离线瓦片地图生产部署流程

## 一、下载地图瓦片

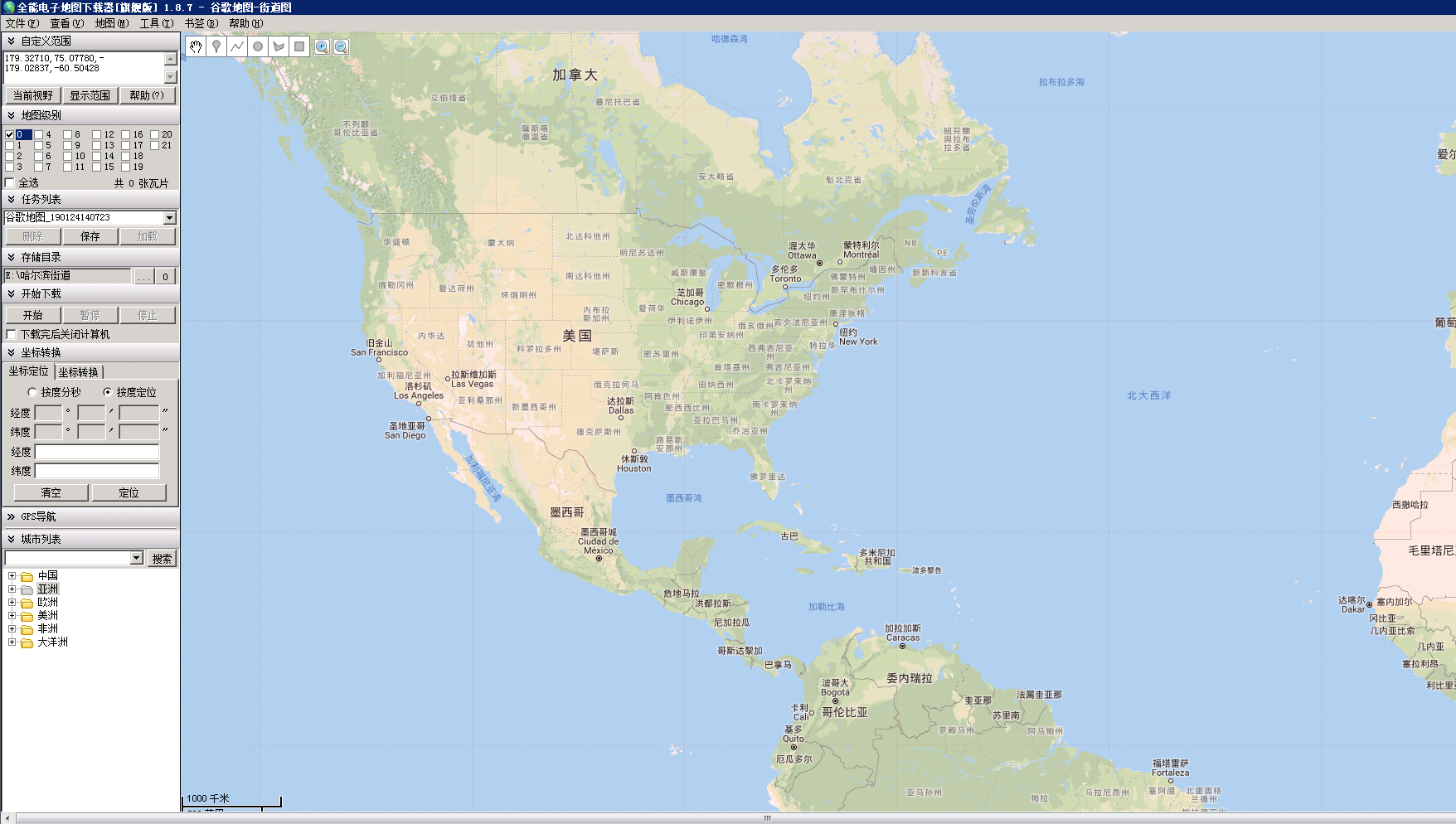
1. 下载瓦片地图，目前网上有一些地图瓦片下载工具，都可以下载png或jpg的瓦片图。

如水经注，全能地图下载器等。

**下图时需要注意：**

一般一个项目，1-6级需要下载全球数据，7-10级需要下载全国数据，11-14需要下载全省数据，15级开始一般是项目所在地区的数据。这样可以保证从小尺度到大尺度浏览都清晰。

