JVM is an implementation of the Java Virtual Machine Specification.

Interprets and compiled [Java](https://www.theserverside.com/definition/Java) binary code (called [bytecode](https://whatis.techtarget.com/definition/bytecode)) for a computer's [processor](https://whatis.techtarget.com/definition/processor) so that it can perform a Java program's [instruction](https://whatis.techtarget.com/definition/instruction)s.

Java programs could be run on any [platform](https://searchservervirtualization.techtarget.com/definition/platform) without having to be rewritten or recompiled by the programmer for each separate platform.

A Java virtual machine makes this possible because it is aware of the specific instruction lengths and other particularities of the platform.

JRE   
It is a software package that contains what is required to run a Java program.

It includes JVM implementation together with an implementation of the [Java Class Library](https://en.wikipedia.org/wiki/Java_Class_Library).

[Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation) distributes a Java Runtime environment with their Java Virtual Machine

called [HotSpot](https://en.wikipedia.org/wiki/HotSpot" \o "HotSpot).

JDK

It is a superset of a JRE and contains tools for Java programmers, e.g. a [javac](https://en.wikipedia.org/wiki/Javac" \o "Javac) compiler.

Java Heap:

When JVM starts it creats HEAP. It is the memory space being used by application that is running to put the objects and JRE classes.

Java Stack memory: (used for currently method running)

Stack is temporary memory to store method variables. Once the method execution is over the memory is getting clear. The values are saved in block of memory and accessable only to the same method

Difference between Heap and Stack :

1. Heap is used by application but stack is used by current method execution
2. Heap contains all objects created in program but stack contains certain reference to them
3. Object stored in Heap can be accessed by all program but primitive local accessed only stack
4. Access mechanism in stack is first-in last out
5. Stack is faster but temporary Heap is available as long as application is running

Heap dump memory log

We can set the JVM parameter to change the location

-XX:HeapDumpPath=/path/to/dumps/directory/java\_pid<pid>.hprof`

**JVisualVM**

We can use JVisualVM to connect to the remote machine and view the CPU utilization, memory sampling, threads, etc. You can also generate a thread dump and memory dump on the remote machine from JVisualVM when connected through the JMX Remote port.

<https://www.baeldung.com/java-heap-dump-capture>

**Class Loader hierarchy**

Class loaders are responsible to load Java classes in to JVM at run time

They are part of JRE. When JVM request a class then class loaders trying to locate the class and load it in to runtime using fully qualified class name

Java classes are not loaded in to memory all at once. They are loaded when needed by application

**ClassLoader Types**

1. **System class loader** 
   1. **loads our own files in the classpath. ( a jars in our classPath environment) this is developers job**
2. **Extension class loaders**
   1. **Loads classes that are extension of the java core classes**
3. **Bootstrap class loader** 
   1. Parent of all the Java classes java.lang