# To automate the Amazon application using Selenium WebDriver with Java

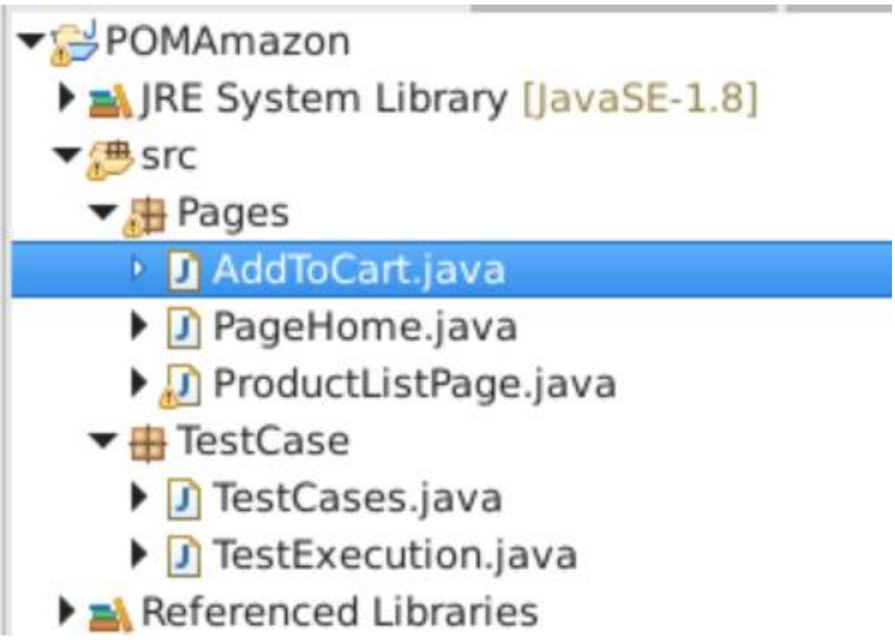
## NAME - SHASHANK SHEKHAR GUPTA

## Git- GITHUB LINK(https://github.com/birdss1/Automating-the-Amazon-application-using-Selenium-Webdriver.git)

**Description:** Automating the Amazon application to get the list of types of a particular product.

## Functionality:

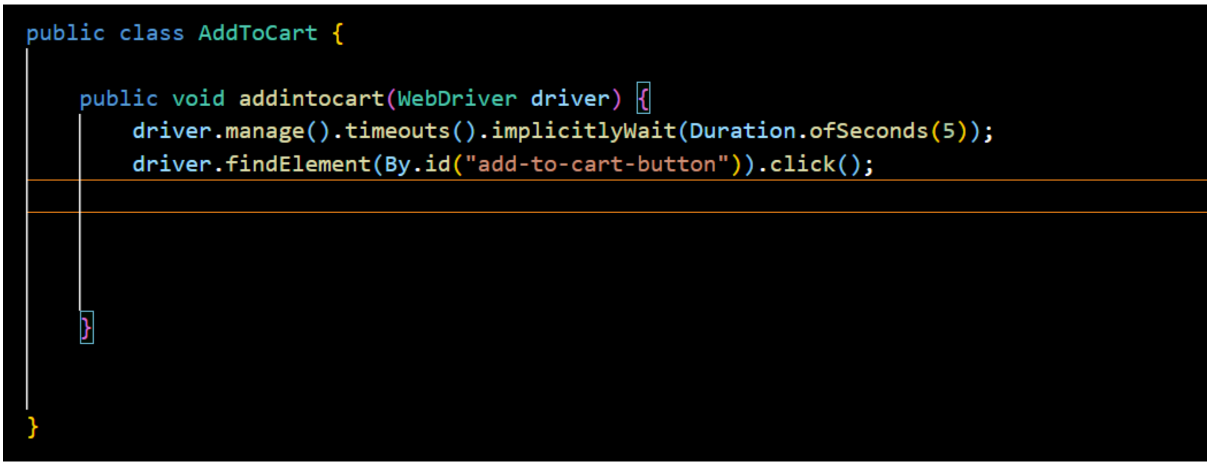
## Created 2 Packages  : Pages and Testcases



```
POMAmazon
  JRE System Library [JavaSE-1.8]
  src
    Pages
      AddToCart.java
      PageHome.java
      ProductListPage.java
    TestCase
      TestCases.java
      TestExecution.java
  Referenced Libraries
```

