+Devs2Blu

Linguagem de programação JAVA

Profª. Heloisa Moura

+Devs2Blu

Linguagem de programação JAVA

O que vamos ver:

- Revisão de arrays e matrizes

+Devs2Blu

Linguagem de programação JAVA

As estruturas de dados clássicas e mais conhecidas são:

- Arrays (“ vetores” e “matrizes”)
- Listas (List)
- Fila (Queues)
- Pilha (Stacks)
- Árvore (Trees)

Estrutura de dados simples

Estruturas de dados mais complexas que podem ser inclusive implementadas com o uso dos arrays

Arrays

- Estrutura homogênea que mantém uma série de elementos de dados de um mesmo tipo.
- Estrutura de dados mais simples existente na maioria das linguagens de programação.
- Pode-se acessar os elementos individuais por meio de uma posição de índice, geralmente numérica.
- Possuem tamanho fixo.

+Devs2Blu

Linguagem de programação JAVA

Os **arrays** são classificados em:

- Unidimensional: **Vetor**

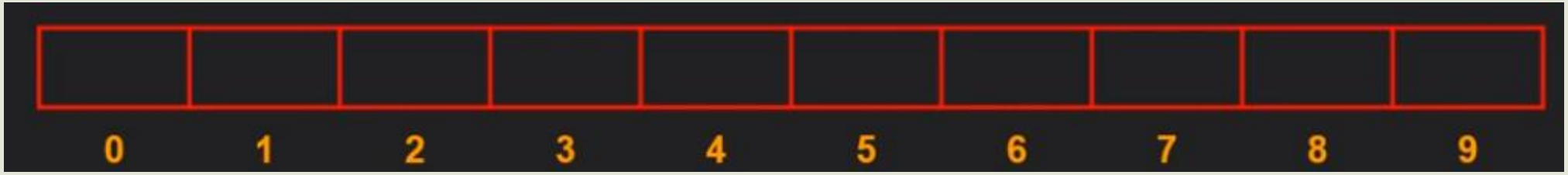
  Pensar como: Uma série de dados em forma de uma linha.

- Bidimensional: **Matriz**

  Pensar como: Uma série de dados em forma de uma tabela.

Arrays em Java

São usados para armazenar uma coleção de dados, como sendo coleções de variáveis do mesmo tipo.



Declaração de arrays em Java

Para usarmos um **array**, primeiro declaramos uma variável para referenciá-lo , e especificamos o tipo de **array** que será referenciado (qualquer tipo, incluindo uma classe).

Essa é a forma mais simples de se declarar um **array** em Java.

Sintaxe : **tipoDado[] NomevariávelReferencia;**
Exemplo:
double[] salários;

Porém, existem formas melhores e mais completa para declarar um **array** em Java.

Como criar um arrays em Java

Criamos um **array** usando o operador **new.** Forma mais recomendada.

Sintaxe:

tipoDado[] NomevariávelReferencia = new tipoDado[tamanhoArray];

[tamanhoArray] = Quantas posições o array terá.

Declaramos a variável array, criamos o array , e atribuímos a referência do array a variável de uma só vez.

Como criar um arrays em Java

- As posições do **array** são como se fossem variáveis individuais, onde podemos gravar um dado por posição.
- Os **arrays** são indexados e acessamos cada uma das suas posições a partir de um numero chamado de índice. Como se fosse uma espécie de endereço.

Nota: As posições do **array** em Java sempre são contadas a partir do índice zero.

Como criar um array em Java.

Exemplo

Vamos criar um array de nome valores com 10 elementos do tipo double:

```
double[] valores =new double[10];
```

Arrays convencionais não permitem que aumentem ou diminuam o numero de posições em tempo de execução. Para isso existem outras estruturas de dados mais adequadas.
Arrays convencionais tem tamanho fixo.

Como criar um array em Java.

- O tipo de dados que vai em cada posição tem que ser o mesmo.
- Os índices são numerados a partir do 0 zero, ou seja: o **array** começa em zero e vai até **variavelReferencia.length -1**

Atribuir valores ao Array.

Length: É um atributo do **array** que verifica o numero de elementos que ele tem. O tamanho que foi definido.

Podemos atribuir valores a um **array** por meio do número de índice de cada posição.

```
valores[0] = 5.0;

valores[1] = 4.8;

valores[2] = 7.9;

valores[3] = 4.2;

valores[4] = 6.4;
```

Atribuir valores ao array.

O **array** também pode ser criado a partir de uma lista de valores, atribuídos durante sua criação:

Int[] números = { 45, 85, 56, 89, 23, 41 , 12, 90};

Chamamos a está lista de **Lista de inicialização**.

