

INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
**PARAÍBA**  
Campus João Pessoa



Tecnologia em Sistemas para Internet

# Linguagem “C”

Struct

valeriacavalcanti.com.br

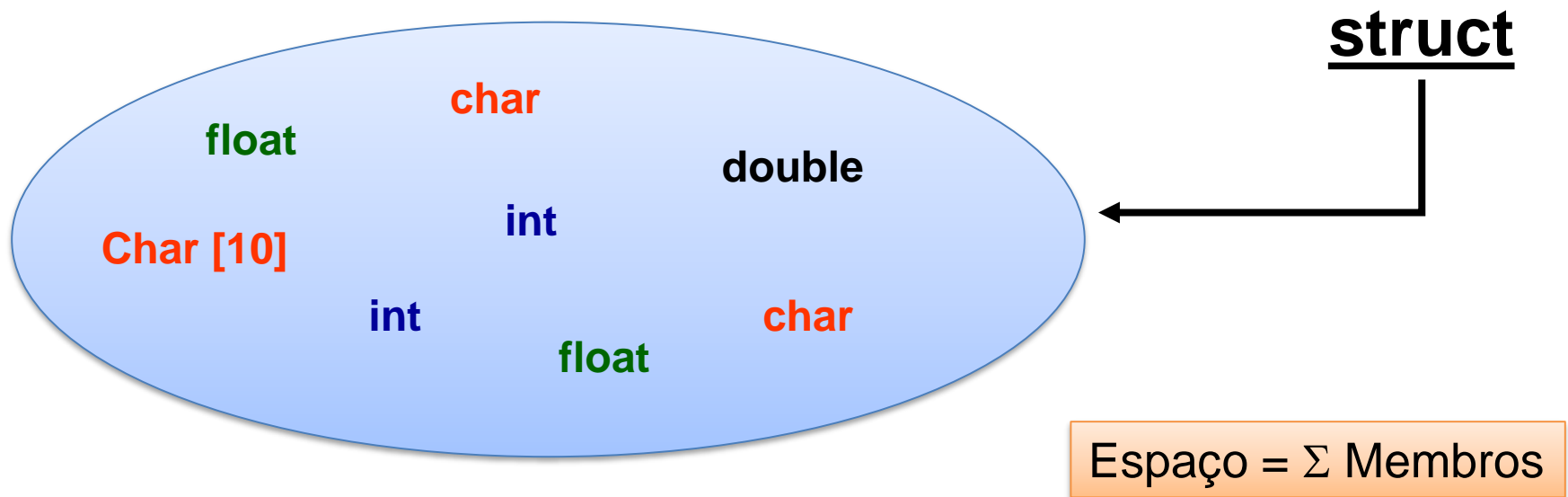
**Profa. Valéria Cavalcanti**

valeria.cavalcanti@ifpb.edu.br



# Definição

- É a definição de um nome para um agrupamento de variáveis, podendo ser de tipos diferentes.



# Declaração

## Formato:

- `struct rótulo {lista de declarações} var_e;`
- A definição do rótulo para estrutura é opcional;
- Os operadores “.” e “->” dão acesso aos membros da estrutura.

## Exemplos:

```
struct Ponto {  
    int x, y;  
};
```

A

```
struct {  
    int x, y;  
}ponto;
```

B

```
struct Ponto {  
    int x, y;  
}p1, p2, p3;
```

C

# Declaração

```
struct Ponto {  
    int x, y;  
};
```

A

*É uma estrutura com rótulo definido (Ponto) e dois membros do tipo inteiro (x e y).*

## Usando a “struct” para declarar:

Variável:            struct Ponto ponto, p1, Ponto;

Array:               struct Ponto pontos[10];

Apontador:          struct Ponto \*ptPonto;

# Declaração

```
struct {  
    int x, y;  
}ponto;
```

B

*É uma estrutura sem rótulo definido, dois membros do tipo inteiro (x e y) e uma variável (ponto).*

## Usando a “struct” para declarar:

Variável: não pode ser declarada;

Array: não pode ser declarado;

Apontador: não pode ser declarado;

# Declaração

```
struct Ponto {  
    int x, y;  
}p1, p2, p3;
```



*É uma estrutura com rótulo definido (Ponto), dois membros do tipo inteiro (x e y) e três variáveis (p1, p2, p3).*

## Usando a “struct” para declarar:

Variável:            struct Ponto p4, p5;

Array:               struct Ponto pts[10];

Apontador:         struct Ponto \*pt;

# Atribuindo Valores

## Inicializando:

- Coloca-se uma lista de inicializadores, separados por vírgulas;
- Se o número de inicializadores for inferior ao número de membros, os que restarem automaticamente serão inicializados com 0 ou NULL, dependendo do seu tipo;
- Comparação entre estruturas !

## Exemplo:

- `struct Ponto p1 = {5, 6}; // x = 5 e y = 6`
- `struct Ponto p2 = {5}; // x = 5 e y = 0`
- `p2 = p1; // copia dados de p1 em p2`

# Estruturas e Funções

- As estruturas podem ser passadas a funções, passando:  
Cada membro separadamente;  
A estrutura inteira;  
Apontador para estrutura.

## Atenção !

A passagem da estrutura é por valor;  
Passagem de array, em C, por default é por referência;  
Array por valor → struct com um vetor.



# Exercício

- **Exemplo\_1:**

Faça um programa que leia do usuário as coordenadas x e y de um ponto, depois use uma função para informar em que quadrante o ponto se encontra.

