

OOP

7/1/25

JAVA

Sun microsystem

Team → The green team

1st name → Green talk → .gt

unofficial name
name change

oak

↓
Java

Ver 1 → 1995

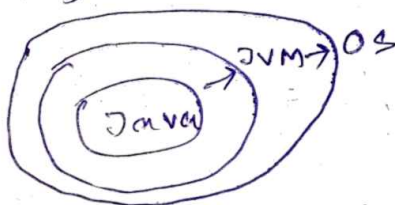
slogan: Write once Run Anywhere.

Q. Is Java oop?

Features of Java:

① Simple → Explicit pointer not available
Platform Independent

② Secure →



③ Robust →

strong library to handle error
memory management
(garbage collector method)
mark & sweep

④ Architecture - neutral.

int always 4 byte in any archi

⑤ Object oriented.

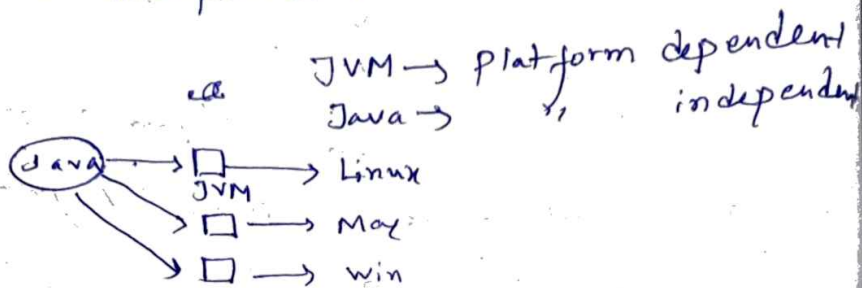
C → functions/implemented → procedural oriented language.
method.

oop
↓

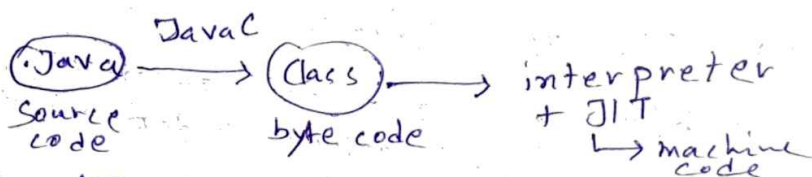
- i) Class/object
- ii) Abstract
- iii) Encapsulation
- iv) Polymorphism
- v) Inheritance.

⑥ Platform independent:

2/1/25



⑦ Interpreted:



In Java for execution of byte code into machine code, interpreter and Just In Time Compiler used together.

3 Translators -

- ① Java ~~compiler~~ compiler
- ② Interpreter
- ③ JIT compiler.

- ⑧ High performance
- ⑨ Dynamic - Memory management during Runtime.
- ⑩ Multithreading -
- ⑪ Distributed - It can work on different network.

