

Conceitos base da linguagem JAVA

Profª. Heloisa Moura

Conceitos base da linguagem JAVA

O que vamos ver:

- Estrutura de seleção condicional If/else, operador ternário e switch/case

Iniciando no Java - Conceitos base da linguagem - Estruturas de condição

Estruturas de Condição

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.

Iniciando no Java - Conceitos base da linguagem - Estruturas if/else

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:

```
1  if (expressão booleana) {  
2      // bloco de código 1  
3  } else {  
4      // bloco de código 2  
5  }
```


Iniciando no Java - Conceitos base da linguagem - Estruturas if/else

if/else

- 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.*

Iniciando no Java - Conceitos base da linguagem - Estruturas if/else

else/if

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

```
1  if (expressão booleana 1) {  
2      // bloco de código 1  
3  } else if (expressão booleana 2) {  
4      // bloco de código 2  
5  } else {  
6      // bloco de código 3  
7  }
```


Iniciando no Java - Conceitos base da linguagem - Estruturas if/else

else/if

- 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. O uso de estruturas de condição aninhadas torna o código difícil de ler e escrever.

Iniciando no Java - Conceitos base da linguagem – Operador ternário

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:

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

condição? valor se for verdadeiro: valor se for falso

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 (:).

Iniciando no Java - Conceitos base da linguagem – Estruturas de condição

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:

```
1  int estoque = //valor recuperado do sistema
2
3  if (estoque >= 100) {
4      System.out.println("Produto com quantidade suficiente.");
5  } else if (estoque < 100 && estoque > 50) {
6      System.out.println("Alerta: Avaliar possibilidade de novo pedido.");
7  } else {
8      System.out.println("ATENÇÃO! Faça um novo pedido.");
9  }
```


Iniciando no Java - Conceitos base da linguagem – Operador ternário

Operador ternário

Podemos utilizar também o operador ternário encadeado ou seja um condicional ternário dentro do outro.

Exemplo prático:

```
15 // condicional ternario encadeado
16 a = 14;
17 b = (a > 14) ? "atrasado": (a < 14)? "adiantado": "No horario";
18 System.out.println("Resultado: " + b);
19
```


Iniciando no Java - Conceitos base da linguagem - Estruturas de condição

Exemplo prático

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.

Compilador online: <https://www.codiva.io/java> ou <https://www.jdoodle.com/online-java-compiler/>
Exemplo de uso do operador ternário:

```
1 | int numeroDias = //valor entre 1 e 30
2 | System.out.println((numeroDias <= 15) ? "Primeira quinzena" : "Segunda quinzena");
```

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.

Iniciando no Java - Conceitos base da linguagem - Estruturas de condição

Exemplo prático Resolução if/else

```
1
2 public class ExercicioCondicionalUm
3 {
4     public static void main (String[] args)
5     {
6         int estoque = 50; //valor recuperado do sistema
7
8         if (estoque >= 100) {
9             System.out.println("Produto com quantidade suficiente.");
0         } else if (estoque < 100 && estoque > 50) {
1             System.out.println("Alerta: Avaliar possibilidade de novo pedido.");
2         } else {
3             System.out.println("ATENCAO! Faca um novo pedido.");
4         }
5     }
6 }
7
```


Iniciando no Java - Conceitos base da linguagem – Estruturas de condição

Exemplo prático

Resolução: Operador ternário

```
1
2 public class ExercicioCondicionalDois
3 {
4     public static void main (String[] args)
5     {
6         int numeroDias = 5; //valor entre 1 e 30
7         System.out.println((numeroDias < 15) ? "Primeira quinzena" : "Segunda quinzena");
8     }
9 }
10
11
```


Iniciando no Java - Conceitos base da linguagem - Estruturas de condição

Exemplo prático

Resolução: Operador ternário encadeado

```
2 public class ExercicioCondicionalOperTernarioDois {
3     public static void main(String[] args) {
4         int a;
5         String b;
6         /*a= 12;
7
8         b = (a == 5) ? " a é igual a 5": " a é diferente de 5";
9         System.out.println("Resultado: " + b);
10
11        b = (a == 12) ? " a é igual a 12": " a é diferente de 12";
12        System.out.println("Resultado: " + b);*/
13
14
15        // condicional ternario encadeado
16        a = 14;
17        b = (a > 14) ? "atrasado": (a < 14)? "adiantado": "No horario";
18        System.out.println("Resultado: " + b);|
19
20    }
21 }
22
```


