

## Programação com java

### Introdução a Strings

O que é uma string?

String não é um tipo primitivo, quando falamos de texto é char!

String é uma cadeia de caractere, um conjunto de char.

Três métodos básicos = length, charAt e getchars.

length:

Retorna o tamanho da String, quantos caracteres tem a String.

CharAt:

Retorna a letra que está na posição desejada, ou seja, se minha String tem como valor o texto "curso", e eu digitar para imprimir a posição 2, irá imprimir a letra "r".

( C<sub>0</sub> U<sub>1</sub> R<sub>2</sub> S<sub>3</sub> O<sub>4</sub> ), sempre começando na posição ZERO.

### Exercícios

1) Segundo a linguagem Java, assinale a opção **INCORRETA**.

- a) O Java diferencia letras maiúsculas de minúsculas.
- b) A palavra-chave **new** introduz uma declaração de classe e é imediatamente seguida pelo nome da classe.
- c) O método **main** é o ponto de partida de cada aplicativo java e deve se iniciar com **public static void main(String[] args)**.
- d) O corpo de cada declaração de classe é delimitada por chaves.
- e) Cada instrução termina com um ponto e vírgula.

R: A palavra **new** introduz uma criação de objetos e não de classes. A criação de classes começa com **class**.


2) Uma pilha é uma versão limitada de uma lista encadeada. Novos nós só podem ser adicionados e retirados de uma pilha pelo topo. Em java, a classe Stack possui métodos de manipulação em pilhas. Em qual pacote a classe Stack está definida?

- a) **java.util**
- b) java.lang
- c) java.net
- d) java.io
- e) java.swt

3) Na linguagem de programação java, que palavra-chave cria um objeto de uma classe específica?

- a) main
- b) public
- c) class
- d) private
- e) new

### **Declaração de variáveis**

Tipo Inteiro  
int idade = 12;  
int idade (int) 12;                      Type cast  
Integer idade = new Integer(12);                      Classe  
 Wrapper Class

### **Tipo Decimal, Real...**

float salario = 1825.54f;  
float salario = (float) 1825.54;  
Float salario = new Float (1825.54);

### **Tipo Caractere, letra...**

char letra = 'G';  
char letra = (char) 'G';  
Character letra = new Character ('G');

### **Tipo Lógico, Boolean (True or False)...**

boolean casado = false;  
boolean casado = (boolean) false;  
Boolean casado = (new Boolean(False));

Família	Tipo Primitivo	Classe Invólucro	Tamanho	Exemplo
Lógico	boolean	Boolean	1 bit	true
Literais	char -	Character String	1 byte 1byte/cada	'A' "JAVA"
Inteiros	byte	Byte	1 byte	127
	short	Short	2 bytes	32 767
	int	Integer	4 bytes	2 147 483
	long	Long	8 bytes	$2^{63}$
Reais	Float	Float	4 bytes	$3.4e^{+38}$
	double	Double	8 bytes	$1.8e^{+308}$

## Entrada de dados

```
import java.util.Scanner;

Scanner leia = new Scanner(System.in);
System.out.println("Digite seu nome ");
String nome = leia.nextLine();
System.out.println("Digite sua nota ");
float nota = leia.nextFloat();
System.out.printf("Olá, %s, sua nota é %.1f", nome, nota);
```

## Saída de dados

```
float nota = 8.5f;

System.out.print("Sua nota é "+ nota);
System.out.println("Sua nota é "+ nota);
System.out.printf("Sua nota é %.2f", nota);
```

```
System.in = Entrada de dados.
System.out = Saída de dados.
```

## Incompatibilidades Números <-> Strings

```
int idade = 30;  
String valor = idade;  
String valor = (String) idade;  
String valor = (int) idade;  
String valor = Integer.toString(idade);
```

## String <-> Números

```
String valor = 30;  
int idade = valor;  
int idade = (int) valor;  
int idade = (String) valor;  
int idade = Integer.parseInt(valor);
```

## Exercícios

1) Em relação aos tipos básicos de dados (tipos primitivos), assinale a opção INCORRETA.

- a) booleano (ou lógico): conjunto de valores falso ou verdadeiro.
- b) vetor: estrutura que suporta NxM posições de um mesmo tipo.
- c) caracter: qualquer conjunto de caracteres alfanuméricos.
- d) inteiro: qualquer número inteiro, negativo, nulo ou positivo.
- e) real: qualquer número real, negativo, nulo ou positivo.

