Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

XML editor

jeXMLe

2018

Albert Kaufman

Vedoucí práce: RNDr. Arnošt Večerka

Studijní obor: Informatika, prezenční forma

Bibliografické údaje

Autor:	Albert Kaufman
Název práce:	XML editor (jeXMLe)
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2018
Studijní obor:	Informatika, prezenční forma
Vedoucí práce:	RNDr. Arnošt Večerka
Počet stran:	32
Přílohy:	1 CD/DVD
Jazyk práce:	český

Bibliograhic info

Author:	Albert Kaufman
Title:	XML editor (jeXMLe)
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2018
Study field:	Computer Science, full-time form
Supervisor:	RNDr. Arnošt Večerka
Page count:	32
Supplements:	1 CD/DVD
Thesis language:	Czech

Anotace

XML je jeden z nejrozšířenějších značkovacích jazyků, který se používá pro serializaci dat. Editor, který je součástí této bakalářské práce, zvládne jak tvořit, tak upravovat dokumenty ve formátu XML. Uživatel může dokument validovat proti DTD nebo XML Schema, zobrazit stromovou strukturu, jednoduše vkládat značky a upravovat dokument obvyklými editačními funkcemi. Důraz byl kladen na celkovou jednoduchost aplikace.

Synopsis

XML is one of the most widely used markup languages used to serialize data. The editor, part of this bachelor thesis, can create and edit XML documents. The user can validate the document against the DTD or XML Schema, view the tree structure, simply insert the tags, and edit the document with the usual editing functions. Emphasis was put on the overall simplicity of the application.

Klíčová slova: editor; XML; značka; strom; validace dokumentu; C#

Keywords: editor; XML; tag; tree; document validation; C#

Děkuji vedoucímu práce za cenné rady, které mi pomohly při vývoji aplikace a rodině za veškerou podporu, kterou mi poskytovala během studia.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce podpis autora

Obsah

1 Úvod **8**
- 1.1 Cíle a členění . . . 8
- 1.2 Název a logo . . . 8
- 1.3 Existující programy . . . 8
  - 1.3.1 XML Copy Editor . . . 9
  - 1.3.2 XML Notepad . . . 9
  - 1.3.3 EditiX XML Editor . . . 10
  - 1.3.4 Oxygen XML Editor . . . 10

2 Úvod do XML **12**
- 2.1 Co je XML . . . 12
- 2.2 Parsery . . . 12
  - 2.2.1 DOM . . . 12
  - 2.2.2 SAX . . . 12
- 2.3 Syntaxe . . . 13
  - 2.3.1 Základy . . . 13
  - 2.3.2 Atributy . . . 13
  - 2.3.3 Entity . . . 14
  - 2.3.4 Komentáře . . . 14
  - 2.3.5 CData . . . 14
  - 2.3.6 Jmenné prostory . . . 14
  - 2.3.7 Správná strukturovanost dokumentu . . . 15
- 2.4 Validace dokumentu . . . 16

3 Použité technologie **17**
- 3.1 Visual Studio . . . 17
- 3.2 .NET . . . 17
- 3.3 C# . . . 17
- 3.4 Windows Form . . . 17

4 Programátorská příručka **18**
- 4.1 Popis tříd . . . 18
  - 4.1.1 Třída hlavni . . . 18
  - 4.1.2 Třída najit . . . 18
  - 4.1.3 Třída nahradit . . . 18
  - 4.1.4 Třída vlozitZnacku . . . 18
  - 4.1.5 Třída stromXML . . . 18
  - 4.1.6 Třída validace . . . 18
  - 4.1.7 Třída nastaveni . . . 19
  - 4.1.8 Třída fonty . . . 19
  - 4.1.9 Třída oProgramu . . . 19
  - 4.1.10 Třída prirucka . . . 19
  - 4.1.11 Třída Program . . . 19

4.1.12 Třída Settings . . . 19
4.2 Omezení aplikace . . . 19
4.3 Možná rozšíření . . . 20

5 Uživatelská příručka **21**
5.1 Požadavky . . . 21
5.2 Instalace . . . 21
5.3 Odinstalace . . . 21
5.4 Hlavní okno . . . 21
5.5 Soubor . . . 22
5.6 Úpravy . . . 23
5.6.1 Najít . . . 23
5.6.2 Nahradit . . . 24
5.7 XML . . . 24
5.7.1 Vložit značku . . . 25
5.7.2 Zobrazit strom . . . 25
5.7.3 Validace pomocí DTD nebo XML Schématu . . . 26
5.8 Možnosti . . . 27
5.8.1 Nastavení . . . 27
5.9 Nápověda . . . 28

Závěr **29**

Conclusions **30**

A Obsah přiloženého CD/DVD **31**

Literatura **32**

Seznam obrázků

1 Logo . . . 8
2 XML Copy Editor . . . 9
3 XML Notepad . . . 10
4 EditiX XML Editor . . . 11
5 Oxygen XML Editor . . . 11
6 Složka s editorem nacházející se na přiloženém DVD . . . 21
7 Hlavní okno . . . 22
8 Najít . . . 24
9 Nahradit . . . 24
10 Vložení značky . . . 25
11 Strom . . . 26
12 Validace . . . 26
13 Nastavení . . . 27

Seznam zdrojových kódů

1 Příklad jednoduchého XML dokumentu . . . 13
2 Příklad elementu s atributy . . . 13
3 Příklad komentáře . . . 14
4 Příklad CData . . . 14
5 Příklad jmenných prostorů . . . 15
6 Příklad správně strukturovaného XML . . . 15
7 Příklad DTD . . . 16
8 Příklad XML Schema . . . 16

1 Úvod

1.1 Cíle a členění

Cílem práce bylo vytvořit aplikaci, která by umožňovala vytvářet a upravovat dokumenty ve formátu XML. Aplikace by měla obsahovat základní editační nástroje jako je například kopírování, vložení, hledání a tak dále. Dále by měla umožňovat zkontrolovat strukturu dokumentu, zda odpovídá XML nebo validovat dokument pomocí DTD nebo XML Schema. Další z cílů bylo zobrazení stromové struktury dokumentu. Posledním hlavním cílem bylo jednoduché vkládání značek. Z věcí navíc bych zmínil zvýrazňování syntaxe, kterou by měl mít každý lepší editor. Mým osobním cílem bylo vytvořit velmi přehledný a jednoduchý XML editor se základními funkcemi.

Práce je rozdělena do pěti kapitol. V úvodní části jsou popsány cíle práce, název aplikace, logo a již existující podobné programy. Ve druhé kapitole se věnuji základům jazyka XML. Ve třetí kapitole objasňuji použité technologie. Ve čtvrté kapitole je programátorská příručka, kde popisuji třídy, omezení aplikace a možné rozšíření. V poslední kapitole s názvem uživatelská příručka je do detailu popsáno uživatelské rozhraní programu a funkce, které uživatel může použít. Zároveň se zde vyskytují pokyny pro instalaci a odinstalaci programu.

1.2 Název a logo

Editor jsem nazval jeXMLe, což znamená *jednoduchý XML editor*. Stejně jak zní název, jsem chtěl i podobně jednoduché logo (Obrázek 1), které je vytvořeno v programu pro vektorovou grafiku s názvem *Zoner Callisto 5*.

