

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL  
FACOM - FACULDADE DE COMPUTAÇÃO

COMPILADORES I (2023/1)  
PROFA. BIANCA DE ALMEIDA DANTAS

**Trabalho Prático**  
**2ª Etapa – Análise Sintática**

## **1 DESCRIÇÃO**

A segunda parte do trabalho prático de nossa disciplina consiste na implementação do analisador sintático descendente preditivo para a linguagem MiniJava modificada, cuja gramática se encontra ao final desse texto.

O analisador sintático deve ser capaz de percorrer o programa fonte, detectar e reportar erros. Não é necessário implementar estratégias de recuperação de erros, entretanto, caso o trabalho forneça alguma estratégia funcional para realizar essa atividade, o trabalho receberá um bônus de até (um) ponto em sua nota total.

## **2 ESPECIFICAÇÕES**

- O trabalho prático poderá ser realizado em grupos de, no máximo, 3 alunos **sem exceções**.
- A linguagem C++ deverá ser utilizada na implementação do trabalho.
- A entrega de todas as etapas deve ser realizada até o dia: **23/06/2023**.

### 3 GRAMÁTICA

A gramática seguinte utiliza as notações  $(N)^*$  para representar 0 ou mais repetições de  $N$  e a notação  $(N)?$  para representar 0 ou 1 repetição de  $N$ . Os tokens da linguagem são representados em **negrito** e os não-terminais em *itálico*.

1.  $Program \rightarrow MainClass (ClassDeclaration)^* EOF$
2.  $MainClass \rightarrow \text{class ID} \{ \text{public static void main (String[ ] ID)} \{ Statement \} \}$
3.  $ClassDeclaration \rightarrow \text{class ID} (\text{extends ID})? \{ (VarDeclaration)^* (MethodDeclaration)^* \}$
4.  $VarDeclaration \rightarrow Type ID ;$
5.  $MethodDeclaration \rightarrow \text{public Type ID} ( (Type ID (, Type ID)^*)? ) \{ (VarDeclaration)^* (Statement)^* \text{return Expression} ; \}$
6.  $Type \rightarrow \text{int}[ ] \mid \text{boolean} \mid \text{int} \mid ID$
7.  $Statement \rightarrow \{ (Statement)^* \}$ 
  - | **if** ( *Expression* ) *Statement* **else** *Statement*
  - | **while** ( *Expression* ) *Statement*
  - | **System.out.println** ( *Expression* ) ;
  - | **ID** = *Expression* ;
  - | **ID** [ *Expression* ] = *Expression* ;
8.  $Expression \rightarrow Expression Op Expression$ 
  - | *Expression* [ *Expression* ]
  - | *Expression* . **length**
  - | *Expression* . **ID** ( ( *Expression* (, *Expression* )^\*)? )
  - | **INTEGER\_LITERAL**
  - | **true**
  - | **false**
  - | **ID**
  - | **this**
  - | **new int** [ *Expression* ]
  - | **new ID** ( )
  - | **!** *Expression*
  - | ( *Expression* )
9.  $Op \rightarrow \&\& \mid < \mid > \mid == \mid != \mid + \mid - \mid * \mid /$