



INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA EM LINGUAGEM C

Conteúdo desta aula:

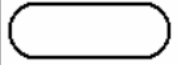
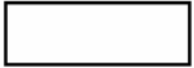
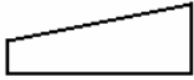


- Revisão sobre tópicos da aula 2 e solução de dúvidas
- Diagrama de blocos (Fluxograma)
- Identificadores
- Tipos básicos em C
- Operadores matemáticos
- Exercícios e compreensão das mensagens de erro



INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA EM LINGUAGEM C

Diagrama de blocos (fluxograma):

O diagrama de blocos é uma forma padronizada e eficaz para representar os passos lógicos de um determinado processamento. Com o diagrama podemos definir uma sequência de símbolos, com significado bem definido, portanto, sua principal função é a de facilitar a visualização dos passos de um processamento.

Símbolo	Função
 TERMINAL	Indica o INICIO ou FIM de um fluxo de dados
 PROCESSAMENTO	Indica o processamento de informações Exemplo: $SOMA = A + B$
 ENTRADA	Indica a entrada de dados através do teclado
 SAÍDA	Mostra mensagens ao usuário: informações, solicitações ou resultados.
 CONECTOR	Indica o sentido do fluxo de dados e conecta símbolos e/ou blocos.



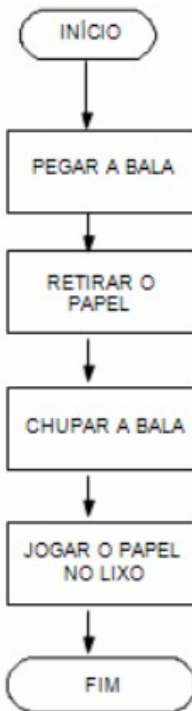
INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA EM LINGUAGEM C

Algoritmo vs Fluxograma:

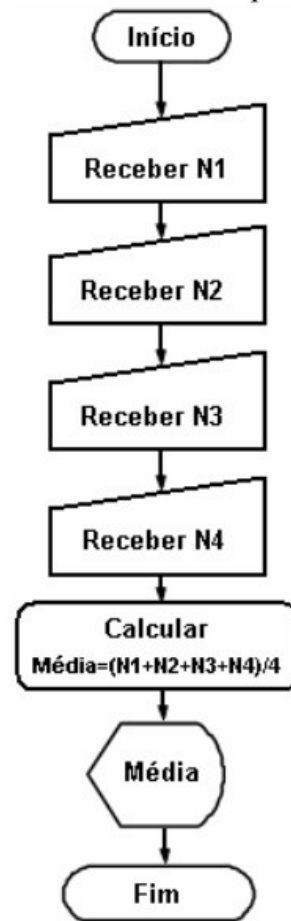
“Chupar uma bala”.

1. Pegar a bala
2. Retirar o papel
3. Chupar a bala
4. Jogar o papel no lixo

“Chupar uma bala”



“Calcular uma média com quatro notas”



O que há de errado aqui ?



INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA EM LINGUAGEM C

Identificadores:

São os nomes que podem ser dados para constantes, variáveis e funções.

Para a escolha destes nomes é necessário seguir algumas regras:

- Um identificador deve iniciar por uma letra ou por um "_" (underscore);
- A partir do segundo caracter pode conter letras, números e underscore;
- Deve-se usar nomes significativos dentro do contexto do programa;
- C é uma linguagem case-sensitive, ou seja, faz diferença entre nomes com letras maiúsculas e nomes com letras minúsculas, por isso, Peso e peso são elementos diferentes;
- Costuma-se usar maiúsculas e minúsculas para separar palavras: "pesoDoCarro";
- Deve ser diferente dos comandos da linguagem;
- Pode conter números a partir do segundo caractere;

Exemplos: nome, _sinalizador, volumeDoPortaMalas, Usuario_1, raioDoCirculo



INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA EM LINGUAGEM C

Variável:

- É uma posição de memória que pode ser acessada através de um identificador
- Pode ter seu conteúdo alterado por um comando de atribuição
- Após a atribuição muda o valor armazenado
- Toda variável é declarada ANTES de sua utilização

Tipos de Variáveis

- Todas as variáveis em C tem um tipo
- Cada tipo define os valores que a variável pode armazenar
- Cada tipo ocupa uma certa quantidade de memória

Tipo	Valores Válidos
char	letras e símbolos: 'a', 'b', 'H', '^', '*', '1', '0'
int	de -32767 até 32767 (apenas números inteiros)
float	de -3.4×10^{38} até $+3.4 \times 10^{38}$ com até 6 dígitos de precisão
double	de -1.7×10^{308} até $+1.7 \times 10^{308}$ com até 10 dígitos de precisão

```
1 // duas barras indicam uma linha de comentario
2 #include <stdio.h>
3 /* Isto é um bloco de comentario
4    escrito em mais de uma linha
5    sem usar as barras em cada linha escrita
6 */
7 int main() {
8     int a,b,c;
9     // a recebe o valor 3
10    a = 3;
11    // b recebe o dobro do valor de a = 6
12    b = a * 2;
13    // c recebe o resultado da soma = 11
14    c = a + b + 2; // c recebe 11
15
16    return 0;
17 }
18
19
```



INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA EM LINGUAGEM C

Constante:

- Constantes são identificadores que não podem ter seus valores alterados durante a execução do programa
- Para criar uma constante existe o comando **#define** que, em geral é colocado no início do programa-fonte

```
1  // Não se coloca ponto-e-vírgula após o valor da constante
2  #define LARGURA_MAXIMA 50
3  #define DIAS_DA_SEMANA 7
4  #define HORAS_POR_DIA 24
5  #define pi 3.1415
6  #include <stdio.h>
7
8  int main (void)
9  {
10     int TotalDeHoras = 0;
11     TotalDeHoras = 10 * DIAS_DA_SEMANA * HORAS_POR_DIA;
12     printf("Total de horas: %d ", TotalDeHoras);
13     // mostra Total de horas: 1680
14     printf("\nPI: %f", pi);
15     // mostra PI: 3.141500
16     return 0;
17 }
18
```



INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA EM LINGUAGEM C

String:

- É uma sequência de caracteres entre aspas duplas: "exemplo de string"

```
1  #include <stdio.h>
2  void main ()
3  {
4      printf("%s","Isto é uma string...\n");
5      // O '\n' serve para avançar para linha seguinte;
6      printf("%s","Strings são textos entre aspas");
7      printf("%s"," - Terceira string");
8      /* mostra:
9      Isto é uma string...
10     Strings são textos entre aspas - Terceira string
11     */
12 }
```



INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA EM LINGUAGEM C

Operadores matemáticos:

* multiplicação

/ divisão

+ soma

- subtração

Quando for necessário realizar algum cálculo que não siga a sequência natural das Operações aritméticas, deve-se usar parênteses para calcular os conjuntos

Exemplo:

```
int a=3, b=9, c=5, resultado;
```

```
resultado = a + b * c; // 48
```

```
Resultado = (a+b) *c; // 60
```




INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA EM LINGUAGEM C

Exercícios:

1. Escreva um programa que recebe duas notas, calcula a média e mostra essa média
2. Escreva um programa que calcule a área de um círculo. Considere um raio inicializado com o valor 5 e pi declarado como uma constante 3.1416. Qual o tipo necessário para a variável **resultado** funcionar adequadamente?
Pesquise como calcular o resultado utilizando a função pow(base, expoente)

```
resultado = pi * pow(raio, 2);
```