1) Selenium :-

Selenium is open-source tool that Automated web browsers. It Provides a single Interface that lets you write test scripts in Programming languages. like Java, C#, PHP, Python and others.

---

2) Locators

locating elements in selenium webdriver is performed by the help of findElement() and findElements() methods Provided by webdriver.

Locator are the way to identify the HTML element on a web page. we can use locators to Perform actions on the text boxes.

id(), name(), classname(), XPath, link text(), Partial link text, CSS selector, tagname.

---

3) XPath - Extensible Markup language.

If the elements are not found by the general locators like id, name, class name etc.. then XPath is used to find an element on the webPage.

Validate XPath : absolute xPath

Relative XPath

1. It is the direct way to Find the element, but the disadvantage of the A. XPath is that if there are any changes made in the Path of the element.

Listeners

Listeners listen to the events that we desired in our script. We can use listeners for reporting.

* whenever any test case fails we can take its screenshot.

* I have used ITestListener which has different methods like onStart, onFinish, onTest, onTest failure etc.

How to Rerun the failed test cases

1. We can run testng-failed.xml
2. Using IRetryAnalyzer Interface
3. Using IAnnotationTransformer (I) and IRetryAnalyzer transform().

Retry.

Custom exception

Custom exception is creating your own exception class and throwing that exception using the 'throw' keyword.

Ex

My Exception in the below code extends the Exception class.

Why we say JAVA is object oriented language?

* In JAVA everything is object based and by creating the objects of classes and by their references we can communicate with the data members and their functions and its supports Pillers of oops - Inheritance, Polymorphism, Encapsulation, Abstraction.

```
int a[] = {2,4,6,8,2};
int temp;
for (int i=0; i<a.length; i++){
 for(int j=i+1; j<a.length; j++) {
    if (a[i] > a[j])
  {
     temp= a[i];
     a[i] = a[j];
     a[j] = temp;
  }}
 for (int k=0; k<a.length; k++){
    System.out.(a[k]+" ");
}}}
```

New keyword

* The new keyword in JAVA instantiates a class by allocating desired memory for an associated new object.

* It then returns a reference to that memory.

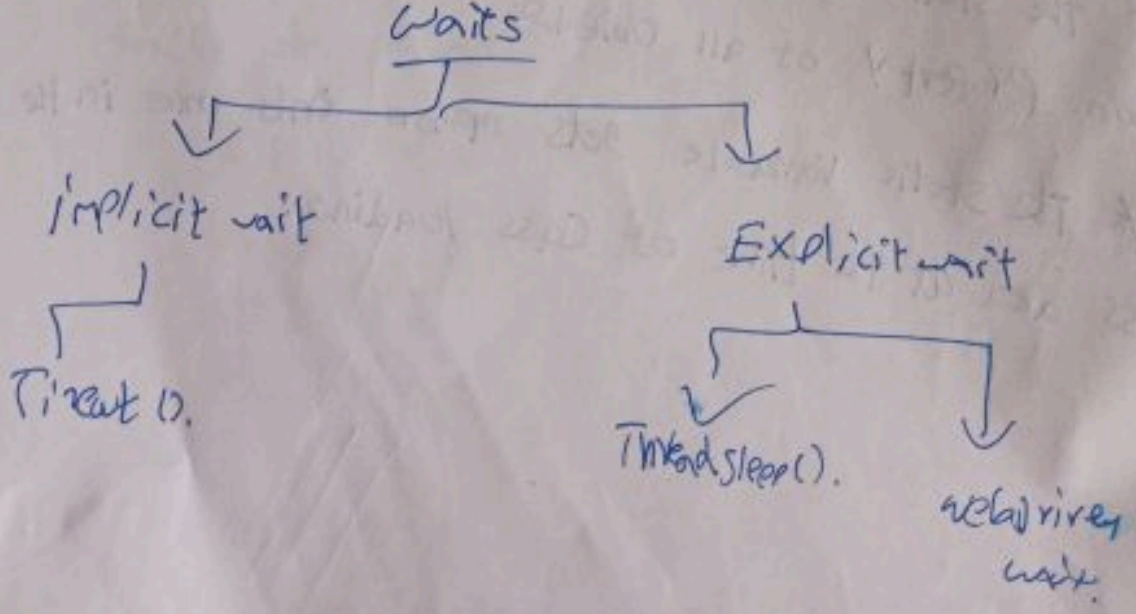
* The new keyword is followed by a call to a Constructor, which instantiates the new object.

