

Enunciado Laboratório 01 de Compiladores.  
Entrega 22/02/2020

**O trabalho pode ser feito em duplas**

Escreva um programa de computador que faça análise léxica do texto de um programa de computador em um subconjunto de Java, para o seguinte subconjunto de tokens abaixo. Você pode criar este programa simulando um autômato finito adaptado para casamento do padrão mais longo com múltiplas alternativas de "estado final" ou utilizar alguma ferramenta para descrição de tokens, tais como JavaCC.

Os tokens que devem ser identificados são:

1. **palavras reservadas.** As palavras reservadas são descritas no item 3.9 do Manual de Especificação do Java 13. Não devem ser consideradas palavras reservadas aquelas chamadas de "restricted keywords" no manual.
2. **identificadores.** Os identificadores devem ser definidos como descritos no item 3.7 do Manual de Especificação do Java 13, *com a restrição das letras ao alfabeto latino, o alfabeto usado no Brasil.*
3. **números inteiros em decimal.** Eles devem ser como descritos no item 3.10 do Manual de Especificação do Java 13, chamado de **DecimalIntegerLiteral** no manual.
4. **números ponto flutuante.** Eles devem ser como descritos no item 3.10.2 do Manual de Especificação do Java 13, restrito à versão decimal de representação, chamado de **DecimalFloatingPointLiteral** no manual.
5. **strings.** Elas devem ser como descritos no item 3.10.5 do Manual de Especificação do Java 13, *sem "escape sequence"s.*
6. **separadores e operadores.** Eles devem ser como descritos nos itens 3.11 e 3.12 do Manual de Especificação do Java 13.

A saída deve conter as informações descrevendo um token por linha.

1. Cada um dentre palavras reservadas, separadores e operadores devem ser descritos por uma linha contendo somente a string que o define;
2. Os demais devem ser descritos por linhas começando com um nome especial, seguido de espaço, seguido da string que define o token, sendo que os nomes especiais são:
  - a. "ID" para identificadores;
  - b. "NUM\_DECIM" para números inteiros;
  - c. "FLOAT\_DECIM" para números de ponto flutuante;
  - d. "STRING" para strings.

Dica: a versão 5.0 do javacc foi mais fácil de instalar no laboratório do IFNMG em 2017.