

Cap.4 - Arrays



Objetivos

- Entender como funciona e verificar como declarar e instanciar Arrays no Java;
- Compreender como popular e percorrer Arrays;



Capítulo 4

Arrays

4.1. Introdução;

4.2. Arrays de Referências;

4.3. Percorrendo um Array;

4.4. Diagrama de Classes: Multiplicidade;

4.5. Arrays Bidimensionais;



4.1. Introdução

Um array (arranjo) é uma coleção de um ou mais objetos, do mesmo tipo, armazenados em endereços adjacentes de memória. Cada objeto é chamado de elemento do array.

No Java, como podemos declarar um Array?

Exemplo:

```
// Declarando uma variável
// do tipo array que guardará
// a referência de um Objeto Array
int[] idades;

// Criando um Objeto Array
// e guardando sua referência
idades = new int[10];
```

No exemplo foi criado um array de Inteiros de 10 posições.

4.1. Introdução

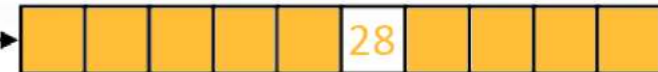
No Java, os índices do array vão de 0 a $n-1$, onde n é o tamanho dado ao array no momento de sua criação.

E para acessarmos/modificarmos uma posição do Array?

Exemplo:

```
// Sexta posição do array  
// recebendo um valor  
idades[5] = 28;
```

idades



Outra estratégia bem interessante na hora de criar um array é utilizar de uma variável inteira para representar seu tamanho:

Exemplo:

```
// Criando um array de tamanho 100  
// usando de uma variável inteira  
int n = 100;  
int[] idades = new int[n];
```

4.2. Arrays de Referências

É comum ouvirmos "array de objetos". Porém, quando criamos um array de alguma Classe, na verdade temos um array de REFERÊNCIAS.

Exemplo:

```
Conta[] minhasContas = new Conta[10];
```

Quantas Contas foram criadas aqui?

Na verdade nenhuma. Foram criados 10 espaços que podem ser utilizados para guardarem referências de Contas. Se nenhuma posição receber referências e forem acessadas como na Figura abaixo, um erro ocorrerá.

```
System.out.println(minhasContas[5].saldo);
```


4.2. Arrays de Referências

"Um array de tipos primitivos guarda valores, um array de objetos guarda referências"

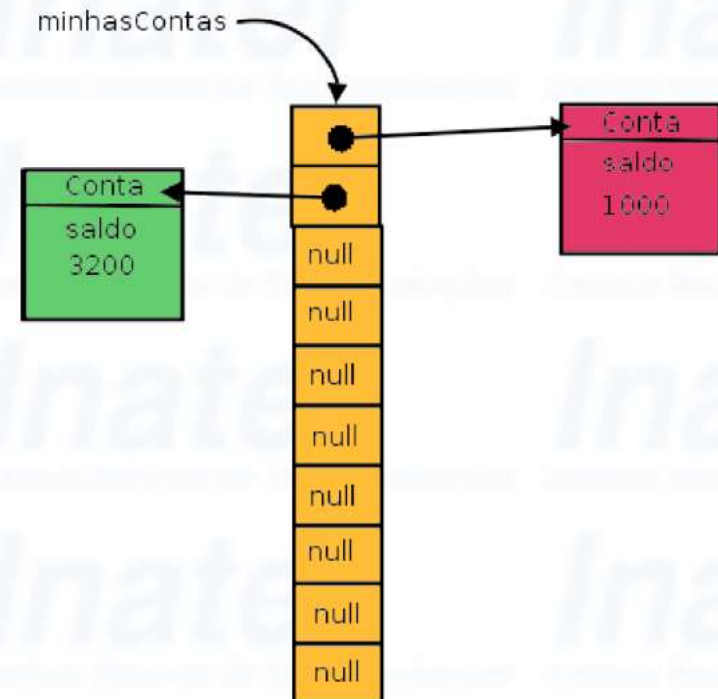
E como podemos preencher o array com as referências?

