



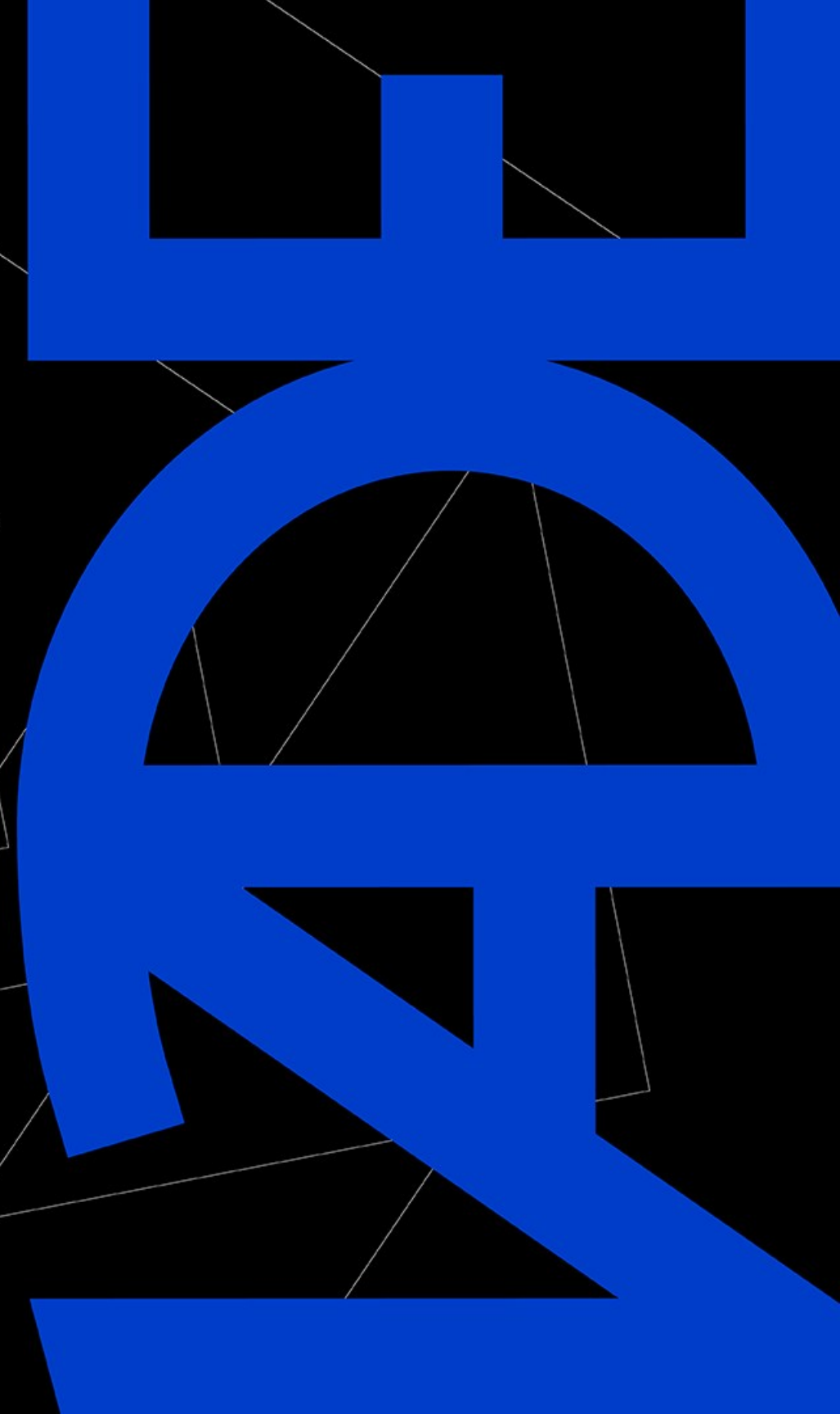
Faculdade de Design,
Tecnologia e Comunicação
 Universidade Europeia

Comandos de Seleção

Fundamentos da Programação

Fernando Marson

fernando.marson@universidadeeuropeia.pt



Fluxo de Execução

Nos programas desenvolvidos até agora, temos uma característica em comum: **a ordem de execução das instruções é sempre igual, todos os comandos são sempre executados e de forma linear**, do início ao fim do código.



