

¿Qué diferencia hay entre una clase abstracta y una interfaz?

En una clase abstracta al menos un método debe ser declarado abstracto mientras que en una interfaz todos los métodos deben ser abstractos.

Una clase abstracta puede heredar de una sola clase (abstracta o no) mientras que una interfaz puede extender varias interfaces de una misma vez.

En una clase abstracta pueden existir variables static, final o static final con cualquier modificador de acceso (*public*, *private*, *protected* o *default*). En una interfaz sólo puedes tener constantes (*public static final*).

En una clase abstracta, los métodos abstractos pueden ser *public* o *protected*. En una interfaz solamente puede haber métodos públicos.

A partir de java 8 se puede usar el método "**default**" que se puede implementar en la clase interface y que su implementación será común para todas las clases que implementen esa interface.

¿Para qué sirve el web.xml de un servlet container como Tomcat?

El archivo web.xml sirve para proporcionar información de configuración y de despliegue (mapear nuestro servlet en el contenedor) para los servlet que conforman el contenedor de Servlet (Tomcat en este caso).

¿Cuáles son las diferencias entre servicios REST y SOAP?

La diferencia principal es que REST es un patrón de arquitectura construido bajo verbos simples que se adaptan bastante fácil con HTTP y SOAP es un protocolo de mensajería basado en XML (mensajes SOAP con una estructura XML) que puede utilizarse por una variedad de protocolos de transporte, incluso el mismo HTTP.

¿Qué partes de una aplicación Web se ejecutan client-side y qué partes server-side?

Server-side, es donde se cuenta encapsulada toda la lógica de negocio (backend).

Client-side, en la interacción del las interfaces de usuario con el sever-side (frontend).

¿Qué significa IoC (Inversion of Control) y para qué sirve?

IoC es una metodología de programación que pretende evitar la dependencia directa entre clases mediante el uso de interfaces. De esta forma, en vez de hacer una llamada directa a una clase concreta proporciona una funcionalidad, se realiza una definición genérica que otorga esa funcionalidad. Este principio establece que las clases de alto nivel de un aplicación no deben depender de las clases de bajo nivel, sino que ambos deben depender de abstracciones. De esta forma, se reduce el acoplamiento entre los componentes.

¿Qué diferencia hay entre herencia y composición?

La herencia es una dependencia fuerte entre clases. La clase hija depende fuertemente de la clase padre y la necesita para funcionar. Cualquier cambio que se realice en la clase padre pudiera tener efecto en las clases hijas y se debe tener cuidado al cambiar el comportamiento de la clase padre. En la composición se tiene una clase contenedora ella va tener las instancias de las otras clases que necesite e implementara las funciones que requiera.

¿Qué diferencia hay entre una librería y un framework?

Una librería es una colección de elementos, un conjunto de rutinas, que a menudo suelen tener un propósito concreto. (funciones, clases, tipos predefinidos, constantes, variables globales, macros, etc)

Un framework es casi lo que el nombre indica: un marco de trabajo. Es un conjunto de herramientas y código para trabajar de acuerdo con una metodología, utilizando unos determinados patrones. También puede incluir otros elementos como herramientas de depuración, diseño gráfico, prototipado, edición,