VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

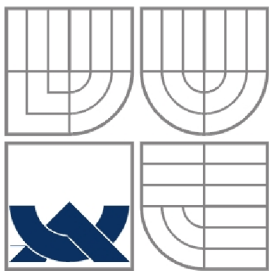
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

UNIVERZÁLNÍ CIZOJAZYČNÝ SLOVNÍK

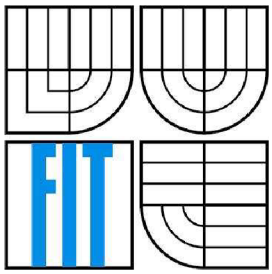
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE ONDŘEJ GRIM
AUTHOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

UNIVERZÁLNÍ CIZOJAZYČNÝ SLOVNÍK

THE GENERIC FOREIGN-LANGUAGE DICTIONARY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — ONDŘEJ GRIM
AUTHOR

VEDOUCÍ PRÁCE — Ing. LUKÁŠ GRULICH
SUPERVISOR

BRNO 2009

Zadání bakalářské práce/8833/2008/xgrimo00

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů Akademický rok 2008/2009

Zadání bakalářské práce

Řešitel: **Grim Ondřej**
Obor: Informační technologie
Téma: **Univerzální cizojazyčný slovník**
Kategorie: Databáze

Pokyny:

1. Analyzujte požadavky na univerzální slovníkovou aplikaci (rychlé přidávání a vyhledávání slov, co nejobecnější struktura - jazyky jsou velmi různé, atd.)
2. Seznamte se s dostupnými nástroji pro tvorbu takové aplikace, vyberte nejvhodnější.
3. Navrhněte vnitřní strukturu aplikace a uživatelské rozhraní.
4. Implementujte, naplňte vhodnými daty, otestujte.
5. Navrhněte možnosti dalšího vývoje.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Grulich Lukáš, Ing.**, UITS FIT VUT
Datum zadání: 1. listopadu 2008
Datum odevzdání: 20. května 2009

L.S.

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

Abstrakt

Tato bakalářské práce se zabývá návrhem a implementací univerzálního cizojazyčného slovníku – to znamená aplikace, která umožní překlad mezi dvěma libovolnými jazyky, které se nacházejí v přiložené databázi. Seznamuje čtenáře s danou problematikou a ukazuje její řešení v programové části této práce.

Abstract

This Bachelor thesis describes design and implementation of universal foreign language vocabulary. That means application which provils translation between two oferent languages, saved in attached database. It introduces reader with problematic and shows its processing in program part of this thesis.

Klíčová slova

C++, Qt, SQLite, grafické uživatelské rozhraní, GUI, databáze, slovník

Keywords

C++, Qt, SQLite, graphical user interface, GUI, database, dictionary

Citace

Grim Ondřej: Univerzální cizojazyčný slovník, bakalářská práce, Brno, FIT VUT v Brně, 2009

Univerzální cizojazyčný slovník

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Lukáše Grulicha. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Ondřej Grim
20. 5. 2009

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Lukáši Grulichovi, bez něhož by tato bakalářská práce nikdy nevznikla. Současně bych chtěl tímto způsobem poděkovat všem, kteří mě po dobu, kdy tato bakalářská práce vznikala, jakýmkoliv způsobem podporovali.

Obsah

Obsah 1
1 Úvod 2
2 Univerzální slovník 4
2.1 Historie a současný stav 4
2.2 Cíl práce 4
3 Teoretický základ 5
3.1 Funkce slovníku 5
3.2 Lingvistika 5
3.3 Jazyky 7
3.4 Kódování 9
3.5 Grafické uživatelské rozhraní 11
3.6 Databáze 14
4 Použité technologie 17
4.1 Qt 17
4.2 SQLite 17
5 Návrh 18
5.1 Databáze 18
5.2 Grafické uživatelské rozhraní 21
6 Implementace 28
6.1 Práce se soubory 28
6.2 Přístup k databázi 28
6.3 Průběh překladu 28
6.4 Řazení slov 30
6.5 Editace záznamů v databázi 31
6.6 Dynamická tvorba tabulek 34
7 Omezení a zhodnocení dalšího vývoje 35
8 Závěr 37

1 Úvod

Když se zamyslíme nad vývojem celé společnosti od počátků dějin až do dnešní doby, zjistíme, že celý svět se formoval do jednotlivých seskupení. Ty se později přetvořily ve státy a v nejrůznější národnosti, které měly svoji vlastní kulturu, své zvyky, náboženství a stejně tak i svůj jazyk.

Veškeré světové jazyky se formovaly do tzv. rodin – např. čínsko-austroasijská, indoevropská, afroasijská, aj. V Evropě se za nejznámější považuje především rodina indoevropská, která se dělí na jednotlivé skupiny jazyků. Bezesporu nejznámějšími jsou jazyky germánské, románské a slovanské. A protože člověk je ZOON POLITIKON, který myslí a ovládá řeč, je věcí samozřejmou, že vznikla základní potřeba komunikace. Jednotlivé komunity se nevyvíjely samostatně, ale navzájem se ovlivňovaly, žily vedle sebe a komunikovaly. Rozdílnosti jejich řeči však vedly k neustálým problémům a k pomalejšímu sdružování. A právě na základě členění společnosti do různých národností s různou kulturou vznikl také prvotní problém – způsob dorozumívání mezi jednotlivými komunitami. Postupně se začínala vytvářet potřeba zdokonalit tuto komunikaci a, pokud možno, naučit se komunikovat ve stejném jazyce.

Ovšem vznik jednotného jazyka je poměrně náročný a do jisté míry i těžce realizovatelný. Proto lidé hledali nějakou alternativu, jak se vzájemně dorozumět. Začaly vznikat soubory slov a jejich ekvivalenty v jiném jazyce. Výsledkem tak byly první překladové slovníky. Ovšem tyto slovníky měly stále určité nevýhody – zpočátku vznikaly pouze tištěné slovníky, které měly pevně danou slovní zásobu s absencí její změny a nemožností přidávání nových slov a nových překladů. Vždy bylo obtížné a časově náročné vyhledávat jednotlivá slova. Stejně tak slovník postrádal překlad celých vět (neuvažujeme-li fráze), včetně gramatického základu. V neposlední řadě musíme zohlednit i obtížnou přenositelnost – s detailností slovníku rostla i jeho objemová velikost. Z dalších nevýhod můžeme uvést i fakt, že vzniklé slovníky překládaly pouze mezi dvěma jazyky. S čím dál větší integrací světové populace vznikla také nová potřeba překládat nejen dva konkrétní jazyky mezi sebou, ale také možnost volby libovolného zdrojového i cílového jazyka. Z výše uvedeného se dá logicky vyvodit, že v tištěné podobě je zcela nemožné vytvořit tak rozsáhlý slovník, který by mezi sebou překládal větší množství jazyků.

Díky stále většímu rozvoji informačních technologií však mohla být tato potřeba uspokojena. Vznikaly nové technologie umožňující vytvářet elektronické slovníky. Ty do jisté míry nahradily slovníky tištěné, neboť odbouraly některé jejich nevýhody, jako je např. přenositelnost, omezenost a pomalé vyhledávání. V současné době se již začínají objevovat i snahy o vytvoření prvních slovníků univerzálních (např. slovník na Google.cz). Z tohoto důvodu se ve své bakalářské práci budu zabývat univerzálním cizojazyčným slovníkem, který jsem navrhnul a implementoval.

V první části této práce se zaměřím na teoretické předpoklady pro návrh takovéto aplikace a zmíním se o použitých technologiích.

Ve druhé části pak blíže popíši funkce a vlastnosti uživatelského rozhraní a detailněji rozeberu implementaci klíčových funkcí slovníku.

2 Univerzální slovník

V této kapitole se budu obecně zabývat cizojazyčnými slovníky. V krátkosti shrnu historii a současný stav elektronických slovníků a cíl této práce.

2.1 Historie a současný stav

První elektronické slovníky vznikly přibližně kolem roku 1994. Většinou šlo o jednoduché programy na textové bázi. Vzhledem k výkonu tehdejších počítačů však překlad trval až několik desítek sekund. Pravděpodobně první on-line slovníky vznikly současně s rozmachem internetu přibližně na přelomu let 1994/95. V současné době existuje již velká řada slovníků od různých vývojových studií. Ve většině z nich je však překlad z jednoho jazyka do druhého pevně dán. V poslední době se začínají objevovat snahy o vytvoření slovníků, u nichž je možné libovolně zvolit jak zdrojový, tak i cílový jazyk. Jako příklad uvádím již zmiňovaný slovník na Google.cz[1].

2.2 Cíl práce

Cílem této práce je navrhnout a implementovat univerzální cizojazyčný slovník, který by se dal využít pro překlad z jednoho libovolného jazyka do jiného. Slovník by měl být zcela editovatelný. Znamená to, že by bylo možné přidávat zcela libovolné nové jazyky i se všemi základními vlastnostmi pro ně definovanými, a to včetně přidávání slov do těchto jazyků i jazyků již v databázi existujících. Samozřejmostí by mělo být také zakomponování těchto nově přidaných údajů do překladového mechanismu.

[1] viz internetová adresa < http://www.google.cz/language_tools >

3 Teoretický základ

Návrh každého projektu je spjat se zjišťováním co největšího počtu informací o problematice, kterou se zabývá. Projekt postavený na základě smyšlených údajů a/nebo nedostatečných znalostí je v praxi nepoužitelný a tím pádem již předem odsouzen k neúspěchu.

3.1 Funkce slovníku

Slovník by se dal definovat jako abecedně seřazená slovní zásoba předem nespecifikovaného rozsahu, k níž může být přistupováno z různých hledisek. Z uvedené definice vyplývá, že je lze rozlišovat podle následujících kritérií [1]:

Rozsah slovníků:

- malé (do 10 000 hesel)
- střední (zhruba 50 000 až 60 000 hesel)
- velké (nad 60 000 hesel)

Přístup ke slovní zásobě:

- slovníky výkladové (jednojazyčné) - slovníky současného jazyka, historických období, etymologické a slovníky obsahující slovní zásobu profesních skupin
- slovníky překladové (vícejazyčné)

V rámci této bakalářské práce nás budou nejvíce zajímat slovníky překladové. Ty slouží pro překlad textu z jednoho jazyka do druhého. Obsahují mnohdy nejen překlad samotný, ale k jednotlivým slovům bývají také přidány i doprovodné informace, jako například komentáře, fráze, výslovnost (psaná a/nebo, v případě elektronické podoby slovníku, jako soubor s mluveným záznamem) a jiné…

Sestavováním slovníků se zabývá tzv. lexikografie (z řeckých slov lexikon = slovník a graphein = psát, česky také „slovníkářství“), která, je jednou z disciplín lingvistiky, vědy o přirozeném lidském jazyku a řeči.

3.2 Lingvistika

Lingvistika [2] je vědní disciplína. Zabývá se jazykem jak obecně, tak i jeho užitím. Je to věda o přirozeném lidském jazyku a řeči, někdy nazývaná též jazykověda. Název pochází z latinského „*lingua*“ = jazyk. Používá velkou řadu různých pojmů, mezi které patří například fón (konkrétní zvuk

představující určitou hlásku), foném (nejmenší zvuková jednotka odlišující jednotlivá slova mezi sebou), hláska (základní jednotka zvukové stránky řeči), morfém (předpona, přípona, vpona, koncovka nebo kořen slova, tzn. nejmenší viditelná část slova nesoucí jeho věcný nebo gramatický význam), lemma (základní podoba lexému, tzn. slova či fráze) a další.

Lingvistika má velmi bohatou historii a prošla dlouhým vývojem. První uvědomění si symbolického charakteru jazyka proběhlo již ve starověku, když lidé zjistili, že zvuky neodpovídají tomu, co označují, a že slova nejsou totéž, co věci (období třetihor – 5 mil. let ante). Později došlo také ke vzniku a rozvoji písma. První úvahy o jazyce pocházejí z Egypta (6000 ante), z Indie (800 – 400 ante) a z Číny (4. – 2. stol. ante). Zde se již jedná o vyspělé civilizace a rozvoj řeči zde souvisel především s jejím náboženským významem. Lingvistika se dělí do několika skupin, které se zabývají různými lingvistickými disciplinami:

Deskriptivní lingvistika – popisuje jazykový systém a způsoby jeho užití v různých komunikačních situacích. Zaměřuje se na vytváření slovní zásoby (slovníky) a na strukturu konkrétního jazyka (gramatiku).

