



COLLECTION OF SCIENTIFIC PAPERS



ISSUE
№45

3RD INTERNATIONAL SCIENTIFIC
AND PRACTICAL CONFERENCE

**INNOVATIVE
APPROACHES
IN MODERN SCIENCE
AND TECHNOLOGY**

NOVEMBER 12-14, 2025
LISBON, PORTUGAL





ISU

INTERNATIONAL SCIENTIFIC UNITY

3rd International Scientific and Practical Conference
**«Innovative Approaches in Modern Science
and Technology»**

Collection of Scientific Papers

November 12-14, 2025
Lisbon, Portugal

UDC 001(08)

Innovative Approaches in Modern Science and Technology: Collection of Scientific Papers with Proceedings of the 3rd International Scientific and Practical Conference. International Scientific Unity. November 12-14, 2025. Lisbon, Portugal. 643 p.

ISBN 979-8-89704-983-7 (series)
DOI 10.70286/ISU-12.11.2025

The conference is included in the Academic Research Index ReserchBib International catalog of scientific conferences.

The collection of scientific papers presents the materials of the participants of the 3rd International Scientific and Practical Conference "Innovative Approaches in Modern Science and Technology" (November 12-14, 2025. Lisbon, Portugal).

The materials of the collection are presented in the author's edition and printed in the original language. The authors of the published materials bear full responsibility for the authenticity of the given facts, proper names, geographical names, quotations, economic and statistical data, industry terminology, and other information.

The materials of the conference are publicly available under the terms of the CC BY-NC 4.0 International license.

ISBN 979-8-89704-983-7 (series)



© Participants of the conference, 2025
© Collection of Scientific Papers "International Scientific Unity", 2025
Official site: <https://isu-conference.com/>

Romaniuk O., Kostenko A. THE ROLE OF DIGITAL LITERACY FOR THE POPULATION OF UKRAINE.....	242
Шворак Д., Корень В. СУЧАСНІ ТЕХНОЛОГІЇ ТА АРХІТЕКТУРНІ ПІДХОДИ ДО РОЗРОБКИ МОБІЛЬНИХ ЗАСТОСУНКІВ НА ПЛАТФОРМІ ANDROID.....	246
Solodovnyk G., Chuieva A. CHOOSING AN INFORMATION PROTECTION SYSTEM UNDER RISK CONDITIONS.....	248
Флегантов Л., Даценко Н. АРХІТЕКТУРНІ ПАТЕРНИ У ПОБУДОВІ ФРЕЙМВОРКІВ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ ВЕБДОДАТКІВ.....	253
Лошик О., Корень В. ВИКОРИСТАННЯ РЕГРЕСІЙНОГО АНАЛІЗУ ДЛЯ ПРОГНОЗУВАННЯ ПОПИТУ У ВЕБОРІЄНТОВАНИХ АНАЛІТИЧНИХ СИСТЕМАХ.....	260
Деркач Т., Соколянський Н. КОМП'ЮТЕРНА ІНЖЕНЕРІЯ: ПЕРСПЕКТИВИ, ПРОБЛЕМИ ТА МОЖЛИВОСТІ ПРОФЕСІЙНОГО РОЗВИТКУ.....	262
Полягушко Л.Г., Ткачук А.Ю. МЕТОДИ ПІДВИЩЕННЯ ЯКОСТІ ВІДЕОЗВ'ЯЗКУ В СИСТЕМАХ ДИСТАНЦІЙНОГО КОНСУЛЬТУВАННЯ.....	264
Dmytriieva I., Yushkovskiy P. DEVELOPMENT OF A METHOD FOR USING GRAPHICAL DIAGRAMS FOR QUERIES TO LARGE LANGUAGE MODELS.....	266
Антощук С.А., Трофименко М.А., Власко Д.Є., Лучик В.Є. ПОРІВНЯННЯ ІНСТРУМЕНТІВ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ХМАРНИХ СЕРЕДОВИЩ AWS, AZURE ТА GOOGLE CLOUD.....	270
Mahmudova M., Valiyev K. ARTIFICIAL INTELLIGENCE AND THE HUMAN BRAIN: SIMILARITIES AND DIFFERENCES.....	272
Вайс Т.Й., Сіра О.В. ЗАСТОСУВАННЯ МЕТОДУ МУРАШИНИХ КОЛОНІЙ ДЛЯ ОПТИМІЗАЦІЇ МОДЕЛЕЙ ПРОГНОЗУВАННЯ ЦІН КРИПТОВАЛЮТ.....	278

АРХІТЕКТУРНІ ПАТЕРНИ У ПОБУДОВІ ФРЕЙМВОРКІВ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ ВЕБДОДАТКІВ

Флегантов Леонід

кандидат фізико-математичних наук, доцент,
професор кафедри інформаційних систем та технологій

Даценко Назар

здобувач вищої освіти магістерського рівня
спеціальність «Інформаційні системи та технології»
Полтавський державний аграрний університет, Україна

В умовах стрімкого розвитку інформаційних технологій та ускладнення архітектури програмних систем, забезпечення якості є одним з головних пріоритетів у життєвому циклі розробки програмного забезпечення.

