

## Implementační dokumentace k 2. úloze do IPP 2019/2020

Jméno a příjmení: Damián Sova

Login: xsovad06

### Interpretácia XML reprezentácie kódu (interpret.py)

Úlohou skriptu je načítať XML reprezentáciu kódu a tento program interpretovať s využitím štandardného vstupu a výstupu.

Skript je možné spúšťať so vstupnými parametrami:

- `--help` (tento parameter sa nedá kombinovať s ostatnými parametrami)
- `--source=file` (súbor s XML reprezentáciou kódu)
- `--input=file` (súbor so vstupom pre samotnú interpretáciu)

Skript využíva hybridnú štruktúru. Väčšina funkčného kódu je uložená v triedach, v triednych metódach. Hlavné telo skriptu je však mimo triedy, či metódy. Tam sa deklarujú a inicializujú globálne premenné, potrebné na riadenie interpretácie.

### Globálne premenné

- `instruction_order` – určuje poradové číslo vykonávanej inštrukcie
- `value_stack` – zásobník na ukladanie hodnôt
- `stack_for_call` – zásobník pre inštrukcie CALL a RETURN

### Použité triedy

- `Errors` – obsahuje chybové kódy, volá sa vždy pri ukončovaní skriptu chybou
- `Xml_Parser` – spracovanie xml kódu a naplnenie poľa inštrukciami, všetko vykoná jediná metóda `parse()`
- `Frames` – práca s rámcami, GF(globálny rámec), LF(lokalný rámec), TF(dočasný rámec), obsahuje triedne premenné reprezentujúce jednotlivé rámce a triedne metódy na ich spravovanie: `add()`, `set()`, `get()` a `get_without_missing_value()`
- `Labels` – práca s návěstiami, obsahuje triednu premennú `Labels` a triedne metódy: `add_label()`, `jump()`
- `Stack` – reprezentuje abstraktný dátový typ (ADT) zásobník
- `Var` – reprezentuje premennú, obsahuje jej meno a typ rámcu
- `Symb` – fiktívna trieda reprezentujúca typy: `int`, `str`, `bool`, `nil`, `Var`
- `Instruction` – hlavná a najdôležitejšia trieda, slúži na interpretáciu zdrojového kódu, kontroluje typy argumentov, zabezpečuje ich nahratie a konvertuje ich podľa potreby, zabezpečuje implementáciu jednotlivých inštrukcií jazyka IPPcode20

Okrem spomínaných tried je interpret zložený aj z pomocných funkcií:

- `process_program_arguments()` – spracovanie vstupných parametrov skriptu
- `open_source_file()` – otvorenie a nahratie vstupného kódu súboru
- `labels_process()` – spracovanie návěstí pred interpretovaním kódu

### Riadenie interpretácie

- inicializácia globálnych premenných

- spracovanie vstupných parametrov skriptu
- nahranie vstupu buď zo súboru, ak je zadáný alebo zo štandardného vstupu
- spracovanie inštrukcii do poľa, kontrola štruktúry a syntaxe XML kódu
- spracovanie návestí pred vykonávaním interpretácie
- vykonávanie inštrukcií
- ukončenie skriptu 0, ak prebehlo všetko bez chyby