

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Верификация и аттестация ПО»
Тема: Автоматизация тестирования

Студент гр. 4303

Шурыгина М.Е..

Преподаватель

Щеголева Н.Л.

Санкт-Петербург

2019

Цель работы.

Автоматизировать сценарии тестирования, описанный во второй лабораторной работе.

Ход выполнения лабораторной работы.

Были изучены 3 способа реализации тестовых драйверов:

- Условно компилируемые драйверы
- В качестве тестового драйвера используется статический метод
- Отдельный «тестирующий» класс

Для реализации тестирования форм ресурса «КиноПоиск» был выбран метод отдельно тестирующего класса из-за относительной простоты реализации и быстродействия.

Тестирование проводилось с использованием веб-драйвера Selenium, библиотеки для автоматизирования тестов JUnit в среде программирования IntelliJ IDEA.

Автоматизации подверглись поля формы «Поиск фильма» - «Название фильма», «Год», «Актер», «Создатели», «Сборы» и формы «Поиск пользователя» - «Логин», «Имя», «Фамилия», «Ссылка vk.com», «Город».

Формы представлены на рисунке 1 и 2.

Искать фильм:

+ год:
полное или частичное название фильма

...или интервал: с по

+ страна: + актер:
актер/актриса

+ прокатчик: + создатели:
режиссер/продюсер/сценарист...

+ МРАА:

+ жанр:
аниме
биография
боевик
вестерн
военный

для выбора нескольких жанров удерживайте CTRL

+ премьера:

+ сборы: \$ млн. в

что искать?

ПОИСК

Рис. 1. Форма «Поиск фильма»

Искать пользователя:

логин пользователя

имя фамилия

ссылка на профиль vk.com

пол: возраст: от до

страна: город:

интересы:

ПОИСК

Рис.2. Форма «Поиск пользователя»

Тестирование полей сводится к проверке выполнения определенных условий. Так как сама форма на этапе ввода не позволяет проверять корректность введенных значений, были разработаны сценарии, позволяющие определить это косвенно. Однако нельзя утверждать однозначно, что протестированные поля с неявным выводом не имеют в случае прохождения теста ошибок.

Для первой формы и некоторым полям второй - алгоритм сводится к следующим действиям:

- 1) Открытие страницы с формой;
- 2) Выбор поля;
- 3) Ввод значения (корректного или некорректного) или значений для полей, нуждающихся в предусловии;
- 4) Нажатие кнопки «Поиск»
- 5) Оценка соответствия результатов ожидаемых и реальных.

Для второй формы:

- 1) Вход в учетную запись ресурса «КиноПоиск» (ввод логина, пароля);
- 2) Вход в настройки профиля;
- 3) Ввод или изменение существующих данных в полях профиля, которые будут тестироваться;
- 4) Открытие страницы с формой;
- 5) Выбор поля;
- 6) Ввод значения (корректного или некорректного) или значений для полей, нуждающихся в предусловии;
- 7) Нажатие кнопки «Поиск»
- 8) Оценка соответствия результатов ожидаемых и реальных.

Таблица с результатами тестирования представлена ниже.

Таблица 1.

№	Поле	Ввод	Предусловия	Ожидаемый результат	Итог
1	«Название фильма»	Операция «Ы» и другие приключения Шурика		Переход на страницу искомого фильма	Ok
2		1+1		Переход на страницу искомого фильма	Ok
3		Chingachgook, die große Schlange		Переход на страницу искомого фильма	Fail
4		Don't Be a Menace to South Central While Drinking Your Juice in the Hood		Текст в xpath ("//*[@class='search_results_topText']") отображен в обрезанном виде	Ok
5		^&*='~@\$%\/ <>		Текст в xpath ("//*[@class='search_results_topText']") не отображен	Fail
6	«Год»	1999		Отображаемые фильмы имеют значение для xpath ("//*[@class='year']") как «1999»	Ok
7		123		Страница с ошибкой	Ok
8		19999		Страница с ошибкой	Ok
9		кек1999		Страница с ошибкой	Ok
10		19к5ек2		Страница с ошибкой	Ok
11		1999кек		Страница с ошибкой	Fail
12		ккек		Страница с ошибкой	Ok
13	«Актер»	Портман	«Название фильма»: черный	Переход на страницу искомого фильма	Ok
14		Portman		Переход на страницу искомого фильма	Ok
15		Портман, Кунис		Переход на страницу искомого фильма	Ok
16		Jürgens	«Название фильма»: и Бог создал	Переход на страницу искомого фильма	Fail
17		12%3Портман	«Название фильма»:	Переход на страницу искомого фильма	Ok

18		Портман12%3	черный	Переход на страницу искомого фильма	Ok
19		~!@\$Портман %^&*		Переход на страницу искомого фильма	Fail
20		~!По@\$рт%м а^н&*		Переход на страницу искомого фильма	Fail
21		Хеннинг Моритцен, Харриет Андерссон, Кари Сюльван, Ингрид Тулин	«Название фильма»: шепоты	Переход на страницу искомого фильма	Fail
22	«Создатель»	Даррен	«Название фильма»: черный	Переход на страницу искомого фильма	Ok
23		Дарен Аронофски		Переход на страницу искомого фильма	Ok
24		Андрес Хайнц, Марк Хейман		Переход на страницу искомого фильма	Ok
25		Uwe Schafer	«Название фильма»: турецкий	Переход на страницу искомого фильма	Ok
26		Schäfer		Переход на страницу искомого фильма	Fail
27		12%3Даррен	«Название фильма»: черный	Переход на страницу искомого фильма	Fail
28		Даррен12%3		Переход на страницу искомого фильма	Ok
29		~Д!а@pp\$%e^ н&*		Переход на страницу искомого фильма	Fail
30		Андрес Хайнц, Марк Хейман, Джон Дж. МакЛафлин, Скотт Франклин, Майк Медавой, Брайан Оливер		Переход на страницу искомого фильма	Ok
31	«Сборы»	320	«Название	Переход на страницу	Fail

