

Lição 6



Estruturas de controle

Objetivos

Ao final desta lição, o estudante será capaz de:

- Usar estruturas de controle de decisão (if e switch) que permitem a seleção de partes específicas do código para execução
- Usar estruturas de controle de repetição (while, do-while e for) que permitem a repetição da execução de partes específicas do código
- Usar declarações de interrupção (break, continue e return) que permitem o redirecionamento do fluxo do programa



Estruturas de Controle

- Permitem mudança na ordem da execução dos programas
- Tipos:
 - Decisão
 - Repetição

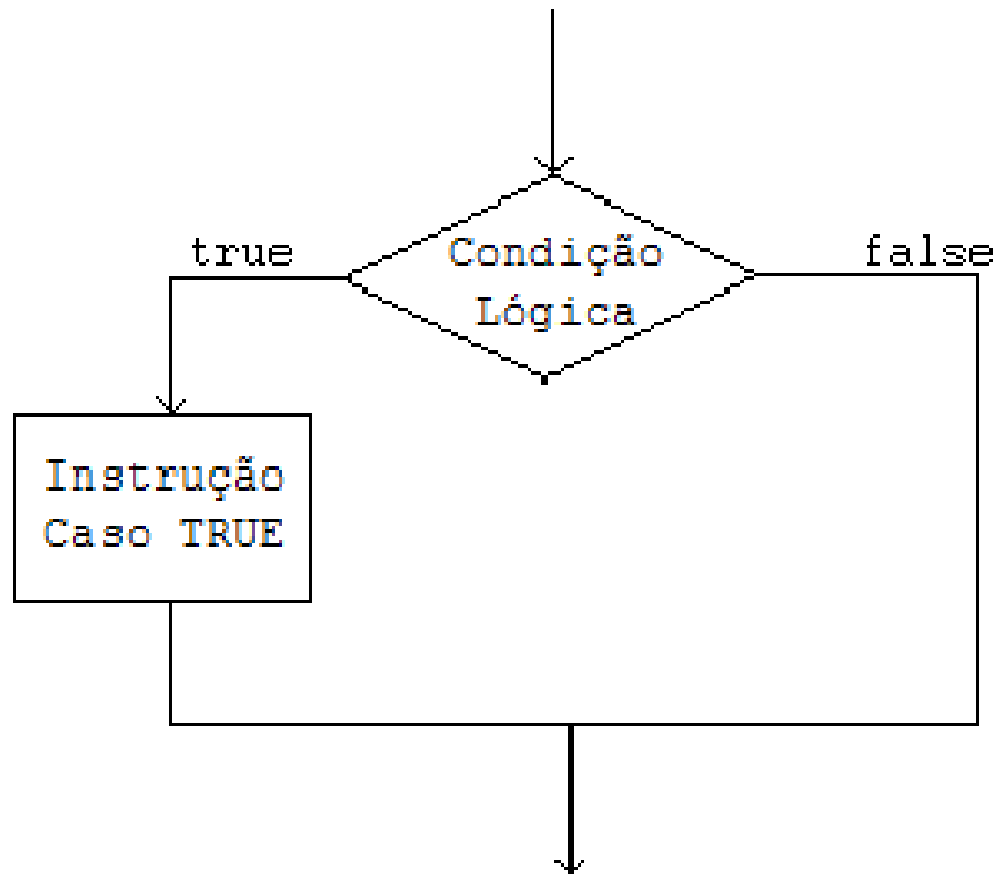
Estruturas de Controle de Decisão

- Permitem executar blocos específicos de instruções
- Tipos:
 - Declarações if
 - Declarações if-else
 - Declarações If-else-if
 - Declarações switch



