

1 Popis struktury

Pro realizaci parceru byl vyzkoušen objektově orientovaný přístup. Skript jsem strukturoval o logické soubory. Hlavní a spustitelný soubor je **parse.php**. Byly implementovány třídy: **LexAnalysis** - která představuje lexikální analyzátor, **SynAnalysis** - představuje syntaktický analyzátor, **Token**, **XML** - třída která je obalem knihovny „XMLWriter“ pro naši účely.

2 Realizace

Třída **SynAnalysis** řídí celý program, jejíž objekt byl vytvořen v **parse.php**. Ve syntaktickém analyzátoru je vytvořen objekt třídy **LexAnalysis** se kterou provádí komunikaci.

2.1 LexAnalysis

Na začátku v lexikálním analyzátoru probíhá načítání ze vstupu zdrojového kódu po řádcích. Podle zadání každý řádek musí obsahovat buď instrukci a operandy, nebo může být prázdný, a nebo je komentář. Řádek může mít na začátku instrukci, operand a pak komentář. Podle zadání komentáře a prázdné řádky nás vůbec nezajímají a proto je přeskočíme. Když řádek obsahuje instrukci a komentář, komentář odstraníme. Dál rozdělíme řádek podle mezer a ověříme správnost instrukce. Když instrukce je správná, bude v lexikální analyzátor předán seznam možných pro ni argumentů, a pomocí regulárních výrazů bude zkontrolována jejich správnost, vrátíme token pro každý argument.

2.2 SynAnalysis

Syntaktický analyzátor, jak bylo uváděno dříve, řídí proces a po načtení lexikálním analyzátozem instrukci posle seznam možných argumentu. Seznam možných argumentu je realizován pro krácení kódu a nejjednodušší implementaci dodávání nové instrukce do jazyků.

Příklad kódu pro předání seznamu možných argumentu

```
private function return_arguments() {  
    return match ($this->instruction_name) {  
        "CREATEFRAME", "PUSHFRAME", "POPFRAME", "RETURN", "BREAK" => [],  
        "DEFVAR", "POPS" => ["var"],  
        "LABEL", "JUMP", "CALL" => ["label"],  
        "PUSHS", "WRITE", "EXIT", "DPRINT" => ["symbol"],  
        ...  
    }  
}
```

Také provádím kontrolu správnosti posloupnosti argumentů instrukci. Například instrukce **ADD** prvním argumentem přijímá proměnnou, druhým a třetím musí být typu **int**. Takže provádím kontrolu druhého a třetího argumentu buď je to proměnná nebo int. Protože napsání konstant v jazyce IPPcod22 je různé a vzniká konverzní otázka brát to jako syntaktickou nebo sémantickou chybu. Dál už provádím převod do XML struktury.

3 Parametry příkazový řádky

Je možnost spustit skript s parametrem **- - help** nebo **- h**. Bude vypsán stručný popis spuštění skriptu. Pro využití jiných parametrů bude vypsána chyba s návratovým kódem **10**