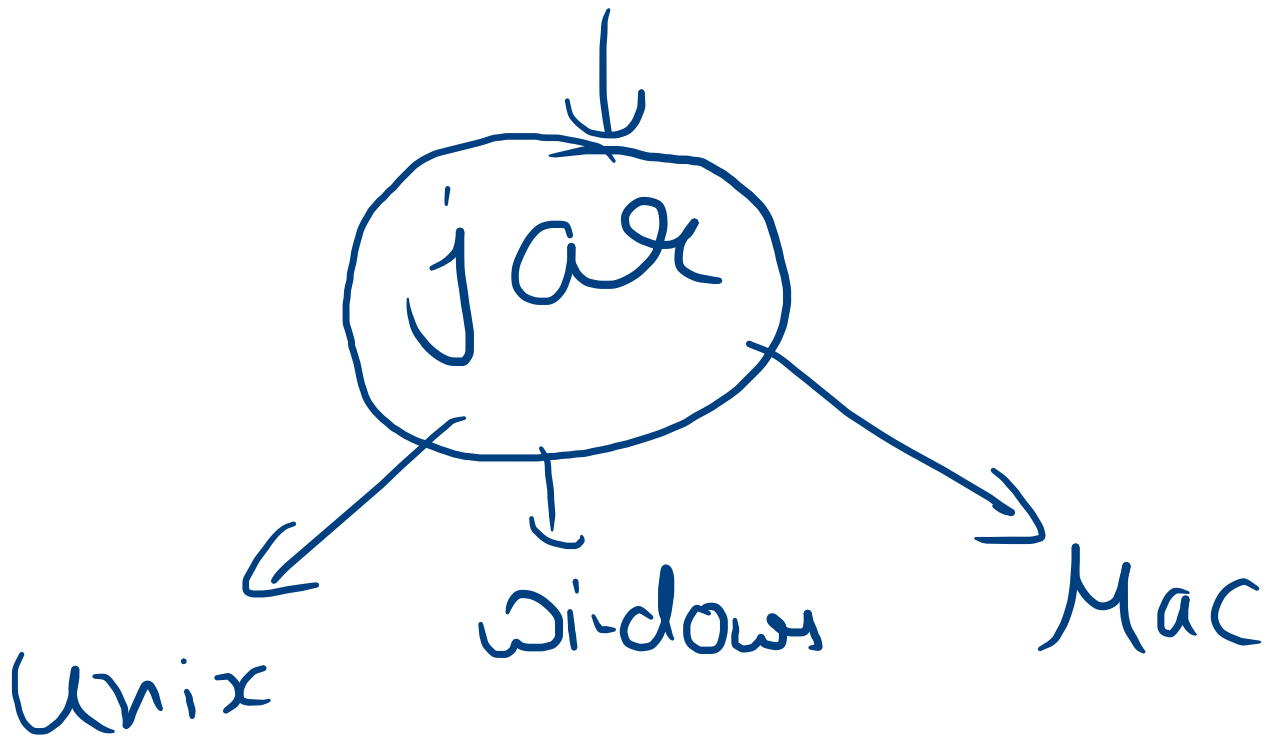
Why Java is so popular?

1. platform Independent
2. Object Oriented Lang.
   - Pure
   - Structured

(Maintainable)