У цьому контексті автоматизація тестування виступає як методологія, що передбачає використання спеціалізованих програмних засобів для виконання тестових сценаріїв з метою верифікації та валідації функціональних і нефункціональних вимог до продукту. Автоматизація тестування полягає у створенні та використанні скриптів, які імітують взаємодію користувача з додатком або виконують запити до API для порівняння фактичного результату роботи системи з очікуваним результатом. На відміну від ручного тестування, яке покладається на людське втручання, автоматизований підхід забезпечує високий рівень повторюваності, точності та масштабованості тестових процедур за допомогою точних скриптів.

Однією з головних проблем автоматизації тестування є висока вартість підтримки тестів. Без чіткої та продуманої архітектури тестовий проект швидко втрачає структуру, перетворюючись на набір тестів, що є складними для підтримки, які стає неможливо підтримувати. Для вирішення цієї проблеми в інженерній практиці використовуються архітектурні патерни перевірені часом, універсальні рішення для типових проблем проектування.

Патерни в автоматизації тестування визначають структуру та взаємозв'язки між компонентами: вони задають структуру, визначають взаємозв'язки між компонентами та забезпечують довгострокову стійкість усієї конструкції. Їх головна мета підвищити читабельність (readability), легкість підтримки (maintainability) та повторного коду (reusability).

Найбільш поширеним та фундаментальним патерном у тестуванні вебдодатків є об'єктна модель сторінки (Page Object Model, POM). Його головна ідея полягає у розділенні відповідальностей: логіка самого тесту (що перевіряється) має бути повністю відокремлена від технічної реалізації взаємодії зі сторінкою (як це робиться) [1].

Згідно з цим патерном, кожна вебсторінка або значущий UI-компонент (наприклад, форма логіну, хедер сайту) представляється у вигляді окремого

класу в програмному кодї «об'єкта сторінки». Цей клас інкапсулює в собі дві ключові речі: елементи сторінки – локатори (селектори) для всіх інтерактивних елементів, що знаходяться на цій сторінці (кнопки, поля вводу, посилання); методи для взаємодії – публічні методи, що імітують дії користувача з цими елементами (наприклад, `login(username, password)`, `clickSearchButton()`, `getProductTitle()`) (рис. 1).

