

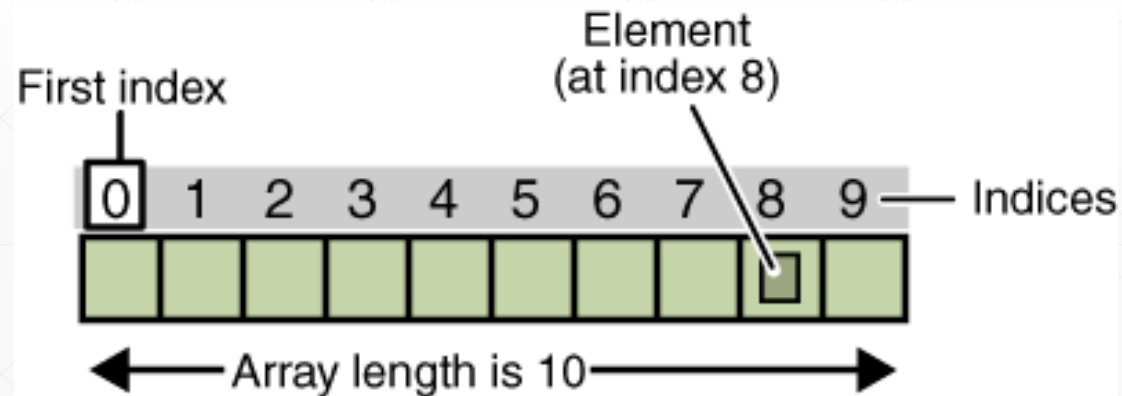
Linguagem de Programação

ADS – Análise e Desenvolvimento de Sistemas
Prof. Vagner Macedo

Aula 12
Vetores e Matrizes

Definição

- Objeto que armazena um número fixo de valores de um mesmo tipo
- A quantidade de elementos é definida na criação do *array*
- Os elementos são armazenados em um único bloco indexado:



Declaração e Inicialização

- Pode ser declarado e inicializado ao mesmo tempo;
- Pode ser inicializado com {};

```
int n[] = new int[3];  
int n[] = new int[] {1, 2, 4, 8};  
int[] n = {1, 2, 2, 3, 3, 3, 6, 6, 6, 6};
```