Obrázek 1: Logo

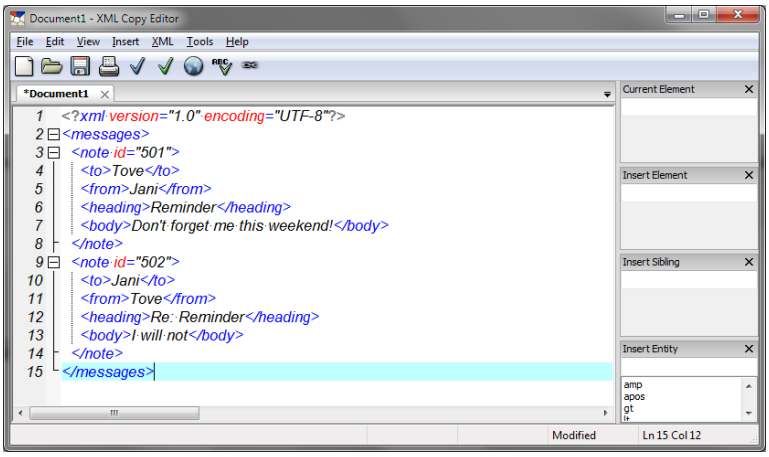
1.3 Existující programy

Editorů na zpracovávání dokumentů ve formátu XML existuje celá řada. Je proto těžké vybrat nějaký reprezentativní vzorek na ukázku, protože upravovat a tvořit XML soubory může zpravidla jakýkoliv textový editor, jelikož je XML prostý text. Jeden příklad za všechny je *Poznámkový blok* od firmy Microsoft. Ten ale neumožňuje například validovat dokumenty nebo jednoduše vkládat značky, což dovedou právě specializované editory na XML. Z toho důvodu se v této podkapitole zaměřím na editory, které jsou na úpravu XML vytvořeny. Konkrétně na dva programy zadarmo a na dva komerční.

1.3.1 XML Copy Editor

Xml Copy Editor [1] vytvořil v roce 2005 Gerald Schmidt. Je zadarmo pro operační systém Windows a Linux. Poslední aktualizace proběhla v roce 2014, takže se domnívám, že se dále nevyvíjí.

Tento editor není tak rozsáhlý jako jiné popisované editory, ale na druhou stranu je velice jednoduchý a přehledný. Editor umí validovat XML dokumenty pomocí DTD, XML Schema a Relax NG Schema. Podporuje zvýraznění syntaxe, XPath, automatické dokončování značek a zamykání značek. Pro přehlednost je rozhodně výhodou tvz. code folding, což znamená zabalování a rozbalování kódu. Kontroluje pravopis a styl. Má vestavěnou podporu XHTML, XSL, DocBook a TEI. Jako bonus umožňuje bezztrátový import a export dokumentů Microsoft Word (pouze pro Windows).

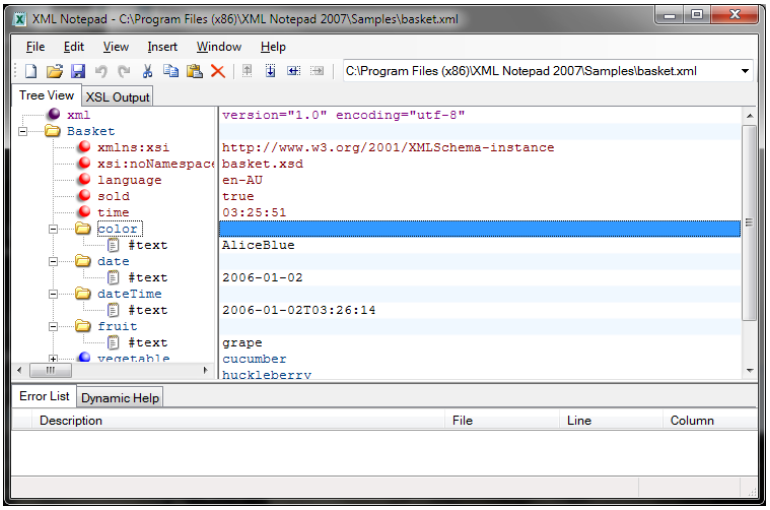


Obrázek 2: XML Copy Editor

1.3.2 XML Notepad

XML Notepad [2] byl původně vytvořen Murray Lowem roku 1998 v programovacím jazyce C++. Později se vývoj přesunul pod křídla Microsoftu kde změnili programovací jazyk na C#. Je zadarmo na operační systém Windows. Má velice přívětivé uživatelské rozhraní pro začínající uživatele.

Editor umožňuje pouze stromový náhled, což je docela velká nevýhoda. Ve stromě je možné myší přesouvat uzly a umisťovat je jinde po stromě. Má dobrý výkon při otevírání větších souborů. Například tří megabajtový soubor se otevírá jednu sekundu. Velkou výhodou je i to, že editor umožňuje validaci dokumentu pomocí XML Schema v reálném čase. Dále je zde nekonečné Zpět/Vpřed. Dobrou vlastností je také postupné vyhledávání jak v textovém, tak ve stromovém náhledu s podporou Regex a XPath výrazů. Nachází se zde i zvýraznění syntaxe jednotlivých uzlů stromu s možností změny barev i typu písma.



Obrázek 3: XML Notepad

1.3.3 EditiX XML Editor

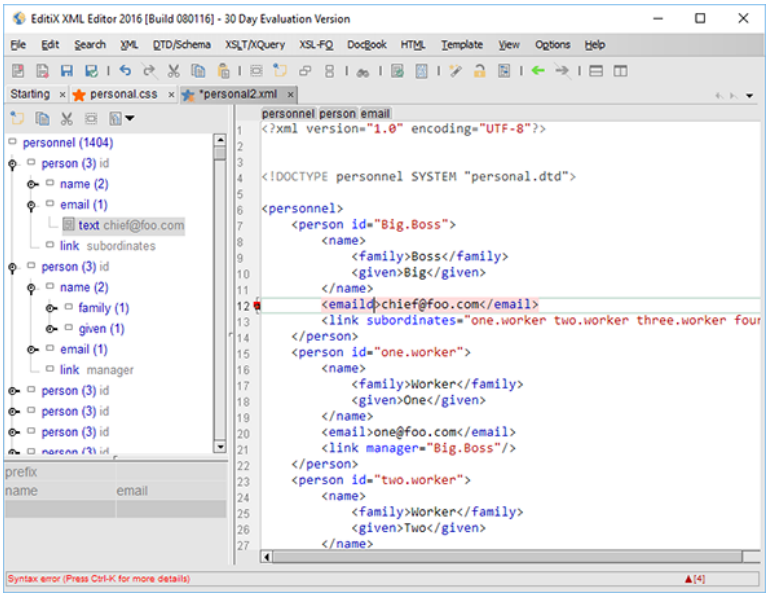
EditiX XML Editor [3] vyvinula v roce 2004 firma JAPISoft SARL. Funguje na operační systému Windows, Mac OS X a Linux. Program je placený a dodnes vyvíjený. Na zkoušku si lze stáhnout 30 denní zkušební verzi. Je samozřejmě možné zakoupit placenou verzi, jejíž cena začíná na 119 dolarech.

