

Západočeská Univerzita v Plzni
Fakulta Aplikovaných Věd

DIPLOMOVÁ PRÁCE



Lenka Reinwartová

**Aplikační rozhraní pro geografickou datovou sadu
židovských hřbitovů**

Katedra matematiky

Vedoucí:
Ing. Karel Jedlička
Geomatika

2010

Děkuji vedoucímu Ing. Karlu Jedličkovi za pomoc při zpracovávání této práce. Také děkuji PhDr. Ing. Václavu Chvátalovi za poskytnutí mapových plánů židovských hřbitovů v digitální podobě a potřebné dokumentace židovského hřbitova Telice, na jejímž základě bylo možné vytvořit aplikační prostředí pro uživatele databáze.

Prohlašuji, že jsem svou diplomovou práci napsala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 13. 5. 2010

Lenka Reinwartová

podpis: _____

Obsah

1 Úvod do problematiky geografické datové sady židovských hřbitovů	5
2 Doplnění datové sady židovských hřbitovů o atributy vrstvy náhrobků, doplnění struktury datového modelu	6
2.1 Popis získaných dat	6
2.2 Struktura geografické datové sady židovských hřbitovů	7
2.3 Zpracování vstupních dat	8
3 Možnosti využití ArcObjects pro návrh aplikačního rozhraní	16
3.1 Porozumění ArcObjects	16
3.2 Unified Modeling Language	17
3.2.1 Diagramy tříd	17
3.3 Objektově orientované programování	22
3.4 Objektově orientované programování za použití component object model	25
3.5 Příklady využití ArcObjects	27
3.5.1 Práce s vrstvou	28
3.5.2 Práce s geometrií	32
3.6 Další zdroje pro podporu při práci s ArcObjects	33
4 Návrh a implementace aplikačního rozhraní pro práci s datovou sadou židovských hřbitovů	34
4.1 Návrh formuláře pro získání dat z náhrobku	34
4.2 Návrh formuláře pro vyhledávání náhrobků podle zadaných informací	36
4.3 Implementace aplikačního rozhraní	39
4.3.1 Tvorba vlastního tlačítka	39
4.3.2 Implementace formuláře pro získání dat z náhrobku	40
4.3.3 Implementace formuláře pro vyhledávání	44
5 Závěr	50
Adresářová struktura přiloženého DVD:	53

Název práce: Aplikační rozhraní pro geografickou datovou sadu židovských hřbitovů

Autor: Lenka Reinwartová

e-mail autora: reinwart@students.zcu.cz

Katedra (ústav): Katedra matematiky

Vedoucí bakalářské práce: Ing. Karel Jedlička

e-mail vedoucího: smrcek@kma.zcu.cz

Abstrakt: Hlavním cílem práce je rozšíření geografické datové sady židovských hřbitovů o atributová data a tvorba aplikačního rozhraní. Mezi hlavní požadavky na vyhotovení a zpracování aplikačního rozhraní patří možnost prohledávání databáze na základě zadaných atributových údajů a kompletní vizualizace veškeré dokumentace nasbírané ke každému náhrobku, tzn. informace o zemřelých osobách, fotografie, přepis a překlad náhrobku.

Klíčová slova: židovský hřbitov, ArcObject, rozhraní, atributová data, proměnná, geodatabáze, prvková třída, Object Model Diagram, formulář, implementace

Title: Application interface for geographical data set of Jewish cemeteries

Author: Lenka Reinwartová

Autor's e-mail address: reinwart@students.zcu.cz

Department: Department of Mathematics

Supervisor: Ing. Karel Jedlička

Supervisor's e-mail address: smrcek@kma.zcu.cz

Abstract: The main purpose of this thesis is to extend a geographical data set of Jewish cemeteries to attributes and to create the application interface. Amongst main requirements for completion and processing the application interface is the possibility of search the data set for a consideration of given attributes and complete visualization of all the documentation collected for each tombstone, which means informations on dead persons, the photo, transcription and translation of each tombstone.

Keywords: Jewish cemetery, ArcObject, interface, attributes, variable, geodatabase, feature class, Object Model Diagram, form, implementation

1 Úvod do problematiky geografické datové sady židovských hřbitovů

V létě roku 2007 jsem přijala zajímavou pracovní nabídku od Národního památkového ústavu v Plzni – mapování židovských hřbitovů. Na základě nasbíraných dat, zajímavých zkušeností a nových kontaktů jsem se rozhodla pro zpracování bakalářské práce, která spojuje celou řadu aspektů geomatiky, počínaje sběrem dat prováděným v terénu, přes zpracování naměřených dat a tvorbu výstupních mapových plánů, až po práci s geografickými informačními systémy – tvorba geografické databáze, vektorizace, transformace a výsledná vizualizace geodatabáze. V průběhu práce byla navržena a sestavena geografická datová sada 28 židovských hřbitovů, které byly později doplněny o další 3 nově zmapované hřbitovy. Vzniklá geografická databáze byla, za použití kartografických znaků a reprezentačních pravidel, upravena do formy vhodného a přehledného mapového dokumentu. Každý prvek obsažený v databázi je, na základě použití atributů, jednoznačně přiřaditelný ke konkrétnímu židovskému hřbitovu, na kterém se nachází. Jednotlivé kroky včetně stručné historie o židovských hřbitovech jsou popsány v bakalářské práci (viz literatura [1]).

V návaznosti na bakalářskou práci, ve které byla zpracována databázová vrstva, vznikla tato diplomová práce, která se zabývá řešením třívrstvé architektury, tedy tvorbou aplikační a prezentační logiky. Aplikační logika zahrnuje návod na zpracování nasbírané dokumentace na židovských hřbitovech a jejich začlenění do geografické datové sady ve formě atributových dat a tvorbu aplikačního rozhraní, které umožní uživateli jednoduché a rychlé prohledávání databáze na základě zadaných atributových údajů. Část zabývající se prezentační logikou zahrnuje tvorbu grafického uživatelského rozhraní, které umožní uživateli snadnou vizualizaci kompletní dokumentace jednotlivých vybraných náhrobků.

2 Doplnění datové sady židovských hřbitovů o atributy vrstvy náhrobků, doplnění struktury datového modelu

V úvodu práce je doplněna a ucelena datová vrstva databáze židovských hřbitovů. Jedná se o doplnění informací o jednotlivých náhrobcích a zesnulých osobách. Data byla získána z ucelené dokumentace židovských hřbitovů začleněných do geografické datové sady (viz obr. 2.2) vytvořené v rámci bakalářské práce (viz [1]). Dokumentace byla poskytnuta PhDr. Ing. Václavem Chvátalem. Jedná se o fotografie náhrobků, přepis a překlad textu z náhrobků a tabulku atributů o zemřelých osobách. Pro snadné doplňování a jednotné zpracování atributových dat ostatních židovských hřbitovů byl vypracován systematický postup popsáný v rámci této části práce.

2.1 Popis získaných dat

Data dostupná v kompletní dokumentaci židovských hřbitovů obsahují:

- *textovou dokumentaci náhrobků* – jedná se o tzv. D-LISTY, které obsahují malou černobílou fotografii náhrobku, identifikační číslo náhrobku (např. TE-001 – jde o náhrobek z židovského hřbitova Telice, který nese číslo jedna), jméno člověka, který je na tomto místě pohřben, datum úmrtí, přepis a překlad textu čitelného z náhrobku a případné komentáře. Textová dokumentace náhrobků je v prvotní formě tvořena ručně, do digitální formy je převedena naskenováním do formátu PNG.
- *fotodokumentaci náhrobků* – jedná se o barevnou fotografii každého náhrobku spolu s cedulkou nesoucí číselné označení daného náhrobku. Složka, ve které je tato dokumentace zařazena nese název NAHROBKY. Fotografie náhrobků jsou rovněž ve formátu PNG.
- *tabulku atributů* (viz obr. 2.1 – data uvedená v tomto obrázku jsou pouze ilustrační) – tabulka je zařazena v adresářové složce REGISTR. Obsahuje následující atributová data o zesnulém: hebrejské jméno, hebrejské jméno otce, občanské jméno, občanské příjmení, datum úmrtí a číslo náhrobku. Jednotlivé znaky a symboly použité v registru jako zkratka nebo např. pro snazší orientaci jsou vysvětleny v textovém dokumentu, který je v adresářové struktuře obdržených dat umístěn ve složce VYSVETL. Pro přehlednost zde uvádím výpis a vysvětlení konkrétních znaků (viz tab. 2.1).

Pozn.: Ukázka kompletní dokumentace je dostupná na přiloženém DVD.

použitá symbolika	vysvětlení
>	„ha-mechune“ = zvaný, řečený (následuje další jméno, přezdívka apod.)
KC	„kohen cedek“ = kněz ctnosti (příslušník kmene Kohenů ¹)
SGL	„sagen Levijim“ = zástupce Levitů (příslušník kmene Levi)
HL	„ha-Levi“ = příslušník kmene Levi
HK	„ha-kohen“ = příslušník kmene Kohenů
~	„eset“ = manželka (následuje jm. manžela, vyskytuje se u vdaných žen)
bez prefixu	jméno otce (je-li známo i příjmení otce, uvádí se v závorce)

Tab. 2.1: Tabulka znaků použitých v registru

Pozn.: Poslední dva symboly se vyskytují v atributovém sloupci „hebrejské jméno otce“. V úvahu připadá i výskyt obou výše uvedených výrazů - jména manžela i jména otce. Výsledný zápis v atributové tabulce vypadá např. takto:

"...|Sara | Josef (Glaser)~Avraham|...".

Ve skutečnosti lze takto zapsané atributy přečíst následujícím způsobem:

„Sara eset Avraham bat Josef Glaser“, což převeďeno do češtiny znamená:

„Sara, manželka Avrahamova, dcera Josefa Glasera“.

JMENO HEBR.	BEN/BAT ~ESET (PRIJM.OTCE)	JMENO OBCAN.	PRIJMENI	DATUM	CN
Bünla	Avraham		Weil	1757-11-25	te001
Nechuma	~ Becalel			1765/1766	te002
Esterl	MeirSGL ~ Jechz.			1781-10-20	te003
Jehuda	Jaakov			1794-03-02	te004
Eicik	Mose			1760-12-10	te005

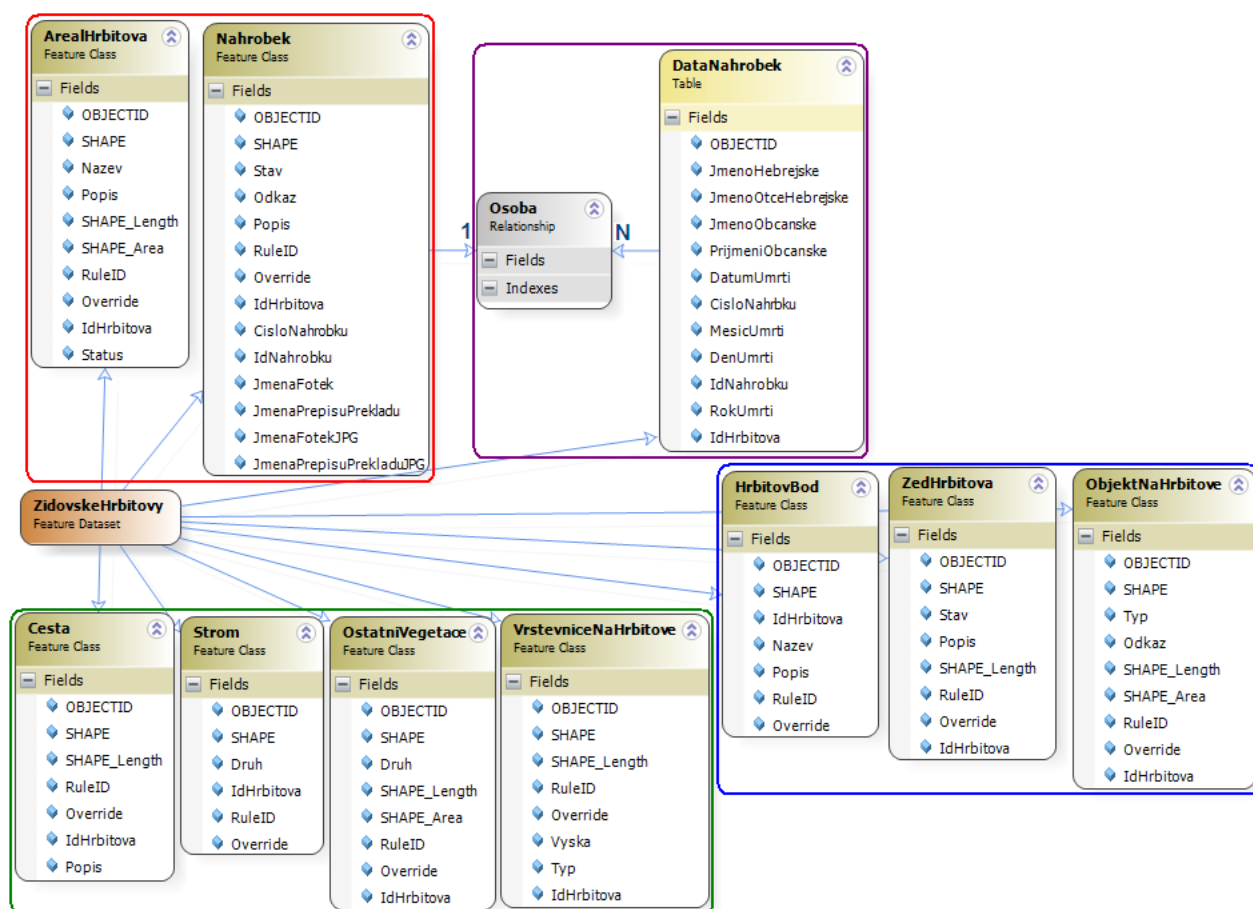
Obr. 2.1: Ukázka tabulky atributů (REGISTRU)

2.2 Struktura geografické datové sady židovských hřbitovů

Jednotlivé třídy databáze židovských hřbitovů jsou hierarchicky rozděleny do čtyř skupin (viz obr. 2.2) podle stupně důležitosti a využitelnosti:

¹ "Kohanité jsou potomci Aronovi; kněží, kteří měli za úkol provádění posvátných rituálů v jeruzalémském Chrámu (= hlavní svatostánek ve starověkém Izraeli). Funkce kohena se datuje od obdržení Desatera a přehled povinností je uveden v 3. knize Mojžíšově, kde je kohenům výslovně zakázán přístup k mrtvým a ke hrobům, s výjimkou hrobů některých příbuzných. Příslušnost ke kmeni se dědí po otci – kohenem (resp. levitou) je každý, jehož otec byl kohen (resp. levita). Z toho vyplývá, že knězem se člověk nemůže stát, musí se jím už narodit. A levité jsou potomci Leviho (jednoho z Jakobových synů) jsou kněžským rodem, stejně jako kohanité, ale jde o „nižší“ kněží, kteří kohenům při bohoslužbě pomáhali." [1]

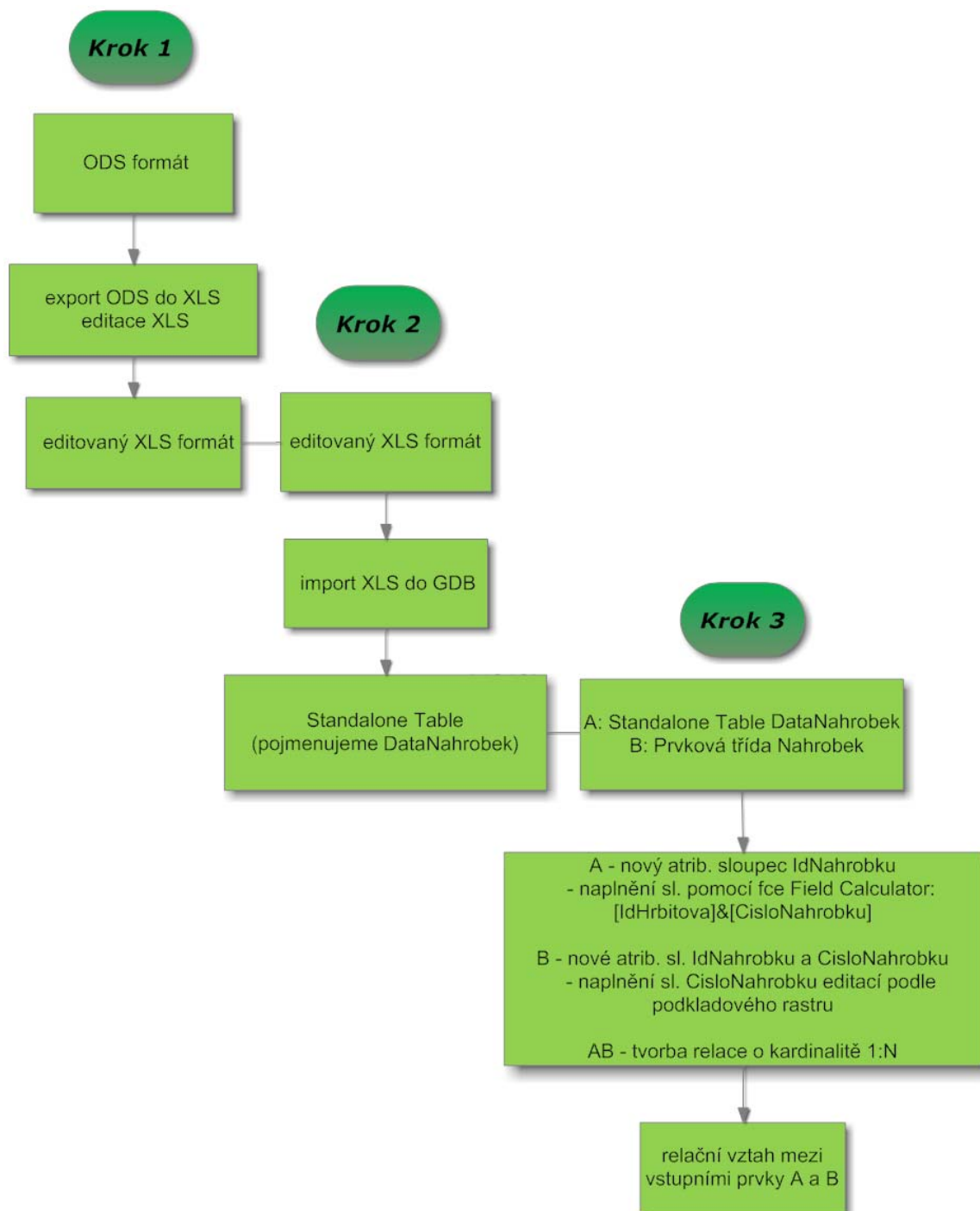
- **základní geometrické prvky** – pro přímý import do PaGIS, databáze spravované Národním památkovým ústavem (skupina označená červeným obdélníkem),
- **zpřesňující údaje** – doplňující informace ke konkrétnímu židovskému hřbitovu (skupina označená modrým obdélníkem),
- **topografický podklad** – topografie napomáhající v orientaci na židovských hřbitovech (skupina označená zeleným obdélníkem),
- **nově doplněná data** – atributová tabulka registru židovských hřbitovů a relace o kardinalitě 1:N, provazující atributovou tabulku s třídou prvků *Nahrobek*, tato data byla k databázi doplněna v rámci této práce (skupina označená fialovým obdélníkem).



