Automation

Disadvantage of Manual ⟶

1) Time Consuming While doing ~~re-de~~ regression testing
2) more resources are required
3) Less accuracy → Human Users may make mistakes so we can not expect more accuracy in MT. (In AT if we provide Correct logic then tst ~~Poot~~ tool provide Correct output)
4) Processing Change request during software maintainance takes more time

Automation Testing →

AT is process of testing the software using an automation tool to find the defects. In this process executing the test & Cases & genrating the results are performed automatically by Automation Tool.

Advantages of AT →

1) Less human resources
2) Fesible for regression testing
3) Fast development & delivery → Automated tests can be executed repetedly & Completed rapidaly, we do not have to wait for weeks to execute the tests, only few hours are enough for execution. Switching from manual to automation reduce the waiting time & boost development.
4) Ability to test on Various platforms
5) Cost of Project will be reduced

6) Project duration will be reduced
7) Saves time

Different Automation tools →

1) Loadrunner
2) Ts Complete
3) Selenium
4) IBM rational function Testes
5) Hp QTP

Selenium

Selenium →

It is functional & regression automation tool. It is Webbased automation tool / Collection of API's

Version of Selenium →

1) **S. IDE** → plug in only for firefox browser. Record & play back tool

2) **S. RC** → RC server will interact with browser & execute automating test cases.

3) **S. Grid** → We can execute test Case ~~remov~~ remotly.

4) **S. Webdriver** → Webdriver is an interface which is present in API forms.

Advantages of Selenium →

1) Open Sources (Free)
2) support multiple languages (Java, Python, Ruby)
3) support multiple browsers (Chrome, IE Safari) ~~Linux, Mac Os~~)

4) support different Operating system (Window, Linux, MacOs)

5) Support Parallel testing

6) Selenium Can integrate third party tool (TestNG, Maven, Jenkin)

5) Disadvantage of Selenium →

1) It Can not handle Window based application (Not support desktop/ Window based application) → AutoIT tool, Sikuli

2) There is not dedicated Support faculty → N no. of forms are available

3) Can not automate Captcha.

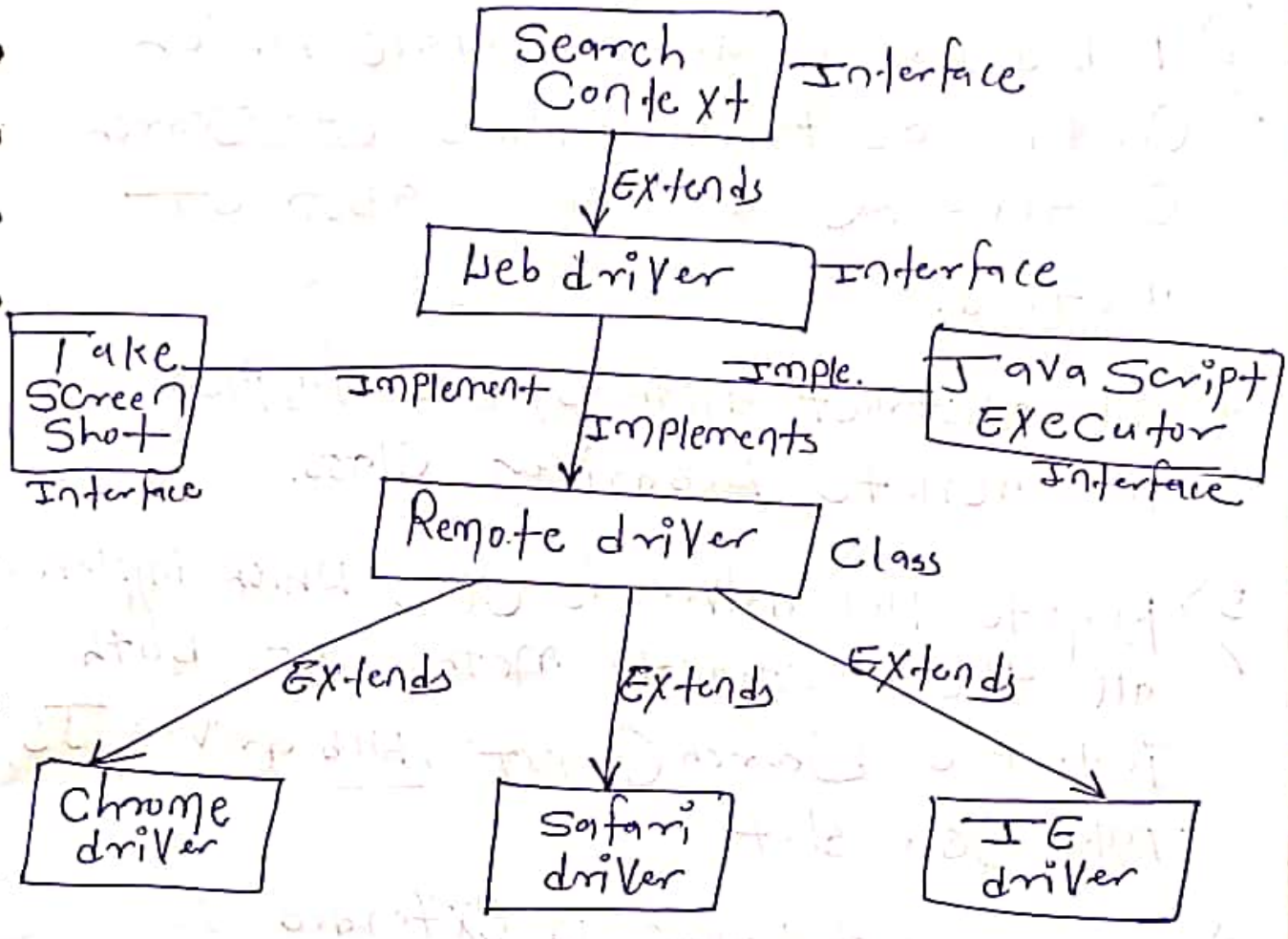
4) Can not genrate good report (i.e TestNG, Extent report Concept)

5) Can not Support Working With excel sheet

6) Image ~~bas~~ based testing is Not supported

Selenium Web driver Architecture →

```
WebDriver driver = new ChromeDriver();
```



Interface → to → Interface → Extends

← to → Class → Implement

1) Search Context is supermost interface which Contains abstract methods.

