

České vysoké učení technické v Praze
Fakulta elektrotechnická



Bakalářská práce

Aplikace pro zkoušení slovní zásoby

Miloslav Cinibulk

Vedoucí práce: Ing. Pavel Kubalík

Studijní program: Elektrotechnika a informatika strukturovaný bakalářský

Obor: Výpočetní technika

srpen 2007

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu mé bakalářské práce Ing. Pavlu Kubalíkovi za cenné připomínky při jejím řešení a dále lidem ve svém okolí za podporu.

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Neratovicích dne 22. 8. 2007

Abstract

The main purpose of this work is to create an application, which would help users practise English vocabulary. An application, which would help them widen and tighten vocabulary and which would thus contribute to adoption of such an important language in these days, as English really is.

Abstrakt

Hlavním účelem této bakalářské práce je navrhnout aplikaci, která by uživatelům pomohla procvičit anglickou slovní zásobu. Aplikaci, která by jim rozšířila a upevnila slovní zásobu a přispěla tak k lepšímu osvojení v dnešní době tak důležitého jazyka, jakým je právě angličtina.

Obsah

| | | |
|--------|---|----|
| 1 | Úvod | 1 |
| 2 | Analýza | 3 |
| 2.1 | Existující řešení | 3 |
| 2.1.1 | A-Test 4.0 | 4 |
| 2.1.2 | Interlex 2.5.0.6 | 5 |
| 2.1.3 | Angličtina V | 6 |
| 2.2 | Výběr způsobu implementace | 6 |
| 3 | Programovací model Windows | 7 |
| 3.1 | Zprávy systému Windows | 9 |
| 4 | Knihovna MFC | 10 |
| 4.1 | Linkování knihovny MFC | 10 |
| 4.2 | Mapa zpráv | 10 |
| 4.3 | Myš a klávesnice | 11 |
| 4.3.1 | Zachycení vstupu z myši | 11 |
| 4.3.2 | Zachycení vstupu z klávesnice | 11 |
| 5 | Návrh řešení | 12 |
| 6 | Implementace | 13 |
| 6.1 | Popis tříd | 14 |
| 6.1.1 | CMainFrame | 14 |
| 6.1.2 | CSlovníkView | 15 |
| 6.1.3 | CLrnDialog | 16 |
| 6.1.4 | CTstDialog1, CTstDialog2, CTstDialog3 | 16 |
| 6.1.5 | CWrtDialog | 17 |
| 6.1.6 | CEditDialog | 17 |
| 6.1.7 | CFindDialog1 | 18 |
| 6.1.8 | CFindDialog2 | 18 |
| 6.1.9 | CStatDialog | 18 |
| 6.1.10 | Ostatní třídy | 18 |
| 6.2 | Uživatelské zprávy | 19 |
| 6.3 | Ukládání | 19 |
| 7 | Testování | 20 |
| 8 | Závěr | 21 |
| 9 | Seznam literatury | 22 |
| A | Uživatelská příručka | 24 |
| A.1 | Funkce | 24 |
| A.2 | Vymezení pojmů | 24 |
| A.3 | Klávesové zkratky a funkční klávesy | 25 |
| A.4 | Hlavní okno | 26 |
| A.5 | Hlavní nabídka | 26 |
| A.5.1 | Soubor | 26 |
| A.5.2 | Zobrazit | 27 |
| A.5.3 | Zkoušení | 27 |
| A.5.4 | Nástroje | 33 |
| A.5.5 | Nápověda | 38 |
| 9.1 | Panel nástrojů | 39 |
| A.6 | Tabulka | 39 |
| A.7 | Stavový řádek | 40 |
| B | Obsah příloženého CD | 41 |

Seznam obrázků

| | |
|--|----|
| Obrázek 2.1: Aplikace A-Test 4.0 | 4 |
| Obrázek 2.2: Aplikace Interlex 2.5.0.6..... | 5 |
| Obrázek 2.3: Aplikace Angličtina V | 6 |
| Obrázek 3.1: Událostmi řízený programovací model systému Windows | 8 |
| Obrázek 6.1: Nový projekt | 13 |
| Obrázek 6.2: Provázání tříd | 14 |
| Obrázek A.1: Hlavní okno aplikace | 26 |
| Obrázek A.2: Exportování | 27 |
| Obrázek A.3: Učení pohled 1 | 28 |
| Obrázek A.4: Učení pohled 2 | 28 |
| Obrázek A.5: Test pohled 1 | 29 |
| Obrázek A.6: Test pohled 2 | 29 |
| Obrázek A.7: Test indikace | 30 |
| Obrázek A.8: Psaní | 30 |
| Obrázek A.9: Psaní indikace | 31 |
| Obrázek A.10: Statistika..... | 32 |
| Obrázek A.11: Hledání pohled 1 | 33 |
| Obrázek A.12: Hledání pohled 2 | 33 |
| Obrázek A.13: Přidávání | 34 |
| Obrázek A.14: Editování | 35 |
| Obrázek A.15: Nastavení karta Zkoušení..... | 36 |
| Obrázek A.16: Nastavení karta Zkoušení rozšíření..... | 37 |
| Obrázek A.17: Nastavení karta Program..... | 38 |
| Obrázek A.18: Panel nástrojů..... | 39 |
| Obrázek A.19: Tabulka | 39 |
| Obrázek A.20: Stavový řádek..... | 40 |

Seznam tabulek

| | |
|--|----|
| Tabulka 3.1: Obecné zprávy systému Windows | 9 |
| Tabulka 4.1: Zprávy z klientské oblasti okna..... | 11 |
| Tabulka 4.2: Virtuální kódy některých kláves..... | 11 |
| Tabulka 5.1: Uživatelské zprávy | 19 |
| Tabulka 7.1: Využití operační paměti 1 | 20 |
| Tabulka 7.2: Využití operační paměti 2 | 20 |
| Tabulka A.1: Klávesové zkratky a funkční klávesy | 25 |
| Tabulka A.2. Příklady správných odpovědí u Psaní..... | 31 |

1 Úvod

Znalost cizích jazyků je bezesporu jedním z předpokladů, jak být v dnešní moderní době úspěšný. Kdo chce například držet krok s nejnovějšími trendy v oblasti komunikace, nových technologií a informatiky, bez dobré znalosti angličtiny se neobejde.

Osvojení cizího jazyka nebývá pro každého snadné. Ne každý má přirozené nadání na jazyky, a tak se poměrně často setkáváme při výuce jazyků s „věčnými začátečníky“. Každému také nemusí vyhovovat výuka jazyků ve skupinkách, a proto raději sáhne po učebnici pro samouky. Jedním z předpokladů osvojení cizího jazyka je zvládnutí gramatiky. Dalším předpokladem je pak odpovídající slovní zásoba.

Cílem této bakalářské práce je navrhnout aplikaci určenou pro zkoušení anglické slovní zásoby, která bude běžet pod operačním systémem Microsoft Windows. Výsledná aplikace by neměla být zbytečně složitá, aby s jejím ovládnutím neměli problémy i méně zkušení uživatelé. Každý uživatel by měl mít možnost tvořit si vlastní slovníky (lekce), ze kterých by ho aplikace průběžně zkoušela, zaznamenávala dosažené výsledky a ty pak přehledně zobrazovala.

Práce je členěna na devět kapitol. První kapitola je úvod. Druhá kapitola se zabývá analýzou problému. Obsahuje přehled již existujících řešení a jejich zhodnocení. Závěr kapitoly se věnuje výběru způsobu implementace. Třetí kapitola stručně seznamuje s programovacím modelem systému Windows. Čtvrtá kapitola se pak věnuje knihovně tříd MFC. V páté kapitole je obecně pojednáno o návrhu řešení, na co byl kladen největší důraz, cíle. Šestá kapitola popisuje implementaci. Tedy obsahuje popis nejdůležitějších tříd použitých v projektu a jejich provázání. Sedmá kapitola se zabývá testováním aplikace. V osmé kapitole je napsáno, čeho bylo dosaženo, co se nepodařilo a celkové zhodnocení práce. Devátá kapitola obsahuje seznam použité literatury. Příloha A obsahuje uživatelskou příručku, která podrobně seznamuje se všemi funkcemi předkládaného řešení. Příloha B obsahuje pouze obsah příloženého CD.

2 Analýza

2.1 Existující řešení

Předmětem zadání je mimo jiné prostudování již existujících řešení problému. Nutno dodat, že kvalitních freewarových programů, které by uživatele procvičily z anglické slovní zásoby, mnoho není. Pro testování jsem si vybral aplikace A-Test 4.0, Interlex 2.5.0.6 a Angličtina V.

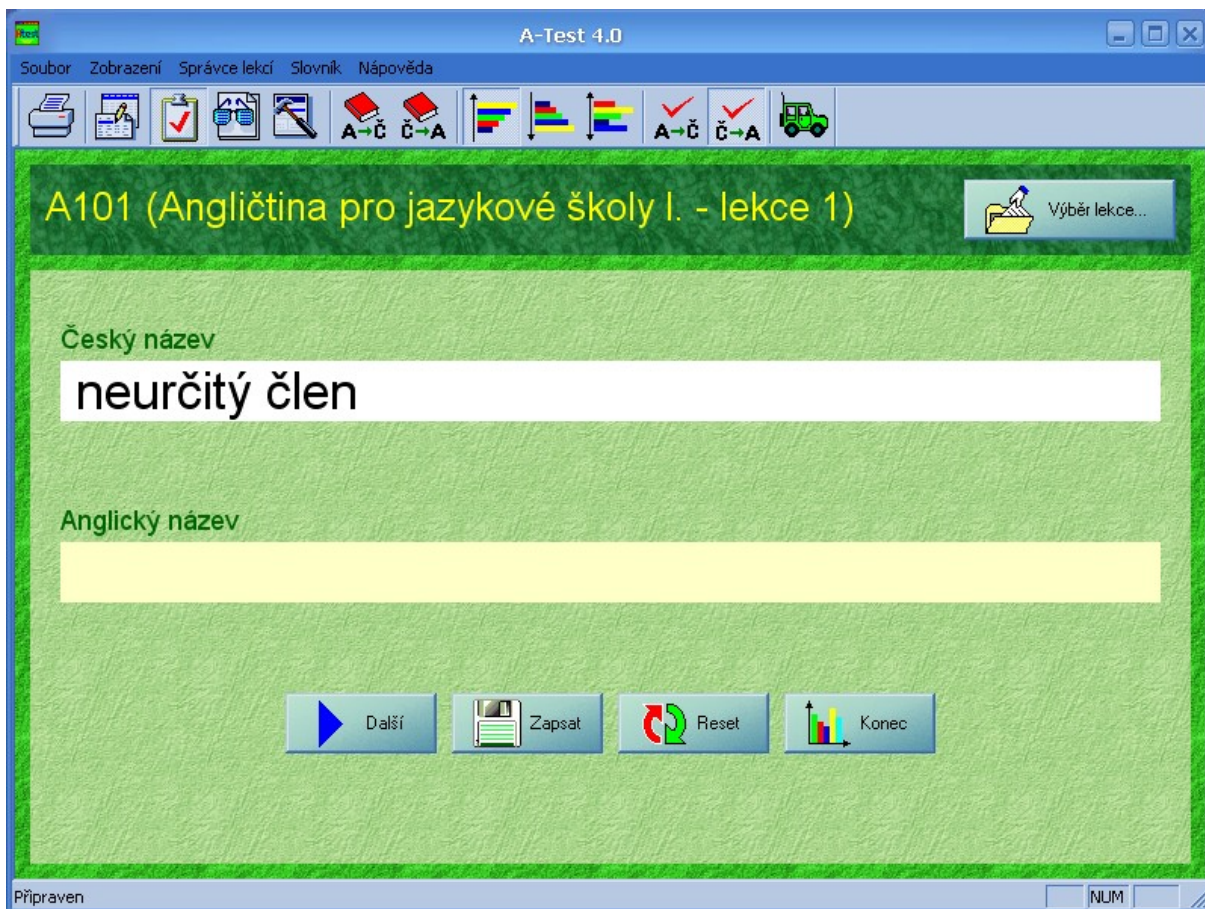
Ani u jednoho z testovaných programů jsem nebyl zcela spokojen s provedením zkoušení. Například učení u programu A-Test 4.0 je vyřešeno tak, že se uživateli stále zobrazuje jak český, tak anglický název právě zkoušeného slova. Jako lepší řešení považuji zobrazit nejdříve anglický název a poté, co si uživatel v klidu rozmyslí překlad, teprve český název (nebo naopak). Dále tento program narozdíl od programu Interlex 2.5.0.6 nerozlišuje ve slovníku slova oddělená čárkou nebo středníkem, což je při zkoušení, kde je úkolem napsat překlad, jistý problém, neboť je nutné napsat bezpodmínečně vše, včetně čárek a středníků. Tedy například na „that“, je-li to zadáno ve slovníku, je zde potřeba napsat přesně „tamten, tamta, onen, ona, ono, tamto, to“, což je prakticky nemožné. Lepší řešení je ve slovníku rozlišovat slova oddělená čárkou nebo středníkem a uznávat i odpověď například „tamto“, což je jistě také správná odpověď.

