**Learning Java**

Java has its own runtime environment (JRE) and API, it is called platform.

Class

Collection of object is called Class. Class is used to indicate group of object, when we say human we won’t make out anything until we say some individual name, Human being represents group of human objects (Ram, Shyam,Mohan)

Object

Any entity that has state and behavior is known as an object.

Example: Pen is an object. Its name is Reynolds, color is white etc. known as its state/property. It is used to write, so writing is its behavior.

Public

public keyword is an access modifier which represents visibility, it means it is visible to all.

Static

static is a keyword. The core advantage of static method is that there is no need to create object to invoke the static method. The main method is executed by the JVM, so it doesn't require to create object to invoke the main method. So it saves memory.

Variable is a name of memory location. There are three types of variables in java: local, instance and static.

A variable which is declared inside the method is called local variable.

A variable which is declared inside the class but outside the method, is called instance variable

A variable that is declared as static is called static variable. It cannot be local.

Example:

1. class A{
2. int data=50; //instance variable
3. static int m=100; //static variable
4. void method(){
5. int n=90; //local variable
6. }

}//end of class

Constructor

Constructor name must be same as the class name, Constructor must not have return type.

A constructor with argument is called as Parameterized constructors

Java method is static because object is not required to call static method if it were non-static method, jvm create object first then call main() method that will lead the problem of extra memory allocation.

Method Overloading

If a class has multiple methods having same name but different in parameters, it is known as Method Overloading

Method Overriding

Method Overriding allow us to create same method in parent and child class with same name and with same arguments. Method Overriding is possible only by inheritance

Static methodcannot be overriddenbecause static method is bound with class

This

This in Java helps in referring to the current object. through dot(.) operator we can access the member variables of the current object. this keyword can also be used to call one constructor from another constructors of the same class.

LinkedList

Linked List: allows to store and manipulate group of data, store duplicate object

Inheritance

we can Inherit only non static member of class. In inheritance private member/variable cannot be called inheritance of public and protected member/variable can be done from different package

Abstract

1 method should be there which is abstract in nature, only method declaration no method body

this method will be extended in other class with body—overriding, cannot create obj of abstract class

public abstract class bank{

public abstract void loan() //no {} braces

public void credit(){ //non abstract method

}}

Interface

interface has abstract method only, interface member are by default public, interface variable are by default public static final

interface member are non implemented

Implementation means method is having body {}

Unimplemented means method is not having body, no code {} e.g.: void test1();

Interface support multiple inheritance

e.g.: public class TestInterface implements InterfaceEg2, InterfaceEg2{}

we cannot create obj of interface

InterfaceEg1 obj = new InterfaceEg1();—-because method are unimplemented

we can create ref of interface

InterfaceEg1 obj = new TestInterface();——reference

HashSet

HashSet store unique values

HashMap

HashMap store value in form of Key Value pair

List/Set

List interface: if only 1 either roll# or name

set Interface: want to store unique name

list interface: unique or duplicate names

Example : HashMap

Map<Integer, String> m = new HashMap<Integer, String>();

m.put(1, "Chirag");

m.put(1, "Chirag1");

set<String> keys= m.keySet();

for(Integer i:keys)

{

syso("Roll # : "+i+"Student name : "+m.get(i));

}

o/p: Roll # :1 Student name : Chirag

Roll # :2 Student name : Chirag1

Example TreeMap:

map

Map<String, String> m = new TreeMap<String, String>();

m.put("fb", "Chirag");

m.put("google", "Chirag1");

set<String> keys= m.keySet();

for(String i:keys)

{

syso("App name : "+i+"Student name : "+m.get(i));

}

o/p: App name # :1 Student name : Chirag

App anme # :2 Student name : Chirag1

Collection

Collection is used to Search, insert, manipulate, delete, sort data

Java.util package

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Interface: Classes:

\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

set, list, queue, Deque array list,vector, linklist, hashset, treeset

Iterable

\_\_\_\_|\_\_\_\_\_

Collection

\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_

list Queue Set

|

Classes: Arraylist----------> class implement Interface HashSet

LinkList LinkHashset

Vector

Stack

Static array list: int a[]=new int[3];

Dynamic array list: ArrayList ar = new Arraylist();

ar.add(10);

ar.add(20);

ar.add(30);

ar.add(30);

syso(ar.size());

can contain duplicate element/values

fetch array value: syso(ar.get(4));

fetch all values: for(int i=0;i<=arr.size;i++)

{syso(ar.get(i));

