



Středoškolská technika 2011

Setkání a prezentace prací středoškolských studentů na ČVUT

VYUŽITÍ TEXTOVÝCH EDITORŮ V OBLASTI VÝVOJE SOFTWARE

Jan Schupke

EDUCAnet - gymnázium a střední odborná škola České Budějovice, s.r.o.

Čečova 2092/66, 370 04 České Budějovice

EDUCAnet – GYMNÁZIUM A STŘEDNÍ ODBORNÁ ŠKOLA
ČESKÉ BUDĚJOVICE S. R. O.

VYUŽITÍ TEXTOVÝCH EDITORŮ V OBLASTI VÝVOJE SOFTWARE

MATURITNÍ PRÁCE

Vypracoval: Jan Schupke
Vedoucí práce: Ing. Marcel Šťastný

České Budějovice, březen 2011

Prohlašuji, že jsem na maturitní práci pracoval samostatně a že jsem uvedl veškerou literaturu, kterou jsem v této práci použil.

V Českých Budějovicích, dne 31. března 2011

Anotace

Tato práce si klade za cíl podat informace o fungování textových editorů, jejich historii, a zároveň shromáždit a prezentovat informace o procesu tvorby jednoduchého editoru. Výstup maturitní práce tvoří kromě samotného dokumentu také funkční aplikace a její zdrojové kódy.

Teoretická část dokumentu se skládá z úvodu do problematiky, výčtu a stručného popisu některých textových editorů a způsobu ukládání dat (formáty souborů a znakové sady).

Praktická část obsahuje podrobnou analýzu aplikace a plánované funkce. V designu jsou tyto funkce rozvedeny a popsány z technického hlediska.

Summary

The goal of this work is to inform its reader about the functioning of text editors and their history, as well as to present information about a creation process of such software. This work's output consists of, except this document itself, an working application with its source codes.

Theoretical part of this document is composed of a topic introduction, a brief list of some of the text editors and a description of text based data storage concept.

Practical part consists of an detailed analysis of the application as well as an list of its expected features. It also features an application design, which describes those functions in a more technical way.

Poděkování:

Děkuji Ing. Marcelovi Šťastnému, vedoucímu maturitní práce, za ochotu a rady při programování aplikace a zpracování maturitní práce.

Obsah

Úvod.....	7
1 Teoretické poznatky.....	8
1.1 Druhy textových editorů.....	8
1.2 Znakové sady.....	9
1.2.1 ASCII – American Standard Code for Information Interchange.....	9
1.2.2 CP.....	9
1.2.3 Kód Kamenických.....	10
1.2.4 ISO.....	10
1.2.5 Windows.....	10
1.2.6 Unicode.....	10
1.3 Formáty souborů.....	10
1.3.1 TXT.....	11
1.3.2 Rich Text Format.....	11
1.3.3 Doc/ODT.....	11
1.3.4 Portable Document Format.....	11
1.3.5 Další.....	12
1.4 Pozice na trhu software.....	12
1.5 Historický vývoj.....	12
1.5.1 Děrný štítek.....	12
1.5.2 Key punch / Dálnopis.....	13
1.5.3 Nástup textových editorů.....	13
1.5.3.1 Vi.....	13
1.5.3.2 T602.....	13
1.5.3.3 SimpleText.....	13
1.5.3.4 OpenOffice.org Writer.....	13
1.5.3.5 Microsoft Word.....	14
1.5.3.6 PSPad.....	14
1.5.3.7 Eclipse.....	14
2 Praktická část – realizace projektu.....	15
2.1 Specifikace požadavků.....	15
2.2 Analýza.....	15
2.2.1 Základní funkce aplikace.....	15
2.2.2 Panely.....	16
2.2.2.1 Menu.....	16
2.2.2.2 Nástrojové lišty.....	17
2.2.2.3 Stavový řádek.....	18
2.3 Design.....	19
2.3.1 Výběh technologií.....	19
2.3.2 Použité componenty a postup při vývoji.....	19
2.3.3 Rozvržení okna.....	19
2.3.3.1 Layout.....	19
2.3.3.2 Pozicování.....	20
2.3.4 Souborová struktura aplikace.....	20
2.3.4.1 Actions.....	20
2.3.4.2 Config.....	20
2.3.4.3 Dialog.....	20

2.3.4.4 Filehandling.....	21
2.3.4.5 GUI.....	21
2.3.4.6 Layout.....	21
2.3.5 Konfigurační soubor a jazykové mutace.....	21
2.3.5.1 Znaková sada jazykových souborů.....	22
2.3.6 Podpora záložek.....	22
2.3.7 Klávesové zkratky.....	22
2.3.8 Podpora listování v historii změn.....	22
2.3.9 Vyhledávání v textu.....	23
2.3.10 Distribuce aplikace.....	23
2.3.11 Struktura dokumentace.....	24
2.4 Možnosti dalšího rozvoje programu.....	24
2.4.1 Odsazování textu.....	24
2.4.2 Zvýrazňování syntaxe.....	25
2.4.3 Pravítka.....	25
2.4.4 Další nastavení.....	25
2.4.4.1 Velikost písma.....	25
2.4.4.2 Vkládání vlastních jazykových souborů.....	25
3 Závěr.....	26
3.1 Výsledná aplikace.....	26
3.2 Přínos práce.....	27
4 Rejstřík.....	28
5 Seznam použitých internetových zdrojů.....	29
6 Přílohy.....	31

Úvod

Na trhu je velké množství textových editorů s nejrůznějšími funkcemi, proto by se mohla jevit volba vývoje zrovna tohoto typu software jako ne moc nápaditá. Můj důvod byl, kromě zájmu získat zkušenosti s programováním GUI, vytvořit jednoduše přizpůsobitelný editor s klíčovými funkcemi, který by nebyl závislý na konkrétní platformě.

Co se týče teoretických informací, práce si klade za cíl zformulovat stručnou historii vývoje textových editorů až po současnost, a zároveň informovat o vlastnostech a způsobu použití nejrozšířenějšíh aplikací daného druhu.

1 Teoretické poznatky

1.1 Druhy textových editorů

Textové editory se dají dělit mnoha způsoby. Hlavní dvě skupiny tvoří wysiwyg a neformátující editory.

