Implementační dokumentace k 2. úloze do IPP 2018/2019
Jméno a příjmení: Jozef Vanický
Login: xvanic09

Popis implementace skriptu test.php

Skript povoluje šest volitelných vstupních parametrů, přičemž je současně možné použít maximálně kombinaci čtyř parametrů. Při porušení této podmínky se skript ukončí s chybou 10. Přepínač „--help“ slouží na výpis nápovědy skriptu, která obsahuje informace o vstupních parametrech. Tento přepínač není možné kombinovat s žádným dalším vstupním argumentem. Zakázanou kombinací vstupních argumentů je dvojice *„--parse-only“ a „--int-only“, dále „--parse-only“ a „--int-script“ a nakonec dvojice „--int-only“ a „--parse-script“*. Při použití jedné z těchto kombinací se skript ukončí s chybou 10. V případě použití neexistujícího argumentu se skript ukončí s chybovým kódem 10.

Skript slouží na spuštění osobních testů pro skripty parse.php a interpret.py, jejichž výsledek vypíše v HTML reprezentaci na standardní výstup. Pokud při spuštění skriptu nebyl zadaný alespoň jeden ze vstupních argumentů *„--parse-script“* nebo *„--int-script“*, jsou očekávány výchozí názvy skriptů, tedy parse.php a interpret.py. Jestliže v zadané složce nebudou dané skripty nalezeny, skript se ukončí s chybou 11.

Když byl zadaný přepínač „--directory=“ obsahující cestu ke složce, bude tato cesta použita na načítání testů. V případě chybné cesty se skript ukončí s chybou 11. Při zadaném přepínači „--recursive“ se rekurzivně prohledají podsložky obsahující další testy.

V případě že uživatel použije přepínač „--parse-only“ nebo „--int-only“, zadá tím skriptu informaci, který z dvou skriptů parse.php nebo interpret.py se bude testovat, druhý skript testovaný nebude.

Na zpracování vstupních argumentů je využita funkce `getopt()` a pole možných povolených argumentů. Po zpracování vstupních argumentů je za využití funkce `generateHeader()` na standardní výstup vygenerována reprezentace HTML hlavičky výstupu a začnou se prohledávat soubory s testy. Pokud u testů některý ze souborů typu .in a .rc není, skript ho dokáže vygenerovat. V případě neúspěchu otevření skriptu či neúspěchu vytvoření souboru, se skript ukončí s chybou 11. Jestliže otevření testovacího souboru typu .out skončí neúspěchem, například nedostatečné oprávnění, skript bude ukončený s chybou 12. Během testování se porovnají návratové kódy v testovacím souboru .rc s návratovým kódem daného testovaného skriptu pro daný test. Stejně se testují taky rozdíly XML reprezentace se souborem .out, pokud se testuje pouze skript parse.php. Výsledky jednotlivých testů se postupně vypisují na standardní výstup. Nakonec je na standardní výstup vypsáno celkové shrnutí testů. Toto shrnutí obsahuje počet spuštěných testů, počet úspěšných a neúspěšných testů a procentuální úspešnost. Všechny tyto informace jsou vypsány formou HTML reprezentace.

Popis implementace skriptu interpret.py

Skript povoluje tři vstupní parametry a to „--help“, „--source=“ a „--input=“, přičemž musí být zadaný aspoň jeden z argumentů „--source=“ a „--input=“ jinak se skript ukončí s chybou 10. Přepínač „--help“ slouží na výpis nápovědy skriptu a informací o vstupních parametrech. Tento přepínač není možné kombinovat s žádným dalším vstupním argumentem. V případě, že jeden z argumentů „--source=“ nebo „--input=“ není zadaný, jeho vstup bude načítaný ze standardního vstupu. Po zpracování vstupních argumentů se načítá XML reprezentace kódu napsaného v IPPcode19. V případě neúspěchu se skript ukončí s chybou 11. Pokud porušuje podmínky „Well formated XML“ tak se skript ukončí s chybou 31. Pokud je porušená struktura XML, skript bude ukončený s chybou 32.

Každá instrukce, proměnná a rámec jsou reprezentovány jako třída. Po inicializace interních struktur (zásobník rámců, zásobník volání, apod.), se začne vykonávat intepretace jednotlivých instrukcí. Pokud je instrukce nevalidní, je skript ukončený s chybou 53. V případě sémantické chyby je skript ukončený s kódem 52. Jestliže hodnota chybí je skript ukončený s chybou 56, v opačném případě je zpracována. Jestliže je hodnota operandu nesouhlasí s jeho charakteristikou bude skript ukončený s chybou 57 a jestliže je řetězec nevalidní, skript se ukončí s chybou 58.

Během zpracování instrukcí se kontroluje sémantika argumentů instrukce. V případě zpracování symbolu je využita metoda checksymb(argtype, value), která identifikuje typ symbolu a jeho hodnotu. Instrukce jsou zpracovávány v cykle. Celá intepretace probíhá v několik cyklech, v prvním cyklu dojde ke kontrole lexikální a syntaktické analýzy, v druhém cyklu k načtení návěstí a jejich pozice v kódu a v posledním cyklu dochází k samotnému vykonávání instrukcí.