

Universidade Federal da Paraíba

Centro de Informática

Departamento de Informática

Linguagem de Programação I

Arrays em Java

▶ Tiago Maritan

▶ tiago@ci.ufpb.br



Arrays (Vetores)

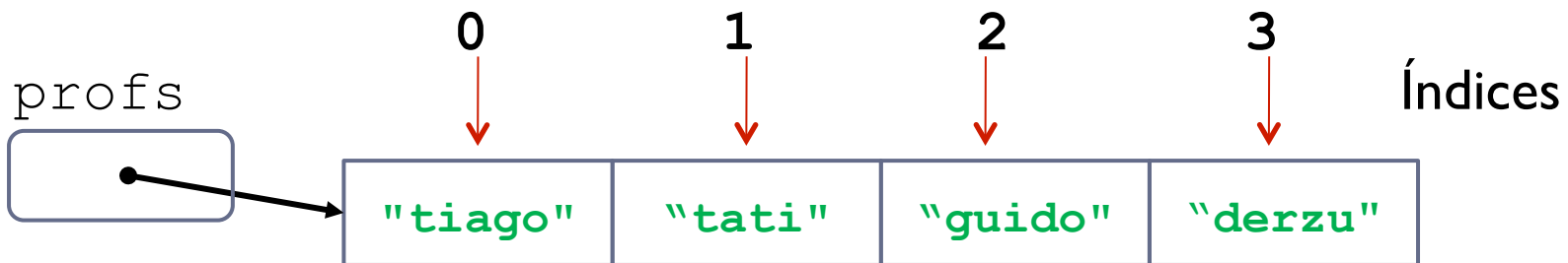
- ▶ Ex1: Se eu preciso manipular 100 nomes de pessoas ao mesmo tempo... é melhor...
 - ▶ Declarar **100 variáveis** (1 para cada pessoa)... ou
 - ▶ Declarar apenas **1 variável** (um vetor com 100 nomes)?

- ▶ Ex2: Se eu quero computar os votos dos 22 jogadores de um jogo ao mesmo tempo... é melhor...
 - ▶ Declarar **22 variáveis contadores** (1 para cada jogador)... ou
 - ▶ Declarar apenas **1 variável** (um vetor com 22 contadores)?



Arrays (Vetores)

- ▶ Armazenam um **conjunto de itens do mesmo tipo** (ou classe) em uma **mesma variável**.
 - ▶ Os itens são armazenados em **forma de tabelas**;
 - ▶ Referenciados por **índices numéricos**;
- ▶ Seu tamanho é pré-estabelecido quando ele é criado



Declaração de Arrays em Java

- ▶ Arrays são declarados usando [e]

```
<tipo>[] <identificador> = new <tipo>[<quant>];
```

```
<tipo> <identificador>[] = new <tipo>[<quant>];
```

- ▶ Ex:

```
String[] nomes = new String[100];  
int[] contVotos = new int[22];  
double medias[] = new double[40];
```



Declaração de Arrays com Inicialização

- ▶ Podem ser declarados e inicializados ao mesmo tempo

```
<tipo> <identificador>[] = { valores };
```

- ▶ Ex:

```
char arrayChar[] = {'a', 'b', 'c', 'd'};  
int votos[] = {0,0,0,0,0};
```



Acesso a elementos do array

- ▶ Através de índices numéricos de 0 até quantidade-1

`<identificador>[índice]`

- ▶ Exemplos

```
contVotos[0] = 12;  
profs[3] = "derzu";  
media[10] = 9.25;  
  
totalVotos += contVotos[5];  
contVotos[1]++;
```



Comprimento de arrays

- ▶ Para acessar o nº de elementos utilize:

`<identificador>.length`

- ▶ Exemplo:

```
int a[] = new int[100];  
for (int i = 0; i < a.length; i++) {  
    a[i] = i;  
}
```



Exercício 1

- ▶ **Parâmetro** `String []args` **do método** `main` **é um array.**
- ▶ Implemente um programa onde o usuário passa 3 números como parâmetros (do programa) e encontre o menor deles.



