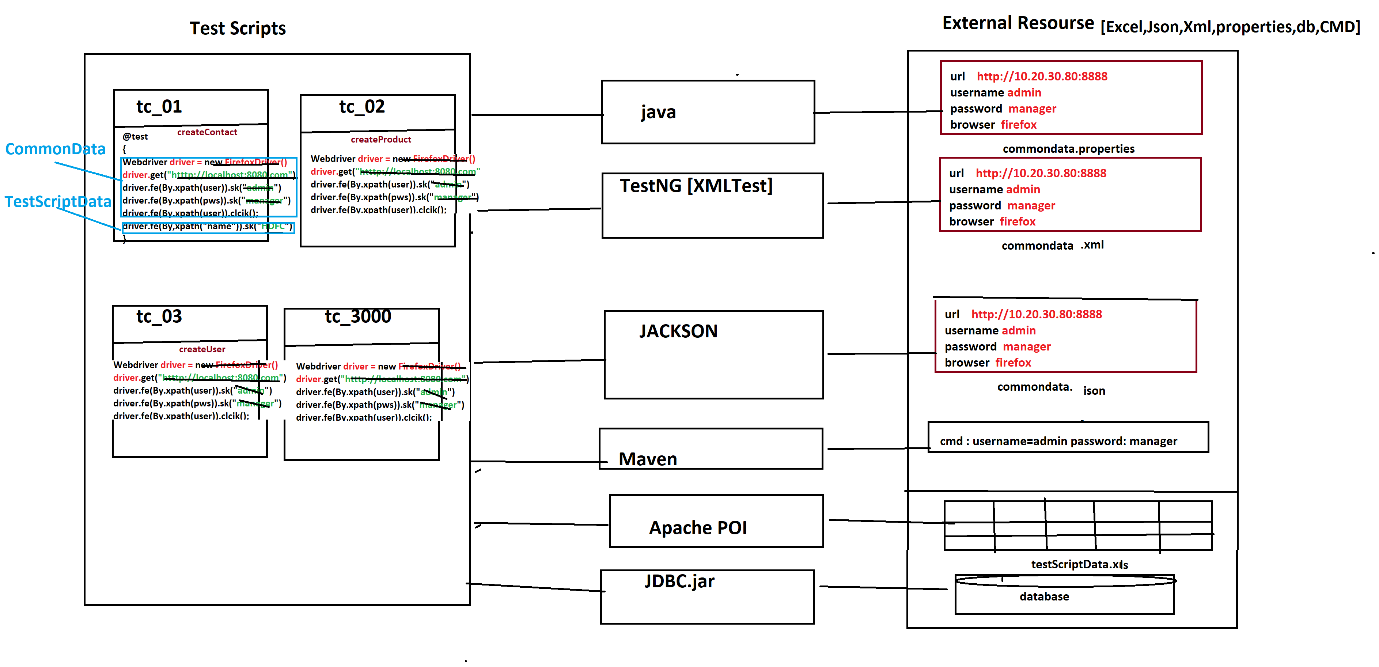
**Data Driven testing**

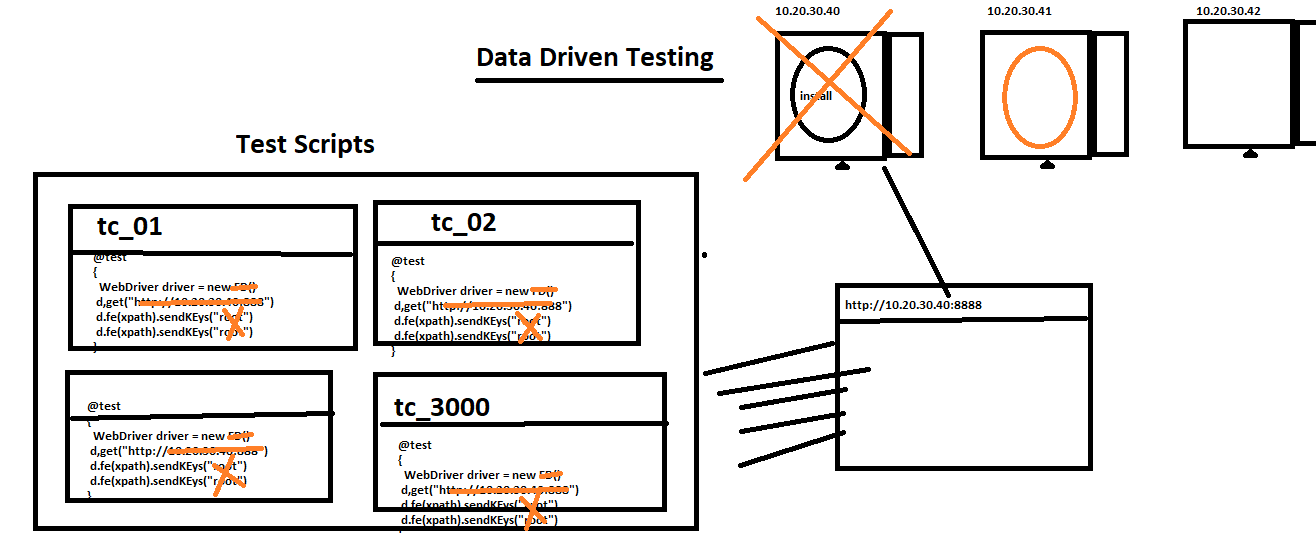
1. **What is Data driven testing?**

Read the data from external recourse & run the test is called Data driven testing (parameterization)



1. Why Data Driven testing?

As per the rule of the automation data shouldn’t not hardcoded(fixed) with in a test scripts, because data modification & maintenance is tedious job when you want to run the test with different data, instead we should get the data from external resource like xlsx, .properties file , db , XML, JSON, CMD Line Data



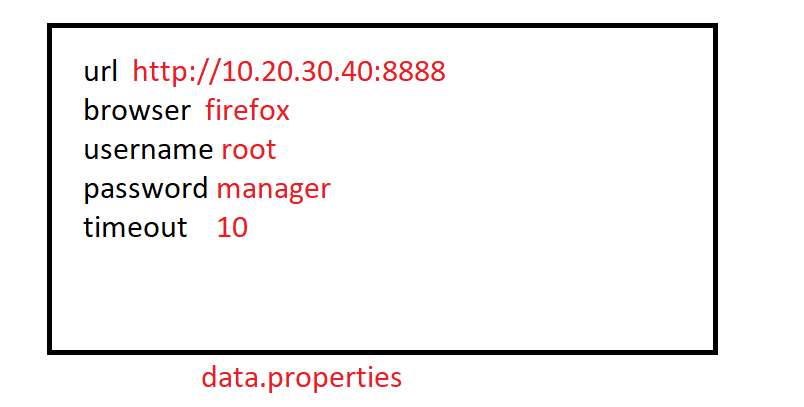
1. What is Advantages of Data driven testing
2. Maintenance of the test data is easy
3. Modification of the test data in external recourse is easy
4. Cross browser /platform testing is easy (means change the browser in property File)
5. Running test scripts in different Environment is easy
6. Running test scripts in different credentials is easy
7. We can create the test data prior the Suite execution (we can also get the data from testData team)
8. Rerunning same test Script multiple time with different data is easy.

**Data driven testing from Properties File**

1. **What is Properties File?**

Properties is java feature file where we can store the data in from of key & values pair,

Key & value data type should be always string.



