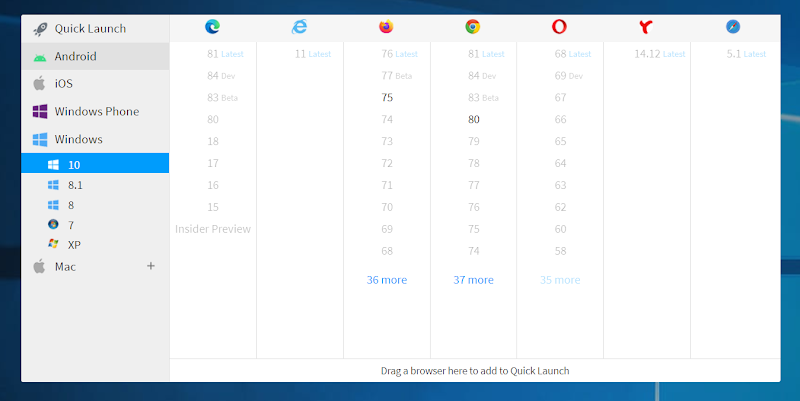
-------------------------------------------------------------------------------------------------------------------------------

**Multiple Browser**

**Factory Pattern**

**Web Driver Manager**



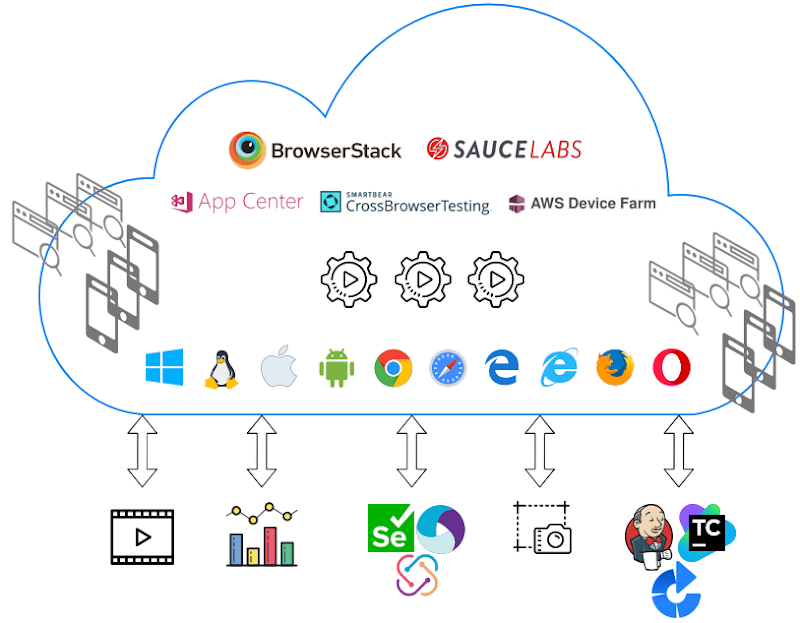
***DAO MINH DAM  
Email:***[*daominhdam@gmail.com*](mailto:daominhdam@gmail.com)

***Skype:****daominhdam (Đào Minh Đảm)*

[***https://automationfc.vn***](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://automationfc.vn%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332486000%26amp;usg%3DAOvVaw0ULTcBPsKaFCYYalgO8eGb&sa=D&source=editors&ust=1628668332554022&usg=AOvVaw0C4Ib8gzaNTzIzLxuY7OOd)*-*[***https://automationfc.com***](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://automationfc.com%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332486000%26amp;usg%3DAOvVaw3mfpJ6IB0_JARld0RwKsMJ&sa=D&source=editors&ust=1628668332554142&usg=AOvVaw2RfHTonxZqYBHw4A2Zfd48)

***Tài liệu này chỉ sử dụng trong khóa học - vui lòng không chia sẻ công khai***  
-------------------------------------------------------------------------------------------------------------------------------

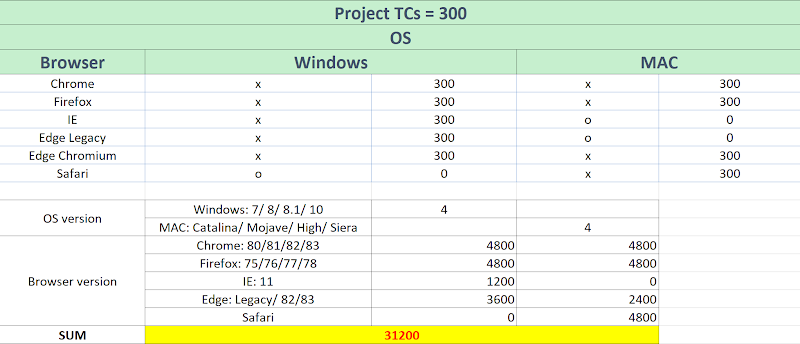
**Cloud Testing Service**



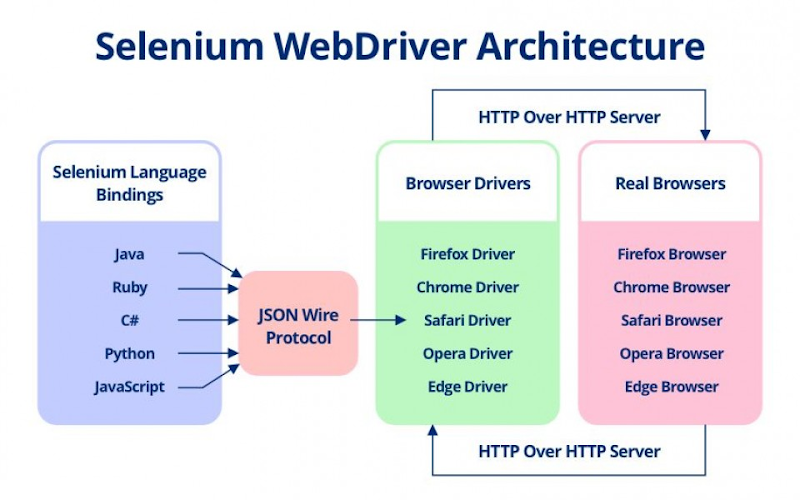
* **Browserstack**
* **Saucelab**
* **Device Farm (Amazon)**
* **Perfector**
* **Kobiton (KMS)**
* **Cross Browser Testing (Smartbear)**
* **Xamarin Test Cloud (Microsoft)/ Visual Studio App Center**
* Reference: [Link](https://blog.testproject.io/2020/02/04/top-5-solutions-executing-test-automation-cloud-comprehensive-comparison/)