因此1-10级的数据基本上是固定的，为了方便，在192.168.10.158(F:\地图瓦片)目录下放了1-10级的谷歌卫星影像和谷歌街道数据，直接拷贝即可，下载只要从11级开始即可。



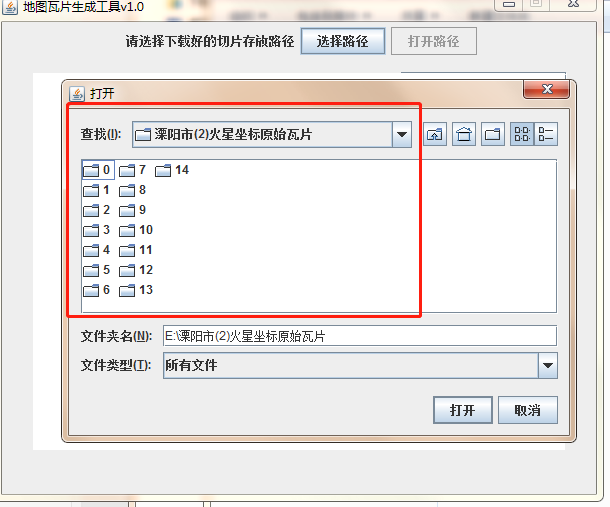
## 二、压缩地图瓦片

利用bundle瓦片生产工具，将下载后的瓦片图片生成bundle文件以供我们uearth使用。

1. 该工具是基于java GUI开发，因此需要java环境(jdk版本为8)。安装java的步骤详见[附录](#_附录：jdk8图解安装)。
2. 在有java环境后，双击tileProject.jar

