JAVA 程序设计

1. Course Overview and Java Introduction

黄　骏

jun.huang@pku.edu.cn

课程介绍

- 黄骏
- 北京大学理科五号楼515S
- jun.huang@pku.edu.cn
- 18310998523

- 每周二7-8节
- 文史201

参考资料

- Java程序设计（第二版）
  - 清华大学出版社、北京交通大学出版社
  - 唐大仕

- 华文慕课：Java程序设计
  - http://www.chinesemooc.org/mooc/4746

成绩标准

- Homework 3 x 10% = 30%
- Course project 40%
- Final Exam 30%

作业说明

- 迟交减50%
- 提交源代码
- 作业-1
  - 俄罗斯方块机器人
- 作业-2
  - 基于数据挖掘的文本分类
- 作业-3
  - TBD，涉及网络

课设说明

- 成绩占比40%
  - 源码、报告和ppt（Due 5.20号0点）
- 题目任选
- 2到5人一组
- 由所有同学根据pre，代码和报告一起打分
- Pre安排在14到16周

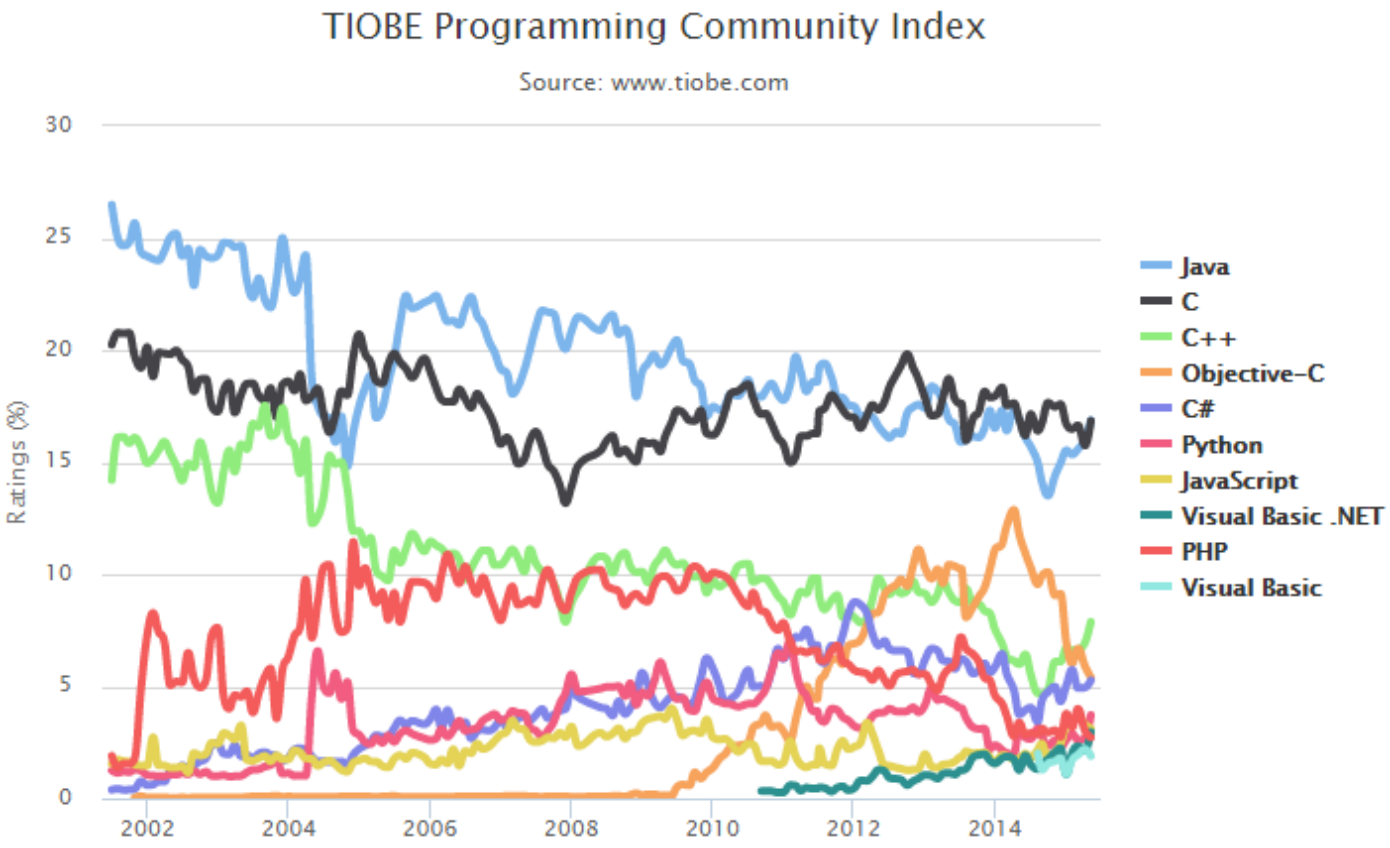
- 第1周　　课程介绍和Java简介
- 第2周　　数据运算、控制流、数组
- 第3周　　面向对象（1）、 HW-1
- 第4周　　面向对象（2）
- 第5周　　面向对象（3）
- 第6周　　异常处理和输入输出
- 第7周　　HW-2
- 第8周　　泛型和集合类
- 第9周　　网络编程
- 第10周　　HW-3
- 第12周　　多线程
- 第13周　　TBD
- 第14周　　Pre（1）
- 第15周　　Pre（2）
- 第16周　　Pre（3）