* The JAVA new keyword is used to create an instance of the class.

Implicity Wait - directs the selenium webdriver to wait for a certain measure time before throwing the exception. Once this time is set, webdriver will wait for the element before the exception occurs.

Explicit wait -

Fluent wait - Fluent wait in selenium marks the maximum amount of time for selenium webdriver to wait for certain condition before



Selenium

28-9-2021

Selenium IDE ->firefox only use
Selenium RC
Selenium webdriver -> we use
Grid ->[onetime vera vera use pannalam] ex window chrome

~~firefox to system use agum~~

How to download Selenium

[~~Jar file~~ build Ref library ->]

Code --- Driver --- Browser

Steps:

Download the jar file
Configure to jar file
Download the driver exe file

Steps:-

Create a java project
Create a new folder [rc Project -->new -->folder -->lib]
Copy and Paste the jar in lib folder from downloads
rc jar -->build path --->add to build Path
Create a new folder [rc Project -->new -->folder -->drivers]
Extract the downloaded driver zip folder
copy and Paste the exe file in driver folder from downloads
create a class

Key:-

chrome web driver . chrome. driver
gecko - webdriver. gecko. driver
ie-webdriver. ie .driver

[gecko-firefox inoru name]

class:-

Chrome - chromeDriver
gecko - FirefoxDriver
ie - InternetExplorerDriver

Locators

Locating elements in Selenium WebDriver is performed with the help of findElement() and findElements() methods provided by WebDriver and WebElements Interface.

WebDriver and WebElement :-I

findElement()-m

returns a webElement object based on a specified search criteria or ends up throwing an exception if it does not find any element matching the search criteria.

findElements()-m

returns a list of webElements matching the search criteria. If no elements are found, it returns an empty list.

By - Abstract class this class having all locators methods in static

===

id()-m

name()-m

className()-m

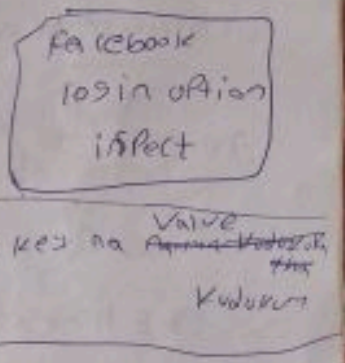
xpath()-m

tagName()-m - purpul color name font la irundha

linkText()-m

PartialLinkText()-m

cssSelector()-m



webElement :-I

sendKeys -m - --To send / Pass the values to text boxes

click()-m --to click any link, image, button, radio button, check box in webpage

<input type="email">

input - Tag Name type - Attribute Name email = Attribute Value

30-9-2021

XPath : XML Path ---Extensible Markup language

1. If the elements are not found by the general locators like id, class, name, etc. then xPath is used to find an element on the web page.

2. Validate XPath : (/) absolute xPath

--> It is the direct way to find the element, but the disadvantage of the absolute xPath is that if there is are any changes made in the Path of the element then that xPath gets failed.

--> The key characteristic of xPath is that it begins with the single forward slash (/), which means you can select the element from the root node.

Relative xPath : (//)

--> Relative xPath starts from the middle of HTML Dom structure. It starts with double forward slash (//). It can search elements anywhere on the webPage.

--> Relative xPath is always preferred as it is not a complete Path from the root element.

XPath Syntax :-

//tagName [@attributename='attributevalue']

XPath index Syntax :-

(//tagName [@attributename = 'attribute value']) [index]

1-10-2021

text:

//tagname[text()='text value']

// Contains:

//tagname[contains(text(),'Partial text')]

//span[contains(text(),'create')]

//span[contains(text(),'in')]

//span[contains(text(),'Account')]

getText() - - method:

Used to print the already existing text in webpage

getAttribute("value") - - method

Used to print to corresponding attribute values.

