

建议

- 记笔记，加注释，写总结
- 不要完全依赖于书和视频
- 建立有效的学习方法
- 学习软件编程的捷径--敲，狂敲

第一章 Java 概述

毕向东

基础常识

- 软件开发
- 人机交互方式
- 常用的DOS命令

软件开发

- 什么是软件？
 - 软件：一系列按照特定顺序组织的计算机数据和指令的集合。
 - 常见的软件：
 - 系统软件：
 - 如：DOS，windows，Linux等。
 - 应用软件：
 - 如：扫雷，迅雷，QQ等。
- 什么是开发？
 - 制作软件

人机交互

- 软件的出现实现了人与计算机之间的更好的交互。
- 交互方式：
 - 图形化界面：这种方式简单直观，使用者易于接受，容易上手操作。
 - 命令行方式：需要有一个控制台，输入特定的指令，让计算机完成一些操作。较为麻烦，需要记录住一些命令。

命令行方式

dos命令行，课程中常见的命令。

- dir：列出当前目录下的文件以及文件夹
- md：创建目录
- rd：删除目录
- cd：进入指定目录
- cd..：退回到上一级目录
- cd/：退回到根目录
- del：删除文件
- exit：推出dos命令行

计算机语言

- 通过**DOS**命令行的演示，发现原来操作计算机就如同和计算机说话一样。
- 我们告诉它做什么，它就可以做什么。前提是，我们和它说的内容它必须识别才可以。这就是计算机语言。

1.1 什么是计算机语言？

- **语言：**是人与人之间用于沟通的一种方式。
例如：中国人与中国人用中文沟通。
而中国人要和韩国人交流，就要学习韩语。
- **计算机语言：**人与计算机交流的方式。
如果人要与计算机交流，那么就要学习计算机语。
计算机语言有很多种，如：**C**，**C++**，**Java**等。
这里，我们选择其中的一种：**Java**语言。

1.2 Java语言概述

- 是SUN（Stanford University Network，斯坦福大学网络公司）1995年推出的一门高级编程语言。
- 是一门面向Internet的编程语言
- 随着Java技术在Web方面的不断成熟，已经成为Web应用程序的首选开发语言。
- 是简单易学，完全面向对象，安全可靠，与平台无关的编程语言。

