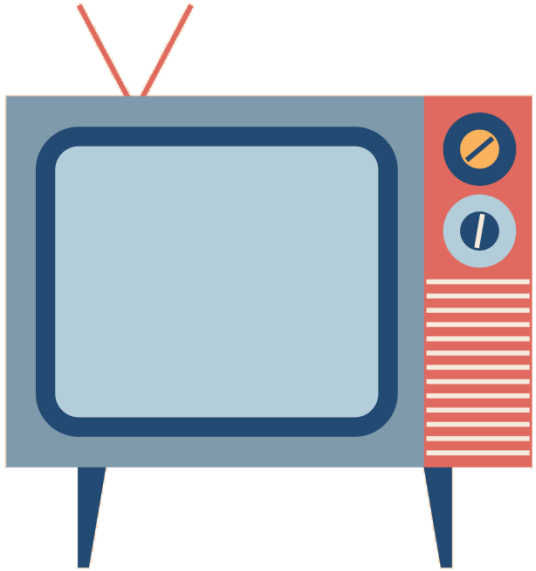
Paradigmas de la Programación

Interface

¿Qué es una Interfaz?

Los objetos definen su interacción con el mundo exterior a través de sus propios métodos incorporados. Gracias a los métodos se forma la interfaz del objeto.

Por ejemplo, los botones en la parte frontal del televisor son una interfaz entre uno y el cableado eléctrico. Pudiendo presionar el botón de "Encendido" para activar o apagar el televisor, por lo tanto, el botón de "**encendido**" sería el método de comunicación con el mundo exterior.



Designed by balasoiu / Freepik

UNIVERSIDAD DEL BÍO-BÍO

Interfaz

Al igual que una clase, una interfaz puede contener una serie métodos y variables, pero los métodos declarados en la interfaz son por defecto abstractos (solo la firma del método, no el cuerpo). En el caso de las variables serán siempre públicas, estáticas y finales.
También se debe tener en consideración que las interfaces no son instanciables (Exceptuando algunos comportamientos especiales).

Para poder usar una interfaz, una clase deberá implementarla ocupando la palabra clave **implements** seguido del nombre de la interfaz.

Declaración e Implementación

Interfaz

```
public interface IProceso {

    public void play();

}
```

Declaración de método sin cuerpo

```
public class Reproductor implements IProceso {

    @Override
    public void play() {
        System.out.println("Ejecutando...");
    }

}
```

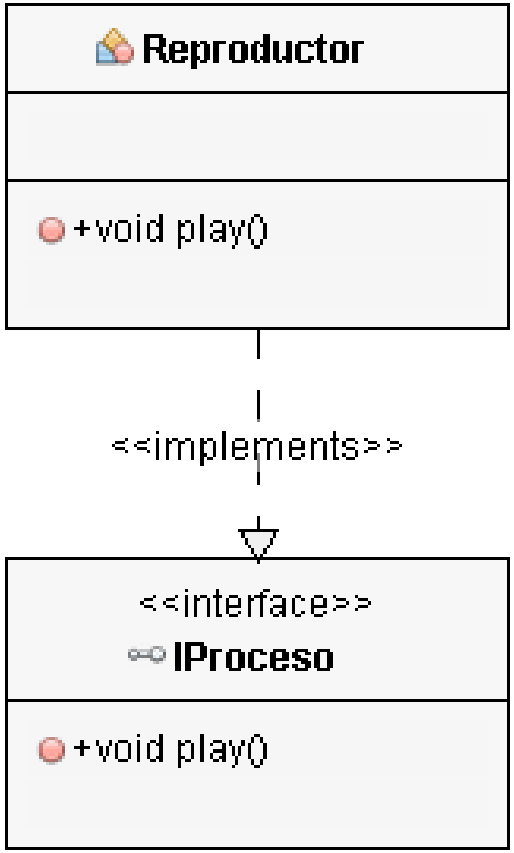
Implementando método de Interfaz

Implementando Interfaz

Observación: Cuando se implementa una interfaz se debe incluir todos los métodos que ella contenga **sobrescribiéndolos**, de lo contrario no se podrá compilar.

