Object-Oriented Programming in Java

Guixing Wu
507- Si Xian Building
gxwu@ustc.edu.cn

Course Material

- http://docs.oracle.com/javase/tutorial/
- Bruce Eckel, Java编程思想，英文版第4版，机械工业出版社，2012.
- Cay S. Horstman, Gary Cornell, Java 核心技术，英文版第9版，人民邮电出版社，2013.
- 工业和信息化部教育与考试中心组编，清华大学计算机与信息管理中心主编， Java语言程序设计，中国铁道出版社，2010.
- 龚永罡、陈昕等编，Java语言程序设计基础教程，清华大学出版社，2009.

What is this class about?

- An introduction to Java
  - Java history, about the Java technology, object-oriented programming concepts
- Learing the Java lanuage
  - Language basics, classes and objects, annotations, interface and inheritance, number and strings, generics,package
- Essential Java Classes
  - Exceptions, Basic I/O, concurrency, regular Expression
- Collections
- Custom netowrking

The Java Technology Phenomenon

Java history

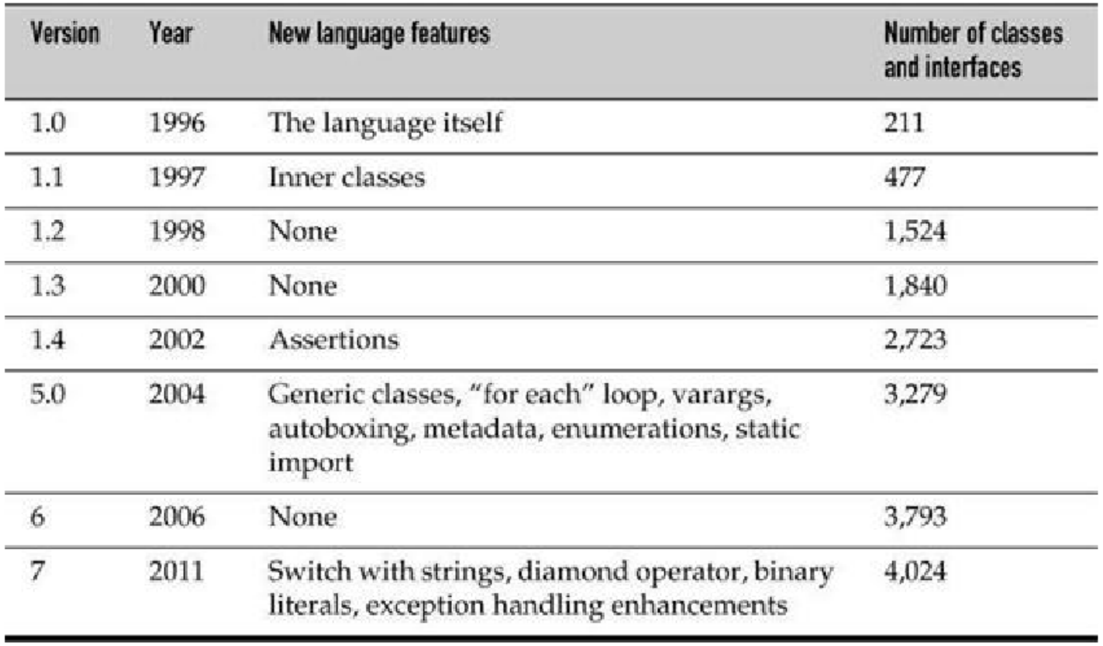
About the Java technology

What can Java technology do?

How will Java technology change my life?

Java History

- Was created in 1991 by James Gosing
- Relased buy Sun in 1995



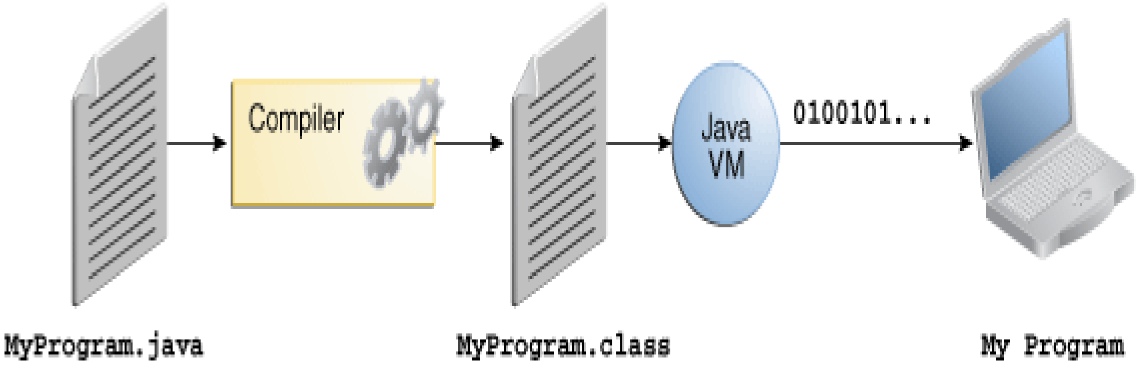
Version	Year	New language features	Number of classes and interfaces
1.0	1996	The language itself	211
1.1	1997	Inner classes	477
1.2	1998	None	1,524
1.3	2000	None	1,840
1.4	2002	Assertions	2,723
5.0	2004	Generic classes, "for each" loop, varargs, autoboxing, metadata, enumerations, static import	3,279
6	2006	None	3,793
7	2011	Switch with strings, diamond operator, binary literals, exception handling enhancements	4,024

