

# Implementační dokumentace k projektu do IPP 2018/2019

Jméno a příjmení: Nikolas Masica

Login: xmasic00

## 1. Skript interpret.py

Skript `interpret.py` má za úkol načíst XML reprezentaci programu (výstup skriptu `parse.php`) a tento program interpretovat s využitím standardního výstupu.

Jako první skript provádí parsování zadaných argumentů a jejich kontrolu pomocí funkce `checkArguments`. Následně funkce `loadInput` načítá data z XML souboru. K práci s XML využívám knihovnu `xml.etree.ElementTree`. Skript po načítání dat pomocí funkce `checkHeader` zkontroluje hlavičku, která musí obsahovat informace o jazyku. Dále kontroluje zda všechny načítané instrukce jsou instrukcemi jazyka `IPPCode19` a zda mají správný `opcode` (rostoucí čísla jdoucí za sebou). Interpret pracuje s několika globálními proměnnými, které byly inicializovány na začátku skriptu.

U každé instrukce se provede jako první kontrola na počet atributů. Pokud skript narazí na nějakou chybu tak je skript ukončen s příslušnou návratovou hodnotou. Dále se do pole ukládají hodnoty argumentů. Každý argument prochází jednotlivými kontrolami. Hlavní funkcí skriptu je funkce `interpret`, která provádí samotnou interpretaci programu a následovně výpis výsledků programu.

## 2. Skript test.php

Skript `test.php` slouží pro automatické testování skriptů `parse.php` a `interpret.py`.

Jako první skript provádí parsování zadaných argumentů a jejich kontrolu pomocí funkce `getopt`. Skript pomocí funkce `preg_match` hledá v zadaném adresáři (případně v adresáři kde se nachází skript) soubory. Následně jsi ukládá názvy souborů do pole, z kterého pak cyklicky testuje všechny soubory. Následně pomocí funkce `exec` spustí skript `parse.php` nebo `interpret.py` se všemi soubory v poli názvu souboru. Výstupy, návratové kódy následně ukládá do proměnných `$parseReturnCode`, `$ParseOutput`, `$InterpretReturnCode`, `$interpretOutput`. Získané návratové hodnoty pak porovnává s návratovými kódy získanými ze souboru s příponou `.rc/.out`. Pokud se výstupy rovnají, tak si skript do proměnné `$html` uloží HTML kód s úspěchem, která bude dál v skriptu vypsána na `stdout` a iteruje proměnnou `$testPassed`. V případě rozdílných hodnot je do proměnné `$html` zapsány HTML kód s neúspěchem a bude iterována proměnná `$testFailed`. Pro porovnávání návratových kódů a výstupu je prováděno externím programem `Diff`, který při úspěšném porovnání vrací 0 jinak 1.

Pokud byl skriptu zadán jeden z argumentů `-parse-only` nebo `-int-only` tak skript `test.php` otestuje pouze určený skript. Pokud byl skript spuštěn bez těchto argumentů tak skript otestuje prve skript `parse.php`. Následně výstup (XML reprezentaci) kódu uloží do dočasné proměnné, která je pak použita při spuštění skriptu `interpret.py`. Po ukončení interpretace kódu skriptem `interpret.py` jsou tyto dočasné soubory vymazány. Na konci skriptu `test.php` je vypsání HTML kódu na `stdout`. Výstupní HTML obsahuje počet testovaných souborů, počet úspěšných testů, počet neúspěšných testů a tabulku s adresáři testu, očekávanými kódy, mými kódy a nakonec výsledek testu.