

Polimorfismo en Java

El **polimorfismo** es otro concepto fundamental en OOP. Significa "muchas formas" y permite que los objetos tomen diferentes formas o comportamientos en función del contexto.

Existen dos tipos principales de polimorfismo en Java:

1. **Polimorfismo en tiempo de compilación (Sobrecarga de métodos).**
2. **Polimorfismo en tiempo de ejecución (Sobrescritura de métodos).**

1. Polimorfismo en tiempo de compilación (Sobrecarga de métodos)

La sobrecarga de métodos permite que una clase tenga múltiples métodos con el mismo nombre pero diferentes parámetros. El compilador decide cuál método ejecutar en función de los argumentos.

Ejemplo de sobrecarga:

```
class MathOperation {  
    public int add(int a, int b) {  
        return a + b;  
    }  
  
    public double add(double a, double b) {  
        return a + b;  
    }  
}
```

En este ejemplo, ambos métodos `add` tienen el mismo nombre pero diferentes tipos de parámetros, lo que constituye un ejemplo de **sobrecarga de métodos**.

2. Polimorfismo en tiempo de ejecución (Sobrescritura de métodos)

La sobrescritura (overriding) de métodos ocurre cuando una subclase proporciona una implementación específica de un método que ya está definido en su superclase. El método de la subclase reemplaza al de la superclase cuando se llama a través de un objeto de la subclase.

Ejemplo de sobrescritura:

```
class Animal {  
    public void makeSound() {  
        System.out.println("Animal hace un sonido");  
    }  
}  
  
class Dog extends Animal {  
    @Override
```

```

    public void makeSound() {
        System.out.println("El perro ladra");
    }
}

class Cat extends Animal {
    @Override
    public void makeSound() {
        System.out.println("El gato maúlla");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal myAnimal = new Animal();
        Animal myDog = new Dog();
        Animal myCat = new Cat();

        myAnimal.makeSound(); // Salida: Animal hace un sonido
        myDog.makeSound();    // Salida: El perro ladra
        myCat.makeSound();    // Salida: El gato maúlla
    }
}

```

En este ejemplo:

- `Dog` y `Cat` sobrescriben el método `makeSound()` de la clase `Animal`.
- Aunque `myDog` y `myCat` son declarados como objetos de tipo `Animal`, cuando se llama a `makeSound()`, se ejecutan las versiones de `Dog` y `Cat`, respectivamente. Esto es polimorfismo en tiempo de ejecución.