



Vetores

ESTRUTURA DE DADOS



VETOR (*Array*)

- É uma das estruturas de dados mais simples e uma das mais utilizadas.
- **Características principais:**
 - Acesso aos elementos através de índices
 - Possuem tamanho finito de elementos
 - Indexação começa com o (zero)
 - Armazenam dados de apenas um tipo
 - Unidimensional: Apenas uma linha
 - Multidimensional: Mais de uma linha (matriz)

VETOR UNIDIMENSIONAL

ARRAY SIMPLES

- `INT(vetor[5]);`

VETOR

10	25	3	157	2
0	1	2	3	4

`Vetor[0] = 10;`

`Vetor[1] = 25;`

`Vetor[2] = 3;`

`Vetor[3] = 157;`

`Vetor[4] = 2;`

VETOR UNIDIMENSIONAL

Vetor.java



VETOR MULTIDIMENSIONAL

MATRIZ

- `INT(vetorB[2,2]);`

Linha

Coluna

	0	1
0	10	25
1	8	954

VETORB

`VetorB[0,0] = 10;`

`Vetor[0,1] = 25;`

`VetorB[1,0] = 8;`

`Vetor[1,1] = 954;`

VETOR MULTIDIMENSIONAL

Matriz.java



VETOR UNIDIMENSIONAL

EXERCICIO

Criar um programa em java que receba 4 notas e mostre a maior.



VETOR UNIDIMENSIONAL

EXERCICIO

Criar um vetor de 20 posicoes de String.



Depois

Criar um método para adicionar um ítem no vetor na primeira posição vazia.

E após listar o vetor com os elementos

VETOR

VANTAGENS X DESVANTAGENS

Vantagens

Inserção rápida.

Acesso muito rápido se o índice for conhecido

Desvantagens

Pesquisa lenta

Remoção lenta

Tamanho fixo

VETOR ORDENADO

CONCEITO

Imagine um vetor no qual os itens são armazenados em ordem ascendente de valores-chaves, ou seja, o menor valor estará no índice 0 (zero) e os demais nos índices posteriores de acordo com o valor.

A este tipo de array chamamos de **vetor ordenado**.

VETOR

10	25	3	157	2
0	1	2	3	4

VETOR ORDENADO

2	3	10	25	157
0	1	2	3	4

VETOR ORDENADO

VANTAGENS X DESVANTAGENS

Vantagens

Pesquisa mais rápida que o vetor não ordenado.

Desvantagens

Inserção Lenta, pois a cada inserção deve-se ordenar o vetor.



Operações Básicas

- 1) Adicionar um elemento no fim do vetor.
- 2) Adiciona um elemento em uma dada posição.
- 3) Pega o elemento de dada posição.
- 4) Remove o elemento de dada posição.
- 5) Verifica se um elemento está armazenado.
- 6) Informa o número de elementos armazenados.

API do JAVA - *Arrays*

- [java.lang.Object](#)
 - **java.util.Arrays**
- Esta classe contém vários métodos para manipulação de vetores (arrays), como ordenação e busca.
- Esta classe também permite arrays serem visualizados como listas.

API do JAVA - *Vector*

- [java.lang.Object](#)
 - [java.util.AbstractCollection](#)
 - [java.util.AbstractList](#)
 - **java.util.Vector**
- A classe **Vector** implementa um array de objetos que pode aumentar.
- Como um array, ele contém componentes que podem ser acessados usando um índice inteiro. No entanto seu tamanho pode crescer ou diminuir quando os itens são adicionados ou removidos.

Operações Básicas

EXERCICIO

- Criar métodos para todas as operações básicas ditas anteriormente, baseando-se no exercício de criar um *array* de 20 posições para armazenar nomes.