Editor má synchronizovaný stromový náhled s textovým. Dále obsahuje pomocníky pro výběr elementů, atributů, hodnot atributů a entit. Z pomocníků se tu ještě nachází pomocník pro uzavírání elementů a asistent pracující s DTD, W3C Schema a Relax NG Schema. Detekuje syntaktické chyby v reálném čase v obou náhledech a s radou, jak chybu opravit. Užitečný je taktéž správce jmenných prostorů. Uživatel může přidávat záložky označeným elementům. Editor podporuje převod DTD, XML Schema a Relax NG Schema mezi sebou. Může zjišťovat a opravovat chyby v externích dokumentech DTD nebo W3C Schema.

1.3.4 Oxygen XML Editor

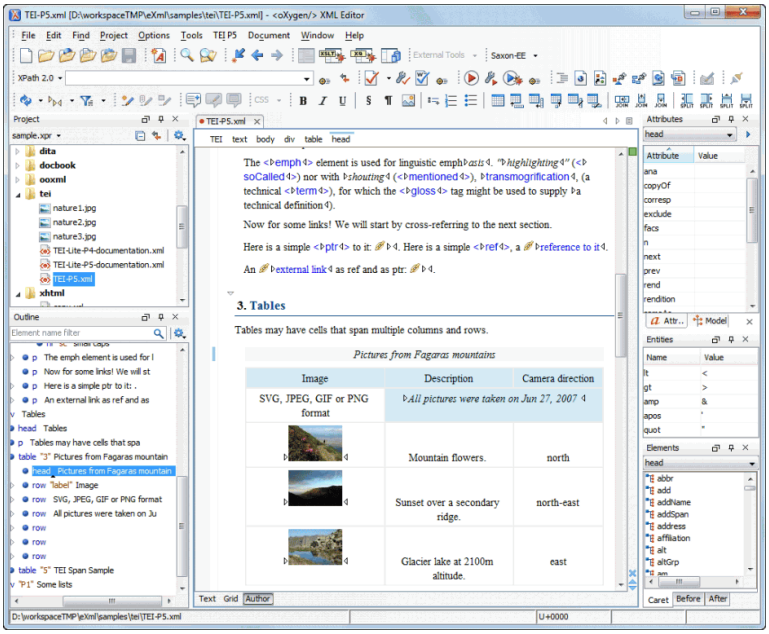
Editor označovaný také jako *<oXygen/>* [4] byl vyvinut na platformě Java vývojáři SyncRO Soft Ltd. Romania roku 2002. Je funkční na operačním systému Windows, Mac OS X a Linux. Je placený a jeho cena začíná na 99 dolarech. Existuje i 30 denní zkušební verze. Oxygen je jeden z nejrozsáhlejších a nejlepších editorů XML na trhu.

Oxygen umožňuje zobrazit XML jako text, strom nebo WYSIWYG, což je text s formátováním, který ukazuje jak výsledné XML bude vypadat vizuálně. Obsahuje našeptávač názvů značek a atributů na základě již napsaných značek nebo značek, které jsou v přidruženém DTD nebo XML Schématu. Dále je zde zabalování/rozbalování kódu, zvýrazňování syntaxe, zamykání značek nebo rychlé opravy překlepů ve značkách. Zásadní funkcionalitou je kolaborace, kdy může upravovat jeden dokument více lidí přes internet a psát si k jednotlivým



Obrázek 4: EditiX XML Editor

částem textu komentáře. Zajímavou funkcí je i validace dokumentu proti mnoha typům schémat v jednom kroku.



Obrázek 5: Oxygen XML Editor

2 Úvod do XML

2.1 Co je XML

XML (eXtentible Markup Language) je značkovací jazyk, jehož kořeny sahají do roku 1986, kdy vnikl jazyk SGML. Z jeho základů vznikl v roce 1998 značkovací jazyk XML. XML je brán za nástupce HTML, který v roce 1998 už nepostačoval. XML je univerzální a otevřený formát. Oproti HTML lze v XML definovat vlastní značky. XML specifikuje, co se nachází v dokumentu. Neříká nic o tom, jak má vypadat. Informace o formátováno se nachází ve stylovém jazyku, například CSS [8]. XML popisuje data, takže je jednoduché ho převést do jiného formátu. Význam má XML i pro databáze, kdy není nutné pracovat přímo s databází [7]. XML se používá hlavně pro uchovávání a přenos dat. Byl vyvinut tak, aby byl čitelný jak pro člověka, tak pro počítač. Hraje důležitou roli v mnoha informačních systémech. Často se používá pro distribuci dat přes internet [6].

2.2 Parsery

Parser [5] je program, který kontroluje syntaktickou správnost dokumentu. Lepší parsery kontrolují také dokument proti nějakému schématu. Je zabudován ve všech prohlížečích, které podporují XML. Mezi nejznámější patří parser od Microsoftu. Z dalších parserů existují například SP a XP. Pro programátora je čtení i zápis XML snadný, jelikož může použít parsery v podobě knihoven a použít je ve svých programech. Program pak nemusí kontrolovat syntaxi dokumentu, protože ji kontroluje parser. Využívají se dvě rozhraní, které nám zpřístupní obsah dokumentu v příjemné formě.

2.2.1 DOM

Rozhraní DOM [5] (Document Object Model) je založeno na stromové hierarchická struktuře, kdy každému elementu odpovídá jeden uzel stromu. DOM obsahuje funkce, které nám umožňují celý strom procházet, modifikovat, mazat a přidávat jednotlivé uzly. Dokument nemusíme procházet od začátku do konce jako je tomu u SAX. DOM se používá pro náročnější operace s dokumentem.

2.2.2 SAX

Rozhraní SAX [5] (Simple API for XML) je založeno na řízení pomocí událostí. Pomocí rozhraní se vytvoří vazba mezi událostmi a naším kódem. Událostmi řízený přístup je rychlý a má malou spotřebu paměti. Jednotlivé události se vyvolávají postupně, jak je členěn dokument.

2.3 Syntaxe

2.3.1 Základy

XML se skládá z elementů, které začínají a končí takzvanou značkou. Říká se tomu počáteční a ukončovací značka. Mezi těmito značkami se nachází obsah elementu, což může být jakýkoliv text nebo další element. Elementy jsou tedy do sebe vnořeny a tvoří hierarchickou strukturu. Proto se elementy nemůžou křížit. Pokud chceme udělat element bez obsahu, stačí napsat *<elementBezObsahu/>*. Každý dokument musí mít kořenový element, který je rodičem všech ostatních, což znamená, že musí být celý dokument zabalený do jednoho elementu [5]. Na začátku každého dokumentu by se měla psát XML deklarace s verzí a kódováním XML. Tento krok je sice nepovinný, ale v některých jazycích by bez deklarace mohli nastat problémy se zobrazením. Pokud se XML deklarace nenapíše, tak je výchozí kódování UTF-8. Počáteční značka se skládá z názvu značky a neomezeného počtu atributů s hodnotou. Název značky má určitá omezení:

- není závislý na velikosti písma
- první znak musí být písmeno nebo podtržítko
- první tři znaky nesmí obsahovat slovo „xml“ v jakékoliv velikosti písmen
- může obsahovat písmena, číslice, pomlčky, podtržítka a tečky
- nesmí obsahovat mezery [6]

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <koren>
3    <prazdnyElement/>
4  </koren>
```

Zdrojový kód 1: Příklad jednoduchého XML dokumentu

2.3.2 Atributy

Atributy [6] se používají pro rozšíření informace. Nemohou obsahovat více hodnot ani tvořit stromovou strukturu. Názvy atributů mají podobné omezení jako názvy značek s tím rozdílem, že je možné jako první tři znaky napsat „xml“ v jakékoliv velikosti písmen. Hodnoty atributů jsou vždy v uvozovkách nebo mezi dvěma apostrofy. Není možné je křížit. Na druhou stranu je možno při uzávorkované hodnotě použít uvnitř apostrofy nebo entitu tohoto symbolu a naopak. Pokud máme více atributů v jedné značce, tak musí mít rozdílné názvy a minimálně musí být mezi nimi jedna mezera.

