

Repaso – Conceptos de Programación Orientada a Objetos

¿Qué es POO?

La programación orientada a objetos (POO) es un paradigma de programación que parte del concepto de "objetos" como base, los cuales contienen información en forma de campos y código en forma de métodos.

¿Qué es un Objeto?

Un objeto es una abstracción de algún hecho o ente del mundo real, con atributos que representan sus características o propiedades, y métodos que emulan su comportamiento o actividad.

¿Qué es un atributo?

Un atributo es una variable que pertenece a una clase y representa las características o propiedades que describen el estado de un objeto. Estos atributos definen las características únicas de cada objeto creado a partir de esa clase.

¿Qué es un método?

Es una función que pertenece a una clase y representa el comportamiento o actividad de un objeto. Los métodos se utilizan para realizar tareas específicas, como acceder o modificar atributos, o realizar cálculos.

Por ejemplo, un objeto "Persona" podría tener los siguientes atributos:

Nombre, apellido, edad, sexo

Y los siguientes métodos:

caminar(), hablar(), comer()

¿Qué es una clase?

Una clase es un molde o plantilla a partir de la cual se crean objetos. Una clase define los atributos y métodos de los objetos que se crearán a partir de ella.

¿Qué es Herencia?

La herencia es el proceso por el cual una clase hereda los atributos y métodos de otra clase.

Conceptos Importantes en Herencia:

extends: Se utiliza para declarar que una clase hereda de otra clase. Por ejemplo, la siguiente declaración declara que la clase "Hijo" hereda de la clase "Padre":

```
class Hijo extends Padre {  
  
}
```

super: Se utiliza para acceder a los miembros de la clase padre desde una subclase. Por ejemplo, el siguiente código muestra cómo acceder al atributo "nombre" de la clase "Padre" desde la clase "Hijo":

```
class Hijo extends Padre {  
  
    public void metodo() {  
  
        System.out.println(super.nombre); // Imprime el nombre del objeto  
  
    }  
  
}
```

final: Se utiliza para declarar que una clase o un método no puede ser heredado. Por ejemplo, la siguiente declaración declara que la clase "Padre" no puede ser heredada:

```
final class Padre {  
  
}
```

¿Qué significa instanciar?

Instanciar significa **crear un objeto a partir de una clase**. El objeto creado tendrá los atributos y métodos definidos en la clase.

abstract: Se utiliza para declarar que una clase es abstracta, lo que significa que no puede ser instanciada. Las clases abstractas solo se pueden utilizar como superclases. Por ejemplo, la siguiente declaración declara que la clase "Padre" es abstracta:

```
abstract class Padre {  
  
    public abstract void metodo();  
  
}
```

implements: Se utiliza para declarar que una clase implementa una interfaz. Una interfaz es un conjunto de métodos que una clase debe implementar. Por ejemplo, la siguiente declaración declara que la clase "Hijo" implementa la interfaz "Interfaz":

```
class Hijo implements Interfaz {  
  
    public void metodo1() {  
  
        // Implementación del método metodo1() de la interfaz  
  
    }  
  
}
```

interface: Se utiliza para declarar una interfaz. Una interfaz es un conjunto de métodos que una clase debe implementar. Por ejemplo, la siguiente declaración declara la interfaz "Interfaz":

```
interface Interfaz {  
  
    public void metodo1();  
  
}
```