Teoretická lingvistika – zkoumá obecné principy fungování přirozeného lidského jazyka, přičemž často využívá výsledky deskriptivní lingvistiky. Cílem je stanovit obecné teorie a hypotézy, které jsou dále testovány s pomocí dat z konkrétních jazyků.

Aplikovaná lingvistika – využívá poznatků deskriptivní a teoretické lingvistiky. Do tohoto odvětví spadá jazyková terapie (např. logopedie), výuka cizích jazyků nebo překlady.

Preskriptivní lingvistika – je zvláštním odvětvím lingvistiky, neboť se nezabývá přímo lidským jazykem, ale jazykem spisovným. Snaží se charakterizovat spisovný jazyk jako soubor jazykových prostředků, které jsou nebo měly být používány ve školách, v oficiálních sdělovacích prostředcích nebo v situacích formálního charakteru. Vytváří jazykové příručky (u nás se jedná o Pravidla českého pravopisu).

Jazyk se skládá z řady podsystémů, které se mohou navzájem do určité míry ovlivňovat. Deskriptivní a teoretická lingvistika se tak dělí do následujících kategorií:

- **Lexikologie** - zkoumá slovní zásobu určitého jazyka a její užití. Slouží jako základ pro lexikografii, tzn. tvorbu slovníků.
- **Fonetika** – zkoumá zvukovou stránku jazyka z pohledu tvorby hlásek ve zvukovém ústrojí, jejich šíření a vnímání.
- **Fonologie** – zkoumá také zvukovou stránku, ale na rozdíl od fonetiky se zabývá funkcí hlásek.
- **Morfologie** – zabývá se tvaroslovím, tzn. ohýbáním a pravidelným odvozováním slov prostřednictvím přípon, předpon a vpon (tzv. morfémy).

- **Syntax** – jinými slovy větná skladba, se zabývá správným pořadím slov ve větě a vztahy mezi těmito slovy.
- **Sémantika** – nauka o významu jednotlivých jazykových výrazů (slov, morfénů, frází, vět i souvětí)
- **Pragmatika** – také pragmatická lingvistika nebo progmalingvistika. Zabývá se jednotlivými slovy s přihlédnutím na mluvčího jako takového, na jeho pohnutky a na konkrétní situaci. Mluvčí může například tónem projevu změnit význam celé věty (třeba věta „Tohle je opravdu užitečný vynález“ může být, v závislosti na kontextu, buď pochvalou, nebo vyjádřením pohrdání).
- **Textová lingvistika** – zabývá se otázkou tzv. propozicionálního pojetí textu, to znamená pravidly a principy pro spojování vět a souvětí do větších celků = textů.

3.3 Jazyky

Jazyk je nedílnou součástí každého národa, stejně jako vlastní území nebo vlajka. Ale co je to vlastně jazyk? Tento pojem by se dal definovat různými způsoby. Jednou z definic může i ta, že jazyk je soustavou konvenčních (ustálených) znaků a pravidel pro jejich užívání. Jazykový znak by se pak dal definovat jako zvuková stopa nebo grafická značka představující určitou část skutečnosti nebo obsahu mysli.

Každý jazyk měl svoji vlastní, mnohdy bouřlivou, historii, která zapříčinila jeho nynější podobu, případně jeho definitivní zánik, ke kterému došlo v průběhu dějin. O tomto vývoji se dále nebudu rozepisovat. Pouze podotknu, že na základě tohoto faktu vzniklo velké množství velice podobných i naprosto odlišných jazyků. Různé prameny uvádějí, že v současné době existuje přibližně kolem sedmi tisíc jazyků, včetně dialektů.

Porovnáme-li různé jazyky, zjistíme, že rozdílný vývoj neměl vliv pouze na poskládání jednotlivých písmen do různých celků (slov) majících v daném jazyce svůj logický význam, ale také zapříčinil vznik speciálních pravidel, podle nichž se tato jednotlivá slova uspořádávají do celých vět. Tím však nechci tvrdit, že výše uvedené platí pro každou dvojici jazyků. Danou skutečnost si demonstrujeme na modelovém příkladu:

Představme si skupinu jdoucích lidí:

1) „Jdeme do kina.“ – věta v českém jazyku vyjadřující aktuální situaci
2) „Ideme do kina.“ – věta ve slovenském jazyku popisující tu samou situaci
3) „We are going to the cinema“ – anglický jazyk, věta má identický význam

Nejdříve porovnejme první a druhou větu. Je zřejmé, že slovosled českého a slovenského jazyka je zcela identický a tyto dva jazyky se liší pouze v rozdílném zápisu některých slov.

Nyní se však zaměřme na znatelnější rozdíly mezi první a třetí větou. Pomineme-li fakt naprosté rozdílnosti všech slov, pak další znatelným rozdílem je, že počet slov v jedné větě (3) neodpovídá počtu slov ve větě druhé (6). Prvním rozdílem je, že české sloveso „Jdeme“ je v angličtině vyjádřeno spojením slov „We are going“, což v doslovném překladu znamená „My jsme jdoucí“. Druhým rozdílem je, že angličtina, oproti češtině, využívá pojení podstatných jmen se členy („the“ v ukázkovém případě).

Jako další možné gramatické rozdíly lze uvést například skloňování, výslovnost a způsob zápisu. Jazyky nejsou jednotné ani v otázce používaných znaků [4]. Vzniklo tak velké množství různých unikátních abeced a také abeced z nich odvozených. Nejrozšířenější abecedou na světě je latinka (používaná asi 2,5 miliardami lidí), která se stala základem velkého množství světových jazyků. Protože tyto jazyky obsahovaly různé hlásky, které se v latince jako takové nevyskytovaly, vzniklo velké množství „mutací“ této abecedy. Pak v každé, tímto způsobem vzniklé, abecedě přibyly znaky specifické pro daný národ, stát či dokonce celou oblast. Latinka se původně skládala z dvaceti znaků (viz Tabulka 1), nyní se však za ni běžně považuje anglická abeceda (viz Tabulka 2).

A	B	C	D	E	F	H	I	K	L
M	N	O	P	Q	R	S	T	V	X

Tabulka 1 : Původní latinka

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	Z

Tabulka 2 : Anglická abeceda

Pro zajímavost lze ještě uvést, že všechny jazyky vycházející z latinky používají arabské číslice (tzn. znaky 0 1 2 3 4 5 6 7 8 9). Abeceda českého jazyka vznikla také z latinky, do které byly přidány pro češtinu specifické znaky (viz Tabulka 3):

Á	Č	Ď	É	Ě	Í	Ň	Ó	Ř	Š	Ť	Ú	Ů	Ý	Ž
á	č	ď	é	ě	í	ň	ó	ř	š	ť	ú	ů	ý	ž

Tabulka 3 : Znaky specifické pro češtinu

V neposlední řadě musím uvést i spřežku „ch“ (velkými písmeny „CH“), která v českém jazyce vystupuje jako samostatné písmeno. V abecedním pořadí se vyskytuje mezi písmeny „h“ a „i“.

Jako příklad abeced, používajících (z drtivé většiny) zcela odlišné znaky, lze uvést například cyrilici (tzv. azbuku), hebrejské písmo, arabské písmo, perskou abecedu, ásámské písmo, bengálské písmo, dévanágarí, japonské a čínské znaky a mnoho dalších. Těmito abecedami se však již

dále zabývat nebudu. Myslím si, že porovnávání dalších jazyků a abeced je, v rámci této práce, kontraproduktivní a nedůležité. Hlavním cílem této podkapitoly totiž bylo pouhé poukázání na nestejnoměrnou rozdílnost mezi různými dvojicemi jazyků.

3.4 Kódování

O dané problematice by se dalo mluvit z různých směrů pohledu – můžeme ji pojmout ze směru kryptografického, transportního, kompresního a dalších. Budeme-li však k problematice přistupovat z pohledu práce s textem na počítači, dalo by se kódování (nebo také znakovou sadu) definovat jako párování určitého počtu bitů s tisknutelným, příp. i netisknutelným, znakem, přičemž počet bitů použitých na každý takovýto znak definuje právě použité kódování.

Než se dostaneme k přiblížení standardizovaných znakových sad, rád bych se zde ještě zastavil a ujasnil pojmy „tisknutelný“ a „netisknutelný“ znak, o nichž jsem se zmínil v posledních řádcích předcházejícího odstavce. Tisknutelným znakem je myšlen ten, který se po napsání zobrazí na zobrazovací jednotce (matematické operátory, závorky, čísla, písmena…). A naopak netisknutelným znakem je ten, který se po napsání na zobrazovací jednotce neobjeví, ale provede určitou operaci. Například znak BEL přinutí počítač „pípnout“, Carriage Return (CR) přesune kurzor na začátek řádku nebo Line Feed (LF) slouží k posunu kuzoru na nový řádek atd. V případě tisku na jehličkové tiskárně slouží Carriage Return k posunutí její tiskové hlavy k levému okraji papíru. Line Feed naopak slouží k pootočení válce tiskárny tak, aby tisková hlava mohla tisknout o jeden řádek níže. Některé z netisknutelných znaků lze použít i při vzájemné komunikaci více počítačů mezi sebou.

Nyní se dostáváme k seznámení s některými významnými standardy na poli kódování (znakových sad) [3]:

ASCII

Zkratka vytvořená z anglického názvu „*American Standard Code for Information Interchange*“, jedná se o první úspěšný standard, který byl zaveden v roce 1963. Pro kódování znaků využívá pouze sedmi bitů, což znamená, že dokáže pokrýt maximálně 2^7 = 128 pozic (resp. znaků). Osmý bit slouží jako kontrolní.

Rozložení ASCII tabulky:

- 0-31 = řídící (netisknutelné) znaky
- 48-57 = numerická čísla (0-9)
- 65-90 = velká písmena anglické abecedy (A-Z)
- 92-122 = malá písmena anglické abecedy (a-z)

- ostatní pozice byly zaplněny matematickými operátory, interpunkčními znaménky a speciálními znaky

V závislosti na rozšiřování i do jiných, než anglicky mluvících zemí, se objevil poněkud zásadní nedostatek daného konceptu. ASCII tabulka byla zcela zaplněna a již nebylo možné přidávat žádné nové znaky. Odpadla tak například možnost lokalizace.

Osmibitové kódování

Situaci částečně vyřešil až vznik 8mi bitových verzí znakových sad, který umožnil v jedné tabulce obsáhnout dalších 128 znaků, celkem tedy 256. Ani tabulka o 256ti prvcích nedokázala uspokojit vícejazyčnou, a stále se rozrůstající, komunitu počítačových uživatelů. Dopomohla však ke vzniku různých ad-hoc tabulek, které umožnily přidání speciálních znaků abecedy, byť třeba jen na lokální úrovni dané země. Tento fakt však způsobil, že jednotlivé tabulky jsou vzájemně nekompatibilní, resp. kompatibilní jsou pouze znaky na pozicích 0-127, jejichž rozložení bylo převzato právě z ASCII.

Mezi významnější znakové sady, které umožňovaly používat mimo jiné i znaky české abecedy, by se daly zahrnout například:

- **CP852** - používaná v MS-DOS = *Microsoft Disk Operating System*; kde byla označována jako „Latin2“. Užívá se pro středoevropské jazyky používající latinku, tzn. například pro češtinu, polštinu, rumunštinu a další.
- **ISO 8859-2** – kódování firmy ISO obsahuje 191 znaků latinky a tak umožňuje používat znaky typické pro bosenštinu, češtinu, dolnolužičtinu, hornolužičtinu, maďarštinu, polštinu, slovenštinu, slovinštinu a další jazyky tohoto typu.
- **Windows-1250** – od společnosti Microsoft, používaná v jejich operačních systémech Microsoft Windows. Mezi podporované jazyky patří například albánština, chorvatština, čeština a polština. Je téměř identická s výše zmiňovanou ISO 8859-2. Jediným rozdílem, v rámci tisknutelných znaků, je občasná záměna některých jejich pozic.
- **Kód Kamenických (KEYBCS2), KOI8-ČS** a další

Unicode

Na základě nedostatků výše zmíněných kódování vznikla potřeba vytvořit jednu univerzální tabulku, která by obsahovala všechny znaky všech známých abeced. V roce 1987 tak spatřil světlo světa projekt Unicode. V tom samém období představila svůj vlastní projekt také organizace ISO. Ten nesl název ISO 10646. A plánovaný cíl těchto projektů? Vytvořit právě výše zmiňovanou tabulku. Výsledkem byla dohoda organizace ISO s tvůrci projektu Unicode (kolem roku 1991), na základě které došlo ke vzniku dvou různých kódování (Unicode a ISO 10646) se stejným rozložením

všech znaků, přičemž přidávání dalších znaků bylo vzájemně koordinováno. V závislosti na tomto přístupu došlo ke vzniku dvou oddělených znakových sad, které obsahují znaky (snad) všech různých abeced a jsou stoprocentně kompatibilní.

