**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO KẾT THÚC HỌC PHẦN**

***Đề Tài:* Xây Dựng Selenium Framework Theo Mô Hình Page Object Model (POM)**

**Giảng viên hướng dẫn : Ths. Tạ Quốc Ý**

**Sinh viên thực hiện : Bùi Hoàng Long**

**Lớp : ST19A1A**

**MSSV : 46811**

Mục lục

[I. TỔNG QUAN VỀ SELENIUM – PAGE OBJECT MODEL 2](#_Toc102416161)

[1.Tổng quan về kiểm thử phần mềm: 2](#_Toc102416162)

[2. Selenium WebDriver 2](#_Toc102416163)

[3. Page Object Model (POM) 8](#_Toc102416164)

## TỔNG QUAN VỀ SELENIUM – PAGE OBJECT MODEL

### 1.Tổng quan về kiểm thử phần mềm:

**Kiểm thử phần mềm** là quá trình thực thi 1 chương trình với mục đích tìm ra lỗi. **Kiểm thử phần mềm** đảm bảo sản phẩm **phần mềm** đáp ứng chính xác, đầy đủ và đúng theo yêu cầu của khách hàng, yêu cầu của sản phẩm đề đã đặt ra 2. Selenium WebDriver

**Selenium Webdriver là** một thành phần quan trọng của bộ công cụ **Selenium**. Đây **là** một bộ công cụ kiểm thử (test) tự động nổi tiếng. Nó được ưu chuộng vì có mã nguồn mở, có khả năng hỗ trợ nhiều ngôn ngữ lập trình. Bộ công cụ này gồm có bốn thành phần **là**: **Selenium** IDE, **Selenium** RC, **Selenium** Grid và **Selenium Webdriver.**

### Tại sao phải kiểm thử phần mềm? Mục tiêu kiểm thử là gì?

* Lý do cần kiểm thử phần mềm:

• Kiểm thử phần mềm là yếu tố quyết định của SQA và khâu điển hình của rà soát đặc tả thiết kế và lập mã.

• Muốn nhìn thấy phần mềm như là một phần tử của hệ thống hoạt động (xem sản phầm) • Hạn chế chi phí phải trả cho các thất bại do lỗi gây ra sau này (hiệu quả)

• Có kế hoạch tốt nâng cao chất lượng cho suốt quá trình phát triển (giải pháp)

* Tầm quan trọng của kiểm thử phần mềm:

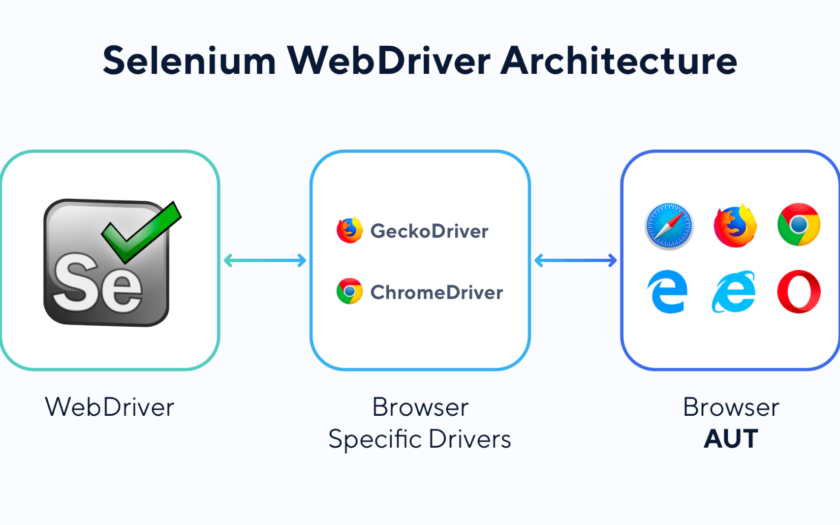
• Chi phí của kiểm thử chiếm: 40% tổng công sức phát triển ≥ 30% tổng thời gian phát triển

• Với phần mềm ảnh hưởng tới sinh mạng chi phí có thể gấp từ 3 đến 5 lần tổng chi phí khác cộng lại.

* Mục tiêu trước mắt: Cố gắng tạo ra các ca kiểm thử để chỉ ra lỗi của phần mềm với chi phí(thời gian, chi phí) thấp nhất. Một ca kiểm thử thắng lợi là phải làm lộ ra khiếm khuyết, đồng thời mang lại các lợi ích phụ.
* Mục tiêu cuối cùng: có một chương trình tốt, chi phí ít

