



Unidade 4

Estrutura de Dados Heterogênea

DEC0012 - Linguagem de Programação I

Curso de Engenharia de Computação

Profa. Andréa Sabedra Bordin

Prof. Anderson Luiz Peres

Introdução

- **Uma estrutura de dados heterogênea** é um **tipo de dado definido pelo usuário** que permite **agregar várias variáveis de tipos de dados diferentes**.
 - Também é chamada de **variável composta heterogênea** ou **registro**.
- Uma estrutura de dados heterogênea é usada para armazenar um conjunto de informações sobre uma **entidade** qualquer.
- Por exemplo: Livro
 - Quais os atributos ou características de um livro ?
 - Título
 - Autor
 - ISBN
 -

Formato geral	Formato geral
<pre>struct Nome { member definition; member definition; ... member definition; }</pre>	<pre>struct Livro { char titulo[50]; char autor[50]; int isbn; }</pre>

- Cada unidade de informação (atributo ou característica) contida em uma estrutura é chamada de **campo**.
- Os campos podem ser de **diferentes tipos primitivos**, ou ainda, podem **representar outras estruturas**.

Declaração de uma variável do tipo estrutura

- A declaração de variáveis desse tipo pode ser feita ao longo do programa usando a palavra-chave **struct** junto com o **nome da estrutura** previamente criada pelo usuário.

- struct Livro livro1;
- struct Livro livro2;

livro1	Titulo: Autor: ISBN:
livro2	Titulo: Autor: ISBN:

- Nesse exemplo são criadas (declaradas) duas variáveis (**livro1** e **livro2**) do tipo **Livro**.

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

struct Livro{
    int isbn;
    char titulo[50];
    char autor[50];
};

int main(){
    struct Livro livro1;

    livro1.isbn=123;
    strcpy(livro1.titulo,"Cem anos de solidão");
    strcpy(livro1.autor,"Gabriel Garcia Marquez");

    printf("----- Dados do Livro-----\n");
    printf("ISBN: %d \n", livro1.isbn);
    printf("Titulo: %s \n", livro1.titulo);
    printf("Autor: %s \n", livro1.autor);
    return 0;
}
```

Acesso aos campos de uma
variável do tipo estrutura

Criação de um vetor de estruturas

- Em C é possível criar um vetor de estruturas.
- Neste caso cada elemento do vetor será uma estrutura.
- O exemplo a seguir cria um vetor chamado **bib**, que contém 5 elementos de **estrutura Livro (do tipo Livro)**.

- `struct Livro bib[5];`
- `bib[0].isbn = 100;`

bib	[0]	isbn: 100 titulo: autor:
	[1]	isbn: titulo: autor:
	[2]	isbn: titulo: autor:
	[3]	isbn: titulo: autor:
	[4]	isbn: titulo: autor:

Estrutura como argumento para uma função

- Estruturas podem ser passadas como argumentos para funções.
- Veja o exemplo a seguir:

```

#include <stdio.h>
#include <string.h>

struct Livro{
    int isbn;
    char titulo[50];
    char autor[50];
};

void imprime(struct Livro livro[3]);

int main(){

    struct Livro livro[3];

    livro[0].isbn=123;
    strcpy(livro[0].titulo,"Cem anos de solidão");
    strcpy(livro[0].autor,"Gabriel Garcia Marquez");
    livro[1].isbn=124;
    strcpy(livro[1].titulo,"O amor dos tempos do cólera");
    strcpy(livro[1].autor,"Gabriel Garcia Marquez");
    livro[2].isbn=125;
    strcpy(livro[2].titulo,"Memórias");
    strcpy(livro[2].autor,"Gabriel Garcia Marquez");

    imprime(livro);

    return 0;
}

void imprime(struct Livro livro[3]){
    for (int i=0;i<3;i++){
        printf("----- Dados do Livro %d -----\\n",i+1);
        printf("ISBN: %d \\n", livro[i].isbn);
        printf("Titulo: %s \\n", livro[i].titulo);
        printf("Autor: %s \\n", livro[i].autor);
    }
}

```

Chama a função imprime e passa o vetor do tipo Livro como argumento

A função imprime recebe como parâmetro o vetor do tipo Livro

Estrutura como retorno de uma função

- Uma estrutura pode ser retornada de uma função.
- Veja o exemplo a seguir:

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

```
struct Livro{
    int isbn;
    char titulo[50];
    char autor[50];
};
```

```
struct Livro setLivro(int isbn, char titulo[50], char autor[50]);
```

```
int main(){
    struct Livro livro;
    livro = setLivro(123,"Cem anos de solidão","Gabriel Garcia Marquez");
    printf("----- Dados do Livro ----\n");
    printf("ISBN: %d \n", livro.isbn);
    printf("Titulo: %s \n", livro.titulo);
    printf("Autor: %s \n", livro.autor);
    return 0;
}
```

```
struct Livro setLivro(int isbn, char titulo[50], char autor[50]){
    struct Livro livro;
    livro.isbn=isbn;
    strcpy(livro.titulo,"Cem anos de solidão");
    strcpy(livro.autor,"Gabriel Garcia Marquez");
    return livro;
}
```

Especifica que o tipo de retorno da função é a estrutura Livro

Chama a função e atribui seu retorno à variável do tipo Livro

Recebe como parâmetro os dados do livro, armazena na variável do tipo Livro e retorna essa variável