

АВТОМАТИЗАЦІЯ ПЕРЕВІРКИ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Юхімов О.С.

Науковий керівник: к.т.н., доц. Маньковська В.С.

Актуальність даної теми. Створення будь-якої сторінки, навіть досвідченими розробниками, неможливе без помилок різного характеру, які можуть погіршити роботу сайту. За витратами часу і людських ресурсів найзатратнішими є етапи розробки, пов'язані з пошуком помилок у готових продуктах. Незважаючи на те, що зусилля, необхідні для внесення невеликих змін, як правило, мінімальні, вони можуть вимагати чималих зусиль для перевірки якості зміненої програми. І при досягненні певного критичного об'єму робіт по тестуванню web-сторінки за короткий проміжок часу одним з оптимальних рішень стає автоматизація процесу тестування.

Метою роботи є зменшення часу, що витрачається на проведення стандартних частовживаних тестів при розробці web-сторінок.

Для досягнення поставленої мети необхідно вирішити такі **задачі**:

- розглянути основні методи тестування при розробці сайтів та програм;
- проаналізувати основні методи і засоби автоматизованого тестування сайтів та програм;
- розробити тестові сценарії (скрипти) для автоматизованого тестування.

Об'єктом дослідження є процес розробки комплексів автоматизованого тестування функціонування web-сторінок.

Предметом дослідження є методи та засоби автоматизованого тестування для web-сторінок та комп'ютерних систем.

Науково-практичний результат полягає у розробці алгоритмічних та програмних засобів для забезпечення автоматизованого тестування веб-сторінок довільного змісту, що дозволило скоротити час на тестування та підвищити частку виявлених помилок, притаманних програмному забезпеченню ранніх стадій імплементації.

Тестування – це перевірка якості програмного забезпечення згідно з документацією до даного програмного забезпечення. Основна задача тестувальника – зменшити кількість неприємних сюрпризів для кінцевого користувача.

Тестування web-сайту, як і тестування будь-якого компоненту в цілому, – техніка контролю якості, що перевіряє відповідність між реальною і очікуваною поведінкою сторінки, яка описана в документації, завдяки кінцевому набору тестів, які обираються певним чином.

Тестування ділиться за такими рівнями:

- модульне;
- інтеграціне;
- системне;
- регресивне.



ФУНКЦІОНАЛЬНЕ ТЕСТУВАННЯ

Є найбільш тривалим етапом перевірки ресурсу, автоматизація якого є головною метою. Суть цього процесу полягає у перевірці всього описаного функціоналу, тобто можливості сторінки в певних умовах вирішувати завдання, потрібні користувачам :

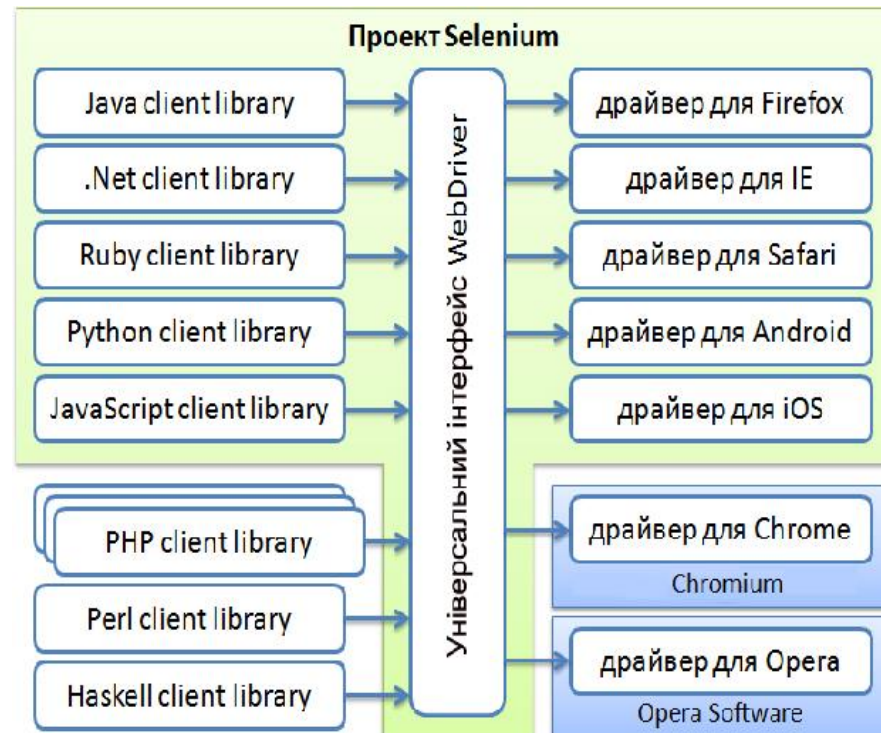
- роботи всіх обов'язкових функцій сайту;
- працездатності форм користувача на сайті (введення тексту, чисел, використання маски, робота з незаповненими полями, довжина символів, що вводять, коректний робота чекбоксів, комбобоксов, radio buttons, логічність установок "за замовчуванням", зворотній зв'язок, додавання коментаря в блог);
- роботи пошуку (включаючи релевантність результатів);
- гіперпосилань, пошук неробочих посилань;
- підвантаження файлів на сервер;
- працездатності лічильників, встановлених на сторінках сайту;
- перегляд на відповідність вмісту сторінок сайту вихідного контенту, наданого замовником;
- перевірка баз даних (пошук, додавання інформації, редагування, видалення, перевірка на дублювання інформації);
- наявності секретних частин (робота з паролями, передача даних, захист).

