MistoZ, FirmaDo, MistoDo, Km) " & _
"SELECT ' " & FODJ & "', ' " & ODJEZD & "', ' " & FPRIJ & "', ' " & _
PRIJEZD & "', ' " & CStr (VZDALKM) & _
" FROM VetyA " & _
"WHERE (Doklad=" & CStr (DOKLAD) & ")")
```

Využit lze i naplnění přes buffer klávesnice. Kupříkladu doplnění hodnot do nového záznamu v podkladu pro fakturaci zajistí tyto příkazy:

```
DoCmd.OpenForm "Fakturovat2", acNormal
DoCmd.GoToRecord , , acNewRec
SendKeys Str(Me.DOKLAD) &"~"& Str(Me.DATUM) &"~"& Me.ODJEZD &"~"& _
Me.FODJ &"~"& Str(Me.KM1 * 2) &"~"
```

Hodnotu dat lze přiřadit přímo, třeba na základě doménového agregačního dotazu:

```
POCATEK = DLookup("[POCKON]", "CisPrac", "OSCISLO=" & [OSCISLO])
KONEC = POCATEK
```

Další možností je přistoupit k datům pomocí rozhraní ADO nebo DAO. U stravného je potřeba sazbu zjištěnou z číselníku zapsat do posledního záznamu cestovního příkazu. Je však nutné zjistit, zda už nějaké stravné není uloženo a v případě jiné hodnoty si vyžádat přepsání. Přesně toto řeší v programu následující procedura:

```
Private Sub STRAVNE(Hodin As Integer)
    Dim rst As DAO.Recordset
    Dim STRAVNE As Integer

    On Error GoTo Err_STRAVNE
    'Zjistí stravné z číselníku
    STRAVNE=DLookup("[Stravne]", "CisStravne", "(Rok=" & Year(DATUMDO) & ") _
        AND (HodOd=" & CStr(Hodin) & ")")
    Set rst = Form_Položky.Recordset

    With rst
        'Podívej se na poslední řádek
        .MoveLast
        'Pokud je uloženo (i nulové) stravné jiné než nalezené v číselníku
        If STRAVNE <> !STRAVNE Then
            'Je-li uloženo stravné jiné než nalezené v číselníku, vyžádej si
            'svolení k přepsání
            If !STRAVNE <> 0 Then
```

```

    If MsgBox("Chcete přepsat stravné " & !STRAVNE & " na " & _
        STRAVNE & "?", vbYesNo) = vbYes Then
        .Edit
        !STRAVNE = STRAVNE
        .Update
    End If
'Je-li uložené stravné nulové, tak ho rovnou doplň
Else
    .Edit
    !STRAVNE = STRAVNE
    .Update
End If
End If
End With

Exit_STRAVNE:
    Exit Sub

Err_STRAVNE:
    MsgBox Err.Description
    Resume Exit_STRAVNE
End Sub

```

Při ladění funkčnosti VBA kódu jsem ve velké míře využíval Immediate okno. To umožňuje okamžité vykonání příkazu¹¹.

2.2.6 Akční dotazy v run-time modulu

Program je možné spustit v plné verzi prostředí MS Access nebo v tzv. Run-time modulu. Zatímco v plné verzi je možný plný vývoj aplikace, run-time slouží pouze ke spuštění. Jeho nevýhodou je také absence některých nastavení, jako třeba vypnutí akčních dotazů. Akční dotaz je potvrzení určité akce uživatelem, např. výmaz dat nebo jejich změna. Naštěstí je možné vypnout potvrzování programově:

¹¹ VINE, Michael A., *Microsoft Access VBA programming for the absolute beginner*, s. 189.

