

Algoritmos e Práticas de Programação

Prof. Alessandro Santos
Estruturas

Estruturas

- O que são estruturas (struct) na linguagem C?

Estruturas (Struct)

- Uma estrutura pode ser vista **como um conjunto de variáveis sob um mesmo nome**, sendo que cada uma delas pode **ter qualquer tipo** (ou o mesmo tipo). A proposta básica por trás da estrutura é criar apenas um tipo de dado que contenha vários membros, que nada mais são do que outras variáveis. **Em outras palavras, é criada uma variável que contém dentro de si outras variáveis.**

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Estruturas (Struct)

Sintaxe para a declaração de uma *struct*.

- struct nome
- {
- tipo1 campo1;
- tipo2 campo2;
- ...
- tipon campoN;
- };

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Estruturas (Struct)

Por que utilizar Estruturas?

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Estruturas (Struct) - Por que utilizar Estruturas?

A principal vantagem do uso de estruturas é agrupar de forma organizada vários tipos de dados diferentes dentro de uma única variável.

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Estruturas (Struct) - Exemplo de estrutura

```
struct cadastro
{
char nome [50] ;
int idade;
char rua [50];
int numero;
};
```



Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Estruturas (Struct) - Outro Exemplo de estrutura

```
struct cadastro
{
char nome[50];
int idade;
char rua[50];
int numero;
};
```

```
struct aluno
{
char nome[50];
int matricula;
float nota1,nota2,nota3;
};
```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Declarando uma Variável do tipo da Estrutura

► **struct** cadastro c;

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Sem utilizar estruturas...

► Imagine ter que declarar 4 cadastros para 4 pessoas diferentes?

```
char nome1[50], nome2[50], nome3[50], nome4[50];  
int idade1, idade2, idade3, idade4;  
char rua1[50], rua2[50], rua3[50], rua4[50];  
int numero1, numero2, numero3, numero4;
```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Utilizando estruturas...

► **struct** cadastro c1, c2, c3, c4;

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

► **Como acessar uma variável numa estrutura?**

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Praticando...

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <conio.h>
5
6  struct cadastro
7  {
8      char nome[50];
9      int idade;
10     char rua[50];
11     int numero;
12 };
13
14 main ()
15 {
16     struct cadastro c;
17
18     strcpy (c.nome, "Gustavo"); //atribuir um dado string para o campo
19
20     c.idade = 20;
21
22     strcpy (c.rua, "Avenida Santo Amaro");
23
24     c.numero = 1570;
25
26     printf("Nome: %s | Idade: %d | Rua: %s , n: %d\n\n", c.nome, c.idade, c.rua, c.numero );
27
28     getch();
29 }

```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Não veio nesta aula?

Procure compreender o Código desenvolvido e qualquer dúvida comente com o seu professor!

- O que fazer se fosse necessário ler os valores dos campos da estrutura do teclado?

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Praticando...

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <conio.h>
4
5  struct cadastro
6  {
7      char nome[50];
8      int idade;
9      char rua[50];
10     int numero;
11 };
12
13 main()
14 {
15     struct cadastro c;
16
17     printf("Digite o seu nome: \n");
18     gets(c.nome); // ler uma string
19     printf("Informe a sua idade: \n");
20     scanf("%d",&c.idade);
21     fflush(stdin); //limpar o buffer (memória)
22     printf("Digite o nome da sua rua: \n");
23     //scanf("%s", c.rua); // outra possibilidade de entrada de dados
24     gets(c.rua);
25     printf("Digite o numero da rua que fica sua casa\n");
26     scanf("%d",&c.numero);
27
28     printf("Nome: %s | Idade: %d | Rua: %s , n: %d\n\n", c.nome, c.idade, c.rua, c.numero );
29
30     getch();
31 }
32

```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Não veio nesta aula?

Procure compreender o Código desenvolvido e qualquer dúvida comente com o seu professor!

Inicialização de Estruturas

► Como inicializar uma estrutura?

