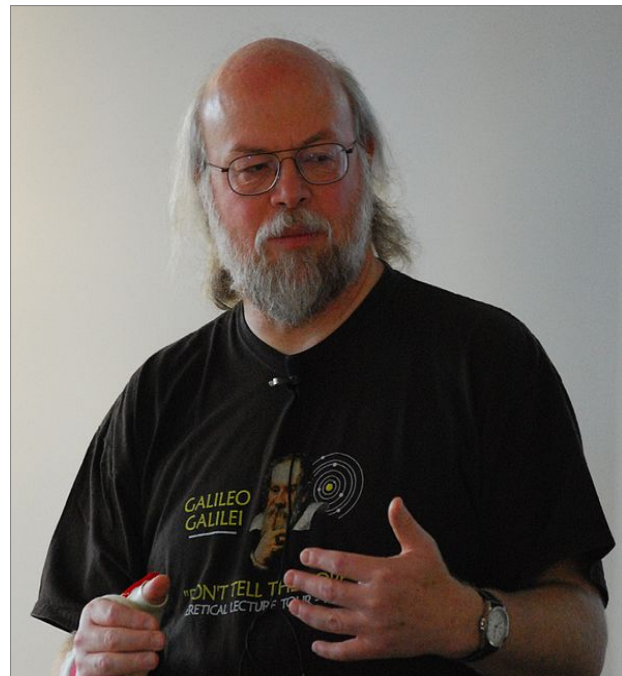


Java

Introduction

- Developed by James Gosling at Sun in the mid-1990s as a result of his "frustration" with C++
- First released in fall 1995
- Java applets would run on web pages
- Now days:
 - Enterprise software - ERP, CRM, banks, government
 - Science laboratories
 - Web applications - Facebook, Twitter, etc.
 - Mobile devices - regular and smart phones (Android)
 - Devices with limited resources - set-top boxes, PDAs, etc.



Version history

- JDK 1.0 (January 23, 1996)
- JDK 1.1 (February 19, 1997) - Inner classes, JavaBeans, JDBC, RMI, reflection, etc.
- J2SE 1.2 (December 8, 1998) - Swing, applet, collections framework, etc.
- J2SE 1.3 (May 8, 2000) - HotSpot JVM, JNDI, etc.
- J2SE 1.4 (February 6, 2002) - Regular expressions, exception chaining, IPv6 support, logging API, etc.
- J2SE 5.0 (September 30, 2004) - Generics, annotations, autoboxing, enumerations, varargs, etc.
- Java SE 6 (December 11, 2006) - Performance improvements in the core and Swing
- Java SE 7 (July 28, 2011) - Dynamic languages (Clojure, Groovy, Scala, Jruby, Jython, Rhino, etc.)
- Java SE 8 (March 18, 2014) - Lambda expressions (closures), default methods, etc.
- Java SE 9 (September 21, 2017) - Modularization of the JDK a.k.a. Project Jigsaw, Java Shell, etc.

Key features

- Concurrent, class-based, object-oriented, general purpose programming language
- Source code is compiled into bytecode which is interpreted by a Java Virtual Machine (JVM)
- OS specific JVMs allow developers to "write once, run everywhere"
- "Easy" to learn and use compared to e.g. C++
 - Built-in memory management (allocation and deallocation) which is a tedious and complex task
 - No support for pointers which are powerful but can be easily misused
 - Supports single inheritance

Java environment

- **JDK** - Java Development Kit
 - Contains tools needed to develop Java programs, and the JRE to run the programs. Tools include: compiler (javac.exe), Java application launcher (java.exe), ...
- **JRE** - Java Runtime Environment
 - Contains JVM, class libraries and other supporting files. It contains the launcher but does not contain any development tools such as compiler, debugger. If you want to run any Java program, you need to have JRE installed in the system.
- **JVM** - Java Virtual Machine
 - When we compile a Java file, output is not an "exe" but a so called class file (extension is ".class"). A class file consists of Java byte code which is understandable by the JVM. Java Virtual Machine interprets the byte code into the machine code depending upon the underlying operating system and hardware combination.
 - It's called "virtual" because it provides a machine interface that does not depend on the underlying operating system and machine hardware architecture.

Editions

- Java Card - For smartcards
- Java Platform, Micro Edition (Java ME) - targets environments with limited resources
- Java Platform, Standard Edition (Java SE) - targets workstation environments
- Java Platform, Enterprise Edition (Java EE) - targets large distributed enterprise or internet environments

Development environment

- Text Editor + JDK
 - Notepad, Notepad++, Sublime Text, Vi, Emacs, etc.
- IDE (Integrated Development Environment) + JDK
 - A software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of:
 - Source code editor
 - Building automation tool
 - Debugger, etc.
 - Well known Java IDEs: IntelliJ IDEA, Eclipse, NetBeans

Exercises

Exercise: HelloWorld

1. Create a new file with the following content and name it “HelloWorld.java”

```
public class HelloWorld {  
  
    /**  
     * The HelloWorldApp class implements an application  
     * that simply prints "Hello Java Developers!" message to standard output.  
     *  
     * @param args  
     */  
    public static void main(String[] args) {  
        System.out.println("Hello Java Developers!");  
    }  
}
```

Exercise: HelloWorld

2. Compile the source file:

```
$ javac HelloWorld.java
```

3. Run the HelloWorld application:

```
$ java HelloWorld
```