```
1  <znacka atribut1="10" atribut2="20">obsah</znacka>
```

Zdrojový kód 2: Příklad elementu s atributy

2.3.3 Entity

V XML existuje 5 předdefinovaných entit zastupující určité řídící znaky, které nemůžeme v textu normálně použít.

- < znamená <
- > znamená >
- & znamená &
- ' znamená ’
- " znamená "[6]

2.3.4 Komentáře

Pro přehlednost můžeme v dokumentu využívat komentáře [5], do kterých je možno napsat cokoliv, kromě dvou pomlček. Komentáře se mohou vyskytovat kdekoliv, kromě vnitřku značky. Komentáře parser přeskakuje.

```
<!-- komentar -->
```

Zdrojový kód 3: Příklad komentáře

2.3.5 CData

CData [5] se využívají, pokud chceme do dokumentu vložit větší množství textu, který obsahuje řídící znaky, a bylo by tak nepohodlné psát znakové entity. Může se například jednat o zdrojový kód. Text v CData může obsahovat cokoliv, kromě sekvence znaků *]]>*.

```
<![CDATA[
  int a = 5;
  printf("%d", a)
]]>
```

Zdrojový kód 4: Příklad CData

2.3.6 Jmenné prostory

Jmenné prostory [5] umožňují v jednom dokumentu kombinovat více sad značek (elementy, atributy). Každá sada značek je definována URI adresou, což je identifikátor informačního zdroje. Jmenné prostory se definují nejčastěji v kořenovém elementu jako atributy následovně: *xmlns:«prefix»="«URI sady značek»"*. Pak stačí dát před název elementu nebo atributu konkrétní prefix jmenného prostoru.

```
<koren xmlns:a="http://www.w3.org/TR/html4/" xmlns:b="
   https://www.w3schools.com/furniture">
  <a:produkt>
    <a:zemePuvodu>Nemecko</h:zemePuvodu>
    <a:barva>Zelena</h:barva>
  </a:produkt>
  <b:produkt>
    <b:nazev>Salam</b:nazev>
    <b:cena>10 Kc</b:cena>
    <b:online>ne</b:online>
  </b:produkt>
</koren>
```

Zdrojový kód 5: Příklad jmenných prostorů

2.3.7 Správná strukturovanost dokumentu

Pokud má XML dokument správnou syntaxi, tak je správně strukturovaný. To kontroluje parser. Zde jsou syntaktická pravidla:

- XML dokument musí mít kořenový element
- elementy musí mít uzavírací značku
- název počáteční a ukončovací značky elementu je závislý na velikosti písma
- elementy musí být řádně vnořené
- hodnoty atributů musí být v uvozovkách nebo apostrofech [6]

```
<?xml version="1.0" encoding="UTF-8"?>
<herniObchod>
  <hra platforma="pc">
    <nazev lokalizace="cz">Mafia</nazev>
    <vyvojar>Illusion Softworks</vyvojar>
    <rok>2002</rok>
  </hra>
  <hra platforma="ps4">
    <nazev lokalizace="en">God of War</nazev>
    <vyvojar>Santa Monica Studio</vyvojar>
    <rok>2018</rok>
  </hra>
</herniObchod>
```

Zdrojový kód 6: Příklad správně strukturovaného XML

2.4 Validace dokumentu

Pokud je dokument validní, znamená to, že je správně strukturovaný a musí odpovídat definici typu dokumentu. Existují dvě definice typu dokumentu:

- DTD - Původní definice typu dokumentu.
- XML Schema - Alternativa založena na XML.

Definice typu dokumentu definuje pravidla s povolenými elementy a atributy, které lze použít v XML dokumentu [6].

```
<!DOCTYPE note
[
<!ELEMENT prvni (druhy,treti,ctvrty,paty)>
<!ELEMENT druhy (#PCDATA)>
<!ELEMENT treti (#PCDATA)>
<!ELEMENT ctvrty (#PCDATA)>
<!ELEMENT paty (#PCDATA)>
]>
```

Zdrojový kód 7: Příklad DTD

```
<xs:element name="prvni">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="druhy" type="xs:string"/>
      <xs:element name="treti" type="xs:string"/>
      <xs:element name="ctvrty" type="xs:string"/>
      <xs:element name="paty" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Zdrojový kód 8: Příklad XML Schema

3 Použité technologie

V této kapitole popisuji použité technologie, které jsem potřeboval při vývoji editoru. Ten jsem vyvíjel v programovacím jazyce C# s knihovna tříd pro tvorbu grafického rozhraní Windows Form. Kód jsem psal ve Visual Studiu 2015 na platformě .NET Framework ve verzi 4.5.2.

3.1 Visual Studio

Visual Studio [9] je vývojové prostředí od Microsoftu. Používá se pro vývoj konzolových aplikací a aplikací s grafickým rozhraním na platformách společnosti Microsoft. Obsahuje editor kódu s podporou IntelliSense (inteligentní dokončování kódu) a refaktorování. Integrovaný debugger pracuje na úrovni kódu i stroje. Mezi vestavěné nástroje patří designer formulářů pro tvorbu GUI, designer webu, tříd a databázových schémat. Je možné přidávat rozšíření. Mezi vestavěné jazyky patří C, C++, C# a VB.NET. Ostatní jazyky lze přidávat pomocí jazykových služeb.

3.2 .NET

.NET [10] je soubor technologií v softwarových produktech, které tvoří celou platformu. Ta je dostupná pro Web, Windows a další. Common Language Infrastructure (CLI) je standardizovaná specifikace jádra .NET. Nejpoužívanější jazyky pro vývoj v .NET jsou C# a Visual Basic .NET.

3.3 C#

C# [11] je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft v roce 2000. Vznikl zároveň s platformou .NET Framework. Je schválen standardizačními komisemi ISO a ECMA. C# je kombinací jazyků C++ a Java, kde hlavně pro Javu je C# konkurencí. C# se používá při vývoji formulářových aplikací pro Windows, databázových programů, softwaru pro mobilní zařízení a tak dále.

