**深 圳 大 学 实 验 报 告**

**课程名称： 互联网编程**

**实验项目名称： 实验1熟悉互联网编程开发环境**

**学院： 计算机与软件学院**

**专业： 软件工程**

**指导教师： 毛斐巧**

**报告人：杨皓翔 学号：2023150139 班级： 软工02**

**实验时间： 2025.2.25、3.4、11**

**实验报告提交时间： 2025年3月30日**

**教务处制**

一、实验目的与内容：

**目的：**掌握面向java的互联网编程开发环境的搭建。学习、掌握java网络程序编写基本步骤，例如，基于Eclipse或MyEclipse或其它IDE开发环境编写简单的一个网络通信程序，掌握编译、运行等基本步骤和操作。

**内容要求：**

1. 搭建开发环境（基本要求，60分）：

（在自己电脑上）下载安装和配置所选择使用的开发环境平台（Eclipse或MyEclipse或其它IDE），并熟悉该开发环境中的常用操作。

请在报告中给出搭建环境成功的截图和简要文字说明，简述自己所了解掌握的常用操作。

1. 尝试编写自己第一个例子网络通信程序（扩展提升要求，30分）：

可试着根据java编程课所学到的java网络编程技术，尝试编写一个有客户端和服务器端，并且两端能进行通信交互的程序。也可根据自已所掌握的其他编程技术，编写一个网络通信程序，只要在程序中应用到了Internet通信即可。

注意：

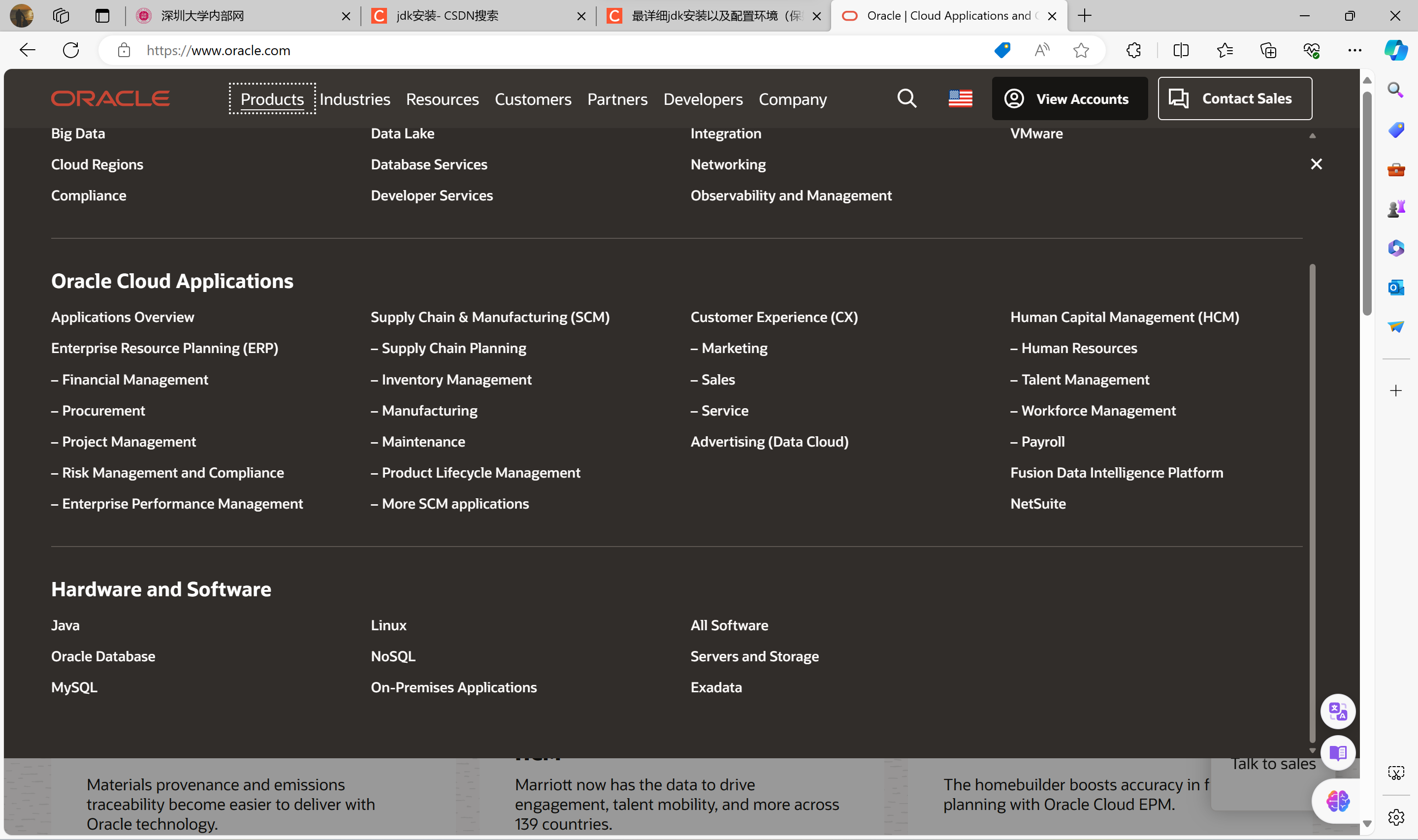
1. 实验报告中需要有实验结果的截屏图像。

二、实验过程和代码与结果

**1.开发环境的搭建 实验过程及结果**

1.1 进入[oracle](https://so.csdn.net/so/search?q=oracle&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/hgnuxc_1993/article/details/_blank)官网，下载jdk

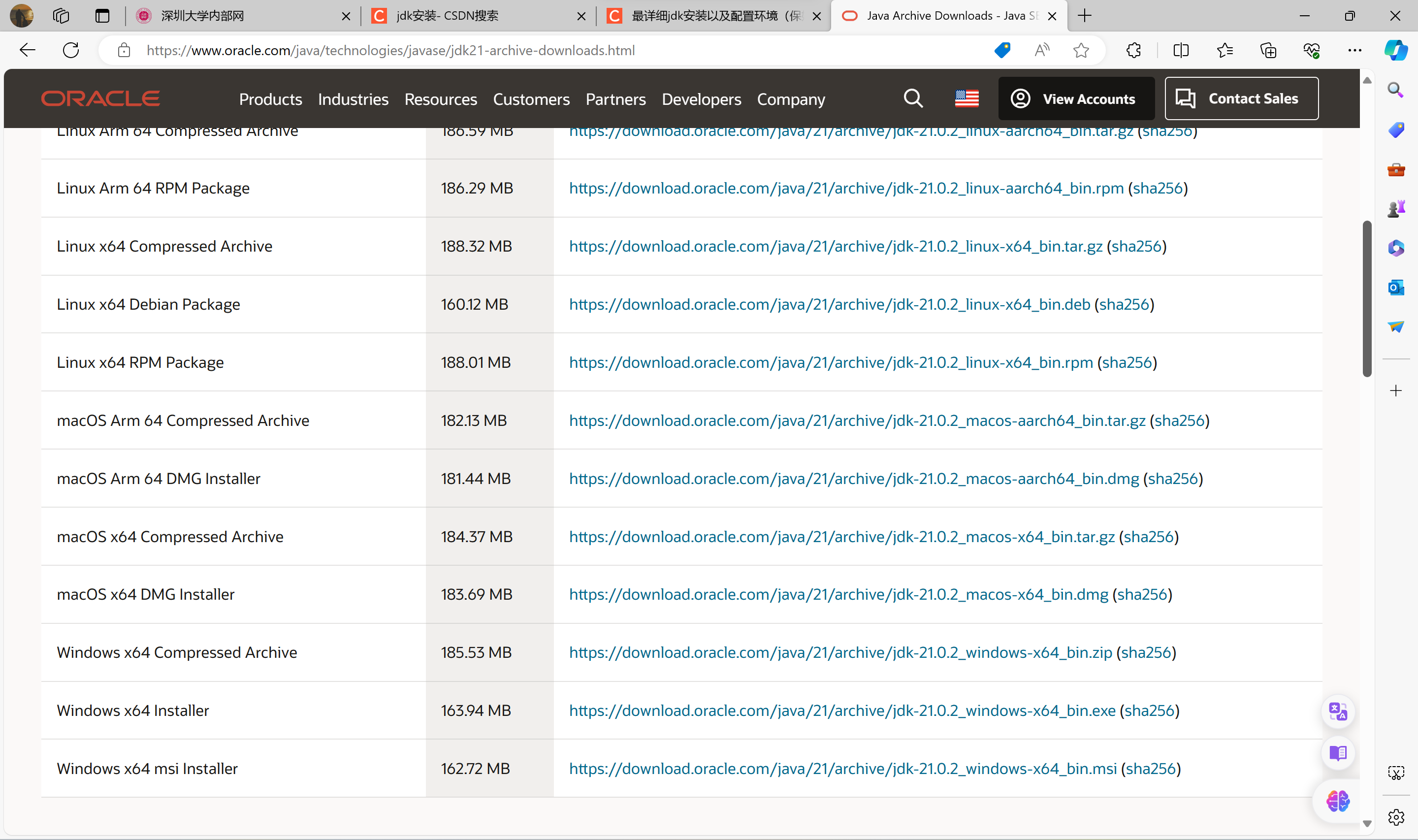
Oracle官网：[Oracle | Cloud Applications and Cloud Platform](https://www.oracle.com/)