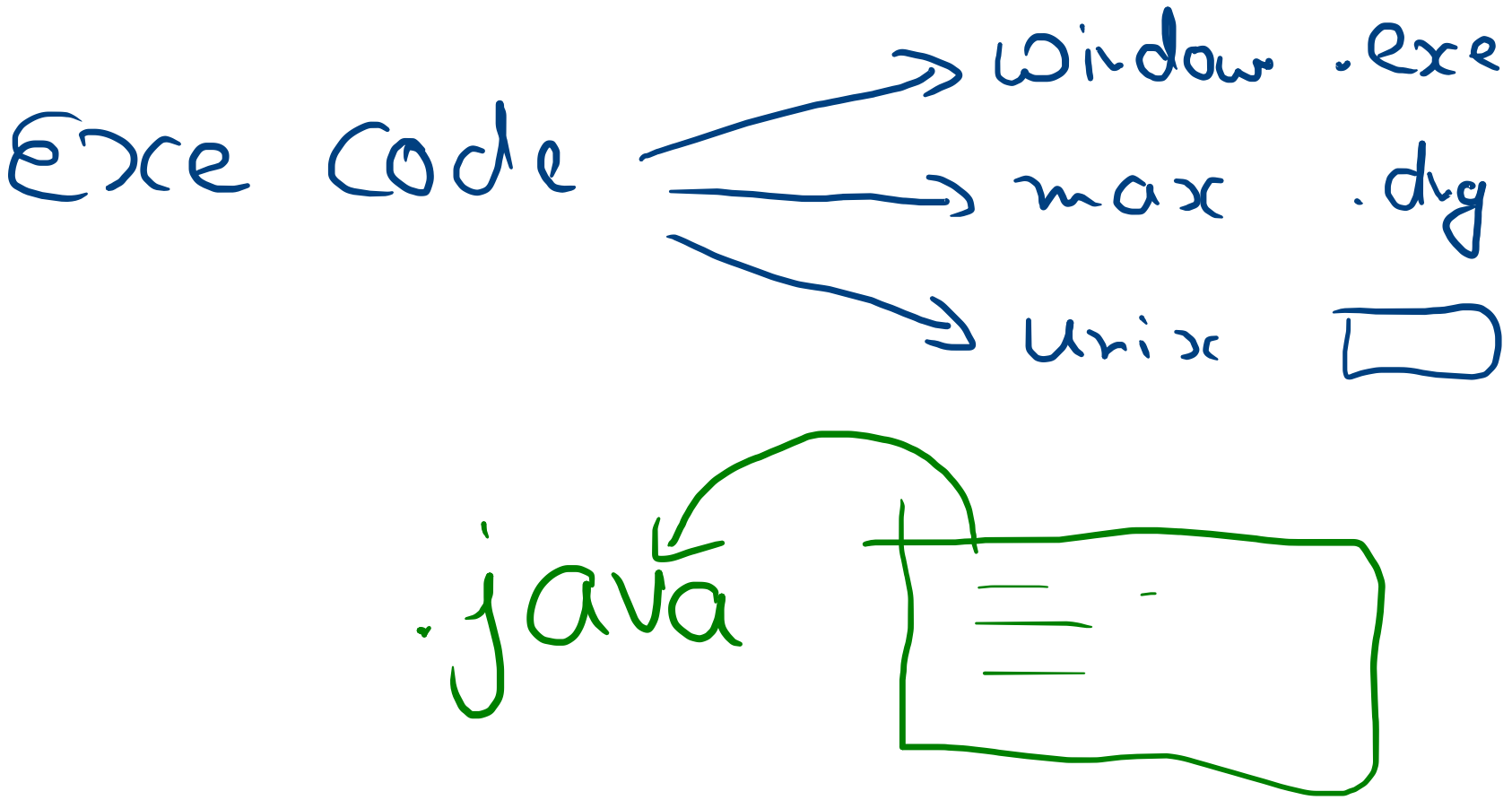
Java executable

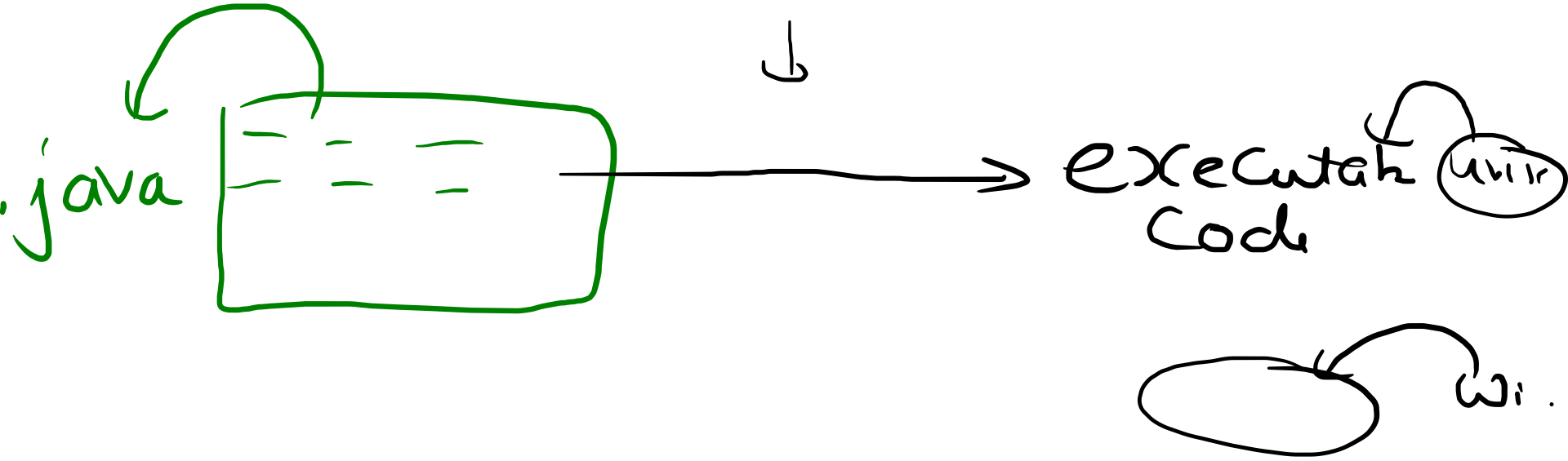


Platform Independence

We can build Java program anywhere & we would be able to run on any O.S

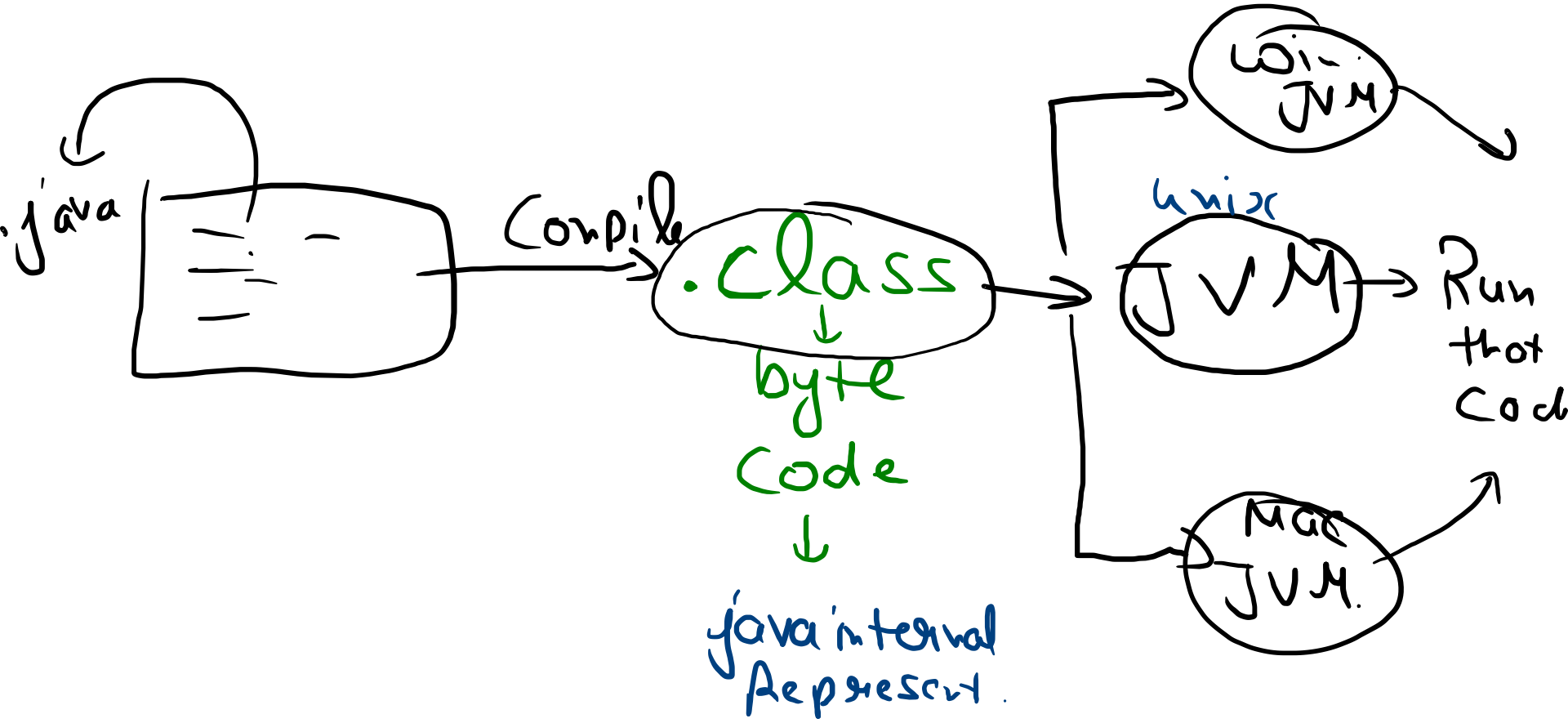
How does Java achieve it ?