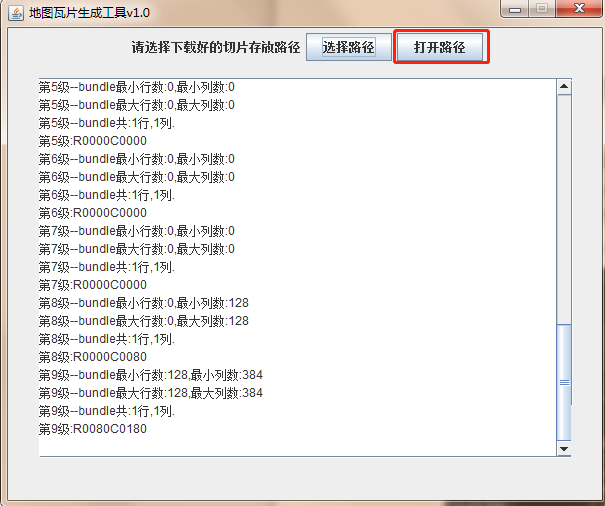


4. 选择瓦片存放路径：

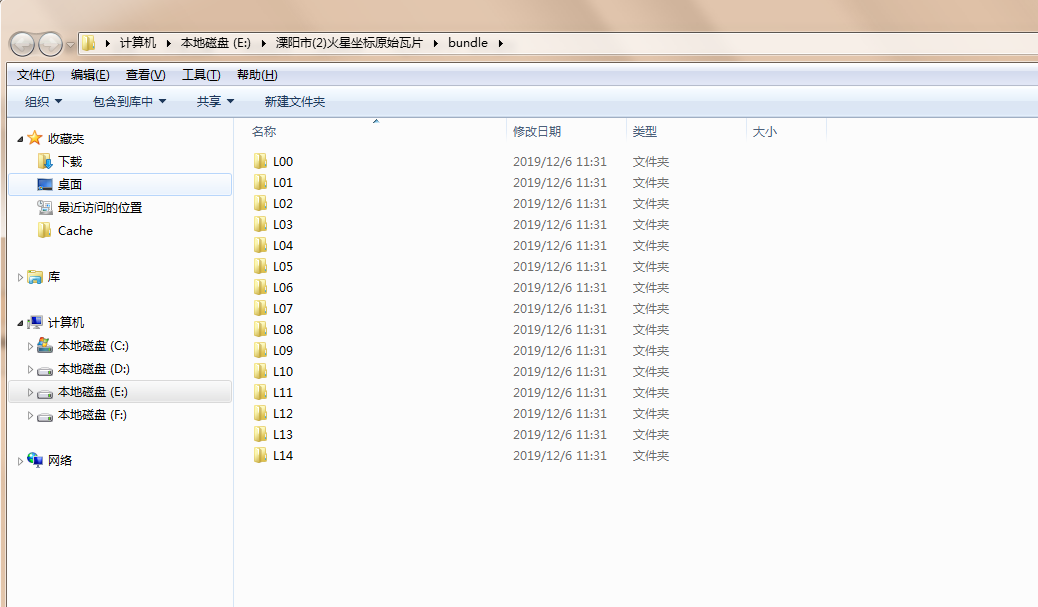


5. 选择完毕后会自动合并生成bundle 成功后会出现如下提示，如果合并过程中有错误，会在控制台打印。

**注意：bundle会在同级目录下生成，工具运行逻辑是读取目录下所有文件，所以运行之前，需保证该目录下只有原生切片，没有已经运行过的bundle文件。**



1. 点击打开路径按钮，会打开瓦片存放路径。bundle瓦片存放路径默认是原瓦片存放路径下的bundle目录。



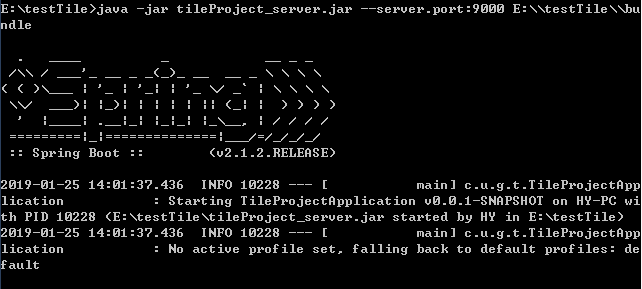
## 三、启动地图服务

7. 找到地图服务包(tileProject\_server.jar) 在存放这个jar包的目录运行命令行

java -jar tileProject\_server.jar --server.port=8068 E:\\testTile\\bundle

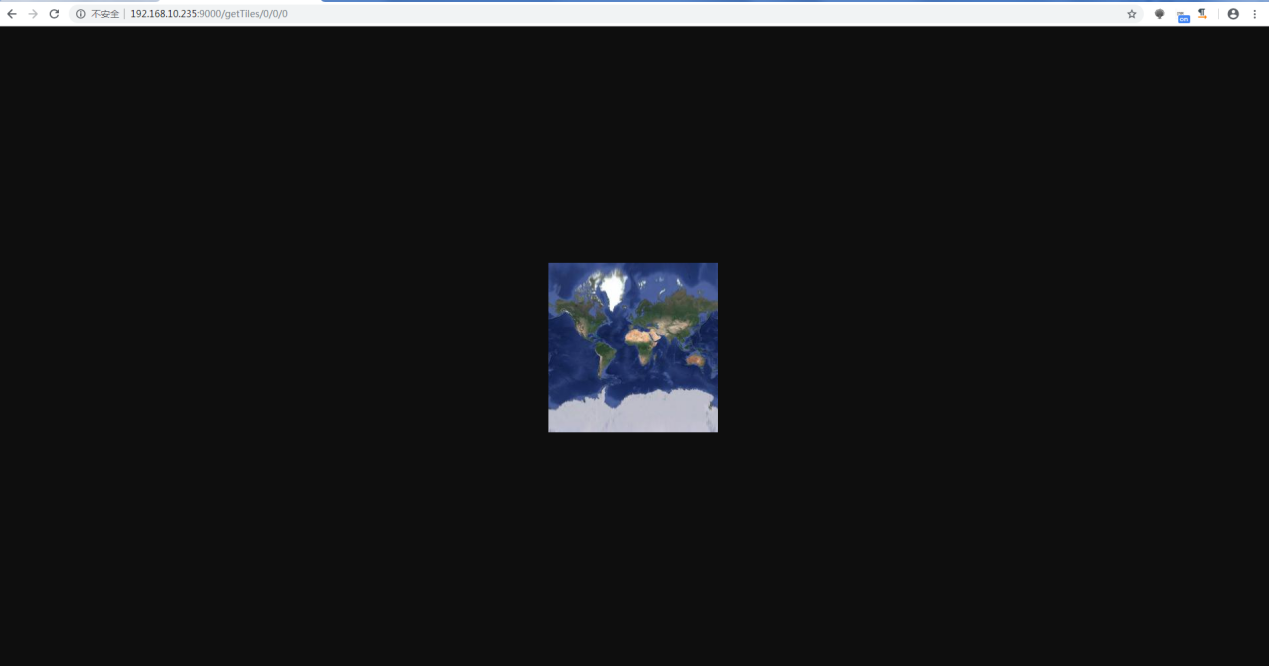
--server.port=8068代表端口号 默认是8066 此参数选填

E:\\testTile\\bundle 代表bundle瓦片存放路径 此参数必填



测试：

浏览器输入http://[ip]:[port]/getTiles/1/0/0如果返回一张图片则部署成功。



uearth中设置tileLayer的url处改为http://[ip]:[port]/getTiles/{z}/{y}/{x}

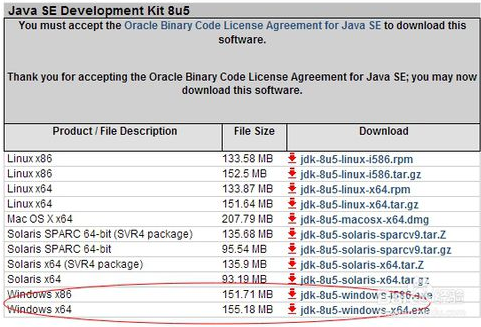
例如：

