

**INSTITUTO
FEDERAL**
Santa Catarina

Estruturas de Decisão (parte 2)

Pensamento Computacional e Algoritmos

Análise e Desenvolvimento de Sistemas

Prof. Adriano Lima

adriano.lima@ifsc.edu.br



INSTITUTO FEDERAL

Santa Catarina

Câmpus São José

PENSAMENTO
COMPUTACIONAL E
ALGORITMOS

Análise e Desenvolvimento
de Sistemas

cores

#111027

#277756

#16ABCD

#FFF4EC

#C74E23

fontes

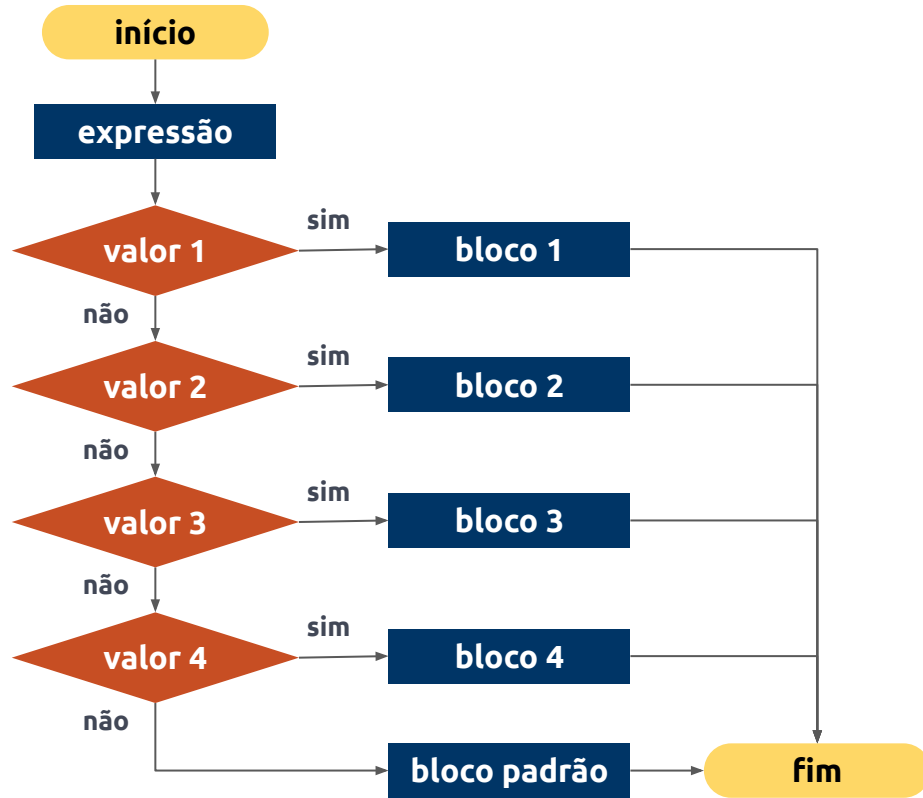
Fira Sans Extra Condensed

Ubuntu

Roboto Mono



switch-case



- "caixa de comutação"
- permite a seleção entre dois ou mais caminhos distintos para execução
- o valor de uma expressão é verificado em uma lista de constantes
- quando a ocorrência é encontrada o bloco de instruções associado é executado
- melhor legibilidade em comparação à estrutura if-else

switch-case

sintaxe

```
switch (expressão) {  
    case valor1:  
        // bloco 1 de instruções  
        break;  
    case valor2:  
        // bloco 2 de instruções  
        break;  
    case valor3:  
        // bloco 3 de instruções  
        break;  
    default:  
        // bloco padrão de instruções  
        break;  
}
```

- cada valor em case deve ser uma expressão de constante exclusiva

switch-case

sintaxe

```
switch (expressão) {  
    case valor1:  
        // bloco 1 de instruções  
        break;  
    case valor2:  
        // bloco 2 de instruções  
        break;  
    case valor3:  
        // bloco 3 de instruções  
        break;  
    default:  
        // bloco padrão de instruções  
        break;  
}
```

- cada valor em case deve ser uma expressão de constante exclusiva
- o tipo do valor deve ser compatível com o tipo da expressão

switch-case

sintaxe

```
switch (expressão) {  
    case valor1:  
        // bloco 1 de instruções  
        break;  
    case valor2:  
        // bloco 2 de instruções  
        break;  
    case valor3:  
        // bloco 3 de instruções  
        break;  
    default:  
        // bloco padrão de instruções  
        break;  
}
```

