

Interfaces

Ojo no nos referimos a interfaces gráficas aún

Las interfaces son una forma de especificar **qué** debe hacer una clase sin especificar el **cómo**.

Las interfaces tienen una semejanza con las clases abstractas, en el sentido que no tiene sentido definir objetos instancia de una interfaz. Igual que las clases abstractas, la clase asociada a una interfaz se compromete a implementar todos los métodos en ellas definidos, PERO en este caso la relación no es de herencia en plenitud, dado que no hay atributos en la definición de una interfaz.

Las interfaces no son clases, sólo especifican requerimientos para la clase que las implementa y cuando usamos objetos de esas clases, los servicios que pueden ser invocados.

¿Cómo se usan las interfaces?

Debemos crear una clase que implementa la interfaz, implementando cada uno de los métodos de la interfaz.

Podemos definir métodos que usen como parámetro objetos que implementen la interfaz. Basta usar el nombre de la interfaz como el tipo del parámetro.

Luego, las instancias de una clase que implemente la Interfaz, pueden tomar el lugar del argumento donde se espere alguien que implemente la interfaz.

Ejemplo: Supongamos la interfaz que describe a un nadador de cuatro estilos (nadaLibre(), nadaDespaldas(), nadaPecho(), nadaMariposa()). En ELO hay muchas instancias de Alumnos, entre ellos hay algunos que pueden ser instancias de AlumnosNadadores, clase que implementa la interfaz Nadador. Con ello donde se necesite un nadador, podremos enviar esos alumnos. Cuando se pida un Nadador en una función, se entiende que sólo se invocarán sus métodos como nadador, pues no interesa si se trata de una instancia de Alumno, Funcionario u otra Clase.

Ejemplo Java

Sea la Interfaz Comparable.java, el contenido de este archivo contendrá sólo el **qué** no el **cómo** (implementación):

```
public interface Comparable
{
    int compareTo(Object otro);
}
```

Luego la clase Employee podría implementar Comparable para comparar dos empleados. La declaración de la clase empleado se modificaría a:

```
class Employee implements Comparable
```

La otra modificación es que la clase Employee debe implementar esta función. Por ejemplo:

```
public int compareTo(Object otroObject)
{
    Employee otro=(Employee) otroObject;
    if (salary < otro.salary) return -1;
    if (salary > otro.salary) return 1;
    return 0;
}
```

Ver ejemplo [EmployeeSortTest.java](#). Destacar invocación a `Array.sort()`

Java sólo permite heredar de una clase, pero permite implementar múltiples interfaces.

Ver ejemplo [TimerTest.java](#)

Este ejemplo muestra las ideas básicas para el manejo de interfaces gráficas de Usuarios (GUIs)