Znaková sada Unicode v současnosti obsahuje více než 100 000 (konkrétně 100 507, dle aktuální verze Unicode 5.1.0.) různých tisknutelných znaků, přičemž každý z nich má svůj jednoznačný číselný kód a název. Něco přes sto tisíc prvků není hraničním číslem, ale Unicode lze ještě dále rozšířit. Skládá se totiž ze 17ti částí, přičemž každá má velikost 2B. Z výše uvedených hodnot vyplývá, že tato znaková sada obsahuje celkem 17 * (2 * 2^8) = 1 114 112 pozic.

Je vhodné se také zmínit o významu UTF. Zkratka vznikla ze slov „*UCS Transformation Format*“ a je jí označován způsob kódování řetězců Unicode/UCS do sekvencí bitů. Zkratka je vždy doplněna číslem, které daný způsob blíže specifikuje – například UTF-16 pro kódování řetězců využívá 16 bitů, tzn. 2 bajty.

Z této podkapitoly je jistě zřejmé, že vytvoření lokalizace programu do konkrétního jazyka (příp. do více, často velmi rozdílných, jazyků) není tak jednoduché, jak si většina běžných počítačových uživatelů, kteří nejsou seznámeni s danou problematikou, představuje.

3.5 Grafické uživatelské rozhraní

Uživatelské rozhraní by se dalo definovat jako prostředek umožňující komunikaci člověka se strojem. Je několik různých druhů těchto rozhraní – textové, hlasové, grafické, příkazový řádek a další. V závislosti na programové části bakalářské práce se zaměříme na grafickou podobu rozhraní. Právě ta je v ní totiž aplikována.

Grafické uživatelské rozhraní [5, 6] se mnohdy označuje zkratkou GUI (z anglického „*Graphical User Interface*“). Základním stavebním prvkem je okno, které obsahuje jednotlivé dílčí prvky jako jsou tlačítka, formuláře, popisky, posuvníky a další. K jeho ovládání se zpravidla používají polohovací zařízení, nejčastěji pak kombinace myši a klávesnice. Za zmínku však stojí, že pro pohyb kurzorem myši lze použít i jiná zařízení, než myš samotnou. V případě vhodného softwaru (příp. hardwarového modulu) lze totiž kurzor ovládat i za pomoci jiných polohovacích zařízení jako jsou například gamepad, joystick, příp. ovladač od herní konzole Nintendo Wii. Pro jeho ovládání lze využít dokonce i ovladač od videa.

Vzhledem k faktu, že se v kapitole 5.2 budeme zabývat konkrétním návrhem grafického uživatelského rozhraní, vysvětlíme si zde všechny základní pojmy [5, 6], k jejichž definicím se ve výše zmiňované kapitole již nebudeme vracet.

Layout

Pro uživatele neviditelný prvek uživatelského rozhraní. Dal by se taktéž definovat jako kontejner určující rozložení jednotlivých prvků do něj vložených a to v rámci konkrétního okna, kterému náleží.

Label (nápis)

Složí pro zobrazování různých textů a popisů v okně. Také se dá použít pro zobrazení textového výstupu aplikace.

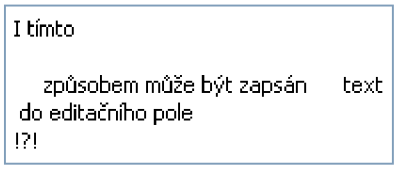
Push Button (tlačítko)

Jedná se o prvek (viz Obrázek 1), jehož použitím (např. kliknutím myši) dojde ke spuštění nějaké činnosti. Například se zobrazí nové okno aplikace, spustí se nějaký výpočet, atd.

Obrázek 1: Tlačítko

Edit Box, Text Edit (editační pole)

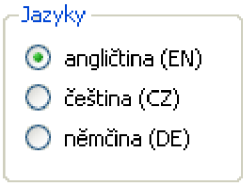
Své uplatnění nachází v místech, kde je požadováno zadání vstupních údajů od uživatele (viz Obrázek 2). Dá se však využít i pro vypsání výsledku. Jednou z vlastností editačního pole je i nastavení jeho editovatelnosti.



Obrázek 2 : Editační pole naplněné textem

Radio button(s) (rádiová tlačítka)

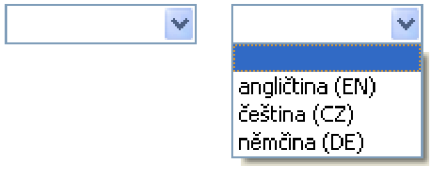
Mohou nastat situace, při kterých se po uživateli požaduje výběr z několika možností (jako příklad bohatě postačí otázka „Jsi muž nebo žena?“, přičemž odpověď na tuto otázku může být pouze řetězec „muž“ nebo „žena“). Tato situace se dá vyřešit pomocí editačního pole, do kterého uživatel celou svoji odpověď vypíše = méně uživatelsky příznivé řešení a vzniká nutnost přidání kontroly zadaného řetězce. Z těchto důvodů existují tzv. rádiová tlačítka. Je to vlastně množina tlačítek, kde je každé z těchto tlačítek opatřeno textem jedné z odpovědí (viz Obrázek 3). Uživatel je tak nucen vybrat jednu konkrétní nabízenou odpověď. Při označení druhé odpovědi ve stejné skupině rádiových tlačítek dojde ke zrušení odpovědi předcházející.



Obrázek 3 : Skupina rádiových tlačítek - výběr z několika možností

Combo Box (seznam s řádkem)

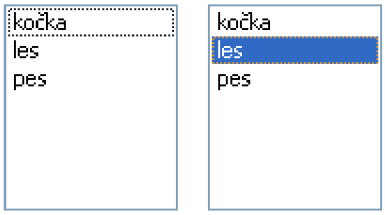
V případě, že nastane situace, kdy má uživatel na výběr s většího počtu možností (5, 10, 20,...), je použití rádiových tlačítek silně nevyhovující – s počtem různých možností totiž úměrně rostou požadavky na prostor pro tento prvek. Pro tyto případy slouží právě seznam s řádkem (viz Obrázek 4). Ten se jeví jako editační pole o velikosti řádku (přesná délka není, samozřejmě, nijak určena), na jehož konci se nachází šipka značící, že po kliknutí na tento prvek uživatelského rozhraní dojde k jeho rozbalení, čímž se zobrazí všechny nabízené možnosti. V podstatě je tak tento výběr zcela shodný s výběrem přes rádiová tlačítka.



Obrázek 4 : Nerozbalený (vlevo) a rozbalený (vpravo) seznam s řádkem

List Widget (seznam)

Seznam by šel teoreticky přirovnat ke „kombinaci s editačním polem a seznamu s řádkem“. Vzhledově totiž připomíná editační pole a pracuje se s ním stejně jako se seznamem s řádkem, až na ten rozdíl, že obsah daného seznamu je zobrazen neustále a ne až po kliknutí na něj (viz Obrázek 5).

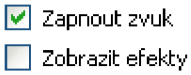


Obrázek 5 : Seznam s označeným (vlevo) a neoznačeným (vpravo) prvkem

Check Box (zatrhávátko)

Pro volby typu „ano“ x „ne“ (např. možnosti typu „Zapnout zvuk“ nebo „Spouštět při startu“) se v grafickém uživatelském rozhraní nachází tzv. zatrhávátko. Jedná se o malé políčko, které má

pouze dva stavy (zapnuto x vypnuto) a, při kombinaci s Labelem, je pro tyto případy mnohem přehlednější a efektivnější, než použití rádiových tlačítek nebo seznamu s řádkem (viz Obrázek 6).



Obrázek 6 : Zapnuté (nahoře) a vypnuté (dole) zatrhávátko

Různých prvků a jejich modifikací je mnohem více, než je ve výše uvedeném výčtu uvedeno. Jejich množství a nabídka se pak liší v závislosti na použitém GUI toolkitu.

Události

Grafické uživatelské rozhraní je řízeno tzv. událostmi (anglicky „*eventy*“). Princip je takový, že při spuštění aplikace dojde k inicializaci prostředí, které posléze přechází do stavu čekání na příchozí událost. V závislosti na ní se následně provede určitá operace, pro danou událost definovaná (např. „Přijde-li zpráva, že uživatel klikl na tlačítko „zobraz“, dojde k otevření nového okna“). Události jsou spojeny buďto s aktivitami uživatele nebo systému.

3.6 Databáze

Na úvod si je vhodné říct, co vlastně znamená pojem „databáze“. Asi nejlépe by šlo databázi definovat jako jisté množství údajů, vztahujících se ke konkrétní problematice. Databází tedy může být například seznam telefonních čísel, registr dárců krve, ale také i katalog zboží na skladě.

Tyto soubory informací vznikaly hlavně proto, že lidský mozek je schopen rychle a efektivně vstřebávat velké množství daných informací (např. obrazové vjemy), ale není schopen si toto množství informací zapamatovat do takové míry, aby je, po předem nespecifikované době, dokázal naprosto identicky zpět interpretovat. Dalším důvodem byla nutnost rychlého získávání různých informací z těchto dat (např. počet různých hudebních alb na skladě, stav klientova účtu, adresa pacienta, atd.).

V minulosti byla data skladována v papírové podobě v obrovských kartotékách. Vyhledávání konkrétních údajů bylo časově náročné, neboť „obsluha“ takovéto databáze musela, na základě poskytnutého klíče (například příjmení a jména), vyhledat konkrétní pořadač v určité části kartotéky a následně prolistovat všechny, např. abecedně seřazené, karty v ní obsažené, aby vyhledala tu kartu, v níž se nacházel hledaný údaj. Velké komplikace pak mohlo způsobit špatné zařazení karty při jejím posledním použití. Dále také modifikace údajů, nutná při některých situacích (např. změna trvalého bydliště osoby), byla poněkud nepraktická – záznam musel být přeškrtnut a vepsáním nahrazen záznamem aktuálním, nebo musel být v kartě vyměněn celý list s tímto údajem, přičemž tento list mohl obsahovat i údaje další, nezměněné, což vedlo ke kompletnímu přepsání daného listu. Současné

papírové archivy dat jsou postupně nahrazovány elektronickými databázemi, které odbourávají většinu nevýhod svých předchůdců. Datová základna (neboli databáze) by se dala definovat jako elektronická podoba výše zmiňovaných souborů informací. Databáze však není pouze systematicky a logicky uspořádaný shluk dat, ale její součástí jsou také programové prostředky pro přístup k datům a k manipulaci s nimi.

Za celou dobu existence elektronického zpracování dat vzniklo, z hlediska způsobu ukládání dat a vazeb mezi nimi, celkem pět různých databázových modelů [8]:

- **síťový** – první z modelů tohoto typu, který byl popsán ve zprávě „*The DBTG April 1971 Report* “
- **hierarchický** – vytvořen ve stejném období (kolem roku 1971). Na základě tohoto modelu byl firmou IBM vytvořen systém IMS.
- **relační** – počátky relačních databází se datují přibližně do období kolem roku 1970.
- **objektový** – 90. léta 20. století, model využívá přístupu uplatňovaného v objektově orientovaných jazycích. Jeho cílem bylo nahrazení relačních systémů.
- **objektově-relační** – původní záměry se objektových modelů nebyly úspěšně splněny, ale došlo ke vzniku kompromisu v podobě objektově-relační technologie.

Pravděpodobně nejrozšířenější je právě model relační a v závislosti na faktu, že součástí programové části bakalářské práce je právě relační databáze, bude se veškerý níže uvedený text vztahovat právě k tomuto modelu.

