



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 :

✓ Clases y métodos abstractos



LEARNING PATHWAY

4.6

Start! 🏁

Polimorfismo y
principios básicos de
diseño

Interfaces

Implementando

Interfaces

OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



Comprender la estructura e implementación de las Interfaces en Java





Rompehielo 🧊



Respondan en el chat o levantando la mano: 🙌

- ¿Que ven en la imagen?
- ¿Para qué sirve?

¿Todos los dispositivos se conectan igual?





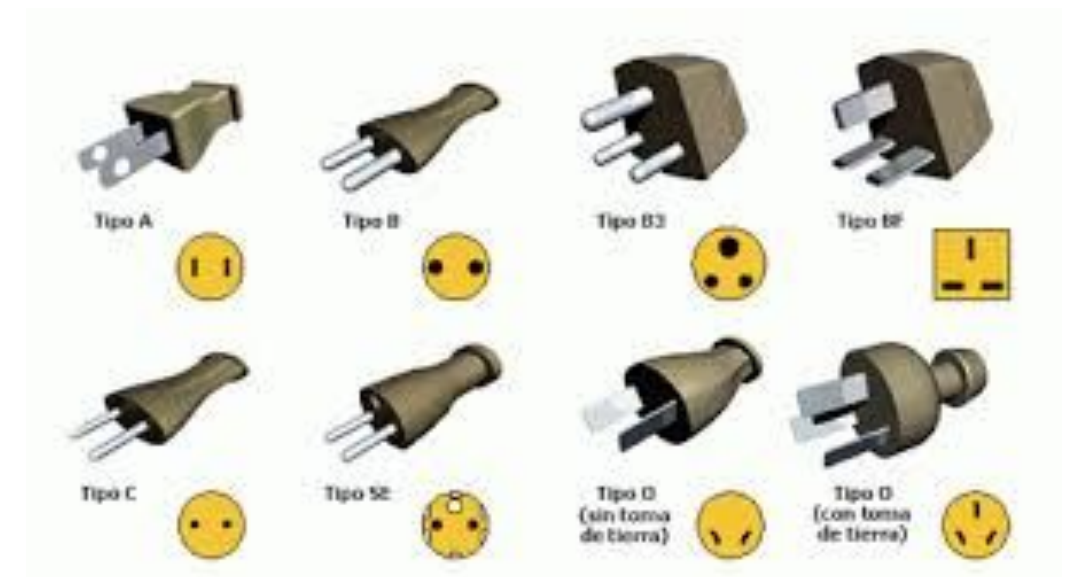
Rompehielo 🧊

Entonces podemos decir que el enchufe define cómo se realiza la conexión eléctrica. Y los cables/adaptadores implementan esa conexión según el dispositivo.

Así introducimos la idea de una **interfaz** como un "contrato" genérico que diferentes implementaciones concretas cumplen a su manera.

Respondan en el chat o levantando la mano: 🙋

¿Qué otras “interfaces” de la vida real conocen?



› Interfaces



Interfaces

¿Qué son y para qué se utilizan?:



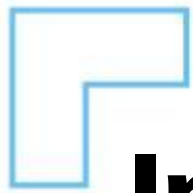
Una interfaz **es una especie de plantilla** para la construcción de clases. Normalmente una interfaz **se compone de un conjunto de declaraciones de cabeceras de métodos** (sin implementar, de forma similar a un método abstracto) que especifican un protocolo de comportamiento para una o varias clases.



Una clase puede implementar una o varias interfaces. Por otro lado, una interfaz puede emplearse también para declarar constantes que luego puedan ser utilizadas por otras clases.

Las interfaces no pueden definir atributos salvo que estos sean estáticos o constantes; es decir, "**static**" o "**final**".





Interfaces

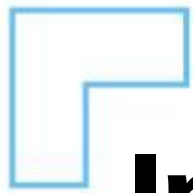
Entonces, ¿en qué se diferencian de una clase abstracta?



Una interfaz puede parecer similar a una clase abstracta, pero existen una serie de diferencias como las que mostramos a continuación:

- Todos los métodos de una interfaz se declaran **implícitamente** como abstractos y públicos.
- Una clase abstracta no puede implementar los métodos declarados como abstractos, **una interfaz no puede implementar ningún método** (ya que todos son abstractos).
- Una interfaz **no declara variables de instancia**.
- Una clase puede implementar varias interfaces, pero sólo puede tener una clase ascendiente directa.
- Una clase abstracta pertenece a una jerarquía de clases mientras que una interfaz no pertenece a una jerarquía de clases. En consecuencia, **clases sin relación de herencia pueden implementar la misma interfaz**.





