МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

Лабораторная работа №2

по дисциплине

Тестирование ПО

(наименование дисциплины)

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Скорынин С. С.

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Саечников Н.И.

(подпись) (фамилия, и.,о.)

С22-СИБ

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2025

**1. Цель работы**

Изучить средства тестирования в Java, разработать набор unit-тестов для сайта Selenium.

**2. Постановка задачи**

Разработать набор unit-тестов для тестирования web- сайта Selenium с использованием Selenium Web-driver (не менее 10).

План тестировани:

1. тест меню навигации
2. проверка выплывающего меню about
3. проверка ссылок для скачивания драйвера для java
4. проверка на переключение языков
5. проверка соц икнонок в футере
6. тест заголовка
7. тест навигации
8. тест видимости элементов
9. тест заполнения поля
10. тест клика по кнопке

**3. Теория:**

**1. Автоматизированное тестирование:**

* **Определение:** Автоматизированное тестирование — это процесс проверки программного обеспечения с использованием специальных инструментов и скриптов, которые автоматизируют выполнение тестовых сценариев. Вместо ручного выполнения тестов, автоматизированные тесты запускаются автоматически и проверяют результаты на соответствие заданным критериям.
* **Цели автоматизированного тестирования:**
  + **Увеличение скорости тестирования:** Автоматизированные тесты выполняются гораздо быстрее, чем ручные тесты, что позволяет быстрее выявлять ошибки.
  + **Повышение точности тестирования:** Автоматизированные тесты выполняются всегда одинаково, исключая человеческий фактор и обеспечивая более точные результаты.
  + **Снижение затрат на тестирование:** Автоматизация позволяет сократить количество ручного труда, что снижает затраты на тестирование в долгосрочной перспективе.
  + **Обеспечение возможности регрессионного тестирования:** Автоматизированные тесты легко запускать повторно после внесения изменений в код, что позволяет быстро выявлять регрессионные ошибки (ошибки, которые возникают в ранее работавшем коде).
  + **Улучшение покрытия кода тестами:** Автоматизированные тесты позволяют охватить большее количество сценариев и вариантов использования, что улучшает покрытие кода тестами.
* **Уровни автоматизированного тестирования (применительно к веб-приложениям):**
  + **Unit-тестирование:** Проверка отдельных компонентов или модулей веб-приложения (например, функций, классов). В контексте Selenium, unit-тесты могут быть направлены на проверку отдельных JavaScript-компонентов.
  + **Интеграционное тестирование:** Проверка взаимодействия между различными компонентами веб-приложения.
  + **Системное тестирование:** Проверка веб-приложения как единого целого, включая все компоненты и интеграции. Selenium чаще всего используется именно для системного тестирования веб-приложений.
  + **Приемочное тестирование:** Проверка веб-приложения на соответствие требованиям заказчика.

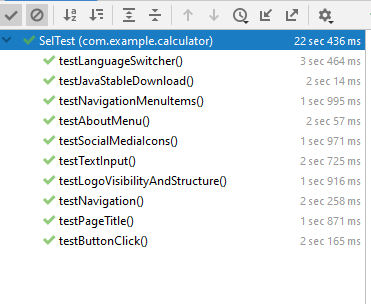
**2. Selenium WebDriver:**

* **Определение:** Selenium WebDriver — это фреймворк для автоматизированного тестирования веб-приложений. Он позволяет управлять браузером программно, имитируя действия пользователя (например, нажатие на кнопки, ввод текста, переходы по ссылкам).
* **Архитектура Selenium WebDriver:**
  + **Selenium Client Libraries (языковые привязки):** Selenium предоставляет библиотеки для различных языков программирования (Java, Python, C#, JavaScript и др.), которые позволяют писать тесты на этих языках.
  + **WebDriver:** WebDriver — это интерфейс, определяющий общий набор методов для взаимодействия с браузером.
  + **Browser Drivers (драйверы браузеров):** Драйверы браузеров — это отдельные исполняемые файлы, которые управляют конкретными браузерами (Chrome, Firefox, Edge, Safari и др.). WebDriver использует драйверы для отправки команд в браузер и получения результатов.

