



# Processamento da Informação

## Estruturas de repetição: “for”

Prof. Jesús P. Mena-Chalco  
CMCC/UFABC

Q2/2018

# Leitura?

```
int t, r=0;
```

```
if (x>y) {  
    t = x;  
    x = y;  
    y = t;  
}
```

```
while (x<=y) {  
    r = r+x;  
    x = x+1;  
}
```

```
int i, j, soma=0;
```

```
i=1;  
while ( i<=x ) {  
    j=1;  
    while (j<=x) {  
        soma = soma + x;  
        j = j+1;  
    }  
    i = i+1;  
}
```

# Leitura?

```
int t, r=0;
```

```
if (x>y) {  
    t = x;  
    x = y;  
    y = t;  
}
```

```
while (x<=y) {  
    r = r+x;  
    x = x+1;  
}
```

```
int i, j, soma=0;
```

```
i=1;
```

```
while ( i<=x ) {  
    j=1;  
    while (j<=x) {  
        soma = soma + x;  
        j = j+1;  
    }  
    i = i+1;  
}
```

# Leitura?

```
int t, r=0;
```

```
if (x>y) {
```

```
    t = x;
```

```
    x = v;
```

```
    y = t;
```

```
}
```

```
while (x<=y) {
```

```
    r = r+x;
```

```
    x = x+1;
```

```
}
```

```
int i, j, soma=0;
```

```
i=1;
```

```
while ( i<=x ) {
```

```
    j=1;
```

```
    while (j<=x) {
```

```
        soma = soma + x;
```

```
        j = j+1;
```

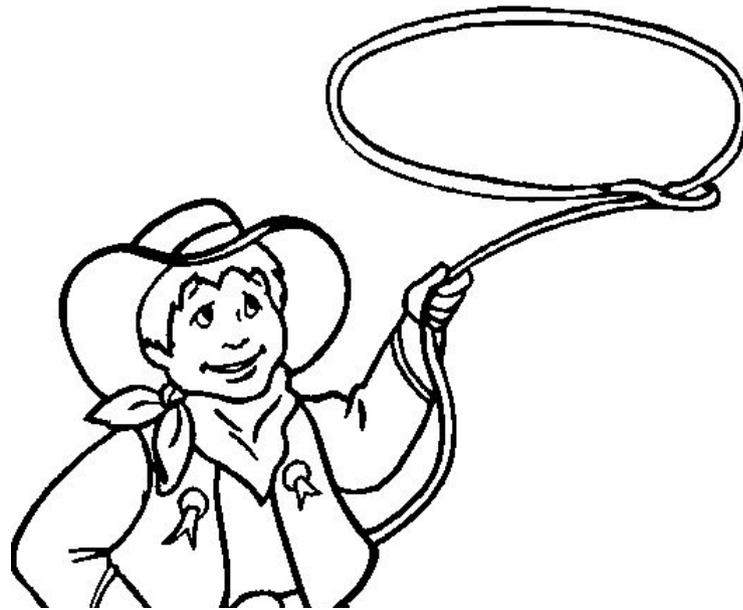
```
    }
```

```
    i = i+1;
```

```
}
```

# Estrutura de repetição: laço

Usado em situações em que é necessário repetir, **um determinado número de vezes**, um conjunto de instruções.



```
1 public class HelloWorld{
2
3     public static void main(String []args) {
4         System.out.println("Alo mundo!");
5         System.out.println("Alo mundo!");
6         System.out.println("Alo mundo!");
7         System.out.println("Alo mundo!");
8         System.out.println("Alo mundo!");
9         System.out.println("Alo mundo!");
10        System.out.println("Alo mundo!");
11        System.out.println("Alo mundo!");
12        System.out.println("Alo mundo!");
13        System.out.println("Alo mundo!");
14    }
15 }
```

```
1 public class HelloWorld
2 {
3     public static void main(String []args) {
4         int n=1;
5
6         while (n<=10) {
7             System.out.println("Alo mundo!");
8             n = n+1;
9         }
10    }
11 }
```

