

Karta przedmiotu

Nazwa i kod przedmiotu	Wytwarzanie oprogramowania wysokiej jakości, PG_00209937						
Kierunek studiów	Informatyka (O)						
Data rozpoczęcia studiów	październik 2024 r.		Rok akademicki realizacji przedmiotu		2025/2026		
Poziom kształcenia	II stopnia		Grupa zajęć		Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć fakultatywnych		
Forma studiów	niestacjonarne		Sposób realizacji		na uczelni		
Rok studiów	2		Język wykładowy		polski		
Semestr studiów	4		Liczba punktów ECTS		6.0		
Profil kształcenia	ogólnoakademicki		Forma zaliczenia		zaliczenie		
Jednostka prowadząca	Rektor -> Wydział Matematyki, Fizyki i Informatyki -> Instytut Informatyki -> Zakład Inżynierii Oprogramowania						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr Tomasz Borzyszkowski				
	Prowadzący zajęcia z przedmiotu		dr Tomasz Borzyszkowski				
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	0.0	0.0	0.0	0.0	20.0	20
	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	20		0.0		80.0	100
Cel przedmiotu	Ukierunkowanie i wsparcie studenta w procesie przygotowywania pracy magisterskiej w obszarze inżynierii oprogramowania. Celem seminarium jest rozwój umiejętności samodzielnego formułowania problemów badawczych, krytycznego przeglądu literatury oraz stosowania zaawansowanych metod i narzędzi zapewniania wysokiej jakości oprogramowania (m.in. w zakresie architektury, niezawodności i testowania). Przedmiot przygotowuje również do publicznej prezentacji wyników własnych prac i obrony tez zawartych w dysertacji.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[INFMU2_K04] rozumie i docenia znaczenie uczciwości intelektualnej w działaniach własnych i innych osób; postępuje etycznie	Student wykazuje postawę uczciwości intelektualnej i rygorystycznie przestrzega zasad etyki akademickiej w procesie przygotowywania pracy dyplomowej, co przejawia się rzetelnym cytowaniem źródeł, poprawnym przypisywaniem autorstwa wykorzystanych koncepcji, algorytmów i fragmentów kodu oraz obiektywnym, wolnym od manipulacji raportowaniem wyników własnych badań.	[SK2] prezentacja/projekt/referat/raport [SK3] opracowanie tekstowe/praca pisemna
	[INFMU2_K01] zna ograniczenia własnej wiedzy i rozumie potrzebę dalszego uczenia się	Student krytycznie ocenia poziom swojej wiedzy i umiejętności w zakresie inżynierii oprogramowania, rozumiejąc, że dynamiczny rozwój technologii (np. nowe wzorce architektoniczne, narzędzia automatyzacji, cyberbezpieczeństwo) i rosnąca złożoność systemów informatycznych wymuszają ciągłe, samodzielne doskonalenie się.	[SK8] obserwacja samodzielnej lub zespołowej pracy studenta
	[INFMU2_K03] potrafi i jest gotów formułować opinie na temat podstawowych zagadnień informatycznych	Student potrafi formułować krytyczne i rzetelne opinie na temat aktualnego stanu wiedzy, standardów rynkowych oraz jakości istniejących rozwiązań informatycznych, co dokumentuje poprzez autorskie wnioski, podsumowania i analizy porównawcze zawarte w tekście realizowanej pracy dyplomowej.	[SK2] prezentacja/projekt/referat/raport
	[INFMU2_U08] potrafi pozyskiwać informacje z literatury fachowej, baz danych, Internetu oraz innych źródeł, integrować je, oceniać ich wiarygodność, dokonywać interpretacji oraz wyciągać wnioski i formułować opinie	Student potrafi przeprowadzić krytyczny przegląd polsko- i anglojęzycznej literatury naukowej oraz dokumentacji technicznej w zakresie metodyk, wzorców i narzędzi zapewniania wysokiej jakości oprogramowania, a następnie zintegrować te informacje w celu zdefiniowania obszaru badawczego własnej pracy magisterskiej.	[SU2] prezentacja/projekt/referat/raport
	[INFMU2_U10] potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia	Student potrafi zidentyfikować luki w swoich kompetencjach technicznych i metodologicznych, niezbędnych do rozwiązania podjętego problemu badawczego, a następnie zaplanować i zrealizować proces samodzielnego opanowania nowych technologii, narzędzi lub metodyk wykorzystywanych w pracy dyplomowej.	[SU2] prezentacja/projekt/referat/raport [SU3] opracowanie tekstowe/praca pisemna
	[INFMU2_W07] ma pogłębioną wiedzę na temat aktualnie obowiązujących przepisów prawnych dotyczących działalności informatyka (działalność dydaktyczna, naukowa i zawodowa) oraz własności intelektualnej	Student wykazuje pogłębioną wiedzę z zakresu ochrony własności intelektualnej i prawa autorskiego w działalności naukowej, ze szczególnym uwzględnieniem zasad rzetelności akademickiej, prawidłowego cytowania źródeł, unikania plagiatu oraz praw majątkowych i osobistych do stworzonego oprogramowania i tekstu pracy magisterskiej.	[SW2] prezentacja/projekt/referat/raport [SW3] opracowanie tekstowe/praca pisemna

