

Introduction to Java World

History of Java

- Story starts in early 1990s
- Sun Microsystems was working on consumer electronics with the objective of keeping price down.
- Software for consumer electronics is developed on the cheapest chips available
- If a different chip set becomes cheaper, the developers must switch to that chip.
- Sun's goal → develop software that was portable so that it could be switched quickly to new hardware.



a snapshot taken at a barbecue with
the Green Team

History



- The (secret) Green project started in 1991 and the goal was to create an experimental, hand-held computer (PDA) called the *7, intended for controlling home appliances.
- James Gosling was the technical leader of the project.
- The character "Duke" was planned to exist on the UI.
- A language was developed for the *7 which was called Oak.
- The *7 never became a product
 - but due to the explosion in popularity of the WWW, the team saw that the language could be very useful there



History

- The team returned to work up a Java technology-based clone of Mosaic they named "WebRunner" (after the movie *Blade Runner*), later to become officially known as the HotJava™ browser in 1994.
- WebRunner was just a demo, but an impressive one: It brought to life, for the first time, animated, moving objects and dynamic executable content inside a Web browser. That had never been done.

WebRunner

WebRunner is a World Wide Web browser that brings true interactivity to the Internet. WebRunner makes the Internet "come alive". It builds on the network browsing techniques established by Mosaic and expands them by adding dynamic behavior that transforms static documents into dynamic applications capable of real-time interactive response. Using WebRunner you can create applications that range from interactive games to dynamic forms to customized newspapers to interactive shopping ... the possibilities are endless.

WebRunner also provides a new way for users to access these applications. Software transparently migrates across the network. There is no such thing as "installing" software. It just comes when you need it. Content developers can embed new software, media types and protocols in WWW pages, and the extensions reach the user's system automatically.

What makes this dynamic behavior possible is Java (formerly "Oak"), the underlying environment in which WebRunner is built. Java is a simple, dynamic, multithreaded, safe, compact and portable object-oriented programming language and runtime system. Java has an architecture-neutral distribution format, so that Java content runs on anyone's WebRunner home page, regardless of the underlying CPU architecture.

WebRunner Alpha2 Features:

- Full-Function WWW Browser
 - HTML-compatible with Mosaic and Netscape
 - supports all standard Internet protocols
 - fast performance
- Enables Interactive Content
- Dynamic Content Loading
 - occurs transparently across the network
 - add new protocols and applications on the fly
 - platform-independent
- WWW Newsreader
- Security and Authentication Support
 - file system protection and code checking
- Includes full Java language and runtime system
- Available on Solaris

WebRunner Beta Features:

- WYSIWYG HTML Editor
 - no HTML knowledge necessary
 - implements all common HTML extensions
 - drag and drop links, images, audio, applications into browser
- Security and Authentication Support
 - S-HTTP, RSA public key encryption and authentication
- HTTP server with support for real-time data feeds
- Available on Solaris, Windows95, and MacOS

For instructions on installing WebRunner, visit <http://tachyon.eng/folio/>



WebRunner



*San Jose Mercury News,
March 23, 1995*

History

- The language was first called Oak (after an oak tree outside Goslings window). However, there was already a language named Oak so the team named the language Java (in recognition of the role that caffeine plays in software development).
- Sun formally announced Java at a major conference in 1995.

History

- JDK1.0 (January 23, 1996)
- JDK1.1 (February 19, 1997)
- J2SE1.2 (December 8, 1998) → Java 2
- J2SE1.3 (May 8, 2000)
- J2SE1.4 (February 6, 2002)
- J2SE5.0 (September 30, 2004) ← Java5
- JSE6 (December 11, 2006)
- JSE7 (July 28, 2011) ← Java7
- JSE8 (Late 2012) ← Java8

History

- Major byproduct of the Green project was "Duke"
- Duke was the "agent" in the Green user interface.
- Became the Java mascot.



