

1 IPP - Dokumentace 1. části

Praser obsahuje hlavní cyklus, který načítá řádek ze STDIN. Načtený řádek následně zpracuje, odstraní přebytečné mezery a komentáře a rozdělí ho na tokeny, které jsou dále analyzovány.

1.1 Třída Parser

V programu se vytváří jedináček této třídy, kterému se při inicializaci předá instance XMLWriter pro zápis XML formátu. Třída obsahuje privátní atributy a metody potřebné pro kontrolu syntaktické a lexikální analýzy tokenů a pro výpis na STDOUT ve formátu XML. Hlavní metodou této třídy je `parse_line`, která slouží pro analýzu tokenů načteného řádku.

1.2 Syntaktická analýza

Je řízena jednoduchou gramatikou:

- `<op-code>`
- `<op-code> <var>`
- `<op-code> <symb>`
- `<op-code> <label>`
- `<op-code> <var> <symb>`
- `<op-code> <var> <type>`
- `<op-code> <var> <symb1> <symb2>`
- `<op-code> <label> <symb1> <symb2>`

Pro zkrácení zde není uvedeno, které operace patří do jakého pravidla. Operace jsou vypsány v zadání. Nejdříve se při analýze tokenů zkontroluje operace, podle které se vybere pravidlo pro další analýzu.

1.3 Lexikální analýza

Je implementována pomocí regulárních výrazů a kontrolních podmínek.

- Identifikátor a návěští - `^[_\\-$%*!\\?a-zA-Z][_\\-$%*!\\?a-zA-Z0-9]*$`
- Číselný literál - `^[+|-]?([1-9][0-9]*|[0-7]+|0[o|O][0-7]+|0[x|X][a-fA-F0-9]+)$`
- Řetězcový literál - `^[^\\x00-\\x20\\x23\\x5C]*((\\\\\\\\[0-9]{3})[^\\x00-\\x20\\x23\\x5C]*)*$`
- Typový literál - `^(int|bool|string)$`