R: Vetor **NÃO** é um **TIPO PRIMITIVO**, e sim uma **ESTRUTURA DE DADOS** que armazena dados de um determinado **TIPO PRIMITIVO**.

2) Marque 1 para dados do tipo inteiro e 2 para dados de outros tipos.

- (2) "582.4"
- (2) .verdadeiro.
- (1) 105
- (1) -102
- (2) "0"
- (2) "informação"
- (2) "0.82"

- a) 2,2,1,1,1,2,1
- b) 2,2,1,1,2,2,2
- c) 1,1,1,2,2,1,2

d) 1,2,2,2,1,1,1

3) Assinale a alternativa que contenha somente nomes válidos de variáveis.

- a) índice. #pagina, contexto
- b) nome1, sobrenome2, senha3
- c) 2-nome, sobrenome, endereco
- d) 1-nome, 2-sobrenome, 3-senha

4) Assinale a alternativa que completa corretamente a lacuna de afirmativa a seguir.

São caracterizados como tipos\_\_\_\_\_os dados numéricos positivos ou negativos, excluindo-se destes qualquer fracionário.

- a) caracteres
- b) lógicos
- c) inteiros
- d) reais

5) Considerando os tipos de dados, relacione as colunas e, a seguir, assinale a alternativa com a sequência correta.

- |                |                           |
|----------------|---------------------------|
| (1) Inteiros   | (1) 35; 0; -56            |
| (2) Reais      | (4) .F.;.V.               |
| (3) Caracteres | (3) "Rua Brigadeiro Lyra" |
| (4) Lógicos    | (2) -0,5; 1,8; -4         |

- a) 3,1,4,2
- b) 2,4,3,1
- c) 1,2,3,4
- d) 1,4,3,2

## Operadores Unários

++	Incremento	a ++	a = a + 1
--	Decremento	a --	a = a - 1

Pré incremento = ++a

Pós incremento = a++

Ex:

```
int num = 5;
int num++;
System.out.println(num);
```

Resultado: 6

O num está valendo 5, quando eu coloco uma variável num++, esse 5 passa a valer 6 porque soma +1.

Ex2:

```
int num = 5;
int valor = 5 + num++;
System.out.println(valor);
```

Se o num passa a valer 6 e a variável valor vale 5, logo a soma dos dois vai ser 11, correto? ERRADO!

O ++ depois do num só vai acrescentar +1 depois de ter feito a soma do valor + num = 10.

```
int num = 5;
int valor = 5 + ++num;
System.out.println(valor);
```

Resultado: 11

O ++ antes do num faz com que o num já passe a valer 6, e a soma dos dois fique 5 + 6 e não 5 + 5 como antes.

## Operadores de Atribuição

+=	Somar e atribuir	a += b	a = a + b
-=	Subtrair e atribuir	a -= b	a = a - b
*=	Multiplicar e atribuir	a *= b	a = a * b
/=	Dividir e atribuir	a /= b	a = a / b
%=	Resto e atribuir	a %= b	a = a % b

Ex:

```
int x = 4;  
x += 2; // x = x + 2  
System.out.println(x);
```

Resultado: 6

## Classe Math

PI	Constante $\pi$	Math.PI	3.1415...
pow	Exponenciação	Math.pow(5,2)	25
sqrt	Raiz quadrada	Math.sqrt(25)	5
cbrt	Raiz Cúbica	Math.cbrt(27)	3

## Arredondamentos

abs	Valor Absoluto	Math.abs(-10)	10
floor	Arredondamento para Baixo	Math.floor(3.9)	3
ceil	Arredondamento para Cima	Math.ceil(4.2)	5
round	Arredondamento Aritmeticamente	Math.round(5.6)	6

Ex:

```
float y = 8.3f;  
int arredondado = (int) Math.floor(y);  
System.out.println(arredondado);
```

Resultado: 9

Única mudança feita no código acima é o valor da variável y e, a classe Math.

## Gerador de números

```
Math.random()
```

0.0 a 1.0



Ex:

```
double aleatorio = Math.random();  
System.out.println(aleatorio);
```

Resultado: 0.362718812323, 0.738265273736, 1.0...