## Inside Pages: 3 classes

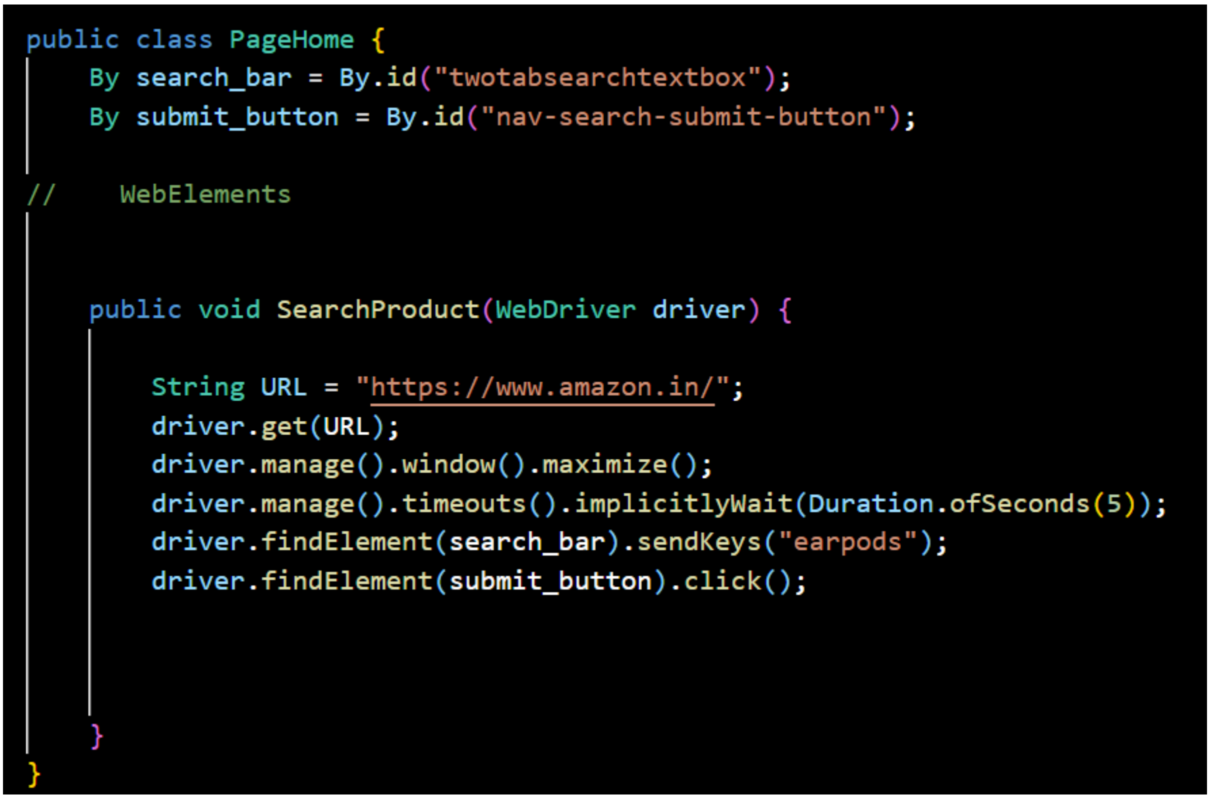
- **AddToCart: Insisde this there is one function called** addintocart. **Which will find the xpath of add to cart button and click on it . This is the last step of this project.**



```
public class AddToCart {

    public void addintocart(WebDriver driver) {
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));
        driver.findElement(By.id("add-to-cart-button")).click();




    }


}
```

1. **PageHome: Inside this there is two global variable names as** search_bar and submit_button. **There is one function name as** SearchProduct **which will open Amazon website then it will maximie window , also somtime our program execution happens faster but amazon**

website takes times to appear for that we will wait for max of 5 sec so that amazon website can appear , and then we will type earpods into search box and we search for it.