What Is Java?

- According to Sun in a white paper:

Java: A simple, object-oriented, network-savvy, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded, dynamic language

Java is Simple

- Java is partially modeled on C++ but greatly simplified and improved.

Examples: Pointers & multiple inheritance often make programming complicated. Java replaces the multiple inheritance in C++ with a simple language construct and eliminates pointers.

Java uses **automatic memory allocation** and **garbage collection**.

Java is Object-Oriented

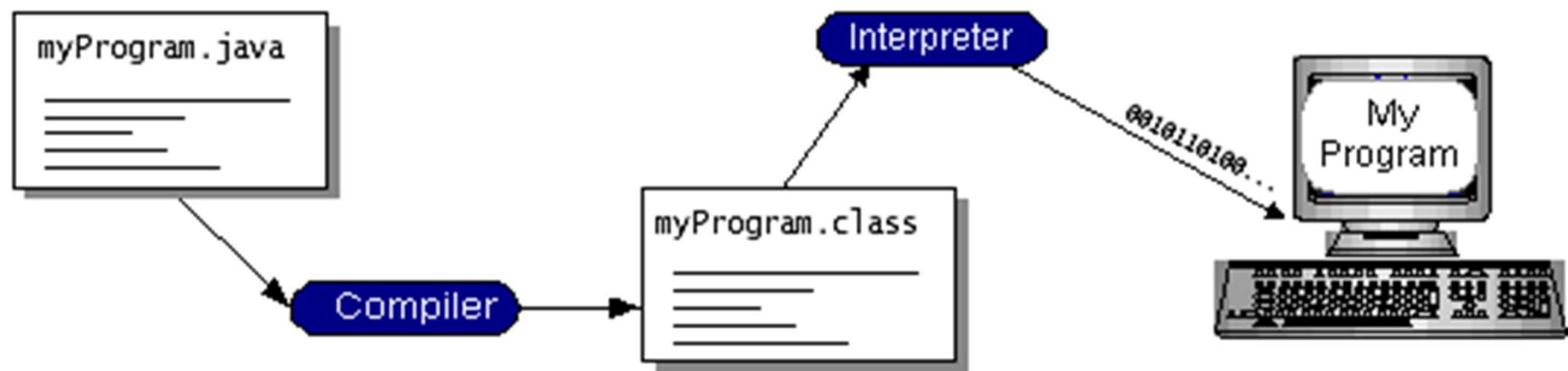
- It is an Object-Oriented programming language.
- It has constructs to implement **encapsulation, polymorphism** and **inheritance**.

Java is Distributed

- Distributed computing involves several computers working together on a network.
- Networking capability is inherently integrated into Java so writing network programs is like sending and receiving data to and from a file.

