

C202-A

Algoritmos e Estruturas de Dados I

07 – Estruturas de Repetição (Comandos while e do-while)

Prof. Edson J. C. Gimenez
soned@inatel.br

2019/Sem1

Estrutura de repetição com teste no início: WHILE (enquanto-faça)

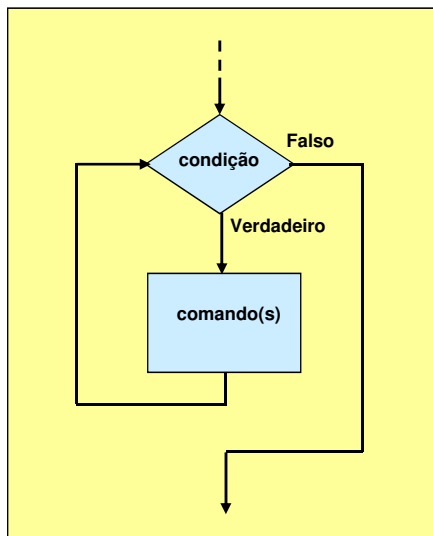
Na estrutura *while*, o comando (ou bloco de comandos) será executado enquanto a condição indicada for verdadeira.

- Ao chegar na estrutura, a condição é testada. Se o resultado for verdadeiro, o comando (ou bloco de comandos) é executado.
- Terminada a execução, a condição é testada novamente. Se o resultado for verdadeiro, o comando (ou bloco de comandos) é executado novamente.
- Terminada a execução, a condição é testada novamente. Se o resultado for verdadeiro, o comando (ou bloco de comandos) é executado novamente.
- •
•
- E assim por diante, até que o teste da condição dê falso como resultado.
- Se o resultado da condição for falso, a estrutura é abandonada

Como a condição é testada no início do laço de repetição, pode acontecer do comando (ou bloco de comandos) não ser executado nenhuma vez.

Quando o laço de repetição possuir mais que um comando (bloco de comandos), estes devem ser delimitados por { }.

Estrutura de repetição com teste no início: WHILE (enquanto-faça)



```
while (condição)
    comando;
```

```
while (condição)
{
    comando1;
    comando2;
    ...
    comando_n;
}
```

Ex1: * usado para validação de uma entrada de dados

```
#include <iostream>
using namespace std;
int main( )
{
    int id;
    cin >> id;
    while ( id <= 0)    // testa a condição
    {
        cin >> id;
    }
    :
    :
}
```

Ex2:

```
#include <iostream>
using namespace std;
int main( )
{
    int i = 1;    //inicializa a variável de controle
    while ( i <= 5)    // testa a condição
    {
        // início do laço de repetição
        cout << i;
        i++;    //incrementa a variável de controle
    }    // término do laço de repetição
    return 0;
}
```

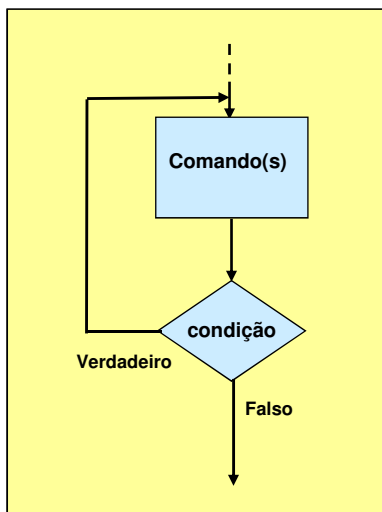
```
#include <iostream>    // usando while para validação
#include <locale>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Portuguese");
    int nota, soma=0, i, n;
    double media;
    cout<<"Quantas notas?(>0) ";
    cin>>n;
    while(n<=0){
        cout<<"Núm. de notas > 0: "; cin>>n;
    }
    for(i=1;i<=n;i++){
        cout << "Entre com a nota "<< i <<" (0 - 100): ";
        cin >> nota;
        while(nota<0 || nota>100){
            cout<<"Digite novamente, de 0 a 100: ";
            cin >> nota;
        }
        soma=soma + nota;
    }
    cout<<"Média das "<< n << " notas = "<<soma/n;
    return 0;
}
```

//usando do-while para validação

```
#include <iostream>
#include <locale>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Portuguese");
    int nota, soma=0, i, n;
    double media;
    do{
        cout<<"Quantas notas?(>0) ";
        cin>>n;
    }while(n<=0);
    for(i=1;i<=n;i++)
    {
        do
        {
            cout << "Entre com a nota "<< i << " (0 - 100): ";
            cin >> nota;
        }while(nota<0 || nota>100);
        soma=soma + nota;
    }
    cout<<"Média das "<< n << " notas = "<<soma/n;
    return 0;
}
```

Estrutura de repetição com teste no final: DO - WHILE (faça - enquanto)

- Semelhante ao comando while (), porém com uma diferença fundamental: **a condição é testada no final do laço.**
- Assim, o(s) comando(s) que compõem o laço de repetição é(são) executado(s) pelo menos uma vez.



```
do
comando;
while (condição);
```

```
do
{
    comando1;
    comando2;
    ...
    comando_n;
} while (condição);
```

Ex4: Faça um programa para ler 10 números e mostrar a média aritmética como resultado.

```
int main ()           // programa de média
{
    float num, soma;   // número lido e soma dos 10 números lidos
    int cont;          // contador
    cont = 0;          //inicialização da variável cont
    soma = 0;          //inicialização da variável soma
    do
    {
        cout << "Digite um número ";
        cin >> num;
        soma = soma + num;    // soma do número lido
        cont = cont + 1;      // contagem dos números lidos
    } while (cont < 10);
    cout << "A média aritmética dos números é " << soma/10;
    return 0;
}
```

Exemplo 5: Faça um programa ler alguns números e só parar de ler quando o número digitado for **-1 (FLAG)**; depois mostrar quantos números foram lidos e a média aritmética desses.

```
#include <iostream>
#include <locale>
using namespace std;
int main ( )
{
    setlocale(LC_ALL, "Portuguese");
    float num, soma=0;    // número lido e soma dos números lidos
    int cont=0;           // contador
    cout << "Digite um número qualquer (-1 para sair): ";
    cin >> num;
    if (num != -1)         // para não ser somado o -1
        do{
            soma = soma + num;    // soma do número lido
            cont = cont + 1;      // contagem dos números lidos
            cout << "Digite um número qualquer (-1 para sair): ";
            cin >> num;
        } while (num != -1);      // condição contrária ao Flag (num == -1)
    // saída de dados
    cout << "Foram lidos " << cont << " números " << endl;
    if (cont > 0)
        cout << "A média aritmética dos números é " << soma/cont;
    return 0;
}
```

Exemplo 6: Elabore um programa que determine e imprima a tabuada do número 7.

a) Usando for

```
int i;  
for(i=1; i<=10; i++)  
    cout<<i<<" x 7= "<<i*7<<endl;
```

b) Usando while

```
int i=1;  
while(i<=10)  
{  
    cout<<i<<" x 7= "<<i*7<<endl;  
    i++;  
}
```

c) Usando do-while

```
int numero;  
numero = 1;  
do  
{  
    cout << numero << " x 7 = " << numero*7 << endl;  
    numero = numero + 1;  
} while (numero <= 10);
```

Exercício Proposto:

Escreva um programa para calcular o fatorial de um valor N fornecido

Obs.: N deve ser um valor inteiro e ≥ 0 .

a) Usando a estrutura while();

b) Usando a estrutura do-while();

c) Usando a estrutura for().