

A. ¿Qué es la Programación Orientada a Objetos (POO)?

La programación orientada a objetos es la implementación de un paradigma de programación que se basa en abstraer un objeto de la vida real a un modelo de programación, en donde el modelo puede contener atributos (propiedades del objeto) y métodos (acciones que puede realizar el objeto).

B. ¿Cuál es la diferencia entre una Clase y una Interfaz?

Las interfaces son solo un molde de métodos, es decir, solo se definen, pero no se implementan. Por otro lado las clases son una plantilla para crear objetos, la cual define sus atributos, y métodos, en esta si se implementan los métodos.

C. ¿Cuál es la diferencia entre un método estático y uno no estático?

Los métodos no estáticos son métodos que dependen del contexto del objeto que los alberga, es decir, se necesita una instancia del objeto para poder ejecutar el método. Por otro lado, los métodos estáticos no necesitan de dicho contexto, por lo cual pueden ser ejecutados sin necesidad de instanciar la clase.

D. ¿Qué es un ciclo de Vida de Software?

Un ciclo de vida de software es un conjunto de fases y actividades que se llevan a cabo desde la concepción de un proyecto de software hasta su finalización y mantenimiento y sus objetivos son minimizar los riesgos del proyecto y, tener una estimación y planificación eficientes.

Los modelos de ciclo de vida del software pueden variar según la metodología que se utilice, por ejemplo en un modelo en cascada sus etapas serian análisis de requerimientos, diseño de la solución, desarrollo y evaluación, despliegue, mantenimiento y evolución.

E. ¿Cuál es la diferencia entre throw; y throw ex; dentro de un catch?

Al utilizar throw en un catch se relanza la excepción capturada, es decir se lanza la excepción original con su pila de errores completa. Por otro lado el throw ex lanza una nueva excepción, por lo cual se pierde la traza de la excepción, en otras palabras, la excepción original se pierde.