

ESTRUTURA DE UM PROGRAMA EM C++

Estrutura de um Programa em C++

```
#include<biblioteca.h>
```

```
void main()
```

```
{
```

Este trecho é reservado para o corpo da função, com a declaração de suas variáveis locais, seus comandos e funções de trabalho;

```
}
```

- Void – definição de de um parâmetro (argumento) não conhecido;
- Main() – a primeira e principal função a ser executada;
- { - inicia ou abre o corpo da função (programa), definição do inicio de um bloco;
- } – finaliza ou fecha o corpo da função, definição da finalização de um bloco;

Exemplo:

```
#include <iostream.h>
```

```
void main ()
```

```
{
```

```
    cout << "olá, mundo";
```

```
}
```

#Include – Tem por finalidade incluir no programa o arquivo de cabeçalho **iostream** (uma biblioteca), o qual possui armazenado o recurso do fluxo de saída **cout**;

void main () – estabelece o inicio do bloco principal do programa;

cout – responsável para solicitar a impressão na tela;

<< - (dois sinais de menor) símbolo utilizado para a transferência da informação, pelo cout, para o monitor;

Meu primeiro programa:

```
#include <iostream.h> //inclui a biblioteca iostream – para as funções de leitura e escrita no console.
```

```
#include <conio.h> //inclui a biblioteca conio – para as funções especiais.
```

```
void main() //função principal do programa, ela quem faz o programa funcionar
```

```
{ //inicio do bloco
```

```
    clrscr(); //limpa a tela do console
```

```
    cout<<"meu primeiro programa"; // imprime no console a frase que esta entre "".
```

```
    getch(); //pausa o programa aguardando algum comando.
```

```
} //Fim do bloco
```

Esse programa permite que seja impresso na tela a escrita **meu primeiro programa**. Os itens depois do “//” são comentários e não influenciam no funcionamento do programa.

Tabela de comparativos

Em Algoritmo	Em C++	Descrição do Comando em C++
Algoritmo	void main()	Função principal do Programa, ela é quem faz o programa rodar.
Início	{	Início de bloco
Fim	}	Fim de bloco
Escreva	Cout	Console Out, ou seja, imprime algo na tela para a visualização do usuário.
Leia	Cin	Console In, ou seja aguarda algum comando do usuário
	<<	Atribui o valor a ser impresso pelo comando Cout
	>>	Atribui o valor recebido do CIN a variável
	Getch()	Pausa o programa, e fica aguardando a interação com o usuário
	Clrscr()	Clear Screen – Limpa a tela do console.
numérico	Int	Variáveis numéricas com valores inteiros, sem a virgula
	float	Variáveis numéricas com valores reais, com a virgula
Literal	Char[n]	Variáveis que recebe caracteres alfanuméricos. N é a quantidade máxima da caracteres a serem recebidos.
Se	if	Comando de condicional
Para	for	Comando de repetição para
Enquanto	while	Comando de repetição enquanto
	Do while	Comando de repetição enquanto

Ao iniciar uma variável em algoritmo era necessário inserir a palavra Declara e depois informar o tipo de variável. Na programação em C++ não é necessário, basta colocar o tipo da variável conforme tabela acima, os tipos de variáveis são tratados abaixo.

Tipos de variáveis:

int numérico inteiro, recebe valores de -32.768 à 32.768 dependendo do PC -2.147.483.648 à 2.147.483.648

float numérico real, recebe valores com virgula, de $-3,4^{38}$ à $3,4^{38}$

double numérico real, recebe valores com virgula, de $-1,7^{308}$ à $1,7^{308}$

char recebe caracteres de texto.

O uso de variáveis:

Em programação variável é uma região da memória do computador, previamente identificada, que tem por finalidade armazenar as informações (dados) de um programa temporariamente. Uma variável armazena apenas um valor por vez. É considerado valor qualquer conteúdo armazenado em uma variável. Um valor está relacionado ao tipo de dado de uma variável, podendo ser numérico, lógico ou caractere.

O nome de uma variável é utilizado para sua identificação e posterior uso dentro de um programa. Sendo assim, é necessário estabelecer algumas regras para escolhê-lo:

- Nomes de variável poder ser atribuídos com um ou mais caracteres;
- O primeiro caractere do nome de uma variável não pode ser um número, sempre deve ser letras;
- O nome de uma variável não pode ter espaços em branco;
- Não pode ser nome de uma variável uma palavra reservada a uma instrução ou identificador da linguagem C++;
- Não pode ser utilizados outros caracteres, a não ser letras e números, com exceção do caractere underline “_”, que pode ser utilizado para simular a separação de duas palavras, como NOME_ALUNO.

Operadores Aritméticos

Operador	Operação	Resultado
+	Soma	Inteiro ou real
-	Subtração	Inteiro ou real
%	Resto	Inteiro ou real
/	Divisão	Inteiro ou real
*	Multiplicação	Inteiro ou real
++	Incremento	Inteiro ou real
--	Decremento	Inteiro ou real
pwo(base,expoente)	Exponenciação	Real
sqrt(valor numérico)	Raiz Quadrada	Real

Para a utilização de pwo() e sqrt(), é necessário incluir a biblioteca Include <math.h>.

Ao usar uma expressão aritmética, deve-se separar as funções entre () para que o calculo seja executado corretamente.

Exemplo:

A área do triangulo fica sendo $A = (B*H)/2$

Media aritmética de 3 números, $M = (N1+N2+N3)/3$

Controle de fluxo de entrada e Saída:

Como já vimos, os comandos CIN e COUT, console in e console out respectivamente, tem por objetivo receber e enviar informações ao console do computador, funcionando como um comunicador entre o computador e o usuário do programa.

Exemplo 1 de programa que utilizando os comandos cin (entrada) e cout (saída): Este programa deve receber 2 números, calcular e mostrar a soma dos números.

```
#include <iostream.h>           //Biblioteca para CIN e COUT
#include <conio.h>               //Biblioteca para funções como clrscr e getch
void main()                    //Estrutura principal do programa
{
    int N1, N2, SOMA;          //declaração de variáveis int – numérica inteiro.
    clrscr();                 //função limpa a tela.

    cout<<"Digite o 1º Número"; //Imprime tudo que esta entre aspas "" na tela do computador
    cin>>N1;                  //Fica aguardando o usuário digitar algo
    cout<<"Digite o 2º Número"; //Imprime tudo que esta entre aspas "" na tela do computador
    cin>>N2;                  //Fica aguardando o usuário digitar algo

    SOMA = N1+N2;             //a variável soma recebe o valor da soma entre N1 e N2
    cout<<"O resultado da Soma é: "<<SOMA; //Imprime tudo que esta entre aspas "" na tela do
//computador o que esta depois o sinal << é a variável soma, e será impresso na tela o valor da variável.
    getch();                  //função pausa programa e aguarda um comando do teclado.
}                             //finaliza o bloco da função principal do programa
```

Exemplo 2: Uso de Variável Char

```
#include <iostream.h>           //Biblioteca para CIN e COUT
#include <conio.h>               //Biblioteca para funções como clrscr e getch
void main()                    //Estrutura principal do programa
{
    char NOME[40];            //declaração de variável char – literal.
    clrscr();                 //função limpa a tela.

    cout<<"Digite seu nome"; //Imprime tudo que esta entre aspas "" na tela do computador
    cin>>NOME;                //Fica aguardando o usuário digitar algo, neste caso o nome
    cout<<"Olá, "<<NOME;       //Imprime tudo que esta entre aspas "" na tela do computador, o que esta
//depois o sinal << é a variável soma, e será impresso na tela o valor da variável.
    getch();                  //função pausa programa e aguarda um comando do teclado.
}
```

Neste segundo exemplo, esta sendo usada uma variável denominada **NOME** com a capacidade de armazenamento de armazenar até **40** caracteres. Observe que o tipo da variável está sendo definido com **char**. O tipo de dado **char** determina apenas um caractere do respectivo tipo, ou seja, se fosse **definido char NOME**, a variável **NOME** somente aceitaria um caractere. O uso do valor **40** no exemplo apresentado proporciona que seja reservado na memória RAM de seu computador o espaço adequado para os **40** caracteres solicitados.