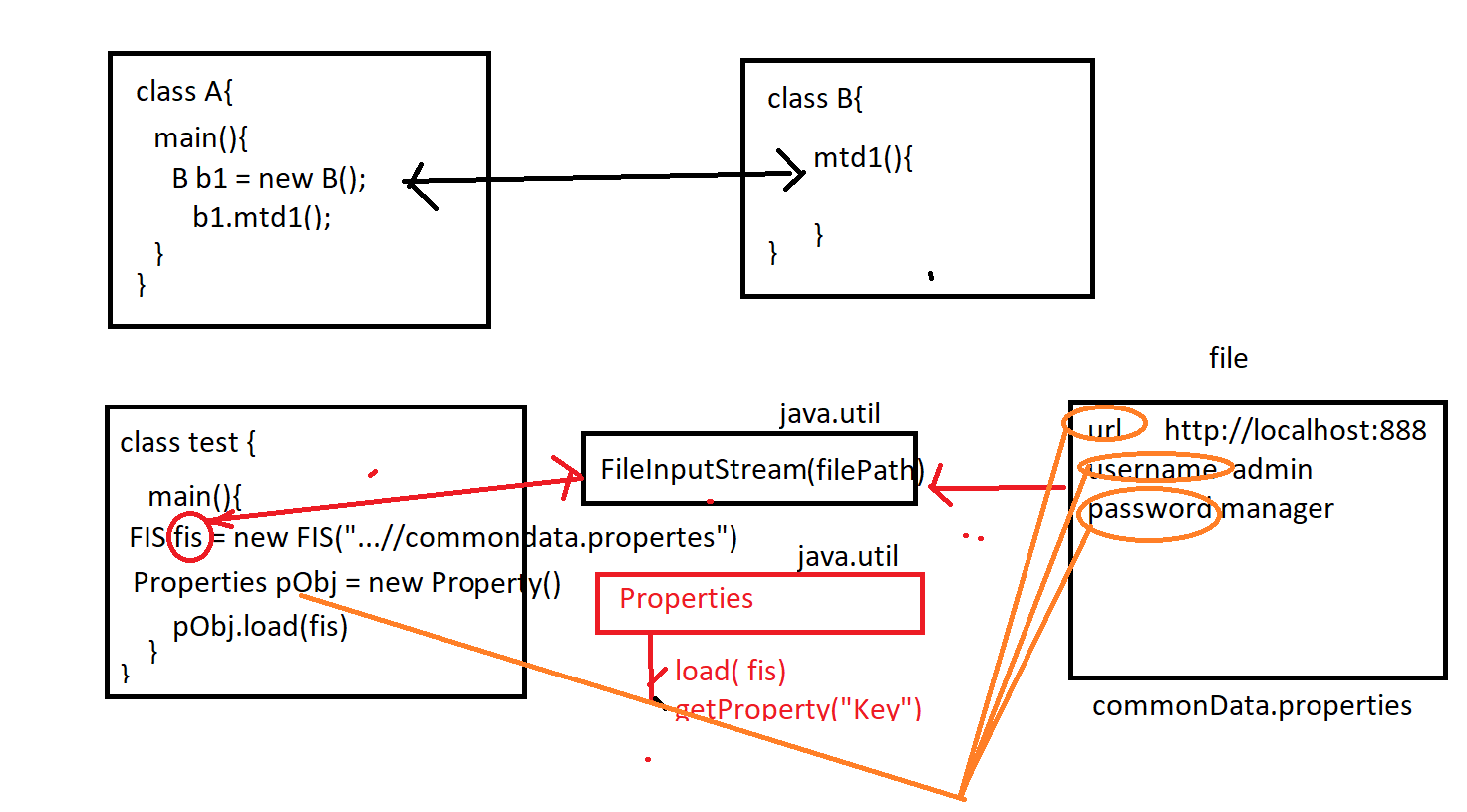
1. **Why Property file?**

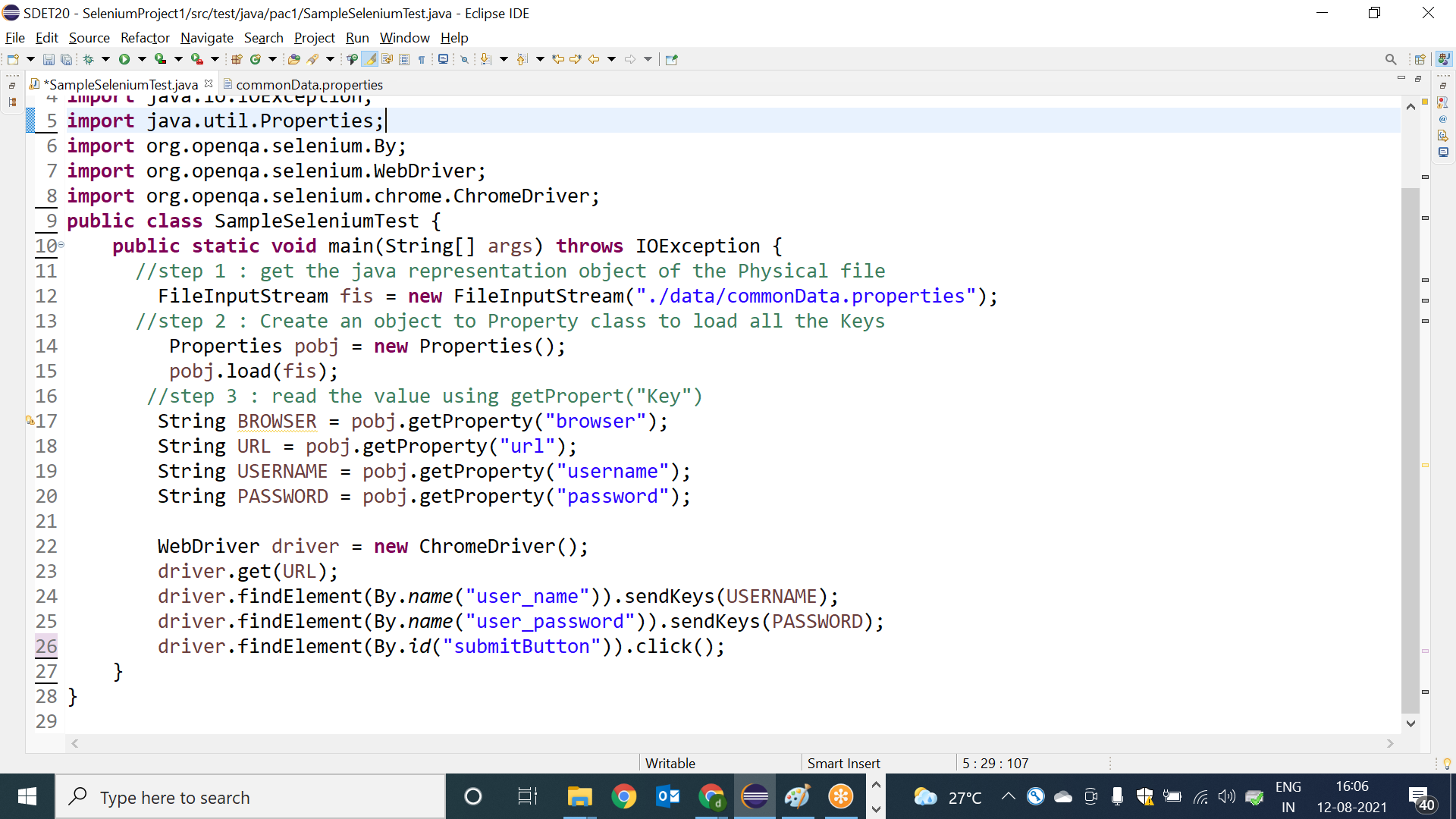
Property file is light weight & faster to read the data compare to any other file, & java as own libraries to read the data

1. **How to read data from properties File?**

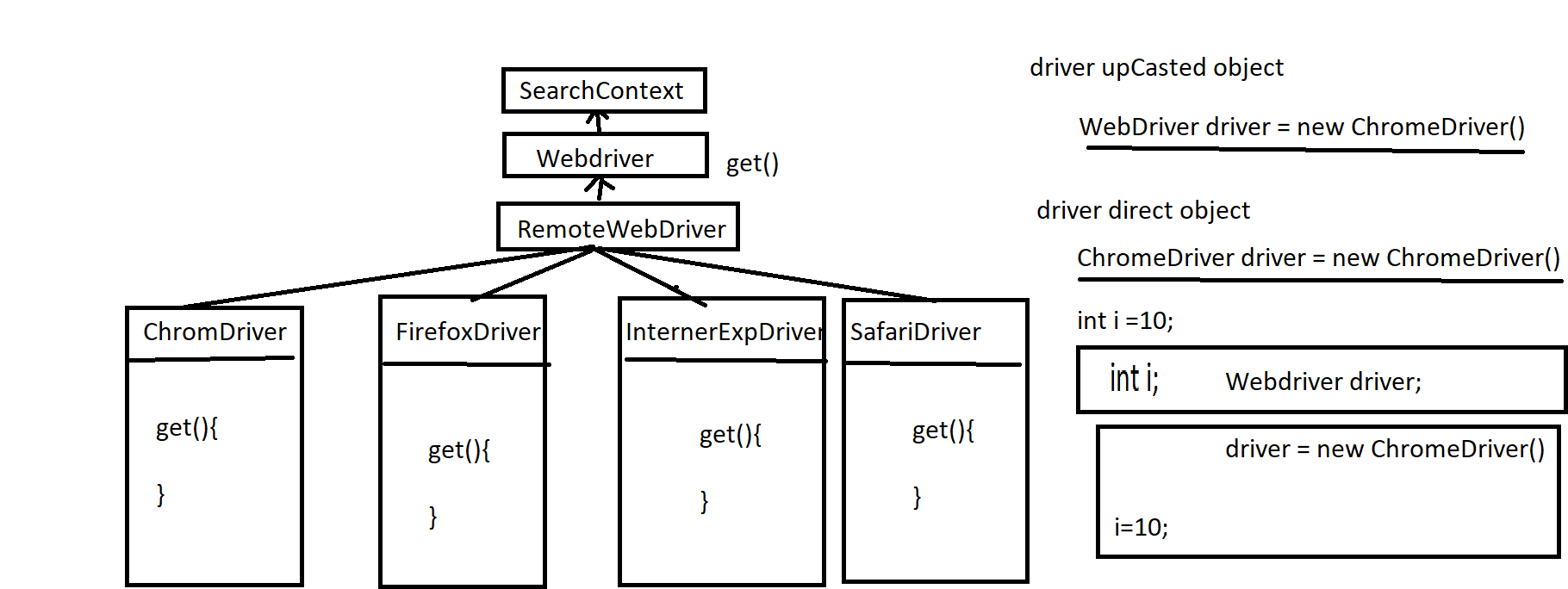
* Get the java representation Object of the Physical file using “FileInputSteam
* Create a Object of “Properties” class & load all the keys
* Read the data using getProperty(“Key”)

Note : properties file light weight & faster in execution compare to Excel





**How to use Browser data in Seleniumtest**



**public** **class** SampleSeleniumTest {

**public** **static** **void** main(String[] args) **throws** IOException {

//step 1 : get the java representation object of the Physical file

FileInputStream fis = **new** FileInputStream("./data/commonData.properties");

//step 2 : Create an object to Property class to load all the Keys

Properties pobj = **new** Properties();

pobj.load(fis);

//step 3 : read the value using getPropert("Key")

String BROWSER = pobj.getProperty("browser");

String URL = pobj.getProperty("url");

String USERNAME = pobj.getProperty("username");

String PASSWORD = pobj.getProperty("password");

WebDriver driver =**null**;

**if**(BROWSER.equals("chrome"))

{

driver = **new** ChromeDriver();

}**else** **if**(BROWSER.equals("firefox")) {

driver = **new** FirefoxDriver();

}**else** **if**(BROWSER.equals("ie")) {

driver = **new** InternetExplorerDriver();

}

driver.get(URL);

driver.findElement(By.*name*("user\_name")).sendKeys(USERNAME);

driver.findElement(By.*name*("user\_password")).sendKeys(PASSWORD);

driver.findElement(By.*id*("submitButton")).click();

}

}

EG : best example for run time Polymorphism , driver object behave differently in run time

Assignment:

Test Case : CreateOrganization

step 1 : login

step 2 : navigate to Organization module

step 3 : click on "create Organization" Button

step 4 : enter all the details & create new Organization

step 5 : verify Organization name in header of the msg

step 6 : logout

Note :

1. Any data should not be hardcoded
2. test should able to run in different browser with minimal changes
3. test script data should get it from EXCEL sheet
4. create a Library for read the data for Excel & Properties File

**Data driven testing from Excel File**

🡺Apache Poi is the open source libraries used to get & write data from all Microsoft documents like Excel , docx , ppt etc

🡺In real time most the company preferred the keep the test script data in Excel, because data will be in well-organized manner , so that modification & maintenance is easier.

Installation steps:

1. Go the Maven Project
2. Edit POM.xml file
3. Go to <https://mavenrepository.com>
4. Search for apache-POI
5. Copy the dependency
6. Add dependency inside the <dependencies> in POM.xml