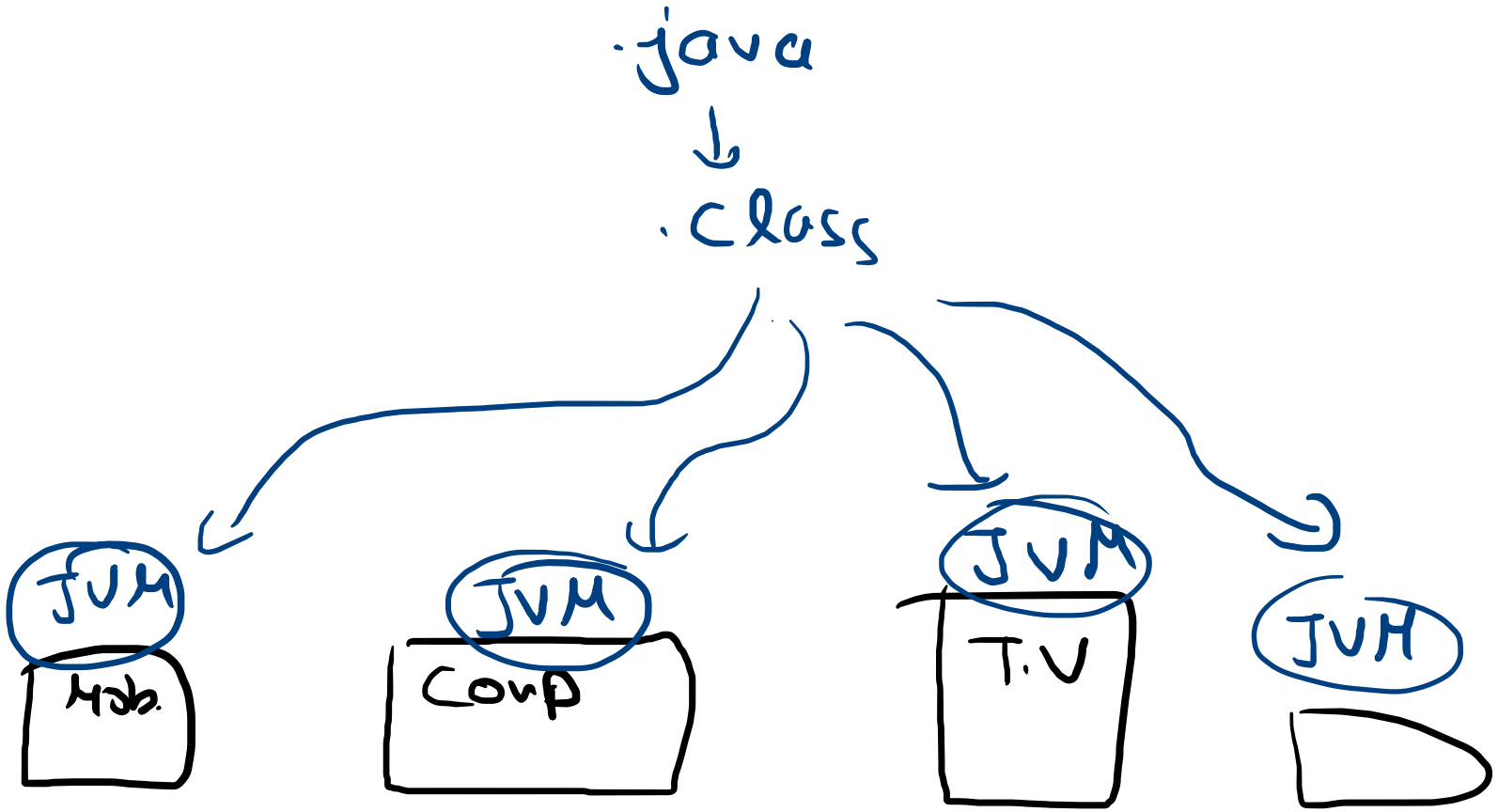


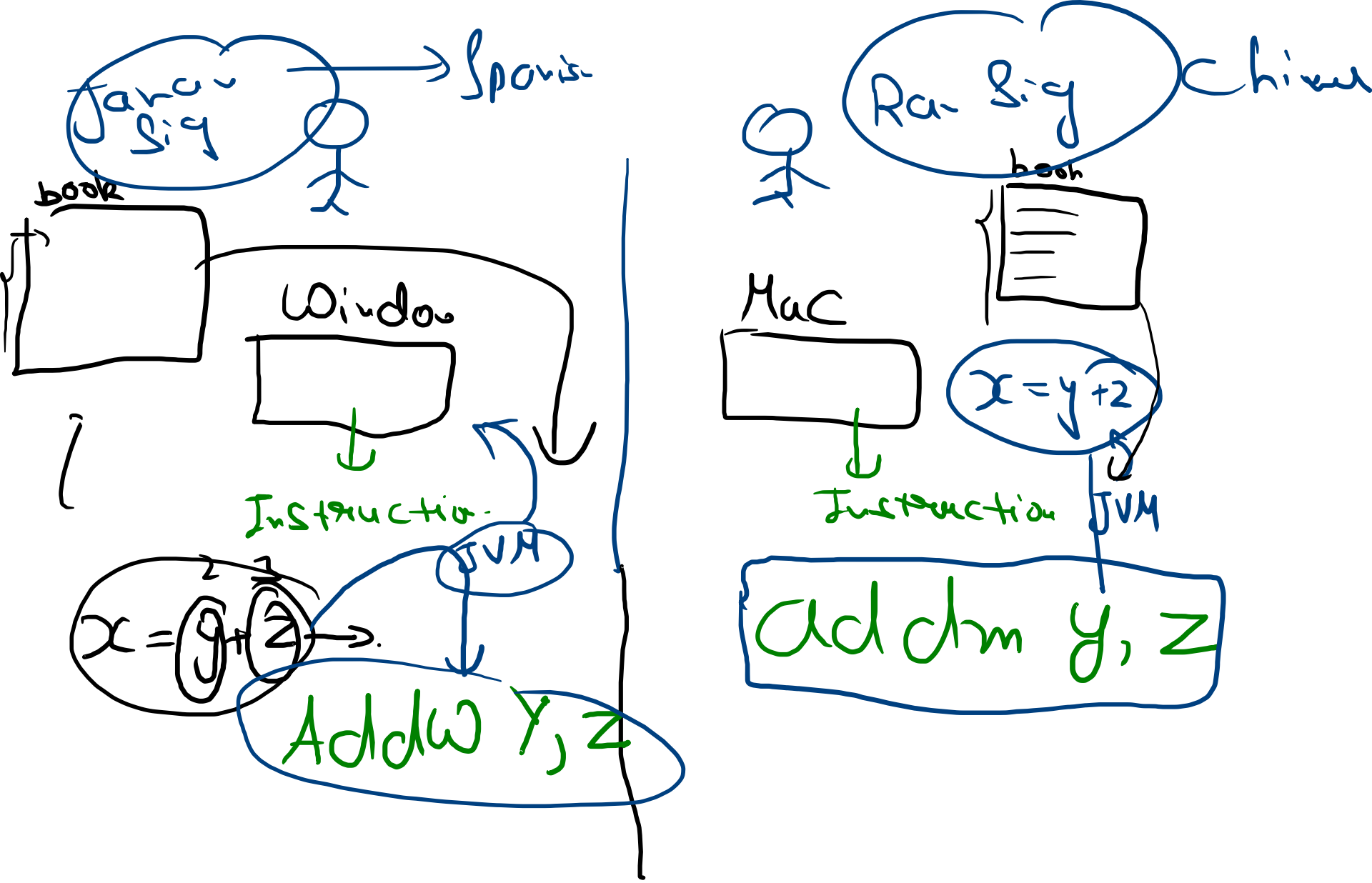

.java
executable
