

Disciplina: Estrutura de Dados

Aula 03: Structs

Professor: Wolney Henrique Queiroz Freitas

Roteiro de Aula

- Structs
 - Conceito;
 - Declarando e Trabalhando com Structs;
 - Array de Struct;
 - Aninhamento de Estruturas.

Structs: Estruturas em C

- Uma estrutura pode ser vista como um conjunto de variáveis sob um mesmo nome;
- A idéia básica por trás da estrutura é criar apenas um tipo de dado que contenha vários campos, que nada mais são do que outras variáveis;
- Em outras palavras, estamos criando uma variável que contém dentro de si outras variáveis.

Variáveis Separadas

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

```
main(){
```

```
    int matricula;
```

```
    char nome[80];
```

```
    matricula = 101220;
```

```
    gets(nome);
```

```
    fflush(stdin);
```

```
    printf("Matricula: %d, Nome: %s", matricula, nome);
```

```
}
```

Structs: Variáveis compondo estrutura

```
#include <stdio.h>

struct aluno {
    int matricula;
    char nome[80];
};

main(){
    struct aluno a;

    a.matricula = 101220;
    gets(a.nome);
    fflush(stdin);

    printf("Matricula: %d, Nome: %s", a.matricula, a.nome);
}
```

Variáveis Separadas

```
#include <stdio.h>

main(){
    int matricula, matricula2;
    char nome[80], nome2[80];

    scanf("%d", &matricula);
    fflush(stdin);
    gets(nome);
    fflush(stdin);

    scanf("%d", &matricula2);
    fflush(stdin);
    gets(nome2);
    fflush(stdin);

    printf("Matricula: %d, Nome: %s", matricula, nome);
    printf("\nMatricula2: %d, Nome2: %s", matricula2, nome2);
}
```

Structs: Variáveis compondo estrutura

```
struct aluno {
    int matricula;
    char nome[80];
};

main(){
    struct aluno a1, a2;

    scanf("%d", &a1.matricula);
    fflush(stdin);
    gets(a1.nome);
    fflush(stdin);

    scanf("%d", &a2.matricula);
    fflush(stdin);
    gets(a2.nome);
    fflush(stdin);

    printf("Aluno1 -> Matricula: %d, Nome: %s", a1.matricula, a1.nome);
    printf("\nAluno2 -> Matricula: %d, Nome: %s", a2.matricula, a2.nome);
}
```

Iniciando Structs com Valores

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

```
struct aluno {  
    int matricula;  
    char nome[80];  
};
```

```
main(){  
    struct aluno a = {101220, "Pedro"};  
  
    printf("Matricula: %d, Nome: %s", a.matricula, a.nome);  
}
```


Copiando Variáveis Structs

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

```
struct aluno {  
    int matricula;  
    char nome[80];  
};
```

```
main(){  
    struct aluno a = {101220, "Pedro"};  
    struct aluno a2 = a;  
  
    printf("Matricula: %d, Nome: %s", a2.matricula, a2.nome);  
}
```

Vamos Praticar

- Implemente um programa que leia o nome, a idade, dia, mês e ano de nascimento de uma pessoa e armazene os dados em uma estrutura;

Array de Structs

```
struct aluno {  
    int matricula;  
    char nome[80];  
};  
  
main(){  
    struct aluno alunos[5];  
    int i;  
  
    for(i = 0; i < 5; i++){  
        scanf("%d", &alunos[i].matricula);  
        fflush(stdin);  
        gets(alunos[i].nome);  
        fflush(stdin);  
    }  
  
    for(i = 0; i < 5; i++){  
        printf("\nMatricula: %d, Nome: %s", alunos[i].matricula, alunos[i].nome);  
    }  
}
```

Structs Aninhados

```
#include <stdio.h>
#include <string.h>

struct curso{
    int codigo;
    char nome[200];
};

struct aluno {
    int matricula;
    char nome[80];
    struct curso curso;
};
```

Structs Aninhados

```
main(){  
    struct curso curso = {1001, "Engenharia Civil"};  
    struct aluno aluno;  
  
    aluno.matricula = 101350;  
    strcpy(aluno.nome, "Marcela");  
    aluno.curso = curso;  
    printf("Aluno: %s, Matricula: %d, Curso: %d - %s",  
        aluno.nome,  
        aluno.matricula,  
        aluno.curso.codigo,  
        aluno.curso.nome);  
}
```

Vamos Praticar

- Faça um programa que leia um array de carros onde cada carro possui os seguintes atributos: marca, modelo, ano e cor.

Vamos Praticar

- Escreva um trecho de código para fazer a criação dos novos tipos de dados conforme solicitado abaixo:
 - Horário: composto de hora, minutos e segundos;
 - Data: composto de dia, mês e ano;
 - Compromisso: composto de uma data, horário e texto que descreve o compromisso.
- Faça um programa para ler e salvar uma lista de compromissos (no máximo 10) de um usuário.



Faci facid FACIMP F&V fmf Presidência
Martha Falcão ISL UNIFAVIP UNI
METROCAMP RUY
BARBOSA | AREA1 UniF&V UniFanor