// FORMA 1

```
Conta novaConta = new Conta();  
novaConta.saldo = 1000;  
minhasContas[0] = novaConta;
```

// FORMA 2

```
minhasContas[1] = new Conta();  
minhasContas[1].saldo = 3200;
```



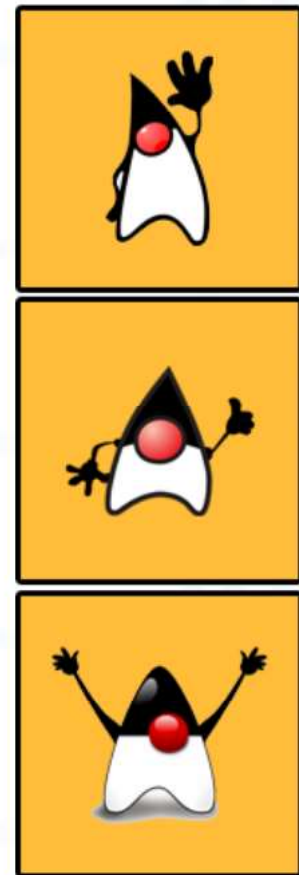
4.3. Percorrendo um Array

Percorrer um array é muito simples.

Exemplo:

```
// FORMA 1
// Percorrendo um array utilizando da
// variável índice i e do atributo length.
for (int i = 0; i < minhasContas.length; i++) {
    if(minhasContas[i] != null)
    {
        System.out.println(minhasContas[i].saldo);
    }
}

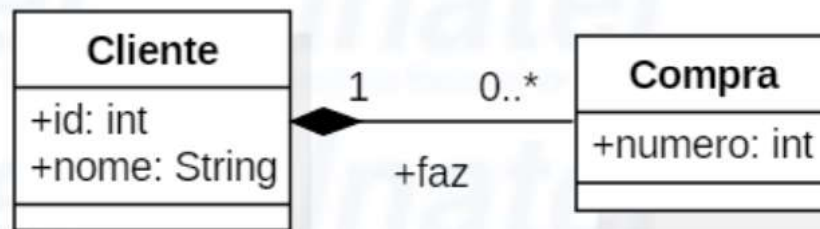
// FORMA 2
// Percorrendo o mesmo array utilizando um
// recurso do Java chamado "Enhanced-for"
for (Conta c : minhasContas) {
    if (c != null) {
        System.out.println(c.saldo);
    }
}
```



4.4. Diagrama de Classes Multiplicidade

Na UML a multiplicidade é usada para definir quantos objetos participam de um relacionamento.

Exemplo:



```
public class Cliente {
    int id;
    String nome;
    Compra[] compras = new Compra[100];
}
```

Um Cliente faz nenhuma ou várias compras, e uma compra é feita por um cliente.

Exemplos de valores para a multiplicidade:

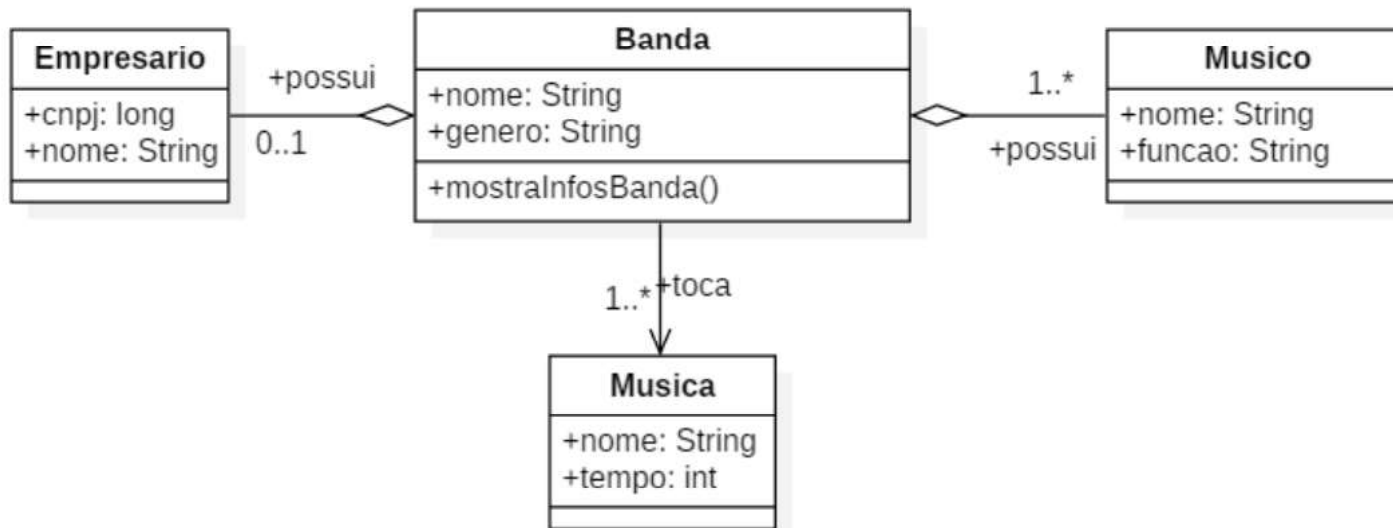
0..1	Zero ou um.
1..1	Um e somente um.
0..*	Zero ou muitos.
*	Muitos.
1..*	No mínimo um ou muitos.
3..5	Mínimo de três e máximo de cinco.