Value -- user inputs are automatically stored in "value" Attribute.

Program

```
Public Class xPath Test{
main();
System.setProperty(. . . );
webDriver driver = new ChromeDriver();
driver.get("https://en-gb.facebook.com");
```

Actions ---- class 5-10-2021

Actions class is selenium for handling keyboard and mouse events.

1. To do mouse over action:

moveToElement() -- m

Syntax:

Actions obj = new Actions (webdriver ref);
Obj.moveToElement(webele ref).perform();

2. To do drag and Drop:

dragAndDrop() --- m

Syntax:

Actions Obj = new Actions (WebDriver ref);
Obj.dragAndDrop (source, Destination).perform();

Action -- class

moveToElement() → move Pani work Panam
dragAndDrop() →
Context click()
double click()

Program:-

```
Public Class Actions {
    main { )
    System.setProperty
    webdriver
    driver.get ("http://www.sreentech");
driver.manage().timeouts().implicitlyWait (20, Timeu
driver.manage().window().maximize();
```

Robot - Class

6-10-2021

VK - Virtual key

Mouse right click
Panni DropDown
use asum

Program:-

```
Public class {

// Actions a = new Actions (driver);
// Robot r= new Robot ();
// WebElement txtusername = driver.findElement (By.id ("email"));
// txtusername.sendkeys ("Dinesh");

//// Double Click using Action
// a.doubleclick (txtusername).perform();

Right click Using Action
// a.contextclick (txtusername).perform();

// r.keyPress (keyEvent.VK_DOWN);
// r.keyRelease (keyEvent.VK_DOWN);

// r. keyPress (keyEvent.VK_DOWN);
// r.keyRelase (keyEvent.VK_DOWN);
// r.keyPress (keyEvent.VK_DOWN);
// r.keyRelase (keyEvent.VK_DOWN);
// r.keyPress (keyEvent.VK_ENTER);
// r.keyRelase (keyEvent.VK_ENTER);

// WebElement txtPassword = driver.findElement (By.id ("Pass"));
// a.contextClick (txtpassword).perform();

// for (int i=0; i<4; i++){
    r.keyPress (keyEvent.VK_DOWN);
    r.keyRelase (keyEvent.VK_DOWN);
}
```

7-10-

Drop Down html code :-

Used for selecting and deselecting option in a drop down.

```
<select>
<option value="IND">India</option>
<option value="us">United states</option>
</select>
```

Select :

```
selectByIndex(10);
selectByvalue("IND");
selectByVisibleText("India")
```

```
Select ref = Select(WebElement RefName);
```

select : Class

1. selectByvalue() - m
2. SelectByVisibleText() - m
3. selectByIndex() - m
4. getOptions() - m
5. getAllselectedoptions() - m
6. GetFirstselectedOption() - m
7. isMultiple() - m
8. deselectByIndex() - m
9. deselectByvalue() - m
10. deselectByvisibleText() - m
11. deselectAll() - m

DropDown - multiple

8-10-

Program:

```
Public class DropDown2 {
    =
    =
driver.get("https://www.spice          ");
    =
WebElement drop1 = driver.findElement(By.id("multi-select"));
    Select s = new Select(drop1);
    //select index
     S.selectByIndex(1);
   // Select Value
     s.selectByValue("Newyork");
   // select Visible Text
     S. selectBy VisibleText("Texas");

   // IS multiple
     boolean multiple = S.ismultiple();
     System.out.println(multiple);

// ALL options
  List<WebElement> options = S.getOptions();
  System.out.println("---All options--");
  for (WebElement x : options) {
     System.out.println(x.getText());
  }

//
  List<WebElement> a = S.getAllSelected options();
  System.out.println("--- My selected options--");
  for (WebElement y : a) {
     System.out.println(y.getText());
  }
```

4

~~Abort~~ - Alert : Interface

Alert is a small message box displayed on the screen to the users

JAVAScript PopUp

Switching the alert :-

Aleart Yesname = driver. SwitchTo ().alert ();

types :-

1. Simple alert --> Contains only ok button .... (we should accept the alert)
2. Confirm alert --> Contains both ok and cancel button (Either we accept or dismiss this alert)
3. Prompt alert --> Contains text box with OK and Cancel button. to insert "yes" to accept the ale (or) we have give "No" to dismiss the alert

methods in Alert :-

accept () --- m --- to accept the alert
dismiss () --- m --- to dismiss the alert
sendkeys () --- m --- to insert the values
get Text () --- m --- to print the text from the Alert.

