Java

Il codice è un file .java, in java il compiler prende il file e lo trasforma NON nel linguaggio macchina ma in byte code che ha estensione .class. il byte code è un linguaggio intermedio che sta a metà tra quello umano e quello della macchina, abbiamo bisogno della java virtual machine per far girare questo codice e una volta che la jvm run il codice viene convertito nel linguaggio macchina così che il pc lo possa comprendere. In C/C++ questo passaggio è saltato. Ma questo è il motivo per cui java è indipendente dalla piattaforma, si occupa la jvm a convertirlo nella macchina giusta.

Java è indipendente: il codice può girare su ogni sistema operativo. La java virtual machine è invece platform dipendente! Ogni sistema operativo ha una java virtual machine che si occupa di tradurre nel proprio linguaggio macchina il codice java.

Architettura di java

Jdk (java development kit) = jre + dev tools

È un pacchetto di file che fornisce l’ambiente per sviluppare e far girare il programma che creiamo con java. Comprende:

* Development tools
* Jre per eseguire il programma
* Il compiler – javac
* Archivio – jar
* Generatore di documenti – javadoc
* Interprete/loader

Jre (java runtime environment) = jvm + library class

Installa pacchetti che servono per eseguire il programma. Comprende:

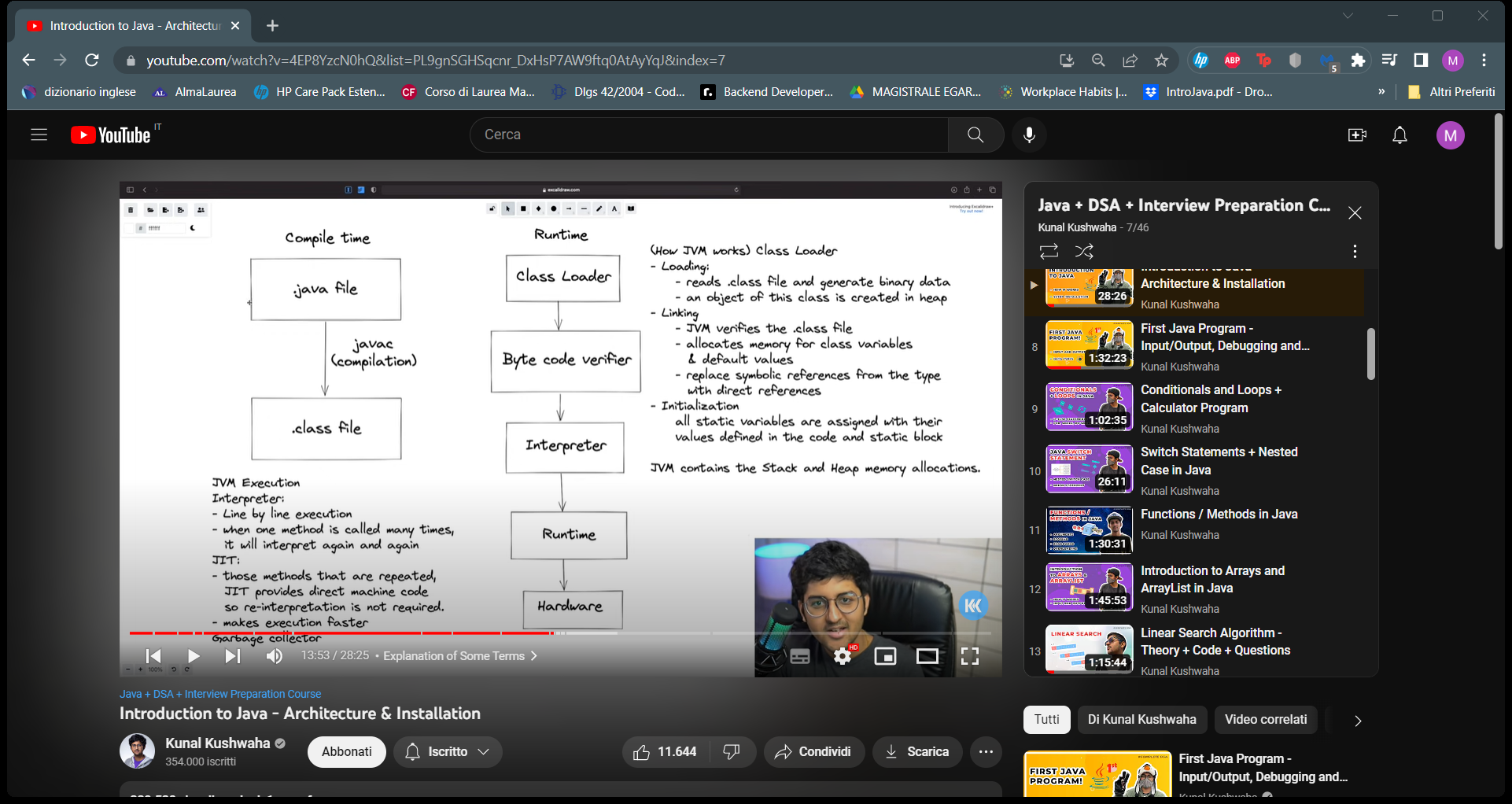
* Deployment technologie
* Interfaccia utente toolkits
* Librerie integrate
* Librerie di base
* Jvm

Dopo aver trasformato in .class quello che succede nella runtime è:

* Il caricatore delle classi carica tutte le classi che servono per eseguire il programma
* Jvm manda il codice al byte code verifier per controllarne il formato

Jvm (java virtual machine)

Jit (just-in-time)



public class Main {}

public: classe è accessibile a tutte le altre classi, file o package

Main è il nome della classe e deve essere sempre con la lettera maiuscola, deve avere il nome del file

Funzioni: blocchi di codice che possiamo riutilizzare all’infinito