Introduction to Java - Architecture & Installation

* How Java code executes ?

- Machine only understands 0's & 1's
- Human Readable Language.
- Java Programming files. .java extensions

$$\begin{tikzcd}
\boxed{\begin{array}{l}\text{.java file} \\ \text{(human} \\ \text{readable)}\end{array}} \arrow[r, "\text{compiler}", "\text{(entire file)}"'] & \boxed{\begin{array}{l}\text{.class file} \\ \text{(byte} \\ \text{code)}\end{array}} \arrow[r, "\text{interpreter}", "\text{(line by line)}"'] & \boxed{\begin{array}{l}\text{Machine} \\ \text{Code} \\ \text{(0 \& 1)}\end{array}}
\end{tikzcd}$$

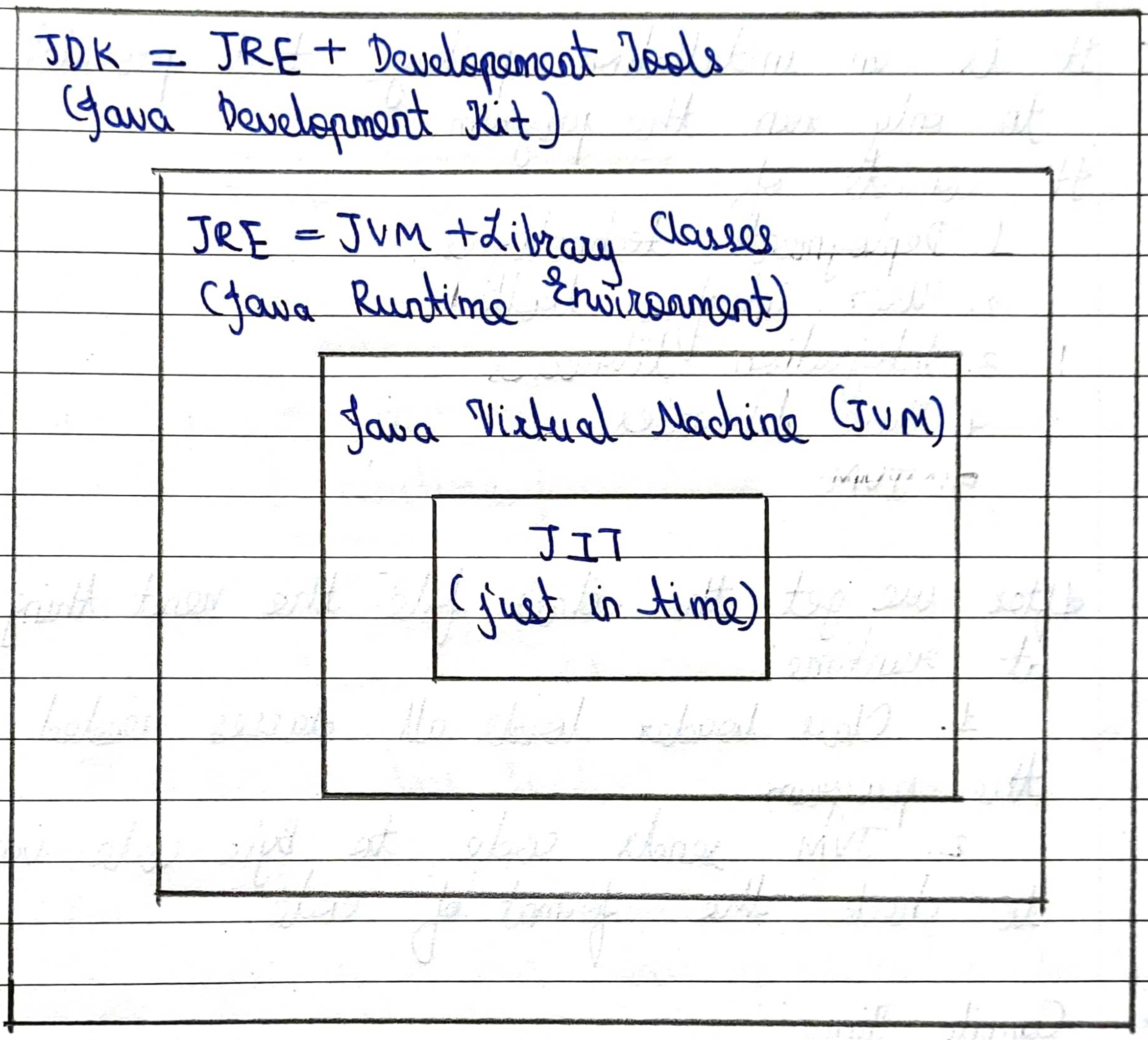
this is the
source code

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

* More about platform independence-

- It means that byte code can run on all OS's
- We need to convert source code to machine code cso computer can understand.
- This executable code is 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

* 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)

* JDK

- Provides environment to develop & run 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. archiever - jar
  5. docs generator - javadoc
  6. interpreter / loader.

