



Karta przedmiotu

Nazwa i kod przedmiotu	Inżynieria oprogramowania, PG_00063885						
Kierunek studiów	Informatyka						
Data rozpoczęcia studiów	październik 2024 r.	Rok akademicki realizacji przedmiotu			2025/2026		
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	2	Język wykładowy			polski		
Semestr studiów	4	Liczba punktów ECTS			3.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			egzamin		
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Inżynierii Oprogramowania						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr inż. Aleksander Jarzębowicz					
	Prowadzący zajęcia z przedmiotu	dr inż. Aleksander Jarzębowicz dr inż. Maciej Kucharski mgr inż. Małgorzata Pykała mgr inż. Rafał Dobrosielski mgr inż. Natalia Tyszka					
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	0.0	30.0	0.0	0.0	60
	W tym liczba godzin zajęć na odległość: 0.0						
	Adres kursu na platformie eNauczanie: <a href="https://enauczanie.pg.edu.pl/2025/course/view.php?id=4477">https://enauczanie.pg.edu.pl/2025/course/view.php?id=4477</a> Moodle ID: 4477 Inżynieria Oprogramowania 2025/2026 (semestr letni) <a href="https://enauczanie.pg.edu.pl/2025/course/view.php?id=4477">https://enauczanie.pg.edu.pl/2025/course/view.php?id=4477</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	2.0	13.0	75		
Cel przedmiotu	Przedmiot "Inżynieria Oprogramowania" jest ukierunkowany na przybliżenie zagadnień związanych z wytwarzaniem oprogramowania w warunkach przemysłowych: złożone systemy, przeznaczone dla rzeczywistego klienta, związane z określoną potrzebą biznesową i gwarancjami jakości, wytwarzane przez duże zespoły deweloperów.  W ramach wykładu omawiane są kluczowe obszary procesu wytwarzania: planowanie i zarządzanie, inżynieria wymagań, analiza i projektowanie, implementacja, testowanie, wdrożenie i utrzymanie, wspomaganie narzędziowe, praca zespołowa.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K6_U03] potrafi zaprojektować, zgodnie z zadaną specyfikacją, oraz wykonać typowe dla kierunku studiów proste urządzenie, obiekt, system lub zrealizować proces, używając odpowiednio dobranych metod, technik, narzędzi i materiałów, korzystając ze standardów i norm inżynierskich, stosując właściwe dla kierunków studiów technologie i wykorzystując doświadczenie zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską	Student potrafi wykonać modele analityczne i projektowe systemu informatycznego posługując się w tym celu narzędziami CASE (Computer Aided Software Engineering).	[SU1] Ocena realizacji zadania
	[K6_W44] zna i rozumie w zaawansowanym stopniu architektury, zasady projektowania oraz metody wsparcia sprzętowego i programowego dla lokalnych i rozproszonych systemów informatycznych, w tym systemów obliczeniowych, baz danych, sieci komputerowych i aplikacji informacyjnych, zasady współpracy człowieka z komputerem, a także działanie i kryteria oceny metod przetwarzania, składowania i przesyłania danych, w tym algorytmów obliczeniowych, sztucznej inteligencji i eksploracji danych oraz standardy i metody administrowania systemami informatycznymi, monitorowania zachodzących w nich procesów oraz uodporniania ich na niepożądane zjawiska i działania	Student rozumie pojęcie architektury systemu informatycznego, wie jakie zagadnienia wchodzą w zakres doboru i zdefiniowania architektury	[SW1] Ocena wiedzy faktograficznej
	[K6_W10] zna i rozumie w zaawansowanym stopniu podstawowe procesy zachodzące w cyklu życia urządzeń, obiektów i systemów technicznych oraz metody wspomagania procesów i funkcji, specyficzne dla kierunku studiów	Student zna modele cyklu życia oprogramowania oraz metodyki wytwarzania oprogramowania.	[SW1] Ocena wiedzy faktograficznej
	[K6_U09] potrafi dokonać krytycznej analizy sposobu funkcjonowania istniejących rozwiązań technicznych związanych z kierunkiem studiów i ocenić te rozwiązania, a także wykorzystać zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską doświadczenie związane z utrzymaniem urządzeń, obiektów i systemów technicznych typowych dla kierunku studiów	Student opracowuje wizję systemu informatycznego zawierającą krytyczną analizę obecnego sposobu funkcjonowania organizacji klienckiej oraz podstawowe wymagania i ograniczenia względem systemu.	[SU1] Ocena realizacji zadania