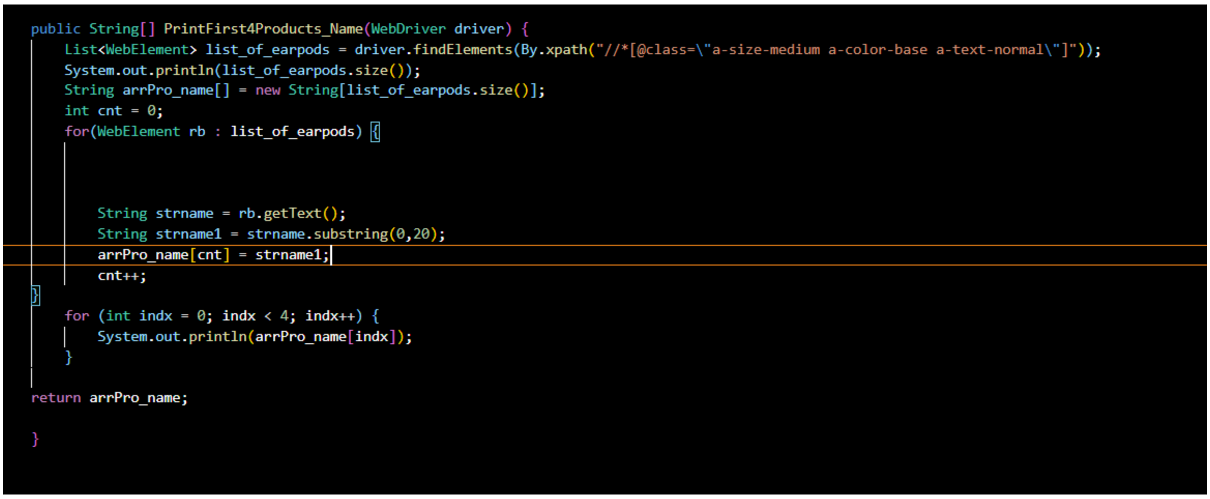
```
public class PageHome {
    By search_bar = By.id("twotabsearchtextbox");
    By submit_button = By.id("nav-search-submit-button");

//    WebElements


    public void SearchProduct(WebDriver driver) {

        String URL = "https://www.amazon.in/";
        driver.get(URL);
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));
        driver.findElement(search_bar).sendKeys("earpods");
        driver.findElement(submit_button).click();



    }
}
```

2. ProductListPage: Inside this we are doing our main work. There is 4 functions inside it , we will discuss about every function in detail here.

1. PrintFirst4Products_Name: **Here we are taking a array of webelement type and basically we are finding first 4 product appear first after searching for earpods .**

