

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

**ОТЧЕТ**  
**по «UI-тестирование» практике**  
**Тема: Тестирование системы Rutube**

Студентки гр. 3383

Варфоломеева А. В.  
Земерова С. Н.  
Солдунова Е. П.

Руководитель

Шевелёва А.М.

Санкт-Петербург  
2025

## ЗАДАНИЕ НА «UI-ТЕСТИРОВАНИЕ» ПРАКТИКУ

Студентки

Варфоломеева А. В.

Земерова С. Н.

Солдунова Е. П.

Группа 3383

Тема практики: UI-тестирование

Задание на практику:

Технологии: Java, Selenide (Selenium), Junit, Maven, логирование

Чеклист: подробное описание созданных тестов в одном файле:

название системы, которую тестируете

название тестов

входные данные (по возможности)

описание теста – то, как это бы делал пользователь.

Тесты:

Нужно написать 10 тестов по одному определенному блоку / функционалу системы. .

Примечание: была выбрана система rutube.ru

Сроки прохождения практики: 25.06.2025 – 08.07.2025

Дата сдачи отчета: 07.07.2025

Дата защиты отчета: 07.07.2025

Студентки		Варфоломеева А. В. Земерова С. Н. Солдунова Е. П.
Руководитель		Шевелёва А.М.

## **АННОТАЦИЯ**

В ходе практики проведено тестирование ключевых функций видеоплатформы Rutube. Основное внимание уделено проверке: поиска видео (по полному и частичному названию), редактирования профиля пользователя (изменение имени, даты рождения, пола), работы с подписками на каналы, а также взаимодействия с контентом (лайки, комментарии). Дополнительно протестированы функционал плейлистов, истории просмотров, загрузки и публикации видео. Критериями оценки выступили корректность отображения интерфейса, сохранение данных, скорость отклика системы и обработка ошибок. Тестирование выполнялось на специально созданном тестовом канале с подготовленным контентом.

## **SUMMARY**

During the practice, testing of the key functions of the Rutube video platform was conducted. The main focus was on checking the following: video search (by full and partial title), user profile editing (changing name, date of birth, gender), working with channel subscriptions, and interacting with content (likes, comments). Additionally, the functionality of playlists, viewing history, video upload, and publication were tested. The evaluation criteria included the accuracy of interface display, data retention, system response speed, and error handling. The testing was carried out on a specially created test channel with prepared content.

## СОДЕРЖАНИЕ

	Введение	4
1.	Чеклист	5
2.	Описание классов и методов	15
3.	UML - диаграммы	50
4.	Тестирование	51
	Заключение	52
	Список использованных источников	53

## **ВВЕДЕНИЕ**

### **Цель:**

Проведение комплексного тестирования функционала видеохостинга Rutube, включая проверку основных пользовательских сценариев взаимодействия с контентом и профилем.

### **Задачи:**

1. Тестирование поиска видео:
  - Проверка корректности поиска по полному и частичному названию.
  - Анализ выдачи результатов и их релевантности.
2. Тестирование профиля пользователя:
  - Проверка редактирования личных данных (имя, дата рождения, пол).
  - Валидация сохранения изменений и отображения в интерфейсе.
3. Тестирование взаимодействия с контентом:
  - Проверка работы лайков (постановка/снятие, изменение счетчика).
  - Анализ функционала комментариев (добавление, редактирование, удаление).
4. Тестирование подписок и плейлистов:
  - Проверка подписки/отписки от каналов.
  - Тестирование создания, редактирования и управления плейлистами.
5. Тестирование истории просмотров и загрузки видео:
  - Проверка корректности отображения истории.
  - Тестирование процесса загрузки и публикации видео.
6. Оценка обработки ошибок:
  - Анализ поведения системы при некорректных действиях пользователя.

### **Методология:**

- Автоматизированное тестирование (JUnit, Selenium).
- Ручное тестирование критических сценариев.
- Использование тестового канала с подготовленным контентом для контроля результатов.

## 1. ОПИСАНИЕ РЕАЛИЗУЕМЫХ ТЕСТОВ

Тестируемая система - [rutube.ru](https://rutube.ru)

Для тестирования создан канал "Практика Лэти Тестирование 2025" с 3-мя видео:

1. "Зачем нужна математика? / Практика 2025 ЛЭТИ 3383", продолжительность - 00:03:05
2. "Искусственный интеллект и его применение в жизни / Практика 2025 ЛЭТИ 3383", продолжительность - 00:02:33
3. "Космос и его составляющие / Практика 2025 ЛЭТИ 3383", продолжительность - 00:02:59

Таблица 1 - Чеклист

№	Название теста	Описание теста	Ожидаемый результат
<b>1</b>	<b>Поиск видео</b>		
1.1	Поиск существующего о видео по полному названию	1. Авторизация на сайте 2. Нажатие на поле "Поиск" 3. Ввод полного названия существующего видео "Искусственный интеллект и его применение в жизни / Практика 2025 ЛЭТИ 3383" 4. Нажатие значка поиска в правом углу поисковой строки	Отображение ленты видеороликов, в которой на первой позиции находится искомое видео "Искусственный интеллект и его применение в жизни / Практика 2025 ЛЭТИ 3383"
1.2	Поиск существующего о видео по частичному названию	1. Авторизация на сайте 2. Нажатие на поле "Поиск" 3. Ввод части названия существующего видео "Искусственный интеллект практика 3383" 4. Нажатие значка поиска в правом углу поисковой строки	Отображение ленты видеороликов, в которой на первой позиции находится видео, название которого включает в себя все введенные ключевые слова - "Искусственный интеллект и его применение в жизни / Практика 2025 ЛЭТИ 3383"

