

Bytecode: A Arte Secreta do Java



Gabriel Ribeiro

Dominando os Seletores em Java

Com Exemplos Reais

Se você está começando no mundo da programação ou quer reforçar sua base em Java, entender como o código "toma decisões" e "repete ações" é essencial.

Neste eBook, você vai aprender os principais seletores da linguagem Java — como if, switch, for e outros — de forma clara, prática e sem enrolação.

Cada seletor é explicado com um exemplo real, para que você não apenas entenda o código, mas veja como ele se encaixa no dia a dia do desenvolvimento.

Vamos direto ao ponto. Boa leitura e bons códigos!



01

Decisão é Tudo: Controle de Fluxo

Decisão é Tudo

Controle de Fluxo com if, else e switch

O if permite que seu programa execute um bloco de código apenas se uma condição for verdadeira.

```
double preco = 150.0;
if (preco > 100) {
    System.out.println("Desconto aplicado!");
}
```

```
String cargo = "admin";
if (cargo.equals("admin")) {
    System.out.println("Acesso total liberado.");
} else {
    System.out.println("Acesso restrito.");
}
```

Com o else, você define um caminho alternativo caso a condição do if seja falsa.

O else if é usado quando você precisa verificar mais de uma condição diferente, uma após a outra.

```
int nota = 85;
if (nota ≥ 90) {
    System.out.println("Excelente");
} else if (nota ≥ 70) {
    System.out.println("Bom");
} else {
    System.out.println("Precisa melhorar");
}
```

```
int mes = 3;
switch (mes) {
    case 1: System.out.println("Janeiro"); break;
    case 2: System.out.println("Fevereiro"); break;
    case 3: System.out.println("Março"); break;
    default: System.out.println("Mês inválido");
}
```

O switch organiza melhor a lógica quando há vários valores fixos para comparar.

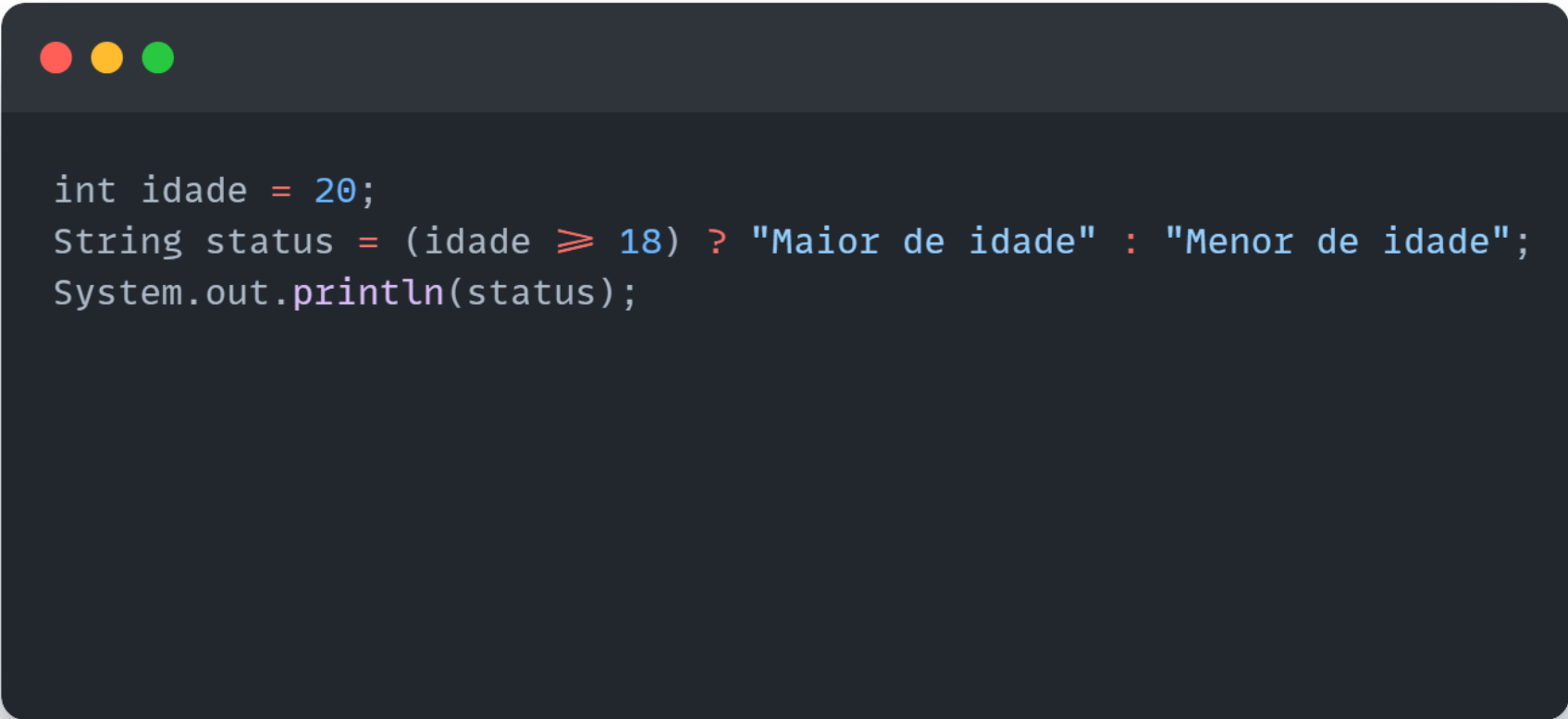
02

Pensando Simples: Decisão em Uma Linha com Operador Ternário

Pensando Simples

Decisão em Uma Linha com Operador Ternário

Esse operador substitui um if-else simples, permitindo uma decisão rápida em uma única linha.



```
int idade = 20;  
String status = (idade ≥ 18) ? "Maior de idade" : "Menor de idade";  
System.out.println(status);
```

03

Repetição com Propósito: Laços de Repetição em Java

Repetição com Propósito

Laços de Repetição em Java

O while repete um bloco de código enquanto a condição for verdadeira.

```
int contador = 1;
while (contador ≤ 5) {
    System.out.println("Contando: " + contador);
    contador++;
}
```

```
Scanner scanner = new Scanner(System.in);
String senha;
do {
    System.out.print("Digite a senha: ");
    senha = scanner.nextLine();
} while (!senha.equals("1234"));
System.out.println("Acesso concedido!");
```

O do-while executa pelo menos uma vez, pois a verificação acontece depois do bloco.

O for é ideal quando você sabe quantas vezes quer repetir algo.

```
for (int i = 1; i ≤ 10; i++) {
    System.out.println("Par: " + (i * 2));
}
```

```
String[] produtos = {"Notebook", "Mouse", "Teclado"};
for (String item : produtos) {
    System.out.println("Produto: " + item);
}
```

O for-each é a forma mais simples de percorrer todos os elementos de um array ou lista.

AGRADECIMENTOS

OBRIGADO POR LER ATÉ AQUI

Esse conteúdo foi gerado com fins didáticos de construção, não foi realizado uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.

<https://github.com/gabrielribv/ebook-bytecode-java>