Prof. Márcio Senhorinha

E-mail

marcio.senhorinha@edu.sc.senai.br

FIEP
SESI
SENAI
IEL

SENAI

Algoritmos

marcio.senhorinha@edu.sc.senai.br

FIEP SESI SENAI IEL

SENAI

Português Estruturado

VARIAVEIS COMPOSTAS HOMOGENEAS

Introdução

Até agora **utilizamos variáveis definidas a partir de tipos de dados básicos**, e como já foi visto, **uma variável só é capaz de armazenar apenas um dado de cada vez**

Introdução

Existem situações, **contudo**, em que **há necessidade de armazenar uma grande quantidade de dados na memória ao mesmo tempo**, **tornando inviável a criação de tantas variáveis.**

Introdução

Imagine o seguinte problema:

Ler 500 números inteiros, após a leitura escrever os números na ordem inversa da entrada.

Exemplo:

Lendo do usuário (20, 45, 100, ..., 34, 78)

Escrever após toda leitura (78, 34, ..., 100, 45, 20)

Introdução

Pelo que conhecemos até o momento de algoritmos, **a única forma para resolver este problema é declararmos 500 variáveis** e atribuir os 500 números a essas variáveis.

É evidente que precisamos de outra maneira para referenciar tais coleções de dados similares

Introdução

Para tanto, **foi criado um novo conceito para alocação de memória** sendo, desta forma, também criado uma nova maneira de definir variáveis, a qual foi **denominada de variável indexada**.

Introdução

Uma variável indexada corresponde a uma sequência de posições de memória, a qual daremos **único nome**, sendo que cada uma destas pode ser acessada através do que conhecemos por índice.

O **índice corresponde normalmente a um valor numérico**.

Introdução

Cada **uma das posições de memória** de uma variável indexada **pode receber valores no decorrer do algoritmo** como se fosse uma variável comum, a única diferença reside na sintaxe de utilização desta variável.

FIEP SESI SENAI IEL

SENAI

VETOR UNIDIMENSIONAL

1 . Vetor Unidimensional

Um **vetor (**também chamado de **array)** é uma estrutura de dados **composta por uma quantidade determinada de elementos de um mesmo tipo primitivo.**

1 . Vetor Unidimensional

Por ter como característica **armazenar dados que seguem um mesmo tipo básico,** dizemos que os **vetores são estruturas de dados homogêneas**

1 . Vetor Unidimensional

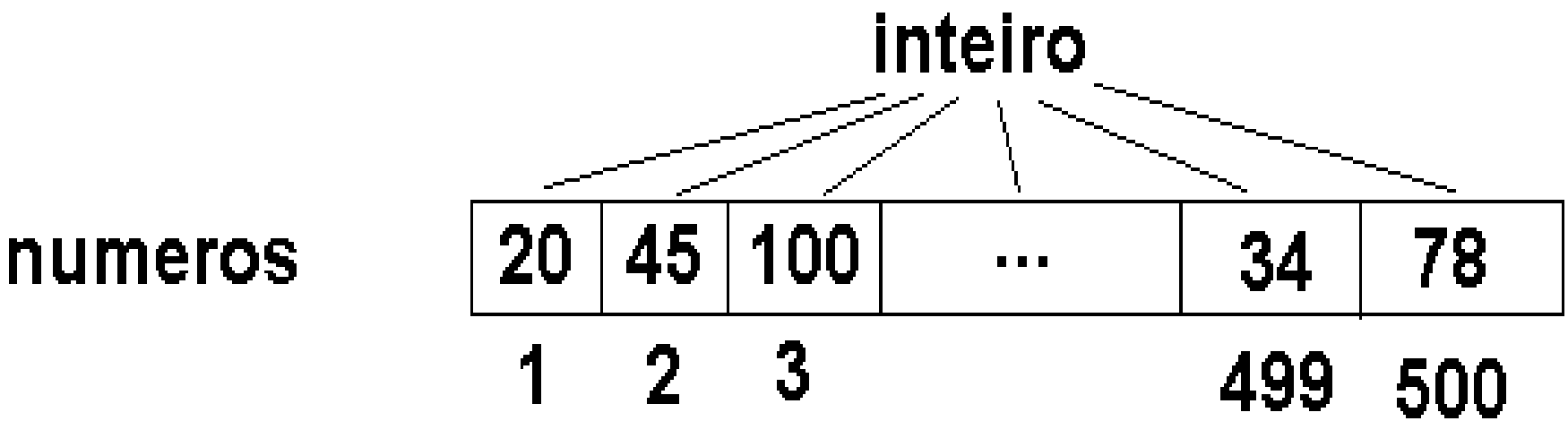
Quando estudamos o **conceito de variável**, **dissemos que ela funciona como uma caixa na memória do computador**, **capaz de armazenar um único dado**

