Name : Arunkumar.B.Jaiswal

CCPPID : MD0171

1. What is the difference between jre, jdk and jvm.

JVM is a acronym for JAVA VIRTUAL MACHINE. It is an abstract machine.

It is called a virtual machine because it doesn’t physically exit. It provides a runtime environment in which java bytecode can be executed.

Following are the main tasks perform by jvm:

* Loads code
* Verifies code
* Execute code
* Provides runtime environment

JRE is an acronym for java runtime environment. The JRE is a set off software tools which is used for developing java applications. It contains a set of libraries + other files that JVM uses at runtime.

SET of libraries

Eg: jar file etc

Other files

JDK stands for java development kit. The JDK is a software development environment which is used to develop java applications and applets. It contains JRE + development tools.

Following are the JDK released by oracle corporation :

* Standard edition java platform
* Enterprise edition java platform
* Micro edition java platform

SET of libraries

Eg: jar file etc

Development tool

Eg:Javac,java,

1. What is process? What is thread and life cycle of thread?

* A process is basically a program in execution. The execution of a process must progress in a sequential fashion.
* A thread, in the context of Java, is the path followed when executing a program. All Java programs have at least one thread, known as the main thread, which is created by the Java Virtual Machine (JVM) at the program’s start, when the main() method is invoked with the main thread.
* In Java, creating a thread is accomplished by implementing an interface and extending a class. Every Java thread is created and controlled by the java.lang.Thread class.

Life cycle of thread

Stop ()

JVM scheduler

Wait () / sleep ()

Notify ()

Run ()

Start ()

1. What is garbage collector? How to call it?

* Garbage collector is a background thread with the lowest priority of 1 in the scale range of 1 to 10.
* Following are the ways to make an object eligible for garbage collector

1. Nulling the reference variable.
2. Re – assigning the reference variable.
3. Isolating the reference variable.

There are two ways to request a JVM to run a garbage collector thread

1. System.gc ();

If you invoke this request will go to jvm to run gc but it is not necessary that it will run the background thread

1. Runtime rt = Runtime.getRuntime();

rt.gc();

1. What is deadlock? How deadlock occurs and how it can be prevented?

* Deadlock situation occurs when two or more processes attempt to access resource, which is locked by another process and therefore can't be shared.

Resource A and resource B are used by process X and process Y

X starts to use A.

X and Y try to start using B

Y 'wins' and gets B first

now Y needs to use A

A is locked by X, which is waiting for Y

Resource 1

Resource 2

1. There is an entity class of hibernate and its data member. If I don’t want any data member to be column of my table ,what I need to do?
2. How you will map one table with another table having one to one or one to many or many to many relationship in hibernate?

You need to annotate your entity with JPA’s *@*Table and *@*SecondaryTable annotations and provide the names of the first and second table as the value of the name parameters.

You need to annotate each attribute which you want to map to the secondary table with a *@*Column annotation and set the name of the secondary table as the value of the table attribute.

1. What is dependency injection?

Dependency injection is basically providing the objects that an object needs instead of having it construct them itself. It's a very useful technique for testing, since it allows dependencies to be mocked or stubbed out.

Dependencies can be injected into objects by many means (such as constructor injection or setter injection). One can even use specialized dependency injection frameworks e.g Spring to do that, but they certainly aren't required. You don't need those frameworks to have dependency injection. Instantiating and passing objects explicitly is just as good an injection as injection by framework.

1. How @Autowired works?

First, and most important - all Spring beans are managed - they "live" inside a container, called "application context".

Second, each application has an entry point to that context. Web applications have a Servlet, JSF uses a el-resolver, etc. Also, there is a place where the application context is bootstrapped and all beans - autowired. In web applications this can be a startup listener.

Autowiring happens by placing an instance of one bean into the desired field in an instance of another bean. Both classes should be beans, i.e. they should be defined to live in the application context.

This means that the context instantiates the objects, we never make new UserServiceImpl() - the container finds each injection point and sets an instance there.

1. How you will iterate over map?

By using Foreach and Map.Entry

Map<String , String> map = . . .

For (Map.Entry<String,String> entry : map.entryset())

{

System.out.println(entry.getKey() + “/” + entry.getValue());

}