UNIVERSIDAD DEL BÍO-BÍO

UML



¿Cuándo usar una Interfaz?

Es necesario utilizar una interfaz cuando un rol deba definirse para las clases, **independientemente del árbol de herencia** entre ellas (No importa qué tan común sean).

Se necesita estandarizar un comportamiento

Designed by rawpixel.com / Freepik

UNIVERSIDAD DEL BÍO-BÍO

Ejemplo

Supongamos que tenemos 3 clases.

1. Robot extiende de Máquina
2. Gato extiende de Animal
3. Fantasma extiende de Paranormal

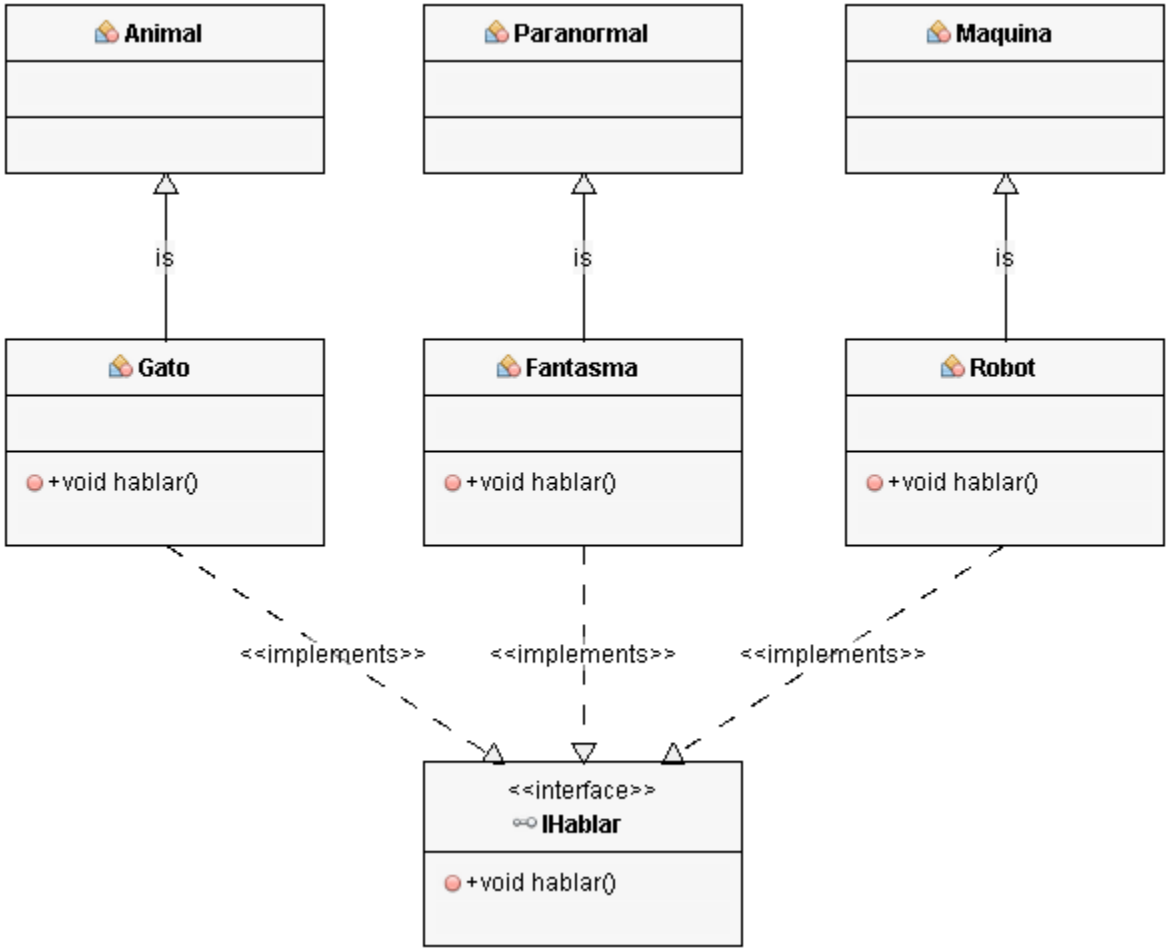
Y existe un proceso que requiere que hablen en X circunstancia.