#### 2.1. Selenium webDriver là gì?

Selenium WebDriver là công cụ kiểm thử tự động các ứng dụng Web

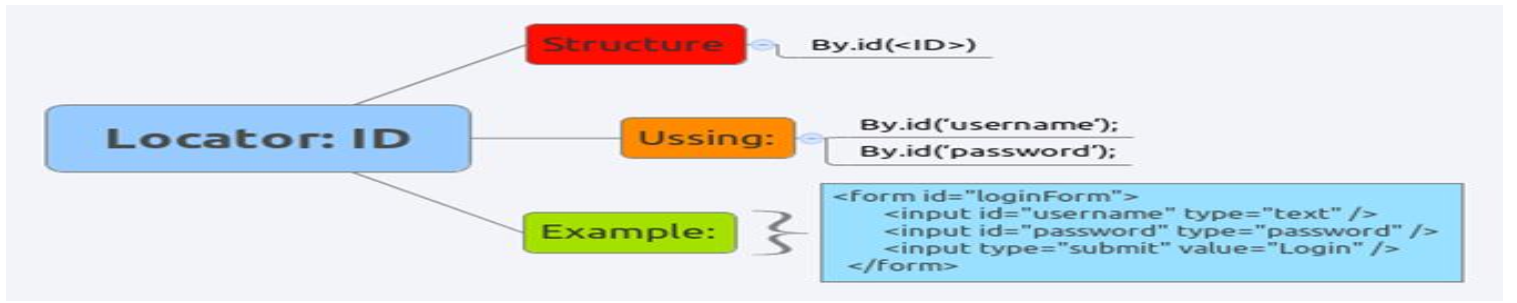


#### 2.2. Tổng quan về đối tượng UI (Locators)

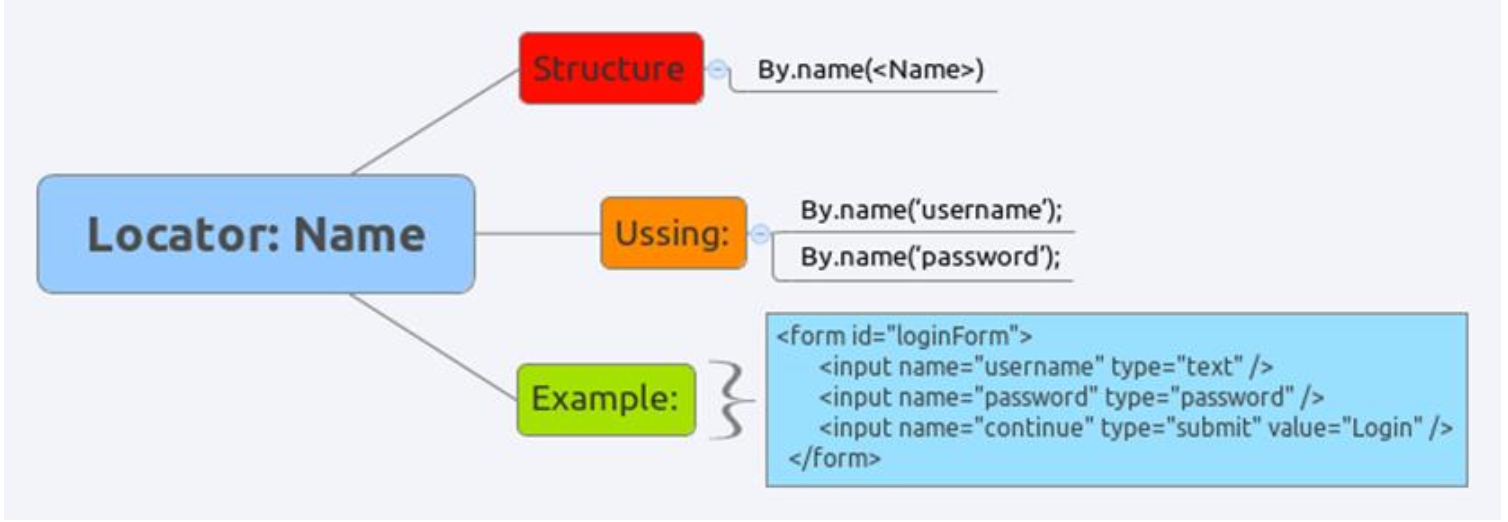
● Trong selenium, các phần tử trên web (WebElement) có vai trò rất quan trọng.  
Selenium hỗ trợ người dùng 7 cách để xác định các phần tử web này (Locator)