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Inicialização de Estruturas

- ▶ **struct** cadastro c = {"Carlos", 18, "Avenida Brasil", 1082 };

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Inicialização de Estruturas

- ▶ **struct** cadastro c = {"Carlos", 18, "Avenida Brasil", 1082 };
- ▶ E se um dado não for atribuído o que acontece??
- ▶ **struct** cadastro c = {"Carlos", 18 };
- ▶ No exemplo acima, o campo *rua* é inicializado com "" e *numero* com zero.

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Array de Estruturas

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Array de Estruturas - Relembrando...

- Sem utilizar estruturas:
- 4 cadastros para 4 pessoas diferentes:

```
char nome1[50], nome2[50], nome3[50], nome4[50];  
int idade1, idade2, idade3, idade4;  
char rua1[50], rua2[50], rua3[50], rua4[50];  
int numero1, numero2, numero3, numero4;
```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Array de Estruturas - Relembrando...

- ▶ Utilizando estruturas:
- ▶ `struct cadastro c1, c2, c3, c4;`

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Array de Estruturas

- ▶ E utilizando *arrays*:
- ▶ `struct cadastro c[4];`

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

► Como ler os valores dos campos de um *array* de estrutura do teclado?

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Praticando...

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <conio.h>
4
5  struct cadastro
6  {
7      char nome[50];
8      int idade;
9      char rua[50];
10     int numero;
11 };
12
13 main()
14 {
15     struct cadastro c[4];
16     int i; //índice do vetor
17
18     for(i=0; i<4;i++)//atribuição de dados para cada índice do vetor de estrutura
19     {
20         printf("Cadastro %d\n",i);
21         printf("Digite o seu nome: \n", i);
22         gets(c[i].nome);
23         fflush(stdin);
24         printf("Informe a sua idade: \n");
25         scanf("%d", &c[i].idade);
26         fflush(stdin); //limpar o buffer (memória)
27         printf("Digite o nome da rua\n");
28         gets(c[i].rua);
29         fflush(stdin);
30         printf("Digite o numero da rua que fica sua casa\n");
31         scanf("%d",&c[i].numero);
32         fflush(stdin);
33     }
34     getch();

```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Não veio nesta aula?

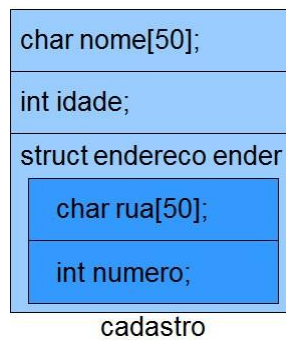
Procure compreender o Código desenvolvido e qualquer dúvida comente com o seu professor!

Estruturas Aninhadas

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Estruturas Aninhadas - Exemplo: *struct* aninhada.

```
struct endereco
{
char rua [50]
int numero;
};
struct cadastro
{
char nome [50] ;
int idade;
struct endereco ender ;
};
```



Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

► Como acessar um campo que está dentro de outra estrutura?

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Praticando...

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <conio.h>
4
5  struct endereco
6  {
7      char rua[50];
8      int numero;
9  };
10
11 struct cadastro
12 {
13     char nome[50];
14     int idade;
15     struct endereco ender;
16 };
17
18 main()
19 {
20     struct cadastro c;
21
22     printf("Digite o seu nome: \n");
23     gets(c.nome); // Ler uma string
24     printf("Informe a sua idade: \n");
25     scanf("%d",&c.idade);
26     fflush(stdin); //limpar o buffer (memória)
27     printf("Digite o nome da sua rua: \n");
28     //scanf("%s", c.rua); // outra possibilidade de entrada de dados
29     gets(c.ender.rua); // acessando um campo que está dentro de outra estrutura
30     printf("Digite o numero da rua que fica sua casa\n");
31     scanf("%d",&c.ender.numero);
32
33     printf("Nome: %s | Idade: %d | Rua: %s , n: %d\n\n", c.nome, c.idade, c.ender.rua, c.ender.numero );
34
35     getch();
36 }

```

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Não veio nesta aula?