1.2 查看本系统的系统类型



1.3 根据系统类型为64 位操作系统，下载对应的jdk （Java SE Development Kit 21.0.2）



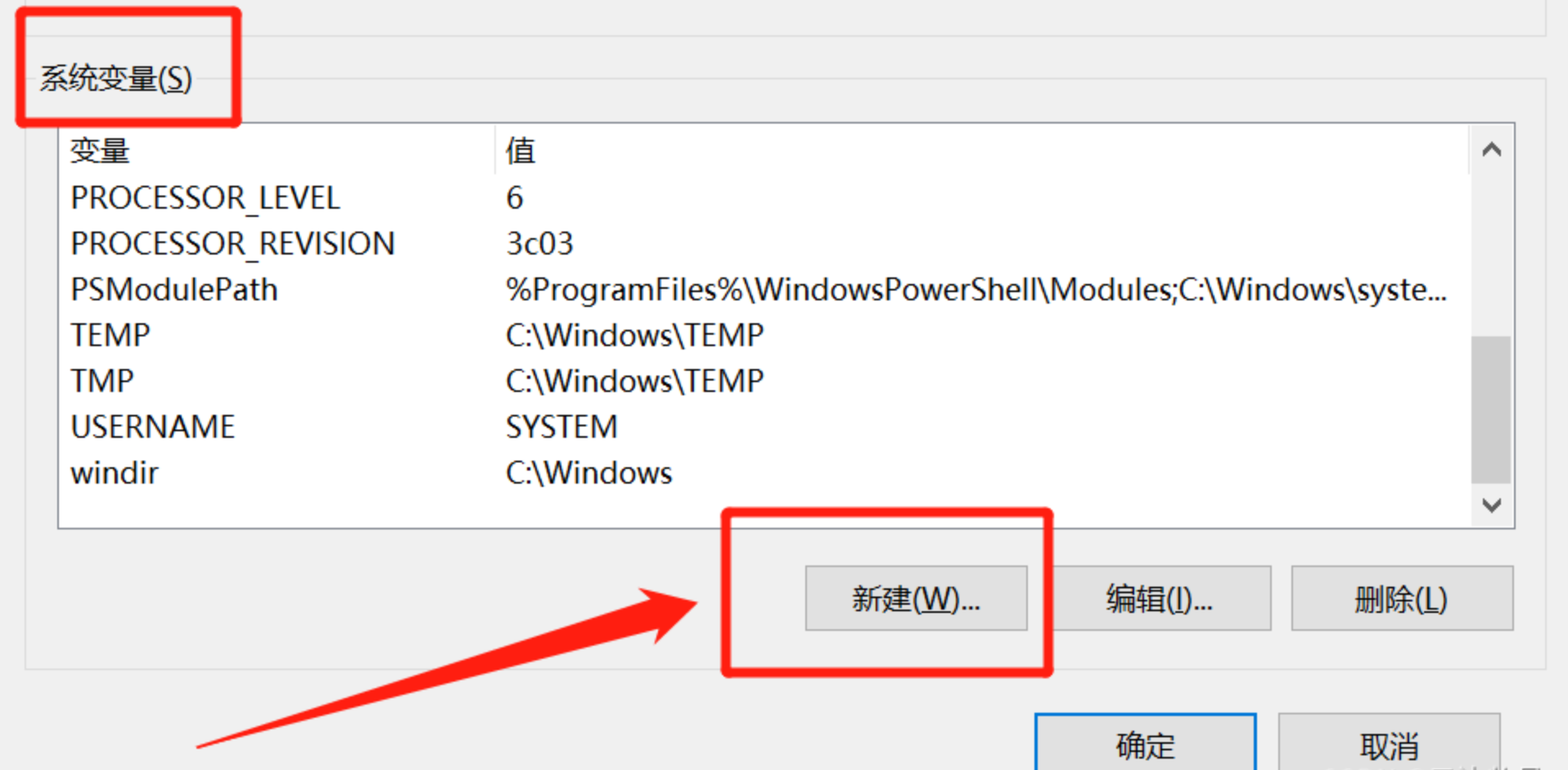
1.4 Jdk21 的安装过程：按照指示来即可

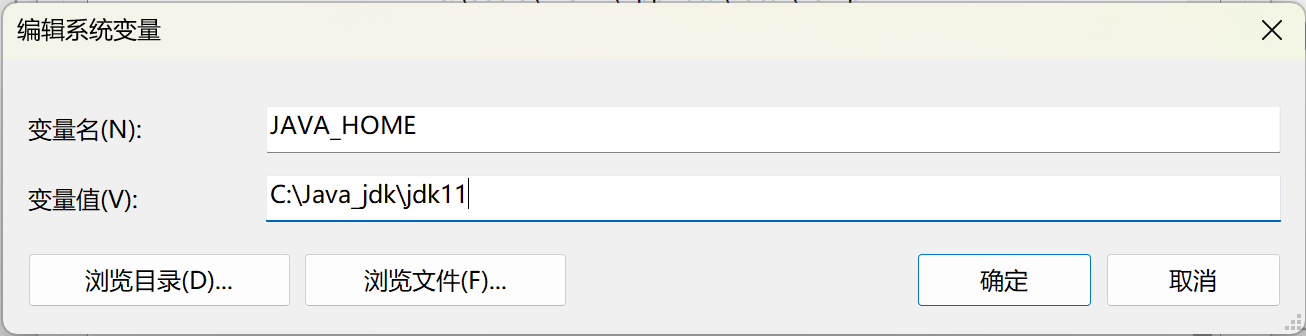
1.5 打开系统信息页面，点进“高级系统设置”，进而点击“环境变量”

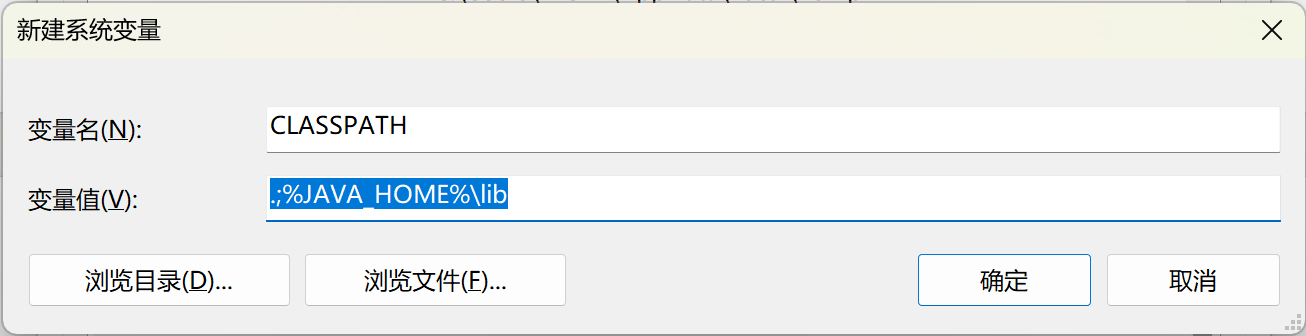
1.6 配环境：先新建系统变量“JAVA\_HOME”



