



## Operadores lógicos

### Operadores lógicos

Muitas vezes precisamos combinar os valores `boolean` s obtidos, por exemplo, com comparações ou diretamente de uma variável. Isso é feito utilizando os operadores lógicos.

Em lógica, as operações mais importantes são: `e` , `ou` , `ou exclusivo` e `negação` .

```
System.out.println(1 == 1 & 1 > 2);    // false.  
System.out.println(1 == 1 | 2 > 1);    // true.  
System.out.println(1 == 1 ^ 2 > 1);    // false.  
System.out.println(!(1 == 1));         // false.
```

[COPIAR CÓDIGO](#)

Antes de terminar a avaliação de uma expressão, eventualmente, o resultado já pode ser descoberto. Por exemplo, quando aplicamos a operação lógica `e` , ao achar o primeiro termo falso não precisamos avaliar o restante da expressão.

Quando usamos esses operadores, sempre os dois lados da expressão são avaliados mesmo nesses casos em que não precisariam.

Para melhorar isso, existem os operadores de curto circuito `&&` e `||` . Quando já for possível determinar a resposta final olhando apenas para a primeira parte da expressão, a segunda não é avaliada:

```
System.out.println(1 != 1 && 1 > 2);  
// false, o segundo termo não é avaliado.
```

```
System.out.println(1 == 1 || 2 > 1);  
// true, o segundo termo não é avaliado.
```

[COPIAR CÓDIGO](#)

A maior dificuldade com operadores de curto circuito é se a segunda parte causa efeitos colaterais (um incremento, uma chamada de método). Avaliar ou não (independente da resposta) pode influenciar no resultado final do programa.

```
public static boolean metodo(String msg) {  
    System.out.println(msg);  
    return true;  
}  
  
public static void main(String[] args) {  
    System.out.println(1 == 2 & metodo("oi"));  
    // imprime oi, depois false  
    System.out.println(1 == 2 && metodo("tchau"));  
    // não imprime tchau, imprime false  
  
    int i = 10;  
    System.out.println(i == 2 & i++ == 0);  
    // imprime false, soma mesmo assim  
    System.out.println(i);  
    // imprime 11  
  
    int j = 10;  
    System.out.println(j == 2 && j++ == 0);  
    // imprime false, não soma  
    System.out.println(j);
```

```
}  
    // imprime 10
```

[COPIAR CÓDIGO](#)