Estrutura de repetição “for each”

- ▶ Permite fazer loop para cada elemento de um array (ou coleção) sem mexer com índices

```
for (<variável>: <array ou coleção>) {  
    <instruções>  
}
```

- ▶ Introduzido no Java SE 5.0

- ▶ Ex:

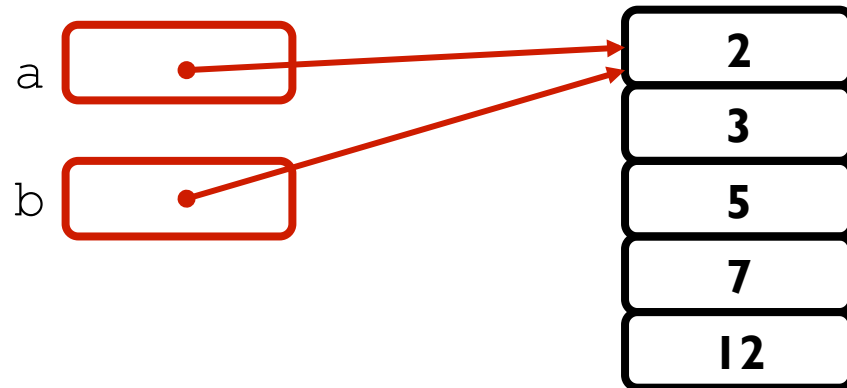
```
int a[] = new int[100];  
...  
for (int elemento : a) {  
    System.out.println(elemento);  
}
```

Cópia de Arrays

- ▶ Quando se atribui uma variável array a outra... ambas **as variáveis irão se referir ao mesmo array**

```
int a[] = {2,3,5,7,11};  
int b[] = a;  
  
b[4] = 12; // agora, a[4] tb é 12;
```

- ▶ Por quê?



Cópia de Arrays

- ▶ Para copiar os valores de um array para outro array, use:

```
int b[] = Arrays.copy(a, a.length);
```

**Array a
ser copiado**

**Tamanho do
novo array**

- ▶ Pode-se aumentar (ou diminuir) o novo array
 - ▶ Elementos restantes receberão valor 0 (ou `false`)

```
int b[] = Arrays.copy(a, 2 * a.length);
```



Arrays Multidimensionais

- ▶ Permitem a criação de vetores com mais de um índice
 - ▶ Valores armazenados em forma de matrizes;

```
<tipo> <ident>[][] = new <tipo>[<quant1>][<quant2>];
```

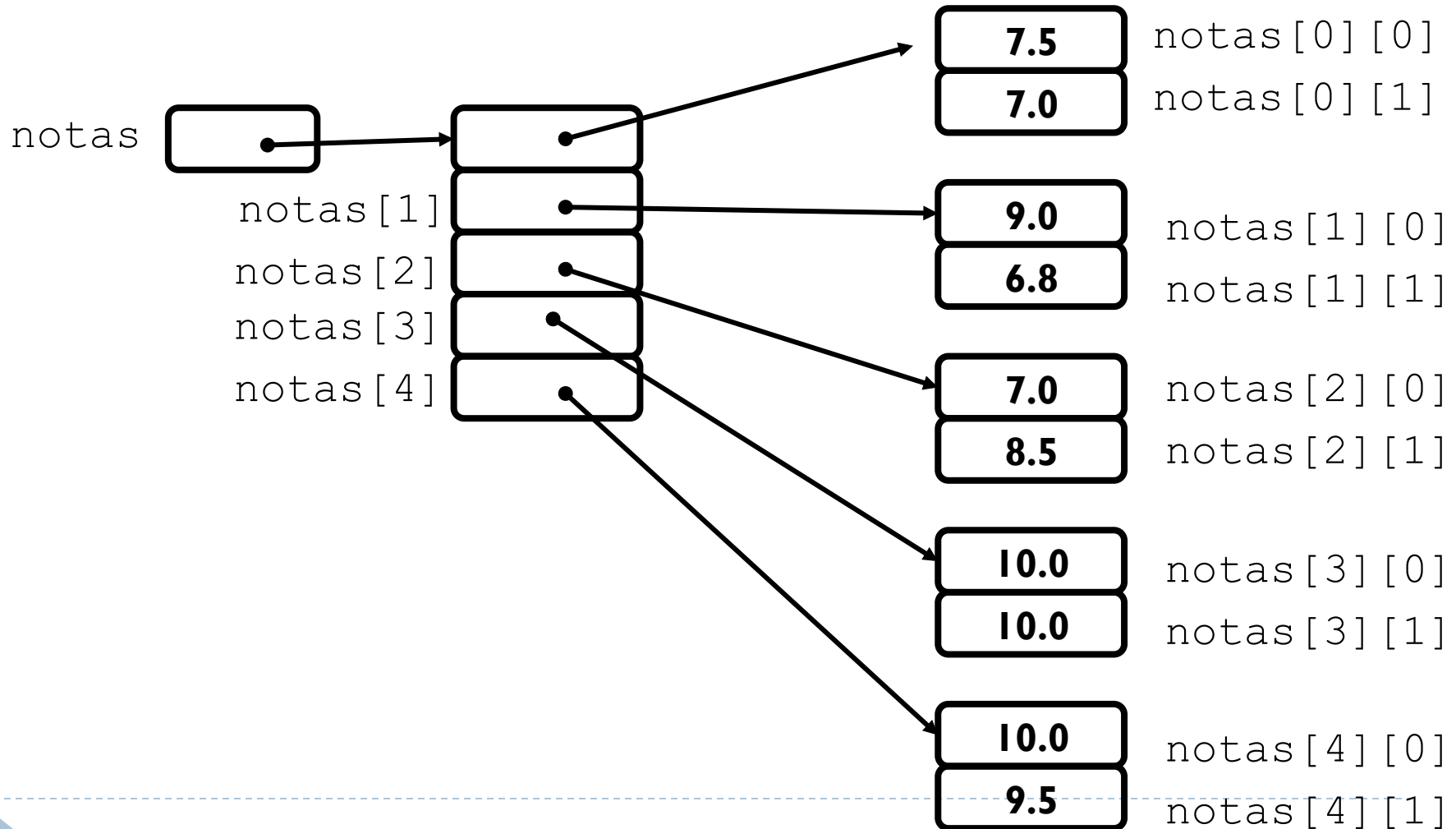
▶ Ex:

```
float notas[][] = new float[5][2];

for (int i = 0; i < 5; i++){
    for (int j = 0; j < 2; j++){
        notas[i][j] = ent.nextFloat();
    }
}
```

