Engenharia de Computação

Fundamentos de Programação

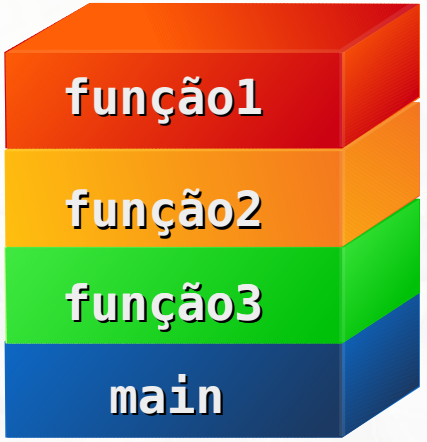
Aula 16 – Funções

Prof. Muriel de Souza Godoi
muriel@utfpr.edu.br

O que é uma função?

- Em programação, uma função é um pedaço de código, dentro de um programa maior, que realiza uma tarefa com uma certa independência do resto do programa;

Função única, grande e difícil de ser entendida



Divide uma tarefa grande em várias tarefas menores

Funções – Porque utilizar?

- **Clareza do código**
  - Separando pedaços de código da função main(), podemos entender mais facilmente o que cada parte do código faz.
- **Reutilização**
  - quando se precisa executar certa tarefa várias vezes ao longo do programa. Utilizando funções não é necessário repetir todo o código várias vezes, além de facilitar a manutenção do código, pois a correção de erros será somente em um lugar.
- **Independência**
  - uma função é relativamente independente do código que a chamou. Uma função pode modificar variáveis globais, mas limitando-se aos dados fornecidos pela chamada de função.

Funções

- **Criando uma função**
  - Toda função deve ser definida antes de sua utilização

```
tipo_retornado nomeFuncao ( parâmetros ){
   /* corpo da função */
   return;
}//fim da função
```

- **Chamadas de funções**
  - Para executar uma função, fazemos uma chamada de função, que é uma instrução composta pelo nome da função:

```
//Função sem parâmetro e sem retorno
nomeFuncao();

//Função passando parametro e salvando o retorno
salvaRetorno = nomeFuncao(parametro);
```

Funções – Exemplo

- Exemplo de uma função sem parâmetros:

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

void mensagem(){// Definição da função
    printf("exemplo de uma funcao simples!\n");
    return;
}//mensagem

int main(){
    printf("Exemplo de chamada de funcao!\n");
    mensagem();// Chamada da função
    return 0;
}//main
```

A execução da função **main** é **suspensa** até a execução completa da função **mensagem**. Após isso, a **main** continua de onde parou;

Funções – Escopo de Variável

- Relembrando:
  - Quanto ao escopo das variáveis, os dois principais tipos são locais e globais;
- **Variáveis locais**
  - São aquelas declaradas dentro do bloco de uma função;
  - Não podem ser usadas ou modificadas por outras funções;
  - Somente existem enquanto a função onde foi declarada estiver sendo executada.
- **Variáveis Globais**
  - São declaradas fora de todos os blocos de funções;
  - São acessíveis em qualquer parte do programa, ou seja, podem ser usadas e modificadas por todas as outras funções;
  - Existem durante toda a execução do programa.

Funções – Escopo de Variável

- Escopo das variáveis em um programa com diversas funções

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

//declaração de variáveis globais

void funcao(parâmetros){//parâmetros têm escopo local
	// declaração das variáveis locais da função
	return;
}//função

int main(){
	//declaração das variáveis locais da main
	return 0;
}//main
```

Funções com parâmetros e retorno

- Os parâmetros e o retorno de devem ter os tipos definidos

```
tipoRet nomeFuncao(tipo param_1, ..., tipo param_n){
   /* corpo da função */
  return retorno;//retorno do tipo definido
}// nomeFuncao
```

- Exemplo:

```
float soma( float n1, float n2) {
  return n1 + n2;
}// soma

int main() {
   float valor = soma(3, 5);//Chama a função
   printf("A soma é %f\n",valor);
   return 0;
}// main
```

Funções com parâmetros e retorno

- **Cabeçalho/ Assinatura** de uma função possui:
  - Tipo do retorno
  - Nome da função
  - Lista de argumentos com tipo

```
float soma( float n1, float n2);
```

- Quando uma função não tem um valor de retorno utiliza-se a palavra void (vazio);
  - void não é um tipo, apenas indica ausência de um valor

Como passar parâmetros para a main?

- **Como passar os valores?**
  - Via linha de comando na execução do programa
- **Exemplo**
  - Em um programa onde o arquivo compilado se chama main e recebe 3 argumentos
  - Deve-se executar com o comando:

```
$./main valor1 valor2 valor3
```

- **Nesse caso ele recebeu 4 argumentos**
  - **1º argumento:** ./main
  - **2º argumento:** valor1
  - **3º argumento:** valor2
  - **4º argumento:** valor3

Por padrão, o primeiro argumento sempre é o nome do arquivo executável

Como passar parâmetros para a main?

- **Como ler os valores?**
  - Cabeçalho correto da função main

```
int main(int argc, char **argv);
```

  - **argc** – quantidade de parâmetros passados (+1)
  - **argv** – valores passados armazenados em uma “matriz”

```
int main(int argc, char **argv){
  printf("Foram passados: %d argumentos\n",argc);
  for (int i = 0; i < argc; i++) {
    printf("O argumento %d é %s\n",i, argv[i]);
  }// for
  return 0;
}// main
```

Como documentar suas funções - DoxyGen

- Antes da função adicione o seguinte comentário:

```
/**
 * Descrição da função
 * \param arg1 descrição do argumento arg1
 * \param arg2 descrição do argumento arg2
 * \return descrição do valor de retorno
 */
```

- **Exemplo**

```
/**
 * Soma dois números reais
 * \param n1 primeiro valor a ser somado
 * \param n2 segundo valor a ser somado
 * \return a soma dos dois valores reais
 */
float soma( float n1, float n2) {
   return n1 + n2;
}// soma
```

Exercícios

- **1)** Faça um algoritmo que implemente uma função que receba 3 números inteiros e retorne o maior valor;
- **2)** Elabore uma função que receba por parâmetro o sexo (caractere) e a altura de uma pessoa (real), calcule e retorne seu peso ideal. Para isso, utilize as fórmulas a seguir.
  - Para homens: (72.7 * altura) - 58
  - Para mulheres: (62.1 * altura) - 44.7
- **3)** Escreva um procedimento que recebe por parâmetro as 3 notas de um aluno e uma letra. Se a letra for A, o procedimento calcula a média aritmética das notas do aluno, se for P, a sua média ponderada (pesos: 5, 3 e 2) e se for S, a soma das notas. O valor calculado também deve ser retornado e exibido na função main.

Em todos os exercícios faça uma função `main` para testar sua função

Exercícios

- **4)** Faça uma função que receba a média final de um aluno por parâmetro e retorne o seu conceito, conforme a tabela abaixo:

Nota	Conceito
[9 - 10]	A
[7 – 9 [	B
[5 - 7 [	C
[0 – 5 [	D

- **5)** Crie uma função que que receba o valor de um inteiro positivo N, calcule e retorne o fatorial desse número.

Em todos os exercícios faça uma função `main` para testar sua função