1.2	Поиск несуществующего видео	1. Авторизация на сайте 2. Нажатие на поле "Поиск" 3. Ввод названия несуществующего видео - случайная сгенерированная строка 4. Нажатие значка поиска в правом углу поисковой строки	Отображение ленты видеороликов, названия которых включают в себя слова, введенные в поисковую строку или отображение надписи "Ничего не нашли"
2	<b>Изменение данных о пользователе</b>		
2.1	Изменение данных в поле "Имя и название канала"	1. Авторизация на сайте 2. Нажатие на иконку профиля 3. Открытие раздела "Профиль" 4. Нажатие на кнопку "Изменить личные данные" 5. Ввод "Арина" в поле "Имя и название канала" вместо "Александр" 6. Нажатие кнопки "Сохранить"	В поле "Имя и название канала" отображается новое значение: "Арина"
2.2	Изменение данных в поле "Дата рождения"	1. Авторизация на сайте 2. Нажатие на иконку профиля 3. Открытие раздела "Профиль" 4. Нажатие на кнопку "Изменить личные данные" 5. Ввод нового значения: "11.04.2005" в поле "Дата рождения" вместо предыдущего: "10.10.2000" 6. Нажатие кнопки "Сохранить"	В поле "Дата рождения" отображается новое значение: "11.04.2005"

2.3	Изменение данных в поле "Пол"	1. Авторизация на сайте 2. Нажатие на иконку профиля 3. Открытие раздела "Профиль" 4. Нажатие на кнопку "Изменить личные данные" 5. Выбор поля со значением "Женский" в переключателе пола для его смены 6. Нажатие кнопки "Сохранить"	Изначальный цвет поля со значением «Женский» (#29333D — темно-синий) изменен на #00A1E7 (голубой)
<b>3</b>	<b>Подписка на канал</b>		
3.1	Подписка на канал	1. Авторизация на сайте 2. Ввод в поле поиска "Практика Лэти Тестирование 2025" 3. Нажатие на иконку канала с названием "Практика Лэти Тестирование 2025" 4. Нажатие кнопки "Подписаться"	Текст кнопки «Подписаться» заменён на «Вы подписаны», а её цвет изменён с #00A1E7 (голубой) на #29333D (темно-синий)
3.2	Отписка от канала	1. Авторизация на сайте 2. Ввод в поле поиска "Практика Лэти Тестирование 2025" 3. Нажатие на иконку канала с названием "Практика Лэти Тестирование 2025" 4. Нажатие кнопки "Вы подписаны" 5. Нажатие на кнопку со значением "Отписаться"	Текст кнопки «Вы подписаны» заменён на «Подписаться», а её цвет изменён с #29333D (темно-синий) на #00A1E7 (голубой)
<b>4</b>	<b>Лайки</b>		
4.1	Поставить лайк	1. Авторизация на сайте. 2. Поиск видео с названием "Зачем нужна математика? / Практика 2025 ЛЭТИ 3383" и переход на него. 3. Нажатие на кнопку лайка под	Значок лайка перекрашен в белый Число лайков увеличилось на 1



		видео.	
4.2	Удаление лайка	1. Авторизация на сайте. 2. Поиск видео с названием "Зачем нужна математика? / Практика 2025 ЛЭТИ 3383" и переход на него - лайк стоит после пункта 4.1. 3. Нажатие на кнопку лайка под видео.	Значок лайка стал прежним (не покрашен) Число лайков уменьшилось на 1
<b>5</b>	<b>Комментарии</b>		
5.1	Добавление комментария	1. Авторизация на сайте. 2. Поиск видео с названием "Космос и его составляющие / Практика 2025 ЛЭТИ 3383" и переход на него. 3. Поиск под видео поля "Ваш комментарий" и нажатие на него. 3. Написание комментария "Хорошее видео!" 4. Нажатие на кнопку "Отправить".	Введённый комментарий "Хорошее видео!" отображается под видео "Космос и его составляющие / Практика 2025 ЛЭТИ 3383"
5.2	Удаление комментария	1. Авторизация на сайте. 2. Поиск видео с названием "Космос и его составляющие / Практика 2025 ЛЭТИ 3383" и переход на него. 3. Поиск под видео своего комментария "Хорошее видео!". 4. Нажатие на значок "Трехточие" для комментария. 5. В появившемся меню нажатие на "Удалить".	Комментарий "Хорошее видео!" не отображается под видео "Космос и его составляющие / Практика 2025 ЛЭТИ 3383"
<b>6</b>	<b>Плейлисты</b>		

6.1	Сохранение плейлиста	<p>1. Поиск канала, у которого есть плейлисты с видео:</p> <ul style="list-style-type: none"> <li>- поиск канала "Практика Лэти Тестирование 2025"</li> <li>- переход на канал "Практика Лэти Тестирование 2025"</li> <li>- переход в раздел "Плейлисты" канала "Практика Лэти Тестирование 2025"</li> </ul> <p>2. Выбор плейлиста с названием "Тестовый плейлист"</p> <p>3. Нажатие на значок "Троеточие" рядом с названием.</p> <p>4. Нажатие в появившемся меню на «Сохранить плейлист».</p>	<p>Плейлист "Тестовый плейлист" корректно отображается в разделе "Моё" - "Плейлисты" и позволяет перейти в его содержимое</p>
6.2	Удаление плейлиста	<p>1. Поиск сохраненного плейлиста:</p> <ul style="list-style-type: none"> <li>- переход в раздел "Плейлисты"</li> </ul> <p>2. Выбор плейлиста с названием "Тестовый плейлист"</p> <p>2. Нажатие на значок "Троеточие" рядом с названием.</p> <p>3. Нажатие в появившемся меню на "Удалить из сохраненных".</p> <p>4. Нажатие в появившемся окне на кнопку "Удалить"</p>	<p>Ранее сохраненный плейлист "Тестовый плейлист" не отображается в разделе "Моё" - "Плейлисты"</p>
7	Плейлист "Смотреть позже"		

