

Sumário da aula 6 e 7

- Resolução de uma ficha formativa no ambiente de programação **Dev-C++** para aplicação prática dos conceitos abordados nas aulas anteriores.



CORREÇÃO DA FICHA FORMATIVA

Questão 1

a) A programação modular é uma forma de dividir o programa em partes menores chamados módulos ou subprogramas. Cada módulo é responsável por realizar uma tarefa específica dentro do programa de forma a desenvolver, testar e manter o código de forma independente. É através da programação modular que é possível organizar o código, torná-lo mais legível de forma a facilitar a leitura e também reutilizá-lo sempre que necessário.

- b) Legibilidade, Reutilização de código e facilidade na organização/manutenção do código.
- Legibilidade - O código fica mais organizado e fácil de compreender porque cada módulo possui uma função bem definida.
 - Reutilização do código - Os módulos quando são bem definidos e construídos podem ser utilizados noutros programas ou em diferentes partes do mesmo programa.
 - Facilidade na manutenção – Os módulos podem ser mais facilmente atualizados ou substituídos sem modificar todo o código já escrito.

CORREÇÃO DA FICHA FORMATIVA

Questão 2

- a) Uma função em linguagem C refere-se a um bloco de código independente dentro de um programa que executa uma tarefa específica. A função retorna sempre um valor (return).
- b) Uma função com retorno é uma função que executa uma tarefa que retorna sempre um valor ao programa. O tipo de retorno poderá ser um número inteiro ou um número flutuante entre outros. São utilizadas quando necessitamos de obter um resultado. Exemplo: As funções sem retorno são funções que executam tarefas mas que não retornam valores, como por exemplo a função void.

```
4 int soma(int a, int b) {  
5     return a + b;  
6 }
```

```
3 void exibirMensagem()  
4 {  
5     printf("Ola, mundo!\n");  
6 }
```

CORREÇÃO DA FICHA FORMATIVA

Questão 2

c) Estrutura básica de uma função em linguagem C

```
1 <tipo de dados primitivo e nome da função> (lista de argumentos)
2 {
3     return <valor>;
4 }
```

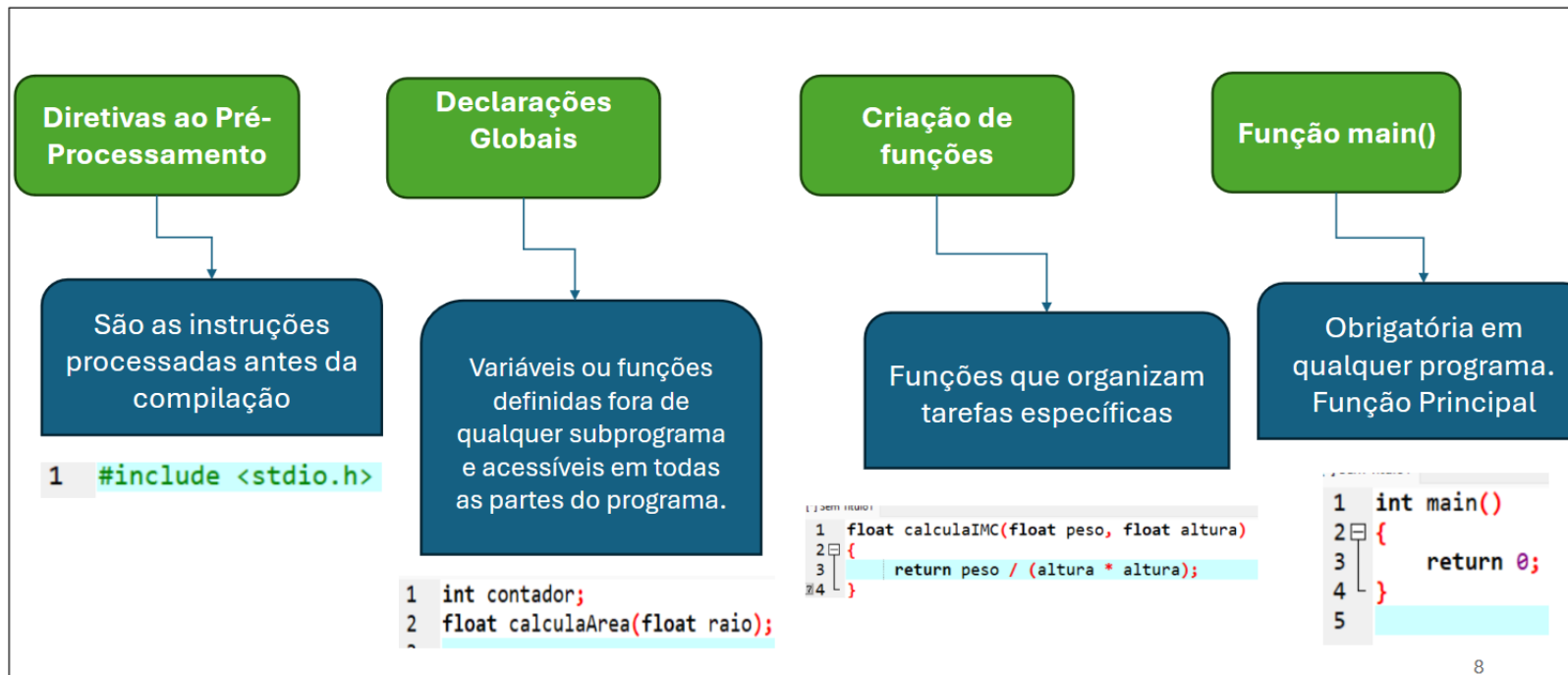
d) Os argumentos de uma função são os valores atribuídos à função quando ela é chamada. Neste exemplo os argumentos da função são os valores atribuídos a “a” e “b”.

```
[*] Sem Título1
1 // Função que calcula a soma
2 int soma(int a, int b) { //
3     return a + b; //
4 }
```

