

Aula 6:

Estruturas de Controle Iterativas

Disciplina: Fundamentos de Programação

Prof. Luiz Olmes

olmes@unifei.edu.br



Nas aulas anteriores...

▶ **O QUE JÁ ESTUDAMOS?**

- ▶ Algoritmos.
- ▶ Linguagem C.
- ▶ Variáveis, operadores e tipos.
- ▶ Estruturas de controle condicionais.

▶ **OBJETIVOS:**

- ▶ Estruturas de controle iterativas:
 - ▶ `while`
 - ▶ `do...while`
 - ▶ `for`

Estruturas de Repetição

- ▶ Existem ocasiões onde é necessário efetuar a repetição de um trecho de um programa um determinado número de vezes.
- ▶ Para estas situações, são usadas estruturas que efetuam o processamento de um determinado bloco de código tantas vezes quanto forem necessárias.
- ▶ Estas estruturas são chamadas de **laços de repetição**, **malhas de repetição**, **estruturas de iteração**, ou **loops**.

Estruturas de Repetição

- ▶ Todo laço de repetição deve possuir uma condição que indique quando deve terminar.
- ▶ Uma condição mal formulada pode prender o programa dentro do laço de repetição.
- ▶ Esta é uma das causas mais comuns para o “travamento” de aplicativos, comumente chamada de **loop infinito**.

Laços de Repetição

- ▶ Definição:

- ▶ Estruturas computacionais que permitem a execução de um bloco de código por determinado número de vezes ou enquanto uma condição for verdadeira.

- ▶ Estruturas de repetição em C:

- ▶ `while()`: comando de repetição com teste lógico no início do loop.
- ▶ `do...while()`: comando de repetição com teste lógico no fim do loop.
- ▶ `for()`: comando de repetição que usa uma variável de controle.

while(): repetição com teste no início do loop

- ▶ A repetição com teste no início do loop é usada para repetir N vezes uma ou mais instruções.
- ▶ Não é necessário conhecer com antecedência o número de repetições.
- ▶ O controle do loop é realizado através de uma **condição**.
- ▶ PARA QUE O PROGRAMA **NÃO** ENTRE EM LOOP INFINITO, ESTA CONDIÇÃO **TEM** QUE SER ALTERADA EM ALGUM MOMENTO **DENTRO** DO LOOP.

`while()`: repetição com teste no início do loop

► Sintaxe:

```
while (condição)
{
}
```

while(): repetição com teste no início do loop

► Sintaxe:

Declaração do comando

```
while (condição)
{
}
```

Condição: o laço é executado enquanto a condição se mantém.

Corpo do while: delimitado por abre/fecha chaves { }

`while()`: repetição com teste no início do loop

- ▶ **Exemplo 1**: ler um valor inteiro `N` e mostrar na tela todos os números de 1 até o valor `N`.

while(): repetição com teste no início do loop

- ▶ **Exemplo 2:** ler um valor inteiro **N** e mostrar a soma dos números de 1 até **N** ($1 + 2 + 3 + \dots + N$).

`while()`: repetição com teste no início do loop

- ▶ **Exemplo 3**: ler um valor inteiro **N** e mostrar a soma seguinte série: $1/1 + 1/2 + 1/3 + \dots + 1/N$.

`while()`: repetição com teste no início do loop

- ▶ **Exemplo 4**: encontrar a primeira potência de 5 maior que 10000.

do...while(): repetição com teste no fim do loop

- ▶ De modo análogo ao comando **while**, o comando **do...while** é utilizado para que um bloco de código seja executado diversas vezes.
- ▶ A diferença entre os dois comandos está na verificação da condição:
- ▶ **while**:
 - ▶ Verificação da condição ocorre no início do bloco.
- ▶ **do...while**:
 - ▶ Verificação ocorre no final do bloco.
 - ▶ Mesmo que a condição seja, logo de início, falsa, a estrutura do loop é executada ao menos 1 vez.