Declaração if

```
if (expressão_lógica)  
    instrução_em_caso_verdadeiro;
```



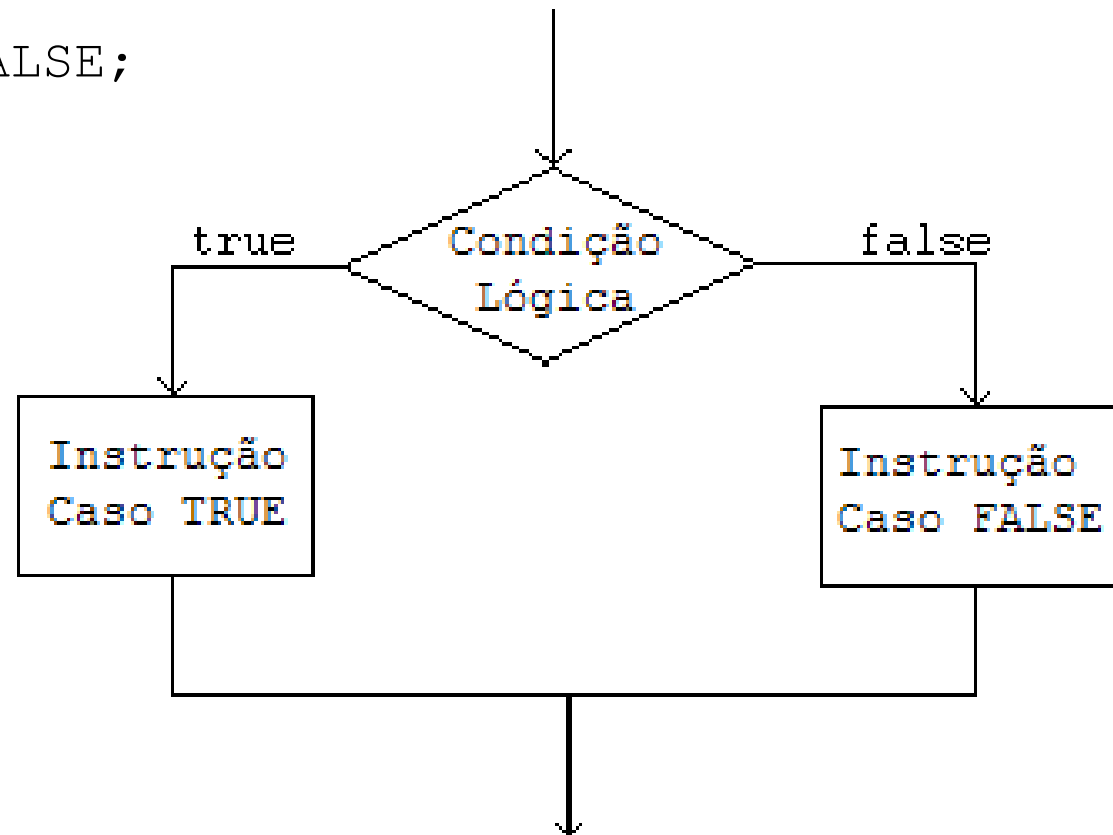
Declaração if: Exemplo

```
int grade = 68;  
if (grade > 60)  
    System.out.println("Congratulations!");
```



Declaração if-else

```
if (expressão_lógica) {  
    instruções_caso_TRUE;  
} else {  
    instruções_caso_FALSE;  
}
```