**Config multiple browser**



* **Link -** [https://docs.google.com/spreadsheets/d/1UrOS4b\_UysxRrDJjFN7F3kP5n-ei3zQC/edit](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://docs.google.com/spreadsheets/d/1UrOS4b_UysxRrDJjFN7F3kP5n-ei3zQC/edit%2523gid%253D345910041%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332488000%26amp;usg%3DAOvVaw3-2dwywd37uhabrsD8tmmw&sa=D&source=editors&ust=1628668332554906&usg=AOvVaw0_2znY9iaGlfrCpPxfmnil)
* **Ngay từ đầu phải clear vs khách hàng/ team là sẽ làm trên những browser nào:**
* Không được ảo tưởng sức mạnh là làm được càng nhiều càng good
* Đừng bao giờ hỏi khách hàng là muốn làm trên bao nhiêu browser?
* Nên làm trên browser nào chính nhất - đang support cho dự án tốt nhất (Khuyến khích nên làm cho Chrome -> ổn định nhất)
* Nếu như define làm nhiều browser ngay từ đầu - thì phải scripting và run pass được trên tất cả các browser đó thì mới đánh Done 1 chức năng đã làm xong
* Chrome/ Firefox nó tỉ lệ pass ngang nhau nhưng qua chạy IE thì fail rất nhiều (Mất thêm khoảng 30-40% effort để làm thêm cho IE) -> Estimate ngay từ đầu



* Selenium 3.xx trở xuống thì vẫn support cho Opera/ PhantomJS
* Selenium 4.xx trở lên ngừng support Opera/ PhantomJS
* Headless browser dùng cho bên FrontEnd để run Unit Test/ Data Analys
* **At testcase layer**
* Each test class
* Note: equalsIgnoreCase - không phân biệt hoa thường
* browserName.equalsIgnoreCase("firefox")
* firefox
* Firefox
* FIREFOX
* fIREfox
* Để chạy được chrome headless thì trước khi khởi tạo browser, cần thêm 2 tham số vào ChromeOptions
* Headless mode has been available on Mac and Linux since Chrome 59. Windows support came in Chrome 60
* headless: trình duyệt sẽ thực thi ở chế độ headless
* window-size: cần set trình duyệt với 1 resolution cố định (nếu ko thì nó có thể hiển thị kích thước như phiên bản web mobile hoặc ở 1 độ phân giải ko cố định) – có thể sử dụng tính năng này để test responsive
* Tham khảo bài viết về Chrome Headless - [Link](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://automationfc.com/2017/11/16/java-webdriver-13-run-test-voi-trinh-duyet-chrome-headless/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332490000%26amp;usg%3DAOvVaw2XydOzZ5m_lNevCuPh82tr&sa=D&source=editors&ust=1628668332555882&usg=AOvVaw0ETGSAo2ILPOmvzT-fXd2M)

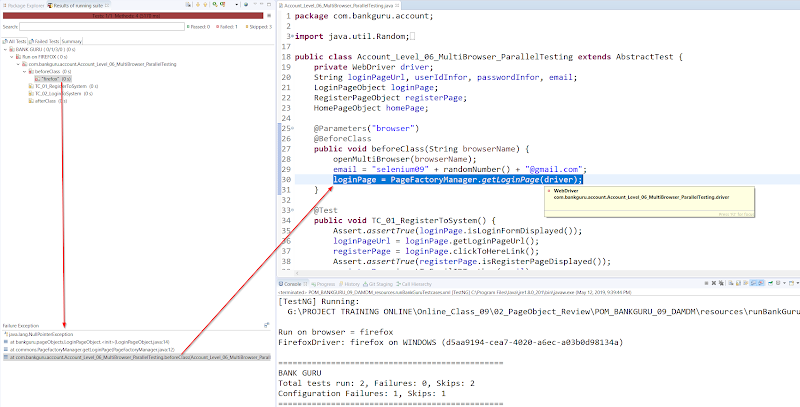
|  |
| --- |
| ChromeOptions chromeOpt = new ChromeOptions(); chromeOpt.addArguments("headless"); chromeOpt.addArguments("window-size=1920x1080"); driver = new ChromeDriver(chromeOpt);  FirefoxOptions firefoxOpt = new FirefoxOptions(); firefoxOpt.addArguments("-headless"); firefoxOpt.addArguments("window-size=1920x1080"); driver = new FirefoxDriver(firefoxOpt); |

|  |
| --- |
| if (browserName.equalsIgnoreCase("firefox")) {                 driver = new FirefoxDriver();         } else if (browserName.equalsIgnoreCase("chrome")) {                 System.setProperty("webdriver.chrome.driver", ".\\resources\\chromedriver.exe");                 driver = new ChromeDriver();         } else if (browserName.equalsIgnoreCase("ie")) {                 System.setProperty("webdriver.chrome.driver", ".\\resources\\IEDriverServer.exe");                 driver = new InternetExplorerDriver();         } else if (browserName.equalsIgnoreCase("chromeheadless")) {                 System.setProperty("webdriver.chrome.driver", ".\\resources\\chromedriver.exe");                 ChromeOptions options = new ChromeOptions();                 options.addArguments("headless");                 options.addArguments("window-size=1366x768");                 driver = new ChromeDriver(options); } |

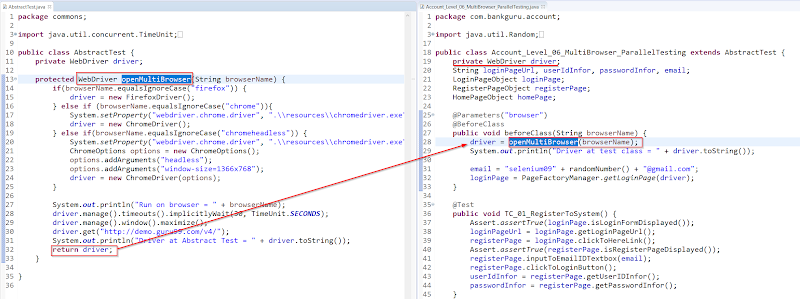
* **Run testcase (.xml)**



* **Nhược điểm:**
* Phải đưa vào Before<xxx> cho từng testclass - khi có sự thay đổi phải update cho từng class
* Khi số lượng browser tăng lên khó quản lí (thêm/ sửa/ xóa)
* Chrome
* Firefox
* IE
* Edge
* Safari
* Opera
* Headless (Chrome/ Firefox)
* Android
* iOS
* Remote
* **At AbstractTest**
* Các hàm dùng chung cho tầng page - viết trong AbstractPage
* Các class pageObject kế thừa AbstractPage => **Nguyên tắc kế thừa (Inheritance)**
* Các hàm dùng chung cho tầng testcase - viết trong AbstractTest
* Các class test ở tầng testcase kế thừa AbstractTest => **Nguyên tắc kế thừa (Inheritance)**
* Sử dụng **Factory Pattern**
* Đưa việc khởi tạo driver/ browser vào thành hàm common trong AbstractTest
* Giải quyết được việc **che giấu sự khởi tạo của đối tượng** ở tầng testcase => **Nguyên tắc đóng gói (Encapsolution in OOP)**
* Dễ dàng sử dụng và bảo trì
* Issue khi apply:
* NullPointerException
* Chưa map driver giữa 2 tầng testcase và AbstractTest