Se você quiser gerar um número de 5 a 10 por exemplo.

```
5 + Math.random() * (10-5)
```

Ex:

```
double aleatorio = Math.random();  
int n = (int) (5 + aleatorio * (10-5));  
System.out.println(n);
```

## Exercícios:

1) Considerando o trecho de código a seguir, construído em Linguagem JAVA, marque a opção que preencha as lacunas:

```
int n = 4;  
System.out.println("O valor da variável é " + n++);
```

Executando as linhas acima, será exibida a mensagem "O valor da variável é \_\_\_\_\_" e o valor final de n após a execução será \_\_\_\_\_.

- a) 3 e 4
- b) 4 e 5
- c) 5 e 5
- d) 3 e 3



2) Considerando o tratamento de tipos primitivos pelo JAVA, analise o código a seguir e indique que o valor será impresso na tabela:

```
int n1 = 2, n2 = 5;  
double resultado = n2/n1;  
System.out.print(resultado);
```

- a) 5.0
- b) 2.0
- c) 2.5
- d) 3.0

3) Execute as três instruções JAVA a seguir e marque a opção que contém o valor que será impresso na tela:

```
int v1 = 7, v2 = v1 % 2, v3 = 2;  
v3 += v2;  
System.out.println(v1 + " " + v2 + " " + v3);
```

- a) 7 2 4
- b) 7 1 1
- c) 7 1 3
- d) 7 0 2

### Operador Ternário

```
maior = n1>n2 ? n1 : n2;
```

Ex:

```
int n1, n2, r;  
n1 = 12;  
n2 = 18;  
r = (n1>n2)?n1:n2;  
System.out.println(r);
```

Se n1 for maior que n2  
o r vai valer n1, se n2  
for maior, o r passa a valer n2.  
O r vai valer a variável de maior  
valor.

## Operadores Relacionais

>	Maior que	5 > 2	true
<	Menor que	4 < 1	false
>=	Maior ou igual a	8 >= 3	true
<=	Menor ou igual a	6 <= 6	true
==	Igual a	9 == 8	false
!=	Diferente de	4 != 5	true

### Exercícios:

1) Dadas as linhas a seguir:

```
int x = 8, y = 3;  
int w = x/y;  
String z = (w % 2 == 0)?"frase1":"frase2";  
System.out.println(z);
```

O que acontecerá se esse código for executado?

- a) Um erro de tipo na variável w
- b) Um erro de atribuição na variável z
- c) Será impresso "frase1"
- d) Será impresso "frase2"

2) Analise o código a seguir e indique o que será impresso na tela:

```
String parte1 = "Curso";  
String parte2 = "Video";  
String parte3 = parte1 + parte2;  
String parte4 = "CursoVideo";  
System.out.println(parte3 == parte4);  
System.out.println(parte3.equals(parte4));
```

- a) CursoVideo
- b) false true
- c) true true
- d) true false

3) Analise o código a seguir e indique o que será impresso na tela:

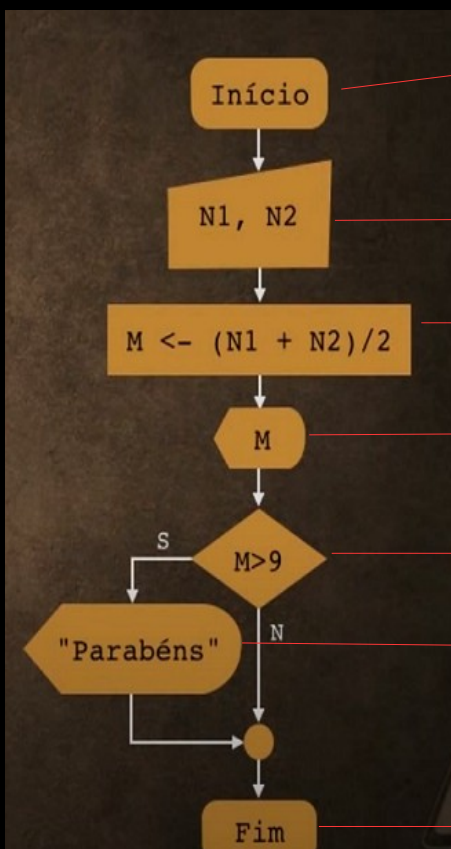
```
boolean val1 = (4>=5), (4<4), val3 = (val1==val2);  
boolean val4 = val1 ^ val3;  
boolean val5 = !val2 && val4;  
System.out.println(val4 + " " + val5);
```