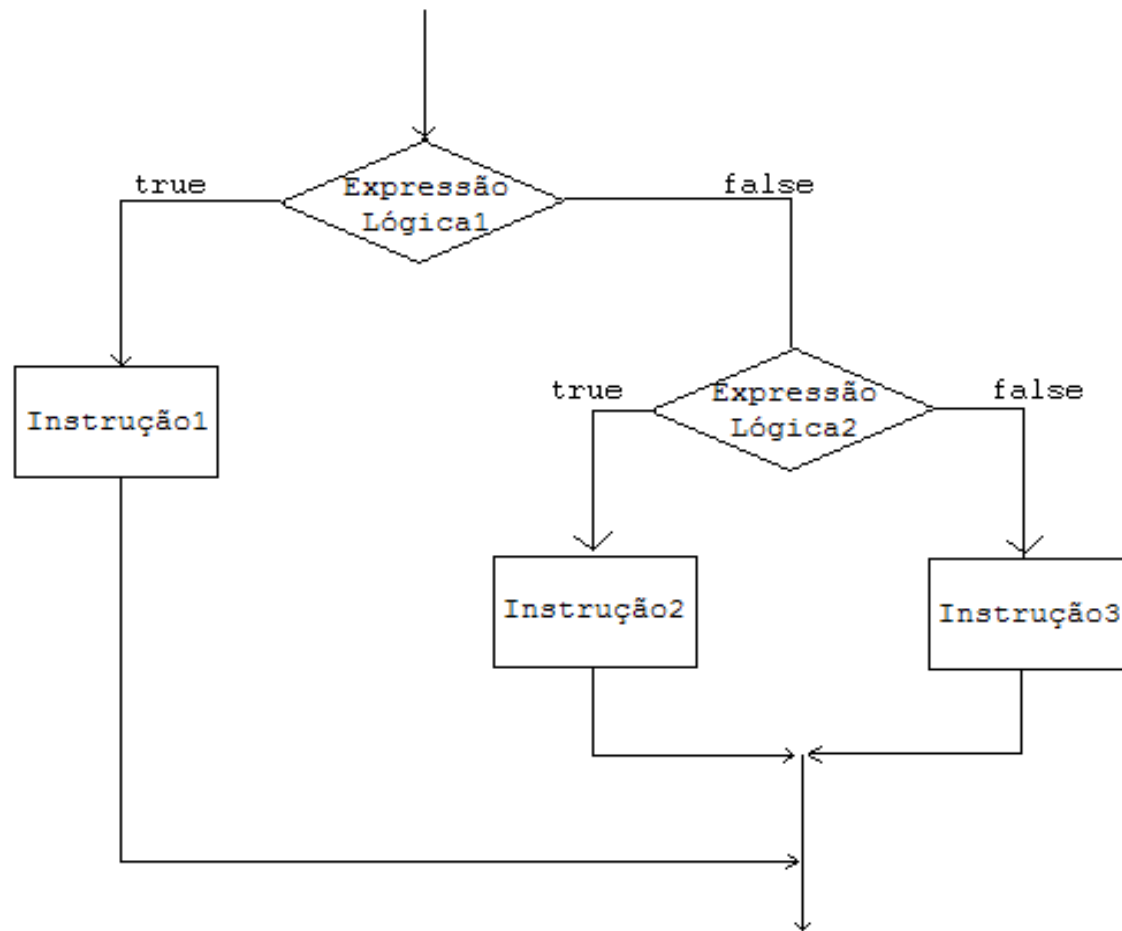
Declaração if-else: Exemplo

```
int grade = 68;  
if( grade > 60 )  
    System.out.println("Congratulations!");  
else  
    System.out.println("Sorry you failed");
```



Declaração if-else-if

```
if (expressão_lógica1)
    instrução1;
else if (expressão_lógica2)
    instrução2;
else
    instrução3;
```



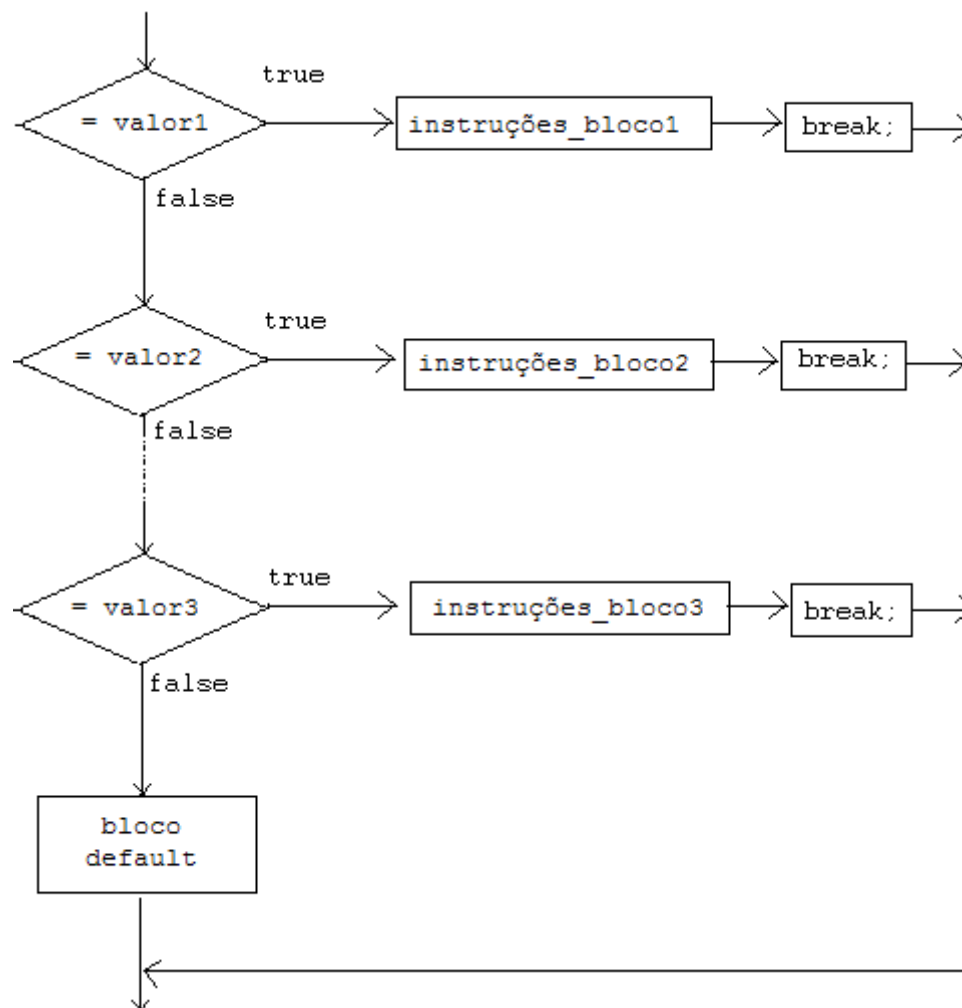
Declaração if-else-if: Exemplo

```
public class Grade {  
    public static void main( String[] args ) {  
        double grade = 92.0;  
        if (grade >= 90) {  
            System.out.println("Excellent!");  
        } else if((grade < 90) && (grade >= 80)) {  
            System.out.println("Good job!");  
        } else if((grade < 80) && (grade >= 60)) {  
            System.out.println("Study harder!");  
        } else {  
            System.out.println("Sorry, you failed.");  
        }  
    }  
}
```



