

## UNIVERSIDAD DE LAS FUERZAS ARMADAS "ESPE"

**NOMBRES:**Meza Josè

**FECHA:**30/06/2024

**TEMAS:** HERENCIA\_POLIMORFISMO\_JSON

### Herencia

#### Concepto

Herencia es un principio fundamental en la programación orientada a objetos (POO) que permite a una clase (llamada subclase o clase derivada) heredar propiedades y comportamientos (atributos y métodos) de otra clase (llamada superclase o clase base). Esto promueve la reutilización del código y facilita la creación de relaciones jerárquicas entre clases.

#### Ventajas

- **Reusabilidad:** Permite reutilizar el código existente, evitando la duplicación.
- **Mantenibilidad:** Facilita el mantenimiento del código, ya que los cambios en la superclase se reflejan automáticamente en las subclases.
- **Organización:** Ayuda a organizar y estructurar el código de manera lógica y coherente.

#### Nomenclatura

- **Superclase:** La clase base de la que se heredan los atributos y métodos.
- **Subclase:** La clase derivada que hereda los atributos y métodos de la superclase.
- **Herencia simple:** Una subclase hereda de una sola superclase.
- **Herencia múltiple:** Una subclase hereda de múltiples superclases (no soportada por todos los lenguajes).

#### Reglas y Modelado

- En UML, la herencia se representa con una línea con una flecha abierta que apunta de la subclase a la superclase.
- Una subclase puede heredar métodos y atributos de su superclase y también puede sobrescribir o extender estos métodos y atributos.

### Polimorfismo

#### Concepto

Polimorfismo es otro principio fundamental en POO que permite que objetos de diferentes clases se manejen a través de la misma interfaz. Esto se logra mediante la sobrescritura y la sobrecarga de métodos.

## **Ventajas**

- **Flexibilidad:** Permite que el mismo código trabaje con diferentes tipos de objetos.
- **Reusabilidad:** Facilita la reutilización del código mediante interfaces comunes.
- **Extensibilidad:** Permite extender el comportamiento del programa sin modificar el código existente.

## **Sobrecarga de Métodos**

Sobrecarga se refiere a definir múltiples métodos con el mismo nombre pero diferentes parámetros dentro de la misma clase. La sobrecarga de métodos permite a una clase tener múltiples versiones de un método, cada una con un conjunto diferente de parámetros.

## **Sobrescritura de Métodos**

Sobrescritura se refiere a redefinir un método en una subclase que ya está definido en la superclase. Esto permite que la subclase proporcione una implementación específica de un método que ya existe en su superclase.

## **Asignación de Objetos a Variables de su Superclase**

Una variable de tipo de superclase puede referirse a un objeto de una subclase. Esto es crucial para lograr el polimorfismo, ya que permite que el mismo código trate objetos de diferentes subclases de manera uniforme.

## **JSON (JavaScript Object Notation)**

### **Concepto**

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos que es fácil de leer y escribir para los humanos y fácil de parsear y generar para las máquinas. JSON se utiliza comúnmente para la transmisión de datos en aplicaciones web (por ejemplo, entre un servidor y un cliente web).

### **Ventajas**

- **Simplicidad:** JSON es simple y fácil de entender tanto para los desarrolladores como para las máquinas.
- **Compatibilidad:** JSON es compatible con la mayoría de los lenguajes de programación modernos.
- **Eficiencia:** JSON es ligero y rápido de parsear y generar.

### **Sintaxis**

La sintaxis de JSON es un subconjunto de la notación de objetos de JavaScript:

- **Objetos:** Conjunto de pares clave/valor encerrados entre llaves `{ }`.
- **Arrays:** Lista ordenada de valores encerrados entre corchetes `[ ]`.
- **Valores:** Cadena de texto, número, `true`, `false`, `null`, objeto, o array.