```
Application.SetOption "Confirm Action Queries", 0
```

```
Application.SetOption "Confirm Document Deletions", 0
```

```
Application.SetOption "Confirm Record Changes", 0
```

2.2.7 Postup vytváření cestovního příkazu

Program je navržen tak, aby obsluze umožnil co nejefektivnější pořízení cestovních příkazů s maximálním komfortem. Obzvláště jsem se snažil o minimalizaci úkonů s klávesnicí. Běžně lze vyplnit a vytisknout příkaz, který řeší pracovní cestu na několik míst výkonu práce, bez jediného stisku klávesy.

Výběr pracovníka je možné zadat výběrem z rozbalovacího pole se seznamem. Po jeho výběru se vyvolá událost **Při změně** a ve VBA proceduře se provede několik základních akcí. Nejprve se zjistí poslední číslo dokladu nižšího než je aktuální pro uvedeného pracovníka. To slouží k zapamatování posledního data jeho příkazu. K němu se, existuje-li, přičte jeden den a nabídne se toto nové datum jako datum odjezdu právě vyplňovaného příkazu a z parametrů pracovníka se nabídne výchozí čas odjezdu. Zároveň se datum odjezdu vyplní i do data příjezdu. Dále se naplní počátek a konec výkonu práce, účel pracovní cesty, určený dopravní prostředek, SPZ vozidla, použité pohonné hmoty a sazba za údržbu vozidla. Pokud přepíšeme cenu paliva, například na základě přiloženého dokladu o nákupu, přepočteme sazbu za kilometr stiskem tlačítka **PHM**. Obvykle provádí pracovník čerpání pohonných hmot jednou za několik dní, tedy do zásoby, a proto program přednostně opisuje cenu z předchozího dokladu pracovníka. Pokud by obsluha změnila v některém dřívějším, již existujícím dokladu cenu paliva, může u následujících příkazů využít tlačítko **min.věta** k opravě údajů následného dokladu. Program se dotáže, v případě změny, zda provést změnu aktuálního záznamu. Pokud pracovník užívá více vozidel, může je vybrat z pole se seznamem a program automaticky změní příslušné původní hodnoty (SPZ, použité palivo, sazby).

Další část hlavičky umožňuje největší optimalizaci pořízení dat. Pomocí tlačítka **Přidej místo** se otevře číselník firem (míst výkonu práce) a po dvojkliku na konkrétním řádku doplní z této tabulky alias do údaje **Místo jednání**. Pokud je těchto míst více, vloží program čárku a další místo. Takto lze postupně přidat všechna místa, na která pracovník během služební cesty jel. V mém případě jsou to obvykle dva až tři zákazníci denně. Samozřejmě lze tato místa vyplňovat i ručně; hlavně v případě, kdy se jedná o jednorázového zákazníka, kterého nemáme a ani nechceme mít trvale v číselníku

firm. Na základě takto vyplněného **Místa jednání** se vygenerují položky příkazu. To se provede tlačítkem **Go!**. Program nejprve provede kontrolu, zda je v hlavičce vyplněno místo a datum počátku a konce pracovní cesty a **Místo jednání**. Pokud tomu tak je, rozebere údaj **Místo jednání** na jednotlivé části. Poté vygeneruje příslušný počet řádků s pořadovým číslem, místem a vzdáleností, kterou najde buďto v číselníku firm nebo v kilometrovníku. Pokud firma neexistuje nebo dosud neznáme vzdálenost od předchozího bodu pracovní cesty, bude uvedena vzdálenost 0 km. Tu můžeme přepsat a stiskem tlačítka **Dopl** v položkách příkazu vložit do kilometrovníku. Tím program průběžně pohodlně naučíme síťové vzdálenosti pro využití v budoucnu. Vzdálenost z místa A do místa B stačí mít zadanou pouze jednou, opačný směr se z tohoto vztahu samozřejmě dohledá. První řádek položek si převezme místo, datum