4.4. Diagrama de Classes

Multiplicidade

Exercício - RockBand

Crie Classes em Java que atendam a seguinte especificação:



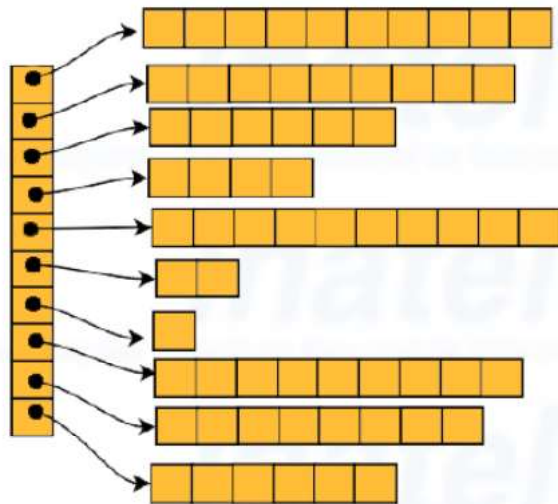
Depois de criada as Classes, monte uma banda com alguns membros, algumas músicas e mostre suas informações.



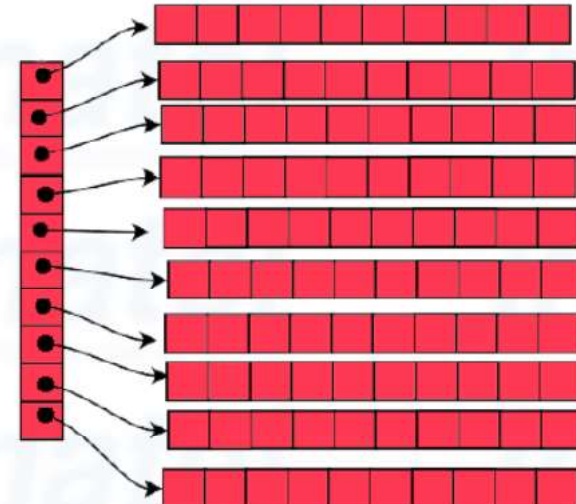
4.5. Arrays Bidimensionais

Arrays podem ter mais de uma dimensão. Um Array Bidimensional em Java é um array de arrays.

Um Array Bidimensional não precisa necessariamente ser quadrado ou retangular (ex: Simulando uma Matriz), mas cada linha pode ou não ter um número diferente de colunas.



Array Bidimensional com número diferente de colunas.



Array Bidimensional Quadrado.

Exemplo de uma Matriz no Java:

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


4.5. Arrays Bidimensionais

Exercício - CampoMinado

Simule um Campo Minado por meio de uma Matriz 3x3 e coloque 1 bomba em uma posição randomica da mesma. Depois, o usuário deverá caminhar pelo Campo Minado, e aquele que conseguir caminhar por todas as posições sem bombas, zera o jogo.

Dicas:

1. Mensagens para quando o jogador:
 - a) Não pisar na bomba: "Muito Bom! x)";
 - b) Pisar na bomba: "BOOOOM! x(GAME OVER!";
 - c) Zerar o jogo: "Parabéns você ZEROU o jogo! \o/";
2. `Random randomGenerator = new Random();`
3. `numAleatorio = randomGenerator.nextInt(3);`
4. Faça um laço que questione o jogador sobre a posição da Matriz que ele deseja caminhar;
5. `Scanner valorTeclado = new Scanner (System.in);`
6. `int valor = valorTeclado.nextInt();`
7. Conte o número de jogadas certas feitas pelo jogador.



FIM DO CAPÍTULO 4



Próximo Capítulo

Modificadores de Acesso
e Atributos de Classe