Hlavní výhodou wysiwyg (formátujících) editorů je fakt, že výsledný dokument, včetně všech grafických úprav, je možné vidět hned v editoru, bez jakéhokoli exportu. Tento typ software se používá hlavně v kancelářském provozu. Hlavní nevýhodou těchto editorů je zakódovanost souboru, kdy je třeba každý soubor otevřít přímo v programu, ve kterém byl vytvořen, nebo použít jiný software, který ale musí podporovat formát souboru. Tomuto typu software se také říká textový procesor.

Dalším druhem jsou neformátující (plain text) editory. Jak je už z názvu jasné, pracují jen se soubory, které obsahují čistý text. Výhodou výstupních souborů typu plain text je jejich mnohdy zanedbatelná velikost, která je určena jen počtem znaků v souboru. Tento typ editorů se hojně využívá při editaci nejrůznějších zdrojových nebo konfiguračních souborů, protože právě tyto jsou řešeny výhradně plain textem.

Plain text editory se dále dělí podle toho, zda mají grafické rozhraní. Typickým příkladem editoru bez grafického rozhraní je Linuxový editor Vi/Vim, který se sice dá spustit v okně, ale je k tomu zapotřebí extra knihovna a navíc to oproti základní textové verzi nepřináší žádné nové funkce. U těchto editorů chybí jakákoli menu navigace a vše se provádí pomocí textových, nebo přesněji několikanoznakových, instrukcí.

Výhodou tohoto typu software je jejich potenciál. S podporou regulárních výrazů a instrukcí pro cyklickou editaci textu je možné upravit neomezené množství textu při téměř nulové námaze a vysoké efektivnosti. Z toho ale zároveň vyplývá jejich nevýhoda, kterou je nutná znalost příkazů, bez kterých se s takovým editorem nedá pracovat.

Graficky řešené editory nabízejí komfort možnosti ovládní myší, použití menu atd., což sice zpomaluje práci, ale je to mnohem jednodušší cesta pro nezkušeného uživatele. Zároveň jsou efektivní při psaní zdrojových kódů, kdy zvýrazňují text podle typu souboru (programovacího jazyka), odsazují text uvnitř bloků a některé umožňují našeptávání tříd atd. Tyto editory jsou mnohem hojněji využívány, proto jich taky existuje mnohem více.

Další specifický druh software pro práci s textem jsou DTP editory. Jsou určeny především pro profesionální nasazení, kde je potřeba využít možností pro složitější sazbu textu.

1.2 Znakové sady

Znaková sada je seznam určitých znaků, kterým je přiřazena číselná reprezentace za účelem uložení v paměti počítače. [1]

1.2.1 ASCII – American Standard Code for Information Interchange

ASCII je tabulka znaků, která definuje anglickou abecedu, číslice a některé další znaky, včetně řídicích. Je to první a neúspěšnější znaková sada. Novější sady vycházejí právě z ní. [2]

Původní ASCII je sedmibitový – má kapacitu 128 znaků. Tyto znaky jsou zachované i u všech osmibitových rozšíření (evropské jazyky atd.), které definují své vlastní znaky pro hodnoty vyšší než 127. [2]

1.2.2 CP

Znakové sady CP jsou kompatibilní s ASCII a jsou použité v systémech DOS. Původní znaková sada je CP437, východoevropská verze potom CP852.

1.2.3 Kód Kamenických

Kód Kamenických je znaková sada pro MS-DOS. Byla vytvořena bratry Jiřím a Marianem Kamenickými v Československu. Je to upravená sada CP437, kde byly znaky 128-171 nahrazeny specifickými znaky české a slovenské abecedy (diakritika). [3]

Znaky byly umístěny na pozice vzhledově podobných původních znaků. To mělo docílit čitelnost dokumentu i na počítačích s původní sadou. [3]

1.2.4 ISO

ISO/IEC 8859-X je osmibitový standard, používaný převážně v Unixových systémech.

1.2.5 Windows

Windows-125X jsou znakové sady používané v systémech Windows. Vycházejí ze standardu ISO. Středoevropská (tedy i česká) sada je označena jako Windows-1250.

1.2.6 Unicode

UTF je znaková sada, jejíž cílem je vytvořit univerzální tabulku znaků, která by byla schopná pojmout všechny znaky a být tak plně kompatibilní s každým jazykem. [4] [5]

Dělí se na UTF-8, kde je jeden znak reprezentovaný bajtem, a UTF-16, kde každý znak zabírá dva bajty. [4] [5]

1.3 Formáty souborů

Každý soubor, včetně textových, je uložený v nějakém formátu – kromě samotného obsahu nese soubor i informace o případném grafickém zpracování, může obsahovat obrázky atd. Uvádím hlavní formáty textových souborů:

1.3.1 TXT

TXT, nebo-li plain text, je čistě textový a neformátovaný soubor. Hlavní výhodou je to, že každý, včetně komplexnějších editorů, umí pracovat s čistým textem, proto je možné soubor tohoto formátu otevřít na prakticky jakémkoli zobrazovacím zařízení.

1.3.2 Rich Text Format

RTF je formát pro jednoduše formátované texty. Je vyvíjený společností Microsoft. RTF soubory je možné číst i v pouhém plaintext editoru, protože se formátování neprovádí zakódováním celého souboru, ale přidáním speciálních příkazů do čistého textu, které se na zobrazovacím zařízení s podporou čtení RTF zpracují a zformátují text.

1.3.3 Doc/ODT

Tyto dva formáty představují nejběžněji používané formáty pro graficky upravený text. Umožňují v sobě nést další objekty, jako například obrázky, tabulky nebo grafy. Narozdíl od .RTF nejsou čitelné pro plaintext editory.

Doc je vyvíjený společností Microsoft a využíváný hlavně v jejich editoru MS Word. ODT se používá při vytváření souborů v aplikaci OpenOffice.org Writer.

1.3.4 Portable Document Format

PDF je souborový formát společnosti Adobe. Jeho hlavní výhodou je přenositelnost mezi zařízeními. PDF soubor může obsahovat jak formátování, tak i jiné netextové objekty, přičemž zaručuje stejné zobrazení nezávisle na hardware nebo software zařízení, na kterém je zobrazovaný. Jedinou podmínkou pro čtení PDF je nainstalovaný editor/čtečka těchto souborů. [6]

Formát je založený na jazyce PostScript, což je značkový jazyk určený ke grafickému popisu dokumentů. [6]

1.3.5 Další

Při programování se setkáváme s vysokým množstvím nejrůznějších přípon souborů, které vyjadřují, jakým programovacím jazykem je daný soubor psaný. V podstatě ale nejde o nic jiného, než plain text soubory, které je díky jejich příponě možné jednodušeji zpracovat aplikacemi jako HTTP server atd.