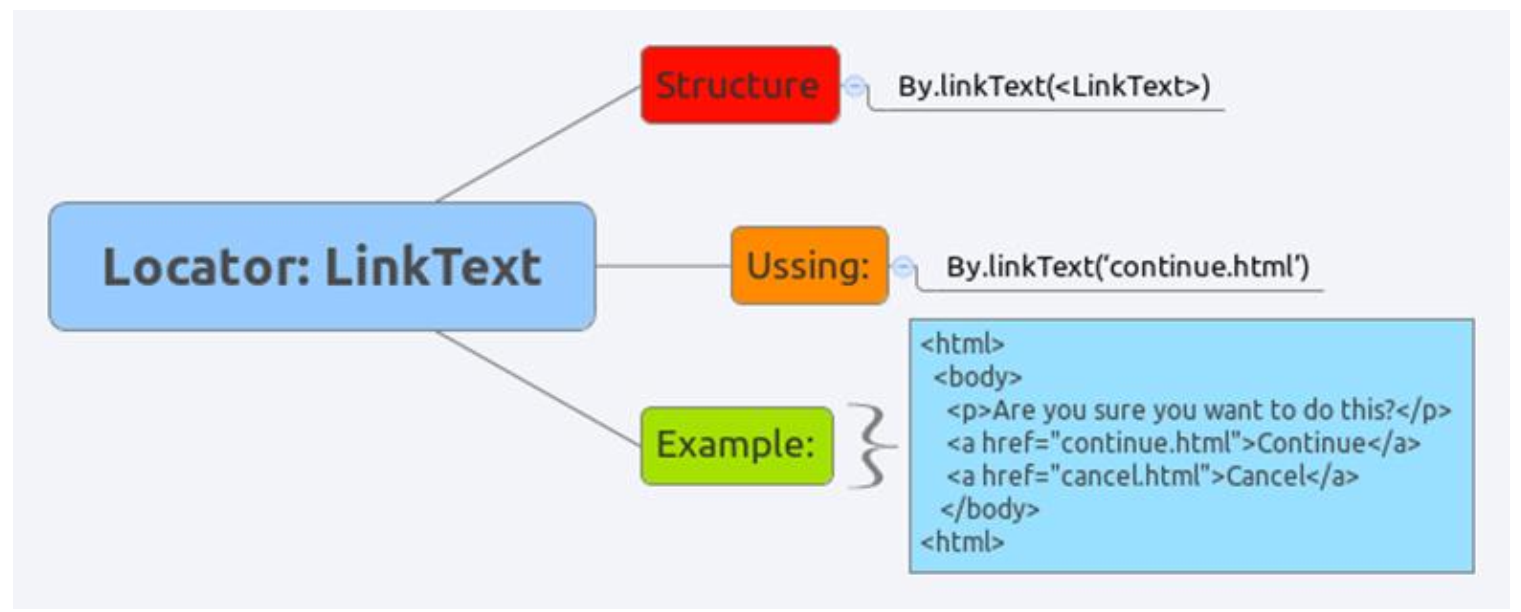
#### 2.3. Xác định phần tử Web theo ID



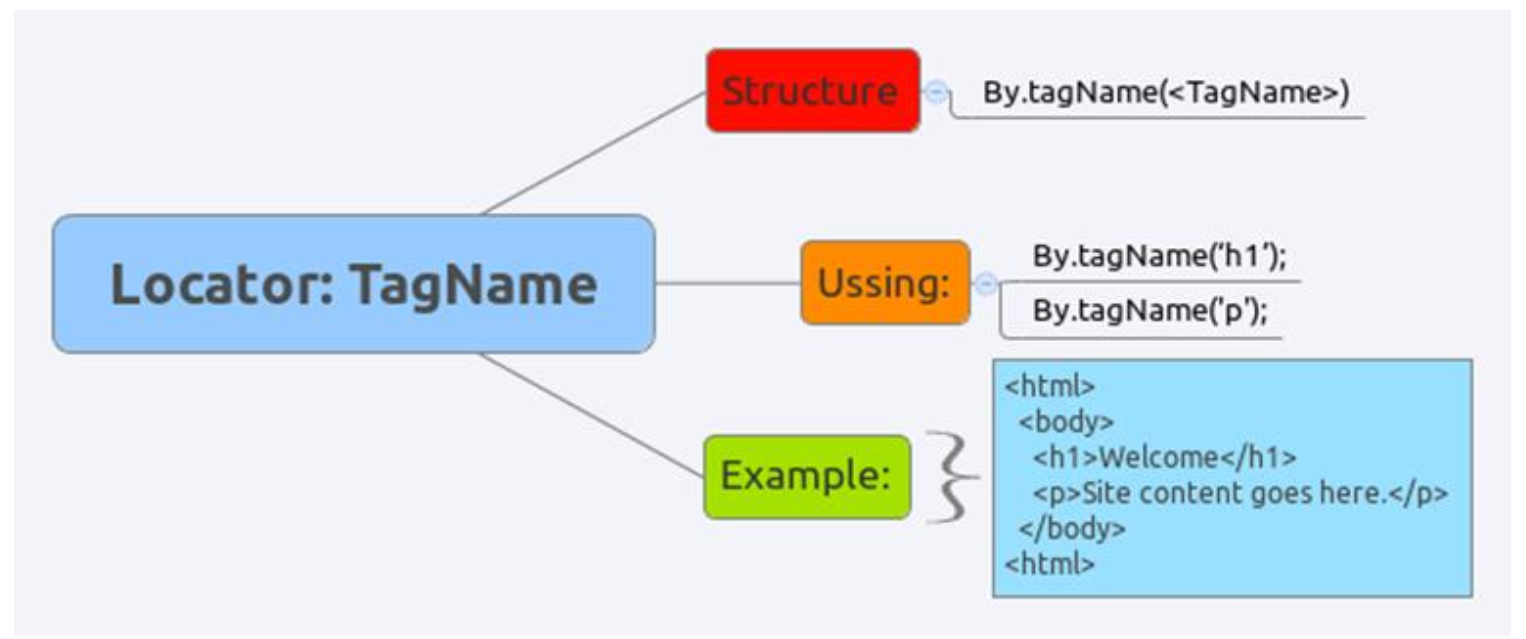
#### 2.4. Xác định phần tử Web theo Name



