

## Tipos de dados compostos heterogêneos

Como vimos em sala de aula, uma variável composta heterogênea é uma estrutura que permite criar seu próprio tipo de dado. Uma estrutura permite que você tenha diferentes variáveis internas com diferentes tipos de dados para cada uma. Ela pode ser utilizada de duas formas:

Modo 1	<pre>struct ID_ESTRUTURA {     TIPO ID_VAR1;     TIPO ID_VAR2;     ... };</pre>
Modo 2	<pre>typedef struct {     TIPO ID_VAR1;     TIPO ID_VAR2;     ... } ID_ESTRUTURA;</pre>

onde:

- ID\_ESTRUTURA é o identificador que você escolhe (cria, gera, dá o nome) para identificar a estrutura;
- TIPO é um tipo de dado existente, como *int*, *float*, *char*, ...
- ID\_VAR é um identificador que você escolhe para sua variável que está dentro da estrutura.

Você pode escolher entre o Modo 1 e/ou o Modo 2, o que muda é a forma de você utilizar seu novo tipo de dado dentro de seu programa. Para ser mais claro, ao utilizar o Modo 1, você deve declarar as variáveis desse novo tipo assim:

```
int main() {  
    // outras variáveis...  
    struct ID_ESTRUTURA nome_da_variavel;  
    // outras variáveis  
    // resto do código  
}
```

E ao utilizar o Modo 2, você deve declarar as variáveis desse novo tipo assim:

```
int main() {  
    // outras variáveis...  
    ID_ESTRUTURA nome_da_variavel;  
    // outras variáveis  
    // resto do código  
}
```

Para que seja visível para todas as funções de seu programa, escreva sua estrutura global, ou seja, fora do escopo de qualquer função, incluindo a função *main*.

### **Exemplo:**

Suponha que seja necessário criar uma lista de carros, cada um com seu nome, ano, modelo e valor. Nas próximas páginas, é apresentado um exemplo (supondo que sejam 10 carros) sem utilizar estrutura, outro utilizando o Modo 1 e outro utilizando o Modo 2 da estrutura (*struct*).

### *Não utilizando a estrutura:*

```
#include <stdio.h>

int main() {
    // temos que declarar 4 variáveis: nome, ano, modelo, valor
    char nome[10][20];
    int ano[10], modelo[10];
    float valor[10];
    //para iteração
    int i;

    for (i=0;i<10;i++) { //Leitura
        printf("Carro %d:\n", i);
        printf("  Digite o nome do carro: ");
        scanf("%s", nome[i]);
        printf("  Digite o ano do carro: ");
        scanf("%d", &ano[i]);
        printf("  Digite o modelo do carro: ");
        scanf("%d", &modelo[i]);
        printf("  Digite o valor do carro: ");
        scanf("%f", &valor[i]);
    }
    for (i=0;i<10;i++) { //Escrita
        printf("Carro %d:\n", i);
        printf("  Nome: %s\n", nome[i]);
        printf("  Ano: %d\n", ano[i]);
        printf("  Modelo: %d\n", modelo[i]);
        printf("  Valor: R$ %.2f\n", valor[i]);
    }
}
```

## *Utilizando a estrutura escrita do Modo 1:*

```
#include <stdio.h>

// criamos um novo tipo de dado que já possuirá o nome, ano, modelo e valor:
struct tipoCarro {
    char nome[20];
    int ano, modelo;
    float valor;
};

int main() {
    // criamos só uma variável com nosso novo tipo de dado:
    struct tipoCarro carro[10];
    //para iteração
    int i;

    for (i=0;i<10;i++) { // Leitura
        // e acessamos os membros de uma só variável, a carro[i]
        printf("Carro %d:\n", i);
        printf("  Digite o nome do carro: ");
        scanf("%s",  carro[i].nome); //acessamos seus membros com ponto "."
        printf("  Digite o ano do carro: ");
        scanf("%d", &carro[i].ano);
        printf("  Digite o modelo do carro: ");
        scanf("%d", &carro[i].modelo);
        printf("  Digite o valor do carro: ");
        scanf("%f", &carro[i].valor);
    }
    for (i=0;i<10;i++) { // Escrita
        printf("Carro %d:\n", i);
        printf("  Nome: %s\n",      carro[i].nome);
        printf("  Ano: %d\n",      carro[i].ano);
        printf("  Modelo: %d\n",   carro[i].modelo);
        printf("  Valor: R$ %.2f\n", carro[i].valor);
    }
}
```

## Utilizando a estrutura escrita do Modo 2:

```
#include <stdio.h>

// criamos um novo tipo de dado que já possuirá o nome, ano, modelo e valor:
typedef struct {
    char nome[20];
    int ano, modelo;
    float valor;
} tipoCarro;

int main() {
    // criamos só uma variável com nosso novo tipo de dado:
    tipoCarro carro[10];
    //para iteração
    int i;

    for (i=0;i<10;i++) { //Leitura
        // e acessamos os membros de uma só variável, a carro[i]
        printf("Carro %d:\n", i);
        printf("  Digite o nome do carro: ");
        scanf("%s",  carro[i].nome); //acessamos seus membros com ponto "."
        printf("  Digite o ano do carro: ");
        scanf("%d", &carro[i].ano);
        printf("  Digite o modelo do carro: ");
        scanf("%d", &carro[i].modelo);
        printf("  Digite o valor do carro: ");
        scanf("%f", &carro[i].valor);
    }

    for (i=0;i<10;i++) { //Escrita
        printf("Carro %d:\n", i);
        printf("  Nome: %s\n",      carro[i].nome);
        printf("  Ano: %d\n",      carro[i].ano);
        printf("  Modelo: %d\n",   carro[i].modelo);
        printf("  Valor: R$ %.2f\n", carro[i].valor);
    }
}
```