



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



PROGRAMACIÓN ORIENTADA A OBJETOS

PROFESOR: Ing. Yadira Franco R

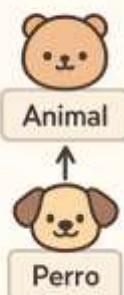
PERÍODO ACADÉMICO: 2025-B

TALLER

TÍTULO:

INVESTIGACIÓN Y PRACTICA

Herencia simple



Herencia jerárquica



Estudiante

Milka Borja

2025 B

Preguntas y Esquemas sobre Herencia en Java (POO)

Preguntas Iniciales para Activar Conocimientos

- ¿Cómo definimos una clase padre y una clase hija en Java?

Una clase padre contiene atributos y métodos generales; una clase hija hereda esos elementos usando la palabra clave extends.

- ¿Es lo mismo una superclase que una subclase? Explica.

No. La superclase es la clase base o padre, mientras que la subclase es la que hereda sus características.

- ¿Qué significa heredar en programación orientada a objetos?

Significa que una clase puede reutilizar atributos y métodos definidos en otra clase sin volver a escribirlos.

- ¿Se puede heredar de una subclase? ¿Por qué sí o por qué no?

Sí. Java permite la herencia multinivel, donde una clase puede heredar de otra que ya es subclase.

- ¿Qué tipos de herencia existen en Java y cómo se representan?

Simple: una clase hereda de otra.

Multinivel: una clase hereda de una subclase.

Jerárquica: varias clases heredan de una misma clase padre.
(No existe herencia múltiple de clases en Java).

Preguntas de Respuesta Directa

- ¿Qué palabra clave se utiliza en Java para indicar que una clase hereda de otra?

Se usa la palabra extends.

Ejemplo: class Estudiante extends Persona {}

- ¿Qué es la herencia en Java y para qué se utiliza?

Es un mecanismo que permite reutilizar código, compartiendo atributos y métodos entre clases relacionadas.

- ¿Puede una clase en Java heredar de más de una clase? Explica por qué.

No. Java no permite herencia múltiple para evitar conflictos entre métodos o atributos con el mismo nombre.

- ¿Qué sucede si una clase hija define un método con el mismo nombre que la clase padre?

La clase hija sobrescribe el método, reemplazando el comportamiento del método de la clase padre.

- ¿Cuál es la diferencia entre herencia y composición en POO?

Herencia: relación “es un” (extends).

Composición: relación “tiene un” (una clase contiene un objeto de otra).

- ¿Cuál es la función de la palabra clave super en una subclase? Da un ejemplo.

Se usa para llamar al constructor o métodos de la clase padre.

```
super(); // llama al constructor del parent
```

```
super.metodoPadre();
```

- ¿Se puede heredar un constructor en Java? ¿Por qué sí o por qué no?

No, los constructores no se heredan, pero se pueden invocar desde la subclase con super().

- ¿Qué significa sobrescribir un método (override)? ¿Cuál es la anotación que se usa para ello?

Es redefinir un método heredado para cambiar su comportamiento.

Se usa la anotación @Override.

- ¿Cuál es la diferencia entre herencia de implementación y herencia de interfaz en Java?

Implementación: hereda código ya escrito (entre clases).

Interfaz: hereda solo las declaraciones de los métodos (sin implementación).

- Imagina que estás desarrollando un sistema de vehículos. ¿Cómo organizarías las clases para aprovechar la herencia? ¿Qué clases serían padres y cuáles hijos?

Tendría una clase padre Vehiculo con métodos como encender() o apagar(), y clases hijas como Auto o Moto que agreguen comportamientos específicos.

- ¿Qué problemas podrían surgir si se abusa de la herencia en un proyecto grande? ¿Cómo podrías evitarlos?

Podría generar estructuras rígidas y difíciles de mantener. Se evita usando **composición** o interfaces cuando sea más adecuado.

- ¿Crees que la herencia promueve la reutilización del código? ¿Cómo lo has comprobado en tus propias prácticas o ejercicios?

Sí. Permite usar el mismo código en varias clases hijas sin repetirlo, por ejemplo al compartir atributos comunes como nombre o edad entre Persona, Estudiante y Empleado.