Assignment - 2

Page No.:
Date:

Q1. Explain the components of JDK ?

→ JDK stands for Java Developement Kit, It internally contains JRE and JVM. where JRE stands for Java Runtime Environment and JVM stands for Java Virtual machine.

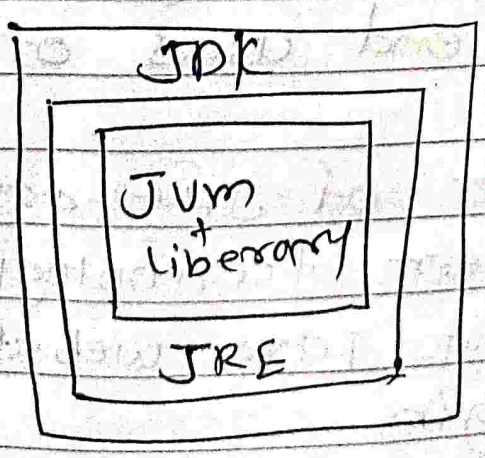
components

JRE-

Java Runetime environment provide the enviornment to execute the Java program. It internally contains JVM which is responsible for executing java program.

JVM

Java virtual machine converts Java code into bytecode and execute the java program



JRE = JVM + library
JDK = JRE + JVM + library.

Page No.:
Date:

Q2. Differentiate between JDK, JVM and JRE

JDK.

The java development kit (JD is a software development environment which is used to develop java application. It contains JRE and JVM.

It also contains development tools to provide an environment to develop java programs.

JRE

Java Runtime environment is an installation package that provides environment to ~~develop~~ run your java programs on machine. JRE is only used by those who only want to run java programs that are end users of your system.

Components of JRE are
- Development technologies, including deployment, Java webstart and Java plugin.

- Java interface toolkits
- integration libraries, other base libraries
- JVM

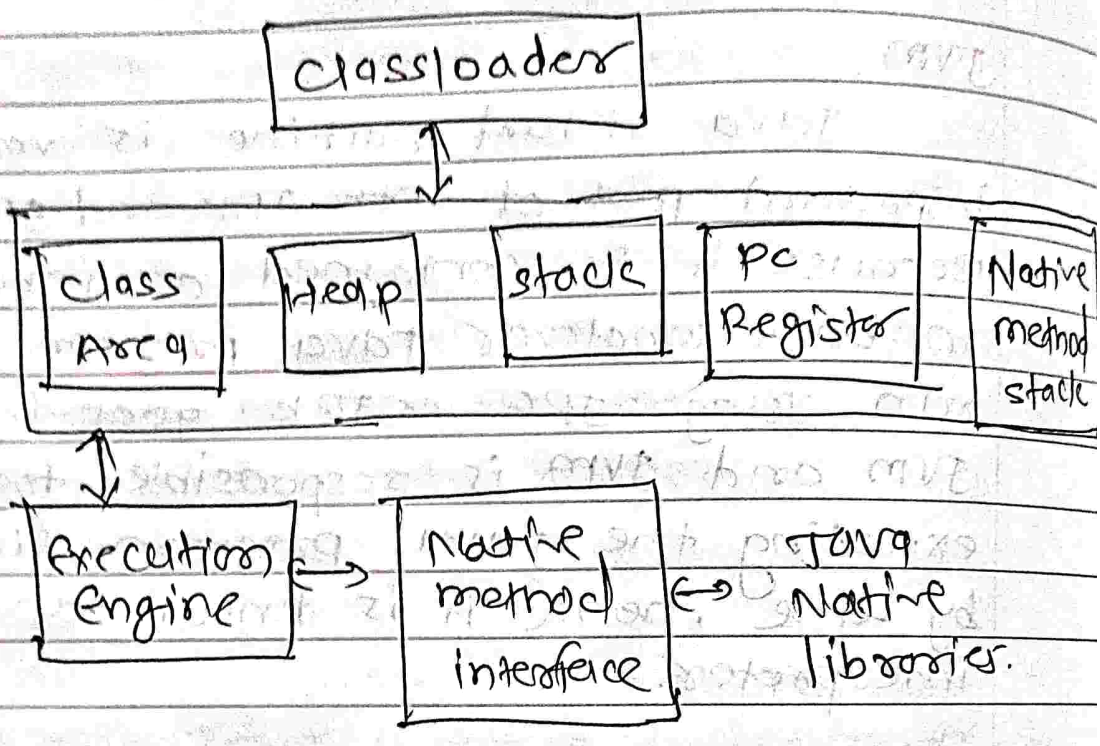
JVM.