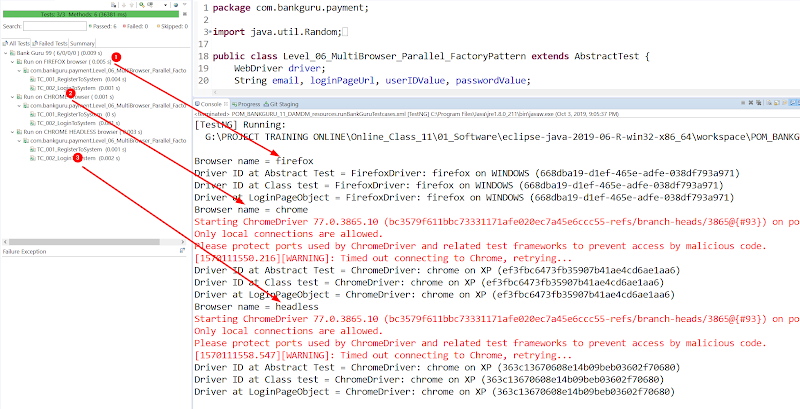
* Xử lí:
* Làm sao để lấy được driver ID từ AbstractTest truyền qua cho test class để chạy
* Return driver ở AbstractTest và ở tầng testcase sẽ get lại



|  |
| --- |
| package commons;  import java.util.concurrent.TimeUnit; import org.openqa.selenium.WebDriver; import org.openqa.selenium.chrome.ChromeDriver; import org.openqa.selenium.chrome.ChromeOptions; import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.ie.InternetExplorerDriver;  public class AbstractTest {         private WebDriver driver;                  protected WebDriver getBrowserDriver(String browserName) {                 if (browserName.equalsIgnoreCase("firefox")) {                         driver = new FirefoxDriver();                 } else if (browserName.equalsIgnoreCase("chrome")) {                         System.setProperty("webdriver.chrome.driver", ".\\resources\\chromedriver.exe");                         driver = new ChromeDriver();                 } else if (browserName.equalsIgnoreCase("ie")) {                         System.setProperty("webdriver.chrome.driver", ".\\resources\\IEDriverServer.exe");                         driver = new InternetExplorerDriver();                 } else if (browserName.equalsIgnoreCase("cheadless")) {                         System.setProperty("webdriver.chrome.driver", ".\\resources\\chromedriver.exe");                         ChromeOptions options = new ChromeOptions();                         options.addArguments("headless");                         options.addArguments("window-size=1920x1080");                         driver = new ChromeDriver(options);                 }                  driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);                 driver.manage().window().maximize();                 driver.get(Constants.DEV\_URL);                 return driver;         } } |

* **Cách dùng tại Test Class**
* 1 - Kế thừa từ class cha để sử dụng các thuộc tính và phương thức của AbstractTest
* 2 - Sử dụng annotation @Parameter của TestNG để get ra giá trị của biến browser trong file runTestcase.xml
* 3 - Map giá trị từ Parameter vào tham số browserName của phương thức beforeClass
* 4 - Gán giá trị browserName (chrome/ firefox/ chromeheadless/..) vào hàm **getBrowserDriver**để khởi tạo đúng browser cần sử dụng
* 5 - Sau khi gán vào hàm **getBrowserDriver**thì driver sẽ được khởi tạo trong AbstractTest và trả về giá trị - giá trị được trả về sẽ gán vào cho biến driver tại Test Class để chạy tiếp đúng luồng trong testcase





**Enum**

* Kiểu dữ liệu đặc biệt được sử dụng để đại diện cho hằng số cố định
* PI = 3.141592653589793
* Thứ trong tuần (MONDAY/ TUESDAY/ WEDNESDAY/THURSDAY/..)
* Mùa trong năm (SPRING/ SUMMER/ WINTER/..)
* Loại Browser (CHROME/ FIREFOX/ IE/ SAFARI/ OPERA/..)
* Loại OS (WIN/ MAC/ UNIX/ LINUX/ SOLARIS)
* Loại Platform (IOS/ ANDROID/ WPHONE)
* Môi trường test (DEV/ TEST/ STAGING/ PROD/ LIVE)
* Các giá trị của Enum là các hằng số - tên của các trường kiểu enum là các chữ cái hoa
* LOGIN\_BUTTON
* NEW\_CUSTOMER\_PAGE
* Hằng số trong Java: static final
* public static final double PI = 3.14;
* Thay vì truyền vào 1 chuỗi để kiểm tra tên browser (phải nhớ được tên browser) ở nhiều vị trí (Class khác nhau) thì sử dụng qua enum (hạn chế sự thay đổi từ 1 team member khác)

|  |
| --- |
| public enum BROWSER {         CHROME, FIREFOX, IE, SAFARI, EDGE, CHEADLESS, FHEADLESS } |

|  |
| --- |
| protected WebDriver getBrowserDriver(String browserName) {                 BROWSER browser = BROWSER.valueOf(browserName.toUpperCase());                  if (browser == BROWSER.FIREFOX) {                         WebDriverManager.firefoxdriver().setup();                         driver = new FirefoxDriver();                 } else if (browser == BROWSER.CHROME) {                         WebDriverManager.chromedriver().setup();                         driver = new ChromeDriver();                 } else if (browser == BROWSER.EDGE) {                         WebDriverManager.edgedriver().setup();                         driver = new EdgeDriver();                 } else if (browser == BROWSER.IE) {                         WebDriverManager.iedriver().arch64().setup();                         driver = new EdgeDriver();                 } else if (browser == BROWSER.SAFARI) {                         driver = new SafariDriver();                 } else if (browser == BROWSER.CHEADLESS) {                         WebDriverManager.chromedriver().setup();                         ChromeOptions options = new ChromeOptions();                         options.addArguments("headless");                         options.addArguments("window-size=1920x1080");                         driver = new ChromeDriver(options);                 }         driver.manage().timeouts().implicitlyWait(GlobalConstants.LONG\_TIMEOUT, TimeUnit.SECONDS);         driver.get(GlobalConstants.AUT\_URL);         return driver;         } |