a čas odjezdu z hlavičky. Čas odjezdu je možno také zkorigovat myší. K tomu slouží tlačítka na +/- **5, 10, 15, 30** a **60**, která upraví čas o příslušný počet minut. Z rozbalovacího seznamu pro hodiny a minuty lze zvolit celkovou dobu pracovní cesty. Tento postup má smysl, proběhla-li cesta v jednom dni a takto zvolená doba se doplní v posledním řádku po sečtení s časem začátku cesty po stisku tlačítka **Čas ukončení**. Velmi často mají pracovníci na služební cestě nárok na stravné. To lze jednoduše doplnit po stisku příslušného tlačítka **5, 12** nebo **18**, což odpovídá minimální době pracovní cesty v hodinách a tomu odpovídající výši stravného. Částka se najde v číselníku stravného podle roku a hodin. Po vyplnění se v hlavičce zobrazí celková částka a vykázaná vzdálenost za příkaz.

Z dalších drobností, které zefektivňují práci je tlačítko +/- **1 den**, které umožní upravit datum počátku a konce pracovní cesty. Na stanovení počátku/konce pracovní cesty slouží i tlačítka **1** a **2**, která doplní výchozí místo z parametrů osoby v číselníku pracovníků. Často může být počátkem a koncem pracovní cesty bydliště nebo pobočka vlastní firmy. Funguje tak, že pokud neexistuje vyplněné datum, přičte/odečte jeden den z minulého záznamu pracovníka, existuje-li tak z tohoto data. Usnadnění vykazování podkladů pro fakturaci zákazníků umožní tlačítko **Fa** v položkách. Je umístěno za každým místem příjezdu a odjezdu. Podle toho doplní do spodní části formuláře záznam s místem a názvem podniku také číslo dokladu, datum a vzdálenost. Do tohoto řádku ještě můžeme doplnit i text s bližším popisem práce nebo jinou poznámkou. K jedné hlavičce může být více záznamů pro fakturaci. Další poznámku můžeme zapsat k samotnému příkazu v hlavičce.

Cestáky - formulář pro zadávání

CESTOVNÍ PŘÍKAZ č. 103 541

Ds. číslo: 17
 Útvar: Podpora IS
 Telefon: 604 658 026
 Normální pracovní doba: od 7:00 do 15:00

1. Příjm. jméno, titul: **Vrána Jaroslav** tento den je středa
 2. Bydliště:

1. Počátek cesty (místo, datum, hodina): Prostějov 19.12.2012 8:10 Místo jednání: Spesk, Výšovice Účel cesty: Aplikacní činnost Konec cesty (místo, datum): Prostějov 19.12.2012 **min věta**

3. Spolucestující: 4. Urč. dopr. pr.: AUV C. paliva: PHM Údržba Celkem

4. Při použití vlastního vozidla: druh: Renault Laguna Grandtour 1.8 SPZ: 3M4 99-11 PHM Natural 7,8 34,90 2,72 3,70 6,42

Nástroje Hledej problém Přidej místo Tiskni příkaz +1den -1den

Čas příjezdu: +5 +10 +15 +30 +60 -5 -10 -15 -30 -60 **Go!** Čas ukončení: 5 12 15 Stravné: 191,56 Kč | 18 | 24

Datum	Odjezd	Příjezd	v hod.	Použ. prostř.	Ujeté km	Zač. prac. výk.	Kon. místní př.	Jízdné a místní př.	Stravné	Noc- ležné	Nutné vedl. výdaje	Celkem
19.12.2012	Prostějov	Prostějov	8:10	-	3	0:00	0:00	0,00	0,00	0,00	0,00	19,26
103541	Speciální škola, o	Prostějov, Teti	0:00	AUV	6	0:00	0:00	0,00	0,00	0,00	0,00	38,52
19.12.2012	Speciální škola, o	Prostějov, Teti	0:00	AUV	9	0:00	0:00	0,00	76,00	0,00	0,00	133,78
103541	ZD Výšovice	Výšovice	0:00	AUV	-	0:00	0:00					3
19.12.2012	ZD Výšovice	Prostějov	14:40									
103541												

Záznam: 1 z 3 Bez filtru Vyhledávání

Doklad	Datum	Místo	Firma	Km	Poznámka
103541	19.12.2012	Výšovice	ZD Výšovice	18	Instalace modulu Účetnictví
103541	19.12.2012	Prostějov, Te	Speciální škola, o.p.s.	6	Konfigurace Kerio Mail serveru

Záznam: 1 z 2 Bez filtru Vyhledávání

Záznam: 4260 z 4266 Bez filtru Vyhledávání

