



欢迎学习Java程序设计

大仕老师

第1章 JAVA语言与面向对象的程序设计



- 1.1 Java语言简介
- 1.2 面向对象程序设计



Java语言简介



认识Java







Java历史与发展



Java是最热门的语言之一



数据来源: IEEE Spectrum

Language Rank	Types	Spectrum Ranking
1. Java	  	100.0
2. C	  	99.2
3. C++	  	95.5
4. Python	 	88.4

3 Billion Devices Run Java

Computers, Printers, Routers, Cell Phones, BlackBerry, Kindle, Parking Meters, Public Transportation Passes, ATMs, Credit Cards, Home Security Systems, Cable Boxes, TVs...

ORACLE®



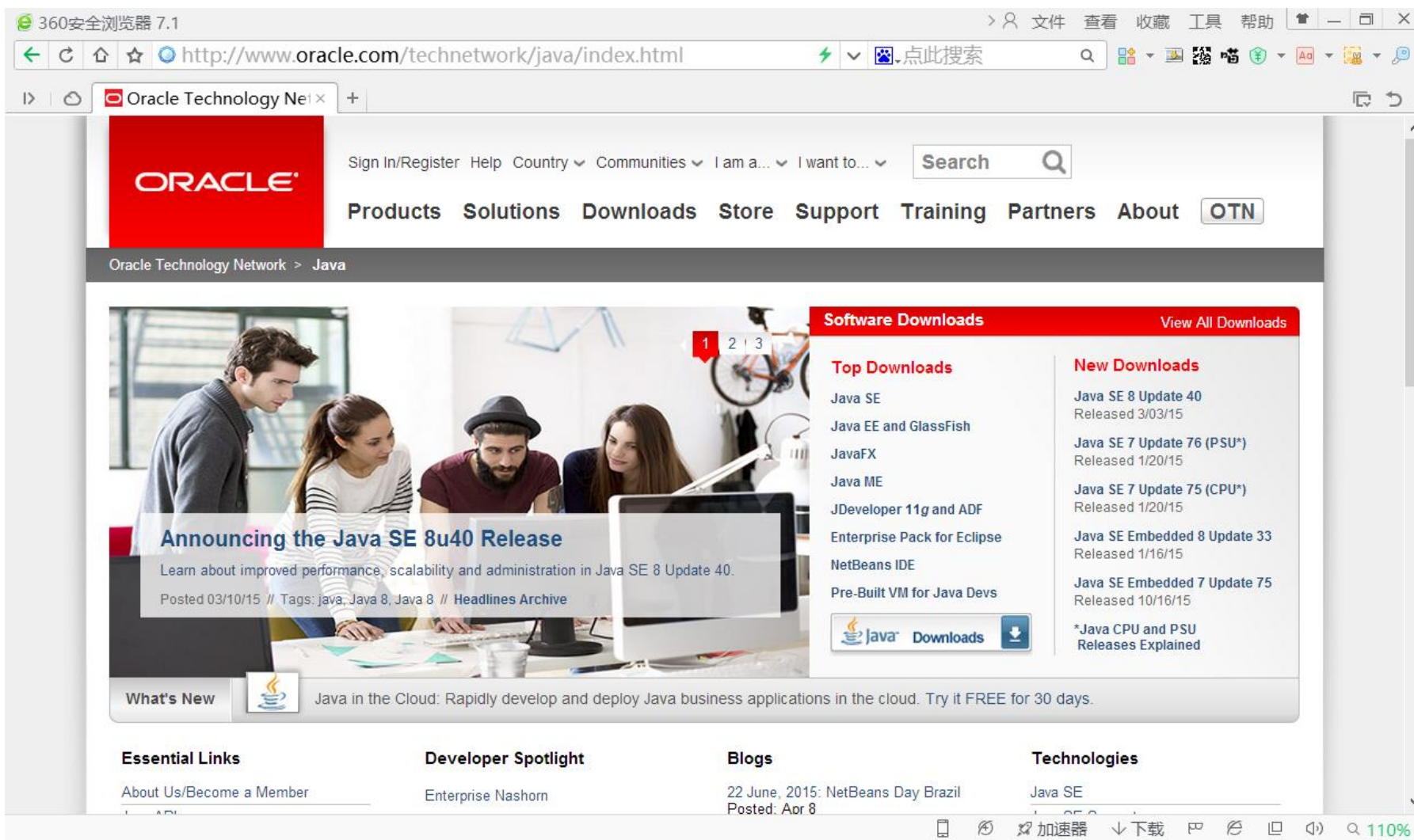
0.1 Java的发展历程

- Internet发展中的两次飞跃:
 - www
 - Java
- Java的出现
 - 1990年 SUN "Green"--开发家用电器软件
 - 1994年 Oak语言
 - 1995年, Java语言
 - SUN公布第一版编译器JDK1.0
 - Sun: Stanford University Network



James Gosling

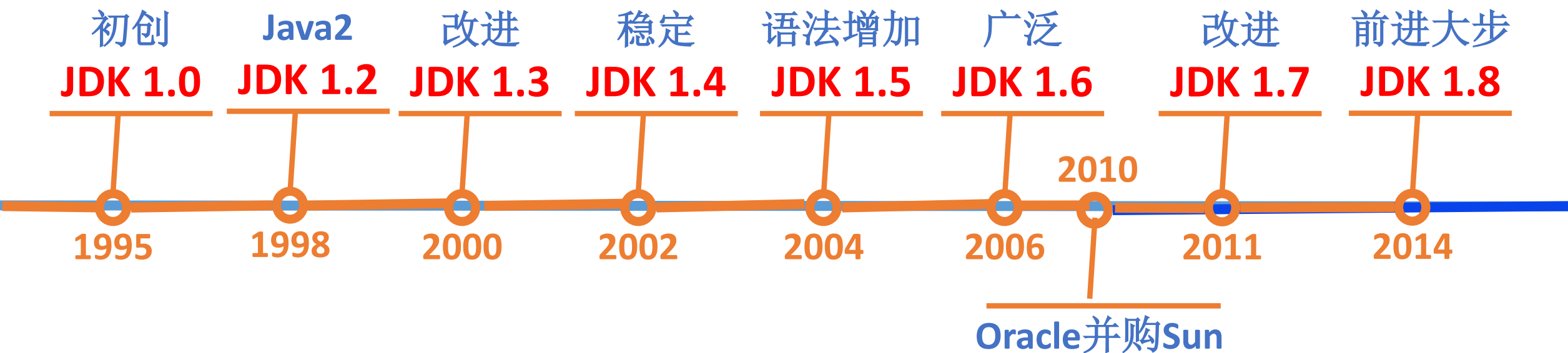






- Java分为三大平台
 - ▣ Java SE 标准版 (J2SE, Java 2 Platform Standard Edition)
 - ▣ Java EE 企业版 (J2EE, Java 2 Platform, Enterprise Edition)
 - ▣ Java ME 微型版 (J2ME, Java 2 Platform Micro Edition)
- Java的开发工具包 JDK (Java Development Kit)

Java发展历程





Java主要改进（可参见教材附录）

- JDK1.4
 - ▣ assert、logging、Java2D、NIO、正则表达式
- Java 5
 - ▣ 泛型、增强的foreach、自动装箱拆箱、枚举、可变长参数、静态引入、注记、printf、StringBuilder
- Java 6
 - ▣ Compiler API(动态编译)、脚本语言支持、WebService支持
- Java 7
 - ▣ 常量等的写法、带资源的try、重抛异常
 - ▣ <http://docs.oracle.com/javase/7/docs/technotes/guides/language/enhancements.html>
 - ▣ <http://www.oracle.com/technetwork/java/javase/jdk7-relnotes-418459.html>

Java的推动力：JCP与JSR



- <http://jcp.org>

- Welcome to jcp.org, home of the **Java Community ProcessSM (JCPSM)** Program. The JCP is the mechanism for developing standard technical specifications for Java technology. Anyone can register for the site and participate in reviewing and providing feedback for the **Java Specification Requests (JSRs)**, and anyone can sign up to become a JCP Member and then participate on the Expert Group of a JSR or even submit their own JSR Proposals.