Outline

- 认识Java
- Java的历史
- Java的运行机制
- Java语言的特点
- Java开发环境

What is Java

- The most popular programming language



Where is Java

- In a wide range of software systems
  - Desktop, web, mobile and embedded applications
  - MATLAB, Eclipse, Minecraft, Hadoop …
- In about 3 billions of devices
  - Scientific supercomputers, data centers, smartphones, printers, game consoles …

Java Platforms

- Java SE (Java Standard Edition)
  - Desktop and server environments
- Java EE (Java Enterprise Edition)
  - Distributed computing and web services
  - Large-scale, multi-tiered enterprise software
- Java ME (Java Micro Edition)
  - Embedded and Mobile devices

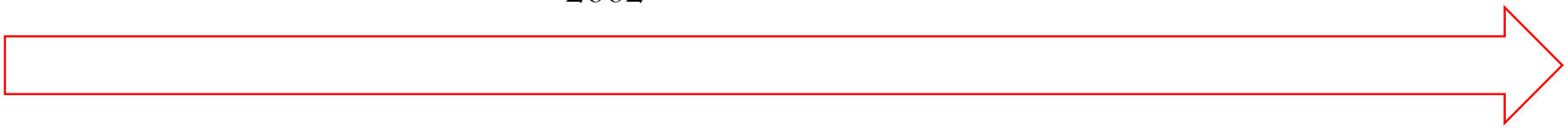
Outline

- 认识Java
- Java的历史
- Java的运行机制
- Java语言的特点
- Java开发环境

The Origin of Java

- 1991: Initiated by *James Gosling* at SUN Microsystems
- Named as "Oak", and then re-named as "Java"
- Originally designed for interactive TV, and then re-targeted to Web apps
- Designed with a C/C++ style syntax
- Java 1.0 released in 1995
- Quickly became popular, mostly outside browsers

