AUTOMATION FRAMEWORK

BY ..PRAKASH MAHAJAN

CONTENT

- Defination of Framework and Types
- Components Of Framework
- Framework Architecture
- Test Data and Common data
- Generic Utility
- POM (Page Object Model)
- TestNG
- Exceptions

Framework

Framework is set of rules or guidelines ,Files and Folders (Standard folder template) that every Automation Engineer should follow while automating the application .

OR

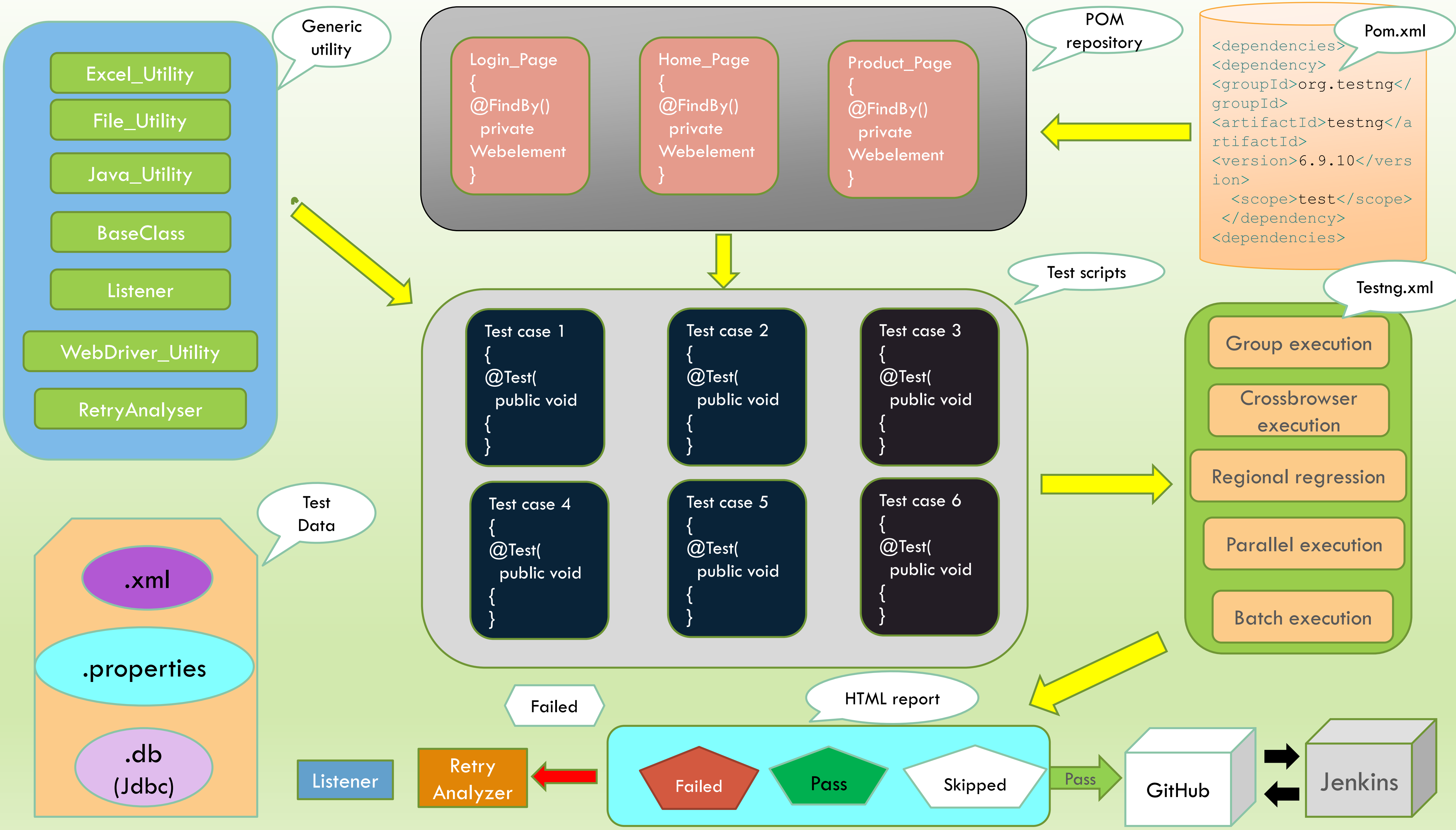
We can say that Framework is a collection of reusable methods (like generic methods) which makes automation engineers life easy and also makes coding efficient.

In my frame work I have designed using Selenium Automation Tool .

Types Of Frameworks

1. Data Driven Framework
2. Keyword Driven Framework
3. Method Driven Framework
4. Page Object Model (POM)
5. TestNG (NG → Next Generation)
6. Modular Framework
7. Hybrid Framework

Framework Architecture



Generic utility

Excel_Utility

File_Utility

Java_Utility

BaseClass

Listener

WebDriver_Utility

RetryAnalyser

POM repository

Login_Page { @FindBy() private Webelement }

Home_Page { @FindBy() private Webelement }

Product_Page { @FindBy() private Webelement }

Pom.xml

```
<dependencies>
<dependency>
<groupId>org.testng</groupId>
<artifactId>testng</artifactId>
<version>6.9.10</version>
  <scope>test</scope>
 </dependency>
<dependencies>
```

Test scripts

Test case 1 { @Test(public void { }

Test case 2 { @Test(public void { }

Test case 3 { @Test(public void { }

Test case 4 { @Test(public void { }

Test case 5 { @Test(public void { }

Test case 6 { @Test(public void { }

Testng.xml

Group execution

Crossbrowser execution

Regional regression

Parallel execution

Batch execution

Test Data

.xml

.properties

.db (Jdbc)

Listener

Retry Analyzer

Failed

HTML report

Failed

Pass

Skipped

Pass

GitHub

Jenkins

Framework used

- ❑ As the part of project I have designed the Hybrid Framework it is nothing but combination of multiple framework like Data Driven ,Keyword Driven , POM ,TestNG and Method Driven .

- ❑ In this framework using Data Driven approach I have achieved the generalization of data that has to be test, this helped to avoid hardcoding of test scripts.

- ❑ By using keyword Driven approach fetching the common data which can be used frequently in every scripts like browser ,URL , valid login credentials from external files like text file or property file where data is fetched using Key – Value by using key we get the value for that particular key.

- ❑ By using POM I avoided the insertion of X-path locator in scripts which reduced the code redundancy and achieved reusability of Web Elements.

- ❑ By using TestNG achieved the control over Test scripts execution that we control the flow of execution of scripts and to generate the Test Report using a suite file (testing.xml).Which also provides data driven testing using Data Provider.

- ❑ By using method driven achieved code reusability by creating Generic utility classes which contains all reusable methods …that we will see in upcoming slides.

Test Data

- Test data which we store in excel files is called as Test Data

- In excel sheet each cell behaves as a one test data

- Excel sheet contains large amount of data that is why it is heavy weighted as compare to property file which is light weight file.

- For each test script test data is different .

- To access or to fetch the data from excel file we need to take the help of Apache poi connection which is third party tool.

