

Algoritmos e Lógica de Programação

- Douglas Baptista de Godoy

Ementa

- Projeto e representação de algoritmos.
- Estruturas de controle de fluxo de execução: sequência, seleção e repetição.
- Tipos de dados básicos e estruturados (vetores e registros).
- Rotinas. Arquivos.
- Implementação de algoritmos usando uma linguagem de programação.

Objetivo

- Analisar problemas computacionais e projetar soluções por meio da construção de algoritmos.

ESTRUTURAS

- **Definições**
- Ao utilizar vetores e matrizes conseguíamos manipular uma grande quantidade de dados por meio de um acesso indexado, no entanto tínhamos a limitação de que todos os elementos deveriam ser do mesmo tipo. Em diversas situações nos deparamos com o fato de ter que armazenar informações relacionadas entre si, mas de tipos distintos.
- A estrutura (STRUCT) permite agregar diversas informações, que podem ser de diferentes tipos. Possibilita gerar novos tipos de dados, além dos definidos pelas linguagens de programação.

ESTRUTURAS

- **Definições**
- Em uma estrutura o acesso aos elementos não é realizado por meio de sua localização, mas sim por meio do nome do campo que se pretende acessar. Cada informação da estrutura é denominada de campo, os quais podem ser de diferentes tipos.
- A sintaxe para declaração de uma estrutura é:

```
struct nome_da_estrutura  
{  
    tipo_de_dado do campo 1;  
    tipo_de_dado do campo 2;  
    ...  
    tipo_de_dado do campo N;  
};
```

ESTRUTURAS

- **Definições**
- A partir da definição da estrutura o programa pode considerar que existe um novo tipo de dado a ser utilizado (nome_da_estrutura). Esse novo tipo de dado é capaz de armazenar informações que podem ser de tipos distintos.

```
struct nome_da_estrutura
{
    tipo_de_dado do campo 1;
    tipo_de_dado do campo 2;
    ...
    tipo_de_dado do campo N;
};
```

ESTRUTURAS

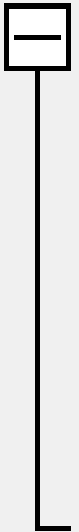
- **Exemplo**
- Consideremos como exemplo a definição de uma estrutura que armazena a ficha de um produto, conforme ilustrado a seguir. As informações a serem armazenadas são o código do produto, a descrição, o preço e o saldo em estoque, as quais representam os campos que a estrutura deve conter.

Código do Produto:
Descrição:
Preço:
Saldo:

ESTRUTURAS

- Exemplo
- A declaração da estrutura para a ficha do produto é:

```
2 struct produto
3 {
4     int codigo;
5     char descricao[50];
6     float preco;
7     int saldo;
8 };
```



ESTRUTURAS

- **Definições**
- Essa declaração indica que o programa poderá utilizar um novo tipo de dado que contém quatro informações.
- É importante destacar que a *struct* só pode ser utilizada dentro do bloco em que foi definida. Isto é, uma *struct* definida dentro das chaves que delimitam a função *main* só poderá ser utilizada por variáveis que também estejam neste bloco.
- Para que a *struct* seja acessível de qualquer parte do programa temos que defini-la fora da função *main*, preferencialmente fora dos *includes*.

ESTRUTURAS

- **Definições**
- Para utilizar uma *struct* temos que declarar uma variável deste tipo do seguinte modo:

nome_da_estrutura nome_da_variável;

- No caso da ficha de produto teríamos a seguinte declaração:

produto ficha;

- Essa declaração nos indica que temos uma variável denominada ficha que é do tipo produto. Nas operações de atribuição, leitura e escrita utilizamos o nome da variável *struct* e seu campo correspondente separado por um caractere “.” (ponto).

ESTRUTURAS

- **Definições**
- A seguir temos um programa que exemplifica as operações leitura e escrita utilizando *struct*. Observa-se que inicialmente temos a declaração da *struct* após a diretiva *include* e no interior da função *main* temos a declaração de uma variável do tipo produto. O acesso a cada um dos campos é realizado por meio do *nome da variável* mais o caractere ponto (".") e *nome do campo*.

ESTRUTURAS

- Exemplo:

```
1  #include <stdio.h>
2  struct produto
3  {
4      int codigo;
5      char descricao[50];
6      float preco;
7      int saldo;
8  };
9
10 int main()
11 {
```

ESTRUTURAS

- Exemplo:

```
10  int main()
11  {
12      produto ficha;
13      printf("\n Digite o codigo do produto: ");
14      scanf ("%d%c", &ficha.codigo);
15      printf("\n Digite a descricao do produto: ");
16      scanf ("%s%c", &ficha.descricao);
17      //gets(ficha.descricao);
18      printf("\n Digite o preco do produto: ");
19      scanf ("%f%c", &ficha.preco);
20      printf("\n Digite o saldo do produto: ");
21      scanf ("%d%c", &ficha.saldo);
22      printf("\n Codigo : %d", ficha.codigo);
23      printf("\n Descricao : %s", ficha.descricao);
24      printf("\n Preco : %.2f", ficha.preco);
25      printf("\n Saldo : %d", ficha.saldo);
26      getchar();
27      return 0;
28  }
```

ESTRUTURAS

- **Exercício:**
- O exemplo anterior ilustra a leitura e escrita da ficha de um produto.
- E se quisermos armazenar a ficha de 10 produtos, precisamos criar 10 variáveis do tipo produto?
- **Resposta correta:** Não! Podemos criar um vetor de *struct*.

ESTRUTURAS

- Exercício

```
1  #include <stdio.h>
2
3  struct produto
4  {
5      int codigo;
6      char descricao[50];
7      float preco;
8      int saldo;
9  };
10
```

ESTRUTURAS

- Exercício

```
11 main ()
12
13 {
14     produto ficha [10];
15     int i;
16     for (i=0; i<10; i++)
17     {
18         printf("\n Digite o codigo do produto:");
19         scanf ("%d", &ficha[i].codigo);
20         printf("\n Digite a descricao do produto: ");
21         scanf ("%s", &ficha[i].descricao);
22         printf("\n Digite o preco do produto: ");
23         scanf ("%f", &ficha[i].preco);
24         printf("\n Digite o saldo do produto: ");
25         scanf ("%d", &ficha[i].saldo);
26     }
27
```


ESTRUTURAS

- Exercício

```
27  
28  
29 ☐  
30  
31  
32  
33  
34  
35  
36  
37
```

```
for (i=0; i<10; i++)  
{  
    printf("\n  PRODUTO  %d", i+1);  
    printf(" Codigo : %d", ficha[i].codigo);  
    printf(" Descricao : %s", ficha[i].descricao);  
    printf("  Preco : %.2f", ficha[i].preco);  
    printf("  Saldo : %d", ficha[i].saldo);  
}  
return(0);  
}
```

Exercícios – STRUCT

- 1) Exemplo 01 do Slide, ou seja, a Struct de Produtos
- 2) Exemplo 02 do Slide, ou seja, a Struct de Produtos com Vetor.
- 3) Utilizando **struct** e **vetor** faça um programa que efetue o cadastro de 10 livros contendo as informações de código, título, autor, área e editora.
- 4) Utilizando **struct** e **matriz** faça o mesmo programa anterior, que cadastre 10 livros contendo informações de código, título, autor, área e editora

Referencias Bibliográficas

- ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de, **Fundamentos da Programação de Computadores**, Pearson Editora, 3ª edição.