3.4 Windows Form

Windows Form [12] je knihovna tříd pro tvorbu grafického rozhraní, která je součástí .NET Frameworku firmy Microsoft. Je nástupcem staršího MFC založeného na C++, ale neposkytuje porovnatelné paradigma a slouží pouze jako platforma pro vytváření uživatelského rozhraní při řešení vícevrstvé architektury. Poskytuje platformu pro vytváření aplikací pro desktop nebo tablety. Nástupcem je WPF (Windows Presentation Foundation).

4 Programátorská příručka

4.1 Popis tříd

Tato podkapitola se věnuje popisu všech tříd aplikace a zajímavých metod jím přidružených.

4.1.1 Třída hlavni

Třída **hlavni** představuje v uživatelském rozhraní hlavní okno. Instance okna je vytvořena ve vstupním bodě aplikace. Největší část okna tvoří komponenta *RichTextBox*, ve které se vyskytuje text. Tato třída obsahuje mnoho metod. Je zde například metoda *public void zvyrazneniSyntaxe()*, která zajišťuje zvýraznění syntaxe při událostech *TextChanged* a *VScroll* nebo metody pro hledání a nahrazování textu.

4.1.2 Třída najit

Tato třída představuje v uživatelském rozhraní okno, které slouží k vyhledávání textu v hlavním okně. Obsahuje metody, které volají metody hledání, jež se nachází ve třídě **hlavni**.

4.1.3 Třída nahradit

Třída **nahradit** představuje okno v uživatelském rozhraní, které je určeno k nahrazovaní textu v hlavním okně. Obsahuje metody, které volají metody nahrazování, jež se nachází ve třídě **hlavni**.

4.1.4 Třída vlozitZnacku

Třída **vlozitZnacku** představuje v uživatelském rozhraní okno, které je určené pro tvoření značek. Obsahuje metody, které ověřují správnost názvů a metodu vložení struktury značky do *RichTextBoxu*, který se nachází ve třídě **hlavni**.

4.1.5 Třída stromXML

Třída **stromXML** reprezentuje v uživatelském rozhraní dialog, v němž je komponenta *TreeView*, ve které se zobrazuje strom. Obsahuje jednu podstatnou metodu *private void vytvor()*, která se použije při inicializaci instance a vytvoří strom.

4.1.6 Třída validace

Tato třída představuje okno v uživatelské rozhraní, které je určené pro validaci dokumentu pomocí DTD nebo XML Schema. Třída obsahuje tyto 2 důležité metody:

- *private void xmlSchema()* - Provádí validaci pomocí XML Schema.
- *private void dtd()* - Provádí validaci pomocí DTD.

4.1.7 Třída nastaveni

nastaveni je třída, jež představuje v uživatelském rozhraní okno, ve kterém se nastavují různé věci. Obsahuje například metodu *public void barvatlacitka(Button a)* pro nastavení barvy tlačítka. Aby se nastavení projevilo v hlavním okně, tak se volá metoda s názvem *zmenitNastavení*, která se nachází ve třídě **hlavni**.

4.1.8 Třída fonty

Tato třída reprezentuje dialog v uživatelském rozhraní, ve kterém se vybírá typ a velikost písma. Při ukládání volá metodu *prepsatfont*, která se nachází ve třídě **nastaveni**.

4.1.9 Třída oProgramu

oProgramu je další třídou představující okno v uživatelském rozhraní. V tomto okně se nachází informace o programu. Žádná zajímavá metoda se zde nenachází.

4.1.10 Třída prirucka

Třída **prirucka** představuje další okno v uživatelském rozhraní. Přes celé okno je komponenta *RichTextBox*, ve kterém se nachází upravená verze uživatelské příručky této bakalářské práce.

4.1.11 Třída Program

Základní třída **Program** obsahuje metodu *Main*, což je hlavní vstupní bod aplikace. V této metodě je vytvořena instance třídy **hlavni**, která reprezentuje hlavní okno aplikace.

4.1.12 Třída Settings

Tato třída obsahuje metody, které umožňují měnit nastavení vlastností aplikace. Tyto vlastnosti a metody jsou využity v aplikaci pro ukládání textů, barev, pozic a velikosti oken.

4.2 Omezení aplikace

Při vývoji jsem se setkal s pár problémy, které se mi nepodařilo vyřešit. Proto je zde ve stručnosti popíšu.

Prvním problémem bylo zvýrazňování syntaxe. Jde o to, že u zvýrazňování syntaxe používám metodu *GetScrollPos*, která určuje a nastavuje pozice posuvníků. *GetScrollPos* je typu *unsigned short*, a tak může pojmout maximálně čísla

od 0 do 65535. Při větším čísle dojde k odchylce a posuvník se přesune na špatnou pozici. Musel jsem proto zvýrazňování syntaxe omezit na maximálně 65535 řádků. Při větším počtu se zvýrazňování syntaxe automaticky vypne.

Druhým problémem je výběr písma. Dostupných je 18 písem podle mého výběru, protože při všech písmech v dialogu *FontDialog* se objevovala chyba, která nešla odchytit pomocí výjimek. Chyba nastala, pokud jsem vybral písmo, které není typu TrueType. Proto jsem vytvořil vlastní dialog s písmy, které jsou TrueType a zároveň by je měla mít většina uživatelů.

Posledním neduhem je validace XML Schema přes externí dokument XSD. Při validaci se totiž objeví chyba, že v XSD není definován kořenový element. Tato chyba je způsobena tím, že externí schéma není definované jako atribut v kořenové elementu, který pak bere jako chybu. Validace tedy funguje, jen je třeba tuto chybu při externím schématu ignorovat.

4.3 Možná rozšíření

Do budoucna by se dala aplikace samozřejmě obohatit. Například by bylo možno přidat funkcionalitu pro lokalizaci, protože v současné době je jeXMLe pouze v češtině. Další nedostatkem je omezenost pouze na operační systém Windows. Určitě by bylo užitečné aplikaci rozšířit na macOS nebo Linux. Poslední zajímavou funkcionalitou by mohlo být obohacení stromového náhledu o přidávání, mazání a editaci uzlů.

5 Uživatelská příručka

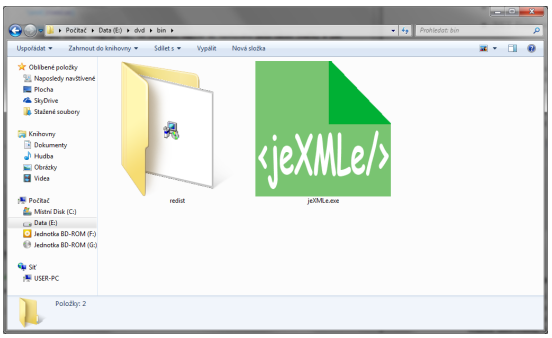
5.1 Požadavky

Požadavky na spuštění aplikace jsou dva:

- Windows Vista nebo novější
- .NET Framework 4.5.2 nebo novější

5.2 Instalace

Jelikož aplikace nemá instalátor, tak je potřeba soubor *jeXMLe.exe* ve složky */bin* na přiloženém DVD spustit přímo nebo zkopírovat kdekoliv na disk. Pokud aplikace nepůjde spustit, tak nesplňujete minimální požadavky. To znamená, že si buď musíte koupit novější verzi Windows nebo nainstalovat .NET Framework 4.5.2, který se nachází ve složce */bin/redist* na přiloženém DVD.