Comandos de Seleção

- Utilizando os **comandos ou estruturas de seleção**, é possível definir se um **trecho de código será** ou **não será executado**, permitindo ainda definir **quais são as condições** que **executam** esse determinado trecho.



Comandos de Seleção

- A combinação desses comandos permite controlar o **fluxo de execução** em vários níveis diferentes, fazendo com que o programador consiga **simular comportamentos a partir de entradas do usuário** ou **através de resultados de cálculos e de expressões**.
- Cada linguagem de programação **oferece diferentes comandos** de seleção, podendo ter diferenças **inclusive em função da versão** da linguagem utilizada.

Comandos de Seleção

Em **Java** existem os seguintes comandos de seleção:

- **if** e **if / else**
- **operador condicional** **ou** **ternário**
- **switch / case**

Comandos de Seleção - if

Estes são alguns exemplos de **escolhas** ou **seleções** que executamos no nosso cotidiano:

- **se** *estiver chovendo*, usar o guarda-chuva.
- **se** *o sinal de trânsito estiver vermelho*, parar o carro.
- **se** *o alarme tocar*, acordar e levantar da cama.
- **se** *estiver frio na rua*, sair de casaco.

Comandos de Seleção - if - Funcionamento

- Com o comando de seleção **if** (**se**) é possível **executar** um trecho de código **se e somente se** uma determinada condição for **verdadeira**.
- Caso **não seja**, o fluxo **segue para a próxima instrução após o if**.



Comandos de Seleção - if - Sintaxe

A sintaxe do comando de seleção **if** (**condicao**) em **Java** é

```
1 if(condicao)
2    // uma instrução apenas
```

Se houver apenas um comando o uso de **{ }** para iniciar um bloco é opcional. A **tabulação em Java** é apenas **indicativa**.

```
1 if(condicao){
2    // várias instruções
3    // várias instruções
4    // várias instruções
5    // várias instruções
6 }
```

Se existirem **múltiplas instruções** a serem executadas pelo **if**, é **obrigatório** o uso de **{** antes da primeira instrução e **}** após a última instrução para definir o bloco.

Comandos de Seleção - if - Exemplo

```
1 import java.util.Scanner;
2
3 class Main {
4     public static void main(String[] args) {
5         Scanner teclado = new Scanner(System.in);
6
7         int idade;
8
9         System.out.print("Informe a sua idade: ");
10        idade = teclado.nextInt();
11        if(idade >= 18){
12            System.out.println("Maior de idade.");
13        }
14        System.out.println("Primeira instrução após o if");
15    }
16 }
```

```
Informe a sua idade: 18
Maior de idade.
Primeira instrução após o if
```

```
Informe a sua idade: 15
Primeira instrução após o if
```