   **Also as name of some product are too big so we are only considering first 20 character of the entire string name.**

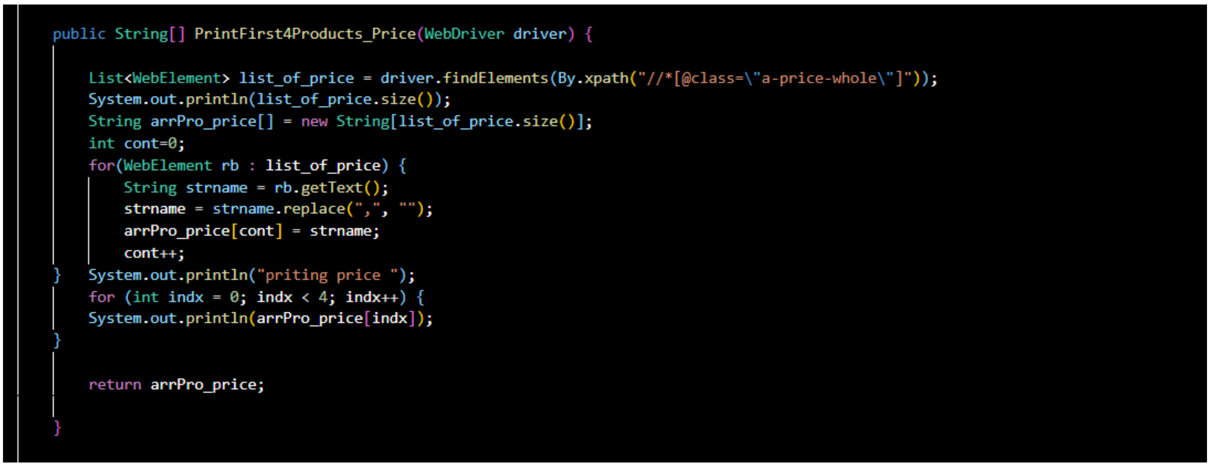
   **We will return string array from this function.**



```
public String[] PrintFirst4Products_Name(WebDriver driver) {
    List<WebElement> list_of_earpods = driver.findElements(By.xpath("//*[@class=\"a-size-medium a-color-base a-text-normal\"]"));
    System.out.println(list_of_earpods.size());
    String arrPro_name[] = new String[list_of_earpods.size()];
    int cnt = 0;
    for(WebElement rb : list_of_earpods) {



        String strname = rb.getText();
        String strname1 = strname.substring(0,20);
        arrPro_name[cnt] = strname1;
        cnt++;
}
    for (int indx = 0; indx < 4; indx++) {
        System.out.println(arrPro_name[indx]);
    }

return arrPro_name;

}
```

2. PrintFirst4Products_Price: **Here we are doing same instead of showing product name we are showing product price by same method as of previous function . Some time prices shown will be 6,999 so it is difficult to compare , so for that we are removing , from the price string .**

**We will return string array from this function.**



```
public String[] PrintFirst4Products_Price(WebDriver driver) {

    List<WebElement> list_of_price = driver.findElements(By.xpath("//*[@class=\"a-price-whole\"]"));
    System.out.println(list_of_price.size());
    String arrPro_price[] = new String[list_of_price.size()];
    int cont=0;
    for(WebElement rb : list_of_price) {
        String strname = rb.getText();
        strname = strname.replace(",", "");
        arrPro_price[cont] = strname;
        cont++;
}   System.out.println("priting price ");
    for (int indx = 0; indx < 4; indx++) {
    System.out.println(arrPro_price[indx]);
}


    return arrPro_price;

}
```

3. ValidateListofProducts: **Here we are doing validation of the product after comparing product name from amazon website and from our database as well as we also comparing product price from amazon website and from databses.**

   **If they matches then we will show output this product matches else not matches.**

   **Here we are considering 4 string arrays 1. Acquiring string array from** PrintFirst4Products_Name **function. 2. Acquiring  string array from** PrintFirst4Products_Price **function. 3.**