}

Generic://new array object

ArrayList<Integer> ar1 = new Arraylist<Integer>();

ar1.add(100);

ar1.add("selenium")//throws error since array is integer type

when not sure which kind of value to be stored

ArrayList<E> ar1 = new Arraylist<E>(); //also to be declared in method

This Keyword

this is used to initilize global and local variable when both are same

Linkedlist:

Linkedlist<String> ll = new LinkedList<String>();

ll.add("test");

ll.add("test1");

ll.add("test2");

syso(ll.get(0));

(ll.set(0, "Tom"));//set value

syso(ll.get(0));

//different ways of printing linkedlist

for loop:

fro(int n=0;n<ll.size;n++)

{syso(ll.get(n));

}

Advance for loop

for(String str : ll)

{ syso(str);

}

Iterator:

iterator<String> it = ll.iterator();

while(it.hasNext())

{syso(it.Next);

}

While loop:

int num=0;

while(l.size()>num)

{syso(ll.get(num));

num++;

}

HashTable:

HashTable h = new HashTable();

h.put("A", "Test");

h.put("B", "Hello");

syso(h.size());

h.put(1, 100);

sysy(h.get(2));

HashTable<Integer,Integer> h1 = new HashTable<Integer,Integer>();

h1.put(1,100);

h1.put(1,"ABC");//not allowed

Array: to store similar datatype values

int i[] = new int[4];

i[0] = 10;

i[1] = 20;

i[2] = 30;

i[4] = 40;

syso(i[2]);

Array index out of bound exception

syso(i.length);

//print all values of array

for(int j=0;j<i.length;j++){

syso(i[j]);

}

Array disadvantage:1. size is fixed--static array--to overcome this problem we use collection- arraylis, hashtable

2. store only single datatype--to overcome this problem we use object array

Object arry: Object ob[] = new Object[6];

ob[0] = "Tom";

ob[1] = 26;

ob[2] = 12.33;

ob[3] = 1/1/1989;

ob[4] = 'm';

ob[5] = "london";

syso(ob[5]);

Class--Object: public class Car(){

int mod;

int wheel;

public static void main(String[] args){

//new car();--is Object of car class

a,b,c are referencing these object

Car a = new Car();

Car b = new Car();

Car c = new Car();

a.mod="BMW";//intiilize variable

b.wheel=4; //initilize variable

syso("before assigning obj ref");

syso(a.mod); syso(a.wheel);

}

}

shifting of obj reference: syso("after assigning obj ref");

a=b; b=c; c=a;

a.mod=10; syso(a.mod); //o/p=10

c.mod=20; syso(a.mod); //o/p=20

Functions: //function and method are same things

public class functioninjava{

public static void main(String[] args){

}//main method, nothing is return inside so no o/p, main method is void because return a value

public static void main(String[] args){//static method

functioninjava obj =new functioninjava();

obj.test(); int l = obj.pqr();syso(l); String s1 = obj.qa(); syso(s1); int d1 = obj.dviision(30, 10); syso(d1);

}

//void means return nothing

//return type void

public void test(){syso("test method");}//non-static method

// int is return type, no input some output

// return type int

public int pqr(){syso("pqr method");//non-static method

int a=10; int b=20; int c=a+b; return c;}

// no input some output

//String is return type

- public String qa(){syso("qa method");//non-static method

String s="Selenium"; return s;}

// x,y are input parameter

//int is return type

public int dviision(int x, int y){syso("dviision method");//non-static method

int d = x/y; return d;}

}

Method overloading: //same method name but with different parameter

public class methodoverloading{

limitation: can not create method inside a method, we can overload main method but with different parameter size

public static void main(String[] args){

methodoverloading obj = new methodoverloading();

obj.sum();

obj.sum(7);

obj.sum(110,5);

}

public void sum(){// zero parameter

syso("sum method");

}

public void sum(int i){//one parameter

syso("sum method"); syso(i);

}

public void sum(int k, int b){//two parameter

syso("sum method"); syso(k+b);

}

static & non static: //same method name but with different parameter

public class staticandnonstatic{

String name = "Tom";//non static variable

static int age = 25;//static variable

public static void main(String[] args){

//call to static method and var

sum();

staticandnonstatic.sum();

syso(age);

syso(staticandnonstatic.age);

// call to non static method

staticandnonstatic obj= new staticandnonstatic();

obj.sendmail();

syso(obj.name);

//can i call static method by using obj reference? Yes

obj.sum();

}

public void sendmail(){// non static method

syso("send mail method");

}

public static void sum(int i){//static method

syso("sum method");

