
IT602: Object-Oriented Programming



Lecture - 02

The Java Ecosystem

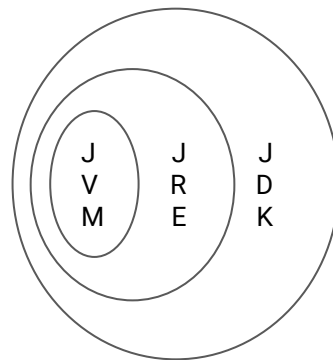
Arpit Rana

20th Jan 2022

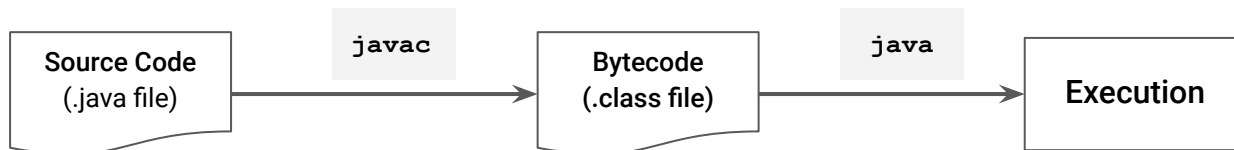
Java Ecosystem

The three most basic parts of the Java ecosystem are -

- Java Virtual Machine (**JVM**),
- Java Runtime Environment (**JRE**), and
- Java Development Kit (**JDK**)



How Java Works



On a terminal editor
(e.g. vim, nano) OR
on a GUI editor
(e.g. gedit, sublime) OR
on an IDE
(e.g. IntelliJ IDEA,
Eclipse, Netbeans)

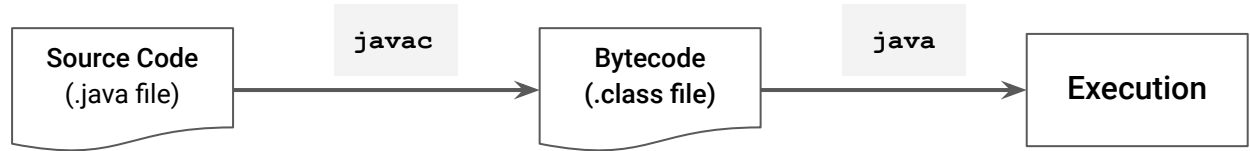
Bytecode is an
intermediate state
in *hexadecimal*
format with
opcode-operand
lines.

Opcodes and operands

Each assembly language statement is split into an opcode and an operand. The opcode is the instruction that is executed by the CPU and the operand is the data or memory location used to execute that instruction.

JVM can interpret
bytecode
(without further
recompilations)
into native machine
language.

How Java Works



On a terminal editor
(e.g. vim, nano) OR
on a GUI editor
(e.g. gedit, sublime) OR
on an IDE
(e.g. IntelliJ IDEA,
Eclipse, Netbeans)

Bytecode is an
intermediate state
in *hexadecimal*
format with
opcode-operand
lines.

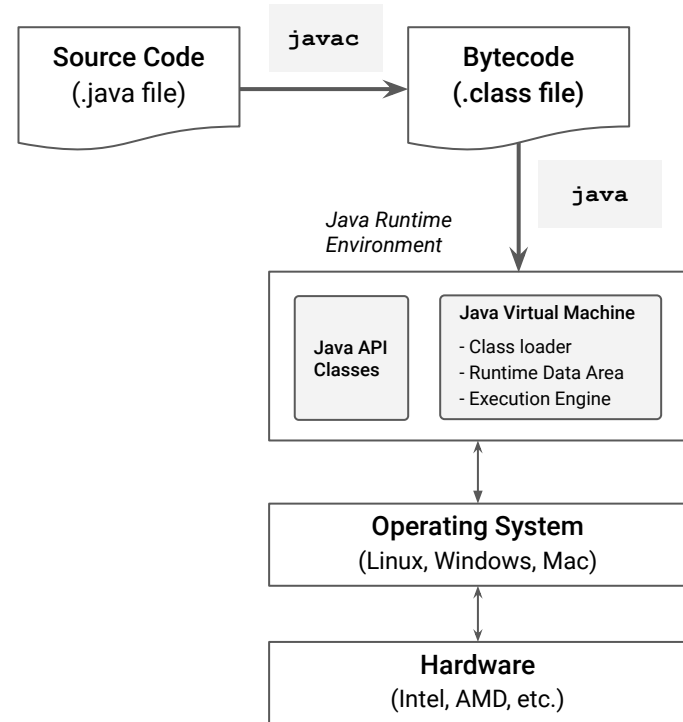
JVM can interpret
bytecode
(without further
recompilations)
into native machine
language.

Bytecode is portable among any JVM
regardless of underlying OS (Linux, Windows, Mac)
and hardware architecture (x86, x64).