U programu Interlex 2.5.0.6 mi vadilo, že po špatné odpovědi je nutné stisknout tlačítko Continue, aby se mohlo pokračovat ve zkoušení, což považuji za zbytečné zdržení. Tento problém se nevyskytuje u aplikace Angličtina V, kde je zkoušení plynulé, bez zbytečných mezikroků. Bohužel zde ale chybí možnost zkoušet slovní zásobu ve směru z angličtiny do češtiny.

Dále se mi zejména u aplikace A-Test 4.0 příliš nelíbilo navržené ovládání. Nastává zde problém především při přepnutí zobrazení, kdy je nepochopitelně nutné pokaždé znovu otvírat slovník, byť ten samý. Dále u této aplikace nejsem spokojen s rozdělením příkazu do skupin v hlavní nabídce, které mně přišlo nelogické. U aplikace Interlex 2.5.0.6 bylo rozdělení příkazů dobré, ale výhrady jsem měl pro změnu k přepínání způsobu zkoušení, které je poněkud zmatečné. Aplikaci A-Test 4.0 bych dále vytkl panel nástrojů, jehož ikonky jsou sice kvalitně graficky navrženy, ale jejich význam je na první pohled málo zřejmý. Ovládání programu Angličtina V je bezproblémové, nicméně je to vykoupeno malým počtem nabízených funkcí.

Následuje shrnutí kladů a záporů jednotlivých aplikací.

2.1.1 A-Test 4.0



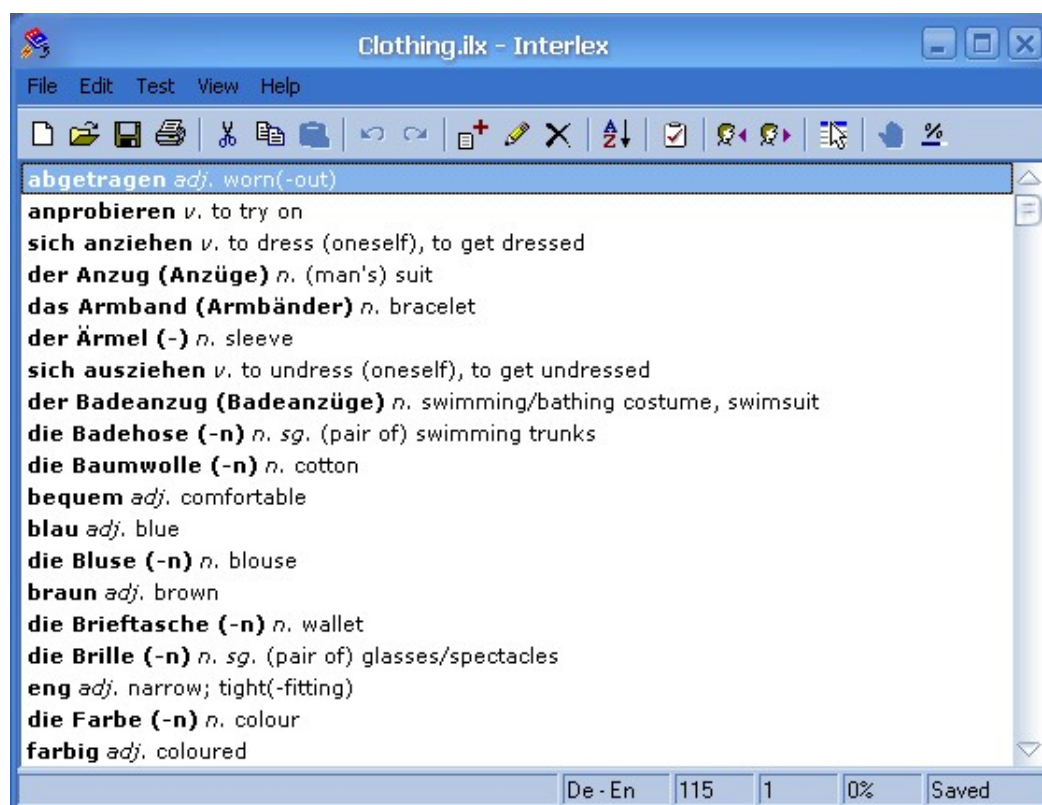
Obrázek 2.1: Aplikace A-Test 4.0

výhody/nevýhody:

- + databáze
- + vzhled ikonek
- málo intuitivní význam ikonek
- chybí výslovnost
- pouze 2 metody zkoušení
- nepohodlné ovládání
- slabé možnosti nastavení
- shareware

stránky: <http://www.ph-software.com>

2.1.2 Interlex 2.5.0.6



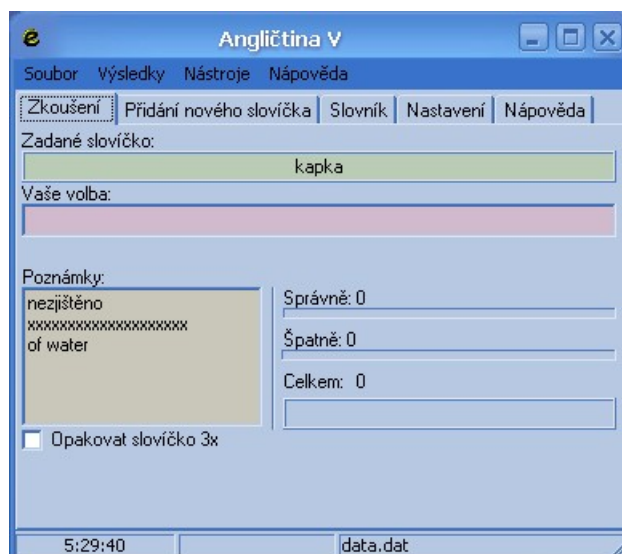
Obrázek 2.2: Aplikace Interlex 2.5.0.6

výhody/nevýhody:

- + nápověda
- + podpora
- + bohaté možnosti nastavení
- + rozlišování slov ve slovníku oddělených čárkou nebo středníkem
- chybí výslovnost
- pouze 2 metody zkoušení
- méně přehledný slovník

stránky: <http://www.vocab.co.uk>

2.1.3 Angličtina V



Obrázek 2.3: Aplikace Angličtina V

výhody/nevýhody:

- + jednoduché ovládání
- málo funkcí
- chybí výslovnost
- pouze 1 metoda zkoušení
- nelze otočit slovník při zkoušení
- slabé možnosti nastavení

stránky: <http://www.slunecnice.cz/sw/anglictina>

2.2 Výběr způsobu implementace

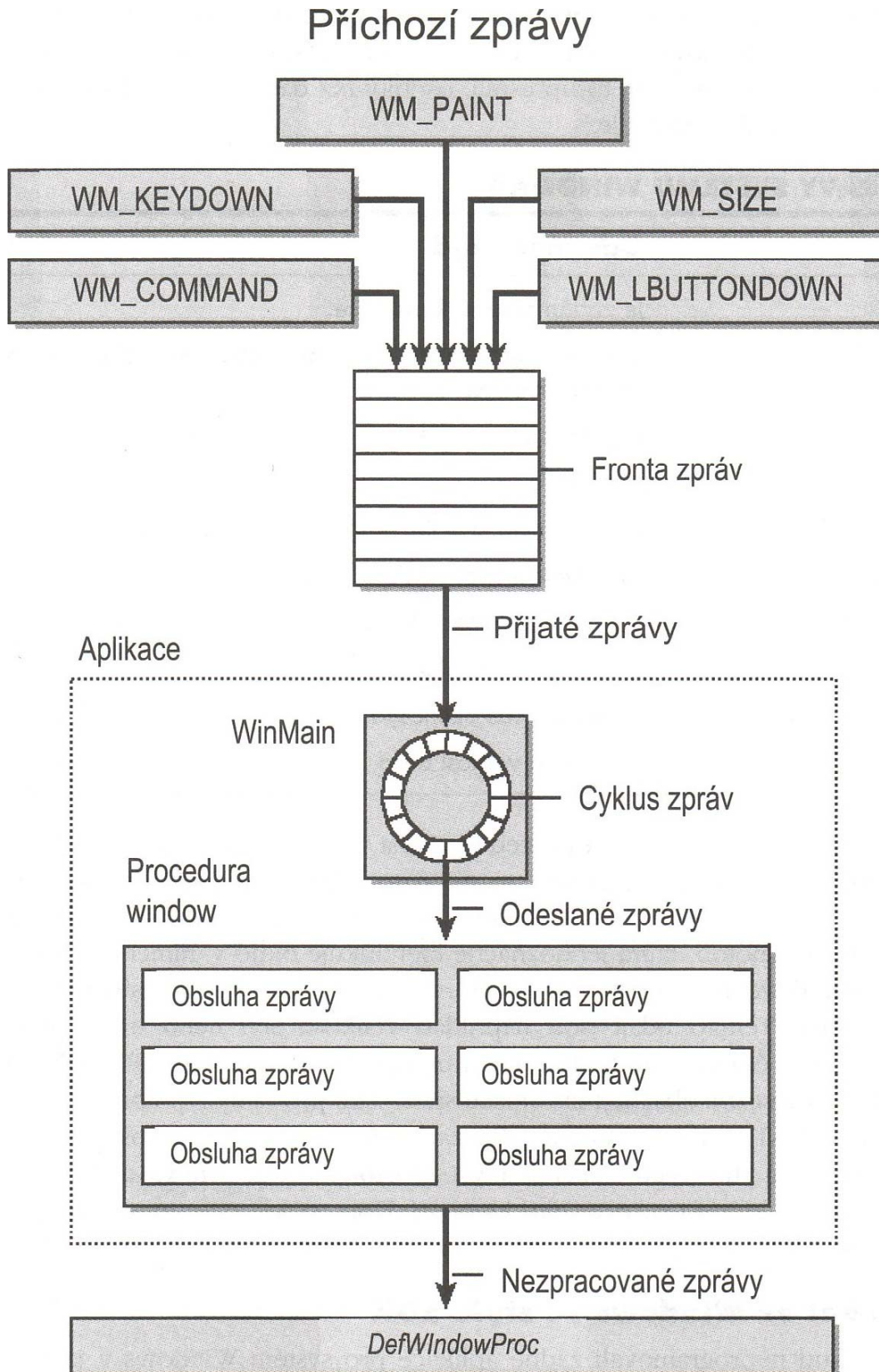
Výsledná aplikace má být dle zadání napsána ve vývojovém prostředí Microsoft Visual Studio. Jako programovací jazyk jsem si zvolil jazyk C++. Hlavním důvodem je především rychlost výsledné aplikace, která by za použití jazyků jako je C# nebo Java, nemusela být optimální. K jazyku C++ jsem se rozhodl využít knihovnu tříd MFC (Microsoft Foundation Class), jelikož je k ní nejvíce dostupných materiálů a považoval jsem ji pro řešení daného problému za nejvhodnější.

3 Programovací model Windows

Programy v systému Windows využívají událostmi řízený programovací model viz. Obrázek 3.1. Aplikace zde reagují na určité události zpracováním zpráv zasílaných operačním systémem. Událostí může být například stisknutí klávesy na klávesnici, kliknutí myši nebo příkaz k překreslení obsahu okna. Vstupním bodem každého programu je funkce nazvaná WinMain, ale většina akcí probíhá ve funkci známé jako procedura okna. Procedura okna zpracovává zprávy zasílané příslušnému oknu. WinMain vytvoří okno a následně vstoupí do cyklu zpráv nebo vyhledá zprávy a odešle je do procedury window. Zprávy čekají na zpracování ve frontě zpráv. Typické aplikace v prostředí Windows zpracovávají větší část přijatých zpráv a mezi tímto zpracováním nedělají téměř nic jiného, než čekají na následující zprávu.

Cyklus zpráv bude ukončen po zpracování zprávy WM_QUIT, která se k aplikaci dostane z fronty zpráv. Tato zpráva signalizuje, že nastal čas ukončit běh aplikace. Zpráva se ve frontě objeví obvykle po volbě příkazu Konec z nabídky Soubor nebo po kliknutí na systémové tlačítko Zavřít, případně po zadání příkazu Zavřít ze systémové nabídky každého okna. Po ukončení cyklu zpráv se řízení vrátí funkci WinMain a aplikace se ukončí.

Procedura okna zpravidla volá všechny ostatní funkce, které využívá k zpracování přijímaných zpráv. Může volat lokální funkce příslušné aplikace nebo funkce rozhraní API, jež poskytuje systém Windows. Funkce API jsou uloženy ve speciálním modulu zvaném dynamicky připojované knihovny neboli knihovny DLL. Rozhraní pro aplikační programy Win32 API zahrnuje stovky funkcí, které mohou být volány příslušnými aplikacemi a slouží ke zpracování různorodých úloh, jako je například vytváření oken, vykreslování čar nebo práce se soubory. V jazyce C je procedura okna zpravidla implementována jako monolitická funkce, zahrnující v sobě rozsáhlý strukturovaný příkaz switch s popisy jednotlivých zpráv. Kódu, jenž je určen k zpracování příslušné zprávy, se říká rutina pro obsluhu zprávy. Zprávy, které aplikace při svém běhu nezpracuje, jsou předávány funkci API nazvané DefWindowProc. Tato funkce poskytuje standardní odezvu na všechny nezpracované zprávy.



