Implementační dokumentace k projektu do IPP 2017/2018
Jméno a příjmení: Lukáš Drahník
Login: xdrahn00

Návrh programu

Při návrhu všech modulů v projektu jsem se snažil dbát na co největší přehlednost kódu, na využívání vestavěných knihoven a funkcí a omezit tedy co nejvíce vlastní kód, který snižuje čitelnost, zvyšuje šanci na možnou chybu a v neposlední řadě zbytečně prodlužuje samotný zdrojový kód.

Ve všech modulech je primárně oddělené zpracování argumentů, zpracování vstupu a zápis na výstup. Je snaha pro veškerý kód, který na to má nárok vytvořit odpovídající funkci/metodou pro znovupoužitelnost a pro zvýšení čitelnosti a rozsekání tak zdrojového kódu na pojmenované sub celky.

Všechny moduly obsahují nápovědu.

Parser (`parse.php`)

Parser je dle zadání psán pro php verzi 5.6. Vzhledem k návrhu všech částí je zde funkce na zpracování argumentů (`parseArgs`), na parsování vstupu (`parseLanguage`) a na výpis na výstup (`writeXml`).

Nejdelší funkce a jádro parseru je metoda `parseLanguage` pro kterou byla vytvořena řada podpůrných funkcí ověřujících například validitu symbolu, labelu, proměnné, konstanty. Tato funkce je implementovaná jako velký switch do kterého se dostanou všechny řádky ze vstupu obsahující instrukce bez komentářů, bez mezer. Pokud zadaná instrukce neexistuje, vrací se chyba, pokud instrukce obsahuje nepovolený počet argumentů, špatné typy argumentů atd. vrací se chyba.

Vzhledem k naimplementovanému rozšíření STATP, které se týká pouze parseru byla vytvořena funkce `writeStats`, která zapisuje do uvedeného souboru statistiky (počet řádků s komentářem a počet instrukcí) dle pořadí v argumentech (každé na zvláštní řádek) v případě vyžádání rozšíření.

Interpret (`interpret.py`)

Interpret je dle zadání psán pro python verzi 3.6. Vzhledem k návrhu všech částí je i zde funkce na zpracování a validování argumentů (`parseCmdArgs, validateCmdArgs`) a provedení samotného programu (`run`).

Intepretování programu má řadu podpůrných funkcí, stejně jako v `parse.php` jsou zde funkce na validování předaného XML, používá se zde modul `xml.etree.ElementTree`, zkráceně `ET`, dále velký switch pro všechny instrukce, které jsou dále psány jako jednotlivé funkce. U každé funkce se kontroluje počet operandů, typ operandů, zda proměnná existuje a další náležitosti.

Na regulární výrazy použité především pro kontrolu názvů je použit modul `re` a metoda `match`, obdoba php metody `preg_match`.

U instrukce `BREAK` se vypisuje obsah všech rámců (GF, LF, TF) a hodnoty případného rozšíření `STATI`.

Rozšíření STATI

Statistiky rozšíření STATI se ukládají do souboru pouze při úspěšně dokončeném programu, pokud dojde k chybě během interpretace, soubor se ani nevytváří.

Instrukce `BREAK` vypisuje při aktivovaném rozšíření i hodnoty rozšíření ukládané do souboru.

Testovací rámec (`test.php`)

Testovací rámec je dle zadání psán pro php verzi 5.6. Stejně jako předešlé moduly jsou i zde oddělené základní funkce kódu do metod, tedy parsování argumentů (`parseArgs`), provedení testů (`runTests`), zápis na výstup (`writeHtml`).

Výsledná HTML stránka umožnovala vkládat javascript, případně css čehož bylo záměrně nevyužito. Tato stránka by měla být co nejvíce strohá. Slouží pouze jako informativní výpis a také by měla být čitelná i v terminálu při nepřesměrování do souboru a zobrazení v prohlížeči. Kvůli přehlednosti jsem zvolil i nezobrazovat stack naplněný konkrétní chybou případně rozdíl v diff a u konkrétních testů zobrazovat pouze false, true na znamení, že test byl proveden úspěšně či nikoliv. Důležité informace jsou zde jasně odlišeny a zvýrazněny.

Celková statistika se zobrazuje vždy jako ukazatel provedených testů správně/celkově + v závorce informace o výsledcích všech detailních testů správně/celkově. Celková statistika se také zobrazuje dle složek viz. zadání,

slouží především pro lepší orientaci pokud jsou testy tříděny na logické celky nebo jsou čerpány z různých zdrojů atp.

Vytváření archivu

Celý projekt je zabalen do souboru s příponou .tar dle zadání a to pomocí příkazu make tar v souboru Makefile. Bylo potřeba se vypořádat s přesunutím dokumentace z podsložky do rootu archivu. Vyřešeno pomocí argumentu `-C doc doc.pdf`.

Reference