



C206/C06 – Programação Orientada a Objetos com Java



Arrays no Java

Prof. Christopher Lima
christopher@inatel.br



Objetivos

☕ Entender o funcionamento de *Arrays* (vetores) no Java

☕ Popular e percorrer *Arrays*



Arrays

- ☕ Coleção de **um ou mais objetos**, do **mesmo tipo**, armazenado em **endereços adjacentes** de memória.
- ☕ Cada objeto é chamado de **“elemento do array”**.



Arrays no Java

☕ Como declarar um Array no Java?

```
//Declarando uma variável  
//que irá guardar a referência  
//para um array  
int []x;  
int[] y;  
int z[];
```



Arrays no Java

☕ Como inicializar um Array no Java?

```
//Declarando uma variável  
//que irá guardar a referência  
//para um array  
int[] x = new int[10]; //Array com 10 posições  
int []y = new int[]; //NÃO COMPILA  
int z[] = {1,2,3,4}; //Array com 4 posicoes inicializado  
//com os valores 1,2,3 e 4
```



Arrays no Java

☕ Assim como C/C++/C#, o primeiro **índice do array começa na posição 0 (zero)**, indo até **n-1**, onde n é o tamanho do array definido no momento de sua criação!

```
String [] professores = new String[4];  
professores[0] = "Guilherme";  
professores[1] = "Marcelo"; // Acessando o elemento na posição 2  
professores[2] = "Soned";  
professores[3] = "RenZo"; // Acessando o elemento na posição 4
```

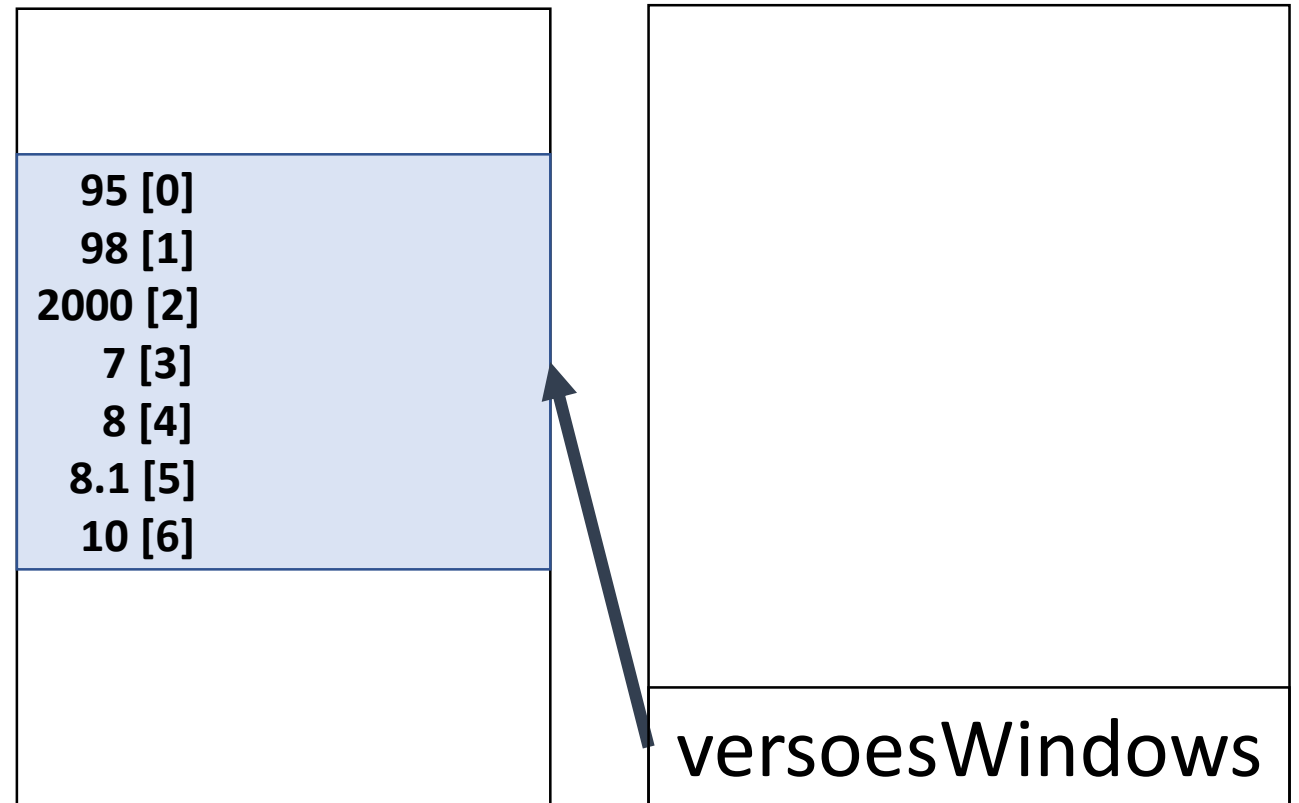
```
//Array de pontos flutuantes (dupla precisao)
```

```
double []versoesWindows = {95,98,2000,7,8,8.1,10};
```