**Getting product name from database . 4. Getting product price from database.**

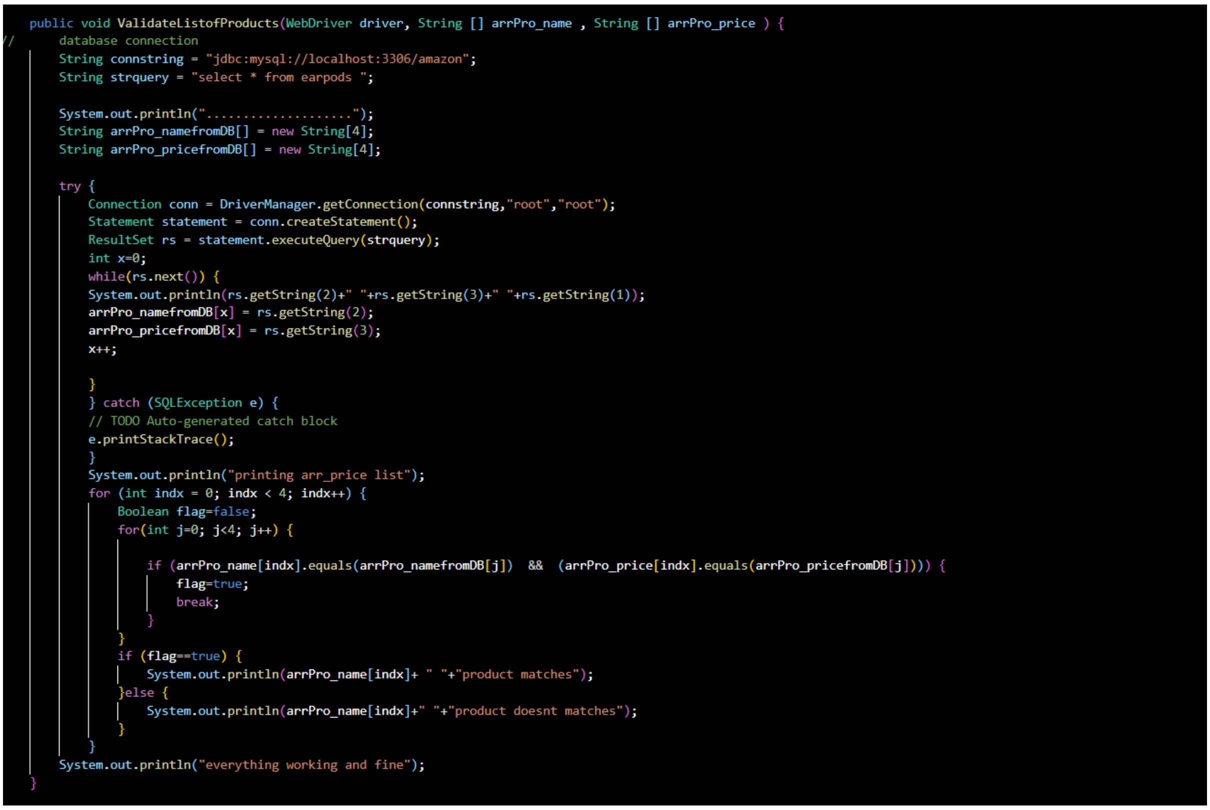
**Prerequisites : we have to create a database and table inside it and also put product name and product price manually so that we can validate.**

**Also here we are connecting our application to the database and getting product name and product price inside a list.**

**Then after having 4 string array we are ready to make comparison.**
**Earlier I used index comparison but due to very dynamic nature of amazon website I didn't get desired results the after ward i change the algorithm of comparison.**
**That comparison is like first we iterate over one string array the at the same time we will look for that string in another string array if that is found then only we will**

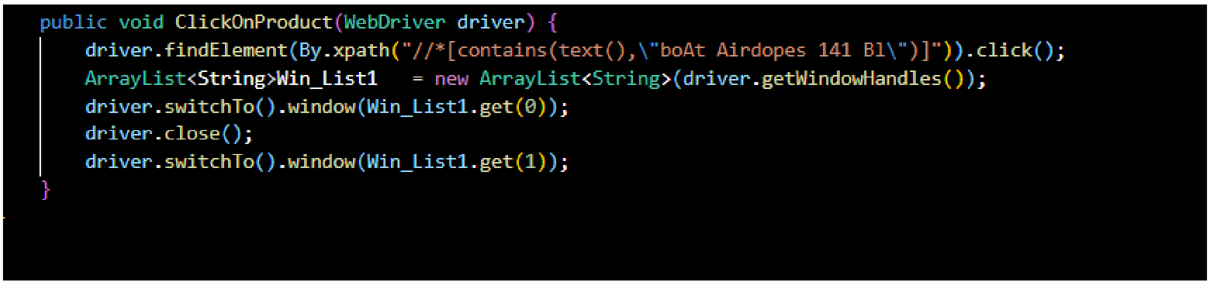
**print product matches. Same for product price also.**

