



Universidade Federal Rural de Pernambuco  
(UFRPE)  
- Unidade Acadêmica de Garanhuns-

## Comandos de Iteração

# Introdução a Linguagem C Parte II

**Prof. Priscilla Kelly Machado Vieira**

# Apresentação do Capítulo

2

- Introdução
- Variáveis
- Iteração definida
- Iteração indefinida
- Comparação entre estruturas
- Exercícios

# Introdução



# Introdução

4

- Todas as linguagens de programação fornecem algum método de repetir execuções
- Evita a repetição sequencial de código
  - Mais flexível e Prático
  - Requer menor memória de armazenamento

```
view plain copy to clipboard print ?
01. import javax.swing.*;
02.
03. public class Exercicio6{
04.     public static void main(String args []){
05.         int idade;
06.
07.         ✨ while(idade!=0){
08.             idade=Integer.parseInt
09.                 (JOptionPane.showInputDialog("Digite a sua idade: "));
10.         }
11.         if (idade>14<=17)
12.
13.     }
14. }
```

# Introdução

- Estrutura de Repetição
  - Estrutura utilizada quando se é necessário efetuar a repetição de um trecho do algoritmo por 0 ou mais vezes;
  - O código passa a ter um tamanho menor, podendo ampliar o processamento, sem alterar o tamanho do código;
  - Utilizam-se loops, laços de repetição ou malhas de repetição.

# Variáveis

6

- São necessários dois tipos de variáveis:

## ★ Variáveis contadoras

```
...  
cont := 0  
<estrutura_de_repetição>  
    ...  
    cont := cont + 1  
    ...  
<fim_da_estrutura_de_repetição>  
...
```

## ★ Variáveis acumuladoras

```
...  
soma := 0  
<estrutura_de_repetição_com_variável_x>  
    ...  
    soma := soma + x  
    ...  
<fim_da_estrutura_de_repetição>  
...
```

# Estruturas de Repetição

7

- Uma estrutura de repetição possui corpo que é executado repetidamente até que uma condição de término seja atingida
- Cada execução do corpo do laço é chamada de iteração
- Tipos:
  - Iteração definida: número de iterações é definido
    - For
  - Iteração indefinida: caso contrário
    - Exemplo: While, repeat-until, do-while

# Estruturas de Repetição

- Iteração definida

```
int main()
{
    int c;
    for (c=1; c <=100; c++) {
        printf("%d ", c);
    }
}
```



# Estruturas de Repetição

- Iteração indefinida – teste no início

```
int main()
{
    int soma = 0;
    int entrada = 100;

    while (entrada != 0) {
        printf("Digite um numero: " );
        scanf("%d", &entrada);
        soma = soma + entrada;
    }
    printf("O resultado e %d", soma);
}
```

# Estruturas de Repetição

10

- Iteração indefinida – teste no fim

```
int main()
{
    int c = 0;
    do {
        c++;
        printf("%d ", c);
    } while (c<100);
}
```

# Estruturas de Repetição

- E se...

```
int main()
{
    int c = 0;
    do {
        c++;
        if (c == 50)
            continue;
        printf("%d ", c);
    } while (c < 100);
```

# Comparação entre Estruturas

12

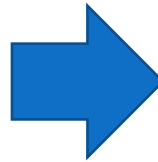
- Toda estrutura “**enquanto**” pode ser convertida para “**repita**”
- Toda estrutura “**para**” pode ser convertida para “**enquanto**”, mas a recíproca não é sempre verdadeira

# Comparação entre Estruturas

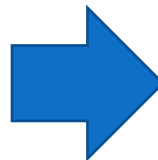
13

- Imprimindo a tabuada do 5

```
início
inteiro:CONT;
CONT<- 1;
enquanto CONT<= 10 faça
  imprima(CONT, "x 5" , CONT*5);
  CONT<- CONT+1;
fim enquanto;
fim.
```



```
início
inteiro:CONT;
para CONT de 1 até 10 passo 1 faça
  imprima(CONT, "x 5" , CONT*5);
fim para;
fim.
```



```
início
inteiro:CONT;
CONT<- 1;
repita
  imprima(CONT, "x 5" , CONT*5);
  CONT<- CONT+1;
até CONT> 10;
fim.
```

# Exercícios

- Lab 02
  - Imprima todos os divisores de um número inteiro positivo. Para o número  $n$  dado, o programa verifica se cada número de 1 até  $n$  é ou não um divisor de  $n$ .
  - Imprima  $n!$