About the Java Technology

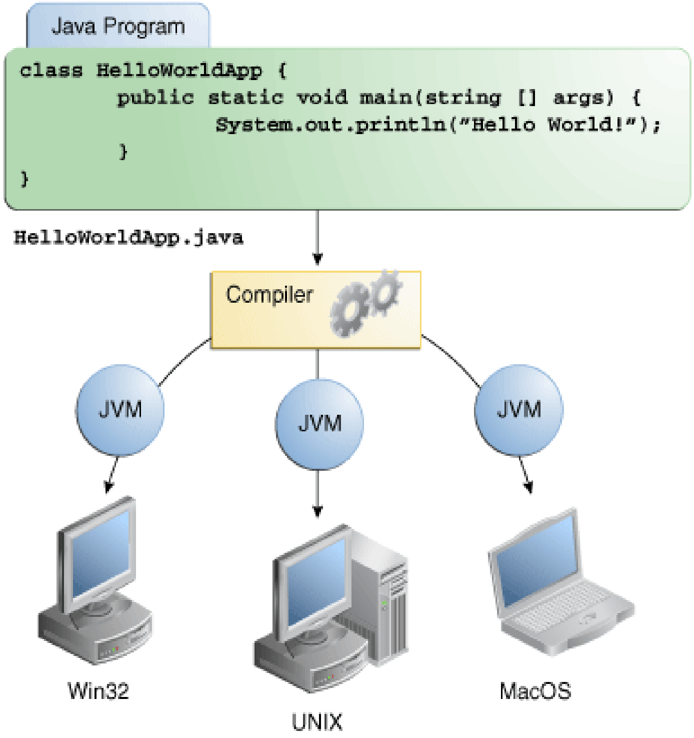
The Java Programming Language

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Multithreaded
- Robust
- Dynamic
- Secure

Java Software Development Process



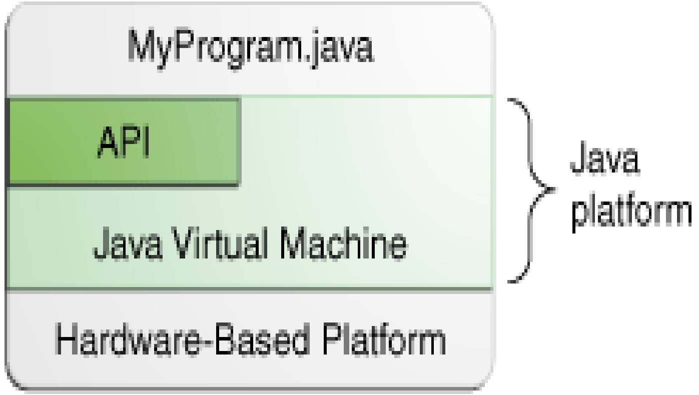
Running Java Applications



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

The Java Platform

- The *Java Virtual Machine*
- The *Java Application Programming Interface* (API)



What Can Java Technology Do?

Java platform gives you the following features:

- **Development Tools**: compiling, running, monitoring, debugging, and documenting your applications.
- **Application Programming Interface (API)**: It offers a wide array of useful classes ready for use in your own applications.
- **Deployment Technologies**: The JDK software provides standard mechanisms such as the Java Web Start software and Java Plug-In software for deploying your applications to end users.
- **User Interface Toolkits**: The JavaFX, Swing, and Java 2D toolkits
- **Integration Libraries**: enable database access and manipulation of remote objects.

How Will Java Technology Change My Life?

