

PARCIAL FINAL POO

AUTOR: JHON JAIRO LEGARDA ERAZO

DOCENTE: ADRIAN IZQUIERDO

INSTITUTO TECNOLÓGICO DEL PUTUMAYO

PROGRAMACIÓN ORIENTADA A OBJETOS

MOCOA - PUTUMAYO

2023

## Tabla de contenido

Parcial Final Punto 1 .....	3
1. Herencia en Programación Orientada a Objetos: .....	3
2. Polimorfismo en Java: .....	3
3. Diferencia entre Herencia Simple y Múltiple en Java: .....	3
4. Interfaz en Java con Método Default: .....	3
5. Sobreescritura de Métodos (Method Overriding) en Java: .....	4

## Parcial Final Punto 1

### 1. Herencia en Programación Orientada a Objetos:

La herencia en programación orientada a objetos es un concepto que permite a una clase heredar propiedades y comportamientos de otra clase. En Java, esto se logra mediante la palabra clave `extends`. Por ejemplo, si tenemos una clase "Vehiculo" y queremos crear una clase "Automovil" que herede de "Vehiculo", la declaración sería:

```
public class Automovil extends Vehiculo {

    // Código de la clase Automovil

}
```

### 2. Polimorfismo en Java:

El polimorfismo es la capacidad de una clase para tomar múltiples formas. En Java, el polimorfismo se puede lograr mediante el uso de interfaces y clases base. Un ejemplo práctico es el uso de un método con el mismo nombre en diferentes clases. Por ejemplo:

```
interface Animal {

    void hacerSonido();

}

class Perro implements Animal {

    public void hacerSonido() {

        System.out.println("Guau");

    }

}

class Gato implements Animal {

    public void hacerSonido() {

        System.out.println("Miau");

    }

}
```

### 3. Diferencia entre Herencia Simple y Múltiple en Java:

En Java, la herencia simple significa que una clase puede heredar de solo una clase base, mientras que la herencia múltiple implica heredar de varias clases base. Java no admite herencia múltiple directa de clases para evitar la complejidad y conflictos que podrían surgir. En cambio, Java utiliza interfaces para lograr cierta forma de herencia múltiple.

### 4. Interfaz en Java con Método Default:

Una interfaz en Java es una colección de métodos abstractos. A partir de Java 8, se pueden incluir métodos con implementación predeterminada utilizando la palabra clave `default`. Por ejemplo:

```
interface MiInterfaz {
    void metodoAbstracto();
    default void metodoDefault() {
        System.out.println("Este es un método default.");
    }
}
```

##### 5. Sobreescritura de Métodos (Method Overriding) en Java:

La sobreescritura de métodos en Java se refiere a la capacidad de una subclase para proporcionar una implementación específica de un método que ya está definido en su clase base. Esto se logra utilizando la misma firma de método en la clase derivada. El propósito principal es lograr el polimorfismo. Por ejemplo:

```
class Animal {
    void hacerSonido() {
        System.out.println("Sonido genérico de un animal");
    }
}

class Perro extends Animal {
    @Override
    void hacerSonido() {
        System.out.println("Guau");
    }
}
```