Declaração switch

```
switch (variável_inteira) {  
  case valor1:  
    instruções_bloco1;  
    break;  
  case valor2:  
    instruções_bloco2;  
    break;  
  case valor3:  
    instruções_bloco3;  
    break;  
  default:  
    instruções_blocoDefault;  
}
```



Declaração switch: Exemplo

```
public class Grade {  
    public static void main( String[] args ) {  
        int grade = 92;  
        switch(grade){  
            case 100:  
                System.out.println("Excellent!");  
                break;  
            case 90:  
                System.out.println("Good job!" );  
                break;  
            case 80:  
                System.out.println("Study harder!" );  
                break;  
            default:  
                System.out.println("Sorry, you failed.");  
        }  
    }  
}
```



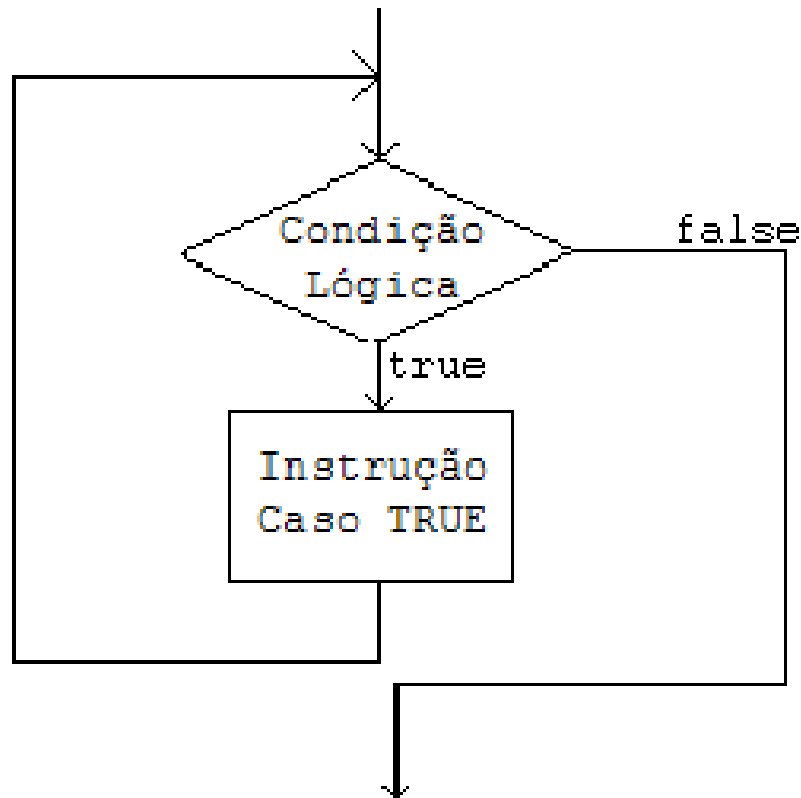
Estruturas de Controle de Repetição

- Permitem executar específicos trechos de código diversas vezes
- Tipos:
 - while
 - do-while
 - for



Declaração while

```
while (expressão_lógica)  
    instruções_enquanto_verdadeiro;
```