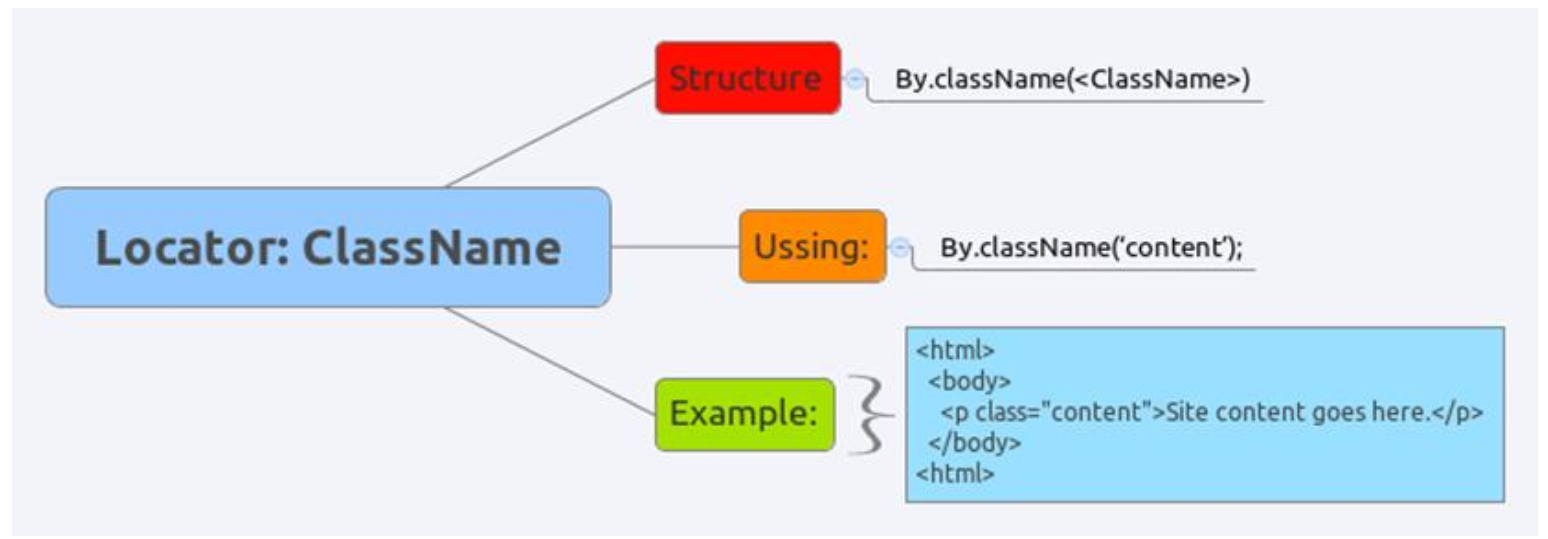
#### 2.5. Xác định phần tử Web theo LinkText