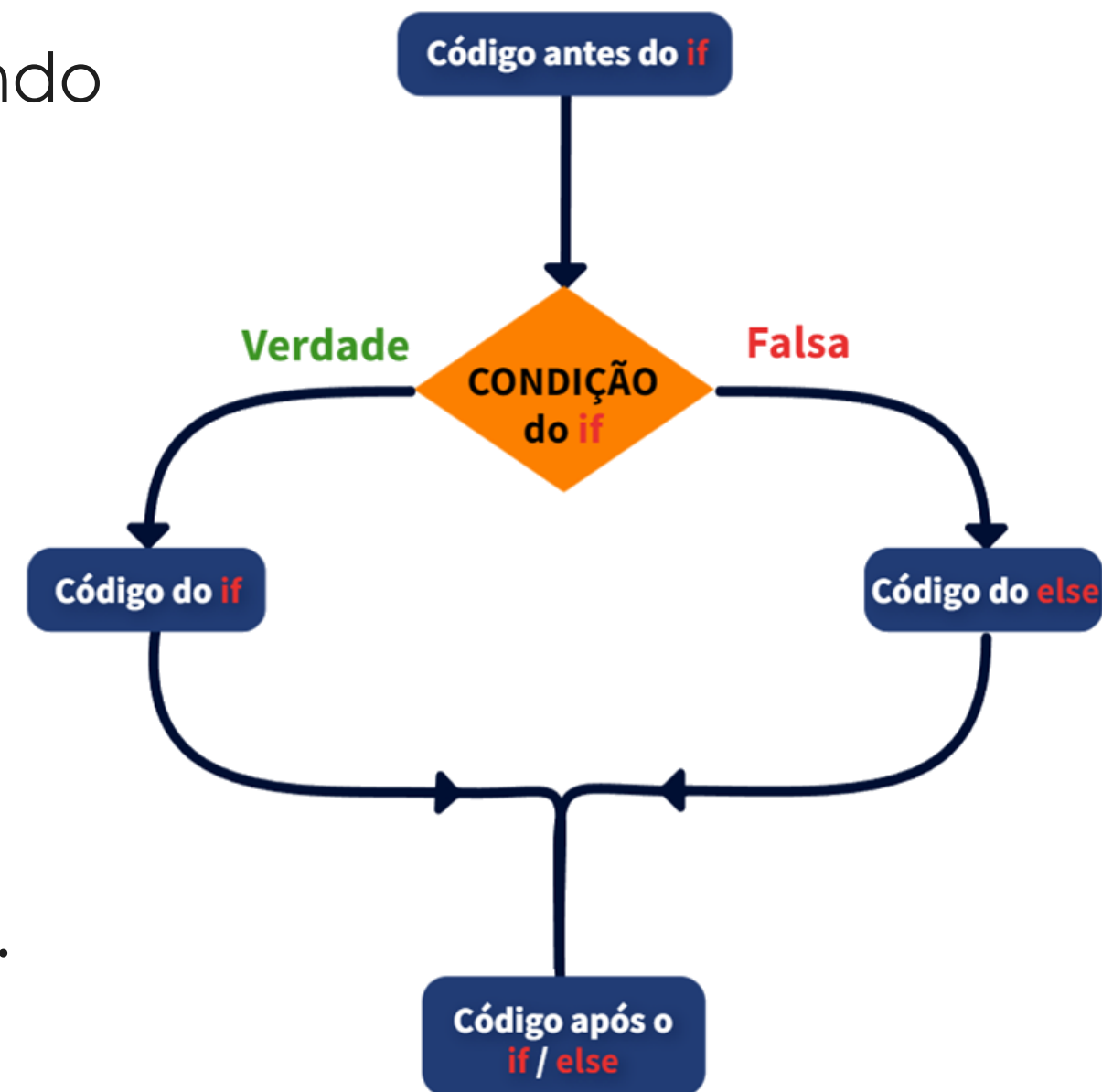
Comandos de Seleção - if/else

Nem sempre decidimos somente **sim** ou **não**, **fazer** ou **não fazer**, às vezes é possível **escolher entre algumas opções**, tais como:

- **se** *estiver chovendo*, usar o guarda-chuva, **senão** usar um boné.
- **se** *o sinal de trânsito estiver vermelho*, parar o carro, **senão se** *estiver amarelo*, diminuir a velocidade e atravessar com cuidado, **senão** continuar dirigindo.
- **se** *for dia de semana*, programar horário para acordar, **senão se** *for sábado*, sair para dançar, **senão** descansar.
- **se** *estiver frio na rua*, usar calça e casaco, **senão** usar bermuda e camiseta.

