

## 1. Úvod

Cílem první části projektu bylo naimplementovat skript `parse.php` v jazyce PHP ve verzi 7.3. Daný skript má za úkol přeložit zdrojový kód IPPcode19 a vygenerovat odpovídající reprezentaci v jazyce XML. Funkcionalita skriptu spočívá v načítání ze standardního vstupu, provedení lexikální a syntaktické analýzy a následném vypísání XML reprezentace na standardní výstup. V případě chyby bude na standardní chybový výstup vrácen náležitý chybový kód.

## 2. Popis implementace

Implementace skriptu pro překlad kódu je rozvržena do 4 modulů, které představují logické části programu. Jednotlivé moduly jsou popsány níže.

### 2.1 `parse.php`

Základní komponentou, která řídí běh celého programu je `parse.php`. Zajišťuje výpis nápovědy, načtení zdrojového kódu, výpis výsledné XML reprezentace nebo případný výpis chybového hlášení na chybový výstup. Načtení vstupu zajišťuje funkce `file_get_contents`, která načte vstup do řetězce. Z něj je poté vyříznuta povinná hlavička programu, která je předána modulu `parser.php` ke kontrole. V případě úspěchu je výpis výstupního XML realizován pomocí knihovny DOM.

### 2.2 `scanner.php`

Lexikální analýzu realizuje modul `scanner.php`. Na vstupu hlavní funkce `getTokens` je pole lexémů, které odpovídá jednomu řádku programu. Scanner iterativně prochází každou položkou pole a pomocí nástroje pro regulární výrazy `regex` zajišťuje identifikaci jednotlivých tokenů. Ignorování komentářů v programu je řešeno vyskočením z tohoto cyklu, protože na daném řádku dle syntaxe IPPcode19 se nemůže již vyskytovat žádná instrukce ani operand.

### 2.3 `parser.php`

Uvedený modul má dvě úlohy. První je kontrola správného formátu hlavičky, tou druhou je syntaktická kontrola samotného kódu. Základním stavebním kamenem syntaktické analýzy je funkce `checkSyntax`, které je jako parametr předán vstupní program v podobě řetězce. Tato funkce řetězec převede na pole, kde každým prvkem pole je jeden řádek programu (tedy jedna instrukce) a iterativně pro každou instrukci volá funkci `getTokens` z modulu `scanner.php`. Ta vrací pole tokenů. Nad tímto polem je poté kontrolován počet a typ operandů dané instrukce. V případě, že ověřování proběhne úspěšně, je volána funkce pro generování dané instrukce v XML.

### 2.4 `token.php`

Za účelem zpráhlednění implementace byl založen modul `token.php`. Obsahuje pouze dvě třídy. První je třída `Token`, která obsahuje jako proměnné typ tokenu a jeho atribut. Jedinou její funkcí je konstruktor pro instancializaci objektů. Druhou je třída `TokenType`, která obsahuje několik polí, jejichž prvky reprezentují typ tokenů.

## Graf relací mezi jednotlivými moduly

