



ENSINO MÉDIO INTEGRADO - INFORMÁTICA

Disciplina de Linguagem de Programação

Aula 19: Estrutura e Definição de Tipo

Gil Eduardo de Andrade

Conceitos Preliminares

<https://docs.microsoft.com/pt-br/cpp/c-language/?view=msvc-170>

Introdução

Uma estrutura, em C, é uma coleção de variáveis referenciada por apenas um nome. As estruturas fornecem uma maneira conveniente para agrupar informações que possuem algum tipo de relação entre si. As informações agrupadas nas estruturas são representadas (armazenadas) em variáveis. Estas variáveis são denominadas membros da estrutura;

Sintaxe / Declaração

```
struct aluno {  
    char nome[50];  
    char curso[50];  
    int turma;  
};
```

- **struct**: palavra reservada para declarar/definir uma nova estrutura ou modelo de dados, especificando um identificador (“aluno”) para ela;
- **char / char / int**: define os membros: **nome**, **curso** e **turma** da nova estrutura “aluno”.

Estruturas ⇒ Variáveis

Ao definirmos uma estrutura temos um modelo do qual é possível gerar variáveis. Ao declararmos uma variável do tipo estrutura, o compilador C aloca memória o suficiente para acomodar todos os seus membros. Considerando que as estruturas são definidas para que seja possível criar um novo modelo de dados, elas são

declaradas globalmente. Isso permite que esse novo modelo de dados seja usado em todo o arquivo fonte C que está sendo codificado.

Sintaxe/Exemplo

```
struct aluno {
    char nome[50];
    char curso[50];
    int turma;
};

int main() {
    struct aluno sala[20];
    return 0;
}
```

- **sala[20]**: variável para o novo modelo de dados (estrutura) criado.

Manipulando os Membros da Estrutura

Para que seja possível manipular (acessar) os membros de uma variável estrutura é utilizado o caractere reservado ponto “.”. Ou seja, considerando o exemplo anterior, para acessar o membro “turma” da variável estrutura “sala” teríamos: “sala.turma”.

Sintaxe/Exemplo

```
struct aluno {
    char nome[50];
    char curso[50];
    int turma;
};

int main() {
```

```
    struct aluno sala[20];  
    scanf("%i", &sala[0].turma);  
    return 0;  
}
```

- **sala[0].turma:** atribui o valor inteiro digitado pelo usuário à primeira posição da variável **sala**, membro **turma**.

Definição de Tipo

As estruturas permitem que novos modelos de dados sejam criados. Estes modelos podem ser definidos como um novo tipo, funcionando de forma análoga aos tipos nativos da linguagem C: **int**, **char**, **float**, **double**, etc.

Para que seja possível definir um novo tipo em C é utilizada a palavra reservada **typedef**. O **typedef** deve ser utilizado juntamente com a declaração do novo modelo de dados (estrutura).

Sintaxe/Exemplo

```
typedef struct {  
    char nome[50];  
    char curso[50];  
    int turma;  
} aluno;  
  
int main() {  
    aluno sala[20];  
    scanf("%i", &sala[0].turma);  
    return 0;  
}
```



Codificação – Linguagem de Programação C

Declarando Estrutura e Manipulando seus Membros

```
#include <stdio.h>

#include <stdio_ext.h>

#include <stdlib.h>

struct aluno{

    char nome[50];

    char curso[50];

    int turma;

};

int main() {

    struct aluno sala[20];

    printf("Digite a turma: ");

    scanf("%i", &sala[0].turma);

    printf("\n> Turma %i", sala[0].turma);

    printf("\n");

    return 0;

}
```

Struct - Criando e Manipulando Estruturas.

(Arquivo-fonte: 19 - Estrutura/manipulando membros.c)

Declarando e Alocando Dinamicamente - Estruturas

```
#include <stdio.h>
#include <stdio_ext.h>
#include <stdlib.h>

struct aluno{

    char nome[50];

    char curso[50];

    int turma;

};

int main() {

    struct aluno *sala = malloc(20 * sizeof(struct aluno));

    printf("Digite a turma: ");

    scanf("%i", &sala[0].turma);

    printf("\n> Turma %i", sala[0].turma);

    printf("\n");

    return 0;

}
```

Struct - Alocando Dinamicamente Estruturas.

(Arquivo-fonte: 19 - Estrutura/alocacao_dinamica.c)

Passando Variável Estrutura para Função

```
#include <stdio.h>
#include <stdio_ext.h>
#include <stdlib.h>

struct aluno{

    char nome[50];

    char curso[50];
```

```
int turma;

};

void recebeEstrutura(struct aluno *sala);

int main() {

    struct aluno *sala = malloc(20 * sizeof(struct aluno));

    printf("Digite a turma: ");

    scanf("%i", &sala[0].turma);

    recebeEstrutura(sala);

    printf("\n");

    return 0;

}

void recebeEstrutura(struct aluno *sala) {

    printf("\n> Turma %i", sala[0].turma);

}
```

Struct - Estrutura como Parâmetro da Função.

(Arquivo-fonte: 19 - Estrutura/passando_estrutura_funcao.c)

Definição de Tipo - A partir da Estrutura

```
#include <stdio.h>

#include <stdio_ext.h>

#include <stdlib.h>

typedef struct {

    char nome[50];

    char curso[50];

    int turma;

}aluno;
```

```
void recebeEstrutura(aluno *sala);

int main() {

    aluno *sala = malloc(20 * sizeof(aluno));

    printf("Digite a turma: ");

    scanf("%i", &sala[0].turma);

    recebeEstrutura(sala);

    printf("\n");

    return 0;

}

void recebeEstrutura(aluno *sala) {

    printf("\n> Turma %i", sala[0].turma);

}
```

Typedef - Definição de um Novo Tipo de Dado

(Arquivo-fonte: 19 - Estrutura/definicao_tipo.c)