- a) true true
- b) false true
- c) false false
- d) true false

## Estruturas condicionais

### Condição Simples

### Pseudocódigo



Início

Leia (N1, N2)

$M \leftarrow (N1 + N2) / 2$

Escreva (M)

Se (M > 9) então

Escreva ("Parabéns")

Fimse

FimAlgoritmo

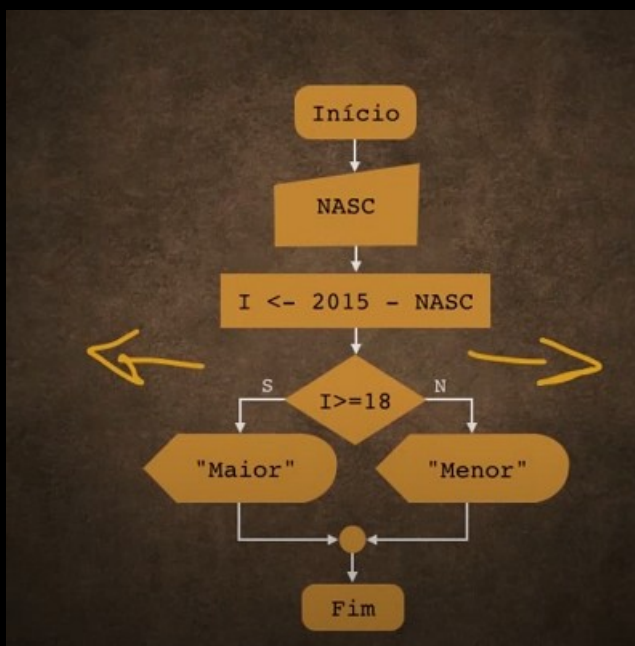
## Na linguagem JAVA

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

float N1, N2, M;
N1 = leia.nextFloat();
N2 = leia.nextFloat();
M = (N1 + N2)/2;
System.out.println(M);

if(M > 9){
System.out.println("Parabéns");
}
```

## Condição Composta



```
idade");
}
```

## Na linguagem JAVA

```
Scanner leia = new
Scanner(System.in);
int NASC, idade;

System.out.print("Digite seu
ano de nascimento ");
NASC =
leia.nextInt();
idade = 2023 - NASC;

System.out.println("Sua idade
é "+ idade);

if(idade >= 18){

System.out.println("Maior de
idade");
}else{
System.out.println("Menor de
```

## Na linguagem JAVA

```
Scanner leia = new Scanner(System.in);
int NASC, idade;
    System.out.print("Digite seu ano de nascimento ");
NASC = leia.nextInt();
idade = 2023 - NASC;
    System.out.println("Sua idade é "+ idade);

if(idade >= 18){
    System.out.println("Maior de idade");
} else{
    System.out.println("Menor de idade");
}
```

## Exercícios:

1) Analise o algoritmo abaixo.

```
Linha 1 início
Linha 2 inteiro Y, X, DIV;
Linha 3 leia(Y);
Linha 4 X = Y % 7;
Linha 5 se X > 10
Linha 6 então DIV = verdadeiro;
Linha 7 senão DIV = 10;
Linha 8 fim se;
Linha 9 fim
```

Analise a opção que apresenta a linha que contém o erro na execução do algoritmo.

- (A) Linha 2 inteiro Y, X, DIV.
- (B) Linha 4  $X = Y \% 7$ .
- (C) Linha 5 se  $X > 10$ .
- (D) Linha 6 então  $DIV = verdadeiro$ .
- (E) Linha 7 senão  $DIV = 10$ .

R: Variável do tipo INTEIRO não pode receber o valor como verdadeiro que é do tipo DOOUBLE.

2) Indique o resultado de "x" do trecho do programa em português estruturado, mostrado abaixo. Para as saídas, considere os seguintes valores:  
A=2; B=3; C=3; D=4

```
se .não. (D < 5) então
    x ← (A+B) * D
senão
    x ← (C+A) * B
fim_se
Escreva x.
```

- a) 15
- b) 10
- c) 25
- d) 20

3) Dada a estrutura representada abaixo:

```
int a = 5, b = 2;
String c;
if (a > b) {
    c = "Primeiro é Maior";
}else{
    c = "Segundo é Maior";
}
```

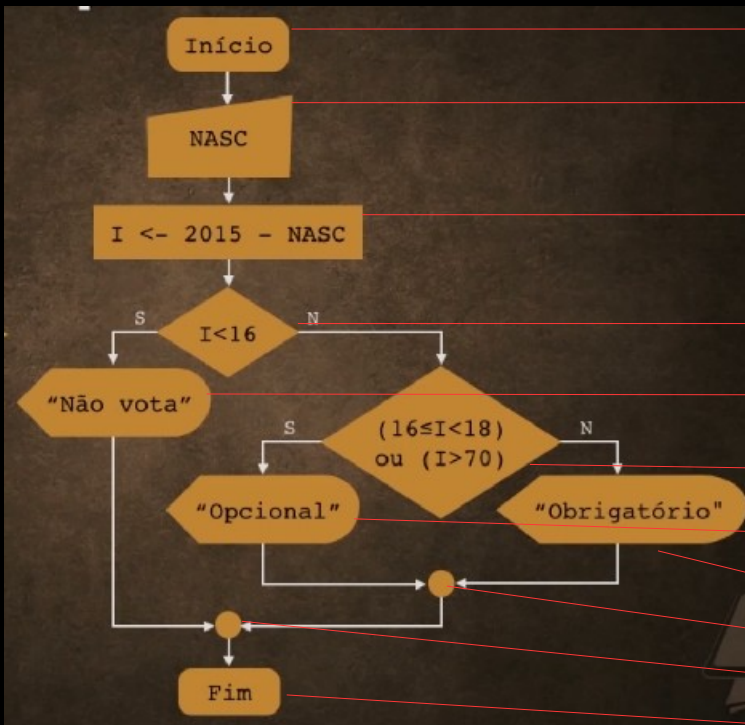
Que linha substituiria a condição apresentada?

- a) c = (a < b)?"Segundo é Maior":"Primeiro é Maior";
- b) c = (a <= b)?"Segundo é Maior":"Primeiro é Maior";
- c) c = (a > b)?"Segundo é Maior":"Primeiro é Maior";
- d) c = (a >= b)?"Primeiro é Maior":"Segundo é Maior";

## Estruturas Condicionais (Parte2)

### Condição Composta Encadeada

### Pseudocódigo



Início

Leia (NASC)

$I \leftarrow 2023 - \text{NASC}$

Se  $(I < 16)$  então

Escreva("Não Vota")  
senao

Se  $(I \geq 16 \text{ e } I < 18)$   
ou  $(I > 70)$  então

Escreva("Opcional")  
senao

Escreva("Obrigatório")

Fimse

Fimse

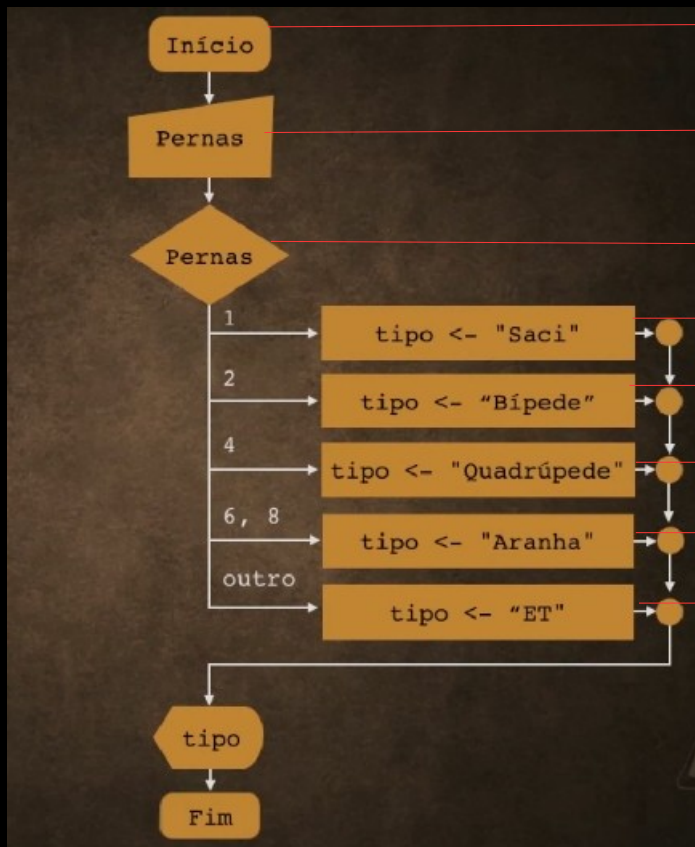
FimAlgoritmo

### Na linguagem JAVA