Trabalhando com arrays.

- Para processar um array geralmente usamos os laços for pois o tamanho do array é conhecido e todos os seus elementos são do mesmo tipo.
- Para imprimirmos os valores do **array** (output) podemos também utilizar o **foreach.**
- Não funciona para atribuirmos os valores do tipo primitivo. Para isso utilizamos o laço **for** padrão.

Exemplo a seguir.

Exemplos práticos

ExemploArray

ExemploArrayTemperatura

Exercicio 1.

Armazenar a temperatura média diária por 1 ano. 5 minutos

Exercício 2.

Criar um vetor A com 5 elementos inteiros. Construir um vetor B de mesmo tipo e tamanho e com os "mesmos" elementos do vetor A, ou seja B[i] = A[i]. 5 minutos

Matrizes

Matrizes

- Em matemática, definimos uma matriz m x n(m por n) como um conjunto de elementos dispostos em m linhas e n colunas.
- Comumente representamos os elementos da matriz com os índices i (linhas do elemento) e j (colunas do elemento).

Matrizes

- Podemos chamar uma Matriz de arrays multidimensionais. (com mais de duas dimensões)
- Uma Matriz de duas dimensões é um array bidimensional, composta por linhas e colunas como uma tabela.
- Considere uma matriz como um conjunto de vetores interligados.
- Matriz == array de arrays

Matrizes

Abaixo a matriz com alguns dados inseridos:

	0	1	2	3
0	**11**			
1			30	
2		25		
3			14	

VALOR 11 = POSIÇÃO 0,0

VALOR 25 = POSIÇÃO 2,1

VALOR 30 = POSIÇÃO 1,2

VALOR 14 = POSIÇÃO 3,2

Matrizes em Java

Problema

Armazenar as 4 notas do ano de 30 alunos.

Matrizes em Java

Solução!!!

Será que a solução é declarar um vetor para cada aluno?

```
double[] aluno1 = new double[4];
double[] aluno2 = new double[4];
double[] aluno3 = new double[4];
double[] aluno4 = new double[4];
.
.
.
```

Matrizes em Java

Solução!!!

Será que a solução é declarar um vetor para cada aluno?

NÃO!!!

Ficaria um código complexo de difícil manutenção.

Matrizes em Java

Solução!!!

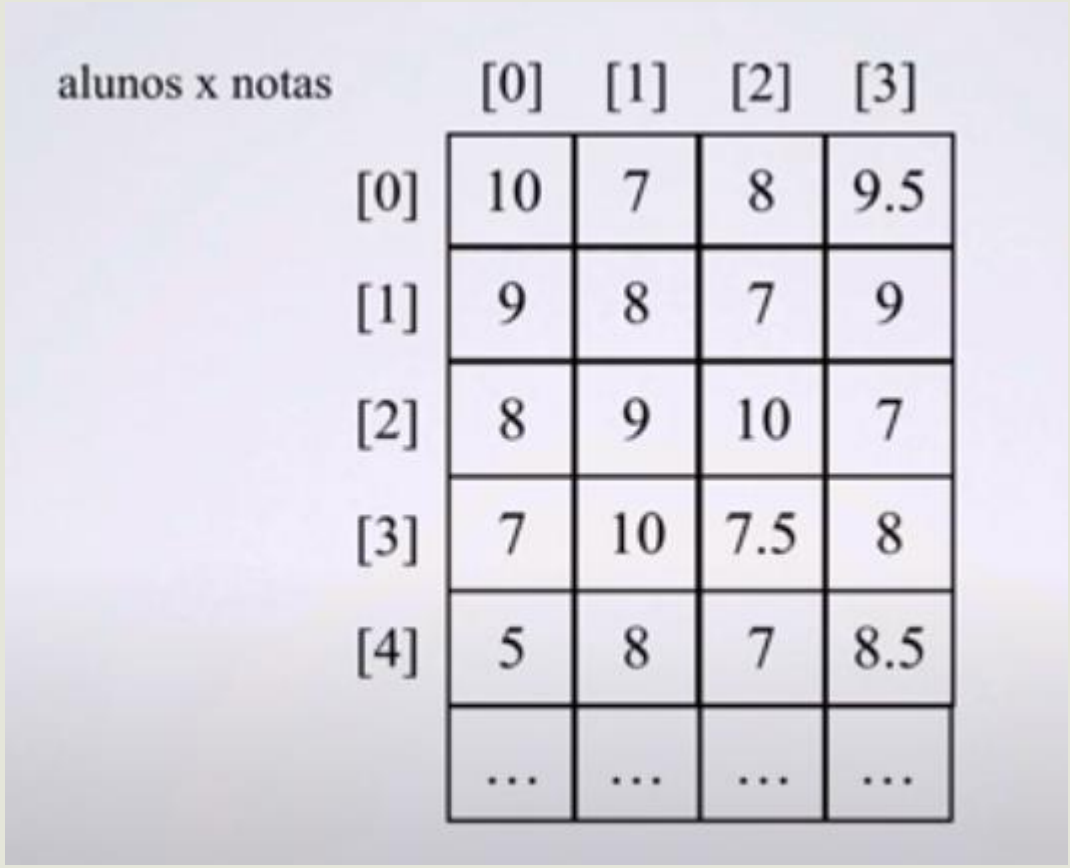
Uma Matriz de 30 linhas e 4 colunas. Como no excel.