Přípona souboru zároveň ulehčuje editoru přiřadit zdrojovému kódu správnou šablonu pro zvýraznění syntaxe.

1.4 Pozice na trhu software

Textový editor patří do základní výbavy každého osobního počítače. Málo editorů se ale prodává samostatně. Wysiwyg editory jsou většinou součástí celého office balíku (v dnešní době jsou nejrozšířenější verze OpenOffice a Microsoft Office), kvalitní vývojářské aplikace obsahují kromě samotného editoru další funkce, jako debugger a compiler. Tento software se označuje jako IDE. Jednoduchých plain text editorů je na trhu velké množství, a i ty kvalitnější jsou dostupné zdarma.

1.5 Historický vývoj

1.5.1 Děrný štítek

Před nástupem modernějších médií se počítačová data ukládala na tzv. Děrné štítky. Jsou vyrobeny z tenkého kartonu a data jsou na nich reprezentována vyražením dírek na určitou pozici. [7]

Nejběžnějším typem jsou osmdesátisloupcové štítky, vyráběly se ale i devadesátisloupcové. Prvně se děrné štítky začaly využívat v 19. století a používaly se až do 80. let 20. století. Nástupcem děrných štítků byly děrné pásky, později diskety. [7]

1.5.2 Key punch / Dálnopis

Key punch je zařízení, používané pro zápis dat na děrné štítky. První zařízení tohoto typu byla plně manuální, později se začal proces automatizovat použitím modernější klávesnice.

1.5.3 Nástup textových editorů

Uvádím příklady některých důležitých editorů, včetně jejich stručné charakteristiky a popisu některých vlastností:

1.5.3.1 Vi

Vi je editor určený pro textové prostředí. V základní verzi nepodporuje žádné grafické prvky, včetně nabídek. Veškerá komunikace uživatele s aplikací probíhá pomocí příkazů, které uživatel zadává v “needitačním/příkazovém” režimu. Program byl původně napsaný roku 1976 pro systém BSD. [8]

1.5.3.2 T602

T602 je český WYSIWYG editor pro MS-DOS. První verze vznikla roku 1989 a byla až do roku 1995 udržovaná firmou Software602. Výhodou T602 byla podpora jak diakritiky, tak i jiných národních znaků. Dále umožňoval funkce jako kopírování textu, vkládání obrázků, tabulek apod. [9]

1.5.3.3 SimpleText

SimpleText je nativní textový editor pro MacOS. Umožňuje formátování textu. Bývá nahrazen modernější aplikací TextEdit. [10]

1.5.3.4 OpenOffice.org Writer

OoO je kancelářský balík, který obsahuje několik aplikací. Pro práci s textem se používá textový procesor Writer. Celý projekt je open source, vydávaný pod licencí LGPL. Je multiplatformní a existují verze pro všechny rozšířené operační systémy. [11]

Podporuje mnoho formátů. V základním nastavení ukládá soubory do formátu OpenDocument (textové soubory .odt) a nativně podporuje exportování do formátu .pdf. [11]

1.5.3.5 Microsoft Word

Word je součástí kancelářského balíku Microsoft Office, vydávaného firmou Microsoft. Původní verze byla pro systém DOS. Aplikace je oficiálně podporována a prodávána jen pro systémy Windows a Mac OS. Narozdíl od OpenOffice není otevřený ani program, ani jeho formát .doc. Postrádá také možnost exportu do .pdf.

Celkově jsou si programy Writer a Word funkcemi velmi podobné, liší se jen v uživatelském rozhraní.

1.5.3.6 PSPad

Tuto aplikaci jsem vybral jako reprezentanta plaintext editorů, optimalizovaných pro editaci zdrojových kódů. Program je šířený jako freeware a je vyvíjený českým vývojářem.

Tato aplikace sloužila jako hlavní inspirace pro mnou vyvíjený program.

1.5.3.7 Eclipse

Eclipse je propracované vývojové prostředí, zaměřené hlavně na vývoj Java aplikací. Je možné aplikaci rozšířit pro podporu jiných jazyků. Stejně tak není problém doinstalovat množství plug-inů.

Projekt je open source. Kromě samotného vývojového prostředí obsahuje i překladač zdrojových kódů a debugger.

2 Praktická část – realizace projektu

2.1 Specifikace požadavků

Cílem je vytvořit plain text editor se základními funkcemi pro ulehčení psaní a editace zdrojových kódů.

Bude obsahovat základní funkce pro práci s textem, dále možnost vyhledávání a listování v historii změn textového pole.

Aplikace bude umožňovat jednoduché nastavení chování okna a textového pole. Dále bude umožňovat změnu jazyka.

Aplikace měla původně obsahovat i rozhraní pro editaci grafických souborů, které ale bylo z analýzy odstraněno pro podceněný odhad časové náročnosti vývoje. Důvodem tohoto kroku bylo umožnění kvalitnějšího zpracování textového editoru.

2.2 Analýza

2.2.1 Základní funkce aplikace

Editor bude zvládat základní obsluhu neformátovaných textových souborů, listování v historii změn souboru a hledání. Aplikace bude nastavitelná podle volby uživatele. Detaily funkcí jsou popsány v dalších odstavcích.

Práce bude probíhat v jednom okně s více záložkami.

Před každou důležitou akcí bude třeba potvrdit záměr (vyskakovací okno “yesno”). Tyto akce jsou zejména:

1. Zavření panelu s neuloženou prací.
2. Přepisování souborů při ukládání (“Uložit jako”).

Většina funkcí bude přístupná přes určitou klávesovou zkratku. Běžné akce (kopírování, vkládání, označování textu etc.) budou dostupné přes jejich standardní zkratky, specifické funkce programu budou mít svoje.

