

JAVA FULL STACK DAY - 2

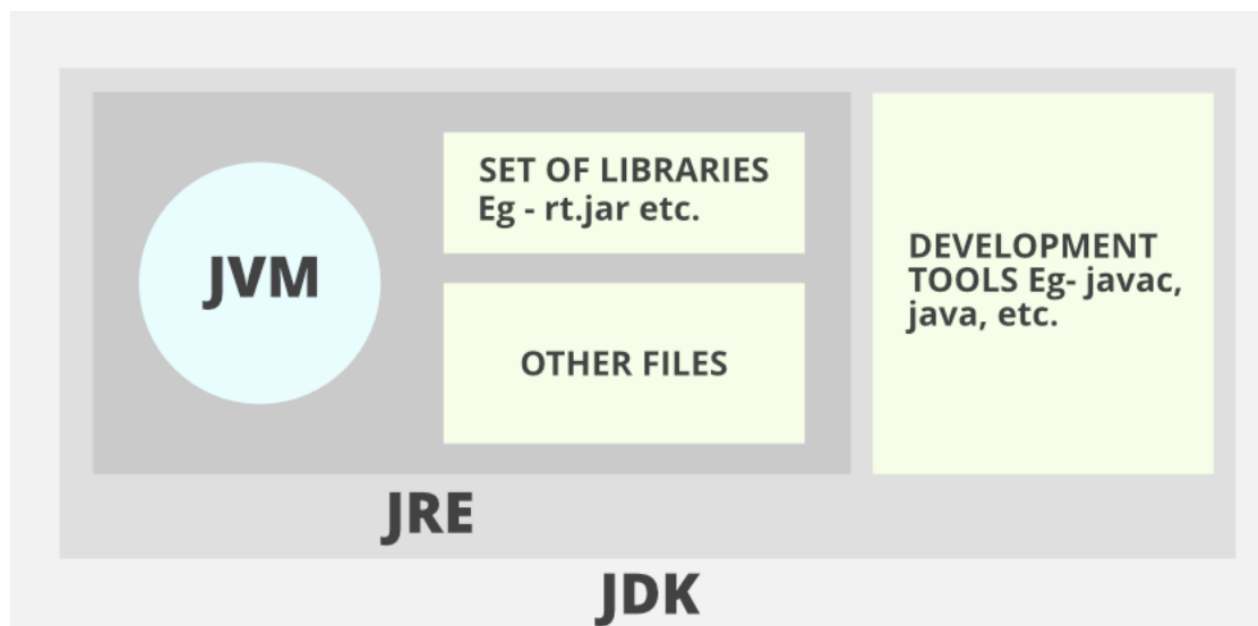
IS JAVA COMPILED OR INTERPRETED

- Java is both compiled or interpreted Language.
- Firstly, Java source code is compiled into byte code.
- And then it is interpreted by JVM for execution.
- So, Java is both Compiled and Interpreted Language.

What is JDK, JRE, JVM?

Java Development Kit (JDK)

- JDK includes JRE and development tools.



JAVA FULL STACK DAY - 2

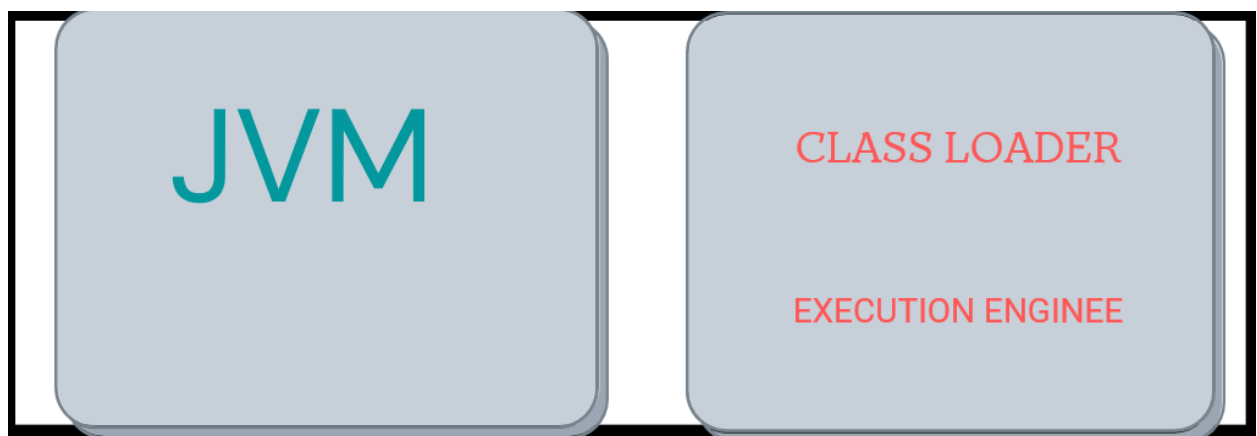
Java Runtime Environment (JRE)

- It is the implementation of JVM.
- It includes java runtime libraries and .class files.