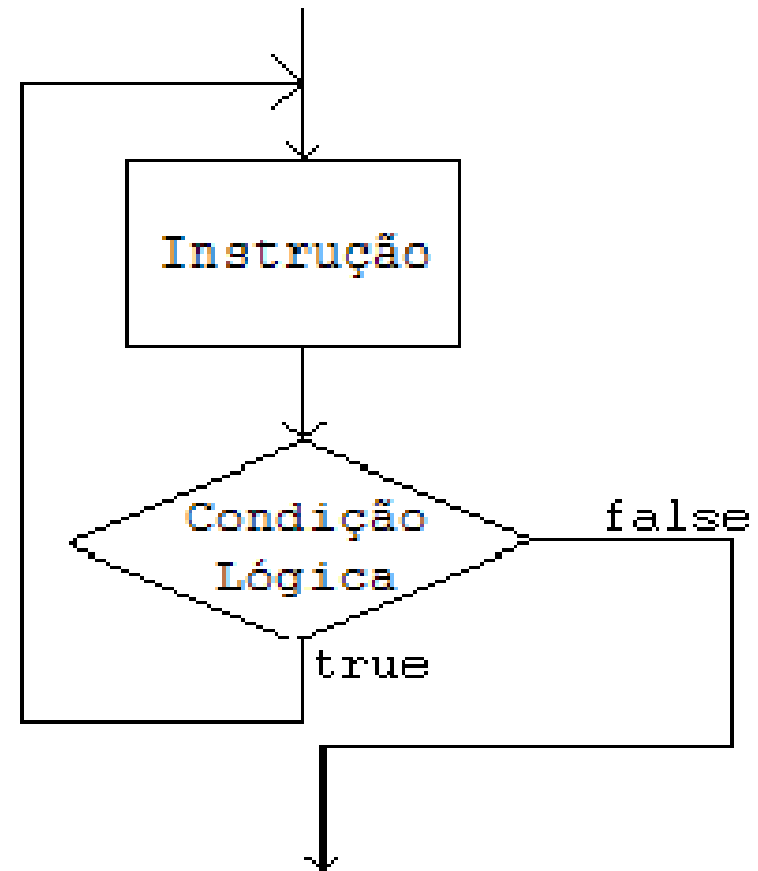
Declaração while: Exemplo

```
int x = 0;  
  
while (x<10) {  
    System.out.println(x);  
    x++;  
}
```



Declaração do-while

```
do {  
    instrução1;  
    instrução2;  
    ...  
} while (expressão_lógica);
```



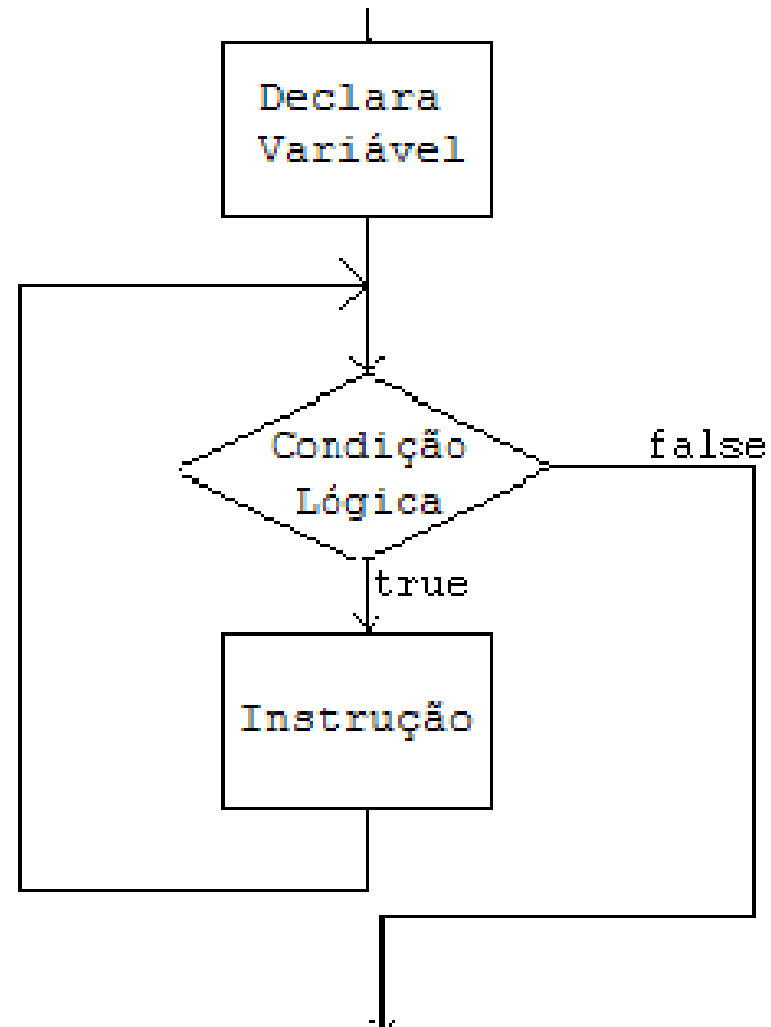
Declaração do-while: Exemplo

```
int x = 0;  
  
do {  
    System.out.println(x);  
    x++;  
}while (x<10);
```



