

Міністерство освіти і науки України  
Харківський національний університет імені В.Н. Каразіна  
Факультет комп'ютерних наук

### **ЛАБОРАТОРНА РОБОТА № 3**

з дисципліни «Математичні методи та технології тестування та верифікації  
програмного забезпечення»

Тема: «Вивчення технології автоматизованого тестування запису та  
відтворення (Record and playback) на прикладі використання інтегрованого  
середовища розробки Selenium IDE»

Виконав студент 2 курсу  
групи КС-21  
Безрук Юрій Русланович  
Перевірив:  
Доцент Нарєжний О. П.

Харків – 2020

Целью данной работы является знакомство с технологией автоматизированного тестирования записи и воспроизведения (Record and playback) на примере использования интегрированной среды разработки Selenium IDE и получение практических навыков для составления тест-сьютов для тестирования веб-приложений.

## ХОД РАБОТЫ

Selenium IDE (integrated development environment) - это инструмент, используемый для разработки тестовых сценариев, устанавливаемый как аддон на браузер.

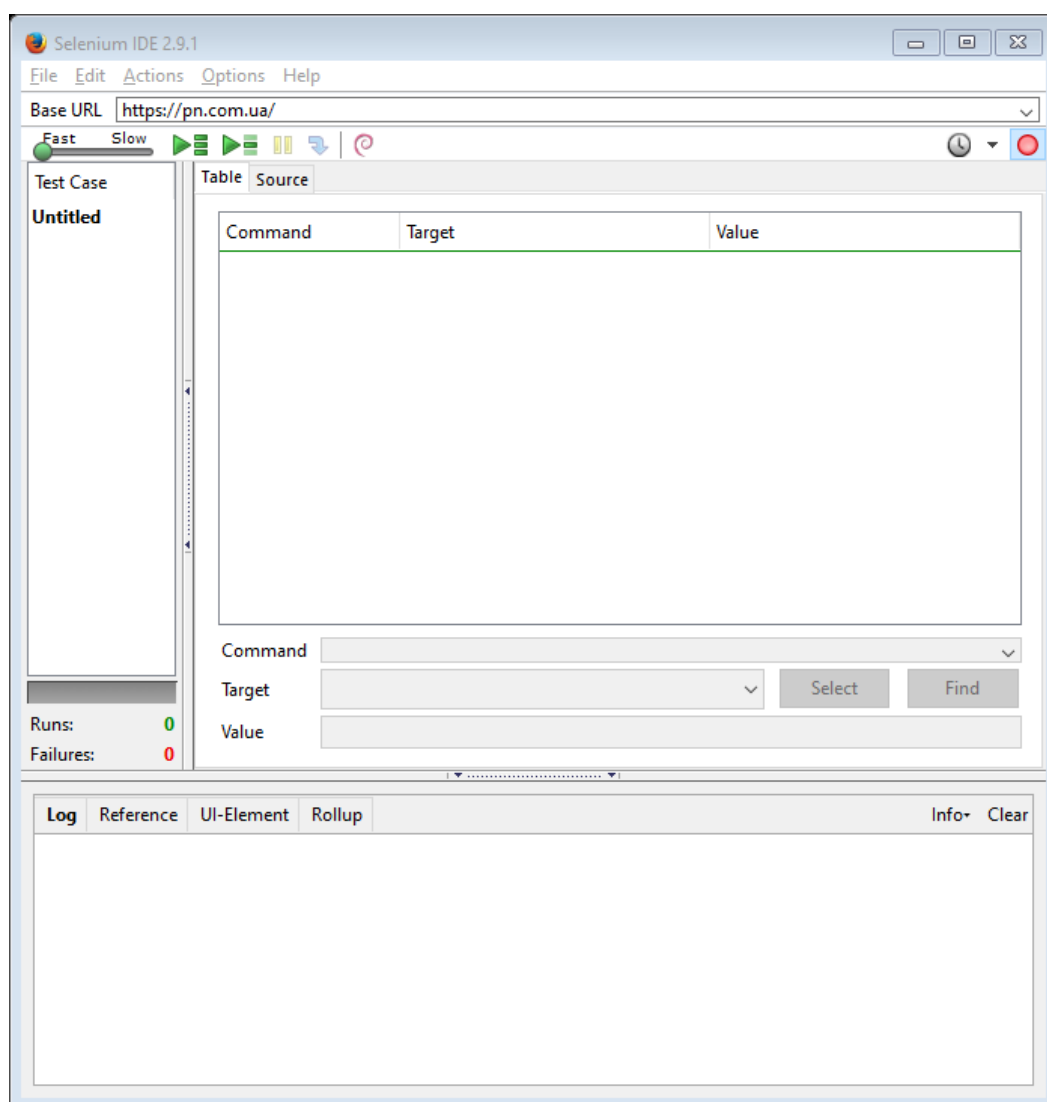


Рисунок 1 - Selenium IDE

# 1. Проверка, содержит ли страница строки «Фильтры», «Сортировка».

Предварительно открываем ссылку сайта при помощи команды `open`, в аргументы которой передается адрес сайта. Далее, для такого рода проверок в Selenium используются команды-считыватели. В данном случае подойдет команда проверки текста `TextPresent`, с префиксом строгости: `assert` – строгая проверка, которая прервет тест в случае неудачи и `verify` – которая запомнит результат и продолжит выполнение теста. После ввода трех команд в тест кейс запускаем его выполнение.