2.2.2 Panely

2.2.2.1 Menu

Nabídka menu bude vypadat následovně:

- 1) Soubor – práce se soubory jako celky:
 1. Nový – Vytvoří novou záložku. Vytvoří se nová instance editačního pole, stavové lišty a levé lišty nástrojů. Jméno dokumentu se implicitně nastaví jako “Dokument”. Toto jméno se po uložení změní na patřičný název.
 2. Otevřít – Otevře soubor do nové záložky. Stejně jako při zakládání nového souboru se vytvoří instance celé nové záložky. Načte se požadovaný soubor podle výběru z dialogu, do textového pole se vypíše obsah souboru a do stavového řádku a popisku tabu se vypíše cesta souboru v souborovém systému, nebo jenom samotný název. Tato funkce je nastavitelná ve vlastnostech aplikace.
 3. Uložit – Uloží aktuální soubor. Soubory se ukládají jako plain text. Položka uložit se pokusí o rychlé uložení a přepsání existujícího souboru. Pokud ještě žádný soubor neexistuje, zavolá metodu pro “Uložit jako”, která je na tento případ připravená.
 4. Uložit jako – Otevře nabídku pro specifikaci uložení. Tato metoda otevře nabídku pokaždé, i když už je soubor na disku uložený. Pokud se uživatel pokusí uložit soubor se jménem, které už na aktuální pozici v souborovém systému existuje, aplikace ho vyzve k potvrzení akce, aby nedošlo k nechtěnému přepsání.

5. Uložit vše – Uloží všechny otevřené záložky. Aplikace jednoduše zavolá možnost “Uložit” pro všechny aktuálně otevřené soubory.
- 2) Úpravy – upravování jednotlivých souborů:
 1. Zpět – Hodnota textového pole se vrátí o jednu akci zpět v historii.
 2. Vpřed – Hodnota textového pole se vyplní textem, který je o jeden krok napřed vzhledem k aktuálnímu stavu historie. Pokud další krok neexistuje, neprovede se nic.
 3. Vyhledat – Tato položka otevře dialog s textovým polem a tlačítky pro vyhledání zadaného řetězce v otevřeném a aktivním tabu.
 - 3) Vlastnosti:
 1. Nastavení aplikace – Tato položka obsahuje možnosti pro nastavení jazyka, volbu mezi zobrazením cesty nebo jména souboru ve stavovém řádku a titulku tabu a volbu pro automatické zalamování řádků, které jsou delší než délka textového pole. Zároveň obsahuje nabídku pro volbu jazyka. Změny, provedené v tomto dialogu, se projeví až po opětovném spuštění aplikace.
 - 4) Nápověda – informace o aplikaci a dokumentace:
 1. Dialog s názvem aplikace, její verzí a případně dalšími informacemi.
 2. Dokumentace s nápovědou k použití, ve formátu HTML, spouští se v podokně aplikace.

2.2.2.2 Nástrojové lišty

Aplikace bude obsahovat dvě lišty nástrojů. Jedna bude obecná pro celou aplikaci a bude obsahovat obecná tlačítka, která ovlivňují program jako celek.

Druhá lišta bude specifická pro každou záložku/soubor zvlášť a bude obsahovat tlačítka, která přímo ovlivní jen daný konkrétní soubor.

Lišty budou obsahovat funkce z menu, které budou reprezentované formou ikon. Akce, které se týkají aktuálního souboru, budou umístěné nalevo od textového pole. Akce pro

celou aplikaci a nezávislé na jednotlivých souborech budou v panelu nad seznamem záložek.

2.2.2.3 Stavový řádek

Bude umístěn ve spodní části okna a bude ukazovat informace o názvu souborů a počtu znaků.

2.3 Design

2.3.1 Výběr technologií

Aplikace je tvořena ve standardní edici Javy 5. Javu jsem vybral pro její platformní nezávislost, objektový přístup a dobrou podporu, ať už z hlediska dokumentace, nebo rozšíření na stanicích.

Grafické rozhraní je řešeno pomocí knihovny Swing.

Dokumentace je tvořena pomocí HTML pro jeho možnost jednoduše použít hyperlink a případně zveřejnit dokumentaci online.

2.3.2 Použité componenty a postup při vývoji

Při psaní aplikace jsou použity běžné componenty (JPanel, JMenu, JButton atd.) Všechny uživatelem vyvolané akce jsou obsluhované samostatnou třídou, jejíž metody se mohou volat z několika míst. Typický příklad je položka menu a tlačítko, které mají obě stejnou funkci.

Sledování akcí je řešeno pomocí listenerů. Ve většině případů je to ActionListener, u textového pole potom KeyListener atd.

2.3.3 Rozvržení okna

Při umisťování komponent do okna aplikace je využito dvou metod:

2.3.3.1 Layout

Pro panely a hlavní prvky aplikace je použit BorderLayout, což umožňuje jednoduché a přizpůsobivé umístění prvků.

Pro menší componenty jako tlačítka nebo textové popisky je potom použit FlowLayout, což nedává možnost nastavit komponentám pozici, ale to u těchto prvků není důležité, protože je vhodné je řadit za sebe, a to je přesně způsob, jakým FlowLayout umisťuje prvky.

Pro lišty s tlačítky pro jednotlivé soubory je aplikovaná třída `VerticalLayout`, což je třída třetí strany, která se chová jako `FlowLayout`, jenom neřadí prvky horizontálně, ale vertikálně.

2.3.3.2 Pozicování

Druhou metodou je vynechání layoutu a pozicování prvků přímo. To je dosaženo nastavením rozměrů komponent metodou `setBounds()`.

2.3.4 Souborová struktura aplikace

Program je strukturovaný v jednom hlavním balíku, který obsahuje šest podbalíčků s třídami se specifickým využitím.

Detailní popis všech tříd je možné vidět v class diagramu aplikace (příloha 4). Kód, a tedy i diagram, obsahují některé třídy, které ve výsledné aplikaci momentálně nejsou použité, ale nebyl důvod je úplně odstraňovat. Mohou sloužit jako základ pro některá rozšíření aplikace, která jsou navržena v závěru práce.

2.3.4.1 Actions

Tento podbalíček obsahuje dvě třídy, které obsahují metody pro obsluhu veškerých akcí aplikace. Tyto metody jsou volané pomocí listenerů připojených k jednotlivým položkám programu.

