Java[J2SE] Docs 5/12/2018

N Hemanth

CONTENTS :

1. About JAVA
2. Features of JAVA
3. Drawbacks of C and C++ that are drawn to java major advantage
4. JVM Architecture
5. Description for JDk/bin folder files

Java is simple, secure , object-oriented ,class-based general purpose programming language.

**Java Applications:**

Java run under over 3 billion devices world wide.it have many applications.

* Desktop applications
* Web applications
* Enterprise applications
* Mobile
* Embedded System etc

**Types of Java Applications** :

1. Standalone applications : it is used to develop Desktop applications mostly run traditional operating systems ex: music player
2. Web applications : it is used to develop server and run dynamic web pages it uses servlets,JSP,springs,hibernate etc
3. Enterprise application : it is used to develop enterprise or industrial applications for this it will use EJB[Enterprise Java Bean]
4. Mobile application : it used to develop small mobile applications.

**Java Platforms/Editions :**

1. J2SE[Java Standard Edition]
2. J2EE[Java Enterprise Edition]
3. J2ME[Java Micro Edition}
4. JavaFX[for Internet applications]

**Features of Java :**

1. Simple
2. Object-oriented
3. Secured
4. Portable
5. Robust
6. Platform independent
7. Interpreted
8. Architectural neutral
9. High performance
10. Multi-threaded
11. Distributed
12. Dynamic
13. **Simple** :

→ java syntax based on c++.

→ java removed complicated features like explicit pointers ,operator overloading.

→ java has automatic garbage collector which removes automatically unreferenced

Objects.

**2. Object-oriented:**

→ everything in java is object , which is like a blueprint for class.

**3. Security : java is known for its security**

→ no explicit pointers

→ java runs inside a virtual machine

→ class loader : it provides security by separated package from classes.

→ Bytecode verifier : java checks weather bytecode is clean or not

→ java provides security.

**4.Platform independent:**

→ java is portable and platform independent write once and run anywhere.

→ for c and c++ it produce machine based .exe files if this files run under another system

It wont work

→ for java where it carries bytecode to any level and can make it work

**5. Robust :**

→ it had strong memory management

→ there is automatic garbage collector

→ type checking and exception handling.

**6.Portable:**

→ java will allow you to carry bytecode to any level and execute on all machines which

Have JVM

**7. High-Performance :**

→ java is faster than traditional interpreted programming languages but it is still slower

Than c and c++.

**8.Distributed :**

→ java is distributed languages which can obtain buy Rmi and ejb allows to transfer files

Destination

**9. Multi-Threaded :**  java supports thread concept which can all web and server side a

Applications

**10.Architectural neutral :**

→ if it is 32 bit or 64 bit pc java has a sample data typing..

**11. Dynamic :**

→ java supports dynamic class loading it also supports native class loading.

-------------------------------------------------------------------------------------------------------------------------

**Drawbacks of c and c++:**

* C and C++ are platform dependent.while java is platform dependent.
* They are used mainly for system programming but java supports wide range of applications including internet.
* C and C++ has no bound checking
* Security issues were high while compared to java
* Complexity of code increases while writing large software.
* C is not a object oriented and c++ supports but its not true
* Both uses pointers which leads to security fixes

--------------------------------------------------------------------------------------------------------------------------

**JVM Architecture :**

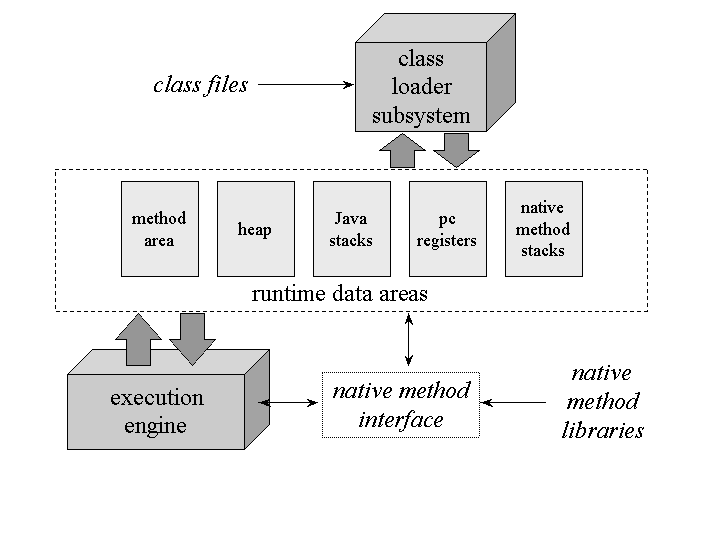
Jvm is a abstract machine where bytecode is executed. Jvm is available on various hardware and software platforms

**Implementation :**

jvm implements is known as JRE[java runtime environment].

**RuntimeInstance :**

to run the java class every time a jvm instance created.



**1.Class loader subsystem:**

class loader subsystem under jvm loads .class file into it it had mainly three types.

**Bootstrapclassloader :**

bootstrapclassloader is super class of extension class loader . it loads rt.jar files which contain all apis of j2se like java.lang,java.util,java.math etc.

