Uniwersytet Gdański

Karta przedmiotu

Nazwa i kod przedmiotu	Inżynieria oprogramowania, PG_00198498		
Kierunek studiów	Informatyka (O)		
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu	2027/2028
Poziom kształcenia	I stopnia - licencjackie	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	5.0
Profil kształcenia	ogólnoakademicki	Forma zaliczenia	zaliczenie
Jednostka prowadząca			
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr Adam Kostulak	
	Prowadzący zajęcia z przedmiotu		

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						

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	0.0	65.0	125

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 obiektowe podejście do projektowania i modelowanie systemu za pomocą języka UML.

Efekty uczenia się przedmiotu

Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
[INFOL3_U06] potrafi dobrać i stosować właściwe metody i narzędzia informatyczne do rozwiązywania złożonych problemów, potrafi wykorzystać posiadaną wiedzę poprzez właściwy dobór źródeł i informacji z nich pochodzących, dokonywanie oceny, krytycznej analizy i syntezy tych informacji	Student potrafi pozyskać i wyspecyfikować wymagania, zamodelować system w języku UML (przypadki użycia, model klas, diagramy sekwencji i stanów) oraz dobrać właściwe metody i narzędzia do analizy, projektowania i testowania oprogramowania, korzystając z dokumentacji i innych źródeł.	[SU2] prezentacja/projekt/referat/ raport [SU4] test/egzamin - ustny lub pisemny
[INFOL3_W04] zna i rozumie zaawansowane zagadnienia w zakresie inżynierii oprogramowania, specyfikacji, walidacji i weryfikacji oprogramowania oraz narzędzi wspomagających proces wytwarzania oprogramowania	Student posiada wiedzę w zakresie modeli cyklu życia oprogramowania, pozyskiwania, analizy i specyfikacji wymagań, weryfikacji i walidacji oraz testowania oprogramowania; zna podstawy obiektowego podejścia do analizy i projektowania oraz modelowania systemu za pomocą języka UML (model klas i model dynamiczny), a także zasady przejścia od modelu do projektu i programu.	[SW4] test/egzamin - ustny lub pisemny [SW2] 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. Obiektowość 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	51.0%	50.0%
kolowium	51.0%	50.0%

Zalecana lista lektur

Podstawowa lista lektur	• I. Sommerville, *Inżynieria oprogramowania*, WNT/PWN. • K. Sacha, *Inżynieria oprogramowania*, PWN. • G. Booch, J. Rumbaugh, I. Jacobson, *UML. Przewodnik użytkownika*, WNT.
Uzupełniająca lista lektur	• M. Fowler, *UML w kropelce*, Oficyna Wydawnicza LTP. • S. Wrycza, B. Marcinkowski, K. Wyrzykowski, *Język UML 2.0 w modelowaniu systemów informatycznych*, Helion. • C. Larman, *Stosowanie UML i wzorców projektowych*, Helion.
Adresy eZasobów	

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.