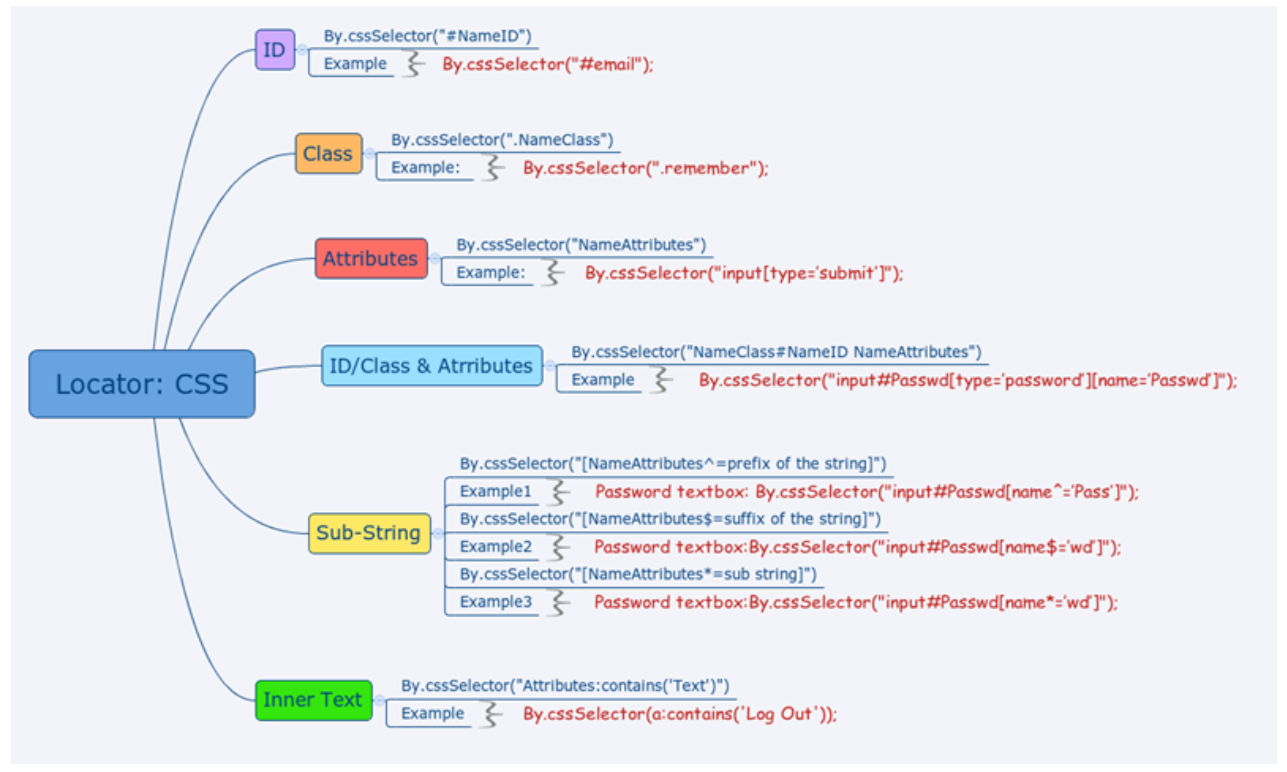
#### 2.6. Xác định phần tử Web theo TagName



#### 2.7. Xác định phần tử Web theo ClassName



#### 2.8. Xác định phần tử Web theo CSS

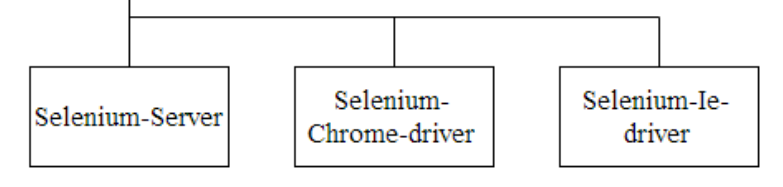


#### 2.9. Xác định phần tử Web theo Xpath

● Xpath là việc sử dụng những biểu thức để duyệt các node trong XML/HTML.  
● XPath là phương thức được sử dụng nhiều nhất  
● Dưới đây là ví dụ cấu trúc HTML của một trang web và cách dùng Xpath để duyệt các  
phần tử trên trang web đó  
● Absolute XPath (XPath tuyệt đối)  
○ Bắt đầu với node gốc hoặc dấu ‘/’  
○ Ưu điểm: tìm element nhanh  
○ Nhược điểm: nếu có sự thay đổi nào giữa các node à xpath sẽ sai  
○ Ví dụ: html/body/div/form/select/option  
○ Khi có thêm một tag giữa body và divà xpath sẽ không tìm được element  
html/body/table/div/form/select/option  
● Relative XPath (XPath tương đối)  
○ Có thể bắt đầu ở bất kỳ node nào với dấu ‘//’  
○ Ưu điểm: xpath ngắn  
○ Nhược điểm: tìm element lâu hơn xpath tuyệt đối  
○ Ví dụ: //select/option  
● Kết hợp giữa xpath tuyệt đối, tương đối  
○ Có thể kết hợp giữa xpath tuyệt đối và tương đối  
○ Ví dụ: html//table/tbody/tr/th

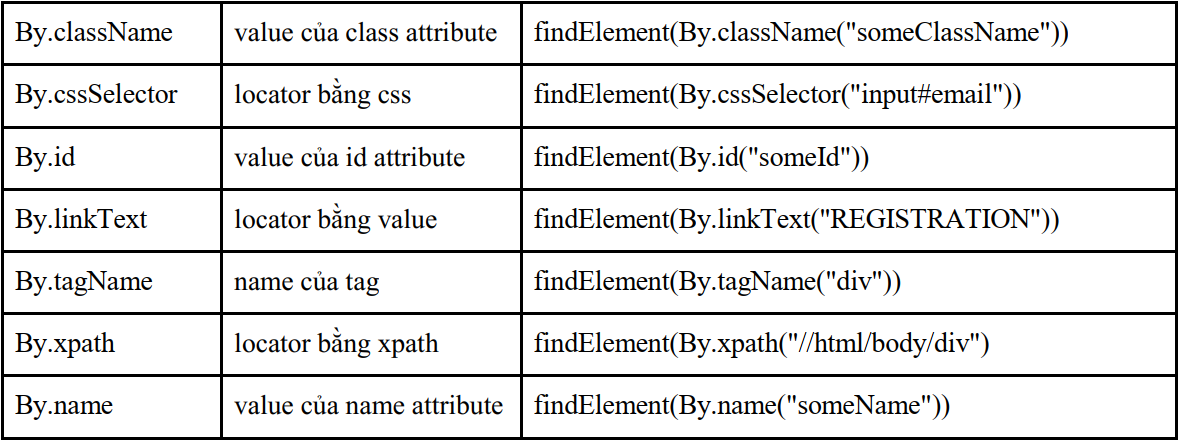
#### 2.10. Các thư viện để chạy Selenium WebDiver