Comandos de Seleção - if/else - Funcionamento

- Para representar as situações anteriores e complementar comando **if**, temos o comando **else** (senão).
- Com o uso do **if** / **else** podemos dividir o fluxo de execução em **duas direções**, caso a condição seja **verdadeira** (**true**), serão executados os comandos do bloco do **if**.
- Se a condição for **falsa** (**false**) serão executados os comandos do bloco do **else**.



Comandos de Seleção - if/else - Sintaxe

- Se após o **if** ou após o **else** existir **apenas uma instrução**, é possível omitir o indicador de bloco **{ }** na parte que contiver apenas um comando.

```
1 if (condicao){  
2     // bloco de comandos a serem executados SE a condição for verdade  
3 }  
4 else{  
5     // bloco de comandos a serem executados SE a condição for falsa  
6 }
```

Comandos de Seleção - if/else - Exemplo

```
1 import java.util.Scanner;
2
3 class Main {
4     public static void main(String[] args) {
5         Scanner teclado = new Scanner(System.in);
6         int idade;
7
8         System.out.print("Informe a sua idade: ");
9         idade = teclado.nextInt();
10        if(idade >= 18){
11            System.out.println("Você é maior de idade.");
12        }
13        else{
14            System.out.println("Você é menor de idade.");
15        }
16    }
17 }
```

Comandos de Seleção - Encadeamento de if/else

Podemos **dividir o fluxo** em **mais do que duas possibilidades** através do uso encadeado de **if / else**, neste caso sempre que um **else** contiver um **if** depois utilizaremos **else if (condicao)**.

```
1 if (condicao1){
2   // bloco de comandos a serem executados se a condicao1 for verdade
3 }
4 else if (condicao2){
5   /* bloco de comandos a serem executados se a condicao1 for falsa
6    e a condicao2 for verdade */
7 }
8 else if (condicao3){
9   /* bloco de comandos a serem executados se as condições 1 e 2 forem
10  falsas e a condicao3 for verdade */
11 }
12 // quantas condições forem necessárias
13 else{
14   /* bloco de comandos a serem executados se nenhuma das condições
15  anteriores for verdade */
16 }
```

Comandos de Seleção - Encadeamento de if/else

```
1 import java.util.Scanner;
2 class Main {
3     public static void main(String[] args) {
4         Scanner teclado = new Scanner(System.in);
5         int numero;
6         System.out.print("Informe um número: ");
7         numero = teclado.nextInt();
8         if(numero < 10){
9             System.out.println("Número menor do que dez.");
10        }
11        else if (numero > 10){
12            System.out.println("Número maior do que dez.");
13        }
14        else
15            System.out.println("É o número dez.");
16    }
17 }
```

Comandos de Seleção - if/else - Regras

Qualquer **operador relacional** poderá ser utilizado dentro de um **if**: **<**, **<=**, **>**, **>=**, **==** ou **!=**.

Após um **if** ou após um **else** podemos ter um **comando único** ou um **bloco de comandos** entre **{ }**.

Diferentes **expressões relacionais** e/ou **variáveis booleanas** poderão ser **concatenadas** utilizando-se os **operadores lógicos** **&&** e **||**.

Comandos de Seleção - if/else - Regras

- O comando **else** é opcional, mas **só pode ser utilizado** caso **exista um comando if** acima, pois **não é possível** utilizar o comando **else** sozinho, **nem dois else** dependendo do mesmo **if**.
- Não é possível usar uma condição diretamente após um **else**.
- Condições só podem ser utilizadas após um **if**.

Comandos de Seleção - Operador Condicional

- O **operador condicional** ou **operador ternário** é uma das opções para realizar o **controle de fluxo** em **Java**, conforme a sintaxe

```
1 (expressão booleana) ? códigoVerdadeiro : códigoFalso;
```

- Inicialmente temos a **variável booleana** ou **expressão** que será avaliada.
- Após o **?** é informado o **código a ser executado** quando a avaliação retornar **true**.
- Separado por **:** está o **código que será executado** quando a avaliação retornar **false**.
- O **uso de parênteses não é obrigatório**, mas **facilita a leitura**.