code

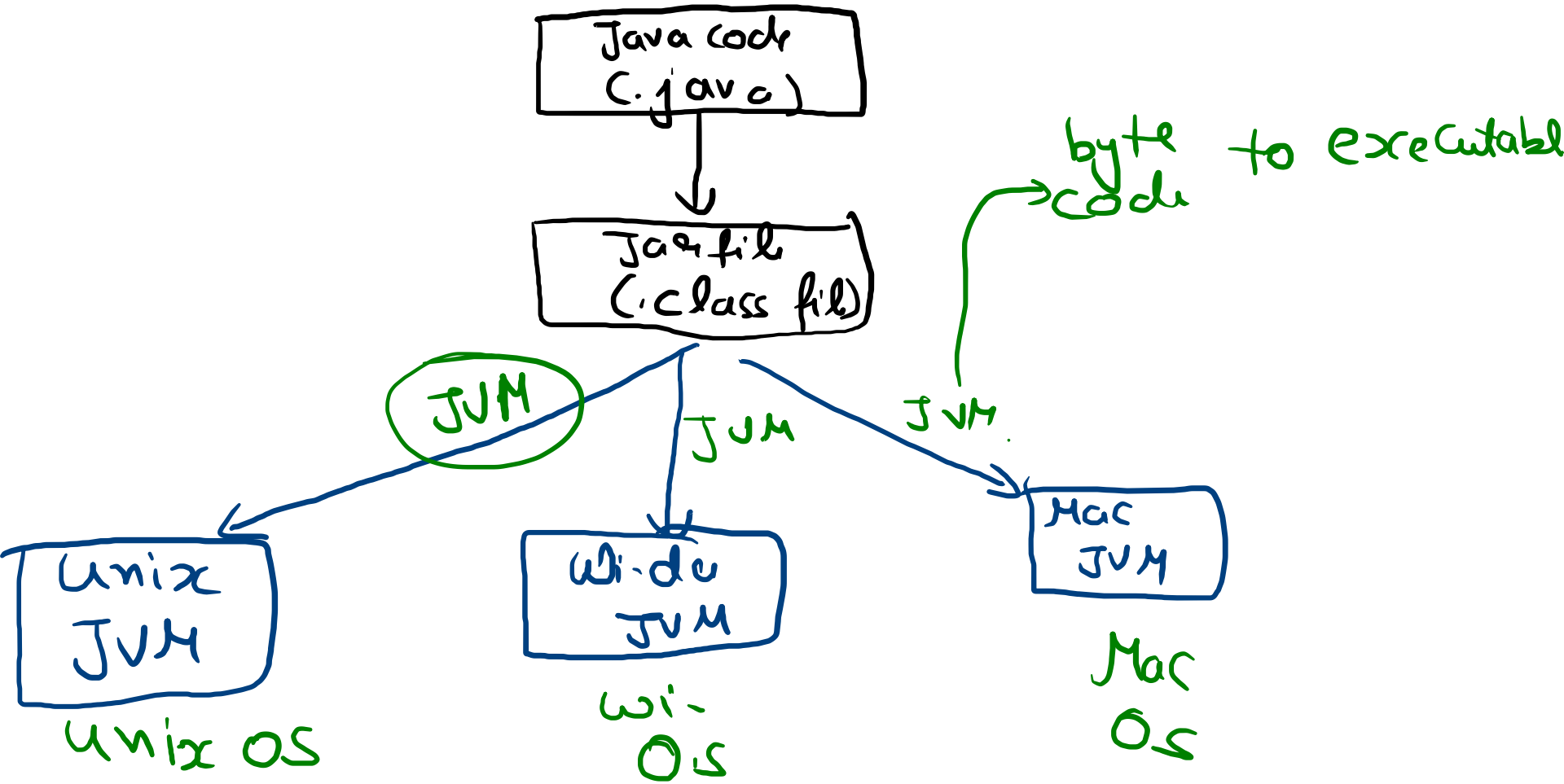
How we able to get diff executable Code in diff O.S.

