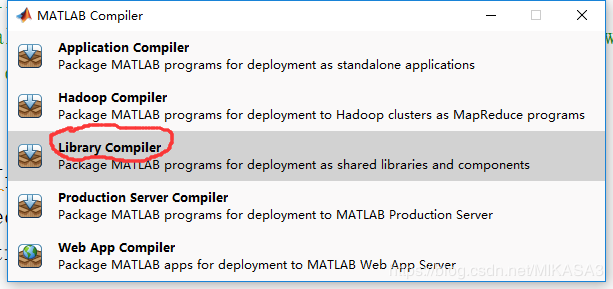
整体流程：



一、Matlab生成jar

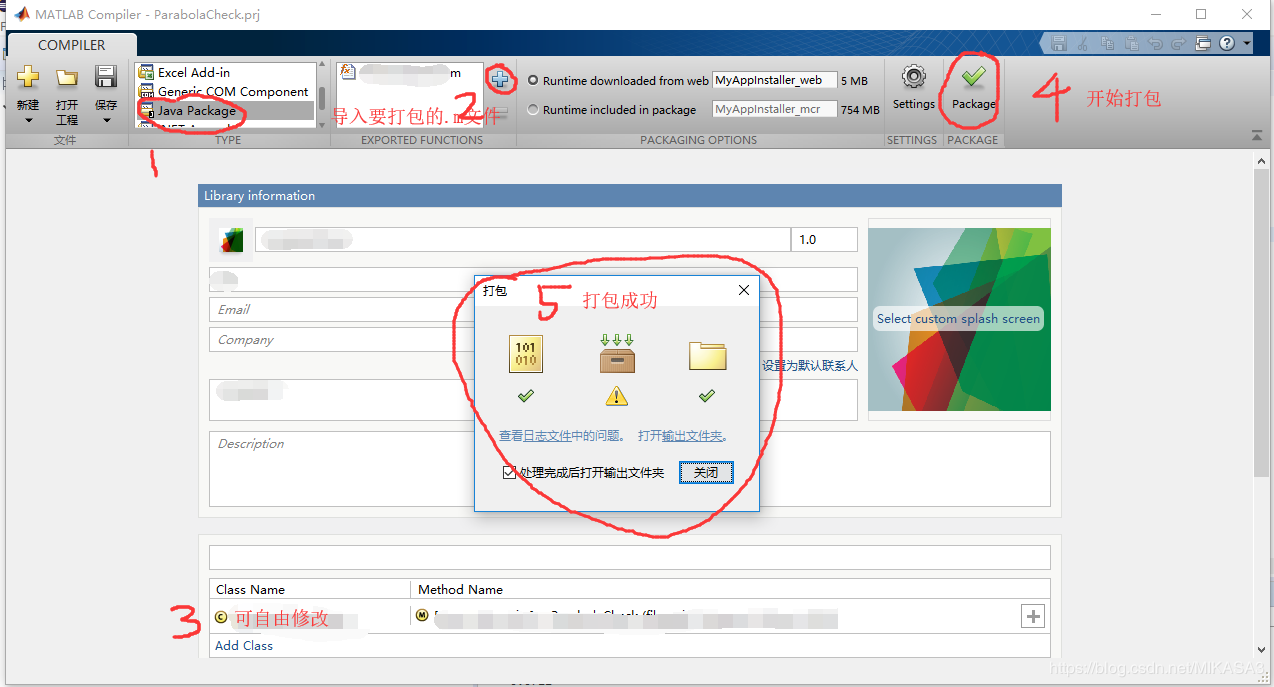
在命令行直接输入：

deploytool



【注意】如果1处没有Java Package选项

需要在matlab的setup.exe中，补充安装选择MATLAB Compiler和MATLAB Compiler SDK



二、检查/配置Java的环境

1、在cmd中java和javac都是有效命令

如果javac无效：

①检查系统变量JAVA\_HOME是否为安装JDK的路径；

②（没有就自己新建）检查系统变量CLASSPATH，注意前面有个点“.”：

.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;

