

# LPG0001 – Linguagem de Programação

## Introdução à Linguagem C

Prof. Rui Jorge Tramontin Junior  
Departamento de Ciência da Computação  
UDESC / Joinville

# Introdução

- A Linguagem C é uma linguagem de alto nível;

# Introdução

- A Linguagem C é uma linguagem de alto nível;
- Porém, também permite a manipulação de aspectos de mais baixo nível, tais como como ponteiros e alocação dinâmica de memória;

# Introdução

- A Linguagem C é uma linguagem de alto nível;
- Porém, também permite a manipulação de aspectos de mais baixo nível, tais como como ponteiros e alocação dinâmica de memória;
- É uma linguagem **compilada**, e os programas gerados em C possuem alto desempenho.

# Estrutura de um Programa

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

```
int main( int argc, char *argv[] ){
```

```
    // Aqui vai o código
```

```
    return 0;
```

```
}
```

# Estrutura de um Programa

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

```
int main( int argc, char *argv[] ){
```

```
// Aqui vai o código
```

```
return 0;
```

```
}
```

- **#include** é uma diretiva de compilação, que faz referência a uma biblioteca de funções;
- É usada para permitir o uso de funções específicas.

# Estrutura de um Programa

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

```
int main( int argc, char *argv[] ){
```

```
    // Aqui vai o código
```

```
    return 0;
```

```
}
```

- **main()** é a função principal, onde vai o código do programa;

# Estrutura de um Programa

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

```
int main( int argc, char *argv[] ) {
```

```
// Aqui vai o código
```

```
return 0;
```

```
}
```

- Os parâmetros da **main()** são usados para receber entrada via console;
- Podem ser omitidos.



# Estrutura de um Programa

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

```
int main( int argc, char *argv[] ){
```

```
// Aqui vai o código
```

```
    return 0;
```

```
}
```

- Aqui vai todo o código, incluindo declaração de variáveis.

# Tipos de Dados

Tipo	Tamanho	Intervalo	Descrição
<i>char</i>	1 byte	-128 a +127	CARACTERE
<i>int</i>	2 ou 4 bytes	-32.768 a 32.767 ou -2.147.483.648 a +2.147.483.647	INTEIRO
<i>float</i>	4 bytes	1.2E-38 a 3.4E+38 (6 casas decimais)	REAL (precisão simples)
<i>double</i>	8 bytes	2.3E-308 a 1.7E+308 (15 casas decimais)	REAL (precisão dupla)
<i>void</i>	1 byte		Não armazena nada

# Declaração de Variáveis

## Pseudocódigo

**X, Y : INTEIRO**

## Linguagem C

**int x, y;**

# Declaração de Variáveis

## Pseudocódigo

**X, Y : INTEIRO**

**N1 : REAL**

## Linguagem C

**int x, y;**

**float n1;**

# Declaração de Variáveis

## Pseudocódigo

**X, Y : INTEIRO**

**N1 : REAL**

**K : CARACTERE**

## Linguagem C

**int x, y;**

**float n1;**

**char k;**

# Declaração de Variáveis

## Pseudocódigo

**X, Y : INTEIRO**

**n1 : REAL**

**K : CARACTERE**

**NOME : CARACTERE**

## Linguagem C

**int x, y;**

**float n1;**

**char k;**

**char nome[20];**

# Formatos (Entrada e Saída)

Especificador de formato	Descrição	Tipos de dados suportados
<i>%c</i>	Caractere	<i>char</i>
<i>%s</i>	Cadeia de caracteres	<i>char [ ]</i>
<i>%i</i> ou <i>%d</i>	Inteiro	<i>int</i>
<i>%f</i>	Real (precisão simples)	<i>float</i>
<i>%lf</i>	Real (precisão dupla)	<i>double</i>

# Entrada e Saída de Dados

- As funções de entrada e saída básicas são:
  - `scanf()` para entrada via teclado;
  - `printf()` para saída no console;



# Entrada e Saída de Dados

- As funções de entrada e saída básicas são:
  - `scanf()` para entrada via teclado;
  - `printf()` para saída no console;
- Ambas são oferecidas pela biblioteca `stdio.h`;

# Entrada e Saída de Dados

- As funções de entrada e saída básicas são:
  - `scanf()` para entrada via teclado;
  - `printf()` para saída no console;
- Ambas são oferecidas pela biblioteca `stdio.h`;
- É necessário fazer o *include* no início do código:

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

# Entrada de Dados

## Pseudocódigo

**LEIA (X)**

## Linguagem C

**scanf ("%i", &x) ;**

# Entrada de Dados

## Pseudocódigo

LEIA (X)

LEIA (N1, K)

