

UNIDADE 5 – ESTRUTURAS DE DADOS

5.1 VARIÁVEIS COMPOSTAS HOMOGÊNEAS

Assim como na Teoria dos Conjuntos, uma variável pode ser interpretada como um elemento e uma Estrutura de Dados como um conjunto. Quando uma determinada Estrutura de Dados for composta de variáveis com o mesmo tipo primitivo, temos um conjunto homogêneo de dados. Podemos considerar que uma variável composta seja como uma alcatéia, e seus elementos (variáveis) sejam como os lobos (que são da mesma espécie).

As variáveis compostas homogêneas correspondem a posições de memória, identificadas por um mesmo nome, individualizadas por índices e cujo conteúdo é de mesmo tipo. O conjunto de 10 notas dos alunos de uma disciplina pode constituir uma variável composta. A este conjunto associa-se o identificador NOTA que passará a identificar não uma única posição de memória, mas 10. A referência ao conteúdo do n ésimo elemento do conjunto será indicada pela notação $NOTA[n]$, onde n é um número inteiro. Supondo-se que em um dado instante a variável composta NOTA contivesse os seguintes valores:

| | | | | | | | | | | |
|-------------|----|----|----|----|----|----|-----|----|----|----|
| NOTA | 60 | 70 | 90 | 60 | 55 | 91 | 100 | 47 | 74 | 86 |
| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

$NOTA[3]$ está referenciando o 3º elemento do conjunto cujo conteúdo é 90.

5.1.1 VARIÁVEIS COMPOSTAS UNIDIMENSIONAIS (VETORES)

Variáveis compostas unidimensionais são variáveis compostas por um conjunto de dados referenciados por um mesmo nome e que necessitam de somente um índice para que seus elementos sejam endereçados.

Sintaxe da declaração:

```
tipo IDENTIFICADOR = vetor [LI..LF] de <tipo>;
IDENTIFICADOR : lista de variáveis;
```

Onde:

- LI : representa o limite inicial do vetor;
- LF : representa o limite final do vetor;
- <tipo> : representa qualquer um dos tipos básicos ou tipo anteriormente definido;
- lista de variáveis : lista de variáveis (separadas por vírgula) que terão o mesmo tipo denotado por IDENTIFICADOR.

Exemplo:

```
tipo NOTAS = vetor [1..10] de real;
```

```
NOTAS : NOTAS_TURMA_A;
```

O número de elementos num vetor é dado pela seguinte relação:

$$N = LF - LI + 1$$

MANIPULAÇÃO:

Ao imaginar o elevador de um prédio, sabemos que este é capaz de acessar qualquer um de seus andares. Entretanto, não basta saber que andar desejamos atingir se não soubermos o nome do edifício, pois qualquer um possui andares. O que precisamos de antemão é saber o nome do edifício e só então nos preocuparmos para qual daqueles andares queremos ir.

O mesmo acontece com os vetores, visto que são compostos por diversas variáveis e, como podem existir muitos vetores, torna-se necessário determinar qual vetor contém o dado desejado e depois especificar em qual posição este se encontra.

O nome do vetor é determinado por meio do identificador que foi utilizado na definição de variáveis, e a posição, por meio da constante, expressão aritmética ou variável que estiver dentro dos colchetes, também denominada **índice**.

Após isolar um único elemento do vetor, poderemos manipulá-lo através de qualquer operação de entrada, saída ou atribuição.

Exemplo:

```
leia V[5];  
V[5] ← V[5] + 1;  
escreva V[5];
```

Exercícios:

42) Dados 10 valores, armazenar num vetor somente os números pares.

43) Faça um algoritmo que leia um vetor de 100 elementos numéricos inteiros e depois verifique se existem elementos iguais a 30. Se sim, escrever as posições em que estão armazenados. Este algoritmo deve, ao final, imprimir todos os números maiores que 20.

44) Dados 2 vetores A e B compostos por 10 números inteiros e ordenados de maneira crescente, compare-os e crie um vetor C com os números maiores um vetor D com os números menores. Mostre os vetores C e D na tela.

45) Dados dois vetores V1 e V2 com 10 números cada, gerar um vetor V3 com a intercalação de V1 e V2.

46) Dado o vetor LISARB-ANARAP-LEVACSAC gerar um novo vetor sendo que a primeira posição do vetor original será a posição final do novo vetor.

