

Programação Estruturada

- Encontro 10 -

Sistemas de Informação
Prof.º Philippe Leal
philippeleal@yahoo.com.br



Introdução

- Um **registro** é uma coleção de várias variáveis, possivelmente de tipos diferentes.
- Na linguagem C, registros são conhecidos como ***struct*** (abreviatura de *structures*).
- Exemplo:

```
struct dados{  
    char nome[50];  
    int idade;  
    float salario;  
};
```



Declaração de uma Struct

- Um **registro** é declarado da seguinte maneira:

```
struct nome_da_struct{  
    tipo1 campo1;  
    tipo2 campo2;  
    tipo3 campo3;  
    ...  
    tipoN campoN;  
};
```

Os membros dos registros são chamados de **elementos** ou **campos**.

- Exemplo anterior:

```
struct dados{  
    char nome[50];  
    int idade;  
    float salario;  
};
```



Declaração de uma variável do tipo Struct

- A forma geral para declarar uma variável do tipo de uma struct criada é:

struct *nome_da_struct* *nome_da_variável*

- Assim, para se declarar uma variável chamada **funcionarios** do tipo da struct **dados** do exemplo, devemos escrever:

struct dados funcionarios;

nome da struct


nome da variável



Acessando os Campos

- A forma geral para acessar cada elemento de uma struct é:

nome_da_variável.nome_do_campo



Observe o **ponto**
separando os nomes

- Então, para acessar o campo **idade** da variável **funcionarios** do exemplo, faça:

```
funcionarios.idade = 21;
```



Exemplo 1

```
1 #include <stdio.h>
2
3 struct dados{
4     char nome[50];
5     int idade;
6     float salario;
7 };
8
9 int main(){
10
11     struct dados funcionarios;
12
13     printf("\nDigite o nome do funcionario, idade e salario: \n");
14     scanf("%s", funcionarios.nome);
15     scanf("%d", &funcionarios.idade);
16     scanf("%f", &funcionarios.salario);
17
18     printf("\n\nDados do funcionario: ");
19     printf("\nNome: %s", funcionarios.nome);
20     printf("\nIdade: %d", funcionarios.idade);
21     printf("\nSalario: %.2f\n\n", funcionarios.salario);
22
23     return 0;
24 }
```



Vetor de Structs

- A forma geral para declarar um vetor de structs é:

```
struct nome_da_struct nome_do_vetor [tamanho_do_vetor]
```

- Portanto, para se declarar um vetor chamado **funcionarios** (com 50 elementos) do tipo da struct **dados** do exemplo, devemos escrever:

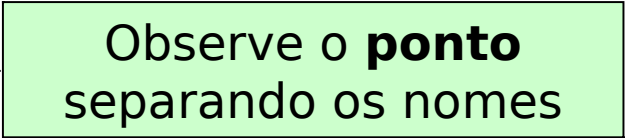
```
struct dados funcionarios[50];
```



Acessando os Campos de Vetor de Structs

- A forma geral para acessar cada elemento de um vetor de structs é:

nome_da_variável[índice].nome_do_campo



Observe o **ponto**
separando os nomes

- Então, para acessar o campo **salário** do sexto elemento do vetor **funcionarios**, faça:

```
funcionarios[5].salario = 1400;
```


Exemplo 2

```
1 #include <stdio.h>
2
3 #define TAM 30
4
5 struct dados{
6     char nome[50];
7     int idade;
8     float salario;
9 };
10
11 int main(){
12
13     struct dados funcionarios[TAM];
14     int i;
15
16     for(i = 0; i < TAM; i++){
17         printf("\nDigite o nome, idade e salario do %do. funcionario: \n", i+1);
18         scanf("%s", funcionarios[i].nome);
19         scanf("%d", &funcionarios[i].idade);
20         scanf("%f", &funcionarios[i].salario);
21     }
22
23     for(i = 0; i < TAM; i++){
24         printf("\n\nDados do %do. funcionario: ", i+1);
25         printf("\nNome: %s", funcionarios[i].nome);
26         printf("\nIdade: %d", funcionarios[i].idade);
27         printf("\nSalario: %.2f\n\n", funcionarios[i].salario);
28     }
29
30     return 0;
31 }
```



Exercícios

- 1)** Faça um algoritmo que utilize a struct dos exemplos anteriores e armazene (em um vetor) os dados de 20 funcionários. Em seguida, imprima quantos deles possuem 30 anos ou mais.
- 2)** Faça um algoritmo que utilize a struct dos exemplos anteriores e armazene (em um vetor) os dados de 30 funcionários. Em seguida, imprima (com duas casas decimais) a soma dos salários dos que possuem 20 anos ou mais.
- 3)** Considere uma struct que armazene o nome, idade, salário e cargo de 25 funcionários. Faça um algoritmo que imprima quantos Diretores ganham mais do que R\$ 5.000,00.