



A REVIEW ON : JAVA

Shweta Garg, Shrishti Vashist, Shruti Aggarwal

CSE department, Dronacharya College of Engineering,
Gurgaon, India

Abstract:

Java is a high level, object-oriented, platform independent language. Java, unlike some languages before it allows for the use of words and commands instead of just symbols and numbers. Java also allows for the creation of advanced data types called objects which represent real world things like a chair or a computer where you can set the attributes of these objects and things they do. Java is very flexible - it can be used to develop software as well as applets.

Keywords: Vulnerabilities- capable of or susceptible to being hurt, Embedded- to fix into a surrounding mass, Intende- purposed, Compliance- the act of conforming, Native- being the place.

Abbreviation: WORA- write once run anywhere, JVM- java virtual machine, SDK- software development kit, JRE- java runtime environment, GPL- general public license, JDBC- java database connectivity, JNDI- java naming and directory interface, AWT- abstract window toolkit.

1. Introduction

Java is a general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to

run on another. Java applications are typically compiled to bytecode (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2012, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class

For Correspondence:

shivigarg101@gmail.com

Received on: September 2013.

Accepted after revision: September 2013.

Downloaded from: www.johronline.com

libraries were developed by Sun from 1991 and first released in 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).



2. History:-

James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time. The language was initially called *Oak* after an oak tree that stood outside Gosling's office; it went by the name *Green* later, and was later renamed *Java*, from Java coffee, said to be consumed in large quantities by the language's creators. Gosling aimed to implement a virtual machine and a language that had a familiar C/C++ style of notation.



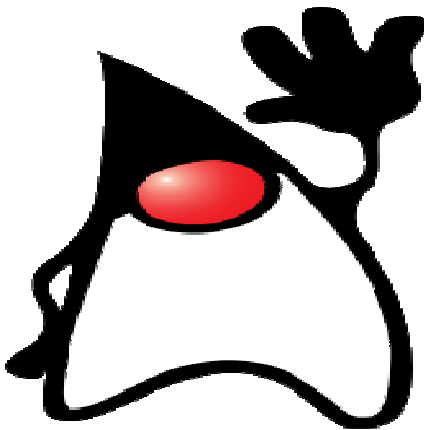
James Gosling, the creator of Java

Sun Microsystems released the first public implementation as Java 1.0 in 1995. It promised "Write Once, Run Anywhere" (WORA), providing no-cost run-times on popular platforms. Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major web browsers soon incorporated the ability to run *Java applets* within web pages, and Java quickly became popular. With the advent of *Java 2* (released initially as J2SE 1.2 in December 1998 – 1999), new versions had multiple configurations built for different types of platforms. For example, *J2EE* targeted enterprise applications and the greatly stripped-down version *J2ME* for mobile applications (Mobile Java). *J2SE* designated the Standard Edition. In 2006, for marketing purposes, Sun renamed new *J2* versions as *Java EE*, *Java ME*, and *Java SE*, respectively. In 1997, Sun Microsystems approached the ISO/IEC JTC1 standards body and later the Ecma International to formalize Java, but it soon withdrew from the process. Java remains a *de facto* standard, controlled through the Java Community Process. At one time, Sun made most of its Java implementations available without charge, despite their proprietary software status. Sun generated revenue from Java through the selling of licenses for specialized products such as the Java Enterprise System. Sun distinguishes between its Software Development Kit (SDK) and Runtime Environment (JRE) (a subset of the SDK); the primary distinction involves the JRE's lack of the compiler, utility programs, and header files.

On November 13, 2006, Sun released much of Java as free and open source software, (FOSS), under the terms of the GNU General Public License (GPL). On May 8, 2007, Sun finished the process, making all of Java's core code available under free software/open-source distribution terms, aside from a small portion of code to which Sun did not hold the copyright.