Heap

Stack

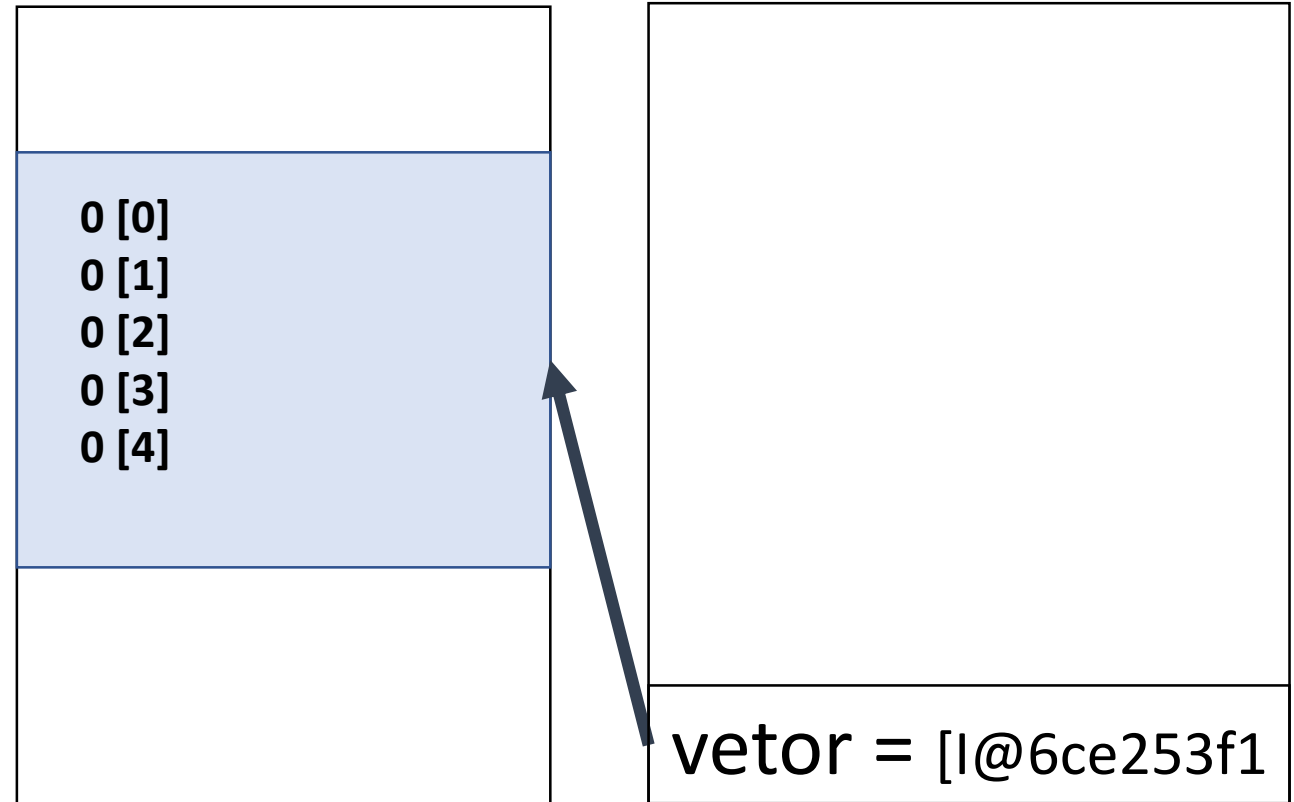




Heap

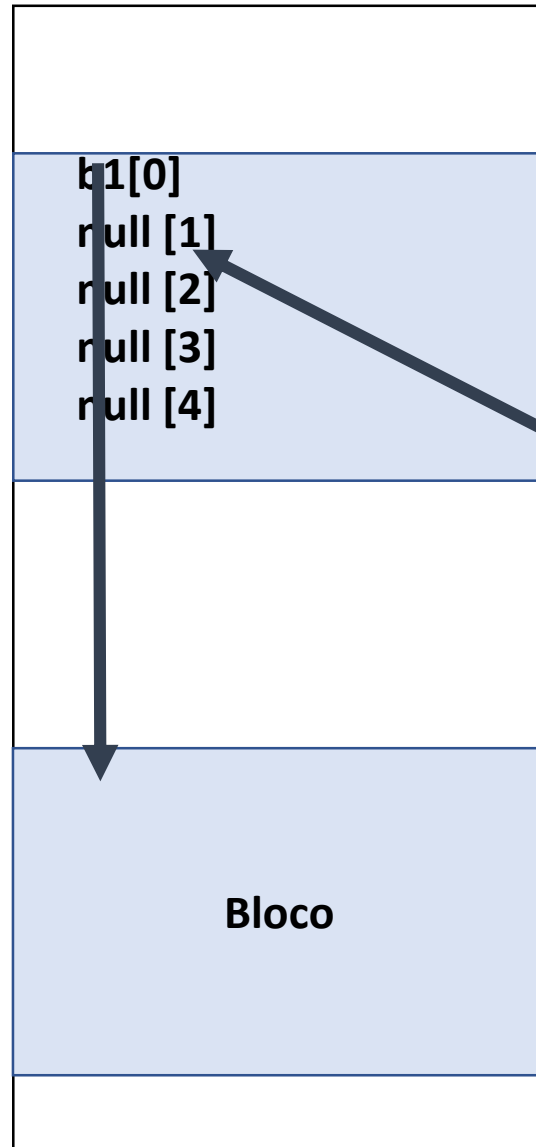
Stack

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

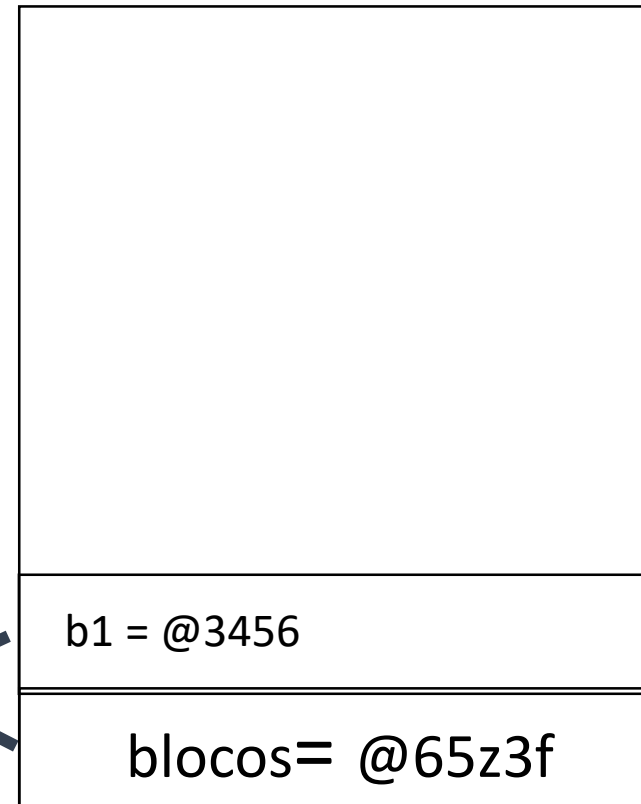