Ingredientes?

```
1 public class HelloWorld
2 {
3     public static void main(String []args) {
4         int n=1;
5
6         while (n<=10) {
7             System.out.println("Alo mundo!");
8             n = n+1;
9         }
10    }
11 }
```

# Estrutura de repetição: while

```
while ( ... ) {  
    ...  
    ...  
    ...  
}
```

```
int n=1;  
while ( n<=10 ) {  
    ...  
    ...  
    n = n+1;  
}
```

# Estrutura de repetição: while

```
while ( ... ) {  
    ...  
    ...  
    ...  
}
```

```
inicialização  
while ( condição ) {  
    ...  
    ...  
    atualização  
}
```

# Estrutura de repetição: for

```
for( inicialização ; condição ; atualização ) {  
    ...  
    ...  
    ...  
}
```

# Estrutura de repetição: for

```
1 public class HelloWorld
2 {
3     public static void main(String []args) {
4
5         for(int n=1; n<=10; n=n+1) {
6             System.out.println("Alo mundo!");
7         }
8     }
9 }
10 }
```



# Exercícios

# Exercício 01: Contagem regressiva

```
Numero 5  
Numero 4  
Numero 3  
Numero 2  
Numero 1  
Boom!
```

# Exercício 01: Contagem regressiva

```
int n = 5;

while (n>0) {
    System.out.println("Numero "+n);
    n = n-1;
}

System.out.println("Boom!");
```

```
for(int n=5; n>0; n=n-1) {
    System.out.println("Numero "+n);
}

System.out.println("Boom!");
```

# Exercício 02: Somatória de números inteiros

```
static int soma(int a, int b) {  
    int soma = 0;  
    int n = a;  
  
    while (n<=b){  
        soma = soma + n;  
        n = n+1;  
    }  
    return soma;  
}
```

```
static int soma (int a, int b) {  
    int soma = 0;  
  
    for(int n=a; n<=b; n=n+1) {  
        soma = soma+n;  
    }  
    return soma;  
}
```

# Exercício 03: Número triangular

Dizemos que um número natural é triangular se ele é produto de três números naturais consecutivos.

Dado um inteiro não negativo  $n$ , crie um método que permita verificar se é triangular.

## Exemplo:

- 120 é triangular, pois  $4 \cdot 5 \cdot 6 = 120$ .
- 2730 é triangular, pois  $13 \cdot 14 \cdot 15 = 2730$ .

**Assinatura:** `static boolean numeroTriangular(int n)`

Devolve “true” se o número for triangular, caso contrário “false”.

# Exercício 03: Número triangular

```
static boolean numeroTriangular2 (int n) {  
    int i;  
    for (i=1; i*(i+1)*(i+2)<=n; i=i+1) {  
        if ( i*(i+1)*(i+2) == n )  
            return true;  
    }  
    return false;  
}
```



# Return e break

# Return

```
static boolean numeroTriangular2 (int n) {  
    int i;  
    for (i=1; i*(i+1)*(i+2)<=n; i=i+1) {  
        if ( i*(i+1)*(i+2) == n )  
            return true;  
    }  
    return false;  
}
```

