

## Implementačná dokumentácia k 1. úlohe do IPP 2022/2023

Meno a priezvisko: Adam Pap

Login: xpapad11

Skript `parse.php` načíta zo štandardného vstupu zdrojový kód od užívateľa. Výstupom skriptu je XML reprezentácia zadaného kódu. Chybové hlášky sú vypisované na štandardný chybový výstup.

V prípade potreby je možné si nechať vypísať na štandardný výstup nápovedu parametrom `--help`.

Vypisovanie a generovanie spracovaného kódu do XML formátu zabezpečuje knižnica `XMLWriter`. Syntaktická kontrola zadaného kódu a samotné generovanie XML tagov s dátami je vykonávaná funkciou `syntax_check_xml_build()`, ako parameter táto funkcia prijíma `STDIN`, `STDOUT`, `STDERR`, pole inštrukcií, ktoré program podporuje, ukazateľ pre `XMLWriter` a pole určené pre štatistiku z rozšírenia `STATP`. Samotný parsing prebieha, že sa vyžiada spracovaný riadok od lexikálneho analyzátoru v cykle, ktorý skončí keď lex. analyzátor vráti značku konca súboru `EOF`. Ak je v danom riadku detekovaná inštrukcia, dôjde k overeniu syntaktickej správnosti (počet argumentov, dátový typ parametrov). Kontroly sú volané prostredníctvom funkcií, ináč by sa v skripte nachádzalo priveľa opakujúceho sa kódu.

Funkcia `num_of_param_check()` kontroluje počet argumentov danej inštrukcie, ako parameter prijíma `$type_of_instruction`, spracúvaný riadok, `STDERR`, kde `$type_of_instruction` je konštanta ktorá udáva pre danú skupinu „case-ov“ počet parametrov ktoré by daná inštrukcia mala mať. V prípade chyby sa vypíše na `STDERR` chyba 23. Funkcia `type_of_params()` kontroluje či je prvý parameter správneho typu, ak nie dôjde k chybe 23. Ako parametre typ parametru, spracúvaný riadok, `STDERR`. Funkcia `symb_param_check()` slúži na kontrolu 2. a 3. parametru u inštrukcií ktoré používajú typ `<symb>` čo môže predstavovať jak premennú tak konstantu. V prípade chyby funkcia vráti chybu 23.

Ak dané kontroly prebehnú v poriadku dôjde k samotnému generovaniu kódu zavolaním funkcie `XML_writer()`, ktorá ako parametre prijíma: ukazateľ pre `XMLWriter`, typ tagu (var, label apod.), hodnota, číslo argumentu (`arg1`, `arg2` apod.), atribút resp. (type). Po úspešnom zápise sa vyžiada nasledujúci riadok na syntaktickú kontrolu.

Lexikálna analýza prebieha pri každom zavolaní funkcie `lexer()` z funkcie syntaktickej analýzy. Ako parametre lexikálna analýza vyžaduje: `STDIN`, `STDOUT`, `STDERR`, pole inštrukcií. Ako prvú vec funkcia `lexer()` načíta vstup po riadkoch a jednotlivé riadky uloží do premennej, s ktorou sa ďalej pracuje: `$line_of_code = fgets($input) == false`, toto načítanie prebieha v cykle do doby než sa nenarazí na koniec súboru `EOF`. Následne sa kontroluje pomocou regulárnych výrazov a funkcie `preg_match()` správnosť daného riadku kódu. Komentáre a iné biele znaky sa pochopiteľne ignorujú. Funkcia vracia pole `$token_data_arr` v ktorom sa nachádzajú tokeny, kvôli rozlíšeniu či sa jedná o inštrukciu, hlavičku apod., ďalej sa tam nachádzajú dáta napr. meno premennej, prípadne v akom rámci (GF/LF/TF) sa nachádza a tiež číslo, ktoré identifikuje inštrukciu (toto číslo je z poľa inštrukcií v knižnici).

Rozšírenie `STATP` zabezpečuje vypísanie rôznych štatistík do textových súborov, volá sa príkazom `--stats=file.txt`. Základom je pole premenných (čítačov) v ktoré sa inkrementujú za splnených podmienok. Samotné zapisovanie do súborov je vyriešené tak aby sa zadané parametre pre štatistiku vypísali do súboru ktorému patria. To je zabezpečené existenciou poľa v ktorom sa nachádzajú názvy súborov ktoré boli rozpoznané ako argumenty zadané skriptu. To ale samo o sebe nestačilo a tak bolo potrebné ešte k samotným štatistickým prepínačom, ktoré definujú ktoré štatistiky sa budú vypisovať, pridať ešte číslo súboru tzv. index, ku ktorému patria.