Heap



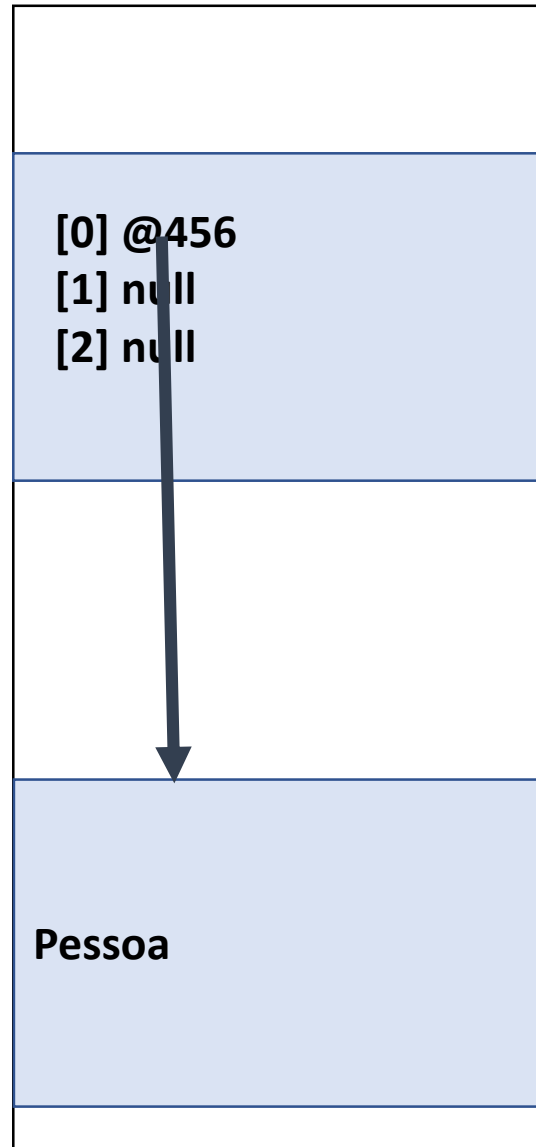
Stack



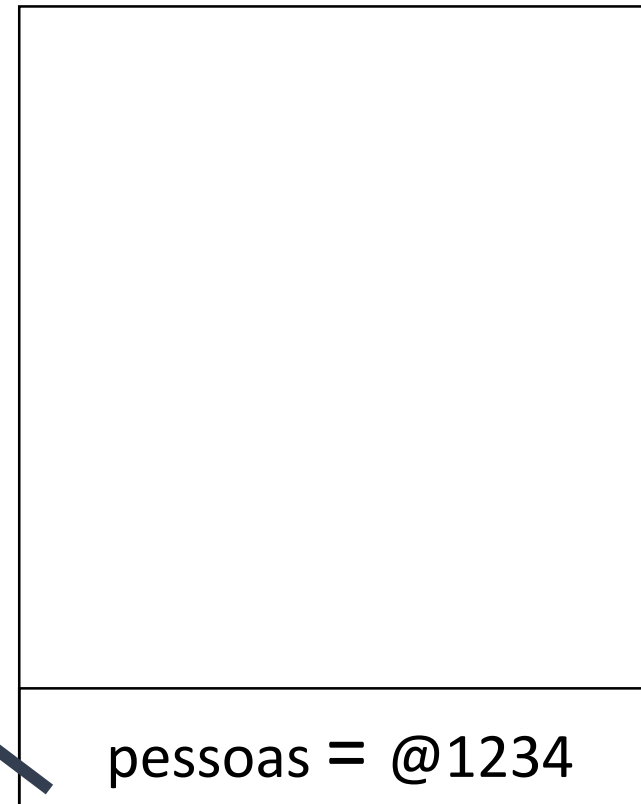
```
Bloco blocos[] = new Bloco[5];  
Bloco b1 = new Bloco();  
blocos[0] = b1;
```



Heap



Stack



```
Pessoa pessoas[] = new Pessoa[3];  
pessoas[0] = new Pessoa();
```



Array de Referências

```
Conta[] contas = new Conta[5]; // Array de "Conta"
```

☕ Quantas contas foram criadas?





