Assignment-3 [Interview Question]

DATE
PAGE

1) What is JDK & explain the components of JDK?

Ans- • It stands for Java development tool kit
• Software development environment which is used to develop java applications & applets.
• It contains private java machine & few other resources such as interpreter/loader, compiler, jar to complete the development.

Components of JDK

appletviewer: used to run & debug java applets without a web browser.

apt: it is an annotation-processing tool

extcheck: it is utility that detects JAR file conflicts

java: the ~~load~~ of loader of java appl" this tool as an interpreter.

javac: it specifies the java compiles which convert source to into byte code

jar: specifies the archiver, which relate class libraries into single jar file.

javafx package: tool to package & sign javafx appl"

javah: the C header & stub generator.

2) Diff" b/w JDK, JRE & JVM

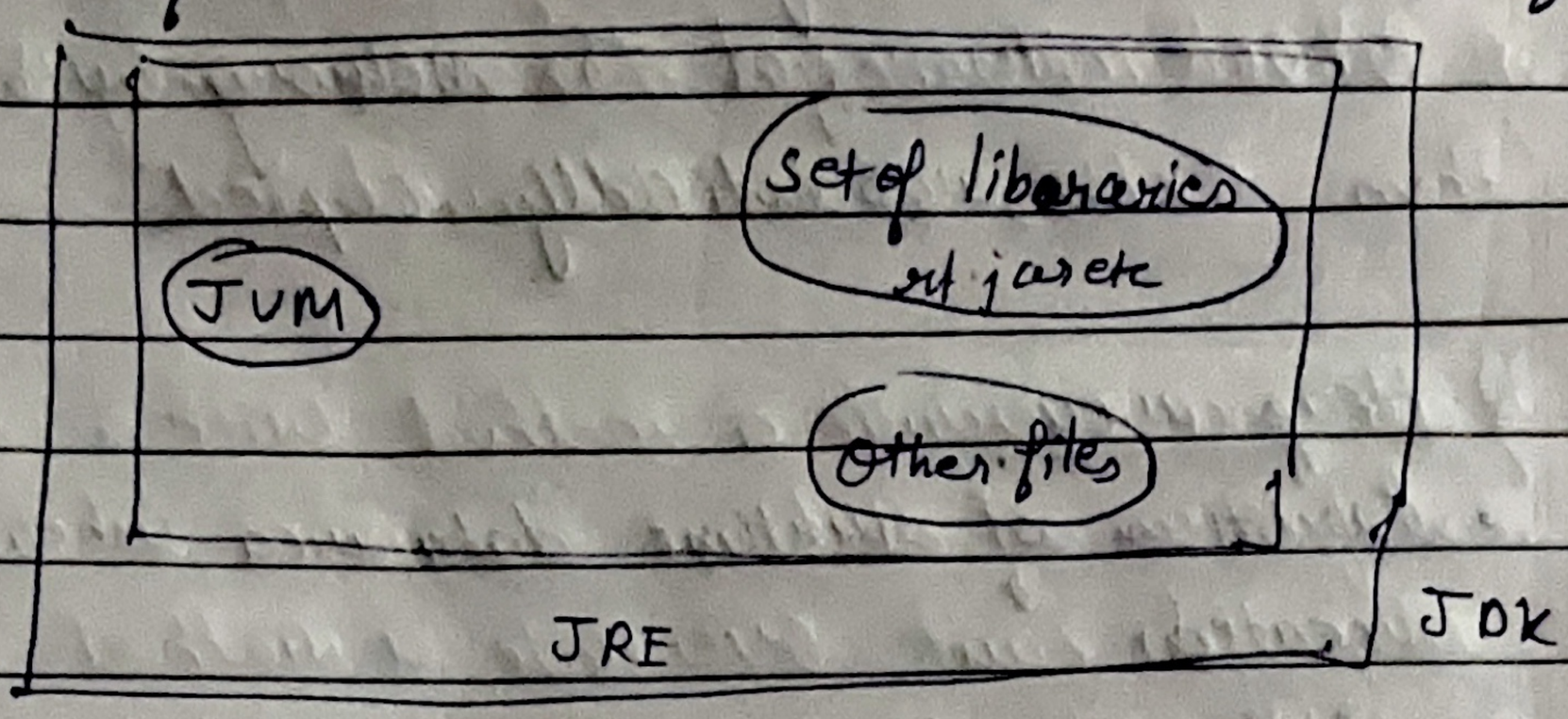
Ans- JVM: • Java virtual machine is abstract machine, doesn't physically exist. &
• It is a specification that provides runtime environment in which java bytecode can be excuted.
• also run other language program & compiled to java

It perform following task:

- Loads code
- Verifies code
- Execute code
- Provide runtime environment

JRE : Java runtime environment

- Set of software tool used for developing java appl.
- Used to provide runtime environment
- Implement of JVM, contains libararies to other files.