## Linguagem C

```
scanf ("%i", &x) ;
```

```
scanf ("%f%c", &n1, &k) ;
```

# Entrada de Dados

## Pseudocódigo

LEIA (X)

LEIA (N1, K)

LEIA (NOME)

## Linguagem C

```
scanf ("%i", &x) ;
```

```
scanf ("%f%c", &n1, &k) ;
```

```
scanf ("%s", &nome) ;
```

# Saída de Dados

## Pseudocódigo

```
ESCREVA ("Nome: ", NOME)
```

## Linguagem C

```
printf("Nome: %s", nome);
```

# Saída de Dados

## Pseudocódigo

```
ESCREVA ("Nome: ", NOME)
```

```
ESCREVAL ("Média entre ", X, " e ", Y, " = ", N1)
```

## Linguagem C

```
printf("Nome: %s", nome);
```

```
printf("Média entre %d e %d = %f\n", x, y, n1);
```

# Saída de Dados

## Pseudocódigo

ESCREVAL ("1-Soma")

ESCREVAL ("2-Subtração")

ESCREVAL ("3-Divisão")



# Saída de Dados

## Pseudocódigo

ESCREVAL ("1-Soma")

ESCREVAL ("2-Subtração")

ESCREVAL ("3-Divisão")

## Linguagem C

```
printf("1-Soma\n2-Subtração\n3-Divisão\n");
```

# Operadores Aritméticos

Pseudocódigo	Linguagem C
+	+
-	-
*	*
/	/
DIV	/
MOD	%

# Atribuição e Oper. Relacionais

Pseudocódigo	Linguagem C
$\leftarrow$	<code>=</code>
<code>=</code>	<code>==</code>
$\neq$	<code>!=</code>
$>$	<code>&gt;</code>
$\geq$	<code>&gt;=</code>
$<$	<code>&lt;</code>
$\leq$	<code>&lt;=</code>

# Operadores Lógicos

Pseudocódigo	Linguagem C
<b>E</b>	<b>&amp;&amp;</b>
<b>OU</b>	<b>  </b>
<b>NAO</b>	<b>!</b>

# Exemplo

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

```
int main() {
```

```
    return 0;
```

```
}
```

# Exemplo

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

```
int main() {
```

```
    float n;
```

```
    return 0;
```

```
}
```

# Exemplo

```
#include <stdio.h>

int main() {
    float n;
    printf("Digite um número: ");

    return 0;
}
```

# Exemplo

```
#include <stdio.h>

int main() {
    float n;
    printf("Digite um número: ");
    scanf("%f", &n);

    return 0;
}
```



# Exemplo

```
#include <stdio.h>

int main() {
    float n;
    printf("Digite um número: ");
    scanf("%f", &n);
    printf("Voce digitou %f\n", n);

    return 0;
}
```

# Estrutura Condicional

## Pseudocódigo

**SE** **x** **=** 0 **ENTAO**

**y** **<-** 0

**FIMSE**

## Linguagem C

**if** ( **x** **==** 0 ) {

**y** = 0;

}

# Estrutura Condicional

## Pseudocódigo

```
SE X = 0 ENTÃO  
    Y ← 0  
FIMSE
```

```
SE X <> 0 ENTÃO  
    Y ← 1  
SENAO  
    Y ← 0  
FIMSE
```

## Linguagem C

```
if ( x == 0 ) {  
    y = 0;  
}
```

```
if ( x != 0 ) {  
    y = 1;  
} else {  
    y = 0;  
}
```

# Exemplo

```
float a, b;
```

# Exemplo

```
float a, b;  
printf("Digite dois números: ");  
scanf("%f%f", &a, &b);
```

# Exemplo

```
float a, b;  
printf("Digite dois números: ");  
scanf("%f%f", &a, &b);  
float media = (a + b) / 2; // declaração e inicialização
```

# Exemplo

```
float a, b;  
printf("Digite dois números: ");  
scanf("%f%f", &a, &b);  
float media = (a + b) / 2; // declaração e inicialização  
if(media >= 7){  
    printf("Aprovado!");  
}
```

# Exemplo

```
float a, b;
printf("Digite dois números: ");
scanf("%f%f", &a, &b);
float media = (a + b) / 2; // declaração e inicialização
if(media >= 7){
    printf("Aprovado!");
}
else{

}
```



# Exemplo

```
float a, b;
printf("Digite dois números: ");
scanf("%f%f", &a, &b);
float media = (a + b) / 2; // declaração e inicialização
if(media >= 7){
    printf("Aprovado!");
}
else{
    if(media >= 1.6){
        printf("Aprovado!");
    }
}
```

