Logotipo

El contenido generado por IA puede ser incorrecto.Ingeniería en Sistemas Computacionales PROGRAMACION ORIENTADA A OBJETOS Oscar David Galvan Alvarez TUE0096

**CONCEPTOS DE PROGRAMACION ORIENTADA A OBJETOS**

**Investigación: Clases y Objetos en POO**

La Programación Orientada a Objetos (POO) es un paradigma que organiza el código en torno a entidades llamadas objetos, los cuales se basan en clases que definen sus características y comportamientos.

**1. Clases y Objetos**

Una clase es como un molde que describe qué atributos y acciones tendrá un objeto. Un objeto es una instancia concreta de esa clase, con datos propios y funciones que puede ejecutar.

**2. Declaración de Clases**

En Python, se usa la palabra clave class para definir una clase. Dentro de ella, el método especial \_\_init\_\_ se encarga de inicializar los atributos cuando se crea un objeto.

**3. Atributos**

Son las propiedades que describen el estado de un objeto. Se definen normalmente en el constructor (\_\_init\_\_) y se accede a ellos mediante self.

**4. Métodos**

Son funciones internas de una clase que definen el comportamiento del objeto. Se escriben como cualquier función, pero siempre reciben self como primer parámetro.

**5. Encapsulamiento**

Este principio busca proteger los datos internos de una clase, exponiendo solo lo necesario. En Python, se usa el prefijo \_ o \_\_ para indicar que un atributo es protegido o privado.

**6. Instanciación**

Crear un objeto a partir de una clase se llama instanciación. Cada instancia tiene sus propios valores para los atributos definidos.

**7. Herencia**

Permite que una clase hija reutilice atributos y métodos de una clase padre. Esto facilita la extensión del código sin duplicarlo.

**8. Clases Base y Derivadas**

La clase base es la que proporciona funcionalidad común. La clase derivada hereda esa funcionalidad y puede modificarla o ampliarla.

**9. Herencia Simple y Múltiple**

* **Simple**: una clase hereda de una sola clase.
* **Múltiple**: una clase hereda de varias clases, combinando sus funcionalidades.

**10. Polimorfismo**

Consiste en que diferentes clases pueden tener métodos con el mismo nombre, pero comportamientos distintos. Esto permite tratar objetos distintos de forma uniforme.

**11. Clases Abstractas**

Son clases que no se pueden instanciar directamente y que definen métodos que deben ser implementados por las clases hijas. En Python se usan con el módulo abc.