黑马程序员
itheima.com

非贷款，0元入学，不1万就业不给1分钱学费，我们已干四年了！

笔记总链接：http://bbs.itheima.com/thread-200600-1-1.html

Chapter 1 Java概述

一、基本常识

1.1 软件开发

1.1.1 什么是软件？

软件是一系列按照特定顺序组织的计算机数据和指令的集合。

P.S.

1、数据就是指现实生活中的年龄、姓名等信息。

2、指令就是告诉计算机如何对数据进行处理。

1.1.2 常见的软件

1. 系统软件（操作系统）

如：DOS（Disk Operating System）、Windows、Linux、Android、iOS、MAC等。

P.S.

DOS系统是命令行方式操作的操作系统，为了便于用户操作，后来产生了图形化界面操作的操作系统，也就是Windows系统。

2. 应用软件

如：扫雷，迅雷，QQ等。

P.S.

系统软件和应用软件都是用计算机语言编写出来的，计算机语言调用底层指令处理数据。

1.1.3 什么是开发？

制作软件。

1.2 人机交互

软件的出现实现了人与计算机之间更好的交互。

1.2.1 交互方式

两种方式：

1. 图形化界面(Graphical User Interface GUI)：这种方式简单直观，使用者易于接受，容易上手操作。

2. 命令行方式(Command Line Interface CLI)：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，需要记住一些命令，如早期的DOS系统。

1.3 计算机语言

1.3.1 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。

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能够识别才可以，这就是计算机语言。

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言。

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。

二、Java语言介绍

2.1 Java语言概述

1. Java语言是SUN公司(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。

2. Java语言是一门面向Internet的编程语言。

3. 随着Java技术在web方面的不断成熟，Java语言已经成为Web应用程序的首选开发语言。

4. Java语言是简单易学，完全面向对象，安全可靠，与平台（操作系统）无关的编程语言。

P.S.

Java是允许使用者将应用程序通过Internet从远端服务器传输到本地机上并执行的一种语言。

2.2 Java语言的三种技术架构

2.2.1 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。

2.2.2 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。

2.2.3 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。

P.S.

1、Java5.0版本后，三种技术架构分别更名为JAVAEE、JAVASE、JAVAME。

2、由于现在已经出现了Android、iOS、Windows Phone等手机操作系统，所以J2ME架构基本上已经不用了。目前，流行的手机软件都是基于这些最新的手机操作系统进行开发。

3、SUN公司已被Oracle公司收购，因此，Java以后会更火。

2.3 Java语言的特点：跨平台性

2.3.1 什么是跨平台性？

通过Java语言编写的应用程序在不同的操作系统平台中都可以运行。

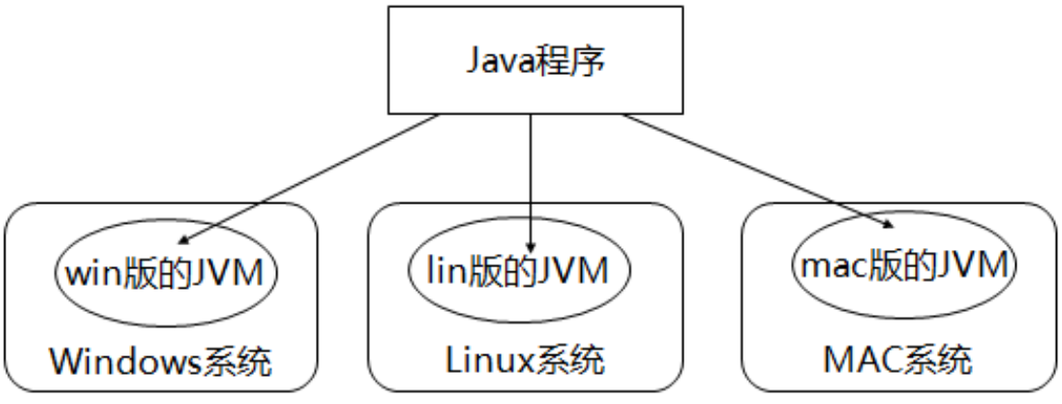
P.S.

国内操作系统市场已被Windows彻底征服，但国外用户操作系统选择项较多。因此，程序员做软件需要考虑跨平台性，而Java语言就具备跨平台性的特点。

2.3.2 原理是什么？

只要在需要运行java应用程序的操作系统中，先安装一个Java虚拟机(JVM：Java Virtual Machine)即可，由JVM来调用操作系统底层指令解析、执行Java程序，从而在该操作系统中运行。

因为有了JVM，所以同一个Java程序各种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，也就是说Java语言具有良好的可移植性。



P.S.

1、JVM是不跨平台的，不同的操作系统有不同版本的JVM。

2、用C++语言编写的程序在Windows系统中可以直接运行，在Linux系统中则不可以。这是因为Windows系统是用C和C++语言编写的，Windows系统中也内置了C和C++的解析器，所以C和C++语言编写的程序可以直接在Windows系统中运行。但Java语言编写的程序如果要在Windows系统中执行，就需要在Windows系统中安装一套能够解析、执行Java程序的软件，也就是JVM（Java虚拟机）。类似的，在Linux、MAC系统中执行Java语言编写的程序也需要安装各自系统对应的JVM。通过这种方式，就实现了Java语言“一次编译，到处运行”的跨平台性。JVM的作用就是搭建了Java语言编写的程序与操作系统之间的桥梁。

三、Java语言的环境搭建

使用任何一门语言之前必须首先搭建环境。

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，就不用再单独安装JRE了。其中的开发工具有编译工具(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。

P.S.

为什么JDK中包含一个JRE呢？

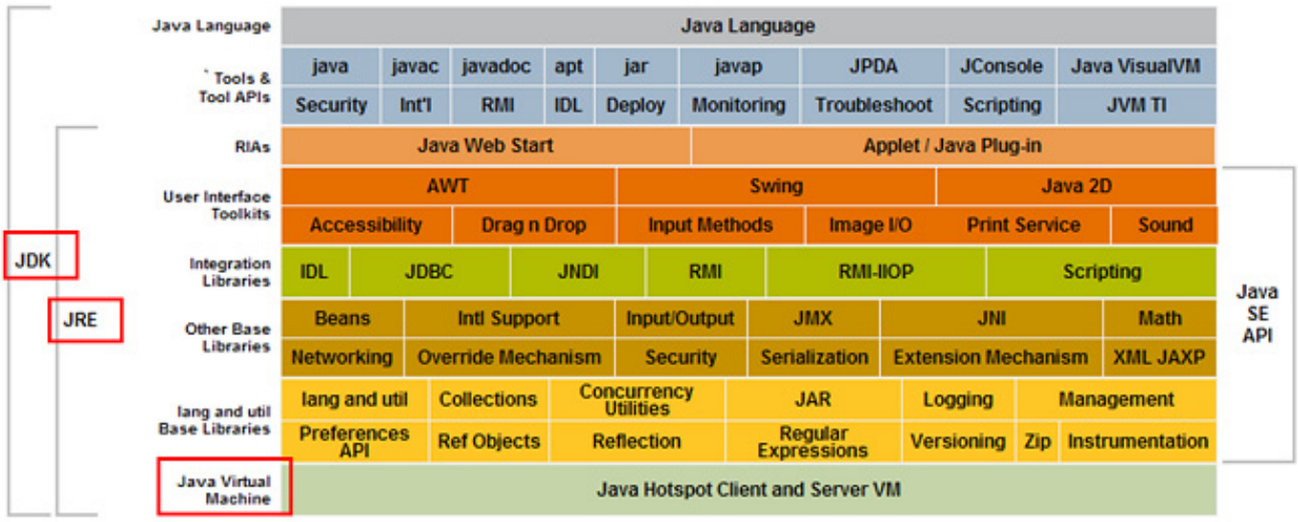
其一，开发完的程序，需要运行一下看看效果，就像exe文件需要在windows环境下运行一样。

其二，也是最重要的，JDK中的开发工具（如javac.exe、java.exe等）其实都是Java语言编写的应用程序，为了方便使用才打包成exe文件，如果没有JRE，那么这些工具是运行不了的。

总结：

JRE：JVM+类库（Java library）。

JDK：JRE+JAVA的开发工具。



3.2 JDK的下载与安装

3.2.1 下载JDK（Java Development Kit Java开发工具包）

官方网站：

www.oracle.com

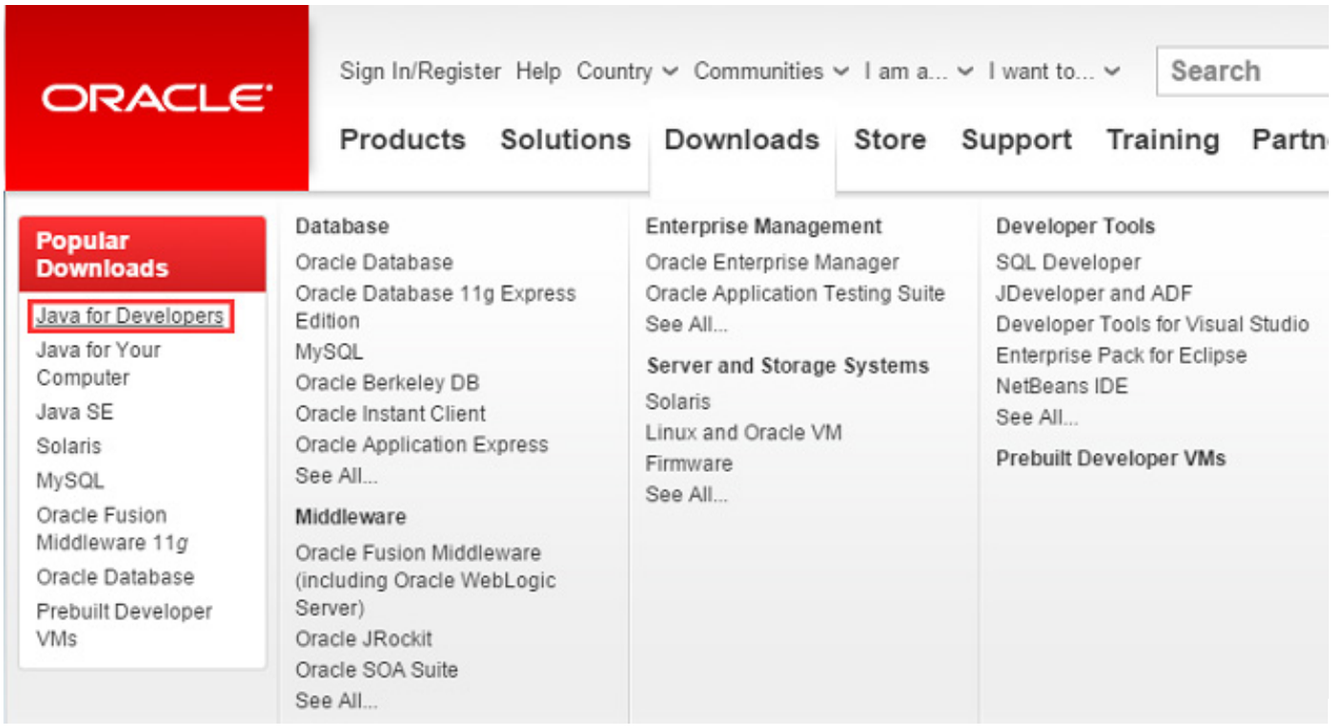
www.java.sun.com

P.S.

