

## Implementační dokumentace k 1. úloze do IPP 2019/2020

**Jméno a příjmení:** Jan Šulavík

**Login:** xsulav01

### Popis skriptu parse.php

Úlohu jsem řešil procedurálním programováním. Jádrem skriptu je cyklus, který čte řádky vstupního souboru zadaného ze standardního vstupu až do jeho konce. Při prvním průchodu je kontrolována přítomnost a správnost povinné hlavičky *'IPPCode20'*, v případě špatné nebo chybějící hlavičky je skript ukončen s odpovídajícím návratovým kódem. Po úspěšné kontrole je volána funkce `makeXMLHeader`, která vytvoří resource `$xw` pomocí funkce `xmlwriter_open_memory` z rozšíření `XMLWriter`. Tento resource ukazuje do paměti, kde se zapisuje XML reprezentace kódu. Funkce dále vytvoří XML hlavičku a kořenový element *program* s atributem *language* s hodnotou *IPPCode20*.

V každém cyklu je řádek nejprve upravován a poté zpracováván. K úpravám patří odstraňování komentářů uživatelsky definovanou funkcí `removeComments`, odstraňování bílých znaků zabudovanými funkcemi `ltrim` a `rtrim`. Následně je řádek rozdělen na jednotlivé tokeny, které reprezentují opcodes a argumenty jazyka *IPPCode20*. Po rozdělení je pole tokenů předáno funkci `getInstruction`, která dle hodnoty opcode volá funkce na kontrolu argumentů podle toho, jaký je jejich očekávaný typ. Opcodes jsou ve skriptu rozděleny do skupin se stejnou syntaxí pro zkrácení celkového zápisu. Je potřeba kontrolovat syntaxi proměnných, symbolů, návěští a typů – o to se starají funkce `varSyntax`, `symbolSyntax`, `labelSyntax` a `typeSyntax`. Tyto funkce kontrolují syntaxi argumentů pomocí regulérních výrazů. Pokud kontrola syntaxe proběhla v pořádku, je pro opcode zavolána funkce `writeInstruction`, která má na starost zápis XML reprezentace instrukce v jazyce *IPPCode20*. V těle funkce nejprve dojde k odstranění bílých znaků ze všech tokenů, poté je vytvořen element *instruction* s argumenty *order* a *opcode*, které jsou naplněny odpovídajícími hodnotami. Poté je volána funkce `writeArgs`, která zapisuje XML elementy představující argumenty instrukce. Funkce je volána cyklem tolikrát, kolik je argumentů a pokaždé zapíše jeden z nich. Jakmile skript vystoupí z hlavního cyklu `while` po přečtení posledního řádku vstupního souboru, je volána funkce `endXML`, která ukončí element *program* a vytiskne obsah paměti, kam ukazuje `$xw`, na standardní výstup jako XML reprezentaci vstupního kódu.

Pokud někdy za běhu skriptu nastane lexikální, syntaktická, případně jiná chyba (např. při přidělování paměti) je skript ihned ukončen s odpovídajícím návratovým kódem.