

## Compiladores - Trabalho Análise Léxica (2019s2)

Linguagem exemplo

```
eep {!  
    #ra INT:  
    #nome STR:  
  
    &setAluno(@ra; @nome) [  
        ~ra => @ra:  
        ~nome => @nome:  
    ]  
  
    *setAluno(202076351; %Marcos%):  
!}
```

As seguintes convenções serão utilizadas:

- A palavra reservada `eep` identifica o início do programa
- O bloco de início e fim do programa deve ser identificado através dos sinais `{ !` e `! }`
- O sinal de `:` determina o término de um comando
- Variáveis devem ser declaradas com o sinal `#`, seguido do nome composto por um conjunto de letras e seu respectivo tipo: `INT` ou `STR`
- A declaração de funções deve ser precedida pelo sinal `&` seguido por um conjunto de letras
- Os parâmetros das funções são identificados pelo caractere `@` seguido por um conjunto de letras
- Os blocos de início e fim de uma função são identificados por `[` e `]`
- Operações de atribuição são identificadas por `=>`
- Strings devem estar entre `%`
- Para que as variáveis sejam utilizadas, devemos colocar o sinal `~` na frente do nome das mesmas
- As funções são chamadas colocando o sinal `*` na frente do nome da função
- Nas funções, os parâmetros são separados por `;` (declaração e chamada)
- Considere o conjunto de palavras reservadas como: `{eep, INT, STR, if, while}`

Escrever um analisador léxico utilizando o JFlex visto em sala de aula para ler um arquivo fonte da linguagem acima (`.eep`) e gerar um arquivo no formato (`lexema, token`). Você deve criar tokens para cada uma das convenções acima. Utilize o Netbeans.

Em grupo: até 4 pessoas

Data da entrega: **16/09/2019** via Moodle e apresentação na aula do dia **17/09**.

Peso: 15% da média final (40% da média final é composta por trabalhos)