Obrázek 3.1: Událostmi řízený programovací model systému Windows

3.1 Zprávy systému Windows

System Windows definuje stovky různých typů zpráv. Název většiny zpráv začíná písmeny „WM“, za nimiž následuje podtržítko. Tabulka 3.1 obsahuje 10 nejobecnějších zpráv.

| Zpráva | Odeslána když |
|----------------|--|
| WM_CHAR | Je zadán znak z klávesnice. |
| WM_COMMAND | Uživatel vybere položku nabídky nebo ovládací prvek odešle hlášení nadřazenému kontejneru. |
| WM_CREATE | Je vytvořeno nové okno. |
| WM_DESTROY | Je zrušeno nové okno. |
| WM_LBUTTONDOWN | Bylo stisknuto levé tlačítko myši. |
| WM_MOUSEMOVE | Byl přesunut ukazatel myši. |
| WM_PAINT | Okno potřebuje překreslit svůj obsah. |
| WM_QUIT | Aplikace bude zrušena. |
| WM_SIZE | Je měněna velikost okna. |

Tabulka 3.1: Obecné zprávy systému Windows

Zpráva se projevuje v podobě volání procedury okna příslušného objektu okna. K tomuto volání jsou připojeny čtyři vstupní argumenty: manipulátor okna, jemuž je zpráva směřována, identifikátor zprávy a dva 32bitové argumenty známé pod označením *wParam* a *lParam*. Manipulátor okna je 32bitovou hodnotou, která jednoznačně identifikuje okno v rámci celého systému. Vnitřně se tato hodnota odvolává na datovou strukturu, v níž systém Windows uchovává údaje týkající se daného okna. Identifikátor zprávy je číselnou hodnotou, která určuje typ zprávy. Argumenty *wParam* a *lParam* obsahují všechny informace nezbytné ke zpracování dané zprávy.

4 Knihovna MFC

MFC (Microsoft Foundation Class) je knihovnou tříd jazyka C++, poskytovanou společností Microsoft k vytvoření objektově orientovaného obalu okolo rozhraní Windows API. Verze 8, uvolněná společně s vývojovým prostředím Microsoft Visual Studio 2005, obsahuje již několik set tříd. V programu typu MFC se zřídka volá přímo funkce rozhraní API. Místo toho se na základě tříd MFC vytváří objekty a volají se funkce, jež jsou členy těchto objektů. Mnoho set členských funkcí definovaných v knihovně tříd je tenkým obalem nad rozhraním Windows API a častokrát má stejné názvy jako odpovídající funkce rozhraní API.

4.1 Linkování knihovny MFC

Existují dva způsoby využití knihovny MFC. Program může být linkován buďto tak, že bude za běhu využívat příslušnou dynamickou knihovnu DLL knihovny MFC (tzv. dynamické linkování knihovny MFC) nebo knihovna MFC bude staticky přilinkována do souboru exe programu (tzv. statické linkování knihovny MFC).

Výhodou dynamického linkování je menší velikost souboru exe a zpravidla vede i k efektivnějšímu využívání systémových prostředků. Výhodou statického linkování je zase nezávislost programu na dalším externím souboru, což je ale na úkor o něco větší velikosti vlastního souboru exe.

4.2 Mapa zpráv

Mapa zpráv je tabulka, která slaďuje zprávy a členské funkce. Ve chvíli, kdy hlavní okno aplikace obdrží zprávu (např. stisknutí klávesy na klávesnici), MFC projde mapu zpráv, zjistí která obslužná rutina je pro zpracování dané zprávy určena a tu pak volá.

Postup pro přidávání map do MFC aplikací:

1. Uvedením příkazu `DECLARE_MESSAGE_MAP` v deklaraci třídy definujeme mapu zpráv.
2. Mapu zpráv implementujeme přidáním maker rozpoznávajících zprávy, které třída bude ošetřovat mezi jednotlivými voláními do kódu mezi příkazy `BEGIN_MESSAGE_MAP` a `END_MESSAGE_MAP`.
3. Přidáme členské funkce k ošetření zpráv.

4.3 Myš a klávesnice

4.3.1 Zachycení vstupu z myši

Systém Windows používá řadu různých zpráv, které podávají hlášení o událostech vyvolaných myši. Tyto zprávy se rozdělují na zprávy vyvolané v klientské oblasti okna a zprávy vyvolané mimo ni. Události odehrávající se mimo klientskou oblast programátor zpravidla ignoruje a jejich obsluhu ponechává na operačním systému. V následující tabulce jsou uvedeny zprávy, prostřednictvím kterých systém Windows podává hlášení o událostech vyvolaných myši v klientské oblasti okna.

| Zpráva | Odeslaná |
|---------------------|--|
| WM_LBUTTONDOWN | Po stisknutí levého tlačítka myši. |
| WM_LBUTTONUP | Po uvolnění levého tlačítka myši. |
| WM_LBUTTONDOWNBLCLK | Po poklepání na levé tlačítko myši. |
| WM_MBUTTONDOWN | Po stisknutí prostředního tlačítka myši. |
| WM_MBUTTONUP | Po uvolnění prostředního tlačítka myši. |
| WM_MBUTTONDOWNBLCLK | Po poklepání na prostřední tlačítko myši. |
| WM_RBUTTONDOWN | Po stisknutí pravého tlačítka myši. |
| WM_RBUTTONUP | Po uvolnění pravého tlačítka myši. |
| WM_RBUTTONDOWNBLCLK | Po poklepání na pravé tlačítko myši. |
| WM_MOUSEMOVE | Při přesouvání ukazatele myši nad klientskou oblastí okna. |

Tabulka 4.1: Zprávy z klientské oblasti okna

4.3.2 Zachycení vstupu z klávesnice

Aplikace systému Windows se o událostech klávesnice dozvídá stejným způsobem, jako se dozvídá o událostech myši, tedy prostřednictvím zpráv. Program přijme zprávu po každém stisknutí klávesy nebo jejím uvolnění. Chceme-li vědět, jakou klávesou uživatel stiskne, aby naše aplikace mohla adekvátně zareagovat, zpracujeme zprávy WM_KEYDOWN a ověříme virtuální kód stisknuté klávesy. Pro uvolnění klávesy zpracujeme zase zprávy WM_KEYUP.

| Virtuální kódy kláves | Příslušné klávesy |
|-----------------------|-------------------|
| VK_RETURN | Enter |
| VK_CONTROL | Ctrl |
| VK_INSERT | Insert nebo Ins |
| VK_DELETE | Delete nebo Del |
| 65 | A |
| 66 | B |
| 67 | C |

Tabulka 4.2: Virtuální kódy některých kláves

5 Návrh řešení

Po prostudování existujících řešení, které mě nikterak nenadchly, jsem při návrhu svého řešení kladl důraz především na jednoduché, intuitivní ovládání a propracovanost zkoušení.

Ovládání aplikace jsem se snažil navrhnout takové, které by žádnému uživateli nedělalo závažnější problémy. Všechny příkazy, které aplikace nabízí, jsem se snažil v hlavní nabídce logicky rozdělit do skupin. Příkazy související s prací se soubory jsem umístil do skupiny příkazů Soubor, příkazy související se zobrazením/skrytím některých částí aplikace do skupiny příkazů Zobrazit, příkazy související se zkoušením do skupiny příkazů Zkoušení a konečně příkazy související s prací se slovníkem do skupiny příkazů Nástroje. Poslední skupinou příkazů je Nápověda, která bývá standardně v každé aplikaci určené pro operační systém Windows.

Dále jsem se při návrhu ovládání snažil dodržovat zavedené klávesové zkratky, jako je například Ctrl+N pro vytvoření nového souboru, Ctrl+O pro otevření souboru, Ctrl+S pro uložení souboru nebo Ctrl+F pro hledání. Pro přidávání položek do slovníku jsem vyhradil klávesu Insert, pro editování klávesu Enter a pro mazání klávesu Delete, což považuji za praktické.

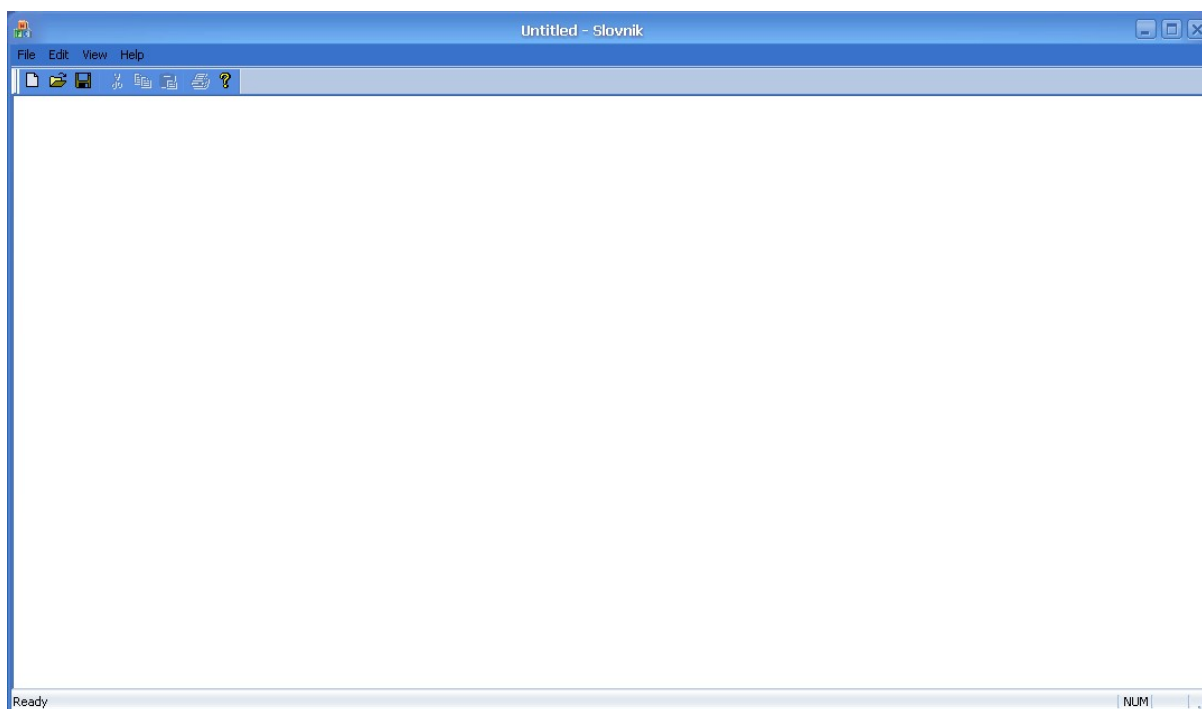
Za důležitou součást aplikací považuji dobře navržený panel nástrojů. Provedení ikoněk by mělo mít pokud možno intuitivní význam a panel nástrojů by neměl zbytečně obsahovat všechny příkazy z hlavní nabídky, ale pouze ty, u kterých se předpokládá vysoká četnost používání. Nezanedbatelné je také kvalitní grafické provedení ikoněk, které celé aplikaci přidá na atraktivitě a zpříjemní ovládání.

Zkoušení jsem při návrhu vyčlenil do samostatných dialogových oken, což má podle mého názoru oproti řešením, které zkoušení přepínají přímo v pohledu hlavního okna, hned několik výhod. V mém řešení se například po spuštění zkoušení zobrazí příslušné dialogové okno a hlavní okno se skryje, což má tu výhodu, že uživatel není zbytečně opticky rozptylován hlavní nabídkou a panelem nástrojů, které ho v to dobu nezajímají. Další výhodou je logičtější ukončení kliknutím na Konec, místo pouhého přepnutí pohledu v hlavním okně nebo ještě hůře kombinací předchozích dvou.

Dále jsem při návrhu zkoušení kladl důraz, aby zkoušení probíhalo plynule a zbytečně neunavovalo. Pro usnadnění ovládání jsem zde vyhradil řadu funkčních kláves. U zkoušení, kde je úkolem vybrat správnou odpověď z několika možností, jsou například vyhrazeny klávesy A, B, C, ... Podrobněji s návrhem řešení seznamuje příloha A Uživatelská příručka.

