Programação Procedimental
GBC014 – 2015/1
Prof. Renan Cattelan – www.facom.ufu.br/~renan

# Prática 7

## Structs

# Structs

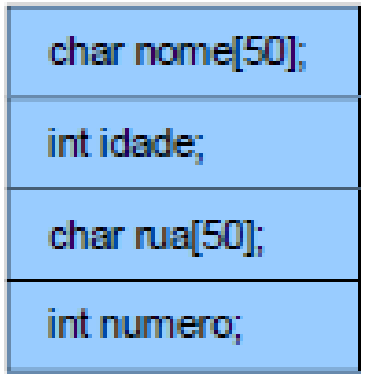
- Estrutura de dados criada a partir de tipos básicos
- Coleção de variáveis

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

# Structs

- Definição

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



cadastro

- 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 };
```

# Exercício 1

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.

```c
#include<stdio.h>

int main(void)
{
    struct Vetor
    {
        float x;
        float y;
        float z;
    };

    struct Vetor v1, v2;

    printf("Digite as coordenadas de v1: ");
    scanf("%f %f %f", &v1.x, &v1.y, &v1.z);

    printf("\nDigite as coordenadas de v2: ");
    scanf("%f %f %f", &v2.x, &v2.y, &v2.z);

    printf("\nv1 + v2 = \(%.2f, %.2f, %.2f\)\n", v1.x+v2.x, v1.y+v2.y, v1.z+v2.z);

    system("pause");
    return 0;
}

```

# Exercício 2

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.

```c
#include<stdio.h>

int main(void)
{
    struct Produto
    {
        int codigo;
        char nome[15];
        float preco;
        int qtd;
    };

    struct Produto vetorDeProdutos[5];

    printf("Preencha o estoque:\n");
    int i;
    for (i=0; i<5; i++) {
        printf("\nDigite o codigo do %do. produto: ", i+1);
        scanf("%d", &vetorDeProdutos[i].codigo);
        printf("Digite o nome do %do. produto: ", i+1);
        scanf("%s", &vetorDeProdutos[i].nome);
        printf("Digite o preco do %do. produto: ", i+1);
        scanf("%f", &vetorDeProdutos[i].preco);
        printf("Digite a quantidade em estoque do %do. produto: ", i+1);
        scanf("%d", &vetorDeProdutos[i].qtd);
    }

```

```
27
28      int codPedido, qtdPedido;
29      do {
30          printf("\n\nAgora faca 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  }
```

# Exercício 3

1) Escreva um programa para criar uma *struct* chamada Ponto, contendo as coordenadas inteiras x e y do plano. Declare 2 pontos, leia as coordenadas x e y de cada um e calcule e imprima a distância entre os pontos.

```c
#include<stdio.h>
#include<math.h>

int main(void)
{
    struct Ponto
    {
        float x;
        float y;
    };

    struct Ponto p1, p2;

    printf("Digite as coordenadas de p1: ");
    scanf("%f %f", &p1.x, &p1.y);

    printf("\nDigite as coordenadas de p2: ");
    scanf("%f %f", &p2.x, &p2.y);

    float distancia = sqrt(pow((p1.x-p2.x),2)+pow((p1.y-p2.y),2));

    printf("\nA distancia entre os pontos eh: %.2f.\n", distancia);

    system("pause");
    return 0;
}
```