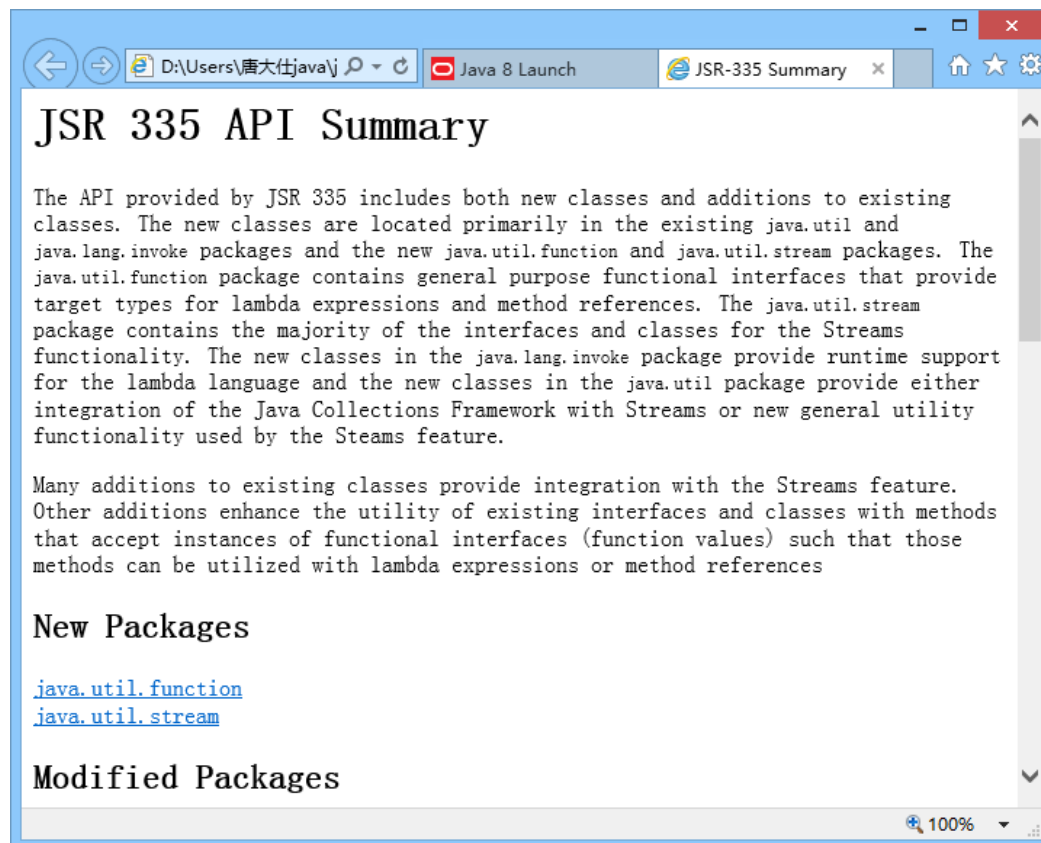


Final Releases

JSR 107, JCache - Java Temporary Caching API
JSR 308, Annotations on Java Types
JSR 310, Date and Time API
JSR 335, Lambda Expressions for the Java Programming Language
JSR 337, Java SE 8 Release Contents
JSR 360, Connected Limited Device Configuration 8
JSR 361, Java ME Embedded Profile



- JSR 335
 - ▣ Lambda Expressions for the Java Programming Language
- 这是Java8最重要的新特性



Java的特点



Java的特点、优势
及与其他语言的比较



Java语言的特点



What Is Java



- Java首先是一种编程语言
 - 它是一种面向对象的编程语言 (OOP)
 - 它的语法结构与C和C++语言极为相似
 - Java所用的语法十分简单

JAVA的特点



- 简单易学
- 面向对象
- 平台无关性
- 安全稳定
- 支持多线程

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.....