Program - 1

```
Public class {
    - - -
    - - -
    - - -
```

1. // simple Alert

```
WebElement a1 = driver.findElement(By.xPath("// button[@id='btn']"));
a1.click ();
// switch to simple alert
Alert targetI = driver. SwitchT
```

TakesScreenshot -- Interface

getScreenshotAS() --- method

Steps :-

typeCast --> TakesScreenshot tk = (TakesScreenshot)driver;

store screenshot in default storage Path

create a new screenshot folder

copy from default Path to new file Path

FileUtils --- class

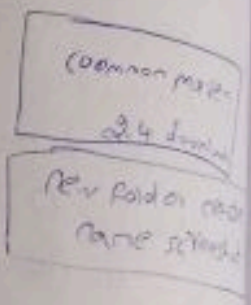
copyFile --- method

common.io.jar

Program :-

```
Public class screenshot {
    ;
    ...
// // Type conversion (or) UP casting
// long a=10;
// int b=20;
// a=b;
// // type (casting) (a) down castings
// int a=10;
// long b=20;
// a=(int)b;

    TakesScreenshot tk = (TakesScreenshot)driver;
    File from = tk.getScreenshotAS(outputType.FILE);
    //System.out.println(from);
    File to = new File("E:\\workspace\\ selenium 8.30 \\...
    FileUtils.copyFile(from,to);
    System.out.println("!--- Done ---!");
```



JAVAScript

exec

insert :

argume

click :-

argum

getAttribute :

return

Scroll down

argument

argument

Program :-

Public

JAVAS

// Send ke

webEle

js.exe

webEle

js.exe

// Get Attri

Object use

System.out

// Conversion

String

13-10-2021

JavaScriptExecutor : Interface - dissault types os operation

executeScript () - m

insert :

arguments [0]. setAttribute ('value', 'green')

click :-

arguments[0] .click()

getAttribute :-

return arguments[0].getAttribute ('value')

Scroll downto:

arguments[0]. scrollIntoView (true) //down
arguments [0] .scrollIntoview (false) //UP

Program :

Public class JS {

JavaScriptExecutor js = (JavaScriptExecutor) driver;

// send keys

WebElement txtusername = driver. findElement (By. id ("email"));
js. executeScript ("arguments[0]. setAttribute ('value', 'Dinesh')", txtUsername);
WebElement txtPassword = driver. findElement (By. id ("Pass"));
js. executeScript ("arguments[0]. setAttribute ('value', '9885')", txtPassword);

// Get Attribute

Object user = js. executeScript ("return arguments[0]. getAttribute ('value')", txtusername)
System. out. println (user);

// converting obj to string (for using Strings methods)

String a = user. toString ();

Kuduvanenna - set
Vangenum na - get

```
// Button click
// WebElement btnLogin = driver.findElement(By.xPath("//button[@name='login']"));
// js.executeScript("arguments[0].click();", btnLogin);

// Scroll down
Thread.sleep(5000);
WebElement help = driver.findElement(By.xPath("//a[text()='Help']"));
js.executeScript("arguments[0].scrollIntoView(true)", help);

// Scroll up
Thread.sleep(5000);
js.executeScript("arguments[0].scrollIntoView(false)", txtUserName);
}
}
```

Navigation Commands

Diff b/t get() & navigate.to()

get()

It is used to go to the particular website, but does not maintain the browser history and cookies.

So we can't use forward and backward button. we click on that, Page will that get scheduled.

