

## 6ª Aula prática

### Estrutura de repetição – Do/While

Tal como a estrutura **While**, permite que um comando (ou uma seqüência de comandos) sejam executados repetidamente, enquanto uma condição for satisfeita. A diferença entre **While** e **Do/While** está na execução do teste:

- **While** executa o teste e, caso o resultado seja verdadeiro, executa o comando ou seqüência de comandos internos. Neste caso, se o teste for falso já na primeira avaliação, não haverá execução alguma;
- **Do/While** executa uma vez o comando ou seqüência de comandos internos e só então realiza o teste. Neste caso fica garantida ao menos uma execução do comando/seqüência de comandos.

Na linguagem C/C++, a estrutura de **Do/While** tem a seguinte forma:

```
do
    comando;
while(condição)
```

Exemplo 1 : programa para exibir uma seqüência de números ímpares.

```
#include <iostream>
using namespace std;
main()
{
    int x=1;
    do {
        cout << x << " " << endl;
        x = x + 2;
    } while ( x < 10 );
    system("pause");
}
```

Exemplo 2 : Foi aberta uma campanha de arrecadação de doações para custear o transplante de medula de uma criança. O valor necessário é R\$10.000,00. O programa abaixo lê os valores das doações que forem feitas até que a quantia necessária seja alcançada, imprimindo o total obtido e o número de doadores.

```
#include <iostream>
using namespace std;
main()
{
    int cont;
    float doacao, soma;
    soma = 0.0;
    cont = 0;
    do{
```

```

    cout << "Digite o valor da doacao: ";
    cin >> doacao;
    soma = soma + doacao;
    cont++; // ou cont = cont + 1;
} while ( soma < 10000.00 );
cout << "Valor total obtido: " << soma << endl;
cout << "Numero de doadores: " << cont << endl;
system("pause");
}

```

Exemplo 3 : Cálculo de uma aproximação do valor de  $e^x$  através dos 20 primeiros termos da série

$$e^x = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} \dots$$

```

#include <iostream>
#include <math.h> // contém funções matemáticas
using namespace std;
main()
{
    float x, fat, e=1.0;
    int cont=1;
    cout << "Digite o expoente: "; cin >> x;
    do {
        cout << endl;
        fat = 1.0;
        for ( int i=2; i<=cont; i++ )
            fat = fat * i;
        e = e + pow(x,cont)/fat;
        cont++;
    } while (cont < 20); // controle do numero de termos da série

    cout << "O valor de e^" << x << " e igual a " << e << endl;
    system("pause");
}

```

Exercícios:

- 1) Por que, no exemplo 3, a variável fat foi declarada do tipo float se o fatorial é sempre um número inteiro?
- 2) Modifique o exemplo 3 de forma a aproveitar o cálculo do fatorial de um número para simplificar o cálculo do fatorial do seu sucessor.
- 3) Modifique o exemplo 3 para que o valor de  $e^x$  calculado tenha uma precisão de 0.000001. A avaliação da precisão pode ser feita calculando-se a diferença entre duas aproximações sucessivas.
- 4) Calcule uma aproximação do valor do  $\cos(x)$  através dos 20 primeiros termos da serie abaixo:

$$\text{COSSENO}(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} \dots$$