Difference between JDK, JRE, and JVM

1. JDK (Java development kit)

A JDK or Java Development Kit in full is a software development package that supports developers in building Java applications. It includes a lot of tools for debugging, developing, monitoring, and testing Java programs.

You technically need a JDK to convert your code into some other format understandable by the JRE (which we will cover later) for it to execute. The JDK toolset includes the Java Runtime Environment (JRE), an interpreter (Java), a compiler (javac), an archiver (Jar), a documentation generator (Javadoc), and some other development tools as mentioned in the first paragraph.

2. JRE (Java runtime environment)

JRE or Java Runtime Environment in full is, as its name already states, a software that helps execute Java programs; it comes in handy at runtime. JRE handles the control to the JVM, which converts the Bytecode code to machine code and finally runs the program; all of this happens at runtime. To get Bytecode, the compiler (javac) should compile the Java code to Bytecode, which will then be interpreted to machine code by the JVM (under JRE) for full execution.

3. JVM (Java virtual machine)

JVM or Java Virtual Machine in full is an engine that provides a runtime environment to drive the Java Code or applications. It converts Java bytecode into machine language. JVM is a part of the Java Runtime Environment (JRE). In other programming languages like C, the compiler produces machine code for a particular system. However, the Java compiler produces code for a Virtual Machine known as Java Virtual Machine, which will then handle the rest (convert Bytecode to machine code and execute it depending on operating systems).