JVM


.java
Compile
.class
byte
Code
java internal
Represent
Win-
JVM
Unix
JVM
Run
that
Cod
Mac
JVM.


.java
.class
JVM
Mob
JVM
Comp
JVM
T.V
JVM


Java Sig
Spanish
book
Window
Instruction
JVM
x = y + z
Addw Y, Z
Raw Sig
Chinese
book
Mac
x = y + 2
Instruction
JVM
Addm y, z


Java code
(.java)
Jar file
(.class file)
byte code
to executable
JVM
JVM
JVM
Unix
JVM
Unix OS
Wi-dv
JVM
Wi-
OS
Mac
JVM
Mac
OS

Java achieves platform Independency 2 concept

1. Byte Code (Internal Rep)
2. JVM

Java Pgm.

1. Compile Java Pgm. (create .class, jar file)

2. Run program. (Run .class)

3. Debug program [check error, check performance]

Consumer perspect.

2 types of peopl

1. Developers
   — Compil & Run the Code

2. User
   — use the program.
   — Run it

Java has 2 thing.

1. JDK (Developer)

— Interested in developing Java pgm

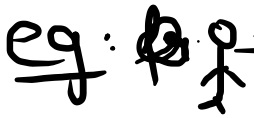
javac + jar + debugg.

+

JRE

JRE (user)

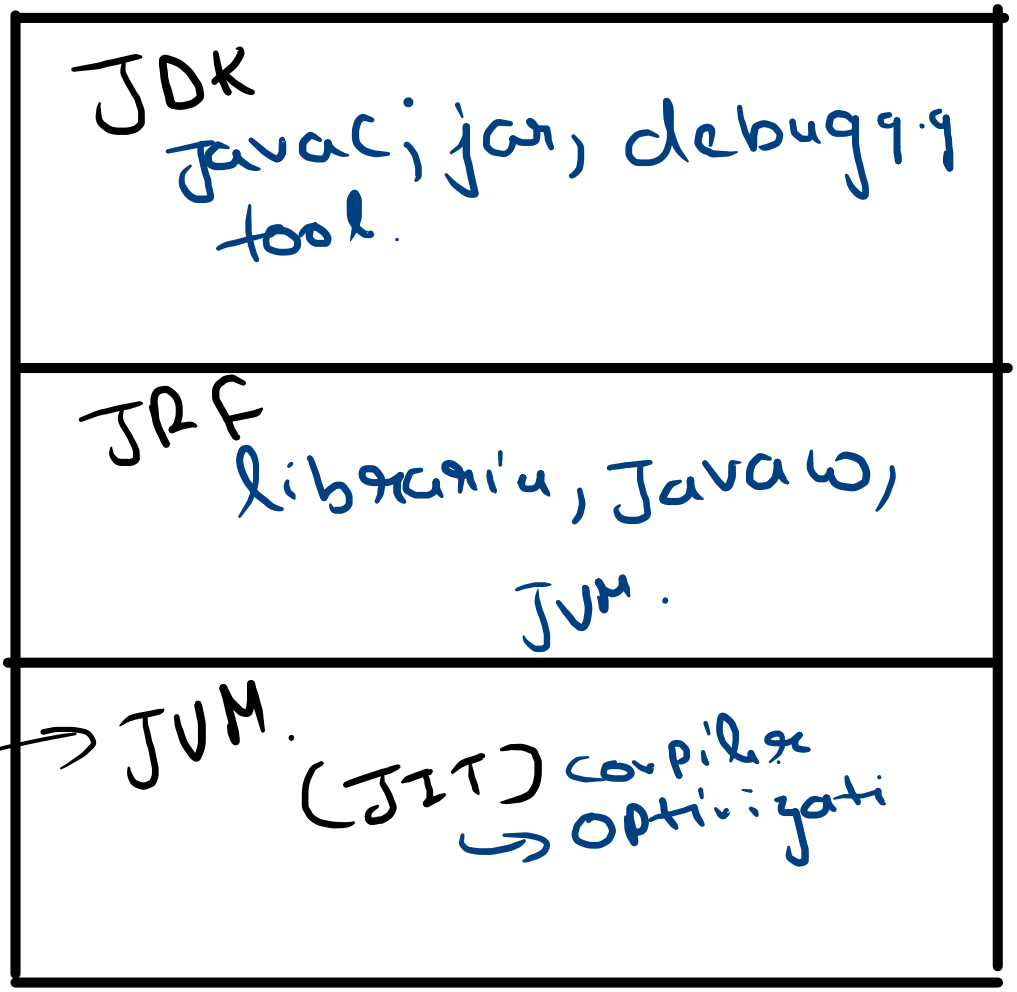
* Run a compiled Java pgm, JRE

eg:  have .jar → 

[Run] JRE

JRE → providing Env. to run java pgm

[JVM + Libraries]



JRE → Contains JVM.