2.3.4.2 Config

Tento balíček obsahuje vzhledem k rozsahu aplikace jen jednu třídu. Účelem této třídy je zařídit ukládání konfiguračních dat do souboru v domovském adresáři uživatele.

2.3.4.3 Dialog

V tomto balíčku jsou umístěné třídy složitějších dialogů. Jsou to dialogy pro nastavení a vyhledávání, a informační okna pro menu "Nápověda".

2.3.4.4 Filehandling

Filehandling je balík pro třídy na obsluhu souborů. Jsou zde umístěné třídy s metodami pro čtení a zápis textu do souboru.

2.3.4.5 GUI

Balík rozhraní obsahuje všechny prvky okna. Jsou to panely, menu, stavový řádek a prvky editačního pole.

2.3.4.6 Layout

Tento balík existuje jen z důvodu oddělení knihovny třetí strany, která slouží pro řešení problému seřazení ikon pod sebou. Tato knihovna byla ale dostupná jen ve formě .java souboru.

2.3.5 Konfigurační soubor a jazykové mutace

Aplikace umožňuje nastavení několika parametrů uživatelem. Hodnoty nastavení si pamatuje i po uzavření a aplikuje při opětovném spuštění. Toto je řešeno pomocí properties souboru s konfiguračními hodnotami. Konfigurační soubor je uložený v domovském adresáři. Pro každého uživatele existuje vlastní konfigurační soubor.

Stejným způsobem se řeší nastavení jazyka, kdy jsou veškeré texty uložené s externím properties souboru a vždy volány v případě potřeby. Pro každý použitý jazyk existuje jeden soubor s texty.

Konfigurace aplikace probíhá ve zvláštním dialogovém okně, které dědí od JDialog a je podoknem aplikace. Obsahuje seznam podporovaných jazyků (JComboBox) a JCheckBoxy pro nastavení zalamování řádků, volby automatické maximalizace okna, a způsobu výpisu jména souboru. Dále je zde vypsána cesta ke konfiguračnímu souboru, pro jeho jednodušší nalezení.

Potvrzení, resp. zrušení volby probíhá stiskem tlačítek “OK” nebo “Zrušit”.

Po stisku “OK” se hodnoty komponent uloží do proměnné obsahující konfigurační hodnoty a následně se zapíše do zmiňovaného souboru. Tento soubor se při každém spuštění aplikace automaticky načte a hodnoty v něm uložené se aplikují.

Pokud soubor při spuštění aplikace neexistuje, vytvoří se nový, prázdný.

2.3.5.1 Znaková sada jazykových souborů

Samotný obsah jazykových dokumentů musí být převedený do číselné ASCII reprezentace, protože způsob načítání textů do aplikace (ResourceBundle) podporuje jen znakovou sadu 8859-1 a při načítání českých znaků dochází k rozkódování.

Pro správné zobrazení českých znaků v aplikaci je třeba textový soubor konvertovat například nástrojem native2ascii, který zajišťuje převod problematických znaků.

Tento nástroj je součástí JDK a spouští se přes příkazovou řádku.

2.3.6 Podpora záložek

Editor pracuje s několika soubory současně. Každý jednotlivý soubor má svou instanci tabu, na kterém jsou umístěny ovládací prvky a samotné editační pole. Pro zajištění funkčnosti této vlastnosti je využito třídy JTabbedPane, která je součástí knihovny Swing.

Třída nepodporuje zobrazení tlačítka pro uzavření na každém tabu. Místo přidávání tlačítka na záložky je tlačítko umístěno do lišty nástrojů každého souboru.

2.3.7 Klávesové zkratky

Každé tlačítko má nastavenou specifickou zkratku řešenou kombinací <ctrl>, <shift> a daných znaků. Seznam zkratek je dostupný v dokumentaci programu.

Vzhledem k tomu, že se tlačítkům nedá jednoduše přiřadit klávesová zkratka, je třeba každému tlačítku přiřadit vlastní akci, nastavit hodnotu KeyStroke a samotnou vykonávající metodu zavolat až uvnitř této konstrukce.

2.3.8 Podpora listování v historii změn

Aplikace podporuje vrácení naposledy provedených změn v textovém poli pomocí tlačítek “Zpět” a “Vpřed”. Pro tyto funkce využívá aplikace vestavěného rozhraní Javy

UndoableEditListener.

2.3.9 Vyhledávání v textu

Vyhledávání využívá přednastavených metod pro práci s řetězcem a textovým polem. Při otevření souboru se nastaví pozice kurzoru na začátek dokumentu, protože výchozí hodnota je poslední znak.

Při vyhledávání se prochází text a pomocí metod `indexOf()` a `setCaretPosition()` se nastaví kurzor na pozici prvního výskytu vyhovujícího řetězce. Potom se označí text následující od kurzoru o délce počtu znaků vyhledávaného textu.

Po dosažení konce dokumentu aplikace vypíše informační hlášku.

2.3.10 Distribuce aplikace

Program je dodáván jako Java archiv (.jar). Kromě samotného archivu obsahuje balík ještě textový dokument s úplným zněním LGPL, HTML dokumentaci, která je přístupná buďto přímo otevřením .html souboru, nebo přes menu “Nápověda” v aplikaci.

Dokumentace je psaná jen anglicky. Jazykové mutace samotného programu se na ni nevztahují. V souborech jsou použité jen HTML a CSS technologie, takže pro čtení není potřeba žádný HTTP server.

Dále balík obsahuje spouštěcí skripty pro systémy Windows (.bat) a Unix (.sh). Obsahují příkaz, který správně zavolá Java archiv a spustí aplikaci.

2.3.11 Struktura dokumentace

Dokumentace obsahuje popis aplikace, výpis všech důležitých vlastností a návod k jejich obsluze. Dále jsou zde uvedeny všechny použitelné klávesové zkratky, aktuální verze programu a seznam známých chyb.

Vše je napsané na jedné stránce a volané do dialogového okna aplikace.

2.4 Možnosti dalšího rozvoje programu

Výslednou aplikaci by bylo možné rozšířit několika užitečnými funkcemi, které by přispěly ke zjednodušení práce.

Následující navrhované funkce nejsou z časových důvodů implementované do současné verze aplikace, ale při případné další aktualizaci programu budou mít vysokou prioritu.