Java Version History



1995
First release

1997
JDK 1.1
Inner classes, JavaBeans, JDBC RMI

2000
J2SE 1.3
HotSpot JVM
Debugger architecture

2004
JavaSE 1.5 (Java 5.0)
Maturity
Scalability
Security

2010
Oracle acquired SUN

2014
Java 8.0
Lambda expression

JDK 1.0
1996

J2SE 1.2
Graphical API
Known as Java2
1998

J2SE 1.4
Stability
Regular Expression, IPv6, XML, Security and Cryptograph
2002

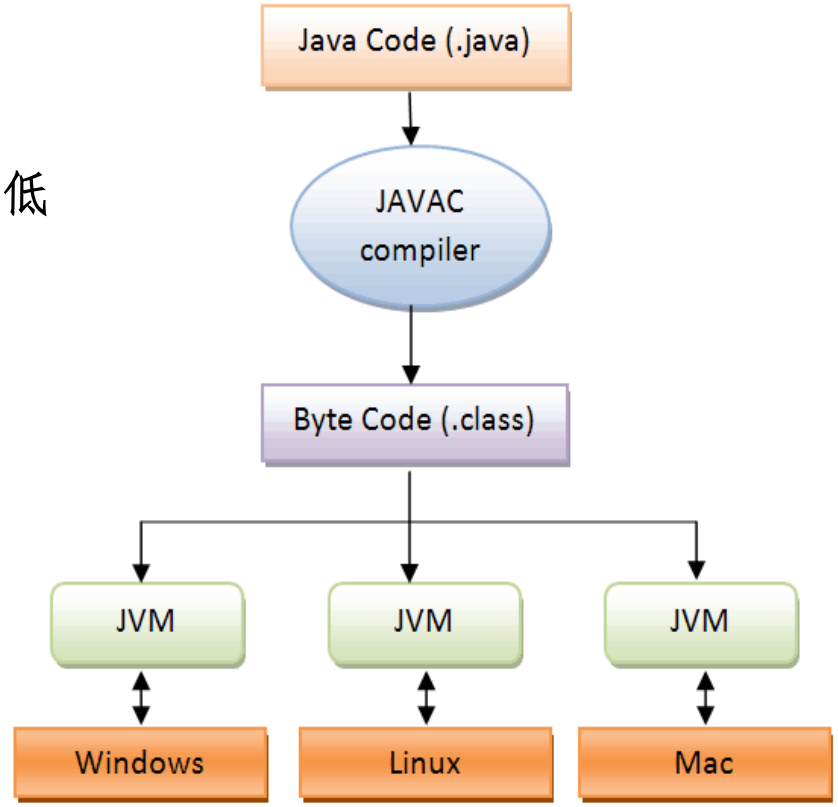
Java 6.0
JDBC 4.0
GUI Improvement
2006

Java 7.0
2011

2014
Java 9.0

Java程序运行机制

- 计算机的高级编程语言类型：
  - 解释型：
    - 运行时翻译并直接执行源程序
    - Matlab, Python, 平台兼容性，效率低
  - 编译型：
    - 执行前将源程序翻译为机器语言
    - C/C++, 高效，兼容性差
- Java 语言是两种类型的结合；
  - Java源程序(source code)
  - 字节码(byte code)
  - 机器码指令



Virtual Machine

- A *virtual machine* is a layer of abstraction that gives a program one simplified and *unified interface for* interacting with a variety of *heterogenous physical computers* and their operating systems

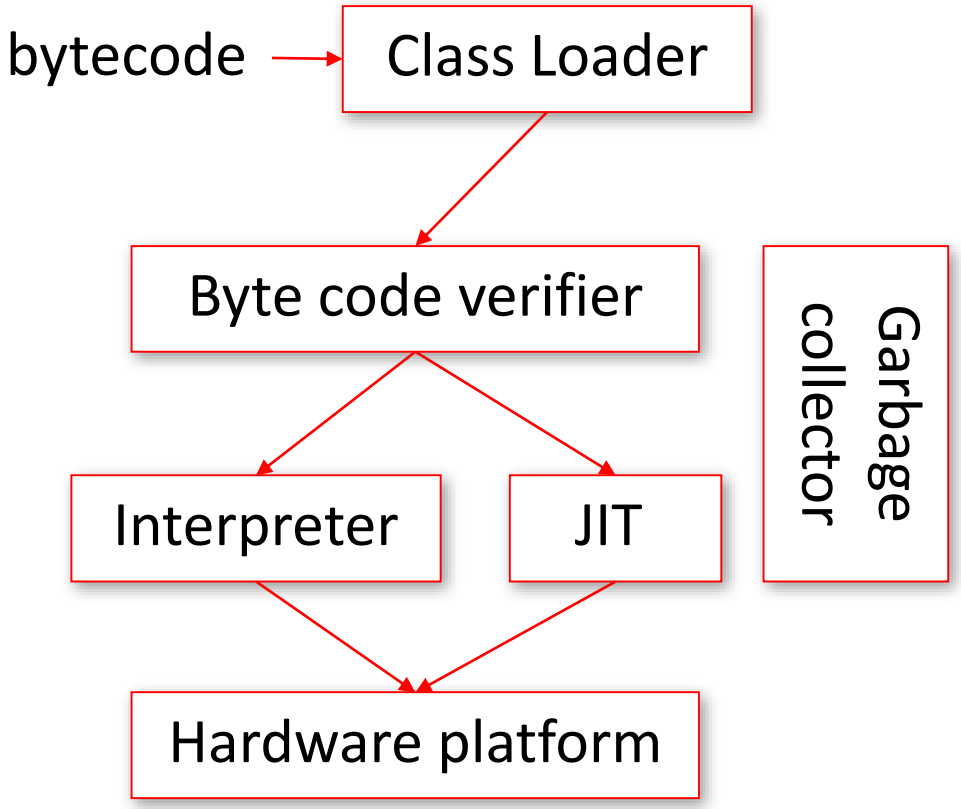
- Not a real hardware machine but a software layer that resembles an hardware platform