Obrázek 6: Složka s editorem nacházející se na přiloženém DVD

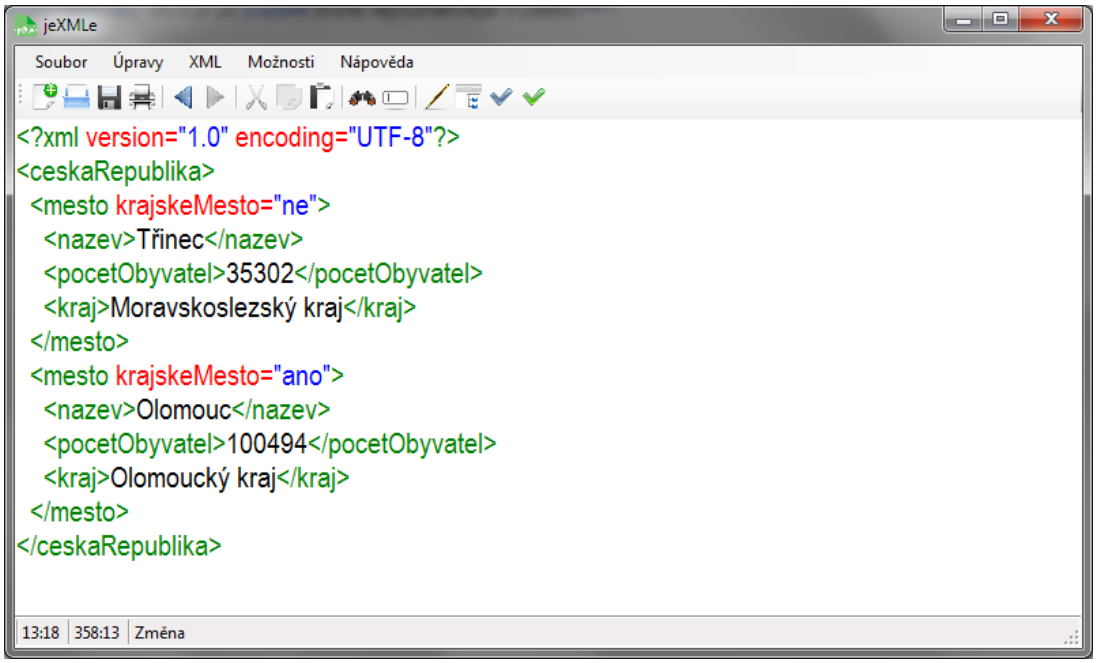
5.3 Odinstalace

Pokud máte aplikaci uloženou na disku a nespouštíte ji přímo z přiloženého DVD, tak k odinstalaci stačí smazat soubor *jeXMLe.exe* z disku. Pokud budete chtít smazat všechny uložená uživatelská data, tak musíte smazat konfigurační soubor *user.config*, který se nachází:
%localappdata%\AlbKauSoft\jeXMLe.exe«URI»\1.0.0.0

5.4 Hlavní okno

Po spuštění aplikace se objeví hlavní okno, ve kterém se nachází textové pole s předem vyplněnou XML deklarací. V levém dolním rohu je celkový počet znaků a řádků v textovém poli. Vedle je pak ještě současná pozice kurzoru v textovém poli. Posledním ukazatelem je indikátor *Změna*, který se zobrazuje, pokud byl text od posledního uložení změněn. Nad textovým polem se vyskytují obrázková

tlačítka základních funkcí programu. Taktéž se některé základní editační funkce objeví při pravém kliku myší kdekoliv do textového pole. Ta nejdůležitější nabídka se nachází na samotném vrchu aplikace. O té si povíme v následujících podkapitolách.



Obrázek 7: Hlavní okno

5.5 Soubor

Tato sekce obsahuje základní práci se souborem.

- Nový - Vytvoří nový dokument s XML deklarací, která nastavuje kódování na UTF-8 a verzi XML na 1.0. Pokud je zobrazen indikátor *Změna* v levém dolním rohu, tak se objeví dialog, jestli chce uživatel uložit dokument vyskytující se v textovém poli.
- Otevřít... - Otevře dokument ve formátu XML.
- Uložit - Pokud text v textovém poli nemá dosud určené umístění, objeví se dialog s určením místa uložení. V opačném případě se dokument uloží do současného umístění bez otevření dialogu.
- Uložit jako... - Otevře se vždy dialog s dotazem o určení místa uložení dokumentu.
- Tisk... - Otevře se dialog s určením parametrů k tisku.
- Konec - Ukončí se program. Pokud je zobrazen indikátor *Změna*, objeví se dialog s dotazem o uložení dokumentu.

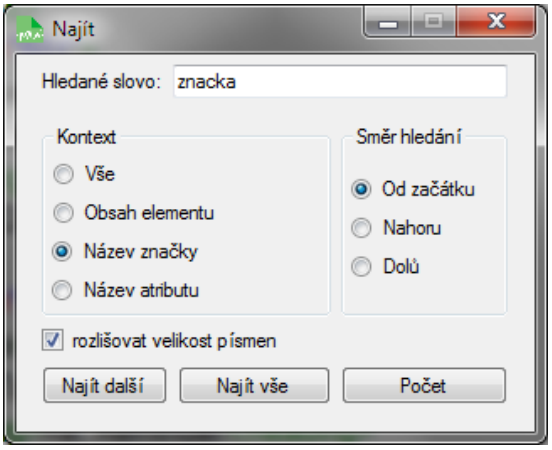
5.6 Úpravy

Tato sekce obsahuje úpravy textu.

- Zpět - Vrátí dokument o krok zpět před provedenou změnou. Limit je 50 kroků zpět.
- Vpřed - Skočí o krok dopředu, pokud byl předtím proveden příkaz Zpět. Limit je 50 kroků vpřed.
- Vyjmout - Uloží se text do schránky a zároveň se vymaže označený text.
- Kopírovat - Uloží se text do schránky.
- Vložit - Vloží obsah schránky na pozici určenou kurzorem v textovém poli.
- Smazat - Smaže označený text.
- Najít - Najde text v textovém poli.
- Nahradit - Nahradí text v textovém poli.