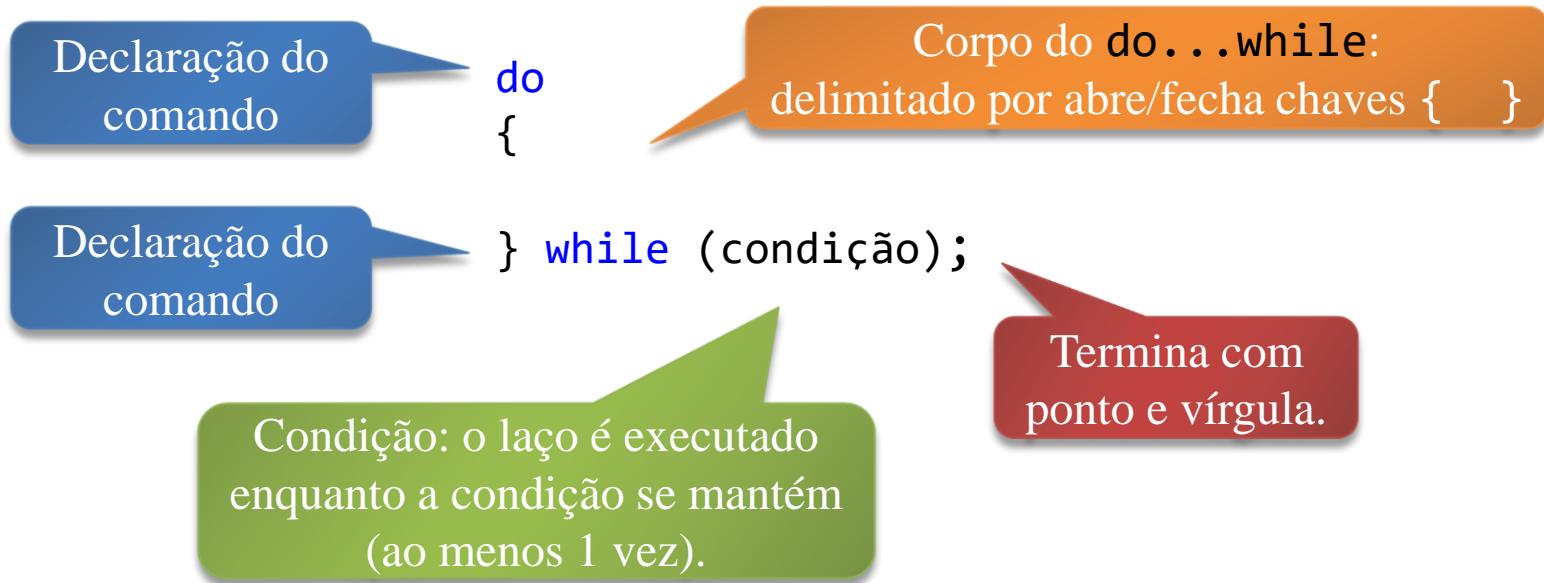
do...while(): repetição com teste no fim do loop

► Sintaxe:

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

do...while(): repetição com teste no fim do loop

► Sintaxe:



`do...while()`: repetição com teste no fim do loop

Atenção!!!

O controle do loop **TAMBÉM** é feito através de uma condição. Portanto, é necessário que essa condição seja **ALTERADA DENTRO** do loop para evitar um loop infinito.

do...while(): repetição com teste no fim do loop

- ▶ **Exemplo 5:** faça um programa que calcule a área de um círculo. O código não deve permitir entrada de dados inválidos (raio ≤ 0).

do...while(): repetição com teste no fim do loop

- ▶ **Exemplo 6:** faça um programa que calcule a área de um triângulo. O código não deve permitir entrada de dados inválidos: $\text{base} \leq 0$, bem como $\text{altura} \leq 0$.

