INTERFAZ COMPARABLE

Walter Martín Lopes

¿Qué es una "*INTERFAZ*"?

Una interfaz en Java es una colección de métodos abstractos y, a partir de Java 8, también puede contener métodos con implementación por defecto y métodos estáticos. Las interfaces **no pueden tener atributos y no pueden ser instanciadas directamente**, además las clases pueden implementar una o más interfaces.

Las interfaces son útiles en Java para lograr la herencia múltiple de tipo, ya que Java no permite que una clase herede de más de una clase. Además, las interfaces son una ***excelente manera de especificar un contrato***, asegurando que la clase implemente ciertos métodos.

Cuando una clase implementa una interfaz, debe proporcionar una implementación para todos los métodos abstractos de esa interfaz. Para implementar una interfaz, se utiliza la palabra clave ***"implements"*** seguida del nombre de la interfaz.

En el caso del código ejemplo, la clase *"Donacion"* implementa la interfaz ***"Comparable<Donacion>"***, lo que indica que las instancias de "Donacion" pueden ser comparadas entre sí utilizando el método **"*compareTo*"**.

¿Para qué sirve la interfaz " *COMPARABLE* "?

La interfaz ***Comparable*** en Java es utilizada para *ordenar objetos de una clase en un orden natural*. La interfaz define un único método llamado ***compareTo***.

La interfaz ***Comparable*** se utiliza generalmente cuando se desea ordenar una colección de objetos, como una lista o un conjunto, basándose en alguna propiedad de los objetos en la colección.

Cuando implementas la interfaz ***Comparable***, debes proporcionar la implementación del método ***compareTo***.

¿Qué hace este método?

> ***Este método toma un objeto del mismo tipo como argumento y devuelve un número entero que representa el resultado de la comparación.***

Es utilizado para comparar dos objetos y determinar su orden relativo. El método toma un objeto como argumento y compara el objeto actual con el objeto pasado como argumento. El resultado de la comparación es un número entero que indica el orden relativo de los dos objetos.

Aquí hay una explicación simple de cómo funciona ***compareTo***:

- Si el objeto actual se considera "menor" que el objeto argumento, el método devuelve un ***número negativo***.
- Si el objeto actual se considera "mayor" que el objeto argumento, el método devuelve un ***número positivo***.
- Si el objeto actual se considera igual al objeto argumento, el método devuelve ***cero***.

El término "menor" o "mayor" depende de cómo definas la lógica de comparación en tu método ***compareTo***.

En el caso de la clase "Donacion", el método ***compareTo*** compara las cantidades de las donaciones. Una vez implementado nos servirá para ordenar automáticamente objetos de la clase "Donacion" en cualquier estructura de datos.

EJEMPLO

```java
package t11ejercicio06;

/**
 *
 * @author Walter Martín Lopes
 */
public class Donacion implements Comparable<Donacion> {

    private String nombre;
    private float cantidad;

    //CONSTRUCTORES
    public Donacion() {...4 lines }

    public Donacion(String nombre, float cantidad) {...4 lines }

    //SETTER Y GETTER
    public void setNombre(String nombre) {...3 lines }

    public void setCantidad(float cantidad) {...3 lines }

    public String getNombre() {...3 lines }

    public float getCantidad() {...3 lines }

    //Implemento el metodo compareTo para especificar el criterio de ordenacion natural
    @Override
    public int compareTo(Donacion t) {
        if (this.cantidad < t.getCantidad()) {
            return 1;
        } else if (this.cantidad > t.getCantidad()) {
            return -1;
        } else {
            return 0;
        }
    }

    //==========METODOS PROPIOS==========
    public void mostrar() {...5 lines }
}
```

```java
package t11ejercicio06;

import ...

/**
 *
 * @author Walter
 */
public class Test {

    public static int mostrarMenu() {...17 lines }

    //OPCIÓN 1==========================================================================
    /*Al implementar la interfaz comparable en la clase donacion, cuando añadimos
    una donacion a la campaña se ordena automaticamente.*/
    public static void anadirDonacion(Campania campania) {
        String nombre;
        float cantidad;
        do {
            System.out.println("-----DONACION-----");
            nombre = pedirNombre();
            cantidad = pedirCantidad();
            campania.introducirDonacion(nombre, cantidad);
            System.out.println("------------------");
        } while (seguir());
    }

    //OPCIÓN 2==========================================================================
    public static void mostrarDonaciones(Campania campania) {...3 lines }

    //OPCIÓN 3==========================================================================
    //Metodo que muestra solo las donaciones cuyo nombre coincide
    public static void mostrarDonaciones(Campania campania, String nombre) {...3 lines }

    //METODOS PEDIR-----------------------------------------
```