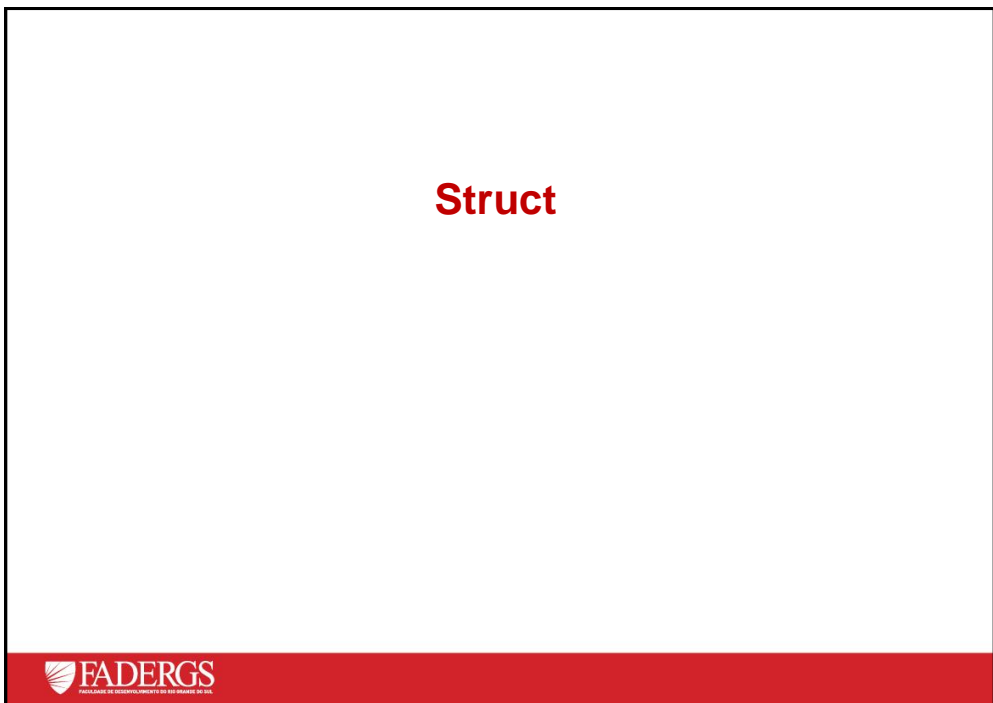




1



2

- **Vetores e Matrizes**

- Estruturas de dados homogêneas
- Armazenam vários valores, porém todos do mesmo tipo

- **Problemas:**

- Coleção de dados de tipos diferentes. Ex.: ficha de cadastro de um cliente.

Nome	char nome[100];
Endereço	char endereco[200];
Telefone	char telefone[9];
Salário	float salario;
Idade	int idade;



3

- **Structs**

- Struct, ou estrutura, é um bloco que armazena diversas informações, de diversos tipos.
- Podem ter quantos elementos e tipos se deseja.
- Pode ser considerado um novo tipo de dado. Um tipo, assim como int, float ou char. Mas um tipo que você criou.
- São muito usadas quando temos elementos em nossos programas que precisam e fazem uso de vários tipos de variáveis e características.



4

- **Structs**

- Tipo de dado estruturado heterogêneo;
- Coleção de variáveis referenciadas sobre um mesmo nome;
- Permite agrupar dados de diferentes tipos numa mesma estrutura:
 - Ao contrário de matrizes que possuem elementos de um mesmo tipo;
- Utilizado para armazenar informações de um mesmo objeto, conjunto de dados relacionados:
 - Carro → cor, marca, ano, placa, chassi
 - Pessoa → nome, idade, endereço

- **Structs**

- Cada componente de um registro pode ser de um tipo diferente:
 - (int, char, ...);
- Estes componentes são referenciados por um identificador;

- **Elementos de um Struct**

- Denominados campos ou membros do registro;
- Exemplo: nome.

• Structs na linguagem C

- A palavra reservada *struct* indica ao compilador que está sendo criado um registro;
- Uma *struct* normalmente é declarada após incluir as bibliotecas e antes da função *main*;

- Sintaxe:

```
struct <nome_da_estrutura> {  
    tipo <nome_variável_campo1>;  
    tipo <nome_variável_campo2>;  
    ...  
};
```



7

Exemplo:

```
struct clientes {  
    char nome[30];  
    char endereco[50];  
    int idade;  
    float salario;  
};
```



8

Linguagem C - *Structs*

- No exemplo anterior foi declarado um tipo *struct cliente*, porém não foi criada nenhuma variável do tipo *struct*;
- A partir desse ponto, poderão ser declaradas variáveis do tipo *struct cliente*. O comando para criar a variável é:

struct clientes dados;



9

- Para facilitar podemos usar a palavra reservada ***typedef*** : um atalho em C para que possamos nos referir a um determinado tipo existente com nomes sinônimos.

```
typedef struct {  
    char nome[30];  
    char endereco[50];  
    int idade;  
    float salario;  
} clientes;
```



10

Linguagem C - *Structs*

- Seguindo o exemplo anterior a struct ***clientes*** possui os campos: nome, endereço, idade e salário.
- Porém esses campos não existem isoladamente, só existem dentro da struct ***clientes***.

