

## Implementační dokumentace k 2. úloze do IPP 2018/2019

**Jméno a příjmení:** Jakub Frejlich

**Login:** xfrejl00

### Skript interpret.py

Tento skript načítá XML reprezentaci jazyka IPPcode19 a interpretuje jednotlivé instrukce. XML formát i vstup pro interpret lze načíst z externích souborů, případně lze jeden z výše uvedených načítat přímo ze standardního vstupu.

Pro načítání XML je využit balíček `ElementTree`. Takto načtené XML je následně zkontrolováno, zda odpovídá formátu ze zadání a neobsahuje nepovolené atributy a jména. U každé instrukce jsou za použití regulárních výrazů lexikálně zkontrolovány argumenty a jejich hodnoty jsou převedeny na příslušné typy. Dále jsou zkontrolovány operační kódy instrukcí a syntax jejich argumentů za použití funkcí `check_arg_number`, `check_variable`, `check_symbol`, `check_label` a `check_type`. Instrukce jsou postupně ukládány do seznamu a následně seřazeny podle atributu `ORDER`.

Pro reprezentaci hodnot je použita třída `t_token` s atributy `type` a `value`. Pro reprezentaci instrukce je použita třída `instruction` s atributy `opcode`, `order`, `arg_count`, `arg1`, `arg2`, `arg3`.

Před samotnou interpretací jsou funkcí `define_labels` definována všechna návěští pro skokové instrukce a jejich indexy jsou uloženy do slovníku. Rámce jsou implementovány taktéž pomocí slovníků. Interpretace je implementována pomocí konečného automatu. Užitím konstrukce `while` je vždy vyhledán stav podle příslušného operačního kódu a instrukce je posléze vykonána. Pro vykonávání instrukcí je důležitá funkce `handle_var`, která ze zápisu `frame@var` získá rámec a jméno proměnné a zkontroluje, jestli je proměnná v daném rámci definovaná, a funkce `handle_symbol`, která vrátí odkaz na konstantu nebo odkaz na proměnnou, u které zkontroluje, jestli je inicializovaná. Interpretace je ukončena vykonáním poslední instrukce, vykonáním instrukce `EXIT` nebo předčasně chybou (návratové kódy viz. třída `exit`).

### Rozšíření

**STATI** – implementováno ve třídě `t_stats`.

**STACK** – implementováno změnou získávání hodnot pro výkon instrukce u nezásobníkových verzí instrukcí.

**FLOAT** – implementováno za použití metod `hex` a `fromhex`.

### Skript test.php

Tento skript postupně spouští testy podle uživatelem upřesněného výběru a generuje HTML statistiku o jejich úspěšnosti na standardní výstup.

Funkce `test` hledá v zadaném adresáři všechny soubory s příponou `.src` a k tomuto souboru dále hledá ve stejném adresáři soubory stejného jména s příponami `.rc`, `.out` a `.in`, případně pokud některý ze výše uvedených souborů chybí, tak je vytvořen nový prázdný soubor (`out`, `in`) nebo soubor obsahující 0 (`rc`). Pro prohledávání adresáře je použita funkce `scandir`. V případě nastaveného rekurzivního prohledávání jsou prohledávány stejným způsobem i všechny podadresáře původně prohledávaného adresáře. Pro vyhledávání testů jsou využity regulární výrazy.

Nalezený test je ihned vykonán funkcí `perform_test`. Je možné provést test pouze pro skript `parse.php` nebo pouze pro skript `interpret.py` nebo pro oba zároveň. Skripty jsou spouštěny funkcí `exec`. Pokud není specifikována cesta k jednotlivým skriptům, bere se skript v aktuálním adresáři. Při neexistenci skriptů je testovací skript ukončen s chybovou hodnotou 11.

Při možnosti `-parse-only` je spuštěn skript `parse.php` a jako zdrojový kód v IPPcode19 je použit obsah souboru s příponou `.src`. Návratová hodnota skriptu se porovná s hodnotou v souboru s příponou `.rc` a výstup v dočasném souboru s příponou `.xml` je porovnán nástrojem `JEXamXML` se souborem s příponou `.out`.

Při možnosti `-int-only` je spuštěn skript `interpret.py` a jako zdrojová XML reprezentace `IPPcode19` je použit obsah souboru s příponou `.src`. Vstup pro `interpret` je načten ze souboru s příponou `.in`. Návrátová hodnota skriptu se porovná s hodnotou v souboru s příponou `.rc` a výstup v dočasném souboru s příponou `.tmp` je porovnán funkcí `diff` se souborem `.out`.

Při testování obou skriptů je spuštěn skript `parse.php` a jako zdrojový kód v `IPPcode19` je použit obsah souboru s příponou `.src`. Dále je spuštěn skript `interpret.py` a jako zdrojová XML reprezentace `IPPcode19` je použit obsah dočasného souboru s příponou `.xml`. Vstup pro `interpret` je načten ze souboru s příponou `.in`. Návrátová hodnota skriptu se porovná s hodnotou v souboru s příponou `.rc` a výstup v dočasném souboru s příponou `.tmp` je porovnán funkcí `diff` se souborem `.out`.

Podle úspěšnosti či neúspěšnosti testu se volá funkce `generate_html_success` nebo `generate_html_failed`. HTML. Statistika obsahuje jméno testu, barevný indikátor úspěšnosti (SUCCESS/FAILED) a případný popis, při neúspěšném testu.

## Rozšíření

**FILES** – jsou procházeny postupně všechny adresáře ze seznamu případně spouštěny rovnou specifikované testy. Regulární výraz pro nalezení všech souborů `"/^(.+)/"` je případně nahrazen regulárním výrazem od uživatele.