Obr. 2.2: Struktura databáze židovských hřbitovů

2.3 Zpracování vstupních dat

Zpracování vstupních dat je možné rozdělit na několik po sobě jdoucích kroků (viz obr. 2.3).



Obr. 2.3: Schéma postupu předzpracování vstupních dat

KROK 1

- **Vstup:** Vstupním formátem pro první krok byl datový formát ODS, který je obsažen v původní dokumentaci získané od PhDr. Ing Václava Chvátala.

- **Operace:** Po importu formátu ODS do geodatabáze² nebylo dodrženo kódování přehlásek, které nebyly zobrazovány správně. Proto bylo nutné před importem dat do geodatabáze provést export registru atributů z formátu ODS do formátu, který přehlásky zachovává v původním stavu. Po několika pokusech byl zvolen formát XLS. Kódování používané pro datové formáty XLS zachovává po importu do geodatabáze všechny symboly, které se v textu registru vyskytují.

Po exportu dat z ODS formátu do XLS bylo nutné přistoupit ještě k editaci XLS tabulky. První řádek byl upraven tak, aby použitý text souhlasil s názvy atributových sloupců. Druhý až n-tý řádek odpovídá atributům daných atributových sloupců. Nakonec je vhodné odstranit zbylé listy XLS souboru, případně přejmenovat list, který zobrazuje atributová data. Tento postup později napomáhá v orientaci v XLS souboru při importu do atributové tabulky databáze židovských hřbitovů.

V této části práce bylo důležité rozhodnout, jak upravit data v registru vzhledem k jejich výslednému použití, které by mělo směřovat ke klientské aplikaci. Proto byl upraven atributový sloupec DatumÚmrti, který je v původních datech definován ve formátu [rok-měsíc-den]. Jelikož by tento atributový sloupec ve výsledné aplikaci měl uživateli sloužit pro vyhledávání podle jednotlivých parametrů (nejčastěji podle atributu rok), bylo vhodné jej rozdělit. To se provádí následujícím způsobem:

- nejdříve je označen sloupec, se kterým chceme pracovat,
- v záložce *Data – Datové nástroje* je spuštěn příkaz *Text do sloupců*, otevře se Průvodce převodem textu do sloupců, díky kterému je možné intuitivně projít celým procesem,
- poté je důležité zvolit typ oddělovače, v tomto případě "-" (pomlčku),
- závěrem je volen cíl, kam budou nově diferencované sloupce umístěny.

Problém nastal pouze v případech, kde není datum úmrtí kompletní a je zaznamenán pouze přelom mezi dvěma roky, např. 1831/1832. Tento formát zůstal po rozdělení stejný. Protože pro vyhledávací formulář byl nakonec použit pouze rok úmrtí (viz kap. 4.2), byl tento speciální případ atributové hodnoty ponechán v původním stavu a zachován tak i pro použití ve vyhledávacím formuláři.

- **Výstup:** Po provedení kroku 1 získáme editovaný datový formát XLS, který je připraven pro import do geodatabáze.

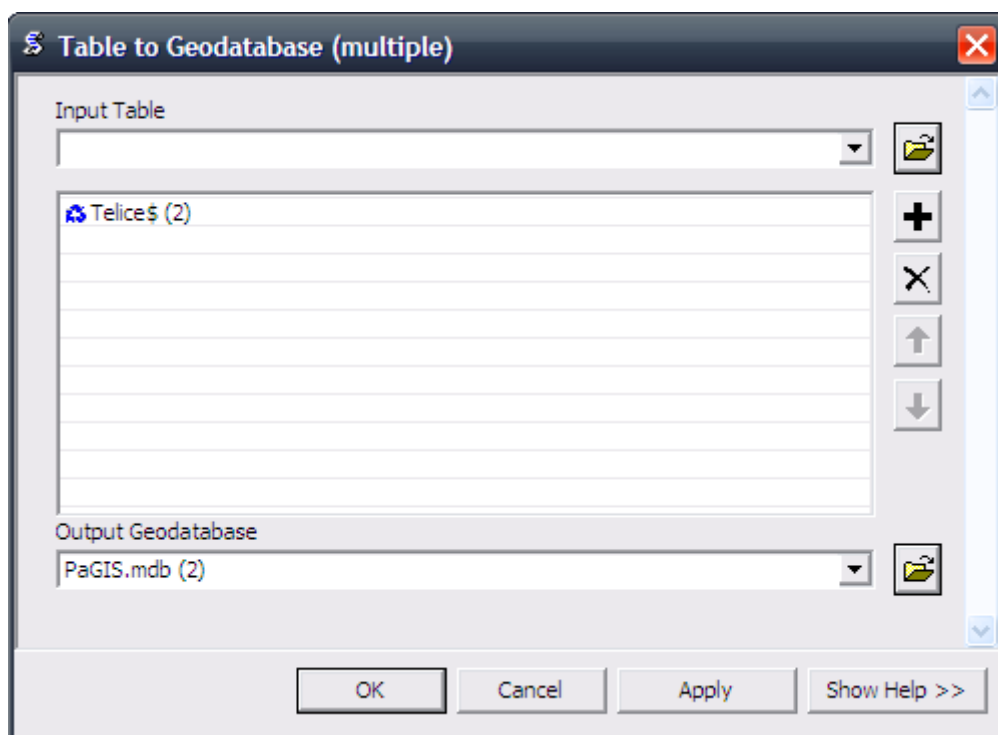
Výše uvedený postup byl proveden ještě před exportem tabulky do geodatabáze, a to v prostředí aplikace Excel sady Microsoft Office 2007.

Krok 2

- **Vstup:** Vstupní prvkem v tomto kroku je shodný s výstupním prvkem kroku 1, jde o editovaný XLS datový formát obsahující konkrétní atributy náhrobků.

² Geodatabáze – je soubor geografických datových sad různých typů používaný v ArcGIS.

- **Operace:** Dalším krokem v této části je import XLS souboru do atributové tabulky začleněné do stávající geodatabáze židovských hřbitovů. V ArcToolbox již existuje předdefinovaná funkce, která tento import umožňuje. Jde o skript s názvem *Table to Geodatabase (multiple)*. Po otevření tohoto skriptu je nutné zadat pouze vstupní tabulku a výstupní geodatabázi, do které se vzniklá atributová tabulka uloží (viz obr. 2.4). Jak je z obr. 2.4 patrné, tento skript umožňuje import více tabulek současně. Jelikož jsem pro vytvoření systematického postupu získala pouze data k židovskému hřbitovu Telice, je v tomto případě zvolena pouze jedna vstupní tabulka. Výstupní adresář odpovídá umístění databáze židovských hřbitovů, do které chceme danou tabulku importovat.
- **Výstup:** Po provedení potřebných operací získáme samostatnou atributovou tabulku (tzv. *Standalone Table*), která je ale součástí geodatabáze. Pro přehlednost je pojmenována *DataNahrobky*.



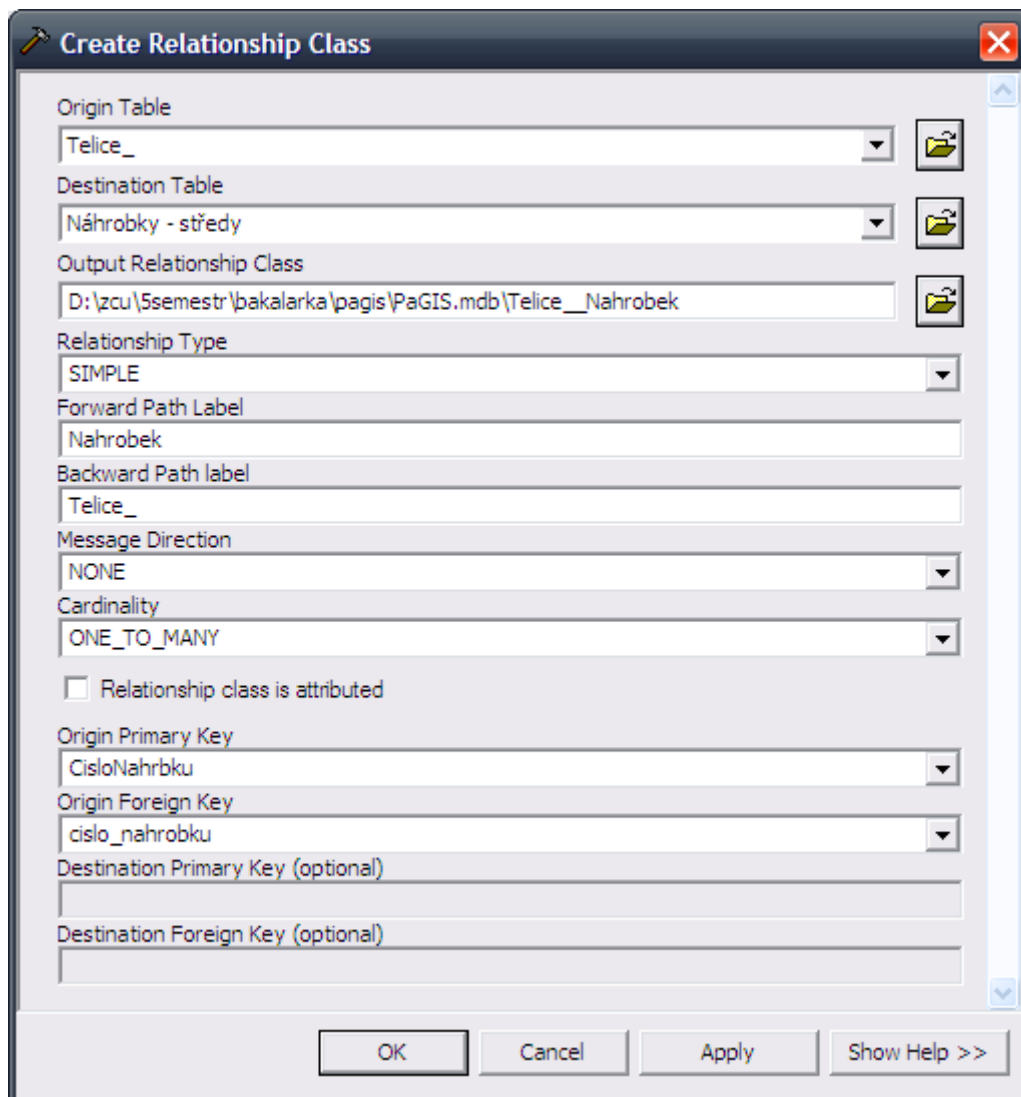
Obr. 2.4: *Table To Geodatabase (multiple)*

Krok 3

- **Vstup:** Vstupní prvky v kroku 3 jsou tentokrát dva. První je opět shodný s výstupním prvkem kroku 2, jde o *Standalone Table* pojmenovanou *DataNahrobky*. Druhým vstupním prvkem je prvková třída *Nahrobek*.
- **Operace:** Dalším krokem po provedeném importu je dokončení editace registru (atributové tabulky *DataNahrobky*), především přidání nového atributového sloupce pojmenovaného *IdNahrobku*. Tento atributový sloupec sestává ze složenin označujících zkratku konkrétního hřbitova a číslo daného náhrobku a byl vytvořen pomocí funkce *FieldCalculator* zápisem *[IdHrbitova]&[CisloNahrobku]*, v případě židovského hřbitova Telice může být výsledkem např. te244.

Stejným postupem byl vytvořen a stejně pojmenován i atributový sloupec v prvkové třídě *Nahrobek*. Tyto atributové sloupce budou dále sloužit pro vytvoření relace (viz obr. 2.5). Před vytvořením relace je však nutné editovat třídu prvků *Nahrobek* tak, že každému z náhrobků přidáme číslo náhrobku na základě podkladového rastru, který je již georeferencován. Editovaná čísla náhrobků jsou ukládána do atributového sloupce *CisloNahrobku*. Tento atributový sloupec obsahuje vzhledem k atributovému sloupci *IdNahrobku* duplicitní data, ale vzhledem k tomu, že je nejprve nutné všechny náhrobky očíslovat a teprve na základě čísla a *IdHrbítova* je naplňován atributový sloupec *IdNahrobku*, doporučuji pro usnadnění práce atributový sloupec *CisloNahrobku* zachovat.

Dále je nutné provázat vloženou atributovou tabulku s konkrétním prvkem v geodatabázi, v tomto případě s jednotlivými náhrobky židovských hřbitovů. Toto provázání je realizováno pomocí relace fungující na základě atributů *IdNahrobku*. Pro vytvoření relace mezi dvěma složkami databáze je nutná identita proměnných atributových sloupců, na jejichž základě je relace vytvářena. V tomto případě mají oba výše zmíněné atributové sloupce proměnnou *Text*.



Obr. 2.5: *Create Relationship Class*

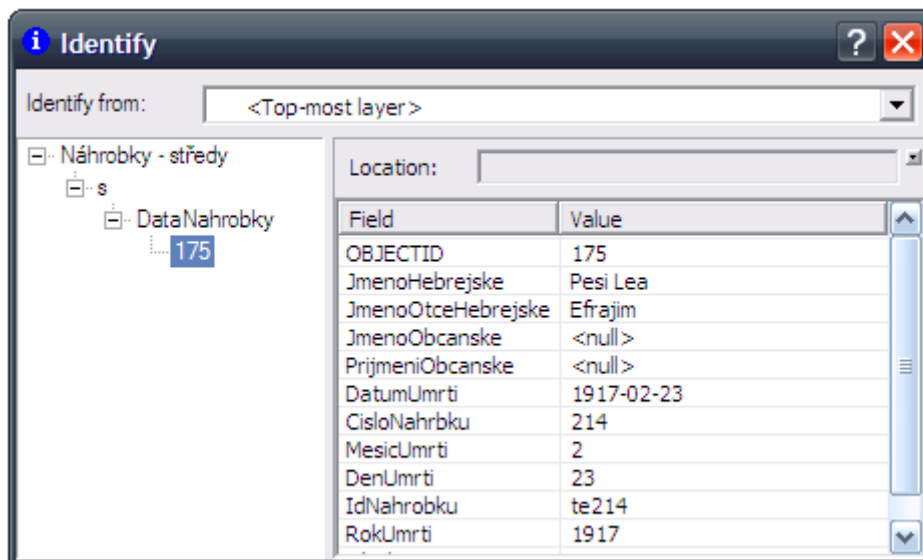
Při vytváření relace je nutné nastavit vazební kardinalitu 1:N (ONE_TO_MANY, viz obr. 2.5), protože k jednomu geometricky definovanému náhrobku (bodu) může být vztaženo větší množství entit (neboli jeden náhrobek může nést informace o dvou a více osobách - zpravidla se jedná o dvě osoby – společně pohřbené manžele). Také se zde však vyskytuje množství případů, kdy se k jednomu geometricky definovanému náhrobku nevztahují žádné entity. Tato situace nastává v případech, kdy je náhrobek natolik poškozen (ať už erozí či vlivy člověka), že není možné nápis přečíst. Dále může tato situace nastat, leží-li náhrobek nápisem k zemi a je-li tak masivní, že nebylo možné ho zvednout. Při vytváření relace nesmíme zapomenout na nastavení primárního a cizího klíče, kterými jsou v tomto případě výše zmíněné atributové sloupce označující číslo náhrobku a identifikátor (zkratku) hřbitova, na kterém se náhrobek nachází.

- **Výstup:** Výstupem kroku 3 je relační vztah mezi dvěma vstupními prvky: *Standalone Table DataNahrobky* a prvkovou třídou *Nahrobek*.

