**JFSD: A-Z of Back-end and Database Development**

**Day 13 : 24 Aug 24**

Inner class

1. Non static inner class
2. Static inner class
3. Anonymous inner class.

**Java 8 and 11 new Features**

1. From java 8 onward interface can contains method with body. But with method we need to use default or static keyword.

The class which implements that interface. That class can override default method but can’t override static method. static method part of interface we need to call using intefaceName.methodName();

1. Functional interface : the interface which contains only one abstract method is known as functional interface. That interface can contains more than one default as well as static method but only one abstract method. to check interface is function interface or not we can use @FunctionaInterface annotation(optional).
2. Lambda expression : It is a like a arrow function in JS or angular or react js. From Java8 onward we can say Java also functional programming because of Lambda expression. Using lambda expression we can use anonymous method (functions).

We can use lambda expression, only for that interface which contains only one abstract method ie functional interface. Functional interface mainly use to achieve lambda expression. Lambda expression method return value without return keyword.

Syntax

InterfaceName referenceName = (parameterList)->expression or body

1. Stream api

Stream : flow of data.

In Collection framework like set, list, queue and map. Collection framework classes in known as in memory data structure. If we want to search any particular record present in collection we need to use pre defined method or looping concept or iterator to check the record.

In Stream on demand we can apply business logic on collection data. Stream hold the data for temporary purpose. Once the work finish on stream it automatically get destroy.

Stream api use few pre defined functional interfaces.

These four interfaces part of function package and function package is sub package of util package part of java 8.

1. Function -🡪 this interface contains one abstract method ie apply it takes T parameter and return R value.
2. Consumer -🡪this interface contains one abstract method ie accept() it take T parameter but not return type
3. Predicate 🡪this interface contains one abstract method ie test() which take T parameter and return boolean value.
4. Supplier 🡪this interface contains one abstradt method ie get() no passing parameter but return T value.

These interface we generally use in stream api

Collection or

Array with any

Value 🡪Stream 🡪Intermediate Operator 1-🡪IO2🡪IO3🡪Terminal Operator

Ie int, float,

Double ,string

User defined class

Object

Intermediate operator or method return type stream itself. Terminal operator return type is non stream ie void, float, char, Boolean, long etc.

Method reference : it is a Java8 features which is use to refer the method without that class implements functional interface. Method reference we can use on functional interface.

1. Static method reference
2. Non static method reference

Static method reference

InterfaceName referenceName = ClassName::methodName

Non static method reference

InterfaceName referenceNae = ClassReferenceName::methodName