			фильма»: черный «Сборы»: больше, Мир	искомого фильма	
32		300	«Название фильма»: черный «Сборы»: меньше, Мир	Переход на страницу искомого фильма	Fail
33		100	«Название фильма»: черный «Сборы»: больше, США	Переход на страницу искомого фильма	Ok
34		110	«Название фильма»: черный «Сборы»: меньше, США	Переход на страницу искомого фильма	Ok
35		2 000 000 000	«Название фильма»: черный «Сборы»: больше, Мир	Страница с ошибкой	Fail
36		abcd	«Название фильма»: черный «Сборы»: меньше, Мир	Страница с ошибкой	Fail
37		1*5+6	«Название фильма»: черный «Сборы»: меньше, Мир	Страница с ошибкой	Fail
38	«Логин»	nogenom		Переход на страницу искомого пользователя	Ok
39		1aaaaaaaaabb bbbbbbbbbcccc cccc		Переход на страницу искомого пользователя	Ok
40		abs_0		Переход на страницу искомого пользователя	Ok
41		aaaaaaaaabb bbbbbbbbbcccc		Текст в xpath ("/*[@class='search_	Fail

		ccccddddd dddeeeeeee1 2345678		results_topText']") отображен в обрезанном виде (55 символов)	
42		!@#%\$^&*()- +=[]{};':",/<>?\n 		Текст в xpath ("//*[@class='search_ results_topText']") не отображен	Fail
43		Поле «Имя» - Иван Поле «Фамилия» - Крузенштерн	Зайти в учетную запись, изменить: «Имя»: Иван , «Фамилия»: Крузенштерн	Переход на страницу искомого пользователя	Ok
44		Поле «Имя» - John Поле «Фамилия» - Johnov	Зайти в учетную запись, изменить: «Имя»: John , «Фамилия»: Johnov	Переход на страницу искомого пользователя	Fail
45	«Имя», «Фамилия»	Поле «Имя» - Preuß Поле «Фамилия» - Preußen	Зайти в учетную запись, изменить: «Имя»: Preuß , «Фамилия»: Preußen	Переход на страницу искомого пользователя	Fail
46		Поле «Имя» - aaa Поле «Фамилия» - aaaaaaaaabb bbbbbcccccc ccccddddd deeeeeeeegg gg	Зайти в учетную запись, изменить: «Имя»: aaa , «Фамилия»: aaaaaaaaabb bbbbbcccccc ccccddddd deeeeeeeegg g	Страница с ошибкой	Fail
47		Поле «Имя» - aaaaaaaaabb	Зайти в учетную	Страница с ошибкой	Fail

		bbbbbbcccccc ccccddddddddd deeeeeeeeeegg g Поле «Фамилия» - aaa	запись, изменить: «Имя»: aaaaaaaaaabb bbbbbbcccccc ccccddddddddd deeeeeeeeeegg g, «Фамилия»: aaa		
48	«Ссылка vk.com»	https://vk.com/ monegon	Зайти в учетную запись, изменить: «Ссылка vk.com»: искомая	Переход на страницу искомого пользователя	Fail
49		https://vk.com/ mon_egon		Переход на страницу искомого пользователя	Fail
50		https://vk.com/ monegon123		Переход на страницу искомого пользователя	Fail
51		https://vk.com/ mooooooooooooo oooooooooooo eeeeegoooooooo ooooon		Страница с ошибкой	Ok
52	«Город»	Москва	Зайти в учетную запись, изменить: «Город»: искомый . Зайти в форму: «Логин» - nogemon	Переход на страницу искомого пользователя	Ok
53		Санкт – Петербург		Переход на страницу искомого пользователя	Ok
54		Midgar		Переход на страницу искомого пользователя	Ok
55		Düsseldorf		Переход на страницу искомого пользователя	Ok
56		123Moscow		Страница с ошибкой	Fail
57		Mooooooooooooo ccccccскваaaa aaaaaaaaaaaa		Страница с ошибкой	Fail

Вывод.

В ходе выполнения лабораторной работы были изучены способы тестирования веб-приложений посредством автоматизации, были проверены поля 2х форм ресурса «КиноПоиск» на наличие ошибок.

На изучение технологий автоматизации ушло 10 часов, на написание самих тестов – ещё 8. Тестирование длилось 19 минут 47 секунд. Однако из-за негибкости такого способа тестирования – приходилось регулярно исправлять и уточнять ожидаемые значения. Автоматизация процесса тестирования позволяет вновь осуществлять проверку системы на однотипные значения для полей ввода, что значительно упрощает и ускоряет рефакторинг продукта.

Для данных форм ресурса было найдено большое количество ошибок (26 тестов провалены из 57), были выявлены проблемы как в самой форме, так и в ресурсе, а именно:

- некорректный ответ БД на запросы (**тест 44** продемонстрировал, что при наличии пользователя с некоторым именем в системе – найти его не удастся, **тесты 48, 49, 50** показали, что поле «Ссылка vk.com» в принципе не активно);

- неправильное формирование некоторых значений в информационных блоках (в **тестах 31, 32** для поля «Сборы» было выявлено, что для фильма «Черный лебедь» система неверно посчитала сборы в мире, что может присутствовать и на некоторых других страницах).

