

Algoritmos e Programação

Laços de Repetição

Prof. Silvana Teodoro

Estruturas de Repetição



Estruturas de Repetição

Existem ocasiões em que é necessário efetuar a repetição de um trecho de programa um determinado número de vezes.

Neste caso, poderá ser criado um *looping* que efetue o processamento de um determinado trecho, tantas vezes quantas forem necessárias.

Os *loopings* também são chamados de **laços de repetição**.

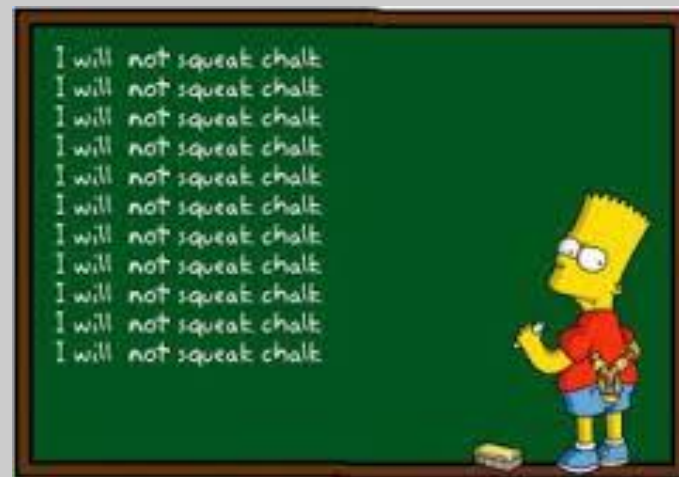


Estruturas de Repetição

Laços de Repetição

Supondo que um programa deva executar um determinado trecho de instruções por cinco vezes.

Com o conhecimento adquirido até este momento, você iria escrever o mesmo trecho, repetindo-o o número de vezes necessárias.



Estruturas de Repetição

Laços de Repetição

Imagine que um programa peça a leitura de um valor para a variável **num** e após multiplique esse valor por 3, colocando a resposta na variável **result** e apresente o valor obtido.

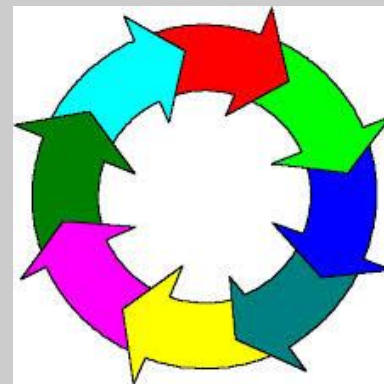
Caso fosse pedido para que essa sequência fosse repetida por três vezes como vocês fariam?

Estruturas de Repetição

Laços de Repetição

Para estes casos existem comandos apropriados que efetuam a **repetição** de determinados trechos de programas o **número de vezes que for necessário**.

A principal vantagem deste recurso é que o programa passa a ter um tamanho menor, podendo sua amplitude de processamento ser aumentada sem alterar o tamanho do código de programação.



Estruturas de Repetição

Laços de Repetição - while

Caracteriza-se por uma estrutura que efetua um teste lógico no início de um looping, verificando se é permitido executar o trecho de instruções subordinado a esse looping.

A estrutura em questão é denominada de enquanto, sendo conseguida com a utilização do conjunto de instruções **while()**.

A estrutura **while()** tem o seu funcionamento controlado por decisão. Sendo assim, **poderá executar um determinado conjunto de instruções enquanto a condição verificada for Verdadeira.**

No momento em que esta condição se torna Falsa, o processamento da rotina é desviado para fora do looping.

Se a condição for Falsa logo de início, as instruções contidas no looping são ignoradas.



Estruturas de Repetição

Laços de Repetição - while

Esta estrutura repete uma seqüência de comandos enquanto uma determinada condição (especificada através de uma expressão lógica) for satisfeita.

```
while (<expressão-lógica>) {  
    <seqüência-de-comandos>  
}
```

<expressão-lógica>	Esta expressão que é avaliada antes de cada repetição do laço. Quando seu resultado for VERDADEIRO, <seqüência-de-comandos> é executada.
}	Indica o fim da <seqüência-de-comandos> que será repetida. Cada vez que a execução atinge este ponto, volta-se ao início do laço para que <expressão-lógica> seja avaliada novamente. Se o resultado desta avaliação for VERDADEIRO, a <seqüência-de-comandos> será executada mais uma vez; caso contrário, a execução prosseguirá a partir do primeiro comando após fimenquanto.

