

Técnicas de Desenvolvimento de Algoritmo



Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

Material Teórico



Introdução a Vetores

Responsável pelo Conteúdo:

Prof. Esp. Alexander Gobbato Albuquerque

Revisão Textual:

Profa. Esp. Márcia Ota

UNIDADE

Introdução a Vetores



- Introdução a Vetores
- Arrays Bidimensionais
- Operações com Matrizes de duas Dimensões



Nesta unidade, estudaremos os conceitos básicos para a criação de algoritmos, entenderemos o conceito de lógica aplicada a programas e utilizaremos pensamento crítico, operacional e lógico, por meio de modelos de repetição ou malhas e com o conceito dos modelos de repetição ou malhas. Além disso, praticaremos nos vetores e matrizes aprendidos nessa aula.

Hoje, veremos alguns assuntos introdutórios na nossa disciplina e aproveito para apresentar-lhes alguns conceitos que utilizaremos na estrutura de todas as nossas unidades.

Para obter um bom aproveitamento, vamos conferir a estrutura desta unidade:

Conteúdo Teórico: nesse link, vocês encontrarão o material principal de estudos na forma de texto escrito.

Atividade de Sistematização: os exercícios disponibilizados são de autocorreção e visam que você pratique o que aprendeu na disciplina e que identifique os pontos em que precisa prestar mais atenção, ou pedir esclarecimentos a seu tutor. Além disso, as notas atribuídas aos exercícios serão parte de sua média final na disciplina.

Atividade de Aprofundamento: é uma atividade dissertativa ou de pesquisa.

Material Complementar e Referências Bibliográficas: nesses links, será possível ampliar seus conhecimentos.

Videoaula: nesses links, serão apresentadas algumas ferramentas na prática e também a resolução de alguns exercícios de forma prática.



Atenção

Lembramos a vocês da importância de realizar todas as atividades propostas dentro do prazo estabelecido para cada Unidade; dessa forma, evitarão que o conteúdo se acumule, evitando-se problemas ao final do semestre.

Uma última recomendação: caso tenham problemas para acessar algum item da disciplina, ou dúvidas com relação ao conteúdo, não deixem de entrar em contato com seu professor tutor através do botão mensagens ou fóruns.

Contextualização

Em algumas situações, perceberemos que para ter um controle de informações teremos que criar várias variáveis, mas não se preocupem! Isso é normal. Para solucionarmos esses problemas, utilizaremos recursos da lógica de programação chamados de malhas ou repetições aplicados a vetores e matrizes.

Com os vetores e matrizes podemos utilizar a mesma variável, apenas informando a posição que iremos armazenar.

Introdução a Vetores



Antes de começar a falar de arrays, devemos lembrar o que é uma variável e para que ela serve. Nas unidades anteriores, vimos que variável é a representação simbólica dos elementos e que ficam armazenadas em memória. Cada variável corresponde a uma posição, cujo conteúdo pode se alterar ao longo do tempo durante a execução de um programa.

No entanto, na variável só pode ser armazenada apenas por um único tipo. Além disso, nós também vimos que as variáveis podem ser de três tipos: numéricas, alfanuméricas e lógicas, e que para declarar uma variável precisamos definir o seu nome e que tipo de dados será armazenado nela.

Antes de definir vetor, imaginemos a seguinte situação:

Criar um programa que leia os nomes de 20 pessoas e, em seguida, apresente-os na mesma ordem em que foram informados.

Então, definiremos uma variável, alocamos um espaço na memória do computador para armazenar uma e somente uma constante por vez, seja ela literal, numérica ou lógica. Quando atribuímos um valor à variável, sobrescrevemos seu conteúdo.

Em algumas rotinas é necessária a manipulação de várias variáveis ao mesmo tempo, por exemplo, imagine que o programa deverá controlar o nome de 100 pessoas, ao invés de criar 100 variáveis é possível a criação de apenas uma, que é definida como array.

Mas como criar um array? É muito simples. Especificamos o nome do array e o número de posições da memória que queremos alocar. Cada posição de memória pode armazenar um valor.

Os arrays são criados para que possam armazenar várias informações do mesmo tipo ou a mesma classe, com um vetor também é possível o armazenamento de vários objetos. Essas diversas informações que foram armazenadas em array são de fácil manipulação, as informações ficam disponíveis em forma de tabelas e podem ser acessadas por meio dos índices.

Nas unidades anteriores, nós vimos que para se chamar um programa em Java nós devemos criar um programa principal com o método main e como parâmetro do método devemos criar um string args[], isso nada mais é do que um array de Strings.

