

## Implementační dokumentace k 1. úloze do IPP 2018/2019

Jméno a příjmení: Martin Macháček

Login: xmacha73

### 1 Úvod & Úloha skriptu

Úlohou bylo vytvořit skript napsaný v jazyce PHP, který přijímá zdrojový soubor nestrukturovaného imperativního jazyka IPPcode19 a pomocí naší implementace řešení vytvoří odpovídající výstup v podobě XML kódu. Skript funguje jako aplikace v příkazové řádce, v případě chyby skript vrací odpovídající chybovou hodnotu. Pokračování projektu předmětu IPP bude v podobě druhé úlohy – interpret.py.

Velice užitečnými se staly získané vědomosti v předmětu IFJ.

### 2 Skript parse.php

Skript parse.php má za úkol, jak je již zmíněné v úvodu, načíst vstupní soubor v jazyce IPPcode19 a přeložit ho na XML podobu a vypsát ho na standardní výstup. K implementaci tohoto řešení je použita knihovna SimpleXML, která poskytuje velice užitečné funkce, a do určité úrovně zjednodušila poslední část skriptu – generování.

První částí skriptu je kontrola zadaných argumentů od uživatele. Skript povoluje jako argument pouze --help, vstupní soubor se zadává v podobě standardního vstupu, tudíž žádný jiný stav v argumentu nemůže nastat a popřípadě skript vrací odpovídající návratovou chybovou hodnotu.

Další částí je načítání dat ze vstupního souboru s využitím kombinace cyklu while a funkce fgets. Zpracování těchto dat je prováděno postupně, po jednotlivých řádcích (příkazech). Nejdříve skript zkontroluje, jestli na prvním řádku vstupního souboru je uvedena hlavička: IPPcode2019, poté ořeže každý řádek o přebytečné mezery/tabulátory, provede odstranění komentářů a postupně posílá tyto čisté příkazy do syntaktické analýzy.

Syntaktická analýza spočívá v kontrole správnosti posloupnosti příkazů. Do syntaktické analýzy je poslán vždy jeden příkaz, kde se nejdříve zkontroluje, do jaké skupiny instrukcí patří. Tyto skupiny jsou rozvrhnuté podle jejich operandů (tří, dvou, jedno operandové a instrukce bez operandu). Po zařazení příkazu do skupiny se každý operand této instrukce zkontroluje a rozkouskuje pomocí regex a v případě korektního zápisu je operand poslán do funkce setup\_instruction, která zařídí přejmenování klíčů (např array[1] se přejmenuje na array[type]) a přidá příkaz do pole všech příkazů pro lepší přehlednost kódu při nadcházejícím generování.

Samotné generování již není prováděno po řádcích v cyklu while, ale je zde využito již zmiňované pole všech příkazů a knihovna SimpleXML, která je schopna převést PHP formu zápisu objektu na samotný XML element. Nejdříve je nutno vytvořit hlavičku s potřebnými atributy, a poté pomocí vnořených cyklů for, které přistupují k jednotlivým příkazům a atributům uložených v poli instructions, začít vytvářet a vnořovat jednotlivé elementy s atributy. Tyto elementy se postupně ukládají do proměnné, kterou skript na konci pouze vypíše pomocí funkce echo.