2) Search Context inherited to Web driver

3) Web driver is an interface which Contain abstract methos of Search Context & its own abstract method.

4) All abstract methods are implemented in remote Webdriver Class.

5) Remote Web driver is Class which implement all the abstract method of both interface Search Context, Web driver, JS, Take Screen shot

6) Remote Webdriver is extended to browsers such as Chrome, IE etc.

7) To run application in ~~multiple~~ single browser i.e Writing test Script by using ~~sig~~ single browser, but run the same Script in multiple browser we need to use runtime polymorphism by using upcasting in selenium.

i.e

Webdriver driver = new Chromedriver();

8) Create an object of Chromedriver class with reference of Webdriver interface.

Methods of Webdriver →

1) Get()→

- This method is use to open an application or to enter url in a web page or browser.
- It's return type is Void.

2) Close()→

- This method is use to close current tab only.
- Return type is Void.

3) Quit()→

- This method is use to close all the tab. ~~or Or whole browser.~~
- Return type is Void.

4) getTitle() →

- This method is use to get title of Web page as an output.
- Return type is String.

5) getCurrentUrl() →

- This method is Use to get Current Url of the Current Web page as an output
- Return type is String.

6) maximize() →

- This method is Use to maximize browser.

Syntax →

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

7) **Navigate** →

- This method is to open an applicatn move forword, backword & refresh the browser.
- Navigate method can be used for alternate method for get method

1) driver.Navigate().back();
2) ——————— forword();
3) ——————— Refresh();

8) **Set Size** →

- This method is use to change size of the browser which accept dimensions arguments.
- Set Size method can't accept directly the width & ht. of browser. This method accept only dimension argument
- Before using setsize() method we need to create the object of dimension class & then pass the width & ht. in the Constructor

of dimension Class.

Dimension D = new Dimension (W, h)

driver.manage().window().setsize(D);

9) getsize →

- Check the browser size

s/s → println(driver.manage().window().getsize());

10) Set position →

- This method is use to change position of browser which accept point argument.
- Before using set position method we need to Create the object of point Class.

Point p = new Point(X, Y)

driver.manage().window().setPosition(p);

11) **get position** →

- Check position of browser in terms of X & Y Co-ordinates.

S/so → println (driver.manage().Window()
.getPosition())

Thread →

- It is java Class use to pause or wait

→ Thread.sleep (3000);

HTML

- Hiper text Mark up language
- It is use to Webpase design

* Locators *

Locators →

- Locators are the way to identify an HTML element on a webpage.
- To identify an element present in webpage we need to use "findElement" method which is present in Web driver.

```
WebDriver driver = new ChromeDriver();
driver.findElement
```

- Find element method will identifies an element with the help of By class which contains static methods.
- All the static methods present in By class are known as locators type.

Locators types ⟶

1) Tagname
2) ID
3) Name
4) Class Name
5) Link Text
6) Partial link text
7) Css Selector
8) X path.

X path ⟶

- It is a syntax or language for finding any element on the Webpage using Xml path expression / X path.
- X path is used to find the location of any element on a Webpage using HTML Dom structure.
- X path is a address of element.

Types of XPath →

1) Absolute XPath
2) XPath by text
3) XPath by Contains
4) XPath by index
5) XPath by attributes
6) Relative XPath

1) XPath by attributes →

Syntax :—

tagname[@ attributname = "attribute Value"]

driver.findElement(By.XPath("//tagname[@ attriname = "attribute Value"]"));

2) XPath by text →

Syntax →

tagname(text()) = "text Value"]

driver.findElement(By.XPath("//tagname[text()="text Value"]"));

3) XPath by Contains ⟶

It is used ⟶

1) When text is long.

2) When there is dynamic ID.

(Dynamic ID ⟶ means the value of ID is changed)

This Contains is Created

by attribute

by text

Syntax ⟶ By attribute

//tagname (Contain @attribute = "attributValue")

driver.findElement(By.XPath("//tagname(contain[attributename="attributeValue"]))));

Syntax ⟶ By text.

//tagname(contains(text(),"textValue"))

driver.findElement(By.xpath("//tagname(contains(text(),"textValue"))"))

4) Xpath by index ⟶

Syntax ⟶

(//tagname[attributename = "attributeValue"])[Indexing]

- Indexing should strat from 1.
  i.e 1 is the default indexing

Disadvantage ⟶ If there is updation in application or if there is addition of new module then the value is index is change.

Name →

When we can use name locator →

1) If id & class Name is duplicate
2) If id attribute & class Name is not present.

When we can't use ~~class~~ Name locator →

1) Name is not present
2) Name is duplicate.

