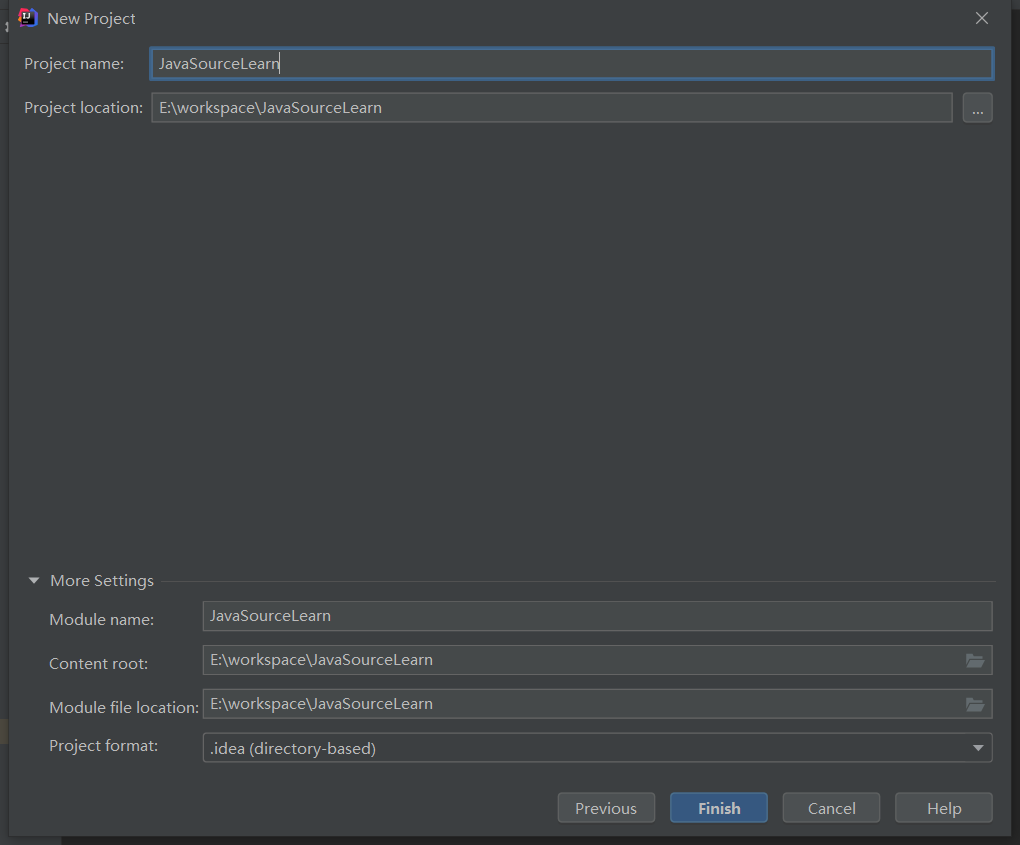
# 本次针对jdk8u版本的搭建

# 新建项目

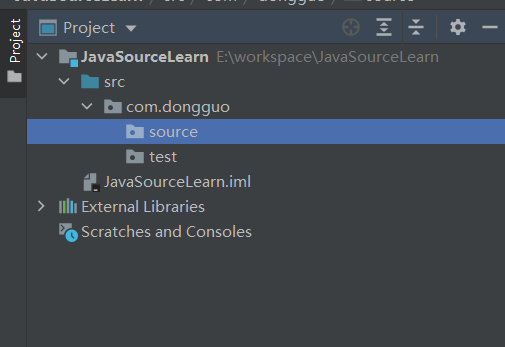
1. 新建java项目

JavaSourceLearn



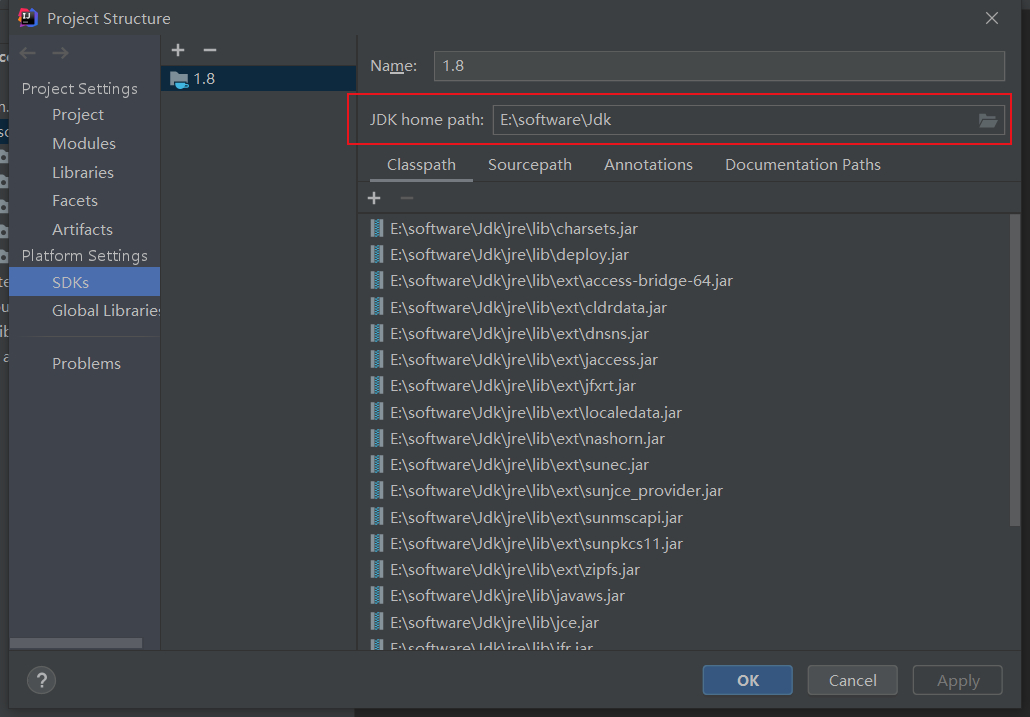
2 新建source包存放源码

3新建test包存放测试案例

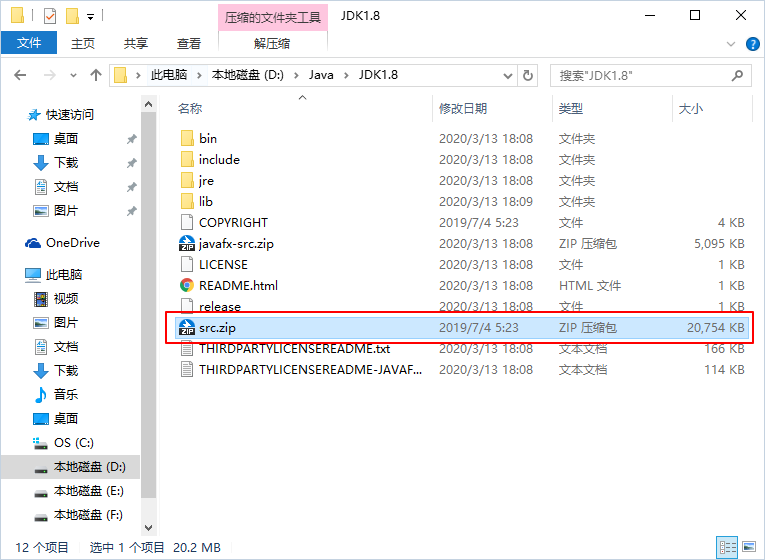


# 获取JDK源码

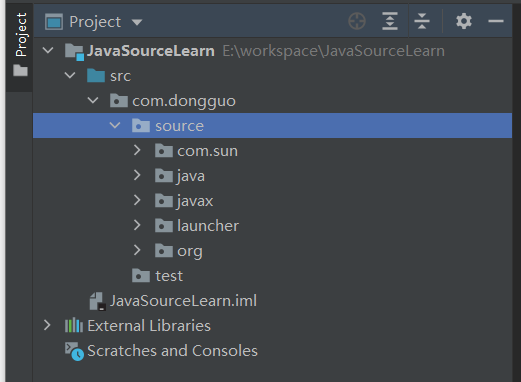