NOTAS

ALUNOS

	A	B	C	D
1	10	7	8	9,5
2	9	8	7	9
3	6,5	7	8	10
4	8	9	9,5	10
5	8,5	10	8,5	9
6	9	7	9	8,5
7	7	8,5	10	9,5

Matrizes em Java



alunos x notas	[0]	[1]	[2]	[3]
[0]	10	7	8	9.5
[1]	9	8	7	9
[2]	8	9	10	7
[3]	7	10	7.5	8
[4]	5	8	7	8.5
	...	...	...	...

Como criar uma Matriz em Java

Sintaxe:

tipoDado[] [] variávelReferencia = new tipoDado[] [];

Onde: Primeiro par de colchetes é as linhas e o segundo par de colchetes é as colunas.

Lembrando: Uma Matriz é um array de arrays unidimensional.

Como criar uma Matriz em Java

Boa pratica: Os nomes de variáveis de matrizes serem no plural.

Exemplo: notasAlunos

Como criar uma Matriz em Java:
Arrays bidimensionais com expressões de criação de array

```
double[] []  notasAlunos  =  new double[ 30 ] [ 4 ] ;
```

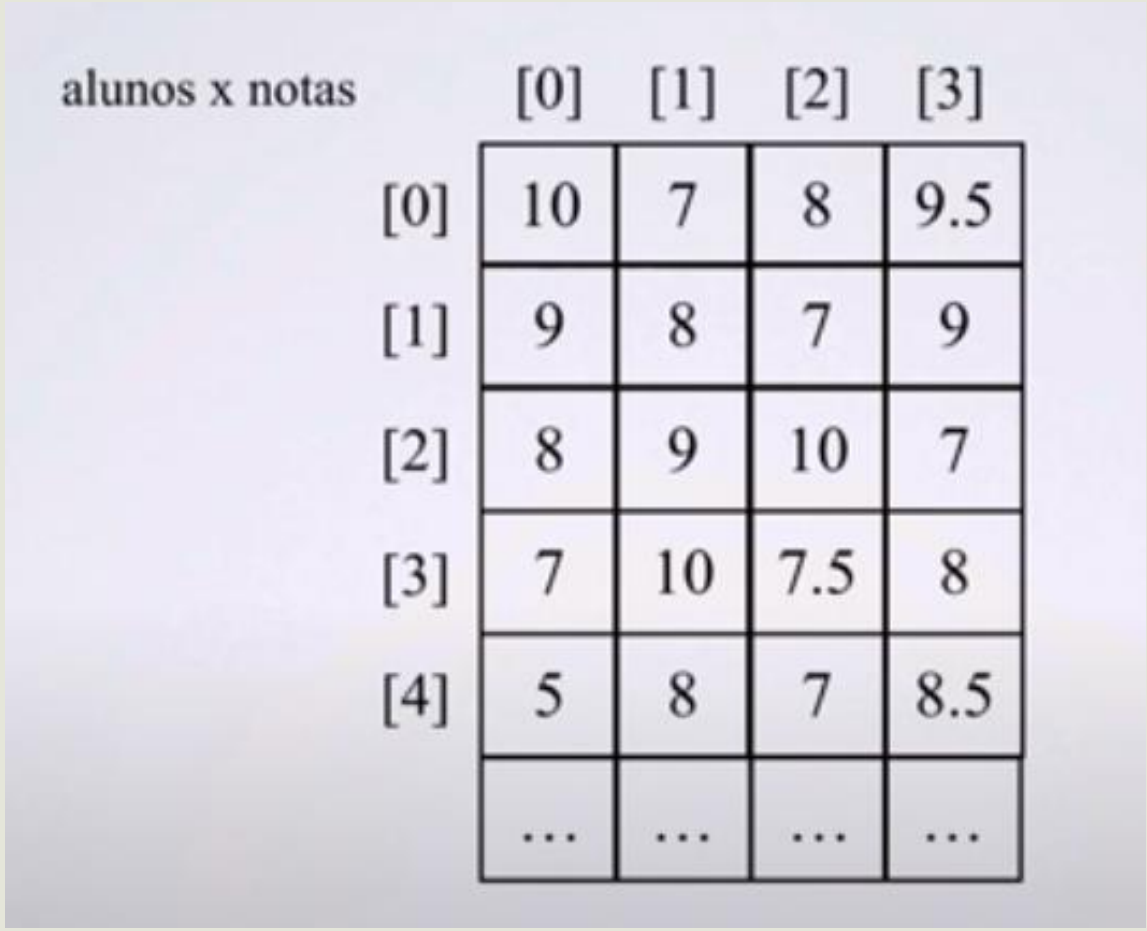
+Devs2Blu

Linguagem de programação JAVA

```
double[] [] notasAlunos = new double[ 30 ] [ 4 ] ;
```

```
notasAlunos[0] [0] = 10;
notasAlunos[0] [1] = 7;
notasAlunos[0] [2] = 8;
notasAlunos[0] [3] = 9,5;

notasAlunos[1] [0] = 9;
notasAlunos[1] [1] = 8;
notasAlunos[1] [2] = 7;
notasAlunos[1] [3] = 9;
```



alunos x notas	[0]	[1]	[2]	[3]
[0]	10	7	8	9.5
[1]	9	8	7	9
[2]	8	9	10	7
[3]	7	10	7.5	8
[4]	5	8	7	8.5
	...	...	...	...

Como criar uma Matriz em Java:

Com inicializadores de array aninhados como abaixo:

```
int [] []  notasAlunos  =  { { 9, 10 }, { 7, 8 } };
