

# Vetores (*arrays*) – Parte 1

Prof. Bruno Nogueira

# Até agora...

---

- ▶ Vimos variáveis que armazenam um único valor de um determinado tipo
  - ▶ Uma variável `int` armazena apenas um valor inteiro
  - ▶ Uma variável `String` armazena apenas uma cadeia de caracteres
  - ▶ Uma variável `double` armazena apenas um valor real
  - ▶ ...
- ▶ Entretanto, em alguns casos, é comum querermos trabalhar com **coleções** de valores de um mesmo tipo
  - ▶ Notas de uma turma
  - ▶ Temperatura nos dias de uma semana
  - ▶ Pontuação de equipes em uma competição
  - ▶ ...

# Vetores

---

- ▶ São tipos especiais de dados usados para armazenar uma coleção de dados
- ▶ Todos os dados armazenados em um determinado vetor são de um mesmo tipo
  - ▶ Por exemplo, um vetor de *int* só pode armazenar valores do tipo inteiro; um vetor de *double* só pode armazenar valores *double*; ...
- ▶ São comumente percorridos e manipulados em loops, especialmente em loops do tipo `for`

# Problema de armazenar coleções

---

- ▶ Considere o problema de receber as datas de um dia da semana e calcular a média destas

```
Scanner teclado = new Scanner (System.in);  
double soma = 0, proximaTemp, media;  
System.out.println("Entre 7 temperaturas");  
for(int i = 0; i < 7; i++) {  
    proximaTemp = teclado.nextDouble();  
    soma = soma + proximaTemp;  
}  
media = soma / 7;
```

- ▶ Resolve o problema da média. Mas e se quisermos armazenar as 7 datas digitadas pelo usuário?
  - ▶ Declarar 7 variáveis e colocar 7 if's para controlar o contador
  - ▶ E se quisermos as temperaturas anuais? 365 variáveis? Inviável!

# Solução: vetores

---

- ▶ Com vetores, conseguimos armazenar em uma mesma variável uma coleção de valores de um mesmo tipo
- ▶ Sintaxe da declaração:

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

- ▶ Exemplos:

- ▶ Vetor com 7 elementos do tipo double:

```
double [] temperatura = new double[7];
```

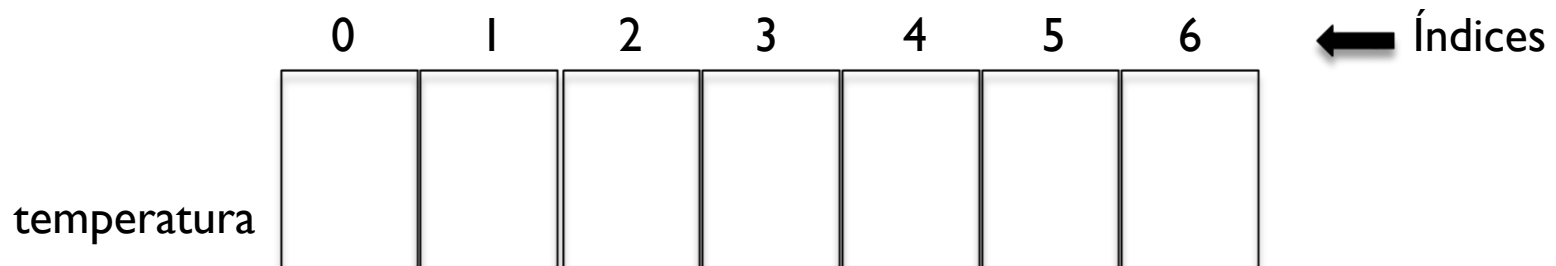
- ▶ Vetor com 80 elementos do tipo char

```
char[] simbolos = new char[80];
```

# Acessando elementos de um vetor

- ▶ Quando criamos um vetor de  $n$  posições, criamos  $n$  variáveis independentes que são armazenadas nesses vetores
- ▶ Por exemplo:

```
double [] temperatura = new double[7];
```



