

Documentação: Java: if/else e o operador ternário

Este documento apresenta as estruturas condicionais if/else e operador ternário, recursos que possibilitam executar diferentes trechos de código com base em uma expressão booleana.

As estruturas de condição possibilitam ao programa tomar decisões e alterar o seu fluxo de execução. É por meio delas que podemos dizer ao sistema: “execute a instrução A caso a expressão X seja verdadeira; caso contrário, execute a instrução B”. Na linguagem Java temos três recursos para criação de estruturas de decisão: `if/else`, `operador ternário` e `switch/case`.

Neste documento apresentamos a opção `if/else` e o `operador ternário`.

Tópicos

[if/else](#)

[else if](#)

[Operador ternário](#)

[Exemplo prático](#)

if/else

A estrutura condicional `if/else` permite ao programa avaliar uma expressão como sendo verdadeira ou falsa e, de acordo com o resultado dessa verificação, executar uma ou outra rotina.

Na linguagem Java o tipo resultante dessa expressão deve ser sempre um `boolean`, pois diferentemente das demais, o Java não converte `null` ou inteiros como 0 e 1 para os valores `true` ou `false`.

Sintaxe do if/else:

```
if (expressão booleana) {  
    // bloco de código 1  
} else {  
    // bloco de código 2  
}
```

As instruções presentes no bloco de código 1 serão executadas caso a expressão booleana seja verdadeira. Do contrário, serão executadas as instruções presentes no bloco de código 2.

O Java utiliza as chaves como delimitadores de bloco e elas têm a função de agrupar um conjunto de instruções. Apesar do uso desses delimitadores ser opcional caso haja apenas uma linha de código, ele é recomendado, pois facilita a leitura e manutenção do código, tornando-o mais legível.

Nota: A declaração do `else` não é obrigatória. Para muitas situações apenas o `if` é suficiente.

else if

Complementar ao `if/else` temos o operador `else if`. Esse recurso possibilita adicionar uma nova condição à estrutura de decisão para atender a lógica sendo implementada.

Sintaxe do `if/else` com `else if`:

```
if (expressão booleana 1) {  
    // bloco de código 1  
}  
else if (expressão booleana 2) {  
    // bloco de código 2  
}  
else {  
    // bloco de código 3  
}
```

Dessa forma, se a expressão booleana 1 for verdadeira, o bloco de código 1 será executado. Caso seja falsa, o bloco de código 1 será ignorado e será testada a expressão booleana 2. Se ela for verdadeira, o bloco de código 2 será executado. Caso contrário, o programa vai ignorar esse bloco de código e executar o bloco 3, declarado dentro do `else`.

Podemos utilizar quantos `else if` forem necessários. Entretanto, o `else` deve ser adicionado apenas uma vez, como alternativa ao caso de todos os testes terem falhado.

Nota: Em trechos de código com muitos `if/else` e `else if`, considere o uso da estrutura `switch/case`, ou mesmo padrões de projeto. O uso de estruturas de condição aninhadas torna o código difícil de ler e escrever.

Operador ternário

O operador ternário é um recurso para tomada de decisões com objetivo similar ao do `if/else`, mas que é codificado em apenas uma linha.

Sintaxe do operador ternário:

```
(expressão booleana) ? código 1 : código 2;
```

Ao avaliar a expressão booleana, caso ela seja verdadeira, o código 1, declarado após o ponto de interrogação (?) será executado; do contrário, o programa irá executar o código 2, declarado após os dois pontos (:).

Nota: Normalmente esse operador é utilizado quando precisamos de uma estrutura de decisão simples, por exemplo, para iniciar uma variável, retornar um valor ou integrar um bloco de código no qual um `if/else` pode torná-lo maior e menos legível.

Exemplo prático

Suponha que você está desenvolvendo um software para controle de estoque que precisa informar como está a quantidade de itens de cada produto: se suficiente, para quantidades superiores a 100; em alerta, para quantidades entre 100 e 50; e abaixo do ideal, para quantidades menores do que 50. Como programar esse código?

Exemplo de uso de `if/else`:

```
int estoque = //valor recuperado do sistema

if (estoque >= 100) {
    System.out.println("Produto com quantidade suficiente.");
} else if (estoque < 100 && estoque > 50) {
    System.out.println("Alerta: Avaliar possibilidade de novo pedido.");
} else {
    System.out.println("ATENÇÃO! Faça um novo pedido.");
}
```

Run

Agora, considere que você precisa de uma estrutura de decisão mais simples, apenas para indicar se estamos na primeira ou na segunda quinzena de um mês.

Exemplo de uso do operador ternário:

```
int numeroDias = //valor entre 1 e 30

System.out.println((numeroDias < 15) ? "Primeira quinzena" : "Segunda quinzena");
```