Core java

By:
Mr.Ganesh Kr. Yadav
Department of Computer Science & Engineering,
ABESIT College of Engineering, Ghaziabad.

Agenda

1. Introduction of JAVA?
2. Features of Java
3. Brief Concept
   - -JDK
   - -JRE
   - -JVM.

Project Execution Traditional Approach

Programming Language & a Platform

History Of Java?

Java Version?

Where it is used?

Features of java??

What comes to your mind ???

Java Runtime Environment

Java OOP Concepts?

Java New features

Java Virtual Machine

Java Development Kit

What is Java?

- **Java** is a **Programming language** and a **Platform**
- Any hardware or software environment in which a program runs, known as a platform. Since Java has its own Runtime Environment (JRE) and API, it is called **platform**.
- **Java Version**

1. JDK Alpha and Beta (1995)
2. JDK 1.1 (19th Feb, 1997)
3. J2SE 1.3 (8th May, 2000) (Kestrol)
4. J2SE 5.0 (30th Sep, 2004)(Tiger)
5. **Java SE 7 (28th July, 2011) (Dolphin)**

JDK 1.0 (23rd Jan, 1996) ie Oak.

J2SE 1.2 (8th Dec, 1998) (playground)

J2SE 1.4 (6th Feb, 2002) (Merlin)

Java SE 6 (11th Dec, 2006) (Mustang)

6. recent Java 17 or JDK17 Java SE 17 (14th Sept, 2021)

What it is Used?

1. Desktop Applications such as acrobat reader, media player, antivirus etc.
2. Web Applications such as irctc.co.in..etc.
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System
6. Smart Card
7. Robotics
8. Games etc.

Types of Java Application?

- There are mainly 4 type of applications that can be created using java:
- **1) Standalone Application**

It is also known as desktop application or window-based application. An application that we need to install on every machine such as media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.

- **2) Web Application**

An application that runs on the server side and creates dynamic page, is called web application. Currently, servlet, jsp, struts, jsf etc. technologies are used for creating web applications in java.

Types of Java Application

- **3) Enterprise Application**

  An application that is distributed in nature, such as banking applications etc. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.

- **4) Mobile Application**

  An application that is created for mobile devices. Currently Android and Java ME are used for creating mobile applications.

Brief History of JAVA

- James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991.
- originally designed for small, embedded systems in electronic appliances like set-top boxes.
- initially called Oak and was developed as a part of the Green project
- In 1995, Oak was renamed as "Java". Java is just a name not an acronym.
- originally developed by James Gosling at Sun Microsystems(which is now a subsidiary of Oracle Corporation) and released in 1995.
- **JDK 1.0 released in(January 23, 1996).**

Features of Java

- Simple
- Object-Oriented
- Platform Independent
- Secured
- Robust
- Architecture Neutral
- Portable
- High Performance
- Distributed
- Multi-threaded

Java is Simple??

- Java is simple in the sense that:
- syntax is based on C++ (so easier for programmers to learn it after C++).
- removed many confusing and/or rarely-used features e.g., explicit pointers, operator overloading etc.
- No need to remove unreferenced objects because there is Automatic Garbage Collection in java.

