

# Arrays

Prof. Ms. Peter Jandl Junior

Prof. Ms. Télvio Orru

Prof. Nathan Silva

J12B

Linguagem de Programação Orientada a Objetos

Ciência da Computação - UNIP – Jundiaí

# Linguagem Java

## conceitos básicos

Um tour pelos novos recursos

# Linguagem Java :: conceitos básicos

- Esta apresentação mostra as características da plataforma de programação Java e os recursos básicos disponíveis na linguagem de mesmo nome.

Variáveis

Arrays

Exemplos

Exercícios

# Java:Variáveis

# Variáveis

- São posições de memória nas quais é possível armazenar valores ou objetos.
- Ao invés de conhecer os endereços de memória, são usados nomes (ou identificadores) que simplificam seu uso.
- Em qualquer lugar de um programa, o uso do nome de uma variável equivale ao uso do valor que ela contém.
- Para definir/alterar o valor de uma variável, usamos a operação de atribuição (= no Java).

# Declaração de Variáveis

- Sintaxe:
- Tipo nome1 [, nome2 [, nome3 [..., nomeN]]];
- Exemplos:
  - int i;
  - float total, preco;
  - byte mediaGrupoTarefa2;
  - double valorMedio;

# Valores literais

- São expressos de modo direto no código-fonte:
  - `int x = 12;`
  - `long b = 100200300;`
  - `double pi_2 = 1.57;`
  - `String lp = "Java";`
  - `boolean flag = true;`
- Literais integrais são do tipo **int**.
- Sufixo **L** usado para indicar literais **long**.
- Literais reais são do tipo **double**.
- Sufixo **f** usado para indicar literais **float**.

# Java:Arrays



# Arrays

- Também conhecidos como arranjos.
- Estruturas homogêneas de dados, ou seja, destinadas ao armazenamento de um ou mais elementos do mesmo tipo.
- Utilizam um bloco contíguo de memória, i.e., criado com uma única operação de alocação de memória; possibilitando que seus elementos sejam organizados em posições sucessivas e de igual tamanho.
- Seus elementos podem acessados, para leitura ou escrita, através de um índice inteiro.

# Arrays

- Os índices são valores inteiros que indicam qual é o elemento desejado.
- Como nas linguagem C ,C++ e C#:
  - primeiro elemento é armazenado sob índice **zero**;
  - segundo sob índice **um**; e assim sucessivamente,
  - até que último índice seja **tamanho** do arranjo - **1**.



# Arrays

- No Java os arrays:
  - São objetos:
    - É necessário efetuar a alocação dinâmica dos arranjos antes de seu uso;
  - São automaticamente "zerados" ao serem criados;
  - Possuem uma propriedade ***length*** que indica seu tamanho, ou seja, que indica o número de elementos do arranjo; e
  - É impossível utilizar índices inválidos (não inteiros ou menores que zero ou maiores que ***length*** - 1).

# Declaração de Arrays

- Sintaxe:

<tipo> nome[ ];

- Exemplo:

int **v**[ ];

// tipo primitivo

double **x**[ ];

// tipo primitivo

String **nome**[ ];

// tipo Objeto

Object **coisas**[ ];

# Alocação de Arrays

Operador de criação de objetos (alocação de memória).

- Dado um arranjo já declarado:  
nome = **new** <tipo> [ num\_elementos];
- Exemplos:  
**v** = new int [10]; // como em C++  
**d** = new double [20];  
**nome** = new String [45];  
**coisas** = new Object [100];
- Observe que o número de elementos pode ser uma variável ou expressão (de tipo **int**).
- Todo array possui uma propriedade **length** que indica seu tamanho!

# Declaração e Alocação Simultâneas

- Também é possível fazer:

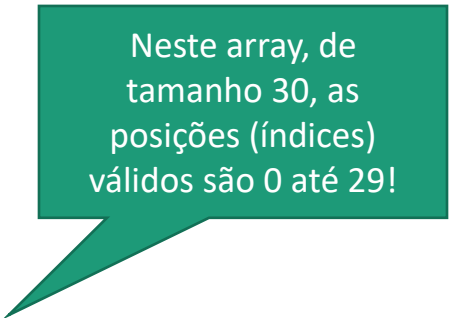
```
int v[ ] = new int [10];  
double x[ ] = new double [20];  
String nome[ ] = new String [45];  
Object coisas[ ] = new Object [100];
```

# Declaração e Inicialização Simultâneas

- Além disso também é possível fazer:  
`int v[ ] = {5, 13, -2, 2034, -192, 0, 10};`  
`double x[ ] = {1.4};` // só um também pode  
`String nome[ ] = { "Pedro", "Lucas", "Matheus" };`  
`Object coisas[ ] = {           new Object(),`  
`new Object(),`  
`objetoExistente   };`
- Nestes casos o compilador efetua a alocação e atribuição necessária para os elementos dados.

# Atribuição de Valor

- Considere um array declarado como:  
`double t[] = new double[30];`
- Por padrão, o array estará preenchido com valores zeros.
- Em cada posição válida podemos atribuir um valor:  
`t[0] = 123.45;`  
`t[1]= 75.31;`  
`:`  
`t[28]= 0.00001;`  
`t[29]= 13;`



Neste array, de tamanho 30, as posições (índices) válidos são 0 até 29!