Common Data

- Common data is nothing but the data which can be used frequently in every test script. Such as browser , valid login credential like username and password

- We used to store the common data in property file which is light weight file and faster than excel file

- To access the common data from property file we use the Key-Value pair. By passing the Key we get in return as value.

- In property file , key value pair are separated by space.

- By default data is stored in String format

Generic Utility

- Generic Utility is one of the main component of framework which is a package that I have created in my framework which contains all generic class .generic class are nothing but the class which contains all reusable codes which is almost part of the all test scripts .
- The generic utilities helps to easy manipulation and maintenance of test scripts.this very much helps to reduce the code size. The generic classes that I have created are as follows :

BaseClass :

This is one of the generic class in generic utility which contains some reusable methods which helps to developed the test scripts fast and which allows easy maintenance .It also reduces the code redundancy.

In this class some reusable methods as

1.setup () which is used to launch the browser
2.login () which is used to login the application
3.logout ()
4.teardown() which is used to close the browser

File_utility

File utility is one of the generic class of generic utility which contains reusable methods which can used in every test script . Methods such as

1. getProperty ()

this method is used used to read the from property file which contains common data such as login credential.

```
public String getPropertyKeyValue(String Key) throws Throwable
{FileInputStream fis = new FileInputStream("./data/config.properties");
Properties prop = new Properties();
prop.load(fis);
String value = prop.getProperty(Key);
return value;}
```

Excel_utility

Excel utility is one of the generic class of generic utility which contains reusable methods which can used in every test script . Methods such as

1. getExcelData()

this method is used to read the from excel file which contains test data such as invalid login credential and some text.

```
public String getDataFromExcel1(String sheetName,int RowNum,int cellNum) throws
EncryptedDocumentException, IOException
{FileInputStream fis = new FileInputStream("./data/creds.xlsx");
Workbook book = WorkbookFactory.create(fis);
DataFormatter format=new DataFormatter();
return format.formatCellValue(book.getSheet(sheetName).getRow(RowNum).getCell(cellNum));}
```

Java_utility

Java utility is one of the generic class of generic utility which contains reusable methods which can used in every test script . Methods such as

1. getRandomNum ()

   this method is used to get the random number for every fetch data

```
public int getRanDomNum()
{
Random ran = new Random();
int RanNum = ran.nextInt(1000);
return RanNum;
}
```

WebDriver_utility

WebDriver utility is one of main the generic class of generic utility which contains reusable of webdriver methods which can used in every test script . Methods such as

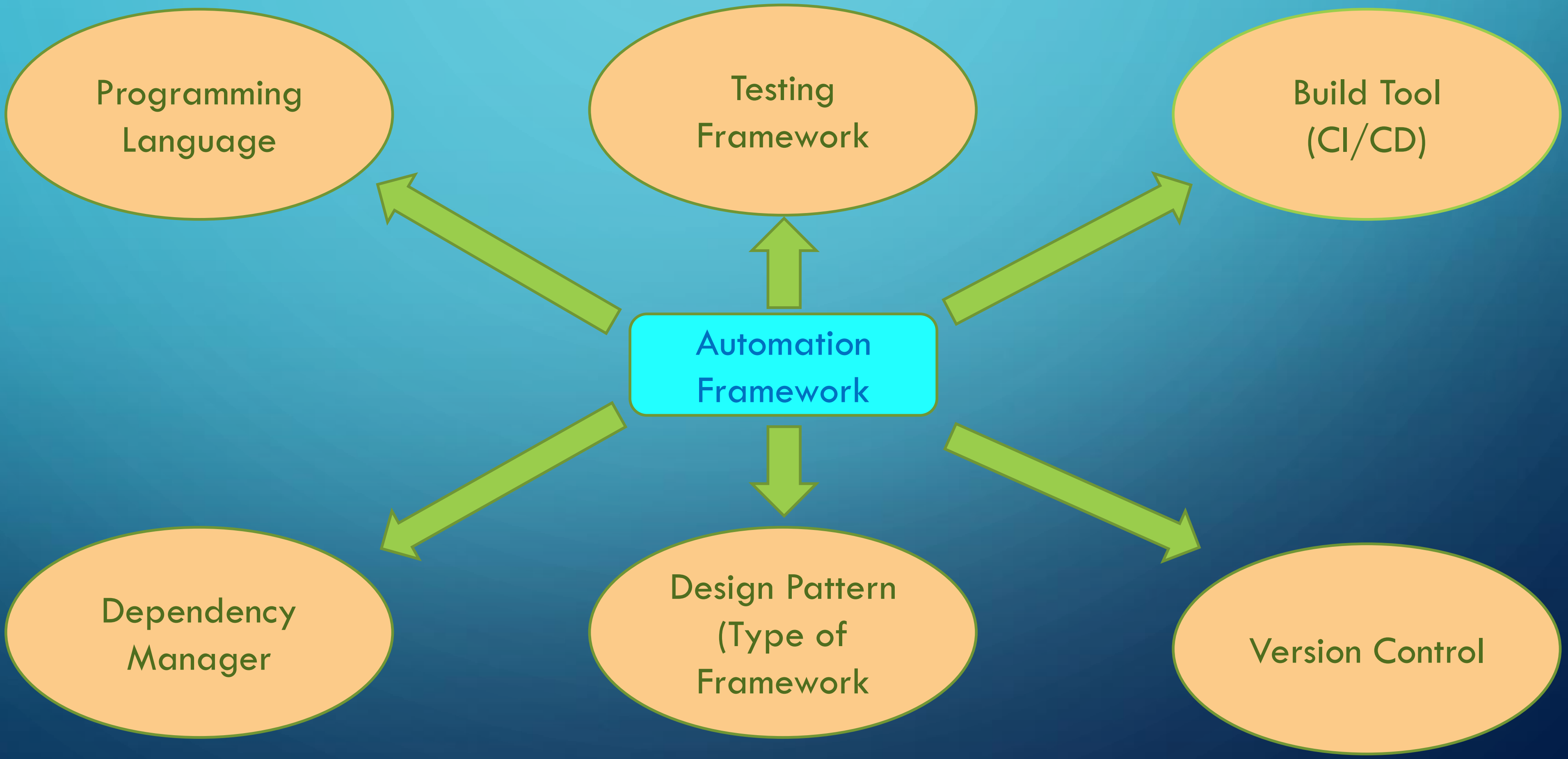
1. waitForPageToLoad()

   this method is used to hold the execution of test scripts till it gets loaded to avoid exception regarding web element

```
public void waitForPageToLoad(WebDriver driver)
{
driver.manage().timeouts().implicitlyWait(20,TimeUnit.SECONDS);
}
```

➢ Components Of Framework

The components that make up a test automation framework include what kind of programming language is used to write scripts ,which testing tool is used ,which build tool is used for continuous integration and continuous development of test scripts ,Dependency manager, Design pattern means type of framework used , and repository to control changes or common repository. In my framework what and all components are used which we will see in next slides.