Aby bylo možné jakkoliv pracovat s databází, musí být součástí programových prostředků databáze i dotazovací jazyk nebo databáze musí být uzpůsobena tak, aby bylo možno pro tvorbu dotazů na tuto databázi použít SQL jazyka vyhovující standardu SQL-92. Součásti databázového jazyka (musí existovat prostředky pro) [9, 10]:

- **jazyk pro definici dat** – (z anglického „*Data Definition Language*“ = DDL) konstrukce sloužící pro popisy nových dat a jejich struktury
- **jazyk pro manipulaci dat** – (z anglického „*Data Manipulation Language*“ = DML) konstrukce umožňující práci s daty (přidávat / upravovat / mazat) a vytváření dotazů na ně
- **jazyk pro řízení dat** – (z anglického „*Data Control Language*“ = DCL) konstrukce zabývající se přístupovými právy

SQL

Deklarativní dotazovací jazyk SQL vznikl v sedmdesátých letech dvacátého století v laboratořích IBM [7]. Deklarativním se nazývá proto, že specifikuje, co se má provést, ale již dále neurčuje, jakým způsobem se má operace provést. Přízvisko „dotazovací“ získal SQL jazyk na základě toho, že slouží pouze pro manipulaci s daty. Jeho vlastnosti ho tak plně předurčují pro práci s databázemi.

A jak práce s deklarativním dotazovacím jazykem probíhá? Představme si situaci, kdy již máme vytvořenu prázdnou tabulku s názvem „jazyky“, která má sloupce „nazev“ a „kod“ (jedná se pouze o demonstraci základního použití SQL, proto nebudeme uvažovat různé typy sloupců). Do této tabulky můžeme vložit záznamy pomocí dotazu „INSERT“ a to hned dvěma způsoby:

1) INSERT INTO jazyky VALUES ('němčina', 'DE');
2) INSERT INTO jazyky (nazev, kod) VALUES ('angličtyna', 'EN');

Druhý způsob najde svoje uplatnění spíše v situacích, kdy vkládáme hodnoty pouze do konkrétních (ne všech) sloupců.

Nyní jsme si uvědomili, že ve slově „angličtina“ není „y“ a chceme tuto chybu napravit. Použijeme tak příkaz „UPDATE“, který slouží ke změně konkrétní hodnoty konkrétního záznamu:

UPDATE jazyky SET nazev = 'angličtina' WHERE nazev = 'angličtyna';

Za klíčovým slovem „WHERE“ se může místo „nazev = 'angličtyna'“ nacházet i „kod = 'EN'“. V případě, že hodnoty nejsou v rámci této tabulky unikátní a tímto způsobem by se mohly nechtěně změnit i další záznamy, můžeme určitý záznam blíže konkretizovat použitím klíčového slova „AND“, tzn. porovnáním obou hodnot, které se v záznamu nacházejí. Tímto dosáhneme následujícího tvaru dotazu:

UPDATE jazyky SET nazev = 'angličtina' WHERE nazev = 'angličtyna' AND kod='EN';

Dále se rozhodneme odstranit záznam o němčině. Budeme přitom uvažovat, že ani jeden ze sloupců neobsahuje pouze unikátní hodnoty:

DELETE * FROM jazyky WHERE nazev = 'němčina' AND kod='DE';

A na konec se rozhodneme odstranit úplně celou tabulku z databáze:

DROP TABLE jazyky;

To byl poslední dotaz, jehož funkci jsem se rozhodl demonstrovat. Jazyk SQL prošel dlouhým vývojem a výše uvedené řádky sloužily pouze k přiblížení jeho použití. V opačném případě popis jazyka vystačil na celou publikaci, jejíž tvorba však není obsahem práce.

Na závěr lze ještě uvést, že existuje celá řada databázových systémů (Oracle, Informix, DB2, PostreSQL a další), přičemž klíčová slova i syntaktický zápis se mohou lišit. Navíc některé systémy mohou obsahovat nové funkce, které pro jiný systém nemusí být definované - např. porovnáme-li SQLite a PostgreSQL, nemá SQLite definováno české řazení.

Použití v aplikacích

Vzhledem k faktu, že je vytváření SQL dotazů (zejména pro běžné uživatele) poněkud méně uživatelsky příznivé a časově náročnější (zejména v případě složitějších dotazů), bývá každá, v praxi používaná, databáze opatřena grafickým uživatelským rozhraním. To, na základě uživatelem zadaných vstupních dat a implementovaných pravidlech, sestavuje dotazy na databázi a zobrazuje jejich výsledky.

4 Použité technologie

V této kapitole si krátce představíme technologie, které byly použité při tvorbě této aplikace.

4.1 Qt

Qt [11] je jednou z knihoven, které slouží pro vytváření grafického uživatelského rozhraní [12]. Pod záštitou Trolltechu (nyní Qt Software [13]) vznikala již od roku 1994. Jedná se o knihovnu programovacího jazyka C++, ale existuje i pro několik dalších jazyků, včetně jazyků Pascal, Java, Perl, Python, PHP, Ruby a Ada. Mimo jiné jednou z jejich klíčových výhod je i multiplatformnost. Aktuální verze knihovny (4.5) vyšla začátkem března roku 2009 a její další vývoj stále pokračuje. Zajímavostí je, že v roce 2008 byl Trolltech koupen společností Nokia (na základě této situace, mimo jiné, došlo ke změně názvu vývojového týmu). Nokia tuto technologii začala hojně využívat ve svých produktech (oficiální stránky uvádějí, že na trhu je v současné době přes 15 milionů linuxových zařízení postavených právě na Qt). Qt Software umožnil také vznik vývojářských prostředí, které jsou primárně určeny pro práci s Qt (Qt Creator pro C++ a Qt Jambi pro Javu). A právě v Qt Creatoru vznikala programová část této bakalářské práce.

4.2 SQLite

SQLite [14] je relačním databázovým systémem obsaženým v relativně malé knihovně napsané v jazyce C. Na rozdíl od databází založených na principu klient-server není samostatným procesem, se kterým program komunikuje, ale jedná se o knihovnu přilinkovanou přímo k programu. Je v něm realizována většina SQL-92 (standardu pro SQL). V porovnání s jinými databázovými systémy má řadu omezení, mezi něž patří například absence cizích klíčů (FOREIGN KEY), datových typů (jedinou výjimkou je sloupec deklarovaný jako INTEGER PRIMARY KEY), českého řazení a mnoho dalších prvků, na které jsou vývojáři u běžných databází zvyklí [15]. Na druhou stranu však v sobě skrývá jednoduchost (menší komplexnost, databáze jako jediný soubor,…) a mnohdy i vyšší rychlost oproti konkurenčním databázovým systémům[2].

[2] Viz výsledky testů z oficiálních internetových stránek < www.sqlite.org/speed.html >

5 Návrh

V této kapitole se blíže seznámíme s návrhem databáze a grafického uživatelského rozhraní. Očekávejte však pouze okrajové přiblížení, jako například vysvětlení významu jednotlivých řádků databázových tabulek Odpovědi na hlubší otázky typu „Jak proběhne přidání slova“ a „Jakým způsobem probíhá překlad“ se dozvíte až v následující (šesté) kapitole.

5.1 Databáze

Databáze aplikace je obsažena v jediném souboru s příponou .s3db, z čehož je zřejmé, že byla vytvořena v SQLite verze 3. Je navržena pro co nejjednodušší a nejefektivnější přidávání jak jednotlivých slov, tak i celých jazyků. Navrženou strukturu databáze prezentuje následující ER diagram (viz Schéma 1):

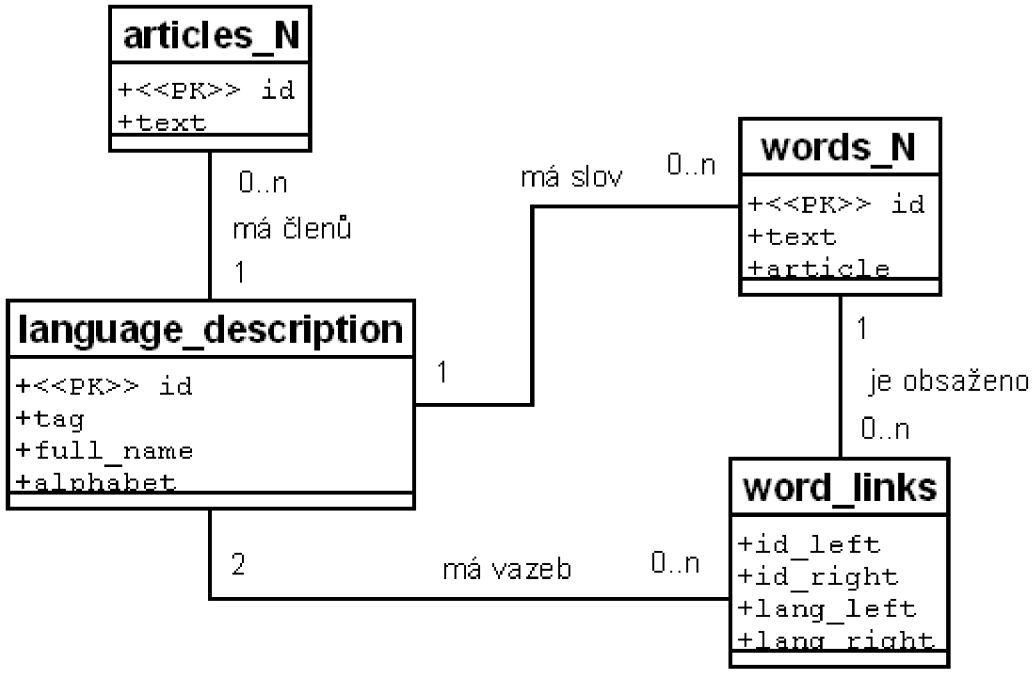


Schéma 1 : ER diagram databáze pro Univerzální cizojazyčný slovník

Z ER diagramu je zřejmé, jaké údaje obsahují jednotlivé tabulky a jaké vazby jsou mezi nimi. Je však nutné podotknout, že počet tabulek obsažených v databázi není přesně stanoven, ale lze jej vypočítat podle vzorce 2 + 2 * N, kde N značí počet jazyků v databázi obsažených – význam této skutečnosti je blíže popsán v kapitolách 6.3 a 6.5, zabývajících se konkrétním použitím databáze.

language_description

Slouží k uložení základních informací o všech jazycích, které se v databázi nacházejí, a k provázání jednotlivých jazyků a tabulek jim náležících. Vstupuje i do průběhu překladu. Jedná se tak o jednu z nejdůležitějších tabulek, nutných pro správný chod aplikace.

Význam jednotlivých sloupců:

Název sloupce	Typ	Význam
id	INTEGER	primární klíč pro přesnou identifikaci každého jazyka
tag	NVARCHAR	kódové označení jazyka
full_name	NVARCHAR	celý název jazyka
alphabet	NVARCHAR	znaky, které se mohou vyskytovat ve slovech

Tabulka 4 : Význam sloupců tabulky language_description

Příklad záznamu v tabulce:

id	tag	full_name	alphabet
0	EN	angličtina	AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

Obrázek 7 : Příklad záznamu v tabulce language_description

articles_N

Každý definovaný jazyk má svoji vlastní tabulku, kde jsou uloženy členy pro něj definované. Proměnná „N“ v názvu tabulky je číslem rovným konkrétnímu id jazyka ze záznamu z language_description (např. „articles_0“).

Význam jednotlivých sloupců:

Název sloupce	Typ	Význam
Id	INTEGER	primární klíč pro přesnou identifikaci každého členu
Text	NVARCHAR	textové vyjádření konkrétního členu

Tabulka 5 : Význam sloupců tabulky articles_N

Příklad záznamu v tabulce:

id	text
1	the
2	a

Obrázek 8 : Příklad záznamů v tabulce articles_N

words_N

Stejně jako členy, i slova pro každý jazyk, jsou uložena ve zvláštní tabulce. Taktéž zde je proměnná „N“ v názvu tabulky rovna konkrétnímu id jazyka ze záznamu z language_description (např. „words_0“).

Význam jednotlivých sloupců:

Název sloupce	Typ	Význam
Id	INTEGER	primární klíč pro přesnou identifikaci každého členu
text	NVARCHAR	textové vyjádření konkrétního členu
article	NVARCHAR	člen vztahující se na konkrétní slovo

Tabulka 6 : Význam sloupců tabulky words_N

Příklad záznamu v tabulce:

id	text	article
1	dog	the
2	cat	the
3	swim	

Obrázek 9 : Příklad záznamů v tabulce words_N

word_links

Aby bylo možné překládat slova z jednoho jazyka do druhého, musí existovat tzv. „překladová tabulka“, která definuje vazby mezi jednotlivými slovy z různých jazyků. K tomuto účelu slouží právě tato tabulka.

