

IMPLEMENTAČNÁ DOKUMENTÁCIA K 2. ÚLOHE IPP 2021/22

Meno a priezvisko: Denis Horil

Login: xhoril01

1. Interpret XML reprezentácie kódu

Interpretácia XML reprezentácie kódu v IPPcode22 je implementovaná v skripte `interpret.py`. Skript načíta zo štandardného vstupu zdrojový súbor v XML formáte a pomocou knižnice `xml.etree.ElementTree` je prevedený na „strom“ – koreňový element (`<program>`), potomok (`<instruction>`), potomok potomka/subelement (`<argx>`, kde x je číslo argumentu) a ich atribúty, iné štruktúry nie sú povolené.

Následne je strom utriedený najskôr podľa poradia argumentov jednotlivých inštrukcií (tj. tag subelementu) a potom vzostupne podľa poradia inštrukcií (`<order>`). V takto utriedenom strome sú nakoniec poradia inštrukcií zmenené na reálnu postupnosť čísel od 1 a celý strom inštrukcií je poslaný ako atribút do triedy `Opcodes`.

V triede `Opcodes` prebieha samotná sémantická analýza jednotlivých inštrukcií a ich argumentov.

2. Implementácia tried

V celom skripte `interpret.py` je implementovaných 5 rôznych tried:

- a) Trieda `Exitus`
 - implementovaná funkcia `err_raise()`, ktorá na základe chybového kódu vypíše chybové hlásenie a ukončí program s daným chybovým kódom.
- b) Trieda `Check`
 - funkcia `language()` skontroluje hlavičku XML súboru.
 - funkcia `string()` spracuje escape sekvencie v reťazci a vráti už upravený reťazec.
 - funkcia `symb()` vracia hodnotu a typ premennej zadanej v argumente `<symb>` - ak je zadaná premenná `<var>`, tak sa premenná vyhladá v príslušnom rámci a vráti jej hodnotu a typ.
 - funkcia `inFrame()` vyhladá premennú v zadanom rámci a vráti jej pozíciu v danom rámci, inak vracia -1.
 - funkcie `math()`, `compare()` a `logic()` spracúvajú aritmetické, relačné a logické inštrukcie.
 - funkcia `writeVar()` aktualizuje/zapíše typ a hodnotu (meno v prípade zápisu) konkrétnej premennej v danom rámci.
- c) Trieda `Stack`
 - implementuje jednoduchý zásobník s inštrukciami `Pop`, `Push`, `Top`, `IsEmpty` a `Size`.
- d) Trieda `Variable`
 - implementuje typ „Premenná“ s atribútmi názov, hodnota a typ. Tento typ je využívaný pri ukladaní premenných do rámcov a následnej práci s nimi.
- e) Trieda `Opcodes`
 - riadi celú interpretáciu
 - sú v nej implementované funkcie pre jednotlivé inštrukcie a k nim pomocné funkcie `letsStartIt()` a `labelListInit()`
 - funkcia `letsStartIt()` spúšťa interpretáciu
 - funkcia `labelListInit()` prechádza cez celý XML súbor a zapisuje do zoznamu `labelList` všetky definované návestia.

3. Pomocné funkcie

K triedam sú ďalej implementované pomocné funkcie `print_help()`, `argReturn()` a `funcStart()`:

- funkcia `print_help()` vypíše nápovedu skriptu.
- funkcia `argReturn()` vráti zoznam argumentov danej inštrukcie.
- funkcia `funcStart()` je implementovaná ako switch, ktorý vracia funkciu, ktorá sa má vykonať na základe názvu inštrukcie.

4. Testovací skript

Spúšťanie testov a ich vyhodnocovanie je implementované v skripte `test.php`. Skript načíta zo vstupu riadiace parametre a uloží ich do odpovedajúcich premenných. Následne sa spustia jednotlivé testy podľa nastavených parametrov. Spúšťanie testov a načítavanie vstupných parametrov je implementované vo funkcii `main()`.

Testy sa spúšťajú vo funkciách `parse_test()`, `int_test()` a `both()`:

- funkcia **`parse_test()`** vyhľadá testovacie súbory v zadanom adresári (alebo aj v podadresároch ak je nastavený parameter `--recursive`) a otvorí ich pre čítanie. Ak neexistuje jeden zo súborov s príponou `.out`, `.in` alebo `.rc`, tak sa vytvoria. Otváranie a prípadné vytváranie súborov zabezpečuje pomocná funkcia `valid_file()`. Následne sa vytvorí pomocný súbor, do ktorého sa uloží výsledná XML reprezentácia kódu vytvorená skriptom `parse.php`. Pomocný súbor sa ďalej v závislosti na riadiacich parametroch a návratovom kóde skriptu `parse.php` buď porovná nástrojom A7Soft JExamXML vo funkcii `jexam()`, alebo sa pošle do funkcie `both()`. Po spustení všetkých testov vracia funkcia späť do `main()` úspešnosť testov.
- funkcia **`both()`** spustí skript `interpret.py`, do parametru skriptu `--source` priradí pomocný súbor s XML reprezentáciou kódu, do parametru `--input` priradí súbor s príponou `.in` a výsledok interpretácie porovná s očakávaným výstupom v súbore s príponou `.out` nástrojom `diff` v prípade, že návratový kód skriptu sa rovná s očakávaným kódom v súbore s príponou `.rc` a tento kód je nulový. Výsledok porovnania (či už návratových kódov alebo výsledných súborov) vráti späť funkcii `parse_test()`.
- funkcia **`int_test()`** vyhľadá v zadanom adresári (prípadne v podadresároch) testovacie súbory s príponami `.src`, `.in`, `.out` a `.rc` (v prípade že `.out`, `.in` alebo `.rc` neexistujú, funkcia `valid_file()` ich vytvorí) a spustí skript `interpret.py`, do parametru `--source` priradí súbor s príponou `.src` a do parametru `--input` priradí súbor s príponou `.in`. Výsledok interpretácie sa uloží do pomocného súboru, návratový kód sa porovná s očakávaným kódom v súbore `.rc` a v prípade, že je tento kód nulový, tak sa pomocný súbor porovná nástrojom `diff` s očakávaným výsledkom v súbore `.out`. Po spustení všetkých testov vracia funkcia späť do `main()` úspešnosť testov.

Funkcie uvedené vyššie využívajú taktiež pomocné funkcie `jexam()` a `valid_file()`:

- funkcia **`jexam()`** porovná očakávaný výstup skriptu `parse.php` s aktuálnym výstupom nástrojom A7Soft JExamXML, ktorý je uložený v priečinku uvedenom riadiacim parametrom `-jexampath`.
- funkcia **`valid_file()`** skontroluje, či existuje súbor so zadaným názvom. Ak áno, tak ho otvorí pre čítanie, ak nie, tak ho vytvorí.

Generovanie dokumentácie zabezpečuje skript `classes.php` a v ňom trieda `Writer`. Skript `test.php` posiela do metódy triedy `Writer` pole názvov testov, pole úspešnosti testov, pole názvov priečinkov jednotlivých testov, počet úspešných testov a počet všetkých testov. Táto metóda vygeneruje jednoduchý kód v HTML 5, ktorý je poslaný na štandardný výstup.