

# Nastavení programu

Lukáš Kovář (xkovl007)

19.4.2017

## **Abstrakt**

Tato práce pojednává o tvorbě konfigurace ve formátu XML, schématu XML, XML transformátoru do JSON, html dokumentace a  $\LaTeX$  dokumentace do předmětu “Značkovací jazyky”

## Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Metodika</b>	<b>3</b>
2.1	Použité nástroje . . . . .	3
<b>3</b>	<b>Popis elementů a atributů</b>	<b>4</b>
<b>4</b>	<b>Popis transformace</b>	<b>5</b>
<b>5</b>	<b>Popis schématu</b>	<b>5</b>
<b>6</b>	<b>Popis L<sup>A</sup>T<sub>E</sub>X</b>	<b>5</b>
<b>7</b>	<b>Popis html</b>	<b>5</b>
<b>8</b>	<b>Popis výstupu JSON</b>	<b>5</b>
<b>9</b>	<b>Závěr</b>	<b>5</b>

## Seznam obrázků

1	Snímek obrazovky doposud nepojmenovaného programu . . . . .	4
---	---	---

## Seznam tabulek

1	Soubory v repozitáři . . . . .	3
2	Využité programy . . . . .	3

# 1 Úvod

Tento dokument pojednává o semestrálním projektu pro předmět "Značkovací jazyky", jehož tématem bylo navrhnout nastavení/konfiguraci (smyslené) aplikace.

Semestrální projekt obsahuje následující soubory:

Název souboru	Popis souboru
<b>popis.html</b>	HTML soubor s popisem tohoto projektu
<b>nastaveni.xml</b>	XML soubor obsahující konfiguraci smyšleného počítačového programu
<b>nastaveni.xsd</b>	XML schéma výše uvedeného XML souboru
<b>xml_to_json.xsl</b>	XML transformační dokument pro převod výše uvedeného XML souboru do JSON souboru
<b>nastaveni.json</b>	JSON soubor vygenerovaný pomocí transformačního dokumentu XSL
<b>nepojmenovany.png</b>	Snímek obrazovky doposud nepojmenovaného programu
<b>popis.tex</b>	L <sup>A</sup> T <sub>E</sub> X soubor s popisem projektu
<b>popis.pdf</b>	PDF soubor s popisem tohoto projektu vygenerovaný pomocí pdflatex ze souboru „popis.tex“
<b>zadani.txt</b>	Zadání tohoto projektu zadané cvičícím/přednášejícím
<b>README.md</b>	Soubor obsahující jednoduchý popis repozitáře github
<b>LICENSE.md</b>	Soubor obsahující text permissivní licence MIT (pro github)

