



Modularização

ALGORITMOS E PROGRAMAÇÃO 2 –
CIÊNCIAS DA COMPUTAÇÃO - UFJ

PROFESSORA: ANA PAULA FREITAS VILELA
BOAVENTURA

O que temos para hoje?

- Introdução;
- Funções;
- Escopo das variáveis (global e local)
- Criação das bibliotecas .h

Já resolvemos
os seguintes
problemas

Calcular a **média aritmética** da nota
obtida pelos alunos de uma
determinada turma da escola

Entrada:

- Número de **alunos** e notas de cada aluno;

Saída:

- Média de cada **turma**.

E, se agora quisermos **calcular a média aritmética** de uma **determinada escola**

Já resolvemos os seguintes problemas

Entrada:

- Número de **turmas** e notas de cada turma;

Saída:

- Média da **escola**.

Que tipos de
conclusões
podemos tirar
dessa situação



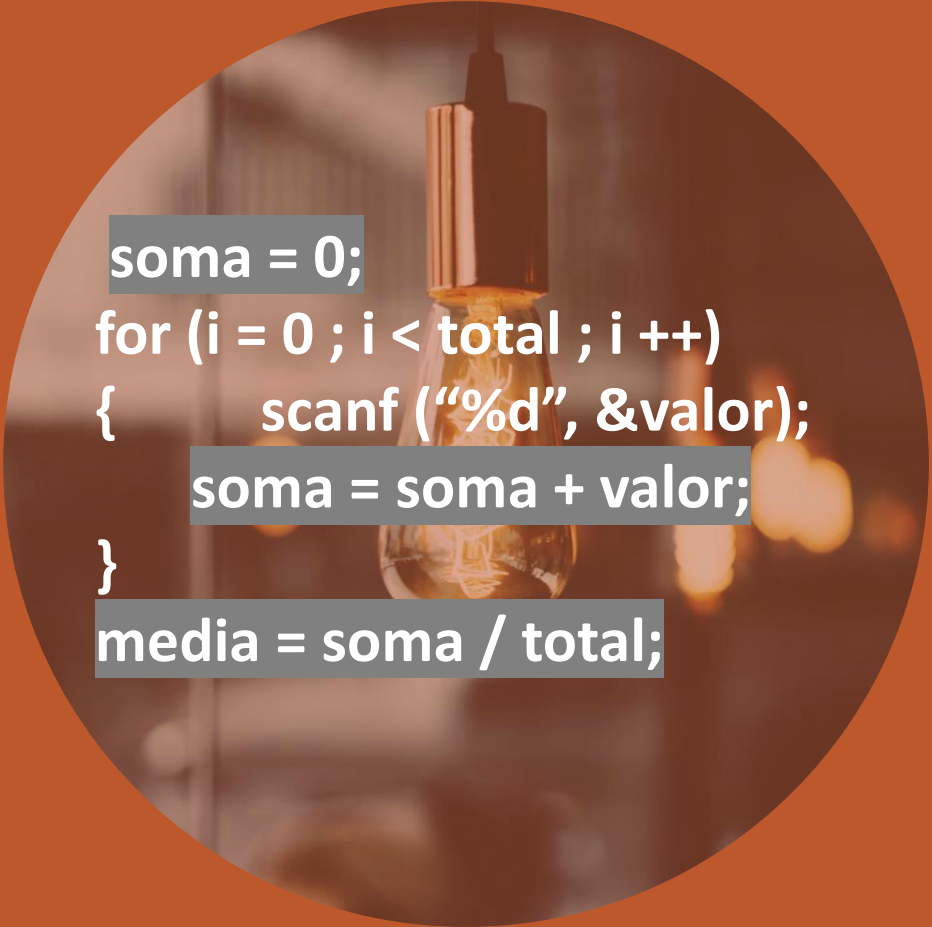
Ideias

1 - A SOLUÇÃO DO PROBLEMA É “FRAGMENTADA”, PRIMEIRO SE CALCULA DE CADA TURMA E DEPOIS DA ESCOLA



Ideias

2 - A SOLUÇÃO DO
PROBLEMA PERPASSA A
MESMA ROTINA:



```
soma = 0;
for (i = 0 ; i < total ; i ++)
{
    scanf ("%d", &valor);
    soma = soma + valor;
}
media = soma / total;
```

Funções



Definição

Situação em que temos um **problema complexo** e o **decompomos** em **subproblemas**;

Dividimos a complexidade e simplifica a solução – **algoritmo modularizado**, pois divide as tarefas em módulos (subalgoritmo);

Cada um desses subproblemas será solucionado por um **subalgoritmo**, que funciona de forma um algoritmo: **entrada – processamento e saída**;

Definição

Logo, o algoritmo completo é dividido num **algoritmo principal** e diversos **subalgoritmos** (tantos quantos forem necessários ou convenientes)

O algoritmo principal é por onde começa a execução, e **chama (ativa / invoca)**, eventualmente, os demais subalgoritmos;

