1. Modular Framework:

Modular framework is the approach where all the test cases are first analyzed to find out the reusable flows. Then while scripting, all these reusable flows are created as functions and stored in external files and called in the test scripts wherever required

Advantages:

-Test scripts can be created in relatively less time as the reusable functions need to be created only once

-Effort required to create test cases is also lesser due to code reuse

-if there are any changes in the reusable functions, the changes need to be done in only a single place. Hence script maintenance is easier

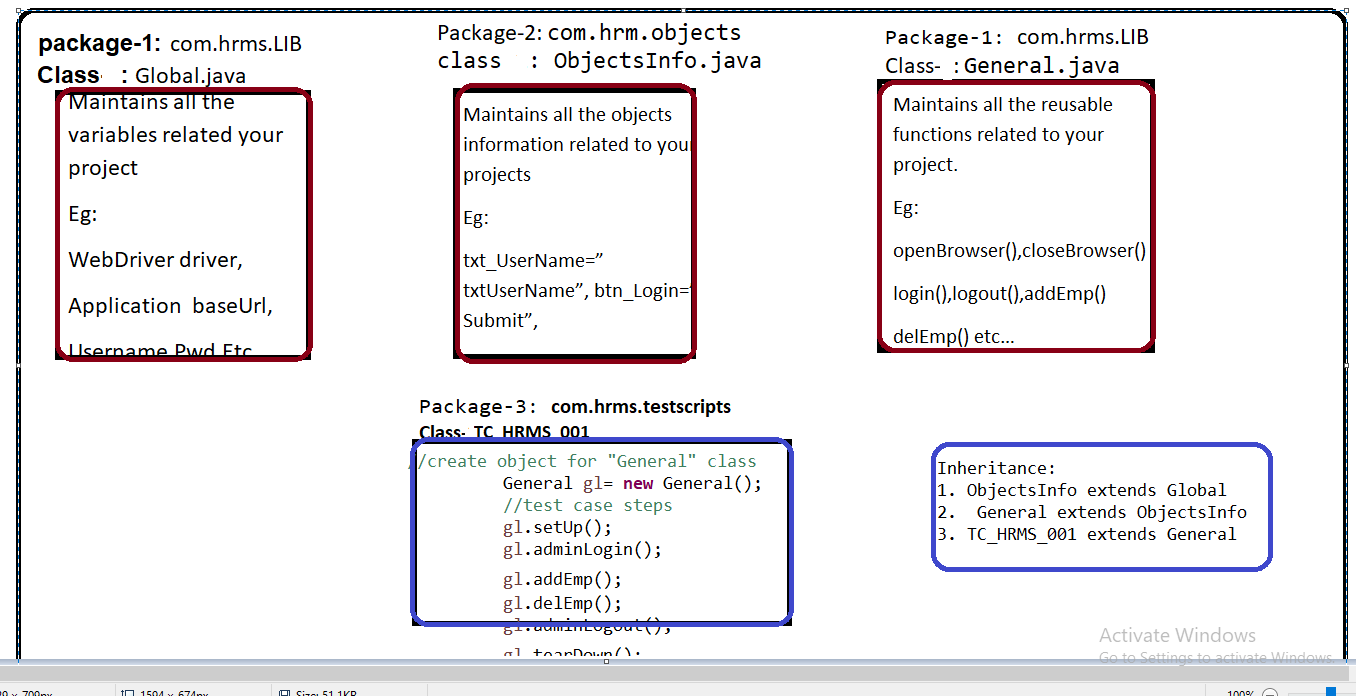
Disadvantages of Modular Framework:

-since data is still hard coded in the script, the same test case can’t be run for multiple data values without changing data after each run

-additional time is spent in analyzing the test cases to identify with reusable flows

-good programming knowledge is required to create and maintain function libraries

