



面向对象高级程序设计

主讲教师：刁丽娟

E-mail: lijuan_diao@126.com

教学目标与能力要求

◆教学目标:

灵活运用已学知识解决实际问题，通过已学知识，学习未学知识，养成良好的程序设计和应用的能力。

◆能力的三个层次:

1、知识

学习前人的知识

2、解决问题的方法

灵活运用学到知识

3、解决问题的思路

创造解决问题的方法

教学方法:

理论

举例

理论回顾

练习

讲解

※ 注重结合工程实践

※ 注重传授自学能力（读，写，查）

课程考核:

★ 本课程为考试课，考试形式是闭卷，

期末成绩占70%，平时成绩占30%。

★ 平时成绩包括

作业（实验） + 课上问题回答 + 上课出勤

教学资源:

预备知识:

英语 + 计算机基本操作 网络学习

网络资源:

Java官网: <http://www.oracle.com/technetwork/java>

程序员社区: <http://www.csdn.net/>

网易云课堂: <http://study.163.com>

世界一流大学免费课程: Coursera

程序员的天堂: <https://github.com/>

教学资源:

参考书:

- 1、Thinking in Java
- 2、唐友，郑萍主编 Java语言程序设计，应用型本科院校“十二五”规划教材 计算机类， 哈尔滨工业大学出版社，2013。
- 3、朱福喜 等编著. JAVA程序设计技巧与开发实例. 北京：人民邮电出版社，2012.
- 4、宛延恺 著. JAVA编程技巧. 北京：机械工业出版社,2008.



本课程的主要内容：

- 第一章 Java语言概述
- 第二章 Java语言基础
- 第三章 基本控制结构
- 第四章 数组、方法与字符串
- 第五章 类和对象
- 第六章 类的继承和多态
- 第七章 接口、抽象类和包
- 第八章 异常处理
- 第九章 集合类和多线程
- 第十章 输入输出流
- 第十一章 数据库编程



第1章 JAVA语言概述

本章目标

- 面向对象程序设计
- JAVA语言的发展史
- JAVA技术体系
- JAVA语言的特点
- JAVA的运行机制
- Java-Application
 - 开发第一个应用程序(application)

面向对象的程序设计

- 对象：是现实世界中某个具体的物理实体在计算机中的映射和体现。
- 类：描述对象的“基本原型”，定义一组对象所能拥有的共同特征，用以说明该组对象的能力与性质。
- 例如：定义电视机类，那么一台海信电视就是一个对象。

面向对象的程序设计

- 面向对象程序设计（Object-oriented programming, OOP）是一种程序设计范型，同时也是一种程序开发的方法。它将对象作为程序的基本单元，将程序和数据封装其中，以提高软件的重用性、灵活性和扩展性。对象指的是类的实例。
- 一个编程范型是一类典型的编程风格（可以对照方法学，是一个从事软件工程的一类典型的风格）。
- 一个编程范型提供了（同时决定了）一个程序员对程序执行的看法。例如，在面向对象编程中，程序员认为程序是一系列相互作用的对象，而在过程式编程中一个程序会被看作是一个无状态的函数计算的序列。

- 面向对象程序设计可以看作一种**在程序中包含各种独立而又互相调用的对象**的思想，这与传统的思想刚好相反：传统的程序设计主张**将程序看作一系列函数的集合，或者直接就是一系列对电脑下达的指令**。面向对象程序设计中的每一个对象都应该**能够接受数据、处理数据并将数据传达给其它对象**，因此它们都可以被看作一个小型的“机器”，即对象。

面向对象的程序设计

- 以电视机为例：以面向对象观点的观点来看的话，电视机是一个类的实例，他由显示屏，遥控器，电源等小的小的对象组成，这些小对象是他的属性；而开关电视机，选台等可以理解为电视机的动作。
- 这种好处是更有利于实现更大型系统，而且维护会更简单些。

面向过程的程序设计

- 面向**过程**的结构化程序设计方法就是**采用面向过程的方法来设计结构化程序**。结构化程序通常包含**一个主过程和若干个子过程**，其中每个子过程都描述了某一个问题的解决方法再由主过程自顶向下调用各子过程，来逐步解决整个问题。**整个执行过程是从主过程开始，再在主过程的结束语句处结束**。因此，结构化程序设计方法是一种数学思维或计算机思维方法，
- 面向过程程序设计通常采用自顶向下设计方法进行设计。在这种设计方法中，程序的每一个步骤直接的函数依赖关系是非常紧密的。要解决一个问题，就要通过一个个所设计好的函数步骤进行，每一个环节都是环环相扣的，都不能出错。一旦一个环节出错，将影响至整个问题的解决结果。面向过程设计语言以C语言、Pascal语言为主。

