

## Implementační dokumentace ke 2. a 3. úloze do IPP 2018/2019

**Jméno a příjmení:** Jan Beran

**Login:** xberan43

### Úvod

Tato dokumentace stručně popisuje skripty `interpret.py` a `test.php`.

### Zadání

Zadáním bylo vytvořit dva skripty; interpret jazyka IPPcode19 převedeného do XML a testovací rámec jak pro interpret, tak pro parser, který byl součástí prvního odevzdání. Pro interpret byl specifikovaným jazykem Python 3.6, pro testovací rámec PHP7.3.

### Implementace interpretu

Interpret je implementován s využitím objektově orientovaných principů, které výrazně usnadnily celou tvorbu skriptu. V mém návrhu je hlavním objektem objekt `interpret`, starající se o samostatnou interpretaci kódu. Rámce pro proměnné a jejich zásobník byly též implementovány jako samostatné objekty.

Interpret na vstupu přijímá XML podobu IPPcode19 a vstupní hodnoty pro interpretaci kódu a na výstupu vypíše výstup interpretovaného kódu, případně chybové hlášení, pokud se interpretovaný kód nebo sám interpret dostane do chybového stavu.

### Spouštění interpretu

Skript se spouští pomocí příkazové řádky jako skript v jazyce Python3. Pracuje s několika parametry:

- `--help`: Vypsání krátké nápovědy k programu
- `--source=source`: specifikuje soubor pro XML reprezentaci IPPcode19, ze kterého mají být načítána data
- `-input=input`: specifikuje soubor se vstupy pro interpretovaný kód

Vždy alespoň jeden z dvojice posledně zmíněných parametrů musí být zadán. Pokud jeden z parametrů chybí, příslušná data se načítají ze standardního vstupu.

### Implementace testovacího rámce

Testovací rámec je implementován funkcionálně s důrazem na znovupoužitelnost pro další projekty, jelikož tento skript plánují využívat i pro testování dalších projektů, bude-li to vhodné. Kromě serveru Merlin, který jsem považoval za referenční stroj, skript funguje (a byl testován) i pod OS Windows, a to z výše zmíněného důvodu.

## Spouštění testovacího rámce

Aplikace se spouští pomocí příkazové řádky jako skript pro PHP7.3. Pracuje s několika parametry:

- `--help`: Vypsání krátké nápovědy k programu
- `--recursive`: Zda mají být při hledání testů prohledávány i podsložky
- `--directory=`*path*: Určení složky, ve které se nacházejí testy. Při absenci tohoto přepínače se testy hledají v aktuálním umístění.
- `--parse-script=`*file*: Určení cesty k analyzátoru zdrojového kódu `parse.php`. Při absenci přepínače bude skript hledán v aktuálním adresáři.
- `--int-script=`*file*: Určení cesty k interpretu zdrojového kódu `interpret.py`. Při absenci přepínače bude skript hledán v aktuálním adresáři.
- `--parse-only`: Při tomto přepínači se bude testovat pouze skript pro analýzu zdrojového kódu. Nesmí se kombinovat s přepínači `--int-script` nebo `--int-only`.
- `--int-only`: Při tomto přepínači se bude testovat pouze skript pro interpretaci zdrojového kódu. Nesmí se kombinovat s přepínači `--parse-script` nebo `--parse-only`.

## Závěr

Skripty byly implementovány samostatně a odpovídají všem náležitostem zadání.