Comandos de Seleção - Operador Condicional

```
1 import java.util.Scanner;
2 class Main {
3     public static void main(String[] args) {
4         Scanner teclado = new Scanner(System.in);
5         System.out.println("Informe a sua velocidade: ");
6         int velocidade = teclado.nextInt();
7         System.out.printf("Você %s de velocidade.", (velocidade > 60) ? "foi multado por excesso" : "estava dentro do limite");
8     }
9 }
```

Comandos de Seleção - switch / case

- O comando **switch / case** é uma **alternativa** às estruturas do tipo **if / else if / else**.
- Possui uma **estrutura mais simples** de controle de fluxo, tornando o **código mais legível** e **menos propenso a erros**.

Comandos de Seleção - switch / case

- Os **case** do **switch** / **case** devem ser sempre do mesmo tipo em função de **Java** ser fortemente tipado.
- Esse comando **permite utilizar variáveis** de controle do tipo **byte**, **short**, **int**, **char**, **enumerações** e **String**.
- **Não é possível** usar **boolean**, **float**, **double** ou **long**.
- Se for utilizado o tipo **char** na **variável de controle**, o **valor do case** deve ser entre **aspas simples**, por exemplo **'A'** e no caso do uso de **String** entre **aspas duplas**, como **"teste"**.

Comandos de Seleção - switch / case

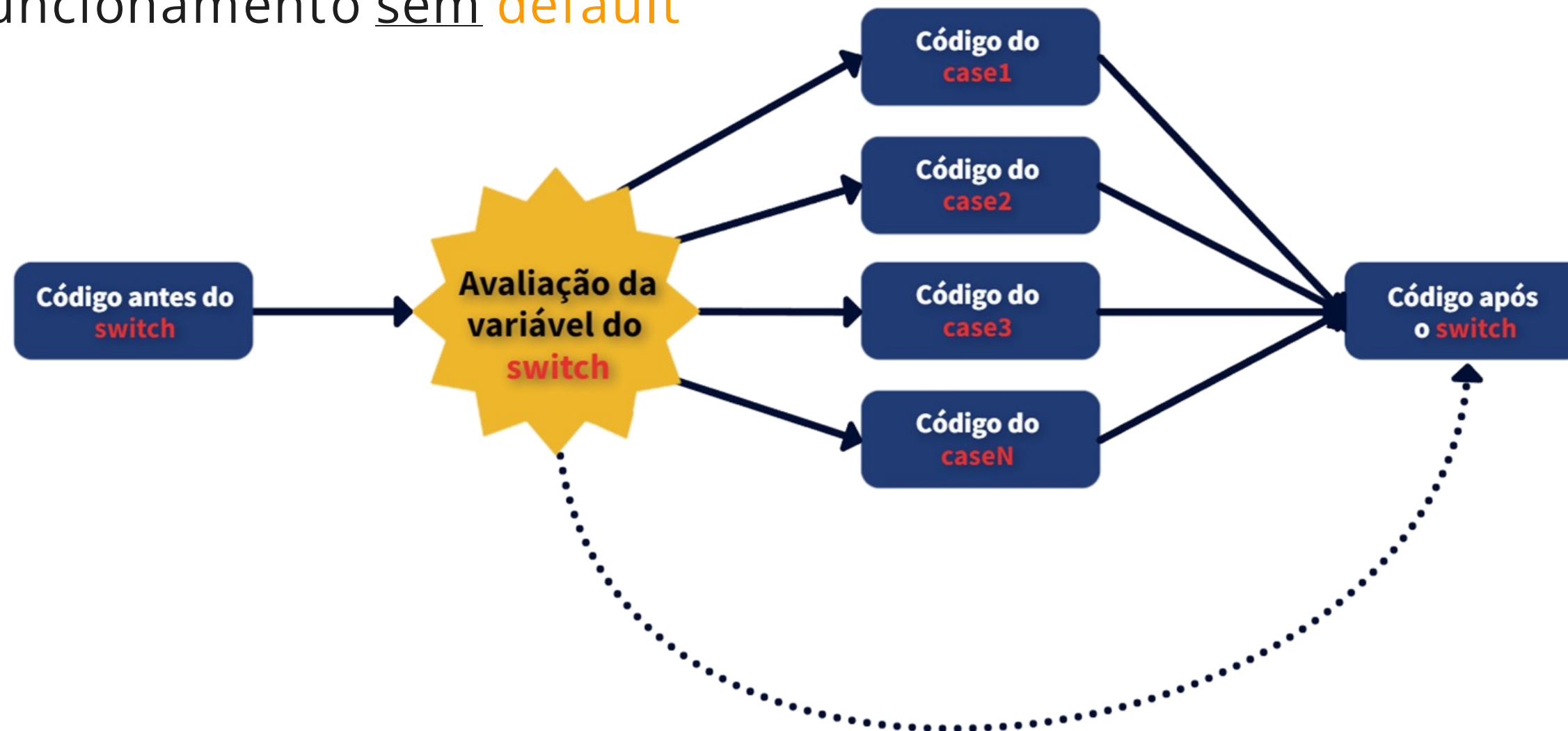
- Não existem blocos com `{ }` no **switch**, por isso o comando **break** é utilizado para finalizar a execução dos comandos do **case** em questão e sair do **switch**.

```
1 switch(variavel){
2     case VALOR1: // comandos para caso a variável seja igual a VALOR1
3         break;
4     case VALOR2: // comandos para caso a variável seja igual a VALOR2
5         break;
6     case VALOR3: // comandos para caso a variável seja igual a VALOR3
7         break;
8     case VALORN: // comandos para caso a variável seja igual a VALORN
9         break;
10
11     default: // comandos para caso a variável não seja nenhum dos valores acima
12 }
```

