

Implementační dokumentace k 2. úloze do IPP 2021/2022

Jméno a příjmení: Dalibor Králík

Login: xkrali20

Interpret.py

Logika:

Interpret som sa rozhodol pokúsiť naprogramovať objektovo orientovane. Najskôr som si navrhol triedy, ktoré som predpokladal že budem potrebovať a následne som im naimplementoval ich atribúty a metódy. Vytvoril som nasledujúce triedy: **xmlReader**, **frame**, **symtableClass** a **interpret**.

Trieda **xmlReader** obsahovala metódy pre spracovanie vstupného XML inputu. Implementovala metódy ako **orderCheck** (kontrola validného poradia inštrukcií), **getOpcode** (zistenie operačného kódu inštrukcie zo XML), **isXMLCorrect** (kontrola nevalidnej XML štruktúry).

Trieda **frame** reprezentovala jednu inštanciu pamäťového rámca. Mala atribúty ako napríklad obsah rámca, typ rámca a veľkosť obsahu rámca.

Trieda **symtableClass** reprezentovala tabuľku symbolov – zásobník vytvorený z pamäťových rámcov. Tabuľka symbolov slúžila na ukladanie premenných (názov, scope, dátový typ, obsah), pamäťových rámcov a dát, ktoré boli pre interpretáciu nevyhnutné. Implementovala metódy ako **popFrame()**, **createFrame()**, **pushFrame()**, **findItem()**, **updateItem()** a **appendItem()**. Obsahovala tiež aj inštanciu dočasného pamäťového rámca, tzv. temporary frame.

Trieda **interpret** obsahovala a implementovala metódy, ktoré mali za úlohu využiť ostatné inštancie vymenovaných tried a za ich pomoci spracovať a interpretovať príkazy vstupného XML. Tieto príklady sa inštanciami triedy **interpret** predávali riadok po riadku a inštancia ich pomocou implementovaných metód spracovala a vypísala výstup na štandardný výstup (STDOUT). Tato trieda obsahovala metódy **isVariable()**, **isType()**, **isLabel()**, **isConstant()**, **isString()**, **isVarNone()**, **isSymb()**, **stringConversion()** a **instructionOperations()**.

Implementácia:

1. Spracovanie argumentov programu pomocou funkcie **arguments()**.
2. Tvorba potrebných tried.
3. Načítanie vstupného xml súboru.
4. Skontrolovanie validného poradia inštrukcií, načítanie labelov do poľa **arrayOfLabels**, prechádzanie vstupného XML a ukladanie inštrukcií do poľa **arrayOfInstructions**.
5. Spracovávanie inštrukcií podľa operačného kódu v metóde **instructionOperations()** triedy **interpret**.

Použité knižnice:

- Sys
- Re
- xml.etree.ElementTree

test.php

Implementácia:

1. Spracovanie a ošetrovanie argumentov programu.
2. Tvorba požadovaných súborov k každému test-case -> súbory s koncovkou .in, .out, .rc.
3. Testovanie jednotlivých test-cases. Podľa argumentov programu buď prebiehali testy len pre parse.php, interpret.py alebo pre oba naraz. Na základe toho sa testy spúšťali iným spôsobom. Na základe výsledkov sa zozbierali dáta, ktoré sa uložili do premenných.
4. Výpis výsledného HTML kódu, ktorý obsahoval dáta z premenných, v ktorých boli uložené výsledky testov. HTML kód som si napísal a uložil do premennej, ktorá sa na záver testovania vypísala.