1.2.1 Java语言的三种技术架构

J2EE(Java 2 Platform Enterprise Edition)企业版

是为开发企业环境下的应用程序提供的一套解决方案。

该技术体系中包含的技术如 Servlet Jsp等，主要针对于Web应用程序开发。

J2SE (Java 2 Platform Standard Edition) 标准版

是为开发普通桌面和商务应用程序提供的解决方案。

该技术体系是其他两者的基础，可以完成一些桌面应用程序的开发。

比如Java版的扫雷。

J2ME(Java 2 Platform Micro Edition)小型版

是为开发电子消费产品和嵌入式设备提供的解决方案。

该技术体系主要应用于小型电子消费类产品，如手机中的应用程序等。

Java5.0版本后，更名为 **JAVAEE** **JAVASE** **JAVAME**

1.2.2 Java语言的特点：跨平台性

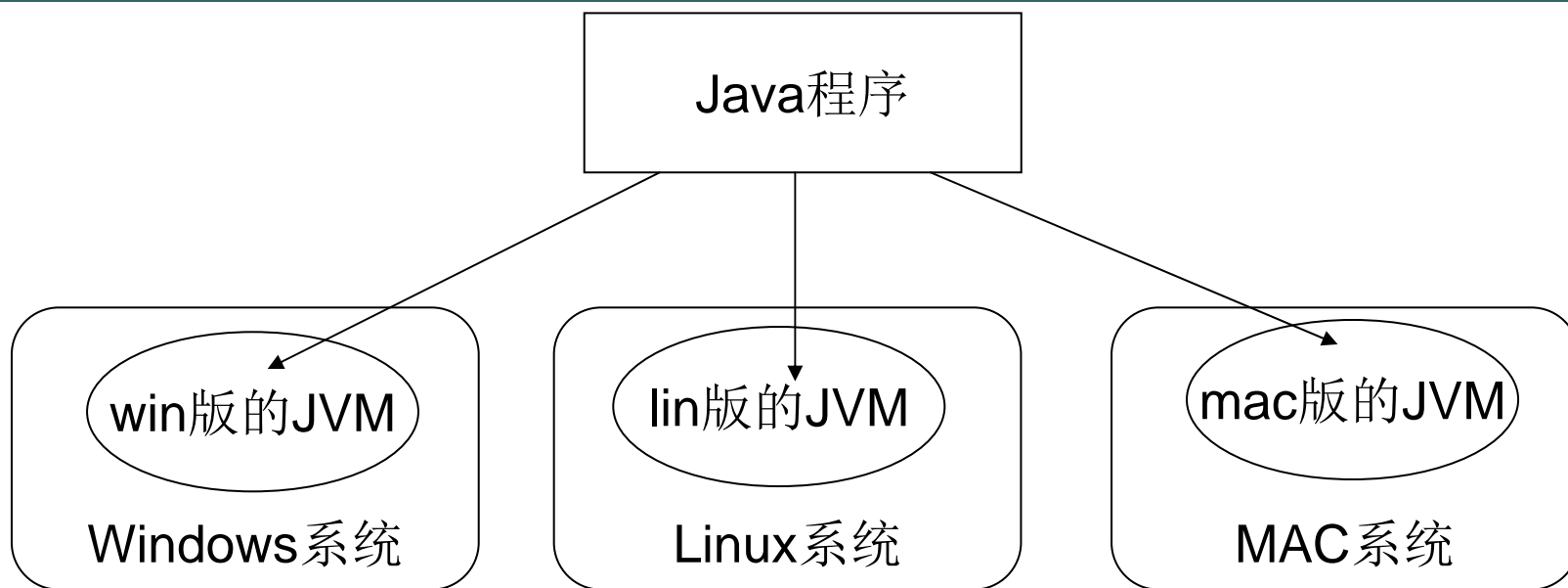
- 什么是跨平台性？

通过Java语言编写的应用程序在不同的系统平台上都可以运行。

- 原理是什么？

只需要在运行Java应用程序的操作系统上，先安装一个Java虚拟机(JVM **J**ava **V**irtual **M**achine)即可。由JVM来负责Java程序在该系统中的运行。

1.2.3 Java语言的特点：跨平台性(图解)



因为有了JVM，所以同一个Java程序在三个不同的操作系统中都可以执行。这样就实现了Java程序的跨平台性。也称为Java具有良好的可移植性。

1.3 Java语言的环境搭建

- 明确什么是JRE,JDK
- 下载JDK
- 安装JDK
- 配置环境变量
- 验证是否成功

1.3.1 什么是JRE,JDK?

JRE(Java Runtime Environment Java运行环境)

包括Java虚拟机(JVM Java Virtual Machine)和Java程序所需的核心类库等, 如果想要运行一个开发好的Java程序, 计算机中只需要安装JRE即可。

JDK(Java Development Kit Java开发工具包)

JDK是提供给Java开发人员使用的, 其中包含了java的开发工具, 也包括了JRE。所以安装了JDK, 就不用在单独安装JRE了。

其中的开发工具: 编译工具(javac.exe) 打包工具(jar.exe)等

简单而言: 使用JDK开发完成的java程序, 交给JRE去运行。

Java6.0 Platform

Java Language	Java Language									
Tools & Tool APIs	java	javac	javadoc	apt	jar	javap	JPDA	JConsole	Java VisualVM	
	Security	Int'l	RMI	IDL	Deploy	Monitoring	Troubleshoot	Scripting	JVM TI	
RIAs	Java Web Start					Applet / Java Plug-in				
User Interface Toolkits	AWT				Swing			Java 2D		
	Accessibility		Drag n Drop		Input Methods		Image I/O	Print Service		Sound
Integration Libraries	IDL	JDBC		JNDI		RMI	RMI-IIOP		Scripting	
Other Base Libraries	Beans		Intl Support		Input/Output		JMX	JNI		Math
	Networking		Override Mechanism		Security		Serialization	Extension Mechanism		XML JAXP
lang and util Base Libraries	lang and util		Collections		Concurrency Utilities		JAR		Logging	Management
	Preferences API		Ref Objects		Reflection		Regular Expressions		Versioning	Zip Instrumentation
Java Virtual Machine	Java Hotspot Client and Server VM									

