

Implementační dokumentace k 2. úloze do IPP 2022/2023

Jméno a příjmení: Kateřina Čepelková

Login: xcepel03

Skript `interpret.py`, napsaný v jazyku `python3` překládá XML reprezentaci programu, napsaném ve zdrojovém kódu jazyka `IPPCODE23` s rozšířením `STACK` a částí `FLOAT`.

1 Parametry

Díky knihovně `argparse` může program načítat 3 druhy parametrů. `--help`, který vypíše nápovědu, `--source=file`, přes který je možné zasílat vstupní soubor s XML reprezentací a `--input=file`, který může nést soubor se vstupy pro instrukci `READ`.

2 Prvotní zpracování XML

Na zpracování a získání XML souboru je využita knihovna `xml.etree.ElementTree`.

Po vyhodnocení parametrů jsou náležitě načteny pomocné soubory (XML a soubor se vstupy). Pokud není jeden z parametrů se souborem zadán, je tato hodnota načtena ze standardního vstupu. Soubor se vstupy je načten a rozdělen do listu podle `'\n'` funkcí `split()` a to kvůli lehčímu parsování pro instrukci `READ`.

Po načtení celého XML souboru je soubor rozdělen na elementy a ty jsou kontrolovány částečně ve funkci `xml_check(root, source)` a částečně v `main`. V `xml_check()` je kontrolována hlavička souboru a atributy instrukcí – zda bylo správně zasláno pořadí a operační kód. Pokud jedna z těchto věcí odporuje očekávání je interpretace ukončena a je zaslán chybový kód 32.

Po vyhodnocení `xml_check()` následuje načítání instrukcí. To je řešeno přímo v `main`. Kontroluje se zda náhodou nebyla zadána neznámá instrukce, neznámý typ atributů instrukce či nepodporované pořadí (všechna záporná čísla a 0). Pokud instrukce projde všemi kontrolami, je uložena do instance třídy `Instruction` a vložena do seznamu tříd `instructions`. Tento seznam je následně seřazen podle pořadí instrukcí. Díky seřazení je následně i jednoduší najít duplikáty. Ke kontrole duplikátů je seznam instrukcí projit znovu a všechna pořadí jsou postupně zapsána do seznamu `orders`. Tento seznam instrukcí je pak převeden na seznam dvojic `order_instructions`, ve kterém se nachází všechny instance instrukcí s jejich pořadím v kódu.

Před předáním seznamu instrukcí interpretu pro finální kontrolu a interpretaci, je pro ulehčení vyhodnocování skoků vytvořen slovník `labels`, ve kterém se nachází všechna návěští s pořadovým číslem instrukce, kde je dané návěští definované. Tento slovník také provádí kontrolou proti duplikátům.

Po dokončení načítání instrukcí a vyhodnocování chyb vycházejících z předávání upraveného XML je vytvořena 1 instance třídy `Interpret` s názvem `interpret` a zavolána její metoda `run()`, přes kterou se následně řídí celá interpretace.

3 Interpretace instrukcí

V metodě `run()` třídy `Interpret` se nachází `while` cyklus, který kontroluje, zda seznam instrukcí pro běh `run_inst` (definován při inicializaci jako kopie `order_instructions` a přenastaven při každém skoku – uloží se do něj všechny instrukce které se nachází po daném návěští) není prázdný. Pokud ano, je interpret ukončen, jinak pokračuje. Po vstupu do `while` cyklu je vyjmuta instrukce z nulté pozice v seznamu `run_inst` a následně je nad jejím názvem (operačním kódem) zavolána funkce `getattr()` (`instr`), přes kterou je zavolána metoda pro příslušný operační kód s parametrem s danou instancí třídy `Instructions`, kde jsou uloženy atributy potřebné pro vyhodnocení.

