



CEFET – RJ / *Campus* Maria da Graça
Centro Federal de Educação Tecnológica
Celso Suckow da Fonseca – Rio de Janeiro



Prof. Cristiano Fuschilo

cristiano.fuschilo@cefet-rj.br

Linguagem e Técnicas de Programação



Bacharelado em
Sistemas de
Informação



COORDENAÇÃO DE
**Automação
Industrial**
Ensino de Qualidade



10ª Aula

Estruturas de Dados



Estrutura de Dados



- Muitas vezes precisamos compor os dados para formar estruturas de dados complexas
- Variáveis compostas homogêneas (Arrays)
 - Conjunto de variáveis de mesmo tipo
- Variáveis compostas heterogêneas
 - Conjunto de variáveis de tipos diferentes
- Chamadas de:
 - Estruturas (Struct)





Aplicação de Estruturas (1)

- Estruturas podem ser usadas para armazenar informações relacionadas
- Exemplo 1: Produto

Livro (char[11])	L	i	n	g	u	a	g	e	m		C
Preço (float)	59,9000										
Autor (char[11])	D	.		R	i	t	c	h	i	e	



Exemplo 2: Ficha de Cliente (Cadastro)



Nome (char[10])	H	e	l	e	n	a				
Idade (int)	30									
Telefone (int)	5555-5555									
Cidade (char[10])	S	a	o		P	a	u	l	o	



Definição de uma estrutura (registro) em C



```
struct identificacao_da_estrutura {  
    tipo1 nome1;  
    tipo2 nome2; ...  
    tipoN nomeN;  
};
```

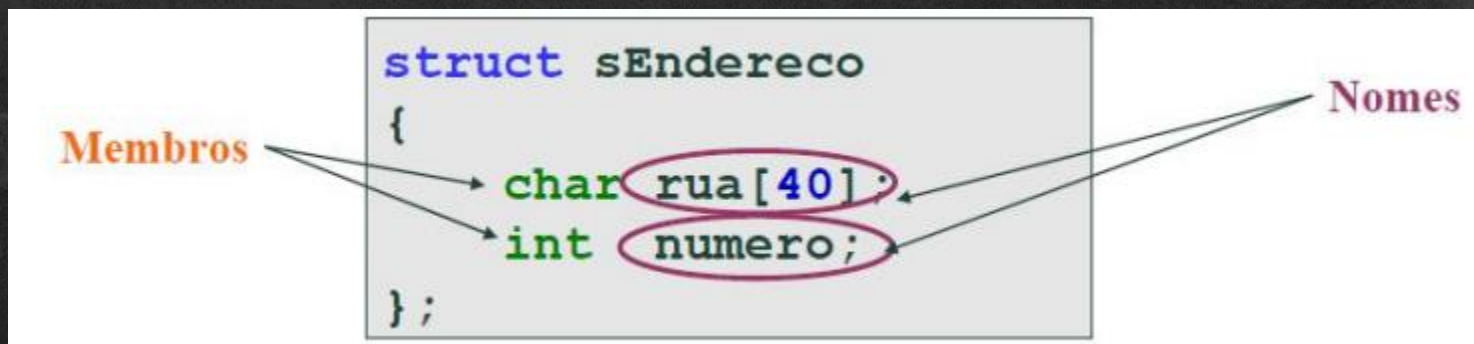
- Uma estrutura é um tipo de dado cujo formato é definido pelo programador
- Definição de uma estrutura (registro) em C



Estruturas



- Variáveis compostas heterogêneas (estruturas) são um conjunto de variáveis de tipos diferentes que são logicamente relacionadas.
- Essas variáveis compartilham o mesmo identificador e ocupam posições consecutivas de memória.
- Para as variáveis de uma estrutura:
 - Elas são denominadas membros;
 - São identificadas por nomes.