Obr. 8 – Typický, pouze myší vyplněný doklad, zdroj vlastní obrázek

2.2.8 Tiskové výstupy

Nejdůležitější tiskovou sestavou projektu je vlastní tiskopis cestovního příkazu. Tisknout můžeme jednotlivé příkazy přímo ve formuláři pro jejich zadávání. Pokud potřebujeme tisk určitého intervalu příkazů využijeme nabídku **Všechny cest'áky** a v náhledu vybereme rozsah stran.

CESTOVNÍ PŘÍKAZ č. 103541		Os. číslo <u>17</u>									
796 01 Prostějov		Útvar <u>Podpora IS</u>									
IČO: 25 50 31 38		Telefon <u>604 658 026</u>									
1. Příjmení, jméno, titul <u>Vrána Jaroslav</u>		Normální pracovní doba od <u>7:00</u> do <u>15:00</u>									
2. Bydliště <u>[redacted]</u>											
Počátek cesty (místo, datum, hodina)	Místo jednání	Účel cesty	Konec cesty (místo, datum)								
<u>Prostějov 19.12.2012 8:10</u>	<u>Spesk, Výšovice</u>	<u>Aplikační činnost</u>	<u>Prostějov 19.12.2012</u>								
3. Spolucející		4. Určený dopravní prostředek <u>AUV</u>									
4. Při použití vlastního vozidla: druh <u>Renault Laguna Grandtour 1.8 16v SPZ 3M4 99-11</u> č. pojistky											
PHM <u>Natural</u> průměrná spotřeba dle technického průkazu <u>7,8</u> l / 100 km, cena 1l PH <u>34,90</u> , sazba za údržbu <u>3,70</u> Kč/1 km											
5. Předpokládaná částka výdajů Kč											
6. Povolená záloha Kč											
vyplacená dne		pokladní doklad č.									
Podpis pokladníka		Datum a podpis pracovníka odpovědného k povolení cesty									
VYÚČTOVÁNÍ PRACOVNÍ CESTY											
7. Zpráva o výsledku pracovní cesty byla podána dne											
Se způsobem provedení souhlasí		Datum a podpis odpovědného pracovníka									
ÚČTOVACÍ PŘEDPIS											
8. Výdajový - příjmový pokladní doklad číslo		Má dát	Dal	Částka	Středisko	Zakázka					
Účtovaná náhrada byla přezkoušena a upravena na Kč		Hra	ná								
Vyplacená záloha Kč		ze	kla								
Doplatek - přeplatek <u>192,00</u> Kč		no	dů								
Slovy: <u>jednostodevadesát dvě</u>		z	zisku								
Datum a podpis pracovníka, který upravil účtování		Poznámka o zaúčtování									
Datum a podpis příjemce (průkaz totožnosti)		Datum a podpis pokladníka		Schválil (datum a podpis)							
Datum	Odjezd Příjezd	hod.	Použ. dopr. průstř.	Ujeté km	Začátek a konec prac.výk	Jízdné a místní přeprava	Stravné (příp.na stravné)	Noclež- né	Nutné vedlejší výdaje	Celkem	Upraveno
19.12.2012	<u>Prostějov</u>	<u>8:10</u>	<u>AUV</u>	<u>3</u>	<u>0:00</u>	<u>0,00</u>	<u>0,00</u>	<u>0,00</u>	<u>0,00</u>	<u>19,26</u>	
	<u>Speciální škola, o.p.s. Prostějov, Tetř</u>	<u>0:00</u>			<u>0:00</u>						
19.12.2012	<u>Speciální škola, o.p.s. Prostějov, Tetř</u>	<u>0:00</u>	<u>AUV</u>	<u>6</u>	<u>0:00</u>	<u>0,00</u>	<u>0,00</u>	<u>0,00</u>	<u>0,00</u>	<u>38,52</u>	
	<u>ZD Výšovice</u>	<u>0:00</u>			<u>0:00</u>						
	<u>Výšovice</u>	<u>0:00</u>			<u>0:00</u>						
19.12.2012	<u>ZD Výšovice</u>	<u>0:00</u>	<u>AUV</u>	<u>9</u>	<u>0:00</u>	<u>0,00</u>	<u>76,00</u>	<u>0,00</u>	<u>0,00</u>	<u>133,78</u>	
	<u>Prostějov</u>	<u>14:40</u>			<u>0:00</u>						
Celkem				<u>18</u>		<u>0,00</u>	<u>76,00</u>	<u>0,00</u>	<u>0,00</u>	<u>191,56</u>	
Záloha											
Doplatek - Přeplatek										<u>192,00</u>	
Datum a podpis účtovatele											

Obr. 9 – Výtisk cestovního příkazu z programu, zdroj vlastní obrázek

Pro zobrazení částky slovy jsem vytvořil funkci ve VBA kódu. Číslo, které funkce vyjádří slovně, jsem omezil na hodnotu 99999. V případě extrémního nárůstu inflace je možné zdrojový kód jednoduše upravit.

