**ctrl+F - включает строку поиска элементов в devTools**

1) WebElement username = driver.findElement(By.id("Email"));

username.clear();<--очищает поле ввода

---------------------------------------------------------------------------------------------------------------

2) boolean selected= driver.findElement(By.xpath("xpath")).isDisplayed(); verifies if a certain element

is present and displayed

---------------------------------------------------------------------------------------------------------------

3) WebElement element = driver.findElement(By.id("Sex-Male"));

boolean status = element.isSelected();

//Or can be written as

boolean staus = driver.findElement(By.id("Sex-Male")).isSelected();

This command only works on input elements such as radio buttons, checkboxes, select options, and menu items.

It is used to determine if an element is selected.

---------------------------------------------------**Метод .click()**------------------------------------------------------------

Например у нас есть какой то элемент ‘button’ и когда мы его нашли, хотим по нём кликнуть, согласно DOM элементам там расположена ссылка на страницу №1

WebElement element = driver.findElement(By.xpath("//button"));

element.click()

Но после того как мы кликнули открывается какая то ссылка №2. Инспектируя в DOM мы видим что в button лежит две ссылки. Особенностью метода .click() является то что Selenium нажимает в середине элемента. Чтобы избежать недопониманий стоит найти именно ту ссылку которую надо кликнуть, перейдя во внутрь вложенности.

Также ссылки могут “плавать” в элементе, быть одна чуть выше, другая чуть ниже.

Если мы тестируем данный продукт то это может явл. багом, если пользователь кликнет он перейдёт на ссылку №2, хотя предполагается что это будет ссылка №1

—--------------------------------------------------- **element.sendKeys** -----------------------------------------------------------

4) WebElement element = driver.findElement(By.id("SubmitButton"));

element.sendKeys("какойто текст" <--вводит текст в выбранном элементе. Также можно имитировать нажатие любой кнопки на клавиатуре.

element.sendKeys("address", Keys.ARROW\_RIGHT); ←В этом случае сначала вводится текст, а затем нажимается стрелка вправо, За это отвечает Class/enum - Keys.

**Из особенностей:** бывает, что в поле куда мы хотим ввести символы стоят уже символы(типа введите пароль …), в этом случае Selenium продолжит ввод текста после имеющегося (старый текст не будет удалён). В этом случае можно использовать:

element.clear(); ← очистит поле для ввода

Также можно отправить на сервер файл или картинку используя данный метод, но указать в sendKeys(“путь к файлу на пк”)

element.submit(); submits relevant information (as required) on a website

//Or can be written as

driver.findElement(By.id("SubmitButton")).submit();

useful when it comes to interacting with forms (or an element within a form) on a web page.

--------------------------------------------------------------------------------------------------------------

5) driver.findElement(By.id("Text")).isEnabled(); verifies if an element is enabled(включён) on the web page

---------------------------------------------------------------------------------------------------------------

6) WebElement element = driver.findElement(By.id("SubmitButton"));

Point point = element.getLocation();

System.out.println("X cordinate : " + point.x + "Y cordinate: " + point.y);

This command retrieves/извлекает the location of a specified element on a web page. It does not require a

parameter and returns the Point object as its result. The X and Y coordinates of the element can be derived

from the Point object returned.

--------------------------------------------------------------------------------------------------------------

7) driver.findElement(By.id("TextBox")).getText();

retrieves the text within a specific web element.

--------------------------------------------------------------------------------------------------------------

8) driver.findElement(By.id("TextBox")).getTagName();

This command does not return the value of the name attribute. It returns the tag. For example, if the code

is <input name=”foo”/>, then this command will return the tag, i.e. “input”.

--------------------------------------------------------------------------------------------------------------

9) WebElement element = driver.findElement(By.id("SubmitButton"));

String attValue = element.getAttribute("id"); //This will return "SubmitButton"

This command retrieves/извлекает the attribute value of a specified element. It uses String as the parameter

and returns a string value as its result.

--------------------------------------------------------------------------------------------------------------

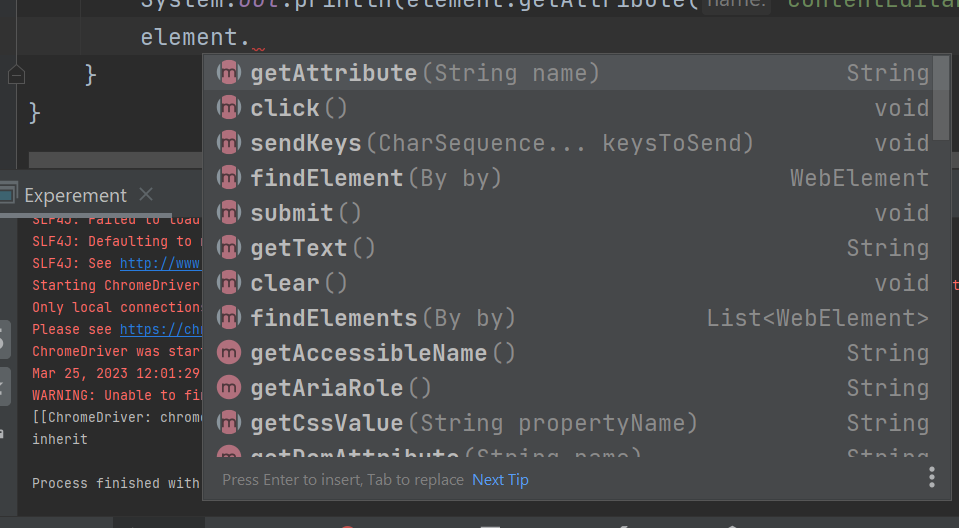
10) WebElement element = driver.findElement(By.id("SubmitButton"));