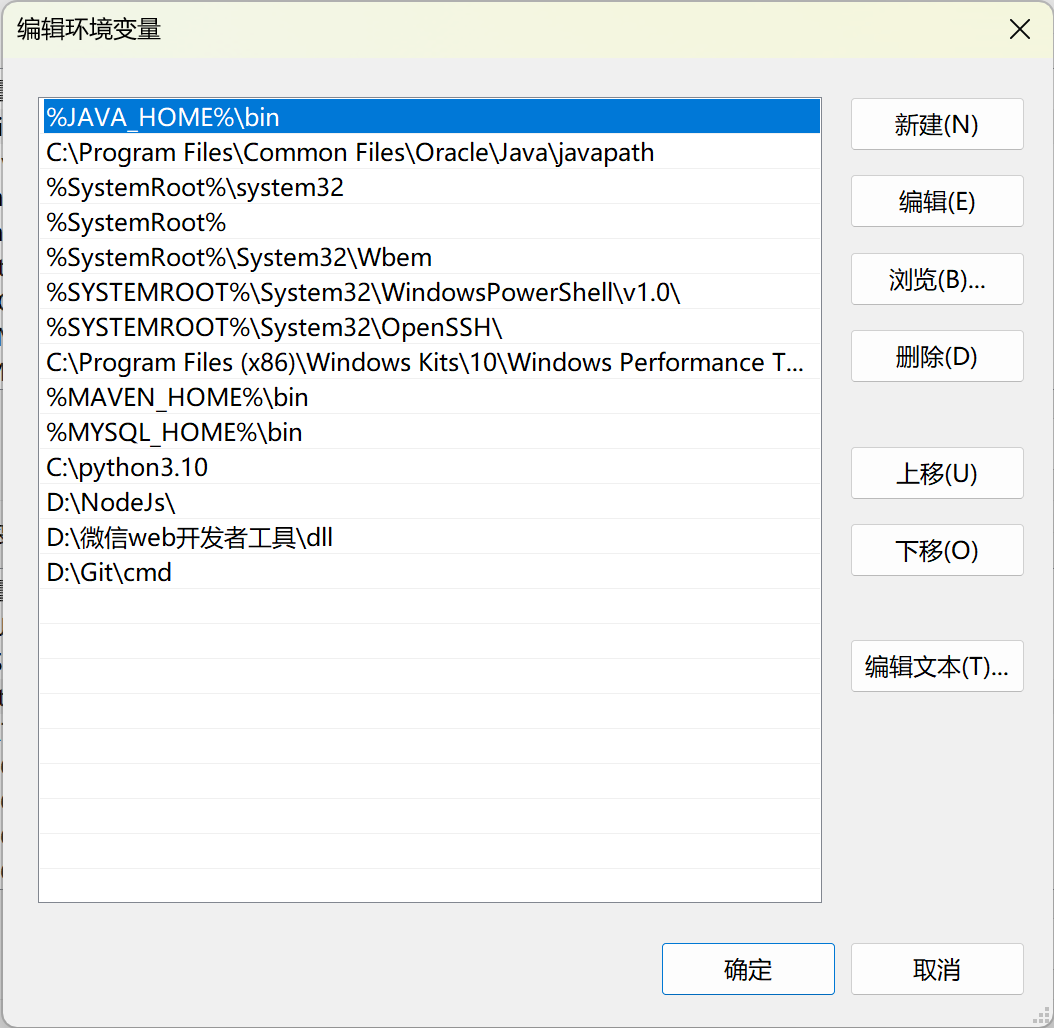
变量值修改为当时JDK安装所在的文件夹



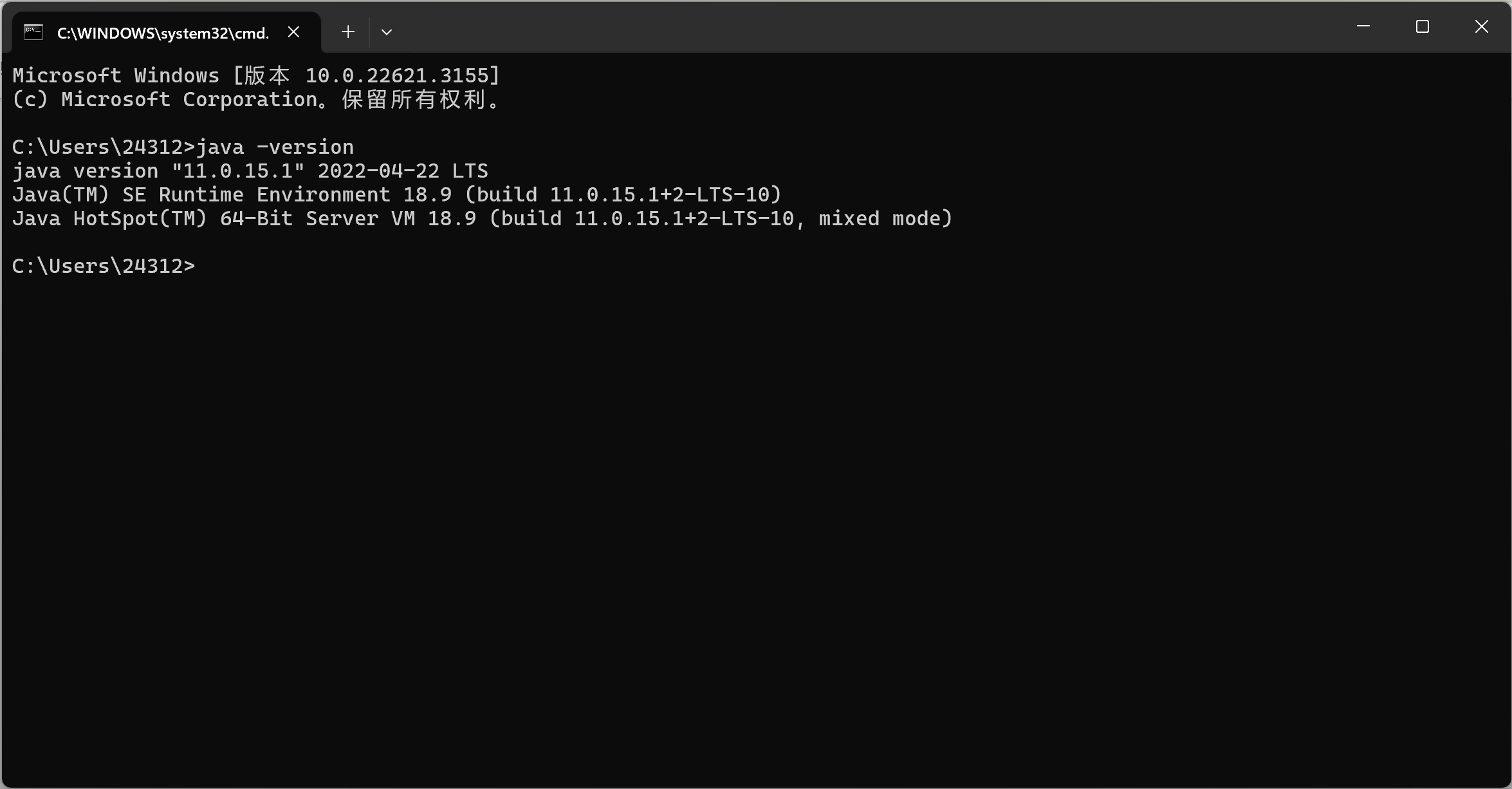
紧接着再点击新建变量名“CLASSPATH”，添加变量值：.;%JAVA\_HOME%\lib



然后在系统变量（S）中找到 Path，单击一下它，紧接着点击 编辑 ，在Path内点击新建“%JAVA\_HOME%\bin”