```
Function CisloSlovy(Cislo)
```

```
' Funkce vrátí číslo slovy
```

```
Dim Jednotky As Integer, Desitky As Integer, Stovky As Integer
```

```
Dim Tisice As Integer, Desetitisice As Long, Statisice As Long
```

```
Dim s As String
```

```
Const Jedn = "jednadvě tři čtyřipět šest sedm osm devědeset"
```

```
Const Des = _
```

```
"deset dvacet třicet čtyřicet padesát šedesát sedmdesátosmdesát devadesát"
```

```
Const Nact = _
```

```
"jedenáct dvanáct třináct čtrnáct patnáct šestnáct sedmnáct osmnáct devatenáct"
```

```
If IsNull(Cislo) Then GoTo Exit_CisloSlovy
```

```
' Zjištění rozsahů čísel
```

```
Jednotky = Cislo Mod 10
```

```
Desitky = (Cislo - Jednotky) Mod 100
```

```
Stovky = (Cislo - Jednotky - Desitky) Mod 1000
```

```
Tisice = (Cislo - Jednotky - Desitky - Stovky) Mod 10000
```

```
Desetitisice = (Cislo - Jednotky - Desitky - Stovky - Tisice) Mod 100000
```

```
Statisice = (Cislo - Jednotky - Desitky - Stovky - Tisice - Desetitisice) Mod 1000000
```

```
' Nula
```

```
If Cislo = 0 Then s = "nula"
```

```
' Jednotky
```

```
If Desitky <> 10 And Jednotky <> 0 Then s = RTrim(Mid(Jedn, (Jednotky - 1) * 5 _  
+ 1, 5))
```

