

Implementační dokumentace k 2. úloze do IPP 2019/2020  
Jméno a příjmení: Daniel Pohančaník  
Login: xpohan03

## Inerpret.py

Ako prvé sa interpret snaží spracovať poskytnutý XML súbor. Pre spracovanie XML súboru je použitá knižnica `elementTree`. Súbor sa spracováva až kým z neho nedostaneme list reťazcov ktoré reprezentujú elementy XML súboru. Nad týmto listom sú potom vykonávané kontroly aby sa zamedzilo výskytu neexistujúcich alebo neplatných atribútov. Jednotlivé inštrukcie sú potom spracované a sú zostavené objekty reprezentujúce inštrukcie. Tento objekt v sebe drží informácie o jeho poradí, operačnom kóde a argumentoch. Argumenty sú taktiež reprezentované triedou aby bolo možné nad nimi vykonávať kontroly a boli schopné uchovať požadované informácie. Zoznam jednotlivých objektov argumentov je súčasťou objektu inštrukcie. Tieto objekty sú potom zoskupené do zoznamu. Nad týmto zoznamom sa následne iteruje a vykonáva sa kód, ktorý prislúcha operačnému kódu inštrukcie.

## Pamäťový model

Premenné sú zoskupené do takzvaných rámcov. Rámce sú tri. Lokálny, Globálny a Dočasný. Globálny rámec je inicializovaný od samého začiatku a je možné do neho okamžite vkladať premenné. Lokálny rámec existuje iba potom ako bol na zásobník rámcov vložený dočasný rámec. Dočasný rámec sa po vložení na zásobník premení na lokálny rámec a prekryje lokálne rámce, ktoré sa tam už nachádzajú. K premenným, ktoré sú uložené v prekrytých rámcoch sa nie je možné dostať. Do dočasného rámca je možné uložiť lokálny rámec z vrchu zásobníku. Tento lokálny rámec bude z vrchu odstránený a bude prístupný len cez dočasný rámec.

Premenné ako také sú v rámci interpretu reprezentované ako dátový typ slovník (dictionary) jazyka Python. Rámec je tým pádom zoznam týchto slovníkov. V slovníku premennej sú uchovávané informácie od názve, type a hodnoty premennej. Každá premenná je na začiatku definovaná hodnotou „unknown“ a typom „uninint“.

## Test.php

Test.php je testovacím skriptom pre interpret a parser. Skript najprv spracuje všetky predané parametre a vyhodnotí cesty k zdrojovým súborom. Ihneď potom skript zozbiera názvy všetkých priečinkov a súborov v zadanom adresári vrátane podadresárov ak sa jedná o rekurzívny režim. Následne skript hľadá všetky súbory s príponou .src. Ich názov si uloží a odstráni z neho príponu súboru pre jednoduchšie vyhľadanie ostatných súborov potrebných k testovaniu. V prípade súborov .out, .rc a .in si ich skript v prípade ich neprítomnosti vytvorí a naplní nulou v prípade .rc súboru a prázdny reťazcom v prípade súborov .out a .in.

Skript následne iteruje cez všetky zdrojové súbory testov, spúšťa ich a ukladá ich návratovú hodnotu a výstup. V prípade nastaveného príznaku „int-only“ sa testuje iba interpret a v prípade príznaku „parse-only“ sa testuje iba parser. Tieto 2 príznaky nemôžu byť nastavené súčasne. Ak nie je nastavený ani jeden tak sa zdrojový kód v jazyku IPPcode20 preloží priloženým interpretom do medzijazyka XML. Kód v tomto jazyku je následne interpretovaný priloženým interpretom.

Ak počas spracovania parserom nastane dôjde k návratovej hodnote rôznej od nuly tak sa táto hodnota priamo porovná s referenčnou návratovou hodnotou uloženou v súbore .rc. Pri jej zhode je test vyhodnotený ako úspešný a pri nezhode ako neúspešný. V týchto prípadoch už nedochádza k interpretácii jazyka XML pretože test zlyhal už vo fáze spracovania zdrojového kódu.

Ak je návratová hodnota parseru nula, tak sa začína s interpretovaním. Výstup je následne rovnakým spôsobom porovnaný s referenčným výstupom a referenčnou návratovou hodnotou. Pre tento účel si skript vytvára dočasné súbory ktoré po vykonaní vymaže.

V prípade režimu „parse-only“ sa na porovnanie použije nástroj JExamXML. Inak je použitý unixový nástroj diff.

Po vykonaní testov je vygenerovaný prehľadný HTML súbor obsahujúci výsledky všetkých testov. Testy sú zoradené podľa adresárov v ktorých sa nachádzajú.