Java Virtual Machine (JVM)

- JVM converts Java byte code into machine language and executes it
- JVM is machine and operating system depended
- Java byte code can be executed on any platform that has a JVM
  - Write once, run anywhere

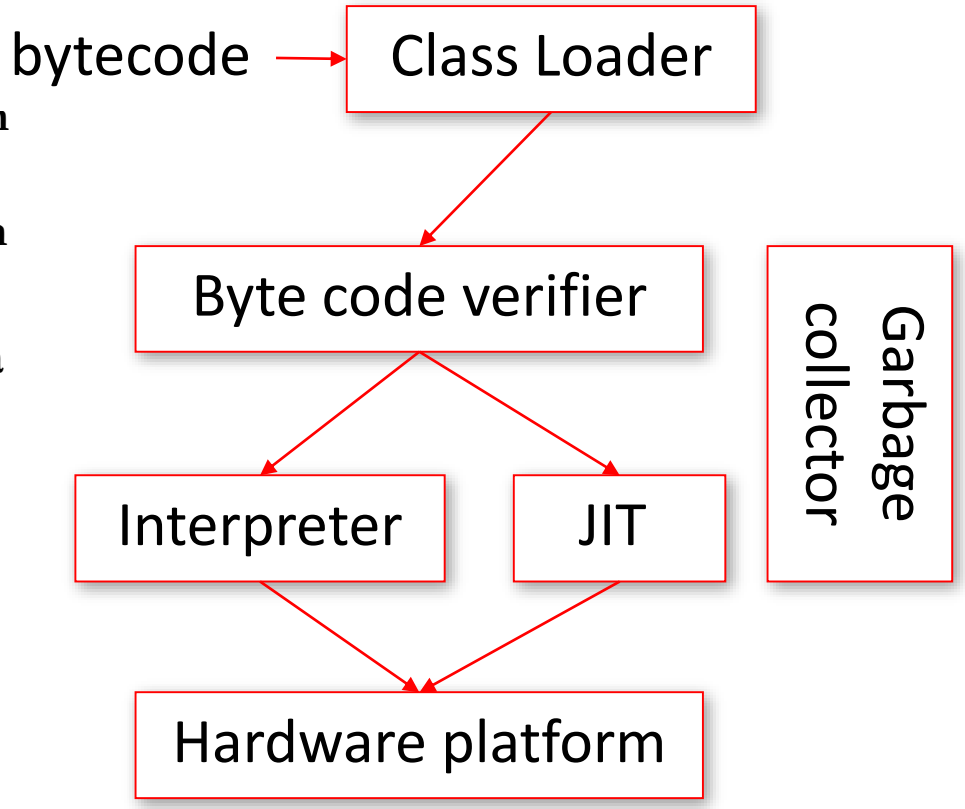
JVM Architecture

- **Class loader**
  - Loading
  - Memory allocation
  - Initialization …



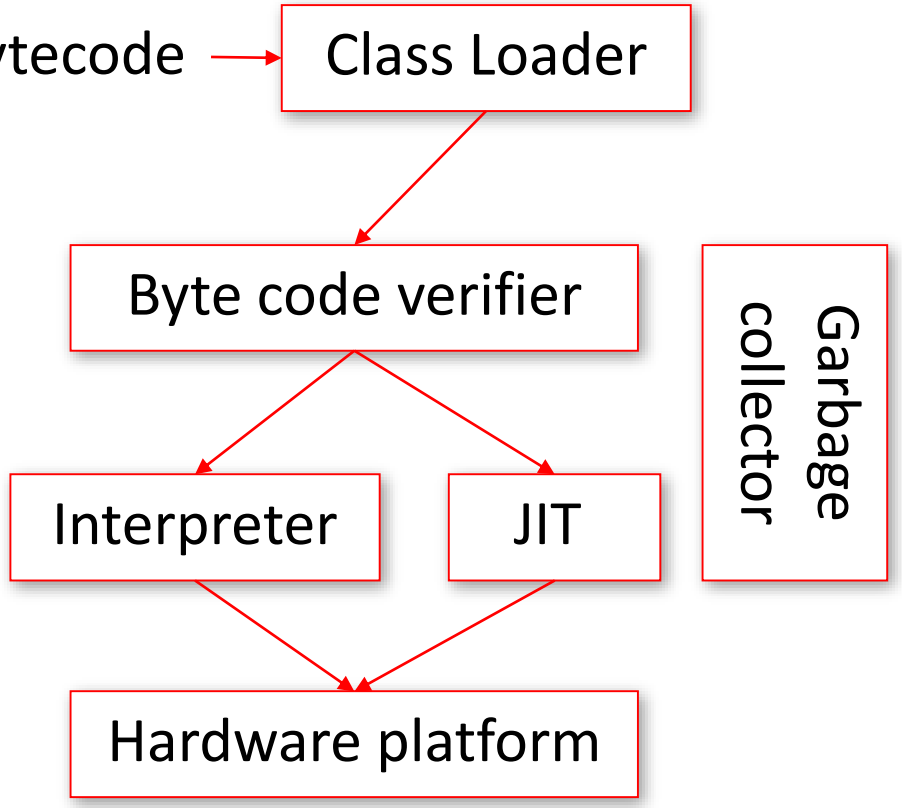
JVM Architecture

- **Byte code verifier**
  - Assures that no user program can crash the host machine or interfere with other operations on the host machine
  - Protect certain methods and data structures belonging to trusted code from access or corruption by untrusted code executing within the same JVM
  - Prevent corruptions caused by programmer errors
  - Crucial component for security