SELENIUM WEBDRIVER

Selenium WebDriver (Selenium 2.0) – це драйвер браузера, тобто програмна бібліотека, яка не має інтерфейсу користувача та дозволяє різним іншим програмам взаємодіяти з браузером, керувати його поведінкою, отримувати від браузера певні дані і змушувати браузер виконувати певні команди. Виходячи з цього визначення, видно, що WebDriver лише надає автотестам доступ до браузера.

За своєю сутністю Selenium WebDriver являє собою:

- специфікацію програмного інтерфейсу для управління браузером;
- референсні реалізації цього інтерфейсу для декількох браузерів;
- набір клієнтських бібліотек для цього інтерфейсу на декількох мовах програмування.




МОЖЛИВОСТІ РОБОТИ SELENIUM WEBDRIVER

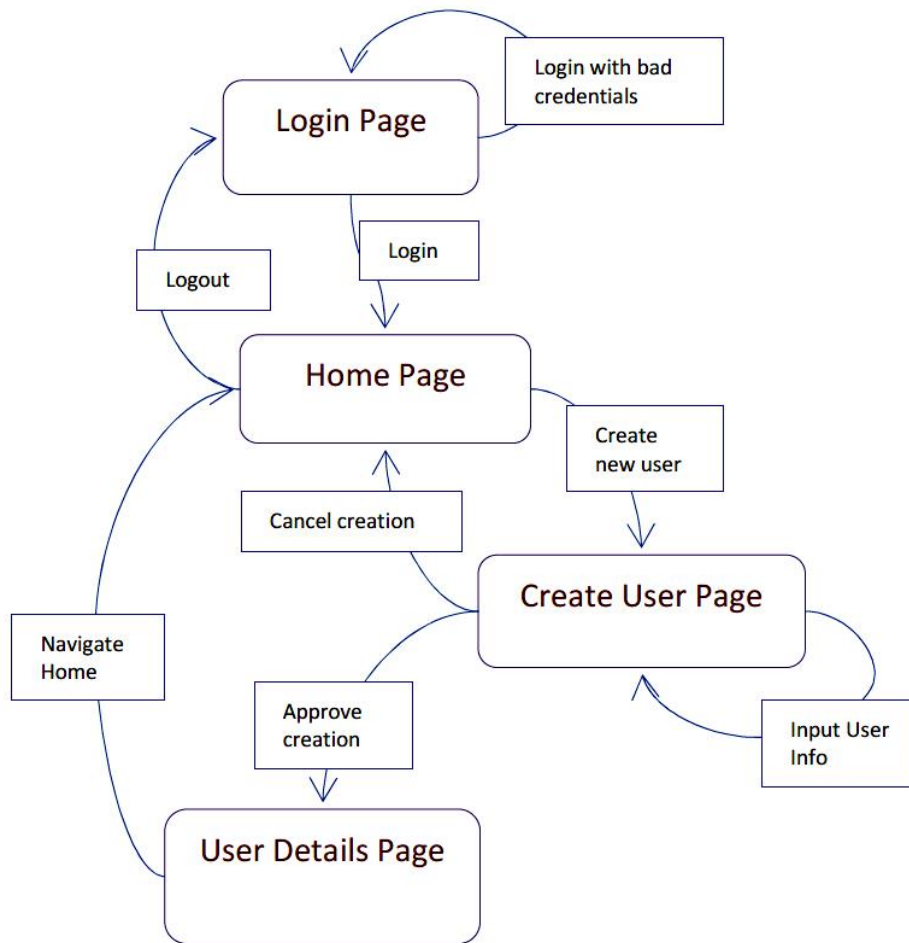
Функції, які можливо виконати з WebDriver:

- Пошук елементів в основному дереві сторінки, отримання їх атрибутів та стилів
- Взаємодія з елементами: фокусування, клік, натиснення на клавіші
- Використання різних типів очікування
- Виконання дій курсором миші (при цьому реальний курсор не використовується)
- Перехід за посиланням, перемикання між вікнами браузера, робота з cookies
- Можливість збереження скріншотів
- Завантажування та скачування файлів
- Виконання скриптів на сторінці
- Перемикання між вкладеними фреймами

Функції, які не виконуються з WebDriver:

- Власне тестування
 - Складання звітів
 - Робота з рівнями подій ОС, а також з діалоговими вікнами браузера (відправка на друк)
 - Клацання по невидимому елементу
 - Перехвачування запитів
- 

ФУНКЦІОНАЛЬНІ КОМПОЗИЦІЙНІ ПАТЕРНИ



Термін патерн дизайну вперше був визначений архітектором Крістофером Олександром. "Кожен патерн описує проблему, яка виникає знову і знову в нашому середовищі, а потім описує основні рішення цієї проблеми, таким чином, що ви можете використовувати це рішення мільйон разів, ніколи не вирішуючи ті ж самі задачі".

Дана група шаблонів описує як можна виділяти частини коду в універсальні одиночні компоненти, призначені для багаторазового використання.

