

## Implementačná dokumentácia k 2. úlohe do IPP 2018/2019

Meno a priezvisko: Adrián Boros

Login: xboros03

### 1. Skript `interpret.py`

Nasledujúca časť popisuje spôsob implementácie skriptu `interpret.py`. Tento skript má za úlohu načítať XML reprezentáciu programu a interpretovať ho.

#### 1.1. Implementácia

Na spracovanie argumentov bola použitá funkcia `getopts` a na spracovanie vstupného súboru knižnica `XMLElementTree`. Skript bol navrhnutý objektovo-orientovane a skladá sa z dvoch tried: `Interpret` a `Variable`. Trieda `Variable` slúži na ukladanie údajov o jednotlivých premenných a to: meno, typ premennej a hodnota. V triede `Interpret` sa vykonáva samotná interpretácia programu.

Pred interpretáciou sa celý vstupný súbor prechádza kvôli kontrole či sa v ňom nenachádzajú nepovolené znaky ktoré by `XMLElementTree` neodhalil, ďalej pri tomto priechode sa do zoznamu ukladajú mená návěstí aby pri inštrukciách skoku bolo možné tento skok previesť a skontrolovať či dané návěstie existuje.

Následná implementácia je už založená na postupnom prechádzaní jednotlivých XML elementov vo `while` cykle a vykonávaní akcií podľa typu spracovanej inštrukcie. Dôvod, prečo sa použil cyklus `while` je ten, že pri inštrukciách skoku sa vracia číslo inštrukcie na ktoré sa má skočiť. Potom sa počítadlo v cykle nastaví na vrátenú hodnotu a tým je zaručené pokračovanie spracovania inštrukcií na mieste kam sa malo skočiť.

Pri jednotlivých inštrukciách sa postupne kontroluje najprv syntaktická správnosť a následne aj sémantická. Na kontrolu syntaxe boli použité regulárne výrazy zo skriptu `parse.php`. Pre ukladanie premenných do rámcov boli vytvorené slovníky ktoré obsahujú rovnaké informácie ako trieda `Variable`. Takisto bol vytvorený zoznam reprezentujúci dátový zásobník pre inštrukcie `PUSHS` a `POPS`. Jedným z problémom bolo konvertovanie escape sekvencií na odpovedajúce znaky, kde problematickým prípadom bolo to, keď premenná typu string obsahovala v sebe ako text číslo bez znaku `/`. Riešením bolo použitie funkcie `re.sub`, ktorá nahradí danú escape sekvenciu na odpovedajúci znak ak sa tam nachádza `/` a za tým 3-miestne číslo.

Ďalším problémom bolo zistiť návratové kódy skriptu pri chybových stavoch. Tento problém sa ale vyriešil implementáciou skriptu `test.php` a následným otestovaním skriptu `interpret.py`.

### 2. Skript `test.php`

Skript `test.php` slúži pre automatické testovanie skriptov `interpret.py` a `parse.php`.

#### 2.1. Implementácia

Spracovanie argumentov prebieha pomocou funkcie `getopt`. Po spracovaní sa skontroluje či nedochádza k nepovoleným kombináciám parametrov ako napr. súčasné nastavenie `int-only` a `parse-only`. Ak nie je nastavený argument `directory`, tak pomocou funkcie `getcwd` sa zistí cesta k aktuálnemu adresáru.

Skript potom iteratívne prechádza zadaný alebo aktuálny adresár pomocou funkcie `scandir` a v prípade zadania argumentu `recursive` aj jeho podadresáre. Všetky názvy súborov sa ukladajú do pola. Následne sa cyklom `foreach` prechádza každý prvok v poli, pomocou `pathinfo` sa zistí prípona súboru a ak sa táto prípona rovná `.src` tak sa volá funkcia na testovanie.

Vo funkcii `test` sa najprv zistí či existujú testovacie súbory s príponami `.in`, `.rc` a `.out`, ak nie tak sa vytvoria príkazom `touch`. V prípade vytvorenia súboru `.rc` sa doňho umiestni hodnota "0". Pred samotným testovaním sa vytvoria pomocné súbory `tmp_parse` a `tmp_inter_out`, do ktorých sa umiestnia výstupy z `parse.php` a `interpret.py`.

Jednotlivé testy prebiehajú podľa toho či sú nastavené parametry `int-only` alebo `parse-only`. V prípade že nie je zadaný ani jeden z týchto argumentov, tak je súbor s príponou `.src` poslaný ako vstup pre `parse.php`, výstup sa umiestni do dočasného súboru `tmp_parse` a následne obsah tohto súboru je presmerovaný na vstup `interpret.py`. Jednotlivé skripty sa spúšťajú príkazom `exec`.

V prípade že je zadáný argument `int-only` tak sa na vstup skriptu `interpret.py` presmerováva súbor s príponou `.src`. Ak je zadáný argument `parse-only`, vstupom je súbor `.src`.

Po prebehnutí testov sa najprv porovnajú návratové kódy. Ak sa nezhodujú, test je považovaný za neúspešný. Ak sa zhodujú, pri `parse-only` sa pomocou nástroja `JExamXML` skontroluje obsah súboru `.out` s obsahom dočasného súboru `tmp_parse`. Ak sa zhodujú, test je považovaný za úspešný. Pri `int-only` sa obsahy súborov kontrolujú príkazom `diff`. Po prebehnutí testov sa dočasné súbory odstránia.

Výpis výsledov na štandardný výstup prebieha príkazom `echo`, kde sa vypíše ID testu, názov testu, očakávaný a aktuálny návratový kód.