

й етичної точки зору, щоб забезпечити її відповідальне та ефективне використання в освітніх установах.

### Список використаних джерел

1. Нова українська школа. Концептуальні засади реформування середньої школи, с. 10. URL: [https://mon.gov.ua/static-objects/mon/sites/1/zagalna %20serednya/nova-ukrainska-shkola-compressed.pdf](https://mon.gov.ua/static-objects/mon/sites/1/zagalna%20serednya/nova-ukrainska-shkola-compressed.pdf) (дата звернення: 11.10.2024).
2. Пометун О., Пироженко Л. Інтерактивні технології навчання : Теорія, практика, досвід : метод. посіб. К., 2004. С. 34–37.
3. Data Visualization in Education. URL: <https://www.wandr.studio/blog/data-visualization-in-education> (дата звернення: 04.10.2024).

## СУЧАСНІ ІНСТРУМЕНТИ ТА ФРЕЙМВОРКИ АВТОМАТИЗОВАНОГО ФУНКЦІОНАЛЬНОГО ТЕСТУВАННЯ WEB-ЗАСТОСУНКІВ

### Малярський Віктор Олегович

здобувач третього (освітньо-наукового) рівня вищої освіти спеціальності 015 «Професійна освіта (за спеціалізаціями)»,  
Тернопільський національний педагогічний університет імені Володимира Гнатюка,  
[vmalarskij@gmail.com](mailto:vmalarskij@gmail.com)

З розвитком галузі автоматизації тестування програмного забезпечення з'явилась велика кількість інструментів і фреймворків, які пропонують різні функціональні можливості та підходи. Це ускладнює здійснення вибору цих засобів для вирішення певних професійних завдань і вимог, особливо в умовах обмеженого бюджету або ресурсів. Також ця проблема є актуальною для викладачів закладів вищої освіти або лекторів приватних курсів з тестування програмного забезпечення, оскільки для створення ефективної методики викладання є важливим правильний підбір інструментів тестування ПЗ для формування навчального матеріалу. Аналіз існуючих інструментів може допомогти зрозуміти їхні переваги й обмеження та вирішити наведені вище проблеми.

*Сучасні інструменти та фреймворки автоматизованого функціонального тестування вебзастосунків*

Найбільш ефективним методом оцінки якості вебзастосунків сьогодення можна вважати розробку та застосування фреймворків для автоматизованого тестування. Це програмні засоби, які є набором інструкцій та інструментів, що дозволяють автоматично проходити налаштовані тестові сценарії за допомогою спеціальних механізмів. Найпоширенішим з них є вебдрайвер [2]. Цей інструмент дозволяє взаємодіяти зі сторінками, елементами та внутрішніми запитами вебзастосунків за допомогою скриптів та команд веббраузеру. Кожен браузер має власний вебдрайвер, який розроблений для взаємодії з ним. До прикладу в Google Chrome – це chromedriver, а для Firefox – geckodriver. Також популярним є застосування фреймворків, які не взаємодіють з командами веббраузерів, а створюють нові, безпосередньо втручаючись у їх роботу.

Переваги фреймворків автоматизації:

Ефективність: фреймворки автоматизації спрощують процес тестування, зменшуючи потребу у повторюваних проходженнях тестових сценаріїв вручну, що призводить до швидшого аналізування якості застосунків та скорочення циклів розробки.

**Послідовність і надійність:** автоматизовані тести виконуються щоразу з однаковими кроками, що допомагає забезпечити послідовність виконання та надійність результатів. Це знижує ймовірність людських помилок і забезпечує більш надійні результати тестів.

**Масштабованість:** фреймворки автоматизації можуть проходити великий обсяг тестів у різних середовищах та конфігураціях, зменшуючи складність тестування в міру розширення функціоналу додатка.

**Раннє виявлення дефектів:** автоматизація допомагає виявляти дефекти на ранніх стадіях процесу розробки, що сприяє швидшому їх виправленню та зниженню витрат.

**Регресійне тестування:** фреймворки автоматизації особливо корисні для регресійного тестування, де ті самі тести потрібно повторювати, щоб переконатися, що нові зміни не вплинули негативно на наявний функціонал.

**Розширене покриття тестами:** автоматизація дозволяє значно розширити покриття тестами, забезпечуючи проходження великої кількості тестових сценаріїв, включаючи граничні та складні сценарії, які можуть бути важкими для ручного тестування.

**Ефективне використання ресурсів:** автоматизація повторюваних тестів дозволяє командам зосередитися на більш складних завданнях тестування та дослідженнях застосунків. Це оптимізує використання людських ресурсів і дозволяє краще розподілити зусилля.

