JVM Architecture

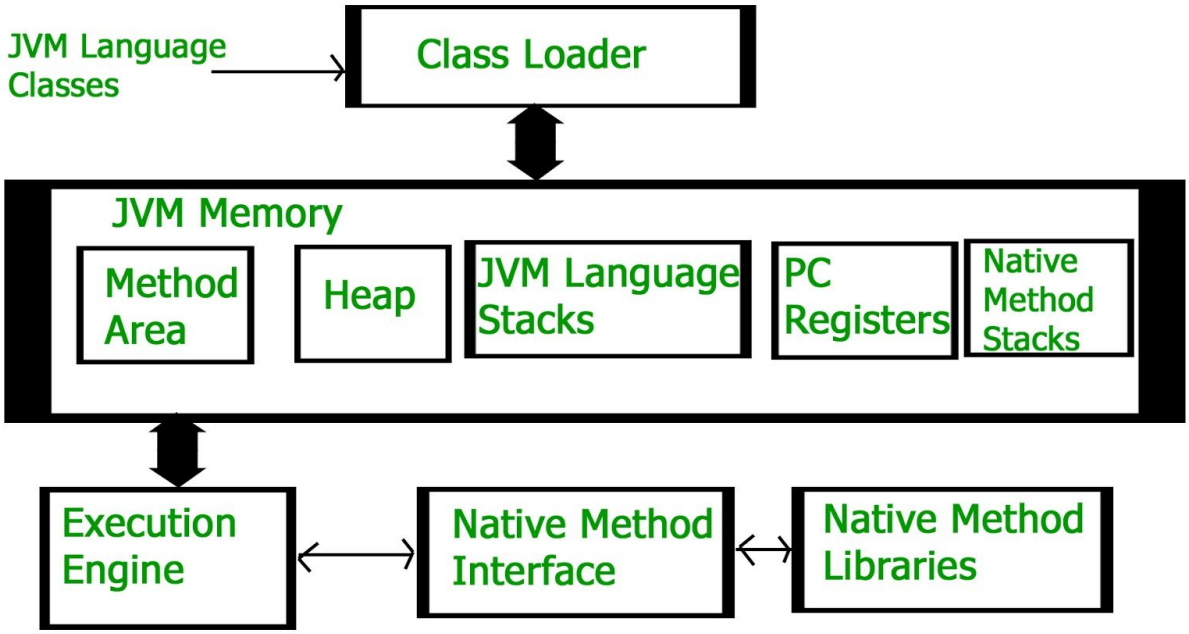
https://www.geeksforgeeks.org/jvm-works-jvm-architecture/

JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides a runtime environment in which Java bytecode can be executed.It can also run those programs which are written in other languages and compiled to Java bytecode. JVM, JRE, and JDK are platform dependent because the configuration of each OS is different from each other. However, Java is platform independent. JVM is one that actually calls the main method present in the java code.

When we compile a .java file then it generate a .class file. This file goes into various steps when we run it. These steps together described in whole JVM

JVM mainly divided into 3 main subsystems

1.Class loader subsystem .
2.JVM memory.
3.Execution engine.



1.Class loader subsystem

It mainly responsible for 3 activities

Loading : The class loader reads the .class file and generate the corresponding binary data and save it in method area.After loading the .class file JVM creates an object of type class to represent this class in heap memory

Linking : performs verification,preparation and resolution

- Verification checks the correctness of .class file
- In preparation JVM allocates the memory for class variables and initializing the memory to default values

- Resolution is a process of replacing symbolic references from the type with direct references. It is done by searching into method area to locate the referenced entity

Initialization : All the static variables are assigned with their values defined in the code and static block. This is executed from top to bottom in class and from parent to child hierarchy.

Generally there are three class loaders

- Bootstrap class loader
- Extension class loader
- System/Application class loader

2.JVM Memory

Method area : In method area all class level information class name, immediate parent class name,methods and variables etc stored. It includes static variables also.

Heap area : Information of all objects is stored in Heap area.there is also one heap area per JVM like method area

Stack area : For every thread JVM creates one run-time stack which is stored here.Every block of stack is called activation record/stack frame which store method calls all local variables of that methods are stored in corresponding frame. After a thread terminate, its run-time stack will be destroyed by JVM

PC Registers : Stores the address of current execution instruction of a thread.Each thread has a separate PC registers

Native method stack : For every thread a separate native stack is created.It stores native method information

3.Execution engine

Execution engine execute the .class file.It reades the byte-code line by line

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 a set of libraries + other files that JVM uses at runtime.

JDK

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 a set of libraries + other files that JVM uses at runtime.

Features of Java

Simple : Java is easy to learn and its syntax is simple. Java removed many complicated and rarely-used features

Ex: pointers and operator overloading

Object-Oriented : java is purely object oriented.Everything in Java is an object. There are many features of oop

object,class,encapsulation,polymorphism,abstraction and inheritance.

Secured : java is a secured programming language. Java is secured because "no explicit pointers and java program run inside a virtual machine sandbox".

Portable : Java is portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.Any machine with java runtime environment can run java programs.

Multi-threading : A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for Web applications, etc.

Platform independent : Java is guaranteed to be write once and run anywhere. On compilation java program compiled into bytecode. This bytecode is platform independent and executed on any machine.

Robust : It uses strong memory management.There is a lack of pointers that avoids security problems.There is automatic garbage collection in java which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.There are exception handling and the type checking mechanism in Java. All these make Java robust.

JAVA Bin files

JAVA : To launch the java application

Javac : To compile java into byte-code

Javadoc : To generate the HTML pages of API documentation from java source file

Javah : generate C header and source file from a java class