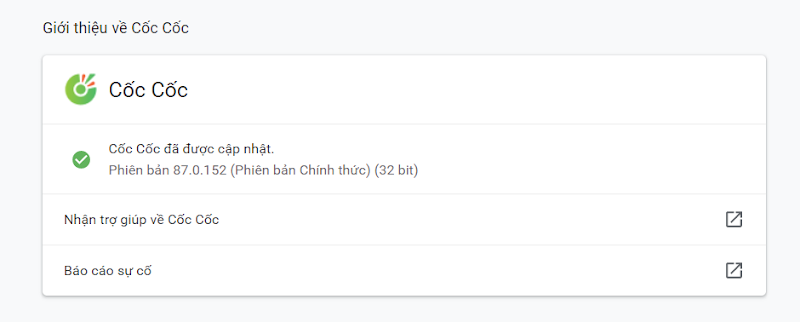
* Sử dụng vs switch/case

|  |
| --- |
| protected WebDriver getBrowserDriver(String browserName) {                 BROWSER browser = BROWSER.valueOf(browserName.toUpperCase());                  switch (browser) {                 case FIREFOX:                         WebDriverManager.firefoxdriver().setup();                         driver = new FirefoxDriver();                         break;                 case CHROME:                         WebDriverManager.chromedriver().setup();                         driver = new ChromeDriver();                         break;                 case EDGE:                         WebDriverManager.edgedriver().arch64().setup();                         driver = new EdgeDriver();                         break;                 case IE:                         WebDriverManager.iedriver().arch64().setup();                         driver = new EdgeDriver();                         break;                 case SAFARI:                         driver = new SafariDriver();                         break;                 case CHEADLESS:                         WebDriverManager.chromedriver().setup();                         ChromeOptions options = new ChromeOptions();                         options.addArguments("headless");                         options.addArguments("window-size=1920x1080");                         driver = new ChromeDriver(options);                         break;                 default:                         break;                 }          driver.manage().timeouts().implicitlyWait(GlobalConstants.LONG\_TIMEOUT, TimeUnit.SECONDS);         driver.get(GlobalConstants.WORDPRESS\_URL);         return driver;         } |

[https://blog.testproject.io/2020/08/04/using-java-enums-in-test-automation](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://blog.testproject.io/2020/08/04/using-java-enums-in-test-automation%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332502000%26amp;usg%3DAOvVaw1oFyHmeJiZlyXQQQirVSfB&sa=D&source=editors&ust=1628668332561221&usg=AOvVaw20eUtDfKjHlNIfqCcWkbVx)

**Cốc Cốc Browser**

* Kiểm tra Cốc cốc version



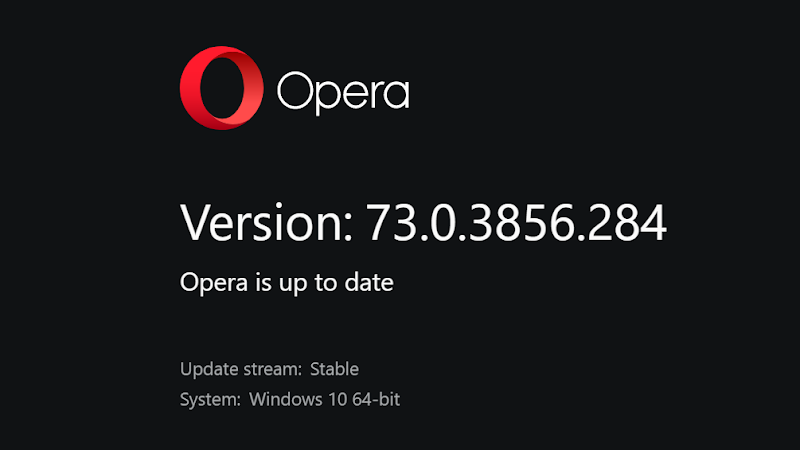
* Setting để khởi tạo/ start Cốc Cốc browser lên

|  |
| --- |
| System.setProperty("webdriver.chrome.driver", ".\\browserDrivers\\chromedriver 81.exe"); ChromeOptions options = new ChromeOptions();         options.setBinary("C:\\Program Files (x86)\\CocCoc\\Browser\\Application\\browser.exe"); driver = new ChromeDriver(options); |

* Tải đúng version của ChromeDriver về
* NOTE: Cứ lấy version hiện tại trừ đi 5 là ra driver version tương ứng
* Cốc cốc version: 87 - ChromeDriver: 81 (81.0.4044.69)
* Cốc cốc version: 91 - ChromeDriver: 85 (85.0.4183.38)

