maven是个项目管理工具，如果我们不告诉它我们的代码要使用什么样的jdk版本编译的话，它就会用maven-compiler-plugin默认的jdk版本来进行处理，这样就容易出现版本不匹配，以至于可能导致编译不通过的问题。

maven的默认编译使用的jdk版本貌似很低，使用maven-compiler-plugin插件可以指定项目源码的jdk版本，编译后的jdk版本，以及编码。

<plugin>

<!-- 指定maven编译的jdk版本,如果不指定,maven3默认用jdk 1.5 maven2默认用jdk1.3 -->

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.1</version>

<configuration>

<!-- 一般而言，target与source是保持一致的，但是，有时候为了让程序能在其他版本的jdk中运行(对于低版本目标jdk，源代码中不能使用低版本jdk中不支持的语法)，会存在target不同于source的情况 -->

<source>1.8</source> <!-- 源代码使用的JDK版本 -->

<target>1.8</target> <!-- 需要生成的目标class文件的编译版本 -->

<encoding>UTF-8</encoding><!-- 字符集编码 -->

<skipTests>true</skipTests><!-- 跳过测试 -->

<verbose>true</verbose>

<showWarnings>true</showWarnings>

<fork>true</fork><!-- 要使compilerVersion标签生效，还需要将fork设为true，用于明确表示编译版本配置的可用 -->

<executable><!-- path-to-javac --></executable><!-- 使用指定的javac命令，例如：<executable>${JAVA\_1\_4\_HOME}/bin/javac</executable> -->

<compilerVersion>1.3</compilerVersion><!-- 指定插件将使用的编译器的版本 -->

<meminitial>128m</meminitial><!-- 编译器使用的初始内存 -->

<maxmem>512m</maxmem><!-- 编译器使用的最大内存 -->

<compilerArgument>-verbose -bootclasspath ${java.home}\lib\rt.jar</compilerArgument><!-- 这个选项用来传递编译器自身不包含但是却支持的参数选项 -->

</configuration>

</plugin>