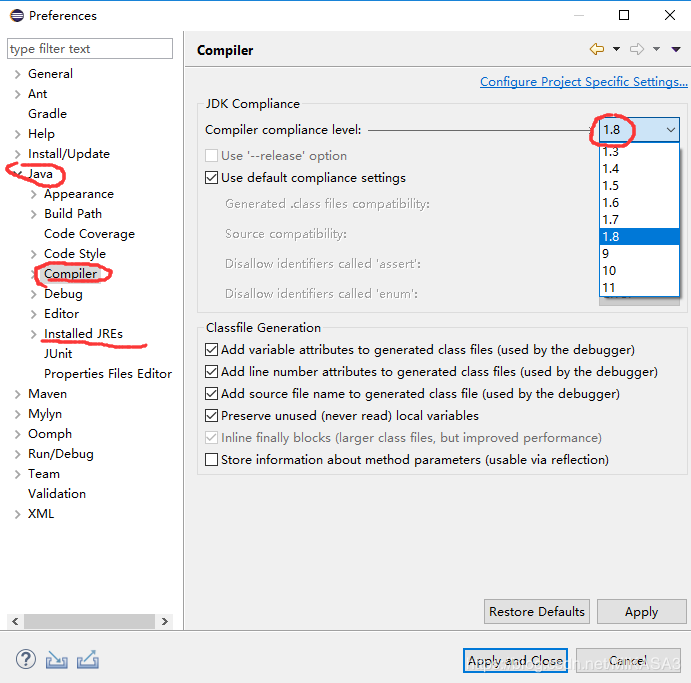
③系统变量PATH:

;%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;