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>4.0.0</version>

</dependency>

<dependency>

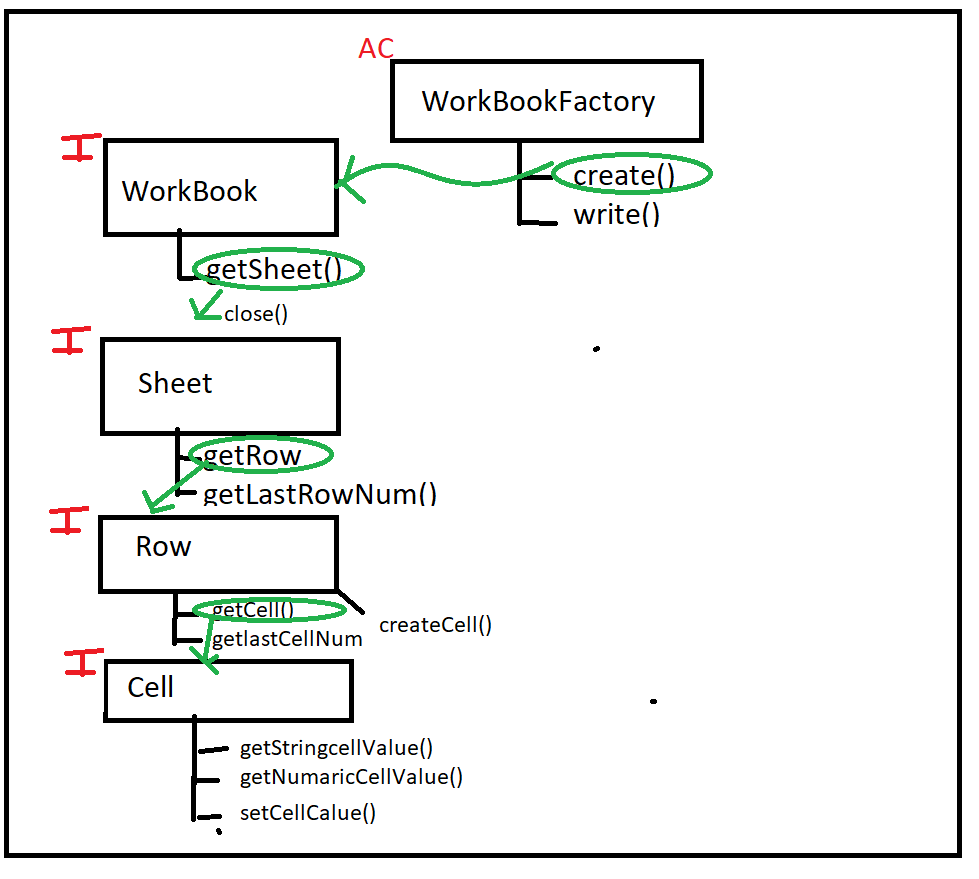
<groupId>org.apache.poi</groupId>

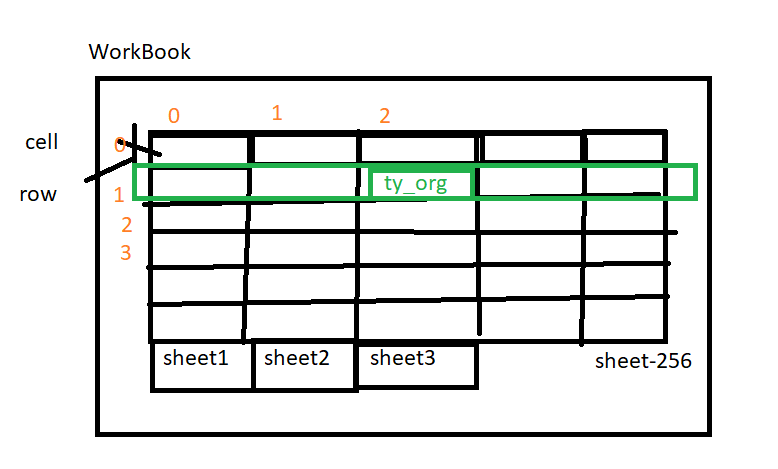
<artifactId>poi-ooxml</artifactId>

<version>4.0.0</version>

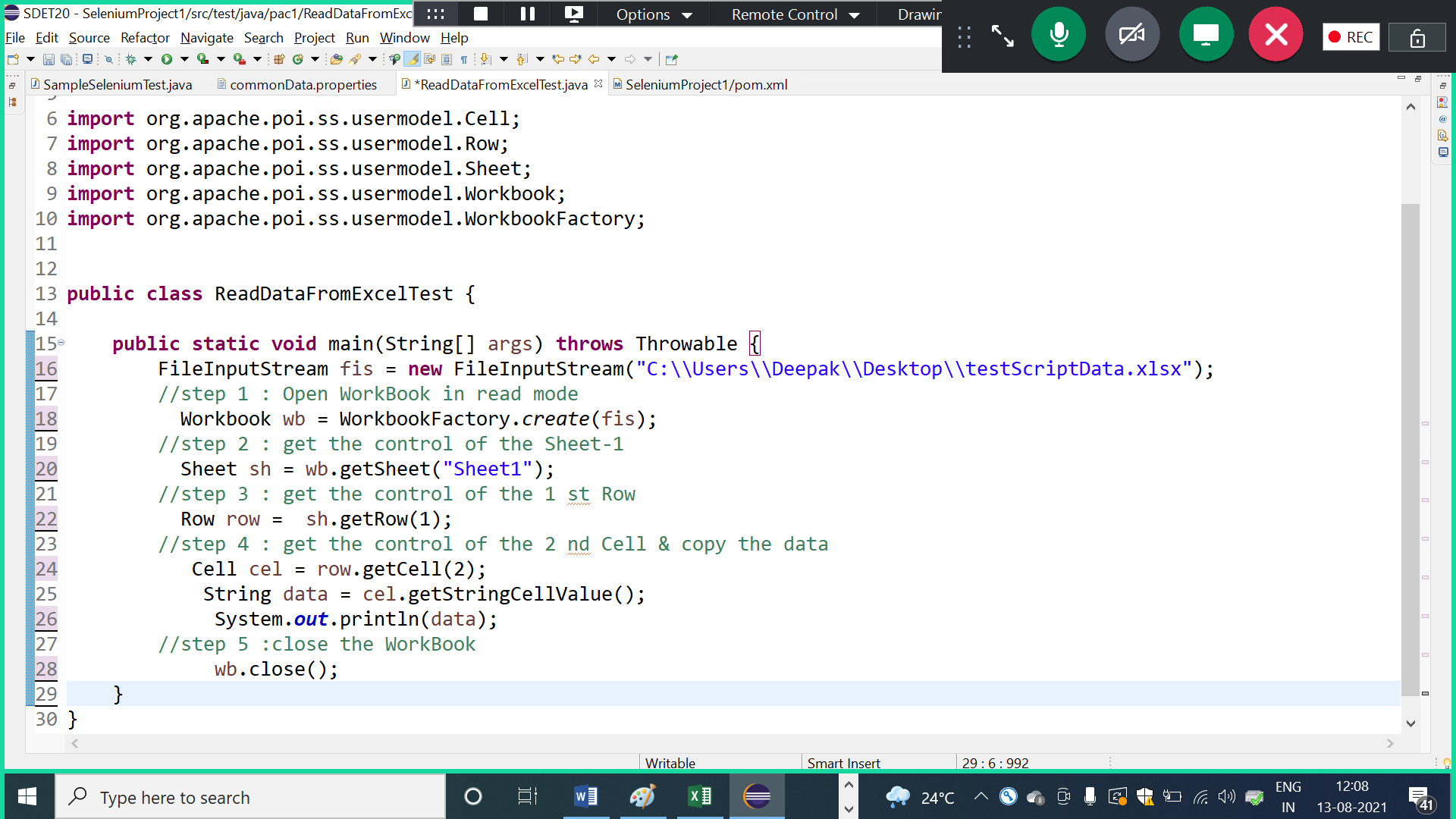
</dependency>

Class diagram of Apache POI

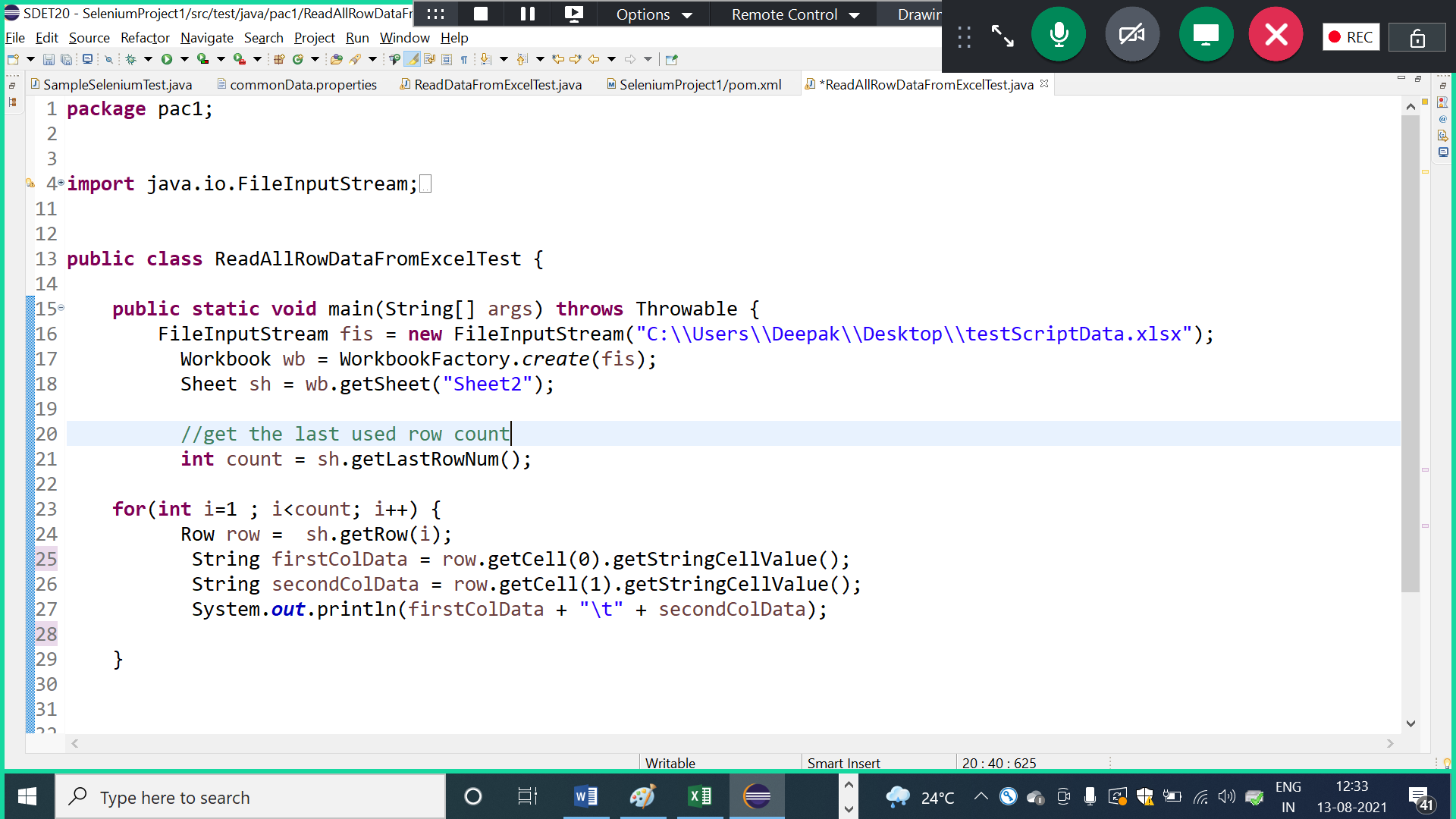




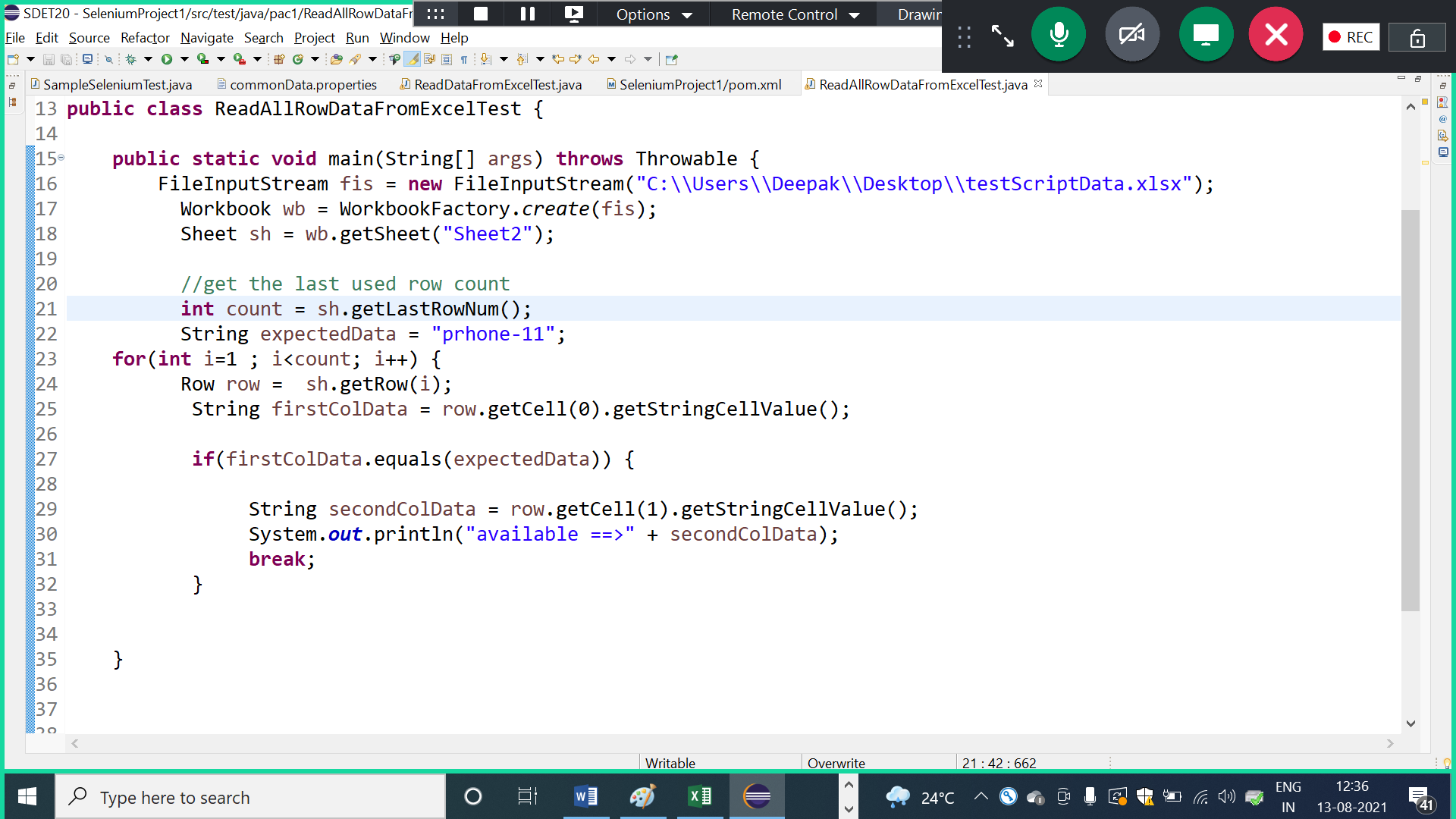
1. Program to read the data from WorkBook