**Although time complexity of comparison changes from o(n) to o(n*n) .**



```java
  public void ValidateListofProducts(WebDriver driver, String [] arrPro_name , String [] arrPro_price ) {
//      database connection
        String connstring = "jdbc:mysql://localhost:3306/amazon";
        String strquery = "select * from earpods ";

        System.out.println("....................");
        String arrPro_namefromDB[] = new String[4];
        String arrPro_pricefromDB[] = new String[4];

        try {
            Connection conn = DriverManager.getConnection(connstring,"root","root");
            Statement statement = conn.createStatement();
            ResultSet rs = statement.executeQuery(strquery);
            int x=0;
            while(rs.next()) {
            System.out.println(rs.getString(2)+" "+rs.getString(3)+" "+rs.getString(1));
            arrPro_namefromDB[x] = rs.getString(2);
            arrPro_pricefromDB[x] = rs.getString(3);
            x++;

            }
            } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            }
            System.out.println("printing arr_price list");
            for (int indx = 0; indx < 4; indx++) {
                Boolean flag=false;
                for(int j=0; j<4; j++) {

                    if (arrPro_name[indx].equals(arrPro_namefromDB[j])  &&  (arrPro_price[indx].equals(arrPro_pricefromDB[j]))) {
                        flag=true;
                        break;
                    }
                }
                if (flag==true) {
                    System.out.println(arrPro_name[indx]+ " "+"product matches");
                }else {
                    System.out.println(arrPro_name[indx]+" "+"product doesnt matches");
                }
            }
        System.out.println("everything working and fine");
  }
```

**4.** ClickOnProduct: **Inside this we are passing xpath  of  any product as mentioned by our instructor and click on it . then it will show that product details inside another woindow. So first I close first window then I get into second window using switch function.**

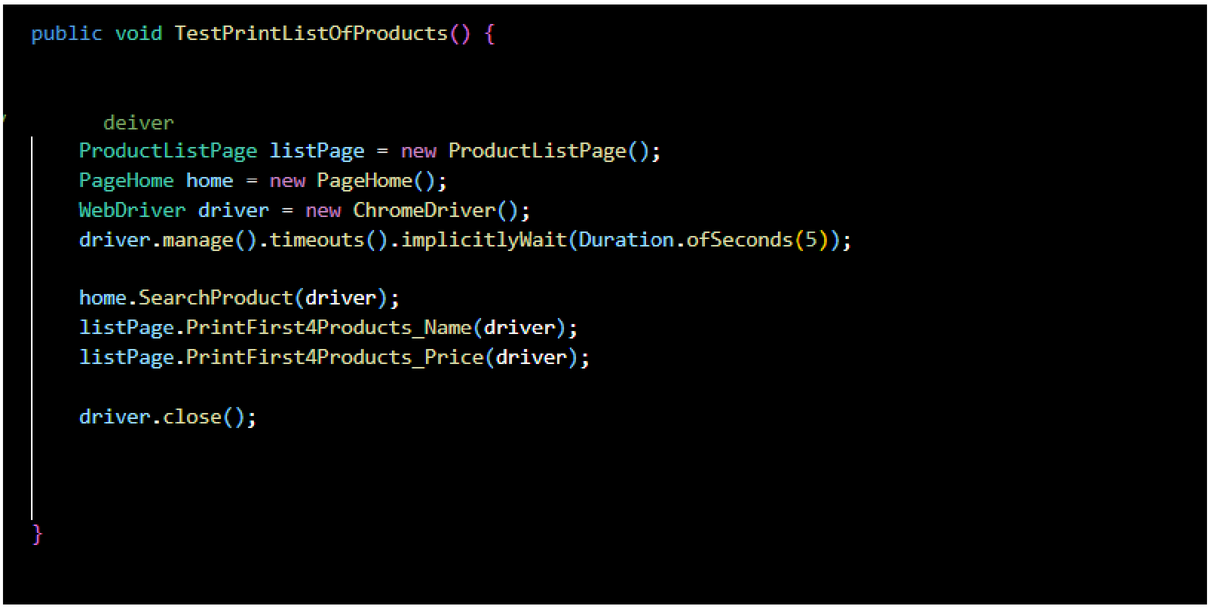


```
public void ClickOnProduct(WebDriver driver) {
    driver.findElement(By.xpath("//*[contains(text(),\"boAt Airdopes 141 Bl\")]")).click();
    ArrayList<String>Win_List1   = new ArrayList<String>(driver.getWindowHandles());
    driver.switchTo().window(Win_List1.get(0));
    driver.close();
    driver.switchTo().window(Win_List1.get(1));
}
```