Java Ecosystem: JVM

Java Virtual Machine -

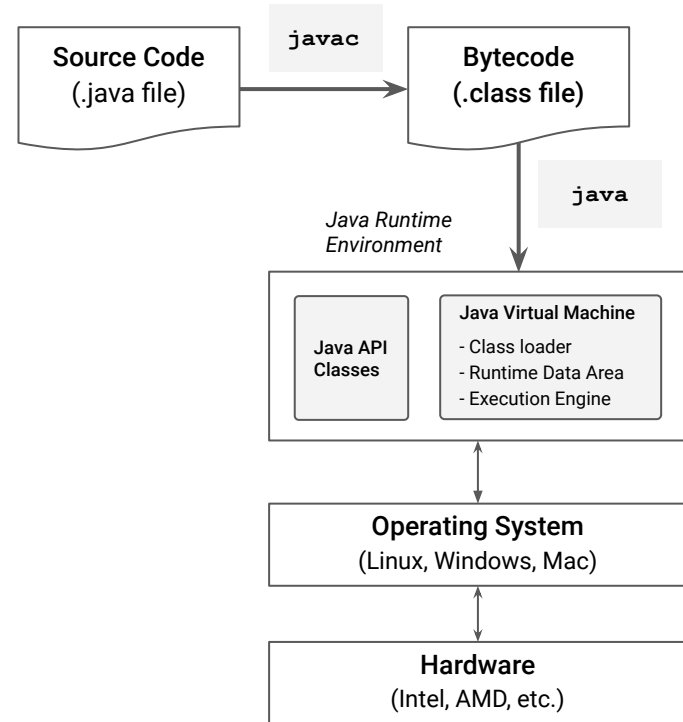
- On running a Java program, an instance of JVM is created
- It prevents the running of the code from clashing with other programs within the system.
 - It acts like a sandbox i.e. it can access only certain resources, programs, and files within the system.
- It resides in the RAM and dynamically loads (loads, links, initializes) class file when referred for the first time (on runtime).
- It executes the instructions in the bytecode line-by-line by reading the data stored in runtime data areas.



Java Ecosystem: JRE

Java Runtime Environment -

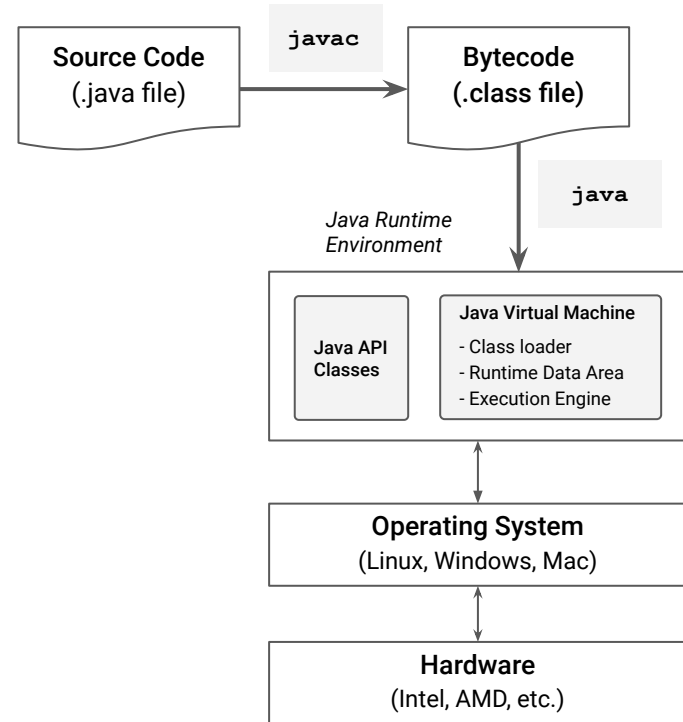
- It is the minimal requirement to run a java application on a computing device.
- It is a collection of JVM and the Java API classes.
- Java API classes include libraries for interaction with the host system, e.g. -
 - font management,
 - communication with the GUI,
 - the ability to play sounds, and
 - plugins for the execution of Java applets in the browser