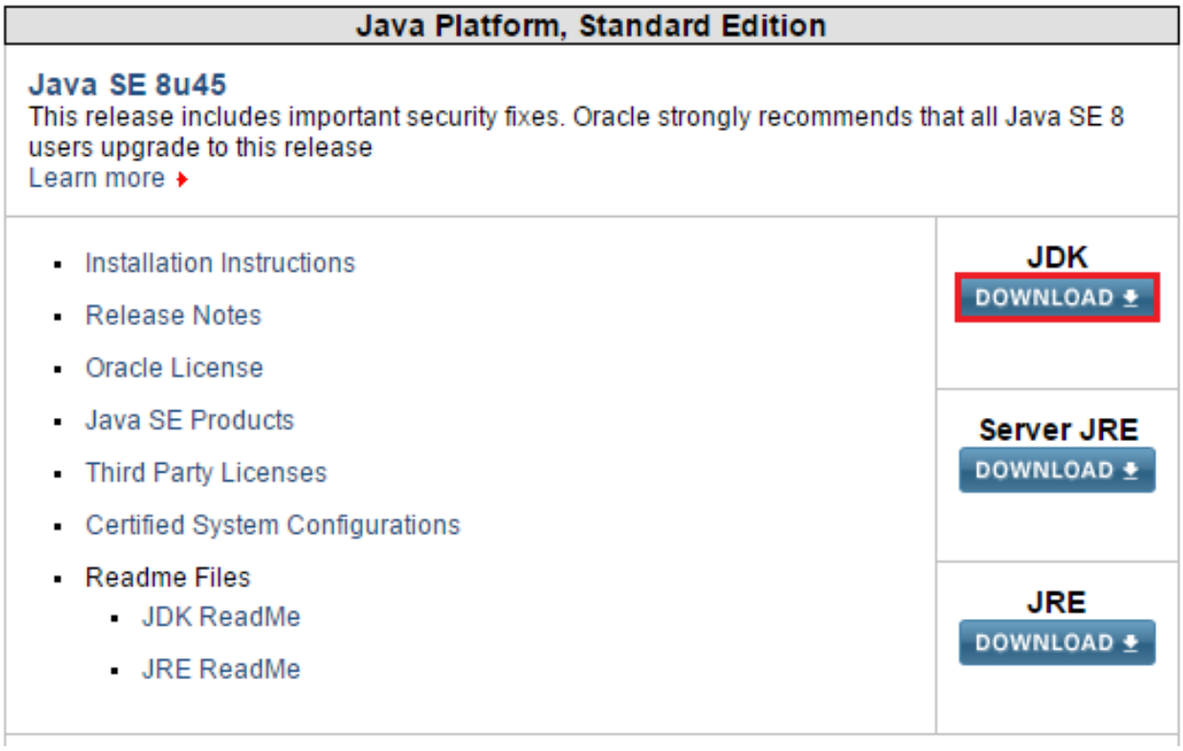
由于Sun公司已被Oracle公司收购，故访问www.java.sun.com会自动跳转到http://www.oracle.com/technetwork/java/index.html 。

JDK下载步骤示意图：

1. 访问www.oracle.com，点击Downloads下的Java for Developers。



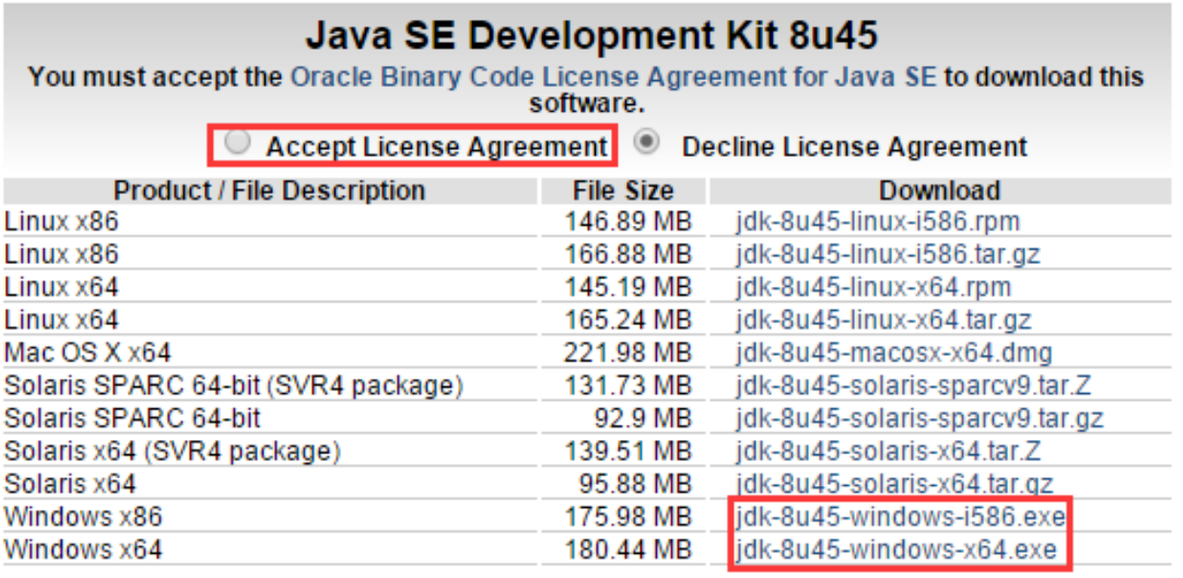
2. 点击JDK下的DOWNLOAD按钮。



P.S.

8u45代表JDK8版本，45代表子版本，u是update（更新）的缩写。

3. 在下载之前，首先需要接受JDK的许可证协议，然后再点击jdk-8u45-windows-i586.exe（32位）/jdk-8u45-windows-x64.exe（64位）进行下载。

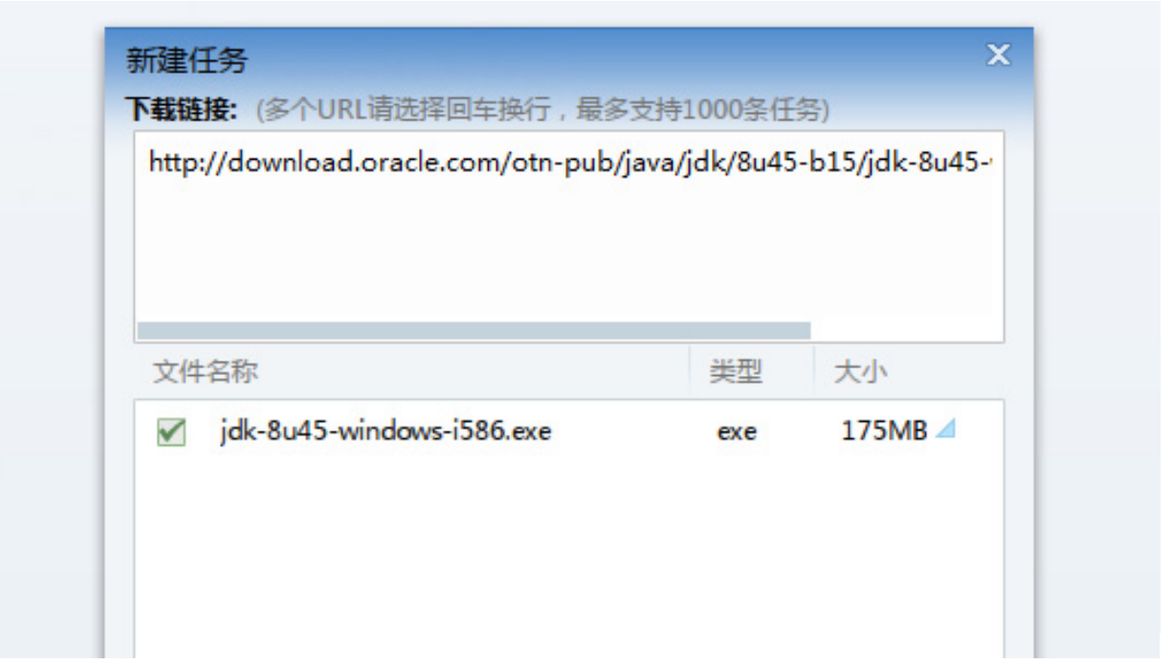


Java SE Development Kit 8u45

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

Accept License Agreement　Decline License Agreement

Product / File Description	File Size	Download
Linux x86	146.89 MB	jdk-8u45-linux-i586.rpm
Linux x86	166.88 MB	jdk-8u45-linux-i586.tar.gz
Linux x64	145.19 MB	jdk-8u45-linux-x64.rpm
Linux x64	165.24 MB	jdk-8u45-linux-x64.tar.gz
Mac OS X x64	221.98 MB	jdk-8u45-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	131.73 MB	jdk-8u45-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	92.9 MB	jdk-8u45-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	139.51 MB	jdk-8u45-solaris-x64.tar.Z
Solaris x64	95.88 MB	jdk-8u45-solaris-x64.tar.gz
Windows x86	175.98 MB	jdk-8u45-windows-i586.exe
Windows x64	180.44 MB	jdk-8u45-windows-x64.exe



P.S.

