

Fundamentos de Programação 1

Linguagem C

“Funções”

Slides 7

Prof. SIMÃO

Funções e variáveis globais

```
#include <stdio.h>

float area = -1.0;

void calc_area_quadrado();
void calc_area_triang_ret();

int main()
{
    int opcao = -1;
    printf ("Cálculo de áreas. \n");
    printf ("0 - Área de um quadrado. \n");
    printf ("1 - Área de um triângulo ret. \n");
    printf ("Informe sua opção: \n");
    scanf ("%d", &opcao);
    switch (opcao)
    {
        case 0: {
            calc_area_quadrado ();
        }
        break;
        case 1: {
            calc_area_triang_ret ();
        }
        break;
        default: {
            printf ("Opcao inválida! \n");
        }
    }
    printf ("O calculo da área é: %f \n", area);
    fflush (stdin);
    getchar();
    return 0;
}
```

```
void calc_area_quadrado ()
{
    float lado = 0.0;

    printf ("Informe o valor do lado \n");
    scanf ("%f", &lado);

    area = lado * lado;
}

void calc_area_triang_ret ()
{
    float lado1 = 0.0;
    float lado2 = 0.0;

    printf ("Informe o valor do primeiro lado \n");
    scanf ("%f", &lado1);

    printf ("Informe o valor do segundo lado \n");
    scanf ("%f", &lado2);

    area = (lado1 * lado2) / 2.0;
}
```

Retorno de Funções - Variáveis locais

```
#include <stdio.h>

float calc_area_quadrado();
float calc_area_triang_ret();

int main()
{   int opcao = -1;
    float area = -1.0;

    printf ("Cálculo de áreas. \n");
    printf ("0 - Área de um quadrado. \n");
    printf ("1 - Área de um triângulo ret. \n");
    printf ("Informe sua opção: \n");
    scanf ("%d", &opcao);

    switch (opcao)
    {
        case 0: {
                area = calc_area_quadrado ( );
            }
            break;
        case 1: {
                area = calc_area_triang_ret ( );
            }
            break;
        default: {
                printf("Opcao inválida! \n");
            }
    }

    printf("O cálculo da área é: %f \n", area);
    fflush(stdin);
    getchar();
    return 0;
}
```

```
float calc_area_quadrado()
{
    float lado = 0.0;
    float result = 0.0;

    printf ("Informe o valor do lado \n");
    scanf ("%f", &lado);
    result = (lado * lado);
    return result;
}

float calc_area_triang_ret()
{
    float lado1 = 0.0;
    float lado2 = 0.0;
    float result = 0.0;
    printf ("Informe o valor do primeiro lado \n");
    scanf ("%f", &lado1);
    printf ("Informe o valor do segundo lado \n");
    scanf ("%f", &lado2);
    result = (lado1 * lado2) / 2;
    return result;
}
```

Funções com parâmetros por valor - 1.

```
#include <stdio.h>
float calc_area_quadrado (float);
float calc_area_triang_ret (float, float);

int main()
{   int opcao = -1;
    float area = -1.0;

    printf ("Cálculo de áreas. \n");
    printf ("0 - Área de um quadrado. \n");
    printf ("1 - Área de um triângulo ret. \n");
    printf ("Informe sua opção: \n");
    scanf ("%d", &opcao);

    switch(opcao)
    {
        case 0: { float lado = 0.0;

                printf("Informe o valor do lado \n");
                scanf("%f", &lado);
                area = calc_area_quadrado ( lado );
            }
            break;

        case 1: { float lado1 = 0.0;
                 float lado2 = 0.0;

                printf ("Informe o valor do primeiro lado \n");
                scanf ("%f", &lado1);
                printf ("Informe o valor do segundo lado \n");
                scanf ("%f", &lado2);
                area = calc_area_triang_ret (lado1, lado2);
            }
            break;
    }
}
```

```
default:
    {
        printf ("Opcao inválida! \n");
    }
}

printf("O cálculo da área é: %f \n", area);

fflush(stdin);
getchar();
return 0;
}

float calc_area_quadrado (float lado)
{
    float result = 0.0;
    result = (lado * lado);
    return result;
}

float calc_area_triang_ret (float lado1, float lado2)
{
    float result = 0.0;
    result = (lado1 * lado2) / 2;
    return result;
}
```