Tabulka 1: Soubory v repozitáři

## 2 Metodika

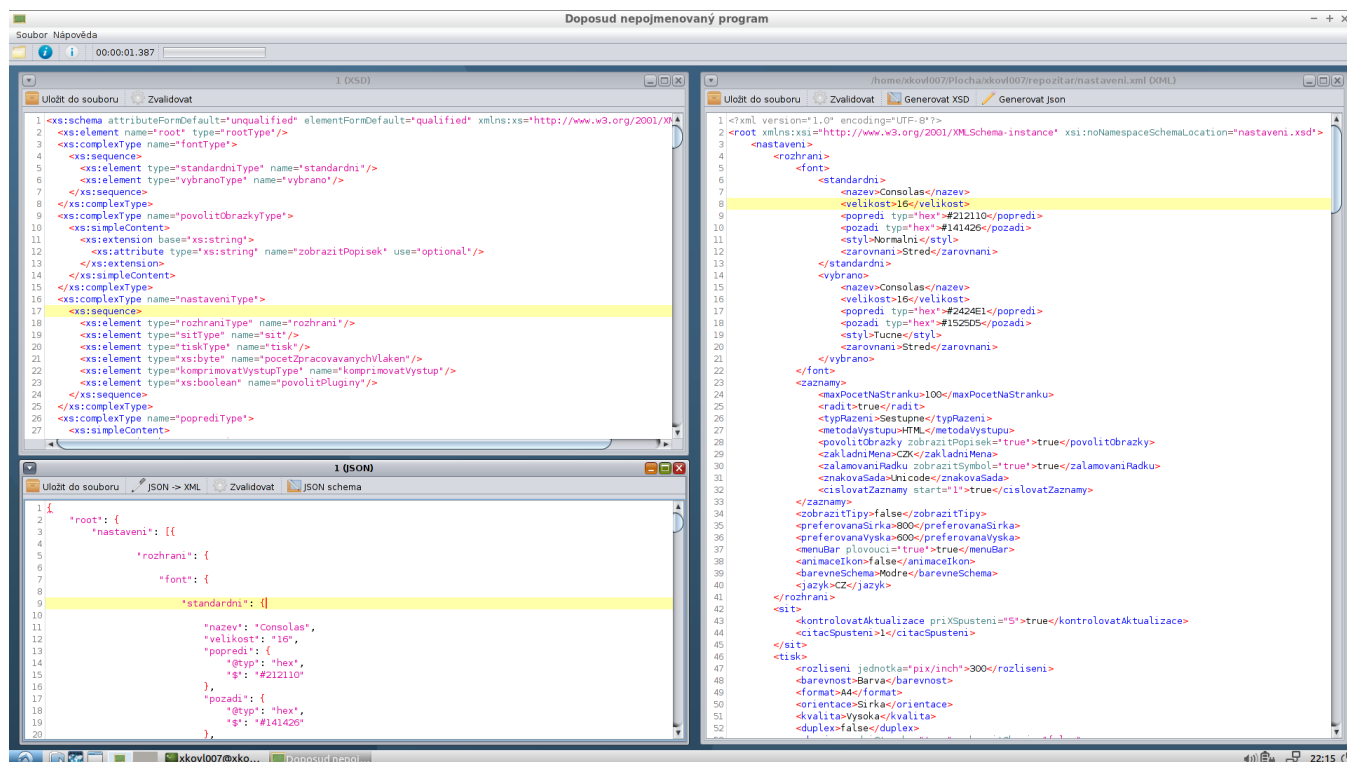
Nejprve byl navržen XML soubor, ze kterého bylo následně derivováno XML schéma a testován transformační dokument pro převod XML souboru do JSON souboru.

### 2.1 Použité nástroje

Pro vytvoření souborů bylo využito většího množství programů a nástrojů, přičemž jeden program dokonce vznikl samostatně pouze pro zjednodušení práce s tímto projektem.

Název nástroje	Důvod využití
<b>NetBeans</b>	Kontrola syntaxe, transformace XML, pretty printing
<b>Doposud nepojmenovaný program</b>	Kontrola syntaxe, kontrola správnosti JSON výstupu, podpora při tvorbě XML schématu, ověření XML schématu, kontrola html, transformace, pretty printing
<b>Kate</b>	Kontrola syntaxe, html, L <sup>A</sup> T <sub>E</sub> X
<b>Bluefish</b>	ditto
<b>wc (word counter)</b>	Program pro počítání znaků/řádek na Unixových operačních systémech
<b>pdflatex</b>	Generování výstupního PDF souboru z vstupního L <sup>A</sup> T <sub>E</sub> X souboru

Tabulka 2: Využité programy



Obrázek 1: Snímek obrazovky doposud nepojmenovaného programu

### 3 Popis elementů a atributů

Zamýšlená konfigurace aplikace se zaměřuje na tisk, velikost oken, síť a určitý druh záznamů.

Zde je uveden výčet některých elementů a atributů, které se vyskytují v XML souboru, jeho schématu a vygenerovaném JSON souboru.

XML obsahuje jeden kořenový element "**root**", do kterého je vnořeno celkem 5 konfigurací (ukázkových záznamů), jak je požadováno v zadání. Do samotné konfigurace jsou vnořeny další elementy, některé elementy mají ještě zadány atributy.

Element "**rozhraní**", který je vnořen do elementu "**nastavení**", představuje konfiguraci (hypotetického) grafického rozhraní, které je zobrazeno koncovému uživateli.

V elementu **"rozhrani"** se nachází element **"font"**, který nastavuje vzhled písma (**"standardni"** a **"vybrano"**), tedy dvě konfigurace, pokud je text nevybrán a vybrán; u sub-elementu **"popredi"** a **"pozadi"** je atributem uvedeno, v jakém formátu je zadaná hodnota barvy uvedena (příklad: hex: #161616 nebo dec: 100,50,100).

Dalším sub-elementem v "**rozhrani**" je i element "**zaznamy**", který může představovat seznam záznamů, které jsou uživateli představovány.