Ресурс содержит информацию о фильмах, имеющих в названии буквы некоторых европейских алфавитов (умлауты), отображающиеся в самом названии на ресурсе. Формы же расширенного поиска по фильмам в поля «Название фильма», «Актер», «Создатель» и поиска по пользователям в поля «Имя» и «Фамилия» - эти буквы не принимают (**тесты 3, 16, 26, 45**).

Система не способна полностью фильтровать некорректный ввод в некоторых полях, которые должны принимать значения в строго определенном диапазоне – это поля:

- «Название фильма», «Логин», не должны принимать некоторые символы (**тесты 5, 42** показали, что фильтрация присутствует, но не абсолютная);

- «Сборы», должно принимать не больше 4х знаков, числа (**тесты 35, 36, 37**);

- «Год создания», должно принимать не больше 4х знаков, числа (**тест 11** продемонстрировал отсутствие фильтрации, не смотря на то, что другие тесты в этом блоке были пройдены, однозначно проверить гипотезу об её присутствии невозможно);

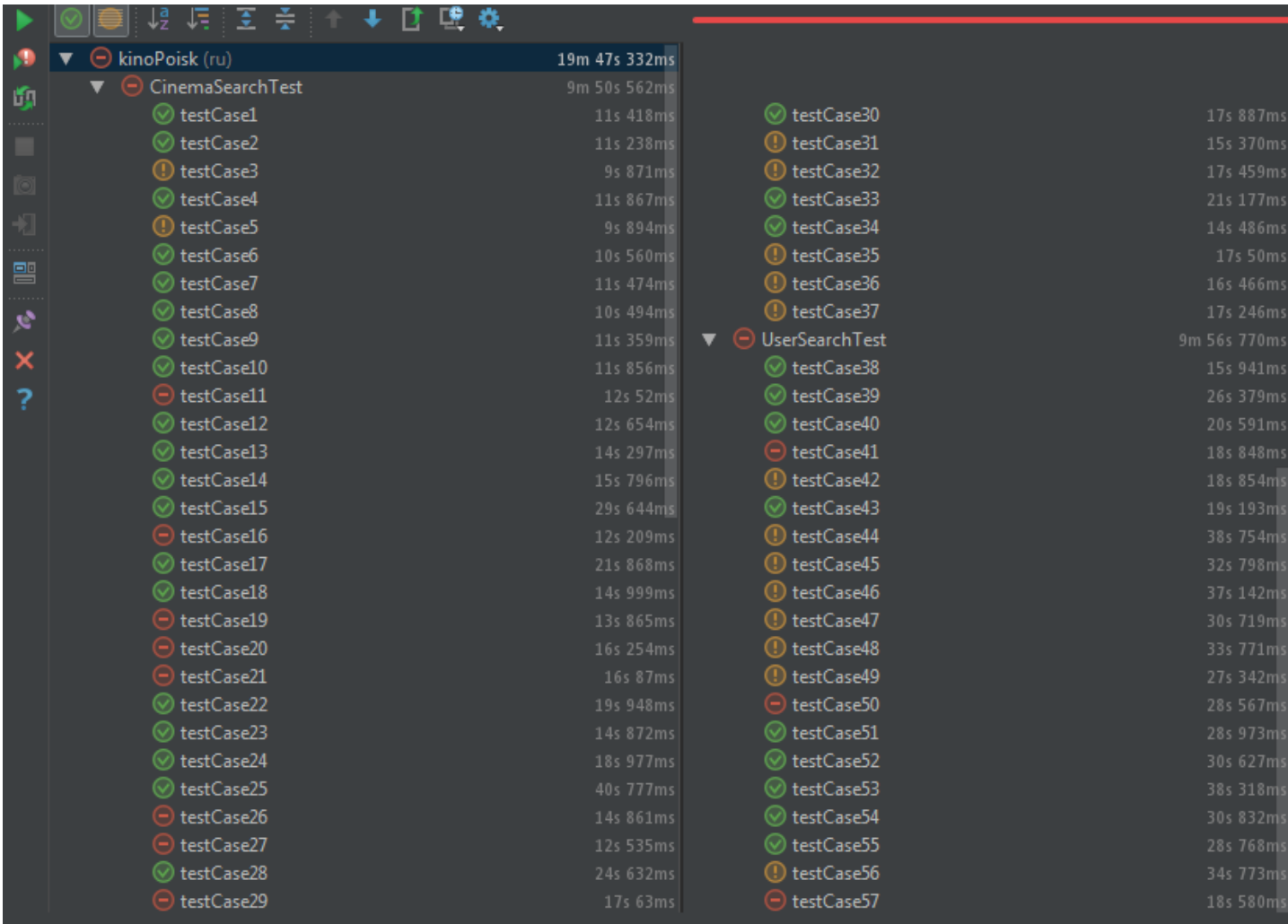
- «Актер» и «Создатель», должны принимать только буквы, запятые и дефис (**тесты 19, 20, 21, 27, 29**).

Поле «Город» во второй форме не должно принимать цифры и строки, содержащие более 32 символов, однако это условие не выполняется (**тесты 56, 57**)

Система сама предъявляет некоторые требования к полям ввода, но не полностью их придерживается. Поле «Логин» в профиле позволяет ввести значение, не превышающее 55 символов, однако форма поиска пользователя обрезает введенное значение до 50 (**тест 41**), а поля ввода «Имя» и «Фамилия» хоть и позволяют вводить до 50 символов, но в самом профиле эти поля вмещают до 30 символов (**тесты 46, 47**).