**var** tileLayer1 = app.create({  
 type:'TileLayer',  
 name: 'tileLayer1',  
 url: 'http://192.168.10.235:9000/getTiles/{z}/{y}/{x}'  
});

## 附录：jdk8图解安装

下面会从JDK的下载、安装、环境变量的配置和其中的一些问题进行详细说明，Let‘s’ go！

STEP1: 官网下载：



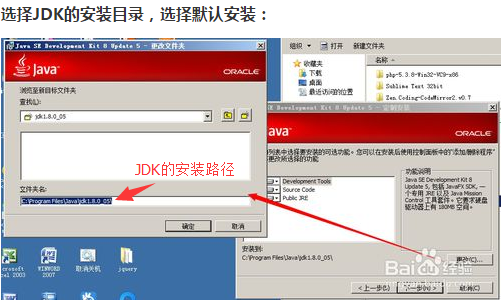
  【版本说明】：WIN7操作系统有32位和64位，分别要下载对应的JDK8版本

    32位：jdk-8u5-windows-i586.exe

    64位: jdk-8u5-windows-x64.exe

    选择安装包时，注意只能是Development kit版本才是开发人员用的。

STEP2:下载完成后，64位JDK的安装过程（重点）



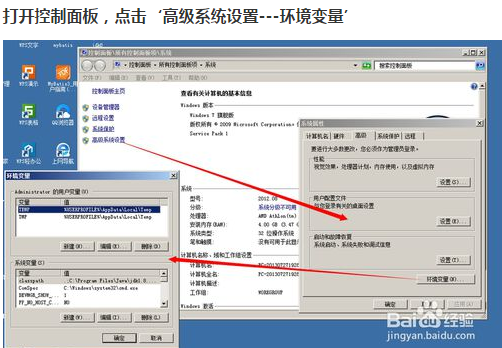
我们可以选择安装路径，也可以默认安装，看自己习惯了。如果更改路径最好不要放在中文或者带空格的目录下，以免将来出现不必要的麻烦。



 【attention】：已经安装JDK的，可不再安装JRE，不过安装了JRE也没影响。

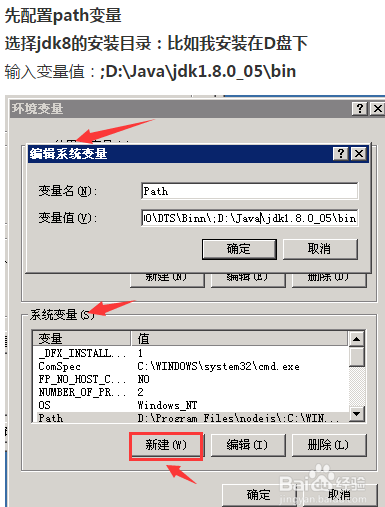
STEP3: 安装完成，配置环境变量

STEP3-1: 系统变量路径

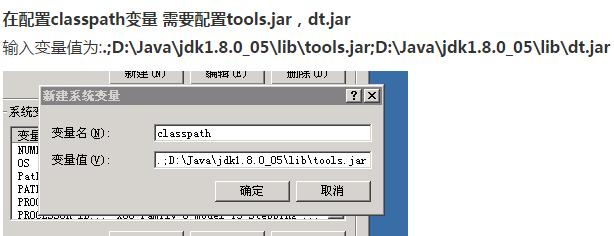


 JRE是Java的运行环境，包括了JVM等一些Java的运行环境，听起来是不是很重要！但是，在我们安装的JDK里面已经有了JRE了，所以，这里可以选择取消，对以后的开发工作一点都不影响，呵呵~当然，装了也没关系滴

