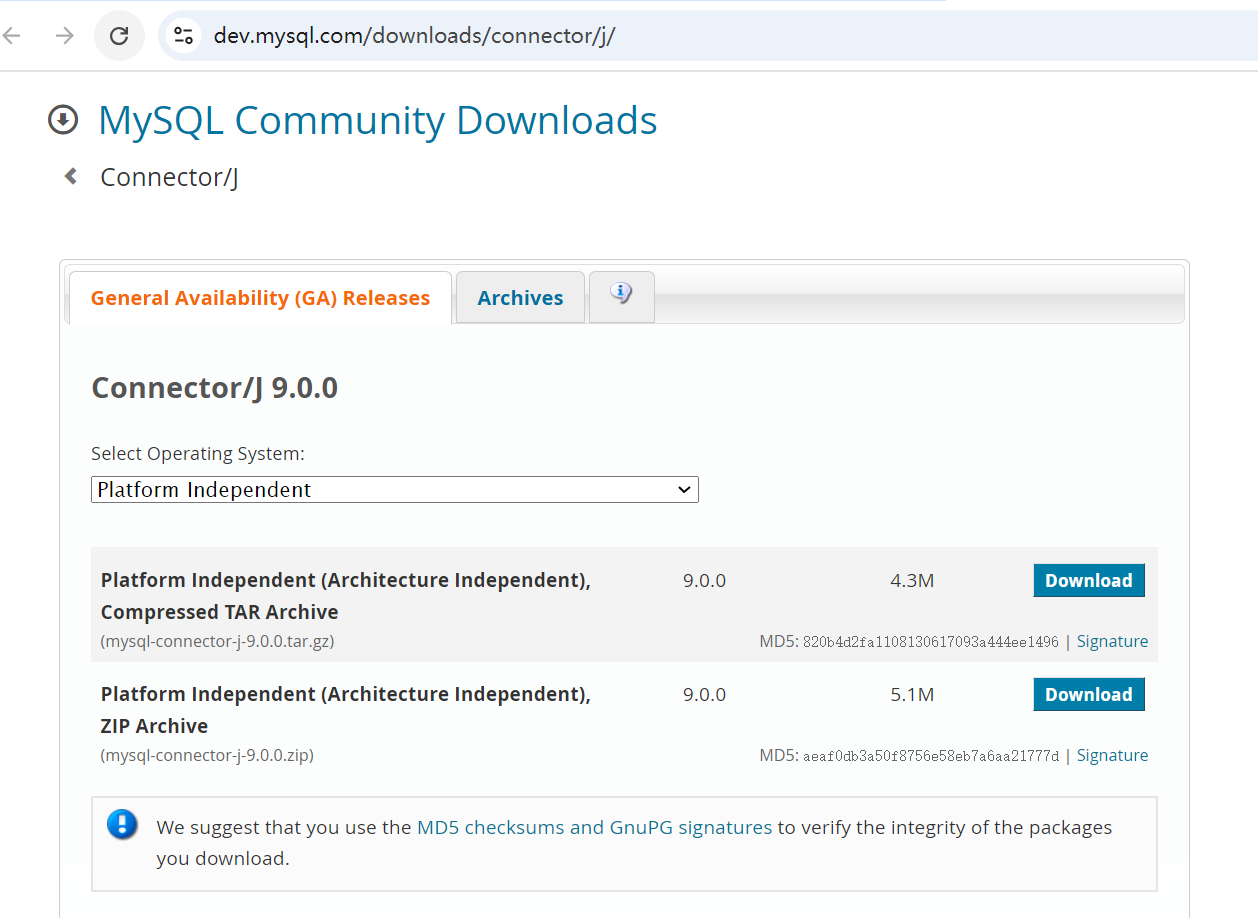
# 一、Java连接数据库MySQL

**测试环境：MySQL 9.0 connector/J 9.0.0 Navicate for MySQL 17(已破译，破译链接https://www.bilibili.com/read/cv38001580/?jump\_opus=1)**

## 下载MySQL连接器

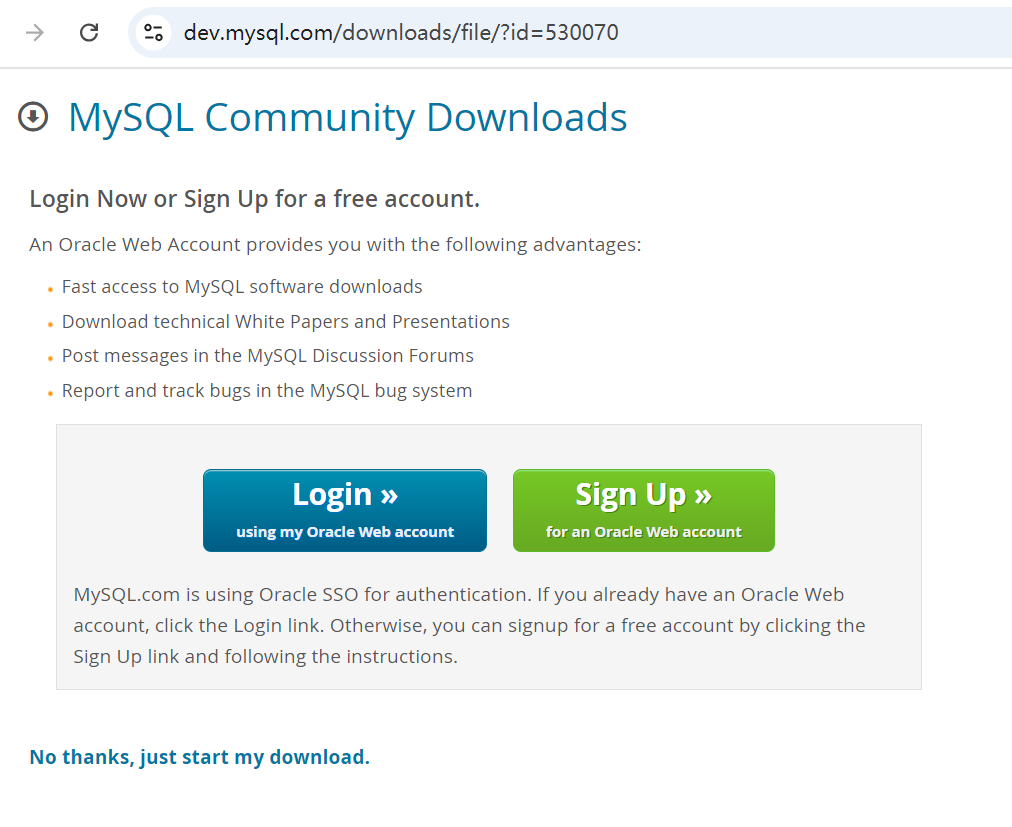
1）进入MySQL官网：[https://dev.mysql.com/downloads/connector/j/](https://dev.mysql.com/downloads/connector/j/" \t "_blank)，下载[mysql-connector-java，因为演示中使用的MySQL版本是9.0，所以下载对应的connector。](https://www.cnblogs.com/junzi/p/6893122.html" \o "发布于 2017-05-23 11:04)