Automation Framework

Programming Language

Testing Framework

Build Tool (CI/CD)

Dependency Manager

Design Pattern (Type of Framework

Version Control

Programming Language

- In my project I have developed the framework using Java programming language because I have used selenium tool for Automation which supports Java and it is mostly used so we have developed using Java.
- And generally every test engineer knows about Java so it is easy to develop and implement. Java supports object oriented phenomena and I have used oops concepts in my frame work.
- That is while implementing Page Object Model I have used the " Encapsulation " concept where I have used " Data hiding" process by using private keyword to declare the web element and to use that I have used the getter methods for accessing each element.
- While implementing generic utility I have used the concept of "Inheritance " by extending the BaseClass (Generic Utility) in the Test Script.
- While implementing generic utility only I have used the "Abstraction " by implementing the ITestListener using the implement keyword in Test Script. Where I have overridden the method "afterFailed " written a implementation.
- While implementing generic utility I have used the "Polymorphism " concept for launching the browser in base class here I have created method Which will accept browser value and decides at run time which browser it should launch.

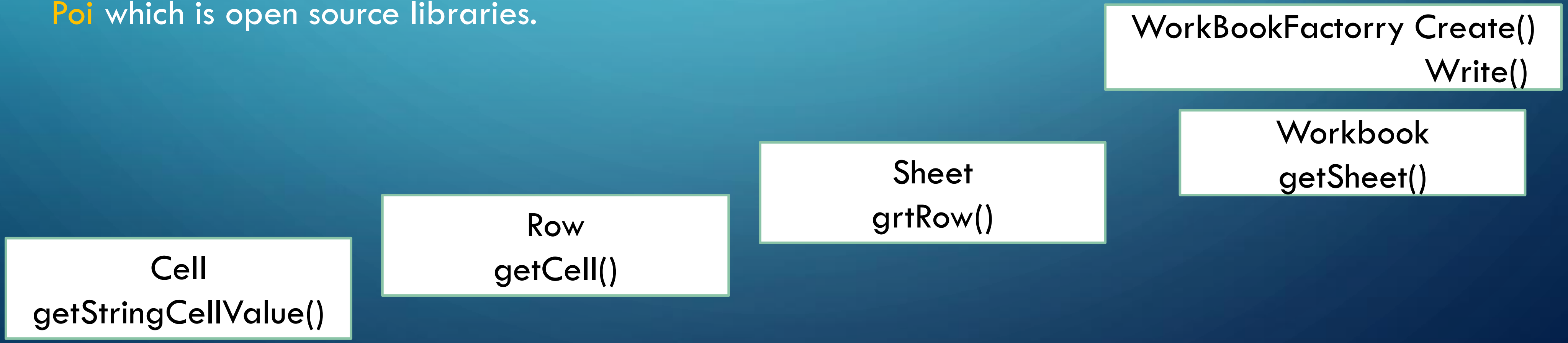
Design Pattern (Type of Framework) :

As I discussed in previous slide that I have used hybrid type to design my framework which is a combination of multiple framework likes Data driven , Keyword driven ,POM ,TestNG and method driven framework.

Lets discussed one by one in details :

- Data Driven Framework : (DDT)

By using DDT we avoid hard coding of scripts here we fetch the data from external files like excel file ,text files , db , XML ,JSON and etc. In my framework I have used Excel file to store the Test Data that is data like as in my project our application consist of many modules and each modules having number of component hence to check the functionality of each component by sending different kind of data inside that to verify functionality, for components like Campaign Name text field ,Organisation Name Text field , Product Name Text Filed and etc .To fetch the data from excel file we need to take the help of Apache Poi which is open source libraries.



WorkBookFactorry Create()
Write()

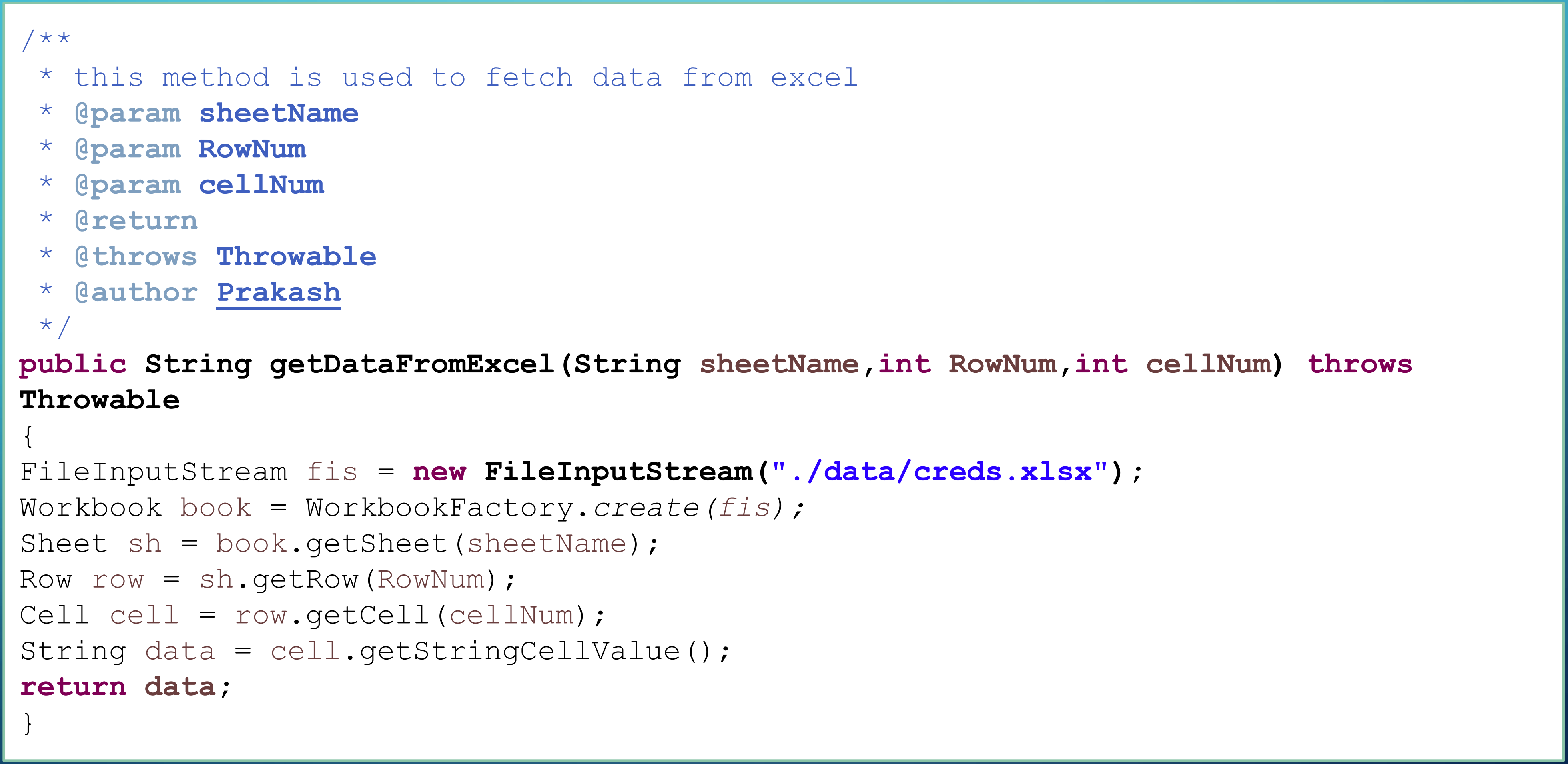
Workbook
getSheet()

Sheet
grtRow()

Row
getCell()

Cell
getStringCellValue()

Below I have taken one example of data driven approach that I have used in my framework which helped in avoiding hard code here we need to pass the relative path as argument while creating the object of FileInputInputStream class which is imported from java.io package..