2.4.1 Odsazování textu

Odsazování textu je funkce, která sleduje zadané znaky, a podle některých kritérií odsadí (při odřádkování) odsazení následujícího řádku o určitý počet mezer nebo tabů. Zároveň udržuje aktuální pozici a při přechodu na nový řádek automaticky aplikuje odsazení předešlého.

Mezi sledované znaky patří především složené závorky, které indikují začátek a konec bloků. Dále potom znaky složených komentářů, které usnadňují formátování většího bloku komentovaného textu.

Dále by se vlastnost dala rozšířit na umožnění parsování celého dokumentu a automatického odsazení, což je užitečné pro zvýšení přehlednosti zejména při čtení nebo úpravě cizího kódu, který není zformátovaný podle konvencí, nebo neobsahuje odsazení vůbec.

2.4.2 Zvýrazňování syntaxe

Textové dokumenty, které obsahují zdrojový kód nějakého programovacího nebo značkovacího jazyka je vhodné barevně zvýraznit pro zvýšení čitelnosti. Barevné rozlišení neovlivňuje samotný dokument, protože se do něj neukládá, jen se aplikuje editorem při otevření souboru.

Zvýraznění se běžně aplikuje na klíčová slova daného jazyka, řetězce mezi úvozovkami a komentáře. Dále mohou být barevně zvýrazněné například proměnné a konstanty.

Nejvhodnější a pravděpodobně nejefektivnější způsob aplikace této funkce do programu by byla implementace hotové knihovny třetí strany, která by obsahovala kompletní výkonný kód a zvýraznila by obsah textového pole, nebo by rovnou aplikovala vlastní textové pole.

2.4.3 Pravítka

Posledním podstatným prvkem pro zvýšení přehlednosti při čtení jsou pravítka, která by byla umístěná horizontálně a vertikálně vedle textového pole a udávala by čísla řádků, resp. počet znaků od levého okraje.

2.4.4 Další nastavení

Následující funkce rozšiřují možnosti dialogového okna pro nastavení.

2.4.4.1 Velikost písma

Momentálně je v aplikaci nastavená pevná velikost písma v textovém poli. Někteří uživatelé by mohli přivítat možnost ruční změny velikosti. Nabídku pro změnu by bylo pravděpodobně nejvhodnější umístit do dialogu nastavení aplikace jako rolovací seznam, který by změnu aplikoval na všechny záložky.

2.4.4.2 Vkládání vlastních jazykových souborů

Aplikace momentálně obsahuje dva nastavitelné jazyky. Otevřením možnosti vkládání vlastních souborů s texty by se otevřela možnost ostatních uživatelů přispět k rozvoji aplikace.

Nejjednodušším způsobem by zřejmě bylo zveřejnit strukturu stávajících souborů pro možnost editace a umožnit upload uživatelem upravených souborů z disku.

3 Závěr

3.1 Výsledná aplikace

Celý program byl tvořený se záměrem využití klíčových funkcí, které budou co nejjednodušeji ovladatelné a snadno přístupné. Veškeré vlastnosti aplikace jsou dostupné jak přes hlavní menu aplikace a dvě lišty nástrojů po stranách textového pole, tak i přes vlastní klávesovou zkratku. Seznam klávesových zkratk je vypsán v dokumentaci (příloha 2).

Okno aplikace je v základním nastavení vykreslované s malými rozměry (příloha 1), aby mohl editor sloužit jako zápisník poznámek např. z jiného okna, které by tedy nemělo být editorem zakryté. Toto je samozřejmě možné změnit a ve vlastnostech aplikace nastavit automatické spouštění v maximalizovaném stavu (příloha 3).

Kromě zmiňované velikosti okna je ošetřené i přetékaní řádků za okraj textového pole, které často působí matoucím dojmem. Někteří uživatelé ocení automatické zalamování a vypisování veškerého textu do editačního pole, proto je k dispozici pro nastavení i tato možnost.

Pro představu o rozsahu otevřeného souboru nabízí aplikace počítadlo znaků v zápatí okna. Toto počítadlo samozřejmě generuje číslo pro každou záložku zvlášť.

Volba jazyka je v dnešní době samozřejmostí, proto aplikace v prezentované verzi nabízí dva jazyky na výběr. S přístupem ke zdrojovým kódům by pro nikoho neměl být problém vytvořit si vlastní překlad.

Dialog nastavení je samozřejmě také proveden co nejpřehledněji a mimo jiné zobrazuje i absolutní cestu ke konfiguračnímu souboru aplikace.

Text dokumentace, tvořený HTML souborem, je čitelný i externě za použití webového prohlížeče. Pro omezené možnosti formátování v dialogu okna Javy jsou ovšem použity jen základní formátovací značky, proto by se dokumentace mohla zobrazovat s malými rozdíly, které by ovšem neměly zhoršovat celkovou čitelnost.

3.2 Přínos práce

I když práce obsahuje určité teoretické informace z oblasti existujícího software pro úpravu textových souborů, hlavním cílem práce bylo samotné programování, prezentace výsledku a jeho zhodnocení.

Všechny implementované vlastnosti programů jsou funkční podle očekávání a studiem zdrojových kódů je možné zjistit způsoby řešení mnoha problémů, které by bez podrobnějšího promyšlení nemusely být zřejmé.

Kromě uvedených vlastností je v práci zmíněna i řada dalších námětů pro úpravu a rozšíření aplikace, které mohou sloužit jako motivace. I když tyto funkce z časových důvodů nebyly implementovány a aplikace proto není rozvedena do takové míry, jaký byl původní záměr, považuji získané a prezentované informace za přínos.