syntax →

```
driver.findElement(By.Name("NameValue"))
```

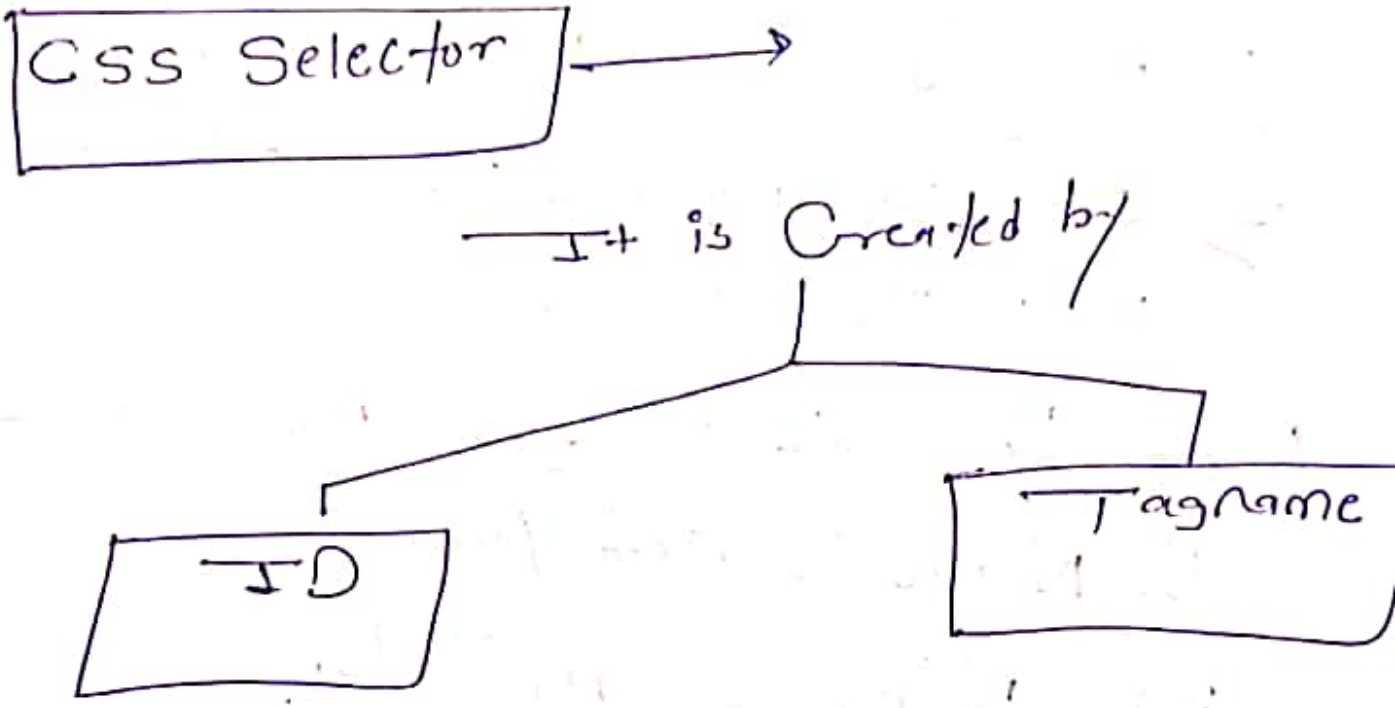
ID →

~~To Create or slect unique XPath for that reason we use ID locator~~

syntax →

```
driver.findElement(By.ID("IDValue"))
```

If dynamic ID then we are using ~~&~~ Xpath by Contains.



Css Selectors in Selenium are String patterns used to identify an element based on a Combination of HTML tag, id, Class & attributes

Absolute XPath	Relative XPath
Absolute XPath starts from root node. (html)	Relative XPath ~~starts from~~ directly jumps to element on Dom.
It Starts With /	//

WebElement Method →

1) ~~Set~~ Send key method
2) Click method
3) Clear method.
4) Get text method.
5) ~~Is~~ Selected method
6) ~~Is~~ Displayed method
7) ~~Is~~ Enabled method

Is Selected →

- This method determines if an element is selected or not.

This method is used on check boxes ☐, radio button ○, & options in select.
- It returns true if the element is selected & false if it is not selected.

Is Displayed →

- This method determines if an element is displayed or not.
- Advantage of this method is that it avoids parsing an elements style attribute.

Is Enabled →

- This method determines if an element is Enabled or not.

→ Is selected, Is Displayed, Is Enabled this all are returning the value in boolean form (i.e. true or false)

* List box →

To Enter date of birth (D/M/Y)

1) By using Select Class →

a) Identify listbox which need to be handled & store it in reference Variable

Ex. WebElement W = driver.FE().

b) Create the object of select Class

Ex. Select S = new select ( )

c) Use select Class method to select option.

Ex. S. select by Value ("string")
S. Select by index (int)
S. select by Visible text ("string")

In the Select Class all the methods are Non-static.

disadvantage of select class →

Date, month & yr. Can't handle at a time, we have handle separately by ~~using~~ creating three object of select class.

- To overcome problem of select class —

b) we use String by creating method

disadvantage → One string store one value.

- To overcome proble of string by — creating method.

a) we use Array.

disadvantage → Array can store only homogenious type of data & size of array is fixed.