5.6.1 Najít

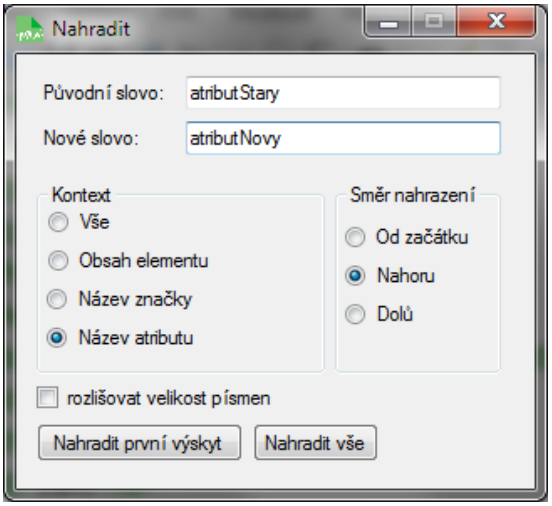
Uživatel může vyhledat slovo v textovém poli od začátku, kdy se vyhledává od počátku do konce. Pokud se slovo na konci nenajde, objeví se zpráva o neúspěšném hledání a pak se vyhledává znovu od začátku. Při změně ostatních parametrů se začne vyhledávat opět od začátku. Vyhledávání *Dolů* a *Nahoru* funguje tak, že se vyhledává od pozice kurzoru v určitém směru. Když se nic nenajde, vyhledávání dál nepokračuje. Dále může uživatel uvést, jestli chce vyhledávat s citlivostí na velikost písmen nebo brát ohled na kontext. Podle kontextu se může hledat buď všude, podle názvů značek, obsahu nejvnitřnějších elementů nebo názvů atributů. Pokud uživatel stiskne *Najít další*, tak se hledá po jednom. *Najít vše* najde všechny výskyty a vypíše je do seznamu s pozicí v textu. Po dvojkliku do seznamu se kurzor přesune a označí vyhledané slovo. Pokud se text změní, je nutno znova aktualizovat seznam stiskem *Najít vše*. Poslední tlačítko je *Počet*, kdy se zobrazí zpráva s počtem nalezených slov. U větších souborů se může stát, že tyto operace budou trvat delší dobu, a proto je nutné chvíli počkat.



Obrázek 8: Najít

5.6.2 Nahradit

Nahrazování funguje podobně jako hledání s tím rozdílem, že po vyhledání slova se slovo přepíše novým. Opět je možno nahrazovat podle kontextu, směru i s citlivostí na velikost písmen. Z tlačítek tu máme *Nahradit*, což nahradí první výskyt slova a *Nahradit vše*, což nahradí všechny výskyty.



Obrázek 9: Nahradit

5.7 XML

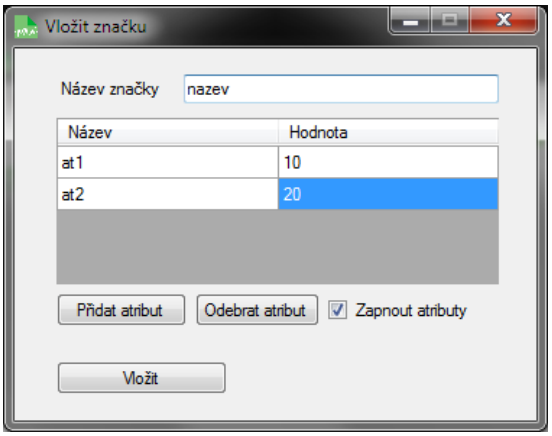
Tato sekce obsahuje specifické funkce pro XML.

- Vložit značku - Vloží na místo určené kurzorem strukturu značky, kterou uživatel zadá v dialogu.

- Zobrazit strom - Zobrazí stromovou strukturu XML dokumentu, který se nachází v textovém poli.
- Kontrola správné struktury - Zkontroluje se struktura XML dokumentu, který se nachází v textovém poli. Pokud je nalezena chyba, vypíše se chybová hláška a po stisknutí tlačítka *OK* skočí kurzor na pozici chyby. Ve druhém případě se objeví, že je dokument správně strukturovaný.
- Validace pomocí DTD nebo XML Schema - Ověří se, jestli dokument odpovídá DTD nebo XML Schema.

5.7.1 Vložit značku

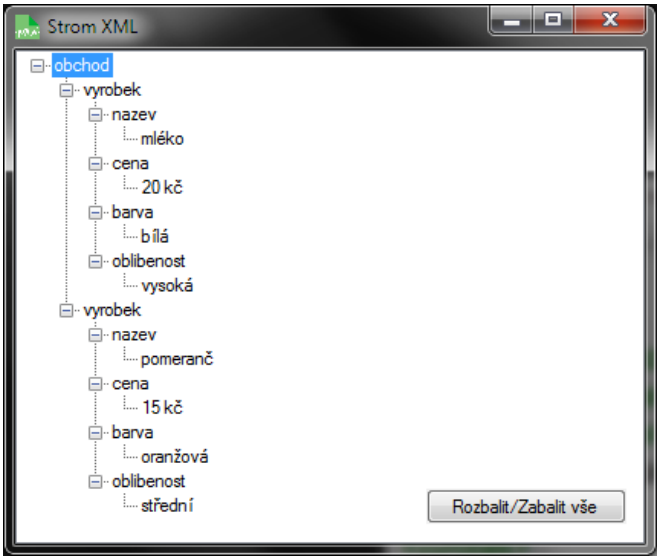
Přidávání značek funguje tak, že uživatel značku napíše do pole *Název značky* a pak může volitelně přidávat atributy pomocí tlačítka *Přidat atribut*. Pokud chce uživatel odstranit atributy, tak je stačí myší označit a kliknout na *Odebrat atribut*. Dále je možnost použít rozepsané atributy později nezaškrtnutím *Zapnout atributy*. Název a hodnota atributu se dá později upravit dvojklikem na konkrétní buňku. Po kliknutí na *Vložit* proběhne kontrola, jestli jsou názvy napsány podle konvencí a jestli se názvy atributů neopakují. Pokud je vše v pořádku, tak se značka vloží na místo určené kurzorem v textovém poli.



Obrázek 10: Vložení značky

5.7.2 Zobrazit strom

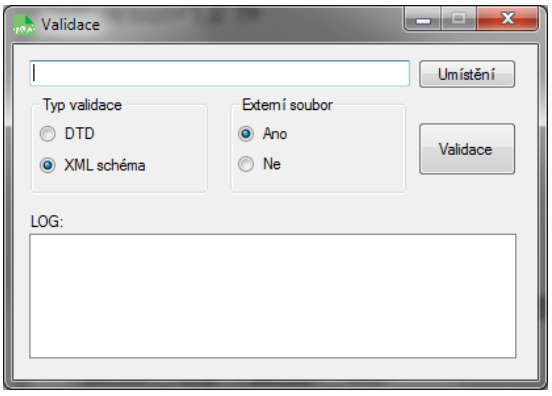
Zobrazí se strom s možností náhledu. Uživatel může strom postupně rozbalovat nebo zabalovat. Stiskem tlačítka *Rozbalit/Zabalit vše* se ve stromu rozbalí všechny uzly. Naopak se vše zabalí a zbude viditelný pouze kořenový element a věci, které se před kořenovým elementem mohou vyskytovat, například XML deklarace.



Obrázek 11: Strom

5.7.3 Validace pomocí DTD nebo XML Schématu

V tomto dialogu uživatel určí, jestli chce XML dokument validovat pomocí DTD nebo XML Schema. Dále je možnost si vybrat, jestli se struktura nachází přímo v dokumentu nebo externě v souboru s příponou .dtd nebo .xsd. U XML Schema, pokud zvolíme externí soubor, se objevuje chyba s chybným kořenovým elementem. Tato chyba se musí ignorovat, jelikož nejde odchytit. Po kliknutí na *Validovat* se provede validace. Pokud se narazí na chyby či upozornění, zobrazí se v LOGU. Po dvojkliku na chybu skočí kurzor na místo chyby v textovém poli. Automaticky se vypne zalomení řádků, aby byla správná lokalizace chybných řádků. Při úpravě dokumentu bude nutné provést validaci znovu, protože řádky chyb v LOGU nebudou odpovídat.



