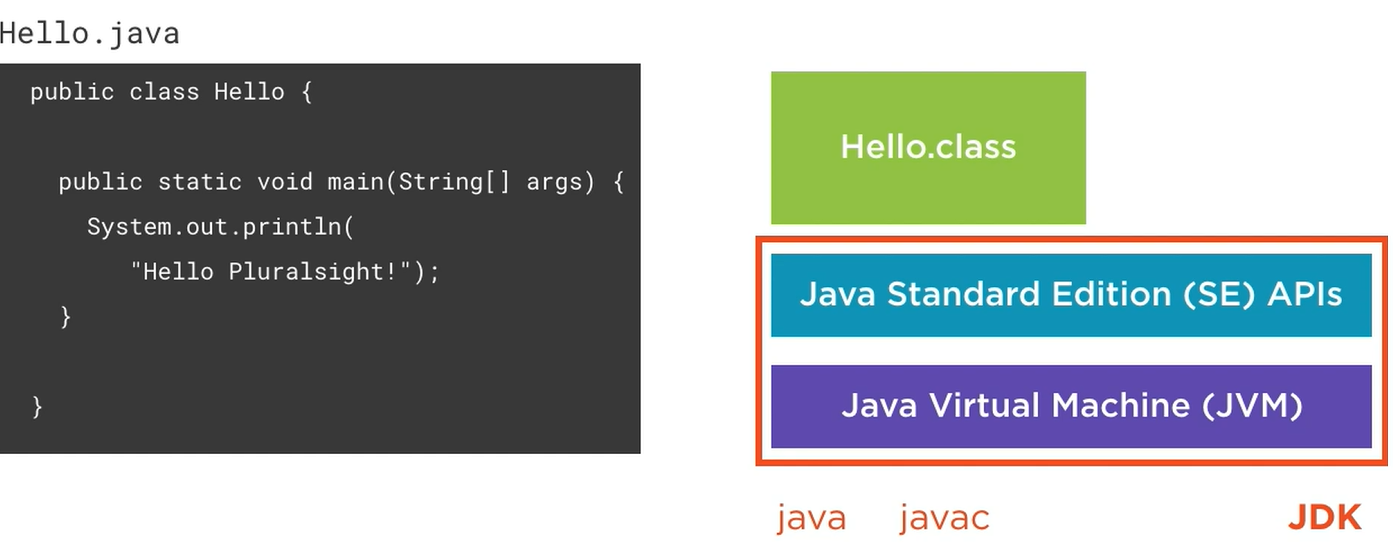
# Setting up a Java Development Environment

The javac compiler will convert the hello.java class in java byte code. The java byte code is system independent, in order to run this byte code we will be needing Java Standard Edition (SE) APIs and Java Virtual Machine (JVM).