Navigate().to()

It is used to go the particular website, but maintain the browser history and cookies

```
System.out.println ("Before select "+gender.isSelected());

Thread.sleep(3000);
gender.click();

System.out.println ("After select "+gender.isSelected());
```

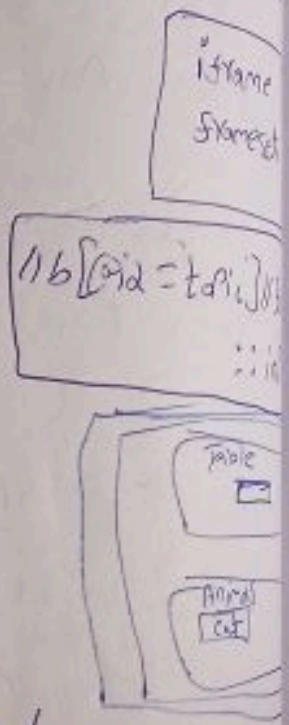
output
true
true
Before
After

Frames

19-10-

Dom Syntax :

```
< iframe id = "876"  name="h876g" />
```




```
driver.switchTo.frame(String id)
driver.switchTo.frame(String name)
driver.switch To.frame(WebElement element)
driver.switd To.frame(int index)
```

http://demo.guru99.com/test/guru99home/

```
< frameset >
```

Program 1:-

```
public class Frames {

    // 1 way by index
    // driver.switchTO().frame(0);

    // 2 way id (or) classname
    // driver.switchTo().frame("frame1");

    // 3 way
    WebElement f = driver.findElement(By.xpath("//iframe[@id='frame1']"));
    driver.switchTo().frame(f);

    WebElement txt1 = driver.findElement(By.xpath("//b[@id='topic']
    txt1.sendKeys("Dinesh");                    //following::input"));
    driver.switchTo().defaultContent();

    driver.switchTo().frame("frame2");
    WebElement drp = driver.findElement(By.id("animals"));
    Select s = new Select(drp);
    s.selectByVisibleText("Avatar");
    driver.switchTo().defaultContent();

    driver.switchTo().frame("frame1");
    driver.switchTo().frame("frame3");

    WebElement check = driver.findElement(By.xpath("//input[@id='a']"));
    check.click();

    driver.switchTo().parentFrame();
    WebElement txt2 = driver.findElement(By.xpath("//b[@id='topic']
                                        //following::input"));
    txt2.sendKeys("Subramanian");
}
```

output

true
true
Before select fails
After send true

19-10-2021

iframe
frameset

Windows Handling

20-10-2021

driver.SwitchTo().window(String url)
driver.SwitchTo().window(String title)
driver.SwitchTo().window(String windowId)

using windowsId we will switch the windows:

To get Parent window Id:

getWindowHandle() - m

To get all windows id:

getWindowHandles() - m

Program 1:-

Public class windowHandle {
=

WebElement close = driver.findElement(By.xPath("//button[text()
Close.click();

WebElement ProductSrc = driver.findElement(By.xPath("//input[@name
ProductSrc.sendKeys("redmi note 7");

WebElement btnSrc = driver.findElement(By.xPath("//div[text()='Redmi
Product.click();
(Ruby Red, 32 GB)]

1. // WINDOW HANDLE BY USING WINDOW ID
String Parentid = driver.getWindowHandle(); //10
System.out.println(Parentid);
Set<String> allid = driver.getWindowHandles(); //10, 28
System.out.println(allid);
for (String eachid : allid) { //10, 20
if (!Parentid.equals(allid)){
driver.switchTo().
}

2. //WINDOW HANDLE BY USING COUNT

String Parentid2 = driver.getWindowHandle();
System.out.println(Parentid2);

Set<String> allids2 = driver.getWindowHandles();
System.out.println(allids2);

// int count 1
// for (String eachid : allids2) {
// if (count == 3) {
// driver.switch().window(eachid);
// }
// count ++;
// }

3. // WINDOW HANDLE USING LIST (SET TO LIST ADD)

List<String> li = new ArrayList<String>();
li.addAll(allids2);

driver.switchTo().window(li.get(2));

WebElement Viewmore = driver.findElement(By.xPath("//span[text()='viewmore']"));

Viewmore.click();

Webtable

21-10-2019

Web Table is made of rows and columns with call.
When we create a table for a web Page that is called a web table.
In HTML, table is created using <table> tag.

