**Algoritmos e Lógica de Programação (Aula 5)**

Variáveis

boolean

int

double

strings

**ENTRADA**

**Scanner**

nextInt()

nextDouble()

nextLine()

**PROCESSAMENTO**

+ - \* / %

math

sqrt()

pow()

rand()

**SAÍDA**

System.out.print()

Para escrever os teste lógico, precisamos:

1. Operadores Relacionais

**>, >=**

**<, <=**

**==, !=**

Lembrando: a atribuição é sempre da direita para a esquerda:

Variável = expressão;

x = 2;

x = a + b;

1. Operadores Lógicos

e **&&** (conjunção)

ou **||** (disjunção)

não **!** (negação)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **a** | **b** | **a&&b** | **a||b** |  | **a** | **!a** |
| V | V | V | V |  | V | F |
| V | F | F | V |  | F | V |
| F | V | F | V |  |  |  |
| F | F | F | F |  |  |  |

**Exercícios**

|  |  |
| --- | --- |
| Considere:  a =2;  b = 7;  c = -5  d = 4  e = 1 | Diga se é V ou F  d >= 7 (Verdadeiro)  b == d + e; (Falso)  c + d <= b – c (Verdadeiro)  a + 1 == d – e (Verdadeiro)  c == - (d +e) (Verdadeiro) |

**Observação:**

1. A resolução de uma expressão tem a seguinte prioridade:

1ª Operadores ou expressões matemáticas (sqrt)

2ª Operadores relacionais

3ª operadores lógicos

1. Se houver parênteses, ela pode mudar prioridades.
2. Entre os operadores lógicos:

1º não !

2º e &&

3º ou ||

|  |  |
| --- | --- |
| **Mais exercícios:**  **b >= 7 && c +d < = b – c**  V && V  V  **! (b >= 7) || c == - (d + e)**  ! V || V  F || V  V | **! (a + b < 8 || a > e)**  ! (F || V)  ! (V)  F |

Na linguagem Java, temos 3 comandos de decisão:

1. SE – SE NÃO : IF – ELSE

if (condição verdadeira) {

Instruções, caso verdadeiro

}

else {

Instruções caso falso

}

**Observações:**

1. ELSE não é obrigatório;
2. Se houver mais que 2 possibilidades, será necessário mais comandos *if* (encadeados ou aninhados);
3. Quando houver só um comando no *if* ou no *else*, as chaves são opcionais.