COMPILADORES

PROF. TARCÍSIO LUCAS

Chegou a hora de implementar nosso PARSER!

:D

Cada time terá a seguinte gramática de referência para ser implementada:

```
<programa> ➞ int main() <bloco>
<bloco>       ➞ { <decl_var>* <comando>* }
<comando> ➞ <comando_básico>
                    | <iteração>
                    | if ( <expr_relacional> ) <comando> else <comando>?
<comando_básico> ➞    <atribuição>
                           | <bloco>
<iteração>            ::=    while ( <expr_relacional> ) <comando>
<atribuição>         ::=    <id> = <expr_arit> ;
<expr_relacional> ::=    <expr_arit> <op_relacional> <expr_arit>
<expr_arit>          ::=    <expr_arit> + <termo>
                           | <expr_arit> - <termo>
                           | <termo>
<termo>                ::=   <termo> * <fator>
                            | <termo> / <fator>
                            | <fator>
<fator>                ::=    ( <expr_arit> )
                            | <id>
                            | <float>
                            | <inteiro>
                            | <char>
```

Declaracões:

```
<decl_var> ::= <tipo> <id> ";"
<tipo> ::= int | float | char
```

Expressões (ordem de precedência):

1. *, /

2. +, -
3. ==, !=, <, >, <=, >=

OBS1: Expressões apenas com os operadores *, /, +, - são expressões aritméticas.
Expressões com os operadores de comparação (==, <, ...) são expressões relacionais.
Não podemos ter mais de um operador relacional em uma expressão.
Podemos ter expressões aritméticas de qualquer lado de um operador relacional. Mas, não podemos ter expressões relacionais em comandos de atribuição.
OBS2: gramática ainda precisa de adaptações para a implementação. Cuidado no Thanos!

Regras gerais:

- Nossa missão aqui será reconhecer um código lido
    - Implementar gramática descrita acima ou criar uma do zero.
    - Não precisaremos gerar uma árvore sintática explicitamente
    - Nosso Parser utilizará a implementação do Scanner e do método *nextToken()*, gerado na atividade de Análise Léxica.
    - A clareza das nossas mensagens de erro segue bem relevante para nós.
    - Indicação da linha e coluna do erro segue um “algo mais” interessante
    - Implemente de forma incremental, um não terminal por vez, testando bem a cada atualização do Parser.