1.3.2 下载JDK

- JDK(Java Development Kit Java开发工具包)
- 官方网址:
 - www.oracle.com 参阅index.html
 - java.sun.com 参阅index1.html

1.3.3 安装JDK

- 傻瓜式安装，下一步即可。
- 建议：安装路径不要有中文或者特殊符号如空格等。
- 当提示安装JRE时，可以选择不安装。
- 具体流程参阅 **JDK6安装配置简程.doc**

1.3.4 配置环境变量path(1)

- 在dos命令行中敲入javac，出现错误提示。

```
C:\Documents and Settings\Administrator>javac  
'javac' 不是内部或外部命令，也不是可运行的程序  
或批处理文件。
```

- 错误原理：当前执行的程序在当前目录下如果不存在，windows系统会在系统中已有的一个名为path的环境变量指定的目录中查找。如果还没有找到，就出现以上的错误提示。
- 所以进入到 jdk安装路径\bin目录下，执行javac。看到javac参数提示信息。

```
D:\jdk1.6.0_22\bin>javac  
用法: javac <选项> <源文件>  
其中, 可能的选项包括:  
-g                生成所有调试信息  
-g:none           不生成任何调试信息  
-g:<lines,vars,source> 只生成某些调试信息
```

1.3.4 配置环境变量path(2)

- 每次执行java的工具都要进入到bin目录下，是非常麻烦的。
- 可不可以任何目录下都可以执行java的工具呢？
- 根据windows系统在查找可执行程序的原理，可以将java工具所在路径定义到path环境变量中，让系统帮我们去找运行执行的程序。

1.3.4 配置环境变量path(3)

- 我的电脑--属性--高级--环境变量
- 编辑path环境变量，在变量值开始处加上java工具所在目录，后面用“;”和其他值分隔开即可。
- 打开DOS命令行，任意目录下敲入javac。
- 如果出现javac 的参数信息，配置成功。
- 这种配置方式，一劳永逸。
- 具体操作流程，可以参阅 JDK6安装配置简程.doc

1.3.4 配置环境变量path(4)

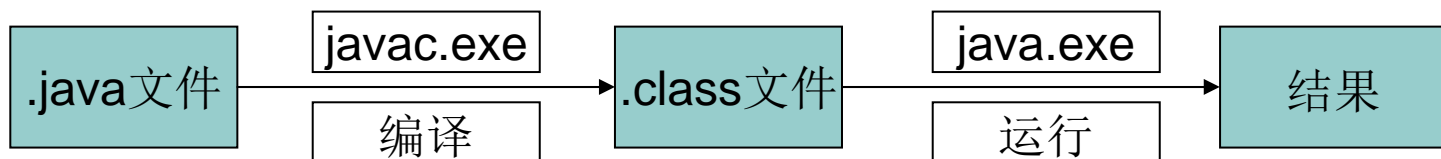
- 临时配置方式：通过dos命令中set命令完成
 - set：用于查看本机的所有环境变量的信息。
 - set 变量名：查看具体一个环境变量的值。
 - set 变量名=：清空一个环境变量的值。
 - set 变量名=具体值：给指定变量定义具体值。
- 想要在原有环境变量值基础上添加新值呢？
 - 首先，通过%变量名%操作符获取到原有环境变量的值。
 - 然后加上新值后在定义给该变量名即可
 - 举例：给path环境变量加入新值
 - set path=新值;%path%
- 注意：这种配置方式只在当前dos窗口有效。窗口关闭，配置消失。

1.3.5 配置技巧

- 为了不因为jdk的目录或者目录名称的改变，而不断更改path的值，而导致对path变量值的误操作，可以通过以下技巧完成。
 - 新建一个环境变量 JAVA_HOME 记录住jdk的目录。
 - 在path中通过%%动态的获取JAVA_HOME的值即可。
 - JAVA_HOME=F:\jdk1.6.0_01
 - path=%JAVA_HOME%\bin;%path%
 - %path%: 动态获取path环境变量的值。
 - %JAVA_HOME%: 动态获取名称为JAVA_HOME环境变量的值。

