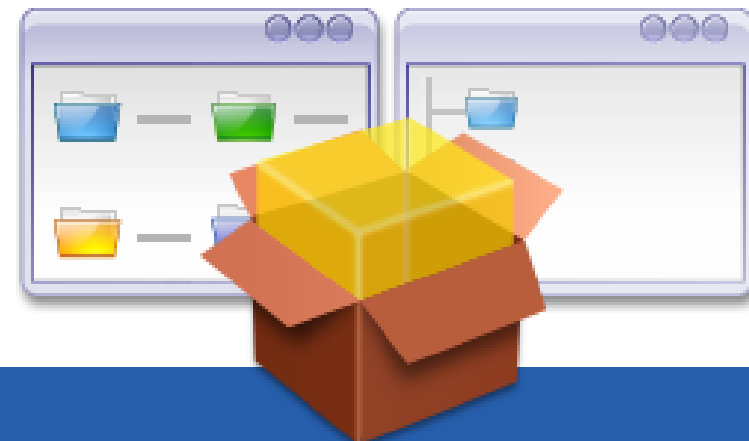


Programação Orientada a Objetos

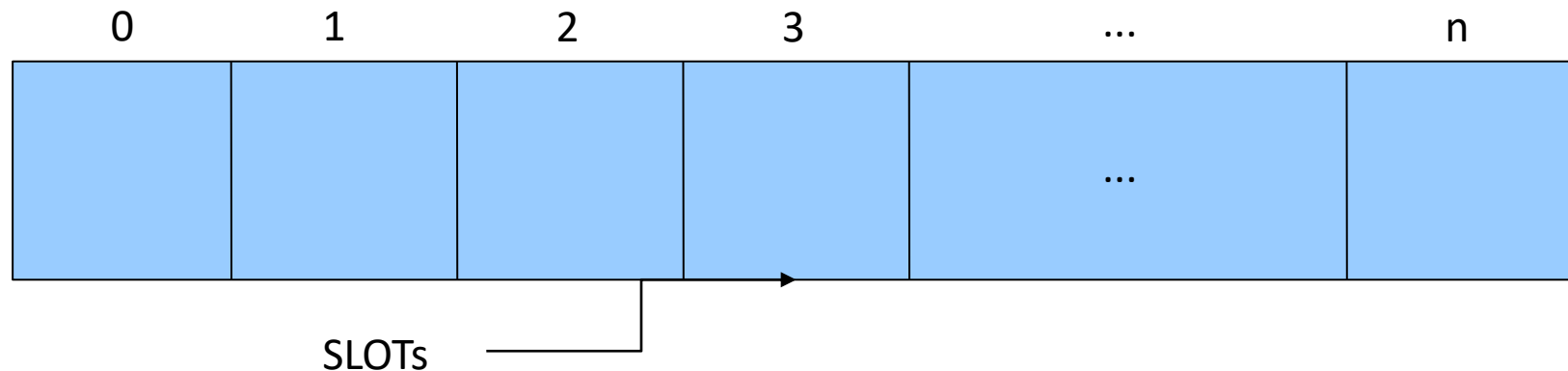
Vetores

Prof. Tito Kenzo



Array

- Em Java é uma coleção de tipos primitivos, referências ou outros Arrays.
- São homogêneos: Só é possível declarar Arrays do mesmo tipo.
- É um “lista” formada por slots numerados para facilitar o acesso a cada item armazenado;



Array

- Você pode guardar qualquer tipo de dado dentro de um vetor;
- Java implementa um array como um objeto;
- Para criação de de um Array, três passos são necessários:
 - **Declaração:** declare a variável como um array;
 - **Construção ou Criação:** crie o objeto do mesmo tipo da variável;
 - **Inicialização:** armazene a informação no Array;

Array: Declaração

- Para declarar uma variável, diga seu tipo e acrescente os colchetes [] (colchetes):

```
String[] nomes;           // Array do tipo referência
int[] numeros;            // Array de inteiros
Funcionario[] fun;        // Array do tipo referência
int inteiros[];           // Array de tipo primitivo
int[] array;              // Array de inteiros
Figura[] f;               // Array do tipo referência
Usuario usu[];            // Array do tipo referência
double salario[];         // Array de tipo primitivo
```

Array: Construção ou Criação

- Como um array em Java é um objeto, você precisa criar a instância desse objeto;
- Usa-se o **new** para criar um Array:

```
int inteiros[] = new int[5];
```

```
String nomes[] = new String[10];
```

```
int []matriculas = new int[10];
```

```
Funcionario[] funcionarios = new Funcionario[10];
```

Array: Inicialização

- Quando um Array é criado, todos os seus elementos são inicializados com valores default.
- Você pode já na declaração inicializar o Array com valores específicos.

- `int[] numeros = {11, 15, 2, 7, 122, 34};`

```
int inteiros[] = {
```

**Só pode ser feito na
declaração do Array**

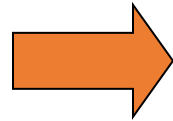
10,

8,

67,

1,

100};



```
inteiros[] = new int[5];  
inteiros[0] = 10;  
inteiros[1] = 8;  
inteiros[2] = 67;  
inteiros[3] = 1;  
inteiros[4] = 100;
```

Array: Inicialização

- Para armazenar uma informação no slot de um array, você precisa acessar a posição correta:

```
int[] matriculas = new int[10];  
matriculas[0] = 1111;  
matriculas[1] = 2222;  
matriculas[2] = 3333;
```

Array: Inicialização

- Elementos de Arrays do tipo referência precisam ser instanciados:

```
Ponto p[] = new Ponto[2]; // crio o vetor  
p[0] = new Ponto();           //inicializo p[0]  
p[1] = new Ponto (10,15);      //inicializo p[1]
```

```
Funcionario[] funcionarios = new Funcionario[20];  
funcionarios[1] = new Funcionario("James");
```


Array Multidimensional

- Em Java, Arrays multi-dimensionais são implementados através de Arrays de Arrays.
- Os Arrays não precisam ser retangulares.
- Exemplo:

```
int duasDime[][] = new int[45][]  
duasDime[0] = new int[15];  
duasDime[1] = new int[5];
```

```
int doisDime = new int[][45]; //illegal  
int triDime = new int[10][20][5]
```

Array: Limites

- Em Java, índices de Array começam com 0.
- Não é possível acessar além dos limites de um Array.
- Para saber o tamanho de uma Array use o atributo length.
- Exemplo:

```
int arr[] = new int[10]
for (int i = 0; i < arr.length; i++)
{
    System.out.println(arr[i]);
}
```

Tarefinha

1. **Escreva uma classe chamada ArrayTeste1.**
2. **No método main declare duas variáveis arrayUm e arrayDois como Arrays de int.**
3. **Instancie arrayUm com capacidade para 10 elementos e inicialize-o com valores de 1 a 10.**
4. **Atribua a referência de arrayUm para arrayDois e imprima seus valores.**
5. **Altere o quinto elemento de arrayDois para o dobro do seu valor e imprima arrayUm.**

Quais valores serão impressos de arrayUm?

Tarefinha

- 1. Escreva uma classe chamada CopiaArray.**
- 2. No método main declare duas variáveis arrayOrigem e arrayDestino como Arrays de int.**
- 3. Instancie arrayOrigem com capacidade para 100 elementos e inicialize-o com valores de 1 a 100.**
- 4. Copie todos os elementos de arrayOrigem para arrayDestino.**
- 5. Apague todos os números ímpares do arrayOrigem.**
- 6. Imprima os elementos de arrayDestino.**