Java语言与网络编程

计算机学（北京）中国矿业大学

计算机学（北京）中国矿业大学

Java语言篇

- 第1章 Java简介
- 第2章 Java语言基础
- 第3章 程序流程控制
- 第4章 类与对象
- 第5章 消息、继承与多态

Java语言篇

- 第6章 数组
- 第7章 字符串类
- 第8章 链表

Java语言篇

- 第9章 文字与图形GUI设计
- 第10章 常用组件GUI设计
- 第11章 高级组件GUI设计

Java语言篇

- 第12章 异常处理
- 第13章 多线程
- 第14章 输入与输出

Web编程篇

- 第1章 Web应用开发概述
- 第2章 Java Web应用开发与运行环境
- 第3章 JSP运行机制与基本语法
- 第4章 JavaBean与设计模式
- 第5章 JSP数据库应用开发与实例

课程考核

- 考核方式：闭卷考试
- 成绩评价：平时成绩*40% + 考试成绩*60%

考勤（15%） 上机实践+实验报告（25%）

考勤缺3次及以上，最终成绩不合格

第一章 Java简介

- 1.1 Java产生
- 1.2 Java特点
- 1.3 Java基本编程环境
- 1.4 Java程序开发过程

1.1 Java产生

- 1990年，Sun公司启动一个James Gosling（Java创始人）项目。尝试使用C++开发用于消费电器中的软件。
- Gosling用了一种新语言Oak来解决这个问题。
- Oak保留了熟悉的C++语法。
- 当Oak成熟时，因特网也正处于戏剧性增长时期，Sun公司的开发小组认识到Oak非常适合Internet编程。
- 1994年，他们完成了一个用Oak编写的早期Web查看器，称为WebRunner，后改名为HotJava。

