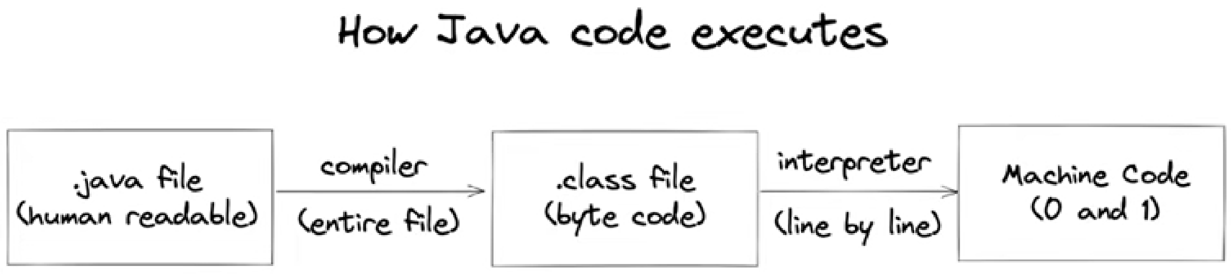
Java and DSA

I .Java architecture

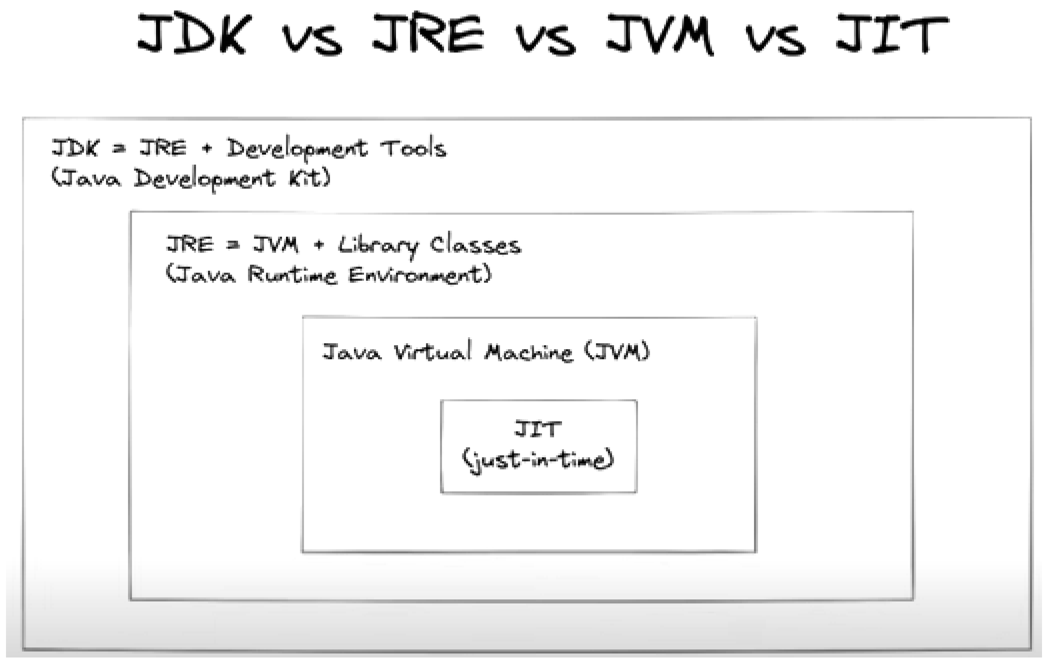
1.How Java code executes:



- Reason why Java is platform independent
The source code will not directly run on a system we need JVM to run this.

