Introduction to Selenium WebDriver and set up

Now pages becomes heavier and heavier, as a results it sometimes take time for them to load. We can see a loading screen in while something load.
Another example is partial content load, e.g. on Facebook we see part of posts, until we scroll down and then another part is loaded. Or when a news page is opened we see the headlines of big news first and then small ones are loaded.
There are several ways to tell browser to wait until something is loaded.

LOADING

Loading

Just wait some time

First way how problem can be solved is just to tell browser to wait some time or sleep for some time.

➢   `Thread.sleep(10000);`

Time here is in milliseconds (1 second x 1000), so the line above says to wait for 10 seconds.

Pros: waits for a specific time.
Cons: no idea if that time is actually too much or too little => can slow down or fail test without real reason.

Wait until element is loaded

In Selenium WebDriver we have 2 waits:
* Implicit
* Explicit

Implicit is set just once:
```
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

It sets the maximum time we will wait for element during "findElement(By)" function, by the end of which element is either successfully loaded or timeout error is seen.

Pros: waits for a specific element, fast to implement.
Cons: each time "findElement(By)" function is called it is being used, so if we want to check if an element is not present we will still have to wait for same time.

Wait until element is loaded

For explicit wait first a WebDriverWait needs to be initialized:

```
WebDriverWait wait = (WebDriverWait) new WebDriverWait(driver, 10).ignoring(StaleElementReferenceException.class);
```

Then a specific `wait.until` **method is called with expected conditions:**

```
wait.until(ExpectedConditions.presenceOfElementLocated(By));
```

Pros: waits for a specific element, doesn't wait when not required.
Cons: needs to be called specifically => slower to implement.

Selenium WebDriver basics - activity

- In this activity, you will:
  - Open the file 'Sample9Task.java'
  - Read the instructions and create the code to complete this program.

Push all your changes to repository

QUESTIONS