Types of WebTable

1. Static web table
2. Dynamic web table

```
<table>
<tr>
<td> id</td>
<td> name</td>
<td> email</td>
</tr>
</table>
```

tr - table row
td - table data
th - table header

Program 1 :-

```
Public class Table{
    ≡
// Table
List <WebElement> tables = driver.findElements(By.tagName(
System.out.Println(tables.size());
// WebElement table1 = tables.get(0);
System.out.Println("---- ALL DATA FROM TABLE ----");
WebElement table1=driver.findElement(By.xPath("//table[@id='cus
String text=table1.getText();
System.out.Println(Text);
```

22-10-2021

oftware build to
ies.

3. Download the latest jar file

In case of any update of jar no need to download again just change the version number alone in dependency it automatically

maven goals :-

clean - clear your cache file (it will clear the last generated report (failed cases) and previous version of jars,

install (automatically) install the software jar when we are adding the dependencies of it.

test - once installation done, we can execute the test scripts.

steps :

1. Download the apache maven zip file :
2. Config the JAVA_HOME and MAVEN_HOME in the environment variable :

User variables :

JAVA_HOME
MAVEN_HOME

System variables :-

JAVA_HOME
MAVEN_HOME
Path

Open Google Search maven download
Click Binary zip archive
Go to pc right click and Advanced Setting
Environment Variable
JAVA_HOME
maven-HOME

3. Verify Maven and JAVA configurred currectly

Cmd --->

java:

java - version

Maven :-

mvn - version

4. Add the maven Plugin in eclipse

5. Create Maven Project in eclipse.

driver =

POM

26-10-2021

Pom - Page object model is a design Pattern in Selenium that creates an object repository for storing all web elements.

* It useful in reducing code duplication and improves test case maintance.

Setter - Keyword
Getter - Variable

types :-

1. with Page factory
2. without Page factory

1. with Page factory :-

@ FindBy
@ FindBys
@ FindAll
@ CacheLookup

why Pom ?

To overcome lots of reworks
Performance issue -- each and every time check the down

for each screen u have to create each class [login, register, ...
each class will contain the locator revelvant to it

in Pom driver.findelement is not used instead of which annotations are used

@ Notation is one kind of class
@ findby acts as webElement, list webElement

WebElement

List <WebElement>

Private
whenever
To main
Page fac
initelem
this (
real tim
No need
In real
Program 1
login. JAV
Public

```
orlLaunch ("https://ww.face");
implwait (10);
Login l=new Login();

fill TextBox(l.getTxtusername(), "maari");
fill TextBox(l.getTexPassword(), "98754");

driver.navigate().refresh();
fill TextBox(l.getTxtusername(), "GreenTech");
fill TextBox(l.getTxtPassword(), "98574");
```

27-10-20

FindBy :-

It will find the only one locator for one webelement

FindBYs :- (Yethavath onnu errornalum output error than)

It used to find one or more locator for same webelement but working based on AND operator, mentioned locators all are valid only going to find the webelement

FindAll :- (Yethavathu onu correct ta irtha pothum.)

It used to find one or more locator for one or same webelement but working based on OR operator, mentioned locators any one is valid going to find the webelement

@ **cachelookup**

It is used for temprory memory management, if any webelement we want to use again and again that time we can use this cachelookup It store the cache memory of b...

Data Driven Framework

28-10-20

EXCel :

Extentions format :

- .XLS:

Row - 2^16

Column - 2^8

- .XLSX:

Row - 2^20

column - 2^14

MACROS supports

JAR:

JXL - .XLS

POI - .XLS + .XLSX

Workbook

- .XLS ---> HSSFWorkbook - C
- .XLSX ---> XSSFWorkbook - C

maven Dep:

```
<dependency>
    <groupId> org . apache . poi </groupId>
    <artifactId> poi - ooxml </artifactId>
    <version> 3.8 - beta4 </version>
</dependency>
```

