



HERENCIA Y POLIMORFISMO

EJEMPLO ANIMALES

Java Básico

Crear el proyecto Animales.

2

- ❑ Crear la clase abstracta **Animal**.
- ❑ Todos los animales tienen especie. Cuando se crean, se debe especificar.
- ❑ Cada animal habla dependiendo la especie, por eso es abstracta.

```
public abstract class Animal {  
  
    String especie;  
  
    public Animal (String n) {  
        especie = n;  
    }  
  
    @Override  
    public String toString () {  
        return especie;  
    }  
  
    public abstract void hablar ();  
}
```

Agregamos un animal al proyecto.

3

- Cuando un perro se crea, se avisa a la clase superior (Animal) la especie del perro.
- Además, el perro habla como perro.

```
public class Perro extends Animal {  
  
    public Perro () {  
        super ("Perro");  
    }  
  
    @Override  
    public void hablar() {  
        System.out.println ("Guau Guau!!");  
    }  
}
```

Agregar la interfaz Domesticado

4

- Todos los animales que estén domesticados, saben hacer algún truco.

```
public interface Domesticado {  
  
    public void truco ();  
}
```

Agregamos un perro domesticado.

5

- Los perros domesticados son perros, y hablan como perros.
- Pero también saben hacer algún truco.

```
public class PerroDomesticado extends Perro implements Domesticado{  
  
    @Override  
    public void truco() {  
        System.out.println ("Se dar la pata.");  
    }  
}
```

Probando las clases

6

- Creamos un perro y lo hacemos hablar.

```
public static void main(String[] args) {  
  
    Perro manchas = new Perro ();  
    manchas.hablar();  
}
```

Probamos la herencia.

7

- Un perro domesticado, también se comporta como perro.
- A eso se le llama Polimorfismo.

```
public static void main(String[] args) {  
  
    Perro manchas = new Perro ();  
    manchas.hablar();  
  
    Perro firulaais = new PerroDomesticado ();  
    firulaais.hablar ();  
}
```

Polimorfismo

8

- No le podemos pedir a firulais que haga truco, porque se está comportando como Perro.
- No todos los perros saben hacer trucos, solo los domesticados.

```
public static void main(String[] args) {  
  
    Perro manchas = new Perro ();  
    manchas.hablar();  
  
    Perro firulais = new PerroDomesticado ();  
    firulais.hablar ();  
  
    firulais.truco ();  
}
```


Polimorfismo

9

- Le decimos a firulais que se comporte como Perro Domesticado, y entonces que haga su truco.
- Eso se logra con un Cast.

```
public static void main(String[] args) {  
  
    Perro manchas = new Perro ();  
    manchas.hablar();  
  
    Perro firulais = new PerroDomesticado ();  
    firulais.hablar ();  
  
    ((PerroDomesticado) firulais).truco();  
}
```

Ejercicio

10

- ❑ Crear la clase Gato. Debe heredar de Animal.
- ❑ Crear la clase GatoDomesticado. Debe heredar de Gato e implementar la interfaz Domesticado.

```
public static void main(String[] args) {  
  
    Domesticado [] animales = new Domesticado[2];  
  
    animales[0] = new GatoDomesticado ();  
    animales[1] = new PerroDomesticado ();  
  
    for (Domesticado animal: animales){  
        System.out.print (animal+" mi truco es: ");  
        animal.truco();  
    }  
}
```