7.1	Добавление видео в плейлист "Смотреть позже"	1. Авторизация на сайте. 2. Выбор видео для сохранения в плейлист: - Поиск видео "Искусственный интеллект и его применение в жизни / Практика 2025 ЛЭТИ 3383" - Переход на видео с названием "Искусственный интеллект и его применение в жизни / Практика 2025 ЛЭТИ 3383" 3. Нажатие на кнопку "Смотреть позже" под данным видео	Видео "Искусственный интеллект и его применение в жизни / Практика 2025 ЛЭТИ 3383" отображается в плейлисте Значок под видео "Смотреть позже" (закладка) перекрашен в белый
7.2	Удаление видео из плейлиста "Смотреть позже"	1. Авторизация на сайте. 2. Поиск для удаления из плейлиста: - Переход в плейлист "Смотреть позже" - Выбор в плейлисте видео с названием "Искусственный интеллект и его применение в жизни / Практика 2025 ЛЭТИ 3383" 2. Нажатие на значок "Трехточие" рядом с названием. 3. Нажатие в появившемся меню на "Смотреть позже".	Видео "Искусственный интеллект и его применение в жизни / Практика 2025 ЛЭТИ 3383" не отображается в плейлисте Значок под видео "Смотреть позже" (закладка) стал прежним (не покрашен)
8	<b>История просмотра</b>		
8.1	Удаление из истории просмотра	1. Авторизация на сайте. 2. Просмотр видео "Зачем нужна математика? / Практика 2025 ЛЭТИ 3383". 2. Переход в раздел "История просмотра". 3. Выбор видео с названием "Зачем нужна математика? /	Видео "Зачем нужна математика? / Практика 2025 ЛЭТИ 3383" не отображается в истории просмотра

		<p>Практика 2025 ЛЭТИ 3383".</p> <p>3. Нажатие на кнопку "Троеточие" рядом с названием видео.</p> <p>4. В появившемся меню нажатие на "Удалить из истории".</p>	
9	<b>Загрузка видео на канал</b>		
9.1	Загрузка видео на канал	<p>1. Подготовка видео для загрузки, которое будет соответствовать ограничениям рутуба (до 5 часов, до 25 ГБ, в требуемом формате) - было создано видео с параметрами:</p> <ul style="list-style-type: none"> <li>- Название - "UI-тестирование / Практика 2025 ЛЭТИ 3383"</li> <li>- Тип файла - mp4</li> <li>- Размер файла - 22,6 МБ</li> <li>- Продолжительность - 00:02:36</li> </ul> <p>А также обложка для видео:</p> <ul style="list-style-type: none"> <li>- Тип файла - jpg</li> <li>- Размер файла - 38,2 КБ</li> </ul> <p>2. Авторизация на сайте.</p> <p>3. Нажатие на кнопку "+" (Добавить) в правом верхнем углу.</p> <p>4. В появившемся окне нажатие на "Загрузить видео", которая переведет на сайт <a href="http://studio.rutube.ru">studio.rutube.ru</a>.</p> <p>5. Нажатие на кнопку "Выбрать файлы" и выбор нужного видеофайла.</p> <p>6. Ожидание окончания загрузки видео.</p> <p>7. Настройка данных о видео: название, описание, обложка,</p>	<p>Видео "UI-тестирование / Практика 2025 ЛЭТИ 3383" успешно отображается в профиле</p> <p>Видео корректно воспроизводится на сайте</p> <p>Миниатюра и название отображаются правильно</p> <p>Ссылку можно открыть и поделиться ею</p>

		<p>доступ к комментариям, добавление в плейлист, планирование времени публикации и т.д.</p> <p>8. Нажатие на кнопку "Опубликовать".</p> <p>9. Ожидание окончания обработки и модерации видео.</p>	
<b>10</b>	<b>Поделиться видео</b>		
10.1	Поделиться видео	<p>1. Авторизация на сайте</p> <p>2. Ввод в поле поиска "Космос и его составляющие / Практика 2025 ЛЭТИ 3383"</p> <p>3. Нажатие на значок поиска</p> <p>4. Нажатие на видео "Космос и его составляющие / Практика 2025 ЛЭТИ 3383"</p> <p>5. Нажатие кнопки "Поделиться"</p> <p>6. Нажатие на значок копирования в поле "Ссылка"</p>	<p>URL открытого видео с названием "Космос и его составляющие / Практика 2025 ЛЭТИ 3383" совпадает со скопированным URL</p>

## 2. ОПИСАНИЕ КЛАССОВ И МЕТОДОВ

### 2.1. Базовые элементы

- *BaseElement* – абстрактный класс для всех UI-элементов.
- *Button*, *Image*, *Input* – классы-наследники *BaseElement* для работы с кнопками, изображениями и полями ввода.
- *VideoCardModule* – компонент карточки видео (например, для отображения в результатах поиска).

#### 2.1.1. BaseElement

Класс *BaseElement* предоставляет общую функциональность для работы с веб-элементами через Selenide, инкапсулируя основные операции и проверки.

##### Константы:

- `WAIT_SECONDS` - определяет стандартное время ожидания элементов (8 сек)

##### Основное поле:

- `baseElement` - хранит ссылку на соответствующий *SelenideElement*

##### Методы:

###### *Конструктор BaseElement()* - инициализация элемента

Метод принимает XPath-шаблон для поиска элемента и значение атрибута для подстановки в шаблон, создает *SelenideElement* по сформированному XPath и не возвращает значений.

###### *isDisplayed()* - Проверка видимости элемента

Метод без аргументов проверяет наличие и видимость элемента в DOM с таймаутом 8 секунд, возвращая `true` при успехе или `false` при любых ошибках (элемент не найден/невидим/другие исключения).

#### 2.1.2. Button

Класс *Button* наследуется от *BaseElement* и предоставляет специализированную функциональность для работы с кнопками в

веб-интерфейсе, включая различные способы их локации и базовые взаимодействия.

### **Константы (шаблоны XPath):**

- ARIA\_LABEL\_XPATH - шаблон для поиска по атрибуту aria-label
- TEXT\_XPATH - шаблон для поиска по тексту кнопки
- ID\_XPATH - шаблон для поиска по ID элемента

### **Основное поле:**

Наследуется baseElement от родительского класса BaseElement

### **Методы:**

#### ***Конструктор Button() - инициализация кнопки***

Метод принимает XPath-шаблон и значение для подстановки, передает их в родительский конструктор для создания SelenideElement и не возвращает значений.

#### ***press() - Нажатие на кнопку***

Метод без аргументов выполняет стандартный клик по кнопке и не возвращает значений.

#### ***byAriaLabel() - Создание кнопки по aria-label***

Метод принимает значение aria-label, создает и возвращает экземпляр Button, используя поиск по этому атрибуту.