28-10-2021

program:

```
public static void main (String[] args) throws IOException{
// file location
File f = new File ("E:\\ ..... ");
// file convert to java
FileInputStream ia = new FileInputstream(f);
// FORMAT WHAT WE USE
Workbook wk = new XSSFWorkbook (ia);
// ENTER TO SHEET
    Sheet sheet = wk.getSheet ("datas");
// WHAT ROW WE WANT
    Row row = sheet.getRow(0);
// Row TO CELL
    Cell cell = row.getCell(0);
    System.out.println (cell);
// GET Row SIZE
    int dd = Sheet.getPhysicalNumberOfRows();
    System.out.println (dd);
// GET CELL SIZE
    int tt = row.getphysicalNumberofcells();
    System.out.println (tt);
    System.out.println ("--- For Values ---");
for (int i=0 ; i<sheet.getPhysicalNumberofRows(); i++){
        Row row2 = sheet.getRow(i);
        System.out.println ("- ..... ");
for (int j=0 ; j<row2.getPhysicalNumberOfCells(); j++)
        Cell Cell2 = row2.getCell(j);
        System.out.println (Cell2);
}
```

Program 1 :-

```
// Type 0 ---> Number, Date
// Type 1 ---> String

if (cellType == 1) {
    String value = cell.getStringCellValue();
    System.out.println(value);
} else {
    if (DateUtil.isCellDateFormatted(cell)) {
        Date date = cell.getDateCellValue();
        SimpleDateFormat s = new SimpleDateFormat("dd-MM-yy");
        String value = s.format(date);
        System.out.println(value);
    } else {
        double db = cell.getNumericCellValue();
        long ln = (long) db;
        String value = String.valueOf(ln);
```

Program 2 — Excel CREATE

```
File f = new File("E:\\...\\Mk.xlsx");
Workbook w = new XSSFWorkbook();
Sheet s = w.createSheet("Flipkart");
Row r = s.createRow(2);
Cell c = s.createCell(3);

c.setCellValue("Manoj");

FileOutputStream ot = new FileOutputStream(f);
w.write(ot);
System.out.println("Done")
```

JUNIT - JAVA UNIT Test

1-11-

---> It is a kind of framework
---> To verify the business logic
---> Easy to identify the Passed test cases, Failed test cases

Jars / Dependency used:

Junit 4.12
Hamcrest 1.3 (Supporting Jar)

Annotations:-

@ BeforeClass ---- LaunchBrowser
@ Before ---- Start Time
@ Test ---- Business logic
@ After ---- End Time
@ After class ---- Quit Browser

Note:

---> @BCforclass and @Afterclass methods should be static
---> Methods should not be Private
---> order of execution will be according / alphabetical order
@ ignore ---- To ignore the particular test cases

Assert: validation or verification Purpose

Methods in Assert:

Assert ---> Class
assertTrue() ---> static method
assertEquals ---> Static method

Program:-

Facebook.java

```
Public Class facebook {

@BeforeClass
  Public Static void beforclass (){
    WebDriverManager.ChromeDriver().setup();
    WebDriver driver = ...
```

Assert - - Class - Hard Assert

2-11-

assertTrue()-m ---> boolean condition
assertEquals()-m ---> Expected vs actual

If any one assert get failed it will terminate the test that file itself.
But continue the Remaining testcases.
And testcase Result as failed.

Program :-

```
@Test // invalid Username and invalid Password
Public Void test1() throws InterruptedException {
//verify url
  Launchurl("http fb");
  Assert.assertTrue("verify url", getCurrentUrl().contains("fac
  loginPage = new LoginPage();
// username
  fillTextBox(l.getTxtUserName(), "mano");
  Asert.assertEquals("verify username", "mano", getAttribut(l.
```

Program :-

```
@Test
Public Void Result(){
Result rs= Junitcore.runclasses(A.class, B.class, C.class);
System.out.println("Runcount..... "+rs.getRunCount());
                  ("Ignore Count..."+rs.getIgnoreCount());
                  ("Failure Count...", FailureCount());
                  ("Run Time...      RunTime());
//Suite Result
System.out.println("Suite Result...
```

Testng : Next Generation

9-11-2021

Advantages (or) About Testng (or) why not Junit

* It Provides the default HTML reports
* Easy to set Priority (we can Priortized the test cases)
* Easy to Pass Parameters (we can Pass input as datas from the file Using Parameter
* Data Provider is Possible (we can Pass the Bulk of datas at a time for the Positive and negative testCases Combinations (field validation)
* Cross browser testing (To check the compatability and Stability of application) and Parallel execution is Possible (To save the time consume)
* Automatically rerun the failed testcases.
* Soft Assert is Possible (verify)
* We can Group the bulk of testcases and then we can Skip or Execute the testCases by using the Group name.