# Return

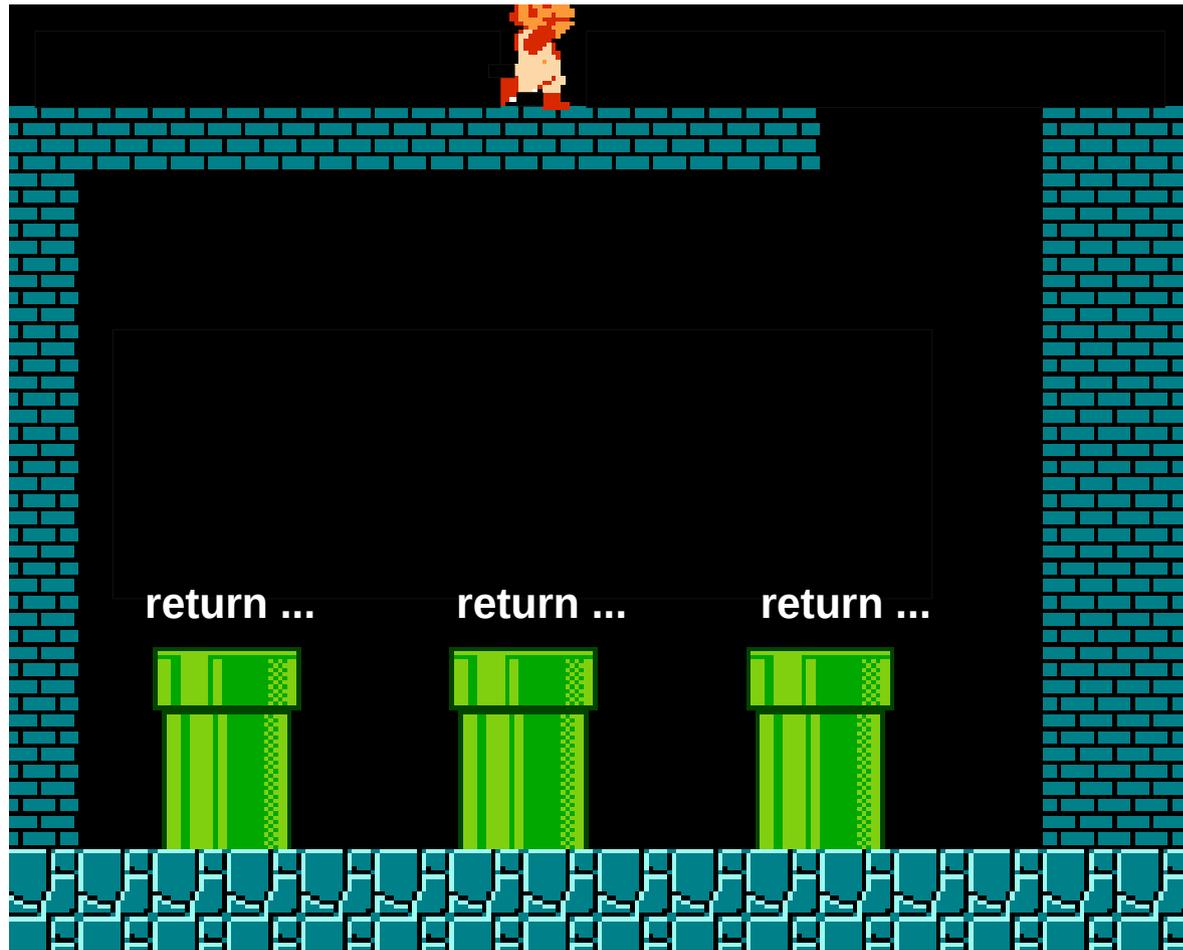
A instrução **return** permite **devolver** um valor e o controle ao programa no contexto da chamada a função.

```
static boolean numeroTriangular2 (int n) {  
    int i;  
    for (i=1; i*(i+1)*(i+2)<=n; i=i+1) {  
        if ( i*(i+1)*(i+2) == n )  
            return true;  
    }  
    return false;  
}
```

```
boolean x;  
x = numeroTriangular2(120);
```