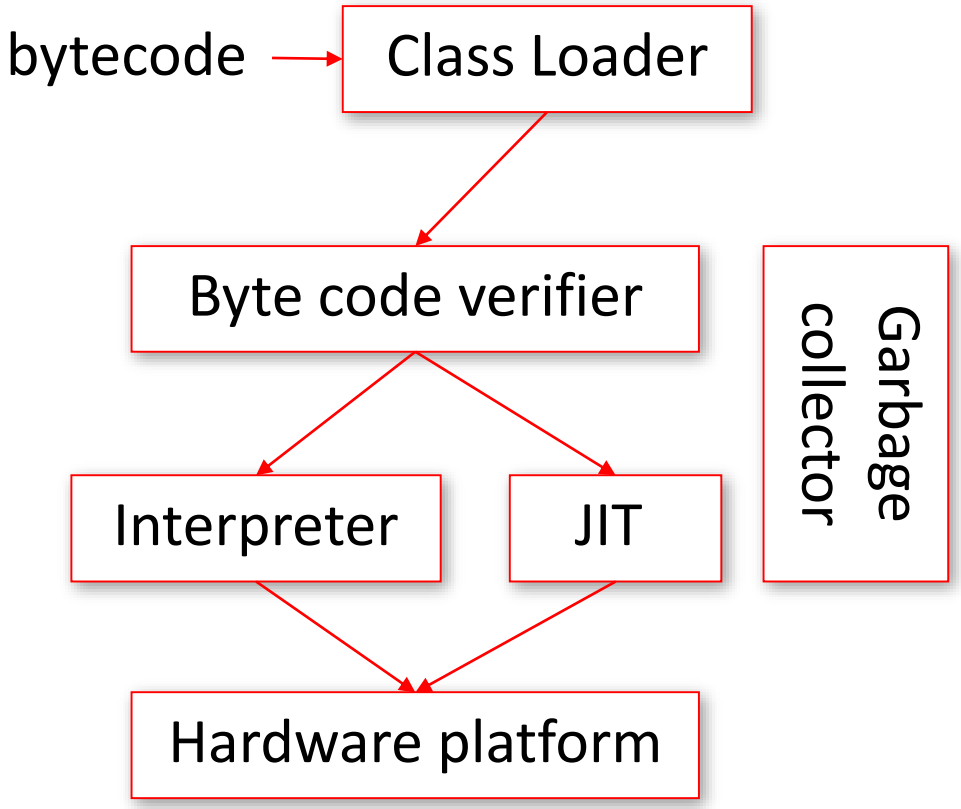
JVM Architecture

- **Interpreter and JIT**
  - Interpreting byte code by executing instructions
  - When Java bytecode is executed by an interpreter, the execution will always be slower than the execution of the same program compiled into native machine language.
  - JIT (just-in-time compiler):
    - translate byte code to machine code while executing
    - applied to frequently executed components



JVM Architecture

- Garbage collector
  - Automatically freeing objects that are no longer used by the program
  - Relieves java programmer from memory management



Java Runtime Environment (JRE)

- **A software package that contains what is required to run a Java program**
  - ***Java virtual machine***
  - ***Java class library*: a set of dynamically loadable libraries that Java applications can call at runtime**

Java Development Kit (JDK)