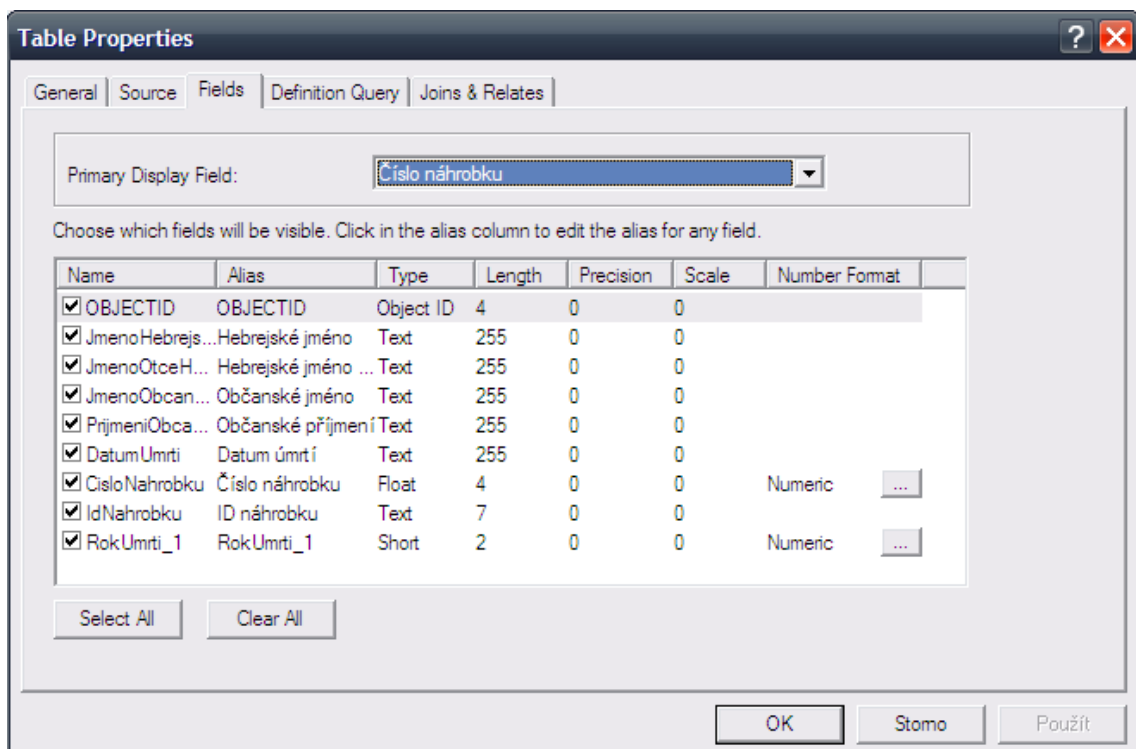
O tom, zda byla relace úspěšně vytvořena, se můžeme přesvědčit pomocí funkce *Identify* (viz obr. 2.6). Po poklepnání na některý ze symbolů označující náhrobek se objeví tabulka *Identify*, ve které jsou vypsány všechny atributy daného prvku. Abychom ale zjistili, zda byla relace opravdu uskutečněna, musíme rozbalit další nabídku přes zobrazenou atributovou hodnotu. V našem případě se jedná o písmeno *s*, tedy konkrétní hodnotu z atributového sloupce *Stav*.

Pozn: Zde se vždy zobrazuje hodnota takového atributového sloupce, který je ve vlastnostech dané třídy prvků zvolen jako primárně zobrazovaný (Primary Display Field).

Objeví se název atributové tabulky, se kterou je tato prvková třída spojena relací (zde *DataNahrobky*) a po jejím rozbalení vidíme hebrejské jméno člověka, který je pod daným náhrobkem pohřben (opět hodnota sloupce, který je pro atributovou tabulku nastaven jako primárně zobrazovaný). Toto nastavení je možné libovolně měnit, nejvhodnější by zřejmě bylo zvolit jako primární pole např. číslo náhrobku (viz obr. 2.7).



Obr. 2.6: *Identify*



Obr. 2.7: Nastavení primárně zobrazovaného sloupce

V průběhu zpracování vstupních dat se operuje hlavně s atributovými daty registru daného židovského hřbitova, tudíž dochází k jednotným zpracovatelským a editačním procesům nad větším množstvím dat, což je časově velmi pohodlné. Přesto se při zpracovávání vstupních dat setkáváme s jednou časově náročnou procedurou, kterou je naplňování atributového sloupce třídy prvků *Nahrobek* označující číslo náhrobku - tuto práci je bohužel nutné provádět ruční editací nad georeferencovanými rastry.

3 Možnosti využití ArcObjects pro návrh aplikačního rozhraní

V rámci tvorby aplikačního rozhraní je nutné provést dotazování, na jehož základě budou získány hledané objekty geodatabáze. Toto dotazování se skládá z více kroků. Aby byla uživateli umožněna jednoduchá práce s geodatabází, bylo nutné dotazování z pohledu uživatele zjednodušit, tedy vytvořit vlastní nástroj, a to prostřednictvím programování s ArcObjects.

Literatura, která popisuje, jak je možné využívat ArcObjects, je z velké části dostupná pouze v anglickém jazyce. Proto byla vytvořena tato část práce, která popisuje jak ArcObjects fungují a jak je možné je používat.

3.1 Porozumění ArcObjects

ArcObjects jsou podle [3] softwarové komponenty, které poskytují služby pro podporu GIS aplikací. Programátoři v ESRI používají k naprogramování těchto tříd programovací jazyk C++. Kód, který obsahují třídy, které tvoří ArcObjects není přístupný uživatelům, ale díky dodržení COM standardů (viz 3.2.4) je možné je používat i v jiných programovacích jazycích, jako je VBA³ (tento programovací jazyk je součástí aplikací ArcMap a ArcCatalog, kde slouží pro práci s ArcObjects), VB⁴, VB.NET⁵, Java⁶, C++⁷, Delphi⁸ a dalších.

Třídy jsou uloženy v dynamicky linkovaných knihovnách (DLL⁹) ve složce bin na vašem počítači. Jedná se o dynamicky přiřazované knihovní soubory. Třídy existují jako kódy v souborech. Ze tříd tvoříme objekty. Například z třídy vrstva (*Layer*) můžeme vytvořit nespočetné množství objektů vrstva (*Layer*).

³ VBA – Visual Basic for Applications je implementací programovacího jazyka Visual Basic 6.

⁴ VB – Visual Basic je programovací jazyk vytvořený společností Microsoft, který pracuje v operačním prostředí Windows.

⁵ VB.NET – Visual Basic .NET je nová generace jazyka Visual Basic postavená na platformě .NET Framework. Jedná se o moderní objektově orientovaný jazyk, který se neustále vyvíjí a který má velmi širokou základnu vývojářů po celém světě.

⁶ Java – objektově orientovaný programovací jazyk, který vyvinula firma Sun Microsystems.

⁷ C++ je objektově orientovaný programovací jazyk, který vyvinul Bjarné Stroustrup a další v Bellových laboratořích AT&Trozšířením jazyka C.

⁸ Delphi je integrované grafické vývojové prostředí firmy Borland určené pro tvorbu aplikací na platformě MS Windows v jazyce Object Pascal (objektové nástavbě Pascal).

⁹ DLL – Dynamic-link library je tzv. dynamická knihovna. Tato knihovna obsahuje takové části kódů, které mohou být sdíleny více programy. Knihovna udržuje kód v binární podobě.

Pro pochopení ArcObjects je nutné porozumět UML¹⁰ pro snadnou orientaci v diagramech tříd, OOP¹¹ pro pochopení práce s objekty a OOP za použití COM pro možnost použití jiných kódů.

V ArcObjects je možné se orientovat dvěma způsoby:

- prostřednictvím modelů tříd ArcObjects nebo
- za pomoci ArcGIS Desktop Resource Center a online vyhledávače.

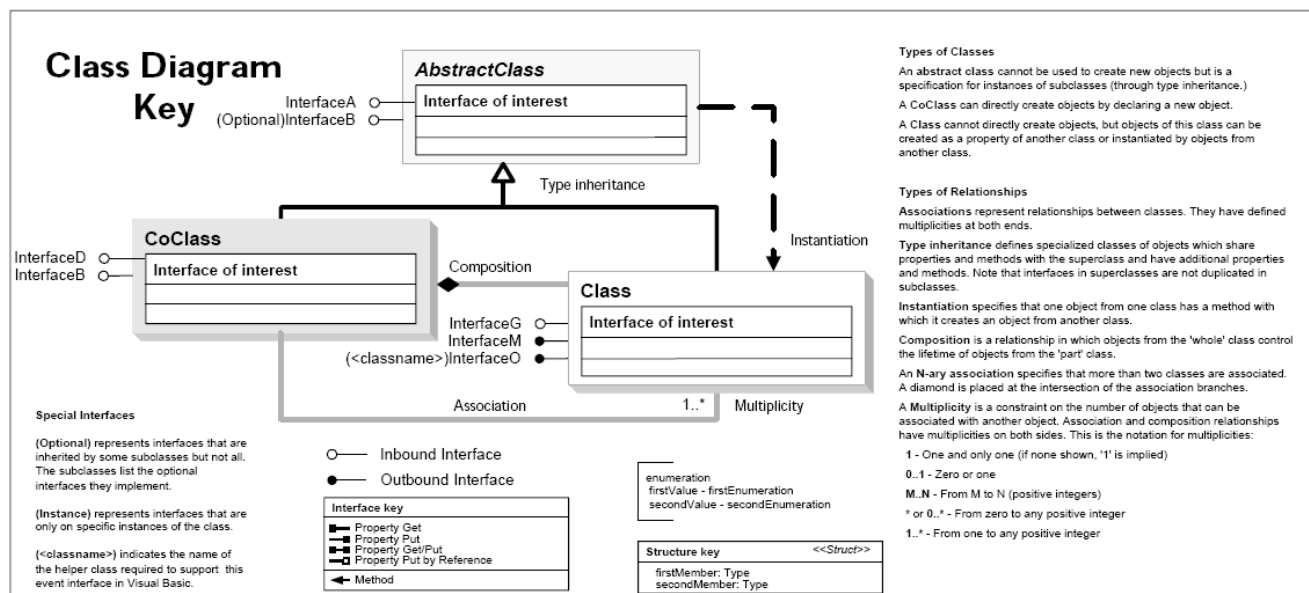
Závěrem této kapitoly budou uvedeny jednoduché obecné ukázky pro tvorbu kódu.

3.2 Unified Modeling Language

Jak již bylo řečeno, pro pochopení ArcObjects je nutné rozumět UML.

3.2.1 Diagramy tříd

Modelové diagramy objektů můžeme připodobnit k mapám GIS. Podobně jako při prozkoumání nového města používáme mapu, stejně tak při pochopení toho, jak jsou organizované ArcObjects použijeme diagramy tříd objektů. Pro pochopení použitých symbolů využijeme legendu (viz obr. 3.1), stejně jako na mapách. Modelové diagramy jsou organizované podle témat pro různé skupiny ArcObjects, např. ArcMap, Display, Geometry, atd.



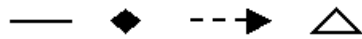
Obr. 3.1: *Legenda Object Model Diagram*

¹⁰ UML – Unified Modeling Language je grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů.

¹¹ OOP – Object-oriented Programming je metodika vývoje softwaru.

V diagramech jsou použity UML symboly. UML je v softwarovém inženýrství grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů. UML je tvořeno mnoha symboly, ale nám jich zatím pro pochopení struktury UML a pro orientaci v ObjectModelDiagram postačí pouze několik (viz obr. 3.2).

- **vzájemné vztahy**



- **třídy**



- **vlastnosti a metody**



Obr. 3.2: 12 základních symbolů použitých v Object Model Diagram

Object Model Diagrams naleznete např. na těchto stránkách:

URL:< http://resources.esri.com/help/9.3/arcgisdesktop/com/vba_start.htm>.

Pozn.: Například do rozhraní IMxApplication se dostaneme přes: ArcObjects library references – ArcMapUI – Interfaces – IM – ImxApplication Interface.

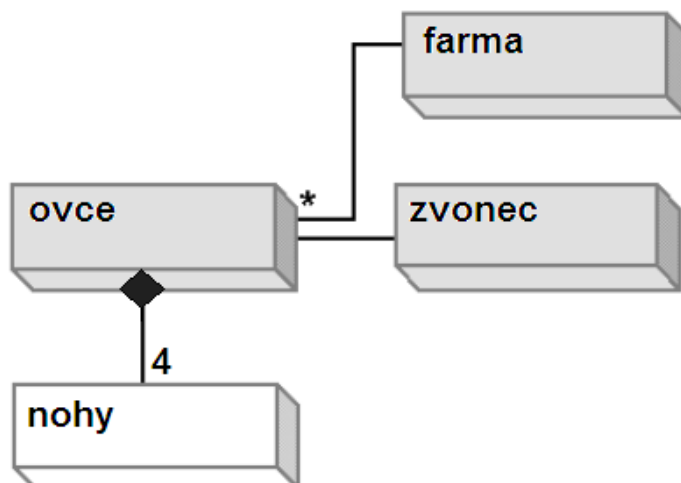
V následující části si popíšeme význam jednotlivých UML symbolů:

- **Asociace** – vyjadřuje vztah dvou objektů, např. jedna *ovce* má jeden *zvonec*.



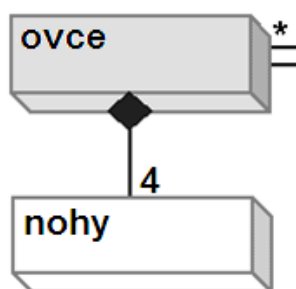
Obr. 3.3: Ukázka asociace

- **Násobnost** – je to vlastností asociace, vyjadřuje vztah o určité kardinalitě (1:1, 1:M nebo N:M), např. na jedné *farmě* je mnoho *ovcí*, může ale vyjadřovat i konkrétní číslo, např. jedna *ovce* má *4 nohy*. Násobnosti je možné označovat v UML mnoha různými způsoby.



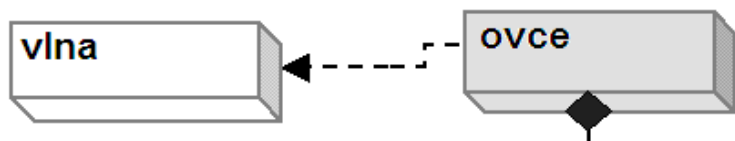
Obr. 3.4: Ukázka násobnosti

- **Kompozice** – vyjadřuje, který objekt se skládá z kterých, např. objekt *ovce* (u kterého je znázorněn černý diamant) je složen z objektů *nohy*, podle kardinality jsou objektu *ovce* přiřazeny 4 objekty *nohy*. Pokud bude objekt *ovce* smazán, bude automaticky odstraněn i objekt *nohy*.



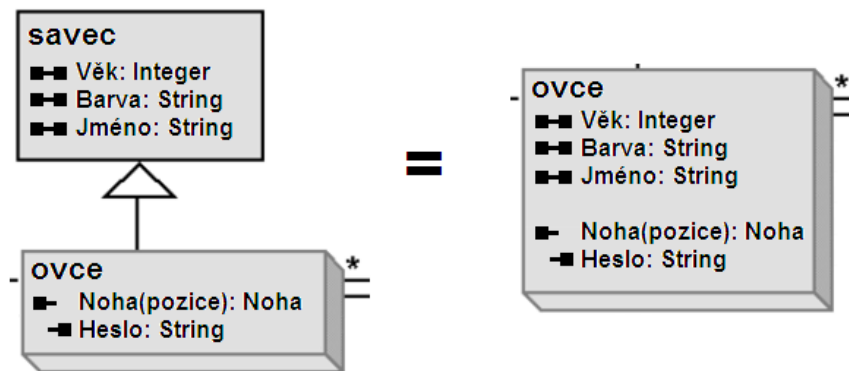
Obr. 3.5: Ukázka kompozice

- **Tvoří** – jeden typ objektu tvoří jiný typ objektu, např. *ovce* tvoří *vlnu*.



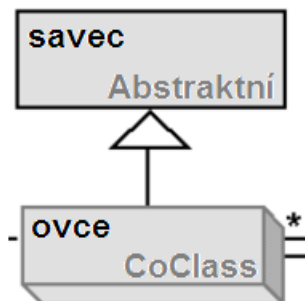
Obr. 3.6: Ukázka „tvoří“

- **Je typem** – tento symbol se čte „ode dna“, začneme objektem pod šipkou, pokračujeme přes šipku a skončíme objektem, ke kterému šipka směřuje, např. *ovce* je typem *savec*. Symbol trojúhelníku zároveň znázorňuje dědičnost. Vlastnosti, které má objekt *savec* jsou děděny objektem *ovce* (*savec* má srst, jako mládě saje mléko, atd. *Ovce* má tyto vlastnosti také). Díky zjednodušení symboliky se využívá generalizace a zpětně pak specializace.



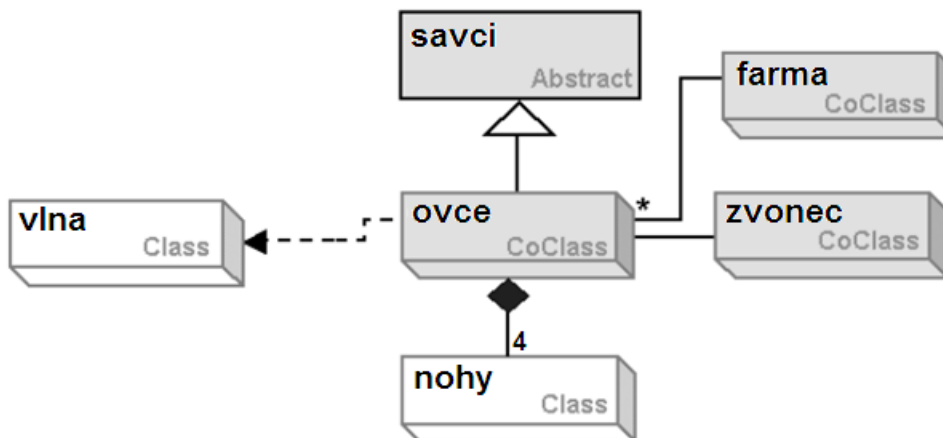
Obr. 3.7: Ukázka „je typem“ a ukázka generalizace

- **Abstraktní třída** – je zobrazena ve 2D tvaru a má šedou výplň. Nemohou z ní být tvořeny objekty. Používá se pro sjednocení a dědění vlastností. Uchovává metody a vlastnosti, které její podtřídy dědí.



