

O U S

A stylized owl logo composed of dark blue geometric shapes: a curved line for the top of the head, two dots for eyes, a vertical line for the beak, and two sets of three small vertical lines for feet.

ОНЛАЙН-ОБРАЗОВАНИЕ

Не забыть включить запись!

Меня хорошо
слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте если все хорошо

Правила вебинара



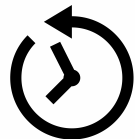
Активно участвуем, стараемся не залипать в одну точку и не клевать носом
(Хотя, если уснете, мы вас не покусаем)



Как только появляются вопросы, не стесняемся спрашивать любыми доступными методами!



Вопросы и дискуссии не по теме занятия обсуждаем в Slack или любыми другими способами, не мешающими ходу занятия



Все вопросы вижу, если не отвечаю, значит отвечу чуть-чуть позже (если вы прямо уверены, что я пропустил, то можно сказать голосом) ;)

Преподаватель



Александр Суворов
Ерам, Senior QA

В тестировании более 13 лет
Преимущественно работаю над большими
корпоративными проектами (Citibank, ВТБ, Русгидро,
Ингосстрах)
Тестирую как бэк, так и фронт.
Языки: Java, C#

Какие виды ожиданий вы знаете?

Цели вебинара

После занятия вы сможете:

1 Уверенно различать Implicit Waits и Explicit Waits

2 Использовать Expected Conditions

3 Попрактиковаться в написании теста

Задержка и виды ожиданий

Ожидания

- Implicit
- Explicit
- Fluent

Задержка

Thread.sleep()

Задание

```
@BeforeMethod
public void setup() throws InterruptedException {

    // initializing driver variable using FirefoxDriver
    driver=new FirefoxDriver();
    // launching gmail.com on the browser
    driver.get("https://gmail.com");
    // maximized the browser window
    driver.manage().window().maximize();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
}

@Test
public void test() throws InterruptedException {
    // saving the GUI element reference into a "element" variable of WebElement type
    WebElement element = driver.findElement(By.id("Email"));
    // entering username
    element.sendKeys("dummy@gmail.com");
    element.sendKeys(Keys.RETURN);
    // entering password
    driver.findElement(By.id("Passwd")).sendKeys("password");
    // clicking signin button
    driver.findElement(By.id("signIn")).click();
    // explicit wait - to wait for the compose button to be click-able
    WebDriverWait wait = new WebDriverWait(driver,30);
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[contains(text(),'COMPOSE')]")));
    // click on the compose button as soon as the "compose" button is visible
    driver.findElement(By.xpath("//div[contains(text(),'COMPOSE')]")).click();
}
```

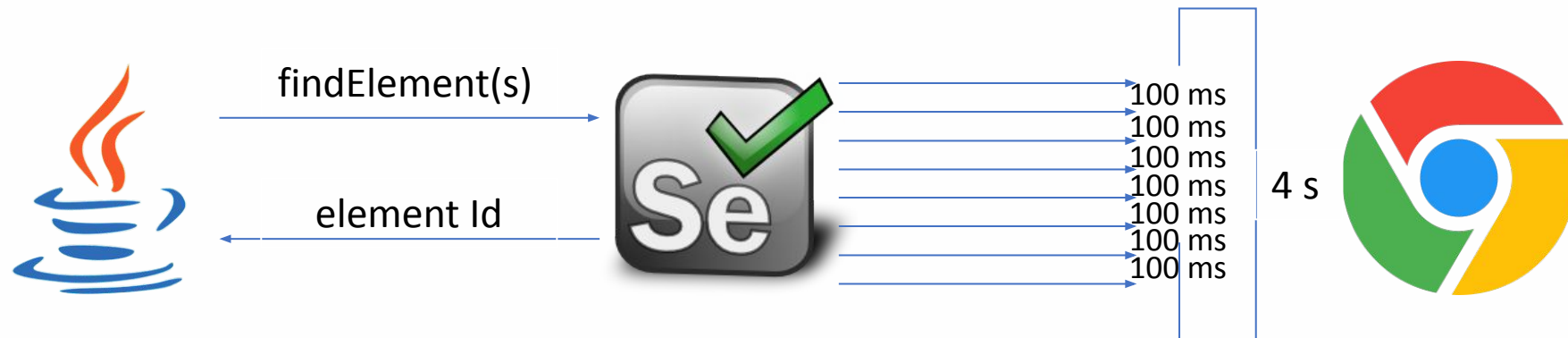
Кто такие ожидания и с чем их едят

Ожидания нужны для синхронизации работы тестируемого приложения и тестового скрипта

- Было ли у вас когда-нибудь такое, что элемент на странице не успевал появиться из-за скорости интернета?
- А бывало такое, что какое-то всплывающее окно перекрывало область клика?
- А такое, что вы вставляли в код `Thread.sleep(#бесконечностьнепредел)`, чтобы дождаться загрузки сайта?