1.1 Java产生

- Java让人联想到印度尼西亚有个重要的盛产咖啡的岛屿，开发人员为这种新的语言起名为Java，其寓意是为世人端上一杯热咖啡。

- 1995年，Oak正式更名为Java，并在SunWorld95中发布。

- 此后，Java知名度如日中天，甚至在1996年1月Java编译器第一版本发布之前，Java已成为Internet发行的行业标准。

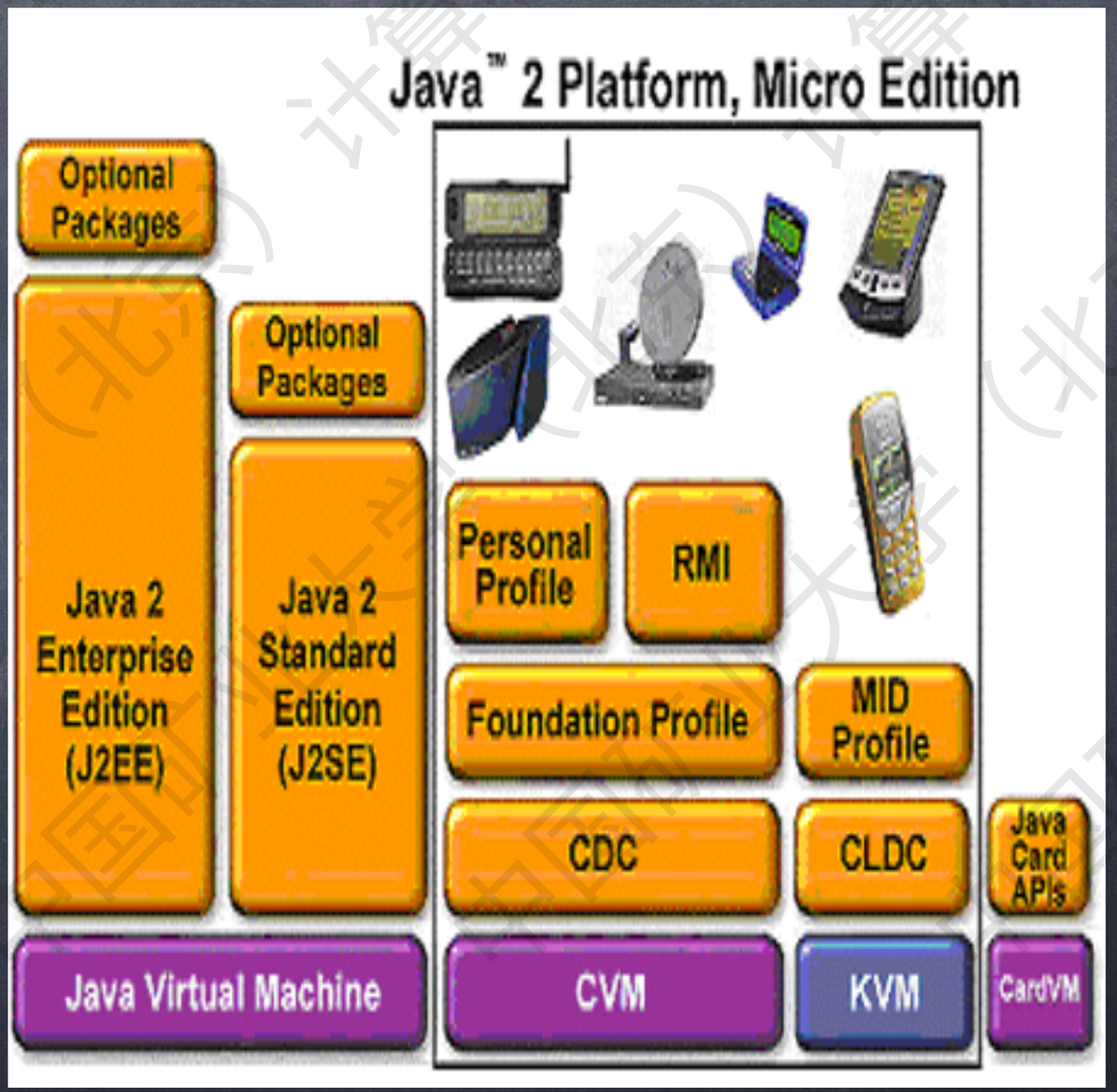
1.1 Java产生

预言：

- Java语言的出现，将会引起一场软件革命。
- 不久的将来，全世界90%的程序代码将用Java语言书写或改写。
- 在工业领域，Java与C++平起平坐或取代C++。

1.1 Java产生

- 目前常用的Java版本：
- J2EE (Java 2 Platform Enterprise Edition)
- J2SE (Java 2 Platform Standard Edition)
- J2ME (Java 2 Platform Micro Edition)



1.2 Java特点

- Sun公司在"Java白皮书"中对Java的定义如下：

Simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multi-thread, and dynamic language.

Java是一种具有"简单、面向对象的、分布式、解释型、健壮、安全、与体系结构无关、可移植、高性能、多线程和动态执行"等特性的语言。

特点之一 简单性(simple)

- Java语言的简单性体现在以下四个方面：

1. Java的风格类似于C++。对C++程序员而言，Java是非常熟悉的风格；从某种意义上讲，Java语言本身是C++的一个变种。
2. Java摒弃了C++中不安全的地方。如指针、内存管理。
3. Java提供自动内存垃圾搜集机制。减轻了编程人员进行内存管理的负担，有助于减少软件错误。
4. Java是完全面向对象的，同时还提供了大量的扩充类库。

特点之二 面向对象(object-oriented)

- 所有面向对象的编程语言至少具备四个特点：

1. 封装性：必须有模块化的性质以及信息隐藏能力。
2. 多态性：不同的对象对同一种信息，可按照对象本身的性质加以回应。
3. 可继承性：可定义一套对象之间的层次关系，下层对象继承了上层对象的特征，借此实现程序代码重复利用。
4. 动态联播性：一旦对象生成以后，使用此对象只需传递信息调用。只在程序执行时，锁定需要的对象，这样可使程序设计具有更大的灵活性。