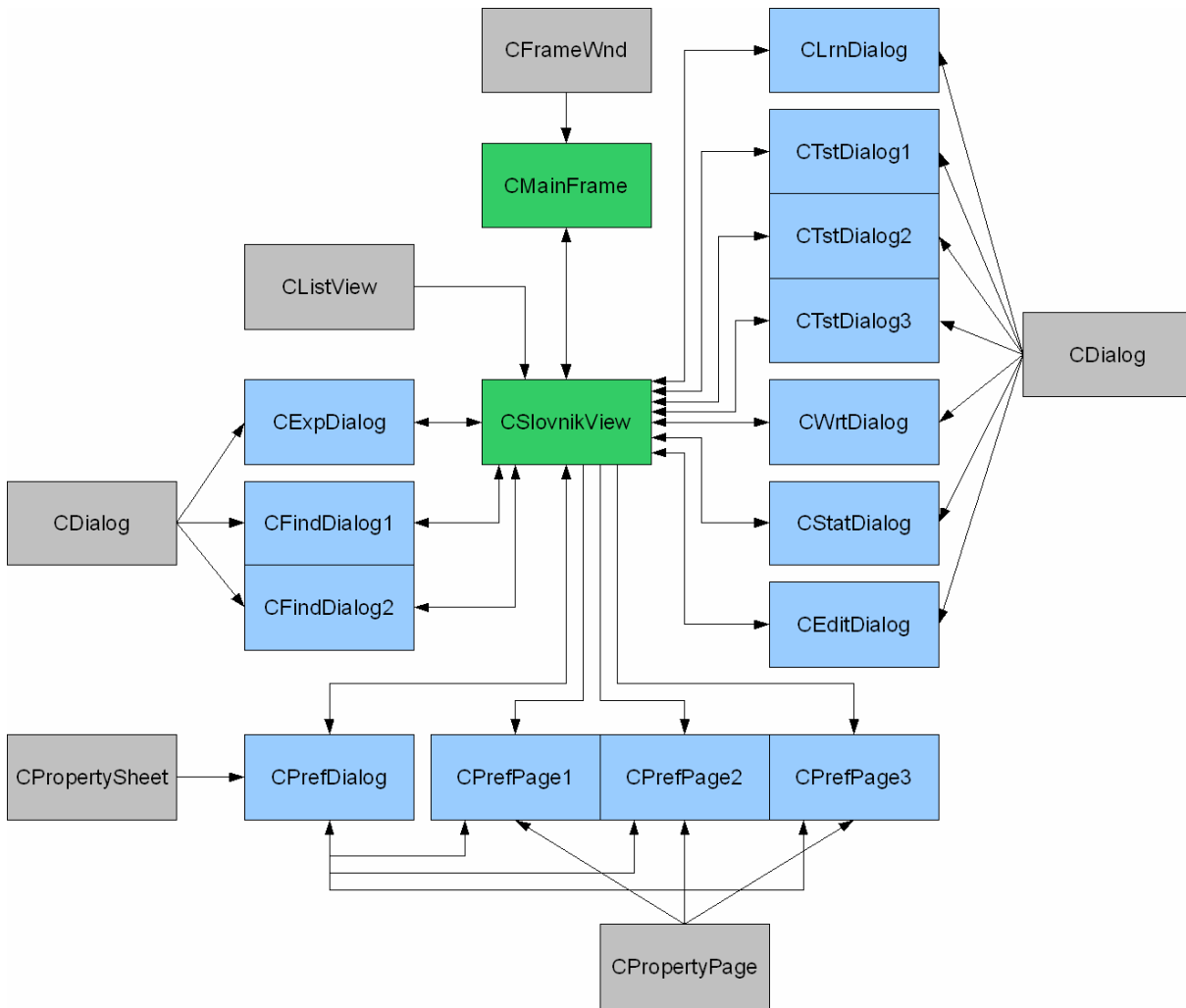
6 Implementace

Vytvoříme-li v Microsoft Visual Studiu 2005 pomocí průvodce nový projekt založený na knihovně MFC, získáme přeložitelný projekt, který po spuštění může vypadat tak, jak je uvedeno na obrázku 6.1. Automaticky jsou vygenerovány potřebné třídy a soubory, které podle nastavení při vytváření projektu zajistí například zobrazení hlavního okna, hlavní nabídky, panelu nástrojů a stavového řádku. Pro další vývoj aplikace je nutné tyto třídy a soubory dle potřeb upravit a samozřejmě doplnit o další třídy.



Obrázek 6.1: Nový projekt

Obrázek 6.2 naznačuje provázání nejdůležitějších tříd použitých v tomto projektu. Zeleně jsou označeny třídy, které vznikly při zakládání projektu, a které byly dále modifikovány. Modře jsou označeny třídy, které byly do projektu přidány ručně. Šedivě jsou označeny rodiče jednotlivých tříd.



Obrázek 6.2: Provázání tříd

6.1 Popis tříd

6.1.1 CMainFrame

CMainFrame je třída, kterou vygeneruje Visual Studio při zakládání nového projektu. Je odvozena od třídy *CFrameWnd* a jejím hlavním úkolem je za pomoci dalších tříd jako je *CToolBar*, *CStatusBar* zobrazit hlavní okno, panel nástrojů a stavový řádek.

V této třídě bylo potřeba upravit:

- počáteční rozměry hlavního okna,
- název aplikace v záhlaví hlavního okna,
- pole indikátorů stavového řádku,
- chování virtuální členské funkce *WindowProc*.

V této třídě bylo potřeba doplnit:

- kód pro zapamatování změn rozměrů a polohy hlavního okna [4],
- kód pro zapamatování změn zobrazení/skrytí panelů nástrojů a stavového řádku,

- metody a obslužné rutiny pro práci s tray ikonkou: *AddTrayIcon*, *RemoveTrayIcon*, *AnimateTrayIcon*, *StopAnimateTrayIcon*, *OnStartAnimateTrayIcon*, *OnStopAnimateTrayIcon*, *OnRunAnimateTrayIcon*, *OnPauseAnimateTrayIcon*, *OnShowInfoTrayIcon*, *OnShowAlertTrayIcon*,
- odeslání uživatelské zprávy, která iniciuje uložení všech potřebných nastavení před ukončením aplikace.

6.1.2 CSlovníkView

CSlovníkView je třída, která vznikne také již při zakládání nového projektu. Její jméno je odvozeno od názvu projektu, ve výchozím stavu je potomkem třídy *CView* a zjednodušeně řečeno, zobrazí v hlavním okně bílé plátno, které je vidět na obrázku 6.1.

Třída *CSlovníkView* je v tomto projektu nejzásadnější a také nejrozsáhlejší třída. Jsou v ní implementovány a řízeny téměř veškeré funkce aplikace.

V této třídě bylo potřeba upravit:

- dědění z *CView* na *CListView*, aby se mohlo v hlavním okně pracovat se seznamem textových řetězců rozděleným do sloupců viz. Obrázek A.19.

V této třídě bylo potřeba doplnit:

- styly ve virtuální členské funkci *PreCreateWindow*, neboli deklarovat jak se má seznam textových řetězců chovat a jaké má mít vlastnosti,
- členskou funkci *OnInitialUpdate*, ve které jsou nadefinovány sloupce spolu s fonty a rozšířenými styly seznamu textových řetězců,
- proměnné, do kterých se ukládají stavy aplikace
- proměnné, do kterých se ukládají volby nastavení,
- metodu *GetTime*, která pro časovač převádí čas z minut na milisekundy,
- obslužné rutiny příkazů hlavní nabídky: *OnFileNew*, *OnFileOpen*, *OnFileSave*, *OnFileSaveAs*, *OnFileImport*, *OnFileExport*, *OnViewPronun*, *OnViewNotice*, *OnViewStatus*, *OnTestingLearning*, *OnTestingTest*, *OnTestingWriting*, *OnTestingTurn*, *OnTestingAuto*, *OnTestingStatistic*, *OnTestingReset*, *OnToolsFind*, *OnToolsNewEntry*, *OnToolsEditEntry*, *OnToolsDelEntry*, *OnToolsDelAll*, *OnToolsSelAll*, *OnToolsEnableAll*, *OnToolsDelDup*, *OnToolsPreferences*,
- obslužné rutiny aktivující/deaktivující příkazy v hlavní nabídce: *OnUpdateFileSave*, *OnUpdateFileExport*, *OnUpdateViewPronun*, *OnUpdateViewNotice*, *OnUpdateViewStatus*, *OnUpdateTestingLearning*, *OnUpdateTestingTest*, *OnUpdateTestingWriting*, *OnUpdateTestingTurn*, *OnUpdateTestingAuto*, *OnUpdateTestingStatistic*, *OnUpdateTestingReset*, *OnUpdateToolsFind*, *OnUpdateToolsEditEntry*, *OnUpdateToolsDelEntry*, *OnUpdateToolsDelAll*, *OnUpdateToolsSelAll*, *OnUpdateToolsEnableAll*, *OnUpdateToolsDelDup*,
- metody realizující funkce aplikace: *LoadFile*, *SaveFile*, *ImportFileTxt*, *ImportFileExd*, *ExportFile*, *RunLearning*, *RunTest*, *RunWriting*, *RunAutoTesting*, *StopAutoTesting*, *ContinueAutoTesting*, *PauseAutoTesting*, *AlertAutoTesting*, *InfoAutoTesting*, *ResetStatistic*, *NewEntry*, *EditEntry*, *DelEntry*, *DelAll*, *SelAll*, *EnableAll*, *DelDup*,
- metody *SavePref* a *LoadPref* pro ukládání a načítání voleb nastavení aplikace,

- metodu *InitWords*, která před každým spuštěním zkoušení zjišťuje, zda-li jsou k dispozici nějaké nenaučené položky,
- metodu *CompileTestingReturn*, která zpracovává návratové hodnoty dialogových oken se zkoušením,
- obslužné rutiny aktualizující údaje stavového řádku: *OnUpdateIndicatorGood*, *OnUpdateIndicatorBad*, *OnUpdateIndicatorDis*, *OnUpdateIndicatorCount*,
- obslužnou rutinu *OnClick* reagující na kliknutí levým tlačítkem myši v seznamu textových řetězců,
- obslužnou rutinu *OnDbClick* reagující na poklepání levým tlačítkem myši v seznamu textových řetězců,
- obslužnou rutinu *OnSize*, která reaguje na změny rozměrů seznamu textových řetězců a přenastavuje šířky sloupců,
- obslužnou rutinu časovače *OnTimer*,
- metodu *PreTranslateMessage* reagující na stisknutí funkční klávesy Insert, Enter nebo Delete,
- obslužné rutiny uživatelských zpráv *OnRunAutoTesting*, *OnSavePref*.

6.1.3 CLrnDialog

Úkolem třídy *CLrnDialog* je na základě volání z třídy *CSlovníkView* zobrazit a ovládat dialogové okno s metodou zkoušení *Učení* viz. Obrázek A.3 a Obrázek A.4. Je odvozena od třídy *CDialog* a obsahuje:

- členskou funkci *OnInitDialog* volanou před prvním zobrazením dialogového okna, jejímž úkolem je zde nastavení fontů, nastavení barev textu, skrytí některých tlačítek, nastavení názvů některých tlačítek,
- metodu *GetItemIndex*, která vybírá ze seznamu textových řetězců vhodný index položky ke zkoušení,
- metodu *SetNewWords*, která na základě indexu, získaného pomocí metody *GetItemIndex*, nastaví v dialogovém okně položku ke zkoušení,
- obslužnou rutinu *OnNext*, která podle stavové proměnné reaguje buď na stisknutí tlačítka >> nebo Vím a aktualizuje statistické údaje,
- obslužnou rutinu *OnIDontKnow*, která reaguje na stisknutí tlačítka Nevím a aktualizuje statistické údaje,
- metodu *SetStatistic*, která nastavuje aktuální údaje v průběžné statistice,
- metodu *GetRan*, která třídě *CSlovníkView* sděluje, zda-li došlo ke zkoušení,
- metodu *PreTranslateMessage* reagující na stisknutí funkční klávesy B, V, N nebo K

6.1.4 CTstDialog1, CTstDialog2, CTstDialog3

Úkolem tříd *CTstDialog1*, *CTstDialog2* a *CTstDialog3* je na základě volání z třídy *CSlovníkView* zobrazit a ovládat dialogové okno s metodou zkoušení *Test* viz. Obrázek A.5. Jednotlivé třídy se liší pouze počtem možností správných odpovědí, se kterým daná třída počítá. Všechny třídy jsou odvozeny od třídy *CDialog* a obsahují:

- členskou funkci *OnInitDialog* volanou před prvním zobrazením dialogového okna, jejímž úkolem je zde nastavení fontů textu,
- metodu *GetItemIndex*, která vybírá ze seznamu textových řetězců vhodný index položky ke zkoušení,
- metodu *SetNewWords*, která na základě indexu, získaného pomocí metody *GetItemIndex*, nastaví v dialogovém okně položku ke zkoušení, dále náhodně vygeneruje a nastaví odpovědi, z nichž je právě jedna správná,
- obslužnou rutinu *OnNext*, která reaguje na stisknutí tlačítka Nevím a aktualizuje statistické údaje,
- obslužné rutiny *OnTest1*, *OnTest2*, *OnTest3* a podle počtu správných odpovědí, až *OnTest5*, které reagují na stisknutí tlačítek a, b, c a podle počtu správných odpovědí až e a vyhodnocují správnost odpovědí,
- metody *ClickIsGood* a *ClickIsBad*, které aktualizují statistické údaje,
- metodu *SetStatistic*, která nastavuje aktuální údaje v průběžné statistice,
- metodu *GetRan*, která třídě *CSlovníkView* sděluje, zda-li došlo ke zkoušení,
- metodu *PreTranslateMessage* reagující na stisknutí funkční klávesy N, K, A, B, C a podle počtu správných odpovědí, až D nebo E.