}

public void sum(int k, int b){//two parameter

syso("sum method"); syso(k+b);

}

**Project Object Model**

POM is a approach- design Pattern

data driven is a frame work

TestNG to write test case

for each and every page we have to create a seperate java class

TestBase.java(webdriver, prop.file)

(3.base level=max window, timout,delete cookie, get.url) (4.config/env variable=url, Unm,Pass, browser)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. page layer 2. Test layer--TestNG-annotation |

| |

--then we will define |\_\_\_\_Page lib/OR--collection of loginpagetest.java

our objects/web elements | all webobjects homepagetest.java

--action/methods(forgotpass, login btn)

--this includes(login pg, hm pg, reg pg, search, add to cart)

1. page layer

2. test layer

3. base level

4. config

5. test data

6. utils(screenshot, send mail)

7. reporting

src/main/java= utilities, page classes. lib

com.crm.qa.pages\_pkg

**LoginPage.java**

public class LoginPage extends TestBase{

we have to define 1.OR/pagefactory

@findBy(name="username")

WebElement username;

@findBy(name="password")

WebElement password;

@findBy(xpath="//input[@type='submit']")

WebElement loginbtn;

//how to initilize the elements with the help of page factory, we will create the constructor of class once again

public LoginPage(){

pageFactory.initElements(driver.this); //this is how we initilize pagefactory, this means current class obj

}

//Actions

public String validateloginpageTitle(){

return driver.getTitle();

}

public HomePage login(String uname, String pass){

username.sendKeys(uname);

password.sendKeys(pass);

loginbtn.click();

return new HomePage();

}

}

**SignUpPage.java**

public class SignUpPage extends TestBase{

}

**HomePage.java**

public class HomePage extends TestBase{

@FindBy(xpath="//td[contains(text(),'User: Naveen k')]")

WebElement userNameLabel;

@FindBy(xpath="//a[contains(text(),'Conatacts')]")

WebElement contactsLink;

@FindBy(xpath="//a[contains(text(),'Deals')]")

WebElement DealsLink;

//create constructor, Initilizing the page object

public HomePage(){

PageFactory.initElement(driver, this);

}

public String verifyHomePageTitle(){

return driver.getTitle();

}

pubic ContactPage clickOnContactsLink(){

contactsLink().click();

return new ContactPage();

}

pubic DealsPage clickOnDealsLink(){

dealsLink().click();

return new DealsPage();

}

}

**ContactPage.java**

public class ContactPage extends TestBase{

}

**DealsPage.java**

public class DealsPage extends TestBase{

}

com.crm.qa.base\_pkg

**TestBase.java**

public class TestBase{

public static WebDriver driver;

public static properties prop;

public TestBase(){ //TestBase constructor

try{

prop=new Properties();

FileInputStream ip = new FIleInputStream(System.getProperty("user.dir")+"src/main/java/com/crm/qa/config/config.properties");

prop.lado(ip);

}catch(FilenotFoundExecption e){

e.printStackTrace();

}catch(IOException e){

e.printStackTrace();

}

}

}

public static void initilization(){

String browser = prop.getProperty("browser");

System.setProperty("WebDriver.chrome.driver", "driver path");

driver= new ChromeDriver();

driver.manage.maxamize.window

driver.manage.deletallcookie

driver.manage.timeout.pageload(20--refer TestUtil.PAGE\_LOAD\_TIMEOUT, TIMEUNIT.SEC);

driver.manage.implicitwait(10--REFER TestUtil.IMPLICITWAIT. TIMEUNIT.SECOND);

driver.get(prop.getProperty("url"));

}

com.crm.qa.config\_pkg

new file=config.properties

url=

uNm=

Pass=

browser=

com.crm.qa.testdata\_pkg

com.crm.qa.utilities\_pkg

com.crm.qa.util\_pkg

public class TestUtil{

public static long PAGE\_LOAD\_TIMEOUT=20;

public static long IMPLICIT\_WAIT = 10;

}

src/test/java= only test cases

com.crm.qa.testcases\_pkg

**LoginPageTest.java**

public class LoginPageTest extends TestBase{

LoginPage loginPage;

HomePage homePage;

//create constructor of Loginpagetest

public LoginPageTest(){

super(); //super is used to initilize properties from base class before initilization

}

@BeforeMethod

public void setUp(){

initilization(); //driver will be initilized

//create obj of login page class

loginpage = new LoginPage();

}

@Test(priority=1)

public void loginPageTitleTest(){

String title = loginPage.validateloginpageTitle();

Assert.assertEquals(title, "ExpectedTitle");

