Chapter-6 Java Framework

What is Java?

- **Programming language** and a **platform**.
- Java is a high level, robust, object-oriented and secure programming language.
- The Java programming language was originally created in 1995 by James Gosling from Sun Microsystems (currently a subsidiary of Oracle Corporation).
- The goal was to provide a simpler and platform-independent alternative to C++.
- Java is a general-purpose programming language that's used in all industries for almost any type of application.

What is Platform?

- Any hardware or software environment in which a program runs.
- Java has its own run time environment known as **JRE** (Java Run-time Environment) and **JVM** (Java Virtual Machine) which converts Java code to machine code. It is called a platform.

What is the Differences between the Java platform and other platforms?

OR

Why java is platform independent engine?

The main differences between the Java platform and other platforms are:

- Java is **platform independent** because of this characteristic. We can write Java code in one platform and can be read/run in/on any other platform i.e. **WORA** (Write Once Read Anywhere). Other languages lack this capability.
- Java platform is a **software-only platform** that runs on the top of other hardware-based platforms, other platforms are mostly hardware software or hardware only and can be run only on hardware based.

- Programmer can develop Java code on **any OS**. Most of the other platforms do not have this capability.
- Java has its own run time environment known as **JRE** (Java Run-time Environment) and **JVM** (Java Virtual Machine) which converts Java code to machine code, whereas this functionality is missing in other platforms.

The Life Cycle of a Java Program

- Java requires the **source code** of our program to be compiled first.
- It gets converted to either machine-specific code or a **bytecode** that is understood by some run-time engine or a virtual machine.
- Not only will the program be checked for syntax errors by a Java compiler, but also some other libraries of Java code can be added **(linked)** to our program after the compilation is complete (deployment stage).
- Technically you can write the source code of our Java program in any plain text editor that we prefer (Notepad, TextEdit, vi, etc.), but to compile your program you need additional tools and code libraries that are included in the Java Development Kit (JDK).

