


Prof. esp. Thalles Canela

- **Graduado:** Sistemas de Informação - Wyden Facimp
- **Pós-graduado:** Segurança em redes de computadores - Wyden Facimp
- **Professor:** Todo núcleo de T.I. (Graduação e Pós) - Wyden Facimp
- **Diretor:** SCS
- **Gerente de Projetos:** Motoca Systems

Redes sociais:

- **Linkedin:** <https://www.linkedin.com/in/thalles-canela/>
- **YouTube:** <https://www.youtube.com/aXR6CyberSecurity>
- **Facebook:** <https://www.facebook.com/axr6PenTest>
- **Instagram:** https://www.instagram.com/thalles_canela
- **Github:** <https://github.com/ThallesCanela>
- **Github:** <https://github.com/aXR6>
- **Twitter:** <https://twitter.com/Axr6S>




O que são Estruturas de Dados Heterogêneas?

- Estruturas de dados heterogêneas são aquelas que permitem armazenar diferentes tipos de dados em uma única unidade lógica.
 - Ao contrário das estruturas homogêneas, como arrays, que armazenam um único tipo de dado.
-



Relevância

- Por que aprender sobre Estruturas Heterogêneas?
 - Elas são fundamentais para modelar e resolver problemas complexos no mundo real, onde várias informações relacionadas precisam ser agrupadas.
-



Palavras utilizadas em C nas explicações a cima

- Vamos esclarecer as siglas e os termos utilizados nos códigos de exemplo fornecidos:



C

- Significado: É a extensão padrão para arquivos de código-fonte escritos na linguagem de programação C.



struct

- Significado: Abreviação de "structure" (estrutura em inglês).
- Em C, é uma palavra-chave usada para definir uma estrutura de dados heterogênea, permitindo combinar dados de diferentes tipos.



strcpy

- Significado: Abreviação de "string copy" (cópia de string em inglês).
- É uma função da biblioteca padrão de C (<string.h>) usada para copiar uma string para outra.



printf

- Significado: Abreviação de "print formatted" (imprimir formatado em inglês).
- É uma função da biblioteca padrão de C (`<stdio.h>`) usada para imprimir dados formatados na saída padrão (geralmente a tela).



int

- Significado: Abreviação de "integer" (inteiro em inglês).
- É um tipo de dado em C usado para representar números inteiros.



float

- Significado: É um tipo de dado em C usado para representar números de ponto flutuante de precisão simples.



double

- Significado: É um tipo de dado em C usado para representar números de ponto flutuante de precisão dupla.



char

- Significado: Abreviação de "character" (caractere em inglês). É um tipo de dado em C usado para representar um único caractere.
- Uma array de char geralmente é usada para representar strings.

Aprofundando no conteúdo

Introdução às Structs

Structs em C

- Em C, a estrutura de dados heterogênea mais comum é chamada de "struct". Uma struct permite agrupar variáveis de tipos de dados diferentes.

c

 Copy code

```
struct Pessoa {  
    char nome[50];  
    int idade;  
    double altura;  
};
```

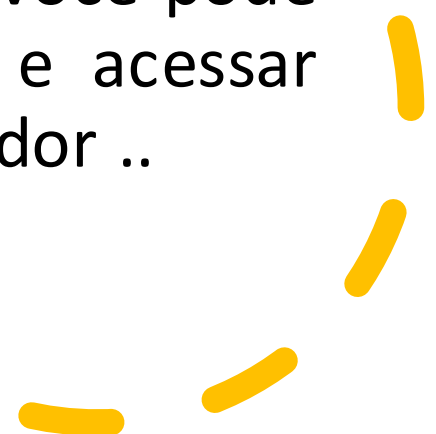
c

 Copy code

```
struct Pessoa pessoa1;  
pessoa1.idade = 25;  
strcpy(pessoa1.nome, "João");  
pessoa1.altura = 1.75;
```

Declarando e Acessando Structs

- Depois de definir uma struct, você pode declarar variáveis desse tipo e acessar seus membros usando o operador ..





