PAGE No.
DATE / /

JDK Assignment [shreyash shedge]

1) Explain the components of JDK?

⇒ JDK stands for Java Development kit.

$$\text{JDK} = (\text{Java Dev. Tools}) + (\text{Java API Docs}) + \underbrace{(\text{rt.Jar}) + (\text{JVM})}_{(\text{JRE})}$$

a) Java Dev. Tools :- Software programs or utilities designed to help developers in coding and maintaing applications. e.g. IDEs, Text editors.

b) Java API docs :- This provides overall documentation of the java platform, which has classes, interfaces, methods and packages.

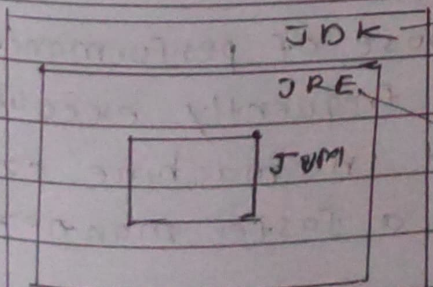
c) rt Jar :- It is called runtime JAR and contains all the classes.

d) JVM :- It is a Java virtual machine that executes java bytecode [compile & execute]

---

2) Differentiate JDK, JVM and JRE?

⇒



→ Includes tools for developing Java application. [compiler & libraries]

→ Provides runtime environment for java application.

3) What is JVM? How does it execute code?

⇒ JVM is [redacted] responsible for converting bytecode to machine-specific code and also performs memory management and security.

and,

Three main components are :-

a) class Loader subsystem.

b) Runtime Data areas → method areas, PC registers.

c) Execution engine → Interpreter, compiler and garbage collection area.

4) Memory management in ~~Java~~ JVM?

→ a) When we run a Java program, the JVM creates a space in memory called the "heap" to store all the objects that our program uses.

b) As our program runs, it creates more and more objects, and the JVM needs to make sure there's enough space in the heap for all of them. For this JVM uses "garbage collector" for reusing the space that's how memory management works.

5) JIT compiler - Just in time

⇒ It is used for the purpose of performance improvement, JIT compiler takes frequently executed bytecode and compiles it in machine code, which is done ~~Father~~ in a faster manner. and more efficiently.

and,

Bytecode

It is a low-level code that is generated by Java compiler after compiling. It is then executed by JVM

It is important because it is same on every machine. which helps Java to run same program on different machines.

6) Ans:-

ⓐ Class Loader :- Loads Java classes into the JVM. it loads the bytecode of a class from the disk into memory and creates class object. (metadata is inside).

ⓑ Method area :- It is a shared memory area that stores the bytecode of all loaded classes. It also stores the runtime constant pool, which is a ~~pe~~ constant values.

ⓒ Heap & stack :- Heap is area that stores all objects created during execution and stack stores the state of each thread (local variables).

ⓓ Execution engine :- Execution of Bytecode. (JIT) is used here.

7) Ans:- Platform independance:-

It uses platform-independant bytecode that is executed ~~[illegible]~~ by JVM as it provides consistent runtime environment which helps us to run on any platform without modification.

8) Ans:-

ⓐ Class loader:- Loading and linking of Java classes during runtime, It loads the bytecode of a class from the disk into memory, and it creates a class object (which has metadata).

ⓑ Garbage collection:-
It is an automatic process of freeing up memory by removing objects that are not needed anymore. JVM is designed to manage memory efficiently on its own.