**2**

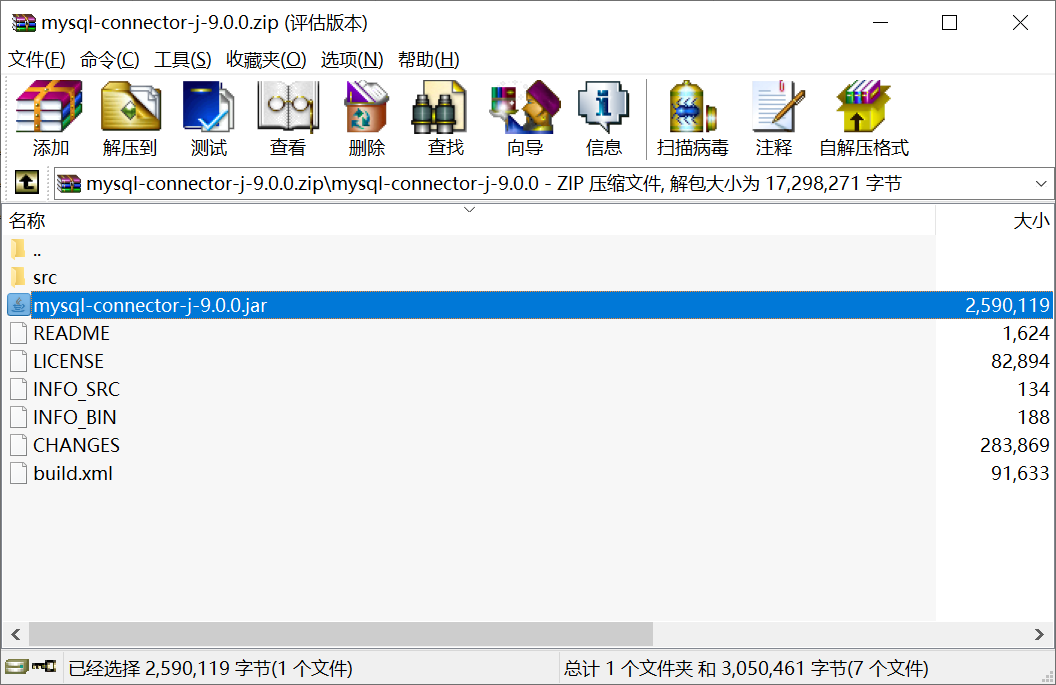
**1**

2）点击Download，会进入下载页面，此时无需登录，直接下载



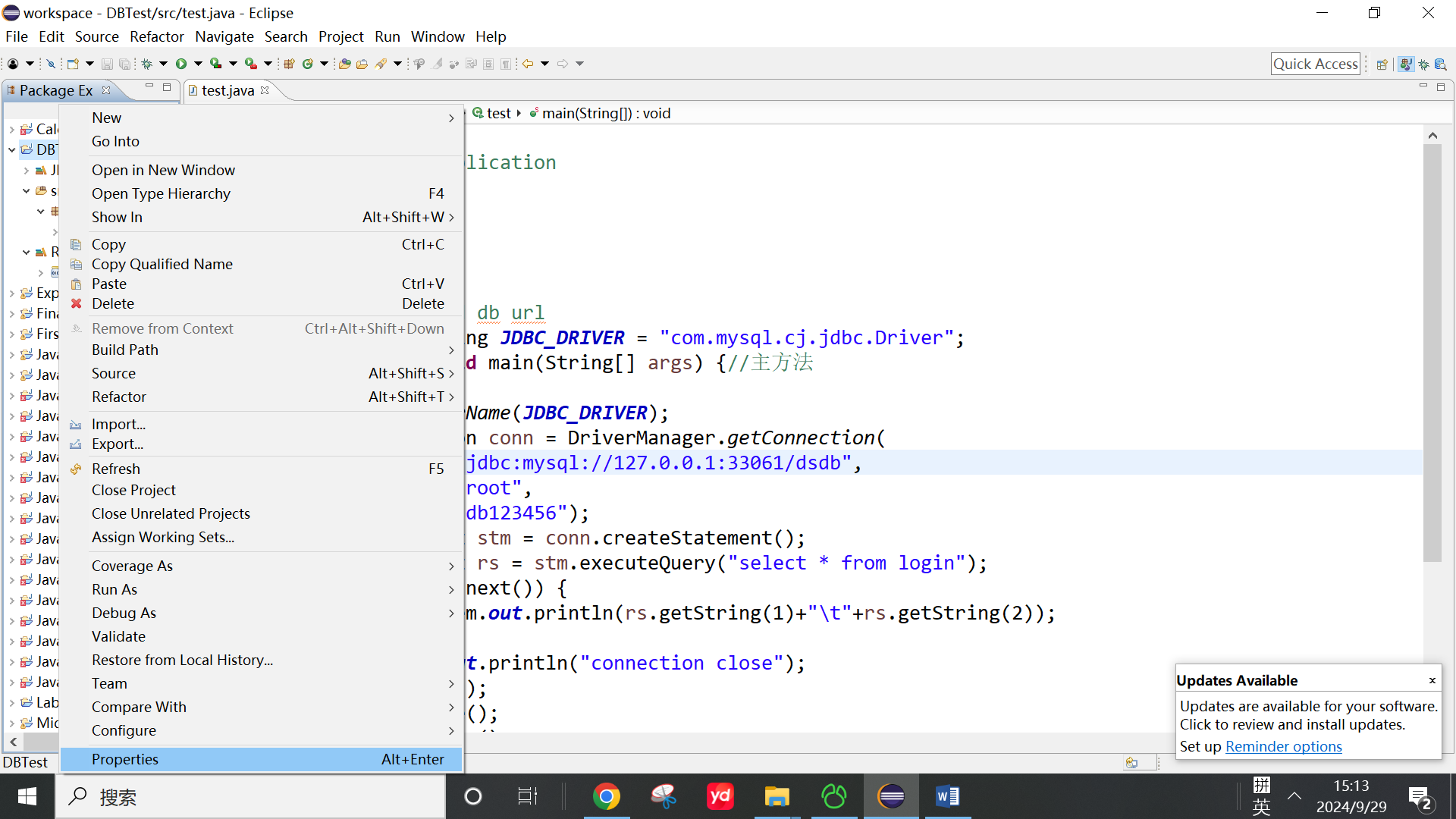
**3**

3）下载后得到zip文件，解压缩后会得到jar包

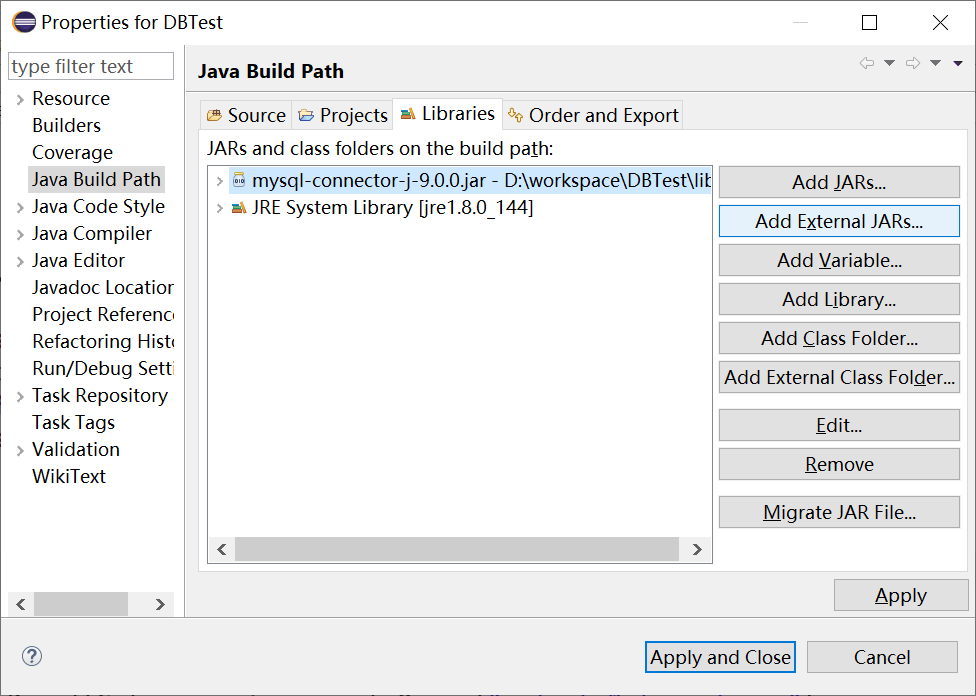


## 项目中添加jar包

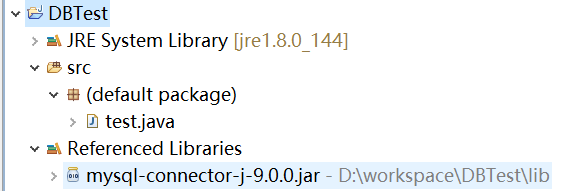
1. 右击创建的java项目，选择Properties，进入属性设置页面



1. 在属性设置页面中，选择“Java Build Path”，选择“Libraries”，选择“Add External JARs”,选择刚刚下载的jar添加至项目中。



1. 加载完成后，项目中可以看到对应的jar包信息：



## 编写代码

java代码要连接数据库，需要进行：

1. 加载驱动
2. 连接数据库
3. 执行命令
4. 获取结果
5. 释放资源

下面分别进行介绍：

1. 加载驱动

Class.forName("com.mysql.cj.jdbc.Driver"); //加载MySQL驱动

1. 连接数据库

连接数据库需要使用java.sql.Connection类；

Connection conn = DriverManager.getConnection(

"jdbc:mysql://127.0.0.1:33061/dsdb", //数据库服务器地址：端口号/数据库名

"root", //账号

"db123456"); //密码

1. 执行命令

执行命令使用java.sql.Statement类；

Statement stm = conn.createStatement();