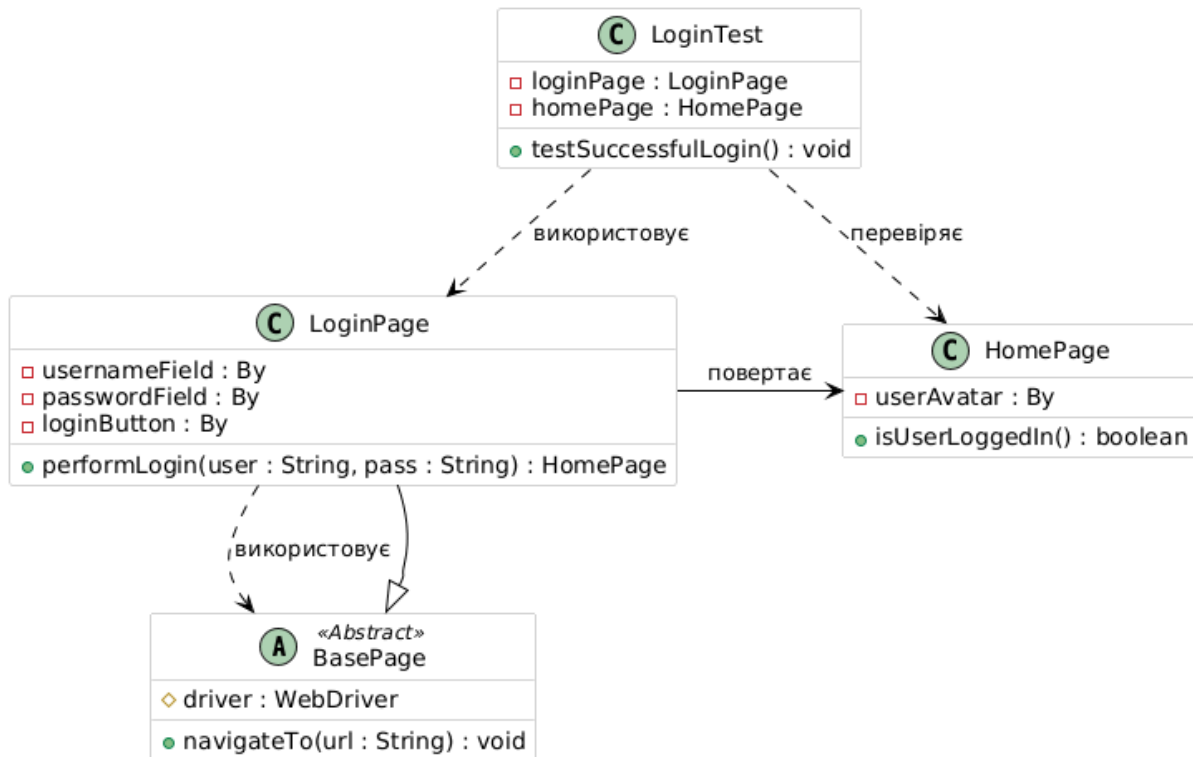


Рисунок 1. Принцип роботи POM.

Таким чином, завдяки POM, сам тест-кейс оперує не технічними деталями, подібними до `driver.findElement(By.cssSelector(«#login-btn»))`, а викликає високорівневі, зрозумілі методи об'єкта сторінки: `loginPage.performLogin(«user», «pass»)`. Головна перевага такого підходу: якщо в майбутньому розробники змінять, наприклад, селектор кнопки входу, то зміни потрібно буде внести лише у відповідному класі Page Object, а не в десятках тестів, які використовують цю кнопку. Це робить тестовий набір надзвичайно стійким до змін в інтерфейсі.

Окрім очевидної переваги у підтримці, використання POM, покращує ще два важливі аспекти – читабельність та повторне використання коду. Завдяки цьому тестові сценарії стають лаконічними, оскільки вони те, що має відбуватись, на бізнес рівні, а не так, як це реалізовано у DOM-елементах. І це робить логіку процесу більш зрозумілою не лише для QA, а й іншим членам команди розробки. Водночас, методи, що використовуються у програмі тестування, перетворюються на багаторазові блоки, які можна комбінувати у різних тест-кейсах, що значною мірою зменшує дублювання коду та прискорює написання нових тестів.

Screenplay Pattern є більш сучасною та просунутою еволюцією ідеї POM. Якщо POM фокусується на структурі вебсторінок, то Screenplay концентрується

на намірах та ролях користувача. Цей патерн прагне зробити тести ще більш читабельними та масштабованими, організовуючи їх навколо дійових осіб [2].

В основі Screenplay покладено декілька головних абстракцій. Центральною фігурою тут є Actor (дійова особа), що представляє користувача, наділеного певними Abilities (здібностями), наприклад, можливістю взаємодіяти з веббраузером. Дійова особа виконує високорівневі Tasks (завдання), що складаються з кількох логічних кроків, наприклад: «Авторизуватися в системі». Кожне таке завдання, у свою чергу, декомпозиється на послідовність низькорівневих, атомарних Interactions (взаємодій), таких як «клікнути на кнопку» або «ввести текст» (рис. 2).