**Opera Browser**

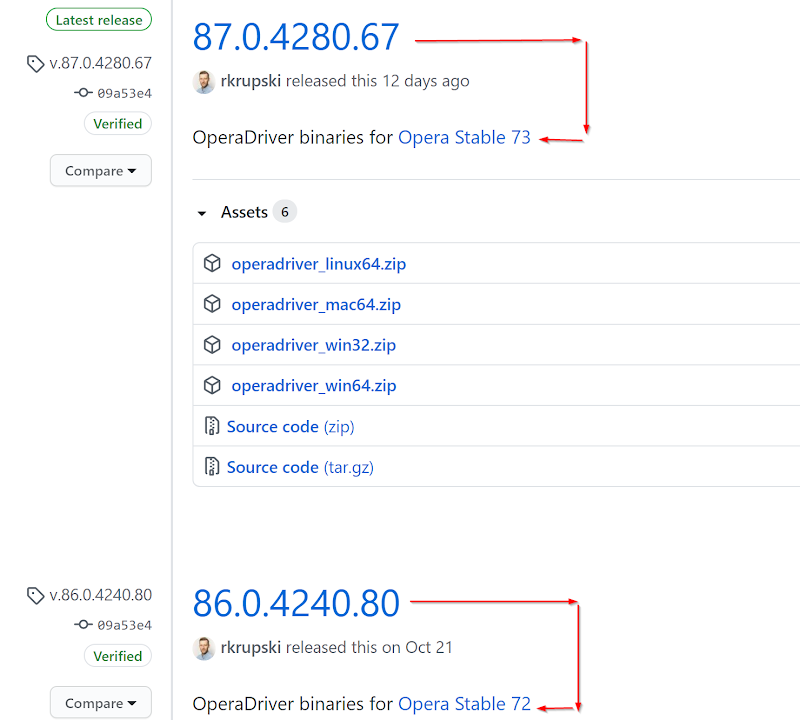
* Kiểm tra Opera version

****

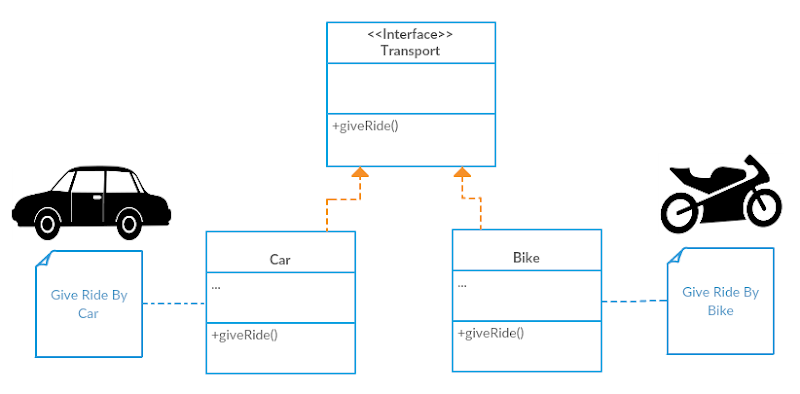
* Setting để tải về Opera Driver

|  |
| --- |
| System.setProperty("webdriver.opera.driver", ".\\browserDrivers\\operadriver.exe"); driver = new OperaDriver(); |

* Kiểm tra Opera Driver version
* [https://github.com/operasoftware/operachromiumdriver/releases](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://github.com/operasoftware/operachromiumdriver/releases%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332505000%26amp;usg%3DAOvVaw0ijqs99LVomDzEhtSotlfk&sa=D&source=editors&ust=1628668332562294&usg=AOvVaw1dnq9H4R6mx7J9Ut4gA1kb)
* [http://npm.taobao.org/mirrors/operadriver/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://npm.taobao.org/mirrors/operadriver/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332505000%26amp;usg%3DAOvVaw1E9jqsw3Yq3RFvxIyu3tQ2&sa=D&source=editors&ust=1628668332562429&usg=AOvVaw1CBghfXDcA0jfPhgHRwsV6)

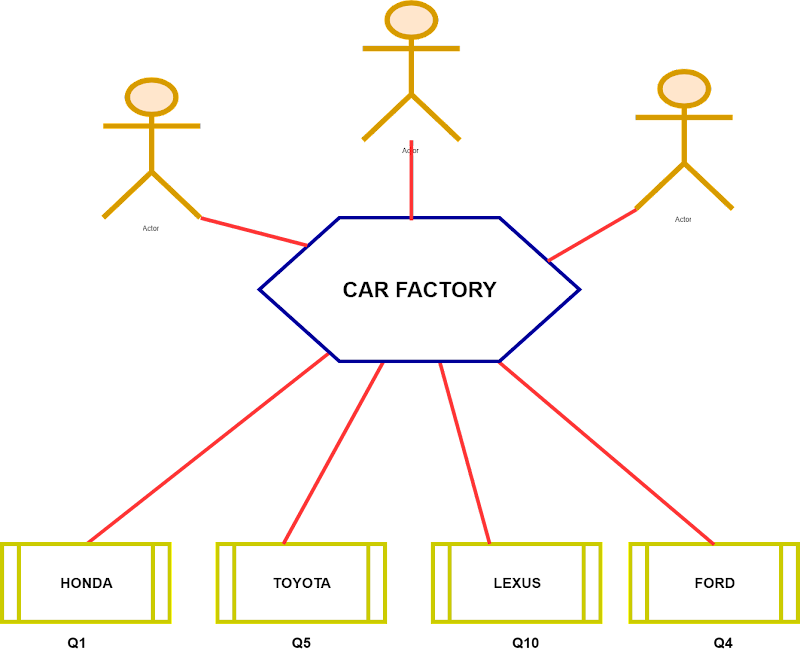


**Factory Pattern**

****

* **Bài toán mua xe ôtô**

|  |
| --- |
| *- Hai vợ chồng A đang sinh sống tại Sài Gòn có nhu cầu mua một chiếc oto, vì chưa có kinh nghiệm gì về xe oto và các hãng xe nên quyết định sẽ đi tới các cửa hàng để xem 1 số loại xe trước khi quyết định mua xe nào. - Buổi sáng 2 vợ chồng ghé qua cửa hàng xe Honda ở quận 1 để xem về các mẫu xe của loại xe Honda, buổi chiều họ ghé qua cửa hàng xe Toyota ở quận 5, buổi tối họ ghé qua cửa hàng xe Ford ở quận 4 và ngày mai sẽ qua cửa hàng xe Lexus tại quận 10*  *- Vì các cửa hàng không nằm tại cùng 1 vị trí nên việc đi lại giữa các cửa hàng tốn rất nhiều thời gian, và trường hợp muốn xem thêm loại xe khác thì khó khăn trong việc đi lại*  *- Làm sao giải quyết được bài toán họ không cần phải đi qua tất cả các cửa hàng mà vẫn có thể xem được bất kì hãng xe mà họ muốn (Ford/ Honda/ Lexus/..) và khi có 1 hãng xe mới vẫn có thể thêm vào cho khách hàng chọn lựa. => Tập trung tất cả các loại xe về 1 địa điểm (Factory - Nhà máy sản xuất) - khi khách hàng yêu cầu xem loại xe nào chỉ cần gọi đúng ra hãng xe (tên loại) mà họ cần và lấy ra để thao tác. Khi thêm vào 1 hãng xe mới không ảnh hưởng đến những chức năng cũ (vẫn hoạt động được)* |