```
int nasc = leia.nextInt();
int i = 2023 - nasc;
if (i < 16) {
    System.out.print("Não Vota");
}else {
    if ((i >= 16 && i < 18) || (i > 70)) {
        System.out.print("Opcional");
    } eles {
        System.out.prit("Obrigatório");
    }
}
```

## Condição de Múltipla Escolha

## Pseudocódigo



Início

leia (Pernas)

Escolha (Pernas)

Caso 1

tipo<-"Saci"

Caso 2

tipo<-"Bípede"

Caso 4

tipo<-"Quadrúpede"

Caso 6, 8

tipo<-"Aranha"

OutroCaso

tipo<-"ET"

FimEscolha

Escreva (tipo)

## Na linguagem JAVA

```
int pernas = teclado.nextInt();
String tipo;
switch (pernas) {
    case 1:
        tipo = "Saci";
        break;
    case 2:
        tipo = "Bípede";
        break;
    case 4:
        tipo = "Quadrúpede";
        break;
    case 6,8;
        tipo = "Aranha";
        break;
    default:
        tipo = "ET";
}
System.out.println(tipo);
```



O problema do **Switch** é porque ele não funciona com intervalos de números, por exemplo eu quero testar de 2 à 5, eu preciso colocar 2, 3, 4, 5. O **Switch** também não serve para números **Reais**, apenas **Inteiros**.

### Exercícios:

1) Observe o trecho de programa:

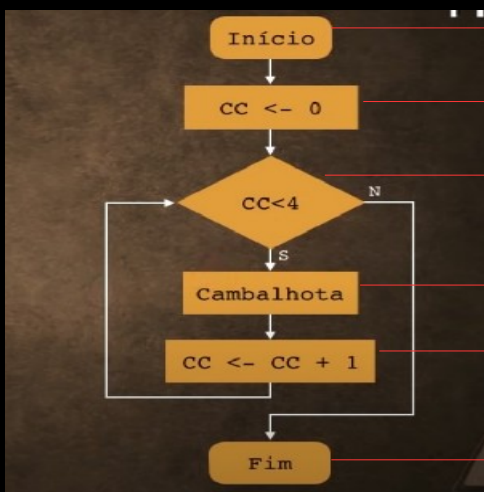
```
public class Exemplo {  
    public static void main(String[] args) {  
        System.out.println("Marinha do Brasil");  
    }  
}
```

Com base na classe Java acima, assinale a opção correta.

- a) O método main necessita receber o array como parâmetro para ser executado.
- b) Os modificadores public e class são desnecessários para executar o código.
- c) Sua execução dependerá da versão da JVM em uso.
- d) Essa classe não será executada pela JVM.
- e) O método main é o primeiro a ser chamado e executado pelo JVM.

### Estrutura de repetição

### Pseudocódigo



Início

CC <- 0

Enquanto (CC < 4) faça

Cambalhota

CC <- CC + 1

FimEnquanto

FimAlgoritmo

## Na linguagem JAVA

→ Enquanto

```
int cc = 0;
while (cc<4) {
    System.out.println("Cambalhota");
    cc++;
}
```