Sun's vice-president Rich Green said that Sun's ideal role with regards to Java was as an "evangelist." Following Oracle Corporation's acquisition of Sun Microsystems in 2009–

2010, Oracle has described itself as the "steward of Java technology with a relentless commitment to fostering a community of participation and transparency". This did not hold Oracle, however, from filing a lawsuit against Google shortly after that for using Java inside the Android SDK (see Google section below). Java software runs on everything from laptops to data centers, game consoles to scientific supercomputers. There are 930 million Java Runtime Environment downloads each year and 3 billion mobile phones run Java. On April 2, 2010, James Gosling resigned from Oracle.



Duke, the Java mascot

3. Principles:-

There were five primary goals in the creation of the Java language :

1. It should be "simple, object-oriented and familiar"
2. It should be "robust and secure"
3. It should be "architecture-neutral and portable"
4. It should execute with "high performance"
5. It should be "interpreted, threaded, and dynamic"

4. Applets, Servlets and Application:-

- An *applet* is designed to be embedded in a Web page, and run by a browser.
- Applets run in a *sandbox* with numerous restrictions; for example, they can't read files and then use the network.
- A *servlet* is designed to be run by a web server.

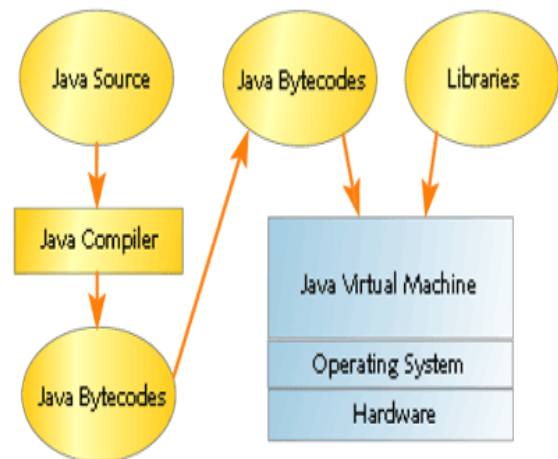
- An *application* is a conventional program.

5. Building Standalone JAVA Programs (on UNIX):-

- Prepare the file foo.java using an editor
- Invoke the compiler: javac foo.java
- This creates foo.class
- Run the java interpreter: java foo

6. Java Virtual Machine:-

- The .class files generated by the compiler are not executable binaries
 - so Java combines compilation and interpretation
- Instead, they contain "byte-codes" to be executed by the Java Virtual Machine
 - other languages have done this, e.g. UCSD Pascal
- This approach provides platform independence, and greater security.



7. Criticism:-

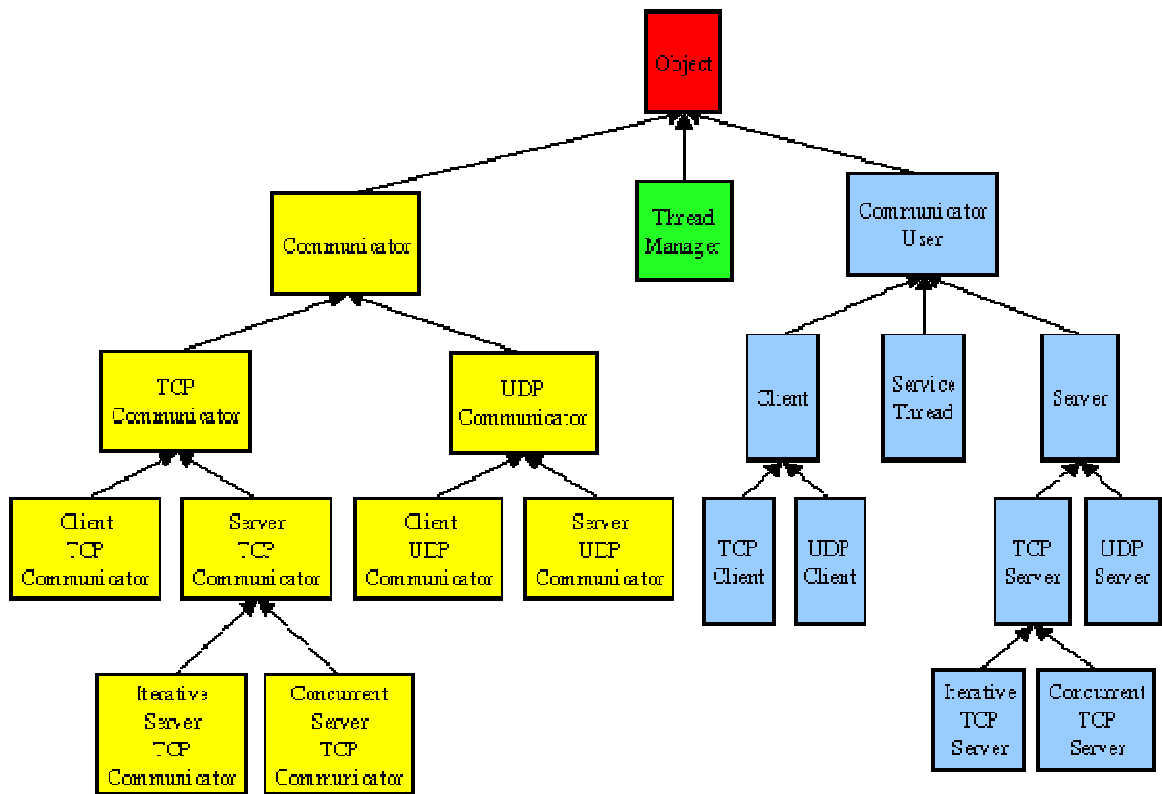
Criticisms directed at Java include the implementation of generics, speed, the handling of unsigned numbers, the implementation of floating-point arithmetic, and a history of security vulnerabilities in the primary Java VM implementation Hotspot.