Obr. 3.8: Ukázka abstraktní třídy

- **CoClass** – je to implementace různých tříd, která je používána klienty pro tvorbu vlastních objektů. CoClass je zobrazena ve 3D tvaru a má šedou výplň. Z CoClass je možné vytvořit objekty pomocí dvou řádků kódu.
- **Třída** – je zobrazena ve 3D tvaru a nemá výplň. Třídy mohou být tvořeny nebo vráceny pouze objekty jiných tříd. Nejprve je nutné vytvořit jiné objekty, aby bylo možné pracovat s třídami. Ze vztahů na obrázku je patrné, že CoClass *ovce* vytváří objekty pro třídu *vlna*. Pokud chceme vytvořit *vlnu*, musíme použít CoClass *ovce*, odkud je možné *vlnu* vytvořit.



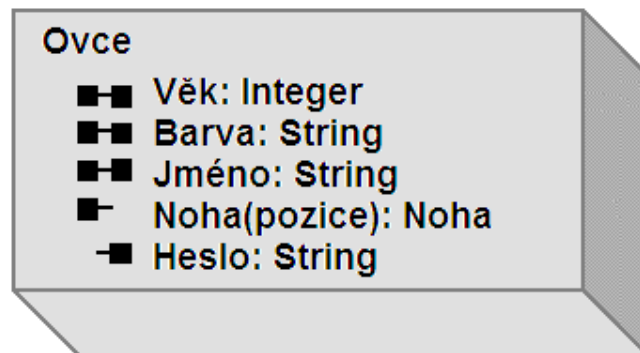
Obr. 3.9: Ukázka třídy a CoClass

- **Vlastnosti objektu čtení a zapisování** – jsou znázorněny symbolem „činek“, který se nachází na levé straně od názvu vlastnosti. Vlastnosti uchovávají atributy objektů a je možné je získat, tedy číst (*get*) nebo nastavit, tedy zapisovat (*set*).

Některé vlastnosti jsou znázorněny směrem ven z objektu, většinou tzv. „levostrannou činkou“, to znamená, že tyto vlastnosti je možné jen získat, ale nemůžeme měnit jejich hodnotu neboli nastavovat je, tzv. *read-only* vlastnosti.

Některé vlastnosti jsou znázorněny naopak směrem dovnitř do objektu, neboli tzv. „pravostrannou činkou“, to znamená, že můžeme jejich hodnotu nastavit, ale nemůžeme ji získat neboli přečíst, tzv. *write-only* vlastnosti, např. heslo používané při zabezpečovacích procedurách.

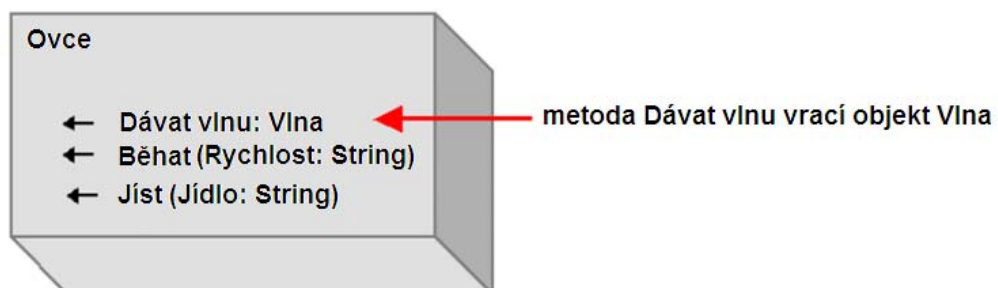
Na pravé straně od názvu vlastnosti je dvojtečka a typ dané vlastnosti. Každá vlastnost si uchovává svoji hodnotu a každá hodnota má daný typ, jako např. Number, String, Date, Object,...Například vlastnost *Věk* je typu Integer (celočíselná hodnota) a vlastnost *Noha* (*pozice*) je typu Objekt, konkrétně objekt nazvaný *Noha*,



Obr. 3.10: Ukázka vlastností „čtení“ a „zápis“

- **Metody** – třídy mají také metody, které jsou znázorněny symbolem „šipky“. Jsou to vlastně akce, které může objekt vykonat. Metody jsou někdy nazývané chováním.

Některé metody mohou vracet hodnoty, jiné ne. Vracené hodnoty jsou hodnotami určitého typu, který je opět znázorněn za dvojtečkou na pravé straně názvu metody. Například metoda *Dávat vlnu* vrací hodnotu typu objekt s názvem *Vlna*. Naopak metoda *Běhat* nevrací podle obrázku žádnou hodnotu.



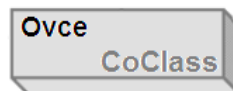
Obr. 3.11: Ukázka metody

3.3 Objektově orientované programování

- **Kód pro tvorbu objektu**

Jak již bylo uvedeno výše, pro vytvoření objektu stačí pouhé dva řádky kódu.

```
Dim MojeOvce As Ovce
Set MojeOvce = New Ovce
```

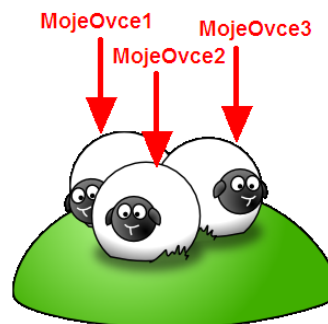


Obr. 3.12: Deklarace a instantizace proměnné *MojeOvce*

První řádek deklaruje proměnnou. Název proměnné můžeme libovolně zvolit, zde je použit název *MojeOvce*. Jako klíčové slovo se používá *As* a následuje už jen jméno *CoClass*, zde *Ovce*.

Druhý řádek nastavuje deklarovanou proměnnou. Jako klíčové slovo se používá *Set*, následuje název proměnné, znaménko rovnosti a nastavení proměnné. Zde je proměnná nastavena pomocí klíčového slova *New* na nový objekt *CoClass Ovce*. Tento objekt je vytvořen v paměti a proměnná nás nyní odkazuje na tento objekt. Můžeme deklarovat tolik proměnných, kolik bude zapotřebí. Každý objekt tak bude mít objektovou proměnnou.

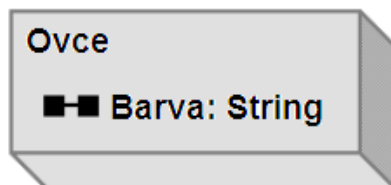
```
Dim MojeOvce1 As Ovce
Set MojeOvce1 = New Ovce
Dim MojeOvce2 As Ovce
Set MojeOvce2 = New Ovce
Dim MojeOvce3 As Ovce
Set MojeOvce3 = New Ovce
```



Obr. 3.13: Deklarace a instantizace více proměnných

- **Kód pro získání vlastnosti**

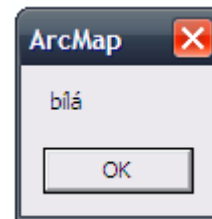
Nejprve je nutné deklarovat proměnnou, která bude téhož datového typu jaký má vlastnost, kterou chceme získat.



Obr. 3.14: Ukázka třídy *Ovce* s vlastností *Barva*

Máme proměnnou *MojeOvce*, která nás odkazuje na objekt *Ovce*. Tento objekt má vlastnost *Barva*, která je datového typu *String*. Deklarujeme novou proměnnou *BarvaOvce*, která je také datového typu *String* a použijeme ji k získání vlastnosti *Barva*.

```
Dim BarvaOvce As String
BarvaOvce = MojeOvce.Barva
MsgBox BarvaOvce
```



Obr. 3.15: Ukázka kódu pro získání vlastnosti

Pokud se chceme přesvědčit, zda se nám skutečně podařilo zjistit, jakou barvu má objekt *MojeOvce*, můžeme použít jednoduché vypsání vlastnosti pomocí funkce *MsgBox* (Message Box).

Pozn.: Pro přístup k vlastnostem a metodám objektu používáme tzv. tečkovou notaci typu: <objekt>.<vlastnost> nebo <objekt>.<metoda>.

- **Kód pro nastavení vlastnosti**

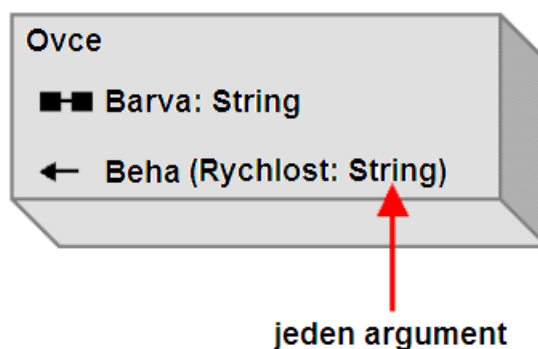
V případě, že chceme nastavit vlastnost objektu, použijeme výše uvedenou tečkovou notaci. Je nutné si být jistý tím, že jsme při nastavování nové vlastnosti použili odpovídající datový typ. Například pokud chceme podle výše uvedeného příkladu nastavit *Barvu* objektu *Ovce* na šedou, použijeme následující kód.



Obr. 3.16: Ukázka kódu pro nastavení vlastnosti

- **Kód pro spuštění metody**

Používáme opět tečkovou notaci, podobně jako u vlastností. Některé metody mohou obsahovat seznam argumentů. Argumenty mají určitý datový typ, stejně jako vlastnosti.



Obr. 3.17: Ukázka třídy *Ovce* s vlastností a metodou

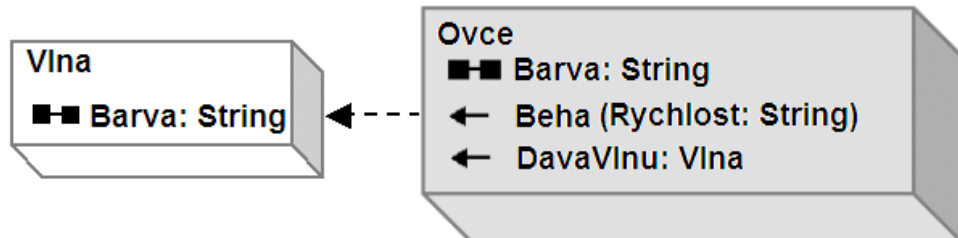
Např. u metody *Beha* je uveden pouze jeden argument *Rychlost*, který je typu *String*. Chceme-li spustit tuto metodu, použijeme následující kód:

```
MojeOvce.Beha "Pomalů"
```

Pokud by bylo u metody *Behá* uvedeno více argumentů, pokračoval by kód čárkou a uvedením hodnoty dalšího argumentu.

- **Kód pro spuštění metody, která vrací hodnotu**

Nejprve musíme deklarovat proměnnou, která bude stejného typu, jako hodnota, kterou chceme získat. Poté nastavíme proměnnou za použití tečkové notace.



Obr. 3.18: Ukázka tříd *Ovce* a *Vlna*

Např. podle výše uvedeného obrázku nejdříve deklaruujeme novou proměnnou *MojeVlna* jako ukazatel na objekt *Vlna*, a poté ji nastavíme tak, aby se do ní uložila vrácená hodnota metody *DavaVlnu*:

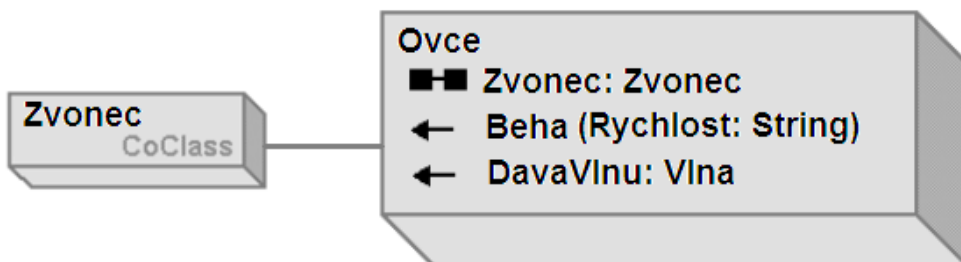
```
Dim MojeVlna As Vlna
Set MojeVlna = MojeOvce.DavaVlnu
```

Poté můžeme ještě např. získat barvu vytvořené *Vlny* a vypsát ji díky *MsgBox* pomocí následujícího kódu:

```
MsgBox MojeVlna.Barva
```

- **Kód pro získávání přístupu k dalšímu objektu**

Sousední objekt je spojen přímkou, která vyjadřuje asociační vazbu o kardinalitě 1:1. Opět pomocí dvou řádků kódu můžeme získat proměnnou ukazující na sousední objekt.



Obr. 3.19: Ukázka třídy *Ovce* a *Zvonec*

Podle výše uvedeného obrázku bude kód vypadat následujícím způsobem:

```
Dim MujZvonec As Zvonec
Set MujZvonec = Ovce.Zvonec
```


Objekt *Ovce* má vlastnost *Zvonec*. Po jejím nastavení (viz výše) získáváme do proměnné *MujZvonec* sousední objekt *Zvonec*. Takže můžeme konstatovat, že nyní máme dvě proměnné:

- proměnnou *MojeOvce*, která ukazuje na objekt *Ovce*,
- proměnnou *MujZvonec*, která ukazuje na objekt *Zvonec*.

Tímto postupem můžeme přes více vazeb získat přístup i ke vzdálenějším objektům.

- **Kdy použít klíčové slovo *Set*?**

Jednou z nejčastějších nástrah pro začínajícího programátora je znalost použití klíčového slova *Set*. Toto klíčové slovo se musí používat, pokud je instanciována jakákoliv objektová proměnná. Naopak klíčové slovo *Set* se nesmí používat v případě přidělování hodnoty proměnné standardnímu datovému typu, jako je *string*, *numbers*, *dates* a další.

Mnemotechnická pomůcka pro toto pravidlo: pokud je možné danou proměnnou zapsat na papír, jedná se o standardní datový typ a klíčové slovo *Set* se nepoužívá. Například proměnnou jméno je možné zapsat na papír, naopak proměnnou mapa zapsat nemůžeme a je nutné při její instanciaci použít klíčové slovo *Set*.

3.4 Objektově orientované programování za použití component object model

Podle [3] je ArcObjects sada softwarových komponent, které jsou napsané v programovacím jazyce C++ a poskytují služby pro podporu GIS aplikací jak pro klienty, tak i servery.

Jak bylo uvedeno výše, pro vývoj ArcObjects byl zvolen jazyk C++, navíc je použit ještě standard Component Object Model (dále jen COM¹²). COM je často považován za specifikaci a seznam pravidel toho, jak jsou implementovány třídy a vytvářeny objekty v paměti a jak tyto objekty mezi sebou komunikují. Komunikace mezi objekty probíhá prostřednictvím rozhraní (viz dále), není proto možné vidět přímo na objekt. Mimo to ale COM také poskytuje pevnou infrastrukturu na úrovni operačního systému pro podporu jakéhokoliv programovacího jazyka, který používá DLL knihovny.

V operačním systému Microsoft Windows je COM infrastruktura umístěna přímo. Pokud chceme, aby ArcObjects fungovaly i v ostatních operačních systémech, musí být COM infrastruktura doplněna.

V kapitole 3.1 bylo zmíněno, že třídy jsou uloženy v knihovnách (DDL). Knihovna poskytuje aplikační programátorské rozhraní (API), které definuje způsob, jakým programy využívají třídy uložené v knihovně.

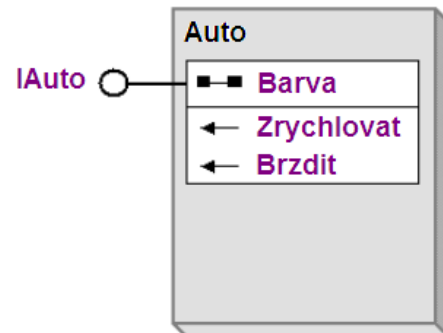
¹² COM - Component Object Model je standard, programovací technika či seznam pravidel pro vytváření tříd nezávislý na programovacím jazyku či operačním systému.

Rozhraní je označované symbolem tzv. „lízátka“ (viz obr. 3.20) a na začátku jeho názvu je vždy písmeno *I* (*Interface*). Výše v textu bylo uvedeno, že třída je soubor vlastností a metod. Pokud mezi ně a třídu bude umístěno rozhraní, je možné říci, že třída má rozhraní a rozhraní je soubor vlastností a metod. Vždy, když se používají vlastnosti nebo metody třídy, musí se k nim přistupovat prostřednictvím rozhraní.