Также, эти поля в профиле не позволяют вводить в них цифры, однако форма цифры принимает и даже находит пользователей. Вероятно, это связано с переходом ресурса «КиноПоиск» под ответственность компании «Яндекс». Изменение способа аутентификации пользователя на сайте повлекло проблемы с несоответствием нынешних требований к заполнениям аутентификационных форм «Яндекса» и «отсутствующих» требований ресурса «КиноПоиск», которые не были устранены.

Результаты тестирования в IDEA представлены на рисунке 3.



kinoPoisk (ru) 19m 47s 332ms	
CinemaSearchTest 9m 50s 562ms	
testCase1	11s 418ms
testCase2	11s 238ms
testCase3	9s 871ms
testCase4	11s 867ms
testCase5	9s 894ms
testCase6	10s 560ms
testCase7	11s 474ms
testCase8	10s 494ms
testCase9	11s 359ms
testCase10	11s 856ms
testCase11	12s 52ms
testCase12	12s 654ms
testCase13	14s 297ms
testCase14	15s 796ms
testCase15	29s 644ms
testCase16	12s 209ms
testCase17	21s 868ms
testCase18	14s 999ms
testCase19	13s 865ms
testCase20	16s 254ms
testCase21	16s 87ms
testCase22	19s 948ms
testCase23	14s 872ms
testCase24	18s 977ms
testCase25	40s 777ms
testCase26	14s 861ms
testCase27	12s 535ms
testCase28	24s 632ms
testCase29	17s 63ms
testCase30	17s 887ms
testCase31	15s 370ms
testCase32	17s 459ms
testCase33	21s 177ms
testCase34	14s 486ms
testCase35	17s 50ms
testCase36	16s 466ms
testCase37	17s 246ms
UserSearchTest 9m 56s 770ms	
testCase38	15s 941ms
testCase39	26s 379ms
testCase40	20s 591ms
testCase41	18s 848ms
testCase42	18s 854ms
testCase43	19s 193ms
testCase44	38s 754ms
testCase45	32s 798ms
testCase46	37s 142ms
testCase47	30s 719ms
testCase48	33s 771ms
testCase49	27s 342ms
testCase50	28s 567ms
testCase51	28s 973ms
testCase52	30s 627ms
testCase53	38s 318ms
testCase54	30s 832ms
testCase55	28s 768ms
testCase56	34s 773ms
testCase57	18s 580ms

Рис 3. Результаты тестирования.

ПРИЛОЖЕНИЕ А

Листинг тестирующих классов

CinemaSearchTest.java

```
package ru.kinoPoisk;
import org.junit.After;
import org.junit.Assert;
import org.junit.Before;
import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.chrome.ChromeDriver;

public class CinemaSearchTest {
    private ChromeDriver driver;

    @Before
    public void setUp() {
        System.setProperty("webdriver.chrome.driver",
"C:\\Users\\Маша\\Downloads\\chromedriver1.exe");

        driver = new ChromeDriver();
        driver.get("https://www.kinopoisk.ru/s/");
    }

    @After
    public void close() {
        driver.quit();
    }

    // Название фильма
    @Test
    public void testCase1() {
        String s = "Операция «Ы» и другие приключения Шурика";
        driver.findElement(By.id("find_film")).sendKeys(s);
        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElements(By.xpath("./a[text() = '" + s +
"']")).size() != 0);
    }

    @Test
    public void testCase2() {
        String s = "1+1";
        driver.findElement(By.id("find_film")).sendKeys(s);
        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElements(By.xpath("./a[text() = '" + s +
"']")).size() != 0);
    }

    @Test
    public void testCase3() {
        String s = "Chingachgook, die große Schlange";
        driver.findElement(By.id("find_film")).sendKeys(s);
        driver.findElement(By.className("el_18")).click();
    }
}
```

```

        Assert.assertTrue(driver.findElements(By.xpath("//a[text() = '" + s +
        ""']")).size() != 0);
    }

    @Test
    public void testCase4() {
        String orig = "Don't Be a Menace to South Central While Drinking Your
        Juice in the Hood";
        driver.findElement(By.id("find_film")).sendKeys(orig);
        driver.findElement(By.className("el_18")).click();

        String s =
        driver.findElement(By.xpath("//*[@class='search_results_topText']/b")).getText()
        ;
        Assert.assertTrue(s.equals("Dont Be a Menace to South Central While
        Drinking Y"));
    }

    @Test
    public void testCase5() {
        driver.findElement(By.id("find_film")).sendKeys("^&*~@$%/\|<>");
        driver.findElement(By.className("el_18")).click();

        Assert.assertFalse(driver.findElement(By.xpath("//*[@class='search_result
        s_topText']/b")).isDisplayed());
    }

    @Test
    public void testCase6() {
        String s = "1999";
        driver.findElement(By.id("year")).sendKeys(s);
        driver.findElement(By.className("el_18")).click();
        Assert.assertTrue(driver.findElements(By.xpath("//*[@class='year' and
        text()='1999']")).size() != 0);
    }

    // Год создания
    @Test
    public void testCase7() {
        String s = "123";
        driver.findElement(By.id("year")).sendKeys(s);
        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElement(By.xpath("//h2")).isDisplayed());
    }

    @Test
    public void testCase8() {
        String s = "19999";
        driver.findElement(By.id("year")).sendKeys(s);
        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElement(By.xpath("//h2")).isDisplayed());
    }

    @Test
    public void testCase9() {
        String s = "kek1999";
    }

