



Karta przedmiotu

Nazwa i kod przedmiotu	Inżynieria oprogramowania, PG_00044135						
Kierunek studiów	Matematyka						
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu			2026/2027		
Poziom kształcenia	II stopnia	Grupa zajęć			Grupa zajęć specjalnościowych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			polski		
Semestr studiów	2	Liczba punktów ECTS			4.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Fizyki Technicznej i Matematyki Stosowanej -> Instytut Matematyki Stosowanej						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr hab. Paweł Pilarczyk					
	Prowadzący zajęcia z przedmiotu	dr hab. Paweł Pilarczyk					
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	0.0	15.0	15.0	0.0	60
	W tym liczba godzin zajęć na odległość: 0.0						
	Adres kursu na platformie eNauczanie: <a href="https://enauczanie.pg.edu.pl/moodle/course/view.php?id=46659">https://enauczanie.pg.edu.pl/moodle/course/view.php?id=46659</a> Moodle ID: 5568 Inżynieria oprogramowania 2027 <a href="https://enauczanie.pg.edu.pl/2025/course/view.php?id=5568">https://enauczanie.pg.edu.pl/2025/course/view.php?id=5568</a>						
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	60		5.0	35.0	100	
Cel przedmiotu	Poznanie podstawowych metod inżynierii oprogramowania i zdobycie umiejętności ich stosowania. Dotyczy pracy grupowej nad projektem informatycznym dużych rozmiarów na wszystkich etapach wytwarzania oprogramowania: od inżynierii wymagań, poprzez analizę wymagań, projektowanie oprogramowania, implementację, testowanie, wdrażanie, aż do etapu konserwacji oprogramowania. Zapoznanie się z podstawowymi zagadnieniami dotyczącymi kierowania projektem informatycznym, jak np. zarządzanie jakością.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K7_W03] wykazuje się znajomością zaawansowanych technik obliczeniowych, wspomagających pracę matematyka i rozumie ich ograniczenia	Zna metody inżynierii oprogramowania służące projektowaniu, tworzeniu, testowaniu i rozwojowi oprogramowania.	[SW1] Ocena wiedzy faktograficznej [SW2] Ocena wiedzy zawartej w prezentacji [SW3] Ocena wiedzy zawartej w opracowaniu tekstowym i projektowym
	[K7_K03] pracuje zespołowo, rozumie konieczność systematycznej pracy nad wszelkimi projektami, które mają długofalowy charakter, rozumie i docenia znaczenie uczciwości intelektualnej w działaniach własnych i innych osób; postępuje etycznie	Umie opisać wybrane procesy wytwarzania oprogramowania (m.in. kaskadowy, iteracyjny, zwinny). Potrafi pracować w zespole nad tworzeniem oprogramowania. Umie przeprowadzić inżynierię wymagań oprogramowania, dokonać analizy wymagań, zaprojektować oprogramowanie. Potrafi używać języka UML do modelowania wybranych aspektów oprogramowania. Zna i rozumie zasady etyki inżyniera oprogramowania.	[SK1] Ocena umiejętności pracy w grupie [SK3] Ocena umiejętności organizacji pracy [SK4] Ocena umiejętności komunikacji, w tym poprawności językowej [SK5] Ocena umiejętności rozwiązywania problemów występujących w praktyce
[K7_U10] rozumie matematyczne podstawy analizy algorytmów i procesów obliczeniowych, konstruuje algorytmy o dobrych własnościach numerycznych, służące do rozwiązywania typowych i nietypowych problemów matematycznych	Potrafi projektować i tworzyć oprogramowanie spełniające zadane wymagania.	[SU1] Ocena realizacji zadania [SU4] Ocena umiejętności korzystania z metod i narzędzi	
Treści przedmiotu	<p>Treści przedmiotu - wykład Wprowadzenie do inżynierii oprogramowania. Zasady etyczne. Procesy wytwarzania oprogramowania. Inżynieria wymagań. Metody strukturalne i metody obiektowe. Modelowanie przy wykorzystaniu UML. Testowanie oprogramowania. Zarządzanie projektami informatycznymi, zarządzanie jakością. Konserwacja oprogramowania. Systemy krytyczne. Manifest zwinnego oprogramowania (Agile Manifesto) i metodyka zwinna, w tym Scrum.</p> <p>Treści przedmiotu - laboratoria Stosowanie metod inżynierii oprogramowania do projektowania konkretnego oprogramowania, w tym opracowywanie diagramów UML, przygotowywanie historyjek użytkownika.</p> <p>Treści przedmiotu - projekt Praktyczne przygotowanie prototypów oprogramowania zgodnie z ustalonymi wymaganiami.</p>		
Wymagania wstępne i dodatkowe	Umiejętność pisania programów komputerowych, np. w Pythonie, C lub HTML/JavaScript.		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	kartkówki (pisemne, 10-minutowe)	60.0%	50.0%
	zadania laboratoryjne, projekt, referaty, prezentacje	60.0%	50.0%
Zalecana lista lektur	Podstawowa lista lektur	Krzysztof Sacha, Inżynieria oprogramowania, PWN 2010.	
		Mariusz Chrapko, Scrum. O zwinnym zarządzaniu projektami. Wydanie II rozszerzone, Helion 2014.	
	Uzupełniająca lista lektur	Ilan Sommerville, Inżynieria oprogramowania, WNT 2003. Roger S. Pressman, Praktyczne podejście do inżynierii oprogramowania, WNT 2004. Russ Miles, Kim Hamilton, UML 2.0. Wprowadzenie, Helion 2007. Kenneth S. Rubin. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Pearson Education, 2013.	
	Adresy eZasobów		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	Inżynieria wymagań. Procesy wytwarzania oprogramowania. Weryfikacja a zatwierdzanie. Metodyka zwinna i jej Manifest Agile. Zasady etyki inżyniera oprogramowania. Cechy INVEST historyjek użytkownika.		
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.