# Return

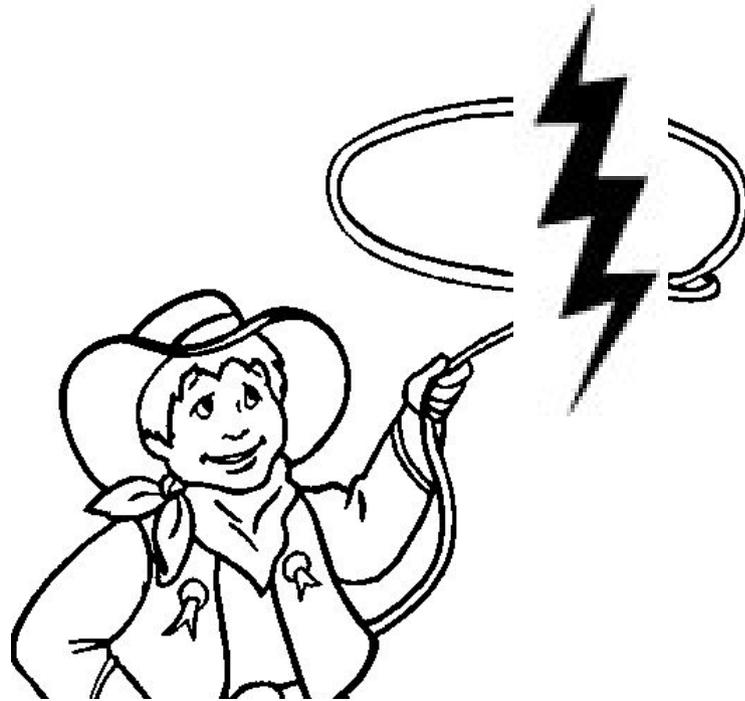
Função/  
Método



(c) Super Mario Bros.

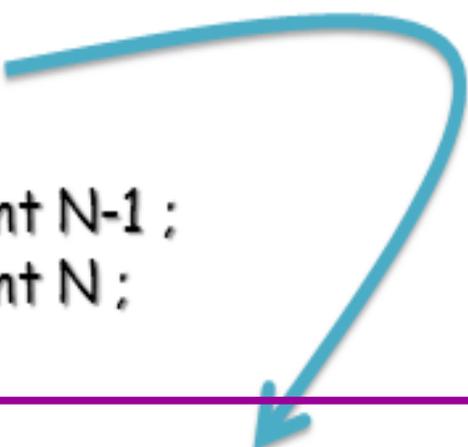
# Break

A instrução **break** permite parar um laço (sempre o mais interno) e saltar para a próxima instrução – após o laço.



# Break

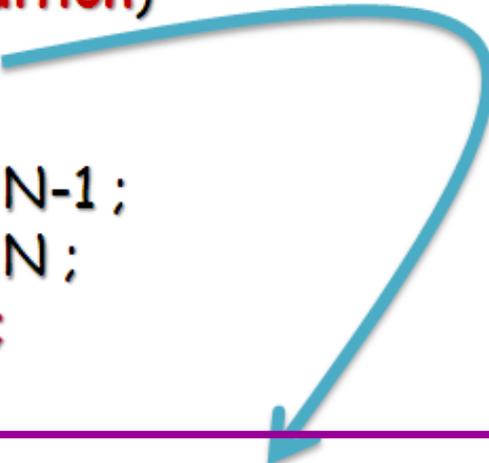
```
for ( initialization ; condition ; increment )  
{  
  Statement 1 ;  
  Statement 2 ;  
  Statement 3 ;  
  .....  
  .....  
  .....  
  break ;  
  Statement N-1 ;  
  Statement N ;  
}
```



OutsideStatement 1;

# Break

```
Initialization;  
while (condition)  
{  
  Statement 1 ;  
  Statement 2 ;  
  Statement 3 ;  
  .....  
  .....  
  if ( If Condition)  
    break;  
  
  Statement N-1 ;  
  Statement N ;  
  Increment;  
}
```

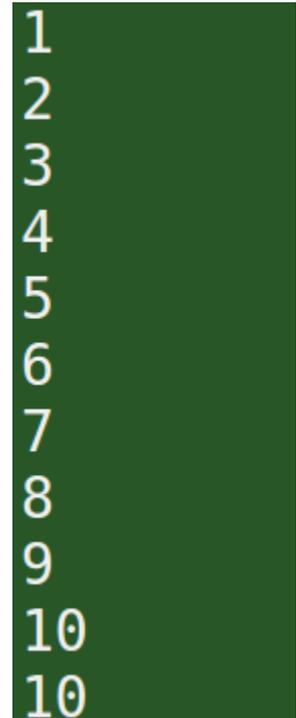


OutsideStatement 1;

# Break – Exemplo 1

```
static void funcaoB (int n) {  
    int x;  
  
    for (x=1; x<=n; x=x+1) {  
        System.out.println(x);  
  
        if (x==10)  
            break;  
    }  
  
    System.out.println(x);  
}
```

funcaoB(40);



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
10



# Atividade em aula

# Questão 1

```
static int m1(int n) {  
    int i, soma=0;  
  
    for(i=0; i<=n; i=i+1) {  
        soma = soma+i;  
    }  
  
    return soma;  
}
```

O método **m1** calcula a somatória dos **n** primeiros números naturais.

