

Implementační dokumentace k 2. úloze do IPP 2020/2021

Jméno a příjmení: Peter Rúček

Login: xrucek00

Analýza požiadaviek

Úlohou bolo vytvoriť skript v jazyku Python 3.8, ktorý interpretuje inštrukcie jazyka IPPcode21 prevedené do XML formátu (pomocou parse.php).

Riešenie

Implementácia je rozdelená do dvoch významových častí:

1. Čítanie XML
2. Interpretácia

Na čítanie XML bola použitá knižnica `xml.etree.ElementTree`, ktorá XML súbor chápe ako strom, a tak sa aj pracuje v programe (root, child). Kontrolujú sa postupne všetky tagy a atribúty inštrukcie a pri lexikálnej a syntaktickej správnosti sa potom daná inštrukcia predá interpretu na interpretáciu. Táto časť sa vykonáva v súbore `interpret.py`.

Interpretácia sa vykonáva v súbore `exec.py`. Pre lepšiu prácu s abstrakciami boli vytvorené triedy `Interpret`, `Frames`, `Instruction`, `Variable`. `Instruction` reprezentuje inštrukciu a teda obsahuje opcode, order a prípadné argumenty. `Variable` reprezentuje premennú, teda jej názov, typ a hodnotu. `Frames` vytvárajú abstrakciu práce so všetkými rámcami (GF, LF, TF). Všetky 3 triedy by sme mohli nazvať iba ako pomocné.

Hlavnou triedou je `Interpret`, ktorý obsahuje už všetko potrebné na interpretáciu program. Obsahuje pole všetkých inštrukcií, zásobník, zásobník volania a rámce. V `interpret.py` sa volajú iba 2 funkcie tejto triedy: `add_instruction()` a `execute()`.

Rozparovaná inštrukcia na opcode, order a args sa pošle interpretu na spracovanie, `interpret.add_instruction(opcode, order, args)`. Vytvorí sa nová inštancia inštrukcie a daná inštrukcia sa uloží do poľa všetkých inštrukcií `instructions`.

Najdôležitejšou, najdlhšou a najkomplikovanejšou funkciou je `execute()`. Ktorá sa zavolá na konci `interpret.py` keď už sú všetky inštrukcie spracované. V tejto funkcii je cyklus ktorý prechádza cez každú inštrukciu v `instructions` a v obrovskej if-else vetve, kde sa podľa `instruction.opcode` rozhoduje ktorá inštrukcia a ako sa má vykonať. V každej elif vetve sa samozrejme vykonáva iná inštrukcia, ale spoločne sa dá povedať, že v každej inštrukcii sa vykonávajú rôzne kontroly, a v prípade chyby sa program ukončí chybným error kódom, v opačnom prípade sa inštrukcia vykoná. Veľa inštrukcií má podobné alebo také isté kontroly, ale vždy inú sémantiku. Najčastejšie sa kontroluje či je daný argument premenná, či daná premenná existuje a pod. Medzi zaujímavé inštrukcie by sme mohli zaradiť `CALL` a všetky druhy `JUMP` inštrukcií. Pri nich sa v poli `instructions` hľadá `LABEL` s daným názvom a zmení sa číslo cyklu (akoby zmena programového čítača).

Po zavolaní `interpret.execute()` sa začne program vykonávať a program sa ukončí s príslušným error kódom.

Použitie

```
python3.8 interpret.py --source=xml_file.xml --input=if_read_else_empty.in
```