3) JVM Arch. with internals

Intro-

- Stands for Java virtual machine
- Used to excute java bytecode & genrate result.
- Software Internally like machine is known as virtual machine.

It internally divide into following

a) Class loader subsystem : It load the class file onto runtime data area using loaders.

b) Runtime Data Area : It is internally divide into following.

- Method area
- Heap
- Javastack
- PC register
- Native method

c) Execution engine : It start execution (with main()-method available from Javastack area.

Uses two translators

i) Interpreter ii) JIT- compiler

4) Explain memory management system of JVM.

Ans-

a) Method area

- Partition of runtime data area
- While class loading static components will get the memory with in the class.
- once main() got the memory with in class then it is automatically copied.

b) Heap area

- Partition of runtime data area where objects are created.
- While object creation instance member will get the memory within object.

c) Java Stack area

- Partition of runtime data area where methods are excuted.
- main() is the first method copied into javastack area & main() method will call remaining method.

d) PC register Area:

- Program Counter: Register will head the status of method execution in java stack area.
- Every method is loaded to javastack area will be opened its own PC register.

e) Native Method area

- Methods from library which are declared with native keyword.
-

5) Where are JIT Compiles & its role, What is bytecode, Why its imp?

Ans:- • JIT compiles is a component of virtual machine that compiles bytecode into machine code at run time to improve performance of java appl."
• It's a dynamic compilation that combines the speed of compile code with flexibility.

Bytecode :- It is a instruct." set of JVM.
• As soon as java program is compiled java bytecode is generated
• It is a machine code (.class)

Importance :- Provide java, it's one of more feature Platform independent.

6) How does java achieve platform independence.

Ans: • It has a feature WORA (write once run everywhere) describe platform independence
• First java source code () compiled by JVM & generated compile code or byte code.
• This bytecode can run on all OS.

7) What is the significance of class loader in java? What is the process of garbage collect." in java?

Ans: Class loader: It is responsible for dynamically loading classes into the JVM during runtime.

Garbage Collection: In java it is process of automatically freeing memory by reclaiming objects that are no longer in use.

8) What are the 4 access modifiers & differ from each other?

Ans: a. Public: Member is accessible from any other class.

b) Protected: Member is accessible within its own package & by subclass

c) default (package-private): Member is accessible only within its own package.

d) Private: Member is accessible only within the class it is declared in.

10) Can you override a method with a diff access modifier in a subclass? explain

Ans: No, you can't override a method with a more restrictive access modifier in a subclass. For instance, if a method is protected in a superclass, it can't be overridden with a private modifier in a subclass bcoz it would violate the liskov substit.ⁿ method, which states that a subclass should be substituable for its superclass.

11) What is diff" b/w protected & default (package-private) access?

Ans: Protected: Member is accessible within the same package & by subclasses in diff. packages.
default (package-private): Member is accessible only within its own package & not by subclasses outside the package.

12) It is possible to make a class private in Java? If yes, where can it be done, & what are the limitation?

Ans: A class can be made private only if it is a nested class within another class. Limitation is that the private class is only accessible within the enclosing class.

14) Can a top-level class in java be declared as protect or private? Why or why not?

Ans: No, because top level classes can only have public or default access. This happens because protected & private access control is meant to restrict access to members within a class hierarchy or within a package.

15) What happens if you declare a variable or method as private in a class & try to access it from another class within the same package?

Ans: It can't be accessed from another class, even if the other class is in the same package. Private members are only accessible within the class they are declared in.

16) Explain the concept of "package-private" or "default" access. How does it affect the visibility of class members?

Ans: It means that class members are accessible only with the same package. This package access level doesnot use any keyword; if no access modifier is specified.

This affects the visibility by restricting access to class members to other classes within the same package, while preventing access from classes outside the package.