

Vetor (Array)

Lógica de Programação

Prof. MSc. Bruno de A. Iizuka Moritani
bruno.moritani@anhembi.br



**Universidade
Anhembi Morumbi**

LAUREATE INTERNATIONAL UNIVERSITIES

Agenda

- Problema?
- Vetor
 - Definição
 - Declaração
 - Sintaxe
 - Exemplos
 - Inicialização de Vetores
 - Índices

Problema...

- Ler 5 notas informadas pelo usuário usando a estrutura de repetição “for”

```
1      import java.util.Scanner;
2
3      public class Exemplo_Slide03 {
4
5      public static void main(String[] args){
6          Scanner entrada = new Scanner(System.in);
7          for (int i = 0; i < 5; i++){
8              System.out.println("Digite a nota: ");
9              float nota = entrada.nextFloat();
10         }
11     }
12 }
```

- O que acontece com a 1ª nota após digitar as 5 notas?



Problema...

- Com o uso de estruturas de repetição,
 - é possível trabalhar com várias entradas de dados do mesmo tipo,
 - mas não é possível o armazenamento de todos esses dados de uma só vez!
- Como resolver esse problema?



Vetor (Array)

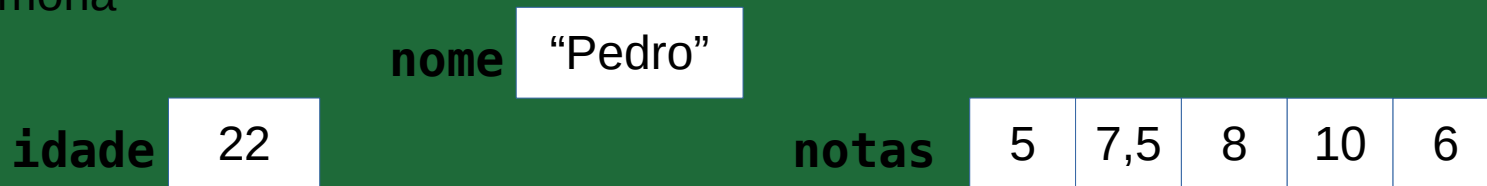


Universidade
Anhembi Morumbi
LAUREATE INTERNATIONAL UNIVERSITIES

Vetor - Definição

- É uma variável homogênea unidimensional
 - Um **conjunto** de variáveis de **mesmo tipo**, que possuem o **mesmo identificador** (nome) e são alocadas **sequencialmente** na memória

Memória



Declaração

- Um vetor é representado por:
 - Tipo
 - tipo de dado armazenado
 - `int`, `float`, `double`, `String`, `char`
 - Identificador
 - nome do vetor (plural)
 - Tamanho (dimensão)
 - quantidade de “elementos” (dados)

```
tipo identificador [ ] = new tipo[tamanho];
```

Declaração - Exemplos

```
double notas[ ] = new double[5];
```

Memória

notas

0	0	0	0	0
---	---	---	---	---

Declaração - Exemplos

```
int idades[ ] = new int[100];
```

Memória

idades

0	0	0	0	0	0	0	0	0	0	...	0	0
---	---	---	---	---	---	---	---	---	---	-----	---	---

Declaração - Exemplos

```
String cidades[ ] = new String[100];
```

Memória

cidades

""	""	""	""	""	""	""	""	""	""	...	""	""
----	----	----	----	----	----	----	----	----	----	-----	----	----

Declaração e Inicialização de Vetores

- Um vetor também pode ser criado a partir de uma lista de valores entre { e } e separados por virgula

```
int[] primos = { 2, 3, 5, 7, 11, 13, 17, 19 };  
char[] dias = { 'd', 's', 't', 'q', 'q', 's', 's' };  
String[] meses = { "jan", "fev", "mar", "abr",  
                   "mai", "jun", "jul", "ago",  
                   "set", "out", "nov", "dez" };
```

?????

OK!

Eu consigo armazenar uma
certa quantidade de valores...

E como que eu “ando”
dentro desse vetor?