```

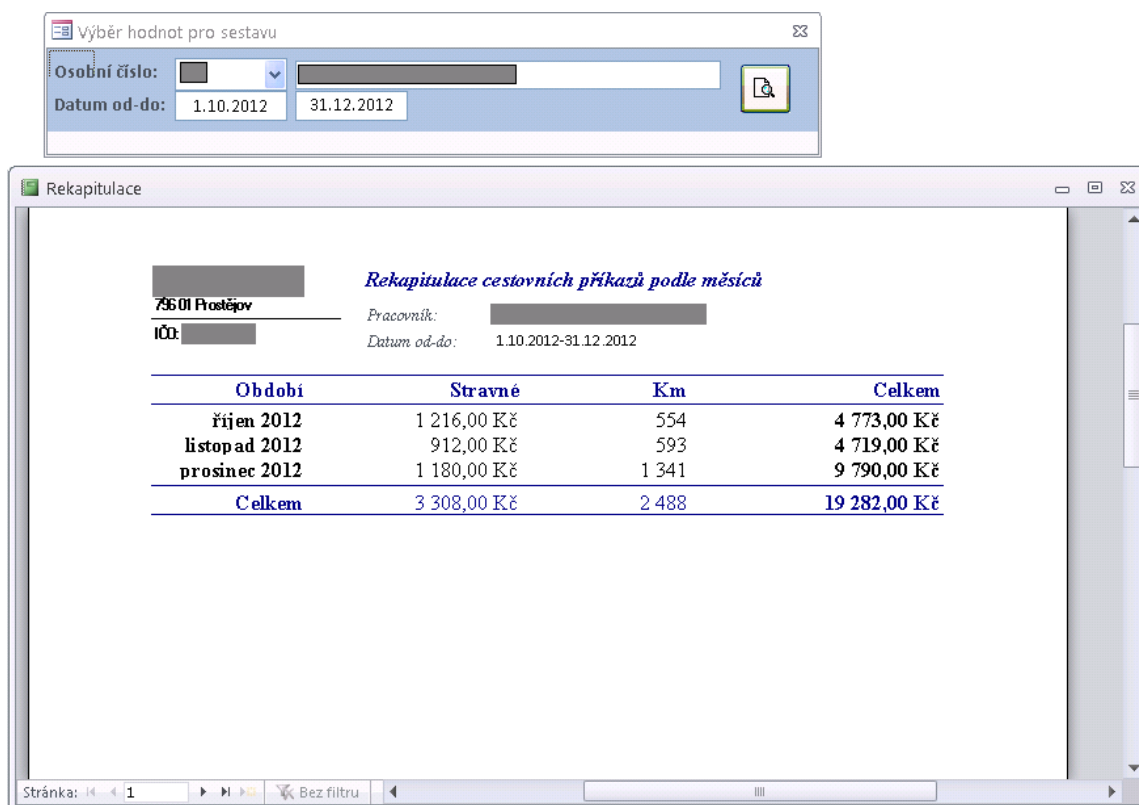
' Desítky
If Desitky > 10 Then s = RTrim(Mid(Des, ((Desitky / 10) - 1) * 9 + 1, 9)) + s
If Desitky + Jednotky = 10 Then s = RTrim(Mid(Des, 1, 9)) + s
If Desitky = 10 And Jednotky > 0 Then s = RTrim(Mid(Nact, (Jednotky - 1) * 10 _
+ 1, 10)) + s
' Stovky
If Stovky = 100 Then s = "jednosto" + s
If Stovky > 100 And Stovky < 500 Then s = RTrim(Mid(Jedn, (Stovky / 100 - 1) _
* 5 + 1, 5)) + "sta" + s
If Stovky > 400 Then s = RTrim(Mid(Jedn, (Stovky / 100 - 1) * 5 + 1, 5)) + "set" + s
' Tisíce
If Desetitisice <> 10000 And Tisice = 1000 Then s = "jedentisíc" + s
If Desetitisice <> 10000 And Tisice = 2000 Then s = "dvatisíce" + s
If Desetitisice <> 10000 And Tisice > 2000 And Tisice < 5000 Then _
s = RTrim(Mid(Jedn, (Tisice / 1000 - 1) * 5 + 1, 5)) + "tisíce" + s
If Desetitisice <> 10000 And Tisice > 4000 Then s = RTrim(Mid(Jedn, _
(Tisice / 1000 - 1) * 5 + 1, 5)) + "tisíc" + s
' Desetitisíce
If Desetitisice > 10000 Then s = RTrim(Mid(Des, (Desetitisice / 10000 - 1) _
* 9 + 1, 9)) + s
If Desetitisice + Tisice = 10000 Then s = RTrim(Mid(Des, 1, 9)) + "tisíc" + s
If Desetitisice = 10000 And Tisice > 0 Then RTrim (Mid(Nact, (Tisice / 1000 - 1) _
* 10 + 1, 10)) + s

CisloSlovy = s
If Cislo > 99999 Then CisloSlovy = "???"