What Can Java Do



- 面向对象的应用开发
- 各类网络应用程序的开发
- 计算过程的可视化
- 动态画面/交互操作
- Internet的系统管理
- 数据库操作
- And more

- 示例 <file:///D:/Program%20Files/Java/jdk1.6/demo/jfc/Java2D/Java2Demo.html>

Java是C++--



- 无直接指针操作
- 自动内存管理
- 数据类型长度固定
- 不用头文件
- 不包含结构和联合
- 不支持宏
- 不用多重继承
- 无类外全局变量
- 无GOTO

Java 运行机制





Java运行机制

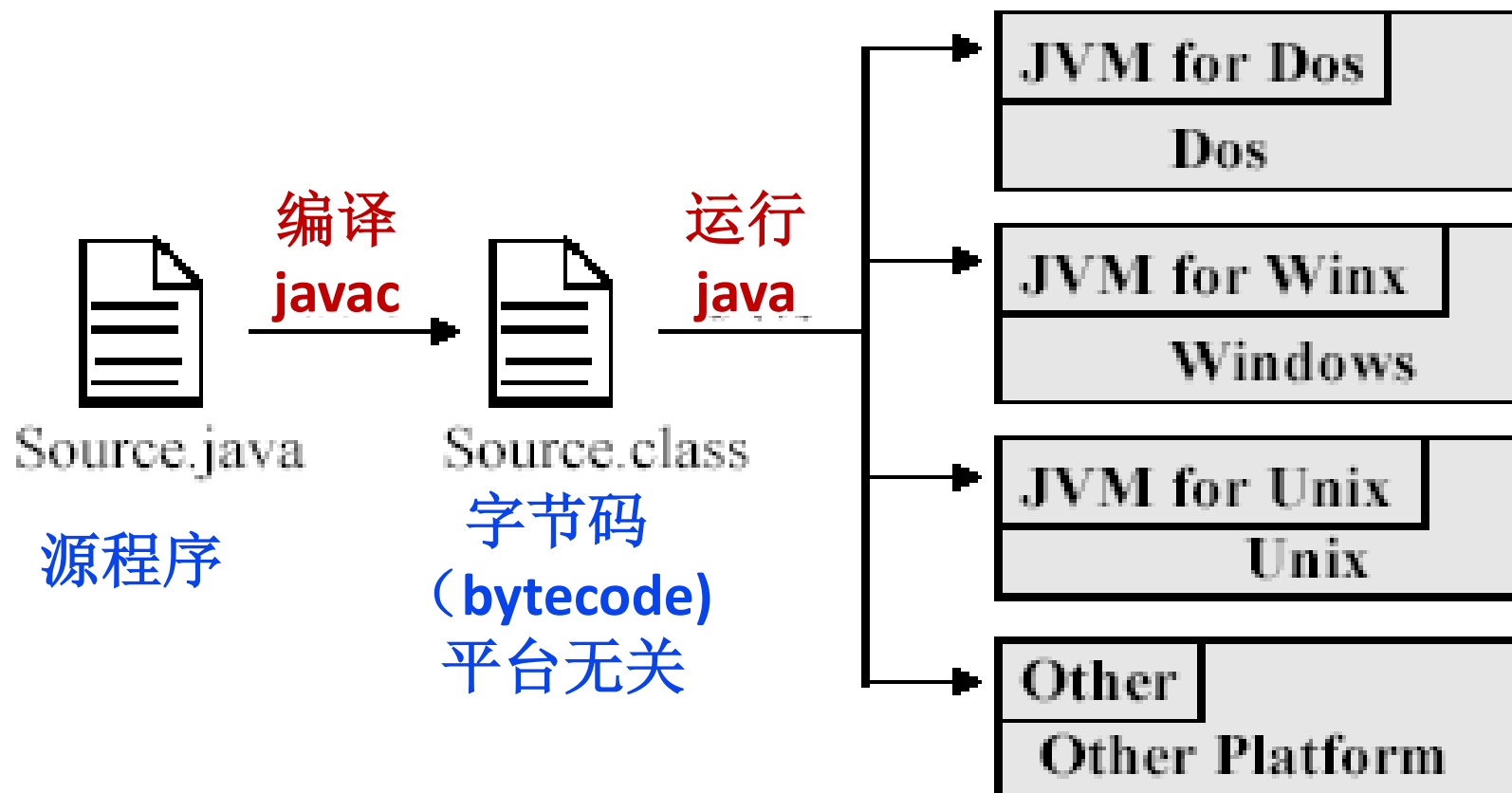


Java三种核心机制



- Java 虚拟机(Java Virtual Machine)
- 代码安全性检测(Code Security)
- 垃圾收集机制(Garbage collection)