1. 打开Project Structure



1. 找到本地JDK安装位置将src.zip解压到项目source包中



解压后

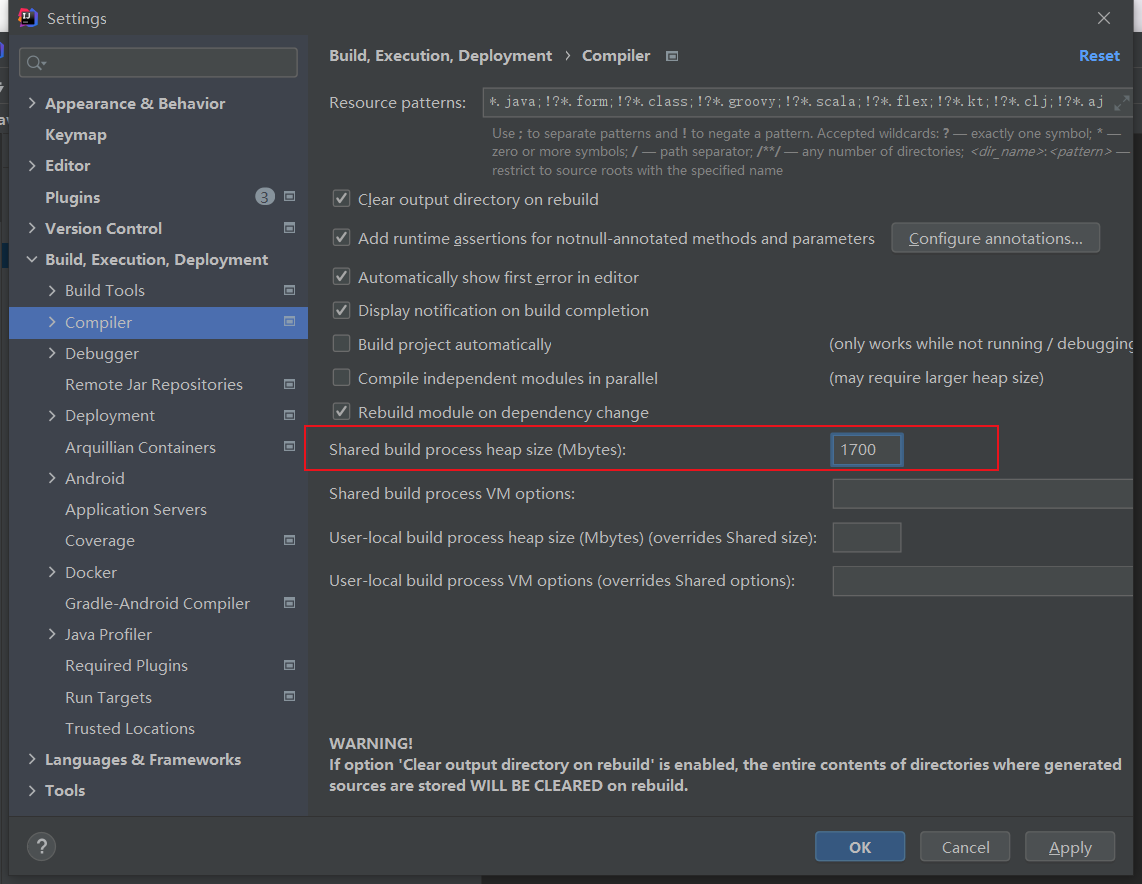


**提示**: 添加源码到项目之后首次运行需要较长时间进行编译,建议先设置好**进程堆**,防止编译失败

**构建进程堆大小**

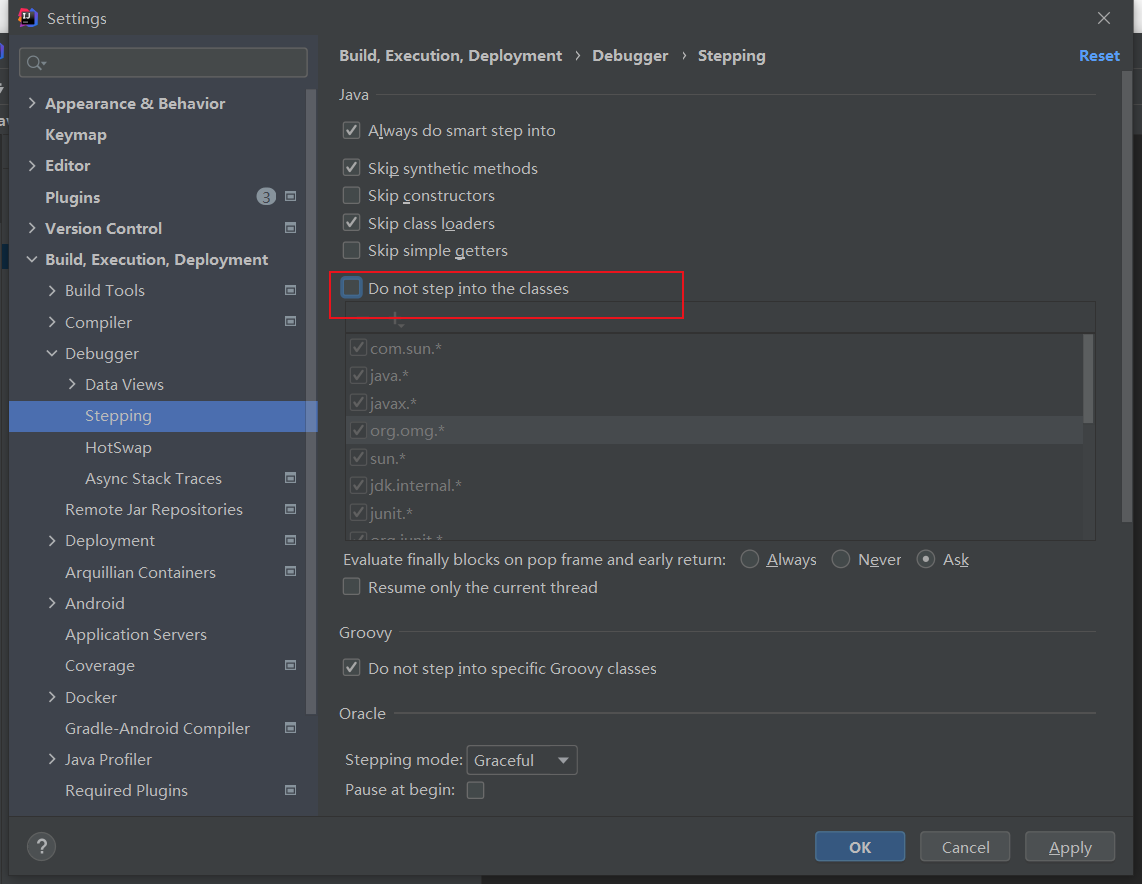
打开File -> Settings -> Build, Execution, Deployment -> Compiler

设置Build process heap size (Mbytes)