47) Dado um vetor com 20 números inteiros crie um algoritmo para ordená-lo em

ordem crescente.

48) Crie um algoritmo para implementar a pesquisa binária. A pesquisa binária consiste em localizar um elemento num vetor ordenado. Para tanto divide-se o vetor ao meio e verifica-se se o elemento está no segmento à esquerda ou à direita. Novamente divide-se o segmento (esquerdo ou direito) ao meio e assim sucessivamente até encontrar o elemento procurado.

5.1.2 VARIÁVEIS COMPOSTAS MULTIDIMENSIONAIS (MATRIZES)

São as variáveis que necessitam mais de um índice para a individualização de seus elementos. As variáveis multidimensionais podem ser bidimensionais, onde o primeiro elemento representa a linha e o segundo a coluna, ou tridimensionais, onde o elementos representa a página (face).

Sintaxe da declaração:

```
tipo IDENTIFICADOR = matriz [LI1..LF1, LI2..LF2, ..., LIn..LFn]
                        de <tipo>;
IDENTIFICADOR : lista de variáveis;
```

Onde:

- LI1..LF1, LI2..LF2, ..., LIn..LFn : são os limites dos intervalos de variação dos índices da variável, onde cada par de limites está associado a um índice;
- <tipo> : representa qualquer um dos tipos básicos ou tipo anteriormente definido;
- lista de variáveis : lista de variáveis (separadas por vírgula) que terão o mesmo tipo denotado por IDENTIFICADOR.

Exemplo:

```
tipo SALA = matriz [1..4, 1..4] de inteiro;
tipo M = matriz [1..3, 1..5, 1..4] de inteiro;
```

SALA : MSALA;

M : MAT;

O número de dimensões da matriz é igual ao número de intervalos. O número de elementos é igual ao produto do número de elementos de cada dimensão:

$$N = (LF1 - LI1 + 1) * (LF2 - LI2 + 1) * \dots * (LFn - LIn + 1)$$

MANIPULAÇÃO:

Considerando que os andares de um prédio são divididos em apartamentos temos uma estrutura multidimensional. Para localizarmos um indivíduo

neste prédio precisaremos de seu nome, o andar e o número do apartamento. Considerando uma estrutura bidimensional (dois índices: andar e apartamento), o primeiro índice indica a linha e o segundo, a coluna.

Exemplo:

| | | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |

MSALA

O elemento hachurado é referenciado por MSALA[2, 3].

Exercícios:

49) Dada uma matriz (4 x 4) faça um algoritmo para multiplicar os elementos da diagonal principal por um valor qualquer, armazenando estes valores num vetor. Os elementos da matriz deverão ser lidos e exibidos ao final do processo, bem como o vetor resultante.

50) Dada uma matriz (4 x 5) faça um programa para calcular a soma de todos os seus elementos.

51) Crie um algoritmo para transpor uma matriz qualquer.

52) Faça um algoritmo que dada uma matriz tridimensional (3 x 4 x 3) armazene valores numéricos inteiros entre 0 e 20. O algoritmo deverá mostrar quantas vezes um determinado valor (entre 0 e 20) está armazenado repetidamente.

5.2 VARIÁVEIS COMPOSTAS HETEROGÊNEAS

Já sabemos que um conjunto homogêneo de dados (tal como uma alcatéia) é composto por variáveis do mesmo tipo primitivo (lobos); porém, se tivéssemos um conjunto em que os elementos não são do mesmo tipo, teríamos então um conjunto heterogêneo de dados. Exemplificando, poderíamos ter um conjunto de animais quadrúpedes, formado por cães (matilha), camelos (cáfila), búfalos (manada), etc.

5.2.1 REGISTROS

Uma das principais estruturas de dados é o registro. Para exemplificar, imagine uma passagem de ônibus, que é formada por um conjunto de dados logicamente relacionados, porém de tipos diferentes, tais como número da passagem (inteiro), origem e destino (caractere), data (caractere), horário (caractere), poltrona (inteiro), distância (real), que são subdivisões do registro (elementos do conjunto), também chamadas de campos. Logo, um registro é composto por campos que são partes que especificam cada um dos dados.

| | |
|------------------|---------------------|
| Número: 00001 | Para: _____ |
| De: _____ | Horário: ____:____ |
| Data: __/__/____ | Distância: _____ km |
| Poltrona: _____ | |

