

-Whats new in Java 8? Explain some of them

La mayor novedad en Java 8 han sido las funciones lambda que abren la puerta a la programación funcional en Java, de manera que ahora se pueden pasar una función como argumento de un método o simplemente asignarla a una variable.

Además se pueden definir nuevas interfaces funcionales de forma que podemos añadir esa programación funcional a las clases que vayamos creando. De alguna manera, esto nos acerca más a la herencia múltiple, que en Java no está totalmente implementada.

También trae streams, optionals, métodos por defecto y estáticos para las interfaces y la nueva forma de referenciar los métodos con el delimitador "::"

-Given the following list implement a solution in order to get even numbers using Java 8 Streams

```
List<Integer> evenList =  
    list.stream()  
        .filter(i -> i % 2 == 0)  
        .collect(Collectors.toList());
```

-What do you notice when you do code review?

Al revisar código, normalmente podemos encontrar partes del código que están poco claras, que necesitan una mayor explicación mediante comentarios o que deben ser reimplementadas de forma más limpia. Además al realizar esta tarea es normal detectar procesos repetitivos que pueden ser extraídos a otros métodos.

-Have you ever worked with Scrum? Tell us what it is, what events do you remember and what roles are involved?

No he trabajado con Scrum, en mi empresa actual estamos intentando implementar Agile.

Scrum es un conjunto de buenas prácticas para trabajar en equipo, con entregables parciales cada poco tiempo. Scrum está especialmente indicado para proyectos en entornos complejos y con muchos cambios o poca definición.

Eventos:

- Sprint: es el tiempo determinado en ejecutar una o varias acciones planificadas. Suele tener una duración de entre 2 y 4 semanas. Cada sprint se pone en marcha cuando se considera terminado el anterior.

- Planificación: es la reunión en la que se define que tareas se van a incluir en el siguiente sprint.

-Reunión diaria: una reunión de no más de 15 minutos para poner en común lo que se ha hecho el día anterior, lo que se tiene previsto hacer el día actual y los problemas que han ido surgiendo.

-Revisión de sprint: valoración general del sprint terminado.

Roles:

-Scrum master: es el jefe de equipo y encargado de resolver las dudas y problemas que a su equipo le puedan surgir

-Product owner: es la persona encargada de la comunicación con el cliente

-Scrum team: el equipo que realiza las tareas. Suelen componerlo entre 5 y 9 personas.

-What access modifiers (or visibility) do you know in Java?

private, protected, public y (default)

-Differences between an abstract class and an interface. When would you use one or the other?

Una interfaz es un conjunto de métodos abstractos (desde java8 se permiten métodos por defecto y estáticos) que obliga a implementar los métodos abstractos en ella declarados a las clases que la implementen.

Una clase abstracta, en cambio es una clase (con sus atributos y sus métodos) que no se puede instanciar, normalmente se usa como una clase genérica que sirva de origen a la implementación de otras clases especializadas que hereden sus atributos y métodos (según visibilidad). La clase abstracta puede definir métodos abstractos que las clases que hereden de ella deberán implementar. Dichas clases especializadas podrán sobrecargar los métodos ya definidos en la clase abstracta para que tengan un comportamiento distinto o adicional (invocando el método `super.nombreMetodo` dentro de la implementación).

Uso las interfaces para dotar de un comportamiento a otra clase y las clases abstractas para aglutinar las partes genéricas que puedan tener un conjunto de clases que difieran entre sí parcialmente.

-What is Maven and why is used? What is Maven life cycle?

Maven es una herramienta que permite simplificar los procesos de generación de recursos y fuentes, compilación, test, empaquetado, instalación y despliegue, principalmente. Es la encargada de acceder a los artefactos con las librerías de código necesarias para la compilación (incluyendo sus versiones explícitas).

El ciclo de vida Maven es la lista predefinida de fases que se ejecutan secuencialmente en la construcción del proyecto.