Funções com parâmetros por valor - 2.

```
#include <stdio.h>
float calc_area_quadrado (float);
float calc_area_triang_ret (float, float);

int main()
{   int opcao = -1;
    float area = -1.0;

    printf ("Cálculo de áreas. \n");
    printf ("0 - Área de um quadrado. \n");
    printf ("1 - Área de um triângulo ret. \n");
    printf ("Informe sua opção: \n");
    scanf ("%d", &opcao);

    switch(opcao)
    {
        case 0: { float lado = 0.0;

                printf("Informe o valor do lado \n");
                scanf("%f", &lado);
                area = calc_area_quadrado (lado);
            }
            break;

        case 1: { float lado1 = 0.0;
                 float lado2 = 0.0;

                printf ("Informe o valor do primeiro lado \n");
                scanf ("%f", &lado1);
                printf ("Informe o valor do segundo lado \n");
                scanf ("%f", &lado2);
                area = calc_area_triang_ret (lado1, lado2);
            }
            break;
```

```
        default:
            {
                printf ("Opcao inválida! \n");
            }
    }

    printf("O cálculo da área é: %f \n", area);

    fflush(stdin);
    getchar();
    return 0;
}

float calc_area_quadrado (float ld)
{
    float result = 0.0;
    result = (ld * ld);
    return result;
}

float calc_area_triang_ret (float ld1, float ld2)
{
    float result = 0.0;
    result = (ld1 * ld2) / 2;
    return result;
}
```

Funções com parâmetros por valor - 3.

```
#include <stdio.h>
float calc_area_quadrado (float);
float calc_area_triang_ret (float, float);

int main()
{   int opcao = -1;
    float area = -1.0;

    printf ("Cálculo de áreas. \n");
    printf ("0 - Área de um quadrado. \n");
    printf ("1 - Área de um triângulo ret. \n");
    printf ("Informe sua opção: \n");
    scanf ("%d", &opcao);

    switch(opcao)
    {
        case 0: { float lado = 0.0;

                printf("Informe o valor do lado \n");
                scanf("%f", &lado);
                area = calc_area_quadrado (lado);
                printf("O lado vale: %f \n", lado);
            }
            break;

        case 1: { float lado1, lado2;

                printf ("Informe o valor do primeiro lado \n");
                scanf ("%f", &lado1);
                printf ("Informe o valor do segundo lado \n");
                scanf ("%f", &lado2);
                area = calc_area_triang_ret (lado1, lado2);
            }
            break;
    }
}
```

```
default:
    {
        printf ("Opcao inválida! \n");
    }
}

printf("O cálculo da área é: %f \n", area);

fflush ( stdin );
getchar();
return 0;
}

float calc_area_quadrado (float ld)
{
    ld = ( ld * ld );

    return ld;
}

float calc_area_triang_ret (float ld1, float ld2)
{
    float result = 0.0;
    result = (ld1 * ld2) / 2;
    return result;
}
```

Funções com parâmetros por retorno -1.

```
#include <stdio.h>
void calc_area_quadrado (float, float *);
void calc_area_triang_ret (float, float, float *);

int main()
{   int opcao = -1;
    float area = -1.0;

    printf("Cálculo de áreas. \n");
    printf("0 - Área de um quadrado. \n");
    printf("1 - Área de um triângulo ret. \n");
    printf("Informe sua opção: \n");
    scanf("%d", &opcao);

    switch (opcao)
    {
        case 0: {
            float lado = 0.0;
            printf("Informe o valor do lado \n");
            scanf("%f", &lado);
            calc_area_quadrado (lado, &area);
        }
        break;
        case 1: {
            float lado1 = 0.0;
            float lado2 = 0.0;
            printf ("Informe o valor do primeiro lado \n");
            scanf ("%f", &lado1);
            printf ("Informe o valor do segundo lado \n");
            scanf ("%f", &lado2);
            calc_area_triang_ret (lado1, lado2, &area);
        }
        break;
    }
}
```

```
if ( ( 0 == opcao) || ( 1 == opcao ) )
{
    printf ("O cálculo da área é: %f \n", area);
}

fflush ( stdin );
getchar();
return 0;
}

void calc_area_quadrado (float lado, float *area)
{
    *area = (lado * lado);
}

void calc_area_triang_ret (float lado1, float lado2, float *area)
{
    *area = (lado1 * lado2) / 2;
}
```