- cada valor em case deve ser uma expressão de constante exclusiva
- o tipo do valor deve ser compatível com o tipo da expressão
- o bloco padrão (default) é executado quando nenhuma constante case coincide com a expressão (opcional)

switch-case

sintaxe

```
switch (expressão) {  
    case valor1:  
        // bloco 1 de instruções  
        break;  
    case valor2:  
        // bloco 2 de instruções  
        break;  
    case valor3:  
        // bloco 3 de instruções  
        break;  
    default:  
        // bloco padrão de instruções  
        break;  
}
```

- cada valor em case deve ser uma expressão de constante exclusiva
- o tipo do valor deve ser compatível com o tipo da expressão
- o bloco padrão (default) é executado quando nenhuma constante case coincide com a expressão (opcional)
- o comando break evita a execução dos blocos posteriores (opcional)

switch-case

exemplo

escrever por extenso o número escolhido

switch-case

exemplo escrever por extenso o número escolhido

```
switch (numero) {  
    case 1:  
        System.out.println("um");  
        break;  
    case 2:  
        System.out.println("dois");  
        break;  
    case 3:  
        System.out.println("três");  
        break;  
    default:  
        System.out.println("Número inválido");  
        break;  
}
```



Comentários



- texto ignorado pelo compilador
- utilidade
 - descrever ou explicar a operação de um trecho de código
 - remover um trecho de código do fluxo de execução

Comentários

tipos

- **comentários de uma linha**
 - começam com //
 - terminam quando termina a linha

exemplos

Comentários

tipos

- comentários de uma linha
 - começam com //
 - terminam quando termina a linha

exemplos

// Este é um comentário de uma linha

Comentários

tipos

- **comentários de uma linha**
 - começam com //
 - terminam quando termina a linha
- **comentários de múltiplas linhas**
 - começam com /*
 - começam com */

exemplos

// Este é um comentário de uma linha

Comentários

tipos

- comentários de uma linha
 - começam com //
 - terminam quando termina a linha
- comentários de múltiplas linhas
 - começam com /*
 - começam com */

exemplos

// Este é um comentário de uma linha

/ Este é um comentário de
múltiplas linhas */*

Entradas a Partir do Teclado

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        // Criando um objeto Scanner para ler entrada do teclado
        Scanner scanner = new Scanner(System.in);

        // Lendo uma linha de texto
        String texto = scanner.nextLine();
        System.out.println("Você digitou: " + texto);

        // Fechando o Scanner
        scanner.close();
    }
}
```



Entradas a Partir do Teclado

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        // Criando um objeto Scanner para ler entrada do teclado
        Scanner scanner = new Scanner(System.in);

        // Lendo um número inteiro
        int numero = scanner.nextInt();
        System.out.println("Você digitou: " + numero);

        // Fechando o Scanner
        scanner.close();
    }
}
```



Entradas a Partir do Teclado

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        // Criando um objeto Scanner para ler entrada do teclado
        Scanner scanner = new Scanner(System.in);

        // Lendo um número de ponto flutuante
        double numeroReal = scanner.nextDouble();
        System.out.println("Você digitou: " + numeroReal);

        // Fechando o Scanner
        scanner.close();
    }
}
```



Entradas a Partir do Teclado

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        // Criando um objeto Scanner para ler entrada do teclado
        Scanner scanner = new Scanner(System.in);

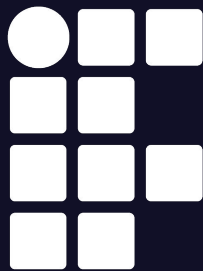
        // Lendo um caracter
        char letra = scanner.next().charAt(0);
        System.out.println("Você digitou: " + letra);

        // Fechando o Scanner
        scanner.close();
    }
}
```



Exercícios

Faça a lista de exercícios 7 disponível no SIGAA.



**INSTITUTO
FEDERAL**
Santa Catarina

Estruturas de Decisão (parte 2)

Pensamento Computacional e Algoritmos

Análise e Desenvolvimento de Sistemas

Prof. Adriano Lima

adriano.lima@ifsc.edu.br



INSTITUTO FEDERAL

Santa Catarina

Câmpus São José

PENSAMENTO
COMPUTACIONAL E
ALGORITMOS

Análise e Desenvolvimento
de Sistemas