Implementační dokumentace k 1. úloze do IPP 2022/2023
Jméno a příjmení: Josef Kuchař
Login: xkucha28

Návrh implementace

Z důvodu jednoduchosti první úlohy nebyl použit strikně objektově orientovaný přístup. Pouze byly vytvořené výčtové typy a jedna třída pro zpřehlednění kódu. Výčtovými typy jsou: *Arg*, *StatusCode*.

Arg je výčet možných argumentů instrukcí: *Variable* (proměnná), *Symbol* (proměnná, nebo literál), *Label* (návěští), *Type* (typ).

StatusCode je vyčet možných návratových kódu popsaných v dokumentaci jazyka IPPcode23. Obsahuje metodu *get*, která vrací hodnotu návratového typu. Toto je třeba kvůli vlastnosti jazyka PHP, který neumožňuje implicitní přetypování výčtového typu na datový typ *int*.

Třída *Re* obsahuje definice užitečných regulárních výrazů. Tyto definice jsou schované ve vlastní třídě, aby nezasahovaly do globálního prostoru. Pro stejný účel by se dala použít functionalita *namespace*.

Implementace také obsahuje globální pole *INSTRUCTIONS*, které obsahuje všechny podporované instrukce a jejich argumenty.

Zpracování argumentů a načtení vstupu

Pro zpracování argumentu příkazové řádky byla použita vestavěná funkce *getopt*. Pokud se použije přepínač *--help*, vypíše se nápověda a program se ukončí s návratovým kódem 0.

Zdrojový kód ze standardního vstupu se načítá pomocí vestavěné funkce *file_get_contents*. To znamená, že se celý vstupní zdrojový kód naráz načte do jednoho řetězce.

Předzpracování vstupu

Před samotnou lexikální a syntaktickou analýzou je vstup předzpracován pro snadnější práci. Nejprve je vstup převeden na pole řádků. Poté je každý řádek pomocí vestavěné funkce *array_map* zpracován. Nejprve jsou pomocí regulárního výrazu odstraněny komentáře. Poté jsou pomocí vestavěné funkce *trim* odstraněny nadbytečné mezery nazačátku a nakonci řádku. Nakonec je řádek podle mezer rozdělen na pole. Prázdné řádky jsou pomocí vestavěné funkce *array_filter* odstraněny. Výstupem této operace je tak pole polí.

Analýza a generování XML

Prvně je kontrolována přítomnost a správnost hlavičky *.IPPCode23*. Poté je každý řádek postupně ověřován. V seznamu *INSTRUCTIONS* se nalezne odpovídající instrukce a je ověřen počet argumentů. Každý argument je poté pomocí odpovídajících regulárních výrazů ověřen. Regulární výraz také slouží pro získání samotné hodnoty argumentu. Například pro int@5 vrátí pouze hodnotu 5. Pro ověření správnosti číslených literálů byly použity oficiální regexy z PHP dokumentace[1].

Pro generování výsledného XML byla zvolena knihovna *SimpleXML*. Jak už jméno napovídá, je opravdu jednoduchá na použití. Automaticky ošetřuje speciální znaky, jak je to napsané ve specifikaci. Generování XML probíhá v průběhu lexikální a syntaktické analýzy. Celé XML je vypsáno na standardní výstup až nakonec.

Testování

Pro ověření implementace byly použity jak studentské testy, tak oficiální příklady.

[1] https://www.php.net/manual/en/language.types.integer.php