#### ***byId() - Создание кнопки по ID***

Метод принимает ID элемента, создает и возвращает экземпляр Button для работы с соответствующей кнопкой.

#### ***byText() - Создание кнопки по тексту***

Метод принимает текст кнопки, создаёт и возвращает экземпляр Button, используя поиск по указанному тексту.

#### ***byXPath() - Создание кнопки по произвольному XPath***

Метод принимает полный XPath, создаёт и возвращает экземпляр Button для работы с элементом по нестандартному локатору.

#### ***getText() - Получение текста кнопки***

Метод без аргументов возвращает видимый текст кнопки в виде строки.

### ***getAttribute()* - Получение значения атрибута**

Метод принимает имя атрибута и возвращает его значение для элемента кнопки в виде строки.

### **2.1.3. Image**

Класс Image наследуется от BaseElement и предоставляет специализированную функциональность для работы с элементами изображений (img) в веб-интерфейсе, включая различные способы их локации.

#### **Константы (шаблоны XPath):**

- CLASS\_XPATH - шаблон для поиска по классу изображения
- TEXT\_XPATH - шаблон для поиска по тексту изображения (альтернативный текст)

#### **Основное поле:**

Наследуется baseElement от родительского класса BaseElement

#### **Методы:**

#### ***Конструктор Image()* - инициализация элемента изображения**

Метод принимает XPath-шаблон и параметр, передает их в родительский конструктор для инициализации SelenideElement без возврата значения.

#### ***byClass()* - Создание элемента изображения по классу**

Метод принимает название класса элемента, создаёт и возвращает экземпляр Image, используя поиск по указанному классу.

#### ***byText()* - Создание элемента изображения по тексту**

Метод принимает alt-текст изображения, создаёт и возвращает экземпляр Image для работы с графическим элементом по альтернативному тексту.

### **2.1.4. Input**

Класс Input наследуется от BaseElement и предоставляет функциональность для работы с элементами ввода (input) в веб-формах. Реализует основные операции взаимодействия с текстовыми полями ввода.



### **Константы (шаблоны XPath):**

- ID\_XPATH - шаблон для поиска по ID элемента
- NAME\_XPATH - шаблон для поиска по атрибуту name
- CLASS\_XPATH - шаблон для поиска по классу элемента

### **Основное поле:**

Наследуется baseElement от родительского класса BaseElement

### **Методы:**

#### ***Конструктор Input() - инициализация поля ввода***

Метод принимает XPath-шаблон и подставляемое значение, передавая их в родительский конструктор для инициализации SelenideElement без возврата результата.

#### ***fill() - Заполнение поля ввода***

Метод принимает строку текста и вводит её в поле ввода через setValue(), не возвращая значений.

#### ***byId() - Создание элемента по ID***

Метод принимает ID элемента, создает и возвращает экземпляр Input для работы с полем ввода.

#### ***byName() - Создание элемента по атрибуту name***

Метод принимает значение атрибута name, создает и возвращает экземпляр Input для работы с полем ввода.

#### ***byClass() - Создание элемента по классу***

Метод принимает название класса, создаёт и возвращает экземпляр Input для работы с полем ввода по указанному классу.

#### ***byXPath() - Создание элемента по произвольному XPath***

Метод принимает полный XPath, создаёт и возвращает экземпляр Input для работы с полем ввода по нестандартному локатору.

### **2.1.5. VideoCardModule**

Класс VideoCardModule наследуется от BaseElement и предоставляет функциональность для работы с карточками видео, которые отображаются на

странице результатов поиска. Позволяет находить и взаимодействовать с элементами карточки видео.

### **Константы (шаблоны XPath):**

- VIDEO\_TITLE\_XPATH - шаблон для поиска по названию видео
- CHANNEL\_NAME\_XPATH - шаблон для поиска по названию канала

### **Основное поле:**

Наследуется baseElement от родительского класса BaseElement

### **Методы:**

#### ***Конструктор VideoCardModule() - инициализация карточки видео***

Метод принимает XPath-шаблон и подставляемое значение, передавая их в родительский конструктор для инициализации SelenideElement, не возвращая значений.

#### ***byVideoTitle() - Создание элемента карточки видео по названию***

Метод принимает название видео, создаёт и возвращает экземпляр VideoCardModule для работы с карточкой видео.

#### ***byChannelName() - Создание элемента карточки видео по названию канала***

Метод принимает название канала, создаёт и возвращает экземпляр VideoCardModule для работы с карточкой видео, найденной по указанному каналу.

#### ***press() - Клик по карточке видео***

Метод выполняет клик по карточке видео, используя параметры инициализации (по названию видео/канала), не принимая аргументов и не возвращая значений.

## **2.2. Страницы (pages)**

- *BasePage* – абстрактный класс с общими методами для всех страниц (например, ожидание загрузки).
- *MainPage* – главная страница (неавторизованный пользователь).
- *MainAfterLoginPage* – главная страница после авторизации.
- *MainChannelPage* – страница канала.

- *ResultsOfSearchPage* – страница с результатами поиска.
- *VideoPage* – страница просмотра видео (лайки, комменOfSearchтариин).
- *HistoryVideoPage* – страница истории просмотров.

### 2.2.1. BasePage

Класс BasePage является базовым абстрактным классом для всех страниц приложения. Предоставляет общую функциональность для работы с веб-страницами, включая базовые элементы и методы взаимодействия.

#### **Константы (шаблоны XPath):**

- BASE\_ELEMENT\_XPATH - базовый шаблон для поиска элементов страницы по атрибуту name

#### **Основные поля:**

- basePage - основной элемент страницы (SelenideElement)  
 - pageClass - класс текущей страницы (используется для навигации)

#### **Методы:**

##### ***Конструктор BasePage() - инициализация страницы***

Метод принимает класс страницы и строковый идентификатор элемента, инициализирует базовый элемент страницы по XPath-шаблону, сохраняя ссылку на класс страницы без возврата значений.

##### ***refresh() - Обновление текущей страницы***

Метод без аргументов перезагружает страницу через Selenide.refresh() и возвращает новый экземпляр текущей страницы с сохранением generic-типа.