JDK involved below things –

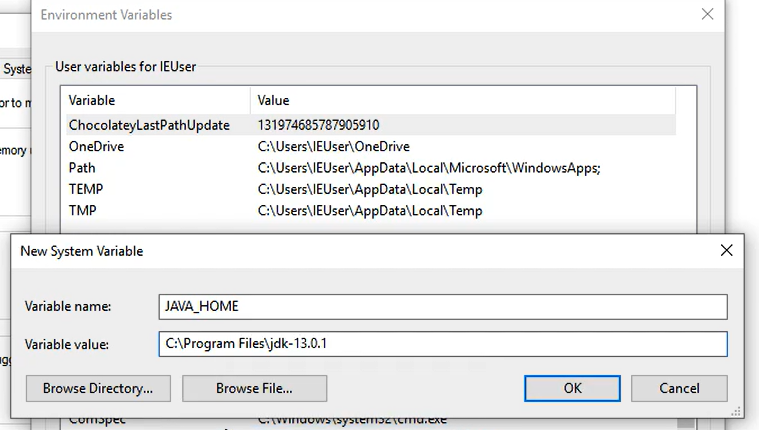


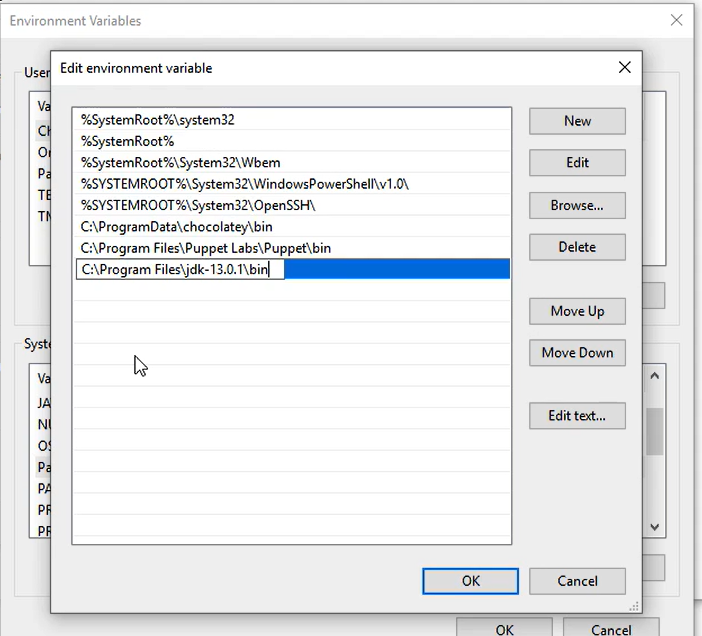


There is another thing called Java Enterprise (EE) APIs edition, these libraries are geared to web application development and enterprise application integrations and database access. Meanwhile JA SE edition edition APIs are more foundation APIs like networking, file access, creating and managing collections, etc. Java EE edition is not a part of JDK. Oracle has donated all the EE libraries to Eclipse foundation which lives on as Jakarta EE.



After downloading the JDK, we need to the JAVA\_HOME environment variable, it should point to the JDK folder and gets used by many tools that use Java. We also need to update the OS system’s PATH variable to include the directory of the JDK that contains tools like the Java compiler and the Java runtime so that we can use it from the command prompt or shell anywhere.





Popular Java IDEs – JetBrains IntelliJ Idea, Eclipse, Apache NetBeans.

The IntelliJ community edition is also the basis for the Google’s Android Studio for Android app developments. It used to be earlier Eclipse based, but now they have switched to IntelliJ.

Package name follows reverse-DNS naming conventions.

Java applications are packages into JAR files where JAR stands for Java Archive. It is simply a zip file containing our application classes in form of compiled code in byte format. This JAR file also contains metadata stored in a MANIFEST.MF file.



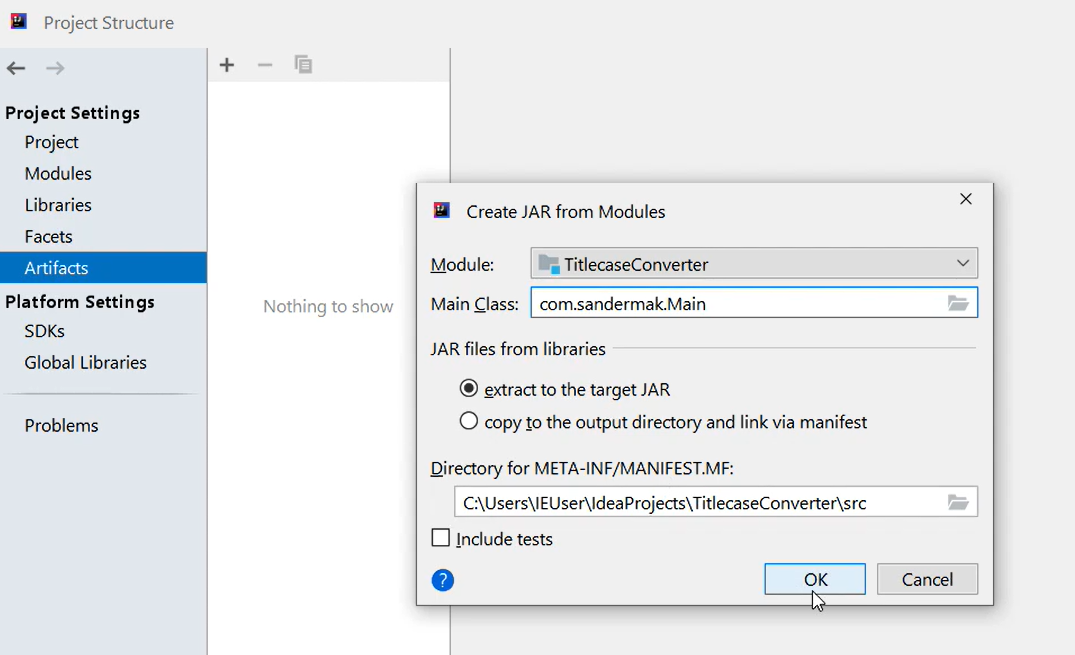
By using tool called “jar” in JDK we can create JAR files.

We metadata we also need to put which class is having main method so that JVM can invoke it properly –

A screenshot of a computer

Description automatically generated

Creating JAR file through IDE –



Instead of creating the JAR file manually through CLI or IDE, we generally use build tools like Maven or Gradle which have these steps to release a new package of the Java code – build -> test -> analyze (static code analysis) -> creating package.

