

Importante: Lembre-se de usar parênteses para isolar as operações que devem ser realizadas primeiro!

Vimos exemplos de precedência com o uso de operadores aritméticos, mas a precedência também aplica-se às operações relacionais e lógicas. Logo após a execução das operações aritméticas, devem ser realizadas as operações relacionais e, em seguida, as operações lógicas. A Tabela 4.7 apresenta os operadores e sua ordem de precedência.

Para exemplificar a precedência dos operadores relacionais e lógicos, vejamos o exemplo a seguir:

$$A \leftarrow B + 2 > 5 \text{ .ou. } C <> 4 \text{ .e. } D = 0$$

Considerando: $B \leftarrow 5$; $C \leftarrow 3$; $D \leftarrow 1$

Substituindo as variáveis pelos valores que lhe foram atribuídos:

$$A \leftarrow 5 + 2 > 5 \text{ .ou. } 3 <> 4 \text{ .e. } 1 = 0$$

1. Calculamos as expressões aritméticas $5 + 2$

$$A \leftarrow 7 > 5 \text{ .ou. } 3 <> 4 \text{ .e. } 1 = 0$$

2. Avaliamos as operações relacionais $7 > 5$ resulta .f.; $3 <> 4$ resulta .v.; $1 = 0$ resulta .f.

$$A \leftarrow \text{.f. .ou. .v. .e. .f.}$$

3. Por último, avaliamos as operações lógicas.

Lembre-se: O operador .e. tem prioridade sobre o operador .ou.; dessa maneira, avaliamos primeiro a expressão .v. e .f., que resulta .f.

Reescrevendo a expressão, temos: $A \leftarrow \text{.f. .ou. .f.}$

4. Como resultado $A \leftarrow \text{.f.}$!

TABELA 4.7 Precedência de operadores

Operador		Função realizada
Algoritmo	Java	
(), []	()	Parênteses e colchetes são usados para agrupar expressões, determinando precedência, a exemplo das expressões matemáticas
^ ou **	pow(x, y)	Operador aritmético de potenciação
*, /	*, /	Operadores aritméticos de multiplicação e divisão
+, -	+, -	Operadores aritméticos de adição e subtração
←	=	Operador de atribuição
=, <, >, <=, >=, <>	==, >, <, >=, <=, !=	Operadores relacionais
.não.	!	Operador lógico de negação
.e.	&&	Operador lógico e
.ou.		Operador lógico ou