

IPP - Principy programovacích jazyků a OPP 2019/2020

Implementační dokumentace k první úloze

Jméno a příjmení: Vojtěch Mimochodek

Login: xmimoc01

Analyzátor kódu v IPPcode20

Script implementovaný v jazyce PHP7.4 má za úkol provést syntaktickou a lexikální analýzu kódu v jazyce IPPcode20 a převést jej na odpovídající XML reprezentaci.

Script pracuje s těmito parametry:

- --help – slouží k vypsání nápovědy
- --stats=file – slouží k výpisu statistik do zvoleného souboru. „File“ označuje jméno daného souboru se statistikami
- --loc – slouží k výpisu všech řádků s instrukcemi do statistik
- --comments – slouží k výpisu všech komentářů do statistik
- --jumps – slouží k výpisu všech instrukcí pro skoky, volání a návraty z volání do statistik
- --labels – slouží k výpisu návěští do statistik

Parametry --loc, --comments, --jumps, --labels lze použít pouze s parametrem --stats=file.

Implementace

Celý projekt je naprogramován v jednom souboru *parse.php*.

Na samém začátku scriptu jsou deklarovány potřebné globální proměnné a inicializace *DOMDocumentu* pro následné generování XML reprezentace.

Script ke svému chodu používán několik funkcí. První důležitou funkcí je *check_arguments(\$argc, \$argv)*. Tato funkce zajišťuje kontrolu vstupní parametrů. V případě chyby v těchto parametrech je script ukončen s návratovou hodnotou 10.

Dalšími funkcemi jsou *generate_instruction(\$dom, \$program, \$words)*, *generate_argument(\$dom, \$type, \$num_arg, \$attribute)* a *generate_symb(\$words)*.

Tyto funkce slouží ke správnému vygenerování XML. První z funkcí zajišťuje generování těla instrukce s atributy *order* a *opcode*. Druhá funkce generuje argumenty těchto instrukcí s atributem *type*. Ten může být *var*, *label*, *type*, *string*, *bool* či *int*. Třetí funkce přímo nic negeneruje, nýbrž upravuje přijatý typ symb a upravuje jej na patřičnou podobu. Po této úpravě je vrácená hodnota z této funkce posílána opět do funkce *generate_argument* k vygenerování.

Pro správnou syntaktickou kontrolu scriptu slouží čtveřice funkcí. Jsou to: *check_var(\$var)*, *check_label(\$label)*, *check_symb(\$symb)* a *check_type(\$type)*. Tyto funkce kontrolují jednotlivé argumenty zadaných instrukcí pomocí regulárních výrazů.

Poslední nejdůležitější funkcí je *parse(\$words, \$words_count, \$dom, \$program)*. Tato funkce přijímá jednotlivé řádky rozdělené na slova uložená v poli. Podle názvu instrukce rozliší, o jakou instrukci se jedná a zařadí je do patřičné kategorie. To se provádí pomocí přepínače *switch*. Kategorie jsou čtyři. První obsahuje instrukce s nula operandy. Druhá s jedním, třetí se dvěma a poslední se třemi. Podle toho o jakou instrukci v dané kategorii se jedná, jsou volány funkce s regulárními výrazy pro syntaktickou a lexikální kontrolu. V případě, že chyba nenastane, jsou volány zmiňované funkce pro generování. Jestliže se script spouští s příslušnými parametry pro vypsání statistik, probíhá zde také inkrementace jednotlivých hodnot, které pak tvoří statistiku. Jedná se o rozšíření STATS.

V části *default*: switche funkce *parse()* probíhá ošetření povinné hlavičky *.IPPCODE20*.

Script funguje na základě smyčky *while*, která načítá ze standartního vstupu, dokud nenarazí na *eof*. Načte vždy jeden řádek. Ten ořeže o přebytečné bílé znaky a komentáře. Řádek rozdělí pomocí integrované funkce *explode()* na slova a uloží je do pole *\$words*. S tímto polem je volána funkce *parse()*.

Za předpokladu, že vše proběhne bez chyb a script není předčasně ukončen s některou z chybových návratových hodnot (př. 22, 23), je na konci zavolána funkce pro výpis vygenerovaného XML či statistik, pokud je o ně požádáno skrz parametry. Nakonec je script je ukončen s návratovou hodnotou 0.