特点之三 分布式(distributed)

- 分布式包括数据分布和操作分布。

  数据分布是指数据可以分散存放于网络上的不同主机上。

  对于数据分布，Java提供一个URL对象(也即网址)，可以访问网络上的对象，其访问方式与访问本地文件系统几乎相同。

  操作分布是指把计算分散由不同主机进行处理。

  对于操作分布，Java可将运算工作交由服务器进行，从而提高整个系统的执行效率，也避免了瓶颈制约。

特点之四 解释型(interpreted)

字节码

一种格式统一的，含有两个字节的二进制文件，可在多种平台上运行的程序。

特点之五 健壮性(Robust)

- Java在编译和运行程序，都要对可能出现的问题进行检查，以消除错误的产生，提供自动垃圾收集来进行内存管理，防止程序员在管理内存时容易产生的错误。

- 通过集成的面向对象的异常处理机制，在编译时，Java提示出可能出现但未被处理的异常，帮助程序员正确地进行选择以防止系统的崩溃。

- Java在编译时，可捕获类型声明中的常见错误，防止动态运行时不匹配问题的出现。

特点之六 安全性(secure)

- Java不支持指针

一切对内存的访问都必须通过对象的实例变量来实现，这样就防止程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

- Java安全模型有三个基本成分

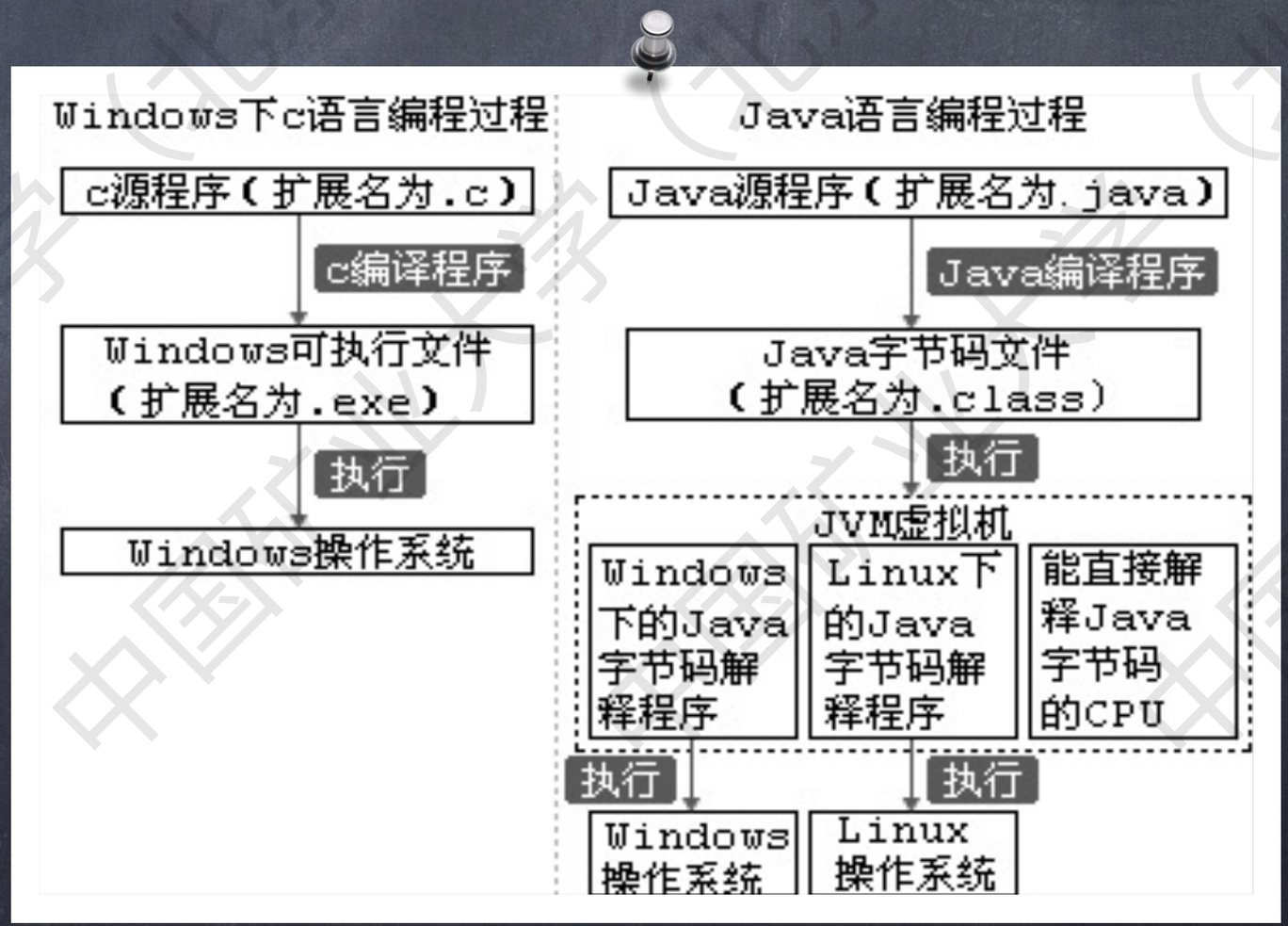
1. 字节码校验器：确保Java程序的编译正确，产生的字节码不出错。
2. 类装载器：负责把来自网络的类装载到其单独的内存区域，避免应用程序之间的相互干扰或破坏。
3. 安全管理机制：确定虚拟机在什么条件下可以活动。