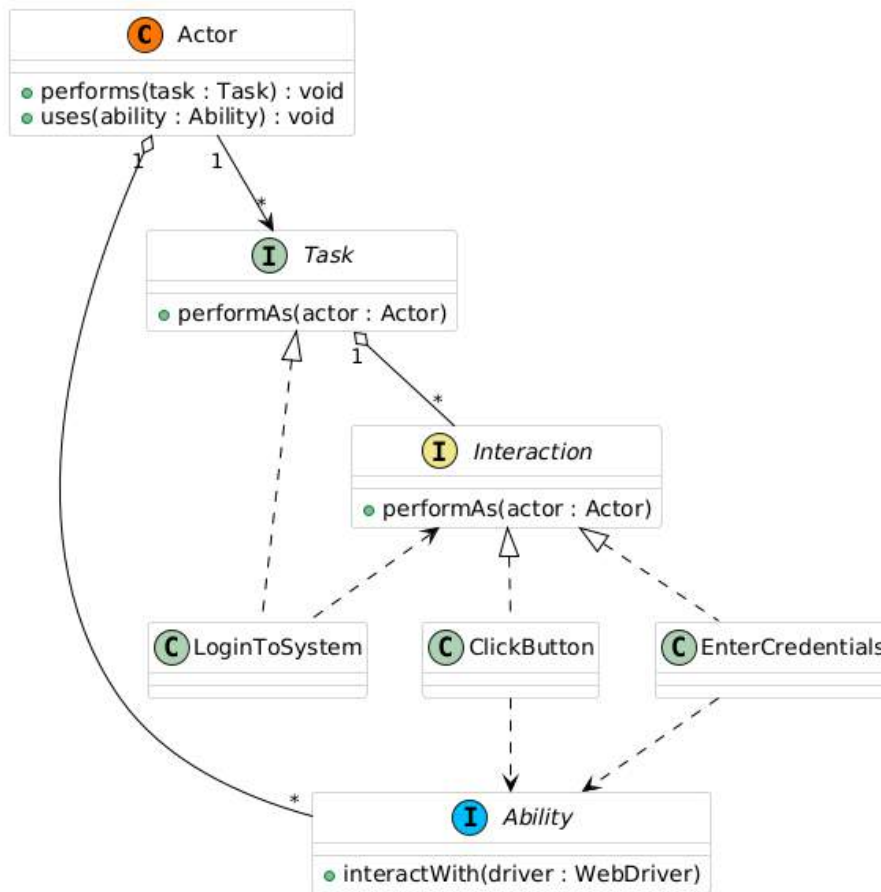


Рисунок 2. Принцип роботи Screenplay Pattern.

Окрім специфічних для тестування патернів, у побудові фреймворків активно використовуються класичні патерни проектування з об'єктно-орієнтованого програмування, такі, як Factory і Singleton.

Factory це патерн, що дозволяє створювати об'єкти, не розкриваючи логіку їх створення (рис. 3). У контексті автоматизації він найчастіше використовується для ініціалізації вебдрайвера. Для цього створюється спеціальний клас WebDriverFactory, який на основі коду конфігураційного файлу (наприклад, рядка «chrome» або «firefox») сам вирішує, екземпляр якого саме драйвера потрібно створити й налаштувати. Це дозволяє легко перемикатися між різними браузерами, не змінюючи код тестів [3].

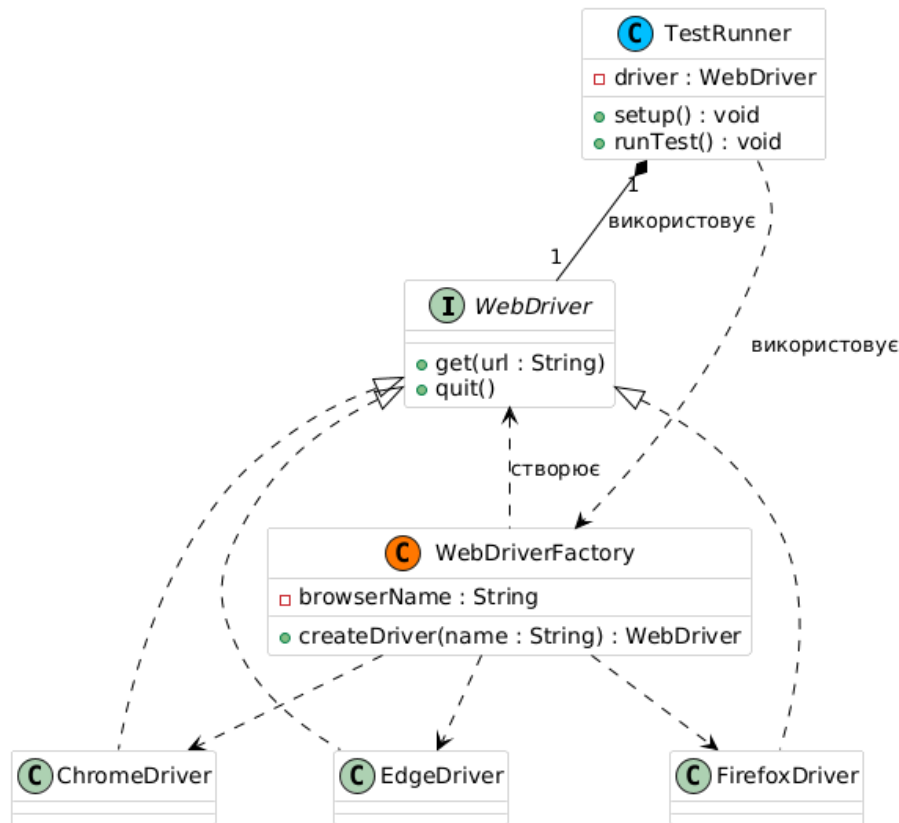


Рисунок 3. Принцип роботи паттерну Factory.