**4. 调试设置**

1. 打开File -> Settings -> Build, Execution, Deployment -> Debugger -> Stepping
2. 取消Do not step into the classes



**说明**: 该设置勾选后可以在Debug时不进入到相关类路径下的方法内

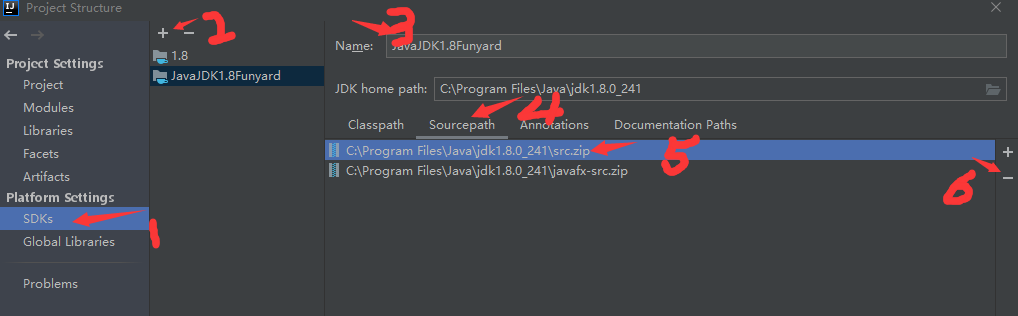
**5. 设置源码路径**

第三步：替换JDK关联

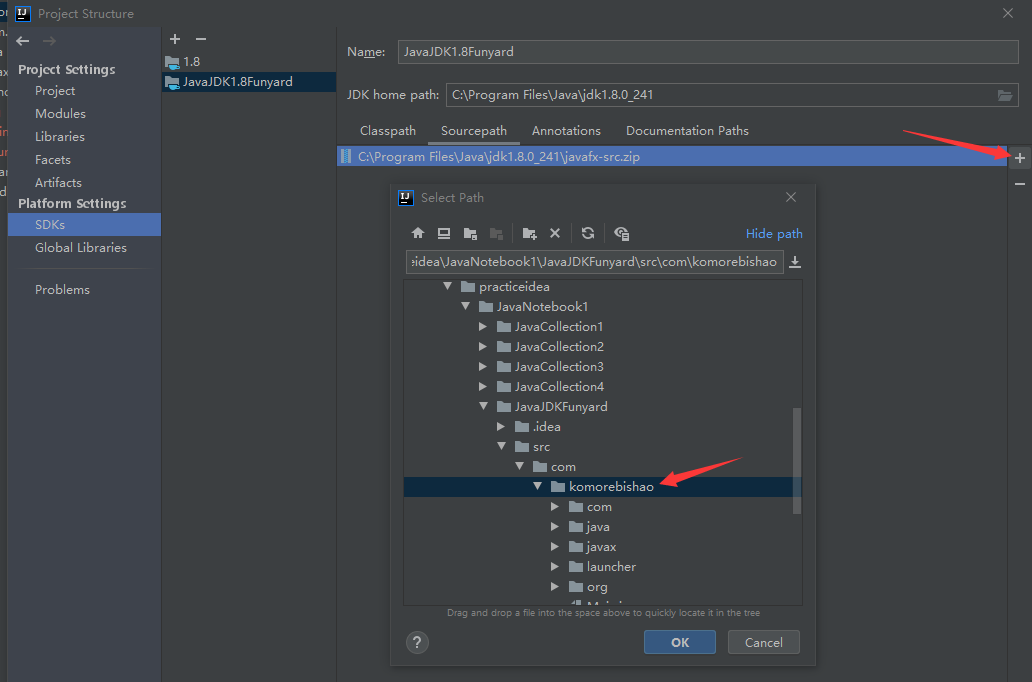
为了能够备注以及避免修改到其他项目所使用的内容，添加一个名为“JavaSourceLearn1.8”的SDK

