What are new features which got introdued in java 8?

--------------------------------------------------------

.Lambda Expression

.Stream API

.Default methods in the interface

.Static methods

.Functional interface

.Optional

.Method references

.Date API

.Nashorn,JavaScript Engine

Main advantages of using java 8?

------------------------------------

Compact code(Less boiler plate code)

More readable and reusable code

More testable code

Parallel operations

What is Lambda Expression?

-------------------------------

The expression through which we can represent an anonymous function, You can call it function without name.

or

Lambda expression is an anonymous function (without name, return type and access modifier and having one lambda (->) symbol)

Anonymous: Nameless / Unknown

Anonymous function: without name, return type and access modifier

Sysntax:()->System.out.println("Lambda Expression");

EX:

Normal programming:

public void add(int a, int b) {

System.out.println(a+b);

}

Equivalent Lambda Expression:

(a,b) ->System.out.println(a+b);

What is Functional interface?

-----------------------------------

The interface who contains only one abstract method but can have multiple default and static method is called functional interface

EX: Runnable----------> run()

Callable----------> call()

Comarable---------> compareTo()

Comparator--------> compare()

How lambda expression and functional interface are related?

------------------------------------------------------------------

Functional interface is used to provide reference expressions --> this is the relation.

Comparator<String> c=(s1,s2)->s1.compareTo(s2);

(s1,s2)->s1.compareTo(s2) : this is lambda expression

Comparator<String>c : this is functional interface

Thus you can see To call lambda expression we need functional interfaces.

Can you create your own functional interface?

---------------------------------------------------

.As we know functional interface is an interface with exactly one single abstract method and can have multiple static or default methods.

.To create our own functional interface you can do follwing steps:

. create an interface

. annotate that with @FunctionalInterface

. Define exactly one abstract method

. There is no restriction on number of static and default methods defined in such and interface.

.Java can implicitly identify functional interface but still you can also annotate it with @FunctionalInterface it just give you the security that in case

if u by mistake add 2 abstract methods then Compiler will throw compile time error

What is Optional in Java 8? What is the use of Optional?Advantages of Java 8 Optional?

-------------------------------------------------------------------------------------------

Optional:

Optional is a final Class introduced as part of Java SE 8. It is defined in java.util package.

It is used to represent optional values that are either exist or not exist. It can contain either one value or zero value. If it contains a value, we can get it. Otherwise, we get nothing.

It is a bounded collection that is it contains at most one element only. It is an alternative to the “null” value.

Main Advantage of Optional is:

It is used to avoid null checks.

It is used to avoid “NullPointerException”.

Explain some of the functional interfaces which are used in Java?

-----------------------------------------------------------------------

There are many functional interfaces that are used in Java. The following are commonly used by the developers.

Function: Function is the most common functional interface out of all. It accepts only one argument and returns a specified result

Consumer: It is the same as Function interface, Consumer is also a functional interface that takes only one argument, and no result is returned by it.

Supplier: Supplier is a type of functional interface in Java that does not accept any argument and still returns the desired result.

Predicate: The type of functional interface in Java that accepts one argument and returns a boolean value is known as Predicate functional interface.

BiFunction: The Bi-Function is substantially related to a Function. Besides, it takes two arguments, whereas Function accepts one argument. Just like Function, it also returns a specified result.

Method References

---------------------

Java 8 Method reference is used to refer method of functional interface . It is compact and easy form of lambda expression. Each time when you are using lambda expression to just referring a method,

you can replace your lambda expression with method reference.

Types of Method References

----------------------------------

Reference to a static method.

.Syntax :

ContainingClass::staticMethodName

Reference to an instance method.

.Syntax :

containingObject::instanceMethodName

Reference to a constructor.

.Syntax :

ClassName::new

or

a lambda expression only calls an existing method. In those cases, it looks clear to refer to the existing method by name. The method references can do this, they are compact, easy-to-read as compared to lambda expression.

A method reference is the shorthand syntax for a lambda expression that contains just one method call.

--

Method references are a special type of lambda expressions. They're often used to create simple lambda expressions by referencing existing methods.

What is Stream API?

-------------------------

Stream API is used to process collections of objects. A stream is a sequence of objects that supports various methods which can be pipelined to produce the desired result

The features of Java stream are –

.A stream is not a data structure instead it takes input from the Collections, Arrays or I/O channels.

