



Aula 07: Introdução a Funções

Introdução a Programação

Túlio Toffolo & Puca Huachi
<http://www.toffolo.com.br>

BCC201 – 2019/1
Departamento de Computação – UFOP

Aula Anterior

- Comandos condicionais
- Exemplos diversos
- Exercícios

Aula de Hoje

- 1 Definição de Função
- 2 Exemplos de Funções
- 3 Protótipo da função
- 4 Valor de retorno
- 5 Exercícios

Aula de Hoje

- 1 Definição de Função
- 2 Exemplos de Funções
- 3 Protótipo da função
- 4 Valor de retorno
- 5 Exercícios

O que é uma função?

É um conjunto de instruções desenhadas para cumprir determinada tarefa e agrupadas em uma unidade com um nome para referí-la.

Por que usar funções?

- Reaproveitar o código, ou seja, permitir que outros programadores utilizem seus códigos;
- Modularizar um programa em partes menores;
- Executar uma tarefa que é frequentemente solicitada;
- Aumentar a legibilidade e manutenibilidade do programa;
- O uso de funções geralmente diminui o tamanho do programa;
- Implementar as chamadas UDF (*User Defined Functions*), para complementar as necessidades do programador na execução de tarefas não suportadas pelo ambiente de programação.

Exemplos de usos de funções

- Leitura de um número inteiro positivo;
- Imprimir um valor em um determinado formato;
- Cálculo do fatorial de um número;
- Encontrar o maior entre dois números;

Na verdade, qualquer sequência de instruções que apareça mais de uma vez no código é candidata a ser uma função.

Exemplos de uso de funções

```
1 // função que calcula a raiz quadrada
2 double x = sqrt(y);
3
4 // função para gerar números aleatórios
5 int numero = rand();
6
7 // definição da função principal de um programa
8 int main() { ... }
```

Bibliotecas C/C++ são compostas de funções, de forma a permitir que o programador reaproveite códigos existentes.

Aula de Hoje

- 1 Definição de Função
- 2 Exemplos de Funções**
- 3 Protótipo da função
- 4 Valor de retorno
- 5 Exercícios

Exemplos: Conversão de Temperaturas

Fahrenheit e Celsius são duas escalas usadas para medir a temperatura.

- Desenvolveremos um programa para converter as temperaturas em Celsius para temperaturas equivalentes em Fahrenheit.
- A fórmula para conversão é:

$$F = 1.8 \times C + 32$$

- Onde C é a temperatura em Celsius e F é a temperatura correspondente em Fahrenheit.

Exemplo: conversão de temperaturas

```
1  #include <stdio.h>
2
3  int main()
4  {
5      double tempC, tempF;
6      printf("Conversão Celsius para Fahrenheit\n");
7      printf("(valor menor que -273.15 encerra o programa)\n\n");
8      printf("Temperatura em Celsius: ");
9      scanf("%lf", &tempC);
10
11     if (tempC >= -273.15) {
12         tempF = 1.8 * tempC + 32;
13         printf("%lf graus Celsius = %lf graus Fahrenheit.\n",
14               tempC, tempF);
15     }
16     return 0;
17 }
```

Exemplo de execução

```
1  Conversão de Celsius para Fahrenheit  
2  (valor menor que -273.15 encerra o programa)  
3  
4  Temperatura em Celsius : 100  
5  100 graus Celsius = 212 graus Fahrenheit.
```

Exemplo: conversão de temperaturas usando função

```
1  #include <stdio.h>
2
3  // protótipo da função
4  double celsiusToFahrenheit(double tempCels);
5
6  // método main (principal)
7  int main()
8  {
9      double tempC, tempF;
10     printf("Conversão Celsius para Fahrenheit\n");
11     printf("(valor menor que -273.15 encerra o programa)\n\n");
12     printf("Temperatura em Celsius: ");
13     scanf("%lf", &tempC);
14
15     if (tempC >= -273.15) {
16         tempF = celsiusToFahrenheit(tempC);
17         printf("%lf graus Celsius = %lf graus Fahrenheit.\n",
18                tempC, tempF);
19     }
20     return 0;
21 }
```

Exemplo: conversão de temperaturas usando função

```
1 // definição da função
2 double celsiusToFahrenheit(double tempCels)
3 {
4     double f;
5     f = 1.8 * tempCels + 32;
6     return f;
7 }
```

OU

```
1 // definição da função
2 double celsiusToFahrenheit(double tempCels)
3 {
4     return 1.8 * tempCels + 32;
5 }
```

Exemplo: conversão de temperaturas usando função

Eliminando o protótipo da função:

```
1  #include <stdio.h>
2
3  // definição da função
4  double celsiusToFahrenheit(double tempCels)
5  {
6      return 1.8 * tempCels + 32;
7  }
8
9  // O main fica identico ao do exemplo anterior
10 int main()
11 {
12     ...
13 }
```