1.7 验证jdk是否配置成功：在电脑桌面按住win＋R键→输入cmd，然后输入java -version：

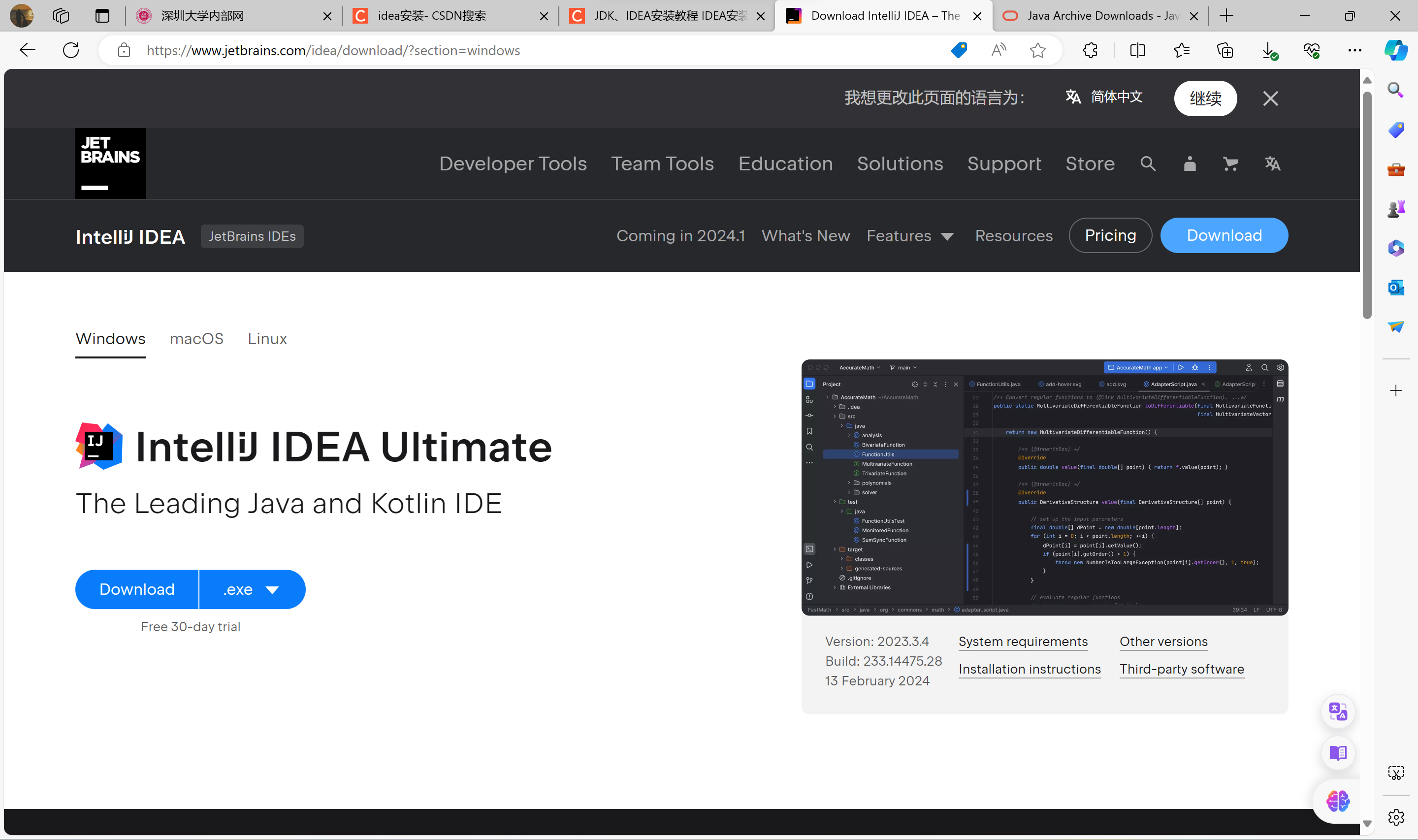


可见jdk已经配置成功

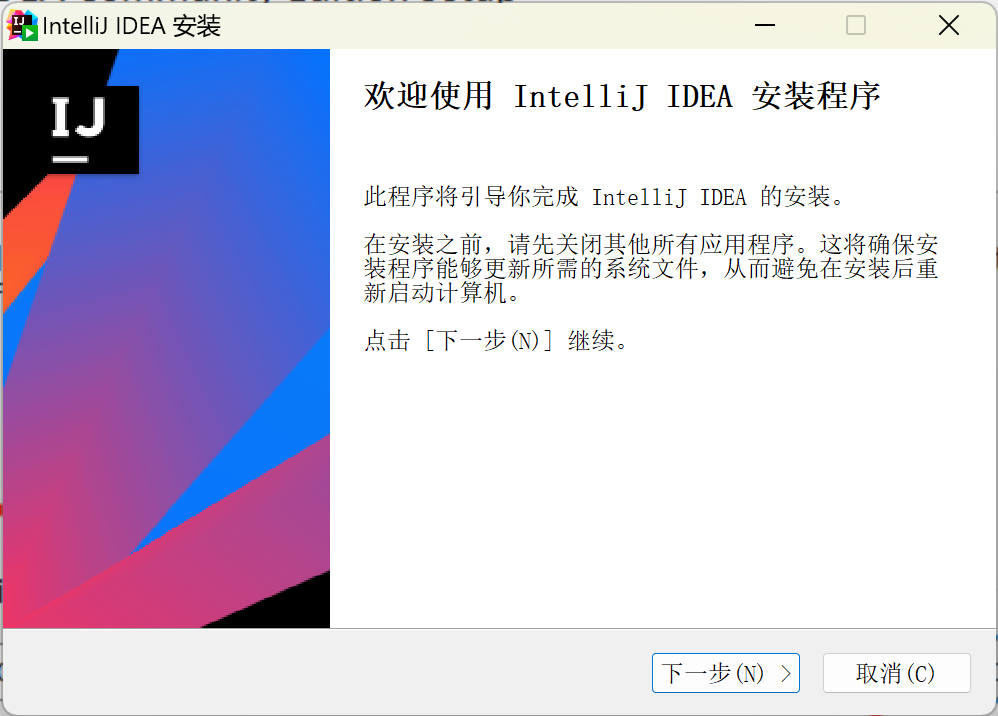
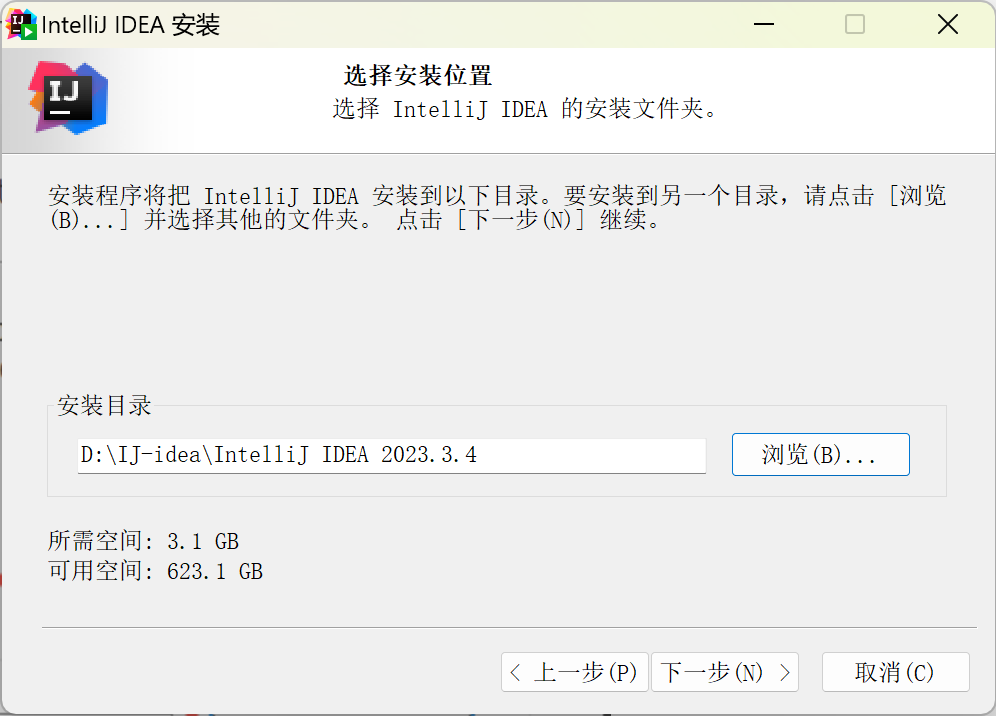
1.8 安装IntelliJ IDEA

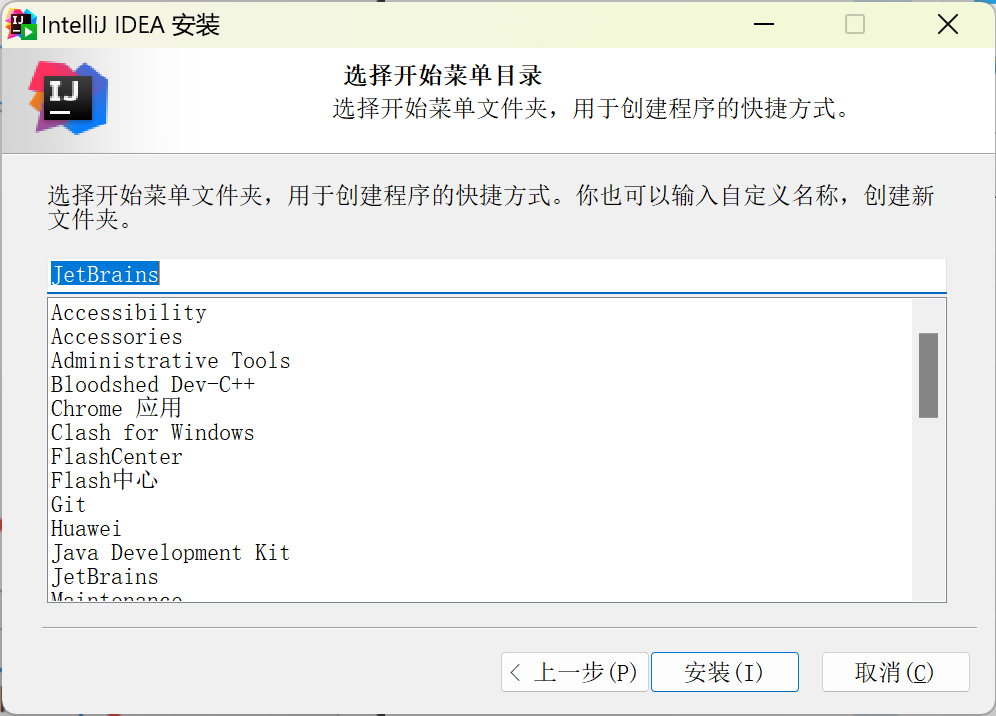
官网链接：[IntelliJ IDEA – the Leading Java and Kotlin IDE (jetbrains.com)](https://www.jetbrains.com/idea/)

