

Interfaces Comparable y Comparator en Java

En Java, las interfaces Comparable y Comparator se utilizan para definir el orden de los objetos.

Son fundamentales para ordenar colecciones como listas o arreglos que contienen objetos personalizados.

1. Interface Comparable

La interfaz Comparable<T> se utiliza para definir un orden natural entre objetos de una clase.

Esta interfaz contiene un único método:

```
public int compareTo(T o);
```

Reglas del método compareTo:

- Retorna un número negativo si `this < o`.
- Retorna cero si `this == o`.
- Retorna un número positivo si `this > o`.

Ejemplo con Comparable:

Supongamos que tienes una clase Persona y quieres ordenarlas por edad de forma natural:

```
class Persona implements Comparable<Persona> {  
  
    String nombre;  
  
    int edad;  
  
  
    public Persona(String nombre, int edad) {  
  
        this.nombre = nombre;  

```

```
        this.edad = edad;
    }

    @Override
    public int compareTo(Persona otra) {
        return Integer.compare(this.edad, otra.edad);
    }
```

```
    @Override
    public String toString() {
        return nombre + " (" + edad + ")";
    }
}
```

Uso:

```
List<Persona> lista = new ArrayList<>();
lista.add(new Persona("Ana", 30));
lista.add(new Persona("Luis", 25));
lista.add(new Persona("Carlos", 35));
```

```
Collections.sort(lista); // Ordena usando compareTo
```

2. Interface Comparator

La interfaz `Comparator<T>` permite definir múltiples criterios de ordenamiento.

Se utiliza especialmente cuando no puedes o no quieres modificar la clase original.

Método principal:

```
int compare(T o1, T o2);
```

Ejemplo con Comparator:

Ordenar la misma clase Persona por nombre alfabéticamente:

```
class ComparadorPorNombre implements Comparator<Persona> {  
    @Override  
    public int compare(Persona p1, Persona p2) {  
        return p1.nombre.compareTo(p2.nombre);  
    }  
}
```

Uso:

```
Collections.sort(lista, new ComparadorPorNombre()); // Ordena por nombre
```

Diferencias clave entre Comparable y Comparator:

Característica	Comparable	Comparator
Ubicación	Dentro de la clase	Clase separada
Método	compareTo(T o)	compare(T o1, T o2)
Modificabilidad	Requiere modificar la clase	No requiere modificar la clase
Uso principal	Orden natural	Orden personalizado o múltiple