

# Tester automatyzujący (kod: K-TESTER-AUTO)

## Opis i cel kursu

Testowanie manualne, jakkolwiek niezbędne i stosowane szeroko w projektach, jest procesem kosztownym i czasochłonnym. Automatyzacja testów umożliwia wielokrotne wykonywanie raz przygotowanego zestawu scenariuszy testowych, co ułatwia bieżącą kontrolę jakości. Dlatego **producenci oprogramowania starają się w jak największym stopniu automatyzować wykonywanie testów.**

Oznacza to jednocześnie, że istnieje **rynkowe zapotrzebowanie** na testerów potrafiących przygotowywać testy zautomatyzowane. Takie kompetencje są w cenie i od razu plasują daną osobę wśród wyżej wykwalifikowanych niż testerzy wyłącznie manualni. Z drugiej strony, posiadane doświadczenie testera manualnego pozytywnie wpływa na jakość tworzonych testów automatycznych - po prostu lepiej można wyczuć jak ułożyć scenariusz, co wymaga sprawdzenia itp. Bardzo często rola testera automatyzującego jest też **pośrednim etapem w drodze do zostania programistą.**

Automatyzacji testów można dokonywać w oparciu o różne języki programowania i narzędzia. Najczęściej podstawą jest ten sam język, którego używa się w projekcie, ale nie jest to konieczne. Popularne są Java, Python, JavaScript, PHP. Oprócz znajomości (podstaw) danego języka, niezbędna jest znajomość bibliotek i narzędzi testowania; dla Javy takimi elementami są **JUnit** (alternatywnie TestNG) czy **Mockito**. Najczęściej testowaniu podlegają aplikacje webowe i wtedy kluczową rolę odgrywa framework do zdalnego badania stron internetowych; niezależnie od wybranego języka programowania najbardziej popularnym rozwiązaniem tego typu jest **Selenium**.

Ten kurs kompleksowo uczy kompetencji wymaganych do tworzenia zautomatyzowanych testów funkcjonalnych w oparciu o języki programowania Python oraz Java i wokół tego koncentrują się zajęcia praktyczne. Kurs pobieżnie porusza także inne tematy, aby osadzić automatyzację testów w szerszym kontekście procesu tworzenia oprogramowania.

## Program

1. Co to jest automatyzacja? Frontend i backend, HTML, CSS
  - HTML atrybuty
  - Struktura pliku html
2. Lokalizowanie elementów na stronie
  - za pomocą name, id, tagu
  - za pomocą selektorów CSS
  - za pomocą XPath
3. Selenium podstawy – teoretyczne
  - Selenium IDE
  - Selenium WebDriver
  - Selenium Grid
4. Środowisko programistyczne Python
  - Wprowadzenie do programowania w Python dla testerów
  - przechowywanie danych w zmiennych
  - operacje na liczbach i tekstach
  - warunki logiczne
  - instrukcja if

## Zapytaj o szczegóły

tel. 22 63 64 164

akademia@alx.pl

## Najbliższe terminy

2024-06-06 (Zdalnie)  
2024-06-06 (Warszawa)  
2024-06-22 (Zdalnie)  
2024-06-22 (Warszawa)  
2024-10-12 (Warszawa)  
2024-10-12 (Zdalnie)  
2024-10-24 (Zdalnie)  
2024-10-24 (Warszawa)

- listy, wybieranie elementów i fragmentów, przeglądanie danych (pętla for)
- IDLE - środowisko programistyczne dostarczone z Pythonem
- Stworzenie projektu w PyCharmie
- 5. Narzędzia do testowania - JUnit i Selenium
  - Skrypt w Selenium
  - Zarządzanie przeglądarką za pomocą webdriver manager
  - Narzędzia deweloperskie w przeglądarce
- 6. Tworzenie testów automatycznych aplikacji internetowych
  - Akcje na elementach
  - Weryfikacja stanu elementu
  - Explicit wait – WebDriverWait
- 7. Framework PyTest
  - Pytest fixtures
  - Pytest parametrize
- 8. Testowanie backend
  - Symulacja i wykonanie testu aplikacji wyszukiwarki
  - Page Object Pattern
  - Raporty z egzekucji testów
- 9. Selenium Grid - utworzenie huba i node
  - Uruchomienie testu na Selenium Grid
- 10. Środowisko programistyczne Javy
  - Podstawy programowania w Java
    - typy podstawowe
    - instrukcja sterującą if
    - switch
    - pętle
    - metody
  - Koncepcje programowania obiektowego
- 11. Selenium – Java interakcje na elementach
  - Klikanie na elementy
  - Pobieranie tekstu
  - Sprawdzenie czy element istnieje na stronie
  - Wybieranie wartości z pola wielokrotnego wyboru
  - Sprawdzanie czy stan oczekiwany zgadza się ze stanem aktualnym
- 12. JUNIT
  - Zastosowanie adnotacji: @BeforeClass, @AfterClass, @Before, @Test

Dla grup zorganizowanych na zamówienie, istnieje również możliwość oparcia programu niniejszego kursu wyłącznie o jeden język programowania (np. Python czy Java).

UWAGA: czas kursu to 80 godzin na żywo z trenerem i grupą oraz ok. 40 godzin pracy samodzielnej w domu.

## Przeznaczenie i wymagania

Zalecana jest znajomość ogólnych zasad testowania, np. po odbyciu kursu Testera Manualnego lub z własnego doświadczenia.

## Certyfikaty

Uczestnicy szkolenia otrzymują imienne certyfikaty sygnowane przez ALX.

### Zapytaj o szczegóły

tel. 22 63 64 164

akademia@alx.pl

### Najbliższe terminy

2024-06-06 (Zdalnie)

2024-06-06 (Warszawa)

2024-06-22 (Zdalnie)

2024-06-22 (Warszawa)

2024-10-12 (Warszawa)

2024-10-12 (Zdalnie)

2024-10-24 (Zdalnie)

2024-10-24 (Warszawa)