**Generic Libraries / Utility**

**What is Generic components in Automation Framework?**

🡺it’s one of the automation framework components which is common for all the application

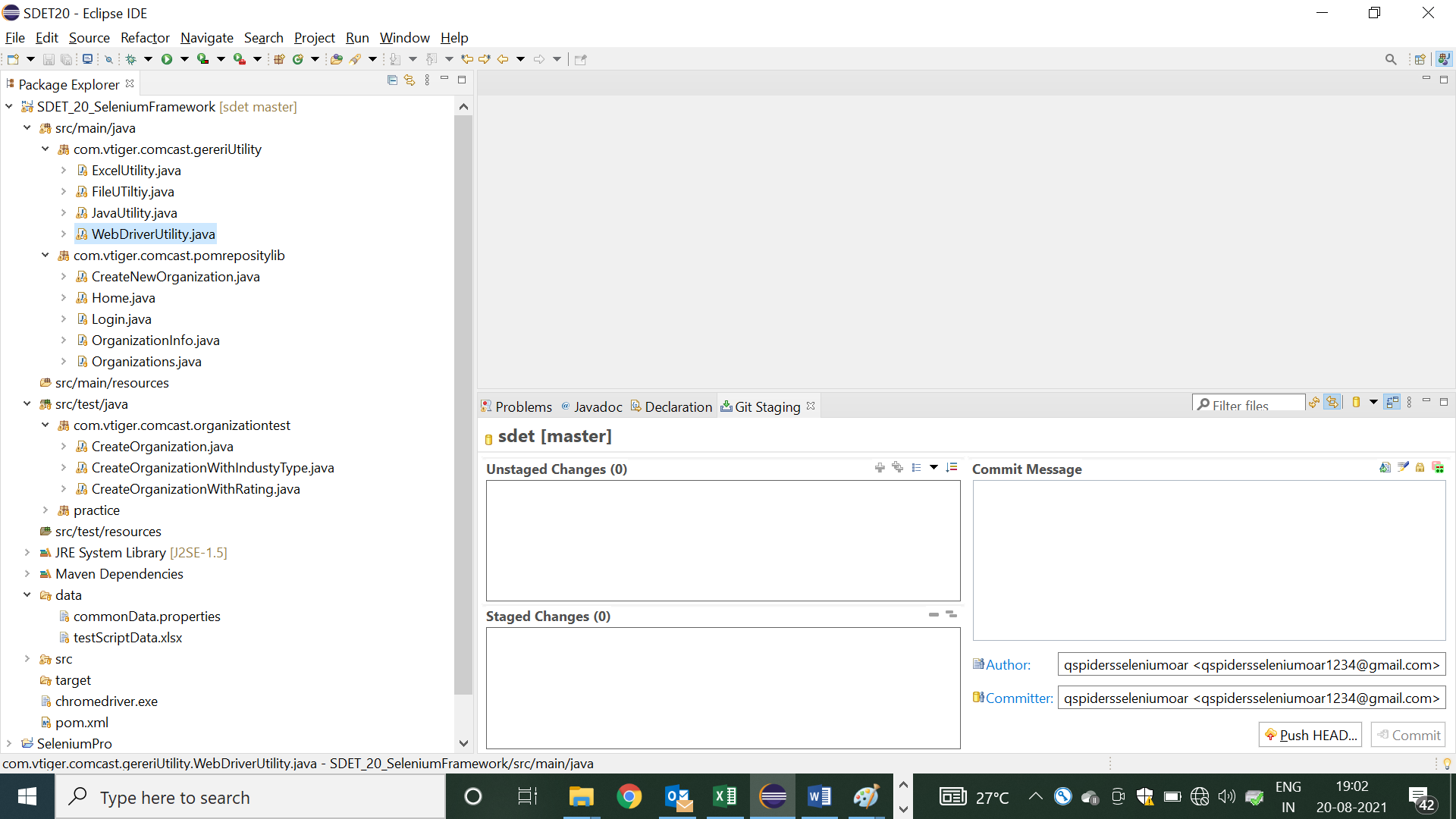
🡺its collection of generic class contains reusable methods / libraries

🡺The methods which can be used to any application is called Generic/common methods