Pokud vytváříme novou proměnnou, deklarujeme ji k rozhraní, ne ke třídě. Pak už jen stačí nastavit novou proměnnou na danou třídu, stejně jako tomu bylo v předešlých případech. Jakmile je takovýto objekt vytvořen, má přístup k vlastnostem a metodám, které jsou uloženy ve třídě, na kterou je daný objekt odkazován. Zde byl vytvořen objekt *MojeAuto*, který je odkazován na třídu *Auto*. Objekt *MojeAuto* má přístup k vlastnostem *Barva* a metodám *Zrychlovat* a *Brzdit*.

```
Dim MojeAuto As IAuto
Set MojeAuto = New Auto
```

```
MojeAuto.Barva = "zelena"
MojeAuto.Zrychlovat
MojeAuto.Brzdit
```

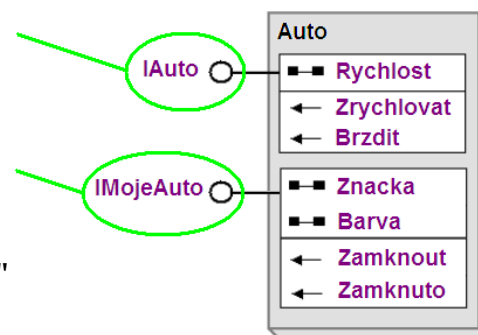


Obr. 3.20: Ukázka kódu s použitím rozhraní

Většina tříd má více než jedno rozhraní. Každé rozhraní obsahuje různé vlastnosti a metody. Podle toho, které z nich budeme potřebovat, přistupujeme ke třídě prostřednictvím konkrétního rozhraní. Např. pokud potřebujeme změnit barvu našeho auta *MojeAuto*, deklarujeme naši proměnnou pomocí rozhraní *IAuto* ze třídy *Auto*, které v sobě zahrnuje vlastnost *Barva*. V případě, že chceme změnit *Značku* našeho auta, deklarujeme proměnnou pomocí rozhraní *IMojeAuto*, které obsahuje vlastnost *Značka*.

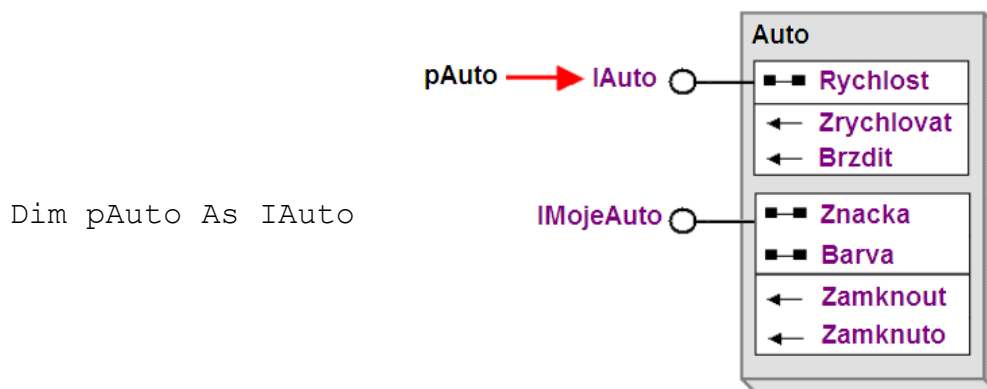
```
Dim pMojeAuto As IAuto
Set pMojeAuto = New Auto
pMojeAuto.Barva = "zluta"
```

```
Dim pMojeAuto As IMojeAuto
Set pMojeAuto = New Auto
pMojeAuto.Značka = "Toyota"
```



Obr. 3.21: Ukázka kódu s použitím dvou rozhraní

Podle jmenné konvence používáme na začátku názvu proměnné písmeno *p* (*pointer = ukazatel*), protože proměnné jsou deklarovány v závislosti na rozhraní, tedy ukazují na konkrétní rozhraní. Na začátku názvu rozhraní pak používáme písmeno *I* (*Interface = rozhraní*), aby bylo možné rozhraní rozeznat. Z následujícího obrázku je vidět, že proměnná *pAuto* ukazuje na rozhraní *IAuto*. Používání jmenné konvence usnadňuje orientaci jak ve vašem vlastním, tak i v cizím kódu.



Obr. 3.22: Ukázka kódu s přepínáním rozhraní

- **Kód pro přepínání rozhraní nad jedním objektem**

Pokud chceme použít vlastnosti a metody obou rozhraní v rámci jednoho objektu, postupujeme následujícím způsobem:

- nejprve vytvoříme proměnnou např. `pAuto` a definujeme ji nad rozhraním `IAuto`,
- nastavíme ji na novou proměnnou `New Auto`, nyní můžeme pro tento objekt libovolně používat vlastnosti a metody, které jsou uloženy pod rozhraním `IAuto` (např. nastavovat barvu auta, nechat ho zrychlovat nebo brzdit),

```
Dim pAuto As IAuto
Set pAuto = New Auto
pAuto.Barva = "cerna"
```

- definujeme novou proměnnou např. `pMojeAuto` nad rozhraním `IMojeAuto`, ale nebudeme vytvářet další objekt `Auto`, jen nastavíme proměnnou `pMojeAuto` na proměnnou `pAuto`. Takto nastavená proměnná může využívat vlastnosti a metody rozhraní `IMojeAuto` (můžeme nastavit značku auta a můžeme ho zamknout),

```
Dim pMojeAuto As IMojeAuto
Set pMojeAuto = pAuto
pMojeAuto.Značka = „Peugeot“
pMojeAuto.Zamknout
```

- nyní máme jeden objekt `Auto`, dvě proměnné, které se na tento objekt odkazují, a každá z proměnných ukazuje na jiné rozhraní. Tato technika vytváření jednoho objektu, který má větší množství proměnných a používá různá rozhraní, se nazývá *QueryInterface* (zkráceně *QI*).

3.5 Příklady využití ArcObjects

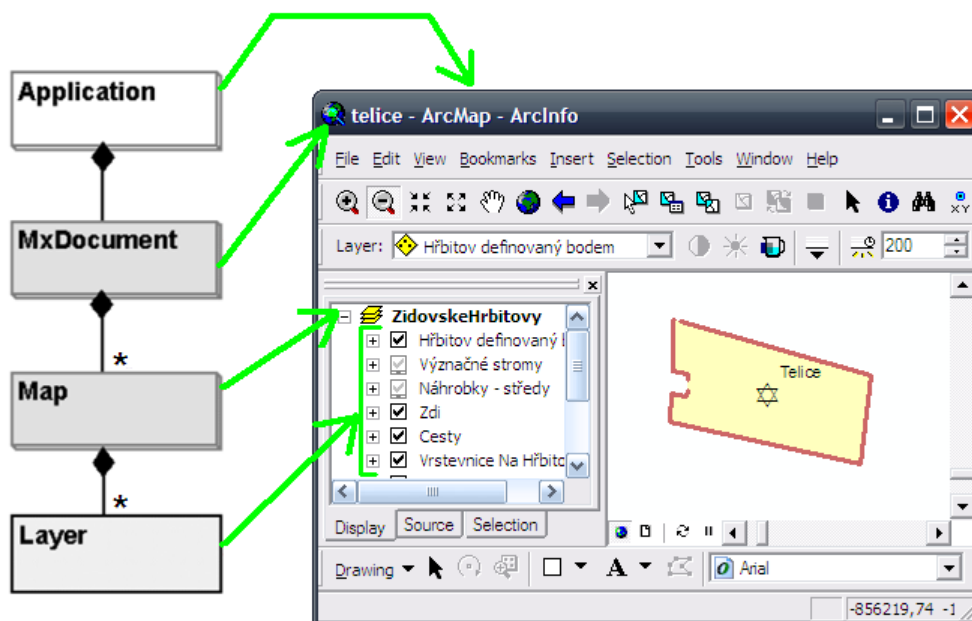
Výše uvedené ukázky byly předváděny na fiktivních, ale názorných příkladech. Nyní se podíváme, jak tento postup funguje při použití skutečných ArcObjects.

3.5.1 Práce s vrstvou

Vezměme si jednoduchý příklad – chceme získat vrstvu a změnit její jméno. Pokud otevřete v aplikaci ArcMap novou vrstvu, její jméno je automaticky nastaveno podle jména *FeatureClass* (třídy prvků). Jméno vrstvy lze samozřejmě změnit uživatelsky po otevření vlastností dané vrstvy. Zde to však bude provedeno pomocí programování tak, aby byla usnadněna práce uživatelům. Postup je možné si rozdělit na několik částí:

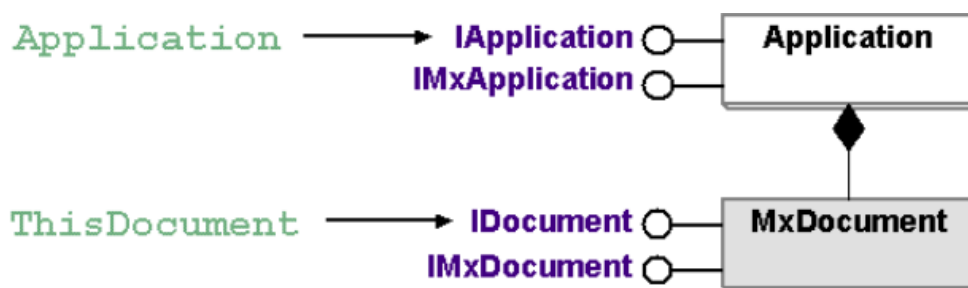
- čtení diagramů, abychom zjistili, jak napsat kód, které třídy budeme potřebovat a jak jsou navzájem objekty organizovány,
- použití *QI* (viz 3.4), abychom mohli v rámci jednoho objektu používat všechny vlastnosti a metody, které budeme potřebovat,
- získání sousedního objektu – většinou se stává, že další objekt, který je třeba použít, není přímo sousedícím objektem, ale je od toho současného vzdálen přes řadu dalších rozhraní, zde opět uplatníme *QI* (viz 3.4). K objektu, který je potřeba, je možné se dostat prostřednictvím postupu po jednotlivých sousedních objektech,
- použití vlastností,
- použití metod.

Po spuštění aplikace ArcMap dochází k vytvoření základních objektů ze tříd *ArcObjects*. Z obrázku 3.23 je patrné, že ve vztazích mezi objekty *Application* (zde ArcMap) a *MxDocument* (obecně každý MXD dokument, zde „telice“) není znázorněna žádná násobnost. Jedná se o kardinalitu 1:1. Oproti tomu relace mezi *MxDocument* a *Map*, stejně tak jako mezi *Map* a *Layer* je vyjádřena vztahem symbolem * (tzn. vztah o kardinalitě 1:M). Znamená to, že jeden mapový dokument se skládá z mnoha map a mapa je tvořena různým počtem vrstev.



Obr. 3.23: Objekty vytvořené spuštěním aplikace ArcMap

Začneme s objektem *MxDocument*. Tento objekt má přednastavenou proměnnou, která se jmenuje *ThisDocument* a ukazuje na rozhraní *IMxDocument* (viz obr. 3.24). Tato proměnná je připravena k použití bez toho, abychom ji deklarovali nebo nastavovali. Stejně tak je přednastavená i proměnná *Application*, která ukazuje na rozhraní *IApplication*. Tyto dvě proměnné jsou instanciovány při každém spuštění aplikace ArcMap či ArcCatalog. Kdykoliv otevřeme aplikaci ArcMap nebo ArcCatalog, vždy se automaticky objeví i objekty *Application* a *MxDocument*, proto můžeme kdykoliv použít i výše uvedené předdefinované proměnné.



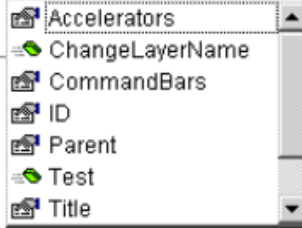
Obr. 3.24: Automatické proměnné

Pokud v editoru pro VBA napíšeme název této proměnné (*ThisDocument*) a uděláme za ní tečku, automaticky se objeví všechny vlastnosti a metody, které obsahuje rozhraní, na které tato proměnná ukazuje, tedy rozhraní *IDocument*.

```
Public Sub Test()

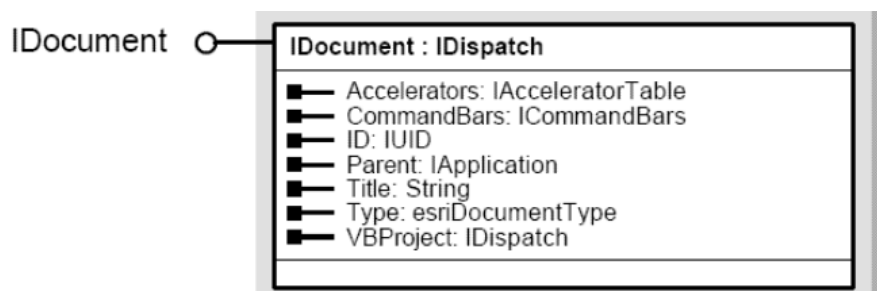
    MsgBox ThisDocument.

End Sub
```



Obr. 3.25: Automatický výpis vlastností a metod

Stejné vlastnosti a metody můžeme vidět i v Object Model Diagram, pokud si najdeme požadované rozhraní *IDocument*, které je součástí třídy *MxDocument*.



Obr. 3.26: Zobrazení v Object Model Diagram

Použijeme metodu *QI*, abychom mohli použít vlastnosti a metody z rozhraní *IDocument* a *IMxDocument*. Budou nám stačit pouhé dva řádky kódu:

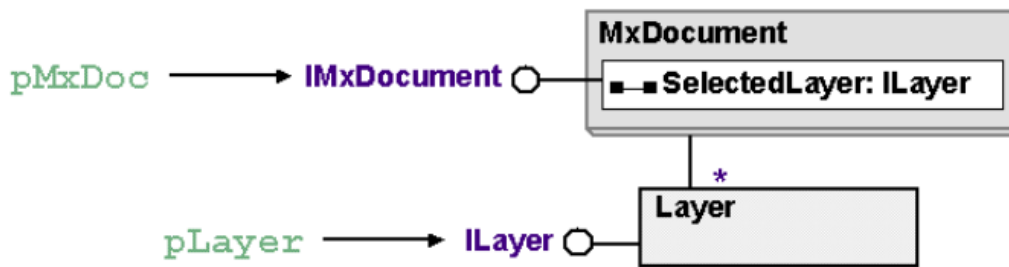
```
Dim pMxDocument As IMxDocument
Set pMxDocument = ThisDocument
```



Obr. 3.27: Použití *QI*

Nyní jsme vytvořili proměnnou *pMxDoc*, která ukazuje na rozhraní *IMxDocument* a zároveň je nastavena na předdefinovanou proměnnou *ThisDocument*, která ukazuje na rozhraní *IMxDocument*.

Rozhraní *IMxDocument* obsahuje vlastnost s názvem *SelectedLayer*, která vrací rozhraní *ILayer* vrstvy, která je právě vybraná. Aby kód proběhl správně, musíme nejprve v aplikaci ArcMap vybrat vrstvu, jejíž název chceme změnit.

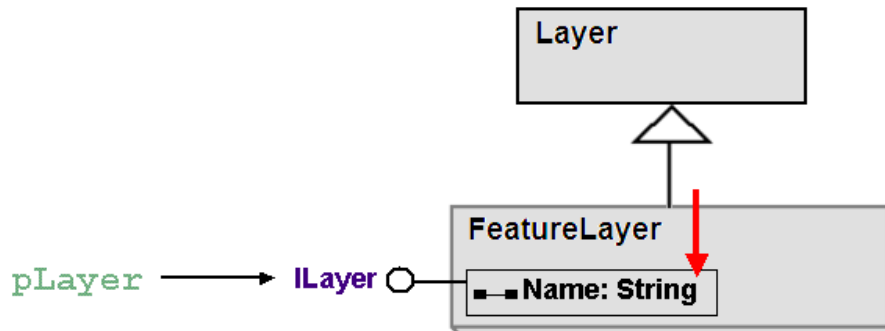


Obr. 3.28: Ukázka vlastnosti SelectedLayer

Nejprve deklaruje novou proměnnou *ILayer*, a pak ji nastavíme na požadovanou vlastnost.

```
Dim pLayer As ILayer
Set pLayer = pMxDocument.SelectedLayer
```

Nyní proměnná *pLayer* ukazuje na rozhraní *ILayer*. Toto rozhraní má vlastnost s názvem *Name*, která vrací jméno vrstvy jako proměnnou typu *String*.

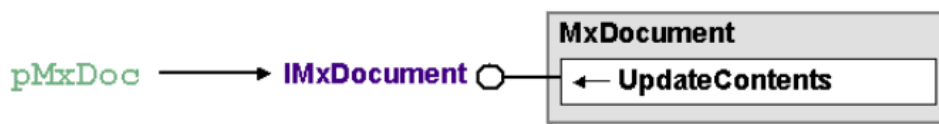


Obr. 3.29: Ukázka nastavení ukazatele a rozhraní

Proto použijeme následující kód a při nastavení jména vrstvy použijeme proměnnou typu *String*, která se umísťuje do dvojitého uvozovky.

```
pLayer.Name = „Cesty na hrbitove“
```

Pokud kód spustíme, jméno vrstvy se změní v současné paměti, ale v aplikaci ArcMap v *Table of Contents* nevidíme nový název vrstvy. *Table of Contents* je také objekt, kterému musíme říct, aby se obnovil v závislosti na nastavení současné paměti. Použijeme metodu z rozhraní *IMxDocument*, která se jmenuje *UpdateContents*.



Obr. 3.30: Ukázka metody UpdateContents

Kód bude vypadat následujícím způsobem:

```
pMxDocument.UpdateContents
```

Pro práce s ArcObjects je vhodné postupovat po určitých krocích:

- najít v Object Model Diagram třídu, rozhraní, vlastnost nebo metodu, se kterou potřebujeme pracovat,
- vybrat si startovací bod – zde je vhodné použít jednu z předdefinovaných proměnných (*Application* nebo *ThisDocument*) či *CoClass*,
- postupovat z jednoho objektu na další za použití *QueryInterface* tak, jak potřebujeme.