* JRE

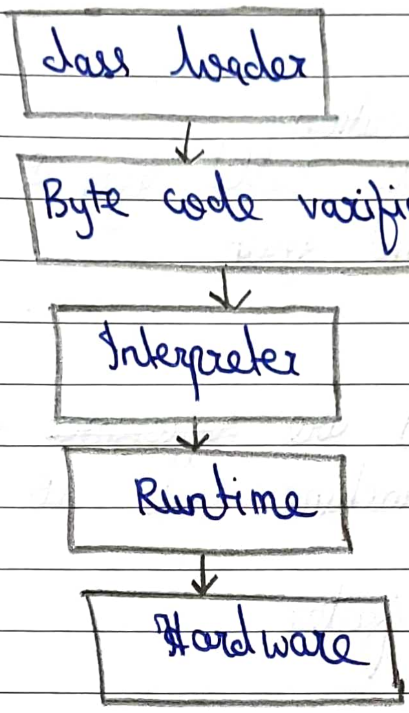
- It is an installation package that provides environment to only run the program.
- It consists of:
  1. Deployment technologies
  2. User interface toolkits
  3. Integration libraries
  4. Base libraries
  5. JVM

- After we get the .class file the next thing happen at runtime:
  1. Class loader loads all classes needed to execute the program.
  2. JVM sends code to Byte code verifier to check the format of code.

* Compile Time

$$\begin{tikzcd}
\boxed{\text{.javafile}} \arrow[d, "\text{javac (compilation)}"] \\
\boxed{\text{.class file}}
\end{tikzcd}$$

* Runtime



* (How JVM works) class loader:

- Loading
  - reads .class file and generate binary data
  - an object of this class is created in the heap

- Linking
  - JVM varifies the .class file
  - allocating memory for class variables & default values
  - replaces symbolic references from the type with direct references.

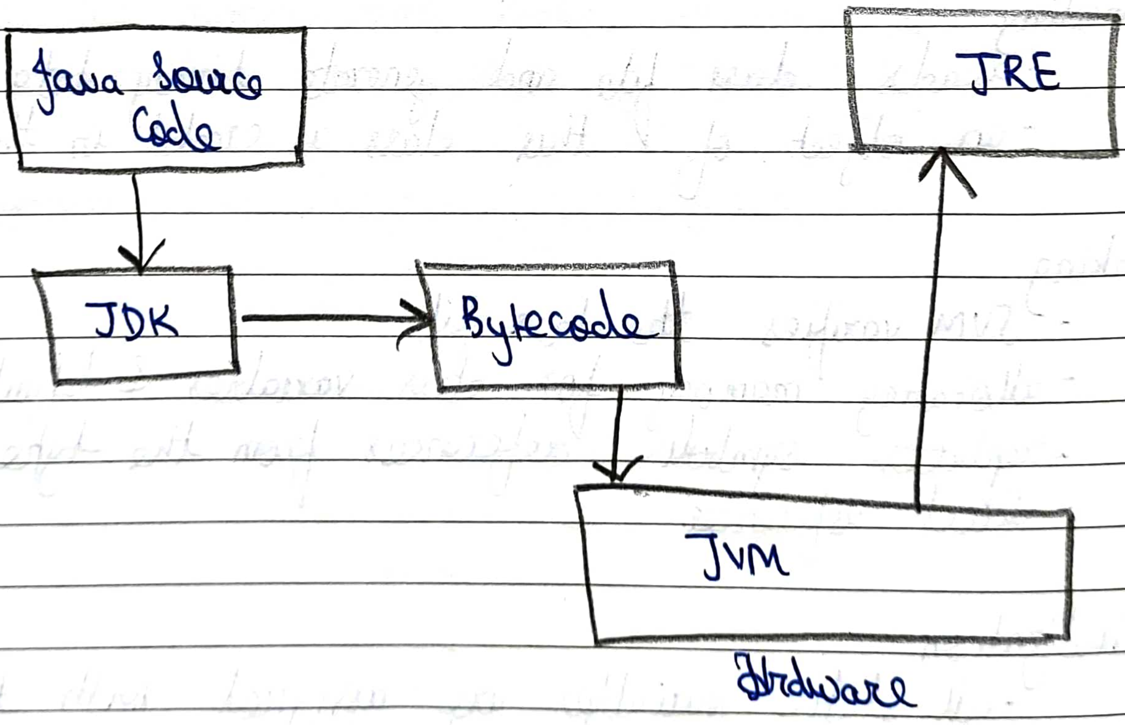
- Initalization
  - all static variables are assigned with their values defined in code and static block.

* JVM Execution

- Interpreter
  - line by line execution
  - when 1 method is called again & again, it will interpret again and again

- JIT
  - that methods that are ~~represented~~ repeated JIT provides direct machine code so re-interpretation is not required
  - makes execution faster
  - Garbage Collection

* All Steps –



* Tools used – Java JDK
  - IDE – Intellij IDEA Community Edition.