Dimension dimensions = element.getSize();

System.out.println(“Height :” + dimensions.height + ”Width : "+ dimensions.width);

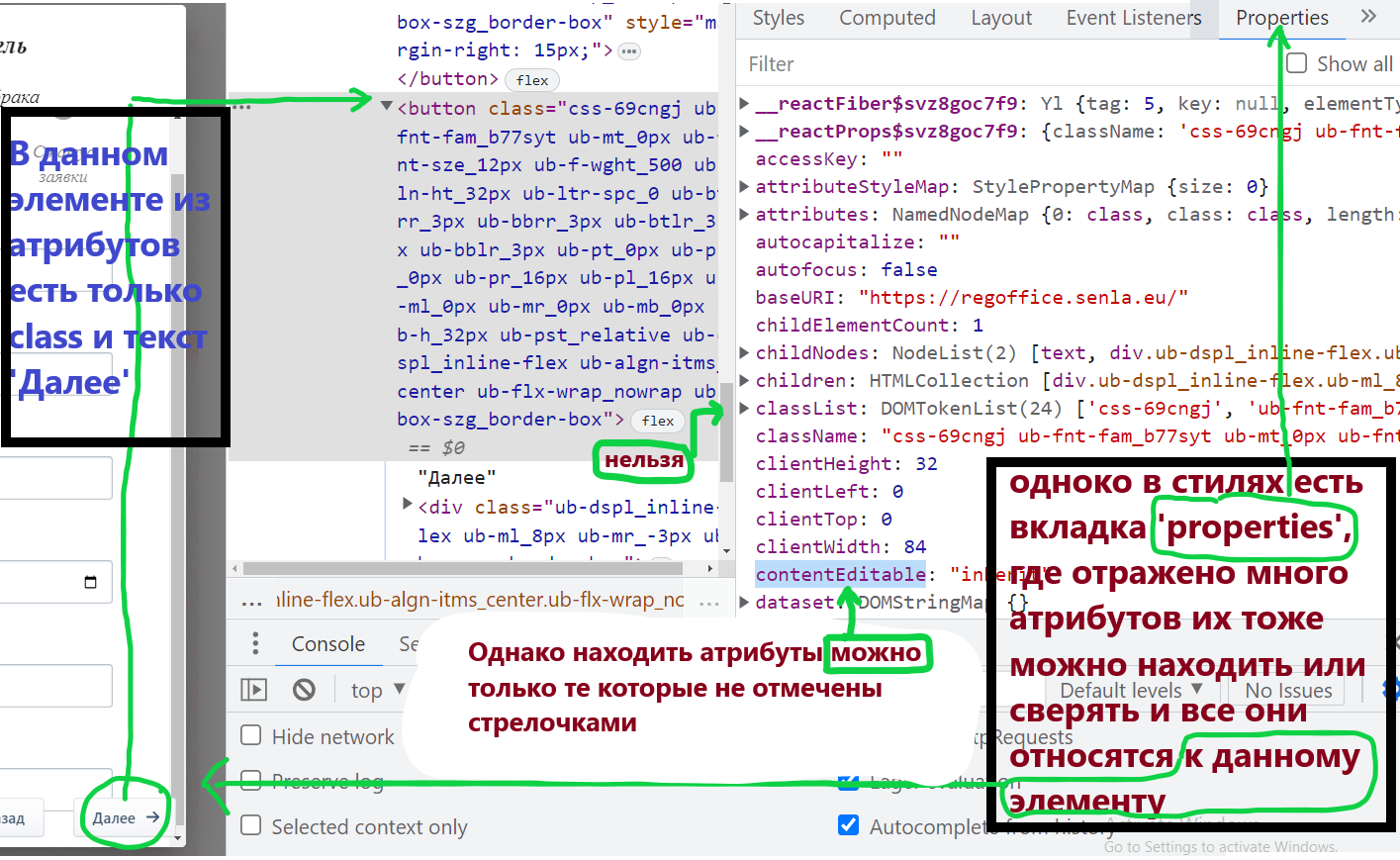
This command retrieves the height and width of a specific rendered element. It does not require a parameter

and returns the Dimension object as its result.

-------------------------------------------------- **element.getAttribute()** ------------------------------------------------------------

Когда нашол элемент, не важно с помощью какого локатора. Одним из методов является getAttribute().

element=driver.findElement(By.xpath("(//button[text()='Далее'])"));



При этом если атрибуты в DOM и Properties повторяются то значения могут отличатся, приоритетными являются те что в Properties, но почему-то находить можно только те, что без стрелки

—------------------------------------------------- **element.getCssValue()** -------------------------------------------------------------

element=driver.findElement(By.xpath("(//button[text()='Далее'])")); ←Нашли нужный элемент

element.getCssValue("background-image"); ←Выводим значение стиля "background-image"

У каждого элемента есть свои атрибуты и применённые какие-то стили, чтобы увидеть стили в наличии того или иного элемента надо посмотреть в вкладку Styles, а затем подставив название CSS стиля в метод .getCssValue(“Название Css стиля”), получим его значение.