Exit_CisloSlovy:
End Function

```

Rekapitulaci cestovních příkazů lze vytisknout za zvolené období. Sestava obsahuje součty údajů **Stravné**, **Kilometry** a **Celkem** za jednotlivé měsíce. Rekapitulace je spustitelná z RibbonMenu **Cestovní příkazy – nabídka**, skupiny **Sestavy**, **Další / Sumář pro účárnu**.



Obr. 10 – Výpis rekapitulace cestovních příkazů za období, zdroj vlastní obrázek

Poslední sestavou v programu je podklad pro fakturaci. Ten významně usnadnil dřívější vyčíslování vzdáleností z papírových tiskopisů při předávání podkladů pro fakturaci účetní.

Uvedený rozsah sestav prozatím splnil nejnужnější požadavky, které jsem si kladl na počátku práce. Z dalšího používání programu jistě vyplyne potřeba dalších reportů.

Závěr

Hlavním cílem bakalářské práce bylo vytvořit v Microsoft Office, konkrétně Microsoft Access v praxi využitelný program, který umožní efektivní pořízení cestovních příkazů a zároveň dokázat, že toto prostředí je pro tvorbu podobných aplikací dostačující. Produkt jsem ověřil při svém vlastním zpracování cestovních příkazů. Ve své praxi jsem je po řadu let vyplňoval ručně, což je při téměř každodenních služebních cestách velmi úmorné.

K dosažení cíle jsem zvolil metodu analýzy a syntézy. Tyto metody se při tvorbě programu osvědčily, protože umožnily jak optimální návrh databáze, který respektuje pravidla normalizace, tak i tvorbu vlastního programu. Díky vlastní zkušenosti s problematikou jsem byl schopen významně zefektivnit samotné pořízení cestovního příkazu. Jednoduchým způsobem je zajištěno vkládání míst výkonu práce, generování řádků i průběžné doplňování nových záznamů do kilometrovníku. Dalším plusem programu je tvorba podkladů pro fakturaci přímo při vytváření příkazu a různé tiskové výstupy.

Cestovní příkazy jsem posledního půl roku vytvářel výhradně pomocí tohoto programu. Jako mnoho jiných lidí, nemám rád vyplňování papírových formulářů. Většinu služebních cest mám ke stálým zákazníkům, jejichž názvy i síťové vzdálenosti k ostatním klientům postupně do databáze vkládám. Nyní jsem opravdu schopen vytvořit běžný cestovní příkaz jen pomocí myši, aniž bych použil klávesnici. Tím se vyplňování stává výrazně komfortnějším.

Jako vedlejší efekt jsem si díky této bakalářské práci prohloubil znalosti ve vývoji aplikací v prostředí Microsoft Office, což se mi může v další profesionální praxi IT konzultanta hodit. Zajisté by bylo možné tuto problematiku řešit i v jiných vývojových prostředích. Pravděpodobně stejně dobrou nebo lepší variantou by byl Visual FoxPro nebo Microsoft Visual Studio v jeho nejnovější verzi 2012 s databází v Microsoft SQL 2012 třeba v bezplatné Express edici. Pro MS Access jsem se rozhodl, protože jsem chtěl využít znalosti nabyté o tomto prostředí během studia.

Cíl, vytýčený v úvodu práce, tedy prokázat, že vývojové prostředí Microsoft Office je dostačující k vytvoření dílčích administrativních agend, konkrétně tvorby cestovních příkazů, byl splněn. Věřím, že by tento program mohl být užitečný i pro mnoho dalších lidí, kteří při výkonu své práci využívají vlastní vozidlo. Dovedu si rovněž představit další rozvoj produktu, především v oblasti statistického vyhodnocení, tvorby grafů nebo exportu a importu dat.