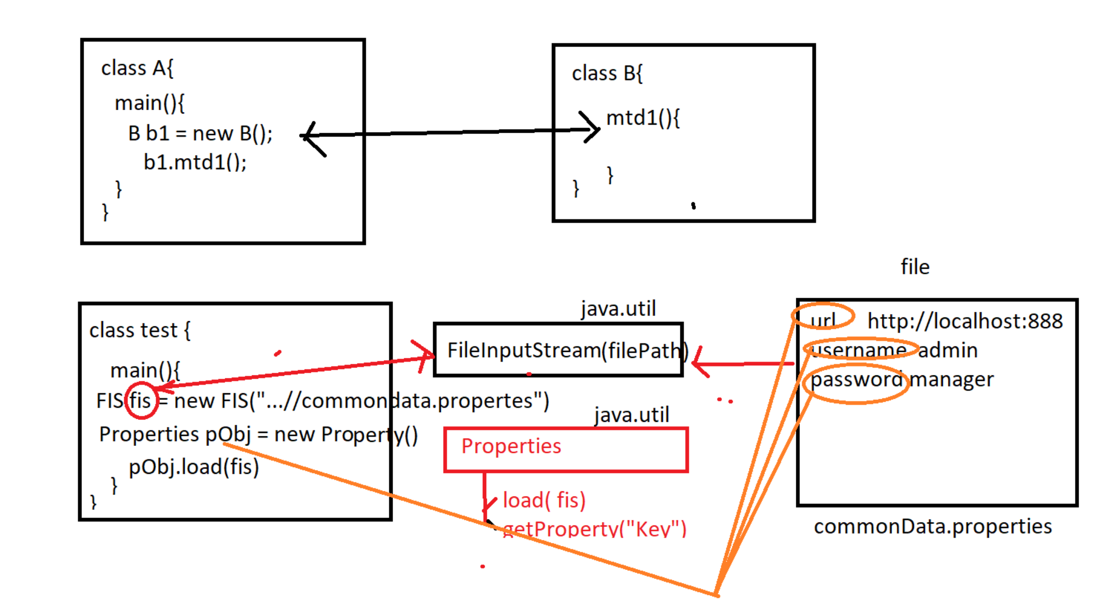


```
/**
 * this method is used to fetch data from excel
 * @param sheetName
 * @param RowNum
 * @param cellNum
 * @return
 * @throws Throwable
 * @author Prakash
 */
public String getDataFromExcel(String sheetName,int RowNum,int cellNum) throws
Throwable
{
FileInputStream fis = new FileInputStream("./data/creds.xlsx");
Workbook book = WorkbookFactory.create(fis);
Sheet sh = book.getSheet(sheetName);
Row row = sh.getRow(RowNum);
Cell cell = row.getCell(cellNum);
String data = cell.getStringCellValue();
return data;
}
```

❖ KeyWord Driven Framework :

We use this approach to achieve the data reusability that is to store the common data which can be frequently used to during execution of every test scripts such as which kind of browser we are going to use ,and the URL of application which is under testing ,and the Valid credential which will require during perfoming testing In this approach we fetch the data from property files where all the data is stored in Key and Value form .Whenever we want to fetch the data we just need to pass the Key and it returns the value that we can be used in Scripts .

In my project our application is access though localhost using URL which is stored in property file whose extension "FileName.properties". Property file is light weight & faster to read the data compare to any other file ,& java has own Class to read the data from property.To fetch the data from property file we need create object of "Properties" Class which is present in "java.util" package. By creating Object of "Properties" using that object we use method as " load("FileObject") " which accepts object reference as argument . And another method "getProperty("Key") " which accepts "Key" as argument and returns "Value"



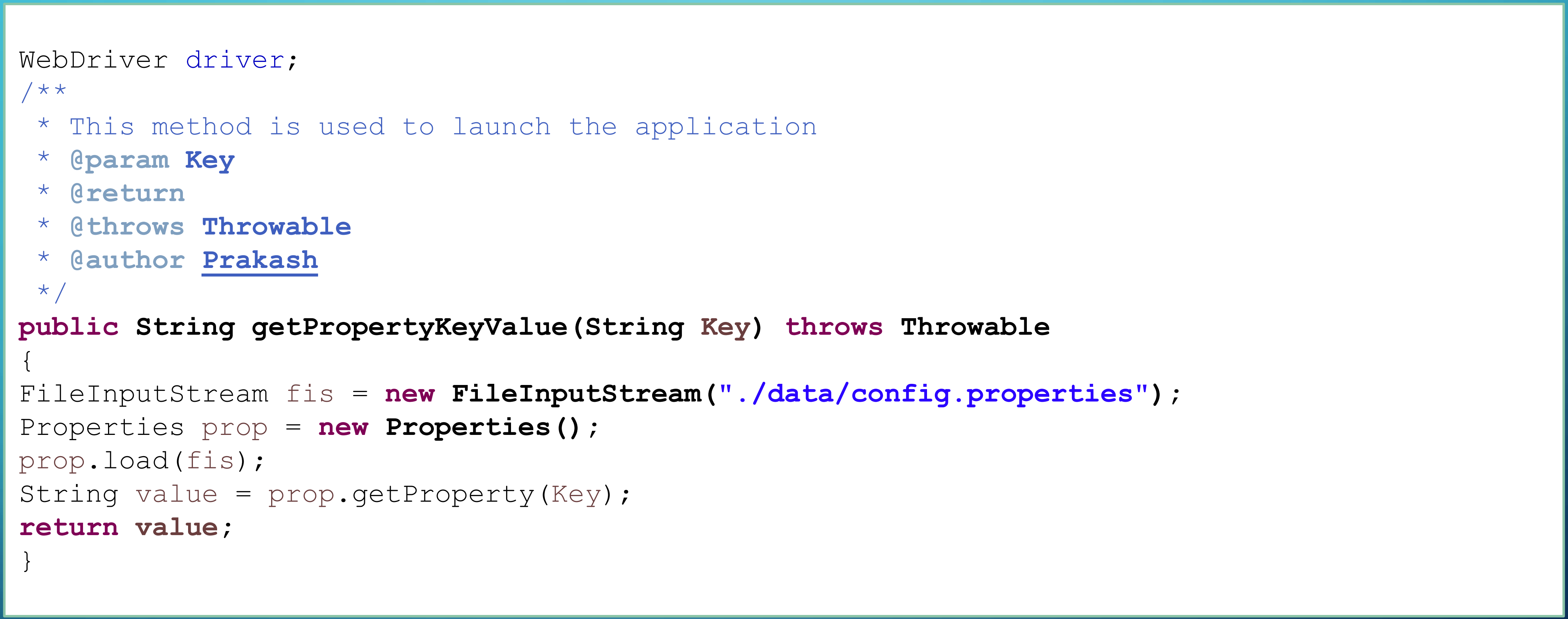
```
class A{
  main(){
    B b1 = new B();
      b1.mtd1();
  }
}
```

```
class B{
  mtd1(){

  }
}
```

```
class test {
  main(){
 FIS fis = new FIS(".../commondata.propertes")
 Properties pObj = new Property()
    pObj.load(fis)
  }
}
```

```
url      http://localhost:888
username admin
password manager
```

