



Recibe una cálida:

# ¡Bienvenida!

---

Te estábamos esperando 😊 

# ➤ Polimorfismo y principios básicos de diseño

---

**Plan formativo:** Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

# HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



# REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

- ✓ Herencia
- ✓ Atributos y constructores en la herencia

# LEARNING PATHWAY

4.6

Start! 🏁

Polimorfismo y  
principios básicos de  
diseño

Herencia y métodos

Veterinaria 2.0

Herencia y Métodos

# OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



**Comprender la implementación de la herencia de métodos en la programación orientada a objetos.**





# Rompehielo

## Contexto:

Imaginen una empresa donde los empleados heredan el método **trabajar()** de una clase **Persona**. Pero los empleados **trabajan de diferente forma**: los gerentes supervisan, los vendedores venden, etc.



## Respondan en el chat o levantando la mano:

¿Cómo creen que se puede modelar esta situación donde todos heredan un método pero lo implementan de forma particular?



# › Herencia y métodos





# Herencia y métodos



## Repasemos algunas características de la Herencia:

- Una subclase hereda de la superclase sus componentes (atributos y **métodos**).
- Los constructores no se heredan.
- Una subclase puede acceder a los miembros públicos y protegidos de la superclase como si fuesen miembros propios.
- Una subclase puede añadir a los miembros heredados, sus propios atributos y métodos (extender la funcionalidad de la clase).
- **También puede modificar los métodos heredados.**
- Una subclase puede, a su vez, ser una superclase, dando lugar a una **jerarquía de clases**.





# Herencia y métodos



## ¿Cómo se comportan los métodos en la Herencia?: Sobreescritura

Los métodos heredados pueden ser redefinidos en las clases hijas. Este mecanismo se denomina sobreescritura. La sobreescritura permite a las clases hijas utilizar un método definido en la superclase.



Una subclase sobreescrive un método de su superclase cuando define un método con las mismas características (nombre, número y tipo de argumentos) que el método de la superclase. Las subclases emplean la sobreescritura de métodos la mayoría de las veces para agregar o modificar la funcionalidad del método heredado de la clase padre.





# Herencia y métodos

## Sobreescritura: **@Override**

La sobreescritura permite que las clases hijas sumen sus métodos en torno al funcionamiento, y esto se logra escribiendo la anotación **@Override** arriba del método que queremos sobreescribir. El método **debe llamarse igual** en la subclase como en la superclase.



```
@Override //indica que se modifica un método heredado  
public void leer(){  
}
```

Cuando en una subclase se redefine un método de una superclase, se oculta el método de la clase padre y todas las sobrecargas del mismo en la clase padre. Por eso para ejecutar el método leer() se debe escribir **super.leer();**

# Evaluación Integradora ✨

¿Listos para un nuevo desafío? En esta clase comenzamos a construir nuestro...

## Trabajo Integrador del Módulo 💪

Iremos completándolo progresivamente clase a clase.



# LIVE CODING

Ejemplo en vivo

**¡Probando los métodos!:**

*Vamos a trabajar sobre la clase **Cuenta** con los métodos **depositar()** y **retirar()**.*

*Luego, crearemos las **subclases** CuentaPesoCL y CuentaUSD que implementarán estos métodos realizando el cambio de moneda correspondiente.*

*Vamos a mostrar por pantalla los resultados para verificar.*

**Tiempo: 30 minutos**



# **Ejercicio N°** **Veterinaria 2.0**



# Veterinaria 2.0

## Consigna 🛠️:

Seguiremos trabajando sobre las clases Animal y Perro creadas en la clase anterior!

1- Declara los siguientes métodos en la superclase Animal: comer(), dormir() y emitirSonido().

2- La subclase Perro tendrá el siguiente comportamiento:

comer(): llama al método comer() de la superclase Animal

dormir(): llama al método dormir() de la superclase

emitirSonido(): return "Guau";



# Veterinaria 2.0

3- Declara otra subclase Gato que también hereda de Animal e implementa:

comer(): llama al método comer() de la superclase Animal

dormir(): llama al método dormir() de la superclase

emitirSonido(): return "Miau";

## En el método main:

Finalmente, crea objetos de tipo Perro y Gato y llama a sus métodos para verificar que el llamado a los métodos y su sobreescritura estén bien implementados!

**Tiempo** : **25 minutos**



○

# ¿Alguna consulta?

+



# RESUMEN

¿Qué logramos en esta clase?

- ✓ Reconocer la importancia de la sobrescritura en los métodos heredados
- ✓ Comprender la implementación de la herencia en métodos

# #WorkingTime

Continuemos ejercitando

**¡Antes de cerrar la clase!** Te invitamos a: 🙌 🙌 🙌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
  - a. Material 1 (Foro)
  - b. *Lectura Módulo 4, Lección 6: página 3*
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

# ¡Muchas Gracias!

Nos vemos en la próxima clase 🙌



Momento: ✚

# Time-out!

🕒 5 min.