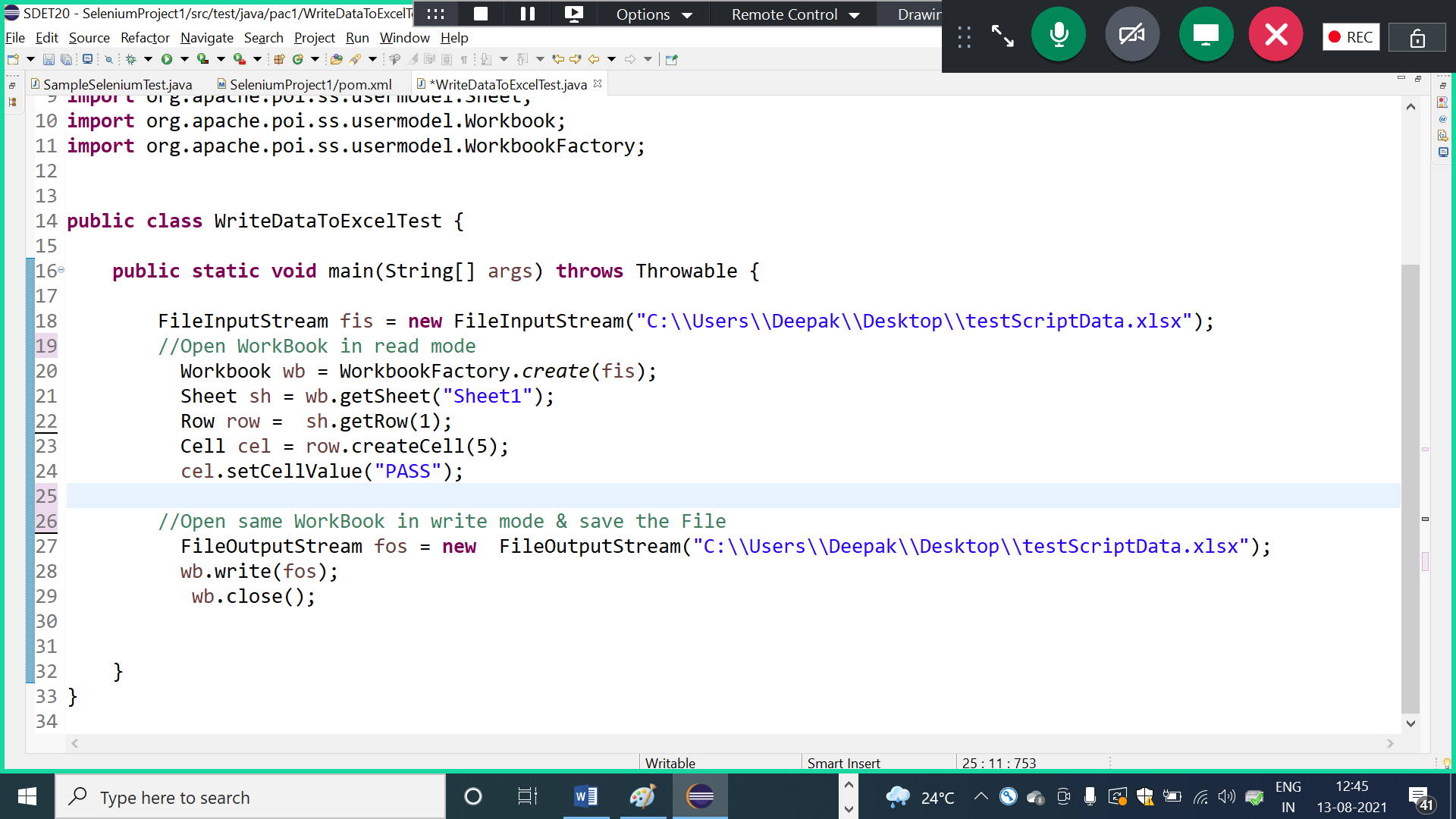
1. Program to read 1 st & 2 nd coloum data from all the Rows

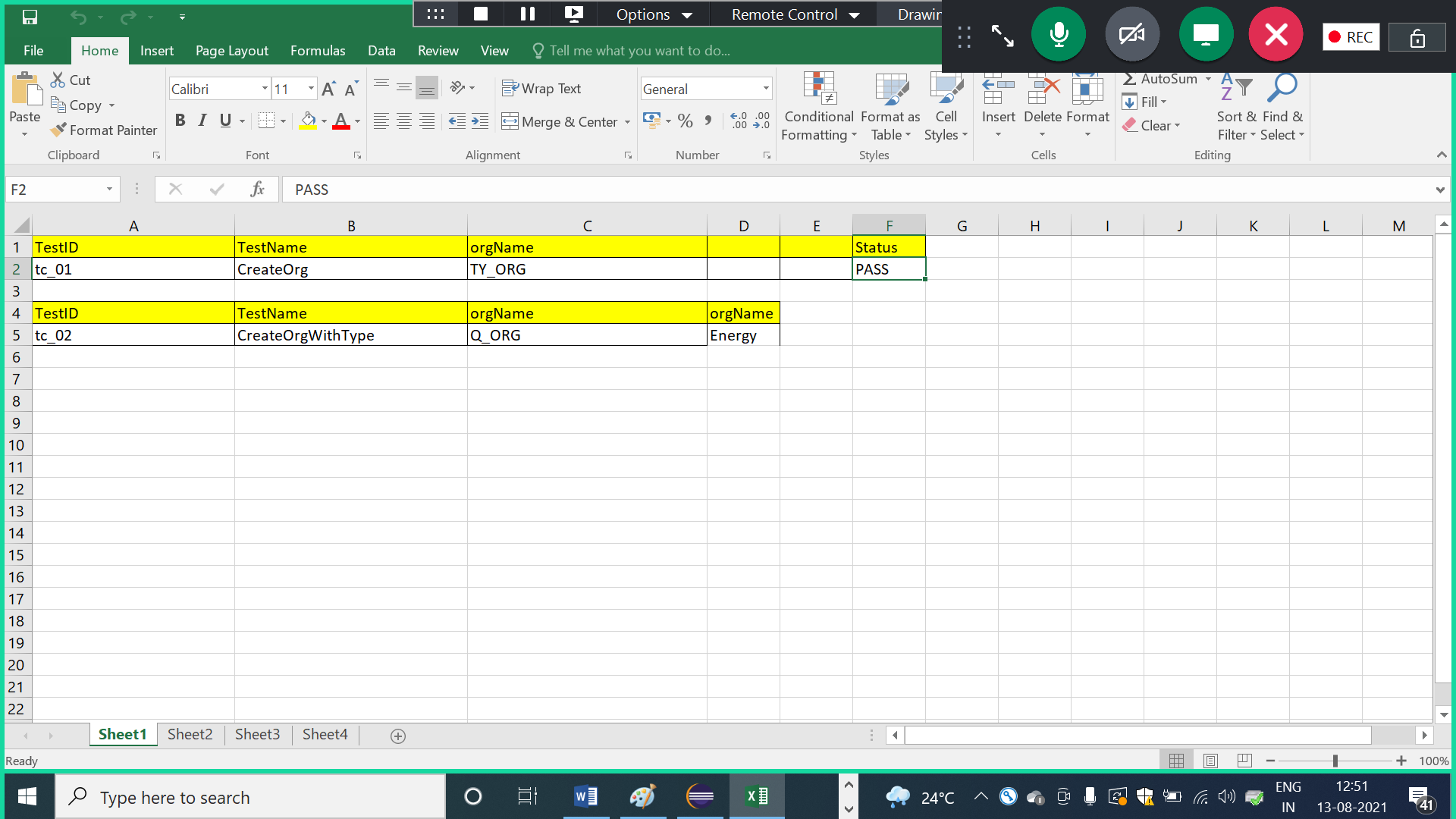


1. Write a program to find specific data from 1 st column , if data is available then read & dispaly next cell data



1. Program to write data back to WorkBook/ Excel





Assignment: 2

Test Case : CreateOrganization

step 1 : login

step 2 : navigate to Organization module

step 3 : click on "create Organization" Button

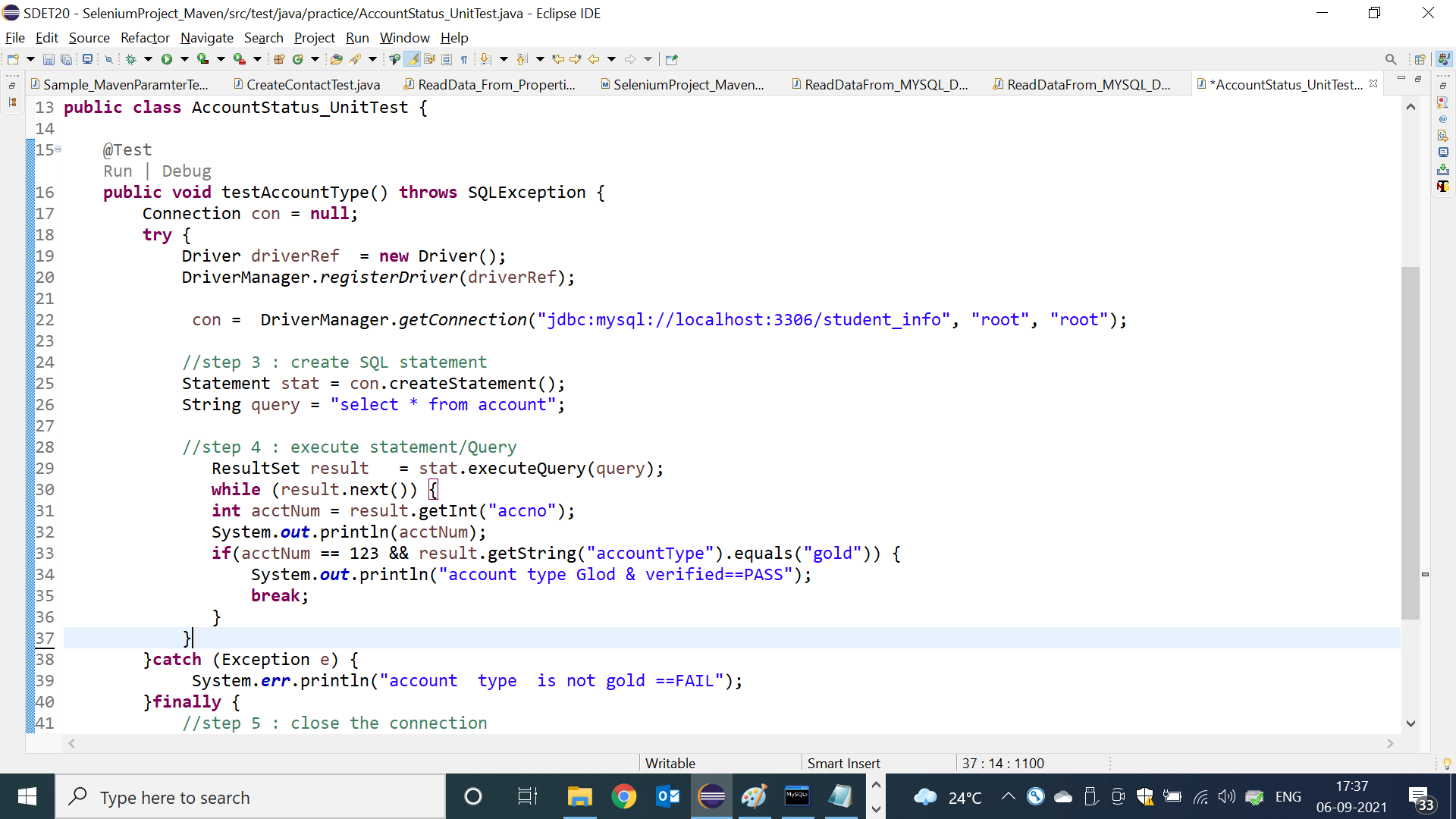
step 4 : enter all the details & create new Organization

