Java

- Key Concepts
- The Platform

Key concepts

1. Compile once, run anywhere

2. Automatic memory management

3. Security

Compile once, run anywhere

The problem: how to run application anywhere?

Solutions:

- Distribute sources, compile by user

- Distribute sources, interpret by interpreter

- Compile to an intermediate code and interpret this code

Automatic memory management

The Garbage Collector is a part of JVM.

It attempts to reclaim garbage, or memory occupied by objects that are no longer in use by the program.

Pluses

- One do not need to manage memory manually

- Memory leaks is less possible

- Code is more simple

Minuses

- One can't deallocate memory manually

- Stop-the-world pauses

- Performance

Security

The application launched in the JMV can do only that allowed by the JVM.

The Platform

- JVM

- Bytecode

- JRE

- JDK

- JIT

- Java Editions

JVM

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

- specification,

- implementation,

- instance.

Bytecode

Java bytecode is the instruction set of the Java virtual machine.

Each bytecode is composed of one, or in some cases two bytes that represent the instruction, along with zero or more passing parameters.

JRE and JDK

Java Runtime Environment (JRE)

- Java Virtual Machine(JVM)

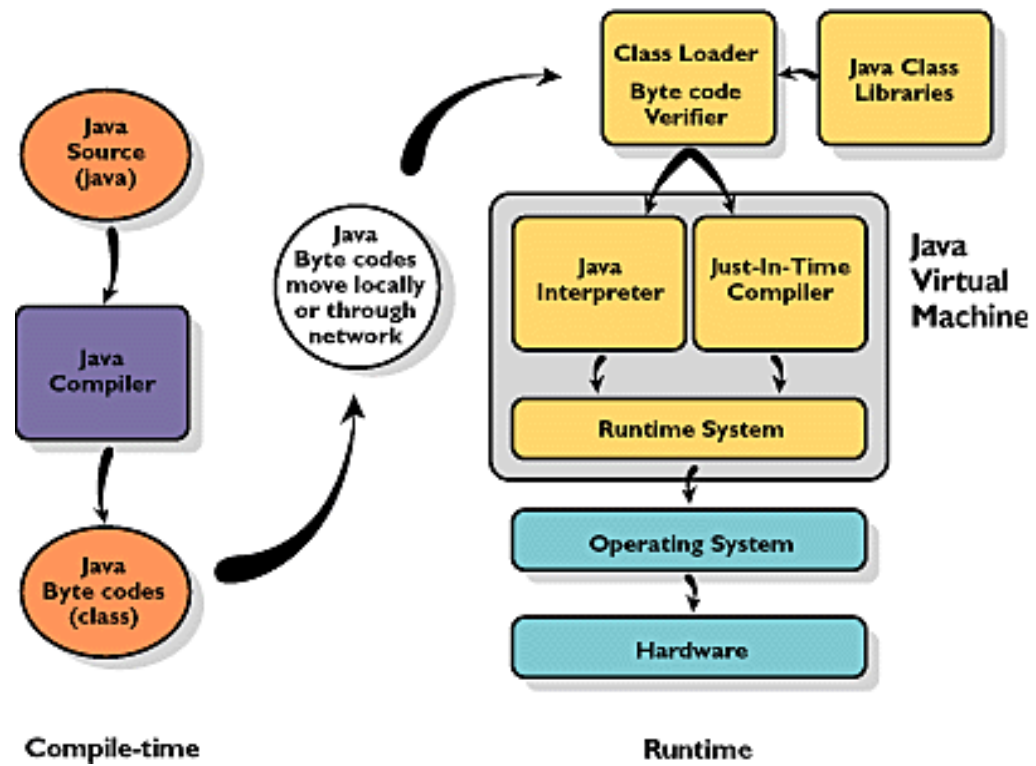- Class Libraries

Java Development Kit

- JRE

- compiler

JIT

Just-in-time (JIT) compilation

A dynamic translation or compilation which is done during execution of a program – at run time.

Java Development and Runtime Environment

Compile-time

Runtime

Java Editions

Java SE – Standard Edition

Java EE – Enterprise Edition

The Java platform (Enterprise Edition) differs from the Java Standard Edition Platform (Java SE) in that it adds libraries which provide functionality to deploy fault-tolerant, distributed, multi-tier Java software, based largely on modular components running on an application server.

Java ME – Micro Edition

The platform for developing applications for mobile devices and embedded systems