JDK → Contai JRE

Java class Loader.

Java pgm → classes

3 type of classes

load

1. class that we write.
2. classes that used in framework.
   eg: Spring
3. class → Java Envi. provid.
   java.lang, java.collecti.

Run a java pgm.

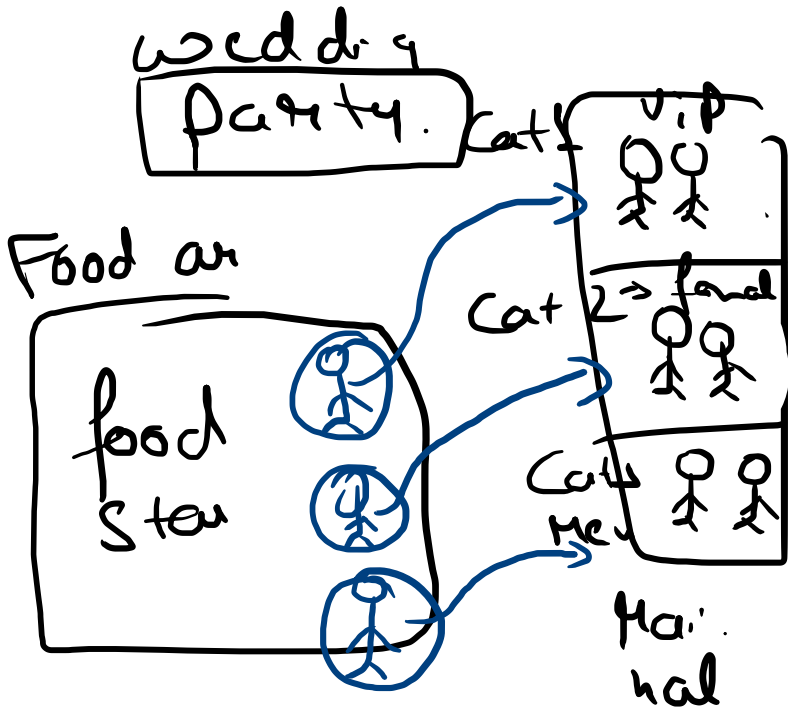
↓

need to load the classes.

↓

Class Load.

(find classes for us)



3 different kind of class Load.

Class load → Searches the class for us

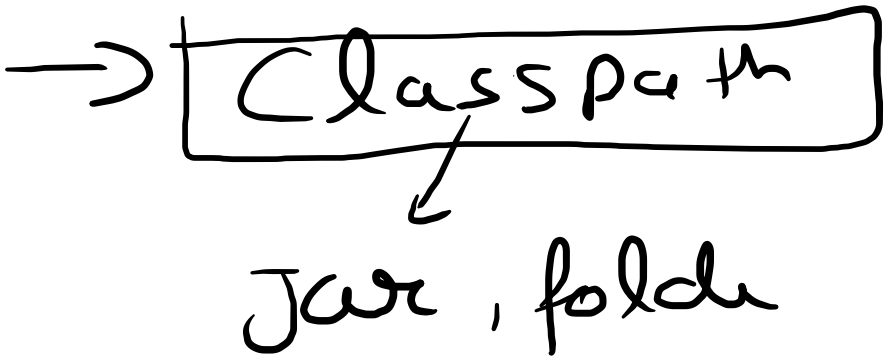
Types of Loads → depends on where they would for the class

3 type

1. Syste class Loader
2. Extension " "
3. Bootstrap class Load

1. System class Loader

— Loading classes from classpath.



2. Extension class Loader.

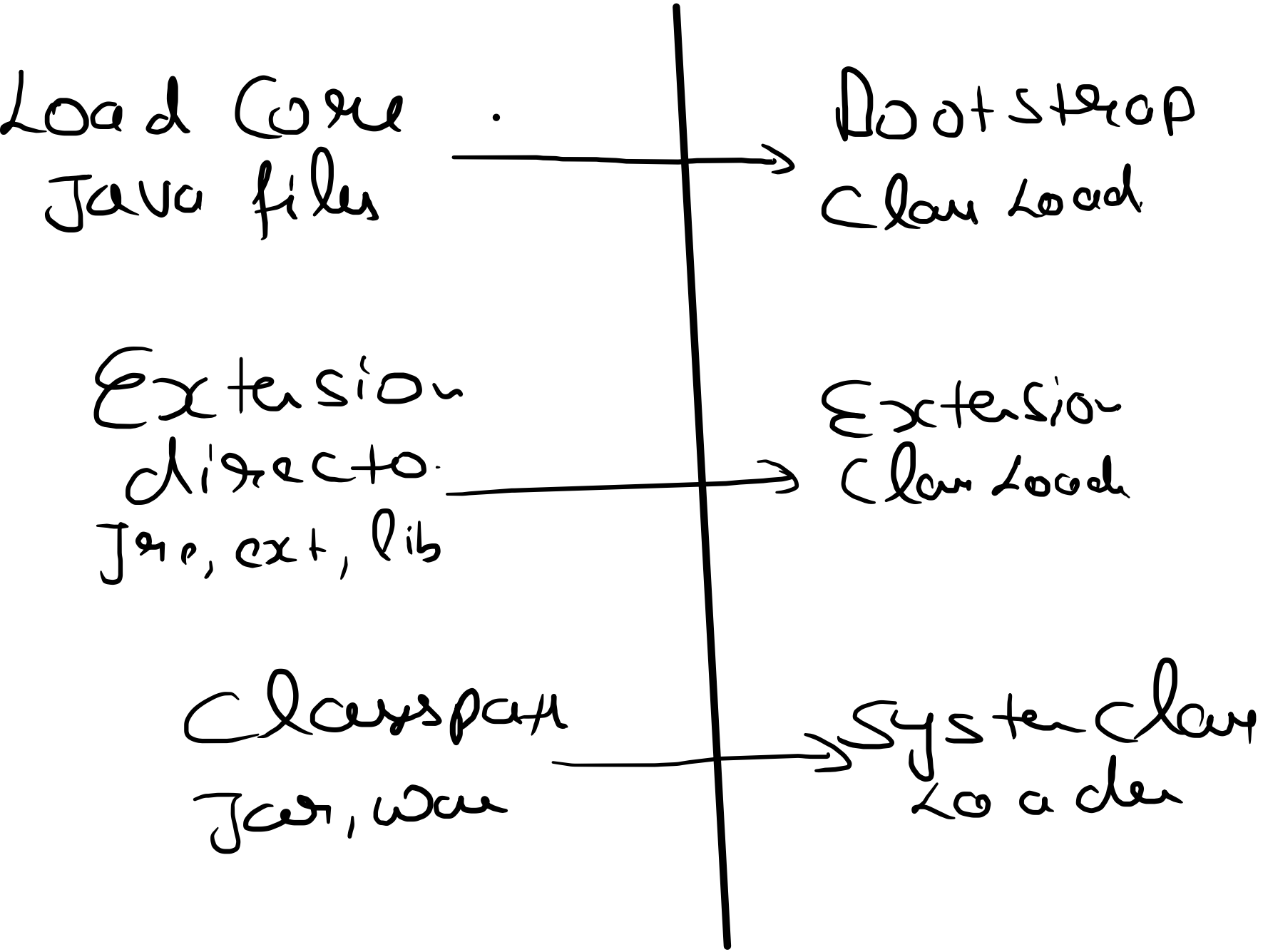
— It search the class in extension directory