Sintaxe da declaração:

```

tipo IDENTIFICADOR = registro
                        tipo1 : campo1;
                        tipo2 : campo2;
                        .           .
                        .           .
                        tipon : campon;
                        fim registro;

```

IDENTIFICADOR : lista de variáveis;

Onde:

- IDENTIFICADOR : representa o nome associado ao tipo registro construído;
- tipo1, tipo2, tipon : representam qualquer um dos tipos básicos ou tipo anteriormente definido;
- campo1, campo2, campon : representam nomes associados a cada campo do registro;
- lista de variáveis : lista de variáveis (separadas por vírgula) que terão o mesmo tipo denotado por IDENTIFICADOR.

Exemplo:

```

tipo REGPASSAGEM = registro
                    inteiro : NUMERO, POLTRONA;
                    caractere : ORIGEM, DESTINO,
                                DATA, HORÁRIO;
                    real: DISTANCIA;
                    fim registro;

```

REGPASSAGEM : PASSAGEM;

O exemplo corresponde à definição de um modelo REGPASSAGEM de um registro e à criação de uma área de memória chamada PASSAGEM, capaz de conter sete subdivisões.

MANIPULAÇÃO:

Em determinados momentos podemos precisar de todas as informações contidas no registro (PASSAGEM) ou de apenas algum campo do registro (como freqüentemente, o número da poltrona).

Quando acessamos o registro genericamente, estamos referenciando obrigatoriamente todos os campos por ele envolvidos.

Exemplo:

```
leia (PASSAGEM);
escreva (PASSAGEM);
```

Para utilizar um campo específico do registro, devemos diferenciar esse campo. Para tal utilizamos o caractere “.” (ponto) para estabelecer a separação entre o nome do registro e o nome do campo.

Exemplo:

```
leia (PASSAGEM.POLTRONA);
escreva (PASSAGEM.NUMERO);
```

Exemplo: Utilizando o registro PASSAGEM:

```
início
  {definições anteriores}
  .
  .
  leia (PASSAGEM.NUMERO);
  leia (PASSAGEM.ORIGEM);
  leia (PASSAGEM.DESTINO);
  .
  .
  escreva (PASSAGEM);
fim.
```

5.2.2 REGISTRO DE CONJUNTOS

Os registros vistos até agora possuíam em seus campos apenas informações de tipos primitivos, entretanto, podemos dispor também de campos que são compostos, ou seja, formados por outros tipos construídos.

Digamos que possuímos um registro de estoque de um produto, contendo como um de seus campos um valor numérico que indique baixas por dia do produto, por dia de semana. Temos então, um vetor de seis posições, na qual cada posição corresponde a um dia útil da semana.

| | | | | | | | |
|---------|---|---|---|--------|---|---|--|
| Nome: | | | | | | | |
| Código: | | | | Preço: | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | |
| Baixa: | | | | | | | |

Para declarar o tipo registro anterior, utilizamos um tipo construído vetor; então, precisamos primeiramente declarar tal tipo, e depois o tipo registro; isto é, precisamos declarar todos os conjuntos que serão incluídos no registro antes da sua declaração.

Sintaxe da declaração:

Exemplos:

a)

```

tipo VDIAS = vetor [1..6] de inteiro;
tipo REGPROD = registro
    caractere : NOME;
    inteiro   : CODIGO;
    real      : PRECO;
    VDIAS     : BAIXA;
fim registro;

```

REGPROD : PRODUTO;

b) Modificar o registro de estoque de um produto a fim de que possa conter as baixas de quatro semanas, utilizando um tipo construído matriz.

| | | | | | | |
|---------|-------|--------|---|---|---|-------|
| Nome: | _____ | | | | | |
| Código: | _____ | Preço: | | | | _____ |
| Baixas: | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

```

tipo MATDIAS = matriz [1..4, 1..6] de inteiro;
tipo REGPROD2 = registro
    caractere : NOME;
    inteiro   : CODIGO;
    real      : PRECO;
    MATDIAS   : BAIXA;
fim registro;

```

REGPROD2 : PRODUTO2;

MANIPULAÇÃO:

A manipulação de um registro de conjunto deve obedecer às manipulações próprias de cada estrutura de dados anteriormente definida.