Неявные ожидания

```
driver.manage().timeouts().implicitlyWait(4, SECONDS);  
driver.findElement(By.cssSelector("div.super"));
```



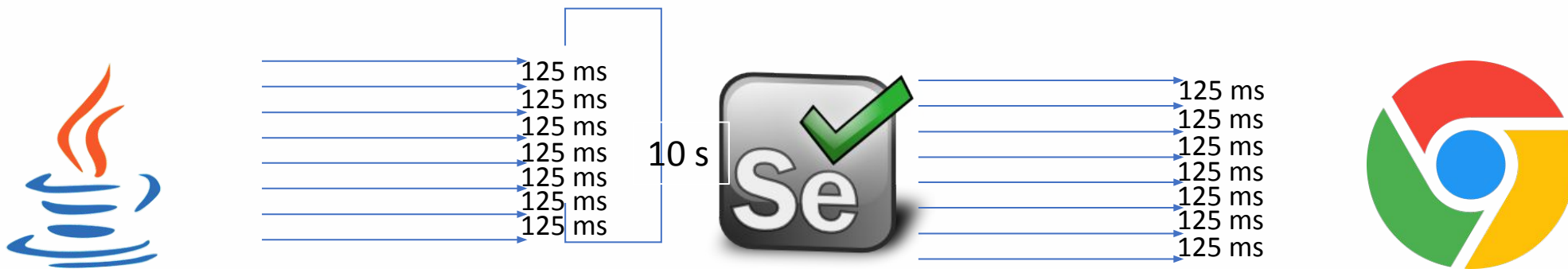
Неявные ожидания

1. Что вернет драйвер, когда найдет элемент по команде `findElement`?
2. Что вернет драйвер, когда найдет элементы по команде `findElements`?
3. Что вернет драйвер, когда НЕ найдет элемент по команде `findElement`?
4. Что вернет драйвер, когда НЕ найдет элементы по команде `findElements`?
5. После того, как юзер жмет на кнопку, с интервалом в 1 секунду появляются картинки. Всего картинок 10.
Через какое время и что именно вернет драйвер, если `implicitlyWait = 12 sec` и выполняется команда `findElements(By.tagName("img"))`

Явные ожидания, Explicit

```
WebDriverWait wait = new WebDriverWait(driver, 10L, 125L);  
wait.until(drv -> drv.findElements(By.cssSelector("div.super")).size() > 10);
```

```
driver = new ChromeDriver();  
WebDriverWait wait = new WebDriverWait(driver, 10);  
WebElement element =  
wait.until(ExpectedConditions.elementToBeClickable(locator));
```



Expected conditions

ExpectedConditions

- `elementToBeClickable()`
- `elementToBeSelected()`
- `visibilityOfElementLocated()`
- `presenceOfElementLocated()`
- ...

<https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/support/ui/ExpectedConditions.html>

Явные ожидания, Fluent

```
driver = new ChromeDriver();  
FluentWait<WebDriver> wait = new  
FluentWait<WebDriver>(driver)  
    .withTimeout(7, TimeUnit.SECONDS)  
    .pollingEvery(500, TimeUnit.MILLISECONDS)  
    .ignoring(NoSuchElementException.class);
```

Задание

Implicit(5) + Explicit(7) = ?