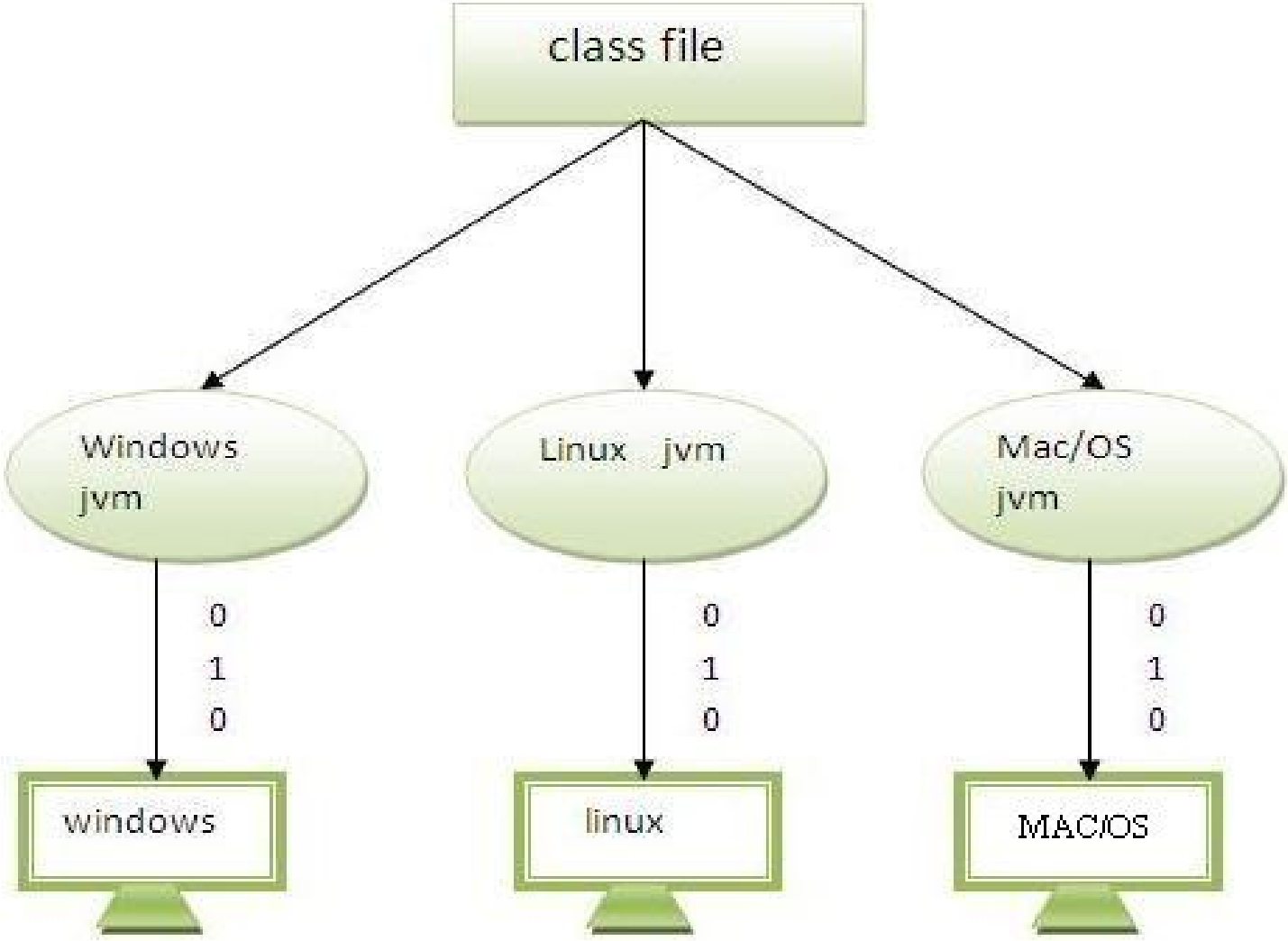
Java is Object Oriented?

- Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behaviour.
- Object-oriented programming(OOPs) is a methodology that simplify software development and maintenace by providing some rulues.
- Basic concepts of OOPs are:
  1. **Object**
  2. **Class**
  3. **Inheritance**
  4. **Polymorphism**
  5. **Abstraction**
  6. **Encapsulation**

Java is Platform Independent?

- A platform is the hardware or software environment in which a program runs. There are two types of platforms software-based and hardware-based. Java provides software-based platform. The Java platform differs from most other platforms in the sense that it's a software-based platform that runs on top of other hardware-based platforms. It has two components:
- **Runtime Environment & API(Application Programming Interface)**
- Java code can be run on multiple platforms e.g. Windows,Linux,Sun Solaris,Mac/OS etc.
- Java code is compiled by the compiler and converted into bytecode.
- This **bytecode** is a platform independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere(WORA).