JDK, JRE and JVM

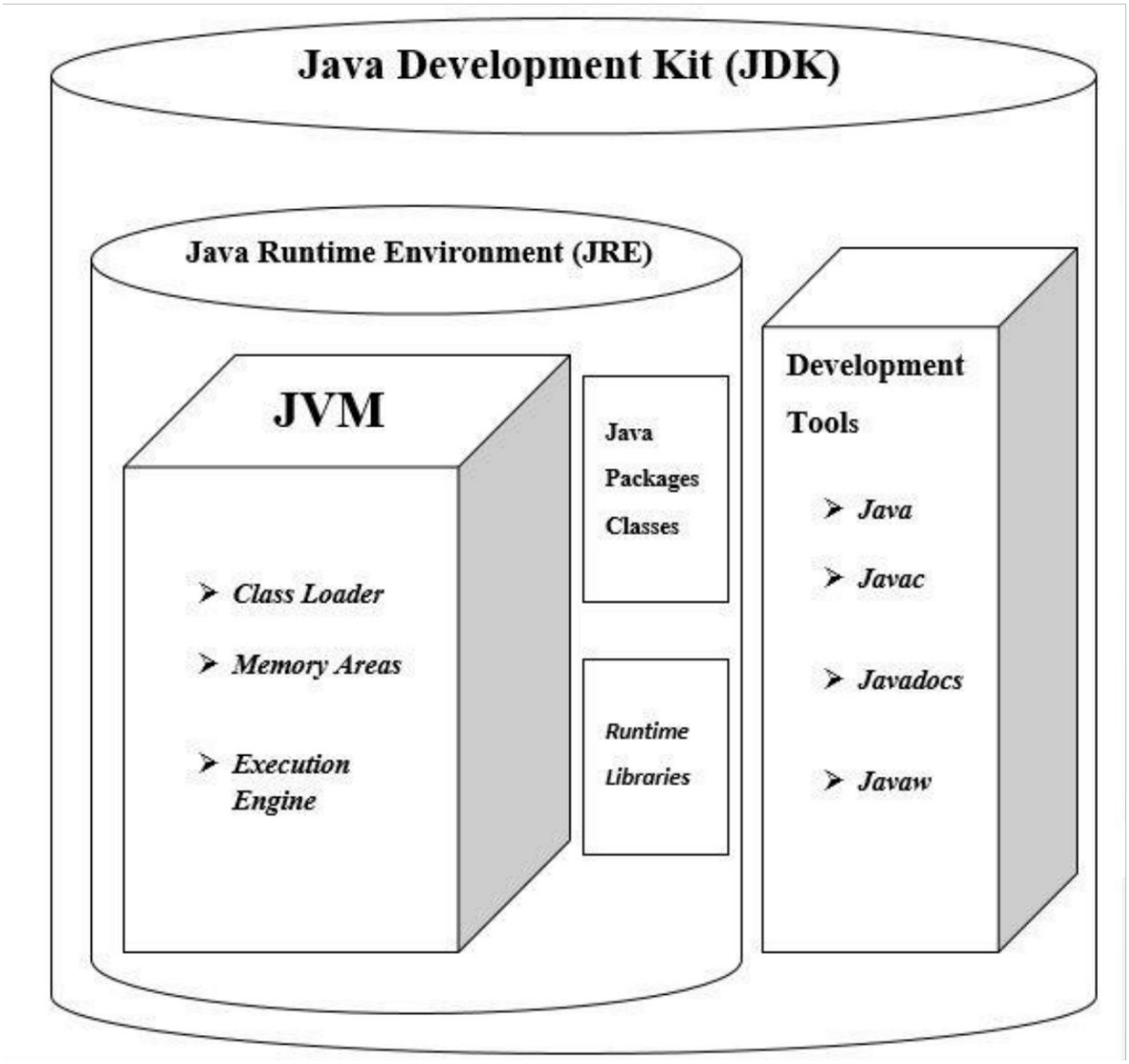

Figure: Architecture of JDK, JRE and JVM

JDK (java Development Kit)

- Java Developer Kit is a software development environment which contains tools needed to develop the Java programs, and JRE to run the programs.

i.e. JDK = *JRE + Development Tools*

- The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), etc. to complete the development of a Java Application
- JDK is mainly targeted for java development (i.e. you can create a Java file (with the help of Java packages), compile a Java file and run a java file).

- JDK tools divided into five categories:
  - Basic Tools
  - Remote Method Invocation (RMI) Tools
  - Internationalization Tools
  - Security Tools
  - Java IDL Tools

Diagram

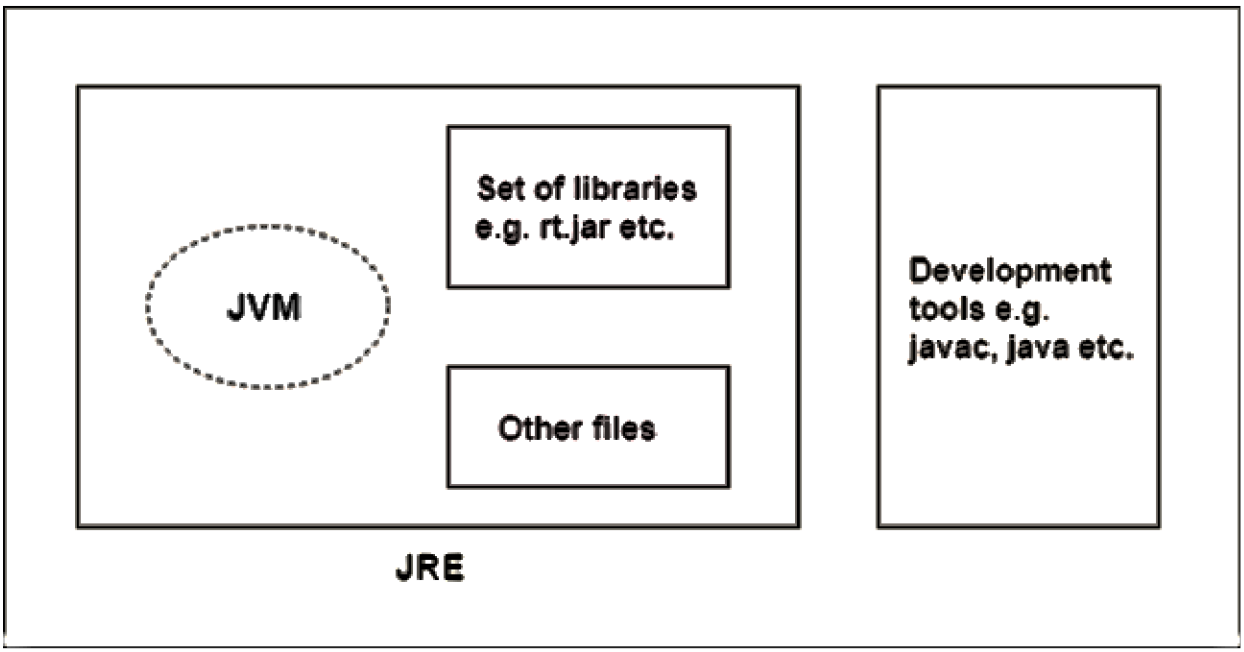

Figure: Java Development Kit (JDK)