Iniciando no Java - Conceitos base da linguagem – Operador ternário

Exercício: Fazer com if/else depois trocar para operador ternário

Salario = 1000

Para salario > 1000 bônus de 10%

Senão bônus de 15%

Imprimir bônus

Iniciando no Java - Conceitos base da linguagem – Operador ternário

Exercício: Utilizar o if/else e depois trocar para o operador ternário



Iniciando no Java - Conceitos base da linguagem – Operador ternário

Operador ternário: Resolução

```
double salario = 1000;  
double bonus = 0.0;  
  
if (salario > 1000) { bonus = salario * 0.10; }  
else { bonus = salario * 0.15; }  
  
System.out.println(bonus);
```

```
double salario = 1000;  
double bonus = salario * (salario > 1000 ? 0.10 : 0.15);  
System.out.println(bonus);
```


Iniciando no Java - Conceitos base da linguagem – Estruturas de controle switch case

Switch case

O switch case é uma estrutura de seleção semelhante ao if com múltiplas seleções. Uma alternativa aos vários if/else aninhados. É uma estrutura muito fácil de utilizar e apresenta uma ótima legibilidade. Porém, trabalha apenas com valores constantes dos tipos primitivos **byte, short, int e char**. Usar com variáveis **boolean, long, double, float** e um objeto resultará em erro de compilação. É possível também enumerar n possíveis blocos de instrução.

Iniciando no Java - Conceitos base da linguagem – Estruturas de controle switch case

Switch case

Sua utilização deve ser feita da seguinte maneira:

```
switch( variável ) {  
    case <possível valor da constante> :  
        < instruções>  
        break;  
  
    case <possível valor da constante> :  
        < instruções>  
        break;  
  
    default:  
        < instruções>  
        break;  
}
```


Iniciando no Java - Conceitos base da linguagem – Estruturas de controle switch case

Switch case

Cada case é um caso no qual os comandos dentro dele são executados se o valor dele for o mesmo que a variável recebida no switch.

É importante lembrar que a utilização do comando break é facultativa, porém indispensável caso se queira que apenas aquele bloco seja executado e não todos os demais abaixo dele.

O bloco de comandos default representa uma condição geral de execução caso nenhuma das anteriores tenha sido atendida, sendo a sua utilização também opcional.

Iniciando no Java - Conceitos base da linguagem – Estruturas de controle switch case

Switch case

Cada case é um caso no qual os comandos dentro dele são executados se o valor dele for o mesmo que a variável recebida no switch.

É importante lembrar que a utilização do comando break é facultativa, porém indispensável caso se queira que apenas aquele bloco seja executado e não todos os demais abaixo dele.

O bloco de comandos default representa uma condição geral de execução caso nenhuma das anteriores tenha sido atendida, sendo a sua utilização também opcional.

Iniciando no Java - Conceitos base da linguagem – Estruturas de controle switch case

```
/**
 * Exemplo de estrutura de seleção SWITCH
 */
public class ExemploSwitch {
    public static void main(String[] args) {
        char nota = 'D';

        switch(nota) {
            case 'A':
                System.out.println("Aluno aprovado. Conceito excelente!");
                break;
            case 'B':
                System.out.println("Aluno aprovado. Conceito bom!");
                break;
            case 'C':
                System.out.println("Aluno aprovado. Conceito medio!");
                break;
            default:
                System.out.println("Aluno reprovado!");
                break;
        }
    }
}
```


Iniciando no Java - Conceitos base da linguagem – Estruturas de Repetições

Switch case

Exercício pratica

Escrever um programa que ao digitar o dia da semana de 1 a 7 imprima o nome do dia da semana.

Iniciando no Java - Conceitos base da linguagem – Estruturas de Repetições laços

EXERCÍCIOS