**Зниження витрат:** автоматизація забезпечує економію бюджету у довгостроковій перспективі, зменшуючи час і зусилля, необхідні для ручного тестування. Ефективність і масштабованість автоматизації можуть призвести до значної економії часу.

**Точність:** автоматизовані тести менш схильні до помилок у порівнянні з ручними тестами. Вони щоразу виконують одні й ті ж кроки з точністю, що призводить до достовірних і стабільних результатів.

**Підтримка безперервного тестування:** фреймворки автоматизації добре інтегруються з CI/CD процесами, дозволяючи безперервне тестування змін у коді. Це допомагає підтримувати якість і стабільність коду протягом усього життєвого циклу розробки [3].

Найпопулярніші фреймворки автоматизації за даними ресурсів BrowserStack [1], LambdaTest, SauseLabs подано у таблиці 1.

Таблиця 1

**Фреймворки автоматизації тестування вебзастосунків**

Назва фреймворку	Короткий опис	Переваги	Недоліки
Selenium	Open Source проєкт створений спільнотою користувачів, який складається з декількох підсистем, таких як IDE, Grid, WebDriver та інші [4]. Вони пов'язані між собою з метою максимального спрощення процесу автоматизації	Підтримка багатьох мов програмування: Java, JavaScript, C#, Python, Ruby, тощо;	Складний в розробці та конфігураціях;
		Підтримка багатьох операційних систем: Windows, Mac, Linux;	Невелика кількість вбудованих методів для роботи з елементами сторінок;
		Підтримка багатьох операційних систем: Windows, Mac, Linux;	Залежний від багатьох додаткових програмних пакетів;

Назва фреймворку	Короткий опис	Переваги	Недоліки
	тестування вебдодатків. Цей фреймворк існує на ринку з 2004 року, проте досі користується популярністю через свою зручність та багатозадачність.	Легкість у вивченні та зручність у використанні;	Дуже рідко оновлюється та підтримується лише спільнотою.
		Можливість запуску тестів у паралельних потоках;	
		Можливість широкої кастомізації налаштувань та методів;	
		Доступ до багатьох існуючих бібліотек та модулів, створених спільнотою, що спрощують роботу та дають можливість використовувати готовий код;	
		Підтримка тестування мобільних застосунків	
Cypress	Фреймворк автоматизованого тестування вебзастосунків, що має власний вебінтерфейс, хмарне середовище, в якому зберігаються результати тестів (лише у платній версії), дозволяє виправляти помилки тестових скриптів прямо в процесі тестування, а також генерує локатори елементів після звичайного натискання по них курсором миші.	Містить багато вбудованих методів для роботи з елементами сторінок а також HTTP запитами;	Не підтримує переходи між різними доменними адресами;
		Зручний користувацький інтерфейс;	Підтримуються лише мови програмування JavaScript і TypeScript;
		Підтримка всіх основних браузерів: Chrome, Edge, Firefox, Safari;	Не підтримуються BDD фреймворки, такі як Cucumber.
		Підтримка E2E та Компонентного тестування	Ланцюгова техніка написання коду (кожен метод повертає об'єкт свого класу) що ускладнює використання найпопулярнішого архітектурного патерну програмування фреймворків – Page Object Model;
Playwright	Найсучасніший фреймворк, який стрімко розвивається,	Підтримка багатьох мов програмування: JavaScript, C#, Python;	Не підтримує мобільне тестування

Назва фреймворку	Короткий опис	Переваги	Недоліки
	та пропонує велику кількість вбудованих функцій для роботи з вебсторінками, АРІ та базами даних. Великою перевагою цього фреймворку є те, що він розгортається прямо у браузері, надсилаючи свої команди у потік Event Loop, що робить проходження тестів набагато швидшим.	Підтримка всіх основних браузерів: Chrome, Edge, Firefox, Safari;	Нестабільний функціонал, дуже часто оновлюється
		Не застосовує вебдрайвер, що надає тестам швидкості та стабільності;	
		Можливість записувати проходження тестів для майбутнього аналізу;	
WebdriverIO	Фреймворк на базі Selenium, що підтримує лише мови програмування JavaScript і TypeScript. Його перевагою є те, що він новіший і розробники досі постійно оновлюють його функціонал на відміну від Selenium. А також він має більше вбудованих функцій для роботи з елементами вебсторінок.	Підтримка всіх основних браузерів: Chrome, Edge, Firefox, Safari;	Підтримуються лише мови програмування JavaScript і TypeScript;
		Легко кастомізується, тобто дозволяє використовувати будь-які підходи до розробки;	Залежність від функціоналу вебдрайвера;
		У новій версії фреймворку, що була випущена влітку 2024 року з'явилась функція автоматичного очікування завантаження всіх елементів та запитів сторінки перед їх верифікацією, що надає більшої стабільності тестам;	Складність міграцій на нові версії через велику кількість програмних пакетів та залежностей; Складність у налаштуванні фреймворку для повної відповідності вимогам.

