



УДК 004.056.55

ПАТЕРН ПРОЕКТУВАННЯ PAGE ОБ'ЄКТ ТА ЙОГО ВИКОРИСТАННЯ В АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ

Студ. – О.Ю.Свирид, гр. МгІТ-2-18
Науковий керівник доц. Т.І. Демківська
Київський національний університет технологій та дизайну

Мета і завдання Головною метою даного дослідження є використання патерну проектування Page Object в створенні фреймворку автоматизованого тестування, в що, також, ввійде розробка макету сайту для проведення демонстрації роботи авто-тестів. Фреймворк автоматизованого тестування має бути виконаний за правилами патерну Page Object. Кожна окрема сторінка макету сайту має бути представлена у вигляді окремого класу, в якій будуть зберігатися елементи цієї сторінки та методи які працюватимуть з ними. Завданням роботи є демонстрація того, як патерн Page Object спрощує написання коду, у цьому випадку автоматизованих тестів.

Об'єкт та предмет дослідження Об'єктом дослідження є засоби та правила патерну Page Object в розробці фреймворку автотестування. Предметом дослідження являються фреймворки та засоби їх реалізації, веб-технології для створення макету сайту.

Результати дослідження Page Object – це патерн дизайну, який є досить популярним в автоматизації тестування завдяки покращенню підтримки авто-тестів(пошук, редагування, написання) та відчутне зменшення кількості дублікації коду.

Головним правилом цього патерну є зберігання тестів та елементів веб-сторінки окремо.

Зазвичай, Page Object виступає у якості об'єктно-орієнтовного класу, що допомагає взаємодіяти з певною HTML сторінкою. Завдяки цьому класу, автотести можуть звертатись до його методів, що працюють з певними елементами цієї сторінки.

Основними складовими класів Page Object є веб-двайвер, локатори, методи, змінні.

- Веб-Драйвер – це портативний фреймворк для тестування веб-застосунків. З його допомогою можна знаходити та взаємодіяти з елементами веб-сторінки.

- Локатор – це унікально ідентифікований елемент веб-сторінки, такі як звичайна кнопка, зображення, поле для вводу, тд.

- Метод – це колекція тверджень, що зібрані разом заради виконання певною операції, наприклад, знайти та натиснути кнопку на веб-сторінці, або ввести текст в поле пошуку, тд.

- Змінна – це контейнер, що тримає певне значення. Змінні можуть приймати значення елементів веб-сторінки.

- Збільшення покриття автоматизованими тестами може призвести до структури фреймворку проекту який важко піддається підтримці, якщо локатори розміщені неправильним чином. Головною причиною цьому може бути дублікація коду, а також, дублікація використання самих локаторів.

Для прикладу можна привести сторінку інтернет магазину.

Кожен подібний сайт може мати хедер з кнопкою каталогу товарів, що відображається на кожній сторінці сайту. Щоб уникнути дублікації коду і не створювати локатор цієї кнопки у класі кожної окремої сторінки, Page Object патерн пропонує створити окремий клас хедеру цього сайту. В самому класі хедеру можна буде створювати локатори хедеру сайту, що дасть змогу звертатися до цього класу та використовувати елементи хедеру, а не створювати нові у класах кожної сторінки (рис.1).

Схема патерну для попереднього прикладу:

PAGE OBJECT FOR HEADERS

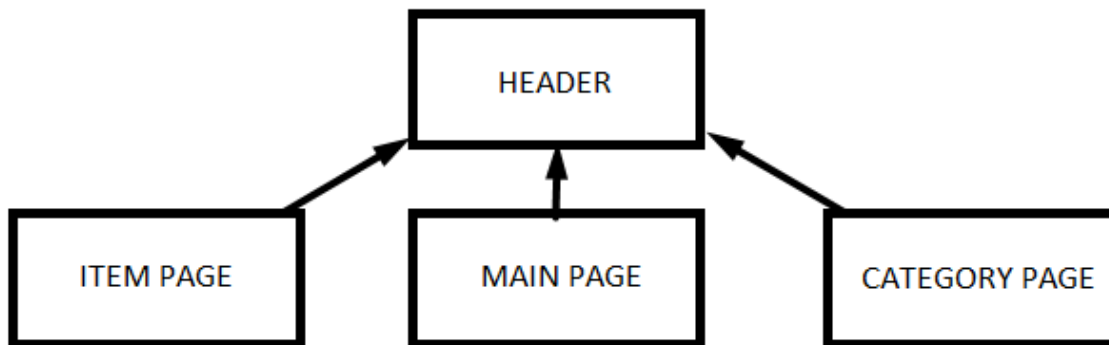


Рис. 1

Патерн Page Object найкраще розкриває свою потрібність для веб-додатків, що складаються з великої кількості веб-сторінок.

Для того щоб розпочати писати свій тестовий фреймворк на базі Page Object'у, спочатку, необхідно зробити аналіз додатку, який буде тестуватися. Потрібно розділити додаток на області, які є подібними одна до одної та створити для кожної області власний клас. Зазвичай, кожен клас виступає окремою веб-сторінкою, але це може бути не завжди так. Як було сказано у попередньому прикладі, Page Object'ом може бути хедер, футер, бокове меню, які теоретично можуть знаходитись на кожній сторінці додатку. Також, у ролі Page Object класу може бути якийсь окремих елемент веб-додатку який містить в собі певну кількість інших елементів з якими будуть працювати тести. Саме сценарії тестів підказують, які області треба взяти до уваги та розділити їх на класи Page Object'и, тому аналіз тестових сценаріїв є важливим елементом у слідуванні цьому патерну.

З вищеперерахованого випливає, що у разі зміні функціоналу, або зовнішнього вигляду веб-сторінки, самі тести не потребуватимуть внесення змін, тільки клас Page Object цієї зміненої сторінки.

Висновки. Проведено дослідження Page Object патерну. Завдяки відокремленню реалізації тестів та елементів веб-сторінки ми отримуємо можливість легкої підтримки фреймворку, тести стають короткими та читабельними, методи та елементи кожної сторінки можна використовувати в багатьох тестах та, у випадку зміни інтерфейсу веб-сторінки, достатньо змінити Page Object, не змінюючи самих тестів. Створено макет сайту та фреймворк автоматизації тестів цього макету за патерном Page Object.

Ключові слова: Page Object, патерн, фреймворк, автотести, веб-сторінка.

ЛІТЕРАТУРА

1. Elfriede Dustin; et al. (2009). Implementing Automated Software Testing. Addison-Wesley Professional; 1 edition (March 14, 2009). 368 pages
2. JavaPoint [Інтернет ресурс]: <https://www.javatpoint.com>
3. AutomatedTesting.info [Інтернет ресурс]: <https://automated-testing.info>
4. Maven [Інтернет ресурс]: <https://maven.apache.org/>
5. Selenium [Інтернет ресурс]: <https://www.seleniumhq.org/>