Foreach not applicable for map

Map.Entry

String constructor wont accept char value, will accept char array

Stream count returns long

Stream vars from outside cant be used on stream unless they are final array or list or map or atomicInteger.

AtomicInteger result = new AtomicInteger();

AtomicReference<String> keyResult = new AtomicReference<>();

Atomic var the value from the memory, so that changes made by other threads are visible;

Comparable interface can be used to provide single way of sorting whereas Comparator interface is used to provide different ways of sorting.

For using Comparable, Class needs to implement it whereas for using Comparator we don’t need to make any change in the class.

HashMap and Hashtable both implements Map interface

HashMap allows null key and values whereas Hashtable doesn’t allow null key and values

Hashtable is synchronized but HashMap is not synchronized.

ArrayList, HashMap, TreeMap, Hashtable classes provide random access to it’s elements.

Vector, Hashtable, are synchronized classes

If we need to sort an array of Objects, we can use Arrays.sort().

Collections.synchronizedCollection(Collection c)

The Big-O notation describes the performance of an algorithm in terms of number of elements in a data structure

ArrayList get(index i) is a constant-time operation and doesn’t depend on the number of elements in the list. So it’s performance in Big-O notation is O(1).

A linear search on array or list performance is O(n) because we need to search through entire list of elements to find the element.

<https://www.journaldev.com/1330/java-collections-interview-questions-and-answers>

<https://www.interviewcake.com/google-interview-questions>

finalize() is a special method in Object class  This method gets called by the garbage collector when the object is getting garbage collected.

One of the Java 7 features is the try-with-resources statement for automatic resource management we can create resources inside try block and use it. Java takes care of closing it as soon as try-catch block gets finished.

From Java 8 onwards, interfaces can have default method implementations. So to handle diamond problem when a common default method is present in multiple interfaces, it’s mandatory to provide implementation of the method in the class implementing them.

A marker interface is an empty interface without any method but used to force some functionality in implementing classes by Java. Some of the well known marker interfaces are Serializable

Java Reflection API provides the ability to inspect and modify the runtime behavior of java application. We can inspect a java class, interface, enum and get their methods and field details.

We can define a class inside a class and they are called nested classes. Any non-static nested class is known as inner class. Inner classes are associated with the object of the class and they can access all the variables and methods of the outer class. Since inner classes are associated with the instance, we can’t have any static variables in them.

We can have local inner class or anonymous inner class inside a class.

Java Classloader is the program that loads byte code program into memory when we want to access any class.

The continue statement skips the current iteration of a for, while or do-while loop. We can use continue statement with the label to skip the current iteration of the outermost loop

If there are other constructors defined, then compiler won’t create default constructor for us

We can use instanceof keyword to check if an object belongs to a class or not.

String accept array of chars, but no char

When we create a String using double quotes, JVM looks in the String pool to find if any other String is stored with same value. If found, it just returns the reference to that String object else it creates a new String object with given value and stores it in the String pool.

When we use new operator, JVM creates the String object but don’t store it into the String Pool. We can use intern() method to store the String object into String pool or return the reference if there is already a String with equal value present in the pool.

new String(byte[] arr) to convert byte array to String

When we use “==” operator, it checks for value of String as well as reference but in our programming, most of the time we are checking equality of String for value only. So we should use equals method to check if two Strings are equal or not.

String Pool is a pool of Strings stored in [Java heap memory](https://www.journaldev.com/4098/java-heap-space-vs-stack-memory).

When the intern method is invoked, if the pool already contains a string equal to this String object as determined by the equals(Object) method, then the string from the pool is returned. Otherwise, this String object is added to the pool and a reference to this String object is returned.

A process is a self contained execution environment and it can be seen as a program or application whereas Thread is a single task of execution within the process.

A daemon thread runs in background and doesn’t prevent JVM from terminating.