Існує ще багато інших фреймворків автоматизації функціонального тестування вебзастосунків, проте їх використання не на стільки популярне на ринку, хоча кожен з них заслуговує окремої уваги та має свої переваги та недоліки.

Сучасні фреймворки для автоматизації функціонального тестування вебзастосунків дозволяють швидко, легко та зручно виявляти помилки на стадії розробки програм, проходити велику кількість тестових сценаріїв за короткий час, що дозволяє оптимізувати процес оцінки якості програмного забезпечення та зекономити виділені ресурси. Основними перевагами популярних фреймворків є підтримка багатьох мов програмування та операційних систем для їх розробки, підтримка різних браузерів для запуску тестів, легкість та зручність у налаштуванні а основне наявність вбудованих методів очікування та валідації елементів вебсторінок. У подальших дослідженнях варто детальніше розглянути можливості використання сучасних фреймворків автоматизації для проведення мобільного тестування а також АРІ тестування. Оскільки з аналізу описаних фреймворків можна зробити висновок, що більшу популярності у використанні

отримують саме мультифункціональні фреймворки, які дозволяють працювати не лише з клієнтською частиною вебзастосунків а і використовувати вбудований допоміжний функціонал для роботи з серверною частиною або мобільними додатками.

### Список використаних джерел

1. Best Test Automation Frameworks in 2024. URL: <https://www.browserstack.com/guide/best-test-automation-frameworks> (дата звернення: 01.11.2024).
2. MDN web docs – WebDriver. URL: <https://developer.mozilla.org/en-US/docs/Web/WebDriver> (дата звернення: 01.11.2024).
3. Oliinyk B., Oleksiuk V. Automation in software testing, can we automate anything we want? URL: <https://ceur-ws.org/Vol-2546/paper16.pdf> (дата звернення: 01.11.24).
4. Selenium Project structure and governance. URL: <https://www.selenium.dev/project/> (дата звернення: 01.11.2024).

## ЦИФРОВИЙ КОНТЕНТ ДЛЯ СУПРОВОДУ ВИВЧЕННЯ ПОНЯТТЯ ДОВЖИНИ В ПОЧАТКОВІЙ ШКОЛІ

### Мислицька Наталія Анатоліївна

доктор педагогічних наук, професор кафедри науково-природничих та математичних дисциплін,  
Комунальний заклад вищої освіти «Вінницький гуманітарно-педагогічний коледж»,  
[mislitskay@gmail.com](mailto:mislitskay@gmail.com)

### Кирилюк Вікторія Василівна

здобувач першого рівня вищої освіти спеціальності Початкова освіта,  
Комунальний заклад вищої освіти «Вінницький гуманітарно-педагогічний коледж»,  
[klitnaviktoria@gmail.com](mailto:klitnaviktoria@gmail.com)

В умовах становлення цифрового суспільства важливу роль відіграє правильно організоване освітнє середовище для вивчення предметів, як в початковій, так і в базовій та профільній школі. В останні де-кілька років МОН України активізувало діяльність щодо системного впровадження цифрового контенту в освітній процес. Саме тому проблема доповнення традиційних дидактичних засобів навчання математики цифровим контентом є актуальною.

Вивчення поняття довжини доцільно починати з мотивації і пояснення учням важливості знати дане поняття й уміти вимірювати довжини об'єктів. Зазначаємо учням, що, коли ми говоримо про довжину, ми говоримо про відстань від одного місця до іншого, від однієї точки до іншої, наприклад, від однієї сторони торта до іншої, відстань від дому до школи. Учням необхідно довести, що зміст терміну «довжина» асоціюється з лінійною протяжністю предмета, щоб у них формувалось розуміння, що чим більшу протяжність має предмет, тим його довжина більша [1, с. 24]. Для цього учитель може наводити приклади з життя, користуватись друкованими дидактичними засобами, задачами з підручника та цифровим контентом у вигляді слайдів з презентації «Вимірювання довжини», де наводяться конкретні приклади використання знань про довжину та її вимірювання.

Важливо у учнів сформувати уявлення, що для вимірювання довжини можна використовувати різні мірки. Слід зауважити, що вимірювання довжини різними мірками передбачено у закладах дошкільної освіти, тому як правило, діти вже ознайомлені з цими уміннями. На уроках математичної освітньої галузі узагальнюється уявлення дітей про залежність числового результату від величини