Test Automation & Advanced Selenium

Lesson 8: Working with Page Object Model (POM)

Capgemini

Lesson Objectives

- Why Page Object Model (POM)?
- What is Page Object Model (POM)?
- Page Object Model (POM) Architecture
- Advantages of Page Object Model (POM)
- Overview of Selenium Design Patterns

Why Page Object Model (POM)?

Problem:

When you are writing functional tests using Selenium, a major part of your code will consist of interactions with the web interface you are testing through the WebDriver API. Most of the times this code does the job of fetching the elements, verify some state of the elements through various assertions and move on to fetch the next element.

Consider the following simple example: Figure 1.1

```
List<WebElement> countryCodes =
driver.findElements(By.id("countryCodes"));
for (WebElement countryCode : countryCodes)
{
     if (countryCode.getText().equals("1043"))
     {
        countryCode.click();
        break;
     }
}
```

Why Page Object Model (POM)? (Cont.)

Following are some of the challenges posed by traditional automation methods as given in Figure 1.1

- Difficult to maintain test scripts: The test script maintenance becomes difficult with time as the test suit grows. If 10 different scripts are using the same page element, with any change in that element, you need to change all 10 scripts. This is time consuming and error prone.
- Test cases are difficult to read: Even with a simple test as this, readability is very poor. There is a lot of WebDriver code, that obscures the purpose of the test, making it slow and difficult to digest.
- Duplication of selectors both inside and across tests: Duplicity of locators makes the test code inefficient.
- Changes in the UI breaks multiple tests often in several places: With any interface, and web interface, it is common that both minor and major changes to the UI is implemented frequently. This could be a new design, restructuring of fields and buttons, and this will likely impact your tests. So, your test fails, and you need to update your selectors.

Why Page Object Model (POM)? (Cont.)

Solution:

- So, instead of having each test fetch elements directly and being fragile towards UI changes, the Page Object Model introduces what is basically a decoupling layer. You create an object that represents the UI you want to test, which could be a whole page or a significant part of it. The responsibility of this object is to wrap HTML elements and encapsulate interactions with the UI, meaning that this is where all calls to WebDriver will go. This is where most web elements are. And this is the only place you need to modify when the UI changes. The Page Object Model (POM) is a new test automation buzz word. It was brought to address the multiple bottlenecks in web application testing like given above.

What is Page Object Model (POM)?

- A Page Object Model is a design pattern that can be implemented using selenium WebDriver
- A Design pattern is a generic solution to a common software design/architecture problem
- Implementation of these design patterns leads to inclusion of best practices and best solution, evolved over the time by others while working with similar problems
- Page Object Model is a design pattern to create object repository for web UI elements
- It essentially models the pages/screen of the application as objects called Page Objects, all the functions that can be performed in the specific page are encapsulated in the page object of that screen
- In this way any change made in the UI will only affect that screens page object class thus abstracting the changes from the test classes

Page Object Model (POM) Architecture

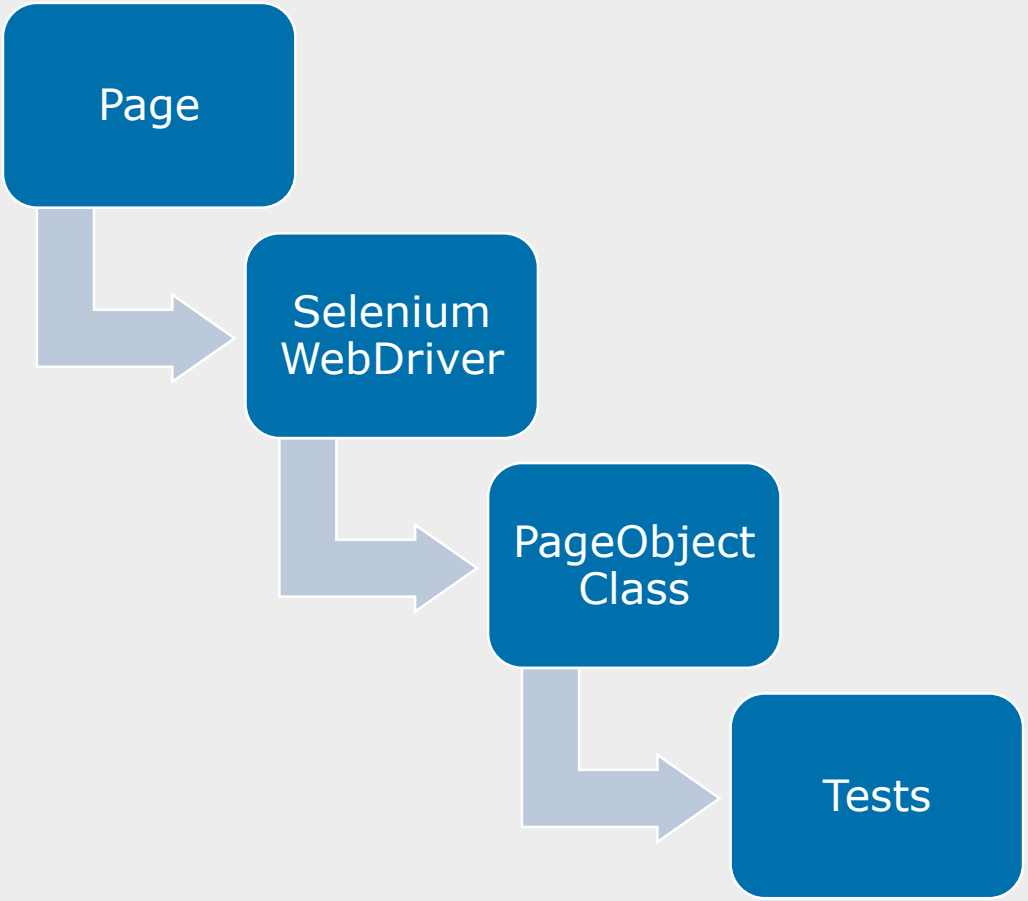

Figure 1.2

Demo – Implementing Page Object Model (POM)

- Simple POM Implementation

Advantages of Page Object Model (POM)

- Increases Code Reusability
- Improves Code Maintainability
- Independent Object Repository
- Readability
- Supports Selenium Frameworks

Summary

In this lesson you have learnt:

- Challenges of traditional automation testing methods
- Approach/Solution to resolve challenges posed by traditional automation testing methods
- Need for Page Object Model (POM)
- Introduction to Page Object Model (POM)
- Overview of Selenium Design Patterns

Review Question

Question 1:

- _____ is an optimized version of the Page Object Model Design pattern.

Question 2:
Page Object model is _____

- Design Pattern
- Unit level Framework
- Pattern Matching

Question 3:
What are the advantages of Page Object Model?

- Increase Code Reusability
- Improved code Maintainability
- Independent Object Repository
- All of the above