Singleton це патерн, що гарантує існування лише одного екземпляра певного класу та надає глобальну точку доступу до нього (рис. 4). У фреймворках автоматизації він часто застосовується для керування екземпляром драйвера, щоб уникнути створення нового вікна браузера для кожного тесту, що значно економить ресурси. Також його використовують для класів, що читають конфігураційні файли, дані з них потрібно завантажити лише один раз на початку виконання всіх тестів [4].

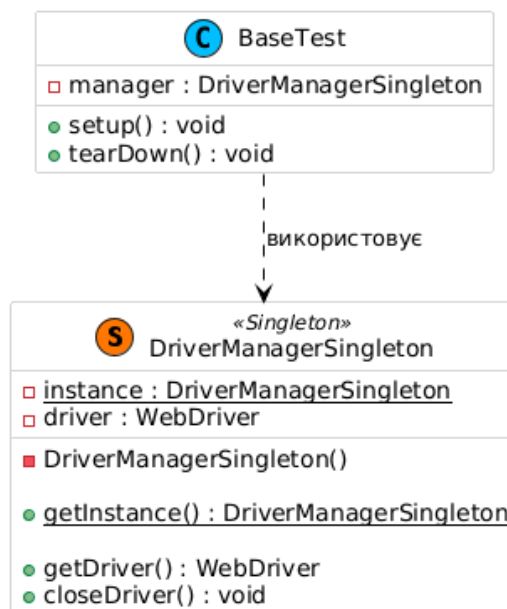


Рисунок 4. Принцип роботи паттерну Singleton.

Поєднання цих архітектурних патернів дозволяє створювати потужні, гнучкі та легкі у підтримці фреймворки, які служать надійною основою для всього процесу автоматизованого тестування.

Хоча розглянуті патерни, такі як POM та Screenplay, визначають фундаментальну структуру та спосіб взаємодії з додатком, ефективний тестовий фреймворк повинен вирішувати ще два важливих завдання: керування тестовими даними та забезпечення максимальної прозорості й читабельності тестів для всіх учасників проекту. Для вирішення цих завдань застосовуються додаткові архітектурні підходи.

Одним із найбільш потужних підходів до організації тестування є Data-Driven Testing (тестування, кероване даними). Його головна ідея полягає в повному відокремленні логіки тестового сценарію від тестових даних, що використовуються в цьому сценарії. Замість того, щоб жорстко «зашивати» вхідні значення (логіни, паролі, пошукові запити) безпосередньо в код тесту, ці дані виносяться у зовнішні джерела [5].

Як приклад, розглянемо тест для перевірки форми входу (рис. 5). Без використання DDT, щоб перевірити 10 різних комбінацій логіна та пароля (валідних, невалідних, порожніх), довелося б написати 10 практично ідентичних тестів або функцій. Це призводить до масивного дублювання коду та ускладнює підтримку.