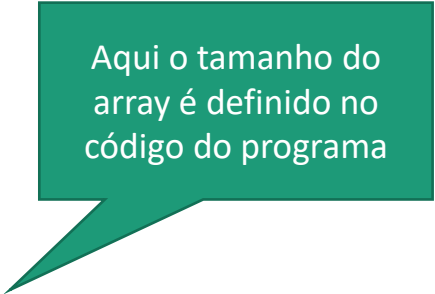
# Uso dos valores

- A combinação do nome do array com um índice, como: `t[1]`, `v[33]`, `lista[8]` equivale ao uso de uma variável individual, ou seja, o valor contido naquela posição é usado no local onde se indexa um array!
- Então, um array de 10 valores inteiros equivale a 10 variáveis do tipo inteiro que tem o mesmo nome e são diferenciadas por meio do índice (ou seja, da posição do valor dentro do array).

# Exemplos

# Exemplo 1

```
public class Arranjo1 {  
    public static void main(String a[]) {  
        //  
        //  
        //  
        int tam =10;  
        // declara e aloca array  
        int v[ ] = new int [tam];  
        // inicia array com inteiros  
        for(int i=0; i<v.length; i++) {  
            // atribui a própria posição como conteúdo  
            v[i] = i;  
        }  
        // exibe array  
        for(int i=0; i<v.length; i++) {  
            System.out.println("v[" + i + "] = " + v[i]);  
        }  
    }  
}
```



Aqui o tamanho do array é definido no código do programa

# Exemplo 2

```
public class Arranjo2 {  
    public static void main(String a[]) {  
        // Lê tamanho com Scanner  
        Scanner teclado = new Scanner(System.in);  
        System.out.print("Digite o tamanho do array: ");  
        int tam = teclado.nextInt();  
        // declara e aloca array  
        int v[ ] = new int [tam];  
        // inicia array com inteiros  
        for(int i=0; i<v.length; i++) {  
            // atribui a própria posição como conteúdo  
            v[i] = i;  
        }  
        // exibe array  
        for(int i=0; i<v.length; i++) {  
            System.out.println("v[" + i + "] = " + v[i]);  
        }  
    }  
}
```

Aqui o tamanho do array é lido pelo programa

# Exemplo 3

```
public class Arranjo3 {  
    public static void main(String a[]) {  
        // Lê tamanho com Scanner  
        Scanner teclado = new Scanner(System.in);  
        System.out.print("Digite o tamanho do array: ");  
        int tam = teclado.nextInt();  
        // declara e aloca array  
        int v[ ] = new int [tam];  
        // inicia array com inteiros  
        for(int i=0; i<v.length; i++) {  
            System.out.print("Valor " + i + "? ");  
            v[i] = teclado.nextInt(); // valor dado pelo usuário  
        }  
        // exibe array  
        for(int i=0; i<v.length; i++) {  
            System.out.println("v[" + i + "] = " + v[i]);  
        }  
    }  
}
```

Aqui o tamanho do array é lido pelo programa

Aqui os valores do array também são lido pelo programa

# Exemplo 4

```
public class Arranjo4 {  
    public static void main(String a[]) {  
        // Lê tamanho com Scanner  
        Scanner teclado = new Scanner(System.in);  
        System.out.print("Digite o tamanho do array: ");  
        int tam = teclado.nextInt();  
        // declara e aloca array  
        int v[ ] = new int [tam];  
        // inicia array com inteiros  
        for(int i=0; i<v.length; i++) {  
            System.out.print("Valor " + i + "? ");  
            v[i] = teclado.nextInt(); // valor dado pelo usuário  
        }  
        // soma array  
        int soma = 0;  
        for(int i=0; i<v.length; i++) {  
            soma = soma + v[i];  
        }  
        System.out.println("Soma: " + soma);  
    }  
}
```

Aqui o tamanho do array é lido pelo programa

Aqui os valores do array também são lidos pelo programa

Aqui os valores do array são somados

# Exercícios

# Exercícios

1. Escreva um programa que leia um arranjo de 10 elementos **inteiros**, exibindo seus valores, sua soma e também sua média.
2. Escreva um programa que leia um arranjo de 20 elementos **reais**, exibindo seus valores, sua soma e também sua média.
3. Escreva um programa que leia 2 arrays A e B, de valores **reais** e de tamanho 15. Com uso de um 3o arranjo, some os arranjos, posição à posição ( $C[0] = A[0] + B[0]$ ,  $C[1] = A[1] + B[1]$  ...), exibindo o array resultante.



# Exercícios

4. Escreva um programa que leia um arranjo de 10 elementos **inteiros**, identificando e exibindo o maior valor presente no arranjo.
5. Escreva um programa que leia um arranjo de 10 elementos **inteiros**, identificando e exibindo o menor valor presente no arranjo e sua posição.
6. Construa um programa que crie um arranjo de **String** com o tamanho indicado pelo usuário (será preciso ler tal valor inteiro em seu início). Preencha o arranjo com Strings dadas pelo usuário, exibindo o arranjo ao final.

# Exercícios

7. Escreva um programa que leia um arranjo com 20 valores **reais** e apresente em separado a soma dos valores positivos e negativos.
8. Escreva um programa que leia um arranjo com 20 valores **caracteres** e apresente em quais posições existem vogais.
9. Escreva um programa que leia um número N **inteiro**, criando um arranjo de tipo **long** com o tamanho do valor dado e preenchido com potências de 2:  $2^0$ ,  $2^1$ ,  $2^2$ ,  $2^3$ ,  $2^4$  ...  $2^N$

# Recomendações de Estudo



- Resolver a Lista I.
- Complementar estudo com:
  - JANDL JR, Peter.  
**Java – Guia do Programador**, 3ª Ed.  
São Paulo: Novatec,  
2015.