INTERFACE AND ABSTRACT

Clase Abstracta

Definición:
Plantilla para otras clases. No se pueden instanciar.
Palabra clave: abstract
Propósito: Definir una estructura común y obligar a implementar métodos.
Características:

- Pueden tener métodos normales y abstractos.
- Las subclases deben implementar los métodos abstractos.

Interfaces:

- Cuando necesitas herencia múltiple
- Cuando defines un contrato para clases no relacionadas
- Para desacoplar código (principio de inversión de dependencias)

Abstract

```
interface Volador {
    void volar();
}

interface Nadador {
    void nadar();
}

class Pato implements Volador, Nadador {
    public void volar() { System.out.println("Vuelo bajo"); }
    public void nadar() { System.out.println("Nado en el lago"); }
}
```

interface

```
interface Volador {
    void volar();
}

interface Nadador {
    void nadar();
}

class Pato implements volador, nadador {
    public void volar() { System.out.println("vuelo bajo"); }
    public void nadar() { System.out.println("nado en el lago"); }
}
```

Polimorfismo

Definición:
Capacidad de un objeto de tomar muchas formas.
Para qué sirve: Permite que un mismo método se comporte de manera diferente según el objeto que lo ejecute.

Interfaces

Definición: Contrato que define métodos (sin implementación).
Palabra clave: implements
Propósito: Lograr polimorfismo y herencia múltiple.
Características:

- Solo tienen métodos abstractos (implícitamente).
- Una clase puede implementar varias interfaces.
- Obligan a implementar todos los métodos definidos.

Interfaces:

- Cuando necesitas herencia múltiple
- Cuando defines un contrato para clases no relacionadas
- Para desacoplar código (principio de inversión de dependencias)