Windows x86对应的是windows 32位系统。

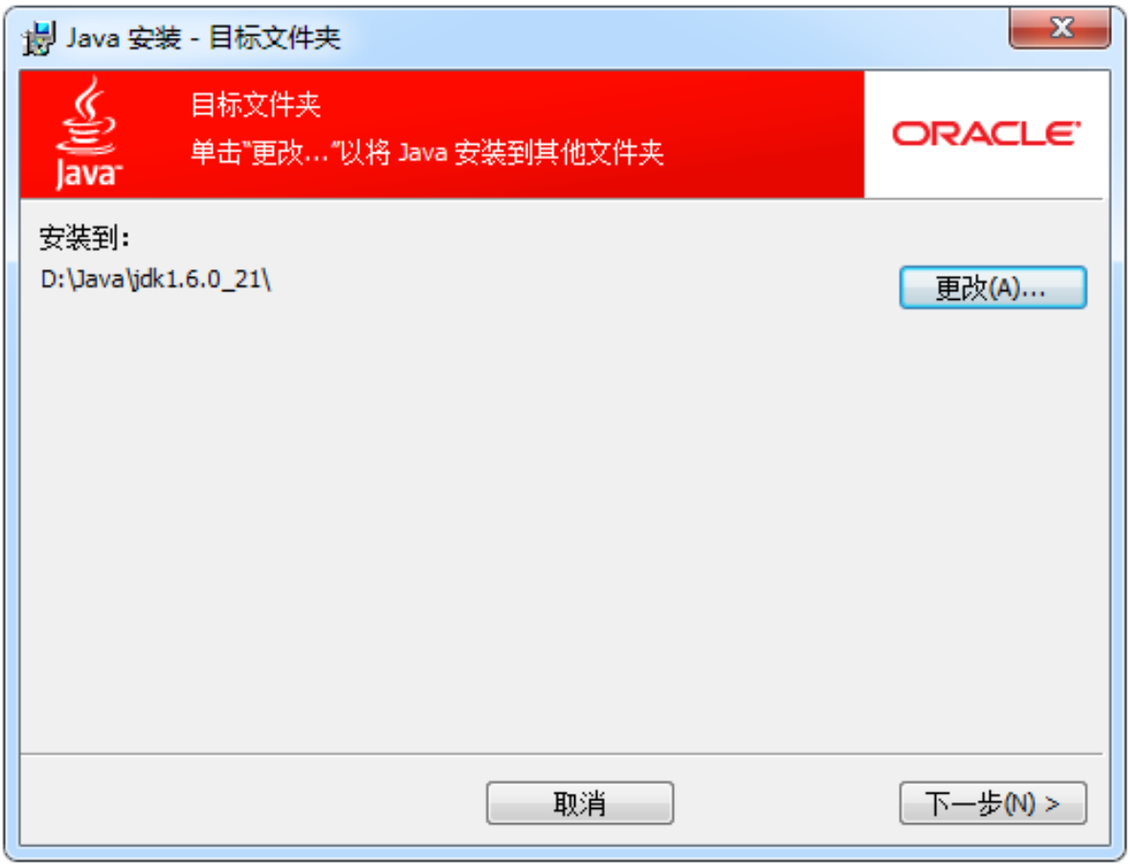
Windows x64对应的是windows 64位系统。

3.2.2 安装JDK

1. 双击“jdk-6u21-windows-i586.exe”文件，点击“下一步”。

2. 继续点击“下一步”。

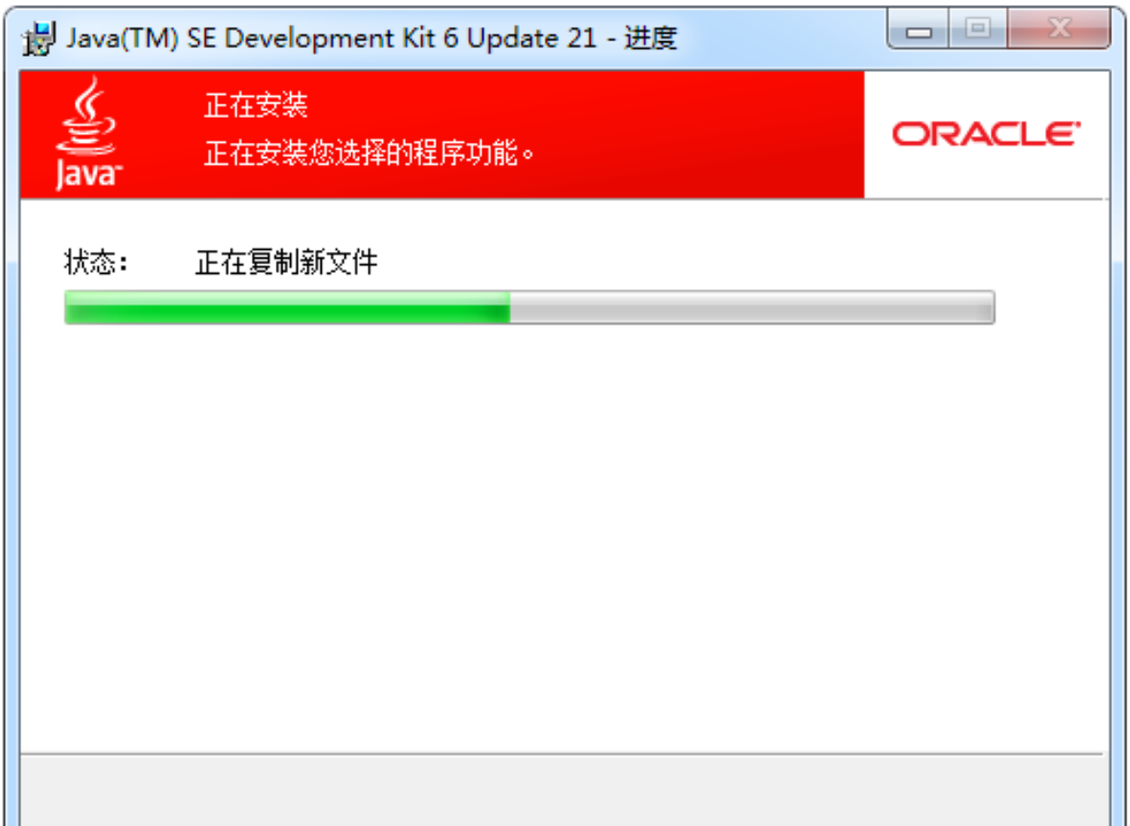
默认安装目录为“C:\Program Files（x86）\Java\jdk1.6.0_21\”，可以通过“更改”按钮对安装路径进行自定义至D:\Java\jdk1.6.0_21\路径下。



P.S.

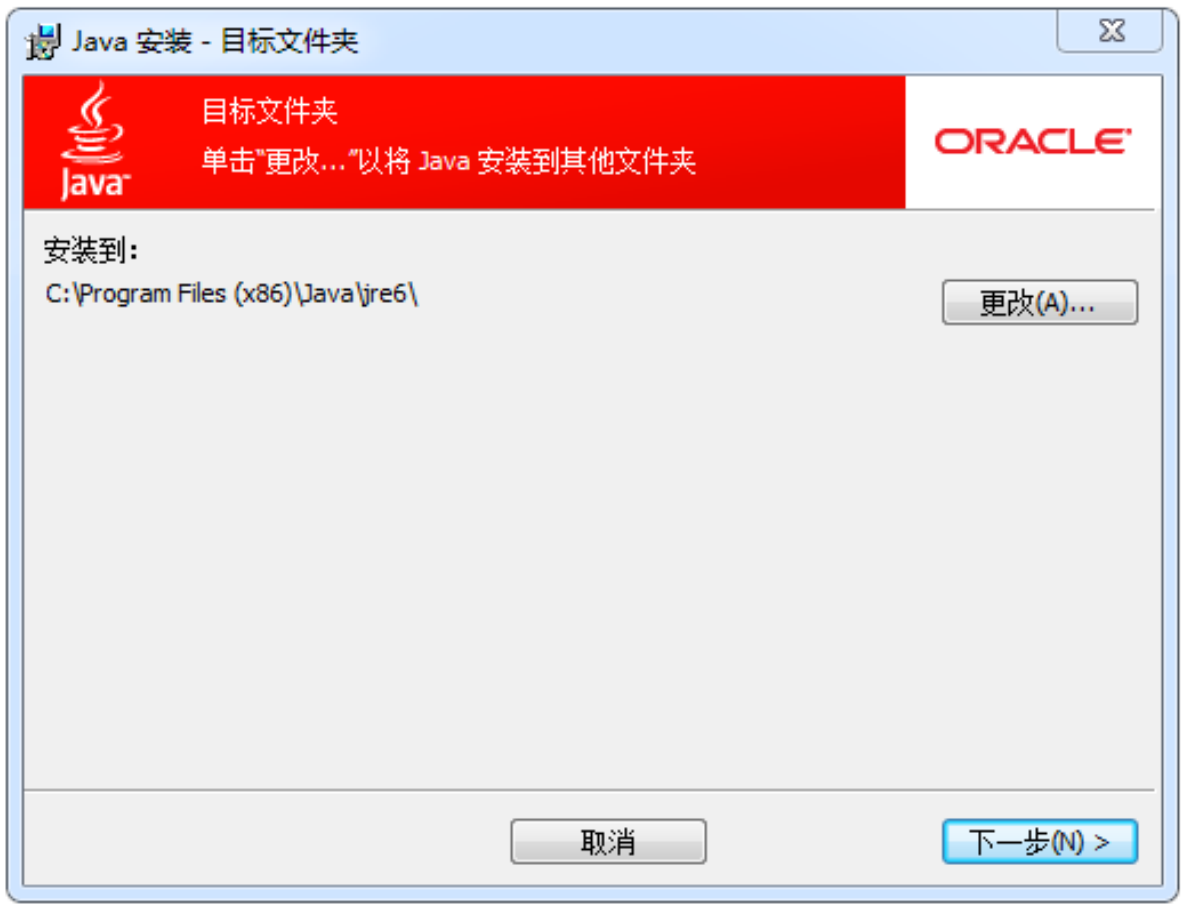
安装路径中不要有中文或者特殊符号如空格等，否则后期开发中可能出现一些莫名其妙的错误。

3. 继续点击“下一步”。



正在安装中...复制文件结束，安装JDK完毕。

4. 接下来出现的对话框是询问是否安装JRE（Java运行环境），这是可选的，因为JDK中已经包含开发环境和运行环境（JRE）两部分。所以不需要安装，一般情况下可以直接点击“取消”按钮。



P.S.

如果你只运行已有的Java程序而不要进行Java程序的开发，那么只需安装JRE即可（因为JRE的体积较小）。

如果点击下一步：

5. 点击“关闭”按钮，安装完毕。

P.S.

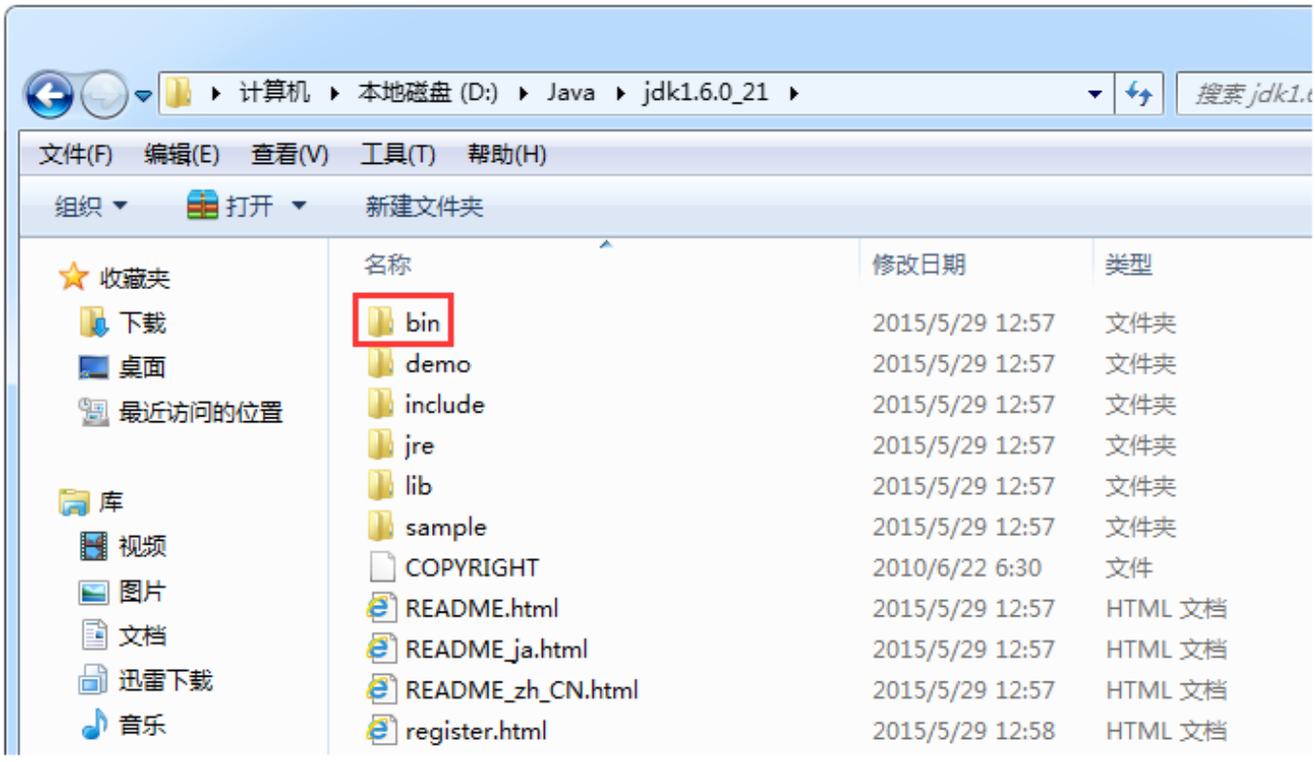
1、JDK无需每次都安装，因为其本身就是绿色版本，可以直接存入U盘，在任何计算机