# Inside TestCase: 2 classes

1. Testcases: **Inside this we have 3 classes.**

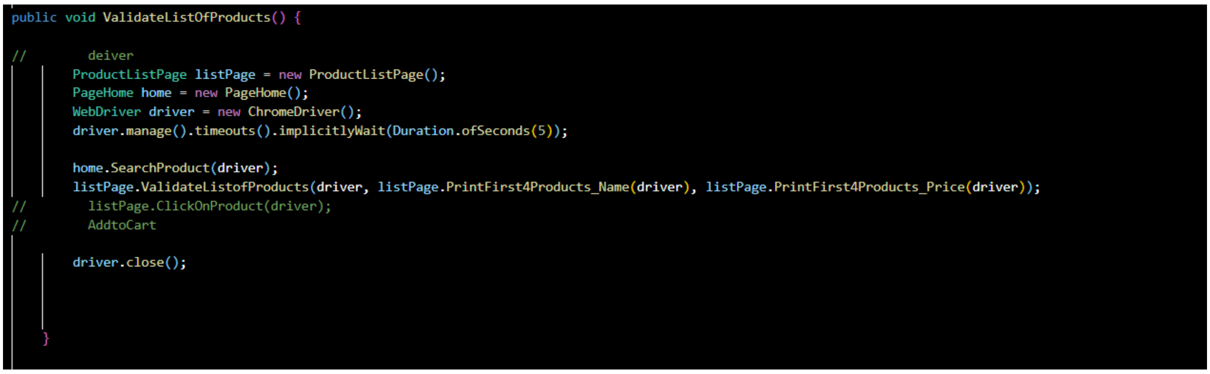
   **1**.TestPrintListOfProducts: **It will print first 4 product name and product price and then we close this driver. We are just calling desired function to do so.**



```
public void TestPrintListOfProducts() {


      deiver
    ProductListPage listPage = new ProductListPage();
    PageHome home = new PageHome();
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));

    home.SearchProduct(driver);
    listPage.PrintFirst4Products_Name(driver);
    listPage.PrintFirst4Products_Price(driver);

    driver.close();



}
```

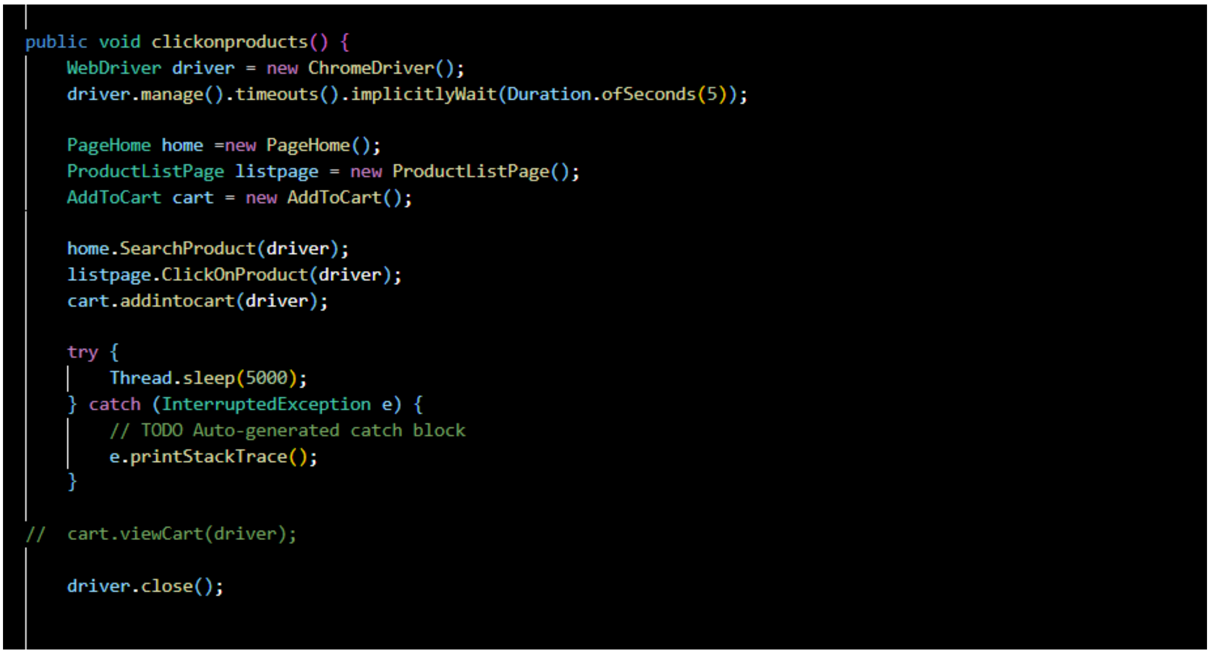
2. ValidateListOfProducts: **Here we are calling desired function so that we can**