Array de Referências

- ☕ Um array de tipos **primitivos** guarda **valores**, um de **objeto** guarda **referências**.
- ☕ Como preencher esse array então?
- ☕ Temos duas formas:



Array de Referências

Forma 1!

```
Conta[] contas = new Conta[5];
```

```
Conta conta1 = new Conta();
```

```
conta1.deposita(quantia: 1000); // Deposita e aumenta o saldo
```

```
contas [0] = conta1; //Associando
```

```
System.out.println(contas[0].saldo);
```



Array de Referências

☕ Forma 2!

```
Conta[] contas = new Conta[5];
```

```
contas[0] = new Conta();  
contas[0].deposita(quantia: 1000); // Deposita e aumenta o saldo
```

```
System.out.println(contas[0].saldo);
```



Array de Referências

☕ Percorrendo um Array

☕ Podemos usar o “For” combinado com *length*

```
double []versoesWindows = {95,98,2000,7,8,8.1,10};  
//observe o membro "length"  
//Ele possui salvo o tamanho do array  
for (int i = 0; i < versoesWindows.length; i++) {  
    System.out.println(versoesWindows[i]);  
}
```



Array de Referências

☕ For-Each ou “*Enhanced For*” (*For* melhorado)

```
double []versoesWindows = {95,98,2000,7,8,8.1,10};  
for (double versao : versoesWindows) {  
    System.out.println(versao);  
}
```

Retorno de
cada posição
(Each) – no
tipo do Array

Nome do Array



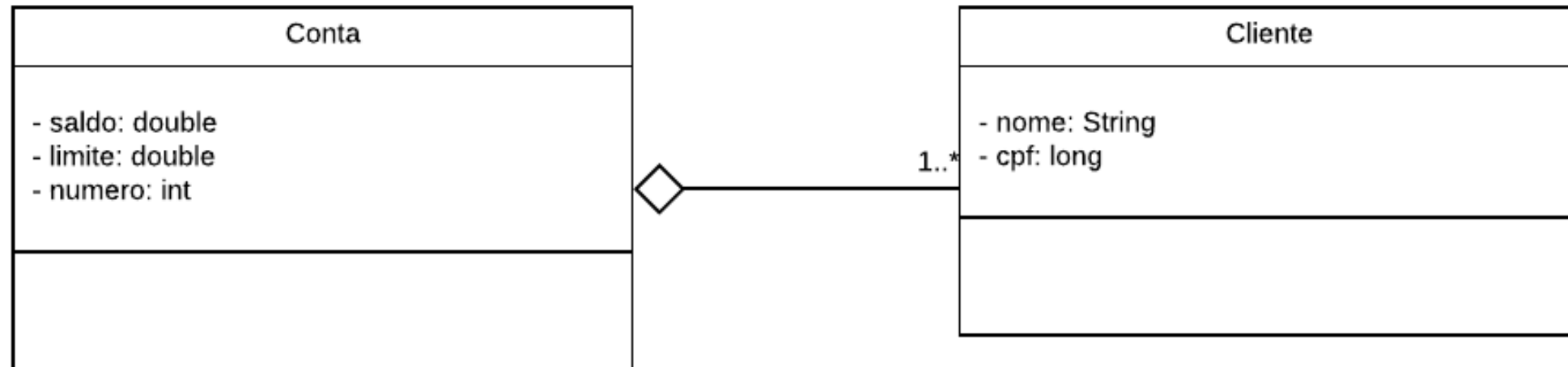
Multiplicidade

```
public class Conta {  
  
    //Membro das classe conta  
    private double saldo;  
    private double limite;  
    private int numero;  
    private Cliente clientes[];
```

☕ Como ficaria o diagrama UML?