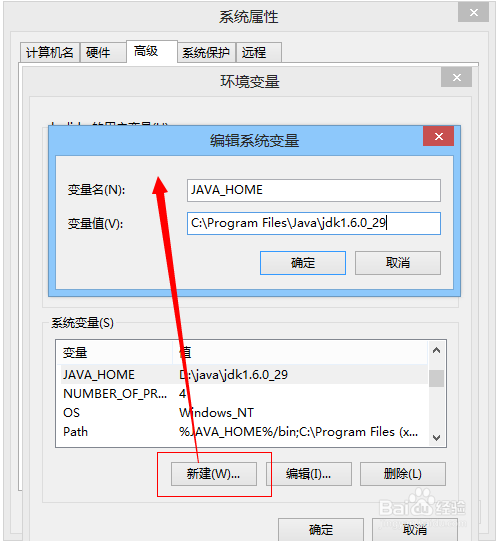
STEP3-2: 系统变量路径添加——path变量



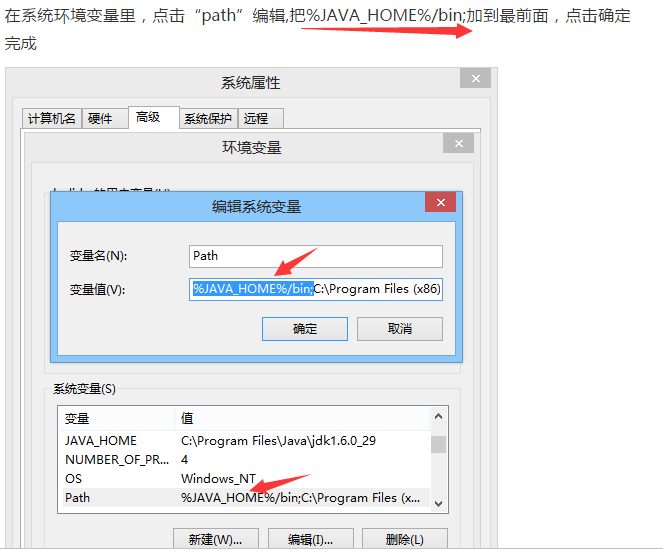
STEP3-3: 系统变量新建——classpath变量



【attention-1】：系统变量配置，也可以借助一个过渡变量，如先定义新变量JAVA\_HOME



定义新变量JAVA\_HOME后，其他变量需要指向JDK路径时，可以引用此变量，避免重复输入造成不必要的错误



classpath变量也可以采用同样的方法来引用



 【说明】：

%%这玩意的作用：动态的获取某一个已存在的环境变量的值。%JAVA\_HOME%就表示了我们的安装路径，以后如果我们的安装路径改变，只需要在JAVA\_HOME的值中改变相应路径就行了。class在Java中是一个关键字，表示一个类，classpath是用来指定Java中类或者包所在的路径，在以后的学习中，我们还需要JVM去寻找Java中的库，这也是需要用classpath来指定路径的，总的来说，这是方便我们的JAVA虚拟机去寻找相应的类文件，JVM会按照classpath环境变量指定的地址依次寻找，已最先找到的为准，所以一般我们将其值设为：.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar最前面有个点，代表的是当前路径，最后面最好不加分号，原因在以后的学习中会遇到的。

如果我们在用别人的电脑开发或者编译用时，随便去更改别人的path可能会招别人恐慌，为此我们还可以设这临时变量，临时变量是在dos命令行中进行设置的，设置之后，只在当前的命令行窗口中有效，设置方式：set path=你需要的路径，set命令的作用：可以用于查看或者设置环境变量的值。故也可以用set path 这样的命令来查看path中的地址值。自己编译完之后，把命令行窗口关闭就啥也不影响了。

 【attention-2】：什么是环境变量

   环境变量一般指在操作系统中用来指定操作系统运行环境的一些参数。其实这句话也不是很好理解，来举个例子吧。如path，当要求系统运行一个程序而没有告诉它程序所在的完整路径时，      系统除了在当前目录下面寻找此程序外，还会到path中指定的路径去找。通俗的讲，你要的东西我这里没有，怎么办呢，那就去path里说的地儿找找吧。