Aula de Hoje

- 1 Definição de Função
- 2 Exemplos de Funções
- 3 Protótipo da função**
- 4 Valor de retorno
- 5 Exercícios

Protótipo

Definição Geral de uma Função

```
1  <tipo_retorno> <nome_função>(<lista_declaracao_parametro>)  
2  {  
3      <corpo_função>  
4  }
```

Onde:

- **<tipo_retorno>**: é o tipo do valor que a função retorna; quando a função não retorna nenhum valor utiliza-se a palavra chave **void**.
- **<nome_função>**: é o identificador que nomeia a função.
- **<lista_declaracao_parametro>**: é uma lista, possivelmente vazia, de declarações separadas por vírgulas, dos parâmetros da função.
- **<corpo_função>**: descreve o comportamento da função.

Definição de funções

Exemplo: Definição da função `celsiusToFahrenheit()`

tipo do retorno da função

identificador do nome da função

lista de parâmetros:

`tempCels`

```
double celsiusToFahrenheit(double tempCels);  
{  
    return 1.8 * tempCelsius + 32;  
}
```

Corpo da função

Escopo de variáveis

As variáveis só existem no bloco onde foram declaradas.

- No exemplo anterior, as seguintes variáveis foram utilizadas:
- `int main()`
 - `tempC`, `tempF`
- `double celsiusToFahrenheit(double)`
 - `tempCels`

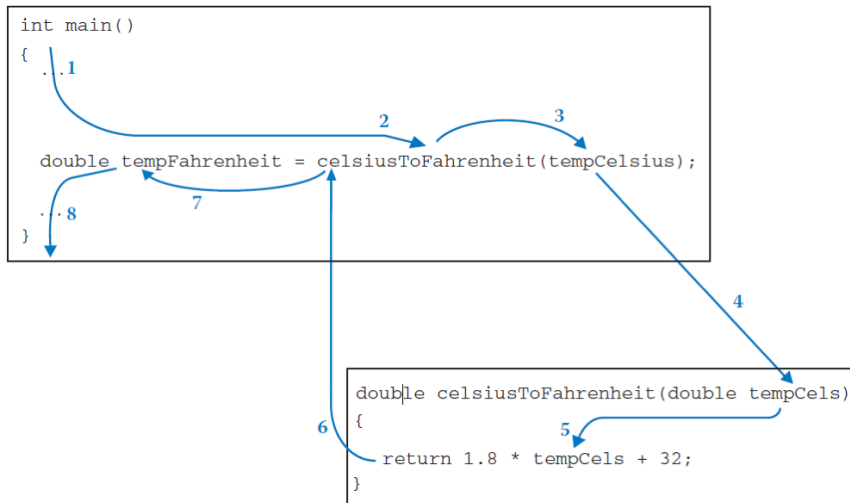
As variáveis `tempC` e `tempF` não podem ser usadas na função `celsiusToFahrenheit()`. De forma análoga, `tempCels` não pode ser usada na função `main()`. Essas variáveis são ditas **locais**.

Escopo de variáveis

Obs. 1: Mesmo que as variáveis possuam o mesmo nome na `main()` e na `celsiusToFahrenheit()`, que é uma declaração correta, o compilador enxerga como variáveis distintas.

Obs. 2: Pode-se declarar variáveis globais, para serem utilizados em todo o programa. Porém, seu uso não é uma boa prática de programação, devendo ser usado apenas quando estritamente necessário.

Fluxo de execução



Aula de Hoje

- 1 Definição de Função
- 2 Exemplos de Funções
- 3 Protótipo da função
- 4 Valor de retorno**
- 5 Exercícios

Exemplo: Função que retorna valor

Função que recebe dois valores e retorna o maior valor.

```
1 // definição da função maior entre 2 números
2 int maior2(int a, int b)
3 {
4     int maior;
5
6     if (a > b)
7         maior = a;
8     else
9         maior = b;
10
11     return maior;
12 }
```

Exemplo: Função que retorna valor

Função que recebe três valores e retorna o maior valor.

```
1 // definição da função maior entre 3 números
2 int maior3(int a, int b, int c)
3 {
4     int maior;
5
6     if ((a > b) && (a > c))
7         maior = a;
8     else {
9         if (b > c)
10            maior = b;
11        else
12            maior = c;
13    }
14    return maior;
15 }
```


Exemplo: Função que não retorna valor

Função que recebe um parâmetro e não retornar nenhum valor.

```
1 // função que imprime um número como moeda
2 void printAsMoney(double n)
3 {
4     printf("R$ %.2lf", n);
5 }
```

Uso:

```
1 int main()
2 {
3     // ...
4     printAsMoney(salario);
5     // ...
6 }
```

