

2.java 的历史、java 跨平台

Java 简介

语言起源

Java 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。
Java 名称的来源 Java 最初被命名为 Oak，目标在于家用电器等小型系统的编程语言，随着市场的变化，SUN 着眼于网络应用的开发。Java 的名字的来源：Java 是印度尼西亚爪哇岛的英文名称，因盛产咖啡而闻名。 机顶盒 高斯林(高司令) 用户想看什么片就看什么...优酷

- 1995 年 5 月 23 日，JAVA 语言诞生 雏形
- 1996 年，1997 分别推出 JDK1.0，JDK1.1 版
- 1999 年，JAVA 被分成 J2SE,J2EE,J2ME
- 2000 年，JDK1.4 发布
- 2004 年 9 月 30 日 18:00PM，J2SE1.5 发布，成为 Java 语言发展史上的又一里程碑。为了表示该版本的重要性，J2SE1.5 更名为 Java SE 5.0
- 2005 年，JAVA 版本正式更名为 JAVAEE,JAVASE， JAVAME
- 2006 年 12 月，SUN 公司发布 JRE6.0
- 2009 年 04 月 20 日，Oracle 以 74 亿美元收购 Sun。取得 java 的版权。
- 2010 年 9 月，JDK7.0 已经发布，增加了简单闭包功能。
- 2011 年 7 月，甲骨文公司发布 java7 的正式版。

2013 年 8 月计算机语言使用排行榜

Position Aug 2013	Position Aug 2012	Delta in Position	Programming Language	Ratings Aug 2013	Delta Aug 2012	Status
1	2	↑	Java	15.978%	-0.37%	A
2	1	↓	C	15.974%	-2.96%	A
3	4	↑	C++	9.371%	+0.04%	A
4	3	↓	Objective-C	8.082%	-1.46%	A
5	6	↑	PHP	6.694%	+1.17%	A
6	5	↓	C#	6.117%	-0.47%	A
7	7	=	(Visual) Basic	3.873%	-1.46%	A
8	8	=	Python	3.603%	-0.27%	A
9	11	↑↑	JavaScript	2.093%	+0.73%	A
10	10	=	Ruby	2.067%	+0.38%	A

疑问：学习 java 需要考证吗？（不需要考的，英语四级证书）

Tip1: JAVA语言概述(了解)

- SUN JAVA认证

在全球范围内，SUN公司的Java技术培训和认证不仅是Java开发技术领域最权威的认证，同时也是开发类认证中很重要的一种，SUN公司的Java技术认证包括4种：

- Sun Certified Java Programmer (SCJP) 1500
- Sun Certified Java Developer (SCJD) 2000~3500
- Sun Certified Web Component Developer for Java 2 Platform Enterprise Edition
- Sun Certified Enterprise Architect for J2EE Technology

web135.net

SUN CERTIFICATION

Denis A Erofeev

HAS FULFILLED ALL REQUIREMENTS AS A

SUN CERTIFIED PROGRAMMER

FOR THE JAVA™ 2 PLATFORM

(J2SE 5.0)







不要考，了解即可。

java 重要特性之 ----- 跨平台(一次编译，到处运行).

java 语言重要特性之跨平台

2.2.1 跨平台

什么是跨平台（计算机系统），跨平台概念是软件开发中一个重要的概念，即不依赖于操作系统，也不信赖硬件环境。一个操作系统下开发的应用，放到另一个操作系统下依然可以运行。

2.2.2. Java 的跨平台

“write once run anywhere”

Java 的跨平台是相对于其他编程语言而言的，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。一般的高级语言如果要在不同的平台上运行，至少需要编译成不同的目标代码。在 windows 编译的 c 语言程序，在 linux 下是不能运行的，想要在 linux 下运行，那么就需要将 c 语言程序源码在 linux 平台上重新编译成 linux 平台的目标代码才行。当然在 linux 下编译的 c 语言程序在 windows 下也是不能运行的。而 **Java 语言在不同平台上运行时不需要重新编译。**