1 . Vetor Unidimensional

Por analogia, **podemos imaginar um vetor como um conjunto de variáveis agrupadas sob um mesmo nome, todas de um mesmo tipo**

1 . Vetor Unidimensional

O vetor passa a ser um conjunto de caixinhas, contendo dados de mesma natureza e que estejam agrupados



1 . Vetor Unidimensional

Abaixo segue a forma de declaração de um vetor.

Os colchetes indicam o intervalo que será utilizado para os índices de cada posição do vetor

Sintaxe:

```
<var>: vetor[inicio..fim] de <tipo>;
```

1 . Vetor Unidimensional

EXEMPLO

Declarando um vetor para armazenar 500 números inteiros

```
NUMEROS: vetor[1..500] de INTEIRO;
```

A **variável é uma só**, no nosso caso declarada como **NUMEROS**, mas é capaz de **armazenar 500 valores inteiros** diferentes

1 . Vetor Unidimensional

Cada posição de um vetor é referenciada pela seguinte sintaxe:

```
<nome_vetor>[<posição>]
```

1 . Vetor Unidimensional

EXEMPLO 01

```
1.  Algoritmo "Exemplo_Vetor"
2.  Var
3.  numeros: vetor[1..500] de inteiro
4.
5.  Inicio
6.   numeros[10] <- 120;
7.    Escreval  ("O  indice  10  do  VETOR  já
   recebeu  o valor 120")
8.    Escreval ("Digite um NUMERO!")
9.    leia (numeros[11])
```

```
9.  Escreval ("A soma  do indice 10  que têm
   armazado o valor 120")
10.   Escreval  ("Mais  a  soma  do valor  que
   digitou e está armazado no INDICE 11 é: ")
11.   escreval (numeros[10]+numeros[11])

12. Fimalgoritmo
```

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro vetor[100]
6          vetor[10]=120
7          escreva("O indice 10 do VETOR já recebeu  o valor 120\n")
8          escreva ("Digite um NUMERO!\n")
9          leia (vetor[11])
10         escreva ("A soma  do indice 10  que têm armazenado o valor 120\n")
11         escreva ("Mais a soma  do valor que digitou e está armazenado no INDICE 11 é: ")
12         escreva ((vetor[10])+ (vetor[11]))
13     }
14 }
```

1 . Vetor Unidimensional

EXEMPLO 02

Uma prova de algoritmos foi feita por **um grupo de 20 alunos**.

Faça um algoritmo **para ler as notas** obtidas pelos alunos, e depois **exibir um relatório das notas iguais ou superiores a 7,5**.

No final **escreva a quantidade de notas iguais ou superiores a 7,5**.

1 . Vetor Unidimensional

EXEMPLO 02

```
1  Algoritmo "notas_boas"
2  Var
3     notas : vetor[1..5] de real;
4     i, cont : inteiro;
5  Inicio
6     para i de 1 ate 5 faca
7        escreval ("Digite as notas dos alunos")
8        leia (notas[i])
9     fimpara;
10    cont<- 0;
11    escreval ("Notas Boas: ")
12    para i de 1 ate 5 faca
13       Se (notas[i]) > 7.5 entao
14          escreval (notas[i])
15          cont<- cont + 1;
16       fimse
17    fimpara
18    escreval ("Quantidade de Notas Superiores:  ",cont)
19 Fimalgoritmo
```

FIEP SESI SENAI IEL

SENAI

ATIVIDADES do 025 Até 028

Atividade 025

Faça um algoritmo para **gerar um vetor de 30 posições**, onde cada elemento corresponde ao quadrado de sua posição.

Escreva depois o vetor resultante

Atividade 025 - Solução