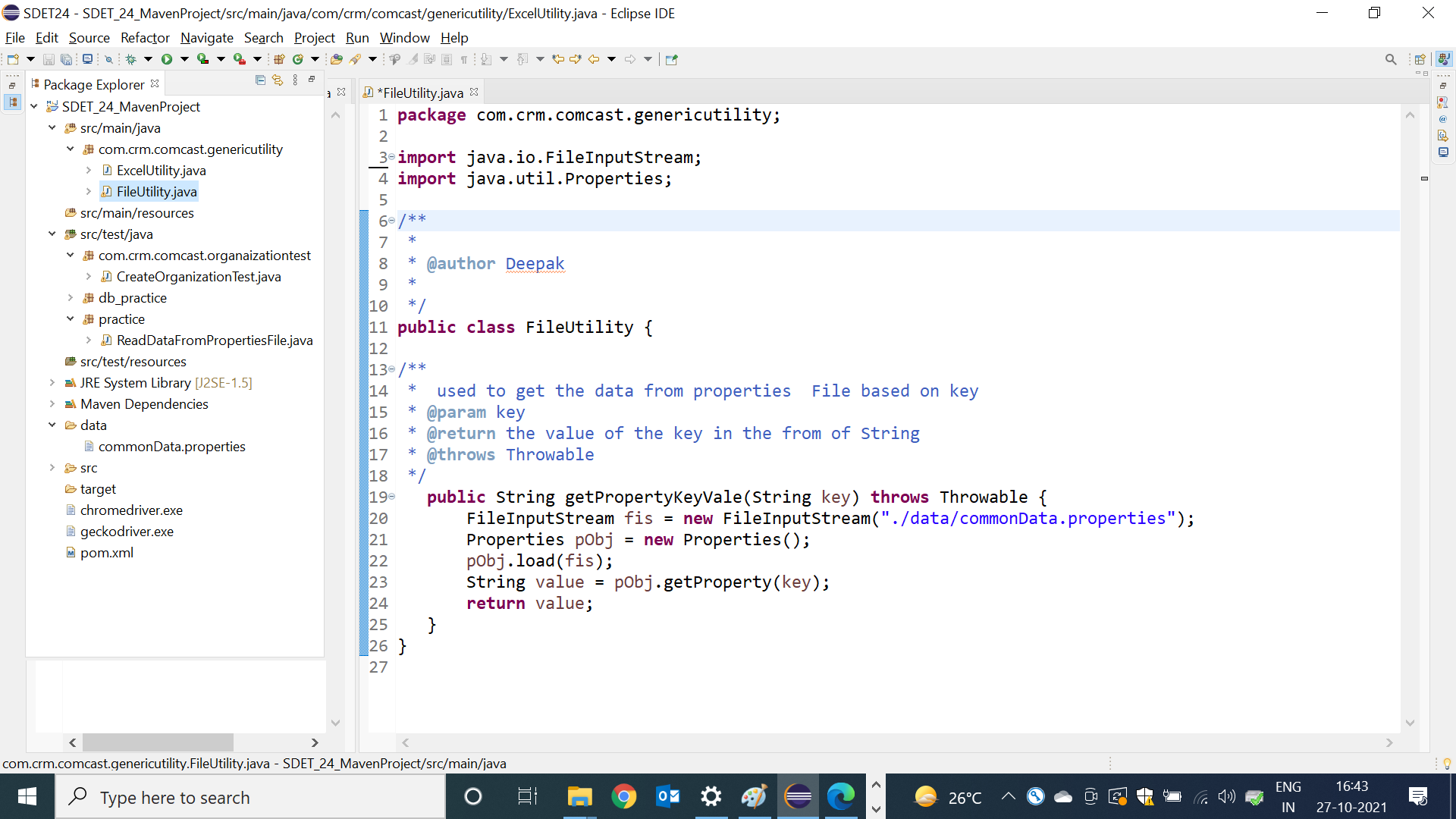
step 5 : verify Organization name in header of the msg

step 6 : logout

Note :

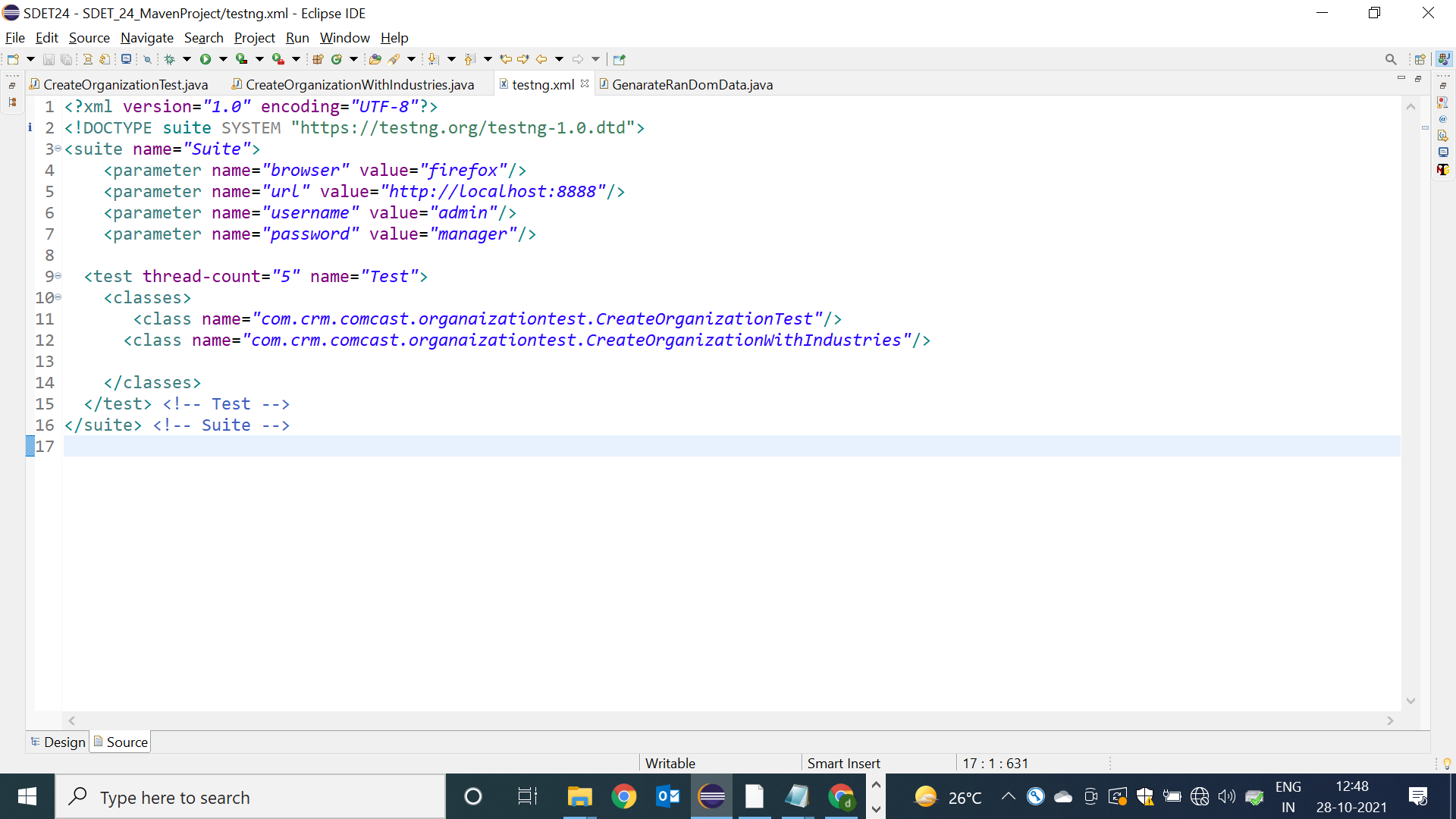
1. Any data should not be hardcoded
2. Common data should come from properties File
3. test should able to run in different browser with minimal changes
4. test scripts data should get from Excel/WorkBook
5. After the execution, write status back to Excel in same Row
6. External File path should be relative

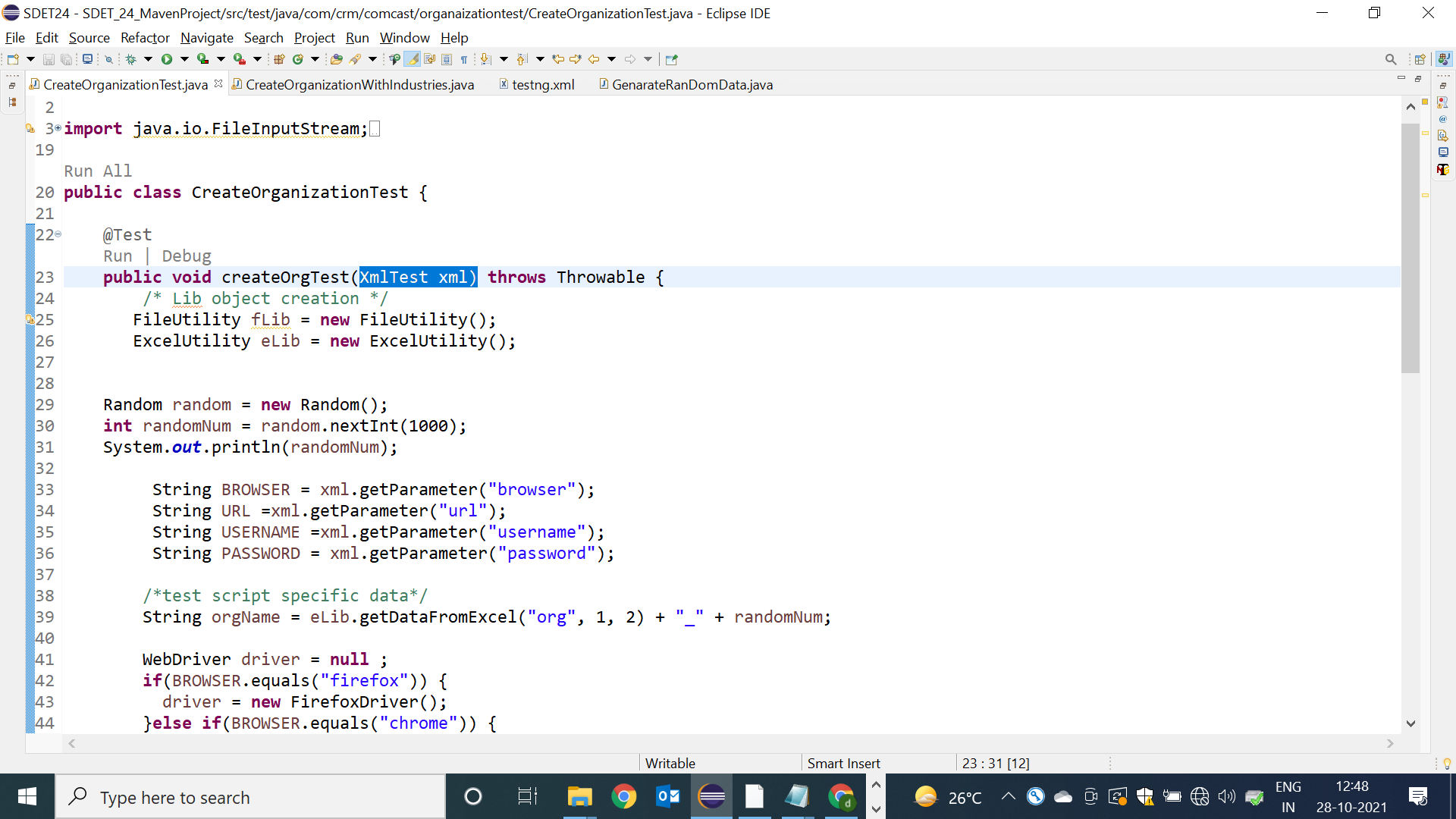




Data driven testing from XML

* Web driver can’t read the data from xml file, in order to achieve this concepts, we have to go for third party tools like testNG & JAXB etc
* To read data from testNG.xml file , we have to use <parameter> tag to declare the key & value pair before<test> and after <suite>, then use XMLTest class with in testing-method to get the data from the XML file