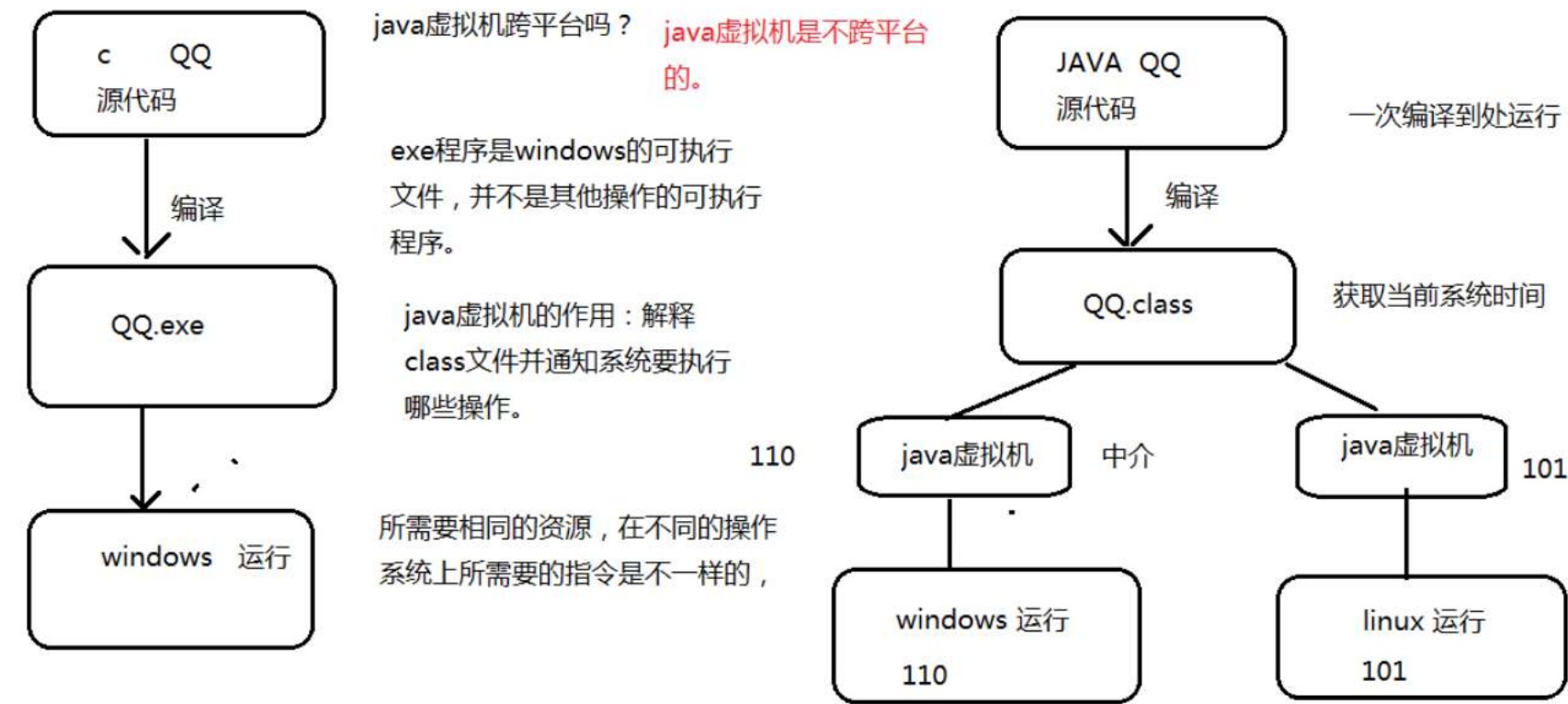
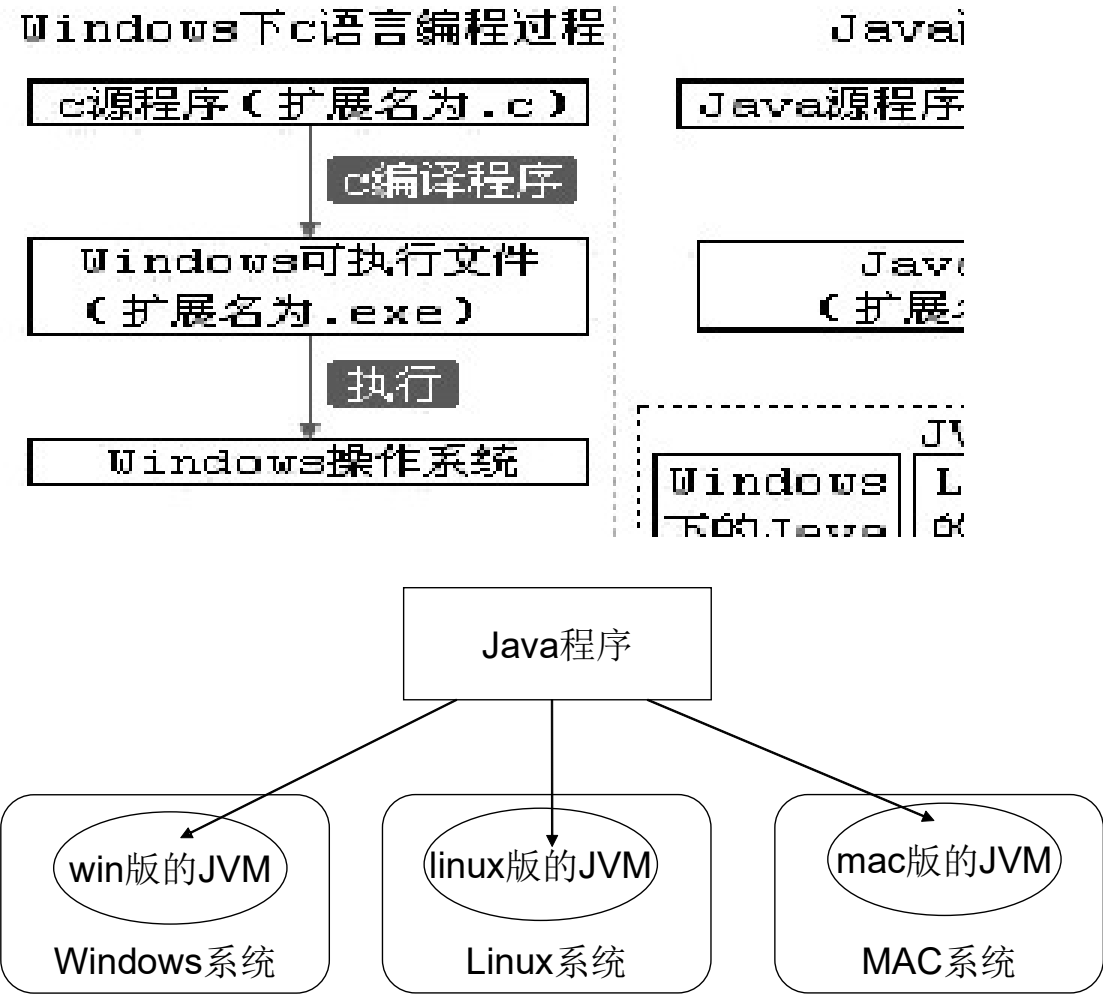
Java 跨平台的原理：Java 的跨平台是通过 Java 虚拟机（JVM）来实现的。