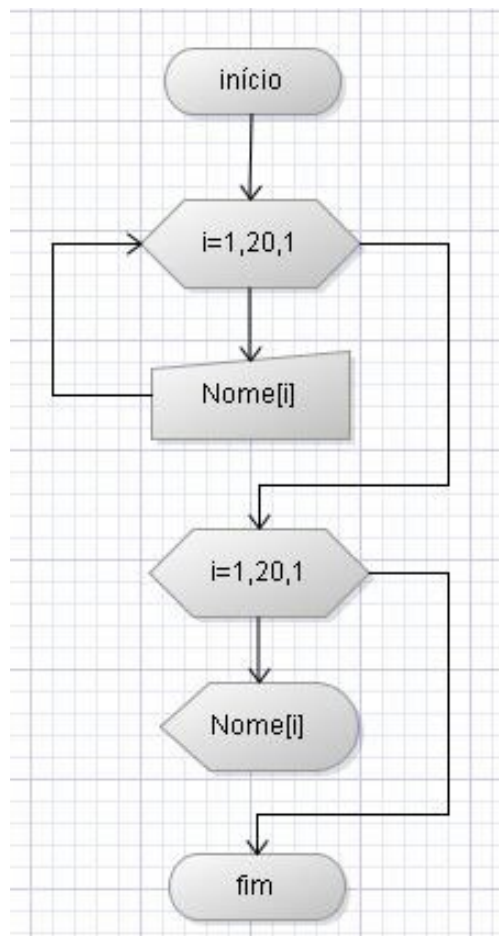
Os arrays estão presentes em todas as linguagens de programação. Em Java, os arrays são objetos que permitem armazenar uma lista de itens relacionados.

Arrays Unidimensionais

Os arrays de uma dimensão são aqueles que possuem um único índice para acessar os elementos. Para resolver a situação citada acima, a fim de criar um programa que leia os nomes de 20 pessoas e, em seguida, apresente-os na mesma ordem em que foram informados, apresentamos o algoritmo da seguinte forma:

1. Definir a variável I do tipo inteira para controlar a repetição.
2. Definir o array NOME do tipo caractere para 20 elementos.
3. Iniciar o programa, fazendo a leitura dos 20 nomes.
4. Apresentar após a leitura, os 20 nomes.

Diagrama de Blocos



Português estruturado

```
Programa LISTA_NOME
início
Var
    Nome: conjunto[1...20] de caracteres
Para i de 1 até 20 passo 1 faça
    Leia Nome[i]
Fim_para
Para i de 1 até 20 passo 1 faça
    escreva Nome[i]
Fim_para
fim
```


Arrays Bidimensionais



Os arrays bidimensionais ou arrays multidimensionais permitem a criação de vetores com mais de um índice, essa característica possibilita que os valores sejam armazenados na forma de matrizes de qualquer dimensão.

Matrizes com mais de duas dimensões são utilizadas com menos frequência, mas em alguns casos é preciso saber trabalhar com essas quantidades de posições. Sendo assim, matrizes com mais de duas dimensões são controladas por malhas de repetição. Os elementos das matrizes de mais de uma dimensão são manipuladas de forma individualizada, sendo a referência sempre por meio dos índices, o primeiro indica a linha e o segundo indica a coluna.

Operações com Matrizes de duas Dimensões



Matrizes de duas dimensões são sempre controladas por linhas e colunas e é representada pelo nome e pelo tamanho. A figura abaixo apresenta uma tabela de 5 x 4, representada da seguinte forma TAB[1..5,1..4], onde TAB é o nome da matriz e as dimensões são de 5 colunas por 4 linhas, com essas dimensões é possível armazenar até 20 elementos.

Matriz: TAB				
	1	2	3	4
1				
2				
3				
4				
5				

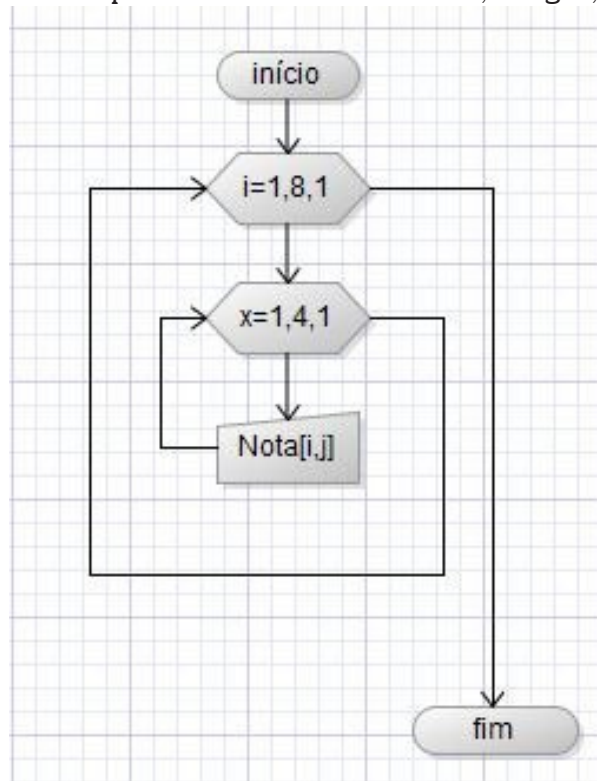
Coluna

Linha

A atribuição de uma matriz é muito parecida com a instrução conjunto já utilizada para a criação da matriz, a sintaxe é NOME_DA_VARIÁVEL[POSIÇÃO1:POSIÇÃO2] de TIPO_DE_DADO, onde POSIÇÃO1 e POSIÇÃO2 são a indicação do tamanho da tabela e o TIPO_DE_DADO é o tipo da tabela.