● Danh sách các thư viện Selenium WebDriver cần cài đặt

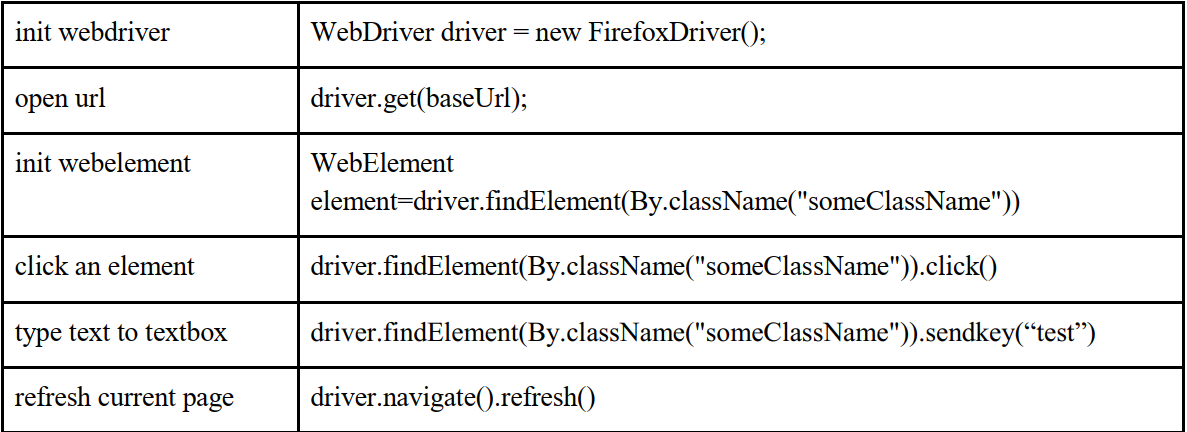


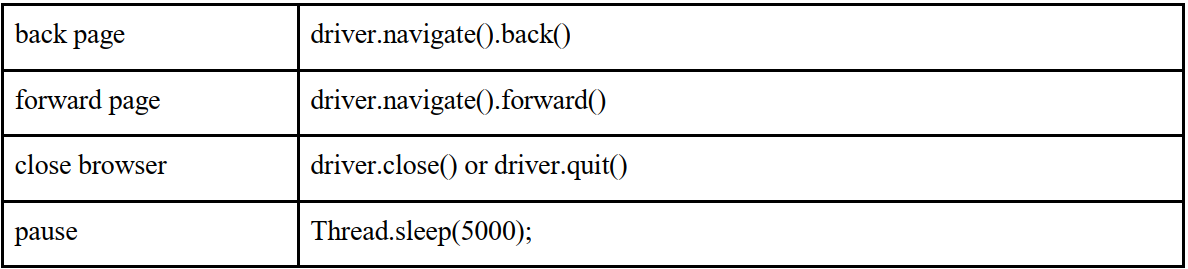
#### 2.11. Các hàm xử lý chung trong Selenium WebDriver

● Locate element sử dụng WebDriver



● Các hàm hay sử dụng





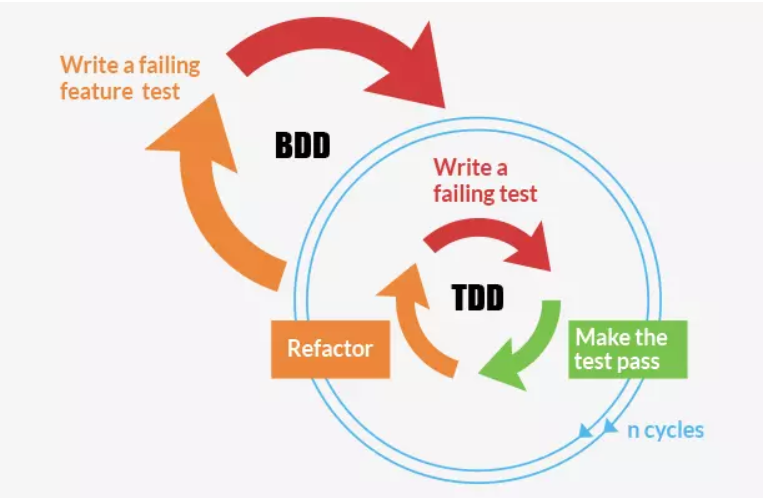
### 3. Behavior Driven Development (BDD)