Exemplo:

a) Para acessar o quanto foi vendido do produto no terceiro dia da quarta semana,

teríamos:

```
PRODUTO2.BAIXA [4, 3]
```

- b) Construir o trecho de algoritmo que, usando a definição de PRODUTO2, escreva o nome do produto, o código, o preço e as baixas da segunda semana.

```
.
.
escreva (PRODUTO2.NOME);
escreva (PRODUTO2.CODIGO);
escreva (PRODUTO2.PRECO);

para J = 1 até 6 faça
    escreva (PRODUTO2.BAIXA [2, J]);
fim para;
```

- c) Construa o trecho do algoritmo que totalize por dia da semana todos os dias do mês.

```
.
.
para J = 1 até 6 faça
    AUX ← 0;
    para I = 1 até 4 faça
        AUX ← AUX + PRODUTO2.BAIXA[I, J];
    fim para;
    escreva (J, 'totalizou ', AUX, ' baixas');
fim para;
.
.
```

5.2.3 CONJUNTO DE REGISTROS

Nas estruturas compostas homogêneas (vetores e matrizes) utilizamos tipos de dados primitivos como sendo os elementos dessas estruturas. Agora utilizaremos como componente dessa estrutura não apenas um tipo primitivo, mas sim os tipos construídos, neste caso os registros. Supondo que quiséssemos manter um registro de informações relativas às passagens rodoviárias de todos os lugares de um ônibus, utilizaríamos um registro referente a cada poltrona (PASSAGEM), e para agrupar todos eles utilizaríamos um conjunto desses registros.

Como possuímos 44 lugares num ônibus, numerados seqüencialmente de 1 até 44, podemos, para uni-los, criar um vetor no qual cada posição é um elemento de tipo construído registro (PASSAGEM).

| |
|----|
| ① |
| 2 |
| 3 |
| . |
| . |
| . |
| 44 |

Número: 00001

De: _____ Para: _____

Data: ___/___/_____ Horário: ____:____

Poltrona: _____ Distância: _____ km

Sintaxe da declaração:

Como possuímos um vetor composto por registros, não podemos declarar esse vetor sem antes ter declarado seus elementos; devemos então declarar primeiro o tipo construído registro e depois o vetor.

Exemplo:

a)

```

tipo REGPASSAGEM = registro
    inteiro : NUMERO, POLTRONA;
    caractere : ORIGEM, DESTINO,
                DATA, HORÁRIO;
    real: DISTANCIA;
fim registro;

```

```

tipo VETPASSAGEM = vetor [1..44] de REGPASSAGEM;
VETPASSAGEM : ONIBUS;

```

b) Declare um conjunto de registros que comporte as informações de estoque de 500 produtos.

```

tipo MATDIAS = matriz [1..4, 1..6] de inteiro;
tipo REGPROD2 = registro
    caractere : NOME;
    inteiro : CODIGO;
    real : PRECO;
    MATDIAS : BAIXA;
fim registro;

```

```

tipo VETESTOQUE = vetor [1..500] de REGPRODUTO2;

```

```

VETESTOQUE : PRODUTOS;

```

MANIPULAÇÃO:

Ao acessar as informações contidas em um conjunto de registros, procedemos utilizando o modo de acesso característico de cada estrutura.

Exemplo:

- a) Se quisermos saber a baixa do décimo produto, da terceira semana do mês e do quarto dia da semana, escreveríamos:

```
PRODUTOS[10].BAIXA[3, 4]
```

- b) Elabore o trecho de um algoritmo que imprima o total de movimentação do estoque para cada um dos 500 produtos:

```
.
.
para N = 1 até 500 faça
    ACM ← 0;
    para I = 1 até 4 faça
        para J = 1 até 6 faça
            ACM ← ACM + PRODUTOS[N].BAIXA[I, J];
        fim para;
    fim para;
    escreva (PRODUTOS[N].NOME, ACM);
fim para;
.
.
```

Exercícios:

- 53) Baseado em seu conhecimento, defina um registro para um cheque bancário.
- 54) De acordo com o exercício anterior, construa um algoritmo que, usando um campo saldo de um registro conta bancária, imprima um relatório dos cheques sem fundos (trabalhe com um conjunto de 800 registros).
- 55) Usando o registro de estoque definido nos exemplos, elabore:
- um algoritmo que imprima o produto de maior saída;
 - um algoritmo que imprima a diferença percentual entre a maior e a menor saída;
 - um relatório com a semana de maior saída de cada produto.