

# Estruturas de dados heterogêneas (registros)

## 1 - Estruturas de dados heterogêneas

Uma estrutura de dados heterogênea permite que uma única variável armazene mais de um valor, sendo que estes valores podem ser de tipos diferentes (por isso o nome heterogêneo). Para tanto, utiliza-se a estrutura de registro – ou **struct** em linguagem C – que é composto por campos diferenciados onde os valores são armazenados.

Por exemplo, podemos criar um registro para armazenar informações de uma pessoa:

```
#include <stdio.h>

struct Pessoa{
    char nome[31];
    int idade;
    float altura;
};
```

Neste caso, a estrutura tem três membros: um array de caracteres para o nome da pessoa, um inteiro para a idade e um float para a altura.

Para utilizar essa estrutura para o armazenamento de um único registro, podemos criar uma variável dessa forma:

```
#include <stdio.h>

struct Pessoa{
    char nome[31];
    int idade;
    float altura;
};

int main(){
    struct Pessoa pessoa;
    .
    .
    .
    return 0;
}
```

E então acessar e modificar os membros do registro usando o operador ponto (.):

```
#include <stdio.h>

struct Pessoa{
    char nome[31];
```

```

int idade;
float altura;
};

int main(){

    struct Pessoa pessoa;

    printf("Informe nome: ");
    scanf(" %30[^\n]", pessoa.nome);

    printf("Informe idade: ");
    scanf("%d", &pessoa.idade);

    printf("Informe altura: ");
    scanf("%f", &pessoa.altura);

    return 0;
}

```

**Importante:** Como as variáveis devem ser criadas a partir de um registro, o registro sempre deve ser declarado no início do código, antes da criação da função **main**. Uma vez declarado o registro, pode-se criar as variáveis deste tipo em qualquer função do programa.

**Exemplo:** Elaborar um programa em C que leia o nome e as 2 notas de um aluno e armazene estes dados em um registro. Calcule a média do aluno e apresente na tela seu nome e a média.

```

#include <stdio.h>

struct Aluno{
    char nome[31];
    float nota1, nota2;
}

int main(){
    struct Aluno aluno;
    float media;

    printf("Informe o nome do aluno: ");
    scanf(" %30[^\n]", aluno.nome);

    printf("Informe nota 1: ");
    scanf("%f", &aluno.nota1);

    printf("Informe nota 2: ");
    scanf("%f", &aluno.nota2);

    media = (aluno.nota1 + aluno.nota2) / 2;
}

```

```
printf("\nNome: %s", aluno.nome);
printf("\nMédia: %.1f", media);

return 0;
}
```

Uma utilização comum de registros é criar vetores ou matrizes de registros. No código anterior, foram armazenados os dados de apenas 1 aluno. Se o enunciado se estendesse para o armazenamento do nome e as 2 notas de 5 alunos de uma turma, o código poderia ser reescrito da seguinte forma:

```
#include <stdio.h>

struct Aluno{
    char nome[31];
    float nota1, nota2;
};

int main(){
    struct Aluno aluno[5];
    float media;
    int i;

    for(i = 0; i < 5; i++){
        printf("Informe o nome do aluno: ");
        scanf (" %30[^\n]", aluno[i].nome);

        printf("Informe nota 1: ");
        scanf ("%f", &aluno[i].nota1);

        printf("Informe nota 2: ");
        scanf ("%f", &aluno[i].nota2);

        media = (aluno[i].nota1 + aluno[i].nota2) / 2;

        printf("\nNome: %s", aluno[i].nome);
        printf("\nMédia: %.1f\n", media);
    }

    return 0;
}
```

## 5 - Atividades

1. Crie uma estrutura representando um aluno de uma disciplina. Essa estrutura deve conter o nome do aluno, o nome da disciplina e as notas do aluno em duas provas. Em seguida, leia os dados de três alunos e os armazena nessa estrutura. Por fim, exiba o nome do aluno, o nome da disciplina e a média dos alunos que foram aprovados com média igual ou superior a 6.

2. Suponha que você está fazendo uma pesquisa de preços para a compra de um tablet. Para auxiliá-lo, escreva um programa que permita o cadastro de cinco registros, cada um contendo o nome de uma loja e o preço do tablet naquela loja. O programa deve calcular a média dos preços e mostrar uma relação contendo o nome das lojas que tenham o tablet com preço abaixo da média.
3. Crie uma estrutura representando um atleta. Essa estrutura deve conter o nome do atleta, seu esporte, idade e altura. Em seguida, leia os dados de 5 atletas. Calcule e exiba os nomes do atleta mais alto e do atleta mais velho.
4. Crie uma estrutura com o código e a quantidade em estoque de 5 produtos. Em seguida, leia um pedido composto por um código de produto e quantidade. Se houver quantidade suficiente para atender ao pedido integralmente, atualize a quantidade em estoque do produto e apresente a mensagem "Estoque atualizado", caso contrário, apresente a mensagem "Quantidade insuficiente em estoque".
5. Escreva um programa que simule contas bancárias com as seguintes especificações:
  - Ao iniciar o programa deverá ser criada uma estrutura para as contas bancárias contendo o número da conta, o nome do titular da conta e o saldo.
  - Devem ser cadastrados 3 clientes diretamente no programa, com saldo inicial de 0.
  - Crie um menu para o programa com as seguintes opções:
    - [1] Listagem de clientes
    - [2] Depósito
    - [3] Saque
    - [4] Fim
  - Para cada depósito ou saque realizado, deverá ser mostrado o nome do titular e o saldo atualizado. O programa não deve permitir saques com valores superiores ao saldo atual do cliente.