Steps:

1. Download the testing jar with latest version 6.14.3 or 6 with supporting jar jcommander version 1
2. Add the testing Jar into eclipse buildPath
3. Add the testing Plugin the eclipse.

Note: No class found Definition error will come here we need to download one more jar called JCommander version 1.27.

Annotations:-

@ BeforeSuite
@ BeforeTest
@ BeforeClass --- launch browser
@ BeforeGroup
@ Beforemethod ---- Start Time
@ Test ---- Bussiness logic
@ Aftermethod --- End Time

Program:-

Public
@ Before
Private
S
3
@ After
Private
3
@ Before
Private
@ After
Private
3
@ Test
Priva
@ Test
Priva
3
@ Tes
3 3

10-11-

ASsertion : Validation (or) verisiation Purpose

Assert : HardAssert

Assert.assertTrue(driver.getCurrentUrl().equals("url"); --- Expected Vs Actual

Assert.assertEquals("url", driver.getCurrentUrl()); --- Actual --- Expected

Verify : Soft Assert

SoftAssert s = new SoftAssert(); - code failnalum code run pannum

s.assertTrue()
s.assertEquals()
s.assertAll() - entha code fail show panni kata assert all use pannum

Suite Level Execution :-

If you want to execute more than one class we can do for suite level execution with help of testing.xml file

Create testing.xml file:

Select all classes which you want to run.
Right click --click on testNG option
Then click convert to testNG
click Finish button
testing.xml file created
To run the testing.xml file -- Rightclick -- Run as -- TestNG Suite

Program 1:

```
@Test
Private Void test2()
//verify url
  LaunchUrl("fb");
  SoftAssert s=new SoftAssert();
  s.assertTrue(getCurrentUrl().contains("facebook"), "verify
  LogInPage l=new LogInPage();
// username
  fillTextBox(l.getTextUserName(), "mano");
  s.assertEquals(getAttri
```

18-11-2021

Rerun failed testcases:

1. manually
2. Automatically

1. manually

Project - Testoutput_failed.xml --- run the file it will run only failed testcases

2. Automatically

With known Failed Testcases

IRetryAnalyzer : Interface.

retry()-m

Without known failed Testcases

Listeners ---- which is going to the listen the failed testcase from the suite file and send those test cases into IAnnotation Transformer interface.

IAnnotation Transformer: I

transform() - - - - -

ITestAnnotation: I

getRetryAnalyzer ---- which is used to get the failed testcases from listeners

setRetryAnalyzer ----- which is used to redirect or send the failed testcases into IRetryAnalyzer interface

Prog:

With known Java

```
Public class withknown implements IRetryAnalyzer {
    int minCount=0, maxCount=5;
  @Override
    Public boolean retry (ITestResult result){
        if (minCount<maxCount){
            minCount ++;
        } return true;
    } return false
}
```

A.JAVA

```
Public class AG
  @Test
    Private void testA1 (){
        Sysout ("Test A1");
  @Test (retryAnalyzer=withknown.class)
    Private void test A2(){
    Scanner s=new Scanner (System.in);
    Assert.assertTrue (S.nextBoolean());
    Sysout ("Test A2");
```

.xml

```
<suite suite-stge = "DEVELOPMENT" name =
        failed suite [default suite]">
<test thread-cat="5" name="default test failed">
  <classes>
    <class name = "ors. rerun.A">
    <methods>
      <include name = "testA2" />
    </methods>
    </class>
```

Without known java

```
Public class without known implements IAnnotation
                                    Transformer{
@Override
Public void transform (ITestAnnotation a, Class
                                    testclass)
    a.SetRetryAnalyzer (withknown.class);
}}
```

rerun testng.xml

```
<suite name ="suite">
<listeners>
<listener class name ="ors.rerun.
                withoutknown"></listener>
```