



# Características das Linguagens de Programação I

**Conteúdo:** Comandos de seleção - parte II

Prof. Dsc. Giomar Sequeiros  
[giomar@eng.uerj.br](mailto:giomar@eng.uerj.br)

# Comando switch

# Estruturas de seleção (switch)

- É usada para tomar decisões com base no valor de uma única expressão. Sintaxe básica:

```
switch (expressao) {  
    case valor1:  
        // bloco de código para valor1  
        break;  
    case valor2:  
        // bloco de código para valor 2  
        break;  
    //...  
    case valorN:  
        // bloco de código para valorN  
        break;  
    default:  
        // bloco de código para qualquer outro valor  
        break;  
}
```

A palavra-chave **break** é usada para sair do switch após o bloco de código ser executado

- A **expressão** é **avaliada** e seu **resultado** é comparado com cada um dos **valores** especificados nos casos (case).
- Se a expressão **corresponder** a um dos valores (valor1, valor2, ..., valorN), o bloco de **código** correspondente a esse caso é **executado**.
- Se a expressão **não corresponder** a nenhum dos valores especificados nos casos, o bloco de código do caso **default** é **executado** (se houver um).

# Exemplo 1

```
Scanner scanner = new Scanner(System.in);  
System.out.print("Digite uma opção (1, 2 ou 3): ");  
int opcao = scanner.nextInt();
```

```
import java.util.Scanner;
```

Crie uma classe e no método main insira o código mostrado:

```
switch (opcao) {  
    case 1:  
        System.out.println("Ação para opção 1.");  
        break;  
    case 2:  
        System.out.println("Ação para opção 2.");  
        break;  
    case 3:  
        System.out.println("Ação para opção 3.");  
        break;  
    default:  
        System.out.println("Opção inválida.");  
}  
scanner.close();
```

# Exemplo 1

- Exemplo anterior usando **if-else**

```
import java.util.Scanner;
```

```
if (opcao == 1) {  
    System.out.println("Ação para opção 1.");  
} else if (opcao == 2) {  
    System.out.println("Ação para opção 2.");  
} else if (opcao == 3) {  
    System.out.println("Ação para opção 3.");  
} else {  
    System.out.println("Opção inválida.");  
}
```

## Exemplo 2: Calculadora

Crie uma classe Calculadora e no método main insira o código abaixo:

```
import java.util.Scanner;
```

```
Scanner scanner = new Scanner(System.in);  
char operador;  
double num1, num2, resultado = 0;  
  
System.out.print("Digite a operação a ser realizada (+, -, *, /): ");  
operador = scanner.next().charAt(0);  
  
System.out.print("Digite o primeiro número: ");  
num1 = scanner.nextDouble();  
  
System.out.print("Digite o segundo número: ");  
num2 = scanner.nextDouble();
```

## Exemplo 2: Calculadora (cont.)

```
switch (operador) {  
    case '+':  
        resultado = num1 + num2;  
        break;  
    case '-':  
        resultado = num1 - num2;  
        break;  
    case '*':  
        resultado = num1 * num2;  
        break;
```



```
        case '/':  
            if (num2 != 0) {  
                resultado = num1 / num2;  
            } else {  
                System.out.println("Erro: Divisão por zero.");  
                scanner.close();  
            }  
            break;  
        default:  
            System.out.println("Operador inválido.");  
            scanner.close();  
    }
```

```
System.out.printf("%.2f %c %.2f = %.2f\n", num1, operador, num2, resultado);  
scanner.close();
```



# Exemplo 2: Calculadora

- Exemplo anterior usando **if-else**

```
if (operador == '+') {  
    resultado = num1 + num2;  
} else if (operador == '-') {  
    resultado = num1 - num2;  
} else if (operador == '*') {  
    resultado = num1 * num2;  
} else if (operador == '/') {  
    if (num2 != 0) {  
        resultado = num1 / num2;  
    } else {  
        System.out.println("Erro: Divisão por zero.");  
        scanner.close();  
    }  
} else {  
    System.out.println("Operador inválido.");  
    scanner.close();  
}
```



# Exercício

---

- Escreva um programa que solicita ao usuário um número de 1 a 12 representando um mês do ano. O programa deve exibir o número de dias neste mês. Use um switch para realizar essa operação.

```
Digite o número do mês: 2  
O mês de fevereiro tem 28 dias.
```

- **Dica:** Lembre-se de que fevereiro tem 28 dias, a menos que seja um ano bissexto (divisível por 4, exceto por anos que são divisíveis por 100, a menos que sejam divisíveis por 400).