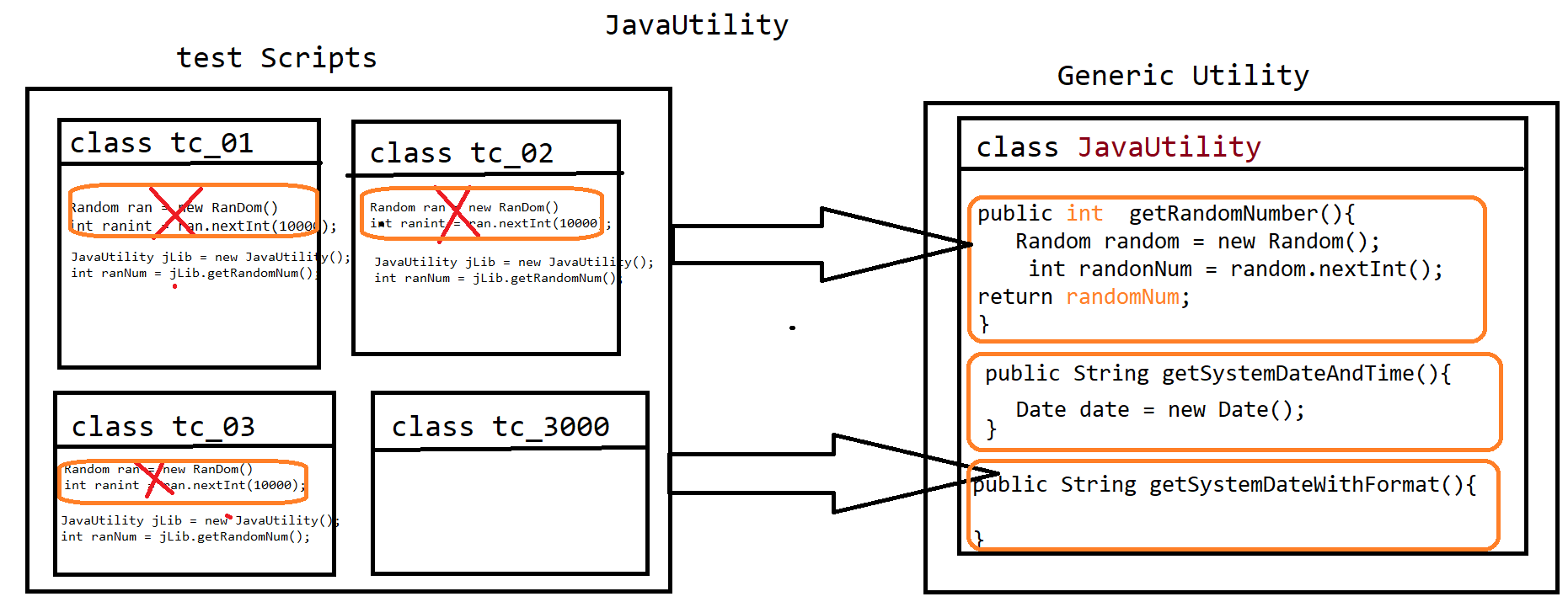
**What is the advantages of Generic components?**

* Reusability of code
* Code Optimization
* Test script development is faster
* Test scripts Code readability
* Generic libraries are common to all automation projects
* Avoid duplicate Code
* no need to remember the syntax every time , just create once & use multiple times