Multiplicidade



☕ Conta possui 1 ou mais Clientes

Multiplicidade



0..1	Zero ou um.
1..1	Um e somente um.
0..*	Zero ou muitos.
*	Muitos.
1..*	No mínimo um ou muitos.
3..5	Mínimo de três e máximo de cinco.



Exercício 0

☕ Crie classes em Java que atendam as especificações UML abaixo

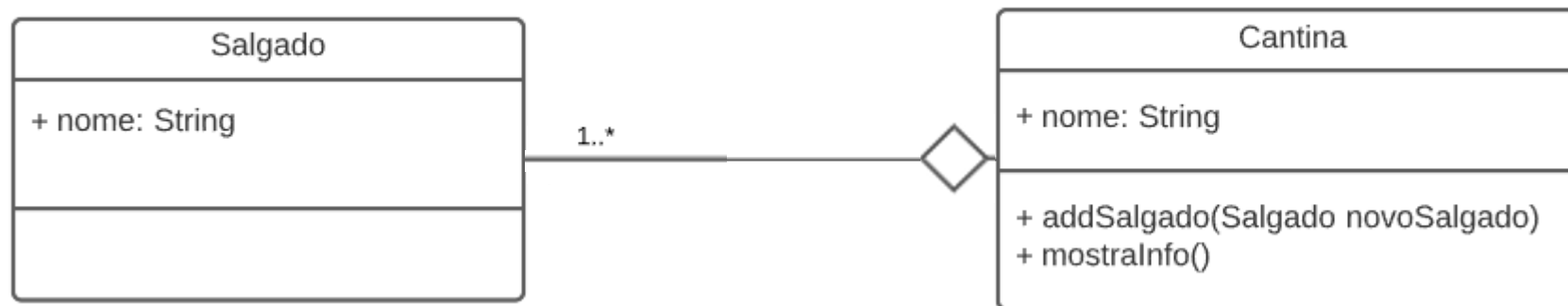
☕ Crie uma classe Main e, nela, crie:

☕ Uma cantina com nome “Cantina do Inatel”

☕ 3 Salgados da sua escolha

☕ Adicione os Salgados na Cantina

☕ Mostre os Salgados adicionados





Arrays Multidimensionais

☕ Podemos também declarar **arrays bidimensionais (matriz)** ou com **várias dimensões**.

