

## Implementační dokumentace k 2. úloze do IPP 2021/2022

Jméno a příjmení: Pavel Tetauer

Login: xtetau00

### Interpret.py

#### 1. Úvod

Úkolem bylo implementovat skript `interpret.py`, který načítá XML reprezentaci kódu, kterou vytvoří `parse.php` v IPP-code22 a s využitím této reprezentace generuje výstup.

#### 2. Implementace

Skript `interpret.py` nejprve zkontroluje, zda jsou správně zadány argumenty. Poté z nich vezme zdrojový a vstupní soubor, které uloží do odpovídajících proměnných.

Následuje použití knihovny `xml.etree.ElementTree`, která vrátí root xml reprezentace.

Tento získaný root je následně použit ve funkcích `check_xml()` a `save_instructions()`. První zmiňovaná funkce `check_xml()` pouze zkontroluje, zda má vstupní XML správnou strukturu. Funkce `save_instructions()` provede kontrolu, zda mají instrukce správné operační kódy, pořadí i tag. Následně jsou tyto instrukce ukládány do třídy `instruction` a zároveň jsou k této třídě přiřazeny i odpovídající argumenty, které jsou uloženy v třídě `argument`. Poté je takto vytvořená třída uložena do seznamu `instructions_list` a nakonec je tento seznam seřazen na základě pořadí instrukce.

Nakonec je zavolána funkce `execute_instructions()`, která je v podstatě jeden velký switch, a zde jsou na základě operačního kódu všechny instrukce provedeny v jim náležících funkcích.

### Test.php

#### 1. Úvod

Následným úkolem bylo k oběma úlohám, tedy `parse.php` a `interpret.py`, vytvořit skript `test.php` pro automatické testování. Skript generuje HTML reprezentaci na standardní výstup

#### 2. Implementace

Obdobně jako u `interpret.py` jsou nejprve zkontrolovány všechny přípustné argumenty a zda jsou v nich zadány validní cesty. Všechny vygenerované výstupy jsou ukládány do složky `tmp`. Testování začíná tak, že se všechny soubory s příponou `.src` uloží do pole `all_tests_array`. Poté následuje cyklus přes všechny testy v `all_tests_array`, ve kterém probíhá veškeré testování na základě uživatelem zvolených argumentů. Vše je vyhodnocováno pomocí `exec`, do kterého jsou vloženy odpovídající příkazy pro spouštění interpretu/parseru. Pokud je obdržén návratový kód 0, následuje porovnání výstupů.

Po proběhnutí všech testů následuje generování výstupního HTML, pro které je vytvořen skript `html_generator`, konkrétně se volá funkce `generate_HTML_file()`. Ta nejprve spustí `HTML_generate_start()`, což vygeneruje začátek HTML souboru, následuje `HTML_generate_table()` s `HTML_generate_tests_lists()`. První funkce generuje tabulku s přehledem úspěšných/neúspěšných testů a druhá vypisuje, které konkrétní testy selhaly/prošly. Nakonec je zavolána funkce `HTML_generate_end()`, která ukončí HTML soubor. Nakonec je smazána složka `tmp` a pomocné soubory v ní.