

# Linguagens de Programação

Aula Teórica 3

Estruturas de Controle em C++

# Estruturas de Controle

**Estrutura de Controle** é um bloco de programação que analisa variáveis e escolhe uma direção através dos parâmetros definidos.

O conceito de estruturas de controle estudadas na linguagem de programação Java, aplica-se também na linguagem C++.

# Comandos de Decisão

Os comandos de decisão, permitem fazer uma escolha, a partir de uma ou mais alternativas da acção que se deve tomar. A selecção é baseada no valor de uma expressão de controle. Um valor pode ser testado através do comando **If, if else, if aninhados e switch.**

# If, If Else

O if é empregue tendo como objectivo o teste de uma condição e uma instrução a ser executada.

```
if (condição)  
comando;
```

O if else é empregue quando se tem mais condições de teste e mais de uma instrução a ser executada.

```
if (condição)  
comando;
```

```
Else
```

```
Comando;
```

# If, If Else

A declaração `else` é opcional, pode-se utilizar para determinar um conjunto de comandos que serão executados caso a condição testada seja falsa.

É possível também aninhar **ifs**, ou seja, fazer uma declaração **if** dentro de outra declaração **if** anterior, este é um método bastante útil na programação.

# Ifs Encadados

```
if ( x > 10)
    { if ( x == 17)
      {
        cout << "x é maior que 10 e igual a 17";}
      else
      {
        cout<< "x é maior que 10 mas não é igual a 17";

      }
    }
    else {
      cout << "x é menor do que 10";}

    return 0;
}
```

# Switch

A declaração switch é uma maneira fácil de se fazer uma tomada de decisão com múltiplas escolhas. Na declaração switch, a variável é sucessivamente testada contra uma lista de inteiros ou constantes caracteres. Quando uma associação é encontrada, o conjunto de comandos associado com a constante é executado.

# switch

```
switch (val) {  
  case <valor inteiro 1> : <comando>  
    break;  
  case <valor inteiro 2> : <comando>  
    break;  
  default : <comando>  
    break;
```



# Estruturas de Repetição

Os comandos de repetição especificam a execução de *loops*.

*Em C++, um loop pode ser determinado através de **While, Do While e For**.*

# While

O comando **while** *antes de cada possível execução a <expressão\_de\_controle> é avaliada.*  
O exemplo abaixo ilustra o comando *while*:

```
int contador = 1;  
while ( contador <= 100 ) {  
    cout << contador << "\n";  
    contador ++; }
```

# While

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

# Do While

A declaração do while é parecida com a declaração while, a diferença fundamental reside no facto de o teste condicional ser feito após a execução dos comandos pertencentes ao laço.

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

# For

Utiliza-se o comando **for** para realizar tarefas repetitivas dentro de um programa.

```
for ( valor_inicial; condição_testada;  
valor_incremento)  
{  
comandos;  
}
```

# Exercícios

1. Faça um programa que pede ao usuário para introduzir um número, o programa deve informar se o número está entre 0 à 10, 11 à 1000 ou 1000 para frente. Use switch e if.
2. Faça um programa que tenha um ciclo while. O programa só deve terminar depois que o usuário pressionar 0.
3. Imprima na tela uma tabuada de 1 à 10.
4. Faça um programa para calcular factoriais

# Exercicios

5. Faça um programa que consoante o valor de entrada, imprima na tela o número de estrelas (asteriscos).
6. Faça um programa que calcule a potência, sendo os dados de entrada (base e expoente) introduzidos pelo usuário.

# Exercicios

7. Elabore um programa que pede os lados de um triângulo e informa se o triângulo é equilátero, isósceles ou escaleno.
8. Faça um programa que simule uma calculadora que realiza as 4 operações básicas.
9. Tendo 4 marcas de carros diferentes, escolher a marca que deseja, após a escolha, deve informar o ano e o País de fabricação.



# Exercicios

10. Faça um programa que consoante um valor inteiro de entrada, visualize na saída o numero de asteristicos em função do valor digitado.

11. Faça um programa que visualiza os números múltiplos de 5 entre 50 e 100.

12. Faça um programa que calcule a soma dos primeiros 50 números pares. Explore as duas possibilidades, recebendo e não os valores por teclado.

# Exercicios

13. Faça um programa que pede a idade de 5 pessoas e visualiza as idades que forem maiores que 18.

14. Faça um programa que visualiza:

a) Números de 500 a 800.

b) Números impares de 30 a 60 e faz o somatório desses valores.

c) Contagem de 2 a 2 de 30 a 60.

d) Números impares de 100 a 50.

# Exercícios

15. Elabore um programa que retorne a tabuada (1-10) de um numero escolhido pelo usuário.
16. Faca um programa que pede ao usuário que introduza n valores, o programa deve retornar na saida os valores ímpares existentes no intervalo.
17. Faca um programa que recebe n números inteiros. O programa deve calcular a media dos valores, indicar o maior e quais deles são primos.

# Exercícios

18. Faça um programa que consoante o valor de entrada, imprima na tela o número de estrelas (asteriscos). Nota: usar ciclo for.
19. Faça um programa que tenha um ciclo while. O utilizador introduz vários números, o programa só deve termina depois que o utilizador pressiona zero (0).
20. Faça um programa que calcula isoladamente a área e perímetro de triângulo isóscele, equilátero e escaleno.

# FIM