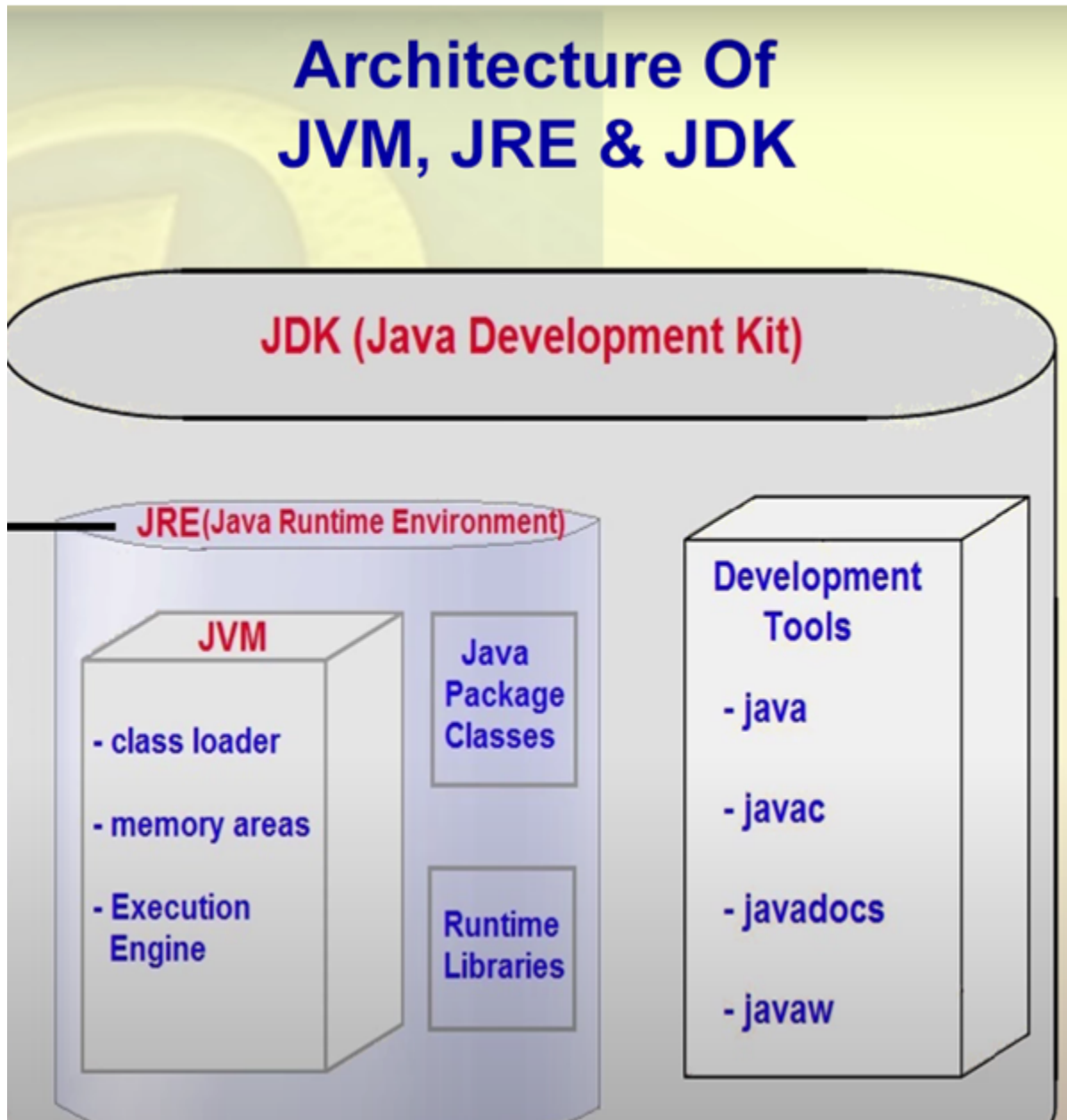
Java Virtual Machine (JVM)

- It is known as Virtual Machine because it doesn't physically exist.
- It will execute the java code byte by code.



JAVA FULL STACK DAY - 2

Summary



JAVA FULL STACK DAY - 2

JVM ARCHITECTURE

1. Class Loader
2. Memory Area
3. Engine Execution
4. Native Method Interface
5. Native Method Library

1. Class Loader

- Class Loader is a subsystem of JVM which is used to load class files.
- Whenever we run java the java program, it is loaded first by the class loader.

2. Memory Area

- It consists of Method Area, Heap Area, Stack Area, PC Registers, Native Method.
- In method area .class files and static variables will be loaded.
- In Heap Area Instance Variables/Objects will be loaded.
- In Stack Area Local variables & Reference variables will be stored.

JAVA FULL STACK DAY - 2

- Program Counter (PC) Registers contains the address of JVM instruction currently being executed.

3. Engine Execution

- It consists of Interpreter, JIT Compiler, Garbage Collector.

Interpreter

- It interpretes the bytecode line by line and then executes.

Just In Time (JIT) Compiler

- It increases the efficiency of Interpreter.
- It compiler the entire bytecode and changes it to native code. So, when a repeated method occurs JIT compiler will automatically provides native code without interpreting again.

Garbage Collector

- It automatically removes the memory for unused variables, or objects which was no long needed.

JAVA FULL STACK DAY - 2

JVM Architecture