**Generic Utility Structure in Automation Project**



1. Java Utility Libraries

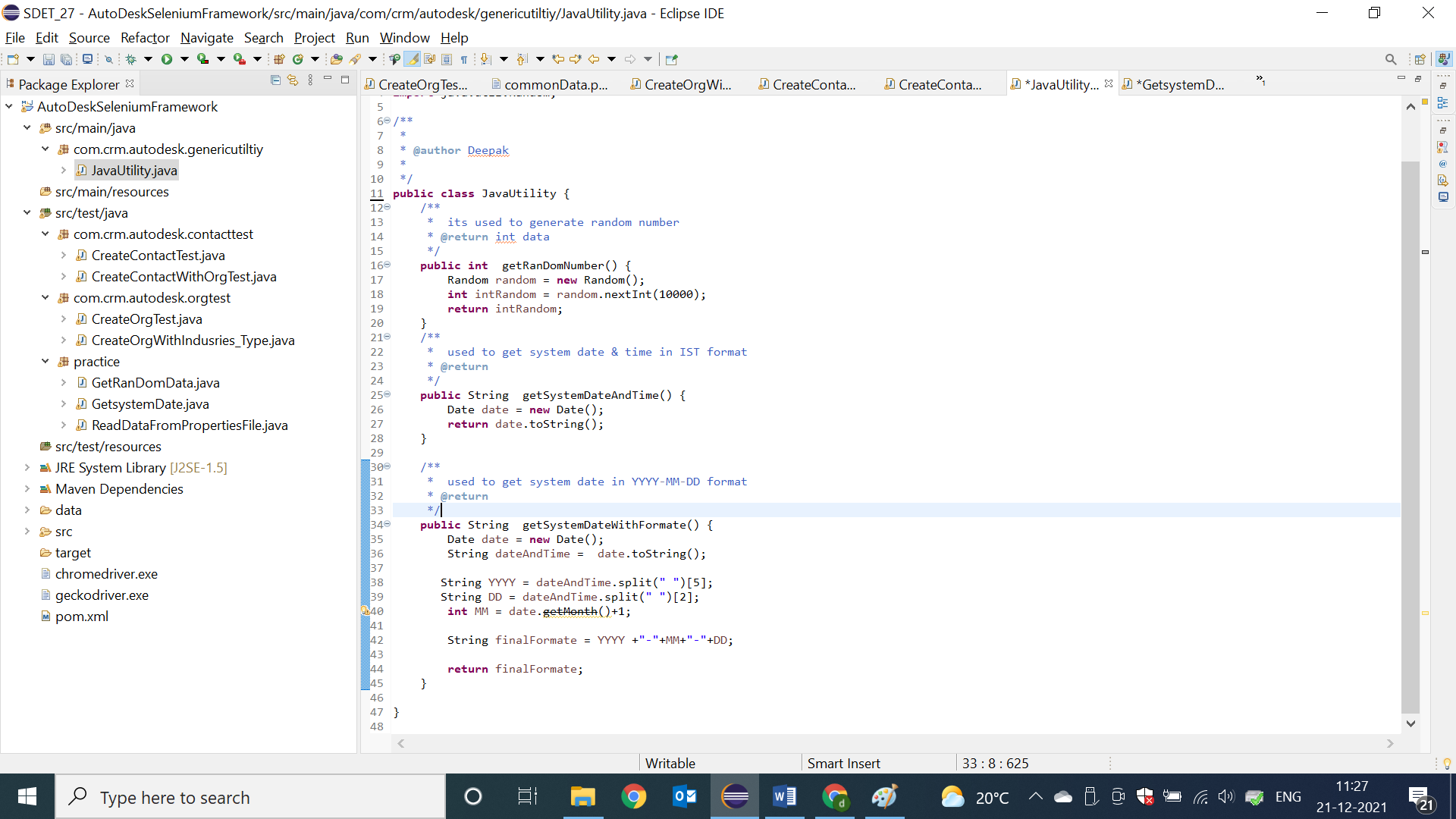


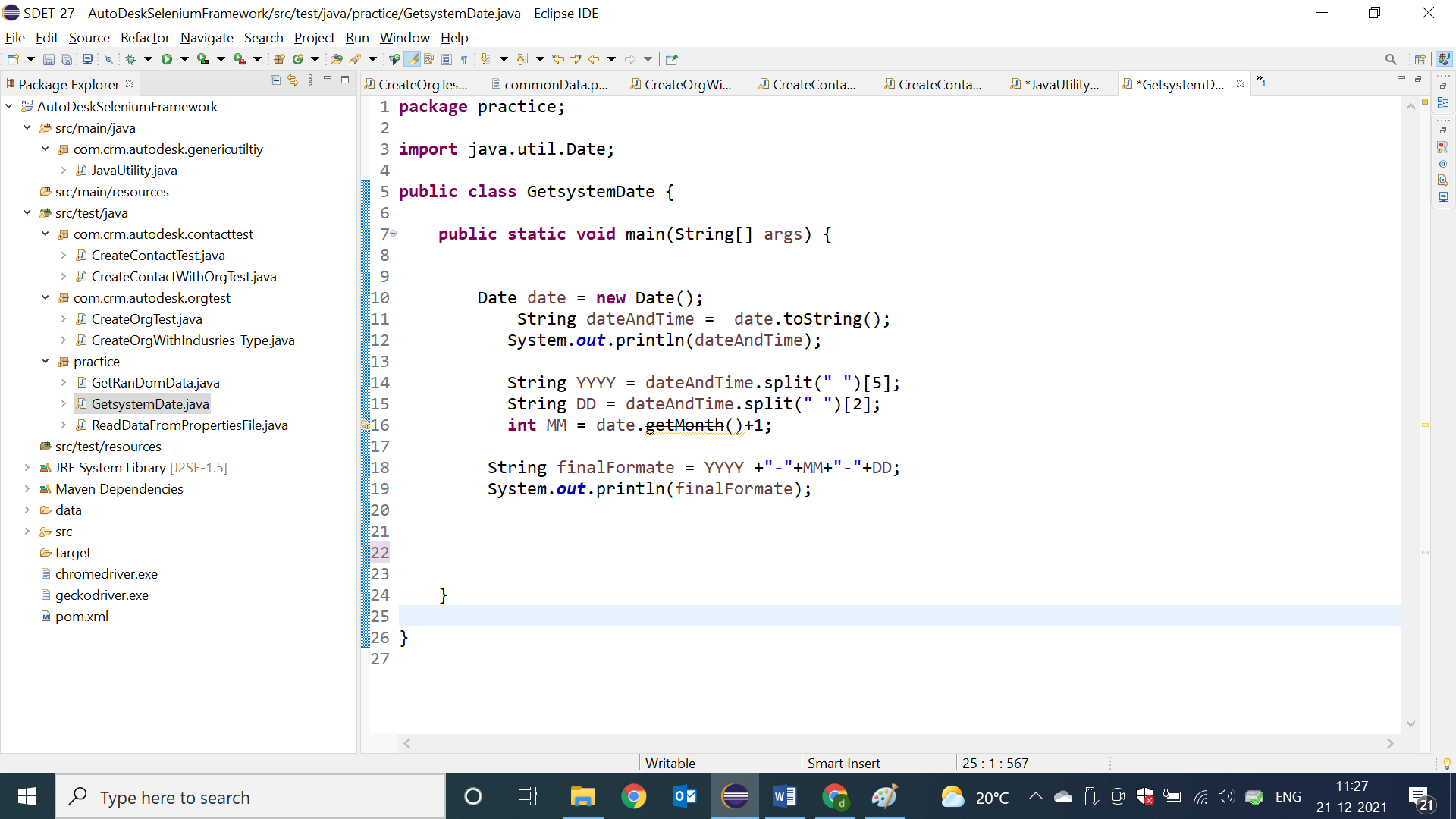
🡺 Java Utility is one class in generic component, which contain java specific methods which can be used across the test Scripts / Application

