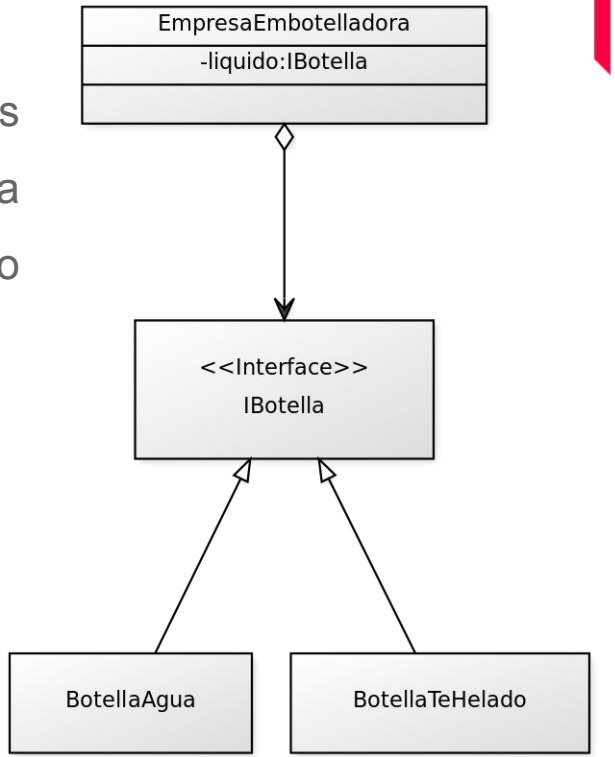
Principios SOLID

Principios O y L

Principio O - Open/Close (Abierto/Cerrado)

Ante un cambio de los requisitos, el diseño de las entidades existentes debe permanecer inalterado, recurriendo a la extensión del comportamiento de dichas entidades añadiendo nuevo código, pero nunca cambiando el código ya existente.



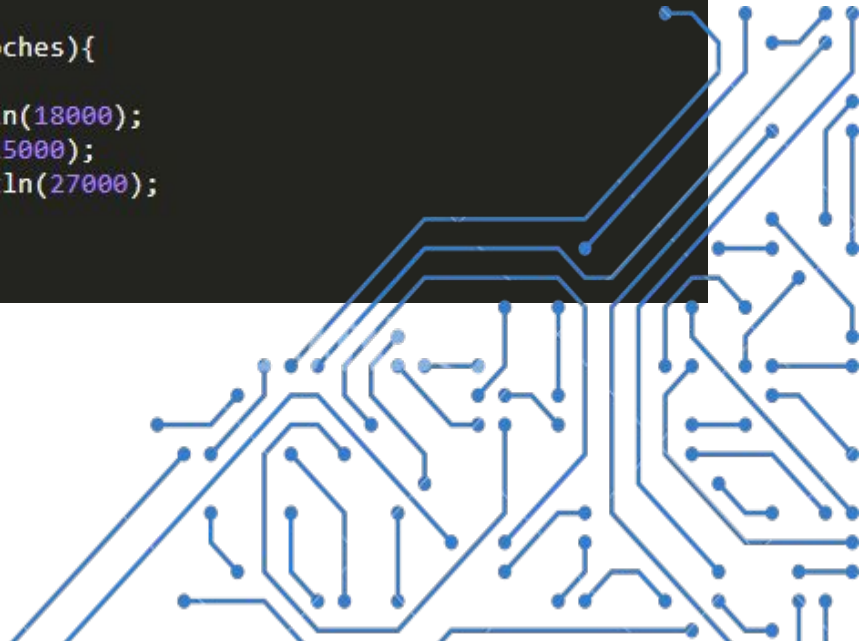
Principio O - Open/Close Ejemplo de mal uso

```
class Coche {
    String marca;

    Coche(String marca){ this.marca = marca; }

    String getMarcaCoche(){ return marca; }
}
```

```
public static void main(String[] args) {
    Coche[] arrayCoches = {
            new Coche("Renault"),
            new Coche("Audi")
    };
    imprimirPrecioMedioCoche(arrayCoches);
}

public static void imprimirPrecioMedioCoche(Coche[] arrayCoches){
    for (Coche coche : arrayCoches) {
        if(coche.marca.equals("Renault")) System.out.println(18000);
        if(coche.marca.equals("Audi")) System.out.println(25000);
    }
}
```

```
Coche[] arrayCoches = {
    new Coche("Renault"),
    new Coche("Audi"),
    new Coche("Mercedes")
};
```

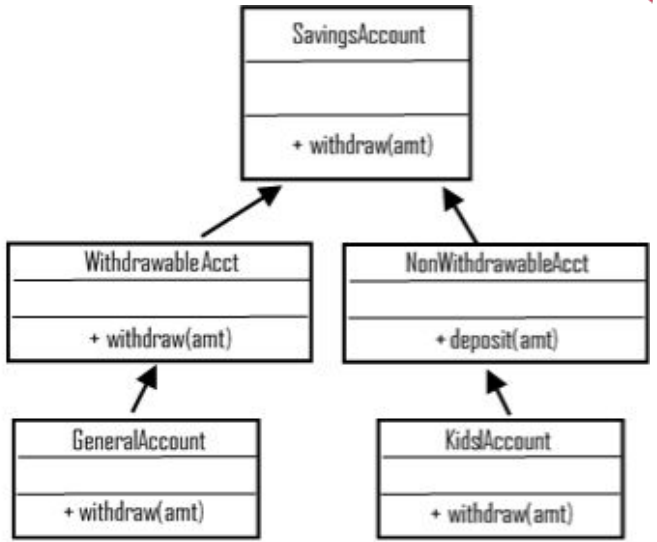
```
public static void imprimirPrecioMedioCoche(Coche[] arrayCoches){
    for (Coche coche : arrayCoches) {
        if(coche.marca.equals("Renault")) System.out.println(18000);
        if(coche.marca.equals("Audi")) System.out.println(25000);
        if(coche.marca.equals("Mercedes")) System.out.println(27000);
    }
}
```



Principio L - Liskov substitution principle (Principio de substitución de Liskov)

Este principio postula que cada subclase usada en nuestro sistema puede ser sustituible por su superclase sin afectar el sistema.

Poder hacer esto, es un síntoma de que nuestra jerarquía de clases está hecha correctamente.



Principio L - Liskov substitution principle (Principio de substitución de Liskov)

Ejemplo de mal uso

```
 Coche[] arrayCoches = {
         new Renault(),
         new Audi(),
         new Mercedes(),
         new Ford()
 };

public static void imprimirNumAsientos(Coche[] arrayCoches){
    for (Coche coche : arrayCoches) {
        if(coche instanceof Renault)
            System.out.println(numAsientosRenault(coche));
        if(coche instanceof Audi)
            System.out.println(numAsientosAudi(coche));
        if(coche instanceof Mercedes)
            System.out.println(numAsientosMercedes(coche));
        if(coche instanceof Ford)
            System.out.println(numAsientosFord(coche));
    }
}
imprimirNumAsientos(arrayCoches);
```

Referencias

https://repositorio.grial.eu/bitstream/grial/354/1/Abierto.pdf

https://www.enmilocalfunciona.io/principios-solid/