JVM (Java Virtual Machine)

- JVM is the main component of Java architecture, and it is the part of the JRE (Java Runtime Environment).
- JVM is an engine which provides runtime environment to drive the Java Code or applications.
- It converts Java bytecode into machines language.
- In other programming languages, the compiler produces machine code for a particular system. But in Java compiler produces code for a Virtual Machine known as JVM.
- First, Java code is compiled then bytecode is generated. This bytecode gets interpreted on different machines and which makes it Platform/Operating system independent
- Bytecode is an intermediary language between java source and host systems.
- As different computers with the different operating system have their JVM, when we submit a ***.class*** file to any operating system, JVM interprets the bytecode into machine level language.

The JVM performs following operation:

- ✓ Loads code (reading code)
- ✓ Verifies code
- ✓ Executes code (linking code to library)
- ✓ Provides runtime environment

JVM provides definitions for the:

- ✓ Memory area
- ✓ Class file format
- ✓ Register set
- ✓ Garbage-collected heap
- ✓ Fatal error reporting etc.

Diagram

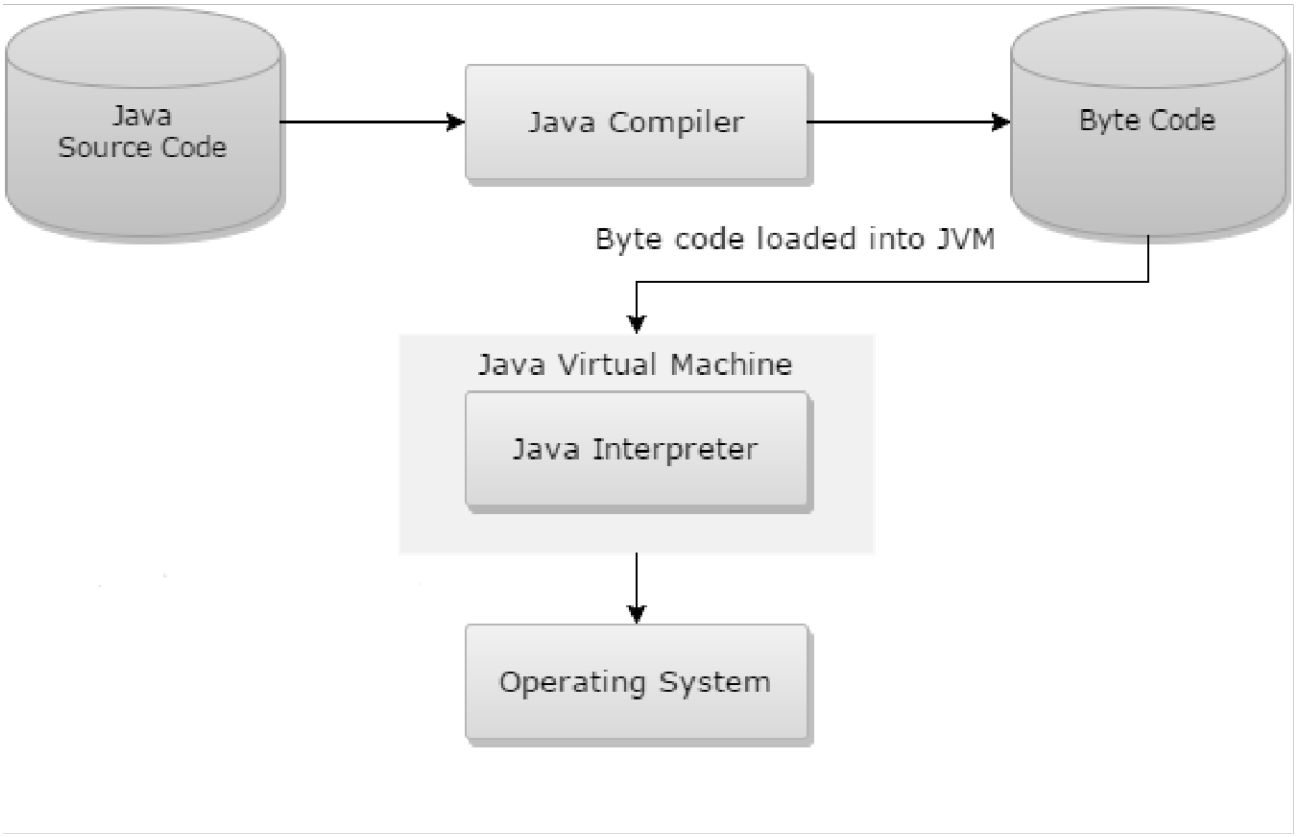

Figure: Java Virtual Machine (JVM)

JRE (Java Runtime Environment)

- It is also written as Java RTE.
- The Java Runtime Environment is a set of software tools which are used for developing Java applications.
- It is used to provide the runtime environment. It is the implementation of JVM.
- It physically exists.
- It contains JVM, class libraries and other supporting files but it does not contains any development tools such as compiler, debugger etc.

  i.e. **JRE** = JVM + Java Package Classes + Runtime Libraries

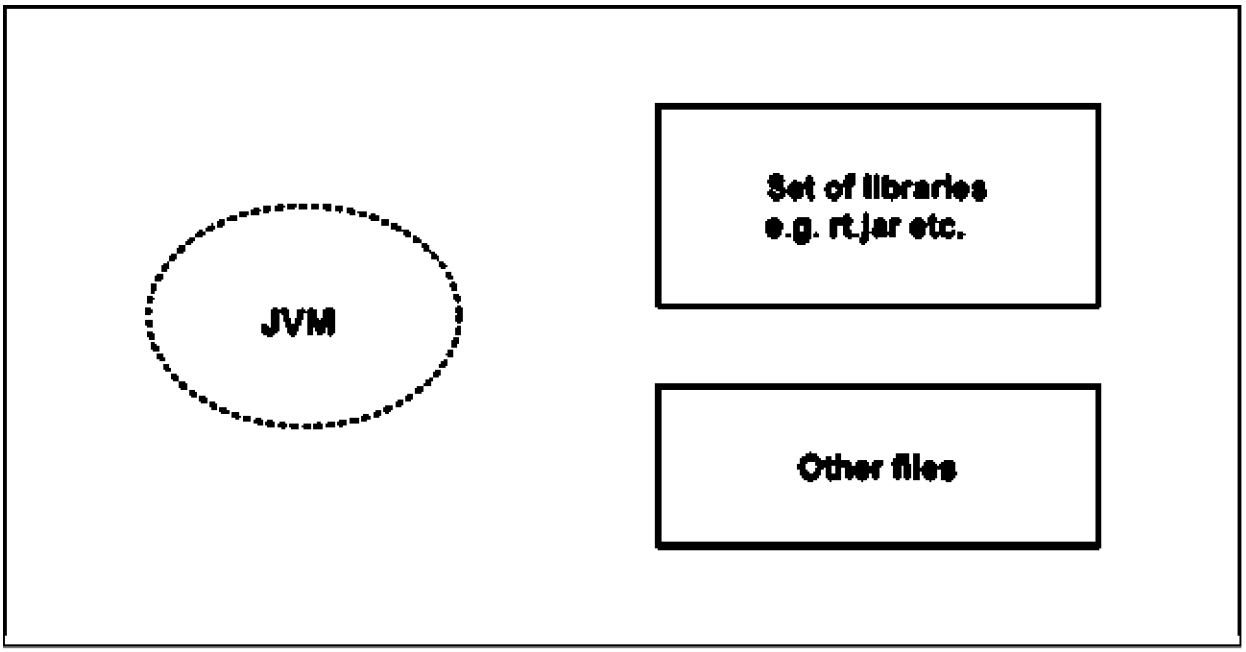

Figure: Java Runtime Environment (JRE)

Difference between JRE, JDK and JVM

Following are few key differences between JRE, JDK and JVM:

i) JRE and JDK come as installer while JVM are bundled with them.

ii) JRE only contain environment to execute java program but doesn't contain other tool for compiling java program.

iii) JVM comes along with both JDK and JRE and created when we execute Java program by giving "java" command