Usando Structs com Funções

Structs podem ser passadas como argumentos para funções e também podem ser retornadas por funções.

```
c Copy code  
  
void mostrarPessoa(struct Pessoa p) {  
    printf("Nome: %s\n", p.nome);  
    printf("Idade: %d\n", p.idade);  
    printf("Altura: %.2f\n", p.altura);  
}  
  
struct Pessoa criarPessoa() {  
    struct Pessoa p;  
    strcpy(p.nome, "Maria");  
    p.idade = 30;  
    p.altura = 1.68;  
    return p;  
}
```



```
#include <stdio.h>

struct Livro {
    char titulo[100];    // linha 1
    char autor[50];      // linha 2
    int anoPublicacao;   // linha 3
};

int main() {
    struct Livro meuLivro; // linha 4

    strcpy(meuLivro.titulo, "O Guia do Mochileiro das Galáxias"); // linha 5
    strcpy(meuLivro.autor, "Douglas Adams"); // linha 6
    meuLivro.anoPublicacao = 1979; // linha 7

    printf("Título: %s\n", meuLivro.titulo); // linha 8
    printf("Autor: %s\n", meuLivro.autor); // linha 9
    printf("Ano de Publicação: %d\n", meuLivro.anoPublicacao); // linha 10

    return 0; // linha 11
}
```

Entendendo um pouco mais...

- Exemplo utilizando struct

Explicação - Parte 1:

- **Linha 1:** `char título[100];` - Esta linha declara uma matriz de caracteres chamada `título` com espaço para 100 caracteres. Ela é usada para armazenar o título do livro.
- **Linha 2:** `char autor[50];` - Esta linha declara uma matriz de caracteres chamada `autor` com espaço para 50 caracteres. Ela é usada para armazenar o nome do autor do livro.
- **Linha 3:** `int anoPublicacao;` - Esta linha declara uma variável inteira chamada `anoPublicacao`. É usada para armazenar o ano em que o livro foi publicado.

Explicação - Parte 2:

- **Linha 4:** `struct Livro meuLivro;` - Aqui, uma variável `meuLivro` do tipo `struct Livro` é declarada. Essa variável agora pode armazenar um título, autor e ano de publicação.
- **Linha 5:** A função `strcpy` é usada para copiar a string "O Guia do Mochileiro das Galáxias" para a variável `título` da struct `meuLivro`.
- **Linha 6:** Semelhante à linha 5, a função `strcpy` é usada para copiar a string "Douglas Adams" para a variável `autor` da struct `meuLivro`.

Explicação - Parte 3:

- **Linha 7:** O ano de publicação, 1979, é atribuído à variável `anoPublicacao` da struct `meuLivro`.
- **Linha 8:** Esta linha imprime o título do livro armazenado na variável `meuLivro`.
- **Linha 9:** Esta linha imprime o nome do autor armazenado na variável `meuLivro`.

Explicação - Parte 4:

- **Linha 10:** Imprime o ano de publicação armazenado na variável meuLivro.
- **Linha 11:** Retorna 0, indicando que o programa terminou com sucesso.


Estruturas Heterogêneas vs. Homogêneas

- Enquanto arrays (estruturas homogêneas) armazenam múltiplos itens do mesmo tipo, as structs (estruturas heterogêneas) podem armazenar itens de diferentes tipos.





Vamos
praticar!

- 
- Crie uma struct chamada "Carro", que deve conter os membros: marca (string), ano (inteiro) e preço (float).
 - Declare uma variável desse tipo, atribua valores aos membros e exiba-os.



Significados:

- Os códigos em C frequentemente fazem uso de diversas bibliotecas padrão e outras bibliotecas auxiliares que são denotadas por várias siglas.
- Abaixo, é fornecida uma lista explicativa para algumas das siglas mais comuns que podem ser encontradas em códigos C.

Bibliotecas em C – Part 1

- `stdio.h`: "Standard Input/Output Header". Essa biblioteca contém funções para entrada e saída, como `printf()` e `scanf()`.
- `stdlib.h`: "Standard Library Header". Inclui funções relacionadas a alocação de memória, controle de processos, conversões e outras. Funções como `malloc()`, `free()`, `exit()` e `atoi()` estão presentes aqui.
- `string.h`: Fornece várias funções para manipulação de strings, como `strcpy()`, `strcat()`, `strlen()` e `strcmp()`.
- `math.h`: Contém várias funções matemáticas como `pow()`, `sqrt()`, `sin()`, `cos()`, etc.

