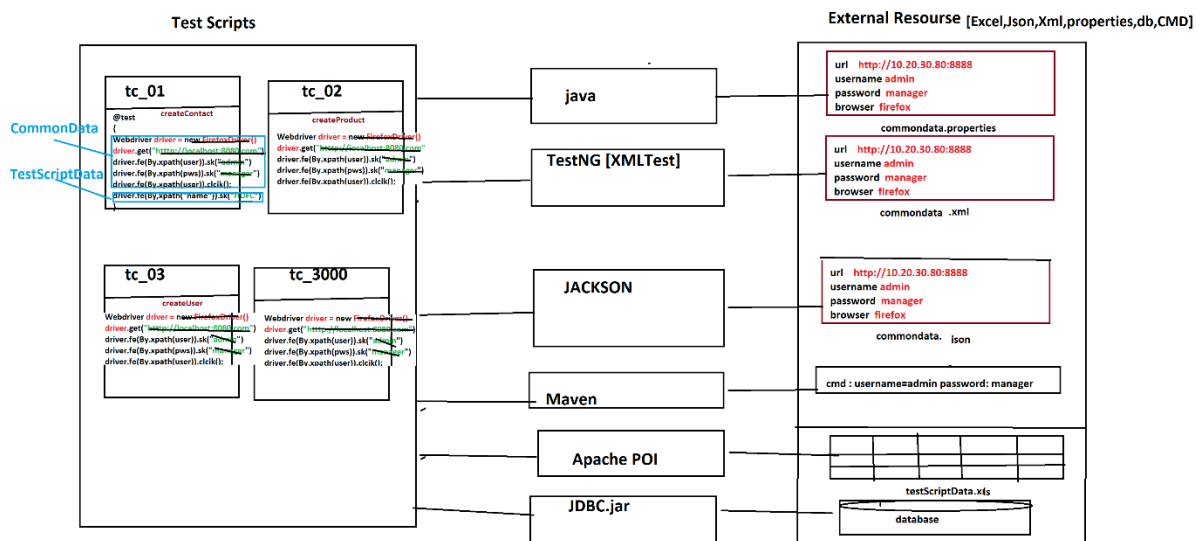


## Data Driven testing

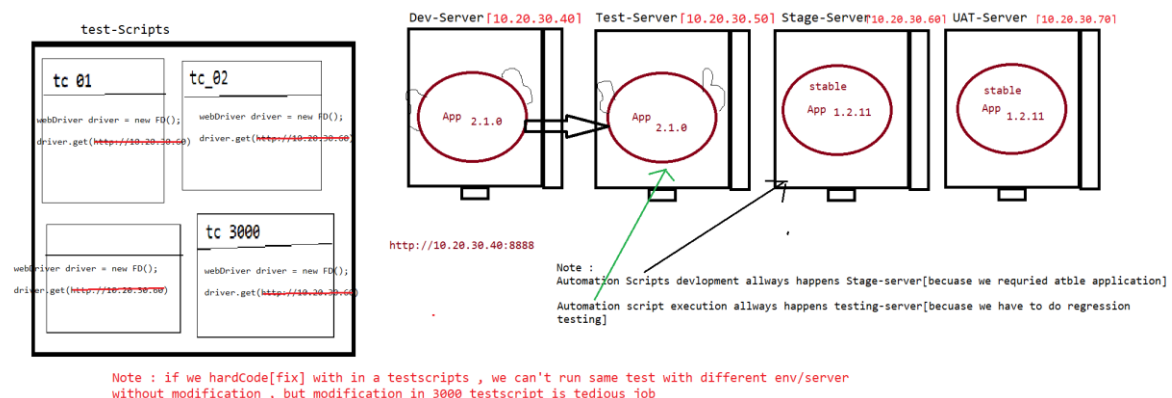
## 1. What is Data driven testing?

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



## 2. 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`



### 3. What is Advantages of Data driven testing

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

```
url http://10.20.30.40:8888
browser firefox
username root
password manager
timeout 10
```