Význam jednotlivých sloupců:

Název sloupce	Typ	Význam
id_left	INTEGER	identifikační číslo slova ze zdrojového (primárního) jazyka
id_right	INTEGER	identifikační číslo slova z cílového (sekundárního) jazyka
lang_left	INTEGER	identifikační číslo zdrojového (primárního) jazyka
lang_right	INTEGER	identifikační číslo cílového (sekundárního) jazyka

Tabulka 7 : Význam sloupců tabulky word_links

Příklad záznamu v tabulce:

id_left	id_right	lang_left	lang_right
4	1	1	0
1	4	0	1

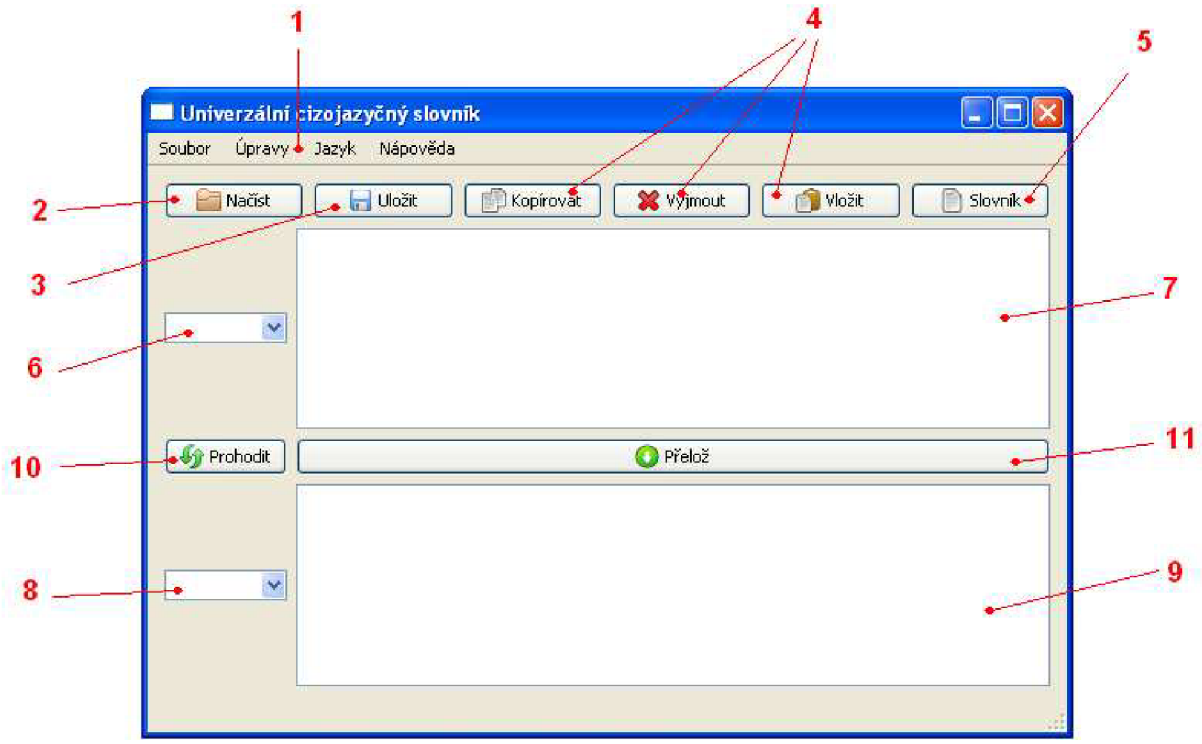
Obrázek 10 : Příklad záznamů v tabulce word_links

5.2 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní sestává z pěti samostatných dialogových oken, přičemž každé z nich má svoji specifickou funkčnost. Nyní si představíme jejich použití:

Překladač

Jedná se o hlavní okno, které se spustí bezprostředně po spuštění aplikace (viz Obrázek 11). Slouží k překladu textu ze zdrojového (dále primárního) jazyka do cílového (dále sekundárního) a k přístupu k dalším částem aplikace.



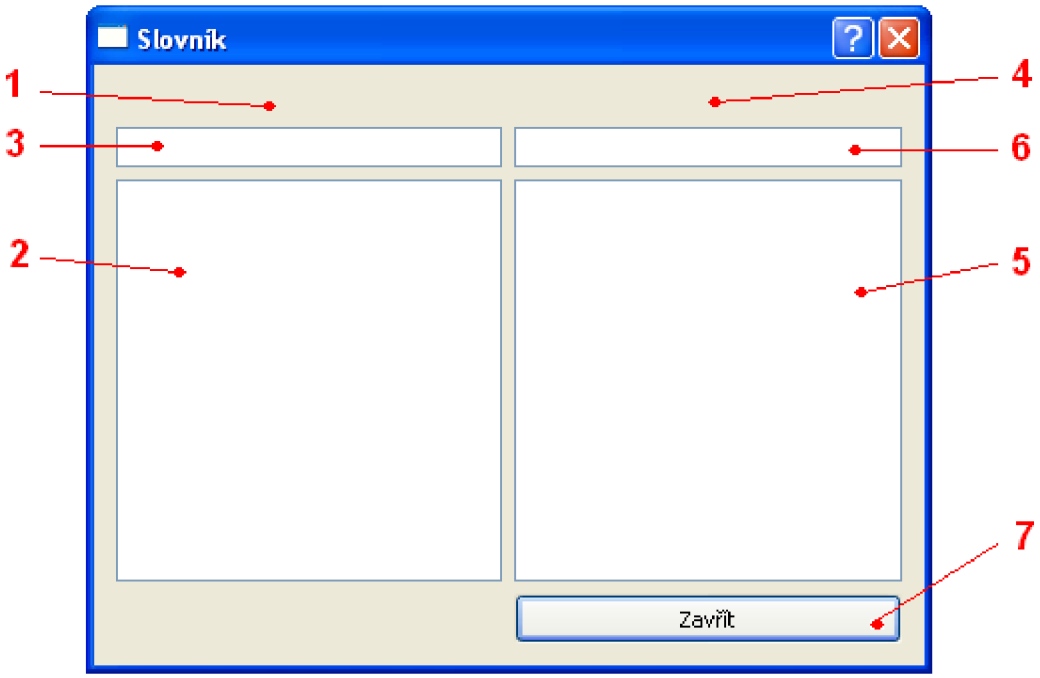
Obrázek 11 : Hlavní okno aplikace (překladač)

1) Hlavní menu obsahující základní funkce aplikace
2) Tlačítko pro načtení souboru obsahujícího text, který má být přeložen.
3) Tlačítko pro uložení přeloženého textu do souboru.
4) Tlačítka pro práci s textem.
5) Tlačítko pro zobrazení nového okna aplikace – slovníku.
6) Seznam s řádkem sloužící pro výběr primárního jazyka. Jednotlivé položky jsou v něm zobrazeny dle tvaru „název_jazyka (kódové_označení_jazyka)“, tzn. například „angličtina (EN)“. Umožňuje výběr ze všech jazyků, které se nacházejí v databázi aplikace. Tento seznam položek je abecedně seřazen dle názvu a to v závislosti na pravidlech českého pravopisu. Implicitně není vybrán žádný jazyk.
7) Tzv. primární editační pole slouží pro zadání zdrojového textu. V primárním poli se taktéž zobrazí text načtený ze souboru. Obsažený text se dá plně modifikovat – upravovat, kopírovat, vkládat i vyjímat.
8) Seznam s řádkem pro výběr sekundárních jazyků. Platí pro něj stejná pravidla jako pro předcházející Seznam s řádkem (viz bod 6).
9) Tzv. sekundární editační pole slouží pro vypsání přeloženého textu. Text obsažený v tomto poli není možné jakkoliv upravovat (dá se pouze kopírovat), ale lze jej uložit do souboru.
10) Tlačítko sloužící k prohození volby primárního a sekundárního jazyka.
11) Po stisknutí tlačítka „Přelož“ dojde, na základě volby primárního a sekundárního jazyka, k překladu zdrojového textu do cílové podoby. Ta je následně zobrazena v sekundárním poli.

V případě, že není vybrán některý z jazyků, je zadaný text do sekundárního pole pouze zkopírován.

Slovník

Tato část aplikace (viz Obrázek 12) se spouští pomocí tlačítka „Slovník“, které se nachází v hlavním okně aplikace – v překladači. Slovník je na překladači zcela závislý – konkrétně na volbě jazyků, na jejichž případnou změnu okamžitě reaguje. Mezi slovníkem a překladačem se dá volně přepínat – využití obou oken současně může usnadnit práci při překladu.



Obrázek 12 : Slovník

1) V této části se zobrazuje kódové označení primárního jazyka vybraného v překladači. Např. je-li v seznamu s řádkem sloužícímu pro nastavení primárního jazyka překladače vybrána položka „angličtina (EN)“, bude na tomto místě zobrazen text „EN“. Defaultně je tato oblast prázdná.
2) Seznam slouží k zobrazení slov vybraného primárního jazyka. Jednotlivá slova je možné označit stisknutím tlačítka myši. Po vybrání některého z nich dojde k jeho překladu, který se poté zobrazí v pravém (sekundárním) seznamu (viz bod 5). Totéž platí i obráceně, a proto se v tomto poli mohou nacházet i překlady slov sekundárního jazyka. Veškeré položky zobrazené v tomto poli jsou vždy abecedně seřazeny v závislosti na pravidlech českého pravopisu.
3) Dané editační pole slouží jako filtr levého (primárního) seznamu. Není-li tento řádek prázdný, jsou ve výše zmíněném seznamu zobrazena pouze ta slova, jejichž začátek je roven řetězci v tomto poli obsaženém.

4) 5) 6) Komponenty mají zcela identické vlastnosti. Jediným rozdílem je, že se vztahují k sekundárnímu jazyku.

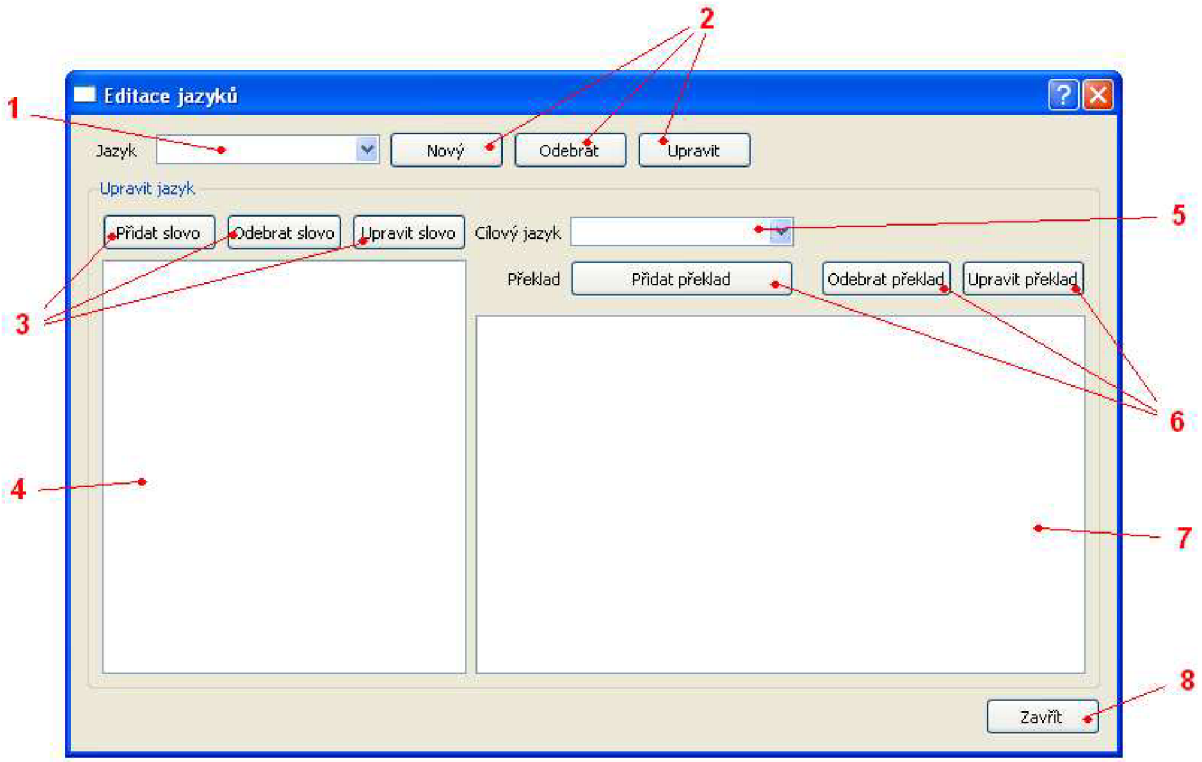
7) Tlačítko slouží pro zavření okna slovníku.