# Questão 2 e 3

```
static int m2(int n) {
    int i, soma=0;

    for(i=0; i<=n; i=i+2) {
        soma = soma+i;
    }

    return soma;
}
```

Os métodos **m2** calcula a somatória dos números pares menores ou iguais a **n**.

```
static int m3(int n) {
    int i, soma=0;

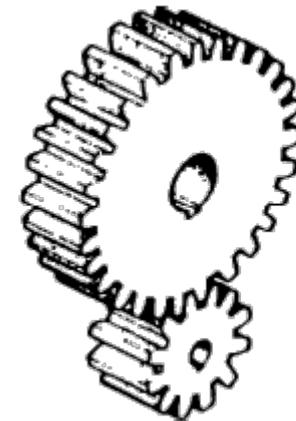
    for(i=n; i>=0; i=i-2) {
        soma = soma+i;
    }

    return soma;
}
```

# Questão 4

```
static int m4(int n) {  
    int r, s, soma=0;  
  
    for(r=1; r<=n; r=r+1) {  
        for(s=1; s<=n; s=s+1) {  
            soma = soma+1;  
        }  
    }  
  
    return soma;  
}
```

O método **m4** calcula, dado um inteiro positivo **n**,  **$n^2$** .



# Questão 5

```
static int m5(int n) {
    int r, s, t, soma=0;

    for(r=1; r<=n; r=r+1) {
        for(s=1; s<=n; s=s+1) {
            for(t=1; t<=n; t=t+1) {
                soma = soma+1;
            }
        }
    }

    return soma;
}
```

O método **m5** calcula, dado um inteiro positivo **n**,  **$n^3$** .



```
1 class Main {
2
3     static int numeroSegundosDia() {
4         int h, m, s;
5         int soma=0;
6
7         for (h=0; h<24; h=h+1) {
8             for (m=0; m<60; m=m+1) {
9                 for (s=0; s<60; s=s+1) {
10                    soma = soma + 1;
11                }
12            }
13        }
14
15        return soma;
16    }
17
18    public static void main(String[] args) {
19        System.out.println( numeroSegundosDia() );
20    }
21
22 }
```

# Questão 6

```
static int f1(int n) {  
    int x, soma=0;  
  
    for(x=1; x<=n; x=x+1) {  
        x = x+1;  
        soma = soma+x;  
    }  
  
    return soma;  
}
```

```
System.out.println( f1(6) );  
System.out.println( f1(9) );
```

12  
30

O método **f1** calcula o somatório dos números pares menores ou iguais a **n**.

# Questão 7

```
static int f2(int n) {  
    int j, soma=0;  
  
    for(j=1; 4*n>=j; j=j+1) {  
        soma = soma+1;  
    }  
  
    return soma;  
}
```

```
System.out.println( f2(6) );  
System.out.println( f2(9) );
```

24  
36

O método **f2** , dado **n**, calcula **4n**.

# Questão 8

```
static int f3(int n) {  
    int a, b, soma=0;  
  
    for(a=1; a<=n; a=a+1) {  
        for(b=a; b<=n; b=b+1) {  
            soma = soma+1;  
        }  
    }  
  
    return soma;  
}
```

Resposta correta +2 pontos  
Caso contrário -2 pontos

(a)  $n^2/2$

(b)  $(n^2 + n)/2$

(c)  $(n^2 - n)/2$

(d) 0

# Desafio

```
static int numeroSegundosDia() {  
    int h, m, s;  
    int soma=0;  
  
    for (h=0; h<24; h=h+1) {  
        for (m=0; m<60; m=m+1) {  
            if (m==30) {  
                break;  
            }  
            for (s=0; s<60; s=s+1) {  
                soma = soma + 1;  
            }  
        }  
    }  
  
    return soma;  
}
```

$$24 \times 30 \times 60 = 43200$$