

Implementační dokumentace k 2. úloze do IPP 2019/2020

Jméno a příjmení: Mirka Kolaříková

Login: xkolar76

1 Interpret.py

1.1 Zadání

Cílem 2. úlohy bylo vytvořit skript v jazyce python3.8, který načte XML reprezentaci programu a tento program interpretuje a generuje výstup.

1.2 Implementace skriptu `interpret.py`

Skript nejdříve zpracuje argumenty z příkazové řádky, a to pomocí knihovny `argparse`. Ke zpracování vstupní XML reprezentace je použita knihovna `ElementTree`.

Poté, co je zkontrolována XML hlavička programu, se ve smyčce provádí lexikální a syntaktická analýza jednotlivých instrukcí pomocí funkce `is_valid_opcode()`. Ke kontrole platných názvů proměnných a návěští jsou použity regulérní výrazy a knihovna `re`. V případě, že je v dané instrukci vše správně, je tato instrukce přidána do globálního slovníku instrukcí. V této smyčce se také vytváří slovník všech definovaných návěští.

Ve druhé smyčce se volají funkce k vykonání jednotlivých instrukcí. Instrukce jsou vykonávány vzestupně podle pořadí `order`. Pro přístup k proměnným jsou využívány vlastní funkce `safe_var_val()` a `get_var_val()`. Při skokových instrukcích je využívána funkce `label_ok()`.

V případě nalezení jakékoliv chyby je celý skript ihned ukončen s odpovídajícím chybovým kódem.

1.3 Rozšíření

Ke skriptu `interpret.py` jsou implementována tři bonusová rozšíření.

1.3.1 FLOAT

Instrukce podporují i práci s typem `float`. Přidány jsou instrukce `DIV`, `INT2FLOAT` a `FLOAT2INT`.

1.3.2 STACK

Jsou přidány zásobníkové varianty instrukcí.

1.3.3 STATI

Na sbírání statistik o počtu vykonaných instrukcí je vytvořen čítač, který se inkrementuje vždy po vykonání instrukce.

Sbírání statistik o počtu inicializovaných proměnných se provádí ve funkci `safe_var_val()`. Před uložením hodnoty do proměnné se funkce podívá, zda tato proměnná již byla inicializovaná. Jestliže ne, čítač je inkrementován o jedna.

Statistiky jsou postupně podle zadaných vstupních parametrů zapsány do souboru `file`.