	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[INFMU2_W01] ma pogłębioną wiedzę z działów matematyki niezbędnych do studiowania informatyki; dobrze rozumie rolę i znaczenie konstrukcji rozumowań matematycznych	Student ma pogłębioną wiedzę na temat matematycznych podstaw inżynierii oprogramowania (m.in. logika formalna w projektowaniu testów, statystyczna analiza niezawodności, metryki złożoności oparte na teorii grafów) i rozumie ich znaczenie w procesie obiektywnego dowodzenia i mierzenia wysokiej jakości systemów informatycznych.	[SW2] prezentacja/projekt/referat/raport
	[INFMU2_U09] potrafi przedstawić wyniki badań w postaci samodzielnie przygotowanej rozprawy (referatu) zawierającej opis i uzasadnienie celu pracy, przyjętą metodologię, wyniki oraz ich znaczenie na tle innych podobnych badań	Student potrafi samodzielnie zredagować kompletną pracę magisterską z zakresu wytwarzania oprogramowania wysokiej jakości, w której w sposób logiczny i poprawny językowo opisuje cel badań, wybraną metodologię inżynierską, uzyskane wyniki (np. metryki kodu, wyniki testów wydajnościowych) oraz poddaje je krytycznej dyskusji na tle istniejących rozwiązań literaturowych lub rynkowych.	[SU2] prezentacja/projekt/referat/raport
	[INFMU2_U02] potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego rozumowania danego tematu lub odnalezieniu brakujących elementów rozumowania	Student potrafi zidentyfikować luki w swojej wiedzy lub w dostępnych rozwiązaniach technologicznych, a następnie precyzyjnie sformułować problem badawczy oraz główne i szczegółowe pytania badawcze dla własnej pracy magisterskiej w obszarze wytwarzania oprogramowania wysokiej jakości.	[SU2] prezentacja/projekt/referat/raport
Treści przedmiotu	[INFMU2_W02] ma pogłębioną wiedzę w zakresie języków formalnych, modeli obliczeń oraz zagadnień złożoności obliczeniowej; zna aparat formalny pozwalający na formułowanie i badanie własności obiektów informatycznych	Student wykazuje pogłębioną wiedzę z zakresu teorii złożoności obliczeniowej i potrafi wykorzystać ten aparat pojęciowy do teoretycznej oraz praktycznej analizy wydajności, skalowalności i optymalizacji algorytmów (np. oceny notacji Big O) stosowanych we własnym projekcie badawczym lub inżynierskim.	[SW2] prezentacja/projekt/referat/raport
	1. Zagadnienia związane z architekturą oprogramowania: - refaktoryzacja zastanego kodu - architektura ewolucyjna i wykorzystanie metryk - zastosowanie AI do poprawy jakości / analizy kodu 2. Technologie wytwarzania oprogramowania: - transpilacja języków programowania - projektowanie programów hybrydowych - algorytmiczne wsparcie (także AI) wytwarzania oprogramowania 3. Zapewnianie jakości i wiarygodności: - specyfikowanie, weryfikowanie i testowanie inteligentnych kontraktów (smart contracts in blockchain technology) oraz zwykłych programów (obiektowych, funkcyjnych i hybrydowych) - dowodzenie poprawności kodu - testowanie podatności oprogramowania na różnego rodzaju ataki		
	Wymagania wstępne i dodatkowe		
	Sposoby i kryteria oceniania osiągniętych efektów uczenia się		
	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Ocena pracy magisterskiej	51.0%	80.0%
	Prezentacje na zajęciach	51.0%	20.0%

Zalecana lista lektur	Podstawowa lista lektur	1. Fowler M., Refaktoryzacja. Ulepszanie struktury istniejącego kodu. Wydanie II, Wydawnictwo Helion, Gliwice 2019. 2. Feathers M. C., Praca z zastanym kodem. Najlepsze techniki, Wydawnictwo Helion, Gliwice 2007 (lub nowsze dodruki). 3. Ford N., Parsons R., Kua P., Budowanie architektur ewolucyjnych. Wsparcie dla ciągłych zmian, Wydawnictwo Helion, Gliwice 2018. 4. Khorikov V., Testy jednostkowe. Zasady, praktyki i wzorce, Wydawnictwo Helion, Gliwice 2021. 5. Antonopoulos A. M., Wood G., Mastering Ethereum. Tworzenie inteligentnych kontraktów i aplikacji rozproszonych, Wydawnictwo Helion, Gliwice 2019.
	Uzupełniająca lista lektur	1. Parr T., Language Implementation Patterns: Create Your Own Domain-Specific and General Programming Languages, The Pragmatic Bookshelf, Raleigh 2010. 2. Janca T., Alice and Bob Learn Application Security, John Wiley & Sons, Indianapolis 2020. 3. Huth M., Ryan M., Logic in Computer Science: Modelling and Reasoning about Systems (2nd Edition), Cambridge University Press, Cambridge 2004. 4. Pierce B. C. i in., Software Foundations, publikacja elektroniczna (Materiały z University of Pennsylvania), udostępniana na licencji open-source i aktualizowana na bieżąco.
	Adresy eZasobów	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania		
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy	

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.