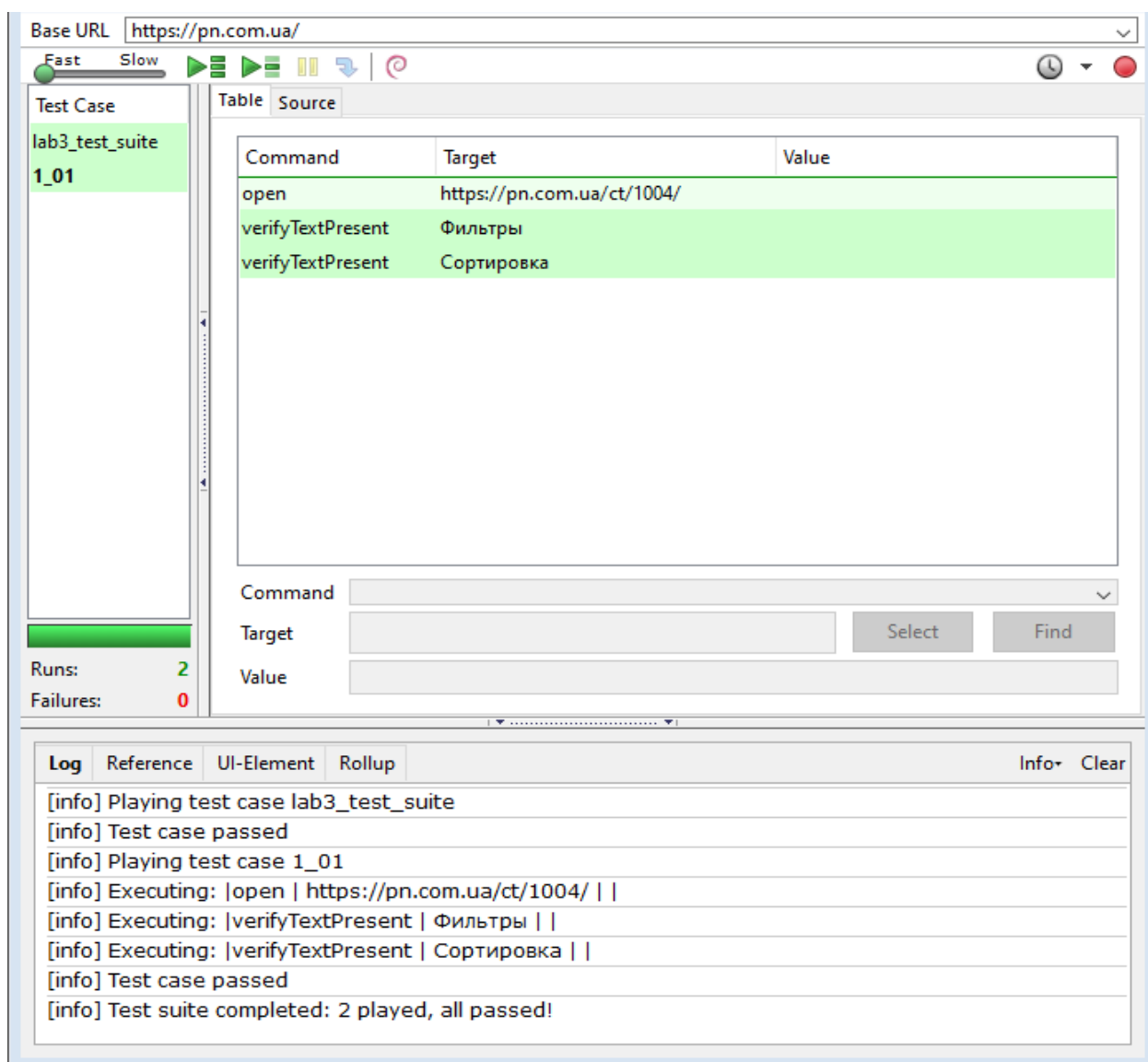


Рисунок 2 - выполнение первого тест-кейса

1_01	
open	https://pn.com.ua/ct/1004/
verifyTextPresent	Фильтры
verifyTextPresent	Сортировка

Рисунок 3 – первый тест-кейс

2. Перейти на украинскую версию и проверить, что вы находитесь на ней.

Для более удобного тестирования действий можно использовать режим записи в Selenium, вместо ввода команд вручную. Для этого нужно нажать на красную кнопку в правом верхнем углу окна. С помощью этого режима тестировщик записывает свои действия, которые потом будет воспроизводить программа. Собственно, активируем запись, и на исследуемом сайте нажимаем на кнопку UA для перехода на украинскую версию. Выключаем запись. В тест-кейс автоматически добавилось несколько команд, которые соответствуют действиям в браузере.

