



UNIVERSIDAD TECNOLÓGICA DEL PERÚ
Facultad de Ingeniería

Programacion Orientada a Objetos

Sesión 5: Herencia simple y Herencia múltiple



Agenda

- Introducción a la herencia
- Herencia simple
- Ejemplos de herencia simple
- Herencia múltiple
- Ejemplos de herencia múltiple

Logros

- Entender la importancia de la herencia en la programación.
- Diferenciar entre herencia simple y múltiple.
- Identificar casos de uso para cada tipo de herencia.

Conocimientos previos



- Conocimientos básicos de programación orientada a objetos.
- Entender la relación entre clases y objetos.
- Conocer la diferencia entre clases base y derivadas.

Introducción a la herencia

- Concepto fundamental en programación orientada a objetos.
- Permite que una clase herede atributos y métodos de otra clase.
- Facilita la reutilización de código y establece una relación entre superclase y subclase.



Herencia simple

- Una clase deriva de una sola superclase.
- Permite extender la funcionalidad de una clase existente sin modificarla.





Ejemplos de herencia simple

- Clase 'Vehículo' y clase 'Coche'.
- Clase 'Persona' y clase 'Estudiante'.

Herencia múltiple

- Una clase puede heredar atributos y métodos de más de una superclase.
- No es soportada en todos los lenguajes de programación debido a problemas como la 'ambigüedad del diamante'.





Ejemplos de herencia múltiple

Realizar el ejercicio en BlueJ y NetBeans

Clase Vehículo:

Atributos: marca (String), modelo (String), año (int)

Métodos: arrancar(), detener()

Clase Coche (hereda de Vehículo):

Atributos adicionales: numeroPuertas (int),
tipoCombustible (String)

Métodos adicionales: acelerar(), frenar()

Clase Motocicleta (hereda de Vehículo):

Atributos adicionales: cilindrada (int), tipoMoto (String)

Métodos adicionales: girar(), equilibrar()

Clase Bicicleta:

Atributos: tipoBici (String), materialMarco (String)

Métodos adicionales: pedalear(), frenar()

Clase MotoBicicleta (hereda de Motocicleta y Bicicleta, lo que representa herencia múltiple):

No tiene atributos adicionales

Métodos adicionales: pedalear(), acelerar(), frenar(), girar()



Conclusiones

- La herencia es una herramienta poderosa en programación orientada a objetos.
- La herencia múltiple puede ser compleja y no es soportada universalmente.
- Es esencial elegir el tipo adecuado de herencia según las necesidades del proyecto.