Dokumentace úlohy JSN: JSON2XML v Perlu do IPP 2012/2013
Jméno a příjmení: František Koláček
Login: xkolac12

# Dokumentace k projektu do předmětu IPP, JSON2XML v Perlu

## Analýza problému

Mým úkolem bylo vytvořit skript v jazyce Perl, který bude provádět konverzi vstupního souboru obsahující JSON do jazyka XML. Formát výsledného XML souboru lze parametrizovat předáním parametrů při spuštění skriptu.

## Postup řešení

Prvním krokem při spuštění skriptu je kontrola a zpracování parametrů, jež byly skriptu předány z příkazové řádky. V dalším kroku se skript pokusí otevřít vstupní soubor obsahujícího JSON a při úspěšném otevření jej načte celý do paměti a vstupní soubor uzavře. Po načtení vstupních dat tyto data skript dekóduje pomocí knihovny `JSON::XS` a takto dekódovaná data rekurzivně prochází a pomocí knihovny `XML::Writer` je zapisuje do výstupního souboru. V případě úspěšné konverze se skript ukončí a vrátí nulovou návratovou hodnotu a v případě chyby se na standardní chybový výstup vypíše chybové hlášení a skript vrátí odpovídající chybovou návratovou hodnotu dle zadání.

## Zpracování parametrů

Pro zpracování parametrů se využívá funkce `parseArguments`, které předávám jako argument pole všech parametrů z příkazové řádky (`@ARGV`). Tato funkce pole sjednotí v jeden řetězec, kde všechny parametry jsou odděleny mezerou a poté je tento řetězec porovnáván se všemi možnými parametry přípustnými dle zadání. Pokud je daný parametr nalezen, nastaví odpovídající globální proměnné, případně zkontroluje, jestli je takto zadaná množina parametrů korektní a pomocí regulárního výrazu tento parametr z tohoto řetězce vyjme. Pokud na konci zpracování parametrů tento řetězec obsahuje něco jiného krom bílých znaků, byl zadán neplatný parametr a program se odpovídajícím způsobem ukončí.

## Zpracování vstupního (JSON) souboru

Pro zpracování vstupních dat ve formátu JSON využívám knihovnu `JSON::XS`, které jako parametr pro metodu `decode` předávám obsah vstupního souboru a tato metoda mi vrátí naplněný objekt typu `Array` nebo `Hash`. V případě chyby knihovna vyvolá výjimku na základě které se skript odpovídajícím způsobem ukončí.

## Generování výsledného XML dokumentu

Pro vytvoření výsledného XML dokumentu je implementována funkce `createXML`, jež využívá knihovnu `XML::Writer`, které při inicializaci předám jako parametr otevřený výstupní soubor reprezentovaný objektem knihovny `IO::File`. Po inicializaci a případném zapsání XML hlavičky a kořenového elementu XML (pokud byly zadány odpovídající parametry) se rekurzivně volají funkce `processArray` (pro zpracování objektu typu `Hash`) a `processHash` (pro zpracování objektu typu `Hash`), které vnitřně volají funkci `processData` pro zpracování hodnot typu klíč => hodnota. Tato funkce volá dvě pomocné funkce `createXMLElement` (vytvoření textového elementu) a `createXMLAttribute` (vytvoření elementu s atributy).

## Zpracování chyb

Pro zpracování chyb v implementovaném skriptu využívám funkce `printError`, které předávám jako první parametr chybové hlášení a jako druhý parametr požadovaný návratový kód při ukončení skriptu.

## Závěr

Pří testování výsledného skriptu jsem využil všech veřejně přístupných testů (včetně těch, které byly přiloženy k zadání) a všemi těmito testy skript prošel. Testování jsem prováděl na operačním systému GNU/Linux Mint a také na referenčním školním serveru Merlin.