面向对象的程序设计三个基本特征

- 封装性
- 继承性
- 多态性

■ 封装

- ★ 封装最好理解了。封装是面向对象的特征之一，是对象和类概念的主要特性。
- ★ 封装，也就是把客观事物封装成抽象的类，并且类可以把自己的数据和方法只让可信的类或者对象操作，对不可信的进行信息隐藏。

■ 继承

- ★ 面向对象编程（OOP）语言的一个主要功能就是 “继承”。继承是指这样一种能力：它可以使用现有类的所有功能，并在无需重新编写原来的类的情况下对这些功能进行扩展。
- ★ 通过继承创建的新类称为“子类”或“派生类”。
- ★ 被继承的类称为“基类”、“父类”或“超类”。
- ★ 继承的过程，就是从一般到特殊的过程。

■ 多态

- ★ 多态是指一个名字具有多种语义，即指同一消息为不同对象所接受时，可以导致不同的操作，在面向对象编程语言中，就是指对象拥有同名，但不同的参数格式。
- ★ 多态的作用是什么呢？我们知道，封装可以隐藏实现细节，使得代码模块化；继承可以扩展已存在的代码模块（类）；它们的目都是为了——代码重用。而多态则是为了实现另一个目的——接口重用！实现“同一接口，多种方法”的功能。

Java实际应用

1. 行业和企业信息化

由于Sun、IBM、Oracle、BEA等国际厂商相继推出各种基于Java技术的应用服务器以及各种应用软件，带动了Java在金融、电信、制造等领域日益广泛的应用。如清华大学计算机系利用Java、XML和Web技术研制开发了多个软件平台，东方科技的Tong Web、金碟的Apusic、中创的Inforweb 等J2EE应用服务器及和佳ERP和宝信ERP等ERP产品，已在许多企业得到应用。

Java实际应用

2. 电子政务及办公自动化

东方科技、金碟、中创等开发的J2EE应用服务器在电子政务及办公自动化中也得到应用：如金碟的Apusic在民政部、广东省市工商局应用；东软电子政务架构EAP平台在社会保险、公检法、税务系统得到应用；中创的Inforweb等Infor系列中间件产品在国家海事局、山东省政府及中国建设银行、民生银行等金融系统应用；无锡永中科技基于Java平台开发的国产化集成办公软件“永中Office”也已在一些省市政府部门得到应用。

Java实际应用

3. 嵌入式设备及消费类电子产品

无线手持设备、通信终端、医疗设备、信息家电（如数字电视、机顶盒、电冰箱）、汽车电子设备等是今年以来比较热门的Java应用领域。在这方面的应用有中国联通CDMA 1X网络中基于Java技术的无线数据增值服务--UniJa。

现在Andriod手机中的应用，基本都是用Java语言编程。

Java实际应用

4. 辅助教学

在辅助教学方面，东南大学与中兴通讯公司开发了远程教学系统，用于本地网上教学、课后学习和异地远程教育，清华大学用Java进行了“计算机软件基础课”教学改革，分析研究Java教学软件BlueJ的汉化方案，电子科技大学应用Java RMI技术进行远程教育，西安电力高等专科学校采用Java技术，开发了交互式电站仿真系统，实现电站锅炉仿真、锅炉膛火焰仿真，为实现网上仿真进行有益探索。

JAVA语言的发展史-前传

- 在这个世界上，熟悉Java历史的人非常多，如果要问一个人Java是从哪年诞生的，也许大多数人都会回答是1995年。但事实上Java早在上个世纪90年代初就开始酝酿了。
- 1991年4月，Sun公司的James Gosling领导的绿色计划(Green Project)开始着力发展一种分布式系统结构，使其能够在各种消费性电子产品上运行。而Green项目组的成员一开始使用C++语言来完成这个项目，由于Green项目组的成员都具有C++背景，所以他们首先把目光锁定了C++编译器，Gosling首先改写了C++编译器，但很快他就感到C++的很多不足，需要研发一种新的语言Java来替代它，一杯飘香的咖啡成为了它的标志。



