Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»

Образовательная программа «Веб-технологии»

ПРАКТИЧЕСКОЕ ЗАДАНИЕ ПО ДИСЦИПЛИНЕ

 «Программная инженерия»

на тему:

 «Лабораторная работа №8. Написание Unit тестов»

**Выполнил:**

Студент группы 181-321

                                      Гусев Ф.А.

Москва 2020

## Тестирование работоспособности pop-up «Заказать звонок».

## @Test

## public void testFormOnCall() throws InterruptedException {

## driver.findElement(By.className("popup-open")).click();

## driver.findElement(By.name("name")).sendKeys("a@a.a");

## driver.findElement(By.name("phone")).sendKeys("9162641800");

## driver.findElement(By.name("agree")).click();

## Thread.sleep(1000);

## driver.findElement(By.name("submit")).click(); //Отправка формы

## Thread.sleep(1000);

## WebElement success = waitForElementPresent("success", "Cannot find element", 10);

}

## Тестирование работоспособности формы регистрации пользователя.

## @Test

## public void testRegistrationForm() throws InterruptedException {

## driver.findElement(By.id("registration")).click();

## driver.findElement(By.name("FIO")).sendKeys("Имя Фамилия");

## driver.findElement(By.name("login")).sendKeys("example@mail.ru");

## driver.findElement(By.name("pass")).sendKeys("12345");

## driver.findElement(By.name("password")).sendKeys("12345");

## driver.findElement(By.name("agree")).click();

## Thread.sleep(1000); //Визуальная пауза

## driver.findElement(By.name("button")).click();

## Thread.sleep(1000);

## WebElement success = waitForElementPresent("success", "Cannot find element", 10);

}

## Тестирование битых ссылок и кодов ответов при переходе по ссылкам.

@Test

public void testLinks() throws InterruptedException {

Map<Integer, List<String>> map = driver.findElements(By.xpath("//\*[@href]"))

.stream() // находим все элементы с атрибутом href

.map(ele -> ele.getAttribute("href")) // получаем их значение

.map(String::trim) // обрезаем конечные проберы

.distinct() // оставляем уникальные ссылки

.collect(Collectors.groupingBy(Main::getResponseCode));

map.get(200);// в скобках указывается код ответов

map.get(403);

map.get(404);

map.get(0); // содержит все неизвестные ссылки

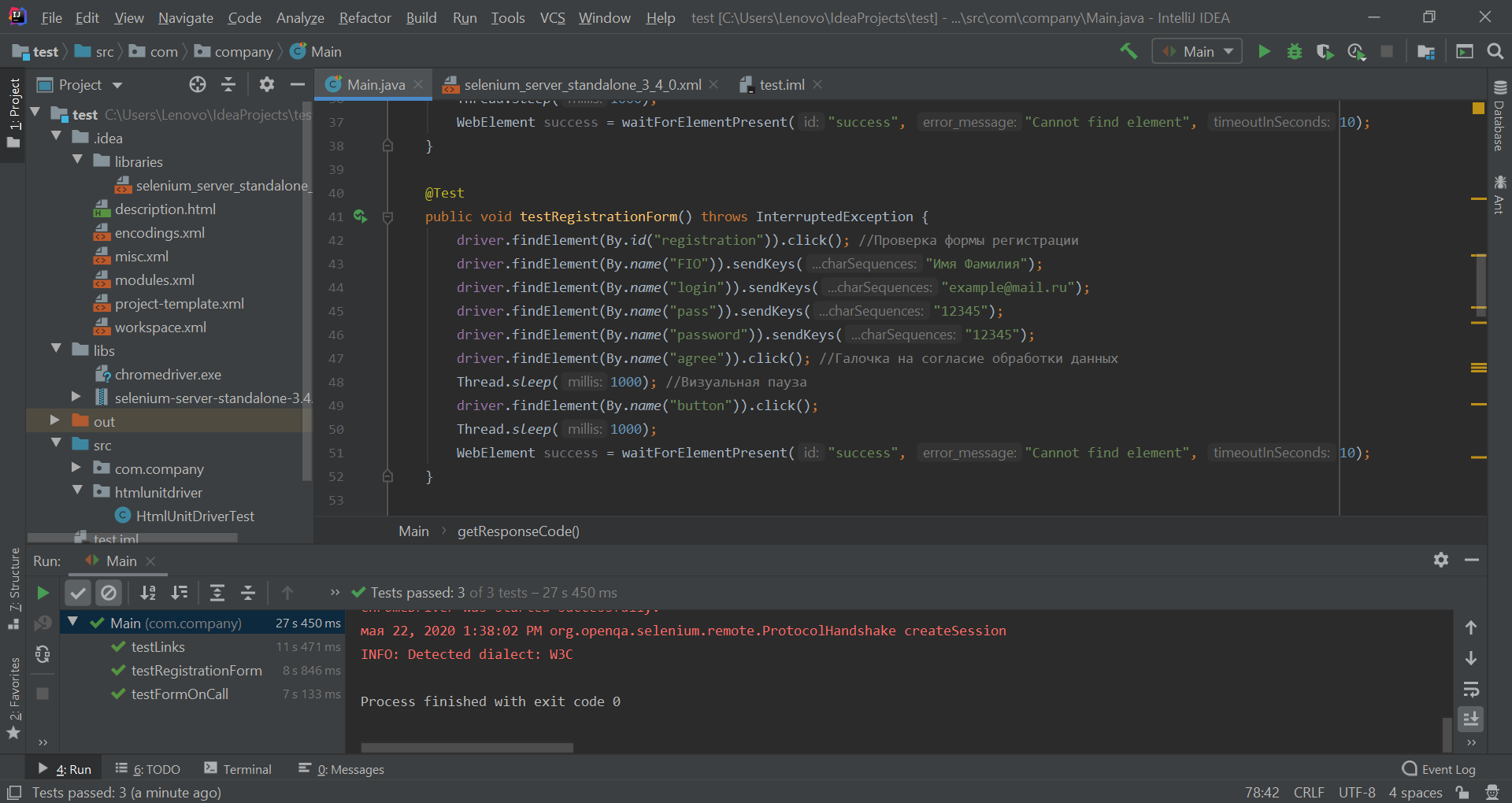
map.get(true); // содержит нормальные ссылки

