

Implementačná dokumentácia k 2. úlohe z IPP 2020/2021

Meno a priezvisko: Richard Seipel

Login: xseipe00

1 Popis interpret.py

Spracovanie argumentov

Po načítaní je zdrojový súbor spracovaný funkciou `parse_source()` na objekt triedy `ElementTree`. Funkciou `validate_source()` sa skontroluje, či má spracovaný zdrojový súbor správnu štruktúru popisu jazyka IPPcode21, pomocou `XMLschema`. Po úspešných kontrolách sa inštrukcie v programe zoradia podľa ich argumentu `order` s pomocou funkcie `get_order()`. Následne sú tieto inštrukcie uložené do premennej `program`.

Interpretácia programu

Na začiatku interpretácie sú do objektu triedy `System` vložené inštrukcie z premennej `program`, dĺžka programu a taktiež dáta jeho vstupu rozdelené po riadkoch, ak boli dopredu zadané argumentom. Trieda `System` obsahuje objekty tried `Frames`, `DataStack`, `Stack`, `Program` a `Instruction`. Samotná interpretácia prebieha v cykle metódy `run_interpret()`. Ak je inštrukcia validná, sú jej operačný kód a argumenty načítané pomocou metódy `load()` triedy `Instruction` a uložené do atribútov objektu tejto triedy. Dáta argumentov inštrukcie sú pred uložením konvertované metódami triedy `ProgramData` a sú ukladané vo forme jej objektov s atribútmi `type` a `value`. Následne je spustená metóda `interpret_instruction()`, ktorá skontroluje operačný kód inštrukcie a spustí správnu funkciu pre jej interpretovanie.

Dátový model

Rámce sú definované v triede `Frames`, kde je každý rámec reprezentovaný slovníkom a lokálne rámce sú reprezentované zásobníkom slovníkov, pričom sa vždy pristupuje len k poslednému uloženému. Premenné sú do rámcov ukladané pod názvom premennej a jej ukladané dáta sú objektom triedy `ProgramData`. Dátový zásobník je definovaný v triede `DataStack`, kde sú dáta ukladané rovnako v objekte triedy `ProgramData`. Pri skokoch a volaniach sa poradie inštrukcie ukladá na zásobník volaní reprezentovaný triedou `Stack`.

2 Popis test.php

Spracovanie argumentov

Na ukladanie informácií o argumentoch slúžia asociatívne polia `$args` a `$arg_flags`, ktoré sú na začiatku naplnené predvolenými dátami a ďalej sú odovzdané do funkcie `parse_arguments()` pre spracovanie argumentov. Výsledné dáta sa odovzdajú do funkcie `run_testing()`.

Spustenie testovania

Vo funkcii `run_testing()` je skontrolovaná prípadná kolízia argumentov a existencia súborov zadanych v argumentoch. Taktiež je vytvorené pole `$output_data`, pre ukladanie dát o spustených testoch v jednotlivých prehľadávaných priečinkoch. Následne je spustené prehľadávanie testovaného priečinka.

Prehľadávanie priečinkov

Prehľadávanie prebieha pomocou funkcie `search_directory()`. Na prechod súborovým systémom je používaný objekt triedy `FileSystemIterator`. Taktiež je vytvorené asociatívne pole `$dir_data` do ktorého sú ukladané dáta o výsledkoch testovania v danom priečinku a na jeho konci je pole pridané do poľa týchto polí `$output_data`. Pri prehľadávaní skript prechádza obsah priečinka a ak narázi na podpriečinkov a rekurzívne prehľadávanie je zapnuté, spustí rekurzívne prehľadávanie aj pre tento podpriečinkov. Ak je nájdený súbor s príponou `src`, je spustený test tohto súboru.

Spustenie testu

Spustenie testu je realizované jednou z troch funkcií `run_both_test()`, `run_int_test()` alebo `run_parser_test()`. Testovanie využíva ostatné súbory v priečinku daného testu s príponami `in`, `out` a `rc` s rovnakým názvom ako súbor s príponou `src` alebo ich prípadne vytvorí. Ďalej zapíše výsledok testu spolu s ďalšími dátami do poľa `$dir_data`.

Výpis výstupu v HTML

Po prejdení a spustení všetkých testov, je funkciou `print_output()` vypísaný výsledný výstup testovania vo formáte HTML.