■ 图1 Java创始人 James Gosling博士

JAVA语言的发展史-幼年1995~1998

- Sun继Green项目后又经过了几年的研究，终于在1995年5月23日在SunWorld' 95上正式发布Java和HotJava浏览器。在同年，有很多公司先后获得了Java许可证，如Netscape在1995年8月，Oracle在1995年10月分别获得了Java许可证。Sun在1995年发布第一个Java版本后，于1996年1月宣布成立新的业务部门——JavaSoft部，这个分主要负责开发、销售并支持基于Java技术的产品。

JAVA语言的发展史-幼年1995~1998

- 在1995年Sun虽然推出了Java，但这只是一种语言，而要想开发复杂的应用程序，必须要有一个的强大的开发库支持还行。因此，Sun在1996年1月23日发布了JDK1.0。这个版本包括了两部分：**运行环境（即JRE）和开发环境（即JDK）**。在运行环境中包括了核心API、集成API，用户界面API，发布技术，Java虚拟机(JVM)五个部分。而开发环境还包括了编译Java程序的编译器（即javac）。在JDK1.0时代，JDK除了AWT（一种用于开发图形用户界面的API）外，其它的库并不完整。

JAVA语言的发展史-幼年1995~1998

- Sun在推出JDK1.0后，紧跟着，Sun在1997年2月18日发布了JDK1.1。JDK1.1相对于JDK1.0最大的改进就是为JVM增加了JIT(即时编译)编译器。JIT和传统的编译器不同，传统的编译器是编译一条，运行完后再将其扔掉，而JIT会将经常用到的指令保存在内容中，在下次调用时就不需要再编译了。这样JDK在效率上有了非常大的提升。

JAVA语言的发展史-幼年1995~1998

- Sun在推出JDK1.1后，接着又推出了数个JDK1.x版本。自从Sun推出Java后，JDK的下载量不断彪升，在1997年，JDK的下载量突破了220,000，而在1998年，JDK的下载量已经超过了2,000,000。
- 虽然在1998年之前，Java被众多的软件企业所采用，但由于当时硬件环境和JVM的技术原因，它的应用却很有限。当时Java主要只使用在前端的Applet以及一些移动设备中。然而这并不等于Java的应用只限于这些领域。在1998年是Java开始迅猛发展的一年。在这一年中，Sun将Java分成了J2EE、J2SE和J2ME。标志着Java已经吹响了向企业、桌面和移动3个领域进军的号角。

JAVA语言的发展史-青少年期1998~2004

- 到1998年，Java已经走过了3个年头。从JDK1.0到JDK1.1.8。JDK1.x经过了9个小版本的发展，已经初具规模。至此，它已经走出了摇篮，可以去独闯世界了。
- 在1998年12月4日。Sun发布了Java的历史上最重要的一个JDK版本：JDK1.2。这个版本标志着Java已经进入Java2时代。这个时期也是Java飞速发展的时期。
- 在Java2时代Sun对Java进行了很多革命性的变化，而这些革命性的变化一直沿用到现在，对Java的发展形成了深远的影响。

JAVA语言的发展史-青少年期1998~2004

- JDK1.2自从被分成了J2EE、J2SE和J2ME三大块，得到了市场的强烈反响。不仅如此，JDK1.2还对它的API分成了三大类。

1、核心API

由Sun公司制定的基本的API，所有的Java平台都应该提供。这就是我们平常所说的Java核心类库。

2、可选API

这是Sun为JDK提供的扩充API，这些API因平台的不同而不同。

3、特殊API

用于满足特殊要求的API。如用于JCA和JCE的第三方加密类库。

JAVA语言的发展史-壮年时期2004~至今

- 虽然从JDK1.4开始，Java的性能有了显著的提高，但Java又面临着另一个问题，那就是复杂。
- 虽然Java是纯面向对象语言，但它对一些高级的语言特性（如泛型、增强的for语句）并不支持。而且和Java相关的技术，如EJB2.x，也由于它们的复杂而很少有人问津。也许是Sun意识到了这一点。因此，在2004年10月，Sun发布了我们期待已久的版本：JDK1.5，同时，Sun将JDK1.5改名为J2SE5.0。和JDK1.4不同，JDK1.4的主题是性能，而J2SE5.0的主题是易用。Sun之所以将版本号1.5改为5.0，就是预示着J2SE5.0较以前的J2SE版本有着很大的改过。
- 之后又推出了6.0 7.0 8.0等等。

