

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра ИС

ОТЧЕТ
по практической работе № 4
по дисциплине «Автоматизация тестирования»
Тема: «Тестирование с помощью Selenium webdriver и Python»

Студент гр. 2372

Соколовский В.Д.

Преподаватель

Турнецкая Е.Л.

Санкт-Петербург

2024

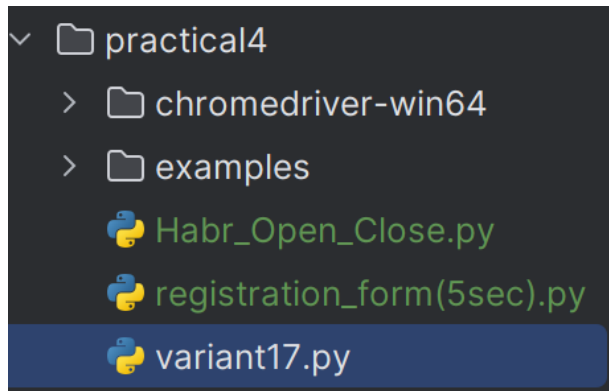
Цель работы: Получение практических навыков по тестированию веб-элементов с помощью Selenium Webdriver и Python.

Поставленные задачи:

1. Установить программное окружение проекта по автоматизированному тестированию.
2. Реализовать проект по запуску браузера под управлением Selenium и отобразить в нем страницу веб-системы.
3. Разработать проект по нахождению веб-элементов на учебном ресурсе с помощью программных методов Selenium.
4. Создать проект по заполнению полей формы регистрации с использованием явных и неявных ожиданий Selenium.
5. Зафиксировать результаты тестирования в отчете.

Выполнение работы:

Создание проекта:



Установленные компоненты программного окружения:

pytest	8.1.1
selenium	4.19.0

Проект по автоматическому открытию самостоятельно выбранного веб-приложения в браузере под управлением Selenium.

Листинг 1 – Код программы в Habr_Open_Close.py

```
from selenium import webdriver
import time
from selenium.webdriver.chrome.service import Service as ChromeService

# Путь к исполняемому файлу chromedriver.exe
chrome_driver_path = 'TestAutomation/practical4/chromedriver-
win64/chromedriver.exe'

# Создание сервиса Chrome
chrome_service = ChromeService(executable_path=chrome_driver_path)

# Создание экземпляра браузера
driver = webdriver.Chrome(service=chrome_service)

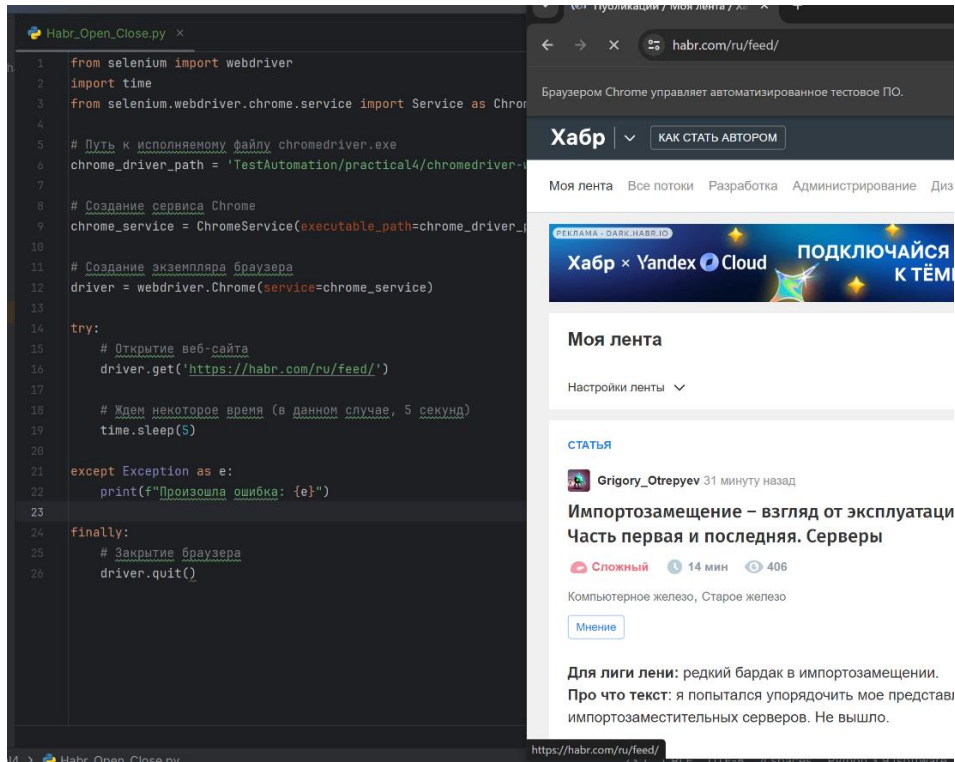
try:
    # Открытие веб-сайта
    driver.get('https://habr.com/ru/feed/')

    # Ждем некоторое время (в данном случае, 5 секунд)
    time.sleep(5)

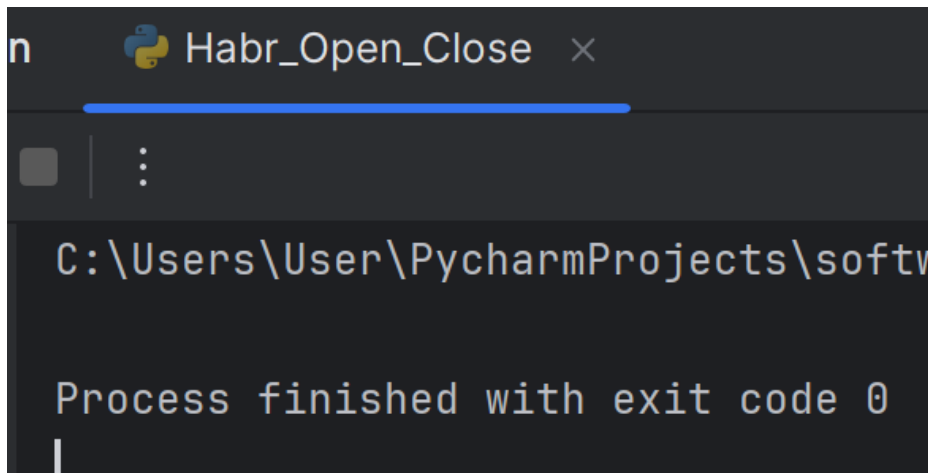
except Exception as e:
    print(f"Произошла ошибка: {e}")

finally:
    # Закрытие браузера
    driver.quit()
```