Exemplo I

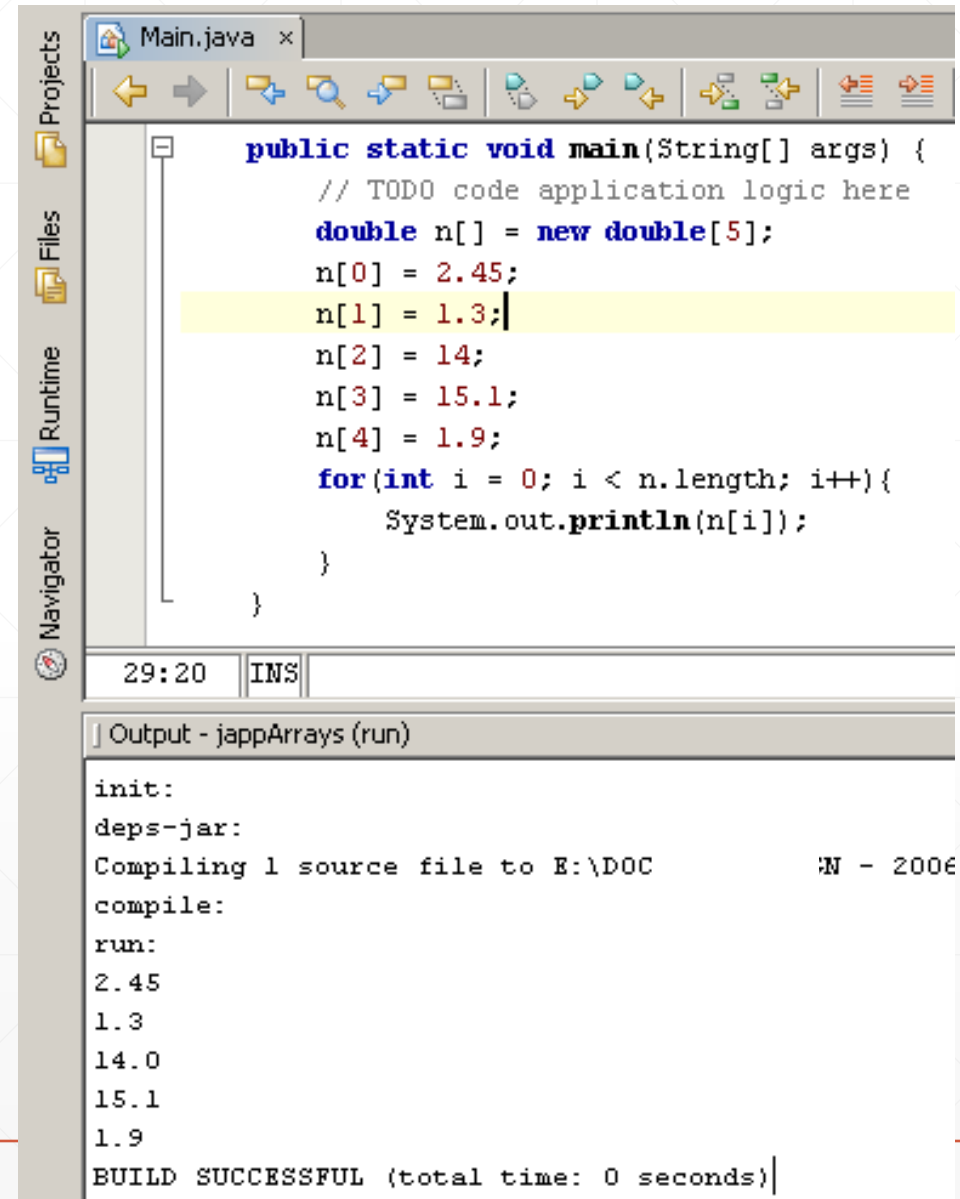
```
int n[] = new int[3];  
n[0] = 1;  
n[1] = 10;  
n[2] = 100;  
for (int i = 0; i < n.length; i++) {  
    System.out.println(n[i]);  
}
```

Exemplo I

```
// é possível redimensionar um vetor já declarado;  
// entretanto, os valores iniciais são zerados  
n = new int[7];  
for (int i = 0; i < n.length; i++) {  
    System.out.println(n[i]);  
}
```

Exemplo II

- No exemplo ao lado, um *array* unidimensional de 5 elementos do tipo *double* está sendo declarado e inicializado.
- No “for”, uma variável de controle (por exemplo, o famoso “i”) acessa os elementos do *array*, atuando como um índice.



The screenshot shows an IDE window titled 'Main.java'. The code defines a `main` method that creates a `double` array `n` of size 5 and initializes it with values 2.45, 1.3, 14, 15.1, and 1.9. A `for` loop prints each element. The output window at the bottom shows the execution results, including compilation and runtime output.

```
public static void main(String[] args) {  
    // TODO code application logic here  
    double n[] = new double[5];  
    n[0] = 2.45;  
    n[1] = 1.3;  
    n[2] = 14;  
    n[3] = 15.1;  
    n[4] = 1.9;  
    for(int i = 0; i < n.length; i++){  
        System.out.println(n[i]);  
    }  
}
```

29:20 INS

Output - jappArrays (run)

```
init:  
deps-jar:  
Compiling 1 source file to E:\DOC      N - 2006  
compile:  
run:  
2.45  
1.3  
14.0  
15.1  
1.9  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Exemplo III

- Neste exemplo, um array de Strings armazena alguns nomes e um for os mostra na tela