🡺 its contains several generic reusable methods like

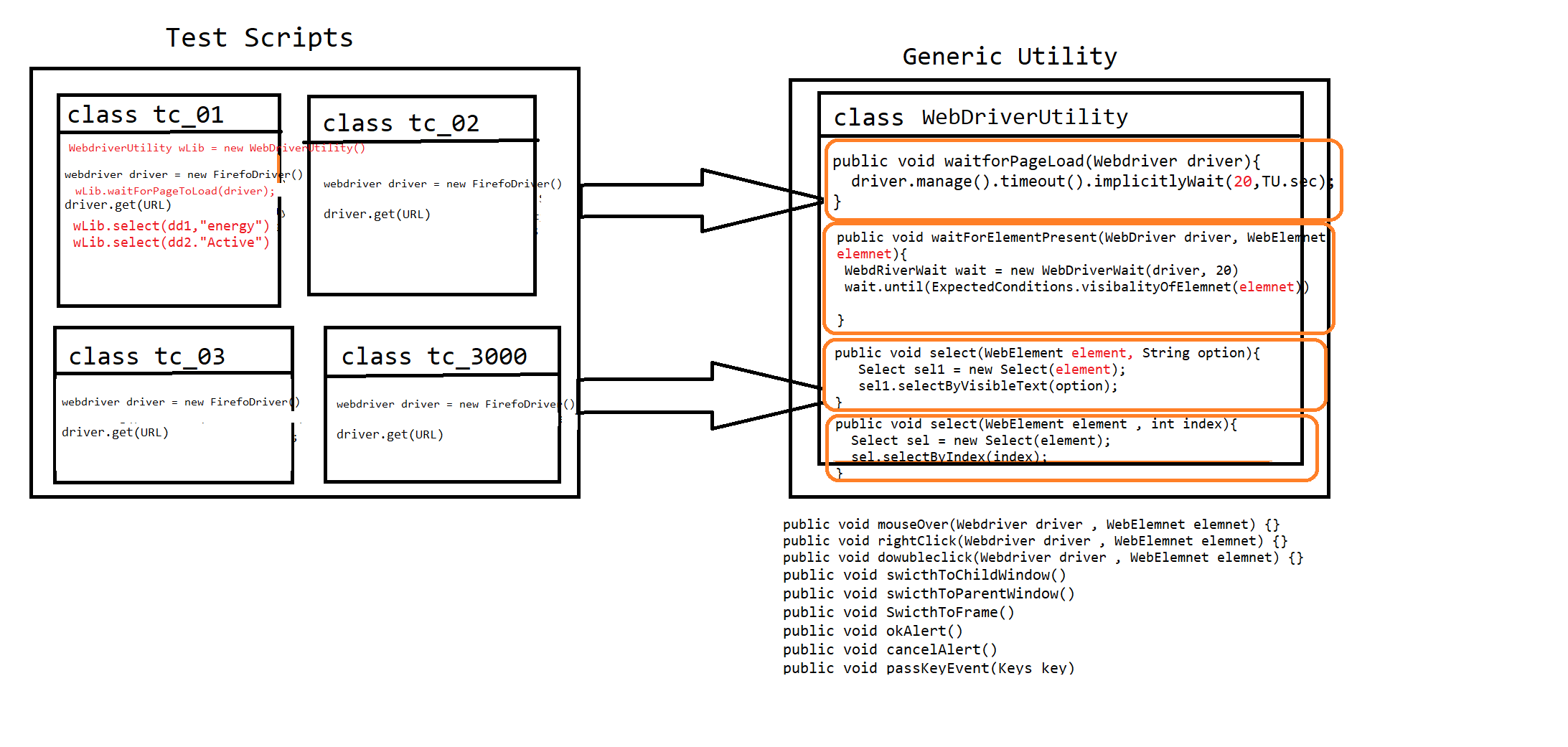
* getRandomNum() : it’s used to generate random number for every invocation
* getSystemDate() : it’s used to generate system date and time

=====================Code===============================





1. WebDriver Utility Libraries



* WebdriverUtility is a Generic class , which contains webdriver specific reusable actions like
* waitForPageToLoad()
* waitForElement()
* select()
* accpertAlert()
* cancelAlert() .Etc

==========Code======================

**package** com.crm.comcast.genericutility;

**import** java.io.File;

**import** java.util.Iterator;

**import** java.util.Set;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.Keys;

**import** org.openqa.selenium.JavascriptExecutor;

**import** org.openqa.selenium.OutputType;

**import** org.openqa.selenium.TakesScreenshot;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.interactions.Actions;

**import** org.openqa.selenium.support.ui.ExpectedConditions;

**import** org.openqa.selenium.support.ui.FluentWait;

**import** org.openqa.selenium.support.ui.Select;

**import** org.openqa.selenium.support.ui.WebDriverWait;

**import** com.google.common.io.Files;

/\*\*

\* its contains WebDriver specific reusable actions

\* **@author** Deepak

\*

\*/

**public** **class** WebDriverUtility {

/\*\*

\* wait for page to load before indentifying any sychronized element in DOM [HTML-Docuent]

\* **@param** driver

\*/

**public** **void** waitForPageToLoad(WebDriver driver) {

driver.manage().timeouts().implicitlyWait(20, TimeUnit.***SECONDS***);

}

/\*\*

\* wait for page to load before indentifying any assychronized[java scripts actions] element in DOM [HTML-Docuent]

\* **@param** driver

\*/

**public** **void** waitForPageToLoadForJSElement(WebDriver driver) {

driver.manage().timeouts().setScriptTimeout(20, TimeUnit.***SECONDS***);

}

/\*\*

\* used to wait for element to be clickable in GUI , & check for specific element for every 500 milli seconds

\* **@param** driver

\* **@param** element

\*/

**public** **void** waitForElemnetToBeClickAble(WebDriver driver, WebElement element) {

WebDriverWait wait = **new** WebDriverWait(driver, 20);

wait.until(ExpectedConditions.*elementToBeClickable*(element));

}

/\*\*

\* used to wait for element to be clickable in GUI , & check for specific element for every 500 milli seconds