Запуск кода:



Завершение:



Проект по тестированию наличия шести предметных карточек на веб-странице индивидуального задания.

Листинг 2 – код программы в variant17.py

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service as ChromeService

# Путь к исполняемому файлу chromedriver.exe
chrome_driver_path = 'TestAutomation/practical4/chromedriver-
win64/chromedriver.exe'

# Создание сервиса Chrome
chrome_service = ChromeService(executable_path=chrome_driver_path)

# Создание экземпляра браузера
browser = webdriver.Chrome(service=chrome_service)

# Адрес сайта
URL = " https://qa-test-selectors.netlify.app/"

# Номер варианта
VARIANT = 17
try:
    # Открытие веб-страницы
    browser.get(URL)

    # Неявное ожидание для загрузки элементов страницы
    browser.implicitly_wait(10)

    # Выбор кнопки с 17 вариантом, по CSS-селектору
    button = browser.find_element(By.CSS_SELECTOR, f'.variant__btn:nth-
child({VARIANT})')

    # Нажатие на кнопку с вариантом
    button.click()

    # Элементы с тегом h1 (заголовки)
    elements_by_tag = browser.find_elements(By.TAG_NAME, "h1")
    print("Элементы с тегом h1:", len(elements_by_tag))

    # Элементы с именем birch-tree
    elements_by_name = browser.find_elements(By.NAME, "cool-bang")
    print("Элементы с именем cool-bang:", len(elements_by_name))

    # Элементы с классом imageContainer
    elements_by_class = browser.find_elements(By.CLASS_NAME, "imageContainer")
    print("Элементы с классом imageContainer:", len(elements_by_class))

    # Элементы с ID maple
    elements_by_id = browser.find_elements(By.ID, "goldy")
    print("Элементы с ID goldy:", len(elements_by_id))