```
1 package prjArray;
2
3 import java.util.Iterator;
4
5 public class TesteArrayBasico {
6
7     public static void main(String[] args) {
8         String nomes[] = new String[4];
9         nomes[0] = "Vagner";
10        nomes[1] = "Fernanda";
11        nomes[2] = "Mariella";
12        nomes[3] = "Victor";
13
14        for (int i = 0; i < nomes.length; i++) {
15            System.out.println(nomes[i]);
16        }
17    }
18
19 }
20
21 }
```

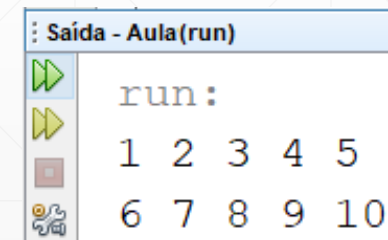
Dimensões

- O par de colchetes de um *array*[] define a dimensão do mesmo;
- A dimensão representa a indexação do *array*;
- Isso significa que um vetor pode ter mais de uma dimensão, de forma que sua indexação pode ser composta. Nesse caso, costumamos chamá-lo de “matriz”;
- Um *array* de uma dimensão é chamado comumente de “**vetor**”.

Array Bi-dimensional

- Repare no exemplo que temos um *array* com dois pares de colchetes
- A quantidade de elementos do *array* pode ser definida por D1 x D2 x... Dn
- No exemplo, dois “for” aninhados “varrem” o *array* através de duas variáveis de controle, atuando com um índice composto de duas dimensões

```
public static void main(String[] args) {  
    int matriz[][] = new int[][]{ {1, 2, 3, 4, 5},  
                                   {6, 7, 8, 9, 10} };  
    for (int l = 0; l < matriz.length; l++){  
        for (int col = 0; col < matriz[l].length; col++){  
            System.out.print(matriz[l][col] + " ");  
        }  
        System.out.println("");  
    }  
}
```

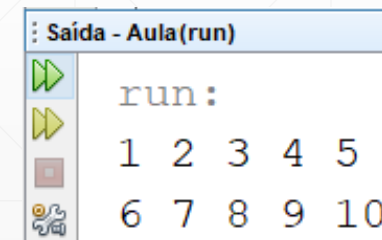


```
Saída - Aula(run)  
run:  
1 2 3 4 5  
6 7 8 9 10
```

Array Bi-dimensional

- No primeiro laço (no laço mais “externo”), o trecho “`i < a2.length`” refere-se à quantidade de linhas da matriz.
- Já no segundo laço (o laço mais “interno”), o trecho “`j < a2[i].length`” significa que, para cada linha “`i`” da iteração anterior, o laço percorrerá todas as colunas existentes.

```
public static void main(String[] args) {  
    int matriz[][] = new int[][]{ {1, 2, 3, 4, 5},  
                                   {6, 7, 8, 9, 10} };  
    for (int l = 0; l < matriz.length; l++){  
        for (int col = 0; col < matriz[l].length; col++){  
            System.out.print(matriz[l][col] + " ");  
        }  
        System.out.println("");  
    }  
}
```



```
Saída - Aula(run)  
run:  
1 2 3 4 5  
6 7 8 9 10
```

Outro exemplo

- Uma matriz matemática exemplifica uma aplicação prática, que é o cálculo da determinante.

```
public static void main(String[] args) {  
    int[][] matriz = new int[3][3];  
    matriz[0][0] = 9; matriz[0][1] = 3; matriz[0][2] = 1;  
    matriz[1][0] = 6; matriz[1][1] = 8; matriz[1][2] = 3;  
    matriz[2][0] = 7; matriz[2][1] = 3; matriz[2][2] = 5;  
  
    int determinante = 0;  
    determinante += matriz[0][0] * matriz[1][1] * matriz[2][2];  
    determinante += matriz[0][1] * matriz[1][2] * matriz[2][0];  
    determinante += matriz[0][2] * matriz[1][0] * matriz[2][1];  
    determinante -= matriz[0][2] * matriz[1][1] * matriz[2][0];  
    determinante -= matriz[0][1] * matriz[1][0] * matriz[2][2];  
    determinante -= matriz[0][0] * matriz[1][2] * matriz[2][1];  
  
    System.out.println(determinante);  
}
```

30:61 INS

Output - jappArrays (run)

compile:
run:
214
BUILD SUCCESSFUL (total time: 1 second)

Obrigado!