Treści przedmiotu	<p>Treści przedmiotu - wykład</p> <ol style="list-style-type: none"> <li>1. Wprowadzenie do przedmiotu</li> <li>2. Zakres i przedmiot inżynierii oprogramowania. Podstawowe motywacje i pojęcia.</li> <li>3. Wstępne założenia projektu i dokument Wizji systemu.</li> <li>4. Obszary działania inżynierii oprogramowania</li> <li>5. Inżynieria wymagań: pozyskiwanie wymagań</li> <li>6. Inżynieria wymagań: analizowanie, specyfikowanie i walidacja wymagań</li> <li>7. Pojęcie modelowania konceptualnego. Języki specyfikacji i modelowania.</li> <li>8. Przypadki użycia</li> <li>9. Obiektowe podejście do analizy systemu w UML</li> <li>10. Modelowanie logicznej struktury systemu: diagramy klas</li> <li>11. Modelowanie struktury: inne diagramy struktury</li> <li>12. Modelowanie dynamiki: diagramy sekwencji i komunikacji</li> <li>13. Modelowanie dynamiki: diagramy czynności i stanów</li> <li>14. Projektowanie: architektura systemu</li> <li>15. Projektowanie: Projekt ogólny (wysokiego poziomu) i projekt klas (szczegółowy)</li> <li>16. Zagadnienia jakości. Metryki projektowania obiektowego.</li> <li>17. Zagadnienia software reuse</li> <li>18. Klasyczne wzorce projektowe</li> <li>19. Inne rodzaje wzorców (wzorce aplikacji internetowych, wzorce analityczne, architektoniczne)</li> <li>20. Projektowanie interfejsu użytkownika: motywacje, pojęcia, techniki</li> <li>21. Projektowanie interfejsu użytkownika: heurystyki Nielsena i przykłady</li> <li>22. Testowanie: pojęcia, umiejscowienie w procesie wytwarzania</li> <li>23. Testowanie: techniki (czarna i biała skrzynka), poziomy testowania, zarządzanie testowaniem</li> <li>24. Przeglądy i inspekcje oprogramowania</li> <li>25. Wdrażanie oprogramowania</li> <li>26. Eksploatacja i utrzymanie oprogramowania</li> <li>27. Zarządzanie konfiguracją i zmianą</li> <li>28. Klasyczny cykl życia oprogramowania</li> <li>29. Nieklasyczne cykle życia i modele wytwarzania oprogramowania</li> <li>30. Dobór modelu wytwarzania do specyfiki projektu</li> <li>31. Zarys problematyki zarządzania projektem informatycznym</li> <li>32. Metodyki wytwórcze i zarządcze</li> <li>33. Charakterystyka metodyk sterowanych planem i zwinnych</li> </ol> <p>Treści przedmiotu - laboratoria</p> <ol style="list-style-type: none"> <li>1. Opracowanie dokumentu wstępnych założeń systemu (Wizja systemu)</li> <li>2. Przygotowanie modelu przypadków użycia systemu</li> <li>3. Przygotowanie analitycznego diagramu klas UML</li> <li>4. Przygotowanie diagramów dynamicznych UML (sekwencji, komunikacji, czynności, stanów) dla wybranych elementów systemu</li> </ol>		
Wymagania wstępne i dodatkowe	<p>Obowiązkowa obecność na zajęciach laboratoryjnych. Konieczne oddanie i akceptacja wszystkich zadań laboratoryjnych. Oddawanie zadań z opóźnieniem skutkuje punktami ujemnymi. Do egzaminu dopuszczeni są jedynie studenci, którzy zaliczyli laboratorium.</p>		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Laboratorium (zadania i sprawdziany)	50.0%	50.0%
	Egzamin pisemny	50.0%	50.0%
Zalecana lista lektur	Podstawowa lista lektur	<ol style="list-style-type: none"> <li>1. Sacha K., Inżynieria Oprogramowania, PWN, 2010</li> <li>2. Pressman R., Software Engineering: a Practitioners Approach, 8th edition, McGraw-Hill, 2014</li> <li>3. Sommerville I., Software Engineering, 9th edition, Addison-Wesley, 2010</li> <li>4. Maciaszek L.: Requirements analysis and system design, Addison-Wesley, 2007</li> <li>5. Booch G., Rumbaugh J., Jacobsen I.: UML przewodnik użytkownika, WNT, 2002</li> <li>6. Fowler M., Scott K.: UML w kropelce 2.0 (ang. UML distilled), Lupus 2005</li> <li>7. McLaughlin B., Pollice G., West D., Head First: Object-Oriented Analysis and Design, Edycja polska (Rusz głową!), Helion, 2008</li> </ol>	
	Uzupełniająca lista lektur	Nie ma wymagań	
	Adresy eZasobów		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania			
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.