* Collection Framework ⟶

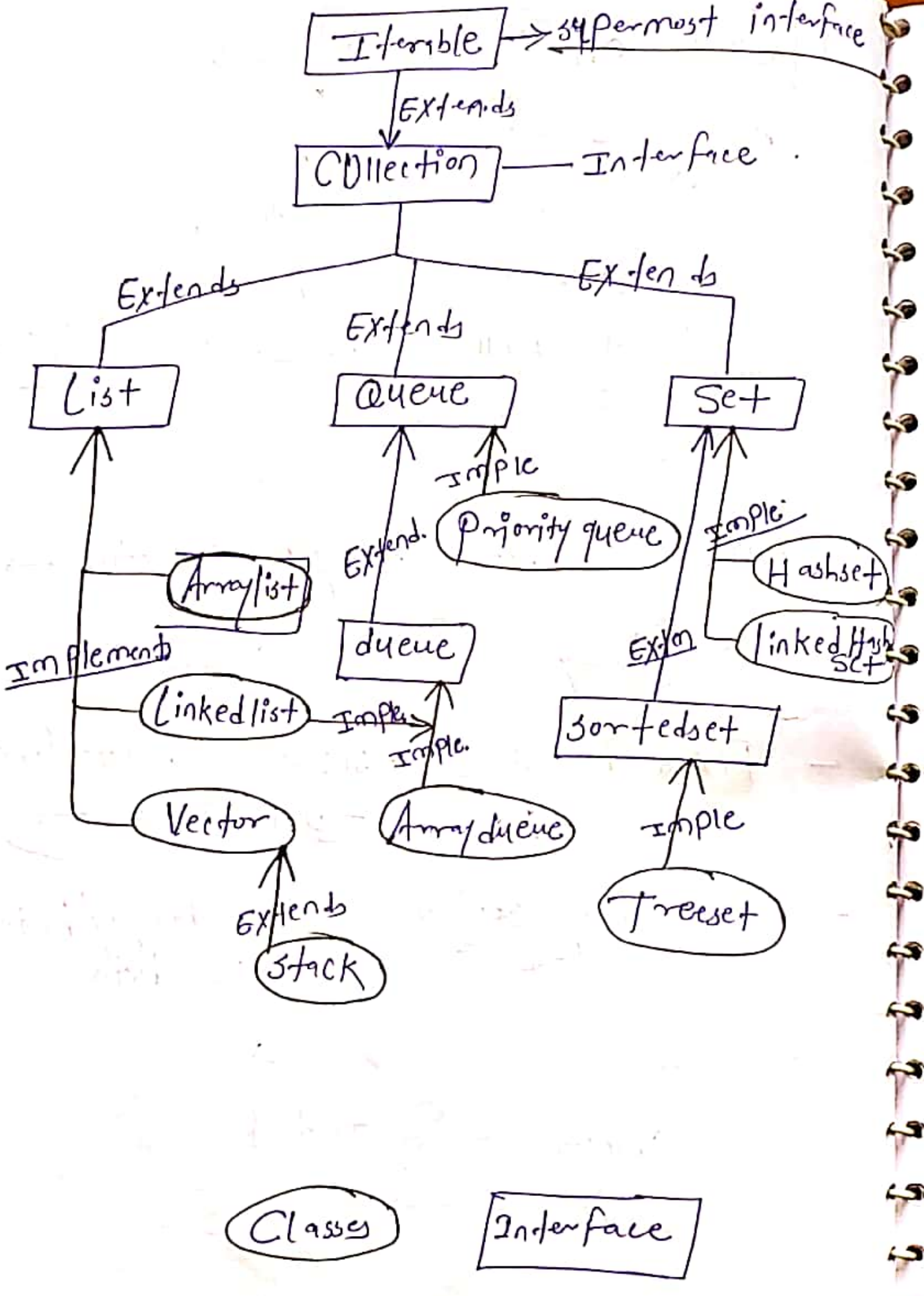
Advantage ⟶

a) Growable in nature.

b) It can store homogenious as well as hetrogenious type of data..

Difference betn Array & Collection ⟶

Array	Collection
- Fixed in size	- size is not fixed (Default size is)
- Store homogenious type data	- Store homo/hetro type data

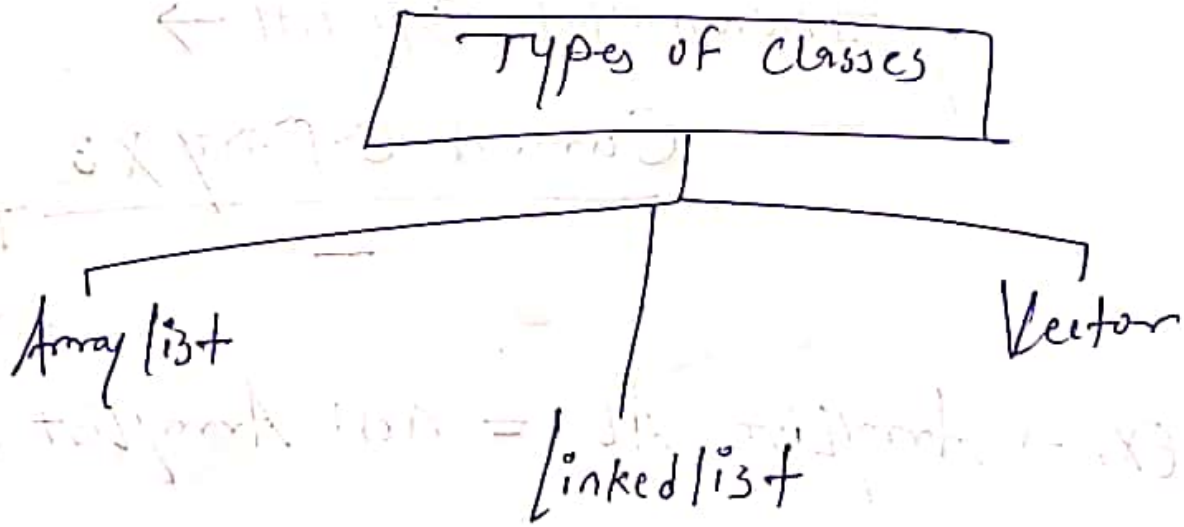


1) List ⟶

- List interface is Child interface of Collection interface.

Advantage ⟶

1) Duplicate are allowed in list
2) Allow any no. of null values
3) order of insertion maintained.



a) Array list ⟶ (AL)

- It is one of the Class which is implemented in list interface

Advantage of AL →

1) store duplicate values/element
2) Maintains insertion order
3) store heterogenous type of data
4) Allows any no null values
5) Default size of arraylist is 10
6) Incremental Capacity →

$$\frac{\text{Current Capacity} \times 3}{2} + 1$$

Ex. → ArrayList AL = new ArrayList();

Methods of arraylist →

i) **size** → to find size of array
println(AL.size())

ii) **add** → Insert new element
AL.add(100);

iii) Remove → to remove from arraylist

AL.remove (indexing)

iv) Print arraylist →

println (AL)

v) Retrive specific element / get specific element →

println (AL.get(2))

vi) Change / replace element →

AL.set (index, new element)

vii) ~~is~~ isempty →

~~AL~~ println (AL.isEmpty());

viii) Reading the element.

a) for (int i=0 ; i<AL.size; i++)

{println (AL.get(i))}

b) for each / advance for loop.

```
for (Object Obj : AL)
{ println (obj)}
```

ix) Add all → To create duplicate of same arraylist

```
AL.addAll (AL)
println (AL)
```

x) Remove all → To remove all value from arraylist

```
AL.removeAll (AL);
println (AL)
println (AL.isEmpty ())
```

Disadvantage Of Arraylist ⟶

- If we add new value at some index then the indexing of all next values as well as element position are changed.

