Implementačná dokumentácia k 2. úlohe do IPP 2020/2021

Meno a priezvisko: Marek Miček
Login: xmicek08

Skript interpret.py

Výsledný skript `interpret.py` využíva viacero modulov, pričom každý zohráva špecifickú úlohu pri výslednej interpretácii. V nekonečnej smyčke sa dotazujeme na už lexikálne a syntakticky skontrolované inštrukcie zo vstupného súboru s XML reprezentáciou zdrojového kódu. V závislosti na operačnom kóde inštrukcie voláme príslušné metódy z ostatných modulov, ktoré vykonajú interpretáciu danej inštrukcie

Modul `arguments.py`

Jeho hlavnou úlohou je spracovať argumenty príkazovej riadky, prípadne vypísať nápovedu pre užívateľa.

Modul `errors.py`

Vypisuje chybové hlášky na štandardný chybový výstup a ukončuje program s príslušným chybovým kódom.

Modul `xmlParser.py`

Zabezpečuje spracovanie vstupného súboru s XML reprezentáciou zdrojového kódu. Kontrolujeme tu lexikálnu a syntaktickú korektnosť každej inštrukcie a ukladáme si metadáta každej inštrukcie, či už operačný kód, poradie danej inštrukcie v rámci zdrojového súboru, typy a hodnoty operandov danej inštrukcie.

Modul `instructions.py`

Zabezpečuje uchovávanie metadát o každej inštrukcii, vrátane jej operandov. Taktiež pomocou viacerých metód tu zabezpečujeme kontrolu toku programu.

Modul `stacks.py`

Obsahuje dve triedy, pričom trieda `DataStack` zabezpečuje manipuláciu s dátovým zásobníkom a trieda `FrameStack` mimo iného aj manipuláciu so zásobníkmi rámcov. V jednotlivých triedach sú metódy implementujúce interpretáciu jednotlivých inštrukcií zo zdrojového kódu. Pokiaľ sú inštrukcie významovo podobné, tak jedna metóda môže implementovať interpretáciu viacerých inštrukcií (napr. metóda `relation_instruction` v triede `FrameStack`).

Skript test.php

Na začiatku spracujeme parametre príkazovej riadky pomocou stavanej funkcie `getopt`, pričom ešte pred tým inicializujeme pri vytváraní inštancie triedy `ArgumentParser` niekoľko atribútov, ktoré uchovávajú informácie kde hľadať testovacie skripty, testy, prípadne program `JExamXML` a jeho konfiguračné nastavenia. Takže pokiaľ užívateľ nezadá tieto informácie na vstupe, tak sa použijú práve tieto implicitné hodnoty.
Nasleduje hľadanie testov v zadanom/aktuálnom adresári. Pokiaľ narazíme na súbor s príponou `.src`, tak si extrahujeme meno a cestu k adresáru kde sa daný test nachádza a inicializujeme si dátové štruktúry pri daný adresár a test, do ktorých si budeme ukladať informácie o úspešnosti daných testov. V dátovej štruktúre pre adresár si ukladáme počet nájdených a prejdených testov a cestu k danému adresáru, keďže tieto informácie robia naše výstupné HTML čitateľnejšie. Taktiež si v poli `results` uchovávame výsledky nášho testovania pre každý nájdený test, konkrétne očakávaný (podľa hodnoty v súbore s príponou `.rc`) a reálny výsledok testu, výsledok porovnania výstupu daného skriptu a výsledok porovnania návratových kódov s referenčnými výsledkami. Výstupy testovaných skriptov ukladáme do dočasne vytvorených súborov pomenovaných po aktuálnom teste s prefixom `tmp_` pre výstup skriptu `parse.php` a `tmp2_` pre výstup skriptu `interpret.py`.

Výstupné HTML

Po spracovaní všetkých nájdených testov nasleduje generovanie výsledného HTML. Každý adresár v ktorom sme našli minimálne jeden testovací súbor je vo výstupnom HTML reprezentovaný tabuľkou, ktorá pre každý nájdený test v danom adresári zobrazuje jeho meno (stĺpec `Test name`), predpokladaný výsledok testu (stĺpec `EXPECTED`), reálny výsledok testu (stĺpec `REAL`), výsledok porovnania výstupu daného skriptu s referenčným výstupom (stĺpec `OUT`) a výsledok porovnania návratového kódu daného skriptu s referenčným návratovým kódom (stĺpec `RC`). Zároveň nad každou tabuľkou je hlavička s adresárom kde sa dané testy z tabuľky nachádzajú a štatistika nájdených a prejdených testov v danom adresári.