

---

# WARSZTATY Z TESTOWANIA OPROGRAMOWANIA SELENIUM

## UWAGI OGÓLNE

Testowanie webowych aplikacji odbiega od testowania desktopowych aplikacji, zwłaszcza jeżeli chodzi o warstwę widoku. Selenium umożliwia nam na testowanie stron www bez używania przeglądarki. Pozwala to na zautomatyzowanie testów aplikacji webowych. Dzięki paru liniom kodu możemy przetestować praktycznie każdy scenariusz, dzięki czemu częściowo odchodzi nam potrzeba 'przeklikania' aplikacji po każdej zmianie.

Podczas tych warsztatów będziemy testować kilka znanych stron internetowych, przechodząc w Selenium proste scenariusze. Projekt z podłączonym Selenium znajduje się w materiałach do zadania w folderze ./selenium/selenium. Pomocne materiały i przykłady można znaleźć na stronie Selenium:

[http://docs.seleniumhq.org/docs/03\\_webdriver.jsp](http://docs.seleniumhq.org/docs/03_webdriver.jsp)

Do działania niektórych testów należy mieć zainstalowaną przeglądarkę Firefox. Żeby włączyć testy należy otworzyć projekt w IDE (np Netbeans, Eclipse) i zbudować projekt. Alternatywnie można wykonać następujące polecenie w folderze ./selenium/selenium:

-mvn clean install

Zadania należy rozwiązywać w przygotowanej klasie testowej SeleniumTest znajdującej się w folderze ./selenium/selenium/src/test/java/com/testowanie/selenium/example.

Powodzenia,

Szymon Ślęzak

## ZADANIA DO WYKONANIA

**Zadanie 1. (1pkt)** Na podstawie testu exampleTest() napisz własny test sprawdzający działanie wybranej wyszukiwarki (np. yahoo, bing). By wykonać to zadanie będziesz musiał znaleźć nazwę elementu, jego id lub klasę w źródle strony. By potem odwołać się do obiektu można użyć odpowiednio:

***driver.findElement(By.name("nazwa"));***

***driver.findElement(By.className("nazwa\_klasy"));***

***driver.findElement(By.id("id"));***

**Zadanie 2. (2pkt)** W tym zadaniu należy przetestować sterownik do przeglądarki Firefox. Inicjalizacja sterownika w kodzie:

***WebDriver driver = new FirefoxDriver();***

Sterownik Firefox umożliwia także testowanie javascript, co nie było możliwe przy pomocy HtmlUnitDriver. Po inicjalizacji sterownika przetestuj autouzupełnianie na stronie „<http://www.google.com/webhp?complete=1&hl=en>”. Najpierw trzeba na stronie odczytać klasę tabelki autouzupełniania. Następnie należy tak jak w przykładowym teście dostać się do inputa i wpisać do niego jakieś zapytanie. Test powinien wstrzymać swoje działanie na kilka sekund by pojawiło się okienko autouzupełniania i na koniec sprawdzić czy to okienko jest widoczne:

```
WebElement table = driver.findElement(By.className("klasa_tabelki"));
```

```
table.isDisplayed();
```

**Zadanie 3. (2pkt)** W tym zadaniu należy przetestować prostą stronę logowania w załączonym do projektu pliku ./selenium/web/login.html. Mechanizm logowania używa JS więc należy skorzystać z FirefoxDriver(). By go uruchomić potrzebujemy ścieżki do strony na naszym dysku:

```
WebDriver driver = new FirefoxDriver();
```

```
driver.get("file://{ścieżka do folderu z zadaniem}/selenium/web/login.html");
```

Należy wpisać login - 'user' oraz hasło - 'password' do odpowiednich inputów na stronie. Następnie należy przycisnąć button o nazwie „Zaloguj”. Służy do tego:

```
element.click();
```

Na koniec należy sprawdzić czy logowanie zakończyło się sukcesem - czy na stronie znajduje się element o klasie „success”.

**Zadanie 4. (1pkt)** Na podstawie poprzedniego zadania napisz test wpisujący błędne (każde poza podanymi) dane logowania i sprawdzający czy na stronie znajduje się element o klasie „fail”.

**Zadanie 5. (3pkt)** Stwórz składający się z kilku kroków scenariusz użytkownika dla wybranej strony (np. wybór i dodanie artykułu do koszyka w sklepie internetowym) i napisz realizujący go test. Pamiętaj, że jeśli strona używa dynamicznych treści (JS, AJAX itp.) musisz używać sterownika Firefox i korzystać z funkcji click() zamiast submit().