Natalie(Vendor technical questions for 15 minutes)

1. New feature in Java 8
2. How Internal Iterator works? Vs external iterator

When the client (i.e. the programmer) controls the iteration, the iterator is called external iterator. When the iterator controls iteration, it is called an internal iterator.

1. splitIterator vs internal iterator

*Splittable Iterator* : it can split some source and it can iterate it.

*Iterator*: ‘Iterator’ is an interface which belongs to collection framework. It allows us to traverse the collection, access the data element and remove the data elements of the collection.

1. Stream api.

https://www.javatpoint.com/java-8-stream

1. the difference between intermediate and terminal operations

The distinction between this operations is that an intermediate operation is lazy while a terminal operation is not. When you invoke an intermediate operation on a stream, the operation is not executed immediately. It is executed only when a terminal operation is invoked on that stream. All intermediate operations return Stream (can be chained), while terminal operations don't.

Intermediate Operations are: filter(), map()

Terminal Operations are: forEach()

1. stream pipelining in Java 8
2. Types of gc()? How to use
3. [Serial Garbage Collector](https://javapapers.com/java/types-of-java-garbage-collectors/#serial-garbage-collector)==for single-thread environment== -XX:+UseSerialGC
4. [Parallel Garbage Collector](https://javapapers.com/java/types-of-java-garbage-collectors/#parallel-garbage-collector)==for multi-threading environment==**JVM Default**
5. [CMS Garbage Collector](https://javapapers.com/java/types-of-java-garbage-collectors/#cms-garbage-collector)==heap memory clean== XX:+USeParNewGC
6. [G1 Garbage Collector](https://javapapers.com/java/types-of-java-garbage-collectors/#g1-garbage-collector)== heap memory clean== –XX:+UseG1GC
7. Stack memory vs. heap memory

Stack==small,faster,short lived, holds references and primitive variable, “stackOverflowError”

1. Memory management

Usally JVM handels it but if you need to manage it then XMS=increase Heap size// XMN=Minimum Heap size//XSS=for stack size

1. Spring: applicationContext vs. BeanContext

The org.springframework.beans.factory.**BeanFactory** and the org.springframework.context.**ApplicationContext** interfaces acts as the IoC container. The ApplicationContext interface is built on top of the BeanFactory interface. It adds some extra functionality than BeanFactory such as simple integration with Spring's AOP, message resource handling (for I18N), event propagation, application layer specific context (e.g. WebApplicationContext) for web application. So it is better to use ApplicationContext than BeanFactory

1. Singleton & Factory

A Factory Pattern or Factory Method Pattern says that just **define an interface or abstract class for creating an object but let the subclasses decide which class to instantiate.** In other words, subclasses are responsible to create the instance of the class.

Singleton Pattern says that just**"define a class that has only one instance and provides a global point of access to it"(static method, private constructor)**

1. Synchronized

To make resources thread safe.

1. External xml bean into spring?