Bibliotecas em C – Part 2

- `ctype.h`: Oferece funções que classificam e convertem caracteres, como `isdigit()`, `isalpha()`, `toupper()`, e `tolower()`.
- `stdbool.h`: Introduzido no C99, fornece um tipo `bool` e as macros `true` e `false`.
- `stdint.h`: Também introduzido no C99, fornece um conjunto de tipos de inteiros de largura fixa.
- `time.h`: Contém funções para manipular data e hora, como `time()`, `ctime()`, e `strftime()`.

Bibliotecas em C – Part 3

- `signal.h`: Fornece maneiras de lidar com sinais (interrupções) como `SIGINT`, `SIGTERM`, etc.
- `setjmp.h`: Usado para manipular o ambiente do programa e realizar saltos não locais usando `setjmp()` e `longjmp()`.
- `errno.h`: Usado para lidar com erros. A variável `errno` é definida aqui.
- `assert.h`: Contém a macro `assert()`, que é usada para manipular afirmações e ajudar na depuração.

Bibliotecas em C – Part 4

- `float.h`: Define limites e propriedades de tipos de ponto flutuante.
- `limits.h`: Define as propriedades dos diferentes tipos de variáveis, como `INT_MAX`, `CHAR_MAX`, etc.
- `stddef.h`: Define vários tipos e macros úteis, como `ptrdiff_t` e `size_t`.

IDEs para Programação em C

- Ao escolher uma IDE, é essencial considerar suas necessidades individuais, como a complexidade do projeto, os recursos desejados e a familiaridade com a interface da IDE.
- Enquanto IDEs instaladas tendem a oferecer mais recursos e flexibilidade, as IDEs online são ótimas para aprendizado rápido, colaboração ou desenvolvimento em máquinas onde não é possível ou desejado instalar software adicional.

IDEs Instaladas

– Parte 1

- **Code::Blocks**
- **Plataformas suportadas:** Windows, Linux, macOS
- **Descrição:** Uma IDE C e C++ de código aberto com um compilador integrado e uma interface amigável.

- **Eclipse CDT (C/C++ Development Tooling)**
- **Plataformas suportadas:** Windows, Linux, macOS
- **Descrição:** Extensão do Eclipse, uma IDE popular, para desenvolvimento em C e C++. Oferece suporte robusto para grandes projetos.

IDEs Instaladas

– Parte 2

- **Microsoft Visual Studio**
- **Plataformas suportadas:** Windows, Linux
- **Descrição:** Uma das IDEs mais populares, suporta múltiplas linguagens, incluindo C e C++. É mais pesada, mas oferece uma ampla gama de recursos.
- **CLion**
- **Plataformas suportadas:** Windows, Linux, macOS
- **Descrição:** Desenvolvida pela JetBrains, é uma IDE moderna que oferece muitos recursos úteis, como análise de código em tempo real e integração com ferramentas de controle de versão.

IDEs Instaladas

– Parte 3

- **NetBeans**
- **Plataformas suportadas:** Windows, Linux, macOS
- **Descrição:** Originalmente conhecida por seu suporte a Java, NetBeans também suporta desenvolvimento em C e C++.
- **Dev C++**
- **Plataformas suportadas:** Windows
- **Descrição:** Uma IDE C/C++ mais antiga, mas ainda amplamente usada, especialmente em ambientes educacionais.

IDEs Instaladas

– Parte 4

- **Geany**
- **Plataformas suportadas:** Windows, Linux, macOS
- **Descrição:** Uma IDE leve com recursos básicos de programação em C.

IDEs Online – Parte 1

- **OnlineGDB**
- **Plataformas suportadas:** Web (acessível de Windows, Linux, macOS)
- **Descrição:** Compilador e depurador online para C. Permite edição, compilação e execução de programas C diretamente no navegador.

- **JDoodle**
- **Plataformas suportadas:** Web (acessível de Windows, Linux, macOS)
- **Descrição:** Uma IDE online que suporta várias linguagens, incluindo C. Possui colaboração em tempo real e versões de compiladores múltiplos.

IDEs Online – Parte 2

- **Compiler Explorer**
 - **Plataformas suportadas:** Web (acessível de Windows, Linux, macOS)
 - **Descrição:** Focado em C e C++, permite que os desenvolvedores vejam a saída do compilador em assembly, útil para otimizações de nível baixo.
-
- **Repl.it**
 - **Plataformas suportadas:** Web (acessível de Windows, Linux, macOS)
 - **Descrição:** Plataforma de codificação online que suporta várias linguagens, incluindo C. É muito interativo e oferece colaboração em tempo real.