Intoduction To Java

CS151

Chris Pollett

Sept. 7, 2005.

Outline

- Java Platform
- Java Run-Time Architecture
- Getting started with Java

Overview of Java

- Java 1.0 released in 1996.
- The Java Development Kit and Java Runtime Environment has continued this numbers system.
- After JDK 1.2 came out SUN started using the terminology Java 2 Platform, but the JDK number stayed 1.2.
- Current JDK, JRE should be 1.5, but Sun has decided to number it 5.
- So now we have Java 5 Platform JDK version 5. Ugh!!
- The Java 2 Platform has three editions geared towards different markets:
  - J2EE (Enterprise Edition for servers)
  - J2SE (Standard Edition)
  - J2ME (Micro edition for handhelds)
- First two use the Java Virtual Machine. Last one, sometimes uses a stripped down version of this machine (KVM).

J2SE Components

- Java Virtual Machine to execute Java programs
- Development tools including compiler, debugger, profiler, and documentation generator.
- Core APIs, including language support, logging, Java Beans, XML processing, and native interface.
- GUI APIs:AWT, Swing, Java2D, JavaSound
- Integration APIs, RMI, CORBA, JDBC, JNDI
- Deployment tools: Java Web Start and Java plug-in

J2SE is what we’ll focus on in this course.

J2EE Components

J2EE is built on top of J2SE (uses JVM)

- Java Servlets and Java Server Pages
- Enterprise Java Beans
- E-mail and messaging services
- Transaction Management

