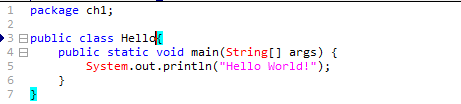
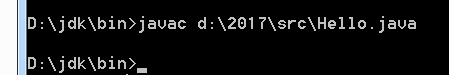
**JDK的安装、环境变量的配置（2017-09-09整理）**

1. **一个例子讲解Java配置环境变量的原理和过程**
2. 三个假设：假设JDK安装在d:\jdk；假设你写的源代码保存在d:\2017\src；假设编译后的class文件保存在d:\2017\class；
3. javac命令：javac命令用来编译java源代码，生成相应的字节码.class文件；
4. DOS进入d:\jdk\bin目录；键入“javac”，系统会提示javac命令包含的选项参数；
5. 在记事本（或者editplus，或者其他编辑器）中写一个Hello.java。

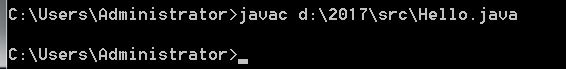


1. 在DOS中进入d:\jdk\bin目录；键入如下：

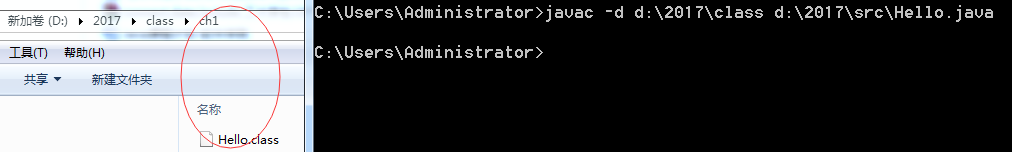


DOS界面没有任何提示，表明编译成功，会在当前目录（DOS提示符目录）生成相应的Hello.class。

1. Java编程中一般要配置PATH环境变量（PATH是一种环境变量名，它的值一般设置到JDK安装目录下的bin目录，本例中 = d:\jdk\bin），方便开发者在任何路径下直接运行javac、java命令，不必再麻烦切换到d:\jdk\bin目录；
2. 假设我们已正确设置环境变量PATH，则可以在任意路径下运行上述命令：



1. javac命令有一个重要的参数，“-d”参数，举例来说：



“-d”参数有两个作用：一是可以自动将编译生成的字节码.class文件保存到指定目录；二是可以根据源文件中的package包定义，自动生成相对应的目录，并将.class文件保存到它的包对应的目录中（如上图，因为Hello.java中有一条package ch1;语句，编译之后在d:\2017\class目录下自动生成ch1目录，并最终将Hello.class编译到ch1目录）

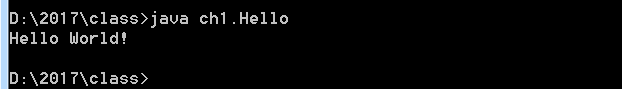
1. javac中的另外一个参数，“-classpath”（也可以用“-cp”简化代替），其作用主要是：在编译当前类时，指定其所依赖的类的加载路径，具体我们后面举例详解；
2. 目前为止，我们完成了一个源文件的编译，并着重讲解了javac的“-d”参数。接下来，我们看看Java程序的解释器java命令。
3. Java命令又叫Java解释器，用来解释执行正确编译之后生成的.class文件（前提是这个类有主函数）。Java命令的执行格式：“java 类名”，其中 类名=包名.类名，本例中ch1.Hello（记住：解释执行时，不用加后缀.class，编译的时候javac后面需要带.java后缀名）



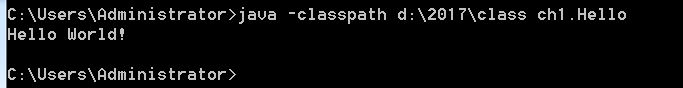
上面的错误，有的同学会说未明确指定ch1.Hello所在的目录，有的同学会尝试如下格式：



上面的错误说明，在解释一个类的时候不能包含其绝对路径，正确的解决途径：A)切换到这个类所在的目录（d:\2017\class）：



B)使用java命令的“-classpath”参数（同javac一样，可以用-cp简化代替）



1. 很多教材中让我们配置CLASSPATH环境变量，这样我们就不必再每次用java解释器时手工敲-classpath参数了，省事。（此处略去环境变量CLASSPATH的配置）。如果正确配置了CLASSPATH环境变量，一般地，我们可以在任意路径下运行我们的代码；
2. CLASSPATH说到底是一个或者多个目录，如果是多个目录，在Windows下使用分号分割。环境变量CLASSPATH的作用：Java虚拟机在装载一个类时，在哪些目录（CLASSPATH）下自动搜索这个类。（回忆一下，javac命令也有-classpath参数）