JRE, JDK has a dir.
ext, jre, lib

— Exted. the extra libraries.

3. Bootstrap class load

— Loads all the Java Core fil..
(Basic java classes)


Load Core
Java files
Bootstrap
Class Load
Extension
director
Jre, ext, lib
Extension
Class Loader
Classpath
Jar, war
System Class
Loader

Wrapper class "12"

* wrapper around the basic primitive variables.

eg: Integer is a wrapper cla around int.

* For all primitive typ, Java provid the wrapper clas.

Need of wrapper class

1. lot of utility Methods.
   - Min_VALUE
   - parseInt
2. Null value
3. Java collection only support objects.

   Wrapper used in java collection

   hashmap.put(key, 5)

   Autoboxing

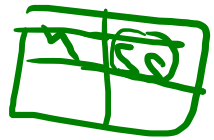
4. Create object from other types

Convert String value to Integer type

```
Integer n = new Integer("55");
```

n = 55

Different ways of creating Wrapper Classes



```
Integer n = new Integer(55); //int
   "    n1 =  "      "  ("55"); //Strig
```

```
Float x1 = new Flood(55.0); //doubl
   "    x2 =  "   "    (55.0f); // flod
Float x3 =  "    "   ("55.0f"); //Strig
```

✓ Character C1 = new Character('C');

✗ // C2 = 'A';

// Character C3 = new Character('A');

// error

valueOf Static Methods

```
Integer x = Integer.valueOf("100");
```

1. using Constructer.

* always create a new object.

2. String valueOf.

* there might be a chance that we are using a cached value.

For eg. the cached value.

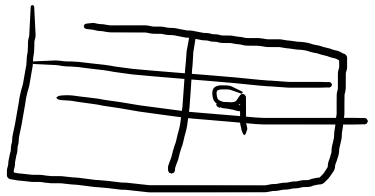
[-128 to 127]

```
public static Integer valueOf(int i)
{
    if(i >= IntegerCache.low && i <= Integer(cache.high))
            return IntegerCache.cache[i +
                                      (- - - -]

    else
        return new Integer(i);


}
```

* Save lot of Memory.
  (Reusig the Object)

* Wrapper classs are immutable

Autoboxing




```
Integer x = 9;
```

valueOf(9)

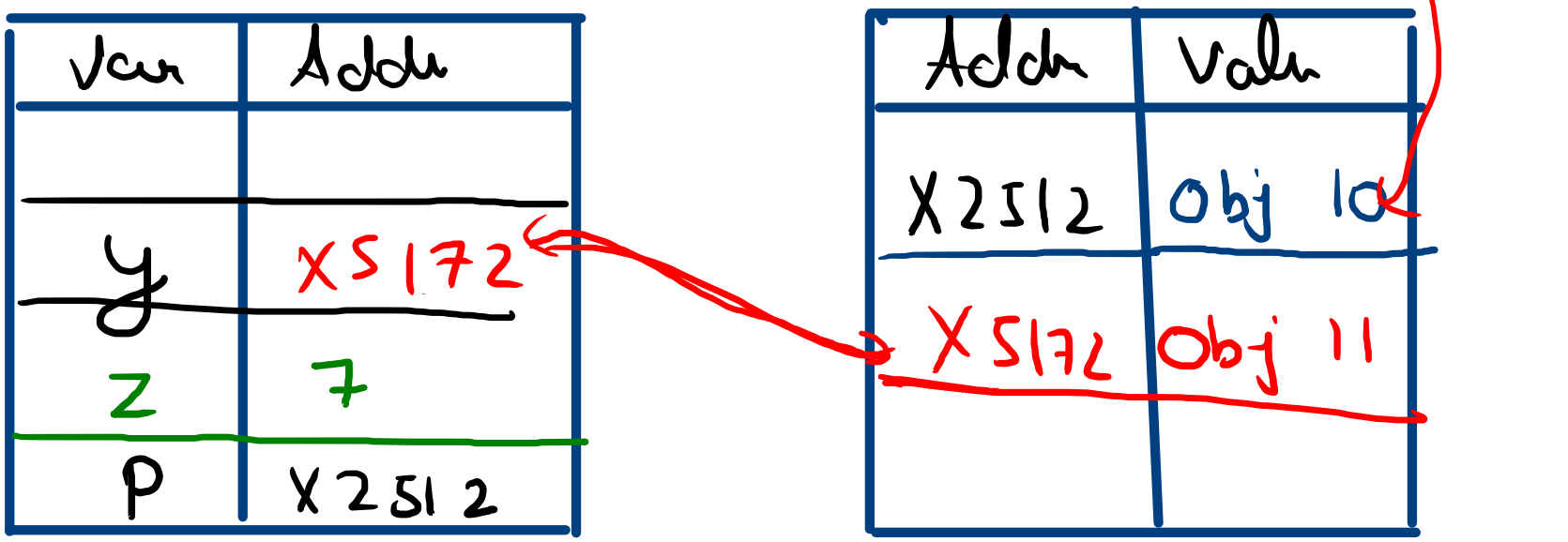
After Java 5,

```
int z = 7;
Integer y = new Integer(10);
y++;
Integer p = 10
```