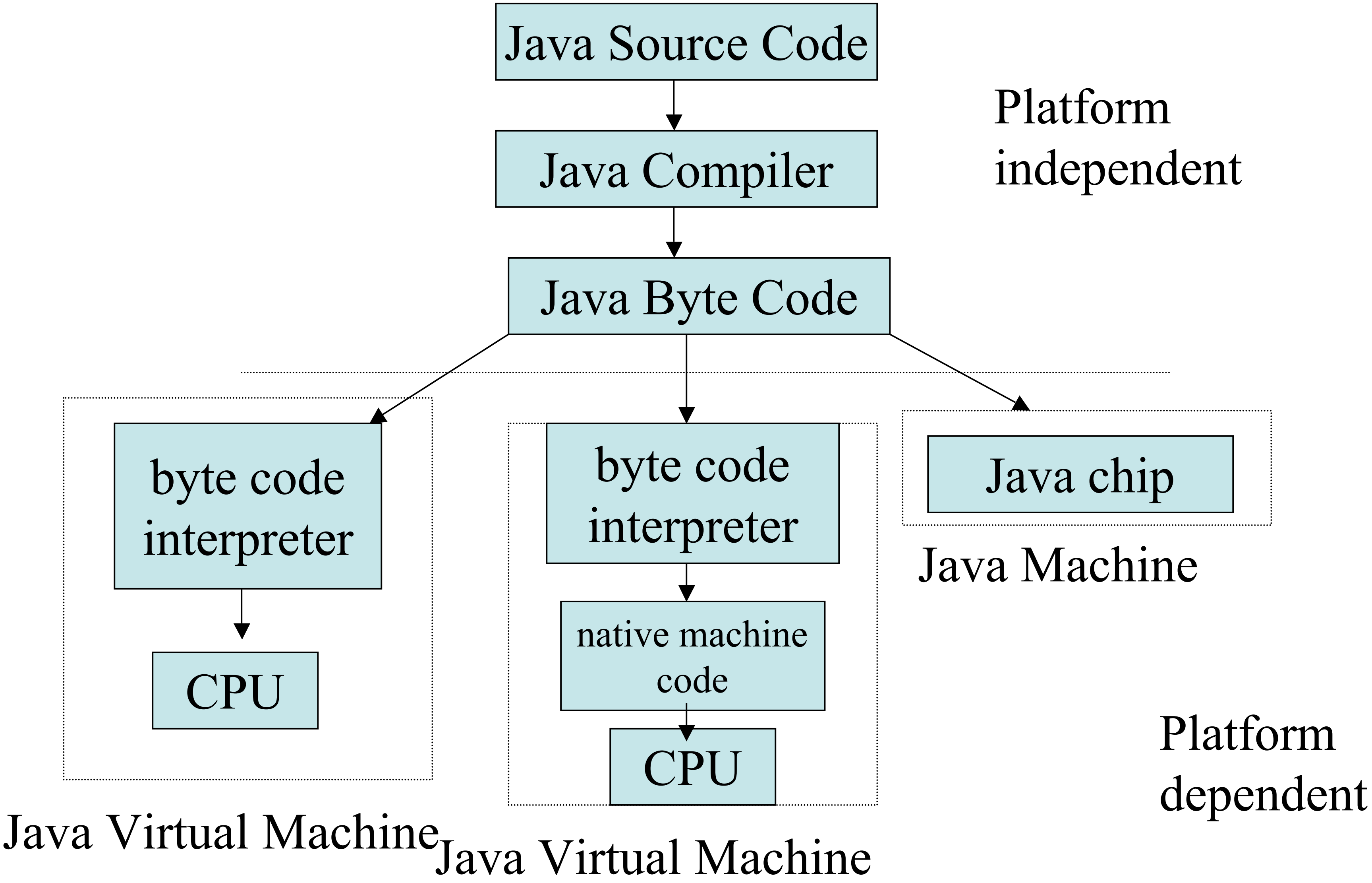
J2ME

- For cell-phone, handhelds, set-top boxes.
- Has two configurations depending on how limited the device is:
  - Connected Device Configuration (CDC) - for high end consumer electronics with megabytes of memory.
  - Connected Limited Device Configuration (CLDC)-for PDAs or Cell-phone, ect with kilobytes of memory.
- J2ME also has various different profiles that one can choose or choose not to support depending what particular kind of need your small device has. Ex: Personal Profile. Mobile Information Device Profile.

Java Run-Time Architecture

- Java is designed to work on different machines, to allow for networked connections, to be secure, and to be efficient.
- The next several slides will try to indicate how these goals were achieved.

Execution Models



Java Virtual Machine

- Java Programs are executed in two stages:
  - Stage 1: We compile source code to byte-code. This byte code is close to machine codes of many processors.
  - Stage 2: Then we execute this byte code depending on the kind of setting we are in. (Three routes of previous slide.) Usually, Stage 2 for us will be executed on a JVM using Just in Time compilation.

Java Byte Code

- These are the instructions which can execute on the Java Virtual machine.
- These instructions are made up of a 1-byte opcode, together with 0 or more operands.
- Operands can vary in length, depending on the opcode.
- The core JVM execute the following loop:

```
do {
fetch opcode;
fetch operands
execute instruction;
}while(no done);
```

JVM Execution

Is very much like a 32-bit (RISC) CPU:

- Has a pc (program counter)
- Has an optop register for the top of the operand stack.
- Has a vars register pointing to the local variables of the current method
- Has a frame register pointing to the execution environment structure.
- Has a garbage collected heap to store objects at run-time.

JVM Instructions Types

- Stack manipulation
- Array management
- Arithmetic and logical operations
- Methods invocation and return
- Exception handling
- Synchronization of thread

Applications and Applets

There are two main types of Java programs:

- applications which are stand-alone and have full access to system resources.
- applets which are embedded in web pages and have restricted access to system resources on the system running the applet.

Security of JVM

- Shielded Memory Addresses -- no way to directly manipulate memory addresses in Java. No way to forge a pointer to an address. Memory de/allocation are done automatically by the JVM.
- Verification of Byte-code: JVM verifies byte code before it executes it. Looks for improper structures and control flows in the byte-code, checks for access-violations, and violations of the type system.
- Run-time Security Manager - is checked against a security policy whenever an applet is about to execute an insecure operation.

Getting Started with Java

- I then showed how to enter and compile a simple hello world application in java.
- I showed some example applets from previous. I showed also the code to embed these applets into web pages, and the source code for these applets.

Basic Structure of Applets

- Must extend the applet class.
- Should have a paint() method
- Each applet gets a rectangular structure to draw into
- paint method allows one to draw into this region
- This method is invoked by the applet context whenever the applet is active and this region become visible.