data.properties

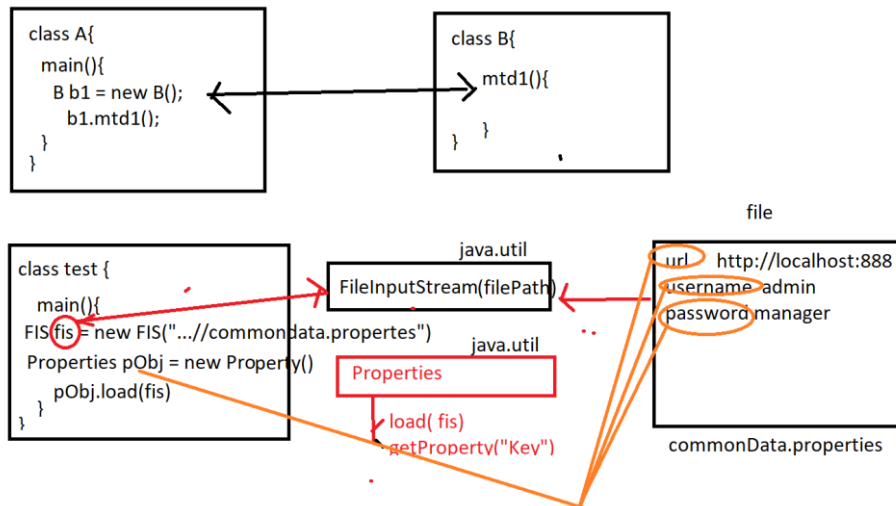
### 2. Why Property file ?

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

### 3. 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

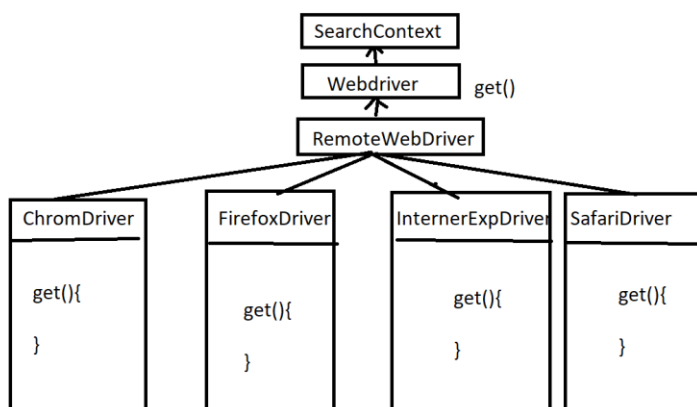


```

SDET20 - SeleniumProject1/src/test/java/pac1/SampleSeleniumTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
SampleSeleniumTest.java commonData.properties
import java.io.IOException;
import java.util.Properties;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
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 getProperty("Key")
        String BROWSER = pObj.getProperty("browser");
        String URL = pObj.getProperty("url");
        String USERNAME = pObj.getProperty("username");
        String PASSWORD = pObj.getProperty("password");

        WebDriver driver = new ChromeDriver();
        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();
    }
}
  
```

## How to use Browser data in Seleniumtest



driver upCasted object

```
WebDriver driver = new ChromeDriver()
```

driver direct object

```
ChromeDriver driver = new ChromeDriver()
```

```
int i =10;
```

```
int i;    Webdriver driver;
```

```
driver = new ChromeDriver()
```

```
i=10;
```

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

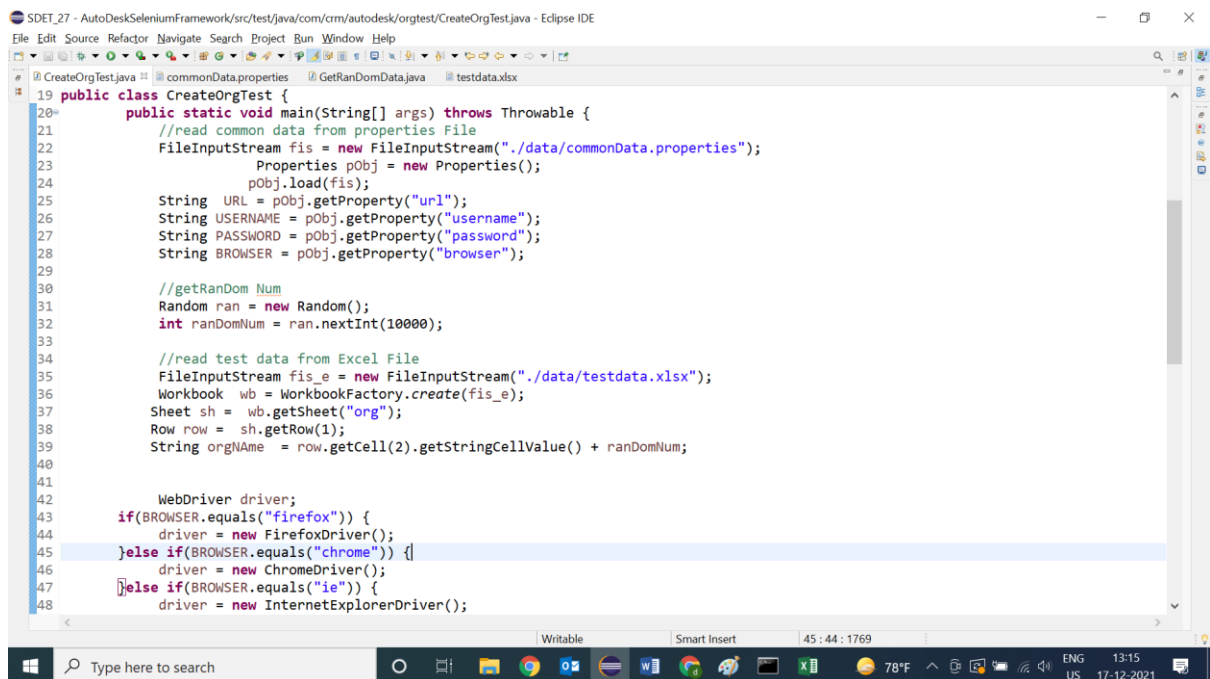
Assignment:

## Test Case : CreateOrganization

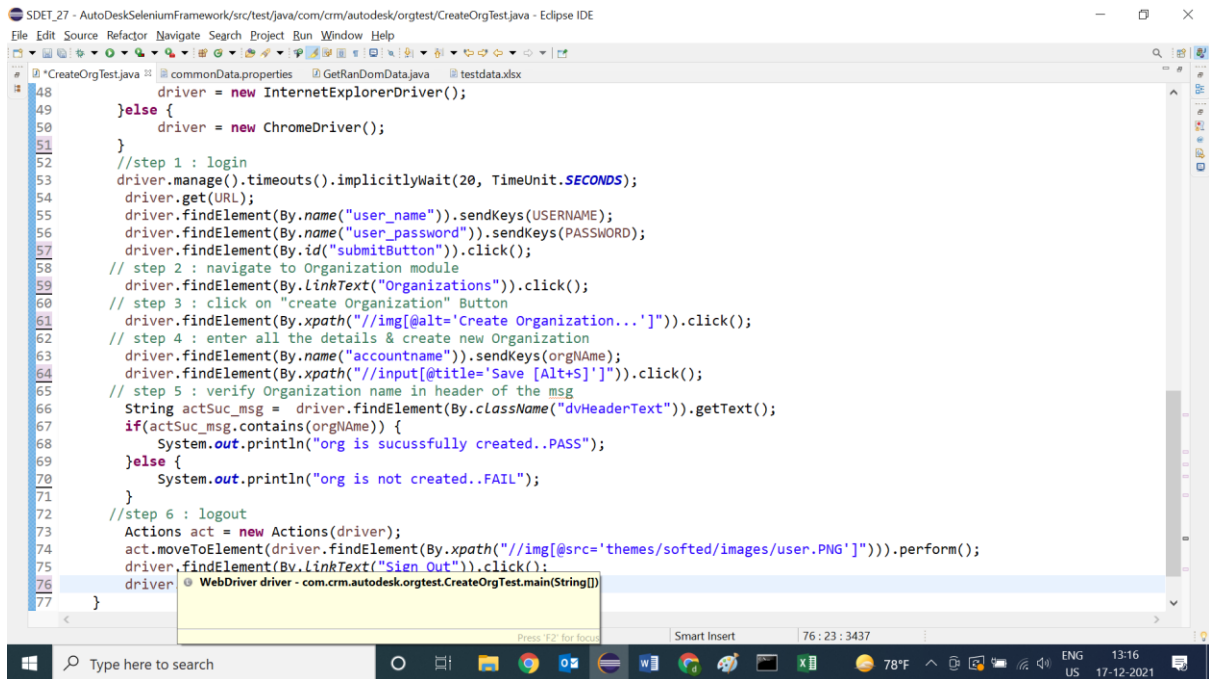
- step 1 : login
- step 2 : navigate to Organization module
- step 3 : click on "create Organization" Button
- step 4 : enter all the details & create new Organization
- step 5 : verify Organization name in header of the msg
- step 6 : logout