* Không apply Factory Pattern:

|  |
| --- |
| public abstract Car {    abstract void view(); } |

|  |
| --- |
| public class Honda extends Car {    @Override    public void view() {        System.out.printf("Honda view");    } } |

|  |
| --- |
| public class Lexus extends Car {    @Override    public void view() {        System.out.printf("Lexus view");    } } |

|  |
| --- |
| public class Toyota extends Car {    @Override    public void view() {        System.out.printf("Toyota view");    } } |

|  |
| --- |
| public class Customer {    public void viewHonda() {        Honda honda = new Honda();        honda.view();    }    public void viewNexus() {        Lexus nexus = new Lexus();        lexus.view();    }    public void viewToyota(){        Toyota toyota = new Toyota();        toyota.view();    } } |

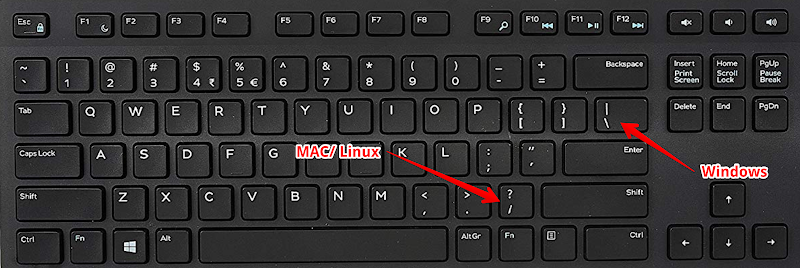
* **Apply Factory Pattern**

|  |
| --- |
| public class CarFactory {     public void viewCar(String carType) {        Car car;        if (carType.equalsIgnoreCase("HONDA")) {            car = new Honda();            car.view();        } else if (carType.equalsIgnoreCase("LEXUS")) {            car = new Lexus();            car.view();        } else if (carType.equalsIgnoreCase("TOYOTA")) {            car = new Toyota();            car.view();        }    } }  public class Customer {         public static void main(String[] args) {                CarFactory carFactory = new CarFactory();                carFactory.viewCar("HONDA");                carFactory.viewCar("LEXUS");                carFactory.viewCar("FORD");         } } |

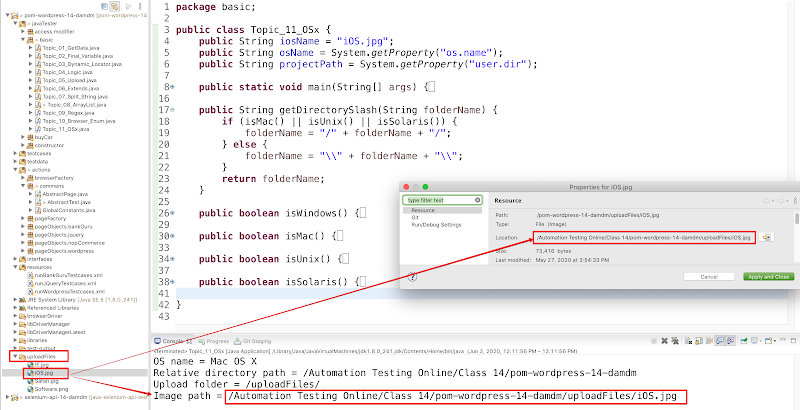
* **AbstractTest (Factory Pattern)**

|  |
| --- |
| protected WebDriver openMultiBrowser(String browserName) {                 if (browserName.equalsIgnoreCase("firefox")) {                         driver = new FirefoxDriver();                 } else if (browserName.equalsIgnoreCase("chrome")) {                         System.setProperty("webdriver.chrome.driver", "..\\chromedriver.exe");                         driver = new ChromeDriver();                 } else if (browserName.equalsIgnoreCase("ie")) {                         System.setProperty("webdriver.chrome.driver", "..\\IEDriverServer.exe");                         driver = new InternetExplorerDriver();                 } else if (browserName.equalsIgnoreCase("chromeheadless")) {                         System.setProperty("webdriver.chrome.driver", "..\\chromedriver.exe");                         ChromeOptions options = new ChromeOptions();                         options.addArguments("headless");                         options.addArguments("window-size=1366x768");                         driver = new ChromeDriver(options);                 }                  driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);                 driver.manage().window().maximize();                 driver.get(Constants.DEV\_URL);                 return driver; } |

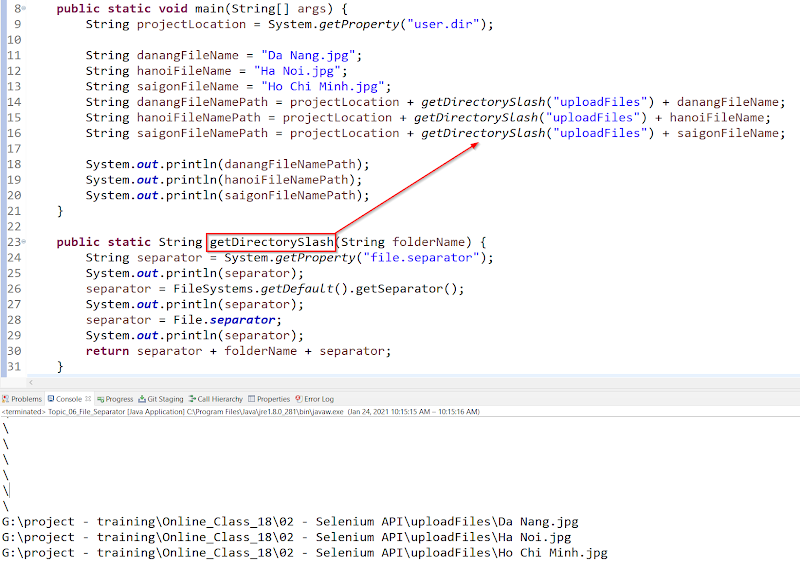
**Multiple OS**



* **Windows**
* Backslash (\)
* **MAC/ Linux (Unix)**
* Forwardslash (/)
* **System properties**
* String osName = System.getProperty("os.name");
* Window/ Mac/ Linux/ Unix/...
* String projectPath = System.getProperty("user.dir");
* G:\Automation Testing Online\Class 14\pom-wordpress-14-damdm
* /Automation Testing Online/Class 14/pom-wordpress-14-damdm
* **Check OS and replace**



* **File Path Separator (Work for MAC/ Linux/ Windows)**
* *For System.getProperty("file.separator"), we can override the value by System.setProperty() or command line -Dfile.separator*
* *Both File.separator and FileSystems.getDefault().getSeparator() will returns the same separator.*