notasAlunos [0] [0] = 9;    notasAlunos [1] [0] = 7
notasAlunos [0] [1] = 10;   notasAlunos [1] [1] = 8
```

Arrays bidimensionais com linha de diferentes comprimentos

A maneira como os arrays multidimensionais são representados os torna bem flexíveis. Os comprimentos das linhas no array **b** não precisam ser os mesmos.

Por exemplo:

```
int [] []  b  =  { { 1, 2 }, { 3, 4, 5 } };
```

Arrays bidimensionais com linha de diferentes comprimento

```
int [] []  b  =  { { 1, 2 }, { 3, 4, 5 } };
```

- Cria array de inteiros **b** com dois elementos (determinados pelo numero de inicializadores de array aninhados) que representam as linhas do array bidimensional.
- Cada elemento de **b** é uma referência a um array unidimensional de int. O array int da linha 0 é um array unidimensional com dois elementos (1 e 2) e o array da linha 1 é um array unidimensional com três elementos (3, 4, e 5).

Arrays bidimensionais com numero de colunas diferentes.

Pode-se criar um array bidimensional em que cada linha tem um numero diferente de colunas como no exemplo:

int [ ] [ ] b = new int[2] [ ] cria duas linhas

b[0] = new int[5]; cria 5 colunas para linha 0.

b[1] = new int[3]; cria 3 colunas para a linha1.

As instruções anteriores criam um array bidimensional com duas linhas. A linha 0 tem cinco colunas e a linha 1 tem três colunas.