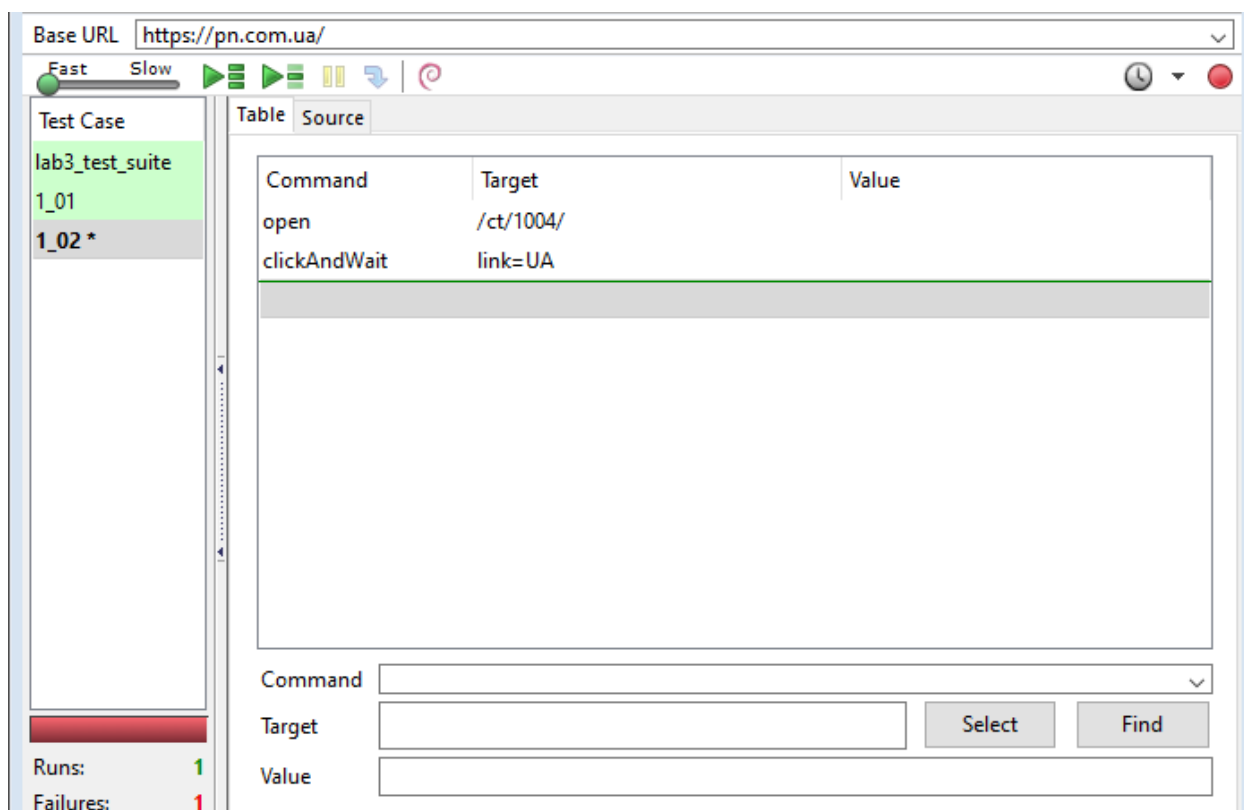


Рисунок 4 - записанные команды

Для того, что бы проверить, выполненлся ли переход на украинскую версию достаточно проверить, нет ли на сайте какого-либо русского слова. Можно даже использовать предыдущие запросы, т.к. они прошли проверку и точно присутствовали на сайте. Команда `verifyTextNotPresent`.