JAVA语言的发展史

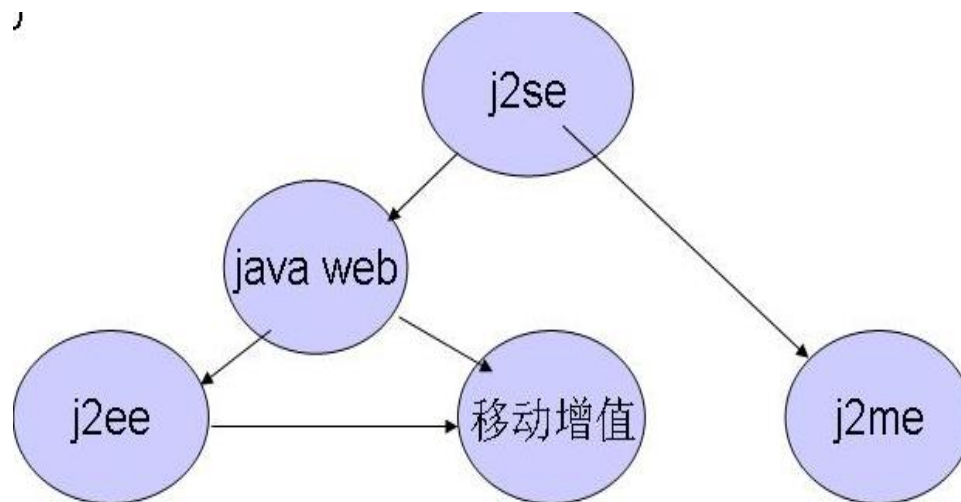
- ◆95年 在SunWorld大会，Sun公司正式推出JAVA
- ◆96年 发布JDK1.0，包括运行环境和开发环境
- ◆98年 发布JDK1.2，标志java已进入JAVA2时代
- ◆JDK1.3
- ◆JDK1.4
- ◆2004年，发布JDK1.5，更名为JAVA5.0
- ◆2006年，发布JDK6.0
- ◆2011年，发布JDK7.0
- ◆目前JDK16.0

JAVA语言的发展史

◆JDK版本

- 1.0: 212类和8个包, 奠定语法和基础类
- 1.1: 514类和23个包, 引入内部类
- 1.2: 1520类和59个包, 引入类集和Swing
- 1.3: 1842类和76个包, 改善性能、引入动态代理和JNDI、Java Sound等API
- 1.4: 2991类和135个包, 引入断言assert和新I/O、正则表达式、XML、JAAS等API
- 5.0(1.5): 3562类和166个包
- ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

JAVA技术体系



◆ J2SE: Java Platform, Standard Edition

- Java技术的核心，提供基础Java开发工具、执行环境和应用程序接口（API），主要用于桌面应用程序的开发

◆ J2EE: Java Platform, Enterprise Edition

- 主要应用于网络程序和企业应用的开发。JavaEE是JavaSE基础上的扩展。

◆ J2ME: Java Platform, Micro Edition

- 主要应用于手持设备开发

JAVA语言的特点

- ◆一种面向对象的语言
- ◆一种与平台无关的语言
- ◆一种健壮的语言，吸收C/C++的特点，但去掉了其影响程序健壮性的部分（如：指针、内存的申请及释放）

JAVA语言的特点

1. 跨平台性

所谓的跨平台性，是指软件可以不受计算机硬件和操作系统的约束而在任意计算机环境下正常运行。这是软件发展的趋势和编程人员追求的目标。之所以这样说，是因为计算机硬件的种类繁多，操作系统也各不相同，不同的用户和公司有自己不同的计算机环境偏好，而软件为了能在这些不同的环境里正常运行，就需要独立于这些平台。

而在Java语言中，Java自带的虚拟机很好地实现了跨平台性。Java源程序代码经过编译后生成二进制的字节码是与平台无关的，但是可被Java虚拟机识别的一种机器码指令。Java虚拟机提供了一个字节码到底层硬件平台及操作系统的屏障，使得Java语言具备跨平台性。

2. 面向对象

面向对象是指以对象为基本粒度，其下包含**属性和方法**。对象的说明用属性表达，而通过使用方法来操作这个对象。面向对象技术使得应用程序的开发变得简单易用，节省代码。**Java**是一种面向对象的语言，也继承了面向对象的诸多好处，如代码扩展、代码复用等。