Method from Collections Class ⟶
(used in arraylist)

i) sort ⟶

```
Collections.sort(AL)
```

→ to print arraylist in reverse order

Syntax → `Collections.sort(AL, Collections.reverseOrder())`

```
println(AL)
```

ii) shuffle ⟶

```
Collection.shuffle(AL)
println(AL)
```

Converting Array into ArrayList →

~~string arr[]= new str~~

string arr[] = { "1", "2", "3" };

~~Syntax →~~

~~ArrayList AL = new ArrayList(~~

Syntax →

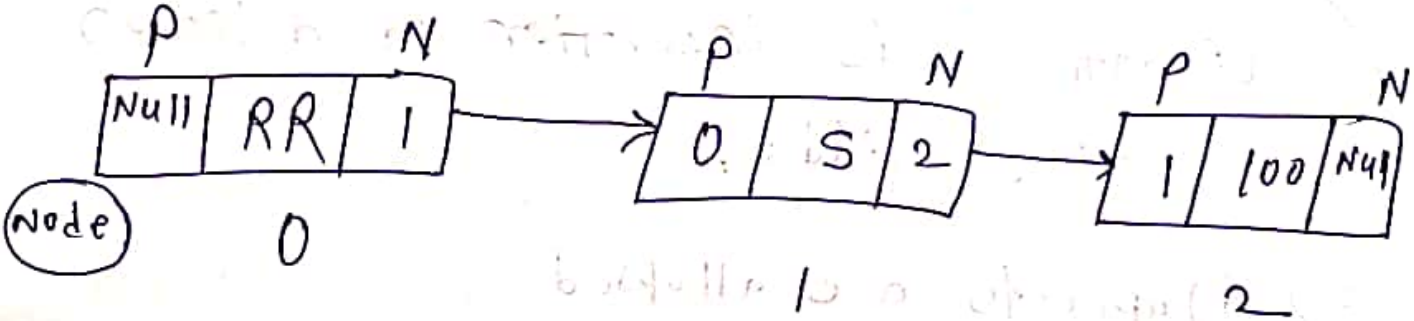
```
ArrayList AL = new ArrayList(Arrays.asList(arr));
Println(AL)
```

b) Linked List ⟶ LL

Syntax ⟶

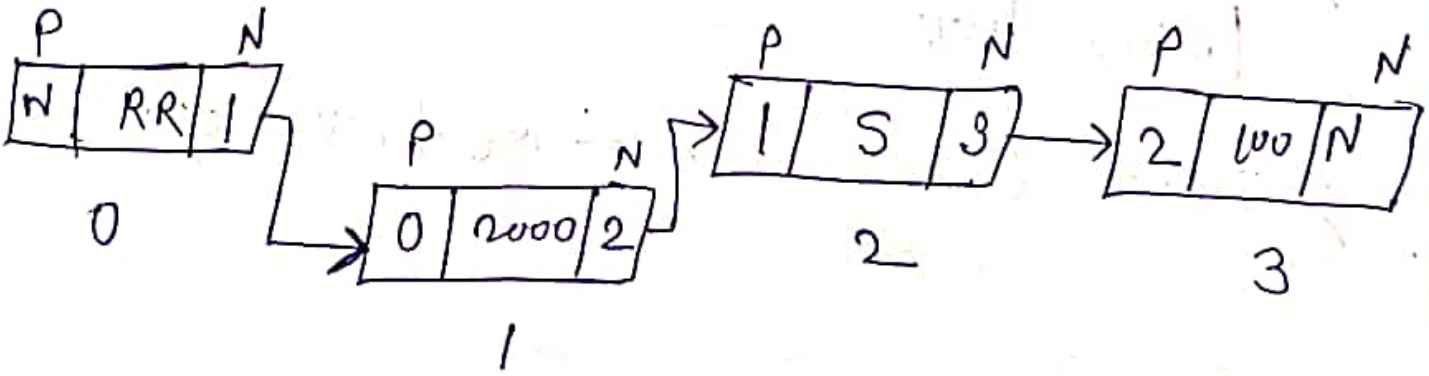
```
LinkedList LL = new LinkedList();
```

```
LL.add("RR")
LL.add(5)
LL.add(100)
```



~~LL.add("2000")~~

```
LL.add(1, 2000)
```



- If we add new value at some indexing then the ~~post~~ ~~next~~ indexing of next nodes are changed position of elements (value) are not changed. → Advantage

1) LL is best choice if our frequent operation is insertion or deletion in the middle.
2) Duplicates are allowed
3) Null insertion is possible
4) insertion order is preserved
5) Underlying data structure is Doubly linked list
6) Worst choise for – Retrieval operation

Methods of LL →

1) size
2) Remove
3) get
4) first index
5) Last index
6) set
7) is empty
8) Addall
9) Remove all
10) Collections — Sort
11) Add first
12) Add Last

Best Choise for Retrieval operation	Best for Insertion & deletion in middle
Worst Chose for Insertion & delition in middle	Worst Choise for Retreval operation
Default Capacity is 10	No default capacity

Vector → Vector is the Class Which implements List interface

ArrayList	Vector
DS – Resizable & growable	DS – Doubly type
Not Legency Class (~~because it is Not available~~)	Legency Class
Performace high	Performance low
$Ic = \frac{cc \times 3}{2} + 1$	$Ic = cc \times 2$
It is non synchronized & not thread safe	It is synchronised & thread safe
Because it is not present in previous Version of java ~~it is added is~~ ~~the~~ that is the reason it is not legency cla	It is available from previous Version