- A instrução **default** é **opcional** e serve para executar comandos quando o conteúdo da variável equivale à nenhum **case** do **switch**.

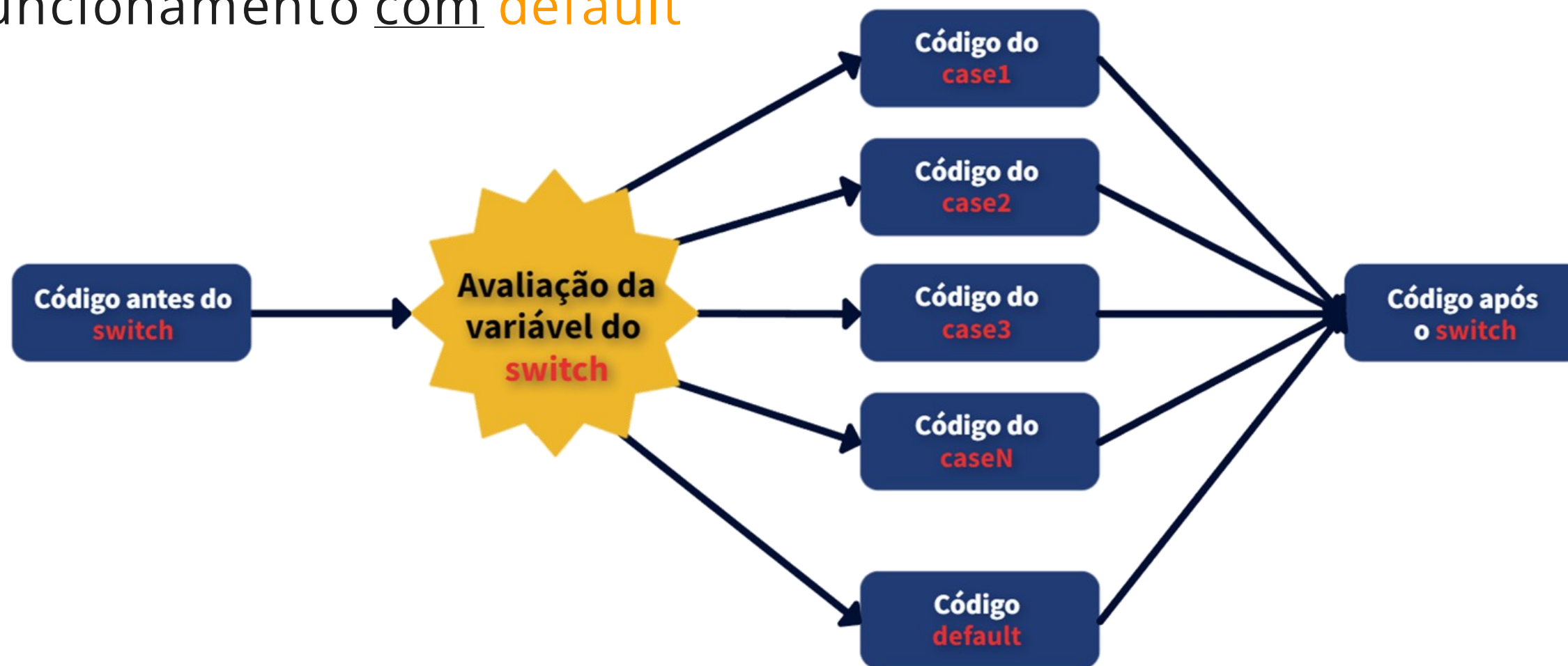
Comandos de Seleção - switch / case

Funcionamento sem default



Comandos de Seleção - switch / case

Funcionamento com default



Comandos de Seleção - switch / case

- Se desejado, é possível omitir o **break** entre dois **cases**, fazendo com que dois valores diferentes executem o mesmo conjunto de instruções.

```
1 switch(variavel){  
2     case VALOR1: // comandos para caso a variável seja igual a VALOR1  
3         break;  
4     case VALOR2: break;  
5     case VALOR3: // comandos para caso a variável seja igual a VALOR2 ou VALOR3  
6         break;  
7     case VALORN: // comandos para caso a variável seja igual a VALORN  
8         break;  
9  
10    default: // comandos para caso a variável não seja nenhum dos valores acima  
11 }
```

Comandos de Seleção - switch / case

Exemplo - char

```
1 class Main {  
2     public static void main(String[] args) {  
3         char letra = 'C';  
4  
5         switch(letra){  