Exemplo: Função sem parâmetro e retorno

Função que **não** possui parâmetro e **não** retornar nenhum valor.

```
1 // Implementação da função que toca um beep
2 void beep(void)
3 {
4     printf("\a");
5 }
```

Uso:

```
1 int main()
2 {
3     ...
4     int x;
5     printf("Digite um número positivo: ");
6     scanf("%d", x);
7     if (x < 0)
8         beep();
9     ...
10 }
```

Exemplo: Função com vários parâmetros

Exemplos de protótipos das funções

```
1 // Verifica se os valores formam um triângulo
2 int ehTriangulo(int a, int b, int c);
3
4 // Calcula o valor da prestação de um produto
5 double valorPrestacao(double valor, double taxa, int numParcelas);
6
7 // Recebe 'F' ou 'M' e imprime o sexo por extenso:
8 // "Feminino" ou "Masculino"
9 void printSexo(char s);
```

Exemplo: Uso de funções em argumentos

Chamadas a funções usadas com argumento de outras funções

```
1 // soma dois números
2 int soma (int m, int n)
3 {
4     return m + n;
5 }
6
7 // Soma o quadrado de dois números
8 int somaQuadrado(int a, int b)
9 {
10     return soma(pow(a, 2), pow(b, 2));
11 }
```

Exemplos

Encontre o perímetro de um triângulo, dados os comprimentos de seus três lados. Use uma função para calcular o perímetro.

Obs.: $P = a + b + c$, onde a , b e c são os lados do triângulo.

Exemplos

```
1  #include <stdio.h>
2
3  float perimetro(float, float, float);
4
5  int main()
6  {
7      float a, b, c;
8      printf("Digite os lados do triângulo: ");
9      scanf("%f %f %f", &a, &b, &c);
10
11     float p = perimetro(a, b, c);
12     printf("Perimetro = %f\n", p);
13     return 0;
14 }
15
16 float perimetro(float l1, float l2, float l3)
17 {
18     return l1 + l2 + l3;
19 }
```

Exemplos

Encontre a área de um triângulo, dados os comprimentos dos três lados.
Utiliza a fórmula de Hero:

$$\sqrt{s(s-a)(s-b)(s-c)}$$

onde s é a metade do perímetro (use a função do exemplo anterior).

Exemplos

```
1  #include <stdio.h>
2  #include <math.h>
3
4  float perimetro(float, float, float);
5  float areaTriangulo(float, float, float);
6
7  int main()
8  {
9      float a, b, c;
10     printf("Digite os lados do triângulo: ");
11     scanf("%f %f %f", &a, &b, &c);
12
13     float area = areaTriangulo(a, b, c);
14     printf("Área = %f\n", area);
15 }
16
17 float areaTriangulo(float l1, float l2, float l3)
18 {
19     float s = perimetro(l1, l2, l3) / 2.0;
20     return sqrt(s * (s-a) * (s-b) * (s-c));
21 }
```


Exemplos

Crie uma função que retorna qual o conceito dada uma nota.
Utilize a tabela a seguir:

Conceito	Nota
A	$9 \leq nota \leq 10$
B	$8 \leq nota < 9$
C	$7 \leq nota < 8$
D	$6 \leq nota < 7$
F	$nota < 6$

Exemplos

```
1 // Função que recebe a nota e retorna qual o conceito ('A', 'B', etc.)
2 char conceito(double nota)
3 {
4     if (9 <= nota && nota <= 10)
5         return 'A';
6     else if (8 <= nota)
7         return 'B';
8     else if (7 <= nota)
9         return 'C';
10    else if (6 <= nota)
11        return 'D';
12    else
13        return 'F';
14 }
```

Passagem de Parâmetros

Os parâmetros formais (variáveis locais, declaradas como parâmetro da função chamada) são inicializados com o valor dos parâmetros reais (variáveis passadas como parâmetro).

- **Passagem por valor** – O valor dos parâmetros formais, **se** alterados durante a execução da função **não** acarretarão em nenhuma modificação no valor dos parâmetros reais (variáveis da função chamadora).
- **Passagem por referência** – Qualquer alteração no valor dos parâmetros formais durante a execução da função será refletida no valor de seus parâmetros reais correspondentes.

Obs: Todos os exemplos mostrados até aqui utilizam passagem por valor.

Aula de Hoje

- 1 Definição de Função
- 2 Exemplos de Funções
- 3 Protótipo da função
- 4 Valor de retorno
- 5 Exercícios**

Exercícios

Exercício para entregar

Crie uma função que retorna 1 se o aluno foi aprovado em uma disciplina e 0 caso contrário, considerando que as seguintes informações são passadas como argumentos: (*i*) número total de aulas de uma disciplina, (*ii*) o número de faltas do aluno, e, finalmente, (*iii*) a nota deste aluno.

- Utilize o seguinte protótipo:

```
1 int aprovado(int, int, double);
```



Perguntas?