POLIMORFISMO

En programación orientada a objetos, el polimorfismo es la capacidad que tienen los objetos de una clase en ofrecer respuesta distinta e independiente en función de los parámetros (diferentes implementaciones) utilizados durante su invocación.

Ejemplos de Código

- Sobrecarga: El más conocido y se aplica cuando existen funciones con el mismo nombre en clases que son completamente independientes una de la otra.

```
class Animal {
  public void makeSound() {
    System.out.println("Grr...");
  }
}
class Cat extends Animal {
  public void makeSound() {
    System.out.println("Meow");
  }
}
class Dog extends Animal {
  public void makeSound() {
    System.out.println("Woof");
  }
}
```

```
public static void main(String[ ] args) {
  Animal a = new Dog();
  Animal b = new Cat();
}
```

```
a.makeSound();
//Outputs "Woof"

b.makeSound();
//Outputs "Meow"
```

- Paramétrico: Existen funciones con el mismo nombre pero se usan diferentes parámetros (nombre o tipo). Se selecciona el método dependiendo del tipo de datos que se envíe.

```
class Overload
{
    void demo (int a)
    {
       System.out.println ("a: " + a);
    }
    void demo (int a, int b)
    {
       System.out.println ("a and b: " + a + "," + b);
    }
    double demo(double a) {
       System.out.println("double a: " + a);
       return a*a;
    }
}
class MethodOverloading
{
    public static void main (String args [])
    {
        Overload Obj = new Overload();
        double result;
        Obj .demo(10);
        Obj .demo(10, 20);
        result = Obj .demo(5.5);
        System.out.println("O/P : " + result);
    }
}
```

```
a: 10
a and b: 10,20
double a: 5.5
O/P : 30.25
```

- Inclusión: Es cuando se puede llamar a un método sin tener que conocer su tipo, así no se toma en cuenta los detalles de las clases especializadas, utilizando una interfaz común.

```
abstract class Piece{
    public abstract void move(byte X, byte Y);
}

class Bishop extends Piece{
 @Override
  public void move(byte X, byte Y){

  }
}
```

Análisis Comparativo

Esto puede parecer un poco confuso pero en definitiva el Polimorfismo consiste en redefinir un método de una clase padre en una clase hija. Mientras que sobrecarga es definir un nuevo método igual que otro viejo, pero cambiando el tipo o la cantidad de parámetros.

- ¿Qué es el término firma?
Una firma es el nombre de un método más su lista de parámetros. Por lo tanto cada método en una misma clase, en términos de sobrecarga, obtiene una firma diferente.

- ¿Diferencias entre los términos Overloading y Overriding?

Overloading significa que un mismo método tiene diferentes firmas mientras que Overriding es el mismo método, por tanto misma firma, al que diferentes clases conectan a través de la herencia.

- ¿Se pueden sobrecargar métodos estáticos?
Sí, es posible tener dos más métodos estáticos con el mismo nombre siempre que se diferencien en los parámetros de entrada.

- ¿Es posible sobrecargar la clase main() en Java?
Sí, es posible siempre que definamos correctamente los parámetros de entrada como en el siguiente ejemplo.

```
class Demo{

    public static void main(String[] args) {
        System.out.println("Hello Folks"); // Hello Folks
        Demo.main("Ducks");
    }

    // Sobrecargando
    public static void main(String arg1) {
        System.out.println("Hello, " + arg1); // Hello Ducks
        Demo.main("Dogs","Cats");
    }
    public static void main(String arg1, String arg2) {
        System.out.println("Hello, " + arg1 + " and "  + arg2); // Hello Dogs and Cats
    }
}
```