

Test.php

1 Spracúvanie zadaných parametrov pri spustení skriptu

Po samotnom spustení skriptu `test.php` dôjde k inicializácii všetkých premenných (reprezentujúcich parametre testu) na predvolené hodnoty uvedené v zadaní. Následne sa vykoná kontrola prítomnosti parametru `--help` a taktiež použitie iných parametrov v kombinácii s týmto parametrom. V prípade, ak bol použitý iba samotný parameter `--help` vypíše sa užívateľovi nápoveda skriptu na štandardný výstup a vykonávanie programu bude ukončené s návratovou hodnotou 0.

Následne dochádza k prípadnému načítaniu všetkých vstupných parametrov skriptu a uloženie ich hodnôt do prislúchajúcich premenných. Nasleduje kontrola použitia nepovolených kombinácií parametrov ako napríklad parameter `--parse-only` s parametrom `--int-only` a kontrola dostupnosti jednotlivých súborov.

Po úspešných kontrolách sa vygenerujú základné parametre skriptovacieho jazyka HTML na štandardný výstup. V prípade, ak bol parameter `--recursive` zadaný, dôjde k zisteniu všetkých rekurzívnych priečinkov v priečinku so samotnými testami (zadaný vstupným parametrom skriptu `--directory`, v opačnom prípade aktuálny priečinok). Nasleduje samotné spúšťanie testovaných skriptov `parse.php` a `interpret.py` (prípadne ich kombinácia) pre každý testovací vstup.

2 Spúšťanie samotných testov

Pred samotným spustením požadovaných skriptov dôjde ku kontrole existencie súborov potrebných k vykonaniu testu. Ak niektorý zo súborov chýba (prípoma `in`, `out` alebo `rc`) dôjde ich vytvoreniu (v prípade súboru so správnym návratovým kódom testu (súbor `file.rc`) bude tento súbor vytvorený s predvolenou hodnotou 0).

Ak užívateľ požaduje iba otestovanie analyzátora (súbor `parse.php`) dôjde k spusteniu tohto skriptu. Následne sa porovná návratová hodnota s referenčnou (uvedená v testovom súbore `test.rc`). Ak je návratová hodnota rovná 0, porovná sa výstup spusteného testu s referenčným výstupom (testovací súbor `test.out`) pomocou nástroja `JExaML` a zapíše sa do tabuľky úspech/neúspech testu spolu s ďalšími údajmi testu. Veľmi podobný postup testovania je aj pri otestovaní iba samotného interpretu (súbor `interpret.py`). Jediným rozdielom je využitie unixového nástroja `diff` namiesto `JExaML` pri návratovom kóde testu 0.

Ak užívateľ požaduje testovanie oboch skriptov dochádza k prvotnému spusteniu analyzátora, zapísanie výstupu do dočasného súboru, ktorý sa ďalej predá na vstup skriptu `interpret.py`. Ten následne porovná výstup rovnako, ako aj pri spustení iba samotného interpretu a zapíše výsledok do HTML tabuľky na štandardný výstup.

Záverom dochádza k vypísaniu štatistiky úspešnosti testov a použitých parametrov pri spúšťaní samotného testu a ukončenie vykonávania programu s návratovou hodnotou 0.

Interpret.py

1 Spracúvanie zadaných parametrov

Na spracúvanie jednotlivých argumentov je použitá knihovňa „argparse“. Ak cesta súboru nebola načítaná, program načíta súbor pomocou štandardného vstupu.

Následne sa celý XML vstupný súbor uloží do dátovej štruktúry stromu (knihovňa `xml.etree.ElementTree`). Postupným prechádzaním celej tejto štruktúry sa pristupuje k jednotlivým položkám načítaného súboru a overuje ich správnosť použitia.

Komentár: Bohužiaľ viac toho nemám naimplementované v `interpret.py`