Platform Independent



Java is Secure

- **Java is Secured because:**
- No explicit pointer
- Programs run inside virtual machine sandbox.
- Class loader- adds security by separating the package for the classes of the local file system from those that are imported from network sources.
- Byte code Verifier- checks the code fragments for illegal code that can violate accesss right to objects.
- Security Manager- determines what resources a class can access such as reading and writing to the local disk.

Some sucurity can also be provided by through SSL, JAAS, cryptography etc.

Java is Robust

Java is Robust because -

- Robust simply means strong. Java uses strong memory management. There are lack of pointers that avoids security problem. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. All these points makes java robust.
- Architecture-neutral.
- There is no implementation dependent features e.g. size of primitive types is set.

Portable High-Performance Distributed

- **Portable**

  We may carry the java byte code to any platform.

- **High-performance**

  Java is faster than traditional interpretation since byte code is "close" to native code still somewhat slower than a compiled language (e.g., C++)

- **Distributed**

  We can create distributed applications in java. RMI and EJB are used for creating distributed applications. We may access files by calling the methods from any machine on the internet.

Multi-Threaded

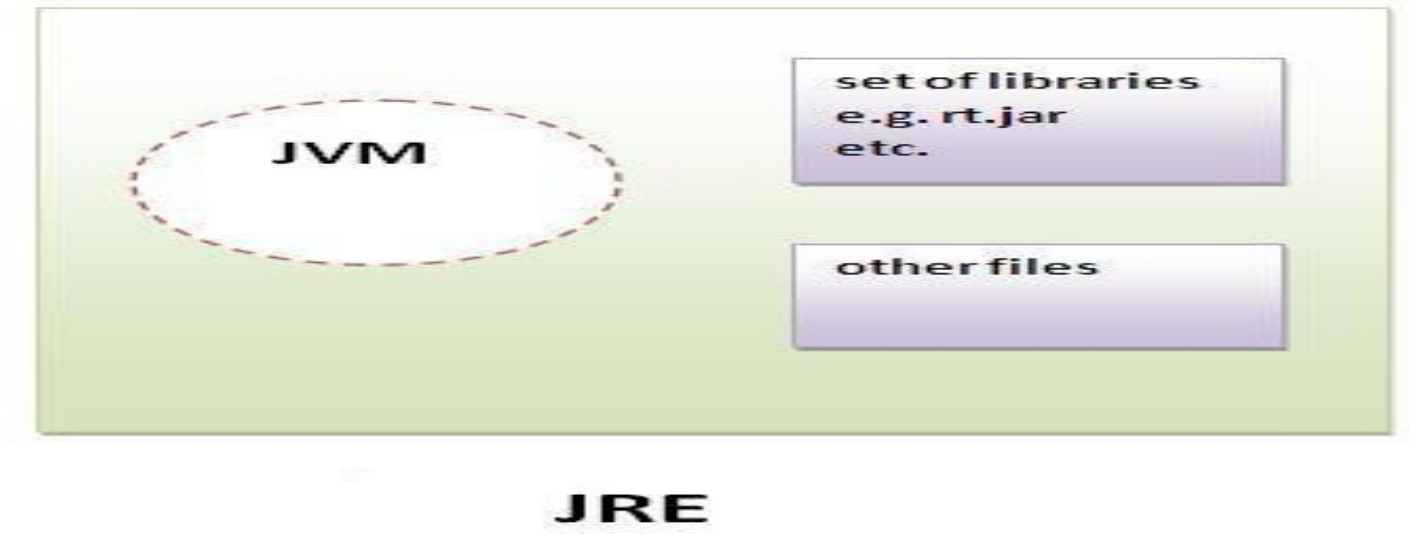
- **Multi-threaded**

  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 shares the same memory. Threads are important for multi-media, Web applications etc.