Na linguagem C, o modulo principal é a função *main* ();

Elementos de um subprograma

A definição de um subalgoritmo consta de:

Cabeçalho: definidos o nome e o tipo do subalgoritmo, bem como os seus parâmetros e suas variáveis locais;

Corpo do subalgoritmo: instruções, que serão executadas cada vez que ele é chamado

tipo Nome_Módulo (parâmetros)

início

**conjunto de declarações
ou (variáveis locais)**

instrução 1

instrução 2

...

instrução N

fim

Funções

As **funções** podem **retornar (ou não) valor**;

Característica útil e amplia as possibilidades de utilização dos módulos;

Caso **não retorne valor**: Módulo com retorno neutro (**void**);

O retorno de valores é a saída do módulo – “resposta” ao módulo ativador;

```
#include <stdio.h>
#include <stdlib.h>
float Media()
{
    int i, total;
    float soma, valor, res;
    soma = 0;
    printf("Digite a quantidade total\n");
    scanf("%d", &total);
    for (i = 1 ; i <= total ; i ++)
    {
        printf("\nDigite o %d valor:",i);
        scanf ("%f", &valor);
        soma = soma + valor;
    }
    res = soma / total;
    return res;
}
```

```
int main()
{
    printf("Média da turma %f", Media( ));
    printf("Média da escola: %f", Media( ));
    getchar();
}
```

Elementos de um subprograma

Elementos de um subprograma

```
#include <stdio.h>
#include <stdlib.h>
```

```
float Media( );//protótipo da função antes
```

```
int main()
{
    printf("Média da turma %f", Media( ));
    printf("Média da escola: %f", Media( ));
    getchar();
}
```

```
float Media()
{
    int i, total;
    float soma, valor, res;
    soma = 0;
    printf("Digite a quantidade total\n");
    scanf("%d", &total);
    for (i = 1 ; i <= total ; i++)
    {
        printf("\nDigite o %d valor:",i);
        scanf ("%f", &valor);
        soma = soma + valor;
    }
    res = soma / total;
    return res;
}
```

Escopo das variáveis

Módulos = subprogramas;

Ideia: funções independentes entre si;

Como vimos no exemplo, algumas variáveis foram criadas na função `media()`;

As variáveis terão validade em virtude do local em que foram declaradas:
variáveis locais e globais;

```
#include <stdio.h>
#include <stdlib.h>
```

```
int Aux; /variável global
```

```
float Media()
{
    int total; //local da Media( )
    //bloco de instrução
}
```

```
int main()
{
    int i,j; //local da main( )
    // bloco de instrução
}
```

Variáveis
locais x
globais

Atenção!!!

É possível haver **coincidência nos identificadores de variáveis globais e locais** – não é uma boa prática de programação, sugestão em iniciar com letras maiúsculas as variáveis globais;

Embora com o mesmo nome, tais variáveis referem a **locais diferentes de memória**;

Variáveis
locais x
globais

E se tivermos uma variável global e local com o mesmo nome (idênticas), como o compilador saberá qual delas está sendo acessada (local ou global?)



Variáveis locais x globais

Durante a execução da função ativada, a **variável local é válida** e a **global** fica temporariamente **inacessível**;

Fora da função referida, o identificador aponta exclusivamente para a **variável global**;

O **tempo de vida** das variáveis locais é curto, visto que começam a existir no início da execução da função e encerram ao final da função – evita conflito entre variáveis (com o mesmo nome);

Incluindo módulos em arquivos de cabeçalho header (tipo .h)

Nome do arquivo: “var.h”

Nome do arquivo: “exemplo.c”

```
#include <stdio.h>
#include <stdlib.h>
```

```
#include "var.h"
```

```
void main()
{
    printf("Média da turma %f", Media( ));
    printf("Média da escola: %f", Media( ));
    getchar();
}
```

```
float Media( )
{
    int i, total;
    float soma, valor, res;
    soma = 0;
    printf("Digite a quantidade total\n");
    scanf("%d", &total);
    for (i = 1 ; i <= total ; i ++ )
    {
        printf("\nDigite o %d valor:", i);
        scanf ("%f", &valor);
        soma = soma + valor;
    }
    res = soma / total;
    return res;
}
```

Assistam aos vídeos

Canal: Programação descomplicada linguagem C

<https://youtu.be/OrF2ydZIELk> (Função)

Canal: De aluno para aluno

<https://youtu.be/Y19q6rgM9eo> (funções)

<https://youtu.be/Ewe0IHVMedQ> (Veremos mais detalhes na próxima aula – passagem de parâmetro por valor)

https://youtu.be/WNT9a_ulqeE (variáveis locais)

<https://youtu.be/zZy48jJ1bs8> (variáveis locais estáticas)

https://youtu.be/Xviupfbd_8g (variável global)

Bóson Treinamentos

<https://youtu.be/QWMmuZtcDIg> (Head files)