

## = Roteiro 02 =

Linguagens Formais e Autômatos - Entrega até 09:00 do dia 25/Out

Prof. José Rui

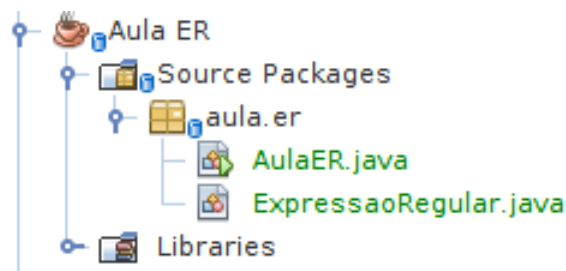
[jose.castro@ifsudestemg.edu.br](mailto:jose.castro@ifsudestemg.edu.br)

### Expressão Regular

#### Descrição do problema

Continuando o trabalho desenvolvido no Roteiro 01, vamos avançar um pouco mais no entendimento de nossas expressões regulares.

Nosso projeto tem as seguintes classes:



Sendo a classe **ExpressaoRegular**, o local onde estão definidas as expressões regulares.

```
12
13 public ExpressaoRegular() {
14
15     BRANCO = "\\s"; // [ \t\n\r\f\v] onde '/' ' (espaço), '\t'
16     BRANCOS = BRANCO + "*";
17
18     DIGITO = "[0-9]";
19     DIGITOS = "(" + DIGITO + "*";
20
21     LETRA = "[A-Za-z]";
22     LETRAS = "(" + LETRA + "*";
23
24     VARIABEL = "(" + LETRA + "(" + LETRA + "|" + DIGITO + ")*";
25
26     INTEIRO = "((-?|\\+?)"+ DIGITOS + ")";
27
28     EXPONENCIAL = "(E(-?|\\+?)"+DIGITOS+" )";
29     FRACAO = "(\\." + DIGITOS+" )";
30     REAL = "(" + DIGITOS + FRACAO + "?" + EXPONENCIAL + "?" + ")";
31
32     DIA = "(1|2|3|4|5)";
33     MES = "(1|2|3|4|5|6|7|8|9|10|11|12)";
34     DATA = DIA + "\\/" + MES + "\\/" + DIGITOS;
35
36 }
```

A atividade do **Roteiro 01**, dentre outras, foi criar uma ER que validasse a chamada/utilização de vetor;

```
1    vetor[7];
2
3    lista[i];
4
5    lista[soma1];
6
```

Vamos observar algumas coisas:

- Na linha 1, a palavra “**vetor**” é um nome de variável. Este nome pode ser qualquer outro desde que respeite a regra de nome de variável. A regra diz que um nome de variável deve iniciar com letra, depois pode vir letra ou dígito diversas vezes.
- Na linha 3, a palavra “**lista**” tem a mesma característica
- Na linha 1, no interior do colchetes apresenta um valor “**inteiro**” (7)
- Já na linha 3 e 5 o interior do colchete apresenta uma variável (i, soma1)

Observe o seguinte trecho de código:

```
1    lista[5];
2
3    int soma(int a, int b);
4
5    produto.nome;
6
7    produto.livros[i].autor;
8
9    lista[ponto.x];
10
```

- Na linha 1, temos a utilização de um vetor de forma simples, conforme visto acima;
- Na linha 3, temos a declaração de um método;
- Na linha 5, temos a utilização de uma classe “**produto**”, acessando o campo “**nome**” ;
- Na linha 7, temos a classe “**produto**”, sendo acessado seu campo “**livros**” que na verdade é um vetor e por isso esta sendo acessado a posição “**i**”, que por sua vez esta acessando o campo “**autor**”
- Na linha 9, temos o acesso ao vetor “**lista**” na posição representada pela classe ponto, campo x.

OBS:

1. Todos os códigos estão **corretos**!
2. Precisamos criar uma expressão regular que aceite todas estas declarações
3. Normalmente, isso tem um nome: **Designador**
4. Assim sendo **Designador** pode ser visto como:
  - nome de variável

- nome de classe
- nome de função
- nome de vetor
- Lembre-se que, em classes, podemos acessar quantos atributos quisermos (produto.custo.x)

## **Atividade 1**

Agora que você entendeu o que seja um DESIGNADOR, faça uma ER para reconhecer estas sentenças.

Não se esqueça de cobrir todos os casos.

## **Atividade 2**

Faça as ER que aceitem: a declaração de funções (linha 1) e chamada de funções linhas 3 e 5

```
1      int soma(int a, int b);  
2  
3      soma(2,5);  
4  
5      soma(3,x);
```

Dica 1: faça uma expressão regular separada para os parâmetros. Perceba que parâmetros nada mais são que declaração de variáveis.

Dica 2: Não se esqueça que o parâmetro pode ser de qualquer tipo (int, float, boolean...) ou mesmo uma classe (Produto, Cachorro, Livro...)

Dica 3: faça uma expressão regular separada para o retorno da sua função.

Dica 4: finalmente agregue todas elas criando a ER da declaração de função.

### **Instruções para entrega**

Compacte seu projeto dentro de uma pasta como R02-nomedadupla e entregue.

OBS: Retire o executável antes de compactar. O gerenciador de e-mail bloqueia executáveis. Se você esquecer de retirar, seu trabalho não chegará no meu e-mail, logo não terá como corrigir. E por favor, sem chorumelllllasssss!!!