

Conceitos base da linguagem JAVA

Profª. Heloisa Moura

Conceitos base da linguagem JAVA

O que vamos ver:

- Conceito de estrutura de dados
- Vetores
- Matrizes

Que são os Arrays

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados

Estruturas de dados

São formas de armazenamento e organização de dados na memória do computador, para que possam ser usados de forma mais eficiente. Podem ser empregados em diferentes tipos de aplicações, sendo em alguns casos bem especializadas e orientadas a tarefas específicas .

Algumas estruturas de dados são de uso “geral”, sendo adequadas inclusive para gerar outras estruturas de dados mais complexas.

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados

São tipos de dados compostos, classificados em estruturas homogêneas e heterogêneas, de acordo com os tipos de dados primitivos que elas contém.

Homogêneas = Mesmo tipo de dados

Ex.: Apenas dados do tipo inteiro, ou tipo literal.

Heterogêneas = Tipos de dados distintos.

Ex.: Pode armazenar ao mesmo tempo dados do tipo inteiro, real literais, caracteres etc.

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados

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

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Arrays

Arrays

- É uma estrutura homogênea que mantém uma série de elementos de dados de 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.

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Arrays

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.

Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Arrays

Arrays em Java

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



Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Arrays

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.

Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Arrays

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.

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.

Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Arrays

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.

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`

Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Arrays

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;
```


Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Arrays

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**.

Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Arrays

Trabalhando com arrays.

Para processar um array geralmente usamos os laços for ou foreach, pois o tamanho do array é conhecido e todos os seus elementos são do mesmo tipo.

Exemplo a seguir.

Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Arrays

Trabalhando com arrays.

Problema!!!

Armazenar a temperatura média diária por 1 ano.

Solucionar no Eclipse.

Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Arrays

Trabalhando com arrays.

Problema!!!

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 $B[i] = A[i]$.

Solucionar no Eclipse

Iniciando no Java - Conceitos base da
linguagem - Estruturas de dados - Arrays

EXERCÍCIOS

Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Matrizes

Matrizes

- Introdução e Prática;
- Manipulação;
- Matrizes de 3 dimensões;
- Matrizes irregulares;
- for aprimorado

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

Matrizes

- Em matemática, definimos uma matriz $m \times n$ (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).

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

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

Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Matrizes

Matrizes

Abaixo temos um exemplo de uma matriz com 4 linhas e 4 colunas:

	0	1	2	3
0				
1				
2				
3				

Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Matrizes

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 30 = POSIÇÃO 1,2

VALOR 25 = POSIÇÃO 2,1

VALOR 14 = POSIÇÃO 3,2

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

Matrizes em Java

Problema

Armazenar as 4 notas do ano de 30 alunos.

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

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];
```

```
·  
·  
·
```


Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

Matrizes em Java

Solução!!!

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

NÃO!!!

Ficaria um código complexo de difícil manutenção.

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

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

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

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

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

Como criar uma Matriz em Java

Sintaxe:

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

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

Lembrando: Uma Matriz é um array de arrays unidimensional.

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

Como criar uma Matriz em Java

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

Exemplo: notasAlunos

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

Arrays bidimensionais com expressões de criação de array

```
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

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

Arrays de arrays unidimensionais

Como os arrays unidimensionais, os arrays multidimensionais podem ser inicializados com inicializadores de array em declarações. Um array bidimensional notasAlunos com duas linhas e duas colunas poderia ser declarado e inicializado 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
```


Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

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 } };
```

Cria array de inteiros b com dois elementos (determinados pelo número 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).

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

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.

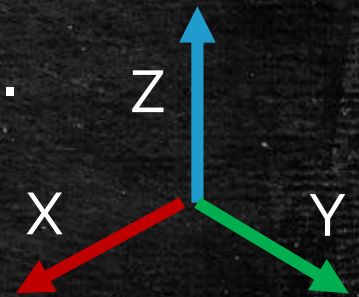
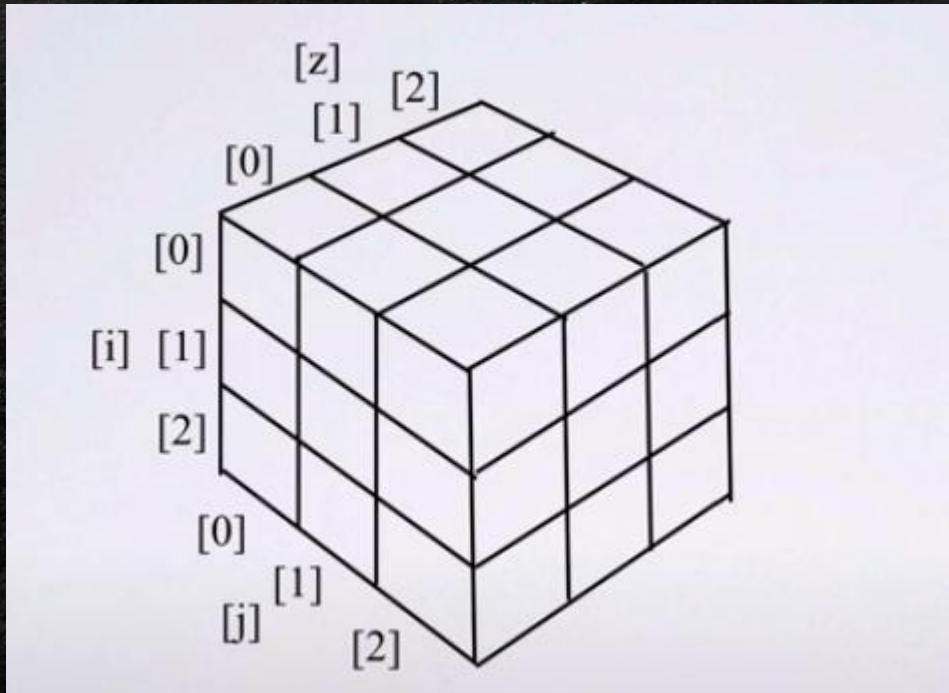
Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Matrizes

Exemplo pratico!

Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Matrizes

Matrizes de 3 dimensões: Array Multidimensional

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

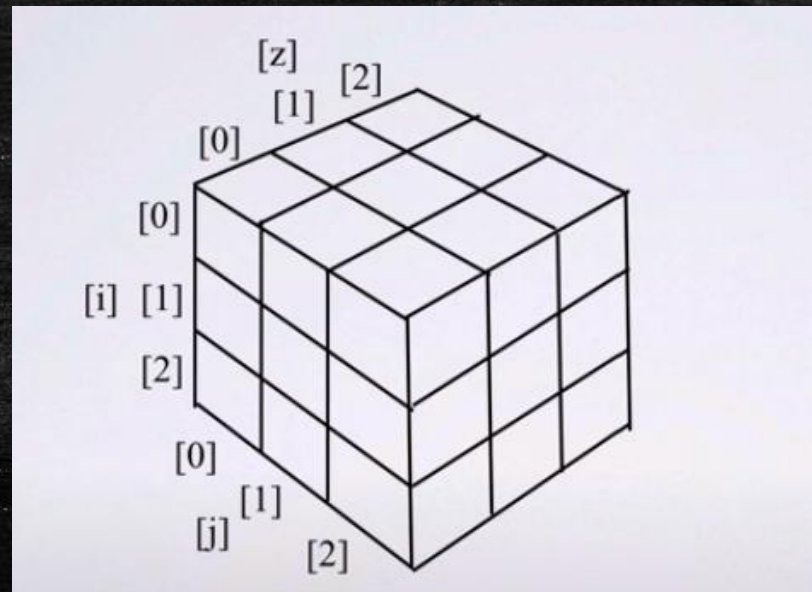


`Matrix[x][y][z]`

Iniciando no Java - Conceitos base da linguagem - Estruturas de dados - Matrizes

Matrizes de 3 dimensões: Array Multidimensional

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



Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

Exemplo Matriz de 3 dimensões:

Faça um programa que crie uma matriz $3 \times 3 \times 3$ 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 ímpares.

Exibir na tela os valores da soma total, soma dos pares e soma dos ímpares.

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

Exemplo de Matriz com numero de colunas diferente

Escreva um programa para guardar dados de uma pesquisa feita com uma amostra de pessoas da avenida mais movimentada da cidade onde você mora.

A pesquisa consiste em perguntar ao cidadão a quantidade de filhos e o nome de cada filho. Guarde essas informações em uma matriz de forma que cada pessoa entrevistada ocupe somente o espaço necessário na memória para guardar os nomes dos filhos. Imprima a quantidade de pessoas entrevistadas, a quantidade de filhos de cada entrevistado e os respectivos nomes dos filhos.

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

Estrutura de repetição – for aprimorado para utilizar em arrays (for each)

A instrução for aprimorado itera pelos elementos de um array sem usar um contador, evitando assim a possibilidade de ultrapassar o limite do array. A instrução for aprimorado se aplica ao java 5.0 em diante. Mas não funciona com as versões mais antigas do Java.

**Sintaxe: for (parâmetro : nomeDoArray)
instrução**

Iniciando no Java - Conceitos base da linguagem – Estruturas de dados - Matrizes

Estrutura de repetição – for aprimorado para utilizar em arrays.

Onde parâmetro tem duas partes um tipo e um identificador (exemplo: `int numero`) e `nomeDoArray` é o array pelo qual iterar. A instrução `for` aprimorada simplifica o código para iterar por um array. Pode ser utilizada apenas para obter elementos do array. Ela não pode ser utilizada para modificar elementos. Para isso utilizar a tradicional instrução `for`.

Iniciando no Java - Conceitos base da
linguagem - Estruturas de dados - Matrizes

Exemplo na pratica
e
Exercícios