Índice

- O índice indica a posição do dado no vetor
- Usado para manipular cada dado do vetor

```
double notas[ ] = new double[5];
```

Memória

notas

0	0	0	0	0
<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>

Índice

Índice

```
int idades[ ] = new int[100];
```

Memória

idades

18	32	24	25	45	67	72	22	18	19	...	35	32
0	1	2	3	4	5	6	7	8	9	...	98	99

1º elemento é sempre o
0 (Zero)

O **último elemento** é
sempre **tamanho - 1**

Índice

Memória

alturas

1.90	1.55	1.73	1.82	1.68	1.78	1.56	1.64	1.65	1.72	...	1.70	1.58
0	1	2	3	4	5	6	7	8	9	...	54	55

P1. Qual é o tamanho do vetor?

– 56

P2. Qual é o índice da 5a nota armazenada?

– 4

P3. Qual é o dado do índice 54?

– 1.70

P4. Qual é o 1o índice do vetor?

– 0

P5. Qual é o último índice do vetor?

– 55

Entrada de Dados

- Mas como entrar com dados nas posições do vetor?

```
double notas[ ] = new double[5];
```

Memória

notas

4.5	6.5	8.0	3.5	6.0
0	1	2	3	4

Índice

Índice

```
notas[0] = 4.5;  
notas[1] = 6.5;  
notas[2] = 8.0;  
notas[3] = 3.5;  
notas[4] = 6.0;
```


Saída de Dados

- Para imprimir os dados de um vetor, também é necessário ir posição por posição.

```
double notas[ ] = new double[5];
```

Memória

notas

4.5	6.5	8.0	3.5	6.0
0	1	2	3	4

Índice

```
System.out.println("Nota: " + notas[ 0 ]);  
System.out.println("Nota: " + notas[ 1 ]);  
System.out.println("Nota: " + notas[ 2 ]);  
System.out.println("Nota: " + notas[ 3 ]);  
System.out.println("Nota: " + notas[ 4 ]);
```

Nota: 4.5

Nota: 6.5

Nota: 8.0

Nota: 3.5

Nota: 6.0

CONSTRUÍDO COM SUCESSO

Percorrendo Dados no Vetor

- É necessário utilizar uma estrutura de repetição...

```
1  import java.util.Scanner;
2
3  ▶ public class Exemplo_Slide18 {
4
5  ▶   public static void main(String[] args){
6
7       Scanner entrada = new Scanner(System.in);
8       double notas[] = new double[5];
9
10      for (int i = 0 ; i < 5 ; i++){
11          System.out.println("Digite uma nota: ");
12          notas[i] = entrada.nextDouble();
13      }
14  }
15 }
```

Percorrendo Dados no Vetor

- Para consultar dados no vetor

```
1  ► public class Exemplo_Slide19 {  
2  
3  ►  public static void main(String[] args){  
4      double notas[] = {10, 9.5, 8, 7, 7.6};  
5      for (int i = 0 ; i < 5 ; i++){  
6          System.out.println("Nota [" + i + "]: " + notas[i]);  
7      }  
8  }  
9  }
```

Propriedade length

- Retorna um inteiro indicando o **tamanho** do vetor

```
1  ▶ public class Exemplo_Slide20 {  
2  
3  ▶  public static void main(String[] args){  
4  
5      double notas[] = new double[8000];  
6      int tamanho;  
7      tamanho = notas.length;  
8  
9      System.out.println("O tamanho do vetor eh... " + tamanho);  
10  
11  }  
12 }
```

Exemplo

1) Ler 10 notas, armazená-las em um vetor e imprimir as notas lidas.

```
1  import java.util.Scanner;
2
3  ▶ public class Exemplo_Slide21 {
4
5  ▶   public static void main(String[] args){
6
7       Scanner entrada = new Scanner(System.in);
8
9       //Declaracao do vetor com 10 elementos
10      double notas[] = new double[10];
11
12      // laço para leitura das 10 notas e armazenamento das notas no vetor
13      for (int i = 0 ; i < notas.length ; i++){
14          System.out.println("Digite uma nota: ");
15          notas[i] = entrada.nextDouble();
16      }
17
18      for (int i = 0 ; i < notas.length ; i++){
19          System.out.println("Nota [" + i + "]: " + notas[i]);
20      }
21
22  }
23  }
```