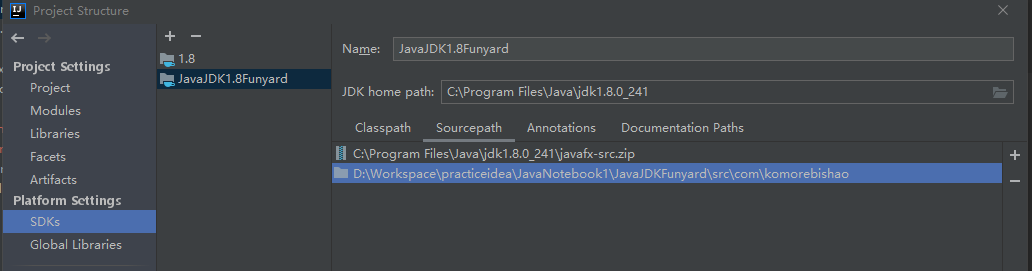
并将Sourcepath修改为当前项目src下的源码内容

**移除Sourcepath中关联的src.zip**

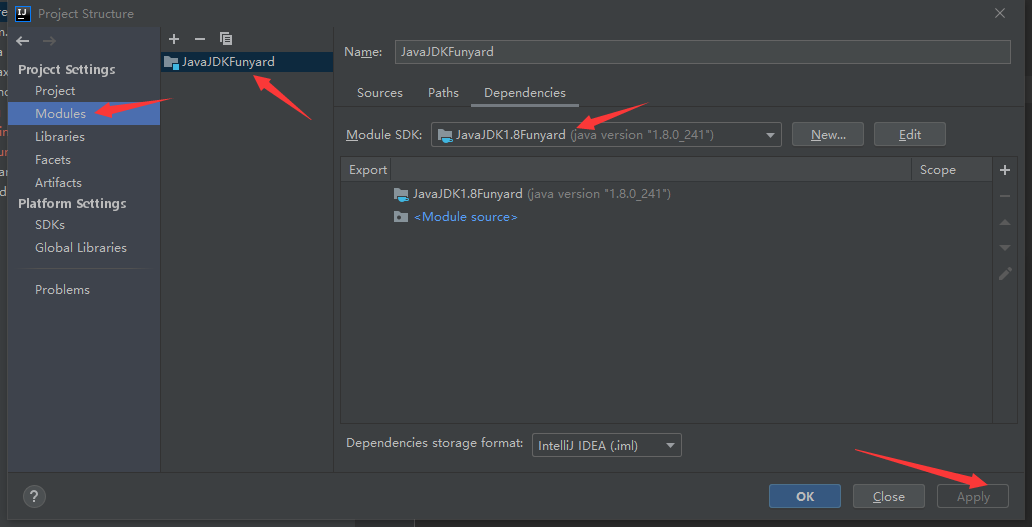


将项目中的JDK源码关联到Sourcepath





修改项目使用的JDK为上面新建的



**说明**: 原先JDK源码包为锁定状态,Debug进入源码后无法修改/添加注释

　这样就可以添加注释了

## 

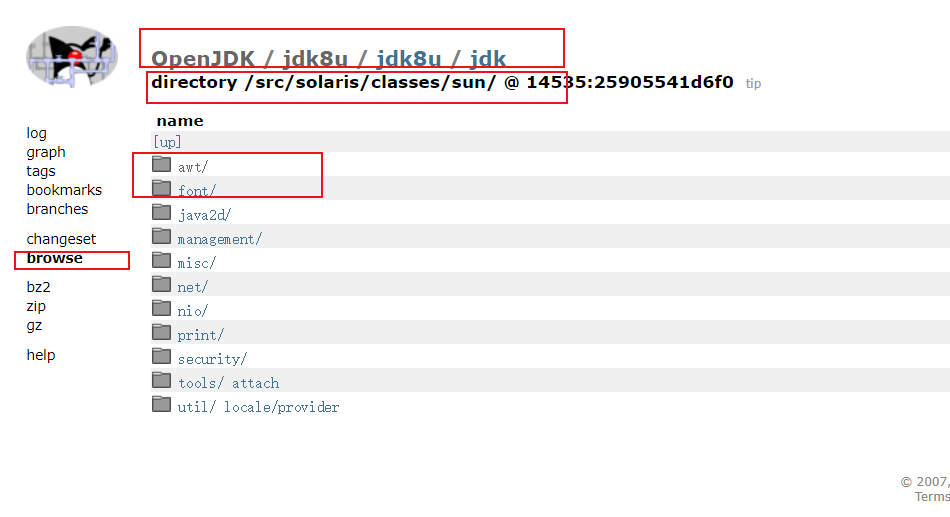
## 3. 解决报错问题

 找不到UNIXToolkit和FontConfigManager

这是因为在Windows平台下缺少了这2个java类文件所导致的。

这2个类文件可在OpenJDK网站上找到，地址：<http://openjdk.java.net/>，打开后点Mercurial，点jdk8u，点jdk，点browse，点src，点solaris，点classes，点sun，可以找到2个文件夹，awt和font，缺少的2个文件分别在这2个文件夹下