More about platform independence
- It means that byte code can run on all operating systems.
- We need to convert source code to machine code so computer can understand
- Compiler helps in doing this by turning it into executable code
- this executable code is a set of instructions for the computer
-After compiling C/C++ code we get .exe file which is platform dependent
-**In Java we get bytecode, JVM converts this to machine code**
- Java is platform-independent but JVM is platform dependent

2.JDK V/S JRE V/S JVM V/S JLT


JDK vs JRE vs JVM vs JIT
JDK = JRE + Development Tools
(Java Development Kit)
JRE = JVM + Library Classes
(Java Runtime Environment)
Java Virtual Machine (JVM)
JIT
(just-in-time)

2.1 JDK

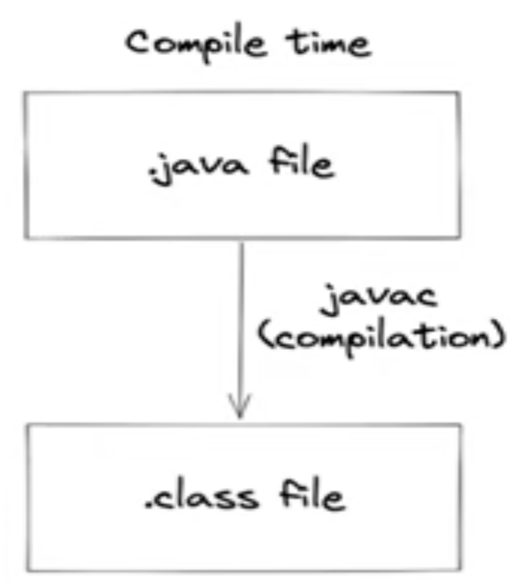
- Provides environment to develop and run the Java program
- It is a package that includes:
1. development tools to provide an environment to develop your program
2. JRE to execute your program
3. a compiler - javac
4. archiver jar
5. docs generator - javadoc
6. interpreter/loader

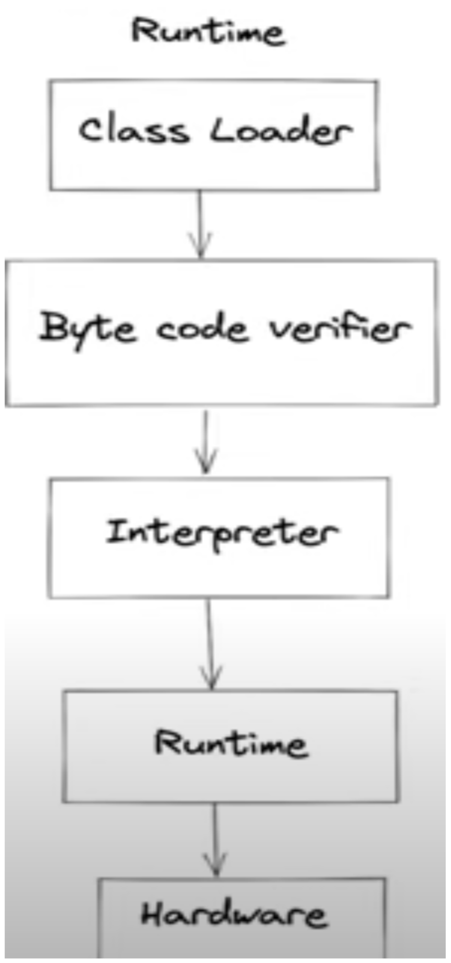
2.2 JRE

-It is an installation package that provides environment to only run the program
- It consists of:
1. Deployment technologies
2. User interface toolkits
3. Integration libraries

4. Base libraries
5. JVM
- After we get the .class file, the next things happen at runtime:
1. Class loader loads all classes needed to execute the program.
2. JVM sends code to Bytecode verifier to check the format of code

2.3 Compile time V/S Runtime





2.4 How JVM works:

1.Class Loader: reads class file and generate binary data
   - an object of this class is created in heap
2. Linking
   - JVM verifies the class file

- allocates memory for class variables & default values
- replace symbolic references from the type with direct references
- Initialization: all static variables are assigned with their values defined in the code and static block
-JVM contains the Stack and Heap memory allocations.

3.JVM Execution

- **Interpreter:Line by line execution when one method is called many times,it will interpret again and again**
- **JIT: those methods that are repeated,JIT provides direct machine code so re-interpretation is not required.**
    - **makes execution faster**