Ao executar o programa ele aceita apenas sequências de caracteres que não possuem espaço em branco. Se um espaço em branco for utilizado, tudo o que estiver escrito após o espaço será desconsiderado pelo programa. Esta é uma característica de funcionamento do recurso e não um defeito operacional dele.

O exemplo 3 mostra um programa que solicita o nome e o sobrenome de uma pessoa.

Exemplo 3:

```
#include <iostream.h>           //Biblioteca para CIN e COUT
#include <conio.h>              //Biblioteca para funções como clrscr e getch
void main()                   //Estrutura principal do programa
{
    char NOME[40], SOBRENOME[40]; //declaração de variável char – literal.
    clrscr();                  //função limpa a tela.

    cout<<"Digite seu nome e sobrenome";//Imprime tudo que esta entre aspas "" na tela do computador
    cin>> NOME >> SOBRENOME;    //Fica aguardando o usuário digitar algo, neste caso o nome

    cout<<"Olá, "<<NOME<<" "<<SOBRENOME; //Imprime tudo que esta entre aspas "" na tela do
    computador, o que esta //depois o sinal << é a variável soma, e será impresso na tela o valor da
    variável.
    getch();                   //função pausa programa e aguarda um comando do teclado.
}
```

Quando o usuário digitar um espaço o programa passara a gravar os valores digitados na segunda variável. Porém se o usuário digitar 3 nomes separados por espaço o terceiro nome será perdido pelo programa.

No exemplo 4 é possível analisar outra forma de receber caracteres sem se preocupar com espaços.

Exemplo 4:

```
#include <iostream.h>
#include <conio.h>
void main()
{
    char NOME[50];
    clrscr();

    cout<<"Digite seu nome e sobrenome";
    cin.getline(NOME, sizeof (NOME));

    cout<<"Olá, "<<NOME;
    getch();
}
```

Isso é possível com a utilização da função **getline()** pertencente ao fluxo de entrada **cin**. Observe que **getline()** efetua a leitura de caracteres até que um retorno de carro ocorra (ao pressionar a tecla **<Enter>**) ou até atingir o valor máximo de caracteres permitidos. Neste contexto a função **sizeof()** retorna o tamanho da variável **NOME**, que no caso do exemplo é de 50. A função **sizeof()** calcula o tamanho de uma variável, seja ela do tipo que for.

Exercícios:

- 1) Faça um programa que receba 4 (quatro) números, calcule e mostre a soma dos números.**
- 2) Faça um programa que receba 3 (três) notas, calcule e mostre a média aritmética entre elas**
- 3) Faça um programa que receba 3 (três) notas e seus respectivos pesos, calcule e mostre a média ponderada dessas notas.**
- 4) Faça um programa que receba o nome e o salário de um funcionário, calcule e mostre o nome do funcionário e seu novo salário, sabendo-se que este sofreu um aumento de 15%.**
- 5) Faça um programa que receba o nome, o salário de um funcionário e o percentual de aumento, calcule e mostre o novo salário.**
- 6) Faça um programa que receba o salário-base de um funcionário, calcule e mostre o salário a receber, sabendo que esse funcionário tem a gratificação de 5% sobre o salário-base e paga imposto de 7% sobre o salário-base.**
- 7) Faça um programa que calcule e mostre a Área de um triângulo. Área do triângulo = $(\text{Base} \times \text{Altura})/2$**
- 8) Faça um programa que receba o Ano de Nascimento de uma pessoa e o ano atual, calcule e mostre: A idade Atual dessa pessoa e quantos anos essa pessoa terá em 2016.**
- 9) Pedro Comprou um saco de Ração com peso em quilos. Pedro possui 2 (dois) gatos para os quais fornece a quantidade de ração em gramas. Faça um programa que receba o peso do Saco de ração e a quantidade de ração fornecida para cada gato. Calcule e mostre quanto restará de ração no saco após 5 (cinco) dias.**

Estrutura Condicional

Operadores Relacionais

As estruturas de condicionais foram desenvolvidas para a tomada de decisão, onde, para tomar uma decisão, é necessário ter uma condição. A condição é a relação existente entre elementos, podendo ser **variável** com **variável** ou **variável** com **constante**.