6             case 'A': System.out.print("Letra A");  
7                 break;  
8             case 'B': System.out.print("Letra B");  
9                 break;  
10            default: System.out.print("Nem A nem B.");  
11        }  
12    }  
13 }
```

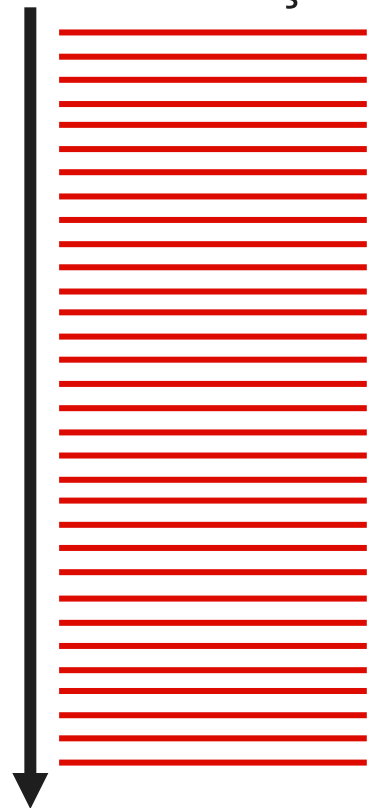
Comandos de Seleção - switch / case

Exemplo - `int`

```
1 import java.util.Scanner;
2
3 class Main {
4     public static void main(String[] args) {
5         int numero;
6         Scanner teclado = new Scanner(System.in);
7
8         System.out.print("Escolha um dos seguintes números (8, 13, 42 ou 75): ");
9         numero = teclado.nextInt();
10
11         switch(numero){
12             case 8: System.out.println("O número 8 foi escolhido.");
13                 break;
14             case 13: System.out.println("Número 13 selecionado.");
15                 break;
16             case 42: System.out.println("42 é o seu número.");
17                 break;
18             case 75: System.out.println("O seu número é 75.");
19                 break;
20             default: System.out.println(numero + " não é uma escolha válida.");
21         }
22     }
23 }
```