8. Class libraries:-

- The Java Class Library are the compiled bytecodes of source code developed by the JRE implementor to support application development in Java. Examples of these libraries are:
 - The core libraries, which include:
 - Collection libraries that implement data structures such as lists, dictionaries, trees,

- sets, queues and double-ended queue, or stacks
- XML Processing (Parsing, Transforming, Validating) libraries
- Security
- Internationalization and localization libraries
- The integration libraries, which allow the application writer to communicate with external systems. These libraries include:
 - The Java Database Connectivity (JDBC) API for database access
 - Java Naming and Directory Interface (JNDI) for lookup and discovery
 - RMI and CORBA for distributed application development
 - JMX for managing and monitoring applications
- User interface libraries, which include:
 - The (heavyweight, or native) Abstract Window Toolkit (AWT), which provides

- GUI components, the means for laying out those components and the means for handling events from those components
- The (lightweight) Swing libraries, which are built on AWT but provide (non-native) implementations of the AWT widgetry
- APIs for audio capture, processing, and playback
- A platform dependent implementation of the Java virtual machine that is the means by which the bytecodes of the Java libraries and third party applications are executed.
- Plugins, which enable applets to be run in Web browsers.
- Java Web Start, which allows Java applications to be efficiently distributed to end-users across the Internet.
- Licensing and documentation.



9. Editions:-

Sun has defined and supports four editions of Java targeting different application environments and segmented many of its APIs so that they belong to one of the platforms. The platforms are:

- Java Card for smartcards.
- Java Platform, Micro Edition (Java ME) — targeting environments with limited resources.
- Java Platform, Standard Edition (Java SE) — targeting workstation environments.
- Java Platform, Enterprise Edition (Java EE) — targeting large distributed enterprise or Internet environments.

The classes in the Java APIs are organized into separate groups called packages. Each package contains a set of related interfaces, classes and exceptions. Refer to the separate platforms for a description of the packages available.

The set of APIs is controlled by Sun Microsystems in cooperation with others through the Java Community Process program. Companies or individuals participating in this process can influence the design and development of the APIs. This process has been a subject of controversy.

Sun also provided an edition called PersonalJava that has been superseded by later, standards-based Java ME configuration-profile pairings.

References:-

1. "The History of Java Technology". Retrieved October 6, 2012.
2. Gosling, James; and McGilton, Henry (May 1996). "The Java Language Environment"
3. "Programming Language Popularity". 2009.
4. "The Java Language Environment". *1.2 Design Goals of the Java™ Programming Language*. Oracle.