

Documentación código Animal

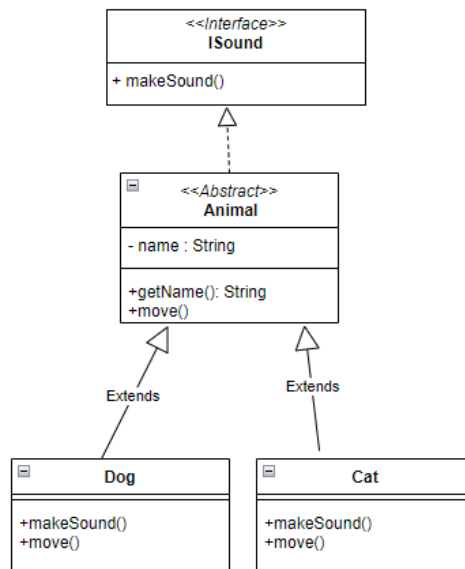
Rodrigo Martínez Zambrano

En el siguiente ejemplo se realiza un código utilizando conceptos básicos de la programación orientada a objetos como el polimorfismo y de java como las clases abstractas y las interfaces.

En el proyecto se tiene:

- Interfaz ISound
- Clase abstracta Animal
- Clase Dog
- Clase Cat

Si se realiza un diagrama de la estructura del proyecto para observar la jerarquía y herencia:



La interfaz únicamente define el método:

```
public interface ISound {
    void makeSound();
}
```

La clase abstracta **Animal** implementa la interfaz, define atributos y métodos en común para los animales que hereden de esta clase.

```

public abstract class Animal implements ISound{ 2 usages 2 i
    private String name; 2 usages

    public Animal(String name){ 2 usages
        this.name = name;
    }

    public String getName(){ 4 usages
        return name;
    }

    //Método que deber ser implementado por las subclases
    public abstract void move(); no usages 2 implementations
}

```

Posteriormente se crean dos clases hijas que heredan de Animal, por lo que deben implementar los métodos definidos tanto en la interfaz como en la clase abstracta.

```

public class Dog extends Animal{ 1 usage

    public Dog(String name){ 1 usage
        super(name);
    }

    //Método de la clase padre Animal
    @Override 2 usages
    public void move() {
        System.out.println(getName() + " corre!");
    }

    //Método de la interfaz
    @Override 2 usages
    public void makeSound() {
        System.out.println(getName() + " Woof Woof !!");
    }
}

```

```

public class Cat extends Animal{ 1 usage

    public Cat(String name){ 1 usage
        super(name);
    }

    //Método de la clase padre Animal
    @Override 2 usages
    public void move() {
        System.out.println(getName() + " salta!");
    }

    //Método de la interfaz
    @Override 2 usages
    public void makeSound() {
        System.out.println(getName() + " Miau Miau !!");
    }
}

```

Como se puede observar, ambas clases implementan los métodos de forma diferente.

En la clase Main es posible realizar el polimorfismo, esto se logra por el objeto que invoque el método, en este caso, cada animal realizará una acción diferente.

```
public class Main {  
    public static void main(String[] args){  
        //Se crean las instancias  
        Animal dog = new Dog( name: "Rocket");  
        Animal cat = new Cat( name: "Raknar");  
  
        //Se usa el polimorfismo con los métodos  
        dog.makeSound();  
        dog.move();  
        System.out.println("*****");  
        cat.makeSound();  
        cat.move();  
    }  
}
```

```
Rocket Woof Woof !!  
Rocket corre!  
*****  
Raknar Miau Miau !!  
Raknar salta!
```