ResultSet rs = stm.executeQuery("select \* from login");

1. 获取结果

Statement类中提供的executeQuery()方法，用于执行查询语句，成功执行后返回一个ResultSet类（结果集）对象。对该对象进行遍历，可以获取查询到的每个数据。

1. 释放资源

rs.close(); //释放结果集资源

stm.close(); //释放命令

conn.close();//释放连接

代码示例：

import java.sql.\*;

public class test {

public static void main(String[] args) {

try {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection conn = DriverManager.getConnection(

"jdbc:mysql://127.0.0.1:33061/dsdb",

"root",

"db123456");

Statement stm = conn.createStatement();

ResultSet rs = stm.executeQuery("select \* from login");

while(rs.next()) {

System.out.println(rs.getString(1)+"\t"+rs.getString(2));

}

System.out.println("connection close");

rs.close();

stm.close();

conn.close();

}catch(SQLException se) {

se.printStackTrace();

}catch(Exception e) {

e.printStackTrace();

}

}

}

# 二、WindowBuilder插件的安装

## 测试环境

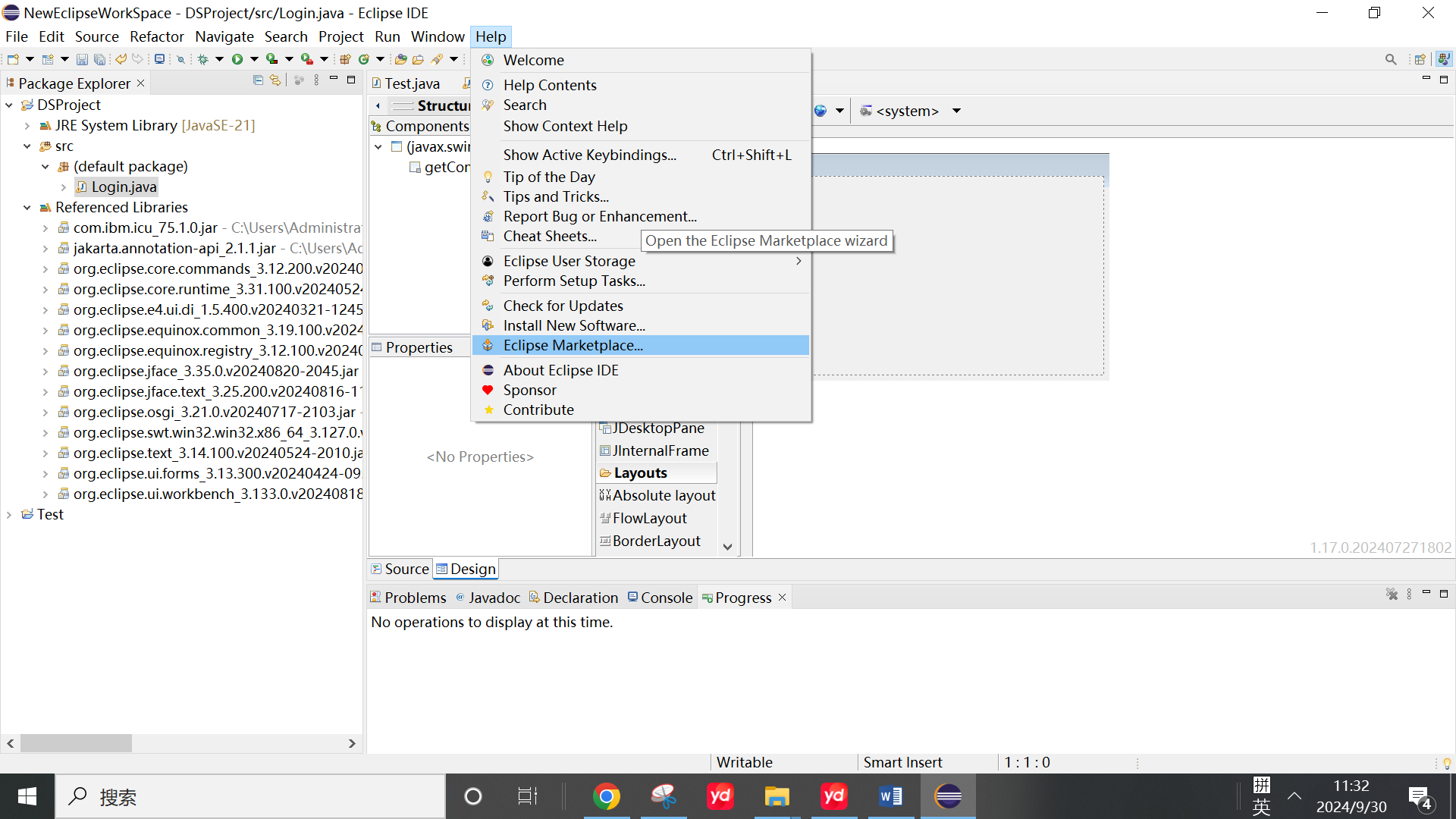
1. JDK21: https://www.oracle.com/cn/java/technologies/downloads/#java21
2. Eclipse IDE 2024-09 R Packages: https://www.eclipse.org/downloads/packages/

选择

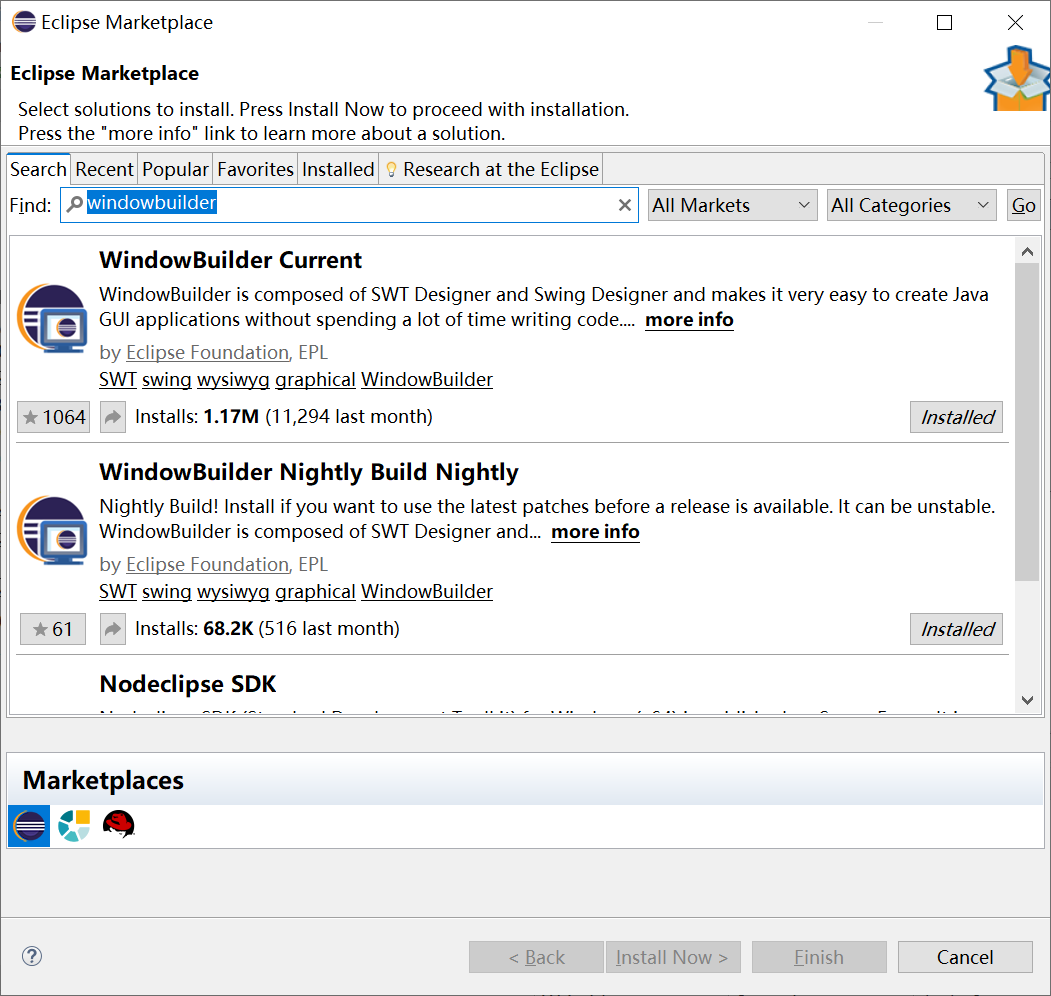
说明：这里建议使用最新版本的Eclipse IDE，然后通过Eclipse Marketplace在线搜索最新的WindowBuilder的插件，如果是老版本的Eclipse IDE，需要下载对应版本的WindowBuilder，如果版本不对的话，可能无法安装，或者安装后看不到WindowBuilder项目类型。

## 插件安装

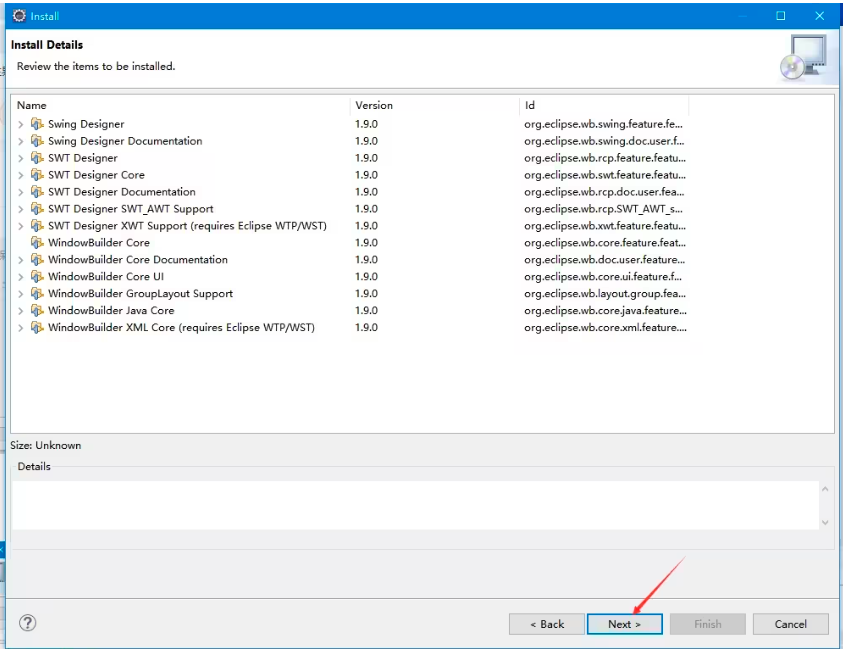
1. 打开Eclipse，选择菜单Help->Eclipse Marketplace