**4. Реализация**

**package** com.example.calculator;  
  
**import** io.github.bonigarcia.wdm.WebDriverManager;  
**import** org.apache.commons.lang3.exception.ExceptionUtils;  
**import** org.junit.jupiter.api.\*;  
**import** org.openqa.selenium.\*;  
**import** org.openqa.selenium.chrome.ChromeDriver;  
**import** org.openqa.selenium.chrome.ChromeOptions;  
**import** org.openqa.selenium.support.ui.ExpectedConditions;  
**import** org.openqa.selenium.support.ui.WebDriverWait;  
  
**import** java.time.Duration;  
**import** java.util.List;  
**import** java.util.Map;  
  
**import static** org.junit.jupiter.api.Assertions.\*;  
  
**public class** SelTest {  
 **private** WebDriver **driver**;  
 **private** WebDriverWait **wait**;  
  
 @BeforeEach  
 **public void** setup() {  
 WebDriverManager.*chromedriver*().setup();  
  
 ChromeOptions options = **new** ChromeOptions();  
 options.addArguments(**"--remote-allow-origins=\*"**);  
 options.addArguments(**"--start-maximized"**);  
 options.addArguments(**"--disable-blink-features=AutomationControlled"**);  
  
 **driver** = **new** ChromeDriver(options);  
 **wait** = **new** WebDriverWait(**driver**, Duration.*ofSeconds*(15));  
 }  
  
 @AfterEach  
 **public void** teardown() {  
 **if** (**driver** != **null**) {  
 **driver**.quit();  
 }  
 }  
  
 *// тест меню навигации* @Test  
 **public void** testNavigationMenuItems() {  
 **driver**.get(**"https://www.selenium.dev/"**);  
  
 List<WebElement> menuItems = **wait**.until(ExpectedConditions.*presenceOfAllElementsLocatedBy*(  
 By.*cssSelector*(**"ul.navbar-nav > li.nav-item"**)));  
  
 String[] expectedItems = {**"About"**, **"Downloads"**, **"Documentation"**, **"Projects"**, **"Support"**, **"Blog"**, **"English"**};  
  
 *assertEquals*(expectedItems.**length**, menuItems.size(),  
 **"Количество пунктов меню не соответствует ожидаемому"**);  
  
 **for** (**int** i = 0; i < expectedItems.**length**; i++) {  
 *assertTrue*(menuItems.get(i).getText().contains(expectedItems[i]),  
 **"Пункт меню "** + expectedItems[i] + **" не найден"**);  
 }  
 }  
  
 *// проверка выплывающего меню about* @Test  
 **public void** testAboutMenu() {  
 **driver**.get(**"https://www.selenium.dev/"**);  
 WebElement aboutMenu = **wait**.until(ExpectedConditions.*visibilityOfElementLocated*(  
 By.*xpath*(**"//li[contains(@class,'nav-item dropdown')]//a[text()='About']"**)));  
 *assertTrue*(aboutMenu.isDisplayed(), **"Пункт меню 'About' не отображается"**);  
  
 aboutMenu.click();  
  
 WebElement dropdownMenu = **wait**.until(ExpectedConditions.*visibilityOfElementLocated*(  
 By.*xpath*(**"//li[contains(@class,'nav-item dropdown')]//div[contains(@class,'dropdown-menu show')]"**)));  
 *assertTrue*(dropdownMenu.isDisplayed(), **"Выпадающее меню не отображается"**);  
  
 Map<String, String> menuUrls = Map.*of*(  
 **"About Selenium"**, **"/about"**,  
 **"Structure and Governance"**, **"/project"**,  
 **"Events"**, **"/events"**,  
 **"Ecosystem"**, **"/ecosystem"**,  
 **"History"**, **"/history"**,  
 **"Get Involved"**, **"/getinvolved"**,  
 **"Sponsors"**, **"/sponsors"**,  
 **"Sponsor Us"**, **"/sponsor"** );  
  
 **for** (Map.Entry<String, String> entry : menuUrls.entrySet()) {  
 WebElement item = dropdownMenu.findElement(  
 By.*xpath*(**".//a[text()='"** + entry.getKey() + **"']"**));  
 *assertTrue*(item.getAttribute(**"href"**).endsWith(entry.getValue()),  
 String.*format*(**"Для пункта '%s' ожидался URL '%s'"**,  
 entry.getKey(), entry.getValue()));  
 }  
 }  
 *// проверка ссылок для скачивания драйвера для java* @Test  
 **public void** testJavaStableDownload() {  
 **driver**.get(**"https://www.selenium.dev/downloads/"**);  
  
