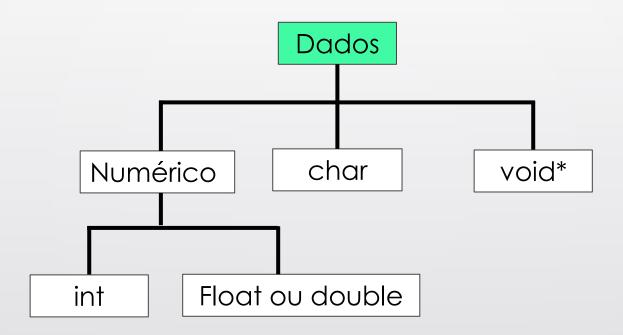
# Registros - structs

Algoritmos e Programação 2 – Ciências da Computação - UFJ

Professora: Ana Paula Freitas Vilela Boaventura

### Das últimas aulas...



Tipos de dados em C



### Motivação

Registro

Campos ou membros do registro

#### Ficha 01

Nome:

Endereço Residencial:

CEP: Cidade:

Telefone fixo: Telefone celular:

E-mail:

Endereço de rede social

O que podemos fazer?

Declarar uma variável primitiva referente a cada um desses campos;

Registro – struct \*

Agrupamento de valores de diferentes, mas de origem comum, numa única estrutura

# Registro – struct

Programadores podem gerar novos tipos de dados, não se limitando ao uso dos tipos de dados primitivos – Registro ou estrutura (struct);

# Definição - struct

Estrutura de dados Homogênea: Conjunto de variáveis do mesmo tipo, como vetores e matrizes; Estrutura de dados Heterogênea: Conjunto de variáveis de tipos diferentes, como os registros (structs);

# Definição - struct

Serve para armazenar informações de um mesmo tipo.



Passo 1: Criar a estrutura

Passo 2: Definir uma variável do tipo que

foi criado no passo 1

### Passo 1: Criar a estrutura

Ficha Matrícula		
Nome	Matrícula	
Ano de ingresso		
Curso		
Média	Índice de prioridade	

```
struct ficha
{ int ano_ingresso, matricula;
    char nome [50], curso [30];
    float media,
    indice_prioridade;
};
```

Passo 2: Definir uma variável do tipo que foi criado no passo 1

Ficha Matrícula		
Nome	Matrícula	
Ano de ingresso		
Curso		
Média	Índice de prioridade	

```
struct ficha
{    int ano_ingresso, matricula;
    char nome [50], curso [30];
    float media,
    indice_prioridade;
}
```

x, y z são variáveis do tipo struct ficha

```
Passo 2: Definir uma variável do tipo que foi criado no passo 1

void main()

{struct ficha a,b; //declaração das variáveis a e b do tipo struct ficha

typedef struct ficha matricula; //redefinição do tipo struct ficha como matricula
}
```

# Acessando campos do registro

### Observação 1

Cada campo funciona como uma variável primitiva, que estão agrupados no registro

# Acessando campos do registro

### Observação 2

```
Acesso aos membros de um registro
String
Para manipular strings (copiar, concatenar, etc)
Biblioteca <string.h>
Copiar strings: copia str2 em str1
strcpy(x.nome,"Jose");
Escrever strings: %s
printf("%s", x.nome);
```

# Carga automática dos campos do registro

```
Sintaxe:

struct nome_estrutura nome_variavel={valor, valor2, ...,
valor n};

Colocamos entre chaves os
valores dos membros da
estrutura, pela ordem em que
foram escritos na sua
definição
```

Variável inicializada no momento da declaração

### Carga inicial da struct

```
struct aluno
{ int matricula, mes, ano, peso;
  char nome[50],instrutor[50];
};
struct aluno x = \{123, 2,2014,85, "Jaqueline", "Maria"\};
printf("Matrícula:%d!\nMes:%d!\nAno:%d!\nNome:%s!\
nInstrutor:%s!\nPeso:%d!\n", x.matricula, x.mes,x.ano, x.nome,
x.instrutor, x.peso);
```

```
struct person
{
Seja uma estrutura char father [50], mother[50];
}John, Mary;
```

Como se diz em inglês? O pai do Jhon? A mãe de Mary? Se diz: Jhon's father. Mary's mother.

```
struct ficha
{
    int ano_ingresso,
    matricula;
    char nome [50], curso
    [30];
    float media,
    indice_prioridade;
}x,y,z;
```

# Acessando os campos do registro

```
x.ano_ingresso = 2020;
x.matricula = 202771;
x.nome = "Maria da Silva";
y.curso = "Biologia";
```

Nome da variável ponto (.) nome do campo

Recapitulando...