Задание

```
@BeforeMethod
public void setup() throws InterruptedException {

    // initializing driver variable using FirefoxDriver
    driver=new FirefoxDriver();
    // launching gmail.com on the browser
    driver.get("https://gmail.com");
    // maximized the browser window
    driver.manage().window().maximize();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
}

@Test
public void test() throws InterruptedException {
    // saving the GUI element reference into a "element" variable of WebElement type
    WebElement element = driver.findElement(By.id("Email"));
    // entering username
    element.sendKeys("dummy@gmail.com");
    element.sendKeys(Keys.RETURN);
    // entering password
    driver.findElement(By.id("Passwd")).sendKeys("password");
    // clicking signin button
    driver.findElement(By.id("signIn")).click();
    // explicit wait - to wait for the compose button to be click-able
    WebDriverWait wait = new WebDriverWait(driver,30);
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[contains(text(),'COMPOSE')]")));
    // click on the compose button as soon as the "compose" button is visible
    driver.findElement(By.xpath("//div[contains(text(),'COMPOSE')]")).click();
}
```


На нашей
стороне

В
браузере

Implicitly

Explicitly

В браузере

На нашей стороне

ждем
появления
элемента в DOM

Сами решаем

Implicitly

В браузере

Ждем появления элемента в
DOM

Explicitly

На нашей стороне

Сами решаем

Implicitly

В браузере

Ждем появления элемента в
DOM

Перепроверяет чаще (100)

Реже (500), но можно менять

Explicitly

На нашей стороне

Сами решаем

Implicitly

В браузере

Ждем появления элемента в
DOM

Перепроверяет чаще (100)

Explicitly

На нашей стороне

Сами решаем

Реже (500), но можно менять

1 запрос – 1 ответ

N

Implicitly

В браузере

Ждем появления элемента в DOM

Перепроверяет чаще (100)

1 запрос – 1 ответ

NoSuchElement

Timeout

Explicitly

На нашей стороне

Сами решаем

Реже (500), но можно менять

N

Implicitly

В браузере

Ждем появления элемента в
DOM

Перепроверяет чаще (100)

1 запрос – 1 ответ

NoSuchElement

Explicitly

На нашей стороне

Сами решаем

Реже (500), но можно менять

N

Timeout

Итоги, тезисы

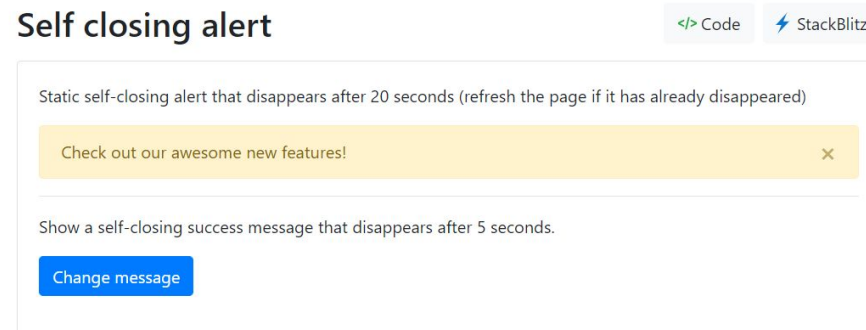
- 1 Ожидания или задержка
- 2 Implicit/Explicit/Fluent что выбрать

Практическая часть!

Задание:

- Зайти на сайт <https://ng-bootstrap.github.io/#/components/alert/examples>

- Найти блок



- Если нажимать раз в секунду на кнопку, то в всплывающем блоке будет меняться текст
- Имплементировать тест на сверку двух значений внутри блока текста после первого нажатия и после второго (интервал 1,5 секунды)

Домашнее задание

Реализуйте автоматический тест, используя Java + Selenium

Шаги теста:

- Открыть в Chrome сайт Яндекс.Маркет – “Электроника” -> “Смартфоны”
- Отфильтровать список товаров: Samsung и Xiaomi
- Отсортировать список товаров по цене (от меньшей к большей)
- Добавить первый в списке Samsung
- Проверить, что отобразилась плашка "Товар {имя товара} добавлен к сравнению"
- Добавить первый в списке Xiaomi
- Проверить, что отобразилась плашка "Товар {имя товара} добавлен к сравнению"
- Перейти в раздел Сравнение
- Проверить, что в списке товаров 2 позиции

Домашнее задание принимается в виде ссылки на GitHub репозиторий

Рефлексия



С какими основными мыслями и инсайтами уходите с вебинара



Каких целей вебинара не удалось достичь

Опрос

Заполните, пожалуйста, опрос о занятии по ссылке в чате

**Спасибо
за внимание!**

