

definitions

A **Java Virtual Machine (JVM)** is an abstract computing machine that enables a computer to run a Java program.

Java Runtime Environment (JRE) is a software package that contains what is required to run a Java program.

The **class loader** is a part of the JRE that dynamically loads Java classes into the JVM.

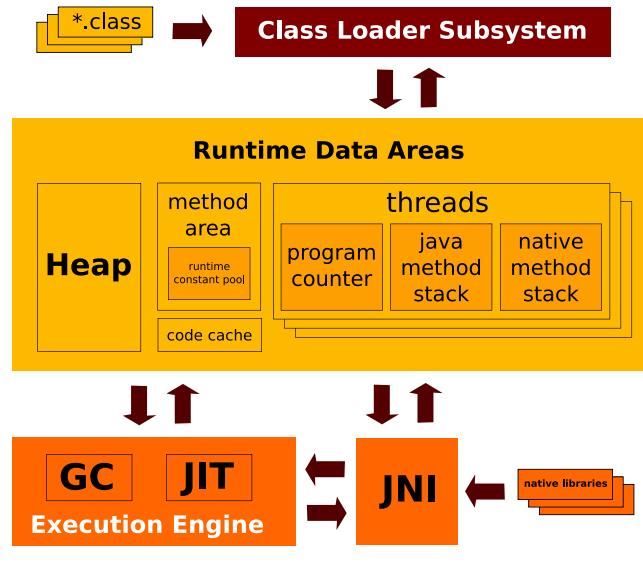
The JVM defines various **runtime data areas** that are used during execution of a program.

The **heap** is the run-time data area from which memory for all class instances and arrays is allocated. Heap storage for objects is reclaimed by an automatic storage management system (known as a **garbage collector**).

The **method area** is the storage area for compiled bytecode.

Each JVM **thread** has a private **stack**, created at the same time as the thread. A stack stores **frames**. A frame is used to store data and partial results, as well as to perform dynamic linking, return values for methods, and dispatch exceptions. Each thread has its own **program counter** register. The PC register contains the address of the JVM instruction currently being executed.

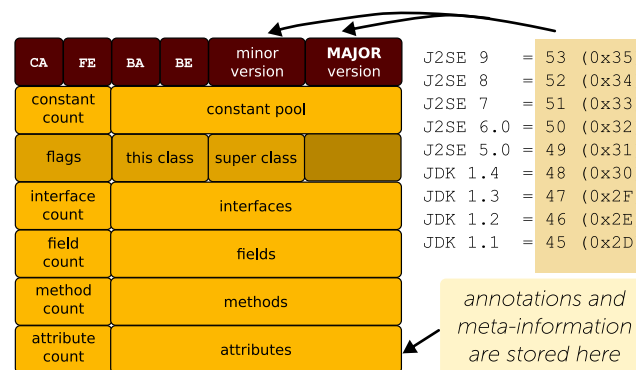
overview



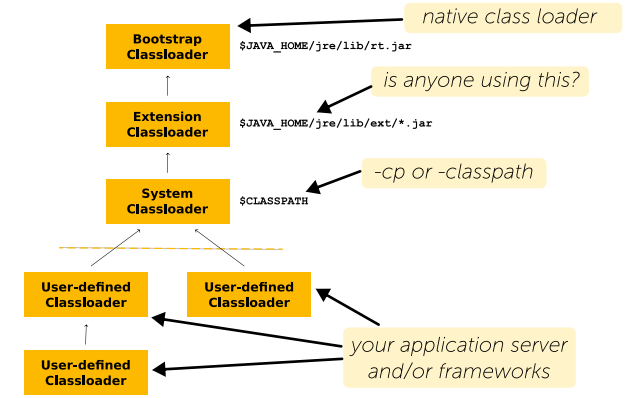
class file

A **class file** is a precisely defined binary format for Java programs. Each class file represents a complete description of **one** class or interface.

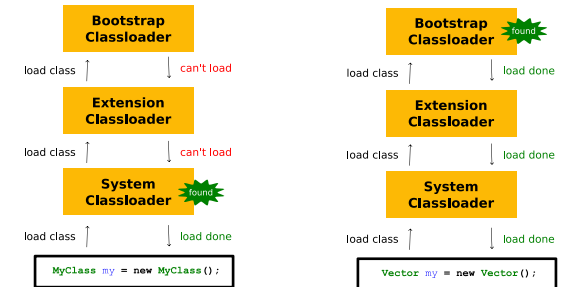
Class file format remains **backward-compatible**.



class loaders



Delegation principle:



Visibility principle:

