

### Objetivo do Tema

Compreender as diferenças entre tipos d dos pré-existentes e tipos abstratos de dados.



### **TIPOS DE DADOS**

- Tipos Primitivos:
  - um tipo estabelece a natureza (característica) do dado que é manipulado por um algoritmo;
  - define o conjunto de valores que uma variável, constante ou função podem receber;
  - tipos de dados básicos (primitivos):
    - inteiro: 13, -6, 7830, -295;
    - real: 23.8, 3.6752, 8.910, 3738.72, 32.0;
    - caractere: "UNICARIOCA", "Computação", "111 + 222 = 333";
    - lógico: FALSO, VERDADEIRO.



### **TIPOS DE DADOS**

- · Tipos Estruturados:
  - uma estrutura de dados é um modo particular de armazenamento e de organização de dados para que possam ser usados
  - os tipos estruturados s\u00e3o exemplos de estruturas de dados:
  - tipos de estruturas de dados:
    - homogêneas (vetores e matrizes): conjuntos de dados formados pelo mesmo tipo de dado básico;
    - · heterogêneas (registros): conjuntos de dados formados por tipos de dados básicos diferentes (campos do registro).



### **TIPOS DE DADOS**

- Tipos Estruturados Homogêneos:
  - um vetor é uma estrutura que armazena os dados em uma linha e em várias colunas, ou seja, é unidimensional;
  - exemplo:

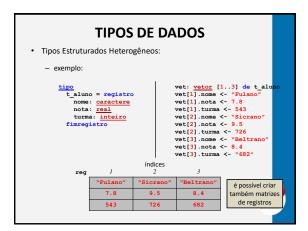
```
vet: vetor [1..5] de real
vet[1] <- 1.2
vet[3] <- vet[1] + vet[2]
vet[4] <- vet[3] * 2
vet[5] <- vet[4] - 3.5
vet[1] <- vet[1] + 2.7
```

### **TIPOS DE DADOS**

- Tipos Estruturados Homogêneos:
  - uma matriz é uma estrutura que armazena os dados em várias linhas e em várias colunas, ou seja, é bidimensional;
  - exemplo:

```
mat: vetor [1..4, 1..3] de real
                       índices
- 2.7
                        2.0
```

# TIPOS DE DADOS • Tipos Estruturados Heterogêneos: - um registro é uma estrutura formada por um conjunto de variáveis de tipos diferentes ou, eventualmente, iguais; - exemplo: tipo t\_func = registro nome: caractere salario: real setor: caractere PRE: logico fimregistro reg: t\_func reg.nome <- "Fulano" reg.setor <- "Yeulano" reg.setor <- "Yeulano"



### TIPOS DE DADOS Tipos Abstratos de Dados: tipos estruturados, em geral, definem uma coleção de valores de um tipo primitivo (vetores e matrizes), ou um agregado de valores de tipos primitivos diferentes (registros); um tipo abstrato de dados (TAD) é uma coleção bem definida de dados a serem armazenados e um grupo de operações que podem ser aplicadas na manipulação desses dados: exemplo: o conjunto dos números inteiros acompanhado das operações de adição, subtração e multiplicação.

### **TIPOS DE DADOS**

- Tipos Abstratos de Dados:
  - características:
    - TAD's são generalizações de tipos primitivos, assim como funções são generalizações de operações primitivas (+, -, \*, /);
    - assim como uma função encapsula partes de um algoritmo, o TAD pode encapsular tipos de dados;
    - em um TAD, a palavra "abstrato" quer dizer "não importa a forma de implementação", ou seja, basta saber a finalidade do tipo e de suas operações;
    - os TAD'S são uma técnica de programação importante, pois facilitam a manutenção e o reuso de código.



### **TIPOS DE DADOS** • Tipos Abstratos de Dados: #include <arquivo\_TAD>.h - implementação: /\* implementação das funções (operações) do TAD \*/ /\* protótipos das funções <arguivo TAD>.c e declarações dos novos tipos e constantes \*/ #include <arquivo\_TAD>.h <arquivo\_TAD>.h programa principal \*/ main.c - razões para incluir <arquivo\_TAD>.h em <arquivo\_TAD>.c: • podem existir definições na interface que são necessárias na implementação; • garantir que as funções implementadas correspondam às funções d

### **TIPOS DE DADOS**

- Tipos Abstratos de Dados:
  - exemplo:
    - criação de um tipo de dado Ponto para representar um ponto no  $\mathbb{R}^2;$
    - operações:
      - **cria**: cria um ponto com coordenadas x e y;
      - libera: libera a memória alocada por um ponto;
      - acessa: retorna as coordenadas de um ponto;
      - atribui: atribui novos valores às coordenadas de um ponto;
      - distancia: calcula a distância entre dois pontos.



### 

### TIPOS DE DADOS • Tipos Abstratos de Dados: - exemplo (arquivo "ponto.c"): Fonto \* cria(float x, float y) { Fonto \*p = (Fonto \*)malloc(sizeof(Fonto)); if (p == NULL) { printf("Mem\242ria insuficiente!\n"); exit(1); } p->x = x; p->y = y; return p; }

```
TIPOS DE DADOS

• Tipos Abstratos de Dados:

- exemplo (arquivo "ponto.c"):

void libera (Ponto *p)

{
    free(p);
}

void acessa (Ponto *p, float *x, float *y)

{
    *x = p->x;
    *y = p->y;
}

void atribui (Ponto *p, float x, float y)

{
    p->x = x;
    p->y = y;
}
```

# TIPOS DE DADOS • Tipos Abstratos de Dados: - exemplo (arquivo "ponto.c"): float distancia (Ponto \*p1, Ponto \*p2) { float dx, dy; dx = p2->x - p1->x; dy = p2->y - p1->y; return sqrt(dx \* dx + dy \* dy); }

### 

## TIPOS DE DADOS • Principais TAD's: - listas; - pilhas; - filas; - árvores; - grafos.

PRÓXIMA AULA: TEMA 2: LISTAS LINEARES COM ALOCAÇÃO ESTÁTICA  CARDOR CARDOR Centro Universitáto	
UNICARIOCA.EDU.BR Millior cantio universifato so mo, selundo o mic	