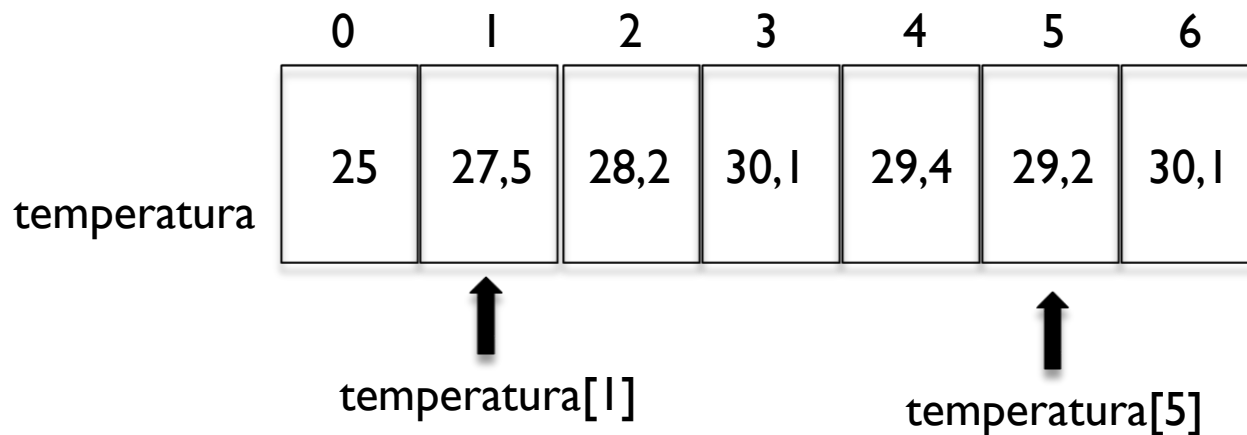
7 variáveis, acessíveis por:

temperatura[0] temperatura[1] temperatura[2] temperatura[3] ... temperatura[6]

# Acessando os elementos de um vetor

---

- ▶ Os índices de um vetor em Java **sempre** começam com 0. Logo, com um vetor de  $n$  posições, os índices válidos são número inteiros no intervalo  $0 \dots n-1$



```

1  import java.util.Scanner;
2  public class VetorTemperaturas
3  {
4      public static void main (String [] args)
5      {
6          Scanner teclado = new Scanner (System.in);
7          double[] temperatura = new double [7];
8          double soma = 0, media;
9
10         System.out.println("Digite 7 temperaturas:");
11         for(int index = 0; index < 7; index ++)
12         {
13             temperatura[index] = teclado.nextDouble();
14             soma = soma + temperatura[index];
15         }
16         media = soma / 7;
17         System.out.println("A média das temperaturas é: " + media);
18
19         System.out.println("As temperaturas são: ");
20         for(int index = 0; index < 7; index++)
21         {
22             if(temperatura[index] < media)
23                 System.out.println(temperatura[index] + " abaixo da média");
24             else if(temperatura[index] > media)
25                 System.out.println(temperatura[index] + " acima da média");
26             else
27                 System.out.println(temperatura[index] + " na média");
28         }
29     }
30 }

```



```

1  import java.util.Scanner;
2  public class VetorTemperaturas
3  {
4      public static void main (String [] args)
5      {
6          Scanner teclado = new Scanner (System.in);
7          double[] temperatura = new double [7];
8          double soma = 0, media;
9
10         System.out.println("Digite 7 temperaturas:");
11         for(int index = 0; index < 7; index ++)
12         {
13             temperatura[index] = teclado.nextDouble();
14             soma = soma + temperatura[index];
15         }
16         media = soma / 7;
17         System.out.println("A média das temperaturas é: " + media);
18
19         System.out.println("As temperaturas são: ");
20         for(int index = 0; index < 7; index++)
21         {
22             if(temperatura[index] < media)
23                 System.out.println(temperatura[index] + " abaixo da média");
24             else if(temperatura[index] > media)
25                 System.out.println(temperatura[index] + " acima da média");
26             else
27                 System.out.println(temperatura[index] + " na média");
28         }
29     }
30 }

