

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

**Jméno a příjmení:** Jan Šulavík

**Login:** xsulav01

## 1. Interpret XML reprezentace kódu (interpret.py)

Skript načítá vstupní XML reprezentaci pomocí modulu `xml.etree.ElementTree` a provádí lexikální a syntaktickou kontrolu vstupu. Ta je zajištěna třídou `SourceParser`, která definuje metody `checkXMLformat` – kontroluje správnost vstupního XML a `checkXMLsyntax` – kontroluje syntaxi kódu ve vstupním XML. Pokud je vstup v pořádku, tak pro každý element `<instruction>` vytvoří objekt třídy `Instruction`. Tato třída drží všechny potřebné informace o jedné instrukci – typ instrukce, její argumenty a pořadí.

Poté, co jsou vytvořeny všechny instrukce, skript vstupuje do hlavního cyklu `while`, který probíhá, dokud lze načíst další instrukci. V cyklu se rozhoduje podle atributu `opcode` o tom, která instrukce se má vykonat.

Skript `interpret.py` pracuje kromě třídy `Instruction` a `SourceParser` také s třídami `ArgumentParser`, `InstructionHandler`, `Frames` a `DataStack`.

`ArgumentParser` – zajišťuje zpracování argumentů při spouštění skriptu `interpret.py`

`InstructionHandler` – definuje metody, které pracují s instrukcemi – např.: `add_to_list` přidá instrukci do instrukčního listu, `get_instruction` vrací instrukci podle čísla zrovna vykonávané instrukce. Také je zde definován slovník návěstí a zásobník volání, se kterými některé instrukce pracují.

`Frames` – definuje všechny rámce a metody, které s nimi pracují.

`DataStack` – jednoduchá třída definující zásobník dat a dvě metody `pop` a `push`.

Pokud interpretace skončí úspěchem, skript vrací 0 a generuje odpovídající výstup. Jestliže nastane při interpretaci chyba, skript vrací návratový kód podle specifikace zadání a krátký popis nastalé chyby.

## 2. Testovací rámec (test.php)

Skript testuje parse.php a interpret.py z předchozích částí úkolu. Podle argumentů zadaných při spuštění testuje buď pouze interpret, pouze parser, nebo případně oba skripty zároveň. Pokud skript najde všechny potřebné soubory, tak vykoná testy obsažené v cestě specifikované argumentem directory.

Pro uložení nastavení skriptu test.php existuje třída Options, která uchovává všechny informace o jeho nastavení podle zadaných argumentů.

Výsledek každého testu se ukládá do objektu třídy TestResults a tyto výsledky se vkládají do pole. Po vykonání všech dostupných testů skript vytvoří HTML dokument, který obsahuje základní informace o proběhlém testování – kolik testů proběhlo celkem a kolik jich uspělo nebo neuspělo.

Součástí html je také tabulka, kde se pro každý test zobrazí informace o něm – návratový kód, výstup, případně chybová hláška a krátký popis, pokud test nedopadl podle očekávaných výsledků.