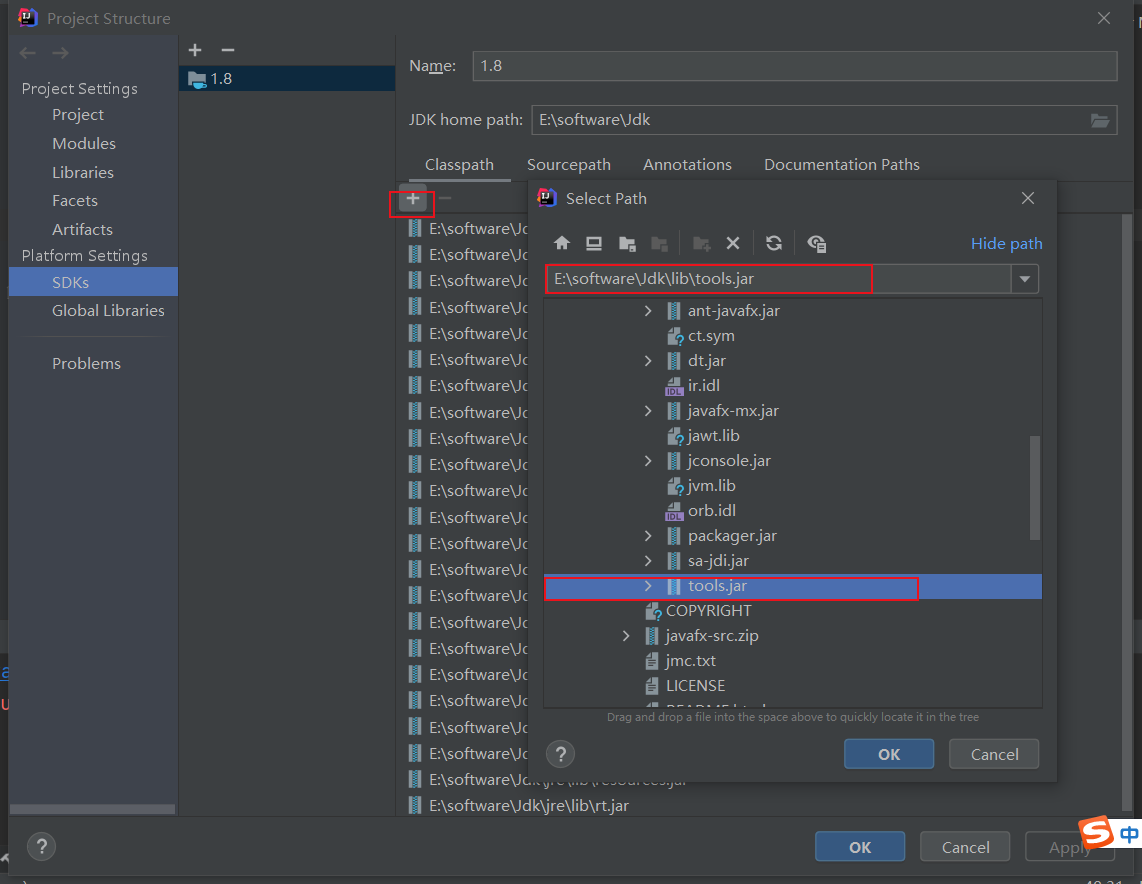
按照下图所示目录结构，新建同样名字的目录和空内容的文件

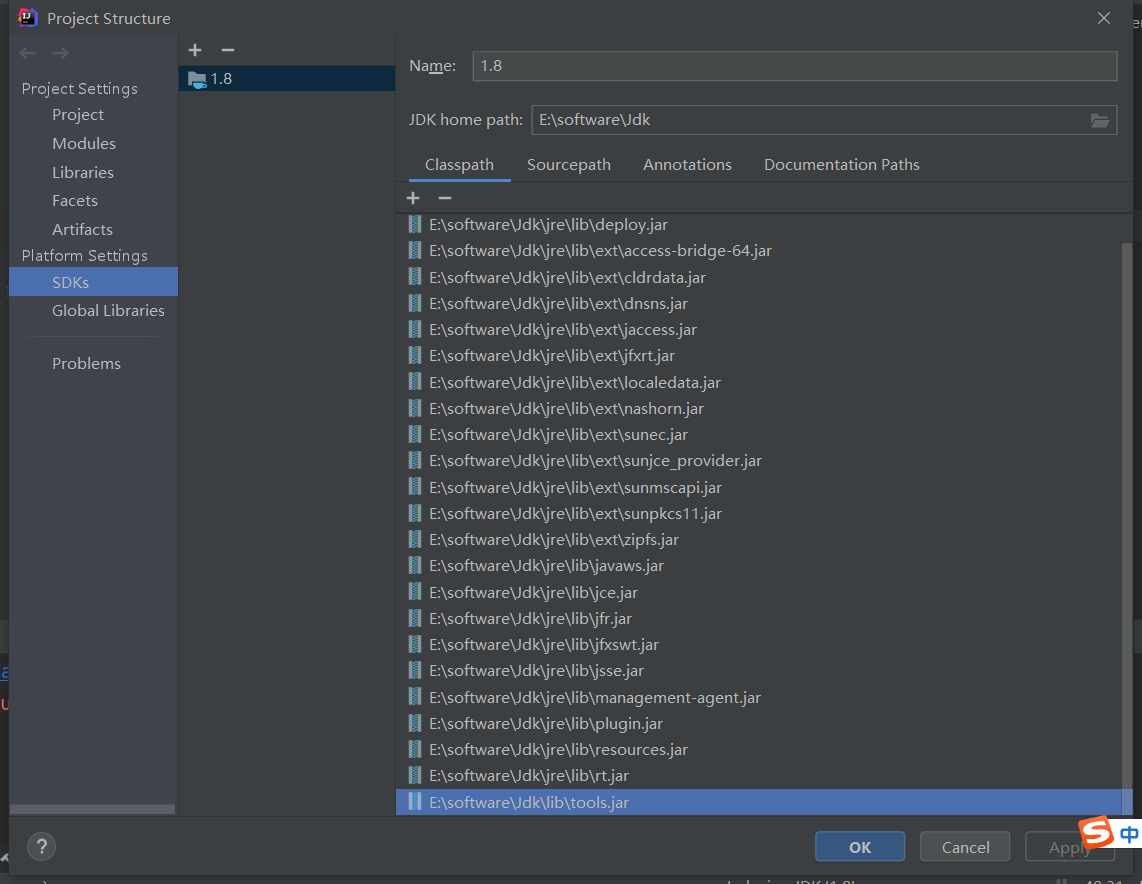
打开openJDK上同样名字的java文件，把对应的内容复制到IDEA

3.3 程序包com.sun.tools.javac.\*不存在

这是缺少tools的jar包所导致的。

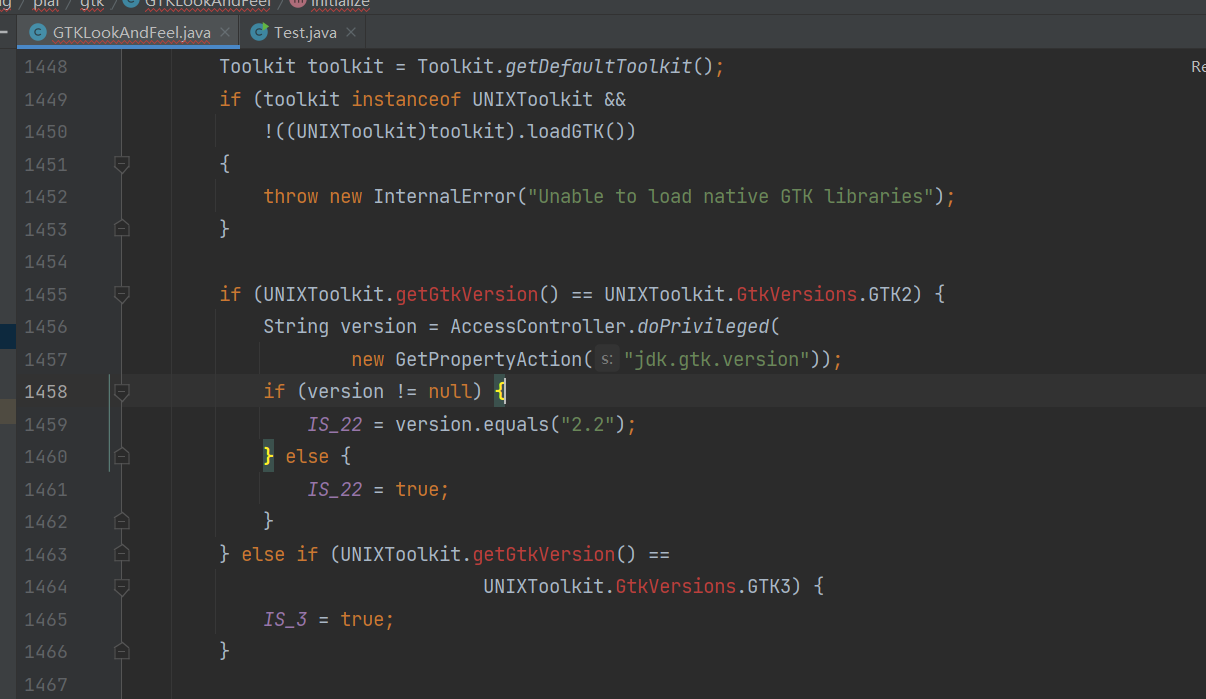
点击**File–>Project Structure–>Libraries**，手动把jdk8的tools.jar包添加，