Observación: En Java las clases máximo pueden tener una clase Padre y dichas clases ya contienen uno, por lo que no se puede crear una clase común entre ellas(Aunque se pudiera no garantizará la ejecución correcta del método, ya que puede haber más de alguna clase que no sobre-escribirá el método del padre).

¿Cómo podemos asegurarnos que cada clase implemente dicho método en común?

UNIVERSIDAD DEL BÍO-BÍO

Solución – Implementando Interfaz



UNIVERSIDAD DEL BÍO-BÍO

Solución – Implementando Interfaz

```java
public interface IHablar {

    public void hablar();

}
```

```java
public class Robot extends Maquina implements IHablar {

    @Override
    public void hablar() {
        System.out.println("Biipbipp!");
    }

}

public class Gato extends Animal implements IHablar {

    @Override
    public void hablar() {
        System.out.println("Miaauuu!");
    }

}

public class Fantasma extends Paranormal implements IHablar{

    @Override
    public void hablar() {
        System.out.println("BuuuUuuuuu!");
    }

}
```

```java
public class Maquina {

}

public class Animal {

}

public class Paranormal {

}
```

UNIVERSIDAD DEL BÍO-BÍO

Solución – Implementando Interfaz

Ahora que está todo bajo una Interfaz, se podrá agrupar en una lista de tipo **IHablar**, gracias al **polimorfismo**.

```
public class Main {

    public static void main(String[] args) {

        ArrayList<IHablar> habladores = new ArrayList<>();

        habladores.add(new Fantasma());
        habladores.add(new Robot());
        habladores.add(new Gato());

        for (int i = 0; i < habladores.size(); i++)
            habladores.get(i).hablar();

    }

}
```

```
run:
BuuuUuuuuu!
Biipbipp!
Miaauuu!
BUILD SUCCESSFUL (total time: 5 seconds)
```

Herencia en Interfaces

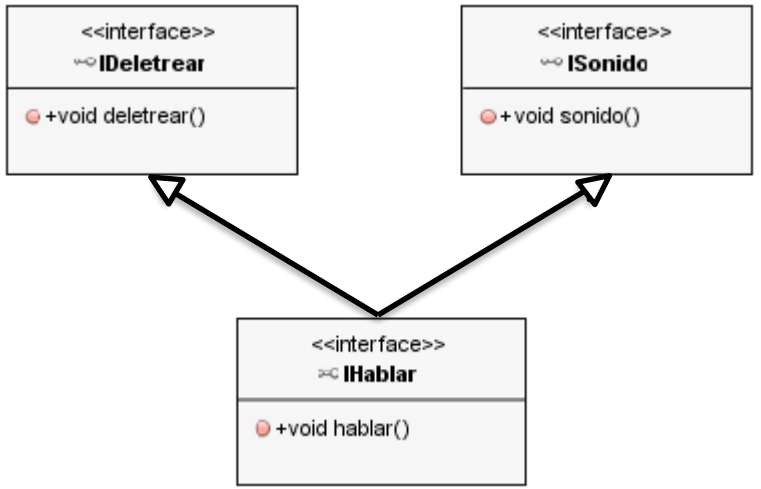
Al igual que una Clase, una interfaz heredará todos los métodos y atributos de sus Padres (Puede ser más de uno).

Observación 1: Una Interfaz puede ser Hija de N interfaces.
Observación 2: Las interfaces extienden de otras interfaces **no las implementan**.

```
public interface IHablar extends IDeletrear, ISonido{

    public void hablar();

}
```



UNIVERSIDAD DEL BÍO-BÍO

Resumen

1. En una interfaz las variables son por defecto públicas, estáticas y finales.
2. Para los método en una interfaz se declarará solamente la firma (Sin cuerpo).
3. Las interfaces son un excelente medio para simular la multi-herencia en Java.
4. No es posible instanciar una interfaz.
5. Una clase puede **implementar** N interfaces.
6. Una interfaz puede **extender** de N interfaces.
7. Si una clase implementa una Interfaz, esta deberá implementar todos los métodos, de lo contrario no se podrá compilar.