 WebElement javaCard = **wait**.until(ExpectedConditions.*visibilityOfElementLocated*(  
 By.*xpath*(**"//div[contains(@class,'card') and .//img[@alt='Java']]"**)));  
  
 WebElement javaTitle = javaCard.findElement(By.*xpath*(**".//p[contains(@class,'card-title')]"**));  
 *assertEquals*(**"Java"**, javaTitle.getText());  
  
 WebElement stableVersion = javaCard.findElement(  
 By.*xpath*(**".//a[contains(@href,'selenium-java-') and contains(@class,'card-link')]"**));  
 *assertTrue*(stableVersion.isDisplayed());  
  
 *assertTrue*(stableVersion.getText().matches(**"\\d+\\.\\d+\\.\\d+.\*"**),  
 **"Некорректный формат версии Java"**);  
  
 String downloadUrl = stableVersion.getAttribute(**"href"**);  
 *assertTrue*(downloadUrl.contains(**"selenium-java-"**) && downloadUrl.endsWith(**".zip"**),  
 **"Некорректная ссылка для скачивания"**);  
  
 WebElement changelogLink = javaCard.findElement(  
 By.*xpath*(**".//a[contains(@href,'CHANGELOG')]"**));  
 *assertTrue*(changelogLink.isDisplayed());  
  
 WebElement apiDocsLink = javaCard.findElement(  
 By.*xpath*(**".//a[contains(@href,'api/java')]"**));  
 *assertTrue*(apiDocsLink.isDisplayed());  
 }  
  
  
 *// проверка на переключение языков* @Test  
 **public void** testLanguageSwitcher() {  
 **driver**.get(**"https://www.selenium.dev/"**);  
  
 WebElement languageSwitcher = **wait**.until(ExpectedConditions.*elementToBeClickable*(  
 By.*xpath*(**"//li[contains(@class, 'dropdown')]//a[contains(., 'English')]"**)));  
 languageSwitcher.click();  
  
 **wait**.until(ExpectedConditions.*visibilityOfElementLocated*(  
 By.*cssSelector*(**"ul.dropdown-menu.show"**)));  
  
 WebElement japaneseOption = **wait**.until(ExpectedConditions.*elementToBeClickable*(  
 By.*xpath*(**"//ul[@class='dropdown-menu show']//a[contains(@href,'/ja/')]"**)));  
 japaneseOption.click();  
  
 **wait**.until(ExpectedConditions.*urlContains*(**"/ja/"**));  
 }  
  
  
 *// проверка соц икнонок в футере* @Test  
 **public void** testSocialMediaIcons() {  
 **driver**.get(**"https://www.selenium.dev/"**);  
  
 List<WebElement> socialIcons = **wait**.until(ExpectedConditions.*presenceOfAllElementsLocatedBy*(  
 By.*cssSelector*(**"footer .col-6.col-sm-4.text-xs-center.order-sm-2 .list-inline-item"**)));  
  
 String[] expectedSocialLinks = {  
 **"linkedin.com/company/4826427"**,  
 **"x.com/SeleniumHQ"**,  
 **"youtube.com/@SeleniumHQProject"**,  
 **"mastodon.social/@seleniumHQ"**,  
 **"bsky.app/profile"**,  
 **"groups.google.com/group/selenium-users"**,  
 **"youtube.com/channel/UCbDlgX\_613xNMrDqCe3QNEw"** };  
  
 *assertEquals*(expectedSocialLinks.**length**, socialIcons.size(),  
 **"Количество социальных иконок не соответствует ожидаемому"**);  
  
 **for** (**int** i = 0; i < socialIcons.size(); i++) {  
 WebElement icon = socialIcons.get(i);  
 *assertTrue*(icon.isDisplayed(), **"Иконка #"** + (i+1) + **" не отображается"**);  
 *assertTrue*(icon.findElement(By.*tagName*(**"a"**)).getAttribute(**"href"**).contains(expectedSocialLinks[i]),  
 **"Ссылка иконки #"** + (i+1) + **" не соответствует ожидаемой"**);  
 }  
 }  
  