Java is Compiled and Interpreted

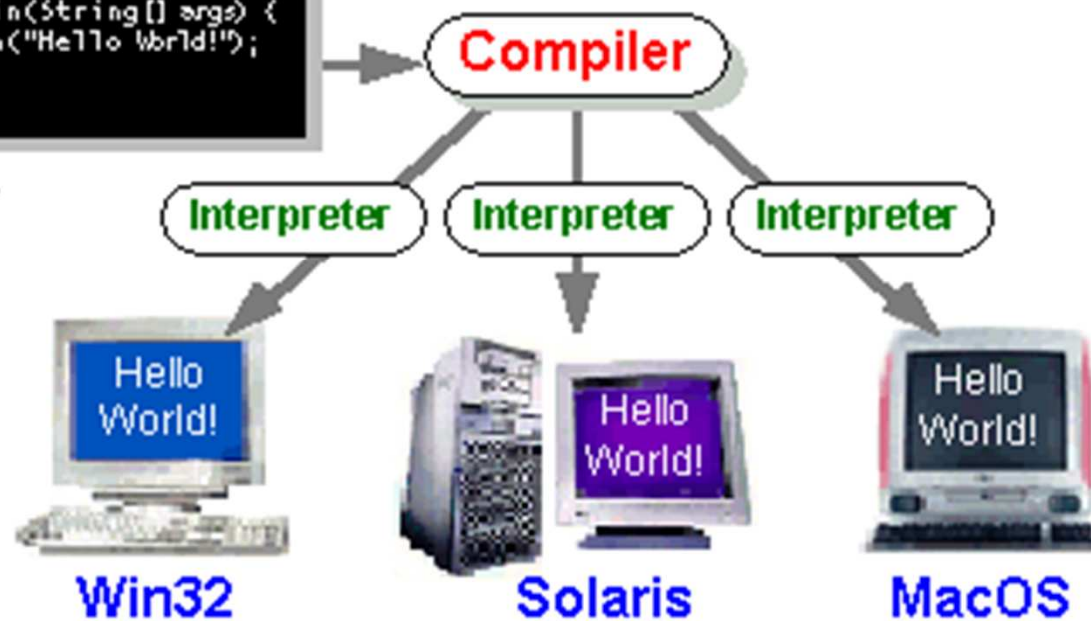
- The Java platform has a **compiler** that translates Java source into a form called ***bytecodes***
- ***Bytecode*** is an architecturally neutral representation of code written in the Java programming language. It is machine-independent and can run on any machine that has a Java interpreter.
- The *bytecode* rather than Java source code, is **interpreted** when you run a Java program.



Java Program

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

HelloWorldApp.java



Java is Robust

- **Robust** means *reliable*.
- Java has eliminated certain error-prone programming constructs found in other languages.
- It doesn't support pointer arithmetic, for example, thereby eliminating the possibility of overwriting memory and corrupting data.
- Java has a runtime exception-handling feature to provide programming support for robustness.
- The programmer **must** write the code to deal with exceptions.

Java is Secure

- This is essential since it lends itself to programming for the Internet & WWW
- If you download a Java applet and run it on your computer, it will not damage your system because Java implements several security mechanisms.
- Java security is based on the premise that *nothing should be trusted*.

Java is Architecture neutral

- The language allows for no implementation-defined features.

Example: the size of "int"

- All Java programs must be compiled into bytecodes before the JVM can run them.

Java is Portable

- Java programs can be run on any platform without being recompiled, making them very portable.
- There are no platform-specific features in Java (size of an int).

Java's Performance

- Surprisingly good compared to other interpreted languages
- e.g. The new Java Development Kit uses the technology known as *just-in-time compilation*.
 - JVM searches for **hot spots** (parts of the bytecode executed frequently)
 - JIT compiler translates the hot spots into underlying machine language

Java is Multithreaded

- This is the capability for a program to perform several tasks simultaneously within a program
Example: Download a video file while playing it
- Used in graphical user interfaces (listen to an audio recording while surfing a Web page)
- Used in network programming (a server can serve multiple clients at the same time).
- Classes are provided from the base language package to create and manage threads

Java is Dynamic

- Java was designed to adapt to an evolving environment.
- You can freely add new methods to a class without affecting its clients.

Example: In the Circle class, you can add a new data property to indicate the color of the circle or a new method to obtain the circumference of the circle. The original client program that uses the circle class remains the same.

- Also, at runtime Java loads classes as they are needed.

Java Technologies

- The main technology platforms
 - Java Platform, Micro Edition (Java ME) — targeting environments with limited resources,
 - Java Platform, Standard Edition (Java SE) — targeting workstation environments, and
 - Java Platform, Enterprise Edition (Java EE) — targeting large distributed enterprise or Internet environments.
 - Java FX — intended to be used to create Rich Internet applications.
- and many other supporting technologies

Java Technologies

- The classes in the Java APIs are organized into separate groups called packages.
 - Each package contains a set of related interfaces, classes and exceptions.
- 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.
→ Java is an **Open Standard!**