

## Variável Composta Heterogênea (Estruturas / Registros)



Centro Federal de Educação Tecnológica Celso Suckow da Fonseca  
CEFET-RJ

# *Estruturas*

- Estruturas são grupos de variáveis relacionadas entre si sob um mesmo nome. É conhecido também como variável composta heterogênea.

## Definição

**struct conta**

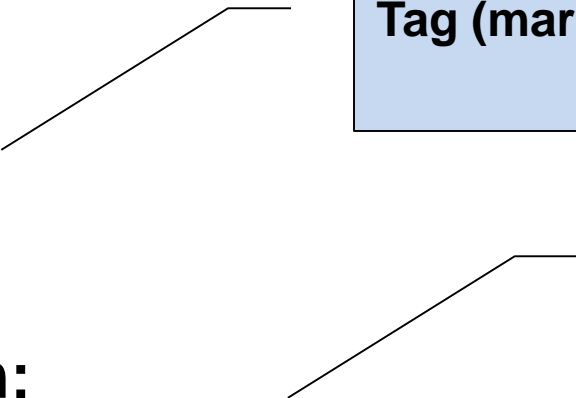
**{**

**int num;**

**string nome;**

**float saldo;**

**};**



**Tag (marca, rótulo, etiqueta) da estrutura**

**Membros da estrutura**

- **Por que heterogênea?**
  - Os membros não precisam ser do mesmo tipo.
  - Podem ser:
    - Tipos básicos (int, float etc)
    - Arrays
    - Outras estruturas
- **A definição de uma estrutura não reserva espaço na memória.**
  - Usa um novo tipo que será usado para declarar variáveis.

## *Declaração de variáveis do tipo definido na estrutura*

```
struct conta c, cc[20], *cPtr;
```

- Declarou a variável c do tipo struct conta.
- Declarou um array cc com 20 elementos do tipo struct conta.
- Declarou \*cPtr como um ponteiro para struct conta.

Igualmente:

```
struct conta  
{  
    int num;  
    string nome;  
    float saldo;  
} c, cc[20], *cPtr;
```

**O rótulo é opcional.**

Atenção:

Se uma definição de estrutura não possuir rótulo, então as variáveis só poderão ser declaradas no momento de sua definição.

## *Inicialização*

```
struct conta c = {121, "Maria", 2500};
```

**Por default, os membros são inicializados automaticamente com 0 (ou NULL, se for um ponteiro).**

- **Operador de membro (operador de ponto)**

```
cout << c.num;
```

- **Operador de ponteiro (operador de seta)**

- Considere: struct conta c, \*cPtr;

```
c.num = 1010;
```

```
c.nome = "Maria";
```

```
c.saldo = 2500;
```

```
cPtr = &c;
```

```
cout << c.num << c.nome;
```

```
cout << cPtr->num << cPtr->nome;
```

```
cout << (*cPtr).num << (*cPtr).nome;
```

# *Typedef*

- Criação de sinônimos (apelidos) para tipos de dados definidos previamente.

**typedef struct conta Conta; ou**

```
typedef struct {  
    int num;  
    string nome;  
    float saldo;  
} Conta;
```