# Modern Java: The Big Picture

Name JavaScript was only used by Netscape, which put JavaScript into the browser in the 90s to ride the surge of interest at that time in Java. Official name of the JavaScript is ECMAScript. Java belongs to Oracle, originally created by Sun Microsystems in the 90s, Oracle acquired Sun Microsystems.

The Java Platform includes programming language as Java, Runtime environment for CPU architecture independent and a standard library. All three are released under Java Development Kit (JDK).

A diagram of software components

Description automatically generated

Philosophy of Java – WORA (write once run anywhere, optimized for readability – reading code is more important than writing code. Maintainability over terseness. Understandable code over clever code, it also followed controlled deprecation.

Java achieved Portability by having JVM for each OS/architecture. Java SE APIs platform-agnostic.

Java driven by Java Community Process (JCP).

Java Classes can be clubbed into Packages and Packages further can be clubbed into Modules.

The JVM takes care of scheduling the operations on the different threads to different CPU cores.

With Java we have the platform-independent byte code and a platform-specific JVM to do just in time compilation for better performance for that specific underlying hardware.

Java is not suitable for real-time systems due to garbage collection like auto driving car, tight with operating system integration which need low level system access instead of this use C or C++, for quick prototyping.

Java syntax inspired by C language.

As python is an interpreted language, it runs slower than Java. Python can be a great fit for developing smaller applications or quick prototypes where performance is not of the utmost importance.

JavaScript was born as a language in the browser, however combining JavaScript with Node.js gives a same managed platform as Java and C#. Node.js took JS out of the browser onto service. Node.js provides a standard library and execution environments for JS on the server.

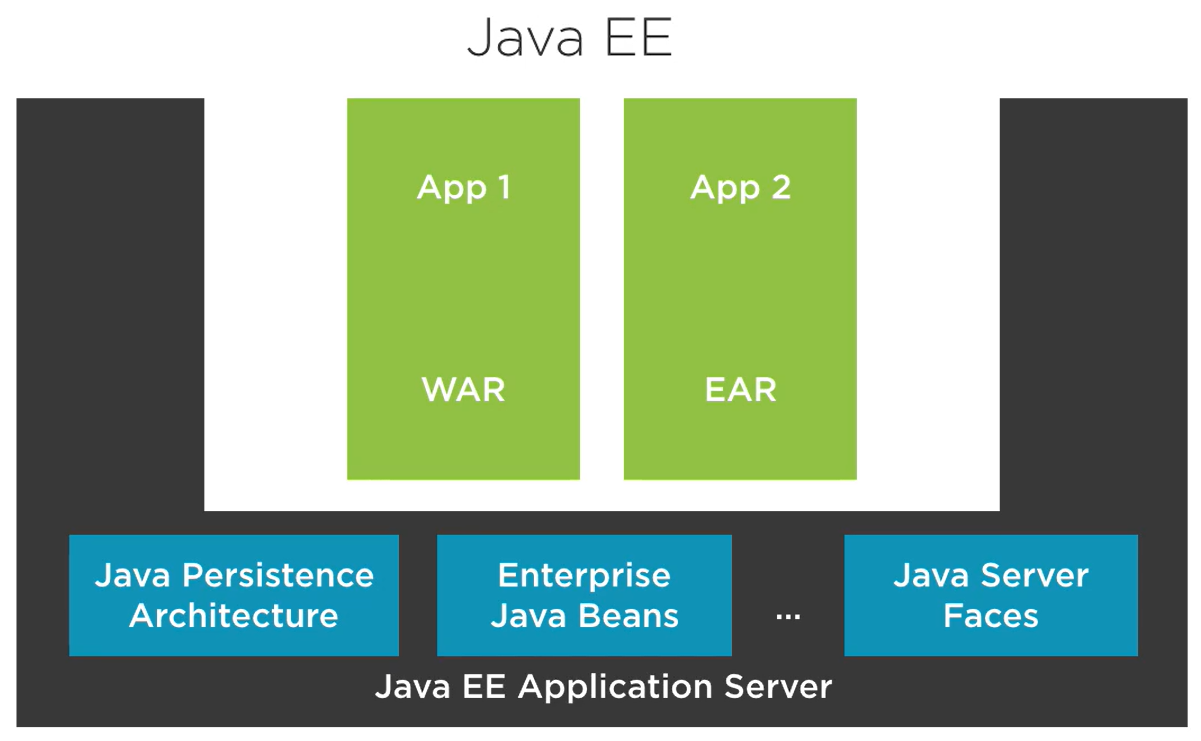
The computational model of JS and Node.js is inherently single-threaded. So if we want to fully use a multi-core machine using JS and Node.js, we need to launch multiple processes, and coordination between different processes is harder and more expensive than coordination between threads in Java.