A relação existente entre os elementos de uma condição é conseguida com a utilização dos operadores relacionais, de acordo com a tabela seguinte:

Símbolo	Significado
==	Igual a
!=	Diferente de (não igual)
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a

Exemplos: $A==B$, $A!=b$, $A>B$, ...

Desvio condicional

Os desvios condicionais de um programa estão associados à utilização da tomada de decisões em um programa de computador. Uma decisão a ser tomada pode ser verdadeira ou falsa. Se verdadeira, pode ser executada uma determinada ação; se falsa, pode ou não ser executada outra ação; portanto, um desvio condicional pode ser simples ou composto.

Condicional Simples

Desvia a execução do programa caso o resultado lógico seja verdadeiro, adentrando em um bloco específico, caso o resultado lógico seja avaliado como falso, não acontece nada.

Em C++ é definido através da instrução **if**, conforme exemplo:

```
if (condição)
{
    Conjunto de instruções
}
```

Para exemplificar melhor o uso, temos o seguinte problema resolvido. Programa calcula e mostra a média aritmética do aluno que teve 2 avaliações, e caso o aluno seja aprovado com nota acima de 7,0 deve apresentar a mensagem "Parabéns pelo resultado!".

```
#include <iostream.h>
#include <conio.h>
void main()
{
    float N1, N2, MEDIA;           //declaração de variáveis float – numérica real.

    clrscr();                      //função limpa a tela.
    cout<<"Digite a 1ª Nota";
    cin>>N1;
    cout<<"Digite a 2ª Nota";
    cin>>N2;

    MEDIA = (N1+N2)/2;
    cout<<"O aluno obteve a média: "<<MEDIA;

    if (MEDIA>7)
    {
        cout<<"Parabéns pelo resultado!";
    }
    getch();
}
```

No exemplo acima o programa será executado normalmente, onde recebe as duas notas e mostra o resultado, porém se o aluno obteve nota maior que 7 o programa executa as instruções que estão no bloco {}, caso o aluno não entre nesta condição o programa não faz nada.

Condicional Composto

Anteriormente foi visto como usar a instrução **if** simples para avaliar uma determinada condição (condicional simples). Agora tratamos o uso das instruções **if... else**.

Numa instrução **if...else** se a condição for **verdadeira**, será executada a instrução que estiver posicionada entre a instrução **if** e a **else**. Sendo a condição **falsa**, será executada a instrução que estiver posicionada logo após a **else**, como é indicado na seguinte sintaxe:

```
if (condição)
{
    conjunto de instruções
}
else
{
    conjunto de instruções
}
```

Obs. Caso após a instrução **if** ou **else**, seja necessário executar somente 1 linha de instruções, não há a necessidade de utilizar as chaves {}.

Para exemplificar consideramos o problema anterior da média do aluno, onde caso a nota for inferior a 7 não acontecia nada e agora colocamos a seguinte mensagem "Se empenhe mais!!!"

```
#include <iostream.h>
#include <conio.h>
void main()
{
    float N1, N2, MEDIA;           //declaração de variáveis float – numérica real.

    clrscr();                      //função limpa a tela.
    cout<<"Digite a 1ª Nota";
    cin>>N1;
    cout<<"Digite a 2ª Nota";
    cin>>N2;

    MEDIA = (N1+N2)/2;
    cout<<"O aluno obteve a média: "<<MEDIA;

    if (MEDIA>7)
        cout<<"Parabéns pelo resultado!";
    else
        cout<<" Se empenhe mais!!!";

    getch();
}
```

Conjunção

Neste caso usamos mais de um teste na condição para a tomada de decisão.

