INTRODUÇÃO À PROGRAMAÇÃO

Aula 9 - Estruturas de Repetição (Parte 1)

Prof. Carlos Alexandre Siqueira da Silva

Campus de Alegre

Revisão da Aula Anterior

- Estruturas Condicionais de programação utilizando a linguagem C
  - Operadores lógicos.
  - Tabela Verdade.
  - Condições Compostas.
  - Condições Aninhadas.
  - Estruturas de Múltiplas Escolhas.

Introdução

No momento, nossos programas são capazes de executar instruções sequenciais e verificar condições, mudando o seu fluxo de execução e acordo com as respostas dessas condições.

Para executarmos nosso programa em situações simples, como por exemplo calcular a nota final de um aluno, isso é o suficiente. Mas e se precisássemos calcular a nota final não de um, mas de 200 alunos?

Nesse caso, é necessário empregar alguma técnica ou método que nos permita repetir trechos de código sem a necessidade de "copiar e colar" esse trecho diversas vezes. A esse método, damos o nome de **Estrutura de Repetição**.

Por que usar estruturas de repetição?

- Muitos problemas exigem que um conjunto de instruções seja executado várias vezes.
- Em vez de repetir código manualmente, usamos laços para automatizar a repetição.
- As estruturas de repetição tornam o código mais compacto, eficiente e fácil de manter.

Estruturas de Repetição

- Estruturas que permitem a a repetição de trechos de código delimitados.
- Podem ser **Determinadas** ou **Indeterminadas**.
  - Determinadas: Repetem um bloco de comandos uma quantidade determinada de vezes, utilizando uma variável auxiliar para contar as execuções.
  - Indeterminadas: Repetem um bloco de comandos uma quantidade indeterminada de vezes, avaliando uma expressão lógica a cada execução.

Obs.: Nessa aula, vamos abordar as Estruturas de Repetição Indeterminadas.

Estruturas de Repetição Indeterminadas

- O ciclo básico de uma repetição indeterminada executa os seguintes passos:

  Entrada $\rightarrow$ Teste da condição $\rightarrow$ Execução do bloco $\rightarrow$ Retorno ao teste

- Esse ciclo se repete até que a condição seja falsa, encerrando o laço.

Comando while

O teste da condição é feito **antes da execução**. Dessa forma, o bloco de comandos pode não ser executado se a condição for falsa desde o início.

Sintaxe:

```
while(expressão lógica){
    Bloco de comandos
}
```

Comando do - while

O teste da condição é feito **após a execução**. Dessa forma, o bloco de comandos sempre será executado ao menos uma vez.

Sintaxe:

```
do{
    Bloco de comandos
} while(expressão lógica);
```

Comando while - Exemplo

Exemplo 1 – Exibir números de 1 a 5:

```c
int num = 1;
while (num <= 5) {
    printf("Contador: %d\n", num);
    num++;
}
```

Comando do - while - Exemplo

Exemplo 2 – Verificar senha digitada

```c
int senha;
do {
    printf("Digite a senha: ");
    scanf("%d", &senha);
} while (senha != 1234);

printf("Acesso permitido!\n");
```

DIferenças entre os comandos while e do - while

- while
  - O teste da condição é feito **antes da execução**.
  - O bloco de comandos pode não ser executado se a condição for falsa desde o início.

- do - while
  - O teste da condição é feito **após a execução**.
  - O bloco de comandos sempre será executado ao menos uma vez.

O que vem por aí...

Na próxima aula, vamos continuar estudando os laços de repetição na programação, conhecendo as **Estruturas de Repetição Determinadas**.