4 Rejstřík

GUI - Graphic User Interface

WYSIWYG – What You See Is What You Get

DTP – Desktop Publishing

ASCII – American Standard Code for Information Interchange

CP – Code Page

UTF - UCS Transformation Format

RTF – Rich Text Format

ODF – Open Document Format

ODT – Open Document Text

PDF – Portable Document Format

HTTP – Hypertext Transfer Protocol

IDE – Integrated Development Environment

BSD – Berkley Software Distribution

DOS – Disk Operating System

LGPL – Lesser General Public Licence

HTML – Hypertext Markup Language

OOo – OpenOffice.org

IEC – International Electrotechnical Commission

5 Seznam použitých internetových zdrojů

- [1] Wikipedia: Znakové sady [cit. 2011-03-16]
URL: <http://cs.wikipedia.org/wiki/Znaková_sada>
- [2] Wikipedia: ASCII [cit. 2011-03-16]
URL: <<http://cs.wikipedia.org/wiki/ASCII>>
- [3] Wikipedia: Kód Kamenických [cit. 2011-03-16]
URL: <http://cs.wikipedia.org/wiki/Kód_Kamenických>
- [4] Wikipedia: UTF-8 [cit. 2011-03-16]
URL: <<http://cs.wikipedia.org/wiki/UTF-8>>
- [5] Wikipedia: Unicode [cit. 2011-03-16]
URL: <<http://cs.wikipedia.org/wiki/Unicode>>
- [6] Wikipedia: PDF [cit. 2011-03-16]
URL: <<http://cs.wikipedia.org/wiki/Pdf>>
- [7] Wikipedia: Děrný štítek [cit. 2011-03-16]
URL: <http://cs.wikipedia.org/wiki/Děrný_štítek>
- [8] Wikipedia: Vi [cit. 2011-03-16]
URL: <<http://cs.wikipedia.org/wiki/Vi>>
- [9] Wikipedia: T602 [cit. 2011-03-16]
URL: <<http://cs.wikipedia.org/wiki/T602>>
- [10] Wikipedia: SimpleText [cit. 2011-03-16]
URL: <<http://en.wikipedia.org/wiki/Simpletext>>
- [11] Wikipedia: OpenOffice.org [cit. 2011-03-16]
URL: <<http://cs.wikipedia.org/wiki/OpenOffice.org>>
- [12] Knihovna FlowLayout [cit. 2011-03-16]
URL: <<http://www.java2s.com/Code/Java/Swing-JFC/AverticallayoutmanagersimilartojavaawtFlowLayout.htm>>
- [13] Knihovna ikon “Tango Icon Library” [cit. 2011-03-16]
URL: <http://tango.freedesktop.org/Tango_Icon_Library>
- [14] Příklad fungování třídy UndoableTextArea [cit. 2011-03-16]
URL: <<http://www.javareference.com/jrexamples/viewexample.jsp?id=16>>

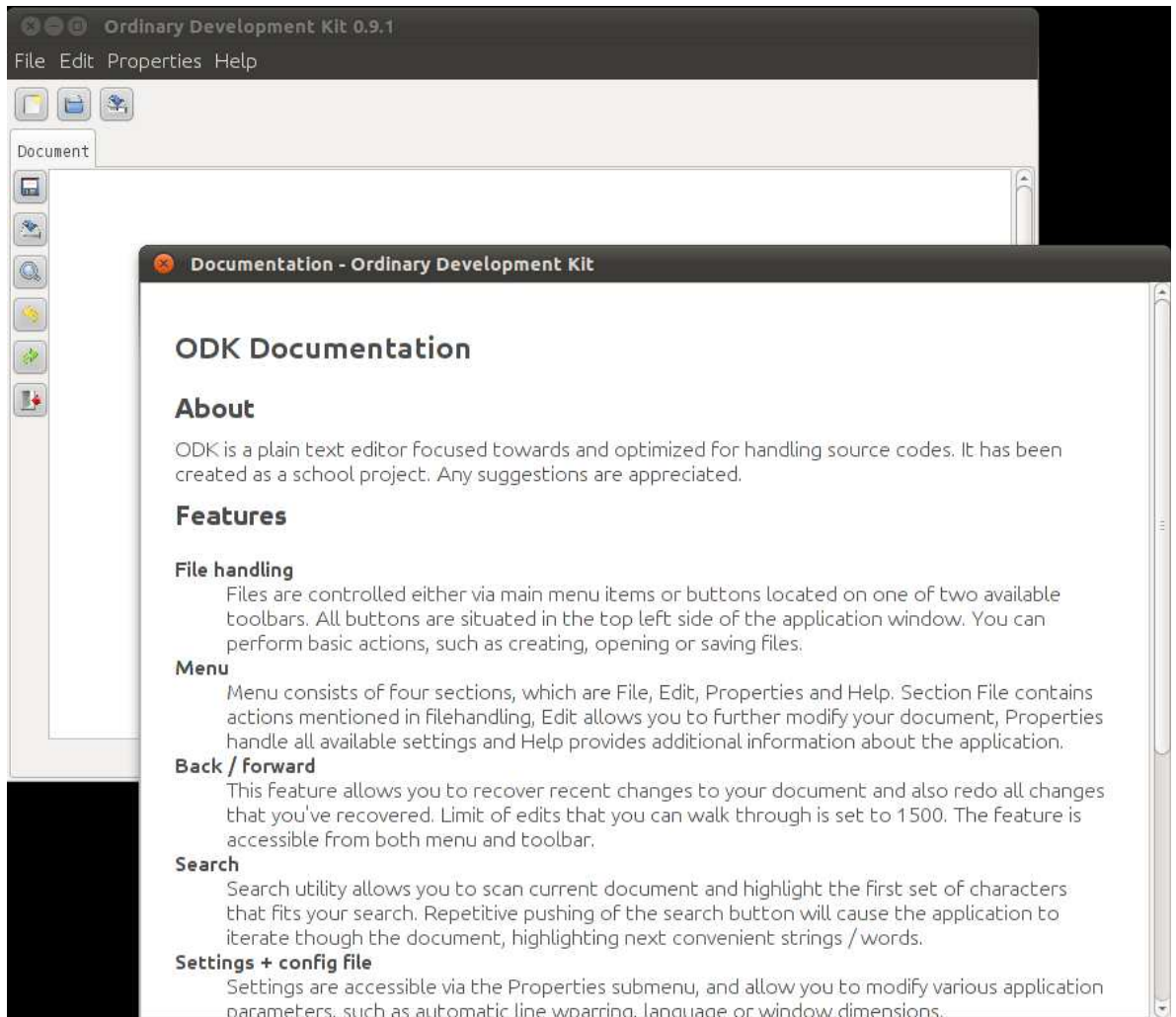
[15] Officiální dokumentace Javy [cit. 2011-03-16]