Obrázek 12: Validace

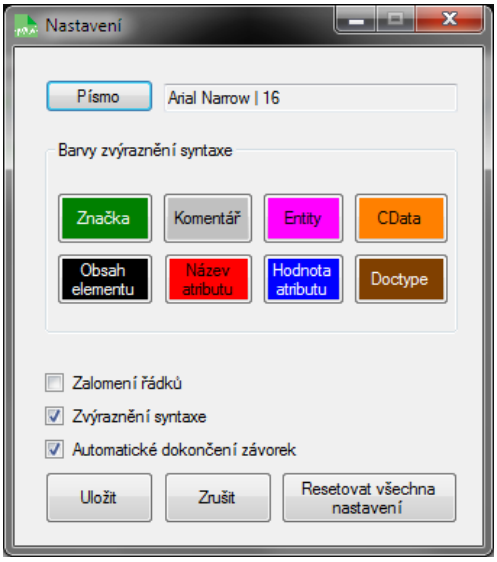
5.8 Možnosti

Tato sekce obsahuje různá nastavení.

- Zalomení řádků - Zalomí se řádky, pokud slovo přeteče za hranici aplikace.
- Zvýraznění syntaxe - Zapne se zvýraznění syntaxe XML dokumentu. Pokud má dokument více než 65535 řádků, tak se zvýraznění syntaxe vypne kvůli technickým limitům použité knihovny.
- Automatické dokončení závorky - Zapne se automatické dokončování závorky. Pokud uživatel vytvoří počáteční značku, vytvoří se automaticky i uzavírací značka. Dále pokud uživatel napíše = za názvem atributu, tak se vytvoří automaticky uvozovky. Poslední výhodou je automatické uzavření komentáře.
- Nastavení - Otevře se dialog s možnostmi.

5.8.1 Nastavení

Uživatel může nastavit barvy zvýraznění syntaxe. Další možností je změna typu písma. Je zde 18 písem dle mého výběru s možností změnit velikost. Omezený výběr písem je kvůli chybě, kterou popisuji v programátorské příručce. Dále se zde nachází tři výše popsaná tlačítka, jako je *Zalomeni řádků*, *Zvýraznění syntaxe* a *Automatické dokončení závorky*. Po stisknutí tlačítka *Uložit* se uloží aktuální nastavení. Dále je zde tlačítko *Resetovat všechna nastavení*, které obnoví všechna nastavení do původní podoby, včetně rozměrů a umístění všech oken aplikace. Tlačítkem *Zrušit* nebo *Křížek* se otevře dialog s dotazem, jestli opravdu chceme změněné nastavení zrušit a nechat původní.



Obrázek 13: Nastavení

5.9 Nápověda

Tato sekce obsahuje uživatelskou příručku a informace o programu.

- Uživatelská příručka - Upravená verze uživatelské příručky této bakalářské práce.
- O programu - Informace o programu a autorovi s odkazem na web programu.

Závěr

Cílem této bakalářské práce bylo sestavit XML editor, který bude schopen pracovat s dokumenty ve formátu XML. Editor měl mít funkce pro práci s textem, jednoduché vkládání značek, validaci dokumentu a zobrazení stromové struktury. Navíc jsem přidal zvýrazňování syntaxe, které ještě více podporuje přehlednost dokumentu. Kladl jsem důraz na celkovou jednoduchost aplikace.

Do budoucna by bylo možné tento program přeložit do více jazyků nebo ho rozšířit na více platforem, jelikož v této době je jeXMLe pouze na operační systém Windows. Pěkným rozšířením by mohla být editace stromu.

Při tvorbě aplikace jsem získal mnoho nových znalostí o jazyce XML a programování v jazyce C#. Výsledkem je tedy jednoduchá aplikace, která může být přínosná pro nenáročné uživatele.

Conclusions

The aim of this bachelor thesis was to create an XML editor, which will be able to work with documents in XML format. The editor had features for text editing, simple tag insertion, document validation, and tree view. In addition, I added a syntax highlighting that further enhances the clarity of the document. I put emphasis on the overall simplicity of the application.

In the future, this program could be translated into more languages or extended to multiple platforms, because at that time it is only for Windows operating system. A nice extension can be tree view editing.

When creating an application, I gained a lot of new knowledge about XML and C# programming. The result is a simple application that can be beneficial to undemanding users.

A Obsah přiloženého CD/DVD

bin/

Spustitelná verze aplikace jeXMLe a složka redist, ve které se nachází instalátor NETFramework452.exe s .NET Frameworkem ve verzi 4.5.2, který je nutný pro běh aplikace.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní zdrojové kódy programu jeXMLe se všemi potřebnými knihovnami a dalšími soubory potřebnými pro bezproblémové vytvoření spustitelných verzí programu.

readme.txt

Instrukce pro spuštění programu jeXMLe, včetně všech požadavků pro jeho bezproblémový provoz.

Literatura

[1] JI, Zane U.: *XML Copy Editor.* [online] 2018 [cit. 2018-07-27]. Dostupné z: http://xml-copy-editor.sourceforge.net/

[2] LOVETT, Chris: *XML Notepad 2007 Design.* [online] 2006 [cit. 2018-07-27]. Dostupné z: https://msdn.microsoft.com/en-us/library/aa905339.aspx

[3] JAPISoft SARL: *EditiX Open Source XML Editor.* [online] 2018 [cit. 2018-07-27]. Dostupné z: http://www.editix.com/index.html

[4] SyncRO Soft Ltd.: *Oxygen XML Editor.* [online] 2018 [cit. 2018-07-27]. Dostupné z: https://www.oxygenxml.com/xml_editor.html

[5] KOSEK, Jiří. *XML pro každého.* Grada Publishing 2000. 164 stran. [cit. 2018-07-27]. ISBN 80-7169-860-1.

[6] W3Schools: *XML tutorial.* [online] 2018 [cit. 2018-07-27]. Dostupné z: https://www.w3schools.com/xml/

[7] *XML základy.* [online] [cit. 2018-07-27]. Dostupné z: https://www.tvorba-webu.cz/xml/

[8] ŘEPIŠOVÁ, Zuzana: *XML - o čem a pro koho?* [online] 2000 [cit. 2018-07-27]. Dostupné z: http://webserver.ics.muni.cz/bulletin/articles/180.html

[9] Wikipedia: Otevřená Encyklopedie: *Microsoft Visual Studio.* [online] 2018 [cit. 2018-07-27]. Dostupné z: https://cs.wikipedia.org/wiki/Microsoft_Visual_Studio

[10] Wikipedia: Otevřená Encyklopedie: *.NET.* [online] 2018 [cit. 2018-07-27]. Dostupné z: https://cs.wikipedia.org/wiki/.NET

[11] Wikipedia: Otevřená Encyklopedie: *C Sharp.* [online] 2018 [cit. 2018-07-27]. Dostupné z: https://cs.wikipedia.org/wiki/C_Sharp

[12] Wikipedia: Otevřená Encyklopedie: *Windows Forms.* [online] 2017 [cit. 2018-07-27]. Dostupné z: https://cs.wikipedia.org/wiki/Windows_Forms