特点之七 中立性(architecture-neutral)

- Java程序被编译成与体系结构无关的字节码，Java解释器得到字节码后，对其进行转换，使其可在不同平台上运行。

"Write Once, Run Anywhere"

一次编译，到处运行



特点之八 可移植性(portable)

1. Java编译人员在进行软件开发时，不必考虑软件运行平台，开发的源代码是可移植的。
2. 源代码经过编译之后形成的二进制代码—字节码，而不管这种字节码是在何种平台上生成的，也即字节码是可移植的。另外，数据类型是绝对长度。
3. 可移植性体现在Java的运行环境上。Java编译器是用Java语言本身编写的，Java的整个运行环境体现了一个定义良好的可移植性接口。

特点之九 高性能(high-performance)

Java作为一种解释型语言，其速度不会超过编译语言C，但远远超过交互式语言。

和其他解释执行的语言如BASIC等不同，Java字节码的设计，使之能很容易地直接转换成对应于特定CPU的机器码，从而具有较高的性能。

特点之十 多线程(multi-thread)

- 多线程概念可近似得理解为多任务，Java可以把一个程序分成多个任务，以便任务易于完成，最大限度利用CPU资源。

- 多线程机制使应用程序能够并行执行，而且同步机制保证对共享数据的正确操作。

- 通过多线程，程序设计者可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制，容易实现网络上的实时交互行为。

特点之十一 动态性(dynamic)

- Java自身的设计使其适合于一个不断发展的环境。在Java类库中可以自由的加入新的方法和实例变量，而不会影响用户程序执行。

- Java通过接口来支持多重继承使之比严格的类继承更灵活、更易扩展。

Java语言特点
-与C/C++的区别

- 不再有全局变量
- 不再有#include和#define等预处理功能
- 不再有structure、union、typedef等
- 不再有函数、不再有指针、不再有多重继承
- 不再有goto语句
- 不再有操作符重载
- 取消自动类型转换，要求强制转换
- 自动内存管理

1.3 Java编程环境

- 1.3.1 Java软件开发工具包JDK (Java Develop Kit)

SUN公司为所有的Java程序员提供免费的Java开发运行环境。

表1.1 Java SDK 开发工具集

工具名称	说　　明
Javac	Java编译器，用于将Java源程序编译成字节码
Java	Java解释器，用于解释执行Java字节码
appletviewer	小应用程序浏览器，用于测试和运行Java applet程序
Javadoc	Java文档生成器
Javap	Java类文件反汇编器
Jdb	Java调试器
Javah	C文件生成器，利用此命令可实现在Java类中调用C++代码

