



Introdução à Programação de Computadores
GSI002 - 2012/2
Profs. Walter de Oliveira / Renan Cattelan

Prática 7

Structs

Structs

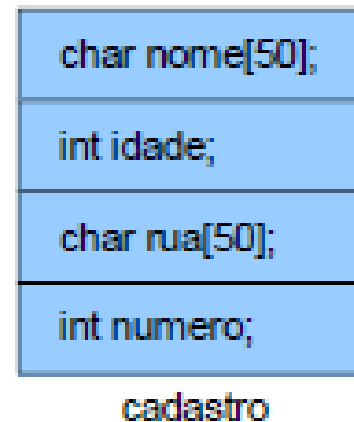
- ❑ Estrutura de dados criada a partir de tipos básicos
- ❑ Coleção de variáveis

```
struct nomestruct{  
    tipo1 campo1;  
    tipo2 campo2;  
    ...  
    tipoN campoN;  
};
```

Structs

□ Definição

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



□ Declaração

```
struct cadastro c;
```

Structs

□ Acesso

- Cada variável da estrutura pode ser individualmente acessada com o operador “.”

```
struct cadastro c;  
strcpy(c.nome, "João");  
scanf("%d", &c.idade);  
strcpy(c.rua, "Avenida 1");  
c.numero = 1082;
```

Structs

□ Inicialização prévia

```
struct ponto {  
    int x;  
    int y;  
};  
struct ponto p1 = { 220, 110 };
```

Lista 6

4. Considerando a estrutura

```
struct Vetor{  
float x;  
float y;  
float z;  
};
```

para representar um vetor no R^3 , implemente um programa que calcule a soma de dois vetores.

```
1  #include<stdio.h>
2
3  int main(void)
4  {
5      struct Vetor
6      {
7          float x;
8          float y;
9          float z;
10     };
11
12     struct Vetor v1, v2;
13
14     printf("Digite as coordenadas de v1: ");
15     scanf("%f %f %f", &v1.x, &v1.y, &v1.z);
16
17     printf("\nDigite as coordenadas de v2: ");
18     scanf("%f %f %f", &v2.x, &v2.y, &v2.z);
19
20     printf("\nv1 + v2 = \ (%.2f, %.2f, %.2f)\n", v1.x+v2.x, v1.y+v2.y, v1.z+v2.z);
21
22     system("pause");
23     return 0;
24 }
25
```

Lista 6

17. Faça um programa que gerencie o estoque de um mercado e:

- Crie e leia um vetor de 5 produtos, com os dados: código (inteiro), nome (máximo 15 letras), preço e quantidade.
- Leia um pedido, composto por um código de produto e a quantidade. Localize este código no vetor e, se houver quantidade suficiente para atender ao pedido integralmente, atualize o estoque e informe o usuário. Repita este processo até ler um código igual a zero.


```
1  #include<stdio.h>
2
3  int main(void)
4  {
5      struct Produto
6      {
7          int codigo;
8          char nome[15];
9          float preco;
10         int qtd;
11     };
12
13     struct Produto vetorDeProdutos[5];
14
15     printf("Preencha o estoque:\n");
16     int i;
17     for (i=0; i<5; i++) {
18         printf("\nDigite o codigo do %do. produto: ", i+1);
19         scanf("%d", &vetorDeProdutos[i].codigo);
20         printf("Digite o nome do %do. produto: ", i+1);
21         scanf("%s", &vetorDeProdutos[i].nome);
22         printf("Digite o preco do %do. produto: ", i+1);
23         scanf("%f", &vetorDeProdutos[i].preco);
24         printf("Digite a quantidade em estoque do %do. produto: ", i+1);
25         scanf("%d", &vetorDeProdutos[i].qtd);
26     }
27
```

```
27
28 int codPedido, qtdPedido;
29 do {
30     printf("\n\nAgora faça seu pedido (codigo e quantidade): ");
31     scanf("%d %d", &codPedido, &qtdPedido);
32     for (i=0; i<5; i++) {
33         if (vetorDeProdutos[i].codigo == codPedido) {
34             if (vetorDeProdutos[i].qtd >= qtdPedido && qtdPedido != 0) {
35                 printf("\nPedido aceito.");
36                 vetorDeProdutos[i].qtd = vetorDeProdutos[i].qtd - qtdPedido;
37                 printf("\nProduto agora com %d unidades em estoque.", vetorDeProdutos[i].qtd);
38             } else {
39                 printf("\nQuantidade insuficiente.");
40             }
41         }
42     }
43 } while (codPedido != 0);
44
45 system("pause");
46 return 0;
47 }
```