\* **@param** driver

\* **@param** element

\* **@param** pollingTime in the form second

\* **@throws** Throwable

\*/

**public** **void** waitForElementWithCumtomTimeOut(WebDriver driver, WebElement element, **int** pollingTime) **throws** Throwable {

FluentWait wait = **new** FluentWait(driver);

wait.~~pollingEvery~~(pollingTime, TimeUnit.***SECONDS***);

wait.wait(20);

wait.until(ExpectedConditions.*elementToBeClickable*(element));

}

/\*\*

\* used to Switch to Any Window based on Window Title

\* **@param** driver

\* **@param** partialWindowTitle

\*/

**public** **void** swithToWindow(WebDriver driver , String partialWindowTitle) {

Set<String> set = driver.getWindowHandles();

Iterator<String> it = set.iterator();

**while** (it.hasNext()) {

String wID = it.next();

driver.switchTo().window(wID);

String currentWindowTitle = driver.getTitle();

**if**(currentWindowTitle.contains(partialWindowTitle)) {

**break**;

}

}

}

/\*\*

\* used to Switch to Alert Window & click on OK button

\* **@param** driver

\*/

**public** **void** swithToAlertWindowAndAccpect(WebDriver driver) {

driver.switchTo().alert().accept();

}

/\*\*

\* used to Switch to Alert Window & click on Cancel button

\* **@param** driver

\*/

**public** **void** swithToAlertWindowAndCancel(WebDriver driver) {

driver.switchTo().alert().dismiss();

}

/\*\*

\* used to Switch to Frame Window based on index

\* **@param** driver

\* **@param** index

\*/

**public** **void** swithToFrame(WebDriver driver , **int** index) {

driver.switchTo().frame(index);

}

/\*\*

\* used to Switch to Frame Window based on id or name attribute

\* **@param** driver

\* **@param** id\_name\_attribute

\*/

**public** **void** swithToFrame(WebDriver driver , String id\_name\_attribute) {

driver.switchTo().frame(id\_name\_attribute);

}

/\*\*

\* used to select the value from the dropDwon based on index

\* **@param** driver

\* **@param** index

\*/

**public** **void** select(WebElement element , **int** index) {

Select sel = **new** Select(element);

sel.selectByIndex(index);

}

/\*\*

\* used to select the value from the dropDwon based on value / option avlaible in GUI

\* **@param** element

\* **@param** value

\*/

**public** **void** select(WebElement element , String text) {

Select sel = **new** Select(element);

sel.selectByVisibleText(text);

}

/\*\*

\* used to place mouse cursor on specified element

\* **@param** driver

\* **@param** elemnet

\*/

**public** **void** mouseOverOnElemnet(WebDriver driver , WebElement elemnet)

{

Actions act = **new** Actions(driver);

act.moveToElement(elemnet).perform();

}

/\*\*

\* used to right click on specified element

\* **@param** driver

\* **@param** elemnet

\*/

**public** **void** rightClickOnElement(WebDriver driver , WebElement elemnet)

{

Actions act = **new** Actions(driver);

act.contextClick(elemnet).perform();

}

/\*\*

\*

\* **@param** driver

\* **@param** javaScript

\*/

**public** **void** executeJavaScript(WebDriver driver , String javaScript) {

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeAsyncScript(javaScript, **null**);

}

**public** **void** waitAndClick(WebElement element) **throws** InterruptedException

{

**int** count = 0;

**while**(count<20) {

**try** {

element.click();

**break**;

}**catch**(Throwable e){

Thread.*sleep*(1000);

count++;

}

}

}

**public** **void** takeScreenshot(WebDriver driver, String screenshotName) **throws** Throwable {

TakesScreenshot ts=(TakesScreenshot)driver;

File src=ts.getScreenshotAs(OutputType.***FILE***);

File dest=**new** File("./screenshot/"+screenshotName+".PNG");

Files.*copy*(src, dest);