- is a superset of a JRE and contains tools for Java programmers, e.g. a javac compiler.

Java语言的特点

- Simple, and object oriented
  - Derived from C/C++
  - Simpler than C/C++
  - Completely object oriented
- Robust and secure
  - Extensive compile-time checking, followed by a second level of run-time checking
- Architecture neutral and portable
  - Write once, run anywhere

Java语言的特点

- High performance
  - Compute-intensive sections can be compiled into native machine code
  - Automatic garbage collection, ensuring high performance memory use
- Threaded, interpreted, and dynamic
  - Thread encapsulated as class, monitored at run-time
  - Linking to classes (even from sources across a network) when needed at run-time

丰富的类库

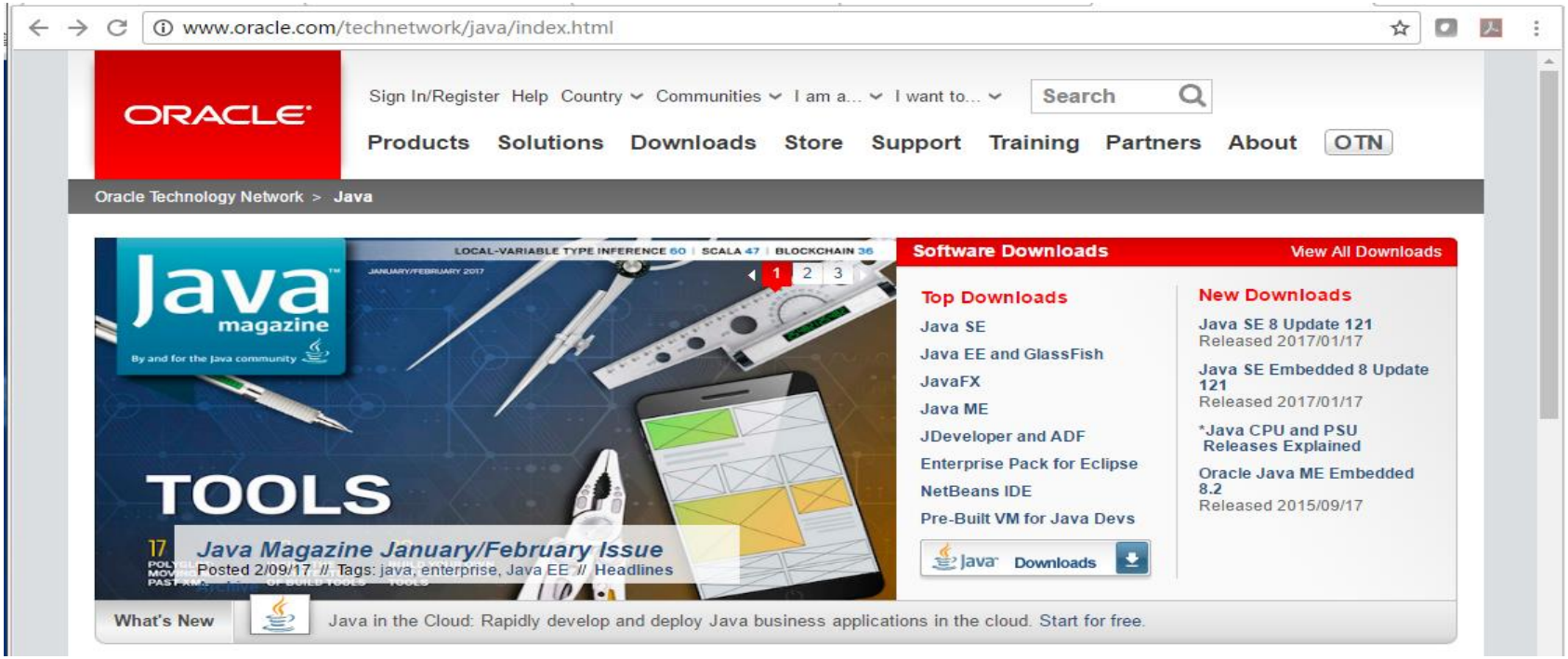
- Java提供了大量的类以满足网络化、多线程、面向对象系统的需要
  - 语言包(package)
  - 实用程序包
  - I/O包
  - 网络包
  - 图形用户界面包
  - And More……