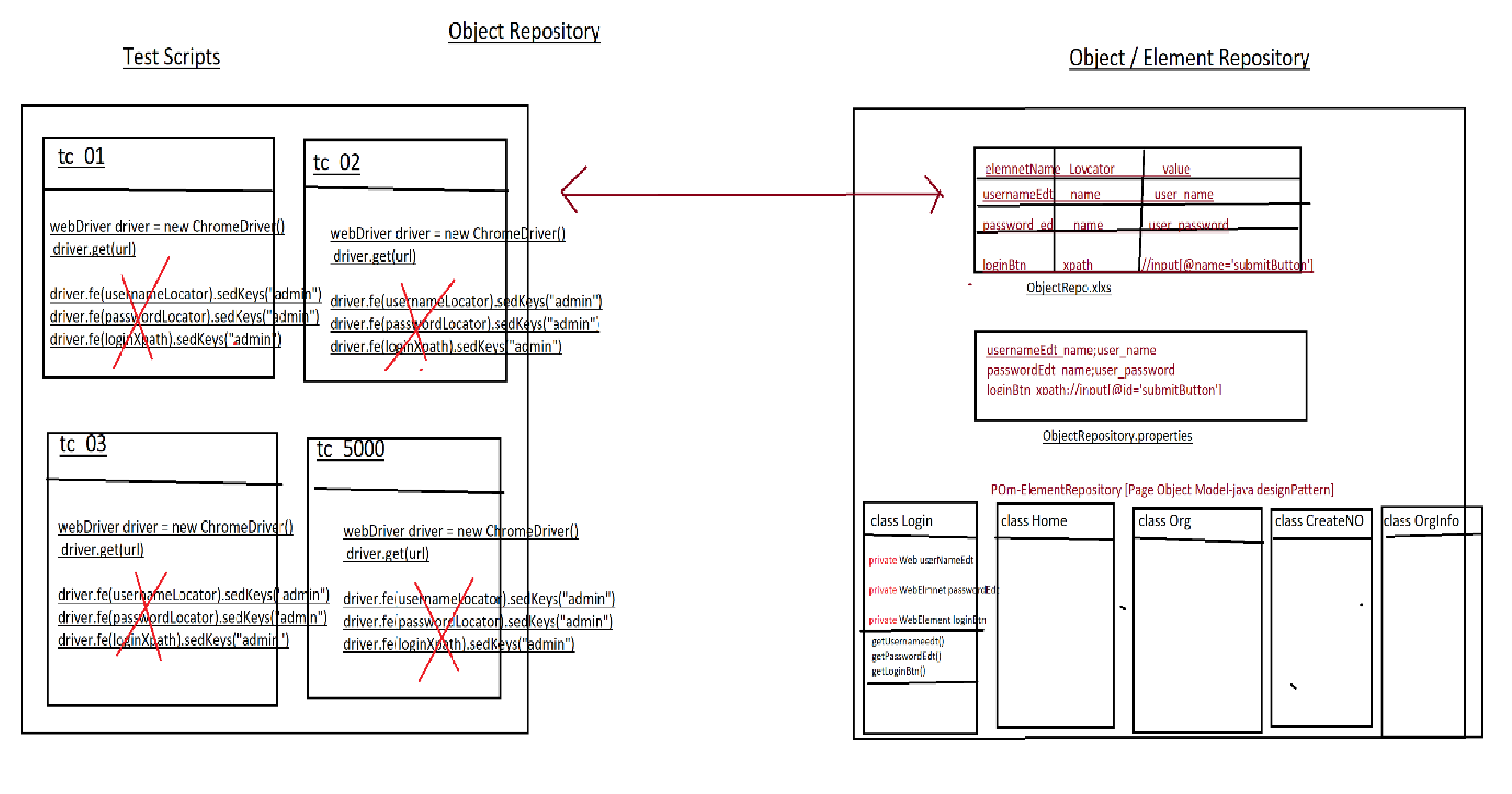
Below I have taken one example of Key driven approach that I have used in my framework which helped in avoiding hard code here we need to pass the relative path as argument while creating the object of FileInputInputStream class which is imported from java.io package..



```
WebDriver driver;
/**
 * This method is used to launch the application
 * @param Key
 * @return
 * @throws Throwable
 * @author Prakash
 */
public String getPropertyKeyValue(String Key) throws Throwable
{
FileInputStream fis = new FileInputStream("./data/config.properties");
Properties prop = new Properties();
prop.load(fis);
String value = prop.getProperty(Key);
return value;
}
```

❖Page Object Model :

Page object model is one of the main component of framework which act as object repository which contains all Web Elements of webpage and business libraries in single class which helps to avoid hard coding of locators in test scripts and reduce code redundancy in scripts. As shown in below diagram.



tc 01 / tc 02 / tc 03 / tc 5000:

```
webDriver driver = new ChromeDriver()
 driver.get(url)

~~driver.fe(usernameLocator).sedKeys("admin")~~
~~driver.fe(passwordLocator).sedKeys("admin")~~
~~driver.fe(loginXpath).sedKeys("admin")~~
```

ObjectRepo.xlxs:

elemnetName	Lovcator	value
usernameEdt	name	user_name
password_ed	name	user_password
loginBtn	xpath	//input[@name='submitButton']

ObjectRepository.properties:

```
usernameEdt name;user_name
passwordEdt name;user_password
loginBtn xpath://inputI@id='submitButton'l
```

class Login:

```
private Web userNameEdt
private WebElmnet passwordEdt
private WebElement loginBtn
getUsernameedt()
getPasswordEdt()
getLoginBtn()
```

Advantages of POM :

Here are some advantages of Page Object Model

1. POM class follows pom java pattern which provides well organise structure.
2. POM class helps to handle “StaleElementException” which we will discuss further in Exception.
3. This helps in maintaining and modification of element easily.
4. Here we directly store web elements in java class
5. Due to flexibility of POM class which is fit for Agile Methodology.
6. Here we use some annotations which are provided by selenium.support.org file.
7. It provides auto healing .
8. Test Scripts become more robust
9. Provides more readability.
10. Development becomes faster.
11. Modification is easy even after GUI changes

Implementation of POM in Framework :

In my framework I have implemented POM . In that I have created special package for POM class as Object Repository .
Object Repository contains all POM class that is , I have created separate pages class for each webpage of application and each class contains all the Web Elements of particular Webpage and business logic at single place so that whenever we need any particular web element then we just need to create Object of that POM class and by using reference we can use that element through specific business logic.
I have designed some POM classes such as :

- LoginPage
- HomePage
- ProductPage
- OrganisationPage
- CampaignPage and so on.