这些都解决了，重新build发现还是有错误

GTKLookAndFeel类依然报红



百度了一下也没有找到原因，觉得这是和Unit平台相关的判断，应该问题不大，注释掉启动果然OK

## 换行注释 debug 错行的问题

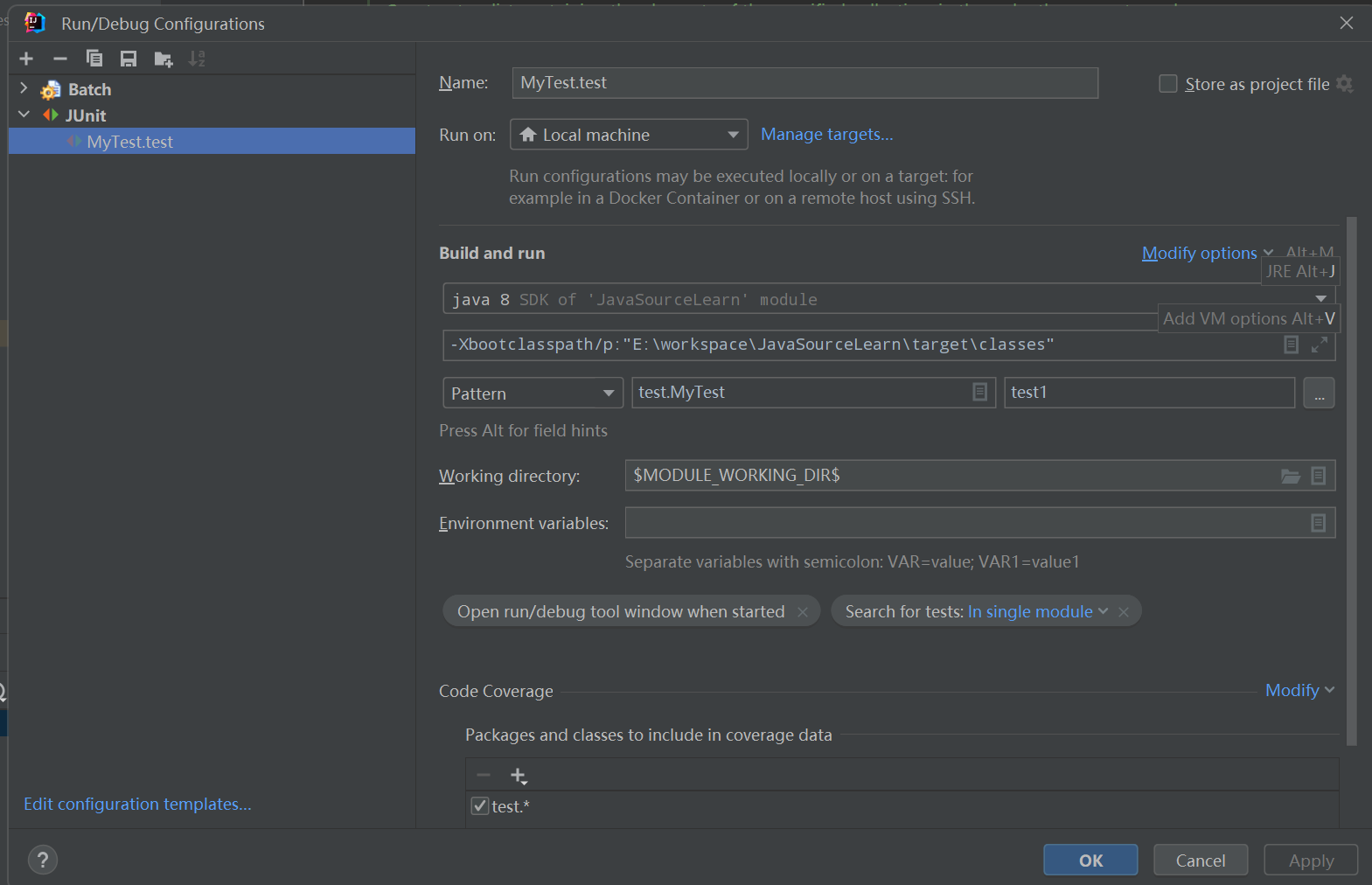
之后在这里的源码中换行注释，打断点 debug 会出现错行的问题，并且无法显示局部变量，提示 source code does not match bytecode。