- Para acessar os campos de uma variável tipo ***struct***, indicamos o nome da variável e depois qual campo desejamos dentro dela.

`clientes cadastro; // declarando uma variável do tipo struct cliente`

- Exemplos:

`cadastro.nome;
cadastro.idade;`

- Exemplo de atribuição de valor:

`cadastro.idade=34;
strcpy(cadastro.nome,"Paulo");
gets(cadastro.nome);
cadastro.salario=1250.55;`

Linguagem C - Structs

- Criando uma struct

```
typedef struct
{
    char nome[20];
    char end[50];
    char telefone[11];
} cliente;
```



13

- Criando uma variável do tipo struct:

cliente cadastro;

- Preenchendo uma *struct*.

```
printf("\n Digite seu Nome : ");
gets(cadastro.nome);
printf("\n Digite seu Endereco : ");
gets(cadastro.end);
printf("\n Digite seu Telefone : ");
gets(cadastro.telefone);
```



14

Linguagem C - *Structs*

- Acessando os dados de uma *struct*:

```
printf("\n Nome : %s", cadastro.nome);  
printf("\n Endereco : %s", cadastro.end);  
printf("\n Telefone : %s", cadastro.telefone);
```



15

Structs com vetor

- Criando uma variável do tipo struct com vetor:

Cliente cadastro[3];

- Preenchendo uma *struct*:

```
for (i=0; i<3; i++){  
    printf("\n Digite seu Nome : ");  
    gets(cadastro[i].nome);  
    printf("\n Digite seu Endereco : ");  
    gets(cadastro[i].end);  
    printf("\n Digite seu Telefone : ");  
    gets(cadastro[i].telefone);  
}
```



16

Linguagem C - *Structs*

- Acessando os dados de uma *struct*:

```
for (i=0; i<3; i++)  
{  
    printf("\n Nome : %s", cadastro[i].nome);  
    printf("\n Endereco : %s", cadastro[i].end);  
    printf("\n Telefone : %s", cadastro[i].telefone);  
    printf("\n\n");  
}
```

EXEMPLOS

Linguagem C - *STRUCTS*

Exemplos:

Criando a STRUCT:

```
struct {  
    char nome[20];  
    int idade;  
    float salario;  
    char end[50];  
    char telefone[11];  
} clientes;
```

Declarando uma variável do tipo STRUCT:

```
clientes cadastros;
```



19

Linguagem C - *STRUCTS*

Exemplos:

Lendo e atribuindo valores a uma STRUCT:

```
strcpy(cadastros.nome, "Flavio");  
  
scanf("%i", &cadastros.idade);  
  
cadastros.salario = 1250.55;  
  
gets(cadastros.end);  
  
strcpy(cadastros.telefone, "51-34345656");
```



20

Linguagem C - *STRUCTS*

Exemplos:

Escrevendo valores de uma STRUCT:

```
printf("\n Nome : %s", cadastros.nome);

printf("\n Idade : %i", cadastros.idade);

printf("\n Salario : %.2f", cadastros.salario);

printf("\n Endereco : %s", cadastros.end);

printf("\n Telefone : %s", cadastros.telefone);
```



21

```
/* acessando os campos de uma estrutura */

#include <stdio.h>

/* criando um novo tipo de dado "produto" */
struct produto
{
    int codigo;
    char nome[50];
    int quantidade;
    float valor_compra;
    float valor_venda;
};

int main()
{
    struct produto item; /* declarando uma variável "item" do tipo "struct produto" */

    printf("Preenchendo a variável 'item'\n");
    printf("Item.....:");
    fgets(item.nome, 50, stdin);
    printf("Código.....:");
    scanf("%d",&item.codigo);
    printf("Quantidade.....:");
    scanf("%d",&item.quantidade);
    printf("Valor de compra.:");
    scanf("%f",&item.valor_compra);
    printf("Valor de venda.:");
    scanf("%f",&item.valor_venda);
    printf("\n");
    printf("Exibindo os dados\n");
    printf("Código.....:%d\n",item.codigo);
    printf("Item.....:%s",item.nome);
    printf("Quantidade.....:%d\n",item.quantidade);
    printf("Valor de compra: %.2f\n",item.valor_compra);
    printf("Valor de venda: %.2f\n",item.valor_venda);

    return 0;
}
```



