



INSTITUTO DE
INFORMÁTICA
UFG

Vetores

Thierson
Couto Rosa

Estruturas - Structs

Thierson Couto Rosa

Instituto de Informática
Universidade Federal de Goiás
Introdução à Programação

17 de novembro de 2020



O tipo de Dado Estrutura

Definição

Uma *estrutura* é um tipo de dados que permite agrupar um conjunto de variáveis sobre um mesmo nome. As variáveis agrupadas podem ser de tipos distintos entre si.

Diferença entre Vetores e Estruturas

- Um vetor também é um agrupamento de variáveis porém, agrupa apenas variáveis de mesmo tipo. Por isso um vetor é uma *variável composta homogênea* (homogênea - mesmo tipo).
- Uma estrutura pode agrupar variáveis de tipos distintos e por isso também é denominada *variável composta heterogênea*.



Declaração de um Tipo Estrutura em C

Formato Geral:

```
1 struct nome_struct{  
2     tipo1 var1, var2,..., varm;  
3     ...  
4     tipon vark, varj;  
5 }
```

Observações

- nome_struct não é o nome de uma variável, e sim, o nome de um tipo novo criado pelo programador.
- O programador pode declarar várias variáveis do tipo nome_struct



Exemplo de Declaração do tipo Estrutura

Declaração do tipo Data

- Suponha que se queira declarar um tipo estrutura *Data* para que se possa criar variáveis compostas formadas por três números inteiros correspondendo, respectivamente a dia, mês e ano.
- A seguinte declaração poderia ser feita na linguagem C para declarar esse tipo de dados:

```
1 struct data{  
2     int dia;  
3     int mes;  
4     int ano;  
5 }
```



Declaração de Variáveis do Tipo Estrutura

Declaração do Tipo em Primeiro Lugar

```
1 //Declaração do tipo struct data
2 struct data{
3     int dia;
4     int mes;
5     int ano;
6 }
7 //Declaração das variáveis do tipo struct data
8 struct data nascimento, morte, vetData[10], matData[3][5];
```

Declaração Conjunta do Tipo e das Variáveis

```
1 //Declaração conjunta do tipo e das variáveis
2 struct data{
3     int dia;
4     int mes;
5     int ano;
6 } nascimento, morte, vetData[10], matData[3][5];
```



Acesso aos Campos de uma Variável do Tipo Struct

Exemplo

```
1 struct data{
2     int dia;
3     int mes;
4     int ano;
5 }
6 struct data nascimento, morte, vetData[10], matData[3][5];
7
8 // ponteiro para uma variavel do tipo struct data
9 struct data * d;
10
11 //acesso ao campo pelo nome da
12 // variavel:
13 nascimento.dia = 10; // Um ponto entre o nome da variavel e o nome do campo.
14
15 //acesso ao campo por um ponteiro:
16 d = &nascimento;
17 (*d).mes = 4; //(*d) e a variavel struct e .mes acessa o campo mes
18 // OU...
19 d->mes = 4; // a seta e formada por - seguido de um >
```



Passagem de Variáveis Struct como Parâmetros de Função

Passagem de Structs

- Quando Uma variável do tipo Struct é passada como parâmetro para uma função, é feita uma cópia de todos os campos da variável no parâmetro correspondente do mesmo tipo.
- Isso implica que se a função alterar um valor de um campo do parâmetro, essa alteração não ocorre na variável passada durante a chamada.
- Lembre-se: *Em C as passagens de parâmetros são feitas por valor.*
- Para que a função altere o campo da variável passada como parâmetro, o parâmetro correspondente deve ser do tipo ponteiro para o tipo struct utilizado e a chamada deve passar o endereço da variável.



Forma Simplificada de dar Nome a Structs

typedef

- Para não utilizar um nome composto para declarar variáveis do tipo struct, isto é, `struct nome a,b c;`, há na linguagem C um forma de atribuir um apelido ao nome de um tipo.
- Isso é possível com o comando: `typedef struct x {...} novoNome .`
- Uma vez utilizado o `typedef` para definir um "apelido" para o tipo de dados, as variáveis podem ser declaradas tanto como sendo do tipo `novoNome` ou do tipo `struct x` .
- Exemplo: as duas declarações são válidas: `novoNome a,b;` `struct x c,d.` As variáveis `a`, `b`, `c`, `d` são armazenadas e tratadas internamente como sendo do mesmo tipo (`struct x`) .



Exemplo

Escreva um programa para ler o nome (no máximo 50 caracteres) a matrícula (int) e as três notas de cada aluno de uma turma de 10 alunos. O programa deve computar a média das três notas de cada aluno e ordenar os alunos em ordem decrescente de média e imprimir a lista dos alunos. Para cada aluno o programa deve imprimir uma linha com: o nome do aluno, sua matrícula e sua média das três notas.

Entrada

Para cada aluno programa deve ler duas linhas. A primeira, contém o nome do aluno terminado com um caractere de quebra de linha. A segunda linha contém um número inteiro correspondente à matrícula, um espaço e três números double separados entre si por um espaço, que correspondem às três notas do aluno.