```
int cc = 0;
while (cc<4){
    cc++;
    if(cc == 2 || cc == 3){
        continue;
    }
    System.out.println(cc);
}
```

Resultado: 1, 4

De acordo com o if, se o **cc** for igual a 2 ou igual a 3, não irá executar esses dois números, e com o comando **continue** a execução continuara sem imprimir o 2 e o 3.

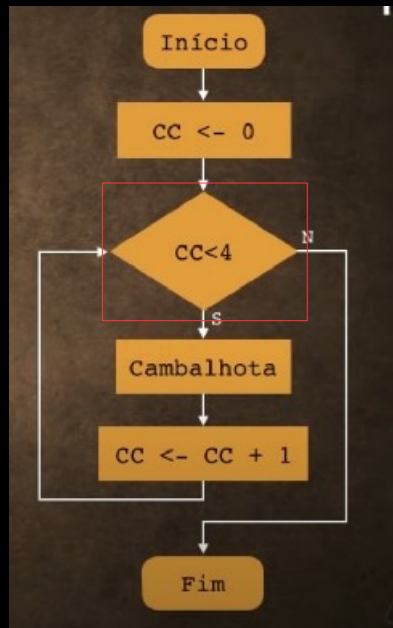
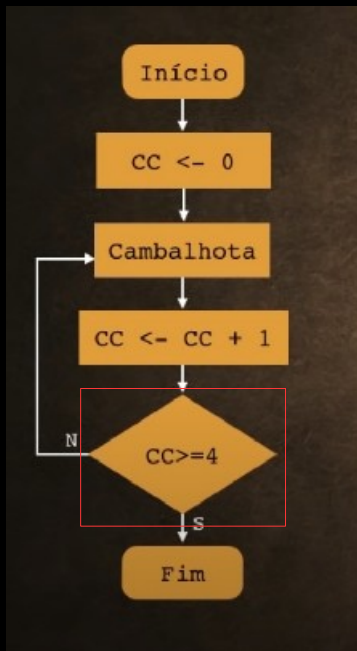
```
int cc = 0;
while (cc<4){
    cc++;
    if(cc == 3){
        break;
    }
    System.out.println(cc);
}
```

Resultado: 1, 2

De acordo com if, Se o cc for igual a 3, irá executar apenas os números 1 e 2 por conta do comando **break**, que para a execução a partir do número 3 em diante.

## Estrutura de repetição (parte 2)

### While



Note que o teste LÓGICO esta acontecendo depois.

### Pseudocódigo

```
Início
  CC ← 0
  Faça
    Cambalhota
    CC ← CC + 1
  Enquanto (CC < 4)
FimAlgoritmo
```

### Na Linguagem JAVA

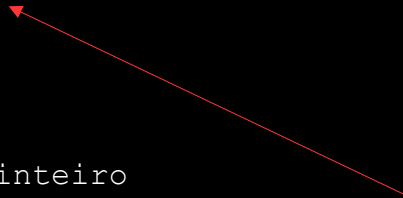
```
int cc = 0;
do {
    System.out.println("Cambalhota");
    cc++;
} while (cc < 4);
```

## Vetor

### Pseudocódigo

n

3	5	8	2
0	1	2	3



Var

n: vetor[0..3] de inteiro

início

n[0] ← 3	Posição 0 recebe o valor 3
n[1] ← 5	Posição 1 recebe o valor 5
n[2] ← 8	Posição 2 recebe o valor 8
n[3] ← 2	Posição 3 recebe o valor 2

### Na linguagem JAVA

```
int n[] = new int[4];  
n[0] = 3;  
n[1] = 5;  
n[2] = 8;  
n[3] = 2;
```

Quando eu já sei quantas posições o vetor vai ter e sei também qual o valor que vai em cada posição, eu posso usar esse outro método.

```
Int n[] = {3,5,8,2};
```

Outro exemplo de for especificamente para vetores.

```
int num[] = {3, 5, 2, 6, 3};  
for(int valor: num) {  
    System.out.println(valor);  
}
```

Colocar o vetor em ordem

```
int n[] = {2, 3, 4, 1, 5};  
Arrays.sort(n);  
for(int valor: n){  
    System.out.println(valor)  
}
```

Resultado: 1, 2, 3, 4, 5

Atribuir o mesmo valor para todas as posições do vetor

```
int n[] = new int[5];  
Arrays.fill(n, 8);  
for(int valor: n) {  
    System.out.println(valor);  
}
```

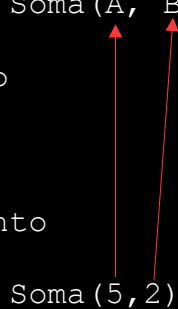
Resultado: 8, 8, 8, 8, 8

## Métodos

Pseudocódigo

```
Procedimento Soma(A, B: Inteiro)  
var  
    S : Inteiro  
Início  
    S ← A + B  
    Escreva(S)  
FimProcedimento
```

Soma(5, 2)



**Na linguagem JAVA**

► Procedimento (Não retorna valores)

```
static void soma (int a, int b){  
    int s = a + b;  
    System.out.print(s);  
}
```

```
soma(5,2);
```

**FIM!**