

## Implementačná dokumentácia k 2. úlohe do predmetu IPP 2020/2021

**Meno a priezvisko:** Tomáš Zatl'ko

**Login:** xzatko02

**Skript:** interpret.py a test.php

### Skript interpret.py

Funkcionalita skriptu interpret.py, ktorý slúži na načítanie XML reprezentácie programu, interpretáciu inštrukcií a následné vygenerovanie výstupu, je rozvrhnutá do 4 funkcií:

1. Funkcia **main** načítava XML reprezentáciu a uloží ju pomocou knižnice ElementTree. Následne kontroluje správnosť užívateľom zadáných parametrov, nad inštrukciami vykonáva lexikálne a syntaktické analýzy – kontroluje sa správnosť dátových typov a regulárnymi výrazmi obsah premenných, pomocou cyklu načítava každú jednu inštrukciu, ukladá a kontroluje si správnosť jej argumentov, prípadne sa ukladajú do zásobníkov – globálny, lokálny a dočasný rámec. Uchováva sa ich názov, dátový typ a hodnota. Vo vetve if-else je podľa názvu inštrukcie pridelená postupnosť príkazov, ktoré sa majú nad danými argumentami vykonať.
2. Funkcia **check\_regex** pomocou regulárnych výrazov kontroluje povolené dátové typy a obsah týchto premenných.
3. Funkcia **check\_arg1\_var\_set\_data\_type** kontroluje prvý argument inštrukcií a nastavuje koncový dátový typ podľa pravidiel práve vykonávanej inštrukcie.
4. Pomocná funkcia **check\_add\_sub\_mul\_idiv** kontroluje a spracováva argumenty inštrukcií add, sub, mul, idiv.

Po všetkých kontrolách sa dané inštrukcie postupne zinterpretujú, vygenerujú prípadný výstup na štandardný výstup a vráti sa príslušná návratová hodnota.

## Skript test.php

Funkcionalita skriptu test.php, ktorý slúži na automatické otestovanie dvoch predom vytvorených skriptov – parse.php a interpret.py, je rozvrhnutá do dvoch častí:

1. Počiatočná inicializácia generovania HTML výstupu.
2. Automatické otestovanie súborov vložených pomocou parametru --directory=file, prípadne v aktuálnej zložke a následné dogenerovanie výsledkov automatických testov do HTML výstupu.

Skript vygeneruje hlavičku HTML a zavolá funkciu **auto\_tests**, v ktorej kontroluje užívateľom zadané parametry. Rozhodne tým o vykonávaní kompletného testovania / testovanie interpretu / testovanie parseru / rekurzívneho testovania alebo vkladania ciest skriptov a JExamXML nástroja. Po kontrole argumentov prebehne cyklus hľadania \*.src súborov v zadanom adresári z dôvodu neskoršieho výpisu štatistík do HTML. V ďalšom cykle prebieha vyhľadávanie a pri prípadnej neexistencii - vytvorenie súborov \*.in, \*.rc a \*.out. V ďalšom kroku sa rozhoduje o vykonávaní skriptu parse.php a interpret.py práve podľa užívateľom zadaných parametrov. Po vykonaní skriptov prebehnú kontroly návratových hodnôt s referenčnými hodnotami ktoré rozhodujú o úspešnosti, prebieha prípadná kontrola XML súborov pomocou nástroja JExamXML alebo kontrola výstupu z interpretu a referenčného výstupu pomocou *diff*. Skript disponuje aj funkciou zobrazenia počtu celkových vykonaných testov, počtu vykonaných testov v danom adresári a počty úspešných / neúspešných testov. V závere cyklu prebieha vymazanie prípadných vytvorených dočasných výstupových súborov.

### Rozhodovanie o úspešnosti testu:

1. Pokiaľ sú návratové hodnoty skriptu a \*.rc súboru rovné 0, vykoná sa kontrola \*.out súborov, pokiaľ sa táto kontrola pomocou **diff** rovná 0 – test úspešne prešiel.
2. Pokiaľ majú návratové hodnoty \*.rc súborov rovnakú hodnotu a zároveň sa nerovnajú 0 – porovnanie \*.out súborov nie je nutné – test úspešne prešiel.
3. Pokiaľ sa návratové hodnoty \*.rc súborov nerovnajú – test neprešiel úspešne.

Na záver jedného behu cyklu sa pri každom testovacom súbore v HTML vypíšu referenčné a skutočné návratové hodnoty a informácia o úspešnosti daného testu. Po testovaní *posledného* \*.src súboru prebehne výpis štatistík úspešnosti z *celého adresára* do HTML. Na záver sa môže (pomocou parametru --recursive) rekurzívne spustiť funkcia **auto\_tests**, ktorá spomedzi súborov zadaného adresára hľadá výskyt podadresárov.