### 3.1. Behavior Driven Development là gì?

Behavior Driven Development (BDD) là một phần mở rộng của TDD. Giống như ở TDD, trong BDD chúng ta cũng viết các bài test trước và sau đó mới code ứng dụng để vượt qua những bài test đó. Sự khác biệt chính của BDD so với TDD chính là:

* Các bài test hay các test case sẽ được viết bằng ngữ pháp tiếng Anh và mô tả rất đơn giản
* Các bài test được xem như là hành vi của ứng dụng dựa trên những nhu cầu thực tế của người dùng với ứng dụng đó
* Sử dụng các ví dụ để làm rõ các yêu cầu của ứng dụng

Sự khác biệt này dẫn đến sự cần thiết phải có một ngôn ngữ có thể định nghĩa, trong một định dạng dễ hiểu để chúng ta có thể áp dụng BDD một cách hiệu quả nhất . Và Gherkin là ngôn ngữ sẽ làm việc này



# 3.2. Features of BDD?

1. Chuyển từ suy nghĩ về "những bài test" sang suy nghĩ về "hành vi của ứng dụng"
2. Sự hợp tác giữa các bên liên quan Business stakeholders, Business Analysts, QA Team and developers đều phải tham gia vào quá trình phát triển
3. Ngôn ngữ phổ biến, dễ dàng để mô tả
4. Hướng đến những giá trị kinh doanh
5. Mở rộng Test Driven Development (TDD) bằng cách sử dụng ngôn ngữ tự nhiên mà các bên liên quan không cần biết kỹ thuật vẫn có thể hiểu được nhau
6. Các framework BDD như Cucumber hoặc JBehave là những người đi tiên phong, đóng vai trò là cầu nối giữa các ngôn ngữ Business và Technical

#### 3.3. Độ phổ biến của BDD

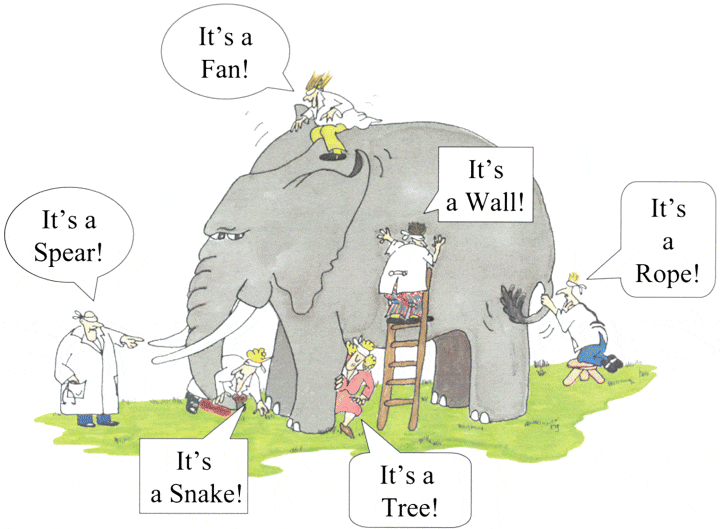
BDD ngày này rất phổ biến và có thể được sử dụng cho Unit test case và cho cả các UI test case . Các công cụ như RSpec (cho Ruby) hoặc trong .NET, người ta thường dùng MSpec hoặc SpecUnit được sử dụng phổ biến cho Unit test theo phương pháp BDD. Ngoài ra, bạn có thể viết specs dựa trên kiểu BDD về tương tác giữa các phần tử UI với nhau . Giả sử bạn đang xây dựng một ứng dụng web, có thể bạn sẽ sử dụng thư viện tự động hóa trình duyệt như WatiR / WatiN hoặc Selenium và viết kịch bản bằng cách sử dụng một trong các framework vừa được đề cập ở trên, hoặc công cụ kiểu Given / When / Then như Cucumber ( cho Ruby) hoặc SpecFlow (cho .NET)

### 3.4. Tại sao phải xử dụng BDD:

Giả sử có một yêu cầu từ khách hàng đối với trang web Thương mại điện tử để tăng doanh số bán sản phẩm với việc triển khai một số tính năng mới trên trang web. Thách thức duy nhất của nhóm phát triển là chuyển đổi ý tưởng của khách hàng thành thứ gì đó thực sự mang lại lợi ích cho khách hàng.

Ý tưởng ban đầu là tuyệt vời. Nhưng thách thức duy nhất ở đây là người đang phát triển ý tưởng không phải là người có ý tưởng này. Nếu người có ý tưởng là một nhà phát triển phần mềm tài năng, thì chúng ta có thể gặp may: ý tưởng có thể được chuyển thành phần mềm hoạt động mà không cần phải giải thích với bất kỳ ai khác. Bây giờ ý tưởng đó cần được truyền đạt và phải đi từ Business Owner (Khách hàng) đến các nhóm phát triển hoặc nhiều người khác.