3. 安全性

安全性可以分为四个层面，即语言级安全性、编译时安全性、运行时安全性、可执行代码安全性。

语言级安全性指**Java**的数据结构是完整的对象，这些封装过的数据类型具有安全性。编译时要进行**Java**语言和语义的检查，保证每个变量对应一个相应的值，编译后生成**Java**类。运行时**Java**类需要类加载器载入，并经由字节码校验器校验之后才可以运行。**Java**类在网络上使用时，对它的权限进行了设置，保证了被访问用户的安全性。

4. 多线程

多线程在操作系统中已得到了最成功的应用。多线程是指允许一个应用程序同时存在两个或两个以上的线程，用于支持事务并发和多任务处理。**Java**除了内置的多线程技术之外，还定义了一些类、方法来建立和管理用户定义的多线程。

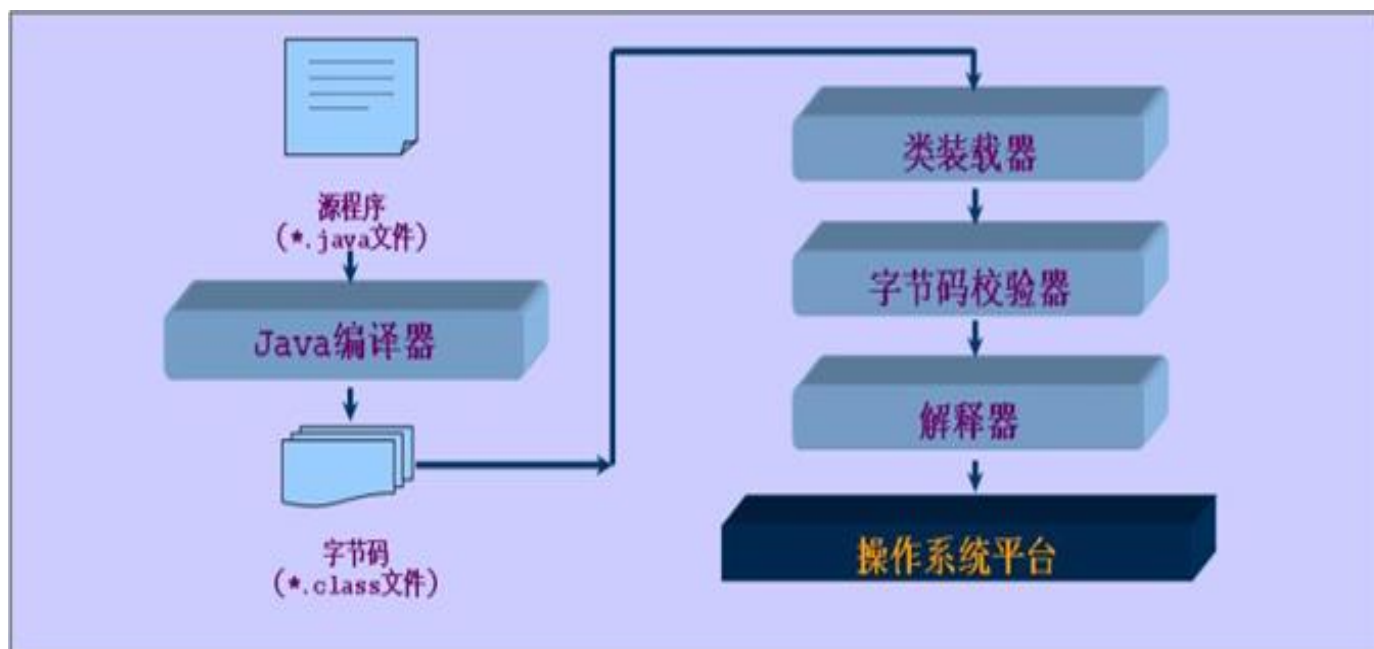
5. 简单易用

Java源代码的书写不拘泥于特定的环境，可以用记事本、文本编辑器等编辑软件来实现，然后将源文件进行编译，编译通过后可直接运行，通过调试则可得到想要的结果。