-What is Git and why is it used? List all Git commands that you know

Git es un repositorio de software, compuesto de un repositorio local en el que trabajamos y un repositorio remoto al que subimos los cambios una vez que lo consideramos terminada la parte que hemos modificado. Git supone varias ventajas frente a otros repositorios remotos de software, principalmente que se puede centralizar la subida a remoto a una sola persona mediante peticiones de commit que han de ser revisadas, lo que es muy útil para proyectos con muchos desarrolladores trabajando al mismo tiempo. Adicionalmente se le puede dotar mediante herramientas que implementas metodologías de cierta capacidad para desarrollar con metodologías ágiles.

Siempre trabajo con el plugin Git para eclipse, con lo que no tengo conocimiento de Git en línea de comando. Los comandos que conozco son: status, fetch, checkout, pull, push, commit, stash, merge, reset, branch, tag, prune, rebase

-What is a Mock? What would you use for?

Los Mocks son objetos que simulan el comportamiento de objetos reales para realizar las pruebas.

Se usan para probar partes del código que usan otras que aún no están desarrolladas, o para probar situaciones que son difíciles de replicar en una prueba.

-How would you explain to someone what Spring is? What can it bring to their projects?

Spring es framework de desarrollo que se compone de multitud de herramientas para facilitar la codificación, por ejemplo a la hora de acceder a una base de datos o crear un servicio REST.

Spring ayuda a reducir los tiempos de desarrollo y en la mayoría de casos quita la complejidad de las tareas de más bajo nivel.

-What's the difference between Spring and Spring Boot?

La principal diferencia entre Spring y Spring Boot es que el segundo es más que nada una extensión del primero, ya que en realidad lo usa por debajo. Spring Boot tiene la ventaja de que es más fácil de configurar, ya que hay muchos de los componentes que se autoconfigura por defecto y la configuración que sí que sea necesaria puede hacerse de forma sencilla. Además nos permite incluir un servidor embebido para desplegar un jar ejecutable en lugar de un war que necesite de un servidor de aplicaciones.

-Do you know what CQRS is? And Event Sourcing?

-Differences between IaaS and PaaS. Do you know any of each type?

La IaaS es la infraestructura como servicio, pagando una cuota. Un ejemplo de esto es AWS.

La PaaS es la plataforma como servicio, es el tercero el que nos alquila la plataforma, un ejemplo de esto es OpenShift de RedHat.

-Explain what a Service Mesh is? Do you have an example?

-Explain what is TDD? What is triangulation?

TDD es la metodología de desarrollo dirigido por las pruebas. Primero se definen y codifican las pruebas, después se desarrolla el software que tiene que cumplirlas.

-Apply the factory pattern with lambda expressions

```
Supplier<Instant> instantSupplier = Instant::now;  
Instant now1 = instantSupplier.get();  
Instant now2 = instantSupplier.get();
```

-PaymentStrategy

```
public interface PaymentStrategy {  
  
    public static final double SERVICE_CHARGE = 5.00;  
    public static final double CREDIT_CARD_FEE = 10.00;  
  
    public double pay(double amount);  
  
    public static PaymentStrategy cashCardStrategy() {  
        return amount -> amount + SERVICE_CHARGE;  
    }  
  
    public static PaymentStrategy creditCardStrategy() {  
        return amount -> amount + SERVICE_CHARGE + CREDIT_CARD_FEE;  
    }  
  
}  
  
public class Main {  
  
    public static void main(String[] args) {  
  
        double productCost = 5.00;  
  
        PaymentStrategy cashStrategy = PaymentStrategy.cashCardStrategy();  
        double amountToPayInCash = cashStrategy.pay(productCost);  
    }  
}
```

```
PaymentStrategy cardStrategy = PaymentStrategy.creditCardStrategy();  
double amountToPayInCard = cardStrategy.pay(productCost);
```

```
}
```

```
}
```