for: repetição com variável de controle

- ▶ Ao contrário dos comandos **while** e **do...while**, a estrutura de repetição **for** utiliza uma variável de controle.
- ▶ A variável de controle permite que um bloco de código seja executado em uma quantidade de vezes **pré-estabelecida**.
- ▶ Para atingir este objetivo, o comando **for** utiliza em sua declaração um contador.
- ▶ Esse contador indica a quantidade de vezes que o programa passa pelo bloco que compõe o **for**.

for: repetição com variável de controle

► Sintaxe:

```
for ( inicialização ; condição ; passo )  
{  
  
}
```

for: repetição com variável de controle

► Sintaxe:

```
for ( inicialização ; condição ; passo )  
{  
}  
}
```

Corpo do for:
delimitado por
abre/fecha chaves

Declaração do comando

Ponto e vírgula
como separador.

Atribuição do valor inicial à
variável de controle.

Incremento / decremento
da variável de controle.

Condição: o laço é executado
enquanto a condição se
mantém.

for: repetição com variável de controle

- ▶ Passo do for:

- ▶ Considerando uma variável de controle “**i**”, expressões como “**i++**”, “**++i**”, “**i--**” e “**--i**” são válidas para realizar o incremento ou decremento em 1 unidade.

- ▶ Quando o passo do comando for **não** é unitário, deve-se declarar toda a expressão de alteração da variável:

- ▶ Exemplo (passo 2):

- ▶ `for (i = 0; i < n; i = i + 2)` ou `for (i = 0; i < n; i += 2)`

- ▶ É errado omitir o operador de atribuição:

- ▶ `for (i = 0; i < n; i + 2)`

for: repetição com variável de controle

- ▶ Passo do for:

- ▶ Considerando uma variável de controle “**i**”, expressões como “**i++**”, “**++i**”, “**i--**” e “**--i**” são válidas para realizar o incremento ou decremento em 1 unidade.

- ▶ Quando o passo do comando for **não** é unitário, deve-se declarar toda a expressão de alteração da variável:

- ▶ Exemplo (passo 2):

- ▶ for (i = 0; i < n; **i = i + 2**) ou for (i = 0; i < n; **i += 2**)



- ▶ É errado omitir o operador de atribuição:

- ▶ for (i = 0; i < n; **i + 2**)

for: repetição com variável de controle

- ▶ Variável de controle de 1 a 100, em passos de 1:

- ▶ `for (i = 1; i <= 100; i++)`

- ▶ Variável de controle de 100 a 1, em passos de 1:

- ▶ `for (i = 100; i >= 1; i--)`

- ▶ Variável de controle de 7 a 77, em passos de 7:

- ▶ `for (i = 7; i <= 77; i += 7)`

- ▶ Variável de controle de 20 a 2, em passos de -2:

- ▶ `for (i = 20; i >= 2; i -= 2)`

for: repetição com variável de controle

- ▶ **Exemplo 7:** faça um programa que some os números pares de 0 a 100.

for: repetição com variável de controle

- ▶ **Exemplo 8:** faça um programa que some os números naturais menores que 1000 que são múltiplos de 3 ou 5 .

for: repetição com variável de controle

- ▶ **Exemplo 9:** faça um programa que leia um valor inteiro n e calcule seu fatorial: $n! = 1 * 2 * 3 * 4 * \dots * n$.

for: repetição com variável de controle

- ▶ **Exemplo 10:** faça um programa que leia o número de termos **N** e mostre o valor da fórmula: $E = 1/0! + 1/1! + 1/2! + 1/3! + 1/4! + \dots + 1/N!$

for: repetição com variável de controle

- ▶ **Exemplo 11:** faça um programa que leia o número de termos **N** e calcule a sequência: $S = 1 - 2 + 3 - 4 + \dots + (2N - 1)$

Dúvidas?



Complemente seus conhecimentos...

- ▶ Procure aprender sobre o uso dos seguintes comandos dentro das estruturas de repetição:
 - ▶ `break`
 - ▶ `continue`



Aula 6:

Estruturas de Controle Iterativas

Disciplina: Fundamentos de Programação

Prof. Luiz Olmes

olmes@unifei.edu.br