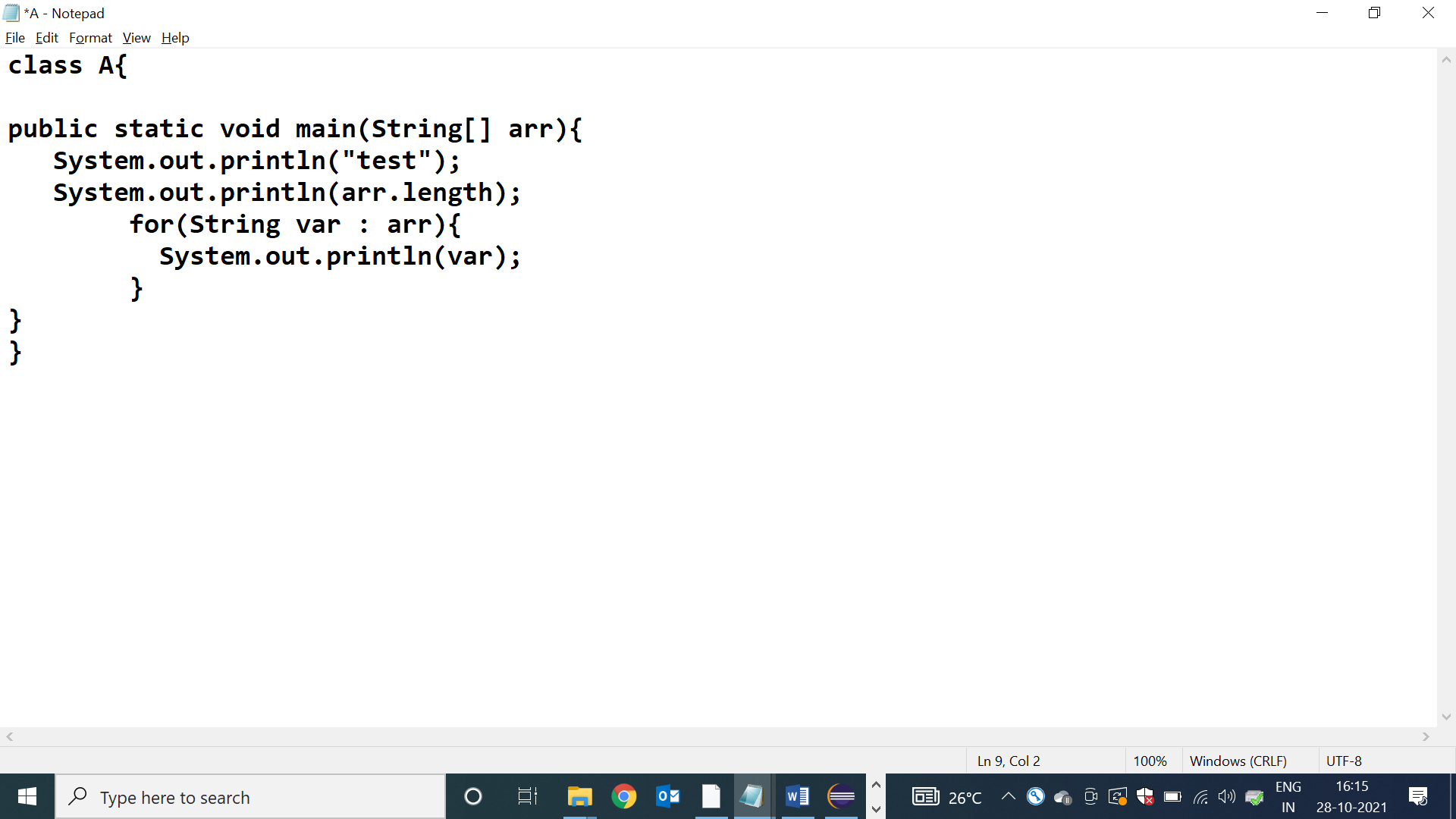




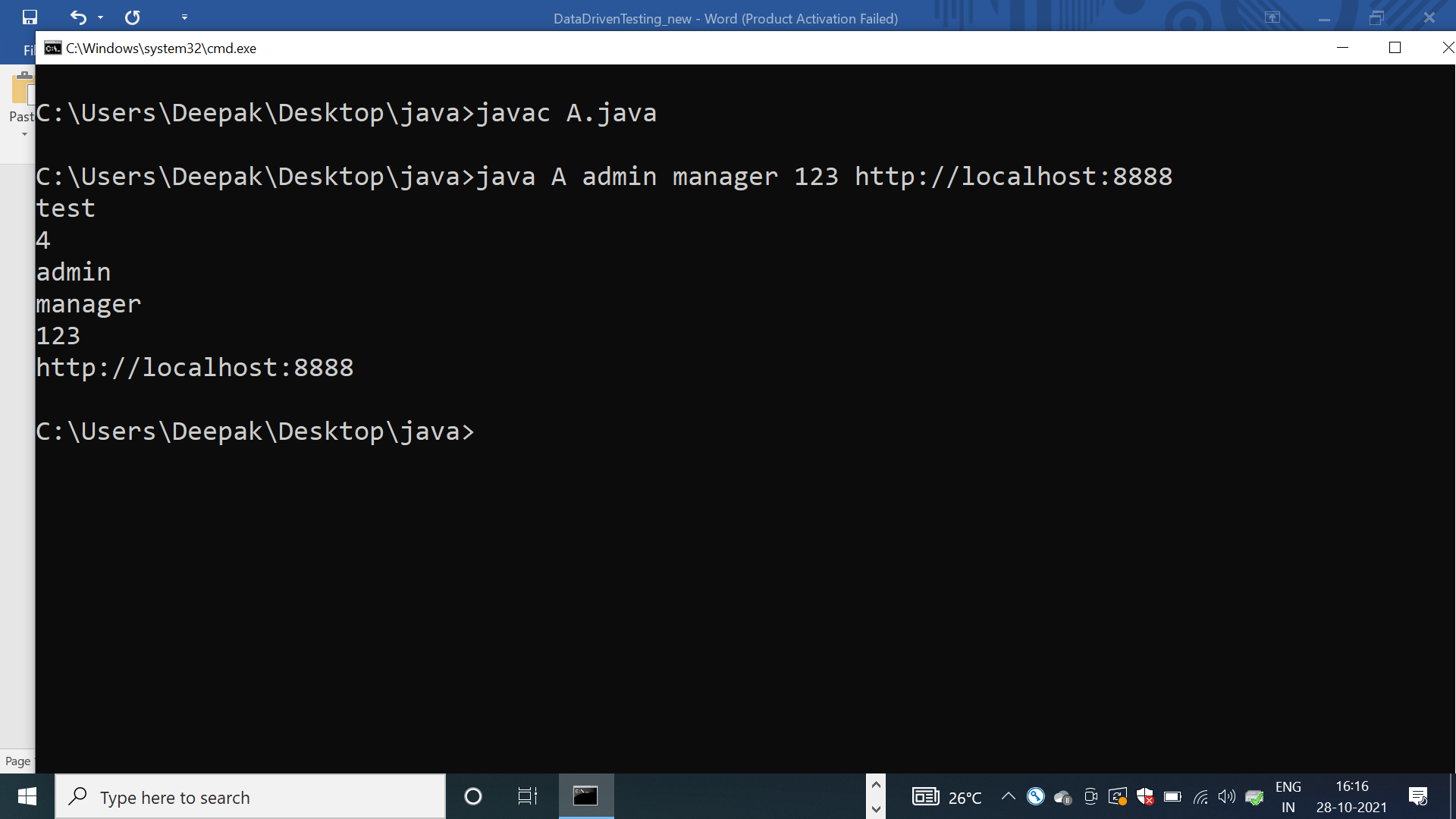
Data driven testing from CMD

**Why java main method arguments is String[] ?**

Java main method arguments is always String[] array because it can receive run time data from cmd



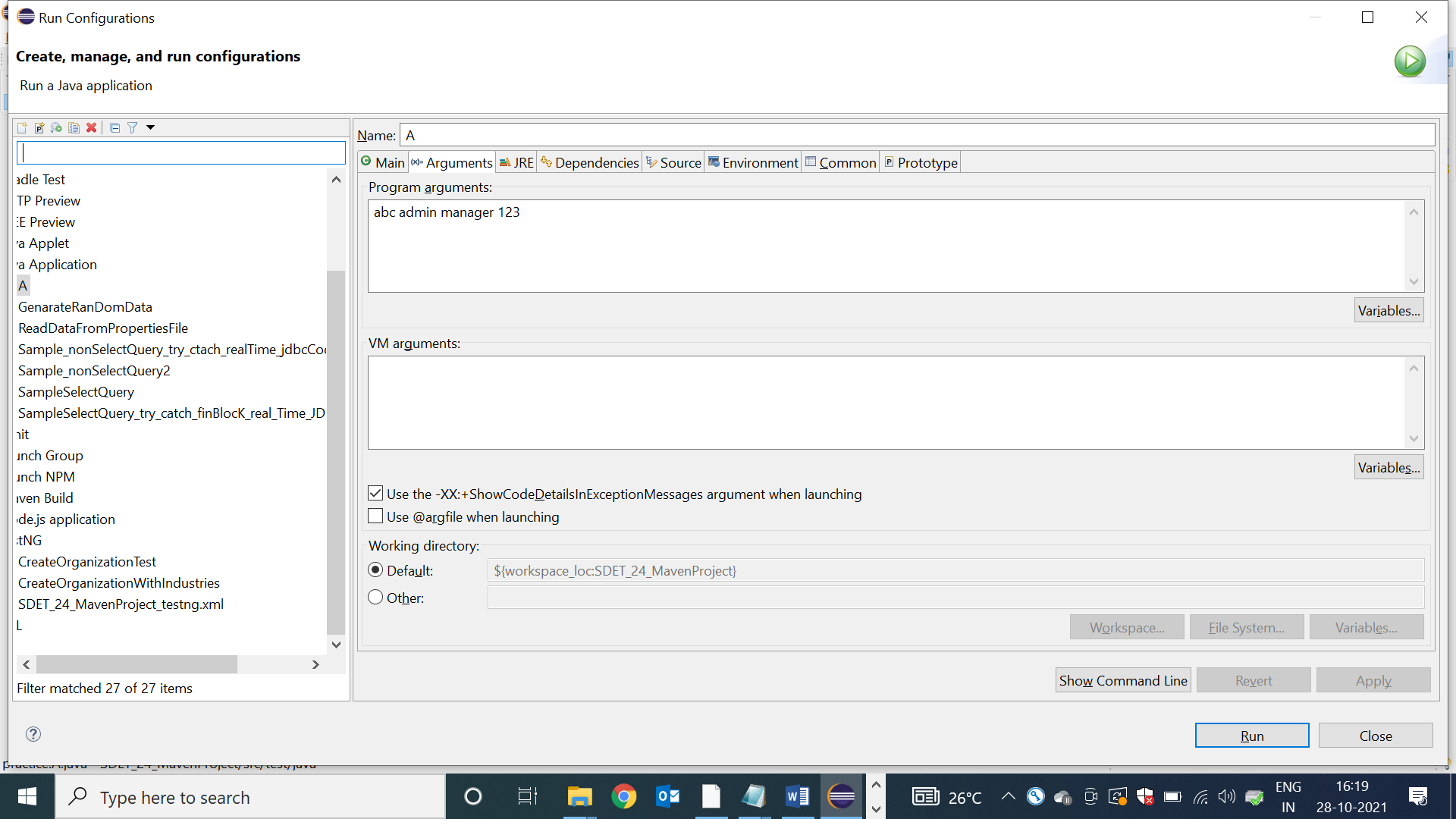
Pass arguments from the CMD



How to pass run time arguments from eclipse IDE ?

