

# Implementační dokumentace k 1. úloze do IPP 2020/2021

Jméno a příjmení: Roman Stepaniuk

Login: xstepa64

## Zadání

Vytvořit php skript pro kontrolu lexikální a sémantické správnosti kódu v jazyce IPPcode21 s využitím DKA a regulárních výrazů a výpisu na standardní výstup XML reprezentaci programu.

## Popis programu

- Práce s argumenty

Program analyzuje vstupní argumenty pomocí `argc`, `argv`. Analýza se provádí ve funkci `«arguments_check($argc, $argv)»`. Analýza argumentů rozšíření se provádí ve stejné funkci a je realizovaná pomocí `«switch case»` a regulárního výrazu (pro získání jména souboru). Když bude využito cokoliv kromě dovolených argumentů, skript se ukončí chybou 10.

- Načtení a výpis v programu

Skript načítá informaci se standardního vstupu (`stdin`). Pro načtení byla využita funkce `«fgets(STDIN)»`. Je volaná v cyklu `«while»`, takže čte řádky do konce vstupu.

Výsledek programu je vypsán na standardní výstup (`stdout`).

Výsledky prací rozšíření se zapisují do souboru. Před zápisem program kontroluje jestli ten soubor je otevřený pro zápis, a když ne, tak výsledkem programu bude chyba 12.

- Lexikální a sémantická analýza vstupního kódu

Analýza vstupního kódu je realizovaná pomocí kombinace stavových automatů (`switch case`) a regulárních výrazů (`preg_match(expr)`).

Zprv program transformuje vstupní řetězec na pole slov pomocí funkce `«explode(' ', $str)»`, pak prochází tímto polem a zapisuje jen podstatné slova (ignoruje bílé znaky všechno za `#`) ve výsledné pole slov. Všechny bílé znaky a komentáře se nebudou už vyskytovat.

Pak proběhne kontrola hlavičky souboru a `«switch case»` automat pro analýzu prvního slova na řádku (předpokládáme že toto bude instrukce) a na tomto základě bude rozhodnutá cesta další kontroly. Pak proběhne analýza počtu argumentu nutné instrukci. Proto bude zavolána funkce `«NeedLen()»`. Dál budou zavolané funkce kontroly typů (argumentů instrukce).

Když se hlavička nenajde, program se ukončí chybou 21, když bude špatná instrukce (element s nultým indexem), tak chybou 22, jinak chybou 23.

V jazyce IPPcode21 máme 4 základní typy, (`var`, `symb`, `type`, `label`), takže pro jejich analýzu byly vytvořeny funkce: `checkVariable()`, `checkSymb()`, `checkType()`, `checkLabel()`, které kontrolují slova pomocí sady podmínek a regulárních výrazů.

Když všechny kontroly projdou dobře, program se ukončí kódem 0.

- Vytvoření výstupního XML kódu

Pro vytvoření výsledného XML byl využit modul `XMLWriter`, který generuje XML na základě vyvolání jeho funkce (např. `«xmlwriter_start_element()»`) a předání tam nutných parametrů. Pak přidává ten kód do svého bufferu a ten buffer byl vypsán na konci programu.

- Implementace rozšíření

Rozšíření bylo implementováno pomocí «switch case» a řady podmínek ve hlavním stavovém automatu programu. Bylo vytvořeno pomocné pole na 7 proměnných, kam se ukládali spočítané výsledky, které par se vypisovali do zadaného souboru.

Bylo vytvořeno dva pomocné poli «jump\_array», «label\_array». Na základě jich hodnot byly spočítané [fwjumps], [backjumps] a [badjumps].

Nejsložitějším problémem rozšíření byl badjumps, který byl implementován tak: po ukončení syntaktické analýzy projít každým prvkem poli «jump\_array» a kontrolovat, jestli ve poli «label\_array» také vyskytne stejná hodnota. Když ne, tak tento navěští neexistuje.