```

```

        driver.findElement(By.id("year")).sendKeys(s);
        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElement(By.xpath("//h2")).isDisplayed());
    }
    @Test
    public void testCase10() {
        String s = "19к5ек2";
        driver.findElement(By.id("year")).sendKeys(s);
        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElement(By.xpath("//h2")).isDisplayed());
    }
    @Test
    public void testCase11() {
        String s = "1999кек";
        driver.findElement(By.id("year")).sendKeys(s);
        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElement(By.xpath("//h2")).isDisplayed());
    }
    @Test
    public void testCase12() {
        String s = "ккек";
        driver.findElement(By.id("year")).sendKeys(s);
        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElement(By.xpath("//h2")).isDisplayed());
    }

    //Актеп
    @Test
    public void testCase13() {
        String act = "Портман";
        String film = "Чёрный лебедь";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[actor]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-big']")).getText();
        Assert.assertTrue(s.equals(film));
    }
    @Test
    public void testCase14() {
        String act = "Portman";
        String film = "Чёрный лебедь";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[actor]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-big']")).getText();
        Assert.assertTrue(s.equals(film));
    }
    @Test
    public void testCase15() {

```

```

        String act = "Портман, Кунис";
        String film = "Чёрный лебедь";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[actor]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-
big']")).getText();
        Assert.assertTrue(s.equals(film));
    }
    @Test
    public void testCase16() {
        String act = "Jürgens";
        String film = "И Бог создал женщину";
        driver.findElement(By.id("find_film")).sendKeys("и Бог создал");
        driver.findElement(By.name("m_act[actor]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-
big']")).getText();
        Assert.assertTrue(s.equals(film));
    }
    @Test
    public void testCase17() {
        String act = "12%3Портман";
        String film = "Чёрный лебедь";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[actor]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-
big']")).getText();
        Assert.assertTrue(s.equals(film));
    }
    @Test
    public void testCase18() {
        String act = "Портман12%3";
        String film = "Чёрный лебедь";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[actor]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-
big']")).getText();
        Assert.assertTrue(s.equals(film));
    }
    @Test
    public void testCase19() {
        String act = "~!@$Портман%^&*";
        String film = "Чёрный лебедь";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[actor]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-
big']")).getText();

```



```

        Assert.assertTrue(s.equals(film));
    }

    @Test
    public void testCase20() {
        String act = "~!По@$рт%ма^н&*";
        String film = "Чёрный лебедь";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[actor]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-big']")).getText();
        Assert.assertTrue(s.equals(film));
    }

    @Test
    public void testCase21() {
        String act = "Хеннинг Моритцен, Харриет Андерссон, Кари Сьюлван, Ингрид Тулин";
        String film = "Шепоты и крики";
        driver.findElement(By.id("find_film")).sendKeys("шепоты");
        driver.findElement(By.name("m_act[actor]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-big']")).getText();
        Assert.assertTrue(s.equals(film));
    }

    //Создатель
    @Test
    public void testCase22() {
        String act = "Даррен";
        String film = "Чёрный лебедь";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[cast]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-big']")).getText();
        Assert.assertTrue(s.equals(film));
    }

    @Test
    public void testCase23() {
        String act = "Дарен Аронофски";
        String film = "Чёрный лебедь";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[cast]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-big']")).getText();
        Assert.assertTrue(s.equals(film));
    }

    @Test
    public void testCase24() {
        String act = "Андрес Хайнц, Марк Хейман";
    }

```

```

        String film = "Чёрный лебедь";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[cast]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-
big']")).getText();
        Assert.assertTrue(s.equals(film));
    }
    @Test
    public void testCase25() {
        String act = "Uwe Schafer";
        String film = "Турецкий для начинающих (сериал 2006 - 2008)";
        driver.findElement(By.id("find_film")).sendKeys("турецкий");
        driver.findElement(By.name("m_act[cast]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-
big']")).getText();
        Assert.assertTrue(s.equals(film));
    }
    @Test
    public void testCase26() {
        String act = "Schäfer";
        String film = "Турецкий для начинающих";
        driver.findElement(By.id("find_film")).sendKeys("турецкий");
        driver.findElement(By.name("m_act[cast]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-
big']")).getText();
        Assert.assertTrue(s.equals(film));
    }
    @Test
    public void testCase27() {
        String act = "12%3Даррен";
        String film = "Чёрный лебедь";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[cast]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-
big']")).getText();
        Assert.assertTrue(s.equals(film));
    }
    @Test
    public void testCase28() {
        String act = "Даррен12%3";
        String film = "Чёрный лебедь";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[cast]")).sendKeys(act);
        driver.findElement(By.className("el_18")).click();

        String s = driver.findElement(By.xpath("//*[@class = 'moviename-
big']")).getText();
        Assert.assertTrue(s.equals(film));
    }