**Extension class loader :**

extension class loader is child of bootstrap class and parent for system/application class loader which loads jar files from $java\_Home/jre/lib/ext directory.

**System/Application class loader :**

it is a child class of extension class loader it loads all class files under class path. When every programs executed it point to present working directory.

**2.class or method area :**

Class or method area stores class structures such as runtime constant pool method data, fields and code fro methods.

**3.Heap:**

It is runtime area where objects were allocated.

**4.stack :**

Stack store frames , it holds local variables and partial result , it is also play key role in method invocation and return.each thread has a private jvm stack created at the same of creation of thread.a new frame is created when every time a method invoke and destroy after its completion .

**5.PC Registers :**

Pc registers holds addressed of currently executing on jvm .

**6. Native stack methods :**

It all contains all native methods used in application .

**7.Execution engine :**

execution engine where programs executes it contain mainly three parts

1. **Virtual processor**
2. **Interpretation :**

where it reads bytecode streams and executes it

1. **JIT-Compiler:**

it transforms instruction of jvm into cpu instruction, also it reduce the code which have similar functionality

**8.Java native Interfaces :**

Java native interface is a framework where it helps to communicate with other programming languages , it uses OS libraries from console input and Output.

-------------------------------------------------------------------------------------------------------------

**JAVA/BIN FILES:**

**Basic Tools:**

AppletViewer:

Applet viewer is used to use applets outside of browser

Extcheck:

Extcheck is used for solve the jar files version conflicts.

Jar:

Create and manage java and archive files[jar]files.

Apt:

Apt is for annotation processing tool

Java:

Launcher for java application , a single launcher for both development and deployment is used for an old .jre is no longer exist

Javac :

Compiler for java programming language

Javadoc:

API documentation generator

Javah:

C header and stub generator , used to write native methods.

Javap:

Class file disassembler

Jdb:

The java debugger.

**Security tools:**

These security tools help you set security policies on your system and create applications that can work within the scope of security policies set at remote sites.

Keytool:

Manages keystores and certificates.

Jarsigner:

Generates and verify JAA signatures.

Policytool:

GUI tool for managing policy files.

These security tools help you obtain, list, and manage Kerberos tickets

Kinit:

Tool for obtaining Kerberos v5 tickets .

Klist:

Command line tool to list entries in credential cache and key tab

Ktab:

Command line tool to help user manage entries in key user table.

**Internalization Tools:**

This tool helps to create localizable applications.

Native2ascii:

Convert text to unicode latin-1

**Remote Method Invocation [RMI] Tools :**

These tools help to create applications that interact over the Web or other network**.**

Rmic :

Generate stubs and skeletons for remote objects.

Rmiregistry:

Remote object registry service

Rmid:

RMI activation system daemon

Serialver:

Return class serialVersionUID.

JAVA IDL and RMI-IIOP Tools :

These tools are used when creating applications that use OMG-standard IDL and CORBA/IIOP.

Idlj:

Idlj generates java bindings from IDL files.IDL[interface definition language] is a generic term that lets a program or object to communicate with another program written in another unknown language. To use CORBA functionality

Orbd:

Provides support for clients to transparently locate and invoke persistent objects on servers in the CORBA environment. ORBD is used instead of the Transient Naming Service, tnameserv. ORBD includes both a Transient Naming Service and a Persistent Naming Service. The orbd tool incorporates the functionality of a Server Manager, an Interoperable Naming Service, and a Bootstrap Name Server. When used in conjunction with the servertool, the Server Manager locates, registers, and activates a server when a client wants to access the server.

Servertool:

Provides ease-of-use interface for the application programmers to register, unregister, startup, and shutdown a server.

Java Deployment Tools:

Utilities for use in conjunction with deployment of java applications and applets on the web.

Javafxpackager :

Packages javafx applications for deployment.  
Pack200:

Transforms jar file into compressed pack200 using the java gzip compresser , the compressed pack files are compressed JARs which can be dirctly deployed saving bandwidth and reducing download time.

Unpack200:

Transforms a packed file of pack200 into jar file.

Java Web Start Tools:

It used for conjunction with java web start

Javaws :

Command line tool for java web start and setting various options

Java Troubleshooting,Profiling,Monitoring and Management Tools:

Jcmd :

JVM diagnostic commands tool - sends diagnostic command request to run a java virtual machine.

Jconsole:

A JMX -compliant graphical tool for monitoring java virtua, machine.it can monitor both local and remote jvms it cal also monitor and manage an application .

Jmc:

Java mission control client includes tools to monitor and manage java application without introducing performance overhead normally associated with these tools.

Jvisualvm :

A graphical tool that provides detailed information about the Java technology-based applications (Java applications) while they are running in a Java Virtual Machine. Java VisualVM provides memory and CPU profiling, heap dump analysis, memory leak detection, access to MBeans, and garbage collection

Java Web Services Tool :

Schemagen :

Schema generator for java architecture for xml binding.

Wsgen:

Tool for generate JAX-WS portable artifacts.

Wsimport:

Tool for generate JAX-WS portable artifacts

Xjc :

Binding compiler for java architecture for xml binding.