**Conta é um sinônimo para struct conta**

**Agora, Conta pode ser usada para declarar variáveis do tipo struct conta.**

**Exemplo:**

**Conta cc[20];**

## Ponteiro

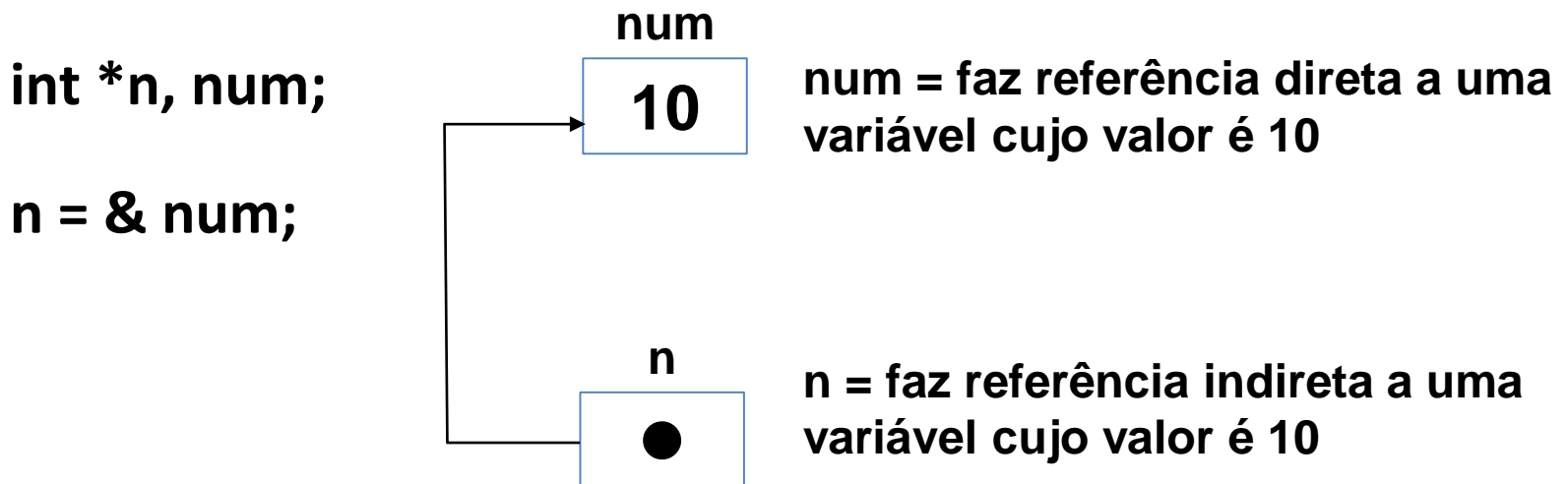


**Centro Federal de Educação Tecnológica Celso Suckow da Fonseca  
CEFET-RJ**



# Ponteiro

- Permitem aos programas simular chamadas por referência, criar e manipular estruturas dinâmicas de dados (listas encadeadas, filas, pilhas, por exemplo).
- São variáveis que armazenam endereços de memória como valores.



## *Declaração e inicialização*

- Devem ser declarados antes de serem usados.

**int \*n;**

- Declara a variável n como um ponteiro para um inteiro.
- Todo ponteiro, na declaração, tem que ter o \* na frente.
- Podem assumir valores como 0, NULL ou um endereço de memória.

# Operador de Ponteiro

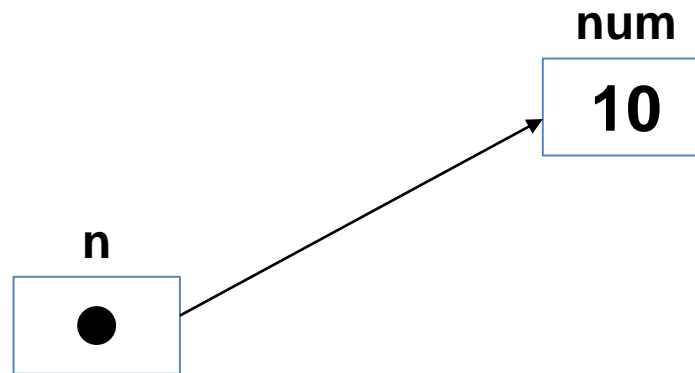
- **& - operador de ponteiro**

- operador unário que retorna o endereço de seu operando.

- **Exemplo:**

```
int num = 10, *n;
```

```
n = &num;
```



- atribui o endereço da variável num à variável de ponteiro n.
- Diz-se que n aponta para num.

