



**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Câmpus Feliz

Arrays e Collections

Prof. Moser Fagundes

Programação II
Técnico em Informática

Sumário

- Classe Arrays
- Classe Collections

Classe Arrays

- A classe **Arrays** ajuda a evitar a “reinventar a roda” fornecendo métodos **estáticos** para manipular **arrays comuns**.
- Não é usada com ArrayList
- Métodos da classe **Arrays**:
 - **sort** para ordenar
 - **binarySearch** para pesquisar

sort

Ordena os arrays

```
package exemplo;

import java.util.Arrays;

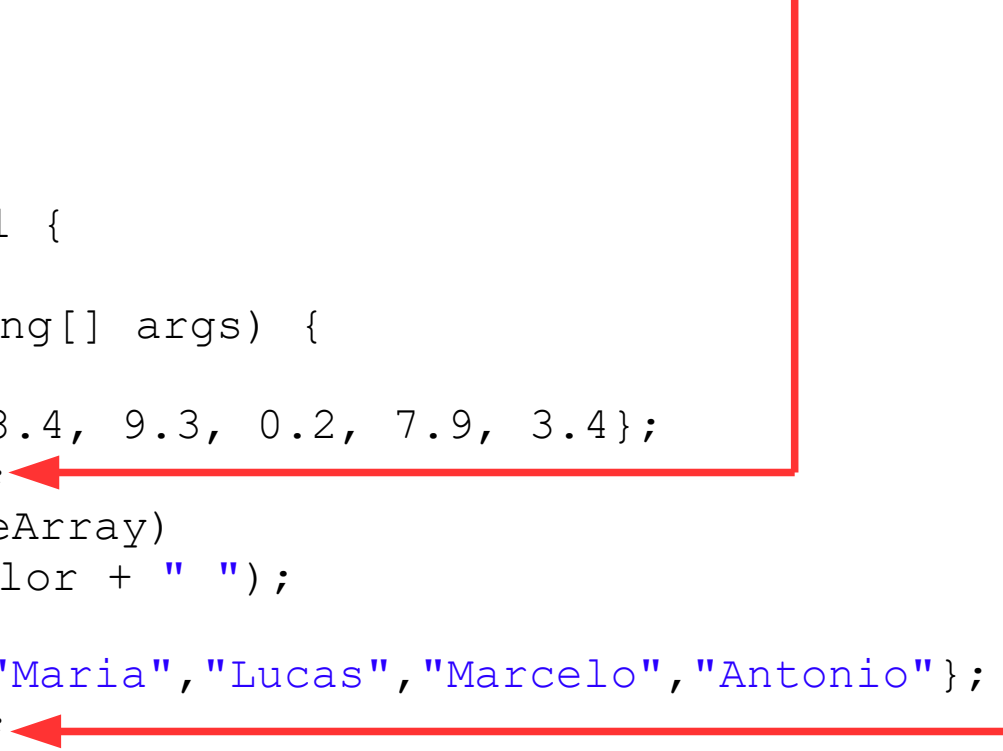
public class ArrayManipulations1 {

    public static void main(String[] args) {

        double[] doubleArray = {8.4, 9.3, 0.2, 7.9, 3.4};
        Arrays.sort(doubleArray);
        for(double valor : doubleArray)
            System.out.println(valor + " ");

        String[] stringArray = {"Maria", "Lucas", "Marcelo", "Antonio"};
        Arrays.sort(stringArray);
        for(String valor : stringArray)
            System.out.println(valor + " ");

    }
}
```



binarySearch

```
package exemplo;
```

```
import java.util.Arrays;
```

```
public class ArrayManipulations4 {
```

```
    public static void main(String[] args) {
```

```
        int[] intArray = {1, 2, 5, 67, 800, 2112};
```

```
        int posicao1 = Arrays.binarySearch(intArray, 80);
```

```
        if(posicao1 >= 0)
```

```
            System.out.println("80 encontrado na posicao " + posicao1);
```

```
        else
```

```
            System.out.println("80 nao encontrado!");
```

```
        int posicao2 = Arrays.binarySearch(intArray, 800);
```

```
        if(posicao2 >= 0)
```

```
            System.out.println("800 encontrado na posicao " + posicao2);
```

```
        else
```

```
            System.out.println("800 nao encontrado!");
```

```
    }
```

```
}
```

Faz uma busca por um elemento e retorna o índice do elemento no array. Se o elemento não está no array, retorna um número negativo.

O array **deve** estar ordenado!



Classe Collections

- A classe **Collections** ajuda a evitar a “reinventar” a roda fornecendo métodos **estáticos** para manipular coleções.
- Não é usada com arrays comuns.
- Métodos da classe **Collections**:
 - **sort** para ordenar
 - **binarySearch** para pesquisar

Classe Pessoa

```
package exemplos;
```

```
public class Pessoa implements Comparable<Pessoa> {  
    private int id;  
    private String nome;  
  
    public Pessoa(int id, String nome) {  
        this.id = id;  
        this.nome = nome;  
    }
```

// Os métodos GET/SET foram omitidos por questão de espaço

```
    public int compareTo(Pessoa p) {  
        if(this.id < p.id)  
            return -1;  
        else if(this.id > p.id)  
            return 1;  
        return 0;  
    }  
}
```

Este método é usado para comparar dois objetos: **this** e **p**.

Com as comparações, é possível ordenar os elementos do ArrayList.

sort

```
package exemplos;
```

```
import java.util.ArrayList;  
import java.util.Collections;
```

```
public class TesteCollections {
```

```
    public static void main(String[] args) {  
        ArrayList<Pessoa> x = new ArrayList<Pessoa>();  
        x.add(new Pessoa("A",33));  
        x.add(new Pessoa("C",15));  
        x.add(new Pessoa("B",20));
```

```
        Collections.sort(x);
```

```
        for (Pessoa p : x) {  
            System.out.println(p.getId() + " : " + p.getNome());  
        }
```

```
    } // Collections.reverse(x);
```

Ordena o
ArrayList



binarySearch

```
package exemplos;
```

```
import java.util.ArrayList;  
import java.util.Collections;
```

```
public class TesteCollections {
```

```
    public static void main(String[] args) {  
        ArrayList<Pessoa> x = new ArrayList<Pessoa>();  
        x.add(new Pessoa("A",33));  
        x.add(new Pessoa("C",15));  
        x.add(new Pessoa("B",20));
```

```
        Collections.sort(x); // É obrigatório ordenar
```

```
        int indice = Collections.binarySearch(x,new Pessoa("A",33));  
        // O retorno é o índice do registro caso exista  
        // Caso não exista, retorna um número negativo  
        System.out.println(indice);
```

```
    }
```

```
}
```

Foi criado um novo objeto para ser usado como parâmetro da busca binária

