

Linguagem de Programação I

UNIESP
Dr^a Alana Moraes



Tipos de Dados

- Existem dois tipos principais de dados:
 - Tipos simples (int, float, double, etc)
 - Tipos estruturados
- Em geral definem uma coleção de valores simples, ou um agregado de valores de tipos diferentes.
 - Ao contrário dos vetores e matrizes, que trabalham com o mesmo tipo de dados.



Tipos de Dados

- Para resolver um problema é necessário escolher uma abstração da realidade, em geral mediante a definição de um conjunto de dados que representa a situação real.
- Diversas técnicas podem ser utilizadas:
 - **Estruturas**
 - Pilhas
 - Filas
 - Listas encadeadas
 - Árvores



• Tipos Abstratos de Dados

- Dados abstratos são utilizados extensivamente como base para o projeto de algoritmos.
- A representação do modelo matemático por trás do tipo abstrato de dados é realizada mediante uma estrutura de dados.
- As estruturas podem conter elementos com qualquer tipo de dados válidos em C (tipos básicos, vetores, strings, ponteiros ou mesmo outras estruturas).



Struct

- As estruturas possibilitam trabalhar com dados heterogêneos, ou seja, dados de vários tipos em uma mesma tabela.
- Em C, isso é conseguido com o comando **struct**, que possibilita a criação de uma ficha virtual para a manipulação de dados.
- As componentes armazenadas dentro de uma estrutura são vulgarmente denominadas **campos** ou **membros** da estrutura.



. Struct

- Visa facilitar manipulação de um conjunto de dados logicamente relacionados, mas de diferentes tipos.
- Para se utilizar uma variável heterogênea é preciso:
 - Passo 1: definir a estrutura
 - Passo 2: declarar as variáveis que irão formar esta estrutura



Sintaxe – definição da estrutura

// Criar uma STRUCT

```
typedef struct
{
    int idade;    //define o campo Idade
    char nome[30]; //define o campo Altura
} pessoa;
```



Struct – declaração de variáveis

```
int main()  
{  
    pessoa p; //esse é o meu struct  
    p.idade = 20;  
    strcpy(p.nome, "Alana Moraes");  
  
    printf("Idade: %d \n", p.idade);  
    printf("Nome: %s \n", p.nome);  
}
```



Vetores de Struct

- Preciso agora armazenar uma lista de 10 pessoas.
- Posso fazer isso com Struct?
 - SIMMMMMM!!



Mesma coisa

```
int main()  
{  
    pessoa p[10]; //vetor de 10 pessoas.  
    p[0].idade = 20;  
    strcpy(p[0].nome, "Alana Morais");  
}
```



Exercício

- Crie um struct para descrever um professor (matricula, nome) em C.
- Declare uma variável global para armazenar as informações de 5 professores.
- O usuário quem deve passar tais informações.
- Ao final imprima essa lista com todas as informações.



Agora é sua vez

- Implemente uma estrutura que sirva para guardar informações de uma conta bancária: agência, conta, titular e saldo. Defina os tipos destes dados de acordo com sua necessidade.
- Ao final, crie uma lista de 5 contas e imprima os valores das contas.



Exercício 2

- Escrever um programa que cadastre o nome, a matrícula e duas notas de vários alunos. Em seguida imprima a matrícula, o nome e a média de cada um deles.



Dúvidas?

alanamm.prof@gmail.com