- **Get started quickly**: it's easy to learn
- **Write less code**: a program written in the Java can be four times smaller than the same program written in C++.
- **Write better code**:
  - it encourages good coding practices;
  - automatic garbage collection helps you avoid memory leaks;
  - object orientation, JavaBeans™ component architecture, and wide-ranging, easily extendible API let you reuse existing, tested code and introduce fewer bugs.
- **Develop programs more quickly**:
  - The Java programming language is simpler than C++;
  - development time could be up to twice as fast
  - programs will also require fewer lines of code.

How Will Java Technology Change My Life?

- **Avoid platform dependencies**: You can keep your program portable by avoiding the use of libraries written in other languages.
- **Write once, run anywhere**: Because applications written in the Java are compiled into machine-independent bytecodes, they run consistently on any Java platform.
- **Distribute software more easily**:
  - With Java Web Start software, users will be able to launch your applications with a single click of the mouse.
  - An automatic version check at startup ensures that users are always up to date with the latest version of your software.
  - If an update is available, the Java Web Start software will automatically update their installation.

"Hello World!" Application for Microsoft Windows

To write your first program, you'll need:

- The Java SE Development Kit X (JDK X)
- A text editor

Creating Your First Application

To create Your first application, HelloWorldApp, you will:

- **Create a source file**
  - A source file contains code, written Java.
  - You can use any text editor to create and edit source files.
- **Compile the source file into a .class file**
  - The Java programming language *compiler* (javac) takes your source file and translates its text into instructions that the Java virtual machine can understand.
  - The instructions contained within this file are known as *bytecodes*.
- **Run the program**
  - The Java application *launcher tool* (java) uses the Java virtual machine to run your application.

Create a Source File

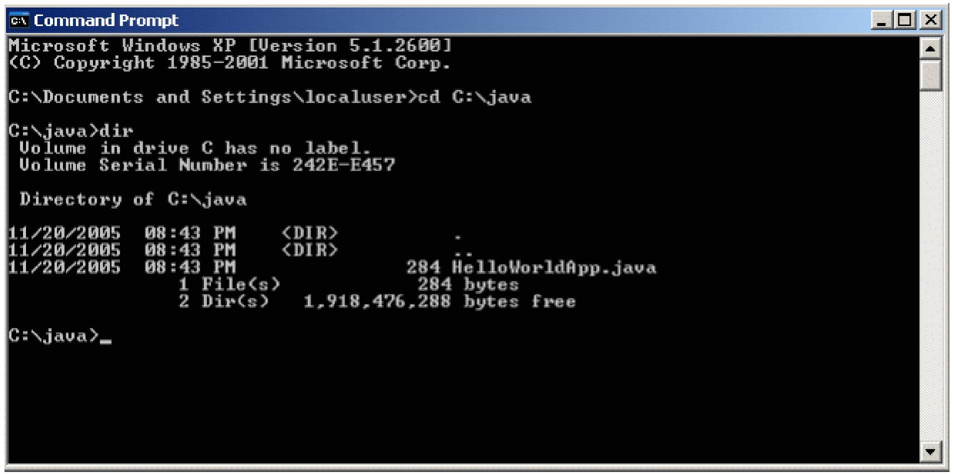
```
/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
```

Note: Type all code, commands, and file names exactly as shown. Both the compiler (javac) and launcher (java) are *case-sensitive*, so you must capitalize consistently.

Compile the Source File into a .class File

Save the code with the name HelloWorldApp.java

Bring up a shell, or "command," window. Change your current directory to the directory where your file is located.



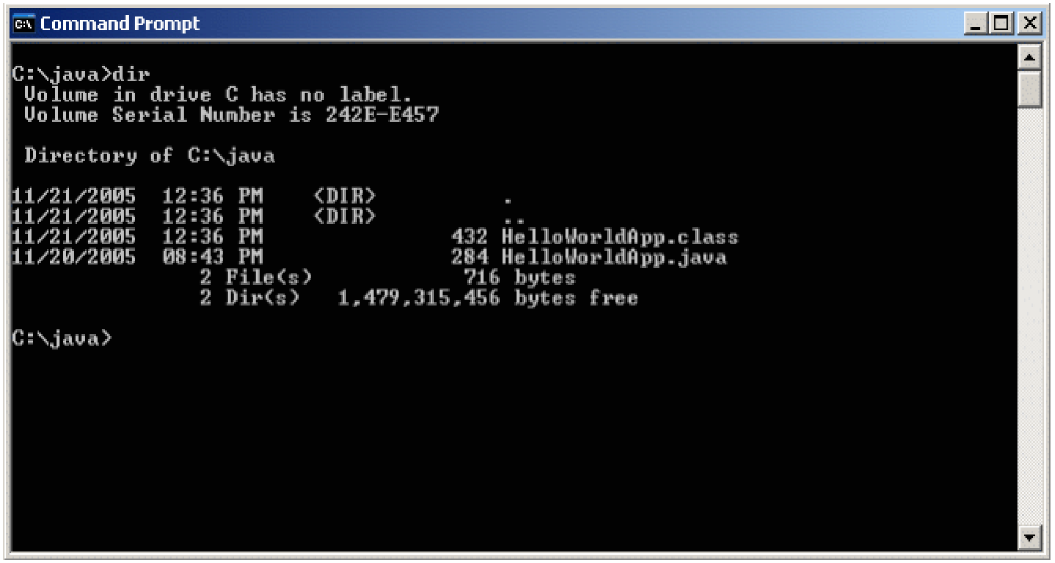
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\localuser>cd C:\java