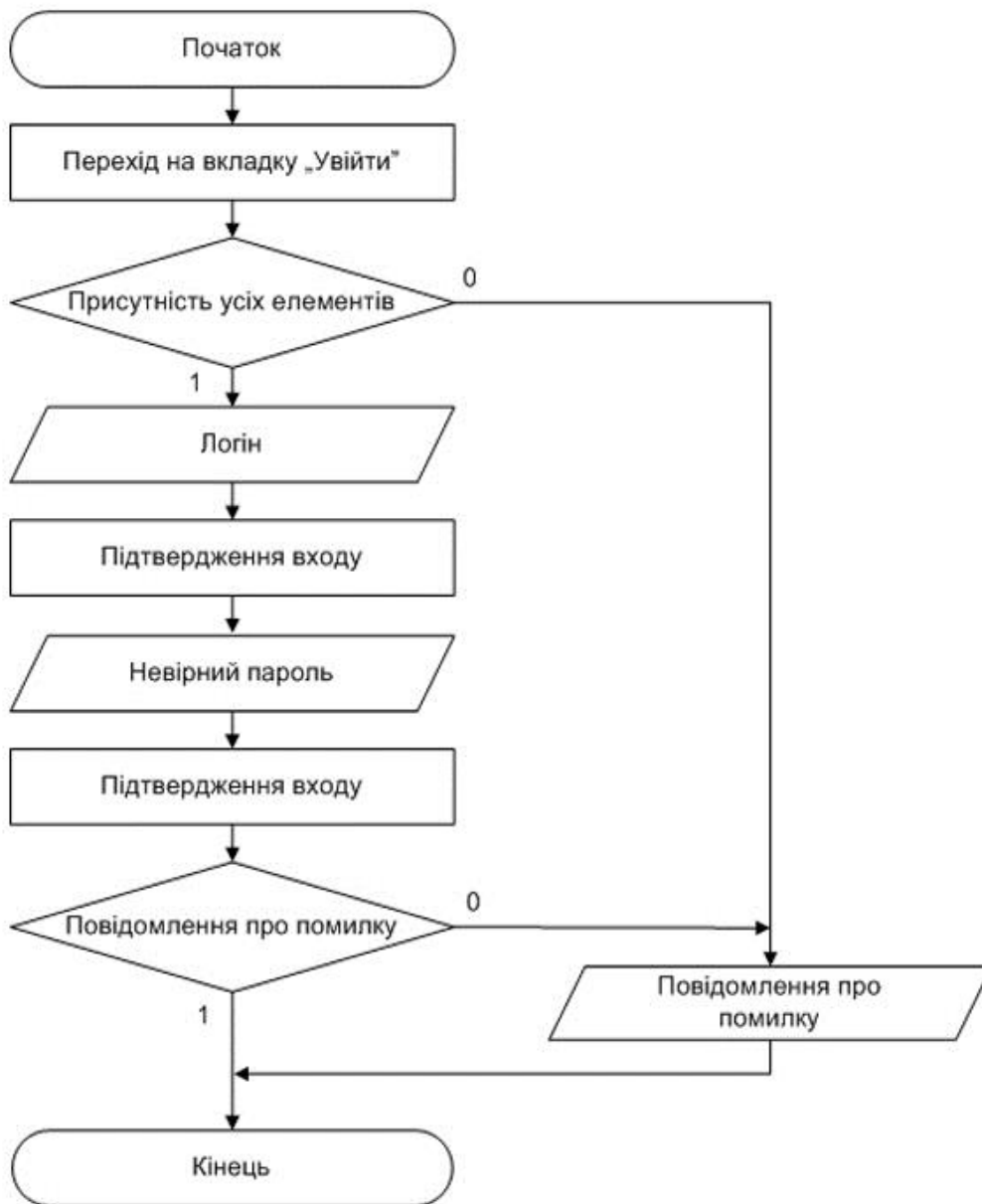


СХЕМА ПРОГРАМИ ТЕСТОВОГО СЦЕНАРІЮ «ВВЕДЕННЯ НЕКОРЕКТНОГО ЗНАЧЕННЯ ПАРОЛЮ»



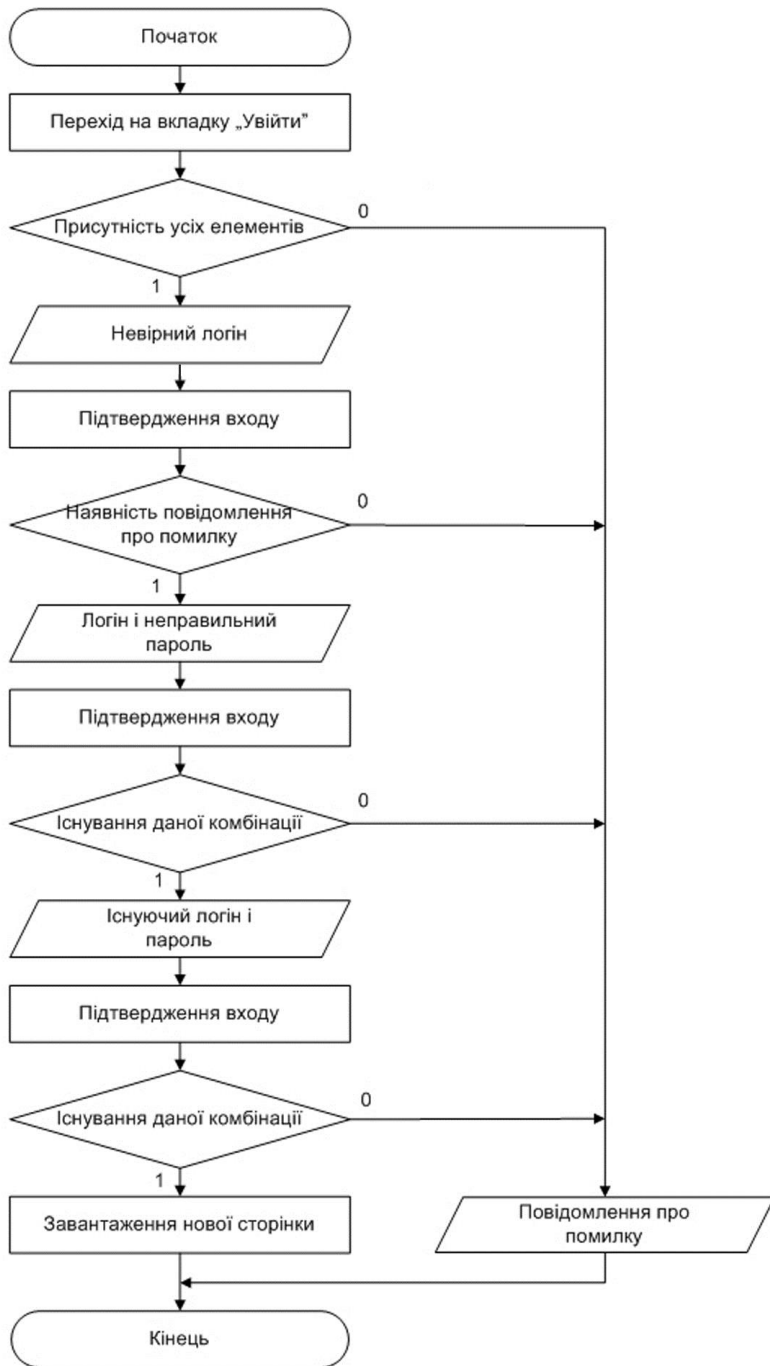


СХЕМА АЛГОРИТМУ МЕТОДУ LOGIN()