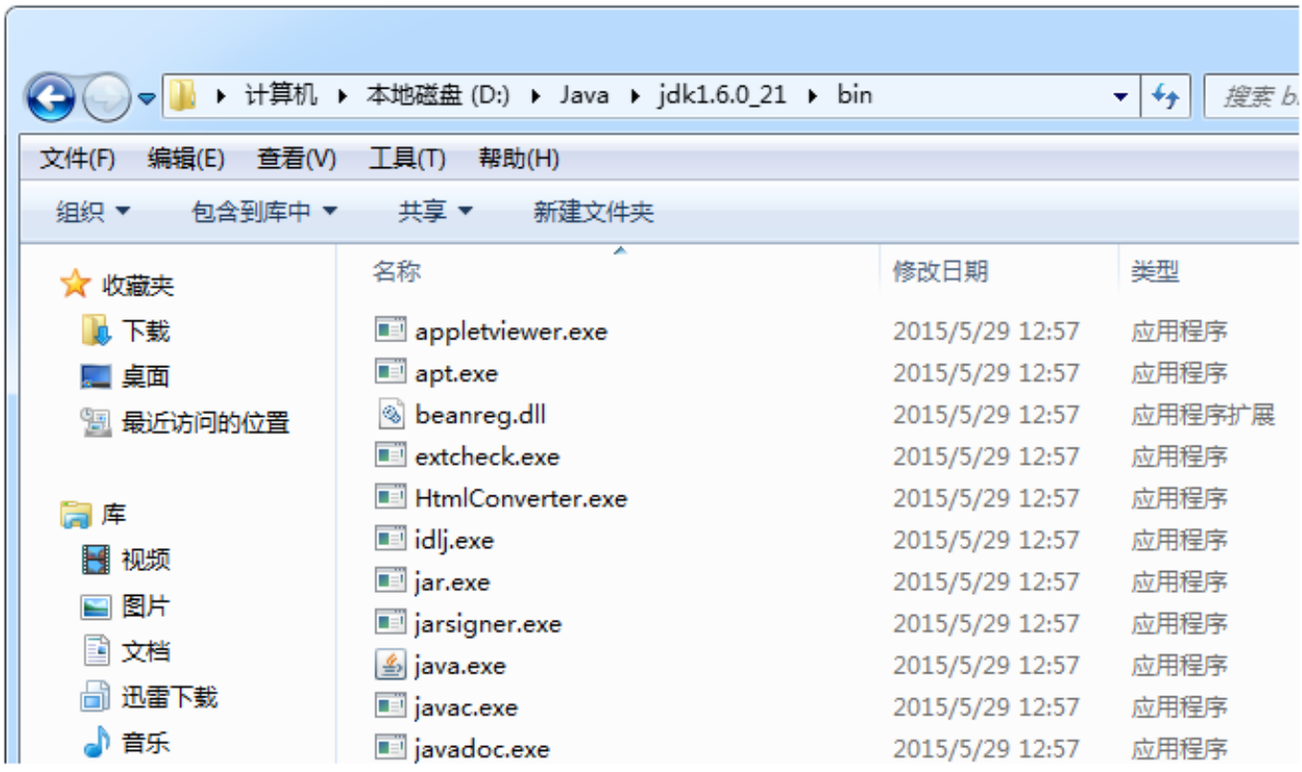
上都可以直接使用。

2、采用安装的方式使用JDK的好处在于其会在注册表中被注册，当JDK出现新版本，会自动更新。

3.3 JDK中的命令行工具

1. JDK安装目录下的bin（binary）文件夹是二进制文件所在的目录，其中的exe文件（都是使用java语言编写）都是开发Java程序所需要的命令文件。



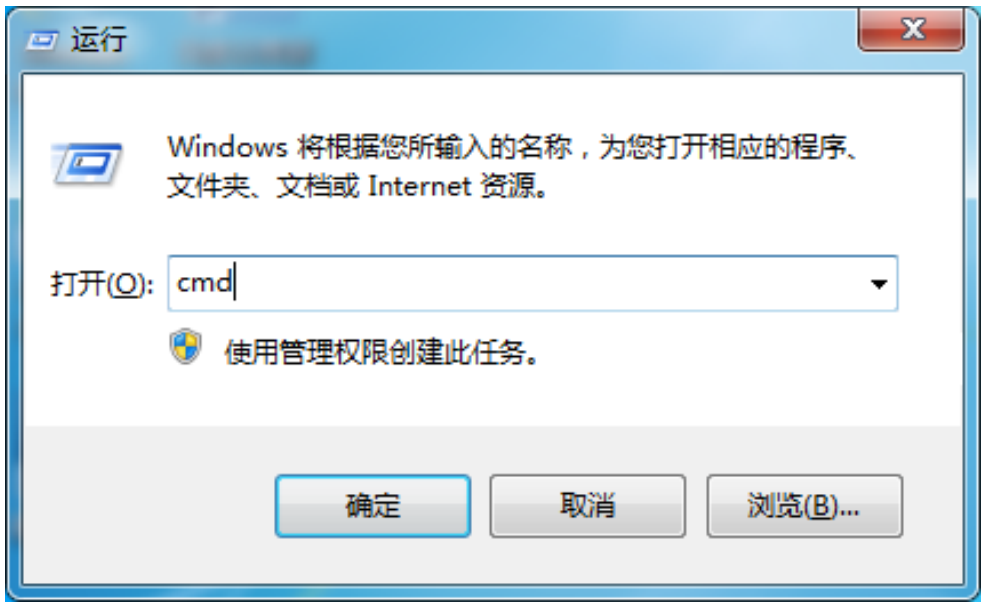


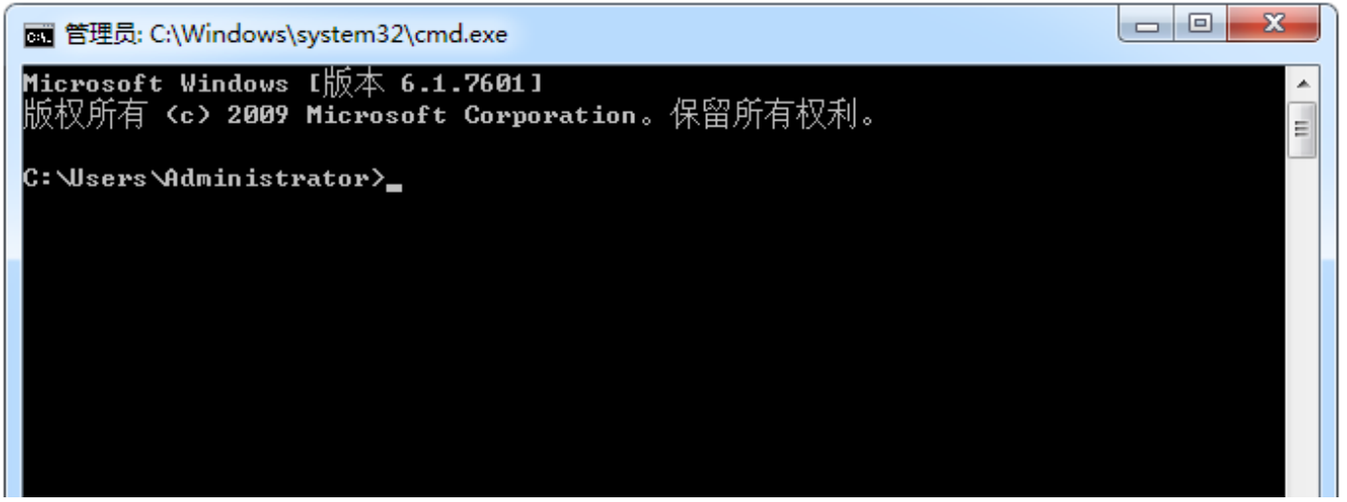
P.S.

这些命令文件并非是图形化方式操作的（双击执行是无效的），而是命令行方式操作的命令文件，所以需要首先打开命令行窗口。

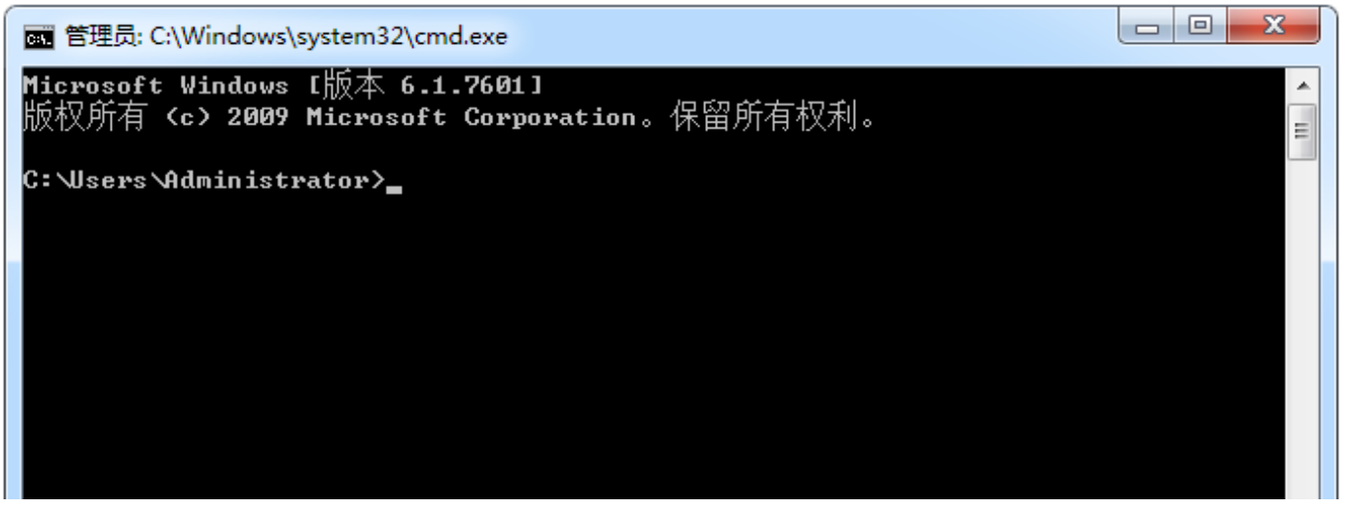
2. 打开DOS命令行窗口有两种方式。

①点击“开始”-->“运行”-->输入“cmd”-->按下“Enter”键。





②点击“开始”-->“所有程序”-->“附件”-->“命令提示符”。



3. 进入到JDK安装目录下的bin目录。

```
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>D:

D:\>cd Java

D:\Java>cd jdk*

D:\Java\jdk1.6.0_21>cd bin

D:\Java\jdk1.6.0_21\bin>
```

```
管理员: C:\Windows\system32\cmd.exe

D:\Java\jdk1.6.0_21\bin>dir
 驱动器 D 中的卷没有标签。
 卷的序列号是 0001-7E71

 D:\Java\jdk1.6.0_21\bin 的目录

2015/05/29  12:57    <DIR>          .
2015/05/29  12:57    <DIR>          ..
2015/05/29  12:57            27,648 appletviewer.exe
2015/05/29  12:57            27,648 apt.exe
2015/05/29  12:57            30,208 beanreg.dll
2015/05/29  12:57            27,648 extcheck.exe
2015/05/29  12:57            49,152 HtmlConverter.exe
2015/05/29  12:57            27,648 idlj.exe
2015/05/29  12:57            27,648 jar.exe
2015/05/29  12:57            27,648 jarsigner.exe
2015/05/29  12:57            27,648 java-rmi.exe
2015/05/29  12:57           139,264 java.exe
2015/05/29  12:57            27,648 javac.exe
```