Exemplo pratico

ExemploMatrizNotasAlunos

ExemploMatriz

ExemploMatrizInicializadores

+Devs2Blu

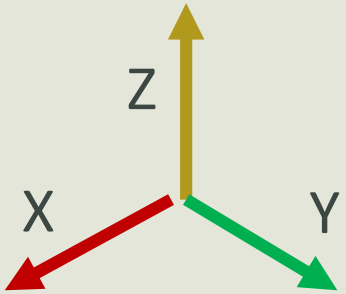
Linguagem de programação JAVA

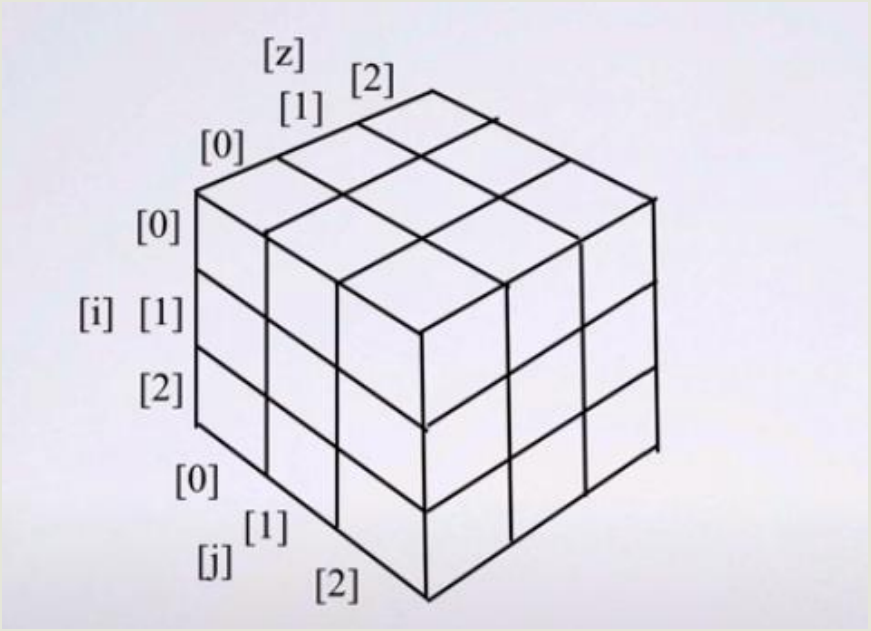
Exercicio 1. Preenchendo e exibindo uma Matriz

Crie um programa que preenche uma matriz 3x3 com valores de 1 a 9 e os exibe. 5 minutos

Matrizes de 3 dimensões: Array Multidimensional

Uma matriz de 3 dimensões é vista como um cubo. composta por 3 arrays unidimensionais.

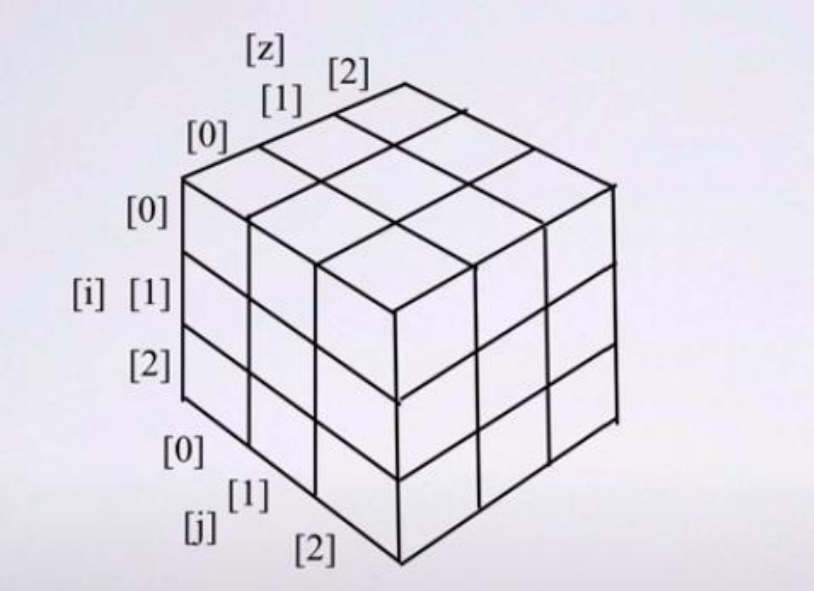




Matrix[x] [y] [z]

Matrizes de 3 dimensões: Array Multidimensional

Para iterarmos uma matriz de 3 dimensões utilizamos o i para as linhas, o j para as colunas e o k para o comprimento.



Matrizes de 3 dimensões: Array Multidimensional

- Sintaxe básica:

```
int[][][] matriz3D = new int[3][3][3];
```

- Acessando Elementos das Matrizes:

```
System.out.println(matriz3D[0][1][2]); // Exibe o elemento na primeira camada, segunda linha e terceira coluna
```

Matrizes de 3 dimensões: Array Multidimensional

- Preenchendo a Matriz 3D

```
int valor = 1;
for (int i = 0; i < matriz3D.length; i++) {
        for (int j = 0; j < matriz3D[i].length; j++) {
                for (int k = 0; k < matriz3D[i][j].length; k++) {
                        matriz3D[i][j][k] = valor++;
}
        }
                }
```

Exemplo pratico

ExemploIniciaMatriz3D

+Devs2Blu

Linguagem de programação JAVA

Exercicio 1 Matriz de 3 dimensões:

Faça um programa que crie uma matriz 3x3x3 onde cada elemento da matriz seja igual a soma dos seus índices (exemplo M=[1,2,1] = 1+2+1= 4).
Obtenha a soma de todos elementos da matriz, obtenha a soma dos elementos cujos valores são pares e a soma dos elementos cujo os valores são impares.
Exibir na tela os valores da soma total, soma dos pares e soma dos impares.

Exercícios de fixação