Estruturas de Controle e Decisão

Prof. Ítalo Assis

Ajude a melhorar este material =]

Encontrou um erro? Tem uma sugestão?

Envie e-mail para italo.assis@ufersa.edu.br

Agenda

- Instruções *if / else*
- Operador ternário
- Instrução *switch / case*

Estruturas condicionais

- O processamento de dados feito em uma aplicação muito frequentemente envolve decisões que devem ser feitas em relação a estes dados
- A capacidade de tomar estas decisões de forma padronizada e programável é uma das razões que faz o processamento de dados por computadores útil e interessante
- Exemplo:
  - Se o saldo for menor que zero, o usuário não poderá retirar dinheiro da conta

Condicionais em Java

- Todas as estruturas de controle de fluxo de um programa são baseadas em **condições**:
  - Estruturas de decisão executarão parte do código se uma condição ocorrer ou não
  - Estruturas de repetição repetirão trechos de código até que uma condição seja cumprida ou enquanto uma condição for válida
- Para construir essas condições, utilizamos **operadores relacionais**
- Operadores relacionais são utilizados para comparar valores
  - As condições têm a forma genérica: *valor operador valor*
    - Exemplo: *delta < 0*
  - São convertidas em um valor *true* ou *false*

Operadores relacionais em Java

- Os operadores da tabela podem ser usados para comparar valores de tipos nativos numéricos
- Para comparar objetos do tipo *String*, podemos utilizar o método *equals*:

  str1.equals(str2)

==	igualdade
!=	diferença
>	maior que
<	menor que
>=	maior ou igual a
<=	menor ou igual a

Operadores lógicos em Java

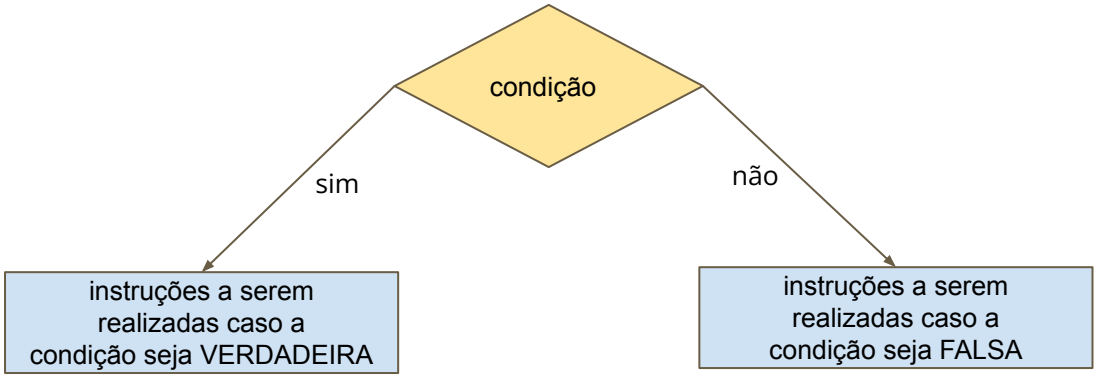
- Valores do tipo *boolean* e resultados de operações ou métodos que retornem valores booleanos podem ser combinados entre si através dos operadores lógicos:

&&	operador de conjunção (E)
II	operador de disjunção (OU)

- Exemplo: *((a >= b) && (a>=c))*
- Há também o operador de negação **!** (NÃO) que inverte o valor de um booleano
  - *!(2>3)* retorna *true*

Estruturas condicionais em Java

- Utilizados para tomar decisões
  - *if ... else ...*
  - Operador ternário
  - *switch*



Instruções *if ... else ...*

- Permite que um bloco de comandos seja executado dependendo do resultado de uma condição
- Se a expressão associada à palavra chave *if* for igual a *true*, o bloco de comandos associado ao *if* será executado
- Caso contrário, será executado o bloco de comandos associado ao *else*
- É possível omitir o *else*

```
if (condição) {
  // instruções a serem
  // realizadas caso a
  // condição seja VERDADEIRA
} else {
  // instruções a serem
  // realizadas caso a
  // condição seja FALSA
}
```

Exemplo

- Escreva um programa que, dadas as coordenadas cartesianas x e y dos centros de dois círculos e os tamanhos de seus respectivos raios, indique se eles estão colidindo