## *Operador de Ponteiro*

- O operador **\*** é denominado como:
  - operador de referência indireta ou operador de desreferenciamento

**cout << \*n;**

**apresenta o valor da variável num, ou seja, 10.**

## *Exemplo - Ponteiro*

```
#include <iostream>
using namespace std;
main()
{
    int a, *aPtr;

    a = 15;
    aPtr = &a;

    cout << "a: " << a;
    cout << "\naPtr aponta para a variável a : " << *aPtr;
    cout << "\nEndereco de a: " << &a;
    cout << "\nEndereco de a atraves de aPtr: " << aPtr;
}
```

## *Exemplo - Ponteiro e Registro*

**Faça um programa que realize o cadastro de contas bancárias com as seguintes informações: número da conta, nome do cliente e saldo. O banco permitirá o cadastramento de 20 contas e não pode haver mais de uma conta com o mesmo número.**

**Crie um menu:**

- 1. Cadastrar Conta**
- 2. Consultar Conta**
- 3. Listar números das contas com saldo negativo**
- 4. Sair**

## *Exemplo (Resposta)*

```
#include <iostream>
#define T 3

using namespace std;

typedef struct{
    int num;
    string nome;
    float saldo;
} Conta;

//Protótipos das funções
int menu();

void cadastrar(int *, Conta *conta);

void consultar(Conta *conta);

void listar(Conta *conta);

int achou(int, Conta *conta, int *);
```

## *Continuação (Programa Principal)*

```
main() {  
    int opcao, achou = 0, i = 0;  
    Conta cc[T];  
    do {  
        opcao = menu();  
        switch(opcao)  
        {  
            case 1: cadastrar(&i, cc); break;  
            case 2: consultar(cc); break;  
            case 3: listar(cc); break;  
        }  
    } while (opcao !=4);  
}
```



## *Continuação (Menu)*

**// Funções**

**int menu()**

**{**

**int op;**

**cout << "\n\n\*\*\*\* Menu Principal \*\*\*\*";**

**cout << "\n1. Cadastrar Conta";**

**cout << "\n2. Consultar conta";**

**cout << "\n3. Listar contas com saldo negativo";**

**cout << "\n4. Sair\n";**

**cout << "Escolha uma opcao: ";**

**cin >> op;**

**return op;**

**}**

## *Continuação (Cadastro)*

```
void cadastrar(int *i, Conta *conta)
{
    int x;
    cout << "Cadastrar Conta\n";
    if (*i < T){

        cout << "\nDigite o numero da conta a ser cadastrada: ";
        cin >> x;
        if (achou(x, conta, i) == 0)
        {
            cout << "Digite o nome: ";
            cin >> conta[*i].nome;
            cout << "Digite o valor a ser depositado: ";
            cin >> conta[*i].saldo;
            conta[*i].num = x;
            *i=*i+1;
        }
        else cout << "\nNumero ja cadastrado.\n";
    } else cout << "\nLimite maximo de contas ultrapassado.\n";
}
```

## *Continuação (Consulta)*

```
void consultar (Conta *conta) {  
    int x, t = 0;  
    cout << "*****\n";  
    cout << "Consultar Conta\n";  
    cout << "\nDigite o numero da conta para consulta: ";  
    cin >> x;  
    if (achou(x, conta, &t) == 1)  
    {  
        //cout << "\n" << conta[*t].num;  
        cout << "\n" << conta[t].nome;  
        cout << "\n" << conta[t].saldo;  
    }  
    else cout << "\nNumero de conta nao cadastrado.\n";  
}
```

## *Continuação (Listagem)*

```
void listar(Conta *conta)
{
    cout << "\n*** Listagem das contas negativas ***\n";
    for (int i = 0; i < T; i++)
    {
        if (conta[i].saldo < 0)
        {
            cout << "\n" << conta[i].num;
            cout << "\t" << conta[i].nome;
        }
    }
}
```

## *Continuação (Busca)*

```
int achou(int num, Conta *conta, int *pos) {  
    int j=0, aux=0;  
    while (aux == 0 && j < T)  
    {  
        if (conta[j].num == num)  
        {  
            aux = 1;  
            *pos = j;  
        }  
        else  
            j++;  
    }  
    return aux;  
}
```

## *Exercício*

- 1. Faça um programa que acrescente ao Exemplo ao menu:**
  - 4. Classificação Crescente de saldo**
  - 5. Listagem Geral das Contas**
  - 6. Exclusão de Conta**
  - 7. Sair**