



# Computação 2

## Aula 4 Funções

Prof<sup>a</sup>. Fabiany  
fabiany1@utfpr.edu.br

# Funções

- *Função* é um subprograma que auxilia o programa principal através da realização de uma determinada subtarefa.
- As funções são chamados dentro do corpo do programa principal como se fossem *comandos*. Após seu término, a execução continua a partir do ponto onde foi chamado.
- É importante compreender que a chamada de uma função simplesmente gera um **desvio provisório no fluxo de execução**.

# Funções

- Uma *função*, além de executar uma determinada tarefa pode retornar um valor para quem a chamou, que é o resultado da sua execução.
- Por este motivo, a chamada de uma função que retorna algum valor, aparece no corpo do programa principal como uma *expressão* e não como um comando.

# Funções

- Cada função, além de ter acesso às variáveis do programa que o chamou (são as variáveis *globais*), pode ter suas próprias variáveis (são as variáveis *locais*), que existem apenas durante sua chamada.

# Funções

- Ao se chamar uma função, também é possível passar-lhe determinadas informações que recebem o nome de *parâmetros* (são valores que, na linha de chamada, ficam entre os parênteses e que estão separados por vírgulas).
- A quantidade dos parâmetros, sua seqüência e respectivos tipos não podem mudar: devem estar de acordo com o que foi especificado na sua correspondente declaração.

# Funções

- Declaração de uma função em C:

```
<tipo-de-dado> <nome-da-função> (<seqüência-de-  
declarações-de-parâmetros>) {  
    //comandos  
}
```

Exemplo:

```
//função principal do C  
int main () {  
    //comandos  
    return 0;  
}
```

Variáveis globais

Variáveis locais

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

```
float lado, area = 0;
```

```
int main()
```

```
{
```

```
int opcao;
```

```
float lado1, lado2;
```

```
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: { printf ("Informe o valor do lado \n");
```

```
scanf ("%f", &lado);
```

```
area = lado * lado; }
```

```
break;
```

```
case 1: { 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; }
```

```
break;
```

```
default: {
```

```
printf ("Opcao inválida! \n");
```

```
}
```

```
}
```

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

```
return 0;
```

```
}
```

# Exemplo – Variável Global

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

```
float area = 0;
```

```
void area_quadrado();
```

```
void area_triang_ret();
```

```
int main()
```

```
{
```

```
    int opcao;
```

```
    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_quadrado ();
```

```
        }
```

```
        break;
```

```
        case 1: {
```

```
            area_triang_ret ();
```

```
        }
```

```
        break;
```

```
        default: {
```

```
            printf ("Opcao inválida! \n");
```

```
        }
```

```
    }
```

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

```
    return 0;
```

```
}
```

```
void area_quadrado ()
```

```
{
```

```
    float lado = 0.0;
```

```
    printf ("Informe o valor do lado \n");
```

```
    scanf ("%f", &lado);
```

```
    area = lado * lado;
```

```
}
```

```
void 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 area_quadrado();  
float area_triang_ret();
```

```
int main()
```

```
{ int opcao;  
  float area = 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 = area_quadrado();  
            }  
    break;
```

```
    case 1: {  
              area = area_triang_ret();  
            }  
    break;
```

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

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

```
}
```

```
float 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 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 area_quadrado (float);
```

```
float area_triang_ret (float, float);
```

```
int main()
```

```
{ int opcao;  
  float lado, lado1, lado2, area = 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: { printf("Informe o valor do lado \n");  
             scanf("%f", &lado);  
             area = area_quadrado (lado);  
           } break;
```

```
    case 1: { printf ("Informe o valor do primeiro lado \n");  
             scanf ("%f", &lado1);  
             printf ("Informe o valor do segundo lado \n");  
             scanf ("%f", &lado2);  
             area = area_triang_ret (lado1, lado2);  
           } break;
```

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

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

```
}
```

```
float area_quadrado (float lado)
```

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

```
float area_triang_ret (float lado1, float lado2)
```

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

# Funções com parâmetros por valor - 1.

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

```
float area_quadrado (float);
```

```
float area_triang_ret (float, float);
```

```
int main()
```

```
{ int opcao;  
  float lado, lado1, lado2, area = 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: { printf("Informe o valor do lado \n");  
              scanf("%f", &lado);  
              area = area_quadrado (lado);  
            } break;
```

```
    case 1: { printf ("Informe o valor do primeiro lado \n");  
              scanf ("%f", &lado1);  
              printf ("Informe o valor do segundo lado \n");  
              scanf ("%f", &lado2);  
              area = area_triang_ret (lado1, lado2);  
            } break;
```

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

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

```
}
```

```
float area_quadrado (float ld)
```

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

```
float 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 area_quadrado (float);
```

```
float 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 = 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 = 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 area_quadrado (float ld)
```

```
{  
  ld = ( ld * ld );  
  
  return ld;  
}
```

```
float area_triang_ret (float ld1, float ld2)
```

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

# Funções com parâmetros por referência

```
#include <stdio.h>
void area_quadrado (float, float *);
void area_triang_ret (float, float, float *);

int main()
{ int opcao;
  float area, lado, lado1, lado2 = 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: { printf("Informe o valor do lado \n");
              scanf("%f", &lado);
              area_quadrado (lado, &area);
            } break;
    case 1: {printf ("Informe o valor do primeiro lado \n");
              scanf ("%f", &lado1);
              printf ("Informe o valor do segundo lado \n");
              scanf ("%f", &lado2);
              area_triang_ret (lado1, lado2, &area);
            } break;
  }
  if ( ( 0 == opcao ) || ( 1 == opcao ) ) {
    printf ("O cálculo da área é: %f \n", area);
  }

  return 0;
}
```

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

void area_triang_ret (float l1, float l2, float *area)
{
  *area = (l1 * l2) / 2;
}
```

# Funções com parâmetros por referência

```
#include <stdio.h>
void area_quadrado (float, float *);
void area_triang_ret (float, float, float *);

int main()
{ int opcao;
  float area, lado, lado1, lado2 = 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: { printf("Informe o valor do lado \n");
              scanf("%f", &lado);
              area_quadrado (lado, &area);
            } break;
    case 1: {printf ("Informe o valor do primeiro lado \n");
              scanf ("%f", &lado1);
              printf ("Informe o valor do segundo lado \n");
              scanf ("%f", &lado2);
              area_triang_ret (lado1, lado2, &area);
            } break;
  }
  if ( ( 0 == opcao ) || ( 1 == opcao ) ) {
    printf ("O cálculo da área é: %f \n", area);
  }

  return 0;
}
```

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

void area_triang_ret (float l1, float l2, float *ar)
{
  *ar = (l1 * l2) / 2;
}
```

# Exercícios

- 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).