Designing the Modular Framework from scratch:



| Project | HRMS\_ModularFrameWork |  |
| --- | --- | --- |
| Package | com.hrms.LIB |  |
|  | Global.java | Maintains all the variables related your project  Eg:  WebDriver driver,  Application baseUrl,  Username.  Pwd  Etc… |
|  | General.java | Maintains all the reusable functions related to your project.  Eg:  openBrowser()  closeBrowser()  login()  logout()  addEmp()  delEmp()  etc… |
| Package | com.hrms.objects |  |
|  | ObjectsInfo.java | Maintains all the objects information related to your projects  Eg:  txt\_UserName=”txtUserName”  btn\_Login=”Submit”  link\_logout=”Logout”  etc… |
| Package | com.hrms.testscripts | All the actual test cases need to written in this package only |
|  | TC\_HRMS\_001 | Call the required methods/functions from “General.java” class |
|  | TC\_HRMS\_002 |  |
|  | TC\_HRMS\_003 |  |

Step 1: create new Java project “MadhukarQAIT\_Modular\_Framework” then Configure WD jars and provide browser server files in “Drivers” folder

Step 2: Create “com.hrms.LIB” package and create class “Global.java”

Maintain all the variables and data related to your project

Note:

Package is a group of classes. If we create one package in Eclipse it will be considered as one folder in your eclipse workspace

Each class in a package considered as one file (.java)

----------------------------------------------------------------------------

**package** com.hrms.LIB;

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

**public** **class** Global {

//Variables info

**public** WebDriver driver;

**public** String baseUrl="https://opensource-demo.orangehrmlive.com/";

**public** String uid="Admin";

**public** String pwd="admin123";

}

----------------------------------------

Step 3: Create “com.hrm.objects” package and create class “ObjectsInfo.java”

Maintain all the objects information related to your projects

**package** com.hrms.objects;

**import** com.hrms.LIB.Global;

**public** **class** ObjectsInfo **extends** Global{

//Objects

**public** String txt\_username="txtUsername";

**public** String txt\_password="txtPassword";

**public** String btn\_login="Submit";

**public** String link\_Welcome\_Admin="Welcome Admin";

**public** String link\_logout="Logout";

}

Step 4:create “General.java” Class in “com.hrms.LIB” package

Create following methods:

1. setup(): to initialize browser
2. teardown(): to close application
3. adminLogin(): to Admin login
4. adminLogout(): to admin logout
5. addEmp(): to add employee
6. deleteEmp(): to delete employee

-------------------------------------------

**package** com.hrms.LIB;

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

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** com.hrms.objects.ObjectsInfo;

**public** **class** General **extends** ObjectsInfo{

**public** **void** setUp() {

System.*setProperty*("webdriver.chrome.driver", "./Drivers//chromedriver.exe");

driver=**new** ChromeDriver();

driver.get(baseUrl);

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(30, TimeUnit.***SECONDS***);

System.***out***.println("Application opened");

}

**public** **void** tearDown() {

driver.quit();

System.***out***.println("Application and Browser are closed");

}

**public** **void** adminLogin() **throws** InterruptedException {

driver.findElement(By.*name*(txt\_username)).sendKeys(uid);

driver.findElement(By.*name*(pwd)).sendKeys(pwd);

driver.findElement(By.*name*(btn\_login)).click();

Thread.*sleep*(5000);

}

**public** **void** adminLogout() **throws** InterruptedException {

driver.findElement(By.*linkText*(link\_Welcome\_Admin)).click();

Thread.*sleep*(3000);

driver.findElement(By.*linkText*(link\_logout)).click();

System.***out***.println("Admin logout");

}

**public** **void** addEmp() {

System.***out***.println("Added new Employee");

}

**public** **void** delEmp() {

System.***out***.println("Deleted employee");

}

}

------------------------------------

Step 5: create package “com.hrms.TestSCript” and call the require test script from “General” class

**package** com.hrms.TestScripts;

**import** com.hrms.LIB.General;

**public** **class** TC\_HRMS\_001 {

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

//create object for "General" class

General gl= **new** General();

//test case steps

gl.setUp();

gl.adminLogin();

gl.addEmp();

gl.delEmp();

gl.adminLogout();

gl.tearDown();

}

}