}

/\*\*

\* pass enter Key appertain in to Browser

\* **@param** driver

\*/

**public** **void** passEnterKey(WebDriver driver) {

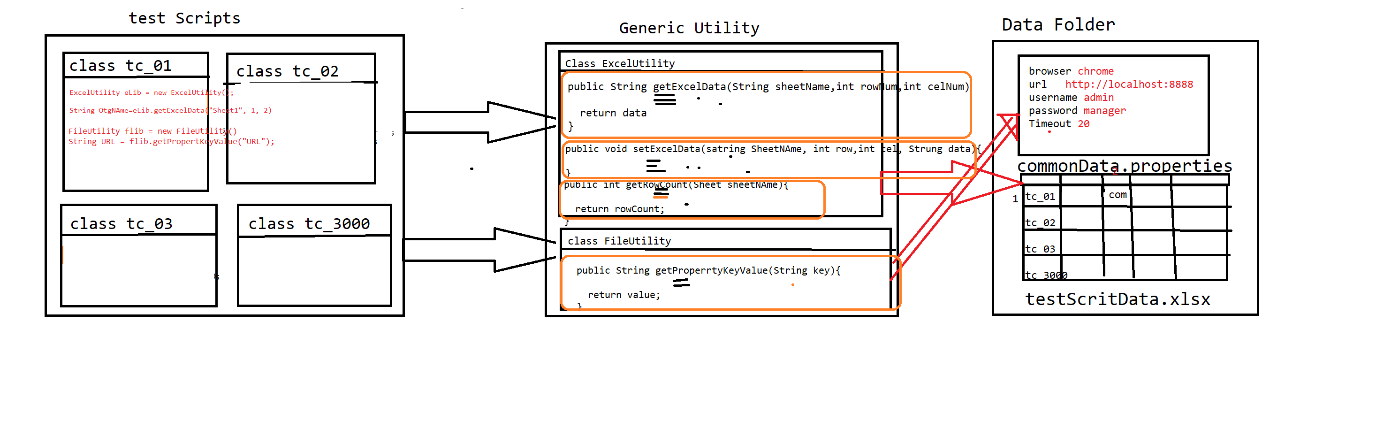
Actions act = **new** Actions(driver);

act.sendKeys(Keys.ENTER).perform();

}

}

1. Excel Utility libraries



* As per the rule of automation, data should not be hardcoded with in the test scripts, so that to get the data from external file like Excel & .propertes file

We go for ExcelUtility & FileUtlity

* Excel Utility class is developed using apache Poi libraries, which is used to read the data from Excel
* FileUtility is used to get the data from .properties file

==========Code=======================

**package** com.vtiger.comcast.gereriUtility;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** org.apache.poi.ss.usermodel.Cell;

**import** org.apache.poi.ss.usermodel.Row;

**import** org.apache.poi.ss.usermodel.Sheet;

**import** org.apache.poi.ss.usermodel.Workbook;

**import** org.apache.poi.ss.usermodel.WorkbookFactory;

/\*\*

\* its developed using Apache POi libraries , which used to handle Microsoft Excel sheet

\* **@author** Deepak

\*

\*/

**public** **class** ExcelUtility {

/\*\*

\* its used read the data from excel base don below arguments

\* **@param** sheetName

\* **@param** rowNum

\* **@param** celNum

\* **@return** Data

\* **@throws** Throwable

\*/

**public** String getDataFromExcel(String sheetName , **int** rowNum, **int** celNum) **throws** Throwable {

FileInputStream fis = **new** FileInputStream("./data/testScriptData.xlsx");

Workbook wb = WorkbookFactory.*create*(fis);

Sheet sh = wb.getSheet(sheetName);

Row row = sh.getRow(rowNum);

String data = row.getCell(celNum).getStringCellValue();

wb.close();

**return** data;

}

/\*\*

\* used to get the last used row number on specified Sheet

\* **@param** sheetName

\* **@return**

\* **@throws** Throwable

\*/

**public** **int** getRowCount(String sheetName) **throws** Throwable {

FileInputStream fis = **new** FileInputStream("./data/testScriptData.xlsx");

Workbook wb = WorkbookFactory.*create*(fis);

Sheet sh = wb.getSheet(sheetName);

wb.close();

**return** sh.getLastRowNum();

}

**public** **void** setDataExcel(String sheetName , **int** rowNum, **int** celNum ,String data) **throws** Throwable {

FileInputStream fis = **new** FileInputStream("./data/testScriptData.xlsx");

Workbook wb = WorkbookFactory.*create*(fis);

Sheet sh = wb.getSheet(sheetName);

Row row = sh.getRow(rowNum);

Cell cel = row.createCell(celNum);

cel.setCellValue(data);