Declaração for

```
for (declaração_inicial; expressão_lógica; salto) {  
    instrução1;  
    instrução2;  
    ...  
}
```



Declaração for: Exemplo

```
for (int i = 0; i < 10; i++) {  
    System.out.println(i);  
}
```

- As instruções mostradas acima são equivalentes ao seguinte laço while:

```
int i = 0;  
while (i < 10) {  
    System.out.print(i);  
    i++;  
}
```



Declarações de Interrupção

- Permitem redirecionar o fluxo de controle de um programa
- Java oferece três declarações de interrupção:
 - break
 - continue
 - return



Declaração unlabeled break

- Encerra uma declaração switch e o fluxo de controle passa para a instrução seguinte
- Utilizado para terminar declarações for, while, ou do-while

Exemplo

```
String names[] = {"Beah", "Bianca", "Lance", "Belle", "Nico", "Yza",  
    "Gem", "Ethan"};  
  
String searchName = "Yza";  
boolean foundName = false;  
for (int i=0; i < names.length; i++) {  
    if (names[i].equals(searchName)) {  
        foundName = true;  
        break;  
    }  
}  
  
if (foundName)  
    System.out.println(searchName + " found!");  
else  
    System.out.println(searchName + " not found.");
```



Declaração labeled break

- Termina um laço exterior que é identificado pelo label na declaração break
- O fluxo de controle segue imediatamente para a instrução seguinte àquela do label

Exemplo

```
int[][] numbers = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
int searchNum = 5;
boolean foundNum = false;
searchLabel: for (int i=0; i<numbers.length; i++) {
    for (int j=0; j<numbers[i].length; j++) {
        if (searchNum == numbers[i][j]) {
            foundNum = true;
            break searchLabel;
        }
    }
}
if (foundNum)
    System.out.println(searchNum + " found!");
else
    System.out.println(searchNum + " not found!");
```



Declaração unlabeled continue

- Retorna para o fim do corpo do laço interno e avalia a expressão lógica que o controla, saltando o restante das instruções deste laço

Exemplo

```
String names[] = {"Beah", "Bianca", "Lance", "Beah"};
int    count = 0;
for (int i=0; i<names.length; i++) {
    if (!names[i].equals("Beah")) {
        continue; // interrompe o fluxo
    }
    count++;
}
System.out.println(count + " Beahs in the list");
```



Declaração labeled continue

- Salta a repetição atual e retorna para próxima instrução indicada pelo label



Exemplo

```
outerLoop: for(int i=0; i<5; i++) {  
    for (int j=0; j<5; j++) {  
        System.out.println("Inside for(j) loop");  
        if (j == 2)  
            continue outerLoop;  
    }  
    System.out.println("Inside for(i) loop");  
}
```



Declaração return

- Abandona o método atual
- Fluxo de controle retorna para o local de chamada do método



Declaração return: retornando valores

- Escreva o valor a ser retornado (ou uma expressão que calcule este valor) após a palavra chave return
- Por exemplo:

```
return ++count;
```

```
return "Hello";
```

- O tipo de dado do valor devolvido deve ser igual ao tipo do valor de retorno declarado no método



Declaração return: sem retorno de valores

- Quando o retorno de um método é declarado como **void**, use a forma de retorno que não devolve um valor
- Por exemplo:

```
return;
```



Sumário

- Estruturas de Controle de Decisão
 - if
 - if-else
 - if-else-if
 - switch
- Estruturas de Controle de Repetição
 - while
 - do-while
 - for
- Estruturas de Interrupção
 - break
 - continue
 - return



Parceiros

- Os seguintes parceiros tornaram JEDI possível em Língua Portuguesa:

