Implementační dokumentace k 1. úloze do IPP 2019/2020
Jméno a příjmení: Vojtěch Jahoda
Login: xjahod06

1 Návrh

Vstup programu jde prvně do částí lexikální analýzy, která funguje na principu konečného automatu, který načítá vstup po řádcích. Načtené řádky poté analyzuje a rozhodne, kterou z 8 skupin parametrů si daná instrukce vyžaduje. Pokud se při načítání vyskytne chyba, tak se program ukončí s chybným stavem. Načtenou intrukci poté posílá syntaktické analýze, která pomocí regulárních výrazů kontroluje jejich správnost, a pokud najde nějakou nesrovnalost, tak se zavolá opět chybový stav.

Pro zjednodušení implementace jsem zvolil objektově orientovaný přístup, kde používám jeden univerzální objekt intrukce pro všechny různé druhy a jako argument mu předávám kolik a jakých parametrů instrukce má očekávat.

2 Implementace

Před začátkem cyklu na čtení řádků se kontrolují parametry, se kterými byl program volán. Pokud se objeví parametr `--help` tak se vypíše nápověda k obsluze programu. Tento parametr může být volán pouze sám, takže pokud je zavolán s více parametry tak nastane chyba. V jiném případě, když jsou nastaveny parametry pro rozšíření `STATP` tak se tyto argumenty uloží do objektu `statistics`, kde se pořadí výpisu uloží jako seznam, který se poté na konci programu prochází a případně vypisuje do souboru.

Program čte vstup po řádcích. Jako první se kontroluje výskyt bílých znaků a komentářů. V případě výskytu komentáře se přičte tento výskyt k objektu statistik k parametru celkového počtu komentářů kvůli rozšíření `STATP`. Čtení komentářů je realizováno pomocí funkce `preg_match`, která pomocí regulárních výrazů určí, kde se na řádku nachází komentář i to, že se nenachází, a následně extrahuje název instukce, parametry a zahodí komentář. V případě, že na řádku se nevyskytuje nic jiného než komentáře nebo prázdný řádek, je přeskočeno na čtení dalšího řádku. Poté, pokud je nastavena proměnná `start` na hodnotu `true`, je vykonána kontrola správnosti hlavičky zdrojového kódu. Při čtení následujícího řádku se již hlavička nekontroluje a přechází se k syntaktické kontrole právě přečteného operačního kódu a jeho argumentů. Nakonec je vygenerován dokument XML s náležitými parametry.

Třída `master_instruction` zajišťuje vytvoření instance instrukce a kontrolu jejích argumentů. Pro vytvoření nové instance třídy je nutné předat parametry pro konstruktor třídy: argumenty instrukce (`$params`), počet načtených argumentů instrukce (`$param_count`) a typy argumentů (jako seznam, `$arg_types`). Jako první se při konstruování třídy, podle typu operačního kódu, kontroluje správnost počtu argumentů. Pokud je počet argumentů v pořádku (lexikální analýza), pokračuje se na syntaktickou kontrolu argumentů. Syntaktická analýza je realizována třídou `master_instruction` pomocí metody `arg`. Je zde využíváno kontroly pomocí regulárních výrazů. Třída rozezná jaký typ argumentu dostala a zpracuje ho pomocí funkce `verify_<typ argumentu>`. Poté vytvoří proměnné svojí třídy s dynamickým jménem podle pořadí daného argumentu. Pokud někde v procesu vytváření instance třídy `master_argument` nastala chyba, třída nastaví svoji proměnnou `error_code` na některý (podle typu chyby) z chybových kódů. Tento stav se kontroluje po každém vytvoření daného objektu a případně se ukončuje běh programu s chybou daného objektu.

Program nakonec vytvoří XML dokumentu pomocí knihovny `DOMDocument`. Následně se vygenerovaný dokument ještě projede sérií regulárních výrazů, které nahradí znaky `"` a `'` za jejich obdobu XML excape sequence.