

Aula 000 - Tipos Abstratos de Dados (TED)

TIPOS DE DADOS

Um TIPO DE DADO define o CONJUNTO DE VALORES que uma variável pode armazenar.

Exemplo:

Em C	Em Java
Int → números inteiros	int → números inteiros
float → números reais (ex: 3.14, -0.01, 0.0)	float → números reais (ex: 3.14)
char → caracteres (ex: 'a', 'Z', '!')	char → caracteres (ex: 'a', 'Z')
bool → verdadeiro (1) ou falso (0) (simulado com int)	boolean → true (verdadeiro) ou false (falso)

Nota: Tipos primitivos são os “blocos básicos” para construção de estruturas maiores.

ESTRUTURAS DE DADOS

Estruturas de dados são formas de organizar e relacionar vários dados para representar informações com significado lógico.

Exemplo prático em C: Usamos 'struct' para agrupar dados relacionados	Exemplo prático em Java: Usamos 'classes' para agrupar dados relacionados
// representa um aluno com nome, idade e nota struct Aluno { char nome[50]; int idade; float nota; };	// uma classe `Aluno` com nome, idade e nota. static class Aluno { String nome; int idade; float nota; }

Nota: Esses exemplos mostram como diferentes tipos de dados são agrupados para representar uma entidade lógica, como um aluno.

TIPOS ABSTRATOS DE DADOS (TAD)

Um **TAD (Tipo Abstrato de Dado)** é um modelo que define:

- a **estrutura** de dados
- as **operações** permitidas sobre ela

O foco de um TAD é **como se usa a estrutura**, e **não em como ela é implementada**.

Operações típicas de um TAD:

- `criar()` → inicializa a estrutura
- `inserir()` → adiciona elemento
- `remover()` → exclui elemento
- `acessar()` → consulta elementos (sem acesso direto aos dados)

Vantagens do uso de TADs:

- **Encapsulamento e segurança dos dados** (Protege os dados internos)
- **Modularidade** (Organiza o código em partes reutilizáveis)
- **Reutilização** (Facilita o uso da estrutura em diferentes programas)

Analogia com arquivos em C:

`FILE *f;` // ponteiro para um tipo abstrato de dados

// Não sabemos como `FILE` é implementado internamente, interagimos com ele por meio de funções, como: `fopen()`, `fclose()`, `fputc()`, `fgetc()`, `fprintf()`, `fscanf()`, etc.

Em Java, os TADs são implementados como classes que:

- Possuem **atributos privados** (para proteger os dados)
- Fornecem **métodos públicos** (para controlar o acesso)

Exemplo:

```
class Aluno {  
    private String nome;  
    private float nota;  
  
    public void setNome(String n) { nome = n; }  
    public float getNota() { return nota; }  
}
```

Nota: Aqui não temos acesso direto aos atributos, só conseguimos acessá-los por meio dos métodos públicos.

Outro exemplo em Java: imagine a classe ‘Scanner’:

```
Scanner sc = new Scanner(System.in);  
int idade = sc.nextInt();
```

Você não sabe como Scanner funciona internamente. Apenas usa os métodos públicos como: `nextInt()`, `nextLine()`, `hasNext()`, etc.

- Isso É um TAD!