* **Code -**[**Link**](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://gist.github.com/daominhdam/84dca32d2c1b78995568781ebda72dab%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332515000%26amp;usg%3DAOvVaw0GIJaTq4Nxd-T378PGYbS-&sa=D&source=editors&ust=1628668332566801&usg=AOvVaw2aMPBOdhSGUFFywxWHW00G)
* **References:**
* [https://mkyong.com/java/how-to-detect-os-in-java-systemgetpropertyosname/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://mkyong.com/java/how-to-detect-os-in-java-systemgetpropertyosname/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332515000%26amp;usg%3DAOvVaw1JQHC6V_JWVjYAHF-ZDCIG&sa=D&source=editors&ust=1628668332567019&usg=AOvVaw3qyA1-Dj5HPia0x2YXcpp4)
* [https://www.baeldung.com/java-path](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://www.baeldung.com/java-path%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332515000%26amp;usg%3DAOvVaw0XqsbJGraWWi5loUiRr0tJ&sa=D&source=editors&ust=1628668332567125&usg=AOvVaw2a-BGyOyPSVEkBFSiayCFg)
* [https://www.journaldev.com/848/java-file-path-absolute-canonical](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://www.journaldev.com/848/java-file-path-absolute-canonical%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332516000%26amp;usg%3DAOvVaw0Kg66UfdxpQgPjGwHEOqiU&sa=D&source=editors&ust=1628668332567220&usg=AOvVaw1Fwdm7RL4BK8jtA4cNbmuv)

**WebDriver Manager**

* **Mục đích**
* Các **browser nâng cấp version** đồng nghĩa với việc các **browser driver** phải được nâng cấp để đồng bộ
* ChromeDriver: [https://chromedriver.storage.googleapis.com/index.html](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://chromedriver.storage.googleapis.com/index.html%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332516000%26amp;usg%3DAOvVaw33SPGVwnbivb3oW91fKdP8&sa=D&source=editors&ust=1628668332567451&usg=AOvVaw13XuBy_jBVtkznKYFvWKSU)
* InternetExplorerDriver: [https://github.com/SeleniumHQ/selenium/wiki/InternetExplorerDriver](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://github.com/SeleniumHQ/selenium/wiki/InternetExplorerDriver%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332517000%26amp;usg%3DAOvVaw2OdyECL0KNGm2XnQV8ag-Q&sa=D&source=editors&ust=1628668332567581&usg=AOvVaw2t2oZZgLlC9JqflQQhxJHS)
* Tải cùng version vs Selenium Server: [http://selenium-release.storage.googleapis.com/index.html?path=2.53/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://selenium-release.storage.googleapis.com/index.html?path%253D2.53/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332517000%26amp;usg%3DAOvVaw3aBqzpSjBHUhmENandKTrA&sa=D&source=editors&ust=1628668332567693&usg=AOvVaw3tOjiSAF3G_ED-Ytz3ODVC)
* GeckoDriver: [https://github.com/mozilla/geckodriver/releases](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://github.com/mozilla/geckodriver/releases%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332517000%26amp;usg%3DAOvVaw0ZYAoc_KasHDXz4-nTsm6G&sa=D&source=editors&ust=1628668332567792&usg=AOvVaw2Zu1Yn6Ck_79toOqDOB0tC)
* OperaDriver:
* [https://github.com/operasoftware/operachromiumdriver/releases](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://github.com/operasoftware/operachromiumdriver/releases%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332518000%26amp;usg%3DAOvVaw17Rk0nUsZR4kHBZe8tL64D&sa=D&source=editors&ust=1628668332567911&usg=AOvVaw2d2VH3n34n7RF_DRJrdKgw)
* [http://npm.taobao.org/mirrors/operadriver/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://npm.taobao.org/mirrors/operadriver/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332518000%26amp;usg%3DAOvVaw2x16InKS5bZckvXj3Chc7E&sa=D&source=editors&ust=1628668332567998&usg=AOvVaw3m0CqLJ9MRsWjItCOvuoCw)
* SafariDriver: [https://automationfc.com/2018/08/25/java-webdriver-18-run-test-tren-trinh-duyet-safari/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://automationfc.com/2018/08/25/java-webdriver-18-run-test-tren-trinh-duyet-safari/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332518000%26amp;usg%3DAOvVaw1kXm6psJ2xy_Ic19QMwsEQ&sa=D&source=editors&ust=1628668332568100&usg=AOvVaw1rSJiUDjzZtrVP4tDrN6ep)
* EdgeDriver:
* [https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332519000%26amp;usg%3DAOvVaw3ps3xJ7_DcHaRLSeoCQuke&sa=D&source=editors&ust=1628668332568243&usg=AOvVaw3qB-bmGMpPKdalCnFrK1P3)
* **Mỗi lần update version phải xóa bản cũ và tải bản mới về thay thế gây mất thời gian**
* **Giải pháp**
* Sử dụng thư viện WebDriverManager để tự động tải về version của browser
* Download libraries: [Link](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://drive.google.com/drive/folders/1BlkF83oSiJZUmY9KqG8FhMz9XT3Szzcy%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332519000%26amp;usg%3DAOvVaw3V7pjKwDTWA6QgueGAjFgr&sa=D&source=editors&ust=1628668332568469&usg=AOvVaw0P9-BjIc1vgFck932dN9Q_)
* WebDriverManager github - [https://github.com/bonigarcia/webdrivermanager](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://github.com/bonigarcia/webdrivermanager%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332519000%26amp;usg%3DAOvVaw0hihWRwjy_QSXyMrZG2_wh&sa=D&source=editors&ust=1628668332568562&usg=AOvVaw0VeaCUqNJgAb4MamCeBjwL)
* Support cho các browser:
* Chrome/ Firefox/ IE/ Opera/ Edge
* **Chưa support cho Safari (ko có quyền truy cập vào để sửa các setting trong hệ thống của hệ điều hành MAC)**
* Thư viện cần sử dụng



* Dùng WebDriverManager version mới nhất mới chạy được cho Edge Chromium
* Setup

|  |
| --- |
| WebDriverManager.chromedriver().setup(); WebDriverManager.firefoxdriver().setup(); WebDriverManager.operadriver().setup(); WebDriverManager.phantomjs().setup(); WebDriverManager.edgedriver().setup(); WebDriverManager.iedriver().setup(); |

* Sử dụng 1 version cụ thể - nếu ko set version cụ thể - nó sẽ luôn tải version mới nhất

