

# Tipos Abstratos de Dados (TAD) ou Registros

Prof. Dr. Marcelo Fernando Rauber

No desenvolvimento de aplicações, é comum o programador querer agrupar informações a respeito de um mesmo objeto, como uma pessoa, produto, PC, etc. Estas informações, muitas vezes são de tipos distintos. Para este fim os Tipos Abstratos de Dados (TAD) ou Registros podem ser empregados.

O que faremos é subdividir uma mesma variável, possibilitando assim armazenar diferentes tipos de dados primitivos nela. Cada um destes sub-dados contidos na variável é chamado de campo ou atributo. Estes campos podem também representar outro TAD. Por isso, são chamados de variáveis compostas heterogêneas.

Representação de uma variável TAD na memória:

umFuncionário	Nome	
	Função	
	Salário	

Em Java, os TADs são definidos através por classes. Como o Java é orientado a objetos, as classes são também usadas como base para orientação de objetos, mas não nos aprofundaremos nesse momento.

Para podermos utilizar um TAD, devemos seguir 3 etapas:

1. Definir a estrutura do TAD
2. Declarar uma ou mais variáveis do TAD definido na etapa anterior
3. Inicializar cada uma das variáveis de TAD;

**1. Definindo a estrutura do TAD.** A definição dos campos varia conforme o contexto do problema.

```
class Tfuncionario {  
    String nome;  
    String funcao;  
    double salario;  
}
```

**2. Declarar uma ou mais variáveis do TAD** definido na etapa anterior

```
Tfuncionario umFuncionario;
```

**3. Inicializar** cada uma das variáveis de TAD;

```
umFuncionario = new Tfuncionario();
```

//ou podemos fazer as etapas 2 e 3 juntas:

```
Tfuncionario outroFuncionario = new Tfuncionario();
```

Uma vez realizadas estas 3 etapas, estamos prontos para utilizar o TAD.

## Manipulando variáveis de TAD

Uma vez cumpridas as 3 etapas anteriores, estamos aptos a armazenar e manipular dados em variáveis de TAD. Para termos acesso a cada campo que compõe o TAD devemos colocar o nome da variável de TAD (repare que não é o nome da classe!!!), seguido de um ponto final e finalmente o nome do campo que desejamos manipular. Cada um dos campos ou atributos deve ser manipulado individualmente.

Exemplo: Preenchendo um campo com um valor informado pelo teclado:

```
System.out.println("Digite o Nome de um funcionário");
umFuncionario.nome = teclado.nextLine();
```

Exemplo 2: Da mesma forma, se quisermos apresentar os campos de um TAD na tela, devemos fazer isso individualmente, campo a campo:

```
System.out.println(umFuncionario.nome);
System.out.println(umFuncionario.funcao);
System.out.println(umFuncionario.salario);
```

Exemplo 3: Um programa completo, com TAD e vetores.

```
1 package javaapplicationexemplotad;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author Prof_Marcelo_Rauber
8  */
9
10 //Definição da estrutura de um TAD para armazenar produtos.
11 class Tproduto {
12     String nome;
13     String descricao;
14     double valor;
15 }
16
17 public class JavaApplicationExemploTAD {
18
19     public static void main(String[] args) {
20         Scanner teclado = new Scanner(System.in);
21
22         //Declarando uma variável de TAD
23         Tproduto umproduto = new Tproduto();
```

```

24 //Preenchendo uma variável de TAD
25 System.out.println("Digite Nome, descri. e valor de um produto");
26 umproduto.nome = teclado.next();
27 umproduto.descricao = teclado.next();
28 umproduto.valor = teclado.nextDouble();
29 //Mostrando o conteúdo de uma variável de TAD
30 System.out.println(umproduto.nome);
31 System.out.println(umproduto.descricao);
32 System.out.println(umproduto.valor);
33
34 //Declarando um Vetor de TAD
35 Tproduto vetProdutos[] = new Tproduto[5];
36 //Preenchendo um Vetor de TAD
37 for (int i = 0; i < 5; i++) {
38     /*A primeira vez que for utilizar cada posição do vetor
39     ou matriz, é necessário inicializá-la. Se não for feito,
40     em tempo de execução do programa, aparecerá
41     a mensagem: java.lang.NullPointerException */
42     vetProdutos[i] = new Tproduto();
43     System.out.println("Digite Nome,descri. e valor de um produto");
44     vetProdutos[i].nome = teclado.next();
45     vetProdutos[i].descricao = teclado.next();
46     vetProdutos[i].valor = teclado.nextDouble();
47 }
48
49 //Mostrando o conteúdo de um vetor de TAD através de uma sub-rotina
50 mostra(vetProdutos);
51 } //fim do programa principal
52
53 //Sub-rotina que mostra o conteúdo de um vetor de produtos
54 static void mostra(Tproduto vet[]) {
55     for (int i = 0; i < vet.length; i++) {
56         System.out.println("dados na posição " + i + "do vet de TAD:");
57         System.out.println("Nome: " + vet[i].nome);
58         System.out.println("Descrição: " + vet[i].descricao);
59         System.out.println("Valor: " + vet[i].valor);
60     }
61 }
62 }
63

```

## **Exercícios**

1 – Utilizando TAD, armazene nome e média (de cada aluno) de 35 alunos na disciplina de Estrutura de Dados. Depois de todos os dados informados, apresente uma listagem de alunos em exame.

2 – Utilizando TAD para armazenar dados de livros (assuma apenas um autor), e, crie uma sub-rotina que recebe esse TAD. Essa sub-rotina deve retornar uma string com a referência bibliográfica no formato ABNT.

3 – Utilizando TAD, crie um programa com o seguinte menu de opções, onde cada opção é resolvida por uma sub-rotina.

Menu:

- 1 – Cadastrar cliente (CPF, nome, endereço e telefone)
- 2 – Apresentar listagem de todos os clientes
- 3 – Procurar clientes pelo nome e apresentar seus dados
- 4 – Apresentar clientes com telefone no código de área 47
- 5 - Sair