Funções com parâmetros por retorno - 2.

```
#include <stdio.h>
void calc_area_quadrado (float, float *);
void calc_area_triang_ret (float, float, float *);

int main()
{   int opcao = -1;
    float area = -1.0;

    printf("Cálculo de áreas. \n");
    printf("0 - Área de um quadrado. \n");
    printf("1 - Área de um triângulo ret. \n");
    printf("Informe sua opção: \n");
    scanf("%d", &opcao);

    switch (opcao)
    {
        case 0: {
            float lado = 0.0;
            printf("Informe o valor do lado \n");
            scanf("%f", &lado);
            calc_area_quadrado (lado, &area);
        }
        break;
        case 1: {
            float lado1 = 0.0;
            float lado2 = 0.0;
            printf ("Informe o valor do primeiro lado \n");
            scanf ("%f", &lado1);
            printf ("Informe o valor do segundo lado \n");
            scanf ("%f", &lado2);
            calc_area_triang_ret (lado1, lado2, &area);
        }
        break;
    }
}
```

```
if ( ( 0 == opcao) || ( 1 == opcao ) )
{
    printf ("O cálculo da área é: %f \n", area);
}

fflush ( stdin );
getchar();
return 0;
}

void calc_area_quadrado (float lado, float *ar)
{
    *ar = (lado * lado);
}

void calc_area_triang_ret (float lado1, float lado2, float *ar)
{
    *ar = (lado1 * lado2) / 2.0;
}
```

Funções com parâmetros por retorno – 3 (alternativo para C++).

```
#include <stdio.h>
void calc_area_quadrado (float, float &);
void calc_area_triang_ret (float, float, float &);

int main()
{   int opcao = -1;
    float area = -1.0;

    printf ( "Cálculo de áreas. \n" );
    printf ( "0 - Área de um quadrado. \n" );
    printf ( "1 - Área de um triângulo ret. \n" );
    printf ( "Informe sua opção: \n" );
    scanf ( "%d", &opcao);

    switch (opcao)
    {
        case 0: {
                float lado = 0.0;
                printf ( "Informe o valor do lado \n" );
                scanf ( "%f", &lado );
                calc_area_quadrado ( lado, area );
            }
            break;
        case 1: {
                float lado1 = 0.0;
                float lado2 = 0.0;
                printf ( "Informe o valor do primeiro lado \n" );
                scanf ( "%f", &lado1 );
                printf ( "Informe o valor do segundo lado \n" );
                scanf ( "%f", &lado2 );
                calc_area_triang_ret ( lado1, lado2, area );
            }
            break;
    }
}
```

```
if ( ( 0 == opcao ) || ( 1 == opcao ) )
{
    printf ("O cálculo da área é: %f \n", area);
}

fflush ( stdin );
getchar();
return 0;
}

void calc_area_quadrado ( float lado, float &ar )
{
    ar = ( lado * lado );
}

void calc_area_triang_ret ( float lado1, float lado2, float &ar
)
{
    ar = ( lado1 * lado2 ) / 2;
}
```

Nesta versão, usa-se o & na frente do parâmetro por referência na função... enquanto na versão anterior usava-se um * na frente do parâmetro tanto quanto em qq utilização dele na função e ainda se fazia necessário usar um & na chamada da função, na frente da variável relacionada ao parâmetro.