Para ler as informações de uma tabela de uma dimensão ou duas dimensões é feita um passo por vez, para entendermos melhor o processo de leitura imagine o seguinte problema:

O sistema deverá ler 4 notas de provas de 8 alunos diferentes, a seguir, veja o diagrama de bloco:



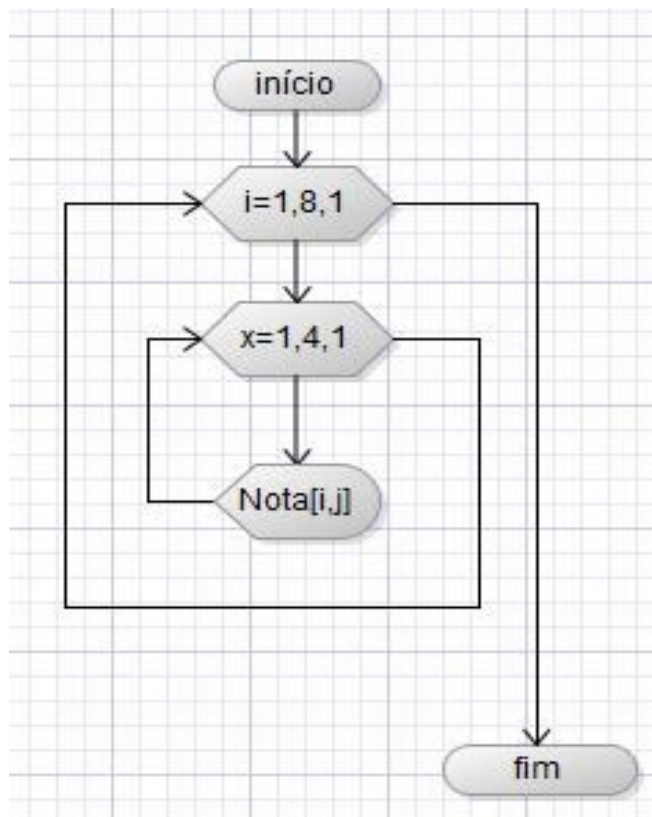
O diagrama de bloco acima representa a gravação de 4 notas de 8 alunos diferentes. O detalhe desse controle é que são variáveis *i* e *x*. A variável *i* controla a quantidade de alunos e a variável *x* controla a quantidade de notas.

Imagine as informações em tabela, onde a variável *i* pode ser representada pela linha e a variável *x* pode ser representada pela coluna.

```

Programa Gravar_Notas
Var
  Nota: conjunto[1..8,1..4] de real
  i,j: inteiro
início
  para i de 1 até 8 passo 1 faça
    para j de 1 até 4 passo 1 faça
      leia Nota[i,j]
    fim_para
  fim_para
fim
  
```

Para a leitura das notas, o processo é bem parecido com o diagrama de bloco de gravação, veja o diagrama abaixo:



Português estruturado

```

Programa Ler_Notas
Var
    Nota: conjunto[1..8,1..4] de real
    i,j: inteiro
início
    para i de 1 até 8 passo 1 faça
        para j de 1 até 4 passo 1
            faça
                escreva Nota[i,j]
            fim_para
        fim_para
    fim_para
fim
  
```

Material Complementar



Explore

OLIVEIRA, J. F e MANZANO, J. A. N. G., **Algoritmos – Lógica para desenvolvimento de programação de computadores.** Ver o conteúdo de Parte II – Técnicas Básicas de Programação - Capítulo 7 – Aplicações práticas de Matrizes do Tipo Vetor.

Referências

FARRER, H. **Algoritmos Estruturados**. 3. ed. Rio de Janeiro: Ltc-Livros Técnicos e Científicos, 1999.

FORBELLONE, A. L. V.; EBERSPACHER, H. F. **Logica de Programação: A Construção de Algoritmos e Estrutura de Dados**. 3. ed. São Paulo: Pearson Prentice Hall, 2008.

WIRTH, N. **Algoritmos e Estruturas de Dados**. Rio de Janeiro: Ltc-Livros Técnicos e Científicos, 1999.



Educação a Distância

Cruzeiro do Sul Educacional

Campus Virtual

www.cruzeirodosulvirtual.com.br

Campus Liberdade

Rua Galvão Bueno, 868

CEP 01506-000

São Paulo SP Brasil

Tel: (55 11) 3385-3000

