

Java Virtual machine

Unidad II Programación Visual



6 de abril de 2019

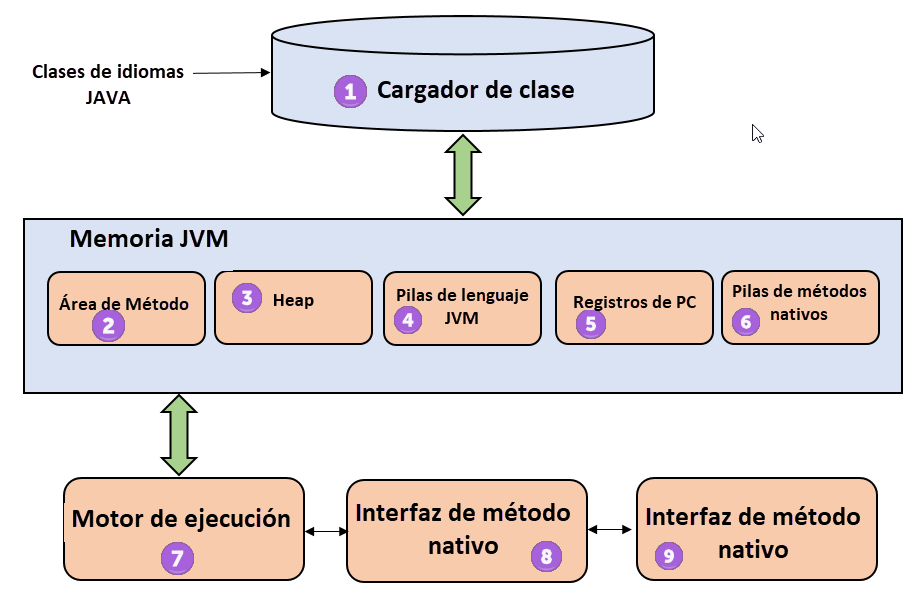
Juan Mauricio garcía meza

Investigación 3

JVM When we compile a program in Java, the compiler turns it into a bytecode language. Java bytecode creates order from all this chaos. Java bytecode is something like the code in Listings 2-3 and 2-4, but Java bytecode isn’t specific to one kind of processor or to one operating system. Instead, a set of Java bytecode instructions runs on any computer.

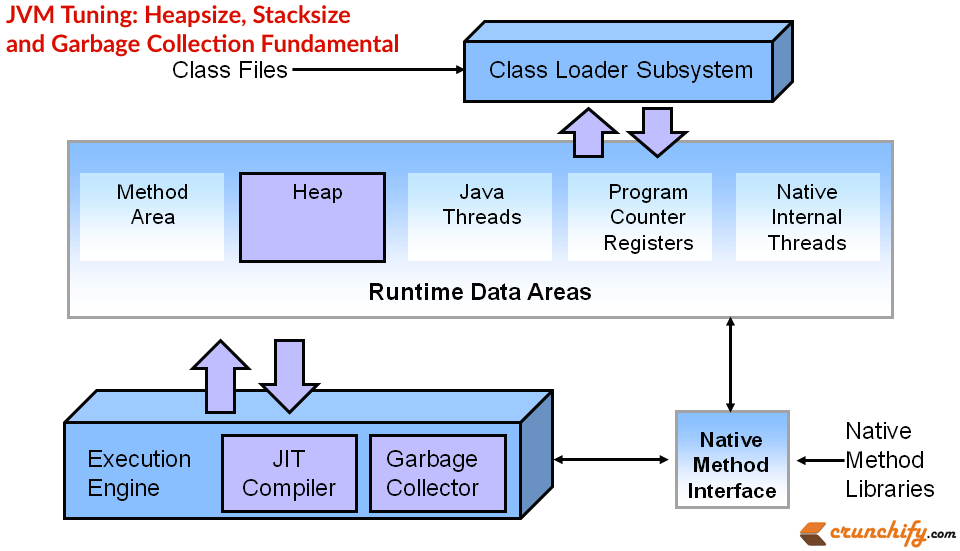
JVM is a engine that provides runtime environment to drive the Java Code or applications. It converts Java bytecode into machines language. JVM is a part of JRE(Java Run Environment).

It stands for Java Virtual Machine JVM has something called classloader, The class loader is a subsystem used for loading class files. It performs three major functions viz. Loading, Linking, and Initialization.



JVM has a Method Area that stores class structures like metadata, the constant runtime pool, and the code for methods. But where we store all the objects and Related instance? The Heap stores All the Objects, their related instance variables, and arrays are stored in the heap. This memory is common and shared across multiple threads.

Finally in order to execute a program, the JVM will need to edit, compile, a linker, load and execute. To combine edit and use different types of instances to work with a lot of other programs. For all this JVM has the Executing Machine, is a type of software used to test hardware, software, or complete systems.



The test execution engine never carries any information about the tested product. And finally to connect every aspect of the JVM with the Java Source Code, there's something call The Native Method Interface, is a framework. That allows Java code which is running in a JVM to call by libraries and native applications.