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

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author GalaxyBook3
 */
public class JJOO implements Comparable<JJOO> {
    private String pais;
    private Integer medallasOro;
    private Integer medallasPlata;
    private Integer medallasBronce;
    private Integer lugarRanking;

    public JJOO(String pais, Integer medallasOro, Integer medallasPlata, Integer medallasBronce, Integer lugarRanking) {
        this.pais = pais;
        this.medallasOro = medallasOro;
        this.medallasPlata = medallasPlata;
        this.medallasBronce = medallasBronce;
        this.lugarRanking = lugarRanking;
    }

    public String getPais() {
```

2025.03.27 20:02:46

```java
        return pais;
    }

    public Integer getMedallasOro() {
        return medallasOro;
    }

    public Integer getMedallasPlata() {
        return medallasPlata;
    }

    public Integer getMedallasBronce() {
        return medallasBronce;
    }

    @Override
    public String toString() {
        return String.format("* Pais: %s, Oro: %d, Plata: %d, Bronce: %d, Ranking: %d",
                pais, medallasOro, medallasPlata, medallasBronce, lugarRanking);
    }

    @Override
    public int compareTo(JJOO other) {
        return Comparator.comparing(JJOO::getMedallasOro).reversed()
                .thenComparing(JJOO::getMedallasPlata).reversed()
                .thenComparing(JJOO::getMedallasBronce).reversed()
                .thenComparing(JJOO::getPais, String.CASE_INSENSITIVE_ORDER)
                .compare(this, other);
    }
```

```java
public static void bubbleSort(JJOO[] array) {
    for (int i = 0; i < array.length - 1; i++) {
        for (int j = 0; j < array.length - i - 1; j++) {
            if (array[j].compareTo(array[j + 1]) > 0) {
                JJOO temp = array[j];
                array[j] = array[j + 1];
                array[j + 1] = temp;
            }
        }
    }
}

public static void insertionSort(JJOO[] array) {
    for (int i = 1; i < array.length; i++) {
        JJOO temp = array[i];
        int j = i - 1;
        while (j >= 0 && array[j].compareTo(temp) > 0) {
            array[j + 1] = array[j];
            j--;
        }
        array[j + 1] = temp;
    }
}

public static void shellSort(JJOO[] array) {
    for (int gap = array.length / 2; gap > 0; gap /= 2) {
        for (int i = gap; i < array.length; i++) {
            JJOO temp = array[i];
```

```java
            int j = i;
            while (j >= gap && array[j - gap].compareTo(temp) > 0) {
                array[j] = array[j - gap];
                j -= gap;
            }
            array[j] = temp;
        }
    }
}

public static void mostrarArray(JJOO[] array) {
    System.out.println(Arrays.toString(array));
}
}
```