### 2.2.2. MainPage

Класс MainPage наследуется от BasePage и представляет главную страницу сайта rutube.ru до авторизации пользователя. Содержит элементы и методы для взаимодействия с неавторизованной версией главной страницы.

#### **Основные элементы:**

- loginButton - кнопка "Вход и регистрация" (инициализируется через Button.byText)

## **Методы:**

### ***Конструктор MainPage()***

Метод без аргументов вызывает родительский конструктор, передавая класс MainPage и строковый идентификатор "main" без возврата значений.

### ***clickLoginButton() - Нажатие на кнопку входа***

Метод без аргументов вызывает нажатие на кнопку входа (loginButton) без возврата значений.

### ***switchToLoginFrame() - Переключение на iframe***

Метод без аргументов переключает контекст на первый iframe страницы (индекс 0) без возврата значений.

### ***fillPhone() - Ввод номера телефона***

Метод без аргументов заполняет поле ввода (ID "phone-or-email-login") значением из TestConfig.getPhone() без возврата значений.

### ***clickContinue() - Нажатие кнопки "Продолжить"***

Метод без аргументов находит и нажимает кнопку (ID "submit-login-continue") без возврата значений.

### ***fillPassword() - Ввод пароля***

Метод без аргументов заполняет поле ввода пароля (ID "login-password") значением из TestConfig.getPassword() без возврата значений.

### ***clickLogin() - Нажатие кнопки "Войти"***

Метод без аргументов нажимает кнопку входа (ID "submit-login") и возвращает экземпляр MainAfterLoginPage.

### ***openMainPage() - Открытие главной страницы***

Метод без аргументов создаёт и возвращает новый экземпляр MainPage.

### ***closePopups() - Закрывание всплывающих окон***

Метод без аргументов последовательно нажимает все доступные видимые кнопки закрытия, обрабатывая возможные исключения без возврата значений.

## **2.2.3. MainAfterLoginPage**

Класс `MainAfterLoginPage` наследуется от `BasePage` и представляет главную страницу Rutube после успешной авторизации пользователя. Содержит элементы и методы для взаимодействия с авторизованной версией главной страницы.

**Основные элементы:**

- `channelIconImage` - иконка профиля пользователя (инициализируется через `Image.byClass`)
- `searchInput` - поле поиска (инициализируется через `Input.byClass`)
- `searchButton` - кнопка поиска (инициализируется через `Button.byAriaLabel`)

**Методы:**

**Конструктор `MainAfterLoginPage()`**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

Отсутствует

**Описание:**

Вызывает родительский конструктор с указанием класса `MainAfterLoginPage` и типа страницы `"mainAfterLogin"`

**`isChannelIconVisible()` - Проверка видимости иконки профиля**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

`boolean` (`true` - если иконка отображается)

**Описание:**

Проверяет видимость элемента `channelIconImage` с помощью метода `isDisplayed()`

### **fillSearchInput() - Ввод поискового запроса**

#### **Аргументы метода:**

1. searchQuery (String) - текст для поиска

#### **Возвращаемое значение:**

Отсутствует

#### **Описание:**

Заполняет поле searchInput переданным значением

### **clickSearchButton() - Нажатие кнопки поиска**

#### **Аргументы метода:**

Отсутствуют

#### **Возвращаемое значение:**

Отсутствует

#### **Описание:**

Нажимает кнопку searchButton для выполнения поиска

### **closePopups() - Закрытие всплывающих окон**

#### **Аргументы метода:**

Отсутствуют

#### **Возвращаемое значение:**

Отсутствует

#### **Описание:**

1. Создает список возможных кнопок закрытия
2. Фильтрует только видимые кнопки
3. Последовательно нажимает все доступные кнопки
4. Обработывает возможные исключения

### **openHistoryVideo() - Открытие истории просмотров**

#### **Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

Экземпляр HistoryVideoPage

**Описание:**

1. Находит и нажимает кнопку перехода в историю просмотров
2. Возвращает новый экземпляр HistoryVideoPage

**openStudioRutubePage() - Открытие студии загрузки видео****Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

Экземпляр StudioRutubePage

**Описание:**

1. Нажимает кнопку "Добавить"
2. Выбирает пункт "Загрузить видео"
3. Возвращает новый экземпляр StudioRutubePage

**2.2.4. MainChannelPage**

Класс MainChannelPage наследуется от BasePage и реализует функционал для работы с главной страницей канала на Rutube. Предоставляет методы для управления подпиской на канал и проверки текущего состояния подписки.

**Основные элементы:**

- subscribeButton - кнопка "Подписаться" (инициализируется через Button.byText)
- subscribedButton - кнопка "Вы подписаны" (инициализируется через Button.byText)
- unsubscribeButton - кнопка "Отписаться" (инициализируется через Button.byText)

**Методы:****Конструктор MainChannelPage()****Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

Отсутствует

**Описание:**

Вызывает родительский конструктор с указанием класса MainPage и типа страницы "mainChannel"

**clickSubscribeButton() - Подписка на канал**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

Отсутствует

**Описание:**

Нажимает кнопку "Подписаться" для оформления подписки на канал

**clickSubscribedButton() - Открытие меню подписки**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

Отсутствует

**Описание:**

Нажимает кнопку "Вы подписаны" для открытия меню управления подпиской

**clickUnsubscribeButton() - Отписка от канала**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

Отсутствует

**Описание:**

Нажимает кнопку "Отписаться" для отмены подписки на канал



## **isSubscribedButtonVisible() - Проверка состояния подписки**

### **Аргументы метода:**

Отсутствуют

### **Возвращаемое значение:**

boolean (true - если отображается кнопка "Вы подписаны")

### **Описание:**

Проверяет видимость кнопки subscribedButton для определения текущего состояния подписки

## **2.2.5. ResultsOfSearchPage**

Класс ResultsOfSearchPage наследуется от BasePage и реализует функционал для работы со страницей результатов поиска на Rutube. Позволяет взаимодействовать с найденными видео и каналами.