三、配置eclipse2019.03

【前提】项目中使用的编译器与本机Java（java -version查看）的版本号一致（例如均为1.8）：

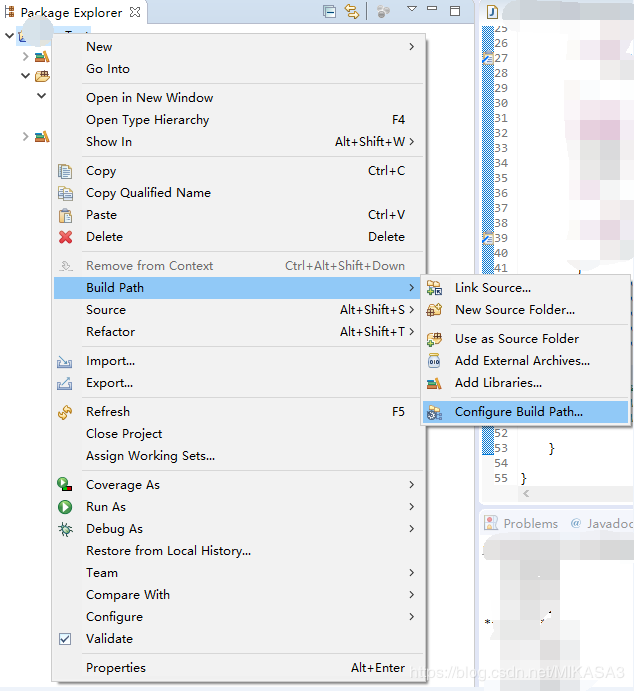
window->preference：



否则报错：

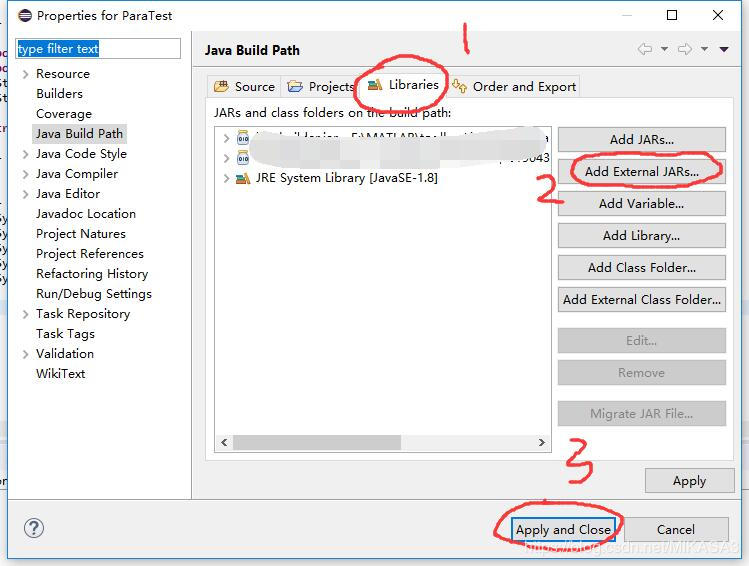
Unsupported major.minor version 52.0

1、新建项目，导入两个jar包：



①MATLABr2018a安装路径\toolbox\javabuilder\jar\javabuilder.jar

②自己刚刚用matlab生成的jar包

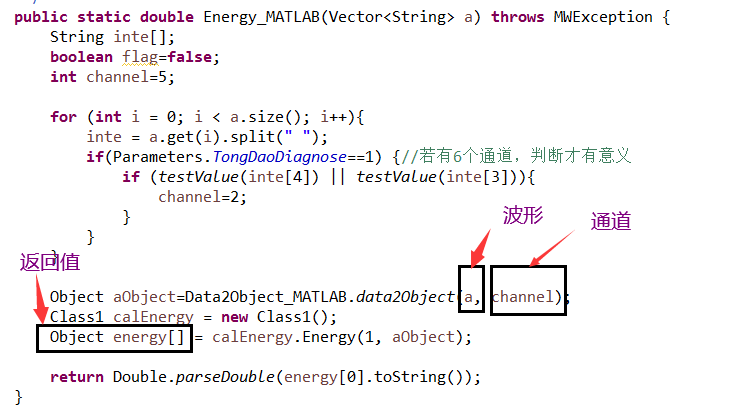


如果上述方法配置完不能运行MATLAB程序，则需要单独安装：

在D:\matlab\matlabs\toolbox\compiler\deploy\win64目录下有个MCRInstaller.exe，点击安装，安装它会自动配置MCR的环境变量，随便提一下网上有种说法就是这个MCR路径必须写在matlab主程序路径后面

调用代码如下：

其中返回值保存为object类型。



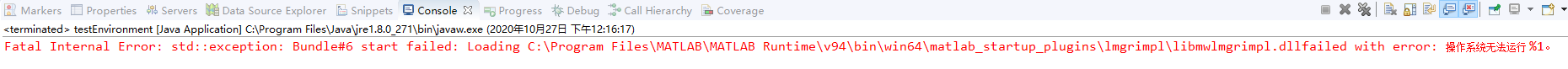
融入后的能量计算，可以看出能量基本符合实际情况。



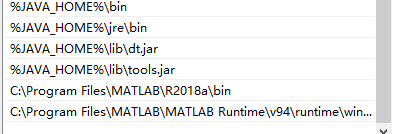
左图：2019.12.21事件 右图：2020.2.14事件

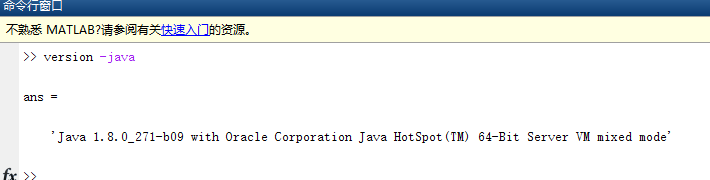
20201027问题汇总

笔者参与的项目由于需要借助MATLA代码来提高Java的编码效率，但在实际使用中发现Java MATLAB混编的问题还挺多的，尤其在不同电脑上进行开发的时候，由于经常切换工作环境，电脑经常更换，又涉及到不同的操作系统，所以配置环境变得极为繁琐和麻烦。最近在一台win10电脑上有遇到了如下问题：



刚开始以为是环境变量配置的问题，查看一下发现不是，环境变量配置没问题，cmd输入java与javac均正常，进入MATLAB输入version -java。





最后在https://blog.csdn.net/HongAndYi/article/details/79427078发现是：

问题原因

出现此问题是由于Windows 的system32目录下两个DLL文件版本太低，这两个文件是SSLEAY32.DLL和LIBEAY32.DLL。

这两个文件在MATLAB安装目录的\bin\win64目录下也有，\bin\win64这个目录是在添加到了系统的PATH变量里了的。但是Matlab编译后的exe程序在运行时，system32目录有高的优先级，会使用system32目录下的这两个文件，导致初始化错误，直接闪退。

解决方法

知道了此原因，有两种解决方法：

（1）将Matlab的安装目录 \bin\win64下的SSLEAY32.DLL和LIBEAY32.DLL复制到windows的system32目录下，覆盖旧版本的文件。但是这样可能导致使用这两个DLL文件的其他程序出现问题。

（2）将Matlab的安装目录 \bin\win64下的SSLEAY32.DLL和LIBEAY32.DLL文件复制到自己编译后的exe同目录下，这样exe文件在运行时就是找自己目录下的这两个DLL文件，而不会使用windows\system32目录下的DLL文件。