

# **Algoritmos e Estruturas de Dados**

**Expressões**

**Java**

# Expressões

Uma expressão é uma combinação de elementos, que podem ser valores, variáveis, operadores e chamadas a funções. Com o uso das expressões é possível realizar cálculos que produzem novos valores, ou seja, fazem a transformação das informações.

# Variáveis

Como visto anteriormente, variáveis são localizações na memória que armazenam dados.

Para associar um valor a uma variável utiliza-se um **operador de atribuição** ao lado direito da variável à qual desejamos atribuir o valor.

## Exemplo

Um exemplo de expressão é

```
soma = 5 + 4 ;
```

em que:

- 5 e 4 são valores,
- + é um operador aritmético,
- = é o operador de atribuição,
- soma é uma variável, e
- ; indica o final da instrução.

Podemos ler a expressão como "*soma **recebe** cinco mais quatro*".

A execução da expressão 5 + 4 expressão resulta no valor 9 , o qual será armazenado na variável soma .

# Operadores

Os operadores são utilizados para construir **expressões**, que podem conter diferentes quantidades de operandos.

## Atribuição

Como dito anteriormente, a atribuição é o operador que determina a passagem de valor para uma variável. Por definição toda variável pode ser seu valor alterado, e a modificação deste valor é realizada com o operador de atribuição.

Operador	Função
=	atribuição

## Exemplo

- `lado = 8 ;`
- `distancia = 49.6 ;`
- `nome = "Adalberto" ;`
- `custo = 5.50 ;`

## Aritméticos

São aqueles utilizados para a execução de operações matemáticas, como adição, multiplicação, subtração, divisão, potenciação, etc.

Operador	Função	Exemplo
<code>+</code>	adição	<code>5+2 = 7</code>
<code>-</code>	subtração	<code>5-2 = 3</code>
<code>*</code>	multiplicação	<code>3*6 = 18</code>
<code>/</code>	divisão	<code>10/2 = 5.0</code>
<code>Math.pow(x, y)</code>	potência. $x^y$	<code>Math.pow(5, 2) = 25</code>
<code>Math.sqrt(x)</code>	raiz quadrada. $\sqrt{x}$	<code>Math.sqrt(16) = 4</code>



## Exemplos

```
int valor_i;  
float valor_f;  
  
valor_i = 5+10;  
System.out.println(valor_i); //15  
valor_i = 8-4;  
System.out.println(valor_i); //4  
valor_i = 6*4;  
System.out.println(valor_i); //24  
valor_i = 10/2;  
System.out.println(valor_i); //5  
valor_f = 10/2;  
System.out.println(valor_f); //5.0  
valor_f = (float) 10/6;  
System.out.println(valor_f); //1.6666666  
valor_i = 25%11;  
System.out.println(valor_i); //3
```

## Exercícios

- Calcule a área de um quadrado de lado  $L$ . Utilize duas variáveis.
- No dia de seu aniversário, Tomás deseja saber qual é a sua idade, em dias. Considere que Tomás está completando  $X$  anos, e cada ano possui 365 dias. Utilize duas variáveis.

## Relacionais

São operadores de comparação entre valores. As expressões realizadas com estes operadores retornam um resultado do tipo lógico, verdadeiro ou falso (FORBELLONE 2022, p.27).

Operador	Função
>	maior que
<	menor que
==	igual
!=	diferente
>=	maior ou igual
<=	menor ou igual

## Exemplos

```
System.out.println(6>7); //false
System.out.println(6<7); //true
System.out.println(6==7); //false
System.out.println(4==4); //true
System.out.println(12>=12); //true
System.out.println(15>=12); //true
System.out.println(12>=12); //true
System.out.println(12<=12); //true
System.out.println(15<=12); //false
System.out.println(8<=12); //true
System.out.println(3!=11); //true
System.out.println(12!=12); //false
```

## Lógicos

Os operadores lógicos são utilizados para a composição de expressões lógicas. As possibilidades de resultados de expressões lógicas considerando as entradas são chamadas de Tabelas-Verdade. São elas:

## Conjunção

A conjunção corresponde ao **e** lógico. Possui resultado verdadeiro apenas quando ambas entradas forem verdadeiras, e falso para os demais casos.

p	q	p e q
V	V	V
V	F	F
F	V	F
F	F	F

## Disjunção

A disjunção corresponde ao **ou** lógico. É falso apenas quando ambas as entradas são falsas. Basta apenas um dos operandos serem verdadeiros para resultar em verdadeiro.

p	q	p ou q
V	V	V
V	F	V
F	V	V
F	F	F



## Negação

A negação corresponde ao **não** lógico. Consiste na inversão lógica do valor de entrada. A negação é um operador unário, ou seja, atua sobre um único operando.

p	não p
V	F
F	V

Em resumo, para os operadores lógicos temos:

Operador	Função
<code>&amp;&amp;</code>	<b>e</b> lógico (conjunção)
<code>&lt;code&gt;    &lt;/code&gt;</code>	<b>ou</b> lógico (disjunção)
<code>!</code>	<b>não</b> lógico (negação)

# Operadores compostos

É possível combinar alguns operadores, que realizam a operação utilizando os parâmetros passados ao operador, e em seguida realiza uma atribuição utilizando a variável à esquerda dos operadores.

Operador	Função
<code>++</code>	incremento de valor 1 (unário)
<code>--</code>	decremento de valor 1 (unário)
<code>+=</code>	adição e atribuição
<code>-=</code>	subtração e atribuição
<code>*=</code>	multiplicação e atribuição

## **Prioridade entre operadores**

As prioridades definem quais operadores serão executados primeiro. Caso possuam a mesma prioridade, a expressão será executada da esquerda para a direita.

Tipo	Símbolo
Operadores pós-fixados operadores pré-fixados <i>cast</i>	<div>x++ x--</div> <div>++x +x -x ~x !x</div> <div>(type) x</div>
multiplicação, divisão	* / %
adição, subtração	+ -
comparação	, <= > >=
igualdade	== !=
e lógico	&&
','	','

Adaptado de [GOODRICH 2013, p.24].

