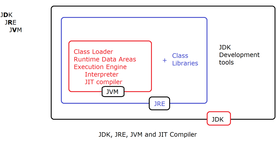
What is the difference between JDK, JRE, JVM and JIT compiler :



JDK, JRE, JVM and JIT Compiler

JDK : [JRE + Development tools]

* It stands for Java Development Kit. It is a developer version so, by using JDK we can **develop** and **execute** java programs.
* In order to develop and execute java programs we have various tools are available

  a) javac : Java Compiler, Used to compile our java source code.

  b) java : Java launcher, Used to execute our .class file OR byte code.

  c) Jdb : Java debugger, Used for degugging purpose.

  d) jconsole : Java Console, Used to display the output in the Console window.

  e) javap : Java profiler, Used to get the complete details of Java classes

  f) javadoc : Java documentation, It is used to generate java documentation

JRE [JVM + class libraries] :

* JRE stands for Java Runtime Environment. It is a client version that means by using JRE we can execute our java program (We cannot develop).

* The JRE folder is not available in java software from JDK 11V so now from Java 11V we can directly execute our java code without compilation by using following command.

java FileName.java

JVM [ClassLoader + Rumtime Data Areas + Execution Engine]

* The main purpose to **load and execute** the .class file.
* It converts the .class file into appropriate operating system format.
* JVM loads the .class file, Verify the correctness of .class file, execute the .class file line by line because internally it contains an interpreter.

What is the difference between Compiler (javac) and Interpreter (JVM contains interpreter) :

Compiler	Interpreter
1) It scans all the code at a time.	1) It scans the code line by line.
2) It displays all the errors and warnings at a time.	2) It will generate the exception in a single line only after that code will not be executed.
3) Debugging is slow.	3) Debugging is fast.
4) After successful compilation, It generates byte code so we need a separate memory to hold this byte code.	4) It will not generate any intermediate file, concurrently it will execute the program line by line.
5) We can delete the .java file after successful compilation.	5) We cannot delete the byte code due to line by line execution
6) C, C++, Java, C# and so on langauges are using compiler	6) Java, Python, JavaScript and so on languages are using interpreter

JIT Compiler :

* It stands for Just in time compiler.
* JVM has interpreter which executes the .class file line by line, It is slow in nature because If we make a mistake at line number 5 then after resoving the problem again It will start the execution from line number 1.
* In order to improve the **execution performance** of java program, We introduced JIT compiler.
* JIT compiler holds **native code instruction** as well as **repeated code instruction** and make it available to the interpreter directly at the time of execution so the prformance of java programs will be increased.

What is the difference between statically typed language and Dynamically typed language ?

Statically Typed Language :

```
                boolean x = true; [For Initialization, Data type is compulsory]
```

* If we want to initialize the variable then **data type is compulsory**, such lanaguegs are known as Statically typed langauge.

  Example of Statically typed langauge : C, C++, JSE, C# and so on

* Once we initialize the variable with a pritcular data type then till the end of the program during the execution, the variable will hold same type of value.

```
        int x = 100;
            x = true; //Invalid
            x = "NIT"; //Invalid
```

Dynamically Typed Language :

* If data type is not compulsory to initialize a variable, such languages are called Dynamically typed languages.

  Example of Dynamically type langauge : Visual Basic, Python, JavaScript and so on.

* In Dynamically typed langauge we can assign all different types of values in the variable.

```
        x = 10     //x is Number type
        x = "NIT"  // x is String type
```