C:\java>dir
 Volume in drive C has no label.
 Volume Serial Number is 242E-E457

 Directory of C:\java

11/20/2005  08:43 PM    <DIR>          .
11/20/2005  08:43 PM    <DIR>          ..
11/20/2005  08:43 PM               284 HelloWorldApp.java
               1 File(s)            284 bytes
               2 Dir(s)   1,918,476,288 bytes free

C:\java>_
```

Compile the Source File into a .class File

At the prompt, type the following command and press **Enter**.

javac HelloWorldApp.java

The compiler has generated a bytecode file, HelloWorldApp.class.



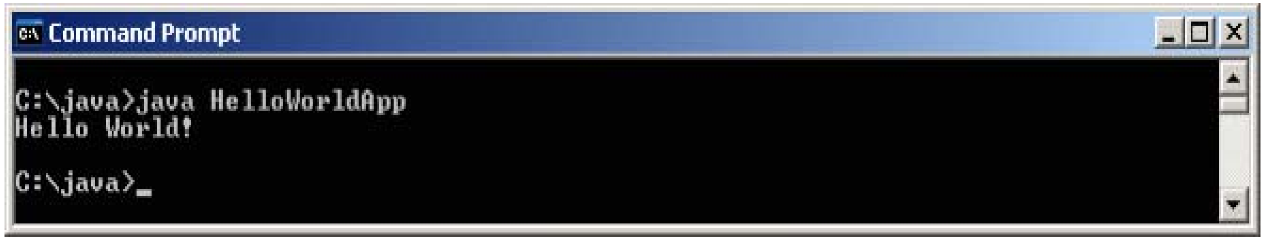
```
C:\java>dir
 Volume in drive C has no label.
 Volume Serial Number is 242E-E457

 Directory of C:\java

11/21/2005  12:36 PM    <DIR>          .
11/21/2005  12:36 PM    <DIR>          ..
11/21/2005  12:36 PM               432 HelloWorldApp.class
11/20/2005  08:43 PM               284 HelloWorldApp.java
               2 File(s)            716 bytes
               2 Dir(s)   1,479,315,456 bytes free

C:\java>
```

Run the Program

In the same directory, enter the following command at the prompt:

```
java HelloWorldApp
```

The next figure shows what you should now see:



```
C:\java>java HelloWorldApp
Hello World!

C:\java>_
```

A Closer Look at the "Hello World!" Application

```
/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
```

The application consists of three primary components:

- source code comments
- the HelloWorldApp class definition
- the main method.

Source Code Comments

- Comments are ignored by the compiler but are useful to other programmers.
- The Java programming language supports three kinds of comments:
  - */* text */*

    The compiler ignores everything from /* to */.
  - */** documentation */*

    This indicates a documentation comment (*doc comment*, for short). The javadoc tool uses doc comments when preparing automatically generated documentation.
  - *// text*

    The compiler ignores everything from // to the end of the line.

The HelloWorldApp Class Definition

The most basic form of a class definition is:

```
class name {
      . . .
}
```

The main Method

- Every application must contain a main method whose signature is:

  public static void main(String[] args)

- The modifiers public and static can be written in either order (public static or static public), but the convention is to usepublic static as shown above.
- You can name the argument anything you want, but most programmers choose "args" or "argv".
- The main method is the entry point for your application and will subsequently invoke all the other methods required by your program.

The main Method

- The main method accepts a single argument: an array of elements of type String.

  ```
  public static void main(String[] args)
  ```

- This array is the mechanism through which the runtime system passes information to your application. For example:

  ```
  java MyApp arg1 arg2
  ```

- Each string in the array is called a *command-line argument*.
- The line:

  ```
  System.out.println("Hello World!");
  ```

  uses the System class from the core library to print the "Hello World!" message to standard output.

Homework

- Read Core Java Volume I Chapter 2
- Intall JDK and Eclipse
- Run "Hello World!" Application