РЕЗУЛЬТАТИ ВІДТВОРЕННЯ СТВОРЕНОГО ТЕСТОВОГО НАБОРУ

Java - bakalavr_new/src/test_FireFow/TestLoginF.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window

Package Explorer JUnit

Finished after 235,217 seconds

Runs: 7/7 Errors: 0 Failures: 0

- test_FireFow.TestAnotherProblem [Runner: JUnit 4] (37,707 s)
 - testAnotherProblem (37,707 s)
- test_FireFow.TestLoginF [Runner: JUnit 4] (27,518 s)
 - testLoginF (27,518 s)
- test_FireFow.TestLogIn [Runner: JUnit 4] (24,855 s)
 - testLogIn (24,855 s)
- test_FireFow.TestVidnovlImia [Runner: JUnit 4] (39,172 s)
 - testVidnovlImia (39,172 s)
- test_FireFow.TestLogInLogOut [Runner: JUnit 4] (40,404 s)
 - testLogInLogOut (40,404 s)
- test_FireFow.TestPasswordF [Runner: JUnit 4] (36,947 s)
 - testPasswordF (36,947 s)
- test_FireFow.TestVidnowlParol [Runner: JUnit 4] (28,567 s)
 - testVidnowlParol (28,567 s)

Failure Trace

Java - bakalavr_new/src/test_Chrome/TestAnotherProblem.java - Eclipse SD

File Edit Source Refactor Navigate Search Project Run Window

Package Explorer JUnit

Finished after 75,366 seconds

Runs: 7/7 Errors: 1 Failures: 4

- test_Chrome.TestPasswordF [Runner: JUnit 4] (10,012 s)
- test_Chrome.TestVidnowlParol [Runner: JUnit 4] (9,120 s)
- test_Chrome.TestLoginF [Runner: JUnit 4] (7,461 s)
- test_Chrome.TestLogIn [Runner: JUnit 4] (8,735 s)
- test_Chrome.TestVidnovlImia [Runner: JUnit 4] (10,102 s)
- test_Chrome.TestLogInLogOut [Runner: JUnit 4] (21,670 s)
- test_Chrome.TestAnotherProblem [Runner: JUnit 4] (8,235 s)