2) Set Interface →

- Set interface is present in java.util package & it extends Collection interface
- It represent the unorder set of element which doesn't allow us to store duplicate element
- We can store only one null value
- Implemented Class is
  a) Hashset
  b) Linked Hashset
  c) Treeset

Imp → Difference betn List & set interface →

List	Set

List	Set
Insertion order is preserved	Insertion order is not preserved
Multiple null element can be stored	Null element can store only once
Element by their position can be access.	Element by their position can not be access.
List interface can be implemented by Arraylist, Linklist & Vector	Set interface can be implemented by hashset, Linked hashset & Vector

Hashset →

- Hashset is the class which implements set interface
- Default capacity is 16 & load factor is 0.75
- Insertion order is not preserved

- Duplicates are not allowed
- Null elements can store only once
- get, set are not supported because Order of insertion is not preserved.
- If we ~~try~~ trying to add duplicate element in hashset using add method it returns false

method from hashset →

1) add
2) Remove → We can remove only by object not by index because insertion order is not pseved
3) Contains
4) is empty
5) Add All
6) Remove all
7) Retain All

Add All method → If we try to add all element from one hashset to another then it display only unique element because ~~hashset is~~ duplicates are not allowed in hashset

EX. → HS1.addAll(HS)
println(HS1)
It print only unique Value

Retain All method → If we are using retain all method in hashset then it displayed only common element

Linked hashset →

- It is the Class which implements Set interface
- Duplicates are not allowed
- Same method implemented which are present in hashset

- Insertion order is preserved
- Default capacity is 16 & LF is 0.75

Hashset	LinkedHashset
- Insertion order is not preserved	- Insertion order is preserved
- get & set method are not present	- get & set method are present.

3) **Queue Interface** →

Java queue interface orders the element in FIFO (first in first out order) manner.

Priority queue →

It is the Class which implements queue interface

Method from Priority queue

1) Adding element
   a) Add
   b) Offer

2) To Print head element
   c) element() → If queue is empty it will throw exception (No such element)
   d) Peek → If Empty it will return null

3) To remove head element
   e) Remove → If queue is ~~Element~~ empty it will throw exception.
   f) poll → If empty it will return Null.

Cursor ⟶

A java Cursor is an iterator, which is used to iterate or traverse or retrive a Collection elements one by one.

This are ⟶ a) Iterator
b) List iterator
c) Enumeration

Property	Enumeration	Iterator	Listiterator
Applicable for	only legacy Class	All Collect^n Classes	only for List Class
movement	only f/w direction.	only f/w	Both f/w & b/w direction
Accessibility	only read access	Both read & remove	Read, Remove & replace
method	2 method	3 method	9 method
	1) hasmore element 2) Next	hasnext next	Remove

Screenshot →

Why ? →

In Case of failed test cases we need Screenshot as a Proof to show the developer.

Imp → How to take Screenshot Using Selenium Webdriver?

1) To take Screenshot Using Selenium webdriver We need to type Cast driver object into TakesScreenshot interface (Type casting)

Ex. → ((TakesScreenShot) driver)

2) Then We need to Call function i.e getScreenshotAs() Here We need to pass the input / parameter

3) This method will return object of type File.

Ex. → File Source = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);

4) When this statement is execute it will take Screenshot but Screenshot available inside / local memory

5) To store screenshot in local driver we need to call a method "Copy()" which is present in "FileHandler" class

6) So this function will accept 2 parameter i.e Source & Dest.

Ex. FileHandler.Copy(Source, dest.);

Programe ⟶

```
driver.get(Url);

File Soure =((TakesScreenShot)driver).getScreenSh
        As(OutPutType.FILE)

File Dest == neW file("Destination Url")

FileHandler.copy(Soure, Dest);
```

Parameterization ⟶

- Para. in Selenium is process to Parameterize the test Scripts in order to pass multiple data to the application at runtime.

- Create an excel sheet With some data & Store it in any drive

1) In excel sheet row index & Coln index is starts from 0 in default.

2) Create an object of "FileInputstream" Class With excel sheet path as an input.

Ex. FileInputstream file = new FileInputstream();

3) To open an excel sheet we need to use a Static method i.e Create() Which is present in WorkbookFactory() Class.

4) To open specific excel sheet we need to use getsheet() method

5) To identify specific row in a sheet we need to use getRow() method.

6) To identify specific cell in a row we need to use getCell() method

7) To fetch string type of information we need to call getStringCellValue() method.

Ex. ⟶

```
FileInputStream file = new FileInputStream(Path)
                                    getSheet(Sheet)
String data = WorkbookFactor.Create("Sheet1").getRow(1)
        .getCell(1).getStringCellValue();
Println(data)
```

Java Switch Statement →

- The JSS executes one statement from multiple condition. ~~It is~~
- It is like if else-if ladder statement.
- The Switch statement works with byte, short, int, long & string.

Syntax →

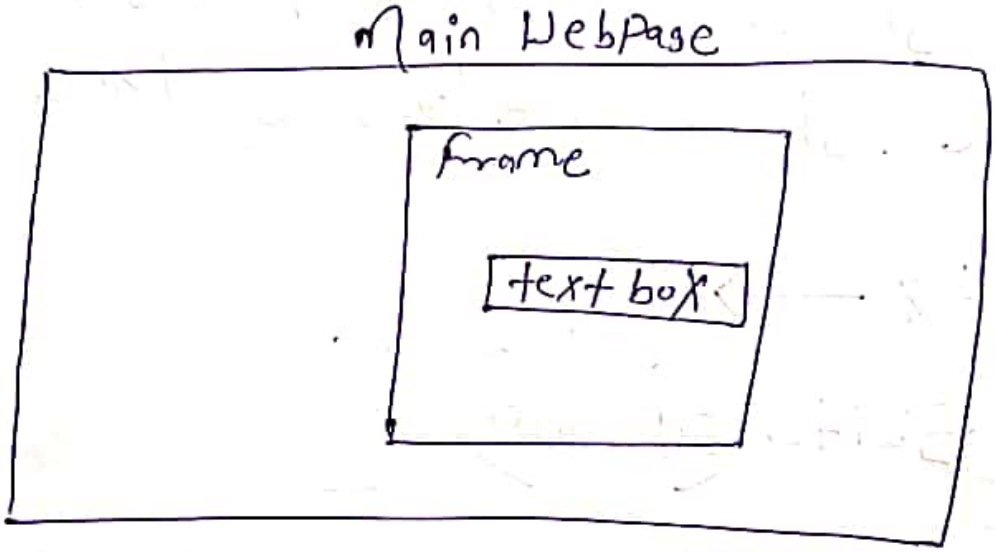
```
Switch(Expression) {

Case 1 :
    // Code to be executed
    break;
Case 2 :
    // ——————
    break;
Case 3 :
    // ——————
    break;
default :
    // Code to be executed if all
      these Cases are not matched.

}
```