Desktop java – initially AWT (Abstract windowing toolkit) was launched to design desktop applications, afterwards Swing was released which used pure Java GUI instead of OS native controls, after this JavaFX has been released which provided the XML based, declarative format for UIs called FXML and provided the advanced UI components with animation and effects and 3D graphics, now it is a part of OpenJFX.

Enterprise Java (Java EE) – in enterprise applications we typically have web front ends, transactional database access, integrations with all kinds of different systems, scheduled tasks, and so on. The Java EE provides these libraries without manually choosing them externally to avoid complexity and increase the productivity of the developer. It offers a set of interoperable enterprise APIs for common enterprise applications tasks.

A diagram of a software application

Description automatically generated with medium confidence

An application server is a container that runs on top of the JVM that can host Java EE applications. 

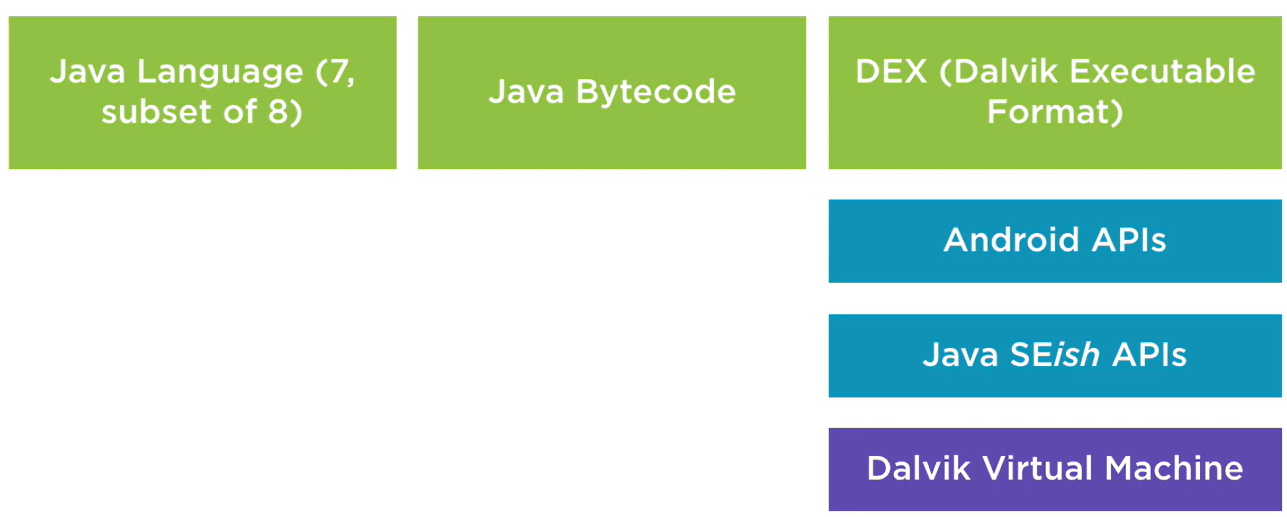
Java EE application servers – Wildfly (Red Hat), WebSphere (IBM), WebLogic (Oracle), Tomcat (Apache).

A competitor to Java EE, called Spring Framework.

Java in the Cloud – we use microframeworks to develop the application for Cloud and one of the popular framework is Spring Boot. Spring Boot is built on top of the Spring Framework and other Libraries that offer solutions for problems that we have in the cloud. It uses libraries from the Netflix open source deck. Netflix is one of the poster children for microservices architectures, they offer components for service discovery, fault tolerance, and everything hat is important in a cloud microservice architecture.

When we develop a spring boot application, ultimately everything is bundled up into a single Java application that runs on top of the JVM. There is no big application server involved anymore. Other popular micro frameworks are MicroProfile, Vert.x and Play Framework.

Android – Java is a primary language for writing Android applications. For Android the Java Bytecode converted into DEX (Dalvik Executable Format) which gets run on the Android platform. Android only supports older Java versions and a subset of the Java SE APIs. The DEX code is executed by Dalvik Virtual Machine instead of JVM which have different design goals like low power consumption of mobile phones.



Popular Java Libraries – EJBs are the way to structure business logic in Java EE applications. Spring framework sort of emerged as a “rebel” framework pitting itself against entrenched Java enterprise development practices. We can read more in the book “J2EE Development without EJB” by Rod Johnson.