# Comando ternário

# Operador ternário

---

- O operador ternário é uma forma concisa de escrever uma instrução if-else simples. Ele tem a seguinte sintaxe:

```
condicao ? valorSeVerdadeiro : valorSeFalso;
```

Onde:

- **condicao**: Uma expressão booleana que é avaliada como true ou false.
- **valorSeVerdadeiro**: O valor ou expressão a ser retornado se a condição for verdadeira (true).
- **valorSeFalso**: O valor ou expressão a ser retornado se a condição for falsa (false).

# Operador ternário: Exemplo 1

- Verificando se um número é par ou ímpar:

```
import java.util.Scanner;
public class ParImpar{
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.print("Digite um número: ");
        int numero = scanner.nextInt();

        String resultado = (numero % 2 == 0) ? "Par" : "Ímpar";

        System.out.println("O número " + numero + " é " + resultado);
    }
}
```

# Operador ternário: Exemplo 2

- Verificando se um dado ano é bissexto

```
import java.util.Scanner;
public class Bissexto{
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.print("Digite um ano: ");
        int ano = scanner.nextInt();

        String bissexto = (ano % 4 == 0 && (ano % 100 != 0 || ano % 400 == 0)) ?
                           "Ano bissexto" : "Ano não bissexto";

        System.out.println(ano + " é " + bissexto);
        scanner.close();
    }
}
```

# Exercício

- A empresa de fornecimento de água resolveu incentivar a economia de água mudando a tabela de cobrança, de forma que proporcionalmente, os clientes que consumirem menos água paguem menos pelo metro cúbico. Todo cliente paga mensalmente uma assinatura de R\$ 7,00, que inclui uma franquia de 10 m<sup>3</sup> de água. Isso significa que para qualquer consumo entre 0 e 10 m<sup>3</sup>, o consumidor paga a mesma quantia de R\$ 7,00 (note que o valor da assinatura deve ser pago mesmo que o consumidor não tenha consumido água). Acima de 10 m<sup>3</sup>, cada metro cúbico subsequente tem um valor diferente, dependendo da faixa de consumo. A tabela a seguir especifica o preço por metro cúbico para cada faixa de consumo:

Faixa de consumo (m <sup>3</sup> )	Preço (por m <sup>3</sup> )
até 10	incluído na franquia
11 a 30	R\$1,00
31 a 100	R\$2,00
101 em diante	R\$5,00

## Exercício (cont.)

- Assim, por exemplo, se o consumo foi de  $120 \text{ m}^3$ , o valor da conta é:

- R\$ 7,00 da assinatura básica;
- R\$ 20,00 pelo consumo no intervalo de 11 a  $30 \text{ m}^3$ ;
- R\$ 140,00 pelo consumo no intervalo de 31 a  $100 \text{ m}^3$ ;
- R\$ 100,00 pelo consumo no intervalo de 101 a  $120 \text{ m}^3$ .

Faixa de consumo ( $\text{m}^3$ )	Preço (por $\text{m}^3$ )
até 10	incluído na franquia
11 a 30	R\$1,00
31 a 100	R\$2,00
101 em diante	R\$5,00

- Logo, o valor total da conta de água é R\$ 267,00.

- Escreva um **programa** que receba como entrada o consumo de água de uma residência em  $\text{m}^3$ , calcule e retorne o valor da conta de água daquela residência.

*Exemplo de entrada/saída: 250 / R\$917,00*

*Exemplo de entrada/saída: 99 / R\$165,00*



# Referências

## BIBLIOGRAFIA BÁSICA:

- DEITEL, Harvery M.. Java : como programar. 10ª ed. São Paulo: Pearson - Prentice Hall, 2017.
- BORATTI, Isaías Camilo. Programação Orientada a Objetos em Java : Conceitos Fundamentais de Programação Orientada a Objetos. 1ª ed. Florianópolis: VisualBooks, 2007.
- SIERRA, Kathy; BATES, Bert. Use a Cabeça! Java. 2ª ed. Rio de Janeiro: Alta Books, 2007.