Outline

- 认识Java
- Java的历史
- Java的运行机制
- Java语言的特点
- Java开发环境

Java Development Kit (JDK)

- Including JRE
- Can be downloaded from Oracle website



http://java.sun.com

下载和安装JDK

http://java.sun.com

下载和安装JDK


Java SE Development Kit 8 Update 131 (64-bit) - 定制安装
Java
ORACLE
从下面的列表中选择要安装的可选功能。您可以在安装后使用控制面板中的"添加/删除程序"实用程序更改所选择的功能
开发工具
源代码
公共 JRE
功能说明
Java SE Development Kit 8 Update 131 (64-bit), 包括 JavaFX SDK, 一个专用 JRE 以及 Java Mission Control 工具套件。它要求硬盘驱动器上有 180MB 空间。
安装到:
E:\Java\jdk1.8.0_131\
更改(C)...
< 上一步(B)
下一步(N) >
取消

安装目录

- bin: 可执行程序和工具
  - java编译器 javac.exe
  - java解释器 java.exe
  - java文档生成器 javadoc.exe
  - java调试器 jdb.exe
- db: 与数据库访问有关的文件
- demo：一些实例程序
- include: 与C相关的头文件
- jre: 运行环境相关文件
- lib：程序库

配置环境变量

我的电脑-属性-高级-环境变量

- **1. 设置JAVA_HOME**

  指代**JDK**安装路径

- **2. 设置Path**

%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;