6.1.5 CWrtDialog

Úkolem třídy *CWrtDialog* je na základě volání z třídy *CSlovníkView* zobrazit a ovládat dialogové okno s metodou zkoušení *Psaní* viz. Obrázek A.8. Je odvozena od třídy *CDialog* a obsahuje:

- členskou funkci *OnInitDialog* volanou před prvním zobrazením dialogového okna, jejímž úkolem je zde nastavení fontů textu,
- metodu *GetItemIndex*, která vybírá ze seznamu textových řetězců vhodný index položky ke zkoušení,
- metodu *SetNewWords*, která na základě indexu, získaného pomocí metody *GetItemIndex*, nastaví v dialogovém okně položku ke zkoušení,
- obslužnou rutinu *OnNext*, která reaguje na stisknutí tlačítka Nevím a aktualizuje statistické údaje,
- obslužnou rutinu *OnOK*, která reaguje na stisknutí tlačítka OK, vyhodnocuje správnost odpovědí a aktualizuje statistické údaje,
- metodu *SetStatistic*, která nastavuje aktuální údaje v průběžné statistice,
- metodu *GetRan*, která třídě *CSlovníkView* sděluje, zda-li došlo ke zkoušení,
- metodu *CompareEntries*, která porovnává dva textové řetězce,
- metodu *RmvDcrtclMrks*, která z textového řetězce s diakritikou, vytvoří textový řetězec bez diakritiky,

6.1.6 CEditDialog

Úkolem třídy *CEditDialog* je na základě volání z třídy *CSlovníkView* zobrazit a ovládat dialogové okno pro přidávání nebo editaci položek viz. Obrázek A.13 a Obrázek A.14. Je odvozena od třídy *CDialog* a obsahuje:

- členskou funkci *OnInitDialog* volanou před prvním zobrazením dialogového okna, jejímž úkolem je zde nastavení fontů textu, nastavení titulku v záhlaví dialogového okna,
- obslužné rutiny *OnPron1* až *OnPron10* reagující na stisknutí tlačítka se znakem výslovnosti,
- obslužnou rutinu *OnCheck* reagující na změny stavu zaškrťovacího tlačítka,
- metodu *GetChecked*, která třídě *CSlovníkView* sděluje, zda-li je zaškrtnuté zaškrťovací tlačítko,

6.1.7 CFindDialog1

Úkolem třídy *CFindDialog1* je na základě volání z třídy *CSlovníkView* zobrazit a ovládat dialogové okno pro vyhledávání položek viz. Obrázek A.11. Je odvozena od třídy *CDialog* a obsahuje:

- obslužnou rutinu *OnOK* reagující na stisknutí tlačítka OK, která ukládá zadaný textový řetězec z editační řádky do proměnné typu *CString* a je-li neprázdný, zavře dialogové okno.

6.1.8 CFindDialog2

Úkolem třídy *CFindDialog2* je na základě volání z třídy *CSlovníkView* zobrazit a ovládat dialogové okno pro zobrazení výsledku vyhledávání viz. Obrázek A.12. Je odvozena od třídy *CDialog* a obsahuje:

- členskou funkci *OnInitDialog* volanou před prvním zobrazením dialogového okna, jejímž úkolem je zde nastavení fontů textu,
- obslužnou rutinu *OnNext* reagující na stisknutí tlačítka Další, jejímž úkolem je nalézt a zobrazit další položku,
- obslužné rutiny *OnNew*, *OnEdit*, *OnDelete* reagující na stisknutí tlačítek Nové, Editovat, Smazat, jejichž úkolem je pouze zavřít dialogové okno s různou návratovou hodnotou a třídě *CSlovníkView* tak sdělit, jak má dále pokračovat,
- metodu *PreTranslateMessage* reagující na stisknutí funkční klávesy D, N, E nebo S.

6.1.9 CStatDialog

Úkolem třídy *CStatDialog* je na základě volání z třídy *CSlovníkView* zobrazit dialogové okno se statistikou viz. Obrázek A.10. Je odvozena od třídy *CDialog* a obsahuje:

- metodu *SetData* volanou z konstruktoru třídy, která nastaví a vypočítá všechny údaje určené k zobrazení,
- členskou funkci *OnInitDialog* volanou před prvním zobrazením dialogového okna, jejímž úkolem je zde nastavení fontů textu, nastavení barev textu, nastavení indikátorů průběhu,
- metodu *SetProgressBars* volanou z členské funkce *OnInitDialog*, která nastaví pozice indikátorů průběhu.

6.1.10 Ostatní třídy

Následuje stručné vysvětlení významu funkčně méně zajímavých nebo převzatých tříd.

6.1.10.1 *CPrefDialog*, *CPrefPage1*, *CPrefPage2*, *CPrefPage3*

Úkolem třídy *CPrefDialog* je za pomoci tříd *CPrefPage1*, *CPrefPage2* a *CPrefPage3* zobrazit dialogové okno s kartami viz. Obrázek A.15 a aktualizovat proměnné obsahující volby nastavení.

6.1.10.2 *CExpDialog*

Úkolem třídy *CExpDialog* je zobrazit dialogové okno pro zvolení kritérií exportování slovní zásoby viz. Obrázek A.2.

6.1.10.3 *CColorStatic*

Třída, jež umožňuje používat v aplikaci barevný text [1].

6.1.10.4 *CEnBitmap*, *CColor*

Převzaté třídy, jež umožňují použít v panelů nástrojů ikonky o 24 bitové barevné hloubce [5].

6.2 Uživatelské zprávy

Uživatelské zprávy používané pro komunikaci mezi třídami *CMainFrame* a *CSlovníkView*:

| Zpráva | Iniciuje |
|--------------------------------|--|
| WM_USER_START_ANIMATE_TRAYICON | Spuštění animace tray ikonky. |
| WM_USER_STOP_ANIMATE_TRAYICON | Zastavení animace tray ikonky. |
| WM_USER_RUN_ANIMATE_TRAYICON | Znovuspuštění animace tray ikonky. |
| WM_USER_PAUSE_ANIMATE_TRAYICON | Pozastavení animace tray ikonky. |
| WM_USER_SHOW_INFO_TRAYICON | Zobrazení informace v podobě balónového tipu. |
| WM_USER_SHOW_ALERT_TRAYICON | Zobrazení upozornění v podobě balónového tipu. |
| WM_USER_RUNTESTING | Zobrazení dialogového okna se zkoušením. |
| WM_USER_SAVEPREF | Uložení voleb nastavení aplikace. |

Tabulka 5.1: Uživatelské zprávy

6.3 Ukládání

Slovníky jsou ukládány v binární podobě a obsahují kromě cizích významů, českých významů, výslovností a poznámek i data pro statistiku, která slouží k zobrazení statistiky i po ukončení a opětovném spuštění aplikace. Data pro statistiku se dají kdykoliv v aplikaci resetovat (vynulovat). Importování a exportování slovní zásoby probíhá v textové podobě.

Volby nastavení samotné aplikace se ukládají do binárního souboru Data.dat. Tento soubor se vytvoří automaticky po prvním ukončení aplikace a je do něj uloženo výchozí nastavení, které se později dále přepisuje.

7 Testování

Aplikace byla testována na počítačové sestavě:

- Operační systém Microsoft Windows XP Professional SP2
- Procesor Athlon XP 2800+
- Operační paměť DDR 896 MB (333 MHz)

Využití operační paměti po spuštění (bez načtení slovníku):

| | |
|----------------------|---------|
| Vocab Pro 1.0 | 4468 kB |
| A-Test 4.0 | 6808 kB |
| Interlex 2.5.0.6 | 5352 kB |
| Anglictina V | 5228 kB |

Tabulka 7.1: Využití operační paměti 1

Využití operační paměti při otevření různě velikých slovníků:

| | Vocab Pro 1.0 | A-Test 4.0 |
|-------------|----------------------|------------|
| 100 slov | 6492 kB | 11288 kB |
| 1000 slov | 6852 kB | 11436 kB |
| 10000 slov | 10640 kB | 12856 kB |
| 50000 slov | 27224 kB | 23156 kB |
| 100000 slov | 50424 kB | 42164 kB |

Tabulka 7.2: Využití operační paměti 2

Je patrné, že aplikace je vhodná zejména pro práci se slovníky do 10000 slov. Při práci se slovníky okolo 100000 již značně zatěžuje operační paměť. Přesto na výkonných moderních počítačích lze pracovat i s takto relativně rozsáhlými slovníky. Praktický význam však budou mít pravděpodobně nejvíce slovníky okolo 100 slov, které jsou opisem lekcí z učebnic.

8 Závěr

Přes počáteční problémy s osvojením práce s knihovnou MFC, která není zcela triviální, se podařilo navrhnout a realizovat aplikaci, která splňuje všechny body zadání. Tedy výsledná aplikace umožňuje přidávání anglických slovíček nebo frází, včetně jejich plné výslovnosti. Umožňuje průběžné zkoušení třemi různými metodami během běžné práce na počítači a to nikoliv náhodně, jak je uvedeno v zadání, ale za každý časový interval, který si každý uživatel sám navolí, což se během návrhu ukázalo jako rozumnější. A konečně umožňuje přehledné a podrobné zobrazování výsledků dosažených při zkoušení.

Aplikace vedle dalších samozřejmých funkcí jako je editování, mazání nebo vyhledávání, umožňuje také automatické ukládání veškerých změn před ukončením, odstraňování duplicitních položek ze slovníků nebo například exportování slovní zásoby podle různých kritérií.

Předkládaná aplikace by měla být již plně použitelná v praxi. Před veřejným šířením by však bylo potřeba ještě navrhnout vlastní systémovou ikonu, doplnit nápovědu a případně také instalátor. Dále by bylo potřeba dalších důkladných testů, jelikož se jistě nepodařilo odstranit všechny chyby, případně drobnější nelogičnosti.

Nejvýznamnější problém bych zatím viděl v prázdných řádcích, které se v některých případech objevují nad položkami v tabulce. Tento problém se, bohužel, dosud nepodařilo uspokojivě vyřešit. Další z věcí, která není zatím zcela dořešena, je nastavování šířek sloupců v tabulce.

Z časových důvodů se již nepodařilo doplnit u dialogových oken se zkoušením plovoucí okénka, která by položky, které se svojí délkou nevejdou do šířky okna, zobrazovala v plné délce. Dále se také nepodařilo nijak významně využít poznámek ve zkoušení.

Jako námět na vylepšení bych viděl doplnění aplikace o indikátory průběhu, které jsou vhodné zejména při časově náročných operacích, jako je načítání extrémně velkých souborů nebo odstraňování duplicitních položek ze slovníků. Další námět na vylepšení je doplnění plovoucí nápovědy u údajů stavového řádku, neboť jejich význam nemusí být na první pohled zřejmý.

9 Seznam literatury

- [1] Jeff Proisse, Programování ve Windows pomocí MFC, Computer Press, Praha 2002
- [2] Radek Chalupa, 1001 tipů a triků pro Visual C++, Computer Press, Brno 2003
- [3] Beck Zaratian, Visual C++ 6.0 Příručka programátora, Computer Press, Brno 2004
- [4] Michael Walz, Preserving window position in a MFC SDI application,
<http://www.codeproject.com/dialog/windowpres.asp?df=100&forumid=2610&exp=0>
- [5] Christian Rodemeyer, Daniel Godson, Generating inactive/disabled images for toolbar,
<http://www.codeproject.com/docking/tbimages.asp?df=100&forumid=15527&exp=0>
- [6] Jonah Bishop, Automatically Size CListView Columns,
<http://www.borngeek.com/code/size-listview-columns.html>
- [7] Nishant Sivakumar, Keyboard messages/accelerators handling in MFC dialog based applications,
<http://www.codeproject.com/dialog/pretransdialog01.asp?df=100&forumid=2969&exp=0&select=1614666>
- [8] Niek Albert, That's a drag, <http://www.codeproject.com/listctrl/Dragger.asp>
- [9] Chris Maunder, Progress Control with Text,
http://www.codeproject.com/miscctrl/text_progressctrl.asp
- [10] Georg Bedenk, How to send a user-defined message with SendMessage, PostMessage or PostThreadMessage, <http://www.codeproject.com/tips/gbTestSDI.asp>
- [11] Chalupa Radek, Jak na tray-ikonu s C++,
<http://www.zive.cz/default.aspx?textartimg=1&article=102795>
- [12] English Pronunciation, http://www.wordreference.com/es/phonetic_symbols.htm
- [13] Free Icons Pack 1.0,
http://www.freeloadscenter.com/Shell_and_Desktop/Computer_Icons/Free_Icons_Pack.html
- [14] XP Artistic Icons Collection 3.0, <http://www.sofation.com/free-xp-artistic-icons-collection-install.html>
- [15] Ludmila Kollmannová, Angličtina pro samouky, Leda, Praha 1997