Business Logic is nothing but the operation we want to perform on particular web element.

To create POM class we need to follow some rule.

Rules for POM class

1. Page class name should be same as class name

2. There should be declaration all the web element as private using @FindBY annotations

3. It should have getter methods to access the private web element

4. For Initialization of page there should be constructor which accepts driver as argument

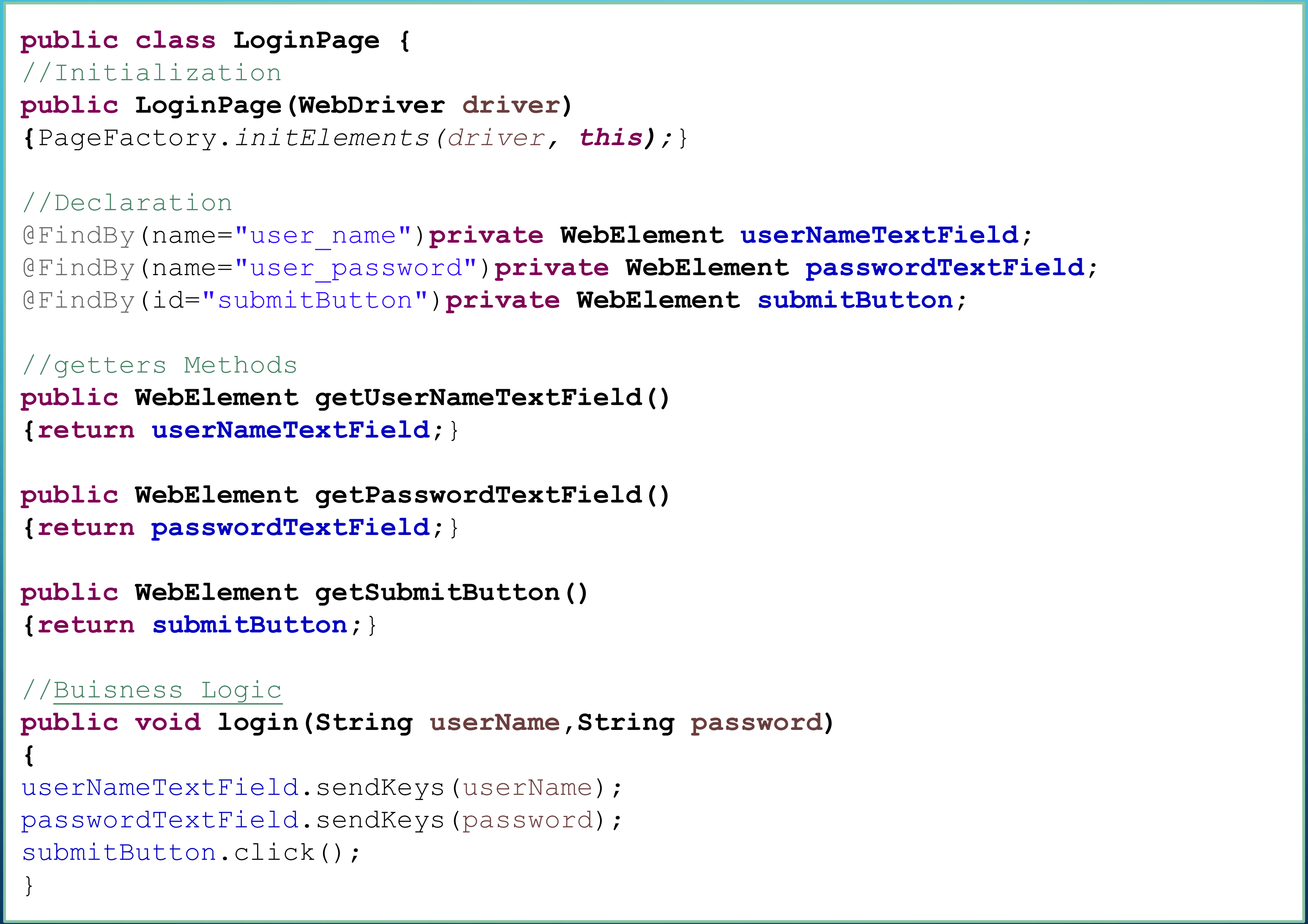
and there should be one mehod in constructor

i.e. initElement(driver,this) which is static method of PageFactory class.

5. After Declaration , Initialization ,and Getter Method we have to create Business logic

To use and perform operation on that particular webelement.

Here is example of POM class that I have created for my framework. Here I have created POM class for LoginPage of our application .



```
public class LoginPage {
//Initialization
public LoginPage(WebDriver driver)
{PageFactory.initElements(driver, this);}

//Declaration
@FindBy(name="user_name")private WebElement userNameTextField;
@FindBy(name="user_password")private WebElement passwordTextField;
@FindBy(id="submitButton")private WebElement submitButton;

//getters Methods
public WebElement getUserNameTextField()
{return userNameTextField;}

public WebElement getPasswordTextField()
{return passwordTextField;}

public WebElement getSubmitButton()
{return submitButton;}

//Buisness Logic
public void login(String userName,String password)
{
userNameTextField.sendKeys(userName);
passwordTextField.sendKeys(password);
submitButton.click();
}
```

Testing Framework

In my framework I have used the testNG framework which is testing framework which can be used for Unit ,Integration ,End to end testing and functional testing which is also tool which is open source and next generation testing tool.

□ Advantages of testNG :

□ By using testNG in framework we getbthe result in HTML formats.

□ By using testNG we can execute the test script in Batch.

□ TestNG provides annotations as follows :

@BeforeSuite @Test @AfterMethod @dataProvider
@BeforeTest @AfterClass
@BeforeClass @AfterTest
@BeforeMethod @AfterSuite

□ TestNG provides different flags which helps to control the execution of scripts :

groups -- used to perform group execution of test script as per requirement
priority – used to execute the test script on priority basis
dependsOnMethod – this flag is used to execute the script which is depend on another script/test
invocationCount – this flag is used to execute the particular test script for number of times
alwaysRun -- this flag is used to run the script always whether other test script got filed or not.
enabled – this flag is used to provide restriction on particular test script if we assign
enabled=true ---execute
enabled=false---does not execute

Continued…….

- ❑ TestNG allows us to take screenshot dynamically whenever testscript get failed
- ❑ TestNG provides assertion which is used for validation.

