



# **Algoritmos e Programação I**

## **Estruturas de Repetição**

Prof. Alexandre Mignon

Como podemos  
imprimir a mesma  
mensagem várias  
vezes na tela?

ALGORITMOS I

ALGORITMOS I

ALGORITMOS I

ALGORITMOS I

ALGORITMOS I

ALGORITMOS I

ALGORITMOS I

ALGORITMOS I

ALGORITMOS I

ALGORITMOS I

# Estruturas de Repetição

- Permitem executar um conjunto de comandos várias vezes.
- Escreve-se os comandos uma única vez, e eles podem ser executados várias vezes.
- São controladas por uma condição.
  - ◆ A interrupção das repetições é controlada pelo resultado da condição.
- As estruturas de repetição são também conhecidas como *laços de repetição*.

# Estruturas de Repetição

→ Em Java, temos três estruturas de repetição:

- ◆ `while` (enquanto)
- ◆ `do...while` (faça...enquanto)
- ◆ `for` (para)

## **while (Enquanto)**

- É uma estrutura que faz um teste lógico no início de um laço, verificando se é permitido executar o trecho de instruções subordinado a esse laço.
- Esta estrutura tem o seu funcionamento controlado por decisão, isto é, uma *condição*.
- Executa um determinado conjunto de instruções enquanto a condição verificada for **verdadeira**.
- No momento em que a condição se torna falsa, o processamento é desviado para fora do laço.
- Se a condição for falsa logo de início, as instruções contidas no laço são ignoradas.

## **while - Sintaxe**

```
while (condição) {  
    comandos...  
}
```

# Exemplo 01

```
public class ExemploWhile01 {  
    public static void main(String[] args) {  
        int i = 1;  
        while (i <= 10) {  
            System.out.println("ALGORITMOS I");  
            i = i + 1;  
        }  
    }  
}
```

# Contador

- Um contador é utilizado para contar o número de vezes que um bloco de instruções é executado.
- É uma variável que recebe o valor dela mesma mais um valor constante.

Exemplo: `i = i + 1;`

- Deve ser inicializado e incrementado.

```
int i = 1;
while (i <= 10) {
    System.out.println("ALGORITMOS I");
    i = i + 1;
}
```

**inicialização**

**incremento**



## Exemplo 02

```
public class ExemploWhile02 {  
    public static void main(String[] args) {  
        int i = 1;  
        int soma = 0;  
        // soma os números de 1 a 10  
        while (i <= 10) {  
            soma = soma + i;  
            i = i + 1;  
        }  
        System.out.println("Soma = " + soma);  
    }  
}
```

# Acumulador

- Um acumulador é usado para acumular valores cada vez que um laço é executado.
- É uma variável que recebe o valor dela mesma mais o valor de outra variável.

Exemplo: `soma = soma + i;`

```
int i = 1;
int soma = 0;
while (i <= 10) {
    soma = soma + i;
    i = i + 1;
}
```

inicialização

acumulação

# Exemplo 03

```
import java.util.Scanner;

/**
 * Lê números positivos até que um negativo seja informado
 * Ao final imprime a soma dos positivos
 */
public class ExemploWhile03 {
    public static void main(String[] args) {
        Scanner leitor = new Scanner(System.in);
        int soma = 0;
        System.out.print("Numero: ");
        int numero = leitor.nextInt();
        while (numero >= 0) {
            soma += numero; // soma = soma + numero;
            System.out.print("Numero: ");
            numero = leitor.nextInt();
        }
        System.out.println("Soma = " + soma);
    }
}
```

# Exercícios

1. Criar um programa que mostre os números pares entre 1 e 100, inclusive. Use o laço `while`.
2. Use o laço *while* para ler 06 valores do usuário. Ao final do laço, imprimir:
  - Soma de todos os valores digitados que são pares;
  - Quantidade de valores digitados que são pares.
  - Média (decimal) de todos os valores digitados que são pares.

# Exercícios

3. Criar um algoritmo que efetua a leitura sucessiva de valores numéricos inteiros positivos ou negativos, finalizando a leitura quando for fornecido o número zero. Ao final, apresente a soma dos números positivos, a soma dos números negativos e a soma das duas parciais.

## **for (Para)**

- É utilizada quando se sabe com antecedência quantas vezes será necessário repetir.
- Esta estrutura tem o seu funcionamento controlado por uma variável contadora.
- Executa um determinado conjunto de instruções certo número de vezes.

## for - Sintaxe

```
for (inicialização; condição; incremento) {  
    comandos...  
}
```

# Exemplo 01

```
public class ExemploFor01 {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 10; i++) {  
            System.out.println("ALGORITMOS I");  
        }  
    }  
}
```



## Exemplo 02

// Apresenta os números pares de 2 a 30.

```
public class ExemploFor02 {  
    public static void main(String[] args) {  
        for (int i = 2; i <= 30; i += 2) {  
            System.out.println(i);  
        }  
    }  
}
```

## **do...while (Faça...Enquanto)**

- Semelhante ao **while**, mas a primeira vez é **SEMPRE** executada.
- Verifica a condição após a execução das instruções do laço.

## do...while - Sintaxe

```
do {  
    comandos...  
} while (condição);
```

# Exemplo 01

```
public class ExemploDoWhile01 {  
    public static void main(String[] args) {  
        int i = 1;  
        do {  
            System.out.println("ALGORITMOS I");  
            i = i + 1;  
        } while (i <= 10);  
    }  
}
```

## Exemplo 02

Criar um algoritmo que efetue a leitura de uma série de valores, representando os preços dos itens comprados por um cliente. Ao final, apresentar na tela o total a ser pago. Para finalizar a entrada de dados o usuário deve informar o valor zero.

## Exemplo 02

```
import java.util.Scanner;

public class ExemploDoWhile02 {
    public static void main(String[] args) {
        Scanner leitor = new Scanner (System.in);
        float preco, soma = 0;
        do {
            System.out.print("Preco: ");
            preco = leitor.nextFloat();
            soma += preco; // soma = soma + preco;
        } while (preco != 0);
        System.out.println("Total: " + String.format("%.2f", soma));
    }
}
```

# O comando *break*

→ Pode ser usado em estruturas de repetição para sair imediatamente do laço.

```
public class ExemploBreak {  
    public static void main(String[] args) {  
        for (int i = 1; i < 1000; i+= 3) {  
            if (i % 10 == 0) {  
                break;  
            }  
            System.out.println("i = " + i);  
        }  
        System.out.println("Fim.");  
    }  
}
```

```
i = 1  
i = 4  
i = 7  
Fim.
```

# Exercícios

4. Criar um programa que apresente os anos bissextos de 2004 a 2050. (2004 é um ano bissexto).
5. Criar um programa que apresente uma contagem regressiva de 100 a 0, com decremento de 10 em 10.
6. Criar um programa que leia um número inteiro e apresente a sua tabuada.
7. Criar um programa que leia dois números inteiros. Se o segundo valor informado for ZERO, deve ser lido um novo valor, ou seja, para o segundo valor não pode ser aceito o valor zero. Apresentar na tela o resultado da divisão do primeiro valor lido pelo segundo valor lido.