

interpret.py

Skript napsaný v python3.8 interpretující XML reprezentaci programu napsaného v imperativním jazyce IPPcode20. Interpret umožňuje sběr statistických údajů interpretace (počet provedených instrukcí nebo maximální počet inicializovaných proměnných) a jejich výpis do souboru. Skript interpretuje jazyk IPPcode20 rozšířený o zásobníkové instrukce a práci s typem float zapsaným v hexadecimální formě.

Spuštění

```
python3.8 interpret.py [--input=file | --source=file] [parametry]
```

Při spuštění musí být zadán alespoň jeden z parametru, a to buď `--input`, nebo `--file`.

- **--help** Vypíše nápovědu.
- **--input=file** Specifikuje soubor s daty pro načítání instrukcí READ, pokud není zadáno, jsou data načítána ze standardního vstupu.
- **--source=file** Specifikuje soubor s XML reprezentací interpretovaného programu, Pokud není zadáno, jsou data načítána ze standardního vstupu.
- **--stats=file** Umožňuje provádět sběr statistik interpretace, specifikovaných statistickými přepínači, do zadaného souboru.
- **--insts** Statistický přepínač vypisující počet provedených instrukcí.
- **--vars** Statistický přepínač vypisující maximální počet inicializovaných proměnných ve všech rámcích.

Implementace

Interpret byl rozdělen pro lepší přehlednost do více modulů (`xml_parse.py`, `instr_tape.py`, `var_frame.py`, `program.py`) a hlavního skriptu obsluhujícího veškerý běh (`interpret.py`).

Po spuštění zkontroluje *interpret.py* parametry spuštění, zda vyhovují pravidlům a nastaví podle nich vnitřní hodnoty. Sběr statistických dat je zdlouhavý, a proto se provádí pouze na vyžádání. *interpret.py* pošle to *xml_parse.py* vstupní xml reprezentaci interpretovaného programu, který je zde zkontrolován na lexikální a syntaktickou správnost. Jednotlivé instrukce jsou vzestupně podle order uloženy v podobě tří adresného kódu do instrukční pásy, současně je vytvořen slovník návěští umožňující predikci skoků (sémantická kontrola skoků je prováděná až dynamicky při interpretaci). Naplněná páska je předána zpět *interpret.py*, který ji pošle na interpretaci do *program.py* zde se podle operačního kódu v instrukční pásce provedou příslušné instrukce, případně jsou sbírány statistická data. Práci s datovými rámci zajišťuje modul *var_frame.py*, který na požádání vrací, nebo nastavuje hodnotu proměnné v datovém rámci určeném předponou jména proměnné. Pro zpracovávání chybných případů (přístup k neexistující proměnné nebo rámci atd.) jsou vyvolávány výjimky, které jsou v *interpret.py*

zachytávány, vypisovány chybové hlášky a skript je ukončen s příslušným chybovým kódem. Při úspěšném ukončení interpretace se současným požadavkem na výpis statistik, vytvoří *interpret.py* výstupní soubor a na každý řádek vypíše statistický údaj podle parametrů spuštění (statistické parametry se můžou opakovat).

test.php

Skript napsaný v php7.4 provádějící automatické testování *interpret.py* nebo *parse.py*. Generuje přehlednou html stránku s výsledky testů, které jsou seskupovány podle složek.

Spuštění

```
php7.4 test.php [argumenty]
```

Nesmí se kombinovat požadavky na pouhou kontrolu parseru, nebo interpretu.

- **--help** Výpis nápovědy.
- **--directory=path** Složka s testy. Není-li uvedena je použita aktuální složka.
- **--recursive** Složky s testy budou prohledávány rekurzivně do hloubky.
- **--parse-script=file** Určí skript použitý pro parsování. Standartně *parse.php* v aktuální složce.
- **--int-script=file** Určí skript použitý pro interpretaci. Standartně *interpret.py* v aktuální složce.
- **--parse-only** Testován bude pouze parser.
- **--int-only** Testován bude pouze interpret.
- **--jexamxml=file** Soubor *jexamxml.jar* pro kontrolu xml výstupu parseru. Standartně je použit */pub/courses/ipp/jexamxml/jexamxml.jar*.
- **--testlist=file** Soubor obsahující na každém řádku složku s testy, nebo test. Nelze kombinovat s **--directory**.
- **--match=regexp** Regulární výraz upřesňující vybírané testy bez přípony.

Implementace

Po spuštění zkontroluje skript správnost použití parametrů. Na základě použití, respektive nepoužití **--testlist=file** se vyplní seznam jmen (cest) testů a seznam složek s testy. Na seznam složek s testy je aplikováno získávání testů, případně i rekurzivní a tyto testy jsou přidávány do seznamu se jmény (cesty). Každý test je zkontrolován, zda existují jeho doplňky s příponami *.out*, *.in* a *.rc*. Pokud některý z těchto souborů neexistuje, je vygenerován. V případě použití **--match=regexp** jsou ze seznamu vyřazeny ty testy, jejichž jméno bez přípony neodpovídá regulárnímu výrazu. Je spuštěn příslušný test. Výstupy parseru jsou v případě návratového kódu 0 testovány pomocí nástroje *jexamxml.jar*. Výstupy interpretu jsou testovány pomocí nástroje *diff*. V průběhu testů se generuje výstupní html stránka, ve které jsou seskupeny testy podle složek. Na závěr je do stránky vepsán souhrn všech testů.