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



```
19 public class CreateOrgTest {
20     public static void main(String[] args) throws Throwable {
21         //read common data from properties File
22         FileInputStream fis = new FileInputStream("./data/commonData.properties");
23         Properties pObj = new Properties();
24         pObj.load(fis);
25         String URL = pObj.getProperty("url");
26         String USERNAME = pObj.getProperty("username");
27         String PASSWORD = pObj.getProperty("password");
28         String BROWSER = pObj.getProperty("browser");
29
30         //getRandom Num
31         Random ran = new Random();
32         int ranDomNum = ran.nextInt(10000);
33
34         //read test data from Excel File
35         FileInputStream fis_e = new FileInputStream("./data/testdata.xlsx");
36         Workbook wb = WorkbookFactory.create(fis_e);
37         Sheet sh = wb.getSheet("org");
38         Row row = sh.getRow(1);
39         String orgName = row.getCell(2).getStringCellValue() + ranDomNum;
40
41         WebDriver driver;
42         if(BROWSER.equals("firefox")) {
43             driver = new FirefoxDriver();
44         }else if(BROWSER.equals("chrome")) {
45             driver = new ChromeDriver();
46         }else if(BROWSER.equals("ie")) {
47             driver = new InternetExplorerDriver();
48         }
```



```
48 driver = new InternetExplorerDriver();
49 }else {
50 driver = new ChromeDriver();
51 }
52 //step 1 : login
53 driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
54 driver.get(URL);
55 driver.findElement(By.name("user_name")).sendKeys(USERNAME);
56 driver.findElement(By.name("user_password")).sendKeys(PASSWORD);
57 driver.findElement(By.id("submitButton")).click();
58 // step 2 : navigate to Organization module
59 driver.findElement(By.LinkText("Organizations")).click();
60 // step 3 : click on "create Organization" Button
61 driver.findElement(By.xpath("//img[@alt='Create Organization...']")).click();
62 // step 4 : enter all the details & create new Organization
63 driver.findElement(By.name("accountname")).sendKeys(orgName);
64 driver.findElement(By.xpath("//input[@title='Save [Alt+S]']")).click();
65 // step 5 : verify Organization name in header of the msg
66 String actSuc_msg = driver.findElement(By.className("dvHeaderText")).getText();
67 if(actSuc_msg.contains(orgName)) {
68 System.out.println("org is succussfully created..PASS");
69 }else {
70 System.out.println("org is not created..FAIL");
71 }
72 //step 6 : logout
73 Actions act = new Actions(driver);
74 act.moveToElement(driver.findElement(By.xpath("//img[@src='themes/softed/images/user.PNG']"))).perform();
75 driver.findElement(By.LinkText("Sign Out")).click();
76 driver
77 }
```

---

## 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.

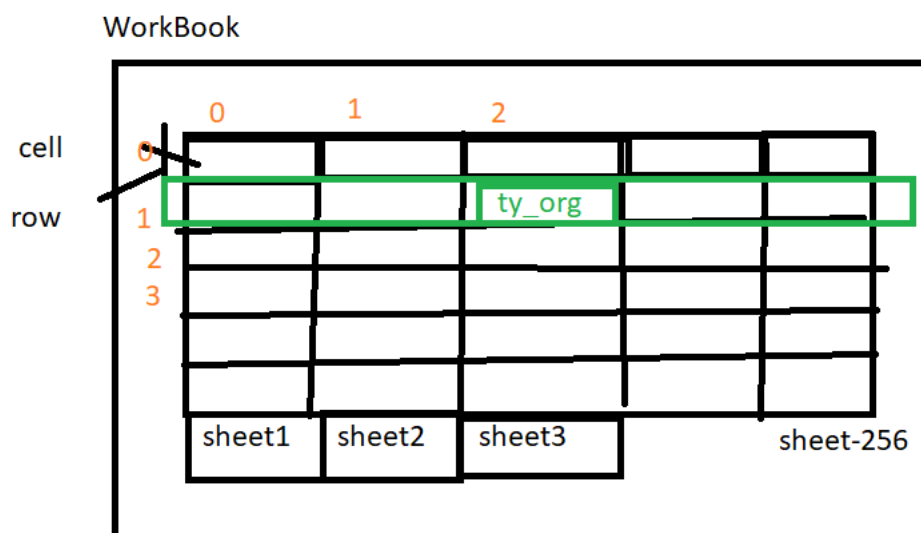
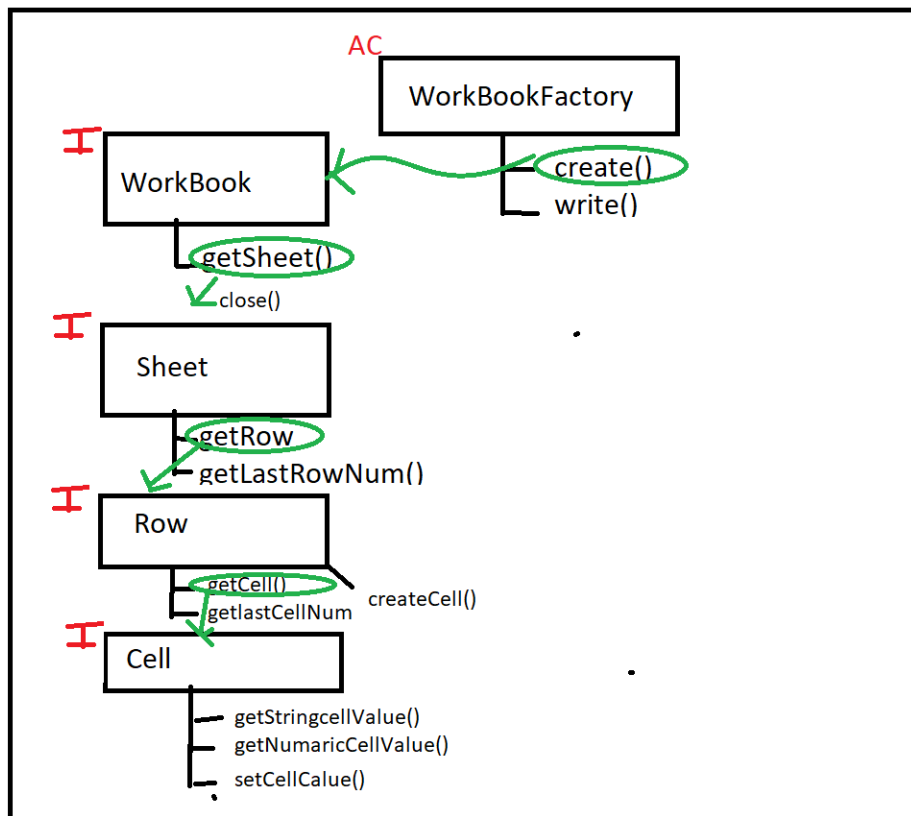
Installation steps:

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

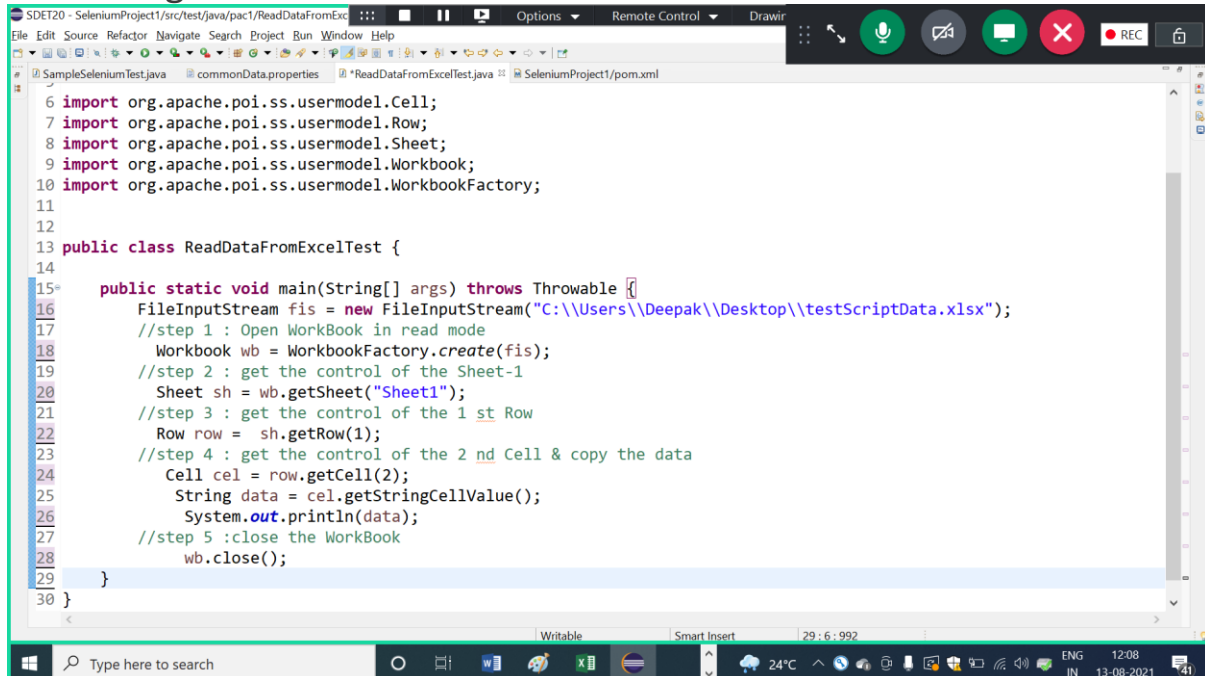
```
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi</artifactId>
  <version>4.0.0</version>
</dependency>
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi-ooxml</artifactId>
  <version>4.0.0</version>
</dependency>
```

