**Difference between JDK, JRE and JVM**

**JVM**

JVM (Java Virtual Machine) is a software. It is a specification that provides runtime environment in which java bytecode can be executed. It not physically exists.

JVMs are not same for all hardware and software, for example for window os JVM is different and for Linux VJM is different. JVM, JRE and JDK are platform dependent because configuration of each OS differs. But, Java is platform independent.

**JRE**

The Java Runtime Environment (JRE) is part of the Java Development Kit (JDK). It contains set of libraries and tools for developing java application. The Java Runtime Environment provides the minimum requirements for executing a Java application. It physically exists. It contains set of libraries + other files that JVM uses at runtime.

**JDK**

The Java Development Kit (JDK) is primary components. It physically exists. It is collection of programming tools and JRE, JVM.

JDK (Java Development Kit)

JDK contains everything that will be required to***develop and run*** Java application.

**JRE (Java Run time Environment)**

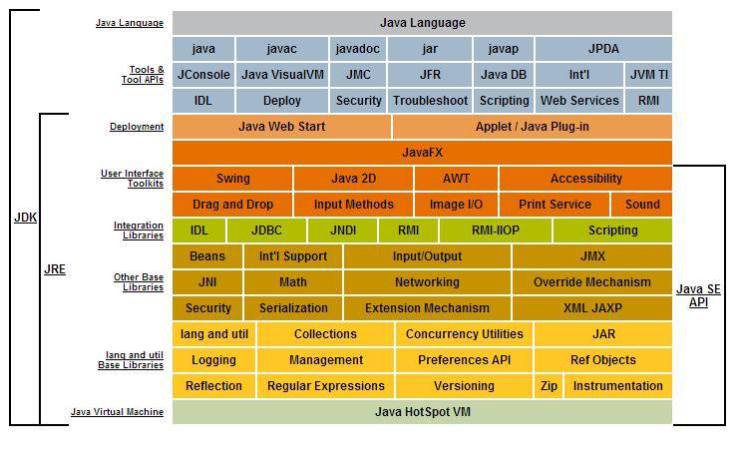
JRE contains everything required to ***run*** Java application which has already been compiled. It doesn’t contain the code library required to develop Java application.

**JVM (Java Virtual Machine)**

JVM is a virtual machine which work on top of your operating system to provide a recommended environment for your compiled Java code. JVM only works with bytecode. Hence you need to compile your Java application(.java) so that it can be converted to bytecode format (also known as the .class file). Which then will be used by JVM to run application. JVM only provide the environment It needs the Java code library to run applications.

**JDK vs JRE vs JVM**

The below chart shows the different features of each of the Java technologies.

**[[](https://i0.wp.com/javabeginnerstutorial.com/wp-content/uploads/2015/07/JDK_JRE_JVM.jpg)](https://i0.wp.com/javabeginnerstutorial.com/wp-content/uploads/2015/07/JDK_JRE_JVM.jpg)**

\*Image Courtesy : Oracle Corporation

Now as per diagram you can identify what is the difference.

JRE = JVM + Required Library to run Application.

JDK = JRE + Required Library to develop Java Application.

Java Portability

In order to understand portability in Java you need to understand what happens to java code from start to finish.

1- Java Source Code (Written by Developer) (Machine Neutral)

2- Compiled Code / Byte Code (Compiled by javac) . (Machine Neutral)

3- Byte Code executed (Executed by JVM) (Machine Specific)

In step 2, javac (Java Compiler) converts Java code to byte code. This can be moved to any machine(Windows / Linux) and executed by JVM. JVM reads the bytecode and generates machine specific code. In order to generate machine specific code JVM needs to be machine specific. So every type of Machine(Windows / Linux / Mac) has a specific JVM. So in this way the coder doesn’t need to bother with generating byte code. JVM takes care of portability. So the final answer is Java is Portable but JVM is Machine Specific.