Java程序的编译与运行



Java虚拟机 (Java Virtual Machine)



- 在一台计算机上由软件或硬件模拟的计算机。
- Java虚拟机(JVM)读取并处理经编译过的字节码class文件。
 - ▣ Java虚拟机规范定义了：
 - 指令集
 - 寄存器集
 - 类文件结构
 - 堆栈
 - 垃圾收集堆
 - 内存区域



Java运行环境(JRE)

- JRE (The Java Runtime Environment)
- $JRE = JVM + API (Lib)$
- JRE运行程序时的三项主要功能：
 - ▣ 加载代码：由class loader 完成;
 - ▣ 校验代码：由bytecode verifier 完成;
 - ▣ 执行代码：由 runtime interpreter完成。
- 理解
 - ▣ 为何Java是跨平台的
 - ▣ 为何Java是安全的

Java自动垃圾回收技术



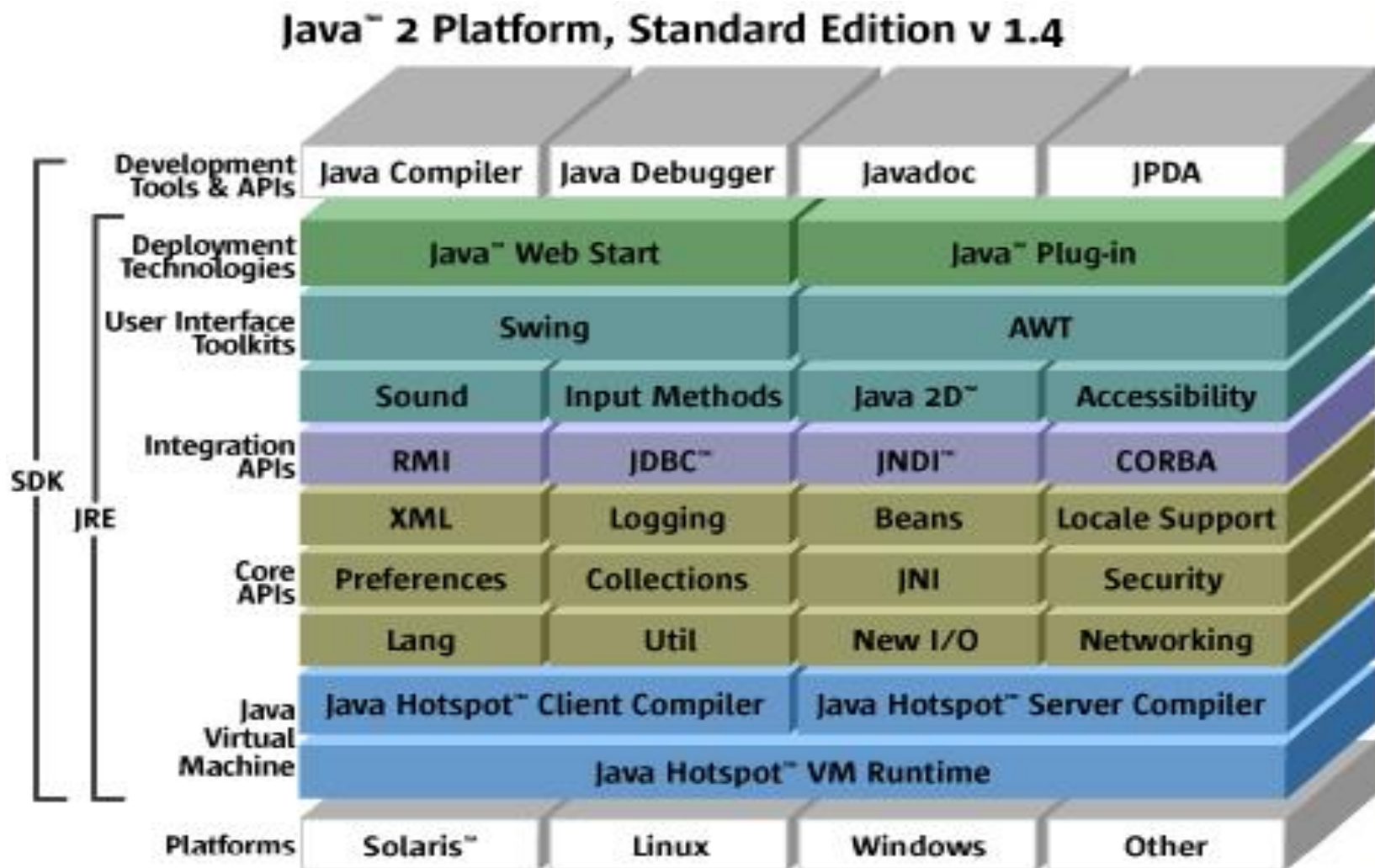
- 垃圾回收(garbage collection)
- 在C/C++ 等语言中，由程序员负责回收无用内存
- Java语言自动垃圾回收
 - 系统级线程跟踪存储空间的分配情况
 - 在JVM的空闲时，检查并释放那些可被释放的存储器空间
 - 程序员无须也无法精确控制和干预该回收过程