Thread new-> Runnable-> Running -> Waiting -> Blocked -> Dead

wait(), notify() and notifyAll() methods have to be called from synchronized method or block

volatile keyword with a variable, all the threads read it’s value directly from the memory and don’t cache it

Thread class setDaemon(true) can be used to create daemon thread in java. We need to call this method before calling start() method

Atomic operations are performed in a single unit of task without interference from other operations. Atomic operations are necessity in multi-threaded environment to avoid data inconsistency.

Callable interface in concurrency package that is similar to Runnable interface but it can return any Object and able to throw Exception.

Once the thread finished executing, it’s state is changed to Dead and it’s considered to be not alive.

It is used to represent optional values that is 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.

ServletConfig is used to pass configuration information to Servlet.

We can’t use RequestDispatcher to invoke servlet from another application because it’s specific for the application. If we have to forward the request to a resource in another application, we can use ServletResponse sendRedirect() method and provide complete URL of another servlet.

HttpServletResponse addCookie(Cookie c)

Usually servlet container loads a servlet on the first client request but sometimes when the servlet is heavy and takes time to loads, we might want to load it on application startup. We can use load-on-startup element with servlet configuration in web.xml

getServletContext().getServerInfo()

anything inside WEB-INF directory can’t be accessed directly in a web application, so we can place our JSP pages in WEB-INF directory to avoid direct access to JSP page from the client browser

jsp implicit objects: out Object, request Object, response Object, config Object, application Object,

session Object, pageContext Object, page Object, exception Object

JSP exception implicit object is not available in normal JSP pages and it’s used in JSP error pages only

Nested class

static class Nested\_Demo {//}

Outer.Nested\_Demo nested = new Outer.Nested\_Demo();

Inner class // OuterClass.InnerClass innerObject = outerObject.new InnerClass();

Association refers to the relationship between multiple objects. It refers to how objects are related to each other and how they are using each other's functionality. Composition and aggregation are two types of association.

Inheritance is an "is-a" relationship. Composition is a "has-a".

You do composition by having an instance of another class C as a field of your class instead of extending

Aggregation implies a relationship where the child can exist independently of the parent.

A ClassNotFoundException means the class file for a requested class is not on the classpath of the application. A NoClassDefFoundErrormeans that the class file existed at runtime, but for some reason the class could not be turned into a Class definition. A common cause is an exception being thrown in static initialization blocks.

The JIT is the JVM’s mechanism by which it can optimize code at runtime. if the code can be optimized at runtime, more often than not, the JIT will find a way.

Shallow comparison. efault implementation of equals method is defined in Java.lang.Object class which simply checks if two Object references (say x and y) refer to the same Object.

Deep Comparison. a class provides its own implementation of equals() method in order to compare the Objects of that class

hashCode() a method provided by java.lang.Object that returns an integer representation of the object memory address. By default, this method returns a random integer that is unique for each instance. Hashcode value is mostly used in hashing based collections like HashMap, HashSet, HashTable….etc. This method must be overridden in every class which overrides equals() method. If two Objects are equal, according to the the equals(Object) method, then hashCode() method must produce the same Integer on each of the two Objects.

Objects are stored on the heap

ConcurrentHashMap is thread-safe not null

Cohesion refers to what the class (or module) can do. Low cohesion would mean that the class does a great variety of actions - it is broad, unfocused on what it should do. High cohesion means that the class is focused on what it should be doing.

As for coupling, it refers to how related or dependent two classes/modules are toward each other. For low coupled classes, changing something major in one class should not affect the other. High coupling would make it difficult to change and maintain your code

ntermediate operations are those operations that return Stream itself allowing for further operations on a stream.

a map operation wraps its return value inside its ordinal type while flatMap does not.

Stream pipelining is the concept of chaining operations together

Map methods are containsKey, containsValue

List.add primero es el index luego el valor, o solo el valor

List, no se puede agregar en un index q sea mayor q el tamano actual de la lista

interrupt() para el thread

The BitSet class creates a special type of array that holds bit values. The BitSet array can increase in size as needed.

IntStream range allows to get a list of determined numbers, min inclusive max exclusive

List<Integer> list = IntStream.range(1,10).boxed().collect(Collectors.toList());