2.2.3. Java 虚拟机（JVM）

Java Virtual Machine 一个软件模拟的计算机。Java 源程序想象成我们的 C 源程序，Java 源程序编译后生成的字节码（.class）就相当于 C 源程序编译后的二进制可执行文件。JVM 虚拟机相当于计算机系统（操作系统+硬件环境），Java 解释器相当于 CPU。

.exe 文件直接在 windows 操作系统下运行，在 CPU 上运行的是机器码（.exe 文件）。
.class 字节码文件是运行在 JVM 虚拟机下的，在 Java 解释器上运行的是 Java 字节码。

Java 解释器相当于运行 Java 字节码的“CPU”，但该“CPU”不是通过硬件实现的，而是用软件实现的。Java 解释器实际上就是特定的平台下的一个应用程序。只要实现了特定平台下的解释器程序，Java 字节码就能通过解释器程序在该平台下运行，这是 Java 跨平台的根本。当前，并不是在所有的平台下都有相应 Java 解释器程序，这也是 Java 并不能在所有的平台下都能运行的原因，它只能在已实现了 Java 解释器程序的平台下运行。



110 是 Windows 上的指令。
101 是 linux 上的指令。

3.配置 path 环境变量、第一个 helloworld 例子

Java 的三大平台

JavaSE

Java SE（Java Platform，Standard Edition）。Java SE 以前称为 J2SE。它允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。Java SE 包含了支持 Java Web 服务开发的类，并为 Java Platform，Enterprise Edition（Java EE）提供基础。