1.4 Java程序开发体验--Hello World

1. 将Java代码编写到扩展名为.java的文件中。
 2. 通过javac命令对该java文件进行编译。
 3. 通过java命令对生成的class文件进行运行。
- 具体操作流程可参阅：Hello World流程图.doc



1.4 Java程序开发体验--编写

- 步骤一：编写

- 选择最简单的编辑器记事本。
- 敲入代码

```
class Demo  
{  
}
```

- 将文件保存成**Demo.java**，这个文件是存放java代码的文件，称为源文件。

1.4 Java程序开发体验--编译

- 步骤二：编译
 - 有了java源文件，将其编译成JVM可以识别的文件。
 - 在该源文件目录下，通过javac编译工具对Demo.java文件进行编译。
 - 如果程序没有错误，没有任何提示，但在当前目录下会出现一个Demo.class文件，该文件称为字节码文件，也是可以执行的java的程序。

1.4 Java程序开发体验--运行

- 步骤三：运行

- 有了可执行的java程序(Demo.class字节码文件)
- 通过运行工具java.exe对字节码文件进行执行。
- 出现提示：缺少一个名称为main的方法。

```
C:\>java Demo  
Exception in thread "main" java.lang.NoSuchMethodError: main
```

- 因为一个程序的执行需要一个起始点或者入口，所以在Demo类中的加入public static void main(String[] args){}
- 对修改后的Demo.java源文件需要重新编译，生成新的class文件后，在进行执行。
- 发现没有编译失败，但也没有任何效果，因为并没有告诉JVM要帮我们做什么事情，也就是没有可以具体执行的语句。
- 想要和JVM来个互动，只要在main方法中加入一句
- System.out.println("hello java");在重新编译，运行即可。

1.5 配置classpath(1)

- 既然class文件(字节码文件)是java的运行文件，可不可以实现，在任意目录下都可以执行某一目录下的class文件呢？
- 根据path环境变量的原理，可以定义一个名为classpath环境变量，将要运行的class文件所在目录定义在该变量中。
- 例：set classpath=c:\
- classpath变量值是java类文件的目录
- path变量是windows程序文件的目录

1.5 配置classpath(2)

- JVM查找类文件的顺序:

- 如果没有配置classpath环境变量, JVM只在当前目录下查找要运行的类文件。
- 如果配置了classpath环境, JVM会先在classpath环境变量值的目录中查找要运行的类文件。
 - 值的结尾处如果加上分号, 那么JVM在classpath目录下没有找到要指定的类文件, 会在当前目录下在查找一次。
 - 值的结尾出如果没有分号, 那么JVM在classpath目录下没有找到要指定的类文件, 不会在当前目录下查找, 即使当前目录下有, 也不会运行。
- 建议: 配置classpath环境变量时, 值的结尾处不要加分号, 如果需要访问当前目录可以用“.”表示。

1.6 创建问题及解决方法

```
C:\>javac Demo.java
javac: 找不到文件: Demo.java
用法: javac <options> <source files>
-help 用于列出可能的选项
```

源文件名不存在或者写错，或者当前路径错误。

```
C:\>java Demo
Exception in thread "main" java.lang.NoClassDefFoundError: Demo
Caused by: java.lang.ClassNotFoundException: Demo
    at java.net.URLClassLoader$1.run(URLClassLoader.java:202)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:190)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:307)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:301)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:248)
Could not find the main class: Demo. Program will exit.
```

类文件名写错，或者类文件不在当前路径下，或者不再classpath指定路径下。

1.6 创建问题及解决方法

```
C:\>javac Demo.java
Demo.java:5: 需要 ';'
        System.out.println("Hello World!")
                        ^
1 错误
```

编译失败，注意错误出现的行数，在到源代码中指定位置排错。

1.7 复习建议

- JDK,JRE,JVM的特点。
- 环境变量的配置path和classpath以及作用。
- Java程序的编写，编译，运行步骤。
- 在配置，编写，编译，运行各个步骤中常见的错误以及解决方法。

作业

1. 简述JDK,JRE的区别。
2. 简述path和classpath环境变量的作用。
3. 独立编写Hello World程序。
 - 1.class的作用
 - 2.main函数的作用
 - 3.输出语句的作用