Anotace

Příjmení a jméno autora:	Vrána Jaroslav
Instituce:	Moravská vysoká škola Olomouc
Název práce v českém jazyce:	Návrh databázového systému "Cestovní příkazy" v Microsoft Accessu
Název práce v anglickém jazyce:	Design of a Travel Orders System Database in Microsoft Access
Vedoucí práce:	Mgr. Zdeňka Krišová
Počet stran:	48
Počet příloh:	0
Rok obhajoby:	2013
Klíčová slova v českém jazyce:	Cestovní příkaz, automobil, Microsoft Access, VBA.
Klíčová slova v anglickém jazyce:	Travel order, car, Microsoft Access, VBA.

Tato bakalářská práce popisuje řešení podnikové agendy Cestovní příkazy. Zaměřuje se na část procesu pro zaměstnance, ve kterém tyto příkazy vyhotovuje. Hlavním cílem je návrh prostřednictvím programu Microsoft Access, popis vytváření cestovního příkazu a ukázání důležitých vlastností tohoto nástroje.

This bachelor thesis describes the solution of business agenda Travel orders. It is focused on the part of employee's process in which he needs these orders to prepare. The main target is the design throw the program Microsoft Access, the describing of creating a travel order and the demonstrating of important features of this tool.

Seznam literatury

ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. Vyd. 1. Překlad Bogdan Kiszka. Brno: Computer Press, 2007, 567 s. ISBN 978-802-5115-039.

KANISOVÁ, Hana a Miroslav MÜLLER. *UML srozumitelně*. 2. aktualiz. vyd. Brno: Computer Press, 2006, 176 s. ISBN 80-251-1083-4.

KOROL, Julitta. *Access 2007 programming by example with VBA, XML, and ASP*. Vyd. 1. Plano, Tex.: Wordware Pub., 2008, 937 s. ISBN 978-159-8220-421.

KUBÁLEK, Tomáš a Markéta KUBÁLKOVÁ. *Databázový systém Microsoft Office Access verze 2007 CZ*. Vyd. 1. Praha: Oeconomica, 2009, 236 s. ISBN 978-80-245-1518-2.

PECINOVSKÝ, Josef. *Excel a Access 2010: efektivní zpracování dat na počítači*. 2., aktualiz. vyd. Praha: Grada, 2011, 198 s. Průvodce (Grada). ISBN 978-80-247-3898-7.

PIERCE, John. *Mistrovství v Microsoft Office 2007*. Vyd. 1. Brno: Computer Press, 2008, 1120 s. ISBN 978-80-251-2066-8.

PÍSEK, Slavoj. *Access 2007: podrobný průvodce*. 1. vyd. Praha: Grada, 2007, 197 s. ISBN 978-802-4719-672.

PÍSEK, Slavoj. *Access 2010: podrobný průvodce*. 1. vyd. Praha: Grada, 2011, 160 s. ISBN 978-80-247-3653-2.

VINE, Michael A. *Microsoft Access VBA programming for the absolute beginner*. Vyd. 3. Boston, MA: Thomson Course Technology, 2007, 358 s. ISBN 15-986-3393-7.

CHAPPLE, Mike. *Database Normalization Basics* [online]. [cit. 2013-03-17]. Dostupné z: <http://databases.about.com/od/specificproducts/a/normalization.htm>

SVĚT SOFTWARE. *Přehled sad Office 2010* [online]. [cit. 2013-01-26]. Dostupné z: <http://www.svetsoftware.cz/microsoft-office-2010-pro-podnikatele/porovnani-edici>

Seznam obrázků

Obr. 1 – Papírová forma cestovního příkazu	9
Obr. 2 – Přehled sad Office 2010	10
Obr. 3 – Uživatelské prostředí	11
Obr. 4 – Prostředí pro VBA	15
Obr. 5 – Návrh tabulky	26
Obr. 6 – Návrh dotazu	32
Obr. 7 – Návrh makra	34
Obr. 8 – Typický, pouze myší vyplněný doklad	39
Obr. 9 – Výtisk cestovního příkazu z programu	40
Obr. 10 – Výpis rekapitulace cestovních příkazů za období	43