Editace jazyků

Do editace jazyků lze přistoupit přes volbu „Editace jazyků“ nabídky „Jazyk“ v hlavním menu aplikace. Slouží k veškerým úpravám databáze jazyků a slov, používané touto aplikací (viz Obrázek 13). Volba primárního a sekundárního jazyka je zcela nezávislá na volbě jazyků v překladači. Změny prováděné v tomto okně jsou ihned uplatňovány. Proto jsou po jeho uzavření volby jazyků v překladači (a tím pádem i data ve slovníku) nastaveny na defaultní = prázdné. Editace jazyků je modálním oknem, což znamená, že od jeho otevření až do jeho zavření není možné pracovat s žádným jiným oknem aplikace.

Než si přiblížíme rozložení komponent a jejich význam, měli bychom si ujasnit některé důležité pojmy:

- **Jazyk** = množina konkrétních slov pro něj definovaná
- **Slovo** = řetězec hlásek (někdy obsahující i pouze jednu hlásku) tvořící ustálený celek, bez jakýchkoliv vazeb na slovo v jiném jazyce, než v jazyce, pro nějž je definován
- **Překlad** = slovo, pro nějž je definována vazba na jedno nebo více slov, a to v jednom nebo více jazycích

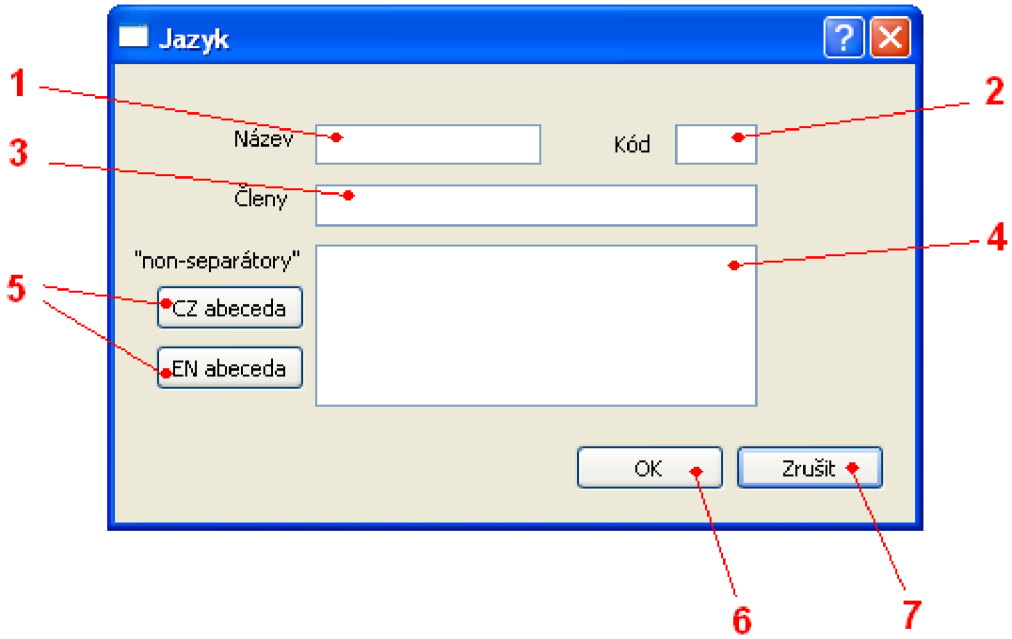


Obrázek 13 : Editace jazyků

1) Seznam s řádkem pro výběr primárního jazyka.
2) Tlačítka pro práci s jazyky: Tlačítko „Nový“ umožňuje vložit zcela nový jazyk, pomocí tlačítka „Upravit“ lze změnit hodnoty některého existujícího jazyka a tlačítko „Odebrat“ vybraný jazyk zcela odstraní z databáze. Zatímco tlačítka „Nový“ a „Upravit“ slouží k otevření dialogu pro úpravu již existujícího jazyka, tlačítko „Odstranit“ žádá pouze o potvrzení požadované operace.
3) Pomocí těchto tlačítek lze pracovat se slovy, tzn. přidat nové nebo odebrat, případně upravit, stávající. Zatímco tlačítka „Přidat slovo“ a „Upravit slovo“ slouží k otevření dialogu pro editaci slova, tlačítko „Odstranit slovo“ žádá pouze o potvrzení požadované operace.
4) Seznam, ve kterém se budou zobrazovat všechna slova vybraného primárního jazyka.
5) Seznam s řádkem pro výběr sekundárního jazyka.
6) Pomocí této trojice tlačítek lze pracovat s překlady vybraného slova z primárního jazyka do jazyka sekundárního, tzn. přidat nový nebo odebrat, případně upravit, stávající. Zatímco tlačítka „Přidat překlad“ a „Upravit překlad“ slouží k otevření dialogu pro editaci slova, tlačítko „Odstranit překlad“ žádá pouze o potvrzení požadované operace.
7) Seznam, ve kterém se budou zobrazovat všechny možné překlady slova z primárního do sekundárního jazyka.
8) Tlačítko sloužící pro zavření daného okna.

Editor jazyka

Editor jazyka (viz Obrázek 14) se zobrazí po stisknutí tlačítka „Nový“ (dialog se zobrazí zcela čistý) nebo „Upravit“ (jednotlivá pole dialogu budou naplněna hodnotami definovanými pro vybraný primární jazyk). Jedná se o modální okno aplikace.



Obrázek 14 : Editor jazyka

1) Editační pole pro zadání názvu jazyka (např. „angličtina“, „němčina“, atd.). Jedná se o povinný údaj.
2) Editační pole pro zadání kódu jazyka (např. „EN“, „DE“…). Je-li kód zadán malými písmeny, dojde k jejich konverzi na písmena velká. Jedná se o povinný údaj.
3) Do tohoto editačního pole lze zapsat členy pro konkrétní jazyk (např. „der“, „das“ pro němčinu nebo „the“, „a“ a „an“ pro angličtinu, atd.). Jednotlivé členy od sebe musí být odděleny čárkou a/nebo mezerou.
4) Dané pole slouží pro zapsání všech znaků, které se mohou ve slovech vyskytnout. Znaky je nutné zadávat „Case Sensitive“, tzn. vypsat malá i velká písmena. Jedná se o povinný údaj.
5) Aby bylo zadávání nového jazyka usnadněno, nacházejí se v tomto dialogu tlačítka, po jejichž stisknutí bude pole „non-separátory“ naplněno danou předpřipravenou abecedou. Vzhledem k jistému omezení, které bude uvedeno v části týkající se omezení a zhodnocení dalšího vývoje aplikace (kapitola 7), se zde nacházejí jen 2 abecedy (česká a anglická).
6) Tlačítko „OK“ slouží pro potvrzení zadaných údajů. Po jeho stisknutí dojde ke kontrole všech pravidel a omezení týkajících se přidání nového jazyka:
   - Pole „název“ nesmí být prázdné nebo nesmí obsahovat název jazyka, který se již v databázi nachází (tento fakt by způsoboval nepřehlednost aplikace)
   - Pole „Zkratka“ nesmí být prázdné nebo nesmí obsahovat zkratku jazyka, který se již v databázi nachází (tento fakt by způsoboval nepřehlednost a špatnou funkčnost aplikace)
   - Pole „non-separátory“ nesmí být prázdné (jazyk bez definovaných „non-separátorů“ by ve skutečnosti neměl definovánu svoji abecedu, což není z logického hlediska možné)

Jsou-li tato omezení splněna, je jazyk přidán do databáze, případně jsou, na základě zadaných hodnot a detekovaných změn, upraveny hodnoty již existujícího jazyka. V opačném případě se zobrazí varovná zpráva s výčtem všech objevených nesrovnalostí a editační dialog zůstane otevřený.

7) Tlačítko ukončí editor jazyka, aniž by došlo k jakýmkoliv změnám v databázi slovníku.

Editor slova

Editor slova (viz Obrázek 15) se zobrazí po stisknutí tlačítka „Nové slovo“ nebo „Nový překlad“ (dialog se zobrazí zcela čistý) a nebo „Upravit slovo“ nebo „Upravit překlad“ (jednotlivá pole dialogu budou naplněna hodnotami definovanými pro konkrétní slovo). Jedná se o modální okno aplikace.

Obrázek 15 : Editor slova

1) Editační pole pro zadání konkrétního slova. Jedná se o povinný údaj.
2) Editační pole pro zadání členu.
3) Po stisknutí tohoto tlačítka dojde k ověření, zda-li se již zadané slovo v daném jazyce nachází. Pokud ne, dojde k jeho přidání do databáze. V opačném případě se zobrazí varovná zpráva a editační dialog zůstane otevřený.
4) Tlačítko zavře editor slova, aniž by došlo k jakýmkoliv změnám v databázi slovníku.

6 Implementace

Se základními stavebními prvky aplikace jsme se již seznámili, a proto nastal čas, abychom se detailněji podívali na řešení jednotlivých klíčových funkcí.

6.1 Práce se soubory

Popisovaná aplikace umí, i když v omezené míře, pracovat se soubory. Konkrétně s prostými textovými dokumenty (např. soubory s příponou .txt) v kódování ANSI. Po načtení takového souboru (tlačítko „Načíst“ v horní části překladače – viz kapitola 5.2) je celý jeho obsah vložen do primárního pole překladače (viz kapitola 5.2).

Do prostého textového souboru lze také ukládat výsledný překlad (text ze sekundárního pole – viz kapitola 5.2). K tomu slouží tlačítko „Uložit“ nacházející se v horní části překladače (viz kapitola 5.2), po jehož stisknutí se zobrazí dialog pro určení názvu souboru a jeho umístění.

6.2 Přístup k databázi

Při překladu zdrojových kódů vstupuje do hry soubor s názvem libsqlite.a. Jedná se o statickou knihovnu, která v průběhu překladové fáze slouží k propojení dynamické knihovny, obsahující funkce definované nad SQLite3 databází (sqlite3.dll), a zdrojových kódů překládané aplikace. Pro správný chod přeložené aplikace není již libsqlite.a potřeba (ovšem sqlite3.dll ano).

6.3 Průběh překladu

Vzhledem k tomu, že se způsob překladu, až na základní myšlenku, v překladači a slovníku/editaci jazyků poněkud liší, rozeberu tuto funkci pro jednotlivé části zvlášť. Začneme jednodušší variantou, kterou je

- Průběh překladu ve slovníku a editaci jazyků

Nechť jsou v překladači vybrány oba jazyky (primární i sekundární), přičemž oba obsahují alespoň jedno slovo sobě vlastní, a je zobrazeno dialogové okno slovníku. Pak po vybrání některého prvku z levého (primárního) seznamu dojde k následujícímu sledu událostí:

1. z vybraného prvku je získán jeho text, a to pomocí funkce getText() nad tímto prvkem definované
2. na základě zjištěného identifikačního čísla primárního jazyka je vybrána konkrétní tabulka „words_N“

3. v této tabulce se vyhledá identifikační číslo konkrétního záznamu (id), v němž se vyskytuje text získaný z vybraného prvku
4. na základě znalosti primárního a sekundárního jazyka a získaného identifikačního čísla primárního slova dojde k vytvoření dotazu typu „SELECT“, jehož výsledkem bude seznam identifikačních čísel všech slov sekundárního jazyka, která mají v tabulce definovánu vazbu s daným primárním slovem
5. podle těchto čísel jsou z tabulky „words_N“, kde N je identifikační číslo sekundárního jazyka, vybrána konkrétní slova (překlady) pro vybraný prvek levého (primárního) seznamu (v případě, že některé překlady existují)
6. před vypsáním výsledných slov do pravého (sekundárního) seznamu (jsou-li nějaká) ještě dojde ke zjištění, zda-li je pro dané slovo definován člen. Pokud ano, je toto slovo opatřeno dvojicí znaků čárka+mezera (,) následované zjištěným členem, a až poté dojde k jeho vypsání do cílového seznamu.
7. Nyní se v sekundárním seznamu nacházejí všechny překlady vybraného prvku definované mezi primárním a sekundárním jazykem. Nejsou-li pro tento prvek definovány žádné vazby, bude cílové pole prázdné.

V případě slovníku funguje tento postup analogicky i pro výběr prvku ze sekundárního seznamu.

