

# Estruturas condicionais

## if/else

Estrutura que permite ao programador condicionar a execução do código conforme as instruções definidas. Se a execução da condicional apresentar valor verdadeiro/true executa a primeira instrução, logo abaixo da declaração. Caso a execução retorne um valor falso/false executa, se houver, as instruções definidas abaixo do else, no segundo bloco de código.

## Exemplificando

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

Sobre o if, temos vários detalhes que lhe permitem aprimorar o uso da estrutura condicional elevando o nível do código:

- IF - Expressão booleana
  - Retorno sempre é um boolean. Logo a condicional de um int é true, de um objeto definido é true também;
  - Testar uma variável null da erro no código, null pointer;
  - Para testar mais de uma condicional ao mesmo tempo, temos 2 modificadores, condicional e, representada por &&, e condicional ou, representada por ||. Muito cuidado, no uso;
  - Podemos “negar” uma condicional, utilizando como prefixo ! (exclamação). Dessa forma, se o resultado da expressão é **true**, com a negação se torna **false**;
- Bloco de código 1 - se retorno positivo/true
  - Pode conter somente esse bloco de código, sem o comando ELSE;
  - Caso o bloco de código tenha somente uma linha de código, não é preciso adicionar as {} (chaves);
  - Existem comandos importantes que podem ser utilizados dentro do if/else

- return → finaliza a execução da função;
- continue → finaliza a execução ao final da estrutura condicional;
- ELSE
  - Não precisa ser definido, ou seja, o segundo bloco de código não necessariamente precisa existir;
  - Caso o bloco de código tenha somente uma linha de código, não é preciso adicionar as {} (chaves);
  - Existem comandos importantes que podem ser utilizados dentro do if/else
    - return → finaliza a execução da função;
    - continue → finaliza a execução ao final da estrutura condicional;

Alguns exemplos:

```
int x = 10;
if (x > 5) {
    System.out.println("x é maior que 5");
} else {
    System.out.println("x é menor ou igual a 5");
}
```

```
String language = "Java";
if (language.equals("Java")) {
    System.out.println("Você está aprendendo Java!");
} else {
    System.out.println("Você está aprendendo outra linguagem.");
}
```

## else if

Esse comando é utilizado quando queremos testar mais condições em sequência na mesma estrutura, ou seja, quando é necessário entrar em um bloco de código antes de dar sequência ao resto do código.

O else if utiliza todas as definições apresentadas anteriormente, é apenas uma extensão, para dar novas possibilidades ao código.

## Exemplificando

```
int x = 10;
int y = 20;
if (x > y) {
    System.out.println("x é maior que y");
} else if (x < y) {
    System.out.println("y é maior que x");
} else {
    System.out.println("x é igual a y");
}
```

```
String nome = "Beto";

if (nome.equals("João")) {
    System.out.println("Bem-vindo, João!");
} else if (nome.equals("Maria")) {
    System.out.println("Bem-vinda, Maria!");
} else if (nome.equals("Pedro")) {
    System.out.println("Bem-vindo, Pedro!");
} else {
    System.out.println("Usuário desconhecido.");
}
```

## Operador ternário

É exatamente como o if/else e else if, tem todos os recursos, sua existencia dá-se somente pela praticidade de escrever em apenas uma linha. Sim, tudo em uma linha.

Nesse caso, vamos direto de exemplos, pois já conhecemos todos os detalhes da estrutura condicional if/else.

```
int idade = 18;
String status = (idade >= 18) ? "adulto" : "adolescente";
```

```
int nota = 70;
String resultado = (nota >= 60) ? ((nota >= 90) ? "A" : "B") : "C";
```

## Switch/case

Diferente do if/else, no switch case a entrada é uma variável não null, de alguns tipos primitivos de dados, enumerações (enums), a classe String e algumas classes dos tipos primitivos.

Em sua estrutura podemos ter quantos blocos de código forem necessários, respeitando as regras de boas práticas, claro. Para cada bloco é feita uma validação e somente acessa o trecho de código caso a validação seja verdadeira.

## Exemplificando

```
switch (expressão) {  
    case valor1:  
        // bloco de código que será executado  
        break;  
    case valor2:  
        // bloco de código que será executado  
        break;  
    case valorN:  
        // bloco de código que será executado  
        break;  
    default:  
        // bloco de código que será executado se nenhum dos cases for aceito  
}
```

No switch/case temos alguns detalhes que são bem importantes para aplicar códigos mais avançados e deixar mais limpo a estrutura, dentro das necessidades:

- switch
  - Recebe por parâmetro uma variável, não é recebe uma comparação;
  - Informar um valor não null;
- case
  - É o comparador lógico, sempre que a expressão presente no case for verdadeira o bloco de código é executado;
  - O bloco case é executado até encontrar o comando break;
  - Pode não haver comando break ao final de um bloco case, dessa forma é feita a comparação com os demais blocos case existentes;
  - O comando default é responsável por executar o código case nenhuma condição presente em case seja verdadeira;
  - O comando break dentro do bloco default não é obrigatório;

## Exemplificando

```
int diaDaSemana = 3;  
String nomeDoDia;
```

```

switch (diaDaSemana) {
    case 1:
        nomeDoDia = "Domingo";
        break;
    case 2:
        nomeDoDia = "Segunda-feira";
        break;
    case 3:
        nomeDoDia = "Terça-feira";
        break;
    case 4:
        nomeDoDia = "Quarta-feira";
        break;
    case 5:
        nomeDoDia = "Quinta-feira";
        break;
    case 6:
        nomeDoDia = "Sexta-feira";
        break;
    case 7:
        nomeDoDia = "Sábado";
        break;
    default:
        nomeDoDia = "Dia inválido";
        break;
}

System.out.println("O dia " + diaDaSemana + " é " + nomeDoDia);

```

```

String nomeDoMes = "maio";
int diasNoMes;

switch (nomeDoMes) {
    case "janeiro":
    case "março":
    case "maio":
    case "julho":
    case "agosto":
    case "outubro":
    case "dezembro":
        diasNoMes = 31;
        break;
    case "abril":
    case "junho":
    case "setembro":
    case "novembro":
        diasNoMes = 30;
        break;
    case "fevereiro":
        diasNoMes = 28;
        break;
    default:
        diasNoMes = 0;
        break;
}

```

```
}  
  
System.out.println("O mês de " + nomeDoMes + " tem " + diasNoMes + " dias");
```

---

## Referências e links úteis

Documentação oficial if/else

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>

Documentação oficial switch/case

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>

Exercícios

<https://github.com/AvilaDaSilva/exercicios-java-basico>