

## Implementačná dokumentácia k 2. úlohe do IPP 2022/2023

Meno a priezvisko: Matej Keznikl

Login: xkezni01

### Popis implementácie skriptu interpret.py:

Skript ako prvé skontroluje správnosť argumentov skriptu. V prípade, že bol zadán nesprávny argument alebo bolo zadané väčšie množstvo argumentov, skript vráti chybovú návratovú hodnotu 10, resp. 11 v prípade chybné kombinácie argumentov. Overenie argumentov skriptu má na starosti trieda *Arguments*. Po kontrole argumentov nasleduje spracovanie vstupu v podobe XML reprezentácie kódu. Spracovanie XML je vykonávané v triede *xml*, pričom je na spracovanie používaná knižnica *xml.etree.ElementTree*. Následne je spracované XML skontrolované na lexikálne a syntaktické chyby, ktoré mohli byť spôsobené užívateľom. V prípade negatívneho výskytu syntaktických a lexikálnych chýb je XML rozparšované na inštanície triedy *Instruction*, ktoré sú následne vložené do zoznamu inštrukcií. Zoznam inštrukcií je taktiež nutné zoradiť podľa argumentu *order* daných inštrukcií, čo je vykonané metódou triedy *xml*, *sort()*. Po skontrolovaní argumentov skriptu a spracovaní XML reprezentácie kódu nasledovala samotná interpretácia jednotlivých inštrukcií v rámci triedy *Interpret*. V rámci statických kontrol boli dopredu pozisťované návestia, ošetrovanie ich redefinície, pričom v prípade chyby skript vráti chybovú návratovú hodnotu 52. Interpretáciu inštrukcií vykonáva metóda *handleInstructions()* triedy *Interpret*, kde sa v rámci *while* cyklu interpretujú inštrukcie zo zoznamu *Instruction\_list* podľa premennej *instruction\_counter*, ktorá v prípade skoku skočí na inštrukciu za návěstím. Samotná interpretácia inštrukcie je volanie inštančnej metódy triedy *Instruction*, ktorá volá konkrétnu metódu (statickú alebo inštančnú) podľa typu inštrukcie, kde inštrukcie môžu byť rozdelené do tried *Arithmetic*, *Relational*, *Logical*, *String*, *ProgramFlow*, *IO*, *Stack* a *Frame*.

### Popis jednotlivých tried:

#### Trieda *Arguments*

Trieda, ktorá má na starosti spracovanie a overenie vstupných argumentov skriptu. Obsahuje metódy *Source* a *Input*, ktoré vracajú popisovače (file descriptors) vstupných súborov.

#### Trieda *Exit*

Trieda má na starosti ukončenie programu s korektným návratovým kódom interpreta a taktiež obsahuje všetky potrebné návratové kódy.

#### Trieda *XML*

Trieda, ktorej úlohou je spracovanie XML reprezentácie kódu zo vstupu. Obsahuje metódu *check*, ktorá skontroluje syntaktickú a lexikálnu korektnosť vstupnej XML reprezentácie kódu, metódu *arg\_count*, ktorá vracia počet argumentov jednotlivých inštrukcií a v prípade chybného počtu ukončí program s korektným návratovým kódom, metódu *parse*, ktorej úlohou je rozparšovanie inštrukcie a jej atribútov na inštanície tried *Instruction*, pričom je výsledná inštančia uložená do zoznamu inštrukcií *Instruction\_list* a metódu *sort*, pričom jej úlohou je zoradenie inštrukcií podľa argumentu *order*.

#### Trieda *Interpret*

Trieda má na starosti interpretáciu jednotlivých inštrukcií. Obsahuje metódu *handleInstructions* v rámci ktorej sa volajú jednotlivé inštrukcie určené na interpretáciu a metódu *handleLabels*, ktorej úlohou je spracovanie návěstí a v prípade redefinície návěstí ukončenie programu s korektným chybovým návratovým kódom.

#### Triedy obsahujúce metódy, ktoré implementujú jednotlivé inštrukcie:

- **Trieda *Arithmetic*** - aritmetické operácie
- **Trieda *Relational*** - relačné operácie
- **Trieda *Logical*** - logické operácie
- **Trieda *String*** - operácie nad reťazcami
- **Trieda *ProgramFlow*** - operácie pre zmenu toku programu
- **Trieda *IO*** - vstupno-výstupné operácie
- **Trieda *Stack*** - operácie nad dátovým zásobníkom
- **Trieda *Frame*** - operácie nad rámcami

**Trieda Var**

Trieda, ktorá reprezentuje premennú, tzn. obsahuje meno, rámec, typ a hodnotu premennej.

**Trieda Constant**

Trieda reprezentuje konštantu v zásobníku, tzn. obsahuje typ a hodnotu.

