

## Implementační dokumentace k 2. úloze do IPP 2019/2020

Jméno a příjmení: Filip Pospíšil

Login: xpospi0f

### **interpret.py**

Hlavní skript programu je kompletně implementován v jednom souboru, který je rozdělen do logických částí. Při implementaci byl kladen důraz na objektovou orientaci.

Ihned po spuštění skriptu dojde k načtení argumentů, které jsou zadány při vstupu. Jejich výčet je možné zjistit voláním `--help` při spouštění skriptu. V případě úspěšného proparsování argumentů dochází k předání hodnot do patřičných proměnných, v opačném případě dochází k ukončení skriptu.

Dále následuje kontrola vstupních souborů – práce s XML souborem je zajištěna pomocí `ElementaryTree`. V případě, že se jedná o nekorektní XML, dochází k okamžitému ukončení skriptu. Jestliže je `source` načítán přímo ze `stdinu`, vše probíhá stejně. Při načítání `inputu` ze souboru kontrolujeme existenci souboru. Při načítání ze `stdinu` v této fázi nedochází k žádné kontrole.

Následně proběhne parsování XML souboru pomocí funkce `check_xml`, která nejdříve zkontroluje, zda obsahuje parametr `program` a zda je parametr `language` nastaven na `IPPCODE20`. Dále projde všechny instrukce, zkontroluje validnost zadání a shodnost s instrukcemi jazyka `IPPCODE20`. Zároveň načte všechny argumenty jednotlivých instrukcí.

Jestliže vše až doteď probíhalo správně, můžeme přejít ke zpracování instrukcí. Funkce `get_instruction` vše načte do seznamu instrukcí a zároveň zaznamená veškerá návěští.

V poslední fázi programu spouštíme samotné interpretování, které je řešeno pomocí cyklu, který prochází instrukce v seznamu jednu po druhé. Ke každé instrukci zavolá její funkci, ve které je definovaná a předá ji veškeré potřebné parametry. Každá funkce má možnost v případě jakéhokoliv problému skript okamžitě ukončit. V takovém případě vrátí `return value` definovanou v zadání projektu.

## Implementační dokumentace k 3. úloze do IPP 2019/2020

Jméno a příjmení: Filip Pospíšil

Login: xpospi0f

### test.php

Hlavní skript programu je kompletně implementován v jednom souboru, který je rozdělen do logických částí. Při implementaci nebylo použito objektivně orientované programování, což se negativně projevilo na době potřebné pro zpracování testů.

Ihned po spuštění skriptu dojde k načtení argumentů, které jsou zadány při vstupu. Jejich výčet je možné zjistit voláním `--help` při spouštění skriptu. V případě úspěšného proparsování argumentů dochází k předání hodnot do patřičných proměnných, v opačném případě dochází k ukončení skriptu.

V případě, že vše proběhlo v pořádku a mám načtené všechny parametry, začínám s hledáním souborů k testování. Výchozí složka pro testy je ta, kde se aktuálně nachází `test.php`, cestu k testům ale mohu přidělit i pomocí paramteru `--directory`. Případně při použití paramteru `--recursive` dojde k prohledání i všech podsložek.

Ke každému `src` souboru by měl být přítomný i soubor `in`. Pokud ho skript nenalezne, soubor si vytvoří. V takovém případě je soubor prázdný. To stejné se opakuje i se soubory `out` a `rc`. V případě `rc` ovšem soubor není prázdný, ale obsahuje návratovou hodnotu 0. To vše probíhá v cyklu, jeden soubor po druhém. Na konci testu je do výstupního HTML5 outputu přidán každý test, včetně informace o tom, zda uspěl, jaká byla očekávaná návratová hodnota a jak byla skutečná návratová hodnota. Pokud skript běžel s parametrem `--parse-only`, dochází ještě k porovnání výstupních XML pomocí `JEXamXML`.

Po proběhnutí všech testů dojde k dokončení výstupu. Celý HTML5 formát je pojat velmi jednoduše, ale přehledně. Po spuštění si lze prohlédnout výsledek každého testu, případně rovnou přeskočit na konec stránky, kde se nachází přehledná tabulka se všemi informacemi o průběhu testování. O tom, jestli `test.php` běžel v režimu `parse+interpret` nebo pouze `--int-only/--parse-only` nás informuje hlavička stránky. Na konci nalezneme informaci o potřebném času k zpracování testu.