Iframe ⟶

- Displaying WebPage as a part of another WebPage is known as a iframe.



How to handle iframe?

1) To handle iframe using Selenium Webdriver We need to switch Selenium focus from main page to frame by using

Syntax ⟶ driver.SwitchTo().frame();

2) In that frame we can pass different input like

- frame id
- frame Class
- frame index
- frame Webelement

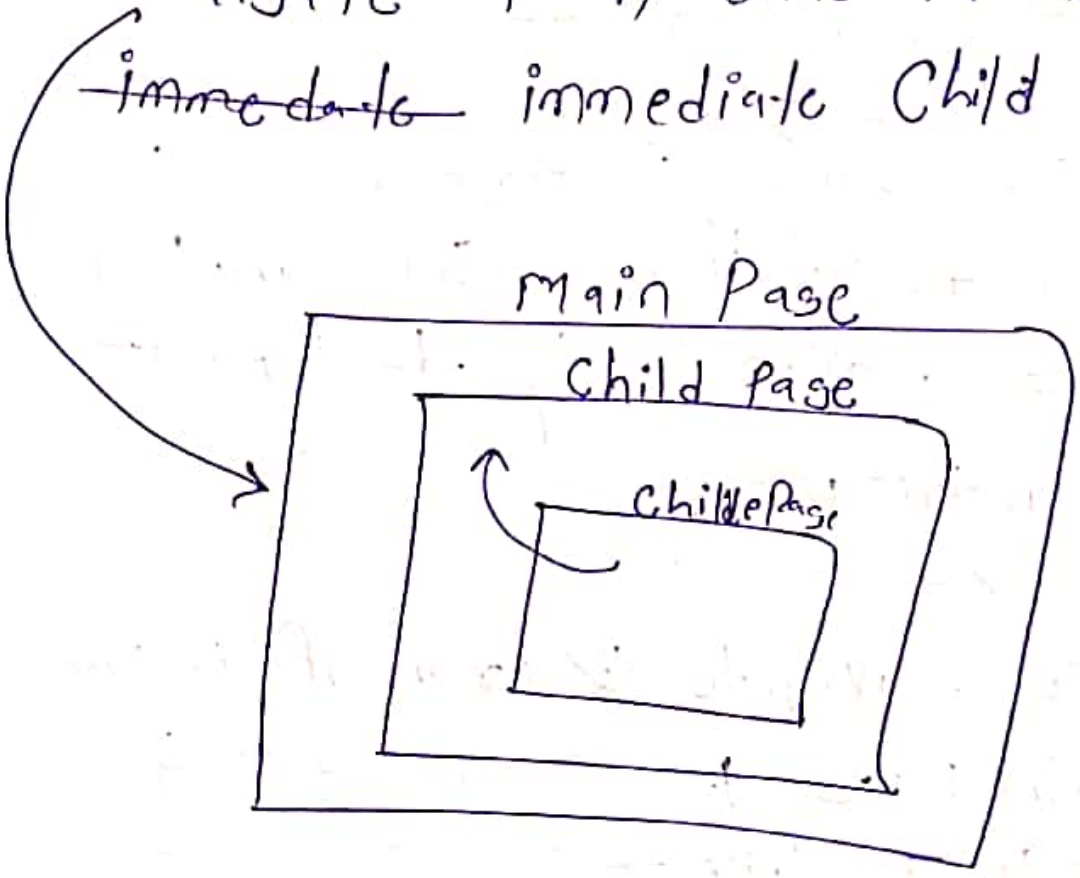
3) Once action performed on Component present in iframe Selenium. Will not navigate by ~~da~~ default to main page.

4) To navigate ~~to~~ from iframe to main page we need to use method like —

— defaultContent()
— parentFrame()

5) If we use driver.SwitchTo.defaultContent() then it will be navigate ~~to~~ from any Child to main page.

6) If we use driver.SwitchTo() - parentframe() the it will navigate from child frame to ~~immediate~~ immediate child frame



Pop Up ⟶

Alert ⟶ (Alert is interface)
An alert in selenium is a message box which appears on screen to give the user some information. It notifies the user ~~wh~~ ~~talk~~ with some specific information or error, asks for permission to perform certain tasks & it also provides warning messages as well.

How to handle Alert in selenium Webdriver ⟶

Alert interface provides the below few methods which are widely used in selenium Webdriver.

1) **Void dismiss ()** ⟶ To click on the "Cancle" button of the alert.
   - driver.SwitchTo().alert.dismiss()

2) **Void accept ()** ⟶ To click on the "Ok" button of the alert

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

3) String getText() → To capture the alert message.
- driver.switchTo().alert().getText();

4) Void Sendkeys() → To send some data to alert box.
- driver.switchTo().alert().Sendkeys();

There are 6 different types of pop-ups available in selenium →

a) Hidden division
b) Alert & Confirmation
c) File upload
d) Window based
e) Child browser
f) File Download

Note → for child browser we have to use WindowHandle/WindowHandles.