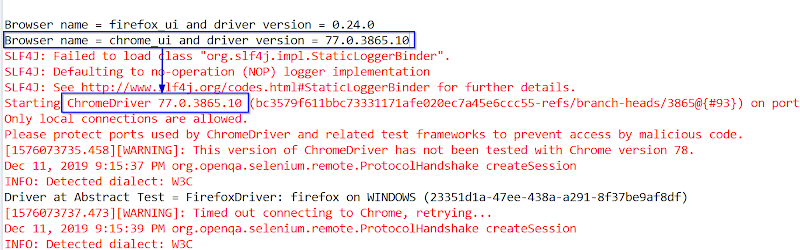


|  |
| --- |
| WebDriverManager.chromedriver().driverVersion("2.26").setup();         WebDriverManager.firefoxdriver().arch32().setup(); WebDriverManager.iedriver().arch32().setup(); |

* Update vào hàm openMultiBrowser trong AbstractTest

|  |
| --- |
| public WebDriver openMultiBrowser(String browserName) {                  if (browserName.equals("firefox")) {                         WebDriverManager.firefoxdriver().arch64().setup();                         driver = new FirefoxDriver();                 } else if (browserName.equals("chrome")) {                         WebDriverManager.chromedriver().setup();                         driver = new ChromeDriver();                 } else if (browserName.equals("chromeheadless")) {                         WebDriverManager.chromedriver().setup();                         ChromeOptions options = new ChromeOptions();                         options.addArguments("headless");                         options.addArguments("window-size=1920x1080");                         driver = new ChromeDriver(options);                 } else if (browserName.equals("internetexplorer")) {                         WebDriverManager.iedriver().arch32().setup();                 }                  driver.get(Constants.DEV\_URL);                 driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);                 driver.manage().window().maximize();                 return driver; } |

* Run test

****

**Reference**

* [https://automationfc.com/2019/05/27/tuong-thich-giua-cac-version/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://automationfc.com/2019/05/27/tuong-thich-giua-cac-version/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332523000%26amp;usg%3DAOvVaw2_BOPDxg6-9h_ee497qYfF&sa=D&source=editors&ust=1628668332569842&usg=AOvVaw0VRFwpKRSLsmhL8eRJ9pgD)
* [https://automationfc.com/2017/11/16/java-webdriver-13-run-test-voi-trinh-duyet-chrome-headless/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://automationfc.com/2017/11/16/java-webdriver-13-run-test-voi-trinh-duyet-chrome-headless/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332523000%26amp;usg%3DAOvVaw3IEcwaguXMOCVjtlKQgknI&sa=D&source=editors&ust=1628668332570002&usg=AOvVaw3qXi7vAfeW20UA7gfcZpui)
* [https://stackjava.com/design-pattern/factory-pattern.html](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://stackjava.com/design-pattern/factory-pattern.html%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332524000%26amp;usg%3DAOvVaw0KScEeGIh56stw1B9V6Rnb&sa=D&source=editors&ust=1628668332570197&usg=AOvVaw0wUmhDrWtlmpXIgEkiGj-h)
* [https://gpcoder.com/4352-huong-dan-java-design-pattern-factory-method/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://gpcoder.com/4352-huong-dan-java-design-pattern-factory-method/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332524000%26amp;usg%3DAOvVaw0qkE4i2fR4FN83aAxCybuM&sa=D&source=editors&ust=1628668332570327&usg=AOvVaw3KqLejND0J8ErQOkg0Q_ax)
* [https://gpcoder.com/4190-huong-dan-java-design-pattern-singleton/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://gpcoder.com/4190-huong-dan-java-design-pattern-singleton/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332524000%26amp;usg%3DAOvVaw1bGn58_mj8WU85_C_gsGWO&sa=D&source=editors&ust=1628668332570447&usg=AOvVaw2Cvcs8AJHp1WHdrkoXoNLM)
* [https://www.lambdatest.com/blog/parameterization-in-testng-dataprovider-and-testng-xml-examples/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://www.lambdatest.com/blog/parameterization-in-testng-dataprovider-and-testng-xml-examples/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332524000%26amp;usg%3DAOvVaw1JonxoxtrIg17r7OzUwMs0&sa=D&source=editors&ust=1628668332570556&usg=AOvVaw3EZQ5OWbWbAE1kVsM_tZLc)
* [http://thachleblog.com/loi-ich-cua-su-dung-enum-java/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://thachleblog.com/loi-ich-cua-su-dung-enum-java/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332525000%26amp;usg%3DAOvVaw2miWtTcZ_1XNToYWyZyUGR&sa=D&source=editors&ust=1628668332570656&usg=AOvVaw2QPY-SLg932pHpYk5GxbFJ)
* [https://o7planning.org/vi/10199/huong-dan-java-enum](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://o7planning.org/vi/10199/huong-dan-java-enum%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332525000%26amp;usg%3DAOvVaw35ui3I9APce1FmSWQeE96J&sa=D&source=editors&ust=1628668332570760&usg=AOvVaw01Zd51D_tHOzsa50-vnz1w)
* [https://gpcoder.com/2473-enum-trong-java/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://gpcoder.com/2473-enum-trong-java/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332525000%26amp;usg%3DAOvVaw2pTPeBzFi0_3rO8NCdzplw&sa=D&source=editors&ust=1628668332570869&usg=AOvVaw03iWH7W58hyYkvmlt0mYHc)
* [http://thachleblog.com/loi-ich-cua-su-dung-enum-java/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://thachleblog.com/loi-ich-cua-su-dung-enum-java/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332525000%26amp;usg%3DAOvVaw2miWtTcZ_1XNToYWyZyUGR&sa=D&source=editors&ust=1628668332570963&usg=AOvVaw2KOrmurzGShw1Wap8QS1b_)
* [http://makeseleniumeasy.com/2020/04/27/usage-of-enum-in-selenium-webdriver-scripts/](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://makeseleniumeasy.com/2020/04/27/usage-of-enum-in-selenium-webdriver-scripts/%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332526000%26amp;usg%3DAOvVaw1vzMFDW5dlzzpBGw1WLWrd&sa=D&source=editors&ust=1628668332571059&usg=AOvVaw0_HcEdyNzotIr6pa7aOhA6)
* [https://www.youtube.com/watch?v=7PZVywJSy9g](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttps://www.youtube.com/watch?v%253D7PZVywJSy9g%26amp;sa%3DD%26amp;source%3Deditors%26amp;ust%3D1628668332526000%26amp;usg%3DAOvVaw2hRR2QQyW3FTu8EhsFzd4b&sa=D&source=editors&ust=1628668332571157&usg=AOvVaw2fWDl22yc092kk_D6HFeDY)