4. 输入javac(.exe可写可不写)，出现如下显示，说明JDK已经可以使用了。

```
管理员: C:\Windows\system32\cmd.exe

D:\Java\jdk1.6.0_21\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
  -nowarn                    不生成任何警告
  -verbose                   输出有关编译器正在执行的操作的消息
  -deprecation               输出使用已过时的 API 的源位置
  -classpath <路径>            指定查找用户类文件和注释处理程序的位置
  -cp <路径>                   指定查找用户类文件和注释处理程序的位置
  -sourcepath <路径>           指定查找输入源文件的位置
  -bootclasspath <路径>        覆盖引导类文件的位置
  -extdirs <目录>              覆盖安装的扩展目录的位置
  -endorseddirs <目录>         覆盖签名的标准路径的位置
  -proc:{none,only}          控制是否执行注释处理和/或编译。
  -processor <class1>[,<class2>,<class3>...]要运行的注释处理程序的名称; 绕过默认
的搜索进程
```

3.4 命令行简介

课程中常见的DOS命令：

dir：列出当前目录下的文件及文件夹。

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。

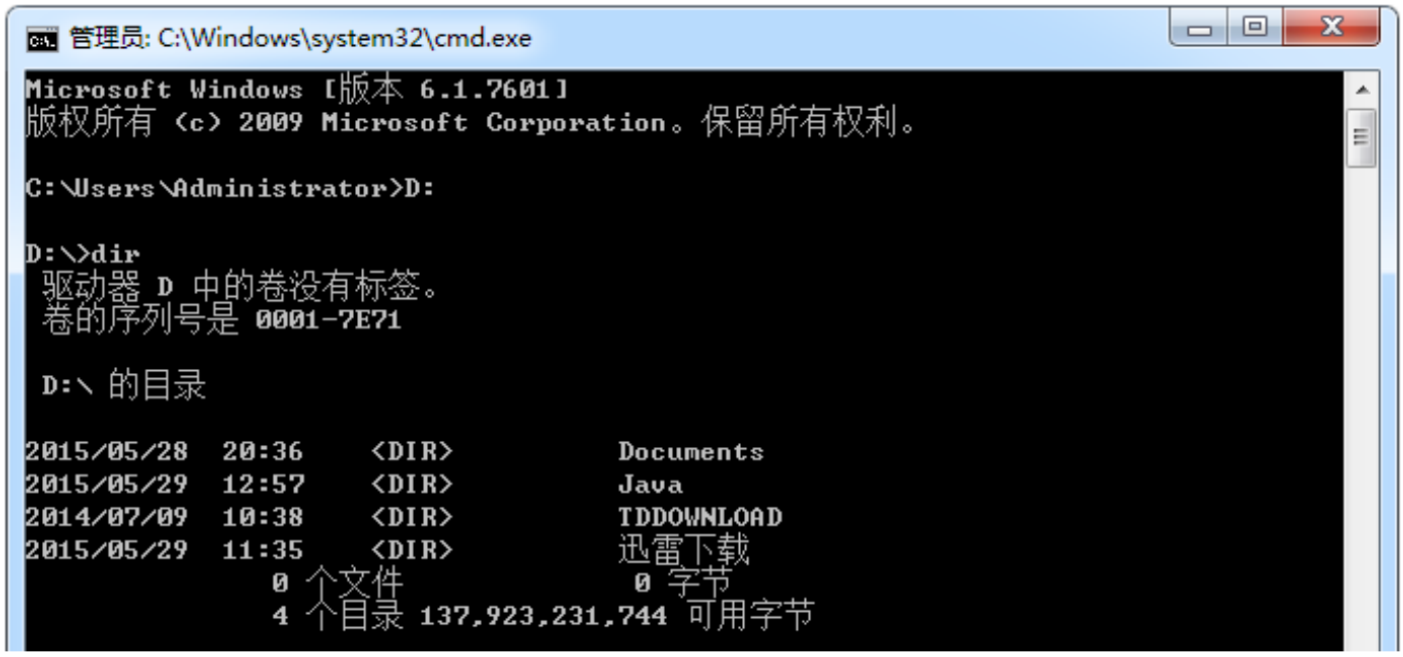
cd \：退回到根目录。

del：删除文件。

exit：退出DOS命令行。

12个示例轻松掌握常用DOS命令：

1. 查看D盘下的目录。



2. 查看某个文件夹下的目录。

```
管理员: C:\Windows\system32\cmd.exe

D:\>cd Java

D:\Java>dir
 驱动器 D 中的卷没有标签。
 卷的序列号是 0001-7E71

 D:\Java 的目录

2015/05/29  12:57    <DIR>          .
2015/05/29  12:57    <DIR>          ..
2015/05/29  12:58    <DIR>          jdk1.6.0_21
               0 个文件              0 字节
               3 个目录 137,923,231,744 可用字节
```

3. 也可以通过通配符的方式更方便地进入某个文件夹。

```
管理员: C:\Windows\system32\cmd.exe

D:\Java>cd jdk*

D:\Java\jdk1.6.0_21>dir
 驱动器 D 中的卷没有标签。
 卷的序列号是 0001-7E71

 D:\Java\jdk1.6.0_21 的目录

2015/05/29  12:58    <DIR>          .
2015/05/29  12:58    <DIR>          ..
2015/05/29  12:57    <DIR>          bin
2010/06/22  06:30             3,409 COPYRIGHT
2015/05/29  12:57    <DIR>          demo
2015/05/29  12:57    <DIR>          include
2015/05/29  12:57    <DIR>          jre
```

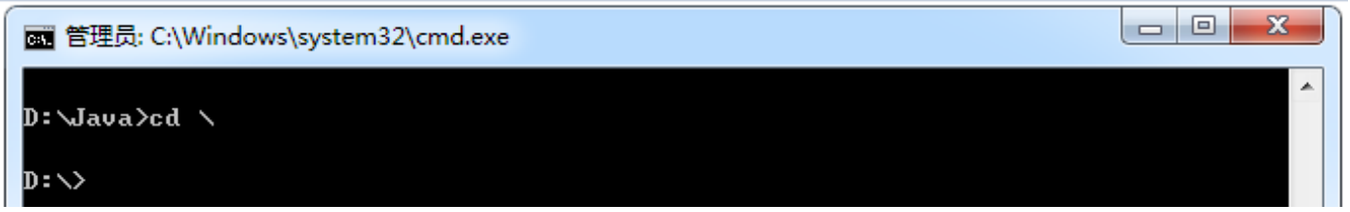
P.S.

"cd jdk*"命令中的“*”是一种通配符，如果当前目录中有多个文件夹名称都能匹配，那么会自动进入第一个名称匹配的文件夹中。

4. 退回到上一级目录。

```
管理员: C:\Windows\system32\cmd.exe

D:\Java>cd jdk*

D:\Java\jdk1.6.0_21>cd ..

D:\Java>_
```

5. 直接进入D盘根目录。



6. 在D盘下根目录下创建新的文件夹。

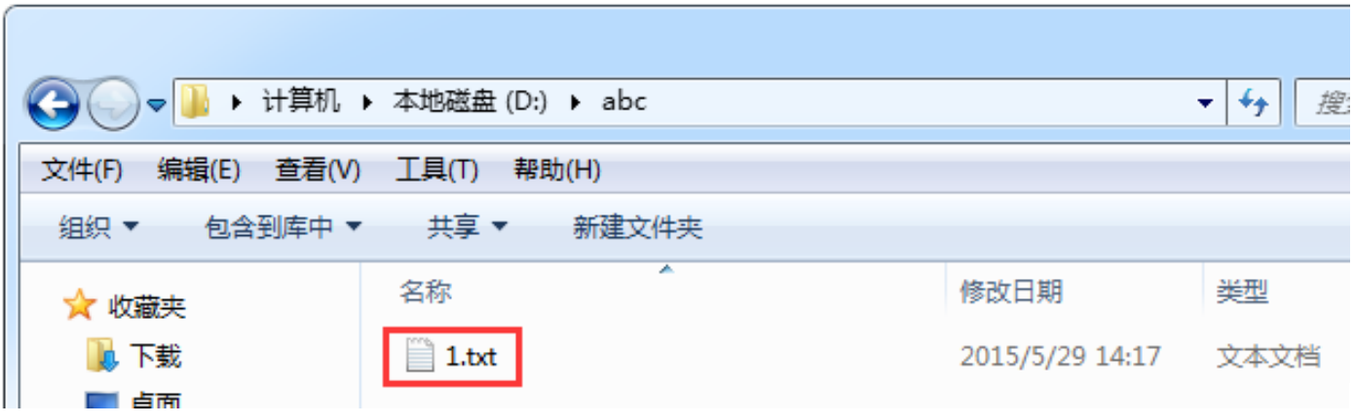
P.S.

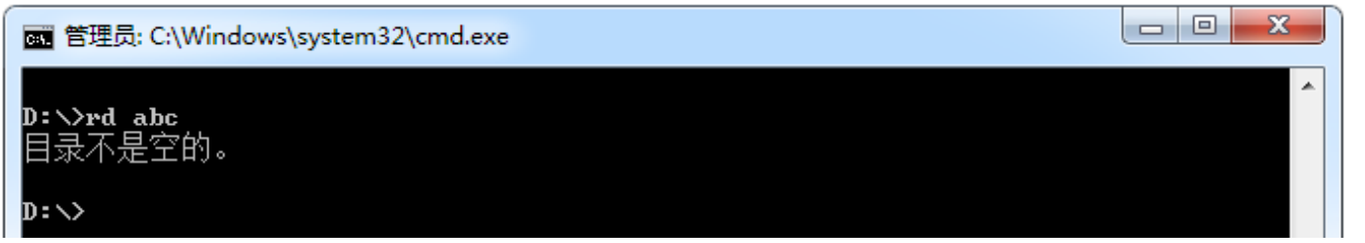
早期操作计算机用DOS命令操作，因为需要记住很多命令，不利于普及，后期发展成图形界面，通过鼠标点击界面的形式操作计算机，底层实际上运行的依然是DOS命令。