```

```

}

@Test
public void testCase29() {
    String act = "~Д!a@pp$%e^н&*";
    String film = "Чёрный лебедь";
    driver.findElement(By.id("find_film")).sendKeys("черный");
    driver.findElement(By.name("m_act[cast]")).sendKeys(act);
    driver.findElement(By.className("el_18")).click();

    String s = driver.findElement(By.xpath("//*[@class = 'moviename-big']")).getText();
    Assert.assertTrue(s.equals(film));
}

@Test
public void testCase30() {
    String act = "Андрес Хайнц, Марк Хейман, Джон Дж. МакЛафлин, Скотт Франклин, Майк Медавой, Брайан Оливер";
    String film = "Чёрный лебедь";
    driver.findElement(By.id("find_film")).sendKeys("черный");
    driver.findElement(By.name("m_act[cast]")).sendKeys(act);
    driver.findElement(By.className("el_18")).click();

    Assert.assertTrue(driver.findElements(By.xpath("./*[text() = '" + film + "' ]")).size() != 0);
}

// Сборы
@Test
public void testCase31() {
    String money = "320";
    driver.findElement(By.id("find_film")).sendKeys("черный");
    driver.findElement(By.name("m_act[box]")).sendKeys(money);

    driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
    driver.findElement(By.xpath("//*[@name='m_act[box_vector]']/*[@value='1']")).click(); //больше
    driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
    driver.findElement(By.xpath("//*[@name='m_act[box_type]']/*[@value='world']")).click(); //в Мире

    driver.findElement(By.className("el_18")).click();
    Assert.assertTrue(driver.findElements(By.xpath("./a[text() = 'Чёрный лебедь']")).size() != 0);
}

@Test
public void testCase32() {
    String money = "300";
    driver.findElement(By.id("find_film")).sendKeys("черный");
    driver.findElement(By.name("m_act[box]")).sendKeys(money);

    driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
    driver.findElement(By.xpath("//*[@name='m_act[box_vector]']/*[@value='2']")).click(); //меньше
    driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
    driver.findElement(By.xpath("//*[@name='m_act[box_type]']/*[@value='world']")).click(); //в Мире

```

```

        driver.findElement(By.className("el_18")).click();

        Assert.assertFalse(driver.findElements(By.xpath("//a[text() = 'Чёрный лебедь']")).size() != 0);
    }

    @Test
    public void testCase33() {
        String money = "100";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[box]")).sendKeys(money);

        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']/*[@value='1']")).click(); //больше
        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
        driver.findElement(By.xpath("//*[@name='m_act[box_type]']/*[@value='usa']")).click(); //в США

        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElements(By.xpath("//a[text() = 'Чёрный лебедь']")).size() != 0);
    }

    @Test
    public void testCase34() {
        String money = "110";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[box]")).sendKeys(money);

        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']/*[@value='2']")).click(); //меньше
        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
        driver.findElement(By.xpath("//*[@name='m_act[box_type]']/*[@value='usa']")).click(); //в США

        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElements(By.xpath("//a[text() = 'Чёрный лебедь']")).size() != 0);
    }

    @Test
    public void testCase35() {
        String money = "2 000 000 000";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[box]")).sendKeys(money);

        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']/*[@value='2']")).click(); //меньше
        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
        driver.findElement(By.xpath("//*[@name='m_act[box_type]']/*[@value='world']")).click(); //в США
    }

```

```

        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElements(By.xpath("//h2")).size() != 0);
    }
    @Test
    public void testCase36() {
        String money = "abcd";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[box]")).sendKeys(money);

        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']/*[@value='2']
"))).click(); //меньше
        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
        driver.findElement(By.xpath("//*[@name='m_act[box_type]']/*[@value='world
']")).click(); //в США

        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElements(By.xpath("//h2")).size() != 0);
    }
    @Test
    public void testCase37() {
        String money = "1*5+6";
        driver.findElement(By.id("find_film")).sendKeys("черный");
        driver.findElement(By.name("m_act[box]")).sendKeys(money);

        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']/*[@value='2']
"))).click(); //меньше
        driver.findElement(By.xpath("//*[@name='m_act[box_vector]']")).click();
        driver.findElement(By.xpath("//*[@name='m_act[box_type]']/*[@value='world
']")).click(); //в США

        driver.findElement(By.className("el_18")).click();

        Assert.assertTrue(driver.findElements(By.xpath("//h2")).size() != 0);
    }
}