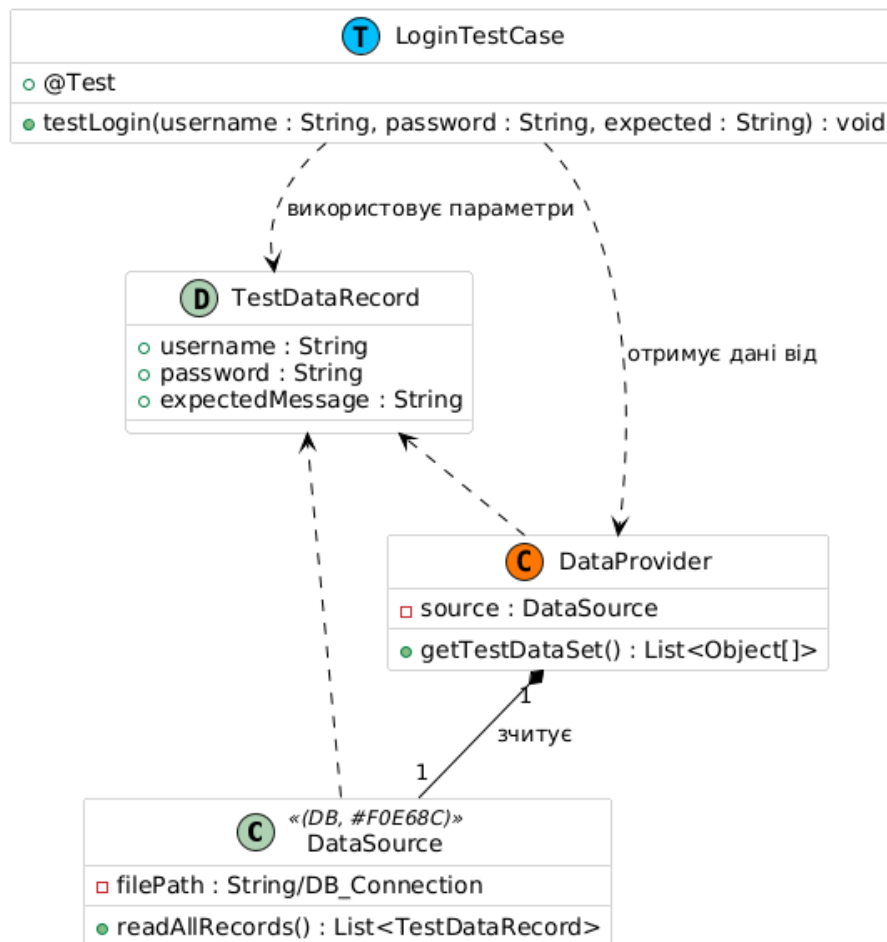


Рисунок 5. Принцип роботи паттерну Data-Driven Testing.

Паттерн DDT вирішує цю проблему. Створюється єдиний тестовий скрипт, який є своєрідним шаблоном. А всі набори даних зберігаються окремо, наприклад, у файлах формату CSV, Excel, JSON, YAML або навіть у базі даних. Фреймворк автоматично зчитує цей файл рядок за рядком і виконує один і той самий тест для кожного набору даних, передаючи їх у якості параметрів.

Ще одним важливим підходом, що удосконалює архітектуру тестування, є Behavior-Driven Development (BDD) (рис. 6). Це не просто патерн, а ціла методологія, що виросла з ідей Test-Driven Development (TDD), і має за мету налагодити ефективну комунікацію між бізнесом та командою розробки [5].

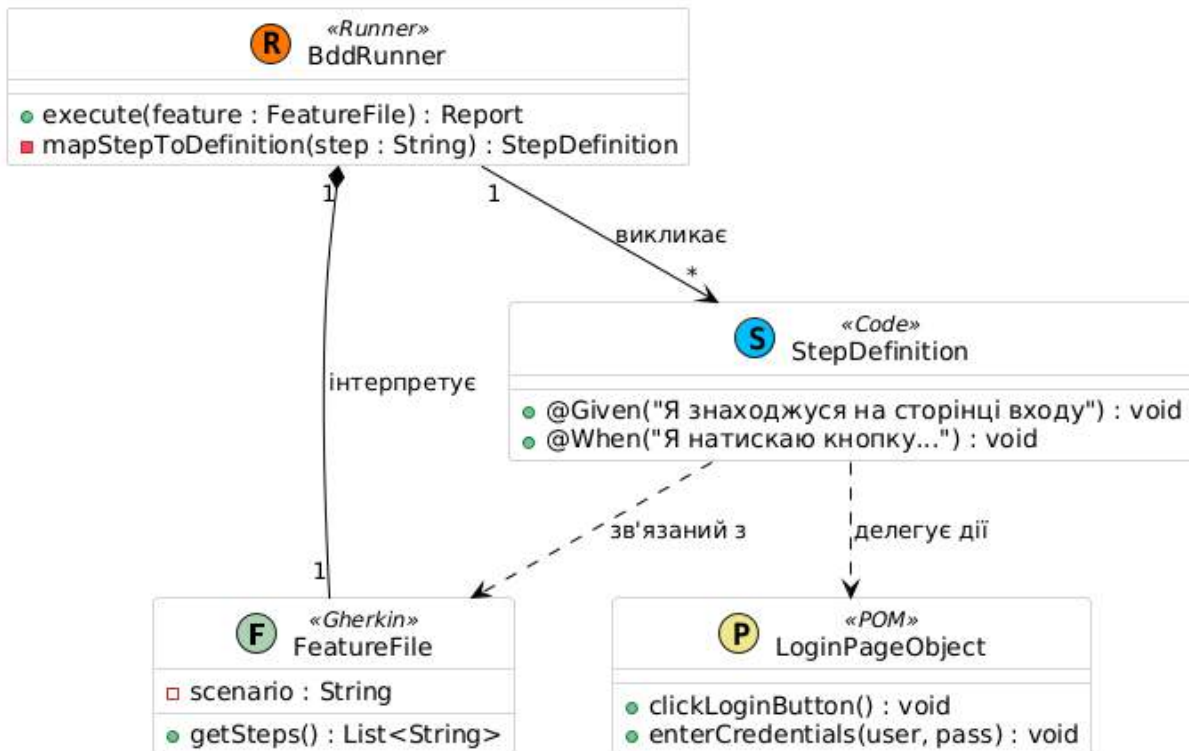


Рисунок 6. Концептуальна структура паттерну Behavior-Driven Development.