    # Элементы с data-type="trees"
    elements_by_selector = browser.find_elements(By.CSS_SELECTOR, '[data-
type="penguins"]')
    print("Элементы с data-type penguins:", len(elements_by_selector))

finally:
    # закрываем браузер после всех манипуляций
    browser.quit()
```

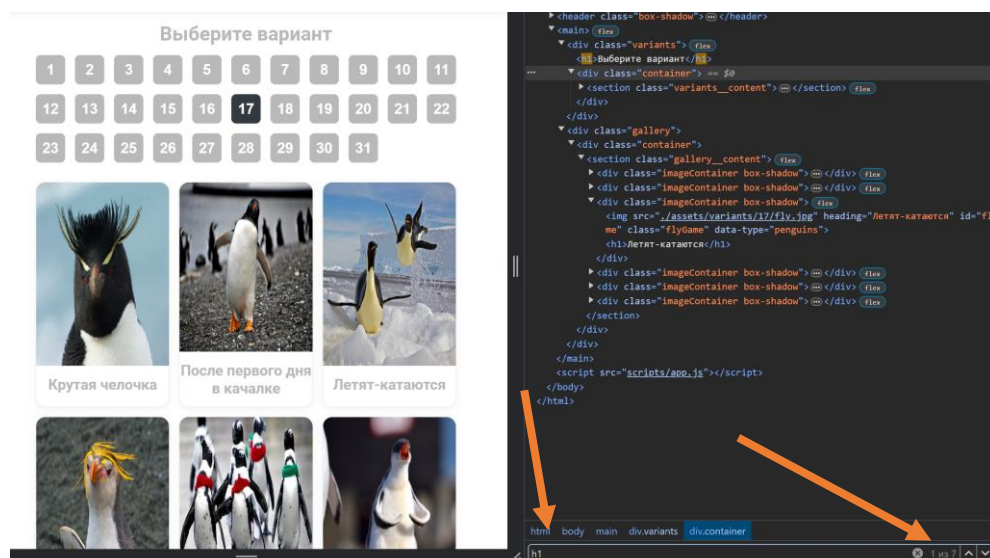
Запуск кода:

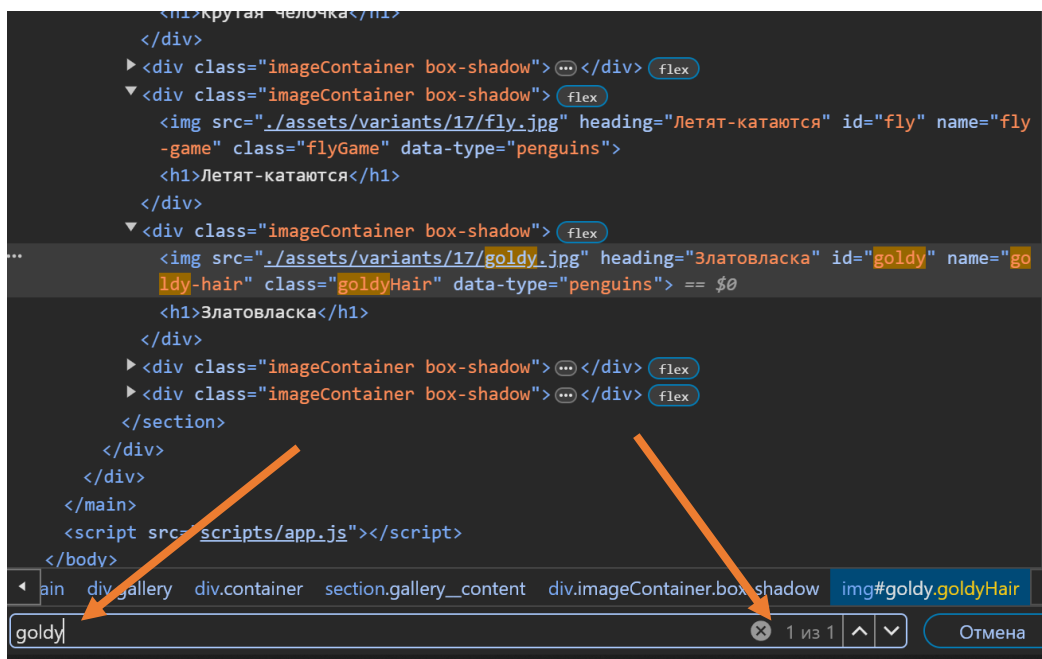
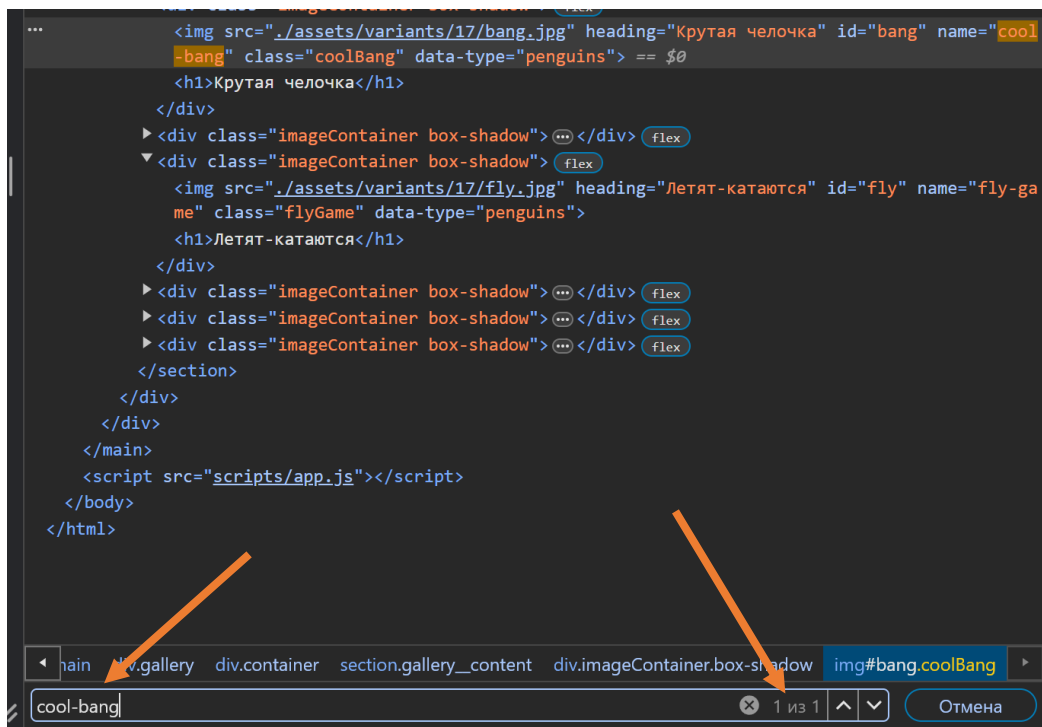
```
variant17.py x
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3 from selenium.webdriver.chrome.service import Service as ChromeService
4
5 # Путь к исполняемому файлу chromedriver.exe
6 chrome_driver_path = 'TestAutomation/practical4/chromedriver-win64/chromedriver.exe'
7
8 # Создание сервиса Chrome
9 chrome_service = ChromeService(executable_path=chrome_driver_path)
10
11 # Создание экземпляра браузера
12 browser = webdriver.Chrome(service=chrome_service)
13
14 # Адрес сайта
15 URL = "https://qa-test-selectors.netlify.app/"
16
```

Run variant17 x

C:\Users\User\PycharmProjects\software_testing\venv\Scripts\python.exe C:\Users\User\Py
Элементы с тегом h1: 7
Элементы с именем cool-bang: 1
Элементы с классом imageContainer: 6
Элементы с ID goldy: 1
Элементы с data-type penguins: 6
Process finished with exit code 0

Проверка, найденных автотестом результатов, через DevTools:





Проект по автозаполнению формы регистрации.

Листинг 3 – код программы registration_form(5sec).py