}

@Test(priority=2)

public void loginTest(){

homePage = loginPage.login(prop.getProperty("username"), prop.getProperty("password"));

}

@AfterMethod

public void tearDown(){

driver.quit();

}

}

**HomePageTest.java**

public class HomePageTest extends TestBase{

LoginPage loginPage;

HomePage homePage;

ContactPage contactPage;

public HomePageTest(){ //constructor of homepagetest

super(); //so the baseclass will be called and obj prop will in initialised

@BeforeMethod

public void setUp(){

initilization(); //driver will be initilized

loginPage = new LoginPage(); //this is homepg and we need to login, so that i can access login page obj

homePage = loginPage.login(prop.getProperty("username"), prop.getProperty("password"));

}

@Test(PRIORITY=1)

public void verifyHomePageTitle(){

String homePageTitle = homePage.verifyHomePageTitle();

Assert.assertEquals(homePageTitle, "Expected", "home page title not match");

}

@Test(PRIORITY=2)

public void clickOnContact(){

contactPage = homePage.clickOnContactLink();

}

@AfterMethod

public void teatDown(){

driver.quit(); }

}

}

Source folder--src/main/resources

**TestNG.xml**

Extend report:

public class ExtentReportListner implements IReporter{

private ExtentReports extent;

public void generateReport(List<XmlSuite> xmlSuite, List<ISuite> suite, String outputDirectory){

extent = new ExtentReport(outputDirectory + File.separator + "Extent.html", true);

for(ISuite suite : suites){

Map<String, iSuiteResult> result = suite.getResults();

for(ISuiteResult r:result.values()){

iTestContext context=r.gerTestContext();

buildTestNodes(context.getPassedTests(),LogStatus.PASS);

buildTestNodes(context.getPassedTests(),LogStatus.FAIL);

buildTestNodes(context.getPassedTests(),LogStatus.SKIP);

}

}

extent.flush();

extent.close();

}

private void buildTestNodes(IResultMap tests, logStatus status){

ExtentTest test;

if(test.size()>0){

for(ITestResult result : tests.getAllResults()){

test = extent.startTest(result.getMethod().getMethodName());

test.setStartedTime(getTime(result.getStartMills()));

test.setEndedTime(getTime(result.getEndMills()));

for(String group: result.getMethod().getGroups())

test.assignCategory(group);

if(result.getThrowable()!=null){

test.log(status, result.getThrowable());

}else{

test.log(status,"Test"+status.toString().toLowerCase()+"ed");

}

extent.endTest(test);

}

}}

private Date getTime(ling millis){

Calendar calendar = Calendar.getInstance();

calendar.setTimeInMillis(millis);

return calendar.getTime();

}}

add to testNG.xml--

<listeners>

<listener>Class-name="com.salesup.listener.ExtentReporterListner.ExtendReportNG"/>

</listeners>

**Selenium Java**

Dynamic xpath==(//"input[@class='classname']")

(//"input[@name='namename']")

(//"input[contains(@class,'classname')]")

Dynamically changing xpath:

id=test\_123

test\_456

test\_789

(//"input[contains(@id,'test\_')]")

(//"input[starts-with(@id,'test\_')]")

clicking on link:

all the links are represnented by <a>

("//a[contains(text(), 'My Account')]")

driver.findElements: TOTOAL # of links

List<WebElement> linklist = driver.findElements(By.tagName("a"));

syso(linklist.size());

driver.findElements: TOTOAL # of input fields

List<WebElement> linklist = driver.findElements(By.tagName("input"));

syso(linklist.size());

print link list:

for(i=0; i<=linklist.size; i++)

{

String linktext = linklist.get(i).getText();

syso(linktext);

Google AutoSuggest: USE LAST DIV CLASS

List<WebElement> list = driver.findElements(By.xpath("//ul[@role='listbox']//li/decendant::div[@class='sbqs\_c']"));

syso(list.size());

Printing all the suggestion:

for(i=0; i<list.size; i++)

{

syso(list.get(i).getText());

if(list.get(i).getText().contains("Java Tutorials"))

{

list.get(i).click();

break;

}

}

}

xpath for login button==//input[@type='submit' and @vlaue='login']

link xpath==//a[text()='Features'] OR //a[contains(text(),'Features')]

button xpath==//button[@type='button'and @class='btn']

javascript executor to click element:

((javascriptExecutor)driver).executeScript("document.getElementsByByName('fdo')[0].click()")

**Appium**

Appium is http server written in Nodejs that creates and handles web driver sessions

