

Karta przedmiotu

Nazwa i kod przedmiotu	Testowanie automatyczne (P) (Wykład), PG_00143998						
Kierunek studiów	Informatyka (P)						
Data rozpoczęcia studiów	październik 2024 r.		Rok akademicki realizacji przedmiotu		2025/2026		
Poziom kształcenia	I stopnia - licencjackie		Grupa zajęć		Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z praktycznym przygotowaniem zawodowym - profil praktyczny		
Forma studiów	stacjonarne		Sposób realizacji		na uczelni		
Rok studiów	2		Język wykładowy		polski		
Semestr studiów	3		Liczba punktów ECTS		3.0		
Profil kształcenia	praktyczny		Forma zaliczenia		zaliczenie		
Jednostka prowadząca							
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr Tomasz Borzyszkowski				
	Prowadzący zajęcia z przedmiotu						
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	30.0	0.0	0.0	45
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	45		0.0		30.0	75
Cel przedmiotu	Zapoznanie z nowoczesnymi metodami zapewniania jakości w projektach informatycznych poprzez testowanie i walidację oprogramowania.Poznanie wymagań stawianych systemom informatycznym (np. dostępność, wydajność) oraz narzędzi wspierających automatyczne testowanie oprogramowanie. Zapoznanie z praktykami stosowanymi w ramach automatyzacji różnego typu testów (np. testy jednostkowe, integracyjne, interfejsu użytkownika).						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[INFL3_U04] potrafi tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz wzorców projektowych	korzysta z zaawansowanych funkcjonalności systemów operacyjnych, w szczególności związanych z aspektami sieciowymi, wirtualizacją, konteneryzacją i innymi technologiami chmurowymi	[SU2] prezentacja/projekt/referat/ raport
	[INFL3_W07] ma wiedzę w zakresie projektowania, wytwarzania, testowania, wdrażania i utrzymania aplikacji webowych oraz ich bezpieczeństwa	ma wiedzę w zakresie wykorzystania narzędzi i środowisk wytwarzania, testowania i utrzymania oprogramowania	[SW2] prezentacja/projekt/referat/ raport
	[INFL3_W08] ma wiedzę w zakresie wykorzystania narzędzi i środowisk wytwarzania, testowania i utrzymania oprogramowania	ma wiedzę w zakresie projektowania, wytwarzania, testowania, wdrażania i utrzymania aplikacji webowych oraz ich bezpieczeństwa	[SW2] prezentacja/projekt/referat/ raport
	[INFL3_K02] potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania	potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania	[SK8] obserwacja samodzielnej lub zespołowej pracy studenta
Treści przedmiotu	Metodologie i rodzaje testów oprogramowania (piramida testów) Planowanie testów oprogramowania w ramach cyklu jego wytwarzania (formułowanie scenariuszy testowych) Automatyzacja testów: - narzędzia automatyzacji - testy automatyczne jako sposób zapewnienia jakości w cyklu wytwarzania oprogramowania - integracja testów automatycznych w proces wdrożenia		
Wymagania wstępne i dodatkowe	Brak		
Sposoby i kryteria oceniania osiąganych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	wykonanie doświadczeń	50.0%	90.0%
	aktywność na zajęciach	0.0%	10.0%
Zalecana lista lektur	Podstawowa lista lektur	1. Beck Kent, TDD. Sztuka tworzenia dobrego kodu Wydawnictwo Helion, 2020. 2. Harry J.W. Percival. TDD w praktyce. Niezawodny kod w języku Python. Wydawnictwo Helion, 2020.	
	Uzupełniająca lista lektur	Brak	
	Adresy eZasobów	Adresy na platformie eNauczanie:	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	1. Korzystając z metodologii TDD, napisz klasę/funkcję o nazwie hamming, obliczająca odległość Hamminga dla podanych danych testowych. 2. Dla zadanych wymagań, scenariuszy i przypadków użycia, zaprojektować i zaimplementować testy behawioralne wykorzystując bibliotekę behave w języku Python.		
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		