Procure compreender o Código desenvolvido e qualquer dúvida comente com o seu professor!

Referências Bibliográficas

FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico. *Lógica de Programação*. Ed. Pearson Brasil, 2000.

MANZANO, José Augusto Navarro Garcia; OLIVEIRA, Jair Figueiredo de. *Algoritmos: Lógica para o desenvolvimento de programação*. São Paulo: Érica, 2004.

MIZRAHI, Victorine Viviane. *Treinamento em linguagem C*. São Paulo: McGraw-Hill, 1990.

PEREIRA, Silvio do Lago. *Algoritmos e Lógica de Programação em C: Uma abordagem Didática*. São Paulo: Érica, 2010.

SANTOS, R. A. R. *Notas de Aula: Arquivo pessoal*. São Paulo: [s.e.], 2018.

SCHILDT, Hebert. *C - Completo e total*. São Paulo: Makron Books, 1995.

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Complemente o seu conhecimento

► Acesse o AVA e complemente o seu estudo com as aulas:

“Conceitos iniciais sobre Registros (Estruturas) em C”,

“Inicialização de estruturas e estruturas aninhadas”,

“Conceitos iniciais sobre Tabelas em C”, e

“Passagem de estruturas e tabelas como parâmetros para funções”.

► Exercite o seu conhecimento no site:

“<https://studio.code.org/s/20-hour>”

► Extra - Lista de Exercícios:

“<https://goo.gl/9hCXMf>”

Prof. Alessandro Santos - Algoritmos e Práticas de Programação - UNINOVE - 2018

Desafio:

Faça o exercício a seguir na linguagem de programação C. Lembre-se que para fazer o exercício a abstração é de suma importância!

1- Crie uma agenda telefônica com 15 registros. Utilize as imagens a seguir para referência do que deve ser programado.

Cadastrar:

```
Menu de Opcoes
1 - Cadastrar
2 - Visualizar
3 - Alterar
4 - Sair
Digite sua opcao: 1
```

Visualizar: OBS: Mostra todos os espaços da agenda.

```
Menu de Opcoes
1 - Cadastrar
2 - Visualizar
3 - Alterar
4 - Sair
Digite sua opcao: 2
```

```
Menu de Opcoes
1 - Cadastrar
2 - Visualizar
3 - Alterar
4 - Sair
Digite sua opcao: 2
cod: 0
Nome: Alessandro
Telefone: 12345678
-----
cod: 1
Nome:
Telefone: 0
-----
```

Alterar:

```
Menu de Opcoes
1 - Cadastrar
2 - Visualizar
3 - Alterar
4 - Sair
Digite sua opcao: 3
Informe codigo que deseja alterar:0
Nome: Alessandro
Informe NOVO nome: Alejandro
Telefone: 12345678
Informe NOVO telefone: 987654321
```

Visualizar: o que foi alterado

```
Menu de Opcoes
1 - Cadastrar
2 - Visualizar
3 - Alterar
4 - Sair
Digite sua opcao: 1
Informe nome: Alessandro
Informe telefone: 12345678_
```

```
Menu de Opcoes
1 - Cadastrar
2 - Visualizar
3 - Alterar
4 - Sair
Digite sua opcao: 2
cod: 0
Nome: Alejandro
Telefone: 987654321
-----
```

```
Menu de Opcoes
1 - Cadastrar
2 - Visualizar
3 - Alterar
4 - Sair
Digite sua opcao: 2
cod: 2
Nome:
Telefone: 0
-----
cod: 3
Nome:
Telefone: 0
-----
cod: 4
Nome:
Telefone: 0
-----
```

Sair:

```
Menu de Opcoes
1 - Cadastrar
2 - Visualizar
3 - Alterar
4 - Sair
Digite sua opcao: 4_
```