

Estrutura de dados heterogêneas em linguagem C: Struct

Definição: uma estrutura de dados heterogênea é uma coleção de variáveis que podem apresentar diversos tipos.

Em linguagem C, tal estrutura é declarada utilizando a palavra-chave struct.

Na literatura, esse tipo de construção é chamado de registro e serve para modelar objetos baseado nos seus atributos.

Sintaxe:

```
struct nome_do_registro {  
    lista de atributos;  
};
```

Exemplo: Registro aluno:

```
struct aluno{  
    char nome[100];  
    char matricula[12];  
    float CR;  
};
```

Declaração: A definição apresentada no exemplo serve como molde para elementos do tipo aluno.

Para criar um elemento (variável) com esse tipo é necessário declarar.

A declaração de uma variável do tipo registro segue a sintaxe:

```
struct nome_do_registro nome_variável;
```

Exemplo:

```
struct aluno A1;  
struct aluno turma[50];
```

Acesso aos campos/atributos:

Para acessar os campos de uma struct basta utilizar o operador "." (ponto) ao lado da variável.

Ex.: Considere o registro aluno:

```
struct aluno A1;  
strcpy(A1.nome, "Robervaldo");  
strcpy(A1.matricula, "20129651990");  
A1.CR = 7.9;
```

Criação de tipos (typedef)

Ao definir uma struct, pode-se atribuir um nome a ela. No exemplo apresentado são criadas variáveis do tipo aluno.

É possível também atribuir a esse tipo um nome mais simplificado utilizando a palavra-chave typedef.

Por exemplo:

```
struct aluno{  
    char nome[100];  
    char matricula[12];  
    float CR;  
};  
typedef struct aluno aluno_t;
```

Após o comando typedef do último exemplo, podemos declarar variáveis do tipo struct aluno utilizando o tipo aluno_t.

Exemplo:

```
aluno_t A1;  
aluno_t turma[50];
```