Celý kód pak bude vypadat následujícím způsobem:

```
Dim pMxDocument As IMxDocument
Set pMxDocument = ThisDocument
Dim pLayer As ILayer
Set pLayer = pMxDocument.SelectedLayer
pLayer.Name = „Cesty na hrbitove“
pMxDocument.UpdateContents
```

3.5.2 Práce s geometrií

Další ukázkou je práce s geometrií. Vytvořme například přímku. Pro tvorbu linie neboli přímky v geometrii je potřebné znát počáteční a koncový bod této přímky. Chceme pracovat s geometrií, podíváme se, co nám nabízí rozhraní *IGeometry*. Zde nalezneme *CoClasses Point* a *Line*. Nejprve si deklaruje potřebné objekty: počáteční (*pPointFrom*) a koncový bod (*pPointTo*) a jednu linii (*pLine*):

```
Dim pLine As ILine
Dim pPointFrom As IPoint
Dim pPointTo As IPoint
```

Poté provedeme instanci deklarováných objektů:

```
Set pLine = New Line
Set pPointFrom = New Point
Set pPointTo = New Point
```

Dále použijeme u vytvořených objektů bodů metody třídy *Point* nazvané *X* a *Y*. Tyto metody nastavují *X* a *Y* souřadnice daného bodu:

```
pPointFrom.X = 10
pPointFrom.Y = 10
pPointTo.X = 10
pPointTo.Y = 20
```

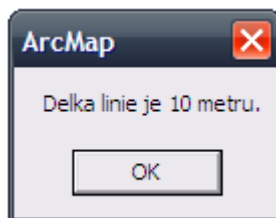
Nakonec použijeme funkci třídy *Line*, pojmenovanou taktéž *PutCoords*, abychom nastavili počáteční a koncový bod přímky. Zde uplatníme již vytvořené body *pPointFrom* a *pPointTo*:

```
pLine.PutCoords pPointFrom, pPointTo
```


Abychom se přesvědčili, zda byla přímka skutečně vytvořena, můžeme vypsát její délku pomocí vlastnosti třídy *Line* nazvané *Length*:

```
MsgBox „Delka linie je „ & pLine.Length & „ metru.“
```

Odpověď je zobrazena na obr. 3.31.



Obr. 3.31: Vypsání MsgBox obsahující údaje o délce přímky

3.6 Další zdroje pro podporu při práci s ArcObjects

Další možností, kde získat důležité informace a podporu při práci s ArcObjects je ArcGIS Desktop Resource Center (viz [3]). Tuto možnost jsem pro práci s ArcObject využívala nejvíce. Pro začátečníky je vhodné použít postranní panel nápovědy, kde je možné prohledávat požadované funkce v záložce *ArcObjects library reference*. Pokud víme, co přesně hledáme, jednoduše napíšeme název rozhraní nebo funkce do vyhledávače v [3]. Po nalezení kompletního popisu rozhraní a funkcí je už snadné se zorientovat co a jak napsat. V mnohých případech jsou v daných dokumentech dokonce odkazy na příklady naprogramovaných a fungujících kódů, což podstatně usnadňuje práci.

Předtím, než začnete vytvářet vlastní kód, je vhodné prohledat diskusní fórum (viz [10]) a zjistit, zda se už někdo nezabýval stejným problémem. Pokud najdete již hotový stejný nebo podobný kód a podaří se vám v něm zorientovat, velice to usnadní celý programovací proces. Pokud by měl člověk při psaní kódu nějaké potíže a potřeboval poradit, je možné přidat na fórum svůj vlastní příspěvek, ale tato akce vyžaduje přihlášení uživatele. Váš dotaz je pak zpracován školeným poradcem a po zodpovězení dotazu vám bude doručeno upozornění e-mailem.

4 Návrh a implementace aplikačního rozhraní pro práci s datovou sadou židovských hřbitovů

Nyní máme potřebné znalosti pro práci s ArcObjects. Je tedy možné začít vytvářet poslední potřebné vrstvy pro doplnění třívrstvé architektury, tedy aplikační a prezentační vrstvu. Aplikační vrstva obsahuje jádro aplikace, logiku a funkce aplikace, výpočty a zpracování. V rámci prezentační vrstvy je vytvářeno grafické uživatelské rozhraní (GUI), které při správném provedení umožňuje kontrolovat vstupní parametry zadávané uživatelem.

4.1 Návrh formuláře pro získání dat z náhrobku

Ve spolupráci se zadavatelem, v tomto případě PhDr. Ing. Václavem Chvátalem, byl vytvořen prvotní formulář, který má za úkol uživateli usnadnit identifikaci každého z náhrobků a zajistit snadný a rychlý přístup k jeho dokumentaci, konkrétně zobrazení fotografie, přepisu a překladu textu na náhrobku a informací o zesnulých osobách. První hrubý návrh formuláře je zobrazen na obr. 4.1.



Obrázek 4.1: První návrh formuláře

Při prvním testování bylo zjištěno, že rozměry fotografie a přepisu s překladem jsou příliš velké na to, aby byly tyto prvky zobrazovány přímo do formuláře, proto byla zvolena forma odkazu pomocí tlačítka. Po kliknutí na toto tlačítko jsou výše uvedené dokumenty zobrazeny v samostatných oknech (viz obr. 4.2). Atributová data byla ponechána v původním formuláři.

Z obr. 4.2 je patrné, že atributová data jsou uváděna až pro dvě osoby. Ve většině případů je u židovských hřbitovů možné nalézt na jednom náhrobku jméno jedné, případně dvou osob (jedná-li se o společně pohřbené manžele). Na židovských hřbitovech se zřídka vyskytují také mohyly nebo památníky zemřelým, na nichž jsou uvedeny více než dvě osoby. Takovéto prvky se však v databázi židovských hřbitovů, zpracované v rámci bakalářské práce (viz [1]), nevyskytují, proto byly i zde zanedbány.

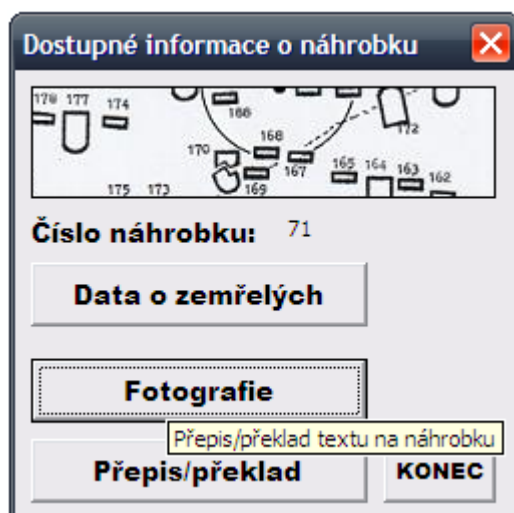
Obrázek 4.2: Druhý návrh formuláře

Ani tento formulář však nebyl dostatečně vyhovující, a to hned z několika důvodů:

- je příliš velký a proto brání v náhledu na mapový dokument,
- pokud se na požadovaném náhrobku vyskytují informace pouze o jedné osobě, zbývá ve formuláři zbytečná prázdná plocha,
- není dostatečně přehledný.

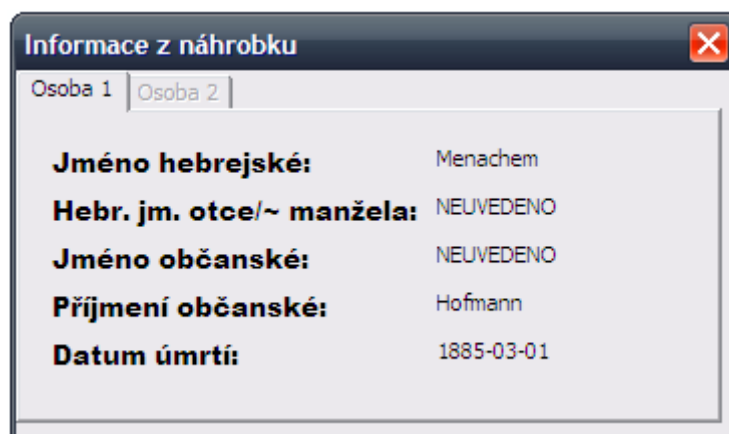
Dalším, a posledním krokem byl návrh jednoduchého a přehledného formuláře, který obsahuje pouze číslo náhrobku, příkazové tlačítko pro ukončení formuláře a tři příkazová tlačítka pro odkazy na fotografii, přepis a překlad náhrobku a data uvedená na náhrobku (viz obr. 4.3).

Protože názvy některých tlačítek jsou příliš dlouhé, byla kvůli zachování přehlednosti formuláře použita pouze jejich část. Tento problém je vyřešen pomocí funkce *ToolTip*, díky níž se celý název formuláře volaného daným tlačítkem objeví, jakmile uživatel najede do prostoru tlačítka myší (viz obr. 4.3).



Obr. 4.3: *Finální návrh formuláře a použití funkce Tooltip*

Zobrazení dat uvedených na náhrobku bylo realizováno pomocí nového formuláře (viz obr. 4.4), který se zobrazí po kliknutí na příkazové tlačítko „Data z náhrobku“.



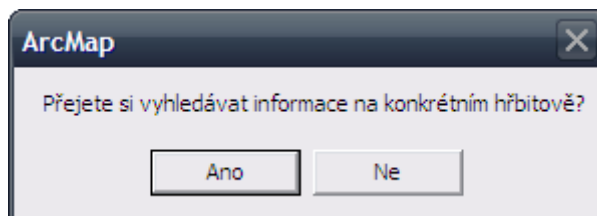
Obr. 4.4: *Formulář zobrazující data uvedená na náhrobku*

Pokud jsou na náhrobku uvedeny informace také o druhé osobě, aktivuje se záložka „Osoba 2“ a je možné mezi oběma záložkami pohodlně přepínat. Implementační řešení pro více než dvě osoby nebylo uskutečněno, protože informace o více než 2 osobách se zpravidla na jednom náhrobku nenachází (viz kapitola 2.3, Krok 3).

4.2 Návrh formuláře pro vyhledávání náhrobků podle zadaných informací

Dalším požadavkem pro aplikační rozhraní bylo umožnit uživateli vyhledávání konkrétních náhrobků podle zadaných informací, kterými uživatel disponuje. Pro urychlení vyhledávacího procesu uživatele jsou vytvořeny dvě varianty tohoto formuláře, které se odvíjí od znalosti názvu židovského hřbitova, na kterém chce uživatel vyhledávat informace.

Po spuštění vyhledávací funkce je uživatel dotázán, zda chce informace vyhledávat na konkrétním hřbitově (viz obr. 4.5). Pokud ano, objeví se formulář, který umožňuje uživateli nejprve vybrat hřbitov, a poté se zbylé *Combo Box*¹³ naplní daty vybraného hřbitova (viz obr. 4.6). Pokud ne, objeví se formulář, jehož *Combo Boxy* jsou automaticky naplněny potřebnými daty všech hřbitovů (viz obr. 4.7).



Obr. 4.5: Volba nabídnutá uživateli po spuštění vyhledávací funkce

Obr. 4.6: Formulář pro vyhledání na konkrétním hřbitově

¹³ *Combo Box* – je kombinací textového pole a seznamu. Obvykle zobrazuje jen položku jednoho řádku, ale po kliknutí na jeho šipku může být rozbalen do seznamu.

Obr. 4.7: Formulář pro vyhledávání na všech hřbitovech

Původní myšlenkou bylo použití *Text Boxů*¹⁴, kam by mohl uživatel přímo zadat konkrétní hledaný údaj. Tato varianta však byla vyloučena z důvodu použití speciální symboliky v registru, která byla vytvořena PhDr. Ing. Chvátalem (viz kap. 2.1, obr. 2.1). Tato symbolika výrazně snižuje pravděpodobnost shody textu zadaného uživatelem s textem uloženým v registru. Proto byla nakonec zvolena varianta *Combo Boxů*.

Postup při využití formuláře pro vyhledávání vypadá následovně:

- nejprve se uživatel rozhodne, zda chce vyhledávat informace na konkrétním hřbitově nebo na všech hřbitovech,
- v případě vyhledávání na konkrétním hřbitově rozbalíme první *Combo Box*, kde jsou uvedeny všechny židovské hřbitovy obsažené v geodatabázi, seřazené podle abecedy,
- vybereme jeden z uvedených židovských hřbitovů a stiskneme tlačítko „Vyber hřbitov“,
- ostatní *Combo Boxy* se naplní daty vztahujícími se k vybranému hřbitovu, data jsou opět seřazena podle abecedy, pokud zvolíme vyhledávání na všech hřbitovech, přeskočíme druhý a třetí krok a začneme zde,
- uživatel vybere z nabízených možností taková data, která jsou mu známa (např. rok úmrtí),
- pokud uživatel nevybere žádnou z možností, bude upozorněn, aby zvolil alespoň jednu,
- po stisknutí příkazového tlačítka „Hledej“ bude vyhledána požadovaná kombinace dat,
- pokud existují náhrobky splňující požadovanou kombinaci, bude možné je zobrazit pomocí tlačítka „Zobrazit“,
- pokud je požadovaná kombinace nereálná, je uživatel vyzván k opravě zadání.

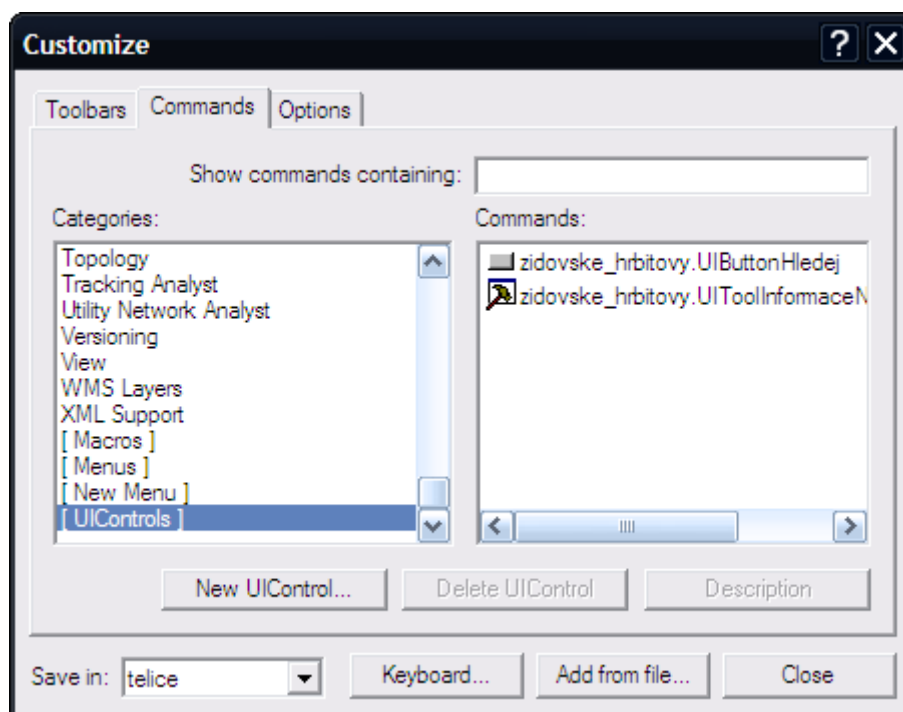
¹⁴ *Text Box* – obdélníkový rámeček na obrazovce, do kterého je možné psát text.

4.3 Implementace aplikačního rozhraní

Po zpracování návrhu aplikačního rozhraní je možné přistoupit k jeho realizaci. Tvorba aplikačního rozhraní byla realizována ve vývojovém prostředí Visual Basic Editor firmy Microsoft, pomocí programovacího jazyka Visual Basic For Applications.

4.3.1 Tvorba vlastního tlačítka

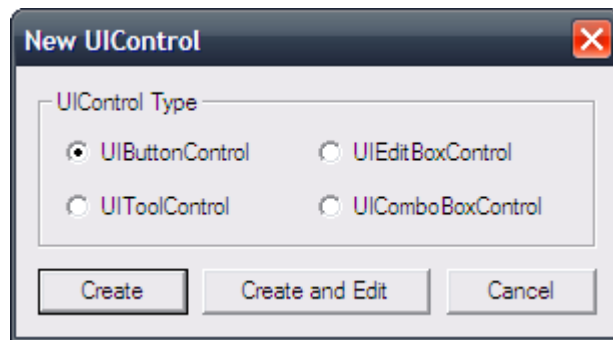
Podle zadavatele bylo nutné vypracovat dva fungující formuláře. Aby bylo možné je spouštět přímo v aplikaci ArcMap, bylo nutné nejprve vytvořit vlastní tlačítka, která jednotlivé formuláře spouští. Toto je možné provést pomocí funkce *Tools – Customize*. Po otevření okna je zobrazena záložka *Commands*. Po nalezení kategorie *[UIControls]* je vytvořeno nové ovládací tlačítko pomocí funkce *New UIControl* (viz obr. 4.8).



Obr. 4.8: Okno funkce *Customize*

Bylo nutné zvolit, jaký typ ovládacího tlačítka chceme vytvořit. Možnosti jsou následující (viz obr. 4.9):

- *UIButtonControl* – toto tlačítko se většinou používá na tzv. *Click Events*, např. pokud kliknutím spustíme nějakou akci.
- *UIToolControl* – tlačítko používající se především pro tzv. *Mouse/Key Events*, např. pokud uživatel kliká přímo do mapového pole.
- *UIEditBoxControl* – používá se, pokud uživatel zadává přímo nějaké informace.
- *UIComboBoxControl* – se pro poskytnutí skupiny dat, ze které je možné vybírat.