Java virtual machine is very important part of both JDK and JRE because it is contained or inbuilt in both. Whatever Java program you run using JRE or JRK goes into JVM and JVM is responsible for executing the Java program line by line, hence it is known as interpreter.

Q2 Architecture of JVM

Q what is the role of JVM in Java? and how does JVM execute Java code

following is architecture of JVM.



Classloader.

There are three class loaders

1. Bootstrap Classloader
2. Extension classloader
3. system/application classloader

The classloader loads class files in class area.

Class Area

class area stores per-structures

such as the runtime constant pool, field and method data, the code for methods.

Head

It is the runtime data area in which objed are allocated.

stack

stack stores frames. It holds local variables and partial results and plays a part in method invocation and return.

Each thread has a private JVM stack, created at the same time as thread.

PC register

PC (program Counter) register contains the address of the Java virtual machine instruction currently being executed.

Native method stack.

It contains all the native methods used in the application

Date:

Execution Engine

It contains 1) A virtual processor
2) Interpreter
3) Just in time compiler

This engine Read bytecode and then execute instruction

JIT Compiler ~~is~~ Compiles part of bytecode that have similar functionality at the same time and hence reduces the amount of time needed for compilation.

Java ~~Na~~ Native Interface

It is a framework which provides an interface. ~~the~~ to communi

Page No.:
Date:

Q4 Explain the memory management system of JVM.

following are parts of memory in JVM

1. method / class area
2. Heap area.
3. stack
4. PC register
5. Native method area.

① method area

class elements like constant pool, class fields, constructor codes, method codes and class level information is stored in method area.

② Heap memory

It allows memory to objects and class during execution, whenever object is created it is stored in heap memory.

③ stack memory

It is based on LIFO. It is static memory allocation, whenever a java method is called a new

block is called, a new block is created in java.
stack memory to hold local or intermediate variables and references to other objects in the method.

④ PC register.

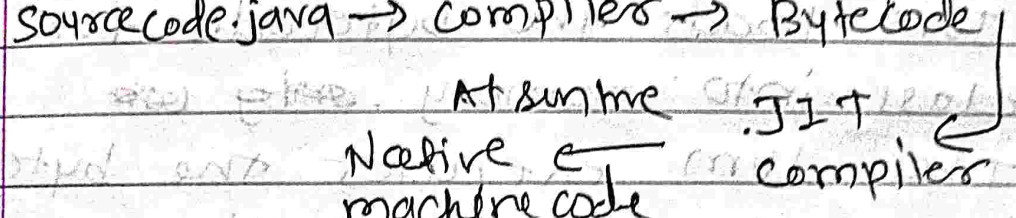
The main function of PC registers is to store the address of currently executing the instruction it also stores the address of threads responsible for executing current instruction.

⑤ Native area

This area is implemented using languages other than java. With the creation of new threads, memory is allocated in this ~~method~~ area for each created thread. The size of the native area can be fixed or dynamic.

Q-5 What are the JIT compiler and its role in the JVM? What is the bytecode and why is it important for java?

JIT is just in time compiler is essential part of JRE that is responsible for the performance optimization of application during runtime.



Bytecode

Bytecode is intermediate representation of java code & executed by JVM.

Bytecode important for security. the JVM can verify the bytecode before executing it. which helps to prevent malicious code from being executed.

Page No.:
Date:

Q6 ~~Diff~~ what is role of JVM in Java? How does the JVM executes Java code.

JVM

Java virtual machine converts Java code into Bytecode. It is also known as interpreter

execution of Java code,

- JVM loads the bytecode for a class into memory.
- The JVM executes the bytecode for a class one instruction at a time this is known as interpretation
- JVM also compiles the bytecode for a class into machine code.
- After that it manages memory used by java program. & reclaims memory that is no longer being used by the program i.e. garbage collection.

Page No.:
Date:

Q7. How does Java achieve platform independence through the JVM?

Java works on WORA (write once run anywhere) principle. This means that java code can be written once and run on any supported platform because JVM interprets the java bytecode ~~and translates it into machine code~~.

JVM is made for all platforms. So we can execute the bytecode on any platform. This makes Java independent.

Page No.:
Date:

Q8. What is significance of the class loader in java? What is the process of garbage collection in java?

→ Classloader

The class loader is part of the java virtual machine that loads classes into memory. The class loader is responsible to load classes in memory.

garbage collection:

Java collection is an automatic process. The process is looking at heap memory, identifying which objects are in use and which are not and deleting the unused objects. An in use object or a referred object means that some part of program still maintains a pointer to that object. An unused or unreferenced object is no longer referenced by any part of your program.

Page No.:
Date:

So the memory used by an unreferenced object can be reclaimed. The programmer does not need to mark objects to be deleted explicitly.
~~Free~~.