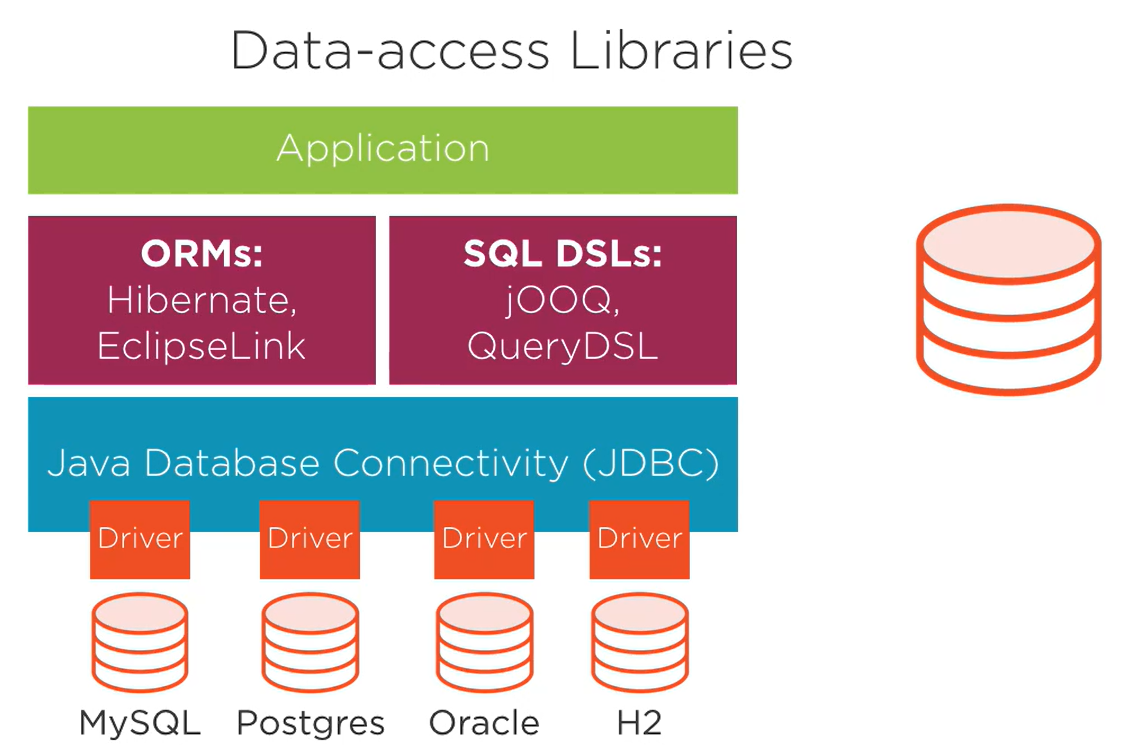
In Java EE application we can use Enterprise Java Beans to create transactional business logic. We can use Java Service Faces (JSF) to create web front ends. It is a very lighter alternative, it introduced the dependency injection (inversion of control), which is based on don’t call us, we will call you. The spring framework, at its core, is a dependency injection container that wires together the classes of our application based on instructions we provided. Sprint is just a framework which governs the structure of our application.

A diagram of a flowchart

Description automatically generated

Other commonly used libraries – Google Guava, Apache Commons, Apache Log4J, Netty, Akka, RxJava, Apache Camel (enterprise application integration), JDBC, Hibernate and EclipseLink are ORMs by which we don’t have to write the SQL queries anymore but Java objects.

There are two approaches to design Java JPA (Java Persistent Architecture) by using ORMs or SQL DSLs –



Java-based Data Processing libraries – Apache Hadoop, Apache Spark, DL4J (deep learning for Java), Cassandra, Neo4J (graph database), ElasticSearch, HDFS.

For unit testing we use Junit and Mockito.

With Maven we describe our builds in an XML file called pom.xml



Build tools – Maven, Gradle. Gradle supports incremental builds and uses Groovy scripts. Gradle is a default-built tool for the Android applications as endorsed by Google.

Continuous integration and quality control – one popular CI server is Jenkins.

Static code analysis tools for Java – Checkstyle, Spotbugs (byte code level), PMD (source code level), SonarQube (more advanced tool).

Other than Java we can use alternative JVM languages like Groovy, Scala, Kotlin, JRuby and JPython.

Kotlin is a “better Java” designed by JetBrains company, it is also endorsed by Google for Android development. Kotlin can compile into byte code and as-well in JavaScript code so we can target both browser and mobile platforms.