Assignment -03.

04 Sep

Date

P. No

(Shivani Gore)

1) Explain components of JDK.

Ans (1) It is software development environment used for developing java applications.

(2) Full form of 'JDK' is 'Jave Development kit'. It includes 'Java Runtime Environment', a documentation generato compiler, and other needed tools.

(3) Most popular JDK are Orade JDK, IBM J9 JDK, Amazon

* components

(1) JVM :- It runs java application, translate bytecode into java code that runs on computer's hardware.

(2) Compiler :- converts java source code into bytecode.

(3) JRE :- It includes JVM & standard java libories needed to run java application

(4) Documentation :- Generates documentation from code comments. It help developer to understand.

2) Differenciate bet^n JDK, JVM, JRE

Ans

① JDK :-

- Software development kit for java developers.
- It includes JVM, compiler, Development tools like javac, Javadoc

② JVM :-

- It is a Virtual machine that runs java bytecode.
- It includes execution engine, memory management
- It includes bytecode intrepretor to execute bytecode instruction.
- If you are running a java program, then you are using JVM

③ JRE :-

- It provides libaries, JVM and other components necessary to run application.
- It includes JVM
- It includes core libaries

③ What is role of JVM ? . How does JVM excute java code

Ans • Role of JVM.

① JVM allows java programs to be run on any device or operating system
② It executes bytecode.
③ The JVM handles memory allocation and deallocation. It includes garbage collector.
④ JVM provide security.
⑤ The JVM uses various techiniques to optimize performance i.e JIT

* Execution :-

① compilation :- Java code (.Java) is compled by 'javac' into .class files.
② The JVM loads .class files into memory.
③ Before execution, the JVM verifies bytecode.
④ JVM can compile bytecode into native machine code at runtime for execution.

4) Explain memory management of JVM

Ans JVM divides memory into -

1) Heap :- The heap is where objects are alloted.
2) Stack :- Each stack frame corresponds to a method call and contain local variables.
Each thread in java has it's own stack which stores local variable
3) Method area :- This are stores class data including data structures, method data
4) PC register :- Each thread has a PC register that keeps crack of the current instruction being executed
5) Native method stacks :-
It stores the state of native method calls

* Garbage Collection :-
JVM garbage collector is the process by which JVM automatically reclaims memory that is no longer in use.

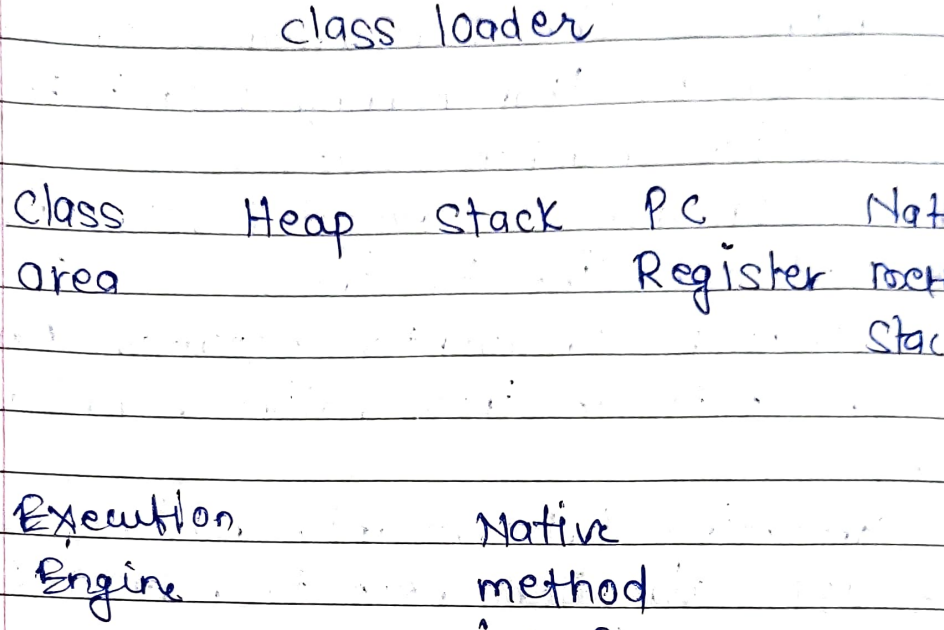
→ memory management :-
- When object is created, memory is allocated from heap
- When object is no longer referenced it becomes eligible for garbage coll.

