

Implementační dokumentace k 2. úloze do IPP 2020/2021

Jméno a příjmení: Lukáš Baštýř

Login: xbasty00

interpret.py

Počáteční část interpretu je velice podobná parser.php. Jako první se zpracují argumenty. Podle nich se následně nastaví `source` a/nebo `input` jako soubor/standardní vstup. U souboru se provede kontrola oprávnění a jestli soubor existuje. Poté se pomocí modulu `xml.etree.ElementTree`¹ načte XML soubor/vstup `source`. Modul při načítání zároveň provádí kontrolu, jestli se jedná o dobře formátované (anglicky *well-formed*²) XML.

Následuje první průchod. Zde se pomocí cyklu projdou jednotlivé elementy XML. Jako první se kontroluje kořenový element, jestli obsahuje správné jméno (štítek, angl. *tag*) a atributy. Následně se kontroluje pořadí instrukcí (duplicitní číslování) a atributy jejich elementů. Dalším vnořeným cyklem se poté kontrolují argumenty/operandy instrukcí (počet, pořadí, atributy). Zároveň se v prvním průchodu ukládají a kontrolují návěští (je možné skákat dopředu i dozadu). Jako poslední se instrukce s argumenty/operandy uloží do listu, který je následně seřazen podle jejich atributu pořadí (*order*). To nemusí odpovídat pořadí zápisu v XML.

V druhém průchodu se již odehrává samotná interpretace. Zde jsem chtěl použít přepínač (angl. *switch*), který by obsahoval jednotlivé skupiny operačních kódů jako možnosti. Vzhledem k tomu, že ho Python3.8 ale nepodporuje, udělal jsem místo toho jednu funkci obsahující podmínění příkazy `if` pro každou skupinu operačních kódů. Při provádění jednotlivých instrukcí se prakticky pokaždé kontroluje počet argumentů (0 - 3), jejich správný typ (řetězec, celé číslo, boolean, nil). V případě proměnné se také kontroluje, jestli již byla definována, případně jestli již má nastavený nějaký typ (nutní například pro `SETCHAR`).

Vzhledem k tomu, že data z XML nemusí být správná (např. špatný typ konstanty), je potřeba kontrolovat všechny typy (jak u argumentů/operandů v XML, tak na vstupu). To zde probíhá prakticky stejně jako tomu bylo v `parse.php` a to převážně pomocí regulárních výrazů ve funkci `get_var_const`. Na rozdíl od `parse.php` se zde ale také rovnou převádí celá čísla z řetězce na číslo ('42' -> 42). V pomocné funkci `check_string` také dojde k přepsání *escape sekvencí* v řetězcích za znak, který reprezentují.

Pokud se při procházení listu nenarazí na žádnou chybu je program postupně interpretován a výstup se vypisuje na standardní výstup. V případě chyby se vypíše zpráva na standardní chybový výstup a program se ukončí s odpovídajícím návratovým kódem.

¹<https://docs.python.org/3/library/xml.etree.elementtree.html>

²<https://www.w3resource.com/xml/well-formed.php>

test.php

`Test.php` je velice jednoduchý testovací skript na otestování výstupu a návratových hodnot u `parse.php` a `interpret.py`. Zprvu se načtou argumenty a podle jejich kombinace vyhodnotí typ testu, zda je potřeba použít program `JExamXML`³ a lokace všech souborů (`parse.php`, `interpret.py`, `JexamXML`, soubor s nastavením pro `JexamXML` a složka testů). Aby se předešlo problém s cestou k souborům, jsou všechny cesty převedeny na absolutní cesty (obsahují kořenový adresář `\`). Pokud se zde narazí na nějakou chybu, program se ukončí s odpovídajícím návratovým kódem.

Test samotný probíhá následovně. Nejdříve se vytvoří prázdný list/struktura `html.table`, kam se budou ukládat výsledky. Podle typu testu (oba, pouze `interpret`, pouze analyzátor) se následně vybere způsob porovnání výsledků. Jedná-li se pouze o testy pro **analyzátor**, je jeho výstup kontrolován s referenčním výstupem pomocí programu `JExamXML` a návratový kód s referenčním návratovým kódem. Pokud má otestovat pouze **interpret**, porovná se jeho výstup s referenčním pomocí programu `diff` a návratový kód s referenčním návratovým kódem. Má-li se testovat jak **analyzátor**, tak **interpret**, spustí se nejdříve **analyzátor** na vstupním souboru. Jeho výstup se následně předá **interpretu** (jeho výstupní kód nás zde nezajímá). Výstup a návratový kód **interpretu** je porovnán stejně, jako kdyby se jednalo pouze o test pro **interpret**.

Postup testů:

- Návratový kód je stejný, ale jiný než nula → test byl úspěšně dokončen a není potřeba kontrolovat výstup.
- Návratový kód je stejný a je nula → musíme zkontrolovat výstup. Pokud je stejný, test byl úspěšný (v opačném případě byl neúspěšný).
- Návratové kódy jsou různé → test se vyhodnotí jako neúspěšný a není potřeba kontrolovat výstup.

Každý test, jeho pořadí a výsledek je poté uložen do `html.table`. Po dokončení všech testů se zavolá funkce `print_html`, která vytiskne celý html soubor s výsledkami testů na standardní výstup.

³program na porovnání XML souborů: <http://www.a7soft.com/jexamxml.html>