```
1.  Algoritmo "carrega_vetor"
2.  Var
3.  numeros : vetor[1..30] de inteiro;
4.    i : inteiro;

5.  Inicio
6.    para i de 1 ate 30 faca
7.       numeros[i] <- i * i;
8.    fimpara
9.       para i de 1 ate 30 faca
10.      escreval (numeros[i])
11.   fimpara
12. Fimalgoritmo
```

Atividade 026

Num concurso público, um candidato respondeu a uma avaliação com **80 questões de múltipla escolha**, onde **cada questão tinha respostas de A até D**.

Faça um algoritmo **para ler o gabarito da prova** e **as respostas do aluno, informando quantas questões ele acertou**.

Atividade 026 - Solução

```
1  Algoritmo "AT_26_Concurso"
2  Var
3     gabarito, prova : vetor[1..80] de caractere;
4     acertos, i : inteiro;
5  Inicio
6     acertos<-0;
7     Escreval ("Digite o gabarito da PROVA do concurso")
8     Escreval (" ")
9     para i de 1 ate 3 faca
10       Escreval ("Gabarito para a questão",i,"  a Letra é:")
11       Leia (gabarito[i])
12    fimpara
13    Escreval (" ")
14    Escreval ("Digite sua RESPOSTA para o concurso")
15    para i de 1 ate 3 faca
16       Escreval ("RESPOTA para a questão",i," é:")
17       leia (prova[i])
18    fimpara
19    para i de 1 ate 3 faca
20       se (prova[i]) = (gabarito[i]) entao
21          acertos<- acertos + 1;
22       fimse
23    fimpara
24    escreval ("Você teve ",acertos ," acerto(s) das QUESTÕES respondidas" )
25 Fimalgoritmo
```

Atividade 027

Ler 30 números reais em um vetor e **depois exibir os números localizados nas posições ímpares**

Atividade 027 - Solução

```
1  Algoritmo "AT_27_Posição_Impares"
2  Var
3     numeros : vetor[1..10] de real;
4     i : inteiro;
5  Inicio
6     para i de 1 ate 10 faca
7        escreval ("Digite numero")
8        leia (numeros[i])
9     fimpara
10    i <-1;
11    escreval ("Esse são os numeros que estão nos indices impares do vetor")
12    enquanto i <= 10 faca
13       escreval (numeros[i])
14       i <- i + 2;
15    fimenquanto
16 Fimalgoritmo
```

Atividade 028

Faça um programa para **ler uma sequência de 40 notas** e **mostrar a média** e **quais as notas maiores que a média**.

Atividade 028 - Solução

```
1  Algoritmo "AT_028_nota_maior_que_media"
2  Var
3     notas : vetor[1..5] de real;
4     i: inteiro;
5     soma, media : real;
6  Inicio
7     soma <- 0;
8     escreval("Digite as notas para calcular a MÉDIA")
9     para i de 1 ate 5 faca
10       escreval(i,"a  Nota")
11       leia(notas[i])
12       soma<- soma + notas[i];
13    fimpara;
14    media<- soma/i;
15    escreval("A média das notas digitadas é:",media)
16    escreval ("Essa(s) é/são a(s) nota(s) acima da média: ")
17    para i de 1 ate 5 faca
18       se notas[i] > media entao
19          escreval (notas[i])
20       fimse;
21    fimpara;
22 Fimalgoritmo
```

REFERÊNCIAS

Slide Logica de Programação – Carlos Iran Chiarello chiarello@spei.br

Fundamentos da Programação de Computadores / ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene aparecida Veneruchi ; 3ª. ed. – São Paulo : Pearson Addison Wesley, 2011.

FIM