

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
FACOM - FACULDADE DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO
COMPILADORES I (2018/2)
PROFA. BIANCA DE ALMEIDA DANTAS

Trabalho Prático - Análise Sintática e Interface Gráfica

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, e de uma interface gráfica para o nosso ambiente de compilação.

O analisador sintático deve ser capaz de percorrer o programa fonte, detectar e reportar erros. Seu programa não precisa utilizar técnicas de recuperação de erros, ou seja, ao primeiro erro detectado, o programa deve parar a execução após reportá-lo.

A interface gráfica desenvolvida servirá para facilitar o processo de edição e compilação do arquivo fonte (o qual deve ter a extensão `.mj`). Ela deverá fornecer, pelo menos, as seguintes funcionalidades:

- Uma área de texto para exibir o código fonte do programa;
- Uma área de texto para exibir mensagens do compilador;
- Opções para abrir e salvar arquivos, bem como para criar um novo;
- Opções para compilar o arquivo aberto.

2 ESPECIFICAÇÕES

- O trabalho prático poderá ser realizado em grupos de, no máximo, 3 alunos **sem exceções**.
- A linguagem Java deverá ser utilizada na implementação do trabalho, sugere-se também a utilização do IDE NetBeans.
- A entrega de todas as etapas deve ser realizada até o dia: **03/12/2018**.
- Qualquer parte do trabalho copiada ou “fortemente inspirada” nos trabalhos de outros grupos receberão nota **zero**.
- Entrevistas **podem** ser realizadas com todos os grupos.

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. $Op \rightarrow \&\& \mid < \mid > \mid == \mid != \mid + \mid - \mid * \mid /$