Implementační dokumentace k projektu k první úloze do IPP 2018/2019

Jméno a příjmení: Pavel Podlužanský

Login: xpodlu01

parse.php

Skript `parse.php` byl odevzdán ve dvou souborech, kde jeden z nich (`scanner.php`) vykonává lexikální analýzu a druhý z nich (`parse.php`), vykonává syntaktickou analýzu. Dále `scanner.php` kontroluje ošetření argumentů a parse.php generuje výsledný XML soubor.

Lexikální analýza

V části `scanner.php`, je implementovaná lexikální analýza pomocí konečného automatu. Konečný automat vždy zpracovává právě jeden řádek, který ověří, jestli jsou operační kód a operandy správně napsané. Scanner řádek po řádku posílá do souboru `parse.php`. Tento řádek posílá tak, jak byl napsaný, kde ví o jeho konci pomocí konce řádku. Zároveň posílá jako token i `EOL` nebo `EOF`. Řádek je poslán do syntaktické analýzy pomocí pole, do kterého je pomocí funkce `array_push` vždy vložen typ tokenu a jeho název.

Při kontrole argumentů, ověřuje program, jestli je správný počet argumentů, nebo jestli se první argument rovná --help. Pokud argumenty nesedí, nebo první argument není –help, vrátí se chybový kód 10.

Syntaktická analýza

Řádky v `parse.php` jsou zpracovávány postupně ve `while` cyklu, který navíc obsahuje switch. Na každém řádku program nejdřív zpracuje operační kód. O tom, který operační kód je právě na řádku rozhoduje zmiňovaný switch. Pak probíhá kontrola, jestli je dodržen správný počet operandů, a jestli náhodou některý z operandů není `EOL` nebo `EOF`. Dále probíha i kontrola, jestli daný typ tokenu může následovat za daným operačním kódem.

XML generování

Na generovaní výsledného xml souboru využívám knihovnu `DomDocument`. Nejdřív se vygeneruje povinná hlavička a následně jsou generovány elementy, podle toho, jaký operační kód nebo operand se nachází na řádku. Pří generování xml jsou speciální znaky jako třeba (&,<,>…) nahrazený pomocí funkce htmlentities(), která konvertuje všechny znaky na jejich reprezentaci v HTML.