

Exercícios C005 – Capítulo 2(Pt. 1)

1. Defina Tokens para os Lexemas encontrados nos códigos abaixo:

A) Função em C++

/ Este programa multiplica x por y */*

```
int main()
{
  int x,y; // Declarando x e y
  x = -2;
  y = 4;
  x = x * y;
  printf("Valor:");
  printf("%d",x);
  return 0;
}
```

R.:

```
<PC, int>; <ID, main>; <(>; <)>; <{>; <PC, int>; <ID, x>; <, >; <ID, y>; <;>;
<ID, x>; <OP, =>; <NUM, -2>; <;>; <ID, y>; <OP, =>; <NUM, 4>; <;>; <ID, x>;
<OP, =>; <ID, x>; <OP, *>; <ID, y>; <;>; <PC, printf>; <(>; <LIT, "Valor">; <)>;
<;>; <PC, printf>; <(>; <LIT, "%d">; <, >; <ID, x>; <)>; <;>; <PC, return>;
<NUM, 0>; <}>;
```

B) Classe em Java

// Uma Classe com uma funcao em Java

```
public class MinhaClasse {
  public static double minhaFuncao(int n)
  {
    // variavel responsavel por fazer a soma
    double soma = 0.0;
    for (int i=1;i<=n;i++)
    {
      soma = soma + i;
    }
    return soma;
  }
}
```

R.: <PC, public>; <PC, class>; <ID, MinhaClasse>; <{>; <ID, public>; <PC, static>; <PC, double>; <ID, minhaFuncao>; <(>; <PC, int>; <ID, n>; <)>; <{>; <PC, double>; <ID, soma>; <OP, =>; <NUM, 0.0>; <;>; <PC, for>; <(>; <PC, int>; <ID, i>; <OP, =>; <NUM, 1>; <;>; <ID, i>; <COMP, <= >; <ID, n>; <;>;

<ID, i>; <OP, ++>; <>>; <{}>; <ID, soma>; <OP, =>; <ID, soma>; <OP, +>; <ID, soma>; <;>; <{}>; <PC, return>; <ID, soma>; <;>; <{}>; <{}>

2. Considere o alfabeto $\Sigma = \{0,1\}$. Para as descrições abaixo, crie Expressões

Regulares (REGEXes) que atendam a cada uma delas:

a. Qualquer sequência de números que iniciam com 10;
 $10[01]^*$

b. Qualquer sequência de números que terminam com três 1's consecutivos;
 $[01]^*111$

c. Qualquer sequência de números que deve começar com 1 e terminar com 0;
 $1[01]^*0$

d. Qualquer sequência de números que deve conter pelo menos três 1s (em qualquer posição da String) – Ex: 111, 010011, 1001101...;
 $[01]^*1[01]^*1[01]^*1$

e. Qualquer sequência de números que deve ter a substring 0101 (ou seja, este conjunto de caracteres todos juntos no meio da sua String) – Ex: 0101, 001011, 11010100...;
 $[0-1]^*0101[0-1]^*$

f. Qualquer sequência de números de comprimento 3;
 $[0-1][0-1][0-1]$

g. Qualquer sequência de números que tem comprimento maior que 3;
 $[0-1]^+ \{3,\}$

h. Qualquer sequência de números que tem comprimento mínimo de três e seu terceiro símbolo é 0;
 $[0-1]^+ \{2\} 0 [0-1]^*$

i. Qualquer sequência de números que tem comprimento de no máximo 3;
 $[0-1]^+ \{,3\}$

j. Qualquer sequência de números que tem comprimento múltiplo de 3 – Ex: 001, 010101...;
 $([01]\{3\})^+$

k. Qualquer sequência de números que formam cadeias que iniciam com 1 e não terminam com 0 – Ex: 1, 1001, 110001...;
 $1[01]^*[1]^+$

l. Cadeias com quantidade par de símbolos – Ex: 00, 01, 1001...;

$[01]$

$([01]\{2\})^+$

m. Cadeias com quantidade ímpar de símbolos – Ex: 0, 111, 10001...;

$[01]([01]\{2\})^*$

n. Qualquer sequência de números que tem comprimento diferente de 3;

$([01]\{1\} \mid [01]\{2\} \mid [01]\{4,\})$

o. Final Boss: Contém pelo menos dois 0's e no máximo um número 1 – Ex:
00, 001, 10000...;

$([0]^+[1]?[0]^+ \mid [1]?[0]^+[0]^+ \mid [0]^+[0]^+[1]?)$