  1.HardAssert---which used for critical features

  2.SoftAssert ---which used for basic features
- ❑ TestNG allows to perform cross browser execution using parameter flag here we can execute scripts using different browser at a time.
- ❑ We can also perform parallel execution executing scripts using multiple thread.
- ❑ It also provides data driven approach here we can read data using “dataprovider” from .xml file
- ❑ Here we can re-execute failed scripts

❖ Insertion of testNG in Framework

In my framework I have inserted testNG as in BaseClass I have used annotation like

@BeforeSuite which followed by method which is used for opening the connections such as data base (JDBC)

@BeforeTest which is followed by method where I have defined the mode of execution of testscripts i.e cross /parallel.
@BeforeClass followed by method where logic written for which browser has to launch

@BeforeMethod followed by method where the logic related to login.

After executing above code actual test script start to execute in which main method is replaced with annotation @Test which is followed by method where all the actual implementation script is there ..
Once the test script get executed …then again control goes to BaseClass now the ..execution starts

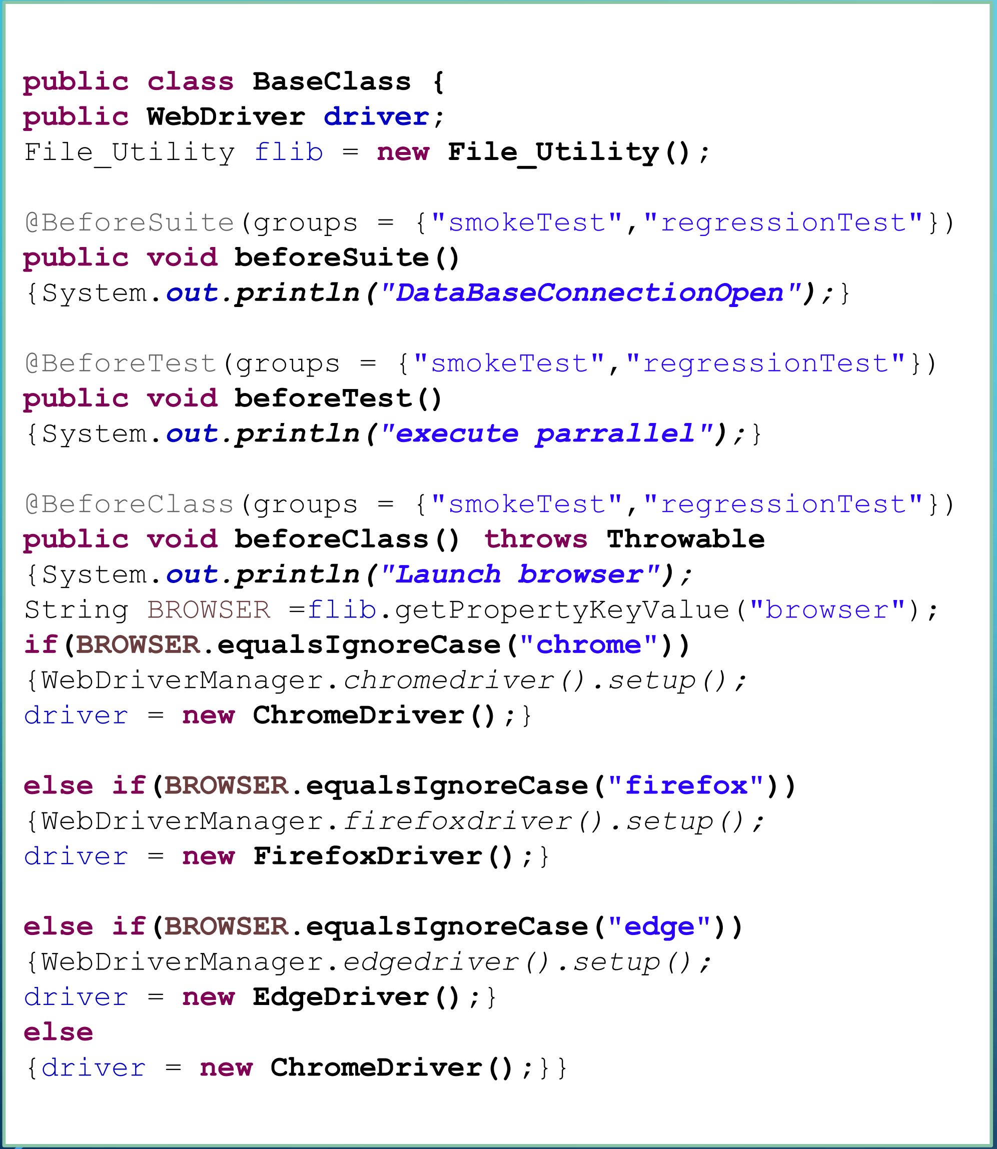
@AfterMethod followed by method which has code related to Logout .

@AfterClass followed by method which has code related to closing the browser.

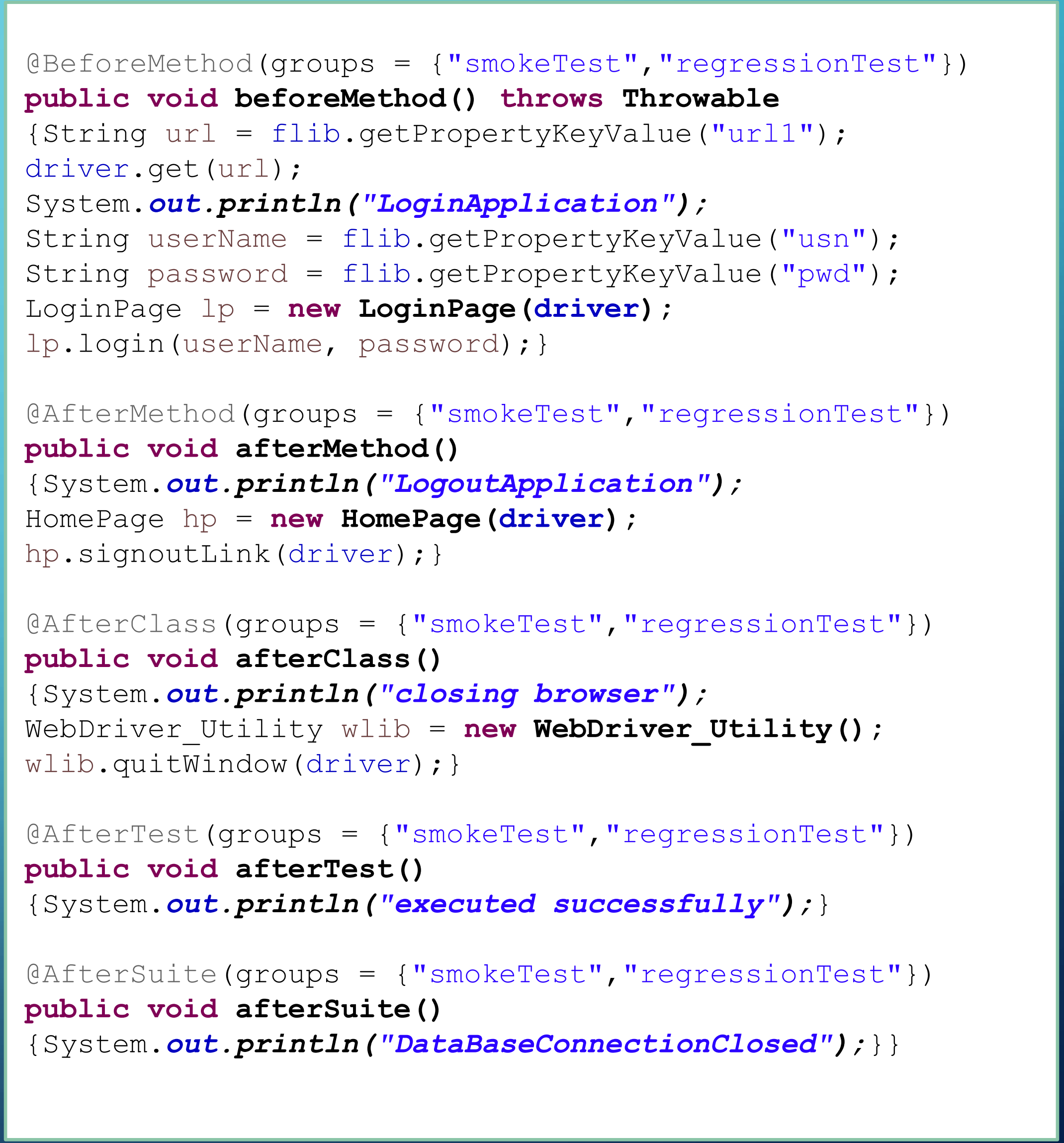
@AfterTest followed by method which has code related to verify execution

