



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PET - PROGRAMA DE EDUCAÇÃO TUTORIAL - ENG. ELÉTRICA

CIT 2014.2

Aula 03 – Estruturas de controle de fluxo interativas (loops)

Autor: *Max Rodrigues Marques*

Carga Horária: 2h



***UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PET - PROGRAMA DE EDUCAÇÃO TUTORIAL - ENG. ELÉTRICA***

□ Oportunidades:

- Centros Acadêmicos – CA;
- Programa de Educação Tutorial – PET;
- Empresas Júniores – Ejs;
- Engenheiros Sem Fronteiras – ESF;
- Bases de Pesquisa;



Estruturas de controle de fluxo

- Estruturas interativas (loops)
 - 1-estrutura for()
 - 2-estrutura while()
 - 3-estrutura do { } while ()



Introdução

- As estruturas de controle de fluxo são fundamentais para qualquer linguagem de programação.
- Sem elas não haveria condições, repetições ou saltos.
- Em C++ e outras linguagens modernas de programação, comandos de iteração (também chamados laços) permite que um bloco de comandos ocorra até uma certa condição.



Laço **for**

- A declaração **for** é o que chamamos de laço ou loop em programação.
- Em **for**, determinamos o número de repetições desejadas através de uma variável de controle que será modificada pelos argumentos da declaração **for**.



Laço for

Ex:

- A sintaxe da declaração é:

```
for(inicio; condição; incremento)
{
    instruções;
}
```



```
1 #include <iostream>
2
3 using namespace std;
4 //contando 10 carneirinhos
5 int main()
6 {
7     int contador=1;
8     for(contador;contador<=10;contador++)
9     {
10         cout<<contador<<" carneirinho(s)!"<<endl;
11     }
12     return 0;
13 }
14 }
15
```



Laço `while`

- Laço que pode ter uma condição de termino definida já no início ou não.
- Necessariamente ele testa a condição e se caso for verdadeiro executa o bloco abaixo, caso seja falso ele vai para a próxima instrução fora do laço.
- Sintaxe:

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




Ex:

```
1 #include <iostream>
2 using namespace std;
3 //outro modo de contar 10 carneirinhos
4 int main()
5 {
6     int contador=1;
7     while(contador<=10)
8     {
9         cout<<contador<<" carneirinho(s)!"<<endl;
10        contador++; /*IMPORTANTE, lembre-se de colocar sempre o incremento ou
decremento, caso contrario, vc entrara em um loop infinito
11        */
12    }
13    return 0;
14 }
15
```



Prática 1

- 1- Escreva um programa que realiza a soma dos 100 primeiros números naturais e exiba na tela o resultado.
- 2- Escreva um programa que calcule a média de idade de uma turma (no mínimo 10 alunos).
- 3- Escreva um programa que recebe um valor digitado pelo usuário e imprime 10 múltiplos deste valor.



Laço `do` { `while`()

- A instrução `do...while()` difere dos laços anteriores porque o teste da condição é realizado no fim do corpo do laço e não antes, como acontecia com os laços `for` e `while`.
- sintaxe:

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



Ex:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int cont=0;
8     do
9     {
10         cout<<"Digite 1 para finalizar o programa!"<<endl;
11         cin>>cont;
12     }
13     while(cont!=1);
14
15     return 0;
16 }
```



Prática 2

1. Escreva um programa que soma os valores positivos digitados pelo usuário até que o mesmo digite um valor negativo.
2. Escreva um programa que recebe um valor e imprime seus divisores (valor menor que 100).
3. Escreva um programa que imprime a tabuada de um número digitado pelo usuário.



Dúvidas?