Introduction of JRE JDK and JVM

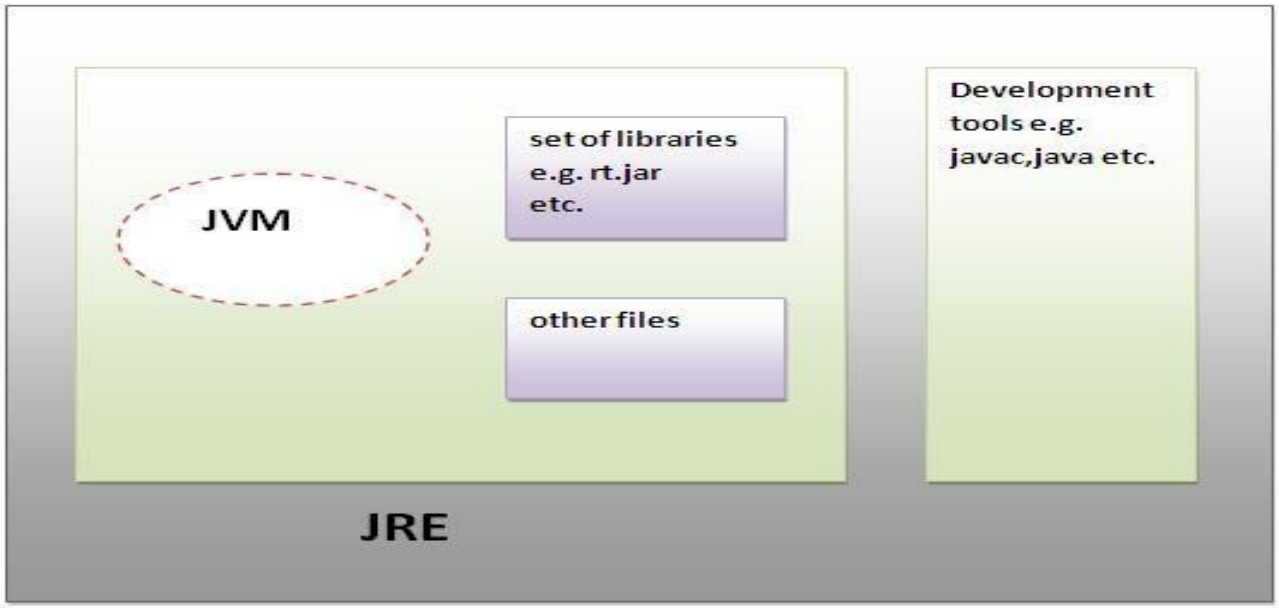
JRE

- JRE is an acronym for Java Runtime Environment. It is used to provide runtime environment. It is the implementation of JVM. It physically Exists. It contains set of libraries + other files that JVM uses at runtime.



Difference between JDK,JRE and JVM

- **JDK**
- JDK is an acronym for Java Development Kit. It physically exists. It contains JRE + development tools.



JVM (Java Virtual Machine)

JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.

- JVMs are available for many hardware and software platforms (i.e.JVM is plateform dependent).

The JVM performs four main tasks:

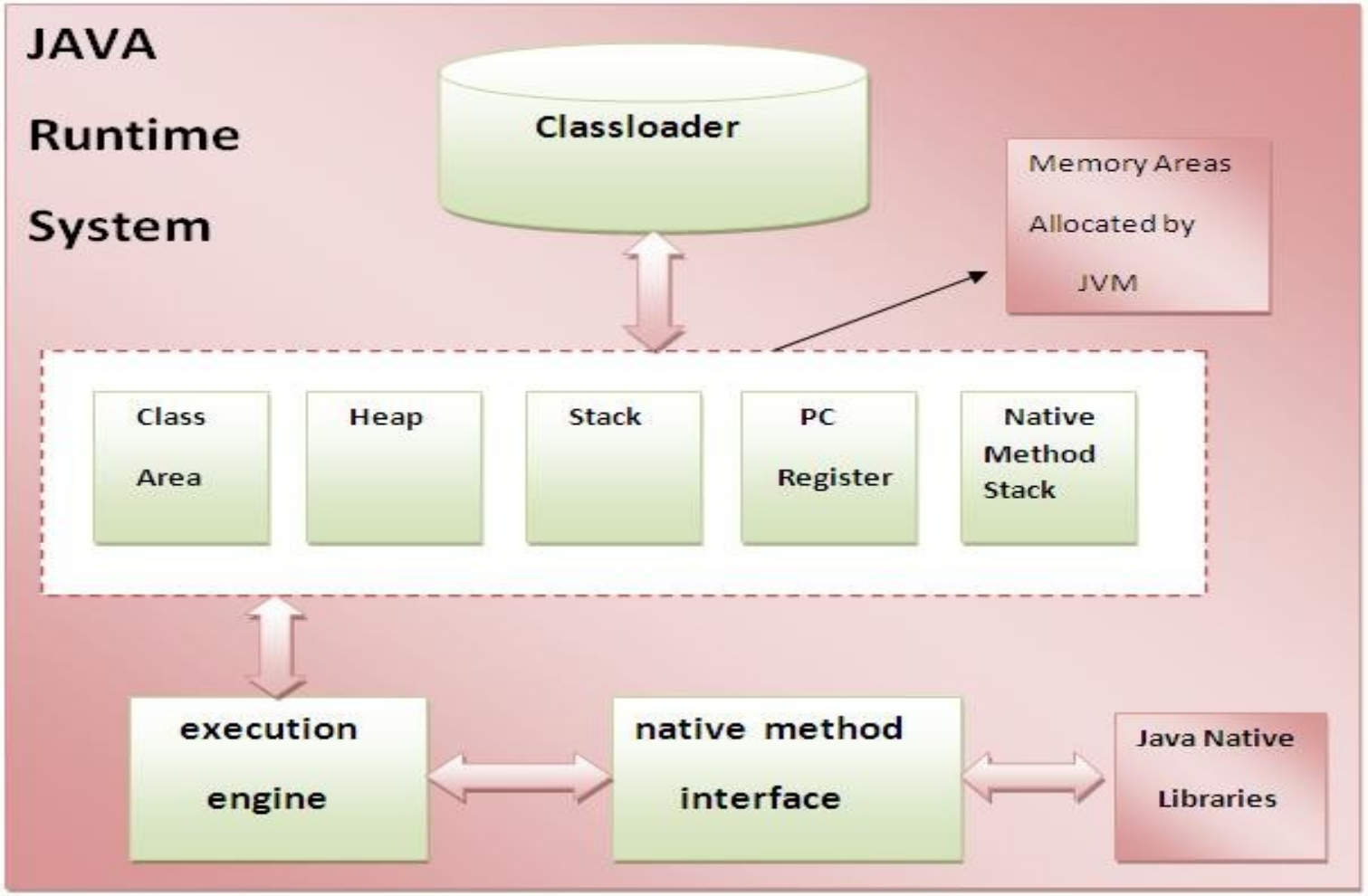
- Loads code
- Verifies code
- Executes code
- Provides runtime environment

JVM (Java Virtual Machine)

JVM provides:

- Memory area
- Class file format
- Register set
- Garbage-collected heap
- Fatal error reporting etc.

Internal Architecture of Java



Internal Architecture of Java

1) **Classloader:** Class loader is a subsystem of JVM it is used to load class files.

2) **Class(Method) Area:** stores per-class structures such as the runtime constant pool, field and method data, the code for methods.

3) **Heap:** It is the runtime data area in which objects are allocated.

4) **Stack:** Java Stack stores frames. It holds local variables and partial results, and plays a part in method invocation and return. Each thread has a private JVM stack, created at the same time as thread.

   A new frame is created each time a method is invoked. A frame is destroyed when its method invocation completes.

5) **Program Counter Register:** PC (program counter) register. It contains the address of the Java virtual machine instruction currently being executed.

Internal Architecture of Java

6) Native Method Stack:

It contains all the native methods used in the application.

7) Execution Engine: It contains:

1) A virtual processor

2) Interpreter: Read byte code stream then execute the instructions.

3) Just-In-Time(JIT) compiler: It is used to improve the performance.JIT compiles parts of the byte code that have similar functionality at the same time, and hence reduces the amount of time needed for compilation. Here the term ?compiler? refers to a translator from the instruction set of a Java virtual machine (JVM) to the instruction set of a specific CPU.

Questions ?