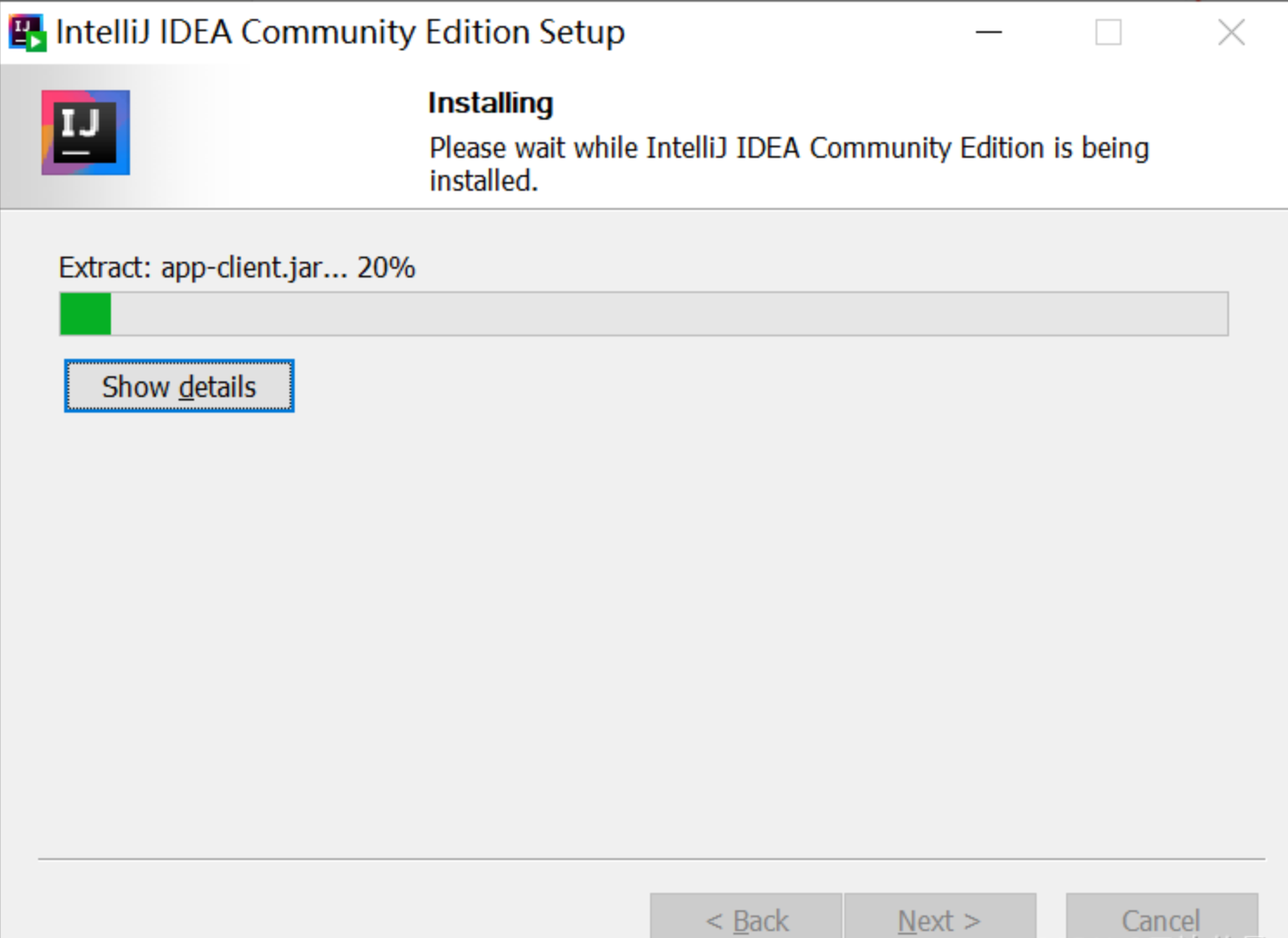
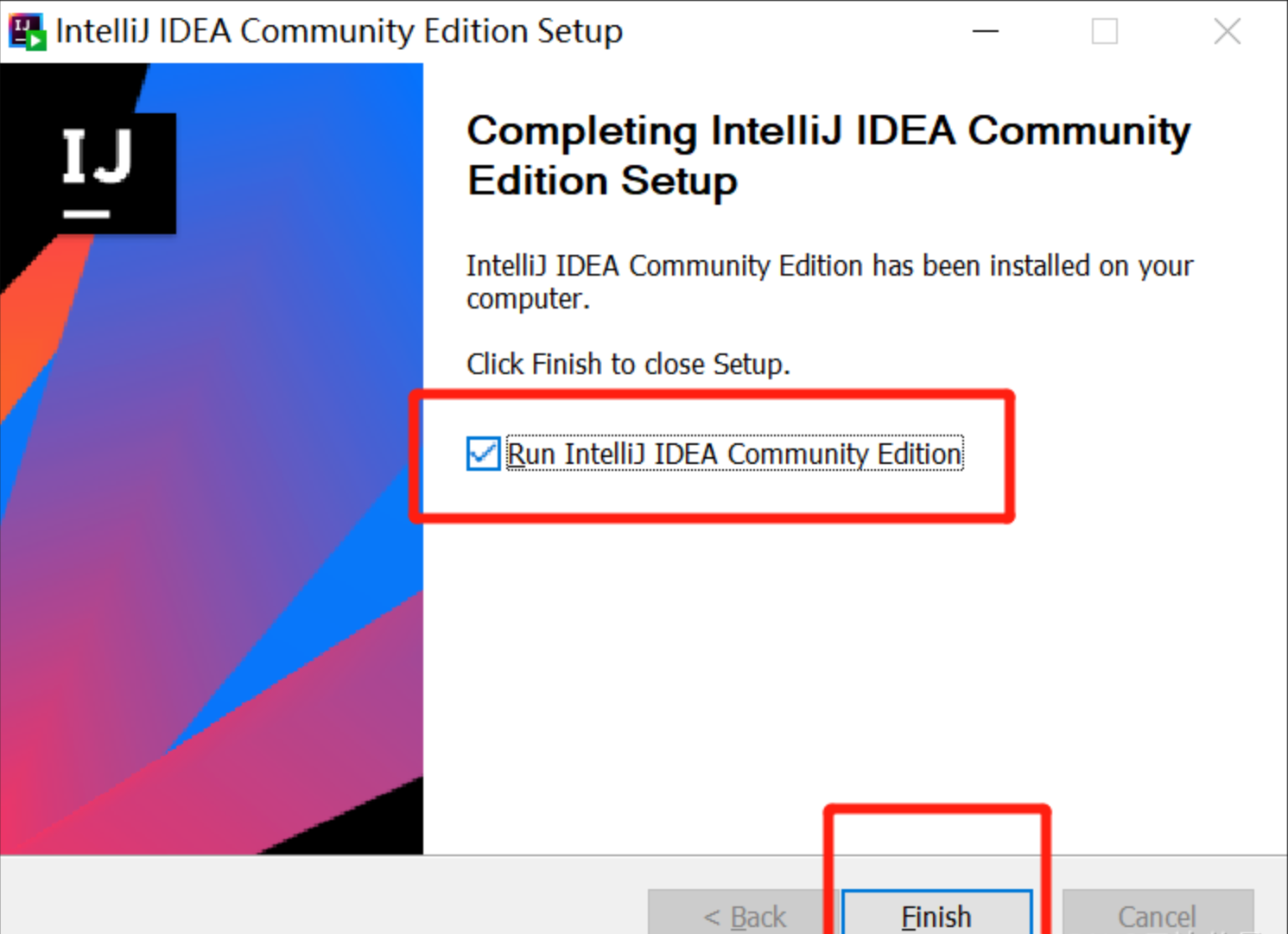
下载windows版本的IDEA：