Operadores E = &&
Tabela lógica:

Condição 1	Condição 2	Resultado
Falsa	Falsa	Falsa
Verdadeira	Falsa	Falsa
Falsa	Verdadeira	Falsa
Verdadeira	Verdadeira	Verdadeira

Operadores ou = ||
Tabela lógica:

Condição 1	Condição 2	Resultado
Falsa	Falsa	Falsa
Verdadeira	Falsa	Verdadeira
Falsa	Verdadeira	Verdadeira
Verdadeira	Verdadeira	Verdadeira

Para exemplificar melhor o uso, temos o seguinte problema resolvido. Programa calcula e mostra a média aritmética do aluno que teve 2 avaliações, e caso o aluno seja aprovado com nota entre 7,0 e 8,0 deve apresentar a mensagem “**Continue assim você teve um bom resultado!**”, caso a nota do aluno seja maior que 8 aparece “**Parabéns pelo resultado!**”, caso a nota do aluno esteja entre 6 e 7 aparece “**Aprovado(a)!**” se menor que 6 aparece a mensagem “**Reprovado!**”.

```
#include <iostream.h>
#include <conio.h>
void main()
{
    float N1, N2, MEDIA;           //declaração de variáveis float – numérica real.
    clrscr();                       //função limpa a tela.
    cout<<"Digite a 1ª Nota";
    cin>>N1;
    cout<<"Digite a 2ª Nota";
    cin>>N2;
    MEDIA = (N1+N2)/2;
    cout<<"O aluno obteve a média: "<<MEDIA;

    if (MEDIA<6)
        cout<<"Reprovado";
    if (MEDIA>6 && MEDIA<7)
        cout<<"Aprovado!";
    if (MEDIA>=7 && MEDIA<8)
        cout<<"Continue assim você teve um bom resultado!";
    if (MEDIA>=8 && MEDIA<10)
        cout<<"Parabéns pelo resultado!";
    getch();
}
```

Resolvendo o problema utilizando o **else**:

```
#include <iostream.h>
#include <conio.h>
void main()
{
    float N1, N2, MEDIA;           //declaração de variáveis float – numérica real.
    clrscr();                       //função limpa a tela.
    cout<<"Digite a 1ª Nota";
    cin>>N1;
    cout<<"Digite a 2ª Nota";
    cin>>N2;
    MEDIA = (N1+N2)/2;
    cout<<"O aluno obteve a média: "<<MEDIA;

    if (MEDIA<6)
        cout<<"Reprovado";
    else
    {
        if (MEDIA>6 && MEDIA<7)
            cout<<"Aprovado!";
        else
        {
            if (MEDIA>=7 && MEDIA<8)
                cout<<"Continue assim você teve um bom resultado!";
            else
            {
                if (MEDIA>=8 && MEDIA<10)
                    cout<<"Parabéns pelo resultado!";
                else
                    cout<<"resultado invalido";
            }
        }
    }
    getch();
}
```


Exercícios

- 1) Faça um programa que receba 2 números e mostre o maior.
- 2) Faça um programa que receba 3 números e mostre em ordem crescente.
- 3) Faça um programa que receba 2 números e mostre o menor.
- 4) Faça um programa que receba 3 números e mostre em ordem decrescente.
- 5) Uma empresa decide dar um aumento de 30% aos funcionários com salários inferiores a R\$ 600,00. Faça um algoritmo que receba o salário do funcionário e mostre o valor do salário reajustado ou uma mensagem, caso o funcionário não tenha direito ao aumento.
- 6) Faça um programa que verifique a validade de uma senha fornecida pelo funcionário. A senha é 4531. O algoritmo deve mostrar uma mensagem de permissão de acesso ou não.
- 7) Faça um programa que receba a idade de uma pessoa e mostre a mensagem de Maioridade ou não.
- 8) Faça um programa que receba a Altura e o sexo de uma pessoa e que calcule e mostre o seu peso ideal, sabendo que: para **homes** $(72,7 \times H) - 58$ e para **mulheres** $(62,1 \times H) - 44,7$