

Karta przedmiotu

Nazwa i kod przedmiotu	Inżynieria oprogramowania, PG_00143568						
Kierunek studiów	Informatyka (O)						
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		
Forma studiów	stacjonarne		Sposób realizacji		na uczelni		
Rok studiów	2		Język wykładowy		polski polski		
Semestr studiów	4		Liczba punktów ECTS		3.0		
Profil kształcenia	ogólnoakademicki		Forma zaliczenia		zaliczenie		
Jednostka prowadząca							
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr inż. Stanisław Witkowski				
	Prowadzący zajęcia z przedmiotu						
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	15.0	0.0	0.0	30
	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	30		0.0		45.0	75
Cel przedmiotu	Celem przedmiotu jest zapoznanie studentów z podstawowymi problemami, metodami, technikami i narzędziami produkcji oprogramowania wysokiej jakości. Przedstawienie różnych modeli cyklu życia oprogramowania, metod wytwarzania oprogramowania, podstaw dokumentacji, analizy wymagań, projektowania, testowania oprogramowania. Omówienie zarządzania analizą wymagań, tworzenia specyfikacji ze zdefiniowanymi metrykami. Omawiane zagadnienia przedstawiane z naciskiem na obiektywne podejście do projektowania i modelowanie systemu za pomocą języka UML.						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu		Sposób weryfikacji i oceny efektu		
	[INFL3_W04] ma uporządkowaną wiedzę w zakresie inżynierii oprogramowania, specyfikacji, walidacji i weryfikacji oprogramowania oraz narzędzi wspomagających proces wytwarzania oprogramowania		Student posiada wiedzę odnośnie podstaw wytwarzania dokumentacji, analizy wymagań, projektowania, testowania oprogramowania		[SW4] test/egzamin - ustny lub pisemny		
	[INFL3_U04] potrafi pracować w zespole informatyków, zarządzać swoim czasem oraz podejmować zobowiązania i dotrzymywać terminy, porozumiewać się przy użyciu różnych technik w tym z wykorzystaniem dedykowanych narzędzi		Student posiada podstawowe umiejętności komunikacji w zespole informatycznym		[SU2] prezentacja/projekt/referat/ raport		
	[INFL3_U06] potrafi projektować, tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz adekwatnych wzorców		Potrafi zarządzać analizą wymagań; potrafi tworzyć specyfikację ze zdefiniowanymi metrykami;t potrafi utworzyć projekt oprogramowania za pomocą języka UML		[SU2] prezentacja/projekt/referat/ raport		

Treści przedmiotu	1. Wprowadzenie; motywacje dla systematycznego wytwarzania oprogramowania. 2. Cykl życia oprogramowania, fazy cyklu klasycznego, miejsce analizy i projektowania w cyklu wytwarzania. 3. Wymagania. Kategorie, pozyskiwanie, analiza i specyfikacja wymagań. Weryfikacja i walidacja; testowanie akceptacyjne. 4. Modelowanie; modelowanie świata a modelowanie systemu, miejsce modelu w analizie i projektowaniu. 5. Przypadki użycia. Diagram, interpretacja, opis ustrukturalizowany, przykłady. 6. Przegląd podstawowych zasad i pojęć paradygmatu obiektowego: obiekt, klasyfikacja, agregacja, dziedzicznie, komunikacja. Obiektość a naturalny dla człowieka sposób postrzegania rzeczywistości. 7. Wstęp do metodyki UML: model klas, model dynamiczny. Związki pomiędzy modelami. 8. Obiekty, klasy, atrybuty, operacje. Związki pomiędzy klasami (asocjacje). Związki jako klasy. Role. Agregacja jako szczególny przypadek związku. Semantyka agregacji. Propagacja operacji jako kryterium agregacji. Przykłady. 9. Dziedziczenie: specjalizacja i generalizacja. Hierarchia dziedziczenia. Redefiniowanie właściwości obiektów w dół hierarchii dziedziczenia, przesłanianie Klasy abstrakcyjne. Obiekt jako wystąpienie klasy i wszystkich jej nadklas. 10. Jak tworzyć model klas? Przykład tworzenia modelu klas. 11. Model dynamiczny: zdarzenia, czynności, akcje. Diagramy sekwencji (interakcji). Powiązanie z m innymi modelami. Jak tworzyć diagram sekwencji. Przykład. 12. Diagram stanów. Stany: proste, złożone, współbieżne. dziedziczenie stanów. Przejścia między stanami. Powiązanie z modelem klas. Jak tworzyć diagram model dynamiczny? Przykład tworzenia modelu dynamicznego. 13. Przejście od modelu do projektu. Podstawowe fazy projektu obiektowego. Podział na podsystemy i moduły, identyfikacja współbieżności, przydział procesorów, przydział zasobów, priorytety. 14. Przejście od projektu do programu.		
Wymagania wstępne i dodatkowe			
Sposoby i kryteria oceniania osiąganych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	projekt	50.0%	50.0%
	kolowium	50.0%	50.0%
Zalecana lista lektur	Podstawowa lista lektur	Materiały wykładów (umieszczone na Portalu Edukacyjnym) I. Sommerville: Inżynieria oprogramowania, WNT 2003Dumnicki R., Kasprzyk A., Kozłowski M.: Analiza i projektowanie obiektowe. Helion, 1998 Eriksson H-E, Penker M.: UML Toolkit. Wiley Computer Publishing, John Wiley & Sons, Inc. 1998 Pressman R.S.: Software Engineering. A Practitioners Approach. McGraw-Hill, Inc. 1992.	
	Uzupełniająca lista lektur	Booch G., Rumbaugh J., Jacobson I.: UML podręcznik użytkownika. WNT, 2001 Eriksson H-E, Penker M.: UML Toolkit. Wiley Computer Publishing, John Wiley & Sons, Inc. 1998.	
	Adresy eZasobów	Uzupełniające Adresy na platformie eNauczanie:	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	Dokonaj analizy wymagań dla zadanego przypadku i opracuj odpowiednie diagramy w języku UML/BPMN		
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.