22

```

#include <conio.h>
#include <stdio.h>

// uma estrutura Pessoa
struct Pessoa{
    char *nome;
    int idade;
};

int main() {
    // declara uma variável do tipo struct
    struct Pessoa cliente;
    cliente.nome = "Osmar J. Silva";
    cliente.idade = 36;

    // obtém os dados
    printf("O nome do cliente e: %s\n", cliente.nome);
    printf("A idade do cliente e: %d\n", cliente.idade);

    return 0;
}

```



23

```

#include <iostream>
#include <cstdlib>
#include <cstring>

using namespace std;

typedef struct data {
    int dia;
    int mes;
    int ano;
} DataDMA;

typedef struct aniversario {
    char nome[50];
    DataDMA nascimento;
} Aniversario;

int main (){
    Aniversario alguem;

    printf("Digite o nome de alguem ");
    fgets(alguem.nome, 50, stdin);
    printf("Digite o dia que esta pessoa nasceu ");
    scanf("%i",&alguem.nascimento.dia);
    printf("Digite o mes que esta pessoa nasceu ");
    scanf("%i",&alguem.nascimento.mes);
    printf("Digite o ano que esta pessoa nasceu ");
    scanf("%i",&alguem.nascimento.ano);

    system("cls");
    printf(" %s nasceu em %i/%i/%i\n\n",alguem.nome, alguem.nascimento.dia,
    alguem.nascimento.mes,alguem.nascimento.ano);

    system("pause");
    return EXIT_SUCCESS;
}

```



24

```
// EXEMPLO BASICO DE STRUCT EM C
#include <stdio.h>
#include <stdlib.h>

struct funcionario {
    char nome[50];
    int idade;
};

main () {
    struct funcionario func[50];
    int i = 0;

    for (i=1; i<3; i++)
    {
        printf("\nNome do %uo Funcionario: ", i);
        scanf("%s", &func[i].nome);

        printf("\nIdade do %uo Funcionario: ", i);
        scanf("%i", &func[i].idade);
    };

    for (i=1; i<3; i++)
    {
        printf("-----\n");
        printf("Nome do %uo Funcionario: %s ", i, &func[i].nome);
        printf("\nIdade do %uo Funcionario: %i \n", i, func[i].idade);
        printf("-----\n");
    };
    return 0;
}
```



25

```
/*Defina uma struct para tratar de alunos. Dentro dessa struct, crie uma variável para armazenar o nome do aluno, e outras para armazenar as notas de Matemática, Física e a média dessas duas notas. Preencha os nomes e notas dos alunos, calculando automaticamente a média deles. Depois exiba tudo isso.
*/

#include <stdio.h>
#include <iostream>
#include <stdlib.h>
#include <string>

int main(void)
{
    struct Alunos
    {
        char nome[30];
        float matematica, fisica, media;
    };

    struct Alunos alunos[5];
    int count;

    for(count = 0 ; count < 5 ; count++)
    {
        fflush(stdin);
        printf("\nNome do aluno %d: ", count+1);
        gets(alunos[count].nome);

        printf("Nota de matematica: ");
        scanf("%f", &alunos[count].matematica);

        printf("Nota de fisica: ");
        scanf("%f", &alunos[count].fisica);

        alunos[count].media = (alunos[count].matematica + alunos[count].fisica)/2;
    }

    printf("\nExibindo nomes e medias:\n");
    for(count = 0 ; count < 5 ; count++)
    {
        printf("\nAluno %d\n", count+1);
        printf("Nome: %s\n", alunos[count].nome);
        printf("Media: %.2f\n", alunos[count].media);
    }

    return 0;
}
```



26

Exercício

Defina um registro empregado para guardar os dados (nome, sobrenome, data de nascimento, RG, data de admissão, salário) de um empregado de sua empresa. Defina um vetor de empregados para armazenar todos os empregados de sua empresa.



27

Exercício

O Sr. João tem cinco táxis e deseja calcular o gasto de combustível, por dia, de cada um dos táxis. Escreva um algoritmo para ler a placa do táxi, a data, a marcação do odômetro (km) no início do dia, a marcação no final do dia. Guardar estes dados, bem como o valor gasto de combustível, quantidade de quilômetros percorridos, a quantidade de litros de combustível gastos em uma Estrutura de Dados Compostos (Structs).

Sabe-se que os táxis do Sr. João percorrem 10 km com um litro de combustível e que o preço do litro do combustível é de R\$ 3,05. Para fazer o cálculo do valor do gasto de combustível utilizar uma função com valor de retorno.

Escrever, para cada táxi, a placa do táxi, a data, a quantidade de quilômetros percorridos, a quantidade de litros de combustível gastos, o valor total (R\$) gasto de combustível.



28