

Switch case ->

O switch com case com setas ('->') foi introduzido a partir do Java 12 como *preview* e consolidado no Java 14, trazendo mais simplicidade e segurança no código. Ele pode ser usado tanto como comando (executar blocos) quanto como expressão (retornar valores).

1. Estrutura básica do switch case : (Switch Statement - *Declaração switch*)

```
switch (opcao) {  
    case 1:  
        System.out.println("Um");  
        break;  
    case 2:  
        System.out.println("Dois");  
        break;  
    default:  
        System.out.println("Outro número");  
}
```

Agora , a estrutura do switch case -> (Switch Expressions - *Expressões switch*).

```
switch (opcao) {  
    case 1 -> System.out.println("Um");  
    case 2 -> System.out.println("Dois");  
    default -> System.out.println("Outro número");  
}
```

Diferenças principais:

- Não precisa de break.
- Cada case não cai no próximo (evita *fall-through*).
- Pode usar como expressão (retornando valores).

2. Usando como expressão (retorno direto)

```
int dia = 3;  
String nomeDia;  
nomeDia = switch (dia) {  
    case 1 -> "Domingo";  
    case 2 -> "Segunda";  
    case 3 -> "Terça";  
    case 4 -> "Quarta";  
    case 5 -> "Quinta";  
    case 6 -> "Sexta";  
    case 7 -> "Sábado";  
    default -> "Dia inválido";  
};  
  
System.out.println("Hoje é " + nomeDia);
```



3. Agrupando vários casos

```
String letra = "a";
String tipo;
tipo = switch (letra) {
    case "a" , "e" , "i" , "o" , "u" -> "Vogal";
    default -> "Consoante";
};
System.out.println(letra + " é uma " + tipo);
```

4. Executando blocos de código

```
int mes = 2;
int dias;
dias = switch ( mês ) {
    case 1 , 3 , 5 , 7 , 8 , 10 , 12 -> 31;
    case 4 , 6 , 9 , 11 -> 30;
    case 2 -> {
        System.out.println("Fevereiro pode ter 28 ou 29 dias.");
        yield 28; // usa yield para retornar valor em blocos
    }
    default -> -1;
};
System.out.println("O mês " + mes + " tem " + dias + " dias.");
```

5. Exemplo prático – Calculadora simples

```
int a = 10, b = 5;
char operador = '+';
int resultado = switch (operador) {
    case '+' -> a + b;
    case '-' -> a - b;
    case '*' -> a * b;
    case '/' -> {
        if (b == 0) {
            System.out.println("Divisão por zero não permitida!");
            yield 0;
        }
        yield a / b;
    }
};
System.out.println("Resultado = " + resultado);
```



6. Exemplo real: Menu com várias opções

```
Scanner sc = new Scanner(System.in);

System.out.println("Escolha uma opção:");
System.out.println("1 - Cadastrar");
System.out.println("2 - Listar");
System.out.println("3 - Excluir");
System.out.println("0 - Sair");

int opcao = sc.nextInt();

switch (opcao) {
    case 1 -> System.out.println("Cadastro iniciado...");
    case 2 -> System.out.println("Listando itens...");
    case 3 -> System.out.println("Excluindo item...");
    case 0 -> System.out.println("Saindo...");
    default -> System.out.println("Opção inválida!");
}
```

O switch case -> torna o código mais limpo, evita erros de esquecer o *break*, **permite usar switch como expressão (retornando valores)** e usa yield em blocos para devolver valores.