.Streams don’t change the original data structure, they only provide the result as per the pipelined methods.

.Each intermediate operation is lazily executed and returns a stream as a result, hence various intermediate operations can be pipelined. Terminal operations mark the end of the stream and return the result.

Important Points/Observations:

.A stream consists of source followed by zero or more intermediate methods combined together (pipelined) and a terminal method to process the objects obtained from the source as per the methods described.

.Stream is used to compute elements as per the pipelined methods without altering the original value of the object.

Default Methods In Java 8

------------------------------

Before Java 8, interfaces could have only abstract methods. The implementation of these methods has to be provided in a separate class. So, if a new method is to be added in an interface, then its implementation code has to be provided in the class implementing the same interface. To overcome this issue,

Java 8 has introduced the concept of default methods which allow the interfaces to have methods with implementation without affecting the classes that implement the interface.

What is difference between iterator and listiterator?

----------------------------------------------------------

.We can use iterator to traverse set and list collections whereas ListIterator can be used with Lists only

.Iterator can traverse in forward direction only whereas ListIterator can be used to traverse in both the directions.

.ListIterator inherits from Iterator interface and comes with extra functionalites like adding an element, replacing an element, getting index position for previous and next elements

What is the need for overriding equals() method in java?

-----------------------------------------------------------

The initial implementation of the equals method helps in checking whether two objects are the same or not but in case you want to compare the objects based on the property you will have to override this method

What is difference between fail-fast and fail-safe Iterators?

---------------------------------------------------------------

Fail-Fast

------------

.This iterator directly works on the collection object itself.

.It doesn't allow to modify the collection while iterating, will throw concurrentModificationException

.This iterator doesn't require any extra memory.

.Example of this iterator are ArryList, Hashmap,Vector, HashSet and these classes are in java.util package.

Fail-Safe

------------

.This iterator works on theclone or copy of the collection object

.It will allow to modify the collection while iterating it doesn't throw any exception.

.This one requires extra memory, consumes heap

.Example of this iterator are ConcurrentHashmap,CopyOnWriteArrayList and these classes are in java.util.concurrent package.

The difference between HashMap and HashTable?

-----------------------------------------------------

1.HashMap is non syschronized it is not thread safe and can't be shared between many threads without proper synchronization code

whereas Hashtable is synchronized. it is thread-safe and can be shared with many threads

2.HasMap allows one null key multiple null values

Whereas HashTable doesn't allow any null key or value

3.HashMap is generally preferred over HashTable if thread synchronization is not needed

When do you use ConcurrentHashMap in java?

---------------------------------------------

.ConcurrentHashmap is better suited for situation where you have multiple readers and one writer or fewer writers since map gets locked only during the write operation.

if youhave an equal number of readers and writer than ConcurrentHashmap will perform in the line of Hashtable or synchronized HashMap

.Segment level locking

What is CopyOnWriteArrayList, how it is differnt than ArrayList and vector?

------------------------------------------------------------------------

.CopyOnWriteArrayList is new implementation introduced in java 1.5 which provides better concurrent access than synchronized List.

.better concurrency is achieved by copying arraylist over each write and replace with original instead of locking

.Also CopyOnWriteArrayList doesn't throw any ConcurrentModification Exception.

.It's different than Arraylist because its thread-safe and ArrayList is not thread-safe and it's different than vector in teams of Concurrency.

.CopyOnWriteArrayList provides better concurrency by reducing contention among readers and writers.

What is the use of Static keyword in java?

-----------------------------------------------

When there is a requirement to share a method or a variable between multiple objects of a class instead of creating separate copies for each object we use static keyword to make a method or variable shared for all objects

What is difference between Comparable and Comparator?

---------------------------------------------------------------

.Comparable interface is present in java.lang package

.Comparator interface is present in java.util package

.Comparable provides compareTo() method to sort elements in java

.Comparator provides compare() method to sort elements in java

.The logic of sorting must be in the same class whose object you are going to sort

.The logic of sorting should be in a separate class to write different sorting based on differnt attributes of objects

.it provides single sorting sequences

.it provides multiple sorting sequences

.This method can sorts the data according to the natural sorting order

.This method sorts the data according to the customized sorting order

.All wraper classes and String class implement the comparable interface

.The only implemented classes of comparator are collator and RuleBasedColator