map.get(false); // содержит неисправные ссылки

System.out.println(map.get(false));

}

1. **Результат тестирования:**



Среда разработки Unit тестов: IntelliJ IDEA 2019 3.1.

Библиотека для написания Unit тестов: Selenium Server 3.4.0

## Листинг файла автотестов:

package com.company;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import java.net.HttpURLConnection;

import java.net.URL;

import java.util.List;

import java.util.Map;

import java.util.stream.Collectors;

public class Main {

private WebDriver driver;

@Before

public void setUp() throws Exception {

System.setProperty("webdriver.chrome.driver", "C:\\Users\\Lenovo\\IdeaProjects\\test\\libs\\chromedriver.exe");

driver = new ChromeDriver();

driver.get("http://localhost/vetclinic");

}

@Test

public void testFormOnCall() throws InterruptedException {

driver.findElement(By.className("popup-open")).click(); //Проверка работоспособности поп-апа

driver.findElement(By.name("name")).sendKeys("a@a.a");

driver.findElement(By.name("phone")).sendKeys("9162641800");

driver.findElement(By.name("agree")).click(); //Галочка на согласие обработки данных

Thread.sleep(1000);

driver.findElement(By.name("submit")).click(); //Отправка формы

Thread.sleep(1000);

WebElement success = waitForElementPresent("success", "Cannot find element", 10);

}

@Test

public void testRegistrationForm() throws InterruptedException {

driver.findElement(By.id("registration")).click(); //Проверка формы регистрации

driver.findElement(By.name("FIO")).sendKeys("Имя Фамилия");

driver.findElement(By.name("login")).sendKeys("example@mail.ru");

driver.findElement(By.name("pass")).sendKeys("12345");

driver.findElement(By.name("password")).sendKeys("12345");

driver.findElement(By.name("agree")).click(); //Галочка на согласие обработки данных

Thread.sleep(1000); //Визуальная пауза

driver.findElement(By.name("button")).click();

Thread.sleep(1000);

WebElement success = waitForElementPresent("success", "Cannot find element", 10);

}

@Test

public void testLinks() throws InterruptedException {

Map<Integer, List<String>> map = driver.findElements(By.xpath("//\*[@href]"))

.stream() // находим все элементы с атрибутом href

.map(ele -> ele.getAttribute("href")) // получаем их значение

.map(String::trim) // обрезаем конечные проберы

.distinct() // оставляем уникальные ссылки

.collect(Collectors.groupingBy(Main::getResponseCode)); // группируем в зависимости от кода ответа

map.get(200);// в скобках указывается код ответов

map.get(403);

map.get(404);

map.get(0); // содержит все неизвестные ссылки

map.get(true); // содержит нормальные ссылки

map.get(false); // содержит неисправные ссылки

System.out.println(map.get(false));

}

@After

public void tearDown(){

driver.close();

}

private static int getResponseCode(String link) {

URL url;

HttpURLConnection con = null;

Integer responsecode = 0;

try {

url = new URL(link);

con = (HttpURLConnection) url.openConnection();

responsecode = con.getResponseCode();

} catch (Exception e) {

// skip

} finally {

if (null != con)

con.disconnect();

}

return responsecode;

}

private WebElement waitForElementPresent(String id, String error\_message, long timeoutInSeconds) {

WebDriverWait wait = new WebDriverWait(driver, timeoutInSeconds);

wait.withMessage(error\_message + "\n");

By by = By.id(id);

return wait.until(

ExpectedConditions.presenceOfElementLocated(by)

);

}

}