   在我们编译程序时，我们需要用到javac这个编译工具，而javac是在java的安装目录下的，当我们在其他目录调用javac时，系统因为在当前目录找不到就会报错，为了方便我们以后编译程          序，不要每次编译的时候就跑到java安装目录中去，所以我们来设置环境变量。

【attention-3】：环境变量的解析

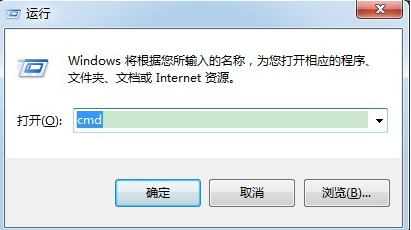
   JAVA\_HOME:  jdk的安装路径

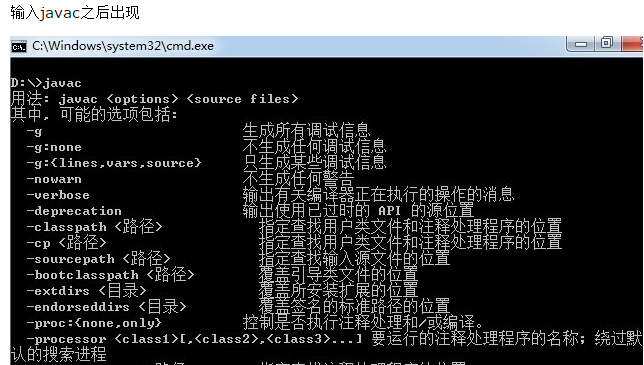
   classpath:  java加载类路径，只有类在classpath中java命令才能识别，在路径前加了个"."表示当前路径

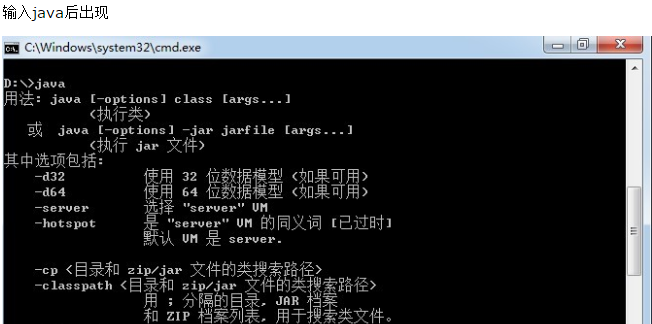
   path： 确保系统在任何路径下都可以识别java,javac命令

STEP4: 验证配置正确与否

**win+R——>输入cmd——>输javac和java，如图：**

****





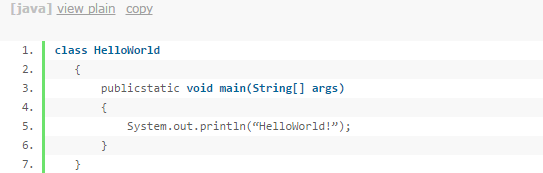
    那就表示设置成功了，接下来可以开始我们的Java学习之旅了！！！！！

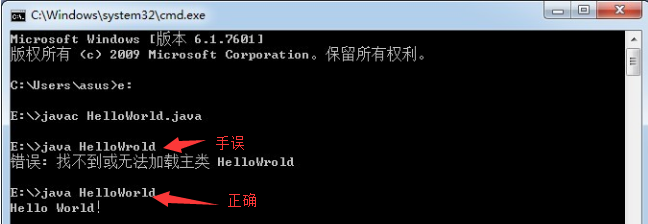
 【补充】

    1、 path环境变量是先在当前目录找执行程序，如果没有，再到path指定目录中去寻找。而classpath是先在classpath环境变量中去找执行程序，找到了，即使当前目录中有同样的执行程序也执行不到；且只要在classpath的值后面加了分号才会当当前目录中来寻找执行程序。

    2、 在设置临时变量时，如果想保留原有的path值，只需在新设的值后面加上%path%这样的语句就行了。如：set path=新路径;%path%;

    3、 在使用javac和java进行编译和执行程序时，如果出现找不到文件的情况，在环境变量设置正确的前提下，只有两种可能：一是目录错误；二是文件名错误，就如我最后所执行的入门程序一样。如：入门小程序





注：本文摘自网络(<https://www.cnblogs.com/cynleely/p/7833518.html>) 感谢原作者。