- Průběh překladu v překladači

V závislosti na výše zmiňované operaci by byla vhodná zmínka o jedné z funkcí jazyka C. Konkrétně o strtok(), která umožňuje, na základě předdefinovaných znaků (tzv. delimiterů), získat jednotlivá slova z řetězce, který je jí předán. Přesněji pracuje tak, že načítá jednotlivé znaky z řetězce, dokud nenarazí na znak, který je klasifikován jako „delimiter“. Jakmile se tomu tak stane, dojde k ukončení načítání a funkce vrátí dosud načtený řetězec (bez posledního načteného znaku).

Proč jsem se nyní o této funkci zmiňoval? Odpověď je jednoduchá. Aby bylo dosaženo stavu, kdy se text rozdělí na jednotlivá slova, a to v závislosti na „non-separátorech“ pro daný jazyk definovaných, musela by nastat jedna z následujících situací:

a) po uživateli by bylo požadováno, aby do kolonky „non-separátory“ (muselo by dojít k přejmenování na „separátory“) vypsal kompletní seznam všech znaků, které by sloužily jako „delimitery“ pro funkci strtok(). Ovšem vzhledem k poměru znaků, které je nutno vypisovat jako „non-separátory“ a znaků, které by musely být vypsány jako „delimitery“, by aplikace byla uživatelsky velice nepříznivá, což je nepřijatelné. A v případě, že bychom uvažovali znakovou sadu Unicode, bylo by vypsání všech „delimiterů“ doslova životním dílem. Nehledě na to, že takové množství dat by nešlo uložit do jednoho pole databáze. I kdyby toto

bylo možné, překlady postavené na základě hledání v takovém množství znaků, by byly pro dnešní běžné počítače nereálnou úlohou.

b) uživatel by byl nucen vypsat veškeré „non-separátory“. Následovalo by spuštění operace, která projde celou tabulku znaků a za pomoci jednoduché podmínky if-else, z uživatelem zadaného seznamu vytvoří seznam opačný. Při tomto způsobu by však vznikaly zbytečné režie a v případě použití znakové sady Unicode také naprosto nereálné úlohy (viz zmínka v bodu a))

c) musela by vzniknout speciální funkce se stejným základem jako již výše zmiňovaná strtok(), která by s „delimitery“ pracovala tzv. „reverzním způsobem“. To znamená, že by ze zadaného řetězce načítala jednotlivé znaky a to tak dlouho, dokud by se načítané znaky nacházely v „delimiterech“ (v daném případě by je bylo vhodné nazývat jako „non-separátory“). Uživatel by zadal jen tyto povolené znaky a vše ostatní, co by se mezi nimi nenacházelo, by bylo automaticky bráno jako nedovolené (slova-oddělující) znaky. Tímto způsobem by odpadly veškeré velké režie a zbytečné operace a navíc by byl udržen uživatelsky přijatelný přístup.

Na základě výše zmíněných poznatků došlo ke vzniku zvláštní funkce revstrtok(), která danou problematiku řeší tak, jak je uvedeno v bodě c).

Nechť jsou v překladači vybrány oba jazyky (primární i sekundární), přičemž oba obsahují alespoň jedno slovo sobě vlastní a v primárním poli se nachází předem nespecifikovaný text. Pak, po stisknutí tlačítka „Přelož“, dojde k následujícímu sledu událostí:

1. všechny znaky neuvedené jako „non-separátory“ budou vloženy do vektoru výsledných překladů
2. narazí-li funkce na „non-separátor“, dojde k načítání jednotlivých znaků do proměnné a to do doby, než funkce revstrtok() narazí na znak v „non-separátorech“ neuvedený
3. postup zjišťování překladu načteného slova je identický s postupem ve slovníku a editaci jazyků, a proto jej nebudu znovu uvádět (body 2 až 5)
4. je vybráno první slovo z výsledků překladu
5. dojde ke zjištění, zda-li je pro dané slovo definován člen. Pokud ano, je k tomuto slovu přidán. Vznikne tak výsledný tvar „člen slovo“
6. výsledek překladu je přidán do vektoru výsledných překladů. Pokud nebyl načtený celý text, vrací se funkce do bodu 1) daného postupu a pokračuje v dalším načítání. V opačném případě je výsledný vektor překladů vložen do tzv. sekundárního pole.

6.4 Řazení slov

Řadící algoritmus

Jelikož databázový systém SQLite nepodporuje řazení slov podle pravidel českého jazyka, je tato operace prováděna až v aplikaci, a to pomocí řadícího algoritmu s názvem Bubble sort. Jedná se

o řazení založené na průchodu polem a na porovnávání jednotlivých dvojic sousedních prvků. Pokud algoritmus odhalí dvojici, která není seřazena dle platných pravidel (vzestupné, příp. sestupné pořadí), jsou tyto dva prvky mezi sebou prohozeny. Seznam řazených údajů je opakovaně procházen, a to až do jeho úplného seřazení (tato situace nastane, pokud na konci aktuálního průchodu zůstanou všechny prvky na svých původních místech). Jednotlivé prvky tak doslova „probublávají" daným seznamem.

Bubble sort je stabilním (zachovává vzájemné pořadí prvků) a přirozeným (s rostoucím množstvím neseřazených prvků roste i časová náročnost) řadícím algoritmem, u kterého dochází k řazení záměnou (viz popis uvedený výše). Je implementačně jednoduchý. Ovšem z pohledu průběhu řazení neefektivní, a to z důvodu časové složitosti (v ideálním případě je složitost lineární, avšak v praxi obvykle kvadratická). Tím je předurčen spíše pro výukové účely a nebo pro řazení menšího množství dat.

Bubble sort byl vybrán z důvodů menší časové náročnosti jeho implementace a velikosti testovací databáze, u níž se (alespoň v této verzi programu) nepředpokládá, že bude obsahovat více než několik set slov pro každý konkrétní jazyk.

Vyhodnocení pořadí prvků

Součástí řadícího algoritmu musí být vždy nějaká funkce (nebo úsek kódu), která slouží ke zjištění, zda-li se porovnávané prvky nacházejí ve správném pořadí a nebo se musejí prohodit. Porovnávání dat se nejčastěji provádí za pomoci jednoduché podmínky typu „if" („Když je první hodnota větší než druhá, prohoď tyto hodnoty mezi sebou"). V případě normálních čísel je použití jasné. Ovšem při porovnávání znaků se pořadí těchto znaků určuje na základě jejich číselné hodnoty ve znakové sadě, a ne podle pravidel pravopisu určitého jazyka. Kolize v podobě špatného seřazení seznamu prvků tak vznikají při řazení malých a velkých písmen, písmen diakritikou a bez diakritiky, a také při zařazení písmene „ch" (to se řadí mezi řetězce začínající na „c", a to i navzdory tomu, že jeho správné místo je, v závislosti na pravidlech českého jazyka, mezi písmeny „h" a „i"). Tato komplikace je v aktuálním případě vyřešena použitím Qt funkce „fromLocaleAware()". Uvedená funkce jí předané prvky seřadí v závislosti na pravopisných pravidlech pro místní a jazykové nastavení operačního systému.

6.5 Editace záznamů v databázi

Jazyky

Nachází-li se uživatel v editoru jazyka, do kterého se dostal přes tlačítko „Nový" nacházející se v horní části Editace jazyků, tak po stisknutí tlačítka „OK" a po úspěšné kontrole, vysvětlené již v popisu uživatelského rozhraní, bude jazyk přidán následovně:

1. Do tabulky language_description je přidán záznam. Identifikační čísla (id) je číslem, pod kterým bude aktuálně přidávaný jazyk vystupovat.
2. Podle id se vytvoří dvě zcela prázdné tabulky příslušející tomuto jazyku (words_N a articles_N, kde N je výše zmiňované identifikační číslo).
3. Pokud byl do kolonky „Členy“ zadán nějaký text, dojde k jeho rozdělení na jednotlivá slova (slovo by se zde dalo definovat jako posloupnost znaků oddělená čárkou a/nebo mezerou) a jejich následné uložení do tabulky articles_N.

Chce-li uživatel změnit již existující jazyk, může použít tlačítko „Upravit“ nacházející se v horní části Editace jazyků. Po jeho stisknutí je zjištěno, který jazyk má být editován (v Editaci jazyků musí být vybrán primární jazyk) a na základě toho jsou kolonky Editoru jazyka naplněny příslušnými hodnotami. Kolonka „Členy“ je naplněna tak, že jsou postupně načítány záznamy z tabulky articles_N. Ty jsou od sebe odděleny dvojicí znaků čárka+mezera. Po stisknutí tlačítka „OK“ dojde ke zjištění změn, které poté budou zaneseny do databáze. Nebudou vytvářeny žádné nové tabulky, ale dojde k upravení již existujících záznamů.

Pokud uživatel, nacházející se v editoru jazyků, vybere primární jazyk a stiskne tlačítko „Odstranit“, zobrazí se dotaz obsahující varování, co se stane, bude-li jazyk opravdu odstraněn. Pokud uživatel svoji akci potvrdí, dojde k odstranění následujících záznamů:

- veškeré vazby z tabulky word_links, které jsou spjaty s odstraňovaným jazykem
- celá tabulka words_N, kde N je id odstraňovaného jazyka
- celá tabulka articles_N, kde N je id odstraňovaného jazyka
- záznam z tabulky language_description týkající se odebíraného jazyka (přesněji řečeno řádek s id rovným N, kde N je id odstraňovaného jazyka)

Slova

Přidat slovo lze přes editor slova, který se zobrazí pomocí tlačítka „Přidat slovo“. To se nachází v levé části Editace jazyků a je aktivní pouze v případě, že je v Editaci jazyků vybrán primární jazyk. Po stisknutí tlačítka „OK“ v zobrazeném editačním dialogu a po úspěšné kontrole, vysvětlené již v popisu uživatelského rozhraní, bude slovo přidáno do tabulky words_N, kde N bude rovno identifikačnímu číslu daného jazyka. Slovo nevytváří žádné vazby s jinými slovy ani jazyky.

Chce-li uživatel změnit již existující slovo, může použít tlačítko „Upravit“ nacházející se v horní části Editace jazyků. To je aktivní pouze v případě, že je vybráno slovo z primárního jazyka. Po jeho stisknutí je zjištěno, které slovo má být editováno, a na základě toho jsou kolonky Editoru slova naplněny příslušnými hodnotami. Po stisknutí tlačítka „OK“ dojde ke zjištění změn, které poté budou zaneseny do databáze a nahradí tak původní údaje.

Slovo lze odstranit jeho vybráním a následným stisknutím tlačítka „Odstraň slovo“. Zobrazí se varovný dialog upozorňující na následky odstranění slova. Po potvrzení akce dojde k odstranění všech záznamů z tabulky word_links spjatých s odstraňovaným slovem (záznamy vybrané na základě identifikačního čísla slova a identifikačního čísla jazyka, kterému náleží). Na závěr dojde k odstranění záznamu z konkrétní tabulky words_N.

Překlady

Přidat překlad lze přes Editor slova, který se zobrazí pomocí tlačítka „Přidat překlad“. To se nachází v pravé části Editace jazyků a je aktivní pouze v případě, že je v Editaci jazyků vybrán primární jazyk a v něm je označeno jedno slovo (dále primární slovo). Po stisknutí tlačítka „OK“ v zobrazeném editačním dialogu dojde ke kontrole, která určí, zda-li se slovo v daném jazyku nachází, či nikoliv:

- Pokud se slovo ve vybraném sekundárním jazyku nenachází, dojde k jeho vložení do tabulky words_N patřičného jazyka. Dále je zjištěno identifikační číslo tohoto slova. Stejná operace je provedena i pro primární slovo. Na základě získaných znalostí o primárním i sekundárním jazyku i slovu dojde k vytvoření dvojice příkazů „INSERT“, které do tabulky word_links vloží vazbu pro překlad dvou konkrétních slov z primárního jazyka do sekundárního. Totéž pravidlo následně vloží i pro překlad v obráceném pořadí (ze sekundárního slova do primárního)
- Je-li slovo v tabulce daného jazyka již obsaženo, je upuštěno od vkládání identického záznamu. Přidání vazeb dvojice slov proběhne stejným způsobem, který byl popsán výše, ovšem s tím rozdílem, že id nového záznamu je v příkazu nahrazeno identifikačním číslem již existujícího záznamu.

Vzhledem k faktu, že úprava překladu probíhá zcela identicky jako by šlo o úpravu primárního slova, nebudu daný průběh znovu opakovat.