URL: <<http://download.oracle.com/javase/1.5.0/docs/api/>>

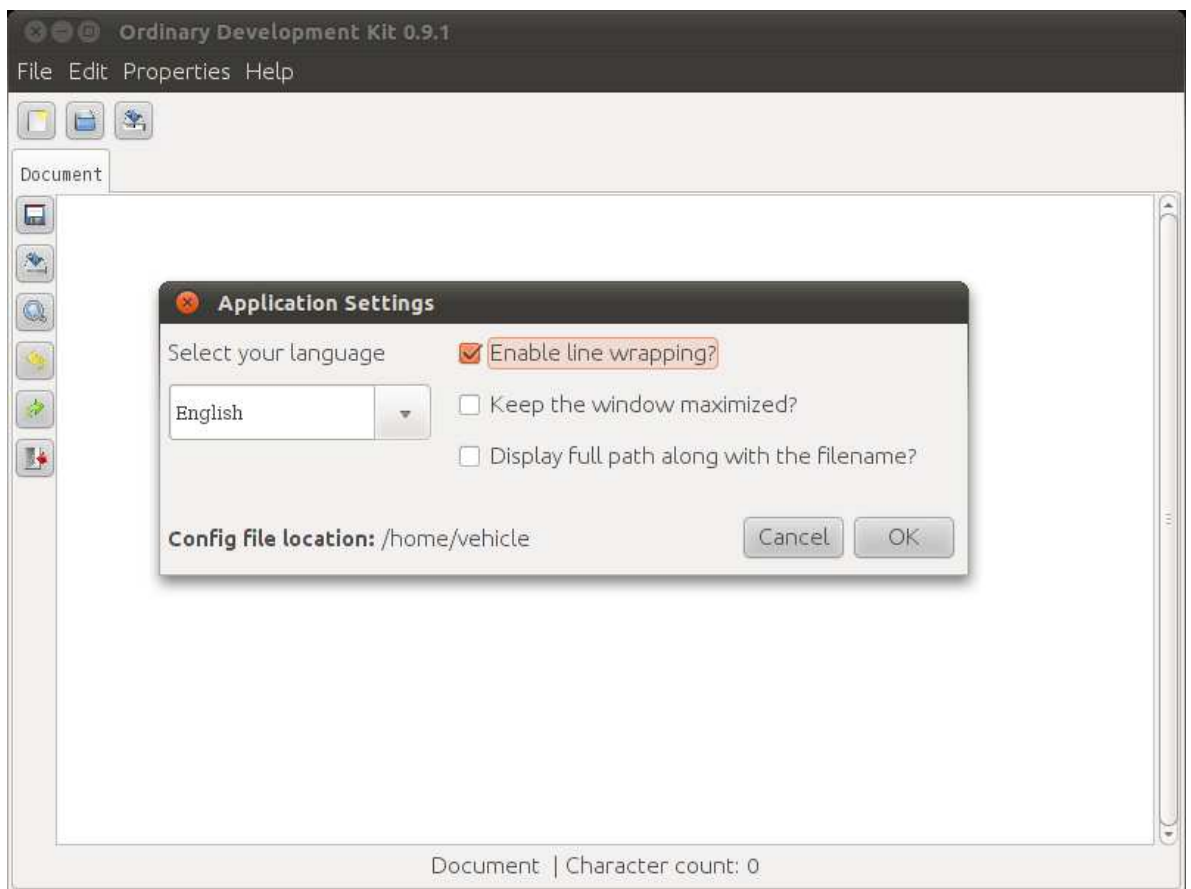
6 Přílohy



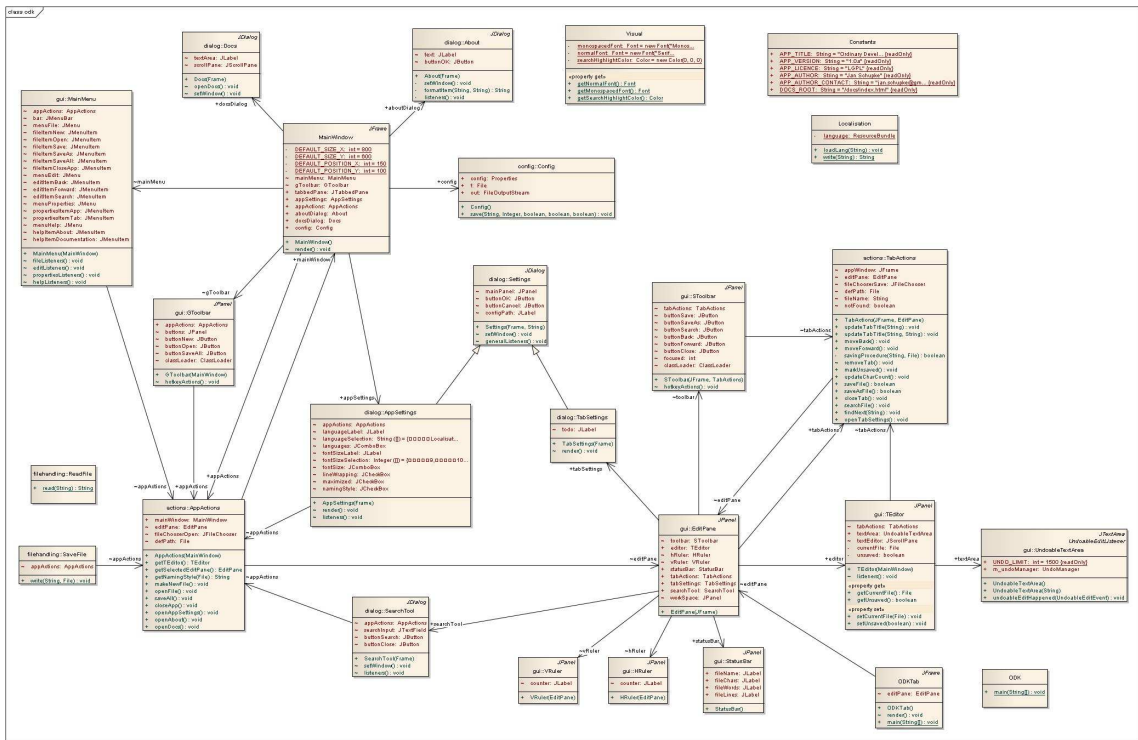
Příloha 1: Hlavní okno aplikace



Příloha 2: Dokumentace aplikace



Příloha 3: Dialog nastavení



Priloha 4: Class diagram (obrazek plne velikosti je prilozhen mimo dokument)