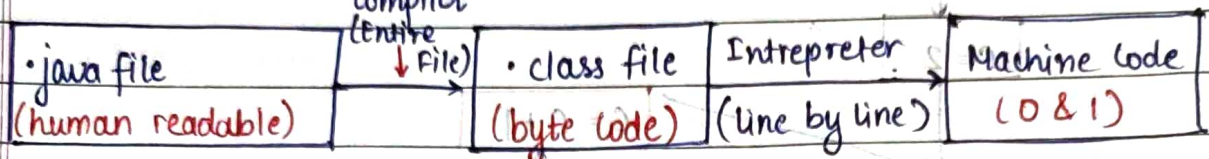
Quote of the Day:
"All our dreams can come true if we have courage to pursue them."

PAGE No.
DATE / /

Intro to Java (Architecture & Installation)

↳ How Java Code Executes:



In Java:
↳ this code will not directly run on a system.
↳ we need JVM (Java virtual Machine) to run this
↳ Reason why Java is platform independent.

Platform Independence:
↳ Byte code can run on all operating system.
↳ 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 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.

DATE / /

Architecture of Java: JDK vs JRE vs JVM vs JIT

JDK = JRE + Development Tools
(Java Development Kit)
↳ JRE = JVM + Library Classes
(Java Runtime Environment)
↳ JVM
(Java virtual machine)
↳ JIT
(Just-in-time)

JDK

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

JRE

- Installation package that provides env. to only run the prgm.
- Consists
  1. Deployment technologies.
  2. User Interface toolkits.

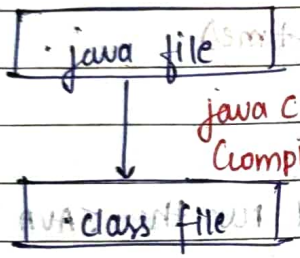
3. Integration libraries.
4. Basic libraries
5. Jvm

- After we get the .class file, the next things happen at runtime:

↳ Class loader loads all classes needed to execute the prgm.
↳ Jvm sends code to Byte code verifier to check the format of code.

Working of Jvm:

Compile Time



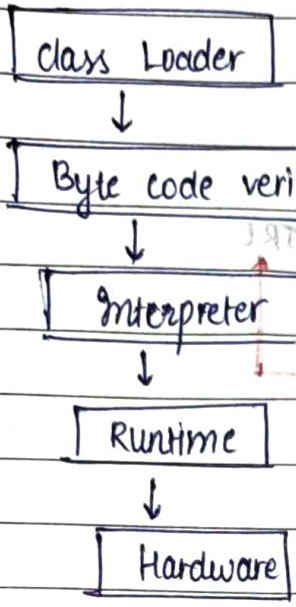
Jvm Execution:

↳ Intrepreter
- Line by line execution.
- One method called ~~again & again~~ many times it will interpret again & again.

↳ JIT
- Methods that are been repeated, JIT provides direct machine code (re-intepration not required).
- Makes execution faster. (Garbage Collector)

DATE / /

Runtime :



(How JVM works) Class Loader

↳ Loading

- reads .class file & generate binary data.
- object of class is created in heap.

↳ Linking

- Jvm verifies .class file.
- Allocate memory ~~from~~ for class variables & default values.
- Replace symbolic references from the type with direct reference.

↳ Initialization

- All static variables are assigned with their values defined in the code & static block

Jvm contains the stack & Heap Memory Allocation.

Working of Java Architecture