Hầu hết các dự án phần mềm đều làm việc theo team và có thêm một số người liên quan làm việc cùng nhau, vì vậy giao tiếp chất lượng tốt với nhau là rất quan trọng đối với thành công của một dự án phần mềm. Như bạn biết, giao tiếp tốt không chỉ là mô tả hùng hồn ý tưởng của bạn cho người khác; bạn cũng cần thu thập các phản hồi để đảm bảo bạn đã hiểu đúng và team của bạn cũng đang hiểu đúng những gì bạn đang trình bày. Đây là lý do tại sao Agile development team đã học cách làm việc theo từng bước nhỏ,bằng cách thường xuyên gửi những chức năng ngay khi nó vừa được hoàn thành cho khách hàng để nhận phản hồi thường xuyên từ khách hàng và cũng để trả lời cho câu hỏi "Đây có phải là thứ khách hàng muốn ?" một cách nhanh nhất.



Với sự giúp đỡ của ngôn ngữ Gherkin, Cucumber có thể giúp cả team tạo ra một ngôn ngữ chung mà không cần phân biệt người đó có cần biết về kỹ thuật hay không. Các bài test được viết bằng Cucumber tương tác trực tiếp với code, nhưng các bài test này được viết bằng ngôn ngữ khá dễ hiểu thường sẽ được bởi các bên liên quan đến kinh doanh. Cucumber test giúp chúng ta loại bỏ nhiều hiểu lầm trong specs để phòng ngừa khả năng tạo ra những đoạn code mơ hồ và không chắc chắn.

## 2. Dự án Kiểm thử kết thúc môn (cá nhân):

Công cụ: Pycham 

Thư viện: Selenium 

Ngôn ngữ: python 

Trình duyệt: Edge 

### 2.1. Cấu trúc dự án:

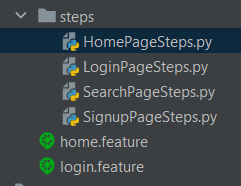
### 

Giải thích Cấu trúc dự án:

\_Configuration:

Nơi chứa dữ liệu để test chức năng hay các địa chỉ URL, địa chỉ file Excel và 1 số thứ

\_Features:



File có đuôi .feature là file định nghĩa các bước, trình tự để test chức năng cho trang web

Forder Steps chức các file python là nơi những đoạn code test chạy tương ứng với trình tự các bước đã được định nghĩa ở file .feature (được viết bằng ngôn ngữ Python)

\_Forder Logs:

Là nơi logs ra trình tự chạy các bước trong quá trình test, mục đích là để người dùng dễ dàng theo dõi được quá trình test từ đó dễ dàng tìm lỗi mà fix chúng

\_Pages :

Chứa các link Xpath của các Element và các event như click, sendkeys, …

\_Reports:

Là nơi chưa các file JSON và file hình ảnh sau khi chạy chương trình test, có thể dùng để check lỗi và là dữ liệu cho web test

\_Utillities:

Là nơi xử lý những chức năng tiện ích như lấy dữ liệu test từ forder Configuration, logs ra trình tự xử lý chức năng, giao tiếp với file excel

### 2.2. Luồng xử lý:

Trong dự án này có 4 test module chúng ta cần quan tâm.

\_trang Home Page:

+ Test title match Home Page

+ Test search feature Home page

\_ trang Login Page:

+ Login feature

+ Sinup feature

#### 2.2.1. Test Title Match Home Page

#### Bước 1: mở trình duyệt:

\_lấy địa chỉ URL từ forder configuration

\_dùng webdriver.Edge để chạy địa chỉ URL trên trình duyệt Edge

Bước 2: kiểm tra title page

\_lấy ra title bằng context.driver.title

Bước 3: Đóng trình duyệt

#### 2.2.2. Search page Home Page

Bước 1: mở trình duyệt

Bước 2: lấy dữ liệu từ file DataTest

Bước 3: dùng vòng lặp để lây trường dữ liệu tương ứng và điền dữ liệu vào ô input search

kiểm tra class xem có trùng với class name nếu trùng thì test passed, ngược lại thì test failed

Bước 4: ghi lại dữ liệu vào file DataTest

Bước 5: đóng trình duyệt

#### 2.2.3. Login feature Login Page

Bước 1: mở trình duyệt

Bước 2: lấy dữ liệu từ file DataTest

Bước 3: dùng vòng lặp để lây trường dữ liệu tương ứng và điền dữ liệu vào ô email và password

kiểm tra xem có đăng nhập thành công không: nếu đăng nhập thành công thì set lại data test passed, ngược lại thì set data test failed

Bước 4: ghi lại dữ liệu vào file DataTest

Bước 5: đóng trình duyệt

#### 2.2.4. Signup feature Login Page

Bước 1: mở trình duyệt

Bước 2: lấy dữ liệu từ forder configuration

\_ghi dữ liệu vào input user name, email, password

Bước 3: nhấn đăng ký

Bước 4: đóng trình duyệt