### **Основные элементы:**

- videoCardModule - карточка видео/канала (инициализируется в зависимости от типа поиска)

### **Методы:**

#### **Конструктор ResultsOfSearchPage()**

### **Аргументы метода:**

1. searchQuery (String) - поисковый запрос
2. type (SearchType) - тип поиска (VIDEO/CHANNEL)

### **Возвращаемое значение:**

Отсутствует

### **Описание:**

1. Вызывает родительский конструктор с указанием класса ResultsOfSearchPage и типа страницы "resultsOfSearch"
2. Инициализирует videoCardModule в зависимости от типа поиска:
  - Для VIDEO: создает карточку по названию видео
  - Для CHANNEL: создает карточку по названию канала

## **clickChannelNameVideoCardModule() - Переход на страницу канала**

### **Аргументы метода:**

Отсутствуют

### **Возвращаемое значение:**

Экземпляр MainPage

### **Описание:**

1. Нажимает на карточку модуля видео
2. Возвращает новый экземпляр MainPage

## **clickVideoNameVideoCardModule() - Переход на страницу видео**

### **Аргументы метода:**

Отсутствуют

### **Возвращаемое значение:**

Экземпляр VideoPage

### **Описание:**

1. Нажимает на карточку модуля видео
2. Возвращает новый экземпляр VideoPage

## **2.2.6. VideoPage**

### **Назначение:**

Класс VideoPage наследуется от BasePage и реализует функционал для работы со страницей просмотра видео на Rutube. Предоставляет методы для взаимодействия с элементами страницы видео, включая систему лайков.

### **Основные элементы:**

- likeButton - кнопка лайка (локатор по XPath)
- likeSvg - SVG-элемент для проверки состояния лайка (локатор по XPath)
- likesCounter - счетчик лайков (локатор по XPath)

### **Методы:**

### **Конструктор VideoPage()**

### **Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

Отсутствует

**Описание:**

Вызывает родительский конструктор с указанием класса VideoPage и типа страницы "video"

**isLiked() - Проверка состояния лайка**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

boolean (true - если лайк активен)

**Описание:**

1. Проверяет атрибут fill у SVG-элемента (значение не "none")
2. Проверяет наличие класса "active" у кнопки лайка
3. Возвращает true, если хотя бы одна проверка прошла успешно
4. Обрабатывает возможные исключения (возвращает false)

**likeVideo() - Постановка лайка**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

Текущий экземпляр VideoPage

**Описание:**

1. Проверяет текущее состояние лайка
2. Нажимает кнопку likeButton, если лайк не активен
3. Возвращает this для поддержки цепочки вызовов

**unlikeVideo() - Снятие лайка**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

Текущий экземпляр VideoPage

**Описание:**

1. Проверяет текущее состояние лайка
2. Нажимает кнопку likeButton если лайк активен
3. Возвращает this для поддержки цепочки вызовов

**getLikesCount() - Получение количества лайков**

**Аргументы метода:**

1. defaultValue (int) - значение по умолчанию

**Возвращаемое значение:**

int (количество лайков или defaultValue)

**Описание:**

1. Проверяет существование счетчика
2. Получает и обрабатывает текстовое значение:
  - Заменяет "тыс." на "000"
  - Удаляет все нецифровые символы
3. Возвращает числовое значение или defaultValue при ошибке

**isLikesCounterExists() - Проверка наличия счетчика**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

boolean (true - если счетчик существует)

**Описание:**

Пытается получить текст счетчика, возвращает true при успехе

## **2.2.7. HistoryVideoPage**

**Назначение:**

Класс HistoryVideoPage наследуется от BasePage и реализует функционал для работы со страницей истории просмотров на Rutube. Предоставляет методы для управления историей просмотров и проверки ее состояния.

**Основные элементы:**

- cleanHistoryButton - кнопка "Очистить историю" (локатор по XPath)
- VIDEO\_CARD\_XPATH - шаблон XPath для поиска карточек видео
- MENU\_BUTTON\_XPATH - шаблон XPath для кнопки меню видео

**Методы:**

**Конструктор HistoryVideoPage()**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

Отсутствует

**Описание:**

Вызывает родительский конструктор с указанием класса HistoryVideoPage и типа страницы "HistoryVideo"

**openVideoMenu() - Открытие меню видео**

**Аргументы метода:**

1. videoTitle (String) - название видео

**Возвращаемое значение:**

Отсутствует

**Описание:**

1. Формирует XPath для указанного видео
2. Находит и нажимает кнопку меню для этого видео

**removeVideoFromHistory() - Удаление видео из истории**

**Аргументы метода:**

1. videoTitle (String) - название видео

**Возвращаемое значение:**

Отсутствует

**Описание:**

1. Находит и нажимает кнопку удаления (иконка корзины) в открытом меню видео
2. Предполагает, что меню было открыто методом `openVideoMenu()`

**isVideoPresent() - Проверка наличия видео**

**Аргументы метода:**

1. `videoTitle (String)` - название видео

**Возвращаемое значение:**

`boolean` (`true` - если видео присутствует)

**Описание:**

1. Формирует XPath для указанного видео
2. Проверяет видимость элемента
3. Возвращает `false` при возникновении исключения

**cleanHistory() - Очистка всей истории**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

Отсутствует

**Описание:**

1. Нажимает кнопку "Очистить историю"
2. Подтверждает действие во всплывающем окне (нажатие кнопки "Очистить")

**isHistoryNotEmpty() - Проверка наличия истории**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

`boolean` (`true` - если история не пуста)

**Описание:**

1. Проверяет наличие хотя бы одного элемента в истории
2. Возвращает false при возникновении исключения

**2.3. Сервисы**

- AuthService – методы для авторизации (логин/логаут).
- SearchService – методы для поиска контента.
- SearchType – перечисление типов поиска (например, VIDEO, CHANNEL).

**2.3.1. AuthService****Назначение:**

Класс AuthService предоставляет сервисный метод для выполнения полного цикла авторизации пользователя на Rutube. Инкапсулирует всю последовательность действий, необходимых для входа в систему.