Linguagem C

Estruturas de Repetição

- Instruções de repetição – **while**
 - Executa um bloco de repetição quando a condição for verdadeira

Sintaxe

```
1. while (condição) {  
2.     // instruções do bloco de repetição  
3. }
```

Exemplo

```
1. int x = 10;  
2. while (x > 0) {  
3.     // instruções de repetição  
4.     x--;  
5. }
```

- Passo 1
 - Verifica se a **condição** é verdadeira
- Passo 2
 - Executa o bloco de repetição
- Passo 3
 - Retorna ao Passo 1
- *Loop* termina quando a **condição** for falsa

Linguagem C

Estruturas de Repetição

- Instruções de repetição – **do{} while()**

- A principal diferença entre os comandos while e do-while é que no segundo o conjunto de instruções do bloco deverá ser executado pelo menos uma vez, obrigatoriamente.

Sintaxe

```
1. do {  
2.    // instruções do bloco de repetição  
3. } while (condição)
```

- Passo 1
 - Executa o bloco de repetição
- Passo 2
 - Verifica se a **condição** é verdadeira
- Passo 3
 - Retorna ao Passo 1
- *Loop* termina quando a **condição** for falsa

Linguagem C

Estruturas de Repetição

- Instruções de repetição – **do{} while()**

Exemplo

```
1. #include <stdio.h>
2. main( )
3. {
4.     int x, xq;
5.     printf ("\n\t Numero Quadrado \n\n");
6.     x = 1;
7.     do {
8.         xq = x*x;
9.         printf("\t %d %d \n", x, xq);
10.        x++;
11.    } while (x <100);
13. }
```

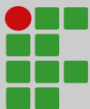
Estruturas de Repetição

Laços de Repetição - for

Os loopings que possuem um **número finito** de execuções poderão ser processados por meio de estrutura de laços contados do tipo **for**.

A declaração **for** é utilizada para definir que um bloco de comandos deve ser executado 'n' vezes, onde 'n' é um número inteiro.

Sendo assim, poderá executar esse bloco um determinado número de vezes.



Linguagem C

Estruturas de Repetição

- Instruções de repetição – **for**
 - Executa um bloco de repetição quando a condição for verdadeira

Sintaxe

```
1. for (inicialização; condição; incremento) {  
2.     // instruções do bloco de repetição  
3. }
```

Exemplo

```
1. int x;  
2. for (x=0; x<10; x++) {  
3.     // instruções de repetição  
4. }
```

- Passo 1
 - Executa a instrução de **inicialização**
- Passo 2
 - Verifica **condição**.
 - **condição** sendo verdadeira, vai ao Passo 3
 - **condição** sendo falsa, termina o *loop*
- Passo 3
 - Executa bloco de repetição
- Passo 4
 - Executa **incremento**
- Passo 5
 - Volta ao Passo 2

Linguagem C

Estruturas de Repetição

- Instruções de repetição – **for**
 - inicialização e incremento podem ter mais de uma instrução
 - Deve-se separar por vírgulas

Sintaxe

```
1. for (inicialização; condição; fim bloco) {  
2.     // instruções do bloco de repetição  
3. }
```

Exemplo

```
1. int x;  
2. for (x=0, y=10; x<10; x++, y--) {  
3.     // instruções do bloco de repetição  
4. }
```

Linguagem C

Estruturas de Repetição

- Instruções de repetição – **for**
 - inicialização e incremento podem ter mais de uma instrução
 - Deve-se separar por vírgulas

Exemplo

```
#include <stdio.h>
main( )
{
    int x;
    for(x=1; x<100; x++) printf("%d\n",x);
}
```

Exemplo

```
#include <stdio.h>
main( )
{
    for ( ; ; ) printf ("loop infinito\n");
}
```

Linguagem C

Estruturas de Repetição

- Instruções para controle de *loop* – **for**

- **break**

- Termina o **loop**

```
1. for (inicialização; condição; fim bloco) {  
2.     // instruções do bloco de repetição 1  
3.     break; // vai para a instrução depois da linha 5  
4.     // instruções do bloco de repetição 2  
5. }
```

- **continue**

- Termina a iteração e executa a próxima, caso a **condição** for verdadeira

```
1. for (inicialização; condição; fim bloco) {  
2.     // instruções do bloco de repetição 1  
3.     continue; // volta para a linha 1  
4.     // instruções do bloco de repetição 2  
5. }
```




Exercícios



Exercícios

- 1) Faça um programa que leia 10 números de matrícula fornecidos um a um pelo usuário. Em seguida, imprima os valores lidos.