Exemplo - Declaração

- Vamos criar uma estrutura de endereço, que possa ser usada como se fosse um tipo de dado posteriormente
- Este código deve vir no início do programa, após os "includes"

```
1  #include <stdio.h>
2
3  struct sEndereco{
4      char rua[40];
5      int numero;
6      char cidade[30];
7      char estado[2];
8      int CEP;
9  };
```



Declaração de uma variável do tipo identificacao_da_estrutura



struct identificacao_da_estrutura nome_da_variavel;

```
struct identificacao_da_estrutura nome_da_variavel;
```





Exemplo - Programa

- Vamos criar uma programa que use a estrutura sEndereco e atribua valores a todas as variáveis da estrutura

```
1  #include <stdio.h>
2  #include <string.h>
3
4  struct sEndereco{
5      char rua[40];
6      int numero;
7      char cidade[30];
8      char estado[2];
9      int CEP;
10 };
11
12 int main(){
13     //cria variavel ender1 como struct sEndereco
14     struct sEndereco ender1;
15
16     strcpy (ender1.rua, "Rua 7 de Setembro");
17     ender1.numero = 405;
18     strcpy (ender1.cidade, "Goiana");
19     strcpy (ender1.estado, "MG");
20     ender1.cep = 36152000;
21 }
```



Declarando, atribuindo, imprimindo



```
1  #include <stdio.h>
2
3  struct sRetangulo{
4      float altura, base, area;
5  };
6
7  int main(){
8      struct sRetangulo ret;
9
10     printf("Digite a altura do retangulo: ");
11     scanf("%f", &ret.altura);
12     printf("Digite a base do retangulo: ");
13     scanf("%f", &ret.base);
14     ret.area = (ret.altura * ret.base)/2;
15     printf("A area do retangulo: %f", ret.area);
16 }
```





Estruturas Rotuladas

- Estruturas rotuladas criam um “rótulo” que pode ser referenciado posteriormente no código.
- Criação de rótulos.

```
struct rotulo_da_estrutura  
{  
    tipo1 nome1;  
    tipo2 nome2;  
    ...  
    tipoN nomeN;  
};
```



Estruturas Rotuladas



```
1  #include <stdio.h>
2
3  struct sHora{
4      int hora, minuto, segundo;
5  };
6
7  int main(){
8      struct sHora H;
9
10     H.hora = 10;
11     H.minuto = 30;
12     H.segundo = 10;
13     printf("%d:%d:%d ", H.hora, H.minuto, H.segundo);
14 }
```



Exercícios



1. Crie uma estrutura Livro com os seguintes campos:
 - ISBN
 - Título
 - Autor
 - Número de Páginas
 - Preço
 - Ano de publicação
- a) Defina uma variável do tipo da Estrutura Livro
- b) Atribua valores para cada um dos campos da estrutura Livro
- c) Imprima os valores dos campos



Exercício 2



2. Escreva um programa que possua uma variável capaz de armazenar o nome, a idade, o sexo e o peso de uma pessoa. Teste a variável atribuindo e lendo os valores dela.
3. Defina um tipo de estrutura rotulada para representar números complexos da forma $a + b.i$, sendo a a parte real e b a imaginária. Crie também uma função para calcular a soma de dois números complexos, codificando também um programa para testar o seu funcionamento





Estruturas Aninhadas

- Estruturas em que um ou mais de seus membros também sejam estruturas.

```
1  #include <stdio.h>
2
3  struct id_estrutural1{
4      tipo1 nome1;
5      tipoN nomeN;
6  };
7
8  struct id_estrutural2{
9      struct id_estrutural1 nome;
10     tipoN nomeN;
11 };

```



Exemplo



```
1  #include <stdio.h>
2
3  struct Hora{
4      int hor;
5      int min;
6      int seg;
7  };
8
9  struct Relogio{
10     struct Hora H;
11     char  modelo[10];
12 };
13
```



Estruturas Aninhadas



```
1  #include <stdio.h>
2  #include <string.h>
3
4  struct Hora{
5      int hor;
6      int min;
7      int seg;
8  };
9
10 struct Relogio{
11     struct Hora H;
12     char  modelo[10];
13 };
14
15 int main(){
16     struct Relogio r1;
17     r1.H.hor = 10;
18     r1.H.min = 15;
19     r1.H.seg = 30;
20     strcpy(r1.modelo, "Casio");
21
22     printf ("Modelo:\t%s\n", r1.modelo);
23     printf ("Hora----\n\t%d\n\t%d\n\t%d", r1.H.hor, r1.H.min, r1.H.seg);
24 }
```