Interfaces

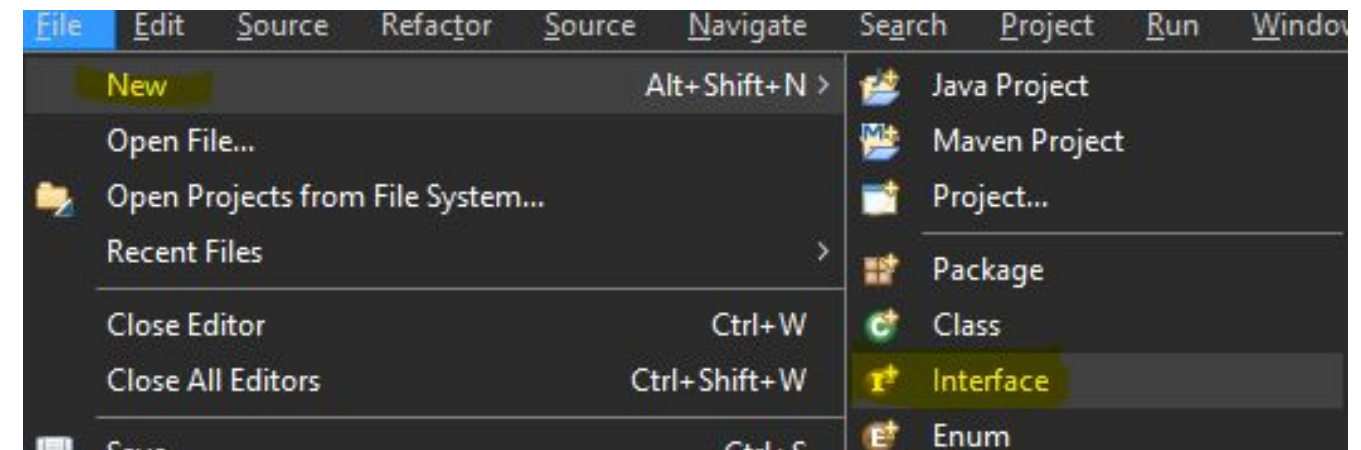
Declaración de una Interfaz

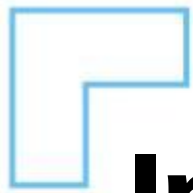


La declaración de una interfaz es similar a una clase, aunque emplea la palabra reservada **interface** en lugar de class y **no incluye ni la declaración de variables de instancia ni la implementación del cuerpo de los métodos** (sólo las cabeceras). La sintaxis de declaración de una interfaz es la siguiente:



```
public interface NombreInterfaz {  
    // Cuerpo de la interfaz ...  
}
```





Interfaces

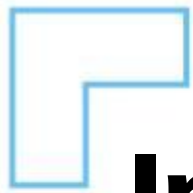
Declaración de métodos y constantes:

Las constantes siempre son finales y se deben inicializar en la misma línea.
Los métodos son públicos y con declarar el tipo de retorno alcanza.



```
public interface MilInterfaz {  
    public final double CONSTANTE= 358,25;  
    public final int ENTERO= 125;  
    void put ( int dato );  
    int get (Object object);
```





Interfaces



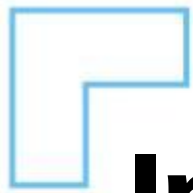
¿Cómo se implementan?

Para declarar una clase que implemente una interfaz es necesario utilizar la palabra reservada **implements** en la cabecera de declaración de la clase.



```
public class Clase implements NombreInterfaz {  
    // Cuerpo de la interfaz ...  
}
```





Interfaces

```
public interface Interfaz {  
    public void metodo();  
    public int sumar();  
}  
  
class Clase implements Interfaz {  
    @Override  
    public void metodo() {  
        System.out.println("Implementacion del método");  
    }  
  
    @Override  
    public int sumar() {  
        int suma = 2 + 2;  
        return suma;  
    }  
}
```

Veamos un ejemplo codificado:

Como podemos ver en la interfaz teníamos dos métodos sin cuerpo y al implementar la interfaz en nuestra clase, los sobrescribimos y les dimos una funcionalidad a dichos métodos.

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

Creando Interfaces: Vamos a modificar el proyecto BilleteraVirtual para hacer uso de las interfaces.

1. *Crear una interfaz **Moneda** con los métodos `getSimbolo()` y `getFactorConversion()`.*
2. *Crear una clase **Dolar** que implemente la interfaz **Moneda**.*
3. *Crear una clase **Euro** que implemente la interfaz **Moneda**.*
4. *Crear un método **convertir()** para la interfaz **Moneda** que permita convertir fondos de una moneda a otra.*

Tiempo: 30 minutos



Ejercicio N° 1

Implementando



Implementando



Intercambiando la clase **Animal** por una interfaz: 🙌

Vamos a intercambiar el ejercicio que veníamos realizando con la clase padre **Animal**, por una interfaz para ver su comportamiento.



Consigna: 📝

Dada la interfaz **Animal** con método abstracto **emitirSonido()**, implementarla en clases **Perro**, **Gato** y **Vaca**.

Tiempo 🕒: 30 minutos

○

¿Alguna consulta?

+



RESUMEN

¿Qué logramos en esta clase?

- ✓ **Comprender la implementación de las Interfaces en Java.**



#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: páginas 5 - 6*
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.