Failure Trace



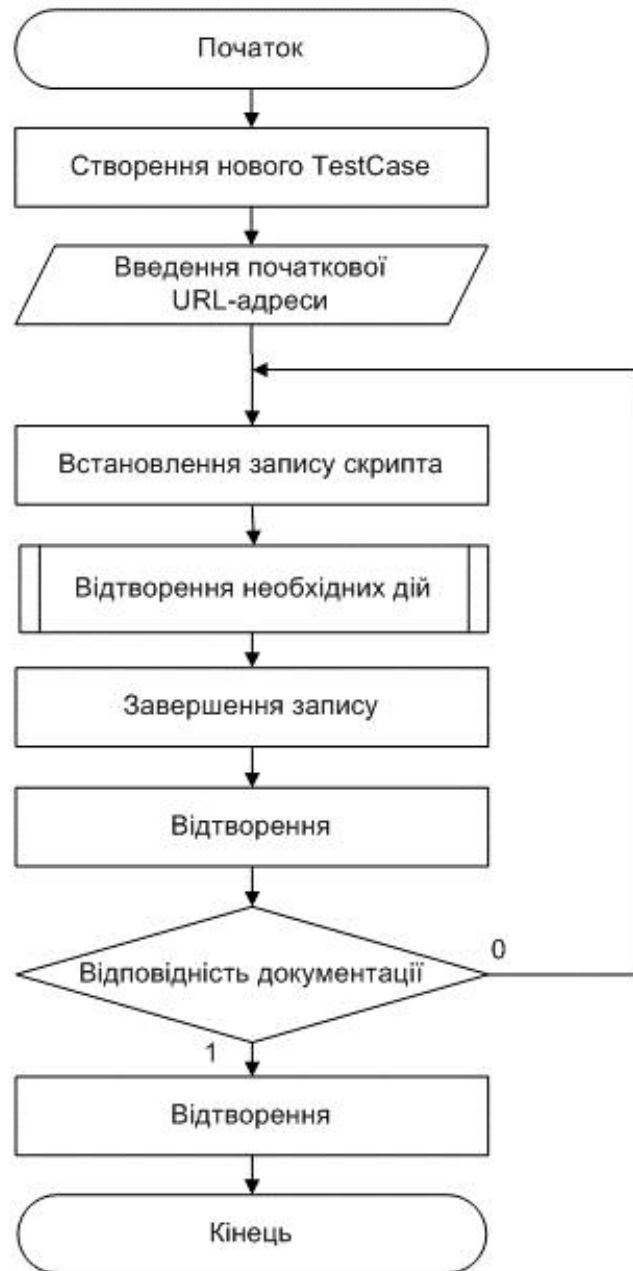


СХЕМА ПРОГРАМИ ЗАПИСУ ТЕСТОВОГО СЦЕНАРІЮ



РЕЗУЛЬТАТИ ВІДТВОРЕННЯ ЗАПИСАНОГО ТЕСТОВОГО НАБОРУ

test_logIn_logOut (tests_gmail_logIn) - Selenium IDE 2.9.0

Base URL: https://www.gmail.com/intl/uk/mail/help/about.html

Command	Target	Value
open	/intl/uk/mail/help/about....	
clickAndWait	id=gmail-sign-in	
type	id=Email	anask95@gmail.com
click	id=next	
type	id=Passwd	hj;lt;bt
clickAndWait	id=signIn	
click	css=span.gb_d.gbii	
verifyElementPresent	css=div.gb_w.gb_ba	
clickAndWait	id=gb_71	

Runs: 8
Failures: 0

Log: [info] Executing: |click | id=next | |
[info] Executing: |type | id=Passwd | hj;lt;bt |
[info] Executing: |clickAndWait | id=signIn | |
[info] Executing: |click | css=span.gb_d.gbii | |
[info] Executing: |verifyElementPresent | css=div.gb_w.gb_ba | |
[info] Executing: |clickAndWait | id=gb_71 | |
[info] Test case passed
[info] Test suite completed: 8 played, all passed!

test_logIn (tests_gmail_logIn) - Selenium IDE 2.9.0

Base URL: https://www.gmail.com/intl/ru/mail/help/about.html

Command	Target	Value
open	/intl/ru/mail/help/about.html	
clickAndWait	id=gmail-sign-in	
type	id=Email	anask95@gmail.com
click	id=next	
type	id=Passwd	hj;lt;bt
clickAndWait	id=signIn	
click	css=span.gb_d.gbii	
verifyElementPresent	css=div.gb_w.gb_ba	
clickAndWait	id=gb_71	

Runs: 1
Failures: 1

Log: [info] Executing: |clickAndWait | id=signIn | |
[info] Test case passed
[info] Playing test case test_logIn
[info] Executing: |open | /intl/ru/mail/help/about.html | |
[info] Executing: |clickAndWait | id=gmail-sign-in | |
[info] Executing: |type | id=Email | anask95@gmail.com | |
[info] Executing: |click | id=next | |
[error] Element id=next not found
[info] Test case failed