例子：类似 QQ、扫雷、飞秋、凌波这样子的桌面应用。

Java EE

Java EE（Java Platform，Enterprise Edition）。这个版本以前称为 J2EE。企业版本帮助开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。Java EE 是在 Java SE 的基础上构建的，它提供 Web 服务、组件模型、管理和通信 API，可以用来实现企业级的面向服务体系结构（service-oriented architecture，SOA）和 Web 2.0 应用程序。

Java ME

Java ME（Java Platform，Micro Edition）。这个版本以前称为 J2ME，也叫 K-JAVA。Java ME 为在移动设备和嵌入式设备（比如手机、PDA、电视机顶盒和打印机）上运行的应用程序提供一个健壮且灵活的环境。Java ME 包括灵活的用户界面、健壮的安全模型、许多内置的网络协议以及对可以动态下载的连接和离线应用程序的丰富支持。基于 Java ME 规范的应用程序只需编写一次，就可以用于许多设备，而且可以利用每个设备的本机功能。

java 的环境的搭建：

jre： java 运行环境。 **jre = java 虚拟机 + 核心类库(辅助 java 虚拟机运行的文件)**

jdk： java 开发工具集 **jdk = jre + java 开发工具。**

java 开发工具，例如，javac 编译工具，java 解释器执行工具。

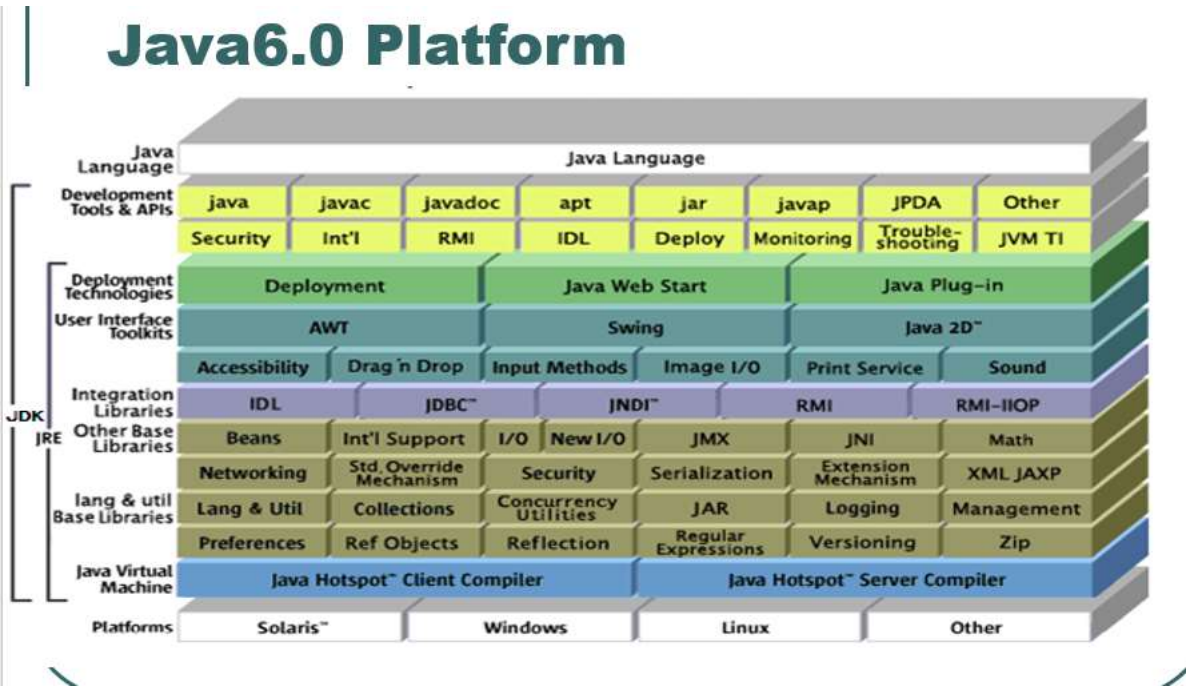
如果只是软件使用者，那么安装 jre 即可，提供运行环境。
如果是软件开发者，那么就需要安装 jdk。因为需要开发工具。