"**zobrazitTipy**" nastavuje, zda se uživatel po startu budou zobrazovat (nepříjemné) tipy programu.

"preferovanaSirka" a "preferovanaVyska" určují, jak velké bude okno aplikace při startu.

"menuBar" označuje lištu nástrojů a zda má tato lišta být zobrazena.

"**nastaveni**" dále obsahuje element "**sit**", který určuje, jakým způsobem bude program pracovat se sítí; v tomto případě je k dispozici sub-element, který určuje, zda-li se budou kontrolovat aktualizace a atribut určuje, jak často tomu tak bude (kolikrát spuštění aplikace), přičemž čítač spuštění aplikace je uveden v dalším elementu, v tomto případě je tedy při každém otevření aplikace soubor s nastavením nutné pokaždé aktualizovat, což v některých případech nemusí být žádoucí, v takovém případě je možné čítač odstranit a nechat pouze boolean hodnotu, jestli má aplikace být aktualizována při každém spuštění.

Element "**tisk**" určuje nastavení tiskárny při posílání výstupu na tiskárnu.

"rozliseni" je uváděno včetně atributu, který určuje jednotku.

Atribut "**uvodniStranka**" u elementu "**okraje**" určuje, zda-li se i úvodní stránka bude řídit okraji zbytku dokumentu. V opačném případě by na tuto úvodní stránku byl vytištěn pouze název dokumentu.

Atribut "**zobrazitOkraje**" zapíná nebo vypíná možnost vytištění okrajů pomocí vodících čar.

"duplex" zapíná/vypíná nastavení tisk na obě strany papíru, pokud to tiskárna umožňuje.

"**orezoveZnacky**" je element, který zapíná/vypíná tisk ořezových značek (při potřebě oříznout výsledný papír), využívá se v profesionálním tisku.

Element "**pocetZpracovavanychVlaken**" určuje, jaké maximální množství vláken bude odstartováno, pokud program vykonává náročnou (ale jednoduše paralelizovatelnou) úlohu.

"**komprimovatVystup**" určuje, zda-li výstupní soubor programu (jeho data) budou komprimována a na jaké úrovni.

"**povolitPluginy**" určuje, zda-li program umožní načtení podprogramů dodávaných třetí osobou.

XML bylo validováno mimo jiné na adrese <http://codebeautify.org/xmlvalidator>

## 4 Popis transformace

Transformační soubor do JSON se poněkud liší od transformací, které byly probírány na cvičení. V tomto případě si transformační soubor automaticky bere názvy elementů, jejich atributů a potomků a vytváří výstupní JSON soubor.

### Varování!

Transformační soubor byl zamýšlen pro transformaci pouze jediného souboru - XML souboru obsaženého v tomto projektu, na němž byl taktéž testován, na jiné XML soubory s jinou strukturou patrně nebude fungovat.

## 5 Popis schématu

Schéma je vygenerováno pomocí vzoru "Venetian blind", který od ostatních vzorů neobsahuje žádné zásadní nevýhody, programem vytvořeným pro tento projekt.

Schéma bylo následně upraveno a byly přidány některé restriktce (u elementů typu string jsou uvedeny výčtové typy, u číselných hodnot minimální a maximální hodnota).

Schéma bylo validováno vůči XML souboru.

## 6 Popis L<sup>A</sup>T<sub>E</sub>X

Byť zadání v L<sup>A</sup>T<sub>E</sub>X není podmínkou získání zápočtu, je tento soubor přesto přiložen v naději získání většího počtu bodů.

Výsledný PDF dokument je vysázen pomocí programu pdflatex.

## 7 Popis html

html dokument byl kontrolován pomocí validátoru specifikovaného v zadání projektu na adrese <https://validator.w3.org/>  
html dokument má 12393 znaků.

## 8 Popis výstupu JSON

Vygenerovaný JSON dokument byl validován pomocí webové služby na adrese <http://jsonlint.com/>

## 9 Závěr

Cílem projektu bylo vytvořit XML soubor obsahující konfiguraci smyšlené aplikace, vytvoření souboru s XML schématem a vytvoření souboru s transformačním modelem do JSON, přičemž pomocí tohoto souboru měl být vygenerován výstupní JSON soubor. Toho bylo docíleno.