Instruções *if ... else ...* aninhadas

- Instruções *if-else* podem ser aninhadas, isto é, os blocos de execução associados ao *if* ou ao *else* podem conter outras instruções *if-else*

```
if (condição1) {
  // condição1 VERDADEIRA
  if (condição2) {
    // condição1 VERDADEIRA
    // condição2 VERDADEIRA
  } else {
    // condição1 VERDADEIRA
    // condição2 FALSA
  }
} else {
  // condição1 FALSA
  if (condição3) {
    // condição1 FALSA
    // condição3 VERDADEIRA
  } else {
    // condição1 FALSA
    // condição3 FALSA
  }
}
```

Exemplo

- Crie um programa que informa quantas raízes reais uma equação de 2º grau possui e quais são

$$ax^2 + bx + c = 0$$

Instruções *if... else...* em cascata

- Blocos de *if-else* podem ser dispostos em cascata, de forma que se uma condição não for satisfeita em um *if*, a seguinte será avaliada e assim em diante.

```
if (condição 1) {
  // condição 1 VERDADEIRA
} else {
  if (condição 2) {
    // condição 1 FALSA
    // condição 2 VERDADEIRA
  } else {
    if (condição 3) {
      // condições 1 e 2 FALSAS
      // condição 3 VERDADEIRA
    } else {
      // condições 1, 2 e 3 FALSAS
    }
  }
}
```

==

```
if (condição 1) {
  // condição 1 VERDADEIRA
} else if (condição 2) {
  // condição 1 FALSA
  // condição 2 VERDADEIRA
} else if (condição 3) {
  // condições 1 e 2 FALSAS
  // condição 3 VERDADEIRA
} else {
  // condições 1, 2 e 3 FALSAS
}
```

Exemplo

- Verifique se um ponto fornecido pelo usuário está dentro (sem contar as bordas), fora ou na borda de uma área retangular
  - O retângulo será definido por dois pontos:
    - Inferior esquerdo e superior direito
  - Os pontos são definidos por coordenada 2D no plano cartesiano (x,y)

Exemplo

- Escreva um programa que leia uma data com dia, mês e ano como números inteiros e retorne a data no formato a seguir:
  - “[dia] de [mês] de [ano]”
  - “9 de agosto de 2021”

Operador ternário

- Usado quando o objetivo de uma avaliação de expressão é simplesmente o de determinar que valor será atribuído a uma variável
- Sintaxe:
  - variavel = (condição ? valorCondVerdadeira : valorCondFalsa);
- Exemplo:
  - *int taxa = (salario > 3000 ? 20 : 15);*

Exemplo

- Escreva um programa que recebe dois números inteiros e informa:
  - O maior entre os dois números
  - O menor entre os dois números

Instrução *switch*

- A instrução *switch* permite que um valor inteiro seja avaliado e, dependendo do valor, o fluxo do programa será modificado para uma posição específica
- Mesmo funcionamento de blocos *if-else* em cascata

```
switch (número) {
  case valor1:
    // comandos caso número == valor1
    break;
  case valor2:
    // comandos caso número == valor2
    break;
  [...]
  default:
    // comandos caso número seja
    //diferente dos valores testados
    break;
}
```

Instrução *switch*

- A instrução *switch* transfere o fluxo do método em execução diretamente para o *case* correspondente, mas não controla o fluxo do programa depois que os comandos associados com o *case* são executados
- O comando *break* faz com que o fluxo de execução seja transferido para a primeira linha depois do bloco *case*

```
switch (número) {
  case valor1:
    // comandos caso número == valor1
    break;
  case valor2:
    // comandos caso número == valor2
    break;
  [...]
  default:
    // comandos caso número seja
    //diferente dos valores testados
    break;
}
```

Exemplo

- Reescreva o exemplo da data utilizando o comando *switch*
- Além disso, informe quantos dias tem o mês da data informada

Os códigos relacionados a esta aula estão disponíveis em

https://github.com/italoaug/Programacao-Orientada-a-Objetos/tree/main/codigos/condicionais

Referências

SANTOS, R. **Introdução à programação orientada a objetos usando JAVA**. 2. ed. Rio de Janeiro: Campus, 2013. 336p.