FileOutputStream fos = **new** FileOutputStream("./data/testScriptData.xlsx");

wb.write(fos);

wb.close();

}

}

1. **File Utility=====================**

**package** com.vtiger.comcast.gereriUtility;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** java.util.Properties;

/\*\*

\*

\* **@author** Deepak

\*

\*/

**public** **class** FileUTiltiy {

/\*\*

\* its used to read the data from commonData.properties File based on Key which you pass as an argument

\* **@param** key

\* **@throws** Throwable

\*/

**public** String getPropertyKeyValue(String key) **throws** Throwable {

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

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

pobj.load(fis);

String value = pobj.getProperty(key);

**return** value;

}

}

================Sample Test Using Generic Utility=======

package com.crm.comcast.conatctTest;

import java.io.FileInputStream;

import java.util.Iterator;

import java.util.Properties;

import java.util.Random;

import java.util.Set;

import java.util.concurrent.TimeUnit;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.Workbook;

import org.apache.poi.ss.usermodel.WorkbookFactory;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.ie.InternetExplorerDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import com.crm.comcast.genericutility.ExcelUtility;

import com.crm.comcast.genericutility.FileUtility;

import com.crm.comcast.genericutility.JavaUtlity;

import com.crm.comcast.genericutility.WebDriverUtility;

/\*\*

\*

\* @author Deepak

\*

\*/

public class CreateConatctWithOrgTest{

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

/\* create object to libraries\*/

JavaUtlity jLib = new JavaUtlity();

WebDriverUtility wLib = new WebDriverUtility();

FileUtility fLib = new FileUtility();

ExcelUtility eLib = new ExcelUtility();

/\* get ramDomData \*/

int randomNum = jLib.getRandomNumber();

/\* read common data from Properties File\*/

String BROWER = fLib.getPropertyKeyValue("browser");

String URL = fLib.getPropertyKeyValue("url");

String USERNAME = fLib.getPropertyKeyValue("username");

String PASSWORD = fLib.getPropertyKeyValue("password");

/\* read test data from Excel File\*/

String orgName = eLib.getDataFromExcel("contact", 4, 2) + randomNum;

String conactName = eLib.getDataFromExcel("contact", 4, 3) + randomNum;

/\* launch the Browser \*/

WebDriver driver = null;

if(BROWER.equals("chrome")) {

driver = new ChromeDriver();

}else if(BROWER.equals("firefox")){

driver = new FirefoxDriver();

}else if(BROWER.equals("ie")){

driver = new InternetExplorerDriver();

}else {

driver = new ChromeDriver();

}

wLib.waitForPageToLoad(driver) ;

driver.get(URL);

/\* step 1 : login to APP \*/

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

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

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

/\* step 1 : navigate to Org Page \*/

driver.findElement(By.linkText("Organizations")).click();

/\* step 2 : navigate to CREATE Org Page\*/

driver.findElement(By.xpath("//img[@title='Create Organization...']")).click();

/\* step 3 : create a new Org \*/

driver.findElement(By.name("accountname")).sendKeys(orgName);

driver.findElement(By.xpath("//input[@title='Save [Alt+S]']")).click();

WebElement headerWb = driver.findElement(By.className("dvHeaderText"));

wLib.waitForElemnetToBeClickAble(driver, headerWb);

/\* step 4 : navigate to Contact Page \*/

driver.findElement(By.linkText("Contacts")).click();

/\* step 5 : navigate to CREATE Coantct Page\*/

driver.findElement(By.xpath("//img[@title='Create Contact...']")).click();

/\* step 6 : create a new contact With org Name \*/

driver.findElement(By.name("lastname")).sendKeys(conactName);

driver.findElement(By.xpath("//input[@name='account\_id']/following-sibling::img")).click();

wLib.swithToWindow(driver, "Accounts");

driver.findElement(By.name("search\_text")).sendKeys(orgName);

driver.findElement(By.name("search")).click();

driver.findElement(By.xpath("//a[text()='"+orgName+"']")).click();

wLib.swithToWindow(driver, "Contacts");

driver.findElement(By.xpath("//input[@title='Save [Alt+S]']")).click();

wLib.mouseOverOnElemnet(driver,driver.findElement(By.xpath("//img[@src='themes/softed/images/user.PNG']")));

driver.findElement(By.linkText("Sign Out")).click();

driver.close();

}

}