```

Instanciação

Atribuição

Acesso

# Como manipular

---

- ▶ Declaração de uma variável

```
double [] temperatura;
```

- ▶ Criando o vetor

```
temperatura = new double [7];
```

- ▶ Atribuindo valor a uma posição do vetor

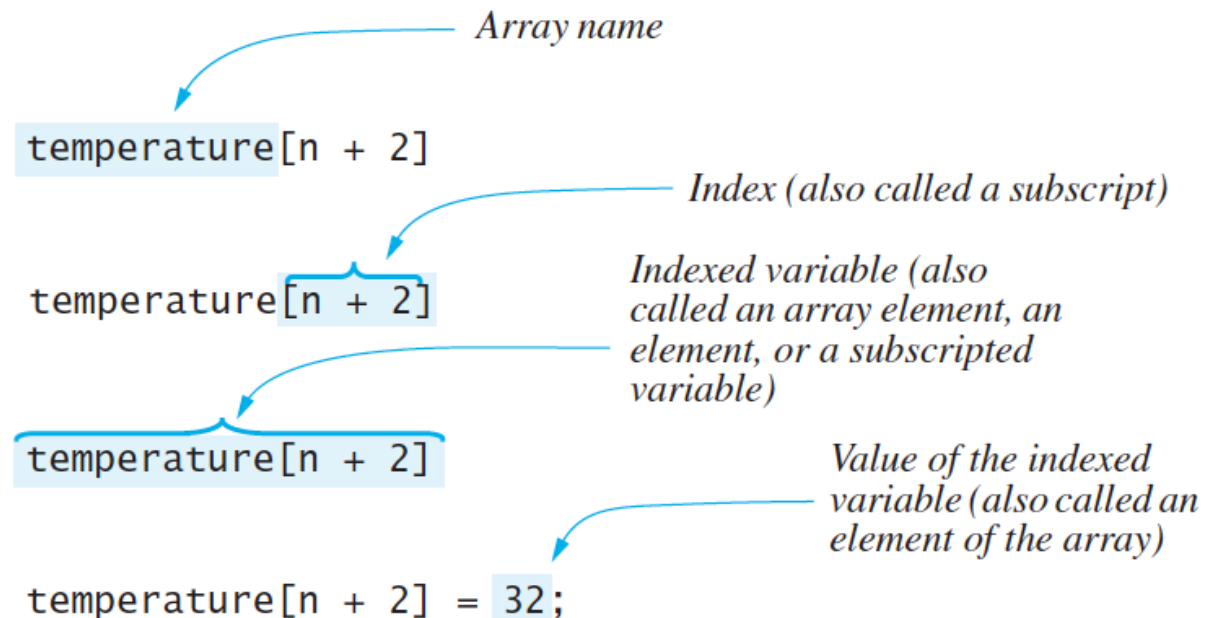
```
temperatura[2] = 5.4;
```

- ▶ Acessando o valor de uma posição do vetor

```
double temp = temperatura[2] + 3;
```

# Alguns detalhes

- ▶ Importante lembrar: vetores são de tipos diferentes dos tipos simples que armazenam
  - ▶ `double[]` é diferente de `double`; `int[]` é diferente de `int`; ...
- ▶ Nomenclatura usual:



# Tamanho de um vetor

---

- ▶ O tipo de dados vetor, em Java, tem um atributo especial que guarda o número de posições do vetor
  - ▶ Atributo length (não usa parênteses)
  - ▶ Retorna o número de posições disponíveis. É diferente do número de posições já preenchidas

```
double [] temperatura = new double [7];  
int tamanhoVetor = temperatura.length;
```

- ▶ Variável tamanhoVetor tem valor 7, nesse exemplo
- ▶ É o número máximo de valores que o vetor pode armazenar

```

1  import java.util.Scanner;
2  public class VetorTemperaturasV2
3  {
4      public static void main (String [] args)
5      {
6          Scanner teclado = new Scanner (System.in);
7          double[] temperatura;
8          double soma = 0, media;
9          int numTemperaturas;
10
11          System.out.println("Quantas temperaturas você tem disponíveis?");
12          numTemperaturas = teclado.nextInt();
13
14          temperatura = new double [numTemperaturas];
15
16          System.out.println("Digite " + numTemperaturas + " temperaturas:");
17          for(int index = 0; index < temperatura.length; index ++)
18          {
19              temperatura[index] = teclado.nextDouble();
20              soma = soma + temperatura[index];
21          }
22          media = soma / temperatura.length;
23          System.out.println("A média das temperaturas é: " + media);
24
25          System.out.println("As temperaturas são: ");
26          for(int index = 0; index < temperatura.length; index++)
27          {
28              if(temperatura[index] < media)
29                  System.out.println(temperatura[index] + " abaixo da média");
30              else if(temperatura[index] > media)
31                  System.out.println(temperatura[index] + " acima da média");
32              else
33                  System.out.println(temperatura[index] + " na média");
34          }
35      }
36  }

