



Preguntas teams is2

Ingeniería del Software II (Universidad de Las Palmas de Gran Canaria)

¿Qué tienen en común el principio abierto cerrado, el principio de sustitución de liskov, el principio de inversión de la dependencia y el principio de inversión de control? el uso de interfaces

¿Cual es el objetivo principal cuando hacemos arquitectura? crear un sw fácil de implementar y de mantener, aislando detalles críticos de la implementación

¿Que diferencia existe entre estilo de código y código limpio? Código limpio hace referencia a que sea un código fácil de mantener y entender; y el estilo de código es la forma de programar de cada usuario

¿Qué diferencia hay entre estilo de código y arquitectura? La arquitectura hace referencia al comportamiento del sw y el estilo de código atiende al código en si

¿Qué es la arquitectura del software? Es la estructura que debe tener el sw, es decir, un conjunto de patrones que nos ayuda a crear un código limpio

¿Por qué es importante un arquitecto de software en una compañía? Para la modularidad del código, es decir, para facilitar la evolución, mantenimiento, etc

¿inyección de dependencias? Mecanismo para restringir el acoplamiento entre módulos

¿Por qué creen que les estoy pidiendo hacer un modelo sin setters? Para que no se puedan modificar los atributos del objeto

Definir que es Coi Principio que afirma que se ha de elegir un diseño por composición antes que por herencia a fin de modelar la realidad del modo mas fiel y humanamente asimilable posible

¿Que otras ventajas aportar Col? Evitamos gastar memoria innecesariamente, ya que la composición implica la eliminación inmediata de los objetos de los que se compone una clase una vez destruida dicha clase

¿Justifica el ahorro de memoria que el modelo de nuestra aplicación sea incorrecto?
No (opinión)

¿Como se llaman las relaciones en las que una clase extiende a otra? generalización

¿Qué significa polimorfismo? Objetos diferentes sean tratados con la misma interfaz

¿Para qué sirve aplicar el principio de sustitución de Liskov? Para garantizar la modularidad

¿En qué consiste el principio de sustitución de Liskov? Cada clase que hereda de otra puede usarse como su padre sin necesidad de conocer las diferencias entre ellas

¿Qué tiene que ver la modularidad con la separación entre interfaz e implementación?
Si no separo la interfaz y la implementación, no tenemos un código modular, ya que los cambios que haga en una afectará a la otra, cuando no debería ser así (la interfaz no

debería cambiarse meramente por cambiar la implementación, a no ser que yo necesite cambiarla)

¿Que diferencia existe entre interfaz/implementación y encapsulamiento? La

interfaz/implementación es el código en sí, sin embargo, el encapsulamiento es una forma de programar que oculta una parte del código para dejar solo a vista del usuario tanto por seguridad como por sencillez la parte del código que necesite

¿Como definirías la encapsulación? Como el proceso por el cual ocultamos información, como por ejemplo la implementación, y mostramos solo lo necesario, haciendo el Código más legible

¿Qué significa que un mueble es modular? que tiene diferentes partes que se pueden añadir o quitar de forma independiente sin que se vea afectada la estructura general del mueble

¿Qué significa exactamente que un software sea modular? Que el sw esta dividido en subprogramas mas específicos

¿Qué es la modularidad? Dividir en componentes o módulos re combinables para construir diferentes aplicaciones de una misma familia. Subdividir un aplicación en partes mas pequeñas

¿Qué implicaciones tiene usar un tipo primitivo o un tipo de objeto? Un tipo primitivo solo almacena un dato en memoria y el tipo de objeto se usa para crear un objeto permitiendo el uso de métodos de esa clase

¿Que diferencia existe entre los atributos/métodos de objeto y clase? Los atributos/métodos de objetos se aplican para el objeto que se crea y son únicos de ese objeto. En cambio, los atributos/métodos de clase son comunes a todos los objetos que se crean de esa clase. Si no me equivoco, la palabra static define a los métodos de clase

¿Qué es una interface? Conjunto de métodos/funciones no definidos que luego serán implementados por otras clases

¿Que es una clase anónima? Se crea en el momento y no tiene nombre (variable completa). Se usan para comparadores concretos

¿Qué utilidad tienen las clases genéricas? Sirven para definir el tipo de contenidos que puede tener un mismo contenedor

¿Que son las clases genéricas en JAVA? Son clases que pueden contener diferentes clases de objetos

¿Qué es un array asociativo? Estructura de datos que almacena datos de forma <clave, valor> (en los Map)

¿Como se implementan en JAVA? Con los MAP

¿Qué beneficios ha tenido la modularidad en este proyecto? La modularidad ha ayudado, tanto a realizar un código que se entiende mejor, como a evitar reescribir código ahorrando tiempo y, por tanto, productividad. Nos ayuda a ahorrar tiempo, en caso de haber algún error

¿Como se debe usar el CamelCase en las palabras compuestas? Por ejemplo "data base" DataBase/dataBase

¿Que les sugiere la palabra Modelo? representación de lo que queremos construir, donde se dicen las características y como va a ser, es decir, es como una plantilla

¿Cual es la diferencia que existe entre los fundamentos y los principios? Mientras que los fundamentos son una parte irrenunciable que forman la base de los proyectos; y los principios pueden variar dependiendo del proyecto, es decir, tu decides cual aplicar al proyecto

¿En que se diferencian/parecen guía de estilo, arquitectura y las prácticas de código limpio?

¿Qué es la arquitectura? Es la organización que tiene el programa para que sus funciones se entiendan por todos y hace que con unas buenas bases se pueda corregir y sea facil de ver que le falta al programa y lo que hay que corregir

¿Como definirías la inyección de dependencias? Un mecanismo que nos permite restringir las dependencias que las clases utilizan para conseguir que los módulos puedan ser reutilizables en aplicaciones de la misma familia. Esto se hace evitando utilizar clases que sean de un "nivel superior" del proyecto, y que en caso de que se usen, no se construyan en la clase, sino que se le suministren en métodos o constructores

¿Que diferencia existe entre encapsulamiento y separar la interfaz de la implementación? Encapsulamiento se basa en ocultar, mientras que la interfaz se basa en separar para generalizar objetos

¿Como definirías encapsulamiento? Consiste en ocultar el estado de un objeto, es decir los valores de sus atributos de manera que solo sea posible modificarlos mediante métodos del mismo objeto

¿Que diferencia hay entre campo, parámetro y variable? Parámetro sería una variable que se le pasa a un método, una variable es un tipo de "almacén" donde guardas datos de los tipos que tu quieres y campo sería como una variable propia de una clase

¿Que diferencia existe entre los atributos/métodos de objeto y clase? os atributos y los métodos de los objetos son diferentes para cada objeto y los de clase son genéricos

¿Qué diferencia existe entre el MVC y el MVP? En el MVC la interacción es indirecta, mientras que en el MVP es directa.

Atributo : son la características (pensar en un coche)

Metodos: que es lo que hace