Fallow below steps to pass run time arguments

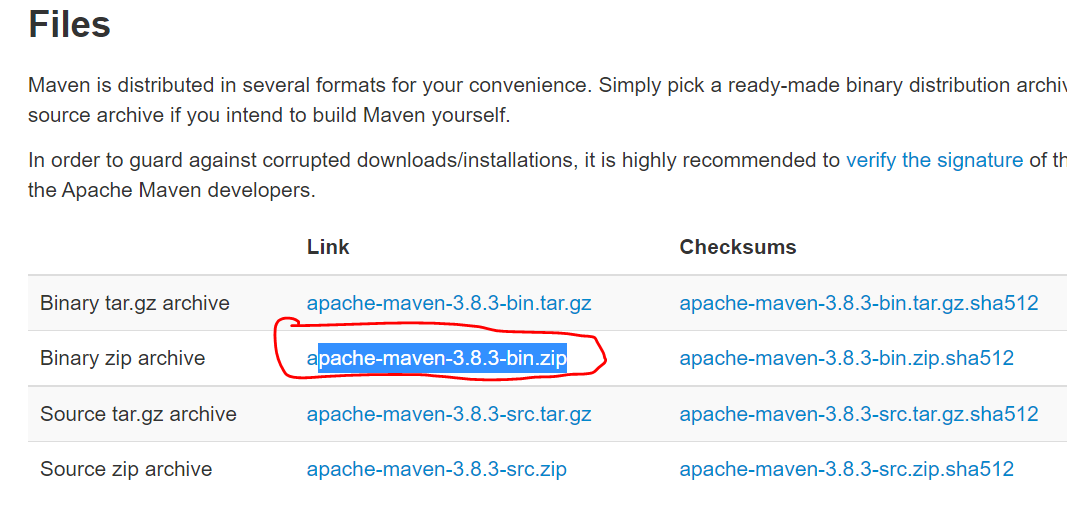
1. Run Java Code once via eclipse IDE
2. Select Java file 🡺 Right Click🡺 Run as🡺run Configuration🡺 click on 🡺 arguments tab



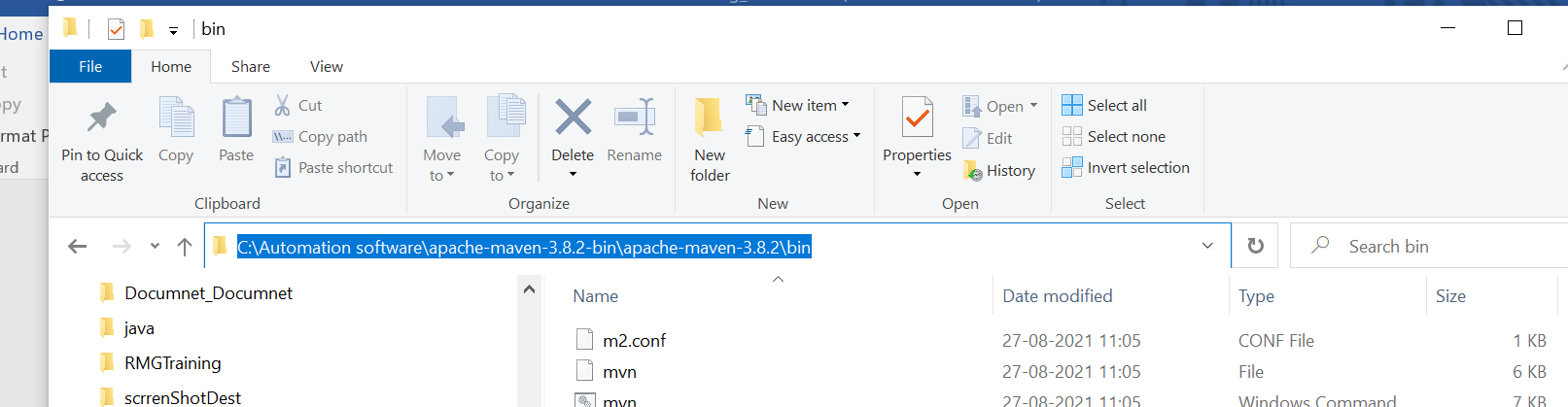
How to pass cmd line arguments to testNG test Scripts ?

Fallow below steps to pass arguments to testNG test

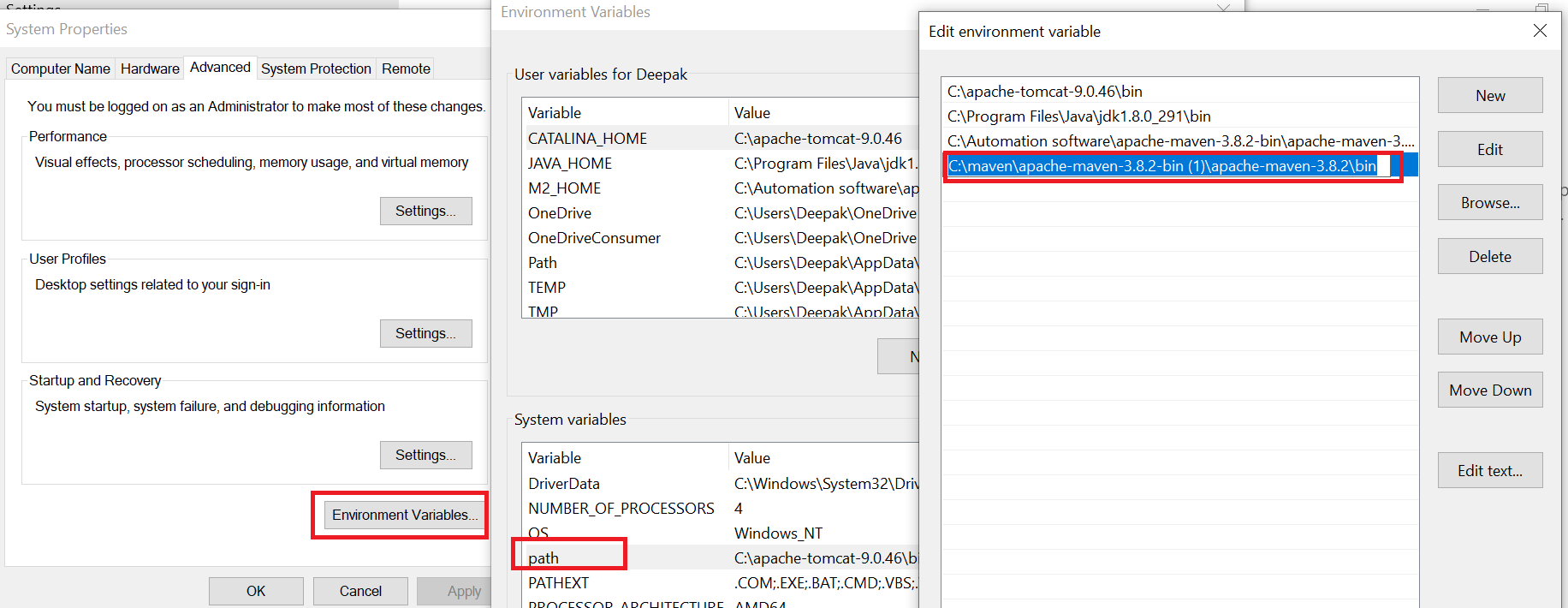
1. **Install maven cmd plugin**
2. Go to google
3. Search for “download maven”
4. click on first link
5. click on below link



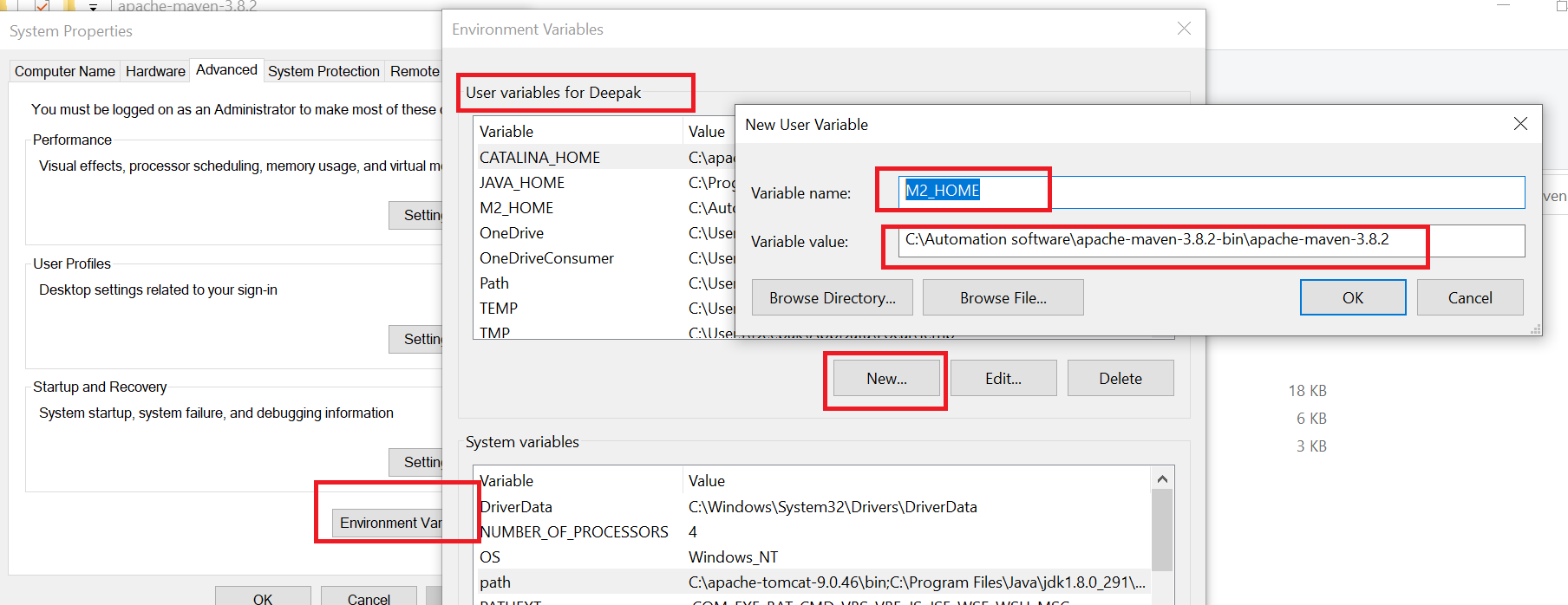
1. EXTRACT THE zip file
2. Get inside the folder and copy the “bin” location