В основі BDD лежить концепція створення єдиної, спільної мови для опису поведінки системи, яка була б зрозумілою для всіх учасників процесу. Для цього використовується спеціальний синтаксис, наприклад Gherkin, який дозволяє описувати тестові сценарії у форматі, наближеному до природної мови. Кожен сценарій структурується за допомогою ключових слів, що описують передумови (Given), дії користувача (When) та очікувані результати (Then). Такий текстовий опис поведінки системи стає «живою документацією» єдиним джерелом правди, яке одночасно є і специфікацією, і виконуваним тестом. З технічної точки зору, кожен крок тестового сценарію пов'язується з конкретним методом у програмному коді, який реалізує дану дію. Таким чином, BDD виступає як додатковий семантичний шар над такими патернами, як POM, що робить автоматизовані тести максимально прозорими та бізнес-орієнтованими.

Для підвищення читабельності самого програмного коду на рівні виклику тестових методів часто застосовується патерн Fluent Interface (рис. 7). Його головна ідея полягає в такому проектуванні методів, що використовуються у коді

тестового сценарію, щоб їх можна було викликати послідовним ланцюжком, який візуально та семантично нагадує зв'язне речення. Це досягається за рахунок того, що кожен метод у ланцюжку повертає екземпляр того ж самого об'єкта, що дозволяє одразу ж викликати наступний метод [6].

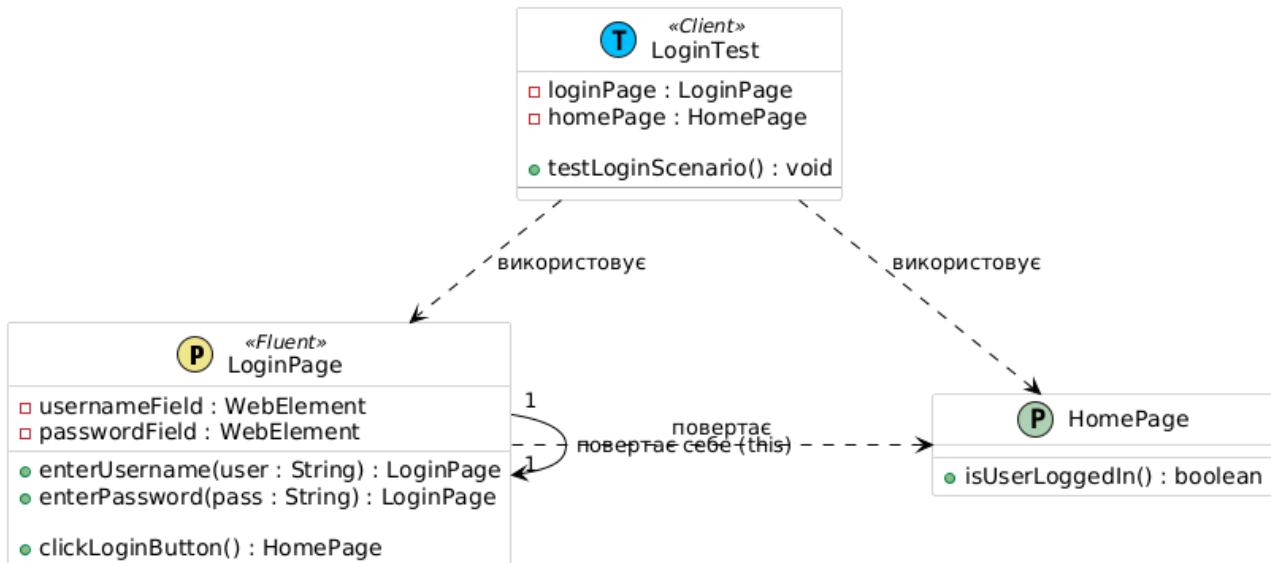


Рисунок 7. Принцип роботи патерну Fluent Interface.