Java SE Downloads

Java Platform (JDK) 8u25

JDK 8u25 & NetBeans 8.0.1

Java Platform, Standard Edition

Java SE 8u25

This release includes important security fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release.
Learn more

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations
- Readme Files

JDK DOWNLOAD

Server JRE DOWNLOAD

JRE DOWNLOAD

Java SDKs and To

- Java SE
- Java EE and Glassfis
- Java ME
- Java Card
- NetBeans IDE
- Java Mission Control

Java Resources

- Java APIs
- Technical Articles
- Demos and Videos
- Forums
- Java Magazine
- Java.net
- Developer Training
- Tutorials
- Java.com

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter (tick the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK MD5 Checksum

Looking for JDK 8 on ARM?
JDK 8 for ARM downloads have moved to the JDK 8 for ARM download page.

Java SE Development Kit 8u25

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ **Accept License Agreement** ◉ **Decline License Agreement**

中国矿业大学 (北京) 计算机系

安装JDK及帮助文档

1. 可以选择安装到任意的硬盘驱动器上，例如安装到D:\jdk1.7目录下。

2. 通常在JDK目录下有bin、demo、lib、jre等子目录，其中bin目录保存了javac、java、appletviewer等命令文件，demo目录保存了许多java的例子，lib目录保存了java的类库文件，jre保存的是java的运行时环境。

3. JDK的安装程序中并不包含帮助文档，因此也必须从网站上下载。通常放在JDK所在目录的docs子目录下面。用浏览器打开docs子目录下的index.html文件就可以阅读所有的帮助文档。

E
E
E
E Support
E Advanced & Suite
mbedded
B
er
ard
/
Java
unity
agazine

Java SE Documentation

A wealth of information is available to help you learn and use Java platform technology.

Visit the Java Platform Standard Edition Technical Documentation site for information on new features and enhancements, Java Tutorials, Developer Guides, API documentation, and much more.

Product License, Commercial Features and Terms

- Binary Code License for Java SE Platform Products (HTML, PDF)
- Overview of the Java SE Product Editions and the Commercial Features available in each edition (HTML, PDF)

README Files

- Java SE and JavaFX README files (HTML)
- Java SE, JavaFX and JRockit THIRDPARTYLICENSEREADME Files (HTML)

README Archives

- README and THIRDPARTYLICENSEREADME text for archived versions of Java SE and JavaFX.

Java SDKs and
Java SE
Java EE and Glas
Java ME
Java Card
NetBeans IDE
Java Mission Cont
Java Resour
Java APIs
Technical Articles
Demos and Video
Forums
Java Magazine
Java.net
Developer Training
Tutorials
Java.com