JAVA的运行机制

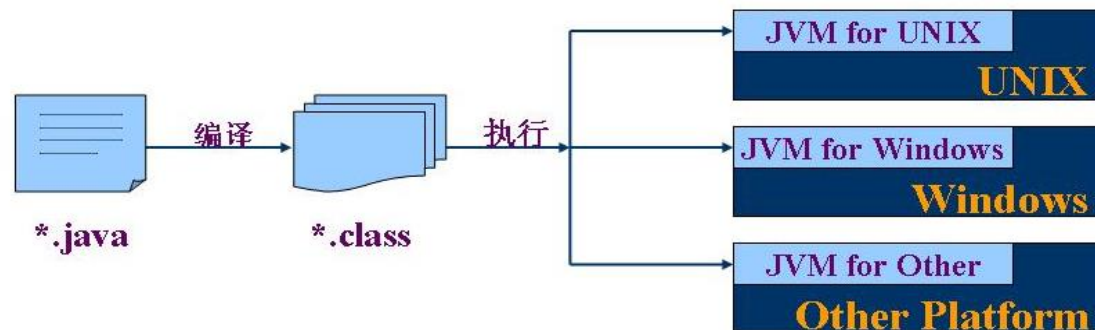
◆两种核心机制

- JAVA虚拟机 (Java Virtual Machine)
- 垃圾收集机制 (Garbage Collection)



JAVA虚拟机

- ◆ java虚拟机是软件模拟的计算机
- ◆ 一个以字节码为机器指令的CPU
- ◆ 针对不同的软硬件平台有不同的实现
- ◆ 屏蔽了底层平台的不同差别，实现了“编译一次，到处运行”



垃圾收集机制

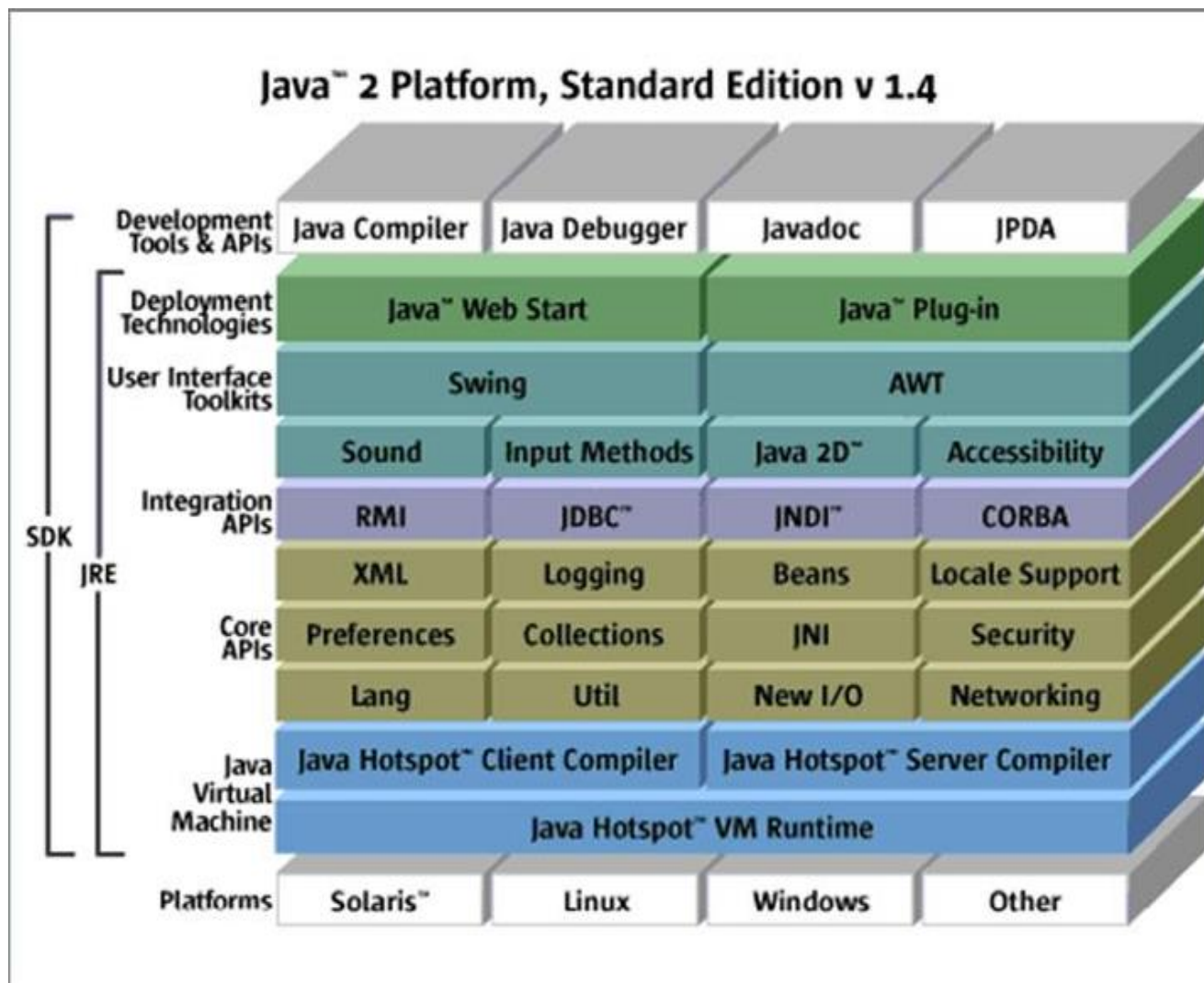
- ◆ 无用内存自动回收-垃圾收集
- ◆ 在C/C++语言中是由程序员负责无用内存回收
- ◆ 在java运行环境中，始终存在着一个系统级的线程，专门跟踪内存的使用情况，定期检测出不再使用的内存，并进行自动回收，避免了内存的泄露，也减轻了程序员的工作量
- ◆ 垃圾收集在Java程序运行过程中自动进行，程序员无法精确控制和干预