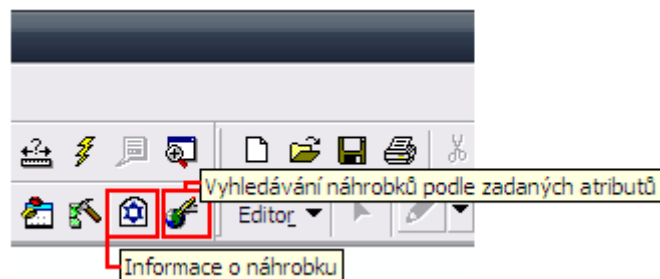
Obr. 4.9: Okno funkce *New UIControl*

Pro formulář zobrazující informace z konkrétního náhrobku bylo zvoleno *UIToolControl*, protože požadujeme, aby se informace zobrazily právě po kliknutí na daný náhrobek, který se nachází v mapovém poli. Pro formulář určený k vyhledávání bylo pak zvoleno *UIButtonControl*, protože potřebujeme použít *Click Event* (událost spuštěná kliknutím).

Pro spouštění nástrojů vzniklého aplikačního rozhraní byly zvoleny ikony zobrazené na obr. 4.10.

- Pro nástroj *Informace o náhrobku* byla vytvořena vlastní ikona znázorňující Davidovu hvězdu (symbol judaismu) umístěnou na židovském náhrobku (stéle).
- U nástroje *Vyhledávání náhrobků podle zadaných atributů* byla zvolena jedna ze základních nabízených ikon reprezentující svým symbolem vyhledávání.

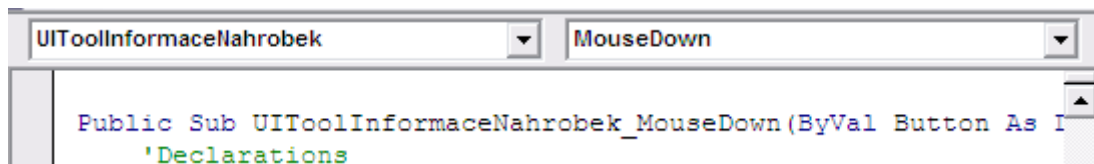
Pro přehlednost byla navíc při implementaci použita funkce *ToolTip*, která umožňuje zobrazení kompletního názvu daného nástroje, pokud nad něj uživatel umístí ukazatel myši.



Obr. 4.10: Ukázka ikon a využití funkce *ToolTip*

4.3.2 Implementace formuláře pro získání dat z náhrobku

Nejprve bylo nutné pro již vytvořené tlačítko nastavit parametry procedury tak, aby procedura byla procedura spuštěna po kliknutí myši do mapového okna. To je možné nastavit právě u tlačítek typu *UIToolControl*, a to jednoduchým krokem: v pravém horním rohu okna určeného pro psaní kódu nastavíme událost jako *MouseDown* (viz obr. 4.11).



Obr. 4.11: Nastavení procedury pro spuštění po kliknutí myši do mapového okna

Jedna z možností implementace formuláře pro získání dat z náhrobku je použití následujících rozhraní:

- *IPoint*, neboli bod, díky němuž získáme souřadnice bodu, na který jsme klikli myší do mapového okna. Kliknutím získáme obrazové souřadnice pixelu, do kterého jsme se trefili. Obrazové souřadnice jsou převáděny do mapových a ty se pak ukládají jako vstupní hodnota procedury. Tyto souřadnice jsou však natolik konkrétní (souřadnice právě jednoho pixelu), že eliminují pravděpodobnost, že by uživatel kliknutím získal právě souřadnice bodu označujícího náhrobek, proto použijeme další rozhraní,
- *IEnvelope*, neboli obálku. Tato funkce umožňuje nastavit okolo bodu o známých souřadnicích pravoúhloou dvourozměrnou obálku libovolné velikosti. Optimální velikost byla nastavena na 0,5 mapových jednotek (pro geodatabázi židovských hřbitovů je nastavena metrická soustava).

Kód realizující výše popsané vypadá takto:

```
'Get the selected point using the MouseDown event by click to
the Map
Set pPoint = New Point
Set pPoint =
pActiveView.ScreenDisplay.DisplayTransformation.ToMapPoint(x, y)

'Put point geometry into Topological Operator
Dim dx As Double
dx = 0.5 'In map units
Dim dy As Double
dy = 0.5 'In map units

Dim px As Double
px = pPoint.x
Dim py As Double
py = pPoint.y

'Set Envelope
Dim pEnvelope As IEnvelope
Set pEnvelope = New Envelope
pEnvelope.PutCoords px + dx, py + dy, px - dx, py - dy
```

Prohledáme obálku a zjistíme, zda se v ní nalézá nějaký bod. Pokud ano, dojde k zobrazení formuláře, pokud ne, zobrazí se okno s informací oznamující, že uživatel klikl vedle.

Nyní je vybrán bod, na který jsme klikli a zobrazen formulář (viz Adresářová struktura příloženého DVD). V dalším kroku potřebujeme získat data vztahující se k vybranému

náhrobku. Ta jsou uložena v samostatné tabulce (*Personal Geodatabase Table*), která je s prvkovou třídou *Nahrobek* provázána pomocí relace (viz obr. 2.2). Použijeme rozhraní:

- *IRelationshipClass*, které umožňuje svou funkcí *GetObjectRelatedToObject* získat objekt provázaný s vybraným bodem relací.

Kód vypadá následovně:

```
'Declaration
Dim pRelationshipClass As IRelationshipClass
Set pRelationshipClass = pEnumRelClass.Next

Dim pOriginObject As IObject
Set pOriginObject = pFeature
Dim pDestinatedObjectSet As ISet

'Fault handling
If pOriginObject Is Nothing Then
    Exit Sub
Else
    Set pDestinatedObjectSet =
pRelationshipClass.GetObjectsRelatedToObject(pOriginObject)
End If

Dim pDestinatedObject As IObject
Set pDestinatedObject = pDestinatedObjectSet.Next

'Get values of the selected feature into the form
Dim pDestinatedObjectNext As IObject
Set pDestinatedObjectNext = pDestinatedObjectSet.Next
```

Získávání potřebných dat pro vyplnění formuláře, načtení dokumentů přepisu/překladu a fotografie náhrobku a veškerá potřebná ošetření, včetně nastavení relativních cest ke všem dokumentům jsou uvedena v DVD příloze (viz Adresářová struktura přiloženého DVD).

Pozn.: Aby bylo možné pracovat s dokumenty fotografie a přepis/překlad, bylo nutné převést z formátu PNG do formátu JPG, který je možné zobrazovat přímo ve formulářích. Zobrazováním dokumentů přímo ve formulářích se předejde nevhodné volbě softwaru pro zobrazování dokumentů. Nemůže tak dojít k potížím jen díky tomu, že by uživatel neměl na svém počítači daný software nainstalovaný. Hromadná konverze byla provedena v softwaru ReaConverter v. 5.5.

Ne ke každému náhrobku obsaženému v geodatabázi existuje také jeho dokumentace. Jedná-li se například o náhrobky typu „black“ tedy sokl bez náhrobku nebo je-li náhrobek označen jako ležící a leží nápisem k zemi, je podrobnější dokumentace zbytečná. Neexistuje ale žádné pravidlo, které by určovalo, u kterého z náhrobků dokumentace chybí. Proto bylo nutné při zobrazování dokumentů fotografie a přepisu/překladu použít jednoduchou funkci, která zjišťuje, zda dokumenty ke konkrétnímu náhrobku vůbec existují. Kód je následující:

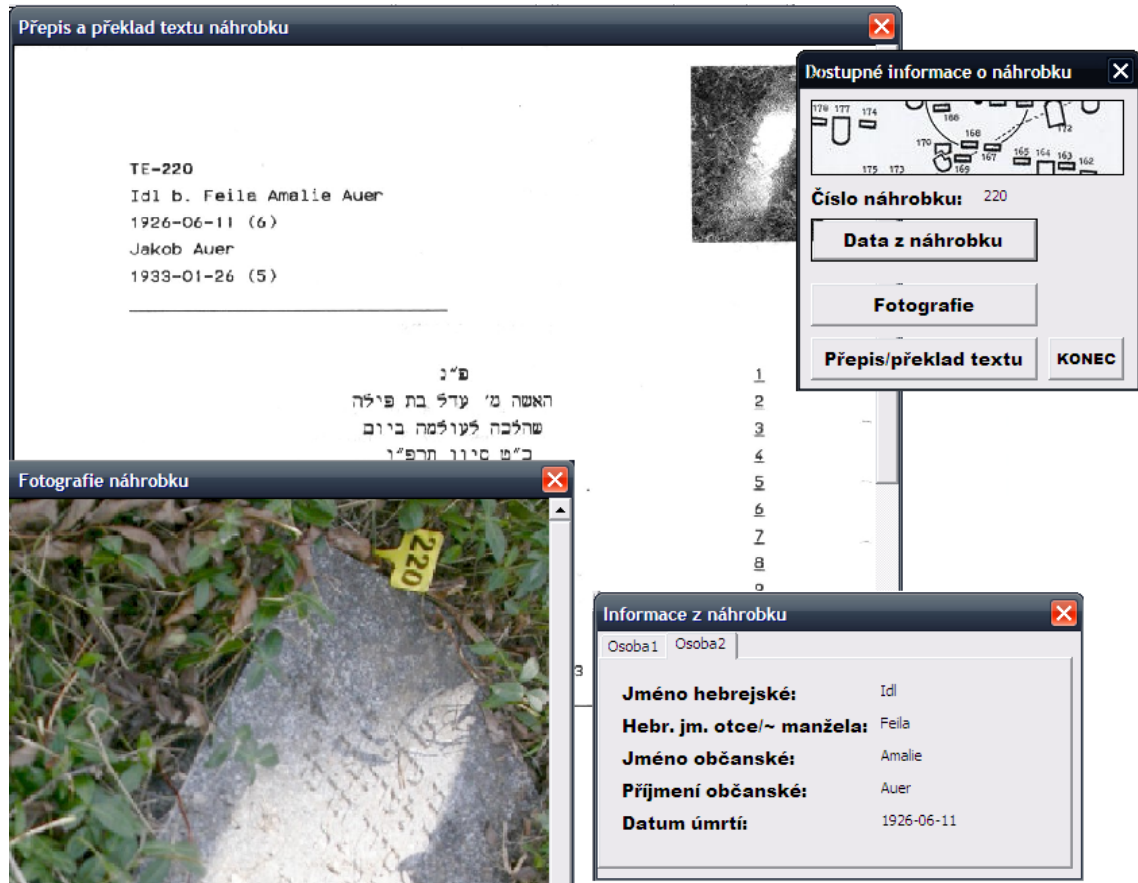
```

Public Function FExists(OrigFile As String)
    Dim fs
    Set fs = CreateObject("Scripting.FileSystemObject")
    FExists = fs.FileExists(OrigFile)
End Function

```

Pozn.: Tento kód byl převzán z [9].

Výsledný formulář pro získání dat z náhrobku včetně všech jeho funkcí je zobrazen na obr. 4.12.



Obr. 4.12: Vizualizace formuláře pro získání dat z náhrobku

4.3.3 Implementace formuláře pro vyhledávání

Posledním formulářem, který bylo podle požadavků zadavatele nutno implementovat, byl formulář, který bude uživateli soužit k vyhledávání a zobrazování vyhledaných údajů (viz obr. 5.4).

Prvním krokem při tvorbě tohoto formuláře bylo vytvoření dotazovacího okna, kde se uživatel rozhodne, zda chce vyhledávat informace na známém hřbitově, nebo jestli název hřbitova nezná. V tomto případě budou prohledávána data všech zdokumentovaných hřbitovů.

Pokud chce uživatel vyhledávat informace na konkrétním hřbitově, je nutné získat názvy všech hřbitovů obsažených v databázi, jejich seřazení podle abecedy a následné vypsání do *Combo Boxu*. Zde je využito rozhraní *IQueryFilter*, které nám umožní řazení vybraného sloupce atributové tabulky dle abecedy:

```
'Arrange the Features in alphabetic order
Dim pQFilter As IQueryFilter
Set pQFilter = New QueryFilter
pQFilter.SubFields = "Nazev"

Dim pQFilterDefinition As IQueryFilterDefinition
Set pQFilterDefinition = pQFilter      'Set pQFilterDefinition on
pQFilter, which contain the field "Nazev"
pQFilterDefinition.PostfixClause = "ORDER BY Nazev"

Dim pFCursor As IFeatureCursor
Set pFCursor = pFClassHrbitovy.Search(pQFilter, True)
```

Deklarujeme proměnou *tmp*, díky které je umožněno relativní nastavení atributového sloupce podle jeho názvu, a konečně naplníme *Combo Box* pomocí cyklu s pevným počtem opakování (*For*).

```
'tmp is used as a variable for storing the number of the field
Nazev and that is used for set the relative path to the required
value
Dim tmp As Long
tmp = pFCursor.FindField("Nazev")

'Fill the Combo Box
Dim x As Integer
Dim pocetHrbitovu As Long
pocetHrbitovu = pFClassHrbitovy.FeatureCount(Nothing)
Dim pFeature As IFeature

For x = 1 To pocetHrbitovu
    Set pFeature = pFCursor.NextFeature
    frmHledej.cboHrbitovy.AddItem pFeature.Value(tmp)
Next x
```

Poté, co uživatel zvolí hřbitov, na kterém chce vyhledávat data, je nutné naplnit i ostatní tři *Combo Boxy*, aby měl uživatel z čeho volit.

Nejprve musíme získat *ID* hřbitova, který byl vybrán v prvním kroku. To lze provést pomocí následujícího kódu:

```
'Get ID of selected Feature from cboHrbitovy
pQFilterID.WhereClause = "Nazev =" & VybranyHrbitov
Set pSelectionSetID =
pFClassHrbitovy.Select (pQFilterID,esriSelectionTypeIDSet,
esriSelectionOptionNormal, Nothing)
pSelectionSetID.Search Nothing, True, pFCursorId

'Relative path to the required value
Dim tmp1 As Long
tmp1 = pFCursorId.FindField("IdHrbitova")

Dim pFeatureHrbitovy As IFeature
Set pFeatureHrbitovy = pFCursorId.NextFeature
IdHrbitova = "" & pFeatureHrbitovy.Value(tmp1) & ""
```

Pomocí následujícího kódu budou z atributové tabulky označené jako *pTableData* vybrány všechny prvky, které mají ve sloupci *IdHrbitova* uvedenou hodnotu, která byla získána předcházejícím kódem, tzn. hodnotu hřbitova, který byl zvolen v prvním *Combo Boxu*.

```
'Get features from selected Hrbitov from Combo Box
pQFilterData.WhereClause = "IdHrbitova =" & IdHrbitova
Set pSelectionSetData = pTableData.Select (pQFilterData,
esriSelectionTypeIDSet, esriSelectionOptionNormal, Nothing)
```

Data jsou opět seřazena podle abecedy. Pomocí *For* cyklu je procházen postupně každý řádek z vybraných dat a je průběžně naplňován další *Combo Box*. Zároveň ale dochází ke kontrole duplicity dat. Pokud se v naplňovaném *Combo Boxu* již jednou daná hodnota vyskytuje, není tam znovu zařazována (viz následující kód):

```
'Going through all the selected Rows in table DataNahrobky and
fill the Combo Box cboJmHebrejske
For y = 1 To pocetDat
  Set pRowJmHebrejske = pCursorJmHebrejske.NextRow
  If pRowJmHebrejske Is Nothing Then
    Exit Sub
  Else
    If IsNull (pRowJmHebrejske.Value(tmp2)) Then
      Set pRowJmHebrejske = pCursorJmHebrejske.NextRow
    Else
      Dim ValueJmHebrejske As String
      ValueJmHebrejske = pRowJmHebrejske.Value(tmp2)
      'Delet duplicated values from Combo Box
      If
        pValueExists.CheckValueExistsJmHebrejske (ValueJmHebrejske)
        = True Then
        Set pRowJmHebrejske = pCursorJmHebrejske.NextRow
      Else: frmHledej.cboJmHebrejske.AddItem
        pRowJmHebrejske.Value (tmp2)
      End If
    End If
  End If
```

```
End If
Next y
```

Zde je ještě uvedena výše použitá funkce, která kontroluje, zda *Combo Box* již vkládanou hodnotu neobsahuje:

```
'Function which check if a value exists in a Combo Box
Public Function CheckValueExistsJmHebrejske (ValueJmHebrejske As
String) As Boolean
    Dim i As Integer
    For i = 0 To frmHledej.cboJmHebrejske.ListCount - 1
        If frmHledej.cboJmHebrejske.List(i) = ValueJmHebrejske Then
            CheckValueExistsJmHebrejske = True
            Exit Function
        End If
    Next i
End Function
```

Výše uvedený postup, který je podle kódu aplikován na naplnění *Combo Boxu* hodnotou hebrejské jméno byl aplikován rovněž pro naplnění zbývajících dvou *Combo Boxů* hodnotami občanské příjmení a rok úmrtí. Kompletní kód je uveden na DVD příloze (viz Adresářová struktura příloženého DVD).

Pro dokončení návrhu formuláře zbývají poslední dva kroky:

- nalezení požadovaných prvků podle zadání uživatele,
- a jejich následné zobrazení v mapovém dokumentu.