JDK (Java开发工具包)

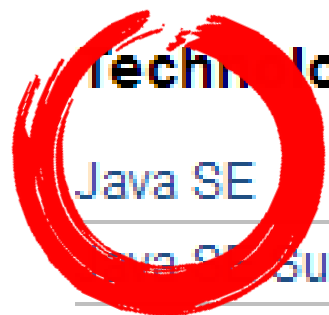
JDK=JRE+Tools

JRE=JVM+API





- 要开发Java程序，要下载JDK
 - <http://java.sun.com>
 - 可以只下载 JavaSE, 也可以下与NetBeans绑定的版本
- 注意：如果只运行Java程序，则下载JRE
 - <http://java.com>
- JDK提供的工具
 - java编译器 javac.exe
 - java执行器 java.exe
 - 文档生成器 javadoc.exe
 - java打包器 jar.exe
 - java调试器 jdb.exe



Technologies

Java SE

Java SE Support

Java SE Advanced & Suite

Java Embedded

Java EE

Java ME

JavaFX

Java Card

Java TV

Java DB

Developer Tools



1.2 面向对象程序设计

- 介绍面向对象的软件开发和面向对象程序设计中的基本方法和基本概念.



面向对象程序设计





对象 (object)

- 对象具有两方面的含义：
 - 在现实世界中：
 - 是客观世界中的一个实体
 - 在计算机世界中：
 - 是一个可标识的存储区域

类 (class)



- 类：具有共同属性和行为的对象集合

- 属性： 变量 (字段 field)
- 行为： 函数 (方法 method)

类

```
class Person{  
    int age;  
    String name;  
    void sayHello(){...}  
}
```

- 类与对象的关系

- 类是对象的抽象(模板)
- 对象是类的**实例**

对象

```
Person p = new Person();
```

- 注：类和对象有时都统称“对象”，为了明确起见，后者称为“对象实例”

面向对象的三大特征



封装性

继承性

多态性



模块化：将属性和行为封装在类中，程序定义很多类。

信息隐蔽：将类的细节部分隐藏起来

用户只通过受保护的接口访问某个类。

```
class Person{  
    private int age;  
    public int getAge(){ return age; }  
    public void setAge(int a){ age=a;}  
    String name;  
    void sayHello(){...}  
}
```



- 继承性

- 父类和子类之间共享数据和方法

父类

```
class Person{  
    int age;  
    String name;  
    void sayHello(){...}  
}
```

- 继承的好处

- 更好地进行抽象与分类

- 增强代码的重用率

- 提高可维护性

子类

```
class Student extends Person  
{  
    String school;  
    double score;  
    void meetTeacher(){ ... }  
}
```


多态性 (polymorphism)



- 多态

- 不同的对象收到同一个消息（调用方法）可产生完全不同的效果

- 实现的细节则由接收对象自行决定

- 例 `foo(Person p){ p.sayHello(); }`

- `foo(new Student());`

- `foo(new Teacher());`



面向对象设计思想的要点

- 认为客观世界由各种对象组成
- 程序的分析 and 设计都围绕着
 - 有哪些对象类
 - 每个类有哪些属性、哪些方法
 - 类之间的关系（继承、关联等）
 - 对象之间发送消息（调用方法）