Exemplo

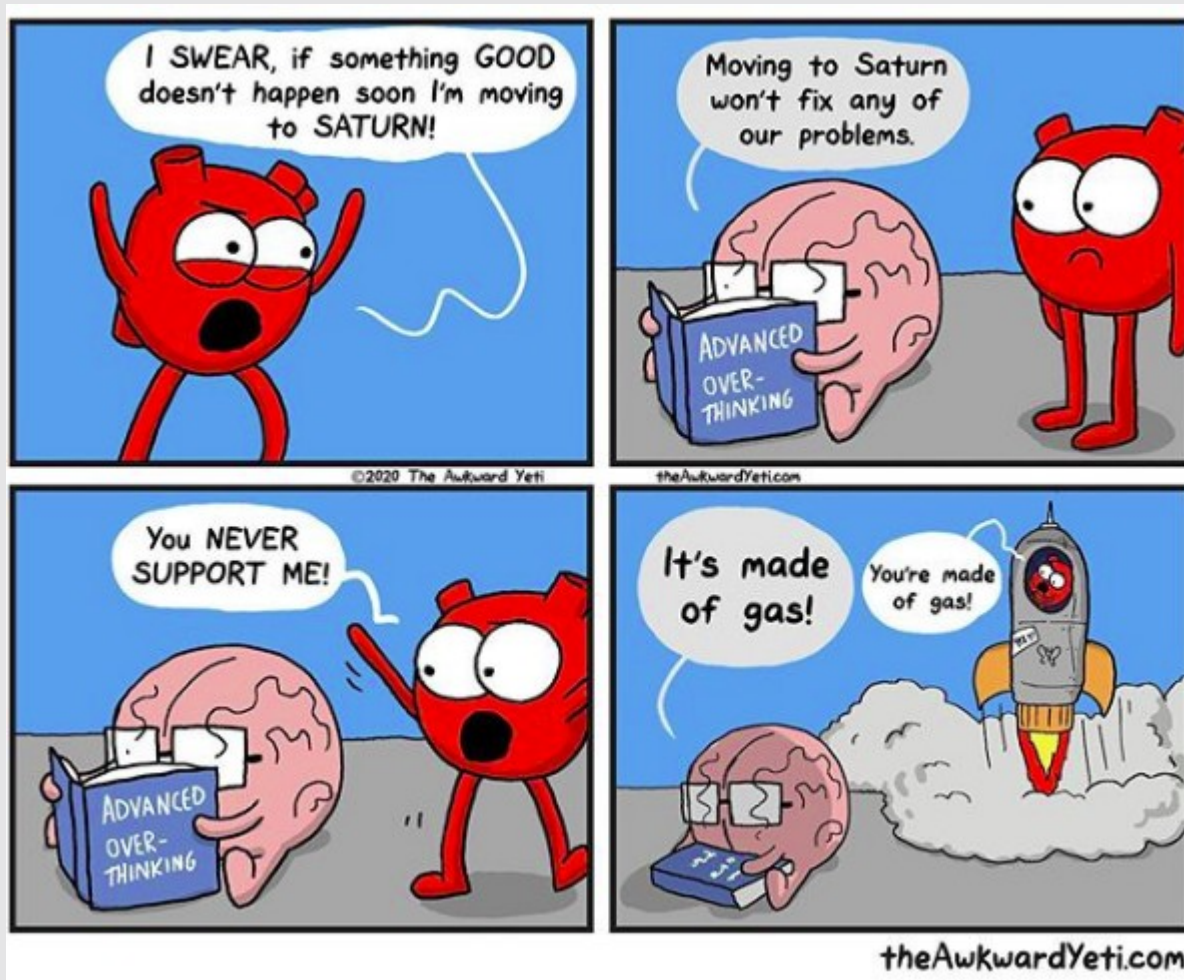
2) Encontrar o maior elemento em um vetor de 5 inteiros;

```
1  import java.util.Scanner;
2
3  ▶ public class Exemplo_Slide22 {
4
5  ▶   public static void main(String[] args){
6       Scanner entrada = new Scanner(System.in);
7       int maior;
8       //Declaracao do vetor com 5 numeros
9       int numeros[] = new int[5];
10      // laço para leitura das 5 numeros e armazenamento dos numeros no vetor
11      for (int i = 0 ; i < numeros.length ; i++){
12          System.out.println("Digite um numero: ");
13          numeros[i] = entrada.nextInt();
14      }
15      //atribuicao do 1o numero do vetor a variavel maior
16      maior = numeros[0];
17      //busca o maior numero
18      for (int i = 0 ; i < numeros.length ; i++){
19          if (numeros[i] > maior){
20              maior = numeros[i];
21          }
22      }
23      System.out.println("O maior numero eh... " + maior);
24  }
25  }
```