```

UserSearchTest.java

```
package ru.kinoPoisk;
import org.junit.After;
import org.junit.Assert;
import org.junit.Before;
import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.chrome.ChromeDriver;
import java.util.concurrent.TimeUnit;

public class UserSearchTest {
    private ChromeDriver driver;
    @Before
    public void setUp() {
        System.setProperty("webdriver.chrome.driver",
"C:\\Users\\Mama\\Downloads\\chromedriver1.exe");
        driver = new ChromeDriver();
        driver.get("https://www.kinopoisk.ru/s/");
    }
    @After
    public void close() {
        driver.quit();
    }

    private By saveButton = By.xpath("//*[@value='сохранить все']");
    private By submitYa = By.xpath("//button");

    private void login(String log, String pass) {
        driver.findElement(By.xpath("//button[text()='Войти']")).click();
        driver.findElement(By.xpath("./*[@name='login']")).sendKeys(log);
        driver.findElement(submitYa).click();
        driver.findElement(By.xpath("./*[@name='passwd']")).sendKeys(pass);
        driver.findElement(submitYa).click();
    }
    private void profileIn() {
        driver.findElement(By.xpath("./*[@class='header-fresh-user-partial-
component__avatar-hover']")).click();
        driver.findElement(By.xpath("./*[@class='header-fresh-user-partial-
component__dropdown-navigation-link-text']")).click();
        driver.findElement(By.xpath("./*[@class='addFriend']")).click();
    }
    private void changeInfo(String name, String surname) {
        driver.findElement(By.name("edit[main][first_name]")).clear();
        driver.findElement(By.name("edit[main][first_name]")).sendKeys(name);

        driver.findElement(By.name("edit[main][last_name]")).clear();
        driver.findElement(By.name("edit[main][last_name]")).sendKeys(surna
me);
        driver.findElement(saveButton).click();
    }
    private void changeVK(String name) {
        driver.findElement(By.name("edit[main][social][vkontakte]")).clear();
        driver.findElement(By.name("edit[main][social][vkontakte]")).sendKe
ys(name);
        driver.findElement(saveButton).click();
    }
}
```

```

private void changeCity(String city) {
    driver.findElement(By.name("edit[main][city]")).clear();
    driver.findElement(By.name("edit[main][city]")).sendKeys(city);
    driver.findElement(saveButton).click();
}

// Login
@Test
public void testCase38() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "nogenom";
    driver.findElement(By.id("find_user")).sendKeys(log);
    driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

    String s =
driver.findElement(By.xpath("//*[@class='nick_name']")).getText();
    Assert.assertTrue(s.equals(log));
}
@Test
public void testCase39() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "1aaaaaaaaabbbbbbbbbbbccccccccc";
    driver.findElement(By.id("find_user")).sendKeys(log);
    driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

    //Assert.assertTrue(driver.findElements(By.xpath("./*[text() = '"+ log
+ "' or text()='"+ log + " ']")).size() != 0);
    Assert.assertTrue(driver.findElements(By.xpath("//*[@class='nick_name']"))
.size() != 0);
}
@Test
public void testCase40() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "abs_0";
    driver.findElement(By.id("find_user")).sendKeys(log);
    driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

    Assert.assertTrue(driver.findElements(By.xpath("./a[text() =
'abs_0']")).size() != 0);
}
@Test
public void testCase41() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "aaaaaaaaabbbbbbbbbbbccccccccccddddddeeeeeeeeeee12345678";
    driver.findElement(By.id("find_user")).sendKeys(log);
    driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

    String s =
driver.findElement(By.xpath("//*[@class='search_results_topText']/b")).getText();
    Assert.assertTrue(s.equals("aaaaaaaaabbbbbbbbbbbccccccccccddddddeeeeeee

```

```

eeee12345"));
    }
    @Test
    public void testCase42() {
        driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
        driver.findElement(By.id("find_user")).sendKeys("^&*~@$%/\|<>");
        driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

        Assert.assertFalse(driver.findElement(By.xpath("//*[@class='search_results
_topText']/b")).isDisplayed());
    }

    //Имя, фамилия
    @Test
    public void testCase43() {
        driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
        String log = "nogenom", pass = "testtest0";
        String n = "Иван", sn = "Крузенштерн";
        login(log, pass);
        profileIn();
        changeInfo(n, sn);

        driver.get("https://www.kinopoisk.ru/s/");
        driver.findElement(By.name("m_act[name]")).sendKeys(n);
        driver.findElement(By.name("m_act[surname]")).sendKeys(sn);
        driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

        Assert.assertTrue(driver.findElements(By.xpath("./a[text() = '" + n + " "
+ sn + "']")).size() != 0);
    }
    @Test
    public void testCase44() {
        driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
        String log = "nogenom", pass = "testtest0";
        String n = "John", sn = "Johnov";
        login(log, pass);
        profileIn();
        changeInfo(n, sn);

        driver.get("https://www.kinopoisk.ru/s/");
        driver.findElement(By.name("m_act[name]")).sendKeys(n);
        driver.findElement(By.name("m_act[surname]")).sendKeys(sn);
        driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

        Assert.assertTrue(driver.findElements(By.xpath("./a[text() = '" + n + " "
+ sn + "']")).size() != 0);
    }
    @Test
    public void testCase45() {
        driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
        String log = "nogenom", pass = "testtest0";
        String n = "Preuß", sn = "Preußen";
        login(log, pass);
        profileIn();

```



```
changeInfo(n, sn);

driver.get("https://www.kinopoisk.ru/s/");
driver.findElement(By.name("m_act[name]")).sendKeys(n);
driver.findElement(By.name("m_act[surname]")).sendKeys(sn);
driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

Assert.assertTrue(driver.findElements(By.xpath("./a[text() = '" + n + " "
+ sn + "']")).size()!=0);
}
@Test
public void testCase46() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "nogenom", pass = "testtest0";
    String n = "aaa", sn =
"aaaaaaaaaabbbbbbbbbbccccccccccddddddddddeeeeeeeeegggg";
    login(log, pass);
    profileIn();
    changeInfo(n, sn);

    driver.get("https://www.kinopoisk.ru/s/");
    driver.findElement(By.name("m_act[name]")).sendKeys(n);
    driver.findElement(By.name("m_act[surname]")).sendKeys(sn);
    driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

    Assert.assertTrue(driver.findElements(By.xpath("//h2")).size()!=0);
}
@Test
public void testCase47() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "nogenom", pass = "testtest0";
    String n = "aaaaaaaaaabbbbbbbbbbccccccccccddddddddddeeeeeeeeegggg", sn =
"aaa";
    login(log, pass);
    profileIn();
    changeInfo(n, sn);

    driver.get("https://www.kinopoisk.ru/s/");
    driver.findElement(By.name("m_act[name]")).sendKeys(n);
    driver.findElement(By.name("m_act[surname]")).sendKeys(sn);
    driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

    Assert.assertTrue(driver.findElements(By.xpath("//h2")).size()!=0);
}
//Ссылка
@Test
public void testCase48() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "nogenom", pass = "testtest0";
    String s = "https://vk.com/monegon";
    login(log, pass);
    profileIn();
    changeVK(s);
```

```

        driver.get("https://www.kinopoisk.ru/s/");
        driver.findElement(By.name("m_act[social]")).sendKeys(s);
        driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

        Assert.assertFalse(driver.findElements(By.xpath("//h2")).size() != 0);
    }
    @Test
    public void testCase49() {
        driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
        String log = "nogenom", pass = "testtest0";
        String s = "https://vk.com/mon_egon";
        login(log, pass);
        profileIn();
        changeVK(s);

        driver.get("https://www.kinopoisk.ru/s/");
        driver.findElement(By.name("m_act[social]")).sendKeys(s);
        driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

        Assert.assertFalse(driver.findElements(By.xpath("//h2")).size() != 0);
    }

    @Test
    public void testCase50() {
        driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
        String log = "nogenom", pass = "testtest0";
        String s = "https://vk.com/monegon123";
        login(log, pass);
        profileIn();
        changeVK(s);

        driver.get("https://www.kinopoisk.ru/s/");
        driver.findElement(By.name("m_act[social]")).sendKeys(s);
        driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

        Assert.assertFalse(driver.findElements(By.xpath("//h2")).size() != 0);
    }
    @Test
    public void testCase51() {
        driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
        String log = "nogenom", pass = "testtest0";
        String s = "https://vk.com/moooooooooooooooooneeeeeeeeeegooooooooooooon";
        login(log, pass);
        profileIn();
        changeVK(s);

        driver.get("https://www.kinopoisk.ru/s/");
        driver.findElement(By.name("m_act[social]")).sendKeys(s);
        driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

        Assert.assertTrue(driver.findElements(By.xpath("//h2")).size() != 0);
    }
    // Город