class A → file name & class name can be different

```

{
    public static void main(String[] args)
    {
        System.out.printf(" ");
    }
}

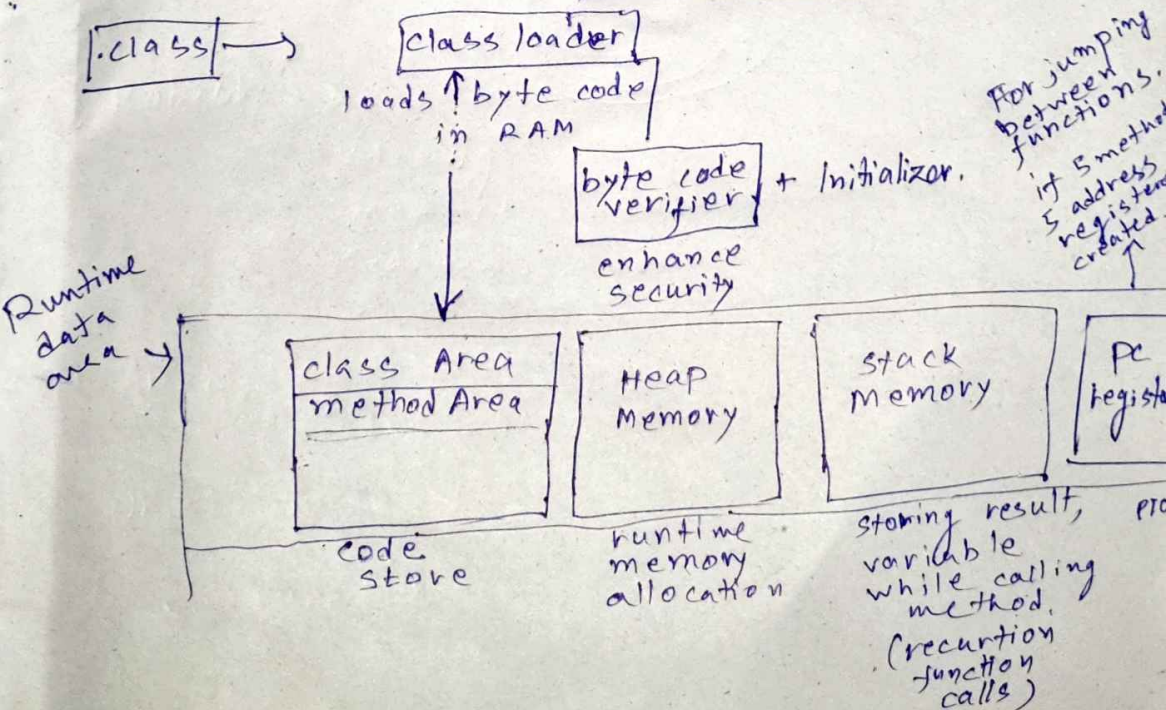
```

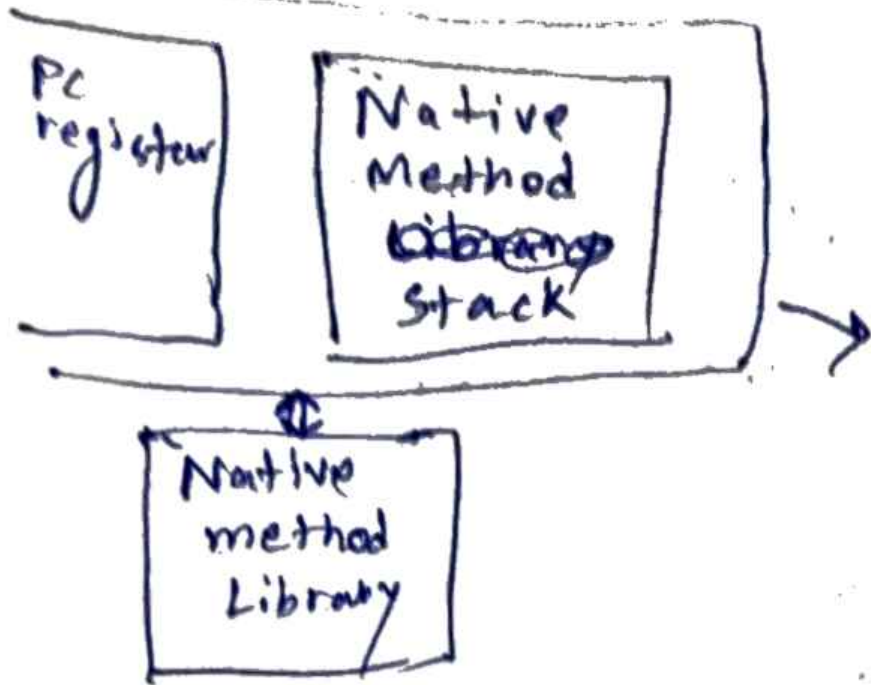
- ① Save as B.java
- ② Open cmd navigate to folder.
- ③ javac B.java ← (A.class will be generated)
- ④ java A

10/1/23

- ① Class name start with ~~A~~ upper case
- ② Class file / byte code is input of JVM
- ③ Java file is input of javac compiler.
- ④ max 1 class should be there.

JVM Architecture





Execution Engine
 Interpreter + JIT
 Executed byte code.

root file of Java → rt.jar.
 class loader

- ① Bootstrap.jar (uploads java classes root file)
- ② Extention
- ③ Application / system class loader (uploads code)

work of ~~load~~ class loader.

- loads byte code in main memory
- verifies the byte code.
- initialize the variables.