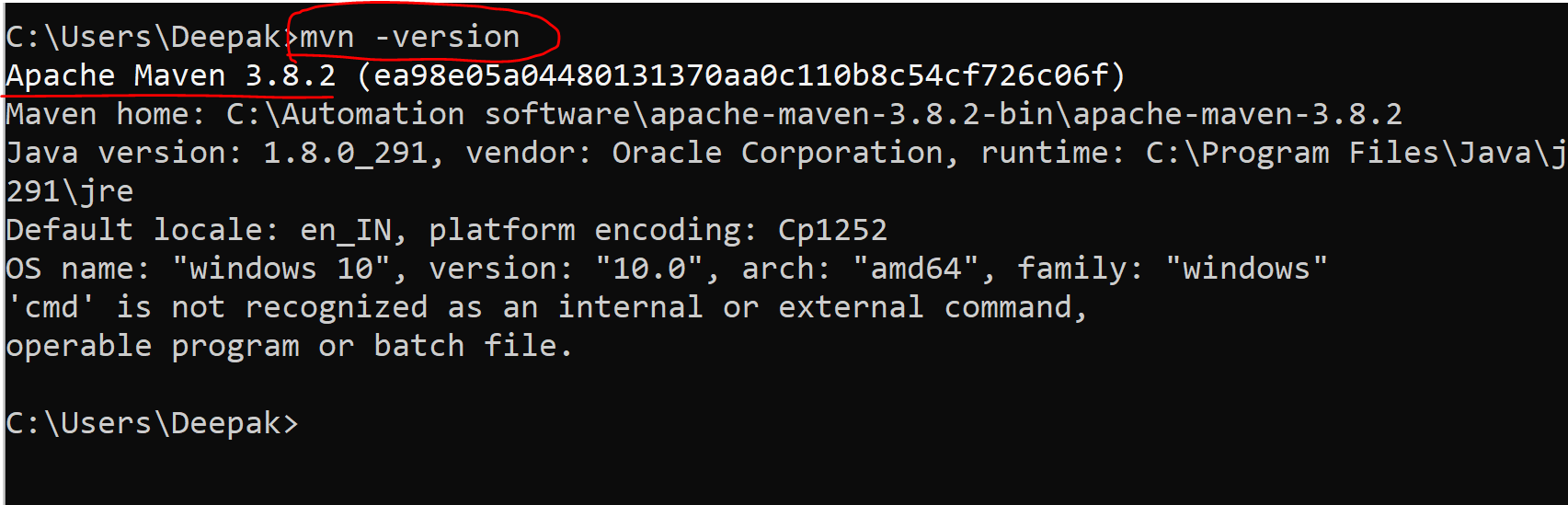
1. In local system go Env variable window & set path variable



8. Create new “M2\_HOME” user variable in Environment variable window

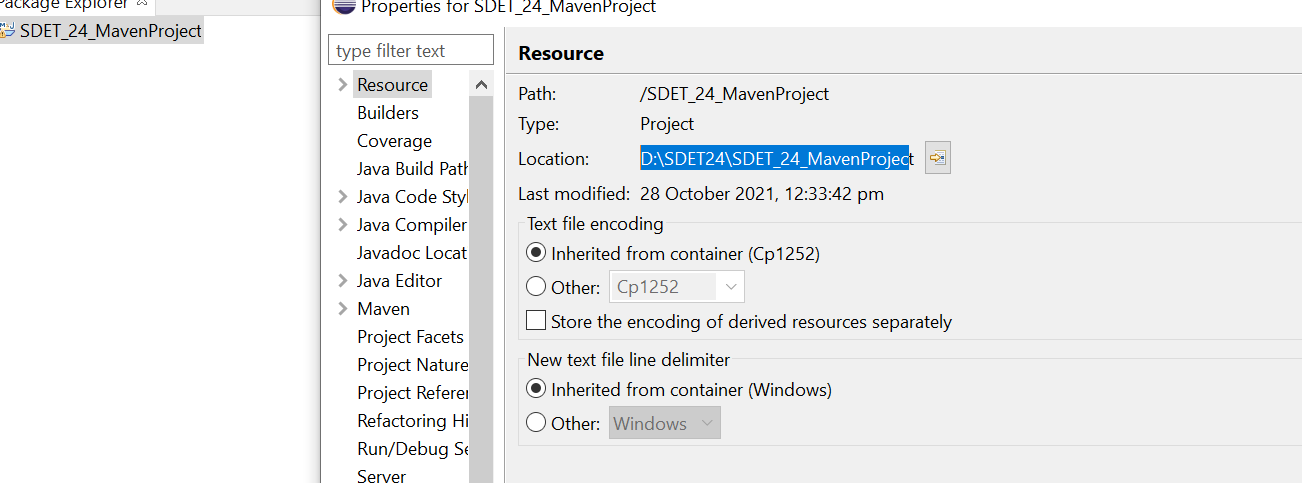
2. **Check maven is installed or not**

Go to CMD and type mvn –version

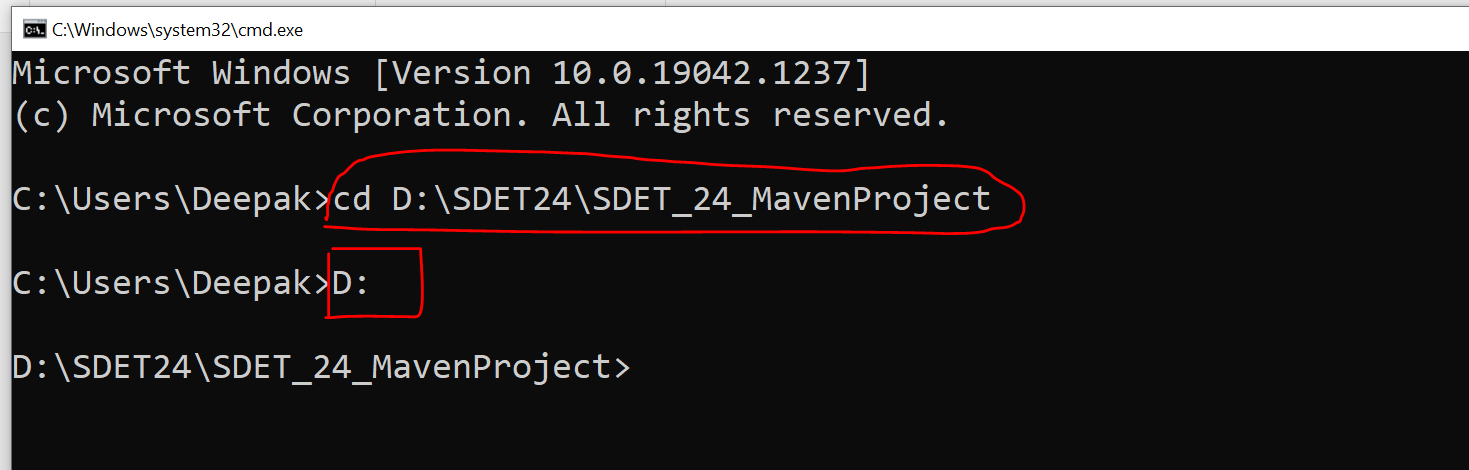


3.copy the location mavnPRoject from the Eclipse IDE

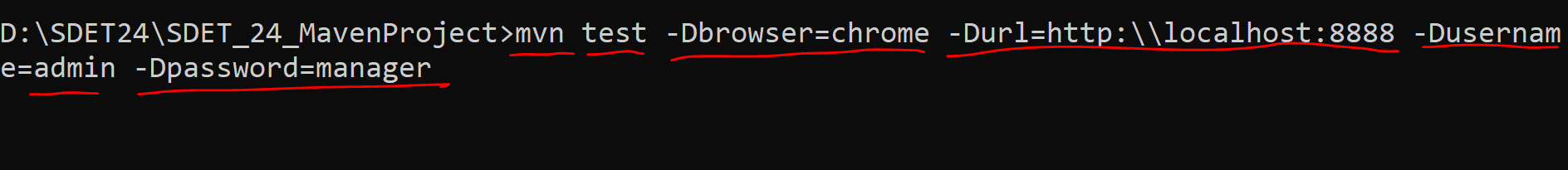
Select ProjectName in Eclipse🡺 rightclick🡺properties 🡺 resourse 🡺 copy the path



4.go to CMD & paste the path

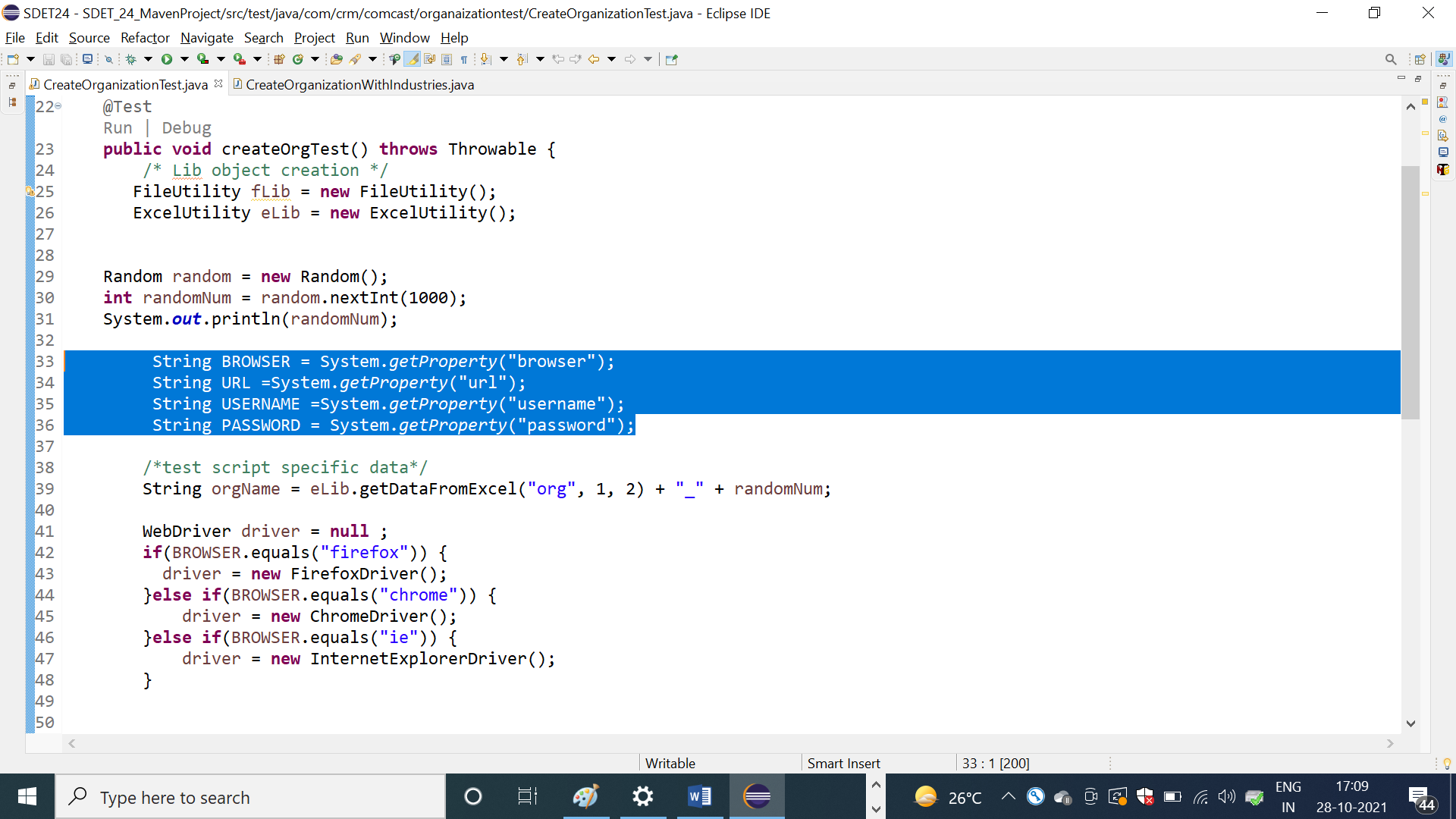


5.pass the parameter from cmd



6.press enter to execute all the testngtest without eclipse IDE

7.to use maven parameter in testNG-class



Note : all the testNG class should end with Test , otherwise test will not be executed by maven via CMD

EG