Java Ecosystem: JDK

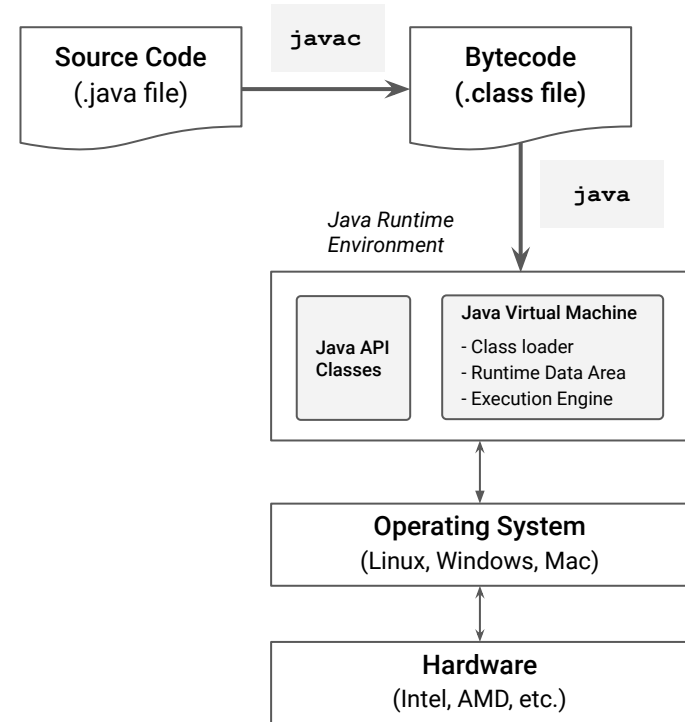
Java Development Kit -

- It contains all the programs that are needed to develop Java programs.
 - Its most important part is the Java Compiler (javac).
- The JDK also includes many auxiliary tools, e.g.
 - a Java disassembler (javap), a utility to create packages of Java applications (jar),
 - A system to generate documentation from source code (javadoc)
- The JDK is a superset of the JRE, i.e. if you have the JDK, then you also have the JRE (and the JVM).



Facts about Java Language

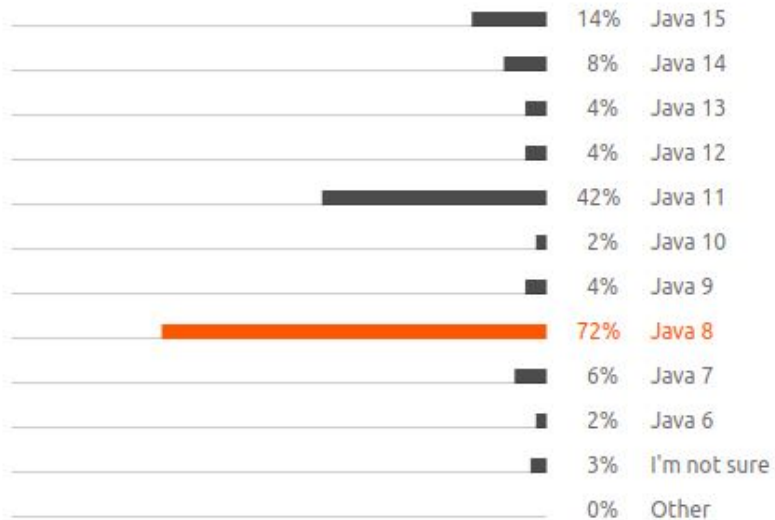
- Java is truly Object-Oriented Programming language.
- Java is both interpreted and compiled language.
- Java is Platform Independent / Architecture Neutral language; however, JVM (and JRE) are platform dependent.
- Java is dynamic and distributed (JVM supports dynamic class loading from the local file system as well as from machines on the network).
- Java is secure (the sandbox model confines the program execution to an instance of the JVM).
- By design, Java is slow due to dynamic linking and run-time interpreting.



Practicalities

In this course, we will use Java version 8.

Which versions of Java do you regularly use?

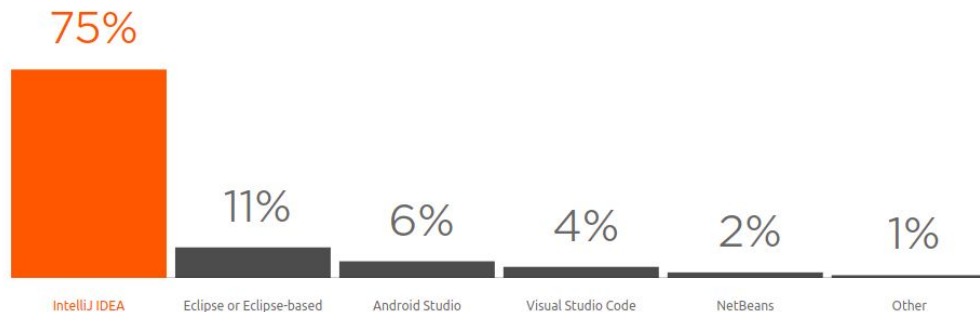


– Sourced from JetBrains

Practicalities

In our labs, our TAs will use IntelliJ IDEA.

Which IDE / editor do you use the most for Java development?



Despite all the measures we've taken to secure a representative pool of respondents, the results may be slightly shifted towards users of JetBrains products, as they are more likely to take the survey.

– Sourced from JetBrains

IT602: Object-Oriented Programming

**Next lecture -
Language
Fundamentals**