Při hledání požadovaných prvků podle zadání uživatele používáme pouze rozhraní *IQueryFilter*, kde uplatníme SQL¹⁵ deklaráci. Následující kód zobrazuje všechny možné kombinace, které mohou nastat při uživatelském vyplňování informací z *Combo Boxů*:

```
'All combinations of selecting features from Combo Boxes
If vybJmHebr <> "" And vybJmObc = "" And vybRokUmrti = "" Then
    pQFilter.WhereClause = "JmenoHebrejske = " & vybJmHebr
ElseIf vybJmHebr <> "" And vybJmObc <> "" And vybRokUmrti = ""
Then
    pQFilter.WhereClause = "JmenoHebrejske = " & vybJmHebr & "
AND PrijmeniObcanske = " & vybJmObc
ElseIf vybJmHebr <> "" And vybJmObc <> "" And vybRokUmrti <> ""
Then
    pQFilter.WhereClause = "JmenoHebrejske = " & vybJmHebr & "
AND PrijmeniObcanske = " & vybJmObc & " AND RokUmrti = " &
vybRokUmrti
ElseIf vybJmHebr = "" And vybJmObc <> "" And vybRokUmrti <> ""
Then
    pQFilter.WhereClause = "PrijmeniObcanske = " & vybJmObc & "
AND RokUmrti = " & vybRokUmrti
ElseIf vybJmHebr = "" And vybJmObc = "" And vybRokUmrti <> ""
Then
    pQFilter.WhereClause = "RokUmrti = " & vybRokUmrti
```

¹⁵ SQL – Structured Query Language, je to standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích.

```

ElseIf wybJmHebr = "" And wybJmObc <> "" And wybRokUmrti = ""
Then
    pQFilter.WhereClause = "PrijmeniObcanske = " & wybJmObc
ElseIf wybJmHebr <> "" And wybJmObc = "" And wybRokUmrti <> ""
Then
    pQFilter.WhereClause = "JmenoHebrejske = " & wybJmHebr & "
    AND RokUmrti = " & wybRokUmrti
ElseIf wybJmHebr = "" And wybJmObc = "" And wybRokUmrti = ""
Then
    MsgBox "Prosím vyberte alespoň jednu z možností.",
    vbExclamation
    Exit Sub
End If

```

Vybrané prvky získáme opět pomocí *For* cyklu. Nyní máme ale vybraná pouze atributová data, musíme znovu využít rozhraní *IRelationshipClass*, které nám umožní získat prvky, které jsou relací provázány s vybranými daty (viz následující kód).

```

For a = 1 To pocet
    Set pRow = pCursor.NextRow
    Set pDestObjectSet =
    pRelationshipClass.GetObjectsRelatedToObject (pRow)
    Set pDestObject = pDestObjectSet.Next
    Set pFeature = pDestObject
    pFSelection.Add pFeature
Next a

```

Zbývá poslední krok, a to zobrazení vybraných prvků v mapovém okně. Zde využijeme (stejně jako při tvorbě prvního formuláře) rozhraní *IEnvelope*. Vyskytuje se zde jeden problém. Vytvořená obálka je rozšířena kolem vybraných prvků. Pokud je ale vybrán pouze jediný prvek, obálka je bodová, tzn. o nulové velikosti. Potom dochází k tomu, že se daný prvek nezobrazuje. Proto bylo nutné vytvořit podmínku, která v tomto případě obálku zvětší, viz následující kód:

```

Sub ZoomToSelection()
Dim pMxDocument As IMxDocument
Dim pMap As IMap
Dim pActiveView As IActiveView
Set pMxDocument = Application.Document
Set pMap = pMxDocument.FocusMap
Set pActiveView = pMap
Dim pEnumFeature As IEnumFeature
Dim pFeature As IFeature
Dim pEnvelope As IEnvelope

'Do not change the extent if there are no selected features
If pMap.SelectionCount > 0 Then
'Retrieve the selected features and get selection's extent
    Set pEnumFeature = pMap.FeatureSelection
    pEnumFeature.Reset
    Set pFeature = pEnumFeature.Next
    Set pEnvelope = New Envelope
    Do While Not pFeature Is Nothing
'The new extent is combination of selected feature's extents

```

```

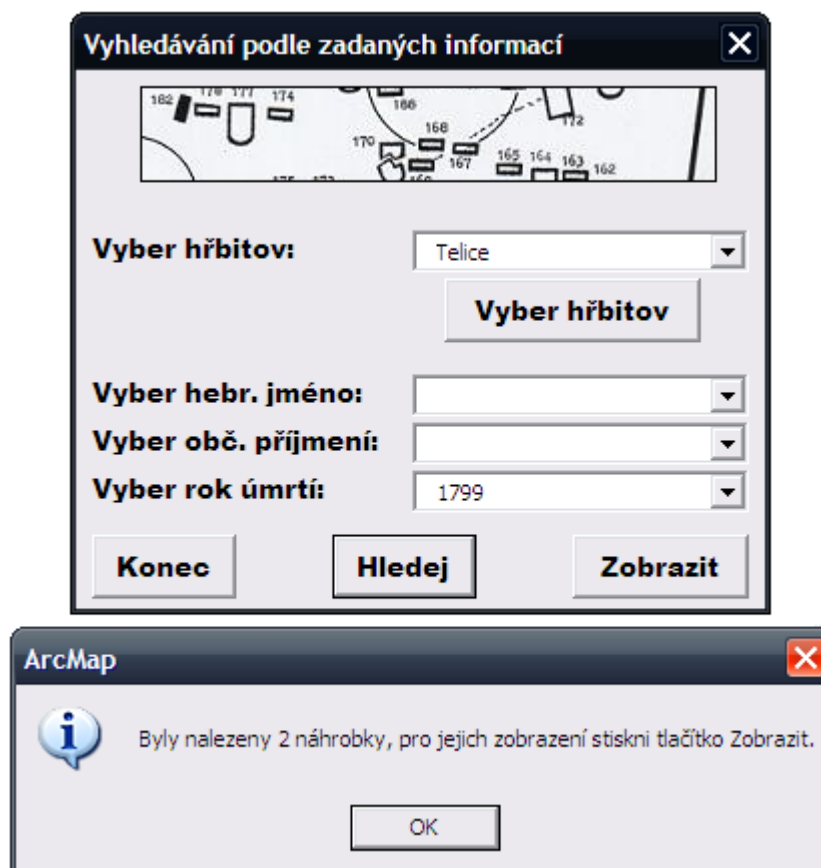
pEnvelope.Union pFeature.Extent
Set pFeature = pEnumFeature.Next
Loop

If pEnvelope.XMin = pEnvelope.XMax Then
'Expand the envelope around a single point
  pEnvelope.XMin = pEnvelope.XMin - 20
  pEnvelope.YMin = pEnvelope.YMin - 20
  pEnvelope.XMax = pEnvelope.XMax + 20
  pEnvelope.YMax = pEnvelope.YMax + 20
Else: pEnvelope.Expand 2.1, 2.1, True
End If
End If

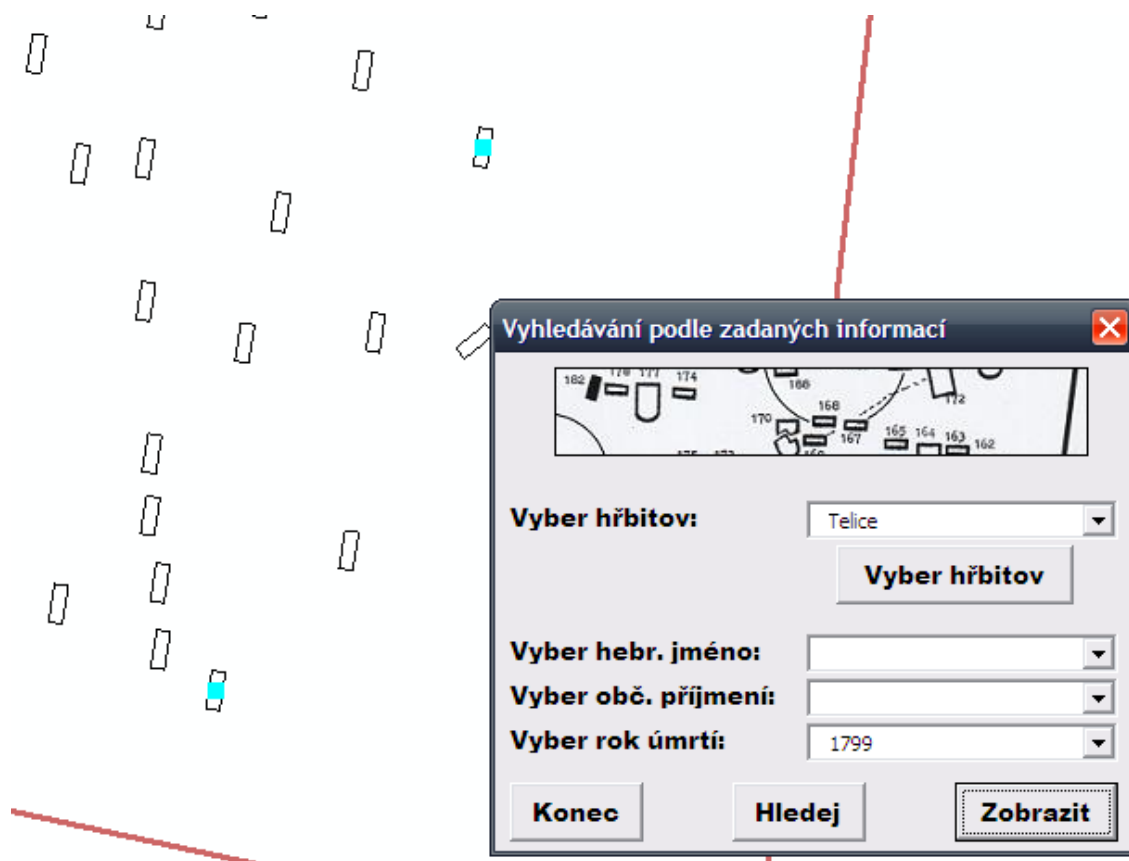
'Set map's extent to the selection's extent
Set pActiveView = pMap
pActiveView.Extent = pEnvelope
pActiveView.Refresh
End Sub

```

Příklad výsledné vizualizace tohoto formuláře po nalezení zadaných prvků je uveden na obrázku 4.13, kde je vidět, že na židovském hřbitově Telice se nachází dvě osoby, které zemřely v roce 1799. Následné zobrazení vybraných prvků je patrné z obrázku 4.14.



Obr. 4.13: Ukázka formuláře po nalezení zadaných údajů



Obr. 4.14: Ukázka zobrazení nalezených údajů

5 Závěr

Diplomová práce, jak bylo zmíněno v úvodu, je pokračováním bakalářské práce [1], v jejímž závěru byl stanoven cíl pro rozšíření vytvořené geodatabáze o datovou strukturu ukládající data o konkrétním náhrobku. V diplomové práci bylo nutné zpracovat vytvořenou geodatabázi do úplné třívrstvé architektury, která se skládá z datové, aplikační a prezentační vrstvy.

V návaznosti na bakalářskou práci byly tedy pro diplomovou práci stanoveny následující cíle:

- doplnění datového modelu o podrobný popis údajů uvedených na náhrobku,
- zjištění možností využití technologie ArcObjects pro návrh aplikačního rozhraní,
- návrh a implementace aplikačního rozhraní pro práci s datovou sadou židovských hřbitovů.

Byly splněny všechny cíle?

- datový model vytvořený v rámci bakalářské práce byl doplněn do datové vrstvy, a to atributovou tabulkou, která obsahuje strukturované údaje uvedené na konkrétním náhrobku. Jedná se o atributy:
 - hebrejské jméno,
 - hebrejské jméno otce,
 - občanské jméno,
 - občanské příjmení,
 - a datum úmrtí,
- po prozkoumání jednotlivých nástrojů dostupných v rámci grafického uživatelského rozhraní aplikace ArcMap bylo zjištěno, že pro tvorbu aplikačního rozhraní používaného pro práci s datovou sadou židovských hřbitovů nejsou tyto nástroje dostačující. Proto bylo nutné za použití ArcObjects vytvořit vlastní nástroje, které umožní rychlý přístup k požadovaným funkcím. V rámci této části práce byla vytvořena stručná „příručka“, která uvede uživatele do problematiky používání ArcObjects,
- tato část práce se zabývala doplněním aplikační a prezentační vrstvy. Návrh aplikačního rozhraní byl stanoven po konzultaci se zadavatelem. Při implementaci byl původní návrh upraven do podoby, která je pro uživatele dostatečně intuitivní.

Volba prostředí pro tvorbu aplikačního rozhraní:

- webová aplikace
- nebo ESRI?

Při volbě prostředí pro tvorbu aplikačního rozhraní bylo nejprve uvažováno prostředí webové aplikace. Ta byla ale zamítnuta zadavatelem, kvůli zneužití dat spojeným se stále aktuálním problémem vandalství na židovských hřbitovech. Protože se jedná o poskytnutí dat Národnímu památkovému ústavu, který používá centrální Geografický informační systém založený na technologii ESRI, bylo pro implementaci nakonec zvoleno prostředí VBA editoru a přímé spojení s aplikací ArcMap od firmy ESRI.

Pro volné poskytnutí dat se nabízela tvorba projektu v aplikaci ArcReader. Tato aplikace je dostupná zdarma, bez nutnosti placení licenčních práv. Bohužel ale aplikace ArcReader podle [8] nepodporuje většinu potřebných *CoClasses* ani tvorbu vlastních nástrojů (*ToolControls*).

Vytvořené aplikační rozhraní je dostatečně intuitivní a umožňuje uživateli prohledávání geodatabáze na základě zadaných parametrů. Vyhledávání je umožněno jak na jednom konkrétním, tak na všech židovských hřbitovech zahrnutých do geodatabáze. Uživateli bylo také umožněno zobrazení dokumentů obsahujících fotografie, přepise a překlade nalezených nárhobků.

Literatura

- [1] REINWARTOVÁ, Lenka. Geografická *datová sada židovských hřbitovů*. Vedoucí práce Ing. Karel Jedlička. 2008 [online].
URL:<http://gis.zcu.cz/studium/dp/2008/Reinwartova_Geograficka_datova_sada_zidovskych_hrbitovu_BP.pdf>.
- [2] *EDN: Esri Developer Network*. [online].
URL:< <http://edn.esri.com/index.cfm?fa=home.welcome>>.
- [3] ESRI. *Customizing ArcGIS Desktop using VBA*. [online].
URL:< http://resources.esri.com/help/9.3/arcgisdesktop/com/vba_start.htm>.
- [4] Burke, R.. *Getting to know ArcObjects: Programming ArcGIS with VBA*. ESRI. New York 2003. ISBN: 1-58948-018-X.
- [5] Kang-Tsung Chang. *Programming ArcObjects with VBA: A Task-Oriented Approach, 2. vydání*. CRC Press, New York 2008. ISBN: 978-1-580-1.
- [6] CHVÁTAL, Václav. *Mapy židovských hřbitovů, 2. vydání*. Muzeum Českého lesa, Tachov 2007.
- [7] Wikipedia. Internetová encyklopedie. [online].
URL: <http://en.wikipedia.org/wiki/Main_Page>
- [8] Customizing MXDs for Use with ArcIMS and ArcReader. [online]
URL:
<<http://edndoc.esri.com/arcobjects/8.3/TechnicalDocuments/CustomizingMxdsforArcIMSandArcReader.htm>>
- [9] See if Files, Folders and Drives Exist – Visual Basic 6.0. [online]
URL: <http://www.martin2k.co.uk/vb6/tips/vb_1.php>
- [10] ESRI Discussion Forums. [online]
URL: <<http://forums.esri.com/forums.asp?c=93>>

Adresářová struktura přiloženého DVD:

Jednotlivé složky a jejich obsah:

- [dokumentace_zh] – tato složka obsahuje dokumentaci židovského hřbitova Telice, kterou shromáždil, a pro účely této diplomové práce poskytl, PhDr. Ing. Václav Chvátal. Dokumentace se skládá z následujících složek:
 - D-LISTY – přepis a překlad textu náhrobku, zmenšenina fotografie náhrobku, formát PNG
 - D-LISTY_jpg – D-LISTY převedené do formátu JPG (pro použití ve VBA)
 - MAPA – mapa židovského hřbitova Telice vytvořená PhDr. Ing. Václavem Chvátalem
 - NAHROBKY – fotografie náhrobků, formát PNG
 - NAHROBKY_jpg – NAHROBKY převedené do formátu JPG (pro použití ve VBA)
 - REGISTR – needitované atributové záznamy o zemřelých osobách

- [zdrojove_kody] – složka obsahuje zdrojové kódy jednotlivých vytvořených tříd a formulářů:
 - [classes]:
 - FExists*: kontroluje existenci volaných soubřů
 - JmHebrejskeExists*: kontroluje existenci volaných atributových údajů
 - JmHebrExistsBezHrbit*: kontroluje existenci volaných atributových údajů
 - JmObcanskeExists*: kontroluje existenci volaných atributových údajů
 - JmObcExistsBezHrbit*: kontroluje existenci volaných atributových údajů
 - RokUmrtiExists*: kontroluje existenci volaných atributových údajů
 - RokUmrtiExistsBezHrbit*: kontroluje existenci volaných atributových údajů
 - ThisDocument*: obsahuje hlavní část kódu
 - Vyhledavani*: vyhledava náhrobky podle zadaných atributových údajů
 - VyhledavaniBzHrbit*: vyhledava náhrobky podle zadaných atributových údajů
 - ZoomTo*: zobrazení vyhledaných prvků

 - [forms]
 - DataNahrobku*: zobrazuje data o konkrétním náhrobku
 - Fotografie*: zobrazuje fotografii náhrobku
 - frmHledej*: formulář pro zadání hledaných údajů
 - frmHledejBezHrbit*: formulář pro zadání hledaných údajů
 - PrepisPreklad*: zobrazuje přepis a překlad náhrobku