开发第一个应用程序

J2SDK & JRE

Software
Development
Kit
(软件开发包)
Java Runtime
Environment
(Java运行环境)

开发需要JDK
用户只需JRE



开发第一个应用程序

◆ 配置java开发环境

- 下载并安装最新版本J2SDK

- <https://www.oracle.com/java>

- 设置Windows环境变量

配置path和classpath

- **path:** windows系统执行命令时要搜寻的路径，用来指定java虚拟机（JVM）所在目录。

- **classpath:** java在编译和运行时要找class的路径，也就是类编译后的字节码文件或要用到的类文件（.jar文件）所在的目录。

开发第一个应用程序

环境变量的设置

JDK包安装完成后，需要设置环境变量。用鼠标右键单击桌面上的图标“计算机”，选择“属性”项，出现标题为“系统特性”的对话框，点击“高级”标签，可以看见有一个“**环境变量**”按钮，如图1.1所示。

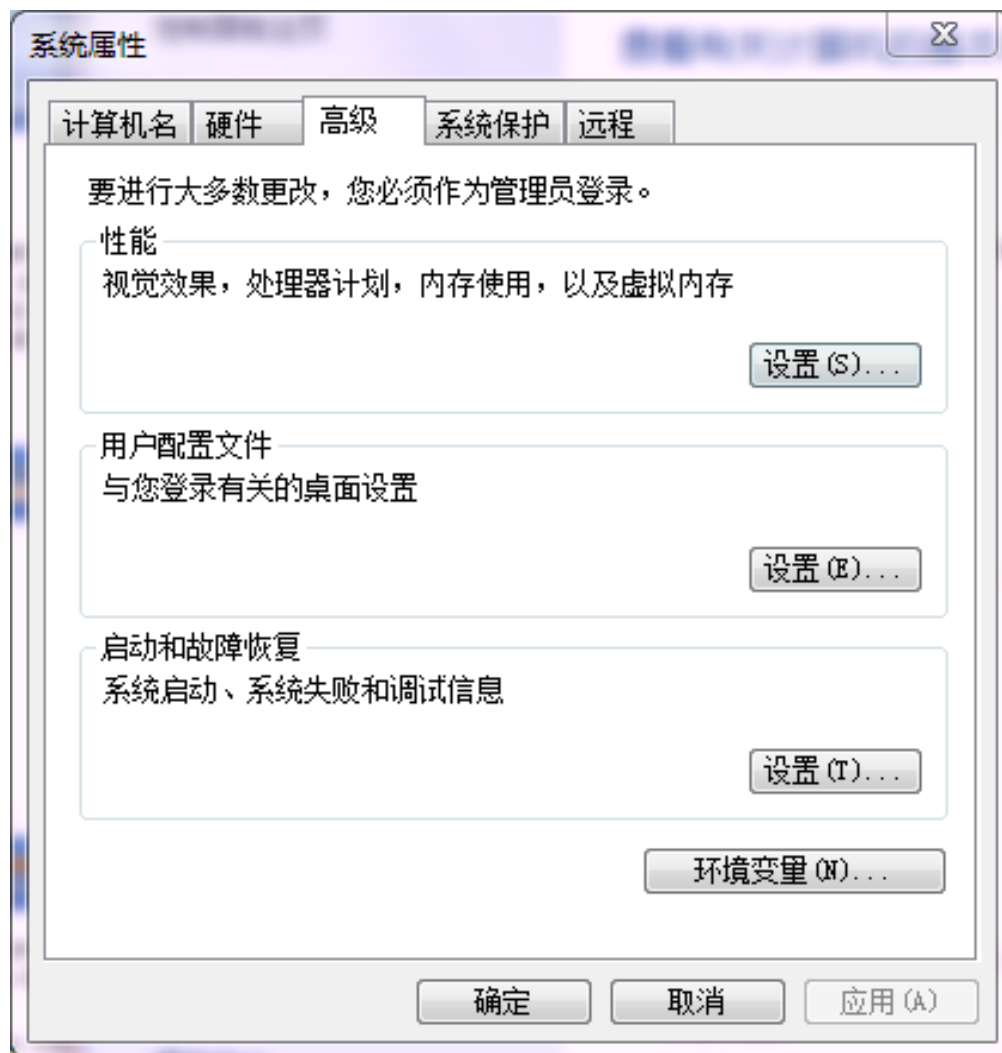


图1.1 “系统特性”对话框

单击“环境变量”按钮，可以看见本机环境变量，如图1.2所示。上面为用户变量，下面为系统变量，随着操作系统或用户环境的不同，变量名、值有所不同。



图1.2 “环境变量”对话框

开发第一个应用程序

注意事项:

- 向PATH变量中追加内容时，先写一个分号，再写java命令的路径
- 确定所写的java命令路径中，确实能找到javac.exe以及java.exe两个可执行文件
- . 表示当前路径；
- PATH使得系统可以在任何路径下识别java命令。
- CLASSPATH为java加载类(class or lib)路径，只有类在classpath中，java命令才能识别。

开发第一个应用程序

◆测试环境变量正确性

- 进入DOS命令窗口：
 - 开始→运行→输入cmd点选确定
- 验证java命令是否可以执行：
 - 在命令窗口输入javac或java，
 - 如果执行后出现有关命令参数的说明，则说明PATH环境变量设置成功，否则需再重新确认PATH环境变量的设置。

■ JAVA是分两部分的：一个是编译，一个是运行。