Exercício



1. Defina um tipo de estrutura para armazenar os dados de um voo:
 1. nomes das cidades de origem e destino,
 2. datas e horários de partida e chegada.
2. Utilize a estrutura hora do exemplo anterior.
3. Crie um programa para testar as funcionalidades criadas, declarando variáveis e funções necessárias



Vetores e Estruturas



- É possível combinar vetores e estruturas para criação de diferentes estruturas de dados.
- Podemos ter uma estrutura contendo um membro do tipo vetor, ou;
- Criar um vetor cujo os elementos sejam estruturas





Declarando vetor de Estruturas

- Dada a estrutura listada abaixo:

```
struct lista {  
    char titulo[30];  
    char autor[30];  
    int regnum;  
    double preco;  
};
```

Membros do
tipo vetor



Declare um vetor com 50 elementos do tipo lista





Declarando vetores de Estruturas

```
struct lista livro[50];
```

- livro é um vetor de 50 elementos.
- Cada elemento do vetor é uma estrutura do tipo struct lista
- O que significa livro[0], livro[1], livro[2], etc?
** Por meio dessa instrução o compilador providencia espaço de memória para 50 estruturas do tipo struct lista.



Exemplo



```
...
struct sEndereco
{
    char rua[40];
    int  numero;
};

int main(void) {

    struct sEndereco listaend[5];

    listaend[0].numero = 100;

    strcpy(listaend[3].rua, "Av. Brasil");
    ...
}
```



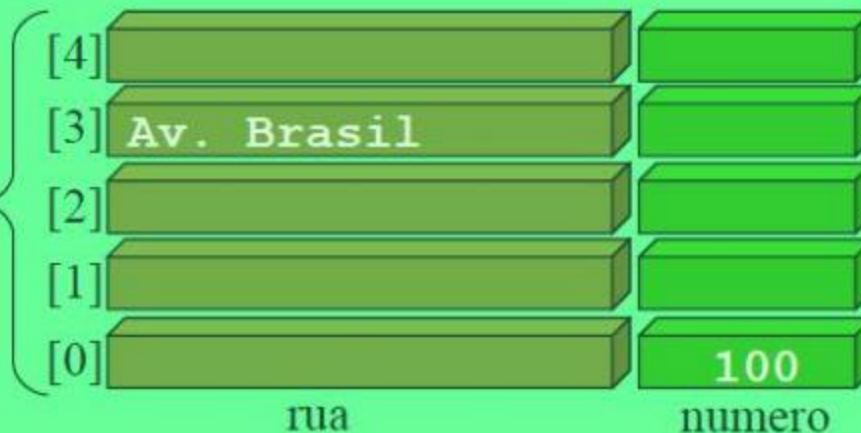
Trecho de exemplo



```
...
struct sEndereco
{
    char rua[40];
    int  numero;
};
int main(void) {
    struct sEndereco listaend[5];
    listaend[0].numero = 100;
    strcpy(listaend[3].rua, "Av. Brasil");
    ...
}
```

ID da
estrutura

listaend:



Na memória



Exemplo



```
#include <stdio.h>

struct sHora {
    int hor;
    int min;
    int seg;
};

int main(void) {

    struct sHora H[5];
    H[0].hor = 10;
    H[0].min = 15;
    H[0].seg = 30;
    printf("%d:%d:%d", H[0].hor, H[0].min, H[0].seg);
    return 0;
}
```



Exercício



1. Crie um programa que permita armazenar o nome, a altura e a data de nascimento de 10 pessoas.
 - Cada pessoa deve ser representada por uma struct.
 - A data de nascimento também deve ser uma struct.
 - O nome, altura e data de nascimento de cada pessoa devem ser informados pelo teclado.
 - Imprimir o nome da pessoa com a maior altura.



