

# Lógica de Programação

Prof. Dr. Bruno Queiroz Pinto

# Estrutura Repetição While

## Problema sem estrutura de repetição

Fazer um programa que lê dois números inteiros. Ao final mostra a soma dos números lidos.

**Entrada:**

**Dois números inteiros**

**Processo:**

Somar os números

**Saída:**

Imprimir o valor resultante da soma

```
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int numero1, numero2, soma;
        numero1 = s.nextInt();
        numero2 = s.nextInt();
        soma = numero1 + numero2;
        System.out.println("\nSoma = " + soma);
    }
}
```

**ou**

```
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int soma = 0;
        soma = soma + s.nextInt();
        soma = soma + s.nextInt();
        System.out.println("\nSoma = " + soma);
    }
}
```

# Estrutura Repetição While

## Problema sem estrutura de repetição

Fazer um programa que lê três números inteiros. Ao final mostra a soma dos números lidos.

**Entrada:**

**Três números inteiros**

**Processo:**

Somar os números

**Saída:**

Imprimir o valor resultante da soma

```
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int numero1, numero2, numero3, soma;
        numero1 = s.nextInt();
        numero2 = s.nextInt();
        numero3 = s.nextInt();
        soma = numero1 + numero2 + numero3;
        System.out.println("\nSoma = " + soma);
    }
}
```

**ou**

```
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int soma = 0;
        soma = soma + s.nextInt();
        soma = soma + s.nextInt();
        soma = soma + s.nextInt();
        System.out.println("\nSoma = " + soma);
    }
}
```

**Processo  
repetitivo**

# Estrutura Repetição While

## Problema com estrutura de repetição

Fazer um programa que lê números inteiros até que um negativo seja lido. Ao final mostra a soma dos números lidos.

### Entrada:

Vários números inteiros, entrar vários números inteiros, até a entrada de um número menor que 0

### Processo:

Somar os números

### Saída:

Imprimir o valor resultante da soma

???????

Exemplo de entradas:

10	20	33
20	2	22
3	3	6
2	4	5
3	0	4
4	-2	-1
3		
2		
1		
2		
-1		

# Estrutura Repetição While

## Problema com estrutura de repetição

### Entrada:

Vários números inteiros, entrar vários números inteiros, até a entrada de um número menor que 0


### Processo:

Somar os números

### Saída:

Imprimir o valor resultante da soma

```
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int numero, soma = 0;
        numero = s.nextInt();
        if (numero >= 0){
            soma = soma + s.nextInt();
        }
        numero = s.nextInt();
        if (numero >= 0){
            soma = soma + s.nextInt();
        }
        // não temos a informação da quantidade de números que teremos
        // como entrada
        System.out.println("\nSoma = " + soma);
    }
}
```



Processo repetitivo

Não dá para resolver esse tipo de problema apenas com estruturas condicionais e sequenciais..

# Estrutura Repetição While

## Sintaxe

```
09.  
10. while ( condição ) {  
11.     comando 1  
12.     comando 2  
13.     :  
14. }  
15.
```

Quando usar: quando não se sabe previamente a quantidade de repetições que será realizada.

## Regra:

Se a condição(linha 10) for

**Verdadeira:** executa o código interno(linhas 11 a 13) e volta para a condição (linha 10).

**Falsa:** pula fora, vai para a linha 15.

O laço de repetição WHILE é executado enquanto uma condição for verdadeira:

# Estrutura Repetição While

É uma **estrutura de controle** que **repete** um bloco de comandos **enquanto** uma **condição** for verdadeira.

**Quando usar:** quando não se sabe previamente a quantidade de repetições que será realizada.

```
import java.util.Scanner;
public class Main{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int numero=0, soma = 0;
        while(numero >= 0){
            soma = soma + numero;
            numero = s.nextInt();
        }
        System.out.println("\nSoma = " + soma);
    }
}
```

**ou**

```
import java.util.Scanner;
public class Main{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int numero=0, soma = 0;
        numero = s.nextInt();
        while(numero >= 0){
            soma = soma + numero;
            numero = s.nextInt();
        }
        System.out.println("\nSoma = " + soma);
    }
}
```

# Teste de mesa

```
x = 5;  
y = 0;  
while (x > 2) {  
    System.out.print(x);  
    y = y + x;  
    x = x - 1;  
}
```

--	--	--

**x**      **y**      **i**

Tela:

--

```
x = 2;  
y = 0;  
while (x < 60) {  
    System.out.println(x);  
    x = x * 2;  
    y = y + 10;  
}
```

--	--	--

**x**      **y**      **i**

Tela:

--

```
x = 100;  
y = 100;  
while (x != y) {  
    System.out.print("olha");  
    x = Math.sqrt(y);  
}
```

--	--	--

**x**      **y**      **i**

Tela:

--



# Teste de mesa

```
x = 0;  
while (x < 5) {  
    y = x * 3;  
    System.out.print(y);  
    x = x + 1;  
}  
System.out.println("Fim");
```

**x**

**y**

**i**

Tela:

```
x = 2;  
y = 10;  
System.out.println("Olá");  
while (x < y) {  
    System.out.println( x + "-" + y);  
    x = x * 2;  
    y = y + 1;  
}
```

**x**

**y**

**i**

Tela:

```
x = 4;  
y = 0;  
i = 0;  
while (i < x) {  
    i = i + 1;  
    y = y + i;  
    System.out.println(i + " " + y);  
}
```

**x**

**y**

**i**

Tela:

# Estrutura Repetição Do\_While

## Sintaxe

```
09.  
10. do {  
11.     comando 1  
12.     comando 2  
13.     :  
14. } while ( condição )  
15.
```

## Regra:

### Bloco de comandos:

- executa o código interno (linhas 11 a 13) e vai para a condição (linha 14).

### Condição

- se a condição (linha 14) for

**Verdadeira:** Execute o bloco de comandos novamente

**Falsa:** pula fora, vai para a linha 15.

Quando usar: quando houver a necessidade de executar no mínimo uma vez.

# Estrutura Repetição While

## Erros Comuns

Um erro muito comum quando se utiliza o WHILE é não atualizar a variável que está sendo verificada na condição, o que causa um laço infinito e o programa 'trava'.

# Estrutura Repetição While

Usos comuns

Criação de Menus:

```
public static void main(String[] args) {  
    Scanner leia = new Scanner(System.in);  
    int op=0;  
  
    while (op != 3) {  
        System.out.println("1-START GAME");  
        System.out.println("2-LOAD GAME");  
        System.out.println("3-EXIT GAME");  
        op = leia.nextInt();  
    }  
}
```

Do\_While recomendado...