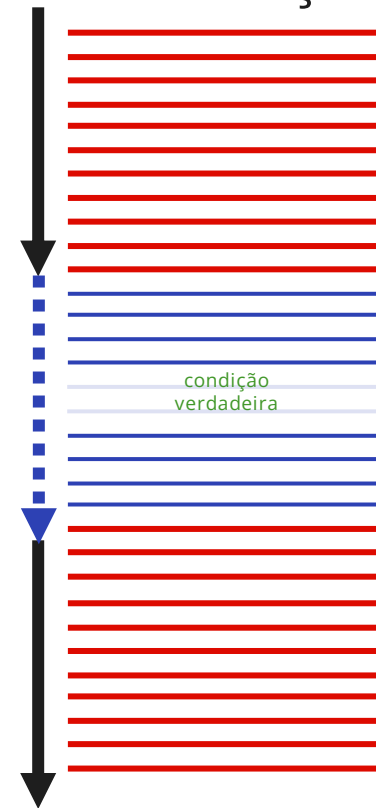
Resumo dos Fluxos de Execução

Fluxo único
de execução



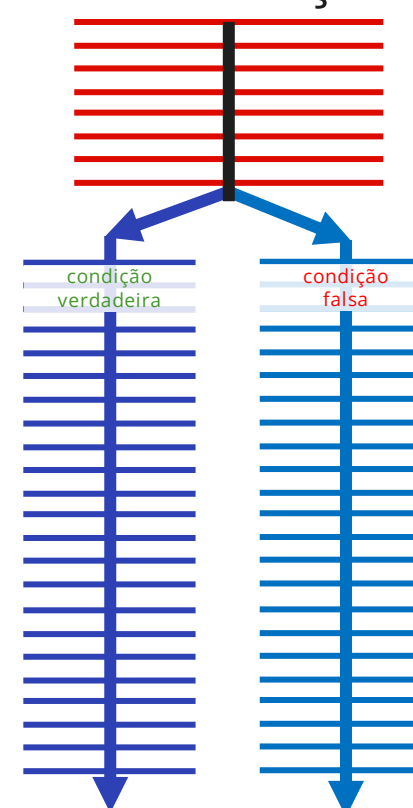
Código sem
instruções de seleção

Fluxo seletivo
de execução



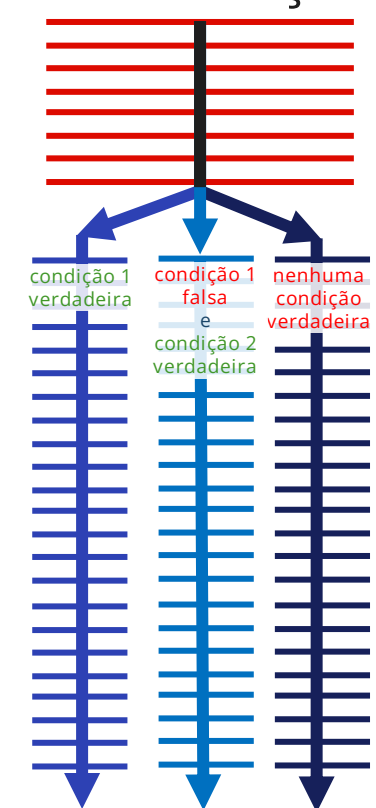
Código com if

Dois fluxos
de execução



Código com if / else
ou
operador ternário

Três ou mais fluxos
de execução



Código com
if / else if / else
ou
switch / case