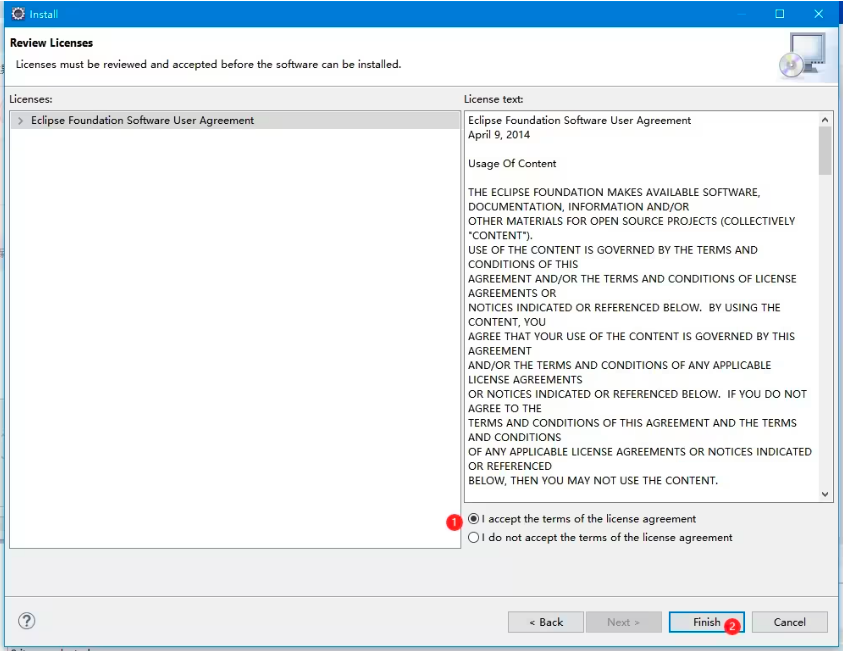
1. 在Search中输入WindowBuilder进行搜索，在搜索结果中安装WindowBuilder Current（此处截图是在安装好插件后截的，所以此处显示Installed，没有安装时此处显示Install）