Considere que o ponto não está localizado nos eixos.

- **Vamos Precisar:**

Estrutura para guardar o ponto;

Função para encontrar o quadrante do ponto;

Programa principal (main) para testar.

# Resposta

```
#include <stdio.h>

struct ponto {
    float x, y;
};

int main() {
    struct ponto p;

    printf("Coordenada 'x': ");
    scanf("%f", &p.x);

    printf("Coordenada 'Y': ");
    scanf("%f", &p.y);

    if (p.x > 0)
        if (p.y > 0) puts("Primeiro quadrante");
        else puts("Quarto quadrante");
    else
        if (p.y > 0) puts("Segundo quadrante");
        else puts("Terceiro quadrante");

    return 0;
}
```

# Exercício

## Exemplo\_2:

Escreva um programa, em C, para ler nome e salário de **10**(dez) empregados.

Após a leitura dos dados deve ser exibido o nome e o salário do empregado que recebe o maior salário.

**Desconsidere empate!**

# Resposta

```
#include <stdio.h>
#define TAM 4

struct empregado {
    char nome[31];
    float salario;
};

int main(){
    int i;
    struct empregado e, maior;

    printf("Informe o nome do 1 empregado: ");
    gets(maior.nome);
    printf("Informe o salario do empregado: ");
    scanf("%f%c", &maior.salario);

    for (i = 2; i <= TAM; ++i){
        printf("Informe o nome do %d empregado: ", i);
        gets(e.nome);
        printf("Informe o salario do empregado: ");
        scanf("%f%c", &e.salario);
        if (e.salario > maior.salario)
            maior = e;
    }

    printf("Maior: %s - %.2f\n", maior.nome, maior.salario);

    return 0;
}
```

# Estruturas Aninhadas

## Definição:

- Uma estrutura pode conter como membro uma outra estrutura.

## Exemplo:

- Estrutura circunferência, composta por um ponto e um raio:

```
struct ponto{  
    float x, y;  
};
```

```
struct circunferencia{  
    struct ponto centro;  
    float raio;  
};
```

# Estruturas Aninhadas

## Inicializando:

- Inicializa da mesma forma das estruturas simples.

## Exemplos:

- `struct circunferencia c1 = {1, 1, 5.6};`
- `struct circunferencia c2 = {{1, 1}, 5.6};`
- `struct circunferencia c3 = {1, 1};`
- `struct circunferencia c4 = {{1, 1}};`
- `struct circunferencia c5 = c1;`

# Estruturas Aninhadas

- **Acessando os Membros:**

```
struct circunferencia{  
    struct ponto centro;  
    float raio;  
};
```

**Variável: struct circunferencia c;**

- Coordenada x do centro (Ponto): c.centro.x;
- Coordenada y do centro (Ponto): c.centro.y;
- Raio da circunferencia: c.raio;

# typedef

- Permite ao programador definir um novo nome a um determinado tipo (sinônimos);

## **Exemplos:**

- `typedef int inteiro;`
- `typedef float real;`
- `typedef struct ponto Ponto;`



# typedef

- Definindo um “novo tipo de dado”, para guardar as informações de empregados:

```
typedef struct {  
    char nome[31];  
    int matricula;  
    float salario;  
    char cargo[20];  
}Empregado;
```

```
Empregado empregado, empregados[10];
```

# typedef

## Exemplo:

- Escreva um programa, em C, para ler o nome e o preço de vários chocolates. O programa deverá encerrar quando for digitado um valor negativo para o chocolate.
- Calcule e exibe o nome do chocolate mais caro e o mais barato.

# typedef

```
#include <stdio.h>

typedef struct {
    char nome[31];
    float valor;
}Chocolate;

int main(){
    int i, qtde = 0;
    Chocolate c, maior, menor;

    printf("Informe o nome do 1 chocolate: ");
    gets(c.nome);
    printf("Informe o valor: ");
    scanf("%f%c", &c.valor);
    maior = c;
    menor = c;

    while (c.valor > 0){
        ++qtde;
        if (c.valor > maior.valor) maior = c;
        else if (c.valor < menor.valor) menor = c;
        printf("Informe o nome do %d chocolate: ", qtde + 1);
        gets(c.nome);
        printf("Informe o valor: ");
        scanf("%f%c", &c.valor);
    }

    if (qtde > 0){
        printf("Maior = %s - %.2f\n", maior.nome, maior.valor);
        printf("Menor = %s - %.2f\n", menor.nome, menor.valor);
    }else puts("Nenhum chocolate informado!");

    return 0;
}
```

# typedef: Exemplo

Data: Define uma data composta por dia, mês e ano.

Pessoa: Define uma pessoa composta por nome e data de nascimento.

```
#include <stdio.h>
#include <string.h>

typedef struct{
    int dia, mes, ano;
}Data;

typedef struct{
    char nome[31];
    Data nascimento;
}Pessoa;

int main(){
    Pessoa p1, p2;

    strcpy(p1.nome, "Fulaninho de Tal");
    p1.nascimento.dia = 1;
    p1.nascimento.mes = 10;
    p1.nascimento.ano = 2000;

    p2 = p1;

    print_pessoa(p1);
    print_pessoa(p2);

    return 0;
}

void print_pessoa(Pessoa p){
    printf("Nome: %s (", p.nome);
    print_data(p.nascimento);
    printf(")\n");
}

void print_data(Data d){
    printf("%d/%d/%d", d.dia, d.mes, d.ano);
}
```