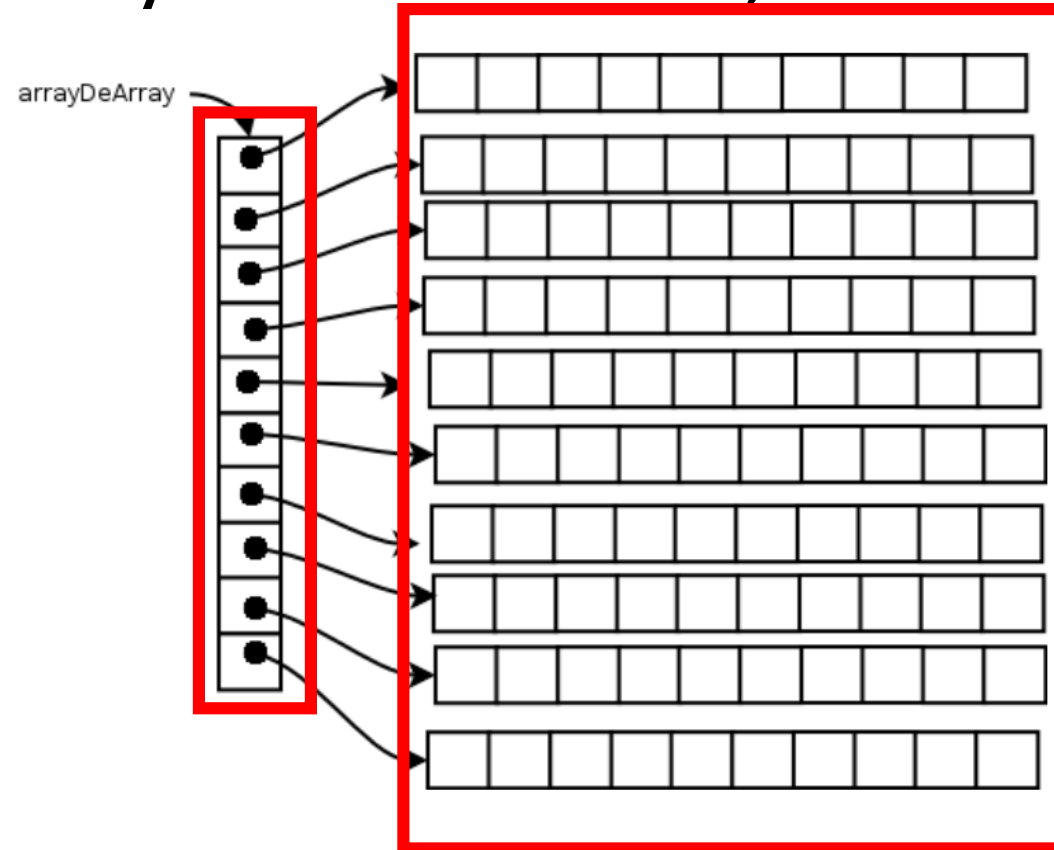
```
int [][] matrizExemplo;  
matrizExemplo = new int[5][5];
```

☕ Nesse exemplo temos um array bidimensional “5 por 5”. Ou ainda, “5 linhas e 5 colunas”. Uma matriz quadrada.



Arrays Multidimensionais

☕ No Java, um array bidimensional, é um “array de arrays”





Arrays Multidimensionais

Preenchendo uma matriz

```
int [][] matrizExemplo;  
matrizExemplo = new int[5][5];  
  
for (int i = 0; i < matrizExemplo.length; i++) {  
    for (int j = 0; j < matrizExemplo[i].length; j++) {  
        matrizExemplo[i][j] = i + j;  
    }  
}
```



Arrays Multidimensionais

Percorrendo uma matriz com o For-Each

```
int [][] matrizExemplo;  
matrizExemplo = new int[5][5];  
  
for (int[] linha : matrizExemplo) {  
    for (int valores : linha) {  
        System.out.print(valores + " ");  
    }  
    System.out.println("");  
}
```




Exercício 1 – Campo Minado (Desafio)

- Simule um jogo de campo minado, através de uma matriz 2x2.
- Coloque uma bomba em uma posição aleatória e peça para o usuário caminhar pelo Campo Minado (pergunte qual posição ele deseja acessar) até que ele consiga caminhar sem atingir nenhuma bomba
- Caso ele pise na mina, avise através de mensagens. Ele deverá percorrer novamente o Campo, mas agora já saberá onde tem uma mina.

```
Random rand = new Random();  
//Gera num aleatório entre 0 e 1  
int x= rand.nextInt(2);  
  
//Para ler do teclado a posicao digitada  
Scanner entrada = new Scanner(System.in);  
int digitado = entrada.nextInt();
```

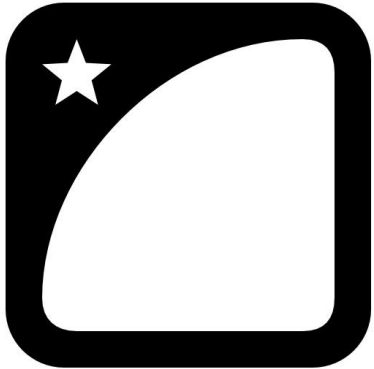


Resolução dos Exercícios



https://github.com/chrislima-inatel/C206_C125





Material Complementar



 Capítulo 14 da apostila FJ-11

 Um pouco de Arrays