```

```

1 import java.util.Scanner;
2 public class VetorTemperaturasV2
3 {
4     public static void main (String [] args)
5     {
6         Scanner teclado = new Scanner (System.in);
7         double[] temperatura;
8         double soma = 0, media;
9         int numTemperaturas;
10
11         System.out.println("Quantas temperaturas você tem disponíveis?");
12         numTemperaturas = teclado.nextInt();
13
14         temperatura = new double [numTemperaturas];
15
16         System.out.println("Digite " + numTemperaturas + " temperaturas:");
17         for(int index = 0; index < temperatura.length; index++)
18         {
19             temperatura[index] = teclado.nextDouble();
20             soma = soma + temperatura[index];
21         }
22         media = soma / temperatura.length;
23         System.out.println("A média das temperaturas é: " + media);
24
25         System.out.println("As temperaturas são: ");
26         for(int index = 0; index < temperatura.length; index++)
27         {
28             if(temperatura[index] < media)
29                 System.out.println(temperatura[index] + " abaixo da média");
30             else if(temperatura[index] > media)
31                 System.out.println(temperatura[index] + " acima da média");
32             else
33                 System.out.println(temperatura[index] + " na média");
34         }
35     }
36 }

```

# Índice fora do intervalo

---

- ▶ Usar um índice menor que 0 ou maior que *length - 1* para acessar ou atribuir valor a uma posição do vetor gera erro em tempo de execução (não de compilação)
  - ▶ Erro do tipo *ArrayIndexOutOfBoundsException*
  - ▶ É mais comum do que se imagina. Geralmente casado pelo erro de “Fora por um”, comentado quando vimos loops
  - ▶ Precisa ser corrigido no código e depois recompilado

```
double [] temperatura = new double [7];  
temperatura[7] = 5.6;
```

ERRO! Índices válidos estão no intervalo de 0 a 6

# Uso de = e == com vetores

- ▶ Com vetores, o uso da = e da == entre variáveis têm algumas especificidades

```
int [] a = new int [3];  
int [] b = new int [3];  
  
for(int i = 0; i < a.length; i++)  
    a[i] = i;  
  
for(int i = 0; i < b.length; i++)  
    b[i] = i;  
  
if(b==a)  
    System.out.println("São iguais");  
else  
    System.out.println("São diferentes");
```

a e b são vetores de 3 posições

elementos de a e b são  
preenchidos com o valor de  
i

Testa se os endereços de  
memória de a e b são os  
mesmos

Resp: São diferentes



# Uso de = e == com vetores

---

- ▶ Com vetores, o uso da = e da == entre variáveis têm algumas especificidades

```
int [] a = new int [3];  
int [] b = new int [5];  
  
for(int i = 0; i < a.length; i++)  
    a[i] = i;  
  
b = a;  
  
if(b==a)  
    System.out.println("São iguais");  
else  
    System.out.println("São diferentes");
```

Não faz uma cópia de a, mas sim associa à mesma variável a o nome b.

Testa se os endereços de memória de a e b são os mesmos

Resp: São iguais

# Uso de = e == com vetores

---

- ▶ Com vetores, o uso da = e da == entre variáveis têm algumas especificidades

```
int [] a = new int [3];  
int [] b = new int [3];  
  
for(int i = 0; i < a.length; i++)  
    a[i] = i;  
  
b = a;  
  
System.out.println(a[2] + " " + b[2]);  
a[2] = 10;  
  
System.out.println(a[2] + " " + b[2]);
```

Uma mudança de valor em a implica em uma mudança em b (são as mesmas posições de memória)

Saída:  
2 2  
10 10

# Dicas

---

- ▶ Use nomes no singular para nomes de vetores
  - ▶ Opte por *simbolo*, *temperatura*, ... , ao invés de *simbolos*, *temperaturas*
  - ▶ Embora um vetor tenha vários valores, o uso comum de um vetor é para se referir a um único valor (*temperatura[indice]*)
- ▶ Não conte com os valores iniciais de um array
  - ▶ Por padrão, todas as posições são inicializadas com valores padrão: `int[ ]` com 0, `boolean[ ]` com `false`, ...
  - ▶ Boa prática envolve o uso de um loop para inicializar todas as posições, como fizemos nos últimos slides
- ▶ Prefira o uso de `length` e `for` para navegar por vetores
  - ▶ Ajuda a evitar exceções de índice inválido

# Exercício

---

- ▶ Faça um programa em Java que preencha um vetor com 5 valores inteiros em qualquer ordem. Depois, seu programa deve percorrer o vetor, encontrar e imprimir na tela:
  - ▶ O menor valor
  - ▶ O maior valor
  - ▶ A média de todos os valores

## Exercício II

---

- ▶ Faça um programa Java que receba do usuário uma lista de  $n$  valores inteiros, em qualquer ordem. Seu programa deve gerar um outro vetor, com esses valores ordenados, e imprimir o resultado na tela.