A Uživatelská příručka

A.1 Funkce

- Tvorba slovníků, včetně anglické výslovnosti a poznámek
- Importování slovní zásoby
- Exportování slovní zásoby podle 3 kritérií
- 3 metody zkoušení
- Automatické zkoušení
- Zobrazení podrobné statistiky zkoušení
- Vyhledávání položek
- Přidávání položek
- Editování položek
- Kontrola duplicity při přidávání/editaci položek
- Mazání položek
- Odstraňování duplicity ze slovníků

A.2 Vymezení pojmů

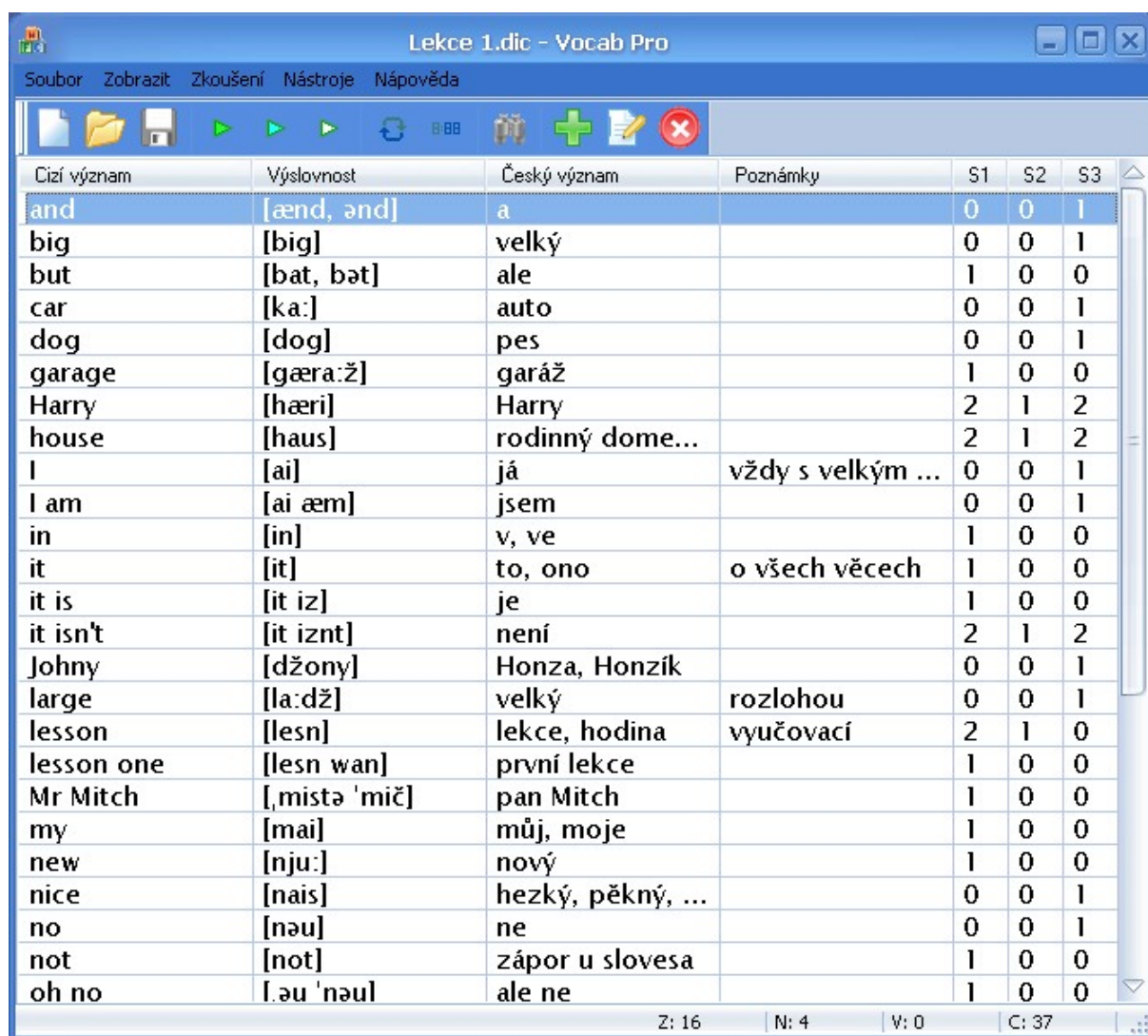
- Pro aplikaci byl zvolen název **Vocab Pro**.
- V aplikaci je použit termín „položka“ místo termínů jako je například „slovíčko“ nebo „slovo“, neboť lépe vystihuje fráze, jež aplikace rovněž podporuje.
- Položka je označena za **naučenou**, pokud na ni bylo odpovězeno správně v počtu (a to včetně případných penalizací za nesprávné odpovědi), který je dán nastavením. Položka pak již není vybírána ke zkoušení.
- Položka je označena za **nenaučenou**, pokud již byla zkoušena a nebylo na ni odpovězeno správně v počtu, který je nutný.
- Položka je označena za **zatím nezkoušenou**, pokud zatím ještě nebyla vybrána ke zkoušení.
- Položka je označena za **vyřazenou** na základě volby *Vyřadit ze zkoušení*.

A.3 Klávesové zkratky a funkční klávesy

| Klávesová zkratka / funkční klávesa | Funkce |
|-------------------------------------|---|
| Ctrl+N | Vytvoří nový slovník. |
| Ctrl+O | Otevře existující slovník. |
| Ctrl+S | Uloží aktivní slovník. |
| Ctrl+L | Spustí Učení. |
| Ctrl+T | Spustí Test. |
| Ctrl+W | Spustí Psaní. |
| Ctrl+Q | Otočí slovník při zkoušení. |
| Ctrl+F | Najde zadanou položku. |
| Ctrl+A | Vybere všechny položky. |
| Insert | Přidá položky. |
| Enter | Smaže vybrané položky. |
| Delete | Smaže označené položky. |
| A | U zkoušení Test vybere možnost a). |
| B | U zkoušení Test vybere možnost b). U zkoušení Učení zobrazí překlad. |
| C | U zkoušení Test vybere možnost c). |
| D | U zkoušení Test vybere možnost d). Nalezne další položku. |
| E | U zkoušení Test vybere možnost e). Edituje nalezenou položku. |
| K | Ukončí zkoušení. |
| N | Spustí nové hledání. |
| S | Smaže nalezenou položku. |

Tabulka A.1: Klávesové zkratky a funkční klávesy

A.4 Hlavní okno



| Cizí význam | Výslovnost | Český význam | Poznámky | S1 | S2 | S3 |
|-------------|---------------|-------------------|-------------------|----|----|----|
| and | [ænd, ənd] | a | | 0 | 0 | 1 |
| big | [big] | velký | | 0 | 0 | 1 |
| but | [bat, bət] | ale | | 1 | 0 | 0 |
| car | [ka:] | auto | | 0 | 0 | 1 |
| dog | [dog] | pes | | 0 | 0 | 1 |
| garage | [gæra:ž] | garáž | | 1 | 0 | 0 |
| Harry | [hæri] | Harry | | 2 | 1 | 2 |
| house | [haus] | rodinný dome... | | 2 | 1 | 2 |
| I | [ai] | já | vždy s velkým ... | 0 | 0 | 1 |
| I am | [ai æm] | jsem | | 0 | 0 | 1 |
| in | [in] | v, ve | | 1 | 0 | 0 |
| it | [it] | to, ono | o všech věcech | 1 | 0 | 0 |
| it is | [it iz] | je | | 1 | 0 | 0 |
| it isn't | [it iznt] | není | | 2 | 1 | 2 |
| Johnny | [džony] | Honza, Honzík | | 0 | 0 | 1 |
| large | [la:dž] | velký | rozlohou | 0 | 0 | 1 |
| lesson | [lesn] | lekce, hodina | vyučovací | 2 | 1 | 0 |
| lesson one | [lesn wan] | první lekce | | 1 | 0 | 0 |
| Mr Mitch | [,mɪstə 'mič] | pan Mitch | | 1 | 0 | 0 |
| my | [mai] | můj, moje | | 1 | 0 | 0 |
| new | [nju:] | nový | | 1 | 0 | 0 |
| nice | [nais] | hezký, pěkný, ... | | 0 | 0 | 1 |
| no | [nəu] | ne | | 0 | 0 | 1 |
| not | [not] | zápor u slovesa | | 1 | 0 | 0 |
| oh no | [.əu 'nəul] | ale ne | | 1 | 0 | 0 |

Obrázek A.1: Hlavní okno aplikace

Hlavní okno aplikace obsahuje hlavní nabídku, panel nástrojů, stavový řádek a tabulku, která je rozdělena na 7 sloupců.

A.5 Hlavní nabídka

Hlavní nabídka je rozdělena na tyto skupiny příkazů:

- Soubor
- Zobrazit
- Zkoušení
- Nástroje
- Nápověda

A.5.1 Soubor

Skupina příkazů Soubor obsahuje základní příkazy pro práci se soubory. Jsou to příkazy: *Nový*, *Otevřít*, *Uložit*, *Uložit jako*, *Importovat*, *Exportovat*, *Konec*.

Aplikace používá koncovku **dic** pro otevírání a ukládání slovníků a koncovku **exd** pro importování/exportování slovní zásoby s výslovností. Pro importování slovní zásoby bez výslovnosti je podporována koncovka **txt**.

Pro vytváření slovní zásoby bez výslovnosti v externím editoru (např. Poznámkový blok), kterou budeme chtít později importovat do nějakého slovníku, je podporován následující formát zadávání dat:

cizí význam 1=český význam 1

cizí význam 2=český význam 2

cizí význam 3=český význam 3

...

cizí význam n=český význam n

Položky lze exportovat podle třech kritérií:

- Pouze vybrané
- Všechny
- Pouze ty, které mají počet špatných reakcí větší nebo roven zadané hodnotě



Obrázek A.2: Exportování

A.5.2 Zobrazit

Skupina příkazů Zobrazit obsahuje příkazy, které slouží k zobrazení/skrytí částí aplikace. Jsou to příkazy: *Panel nástrojů*, *Stavový řádek*, *Výslovnost*, *Poznámky*, *Stavy*.

A.5.3 Zkoušení

Skupina příkazů Zkoušení obsahuje veškeré příkazy, které se vztahují ke zkoušení. Jsou to příkazy: *Učení*, *Test*, *Psaní*, *Otočit*, *Automatické zkoušení*, *Statistika*, *Resetovat*.

A.5.3.1 Metoda Učení

Slovní zásobu lze zkoušet třemi způsoby. První metoda je nazvána jako *Učení* a je vhodná zejména v případě, kdy danou slovní zásobu vůbec nebo velice špatně ovládáme. Metoda je simulací klasického učení, kdy si v knize zakryjeme jeden ze sloupců a postupně ho odkrýváme.



Obrázek A.3: Učení pohled 1

Spustíme-li metodu *Učení*, zobrazí se nám dialogové okno, které je na obrázku A.3. Tam máme uvedenu zkoušenou položku včetně výslovnosti (je-li ve slovníku zadána a je-li povolena v nastavení), u které si v prvním kroku nejprve rozmyslíme, jaký je překlad. V dalším kroku klikneme na tlačítko >> nebo stiskneme klávesu B a dialogové okno přejde do stavu, který je na obrázku A.4.



Obrázek A.4: Učení pohled 2

Tam je uvedena správná odpověď a na nás je, abychom potvrdili, zda-li jsem to věděli či nikoliv. Poté se celý proces opakuje s další položkou.

A.5.3.2 Metoda Test

Druhá metoda je nazvána jako *Test* a je vhodná zejména při lepší znalosti dané slovní zásoby. Spustíme-li metodu *Test*, zobrazí se nám dialogové okno, které je na obrázku A.5. Tam máme uvedenu zkoušenou položku a podle nastavení 3 až 5 možností správných překladů. Naším úkolem je vybrat právě jeden správný překlad. To můžeme provést buďto klasicky výběrem pomocí levého tlačítka myši nebo stisknutím odpovídající klávesy A, B, C, ...



Obrázek A.5: Test pohled 1

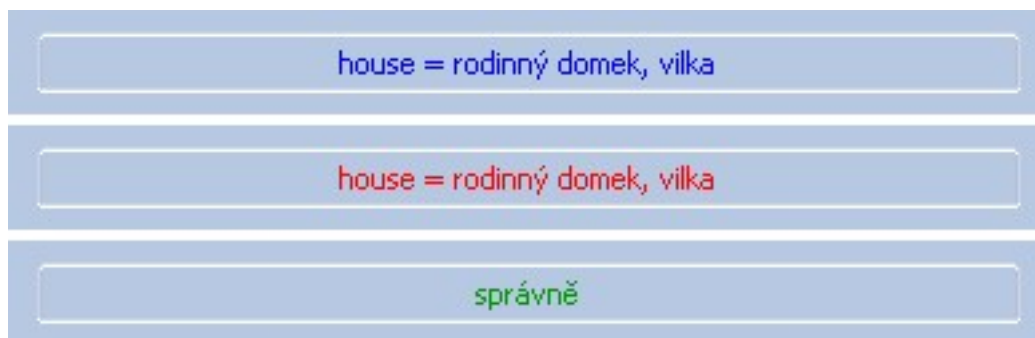
Nevíme-li správnou odpověď nebo nejsme-li si příliš jisti a nechceme zbytečně hádat (ani to není doporučeno), klikneme na tlačítko Nevím nebo stiskneme klávesu N a dialogové okno přejde do stavu, který je na obrázku A.6.