Appium support android, iOS, firefoxOS using vendor-provided framework

vendor-provided framework for

iOS- apple’s UIAutomation

Android- UIAutomator

Appium receives request/command from client lib and execute the command in target device/emulator and respond with http response

appium works as client/server architecture

appium support Ruby, Python, Java,

so what ever code we write in java is called client lib for appium = Appium Client

jar files Vendor provide framework request bootstrap.jar

Appium client <—> Appium Server <—> UIAutomator/selendroid <—> Emulator

jar files Vendor provide framework request bootstrap.js

Appium client <—> Appium Server <—> UIAutomation <—> Simulator

appium commands are executed in context of session

client initiate a session with server in way specific to each lib but they end up sending POST/session request to server with JSON obj called desired capabilities

at this point server will start up the session and respond with session id

automationName, platformName, PlatformVersion, deviceName, app

capabilities.setCapability("platformName", "iOS");

capabilities.setCapability("platformVersion", "10.2");

capabilities.setCapability("deviceName", "iPad Air");

capabilities.setCapability("automationName", "XCUITEST");

capabilities.setCapability("app", "/Users/siya-ui/Library/Developer/Xcode/DerivedData/RapidRMS-hcwddznqzqirudfffvmlkxkdnhet/Build/Products/Debug-iphonesimulator/RapidRMS.app");

IOSDriver driver = new IOSDriver(new URL("http://0.0.0.0:4723/wd/hub"), capabilities);

File scrFile11 = driver.getScreenshotAs(OutputType.FILE);

File targetFile11 = new File("/Users/siya-ui/Documents/rajesh gupta/sel results/Deptcase42" + System.currentTimeMillis() +".jpg");

driver.swipe(624, 366, -700, 300, 500);

WebElement tab30=driver.findElement(By.xpath("//XCUIElementTypeOther/XCUIElementTypeOther/XCUIElementTypeOther/XCUIElementTypeOther/XCUIElementTypeTable/XCUIElementTypeCell/XCUIElementTypeStaticText"));

JavascriptExecutor jscase30 = (JavascriptExecutor) driver;

HashMap scrollObjectcase30 = new HashMap();

scrollObjectcase30.put("direction", "left");

scrollObjectcase30.put("element",((RemoteWebElement) tab30).getId());

scrollObjectcase30.put("text", ((RemoteWebElement) tab30).getText());

jscase30.executeScript("mobile: scroll", scrollObjectcase30);

**REST API**

Rest API:

what is api: non UI testing, by providing jar file, locally available, not over network

what is web service: not providing jar file but over http/network call

SOAP Rest

protocol pattern

simple object access protocol Representational state protocol

soap can’t use rest rest can

soap use services rest uses URI

JAX-WS- security JAX-RS security

soap permit xml only rest allow html, xml, json text

HTTP Method:

C create- Post, R- reterive- Get, U update- Put, D delete- Delete

HTTP Request- URI, Header, Payload(JSON/xml)

HTTP Response- Status, Response payload, success msg

URL(end point) + service URL = URI—Uniform Resource Identifier

Path parameter defined by /, query parameter defined by ?

to post= Post/Body/Raw/Json—content type= application/json

swagger—api documentation—brother of Postman

selenium dependencies for rest:

httpclient maven, http-core, json lib

**GIT**

Git is tool and GitHub is webService

Git services: wiki, test management, Bug tracking, hosting service

1. login to Github

2. New Repository(+)icon

3. Eclipse project

4. go to project location in cmd prompt

5. initialize git- git init

6. .git file will be created-check by: ls -alt

7. git remote add origin git@github.com:naveenanimation20/DemoRepo.git: copy from git login page

|

git url of git repository

8. git status

9. git add .

10. git status

11. git commit -m “comments”

12. git push origin master (branch name)

13. SSH key—GIT—settings—SSH and GPG keys

14. docs.joyent.com—ssh key for windows/mac

Pull code

15. go to drive

16. create dir: mkdir dirname

17. git clone url

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

to commit go to location in cmd

1. create new branch= git checkout -b "branchname"

2. list all branch= git branch

3. git status= all commited and uncommited code

4. git add .= to add all the code===for indexing in repository

4.a. git add filepath==to ad specific file to repoitory

5. git commit -m "comments what is change"===to commit to repository

6. git push= to push the code

7. git pull= to pull all code from repository

8. untracked file= file newly added to branch which is not yet pushed to repository

clone code:

1. download & install git

2. go to local code location

3. right click git bash here- it will open cmd

4. type git clone paste git cloned url