

Implementačná dokumentácia k 2. úlohe do IPP 2020/2021

Meno a priezvisko: Jakub Zaukolec

Login: xzauko00

1 Interpret XML reprezentácie kódu - interpret.py

1.1 Úloha a princíp skriptu

Úlohou tohto skriptu je interpretovať XML kód, ktorý je načítaný buď zo štandardného vstupu alebo pomocou parametra `--source`. Výstup (interpretácia) programu ide na štandardný výstup. Keď náhodou nastane chyba, tak sa vypíše hláška na štandardný chybový výstup a skript sa ukončí s danou chybovou návratovou hodnotou. Skript sa skladá zo štyroch modulov `argControll.py`, `structures.py`, `xmlParse.py` a hlavného modulu `interpret.py`.

1.2 Moduly skriptu

1.2.1 `argControll.py`

Tento modul sa stará o spracovanie parametrov skriptu. Skladá sa z jednej triedy `Arguments`, ktorá obsahuje pár atributov a jednu metódu `argCheck`, ktorá skontroluje správnosť parametrov. Pokiaľ je skript spustený s parametrami `--source` alebo `--input`, poprípade oboma, tak do atribútov si uloží cesty daných súborov.

1.2.2 `xmlParse.py`

Tento modul má za úlohu sparsovať vstupný XML kód. Skladá sa z jednej triedy `XMLParser`, ktorá obsahuje 5 metód. 4 metódy slúžia čisto na kontrolu správneho zápisu labelu, premennej, konštanty a typu. Hlavná metóda `xmlParse` ako prvé načíta XML kód a skontroluje jeho štruktúru, pokiaľ sedí štruktúra XML kódu (elementy: `instruction` a `arg`) tak sa prejde na syntaktickú a lexikálnu kontrolu. Popri prevádzaní syntaktickej a lexikálnej kontroly pridáva inštrukcie do slovníku inštrukcií.

1.2.3 `structures.py`

Tento modul obsahuje viacero tried, väčšina z nich slúži len ako štruktúra pre jednoduchšie spracovanie programu. Trieda `Instruction` reprezentuje štruktúru, ktorá sa skladá iba z mena inštrukcie a jej argumentov. Trieda `InstrDict` je niečo ako zoznam (slovník) všetkých inštrukcií, ktoré sa v programe nachádzajú, slovník je vytváraný pri syntaktickej a lexikálnej analýze XML kódu. Trieda `Variables` drží informácie a stará sa o prácu s premennými v daných rámcoch. Obsahuje metódy ako `defvar` pre definíciu premennej, `getTypeAndValue`, ktorá vráti typ a hodnotu premennej, poprípade konštanty a `setTypeAndValue`, ktorá nastavi typ a hodnotu premennej. Okrem týchto metód ešte obsahuje metódy pre prácu s dočasným rámcom. Trieda `Var` reprezentuje štruktúru pre premennú, drží jej typ a hodnotu. Poslednými triedami sú triedy `Stack` a `CallReturnStack`, sú implementované ako zásobníky. `Stack` je zásobník pre inštrukcie `pop` a `push` a `CallReturnStack` je zásobník pre inštrukcie `Call` a `Return`.

1.2.4 `interpret.py`

Hlavný modul programu, začína tým, že si volá metódu pre skontrolovanie argumentov a následne si zavolá metódu pre skontrolovanie vstupného XML kódu. Pokiaľ všetko prebehne bez chýb tak vstupujeme do hlavného `while` cyklu, ktorý beží, pokiaľ nevykonáme všetky inštrukcie. Tento `while` cyklus je rozvetvený na `vel` a `if else` vetvy, kde každá z nich sa stará práve o jednu inštrukciu, kontroluje jej sémantiku a vykonáva ju. Pokiaľ všetko prebehne bez chýb tak skript končí s návratovou hodnotou 0

2 Testovací rámec - test.php

2.1 Úloha a princíp skriptu

Úlohou tohto skriptu je otestovať správnosť a funkčnosť skriptov `parse.php` a `interpret.py`. Skript je navrhnutý objektovo orientovaným spôsobom a skladá sa z 3 hlavných tried: `Arguments`, `HTML_generator` a `Tests`.

2.2 Triedy skriptu

2.2.1 trieda `Arguments`

Táto trieda slúži ku sparsovaniu všetkých parametrov skriptu. Obsahuje atribúty, ktoré reprezentujú jednotlivé parametre skriptu, napríklad: `$directory`, ktorý drží cestu k súboru s testami, `$recursive`, ktorý drží booleanovskú hodnotu `true` alebo `false` podľa toho či bol zadán parameter `--recursive` a ďalšie atribúty, týkajúce sa všetkých parametrov. Ďalej trieda obsahuje metódu `parse_arguments`, ktorá sa stará o spracovanie parametrov pomocou `getopt` a taktiež, obsahuje metódu `check_paths`, ktorá kontroluje validitu zadáných ciest k súborom. Ak nastane chyba pri spracovaní parametrov, skript končí s návratovou hodnotou 10 a pokiaľ nastane pri kontrole ciest k súborom, tak skript končí s návratovou hodnotou 41.

2.2.2 trieda `HTML_generator`

Trieda sa stará o výpis HTML kódu na štandardný výstup. Obsahuje tri základné metódy. Metóda `generate_head`, ktorá sa stará o vygenerovanie HTML hlavičky, metóda `generate_tail`, ktorá generuje chvost (ukončenie) HTML kódu a metóda `generate_test`, ktorá je volaná po každom prevedení testu. Generuje jeden riadok tabuľky v ktorom sa nachádzajú informácie o poradí testu, ceste k testu, mena testu a o jeho úspešnosti.

2.2.3 trieda `Tests`

Je to hlavná trieda, ktorá sa stará o samotné prevádzanie jednotlivých testov. Táto trieda dedí z triedy `Arguments` aby mala prístup ku všetkým atribútom triedy `Arguments` a nemuseli by byť predávané parametrami. Skladá sa z dvoch hlavných metód. Metóda `start_tests` sa stará o prechádzanie jednotlivých súborov. Funguje na princípe zásobníka. Do zásobníka je na začiatku pridaný hlavný adresár s testami, ktorý bol zadán ako parameter skriptu. Telo sa ďalej skladá z `while` cyklu, ktorý beží pokiaľ tento zásobník není prázdny. Ďalej je v ňom ešte jeden `while` cyklus, ktorý prechádza jednotlivé súbory v aktuálnom adresári. Ak sa náhodou jedná o priečinok a pri spustení bol zadán parameter `--recursive`, tak tento priečinok pridá do zásobníka priečinkov. Ak sa jedná o súbor s príponou `src`, tak sa skontroluje prítomnosť súborov s príponami: `in`, `rc` a `out` a poprípadе sa dogenerujú a volá sa metóda `exec_test`.

Metóda `exec_test` sa stará o prevdenie testu. Skladá sa z troch `if else` vetiev, podľa toho či sa testuje iba `parser` alebo iba `interpret` alebo oboje. Vykoná test, porovná správnosť návratovej hodnoty a poprípadе vykoná aj `diff` alebo XML porovnávanie a zavolá metódu `generate_test`, ktorá sa postará o HTML generáciu.