Obrázek A.6: Test pohled 2

Tam máme modře uvedenu správnou odpověď a nabídnutu další položku ke zkoušení. V případě zvolení špatné možnosti překladu se správná odpověď zobrazí nikoliv modře, ale

červeně. V případě zvolení správné možnosti překladu se zobrazí pouze vyrozumění „správně“ zeleně. Všechny možnosti indikace jsou zobrazeny na obrázku A.7.



Obrázek A.7: Test indikace

A.5.3.3 Metoda Psaní

Poslední, třetí metoda je nazvána jako *Psaní* a je vhodná zejména při dobré znalosti dané slovní zásoby. Spustíme-li metodu *Psaní*, zobrazí se nám dialogové okno, které je na obrázku A.8. Tam máme uvedenu zkoušenou položku a pod ní editační řádek pro psaní. Naším úkolem je napsat správný překlad – ten, který je uvedený ve slovníku. Čili pro zkoušenou položku „house“ je v našem případě správná odpověď „rodinný domek, vilka“. Aplikace toleruje i odpověď „rodinný domek“ nebo „vilka“, tedy dokáže ve slovníku rozlišit slova oddělená čárkou nebo středníkem. Lze také pro pohodlnější psaní vynechávat diakritiku. Příklady správných odpovědí, které jsou u této metody zkoušení tolerovány, jsou uvedeny v tabulce A.2.



Obrázek A.8: Psaní

Vyrozumění po správné/špatné odpovědi nebo po kliknutí na tlačítko Nevím je shodné jako u metody *Test*. Jediný rozdíl je, že v nastavení aplikace si můžeme navolit, zda-li chceme mít možnost ihned se opravit (např. z důvodu překlepu nebo napsání synonyma, které není zrovna uvedené ve slovníku). Tedy chceme-li po špatné odpovědi nechat ke zkoušení stávající položku nebo automaticky nabídnout další. Zvolíme-li si v nastavení aplikace, že chceme mít možnost opravy, pak se po špatné odpovědi nezobrazuje správná odpověď, ale pouze vyrozumění „špatně“ viz. Obrázek A.9. Na další položku se pak dostaneme buďto napsáním správné odpovědi nebo kliknutím na tlačítko Nevím.



Obrázek A.9: Psaní indikace

U této metody zkoušení není podporováno ovládání pomocí kláves, neboť ty jsou zde vyhrazeny pro psaní textu a různé kombinace například s klávesou Ctrl by neměly praktický význam.

| Ve slovníku je | Možná správná odpověď |
|----------------------------|--|
| učitel/ka | učitel/ka ucitel/ka učitel učitelka ucitel ucitelka |
| mýt (se), prát (prádlo) | mýt (se), prát (prádlo) myt (se), prat (prádlo) mýt (se) mýt se mýt prát (prádlo) prát prádlo prát myt (se) myt se myt prat (pradlo) prat pradlo prat |
| volný, svobodný; bezplatný | volný, svobodný; bezplatný volny, svobodny; bezplatny volný svobodný bezplatný volny svobodny bezplatny |

Tabulka A.2. Příklady správných odpovědí u Psaní

Pro všechny tři výše popsané metody zkoušení je společný průběžný údaj o stavu již naučených položek.

A.5.3.4 Otočení směru zkoušení

Aplikace umožňuje zkoušení jak ve směru z angličtiny do češtiny, tak i ve směru z češtiny do angličtiny. Slouží k tomu funkce *Otočit*, která podle toho, zda-li je navolena nebo ne, otáčí slovník při zkoušení. Výchozí směr je z angličtiny do češtiny.

A.5.3.5 Automatické zkoušení

Smyslem automatického zkoušení je uživatele průběžně zkoušet při běžné práci na počítači.

Přesněji funguje tak, že ho zapneme, program si zminimalizujeme do ikonky vedle hodin a za každý časový interval, který si nastavíme v nastavení aplikace, vyskočí dialogové okno se zvolenou metodou zkoušení, které nás jednorázově vyzkouší z jednoho slovíčka popřípadě fráze. Vhodné jsou zde zejména metody *Test* nebo *Psaní*. Na zkoušenou položku patřičně zareagujeme a dialogové okno se zkoušením se samo automaticky zavře. Poté se zobrazí malé informační okénko (ne u metody *Učení*), které nás informuje o správnosti naší reakce a případně sdělí správnou odpověď.

Můžeme si také nastavit, že nechceme, aby dialogové okno se zkoušením automaticky vyskakovalo, ale že chceme být pouze pomocí balónového tipu elegantně upozorněni, že je připraveno zkoušení a my si sami kliknutím na ikonku vedle hodin určíme, že chceme již reagovat.

A.5.3.6 Statistika

Statistika viz. Obrázek A.10 obsahuje všechny důležité údaje vztahující se k právě otevřenému slovníku. Lze ji spustit buď z hlavní nabídky, nebo je-li to povoleno v nastavení, spouští se automaticky po ukončení každém zkoušení.



Obrázek A.10: Statistika

A.5.3.7 Resetování

Je-li daný slovník (lekce) celý naučen, a chceme-li si ho zopakovat, funkce *Resetovat* resetuje statistiku a všechny stavy.

A.5.4 Nástroje

Skupina příkazů Nástroje obsahuje příkazy, které se vztahují ke slovníku nebo celé aplikaci. Jsou to příkazy: *Najít*, *Přidat*, *Editovat*, *Smazat*, *Smazat vše*, *Vybrat vše*, *Povolit vše*, *Odstranit duplicitu*, *Nastavení*.

A.5.4.1 Vyhledávání

Vyhledání položky ve slovníku bude pravděpodobně nejčastěji z důvodu, že ji budeme chtít buďto nějakým způsobem upravit nebo smazat. Podle toho byla také funkce *Najít* navržena. Hledaný název položky nejprve zadáme do kolonky v dialogovém okně, které je uvedeno na obrázku A.11. Poté klikneme na tlačítko *Najít* nebo stiskneme klávesu Enter.



Obrázek A.11: Hledání pohled 1

Jestliže jsme nezadali žádný text, neprovede se nic. V případě, že jsme nějaký text zadali a žádná položka nebyla nalezena, zobrazí se informační okno, které nás o tom informuje. V případě, že nějaká položka nalezena byla, zobrazí se dialogové okno viz. Obrázek A.12, kde je přehledně uvedena.



Obrázek A.12: Hledání pohled 2

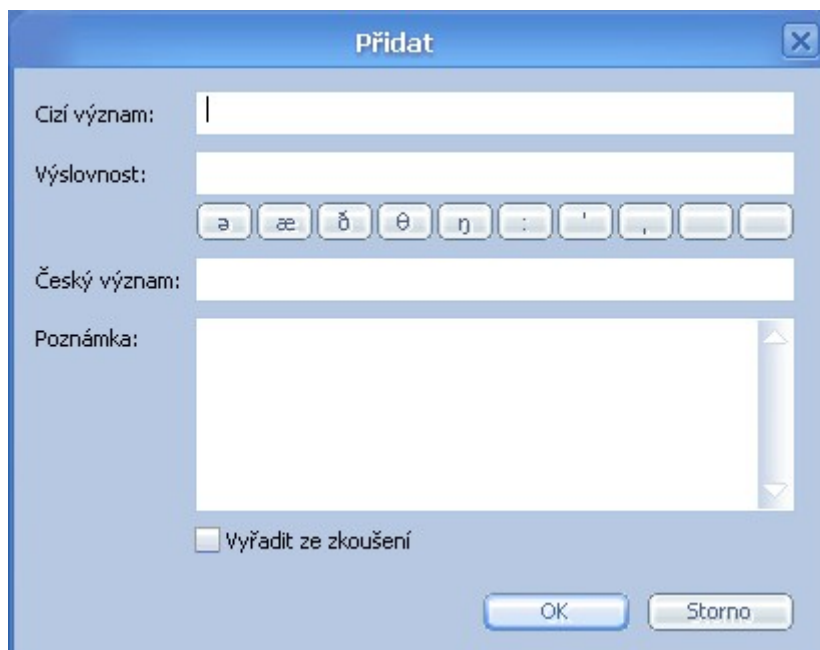
Nejsme-li s nalezenou položkou spokojeni, můžeme se pokusit najít další položku, která obsahuje hledaný název, zvolením volby *Další*. Chceme-li spustit nové hledání, zvolíme volbu *Nové*. Pro editování nalezené položky zvolíme volbu *Editovat* a pro smazání volbu *Smazat*. Nechceme-li provést nic, zvolíme *Storno*.

Funkce *Najít* hledá položky zároveň jak v cizím významu, tak v českém významu. Dále rozlišuje položky oddělené čárkou nebo středníkem.

A.5.4.2 Přidávání

Pro přidávání nových položek do slovníku slouží příkaz *Přidat*. Při přidávání je potřeba vyplnit jak kolonku *Cizí význam* tak kolonku *Česky význam*. Kolonky *Výslovnost* a *Poznámka* jsou nepovinné. Výslovnost se zadává pomocí tlačítek, které jsou umístěny pod kolonkou *Výslovnost* viz. Obrázek A.13. Dále při přidávání můžeme zatrhnout volbu *Vyřadit ze zkoušení*, čímž zajistíme, že daná položka nebude vybírána ke zkoušení.

Aplikace před vložením nové položky do slovníku umí automaticky kontrolovat, zda-li tam náhodou není již vložena položka se shodným cizím a českým významem a případně na to upozorní.



Obrázek A.13: Přidávání

A.5.4.3 Editování

Položky ve slovníku lze upravovat. Slouží k tomu příkaz *Editovat*, pro který platí shodná pravidla jako pro příkaz *Přidat*. Chceme-li nějakou položku editovat, musíme ji nejdříve vybrat levým tlačítkem myši. Dialogové okno pro editaci můžeme zobrazit přímo poklepáním myši na položku, kterou chceme editovat.



Obrázek A.14: Editování

A.5.4.4 Mazání

Pro mazání položek ve slovníku slouží příkaz *Smazat*. Mazat lze pouze položky vybrané levým tlačítkem myši. Levým tlačítkem myši můžeme vybrat buď jednu položku nebo více při stisknutí klávese Ctrl. Chceme-li smazat všechny položky, použijeme příkaz *Smazat vše* nebo kombinaci příkazů *Vybrat vše* + *Smazat*. Aplikace se před smazáním vždy zeptá, zda-li vybrané položky chceme opravdu smazat.

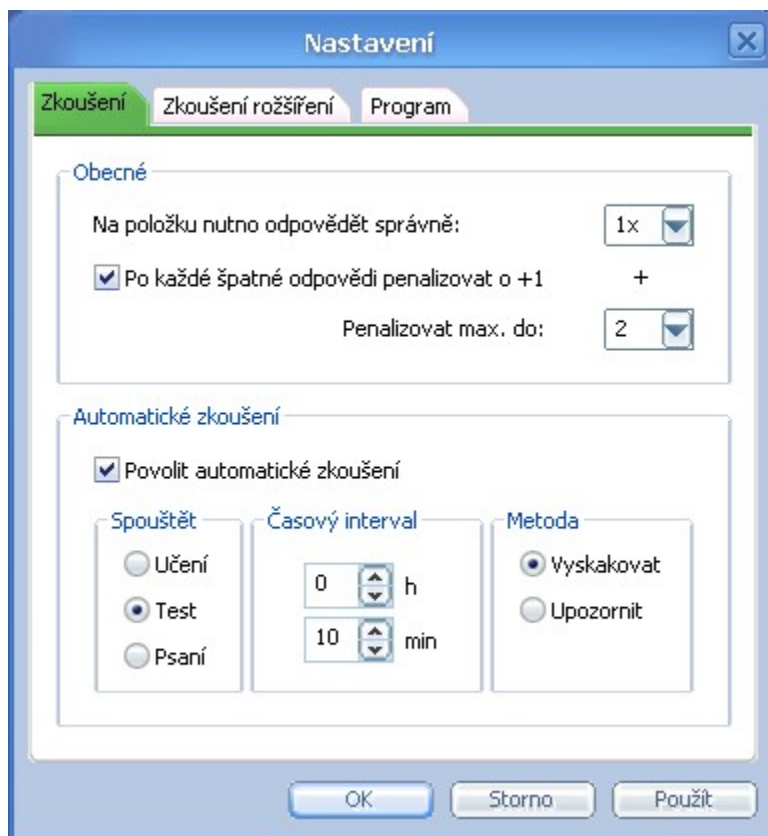
A.5.4.5 Další funkce

Další funkce, které aplikace nabízí, jsou *Povolit vše* a *Odstranit duplicitu*. Funkce *Povolit vše* povolí všechny položky vyřazené ze zkoušení. Funkce *Odstranit duplicitu* projde celý slovník a nalezne všechny položky, které se tam vyskytují dvakrát a vícekrát. Z nich nechá vždy pouze jednu (položky s výslovností mají přednost) a ostatní smaže.

