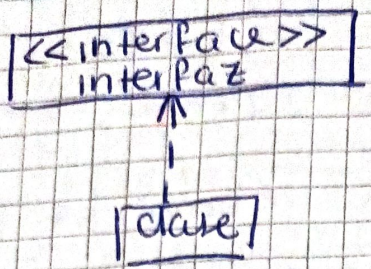
PRACTICA NIVEL 3

① 

La relación es DEPENDENCIA, la clase USA la interfaz.

Ejemplo:

Podríamos tener la clase Pintor que tiene el método colorear un objeto y el objeto que recibe debe ser de la interfaz coloreable

```
public class Pintor {
    public void colorear ( Coloreable objeto, color color){
        objeto.cambiarDeColor (color);
    }
}
public interface Coloreable {
    public void cambiarDeColor ( color color);
}
```

② Principios de diseño

DRY: se ve código repetido en las clases Producto y Suma en el método único de cada una.

SRP: las clases producto y suma no tienen responsabilidad única (piden el nº, hacen la suma/multiplicación y lo muestran).

funcionamiento "servido"

Patrón de diseño Template Method.

Primero podemos definir los pasos del algoritmo:
- pedir nº al cliente
- hacer la operación
- mostrar resultado

estos pasos pueden ser definidos en una clase abstracta y que las subclases Producto y Suma sobreescriban sobre los mismos (o los implementen mejor dicho), de manera personalizada).

③ cómo se usaría desde otra clase.

```
public class main {
    Sumador sumador = Sumador.getInstance();
    System.out.println(sumador.sumar(3,6));
}
```

Viola SRP, pues sumador hace la suma y se asegura que tenga una única instancia

PATRÓN DE DISEÑO Singleton.