CORREÇÃO DA FICHA FORMATIVA

Questão 2

- e) A estrutura de um programa em C é constituída pelas diretivas ao pré-processador, as declarações globais, a função main() e a declaração de funções:



CORREÇÃO DA FICHA FORMATIVA

Questão 3

Função sem retorno que verifica se um determinado número à escolha do utilizador é par ou ímpar:

```
3 void verificaParImpar(int numero)
4 {
5     if (numero % 2 == 0)
6         printf("O número %d é par.\n", numero);
7     else
8         printf("O número %d é impar.\n", numero);
9 }
```

CORREÇÃO DA FICHA FORMATIVA

Questão 3

Função com retorno que compare a idade média de 2 alunos:

```
10 float calcularIdadeMedia(float idade1, float idade2)
11 {
12     return (idade1 + idade2) / 2.0;
13 }
```

CORREÇÃO DA FICHA FORMATIVA

Questão 3

Corpo da Função int main():

```
14
15 int main()
16 {
17     int opcao;
18     setlocale(LC_ALL, "portuguese");
19     do
20     {
21         printf("\n==== Menu ==== \n");
22         printf("1 - Verificar se um número e par ou ímpar \n");
23         printf("2 - Comparar a idade média de 2 alunos \n");
24         printf("3 - Sair \n");
25         printf("Escolha uma opção: ");
26         scanf("%d", &opcao);
27
28         switch (opcao)
29         {
30             case 1:
31             {
32                 int numero;
33                 printf("\nInsira um número inteiro: ");
34                 scanf("%d", &numero);
35                 verificaParImpar(numero);
36                 break;
37             }
38             case 2:
39             {
40                 float idade1, idade2;
41                 printf("\nInsira a idade do primeiro aluno: ");
42                 scanf("%f", &idade1);
43                 printf("Insira a idade do segundo aluno: ");
44                 scanf("%f", &idade2);
45                 printf("A idade media é %.2f \n", calcularIdadeMedia(idade1, idade1));
46                 break;
47             }
48             case 3:
49                 printf("\nA sair do programa... Obrigado! \n");
50                 break;
51             default:
52                 printf("\nOpcao invalida. Tente novamente. \n");
53         }
54     } while (opcao != 3);
55
56     return 0;
57 }
```

Questão 3 – Função sem Retorno

CORREÇÃO DA FICHA FORMATIVA

[*] Sem Título1

```
1  #include <stdio.h>
2
3  // Função sem retorno para verificar se um número é par ou ímpar
4  void verificarParImpar(int numero) {
5      if (numero % 2 == 0) {
6          printf("O número %d é par.\n", numero);
7      } else {
8          printf("O número %d é ímpar.\n", numero);
9      }
10 }
11
12 int main() {
13     int numero;
14
15     printf("Digite um número para verificar se é par ou ímpar: ");
16     scanf("%d", &numero);
17
18     verificarParImpar(numero); // Chama a função
19
20     return 0;
21 }
22
```

Questão 3 – Função com Retorno

CORREÇÃO DA FICHA FORMATIVA

[*] Sem Título1

```
1  #include <stdio.h>
2
3  // Função com retorno para calcular a idade média entre dois alunos
4  float calcularIdadeMedia(int idade1, int idade2) {
5      return (idade1 + idade2) / 2.0;
6  }
7
8  int main() {
9      int idade1, idade2;
10     float media;
11
12     printf("Digite a idade do primeiro aluno: ");
13     scanf("%d", &idade1);
14
15     printf("Digite a idade do segundo aluno: ");
16     scanf("%d", &idade2);
17
18     // Chama a função calcularIdadeMedia e armazena o retorno
19     media = calcularIdadeMedia(idade1, idade2);
20
21     printf("A idade média dos dois alunos é: %.2f\n", media);
22
23     return 0;
24 }
```

FIM DA AULA 6 e 7