5) What are JIT compiler and it's role in the JVM. What is bytecodes & why it is important for java

Ans.) 1) jit compiler are component of JVM
2) It improves the performance of java application by transferring java bytecode into native machine code.
3) When a program is executed, JVM interpr the bytecode line by line.
4) JIT compiler frequently executed part of the bytecode into native machine code.
5) By converting bytecode into native code the JIT compiler help to reduce the overhead & improve performance

* Bytecode :- It is a set of instruction of JVM that is generated from java sourc code by the compiler.
- It is low level, platform independent
- Bytecode provide security
- JVM use bytecode.

6) Describe the architecture of JVM



1) classloader :- It is a subsystem of JVM which is used to load files.
Types :-
1) Bootstrap classloader :- It loads rt.jar file which contain java.lang java.net, java.util package
2) Extension classloader :- child classloader of bootstrap & parent classloader system classloader

2) Heap :- Runtime data area in which Obj are allocated

3) class :- It stores per class structure
4) Stack :- Stores into frames. holds local variable
5) Pc Register :- Contains the address of JVM
6) Native method Stack :- Contains all native methods used in the application
7) Native Interface :- It is framework which provides an interface to communicate with another application.

7) How does java achieve platform independence through jvm

Ans 1) Java source code is written in .java & compiled by javac called as bytecode which stored in .class.
2) The JVM is an abstraction layer betn java bytecode & OS & hardware
3) JVM often includes a JIT compiler that translates bytecode into native code.
4) JVM ~~en~~ have the mechanism that ensures bytecode is secure.

8) What is significance of class loader. Pro. of garbage collection.

Ans Significance :-

1) Loading classes :-

1) Dynamic loading :- Dynamically loads cla. from various sources into JVM

2) classpath :- It define the path where JVM should look for class file

2) class verification :- Before class executed classloader verifies in bytecode to ensur. safety

3) Namespace management :- It creates separate namespace for different classes which prevent naming conflict.

* Garbage collection :-
It identifies Object that are no longer reachable from any live threads helping to reclaim unused memory.

9) What are 4 access modifiers in java and how they differ from each other

Ans

1) Public :- The member is accessible from any other class in any package use public when you want to expose member.

2) Protected :- The member is accesible within its own package and subclass

3) default :-
The member is accessible only within its own package

4) Private :-
The member is accessible only within it's own class

10) Difference between public protected and default access modifiers.

Ans 1) public :- The member is accessible from any other class in any package

e.g. public class test {
    public int value;
}

2) Protected :- The member is accessible with it's own package & by subclass

e.g.
public class test {
    protected int value;
}

3) Defaut :- The member is accessible only within it's own package

class test {
    int value;
}

11) can you overide a method with a different access modifier in a subclass ?

Ans No, we cannot overide a method with a different access modifier in a subclass

When overriding a method, the access modifier of overridden method in subclass must be same as or more accessible that access modifier

12) Is it possible to make a class private in java

Ans In java we cannot declare a class as private. we can declare inner class as private

* limitation :-

1) Top level - classes can be declared as public or default.
2) Java does not support declaring.

13) What happens if you declare a variable or method as private in a class and try to access it from another class within same package

Ans In java, if you declare a variable or method as private in a class, it is only accessible within that class itself. This means that even if you try to access it from another class within same package, there will be an restriction

14) Explain the concept of 'package-private' or 'default' access. How does it affect the visibility of class member.

Ans :- 'Package-private' access is the access level applied to class member methods when no explicit modifier is specified

- When a top level class is declared without any class access modifier, it is package private

15) Difference bet" protected & default

Ans Both control the visibility of class members.

① Protected - members maked as protected are accessible to all other classes within same package
protected members are also accessible in subclass

- protected access is useful for enabling inheritance

→ Default :-

- members with default access are accessible only with same package
- It does not allow access from Subclass
- does not allow access from subclass