

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

KARTA OPISU PRZEDMIOTU

Wydział		Informatyki	
Kierunek		Informatyka	
Specjalność			
Semestr	VI	Program studiów, dla którego obowiązuje sylabus	2024/2025
Stopień studiów	I		

Nazwa przedmiotu	Automatyzacja testów II			
Kod przedmiotu	ATII			
Łączna liczba godzin	30	Tryb	stacjonarny	niestacjonarny
Profil kształcenia	Ogólnoakademicki (A)		Praktyczny (P)	
Forma zajęć	laboratorium			
Język przedmiotu	polski			
Liczba punktów ECTS	2			

Prowadzący zajęcia	
Forma prowadzonych zajęć	Laboratorium
Wymiar zajęć	30 h
Stopień (tytuł) naukowy	
Imię	
Nazwisko	

Wymagania wstępne	Podstawowa znajomość programowania w co najmniej jednym języku, wiedza z zakresu podstaw inżynierii oprogramowania, znajomość podstawowych struktur danych i algorytmów.
Założenia i cele przedmiotu	Kurs skupia się na zaawansowanych technikach automatyzacji testów oprogramowania. Studenci poznają wzorce projektowe (Page Object Model), automatyzację testów API, integrację z procesami CI/CD, a także testowanie wydajnościowe. Celem jest przygotowanie studentów do tworzenia skalowalnych, trwałych i elastycznych zestawów testów automatycznych w środowiskach produkcyjnych.
Metody dydaktyczne	<ol style="list-style-type: none"> 1. Prezentacje multimedialne. 2. Pokazy przykładowych rozwiązań problemów. 3. Rozwiązywanie zadań praktycznych.

Efekty uczenia się (odniesienie do charakterystyk poziomów Polskiej Ramy Kwalifikacji)		Odniesienie do efektów dla kierunku	Odniesienie do efektów uczenia się wg Polskiej Ramy Kwalifikacji
WIEDZA – absolwent zna i rozumie:	W01. Zaawansowane wzorce projektowe stosowane w automatyzacji testów (np. Page Object Model, Screenplay Pattern) oraz ich wpływ na utrzymanie kodu testowego.	K_W06 K_W07 K_W14	P6S_WG P6S_WG_INŻ

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

	<p>W02. Zasady integracji testów automatycznych z pipeline CI/CD, zapewniające ciągłą weryfikację jakości oprogramowania.</p> <p>W03. Podstawy testowania usług chmurowych, automatyzacji środowisk testowych i wykorzystania kontenerów (Docker) do elastycznego zarządzania infrastrukturą.</p> <p>W04. Metody testowania API i usług sieciowych, w tym walidację odpowiedzi JSON/XML, kody statusu i bezpieczeństwo warstwy usług.</p> <p>W05. Podstawy analizy wyników testów wydajnościowych i metody identyfikowania wąskich gardeł aplikacji.</p>	<p>K_W19 K_W20</p>	
UMIEJĘTNOŚCI – absolwent potrafi:	<p>U01. Zastosować wzorce projektowe automatyzacji testów do budowy łatwych w utrzymaniu i rozszerzaniu projektów testowych.</p> <p>U02. Zintegrować testy automatyczne z narzędziami CI/CD (np. Jenkins, GitLab CI) w celu automatycznego uruchamiania testów po wdrożeniu zmian w kodzie.</p> <p>U03. Zaprojektować i wykonać testy API, używając narzędzi takich jak Postman, REST Assured, oraz zautomatyzować je w ramach projektu testowego.</p> <p>U04. Przeprowadzić testy wydajnościowe i interpretować ich wyniki, wprowadzając optymalizacje i rekomendacje dla zespołu deweloperskiego.</p> <p>U05. Zarządzać środowiskami testowymi, korzystając z kontenerów i narzędzi do provisioningu (np. Docker, Ansible) w celu skalowalnego i powtarzalnego uruchamiania testów.</p>	<p>K_U01 K_U02 K_U03 K_U04 K_U08 K_U09 K_U11 K_U15 K_U24</p>	<p>P6S_UW P6S_UW_INŻ P6S_UO P6S_KK P6S_UK</p>
KOMPETENCJE SPOŁECZNE – absolwent jest gotów do	<p>K01. Pracy w zespole, przyjmując w nim różne role.</p> <p>K02. Krytycznej oceny dostępnego oprogramowania i rozwiązań programistycznych.</p> <p>K03. Propagowania kodeksu etyki zawodowej i dbania o jakość oprogramowania.</p>	<p>K_K04 K_K05 K_K06</p>	<p>P6S_UO P6S_KR P6S_KK</p>

Lp.	Tematyka zajęć	Liczba godzin
Forma zajęć – laboratorium		
1	Zaawansowane techniki automatyzacji testów. Wzorce projektowe w automatyzacji (Page Object Model). Parametryzacja testów i dane testowe. Obsługa wyjątków i błędów w skryptach testowych.	6
2	Continuous Integration i Continuous Testing. Wprowadzenie do CI/CD. Konfiguracja	5

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

	środowiska CI (np. Jenkins). Integracja testów automatycznych z pipeline CI/CD.	
3	Automatyzacja testów API i usług sieciowych. Wprowadzenie do testowania API. Narzędzia do testowania API (Postman, REST Assured). Pisanie i automatyzacja testów API.	6
4	Zarządzanie środowiskami i danymi testowymi. Tworzenie i utrzymanie środowisk testowych. Anonimizacja i generowanie danych testowych. Wykorzystanie kontenerów (Docker) w testowaniu.	5
5	Testowanie wydajnościowe z wykorzystaniem narzędzi automatyzacji. Wprowadzenie do testów wydajnościowych. Narzędzia do testowania wydajności (np. JMeter). Konfiguracja i uruchamianie testów obciążeniowych. Zaliczenie.	8

Forma i warunki zaliczenia przedmiotu	Wykonanie projektów. Częstkowe prezentacje, zdawanie raportów, obrona projektów.	
Metody weryfikacji efektów uczenia się		Nr efektu uczenia się z sylabusu
	Ocena projektów i częstkowych prezentacji.	W01-W05, U01-U05, K01-K03

Literatura podstawowa	<ol style="list-style-type: none"> 1. A. Roman, <i>Testowanie i jakość oprogramowania. Modele, techniki, narzędzia</i>, Wydawnictwo Naukowe PWN, Warszawa 2015. 2. R. Smilgin, <i>Zawód tester. Od decyzji do zdobycia doświadczenia</i>, Wydawnictwo Naukowe PWN, Warszawa 2016.
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. K. Zmitrowicz, <i>Automatyzacja testów aplikacji webowych z wykorzystaniem Selenium WebDriver</i>, Helion, Gliwice 2020. 2. J. Osowski, <i>Testowanie aplikacji mobilnych w praktyce</i>, Helion, Gliwice 2021.

Nakład pracy studenta	
	Liczba godzin
Zajęcia dydaktyczne	30
Przygotowanie się do zajęć	5
Studiowanie literatury	5
Udział w konsultacjach	2
Przygotowanie projektu / eseju / prezentacji itp.	18
Przygotowanie się do egzaminu / zaliczenia	-
Inne	-
ŁĄCZNY nakład pracy studenta w godz.	60
Liczba punktów ECTS	2