配置类路径

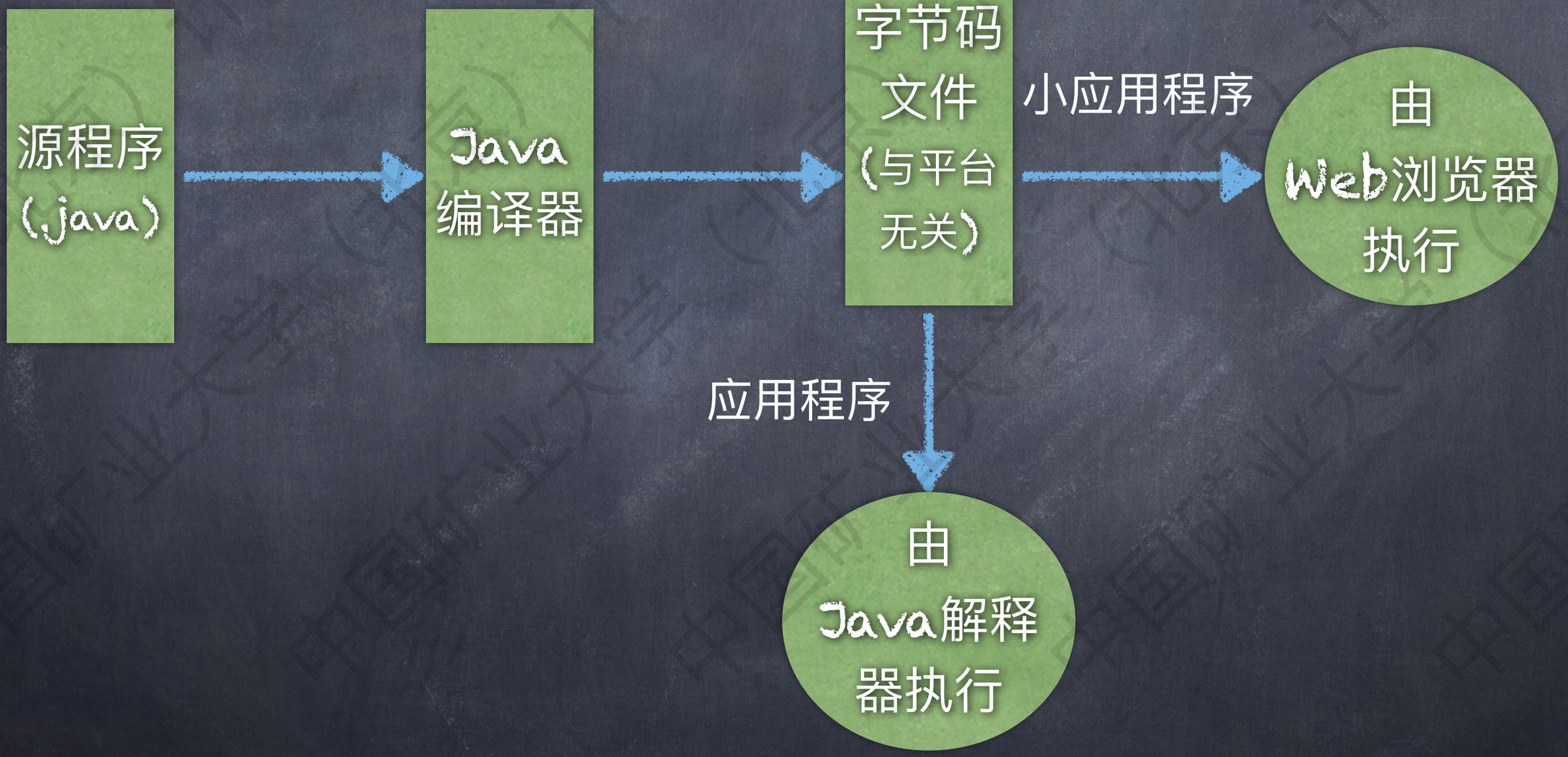
在安装完JDK之后，必须配置类路径classpath和环境变量path，JDK才能正常工作。

右键单击“我的电脑”-选择“属性”-选定“系统特性”-选择“高级”-选择“环境变量”，在“环境变量”窗口中编辑classpath和path。

```
classpath= .;d:\jdk1.8\lib\tool.jar;

path= %path%;d:\jdk1.8\bin;
```

1.4 Java程序开发过程



1. 源程序：Java源程序是以.java为后缀的简单文本文件。

2. 字节码的编译生成：高级语言程序由源代码到目标代码的生成过程称为编译。Java程序中源代码经编译生成的目标码为两个字节的字节码（16位）。

3. 字节码的解释：字节码不能直接运行在一般的操作系统平台上，而必须运行在“Java虚拟机”上。运行Java程序时首先应该启动这个虚拟机，然后由它来解释、执行Java的字节码文件。

1.4.1 Java Application 程序建立及运行

1. 利用某一种文本编辑器建立Java源程序文件；
2. 利用Java编译器（Javac）编译该application，生成.class字节码文件；
3. 利用解释器(Java)解释字节码文件，完成该程序的运行过程。