让BootStrapClassLoader优先加载我们的类

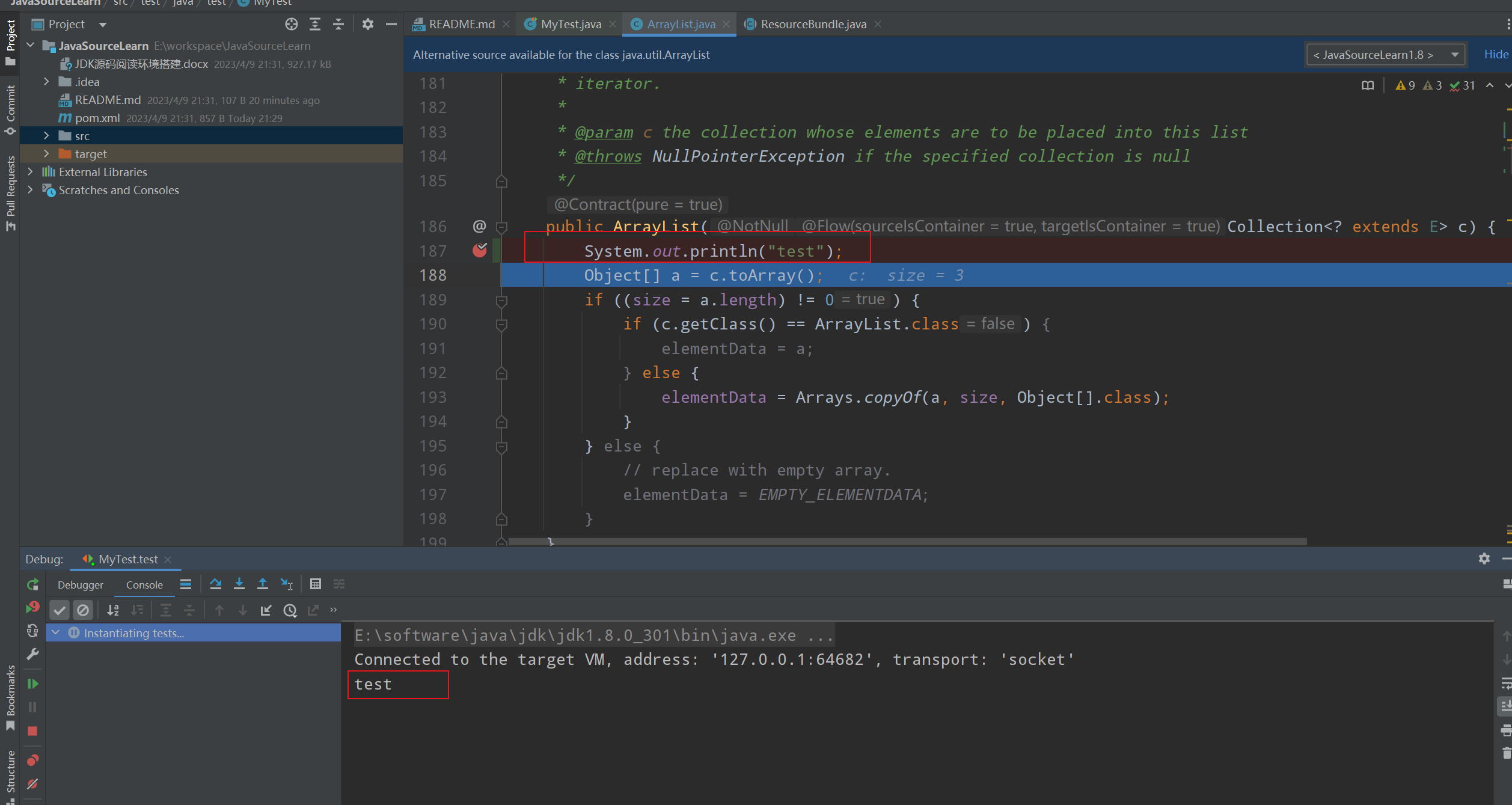
-Xbootclasspath/p:"E:\workspace\JavaSourceLearn\target\classes"

## 新建测试

class MyTest {  
 @Test  
 public void test1() {  
 String array[] = {"hello", "world", "java"};  
 List<String> list = new ArrayList<String>(Arrays.*asList*(array));  
 list.add("zhangsan");  
 System.*out*.println(list);  
 list.remove("zhangsan");  
 System.*out*.println(list);  
 }