A.5.4.6 Nastavení

Nastavení aplikace je rozděleno na karty *Zkoušení*, *Zkoušení rozšíření* a *Program*.

Na kartě *Zkoušení* ve skupině *Obecné viz. Obrázek A.15*, lze nastavit kolikrát, je na danou položku nutno odpovědět správně (ne nutně za sebou v případě vypnuté penalizace), aby byla uznána za naučenou. Je-li položka uznána za naučenou, není již dále vybírána ke zkoušení.



Obrázek A.15: Nastavení karta Zkoušení

Dále zde můžeme zapnout penalizaci a nastavit její hranici. Smyslem penalizace je u položek, na které špatně odpovíme, navýšit jejich počet správných odpovědí, který je nutný k jejich uznání za naučené. Navyšuje se vždy o +1 za každou špatnou odpověď až do zadané hranice.

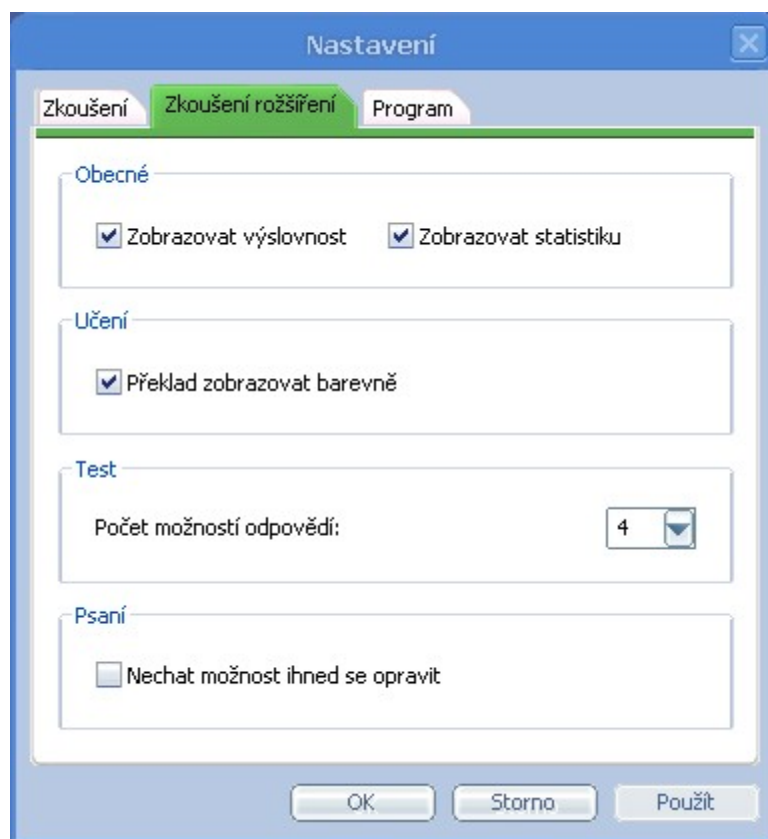
Máme-li například nastavení shodné s obrázkem A.15, mohou nastat tyto scénáře:

- Na zkoušenou položku odpovíme hned napoprvé správně a je uznána za naučenou.
- Na zkoušenou položku odpovíme hned napoprvé špatně a díky uplatnění penalizace za špatnou odpověď musíme na ni odteď odpovědět správně již nikoliv 1x, ale 2x. Poté na tuto položku odpovíme již 2x po sobě správně a je uznána za naučenou.
- Na zkoušenou položku odpovíme hned napoprvé špatně a díky uplatnění penalizace za špatnou odpověď musíme na ni odteď odpovědět správně již nikoliv 1x, ale 2x. Poté na tuto položku odpovíme opět špatně a díky uplatnění penalizace za špatnou odpověď musíme na ni odteď odpovědět správně již nikoliv 2x, ale 3x. Poté na tuto položku odpovíme opět špatně, ale penalizace se tentokrát už neuplatní, protože by byla překročena hranice $1x + 2$. Čili při každé další špatné odpovědi již k žádnému navýšení nedojde. Odpovíme-li poté na zkoušenou položku 1x správně, zbývá nám na ni odpovědět správně již pouze 2x. Odpovíme-li poté zase 1x špatně, penalizace se opět uplatní, neboť hranice $1x + 2$ nebude překročena. Tedy musíme opět odpovědět 3x po sobě správně.

Nastavení ve skupině *Automatické zkoušení* bylo blíže vysvětleno v odstavci A.5.3.5.

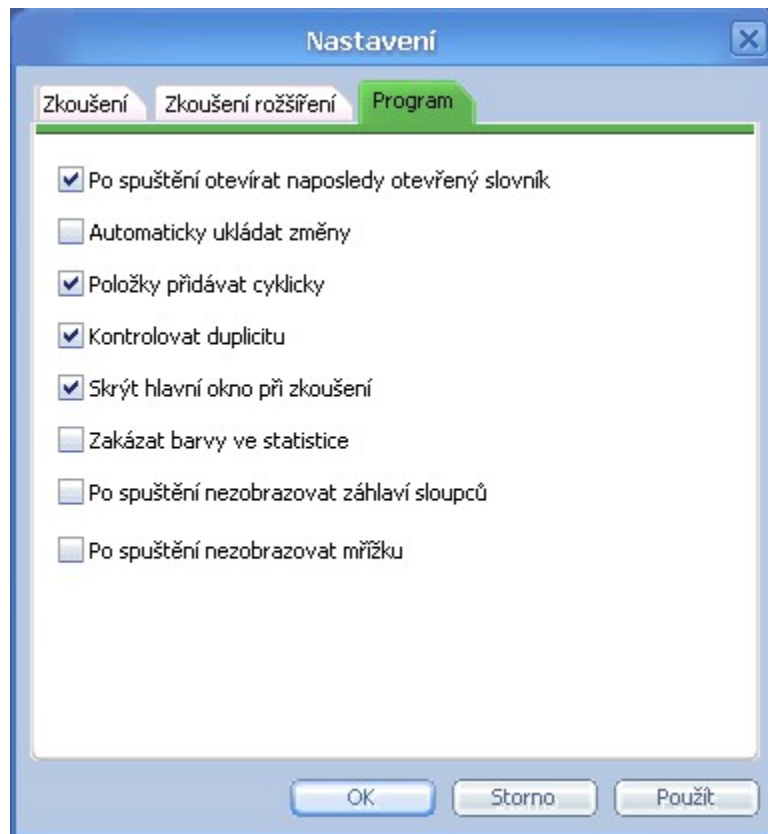
Na kartě *Zkoušení rozšíření* viz. Obrázek A.16 si můžeme navolit, zda-li chceme v průběhu zkoušení zobrazovat výslovnost, zda-li po ukončení zkoušení chceme automaticky zobrazit statistiku, zda-li chceme při zkoušení *Učení* zobrazovat překlad barevně, u zkoušení *Test* si

můžeme navolit počet možností odpovědí v rozsahu 3 až 5 a konečně u zkoušení *Psaní* si můžeme nechat možnost ihned se opravit, což bylo blíže vysvětleno v odstavci A.5.3.3.



Obrázek A.16: Nastavení karta Zkoušení rozšíření

Na kartě *Program* viz. Obrázek A.17 si můžeme navolit, zda-li chceme po spuštění aplikace automaticky otevřít naposledy otevřený slovník, zda-li chceme automaticky ukládat změny před ukončením aplikace, zda-li chceme cyklicky přidávat položky, nebo-li chceme-li, aby se dialogové okno viz. Obrázek A.13 po přidání nové položky automaticky znovu objevilo pro přidání další položky a nemuselo se spouštět manuálně, zda-li chceme při přidávání a editaci kontrolovat, jestli se přidávaná/editovaná položka ve slovníku již nevyskytuje, zda-li chceme skrývat hlavní okno při zkoušení, zda-li chceme zakázat barvy ve statistice a konečně zda-li chceme po spuštění zobrazovat záhlaví sloupců a mřížku v tabulce viz. Obrázek A.19.



Obrázek A.17: Nastavení karta Program

A.5.5 Nápověda

Obsahuje příkaz *O programu Vocab Pro*, který zobrazí informace o verzi aplikace, autorských právech a výrobci.

9.1 Panel nástrojů



Obrázek A.18: Panel nástrojů

Panel nástrojů obsahuje některé příkazy, které jsou funkčně ekvivalentní s hlavní nabídkou. Zleva:

- Nový
- Otevřít
- Uložit
- Spustit Učení
- Spustit Test
- Spustit Psaní
- Otočit
- Automatické zkoušení
- Najít
- Přidat
- Editovat
- Smazat

Najedeme-li na jakoukoliv ikonku myší, zobrazí se plovoucí nápověda, která nám blíže specifikuje význam ikonky.

A.6 Tabulka

| Cizí význam | Výslovnost | Český význam | Poznámky | S1 | S2 | S3 |
|-------------|---------------|------------------|------------------|----|----|----|
| and | [ænd, ənd] | a | | 1 | 0 | 0 |
| big | [big] | velký | | 1 | 0 | 0 |
| but | [bat, bət] | ale | | 1 | 0 | 0 |
| car | [ka:] | auto | | 1 | 0 | 0 |
| dog | [dog] | pes | | 1 | 0 | 0 |
| garage | [gæra:ʒ] | garáž | | 1 | 0 | 0 |
| Harry | [hæri] | Harry | | 1 | 0 | 0 |
| house | [haus] | rodinný dome... | | 1 | 0 | 0 |
| I | [ai] | já | vždy s velkým... | 1 | 0 | 0 |
| I am | [ai æm] | jsem | | 1 | 0 | 0 |
| in | [in] | v, ve | | 1 | 0 | 0 |
| it | [it] | to, ono | o všech věcech | 1 | 0 | 0 |
| it is | [it iz] | je | | 1 | 0 | 0 |
| it isn't | [it iznt] | není | | 1 | 0 | 0 |
| Johnny | [džony] | Honza, Honzík | | 1 | 0 | 0 |
| large | [la:dʒ] | velký | rozlohou | 1 | 0 | 0 |
| lesson | [lesn] | lekce, hodina | vyučovací | 1 | 0 | 0 |
| lesson one | [lesn wan] | první lekce | | 1 | 0 | 0 |
| Mr Mitch | [,mista 'mič] | pan Mitch | | 1 | 0 | 0 |
| my | [mai] | můj, moje | | 1 | 0 | 0 |
| new | [nju:] | nový | | 1 | 0 | 0 |
| nice | [nais] | hezký, pěkný,... | | 1 | 0 | 0 |
| no | [nəu] | ne | | 1 | 0 | 0 |
| not | [not] | zánor u slovesa | | 1 | 0 | 0 |

Obrázek A.19: Tabulka

Význam prvních 4 sloupců je zřejmý. Sloupce S1, S2 a S3 vyjadřují stavy jednotlivých položek a mají následující význam:

S1: Říká, kolikrát je ještě nutno na danou položku odpovědět správně, aby byla uznána za naučenou.

S2: Říká, kolikrát bylo na danou položku v součtu odpovězeno špatně nebo kliknuto na tlačítko Nevím.

S3: Může nabývat třech stavů: 0 – položka nebyla dosud ještě zkoušena, 1 – na položku bylo naposledy odpovězeno správně, 2 – na položku bylo naposledy odpovězeno špatně.

Položky se dají vybírat pomocí levého tlačítka myši. Spolu se stisknutou klávesou Ctrl lze vybrat i více položek najednou, což se může hodit například při exportování nebo mazání. Všechny položky můžeme vybrat pomocí klávesové zkratky Ctrl+A.

Nevejde-li se položka celá do šířky sloupce, je opticky oříznuta a zakončena třemi tečkami. Najedeme-li na ni myší, zobrazí se plovoucí okénko, ve kterém je zobrazená v plné délce.

A.7 Stavový řádek



Obrázek A.20: Stavový řádek

Stavový řádek obsahuje nápovědu, která blíže specifikuje příkaz, na který se najede myší v hlavní nabídce nebo panelu nástrojů. Dále obsahuje čtyři údaje, jejichž význam je následující:

Z (Znám): Počet naučených položek

N (Neznám): Počet nenaučených položek

V (Vyřazeno): Počet vyřazených položek

C (Celkem): Celkový počet položek

B Obsah příloženého CD

| | | |
|------------|------------|--|
| [složka] | Debug | Zkompilovaná aplikace v režimu debug |
| [složka] | Release | Zkompilovaná aplikace v režimu release |
| [složka] | Source | Zdrojové soubory aplikace |
| [složka] | Text_DOC | Text bakalářské práce ve formátu DOC |
| [složka] | Text_PDF | Text bakalářské práce ve formátu PDF |
| [složka] | Vocabulary | Slovníky aplikace |
| [soubor] | Readme.txt | Informace k obsahu CD |