Window handle →

It is used to get the id of window to switch from one window to another window

Syntax → for id of single window

~~driver.switch~~

driver.getWindowHandle();

Syntax → for id of multiple windows

driver.getWindowHandles();

Imp

Get Window Handle	Get Window Handles
Return type → String	Return Type → Set<String>
It get the address of active browser	By using this we get address of all the browsers.
Syntax → String WID = driver.getWindowHandle	Syntax → Set<String> WID = driver.getWindowHandles

Difference betn findElement & findElements →

Parameter	findElement	findElements
Single WebElement Present	Return single Element	———
Multiple WebElement Present	Return 1st element	Return all element
Element is Not Present	Throws — NoSuchElement Exception	Does Not throw Exception Return Zero elements
Return Type	WebElement	List<WebElement>
Access	Direct Access	Using Iteration for loop

Actions Class

- AC is an ability provided by selenium for handling keyboard & mouse events.

It handles drag & drop, Clicking on multiple elements with the Control key, among others

Ac is defined & invoked using the syntax →

Actions AC = new Actions(driver);

methods of actions Class →

Action Class is useful mainly for mouse & keyboard actions. In order to perform such action selenium provides various methods.

- Mouse actions in Selenium →

1) moveToElement() → Shift the

2) doubleClick() → Perform double click on the element

3) ClickAndHold() → Performs long click on the Mouse without releasing it.

4) dragAndDrop() → Drags the element from one ~~from~~ point & drops to another.

5) ContextClick() → Performs right Click on the Mouse.

Keyboard Actions in Selenium →

1) Sendkeys() → Sends a series of keys to the element.

2) keyUP() → Perform key release.

3) keyDown() → Performs keyPress Without release.

Without is the difference between → build(). perform(). & perform

- buid() method in actions Class is Use to Create Chain of action.

- perform() method in actions class is use to execute chain of action which are build using action build method.

JavaScript Executer →

- It is interface
- JSE provides 2 methods

executscript- executAsyscript

to run a javascript on the selected window or current page.

Why we need JSE →

- In Selenium webdriver locators are used to identify & perform operations on webpage.
- In case these locators do not work we can use javascript executors to perform an desired operation on a web Element.

Selenium support JSE there is no need for an extra plugin or addon we just import script as to use JavascriptExecutor.

- JSE is also used for highlight border, Scroll down

1) Highlight border →

Syntax →

```
JavascriptExecutor JS = (JavascriptE... driver);
JS.Executescript("arguments[0].style
.border = 'pixel Solid Colour'", Args);
```

2) Highlight background →

Syntax. →

```
JS.Executescript("arguments[0].style.background
= 'pixel Solid Colour'", Args);
```

3) Scroll →

a) By ~~Element~~ Index →

i) Top to bottom →

Syntax →

JS. Executescript ("Window. ScrollBy(0, 7000)", "")

ii) Bottom to Top →

Syntax →

JS. Executscript (Window, scrollBy(0, -3000)", "")

b) Directly scroll →

i) From top to bottom →

Syntax →

JS. executescript ("Window. scrollBy (0, document. body. ScrollHeight)", "")

ii) From bottom to top →

JS. executescript ("document. documentElement. Scroll Top = 0", "")

c) Sc By Element →

Syntax →

JavascriptExecutor JS =

(JavascripExecutor) driver;

J.S. executescript ("arguments[0].ScrollIntoView();", WebElement);

4) Click Action →

JavascriptExecutor JS = (JavascriptExecutor) driver;

JS.executescript("arguments[0].Click()", Webelement);

5) Send keys ⟶

Syntax ⟶

JavascriptExecutor JS = (JavascriptExecutor) driver;

1st Way ⟶

~~JS.executescript("arguments[0]~~

J.executscript("document.getElement-By Id/Xpath('——').Value='Name';");

2nd Way ⟶

JS.executscript("arguments[0].Value='Name';", WebElement);

Q How to find the hidden element in selenium

⟶ By using Javascriptexecutor (Because it directly find-element from dom

Synchronization

- When 2 or more Components involved to perform any action, we expect these Components to work together with the same pace (गति). The Coordination betn these Components to run parallely is called synchronization.

What is synchronization problem in ATQ

- Sometimes Webpage takes more time than speed of Webdriver & in such cases we face exceptions such as NoSuchElementException or TimeoutException etc.

Use of synchronization ⟶

1) synchronizatn in Selenium helps the user to troubleshoot issues when launching or navigating to diff. Webpages while executing the Selenium Scripts.

2) At times, there can be a lot of Ajax Components or some images & when we want to interact with these elements it may not visible. Thus a time fallback can be seen in such cases we se "Element Not Visibe Exception".

- By using the synchronization (which selenium provided in form of wits) we solve the problem

Synchronization in the form of different types of Wait ⟶
(Wait in Selenium)

1) Implicit Wait
2) Explicit Wait
3) Fluent Wait

This Wait are dynamic Waits ⟶

EX ⟶ When we given Time out Value of 20 sec. If the element is loaded in 5 sec. the the rest 15sec. will be ignored.

1) Implicit Wait ⟶

An IW is to tell webdriver to poll the Dom for a Certain amt. of time when trying to find an Element if they are not immediately available.

syntax →

driver.manage().timeouts().

implicitly wait (Duration, ~~Timeout~~ TimeUnit.Sec)

- The default setting is 0.
- Once set the implicit ~~wai~~ wait is set for the life of WebDriver object instance.
- This is apply globally, ~~thatly~~ therefore it is applicable for all the ~~tube~~ web elements throughout the driver instance
- This will accept 2 parameter i.e duration & timeUnit.

2) **EXPLICIT wait** →

- EW waits for certain condition till specific element is not ~~four~~ loaded.

- This is used for certain element when those elelement is not available immediately & need some time..

Syntax ⟶

```
WebDriverWait MyWait =
    new WebDriverWait(driver, Duration.
                    Of Sec.(15));
WebElemen Element = MyWait.Until(ExpectedConditions.VisibilityOfElementLocated
                    (By id("idname")));
Element.Action();
```