# Exemplo

```
float a, b;
printf("Digite dois números: ");
scanf("%f%f", &a, &b);
float media = (a + b) / 2; // declaração e inicialização
if(media >= 7){
    printf("Aprovado!");
}
else{
    if(media >= 1.6){
        printf("Aprovado!");
    }
    else{
        printf("Reprovado!");
    }
}
```

# Estrutura Condicional

## Pseudocódigo

**ESCOLHA** código

**FIMESCOLHA**

# Estrutura Condicional

## Pseudocódigo

**ESCOLHA** codigo

**CASO** 1

**ESCREVA** ("Não perecível")

**FIMESCOLHA**

# Estrutura Condicional

## Pseudocódigo

**ESCOLHA** codigo

**CASO** 1

**ESCREVA**("Não perecível")

**CASO** 2, 3

**ESCREVA**("Perecível")

**FIMESCOLHA**

# Estrutura Condicional

## Pseudocódigo

**ESCOLHA** codigo

**CASO** 1

**ESCREVA**("Não perecível")

**CASO** 2, 3

**ESCREVA**("Perecível")

**CASO** 8 **ATE** 15

**ESCREVA**("Higiene")

**FIMESCOLHA**

# Estrutura Condicional

## Pseudocódigo

**ESCOLHA** codigo

**CASO** 1

**ESCREVA** ("Não perecível")

**CASO** 2, 3

**ESCREVA** ("Perecível")

**CASO** 8 **ATE** 15

**ESCREVA** ("Higiene")

**OUTROCASO**

**ESCREVA** ("Erro!")

**FIMESCOLHA**

# Estrutura Condicional

## Pseudocódigo

```
ESCOLHA codigo
    CASO 1
        ESCREVA("Não perecível")

    CASO 2, 3
        ESCREVA("Perecível")

    CASO 8 ATE 15
        ESCREVA("Higiene")

    OUTROCASO
        ESCREVA("Erro!")
FIMESCOLHA
```

## Linguagem C

```
switch ( codigo ){
```



# Estrutura Condicional

## Pseudocódigo

```
ESCOLHA codigo
    CASO 1
        ESCREVA("Não perecível")

    CASO 2, 3
        ESCREVA("Perecível")

    CASO 8 ATE 15
        ESCREVA("Higiene")

    OUTROCASO
        ESCREVA("Erro!")
FIMESCOLHA
```

## Linguagem C

```
switch ( codigo ){
    case 1:
        printf("Não perecível");
        break;

    }
}
```

# Estrutura Condicional

## Pseudocódigo

```
ESCOLHA codigo
    CASO 1
        ESCREVA("Não perecível")

    CASO 2, 3
        ESCREVA("Perecível")

    CASO 8 ATE 15
        ESCREVA("Higiene")

    OUTROCASO
        ESCREVA("Erro!")
FIMESCOLHA
```

## Linguagem C

```
switch ( codigo ){
    case 1:
        printf("Não perecível");
        break;
    case 2:
    case 3:
        printf("Perecível");
        break;

}
```

# Estrutura Condicional

## Pseudocódigo

```
ESCOLHA codigo
    CASO 1
        ESCREVA("Não perecível")

    CASO 2, 3
        ESCREVA("Perecível")

    CASO 8 ATE 15
        ESCREVA("Higiene")

    OUTROCASO
        ESCREVA("Erro!")
FIMESCOLHA
```

## Linguagem C

```
switch ( codigo ){
    case 1:
        printf("Não perecível");
        break;
    case 2:
    case 3:
        printf("Perecível");
        break;
    case 8 ... 15:
        printf("Higiene");
        break;
}
```

# Estrutura Condicional

## Pseudocódigo

```
ESCOLHA codigo
    CASO 1
        ESCREVA("Não perecível")

    CASO 2, 3
        ESCREVA("Perecível")

    CASO 8 ATE 15
        ESCREVA("Higiene")

    OUTROCASO
        ESCREVA("Erro!")

FIMESCOLHA
```

## Linguagem C

```
switch ( codigo ){
    case 1:
        printf("Não perecível");
        break;
    case 2:
    case 3:
        printf("Perecível");
        break;
    case 8 ... 15:
        printf("Higiene");
        break;
    default:
        printf("Erro!");
}
```

# Exemplos Práticos

1. Leia três valores para os lados de um triângulo. O algoritmo deve verificar se o triângulo é equilátero, isósceles ou escaleno.
2. Leia o nome e o sexo de uma pessoa, apresentando como saída uma das seguintes mensagens: “Ilmo. Sr.”, para o sexo masculino, ou a mensagem “Ilma. Sra.”, para o sexo feminino, seguida do seu nome.