---

# Class diagram of Apache POI

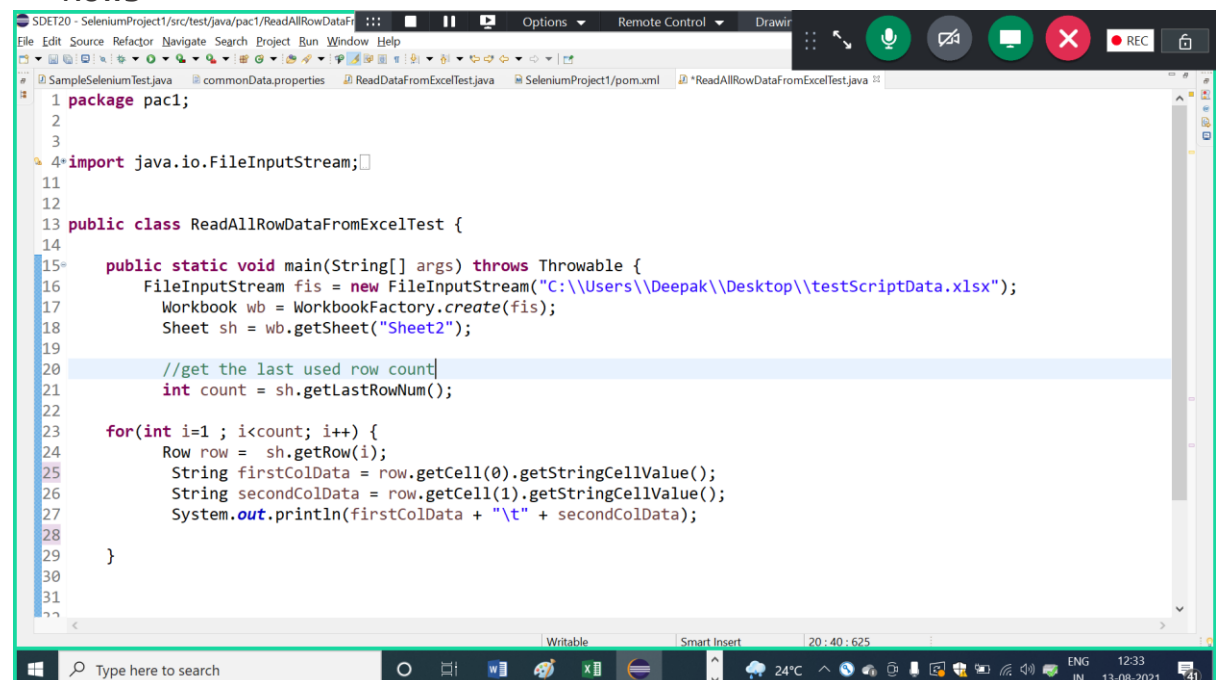


## 1. Program to read the data from WorkBook



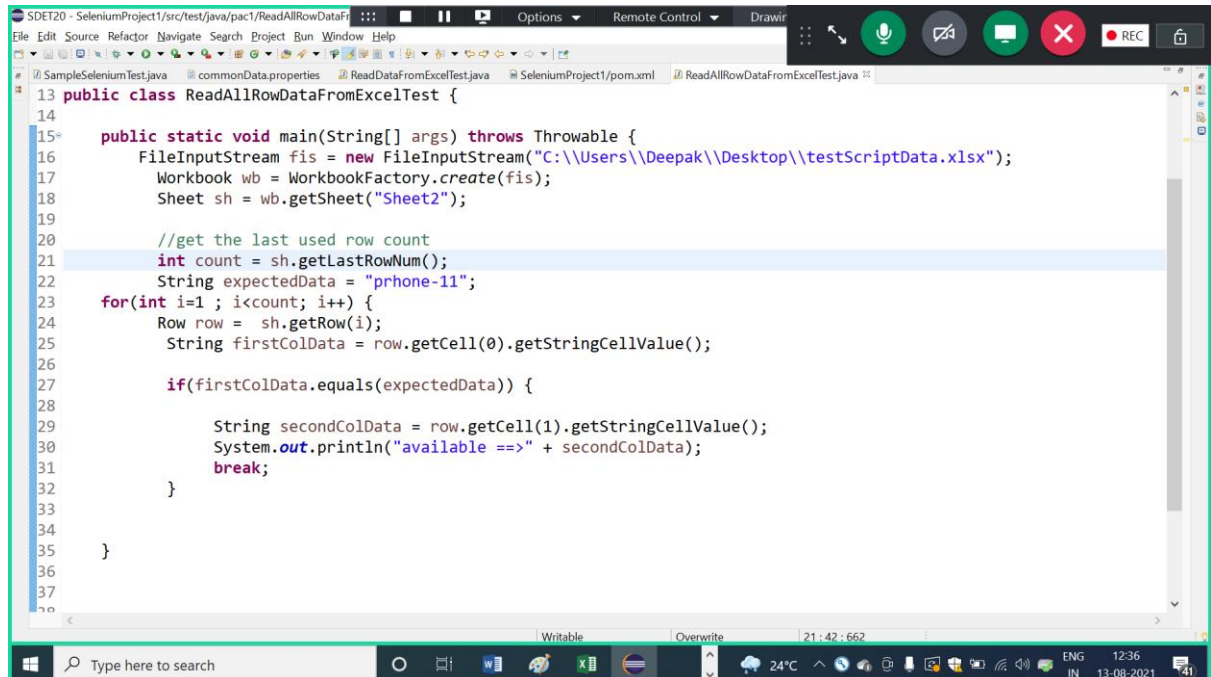
```
1  import org.apache.poi.ss.usermodel.Cell;
2  import org.apache.poi.ss.usermodel.Row;
3  import org.apache.poi.ss.usermodel.Sheet;
4  import org.apache.poi.ss.usermodel.Workbook;
5  import org.apache.poi.ss.usermodel.WorkbookFactory;
6
7  public class ReadDataFromExcelTest {
8
9      public static void main(String[] args) throws Throwable {
10         FileInputStream fis = new FileInputStream("C:\\Users\\Deepak\\Desktop\\testScriptData.xlsx");
11         //step 1 : Open WorkBook in read mode
12         Workbook wb = WorkbookFactory.create(fis);
13         //step 2 : get the control of the Sheet-1
14         Sheet sh = wb.getSheet("Sheet1");
15         //step 3 : get the control of the 1 st Row
16         Row row = sh.getRow(1);
17         //step 4 : get the control of the 2 nd Cell & copy the data
18         Cell cel = row.getCell(2);
19         String data = cel.getStringCellValue();
20         System.out.println(data);
21         //step 5 :close the WorkBook
22         wb.close();
23     }
24 }
```

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



```
1  package pac1;
2
3
4  import java.io.FileInputStream;
5
6  public class ReadAllRowDataFromExcelTest {
7
8      public static void main(String[] args) throws Throwable {
9         FileInputStream fis = new FileInputStream("C:\\Users\\Deepak\\Desktop\\testScriptData.xlsx");
10         Workbook wb = WorkbookFactory.create(fis);
11         Sheet sh = wb.getSheet("Sheet2");
12
13         //get the last used row count
14         int count = sh.getLastRowNum();
15
16         for(int i=1; i<count; i++) {
17             Row row = sh.getRow(i);
18             String firstColData = row.getCell(0).getStringCellValue();
19             String secondColData = row.getCell(1).getStringCellValue();
20             System.out.println(firstColData + "\t" + secondColData);
21         }
22     }
23 }
```