7. 删除文件夹。

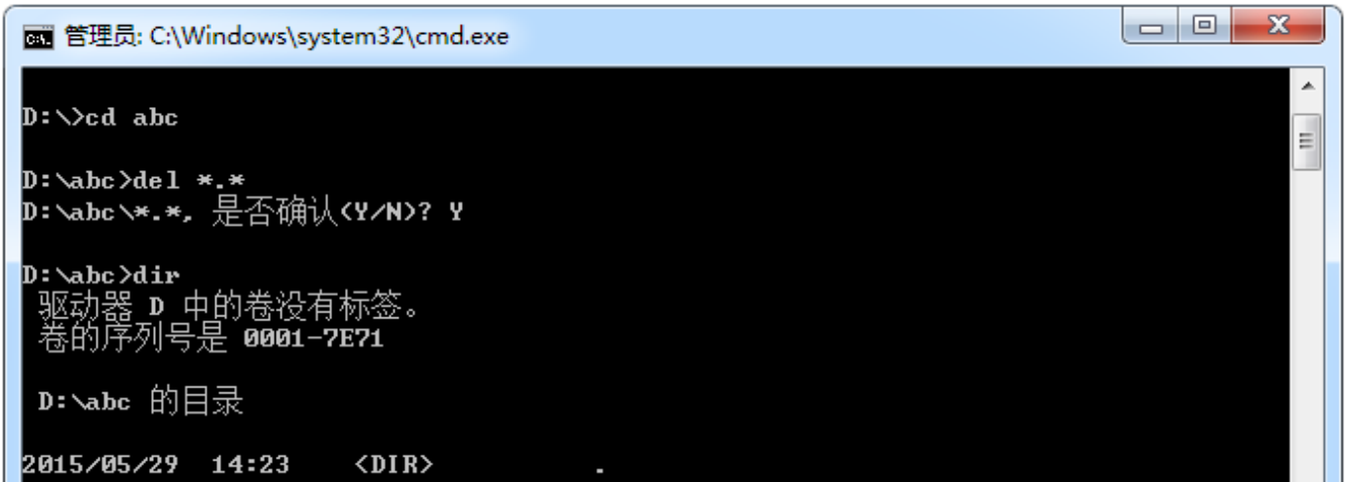
abc文件夹已经被删除：

8. 新创建一个文件夹，并在该文件夹目录下创建一个文件。由于该文件夹下不是空的，删除该文件夹失败。这是Windows为了防止误删除采取的机制。





9. 只有将该文件夹目录下所有文件及文件夹清空，才能成功删除该文件夹。



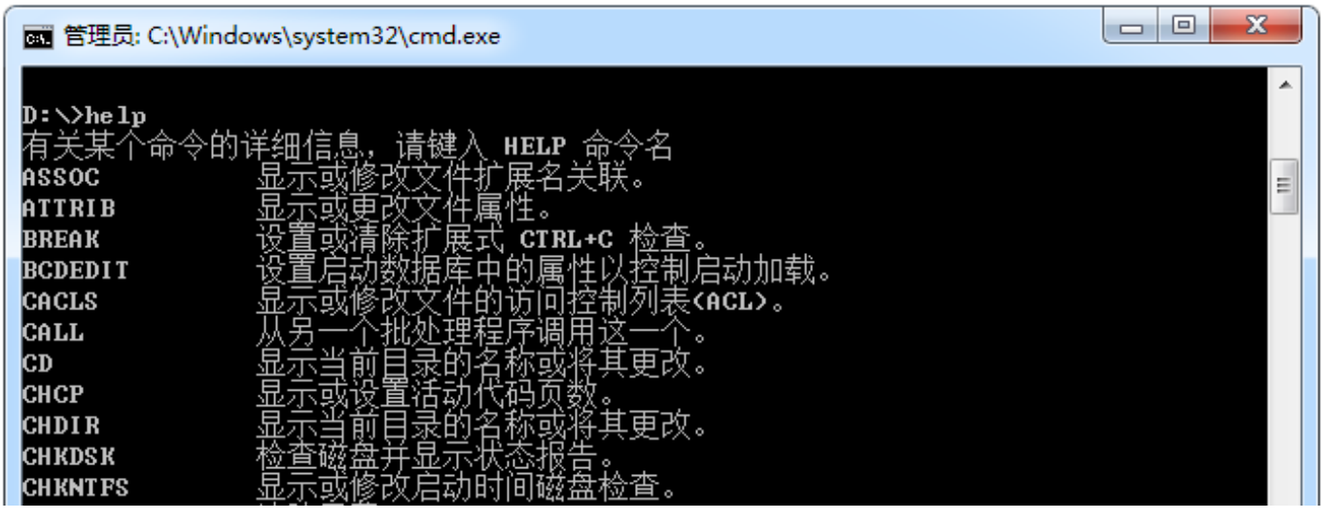
```
2015/05/29  14:23    <DIR>          ..
               0 个文件              0 字节
               2 个目录 137,923,231,744 可用字节

D:\abc>cd ..

D:\>rd abc
```

P.S.

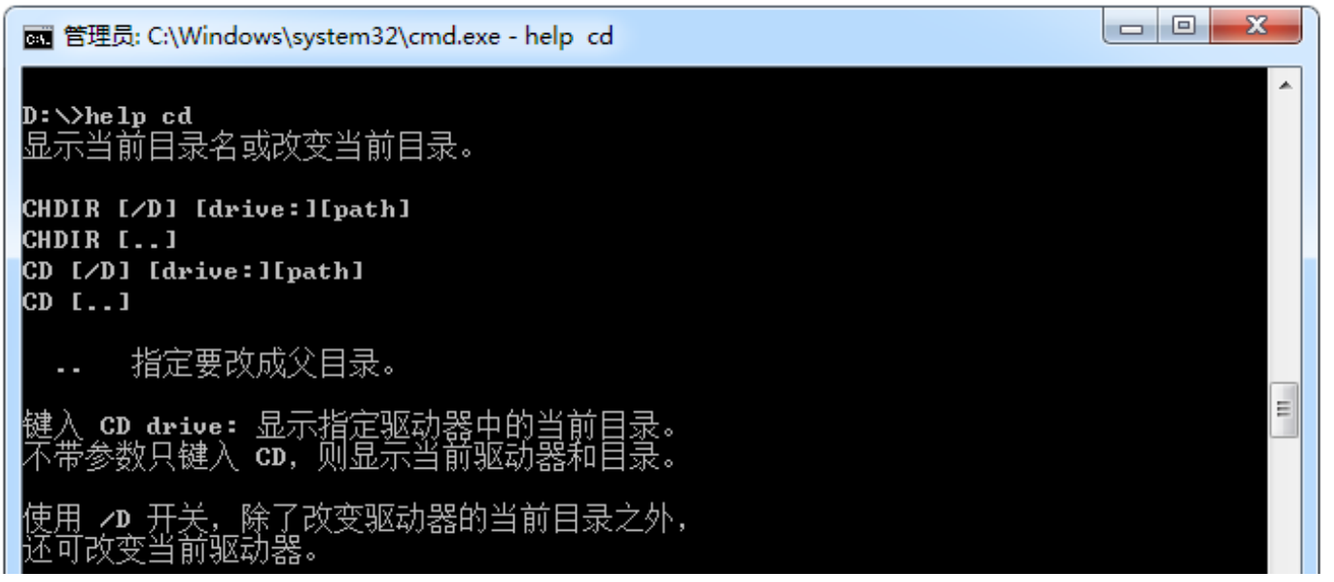
1、 “del *”或“del *.*”命令表示删除当前目录下所有的文件。

2、Windows系统删除文件是从内向外删，文件夹内部的文件如果没有被完全删除，文件夹删除就必定会失败。

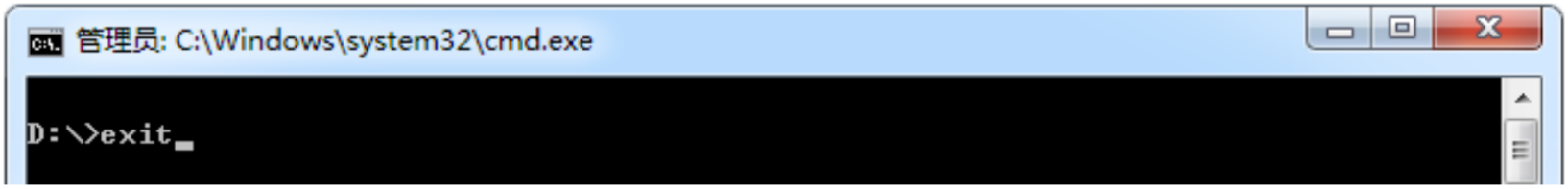
10. 输入help命令，可以查询到所有的命令功能。



11. 输入help命令，也可以查询某个命令的功能。



12. 输入exit命令，可以退出DOS命令行窗口。



3.5 环境变量配置

为了能在DOS命令行窗口中，在任何目录下都能够执行javac命令，就需要将javac.exe命令文件所在目录的路径放入path环境变量中。

原理：

在DOS命令提示符窗口中输入某个命令后，Windows系统会首先在当前目录下查找是否存在该命令文件可以执行。如果没有，Windows系统就会在path环境变量路径中查找。如果查找到，就会执行该命令。如果还没有找到，那么就会提示如下信息。

3.5.1 配置环境变量的具体步骤如下：

1. 右击“计算机”--> “属性”。

2. 点击“高级系统配置”--> “高级”选项卡--> “环境变量”。

3. 将javac命令文件所在目录的路径放入path路径中。

P.S.

1、一定要将javac命令文件所在目录的路径放入path环境变量的开头。否则，如果计算机上以前已经安装了其他版本的JDK并且也已经配置了path环境变量，那么先执行的javac命令就是以前的JDK安装目录下的javac命令。因为windows系统是按照path路径从头向后搜索各个目录的。

2、环境变量中的各个路径之间一定要以分号进行分隔。

3、设置path环境变量的时候千万不要删除掉原来的内容。否则，很多程序将无法运行。

4. 点击确定。然后，一定要新开一个新的DOS命令行窗口，再输入javac命令，如果出现如下显示，说明path环境变量配置成功。

```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>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
  -nowarn                    不生成任何警告
  -verbose                   输出有关编译器正在执行的操作的消息
  -deprecation               输出使用已过时的 API 的源位置
  -classpath <路径>           指定查找用户类文件和注释处理程序的位置
  -cp <路径>                  指定查找用户类文件和注释处理程序的位置
  -sourcepath <路径>          指定查找输入源文件的位置
```

P.S.

如果一台计算机上安装了多个版本的JDK，通过javac -version可以确定当前使用的JDK版本。

```
C:\Users\Administrator>javac -version
javac 1.6.0_21
```

3.5.2 环境变量配置技巧

按照以上步骤配置path环境变量，会很容易产生因JDK安装目录的改变，而不断修改path环境变量，从而导致对path环境变量的误操作，风险很大。可以通过如下技巧解决：

1. 新创建一个环境变量JAVA_HOME记录jdk安装目录的路径。

2. 在path环境变量中通过“%%”动态的获取JAVA_HOME的值即可。

通过这种方式，如果JDK安装目录改变了，那么只需要修改JAVA_HOME环境变量即可，而不用再修改path环境变量。

P.S.