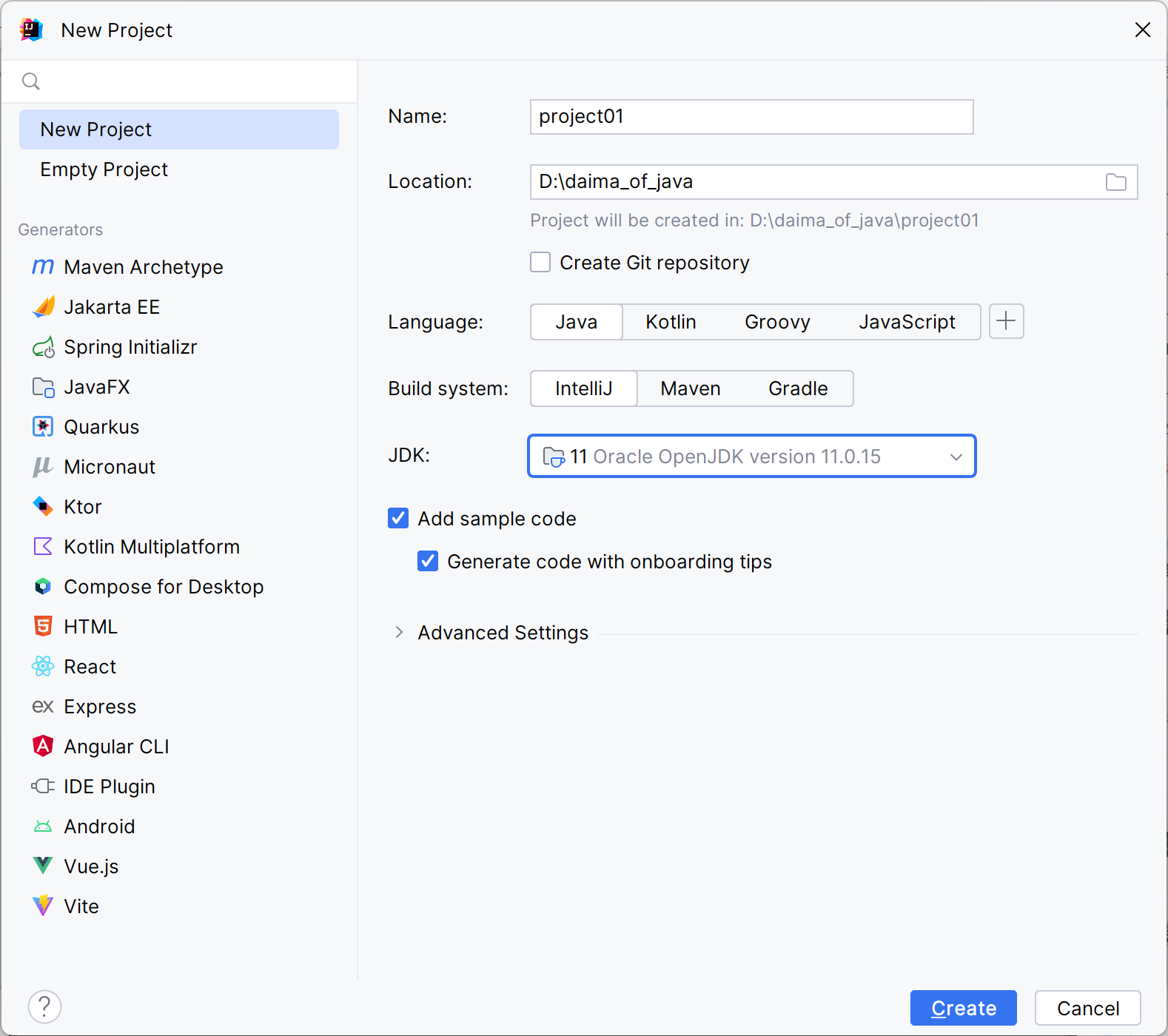
1.9 安装过程：

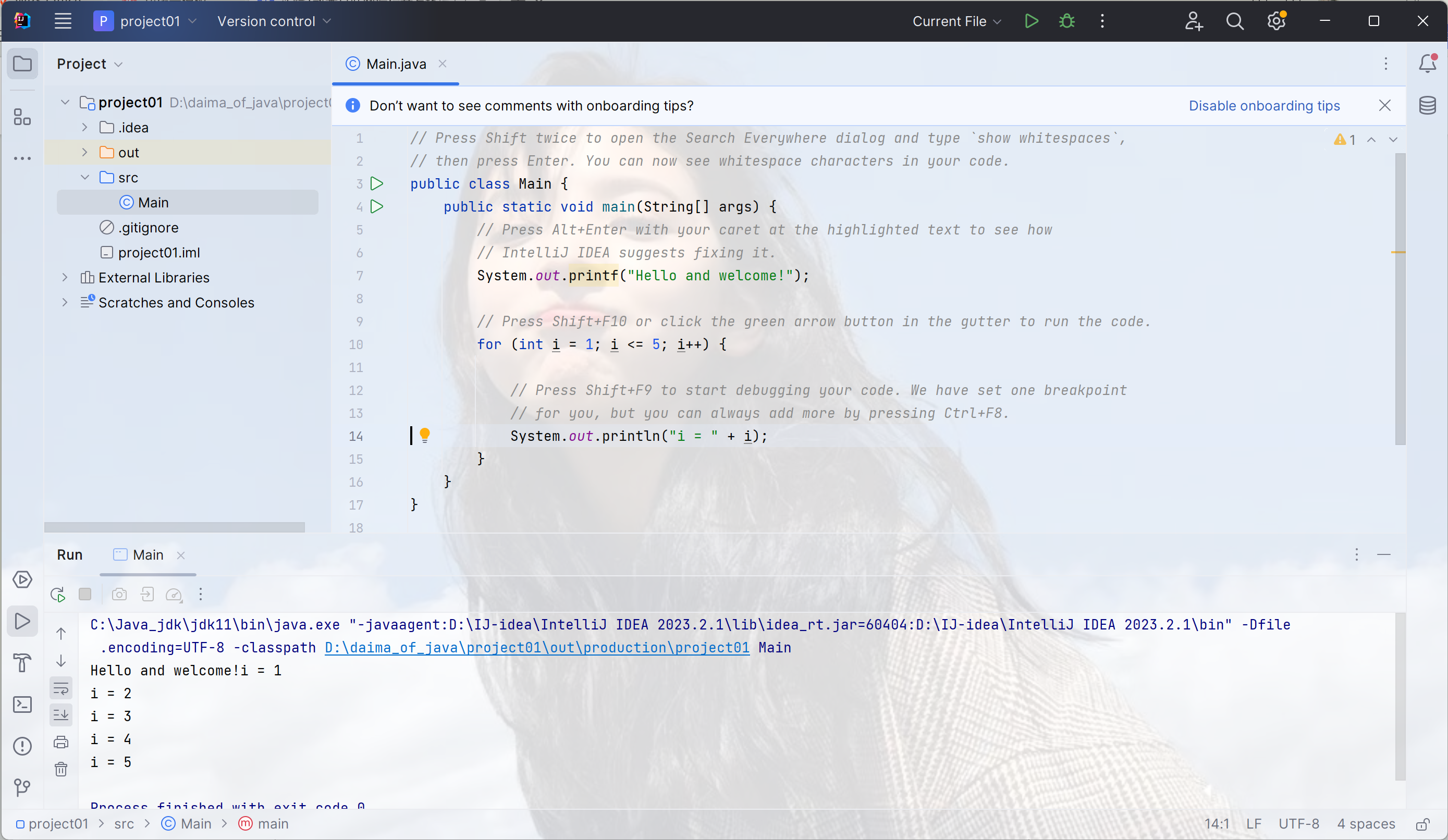
 

1.10 新建一个项目，设置项目名，文件路径，配置JDK版本



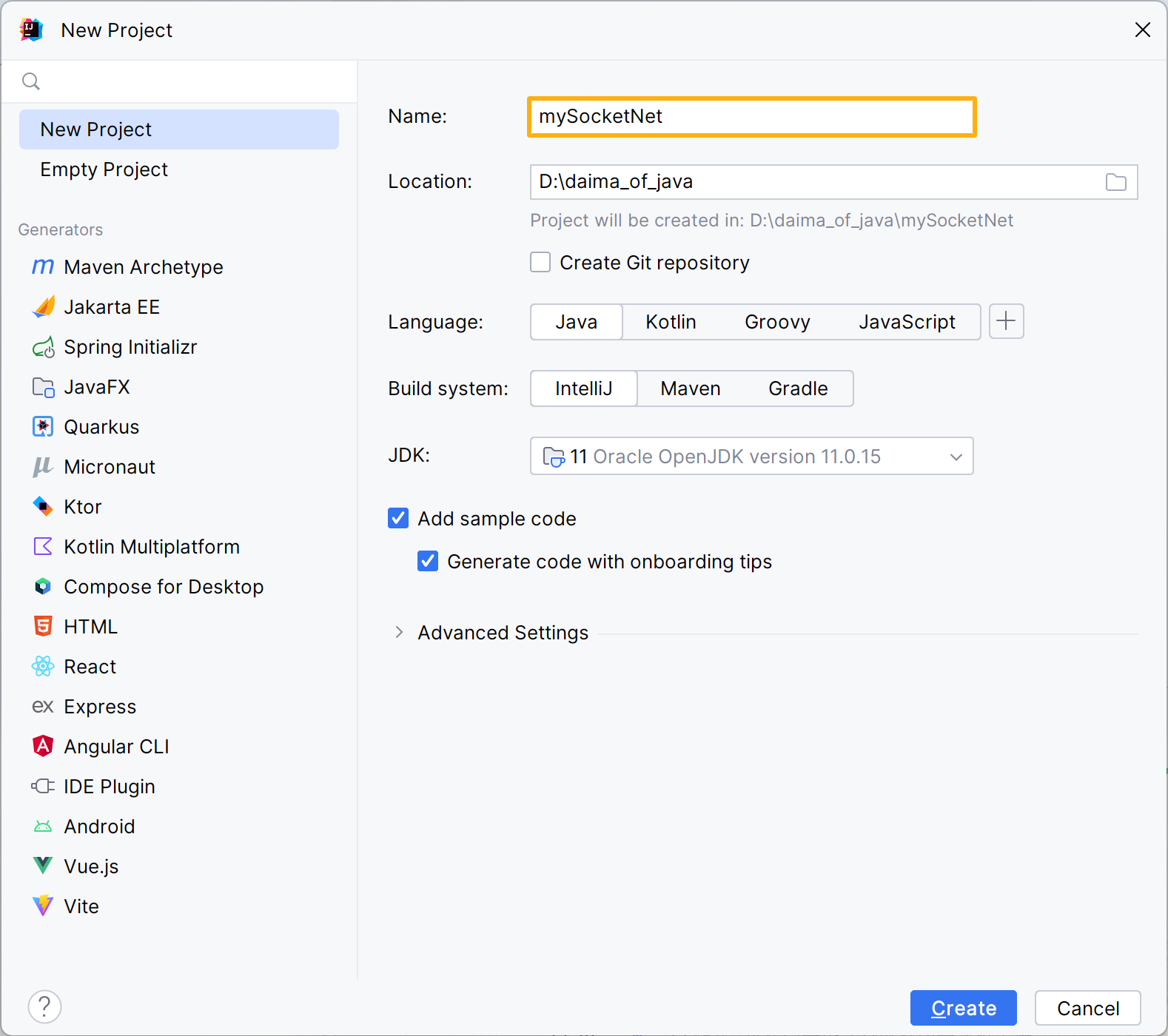
1.11 点击右上角运行图标，测试环境是否配置成功



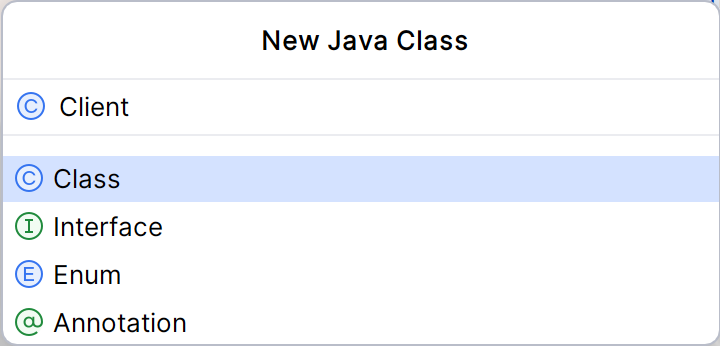
根据输出结果，可见，环境已配置成功

**2. 自己建立的程序实验过程、结果及代码**

2.1 创建一个新的project，命名为：mySocketNet，配置jdk版本为11



2.2 创建服务端Sever.java以及客户端Client.java

2.3 编写服务端程序

1. 创建服务端对象 并 绑定 6666端口，然后监听等待客户端连接
2. 向客户端发送通知，告知通信连接已建立
3. 接受客户端发来的消息
4. 释放socket和服务器资源

具体代码如下：

*//引入相关模块*

import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStream;  
import java.net.ServerSocket;  
import java.net.Socket;  
*//服务端*public class Sever {  
 public static void main(String[] args) throws IOException {  
 *//创建服务端对象 并 绑定 6666端口* ServerSocket serverSocket = new ServerSocket(6666);  
 *//监听等待客户端连接* Socket socket = serverSocket.accept();  
  
 *//从socket中获取输入流、输出流* InputStream ins = socket.getInputStream();  
 OutputStream os = socket.getOutputStream();  
 *//将字节流进行转化，以防止 中文乱码 问题* InputStreamReader insReader = new InputStreamReader(ins);  
  
 *//向客户端发送通知，告知通信连接已建立* os.write("尊敬的用户，欢迎来到留言墙，小墙墙将耐心倾听您的想法！\n".getBytes());  