Překlad lze odstranit jeho vybráním z pravého pole překladů a následným stisknutím tlačítka „Odstraň překlad“. Stejně jako při odstraňování slova, i zde je požadováno akci potvrdit. Oproti odstranění slova se však odstranění překladu velice liší. Nedojde k odstranění záznamu konkrétního slova, které v této situaci vystupuje jako překlad, ani k odstranění všech vazeb s ním. Z databáze bude odstraněna pouze vazba mezi primárním slovem a sekundárním slovem (překladem) - vazbou je myšlen vztah mezi oběma slovy, a to v obou pořadích (tzn. i pro případ, že dojde k prohození jazyků, čímž se změní i role primárního a sekundárního slova).

6.6 Dynamická tvorba tabulek

Při implementaci na úrovni databázové vrstvy jsou z hlediska tvorby tabulek možné tyto základní přístupy:

a) přístup, který lze nazvat statickým: navrhneme strukturu tabulek a jejich vzájemné vztahy, tabulky necháme v databázi fyzicky vytvořit a dále již strukturu tabulek neovlivňujeme. Výjimkou je samozřejmě fakt samotného dalšího vývoje aplikace, který si další změny na databázové úrovni může vyžádat.

b) přístup, který lze nazvat dynamickým: postupujeme jako v bodě a), ale navíc umožníme aplikaci v rámci naprogramované logiky vytvářet některé tabulky za jejího běhu.

Nevýhodou tohoto řešení jsou vyšší nároky na administraci databáze s takto tvořenými tabulkami - jakákoliv databáze má vždy přidělen omezený fyzický prostor pro uložení dat a ten může být v tomto případě poměrně rychle vyčerpán.

Výhody tohoto řešení leží v různých úrovních:

- v aplikaci se vyskytují pouze 4 různé druhy tabulek, které neobsahují žádné komplikované vazby.
- vzniknout mohou pouze tabulky, které nemají „řídící“ funkci
- přidání / odebrání tabulek probíhá v závislosti na přidání / odebrání záznamu tabulky s názvem „language_description“. Tabulky tak nevznikají bezdůvodně. Při odstranění záznamu z tabulky „language_description“ dojde k odstranění jak obou, pro tento záznam, vzniklých tabulek, tak i veškerých odkazů na ně i na jejich obsah. Z uvedeného vyplývá, že v žádném případě (uvažujeme-li zásahy do databázové struktury pouze prostřednictvím uživatelského rozhraní slovníku) nedojde k dotazu na neexistující tabulku.
- Takto vytvářená struktura databáze je uzpůsobena k použití i aplikacemi, pro které nebyla cílově vyvíjena. Příkladem je možnost prohlížení databáze pomocí tomu uzpůsobeného externího nástroje (SQLite Administrator, SQLite Maestro aj.)
- vyhodnocování dotazů nad tabulkami s menším počtem hodnot bez použití indexačních mechanismů nad těmito hodnotami je obecně rychlejší (pro tento případ uvažujme porovnání velikostně rozdílnějších tabulek – např. dvě tabulky čítající 500 a 60 000 záznamů)
- SQL dotazy jsou jednodušší a přehlednější (menší výskyt složených podmínek)

7 Omezení a zhodnocení dalšího vývoje

V poslední z kapitol, zabývajících se realizací programové části bakalářské práce, se zaměřím na různá omezení, ke kterým, z nějakých opodstatněných příčin, v aplikaci došlo. Také zde zmíním různé směry jejího dalšího vývoje.

Nejdříve se zmíním o jediném závažnějším omezení aplikace. Jedná se o vyřešení práce s kódováním. V pokročilejších stádiích implementace byly zjištěny komplikace při práci s databází, konkrétně při jejím plněním záznamy. Výsledkem bylo, že znaky, které se nenacházely v lokální osmibitové znakové sadě, se do databáze ukládaly jako otazníky. Daný problém nebyl úspěšně vyřešen a tak umí slovník pracovat pouze se znaky, které se v lokální znakové sadě nacházejí. Řešením tohoto nedostatku by mohlo být přepracování kódování textových řetězců, ke kterým v aplikaci dochází.

Dostáváme se tak k požadavkům na operační systém. Aplikace, kterou zde popisuji, je na operačním systému silně závislá a byla testována pouze na OS Microsoft Windows XP (konkrétně na jeho české lokalizaci). V případě použití jiného operačního systému, příp. jiného než českého jazykového nastavení (Místní a jazykové nastavení v Ovládacích panelech), není zaručena správná funkčnost aplikace (dle pravidel českého jazyka). Na výše zmiňovaném nastavení je závislá i znaková sada, která je použita pro data v aplikaci.

Aplikace by se dala obohatit o řadu dalších, ne však nezbytných, vylepšení. Jako první by se dalo uvést rozlišování přeloženého a nepřeloženého textu. Nynější způsob překladu by se dal popsat jako „vkládání přeložených či nepřeložených slov do vektoru, který je na konci celého překladového cyklu vypsán do výsledného okna“. Toto vylepšení by tudíž zvýšilo přehlednost výsledného textu.

Výsledný překlad by byl také dokonalejší, kdyby se uvažovala různá gramatická pravidla specifická pro každý jazyk. Vztáhneme-li tuto skutečnost na český jazyk, asi nejzákladnějším pravidlem, které by zlepšilo překladové schopnosti slovníku, by bylo skloňování. V závislosti na této skutečnosti by však bylo nutné do databáze ukládat slovní druhy a rody pro všechna slova. Následný překlad by byl zkomplikován nutností kontrolovat celé věty (resp. částí vět) a vkládáním správných tvarů přídavných jmen a sloves na základě rodů podstatných jmen, v těchto větách (resp. částech vět) obsažených. V aktuální verzi aplikace jsou slova překládána bez jakýchkoliv vazeb na kontextu věty nebo okolní slova.

Vzhledem k univerzálnosti slovníku by se dalo uvažovat o reálné šanci jeho uplatnění v zahraničí (tzn. ne jen v České republice). V závislosti na této skutečnosti lze jako jedno z vhodných dalších vylepšení uvažovat i vícejazyčné uživatelské rozhraní s možností výběru konkrétní podoby.

Aby se dala databáze slovníku snadněji a efektivněji plnit daty, určitě stojí za zmínku implementace pole, do něhož by se dal vložit internetový odkaz (příp. cesta v diskovém prostoru) na soubor určitého tvaru a obsahu, za pomocí kterého by se naplnila tabulka slov pro konkrétní jazyk

(words_N). Jelikož bylo grafické rozhraní vytvořeno spíše pro drobnější zásahy do databáze, vytvoření kompletní databáze slov pro konkrétní jazyk a jejich navázání na slova v jiných jazycích by, za pomoci funkcí uživatelského rozhraní, bylo doslova nereálné.

Aktuální verze aplikace dokáže načítat text obsažený v prostém textovém souboru (např. soubory s příponou .txt) a do tohoto typu souborů taktéž ukládat. Pokud by se tato aplikace měla vyrovnat komerčním verzím různých programů i z pohledu práce se soubory, rozhodně by se, jako o dalším směru vývoje, dalo uvažovat o rozšíření podporovaných textových souborů i na další typy (například soubory s příponou .doc).

V případě dalšího vývoje aplikace lze počítat také se zvětšením databáze slov. V takovém případě by se dalo uvažovat o přepracování, tzn. nahrazení, řadícího algoritmu (aktuálně Bubble sort) algoritmem rychlejším. Patrně nejlepší volbou by byl řadící algoritmus „Heapsort“. Tento dosahuje lineárně-logaritmické složitosti (pokud bychom si ji měli vyjádřit pomocí vzorce, pak O(N log N), kde N je proměnná popisující velikost vstupních dat) a má jen konstantní nároky na paměť.

Na úplný závěr této kapitoly je vhodné uvést, že „Univerzální cizojazyčný slovník“, který vznikl v rámci bakalářské práce, je spíše kostrou pro velice rozsáhlý a komplexní projekt a ne konkurenceschopnou aplikací.

8 Závěr

Cílem této bakalářské práce byla implementace cizojazyčného slovníku s možností úpravy databáze jednotlivých slov a jazyků. Tato aplikace by se jistě dala ještě mnoha způsoby zdokonalit a vylepšit. Dle mého subjektivního názoru jsem však hlavní cíl zadání splnil. Během práce na tomto slovníku jsem se naučil řadu nových věcí a získal jsem velké množství zkušeností a znalostí, které se mi zcela jistě budou hodit v budoucím profesním životě. Dospěl jsem také k názoru, že vytvářet cizojazyčný slovník je velmi náročné a vytvoření konkurenceschopného univerzálního slovníku je námětem na několik semestrů, resp. na více bakalářských prací.

Myslím si, že éra pevných předdefinovaných vazeb mezi dvěma konkrétními jazyky již postupně končí. Dalším milníkem vývoje slovníků je dosažení univerzálnosti a efektivního využití lexikálních pravidel každého jazyka.

Literatura

[1] Wikipedia: Slovník [online]. Aktualizováno 2009-05-09 [cit. 2009-05-09]. Dostupné na URL: < http://cs.wikipedia.org/wiki/Slovn%C3%ADk >

[2] Wikipedia: Lingvistika [online]. Aktualizováno 2009-05-09 [cit. 2009-05-09]. Dostupné na URL: < http://cs.wikipedia.org/wiki/Lingvistika >

[3] Wikipedia: Znaková sada [online]. Aktualizováno 2009-04-24 [cit. 2009-04-30]. Dostupné na URL: < http://cs.wikipedia.org/wiki/Znakov%C3%A1_sada >

[4] Wikipedia: Latinka [online]. Aktualizováno 2009-05-07 [cit. 2009-05-09]. Dostupné na URL: < http://cs.wikipedia.org/wiki/Latinka >

[5] Zemčík, P.: Slajdy pro předmět ITU (Tvorba uživatelských rozhraní) [online]. [cit. 2009-05-05]. Dostupné na URL: < https://www.fit.vutbr.cz/study/courses/index.php?id=5891 >

[6] Wikipedia: Grafické uživatelské rozhraní [online]. Aktualizováno 2009-04-24 [cit. 2009-05-05]. Dostupné na URL: < http://cs.wikipedia.org/wiki/Grafick%C3%A9_u%C5%BEivatelsk%C3%A9_rozhran%C3%AD >

[7] Wikipedia: SQL [online]. Aktualizováno 2009-01-16 [cit. 2009-05-02]. Dostupné na URL: < http://cs.wikipedia.org/wiki/SQL >

[8] Wikipedia: Databáze [online]. Aktualizováno 2009-02-22 [cit. 2009-05-03]. Dostupné na URL: < http://cs.wikipedia.org/wiki/Datab%C3%A1ze >

[9] Interval.cz: Databáze a jazyk SQL [online]. Aktualizováno 2000-08-04 [cit. 2009-05-03]. Dostupné na URL: < http://interval.cz/clanky/databaze-a-jazyk-sql/ >

[10] Zendulka, J.: Slajdy pro předmět IDS (Databázové systémy) [online]. [cit. 2009-05-02]. Dostupné na URL: < https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IDS-IT/lectures >

[11] Wikipedia: Qt (knihovna) [online]. Aktualizováno 2009-04-15 [cit. 2009-05-01]. Dostupné na URL: < http://cs.wikipedia.org/wiki/Qt_(knihovna) >

[12] Trolltech: Online Reference Documentation [online]. [cit. 2009-05-01]. Dostupné na URL: < http://doc.trolltech.com/ >

[13] Qt Software: Oficiální internetové stránky [online]. [cit. 2009-05-01]. Dostupné na URL: < http://www.qtsoftware.com/ >

[14] SQLite: Oficiální internetové stránky [online]. [cit. 2009-04-29]. Dostupné na URL: < http://www.sqlite.org/ >

[15] Root.cz: SQLite – ultra lehké SQL [online]. Aktualizováno 2003-08-04 [cit. 2009-04-29]. Dostupné na URL: < http://www.root.cz/clanky/sqlite-ultra-lehke-sql/ >

[16] Wikipedia: Řadicí algoritmus [online]. Aktualizováno 2009-04-02 [cit. 2009-05-11]. Dostupné na URL: < http://cs.wikipedia.org/wiki/%C5%98adic%C3%AD_algoritmus >

Seznam příloh

Příloha 1. CD se zdrojovými texty, podpůrnými materiály a spustitelnou verzí aplikace