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 是整个 Java 的核心，包括了 Java 运行环境、Java 工具和 Java 基础类库。

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。



安装 jdk 的路径要注意： 不能带有中文路径或者是空格。



安装过程中，先安装 jdk，然后又提示你安装 jre，实际上这个 jre 完全可以不安装，因为前面安装的 jdk 已经包含了 jre 了。

课程使用的是 jdk7.0 版本。

jdk 目录的介绍：

- bin: 存放的是 java 的开发工具。
- db : JDK7 附带的一个轻量级的数据库，名字叫做 Derby。
- include : 存放的调用系统资源的接口文件。java 程序中向 os 调用的时候，需要的本地 os 接口文件。
- jre : java 运行环境
- lib : 核心类库。
- src.zip : java 源代码

java.exe 启动 java 虚拟机解释并执行指定的 class 文件。

java 工具的使用格式：

java 指定的 class 文件

javac.exe 启动 java 编译器对指定的 java 源文件进行编译。

编写第一个 java 的程序：

第一步： 新建一个 txt 文件。把后缀名改成 java。

第二步： **java 代码全部都是写到类上面的。也就是在 java 中，类是程序组织的最小单位。**

如何编写一个类：

使用一个关键字 class 即可定义个类。

格式：

```
class 类名{  
  
    } 备注： 大括号表示是一个类的范围，在类的范围之内都可以写代码，
```

第三步： 编写一个主方法。 主方法的写法是固定的，是一个程序的入口，代码从主方法开始执行，

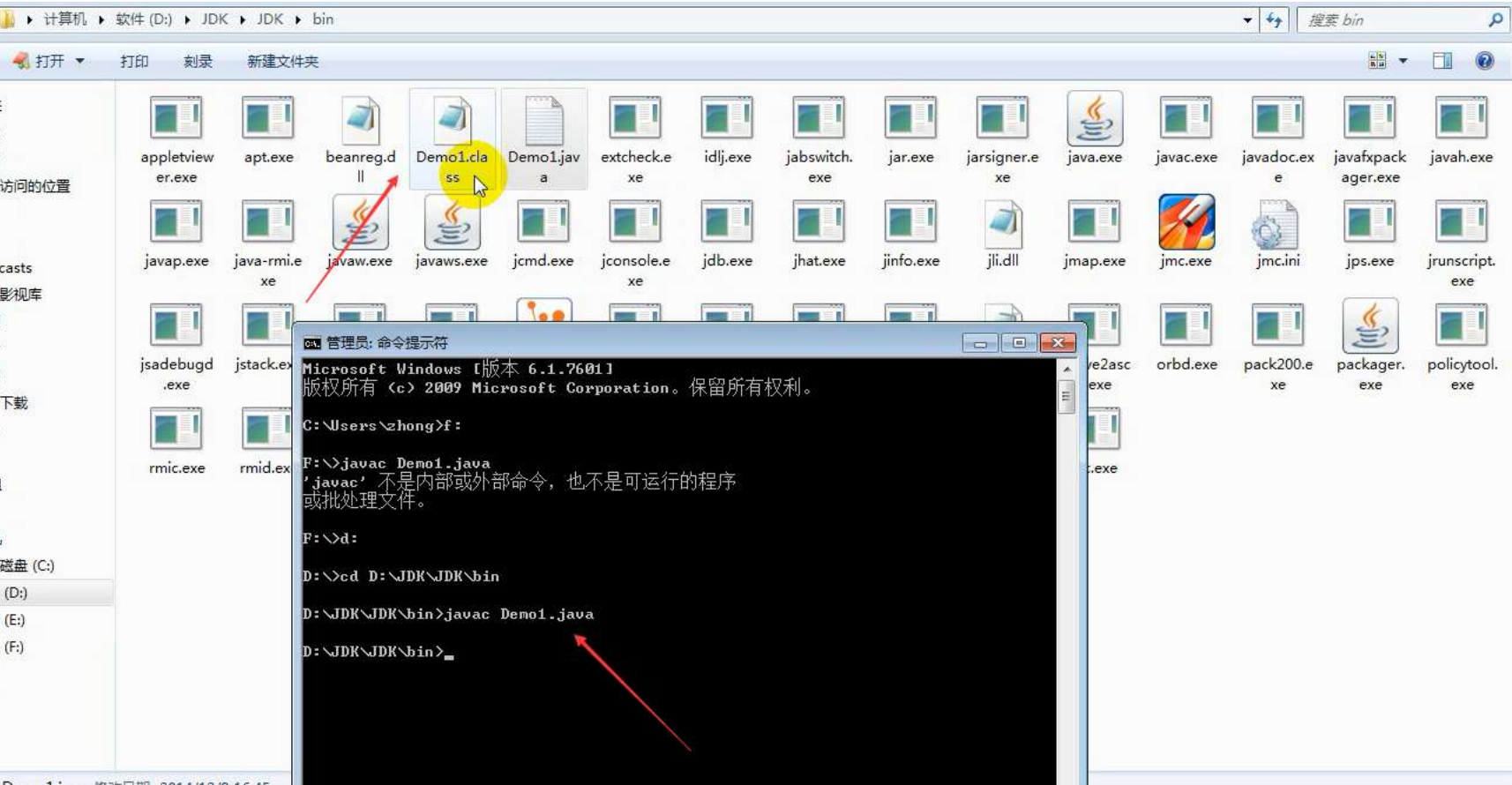
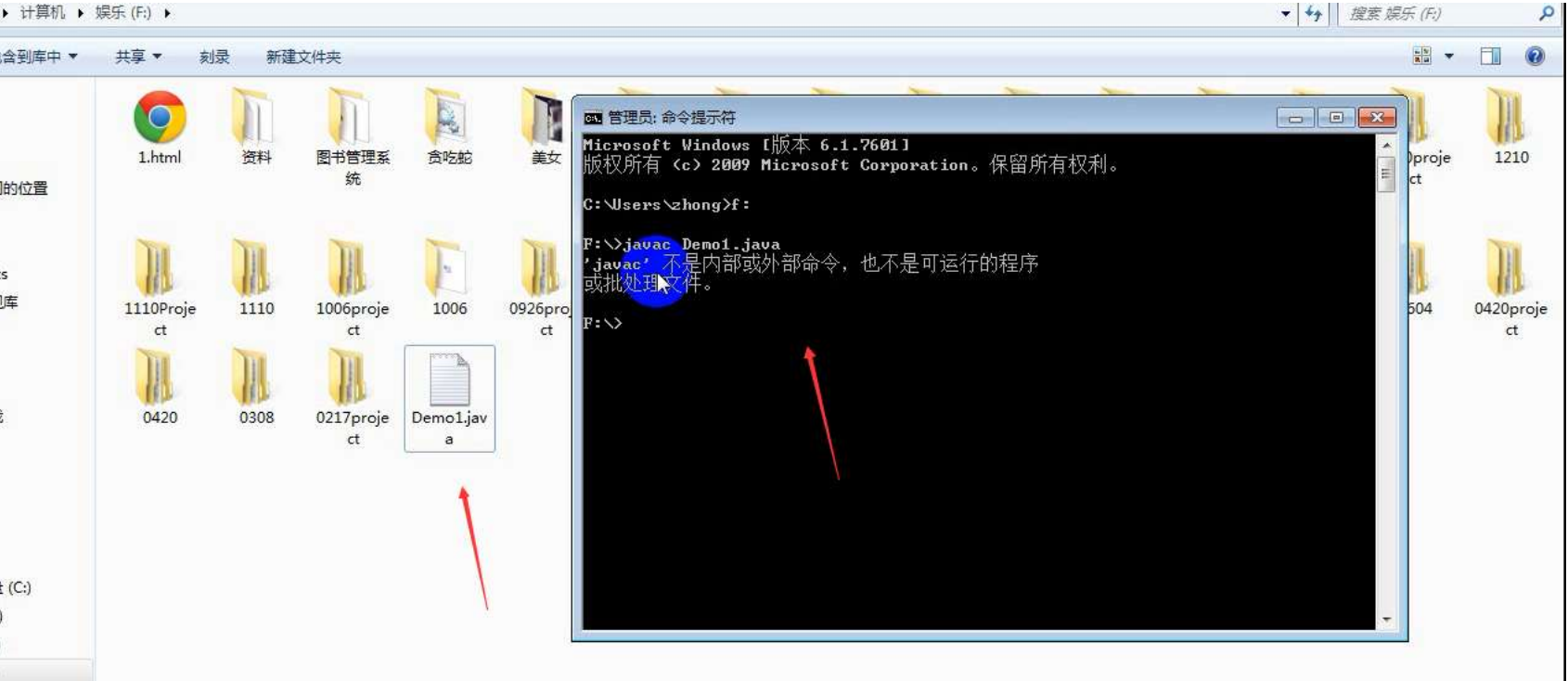
当直接双击一个可执行文件一闪而过就关闭了，那么一般说明，这个可执行文件需要一个控制台中来执行它。