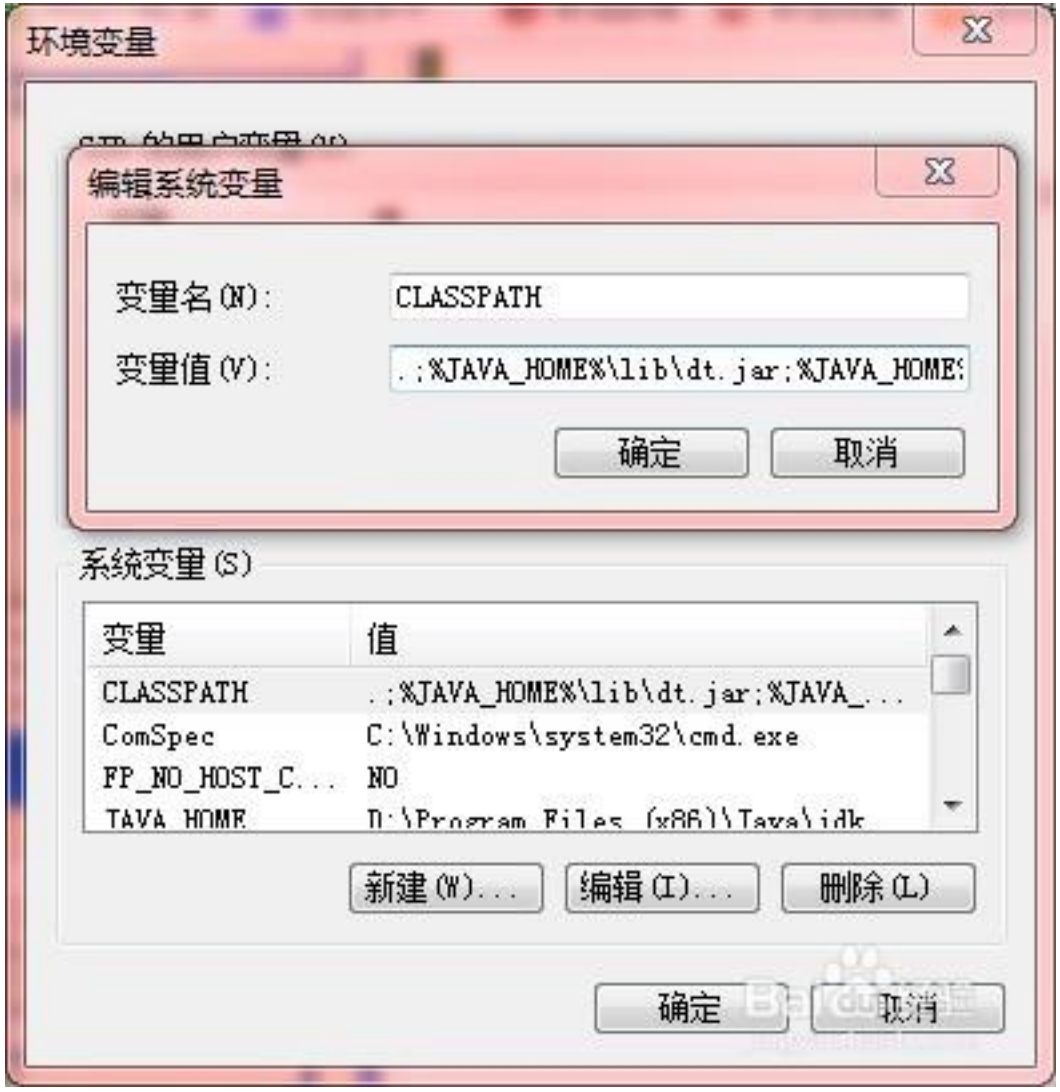
让系统识别**java**、**javac**等命令

配置环境变量

- 3. 设置CLASSPATH

.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;

- 告诉**jvm**要使用或执行的**class**放在什么路径上，便于**JVM**加载**class**文件，**.;**表示当前路径，**tools.jar**和**dt.jar**为类库路径
- **tools.jar**为工具类库，工具的实现，**RMI**等.
- **dt.jar:** 运行环境类库，主要是**swing**包(用于**GUI**)
- 基本类库和扩展类库由**jvm**自动加载，不需要在**classpath**中设置



程序的编辑

- 程序编辑：编辑器——文件名要与public class的类名一致。区分大小写。
- 程序编译——转换为字节码文件，扩展名.class . 其中包含 java虚拟机的指令。编译可以使用JDK工具javac.exe。
- 程序的运行——执行 .class文件中的指令的过程。

HelloWorld程序

```
// 源文件命名必须与定义的类名一致，区分大小写
// 以下源代码所在文件必须命名为HelloWorld.java
public class HelloWorld // 定义类
{
        public static void main (String[] args) // 主函数
        {
                // 打印字符串Hello World
                System.out.println (“Hello World”);
        }
}
```

编译和运行

- 打开命令行终端
- 到**HelloWorld.java**所在目录

javac HelloWorld.java //编译,得到HelloWorld.class
java HelloWorld //执行HelloWorld.class

Java-S18

该二维码7天内(2月26日前)有效，重新进入将更新