Застосування патерну Fluent Interface перетворює послідовність окремих команд на єдиний, логічно завершений вираз, що значно покращує розуміння логіки тесту. Такий код не лише виглядає елегантніше, але й краще відображає послідовність дій користувача, ніби розповідаючи історію його взаємодії з додатком. Патерн Builder є більш структурованою та формалізованою версією цього підходу і є особливо корисним при конструюванні складних тестових об'єктів або підготовці тестового оточення з багатьма параметрами [3].

Комплексне застосування цих патернів та підходів – POM для структурування взаємодії з UI, DDT для керування даними, BDD для бізнес-орієнтованого опису сценаріїв та Fluent Interface для підвищення читабельності коду, – дозволяє створювати не просто гнучкий набір тестів, а повноцінну, масштабовану та легку в підтримці екосистему автоматизації тестування вебдодатків.

Список використаних джерел

1. Pcloudy. All You Need To Know About Automation Testing Life Cycle. Pcloudy: вебсайт. 2023. URL: <https://www.pcloudy.com/blogs/all-you-need-to-know-about-automation-testing-life-cycle/> (дата звернення: 03.11.2025).
2. Khurana P. Screenplay Pattern Approach in Selenium. BrowserStack: вебсайт. 2023. URL: <https://www.browserstack.com/guide/screenplay-pattern-approach-in-selenium> (дата звернення: 03.11.2025).
3. Hasnaoui S. Design Patterns: Creational Patterns. Medium: вебсайт. 2024. URL: <https://medium.com/@saif-hasnaoui/design-patterns-creational-patterns-e5ac08e57ef2> (дата звернення: 03.11.2025).

4. Shankar A. Python Design Patterns Every AI Developer Should Know. ITNEXT: вебсайт. 2024. URL: <https://itnext.io/python-design-patterns-every-ai-developer-should-know-dc445b688793> (дата звернення: 03.11.2025).
5. Mali A., Wagh M. Leveraging behavior-driven development and data-driven testing for scalable and robust test automation in modern software development. International Journal of Research in Engineering and Science (IJRES), 2024. Vol. 12, Issue 6.
6. Dranidis D. Define your Fluent Interface (DSL) to improve the readability of your tests. Dranidis: вебсайт. 2023. URL: <https://dranidis.gr/posts/define-your-dsl-for-tests/> (дата звернення: 03.11.2025).

DOI 10.70286/ISU-12.11.2025.007

ВИКОРИСТАННЯ РЕГРЕСІЙНОГО АНАЛІЗУ ДЛЯ ПРОГНОЗУВАННЯ ПОПИТУ У ВЕБОРІЄНТОВАНИХ АНАЛІТИЧНИХ СИСТЕМАХ

Лошик Олександр

здобувач вищої освіти магістерського рівня
спеціальності інженерія програмного забезпечення

Корень Віталій

Асистент

Кафедра інженерії програмного забезпечення
Луцький національний технічний університет, Україна

У сучасних економічних умовах ефективність роботи підприємств значною мірою залежить від здатності прогнозувати обсяги попиту на продукцію та оптимізувати управлінські рішення. Розвиток електронної комерції, маркетплейсів і роздрібною торгівлі вимагає інструментів, які можуть аналізувати великі обсяги даних у реальному часі, виявляти закономірності у поведінці споживачів і передбачати майбутні тренди. Одним із найперспективніших напрямів у цій галузі є інтеграція регресійного аналізу у веборієнтовані аналітичні системи, що поєднують методи машинного навчання, аналітику даних та зручні користувацькі інтерфейси.

Метою дослідження є вивчення можливостей застосування регресійних моделей у процесі прогнозування попиту на товари в інтерактивних вебсистемах, а також визначення оптимального набору технологій для побудови таких аналітичних рішень.

Регресійний аналіз розглядається як універсальний метод моделювання залежностей між обсягами продажів та факторами впливу – такими, як ціна, сезонність, рекламні витрати, дохід споживачів або зовнішні ринкові умови [1]. Наукові праці дослідників показують, що регресійні методи забезпечують

Collection of Scientific Papers
with Proceedings of the 3rd International Scientific and Practical Conference
«Innovative Approaches in Modern Science and Technology»
November 12-14, 2025
Lisbon, Portugal

Organizing committee may not agree with the authors' point of view.
Authors are responsible for the correctness of the papers' text.

Contact details of the organizing committee:
Sole Proprietor Viktoriia Tsiundyk
E-mail: info@isu-conference.com
URL: <https://isu-conference.com/>

Certificate of the subject of the publishing business: ДК №7980 of 03.11.2023.