 *//关闭输出通道，防止 客户端接收阻塞* socket.shutdownOutput();  
  
 *//接受客户端发来的消息* int ch,flag=1;  
 while ((ch = insReader.read()) != -1) {  
 if(flag==1) System.*out*.print("用户留言：");  
 flag = 0; *//防止 每接受一个字符就打印一次提示语  
 //将依次读取到的字符进行输出* System.*out*.print((char)ch);  
 if((char)ch == '\n'){ *//已接收到一句完整的消息* flag = 1; *//可以打印提示语* }  
 }  
 System.*out*.println("用户关闭了通信！");  
  
 *//释放socket和服务器资源* socket.close();  
 serverSocket.close();  
 }  
}

2.4 编写客户端程序

1. 创建socket对象 并 连接服务端
2. 接受服务器发来的提示消息，准备发送消息给服务器
3. 不断发送信息给服务器
4. 释放socket资源

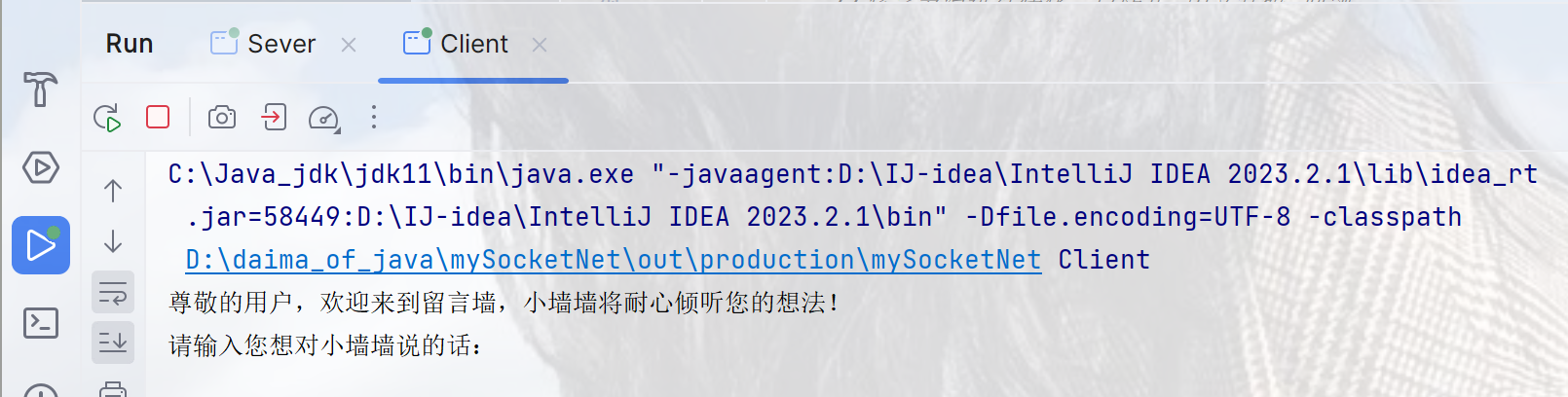
具体代码如下：

*//引入相关模块*import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStream;  
import java.net.Socket;  
import java.util.Scanner;  
*//客户端*public class Client {  
 public static void main(String[] args) throws IOException {  
 *//创建socket对象 并 连接服务端* Socket socket = new Socket("127.0.0.1",6666);  
 *//创建Scanner对象 以 手动输入数据* Scanner sc = new Scanner(System.*in*);  
 *//从socket中获取输出流、输入流* OutputStream os = socket.getOutputStream();  
 InputStream ins = socket.getInputStream();  
 *//将字节流进行转化，以防止 中文乱码 问题* InputStreamReader insReader = new InputStreamReader(ins);  
  