@AfterSuite followed by method which has code related to closing the connection of database.

Here is example of I have inserted testNG in my framework in BaseClass which is extended by scripts



```
public class BaseClass {
public WebDriver driver;
File_Utility flib = new File_Utility();

@BeforeSuite(groups = {"smokeTest","regressionTest"})
public void beforeSuite()
{System.out.println("DataBaseConnectionOpen");}

@BeforeTest(groups = {"smokeTest","regressionTest"})
public void beforeTest()
{System.out.println("execute parrallel");}

@BeforeClass(groups = {"smokeTest","regressionTest"})
public void beforeClass() throws Throwable
{System.out.println("Launch browser");
String BROWSER =flib.getPropertyKeyValue("browser");
if(BROWSER.equalsIgnoreCase("chrome"))
{WebDriverManager.chromedriver().setup();
driver = new ChromeDriver();}

else if(BROWSER.equalsIgnoreCase("firefox"))
{WebDriverManager.firefoxdriver().setup();
driver = new FirefoxDriver();}

else if(BROWSER.equalsIgnoreCase("edge"))
{WebDriverManager.edgedriver().setup();
driver = new EdgeDriver();}
else
{driver = new ChromeDriver();}}
```



```
@BeforeMethod(groups = {"smokeTest","regressionTest"})
public void beforeMethod() throws Throwable
{String url = flib.getPropertyKeyValue("url1");
driver.get(url);
System.out.println("LoginApplication");
String userName = flib.getPropertyKeyValue("usn");
String password = flib.getPropertyKeyValue("pwd");
LoginPage lp = new LoginPage(driver);
lp.login(userName, password);}

@AfterMethod(groups = {"smokeTest","regressionTest"})
public void afterMethod()
{System.out.println("LogoutApplication");
HomePage hp = new HomePage(driver);
hp.signoutLink(driver);}

@AfterClass(groups = {"smokeTest","regressionTest"})
public void afterClass()
{System.out.println("closing browser");
WebDriver_Utility wlib = new WebDriver_Utility();
wlib.quitWindow(driver);}

@AfterTest(groups = {"smokeTest","regressionTest"})
public void afterTest()
{System.out.println("executed successfully");}

@AfterSuite(groups = {"smokeTest","regressionTest"})
public void afterSuite()
{System.out.println("DataBaseConnectionClosed");}}
```

Dependency Manager :

In my framework I have used the MAVEN framework to manage the dependencies in pom.xml file
This helps to reduce the hectic job of downloading the jar files for different application and to manage
Their updates here we just need to add the dependency in pom.xml file with required version we just
need to save that file it automatically downloads all required files. So ultimately it helps make test
Engineers life easy.

Here is sample code for pom.xml which is used in my framework:

```
<dependencies>
 <dependency>
   <groupId>org.seleniumhq.selenium</groupId>
     <artifactId>selenium-java</artifactId>
   <version>4.1.2</version>
 </dependency>
 <dependency>
  <groupId>org.testng</groupId>
   <artifactId>testng</artifactId>
  <version>6.9.10</version>
  <scope>test</scope>
 </dependency>
<dependencies>
```

Build Tool : (CI/CD tool)

- In my project we used the build tool which is JENKINS which is open automation source server.
- Jenkins continuous Integration and Continuous development tool where we used to convert all our error free code to build by using various plugins .
- The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.
- which enables developers around the world to reliably build, test, and deploy their software.
- Jenkins uses stages to run a collection of steps

Version control : (Common Repository)

- For version control we have used the GitHub repository where I used to push my build /code and pull the code/build for changes.
- This helps to continuous development of build whenever I want to do some changes in build which was push by other Test Engineer then I did not need to contact that person I can directly access it and pull it system to do required changes.
- We used to store the our scripts in the GitHub because it is common repository.
- This save the most time in designing framework.

Exceptions

- IllegalStateException → (Java Exception/unchecked) :
  This exception occurs when the path of driver executable is not set by default .

- Interupted Exception → (Java Exception/checked) :
  This exception comes whenever we use Thread.sleep

- NoSuchElementException → (selenium/Unchecked) :
  We get this exception whenever locator is unable to find element on webpage.

- InvalidSelectorException → (Selenium/Unchecked) :
  We get this exception whenever there is a syntax error in cssSelector or xpath .

- TimeOutException →(Selenium/Unchecked) :
  We get this exception after the timeout specified in explicit wait get overs.

- IndexOutOfBoundException → (Java/unchecked) :
  we get this exception whenever we use (i<=count) instead of (i<count) in the for loop for Array iteration.

- UnExpectedTagNameException → (selenium/unchecked) :
  we get this exception whenever we pass the webelement to the select class constructor which is not a type of List Box

Continued….

- UnsupportedOperationException →(Selenium/unchecked)

  we get this exception whenever we try to use deselect methods on single list box .
- NoAlertPresentException → (Selenium/unchecked)

  we get this exception whenever we try to perform action on the alert pop-up ,when the popup itself is not present .
- UnHandledAlertException →(Selenium/unchecked)

  we get this exception whenever we try to perform action on the browser without handling the alert popup
- WebDriverException → (Selenium/unchecked)

  we get this exception whenever we pass relative path to the sendKeys() method.
- InvalidArgumentException → (selenium/unchecked)

We get this exception whenever we pass wrong path of the file.

- AWTException → (Java/checked)

we get this exception whenever we use robot class.

- IOException → (java/checked)

we get this exception whenever we use runtime class.

- StaleElementReferenceException → (selenium/unchecked)

we get this exception whenever we try to access any webelement whose address got old whenever page get refreshed.

- NosuchFrameException →(selenium/unchecked)

  we get this exception whenever we pass invalid argument to the frame method