РЕЗУЛЬТАТ ВИКОНАННЯ ТЕСТОВОГО НАБОРУ СТОРІНКИ PAGELOGIN

```
[info] Playing test case test_logIn
[info] Executing: |open | /intl/ru/mail/help/about.html |
|
[info] Executing: |clickAndWait | id=gmail-sign-in | |
[info] Executing: |type | id=Email |
anask95@gmail.com |
[info] Executing: |click | id=next | |
[info] Executing: |type | id=Passwd | hj;lt;ytb |
[info] Executing: |clickAndWait | id=signIn | |
[info] Test case passed
[info] Playing test case test_logOut
[info] Executing: |click | css=span.gb_d.gbii | |
[info] Executing: |verifyElementPresent |
css=div.gb_w.gb_ba | |
[info] Executing: |clickAndWait | id=gb_71 | |
[info] Executing: |verifyElementPresent |
//div[2]/div[2]/div | |
[info] Test case passed
[info] Playing test case test_login_f
[info] Executing: |open | /intl/ru/mail/help/about.html |
|
[info] Executing: |clickAndWait | id=gmail-sign-in | |
[info] Executing: |type | id=Email | nepravulnuiLogin |
[info] Executing: |click | id=next | |
[info] Executing: |verifyText | id=errorMsg_0_Email |
Не удалось распознать адрес электронной почты. |
[info] Test case passed
[info] Playing test case test_password_f
[info] Executing: |open | /intl/ru/mail/help/about.html |
|
[info] Executing: |clickAndWait | id=gmail-sign-in | |
[info] Executing: |type | id=Email |
anask95@gmail.com |
[info] Executing: |click | id=next | |
[info] Executing: |type | id=Passwd | asdasdasd |
[info] Executing: |clickAndWait | id=signIn | |
[info] Executing: |verifyText | id=errorMsg_0_Passwd |
Указан неправильный адрес или пароль |
[info] Test case passed
```

```
[info] Playing test case test_vidnowlParol
[info] Executing: |open | /intl/ru/mail/help/about.html | |
[info] Executing: |clickAndWait | id=gmail-sign-in | |
[info] Executing: |clickAndWait | link=exact:Нужна помощь? | |
[info] Executing: |click | id=1 | |
[info] Executing: |verifyText | css=div.secondary | Чтобы сбросить пароль, введите адрес,
с помощью которого вы входите в свой аккаунт. Это может быть адрес Gmail или другой
электронной почты, а также адрес, который вы используете для работы в Google Apps. |
[info] Executing: |type | name=Email | anask95@gmail.com |
[info] Executing: |clickAndWait | //input[@value='Продолжить'] | |
[info] Test case passed
[info] Playing test case test_vidnovl_Imia
[info] Executing: |open | /intl/ru/mail/help/about.html | |
[info] Executing: |clickAndWait | id=gmail-sign-in | |
[info] Executing: |clickAndWait | link=exact:Нужна помощь? | |
[info] Executing: |click | id=2 | |
[info] Executing: |clickAndWait | //input[@value='Продолжить'] | |
[info] Executing: |click | id=1 | |
[info] Executing: |verifyElementPresent | id=fu_email | |
[info] Executing: |verifyText | css=p.disabledtext | Подсказка. Вы могли добавить этот
адрес эл. почты в свой аккаунт при регистрации в Gmail. |
[info] Test case passed
[info] Playing test case test_another_problem
[info] Executing: |open | /intl/ru/mail/help/about.html | |
[info] Executing: |clickAndWait | id=gmail-sign-in | |
[info] Executing: |clickAndWait | link=exact:Нужна помощь? | |
[info] Executing: |click | id=3 | |
[info] Executing: |verifyText | css=#hideable-box3 > div.hideable-box > div.secondary |
Введите имя пользователя для входа в аккаунт Google. Это может быть либо адрес
Gmail либо другой адрес электронной почты, связанный с вашим аккаунтом. |
[info] Executing: |verifyElementPresent | name=Email2 | |
[info] Test case passed
[info] Test suite completed: 8 played, all passed!
```



Створений скрипт на мові Java

The screenshot displays the Eclipse IDE interface with a Java Selenium test script. The Package Explorer on the left shows the project structure: `bakalavr_new` with sub-packages `test_Chrome`, `test_FireFow`, and `JUnit`. The main editor shows the code for `TestLogin.java` in the `test_FireFow` package. The code uses Selenium WebDriver to interact with a Gmail login page. The `setUp()` method initializes the driver and sets the base URL. The `testLogin()` method performs the login steps: clicking 'sign-in', clearing the email field, entering the email, clearing the password field, and entering the password.

```
package test_FireFow;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class TestLogin {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

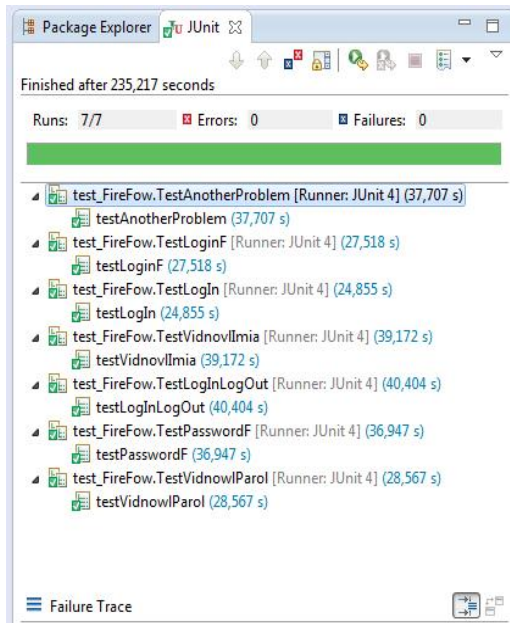
    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "https://www.gmail.com/intl/ru/mail/help/about.html";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testLogin() throws Exception {
        driver.get(baseUrl);
        driver.findElement(By.id("gmail-sign-in")).click();
        driver.findElement(By.id("Email")).clear();
        driver.findElement(By.id("Email")).sendKeys("anask95@gmail.com");
        driver.findElement(By.id("next")).click();
        driver.findElement(By.id("Passwd")).clear();
        driver.findElement(By.id("Passwd")).sendKeys("hj;ltybt");
        driver.findElement(By.id("signIn")).click();
    }
}
```