**Методы:**

**auth() - Выполнение авторизации (статический метод)**

**Аргументы метода:**

Отсутствуют

**Возвращаемое значение:**

MainAfterLoginPage (экземпляр авторизованной главной страницы)

**Описание:**

Выполняет полный цикл авторизации, включая следующие шаги:

1. Открытие главной страницы (неавторизованной)
2. Закрытие возможных всплывающих окон
3. Нажатие кнопки входа/регистрации
4. Переключение на iframe формы входа
5. Ввод номера телефона (из TestConfig)
6. Нажатие кнопки "Продолжить"
7. Ввод пароля (из TestConfig)

8. Нажатие кнопки "Войти"
9. Закрытие возможных всплывающих окон после авторизации
10. Возврат экземпляра авторизованной главной страницы

### **2.3.2. SearchType**

#### **Назначение:**

Перечисление SearchType определяет типы элементов, которые могут быть найдены через поисковую систему Rutube. Используется для указания типа ожидаемых результатов поиска и соответствующей обработки этих результатов.

#### **Значения:**

- если введено название видео (VIDEO), переход на страницу видео
- если введено название канала (CHANNEL), переход на страницу канала

### **2.3.3. SearchService**

#### **Назначение:**

Класс SearchService предоставляет сервисный метод для выполнения поиска на Rutube. Инкапсулирует логику взаимодействия с поисковой системой, возвращая страницу с результатами поиска.

#### **Методы:**

**search() - Выполнение поискового запроса (статический метод)**

#### **Аргументы метода:**

1. searchQuery (String) - поисковый запрос
2. type (SearchType) - тип искомого контента (VIDEO/CHANNEL)

#### **Возвращаемое значение:**

ResultsOfSearchPage (экземпляр страницы результатов поиска)

#### **Описание:**

Выполняет полный цикл поиска, включая следующие шаги:

1. Создает экземпляр авторизованной главной страницы
2. Вводит поисковый запрос в поисковую строку
3. Нажимает кнопку поиска



4. Возвращает экземпляр страницы результатов с указанным типом поиска

## 2.4. Тесты

- `BaseTest` – базовый класс для тестов (настройка `WebDriver`, общие методы).
- `LoginTest` – тесты авторизации.
- `LikeVideoTest` – тесты лайков (как в вашем примере).
- `SubscribeTest` – тесты подписок на каналы.
- `DeleteFromHistoryTest` – тесты удаления из истории просмотров.
- `ShareTest` – тесты кнопки "Поделиться".
- `TestConfig` – конфигурация тестов (URL, учетные данные).

### 2.4.1. `BaseTest`

#### **Назначение:**

Абстрактный класс `BaseTest` служит базовым классом для всех тестовых классов в проекте. Содержит общие настройки и методы, необходимые для выполнения UI-тестов с использованием `Selenide`.

#### **Константы:**

- `BASE_URL` - URL главной страницы Rutube ("<https://rutube.ru/>")

#### **Методы:**

##### **`init()`**

#### **Аннотация:**

`@BeforeEach`

#### **Описание:**

Выполняется перед каждым тестом:

1. Вызывает метод `setUp()` для настройки браузера
2. Открывает базовую страницу через метод `open()`

##### **`tearDown()`**

#### **Аннотация:**

`@AfterEach`

#### **Описание:**

Закрывает веб-драйвер после выполнения каждого теста

#### **Основные методы:**

**open()**

#### **Описание:**

1. Открывает базовую страницу (BASE\_URL)
2. Проверяет видимость body-элемента для подтверждения загрузки страницы

**setUp()**

#### **Описание:**

Настраивает параметры тестового окружения:

1. Браузер: Chrome
2. Размер окна: 1920x1080
3. Режим headless: false (браузер виден)
4. Таймаут ожидания элементов: 40 секунд

### **2.4.2. LoginTest**

#### **Назначение:**

Класс LoginTest наследуется от BaseTest и содержит тесты, связанные с авторизацией пользователя на Rutube. Проверяет корректность работы системы входа и отображение элементов для авторизованных пользователей.

#### **Тестовые методы:**

**showConfig()**

#### **Аннотация:**

@Test

#### **Описание:**

Выводит в консоль тестовые учетные данные:

1. Получает тестовый телефон через TestConfig.getPhone()
2. Получает тестовый пароль через TestConfig.getPassword()

3. Выводит значения в стандартный вывод (System.out)

### **userIconIsDisplayedAfterSuccessfulLogin()**

#### **Аннотация:**

@Test

#### **Описание:**

Проверяет успешность авторизации:

1. Выполняет авторизацию через AuthService.auth()
2. Проверяет видимость иконки канала через isChannelIconVisible()
3. Проверяет, что иконка видима (Assertions.assertTrue)

#### **Проверяемые условия:**

- После авторизации иконка пользователя должна отображаться
- Если иконка не видна - авторизация не прошла успешно

### **2.4.3. LikeVideoTest**

#### **Назначение:**

Класс LikeVideoTest наследуется от BaseTest и содержит комплексные тесты для проверки функционала лайков на Rutube. Проверяет корректность работы системы лайков, включая постановку и снятие лайков, а также изменение счетчика.

#### **Тестовые методы:**

#### **testLikeVideo()**

#### **Аннотация:**

@Test

#### **Описание:**

Проверяет процесс постановки лайка:

1. Авторизация через AuthService.auth()
2. Поиск тестового видео через SearchService.search()
3. Переход на страницу видео

4. Подготовка состояния (снятие лайка, если он уже стоит)
5. Фиксация начального количества лайков
6. Постановка лайка через likeVideo()
7. Проверки:
  - Состояние лайка (должно стать активным)
  - Увеличение счетчика лайков на 1

**Проверяемые условия:**

- Лайк должен корректно активироваться
- Счетчик лайков должен увеличиться ровно на 1

**testUnlikeVideo()**

**Аннотация:**

@Test

**Описание:**

Проверяет процесс снятия лайка:

1. Авторизация через AuthService.auth()
2. Поиск тестового видео через SearchService.search()
3. Переход на страницу видео
4. Подготовка состояния (постановка лайка, если его нет)
5. Фиксация начального количества лайков
6. Снятие лайка через unlikeVideo()
7. Проверки:
  - Состояние лайка (должно стать неактивным)
  - Уменьшение счетчика лайков на 1

**Проверяемые условия:**

- Лайк должен корректно деактивироваться
- Счетчик лайков должен уменьшиться ровно на 1

**2.4.4. SubscribeTest**

**Назначение:**

Класс `SubscribeTest` наследуется от `BaseTest` и содержит тесты для проверки функционала подписки/отписки на каналы Rutube. Проверяет корректность изменения состояния кнопки подписки после соответствующих действий.