Passo 1: Criar a estrutura

```
int ano_ingresso, matricula;
char nome [50], curso [30];
float media, indice_prioridade;
```

Passo 2: Definir a variável

struct ficha a;

struct ficha

**}**;

Passo 3: Acessar o campo da variável a do tipo struct ficha

```
a.ano_ingresso = 2022;
strcpy (a.nome, "José da Silva");
```



# Vetor de structs

# O que muda?

Passo 1: Criar a estrutura

```
struct ficha
{
    int ano_ingresso, matricula;
    char nome [50], curso [30];
    float media, indice_prioridade;
};
```

Passo 2: Definir o vetor v

struct ficha v[10];

Passo 3: Acessar o elemento do vetor v do tipo struct ficha

```
v[0].ano_ingresso = 2022;
strcpy (v[i].nome, "José da Silva");
```

# Carga automática

```
struct ficha
{
    int ano_ingresso, matricula;
    char nome [50], curso [30];
    float media, indice_prioridade;
};
struct ficha vetor [] = { {valor1, valor2, ..., valor n}, {valor1, valor2, ..., valor n},;
}
```

### Mão na massa

 Usando uma struct matricula com os campos: número de matrícula, nome e 2 notas, crie um vetor que leia 3 vetores do tipo struct matricula. Ao final, mostre a média de cada aluno.

```
#include <stdio.h>
                                         for(i=0; i<3; i++){
                                              printf("\nDigite o nome do %d aluno\n",i+1);
#include <stdlib.h>
                                             scanf("%[^\n]s",t[i].nome);
#include <locale.h>
//setlocale(LC_ALL, "Portuguese");
                                             fflush(stdin);
//setlocale(LC ALL,'"');
                                              printf("\n Matrícula do %d aluno\n",i+1);
                                             scanf("%d",&t[i].matricula);
                                             for (i=0; i<3; i++) {
int main()
 setlocale(LC_ALL, "Portuguese");
                                                printf("\n%d nota do %d aluno\n",j+1,i+1);
                                                scanf("%f",&t[i].nota[j]);
  //definição da estrutura
  struct turma
                                             fflush(stdin);
     char nome [50];
    int matricula:
                                           for(i=0; i<3; i++) {
     float nota[3];
                                              printf("\n%d\t%s",t[i].matricula,t[i].nome);
                                             for (i=0; i<3; i++)
  //declaração das variáveis
                                                printf("\t%0.1f",t[i].nota[i]);
  struct turma t[3];
  float media[3];
                                              printf("\tmédia: %f\n",(t[i].nota[0]+t[i].nota[1]
  int i,j;
                                               +t[i].nota[2])/3);
                                        return 0;
```

## Structs Aninhadas

```
struct s_pessoa
struct pessoa
      int dia, ano;
                                   char nome [50];
                                   int salario, idade;
      char mes[30];
                                   struct pessoa x;
};
                             };
s_pessoa a = {"Maria", 22, 3500, {11, "abril", 2024} };
printf ("nome:= %s, idade = %d, salario = %d, dia = %d, mês
= %s, ano = %d", a.nome, a.idade, a.salario, a.x.dia,
a.x.mes, a.x.ano);
```

### Assistam aos vídeos

- https://youtu.be/tbvo 4QFyzqQ
- https://youtu.be/34\_5n\_NkDYU

### Assistam aos vídeos

- https://www.youtube. com/watch?v=MatsU Ce5uZw&list=PL8iN9F Q7\_jt4DJbeQqv--jpTy-2gTA3Cp&index=35
- https://www.youtube. com/watch?v=CAnQ 6i8OwJA&list=PL8iN9F Q7\_jt4DJbeQqv--jpTy-2gTA3Cp&index=36