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

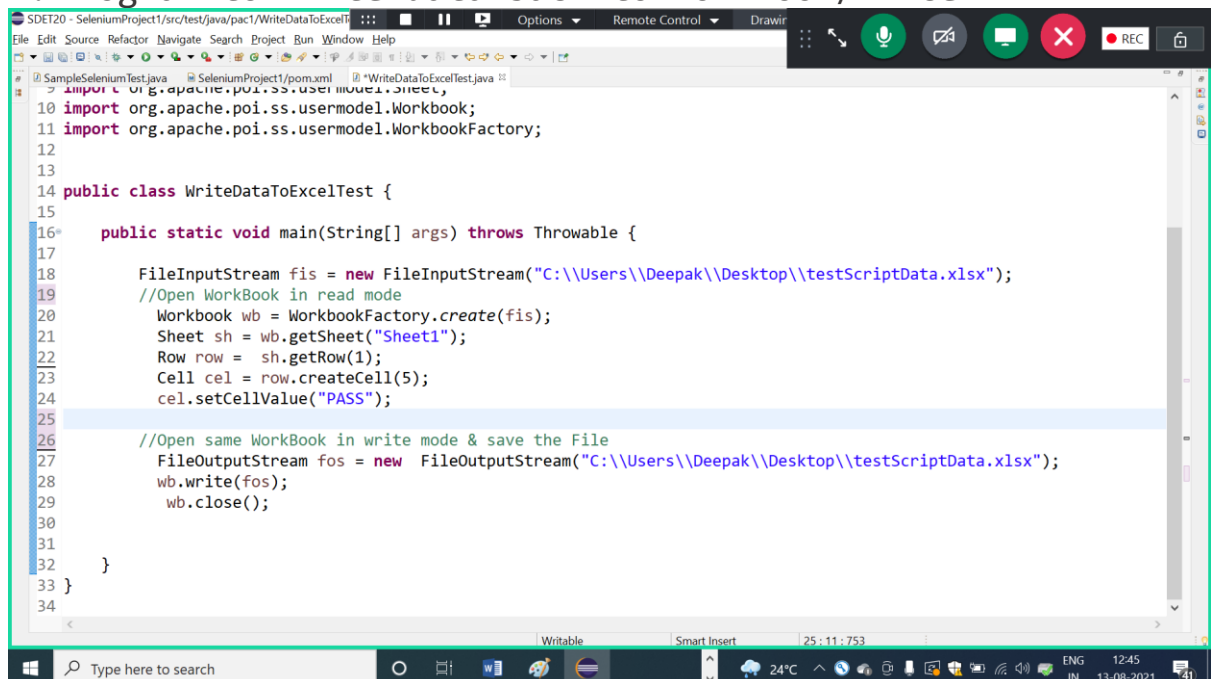


The screenshot shows an IDE window titled 'SDET20 - SeleniumProject1/src/test/java/pac1/ReadAllRowDataFromExcelTest.java'. The code is as follows:

```
13 public class ReadAllRowDataFromExcelTest {
14
15     public static void main(String[] args) throws Throwable {
16         FileInputStream fis = new FileInputStream("C:\\Users\\Deepak\\Desktop\\testScriptData.xlsx");
17         Workbook wb = WorkbookFactory.create(fis);
18         Sheet sh = wb.getSheet("Sheet2");
19
20         //get the last used row count
21         int count = sh.getLastRowNum();
22         String expectedData = "prhone-11";
23         for(int i=1 ; i<count; i++) {
24             Row row = sh.getRow(i);
25             String firstColData = row.getCell(0).getStringCellValue();
26
27             if(firstColData.equals(expectedData)) {
28
29                 String secondColData = row.getCell(1).getStringCellValue();
30                 System.out.println("available ==>" + secondColData);
31                 break;
32             }
33         }
34     }
35 }
36
37
38 }
```

The IDE interface includes a menu bar (File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help), a toolbar with various icons, and a status bar at the bottom showing 'Type here to search', 'Writable', 'Overwrite', and system information (21:42:662, 24°C, 13-08-2021).

4. Program to write data back to WorkBook/ Excel



The screenshot shows an IDE window titled 'SDET20 - SeleniumProject1/src/test/java/pac1/WriteDataToExcelTest.java'. The code is as follows:

```
10 import org.apache.poi.ss.usermodel.Workbook;
11 import org.apache.poi.ss.usermodel.WorkbookFactory;
12
13
14 public class WriteDataToExcelTest {
15
16     public static void main(String[] args) throws Throwable {
17
18         FileInputStream fis = new FileInputStream("C:\\Users\\Deepak\\Desktop\\testScriptData.xlsx");
19         //Open WorkBook in read mode
20         Workbook wb = WorkbookFactory.create(fis);
21         Sheet sh = wb.getSheet("Sheet1");
22         Row row = sh.getRow(1);
23         Cell cel = row.createCell(5);
24         cel.setCellValue("PASS");
25
26         //Open same WorkBook in write mode & save the File
27         FileOutputStream fos = new FileOutputStream("C:\\Users\\Deepak\\Desktop\\testScriptData.xlsx");
28         wb.write(fos);
29         wb.close();
30
31     }
32 }
33
34 }
```

The IDE interface includes a menu bar (File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help), a toolbar with various icons, and a status bar at the bottom showing 'Type here to search', 'Writable', 'Smart Insert', and system information (25:11:753, 24°C, 13-08-2021).



TestID	TestName	orgName	Status
tc_01	CreateOrg	TY_ORG	PASS
TestID	TestName	orgName	orgName
tc_02	CreateOrgWithType	Q_ORG	Energy

## How to generate random number using java

```

1 package practice;
2
3 import java.util.Random;
4
5 public class GetRanDomData {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9
10        Random ran = new Random();
11        int ranDomNum = ran.nextInt(10000);
12        System.out.println(ranDomNum);
13    }
14
15 }
16

```