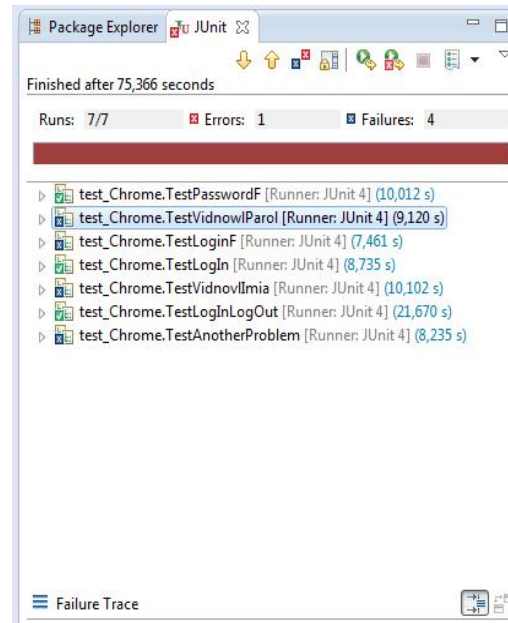
The Problems view at the bottom shows 15 warnings related to unused imports:

Description	Resource	Path	Location	Type
Warnings (15 items)				
Build path specifies execution environment J...	bakalavr_new		Build path	JRE System Li...
The import java.util.regex.Pattern is never use...	TestGoogle.java	/bakalavr_new/src/...	line 4	Java Problem
The import java.util.regex.Pattern is never use...	TestLogin.java	/bakalavr_new/src/...	line 3	Java Problem
The import org.hamcrest.CoreMatchers is ne...	TestGoogle.java	/bakalavr_new/src/...	line 8	Java Problem
The import org.hamcrest.CoreMatchers is ne...	TestLogin.java	/bakalavr_new/src/...	line 7	Java Problem





Результат тестування – без помилки



Результат тестування – з помилкою



ВИСНОВОК

В дані роботі були розглянуті основні поняття тестування взагалі та web-тестування зокрема, наведена найбільш поширена класифікація тестування, розглянуті глобальні рівні тестування та їх особливості.

Були виділені основні етапи тестування web-сайту та сторінок: функціональне тестування, тестування верстки, безпеки сайту, валідності та продуктивності коду, тестування на дружність, на сумісність з різними браузерами і ОС.

Також у роботі проведено класифікацію дизайн патернів функціонального автоматизованого тестування за їх призначенням. Було виділено такі патерни: імплементації тестів, архітектурні, функціональні композиційні, процесуальні та патерни групування тестів. Визначено, що рушійною силою побудови тестувального рішення є вибір конкретного патерну імплементації тестів. Головна увага приділялась функціональним композиційним патернам, тому що на основі них, проводиться подальша розробка скриптів. Було згруповано базові методи відповідно до сторінок, які їх представляють: сторінка PageLogin з методом LogIn(); сторінка CreateUserPage з основним методом Create NewUser(); сторінка PageHome з методом HomePage().

Розроблено набір тестових сценаріїв (скриптів), відповідно до основних методів, за допомогою яких можливе скорочення витрат часу на процес повного тестування продукту. Скрипти придатні для тестування web-сторінок будь-якого змісту з подібною логічною структурою сторінки входу у акаунт.

ДЯКУЮ ЗА ВАШУ УВАГУ!