V této metodě je instrukce vyhodnocená, ale prvně je zkontrolováno, zda sedí počet argumentů (přes funkci `arg_cnt_check(instr, num)` a jejich typy. Pokud některá z těchto podmínek není splněna je interpretace ukončena s příslušným exit kódem. Na získání hodnoty a typu z argumentu jsou využity metody třídy `Instance` `get_type_from_arg(instr, pos)` a `get_value_from_arg(instr, pos)`, které vrátí příslušnou

hodnotu jak z již deklarované proměnné tak přímo ze zaslaného argumentu. Pro získání proměnné na kterou se má ukládat určitá hodnota je využitý kód:

```
name_arg1 = inst.get_arg_value('arg1')
self.get_frame(name_arg1[:3])[ name_arg1[3:]] =
```

Prvně je načten název a rámec proměnné díky metodě třídy `Instruction` `get_arg_value(inst, 'arg1')` ve formátu `'FR@nazev'`. Následně metoda třídy `Interpret` `get_frame(name_arg1[:3])` navrátí žádaný rámec, díky tomu, že v prvních 3 znacích jména instrukce se nachází její rámec. V tomto rámci, které jsou definovány jako slovníky s prvky ve formátu `'název instrukce:(hodnota, typ)'`, je nalezena hodnota odpovídající klíči získaném z druhé části (od 3. znaku) názvu proměnné.

4 Třídy

4.1 Instruction

Třída pro ukládání instrukcí a jejich atributů. Ukládá pořadí, název a slovník argumentů `{arg1, arg2 arg3}` instrukce. Obsahuje metody `add_argument(arg_type, value, key)`, jenž vytvoří instanci třídy `Argument` a přidá ji do slovníku argumentů a `arg_cnt()`, který vrátí počet uložených argumentů. Metody `get_arg_type(pos)` a `get_arg_value(pos)` vrátí typ/hodnotu argumentu s určitým klíčem (`pos`) ve slovníku argumentů.

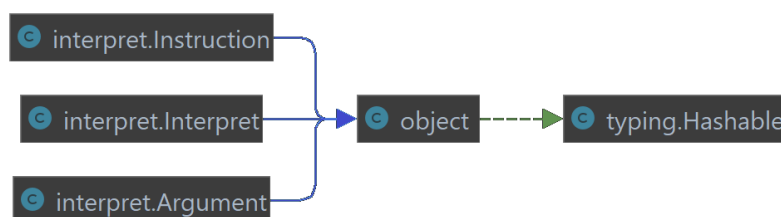
4.2 Argument

Třída pro ukládání atributů argumentů instrukce. Ukládá se v ní typ a hodnota každého argumentu. Stejně jako proměnné jsou všechny hodnoty ukládány jako string, kvůli lehčímu tisku, takže se pro využití v jiných instrukcích musí přetypovávat.

4.3 Interpret

Třída pro interpretaci instrukcí. Její atributy jsou seznam instrukcí, seznam instrukcí pro běh `run_inst`, seznam návěstí, vstupy souboru rozděleny na řádky. Dále se zde nachází počítadlo vykonaných instrukcí, zásobník volání pro instrukci `RETURN` `last_pos`, datový zásobník `stack` a všechny rámce, kde globální `GF` a dočasný `TF` jsou definované jako slovníky, s prvky ve formátu `'název instrukce:(hodnota, typ)'`, a v zásobníku pro lokální rámce `LF` jsou proto uloženy právě slovníky.

Metody tu jsou pojmenovány podle všech základních instrukcí (s určitými výjimkami, kterým bylo nutné přidat `_` před jméno kvůli redefinici) a i instrukcí z rozšíření `STACK` a `FLOAT`. Tyto metody podle instrukcí mají za úkol interpretovat. Dalšími pomocnými metodami jsou `get_frame(frame)`, která vrací hledaný rámec, `get_touple(var)`, který navrátí dvojici proměnné (hodnota, typ) a `get_value_from_arg(inst, arg_num)` a `get_type_from_arg(inst, arg_num)`, které slouží k navrácení žádané hodnoty ať už je zadaný argument proměnná, nebo přímo hodnota. Poslední metodou je `run()`, přes kterou je řízena interpretace a to s pomocí atributu pro běh `run_inst`,



5 UML diagram