```
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service as ChromeService

# Путь к исполняемому файлу chromedriver.exe
chrome_driver_path = 'TestAutomation/practical4/chromedriver-
win64/chromedriver.exe'

# Создание сервиса Chrome
chrome_service = ChromeService(executable_path=chrome_driver_path)

# Создание экземпляра браузера
browser = webdriver.Chrome(service=chrome_service)
URL = "https://qa-course.netlify.app/registration-form-timer"

try:
    # Открытие веб-страницы
    browser.get(URL)

    # Неявное ожидание для загрузки элементов страницы
    browser.implicitly_wait(5)

    # Выбор элемента по имени "firstName"
    input1 = browser.find_element(By.NAME, "firstName")
    input1.send_keys("Ivan")

    # Выбор элемента по имени "lastName"
    input2 = browser.find_element(By.NAME, "lastName")
    input2.send_keys("Petrov")

    # Выбор элемента по имени "city"
    input3 = browser.find_element(By.NAME, "city")
    input3.send_keys("Moscow")

    # Выбор элемента по имени "email"
    input4 = browser.find_element(By.NAME, "email")
    input4.send_keys("mail@mail.com")

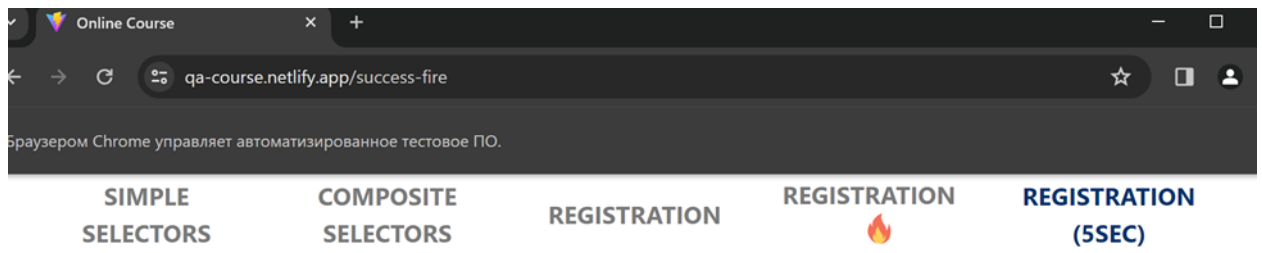
    # Вызов JS-скрипта поиска и нажатия кнопки подтверждения
    script = '''document.querySelector('button[type="submit"]').click();'''
    browser.execute_script(script)

except Exception as e:
    print(f"Произошла ошибка: {e}")

finally:
    # Задержка перед закрытием браузера
    time.sleep(5)

    # закрываем браузер после всех манипуляций
    browser.quit()
```

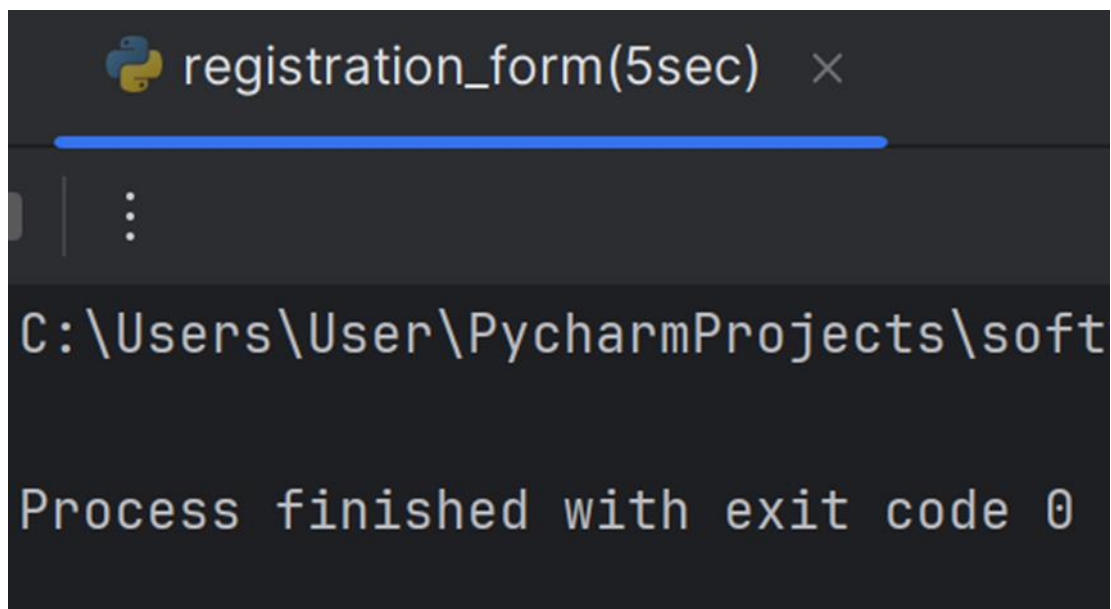

Запуск кода:



Регистрация пройдена успешно



Завершение:



Выводы:

В ходе выполнения работы по получению практических навыков по тестированию веб-элементов с использованием Selenium WebDriver и Python были достигнуты определенные результаты и приобретены ценные знания и навыки.

Первоначальная цель работы заключалась в освоении методов автоматизированного тестирования веб-элементов с помощью инструментов Selenium WebDriver и языка программирования Python. В ходе выполнения работы были изучены основные принципы работы с Selenium WebDriver, включая настройку среды PyCharm, создание и запуск автотестов, а также анализ результатов тестирования.

Одним из ключевых результатов работы стало овладение навыками написания автотестов с использованием Selenium WebDriver и Python. Были изучены основные методы и функции для взаимодействия с веб-элементами, такие как поиск элементов по различным критериям, ввод текста, клики на элементы, получение и анализ атрибутов элементов и многое другое.

В целом выполнение данной работы позволило значительно расширить кругозор в области тестирования веб-элементов и приобрести ценный практический опыт работы с инструментами автоматизации тестирования. Полученные знания и навыки будут полезны в дальнейшей работе над проектами в области разработки программного обеспечения и тестирования веб-приложений.

Список использованных источников:

1. Документация по Selenium Webdriver. URL: <https://www.selenium.dev/documentation/webdriver/>
2. Selenium WebDriver.
URL: <https://testengineer.ru/selenium-webdriver/>