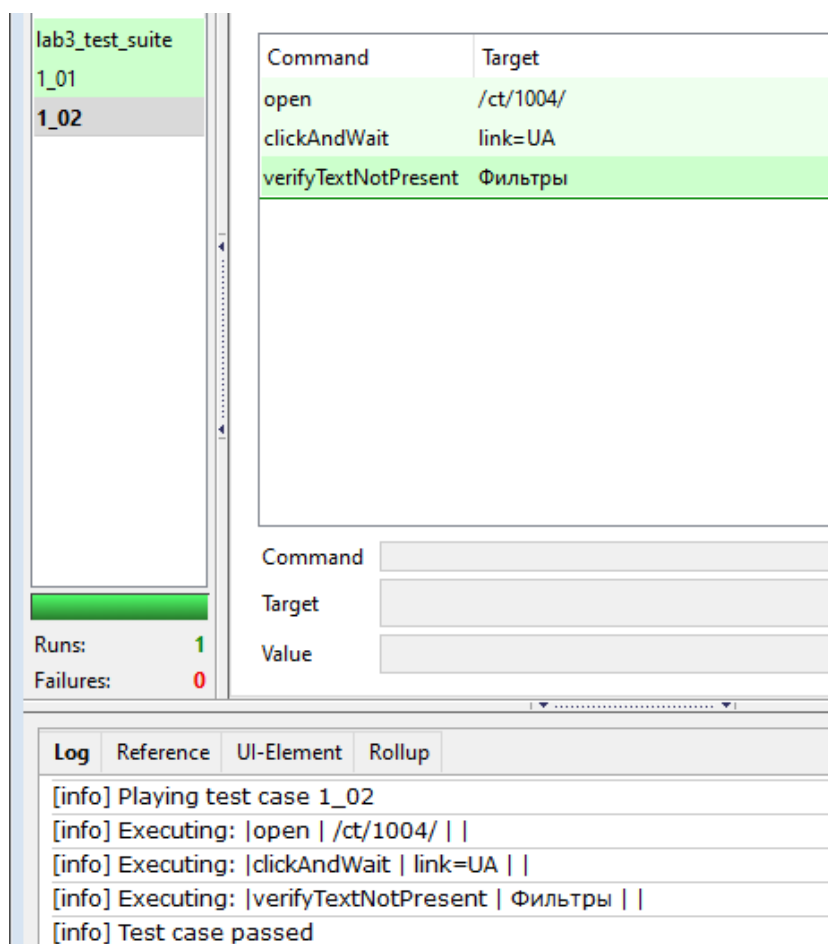


Рисунок 5 - успешно выполненный тест-кейс 2

1_02	
open	/ct/1004/
clickAndWait	link=UA
verifyTextNotPresent	Фильтры

Рисунок 6 - второй тест-кейс

### 3. Убедиться, что вы на первой странице.

Для того, что бы узнать, на первой ли мы странице, можно обратиться по XPath к панели страничек, и пройдя по ссылкам, просмотреть их атрибуты. Элемент, у которого атрибут class будет равен “page active” – текущая страница. Проверяем, равняется ли он 1. Если да, то мы на первой странице.

1_03		
verifyText	//div[@id='column-center']/section/div[1]/div[2]/ul/li[@class="page active"]/a	1

Рисунок 7 - третий тест-кейс

### 4. Вернуться на главную страницу.

Переход на главную страницу можно снова реализовать при помощи записи действий.

1_04		
clickAndWait	css=ul.breadcrumb > li > a > span	

Рисунок 8 - четвёртый тест-кейс

### 5. Выполнить на главной странице поиск (слово в таблице 1).

Убедиться, что вы выполнили правильный поиск.

С помощью записи получаем команды ввода поиска. Для примера введем «ноутбук». Selenium запишет нажатие на поиск, ввод слова и нажать на кнопку поиска. Для того, что бы удостовериться в результатах, через XPath проверяем описание первого объекта в результатах. Если первое ключевое слово – «Ультрабук», поиск выполнен правильно.

1_05		
open	/	
type	name=fn	ноутбук
clickAndWait	css=input[type="submit"]	
verifyText	//div[@id='column-center']/section/div[3]/article/div[2]/p/a/text()[1]	Ультрабук

Рисунок 9 - пятый тест-кейс

6. Выполнить другой поиск, любой другой поиск. Убедиться, что первые три результата не совпадают с прошлой категорией.

