

Interface y Comparable

Gerny Díaz - 2022172980
Jurgenn Morales - 2016145484
Franco Rojas - 2022437823
Estefani Valverde - 2021554564





Clase Interface

Una interface en Java es una especie de plantilla, marco o receta, para realizar una tarea. Consiste en una colección de propiedades constantes estáticas y métodos abstractos sin cuerpo (no hay código ni sentencias dentro de ellos) que nos especifica qué se debe hacer, pero no cómo hacerlo.



¿Para qué se utiliza?

Por lo general se utilizan interfaces cuando se quiere unificar nombres de métodos pero seguir manteniendo comportamientos que no tienen nada que ver uno con el otro.



Utilidades

Las principales utilidades de las ***interfaces*** son ahorrar código, refactorizaciones más limpias, reutilización de métodos y modificaciones más simples en nuestros programas.

Generalmente se utiliza este recurso cuando queremos unificar comportamientos en las clases inferiores, que no tienen nada que ver una con la otra.

Las ***interfaces*** son el factor común entre clases que no pertenecen a una misma jerarquía.



Toma en cuenta lo siguiente

Lo definimos utilizando la palabra clave de la interface.

Puede contener solo variables estáticas.

No puede contener un constructor porque las interfaces no pueden ser instanciadas.

Las interfaces pueden extender otras interfaces.

Una clase puede implementar cualquier número de interfaces.



Ejemplo en Java

Área y Perímetro de figuras geométricas





Clase Comparable

Una clase comparable es una interfaz que está creada o desarrollada dentro del JRE de java, por lo cual es necesario tan solo implementarla directamente en las clases.

```
public class Estudiante implements Comparable<Estudiante>{
```



¿Para qué se utiliza?

Comparable tiene un método llamado `CompareTo()` el cual permite ordenar un objeto de acuerdo a un atributo especificado (string, integer, double, entre otros), ya sea en orden ascendente o descendente.



¿Cómo funciona CompareTo()?

CompareTo() es un método abstracto que retorna un valor entero para todos los casos:

- $n=-1$, si el objeto es menor.
- $n=1$, si el objeto es mayor.
- $n=0$, si son iguales.



¿Cómo comparar atributos?

Existen dos formas para comparar atributos:

```
@Override
public int compareTo(Estudiante o) {
    return obteberEdad() - o.obteberEdad();
}
```

```
@Override
public int compareTo(Estudiante o) {
    return this.ira.compareTo(o.getIra());
}
```



¿Cómo comparar atributos?

Para invocar el método `compareTo()` y lograr su correcto funcionamiento, primero debe existir un `Array` o `ArrayList` dentro de la clase donde se va a realizar el ordenamiento. Segundo se necesita una colección (`Collections`) el cual nos permitirá agilizar el acceso a los datos y del método `sort` que nos permite ordenar un array. Después solo es necesario imprimir para mostrar el resultado.

```
public void ordenamiento() {  
    Collections.sort(estudiantes);  
}
```



Ejemplo en Java

Ordenar personas por edad





Bibliografía

[Interface en Java ¿Qué son y cómo se implementan? \(open-bootcamp.com\)](https://open-bootcamp.com/interfaz-en-java-que-son-y-como-se-implementan/)

[Interfaces en Java - dCodinGames](#)