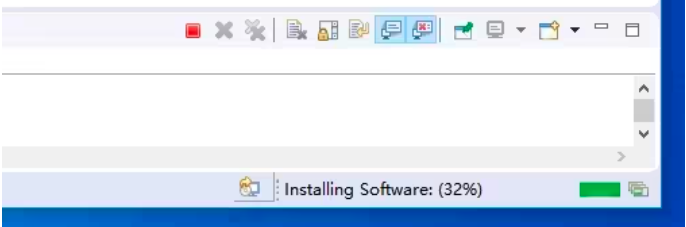
1. 按照提示，确认安装所有选项，然后点击Next……

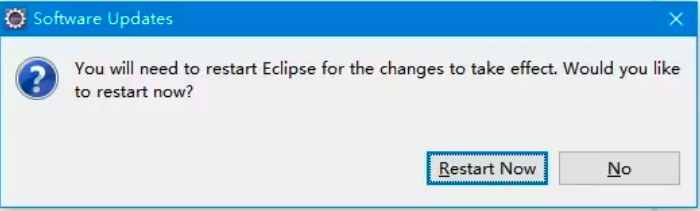


1. 同意条款，点击完成



1. Eclipse右下角会显示安装进度，安装完成后会进行重启Eclipse：

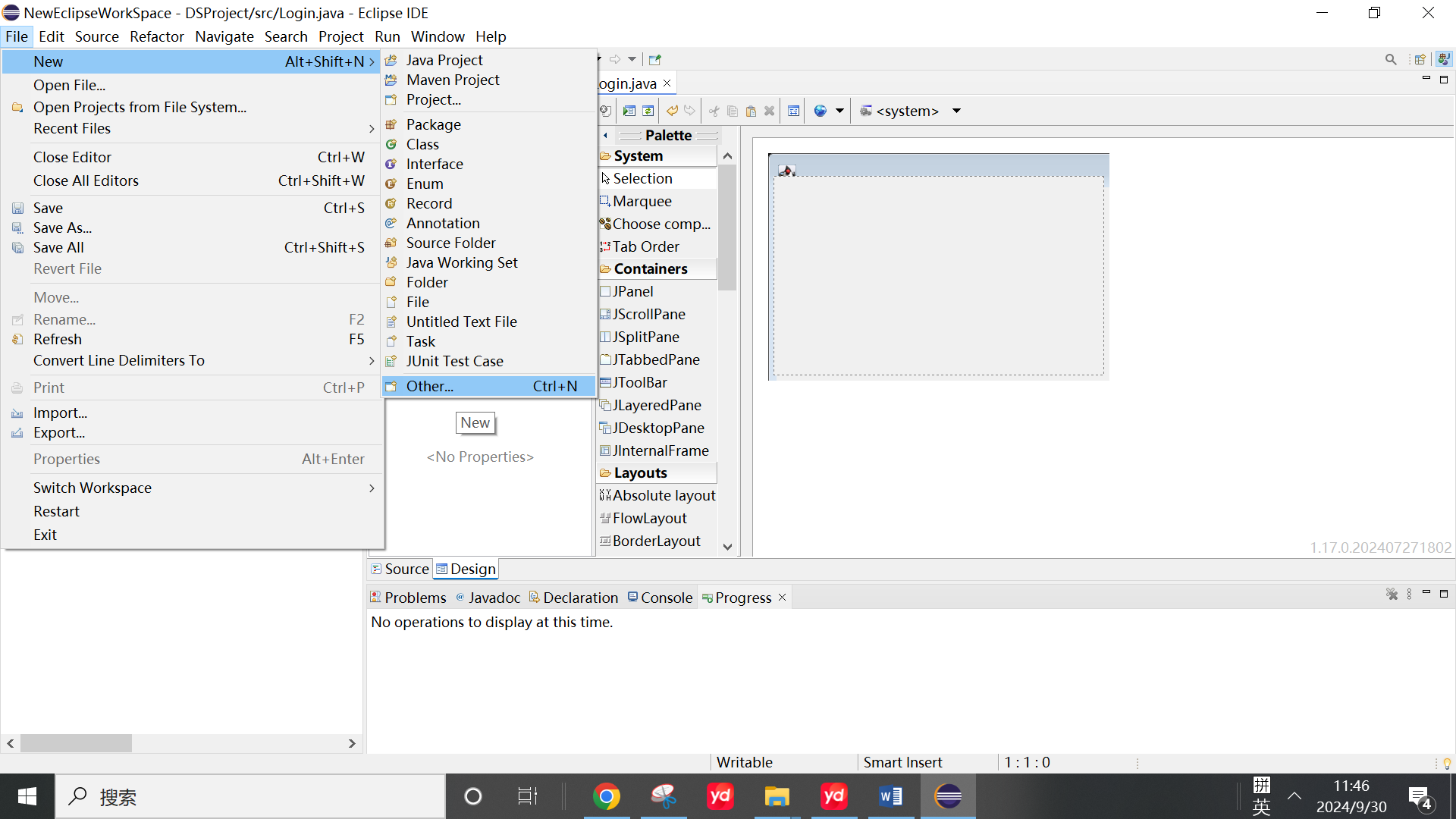




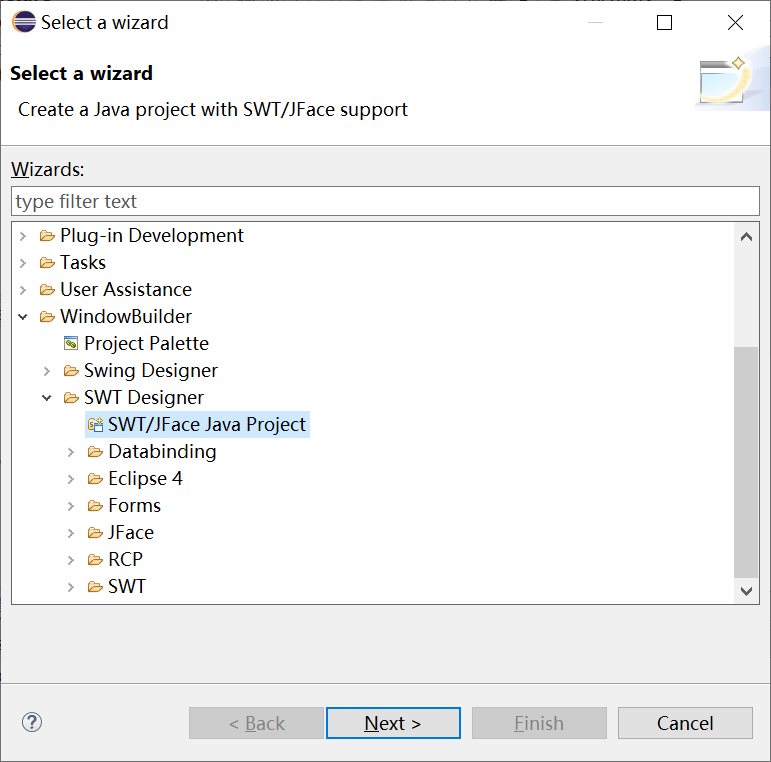
## 可视化下创建GUI应用程序

1. 创建项目