- 1、javac：负责的是编译的部分，当执行javac时，会启动java的编译器程序。对指定扩展名的.java文件进行编译。生成了jvm可以识别的字节码文件。也就是class文件，也就是java的运行程序。
- 2、java：负责运行的部分，将启动jvm，加载运行时所需的类库，并对class文件进行执行。



javac.exe： 将.java文件编译成.class文件
java.exe： 解释执行.class文件

开发第一个应用程序

◆开发HelloWorld应用程序

- 选择合适的文本编辑器
- 集成的开发环境

◆打开记事本，编写以下代码：

例子1.1

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

开发第一个应用程序

◆编译

- 运行cmd，打开windows窗口，切换到对应的.java源文件所在的位置。
- `javac HelloWorld.java`

◆运行

- `java HelloWorld`

开发第一个应用程序

◆注意点:

- JAVA源文件以” java” 为扩展名。源文件的基本组成部分是类，如本例中的HelloWorld类
- 一个源文件中最多只能有一个public类。其它类的个数不限，如果源文件中包含一个public类，它必须按该类命名。
- Java应用程序的执行入口是main()方法。它有固定的书写格式：

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

- JAVA语言严格区分大小写;JAVA方法有一条条语句构成，每个语句以分号结束。

总结

- ◆ 了解Java语言的发展历史
- ◆ 掌握Java语言的运行机制
- ◆ 掌握path和classpath的作用
 - path: 编译java源程序的javac.exe 和用来执行 *.class 文件的 java.exe 等命令的路径
 - classpath : 类编译后的字节码文件或要用到的类文件(.jar 文件) 所在的目录
- ◆ 掌握第一个Java程序, 验证java程序的运行原理



北華航天工業學院
NORTH CHINA INSTITUTE OF AEROSPACE ENGINEERING

谢谢！