Arrays Multidimensionais

- São “arrays de arrays”



Arrays Multidimensionais

- ▶ Loops “for each” não fazem loop nos elementos de um array bidimensional automaticamente
 - ▶ Faz loop pelas **linhas** (que são **arrays unidimensionais**)
- ▶ Necessário dois loops aninhados;

```
float notas[][] = new float[5][2];  
...  
// Seleciona uma linha  
for (double []linha: notas){  
    // Percorre os elementos de uma linha  
    for (double value: linha){  
        System.out.println(value);  
    }  
}
```



Exercício 01

- ▶ **Suponha que se deseje processar um conjunto de valores representado altura e sexo (M/F) de um grupo de 10 pessoas. Escreva um programa em Java que:**
 - (a) Leia este conjunto de dados e armazene-o em dois arrays vinculados, um dos quais contém as alturas e o outro contém os sexos dos indivíduos.
 - (b) Determine a maior e a menor altura dentre esses indivíduos, indicando o sexo do indivíduo de maior altura e o sexo do indivíduo de menor altura.
 - (c) Encontre a média de altura entre os indivíduos do sexo feminino (representados no programa pelo caractere 'F') e a média de altura entre os indivíduos do sexo masculino (representados no programa pelo caractere 'M').
 - (d) Determine o número total de indivíduos de cada sexo.
-



Exercício 02

- ▶ **Escreva um programa em Java que solicita ao usuário para introduzir n valores inteiros, lê estes números e apresenta, logo antes de encerrar, o seguinte:**
 - (i) Todos os números introduzidos
 - (ii) O menor valor introduzido
 - (iii) O maior valor introduzido
 - (iv) A média dos valores introduzidos.

O valor n deve ser o primeiro dado introduzido pelo usuário, mas você deve estipular em seu programa um valor máximo para n (não use números mágicos).



Exercício 03

- ▶ Escreva um programa que lança um dado n vezes e imprime o percentual de surgimento de cada face do dado. O valor n é introduzido pelo usuário, sendo que 0 encerra o programa. Seu programa deverá utilizar um array para armazenar os números de aparecimento de cada face. Aproveite-se deste fato para evitar o uso de instruções condicionais (**switch ou if**) no **cálculo dos percentuais**.
- ▶ **Dica: O método `Math.random()` gera um número aleatório entre 0 e 1.**



Universidade Federal da Paraíba

Centro de Informática

Departamento de Informática

Linguagem de Programação I

Arrays em Java

▶ Tiago Maritan

▶ tiago@ci.ufpb.br