**Тестовые методы:****`subscribeButtonIsChangedAfterSubscription()`****Аннотация:**

@Test

**Описание:**

Проверяет изменение состояния кнопки после подписки на канал:

1. Авторизация через `AuthService.auth()`
2. Поиск канала "Практика Лэти Тестирование 2025" через `SearchService.search()`
3. Переход на страницу канала через `clickChannelNameVideoCardModule()`
4. Нажатие кнопки "Подписаться" через `clickSubscribeButton()`
5. Проверка изменения состояния кнопки на "Вы подписаны"

**Проверяемые условия:**

- После подписки кнопка должна изменить текст на "Вы подписаны"
- Утверждение через `Assertions.assertTrue()`

**`subscribeButtonIsChangedAfterUnsubscription()`****Аннотация:**

@Test

**Описание:**

Проверяет изменение состояния кнопки после отписки от канала:

1. Авторизация через `AuthService.auth()`
2. Поиск канала "Практика Лэти Тестирование 2025" через `SearchService.search()`

3. Переход на страницу канала через `clickChannelNameVideoCardModule()`
4. Открытие меню подписки через `clickSubscribedButton()`
5. Нажатие кнопки "Отписаться" через `clickUnsubscribeButton()`
6. Проверка изменения состояния кнопки на "Подписаться"

**Проверяемые условия:**

- После отписки кнопка должна изменить текст на "Подписаться"
- Утверждение через `Assertions.assertFalse()`

### **2.4.5. DeleteFromHistoryTest**

**Назначение:**

Класс `DeleteFromHistoryTest` наследуется от `BaseTest` и содержит тесты для проверки функционала управления историей просмотров на Rutube. Проверяет корректность удаления отдельных видео и полной очистки истории.

**Тестовые методы:**

**`testRemoveSingleVideoFromHistory()`**

**Аннотация:**

`@Test`

**Описание:**

Проверяет удаление конкретного видео из истории:

1. Авторизация через `AuthService.auth()`
2. Переход в историю просмотров через `openHistoryVideo()`
3. Проверка наличия тестового видео в истории
4. Открытие меню видео и удаление его из истории
5. Проверка отсутствия видео в истории после удаления

**Проверяемые условия:**

- Видео должно присутствовать в истории перед удалением

- Видео должно отсутствовать после удаления

### **testCleanEntireHistory()**

#### **Аннотация:**

@Test

#### **Описание:**

Проверяет полную очистку истории просмотров:

1. Авторизация через AuthService.auth()
2. Переход в историю просмотров
3. Проверка, что история не пуста
4. Выполнение полной очистки истории
5. Проверка, что история стала пустой

#### **Проверяемые условия:**

- История должна содержать видео перед очисткой
- История должна быть пустой после очистки

### **2.4.6. ShareTest**

#### **Назначение:**

Класс ShareTest наследуется от BaseTest и содержит тесты для проверки функционала "Поделиться" на Rutube. Проверяет корректность работы системы копирования ссылки на видео.

#### **Методы:**

**copiedLinkEqualsOpen()** - проверка что скопированная ссылка при действии "Поделиться" совпадает с ссылкой открытого видео

#### **Аннотация:**

@Test

#### **Описание:**

1. Авторизация на сайте
2. Ввод в поиске "Космос и его составляющие / Практика 2025 ЛЭТИ 3383"



3. Нажатие на карточку видео
4. Нажатие кнопки "Поделиться"
5. Сравнение текущей ссылки и полученной в результате копирования

#### **2.4.7. TestConfig**

##### **Назначение:**

Класс TestConfig предоставляет централизованный доступ к тестовым учетным данным для авторизации на Rutube. Реализует гибкую систему получения конфиденциальных данных из различных источников.

##### **Методы:**

##### **getPhone() (статический метод)**

##### **Возвращаемое значение:**

String (номер телефона)

##### **Описание:**

Получает номер телефона в следующем порядке приоритетов:

1. Из системных свойств (параметр -Drutube.phone)
2. Из переменных окружения (RUTUBE\_PHONE)
3. Если данные не найдены - выбрасывает исключение с подсказкой

##### **Исключения:**

RuntimeException с инструкцией по настройке

##### **getPassword() (статический метод)**

##### **Возвращаемое значение:**

String (пароль)

##### **Описание:**

Получает пароль:

1. Из системных свойств (параметр -Drutube.password)
2. Если данные не найдены - выбрасывает исключение с подсказкой

##### **Исключения:**

RuntimeException с инструкцией по настройке

### **3. UML-ДИАГРАММЫ**

## ЗАКЛЮЧЕНИЕ

В результате работы было проведено комплексное тестирование функционала видеохостинга Rutube. Были протестированы основные пользовательские сценарии взаимодействия с контентом и личным профилем.

Протестированы такие сценарии как: авторизация, поиск видео, написание и удаление комментария, постановка лайка и его удаление, удаление видео из истории просмотра, работа с плейлистами (в частности с плейлистом “Смотреть позже”), загрузка видео на канал, функция “Поделиться видео” и изменение профиля пользователя. Для тестирования реализованы классы страниц, элементов страниц и тестов. Класс элементов страниц защищен от тестирующих классов, взаимодействие с элементами производится только через методы класса страниц.

Для проведения тестирования были созданы аккаунты на платформе Rutube и загружены видеоматериалы, а также объединены в общий плейлист.

В ходе выполнения работы было использовано автоматизированное тестирование (JUnit, Selenium) и ручное тестирование.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Automated Testing – XPath в веб-тестировании: написание и отладка - <https://automated-testing.info/t/xpath-v-web-testirovanii-napisanie-i-otladka/23808>
2. <https://www.youtube.com/watch?v=IGaAZWmqZEE&t=2204s>
3. <https://www.youtube.com/watch?v=zGYdbHACmLI&t=459s>
4. Habr - <https://habr.com/ru/companies/vk/articles/806661/>
5. Baeldung - <https://www.baeldung.com/selenide>