%JAVA_HOME%表示动态获取名称为JAVA_HOME环境变量的值。

3.5.3 环境变量临时配置方式

如果在别人的计算机上进行Java程序的开发，设置path这样的系统环境变量就不太好，那么就可以采用设置临时环境变量的方式（通过DOS命令中set命令完成）。

1. 用set命令查看本机的所有环境变量的信息。

```
C:\Users\Administrator>set
ALLUSERSPROFILE=C:\ProgramData
APPDATA=C:\Users\Administrator\AppData\Roaming
CommonProgramFiles=C:\Program Files\Common Files
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
CommonProgramW6432=C:\Program Files\Common Files
COMPUTERNAME=PC-20140621KFIX
ComSpec=C:\Windows\system32\cmd.exe
FP_NO_HOST_CHECK=NO
HOMEDRIVE=C:
HOMEPATH=\Users\Administrator
LOCALAPPDATA=C:\Users\Administrator\AppData\Local
LOGONSERVER=\\PC-20140621KFIX
NUMBER_OF_PROCESSORS=4
OS=Windows_NT
Path=D:\Java\jdk1.6.0_21\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\
```

2. 用set命令（"set 命令名"）查看具体某一个环境变量的值。

```
C:\Users\Administrator>set path
Path=D:\Java\jdk1.6.0_21\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
```

3. 用set命令（"set 变量名="）清空一个环境变量的值。

```
C:\Users\Administrator>set path=

C:\Users\Administrator>set path
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
```

4. 用set命令（"set 变量名=具体值"）给指定环境变量定义具体值。

```
C:\Users\Administrator>set path=D:\Java\jdk1.6.0_21\bin

C:\Users\Administrator>set path
Path=D:\Java\jdk1.6.0_21\bin
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
```

5. 想要在原有环境变量值基础上添加新值。

首先，通过“%变量名%”操作符获取到原有环境变量的值。然后，追加上新值，再赋值给该变量名即可。

```
C:\Users\Administrator>set path=D:\Java\jdk1.6.0_21\bin;%path%

C:\Users\Administrator>set path
Path=D:\Java\jdk1.6.0_21\bin;D:\Java\jdk1.6.0_21\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
```

P.S.

1、临时配置环境变量的方式只在当前DOS命令行窗口有效。窗口关闭，配置即消失。

2、cls是清屏命令。

```
%ERRORLEVEL% - 扩展到当前 ERRORLEVEL 数值。
%CMDEXTVERSION% - 扩展到当前命令处理器扩展版本号。
%CMDCMDLINE% - 扩展到调用命令处理器的原始命令行。
%HIGHESTNUMANODENUMBER% - 扩展到此计算机上的最高 NUMA 节点号。
C:\Users\Administrator>cls
```

```
C:\Users\Administrator>
```

四、Java程序开发体验

4.1 编写运行Hello World

.java文件 → javac.exe 编译 → .class文件 → java.exe 运行 → 结果

1. 选择最简单的编辑器：记事本。将Java代码编写入扩展名为.java的文件（源文件）中。

```
class Demo{
}
```

Hello World：代表学习计算机语言的第一个入门小程序。现在泛指接触任何新事物的第一步。

Class：是java中的关键字，用于定义类，java语言的程序代码都需要定义在类中。

关键字：被java语言赋予了特殊含义的单词。

Demo：为了方便使用这个类，给类自定义的类名。

{}：定义该类中代码的范围。

P.S.

1、写代码，阅读性第一，功能性第二，一定要注意写代码的格式！

2、源文件名和类名可以不一致，但当class前有修饰符public时，则必须一致。

2. 通过javac命令对该java文件进行编译。

①有了java源文件，将其编译成JVM可以识别的文件。

②在该源文件目录下，通过javac编译工具对Demo.java文件进行编译。

③如果程序没有错误，没有任何提示，就会在当前目录下出现一个Demo.class文件，该文件称为字节码文件，也就是可以执行的java的程序。

```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>D:

D:\>cd code\day01

D:\code\day01>javac Demo.java
```

3. 通过java命令对生成的class文件进行运行。

①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。

```
D:\code\day01>java Demo
Exception in thread "main" java.lang.NoSuchMethodError: main
```

P.S.

使用java命令运行class文件，没有必要加上.class文件后缀。

②因为一个程序的执行需要一个起始点或者入口，所以在Demo类中加入如下代码。

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

③对修改后的Demo.java源文件需要重新编译，生成新的class文件后，再执行。

发现编译成功，但也没有任何效果，因为程序并没有告诉JVM要帮我们做什么事情，也就是没有可以具体执行的语句。

```
D:\code\day01>javac Demo.java

D:\code\day01>java Demo
```

④如果想要和JVM来个互动，只要在main方法中加入一句System.out.println("hello java");。因为对程序进行了改动，所以需要再重新编译，然后运行即可。

```
class Demo{
        public static void main(String[] args){
                System.out.println("hello world");
        }
}
```

```
D:\code\day01>javac Demo.java

D:\code\day01>java Demo
hello world
```

P.S.

1. main方法的作用在于保证一个类可以独立运行，因为它是程序的入口。

2. System.out.print();语句用于告诉系统输出打印数据，可以将括号中的内容打印在控制台上。通过该语句可以直接在控制台看到jvm运行java程序后的结果。

3. System.out.println();语句与System.out.print();语句的区别是，前者打印后并且换行。

```
class Demo{
        public static void main(String[] args){
                System.out.println("hello world");
                System.out.print("haha");
                System.out.println("hehe");
        }
}
```

```
D:\code\day01>javac Demo.java

D:\code\day01>java Demo
hello world
hahahehe
```

4、javac命令的作用是对java程序进行语法性检查，一旦出错，就会打印出错误信息。

```
class Demo{
        public static void main(String[] args){
                System.out.println("hello world")
        }
}
```

```
D:\code\day01>javac Demo.java

D:\code\day01>java Demo
hello world
hahahehe

D:\code\day01>javac Demo.java
Demo.java:3: 需要 ';'
                System.out.println("hello world")
                                                 ^
1 错误
```

由于System.out.println("hello java")语句后面缺少分号，所以报错。根据报错的行数，找到错误的地方，进行修改。然后重编编译，运行。

P.S.

需要注意的是java会出现连锁错误反应，也就是一个错误，会导致后面程序的多个错误。这时候只需要修改第一个错误，后面的错误都会解决掉。

5、java命令可以启动JVM，然后找到相应的class文件，再寻找程序入口，也就是main方法，然后调用该方法执行java程序。

4.2 常见错误信息-找不到文件

```
D:\code\day01>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
```

