Dokumentace úlohy JSN: JSN2PHP v PHP 5 do IPP 2016/2017
Jméno a příjmení: Jiří Matějka
Login: xmatej52

1 Zadání, použité funkce a třídy

Úkolem bylo vytvořit script, který převede text ve fromátu json do formátu xml. Jazyk php obsahuje třídu XMLWriter, která mi zpracování této úlohy značně usnadnila. Dále jsem pro načtení vstupních dat použil funkci `json_decode()`.

2 Postup řešení

2.1 Zpracování parametrů programu

Parametry (argumenty) programu jsem zpracoval pomocí třídy, ve které uchovávám nastavení celého programu (jména elementů pro pole, itemy, zda se má generovat hlavička dokumentu či ne, ...) a probíhala tam kontrola, zda všechny argumenty byly zadány ve správném formátu.

2.2 Načítání vstupních dat

Vstupní soubor byl načten najednou pomocí volání funkce `file_get_contents()`, která načte obsah celého souboru do proměnné. Nad touto proměnnou jsem zavolal už jen funkci `json_decode()`, která mi v případě úspěchu vrátila pole připravené pro další zpracování.

2.3 Zpracování načtených vstupních dat

Zpracování dat byla pro mě nejtěžší část této úlohy, hlavně proto, že script měl spoustu parameterů a každý parametr mohl úplně změnit výstupní formát. Ve scriptu jsou funkce určené na řešení jednotlivých podúloh: zápis objektu, pole, přidání atribut (popřípadě tvorba nových elementů) a přidání datových typů. Zápis vnořených polí a objektů řeším rekurzí.

2.4 Vytvoření výstupního souboru

Výstupní soubor se otvírá až poté, co je celý vstup zpracován, a pouze tehdy, když při zpracování nenastala žádná chyba. Zápis do souboru je dělán voláním funkce `fwrite()` nad otevřeným souborem a nad prvkem třídy XMLWriter.

2.5 Řešení chybových stavů

Při kontrole návratových hodnot některých volaných funkcí jsem měl problém rozlišit, kdy při podmíněném skoku psát '==' a kdy psát '==='. Navíc se mi na chybový výstup často vypisovalo varování, které se mi nelíbilo. Nakonec se mi tento problém podařilo vyřešit tak, že jsem každou chybu vyhodnotil jako výjimku a tu jsem se snažil odchytit.

2.6 Testování

Během tvorby scriptu jsem script testoval na svém stroji, (Ubuntu 16.04) a to interpretem PHP 7. Až když se výsledek výstupu programu blížil přiloženým testům, začal jsem tvořit vlastní testy a program jsem spouštěl na školním serveru Merlin s verzí interpretu PHP 5.6.

2.7 Odevzdání

Úlohu jsem se snažil dokončit tak, abych stihl pokusné odevzdání. Až na pár drobných úprav se mi to podařilo i s rezervou několika dnů a hodnocení programu bylo v rozmezí od 80% – 100%. Dokumentaci jsem tvořil až po pokusném odevzdání a to pomocí LaTeXu. Při dokumentaci kódu jsem se snažil dodržet formát dokumentovacího nástroje Doxygen pro dokumentaci funkcí, proměnných, tříd a metod.

3 Shrnutí

Díky tomuto projektu jsem si vyzkoušel práci s jazykem PHP. Velice se mi líbily dokumentační stránky k jednotlivým funkcím a třídám. Nevyhovovala mi především volnost, kterou mi php poskytovalo, zejména v podobě přetypování.