 *//接受服务器发来的提示消息* int ch;  
 while ((ch = insReader.read()) != -1) {  
 *//将依次读取到的字符进行输出* System.*out*.print((char)ch);  
 }  
 *//不断发送信息给服务器* while (true){  
 System.*out*.print("请输入您想对小墙墙说的话：");  
 *//手动输入消息* String str = sc.nextLine();  
 *//循环退出条件* if(str.equals("拜拜") || str.equals("88"))break;  
 *//发送消息给 服务端* os.write((str+"\n").getBytes());  
 System.*out*.println("小墙墙已收到了您的留言！");  
 }

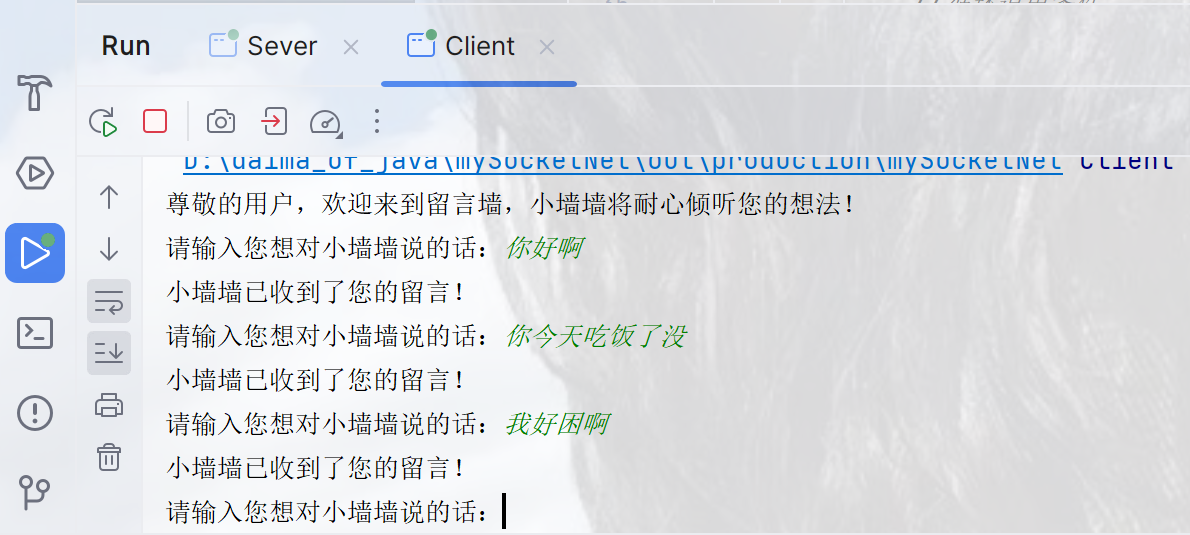
System.*out*.println("会话结束，小墙墙还会想你的！");  
 *//释放socket资源* socket.close();  
 }  
}

2.5 检验程序实现的效果：

2.5.1先打开服务端，再打开客户端，发现客户端已收到服务端发来的提示信息：



2.5.2 让客户端连续向服务端提交留言消息：

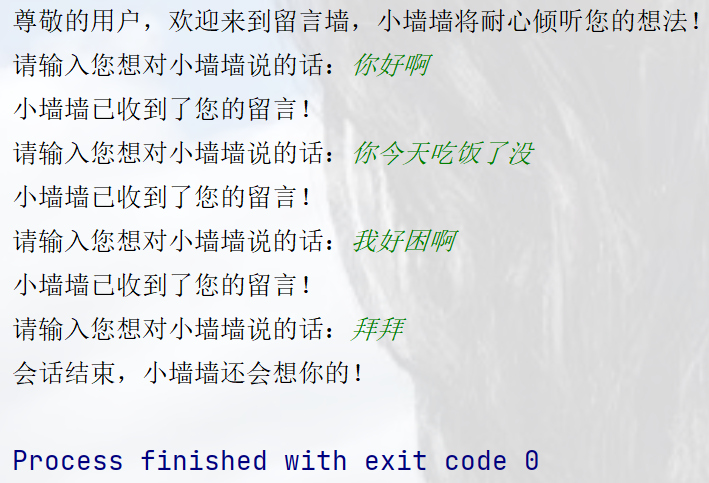


2.5.3 查看服务端接收的消息通知列表：

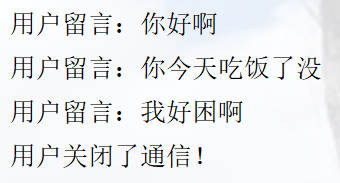


2.5.4结束会话：

首先客户端程序关闭：



其次，服务端程序关闭：



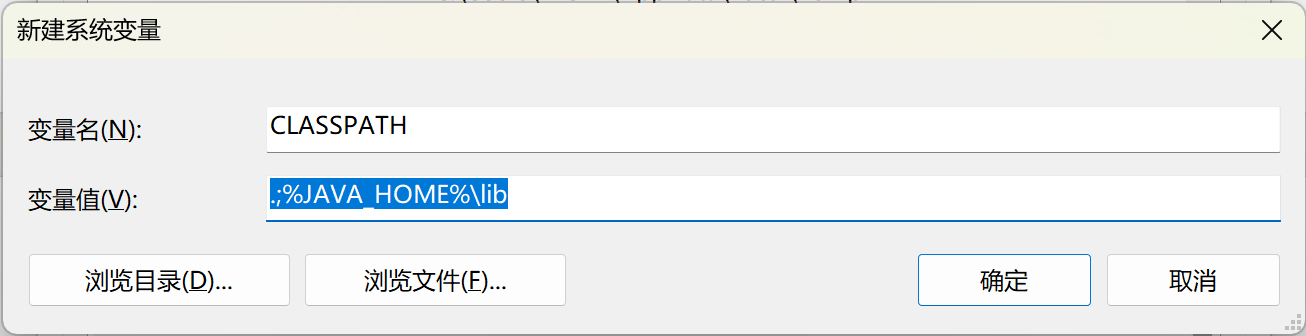
三、实验总结

（此处写写比如遇到的错误，以及解决方法，你的所想、所得等）

实验中遇到的问题及解决方法：

问题一：在命令行输入java -version后，系统提示找不到jdk，即环境没有配置成功

解决一：在csdn上搜索找到指导方案，发现是CLASSPATH没有配置



问题二：IDEA专业版未激活，只能试用30天

解决二：使用以学生邮箱激活的Jetbrains账号登录，即可使用专业版软件。或者使用社区版

问题三：服务端接受到的消息出现中文乱码问题

解决三：查询资料发现是因为字节流一次读写一个字节，而中文是两个或三个字节，因此使用转换流可以解决此问题

*//从socket中获取输入流、输出流*InputStream ins = socket.getInputStream();

*//将字节流进行转化，以防止 中文乱码 问题*InputStreamReader insReader = new InputStreamReader(ins);

问题四：服务端给客户端发送提示消息时，客户端在接受处阻塞了

解决四：服务端关闭输出流，防止接收处阻塞

*//关闭输出通道，防止 客户端接收阻塞*socket.shutdownOutput();

**实验心得：**

搭建开发环境是学习编程的重要一步，我选择了 IntelliJ IDEA 作为我的开发环境，并成功安装和配置了它。通过 IntelliJ IDEA，我创建了一个新的 Java 项目，并编写了一个简单的 HelloWorld 程序进行测试。在这个过程中，我学会了如何创建项目、创建类文件，并运行程序进行测试。我还学会了如何导入外部库、调试程序以及使用一些常用快捷键。

在掌握了基本的 Java 编程知识后，我尝试编写了一个简单的客户端-服务器端通信程序。在客户端程序中，我使用了 `Socket` 类来创建套接字，并连接到服务器端进行通信。在服务器端程序中，我使用了 `ServerSocket` 类来监听客户端的连接请求，并通过多线程处理多个客户端的请求。通过编写这个网络通信程序，我深入了解了 Java 网络编程的原理和应用，以及网络通信在实际应用中的重要性。这次实验让我对编程学习的前景充满了信心，我相信通过不断地学习和实践，我会成为一名优秀的程序员。

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：毛斐巧  2025 年 月 日 |
| 备注： |