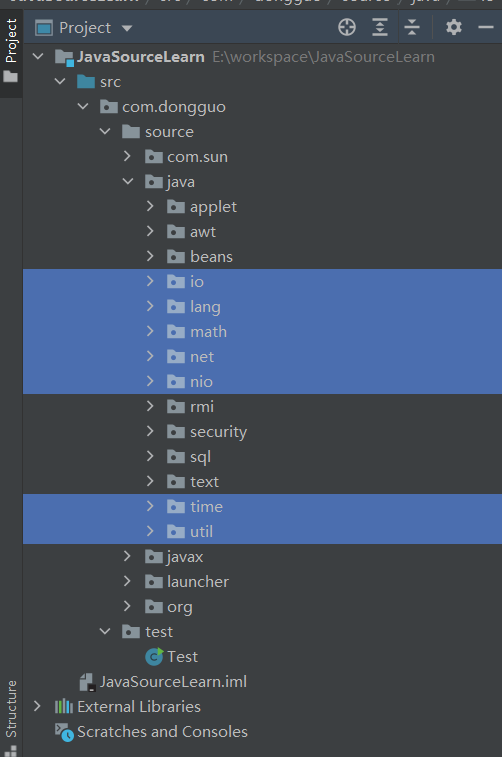


启动调试，F7调试进入源码内部



此时进入到我们自己搭建的源码阅读环境中。并且可以在源码中添加自己的注释。

## 源码阅读重点部分



## 阅读顺序

JDK 中的代码非常多，不可能、也没必要全部读完，因此要有的放矢。从整体上来讲，顺序大概是：

集合框架类

主要包括 Collection、Map、Queue 等组成的一系列常用类和接口，包括 ArrayList、LinkedList、HashMap 等。

这部分内容日常开发使用较多，而且面试高频出现，因此可以先从这里入手。

并发包

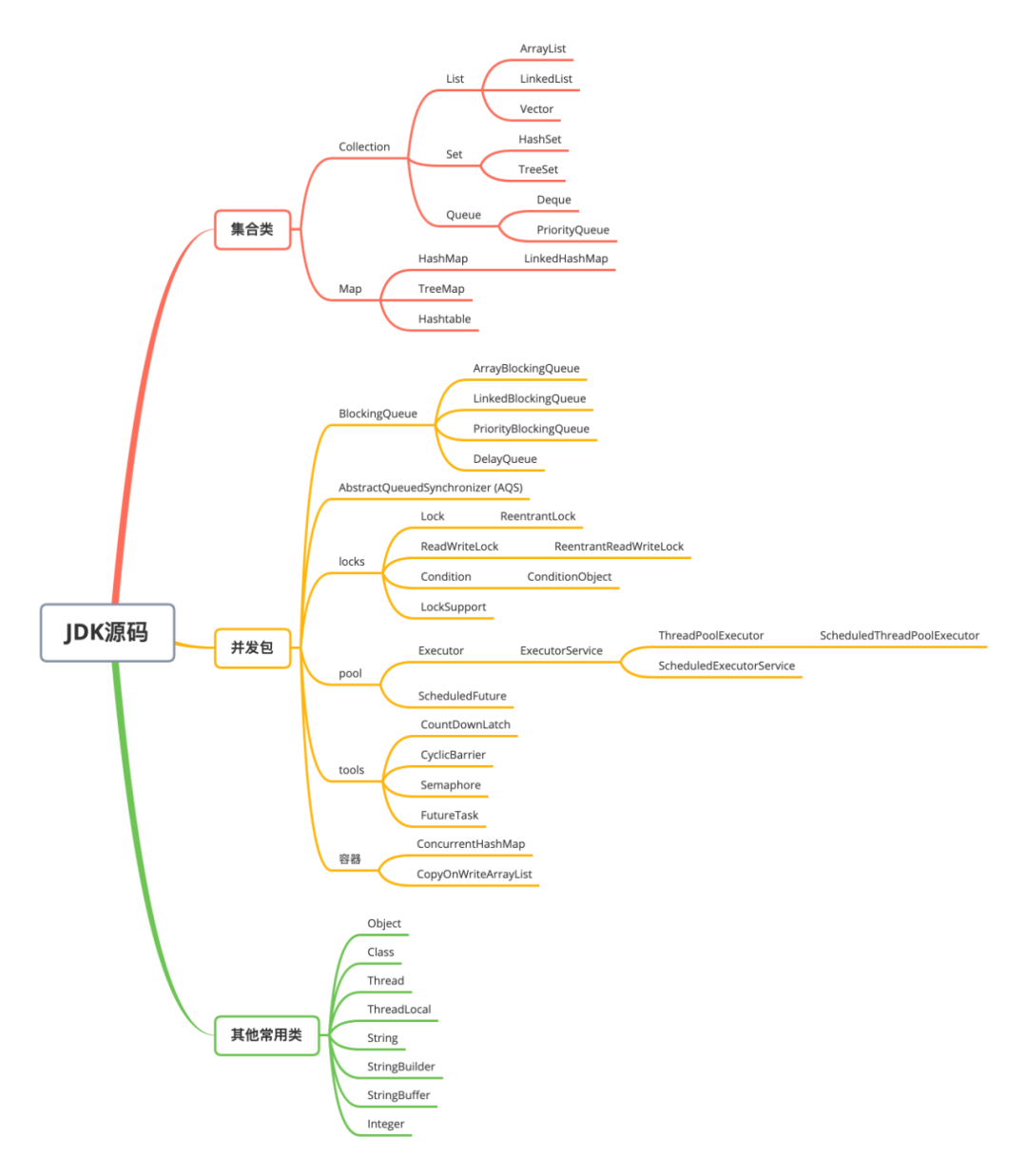
即 java.util.concurrent (J.U.C) 包下的常用类，包括 ReentrantLock、ThreadPoolExecutor、AQS 等。

该部分提供了并发编程的常用工具类，也是面试高频。

其他常用类

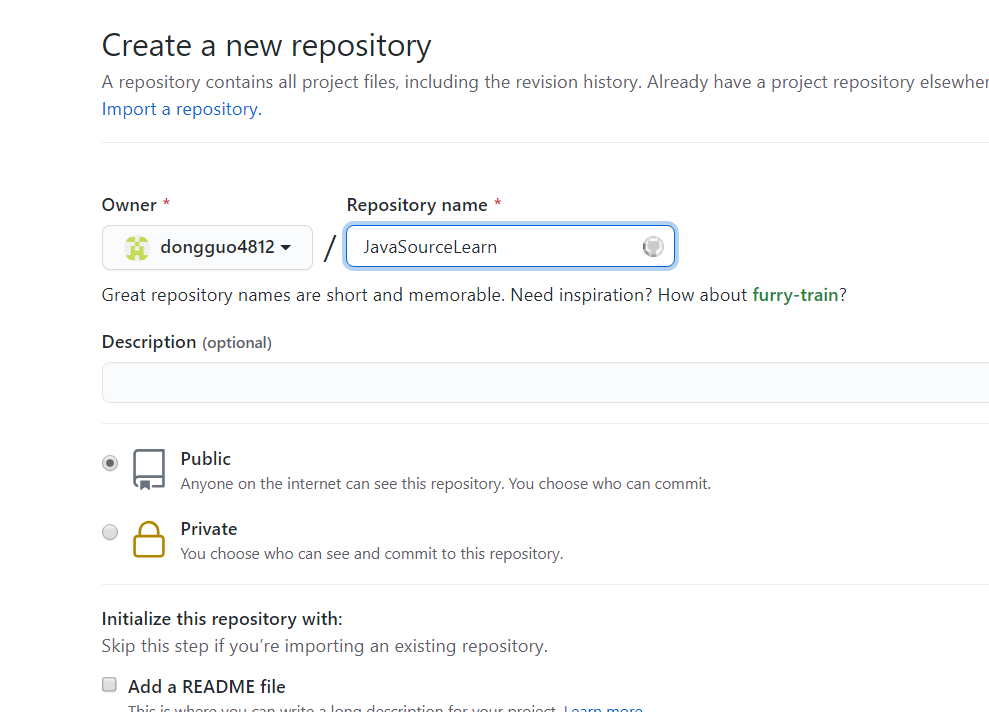
例如 ThreadLocal、String、StringBuilder、StringBuffer 等。

整体概览如下：

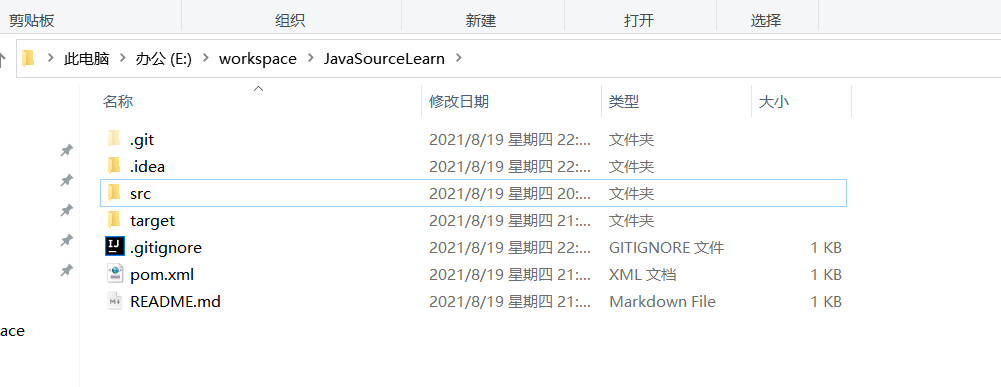


提交到github

git创建新的项目



将项目git clone 到本地



设置好ignore忽略没必要的文件进行提交

git add

由于文件过多 add需要一段时间

git commit “JavaSourceLearn源码初始化”

git push