Dúvidas



Exercícios



Exercícios

- 1) Leia 10 valores inteiros e armazene-os em um Vetor. Em seguida, imprima todos os elementos.
- 2) Leia 15 nomes de pessoas e armazene-os em um Vetor. Em seguida, mostre todos os nomes em sequência contrária.
- 3) Leia 8 elementos em um vetor inteiro A. Construa um outro vetor B, de mesma dimensão (tamanho) de A, com seus elementos sendo a multiplicação dos elementos de A por 3. Mostre os elementos de B.

Exercícios

4) Leia dois vetores A e B inteiros, de mesma dimensão. Construa o vetor C, de mesma dimensão, cujo elementos de C é a subtração dos elementos de A por B. Mostre o vetor C.

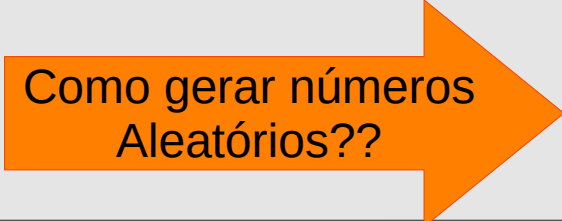
5) Leia dois vetores reais A (de tamanho 4) e B (de tamanho 7). Construa um 3º vetor que seja a junção de A e B (colocar os elementos de A seguidos dos elementos de B).

Desafios

6) Sortear 1000 números inteiros e armazenar em um vetor. Percorrer o vetor e encontrar o menor elemento.

7) Criar um vetor para armazenar 10.000 números inteiros gerados de forma aleatória (entre 1 e 10). Após gerar o vetor, calcular a quantidade de vezes que cada número foi sorteada.

- O número 1 aparece 5 vezes
- O número 2 aparece 2 vezes
-
- O número 10 aparece 3 vezes



Como gerar números
Aleatórios??

Classe Random

- Classe utilizada para gerar números aleatórios
- É necessário fazer o import

```
import java.util.Random;  
....  
Random random = new Random();
```

Gerando números inteiros

- O método **nextInt()** gera números inteiros

```
int sorteado;  
  
//sorteia um inteiro (dentro do range de inteiros)  
sorteado = random.nextInt();  
  
//sorteia um inteiro entre 0 e TAM-1  
sorteado = random.nextInt( 10 );  
  
/*sorteia um inteiro entre 0 e TAM-1 e soma 1 ao  
valor sorteado */  
sorteado = 1 + random.nextInt( 10 ); //entre 1 e 10
```

Gerando números Ponto Flutuante

- O **nextDouble()** – sorteia um valor entre 0 e 1.
- Esse método não aceita parâmetros!

```
double sorteado;  
  
//sorteia um número entre 0 e 1  
sorteado = random.nextDouble();  
  
//sorteia um número entre 10 e 100  
sorteado = 10 + random.nextDouble()*90;
```

.....

Obrigado!
Bom Dia!
Boa Noite!

.....

bruno.moritani@anhembi.br



**Universidade
Anhembi Morumbi**
LAUREATE INTERNATIONAL UNIVERSITIES