

CURSO: *Ciência da Computação*
DISCIPLINA: *Compiladores*
Prof. Flávio Oliveira

Exercício 2 GA: analisador sintático para uma linguagem de programação

O objetivo do exercício é construir e testar um analisador sintático para a linguagem FOOL (Fake Object-Oriented Language), apresentada no exercício 1 do GA, usando um gerador de analisadores sintáticos. O analisador deve receber um arquivo de entrada contendo um programa em FOOL e gerar como saída as árvores sintáticas das construções encontradas. Para facilitar, reproduzimos abaixo a especificação da linguagem. Sua tarefa é:

1. Estudar gerador de analisadores sintáticos (sugere-se CUP, CUP2 ou ANTLR).
2. Especificar a gramática da linguagem.
3. Gerar e testar o analisador.

O trabalho pode ser desenvolvido em duplas.

Entregar o arquivo com a especificação da gramática, um arquivo de teste e o resultado do teste, na sala correspondente do Moodle na data indicada.

LINGUAGEM FOOL

1. Declaração de classe – declara uma nova classe, com seus atributos e métodos. A forma geral de uma declaração de classe é:

```
{class <identificador> <atributos> <métodos>}
```

2. Declaração de atributo – atributos possuem nome e tipo, que pode ser **bool** ou **int**. A forma geral de uma declaração de atributo é:

```
<tipo> <identificador>;
```

3. Declaração de método – declara um método, que possui um tipo, um nome, argumentos e comandos. A forma geral de uma declaração de método é:

```
<tipo> <identificador>(<argumentos>) {<comandos>}
```

Os tipos possíveis para métodos são **void**, **bool** ou **int**. Cada argumento tem o mesmo formato de uma declaração de atributo, com a diferença de que argumentos são separados por vírgulas. Comandos são separados por ponto-e-vírgula.

4. Comandos – os comandos disponíveis são:
 - a. Atribuição: `<identificador> = <expressão aritmética ou expressão booleana>`
 - b. Condicional:
`if (<expressão booleana>) <comando>; ou`
`if (<expressão booleana>) <comando>; else`
`<comando>; (atenção: podem ser aninhados)`
 - c. Chamadas de métodos (da mesma classe)
 - d. **return** `<expressão aritmética ou expressão booleana>`
5. Expressões aritméticas podem ter constantes numéricas, operadores de soma (+) e multiplicação (*), chamadas de métodos e identificadores. Os identificadores possuem o mesmo formato usado em Java.
6. Expressões booleanas – podem ter identificadores, as constantes **True** e **False**, operadores relacionais (`==`, `<`, `>`), os operadores lógicos **not**, **and**, **or** e parênteses.
7. Um programa em Fool é uma declaração de classe.

O trabalho pode ser desenvolvido em duplas.

Entregar o arquivo .jflex, o arquivo .java gerado, um arquivo de teste e o resultado do teste, na sala correspondente do Moodle na data indicada.