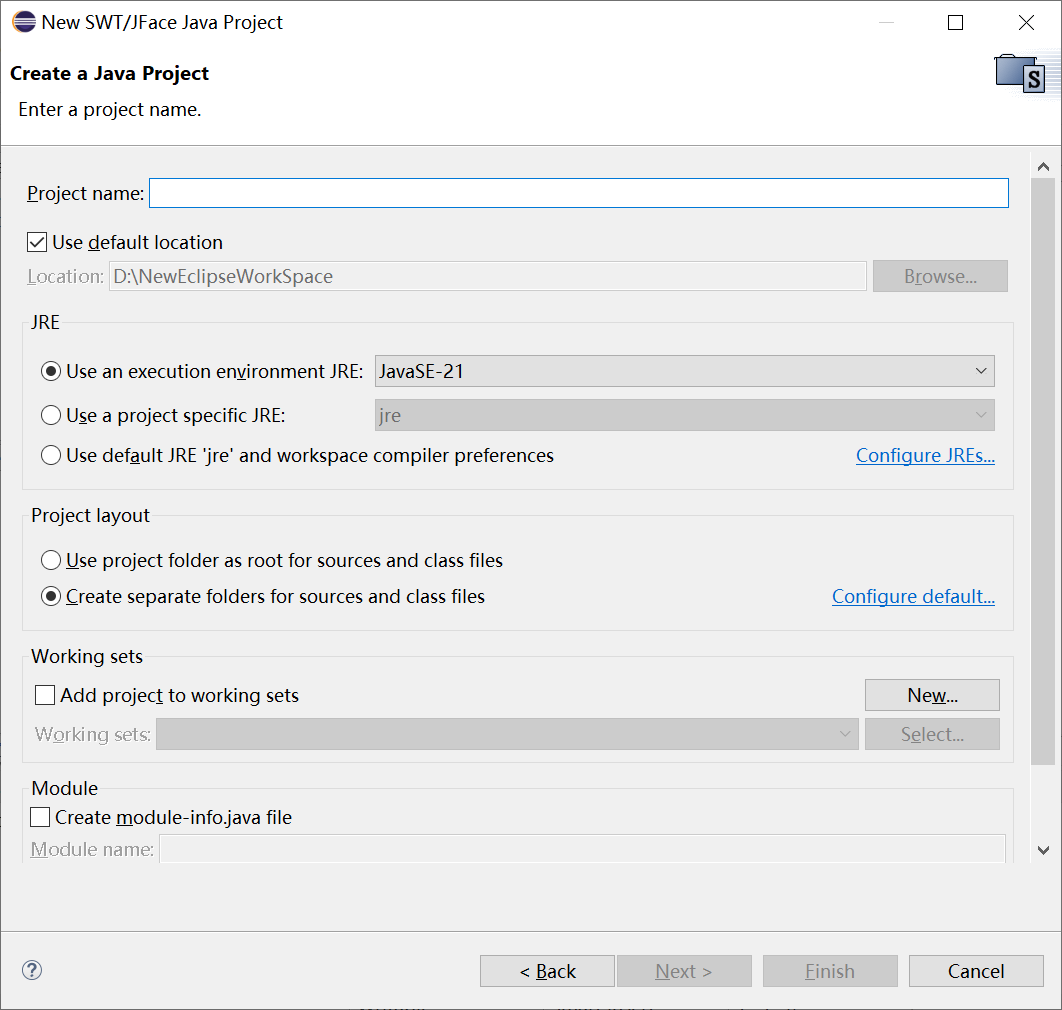
菜单中选择：File->New->Other



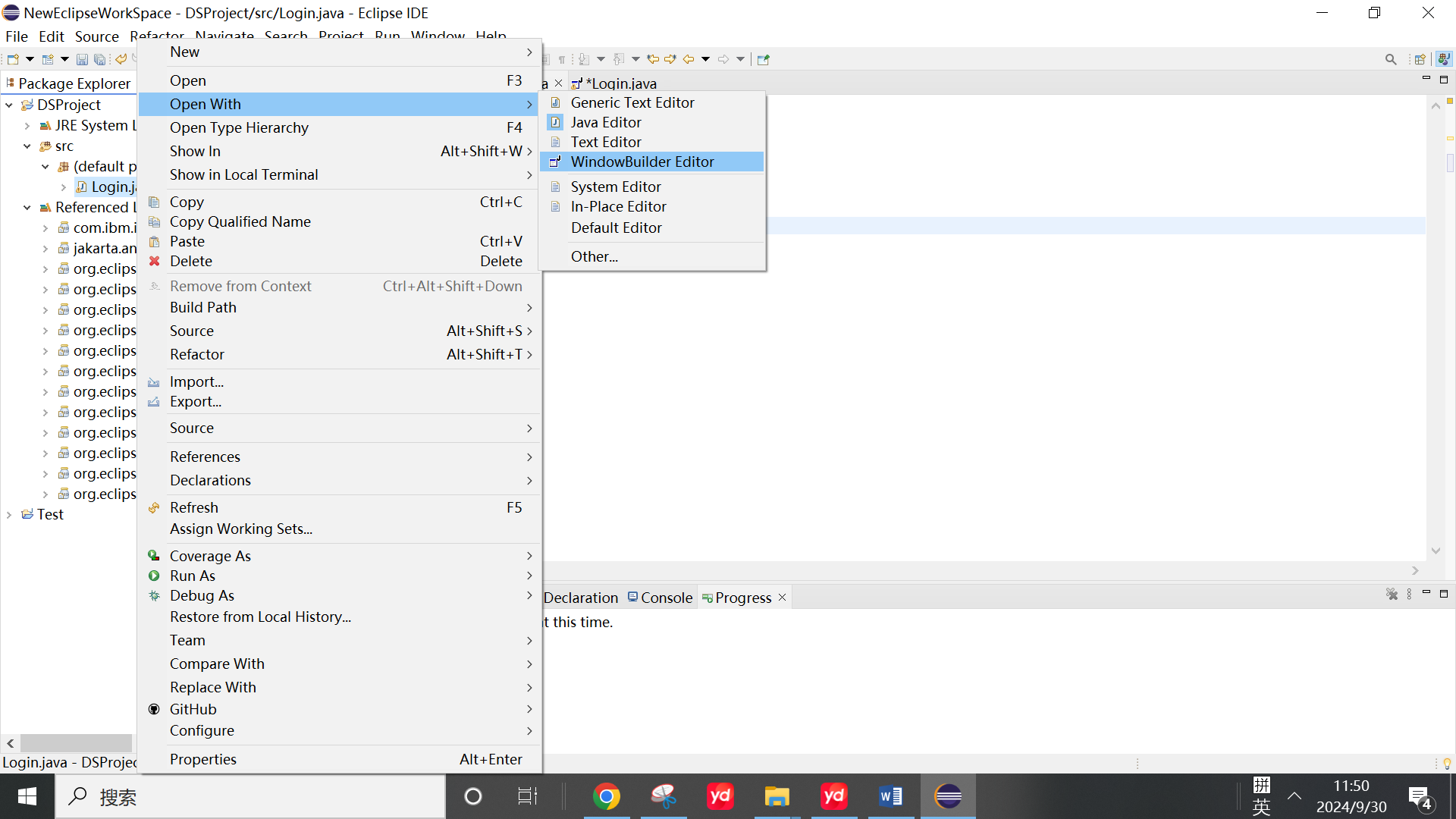
在弹出的对话框中，选择WindowBuilder下的Swing Designer/SWT Designer，创建SWT/JFace Java Project项目



填写Project信息即可：



1. 在创建的项目中创建一个类，该类必须继承自JFrame，然后使用Open With->WindowBuilder Editor，就可以打开可视化窗口



1. 可视化窗口如下所示：

