

NOMBRE: LUIS FERNANDO CUEVA FLORES

MATERIA: DESARROLLO WEB AVANZADO

PROFESOR: DIEGO MEDARDO SAAVEDRA GARCIA

¿Cuál es la diferencia entre una variable y una propiedad en C#?

una variable es un espacio de almacenamiento que guarda un valor(int, double,float,string)

las propiedades permiten un control más preciso sobre la lectura y escritura de datos

Explica cómo funciona la estructura if y por qué es útil en programación.

La estructura de control if-else permiten ejecutar diferentes bloques de código según si se cumple o no una condición que retorna un valor booleano para saber si cumple la condición.

¿Qué ventajas ofrece la programación orientada a objetos en comparación con otros paradigmas de programación?

La programación orientada a objetos (POO) es ventajosa por su capacidad de reutilización eficiente del código a través de herencia y composición, simplificando el diseño y mejorando la modularidad del software. Al abstraer conceptos del mundo real mediante la creación de clases, oculta detalles complejos y facilita el desarrollo. La encapsulación mejora la seguridad y el mantenimiento al ocultar la implementación interna, y el polimorfismo proporciona adaptabilidad y flexibilidad, acelerando el desarrollo y facilitando el mantenimiento del código en proyectos complejos.

¿Cuándo usarías la herencia en un diseño de clases?

La herencia en POO se emplea cuando se busca crear una nueva clase que extienda y aproveche las características de una clase existente. En este proceso, la nueva clase hereda propiedades y métodos de la clase base, permitiendo la reutilización eficiente del código. La herencia es valiosa para representar relaciones jerárquicas entre clases y compartir comportamientos comunes.

¿Por qué es importante la encapsulación en programación orientada a objetos?

La encapsulación en POO es importante porque permite ocultar los detalles internos de una clase y solo revelar la información necesaria para interactuar con objetos. Este concepto simplifica la gestión del código, mejora la modularidad y facilita las actualizaciones al controlar el acceso a datos y métodos. Además, contribuye a la seguridad del código y promueve cambios sin afectar otras partes del sistema.