如何启动 java 编译对 java 的源文件进行编译？

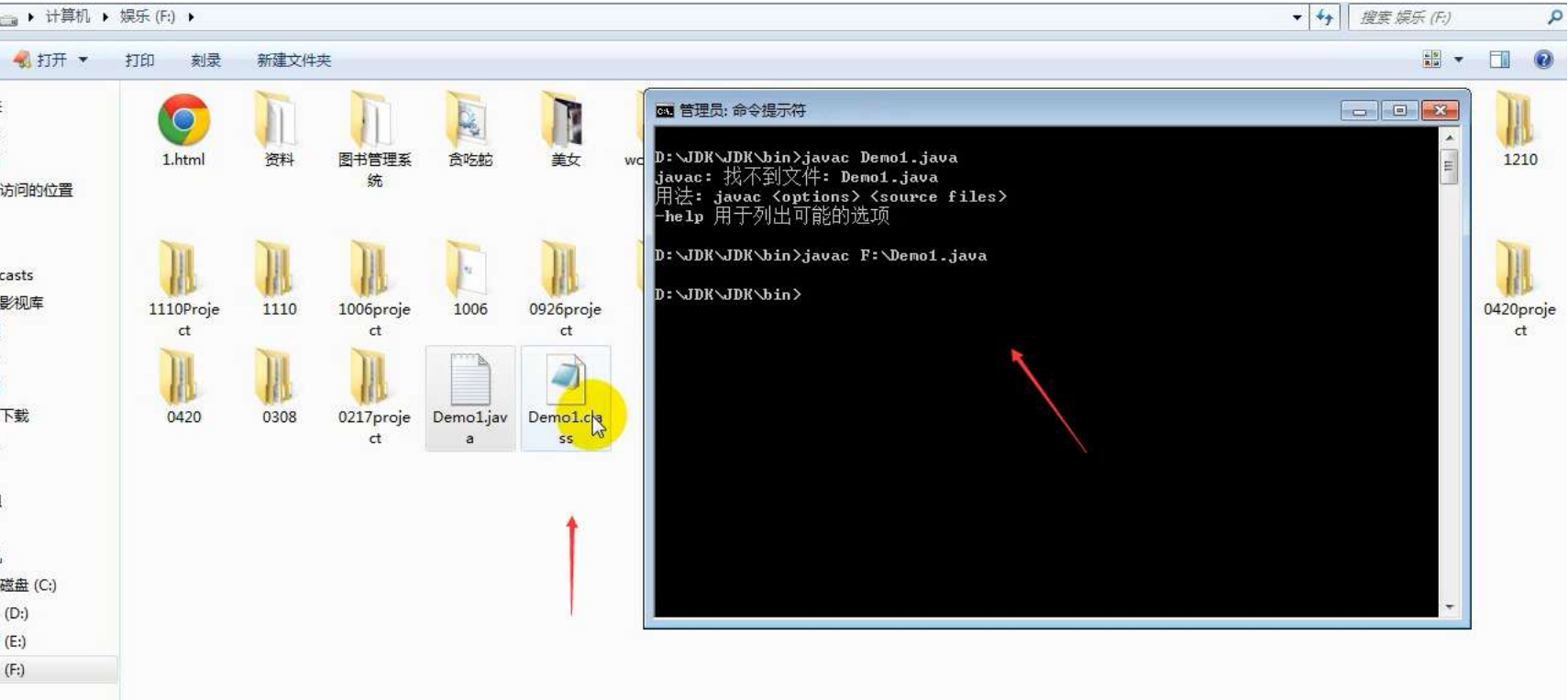
javac 的使用格式：

javac java 源文件。

编译的过程中我们就面临 的问题：每次编译的时候都需要写上 java 源文件的完整路径。 烦!!!
即需要在 javac 可执行文件的目录下，执行"javac java 源文件完整路径"



这样是不好的，将源文件放到了 javac 的安装目录了，不好。乱。



每次都需要写全路径，麻烦。

解决方案： 如果可以在任意的路径下都可以找到 java 的开发工具，那么该问题就解决了。

在控制台写一个可执行文件的文件名，然后能够执行的原理：

在控制台写一个可执行文件的文件名,那么系统首先会在控制台当前路径下去搜索是否存在该文件，如果可以找到指定的文件，那么就执行该文件，如果找不到该文件，那么系统还会根据 **path** 的环境变量所保存的路径信息下去搜索是否有指定的文件，如果能找到，那么就执行该文件。

```
D:\eclipse4.3\adt-bundle-windows-x86-20131030\adt-bundle-windows-x86-20131030\sdk\platform-tools;
C:\Program Files\Intel\iCLS Client\;
%SystemRoot%\system32;
%SystemRoot%
%SystemRoot%\System32\Wbem;%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\;C:\Program Files\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files\Intel\Intel
(R) Management Engine Components\IPT;C:\Program Files\Intel\OpenCL SDK\2.0\bin\x86;D:\mysql\bin;C:\Program Files\TortoiseSVN\bin;D:\firefox\quicktime\QTSystem\
```

老师的 PATH 环境变量中保存的内容。
SystemRoot 是 Windows 系统的根路径。



即这个目录。