**validate product , connect to database and show product details.**



```
public void ValidateListOfProducts() {

//        deiver
        ProductListPage listPage = new ProductListPage();
        PageHome home = new PageHome();
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));

        home.SearchProduct(driver);
        listPage.ValidateListofProducts(driver, listPage.PrintFirst4Products_Name(driver), listPage.PrintFirst4Products_Price(driver));
//          listPage.ClickOnProduct(driver);
//          AddtoCart

        driver.close();



    }
```

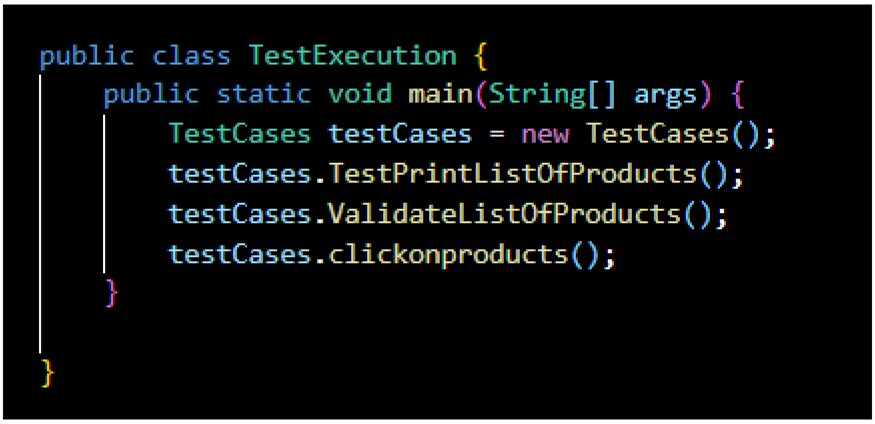
**3.** clickonproducts: **Here  we are clicking on the product and then adding into cart the cart.**



```
public void clickonproducts() {
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));

    PageHome home =new PageHome();
    ProductListPage listpage = new ProductListPage();
    AddToCart cart = new AddToCart();

    home.SearchProduct(driver);
    listpage.ClickOnProduct(driver);
    cart.addintocart(driver);

    try {
        Thread.sleep(5000);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

//  cart.viewCart(driver);

    driver.close();
```

**4.** TestExecutive: **There is only one function inside it i.e. main function . we don't have main function until in any cprevious classes , here only we have main function, so that we can execute**

**program from here only. So we calling desired testcase function here and letting program to run and show the ouput .**



```
public class TestExecution {
    public static void main(String[] args) {
        TestCases testCases = new TestCases();
        testCases.TestPrintListOfProducts();
        testCases.ValidateListOfProducts();
        testCases.clickonproducts();
    }

}
```