Estrutura de Decisão (IF) Comando Else e Aninhamento If-Else no JAVA

Estrutura de Decisão

As estruturas de decisão (também conhecidas como Estruturas Condicionais) determinaram que o programador especifique uma ou mais **condição** a ser demonstrada pelo programa, necessariamente com uma instrução (ou conjunto) a ser executada se o resultado do teste for verdadeiro, e, opcionalmente, outra instrução ou conjunto de instruções caso o resultado do teste seja falso.

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.

Comando IF = Se ...

A idéia do comando if é verificar se uma dada condição é verdadeira ou falsa.

if (**condição booleana**){  
    comandos a serem executados se a condição for verdadeira;  
}

A **condição** é verificada a cada passagem pela estrutura IF. Se a condição for satisfeita (V), então a lista de instruções que se encontra entre chaves será feita. Se a condição NÃO for satisfeita (F), então serão feitas as instruções existentes logo após o fecha chaves.

Comando Else = Senao ...

Variação 1 - IF  ELSE

**if**(**condição booleana**)  
   **{** comandos a serem executados se a condição for verdadeira; **}**  
**else**  
    **{** comandos a serem executados se a condição for falsa; **}**

**......**

Neste caso, se a condição colocada após o IF não obtiver resultado verdadeiro, automaticamente serão feitas as **instruções que estão dentro do ELSE**, desconsiderando aquelas que estão abaixo do IF. Caso a condição seja VERDADEIRA, serão feitas as **instruções que estão entre chaves abaixo do IF.**

Quando acabar tanto a lista de instruções abaixo do IF quanto a lista de instruções referente ao ELSE**, automaticamente serão desenvolvidas as instruções que estão após a lista de instruções do ELSE (....).**

Variação 2 - IF  ELSEIF ELSE

**if**(**condição booleana 1**)  
   **{** comandos a serem executados se a condição for verdadeira; **}**  
**else if** (**condição booleana 2**)  
    **{** comandos a serem executados se a segunda condição for verdadeira; **}**  
**else**  
**{** comandos a serem executados se nenhuma condição for verdadeira; **}**

**......**

Neste caso, se a condição colocada após o IF não obtiver resultado verdadeiro, automaticamente serão verificada a condição do ELSE IF. Se esta condição for verdadeira, faz-se a lista de isntruções referente ao ELSE IF. Caso nãp seja verdadeira, faz-se a lista de isntruções**que estão dentro do ELSE**, desconsiderando aquelas que estão abaixo do IF e do ELSE IF

Caso a condição seja do IF VERDADEIRA, serão feitas as **instruções que estão entre chaves abaixo do IF e, ao acabar a lista de instruções,**automaticamente serão desenvolvidas as instruções que estão após a lista de instruções do ELSE, ou seja, referente ao  (....).

Caso a condição do ELSE IF for VERDADEIRA, serão feitas as **instruções que estão entre chaves abaixo do ELSE IF e, ao acabar a lista de instruções,**automaticamente serão desenvolvidas as instruções que estão após a lista de instruções do ELSE, ou seja, referente ao  (....).

Caso a condição do IF e do ELSE IF forem falsas, serão feitas as **instruções que estão entre chaves abaixo do ELSE e, ao acabar a lista de instruções,**automaticamente serão desenvolvidas as instruções que estão após a lista de instruções do ELSE, ou seja, referente ao   (....).

**Operadores de Comparação**

Os *operadores de comparação*, também chamados de operadores relacionais, são símbolos usados para instruir o computador a testar se uma proposição que classifica um valor em relação a outro é verdadeira ou falsa.

| **Operador** | **Nome** | **Descrição** |
| --- | --- | --- |
| + | Mais | Denota uma soma entre dois valores. |
| - | Menos | Especifica uma subtração entre dois valores. |
| \* | Vezes | Efetua a multiplicação entre dois valores. |
| / | Dividido por | Determina uma divisão entre dois valores. |
| ++ | Operador de Incremento | Adiciona 1 ao valor de uma variável. |
| -- | Operador de Decremento | Subtrai 1 do valor de uma variável. |

| **Operador** | **Nome** | **Descrição** |
| --- | --- | --- |
| == | Igual á | Retorna true se os operandos forem iguais |
| != | Diferente de | Retorna true se os operandos são diferentes |
| > | Maior que | Retorna true se o operando à esquerda for maior que o da direita |
| < | Menor que | Retorna true se o operando à esquerda for menor que o da direita |
| <= | Menor ou igual á | Retorna true se o operando à esquerda for menor ou igual ao da direita |
| >= | Maior ou igual à | Retorna true se o operando à esquerda for maior ou igual ao da direita |

## Operadores Lógicos

Um operador lógico é um símbolo usada para ordenar que o computador verifique a veracidade de uma proposição composta, ou da negação de uma proposição.

Uma proposição composta combina duas proposições mais simples, e de quebra faz alguma afirmação acerca do valor verdade de cada uma. É essa afirmação que o computador vai verificar e atribuir um valor verdade à ela.

O operador & (E lógico) forma uma proposição composta que assume que ambas proposições que a compõem são verdadeiras.

O operador | (OU lógico) forma uma proposição composta que assume que pelo menos uma das proposições simples é verdadeira.

O operador ^ (OU exclusivo), forma uma proposição composta que assume que apenas uma das proposições que a compõem é verdadeira, enquanto a outra é necessariamente falsa.

O operador ! (NÃO lógico) é usado para negar uma proposição. Como dito anteriormente, a negação de uma proposição tem o valor verdade oposto da proposição original.

| **Operador** | **Nome** | **Descrição** |
| --- | --- | --- |
| & | E lógico | Retorna true se ambos os operandos forem verdadeiros. |
| | | OU lógico | Retorna true se pelo menos um dos operandos for verdadeiro. |
| ^ | XOR (OU exclusivo) | Retorna true se apenas um dos operandos for verdadeiro. |
| && | E lógico em curto-circuito | Retorna true se ambos os operandos forem verdadeiros. |
| || | OU lógico em curto circuito | Retorna true se pelo menos um dos operandos for verdadeiro. |
| ! | NÃO lógico | Retorna false caso o operando seja avaliado como verdadeiro, ou true se for  falso. |

Aninhamento If-Else

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.”);

}