错误可能原因：

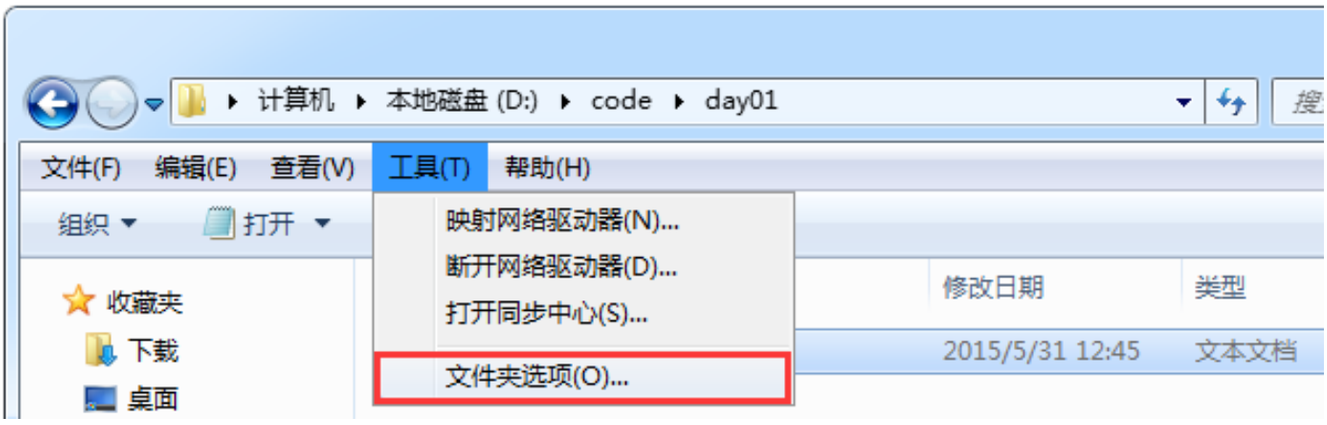
1、文件名写错。

2、类文件不在当前路径下或者不在classpath（后面会将讲到）指定路径下。

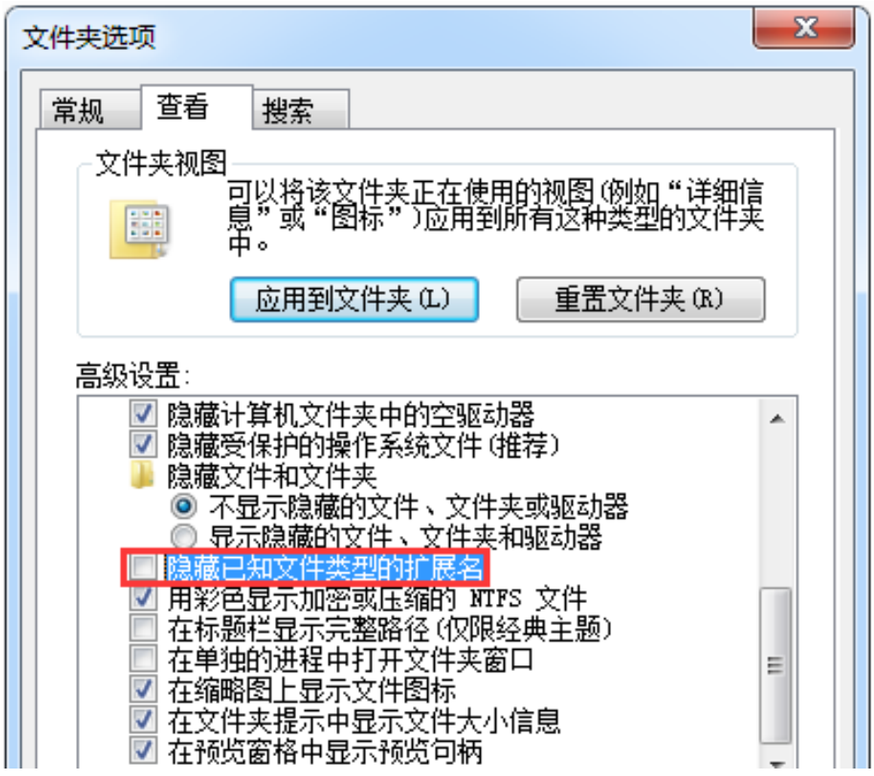
3、后缀名问题。

“后缀名问题”查找及解决方案如下：

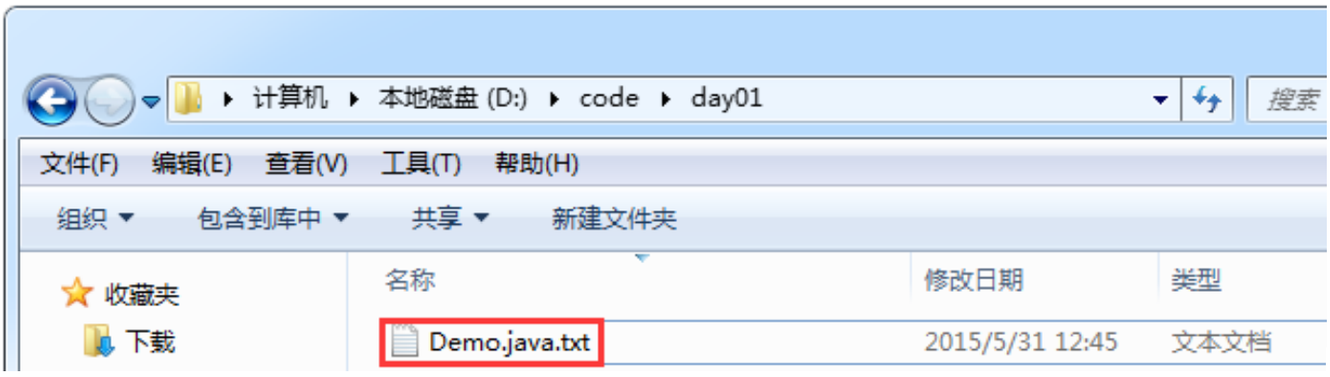
1. 点击“工具”--> “文件夹选项”。



2. 点击“查看”选项卡，并且勾选掉“隐藏已知文件类型的扩展名”。



3. 然后就可以看到文件真正的后缀名为txt，所以找不到Demo.java文件。只需将修改后缀名为java，再重新编译、运行即可。



4.3 配置classpath环境变量

由于可能频繁执行多个class文件，并且多个class文件可能存储在不同的目录下，那么每次都在命令提示符窗口中切换目录会相当的麻烦。

classpath环境变量的作用类似于path环境变量，但是它的作用在于告诉JVM去哪里找到class文件。

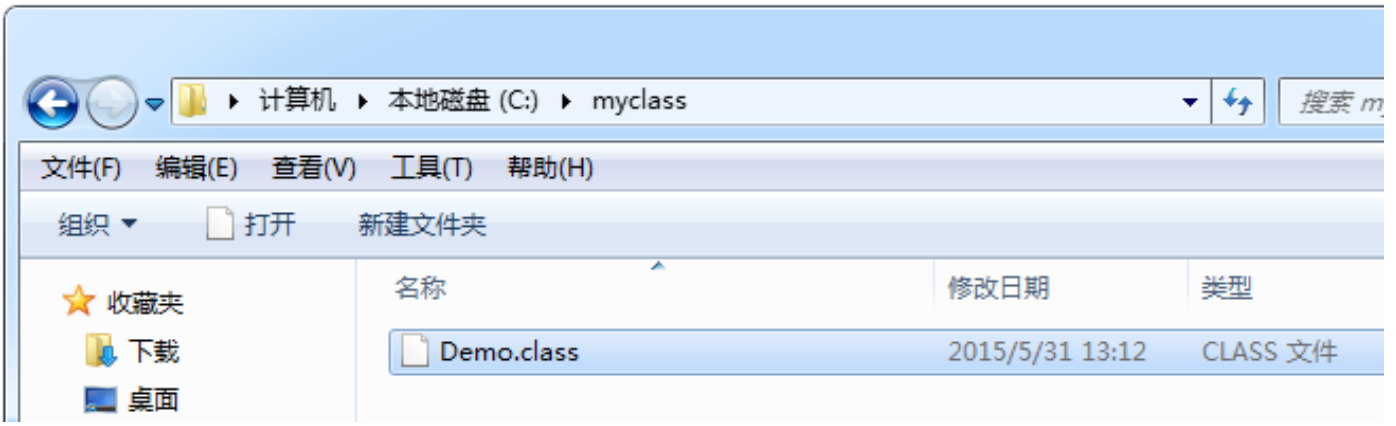
JVM查找类文件的顺序：

1. 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。

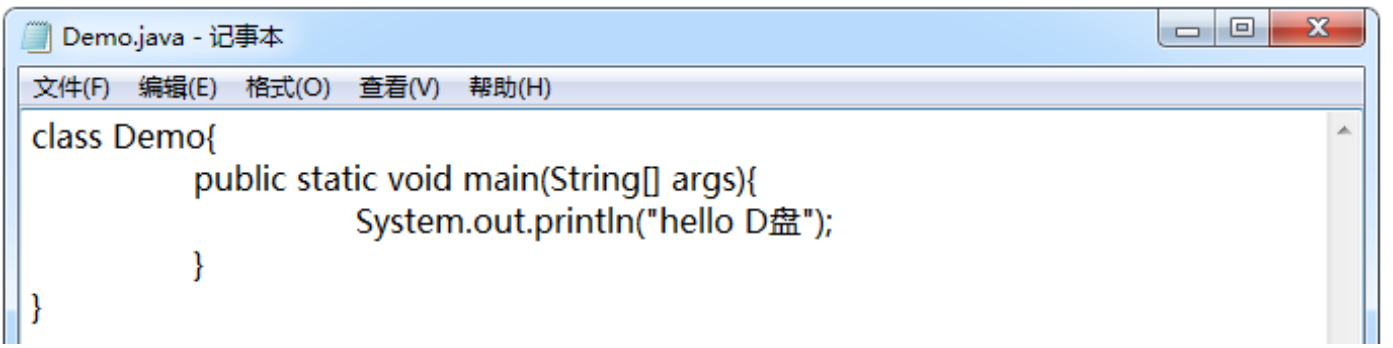
2. 如果配置了classpath环境，JVM会先在classpath环境变量值的目录中查找要运行的类文件。

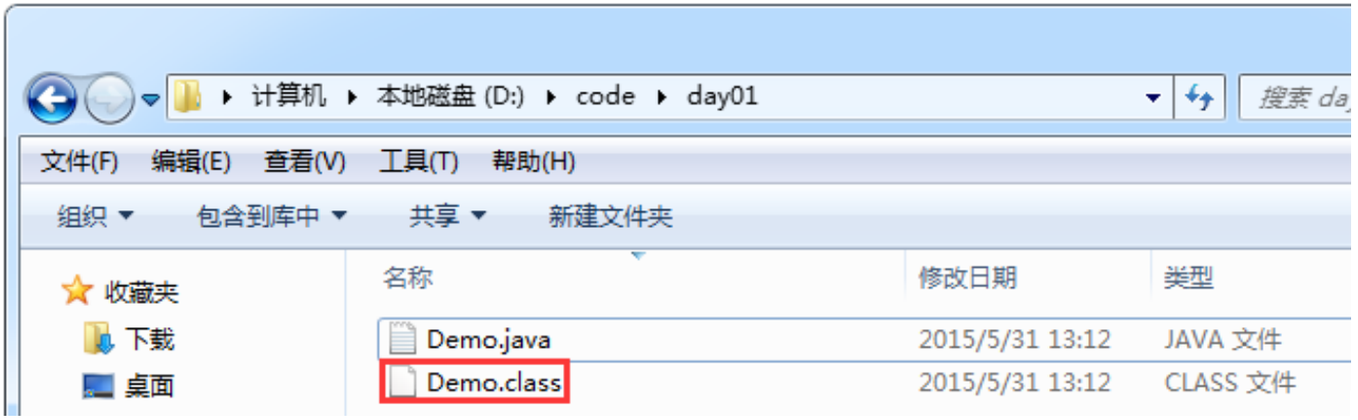
示例：

1、在C盘根目录下新创建一个文件夹，命名为myclass，将D:\code\day01目录下的Demo.class文件剪切到此文件夹下。

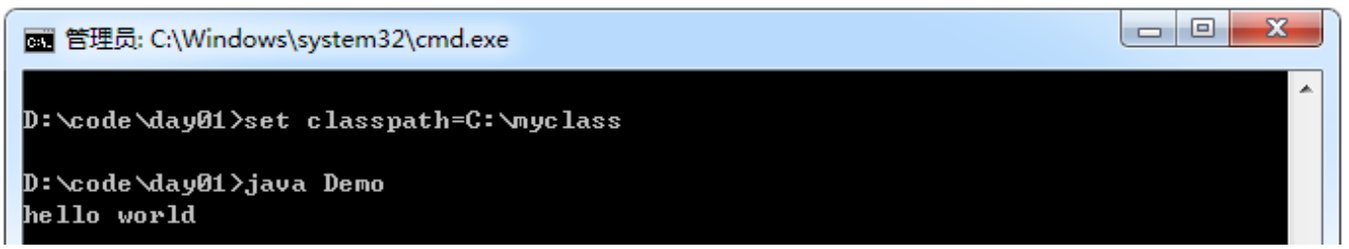


2、将D:\code\day01文件夹中的源文件Demo.java修改为在控制台中打印“hello D盘”，重新编译，生成Demo.class。





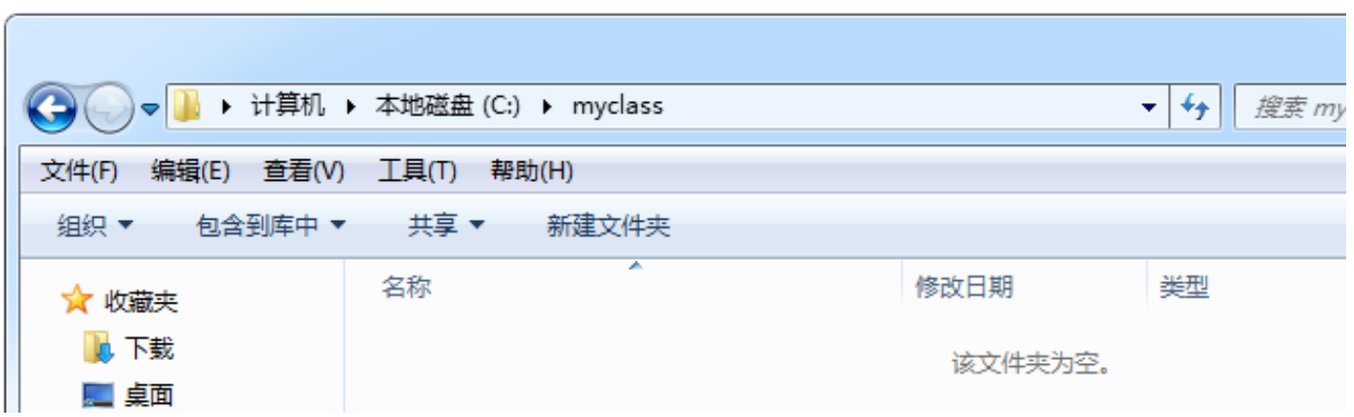
3、 如果想要执行C盘根目录下myclass文件夹中的Demo.class，又不想切换目录，由于JVM会先在classpath环境变量值的目录中查找要运行的类文件，可以通过设置环境变量classpath为“c:\myclass”实现。



3. classpath环境变量值的结尾处如果没有分号，那么JVM在classpath目录下没有找到要指定的类文件，也不会在当前目录下查找。即使当前目录下有，也不会运行。

示例：

1、删除C:\myclass目录下的Demo.class。



2、在D:\code\Day01目录下运行Demo.class文件，报错。

这是因为classpath环境变量已经被赋值，所以即使在D:\code\Day01目录下存在Demo.class文件，JVM也根本不会去查找。

```
管理员: C:\Windows\system32\cmd.exe

D:\code\day01>set classpath
classpath=C:\myclass

D:\code\day01>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.
```

4. classpath环境变量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，会在当前目录下再查找一次。

示例：

为环境变量classpath赋值后加上一分号，然后在D:\code\day01目录下运行Demo.class文件，由下图可见，运行成功。

```
管理员: C:\Windows\system32\cmd.exe

D:\code\day01>set classpath=C:\myclass;

D:\code\day01>java Demo
hello D盘
```

P.S.

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 “.” （代表当前目录）表示，这样可读性更好一些。

示例：

```
管理员: C:\Windows\system32\cmd.exe

D:\code\day01>set classpath=.;%classpath%

D:\code\day01>java Demo
hello D盘
```

~END~

~爱上海，爱黑马~