Аналогичным образом выполняется другой поиск, для примера «мониторы». После чего запускаем проверку описаний первых трех товаров, и, с помощью команды `verifyNotText` (возвращает истину, если у параметра не такой текст) с атрибутом прошлого запроса, проверяем что результат другой.

1_06		
open	/ct/5202/	
click	name=fn	
type	name=fn	монітори
clickAndWait	css=input[type="submit"]	
verifyNotText	//div[@id='column-center']/section/div[3]/article[1]/div[2]/p/a/text()[1]	ноутбук
verifyNotText	//div[@id='column-center']/section/div[3]/article[2]/div[2]/p/a/text()[1]	ноутбук
verifyNotText	//div[@id='column-center']/section/div[3]/article[3]/div[2]/p/a/text()[1]	ноутбук

Рисунок 10 - шестой тест-кейс

## ВЫВОДЫ

Selenium IDE является удобным инструментом, позволяющим пользователям выполнять такой процесс как автоматизированное тестирование. Благодаря его свойствам записи и воспроизведения становится возможным более эффективное отслеживание ошибок при работе того или иного веб-продукта. Пользователь может создавать, редактировать, отлаживать и выполнять автоматизированные тесты, а также собирать статистику их исполнения. После работы методом записи и воспроизведения (Record and Playback) можно сделать вывод, что такая технология проста, она обеспечивает высокую скорость создания тест-кейсов, хотя и качество такого тестирования в результате оказывается достаточно низкой.

Таким образом, в ходе выполнения данной лабораторной работы было проведено ознакомление с технологией автоматизированного тестирования записи и воспроизведения с помощью использования интегрированной среды разработки Selenium IDE.

В результате работы был составлен тест-сьюит из 6-ти тест-кейсов для тестирования веб-приложений.