

## Implementační dokumentace k 2.úloze do IPP2022/2023

Jméno a příjmení: Martin Priessnitz

Login: xpries01

### Implementace skriptu interpret.py

Skript *interpret.py* čte ze souboru nebo ze standardního vstupu. Výstupem je provedení samotného programu, případně vypsaní na standardní výstup. Chybová hlášení jsou vypisována na standardní výstup.

Struktura programu

Úlohu jsem řešil pomocí OOP. Vytvořil jsem si třídy na hlavní části programu (*class Interpret*, *class Program*, *class Instruction*, *class Argument*, *class Frame* a *class Memory*).

#### *Class Program*

*Class Program* definuje vnitřní reprezentaci načteného zdrojového kódu. Konstruktor vyžaduje kořenový element zdrojové XML struktury. Obsahuje jedinou proměnnou *self.ins*, což je seznam instrukcí programu. Provádí nezbytný preprocessing např.: kontrolu orderu a seřazení atd. . Parsování vnitřních XML elementů deleguje na třídu *Instruction*. Je to instance třídy *Instruction*.

#### *Class Instruction*

*Class Instruction* reprezentuje jednu instrukci ze zdrojového kódu. Ověřuje a načítá atributy elementu instruction. Parsuje vnitřní elementy a deleguje je na třídu *Argument*. Obsahuje *self.order*, *opcode*(všechni možné typy instrukcí) a *arg*(což je seznam argumentů). Je to instance třídy *Argument*

#### *Class Argument*

*Class Argument* kontroluje typy argumentů.

#### *Class Frame*

*Class Frame* reprezentuje jeden frame v paměti. Proměnné si ukládá formou slovníku, kde klíč je název proměnné a hodnota je hodnota proměnné.

#### *Class Memory*

*Class Memory* reprezentuje paměťový model program. Spolu s instancí třídy *Interpret* jednoznačně určuje aktuální stav výpočtu. Obsahuje všechny framy(*GF*, *TF* a zásobník *LF*).

#### *Class Interpret*

Top-level třída, která definuje vnější rozhraní interpretu. V konstruktoru projde program a načte všechna návěští. Interpretace se spouští skrz funkci *run()*.