  
  
 *// тест заголовка* @Test  
 **public void** testPageTitle() {  
 **driver**.get(**"https://www.selenium.dev/"**);  
 **wait**.until(ExpectedConditions.*titleContains*(**"Selenium"**));  
 *assertTrue*(**driver**.getTitle().contains(**"Selenium"**));  
 }  
  
 *// тест навигации* @Test  
 **public void** testNavigation() {  
 **driver**.get(**"https://www.selenium.dev/"**);  
  
 WebElement docsLink = **wait**.until(ExpectedConditions.*elementToBeClickable*(  
 By.*xpath*(**"//nav//div//ul//li//a//span[contains(text(), 'Documentation')]"**)));  
 docsLink.click();  
 **wait**.until(ExpectedConditions.*urlContains*(**"documentation"**));  
 *assertTrue*(**driver**.getCurrentUrl().contains(**"https://www.selenium.dev/documentation/"**));  
 }  
  
 *// тест видимости элементов* @Test  
 **public void** testLogoVisibilityAndStructure() {  
 **driver**.get(**"https://www.selenium.dev/"**);  
  
 WebElement logo = **wait**.until(ExpectedConditions.*visibilityOfElementLocated*(  
 By.*xpath*(**"//a[contains(@class, 'navbar-brand')]"**)));  
 *assertTrue*(logo.isDisplayed(), **"Логотип не отображается"**);  
  
 WebElement svg = logo.findElement(By.*xpath*(**".//\*[local-name()='svg']"**));  
  
 WebElement svgTitle = svg.findElement(By.*xpath*(**".//\*[local-name()='title']"**));  
 *assertEquals*(**"Selenium logo green"**, svgTitle.getText(),  
 **"Некорректный title у SVG логотипа"**);  
  
 List<WebElement> paths = svg.findElements(By.*xpath*(**".//\*[local-name()='path']"**));  
 *assertTrue*(paths.size() >= 10, **"В логотипе должно быть не менее 10 path-элементов"**);  
 }  
  
 *// тест заполнения поля* @Test  
 **public void** testTextInput() {  
 **driver**.get(**"https://www.selenium.dev/documentation/"**);  
  
 WebElement searchButton = **wait**.until(ExpectedConditions.*elementToBeClickable*(  
 By.*cssSelector*(**"button.DocSearch-Button"**)));  
  
 ((JavascriptExecutor)**driver**).executeScript(**"arguments[0].click();"**, searchButton);  
  
 WebElement searchInput = **wait**.until(ExpectedConditions.*visibilityOfElementLocated*(  
 By.*cssSelector*(**"input.DocSearch-Input"**)));  
  
 String searchText = **"WebDriver"**;  
 searchInput.sendKeys(searchText);  
  
 **try** { Thread.*sleep*(500); } **catch** (InterruptedException e) {  
 System.***out***.println(ExceptionUtils.*getMessage*(e));  
  
 }  
  
 searchInput.sendKeys(Keys.***ENTER***);  
  
 **wait**.until(ExpectedConditions.*or*(  
 ExpectedConditions.*urlContains*(**"webdriver"**),  
 ExpectedConditions.*presenceOfElementLocated*(By.*cssSelector*(**"div.search-results"**))  
 ));  
 }  
  
 *// тест клика по кнопке* @Test  
 **public void** testButtonClick() {  
 **driver**.get(**"https://www.selenium.dev/"**);  
  
 WebElement aboutBtn = **wait**.until(ExpectedConditions.*elementToBeClickable*(  
 By.*xpath*(**"//footer//a[contains(text(), 'About Selenium')]"**)));  
  
 ((JavascriptExecutor)**driver**).executeScript(**"arguments[0].click();"**, aboutBtn);  
  
 **wait**.until(ExpectedConditions.*urlContains*(**"about"**));  
 *assertTrue*(**driver**.getCurrentUrl().contains(**"about"**),  
 **"Не удалось перейти на страницу информации"**);  
 }  
}

Результаты тестов:



**Вывод:**

Были изучены средства тестирования в Java для web приложений, разработаны модульные тесты для отдельных объектов сайта, было выполнено 10 тестов.