```

```

@Test
public void testCase52() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "nogenom", pass = "testtest0";
    String c = "Москва";
    login(log, pass);
    profileIn();
    changeCity(c);

    driver.get("https://www.kinopoisk.ru/s/");
    driver.findElement(By.id("find_user")).sendKeys(log);
    driver.findElement(By.name("m_act[city]")).sendKeys(c);
    driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

    String s =
driver.findElement(By.xpath("//*[@class='nick_name']")).getText();
    Assert.assertTrue(s.equals(log));

}

@Test
public void testCase53() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "nogenom", pass = "testtest0";
    String c = "Санкт - Петербург";
    login(log, pass);
    profileIn();
    changeCity(c);

    driver.get("https://www.kinopoisk.ru/s/");
    driver.findElement(By.id("find_user")).sendKeys(log);
    driver.findElement(By.name("m_act[city]")).sendKeys(c);
    driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

    String s =
driver.findElement(By.xpath("//*[@class='nick_name']")).getText();
    Assert.assertTrue(s.equals(log));

}

@Test
public void testCase54() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "nogenom", pass = "testtest0";
    String c = "Midgar";
    login(log, pass);
    profileIn();
    changeCity(c);

    driver.get("https://www.kinopoisk.ru/s/");
    driver.findElement(By.id("find_user")).sendKeys(log);
    driver.findElement(By.name("m_act[city]")).sendKeys(c);
    driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

    String s =

```

```

driver.findElement(By.xpath("//*[@class='nick_name']")).getText();
    Assert.assertTrue(s.equals(log));

}
@Test
public void testCase55() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "nogenom", pass = "testtest0";
    String c = "Düsseldorf";
    login(log, pass);
    profileIn();
    changeCity(c);

    driver.get("https://www.kinopoisk.ru/s/");
    driver.findElement(By.id("find_user")).sendKeys(log);
    driver.findElement(By.name("m_act[city]")).sendKeys(c);
    driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

    String s =
driver.findElement(By.xpath("//*[@class='nick_name']")).getText();
    Assert.assertTrue(s.equals(log));

}
@Test
public void testCase56() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "nogenom", pass = "testtest0";
    String c = "123Moscow";
    login(log, pass);
    profileIn();
    changeCity(c);

    driver.get("https://www.kinopoisk.ru/s/");
    driver.findElement(By.id("find_user")).sendKeys(log);
    driver.findElement(By.name("m_act[city]")).sendKeys(c);
    driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

    Assert.assertTrue(driver.findElements(By.xpath("/h2")).size() != 0);

}
@Test
public void testCase57() {
    driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
    String log = "nogenom", pass = "testtest0";
    String c = "!@#$$%^&*()-+={}';:\",/<>?\\|;";
    login(log, pass);
    profileIn();
    changeCity(c);

    driver.get("https://www.kinopoisk.ru/s/");
    driver.findElement(By.id("find_user")).sendKeys(log);
    driver.findElement(By.name("m_act[city]")).sendKeys(c);
    driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

```

```

        Assert.assertTrue(driver.findElements(By.xpath("//h2")).size()!=0);

    }

    @Test
    public void testCase58() {
        driver.manage().timeouts().implicitlyWait(1, TimeUnit.SECONDS);
        String log = "nogenom", pass = "testtest0";
        String c = "MoooooooooooooooooooooCKBaaaaaaaaaaaaaaaa";
        login(log, pass);
        profileIn();
        changeCity(c);

        driver.get("https://www.kinopoisk.ru/s/");
        driver.findElement(By.id("find_user")).sendKeys(log);
        driver.findElement(By.name("m_act[city]")).sendKeys(c);
        driver.findElement(By.cssSelector("#searchAdv > form.form_4 >
input.el_9.submit.nice_button")).click();

        Assert.assertTrue(driver.findElements(By.xpath("//h2")).size()!=0);

    }
}

```