1. Java源程序命名注意事项

(1)如果源文件中有多个类，那么只能有一个类是public类

(2)如果有一个类是public类，那么源文件的名字必须与这个公共类的名字完全相同，扩展名为.java

(3)如果源文件没有public类，那么源文件的名字只要与其中某个类的名字相同，并且扩展名是.java就可以了。

- 【程序示例c1_1.java】

```
public class c1_1
{
    public static void main(String args[])
    {
        System.out.println("My first Java Application!");
    }
}
```

输入编辑完成后，保存文件。

程序分析

(1) public class c1_1

声明此程序要建立一个新类，类名为c1_1。public指出这个类是公共类，而这个类定义的内容就在后面紧跟的花括号内。

任何Java程序都必须以类的形式出现，一个程序中可以定义若干个类，但只能定义一个public类。

定义类必须用关键字class作为标志。如果在一个程序中只定义了一个public类，那么类名一定是文件名，否则编译会出错。

(2) public static void main(String args[ ])
用main定义了一个主方法，当程序执行时，解释器会找主方法，它是程序的入口点，若无此方法，解释器会显示错误信息。

```
System.out.println("My first Java Application!");
```

此语句的功能是输出字符串：

“My first Java Application!”。

(3) 在System.out.println( )中：

System是Java类库中的一个类，利用此类可以获得Java运行环境的有关信息和输入输出信息等；

out是System类中的一个对象；

println( )是out对象的一个方法，此方法的作用是向标准输出设备(这里是显示器)输出参数指定的字符串，输出完成后光标定位在下一行。

1.4.2 Java Applet 程序建立及运行

- Java Applet在WWW中引入动态交互的内容，使网络更广泛地深入社会生活的方方面面。

- Java Applet的源代码编辑、字节码的编译生成过程与Java Application相同，但它不是可以独立运行的程序，它的字节码文件必须嵌入到HTML程序的文件中，并由WWW浏览器来解释执行Java Applet的字节码程序。

1. Java Applet程序建立步骤

1) 利用文本编辑器建立Java源程序文件

2) 利用Java编译器编译该Java Applet，产生.class字节码文件

3) 建立一个HTML文件，在其中嵌入Java字节码文件

4)用WWW浏览器或AppletViewer装入该HTML文件，使Applet运行。

- 【程序示例c1_2.java】

```
import java.awt.*;
import java.applet.*;
public class c1_2 extends Applet
{
   public void paint(Graphics g)
   {
      g.drawString("Java Now!", 25, 25);
   }
}
```

程序分析

(1) import语句

Java的类库中存储了许多已编写好的类，将这些类按照功能分为许多包，提供给编程序人员使用。编写Java程序时，若要使用Java类库中的类，则必须先用import语句将其引用。

```
import java.awt.*;
import java.applet.*;
```

(2) public class c1_2 extends Applet

声明此程序要建立一个公共类，类名为c1_2。

extends Applet说明该类是Applet的子类，因为Applet类具有处理图形用户接口(GUI)的功能，其中定义的数据成员和方法规定了WWW浏览器如何解释Java Applet程序。根据子类继承父类的原则，可使WWW浏览器顺利地实现用户程序的功能。

(3) public void paint(Graphics g)

定义了一个名为paint的方法，该方法用于重画Java Applet对象的内容。paint( )方法的参数g是属于Graphics类的对象，当程序执行到这里时，g作为参数被传递至paint( )中。

(4) g.drawString("Java Now!", 25, 25);

语句中的g.drawString( )方法用于将字符串"Java Now!"显示在applet窗口。

g.drawString( )方法有三个参数：

参数1是以双引号引起的字符串，它便是窗口中显示的内容；

参数2，参数3分别是applet窗口的x和y坐标(以像素点为单位)。对于Java Applet的更多知识，将在后面章节陆续介绍。