(1)

(Variable Left Side)

(Right Side)

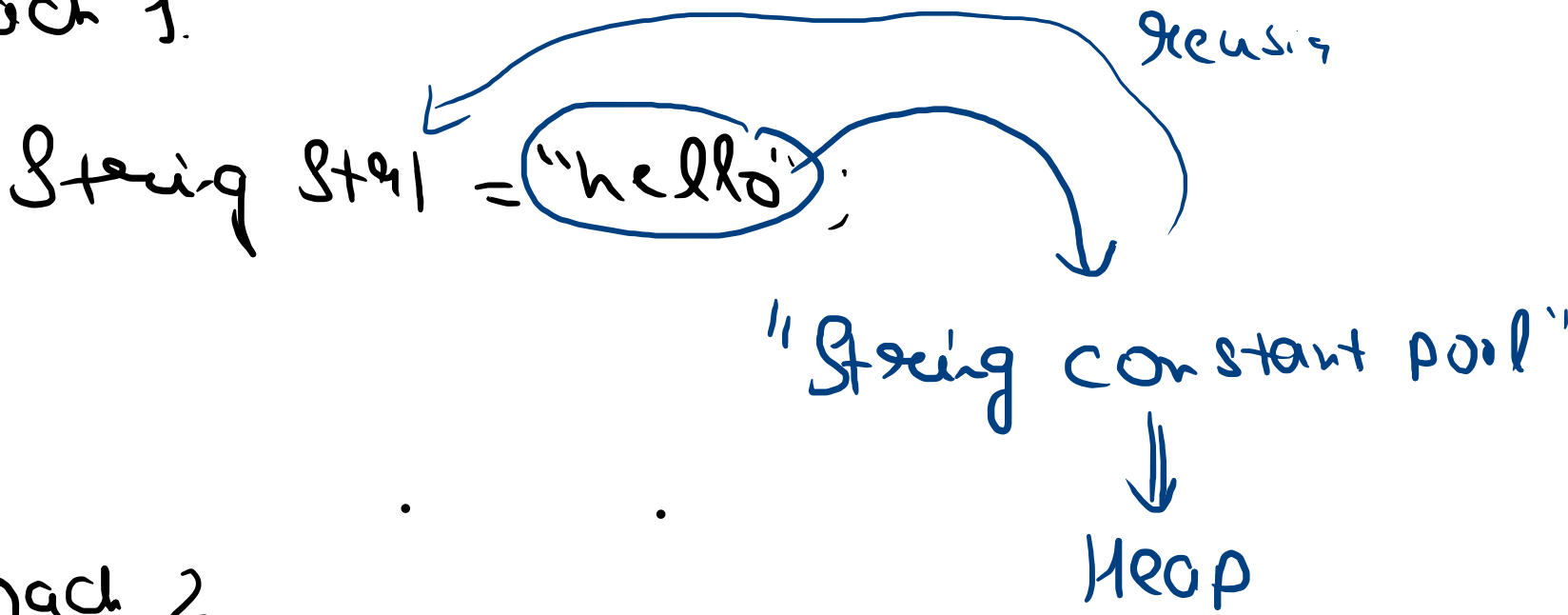


String Immutable.

concat

+

Approach 1.



Approach 2.

String str2 = new String("value");

$S3 = S3 + S2$

$S4 = hV$

$S5 = hVV$

$S3 = "h"$

$S2 = "V"$

Performance Impact.

StringBuilder
(not thread
Safe)

StringBuffer
(thread
Safe)

Synchronized

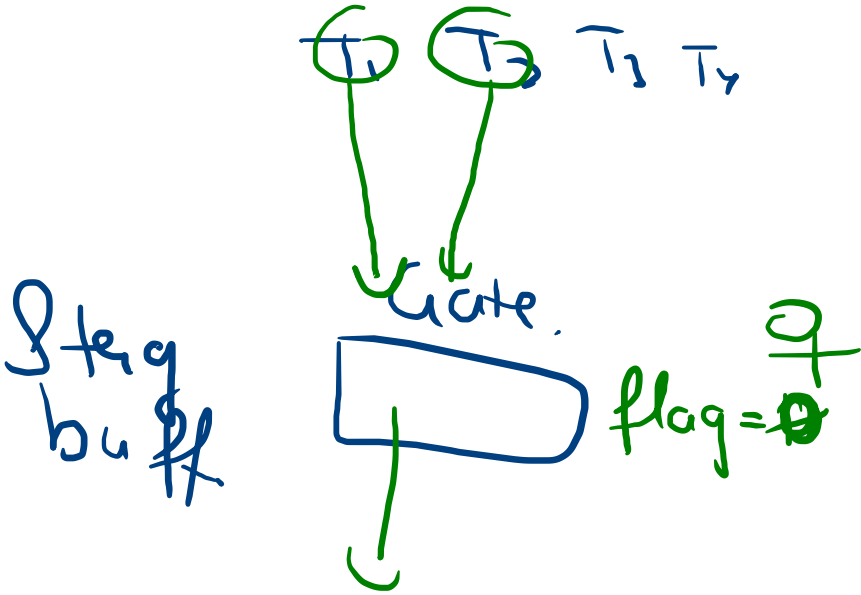
String.

StringBuilder.

T_1 T_2 T_3 T_4

String build

Synchronization problem occ.



2 ways

1st way → Notion way

2nd way