Isto só funciona em C++...
Não funciona em C puro

Funções com parâmetros por retorno — 3 (alternativo para C++).

```
#include <stdio.h>
void calc_area_quadrado (float, float &);
void calc_area_triang_ret (float, float, float &);

int main()
{
    int opcao = -1;
    float area = 1.0;

    printf ( "Cálculo de áreas. \n" );
    printf ( "0 - Área de um quadrado. \n" );
    printf ( "1 - Área de um triângulo ret. \n" );
    printf ( "Informe sua opção: \n" );
    scanf ( "%d", &opcao);

    switch (opcao)
    {
        case 0: {
            float lado = 0.0;
            printf ( "Informe o valor do lado \n" );
            scanf ( "%f", &lado );
            calc_area_quadrado ( lado, area );
        }
        break;
        case 1: {
            float lado1 = 0.0;
            float lado2 = 0.0;
            printf ( "Informe o valor do primeiro lado \n" );
            scanf ( "%f", &lado1 );
            printf ( "Informe o valor do segundo lado \n" );
            scanf ( "%f", &lado2 );
            calc_area_triang_ret ( lado1, lado2, area );
        }
        break;
    }
}
```

```
    if ( ( 0 == opcao ) || ( 1 == opcao ) )
    {
        printf ( "O cálculo da área é: %f \n", area );
    }

    fflush(stdin);
    getch();
    return 0;
}

void calc_area_quadrado ( float lado, float &ar )
{
    ar = ( lado * lado );
}

void calc_area_triang_ret ( float lado1, float lado2, float &ar )
{
    ar = ( lado1 * lado2 ) / 2;
}
```

Nesta versão, usa-se o & na frente do parâmetro por referência na função... enquanto na versão anterior usava-se um * na frente do parâmetro tanto quanto em qq utilização dele na função e ainda se fazia necessário usar um & na chamada da função, na frente da variável relacionada ao parâmetro.

Isto só funciona em C++...
Não funciona em C puro

Exercício 1

- Elaborar um programa que calcule o valor de PI por meio de uma função específica para tal. O valor de PI, na função, deve ser calculado utilizando a seguinte série:

$$S = 1 - \left(+\frac{1}{3^3} - \frac{1}{5^3} + \frac{1}{7^3} - \dots \right) \quad \text{sendo PI} = \sqrt[3]{S \times 32}$$

- O número de termos da série deverá ser maior que 10 e informado pelo usuário. Este número servirá de parâmetro para a função em questão.
- Certamente, na elaboração, deve ser considerada a qualidade da solução (e.g. uso apropriado das estruturas algorítmicas e **não** uso de variáveis globais), sua consistência (e.g. variáveis validadas) e sua 'elegância' (e.g. caligrafia adequada e indentação).

Exercício 2

- Elaborar um programa que permita receber as notas de cada aluno de uma turma de n alunos, sendo n informado pelo usuário e não podendo ultrapassar 30. O programa deve ser capaz de
 - Receber as notas das seis (6) parciais de cada aluno, bem como calcular e informar a sua média (ma)
 - Para cada aluno, identificado por um número único (contador), informar:
 - “Reprovado” se média (ma) menor que 5.0 (isto é, $ma \geq 0$ e $ma < 5.0$)
 - “Exame” se média (ma) entre 5.0 e 6.9 (isto é, $ma \geq 5.0$ e $ma \leq 6.9$).
 - “Aprovado” se média (ma) maior que 6.9 (isto é, $ma > 6.9$ e $ma \leq 10.0$).
 - Os dois tópicos acima devem estar em uma função que retorna o valor da média do aluno. Esta função (bem como qualquer outra função da resolução deste exercício) não deve fazer uso de variável global.
 - Calcular a média geral da turma (mt).
 - Calcular e informar a maior média (mar) obtida na turma.
 - Calcular e informar a menor e 2ª menor média obtida na turma.
 - Os cálculos dos dois tópicos acima devem ser feito por meio de um conjunto de funções, sendo que estas não poderão ter comunicação com o usuário (e.g. não utilizar comando ‘printf’ neles). Assim sendo, eles deverão utilizar parâmetros apropriados e/ou retorno (cf. o caso).

Exercício 3

- 1) Faça um programa que leia as coordenadas de um quadrado (x_1, y_1) (x_2, y_2) (x_3, y_3) (x_4, y_4) e calcule o perímetro do quadrado formado pelas coordenadas lidas através de uma função que retorna o perímetro do quadrado para o programa principal.

- 2) Elabore um programa que leia um valor e imprima:
 - ✓ se o valor é par ou ímpar
 - ✓ se é divisível por 5
 - ✓ seu fatorial.
 - ✓ sua tabuada.

Obs.: Utilize uma função para executar cada uma das tarefas anteriores e utilize somente variáveis locais e parâmetros (não use variáveis globais).

Fazer exercícios outros:

os da apostila, por exemplo.